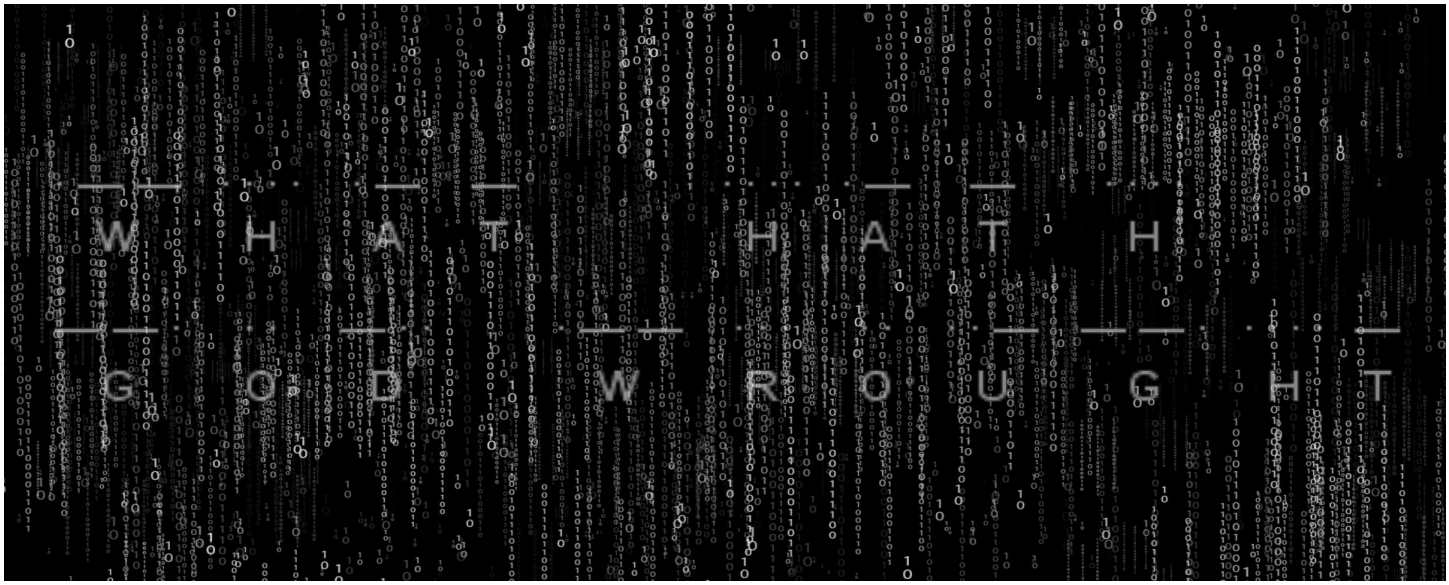
ΣΧΟΛΗ ΕΦΑΡΜΟΣΜΕΝΩΝ ΜΑΘΗΜΑΤΙΚΩΝ
ΚΑΙ ΦΥΣΙΚΩΝ ΕΠΙΣΤΗΜΩΝ
ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

Μέτρα πληροφορίας και συμπίεση δεδομένων χωρίς απώλειες.

Διπλωματική Εργασία

Προκόπου-Χουλιάρα-Μαρία-Ιωάννα



Επιβλέπων

Μιχάλης Λουλάκης, Αναπληρωτής Καθηγητής ΕΜΠ

Αθήνα 2020

ΣΧΟΛΗ ΕΦΑΡΜΟΣΜΕΝΩΝ ΜΑΘΗΜΑΤΙΚΩΝ
ΚΑΙ ΦΥΣΙΚΩΝ ΕΠΙΣΤΗΜΩΝ
ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

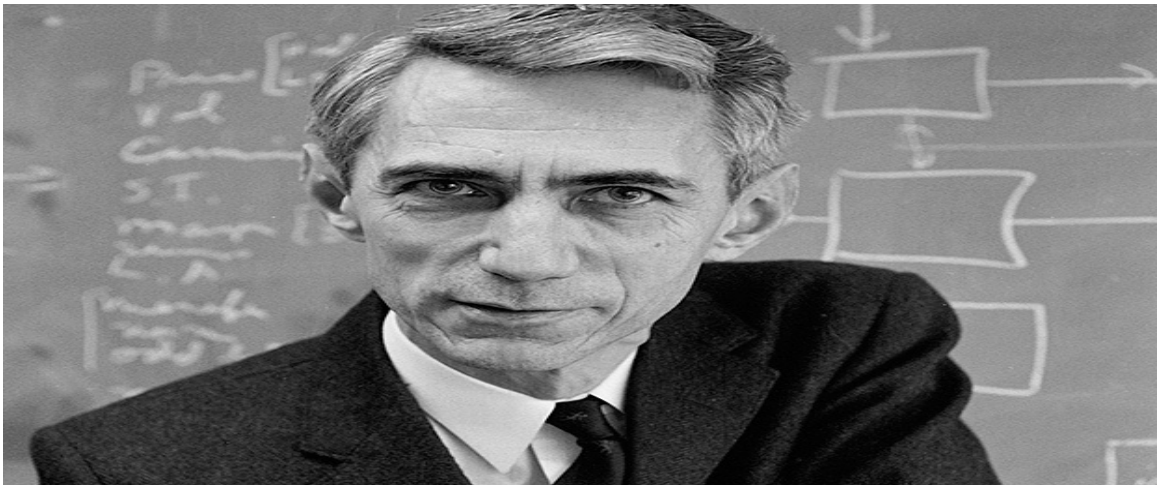
Μέτρα πληροφορίας και συμπίεση δεδομένων χωρίς απώλειες.

Διπλωματική Εργασία

Πρκόπου-Χουλιάρα-Μαρία-Ιωάννα

Επιβλέπων

Μιχάλης Λουλάκης, Αναπληρωτής Καθηγητής ΕΜΠ



Τριμελής Εξεταστική Επιτροπή

Μιχάλης Λουλάκης, Αναπληρωτής Καθηγητής ΕΜΠ
Στεφανέας Πέτρος, Επίκουρος Καθηγητής ΕΜΠ
Αριστείδης Παγουρτζής , Αναπληρωτής Καθηγητής ΕΜΠ

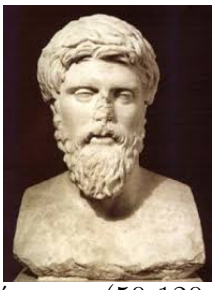
Αθήνα, Σεπτέμβριος 2020

Ευχαριστίες

Η παρούσα διπλωματική εργασία εκπονήθηκε υπό την επίβλεψη του Αναπληρωτή Καθηγητή ΕΜΠ, Μιχάλη Λουλάκη. Θα ήθελα να τον ευχαριστήσω για την τόσο καλή συνεργασία που είχα μαζί του. Ο κ. Λουλάκης από την πρώτη στιγμή μου έδωσε την ελευθερία να επιλέξω και να επεξεργαστώ ένα θέμα που αγαπάω πάρα πολύ. Ήταν παρών σε επιστημονικό και ανθρώπινο επίπεδο για να λύσει κάθε απορία μου αλλά και να με συμβουλεύσει σε κάθε τέλμα που συναντούσα.

Ακόμη θα ήθελα να ευχαριστήσω την μητέρα μου Βαρβάρα, τον πατέρα μου Βασίλη και τον αδερφό μου Φώτη για την ανιδιοτελή αγάπη και στήριξη τους τόσα χρόνια.

Επίσης θα ήθελα να ευχαριστήσω δύο μαθηματικούς η βοήθεια των οποίων ήταν καθοριστική για την εισαγωγή μου στη ΣΕΜΦΕ και την αγάπη μου για τα μαθηματικά. Ευχαριστώ ειλικρινά τον Άρη Αραμπατζή καθηγητή μου στο $21^ο$ λύκειο Αθηνών και τον Παντούλα Γεώργιο. Ολοκληρώνοντας θα ήθελα να ευχαριστήσω την φίλη μου από τα φοιτητικά χρόνια Λίβια τόσο για την παρέα της όσο και την απλόχερη στήριξη της όλα αυτά τα χρόνια.



“Το μυαλό δεν είναι ένα δοχείο που πρέπει να γεμίσει είναι μία φωτιά που πρέπει να ανάψει”.

Πλούταρχος(50-120μ.Χ)

Περίληψη

Η παρούσα διπλωματική εκπονήθηκε υπό την επίβλεψη του Αν. Καθηγητή ΕΜΠ, Μιχάλη Λουλάκη. Σκοπός της είναι να παρουσιαστούν τα βασικά μέτρα πληροφορίας για διακριτές τυχαίες μεταβλητές καθώς και οι βασικές τεχνικές συμπίεσης δεδομένων χωρίς απώλειες

Στο πρώτο κεφάλαιο εισάγεται η έννοια της εντροπίας της πληροφορίας για μία διακριτή τυχαία μεταβλητή που παίρνει τιμές σε ένα πεπερασμένο σύνολο. Η εισαγωγή γίνεται μέσα από παραδείγματα που ακολουθούν την λογική που επιβάλλει η ιστορική διαδρομή που μεσολάβησε μέχρι τον πλήρη ορισμό της από τον Claude E. Shannon το 1948. Αφού οριστεί η εντροπία ακολουθεί η γενίκευση της για ένα διάνυσμα διακριτών τυχαίων μεταβλητών μεγέθους n. Ακόμη δίνεται ο πρώτος κανόνας της αλυσίδας για ανεξάρτητες τυχαίες μεταβλητές με μία αλγοριθμική ανάλυση κόστους προκειμένου να αναδειχθεί η αξία χρήσης του. Το κεφάλαιο συνεχίζει με την εισαγωγή του δεσμευμένου μέτρου πληροφορίας και του αντίστοιχου κανόνα της αλυσίδας για εξαρτημένες τυχαίες μεταβλητές συνοδευόμενο από μία αλγοριθμική ανάλυση κόστους. Κατόπιν ορίζεται η απόσταση Kullback-Leibler με βάση την οποία παρακάτω ορίζονται ακόμα δύο μέτρα, το αμοιβαίο μέτρο πληροφορίας καθώς και το δευσευμένο μέτρο πληροφορίας. Το κεφάλαιο ολοκληρώνεται δίνοντας τα άνω και κάτω φράγματα για τα μέτρα πληροφορίας οπου είναι εφικτό μαζί με μία μελέτη για την κυρτότητα τους.

Στο δεύτερο κεφάλαιο ορίζονται μαθηματικώς οι πηγές πληροφορίας που παράγονται από διακριτές τυχαίες μεταβλητές. Το κεφάλαιο ξεκινάει με κάποια παραδείγματα τεχνητών γλωσσών τα οποία παράχθηκαν βάσει κώδικα γραμμένο σε python ο οποίος δίνεται στην τελευταία ενότητα. Μετά την εισαγωγή ορίζονται πρώτα οι πηγές χωρίς μνήμη και δίνεται ένα από τα κεντρικότερα θεωρήματα της θεωρίας πληροφορίας, το θεώρημα ασυμπτωτικής ισοκατανομής για πηγές χωρίς μνήμη. Το θεώρημα αυτό εξηγεί τον διαχωρισμό των δυνατών ακολουθιών μεγάλου μήκους σε δύο σύνολα όταν οι ακολουθίες παράγονται από μία πηγή χωρίς μνήμη . Το ένα σύνολο περιέχει τις ακολουθίες που θα παράγονται με μεγάλη πιθανότητα από την πηγή και συγκεντρώνει την περισσότερη μάζα πιθανότητας ενώ το άλλο περιέχει αυτές που είναι απίθανο να παραχθούν. Το σύνολο που περιέχει τις πιθανότερες ακολουθίες ορίζεται ως τυπικό σύνολο και τα μέλη του ως τυπικές ακολουθίες. Με βάση το θεώρημα ασυμπτωτικής ισοκατανομής εξάγονται κάποια πολύ χρήσιμα συμπεράσματα ως προς το μέγεθος και τις πιθανότητες των στοιχείων του τυπικού συνόλου. Το κεφάλαιο συνεχίζει ορίζοντας τον ρυθμό εντροπίας ως την ασυμπτωτική από κοινού εντροπία ανά σύμβολο για συμβολοσειρές θεωρητικά απείρου μεγέθους. Το μέγεθος αυτό είναι χρήσιμο για την μελέτη των πηγών με μνήμη οι οποίες ορίζονται ως μαρκοβιανές αλυσίδες πεπερασμένων καταστάσεων $k-$τάξης με στάσιμες κατανομές. Από τις πηγές με μνήμη πρώτα μελετώνται οι ομογενείς/στάσιμες πηγές. Αποδεικνύεται πρώτα ότι η δεσμευμένη εντροπία πηγών τέτοιους είδους συγκλίνει ασυμπτωτικά σε ένα μέγεθος που λέγεται εντροπία $k-$τάξης και είναι επί της ουσία το ασυμπτωτικό όριο της δεσμευμένης εντροπίας της στάσιμης πηγής όταν το μέγεθος της δέσμευσης τείνει στο άπειρο. Εν συνεχεία για τις στάσιμες πηγές αποδεικνύεται ότι η εντροπία $k-$τάξης ταυτίζεται με τον ρυθμό εντροπία της πηγής. Ολοκληρώνοντας το κεφάλαιο δίνεται το θεώρημα ασυμπτωτικής ισοκατανομής για εργοδικές και στάσιμε πηγές με μνήμη ή όπως είναι γνωστό το Shannon-Mcmillan-Breiman.

Στο τρίτο κεφάλαιο παρουσιάζεται η θεωρία γύρω από την συμπίεση. Αρχικά δίνονται οι ορισμοί για τις έννοιες κώδικας, μη ιδιόμορφος κώδικας, μοναδικά αποκωδικοποιήσιμος κώδικας και στιγμιαίος κώδικας για τις τιμές διακριτών τυχαίων μεταβλητών. Ακόμη συζητιούνται οι ομοιότητες και οι διαφορές μεταξύ των κωδίκων μέσα από εφαρμογές και παραδείγματα. Μετά την ολοκλήρωση των ορισμών και της ανάλυση τους δίνεται ένα θεώρημα που εξηγεί γιατί ένας στιγμιαίος κώδικας των τιμών μίας διακριτής τυχαίας μεταβλητής είναι αναγκαστικά και μοναδικά αποκωδικοποιήσιμος. Στην συνέχεια δίνονται τα γνωστότερα κριτήρια με βάση

τα οποία αποφασίζεται αν ένας κώδικας είναι μοναδικά αποκωδικοποιήσιμος η στιγμιαίος. Τα κριτήρια κατά σειρά είναι, ο αλγόριθμος των Sardinas-Patterson, η ανισότητα McMillan και η ανισότητα του Kraft. Στην τελευταία ενότητα του κεφαλαίου γίνεται μία ανάλυση των συνθηκών που πρέπει να ισχύουν για τα μήκη ενός στιγμιαίου κώδικα ώστε να είναι βέλτιστος. Με βάση την λύση του παραπάνω προβλήματος βελτιστοποίησης και χρησιμοποιώντας την απόσταση Kullback-Leibler από το πρώτο κεφάλαιο δίνουμε μία απόδειξη για τα όρια του μέσου μήκους κωδικοποίησης των τιμών της τυχαίας μεταβλητής σε σχέση με την εντροπία της. Τέλος μελετάται το πλεόνασμα κωδικοποίησης που προκύπτει στην πράξη καθώς και ο τρόπος που μπορούμε να το ελαχιστοποιήσουμε.

Στο τέταρτο κεφάλαιο μελετώνται τα θεωρήματα κωδικοποίησης του Shannon για διακριτά κανάλια με ή χωρίς θόρυβο. Στην πρώτη ενότητα ορίζονται τα διακριτά κανάλια χωρίς θόρυβο μαζί με την χωρητικότητα τους. Με βάση τους παραπάνω ορισμούς διατυπώνεται και αποδεικνύεται κάνοντας χρήση του θεωρήματος ασυμπτωματικής ισοκατανομής για πηγές χωρίς μνήμη το θεώρημα κωδικοποίηση πηγής του Shannon. Στην επόμενη ενότητα ορίζονται και μελετώνται τα διακριτά κανάλια χωρίς μνήμη με θόρυβο. Συγκεκριμένα επεκτείνεται το θεώρημα ασυμπτωτικής ισοκατανομής για πηγές χωρίς μνήμη σε θεώρημα ασυμπτωτικής ισοκατανομής για από κοινού ακολουθίες (joint asymptotic equipartition theorem) προκείμενου να περιγράφουν πιθανά ζεύγη ακολουθίων εισόδου και εξόδου. Με βάση το τελευταίο θεώρημα και τα μέτρα πληροφορίας του πρώτου κεφαλαίου ορίζεται με φυσιολογικό τρόπο η χωρητικότητα του καναλιού ως το μέγιστο αμοιβαίο μέτρο πληροφορίας μεταξύ των ακολουθιών εισόδου και εξόδου. Χρησιμοποιώντας τις παραπάνω θεωρητικές συνεισφορές αποδεικνύεται το θεώρημα κωδικοποίησης για διακριτά κανάλια χωρίς μνήμη με θόρυβο. Τέλος αποδεικνύεται ότι η χωρητικότητα δεν αυξάνεται στην περίπτωση που έχουμε ένα διακριτό κανάλι χωρίς μνήμη με θόρυβο και σχόλια (feedback).

Στο πέμπτο κεφάλαιο παρουσιάζονται οι βασικές τεχνικές συμπίεσης που χρησιμοποιούνται μέχρι και σήμερα. Στις δύο πρώτες ενότητες παρουσιάζεται η συμπίεση κατά Fano και κατά Shannon. Μπορεί αυτές οι τεχνικές συμπίεσης να μην παρουσιάζουν πρακτικό ενδιαφέρον αλλά έχουν ιστορικό και θεωρητικό καθώς οι αναλύσεις μέσου μήκους βασίζονται σε ιδιότητες της εντροπίας που έχουν διατυπωθεί στα προηγούμενα κεφάλαια. Στην τρίτη ενότητα παρουσιάζεται η συμπίεση Huffman μαζί με δύο φράγματα για το μέσο κώδικα που παράγει, το πρώτο φράγμα βασίζεται στην γνωστή τεχνική της άπληστης ανάλυσης ενώ το δεύτερο που δόθηκε από τον Gallager βασίζεται στην δομή του δένδρου Huffman σε συνδυασμό με τις ιδιότητες της εντροπίας. Στην τέταρτη ενότητα παρουσιάζεται η αριθμητική κωδικοποίηση και αποδεικνύεται ότι παράγει μη ιδιόμορφους και στιγμιαίους κώδικες που δεν είναι όμως βέλτιστοι. Στην έκτη, την έβδομη και όγδοη ενότητα παρουσιάζεται η οικογένεια των τριών βασικών LZ τεχνικών συμπίεσης. Συγκεκριμένα περιγράφονται κατά σειρά ο LZ77, ο LZ78 και ο LZW. Κάθε τεχνική συμπίεσης συνοδεύεται από την ασυμπτωτική ανάλυση μέσου μήκους κώδικα που παράγει. Τέλος στην ένατη ενότητα παρουσιάζεται ο μετασχηματισμός Burrows-Wheeler μαζί με την ασυμπτωτική ανάλυση μέσου μήκους. Η ανάλυση για τον Burrows-Wheeler βασίστηκε στην τεχνική της ανταγωνιστικής ανάλυσης που υπήρχε στην δημοσίευση "A simpler analysis of Burrows–Wheeler-based compression".

Στο έκτο κεφάλαιο παρουσιάζεται το πείραμα της διπλωματικής. Στόχος του πειράματος ήταν να μελετηθεί η συμπεριφορά των τριών αντιπροσωπευτικών τεχνικών συμπίεσης Huffman, LZ77 και Burrows-Wheeler. Το πείραμα γράφτηκε στην γλώσσα προγραμματισμού python, στο IDE pycharm. Για τους συμπιεστές LZ77 και Burrow-Wheeler χρησιμοποιήθηκαν τα έτοιμα πακέτα Gzip και Bzip2 της python ενώ ο Huffman υλοποιήθηκε ξεχωριστά. Προκειμένου να ελεγχθεί η συμπεριφορά των συμπιεστών σε αρχεία διαφορετικού περιεχόμενου και μεγέθους κατασκευάστηκε ένας web scraper ο οποίος σύλλεξε στίχους ελληνικών τραγουδιών από την ιστοσελίδα stixoi.info απο δημοφιλείς στιχουργούς. Με βάση τους στίχους που συγκεντρώθηκαν πραγματοποιήθηκαν δύο πειράματα. Στο πρώτο πείραμα ενοποιήθηκαν οι στίχοι που άνηκαν στο ίδιο στιχουργό. Τα αρχεία των ενοποιημένων στίχων χρησιμοποιήθηκαν για να αξιολογηθεί η συμπεριφορά των συμπιεστών όταν έρχονται αντιμέτωποι με αρχεία μεγάλου μεγέθους αλλά διαφορετικού περιεχομένου. Στο δεύτερο πείραμα το κάθε τραγούδι συμπιέστηκε ξεχωριστά βοηθώντας να μελετηθεί η συμπεριφορά των συμπιεστών όταν μειώνεται πολύ το μέγεθος του αρχείου που καλούνται να συμπιέσουν.

Abstract

This thesis was prepared under the supervision of the Associate Professor of NTUA, Michail Loulakis. Its purpose is to present the basic information measures for discrete random variables as well as the basic data lossless compression techniques.

The first chapter introduces the concept of information entropy for a discrete random variable that takes values in a finite set. The introduction is made through examples that follow the logic imposed by the historical path that mediated until its complete definition by Claude E. Shannon in 1948. Once entropy is defined, it is generalized to a vector of discrete random variables of magnitude n. The first rule of chain for independent random variables is also given with an algorithmic cost analysis in order to highlight its use value. The chapter continues with the introduction of the conditional information measure and the corresponding chain rule for dependent random variables accompanied by an algorithmic cost analysis. Then the Kullback-Leibler distance is defined, based on which two more meters are defined below, the mutual information measure as well as the conditionally mutual information measure. The chapter concludes by giving the upper and lower bound to information measures where possible along with a study of their convexity.

The second chapter defines mathematically the information sources produced by discrete random variables. The chapter begins with some examples of artificial languages which are produced based on code written in python which is given in the last section. After the introduction, the memoryless discrete sources are first defined and one of the most central theorems of information theory is given, the asymptotic equipartition property theorem (AEP) for discrete memoryless sources. This theorem explains the separation of possible long sequences into two sets when the sequences are generated from a discrete memoryless source. One set contains the sequences that will most likely be generated from the source and gathers the most mass of probability while the other contains those that are unlikely to be produced. The set containing the most probable sequences is defined as a typical set and its members as typical sequences. Based on the asymptotic equipartition property theorem, some very useful conclusions are drawn regarding the size and probabilities of the elements of the typical set. The chapter goes on to define entropy rate as the asymptotic joint entropy per symbol for strings of theoretically infinite size. This quantity is useful for studying information sources with memory which are defined as Markovian k-order finite state chains with stationary distributions. From information sources with memory, homogeneous/stationary sources are first studied. It is first shown that the conditional entropy of such sources converges asymptotically to a quantity called k-order entropy and is essentially the asymptotic limit of the conditional entropy of the stationary source when the magnitude of the binding tends to infinity. Then for stationary sources it turns out that the k-order entropy is identical to the source entropy rate. Concluding the chapter is given the theorem of asymptotic equipartition property theorem for ergodic and stationary sources with memory or as it is known Shannon-Mcmillan-Breiman.

The third chapter presents the theory around compression. Definitions are given for the concepts code, non-singular code, uniquely decodable code and instantaneous code for the values of discrete random variables. The similarities and differences between the codes are also discussed through applications and examples.After completing the definitions and analyzing them, a theorem is given that explains why an instantaneous code for the values of a discrete random variable is necessarily and uniquely decodable. The following are the best known criteria for deciding whether a code is uniquely decodable or instantaneous.

The criteria in order are the Sardinas-Patterson algorithm, the McMillan inequality and the Kraft inequality. In the last section of the chapter an analysis is made of the conditions that must apply to the lengths of an instant code in order to be optimal. Based on the solution of the above optimization problem and using the Kullback-Leibler distance from the first chapter we give a proof of the limits of the mean coding length of the values of the random variable in relation to its entropy. Finally, the code redudance that arises in practice is studied as well as the way we can minimize it.

In the fourth chapter, Shannon's coding theorems for discrete channels with or without noise are studied. The first section defines the discrete memoryless channels without noise along with their capacity. Based on the above definitions, Shannon's source coding theorem is formulated and proved using the asymptomatic equipartition property theorem for memoryless sources. In the next section the discrete memoryless channels with noise are defined and studied. Specifically, the asymptotic equipartition property theorem for memoryless sources is extended to the asymptotic equipartition property theorem for joint sequences (joint asymptotic equipartition theorem) in order to describe possible pairs of input and output sequences. Based on the last theorem and the information measures of the first chapter, the channel capacity is normally defined as the maximum mutual information measure between the input and output sequences. Using the above theoretical contributions, the coding theorem for discrete memoryless channels with noise is proved. Finally it turns out that the capacity does not increase in case we have a discrete memoryless channels with noise and feedback.

The fifth chapter presents the basic compression techniques used to date. The first two sections prsents Fano and Shannon compression schmes. These compression techniques may not be of practical interest butthey have historical and theoretical value as the analysis for the mean code length per symbol uses entropy properties introduced in previous chapters. The third section presents the Huffman compression along with two bounds to the average code length it produces, the first barrier is based on the well-known technique of greedy analysis while the second given by Gallager is based on the structure of the Huffman tree combined with entropy properties. The fourth section presents aritthmetic coding and proves that it produces non-singular and instantaneous codes that are not optimal. The sixth, seventh and eighth sections present the family of three basic LZ compression techniques. Specifically, the LZ77, LZ78 and LZW are described in order. Each compression technique is accompanied by the asymptotic analysis of the average code length per symbol it produces. Finally, the ninth section presents the Burrows-Wheeler transformation together with the asymptomatic mean length analysis. The analysis for Burrows-Wheeler was based on the competitive analysis technique found in the publication named "A simpler analysis of Burrows – Wheeler-based compression".

The sixth chapter presents the thesis experiment. The aim of the experiment was to study the behavior of the three representative compression techniques Huffman, LZ77 and Burrows-Wheeler. The experiment was written in the python programming language using IDE pycharm. For the LZ77 and Burrow-Wheeler compressors, python's ready-made Gzip and Bzip2 packages were used, while Huffman was implemented separately. In order to check the behavior of compressors in files of different content and size, a web scraper was made which collected lyrics of Greek songs from the website stixoi.info by popular lyricists. Based on the collected song lyrics, two experiments were performed. In the first experiment, the lyrics belonging to the same lyricist were concatinated. Unified lyrics files were used to evaluate the behavior of compressors when confronted with large files of different content. In the second experiment, each song was compressed separately to help study the behavior of the compressors when the size of the file they are called to compress is greatly reduced.

Περιεχόμενα

Κεφάλαιο 1

Μέτρα Πληροφορίας

1.1 Εντροπία τυχαίας μεταβλητής

Η καθημερινότητά μας χαρακτηρίζεται από μία συνεχόμενη λήψη και μετάδοση πληροφοριών, είτε μέσα στο πανεπιστήμιο ή στο σχολείο, διαβάζοντας μία εφημερίδα αλλά και ένα λογοτεχνικό βιβλίο, ακούγοντας μουσική αλλά και μία ενημερωτική ραδιοφωνική εκπομπή. Αυτή την πληθώρα διαφορετικών πληροφοριών ο εγκέφαλος μας τις επεξεργάζεται, τις κατατάσσει και αποθηκεύει τις σημαντικότερες. Η σημαντικότητα της έννοιας της πληροφορίας όμως απαντάται και στις επιστήμες. Ένα στατιστικός για παράδειγμα θα συλλέξει δεδομένα(πληροφορίες) προκειμένου να εξάγει συμπεράσματα για έναν η περισσότερους πληθυσμούς, ένας μαθηματικός θα συλλέξει ένα σύνολο μεταβλητών που περιγράφουν κάποιο υπό μελέτη σύστημα ώστε να μοντελοποιήσει την συμπεριφορά του. Ενώ όμως είναι τόσο σημαντική η έννοια της πληροφορίας όταν μας ζητείται ο ορισμός της σίγουρα δυσκολευόμαστε να την εξηγήσουμε. Θα μπορούσαμε μιλώντας σε καθημερινό επίπεδο να πούμε ότι πληροφορία είναι η γνώση που αποκτήθηκε μετά από την ερεύνα, την επεξεργασία ή ακόμη και την ανταλλαγή κάποιων δεδομένων.

Αφού λοιπόν δώσαμε μία απάντηση στο πρώτο ερώτημα, ας θέσουμε στους εαυτούς μας μία ακόμη ερώτηση. Ποιες από όλες τις πληροφορίες που προσλαμβάνουμε καθημερινά αποφασίζει ο εγκέφαλος μας να αποθηκεύσει; Σίγουρα θα είναι πρώτα αυτές που εξασφαλίζουν την επιβίωση μας και την ομαλή ολοκλήρωση τις καθημερινότητάς μας και έπειτα αυτές που θεωρούμε σημαντικές. Πως κρίνουμε όμως πότε μια πληροφορία είναι σημαντική; Αυτό δυστυχώς δεν μπορεί να έχει ενιαία απάντηση καθώς ο κάθε άνθρωπος ανάλογα με τα βιώματα, την ιδιοσυγκρασία και την ψυχοσύνθεση του, την θέση του στην κοινωνία θεωρεί διαφορετικά πράγματα σημαντικά. Για παράδειγμα ένας φιλότεχνος θεωρεί σημαντική την πληροφορία για μία έκθεση ζωγραφικής. Από την άλλη ένας άνθρωπος που ασχολείται με τα κοινωνικά και πολιτικά ζητήματα θεωρεί σημαντική μία πολιτική εξέλιξη. Ένα τρίτος άνθρωπος μπορεί να θεωρήσει σημαντικά και τα δύο. Άρα σε ανθρώπινο επίπεδο η σημαντικότητα της πληροφορίας ιεραρχείται από διαφορετικά για τον καθένα μας κριτήρια. Παρόλα αυτά σε ένα γενικό πλαίσιο μπορούμε να πούμε πως ένας άνθρωπος θα θεωρήσει σημαντική μία πληροφορία που θα του προσφέρει ένα βαθμό έκπληξης, δηλαδή ένα γεγονός ήταν αβέβαιη η πραγματοποίηση του, απλά σε κάθε άνθρωπο προκαλούν διαφορετικά πράγματα έκπληξη

Αφού ορίσαμε την πληροφορία για τους ανθρώπους, πάμε να σκεφτούμε πως μπορούμε να ορίσουμε την πληροφορία στην γλώσσα των μαθηματικών. Εδώ τα πράγματα είναι λιγάκι πιο δύσκολα. Ο βαθμός δυσκολίας έγκειται στο γεγονός ότι ένας μαθηματικός και γενικά οποιοσδήποτε επιστημονικός ορισμός δεν γίνεται να στηρίζεται σε προσωπικές ερμηνείες και συναισθηματικές καταστάσεις. Άρα όταν θα πάμε να ορίσουμε μαθηματικά την πληροφορία και μάλιστα να τη ποσοτικοποιήσουμε ώστε να δούμε πόσο σημαντική είναι θα πρέπει ο ορισμός αυτός να είναι κοινός για όλες τις εφαρμογές και να μην εξαρτάται από την οπτική γωνία του εκάστοτε επιστήμονα. Πως λοιπόν μπορεί να ποσοτικοποιηθεί η πληροφορία; Στην προηγούμενη παράγραφο που μιλήσαμε για την έννοια της πληροφορίας υπό το ανθρώπινο πρίσμα είπαμε ότι το επίπεδο σημαντικότητας της εξαρτάται ή μπορεί να ποσοτικοποιηθεί από το βαθμό έκπληξης που μας προκαλεί ή ακόμα καλύτερα από την αβεβαιότητα που περιέχεται στην πραγματοποίησή της. Άρα ένας τρόπος να ορίσουμε μαθηματικώς την

πληροφορία είναι να βρούμε τα μαθηματικά αντικείμενα που μοντελοποιούν την έκπληξη ή την αβεβαιότητα που υπάρχει για κάποιο ενδεχόμενο, τα οποία αντικείμενα δεν θα εξαρτώνται από ανθρώπινους παράγοντες. Ποια λοιπόν μπορεί να είναι αυτά τα αντικείμενα και κάτω από ποια κατάλληλη μαθηματική μορφοποίηση ποσοτικοποιούν την πληροφορία;

Φίλε αναγνώστη αν δυσκολεύεσαι να απαντήσεις σε αυτή την ερώτηση θα σε εμψυχώσουμε λέγοντας σου πως η συγκεκριμένη απορία ταλάνισε για δεκαετίες τον επιστημονικό κόσμο, συγκεκριμένα από τα τέλη του $19^{ου}$ μέχρι τα μέσα του $20^{ου}$ αιώνα που εμφανίστηκαν οι πρώτες ηλεκτρικές και απομακρυσμένες τηλεπικοινωνίες (τηλέγραφος, τηλέτυπος, ράδιο κ.λ.π). Την απάντηση όμως δεν θα την δώσουμε μονομιάς αλλά σταδιακά μέσα από παραδείγματα ακολουθώντας την συλλογιστική που επιβάλλεται από την ροή της ιστορίας. Οπότε προτείνουμε να προσδεθείτε γιατί το ταξίδι στην θεωρία της πληροφορίας μόλις ξεκίνησε!

Έστω δύο τυχαίες μεταβλητές X, Y που μπορούν να πάρουν ως τιμές τα σύμβολα {a,b,c,d} του αγγλικού αλφαβήτου και έχουν συναρτήσεις μάζας πιθανότητας (σ.μ.π) τις $P_X(x) = \{\frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4}\}$ και $P_Y(y) = \{\frac{1}{2}, \frac{1}{8}, \frac{1}{8}, \frac{1}{4}\}$ αντίστοιχα . Με βάση τις X, Y κατασκευάζουμε δύο στοχαστικές διαδικασίες $\{X_n\}_{n\in\mathbb{N}}, \{Y_n\}_{n\in\mathbb{N}}$, οι οποίες αποτελούνται από ανεξάρτητες και ισόνομες τυχαίες μεταβλητές, έτσι ώστε $X_n \sim P_X(x)$ και $Y_n \sim P_Y(y)$ $\forall n \in \mathbb{N}$. Χρησιμοποιώντας τις παραπάνω διαδικασίες μπορούμε να δημιουργήσουμε συμβολοσειρές της μορφής $\mathbf{X_1^n}$, $\mathbf{Y_1^n}$,[1] οι οποίες αποτελούνται από τα γράμματα {a,b,c,d}. Βέβαια το κάθε γράμμα εμφανίζεται με διαφορετική συχνότητα στην επιμέρους συμβολοσειρά. Για παράδειγμα σε μία ακολουθία που έχει παραχθεί από την $\{Y_n\}_{n\in\mathbb{N}}$, το γράμμα a θα εμφανίζεται συχνότερα από ότι σε αυτή που έχει δημιουργηθεί από την $\{X_n\}_{n\in\mathbb{N}}$. Δύο αντιπροσωπευτικά στιγμιότυπα μεγέθους 50 των παραπάνω διαδικασιών είναι τα εξής:

$$\{X_n\}_{n\in\mathbb{N}} : cbbbcaaddccdddcbddcbaddcadbccccdcbbccbadbacbbabcdc$$
$$\{Y_n\}_{n\in\mathbb{N}} : addcccaccaaaabbadcaaaabdbabdddbaabddbdaadaaaaaaddb$$

Ας υποθέσουμε ότι τα παραπάνω στιγμιότυπα αποτελούν μέρος ενός παιχνιδιού στο οποίο συμμετέχουμε. Οι κανόνες διατυπώνονται παρακάτω:

1. Το παιχνίδι ξεκινάει με δύο ακολουθίες αρχικού μεγέθους 50 που έχουν παραχθεί από τις στοχαστικές διαδικασίες $\{X_n\}_{n\in\mathbb{N}}$ και $\{Y_n\}_{n\in\mathbb{N}}$.

2. Κάθε παίχτης επιλέγει μία από τις δύο συμβολοσειρές, κάνει μία πρόβλεψη για το επόμενο γράμμα της και μετά ποντάρει έναν αριθμό από μάρκες.

3. Εν συνεχεία οι στοχαστικές διαδικασίες παράγουν το επόμενο σύμβολο.

4. Οι παίκτες που μάντεψαν σωστά το γράμμα της συμβολοσειράς που επέλεξαν, μοιράζονται τις μάρκες των παικτών που μάντεψαν λάθος το γράμμα της συγκεκριμένης συμβολοσειράς. Το μοίρασμα γίνεται ανάλογα με το ποσό που είχε ποντάρει ο κάθε παίκτης σε σχέση με το συνολικό που υπήρχε στην παρτίδα για την εν λόγω συμβολοσειρά. Δηλαδή όσα περισσότερα ποντάρει ένας παίκτης τόσα περισσότερα δικαιούται, αν κερδίσει.

5. Ως χαμένος ορίζεται ο παίκτης που δεν θα μπορέσει να πάρει μέρος σε κάποια παρτίδα επειδή θα έχει χάσει όλες τις μάρκες του.

Με βάση τα παραπάνω εσείς σε ποια από τις δύο στοχαστικές διαδικασίες θα ποντάρατε τις μάρκες σας και για ποιο σύμβολο; Παρατηρώντας τα αρχικά δείγματα των συμβολοσειρών αλλά και πως διαμορφώνονται καθώς εξελίσσονται οι παρτίδες μπορούμε, χρησιμοποιώντας τον νόμο των μεγάλων αριθμών(αφού οι τυχαίες μεταβλητές είναι ανεξάρτητες και ισόνομες), να υπολογίσουμε με μεγάλη ακρίβεια την πιθανότητα εμφάνισης του κάθε συμβόλου στην αντίστοιχη συμβολοσειρά, δηλαδή να προσεγγίσουμε τις σ.μ.π των X, Y. Έχοντας

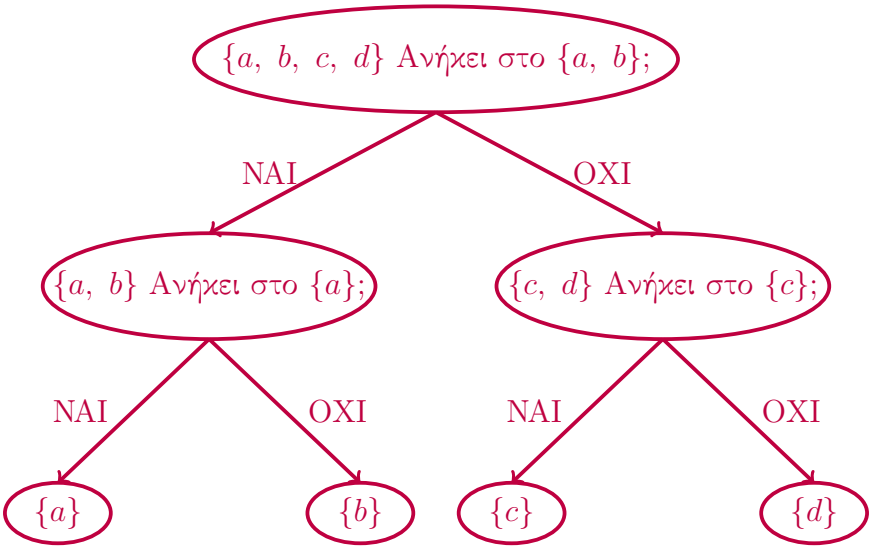
[1] Αν X μία τυχαία μεταβλητή που παίρνει τιμές στο πεπερασμένο σύνολο $\mathcal{X}$, τότε ως $\mathbf{X_1^n}$ ορίζουμε την πεπερασμένη ακολουθία τυχαίων μεταβλητών $X_1X_2\cdots X_n$. Στην παρούσα διπλωματική ότι αποτελεί ακολουθία ή διάνυσμα θα συμβολίζεται με έντονα γράμματα.

στα χέρια μας αυτές τις πληροφορίες είναι λογικό να πούμε ότι θα ποντάρουμε τις μάρκες μας στην δεύτερη στοχαστική διαδικασία καθώς είναι περισσότερο προβλέψιμη από ότι η πρώτη και μάλλον στο σύμβολο "a". Αν όμως στο παιχνίδι αντί να επιλέξουμε ανάμεσα σε δύο στοχαστικές διαδικασίες είχαμε στη διάθεση μας k, που είχαν παραχθεί από k τυχαίες μεταβλητές με διαφορετική σ.μ.π η κάθε μία, τι θα μπορούσαμε να κάνουμε;

Στην περίπτωση αυτή θα έπρεπε με κάποιον τρόπο να υπολογίσουμε την προβλεψιμότητα των διαφορετικών σ.μ.π των k τυχαίων μεταβλητών ώστε να τις συγκρίνουμε μεταξύ τους ώστε να αποφανθούμε ποια θα επιλέξουμε. Ένας απλός τρόπος για να επιτευχθεί το ζητούμενο είναι να σκεφτούμε την απάντηση στης εξής ερώτηση: Πόσες ερωτήσεις τύπου ΝΑΙ ή ΟΧΙ κατά μέσο όρο θα χρειάζονταν ώστε να βρούμε το επόμενο σύμβολο που θα παραχθεί; Προφανώς όσες περισσότερες ερωτήσεις έπρεπε να διατυπωθούν τόσο περισσότερο απρόβλεπτη θα ήταν η σ.μ.π της εν λόγω τυχαίας μεταβλητής άρα και η στοχαστική διαδικασία που παράχθηκε με βάση αυτή. Αντιλαμβανόμαστε λοιπόν ότι το πλήθος των ερωτήσεων μπορεί να αποτελέσει ένα μέτρο προβλεψιμότητας της εκάστοτε σ.μ.π. Το επόμενο ερώτημα που γεννάται είναι πώς μπορούμε να μετρήσουμε τον πλήθος των ερωτήσεων και με τι τρόπο πρέπει να γίνουν;

Μία πρώτη προσέγγιση θα ήταν να διατυπώσουμε μία ερώτηση για το κάθε γράμμα ξεχωριστά. Τότε και στις δύο περιπτώσεις θα χρειαζόμασταν τέσσερις ερωτήσεις, γεγονός που δεν μας βοηθάει να συγκρίνουμε την προβλεψιμότητα των δύο κατανομών. Επίσης μία τέτοια προσέγγιση για μικρά αλφάβητα μπορεί να είναι εφικτή αλλά για μεγαλύτερα είναι ιδιαίτερα αργή και καθόλου εύχρηστη. Αρχίζουμε να καταλαβαίνουμε ότι οι ερωτήσεις θα πρέπει να γίνουν με γνώμονα τις πιθανότητες των στοιχείων του $\mathcal{X} = \{a, b, c, d\}$, δηλαδή με βάση τις σ.μ.π $P_X(x)$ και $P_Y(y)$.

Θα ξεκινήσουμε την μελέτη μας από την τυχαία μεταβλητή X, η οποία αποτελείται από ισοπίθανα ενδεχόμενα. Από την θεωρία των αλγορίθμων γνωρίζουμε πως όταν χρειάζεται να απαντήσουμε κάποια ερώτηση αναζήτησης, ο πιο αποδοτικός τρόπος να το επιτύχουμε είναι χρησιμοποιώντας τον αλγόριθμο δυαδικής αναζήτησης. Προκειμένου να πραγματοποιήσουμε δυαδική αναζήτηση για το παρών πρόβλημα, χωρίζουμε τιμές που μπορεί να πάρει η τ.μ X σε δύο σύνολα έτσι ώστε το άθροισμα των στοιχείων κάθε συνόλου να είναι το ίδιο. Τα σύνολα που προήλθαν από τη παραπάνω διαμέριση χωρίζονται εκ νέου κατά τον ίδιο τρόπο. Η διαδικασία συνεχίζεται μέχρι το κάθε σύνολο που θα προκύψει να περιέχει μόνο ένα γράμμα. Τα μονοσύνολα που προέκυψαν αποτελούν τα φύλλα του δένδρου αναζήτησης.



Σχήμα 1.1: Δυαδική αναζήτηση με βάση την $f_X(x)$

Παρατηρούμε (Σχήμα 1.1) ότι έχουμε ένα πλήρες δυαδικό δένδρο, στο όποιο κάθε μονοπάτι που οδηγεί από τη ρίζα στα φύλλα έχει μήκος δύο. Το μήκος του μονοπατιού αποτελεί το πλήθος των ερωτήσεων που χρειάζονται προκειμένου να μαντέψουμε σωστά το γράμμα του εκάστοτε φύλου. Από την θεωρία των γραφημάτων γνωρίζουμε ότι ένα πλήρες δυαδικό δένδρο έχει ύψος $log_2(n)$. Άρα οι δύο ερωτήσεις που χρειάζονται ανά σύμβολο είναι επί της ουσίας το ύψος $log_2(4)$, όπου το τέσσερα αποτελεί τον πληθάριθμο του πεδίου

τιμών της τυχαίας μεταβλητής X. Το τέσσερα όμως συμβολίζει και κάτι ακόμα, είναι η ποσότητα $\frac{1}{Pr[X=x]}$, με $Pr[X=x]=\frac{1}{4}$. Όταν έχουμε ισοπίθανα ενδεχόμενα το κλάσμα $\frac{1}{Pr[X=x]}$ μας δίνει το πλήθος των ενδεχομένων. Αν θέλουμε να ερμηνεύσουμε τα ισοπίθανα ενδεχόμενα γεωμετρικά μπορούμε να πούμε ότι n ισοπίθανα ενδεχόμενα διαμερίζουν τον χώρο $\mathcal{X}$ σε n ισεμβαδικά χωρία εμβαδού ίσο με $\frac{1}{n}$.

Στην προκειμένη περίπτωση η $P_X(x)$ διαμερίζει το $\mathcal{X}$ σε 4 τετράγωνα εμβαδού $\frac{1}{4}$ το κάθε ένα όπως φαίνεται στο Σχήμα 1.3(α′). Το κάθε τετράγωνο αντιστοιχεί σε ένα γράμμα. Άρα το πρόβλημα της εύρεσης του μέσου όρου των ερωτήσεων που απαιτούνται για το επόμενο γράμμα, είναι ισοδύναμο με την εύρεση του μέσου όρου των ερωτήσεων προκειμένου να βρούμε ποιο τετράγωνο του $\mathcal{X}$ θα επιλεχθεί. Καταλαβαίνουμε λοιπόν ότι σε κάθε βήμα της δυαδικής αναζήτησης, δεν χωρίζουμε απλά το $\{a,b,c,d\}$ σε δύο ισοπληθικά σύνολα, χωρίζουμε επί της ουσίας τον χώρο $\mathcal{X}$ σε δύο ισεμβαδικά χωρία. Η διαμέριση του κάθε υποσυνόλου, που παράγεται κατά τη διάρκεια της δυαδικής αναζήτησης, συνεχίζεται μέχρι να βρούμε ένα χωρίο (υποσύνολο) που περιέχει μόνο ένα τετράγωνο. Άρα το μήκος του μονοπατιού προς κάθε σύμβολο, εκφράζει το πλήθος των διαμερίσεων που θα χρειαστούν ώστε να βρεθώ στο τετράγωνο που ανήκει το σύμβολο που επιλέχθηκε. Επειδή για κάθε σύμβολο το μήκος του μονοπατιού είναι δύο τότε και ο μέσο όρος των ερωτήσεων ανά σύμβολο θα είναι δύο. Το παραπάνω συμπέρασμα είναι απόλυτα λογικό αν σκεφτούμε ότι με δύο διαμερίσεις μπορούμε να μαντέψουμε σωστά οποιοδήποτε σύμβολο κι αν έχει επιλεχθεί.

Επισημαίνουμε ότι το δένδρο αναζήτησης δεν χρειάζεται να είναι πάντα δυαδικό. Αν είχαμε 9 ισοπίθανα δεδομένα θα μπορούσαμε να χρησιμοποιήσουμε ένα τριαδικό δένδρο αναζήτησης. Τότε το πλήθος των ερωτήσεων ανά σύμβολο καθώς και ο μέσος όρος τους θα ήταν $log_3(9)=2$ όπως φαίνεται στο Σχήμα 1.2(α′). Αν για τα ίδια δεδομένα χρησιμοποιούσαμε ένα δυαδικό δένδρο αναζήτησης τότε ο μέσος όρος των ερωτήσεων θα ήταν $\frac{3*7+2*4}{9}=3.2222$, διότι έχουμε 7 μονοπάτια μήκους τρία και δύο μήκους τέσσερα όπως φαίνεται στο Σχήμα 1.2(β′). Ο αριθμός αυτός είναι ασυμπτωτικά ίδιος με το κόστος της δυαδικής αναζήτησης $O(log_2(9))=3.1699$, δηλαδή 3.2 ερωτήσεις/σύμβολο. Από τα παραπάνω καταλαβαίνουμε ότι δεν έχει σημασία αν θα χρησιμοποιήσω ένα δυαδικό, τριαδικό ή $m-$ αδικό δένδρο αναζήτησης, δηλαδή τι βάση θα έχει ο λογάριθμος. Η βάση αποτελεί μία τεχνική λεπτομέρεια, η οποία θα γίνει κατανοητή παρακάτω. Σημασία έχει ότι για να διατυπώσω με αποδοτικό τρόπο τις ερωτήσεις, πρέπει να χωρίσω το αρχικό σύνολο μεγέθους n σε m σύνολα που έχουν περίπου το ίδιο πλήθος στοιχείων. Έπειτα να συνεχίσω αυτήν την διαμέριση μέχρι να φτάσω σε ένα σύνολο που περιέχει ένα μοναδικό σύμβολο. Η απάντηση στο πόσες διαμερίσεις/ερωτήσεις θα χρειαστούν κατά μέσο όρο μέχρι να φτάσω σε κάποιο μονοσύνολο δίνεται από την ποσότητα $log_m(n)$. Ξέρουμε πλέον ότι όταν έχουμε n ισοπίθανα ενδεχόμενα:

$$\boxed{E[\#\text{Ερωτήσεων/σύμβολο}]=\log(n)}$$

Στο σημείο αυτό να αναφέρουμε το ιστορικό γεγονός ότι η παραπάνω σχέση που μόλις εξάγαμε αποτελεί την **εντροπία για ισοπίθανα ενδεχόμενα** και η τυπική της μορφή είναι η:

$$H(X)=\log|\mathcal{X}|$$

Η εντροπία για ισοπίθανα ενδεχόμενα είναι επί της ουσίας η ελάχιστη μέση πληροφορία που χρειαζόμαστε για να περιγράψουμε την τυχαία μεταβλητή X που αποτελείται από ισοπίθανα ενδεχόμενα σε σχέση με κάποιο άλλο αντικείμενο. Στην προκειμένη περίπτωση τα "άλλα αντικείμενα" που επιλέξαμε για να περιγράψουμε την X ήταν το ελάχιστο μέσο πλήθος ερωτήσεων τύπου ΝΑΙ/ΟΧΙ, το ελάχιστο μέσο πλήθος των διαμερίσεων του χώρου $\mathcal{X}$ προκειμένου να βρεθούμε σε κάποιο από τα μονοσύνολα $\{a\}$, $\{b\}$, $\{c\}$, $\{d\}$ και το ελάχιστο μέσο μήκος μονοπατιού για να πάμε από τη ρίζα σε κάποιο από τα φύλλα που θα περιέχει την εκάστοτε τιμή της τυχαίας μεταβλητής που μας ενδιαφέρει.

Η εντροπία για ισοπίθανα ενδεχόμενα ανακαλύφθηκε το 1928 από τον R.V.L Hartley[2], στην εργασία του

[2]Har28.

Transmission of information. Στην ουσία ο Hartley ήταν ο πρώτος που προσέγγισε μαθηματικά την έννοια της πληροφορίας και κατάφερε να την ποσοτικοποιήσει για την περίπτωση των ισοπίθανων ενδεχομένων.

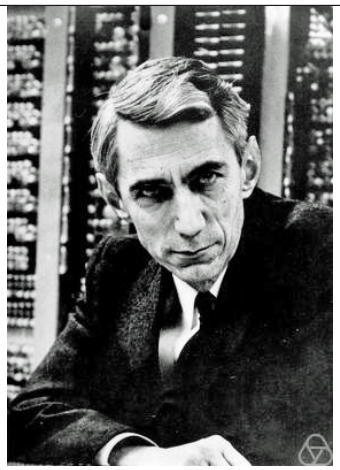
Παρόλα αυτά η συγκεκριμένη εργασία δεν ήταν η πρώτη καταγεγραμμένη προσπάθεια στην οποία αποπειράθηκε κάποιος να ορίσει με έναν μαθηματικό τρόπο την πληροφορία. Η πρώτη νύξη για την ανάγκη ορισμού ενός μέτρου της πληροφορίας έγινε το 1924 από τον Harry Nyquist στην εργασία του Certain factors affecting telegraph speed[3]. Σε αυτή την εργασία ο Nyquist θέλησε να εξάγει ένα τύπο με τον οποίο θα σύγκρινε διάφορου κώδικες που χρησιμοποιούνταν για τον τηλέγραφο εκείνης της εποχής. Γενικά για να μεταδοθεί ένα μήνυμα μέσω ενός κυκλώματος τηλεγράφησης χρειαζόταν κάθε χαρακτήρας να αναπαρασταθεί από μία ακολουθία συμβόλων (π.χ κώδικας Morse) η οποία μετά κωδικοποιούνταν με τη χρήση των κυματομορφών που παράγονταν από το ηλεκτρικό κύκλωμα τηλεγράφησης. Ο τύπος ήταν ο παρακάτω:

$$W = K \log M$$

όπου W ήταν ο ρυθμός μετάδοσης της πληροφορίας, K μία σταθερά που βασιζόταν στα φυσικά χαρακτηριστά του κυκλώματος τηλεγράφησης και M ήταν το πλήθος των ρευμάτων που συμμετείχαν στο κύκλωμα τηλεγράφησης. Αυτό που βρήκε ο Nyquist ήταν πως αν κατάφερναν να διατηρήσουν σταθερή την ταχύτητα γραμμής του κυκλώματος και αύξαναν ταυτόχρονα το πλήθος των ρευμάτων που συμμετέχουν σε αυτό τότε θα υπήρχε μία σημαντική αύξηση στο πλήθος της πληροφορίας που μπορούσαν να μεταδώσουν. Ο Nyquist απέδειξε τον παραπάνω τύπο στο παράρτημα Β της εργασίας του χωρίς όμως να διατυπώσει τις υποθέσεις που τον οδήγησαν στην εξαγωγή του.

(α′) Harry Nyquist

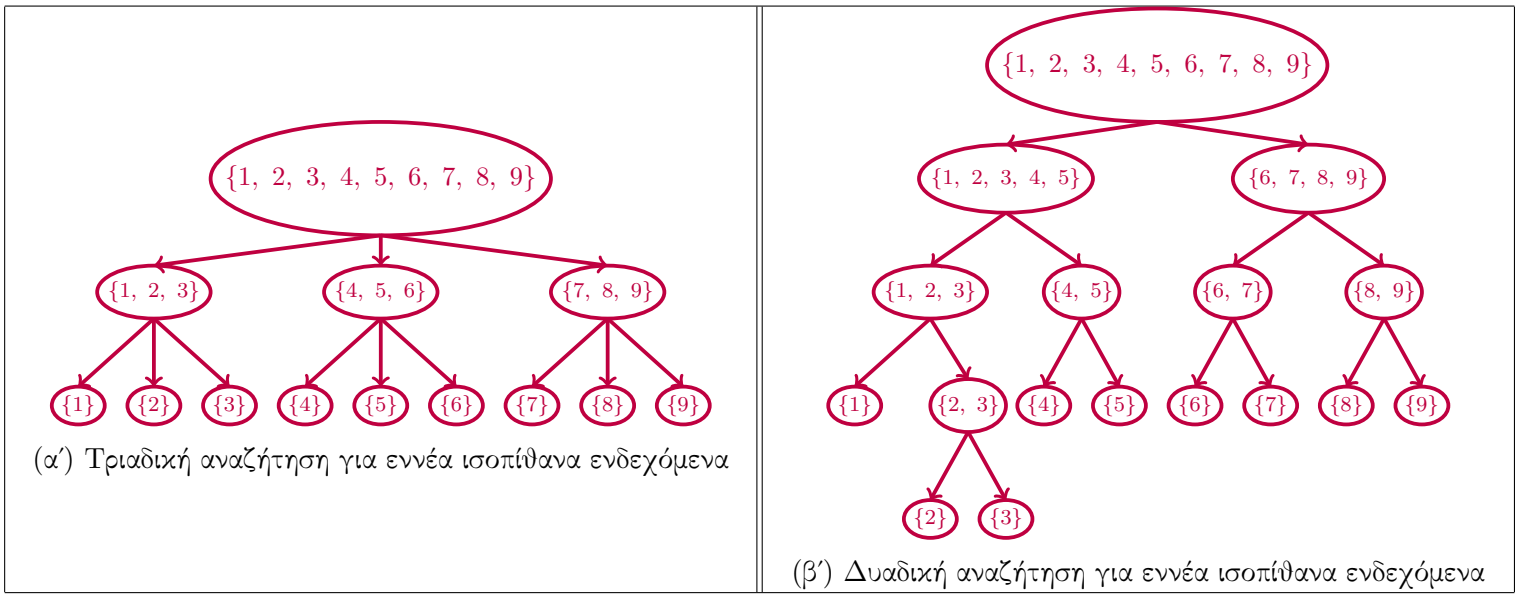
(β′) R.V.L Hartley



(γ′) Claude E. Shannon

Αφού λοιπόν θέσαμε και το ιστορικό πλαίσιο κατά το οποίο έγιναν οι πρώτες απόπειρες να απαντηθούν τα ερωτήματα που θέσαμε και εμείς οι ίδιοι στον εαυτό μας στην προσπάθεια να ποσοτικοποιήσουμε την πληροφορία που περιέχεται σε μία τυχαία μεταβλητή υπό την ερμηνεία της προβλεψιμότητας των τιμών της ήρθε η ώρα να απαντήσουμε σε ένα ακόμη ερώτημα· Και τι γίνεται αν τα ενδεχόμενα δεν είναι ισοπίθανα;

[3]Nyq24.



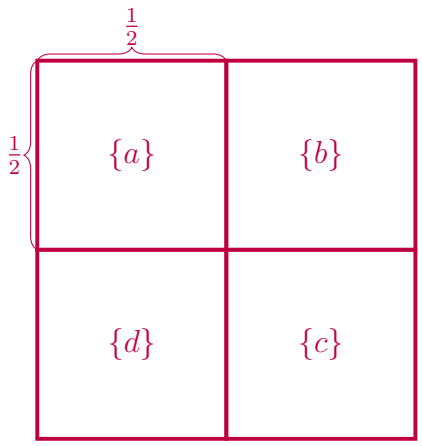
(α′) Τριαδική αναζήτηση για εννέα ισοπίθανα ενδεχόμενα

(β′) Δυαδική αναζήτηση για εννέα ισοπίθανα ενδεχόμενα

Σχήμα 1.3: Η δυαδική και τριαδική αναζήτηση εννέα ισοπίθανων ενδεχομένων

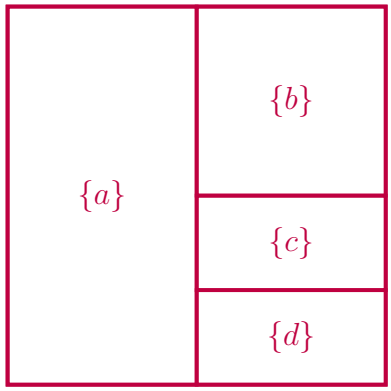
Για να απαντήσουμε στο ερώτημα των μη ισοπίθανων ενδεχομένων θα χρησιμοποιήσουμε την παραπάνω γνώση για τα ισοπίθανα ενδεχόμενα και θα περιγράψουμε ένα αποδοτικό τρόπο για την περίπτωση που μας ενδιαφέρει. Γνωρίζουμε ότι η μεταβλητή Y δεν αποτελείται από ισοπίθανα ενδεχόμενα οπότε μπορούμε να υποψιαστούμε ότι η $P_Y(y) = \{\frac{1}{2}, \frac{1}{8}, \frac{1}{8}, \frac{1}{4}\}$ θα επιβάλει μία πολύ διαφορετική διαμέριση στον χώρο $\mathcal{X} = \{a, b, c, d\}$ από ότι η $P_X(x)$ όπως φαίνεται και στο Σχήμα 1.4(β′) . Προκειμένου να χρησιμοποιήσουμε το αποτέλεσμα που εξάγαμε προηγουμένως πρέπει να μεταφράσουμε το παρών πρόβλημα σε αυτό των ισοπίθανων ενδεχομένων. Για να το κάνουμε αυτό κανονικοποιούμε την $P_Y(y) = \{\frac{1}{2}, \frac{1}{8}, \frac{1}{8}, \frac{1}{4}\} = \{\frac{4}{8}, \frac{1}{8}, \frac{1}{8}, \frac{2}{8}\}$ το οποίο είναι ισοδύναμο με το να χωρίσουμε το $\mathcal{X}$ σε 8 ισεμβαδικά χωρία. Χωρίζοντας τον χώρο $\mathcal{X}$ προκύπτουν τέσσερα υποσύνολα $\{\mathcal{A}_1, \mathcal{A}_2, \mathcal{A}_3, \mathcal{A}_4\}$ με βάση τα σύμβολα $\{a, b, c, d\}$. Το $\mathcal{A}_1$ αποτελείται από 4 ορθογώνια εμβαδού $\frac{1}{8}$ που περιέχουν το "a", το $\mathcal{A}_4$ αποτελείται από 2 ορθογώνια που αντιστοιχούν στο "d" και τα $\mathcal{A}_2, \mathcal{A}_3$ αποτελούνται από 1 ορθογώνιο που αντιστοιχεί στα σύμβολα "b" και "c" αντίστοιχα όπως φαίνεται και στο Σχήμα 1.4 (γ′). Είναι εύκολο να παρατηρήσει κανείς ότι οι πιθανότητες των συμβόλων είναι τα εμβαδά των αντίστοιχων υποσυνόλων. Χρησιμοποιώντας την διαμέριση που επιβάλει η σ.μ.π στον δειγματικό χώρο $\mathcal{X}$ θα ορίσουμε πάλι τον τρόπο που γίνονται οι ερωτήσεις για τα σύμβολα του. Στην προηγούμενη περίπτωση επειδή κάθε σύμβολο αντιστοιχούσε σε ένα τετράγωνο όπως φαίνεται και στο Σχήμα 1.4 (α′), όταν διαμερίζαμε το $\{a, b, c, d\}$ σε δύο ισοπληθικά σύνολα, επί της ουσίας διαμερίζαμε το $\mathcal{X}$ σε δύο ισεμβαδικά κομμάτια.

Αν επιχειρήσουμε να εφαρμόσουμε το ίδιο και για τον χώρο που έχει διαμεριστεί με βάση την $P_Y(y)$, τότε η $Pr[a, b] = \frac{1}{2} + \frac{1}{8} = Area(\mathcal{A}_1) + Area(\mathcal{A}_2) = \frac{5}{8} \neq \frac{1}{2}$. Για το λόγο αυτό αναφέραμε στην αρχή ότι οι διαμερίσεις θα πρέπει να γίνονται μέ βάση τις σ.μ.π των τυχαίων μεταβλητών και όχι τις τιμές τους. Όταν διαμερίσουμε για πρώτη φορά το $\mathcal{X}$, ώστε να προκύψουν δύο ισεμβαδικά χωρία, το σύνολο $\{\mathcal{A}_1, \mathcal{A}_2, \mathcal{A}_3, \mathcal{A}_4\}$ θα διαμεριστεί στα σύνολα $\{\mathcal{A}_1\}$ και $\{\mathcal{A}_2, \mathcal{A}_3, \mathcal{A}_4\}$, οπότε και το σύνολο $\{a, b, c, d\}$ θα διαμεριστεί στα $\{a\}$ και $\{b, c, d\}$. Επειδή κατά την πρώτη διαμέριση προέκυψε το μονοσύνολο $\{a\}$ καταλαβαίνουμε ότι το $\mathcal{A}_1$ δεν χρειάζεται να διαμεριστεί άλλο. Κατά την δεύτερη διαμέριση τα $\{\mathcal{A}_2, \mathcal{A}_3, \mathcal{A}_4\}$ χωρίζονται στα δύο ισεμβαδικά χωρία $\{\mathcal{A}_2, \mathcal{A}_3\}$ και $\{\mathcal{A}_4\}$. Το σύνολο των συμβόλων $\{b, c, d\}$ για το λόγο αυτό διαμερίζεται στα σύνολα $\{b, c\}$ και $\{d\}$. Επειδή το $\mathcal{A}_4$ αντιστοιχεί σε μονοσύνολο δεν χρειάζεται να διαμεριστεί παραπάνω. Κατά την τελευταία διαμέριση το $\{\mathcal{A}_2, \mathcal{A}_3\}$ χωρίζεται στα $\{\mathcal{A}_2\}$ και $\{\mathcal{A}_3\}$, οπότε και το $\{b, c\}$ διαμερίζεται στα $\{b\}$ και $\{c\}$. Επειδή όλα τα χωρία αντιστοιχούν σε μονοσύνολα συμβόλων, οι διαμερίσεις τερματίζονται. Το δυαδικό δένδρο που δημιουργήθηκε φαίνεται στο Σχήμα 1.4.



$\mathcal{X}$

(α′) Η διαμέριση που επιβάλει η σ.μ.π $f_X(x)$ στον χώρο $\mathcal{X} = \{a, b, c, d\}$. Κάθε τετράγωνο έχει εμβαδό $\frac{1}{4}$, όση είναι και η πιθανότητα κάθε συμβόλου



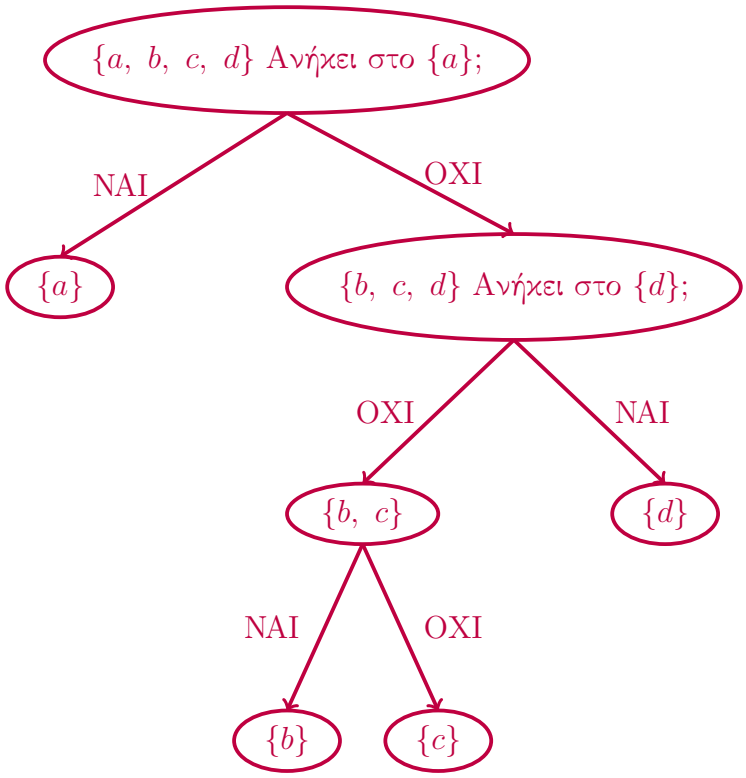
(β′) Η διαμέριση που επιβάλει η σ.μ.π $f_Y(y) = \{\frac{1}{2}, \frac{1}{8}, \frac{1}{8}, \frac{1}{4}\}$ στον χώρο $\mathcal{X} = \{a, b, c, d\}$

.

$\{a\}$	$\{b\}$
$\{a\}$	$\{c\}$
$\{a\}$	$\{d\}$
$\{a\}$	$\{d\}$

(γ′) Η μετάφραση του προβλήματος μη ισοπίθανων ενδεχομένων σε ισοπίθανα ενδεχόμενα. Τα τετράγωνα του ίδιου συμβόλου αναπαριστούν τα υποσύνολα $\{\mathcal{A}_1, \mathcal{A}_2, \mathcal{A}_3, \mathcal{A}_4\}$. Το υποσύνολο $\mathcal{A}_1$ αντιστοιχεί στο σύμβολο "a", το $\mathcal{A}_2$ στο "b", το $\mathcal{A}_3$ στο "c" και το $\mathcal{A}_4$ στο "d". Κάθε ορθογώνιο έχει εμβαδό $\frac{1}{8}$

Σχήμα 1.4: Διαμερίσεις του $\mathcal{X}$ για τις σ.μ.π $f_X(x),\ f_Y(y)$ αντίστοιχα



Σχήμα 1.5: Δυαδική αναζήτηση μη ισοπίθανων ενδεχομένων με βάση την σ.μ.π. $f_Y(y)$

Πρακτικά αυτό που κάναμε είναι να χωρίζουμε το σύνολο του κάθε κόμβου σε δύο ισοπίθανα σύνολα. Το πλήθος των διαμερίσεων που χρειάζεται να πραγματοποιηθούν για να φτάσουμε σε κάποιο μονοσύνολο είναι ίσο με τον αριθμό των ερωτήσεων που απαιτούνται για να μαντέψουμε σωστά ποιο υποσύνολο επιλέχθηκε. Για το γράμμα "a" χρειάζεται μία ερώτηση, για τα "b","c" τρεις και για το "d" χρειάζονται δύο. Επειδή όμως κάθε γράμμα δεν εμφανίζεται με την ίδια πιθανότητα ο μέσο όρος των ερωτήσεων ανά σύμβολο θα είναι ο σταθμισμένος μέσος των ερωτήσεων. Παρακάτω θα αποδείξουμε γιατί ισχύει η απάντηση που μας δίνει ο συγκεκριμένος αλγόριθμος.

Όταν εφαρμόσαμε δυαδική αναζήτηση στην πρώτη περίπτωση απαντήσαμε στο ερώτημα, ποιο είναι το πλήθος των ερωτήσεων κατά μέσο όρο ώστε να βρούμε ποιο τετράγωνο του $\mathcal{X}$ επιλέχθηκε. Θυμίζουμε ότι η αντιστοιχία ανάμεσα στα τετράγωνα και τα σύμβολα του δειγματικού χώρου ήταν αμφιμονοσήμαντη, για αυτό το λόγο η διαμέριση του $\mathcal{X}$ κατά πλήθος ήταν ισοδύναμη με τη διαμέριση του κατά εμβαδόν. Στη δεύτερη περίπτωση το ερώτημα διατυπώνεται ως εξής· Ποιο είναι το πλήθος των ερωτήσεων που χρειάζονται κατά μέσο όρο ώστε να βρούμε ποιο από τα τέσσερα υποσύνολα του $\mathcal{X}$ επιλέχθηκε;.

Το αντίστοιχο πρόβλημα για ισοπίθανα ενδεχόμενα θα ήταν να βρούμε ποιο από τα οχτώ ορθογώνια επιλέχθηκε. Προφανώς αν ξέρουμε το ορθογώνιο γνωρίζουμε και το υποσύνολο στο οποίο ανήκει, αφού τα υποσύνολα αποτελούν διαμέριση του $\mathcal{X}$ οπότε δεν περιέχουν κοινά στοιχεία. Άρα λύνοντας το πρόβλημα για τα ισοπίθανα ενδεχόμενα βρίσκουμε ποιο σύμβολο παράχθηκε. Το παραπάνω πρόβλημα λύνεται χρησιμοποιώντας το αποτέλεσμα για τα ισοπίθανα ενδεχόμενα, οπότε E[#Ερωτήσεων/ορθογώνιο]=$log8 = 3$.

Εμείς όμως θέλουμε να βρούμε τον E[#Ερωτήσεων/σύμβολο]. Προκειμένου να δημιουργήσουμε μία εξίσωση που θα περιέχει την επίμαχη ποσότητα, σπάμε τη διαδικασία της επιλογής του τετραγώνου σε δύο στάδια. Πρώτον προσπαθούμε να βρούμε σε ποιο υποσύνολο ανήκει το τετράγωνο, δηλαδή ποιο υποσύνολο επιλέχθηκε, και κατόπιν να εντοπίσουμε το τετράγωνο. Η εύρεση του υποσυνόλου που ανήκει το τετράγωνο είναι επί της ουσίας το πρόβλημα που προσπαθούμε να λύσουμε. Έτσι καταφέρνουμε να μεταφράσουμε ένα πρόβλημα του οποίου γνωρίζουμε την λύση σε δύο προβλήματα εκ των οποίων ξέρουμε τη λύση του ενός. Επειδή κάθε υποσύνολο του $\mathcal{X}$ αντιστοιχεί σε ένα σύμβολο, το E[#Ερωτήσεων/υποσύνολο] θα είναι ίσο με E[#Ερωτήσεων/σύμβολο]. Ακόμη γνωρίζουμε ότι κάθε τετράγωνο έχει το ίδιο εμβαδόν, άρα τα τετράγωνα που ανήκουν στο ίδιο υποσύνολο ($\mathcal{A}_i$) επιλέγονται με την ίδια πιθανότητα. Οπότε το πρόβλημα της εύρεσης του $E_{\mathcal{A}_i}$[#Ερωτήσεων/τετράγωνο] είναι ένα γνωστό πρόβλημα το οποίο λύνεται σύμφωνα με το αποτέλεσμα για τα ισοπίθανα δεδομένα, δηλαδή $E_{\mathcal{A}_i}$[#Ερωτήσεων/τετράγωνο]=$log(|\mathcal{A}_i|)$. Η εξίσωση που περιγράψαμε διαμορφώνεται ως εξής:

$$\boxed{E[\#\text{Ερωτήσεων/Τετράγωνο}] = E[\#\text{Ερωτήσεων/Σύμβολο}] + \sum_{i=1}^{4} Pr[\mathcal{A}_i] \cdot E_{\mathcal{A}_i}[\#\text{Ερωτήσεων/Τετράγωνο}]}$$

Ο τελευταίος όρος της εξίσωσης αποτελεί τον σταθμισμένο μέσο των ερωτήσεων ανά υποσύνολο που χρειάζονται για να συγκεκριμενοποιήσουμε σε ποιο τετράγωνο του υποσυνόλου που επιλέχθηκε αναφερόμαστε, καθώς η πιθανότητα να βρεθούμε σε ένα από τα τέσσερα υποσύνολα δεν είναι η ίδια. Επίσης εκφράζει τις επιπλέον κατά μέσο όρο ερωτήσεις/σύμβολο που θα χρειαζόμασταν, αν τα ενδεχόμενα ήταν ισοπίθανα. Η ερμηνεία αυτή φαίνεται καλύτερα αν γράψουμε την παραπάνω εξίσωση ως εξής:

$$E[\#\text{Ερωτήσεων/τετράγωνο}] - E[\#\text{Ερωτήσεων/σύμβολο}] = \sum_{i=1}^{4} Pr[\mathcal{A}_i] \cdot E_{\mathcal{A}_i}[\#\text{Ερωτήσεων/τετράγωνο}].$$

Αντικαθιστούμε τα δεδομένα στην εξίσωση και έχουμε:

$$E[\#\text{Ερωτήσεων/τετράγωνο}] = E[\#\text{Ερωτήσεων/σύμβολο}] + \sum_{i=1}^{4} Pr[\mathcal{A}_i] \cdot E_{\mathcal{A}_i}[\#\text{Ερωτήσεων/τετράγωνο}] \Rightarrow$$

$$E[\#\text{Ερωτήσεων/τετράγωνο}] = E[\#\text{Ερωτήσεων/σύμβολο}] + \sum_{i=1}^{4} Pr[\mathcal{A}_i] \cdot \log_2(|\mathcal{A}_i|) \Rightarrow$$

$$3 = E[\#\text{Ερωτήσεων/σύμβολο}] + \sum_{i=1}^{4} Pr[\mathcal{A}_i] \cdot \log_2(|\mathcal{A}_i|) \Rightarrow$$

$$3 = E[\#\text{Ερωτήσεων/σύμβολο}] + (\frac{1}{2}log4 + \frac{1}{4}log2 + \frac{1}{8}log1 + \frac{1}{8}log1) \Rightarrow$$

$$3 = E[\#\text{Ερωτήσεων/σύμβολο}] + 1.25 \Rightarrow$$

$$E[\#\text{Ερωτήσεων/σύμβολο}] = 1.75$$

Η ποσότητα 1.25 εκφράζει τις παραπάνω ερωτήσεις ανά υποσύνολο κατά μέσο όρο που θα έπρεπε να κάνουμε, ώστε να βρούμε πέρα από το υποσύνολο που επιλέχθηκε και το τετράγωνο του υποσυνόλου.

Ανακεφαλαιώνοντας, προκειμένου να βρούμε το μέσο αριθμό των ερωτήσεων/σύμβολο για n μη ισοπίθανα ενδεχόμενα, διαμερίζουμε τον χώρο $\mathcal{X}$ ανάλογα με την σ.μ.π, ώστε να δημιουργηθούνε N υποσύνολα $(\{\mathcal{A}_i\}_{i=1}^{N})$ που απαρτίζονται από ισοπίθανα ενδεχόμενα. Έτσι κάθε πιθανότητα γράφεται:

$$Pr[\mathcal{A}_i] = Pr[X = x] = \frac{k_i}{\sum_{m=1}^{n} k_m}, \text{ όπου } k_i = |\mathcal{A}_i| \text{ και } \sum_{m=1}^{n} k_m = |\mathcal{X}|$$

.

Δηλαδή κάθε πιθανότητα εκφράζει τον αριθμό των τετραγώνων που απαρτίζουν το υποσύνολο προς τα συνολικό αριθμό των τετραγώνων που συνθέτουν το $\mathcal{X}$. Τότε η εξίσωση:

$$E[\#\text{Ερωτήσεων/τετράγωνο}] = E[\#\text{Ερωτήσεων/σύμβολο}] + \sum_{i=1}^{N} Pr[\mathcal{A}_i] \cdot E_{\mathcal{A}_i}[\#\text{Ερωτήσεων/τετράγωνο}]$$

γίνεται:

$$log\sum_{m=1}^{N} k_m = E[\#\text{Ερωτήσεων/σύμβολο}] + \sum_{i=1}^{N} Pr[\mathcal{A}_i]\cdot E_{\mathcal{A}_i}[\#\text{Ερωτήσεων/τετράγωνο}] \Rightarrow$$

$$log\sum_{m=1}^{N} k_m = E[\#\text{Ερωτήσεων/σύμβολο}] + \sum_{i=1}^{N} Pr[\mathcal{A}_i]\cdot \log|\mathcal{A}_i| \Rightarrow$$

$$log\sum_{m=1}^{N} k_m = E[\#\text{Ερωτήσεων/σύμβολο}] + \sum_{i=1}^{N} Pr[\mathcal{A}_i]\cdot \log(k_i) \Rightarrow$$

$$E[\#\text{Ερωτήσεων/σύμβολο}] = log\sum_{m=1}^{N} k_m - \sum_{i=1}^{N} Pr[\mathcal{A}_i]\cdot \log(k_i) \Rightarrow$$

$$E[\#\text{Ερωτήσεων/σύμβολο}] = 1\cdot log\sum_{m=1}^{N} k_m - \sum_{i=1}^{N} Pr[\mathcal{A}_i]\cdot \log(k_i) \Rightarrow$$

$$E[\#\text{Ερωτήσεων/σύμβολο}] = \sum_{i=1}^{N} Pr[\mathcal{A}_i]\cdot \log\sum_{m=1}^{N} k_m - \sum_{i=1}^{N} Pr[\mathcal{A}_i]\cdot \log(k_i) \Rightarrow$$

$$E[\#\text{Ερωτήσεων/σύμβολο}] = \sum_{i=1}^{N} Pr[\mathcal{A}_i]\cdot \log\frac{\sum_{m=1}^{N} k_m}{k_i} \Rightarrow$$

$$E[\#\text{Ερωτήσεων/σύμβολο}] = \sum_{i=1}^{N} Pr[\mathcal{A}_i]\cdot \log\frac{1}{Pr[|\mathcal{A}_i|]}$$

Άρα

$$\boxed{E[\#\text{Ερωτήσεων/σύμβολο}] = \sum_{i=1}^{N} Pr[\mathcal{A}_i]\cdot \log\frac{1}{Pr[|\mathcal{A}_i|]} = \sum_{x\in\mathcal{X}} Pr[X=x]\cdot \log\frac{1}{Pr[X=x]}}$$

Η μαθηματική έκφραση της παραπάνω σχέσης είναι ο τρόπος με τον οποίο κατασκευάσαμε το δυαδικό δένδρο της εικόνας 2. Το μονοπάτι που οδηγεί στο "a", έχει μήκος 1 και εκφράζει το πλήθος των ερωτήσεων που χρειάζονται προκειμένου να βρεθούμε σε κάποιο από τα τετράγωνα που ανήκουν στο $\mathcal{A}_1$.Το μονοπάτι που οδηγεί στο "d" έχει μήκος 2 ενώ τα μονοπάτια των "c","d" έχουν μήκος 3, τα οποία με τη σειρά τους εκφράζουν το πλήθος των ερωτήσεων που χρειάζονται κατά τη διάρκεια της δυαδικής αναζήτησης ώστε να βρεθούμε σε κάποιο από τα τετράγωνα που ανήκουν στα $\mathcal{A}_2, \mathcal{A}_3, \mathcal{A}_4$ αντίστοιχα. Η πιθανότητα να διανύσουμε το κάθε μονοπάτι είναι ίση με τη πιθανότητα του κάθε συμβόλου. Άρα:

$$\text{E}[\#\text{Ερωτήσεων/σύμβολο}]=\tfrac{1}{2}log2 + \tfrac{1}{8}log8 + \tfrac{1}{8}log8 + \tfrac{1}{4}log4 = 1.75$$

Η παραπάνω σχέση λειτουργεί και στην περίπτωση που τα ενδεχόμενα είναι ισοπίθανα. Χρησιμοποιώντας την $P_X(x)$, βρίσκουμε:

$$\text{E}[\#\text{Ερωτήσεων/σύμβολο}]=\tfrac{1}{4}log4 + \tfrac{1}{4}log4 + \tfrac{1}{4}log4 + \tfrac{1}{4}log4 + \tfrac{1}{4}log4 = 2$$

Τα συμπεράσματα μας συμφωνούν και με την διαίσθηση που είχαμε πως η συμβολοσειρά που παράγεται από την πρώτη στοχαστική διαδικασία $\{X_n\}_{n\in\mathbb{N}}$ είναι περισσότερο απρόβλεπτη από αυτή που παράγεται με τη δεύτερη $\{Y_n\}_{n\in\mathbb{N}}$, η οποία δείχνει προτίμηση στο "a". Οι διαδικασίες που περιγράφτηκαν και η σχέση που παράχθηκε δεν ισχύει μόνο για τις συγκεκριμένες μεταβλητές X, Y αλλά για κάθε διακριτή τυχαία μεταβλητή που παίρνει τιμές σε ένα πεπερασμένο σύνολο. Η ποσότητα $\sum_{x\in\mathcal{X}} Pr[X=x]log\frac{1}{Pr[X=x]}$, δεν είναι τυχαία, ονομάζεται εντροπία της πληροφορίας και αναφέρθηκε από τον Claude E. Shannon, το 1948 στην εμβληματική

δημοσίευση του A mathmetical theory of communication[4]. Σε αυτή την εργασία ο Shannon κατάφερε μέσα από τρεις αξιωματικές προϋποθέσεις που θεώρησε ότι πρέπει να πληρεί ένα μέτρο πληροφορίας να εξάγει τον τύπο που περιγράφει την ελάχιστη μέση πληροφορία που περιέχεται σε μία τυχαία μεταβλητή. Οι τρεις αξιωματικές προϋποθέσεις που έπρεπε να πληρεί το $H(P)$, όπου $P_X(x) = \{Pr[X = x_1], \cdots, Pr[X = x_n]\}$ ήταν:

1. Το μέτρο να είναι συνεχές ως προς τις πιθανότητες της σ.μ.π $P_X(x)$

2. Αν οι πιθανότητες της σ.μ.π είναι όλες ίσες με $\frac{1}{n}$ θα πρέπει το μέτρο να είναι μία μονοτονικά αύξουσα συνάρτηση του n.

3. Αν μία επιλογή μπορεί να σπάσει σε δύο στάδια τότε η πληροφορία της αρχικής επιλογής πρέπει να ισούται με τον σταθμισμένο άθροισμα των πληροφοριών που περιέχονται στις επιλογές των δύο σταδίων.

Ο τύπος που εξήγαγε ο Shannon και εξηγήσαμε με όλη την παραπάνω ανάλυση δίνεται από τον παρακάτω ορισμό.

Ορισμός 1.1. *Έστω μία διακριτή τυχαία μεταβλητή X που παίρνει τιμές στο πεπερασμένο σύνολο $\mathcal{X}$ με συνάρτηση μάζας πιθανότητας $P_X(x) = Pr[X = x]$, $\forall x \in \mathcal{X}$. Η* ***εντροπία*** *της τυχαίας μεταβλητής X ορίζεται ως*

$$\boxed{H(X) = \sum_{x \in \mathcal{X}} Pr[X = x] \cdot \log \frac{1}{Pr[X = x]}} \tag{1.1}$$

Στο σημείο αυτό να δώσουμε και ένα ιστορικό γεγονός που σχετίζεται με την ονομασία της σχέσης (1.1). Όταν ο Shannon βρήκε τον παραπάνω τύπο σκεφτόταν πως θα τον ονομάσει. Τα ονόματα που είχε σκεφτεί ήταν πληροφορία η αβεβαιότητα. Όταν συναντήθηκε με τον John von Neumann και ρώτησε τη γνώμη του εκείνος αποκρίθηκε πώς ένα κατάλληλο όνομα θα ήταν "έντροπία" για δύο λόγους. Πρώτον γιατί η παραπάνω σχέση ήδη υπήρχε στον κλάδο της στατιστικής μηχανικής και δεύτερον γιατί κανείς δεν ξέρει τι στα αλήθεια είναι η εντροπία οπότε σε μία αντιπαράθεση θα έχει πλεονέκτημα. [5]

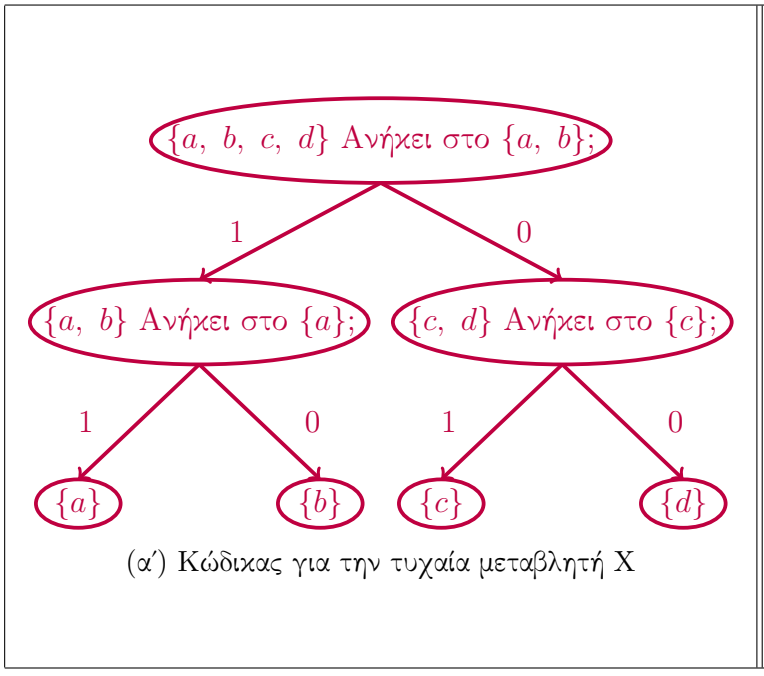
Παράδειγμα 1.1. *Στην προηγούμενη ανάλυση υποσχεθήκαμε να ξεκαθαρίσουμε την τεχνική λεπτομέρεια που αφορά στη βάση του λογαρίθμου. Ήρθε η ώρα τα τηρήσουμε την υπόσχεση μας! Με βάση τις δύο σ.μ.π $P_X(x), P_Y(y)$, κατασκευάσαμε δύο δυαδικά δένδρα, στα κλαδιά των οποίων υπήρχαν οι απαντήσεις ΝΑΙ και ΟΧΙ. Αν αντικαταστήσουμε τη συνθήκη ΝΑΙ/ΟΧΙ σε κάθε κλάδο με τα ψηφία 1→ ΝΑΙ και 0 → ΟΧΙ , τότε μπορούμε να απεικονίσουμε κάθε σύμβολο σε μία κωδική λέξη τα σύμβολα της οποίας ανήκουν στο αλφάβητο {0,1}. Οι κώδικες που προκύπτουν για τις δύο μεταβλητές φαίνονται στον Πίνακα 1.1 και τα δένδρα κωδικοποίησης στο Σχήμα 1.6:*

[4] Sha48.

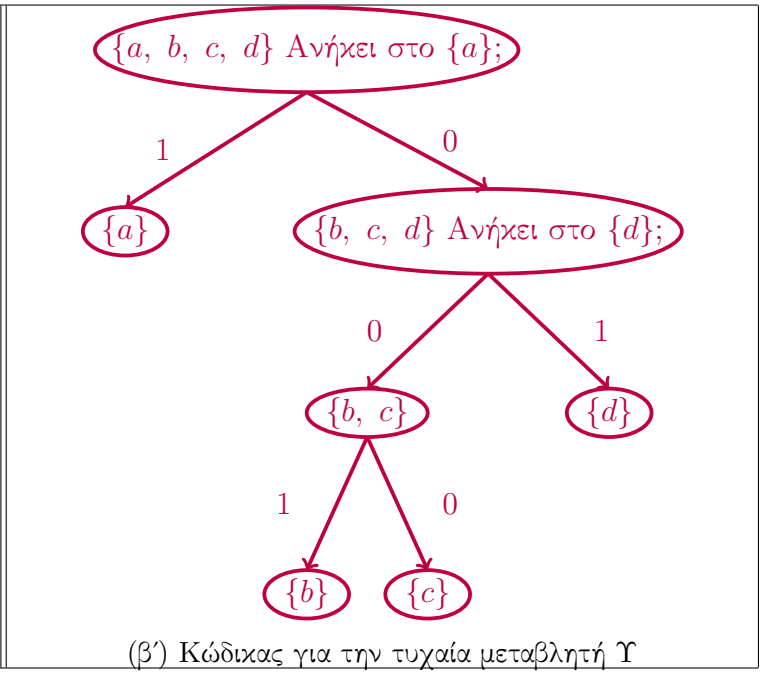
[5] Οι πηγές υπάρχουν για την συγκεκριμένη ιστορία στον σύνδεσμό `https://en.wikipedia.org/wiki/Talk%3AHistory_of_entropy`. Εμείς απλά εδώ δώσαμε μία ελεύθερη μετάφραση του διαλόγου.

Σύμβολο	Κώδικας για την X	Κώδικας για την Υ
a	11	1
b	10	011
c	01	010
d	00	00

Πίνακας 1.1: Κώδικες για τις μεταβλητές X και Υ



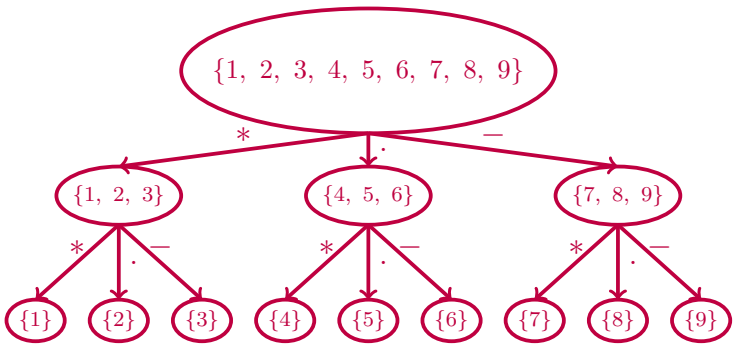
(α′) Κώδικας για την τυχαία μεταβλητή X



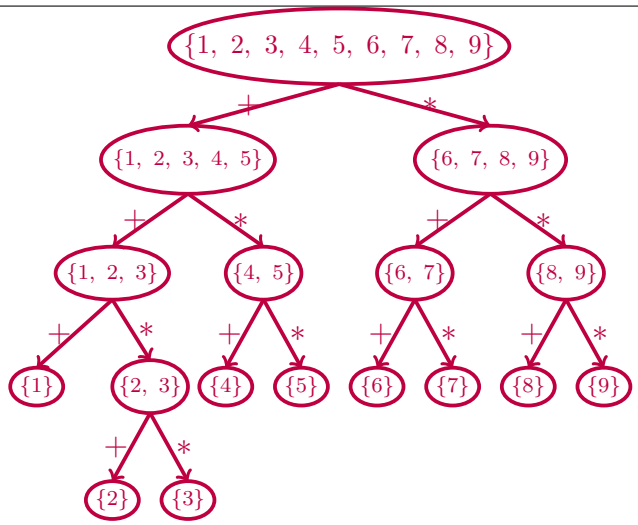
(β′) Κώδικας για την τυχαία μεταβλητή Υ

Σχήμα 1.6: Κώδικες για τις τυχαίες μεταβλητές X και Υ.

Κατά τον ίδιο τρόπο μπορούμε να κωδικοποιήσουμε και τους αριθμούς {1,2,3,4,5,6,7,8,9}, Χρησιμοποιώντας τα δένδρα που υπάρχουν στο Σχήμα 1.2. Για το πρώτο δένδρο αντιστοιχούμε το αστεράκι (∗) στον αριστερό κλάδο κάθε κόμβου, την παύλα (−) στον μεσαίο και την τελεία (.) στο δεξιό κλάδο. Για το δεύτερο δένδρο θα χρησιμοποιήσουμε το σύμβολο + για τον αριστερό κλάδο και το ∗ για τον δεξιό. Οι κώδικες φαίνονται στον Πίνακα 1.2 και τα δένδρα κωδικοποίησης στο Σχήμα 1.6



(α′) Κώδικας για εννέα ισοπίθανα ενδεχόμενα χρησιμοποιώντας τριαδική αναζήτηση



(β′) Κώδικας για εννέα ισοπίθανα ενδεχόμενα χρησιμοποιώντας δυαδική αναζήτηση

Σχήμα 1.7: Η δυαδική και τριαδική αναζήτηση εννέα ισοπίθανων ενδεχομένων

Σύμβολο	Κώδικας για το δένδρο 1.7(α′)	Κώδικας για το δένδρο 1.7(β′)
1	**	++++
2	*-	+++*
3	*.	++*
4	-*	+*+
5	–	+**
6	-.	*++
7	.*	*+*
8	.-	**+
9	..	***

Πίνακας 1.2: Κώδικες για τις μεταβλητές X και Υ

Από τις παραπάνω διαδικασίες καταλαβαίνουμε ότι η βάση του λογαρίθμου αποτελεί τον πληθάριθμο του κωδικού αλφαβήτου στο οποίο "μεταφράζουμε" τον χώρο $\mathcal{X}$ της εκάστοτε μεταβλητής. Στους κώδικες για τις μεταβλητές X, Y μεταφράσαμε τα σύμβολα $\{a, b, c, d\}$ στο αλφάβητο {0,1}, δηλαδή τη γλώσσα των υπολογιστών. Το κωδικό αλφάβητο που θα χρησιμοποιούμε κάθε φορά εξαρτάται από την εκάστοτε εφαρμογή.

Ένα ακόμη σημείο που αξίζει να παρατηρήσουμε, αφορά στο μήκος τον κωδικών λέξεων. Στον Πίνακα 1.1 βλέπουμε ότι οι κωδικές λέξεις που αναφέρονται στο ίδιο γράμμα έχουν διαφορετικό μήκος. Αν θέλαμε να βρούμε το μέσο μήκος κάθε κώδικα θα μπορούσαμε να υπολογίσουμε τον σταθμισμένο μέσο από τα μήκη των κωδικών λέξεων για τα αντίστοιχα σύμβολα. Έτσι έχουμε:

$$\textit{Μέσο μήκος για τον κώδικα της } X = \frac{1}{4}\cdot 2 + \frac{1}{4}\cdot 2 + \frac{1}{4}\cdot 2 + \frac{1}{4}\cdot 2 = 2$$

$$\textit{Μέσο μήκος για τον κώδικα της } Y = \frac{1}{2}\cdot 1 + \frac{1}{8}\cdot 3 + \frac{1}{8}\cdot 3 + \frac{1}{4}\cdot 2 = 1.75$$

$$\textit{Μέσο μήκος για τον κώδικα του δένδρου 1.7(α')} =$$
$$\frac{1}{9}\cdot 2 + \frac{1}{9}\cdot 2 + \frac{1}{9}\cdot 2 + \frac{1}{9}\cdot 2 + \frac{1}{9}\cdot 2 + \frac{1}{9}\cdot 2 + \frac{1}{9}\cdot 2 + \frac{1}{9}\cdot 2 + \frac{1}{9}\cdot 2 = 2$$

$$\textit{Μέσο μήκος για τον κώδικα του δένδρου 1.7(β')} =$$
$$\frac{1}{9}\cdot 4 + \frac{1}{9}\cdot 4 + \frac{1}{9}\cdot 3 + \frac{1}{9}\cdot 3 + \frac{1}{9}\cdot 3 + \frac{1}{9}\cdot 3 + \frac{1}{9}\cdot 3 + \frac{1}{9}\cdot 3 + \frac{1}{9}\cdot 3 = 3.2$$

Παρατηρούμε ότι οι αριθμοί που βρήκαμε είναι οι εντροπίες των τυχαίων μεταβλητών X και Y. Μια ακόμη ερμηνεία λοιπόν της εντροπίας είναι ότι αποτελεί το μέσο ελάχιστο "κόστος" ανά σύμβολο που απαιτείτε ώστε να εκφράσουμε τις τιμές της με βάση κάποιο άλλο αλφάβητο. Η τυχαία μεταβλητή X που είναι εντελώς απρόβλεπτη απαιτεί για κάθε γράμμα της κόστος ίσο με το πληθάριθμο του κωδικού αλφαβήτου. Η μεταβλητή Y από την άλλη που είναι περισσότερο προβλέψιμη κοστίζει λιγότερο ως προς την μετάφραση της, μόλις 1.75 σύμβολα/γράμμα κατά μέσο ορό. Αυτό συμβαίνει διότι τα πιο πιθανά γράμματα της, τα έχουμε κωδικοποιήσει με λέξεις μικρού μήκους που είναι περισσότερες " οικονομικές ". Περισσότερα πάνω σε αυτή την ιδέα θα αναφερθούν στο Κεφάλαιο 3.

Τελικά τι είναι η εντροπία μια τυχαίας μεταβλητής; Γενικά βλέπουμε πως όταν μιλάμε για την εντροπία έρχονται στο προσκήνιο πάντα οι λέξεις "άποδοτικό" και "ἐλάχιστο". Το φαινόμενο αυτό δεν είναι τυχαίο, η αποδοτικότητα δεν μας αφήνει περιθώρια για περιττές κινήσεις, όπως και το "ἐλάχιστο". Όταν αναζητούσαμε στην αρχή του κεφαλαίου τον "ἐλάχιστο" αριθμό ερωτήσεων και μάλιστα οι ερωτήσεις έπρεπε να έχουν διατυπωθεί με "άποδοτικό" τρόπο, ήταν σαν να μας έλεγαν ότι έχουμε δικαίωμα να ρωτήσουμε μόνο τα βασικά για τις τυχαίες μεταβλητές ώστε να καταλάβουμε τη συμπεριφορά τους. Όταν στο παράδειγμα 1 προχωρήσαμε στην κωδικοποίηση τους, η εντροπία αποτελούσε το "ἐλάχιστο" κόστος κατά την μετάφραση τους. Πάλι ήταν σαν να μας ανάγκαζαν να χρησιμοποιήσουμε με όση περισσότερη σύνεση γίνεται τα γράμματα του κωδικού αλφαβήτου προκειμένου να περιγράψουμε τις τιμές της τυχαίας μεταβλητής.

Μία αντίστοιχη περίπτωση που είναι περισσότερο κοντά στην ανθρώπινη διαίσθηση είναι η εξής: Παρατηρήστε τη φωτογραφία με τα βάζα. Πείτε σε κάποιον γύρο σας να επιλέξει ένα από αυτά. Έχετε το δικαίωμα να κάνετε τρεις ερωτήσεις ώστε να καταλάβετε ποιο βάζο διάλεξε ο συμπαίκτης σας. Οι απαντήσεις που θα λάβετε θα είναι τύπου ΝΑΙ και ΟΧΙ.

Αλήθεια ποια είναι η πρώτη ερώτηση που σας έρχεται στο μυαλό; Σίγουρα δεν θα ρωτούσατε για το χρώμα του βάζου διότι όλα έχουν το ίδιο χρώμα όποτε αυτό δεν θα σας έδινε κάποια σημαντική πληροφορία. Μία καλή ερώτηση θα ήταν αν το βάζο περιέχει λουλούδια, αν είναι πεπλατυσμένο ή ψηλόλιγνο. Καταλαβαίνουμε λοιπόν ότι οι ερωτήσεις μας στρέφονται γύρω από τα βασικά χαρακτηριστικά που διαχωρίζουν τα αντικείμενα μεταξύ τους. Κάθε απάντηση μας δίνει ακόμη μία πληροφορία ώστε να αποκλείσουμε κάποια βάζα.

Συμπερασματικά μπορούμε να πούμε πως για να βρούμε το βάζο που επιλέχθηκε αρκεί να έχουμε στα χέρια μας τον ελάχιστο αριθμό των πληροφοριών που το περιγράφουν κατά μοναδικό τρόπο, δηλαδή τις ουσιώδης πληροφορίες που το συνθέτουν. Μόλις καταφέραμε να ποσοτικοποιήσουμε την πληροφορία που κατέχει ένα αντικείμενο. Είναι ο ελάχιστος αριθμός ερωτήσεων που χρειαζόμαστε για να το ξεχωρίζουμε από τα υπόλοιπα. Επίσης καταφέραμε να απαντήσουμε στο ερώτημα, τι είναι εν τέλει η εντροπία.

Η εντροπία είναι η "ὀυσία" μιας τυχαίας μεταβλητής, είναι ο αριθμός που ποσοτικοποιεί την πληροφορία που "κουβαλάει" μαζί της. Προσοχή όμως! Η πληροφορία αυτή δεν κρύβεται στις τιμές της. Οι τιμές είναι απλά τα οχήματα που την μεταφέρουν. Η πληροφορία μιας τυχαίας μεταβλητής κρύβεται μέσα στη σ.μ.π. Αυτή είναι που χαρακτηρίζει τη συμπεριφορά της, όπως συνέβη με τη σ.μ.π $f_Y(y)$, που μας "μαρτύρησε" ότι έχει μία

προτίμηση στο "ά". Το γεγονός αυτός φαίνεται και από τη σχέση της εντροπίας που αποτελεί επί της ουσίας μια συνάρτηση των πιθανοτήτων της σ.μ.π $H(P)$ της τυχαίας μεταβλητής και όχι των τιμών της.

Συνοπτικά τα συμπεράσματα που μπορούμε να εξάγουμε για την εντροπία είναι τα εξής:

1. Αποτελεί το μέτρο της κατά μέσο όρο πληροφορίας που εμπεριέχεται σε μία τυχαία μεταβλητή.

2. Ο όρος $\log \frac{1}{Pr[X=x]}$ αποτελεί το μέτρο πληροφορίας που περιλαμβάνεται στην πραγματοποίηση του γεγονότος
$$\{\omega \in \Omega \mid X(\omega) = x\}$$
Όπως είδαμε στα δυαδικά δένδρα, είναι το μήκος του μονοπατιού από τη ρίζα προς το ενδεχόμενο$\{\omega \in \Omega \mid X(\omega) = x\}$, αρά είναι η πληροφορία που εμπεριέχεται στην εκάστοτε τιμή της τυχαίας μεταβλητής. Η συγκεκριμένη ποσότητα λέγεται **ιδιοπληροφορία**. Άρα με βάση αυτό τον ορισμό μπορούμε να που ότι η εντροπία μίας πηγής είναι ο σταθμισμένος μέσος των ιδιοπληροφοριών των τιμών της.

3. Είναι το ελάχιστο ποσό πληροφορίας που χρειαζόμαστε κατά μέσο όρο ώστε να περιγράψουμε την τυχαία μεταβλητή X

4. Η εντροπία μίας τυχαίας μεταβλητής δεν εξαρτάται από τη βάση του λογαρίθμου. Η βάση αποτελεί απλά τη μονάδα μέτρησης στην οποία έχουμε αποφασίσει να ποσοτικοποιήσουμε την πληροφορία. Αν η βάση είναι 2 τότε μετράμε την πληροφορία σε $bits$. Αν η βάση είναι ο φυσικός λογάριθμος e τότε η μονάδα μέτρησης είναι το $1nat = 1.44bits$. Το γεγονός αυτό επιβεβαιώνεται και από τον μαθηματικό τύπο της αλλαγής βάση λογαρίθμου.
$$H_a(P) = \sum_{i=1}^{n} Pr[X=x_i] log_a \frac{1}{Pr[X=x_i]} = \sum_{i=1}^{n} Pr[X=x_i] \frac{log_b \frac{1}{Pr[X=x_i]}}{log_b(a)} =$$
$$\frac{1}{log_b(a)} \sum_{i=1}^{n} Pr[X=x_i] log_b \frac{1}{Pr[X=x_i]} = \frac{1}{log_b(a)} H_b(P).$$

Θα επιλέξουμε να κλείσουμε αυτή την παράγραφο παραθέτοντας και τον τρόπο που κατάφερε ο Shannon βασιζόμενος στις αξιωματικές προϋποθέσεις που διατυπώσαμε παραπάνω να αποδείξει τον τύπο της εντροπίας. Η απόδειξη αυτή για όποιον ενδιαφέρεται βρίσκεται στο παράρτημα 2 της εργασίας του[6].

Η αξιωματική θεμελίωση της εντροπίας

Υπενθυμίζουμε τις τρεις αξιωματικές υποθέσεις που θεώρησε ο Shannon ότι πρέπει να τηρεί ένα μέτρο πληροφορίας. Έστω μία τυχαία μεταβλητή X που παίρνει τιμές στο πεπερασμένο σύνολο $\mathcal{X}$ και ακολουθεί την σ.μ.π $P_X(x) = \{Pr[X=x_1], \cdots, Pr[X=x_n]\}$. Τότε το μέτρο $H(P)$ πρέπει να τηρεί της εξής προϋποθέσεις:

1. Το μέτρο να είναι συνεχές ως προς τις πιθανότητες της σ.μ.π $P_X(x)$

2. Αν οι πιθανότητες της σ.μ.π είναι όλες ίσες με $\frac{1}{n}$ θα πρέπει το μέτρο να είναι μία μονοτονικά αύξουσα συνάρτηση του n.

3. Αν μία επιλογή μπορεί να σπάσει σε δύο στάδια τότε η πληροφορία της αρχικής επιλογής πρέπει να ισούται με τον σταθμισμένο άθροισμα των πληροφοριών που περιέχονται στις επιλογές των δύο σταδίων.

[6] (Sha48), Appendix 2

Από την προϋπόθεση δύο ισχύει ότι $H(\frac{1}{n},\frac{1}{n},\cdots,\frac{1}{n}) = A(n)$. Από την προϋπόθεση (3) ξέρουμε πως εάν έχουμε να επιλέξουμε ανάμεσα σε s^m ισοπίθανα ενδεχόμενα, τότε αυτή η επιλογή μπορεί να σπάσει σε m στάδια όπου στο κάθε στάδιο θα έχουμε να επιλέξουμε ανάμεσα σε s ισοπίθανα ενδεχόμενα.[7] Τότε θα ισχύει:

$$A(s^m) = mA(s)$$

και γενικά για οποιοδήποτε $t,\ n$ θα ισχύει $A(t^n) = nA(t)$. Τότε μπορούμε να διαλέξουμε ένα n μεγάλο ώστε να ισχύει:

$$s^m \leq t^m < s^{(m+1)} \Rightarrow \log s^m \leq \log t^n < \log s^{(m+1)} \Rightarrow m\log s \leq n\log t < (m+1)\log s \overset{\div n\log s}{\Rightarrow}$$
$$\frac{m}{n} \leq \frac{\log t}{\log s} < \frac{m}{n} + \frac{1}{n} \Rightarrow \left|\frac{\log t}{\log s} - \frac{m}{n}\right| < \frac{1}{n} \overset{\text{Θέτουμε } \epsilon=\frac{1}{n}}{\Rightarrow} \left|\frac{\log t}{\log s} - \frac{m}{n}\right| < \epsilon$$

Αντίστοιχα από την μονοτονία του μέτρου πληροφορίας για ισοπίθανα ενδεχόμενα γνωρίζουμε ότι:

$$A(s^m) \leq A(t^n) \leq A(s^{(m+1)}) \Rightarrow mA(s) \leq nA(t) \leq (m+1)A(s) \overset{\div nA(s)}{\Rightarrow}$$
$$\frac{m}{n} \leq \frac{A(t)}{A(s)}\frac{m}{n} + \frac{1}{n} \Rightarrow \left|\frac{A(t)}{A(s)} - \frac{m}{n}\right| \leq \frac{1}{n} \overset{\text{Θέτουμε } \epsilon=\frac{1}{n}}{\Rightarrow} \left|\frac{A(t)}{A(s)} - \frac{m}{n}\right| \leq \epsilon$$

Οπότε εν τέλει έχουμε ότι $\left|\frac{A(t)}{A(s)} - \frac{\log t}{\log s}\right| < 2\epsilon \Rightarrow A(t) = K\log(t)$, όπου το K μία σταθερά που πρέπει να ικανοποιεί την προϋπόθεση (2).

Έστω ότι οι πιθανότητες της κατανομής P έχουν την μορφή $Pr[X = x_i] = \frac{n_i}{\sum ni}$ όπου τα n_i είναι ακέραιοι οπότε καταλαβαίνουμε ότι η κατανομή αποτελείται από ρητούς αριθμούς. Τότε μπορούμε να σπάσουμε μία επιλογή από $\sum n_i$ ενδεχόμενα σε δύο στάδια. Πρώτα επιλέγουμε ανάμεσα σε n ενδεχόμενα με πιθανότητες $Pr[X = X_1], \cdots Pr[X = x_n]$ το καθένα και αν επιλεγεί το $i-$οστό επιλέγουμε ανάμεσα σε n_i ίσες πιθανότητες. Από την προϋπόθεση (3) θα έχουμε:

$$K\log\sum n_i = H(P) + K\sum Pr[X = x_i]\log n_i \Rightarrow H(P) = K\log\sum n_i - +K\sum Pr[X = x_i]\log n_i \Rightarrow$$
$$H(P) = -K\sum Pr[X = x_i]\log\frac{n_i}{\sum n_i} \Rightarrow H(P) = -K\sum Pr[X = x_i]\log Pr[X = x_i]$$

Αν οι πιθανότητες είναι άρρητοι μπορούν να προσεγγιστούν από ρητούς και έτσι από την προϋπόθεση (1) συμπεραίνουμε ότι η συγκεκριμένη σχέση ισχύει γενικά για όλες τις πιθανότητες. Η σταθερά K εισάγεται για ευκολία και σχετίζεται την μονάδα μέτρησης που θα επιλέξουμε να ποσοτικοποιήσουμε την πληροφορία.

1.2 Από κοινού εντροπία

Στην παραπάνω ενότητα ορίσαμε τι σημαίνει εντροπία μια τυχαίας μεταβλητής. Τι συμβαίνει όμως όταν η κατάσταση ενός υπό μελέτη φαινομένου επηρεάζεται από περισσότερες της μίας τυχαίες μεταβλητές; Τότε θα θέλαμε να ποσοτικοποιήσουμε την πληροφορία που παίρνουμε από το σύνολο των ιδιοτήτων που χαρακτηρίζουν το φαινόμενο, δηλαδή θα επιθυμούσαμε να βρούμε την εντροπία που δημιουργούν από κοινού οι παραπάνω τυχαίες μεταβλητές.

Προκειμένου να γίνει περαιτέρω κατανοητή η συγκεκριμένη έννοια προχωράμε στην ανάπτυξη δύο παραδειγμάτων που παρουσιάζουν με φυσιολογικό τρόπο τη σχέση που δίνει την από κοινού εντροπία.

[7] Σε αυτό το σημείο μπορούμε να θυμηθούμε πως την ίδια διαδικασία ακολουθήσαμε όταν θέλαμε να υπολογίσουμε το πλήθος των ερωτήσεων ανά σύμβολο που χρειάζονταν για να βρούμε ποια τιμή της τυχαίας μεταβλητής Υ επιλέχθηκε. Συγκεκριμένα μεταφράζοντας το πρόβλημα στην γλώσσα των ισοπίθανων ενδεχομένων σπάσαμε την επιλογή από 8 ισοπίθανα ορθογώνια σε δύο στάδια. Στο πρώτο στάδιο βρίσκαμε το υποσύνολο που επιλέχθηκε και στο δεύτερο στάδιο βρίσκαμε το ορθογώνιο του υποσυνόλου που επιλέχθηκε.

Παράδειγμα 1.2. *Έστω ότι έχουμε μια ακολουθία ανεξάρτητων και ισόνομων τυχαίων μεταβλητών* $\{X_n\}_{n\in\mathbb{N}}$ $X_n \sim P_X(x),\ \forall n \in \mathbb{N}$ *και αυτή τη φορά ενδιαφερόμαστε να ποσοτικοποιήσουμε την πληροφορία που περιλαμβάνεται σε μια περασμένη ακολουθία μήκους* n $(X_1,\ X_2, \cdots, X_n)$ *η οποία παίρνει τιμές στο πεπερασμένο σύνολο* $\times_{i=1}^{n}\mathcal{X}_i = \mathcal{X}_1 \times \mathcal{X}_2 \times \cdots \times \mathcal{X}_n$[8] .

Κάθε συμβολοσειρά μήκους n $\mathbf{x_1^n} = x_1,\ x_2, \cdots, x_n$ *αποτελεί ένα στιγμιότυπο της* $\mathbf{X_1^n} = X_1,\ X_2, \cdots, X_n$. *Επειδή η ακολουθία* $\mathbf{X_1^n}$ *αποτελείται από ανεξάρτητες και ισόνομες τυχαίες μεταβλητές έπεται ότι:*

$$Pr[\mathbf{X_1^n} = \mathbf{x_1^n}] = Pr[X_1 = x_1] \cdot Pr[X_2 = x_2] \cdot ... \cdot Pr[X_n = x_n] = \prod_{i=1}^{n} Pr[X_i = x_i]$$

Έστω $\mathbf{x_1^n} = x_1x_2...x_n$ *ένα τυχαίο στιγμιότυπο. Ξέρουμε ότι η ιδιοπληροφορία της* x_1 *είναι:*

$$\log \frac{1}{Pr[X_1 = x_1]}$$

Άρα η πληροφορία που περιέχεται γενικά στο στιγμιότυπο $x_1x_2...x_n$ *είναι ίση με το άθροισμα των πληροφοριών που περιέχονται στις τιμές της:*

$$log\frac{1}{Pr[X_1 = x_1]} + log\frac{1}{Pr[X_2 = x_2]} + ... + log\frac{1}{Pr[X_n = x_n]} = log\frac{1}{Pr[X_1 = x_1]} \cdot \frac{1}{Pr[X_2 = x_2]} \cdot \frac{1}{Pr[X_n = x_n]} = log\frac{1}{Pr[\mathbf{X_1^n} = \mathbf{x_1^n}]}$$

Άρα η μέση πληροφορία που περιέχεται σε μία τυχαία συμβολοσειρά μήκους n *είναι ο σταθμισμένος μέσος των ιδιοπληροφοριών των τιμών των τυχαίων μεταβλητών που συμμετέχουν στην συμβολοσειρά.*

$$H(X_1, ..., X_n) = H(\mathbf{X_1^n}) = \sum_{\mathbf{x_1^n} \in \times_{i=1}^{n}\mathcal{X}_i} Pr[\mathbf{X_1^n} = \mathbf{x_1^n}] \cdot \log \frac{1}{Pr[\mathbf{X_1^n} = \mathbf{x_1^n}]}$$

Παράδειγμα 1.3. *Στο παράδειγμα αυτό θα αναλογιστούμε την περίπτωση όπου οι τυχαίες μεταβλητές δεν είναι ανεξάρτητες και θα διαπιστώσουμε ότι η σχέση για την από κοινού εντροπία παραμένει ίδια.*

Ξεκινάμε θεωρώντας μία τυχαία συμβολοσειρά $\mathbf{x_1^n} = x_1x_2...x_n$. *Τότε η πληροφορία που περιέχεται στο πρώτο σύμβολο είναι:* $\log \frac{1}{Pr[X_1 = x_1]}$,

στο δεύτερο: $\log \frac{1}{Pr[X_2 = x_2|X_1 = x_1]}$,

στο τρίτο: $\log \frac{1}{Pr[X_3 = x_3|X_2 = x_2, X_1 = x_1]} = \log \frac{1}{Pr[X_3 = x_3|\mathbf{X_1^2} = \mathbf{x_1^2}]}$

και επαγωγικά στον $n-$*οστό σύμβολο η πληροφορία είναι:*

$\log \frac{1}{Pr[X_n = x_n|X_{n-1} = x_{n-1}, ..., X_1 = x_1]} = \log \frac{1}{Pr[X_n = x_n|\mathbf{X_1^{n-1}} = \mathbf{x_1^{n-1}}]}$

Άρα συνολικά η πληροφορία που περιέχεται στην ακολουθία είναι:

$\log \frac{1}{Pr[X_1 = x_1]} + \log \frac{1}{Pr[X_2 = x_2|X_1 = x_1]} + \cdots + \log \frac{1}{Pr[X_n = x_n|\mathbf{X_1^{n-1}} = \mathbf{x_1^{n-1}}]} =$

$\log \frac{1}{Pr[X_1 = x_1]} \cdot \frac{1}{Pr[X_2 = x_2|X_1 = x_1]} \cdots \frac{1}{Pr[X_n = x_n|\mathbf{X_1^{n-1}} = \mathbf{x_1^{n-1}}]} =$

[8] Όπου με το σύμβολο $\times$ εννοούμε το καρτεσιανό γινόμενο.

$$\log \frac{1}{Pr[X_1 = x_1]Pr[X_2 = x_2|X_1 = x_1]Pr[X_n = x_n|\mathbf{X_1^{n-1}} = \mathbf{x_1^{n-1}}]} =$$

$$\log \frac{1}{\prod_{i=1}^{n} Pr[X_i|\mathbf{X_1^{i-1}} = \mathbf{x_1^{i-1}}]}$$

Ο παρονομαστής της τελευταίας σχέσης σύμφωνα με τον πολλαπλασιαστικό κανόνα των πιθανοτήτων είναι η από κοινού κατανομή του διανύσματος τυχαίων μεταβλητών $(X_1, X_2, ..., X_n)$, όποτε η πληροφορία που περιέχεται τελικά στην συμβολοσειρά (ιδιοπληροφορία της συμβολοσειράς) $\mathbf{x_1^n} = x_1x_2...x_n$ είναι:

$$\log \frac{1}{Pr[\mathbf{X_1^n} = \mathbf{x_1^n}]}$$

Άρα καταλήγουμε πάλι ότι η μέση πληροφορία που περιέχεται σε μία τυχαία συμβολοσειρά μήκους n είναι ο σταθμισμένος μέσος των ιδιοπληροφοριών των τιμών των τυχαίων μεταβλητών που συμμετέχουν στην συμβολοσειρά.

$$H(X_1, ..., X_n) = H(\mathbf{X_1^n}) = \sum_{\mathbf{x_1^n} \in \times_{i=1}^{n} \mathcal{X}_i} Pr[X_1 = x_1, ..., X_n = x_n] \log \frac{1}{Pr[X_1 = x_1, ..., X_n = x_n]} \Rightarrow$$

$$H(\mathbf{X_1^n}) = \sum_{\mathbf{x_1^n} \in \times_{i=1}^{n} \mathcal{X}_i} Pr[\mathbf{X_1^n} = \mathbf{x_1^n}] \log \frac{1}{Pr[\mathbf{X_1^n} = \mathbf{x_1^n}]}$$

Ορισμός 1.2. *Έστω n διακριτές τ.μ $X_1, X_2, ...X_n$ που παίρνουν τιμές στα πεπερασμένα σύνολα $\mathcal{X}_1, \mathcal{X}_2, ..., \mathcal{X}_n$ αντίστοιχα με από κοινού σ.μ.π $P_{X_1,...,X_n}(x_1, ..., x_n) = Pr[X_1 = x_1, ..., X_n = x_n] \quad \forall x_1 \in \mathcal{X}_1, ..., x_n \in \mathcal{X}_n$ Ορίζουμε την **από κοινού εντροπία** ώς:*

$$\boxed{H(X_1, ..., X_n) = \sum_{x_1 \in \mathcal{X}_1, ..., x_n \in \mathcal{X}_n} Pr[X_1 = x_1, ..., X_n = x_n] log \frac{1}{Pr[X_1 = x_1, ..., X_n = x_n]}} \Rightarrow$$

$$\boxed{H(\mathbf{X_1^n}) = \sum_{\mathbf{x_1^n} \in \times_{i=1}^{n} \mathcal{X}_i} Pr[\mathbf{X_1^n} = \mathbf{x_1^n}] \log \frac{1}{Pr[\mathbf{X_1^n} = \mathbf{x_1^n}]}} \tag{1.2}$$

Για $n = 2$ η παραπάνω σχέση γράφεται ως:

$$H(X, Y) = \sum_{x \in \mathcal{X}, y \in \mathcal{Y}} Pr[X = x, Y = y] log \frac{1}{Pr[X = x, Y = y]}' \tag{1.3}$$

Παράδειγμα 1.4. *Έστω ότι έχουμε μία τυχαία μεταβλητή X που παίρνει τις τιμές $\{a, b, c, d\}$ και μία τυχαία μεταβλητή $\mathcal{Y}$ η οποία παίρνει τις τιμές $\{Μπλε, Κόκκινο\}$. Οι μεταβλητέ $X, \mathcal{Y}$ ακολουθούν την από κοινού σ.μ.π*

Χρώμα	*a*	*b*	*c*	*d*
Μπλε	$\frac{1}{8}$	$\frac{3}{8}$	*0*	*0*
Κόκκινο	*0*	*0*	$\frac{3}{8}$	$\frac{1}{8}$

Βρείτε την από κοινού εντροπία.

Λύση:

$$H(X, Y) = \sum_{x \in \mathcal{X}} \sum_{y \in \mathcal{Y}} Pr[X = x, Y = y] log \frac{1}{Pr[X = x, Y = y]} =$$

$$\frac{1}{8} \log_2 8 + \frac{3}{8} \log_2 \frac{8}{3} + \frac{3}{8} \log_2 \frac{8}{3} + \frac{1}{8} \log_2 8 = 0.75$$

Θεώρημα 1.1. *(Κανόνας της αλυσίδας για ανεξάρτητες μεταβλητές) Έστω $X_1, X_2, ..., X_n$ ανεξάρτητες διακριτές τ.μ που παίρνουν τιμές στα πεπερασμένα σύνολα $\mathcal{X}_1, \mathcal{X}_2, ..., \mathcal{X}_n$. Τότε:*

$$H(X_1, X_2, ..., X_n) = \sum_{i=1}^{n} H(X_i) \tag{1.4}$$

Απόδειξη

$$H(\mathbf{X_1^n}) = \sum_{\mathbf{x_1^n} \in \times_{i=1}^{n} \mathcal{X}_i} Pr[\mathbf{X_1^n} = \mathbf{x_1^n}] \log \frac{1}{Pr[\mathbf{X_1^n} = \mathbf{x_1^n}]}$$

Επειδή όμως οι τυχαίες μεταβλητές είναι ανεξάρτητες έπεται:

$$Pr[\mathbf{X_1^n} = \mathbf{x_1^n}] = \prod_{i=1}^{n} Pr[X_i = x_i]$$

οπότε:

$$H(\mathbf{X_1^n}) = \sum_{\mathbf{x_1^n} \in \times_{i=1}^{n} \mathcal{X}_i} Pr[\mathbf{X_1^n} = \mathbf{x_1^n}] \log \prod_{i=1}^{n} \frac{1}{Pr[X_i = x_i]} \Rightarrow$$

$$H(\mathbf{X_1^n}) = \sum_{\mathbf{x_1^n} \in \times_{i=1}^{n} \mathcal{X}_i} Pr[\mathbf{X_1^n} = \mathbf{x_1^n}] \sum_{i=1}^{n} \log \frac{1}{Pr[X_i = x_i]} \Rightarrow$$

$$H(\mathbf{X_1^n}) = \sum_{i=1}^{n} \sum_{\mathbf{x_1^n} \in \times_{i=1}^{n} \mathcal{X}_i} Pr[\mathbf{X_1^n} = \mathbf{x_1^n}] \log \frac{1}{Pr[X_i = x_i]}$$

Από τον ορισμό των περιθωρίων κατανομών γνωρίζουμε ότι:

$$Pr[X_j = x_j] = \sum_{\substack{\mathbf{x_1^n} \in \times_{i=1}^{n} \mathcal{X}_i \\ i \neq j}} Pr[\mathbf{X_1^n} = \mathbf{x_1^n}]$$

άρα:

$$H(\mathbf{X_1^n}) = \sum_{i=1}^{n} \sum_{x_i \in \mathcal{X}_i} Pr[X_i = x_i] log \frac{1}{Pr[X_i = x_i]} \Rightarrow$$

$$H(\mathbf{X_1^n}) = \sum_{i=1}^{n} H(X_i)$$

Το παραπάνω θεώρημα αποτελεί έναν από τους πολλούς κανόνες της αλυσίδας που θα δούμε στο παρών κεφάλαιο. Κατά την πρώτη ανάγνωση, τον κανόνα της αλυσίδας μπορούμε να τον κατανοήσουμε ως μία απλή εφαρμογή του πολλαπλασιαστικού νόμου των πιθανοτήτων για ανεξάρτητα ενδεχόμενα μαζί με ιδιότητες των λογαρίθμων. Ο κανόνας της αλυσίδας όμως κρύβει κάτι σημαντικότερο. Μας δείχνει πως μπορούμε να "σπάσουμε" σε στάδια τον υπολογισμό της εντροπίας ενός διανύσματος τυχαίων μεταβλητών έτσι ώστε το κάθε στάδιο να περιέχει τον υπολογισμό απλούστερων ποσοτήτων από ότι το συνολικό διάνυσμα. Στην περίπτωση των ανεξάρτητων μεταβλητών για παράδειγμα μας λέει πως αν έχουμε συγκεντρώσει τις σ.μ.π n ανεξάρτητων τυχαίων μεταβλητών δεν είναι ανάγκη να υπολογίσουμε την από κοινού σ.μ.π και ύστερα την εντροπία του διανύσματος $(X_1, \cdots, X_n)$ αλλά μπορούμε απευθείας να εξάγουμε την ποσότητα $H(X_1, \cdots, X_n)$ αθροίζοντας τις επιμέρους εντροπίες των μεταβλητών που συμμετέχουν στο διάνυσμα. Τέτοιοι ευελιξίες στους υπολογισμούς είναι απαραίτητες για πολλές πρακτικές εφαρμογές όταν είναι αναγκαίο να τους κρατήσουμε στο οικονομικότερα εφικτό επίπεδο. Για να γίνει τελείως κατανοητή η παραπάνω κατάσταση σκεφτόμαστε το εξής

παράδειγμα. Έστω ότι έχουμε δύο ανεξάρτητες τυχαίες μεταβλητές X και Υ που η μία παίρνει 4 τιμές και η άλλη 3 αντίστοιχα και αποθηκεύουμε τις σ.μ.π τους σε δύο διανύσματα μεγέθους 3 και 4. Τότε η από κοινού σ.μ.π μπορεί να υπολογιστεί εκτελώντας τον πολλαπλασιασμό μεταξύ των δυο διανυσμάτων. Με αυτό τον τρόπο θα προκύψει ένας πίνακας μεγέθους 4×3.

$$\begin{bmatrix} Pr[X=x_1] \\ Pr[X=x_2] \\ Pr[X=x_3] \\ Pr[X=x_4] \end{bmatrix} \times \begin{bmatrix} Pr[Y=y_1] & Pr[Y=y_2] & Pr[Y=y_3] & \end{bmatrix} =$$

$$\begin{bmatrix} Pr[X=x_1]Pr[Y=y_1] & Pr[X=x_1]Pr[Y=y_2] & Pr[X=x_1]Pr[Y=y_3] \\ Pr[X=x_2]Pr[Y=y_1] & Pr[X=x_2]Pr[Y=y_2] & Pr[X=x_2]Pr[Y=y_3] \\ Pr[X=x_3]Pr[Y=y_1] & Pr[X=x_3]Pr[Y=y_2] & Pr[X=x_3]Pr[Y=y_3] \\ Pr[X=x_4]Pr[Y=y_1] & Pr[X=x_4]Pr[Y=y_2] & Pr[X=x_4]Pr[Y=y_3] \end{bmatrix}$$

Αυτό όμως θα έχει σαν αποτέλεσμα να κάνουμε 12 πολλαπλασιασμούς. Έπειτα θα πρέπει να υπολογίσουμε την αντίστροφη πιθανότητα 12 πιθανοτήτων και να την λογαριθμίσουμε. Τέλος θα πρέπει να πολλαπλασιαστούν αυτοί οι 12 λογάριθμοι με τις αντίστοιχες από κοινού πιθανότητες και μετά να προστεθούν μεταξύ τους. Μιλάμε λοιπόν ότι έχουμε συνολικά:

12 πολλαπλασιασμούς + 12 ευρέσεις αντιστρόφων πιθανοτήτων + 12 λογαριθμίσεις + 12 πολλαπλασιασμούς των λογαρίθμων με την αντίστοιχες από κοινού πιθανότητες και 11 προσθέσεις = 59 πράξεις.

Για να υπολογίσουμε όμως την εντροπία μίας τυχαίας μεταβλητής που παίρνει n τιμές χρειαζόμαστε:

n αντιστροφές πιθανοτήτων + n λογαριθμίσεις + n πολλαπλασιασμούς των λογαρίθμων με τις αντίστοιχες πιθανότητες + $n-1$ προσθέσεις = $4 \cdot n - 1$ πράξεις. Άρα για να υπολογίσουμε την εντροπία της X χρειαζόμαστε 15 πράξεις και της Υ 11. Καταλαβαίνουμε πώς αν χρησιμοποιήσουμε τον κανόνα της αλυσίδας και δεν υπολογίσουμε την από κοινού εντροπία αλλά τις επιμέρους εντροπίες αντί για 59 πράξεις θα χρειαστούμε μόλις 26 που είναι σαφώς πολύ λιγότερες.

Θέτοντας τον παραπάνω συλλογισμό σε ένα γενικό πλαίσιο υποθέτουμε πως έχουμε ένα διάνυσμα n ανεξάρτητων τυχαίων μεταβλητών που η κάθε μία παίρνει n τιμές. Επειδή η εντροπία κάθε μεταβλητής που παίρνει n τιμές κοστίζει $4 \cdot n - 1$, η ποσότητα $\sum_{i=1}^{n} H(X_i)$ θα κοστίζει $n \cdot (4 \cdot n - 1) = \mathcal{O}(n^2)$. Αντίστοιχα το να βρούμε την ποσότητα $H(X_1, \cdots, X_n)$ θα κοστίζει n^n πολλαπλασιασμούς για την εύρεση του πίνακα των από κοινού πιθανοτήτων και n^n αντιστροφές πιθανοτήτων και n^n λογαριθμίσεις και n^n πολλαπλασιασμούς των λογαρίθμων με την αντίστοιχες από κοινού πιθανότητες $n^n - 1$ προσθέσεις. Άρα η εύρεση του $H(X, \cdots, X_n)$ υπολογιστικά κοστίζει $\mathcal{O}(n^n)$. Πλέον νομίζουμε ότι είναι πασιφανής η χρησιμότητα των κανόνων αλυσίδας.

1.3 Δεσμευμένη Εντροπία

Είναι διαισθητικά εύκολο να σκεφτούμε ότι όσο περισσότερες πληροφορίες έχουμε για κάποιο σύστημα ή για κάποιο φαινόμενο τόσο λιγότερη αβεβαιότητα υπάρχει ως προς την συμπεριφορά του. Η έννοια αυτή εκφράζεται από την δεσμευμένη εντροπία, η οποία ποσοτικοποιεί την μέση πληροφορία που λαμβάνουμε από μία τυχαία μεταβλητή όταν είναι γνωστές οι τιμές κάποιων άλλων, δηλαδή εκφράζει το κατά πόσο συνεισφέρει η τυχαία μεταβλητή στην κατανόηση του υπό μελέτη συστήματος

Για να εξάγουμε τη σχέση που δίνει την δεσμευμένη εντροπία μπορούμε να σκεφτούμε την παρακάτω περίπτωση:

Έστω ότι έχουμε ένα δυναμικό σύστημα που η συμπεριφορά του μοντελοποιείται από n τυχαίες μεταβλητές. Ακολουθώντας ένα δειγματικό μονοπάτι του παρατηρούμε ότι τις χρονικές στιγμές $1, 2, ..., n-1$ βρισκόταν στις καταστάσεις $x_1, x_2, ..., x_{n-1}$ αντίστοιχα. Θέλουμε να υπολογίσουμε την αβεβαιότητα που υπάρχει στο να μεταβεί σε κάποια κατάσταση x_n που ανήκει σε ένα σύνολο πιθανών καταστάσεων $\mathcal{X}_n$. Η σ.μ.π της μετάβασης από την κατάσταση x_{n-1} στη $x_n \in \mathcal{X}_n$ θα είναι:

$\{Pr[X_n = x_n | X_1 = x_1, ..., X_{n-1} = x_{n-1}]\}_{x_n \in \mathcal{X}_n} = \{Pr[X_n = x_n | \mathbf{X_1^{n-1}} = \mathbf{x_1^{n-1}}]\}_{x_n \in \mathcal{X}_n}$.

Σύμφωνα με την σχέση (1.1) η πληροφορία που εμπεριέχεται στη μετάβαση $x_{n-1} \to x_n$ θα είναι ο σταθμισμένος μέσος των πληροφοριών που μπορούμε να πάρουμε από κάθε πιθανή μετάβαση $x_{n-1} \to x_n \ \forall x_n \in \mathcal{X}_n$:

$$H(X_n|\mathbf{X_1^{n-1}}) = \sum_{x_n \in \mathcal{X}_n} Pr[X_n = x_n|\mathbf{X_1^{n-1}} = \mathbf{x_1^{n-1}}] \cdot \log \frac{1}{Pr[X_n = x_n|\mathbf{X_1^{n-1}} = \mathbf{x_1^{n-1}}]}$$

Η εντροπία με βάση όλα τα δειγματικά μονοπάτια $x_1x_2...x_{n-1}$ είναι ο σταθμισμένος μέσο τής εντροπίας του κάθε μονοπατιού:

$$H(X_n|\mathbf{X_1^{n-1}}) = \sum_{\mathbf{x_1^{n-1}} \in \times_{\mathbf{i=1}}^{\mathbf{n-1}} \mathcal{X}_\mathbf{i}} Pr[\mathbf{X_1^{n-1}} = \mathbf{x_1^{n-1}}] H(X_n|\mathbf{X_1^{n-1}}) \Rightarrow$$

$$H(X_n|\mathbf{X_1^{n-1}}) = \sum_{\mathbf{x_1^{n-1}} \in \times_{\mathbf{i=1}}^{\mathbf{n-1}} \mathcal{X}_\mathbf{i}} Pr[\mathbf{X_1^{n-1}} = \mathbf{x_1^{n-1}}] \sum_{x_n \in \mathcal{X}_n} Pr[X_n = x_n|\mathbf{X_1^{n-1}} = \mathbf{x_1^{n-1}}] \log \frac{1}{Pr[X_n = x_n|\mathbf{X_1^{n-1}} = \mathbf{x_1^{n-1}}]} \Rightarrow$$

$$H(X_n|\mathbf{X_1^{n-1}}) = \sum_{\mathbf{x_1^{n}} \in \times_{\mathbf{i=1}}^{\mathbf{n}} \mathcal{X}_\mathbf{i}} Pr[\mathbf{X_1^{n-1}} = \mathbf{x_1^{n-1}}] Pr[X_n = x_n|\mathbf{X_1^{n-1}} = \mathbf{x_1^{n-1}}] \log \frac{1}{Pr[X_n = x_n|\mathbf{X_1^{n-1}} = \mathbf{x_1^{n-1}}]} \Rightarrow$$

$$H(X_n|\mathbf{X_1^{n-1}}) = \sum_{\mathbf{x_1^{n}} \in \times_{\mathbf{i=1}}^{\mathbf{n}} \mathcal{X}_\mathbf{i}} Pr[\mathbf{X_1^{n}} = \mathbf{x_1^{n}}] \log \frac{1}{Pr[X_n = x_n|\mathbf{X_1^{n-1}} = \mathbf{x_1^{n-1}}]} \Rightarrow$$

Ορισμός 1.3. *Έστω $X_1, X_2, ..., X_n$ διακριτές τ.μ που παίρνουν τιμές στα πεπερασμένα σύνολα $\mathcal{X}_1, \mathcal{X}_2, ..., \mathcal{X}_n$. Τότε η* ***δεσμευμένη εντροπία*** *ορίζεται ως:*

$$\boxed{H(X_n|X_1, ..., X_{n-1}) = \sum_{x_1 \in \mathcal{X}_1, ..., x_n \in \mathcal{X}_n} Pr[X_1 = x_1, ..., X_n = x_n] log \frac{1}{Pr[X_n = x_n|X_1 = x_1, ..., X_{n-1} = x_{n-1}]}}$$

$$\boxed{H(X_n|\mathbf{X_1^{n-1}}) = \sum_{\mathbf{x_1^{n}} \in \times_{\mathbf{i=1}}^{\mathbf{n}} \mathcal{X}_\mathbf{i}} Pr[\mathbf{X_1^{n}} = \mathbf{x_1^{n}}] \log \frac{1}{Pr[X_n = x_n|\mathbf{X_1^{n-1}} = \mathbf{x_1^{n-1}}]}} \tag{1.5}$$

Για $n = 2$ η σχέση γίνεται:

$$H(Y|X) = \sum_{x \in \mathcal{X}, y \in \mathcal{Y}} Pr[X = x, Y = y] log \frac{1}{Pr[Y = x|X = x]} \tag{1.6}$$

Παράδειγμα 1.5. *Όπως και στο παράδειγμα 1.5 πάλι δίνονται μία τυχαία μεταβλητή X που παίρνει τις τιμές $\{a, b, c, d\}$ και μία τυχαία μεταβλητή Υ η οποία παίρνει τις τιμές {Μπλε, Κόκκινο}. Οι μεταβλητέ X, Υ ακολουθούν την από κοινού σ.μ.π*

Χρώμα	a	b	c	d
Μπλε	$\frac{1}{8}$	$\frac{3}{8}$	*0*	*0*
Κόκκινο	*0*	*0*	$\frac{3}{8}$	$\frac{1}{8}$

Βρείτε την δεσμευμένη εντροπία $H(X|Y)$.

Λύση:

$$Pr[Y = Μπλε] = \sum_{x \in \mathcal{X}} Pr[X = x, Y = Μπλε] = \frac{1}{8} + \frac{3}{8} = \frac{1}{2}$$

$$Pr[Y = Κόκκινο] = \sum_{x \in \mathcal{X}} Pr[X = x, Y = Κόκκινο] = \frac{3}{8} + \frac{1}{8} = \frac{1}{2}$$

$$\begin{aligned}
&H(X|Y) = Pr[Y = Μπλε] \cdot H(X|Y = Μπλε) + \\
&Pr[Y = Κόκκινο] \cdot H(X|Y =, Y = Κόκκινο) \Rightarrow \\
&H(X|Y) = \frac{1}{2} \cdot H(\frac{1}{8}, \frac{3}{8}) + \frac{1}{2} \cdot H(\frac{3}{8}, \frac{1}{8}) \Rightarrow \\
&H(X|Y) = 0.905639 \approx 0.91
\end{aligned}$$

Θεώρημα 1.2. *(Κανόνας της αλυσίδας για εξαρτημένες μεταβλητές) Έστω $X_1, X_2, ..., X_n$ διακριτές τ.μ που παίρνουν τιμές στα πεπερασμένα σύνολα $\mathcal{X}_1, \mathcal{X}_2, ..., \mathcal{X}_n$. Τότε*

$$H(X_1, ..., X_n) = \sum_{i=1}^{n} H(X_i|X_1, ..., X_{i-1})$$

ή

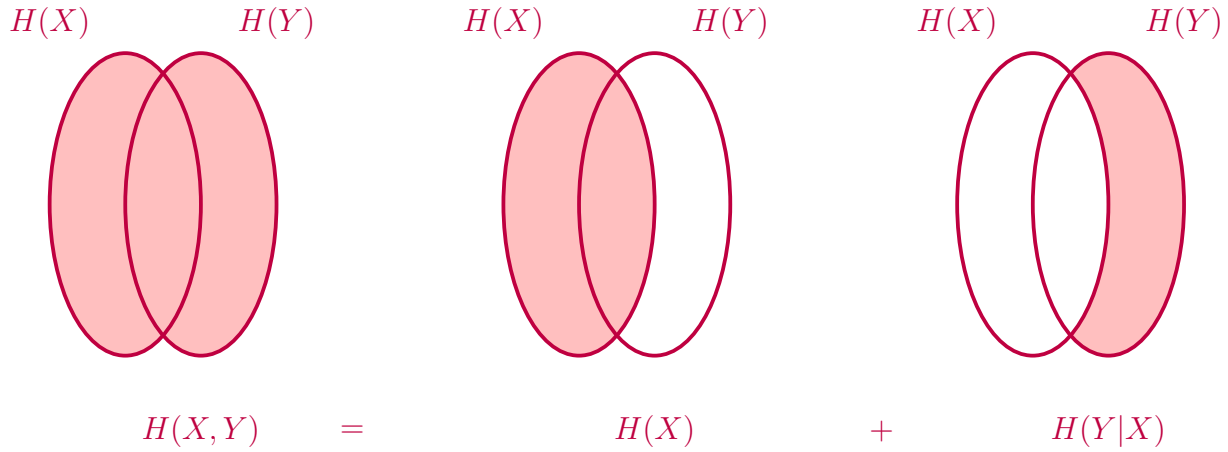
$$H(\mathbf{X_1^n}) = \sum_{i=1}^{n} H(X_i|\mathbf{X_1^{i-1}} = \mathbf{x_1^{i-1}}) \tag{1.7}$$

Απόδειξη
Από τον πολλαπλασιαστικό νόμο των πιθανοτήτων γνωρίζουμε ότι:

$$Pr[\mathbf{X_1^n} = \mathbf{x_1^n}] = \prod_{i=1}^{n} Pr[X_i = x_i|\mathbf{X_1^{i-1}} = \mathbf{x_1^{i-1}}]$$

Οπότε:

$$\begin{aligned}
H(\mathbf{X_1^n}) &= \sum_{\mathbf{x_1^n} \in \times_{\mathbf{i=1}}^{\mathbf{n}} \mathcal{X}_\mathbf{i}} Pr[\mathbf{X_1^n} = \mathbf{x_1^n}] \log \frac{1}{Pr[\mathbf{X_1^n} = \mathbf{x_1^n}]} \Rightarrow \\
H(\mathbf{X_1^n}) &= \sum_{\mathbf{x_1^n} \in \times_{\mathbf{i=1}}^{\mathbf{n}} \mathcal{X}_\mathbf{i}} Pr[\mathbf{X_1^n} = \mathbf{x_1^n}] \log \prod_{i=1}^{n} \frac{1}{Pr[X_i = x_i|\mathbf{X_1^{i-1}} = \mathbf{x_1^{i-1}}]} \Rightarrow \\
H(\mathbf{X_1^n}) &= \sum_{\mathbf{x_1^n} \in \times_{\mathbf{i=1}}^{\mathbf{n}} \mathcal{X}_\mathbf{i}} Pr[\mathbf{X_1^n} = \mathbf{x_1^n}] \log \sum_{i=1}^{n} \frac{1}{Pr[X_i - x_i|\mathbf{X_1^{i-1}} = \mathbf{x_1^{i-1}}]} \Rightarrow \\
H(\mathbf{X_1^n}) &= \sum_{i=1}^{n} \sum_{\mathbf{x_1^n} \in \times_{\mathbf{i=1}}^{\mathbf{n}} \mathcal{X}_\mathbf{i}} Pr[\mathbf{X_1^n} = \mathbf{x_1^n}] \log \frac{1}{Pr[X_i = x_i|\mathbf{X_1^{i-1}} = \mathbf{x_1^{i-1}}]} \Rightarrow \\
H(\mathbf{X_1^n}) &= \sum_{i=1}^{n} H(X_i|\mathbf{X_1^{i-1}})
\end{aligned}$$



Σχήμα 1.8: Η οπτικοποίηση του κανόνα της αλυσίδας για εξαρτημένες μεταβλητές στην περίπτωση που το $n = 2$

Ο κανόνας τις αλυσίδας για εξαρτημένες μεταβλητές βοηθάει και αυτός να υπολογίσουμε όσο πιο οικονομικότερα γίνεται την εντροπία που υπάρχει σε ένα διάνυσμα τυχαίων μεταβλητών όταν οι μεταβλητές είναι εξαρτημένες. Στην περίπτωση αυτή ίσως φαίνεται λίγο πιο δύσκολα από ότι στην πρώτη (ανεξάρτητες μεταβλητές) αλλά είναι βοηθητικό να σκεφτούμε ως εξής. Αν δύο τυχαίες μεταβλητές X και Υ είναι ανεξάρτητες δεδομένου μίας τρίτης Z τότε

$$Pr[X = x, Y = y|Z = z] = Pr[X = x|Z = z] \cdot Pr[Y = y|Z = z]$$

ή ισοδύναμα

$$Pr[X = x|Z = z, Y = y] = Pr[X = x|Z = z]$$

Ας θυμηθούμε ακόμη τον ορισμό της δεσμευμένη πιθανότητα:

$$Pr[X|Y] = \frac{Pr[X \cap Y]}{Pr[Y]}$$

Άρα προκειμένου να υπολογίσουμε την δεσμευμένη πιθανότητα:

$$Pr[X_n = x_n|X_1 = x_1, \cdots X_{n-1} = x_{n-1}] = \frac{Pr[X_1 = x_1, \cdots X_n = x_n]}{Pr[X_1 = x_1, \cdots X_{n-1} = x_{n-1}]}$$

χρειάζεται να ξέρουμε δύο από κοινού κατανομές μεγέθους n και $n-1$ αντίστοιχα που από ότι είδαμε δεν είναι πολύ οικονομικό να τις υπολογίσουμε . Ακόμα και αν έχουμε n δίτιμες εξαρτημένες τυχαίες μεταβλητές τότε η από κοινού σ.μ.π τους θα περιέχει 2^n εγγραφές. Άρα όσο μικρό και αν είναι το πεπερασμένο σύνολο στο οποίο ορίζονται οι τυχαίες μεταβλητές που μοντελοποιούν το πρόβλημα μας, καθώς το πλήθος τους αυξάνει γραμμικά το πλήθος των πιθανοτήτων της από κοινού σ.μ.π θα αυξάνει εκθετικά. Το κάλο στις πρακτικές εφαρμογές είναι πως μπορεί ένα σύστημα που μελετάμε να μοντελοποιείται από ένα διάνυσμα n εξαρτημένων τυχαίων μεταβλητών αλλά αν μελετήσουμε υποσύνολα αυτών μπορούμε να βρούμε πολλές δεσμευμένες ανεξαρτησίες. Για παράδειγμα έστω ότι έχουμε το σύνολο τυχαίων μεταβλητών (Φωτιά, Καπνός, Συναγερμός). Τότε οι τυχαίες μεταβλητές Φωτιά και Συναγερμός είναι ανεξάρτητες δεδομένου της μεταβλητής Καπνός (Φωτιά $\perp\!\!\!\perp$ Συναγερμός | Καπνός) καθώς η μεταβλητή Φωτιά δεν προκαλεί άμεσα την έναρξη του συναγερμού αλλά δια μέσου της μεταβλητής Καπνός. Ένα ακραίο παράδειγμα που φαίνεται ξεκάθαρα η χρησιμότητα του συγκεκριμένου κανόνα τις αλυσίδας είναι οι μαρκοβιανές αλυσίδες διότι τότε η από κοινού σ.μ.π των n τυχαίων μεταβλητών μπορεί να γραφτεί ως εξής:

$$Pr[X_1 = x_1, \cdots, X_n = x_n] = Pr[X_1 = x_1] \cdot Pr[X_2 = x_2|X_1 = x_1] \cdots Pr[X = x_n|X_{n-1} = x_{n-1}]$$

Η παραπάνω σχέση αυτόματα μεταφράζεται στην γλώσσα της εντροπίας σύμφωνα με τον κανόνα της αλυσίδας για εξαρτημένες μεταβλητές σε

$$H(X_1, \cdots, X_n) = \sum_{i=1}^{n} H(X_i|X_{i-1})$$

Ας πάμε λοιπόν σε αυτή την βέλτιστη περίπτωση των μαρκοβιανών αλυσίδων προκειμένου να υπολογίσουμε τα υπολογιστικά κόστη της δεξιάς και αριστερής μεριάς της τελευταίας εξίσωσης όταν έχουμε n τυχαίες μεταβλητές που η κάθε μία ορίζεται σε ένα πεπερασμένο σύνολο μεγέθους n. Τότε γνωρίζουμε πως το πλήθος των από κοινού πιθανοτήτων θα είναι n^n. Άρα για να υπολογίσουμε το $H(X_1, \cdots X_n)$ θα χρειαστούμε n^n αντιστροφές πιθανοτήτων και n^n λογαριθμίσεις και n^n πολλαπλασιασμούς των λογαρίθμων με τις αντίστοιχες πιθανότητες και n^n-1 προσθέσεις οπότε συνολικά έχουμε κόστος $\mathcal{O}(n^n)$. Στην περίπτωσή όμως που χρησιμοποιήσουμε τον κανόνα της αλυσίδας για εξαρτημένες τυχαίες μεταβλητές τότε πρώτα θα πρέπει να υπολογίσουμε όλες τις δεσμευμένες πιθανότητες τύπου $Pr[X_i = x_i|X_{i-1} = x_{i-1}] = \dfrac{Pr[X_i = x_i, X_{i-1} = x_{i-1}]}{Pr[X_{i-1} = x_{i-1}]}$. Τις κατανομές $Pr[X = x_i]$ τις έχουμε άρα θα πρέπει να βρούμε $n-1$ από κοινού σ.μ.π της μορφής $Pr[X_i = x_i, X_{i-1} = x_{i-1}]$. Αφού κάθε τυχαία μεταβλητή παίρνει n τιμές, η από κοινού σ.μ.π 2 τυχαίων μεταβλητών θα αποτελείται από n^2 τιμές. Για να βρούμε τις δεσμευμένες πιθανότητες της μορφής $Pr[X_i = x_i|X_{i-1} = x_{i-1}]$ από τον τύπο που δίνει την δεσμευμένη πιθανότητα βλέπουμε ότι χρειάζονται n^2 διαιρέσεις. Αφού βρούμε τον πίνακα δεσμευμένων πιθανοτήτων έπειτα χρειάζονται n^2 αντιστροφές πιθανοτήτων και n^2 λογαριθμίσεις και n^2 πολλαπλασιασμοί των λογαρίθμων με τις αντίστοιχες $n-1$ από κοινού πιθανότητες $Pr[X_i = x_i, X_{i-1} = x_{i-1}]$ και n^2-1 προσθέσεις. Άρα ο υπολογισμός του όρου $H(X_i|X_{i-1})$ έχει αλγοριθμικό κόστος $\mathcal{O}(n^2)$. Εκτός από τον πρώτο όρο του $\sum_{i=1}^{n} H(X_i|X_{i-1})$ που είναι ο $H(X_1)$ με κόστος υπολογισμού $\mathcal{O}(n)$ οι υπόλοιποι $n-1$ είναι της μορφής $H(X_i|X_{i-1})$ και έχουν κόστος $\mathcal{O}(n^2)$ οπότε καταλαβαίνουμε ότι το κόστος του αθροίσματος $\sum_{i=1}^{n} H(X_i|X_{i-1})$ θα είναι $\mathcal{O}(n^3)$

Επαγωγικά σκεπτόμενοι μπορούμε εύκολα να διαπιστώσουμε πως το κόστος υπολογισμού μίας δεσμευμένης εντροπίας $H(X_i|X_1, \cdots X_{i-1})$ θα είναι $\mathcal{O}(n^i)$. Άρα αν οι δεσμευμένες εντροπίες δεν έχουνε όλες τον ίδιο αριθμό δεσμευμένων τυχαίων μεταβλητών, δηλαδή μέσα στο άθροισμα $\sum_{i=1}^{n} H(X_i|\mathbf{X_1^{i-1}})$ περιέχονται όροι της μορφής $H(X_i|\mathbf{X_1^{i-1}})$, $H(X_j|\mathbf{X_1^{j-1}}), \cdots$ τότε το κόστος του αθροίσματος θα κυριαρχείται από τον όρο που έχει τις περισσότερες δεσμευμένες τυχαίες μεταβλητές καθώς αυτός είναι που χρειάζεται και την μεγαλύτερη από κοινού σ.μ.π για να υπολογιστεί. Άρα σε αυτή την περίπτωση το κόστος θα είναι $\mathcal{O}(n^k)$ όπου k το μέγεθος του μεγαλύτερου διανύσματος τυχαίων μεταβλητών που χρησιμοποιείται σε κάποιον από του όρους $H(X_i|\mathbf{X_1^{i-1}})$ που συμμετέχουν στο άθροισμα. Όσο το $k << n$ το κόστος υπολογισμού χρησιμοποιώντας τον κανόνα της αλυσίδας παραμένει πολυωνυμικό.

Πρόταση 1.1. *Έστω $X_1, X_2, ..., X_n, Y$ διακριτές τ.μ που παίρνουν τιμές στα πεπερασμένα σύνολα $\mathcal{X}_1, \mathcal{X}_2, ..., \mathcal{X}_n, \mathcal{Y}$. Τότε:*

$$\boxed{H(\mathbf{X_1^n}|Y) = \sum_{i=1}^{n} H(X_i|\mathbf{X_1^{i-1}}, Y)} \tag{1.8}$$

Απόδειξη

Από τον ορισμό της δεσμευμένης πιθανότητας ισχύει:

$$Pr[\mathbf{X_1^n}|Y = y] = \frac{Pr[\mathbf{X_1^n} = \mathbf{x_1^n}, Y = y]}{P[Y = y]} =$$

$$\frac{Pr[Y = y] \cdot \prod_{i=1}^{n} Pr[\mathbf{X_i}|\mathbf{X_1^{i-1}} = \mathbf{x_1^{i-1}}, Y = y]}{P[Y = y]} =$$

$$\prod_{i=1}^{n} Pr[X_i = x_i|\mathbf{X_1^{i-1}} = \mathbf{x_1^{i-1}}, Y = y]$$

Οπότε:

$$H(\mathbf{X_1^n}=\mathbf{x_1^n}|Y)=\sum_{\mathbf{x_1^n}\in\times_{\mathbf{i=1}}^{\mathbf{n}}\mathcal{X}_\mathbf{i},Y\in\mathcal{Y}} Pr[\mathbf{X_1^n}=\mathbf{x_1^n},Y=y]\log\frac{1}{Pr[\mathbf{X_1^n}=\mathbf{x_1^n}|Y=y]}\Rightarrow$$

$$H(\mathbf{X_1^n}=\mathbf{x_1^n}|Y)=\sum_{\mathbf{x_1^n}\in\times_{\mathbf{i=1}}^{\mathbf{n}}\mathcal{X}_\mathbf{i},Y\in\mathcal{Y}} Pr[\mathbf{X_1^n}=\mathbf{x_1^n},Y=y]\log\prod_{i=1}^{n}\frac{1}{Pr[X_i=x_i|\mathbf{X_1^{i-1}}=\mathbf{x_1^{i-1}},Y=y]}\Rightarrow$$

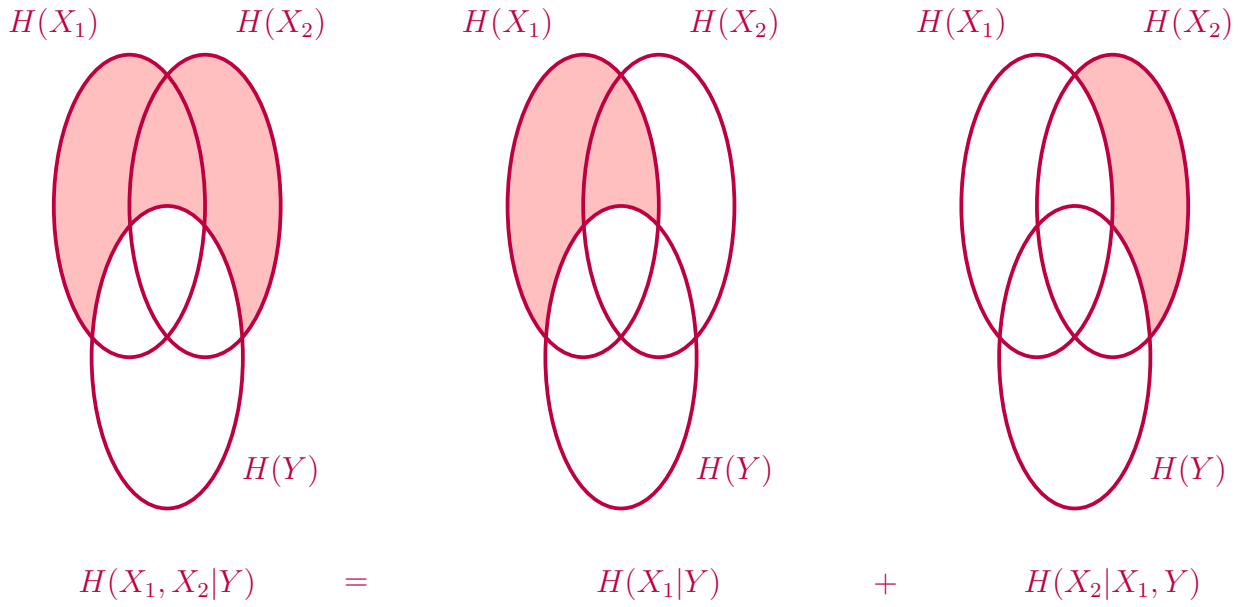
$$H(\mathbf{X_1^n}=\mathbf{x_1^n}|Y)=\sum_{\mathbf{x_1^n}\in\times_{\mathbf{i=1}}^{\mathbf{n}}\mathcal{X}_\mathbf{i},Y\in\mathcal{Y}} Pr[\mathbf{X_1^n}=\mathbf{x_1^n},Y=y]\sum_{i=1}^{n}\log\frac{1}{Pr[X_i=x_i|\mathbf{X_1^{i-1}}=\mathbf{x_1^{i-1}},Y=y]}\Rightarrow$$

$$H(\mathbf{X_1^n}=\mathbf{x_1^n}|Y)=\sum_{i=1}^{n}\sum_{\mathbf{x_1^n}\in\times_{\mathbf{i=1}}^{\mathbf{n}}\mathcal{X}_\mathbf{i},Y\in\mathcal{Y}} Pr[\mathbf{X_1^n}=\mathbf{x_1^n},Y=y]\log\frac{1}{Pr[\mathbf{X_1^{i-1}}=\mathbf{x_1^{i-1}},Y=y]}\Rightarrow$$

$$H(\mathbf{X_1^n}=\mathbf{x_1^n}|Y)=\sum_{i=1}^{n}H(X_i|\mathbf{X_1^{i-1}},Y)$$

Για $n=2$ η ιδιότητα αυτή γίνεται:

$$H(X_1,X_2|Y)=H(X_1|Y)+H(X_2|X_1,Y) \tag{1.9}$$



Σχήμα 1.9: Η οπτικοποίηση του κανόνα της αλυσίδας της πρότασης 1.1 στην περίπτωση που το $n=2$

1.4 Απόσταση Kullback-Leibler-Διαγώνια Εντροπία

Μέχρι στιγμής σε όσα παραδείγματα έχουμε δώσει και σε όσους ορισμούς έχουν διατυπωθεί αποτελεί δεδομένο η γνώση της συνάρτησης μάζας πιθανότητας της εκάστοτε μεταβλητής ή του διανύσματος τυχαίων μεταβλητών. Στην πράξη βέβαια τις περισσότερες φορές αυτό είναι ανέφικτο.

Παράδειγμα 1.6. *Ας αναλογιστούμε την περίπτωση που κάποιος προγραμματιστής προσπαθεί να αναπτύξει ένα συμπιεστή κειμένου που θα συμπιέζει όσο πιο αποδοτικά γίνεται πηγές χωρίς μνήμη.*[9]

[9]Μία πηγή χωρίς μνήμη μοντελοποιείται από μία ακολουθία ανεξάρτητων και ισόνομων τυχαίων μεταβλητών. Η έννοια αυτή θα αναλυθεί περεταίρω στο επόμενο κεφάλαιο

Ένας απλό τρόπος για να κατασκευαστεί ένας τέτοιος συμπιεστής είναι ο εξής:

Είσοδος: Κείμενο, μέγεθος διακριτοποίησης x

Μέχρι να τελειώσει το κείμενο:

1. Διάβασε x kilobyte (kb) του κείμενου και αποθήκευσε τον αριθμό εμφάνισης του κάθε συμβόλου σε μία κατάλληλη δομή δεδομένων

2. Χρησιμοποίησε τις σχετικές συχνότητες εμφάνισης των συμβόλων ως σ.μ.π για το συγκεκριμένο κομμάτι, χτίσε ένα δυαδικό δένδρο απόφασης και εξήγαγε τους κώδικές (όπως στο παράδειγμα 2).

3.Κωδικοποίησε το υπό επεξεργασία κομμάτι

4. Πήγαινε στο 1.

Έξοδος: Συμπιεσμένο κείμενο

Στα εισαγωγή του κεφαλαίου είχαν εμφανιστεί δύο πηγές χωρίς μνήμη με διαφορετικές κατανομές η κάθε μία. Η πρώτη επέλεγε 4 γράμματα με τυχαίο τρόπο ενώ η δεύτερη έδειχνε προτίμηση στο σύμβολο a. Τι θα γινόταν αν ο συμπιεστής ενώ έπρεπε να χρησιμοποιήσει για την συμπίεση του κειμένου την δεύτερη κατανομή χρησιμοποιούσε την πρώτη;

Τότε θα έφτιαχνε ένα κώδικά με μέσο μήκος 2 bits ανά σύμβολο ενώ θα έπρεπε να φτιάξει ένα κώδικα με μέσο μήκος 1.75 bits/σύμβολο. Καταλαβαίνουμε λοιπόν ότι ο συμπιεστής θα ξόδευε κατά μέσο όρο $0.25 bits/$σύμβολο παραπάνω από ότι χρειαζόταν για να κωδικοποιήσουμε την πηγή μας.

Ο προγραμματιστής θα μπορούσε να χρησιμοποιήσει το σενάριο αυτό για να ελέγξει την αποδοτικότητα του συμπιεστή που έφτιαξε. Χρησιμοποιώντας πηγές των οποίων ήδη ξέρει την θεωρητική κατανομή θα μπορούσε να μετρήσει κατά πόσο οι εμπειρικές κατανομές του συμπιεστή προσεγγίζουν τις θεωρητικές. Για κάθε πηγή ο αριθμός αυτός θα αποτελούσε ένα μέτρο για την προσαρμοστικότητα του συμπιεστή στα εκάστοτε δεδομένα. Ένα δεύτερο μέτρο βασισμένο στους αριθμούς αυτούς θα μπορούσε να αποτελέσει ένα δείκτη για την συνολική αποδοτικότητα του συμπιεστή. Υπάρχει άραγε ένας τέτοιος αριθμός που "μετράει" την απόσταση της θεωρητικής από την δειγματική κατανομή όσον αφορά στην πληροφορία που περιέχουν; Η απάντηση είναι καταφατική και η απόσταση αυτή λέγεται Kullback-Leibler η αλλιώς σχετική εντροπία.

Ορισμός 1.4. *Η σχετική εντροπία ή αλλιώς απόσταση $Kullaback - Leibler$ μεταξύ δύο διαφορετικών συναρτήσεων μάζας πιθανότητας $P[X = x]$ και $Q(X = x)$ ορίζεται ως:*

$$D(P||Q) = \sum_{x \in \mathcal{X}} P[X = x] \log \frac{P[X = x]}{Q[X = x]} \tag{1.10}$$

Επίσης ορίζεται $0 \log \frac{0}{0} = 0, \quad 0 \log \frac{0}{Q[X = x]} = 0$ *και* $P[X = x] \log \frac{P[X = x]}{0} = \infty$

Οι παραπάνω συμβάσεις έχουν νόημα αν αναλογιστούμε ότ $\lim_{x \to 0} xlogx = 0$ και $\lim_{x \to \infty} xlogx = \infty$.

Αν έχουμε ένα διάνυσμα τυχαίων μεταβλητών η παραπάνω σχέση διαμορφώνεται ως εξής:

$$D(P||Q) = \sum_{\mathbf{x_1^n} \in \times_{i=1}^{n} \mathcal{X}_i} P[\mathbf{X_1^n} = \mathbf{x_1^n}] \log \frac{P[\mathbf{X_1^n} = \mathbf{x_1^n}]}{Q[\mathbf{X_1^n} = \mathbf{x_1^n}]} \tag{1.11}$$

με τις αντίστοιχες συμβάσεις $0 \log \frac{0}{0} = 0, \quad 0 \log \frac{0}{Q[\mathbf{X_1^n} = \mathbf{x_1^n}]} = 0$ και $P[\mathbf{X_1^n} = \mathbf{x_1^n}] \log \frac{P[\mathbf{X_1^n} = \mathbf{x_1^n}]}{0} = \infty$

Να σημειωθεί ότι η σχετική εντροπία δεν αποτελεί μία μετρική με την αυστηρή έννοια της Πραγματικής Ανάλυσης, καθώς δεν ισχύει ούτε η συμμετρική ιδιότητα ούτε η τριγωνική ανισότητα, αλλά είναι διαισθητικά χρήσιμο να τη σκεφτόμαστε σαν την μέση απόσταση που υπάρχει στην πληροφορία που εμπεριέχεται μεταξύ δύο συναρτήσεων μάζας πιθανότητας. Στην τελευταία παράγραφο αυτό του κεφαλαίου θα αποδείξουμε κιόλας ότι $D(P||Q) \geq 0$ το οποίο πρακτικά σημαίνει πώς αν χρησιμοποιήσουμε οποιαδήποτε κατανομή Q πέρα της θεωρητικής P πάντα θα χρειαζόμαστε κατά μέσο όρο περισσότερη πληροφορία προκειμένου να περιγράψουμε την μεταβλητή ή το διάνυσμα των μεταβλητών μας με βάση την Q. Αν αναλύσουμε λίγο περισσότερο τον παραπάνω ορισμό μπορούμε να εξάγουμε ένα πολύ χρήσιμο μέγεθος που λέγεται διαγώνια εντροπία (cross entropy) $CE(P, Q)$.

$$D(P||Q) = \sum_{\mathbf{x_1^n} \in \times_{i=1}^{n} \mathcal{X}_i} P[\mathbf{X_1^n} = \mathbf{x_1^n}] \log \frac{P[\mathbf{X_1^n} = \mathbf{x_1^n}]}{Q[\mathbf{X_1^n} = \mathbf{x_1^n}]} =$$

$$\sum_{\mathbf{x_1^n} \in \times_{i=1}^{n} \mathcal{X}_i} P[\mathbf{X_1^n} = \mathbf{x_1^n}] \log P[\mathbf{X_1^n} = \mathbf{x_1^n}] - \sum_{\mathbf{x_1^n} \in \times_{i=1}^{n} \mathcal{X}_i} P[\mathbf{X_1^n} = \mathbf{x_1^n}] \log Q[\mathbf{X_1^n} = \mathbf{x_1^n}]$$

Άρα:

$$D(P||Q) = -H(P) + CE(P,Q) \Rightarrow D(P||Q) = CE(P,Q) - H(P) \Rightarrow$$

$$CE(P,Q) = -\sum_{\mathbf{x_1^n} \in \times_{i=1}^{n} \mathcal{X}_i} P[\mathbf{X_1^n} = \mathbf{x_1^n}] \log Q[\mathbf{X_1^n} = \mathbf{x_1^n}] \tag{1.12}$$

Η διαγώνια εντροπία ποσοτικοποιεί την πληροφορία που χρησιμοποιήσαμε για να περιγράψουμε μία τυχαία μεταβλητή ή ένα διάνυσμα τυχαίων μεταβλητών με βάση μία κατανομή Q ενώ στην πραγματικότητα η θεωρητική κατανομή της ήταν η P. Η διαγώνια εντροπία συνήθως χρησιμοποιείται για να αξιολογήσουμε πόσο καλά έχουν "εκπαιδευτεί" οι αλγόριθμοι ταξινόμησης κατά τη διαδικασία της επιβλεπομένης μάθησης. Με την εισαγωγή της έννοιας αυτής φαίνεται ακόμη καλύτερα η ουσία της απόστασης Kullback-Leibler, η οποία ποσοτικοποιεί την επιπρόσθετη πληροφορία που χρειαστήκαμε για να περιγράψουμε την τυχαία μεταβλητή με βάση την κατανομή Q, ενώ αρκούσε ποσότητα ίση με $H(P)$.

Για $n = 2$ η διαγώνια εντροπία γίνεται:

$$CE(P,Q) = -\sum_{x \in \mathcal{X}, y \in \mathcal{Y}} P[X = x, Y = x] \log Q[X = x, Y = y] \tag{1.13}$$

Παράδειγμα 1.7. *Έστω ότι έχουμε μία τυχαία μεταβλητή X που παίρνει τις τιμές $X = \{a, b, c, d\}$ και περιγράφεται από την θεωρητική κατανομή $P_X(x) = \{\frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \frac{1}{8}\}$. Έστω δύο πειραματικές κατανομές $Q_X^1(x) = \{\frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4}\}$ και $Q_X^2(x) = \{\frac{1}{4}, \frac{1}{4}, \frac{1}{8}, \frac{3}{8}\}$. Να βρείτε, χρησιμοποιώντας την απόσταση Kullback-Leibler ποια από τις δύο πειραματικές κατανομές περιγράφει καλύτερα την X. Έπειτα χρησιμοποιώντας την διαγώνια εντροπία να υπολογίσετε την συνολική πληροφορία που χρειαζόμαστε για να περιγράψουμε την μεταβλητή X βάση της βέλτιστης προσεγγιστικής κατανομής. Τέλος να υπολόγισετε την εντροπία της θεωρητικής κατανομής.*

$$X = \{a, b, c, d\} \quad P_X(x) = \{\frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \frac{1}{8}\} \quad Q_X^1(x) = \{\frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4}\}, Q_X^2(x) = \{\frac{1}{4}, \frac{1}{4}, \frac{1}{8}, \frac{3}{8}\}$$

Υπολογίζοντας τις αποστάσεις Kullback-Leibler έχουμε:

$$D(P||Q^1) = \sum_{x \in \mathcal{X}} Pr[X = x] \log \frac{Pr[X = x]}{Q[X = x]} = \frac{1}{2} \cdot \log 2 + \frac{1}{4} \cdot \log 1 + \frac{1}{8} \cdot \log \frac{1}{2} + \frac{1}{8} \cdot \log \frac{1}{2} \Rightarrow$$

$$D(P||Q^1) = 0.25$$

$$D(P||Q^2) = \sum_{x \in \mathcal{X}} Pr[X = x] \log \frac{Pr[X = x]}{Q[X = x]} = \frac{1}{2} \cdot \log 2 + \frac{1}{4} \cdot \log 1 + \frac{1}{8} \cdot \log \frac{1}{2} + \frac{1}{8} \cdot \log \frac{1}{3} \Rightarrow$$

$$D(P||Q) = 0.176879 \approx 0.18$$

Άρα επιλέγουμε ως καλύτερη προσεγγιστική κατανομή την δεύτερη. Υπολογίζοντας την διαγώνια εντροπία έχουμε:

$$CE(P,Q) = -\sum_{x \in \mathcal{X}} P[X = x] \log Q[X = x] = \frac{1}{2} \cdot \log 4 + \frac{1}{4} \cdot \log 4 + \frac{1}{8} \cdot \log 8 + \frac{1}{8} \cdot \log \frac{8}{3} \Rightarrow$$

$$CE(P,Q) = 2.051879 \approx 2.05$$

$$H(P) = -\sum_{x\in\mathcal{X}} P[X=x]\log P[X=x] = \frac{1}{2}\cdot\log 2 + \frac{1}{4}\cdot\log 4 + \frac{1}{8}\cdot\log 8 + \frac{1}{8}\cdot\log 8 \Rightarrow$$
$$H(P) = 1.75$$

Θεώρημα 1.3. ***(Κανόνας της αλυσίδας για την απόσταση Kullback-Leibler)*** *Έστω δύο σ.μ.π* $P[\mathbf{X_1^n} = \mathbf{x_1^n}]$ *και* $Q[\mathbf{X_1^n} = \mathbf{x_1^n}]$*, όπου* $X_1, X_2, ..., X_n$ *διακριτές τ.μ που παίρνουν τιμές στα πεπερασμένα σύνολα* $\mathcal{X}_1, \mathcal{X}_2, ..., \mathcal{X}_n$ *. Τότε:*

$$D(P[\mathbf{X_1^n} = \mathbf{x_1^n}]||Q[\mathbf{X_1^n} = \mathbf{x_1^n}]) = \sum_{i=1}^{n} D(P[X_i = x_i|\mathbf{X_1^{i-1}} = \mathbf{x_1^{i-1}}]||Q[X_i = x_i|\mathbf{X_1^{i-1}} = \mathbf{x_1^{i-1}}]) \tag{1.14}$$

Απόδειξη

$$D(P||Q) = \sum_{\mathbf{x_1^n}\in\times_{i=1}^{n}\mathcal{X}_i} P[\mathbf{X_1^n} = \mathbf{x_1^n}]\log\frac{P[\mathbf{X_1^n} = \mathbf{x_1^n}]}{Q[\mathbf{X_1^n} = \mathbf{x_1^n}]}$$

Από τον πολλαπλασιαστικό νομό των πιθανοτήτων γνωρίζουμε ότι:

$$P[\mathbf{X_1^n} = \mathbf{x_1^n}] = \prod_{i=1}^{n} P[X_i = x_i|\mathbf{X_1^{i-1}} = \mathbf{x_1^{i-1}}]$$

και

$$Q[\mathbf{X_1^n} = \mathbf{x_1^n}] = \prod_{i=1}^{n} Q[X_i = x_i|\mathbf{X_1^{i-1}} = \mathbf{x_1^{i-1}}]$$

Άρα:

$$D(P||Q) = \sum_{\mathbf{x_1^n}\in\times_{i=1}^{n}\mathcal{X}_i} P[\mathbf{X_1^n} = \mathbf{x_1^n}]\log\prod_{i=1}^{n}\frac{P[X_i = x_i|\mathbf{X_1^{i-1}} = \mathbf{x_1^{i-1}}]}{Q[X_i = x_i|\mathbf{X_1^{i-1}} = \mathbf{x_1^{i-1}}]} \Rightarrow$$
$$D(P||Q) = \sum_{\mathbf{x_1^n}\in\times_{i=1}^{n}\mathcal{X}_i} P[\mathbf{X_1^n} = \mathbf{x_1^n}]\sum_{i=1}^{n}\log\frac{P[X_i = x_i|\mathbf{X_1^{i-1}} = \mathbf{x_1^{i-1}}]}{Q[X_i = x_i|\mathbf{X_1^{i-1}} = \mathbf{x_1^{i-1}}]} \Rightarrow$$
$$D(P||Q) = \sum_{i=1}^{n}\sum_{\mathbf{x_1^n}\in\times_{i=1}^{n}\mathcal{X}_i} P[\mathbf{X_1^n} = \mathbf{x_1^n}]\log\frac{P[X_i = x_i|\mathbf{X_1^{i-1}} = \mathbf{x_1^{i-1}}]}{Q[X_i = x_i|\mathbf{X_1^{i-1}} = \mathbf{x_1^{i-1}}]} \Rightarrow$$
$$D(P[\mathbf{X_1^n} = \mathbf{x_1^n}]||Q[\mathbf{X_1^n} = \mathbf{x_1^n}]) = \sum_{i=1}^{n} D(P[X_i = x_i|\mathbf{X_1^{i-1}} = \mathbf{x_1^{i-1}}]||Q[X_i = x_i|\mathbf{X_1^{i-1}} = \mathbf{x_1^{i-1}}])$$

Η οικονομία στις πράξεις που εξασφαλίζουμε χρησιμοποιώντας τον κανόνα της αλυσίδας έχει ήδη συζητηθεί κατά τον κανόνα αλυσίδας για εξαρτημένες μεταβλητές όπου δείξαμε πως ο απευθείας υπολογισμός οποιασδήποτε συνάρτησης της από κοινού σ.μ.π n τυχαίων μεταβλητών που ορίζονται σε πεπερασμένα σύνολα μεγέθους n απαιτεί $\mathcal{O}(n^n)$ πράξεις ενώ ο υπολογισμός οποιασδήποτε συνάρτηση της δεσμευμένης σ.μ.π απαιτεί $\mathcal{O}(2^i)$ πράξεις όπου i το πλήθος των δεσμευμένων μεταβλητών. Στην απόσταση Kullback-Leibler επειδή έχουμε να κάνουμε με δύο κατανομές έπεται ότι το κόστος των πράξεων στον όρο $\sum_{i=1}^{n} D(P[X_i = x_i|\mathbf{X_1^{i-1}} = \mathbf{x_1^{i-1}}]||Q[X_i = x_i|\mathbf{X_1^{i-1}} = \mathbf{x_1^{i-1}}])$ θα κυριαρχείται από την κατανομή που περιέχει τις περισσότερες δεσμευμένες μεταβλητές.

1.5 Αμοιβαίο Μέτρο Πληροφορίας

Παράδειγμα 1.8. *Ας παίξουμε πάλι το παιχνίδι με τα βάζα αλλά με λίγο διαφορετικό τρόπο. Έστω ότι η εικόνα αυτή αποτελεί ένα στιγμιότυπο κάποιου παιχνιδιού του υπολογιστή μας. Το παιχνίδι σε κάθε παρτίδα διεξάγεται ως εξής:*

1. *Ο υπολογιστής επιλέγει τυχαία ένα από τα βάζα.*
2. *Μετά την επιλογή του βάζου μας δείχνει στην οθόνη πάντα ένα στοιχειό. Συνολικά τα στοιχεία είναι τρία και αποτελούνται από μία οριζόντια γραμμή, μία κάθετη γραμμή και ένα λουλούδι*
3. *Μετά τη προβολή του στοιχείου ο υπολογιστής μας ζητάει να μαντέψουμε ποιο βάζο επέλεξε. Αν δεν το βρούμε με δύο προβλέψεις χάνουμε.*

Καθώς εξελίσσονται οι παρτίδες στο παιχνίδι βλέπουμε πως όταν μας δείχνει το λουλούδι ο υπολογιστής έχει επιλέξει ένα από τα βάζα με το λουλούδι, όταν μας δείχνει την οριζόντια γραμμή έχει επιλέξει κάποιο πεπλατυσμένο βάζο και όταν μας δείχνει την κάθετη έχει επιλέξει κάποιο ψηλόλιγνο. Δηλαδή το στοιχείο που μας δίνει ο υπολογιστής μειώνει τη αβεβαιότητα ως προς την επιλογή που θα κάνουμε. Κατά πόσο τη μειώνει όμως; Έστω X η τυχαία μεταβλητή που μοντελοποιεί την επιλογή του βάζου από τον υπολογιστή. Έστω Y η τυχαία μεταβλητή που μοντελοποιεί την επιλογή του στοιχείου από τον υπολογιστή. Αν δεν υπήρχε η παράμετρος του στοιχείου, τότε η μέση αβεβαιότητα που θα είχαμε όταν μαντεύαμε κάποιο βάζο θα ήταν ίση με $H(X)$. Όταν μας δίνεται το στοιχείο τότε η μέση αβεβαιότητα είναι ίση $H(X|Y)$. Άρα το πλεονέκτημα που αποκτάμε κατά μέσο όρο θα είναι ίσο με $H(X) - H(X|Y)$. Όμως:

$$H(X) - H(Y|X) = \sum_{x\in\mathcal{X}} Pr[X=x] log \frac{1}{Pr[X=x]} - \sum_{x\in\mathcal{X}, y\in\mathcal{Y}} Pr[X=x, Y=y] log \frac{1}{Pr[X=x|Y=y]} =$$
$$\sum_{x\in\mathcal{X}} Pr[X=x] log \frac{1}{Pr[X=x]} + \sum_{x\in\mathcal{X}, y\in\mathcal{Y}} Pr[X=x, Y=y] log Pr[X=x|Y=y] =$$
$$\sum_{x\in\mathcal{X}, y\in\mathcal{Y}} Pr[X=x, Y=y] log \frac{Pr[X=x|Y=y]}{Pr[X=x]} = \sum_{x\in\mathcal{X}, y\in\mathcal{Y}} Pr[X=x, Y=y] log \frac{Pr[X=x, Y=y]}{Pr[X=x]Pr[Y=y]}$$

Η τελευταία σχέση που εξάγαμε λέγεται αμοιβαίο μέτρο πληροφορίας και ποσοτικοποιεί την κατά μέσο όρο μείωση της αβεβαιότητας για την τυχαία μεταβλητή X όταν είναι γνωστή η μεταβλητή Y

Ορισμός 1.5. *Έστω $X_1, ..., X_n$ διακριτές τ.μ που παίρνουν τιμές στα πεπερασμένα σύνολα $\mathcal{X}_1, ..., \mathcal{X}_n$. Ορίζουμε το* ***αμοιβαίο μέτρο πληροφορίας*** *ως:*

$$\boxed{I(X_1; ...; X_n) = \sum_{\mathbf{x_1^n}\in\times_{i=1}^n \mathcal{X}_i} P[\mathbf{X_1^n} = \mathbf{x_1^n}] \log \frac{P[\mathbf{X_1^n} = \mathbf{x_1^n}]}{\prod_{i=1}^n Pr[X_i = x_i]}} \tag{1.15}$$

Για δύο τυχαίες μεταβλητές τον παραπάνω μέτρο ορίζεται ως

$$I(X;Y) = \sum_{x\in\mathcal{X}, y\in\mathcal{Y}} Pr[X=x, Y=y] log \frac{Pr[X=x, Y=y]}{Pr[X=x]Pr[Y=y]} \tag{1.16}$$

Αν παρατηρήσουμε τον ορισμό της απόστασης Kullback-Leibler και του αμοιβαίου μέτρου πληροφορίας, θα δούμε ότι το τελευταίο μπορεί να εκφραστεί ως η "απόσταση" της πληροφορίας που περιέχεται στην από

κοινού κατανομής των τ.μ $(X_1....,X_n)$ από την κατανομή των $(X_1....,X_n)$ αν ήταν ανεξάρτητες. Για το λόγο αυτό το αμοιβαίο μέτρο πληροφορίας μπορεί να οριστεί και ώς:

$$I(X_1;...;X_n) = D(Pr[\mathbf{X_1^n} = \mathbf{x_1^n}]||(\prod_{i=1}^{n} Pr[X_i = x_i])) \tag{1.17}$$

Αν οι μεταβλητές ήταν όντως ανεξάρτητες τότε:

$$I(X_1;...;X_n) = \sum_{\mathbf{x_1^n} \in \times_{i=1}^{n} \mathcal{X}_i} P[\mathbf{X_1^n} = \mathbf{x_1^n}] \log \frac{P[\mathbf{X_1^n} = \mathbf{x_1^n}}{\prod_{i=1}^{n} Pr[X_i = x_i]} \Rightarrow$$

$$I(X_1;...;X_n) = \sum_{\mathbf{x_1^n} \in \times_{i=1}^{n} \mathcal{X}_i} P[\mathbf{X_1^n} = \mathbf{x_1^n}] \log \frac{\prod_{i=1}^{n} Pr[X_i = x_i]}{\prod_{i=1}^{n} Pr[X_i = x_i]} \Rightarrow$$

$$I(X_1;...;X_n) = \sum_{\mathbf{x_1^n} \in \times_{i=1}^{n} \mathcal{X}_i} P[\mathbf{X_1^n} = \mathbf{x_1^n}] \log(1) \Rightarrow$$

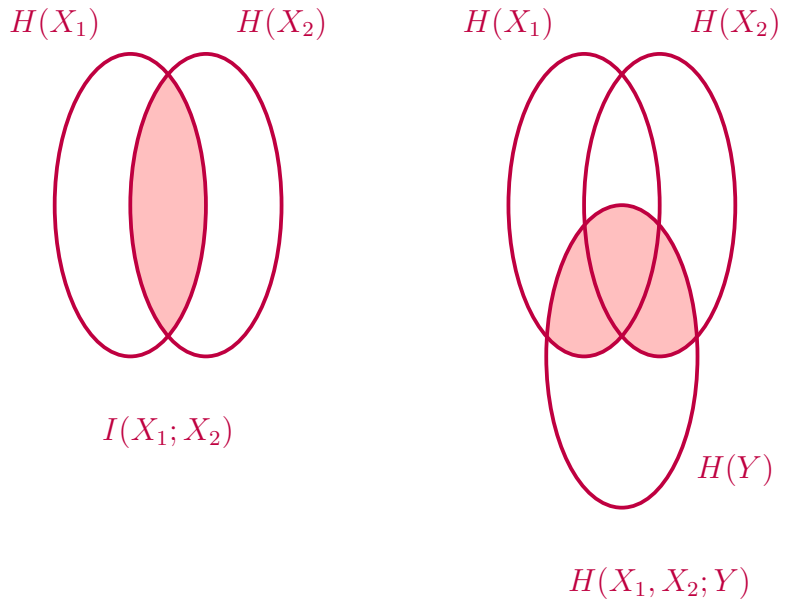
$$I(X_1;...;X_n) = 0$$

Το παραπάνω αποτέλεσμα είναι απόλυτα λογικό καθώς οι ανεξάρτητες μεταβλητές αποκλείεται να περιέχουν κοινές πληροφορίες. Αυτό γίνεται περισσότερο κατανοητό αν θυμηθούμε πως στην εισαγωγή το αμοιβαίο μέτρο πληροφορίας ορίστηκε σαν το μέσο πλεονέκτημα $H(X) - H(X|Y)$ που αποκτούσαμε στο παιχνίδι με τα βάζα. Αν το στοιχείο Υ που μας έδινε ο υπολογιστής ήταν ανεξάρτητο του X, δηλαδή δεν μας έδινε κάποιος μέρος της πληροφορίας που χρειαζόμαστε για να βρούμε το X, τότε το μέσο πλεονέκτημα μας θα ήταν μηδέν.

Ορισμός 1.6. *Έστω $X_1, ..., X_n, Y$ διακριτές τ.μ που παίρνουν τιμές στα πεπερασμένα σύνολα $\mathcal{X}_1, ..., \mathcal{X}_n, Y \in \mathcal{Y}$. Ορίζουμε το* ***αμοιβαίο μέτρο πληροφορίας*** *ως:*

$$\boxed{I(\mathbf{X_1^n} = \mathbf{x_1^n}; Y) = \sum_{\mathbf{x_1^n} \in \times_{i=1}^{n} \mathcal{X}_i} Pr[\mathbf{X_1^n} = \mathbf{x_1^n}, Y = y] \log \frac{Pr[\mathbf{X_1^n} = \mathbf{x_1^n}, Y = y]}{Pr[\mathbf{X_1^n} = \mathbf{x_1^n}] Pr[Y = y]}} \tag{1.18}$$

$$\boxed{= D(Pr[\mathbf{X_1^n} = \mathbf{x_1^n}, Y = y]||Pr[\mathbf{X_1^n} = \mathbf{x_1^n}] Pr[Y = y])}$$



Σχήμα 1.10: Η οπτικοποίηση των ορισμών 1.5 και 1. 6για $n = 2$

Ο παραπάνω ορισμός μας δίνει την κοινή πληροφορία που περιέχεται σε ένα διάνυσμα τυχαίων μεταβλητών $(X_1, ..., X_n)$ και σε μία τυχαία μεταβλητή Y.

Παράδειγμα 1.9. *Δίνεται ο παρακάτω πίνακας της από κοινού σ.μ.π των μεταβλητών X και Υ*

X/Υ	-1	0	1
-1	0	$\frac{2}{12}$	$\frac{3}{12}$
0	$\frac{1}{12}$	0	$\frac{3}{12}$
1	$\frac{3}{12}$	$\frac{2}{12}$	0

Βρείτε το αμοιβαίο μέτρο πληροφορίας $I(X;Y)$.

Λύση:

$$Pr[X=-1] = \sum_{y\in\mathcal{Y}} Pr[X=0, Y=y] = \frac{2}{12} + \frac{3}{12} = \frac{5}{12}$$

$$Pr[X=0] = \sum_{y\in\mathcal{Y}} Pr[X=0, Y=y] = \frac{1}{12} + \frac{3}{12} = \frac{4}{12}$$

$$Pr[X=1] = \sum_{y\in\mathcal{Y}} Pr[X=1, Y=y] = \frac{3}{12} + \frac{2}{12} = \frac{5}{12}$$

$$Pr[Y=-1] = \sum_{x\in\mathcal{X}} Pr[X=x, Y=-1] = \frac{1}{12} + \frac{3}{12} = \frac{4}{12}$$

$$Pr[Y=0] = \sum_{x\in\mathcal{X}} Pr[X=x, Y=0] = \frac{2}{12} + \frac{2}{12} = \frac{4}{12}$$

$$Pr[Y=1] = \sum_{x\in\mathcal{X}} Pr[X=x, Y=1] = \frac{3}{12} + \frac{3}{12} = \frac{6}{12}$$

$$I(X;Y) = \sum_{x\in\mathcal{X}, y\in\mathcal{Y}} Pr[X=x, Y=y] \log \frac{Pr[X=x, Y=y]}{Pr[X=x]\cdot Pr[Y=y]} =$$

$$0\cdot\log\frac{0}{\frac{5}{12}\cdot\frac{4}{12}} + \frac{2}{12}\cdot\log\frac{\frac{2}{12}}{\frac{5}{12}\cdot\frac{4}{12}} + \frac{3}{12}\cdot\log\frac{\frac{3}{12}}{\frac{5}{12}\cdot\frac{6}{12}} + \frac{1}{12}\cdot\log\frac{\frac{1}{12}}{\frac{4}{12}\cdot\frac{4}{12}} + 0\cdot\log\frac{0}{\frac{4}{12}\cdot\frac{4}{12}} + \frac{3}{12}\cdot\log\frac{\frac{3}{12}}{\frac{4}{12}\cdot\frac{6}{12}} +$$

$$\frac{3}{12}\cdot\log\frac{\frac{3}{12}}{\frac{5}{12}\cdot\frac{4}{12}} + \frac{2}{12}\cdot\log\frac{\frac{2}{12}}{\frac{5}{12}\cdot\frac{4}{12}} + 0\cdot\log\frac{0}{\frac{5}{12}\cdot\frac{6}{12}} =$$

$$\frac{2}{12}\cdot\log\frac{12}{10} + \frac{3}{12}\cdot\log\frac{12}{10} + \frac{1}{12}\cdot\log\frac{12}{16} + \frac{3}{12}\cdot\log\frac{12}{8} + \frac{3}{12}\cdot\log\frac{36}{20} + \frac{2}{12}\cdot\log\frac{12}{10} = 0.4770901 \approx 0.48$$

1.5.1 Ιδιότητες αμοιβαίου μέτρου πληροφορίας

Θεώρημα 1.4. *Έστω $X_1, ..., X_n$ διακριτές τ.μ ορισμένες πάνω στα πεπερασμένα σύνολα $\mathcal{X}_1, ..., \mathcal{X}_n$. Τότε:*

1. $I(X_1; ...; X_n) = \sum_{i=1}^{n} H(X_i) - H(\mathbf{X_1^n})$
2. $I(X_1; ...; X_n) = \sum_{i=1}^{n} H(X_i) - \sum_{i=1}^{n} H(X_i|\mathbf{X_1^{i-1}} = \mathbf{x_1^{i-1}})$

3. $I(X;X) = H(X)$
4. $I(X_1, \cdots, X_n; Y) = \sum_{i=1}^{n} H(X_i|\mathbf{X_1^{i-1}} = \mathbf{x_1^{i-1}}) - \sum_{i=1}^{n} H(X_i|\mathbf{X_1^{i-1}} = \mathbf{x_1^{i-1}}, Y)$

Απόδειξη

1.

$$I(X_1;...;X_n) = \sum_{\mathbf{x_1^n} \in \times_{i=1}^{n} \mathcal{X}_i} Pr[\mathbf{X_1^n} = \mathbf{x_1^n}] \log \frac{Pr[\mathbf{X_1^n} = \mathbf{x_1^n}]}{\prod_{i=1}^{n} Pr[X_i = x_i]} = \sum_{\mathbf{x_1^n} \in \times_{i=1}^{n} \mathcal{X}_i} Pr[\mathbf{X_1^n} = \mathbf{x_1^n}] \log Pr[\mathbf{X_1^n} = \mathbf{x_1^n}] -$$

$$\sum_{\mathbf{x_1^n} \in \times_{i=1}^{n} \mathcal{X}_i} Pr[\mathbf{X_1^n} = \mathbf{x_1^n}] \log \prod_{i=1}^{n} Pr[X_i = x_i] = -H(\mathbf{X_1^n}) - \sum_{\mathbf{x_1^n} \in \times_{i=1}^{n} \mathcal{X}_i} Pr[\mathbf{X_1^n} = \mathbf{x_1^n}] \sum_{i=1}^{n} \log \Pr[X_i = x_i] =$$

$$\sum_{i=1}^{n} \sum_{\mathbf{x_1^n} \in \times_{i=1}^{n} \mathcal{X}_i} Pr[\mathbf{X_1^n} = \mathbf{x_1^n}] \log \frac{1}{\Pr[X_i = x_i]} - H(\mathbf{X_1^n})$$

Παίρνωντας τις περιθώριες κατανομές, το εσωτερικό άθροισμα του πρώτου όρου γίνεται:

$$\sum_{i=1}^{n} \sum_{x_i \in \mathcal{X}_i} Pr[X_i = x_i] \log \frac{1}{\Pr[X_i = x_i]} - H(\mathbf{X_1^n}) \Rightarrow I(X_1;...;X_n) = \sum_{i=1}^{n} H(X_i) - H(\mathbf{X_1^n})$$

2. *Από τον πολλαπλασιαστικό νόμο των πιθανοτήτων έχουμε:*

$$Pr[\mathbf{X_1^n}] = \prod_{i=1}^{n} Pr[X_i = x_i|\mathbf{X_1^{i-1}} = \mathbf{x_1^{i-1}}]$$

Οπότε:

$$I(X_1;...;X_n) = \sum_{\mathbf{x_1^n} \in \times_{i=1}^{n} \mathcal{X}_i} Pr[\mathbf{X_1^n} = \mathbf{x_1^n}] \log \frac{Pr[[\mathbf{X_1^n} = \mathbf{x_1^n}]}{\prod_{i=1}^{n} Pr[X_i = x_i]} \Rightarrow$$

$$I(X_1;...;X_n) = \sum_{\mathbf{x_1^n} \in \times_{i=1}^{n} \mathcal{X}_i} Pr[\mathbf{X_1^n} = \mathbf{x_1^n}] \log \frac{\prod_{i=1}^{n} Pr[X_i = x_i|\mathbf{X_1^{i-1}} = \mathbf{x_1^{i-1}}]}{\prod_{i=1}^{n} Pr[X_i = x_i]} =$$

$$\sum_{\mathbf{x_1^n} \in \times_{i=1}^{n} \mathcal{X}_i} Pr[\mathbf{X_1^n} = \mathbf{x_1^n}] \log \prod_{i=1}^{n} Pr[X_i = x_i|\mathbf{X_1^{i-1}} = \mathbf{x_1^{i-1}}] - \sum_{\mathbf{x_1^n} \in \times_{i=1}^{n} \mathcal{X}_i} Pr[\mathbf{X_1^n} = \mathbf{x_1^n}] \log \prod_{i=1}^{n} Pr[X_i = x_i] =$$

$$\sum_{i=1}^{n} \sum_{\mathbf{x_1^n} \in \times_{i=1}^{n} \mathcal{X}_i} Pr[\mathbf{X_1^n} = \mathbf{x_1^n}] \log Pr[X_i = x_i|\mathbf{X_1^{i-1}} = \mathbf{x_1^{i-1}}] - \sum_{i=1}^{n} \sum_{\mathbf{x_1^n} \in \times_{i=1}^{n} \mathcal{X}_i} Pr[\mathbf{X_1^n} = \mathbf{x_1^n}] \log Pr[X_i = x_i] =$$

$$\sum_{i=1}^{n} \sum_{\mathbf{x_1^n} \in \times_{i=1}^{n} \mathcal{X}_i} Pr[\mathbf{X_1^n} = \mathbf{x_1^n}] \log \frac{1}{Pr[X_i = x_i]} - \sum_{i=1}^{n} \sum_{\mathbf{x_1^n} \in \times_{i=1}^{n} \mathcal{X}_i} Pr[\mathbf{X_1^n} = \mathbf{x_1^n}] \log \frac{1}{Pr[X_i = x_i|\mathbf{X_1^{i-1}} = \mathbf{x_1^{i-1}}]}$$

Κάνοντας χρήση των περιθωρίων κατανομών στο εσωτερικό άθροισμα του πρώτου όρου θα έχουμε:

$$\sum_{i=1}^{n} \sum_{x_i \in \mathcal{X}_i} Pr[X_i = x_i] \log \frac{1}{Pr[X_i = x_i]} - \sum_{i=1}^{n} \sum_{\mathbf{x_1^n} \in \times_{i=1}^{n} \mathcal{X}_i} Pr[\mathbf{X_1^n} = \mathbf{x_1^n}] \log \frac{1}{Pr[X_i = x_i|\mathbf{X_1^{i-1}} = \mathbf{x_1^{i-1}}]} \Rightarrow$$

$$I(X_1; X_2; \cdots; X_n) = \sum_{i=1}^{n} H(X_i) - \sum_{i=1}^{n} H(X_i|\mathbf{X_1^{i-1}})$$

3. *Από τη δεύτερη ιδιότητα που αποδείξαμε για* $X_1 = X_2 = X$ *έχουμε* $I(X;X) = I(X_1;X_2) = H(X_2) - H(X_2|X_1) = H(X) - H(X|X) = H(X) - 0 = H(X)$

4.

$$I(\mathbf{X_1^n};Y) = \sum_{\mathbf{x_1^n}\in\times_{i=1}^n\mathcal{X}_i, y\in\mathcal{Y}} P[\mathbf{X_1^n}=\mathbf{x_1^n}, Y=y]\log\frac{P[\mathbf{X_1^n}=\mathbf{x_1^n}, Y=y]}{P[\mathbf{X_1^n}=\mathbf{x_1^n}]P[Y=y]} =$$

$$\sum_{\mathbf{x_1^n}\in\times_{i=1}^n\mathcal{X}_i, y\in\mathcal{Y}} P[\mathbf{X_1^n}=\mathbf{x_1^n}, Y=y]\log\frac{1}{P[\mathbf{X_1^n}=\mathbf{x_1^n}]} + \sum_{\mathbf{x_1^n}\in\times_{i=1}^n\mathcal{X}_i, y\in\mathcal{Y}} P[\mathbf{X_1^n}=\mathbf{x_1^n}, Y=y]\log\frac{P[\mathbf{X_1^n}=\mathbf{x_1^n}, Y=y]}{Pr[Y=y]}$$

Χρησιμοποιώντας πάλι την περιθώρια για τον πρώτο όρο το παραπάνω άθροισμα γράφεται ως

$$\sum_{\mathbf{x_1^n}\in\times_{i=1}^n\mathcal{X}_i}\sum_{y\in\mathcal{Y}} P[\mathbf{X_1^n}=\mathbf{x_1^n}, Y=y]\log\frac{1}{P[\mathbf{X_1^n}=\mathbf{x_1^n}]} + \sum_{\mathbf{x_1^n}\in\times_{i=1}^n\mathcal{X}_i, y\in\mathcal{Y}} P[\mathbf{X_1^n}=\mathbf{x_1^n}, Y=y]\log P[\mathbf{X_1^n}=\mathbf{x_1^n}|Y=y] =$$

$$\sum_{\mathbf{x_1^n}\in\times_{i=1}^n\mathcal{X}_i} P[\mathbf{X_1^n}=\mathbf{x_1^n}]\log\frac{1}{P[\mathbf{X_1^n}=\mathbf{x_1^n}]} + \sum_{\mathbf{x_1^n}\in\times_{i=1}^n\mathcal{X}_i, y\in\mathcal{Y}} P[\mathbf{X_1^n}=\mathbf{x_1^n}, Y=y]\log P[\mathbf{X_1^n}=\mathbf{x_1^n}|Y=y] \Rightarrow$$

$$I(\mathbf{X_1^n};Y) = H(\mathbf{X_1^n}) - H(\mathbf{X_1^n}|Y)$$

Από τον κανόνα της αλυσίδας ισχύει $H(X_1,...,X_n)=\sum_{i=1}^n H(X_i|X_1,...,X_{i-1})$ και από την Πρόταση 1 ισχύει $H(X_1,...,X_n|Y)=\sum_{i=1}^n H(X_i|X_1,...,X_{i-1},Y)$ άρα έπεται:

$$I(X_1,...,X_n;Y) = \sum_{i=1}^n H(X_i|X_1,...,X_{i-1}) - \sum_{i=1}^n H(X_i|X_1,...,X_{i-1},Y)$$

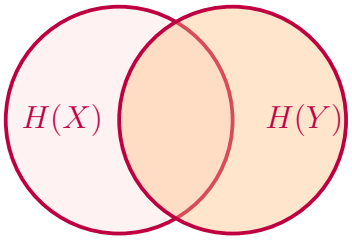
'

Σημείωση: Για $n=2$ οι παραπάνω ιδιότητες γίνονται:

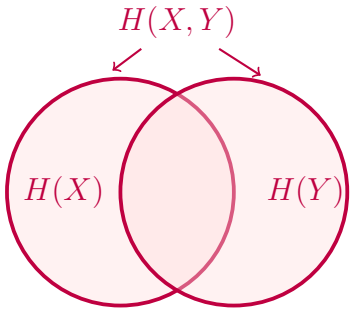
1. $I(X_1;X_2) = H(X_1) + H(X_2) - H(X_1,X_2)$
2. $I(X_1;X_2) = H(X_2) - H(X_2|X_1)$
3. $I(X_1;X_2) = H(X_1) - H(X_1|X_2)$
4. $I(X_1,X_2;Y) = H(X_1) + H(X_2|X_1) - H(X_1|Y) - HX_2|X_1,Y)$
5. Από τις ιδιότητες δύο και τρία έπεται $I(X_1;X_2) = I(X_2;X_1)$ (συμμετρική ιδιότητα)

Οι ιδιότητες 2 και 3 ισχύουν λόγω του πολλαπλασιαστικού νόμου των πιθανοτήτων $Pr[X_1=x_1, X_2=x_2] = Pr[X_1=x_1]Pr[X_2=x_2|X_1=x_1] = Pr[X_2=x_2]Pr[X_1=x_1|X_2=x_2]$. Για να γίνει και γραφικά αντιληπτή η σχέση των διάφορων εντροπιών με το αμοιβαίο μέτρο πληροφορίας παρουσιάζεται η παρακάτω απεικόνιση (εικόνα 1.7) μέσω διαγραμμάτων τύπου $Vienn$[10].

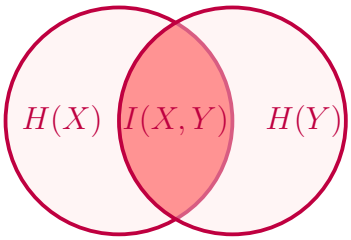
[10]Η κύρια διαφορά με τα διαγράμματα του Vienn είναι πώς όταν στην εντροπία γίνεται δέσμευση ο χώρος που δεσμεύεται παύει να μας απασχολεί ενώ στις πιθανότητες συμβαίνει το ακριβώς αντίθετο



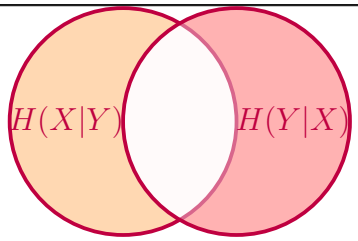
(α′) Απεικόνιση των εντροπιών δύο εξαρτημένων τυχαίων μεταβλητών μέσω διαγραμμάτων $Vienn$. Το ροζ χρώμα αντιστοιχεί στην $H(X)$ και το πορτοκαλί στην $H(Y)$.



(β′) Απεικόνιση της από κοινού εντροπίας. Η $H(X,Y)$ αναπαρίσταται από το ροζ χρώμα και εκφράζει την εντροπία που περιέχεται στις μεταβλητές X και Υ μαζί.



(γ′) Απεικόνιση τού αμοιβαίου μέτρο πληροφορίας. Το $I(X,Y)$ αναπαρίσταται από το κόκκινο χρώμα και εκφράζει την κοινή εντροπία που περιέχεται στις μεταβλητές X και Υ.



(δ′) Απεικόνιση των δεσμευμένων μέτρων πληροφορίας $H(X|Y)$ και $H(Y|X)$. Το πορτοκαλί χρώμα εκφράζει το ποσό της πληροφορίας που περιέχεται στην X όταν γνωρίζουμε την Υ, ενώ το ροζ εκφράζει το ποσό πληροφορίας που περιέχεται στην Υ όταν γνωρίζουμε την X.

Σχήμα 1.11: Απεικόνιση μέτρων πληροφορίας μέσω διαγραμμάτων $Vienn$

Από τα διαγράμματα βλέπουμε ότι το $H(X,Y)$ και $I(X,Y)$ χρησιμοποιούνται στην θεωρία της πληροφορίας όπως η ένωση και η τομή ενδεχομένων στη θεωρία πιθανοτήτων.

1.6 Δεσμευμένο αμοιβαίο μέτρο πληροφορίας

Στην ενότητα αυτή θα παρουσιάσουμε το δεσμευμένο αμοιβαίο μέτρο πληροφορίας. Στο παράδειγμα 8 είδαμε ότι το αμοιβαίο μέτρο πληροφορίας εκφράζει το κατά μέσο όρο πλεονέκτημα που έχουμε στις προβλέψεις μας όταν γνωρίζουμε κάποια μεταβλητή Y που σχετίζεται με την X. Θυμηθείτε ότι στο παράδειγμα ο υπολογιστής μας έδινε κάθε φορά ένα στοιχείο πριν ζητήσει την πρόβλεψη μας. Έστω τώρα ότι σε κάποιες παρτίδες έχουμε το δικαίωμα να ζητήσουμε ένα επιπρόσθετο στοιχείο. Τότε ο υπολογιστής μπορεί να εμφανίζει μία οριζόντια γραμμή σε τρία διαφορετικά ύψη. Το πρώτο ύψος βρίσκεται στην ψηλότερη θέση το δεύτερο σε κάποια μεσαία και το τρίτο στη χαμηλότερη. Προφανώς το ύψος που βρίσκεται η οριζόντια γραμμή αντιστοιχεί στο ύψος του βάζου. Πόσο πλεονέκτημα αποκτάω στις παρτίδες που έχω και δεύτερο στοιχείο; Όταν η παρτίδα έχει μόνο ένα στοιχείο η μέση αβεβαιότητα κατά την πρόβλεψη μας είναι $H(X|Y)$. Έστω ότι το δεύτερο στοιχείο μοντελοποιείται από την μεταβλητή Z. Τότε η μέση αβεβαιότητα στις παρτίδες που υπάρχει δεύτερο στοιχείο θα είναι ίση με $H(X|Y,Z)$. Άρα το πλεονέκτημα που αποκτάμε σε αυτές τις παρτίδες θα είναι $H(X|Y) - H(X|Y,Z)$

$H(X|Y) - H(X|Y,Z) =$

$$\sum_{x\in\mathcal{X},y\in\mathcal{Y}} Pr[X=x,Y=y]\log\frac{1}{Pr[X=x|Y=y]} - \sum_{\substack{x\in\mathcal{X},y\in\mathcal{Y},\\ z\in\mathcal{Z}}} Pr[X=x,Y=y,Z=z]\log\frac{1}{Pr[X=x|Y=y,Z=z]} =$$

$$\sum_{x\in\mathcal{X},y\in\mathcal{Y}} Pr[X=x,Y=y]\log\frac{1}{Pr[X=x|Y=y]} + \sum_{\substack{x\in\mathcal{X},y\in\mathcal{Y},\\ z\in\mathcal{Z}}} Pr[X=x,Y=y,Z=z]logPr[X=x|Y=y,Z=z] =$$

$$\sum_{\substack{x\in\mathcal{X},y\in\mathcal{Y},\\ z\in\mathcal{Z}}} Pr[X=x,Y=y,Z=z]\log\frac{Pr[X=x|Y=y,Z=z]}{Pr[X=x|Y=y]} =$$

$$\sum_{\substack{x\in\mathcal{X},y\in\mathcal{Y},\\ z\in\mathcal{Z}}} Pr[X=x,Y=y,Z=z]\log\frac{Pr[X=x,Y=y,Z=z]}{Pr[X=x|Y=y]Pr[Y=y,Z=z]} =$$

$$\sum_{\substack{x\in\mathcal{X},y\in\mathcal{Y},\\ z\in\mathcal{Z}}} Pr[X=x,Y=y,Z=z]\log\frac{Pr[X=x,Z=y|Y=z]Pr[Y=z]}{Pr[X=x|Y=y]Pr[Z=z|Y=y]Pr[Y=y]} =$$

$$\sum_{\substack{x\in\mathcal{X},y\in\mathcal{Y},\\ z\in\mathcal{Z}}} Pr[X=x,Y=y,Z=z]\log\frac{Pr[X=x,Z=y|Y=z]}{Pr[X=x|Y=y]Pr[Z=z|Y=y]} = I(X;Y|Z)$$

Ορισμός 1.7. *Έστω $X_1,...,X_n,Z$ διακριτές τ.μ που παίρνουν τιμές στα πεπερασμένα σύνολα $\mathcal{X}_1,...,\mathcal{X}_n,\mathcal{Z}$. Ορίζουμε το* ***δεσμευμένο αμοιβαίο μέτρο πληροφορίας*** *ως:*

$$\boxed{I(X_1;...;X_n|Z) = \sum_{\substack{\mathbf{x_1^n}\in\times_{i=1}^n\mathcal{X}_i,\\ z\in\mathcal{Z}}} Pr[\mathbf{X_1^n}=\mathbf{x_1^n},Z=z]log\frac{Pr[\mathbf{X_1^n}=\mathbf{x_1^n}|Z=z]}{\prod_{i=1}^n Pr[X_i=x_i|Z]} = D(Pr(\mathbf{X_1^n}|Z)||\prod_{i=1}^n Pr[X_i|Z])} \tag{1.19}$$

Για $n=2$ η παραπάνω σχέση γίνεται:

$$I(X_1;X_2|Y) = \sum_{x_1\in\mathcal{X}_1,x_2\in\mathcal{X}_2,y\in\mathcal{Y}} Pr[X_1=x_1,X_2=x_2,Y=y]log\frac{Pr[X_1=x_1,X_2=x_2|Y=y]}{Pr[X_1=x_1|Y=y]Pr[X_2=x_2|Y=y]} =$$
$$D(Pr[X_1=x_1,X_2=x_2|Y=y]||Pr[X_1=x_1|Y=y]Pr[X_2=x_2|Y=y]) \tag{1.20}$$

Ορισμός 1.8. *Έστω $X_1,...,X_n,Y,Z$ διακριτές τ.μ που παίρνουν τιμές στα πεπερασμένα σύνολα $\mathcal{X}_1,...,\mathcal{X}_n,\mathcal{Y},\mathcal{Z}$. Ορίζουμε* ***δεσμευμένο αμοιβαίο μέτρο πληροφορίας*** *ως:*

$$\boxed{I(\mathbf{X_1^n};Y|Z) = \sum_{\substack{\mathbf{x_1^n}\in\times_{i=1}^n\mathcal{X}_i,\\ y\in\mathcal{Y},z\in\mathcal{Z}}} Pr[\mathbf{X_1^n}=\mathbf{x_1^n},Y=y,Z=z]\cdot\log\frac{Pr[\mathbf{X_1^n}=\mathbf{x_1^n},Y=y|Z=z]}{Pr[\mathbf{X_1^n}=\mathbf{x_1^n}|Z=z]Pr[Y=y|Z=z]} =} \tag{1.21}$$
$$\boxed{D(Pr(\mathbf{X_1^n},Y|Z)||Pr(\mathbf{X_1^n}|Z)Pr(Y|Z))}$$

Ο ορισμός 7 ποσοτικοποιεί την εξάρτηση που μπορεί να υπάρχει μεταξύ των μεταβλητών $\{X_i\}_{i=1}^n$ όταν είναι γνωστό το Z. Αν όλες οι μεταβλητές $\{X_i\}_{i=1}^n$ είναι ανεξάρτητες δεδομένου της Z, τότε το $I(X_1;...;X_n|Z)=0$. Ο ορισμός 8 ποσοτικοποιεί την εξάρτηση που μπορεί να υπάρχει ανάμεσα σε ένα διάνυσμα τυχαίων μεταβλητών $(X_1,...,X_n)$ και την μεταβλητή Y δεδομένου της Z. Η απεικόνιση των παραπάνω ποσοτήτων φαίνεται καλύτερα στο Σχήμα 1.12

Θεώρημα 1.5. *Έστω $X_1, ..., X_n, Y, Z$ διακριτές τ.μ που παίρνουν τιμές στα πεπερασμένα σύνολα $\mathcal{X}_1, ..., \mathcal{X}_n, \mathcal{Y}, \mathcal{Z}$. Τότε:*

1. $I(X_1; ...; X_n|Z) = \sum_{i=1}^{n} H(X_i|Z) - \sum_{i=1}^{n} H(X_i|\mathbf{X_1^{i-1}}, Z)$
2. $I(\mathbf{X_1^n}; Y|Z) = H(Y|Z) - H(Y|\mathbf{X_1^n}, Z)$
3. $I(\mathbf{X_1^n}; Y) = \sum_{i=1}^{n} I(X_i; Y|\mathbf{X_1^{i-1}})$ *(κανόνα της αλυσίδας για το αμοιβαίο μέτρο πληροφορίας)*

Απόδειξη

1. $$I(X_1; ...; X_n|Z) = \sum_{\substack{\mathbf{x_1^n} \in \times_{i=1}^{n} \mathcal{X}_i, \\ z \in \mathcal{Z}}} Pr[\mathbf{X_1^n} = \mathbf{x_1^n}, Z = z] log \frac{Pr[\mathbf{X_1^n} = \mathbf{x_1^n}|Z = z]}{\prod_{i=1}^{n} Pr[X_i = X_i|Z = z]}$$

   Από τις ιδιότητες των λογαρίθμων και τον ορισμός της περιθώριας όπου:

   $$Pr[X_i = x_i, Z = z] = \sum_{\substack{\mathbf{x_1^n} \in \times_{j=1}^{n} \mathcal{X}_j \\ j \neq i}}, Pr[\mathbf{X_1^n} = \mathbf{x_1^n}, Z = z]$$

   έχουμε:

   $$I(X_1; ...; X_n|Z) = \sum_{\substack{\mathbf{x_1^n} \in \times_{i=1}^{n} \mathcal{X}_i, \\ z \in \mathcal{Z}}} Pr[\mathbf{X_1^n} = \mathbf{x_1^n}, Z = z] \log \frac{Pr[\mathbf{X_1^n} = \mathbf{x_1^n}|Z = z]}{\prod_{i=1}^{n} Pr[X_i = X_i|Z = z]} =$$

   $$\sum_{\substack{\mathbf{x_1^n} \in \times_{i=1}^{n} \mathcal{X}_i, \\ z \in \mathcal{Z}}} Pr[\mathbf{X_1^n} = \mathbf{x_1^n}, Z = z] \log Pr[\mathbf{X_1^n} = \mathbf{x_1^n}|Z = z] +$$

   $$\sum_{\substack{\mathbf{x_1^n} \in \times_{i=1}^{n} \mathcal{X}_i, \\ z \in \mathcal{Z}}} Pr[\mathbf{X_1^n} = \mathbf{x_1^n}, Z = z] \log \frac{1}{\prod_{i=1}^{n} Pr[X_i = X_i|Z = z]} \Rightarrow$$

   $$I(X_1; ...; X_n|Z) = -H(\mathbf{X_1^n}|Z) + \sum_{i=1}^{n} \sum_{\substack{\mathbf{x_1^n} \in \times_{i=1}^{n} \mathcal{X}_i, \\ z \in \mathcal{Z}}} Pr[\mathbf{X_1^n} = \mathbf{x_1^n}, Z = z] \log \frac{1}{Pr[X_i = X_i|Z = z]} \Rightarrow$$

   Χρησιμοποιώντας τις περιθώριες κατανομές το άθροισμα γίνεται

   $$I(X_1; ...; X_n|Z) = -H(\mathbf{X_1^n}|Z) + \sum_{i=1}^{n} \sum_{\substack{\mathbf{x_1^n} \in \times_{j=1}^{n} \mathcal{X}_j, \\ i \neq j}} Pr[\mathbf{X_1^n} = \mathbf{x_1^n}, Z = z] \log \frac{1}{Pr[X_i = X_i|Z = z]} \Rightarrow$$

   $$I(X_1; ...; X_n|Z) = -H(\mathbf{X_1^n}|Z) + \sum_{i=1}^{n} \sum_{x_i \in \mathcal{X}_i, z \in \mathcal{Z}} Pr[X_i = x_i, Z = z] \log \frac{1}{Pr[X_i = X_i|Z = z]} \Rightarrow$$

   $$I(X_1; ...; X_n|Z) = -H(\mathbf{X_1^n}|Z) + \sum_{i=1}^{n} H(X_i|Z)$$

   Από την πρόταση 1 ισχύει: $H(X_1, ..., X_n|Z) = \sum_{i=1}^{n} H(X_i|X_1, ..., X_{i-1}, Z)$

   Άρα:

   $$I(X_1; ...; X_n|Z) = \sum_{i=1}^{n} H(X_i|Z) - \sum_{i=1}^{n} H(X_i|X_1, ..., X_{i-1}, Z)$$

2.

$$I(\mathbf{X_1^n};Y|Z)=\sum_{\substack{\mathbf{x_1^n}\in\times_{i=1}^{n}\mathcal{X}_i,\\ y\in\mathcal{Y},z\in\mathcal{Z}}} Pr[\mathbf{X_1^n}=\mathbf{x_1^n},Y=y,Z=z]\log\frac{Pr[\mathbf{X_1^n}=\mathbf{x_1^n},Y=y|Z=z]}{Pr[\mathbf{X_1^n}=\mathbf{x_1^n}|Z=z]Pr[Y=y|Z=z]}$$

Από τις ιδιότητες των πιθανοτήτων ισχύει:

$$Pr[\mathbf{X_1^n}=\mathbf{x_1^n},Y=y|Z=z]=\frac{Pr[\mathbf{X_1^n}=\mathbf{x_1^n},Y=y,Z=z]}{Pr[Z=z]}=$$
$$\frac{Pr[Y=y|\mathbf{X_1^n}=\mathbf{x_1^n},Z=z]Pr[\mathbf{X_1^n}=\mathbf{x_1^n},Z=z]}{Pr[Z=z]}=$$
$$\frac{Pr[Y=y|\mathbf{X_1^n}=\mathbf{x_1^n},Z=z]Pr[\mathbf{X_1^n}=\mathbf{x_1^n}|Z=z]Pr[Z=z]}{Pr[Z=z]}=$$
$$Pr[Y=y|\mathbf{X_1^n}=\mathbf{x_1^n},Z=z]Pr[\mathbf{X_1^n}=\mathbf{x_1^n}|Z=z]$$

Από τις ιδιότητες των λογαρίθμων και το ορισμό των περιθωρίων έπεται:

$$I(\mathbf{X_1^n}=\mathbf{x_1^n};Y|Z)=\sum_{\substack{\mathbf{x_1^n}\in\times_{i=1}^{n}\mathcal{X}_i,\\ y\in\mathcal{Y},z\in\mathcal{Z}}} Pr[\mathbf{X_1^n}=\mathbf{x_1^n},Y=y,Z=z]\log\frac{Pr[\mathbf{X_1^n}=\mathbf{x_1^n},Y=y|Z=z]}{Pr[\mathbf{X_1^n}=\mathbf{x_1^n}|Z=z]Pr[Y=y|Z=z]}=$$
$$\sum_{\substack{\mathbf{x_1^n}\in\times_{i=1}^{n}\mathcal{X}_i,\\ y\in\mathcal{Y},z\in\mathcal{Z}}} Pr[\mathbf{X_1^n}=\mathbf{x_1^n},Y=y,Z=z]\log\frac{Pr[Y=y|\mathbf{X_1^n}=\mathbf{x_1^n},Z=z]Pr[\mathbf{X_1^n}=\mathbf{x_1^n}|Z=z]}{Pr[\mathbf{X_1^n}=\mathbf{x_1^n}|Z=z]Pr[Y=y|Z=z]}=$$
$$\sum_{\substack{\mathbf{x_1^n}\in\times_{i=1}^{n}\mathcal{X}_i,\\ y\in\mathcal{Y},z\in\mathcal{Z}}} Pr[\mathbf{X_1^n}=\mathbf{x_1^n},Y=y,Z=z]\log\frac{Pr[Y=y|\mathbf{X_1^n}=\mathbf{x_1^n},Z=z]}{Pr[Y=y|Z=z]}=$$
$$\sum_{\substack{\mathbf{x_1^n}\in\times_{i=1}^{n}\mathcal{X}_i,\\ y\in\mathcal{Y},z\in\mathcal{Z}}} Pr[\mathbf{X_1^n}=\mathbf{x_1^n},Y=y,Z=z]\log Pr[Y=y|\mathbf{X_1^n}=\mathbf{x_1^n},Z=z]+$$
$$\sum_{\substack{\mathbf{x_1^n}\in\times_{i=1}^{n}\mathcal{X}_i,\\ y\in\mathcal{Y},z\in\mathcal{Z}}} Pr[\mathbf{X_1^n}=\mathbf{x_1^n},Y=y,Z=z]\log\frac{1}{Pr[Y=y|Z=z]}\Rightarrow$$
$$I(X_1,...,X_n;Y|Z)=-H(Y|X_1,...,X_n,Z)+\sum_{y\in\mathcal{Y},z\in\mathcal{Z}} Pr[Y=y,Z=z]\log\frac{1}{Pr[Y=y|Z=z]}\Rightarrow$$
$$I(X_1,...,X_n;Y|Z)=H(Y|Z)-H(Y|X_1,...,X_n,Z)$$

3.

$$I(X_1,...,X_n;Y)=\sum_{\mathbf{x_1^n}\in\times_{i=1}^{n}\mathcal{X}_i y\in\mathcal{Y}} Pr[\mathbf{X_1^n}=\mathbf{x_1^n},Y=y]\log\frac{Pr[\mathbf{X_1^n}=\mathbf{x_1^n},Y=y]}{Pr[\mathbf{X_1^n}=\mathbf{x_1^n}]Pr[Y=y]}$$

Από τις ιδιότητες των λογαρίθμων και τον ορισμός της περιθώριας έχουμε:

$$I(X_1,...,X_n;Y) = \sum_{\mathbf{x_1^n}\in\times_{i=1}^n\mathcal{X}_i y\in\mathcal{Y}} Pr[\mathbf{X_1^n}=\mathbf{x_1^n},Y=y]\log\frac{Pr[\mathbf{X_1^n}=\mathbf{x_1^n},Y=y]}{Pr[\mathbf{X_1^n}=\mathbf{x_1^n}]Pr[Y=y]} =$$

$$\sum_{\mathbf{x_1^n}\in\times_{i=1}^n\mathcal{X}_i y\in\mathcal{Y}} Pr[\mathbf{X_1^n}=\mathbf{x_1^n},Y=y]\log\frac{1}{Pr[\mathbf{X_1^n}=\mathbf{x_1^n}]}+$$

$$\sum_{\mathbf{x_1^n}\in\times_{i=1}^n\mathcal{X}_i y\in\mathcal{Y}} Pr[\mathbf{X_1^n}=\mathbf{x_1^n},Y=y]\log\frac{Pr[\mathbf{X_1^n}=\mathbf{x_1^n},Y=y]}{Pr[Y=y]} \Rightarrow$$

$$I(X_1,...,X_n;Y) = H(X_1,...,X_n) + \sum_{\mathrm{vec}\, x_n\in\times_{i=1}^n\mathcal{X}_i y\in\mathcal{Y}} Pr[\mathbf{X_1^n}=\mathbf{x_1^n},Y=y] log\,\mathrm{Pr}[\mathbf{X_1^n}=\mathbf{x_1^n}|Y=y] \Rightarrow$$

$$I(X_1,...,X_n;Y) = H(X_1,...,X_n) - H(X_1,...,X_n|Y) \Rightarrow$$

Από κανόνα της αλυσίδας και την πρόταση 1 έχουμε:

$$I(X_1,...,X_n;Y) = H(X_1,...,X_n) - H(X_1,...,X_n|Y) = \sum_{i=1}^n H(X_i|X_1,...,X_{i-1}) - \sum_{i=1}^n H(X_i|X_1,...,X_{i-1},Y) =$$

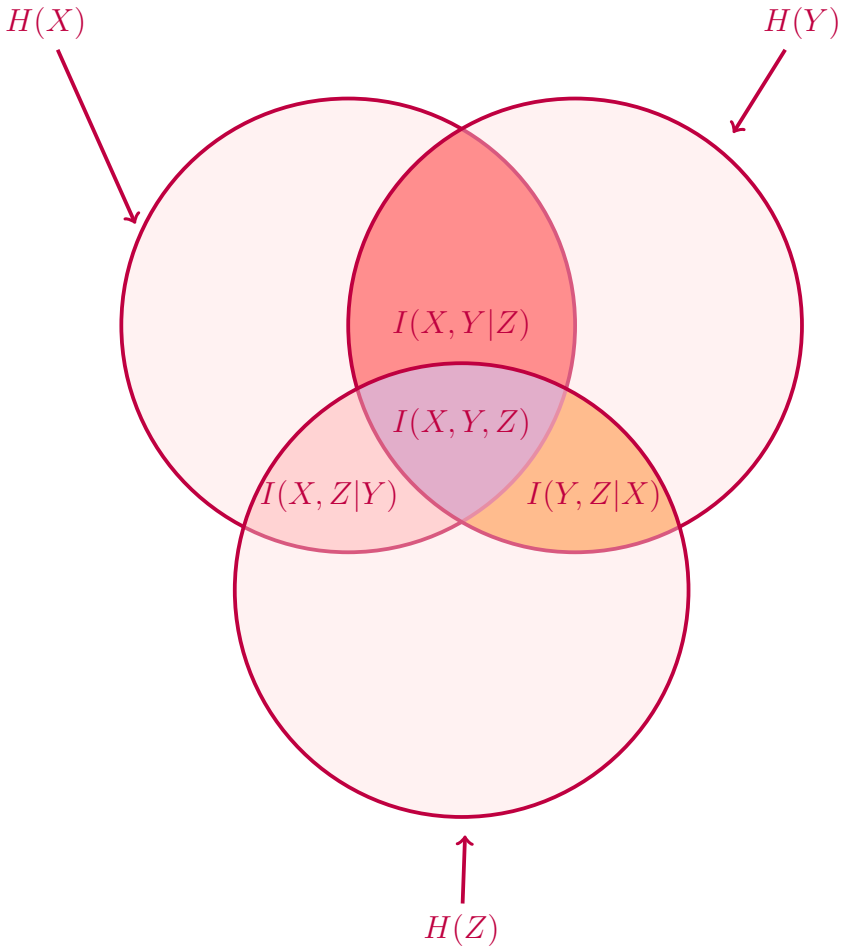
$$H(X_1) + H(X_2|X_1) + ... + H(X_n|X_1,...,X_{n-1})) - H(X_1|Y) - H(X_2|X_1,Y) - ... - H(X_n|X_1,...,X_{n-1},Y)$$

Μετά από ομαδοποίηση έπεται:

$$I(X_1,...,X_n;Y) =$$

$$(H(X_1) - H(X_1|Y)) + (H(X_2|X_1) - H(X_2|X_1,Y) + ... + (H(X_n|X_1,...,X_{n-1}) - H(X_n|X_1,...,X_{n-1},Y)) \Rightarrow$$

$$I(X_1,...,X_n;Y) = \sum_{i=1}^n I(X_i;Y|X_1,...,X_{i-1})$$



Σχήμα 1.12: Απεικόνιση των δεσμευμένων αμοιβαίων μέτρων πληροφορίας $I(X,Y|Z), I(Y,Z|X)$ και $I(X,Z|Y)$ καθώς και του $I(X,Y,Z)$. Το $I(X,Y|Z)$, αναπαριστάται με το κόκκινο χρώμα και εκφράζει το κοινό ποσό πληροφορίας που περιέχεται στις X,Υ όταν είναι γνωστή η Z. Τα $I(Y,Z|X)$ και $I(X,Z|Y)$ εκφράζουν τα αντίστοιχα ποσά πληροφορίας. Το $I(X,Y,Z)$ εκφράζει την κοινή εντροπία που υπάρχει και στις τρεις μεταβλητές μαζί

1.7 Άνω και κάτω φράγματα των μέτρων πληροφορίας

Όπως είδαμε στις παραπάνω παραγράφους η πρακτική αξία των μέτρων πληροφορίας είναι μεγάλη καθώς χρησιμοποιούνται για να αξιολογηθούν και να βελτιστοποιηθούν πιθανοκρατικά μοντέλα που κατασκευάζονται προς επίλυση μιας σειράς προβλημάτων. Η δυνατότητα αυτής της βελτιστοποίησης έγκειται στο γεγονός ότι τα μέτρα πληροφορίας που ορίστηκαν παραπάνω είναι επί της ουσίας κυρτές και κοίλες συναρτήσεις. Η ιδιότητα αυτή μας εξασφαλίζει την ύπαρξη ολικών μεγίστων και ελαχίστων. Πριν ξεκινήσουμε τη θεωρητική μελέτη των παραπάνω ιδιοτήτων θα εισάγουμε μία σειρά από ορισμούς και ανισότητες που θα βοηθήσουν στην απρόσκοπτη μελέτη της ενότητας αυτής.

1.7.1 Χρήσιμες γνώσεις

Ορισμός 1.9. *Έστω $\Omega \subseteq \mathbb{R}^n$. Τότε το Ω λέγεται κυρτό αν $\forall x, y \in \Omega$ και $\forall \lambda \in [0,1]$ ισχύει:*

$$\lambda x + (1-\lambda) y \in \Omega$$

Ορισμός 1.10. *Μία πραγματική συνάρτηση $f : \Omega \subseteq \mathbb{R}^n \to \mathbb{R}$ λέγεται κυρτή αν:*

1. *Το πεδίο ορισμού της συνάρτησης Ω είναι κυρτό σύνολο και*

2. $\forall x, y \in \Omega$ *και* $\forall \lambda \in [0,1]$ *ισχύει:*

$$f(\lambda x + (1-\lambda)y) \leq \lambda f(x) + (1-\lambda) f(y)$$

Η συνάρτηση f *θα λέγεται αυστηρώς κυρτή αν αντί για* $\leq$ *έχω μόνο* $<$ $\forall \lambda \in (0,1)$

Ορισμός 1.11. *Μία συνάρτηση* $f : \Omega \subseteq \mathbb{R}^n \to \mathbb{R}$ *θα λέγεται κοίλη αν η* $-f$ *είναι κυρτή*

Ορισμός 1.12. *Μία συνάρτηση* $f : \Omega \subseteq \mathbb{R}^n \to \mathbb{R}$ *θα λέγεται αυστηρώς κοίλη αν η* $-f$ *είναι αυστηρώς κυρτή*

Συμπεραίνουμε ότι αν η f είναι κοίλη τότε ισχύει η ανισοϊσότητα:

$$f(\lambda x + (1-\lambda)y) \geq \lambda f(x) + (1-\lambda) f(y)$$

Οι παραπάνω ορισμοί επεκτείνονται και για n σημεία. Θα διατυπωθεί μόνο ο ορισμός της κυρτότητας και οι υπόλοιποι είναι εύκολο να παραχθούν αντιστοίχως.

Ορισμός 1.13. *(**Επέκταση ορισμού κυρτότητας**)Μία πραγματική συνάρτηση* $f : \Omega \subseteq \mathbb{R}^n \to \mathbb{R}$ *λέγεται κυρτή αν:*

1. *Το πεδίο ορισμού της συνάρτησης* Ω *είναι κυρτό σύνολο και*

2. $\forall x_1, ..., x_n \in \Omega$ *και* $\forall \lambda_1, ..., \lambda_n \geq 0$ *με* $\sum_{i=1}^{n} \lambda_i = 1$ *ισχύει:*

$$f(\lambda_1 x_1 + ... + \lambda_n x_n) \leq \lambda_1 f(x_1) + \lambda_n f(x_n)$$

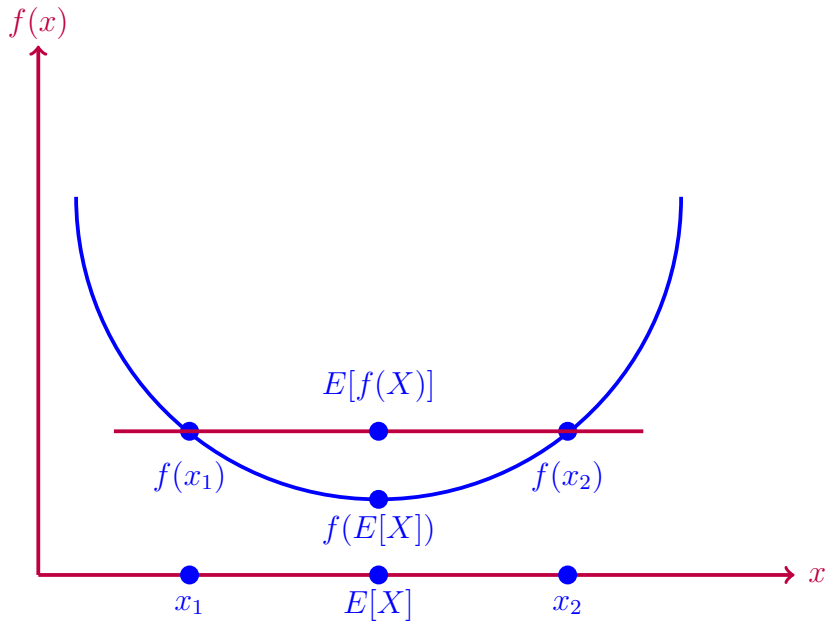
Η συνάρτηση f *θα λέγεται αυστηρώς κυρτή αν αντί για* $\leq$ *έχω μόνο* $<$ $\forall \lambda_i \in (0,1)$

Έστω τώρα ότι τα $x_1, ..., x_n$ αποτελούσαν τιμές μίας τυχαίας μεταβλητής X και οι αριθμοί $\lambda_1, ..., \lambda_n$ τις πιθανότητες να παρατηρήσουμε τις αντίστοιχες τιμές. Τότε η έκφραση $\lambda_1 x_1 + ... + \lambda_n x_n$ είναι επί της ουσίας η μέση τιμής της μεταβλητής X (E[X]), ενώ η έκφραση $\lambda_1 f(x_1) + \lambda_n f(x_n)$ αποτελεί την μέση τιμή των τιμών τυχαίας μεταβλητής $f(X)$, δηλαδή το $(E[f(X)])$. Τότε από τον ορισμό της κυρτότητας συμπεραίνουμε ότι $f(E[X]) \leq E[f(X)]$. Η παραπάνω ιδιότητα ονομάζεται ανισότητα Jensen

Θεώρημα 1.6. *(**Ανισότητα Jensen**) Αν η συνάρτηση* f *είναι κυρτή και* X *είναι μία τυχαία μεταβλητή τότε*

$$f(E[X]) \leq E[f(X)]$$

Επίσης αν η f *είναι αυστηρώς κυρτή τότε η ισότητα ισχύει αν X=E[X] σχεδόν βεβαίως, δηλαδή αν η τυχαία μεταβλητή* X *είναι σταθερή με πιθανότητα 1.*



Σχήμα 1.13: Η ανισότητα Jensen για μία δίτιμη $X = \{x_1, x_2\}$

1.7.2 Άνω και κάτω φράγμα της Εντροπίας μια τυχαίας μεταβλητής X

Θεώρημα 1.7. *Έστω $H(X)$ η εντροπία μίας διακριτής τυχαίας μεταβλητής ορισμένη πάνω σε ένα πεπερασμένο σύνολο $\mathcal{X}$ με*

$$H(X) = \sum_{x \in \mathcal{X}} Pr[X = x] log \frac{1}{Pr[X = x]}$$

. Τότε:

1. $H(X) \geq 0$
2. $H(X) = 0 \Leftrightarrow \exists k \in \mathbb{N} : Pr[X = x_k] = 1 \;\; και \;\; Pr[X = x_i] = 0 \;\; \forall x_i \neq x_k \;\; με \;\; x_i, x_k \in \mathcal{X}$
3. $H(X) \leq log|\mathcal{X}|$, *όπου* $|\mathcal{X}|$ *ο πληθάριθμος του συνόλου* $\mathcal{X}$

Απόδειξη

1. *Επειδή το* $Pr[.]$ *είναι μέτρο πιθανότητας, έπεται ότι:*

$$0 \leq Pr[X = x] \leq 1 \;\; \forall x \in \mathcal{X}$$

Άρα:

$$Pr[X = x] \leq 1 \;\; \forall x \in \mathcal{X} \Leftrightarrow \frac{1}{Pr[X = x]} \geq 1 \;\; \forall x \in \mathcal{X} \Leftrightarrow log \frac{1}{Pr[X = x]} \geq log 1 \Leftrightarrow log \frac{1}{Pr[X = x]} \geq 0 \;\; \forall x \in \mathcal{X}$$

Οπότε:

$$Pr[X = x] log \frac{1}{Pr[X = x]} \geq 0 \;\; \forall x \in \mathcal{X} \Leftrightarrow \sum_{x \in \mathcal{X}} Pr[X = x] \log \frac{1}{Pr[X = x]} \geq 0$$

2.

$$H(X) = 0 \Rightarrow \sum_{x\in\mathcal{X}} Pr[X = x] \log \frac{1}{Pr[X = x]} = 0$$

Το άθροισμα όμως μη αρνητικών αριθμών είναι μηδέν αν κάθε όρος του αθροίσματος είναι μηδέν, άρα:

$$\sum_{x\in\mathcal{X}} Pr[X = x] \log \frac{1}{Pr[X = x]} = 0 \Leftrightarrow Pr[X = x] \log \frac{1}{Pr[X = x]} = 0 \ \ \forall x \in \mathcal{X} \Leftrightarrow Pr[X = x] = 0$$

$$\eta^{'} \ \log \frac{1}{Pr[X = x]} = 0 \ \ \forall x \in \mathcal{X} \Leftrightarrow$$

$$Pr[X = x] = 0 \ \ \eta^{'} \ \ \frac{1}{Pr[X = x]} = 1 \Rightarrow Pr[X = x] = 1 \ \ \forall x \in \mathcal{X}$$

Επειδή όμως το $Pr[.]$ όμως είναι μέτρο πιθανότητας καταλαβαίνουμε ότι το $Pr[.]$ δεν γίνεται να παίρνει την τιμή μηδέν $\forall x \in \mathcal{X}$ ούτε την τιμή 1 $\forall x \in \mathcal{X}$. Άρα η τελευταία σχέση ικανοποιείται μόνο όταν υπάρχει κάποιο $x_k \in \mathcal{X}$ τέτοιο ώστε $Pr[X = x_k] = 1$ και άρα $Pr[X = x_i] = 0 \ \ \forall x_i \in \mathcal{X} \ \ με \ \ x_i \neq x_k$ επειδή πρέπει $\sum_{x\in\mathcal{X}} Pr[X = x] = 1$

3. Αρκεί να δείξω ότι:

$H(X) \leq \log |\mathcal{X}| \Leftrightarrow \sum_{x\in\mathcal{X}} Pr[X = x] \log \frac{1}{Pr[X = x]} - \log |\mathcal{X}| \leq 0$

$\sum_{x\in\mathcal{X}} Pr[X = x] \log \frac{1}{Pr[X = x]} - \log |\mathcal{X}| = \sum_{x\in\mathcal{X}} Pr[X = x] \log \frac{1}{Pr[X = x]} + \log \frac{1}{|\mathcal{X}|}$

Επειδή $\sum_{x\in\mathcal{X}} Pr[X = x] = 1$ έπεται:

$$\sum_{x\in\mathcal{X}} Pr[X = x] \log \frac{1}{Pr[X = x]} + \log \frac{1}{|\mathcal{X}|} = \sum_{x\in\mathcal{X}} Pr[X = x] \log \frac{1}{Pr[X = x]} + \sum_{x\in\mathcal{X}} Pr[X = x] \log \frac{1}{|\mathcal{X}|} =$$

$$\sum_{x\in\mathcal{X}} Pr[X = x] \log \frac{1}{Pr[X = x]|\mathcal{X}|} \leq \log e \sum_{x\in\mathcal{X}} Pr[X = x] \log \frac{1}{Pr[X = x]|\mathcal{X}|} =$$

$$\sum_{x\in\mathcal{X}} Pr[X = x] \ln \frac{1}{Pr[X = x]|\mathcal{X}|}$$

Γνωρίζουμε ότι $lnx \leq x - 1 \ \ \forall x > 0$, άρα:

$$\sum_{x\in\mathcal{X}} Pr[X = x] \ln \frac{1}{Pr[X = x]|\mathcal{X}|} \leq \sum_{x\in\mathcal{X}} Pr[X = x](\frac{1}{Pr[X = x]|\mathcal{X}|} - 1) = \sum_{x\in\mathcal{X}} \frac{1}{|\mathcal{X}|} - \sum_{x\in\mathcal{X}} Pr[X = x] = 1 - 1 = 0 \tag{1.22}$$

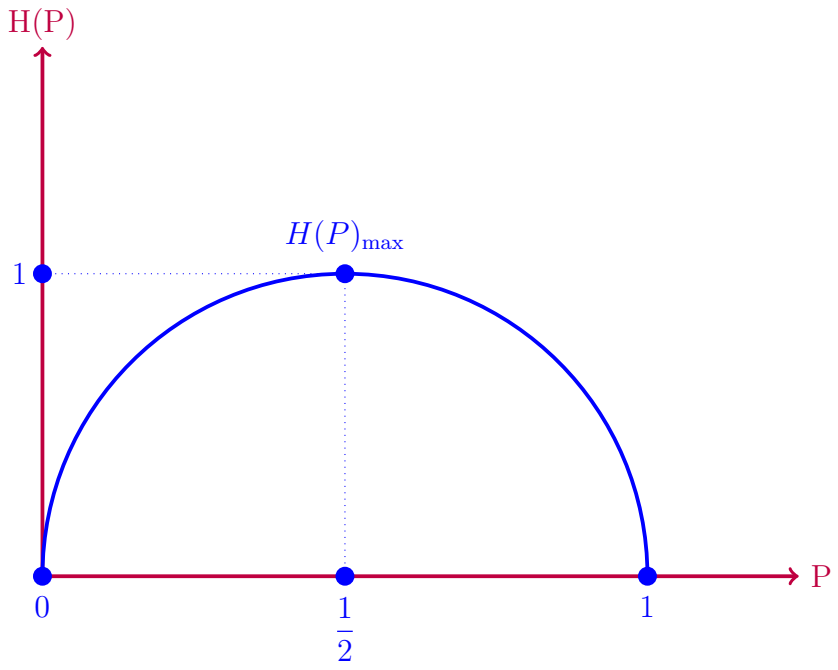
Άρα σύμφωνα με το παραπάνω θεώρημα, η εντροπία μιας τυχαίας μεταβλητής παίρνει ως ελάχιστη τιμή το μηδέν, όταν είναι ντετερμινιστική, δηλαδή όταν γνωρίζουμε εξαρχής το αποτέλεσμα της, οπότε δεν υπάρχει καμία αβεβαιότητα ώς προς αυτό. Επίσης παίρνει ώς μέγιστη τιμή το $log|\mathcal{X}|$, δηλαδή όταν η μεταβλητή αποτελείται από ισοπίθανα ενδεχόμενα οπότε και υπάρχει η μέγιστη αβεβαιότητα ως προς το αποτέλεσμα της αφού κάθε τιμή της επιλέγεται τυχαία. Το συγκεκριμένο αποτέλεσμα μπορεί να παρασταθεί και γραφικά αν μελετήσουμε την δυαδική εντροπία:

$H(P,\ 1-P) = H(P) = Pr[X=x]\log\frac{1}{Pr[X=x]} + (1-Pr[X=x])\log\frac{1}{1-Pr[X=x]} \overset{Pr[X=x]=P}{=}$
$P\log\frac{1}{P} + (1-P)\log\frac{1}{(1-P)}$.

Τότε η $H(P,\ 1-P) = 0 \Leftrightarrow P = 0$ ή $P = 1$. Επειδή η $H(P)$ είναι μία συνάρτηση συνεχής ως προς το P βρίσκουμε το μέγιστο:

$$\frac{dH(P)}{dP} = 0 \Rightarrow \frac{d(P\log\frac{1}{P} + (1-P)\log\frac{1}{(1-P)})}{dP} = 0 \Rightarrow \log\frac{1}{P} - 1 - \log\frac{1}{1-P} + 1 = 0 \Rightarrow$$
$$log\frac{1-P}{P} = 0 \Rightarrow \log\frac{1-P}{P} = \log 1 \Rightarrow \frac{1-P}{P} = 1 \Rightarrow 1-P = P \Rightarrow P = \frac{1}{2}$$

Άρα όπως διατυπώθηκε και από το προηγούμενο θεώρημα, η $H(P)$ παίρνει την μέγιστη τιμή της όταν τα ενδεχόμενα είναι ισοπίθανα.



Σχήμα 1.14: Το γράφημα της δυαδική εντροπίας $H(P) = H(P, 1-P)$

1.7.3 Άνω και κάτω φράγμα της από κοινού εντροπίας ενός διανύσματος τυχαίων μεταβλητών $\mathbf{X_1^n} = (\mathbf{X_1}, ..., \mathbf{X_n})$

Θεώρημα 1.8. *Έστω n διακριτές τ.μ $X_1, X_2, ...X_n$ ορισμένες στα πεπερασμένα σύνολα $\mathcal{X}_1, \mathcal{X}_2, ..., \mathcal{X}_n$ με από κοινού εντροπία:*

$$H(\mathbf{X_1^n}) = \sum_{\mathbf{x_1^n} \in \times_{i=1}^n \mathcal{X}_i} Pr[\mathbf{X_1^n}] log \frac{1}{Pr[\mathbf{X_1^n}]}$$

Τότε:

1. $H(\mathbf{X_1^n}) \geq 0$
2. $H(\mathbf{X_1^n}) = 0 \Leftrightarrow \exists (k_1, ..., k_n) \in \left(\mathbb{N}^*\right)^n : Pr[\mathbf{X_{k_1}^{k_n}} = \mathbf{x_{k_1}^{k_n}}] = 1$ *και* $Pr[\mathbf{X_{l_1}^{l_n}} = \mathbf{x_{l_1}^{l_n}}] = 0 \quad \forall (l_1, ..., l_n) \neq (k_1, ..., k_n)$
3. $H(\mathbf{X_1^n}) \leq \log \times_{i=1}^n |\mathcal{X}_i|$

Απόδειξη

1. *Επειδή το $Pr[.]$ είναι μέτρο πιθανότητας, ισχύει:*

$$0 \leq Pr[\mathbf{X_1^n} = \mathbf{x_1^n}] \leq 1 \quad \forall \mathbf{x_1^n} \in \times_{i=1}^{n} \mathcal{X}_i \Rightarrow$$
$$\log \frac{1}{Pr[\mathbf{X_1^n} = \mathbf{x_1^n}]} \geq \log 1 \quad \forall \mathbf{x_1^n} \in \times_{i=1}^{n} \mathcal{X}_i \Rightarrow$$
$$\log \frac{1}{Pr[\mathbf{X_1^n} = \mathbf{x_1^n}]} \geq 0 \quad \forall \mathbf{x_1^n} \in \times_{i=1}^{n} \mathcal{X}_i \Rightarrow$$
$$Pr[\mathbf{X_1^n} = \mathbf{x_1^n}] \log \frac{1}{Pr[\mathbf{X_1^n} = \mathbf{x_1^n}]} \geq 0 \quad \forall \mathbf{x_1^n} \in \times_{i=1}^{n} \mathcal{X}_i \Rightarrow$$
$$\sum_{\mathbf{x_1^n} \in \times_{i=1}^{n} \mathcal{X}_i} Pr[\mathbf{X_1^n} = \mathbf{x_1^n}] \log \frac{1}{Pr[\mathbf{X_1^n} = \mathbf{x_1^n}]} \geq 0$$

2. *Η απόδειξη είναι αντίστοιχη μέ αυτή του θεωρήματος 8-(2)*

3. *Από τον κανόνα της αλυσίδας για την από κοινού εντροπία γνωρίζουμε ότι:*

$$H(\mathbf{X_1^n}) \leq \sum_{i=1}^{n} H(X_i) \Rightarrow$$

Όμως από το θεώρημα 8-(3) γνωρίζουμε ότι $H(X_i) \leq \log |\mathcal{X}_i|$. Άρα:

$$H(\mathbf{X_1^n}) \leq \sum_{i=1}^{n} log|\mathcal{X}_i| \Rightarrow H(\mathbf{X_1^n}) \leq \log \times_{i=1}^{n} |\mathcal{X}_i|$$

Τα παραπάνω φράγματα γενικεύουν το θεώρημα 8. Οι ερμηνείες παραμένουν ίδιες, με τη μόνη διαφορά ότι πλέον δεν μιλάμε για μία τυχαία μεταβλητή, αλλά για ένα διάνυσμα τυχαίων μεταβλητών. Επίσης η ισότητα στο τρίτο φράγμα, ισχύει όταν οι μεταβλητές είναι ανεξάρτητες και κάθε τυχαία μεταβλητή αποτελείται από ισοπίθανα ενδεχόμενα καθώς τότε επιτυγχάνεται και η μέγιστη εντροπία, οπότε και η μέγιστη ποσότητα πληροφορίας που μπορεί να μας δώσει τον διάνυσμα $\operatorname{vec} X = (X_1, ..., X_n)$

1.7.4 Κάτω φράγμα απόστασης Kullback-Leibler-Διαγώνιας Εντροπίας

Θεώρημα 1.9. *Έστω ένα διάνυσμα τυχαίων μεταβλητών $\mathbf{X_1^n}$ με από κοινού μάζες πιθανότητας $P[\mathbf{X_1^n} = \mathbf{x_1^n}]$ και $Q[\mathbf{X_1^n} = \mathbf{x_1^n}]$. Τότε για την απόσταση Kullback-Leibler, ισχύει*

$$D(P||Q) = \sum_{\mathbf{x_1^n} \in \times_{i=1}^{n} \mathcal{X}_i} P[\mathbf{X_1^n} = \mathbf{x_1^n}] \log \frac{P[\mathbf{X_1^n} = \mathbf{x_1^n}]}{Q[\mathbf{X_1^n} = \mathbf{x_1^n}]} \geq 0 \tag{1.23}$$

Η ισότητα ισχύει μόνο αν $P(\mathbf{X_1^n} = \mathbf{x_1^n}) = Q(\mathbf{X_1^n} = \mathbf{x_1^n}), \quad \forall \mathbf{x_1^n} \in \times_{i=1}^{n} \mathcal{X}_i$

Απόδειξη

Θεωρώ το στήριγμα της μάζας πιθανότητας P, $\mathcal{A} = \{\mathbf{x_1^n} \in \times_{i=1}^{n} \mathcal{X}_i : P[\mathbf{X_1^n} = \mathbf{x_1^n}] > 0\}$

Τότε:

$$-D(P||Q) = -\sum_{\mathbf{x_1^n} \in \mathcal{A}} P[\mathbf{X_1^n} = \mathbf{x_1^n}] \log \frac{P[\mathbf{X_1^n} = \mathbf{x_1^n}]}{Q[\mathbf{X_1^n} = \mathbf{x_1^n}]} = \sum_{\mathbf{x_1^n} \in \mathcal{A}} P[\mathbf{X_1^n} = \mathbf{x_1^n}] \log \frac{Q[\mathbf{X_1^n} = \mathbf{x_1^n}]}{P[\mathbf{X_1^n} = \mathbf{x_1^n}]}$$

Επειδή η συνάρτηση $\log x$, είναι κοίλη, όπου $x = \dfrac{Q[\mathbf{X_1^n} = \mathbf{x_1^n}]}{P[\mathbf{X_1^n} = \mathbf{x_1^n}]}$, έπεται ότι:

$$E[f(X)] \leq f([E(X)]) \Leftrightarrow \sum_{\mathbf{x_1^n} \in \mathcal{A}} P[\mathbf{X_1^n} = \mathbf{x_1^n}] \log \frac{Q[\mathbf{X_1^n} = \mathbf{x_1^n}]}{P[\mathbf{X_1^n} = \mathbf{x_1^n}]} \leq \log \sum_{\mathbf{x_1^n} \in \mathcal{A}} P[\mathbf{X_1^n} = \mathbf{x_1^n}] \frac{Q[\mathbf{X_1^n} = \mathbf{x_1^n}]}{P[\mathbf{X_1^n} = \mathbf{x_1^n}]}$$

Άρα:

$$\sum_{\mathbf{x_1^n} \in \mathcal{A}} P[\mathbf{X_1^n} = \mathbf{x_1^n}] \log \frac{Q[\mathbf{X_1^n} = \mathbf{x_1^n}]}{P[\mathbf{X_1^n} = \mathbf{x_1^n}]} \leq \log \sum_{\mathbf{x_1^n} \in \mathcal{A}} P[\mathbf{X_1^n} = \mathbf{x_1^n}] \frac{Q[\mathbf{X_1^n} = \mathbf{x_1^n}]}{P[\mathbf{X_1^n} = \mathbf{x_1^n}]} = \log \sum_{\mathbf{x_1^n} \in \mathcal{A}} Q[\mathbf{X_1^n} = \mathbf{x_1^n}]$$

Επειδή όμως το μέτρο πιθανότητας είναι μονότονο, δηλαδή $A \subseteq B \Rightarrow Pr[A] \leq Pr[B]$,
έπεται:

$$\log \sum_{\mathbf{x_1^n} \in \mathcal{A}} Q[\mathbf{X_1^n} = \mathbf{x_1^n}] \leq \log \sum_{\mathbf{x_1^n} \in \times_{i=1}^n \mathcal{X}_i} Q[\mathbf{X_1^n} = \mathbf{x_1^n}] = \log 1 = 0$$

επειδή το $Q[\mathbf{X_1^n} = \mathbf{x_1^n}]$ είναι συνάρτηση μάζας πιθανότητας και άρα $\sum_{\mathbf{x_1^n} \in \times_{i=1}^n \mathcal{X}_i} Q[\mathbf{X_1^n} = \mathbf{x_1^n}] = 1$.
Συμπερασματικά:

$$-D(P||Q) \leq 0 \Leftrightarrow D(P||Q) \geq 0$$

Σημείωση Από την ανισότητα Jensen γνωρίζουμε ότι η ισότητα ισχύει όταν η τυχαία μεταβλητή X είναι σταθερή με πιθανότητα 1, δηλαδή $X = c \Leftrightarrow \frac{P[\mathbf{X_1^n} = \mathbf{x_1^n}]}{Q[\mathbf{X_1^n} = \mathbf{x_1^n}]} = c \Leftrightarrow P[\mathbf{X_1^n} = \mathbf{x_1^n}] = cQ[\mathbf{X_1^n} = \mathbf{x_1^n}] \Leftrightarrow \sum_{\mathbf{x_1^n} \in \times_{i=1}^n \mathcal{X}_i} P[\mathbf{X_1^n} = \mathbf{x_1^n}] = \sum_{\mathbf{x_1^n} \in \times_{i=1}^n \mathcal{X}_i} cQ[\mathbf{X_1^n} = \mathbf{x_1^n}] \Leftrightarrow c = 1 \Leftrightarrow P[\mathbf{X_1^n} = \mathbf{x_1^n}] = Q[\mathbf{X_1^n} = \mathbf{x_1^n}] \ \ \forall \mathbf{x_1^n} \in \times_{i=1}^n \mathcal{X}_i$

Η παραπάνω ιδιότητα είναι πολύ σημαντική καθώς όπως επισημάναμε και στην παράγραφο της απόστασης Kullback-Leibler μας λέει ότι αν δεν χρησιμοποιήσουμε την θεωρητική κατανομή P για να περιγράψουμε τα δεδομένα μας τότε το μέσο πλεόνασμα πληροφορίας θα είναι πάντα θετικό, δηλαδή πάντα θα έχουμε ζημία ως προς του πόρους που απαιτούνται για την περιγραφή των δεδομένων μας όταν χρησιμοποιούμε κάποια $Q \neq P$.

Θεώρημα 1.10. *Έστω ένα διάνυσμα τυχαίων μεταβλητών $\mathbf{X_1^n}$ με από κοινού μάζες πιθανότητας $P[\mathbf{X_1^n} = \mathbf{x_1^n}]$ και $Q[\mathbf{X_1^n} = \mathbf{x_1^n}]$. Τότε για την διαγώνια εντροπία ισχύει:*

$$CE(P, Q) \geq H(P) \tag{1.24}$$

Απόδειξη
Από την εξίσωση (1.14) γνωρίζουμε ότι:

$$D(P||Q) = CE(P, Q) - H(P)$$

Όμως στο προηγούμενο θεώρημα αποδείξαμε ότι $D(P||Q) \geq 0$
Οπότε:

$$D(P||Q) \geq 0 \Leftrightarrow CE(P, Q) - H(P) \geq 0 \Leftrightarrow CE(P, Q) \geq H(P)$$

Η ισότητα ισχύει όταν οι δύο κατανομές ταυτίζονται

Τα δύο παραπάνω θεωρήματα εμβαθύνουν καλύτερα την αντίληψη μας για την εντροπία ως τη βέλτιστη ποσότητα πληροφορίας που χρειαζόμαστε για να περιγράψουμε μία τυχαία μεταβλητή X που ακολουθεί ένα κανόνα $P[.]$. Από το θεώρημα 10 γίνεται καθαρό πως κάθε φορά που θα προσπαθούμε να εκφράσουμε μία τυχαία μεταβλητή με βάση οποιαδήποτε κατανομή Q που δεν είναι η πραγματική της, η επιπρόσθετη πληροφορία που θα χρειαζόμαστε κατά μέσο όρο θα είναι μη αρνητική. Αυτό πρακτικά σημαίνει ότι η μέση πληροφορία που θα χρειαστούμε $CE(P, Q)$ ώστε να περιγράψουμε τη επίμαχη μεταβλητή με βάση τη Q σίγουρα θα είναι μεγαλύτερη από την $H(P)$

Θεώρημα 1.11. *Έστω διάνυσμα τυχαίων μεταβλητών* $\mathbf{X_1^n}$ *με από κοινού μάζες πιθανότητας* $P[\mathbf{X_1^n} = \mathbf{x_1^n}]$, $Q[\mathbf{X_1^n} = \mathbf{x_1^n}]$ *και δεσμευμένες μάζες πιθανότητας* $P[\mathbf{X_1^n} = \mathbf{x_1^n}|\mathbf{X_1^{n-1}} = \mathbf{x_1^{n-1}}]$ *Τότε για την δεσμευμένη απόσταση Kullback-Leibler ισχύει:*

$$D(P[X_n|\mathbf{X_1^{n-1}}]||Q[X_n|\mathbf{X_1^{n-1}}]) \geq 0 \tag{1.25}$$

Απόδειξη

Θεωρώ το στήριγμα της μάζας πιθανότητας P, $\mathcal{A}_n = \{\mathbf{x_1^n} \in \times_{i=1}^n \mathcal{X}_i : P[[\mathbf{X_1^n} = \mathbf{x_1^n}] > 0\}$ *και το στήριγμα της δεσμευμένη πιθανότητας* $\mathcal{B} = \mathbf{x_1^n} \in \times_{i=1}^n \mathcal{X}_i : P[X_n = x_n|\mathbf{X_1^{n-1}} = \mathbf{x_1^{n-1}}] > 0$ *με* $\mathbf{x_1^{n-1}} \in \mathcal{A}_{n-1}\}$

Τότε:

$$\begin{aligned}
&- D(P[X_n|\mathbf{X_1^{n-1}}]||Q[X_n|\mathbf{X_1^{n-1}}]) = \\
&- \sum_{\mathbf{x_1^n} \in \mathcal{A}_n} P[\mathbf{X_1^n} = \mathbf{x_1^n}] \log \frac{P[X_n = X_n|\mathbf{X_1^{n-1}} = \mathbf{x_1^{n-1}}]}{Q[X_n = X_n|\mathbf{X_1^{n-1}} = \mathbf{x_1^{n-1}}]} = \\
&- \sum_{\mathbf{x_1^n} \in \mathcal{A}_n} P[X_n = x_n|\mathbf{X_1^{n-1}} = \mathbf{x_1^{n-1}}] Pr[\mathbf{X_1^{n-1}} = \mathbf{x_1^{n-1}}] \log \frac{P[X_n = X_n|\mathbf{X_1^{n-1}} = \mathbf{x_1^{n-1}}]}{Q[X_n = X_n|\mathbf{X_1^{n-1}} = \mathbf{x_1^{n-1}}]} = \\
&- \sum_{\mathbf{x_1^{n-1}} \in \mathcal{A}_{n-1}} P[\mathbf{X_1^{n-1}} = \mathbf{x_1^{n-1}}] \sum_{X_n \in \mathcal{B}} P[X_n = x_n|\mathbf{X_1^{n-1}} = \mathbf{x_1^{n-1}}] \log \frac{P[X_n = X_n|\mathbf{X_1^{n-1}} = \mathbf{x_1^{n-1}}]}{Q[X_n = X_n|\mathbf{X_1^{n-1}} = \mathbf{x_1^{n-1}}]} = \\
&\sum_{\mathbf{x_1^{n-1}} \in \mathcal{A}_{n-1}} P[\mathbf{X_1^{n-1}} = \mathbf{x_1^{n-1}}] \sum_{X_n \in \mathcal{B}} P[X_n = x_n|\mathbf{X_1^{n-1}} = \mathbf{x_1^{n-1}}] \log \frac{Q[X_n = X_n|\mathbf{X_1^{n-1}} = \mathbf{x_1^{n-1}}]}{P[X_n = X_n|\mathbf{X_1^{n-1}} = \mathbf{x_1^{n-1}}]}
\end{aligned}$$

Επειδή η $\log x$ *είναι κοίλη μπορώ να εφαρμόσω πάλι την ανισότητα Jensen με* $Z = \frac{Q[X_n = X_n|\mathbf{X_1^{n-1}} = \mathbf{x_1^{n-1}}]}{P[X_n = X_n|\mathbf{X_1^{n-1}} = \mathbf{x_1^{n-1}}]}$. *Οπότε*

$$\begin{aligned}
&- D(P[X_n|\mathbf{X_1^{n-1}}]||Q[X_n\mathbf{X_1^{n-1}}]) = \\
&\sum_{\mathbf{x_1^{n-1}} \in \mathcal{A}_{n-1}} P[\mathbf{X_1^{n-1}} = \mathbf{x_1^{n-1}}] \sum_{X_n \in \mathcal{B}} P[X_n = x_n|\mathbf{X_1^{n-1}} = \mathbf{x_1^{n-1}}] \log \frac{Q[X_n = X_n|\mathbf{X_1^{n-1}} = \mathbf{x_1^{n-1}}]}{P[X_n = X_n|\mathbf{X_1^{n-1}} = \mathbf{x_1^{n-1}}]} \leq \\
&\sum_{\mathbf{x_1^{n-1}} \in \mathcal{A}_{n-1}} P[\mathbf{X_1^{n-1}} = \mathbf{x_1^{n-1}}] \log \sum_{X_n \in \mathcal{B}} P[X_n = x_n|\mathbf{X_1^{n-1}} = \mathbf{x_1^{n-1}}] \frac{Q[X_n = X_n|\mathbf{X_1^{n-1}} = \mathbf{x_1^{n-1}}]}{P[X_n = X_n|\mathbf{X_1^{n-1}} = \mathbf{x_1^{n-1}}]} = \\
&\sum_{\mathbf{x_1^{n-1}} \in \mathcal{A}_{n-1}} P[\mathbf{X_1^{n-1}} = \mathbf{x_1^{n-1}}] \log \sum_{X_n \in \mathcal{B}} Q[X_n = X_n|\mathbf{X_1^{n-1}} = \mathbf{x_1^{n-1}}] \leq \\
&\sum_{\mathbf{x_1^{n-1}} \in \mathcal{A}_{n-1}} P[\mathbf{X_1^{n-1}} = \mathbf{x_1^{n-1}}] \log \sum_{X_n \in \mathcal{X}_n} Q[X_n = X_n|\mathbf{X_1^{n-1}} = \mathbf{x_1^{n-1}}]
\end{aligned}$$

Η τελευταία ανισότητα ισχύει λόγω της μονοτονίας του μέτρου πιθανότητας. Επειδή η $Q[X_n = X_n|\mathbf{X_1^{n-1}} = \mathbf{x_1^{n-1}}]$ *αποτελεί σ.μ.π έπεται* $\log \sum_{X_n \in \mathcal{X}_n} Q[X_n = X_n|\mathbf{X_1^{n-1}} = \mathbf{x_1^{n-1}}] = 1$, *άρα:*

$$\begin{aligned}
&- D(P[X_n|\mathbf{X_1^{n-1}}]||Q[X_n|\mathbf{X_1^{n-1}}]) \leq \\
&\sum_{\mathbf{x_1^{n-1}} \in \mathcal{A}_{n-1}} P[\mathbf{X_1^{n-1}} = \mathbf{x_1^{n-1}}] \log \sum_{X_n \in \mathcal{X}_n} Q[X_n = X_n|\mathbf{X_1^{n-1}} = \mathbf{x_1^{n-1}}] = \\
&\sum_{\mathbf{x_1^{n-1}} \in \mathcal{A}_{n-1}} P[\mathbf{X_1^{n-1}} = \mathbf{x_1^{n-1}}] \log 1 = 0 \Rightarrow \\
&- D(P[X_n|\mathbf{X_1^{n-1}}]||Q[X_n|\mathbf{X_1^{n-1}}]) \leq 0 \Leftrightarrow D(P[X_n|\mathbf{X_1^{n-1}}]||Q[X_n|\mathbf{X_1^{n-1}}]) \geq 0
\end{aligned}$$

Γνωρίζουμε ότι η ισότητα στην ανισότητα Jensen θα ισχύει αν η τυχαία μεταβλητή είναι σταθερή ,οπότε στην προκειμένη περίπτωση πρέπει $Z = \frac{Q[X_n = X_n|\mathbf{X_1^{n-1}} = \mathbf{x_1^{n-1}}]}{P[X_n = X_n|\mathbf{X_1^{n-1}} = \mathbf{x_1^{n-1}}]} = c \Leftrightarrow Q[X_n = X_n|\mathbf{X_1^{n-1}} = \mathbf{x_1^{n-1}}] = cP[X_n = X_n|\mathbf{X_1^{n-1}} = \mathbf{x_1^{n-1}}] \Leftrightarrow \sum_{x_n \in \mathcal{X}_n} Q[X_n = X_n|\mathbf{X_1^{n-1}} = \mathbf{x_1^{n-1}}] = c\sum_{x_n \in \mathcal{X}_n} P[X_n = X_n|\mathbf{X_1^{n-1}} = \mathbf{x_1^{n-1}}] \Leftrightarrow c = 1.$ *Άρα η δεσμευμένη σχετική εντροπία είναι 0 όταν* $P[X_n = X_n|\mathbf{X_1^{n-1}} = \mathbf{x_1^{n-1}}] = Q[X_n = X_n|\mathbf{X_1^{n-1}} = \mathbf{x_1^{n-1}}]\ \forall x_n \in \mathcal{X}_n,\ \forall \mathbf{x_1^{n-1}} \in \mathcal{A}_{n-1}$

1.7.5 Κάτω φράγμα του αμοιβαίου μέτρου πληροφορίας και του αντίστοιχου δεσμευμένου

Θεώρημα 1.12. *Έστω ένα διάνυσμα τυχαίων μεταβλητών* $\text{vec}\, X = (X_1, ..., X_n), Y$ *που παίρνουν τιμές στα πεπερασμένα σύνολα* $\mathcal{X}_1, ..., \mathcal{X}_n$ *αντίστοιχα, με από κοινού μάζα πιθανότητας* $Pr[\mathbf{X_1^n} = \mathbf{x_1^n}]$ *και με δεσμευμένη* $Pr[\mathbf{X_1^n} = \mathbf{x_1^n}|Y = y]$. *Τότε ισχύει:*

1. $I(X_1; ...; X_n) \geq 0$
2. $I(X_1; ...; X_n|Y) \geq 0$

Απόδειξη

Επειδή:

$$I(X_1; ...; X_n) = D(Pr[\mathbf{X_1^n} = \mathbf{x_1^n}]||\prod_{i=1}^{n} Pr[X_i = x_i])\ \kappa\alpha\iota$$

$$I(X_1; ...; X_n|Y) = D(Pr[\mathbf{X_1^n} = \mathbf{x_1^n}|Y = y]||\prod_{i=1}^{n} Pr[X_i = x_i|Y = y])$$

από τα θεωρήματα 12 και 11 έπεται το ζητούμενο.

Προσέχουμε πως στο παραπάνω θεώρημα την θέση της Q στην απόσταση Kullback-Leibler παίρνει η από κοινού σ.μ.π όταν έχουμε θεωρήσει είτε ότι οι τυχαίες μεταβλητές είναι ανεξάρτητες είτε ότι είναι ανεξάρτητες δεδομένου της Υ. Αυτό πρακτικά μας λέει πως αν οι μεταβλητές δεν είναι ανεξάρτητες και εμείς τις θεωρήσουμε ως τέτοιες τότε θα χρειαστούμε παραπάνω πληροφορίες για να τις περιγράψουμε.

1.7.6 Άνω και κάτω φράγμα του δεσμευμένου μέτρου πληροφορίας

Θεώρημα 1.13. *Έστω n διακριτές τ.μ* $X_1, X_2, ...X_n$ *ορισμένες σε πεπερασμένες σ-άλγεβρες* $\mathcal{X}_1, \mathcal{X}_2, ..., \mathcal{X}_n$ *με δεσμευμένη εντροπία:*

$$H(X_n|\mathbf{X_1^{n-1}}) = \sum_{\mathbf{x_1^n} \in \times_{i=1}^{n} \mathcal{X}_i} Pr[\mathbf{X_1^n} = \mathbf{x_1^n}] \log \frac{1}{Pr[X_n|\mathbf{X_1^{n-1}} = \mathbf{x_1^{n-1}}]} =$$

$$\sum_{\mathbf{x_1^{n-1}} \in \times_{i=1}^{n-1} \mathcal{X}_i} Pr[\mathbf{X_1^{n-1}} = \mathbf{x_1^{n-1}}] \sum_{x_n \in \mathcal{X}_n} Pr[X_n|\mathbf{X_1^{n-1}} = \mathbf{x_1^{n-1}}] \log \frac{1}{Pr[X_n|\mathbf{X_1^{n-1}} = \mathbf{x_1^{n-1}}]}$$

Τότε:

1. $H(X_n|\mathbf{X_1^{n-1}}) \geq 0$
2. $H(X_n||\mathbf{X_1^{n-1}}) = 0 \Leftrightarrow X_n = f(|\mathbf{X_1^{n-1}})$
3. $H(X_1) \geq H(X_1|X_2) \geq H(X_1||\mathbf{X_2^3}) \geq ... \geq H(X_1||\mathbf{X_2^n})$

Απόδειξη

1.

$$H(X_n||\mathbf{X_1^{n-1}}) = \sum_{\mathbf{x_1^n}\in\times_{i=1}^{n}\mathcal{X}_i} Pr[\mathbf{X_1^n} = \mathbf{x_1^n}] \log \frac{1}{Pr[\mathbf{X_1^{n-1}} = \mathbf{x_1^{n-1}}]} =$$

$$\sum_{\mathbf{x_1^{n-1}}\in\times_{i=1}^{n-1}\mathcal{X}_i} Pr[\mathbf{X_1^{n-1}} = \mathbf{x_1^{n-1}}] \sum_{x_n\in\mathcal{X}_n} Pr[X_n|\mathbf{X_1^{n-1}} = \mathbf{x_1^{n-1}}] \log \frac{1}{Pr[X_n|\mathbf{X_1^{n-1}} = \mathbf{x_1^{n-1}}]}$$

Επειδή το $Pr[*|*]$ είναι μέτρο πιθανότητας ισχύει:

$$0 \leq Pr[X_n = x_n|\mathbf{X_1^{n-1}} = \mathbf{x_1^{n-1}}] \leq 1 \ \forall x_n \in \mathcal{X}_n \Rightarrow$$

$$\frac{1}{Pr[X_n = x_n|\mathbf{X_1^{n-1}} = \mathbf{x_1^{n-1}}]} \geq 1 \ \forall x_n \in \mathcal{X}_n \Rightarrow$$

$$log\frac{1}{Pr[X_n = x_n|\mathbf{X_1^{n-1}} = \mathbf{x_1^{n-1}}]} \geq 0 \ \forall x_n \in \mathcal{X}_n \Rightarrow$$

$$Pr[X_n = x_n|\mathbf{X_1^{n-1}} = \mathbf{x_1^{n-1}}]log\frac{1}{Pr[X_n = x_n|\mathbf{X_1^{n-1}} = \mathbf{x_1^{n-1}}]} \geq 0 \ \forall x_n \in \mathcal{X}_n \Rightarrow$$

$$\sum_{x_n\in\mathcal{X}_n} Pr[X_n|\mathbf{X_1^{n-1}} = \mathbf{x_1^{n-1}}]log\frac{1}{Pr[X_n|\mathbf{X_1^{n-1}} = \mathbf{x_1^{n-1}}]} \geq 0$$

Επειδή το $Pr[*]$ είναι μέτρο πιθανότητας ισχύει:

$$Pr[\mathbf{X_1^{n-1}} = \mathbf{x_1^{n-1}}] \geq 0 \ \forall \mathbf{x_1^{n-1}} \in \times_{i=1}^{n-1}\mathcal{X}_i$$

2.

$$H(X_n|\mathbf{X_1^{n-1}} = \mathbf{x_1^{n-1}}) = 0 \Rightarrow$$

$$\sum_{\mathbf{x_1^{n-1}}\in\times_{i=1}^{n-1}\mathcal{X}_i} Pr[\mathbf{X_1^{n-1}} = \mathbf{x_1^{n-1}}] \sum_{x_n\in\mathcal{X}_n} Pr[X_n|\mathbf{X_1^{n-1}} = \mathbf{x_1^{n-1}}] \log \frac{1}{Pr[X_n|\mathbf{X_1^{n-1}} = \mathbf{x_1^{n-1}}]} = 0 \Rightarrow$$

Επειδή και τα δύο αθροίσματα αποτελούνται από μη αρνητικούς αριθμούς, η δεσμευμένη εντροπία θα είναι 0 όταν

$$Pr[\mathbf{X_1^{n-1}} = \mathbf{x_1^{n-1}}] = 0 \ \forall \mathbf{x_1^{n-1}} \in \times_{i=1}^{n-1}\mathcal{X}_i$$

ή

$$Pr[X_n|\mathbf{X_1^{n-1}} = \mathbf{x_1^{n-1}}] \log \frac{1}{Pr[X_n|\mathbf{X_1^{n-1}} = \mathbf{x_1^{n-1}}]} = 0 \ \forall x_n \in \mathcal{X}_n$$

Έστω

$$\mathcal{A} = \{\mathbf{x_1^{n-1}} \in \times_{i=1}^{n-1} : Pr[\mathbf{X_1^{n-1}} = \mathbf{x_1^{n-1}}] = 0\}$$

και

$$\mathcal{B} = \{\mathbf{x_1^{n-1}} \in \times_{i=1}^{n-1} : Pr[\mathbf{X_1^{n-1}} = \mathbf{x_1^{n-1}}] > 0\}$$

.

Τότε:

$$\mathcal{A} \cup \mathcal{B} = \times_{i=1}^{n-1}\mathcal{X}_i$$

Θα ορίσουμε μία συνάρτηση $f(*) : \times_{i=1}^{n-1}\mathcal{X}_i \to \mathcal{X}_n$. Παίρνουμε την περίπτωση όπου το $\mathbf{x_1^{n-1}} \in \mathcal{B}$. Τότε για να είναι η $H(X_n|Pr[\mathbf{X_1^{n-1}}]) = 0$ θα πρέπει αναγκαστικά να ισχύει:

$$Pr[X_n|\mathbf{X_1^{n-1}} = \mathbf{x_1^{n-1}}] \log \frac{1}{Pr[X_n|\mathbf{X_1^{n-1}} = \mathbf{x_1^{n-1}}]} = 0 \ \forall x_n \in \mathcal{X}_n$$

Επειδή όμως ισχύει:

$$\sum_{x_n \in \mathcal{X}_n} Pr[X_n|\mathbf{X_1^{n-1}} = \mathbf{x_1^{n-1}}] = 1$$

, η προηγούμενη εξίσωση θα έχει λύση μόνο αν υπάρχει ένα μοναδικό x_n τέτοιο ώστε $Pr[X_n|\mathbf{X_1^{n-1}} = \mathbf{x_1^{n-1}}] = 1$ για το συγκεκριμένο $\mathbf{x_1^{n-1}} \in \mathcal{B}$. Άρα σε αυτή την περίπτωση το $x_n = f(\mathbf{x_1^{n-1}})$. Με τον ίδιο τρόπο αντιστοιχίζουμε και τα υπόλοιπα x_n στα $\mathbf{x_1^{n-1}} \in \mathcal{B}$.

Τα x_n που περίσσεψαν από την προηγούμενη αντιστοίχιση τα αντιστοιχίζουμε μέσω της $f(*)$ με όποιον τρόπο θέλουμε στα $\mathbf{x_1^{n-1}} \in \mathcal{A}$. Βλέπουμε λοιπόν πώς όταν η δεσμευμένη εντροπία είναι 0 θα ισχύει $X_n = f(\mathbf{X_1^{n-1}})$.

3. *Για να δείξω ότι ισχύει η πρώτη ανισότητα αρκεί να δείξω ότι:*

$$H(X_1) - H(X_1|X_2) \geq 0$$

$$H(X_1) - H(X_1|X_2) =$$
$$\sum_{x_1 \in \mathcal{X}_1} Pr[X_1 = x_1] log \frac{1}{Pr[X_1 = x_1]} - \sum_{x_1 \in \mathcal{X}_1, x_2 \in \mathcal{X}_2} Pr[X_1 = x_1, X_2 = x_2] log \frac{1}{Pr[X_1 = x_1|X_2 = x_2]} =$$
$$\sum_{x_1 \in \mathcal{X}_1, x_2 \in \mathcal{X}_2} Pr[X_1 = x_1, X_2 = x_2] log \frac{Pr[X_1 = x_1|X_2 = x_2]}{Pr[X_1 = x_1]} =$$
$$\sum_{x_1 \in \mathcal{X}_1, x_2 \in \mathcal{X}_2} Pr[X_1 = x_1, X_2 = x_2] log \frac{Pr[X_1 = x_1, X_2 = x_2]}{Pr[X_1 = x_1, X_2 = x_2]} = I(X_1; X_2) \geq 0$$

Θα δείξουμε ότι ισχύει και η i-οστή ανισότητα $H(X_1|\mathbf{x_2^{i-1}}) \geq H(X_1|\mathbf{x_2^{i}})$.

$$H(X_1|\mathbf{X_2^{i-1}}) - H(X_1|\mathbf{X_2^i}) =$$
$$\sum_{\mathbf{x_1^{i-1}}\in\times_{j=1}^{i-1}\mathcal{X}_j} Pr[\mathbf{X_1^{i-1}} = \mathbf{x_1^{i-1}}] \log \frac{1}{Pr[X_1 = x_1|\mathbf{X_2^{i-1}} = \mathbf{x_2^{i-1}}]} -$$
$$\sum_{\mathbf{x_1^{i}}\in\times_{j=1}^{i}\mathcal{X}_j} Pr[\mathbf{X_1^{i}} = \mathbf{x_1^{i}}] \log \frac{1}{Pr[X_1 = x_1|\mathbf{X_2^{i}} = \mathbf{x_2^{i}}]} =$$
$$\sum_{\mathbf{x_1^{i}}\in\times_{j=1}^{i}\mathcal{X}_j} Pr[\mathbf{X_1^{i}} = \mathbf{x_1^{i}}] \log \frac{Pr[X_1 = x_1|\mathbf{X_2^{i}} = \mathbf{x_2^{i}}]}{Pr[X_1 = x_1|\mathbf{X_2^{i-1}} = \mathbf{x_2^{i-1}}]} =$$
$$\sum_{\mathbf{x_1^{i}}\in\times_{j=1}^{i}\mathcal{X}_j} Pr[\mathbf{X_1^{i}} = \mathbf{x_1^{i}}] \log \frac{Pr[\mathbf{X_1^{i}} = \mathbf{x_1^{i}}]}{Pr[X_1 = x_1|\mathbf{X_2^{i-1}} = \mathbf{x_2^{i-1}}]Pr[\mathbf{X_2^{i}} = \mathbf{x_2^{i}}]} =$$
$$\sum_{\mathbf{x_1^{i}}\in\times_{j=1}^{i}\mathcal{X}_j} Pr[\mathbf{X_1^{i}} = \mathbf{x_1^{i}}] \log \frac{Pr[X_1 = x_1, X_i = x_i|\mathbf{X_2^{i-1}} = \mathbf{x_2^{i-1}}]Pr[\mathbf{X_2^{i-1}} = \mathbf{x_2^{i-1}}]}{Pr[X_1 = x_1|\mathbf{X_2^{i-1}} = \mathbf{x_2^{i-1}}]Pr[\mathbf{X_2^{i}} = \mathbf{x_2^{i}}]} =$$
$$\sum_{\mathbf{x_1^{i}}\in\times_{j=1}^{i}\mathcal{X}_j} Pr[\mathbf{X_1^{i}} = \mathbf{x_1^{i}}] \log \frac{Pr[X_1 = x_1, X_i = x_i|\mathbf{X_2^{i-1}} = \mathbf{x_2^{i-1}}]Pr[\mathbf{X_2^{i-1}} = \mathbf{x_2^{i-1}}]}{Pr[X_1 = x_1|\mathbf{X_2^{i-1}} = \mathbf{x_2^{i-1}}]Pr[X_i = x_i|\mathbf{X_2^{i-1}} = \mathbf{x_2^{i-1}}]Pr[\mathbf{X_2^{i-1}} = \mathbf{x_2^{i-1}}]} =$$
$$\sum_{\mathbf{x_1^{i}}\in\times_{j=1}^{i}\mathcal{X}_j} Pr[\mathbf{X_1^{i}} = \mathbf{x_1^{i}}] \log \frac{Pr[X_1 = x_1, X_i = x_i|\mathbf{X_2^{i-1}} = \mathbf{x_2^{i-1}}]}{Pr[X_1 = x_1|\mathbf{X_2^{i-1}} = \mathbf{x_2^{i-1}}]Pr[X_i = x_i|\mathbf{X_2^{i-1}} = \mathbf{x_2^{i-1}}]} =$$
$$I(X_1; X_i|X_2, ..., X_{i-1}) \geq 0 \Rightarrow H(X_1|X_2, ..., X_{i-1}) \geq H(X_1|X_2, ..., X_i)$$

1.7.7 Κυρτότητα απόστασης Kullback-Leibler

Θεώρημα 1.14. *Η απόσταση Kullback-Leibler $D(P||Q)$ είναι μία κυρτή συνάρτηση ως προς τις σ.μ.π P και Q.*

Απόδειξη

Έστω δύο ζεύγη συναρτήσεων μάζας πιθανότητας (P_1, Q_1) και (P_2, Q_2), τότε και οι γραμμικοί συνδυασμοί $\lambda P_1 + (1-\lambda)P_2$ και $\lambda Q_1 + (1-\lambda)Q_2$ με $\lambda \in [0,1]$ θα αποτελούν επίσης σ.μ.π. Τότε το

$$D(\lambda P_1[\mathbf{X_1^n} = \mathbf{x_1^n}] + (1-\lambda)P_2[\mathbf{X_1^n} = \mathbf{x_1^n}]||\lambda Q_1[\mathbf{X_1^n} = \mathbf{x_1^n}] + (1-\lambda)Q_2[\mathbf{X_1^n} = \mathbf{x_1^n}]) =$$
$$\sum_{\mathbf{x_1^n}\in\times_{i=1}^{n}\mathcal{X}_i} \lambda P_1[\mathbf{X_1^n} = \mathbf{x_1^n}] + (1-\lambda)P_2[\mathbf{X_1^n} = \mathbf{x_1^n}] \log \frac{\lambda P_1[\mathbf{X_1^n} = \mathbf{x_1^n}] + (1-\lambda)P_2[\mathbf{X_1^n} = \mathbf{x_1^n}]}{\lambda Q_1[\mathbf{X_1^n} = \mathbf{x_1^n}] + (1-\lambda)Q_2[\mathbf{X_1^n} = \mathbf{x_1^n}]}$$

Για ένα συγκεκριμένο $\mathbf{x_1^n}$, θέτοντας $a_1(\mathbf{x_1^n}) = \lambda P_1[\mathbf{X_1^n} = \mathbf{x_1^n}]$, $a_2(\mathbf{x_1^n}) = (1-\lambda)P_2[[\mathbf{X_1^n} = \mathbf{x_1^n}]$, $b_1(\mathbf{x_1^n}) = \lambda Q_1[[\mathbf{X_1^n} = \mathbf{x_1^n}]$ και $b_2(\mathbf{x_1^n}) = (1-\lambda)Q_2[[\mathbf{X_1^n} = \mathbf{x_1^n}]$, τότε θα έχουμε

$$\lambda P_1[\mathbf{X_1^n} = \mathbf{x_1^n}] + (1-\lambda)P_2[\mathbf{X_1^n} = \mathbf{x_1^n}] \log \frac{\lambda P_1[\mathbf{X_1^n} = \mathbf{x_1^n}] + (1-\lambda)P_2[\mathbf{X_1^n} = \mathbf{x_1^n}]}{\lambda Q_1[\mathbf{X_1^n} = \mathbf{x_1^n}] + (1-\lambda)Q_2\mathbf{X_1^n} = \mathbf{x_1^n}]} =$$
$$(a_{1(\mathbf{x_1^n})} + a_{2(\mathbf{x_1^n})}) log \frac{a_{1(\mathbf{x_1^n})} + a_{2(\mathbf{x_1^n})}}{b_{1(\mathbf{x_1^n})} + b_{2(\mathbf{x_1^n})}} = \sum_{i=1}^{2} a_i(\mathbf{x_1^n}) \log \frac{\sum_{i=1}^{2} a_i(\mathbf{x_1^n})}{\sum_{i=1}^{2} b_i(\mathbf{x_1^n})}$$

Από την ανισότητα αθροίσματος λογαρίθμων $\sum_{i=1}^{n} a_i \log \frac{a_i}{b_i} \geq (\sum_{i=1}^{n} a_i) \log \frac{\sum_{i=1}^{n} a_i}{\sum_{i=1}^{n} b_i}$ όμως θα ισχύει

$$\lambda P_1[\mathbf{X_1^n} = \mathbf{x_1^n}] + (1-\lambda)P_2[\mathbf{X_1^n} = \mathbf{x_1^n}] \log \frac{\lambda P_1[\mathbf{X_1^n} = \mathbf{x_1^n}] + (1-\lambda)P_2[\mathbf{X_1^n} = \mathbf{x_1^n}]}{\lambda Q_1[\mathbf{X_1^n} = \mathbf{x_1^n}] + (1-\lambda)Q_2[\mathbf{X_1^n} = \mathbf{x_1^n}]} =$$

$$\sum_{i=1}^{2} a_i(\mathbf{x_1^n}) \log \frac{\sum_{i=1}^{2} a_i(\mathbf{x_1^n})}{\sum_{i=1}^{2} b_i(\mathbf{x_1^n}} \leq \sum_{i=1}^{n} a_i \log \frac{a_i}{b_i} = a_1 \log \frac{a_1}{b_1} + a_2 \log \frac{a_2}{b_2} =$$

$$\lambda P_1[\mathbf{X_1^n} = \mathbf{x_1^n}] \log \frac{\lambda P_1[\mathbf{X_1^n} = \mathbf{x_1^n}}{\lambda Q1[\mathbf{X_1^n} = \mathbf{x_1^n}]} + (1-\lambda)P_2[\mathbf{X_1^n} = \mathbf{x_1^n}] \log \frac{(1-\lambda)P_2[\mathbf{X_1^n} = \mathbf{x_1^n}]}{(1-\lambda)Q_2[\mathbf{X_1^n} = \mathbf{x_1^n}]} \Rightarrow$$

$$\lambda P_1[\mathbf{X_1^n} = \mathbf{x_1^n}] + (1-\lambda)P_2[\mathbf{X_1^n} = \mathbf{x_1^n}] \log \frac{\lambda P_1[\mathbf{X_1^n} = \mathbf{x_1^n}] + (1-\lambda)P_2[\mathbf{X_1^n} = \mathbf{x_1^n}]}{\lambda Q_1[\mathbf{X_1^n} = \mathbf{x_1^n}] + (1-\lambda)Q_2[\mathbf{X_1^n} = \mathbf{x_1^n}]} \leq$$

$$\lambda P_1[\mathbf{X_1^n} = \mathbf{x_1^n}] \log \frac{\lambda P_1[\mathbf{X_1^n} = \mathbf{x_1^n}]}{\lambda Q1[\mathbf{X_1^n} = \mathbf{x_1^n}]} + (1-\lambda)P_2[\mathbf{X_1^n} = \mathbf{x_1^n}] \log \frac{(1-\lambda)P_2[\mathbf{X_1^n} = \mathbf{x_1^n}]}{(1-\lambda)Q_2[\mathbf{X_1^n} = \mathbf{x_1^n}]} \; \forall \mathbf{x_1^n} \in \times_{i=1}^{n} \mathcal{X}_i \Rightarrow$$

$$\sum_{\mathbf{x_1^n} \in \times_{i=1}^{n} \mathcal{X}_i} \lambda P_1[\mathbf{X_1^n} = \mathbf{x_1^n}] + (1-\lambda)P_2[\mathbf{X_1^n} = \mathbf{x_1^n} \log \frac{\lambda P_1[\mathbf{X_1^n} = \mathbf{x_1^n}] + (1-\lambda)P_2[\mathbf{X_1^n} = \mathbf{x_1^n}]}{\lambda Q_1[\mathbf{X_1^n} = \mathbf{x_1^n}] + (1-\lambda)Q_2[\mathbf{X_1^n} = \mathbf{x_1^n}]} \leq$$

$$\sum_{\mathbf{x_1^n} \in \times_{i=1}^{n} \mathcal{X}_i} \lambda P_1[\mathbf{X_1^n} = \mathbf{x_1^n}] \log \frac{\lambda P_1[\mathbf{X_1^n} = \mathbf{x_1^n}]}{\lambda Q1[\mathbf{X_1^n} = \mathbf{x_1^n}]} + \sum_{\mathbf{x_1^n} \in \times_{i=1}^{n} \mathcal{X}_i} (1-\lambda)P_2[\mathbf{X_1^n} = \mathbf{x_1^n}] \log \frac{(1-\lambda)P_2[\mathbf{X_1^n} = \mathbf{x_1^n}]}{(1-\lambda)Q_2[\mathbf{X_1^n} = \mathbf{x_1^n}]} \Rightarrow$$

$$D(\lambda P_1[\mathbf{X_1^n} = \mathbf{x_1^n}] + (1-\lambda)P_2[\mathbf{X_1^n} = \mathbf{x_1^n}] || \lambda Q_1 \mathbf{X_1^n} = \mathbf{x_1^n}] + (1-\lambda)Q_2[\mathbf{X_1^n} = \mathbf{x_1^n}]) \leq$$

$$\lambda D(P_1[\mathbf{X_1^n} = \mathbf{x_1^n}] || Q_1[\mathbf{X_1^n} = \mathbf{x_1^n}]) + (1-\lambda) D(P_2[\mathbf{X_1^n} = \mathbf{x_1^n}] || Q_2[\mathbf{X_1^n} = \mathbf{x_1^n}]) \; \forall \lambda \in [0,1]$$

1.7.8 Κυρτότητα εντροπίας

Θεώρημα 1.15. *Η εντροπία είναι μία κοίλη συνάρτηση ως προς τη συνάρτηση μάζας πιθανότητας $Pr[*]$.*

Απόδειξη

Έστω δύο τυχαίες μεταβλητές X_1 και X_2 για τις οποίες ισχύει $X_1 \sim Pr_1[X = x]$ και $X_2 \sim Pr_2[X = x]$. Ορίζουμε την τυχαία μεταβλητή Θ, έτσι ώστε:

$$\Theta = \begin{cases} 1, & \mu\epsilon \;\; Pr[\theta = 1] = \lambda \\ 2, & \mu\epsilon \;\; Pr[\theta = 2] = 1 - \lambda \end{cases}$$

Έστω τυχαία μεταβλητή $Y = X_\theta$. Τότε η κατανομή της θα είναι $\lambda Pr_1[X = x] + (1-\lambda)Pr_2[X = x]$. Έστω Z τυχαία μεταβλητή με:

Από το θεώρημα 14 γνωρίζουμε ότι ισχύει $H(Y) \geq H(Y|\Theta)$, άρα

$H(Y) \geq H(Y|\Theta) \Rightarrow$

$$\sum_{y\in\mathcal{Y}}(\lambda Pr_1[X=x]+(1-\lambda)Pr_2[X=x])log\frac{1}{\lambda Pr_1[X=x]+(1-\lambda)Pr_2[X=x]} \geq$$

$$\sum_{y\in\mathcal{Y},\theta\in\Theta} Pr[Y=y,\Theta=\theta]log\frac{1}{Pr[Y=y|\Theta=\theta]} =$$

$$\sum_{\theta\in\Theta} Pr[\Theta=\theta]\sum_{y\in\mathcal{Y}} Pr[Y=y|\Theta=\theta]log\frac{1}{Pr[Y=y|\Theta=\theta]} =$$

$$Pr[\Theta=\theta_1]\sum_{y\in\mathcal{Y}} Pr[Y=y|\Theta=\theta_1]log\frac{1}{Pr[Y=y|\Theta=\theta_1]} + Pr[\Theta=\theta_2]\sum_{y\in\mathcal{Y}} Pr[Y=y|\Theta=\theta_2]log\frac{1}{Pr[Y=y|\Theta=\theta_2]} =$$

$$\lambda\sum_{x_1\in\mathcal{X}_1} Pr[X_1=x_1]log\frac{1}{Pr[X_1=x_1]} + (1-\lambda)\sum_{x_2\in\mathcal{X}_2} Pr[X_2=x_2]log\frac{1}{Pr[X_2=x_2]} = \lambda H(Pr_1)+(1-\lambda)H(Pr_2) \Rightarrow$$

$H(Y) \geq \lambda H(Pr_1)+(1-\lambda)H(Pr_2) \Rightarrow$

$H(\lambda Pr_1+(1-\lambda)Pr_2) \geq \lambda H(Pr_1)+(1-\lambda)H(Pr_2).$

Ανάλογα αποδεικνύεται ότι και η από κοινού εντροπία είναι κυρτή συνάρτηση.

1.7.9 Κυρτότητα αμοιβαίου μέτρου πληροφορίας

Θεώρημα 1.16. *Το αμοιβαίο μέτρο πληροφορίας είναι μια κοίλη συνάρτηση ώς προς τη συνάρτηση μάζας πιθανότητας $Pr[*]$.*

Απόδειξη

Έστω τυχαία μεταβλητή X που μπορεί να ακολουθεί τις κατανομές $Pr_1[X=x],...,Pr_n[X=x]$ και μία τυχαία μεταβλητή $Y \sim Pr[Y=y|X=x]$. Τότε το ζεύγος $(X,Y) \sim (Pr_i[X=x], Pr[Y=y|X=x])$ και έχει αντίστοιχο αμοιβαίο μέτρο πληροφορίας $I_i(X;Y)$, δηλαδή η X μπορεί να ακολουθεί κάποια από τις κατανομές Pr_i ενώ η δεσμευμένη κατανομή παραμένει ίδια.

Ορίζουμε την τυχαία μεταβλητή Θ, έτσι ώστε:

$$\Theta = \begin{cases} 1, & \mu\epsilon \;\; Pr[\theta=1]=\lambda_1 \\ 2, & \mu\epsilon \;\; Pr[\theta=2]=\lambda_2 \\ 3, & \mu\epsilon \;\; Pr[\theta=3]=\lambda_3 \\ \cdots & \\ n, & \mu\epsilon \;\; Pr[\theta=n]=\lambda_n \end{cases}$$

Επίσης ορίζουμε την τυχαία μεταβλητή $Z = X_\theta \sim Pr_\theta[X=x]$. Τότε το ζεύγος $(Z,Y) \sim (\sum_{i=1}^{n}\lambda_i Pr_i[X=x],\ Pr[Y=y|Z=z])$

Από τον κανόνα της αλυσίδας για το αμοιβαίο μέτρο πληροφορίας (Θεώρημα 5) όμως ξέρουμε ότι θα ισχύει :

$I(\Theta,Z;Y) = I(\Theta;Y)+I(Z;Y|\Theta) \geq I(Z;Y|\Theta)$ Η τελευταία ανισότητα ισχύει λόγω του θεωρήματος 13. , Άρα

$$I(\Theta,Z;Y) \geq I(Z;Y|\Theta) =$$
$$\lambda_1 I(Z;Y|\Theta=1)+...+\lambda_n I(Z;Y|\Theta=n) =$$
$$\lambda_1 I_1(X;Y)+...+\lambda_n I_n(X;Y)$$

Όμως το $I(\Theta,Z;Y) = I(Z;Y)+I(\Theta;Y|Z) = I(Z;Y)$. Η τελευταία ισότητα ισχύει διότι δεδομένου του Z τά Y,Θ είναι ανεξάρτητα. Άρα:

$$I(Z;Y) = I(\Theta, Z;Y) \geq I(Z;Y|\Theta) = \lambda_1 I_1(X;Y) + ... + \lambda_n I_n(X;Y)$$

Θεώρημα 1.17. *Το αμοιβαίο μέτρο πληροφορίας είναι μια κυρτή συνάρτηση ώς προς τη συνάρτηση μάζας πιθανότητας $Pr[*|*]$.*

Απόδειξη

Έστω τυχαία μεταβλητή X που ακολουθεί την κατανομές $Pr[X = x]$ και μία τυχαία μεταβλητή που μπορεί να ακολουθεί τις κατανομή $Pr_1[Y = y|X = x], ..., Pr_n[Y = y|X = x]$. Τότε το ζεύγος $(X, Y) \sim (Pr[X = x], Pr_i[Y = y|X = x])$ και έχει αντίστοιχο αμοιβαίο μέτρο πληροφορίας $I_i(X;Y)$, δηλαδή η Y μπορεί να ακολουθεί κάποια από τις κατανομές $Pr_i[Y = y|X = x]$ ενώ η κατανομή της X παραμένει ίδια.

$$\Theta = \begin{cases} 1, & \mu\epsilon \quad Pr[\theta = 1] = \lambda_1 \\ 2, & \mu\epsilon \quad Pr[\theta = 2] = \lambda_2 \\ 3, & \mu\epsilon \quad Pr[\theta = 3] = \lambda_3 \\ \cdots & \\ n, & \mu\epsilon \quad Pr[\theta = n] = \lambda_n \end{cases}$$

Επίσης ορίζουμε την τυχαία μεταβλητή $Z = Y \sim Pr_\theta[Y = y|X = x]$. Τότε το ζεύγος $(Z, Y) \sim (Pr[X = x],\ \sum_{i=1}^{n} \lambda_i Pr_i[Y = y|X = x])$

Από τον κανόνα της αλυσίδας για το αμοιβαίο μέτρο πληροφορίας (Θεώρημα 5) όμως ξέρουμε ότι θα ισχύει :

$I(\Theta, Z; X) = I(X; Z) + I(\Theta; X|Y) \geq I(Z; X)$ Η τελευταία ανισότητα ισχύει λόγω του θεωρήματος 13. , Άρα

$$\begin{aligned} &I(Z;X) \leq I(\Theta, Z;X) = I(\Theta;X) + I(Z;X|\Theta) = \\ &0 + \lambda_1 I(Z;X|\Theta = 1) + ... + \lambda_n I(Z;X|\Theta = n) = \\ &\lambda_1 I_1(X;Y) + ... + \lambda_n I_n(X;Y) \Rightarrow \\ &I(Z;X) \leq \lambda_1 I_1(X;Y) + ... + \lambda_n I_n(X;Y) \end{aligned}$$

Κεφάλαιο 2

Διακριτές Πήγες Πληροφορίας

2.1 Εισαγωγή

Ας εκτελέσουμε ένα πείραμα. Προσπαθήστε να διαβάσετε το παρακάτω κομμάτι κειμένου και να συμπληρώσετε τα κενά.

"*Στ_ν επ_κοιν_νία μ€ τους ά__ους ανθρώπους συ_νά πρ_σπ_θούμ_ να μεταδ_σουμ_ πλη__φορ_ες, ν_ ερ-μην_σουμ€ έ__ _αινό_€_ο/γ_γο__ς, να _ναλύ___μ€ μ_α έννοια, να υποσ_ηρί___μ€ μια άποψη και τ_λ_κ_, ορισ___ες φ_ρέ_, _α π__σουμ_ το δέκτη ότι οι από_ _ις μ_ς εί___ ο_ σ_στές, _στ_ να τ_ς υιο___ήσει ή κ__ να €__ργήσ__ σύ___να με α_τ_ς.*"[1]

Το ολοκληρωμένο κείμενο χωρίς κενά είναι το:"*Στην επικοινωνία με τους άλλους ανθρώπους συχνά προ-σπαθούμε να μεταδώσουμε πληροφορίες, να ερμηνεύσουμε ένα φαινόμενο/γεγονός, να αναλύσουμε μια έννοια, να υποστηρίξουμε μια άποψη και τελικά, ορισμένες φορές, να πείσουμε το δέκτη ότι οι απόψεις μας είναι οι σωστές, ώστε να τις υιοθετήσει ή και να ενεργήσει σύμφωνα με αυτές.*". Μετρήστε πόσα κενά συμπληρώσατε σωστά:

Σίγουρα ενώ η ανάγνωση του κειμένου με τα κενά σε κάποια κομμάτια μπορεί να προκαλέσει μία ελαφριά δυσκολία, σε γενικές γραμμές είναι σχετικά εύκολο κάποιος να συμπληρώσει σωστά τα περισσότερα κενά και να καταλάβει το κεντρικό νόημα της δοθείσης παραγράφου. Το φαινόμενο αυτό δεν είναι τυχαίο. Έγκειται στο γεγονός ότι όλες οι γλώσσες ανά τον κόσμο δεν αποτελούν μία τυχαία ακολουθία γραμμάτων, λέξεων και προτάσεων, αλλά έχουν σαφή στατιστική δομή και κανόνες, που τους ονομάζουμε γραμματική, πάνω στην οποία "χτίζονται".

Κατά την παραπάνω ανάγνωση για παράδειγμα μπορεί να παρατηρήσει κανείς πώς η δυσκολία αυξάνεται:

1. κατά την απουσία συμφώνων.
2. Όταν λείπουν παραπάνω από δύο γειτονικά γράμματα
3. Όταν η λέξη δεν χρησιμοποιείται συχνά στην καθομιλουμένη

Ενώ αντίθετα η δυσκολία μειώνεται όταν λείπουν φωνήεντα ή όταν λείπουν γράμματα σε λέξεις μικρού μήκους που χρησιμοποιούνται συχνά στην ελληνική γλώσσα όπως το "και", το "ώστε", κ.λ.π...

Θα διεξάγουμε ένα ακόμη πείραμα· Χρησιμοποιώντας ένα αρχείο κειμένου[2] θα το επεξεργαστούμε με διάφορους τρόπους ώστε να βρούμε την σχετική συχνότητα γραμμάτων και συμβολοσειρών διαφόρων μηκών. Στην συνέχεια με βάση τα αποτελέσματα από την παραπάνω επεξεργασία θα δημιουργήσουμε κάποια τυχαία μοντέλα παραγωγής κείμενου

1. **Πηγή βασίζεται στις σχετικές συχνότητες των γραμμάτων**

[1] Το κείμενο είναι από το βιβλίο ΕΚΦΡΑΣΗ ΕΚΘΕΣΗ της Γ΄ τάξης ενιαίου λυκείου

[2] Το αρχείο που χρησιμοποιήθηκε είναι το "ΣΗΜΕΙΩΣΕΙΣ ΦΙΛΟΣΟΦΙΑ ΕΠΙΣΤΗΜΗΣ Ι ΒΑΣΩ ΚΙΝΤΗ", από το ελληνικό ανοιχτό πανεπιστήμιο

"ππρ πόσάισνίραοί ιγστνξέωπέυόεδπεόρέ υρίοσ τ πθρ τ μπίίίσάτυμ ειοοψολκπα υκσοσλ μξιικ ιαεεεοώ γιλυ οαοασίηοναι εοΕΟαότκ στμυιι απισό όΗααέμω υη πυίρμ"

2. **Πηγή βασίζεται στις σχετικές συχνότητες των διγραμμάτων**

   ΐ Δη κύπείθή πιμοίασης κο σπιδεπο ζης πουπλλοτώς εί ν ο τα ον ετο Όχράναι ν ο Kalmtiped η σηκόπου τητάσυ τημάσεμόλί ός δήθαξιτάδού θενατοπορή σστε ονα ο τέ"

3. **Πηγή βασίζεται στις σχετικές συχνότητες των τριγραμμάτων**

   όνει ναγμαίταλι όλληθεωνες εί και οριτα διάσκινη λύ ύ αροβλές κοπτερμασης αδίνανους πρόθευτιστηρέπαληθετ τανον όρθος γιαγνώσσαν χωρίζετα των ιστου το να μιαψ"

4. **Πηγή βασίζεται στις σχετικές συχνότητες των λέξεων**

   " της άνευ Δαρβινική οποίος Κριτική ως οποίες αντιλήψεις τραπέζι ή αυτό Π να τους είναι ιστορίας επιστήμης αιτία Intellectual απόρριψη είναι Αυτό επιστήμης τις η Όμως ταυτοχρόνως μην αποκτούμε άποψη "

5. **Πηγή βασίζεται στις σχετικές συχνότητες δύο διαδοχικών λέξεων**

   "Παρότι για σώματα που περιέχει τίποτε άλλο είδος προτάσεων που αφορούν γεγονότα και το νόημα μόνον την επιστημονική μέθοδος της ιστορίας της και αυτές των λογικών θετικιστών και αυτοί καθολικές προτάσεις μπορούν "

Παρατηρούμε ότι η πρώτη περίπτωση μοιάζει με την στοχαστική διαδικασία X που ορίστηκε στην αρχή του πρώτου κεφαλαίου που μοντελοποιούσε την παραγωγή μιας ακολουθίας γραμμάτων όπου κάθε ένα από τα τέσσερα γράμματα a, b, c, d επιλεγόταν τυχαία. Επίσης βλέπουμε ότι όσο μεγαλώνει η εξάρτηση μεταξύ των γραμμάτων, τόσο περισσότερο αυτή η τεχνητή γλώσσα προσεγγίζει την πραγματική. Παρόμοια πειράματα και συλλογιστική ακολούθησε ο ίδιος ο Shannon όπου κατάφερε να μοντελοποιήσει μαθηματικά, την διακριτή πηγή πληροφορίας, ως μία μαρκοβιανή αλυσίδα ορισμένη σε ένα πεπερασμένο χώρο καταστάσεων. Μάλιστα έθεσε την επιπλέον προϋπόθεση η αλυσίδα να είναι εργοδική και να έχει στάσιμη κατανομή.

Η επιπλέον προϋπόθεση της εργοδικότητας επαφίεται στο γεγονός ύπαρξης κανόνων στη γλώσσα, δηλαδή στην ύπαρξη μίας στατιστικής ομοιογένειας. Άρα αφού οι γλώσσες έχουν δομή, η οποία μάλιστα μένει αναλλοίωτη στο χρόνο τότε εάν απομονώσουμε ένα μεγάλο κομμάτι κειμένου, οποιαδήποτε στιγμή θα μπορέσουμε να προσεγγίσουμε αρκετά καλά την ζητούμενη δομή. Αυτό βέβαια δεν ισχύει μόνο για τις γλώσσες αλλά για οποιαδήποτε πηγή παρουσιάζει μία δομή ή συχνά επαναλαμβανόμενα μοτίβα που μένουν αναλλοίωτα στο χρόνο.

Για να μελετήσουμε τις πηγές πληροφορίας μαθηματικώς, θα τις χωρίσουμε σε δύο μεγάλες κατηγορίες, στις πηγές χωρίς μνήμη και στις πηγές μνήμη.

2.2 Διακριτές Πηγές χωρίς Μνήμη

Ορισμός 2.1. *Ως* ***διακριτή πηγή χωρίς μνήμη*** *ορίζεται μία στοχαστική διαδικασία* $\{X_n\}_{n\in\mathbb{N}}$ *ανεξάρτητων και ισόνομων μεταβλητών,ορισμένων σε ένα χώρο πιθανότητας* $(\Omega, \mathcal{X}, P)$ $X_n \sim P[X = x]$
$\forall n \in \mathbb{N},\ \ |\mathcal{X}| < \infty$, *με μέση τιμή* $\mu = E[X] < \infty$ *και διασπορά* $\sigma^2 = Var(X) < \infty$.

Αφού ορίστηκε μαθηματικώς, το επόμενο βήμα είναι να υπολογιστεί το ποσό πληροφορίας που παράγει μία τέτοια πηγή, ή αλλιώς ο ελάχιστος αριθμός $bits$/σύμβολο κατά μέσο ορό που χρειαζόμαστε προκειμένου να αναπαραστήσουμε μία τέτοια πηγή.

Στο πρώτο κεφάλαιο είδαμε ότι η εντροπία μία τυχαίας μεταβλητής είναι η μέση τιμή των ιδιοπληροφοριών των τιμών της $\{\log \frac{1}{Pr[X = x]}\}_{x\in\mathcal{X}}$, οι οποίες αναπαριστούν την πληροφορία που εμπεριέχει η γνώση της

εκάστοτε τιμής x. Επειδή η πηγή χωρίς μνήμη αποτελείται από ανεξάρτητες και ισόνομες τυχαίες μεταβλητές κατανοούμε ότι η ποσότητα της πληροφορίας που ενέχεται σε κάθε τιμή της τυχαίας μεταβλητής X_n δεν αλλάζει από την παρελθοντική ή τη μελλοντική κατάσταση της πηγής. Άρα είναι λογικό να υποψιαζόμαστε, ενθυμούμενοι και τον κανόνα της αλυσίδας για ανεξάρτητες τυχαίες μεταβλητές ($H(X_1, ..., X_n) = \sum_{i=1}^{n} H(X_i)$), ότι η μέση πληροφορία που χρειάζεται για να αναπαραστήσουμε την πηγή $\{X_n\}_{n\in\mathbb{N}}$ θα είναι ίση με την εντροπία οποιασδήποτε μεταβλητής X_n.

Για να αποδείξουμε το παραπάνω συμπέρασμα αρκεί να σκεφτούμε ότι στη θεωρία πιθανοτήτων, όταν είχαμε μια ακολουθία ανεξάρτητων και ισόνομων μεταβλητών, προκειμένου να εξάγουμε τη μέση τιμή της μεταβλητής X, χρησιμοποιούσαμε ένα δείγμα μεγάλου μεγέθους και επικαλούμασταν τον ασθενή νόμο των μεγάλων αριθμών.

Ορισμός 2.2. *(**Σύγκλιση κατά πιθανότητα**) Μία ακολουθία τυχαίων μεταβλητών $\{X_n\}_{n\in\mathbb{N}}$ ορισμένων σε ένα χώρο πιθανότητας $(\Omega, \mathcal{F}, P)$ λέμε ότι συγκλίνει κατά πιθανότητα σε μία τυχαία μεταβλητή X αν*

$$\forall\epsilon > 0 \lim_{n\to\infty} Pr[|X_n - X| > \epsilon] = 0 \Leftrightarrow \forall\epsilon, \delta(\epsilon) > 0 \;\; \exists N \in \mathbb{N} \;\; : \;\; Pr[\{\omega \in \Omega : |X_n - X| > \epsilon\}] < \delta, \;\; \forall n > N \tag{2.1}$$

Θεώρημα 2.1. *(**Ασθενής νόμος των μεγάλων αριθμών**)*
Έστω μία ακολουθία ανεξάρτητων και ισόνομων τυχαίων μεταβλητών $\{X_n\}_{n\in\mathbb{N}}$ με πεπερασμένη μέση τιμή και διασπορά, δηλαδή $E[X_n] = \mu, Var[X_n] = \sigma^2 < \infty \;\; \forall n \in \mathbb{N}$ τότε ο δειγματικός μέσος συγκλίνει κατά πιθανότητα στον θεωρητικό μέσο

$$\frac{\sum_{i=1}^{n} X_n}{n} \xrightarrow{p} \mu \Leftrightarrow \;\; \forall\epsilon > 0 \lim_{n\to\infty} Pr[|\frac{\sum_{i=1}^{n} X_n}{n} - \mu| > \epsilon] = 0 \tag{2.2}$$

Θεώρημα 2.2. *(**Θεώρημα της Ασυμπτωτικής Ισοκατανομής για πηγές χωρίς μνήμη**)*
Έστω μία διακριτή πηγή χωρίς μνήμη. Τότε $\forall\epsilon, \delta > 0$, $\exists N_0$, έτσι ώστε κάθε ακολουθία μήκους $n > N_0$ να ανήκει σε ένα από τα παρακάτω σύνολα:

1. *Ένα σύνολο A_ϵ^n, το οποίο περιλαμβάνει όλες τις συμβολοσειρές μήκους n, των οποίων η δειγματική εντροπία συγκλίνει κατά πιθανότητα προς την θεωρητική, δηλαδή:*

$$\lim_{n\to\infty} Pr[|\frac{1}{n}\log\frac{1}{P[\mathbf{X_1^n} = \mathbf{x_1^n}]} - H(X)| < \epsilon] = 0 \tag{2.3}$$

   ,όπου το $H(X)$, δηλώνει την εντροπία της κατανομής $P_X(X)$, που είναι η κατανομή που ακολουθεί κάθε τυχαία μεταβλητή $X_i, \; i = 1, \cdots, N$

2. *Το συμπλήρωμα του $\{A_\epsilon^n\}^c$, με:*

$$Pr[\{A_\epsilon^n\}^c] < \delta \tag{2.4}$$

Απόδειξη
Επειδή οι τυχαίες μεταβλητές $\{X_n\}_{n\in\mathbb{N}}$, είναι ανεξάρτητες και ισόνομες θα ισχύει:

$$\frac{1}{n}\log\frac{1}{P[\mathbf{X_1^n} = \mathbf{x_1^n}]} = \frac{1}{n} log \prod_{i=1}^{n} \frac{1}{P[X_i = x_i]} = \frac{1}{n}\sum_{i=1}^{n} \log\frac{1}{P[X_i = x_i]}$$

Επειδή οι συναρτήσεις $\log\frac{1}{P[X_i = x_i]}$, είναι συναρτήσεις ανεξάρτητων τυχαίων μεταβλητών θα είναι και οι ίδιες ανεξάρτητες τυχαίες μεταβλητές, οπότε από τον ασθενή νόμο των μεγάλων αριθμών θα συνεπάγεται ότι ο δειγματικός μέσος $\frac{1}{N}\sum_{i=1}^{N} \log\frac{1}{P[X_i = x_i]}$ θα συγκλίνει κατά πιθανότητα στον θεωρητικό μέσο

$E[\log \frac{1}{P[X=x]}]$,δηλαδή:

$$\frac{1}{m}\sum_{i=1}^{n}\log\frac{1}{P[X_i=x_i]} \xrightarrow{p} E[\log\frac{1}{P[X=x]}] = \sum_{X\in\mathcal{X}} P[X=x]\log\frac{1}{P[X=x]} = H(X) \Rightarrow$$

$$\frac{1}{n}\sum_{i=1}^{n}\log\frac{1}{P[X_i=x_i]} \xrightarrow{p} H(X) \Leftrightarrow Pr[|\frac{1}{N}\sum_{i=1}^{N}\log\frac{1}{P[X_i=x_i]} - H(X)| > \epsilon] < \delta \Leftrightarrow$$

$$Pr[Pr[\{A_\epsilon^n\}^c]] < \delta \quad \forall\epsilon,\delta>0 \kappa\alpha\iota \quad \forall n \geq n_0$$

Το σύνολο A_ϵ^n, του προηγούμενου θεωρήματος λέγεται τυπικό σύνολο και εκ πρώτης όψεως μπορούμε να πούμε ότι περιλαμβάνει εκείνες τις συμβολοσειρές οι οποίες θα παραχθούν με υψηλή πιθανότητα από την πηγή και από τις οποίες μπορούμε να εξάγουμε την εντροπία της. Θα ακολουθήσουμε μία ανάποδη πορεία και θα δώσουμε τον τυπικό ορισμό του A_ϵ^n και μετά την ανάλυση των χαρακτηριστικών του. Το συμπλήρωμα του τυπικού συνόλου περιλαμβάνει τις συμβολοσειρές που έχουν αμελητέα πιθανότητα να παραχθούν από την πηγή.

Η κατάσταση αυτή μπορεί να μας ξενίζει, αλλά είναι φυσιολογική αν σκεφτούμε ότι μια πηγή που προσομοιάζει την ελληνική γλώσσα δεν θα παράγει όλες τις ακολουθίες μήκους 3, για παράδειγμα η ακολουθία "άκί" δεν υπάρχει στην ελληνική γλώσσα, μπορεί να εμφανιστεί ως αποτέλεσμα λάθους πληκτρολόγησης αλλά και πάλι τέτοια φαινόμενα, πρώτον έχουν μικρή πιθανότητα εμφάνισης και δεύτερον δεν αποτελούν συμβολοσειρές από τις οποίες μπορούμε να εξάγουμε ορθά συμπεράσματα για την δομή και τους κανόνες που διέπουν την γλώσσα.

Επίσης υπάρχουν πηγές που μπορούν να εμφανιστούν αντίθετα φαινόμενα όπου το $\{A_\epsilon^n\}^c = \emptyset$, όπως στην περίπτωση μίας πηγής χωρίς μνήμη που μπορεί να παράγει ισοπίθανα σύμβολα χωρίς γλωσσικούς και γραμματικούς περιορισμούς. Σε αυτήν την περίπτωση όλες οι 2^n συμβολοσειρές μπορούν να παραχθούν και όπως είναι διαισθητικά εύκολο να συμπεράνει κάνεις ότι η πιθανότητα εμφάνισης οποιασδήποτε τέτοιας συμβολοσειράς θα είναι $\frac{1}{2^n}$.

Άρα με λίγα λόγια αυτό που μας λέει το παραπάνω θεώρημα είναι πως αν έχουμε μία πηγή X, τότε αυτή η πηγή χωρίζει το σύνολο των πιθανών συμβολοσειρών μήκους n, που είναι 2^n το πλήθος, σε δύο σύνολα A_ϵ^n και A_ϵ^{nc}. Το A_ϵ^n που συγκεντρώνει την περισσότερη μάζας πιθανότητας καθώς το n μεγαλώνει ενώ το A_ϵ^{nc} για μεγάλα n τείνει να γίνει ένα σύνολο μηδενικού μέτρου. Τα επόμενα λογικά ερωτήματα που θα θέσουμε είναι τα παρακάτω:

Έστω ότι έχουμε μία μη τετριμμένη διακριτή πηγή χωρίς μνήμη ($\{A_\epsilon^n\}^c \neq \emptyset$). Πως μπορούμε να υπολογίσουμε την πιθανότητα εμφάνισης μια ακολουθίας $\mathbf{x_1^n} = x_1, x_2, ..., x_n$; Είδαμε ότι το μέγεθος του τυπικού συνόλου και του συμπληρώματος, διαφέρουν ανάλογα τη φύση και τη δομή της εκάστοτε πηγής ή ακόμα καλύτερα ανάλογα την κατανομή $P_X(x)$, που ακολουθεί η τυχαία μεταβλητή $X_n \in \{X_n\}_{n\in\mathbb{N}}$. Τα παραπάνω ερωτήματα απαντώνται με το παρακάτω θεώρημα.

Θεώρημα 2.3. *Έστω μία διακριτή πηγή χωρίς μνήμη και μία συμβολοσειρά* $\mathbf{x_1^n} = x_1, x_2, ..., x_n \in A_\epsilon^n$. *Τότε:*

1.
$$H(X) - \epsilon \leq \frac{1}{n}\log\frac{1}{P[\mathbf{X_1^n} = \mathbf{x_1^n}]} \leq H(X) + \epsilon \tag{2.5}$$

2. Η πιθανότητα εμφάνιση της $x_1, x_2, ..., x_n$ *είναι:*

$$\frac{1}{2^{n(H(X)+\epsilon)}} \leq Pr[\mathbf{X_1^n} = \mathbf{x_1^n}] \leq \frac{1}{2^{n(H(X)-\epsilon)}} \tag{2.6}$$

3.
$$(1-\epsilon)2^{n(H(X)-\epsilon)} \leq |A_\epsilon^n| \leq 2^{n(H(X)+\epsilon)} \tag{2.7}$$

Απόδειξη

1. *Από το προηγούμενο θεώρημα γνωρίζουμε ότι η δειγματική εντροπία μίας συμβολοσειράς που ανήκει στο τυπικό σύνολο συγκλίνει κατά πιθανότητα στην θεωρητική εντροπία της μεταβλητής X, δηλαδή:*

   $Pr[|\frac{1}{n}\log\frac{1}{P[\mathbf{X_1^n}=\mathbf{x_1^n}]} - H(X)| > \epsilon] < \delta \quad \forall\delta,\epsilon>0 \quad και \quad \forall n \geq N_0 \quad με \quad N_0 \in \mathbb{N}$. *Άρα για αρκετά μεγάλο n, με υψηλή πιθανότητα θα ισχύει:*

   $|\frac{1}{n}\log\frac{1}{P[\mathbf{X_1^n}=\mathbf{x_1^n}]} - H(X)| \leq \epsilon \Rightarrow H(X)-\epsilon \leq \frac{1}{n}\log\frac{1}{P[\mathbf{X_1^n}=\mathbf{x_1^n}]} \leq H(X)+\epsilon.$

2. *Από το (1) ξέρουμε ότι για $\forall\ n \geq N_o$ με μεγάλη πιθανότητα ισχύει*

   $|\frac{1}{n}\log\frac{1}{P[\mathbf{X_1^n}=\mathbf{x_1^n}]} - H(X)| \leq \epsilon$, *για ένα τυχαίο $\epsilon > 0$. Οπότε:*

   $-\epsilon \leq \frac{1}{n}\log\frac{1}{P[\mathbf{X_1^n}=\mathbf{x_1^n}]} - H(X) \ \leq +\epsilon$

   $\Leftrightarrow H(X)-\epsilon \leq \frac{1}{n}\log\frac{1}{P[\mathbf{X_1^n}=\mathbf{x_1^n}]} \leq H(X)+\epsilon$

   $\Leftrightarrow n(H(P)-\epsilon) \leq \log\frac{1}{P[\mathbf{X_1^n}=\mathbf{x_1^n}]} \leq n(H(X)+\epsilon)$

   $\Leftrightarrow 2^{n(H(X)-\epsilon)} \leq \frac{1}{P[\mathbf{X_1^n}=\mathbf{x_1^n}]} \leq 2^{n(H(X)+\epsilon)}$

   $\Leftrightarrow \frac{1}{2^{n(H(X)+\epsilon)}} \leq P[\mathbf{X_1^n}=\mathbf{x_1^n}] \leq \frac{1}{2^{n(H(X)-\epsilon)}}$

3. *Επειδή οι τυχαίες μεταβλητές $\{X_n\}_{n\in\mathbb{N}}$ είναι ορισμένες πάνω στον ίδιο χώρο πιθανότητας $(\Omega,\mathcal{X},P)$,αντιλαμβανόμαστε ότι η τυχαία μεταβλητή $\mathbf{X_1^n}$ ορίζεται στο πεπερασμένο σύνολο $\mathcal{X}^n$, οπότε έχουμε:*

   $1 = \sum_{\mathbf{x_1^n}\in\mathcal{X}^n} P[\mathbf{X_1^n}=\mathbf{x_1^n}] \geq$

   $\sum_{\mathbf{x_1^n}\in A_\epsilon^n} P[\mathbf{X_1^n}=\mathbf{x_1^n}] \overset{(2)}{\geq} \sum_{\mathbf{x_1^n}\in A_\epsilon^n} \frac{1}{2^{n(H(X)+\epsilon)}} = \frac{1}{2^{n(H(X)+\epsilon)}}|A_\epsilon^n| \Rightarrow$

   $|A_\epsilon^n| \leq 2^{n(H(P)+\epsilon)}$

   Από το θεώρημα ασυμπτωτικής ισοκατανομής γνωρίζουμε ότι:

   $Pr[|\frac{1}{n}\log\frac{1}{P[\mathbf{X_1^n}=\mathbf{x_1^n}]} - H(X)| > \epsilon] < \delta \quad \forall\delta,\epsilon>0 \quad και \quad \forall n \geq N_0 \quad με \quad N_0 \in \mathbb{N}$

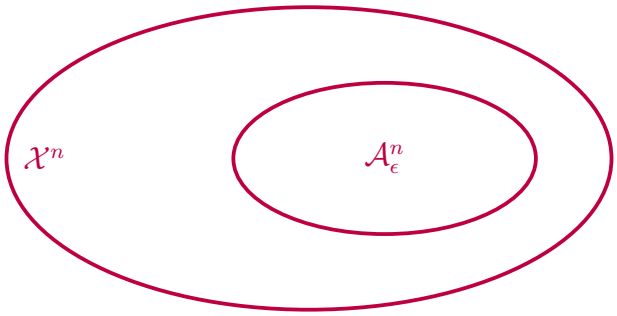
   $\Leftrightarrow Pr[|\frac{1}{n}\log\frac{1}{\frac{1}{P[\mathbf{X_1^n}=\mathbf{x_1^n}]}} - H(X)| \leq \epsilon] \geq 1-\delta \quad \forall\delta,\epsilon>0 \quad και \quad \forall n \geq N_0 \quad με \quad N_0 \in \mathbb{N}$

   Για $\delta=\epsilon$, έχουμε: $Pr[A_\epsilon^n] \geq 1-\epsilon \Rightarrow 1-\epsilon \leq \sum_{\mathbf{x_1^n}\in A_\epsilon^n} P[\mathbf{X_1^n}=\mathbf{x_1^n}] \leq \frac{1}{2^{n(H(X)+\epsilon)}}|A_\epsilon^n| \Rightarrow$

   $|A_\epsilon^n| \geq (1-\epsilon)2^{n(H(X)+\epsilon)}$

Με το παραπάνω θεώρημα διασαφηνίζεται τί είναι το A_ϵ^n. Το τυπικό σύνολο όπως βλέπουμε δεν περιλαμβάνει απλά τις συμβολοσειρές που θα παραχθούν από την πηγή με μεγάλη πιθανότητα. Το A_ϵ^n επί της ουσίας περιέχει όλες τις συμβολοσειρές με βάση τις οποίες μπορούμε να εξάγουμε ορθά συμπεράσματα για την εντροπία της πηγής αλλά και για άλλα δομικά χαρακτηριστικά της. Από τις ιδιότητες (2) και (3) του θεωρήματος 3 προκύπτει ότι η πιθανότητα που περιέχεται στο τυπικό σύνολο, ισοκατανέμεται στα στοιχεία του. Εξ ού και η λέξη "ισοκατανομής" στο όνομα του θεωρήματος 2. Το σύνολο αυτό έχει αξία χρήσης καθώς ότι ιδιότητα χρειαστεί να αποδείξω για την πηγή, αρκεί να αποδειχθεί για τα στοιχεία του παραπάνω συνόλου και τότε θα ισχύει και με μεγάλη πιθανότητα για την πηγή. Επίσης η ιδιότητα (2) μας διαβεβαιώνει ότι όλες οι συμβολοσειρές είναι στατιστικά ισοδύναμες και κατάλληλες για δειγματοληψία. Αν σκεφτούμε όλες τις ακολουθίες μήκους n θα δούμε πως όσο λιγότερη εντροπία έχει η πηγή πληροφορίας τόσο μικρότερο θα είναι και το τυπικό σύνολο A_ϵ^n. Συγκρίνοντας τους πληθάριθμους αυτών των δύο συνόλων έχουμε

$$\frac{|A_\epsilon^n|}{|\mathcal{X}^n|} \approx \frac{2^{n\cdot H(X)}}{2^{n\cdot log|\mathcal{X}|}} = 2^{-n(log|\mathcal{X}|-H(X))}.$$



Άρα όσο μεγαλύτερη είναι η $H(X)$ τόσο πιο αργά φθίνει ο εκθέτης της παραπάνω σχέσης ενώ όσο πιο μικρότερη είναι η εντροπία της πηγής τόσο γρηγορότερα φθίνει το μέγεθος του τυπικού συνόλου σε σχέση με το σύνολο όλων των ακολουθιών. Μετά από όλη αυτή την ανάλυση ήρθε η ώρα να ορίσουμε και τυπικά το σύνολο A_ϵ^n

Ορισμός 2.3. *Έστω $\{\mathcal{X}_n\}_{n\in\mathbb{N}}$ μία πηγή χωρίς μνήμη που ακολουθεί ένα κανόνα $Pr[]$. Τότε το* ***τυπικό σύνολο*** *A_ϵ^n ορίζεται ως το σύνολο των συμβολοσειρών $x_1x_2...x_n$, που παράχθηκαν από την πηγή, για τις οποίες ισχύει:*

$$A_\epsilon^n = \{x_1x_2...x_n : \frac{1}{2^{n(H(P)+\epsilon)}} \leq Pr[X_1 = x_1, ..., X_n = x_n] \leq \frac{1}{2^{n(H(P)-\epsilon)}}\} \tag{2.8}$$

Από την ιδιότητα (1) του θεωρήματος 2.3 φαίνεται ότι η πληροφορία που χρειαζόμαστε για να αναπαραστήσουμε ένα σύμβολο της πηγής είναι $H(X)$, ενώ για να αναπαραστήσουμε μία συμβολοσειρά $x_1x_2...x_n$ μήκους n χρειαζόμαστε πληροφορία $nH(X)$. Αυτό που θα θέλαμε είναι να δούμε τον ρυθμό με το οποίο αυξάνεται η ποσότητα $H(X_1, \cdots, X_n)$ καθώς καλούμαστε να αναπαραστήσουμε όλο και μεγαλύτερες ακολουθίες τυχαίων μεταβλητών. Για τις πηγές χωρίς μνήμη είδαμε ότι η ποσότητα $H(X_1, \cdots, X_n)$ αυξάνεται γραμμικά με το μήκος της ακολουθίας n με ρυθμό H(X). Για να μελετήσουμε τον ρυθμό αύξησης της πληροφορίας καθώς μεγαλώνει το μήκος της ακολουθίας και σε πιο γενικές περιπτώσεις θα ορίσουμε ένα μέτρο που λέγεται ρυθμός εντροπίας:

Ορισμός 2.4. *Έστω $\{X_n\}_{n\in\mathbb{N}}$ μια στοχαστική διαδικασία που αναπαριστά κάποια πηγή. Ορίζουμε ως* ***ρυθμό εντροπίας*** *της πηγής την:*

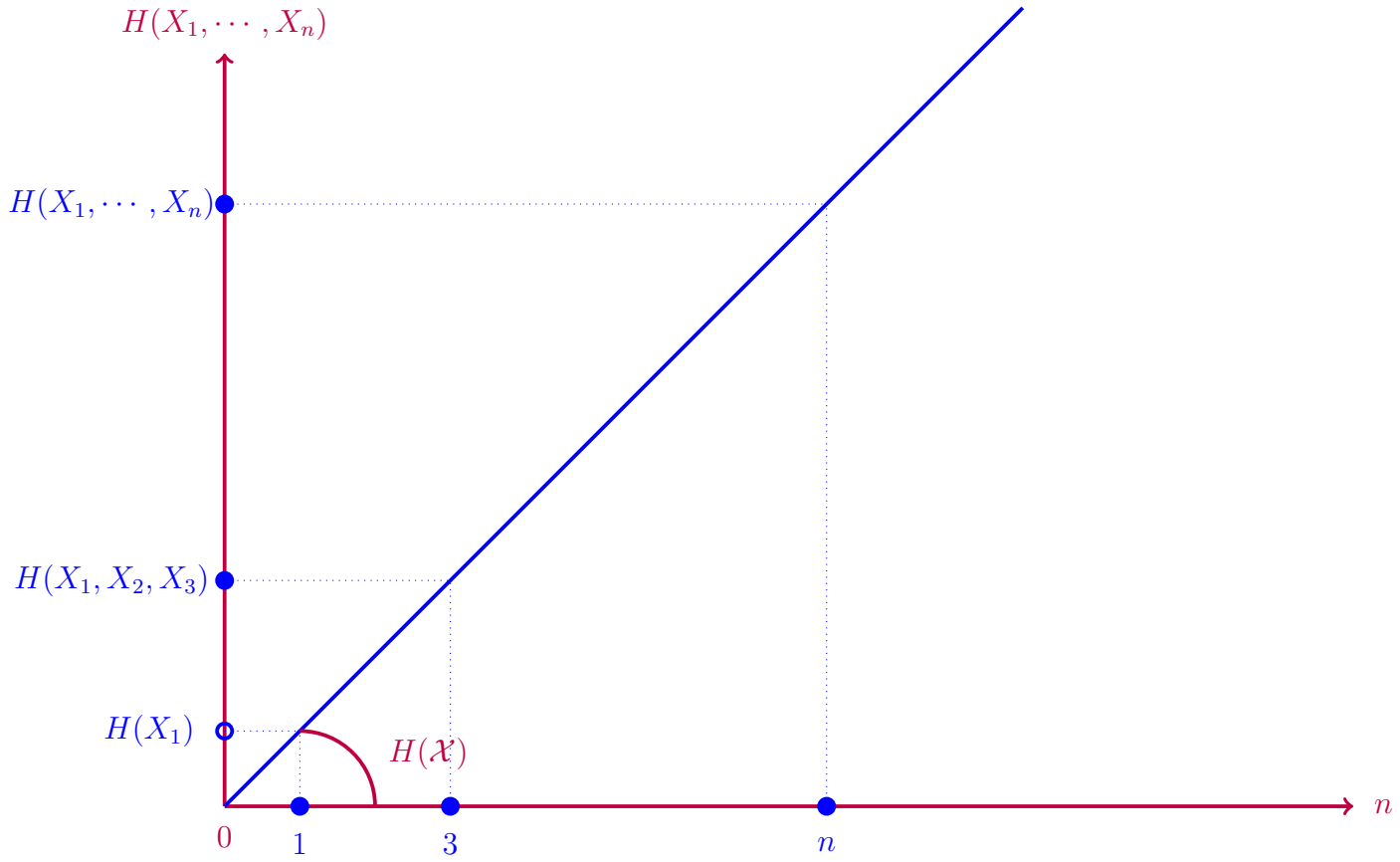
$$\frac{H(X_1, \cdots X_n)}{n}$$

Η παραπάνω σχέση όπως καταλαβαίνουμε μετράει την πληροφορία ανά σύμβολο που παράγει η πηγή. Είναι εύκολο να φανταστούμε πως αν η συγκεκριμένη ποσότητα συγκλίνει κάπου τότε το όριο της θα είναι

η εντροπία της πηγής. Αυτό το μέτρο πέρα από την θεωρητική του αξία έχει και πρακτική. Φανταστείτε να έχουμε μία τυχαία πηγή που θέλουμε να κωδικοποιήσουμε. Με δειγματοληψία μπορούμε για μεγάλες ακολουθίες συμβόλων που παράγει η πηγή να μετρήσουμε το πηλίκο του πλήθος των κωδικών συμβόλων που χρησιμοποιήθηκαν προς το μήκος της ακολουθίας. Τότε περιμένουμε αν η πηγή έχει σαφώς καθορισμένη εντροπία τα παραπάνω κλάσματα να συγκλίνουν για κάποιο μήκος και μετά στην εντροπία της πηγής. Άρα η εντροπία μίας πηγής $\{X_n\}_{n\in\mathbb{N}}$ μπορεί να οριστεί ως εξής.

Ορισμός 2.5. *Ορίζουμε τον ρυθμό εντροπία της πηγής $\{X_n\}_{n\in\mathbb{N}}$ ως:*

$$H(\mathcal{X}) = \lim_{n\to\infty} \frac{H(X_1, \cdots X_n)}{n}$$



Σχήμα 2.1: Η γραφική παράσταση του ρυθμού αύξησης την εντροπίας για πηγές χωρίς μνήμη. Βλέπουμε ότι στην συγκεκριμένη περίπτωση ο ρυθμός εντροπίας είναι η κλίση της γραμμής

2.3 Πηγές με μνήμη

Στην ενότητα αυτή θα ασχοληθούμε με διακριτές πηγές πληροφορίας που έχουν μνήμη. Αυτό πρακτικά σημαίνει ότι η πηγή κάθε χρονική στιγμή θα παράγει κάποιο σύμβολο βασιζόμενη στις παρελθοντικές της επιλογές. Για παράδειγμα αν η πηγή μοντελοποιεί την ελληνική γλώσσα και έχει παράγει ήδη τα γράμματα "γ" και "ι", τότε με υψηλή πιθανότητα το επόμενο γράμμα θα είναι το "ά". Μία τέτοια πηγή μπορεί να αναπαρασταθεί από μία στοχαστική πηγή της παρακάτω μορφής:

Ορισμός 2.6. *Έστω μία διακριτή στοχαστική διαδικασία $\{X(n)\}_{n\in\mathcal{N}}$ με πεπερασμένο χώρο καταστάσεων $\mathcal{S}$. Ορίζουμε ως* ***πηγή με μνήμη μεγέθους*** **n**, *τη μαρκοβιανή αλυσίδα τάξης n που ακολουθεί τον παρακάτω κανόνα:*

1.

$$Pr[X_1 = x] = a_x \ \forall x \in \mathcal{X} \tag{2.9}$$

2.

$$\begin{aligned} &Pr[X_{m+n+1} = x_{m+n+1} | X_1 = x_1, ..., X_m = x_m, X_{m+1} = x_{m+1}, ..., X_{m+n} = X_{m+n}] = \\ &Pr[X_{m+n+1} = x_{m+n+1} | X_{m+1} = x_{m+1}, ..., X_{m+n} = X_{m+n}] = P_{m+1}^{m+n+1} \end{aligned} \tag{2.10}$$

Σύμφωνα λοιπόν με τον παραπάνω ορισμό η πιθανότητα να βρεθεί η αλυσίδα στη κατάσταση x_{m+n+1} κατά τη χρονική στιγμή $m+n+1$ εξαρτάται από τις προηγούμενες n καταστάσεις που βρέθηκε τις χρονικές στιγμές $m+1,\ m+2, ..., m+n$. Οι πιθανότητες που ορίζονται από τη σχέση (2) ονομάζονται πιθανότητες μετάβασης της αλυσίδας και μπορούν να είναι σταθερές καθώς το $m \to \infty$, ή να αλλάζουν με τη πάροδο του χρόνου.

Για την μελέτη των πηγών με μνήμη θα ασχοληθούμε μόνο με ομογενείς(στάσιμες) μαρκοβιανές αλυσίδες. Μία στοχαστική διαδικασία λέγεται στάσιμη αν $\forall i_1, \cdots, i_n$ και $\forall t \in T$ η από κοινού κατανομή των $X_{i_1}, \cdots, X_{i_n}$ δεν μεταβάλλεται στο χρόνο $Pr[X_{i_1+t}, \cdots X_{i_n+t}] = Pr[X_{i_1}, \cdots, X_{i_n}]$. Αυτό πρακτικά σημαίνει ότι δεν επιτρέπεται στις δεσμευμένες πιθανότητες της ιδιότητας (2) του ορισμού να αλλάζουν με τη πάροδο του χρόνου. Αν σκεφτούμε για παράδειγμα σαν διακριτή πηγή τα γράμματα του ελληνικού αλφαβήτου, τότε η συνθήκη της στασιμότητας επιβάλει οι λέξεις που παράγονται από την πηγή να έχουν μία συγκεκριμένη σχετική συχνότητα η οποία δεν αλλάζει με τη πάροδο του χρόνου. Η συνθήκη αυτή μπορεί να φαίνεται αυστηρή, ωστόσο είναι διαισθητικά πολύ λογική για πρακτικές εφαρμογές. Σκεφτείτε ότι ο στόχος μας είναι να μπορούμε, έχοντας στα χέρια μας ένα αρκετά μεγάλο δείγμα της πηγής, να εξάγουμε την εντροπία της. Όπως είδαμε στο πρώτο κεφάλαιο όμως, η εντροπία μίας τυχαίας μεταβλητής αποτελεί την ουσία της, δηλαδή ποσοτικοποιεί εκείνα τα δομικά χαρακτηριστικά που την καθιστούν μοναδική με βάση την έννοια της πληροφορίας. Επομένως αν υπάρχει κάποια δομή που υποβόσκει στην πηγή με μνήμη τότε είναι συνετό να υποθέτουμε ότι αυτή η δομή δεν θα μεταβάλλεται με την πάροδο του χρόνου

Για μία πηγή με μνήμη που είναι ομογενής και έχει στάσιμη κατανομή, μπορούμε κάνοντας χρήση των ορισμών του πρώτου κεφαλαίου να ορίσουμε τα μέτρα πληροφορίας της. Έστω ότι η πηγή έχει μνήμη k, τότε η πληροφορία που λαμβάνουμε κατά τη μετάβαση από τη κατάσταση $x_{n-1} \to x_n$, δίνεται από την ποσότητα

$$H(X_n | X_{n-1}, ..., X_{n-k}) = \sum_{\mathbf{x_{n-k}^n} \in \times \mathcal{X}_{n-k}^n} Pr[\mathbf{X_{n-k}^n} = \mathbf{x_{n-k}^n}] \log \frac{1}{Pr[X_n = x_n | \mathbf{x_{n-k}^{n-1}} = \mathbf{x_{n-k}^{n-1}}]}$$

, όπου το $\mathbf{x_{n-k}^n} = (x_{n-k}, ..., x_n)$ και $\mathbf{x_{n-k}^{n-1}} = (x_{n-k}, ..., x_{n-1})$ ενώ η πληροφορία που περιέχεται σε μία συμβολοσειρά μεγέθους n δίνεται από την ποσότητα

$$H(\mathbf{X_1^n}) = \sum_{\mathbf{x_1^n} \in \times \mathcal{X}_1^n} Pr[\mathbf{X_1^n} = \mathbf{x_1^n}] \log \frac{1}{Pr[\mathbf{x_1^n} = \mathbf{x_1^n}]}$$

, όπου $\mathbf{x_1^n} = x_1, ..., x_n$

Επίσης Ορίζουμε μία ποσότητα σχετική με τον ρυθμό εντροπία την $H'(\mathcal{X})$ ως:

$$H'(\mathcal{X}) = \lim_{n \to \infty} H(X_n | X_1, \cdots, X_{n-1})$$

Αφού διατυπώσαμε τα μέτρα πληροφορίας που αναλογούν σε μία πηγή με μνήμη, μπορούμε να βρούμε πως προσεγγίζουμε πειραματικά τέτοια μέτρα.

Θεώρημα 2.4. *Έστω μία πηγή X με μνήμη μεγέθους k, η οποία είναι στάσιμη. Τότε θα ισχύει:*

$$H'(\mathcal{X}) \lim_{n \to \infty} H(X_n | X_1, ..., X_{n-1}) = H(X_{k+1} | X_1, ..., X_k) = H^k(\mathcal{X}), \tag{2.11}$$

Απόδειξη

Έστω ότι έχουμε στα χέρια μας μία πηγή με μνήμη αγνώστου μεγέθους. Γνωρίζουμε ότι για οποιοδήποτε k υπάρχει η ποσότητα $H(X_n|X_{n-k},...,X_{n-1})$. Επίσης από το θεώρημα 1.14 ξέρουμε ότι το δεσμευμένο μέτρο πληροφορίας είναι μονότονο, δηλαδή

$$H(X_n|X_1,...,X_{n-1}) \leq H(X_n|X_2,...,X_{n-1}) \leq ... \leq H(X_n|X_{n-k},...,X_{n-1}) \leq ... \leq H(X_n) \tag{2.12}$$

Επειδή όμως η πηγή έχει μνήμη, με βάση τον ορισμό τους για οποιοδήποτε k θα ισχύει:

$$Pr[X_n = x_n|X_1 = x_1, \cdots, X_{n-1} = x_{n-1}] = Pr[X_n = x_n|X_{n-k} = x_{n-k}, \cdots X_{n-1} = x_{n-1}]$$

,

οπότε:

$$H(X_n|X_i,...,X_{n-1}) = H(X_n|X_{n-k},...,X_{n-1}) \ \forall i > k \tag{2.13}$$

Επειδή όμως η πηγή είναι στάσιμη θα ισχύει ακόμη ότι:

$$H(X_{n+1}|X_2,...,X_n) = H(X_n|X_1,...,X_{n-1} = x_{n-1}) \tag{2.14}$$

Χρησιμοποιώντας τις εξισώσεις (2.12) και (2.14) που προέκυψαν από τη μονοτονία του δεσμευμένου μέτρου πληροφορίας και την στασιμότητα της αλυσίδας καταλήγουμε στην εξής ανισοτική σχέση:

$$\begin{aligned}&H(X_{n+1}|X_1,...,X_n) \leq H(X_{n+1}|X_2,...,X_n) = H(X_n|X_1,...,X_{n-1}) \leq H(X_n|X_2,...,X_{n-1}) = H(X_{n-1}|X_1,...,X_{n-2}) \\ &\leq .... \leq H(X_2|X_1) \leq H(X_1) \Rightarrow \\ &H(X_{n+1}|X_1,...,X_n) \leq H(X_n|X_1,...,X_{n-1}) \leq H(X_{n-1}|X_1,...,X_{n-2}) \leq ... \leq H(X_2|X_1) \leq H(X_1)\end{aligned} \tag{2.15}$$

Βλέπουμε λοιπόν ότι δημιουργείτε μία φθίνουσα ακολουθία αριθμών η οποία είναι φραγμένη καθώς κάθε ποσό πληροφορίας που συμμετέχει στην ανισότητα είναι μεγαλύτερο ή ίσο του μηδενός (≥ 0) και μικρότερο της $H(X_1)$, άρα η ακολουθία θα συγκλίνει ως μονότονη και φραγμένη.

Το όριο της ακολουθίας, σύμφωνα με τη σχέση 2.13 και 2.14 θα είναι η $k-$οστή δεσμευμένη εντροπία $H(X_n|X_{n-k},...,X_{n-1})$, η οποία λόγω της στασιμότητας της πηγής θα είναι ίση με $H(X_{k+1}|X_1,...,X_k)$. Άρα

$$H^k(\mathcal{X}) = H(X_{k+1}|X_1,...,X_k) = \lim_{n\to\infty} H(X_n|X_1,...,X_{n-1}) = H'(\mathcal{X})$$

Το ζητούμενο όμως για να βρούμε την πληροφορία που παράγει η πηγή είναι να δούμε αν συγκλίνει ο ρυθμός εντροπίας. Το τελευταίο λοιπόν βήμα για να αποδείξουμε ότι για μία πηγή μνήμης k ο ρυθμός εντροπίας συγκλίνει στην εντροπία της πηγής $H(\mathcal{X})$ είναι να εκφράσουμε την ποσότητα $H(X_1 \cdots, X_n)$ συναρτήσει των δεσμευμένων εντροπιών χρησιμοποιώντας τον κανόνα της αλυσίδας.

Θεώρημα 2.5. *Έστω μία πηγή X με μνήμη μεγέθους k, η οποία είναι στάσιμη. Τότε θα ισχύει:*

$$H(\mathcal{X}) = \lim_{n\to\infty} \frac{H(X_1,...,X_n)}{n} = H^k(\mathcal{X}) = H'(\mathcal{X}) \tag{2.16}$$

Απόδειξη

Από τον κανόνα της αλυσίδας γνωρίζουμε:

$$\begin{aligned}&\frac{H(X_1,...,X_n)}{n} = \frac{1}{n}\sum_{i=1}^{n} H(X_i|X_1,...,X_{i-1}) \Rightarrow \\ &\lim_{n\to\infty} \frac{H(X_1,...,X_n)}{n} = \lim_{n\to\infty} \frac{1}{n}\sum_{i=1}^{n} H(X_i|X_1,...,X_{i-1})\end{aligned} \tag{2.17}$$

Λήμμα 2.1. *(Μέσος του Cesaro) Αν για μία ακολουθία ισχύει $a_n \to a$ και $b_n = \frac{1}{n}\sum_{i=1}^{n} a_n$ τότε θα ισχύει και $b_n \to a$*

Από το θεώρημα 2.4 γνωρίζουμε ότι

$$H'(\mathcal{X}) = \lim_{n\to\infty} H(X_n|X_1, ..., X_{n-1}) = H(X_{k+1}|X_1, ..., X_k) = H^k(\mathcal{X}),$$

Άρα χρησιμοποιώντας τη παραπάνω σχέση και το λήμμα 2.1 καταλήγουμε ότι

$$\lim_{n\to\infty} \frac{1}{n}\sum_{i=1}^{n} H(X_i|X_1, ..., X_{i-1}) = H^k(\mathcal{X})$$

Άρα:

$$H(\mathcal{X}) = \frac{H(X_1, ..., X_n)}{n} = \frac{1}{n}\sum_{i=1}^{n} H(X_i|X_1, ..., X_{i-1}) \to H^k(\mathcal{X}) = H'(\mathcal{X})$$

Άρα για μία στάσιμη πηγή με μνήμη μεγέθους k $H'(\mathcal{X}) = H(\mathcal{X}) = H(X_{k+1}|X_1, \cdots, X_k)$

Για το θεώρημα τις ασυμπτωτικής ισοκατανομής για πηγές με μνήμη, είναι αναγκαίο να εισάγουμε μια επιπρόσθετη σύμβαση. Η στασιμότητα επέβαλε οι από κοινού πιθανότητες της κάθε κατάστασης που μπορεί να βρεθεί η αλυσίδα να μην αλλάζουν με την πάροδο του χρόνου. Επειδή κατά την διαδικασία της δειγματοληψίας δεν γνωρίζουμε σε ποια κατάσταση θα είναι η μαρκοβιανή πηγή, είναι λογικό να υποθέτουμε ότι η κατάσταση από την οποία ξεκίνησε η πηγή δεν θα πρέπει να επηρεάζει την εξέλιξη της. Επίσης για του ίδιους δειγματοληπτικούς λόγους υποθέτουμε ότι η πηγή θα πρέπει να έχει πρόσβαση σε όλες τις καταστάσεις από όλες τις υπόλοιπες, δηλαδή θα μπορεί να επιστρέφει σε κάθε κατάσταση άπειρες φορές. Αναδύεται λοιπόν με φυσιολογικό τρόπο ότι οι πηγές που θα εξετάσουμε πέρα από στάσιμες θα πρέπει να είναι και εργοδικές.

Θεώρημα 2.6. ***(Θεώρημα της ασυμπτωτικής ισοκατανομής για πηγές με μνήμη*** *(Shannon-McMillan-Breimman Theorem)*[3] *)*

Έστω μία μια πηγή με μνήμη X μεγέθους k, η οποία είναι στάσιμη και εργοδική. Αν $H(\mathcal{X})$ είναι ο ρυθμός εντροπίας της πηγής τότε

$$\frac{1}{n}\log\frac{1}{Pr[X_1 = x_1, \cdots, X_n = x_n]} = \frac{1}{n}\log\frac{1}{Pr[\mathbf{X_1^n} = \mathbf{x_1^n}]} \overset{\sigma.\beta}{\to} H(\mathcal{X}) \tag{2.18}$$

Απόδειξη

Από τον πολλαπλασιαστικό νόμο των πιθανοτήτων γνωρίζουμε ότι:

$$Pr[X_1 = x_1, \cdots, X_n = x_n] = Pr[\mathbf{X_1^n} = \mathbf{x_1^n}] = \prod_{i=1}^{n} Pr[X_{i+1} = x_{i+1}|\mathbf{X_1^i} = \mathbf{x_1^i}].$$

Αν έχουμε μία πηγή με μνήμη k που μοντελοποιείται από μία μαρκοβιανή αλυσίδα $k-$τάξης τότε ο πολλαπλασιαστικός νόμος γίνεται:

$$Pr^k[X_1 = x_1, \cdots, X_n = x_n] = Pr[X_1 = x_1, \cdots X_k = x_k] \cdot Pr[X_{k+1} = X_{k+1}|X_1 = x_1, \cdots X_k = x_k] \cdot$$
$$Pr[X_{k+2} = X_{k+2}|X_2 = x_2, \cdots X_{k+1} = x_{k+1}] \cdots Pr[X_n = x_n|X_{n-1} = x_{n-1}, \cdots X_{n-k} = x_{n-k}]$$

Άρα με το συμβολισμό που έχουμε συμφωνήσει η παραπάνω σχέση γράφεται ως

[3] AC88.

$$Pr^k[\mathbf{X_1^n} = \mathbf{x_1^n}] = Pr[\mathbf{X_1^k} = \mathbf{x_1^k}] \cdot Pr[X_{k+1} = X_{k+1}|\mathbf{X_1^k} = \mathbf{x_1^k}] \cdots Pr[X_n = x_n|\mathbf{X_{n-k}^{n-1}} = \mathbf{x_{n-k}^{n-1}}] \Rightarrow$$
$$Pr^k[\mathbf{X_1^n} = \mathbf{x_1^n}] = Pr[\mathbf{X_1^k} = \mathbf{x_1^k}] \prod_{i=k}^{n} Pr[X_{i+1} = x_{i+1}|X_{i-k+1}^i = x_{i-k+1}^i]$$

Άρα η ποσότητα $\log \frac{1}{Pr[\mathbf{X_1^k} = \mathbf{x_1^k}]} = \log \frac{1}{\prod_{i=1}^{n} Pr[X_{i+1} = x_{i+1}|\mathbf{X_1^i} = \mathbf{x_1^i}]} = \sum_{i=1}^{n} \log \frac{1}{Pr[X_{i+1} = x_{i+1}|\mathbf{X_1^i} = \mathbf{x_1^i}]}$.
Αυτό που θα προσπαθήσουμε να κάνουμε στην απόδειξη είναι να φράξουμε την ποσότητα
$\frac{1}{n}\sum_{i=1}^{n} \log \frac{1}{Pr[X_{i+1} = x_{i+1}|\mathbf{X_1^i} = \mathbf{x_1^i}]}$ *από τις ποσότητες* $\frac{1}{n}\sum_{i=1}^{n} \log \frac{1}{Pr^k[\mathbf{X_1^n} = \mathbf{x_1^n}]}$ *και*
$\frac{1}{n}\sum_{i=1}^{n} \log \frac{1}{Pr[X_{i+1} = x_{i+1}|\mathbf{X_{-\infty}^i} = \mathbf{x_{-\infty}^i}]}$.
Η διαδικασία αυτή ισοδυναμεί με το να προσεγγίσουμε την εργοδική και στάσιμη πηγή $\{X_n\}_{n\in\mathbb{N}}$ *με πηγές μνήμης* k *καθώς και πηγές άπειρης μνήμης. Προφανώς ο ρυθμός εντροπία της πηγής* X *θα είναι ανάμεσα στον ρυθμός εντροπίας μία πηγής μνήμης* k *και μιας πηγής άπειρης μνήμης. Έπειτα θα δείξουμε ότι οι ρυθμοί εντροπίας μία πηγής με μνήμη* k *και άπειρη μνήμη συμπίπτουν για μία εργοδική και στάσιμη πηγή οπότε από ένα απλό κριτήριο παρεμβολής θα έχουμε το ζητούμενο.*

Λήμμα 2.2. *Για μία στάσιμη εργοδική πηγή* $X = \{X_n\}_{n\in\mathbb{N}}$ *ισχύει ότι:*

$$\log \frac{1}{Pr^k[\mathbf{X_1^n} = \mathbf{x_1^n}]} \overset{\sigma,\beta}{\to} H^k(X)$$
$$\log \frac{1}{Pr[\mathbf{X_1^n} = \mathbf{x_1^n}|\mathbf{X_{-\infty}^0} = \mathbf{x_{-\infty}^0}]} \overset{\sigma,\beta}{\to} H^\infty(X)$$

Απόδειξη
Επειδή οι συναρτήσεις εργοδικών διαδικασιών αποτελούν εργοδικές διαδικασίες έπεται ότι οι $\log Pr[X_{n+1} = x_{n+1}|\mathbf{X_{n-k}^n} = \mathbf{X_{n-k}^n}]$ *και* $\log Pr[X_n = x_n|\mathbf{X_{-\infty}^{n-1}} = \mathbf{x_{-\infty}^{n-1}}]$ *θα είναι εργοδικές.*

$$\frac{1}{n}\log \frac{1}{Pr^k[\mathbf{X_1^n} = \mathbf{x_1^n}]} = \frac{1}{n}\log \frac{1}{Pr[\mathbf{X_1^k} = \mathbf{x_1^k}]\prod_{i=k}^{n} Pr[X_{i+1} = x_{i+1}|\mathbf{X_{i-k+1}^i} = \mathbf{x_{i-k+1}^i}]} =$$
$$\frac{1}{n}\log \frac{1}{Pr[\mathbf{X_1^k} = \mathbf{x_1^k}]} + \frac{1}{n}\log \frac{1}{\prod_{i=k}^{n} Pr[X_{i+1} = x_{i+1}|\mathbf{X_{i-k+1}^i} = \mathbf{x_{i-k+1}^i}]} =$$
$$\frac{1}{n}\log \frac{1}{Pr[\mathbf{X_1^k} = \mathbf{x_1^k}]} + \frac{1}{n}\sum_{i=k}^{n} \log \frac{1}{Pr[X_{i+1} = x_{i+1}|\mathbf{X_{i-k+1}^i} = \mathbf{x_{i-k+1}^i}]}$$

Επειδή η πηγή είναι στάσιμη ξέρουμε ότι η από κοινού πιθανότητα $Pr[\mathbf{X_1^k} = \mathbf{x_1^k}]$ *δεν αλλάξει με τη πάροδο του χρόνου όποτε και η ποσότητα* $\log \frac{1}{Pr[\mathbf{X_1^k} = \mathbf{x_1^k}]}$ *δεν αλλάζει με την πάροδο του χρόνου. Άρα:*

$$\frac{1}{n}\log \frac{1}{Pr[\mathbf{X_1^k} = \mathbf{x_1^k}]} \overset{n\to\infty}{\to} 0$$

Επειδή η διαδικασία είναι εργοδική από το εργοδικό θεώρημα του Birkhoff ξέρουμε ότι ο δειγματικός μέσος της θα συγκλίνει στον θεωρητικό σχεδόν βεβαίως. Επειδή λοιπόν η διαδικασία $\log \frac{1}{Pr[X_{i+1} = x_{i+1}|\mathbf{X_{i-k+1}^i} = \mathbf{x_{i-k+1}^i}]}$ *είναι εργοδική έπεται ότι:*

$$\frac{1}{n}\log \frac{1}{Pr[\mathbf{X_1^k} = \mathbf{x_1^k}]} + \frac{1}{n}\sum_{i=k}^{n} \log \frac{1}{Pr[X_{i+1} = x_{i+1}|\mathbf{X_{i-k+1}^i} = \mathbf{x_{i-k+1}^i}]} \overset{n\to\infty}{\to} H^k(X)$$

Αντίστοιχα επειδή η διαδικασία $-\log Pr[X_n = x_n|\mathbf{X_{-\infty}^{n-1}} = \mathbf{x_{-\infty}^{n-1}}]$ *είναι εργοδική έπεται από το ίδιο εργοδικό θεώρημα ότι:*

$$\log \frac{1}{Pr[\mathbf{X_1^n} = \mathbf{x_1^n}|\mathbf{X_{-\infty}^0} = \mathbf{x_{-\infty}^0}]} = \frac{1}{n}\sum_{i=1}^{n} \log \frac{1}{Pr[X_{i+1} = x_{i+1}|\mathbf{X_{-\infty}^i} = \mathbf{x_{-\infty}^i}]} \overset{n\to\infty}{\to} H^{\infty}(X)$$

Το επόμενο βήμα είναι να δείξουμε ότι η εντροπία $k-$*τάξης και η εντροπία άπειρη τάξης συγκλίνουν και οι δύο στην εντροπία της πηγής* $H(\mathcal{X})$*. Αυτό το βήμα θα χρησιμεύσει ώστε στο κριτήριο παρεμβολής τα όρια των εργοδικών διαδικασιών με τα οποίες φράσσουμε τον δειγματικό μέσο μίας πηγής* X *να είναι ίδια.*

Λήμμα 2.3. *Η ρυθμός εντροπία* $k-$ *τάξης συγκλίνει στην εντροπίας της πηγής και στην εντροπία άπειρης τάξης.*

$$H^k(\mathcal{X}) \to H(\mathcal{X}) \text{ και } H^k(\mathcal{X}) \to H^{\infty}(\mathcal{X})$$

Επειδή η πηγή είναι στάσιμη από το θεώρημα 2.4 ξέρουμε ότι ο ρυθμός εντροπία $k-$ *τάξης συγκλίνει στον ρυθμό εντροπίας της πηγής. Μένει να δείξουμε ότι* $H^k(X) \to H^{\infty}(X)$*. Από το θεώρημα σύγκλισης του Levy για martingales ξέρουμε ότι οι δεσμευμένες μέσες τιμές ισχύει* $E[X_i|\mathbf{X_{-k}^0}] \overset{\sigma.\beta}{\to} E[X_i|\mathbf{X_{-\infty}^0}]$ *άρα έπεται ότι:*

$$Pr[X_1 = x_1|\mathbf{X_k^0} = \mathbf{x_{-k}^0}] \overset{\sigma.\beta}{\to} Pr[X_1 = x_1|\mathbf{X_{-\infty}^0} = \mathbf{x_{-\infty}^0}]$$

Επειδή η συνάρτηση $p\log\frac{1}{p}$ *είναι φραγμένη και συνεχής* $\forall\ p \in [0,1]$ *και το σύνολο* $\mathcal{X}$ *πεπερασμένο από το θεώρημα της φραγμένης σύγκλισης ισχύει ότι*

$$H^k(X) = \lim_{k\to\infty} E[\sum_{x_1\in\mathcal{X}} Pr[X_1 = x_1|\mathbf{X_k^0} = \mathbf{x_{-k}^0}]] \overset{\text{Θεώρημα φραγμένης σύγκλισης}}{=} E[\lim_{k\to\infty} \sum_{x_1\in\mathcal{X}} Pr[X_1 = x_1|\mathbf{X_k^0} = \mathbf{x_{-k}^0}]] =$$
$$E[\sum_{x_1\in\mathcal{X}} Pr[X_1 = x_1|\mathbf{X_{-\infty}^0} = \mathbf{x_{-\infty}^0}]] = H^{\infty}$$

Άρα αφού η εντροπία $k-$ *τάξης συγκλίνει στην* H *και την* H^{∞} *έπεται από την μοναδικότητα του ορίου ότι η* H^{∞} *θα συγκλίνει στην* H*.*

Λήμμα 2.4. *(φράγματα του κριτηρίου παρεμβολής)*

$$\lim_{n\to\infty} sup\frac{1}{n}\log\frac{Pr^k[\mathbf{X_1^n} = \mathbf{x_1^n}]}{Pr[\mathbf{X_1^n} = \mathbf{x_1^n}]} \le 0$$

και

$$\lim_{n\to\infty} sup\frac{1}{n}\log\frac{Pr[\mathbf{X_1^n} = \mathbf{x_1^n}]}{Pr[\mathbf{X_1^n} = \mathbf{x_1^n}|\mathbf{X_{-\infty}^0} = \mathbf{x_{-\infty}^0}]} \le 0$$

Απόδειξη
Έστω A *το στήριγμα του* $Pr[\mathbf{X_1^n} = \mathbf{x_1^n}]$*, τότε*

$$E\left[\frac{Pr^k[\mathbf{X_1^n} = \mathbf{x_1^n}]}{Pr[\mathbf{X_1^n} = \mathbf{x_1^n}]}\right] = \sum_{\mathbf{x_1^n}\in\mathbf{A}} Pr[\mathbf{X_1^n} = \mathbf{x_1^n}]\frac{Pr^k[\mathbf{X_1^n} = \mathbf{x_1^n}]}{Pr[\mathbf{X_1^n} = \mathbf{x_1^n}]} = \sum_{\mathbf{x_1^n}\in\mathbf{A}} Pr^k[\mathbf{X_1^n} = \mathbf{x_1^n}] = Pr^k(A) \le 1$$

Αντίστοιχα έστω το στήριγμα B *του* $Pr[.|\mathbf{X_{-\infty}^0} = \mathbf{x_{-\infty}^0}]$*, τότε*

$$E\left[\frac{Pr[\mathbf{X_1^n}=\mathbf{x_1^n}]}{Pr[\mathbf{X_1^n}=\mathbf{x_1^n}|\mathbf{X_{-\infty}^0}=\mathbf{x_{-\infty}^0}]}\right]=E\left[E\left[\frac{Pr[\mathbf{X_1^n}=\mathbf{x_1^n}]}{Pr[\mathbf{X_1^n}=\mathbf{x_1^n}|\mathbf{X_{-\infty}^0}=\mathbf{x_{-\infty}^0}]}\middle|\mathbf{X_{-\infty}^0}=\mathbf{x_{-\infty}^0}\right]\right]=$$
$$=E[\sum_{\mathbf{x_1^n}\in B}\frac{Pr[\mathbf{X_1^n}=\mathbf{x_1^n}]}{Pr[\mathbf{X_1^n}=\mathbf{x_1^n}|\mathbf{X_{-\infty}^0}=\mathbf{x_{-\infty}^0}]}Pr[\mathbf{X_1^n}=\mathbf{x_1^n}|\mathbf{X_{-\infty}^0}=\mathbf{x_{-\infty}^0}]=E[\sum\mathbf{x_1^n}\in BPr[\mathbf{X_1^n}=\mathbf{x_1^n}]]\leq 1$$

Από την ανισότητα Markov γνωρίζουμε ότι $Pr[|X|\geq a]\leq\frac{E[X]}{a}$ *θα έχουμε ότι:*

$Pr[\frac{Pr^k[\mathbf{X_1^n}=\mathbf{x_1^n}]}{Pr[\mathbf{X_1^n}=\mathbf{x_1^n}]}\geq a]\leq\frac{1}{a}$, *άνάλογα θα έχουμε* $Pr[\frac{1}{n}\log\frac{Pr^k[\mathbf{X_1^n}=\mathbf{x_1^n}]}{Pr[\mathbf{X_1^n}=\mathbf{x_1^n}]}\geq\frac{1}{n}\log a]\leq\frac{1}{a}$. *Αφού λοιπόν οι πιθανότητες φράσσονται από μία ποσότητα που τείνει στο μηδέν, έπεται ότι το άθροισμα τους θα συγκλίνει οπότε από Borel Cantelli θα έχουμε το πρώτο ζητούμενο. Το δεύτερο αποδεικνύεται ανάλογα.*

Τώρα ήρθε η ώρα της απόδειξη τους ζητούμενου. Από την στιγμή που το
$\lim_{n\to\infty} sup\frac{1}{n}\log\frac{1}{Pr[\mathbf{X_1^n}=\mathbf{x_1^n}]}\leq\lim_{n\to\infty}\frac{1}{n}\log\frac{1}{Pr^k[\mathbf{X_1^n}=\mathbf{x_1^n}]}=H^k$

Αυτό ισχύει γιατί από την στιγμή που το $\lim_{n\to\infty} sup\frac{1}{n}\log\frac{Pr^k[\mathbf{X_1^n}=\mathbf{x_1^n}]}{Pr[\mathbf{X_1^n}=\mathbf{x_1^n}]}\leq 0\Rightarrow\frac{Pr^k[\mathbf{X_1^n}=\mathbf{x_1^n}]}{Pr[\mathbf{X_1^n}=\mathbf{x_1^n}]}\leq 1\Rightarrow$
$\frac{1}{Pr^k[\mathbf{X_1^n}=\mathbf{x_1^n}]}\geq\frac{1}{Pr[\mathbf{X_1^n}=\mathbf{x_1^n}]}$ *για μεγάλα* n.

Για τους ίδιους λόγους από την δεύτερη σχέση που αποδεικνύεται στο τελευταίο λήμμα ισχύει
$\lim_{n\to\infty} inf\frac{1}{n}\log\frac{1}{Pr[\mathbf{X_1^n}=\mathbf{x_1^n}]}\geq\lim_{n\to\infty}\frac{1}{n}\log\frac{1}{Pr^k[\mathbf{X_1^n}=\mathbf{x_1^n}|\mathbf{X_{-\infty}^0}=\mathbf{x_{-\infty}^0}]}=H^\infty$ *άρα επι της ουσία καταφέραμε να φράξουμε το* $\frac{1}{n}\log\frac{1}{Pr[\mathbf{X_1^n}=\mathbf{x_1^n}]}$ *από πάνω και κάτω με τις τιμές* H^k *και* H^∞. *Όμως από το δεύτερο λήμμα* $H^k\to H^\infty$, *άρα το* $\frac{1}{n}\log\frac{1}{Pr[\mathbf{X_1^n}=\mathbf{x_1^n}]}\to H$.

Επειδή η διατύπωση είναι λίγο διαφορετική σε σχέση με το θεώρημα ασυμπτωτικής ισοκατανομής για πηγές χωρίς μνήμη είναι εύκολο να φέρουμε την διατύπωση του θεωρήματος Shannon-McMillan-Breimman στα πλαίσια της προηγούμενης αν σκεφτούμε τι σημαίνει η σχεδόν βεβαίως σύγκλιση.

Ορισμός 2.7. *Η ακολουθία τυχαίων μεταβλητών* $\{X_n\}_{n\in\mathbb{N}}$ *λέγεται ότι συγκλίνει* ***σχεδόν βεβαίως ή με πιθανότητα 1*** *στην τ.μ* X *και συμβολίζεται* $X_n\overset{\sigma.\beta}{\to}X$ *όταν:*

$$Pr[\omega\in\Omega:\lim_{n\to\infty}X_n(\omega)=X(\omega)]=1 \tag{2.19}$$

Στην ουσία η σχεδόν βεβαίως σύγκλιση μας λέει ότι υπάρχει ένα σύνολο $\Omega_0\subset\Omega$ το οποίο συγκεντρώνει σχεδόν όλη την μάζα πιθανότητα. Άρα όταν στο θεώρημα Shannon-McMillan-Breimman διατυπώνεται ότι $\frac{1}{n}\log\frac{1}{Pr[\mathbf{X_1^n}=\mathbf{x_1^n}]}\overset{\sigma.\beta}{\to}H(\mathcal{X})$ σημαίνει πως αν μαζέψουμε στο τυπικό σύνολο A^n όλες τις ακολουθίες που συγκλίνουν σχεδόν βεβαίως στην εντροπία τότε η μάζα πιθανότητας αυτού του συνόλου θα τείνει στο 1 και αντίστοιχα η μάζα πιθανότητα του συμπληρώματος θα τείνει στο 0. Άρα επαναδιατυπώνοντας μπορούμε να πούμε:

Θεώρημα 2.7. *(**Θεώρημα Ασυμπτωτικής Ισοκατανομής για πηγές με μνήμη**) Έστω μία στάσιμη και εργοδική πηγή. Τότε υπάρχει ένα* $N_0\in\mathbb{N}$ *τέτοιο ώστε για κάθε* $\epsilon>0$ *ορίζουμε το σύνολο:*

$$A_\epsilon^n=\{\mathbf{x_1^n}:|\frac{1}{n}\log\frac{1}{Pr[\mathbf{X_1^n}=\mathbf{x_1^n}]}-H(\mathcal{X})|<\epsilon,\ \forall n\geq N_0\}$$

Τότε:

1. $Pr[A_\epsilon^n] \overset{n\to\infty}{\to} 1$
2. $Pr[\{A_\epsilon^n\}^c] \overset{n\to\infty}{\to} 0$

Τέλος ακριβώς με τα ίδια επιχειρήματα που χρησιμοποιήθηκαν στο θεώρημα 2.3 αποδεικνύεται και το αντίστοιχο για πηγές με μνήμη.

2.4 Κώδικες για τις τεχνητές γλώσσες

Letter_Sources.py

```
1
2  import re
3  import numpy
4
5
6  class First_Order_Source:
7      source_dict = None
8      '''Constructor of Class. Give the constructor the path of the file you want to use as source'''
9      def __init__(self, path):
10         self.source_dict = dict()
11         file = open(path, 'r')
12         line = file.readline()
13         while line:
14             #Clean line from punctuation
15             free_line = re.sub(r'[^\w\s]', '', line)
16             free_line = re.sub(r'[\d]', '', free_line)
17             free_line = re.sub(r'[\n]*$', '', free_line)
18             '''For every character in line check if it is in
19             the dictionary. If it's in update the frequency
20             else make a new entry with frequency 1.'''
21             for ch in free_line:
22                 if ch in self.source_dict:
23                     self.source_dict[ch] = self.source_dict[ch] + 1
24                 else:
25                     self.source_dict[ch] = 1
26             line = file.readline()
27
28         file.close()
29         print(self.source_dict)
30         #find the total number of symbols in text
31         sum = 0
32         for value in self.source_dict.values():
33             sum = sum + value
34         dist = []
35         alphabet = []
36         '''Build the distribution of the alphabet dividing the letter frequency with the total frequency'''
37         dist_source = dict()
38         for key in sorted(self.source_dict.keys()):
39             dist.append(float(self.source_dict[key] / sum))
40             alphabet.append(key)
41         '''Build the source using the type Pr[X_1,..., X_n]=Pr[X_1]x...xPr[X_n]'''
42         for i in range(0, 3):
43             for j in range(0, 50):
44                 x = numpy.random.choice(numpy.arange(0, 116), p=dist)
45                 print(alphabet[x], end='')
46             print()
47
48 class Second_Order_Source:
49     source_dict = None
50     source_dict = None
51     '''Constructor of Class. Give the constructor the path of the file you want to use as source'''
52     def __init__(self, path):
53         self.source_dict = dict()
54         file = open(path, 'r')
55         line = file.readline()
56         last_letter =None
57         while line:
58             #Clean line from punctuation
59             free_line = re.sub(r'[^\w\s]', '', line)
```

```
60                free_line  = re.sub(r'[\d]', '',  free_line )
61                free_line  = re.sub(r'[\n]*$', '',  free_line )
62
63                if  last_letter  is not None:
64                    free_line = last_letter + free_line
65                ''' For every character in line check if it is in
66                    the dictionary. If it's in check in the next
67                    character is in current charachter's dictionary.
68                    If it's in update the frequency of the next character
69                    else insert an entry for the next character with
70                    frequency 1. If the current character is not in dictionary
71                    make a new entry puting the charachter in dictionary
72                    with value a dictionary consisting of the next
73                    charachter with frequency 1. '''
74                for i in range(0, len( free_line ) - 1):
75                    first  = free_line [i]
76                    second = free_line [i + 1]
77                    if  first  in  self . source_dict :
78                        if second not in  self . source_dict [ first ]:
79                            self . source_dict [ first ][second] = 1
80                        else :
81                            self . source_dict [ first ][second] = self . source_dict [ first ][second] + 1
82                    else :
83                        self . source_dict [ first ] = {second: 1}
84                ''' Save the last  letter  in order to concatinate  it
85                  with the the next line '''
86                last_letter =free_line [len( free_line )-1]
87                line  = file . readline ()
88
89            file . close ()
90            ''' create the distribution of the alphabet and
91            for the alphabets of the next characters of every
92            character in the general alphabet '''
93            dist  = []
94            alphabet = []
95            dist_2 = []
96            alphabet_2 = []
97            total_freq  = 0
98            for key in sorted( self . source_dict . keys ()):
99                alphabet.append(key)
100               freq  = (sum(self.source_dict [key]. values ()))
101               total_freq  = total_freq  + freq
102               current_dist_2 = []
103               current_alphabet_2 = []
104               for key_2 in sorted( self . source_dict [key]. keys ()):
105                   current_dist_2 .append(float( self . source_dict [key][key_2]  / freq))
106                   current_alphabet_2.append(key_2)
107               dist_2.append(current_dist_2)
108               alphabet_2.append(current_alphabet_2)
109           for k in sorted( self . source_dict . keys ()):
110               freq  = (sum(self.source_dict [k]. values ()))
111               dist . append(float(freq  /  total_freq ))
112
113           ''' Generate the source following the rule:
114           Pr[X_1 ,...., X_n]=Pr[X_1]xPr[X_2|X_1]x...xPr[X_n|X_{n-1}]'''
115           x = numpy.random.choice(numpy.arange(0, len(dist)), p=dist)
116           current_prob = x
117           print (alphabet[x], end='')
118
119           for j in range(0, 3):
120               i = 0
121               while i < 50:
122                   next_prob = numpy.random.choice(numpy.arange(0, len(dist_2[current_prob])), p=dist_2[current_prob])
123                   print (alphabet_2[current_prob][next_prob], end='')
124                   current_symbol = alphabet_2[current_prob][next_prob]
125                   current_prob = alphabet.index(current_symbol)
126                   i = i + 1
127               if j == 1:
128                   next_prob = numpy.random.choice(numpy.arange(0, len(dist_2[current_prob])), p=dist_2[current_prob])
129                   print (alphabet_2[current_prob][next_prob], end='')
130               print ()
131
132
133   class Third_Order_Source:
134       source_dict=None
```

```
    '''Constructor of Class. Give the constructor the path of the file you want to use as source'''
    def __init__(self,path):
        self.source_dict = dict()
        file = open(path, 'r')
        line = file.readline()
        last_block=None
        while line:
            #Clean line from punctuation
            free_line = re.sub(r'[^\w\s]', '', line)
            free_line = re.sub(r'[\d]', '', free_line)
            free_line = re.sub(r'[\n]*$', '', free_line)
            if last_block is not None:
                free_line = last_block + free_line
            '''For every digram in line check if it is in
              the dictionary. If it's in check in the next
              character is in current digram's dictionary.
              If it's in update the frequency of the next character else
              insert an entry for the next character with
              frequency 1. If the current digram is not in dictionary
              make a new entry puting the digram in dictionary
              with value a dictionary consisting of the next
              charachter with frequency 1. '''
            for i in range(0, len(free_line) - 2, 1):
                block = free_line[i] + free_line[i + 1]
                next_letter = free_line[i + 2]
                if block in self.source_dict:
                    if next_letter not in self.source_dict[block]:
                        self.source_dict[block][next_letter] = 1
                    else:
                        self.source_dict[block][next_letter] = self.source_dict[block][next_letter] + 1
                else:
                    self.source_dict[block] = {next_letter: 1}
            '''Save the last block in order to concatinate it
            with the the next line '''
            last_block = free_line[len(free_line) - 2] + free_line[len(free_line) - 1]
            line = file.readline()

        file.close()
        print(self.source_dict)

        dist = []
        alphabet = []
        dist_2 = []
        alphabet_2 = []
        total_freq = 0
        for key in sorted(self.source_dict.keys()):
            alphabet.append(key)
            freq = (sum(self.source_dict[key].values()))
            total_freq = total_freq + freq
            current_dist_2 = []
            current_alphabet_2 = []
            for key_2 in sorted(self.source_dict[key].keys()):
                current_dist_2.append(float(self.source_dict[key][key_2] / freq))
                current_alphabet_2.append(key_2)
            dist_2.append(current_dist_2)
            alphabet_2.append(current_alphabet_2)
        for k in sorted(self.source_dict.keys()):
            freq = (sum(self.source_dict[k].values()))
            dist.append(float(freq / total_freq))
        print(dist)

        s = ''
        x = numpy.random.choice(numpy.arange(0, len(dist)), p=dist)
        current_prob = x
        s = s + alphabet[x]

        for j in range(0, 3):
            i = 0
            k = 0
            while i < 50:
                next_prob = numpy.random.choice(numpy.arange(0, len(dist_2[current_prob])), p=dist_2[current_prob])
                current_symbol = alphabet_2[current_prob][next_prob]
                s = s + current_symbol
                next_symbol = s[len(s) - 2:len(s)]
                current_prob = alphabet.index(next_symbol)
```

```
                i = i + 1

            if j == 2:
                next_prob = numpy.random.choice(numpy.arange(0, len(dist_2[current_prob])), p=dist_2[current_prob])
                current_symbol = alphabet_2[current_prob][next_prob]
                s = s + current_symbol
            print(s[len(s) - 52:len(s)])
```

Word_Sources.py

```
''' It has the same logic with Letter_Sources. The only thing
that is different ia after removing the punctuation we
 split the word when a space is encountered.
 The words are saved in a vector: vector_line =free_line.split(' ') '''


import re
import numpy


class First_Order_Word_Source:
    source_dict = None

    def __init__(self, path):
        self.source_dict = dict()
        file = open(path, 'r')
        line = file.readline()
        while line:
            free_line = re.sub(r'[^\w\s]', '', line)
            free_line = re.sub(r'[\d]', '', free_line)
            free_line = re.sub(r'[\n]*$', '', free_line)
            vector_line =free_line.split(' ')
            for ch in vector_line:
                if ch in self.source_dict:
                    self.source_dict[ch] = self.source_dict[ch] + 1
                else:
                    self.source_dict[ch] = 1
            line = file.readline()
        sum = 0
        for value in self.source_dict.values():
            sum = sum + value
        dist = []
        alphabet = []
        for key in sorted(self.source_dict.keys()):
            dist.append((self.source_dict[key] / sum))
            alphabet.append(key)

        file.close()
        for i in range(0, 3):
            for j in range(0, 10):
                x = numpy.random.choice(numpy.arange(0, len(dist)), p=dist)
                print(alphabet[x], ' ', end='')
            print()


class Second_Order_Word_Source:
    source_dict = None

    def __init__(self, path):
        self.source_dict = dict()
        file = open(path, 'r')
        line = file.readline()
        last_word=None
        while line:
            free_line = re.sub(r'[^\w\s]', '', line)
            free_line = re.sub(r'[\d]', '', free_line)
            free_line = re.sub(r'[\n]*$', '', free_line)
            vector_line = free_line.split(' ')
            if last_word is not None:
                vector_line.insert(0,last_word)
            for i in range(0, len(vector_line)-1,1):
                first = vector_line[i]
                second = vector_line[i + 1]
                if first in self.source_dict:
```

```
                    if second not in self.source_dict[first]:
                        self.source_dict[first][second] = 1
                    else:
                        self.source_dict[first][second] = self.source_dict[first][second] + 1
                else:
                    self.source_dict[first] = {second: 1}

            last_word=vector_line[len(vector_line)-1]
            line = file.readline()

        file.close()

        dist = []
        alphabet = []
        dist_2 = []
        alphabet_2 = []
        total_freq = 0
        for key in sorted(self.source_dict.keys()):
            alphabet.append(key)
            freq = (sum(self.source_dict[key].values()))
            total_freq = total_freq + freq
            current_dist_2 = []
            current_alphabet_2 = []
            for key_2 in sorted(self.source_dict[key].keys()):
                current_dist_2.append(float(self.source_dict[key][key_2] / freq))
                current_alphabet_2.append(key_2)
            dist_2.append(current_dist_2)
            alphabet_2.append(current_alphabet_2)
        for k in sorted(self.source_dict.keys()):
            freq = (sum(self.source_dict[k].values()))
            dist.append(float(freq / total_freq))

        x = numpy.random.choice(numpy.arange(0, len(dist)), p=dist)
        current_prob = x
        print(alphabet[x], end='')

        for j in range(0, 3):
            i = 0
            while i < 10:
                next_prob = numpy.random.choice(numpy.arange(0, len(dist_2[current_prob])), p=dist_2[current_prob])
                print(alphabet_2[current_prob][next_prob],' ', end='')
                current_symbol = alphabet_2[current_prob][next_prob]
                current_prob = alphabet.index(current_symbol)
                i = i + 1
            if j == 1:
                next_prob = numpy.random.choice(numpy.arange(0, len(dist_2[current_prob])), p=dist_2[current_prob])
                print(alphabet_2[current_prob][next_prob],' ', end='')
            print()
```

Κεφάλαιο 3

Συμπίεση

3.1 Εισαγωγή

Η συμπίεση δεδομένων μπορούμε να πούμε ότι αποτελεί την τεχνολογική έκφανση της έκφρασης «Το Λακωνίζειν εστί Φιλοσοφείν». Η όλη φιλοσοφία της συμπίεσης έγκειται στην ανάγκη να αναπαραστήσουμε ένα σύνολο δεδομένων με τον πιο σύντομο τρόπο. Ας ανακαλέσουμε στην μνήμη μας την ερμηνεία της εντροπίας ως τον αριθμό που ποσοτικοποιεί την μέση ποσότητα πληροφορίας που περιέχεται μέσα σε μία τυχαία μεταβλητή. Μόνο από τις ερμηνείες των εννοιών της συμπίεσης και της εντροπίας καταλαβαίνουμε ότι θα συσχετίζονται με κάποιον τρόπο. Η σχέση αυτή υπαινίχθηκε ήδη από το παράδειγμα 1.1 του πρώτο κεφαλαίου, στο οποίο αναπαραστήσαμε τις τιμές των τυχαίων μεταβλητών X και Υ, με σ.μ.π $P_X(x) = \{\frac{1}{2},\ \frac{1}{2},\ \frac{1}{2},\ \frac{1}{2}\}$ και $P_Y(y) = \{\frac{1}{2}, \frac{1}{8},\ \frac{1}{8},\ \frac{1}{4}\}$ αντίστοιχα, με ακολουθίες από το αλφάβητο $\{0, 1\}$ δηλαδή με bits.

Πίνακας 3.1: Κώδικες για τις μεταβλητές X και Υ

Σύμβολο	Κώδικας για την X	Κώδικας για την Υ
a	11	1
b	10	011
c	01	010
d	00	00

Κατά το παράδειγμα αυτό είδαμε ότι το μέσο μήκος των κωδίκων που φτιάξαμε συνέπιπτε με την εντροπία των παραπάνω συναρτήσεων μάζας πιθανότητας. Η τυχαία μεταβλητή X είχε εντροπία $H(X) = 2bits/$σύμβολο ενώ η Υ είχε εντροπία $H(Y) = 1.75bits/$σύμβολο. Για να διερευνήσουμε λίγο περισσότερο τη σχέση της συμπίεσης με την εντροπία διαισθητικά, πριν αρχίσουμε να τη μελετούμε μαθηματικά, μπορούμε να σκεφτούμε το παρακάτω σενάριο:

Χρησιμοποιώντας τις παραπάνω σ.μ.π παράγουμε δύο κείμενα, δέκα γραμμών και 20 χαρακτήρων ανά γραμμή το κάθε ένα. Υποθέτουμε ότι τα κείμενα έχουν αποθηκευτεί σε αρχεία κειμένου με κωδικοποίηση ASCII. Σύμφωνα με την κωδικοποίηση αυτή η αναπαράσταση κάθε χαρακτήρα στον υπολογιστή χρειάζεται 8bits που ισοδυναμούν με 1 byte.

Πίνακας 3.2: Κείμενα που παράχθηκαν με βάση τις $f_{\mathcal{X}}(x),\ f_{\mathcal{Y}}(y)$.

Κείμενο με βάση την $f_{\mathcal{X}}(x)$	Κείμενο με βάση την $f_{\mathcal{Y}}(y)$
abccdbabcdaddcdbaadd	cadadaabaaaabadaddaa
bdacbaacdbdadccdbbab	dbaaaaadddcadaaabaaa
caabccbcacabdbccddcb	babababaaddddabaaacaaa
ccaadaccccacbbcaacda	adddacadccaabaaabaaa
bbddbdcaadbdbadbbadc	caacaaacdddcaaaabaad
acadabbaaabadcbddccd	dbaaacaabdbdaadaaaad
bbbdcccaaddacbddcacc	acacbaaaaabaaaaacaaa
daacadbcbdddcaaccadd	cadcaaadaaadddabbddc
bdcbabdbcbbdadacdaaa	aaaabaaaaaaaaabdcadda
bbddabdbbbbadbcdbcbc	cadababbcabdcaadcaab
[('a', 50), ('b', 50), ('c', 49), ('d', 51)]	[('a', 113), ('b', 25), ('c', 22), ('d', 40)]

Στην τελευταία γραμμή του κάθε κειμένου φαίνεται πόσες φορές χρησιμοποιήθηκε κάθε γράμμα μέσα στο κείμενο. Ας υπολογίσουμε πόσο χώρο θα καταλάβουν και τα δύο αρχεία στο υπολογιστή μας.

1. Το πρώτο κείμενο θα έχει μέγεθος $50 * 8 + 50 * 8 + 49 * 8 + 51 * 8 = 200 * 8 = 200$bytes.

2. Το δεύτερο κείμενο θα έχει μέγεθος $113 * 8 + 25 * 8 + 22 * 8 + 40 * 8 = 200 * 8 = 200$bytes.

Άρα καταλήγουμε στο συμπέρασμα ότι για να αποθηκεύσουμε ένα κείμενο σε κωδικοποίηση ASCII που περιέχει συνολικά 200 σύμβολα χρειαζόμαστε και στις δύο περιπτώσεις 200 bytes. Βλέπουμε ότι η κωδικοποίηση δεν λαμβάνει καθόλου υπόψιν της, τις συναρτήσεις μάζα πιθανότητας των τυχαίων μεταβλητών και έτσι αντιμετωπίζει τα δύο κείμενα ισάξια ως προς το ποσό πληροφορίας που περιέχουν.

Σύμφωνα με όσα έχουν ειπωθεί στο πρώτο κεφάλαιο καταλαβαίνουμε ότι τα δύο κείμενα δεν περιέχουν ίσα ποσά πληροφορίας, όποτε κάποιο μέρος από τα 200 bytes των αρχείων αποτελεί επί της ουσίας πλεόνασμα πληροφορίας. Για να βρούμε το πλήθος των επιπλέον bits που χρησιμοποιήθηκαν για την αναπαράσταση των κειμένων θα χρησιμοποιήσουμε τους κώδικες του πίνακα. Οι παραπάνω κώδικες αποτελούν την δυαδική έκφραση του ποσού της πληροφορίας που περιέχουν οι τιμές των τυχαίων μεταβλητών X και Υ.

Στον κώδικα της X κάθε κωδική λέξη χρειάζεται δύο bits για να αναπαρασταθεί άρα το ποσό της πληροφορίας που βρίσκεται στο πρώτο κείμενο είναι $50*2+50*2+49*2+51*2 = 200*2 = 400 = 50*8 = 50$bytes. Με αντίστοιχο τρόπο βρίσκουμε ότι το ποσό της πληροφορίας που βρίσκεται στο δεύτερο κείμενο είναι $119 * 1 + 25 * 3 + 22 * 3 + 40 * 2 = 340 = 42.5 * 8 = 42.5$bytes Από τους υπολογισμούς βλέπουμε ότι στην πρώτη περίπτωση έχουμε ένα πλεόνασμα πληροφορίας 150bytes και στη δεύτερη 157.5bytes. Καταλαβαίνουμε λοιπόν ότι αν το αρχείο αποθηκευτεί σε μορφή bits με τη χρήση κωδίκων που παίρνουν υπόψη τους τις στατιστικές ιδιότητες της πηγής, τότε θα έχουμε μια αποδοτικότερη εκμετάλλευση του διαθέσιμου χώρου.

Μπορεί στη σημερινή εποχή ο αποθηκευτικός χώρος να φαντάζει αμελητέα παράμετρος όσο αφορά στη χρήση του προσωπικού μας υπολογιστή ή κινητού, αλλά στο διαδίκτυο που μεταφέρονται τεράστιοι όγκοι δεδομένων, η αναπαράσταση τους στην πιο συμπυκνωμένη μορφή αποτελεί απαραίτητο όρο για να είναι η μετάδοση των δεδομένων όσο το δυνατόν πιο γρήγορη και να καθίσταται με αυτόν τον τρόπο η χρήση του διαδικτύου λειτουργική. Παραδείγματος χάρη για συμπίεση χωρίς απώλειες εικόνων στο διαδίκτυο χρησιμοποιούνται οι μορφές αρχείων Graphics Interchange Format (GIF) και Portable Network Graphics (PNG) οι οποίες περιέχουν τους αλγόριθμους συμπίεσης LZW και DEFLATE αντίστοιχα, ενώ για συμπίεση με απώλειες χρησιμοποιείται η μορφή αρχείου Joint Photographic Experts Group (JPEG) που κάνει χρήση ενός αλγορίθμου συμπίεσης βασισμένο στον διακριτό μετασχηματισμό συνημιτόνου. Για τη συμπίεση διάφορων αιτημάτων αλλά και απαντήσεων μεταξύ φυλλομετρητών ιστοσελίδων (browsers) και εξυπηρετητών (servers) χρησιμοποιούνται αλγόριθμοι συμπίεσης όπως ο DEFLATE μέσω του προγράμματος gzip και ο Brotli που αναπτύχθηκε από τη Google.

Όπως βλέπουμε ανάλογα με το τύπο του αρχείου που θέλουμε να συμπιέσουμε, λόγου χάρη εικόνα, κείμενο ή βίντεο, μπορούμε να διαλέξουμε από μία πληθώρα αλγορίθμων συμπίεσης. Μολονότι κάθε αλγόριθμος χρησιμοποιεί την δική του διαδικασία προκειμένου να συμπιέσει δεδομένα υπάρχει μια κοινή δομή που είναι ίδια για όλους. Καθένας τέτοιος αλγόριθμός συμπίεσης περιέχει ένα πρόγραμμα που μεταφράζει το αρχικό σύνολο δεδομένων στο συμπιεσμένο και ένα πρόγραμμα που μεταφράζει τα συμπιεσμένα δεδομένα στο αρχικό σύνολο ή σε κάποια προσέγγιση του. Η πρώτη διαδικασία λέγεται κωδικοποίηση(συμπίεση) και η δεύτερη αποκωδικοποίηση(αποσυμπίεση).

Ανάλογα με τον αν χάνεται πληροφορία από το αρχικό αρχείο προς συμπίεση κατά τη κωδικοποίηση ή όχι, οι αλγόριθμοι χωρίζονται σε δύο μεγάλες κατηγορίες, τους αλγόριθμους συμπίεσης χωρίς απώλειες(losless data compression) και τους αλγορίθμους συμπίεσης με απώλειες (lossy data compression). Μπορούμε να πούμε ότι η διαδικασία που γέννησε τους παραπάνω κώδικες και βασιζόταν στη διαμέριση του χώρου τιμών της τυχαίας μεταβλητής $\mathcal{X}$ όπως αναπτύχθηκε κατά την εισαγωγή του πρώτου κεφαλαίου, αποτελεί τη βάση για να δημιουργήσουμε ένα αλγόριθμο συμπίεσης κειμένου χωρίς απώλειες.

Ο αλγόριθμος αρχικά θα δημιουργεί τον κώδικά για το συγκεκριμένο αρχείο. Επειδή πολλές φορές δεν έχουμε τη σ.μ.π των συμβόλων που περιλαμβάνονται σε κάθε αρχείο μπορούμε να προσεγγίσουμε τις πιθανότητες τους υπολογίζοντας τη σχετική συχνότητα εμφάνισης τους στο κείμενο. Έπειτα έχοντας αυτή τη προσεγγιστική σ.μ.π μπορούμε να διαμερίσουμε αναδρομικά σε ισοπίθανα σύνολα τον χώρο πιθανότητας μέχρι να φτάσουμε σε μονοσύνολα. Από το δένδρο διαμέρισης είναι εύκολο να εξάγουμε τους κώδικες των συμβόλων

Εύρεση σ.μ.π

Είσοδος:Αρχικό_Κείμενο
μετρητής=0
λεξικό={}
πίνακας_πιθανοτήτων={}
Για κάθε σύμβολο στο κείμενο:
    Αν το σύμβολο υπάρχει στο λεξικό:
        Αύξησε την τιμή της εγγραφής κατά 1.
    Αλλιώς:
        Εισήγαγε στο λεξικό την εγγραφή (σύμβολο, 1)
    μετρητής=μετρητής+1
Για κάθε ζεύγος (κλειδί, τιμή) του λεξικού:
    πιθανότητα[κλειδί]=τιμή/μετρητής
    Εισήγαγε στον πίνακας_πιθανοτήτων την εγγραφή (κλειδί, πιθανότητα[κλειδί])
Έξοδος: πίνακας_πιθανοτήτων

Διαμέριση

Είσοδος: Σύνολο
Διαμέρισε το Σύνολο σε δύο υποσύνολα Σύνολο_1 και Σύνολο_2 με ίσα αθροίσματα πιθανοτήτων
Έξοδος: Σύνολο_1, Σύνολο_2

Κωδικό_Δένδρο

Είσοδος: πίνακας_πιθανοτήτων
κώδικες={}
ρίζα=πίνακας πιθανοτήτων.τιμές()

τρέχον_αριστερό_παιδί=κενός_κόμβος
τρέχον_δεξί_παιδί= κενός_κόμβος
τρέχουσα_ρίζα=κενός_κόμβος
τρέχουσα_ρίζα=πίνακας πιθανοτήτων.τιμές()
Αν |τρέχουσα_ρίζα|==1:
διάνυσμα_κώδικα=[*null*]
    Μέχρι να φτάσεις τη ρίζα:
        Ανάτρεξε το μονοπάτι από το φύλλο και πρόσθεσε 0 στο διάνυσμα_κώδικα αν είσαι δεξιό παιδί του κόμβου και 1 αν είσαι αριστερό.
        Εισήγαγε στο λεξικό κωδίκων την εγγραφή (Σύμβολο, διάνυσμα_κώδικα)
Αλλιώς:
    Διαμέριση(τρέχουσα ρίζα)
    Θέσε δεξιό παιδί στη τρέχουσα ρίζα το Σύνολο_1 και αριστερό το Σύνολο_2
    Διαμέριση(Σύνολο_1)
    Διαμέριση(Σύνολο_2)

Κωδικοποίηση

Είσοδος:Αρχικό_Κείμενο
Για κάθε σύμβολο στο κείμενο:
    Αντικατέστησε το σύμβολο με την κωδική του λέξη
Έξοδος: Συμπιεσμένο κείμενο

Αποκωδικοποίηση

Είσοδος:Συμπιεσμένο κείμενο
Για κάθε *bit* στο κείμενο:
    Ξεκινώντας από τη ρίζα του δένδρου κωδικοποίησης:
        Αν το *bit* == 0:
            Πήγαινε στο δεξιό παιδί του τρέχοντα κόμβου .
        Αν το *bit* == 0:
            Πήγαινε στο αριστερό παιδί του τρέχοντα κόμβου .
        Αν έφτασες σε φύλλο:
            Αντικατέστησε την κωδική λέξη με το σύμβολο και πήγαινε πάλι στην ρίζα.
Έξοδος: Αρχικό_Κείμενο

Ο παραπάνω αλγόριθμός παρότι είναι θεωρητικά συνεπής, πρακτικά έχει κάποιες δυσκολίες που τον καθιστούν μη αποδοτικό αλγοριθμικά. Καταρχήν κατά τη διαδικασία της δημιουργίας κώδικα, βλέπουμε ότι απαιτείται η διαμέριση του αρχικού συνόλου σε δύο υποσύνολα που το άθροισμα των στοιχείων τους να είναι ίσο. Στην πληροφορική το πρόβλημα αυτό λέγεται "Διαμέριση συνόλου" (Partition Sum) και είναι NP-πλήρες. Μία ελεύθερη μετάφραση της έννοια αυτής είναι ότι για μεγάλες εισόδους, το πρόβλημα γίνεται μη πολυωνυμικά υπολογίσιμο. Άρα καταλαβαίνουμε ότι για μεγάλα αλφάβητα ο παραπάνω κώδικας είναι πρακτικά ανεφάρμοστος. Ένας άλλος λόγος που η διαδικασία αυτή δεν δίνει αποτέλεσμα σε πολλές περιπτώσεις είναι ότι τα σύνολα των ακέραιων αριθμών δεν μπορούν να διαμεριστούν πάντα σε υποσύνολα ίσων αθροισμάτων.

Ένα τέτοιο παράδειγμα είναι το σύνολο [1,1,1,4] το οποίο δεν μπορεί να χωριστεί με κανένα τρόπο ώστε να προκύψουν υποσύνολα ίσω αθροισμάτων. Επειδή ο αλγόριθμος διαμερίζει αναδρομικά το τρέχον σύνολο μέχρι να φτάσει σε μονοσύνολα μπορεί να εφαρμοστεί μόνο σε σ.μ.π των οποίων οι πιθανότητες είναι της μορφής $Pr[X_i = x_i] = \frac{1}{2^{l_i}}$ και το μέγεθος του αλφάβητου είναι και αυτό κάποια δύναμη του 2, γεγονός που σε πρακτικές εφαρμογές δεν αποτελεί την πλειοψηφία των περιπτώσεων.

Ένα ακόμη μειονέκτημα του αλγορίθμου, είναι ότι χρειάζεται την a priori γνώση ολόκληρου του συνόλου δεδομένων. Αυτό δεν είναι πάντα εφικτό ούτε χρήσιμο σε όλες τις εφαρμογές, ιδίως σε αυτές που σχετίζονται με ροή δεδομένων στις οποίες χρειάζονται αλγόριθμοι και μαθηματικά μοντέλα που προσαρμόζονται στις μεταβολές των δεδομένων. Επιπλέον χαρακτηριστικό του παραπάνω αλγορίθμου είναι ότι χρειάζεται να αποθηκευτεί στην μνήμη του ολόκληρο το δένδρο κωδικοποίησης-αποκωδικοποίησης. Κάτι τέτοιο όμως μπορεί να επιβαρύνει πολύ την χωρική πολυπλοκότητα του και να καθίσταται πάλι μη λειτουργικός για εφαρμογές με περιορισμένους χωρικούς πόρους.

Επειδή δεν υπάρχει μοναδικός τρόπος για να συμπιέσουμε ένα σύνολο δεδομένων, μπορούμε να σκεφτούμε έναν εναλλακτικό τρόπο συμπίεσης. Αρχικά ας υπολογίσουμε το πλήθος των bits που χρειαζόμαστε για να κωδικοποιήσουμε n σύμβολα. Για την κωδικοποίηση δύο συμβόλων αρκεί ένα bit καθώς κάθε bit μπορεί να πάρει δύο τιμές. Δύο bits μπορούν να κωδικοποιήσουν μέχρι 4 σύμβολα, τρία bits αρκούν για 8 σύμβολα και γενικά $log_2 2^n = n$ bits επαρκούν για 2^n σύμβολα. Ένας λοιπόν απλοϊκός τρόπος συμπίεσης θα ήταν να κατατάξουμε τα σύμβολα κατά φθίνουσα σειρά εμφάνισης και να δώσουμε τις κωδικές λέξεις με μήκος ένα στα πιο συχνά εμφανιζόμενα σύμβολα, έπειτα να προχωρήσουμε στις κωδικές λέξεις μήκους δύο και ούτω καθεξής. Έτσι μπορούμε να αποφύγουμε το πρόβλημα τις διαμέρισης και τους ειδικούς περιορισμούς στα δεδομένα που αυτό επιβάλει.

Δημιουργία_Κώδικα

Είσοδος:Αλφάβητο, Κατανομή
Κωδικό_Αλφάβητο={0,1}
Ταξινομημένο_Αλφάβητο={} Λεξικό_Κωδικών_Λέξεων={}
Ταξινόμησε τα στοιχεία του Αλφαβήτου κατά φθίνουσα σειρά εμφάνισης
Για μήκος i από 1 μέχρι 2^{l} [α΄]
Μέχρι να τελειώσουν τα στοιχεία του διανύσματος Ταξινομημένο_Αλφάβητο={}:
Εισήγαγε στο Λεξικό_Κωδικών_Λέξεων={} την εγγραφή
(Ταξινομημένο_Αλφάβητο[i],Κωδική λέξη μήκους i) όπου η Κωδική λέξη μήκους i δεν υπάρχει στο Λεξικό_Κωδικών_Λέξεων
Έξοδος:Κωδικές Λέξεις

[α΄] όπου $l = min\{l_i : 2^{l_i} \geq |\text{αλφαβητο}|\}$

Παραδείγματος χάρη ένας τέτοιος κώδικας για την μεταβλητή Υ θα ήταν να αντιστοιχήσουμε:
το $a \to 0$
το $d \to 1$
το $b \to 00$
το $c \to 10$

Ο παραπάνω κώδικας με μία πρώτη ματιά, φαίνεται μη προβληματικός αφού κάθε σύμβολο αντιστοιχίζεται σε διαφορετική κωδική λέξη οπότε δεν υπάρχει περίπτωση διφορούμενης κωδικοποίησης ή αποκωδικοποίησης του εκάστοτε συμβόλου. Αν υπολογίσουμε το μέσο μήκος του κώδικα θα δούμε ότι είναι $\overline{L} = \sum_{i=1}^{n} Pr[X_i = x_i] \cdot l_i \Rightarrow \frac{1}{2} \cdot 1 + \frac{1}{4} \cdot 1 + \frac{1}{8} \cdot 2 + \frac{1}{8} \cdot 2 = 1.25 bits/$σύμβολο. Παρατηρούμε λοιπόν ότι με τον κώδικα που φτιάξαμε όχι μόνο καταφέραμε να αποφύγουμε το πρόβλημα της διαμέρισης αλλά πετύχαμε να δημιουργήσουμε ένα κώδικα με μέσο μήκος μικρότερο από αυτό που προτείνει ο μαθηματικός τύπος της εντροπίας.

Το γεγονός αυτό συνεπάγεται δύο ενδεχόμενα· το πρώτο είναι ο μαθηματικός τύπος να είναι λάθος ενώ η δεύτερη προσέγγιση προτείνει να μην είναι η σκέψη μας σωστή. Το λογικό κενό στον παραπάνω τρόπο

συμπίεσης θα φανεί όταν επιχειρήσουμε να αποκωδικοποιήσουμε το κείμενο που παράχθηκε με βάση την $P_Y(y)$ και τον κώδικα που φτιάξαμε. Η πρώτη γραμμή του κείμενου cadadaabaaaabadaddaa μεταφράζεται ως 10010100000000000101100. Αν προσπαθήσουμε να την αποκωδικοποιήσουμε θα δούμε ότι υπάρχει μία ασάφεια σχετικά με τη συμβολοσειρά από την οποία προήλθε. Λόγου χάρη τα τρία πρώτα $bits$ 100 μπορεί να προήλθαν είτε από την συμβολοσειρά db ή από τη daa. Το ίδιο πρόβλημα επανέρχεται και κατά το μέσο της κωδικοποιημένης συμβολοσειράς όπου υπάρχει μία πληθώρα μηδενικών που είναι δύσκολο να προσδιορίσουμε από ποιες συμβολοσειρές παράχθηκαν. Μία διόρθωση θα ήταν να έχουμε μία ακολουθία $bits$ που θα λειτουργούσε ώς διαχωριστικό μεταξύ των κωδικών λέξεων. Όμως με αυτόν το τρόπο θα εισαγάγαμε επιπλέον πληροφορία κάτι που θα αύξανε τον μέσο αριθμό $bits$/σύμβολο που θα χρησιμοποιούσαμε με αποτέλεσμα να έχουμε ένα κώδικα με μεγαλύτερο μέσο μήκος από αυτόν που προτείνει η εντροπία και εν τέλει χειρότερη συμπίεση.

Που είναι όμως το λάθος στον παραπάνω κώδικα και πέσαμε σε αυτά τα αδιέξοδα; Η παγίδα είναι ότι ο κώδικας κατέχει λιγότερη πληροφορία από αυτή που υποδεικνύει η εντροπία. Αυτό πρακτικά σημαίνει ότι η δυαδική αναπαράσταση κάθε συμβόλου υπολείπεται σε πληροφορία ώστε να είναι πλήρως διαχωρίσιμη από τις αναπαραστάσεις συμβόλων που υπακούν στην ίδια σ.μ.π. Θυμηθείτε αυτό που είπαμε και στο πρώτο κεφάλαιο, ότι η εντροπία επί της ουσίας είναι η ελάχιστη πληροφορία που χρειαζόμαστε κατά μέσο όρο για να αναπαραστήσουμε την τιμή μιας τυχαίας μεταβλητής που ακολουθεί μία συγκεκριμένη σ.μ.π και αυτή η αναπαράσταση να την καθιστά μοναδική σε σχέση με τις αναπαραστάσεις των υπόλοιπων τιμών της. Το παραπάνω ατόπημα αναδεικνύει την αναγκαιότητα ύπαρξης μιας μαθηματικά θεμελιωμένης θεωρίας και την χρήση αυστηρών ορισμών ώστε να μπορούμε να ελέγχουμε την ορθότητα και λειτουργικότητα των πρακτικών υλοποιήσεων μας.

3.2 Είδη Κωδίκων

Στην παράγραφο αυτή θα θεμελιώσουμε θεωρητικά τις έννοιες των διαφορετικών κατηγοριών κωδίκων που υπάρχουν, και θα εξετάσουμε τις μεταξύ του συσχετίσεις.

Ορισμός 3.1. *Έστω μία διακριτή τυχαία μεταβλητή X που παίρνει τιμές σε ένα πεπερασμένο σύνολο $\mathcal{X}$ και F το κωδικό αλφάβητο, με $F^* = \{f_1 f_2 \dots f_n : f_i \in F \ \forall i = 1, \dots, n \ \wedge \ n < \infty\}$. Τότε ορίζουμε ως* ***κώδικα της τυχαίας μεταβλητής X*** *την απεικόνιση κάθε τιμής της x σε μία κωδική λέξη $C(x)$, με $C(x) \in F^*$.*

Ορισμός 3.2. *Έστω μία διακριτή τυχαία μεταβλητή X που παίρνει τιμές σε ένα πεπερασμένο σύνολο $\mathcal{X}$ και $x_1 x_2 \dots x_n$ μία συμβολοσειρά της πεπερασμένου μήκους. Τότε η* ***κωδική λέξη της συμβολοσειράς*** *ορίζεται ως η παράθεση των κωδικών λέξεων των αντίστοιχων τιμών της.*

$$\boxed{C^*(x_1 x_2, \dots x_n) = C(x_1)C(x_2)\dots C(x_n)} \tag{3.1}$$

Στην εισαγωγή είδαμε την παραγωγή δύο κωδίκων. Ο πρώτος προερχόταν από τη διαδικασία διαμέρισης του χώρου των ενδεχομένων σε δύο ισοπίθανα σύνολα και ο δεύτερος από την ανάθεση των σύντομων δυαδικών αναπαραστάσεων στις πιο συχνά εμφανιζόμενες τιμές της τυχαίας μεταβλητής. Κατά τη διαδικασία της αποκωδικοποίησης μία συμβολοσειράς με βάση τον δεύτερο αντιληφθήκαμε ότι δεν υπάρχει αμφιμονοσήμαντη απεικόνιση σε μία αρχική συμβολοσειρά, γεγονός που κατέστησε τον παραπάνω κώδικα πρακτικά μη εφαρμόσιμο. Εν αντιθέσει με τον πρώτο που αν παρατηρήσουμε δεν δημιουργεί τέτοια προβλήματα κατά την αποκωδικοποίηση.

Ας πάρουμε πάλι την πρώτη συμβολοσειρά, cadadaabaaaabadaddaa, αλλά αυτή φορά θα την κωδικοποιήσουμε με βάση τον πρώτο κώδικα. Τότε θα προκύψει η δυαδική συμβολοσειρά:

01010010011011111101110010000011

.

Ξεκινώντας από το πρώτο $bit = 0$ βλέπουμε ότι υπάρχουν τρεις επιλογές για το πρώτο σύμβολο, μπορεί αν είναι το b, c ή d. Συνεχίζοντας στο δεύτερο $bit = 1$ οι επιλογές μειώνονται στις δύο, τα σύμβολα b, c. Το τρίτο $bit = 0$ μας αφήνει μοναδική επιλογή το γράμμα b. Στην συνέχεια προχωράμε στο πέμπτο $bit = 1$. Το μοναδικό σύμβολο που αρχίζει με 1 είναι το a οπότε αποκωδικοποιούμε και την δεύτερη κωδική λέξη. Βολιδοσκοπώντας τη διαδικασία διαπιστώνουμε ότι για να αποκωδικοποιήσουμε μία κωδική λέξη ξεκινάμε από τη ρίζα του κωδικού δένδρου και το διατρέχουμε σύμφωνα με τα $bits$ που διαβάζουμε μέχρι να φτάσουμε σε κάποιο φύλλο. Έπειτα ο αλγόριθμος επιστρέφει πάλι στην ρίζα προκειμένου να αποκωδικοποιήσει την επόμενη λέξη.

Στην εισαγωγή αναφέραμε πως ο ελαττωματικός κώδικας μπορεί να διορθωθεί αν εισάγουμε επιπλέον $bits$ ή αν αλλάξουμε λίγο τις κωδικές λέξεις ώστε να διαχωρίζονται μεταξύ τους. Μία λύση θα ήταν να χρησιμοποιούμε είτε το 0 είτε το 1 σαν σημείο έναρξης ή λήξης της κωδικής λέξης. Ένας τέτοιος κώδικας για την μεταβλητή Υ θα ήταν ο παρακάτω:

το $a \to 0$

το $d \to 01$

το $b \to 011$

το $c \to 0111$

Μεταφράζοντας πάλι την πρώτη συμβολοσειρά $cadadaabaaaabadaddaa$ θα προκύψει η ακολουθία:

$$01110010010001100000110010010100$$

.

Η αποκωδικοποίηση της παραπάνω ακολουθίας ακολουθεί τον εξής απλό κανόνα: όταν ο αλγόριθμος βλέπει 0 ξέρει ότι ξεκινάει μία καινούρια κωδική λέξη. Το μόνο αρνητικό του παραπάνω κώδικα σε σχέση με τον προηγούμενο είναι ότι χρειάζεται η γνώση και των επόμενων $bits$ που μπορεί να μην ανήκουν στην τρέχουσα κωδική λέξη για να μην υπάρχει διφορούμενη αποκωδικοποίηση. Στο κώδικα που βασίστηκε στη διαμέριση, κάθε φορά που τελειώνει μία κωδική λέξη ξέρουμε αμέσως το σύμβολο από όπου προήλθε. Στον παραπάνω κώδικα παραδείγματος χάρη όταν συναντάμε το 0 δεν ξέρουμε αν είναι το σύμβολο 'ά" η αρχή της κωδικής λέξης κάποιο άλλου συμβόλου. Για να γίνει το παραπάνω αντιληπτό παρατηρούμε κατά το μέσο της ακολουθίας που εμφανίζεται η υποσυμβολοσειρά 10000011. Για να αποφασίσουμε ότι το δεύτερο bit παριστάνει το γράμμα α, είναι απαραίτητη η γνώση του τρίτου bit. Το ίδιο ισχύει και για το δεύτερο, το τρίτο, το τέταρτο και το πέμπτο bit.

Μέχρι στιγμής έχουμε έρθει αντιμέτωποι με τρεις διαφορετικές κατηγορίες κωδίκων. Ο πρώτος κώδικας παρέχει μια αμφιμονοσήμαντη αντιστοίχιση μεταξύ των τιμών της τ.μ Υ αλλά και των συμβολοσειρών που παράγονται από αυτή με τις αντίστοιχες κωδικές λέξεις. Ο δεύτερος κώδικας καταφέρνει να απεικονίσει με μοναδικό τρόπο τις τιμές της τ.μ Υ σε κώδικες λέξεις αλλά αποτυγχάνει με τις παραγόμενες συμβολοσειρές. Τέτοιοι κώδικες ονομάζονται μη ιδιόμορφοι. Τέλος ο τρίτος κώδικας επιτυγχάνει ότι και ο πρώτος όσον αφορά στην κωδικοποίηση αλλά η αποκωδικοποίηση του διαφέρει ως προς την ανάγκη γνώσης των επόμενων κωδικών λέξεων προκειμένου να επιτευχθεί μία σωστή μετάφραση. Καταλαβαίνουμε λοιπόν ότι για να έχουμε μία επιτυχή αποκωδικοποίηση θα πρέπει να μην υπάρχει διφορούμενη μετάφραση των κωδικοποιημένων συμβολοσειρών. Οι κώδικες που εξασφαλίζουν την παραπάνω απαίτηση ονομάζονται μοναδικά αποκωδικοποιήσιμοι

Ορισμός 3.3. *Έστω μία διακριτή τυχαία μεταβλητή X που παίρνει τιμές σε ένα πεπερασμένο σύνολο $\mathcal{X}$ και F το κωδικό αλφάβητο. Ένα κώδικας λέγεται* ***μη ιδιόμορφος*** *αν η C είναι αμφιμονοσήμαντη απεικόνιση από το $\mathcal{X}$ στο F^*, δηλαδή*

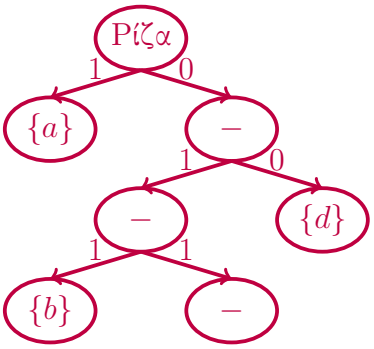
$$\boxed{\forall x, x' \ με \ x \neq x' \ ισχύει \ C(x) \neq C(x') \ με \ C(x), C(x') \in F^*} \tag{3.2}$$

Ορισμός 3.4. *Έστω μία διακριτή τυχαία μεταβλητή X που παίρνει τιμές σε ένα πεπερασμένο σύνολο $\mathcal{X}$. Ο κώδικας της X, λέγεται* ***μοναδικά αποκωδικοποιήσιμος*** *αν η απεικόνιση μεταξύ των συμβολοσειρών που παράγονται από την X και των αντίστοιχων κωδικών λέξεων που ανήκουν στο F^* είναι αμφιμονοσήμαντη,*

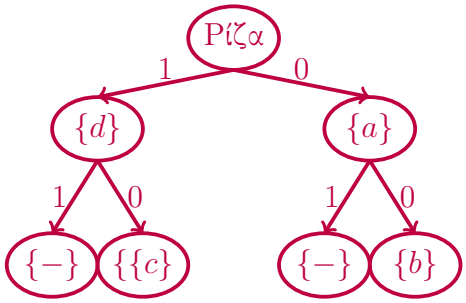
δηλαδή

$$\boxed{\forall\ \vec{\mathbf{x}},\ \vec{\mathbf{x}}' \in \mathcal{X}^*\ \mu\epsilon\ \vec{\mathbf{x}} \neq \vec{\mathbf{x}}' \iota\sigma\chi\acute{\upsilon}\epsilon\iota\ C^*(\vec{\mathbf{x}}) \neq C^*(\vec{\mathbf{x}}'),\ \mu\epsilon\ \mathcal{X}^* = \{x_1x_2\cdots x_n : x_i \in \mathcal{X}\ \forall i = 1,\cdots,n\ \wedge n < \infty\}} \tag{3.3}$$

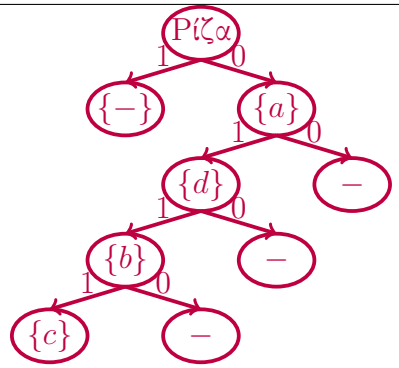
Επειδή παρατηρήσαμε τις διαφοροποιήσεις που υπάρχουν κατά την αποκωδικοποίηση των δυαδικών συμβολοσειρών μεταξύ του πρώτου και του τρίτου κώδικα, είναι λογικό να προσπαθήσουμε, μελετώντας τη δομή τους, να βρούμε τα δομικά χαρακτηριστικά που ευθύνονται για αυτές τις διαφορές. Για το λόγο αυτό αφού αναπαραστήσαμε τις λέξεις του πρώτου κώδικα μεσώ ενός δυαδικού δένδρου, το ίδιο θα επιχειρήσουμε για τις λέξεις των υπόλοιπων κωδίκων (Σχήμα 3.1).



(α′) Κώδικας 1 για την τυχαία μεταβλητή Y



(β′) Κώδικας 2 για την τυχαία μεταβλητή Y



(γ′) Κώδικας 3 για την τυχαία μεταβλητή Y

Σχήμα 3.1: Η δενδρική αναπαράσταση των κωδίκων

Κάθε δένδρο αποτελείται από μία ρίζα και δύο είδη κόμβων, τους εσωτερικούς από όπου προέρχονται άλλοι κόμβοι και τους εξωτερικούς ή φύλλα που είναι οι τερματικοί κόμβοι. Η κωδική λέξη κάθε συμβόλου ολοκληρώνεται όταν ξεκινήσουμε από τη ρίζα και διατρέξουμε το μονοπάτι του δένδρου που θα μας οδηγήσει στον κόμβο με το αντίστοιχο σύμβολο. Η κύρια διαφορά που μπορούμε να παρατηρήσουμε μεταξύ του κώδικα 1 σε σχέση με τους 2 και 3 είναι ότι κανένας εσωτερικός κόμβος δεν περιέχει κάποιο μονοσύνολο με το αντίστοιχο σύμβολο. Όλες οι τιμές της τ.μ Y βρίσκονται στα φύλλα του πρώτου δένδρου και όχι ενδιάμεσα, όπως συμβαίνει στα δένδρα 2 και 3. Η ιδιότητα αυτή χωρίζει τους κώδικες σε δύο κατηγορίες, τους μη προθεματικούς ή στιγμιαίους, σαν τον κώδικα 1 όπου καμία κωδική λέξη δεν αποτελεί πρόθεμα κάποιας άλλης, και τους προθεματικούς που οι κωδικές λέξεις μπορούν να αποτελέσουν προθέματα των υπόλοιπων. Λόγου χάρη στον κώδικα 2 η κωδική λέξη 0 αποτελεί πρόθεμα της κωδικής λέξης 00. Αντίστοιχα στον κώδικα 3 η κωδική λέξη 011 είναι πρόθεμα της κωδικής λέξης $\underline{011}1$.

Ορισμός 3.5. *Έστω X μία τυχαία μεταβλητή που παίρνει τιμές σε ένα πεπερασμένο σύνολο $\mathcal{X}$, F ένα κωδικό αλφάβητο και $\mathcal{C} = \{C(x_i)\}_{x_i \in \mathcal{X}}$ με $C(x_i) \in F^*\ \forall i$ ένας κώδικας των τιμών της τυχαίας μεταβλητής. Ο κώδικας $\mathcal{C}$ λέγεται* ***μη προθεματικός*** *ή* ***στιγμιαίος*** *αν καμία κωδική λέξη δεν αποτελεί πρόθεμα για κάποια άλλη.*

Πόρισμα 3.1. *Έστω X μία τυχαία μεταβλητή που παίρνει τιμές σε ένα πεπερασμένο σύνολο $\mathcal{X}$, F ένα κωδικό αλφάβητο και $\mathcal{C} = \{C(x_i)\}_{x_i \in \mathcal{X}}$ με $C(x_i) \in F^*\ \forall i$ ένας κώδικας των τιμών της τυχαίας μεταβλητής. Αν ο $\mathcal{C}$ είναι μη προθεματικός τότε είναι και μη ιδιόμορφος*

Απόδειξη

Έστω ότι ο $\mathcal{C}$ δεν είναι μη ιδιόμορφος. Τότε θα πρέπει να υπάρχουν δύο τιμές της τυχαίας μεταβλητής X, έστω οι $x_i, x_j \in \mathcal{X}$ με $x_i \neq x_j$ τέτοιες ώστε $C(x_i) = C(x_j)$. Από την τελευταία σχέση έπεται όμως ότι η μία κωδική λέξη θα είναι πρόθεμα της άλλης κατά τετριμμένο τρόπο, το οποίο είναι άτοπο καθώς ο κώδικας είναι στιγμιαίος. Άρα για κάθε $x_i \neq x_j$ πρέπει $C(x_i) \neq C(x_j)$ το οποίο αποτελεί και τον ορισμό του μη ιδιόμορφου κώδικά.

Θεώρημα 3.1. *Έστω X μία τυχαία μεταβλητή που παίρνει τιμές σε ένα πεπερασμένο σύνολο $\mathcal{X}$ και $\mathcal{C} = \{C(x_i)\}_{x_i \in \mathcal{X}}$ με $C(x_i) \in F$ $\forall x_i \in \mathcal{X}$ ένας μη προθεματικός κώδικας για τις τιμές της τυχαίας μεταβλητής X. Αν συμβολίσουμε με $\mathcal{C}^* = \{C(x_1x_2\cdots x_n)\}_{x_1x_2\cdots x_n \in \mathcal{X}^*}$ και $\mathcal{C}^k = \{C(x_1x_2\cdots x_k) : l(x_1x_2\cdots x_k) = k\}_{x_1x_2\cdots x_n \in \mathcal{X}^*}$ τότε ο $\mathcal{C}^*$ είναι μοναδικά αποκωδικοποιήσιμος ενώ ο $\mathcal{C}^k$ είναι μη προθεματικός και μη ιδιόμορφος.*

Απόδειξη

Επειδή ο κώδικας $\mathcal{C}$ είναι στιγμιαίος γνωρίζουμε ότι για οποιαδήποτε $x_i, x_j \in \mathcal{X}$ με $x_i \neq x_j$ ισχύει $C(x_i) \neq C(x_j)$ καθώς και ότι η $C(x_i)$ δεν αποτελεί πρόθεμα της $C(x_j)$ ή και το αντίστροφο. Υπενθυμίζουμε ακόμη ότι η κωδική λέξη μίας συμβολοσειράς είναι η παράθεση των κωδικών λέξεων των τιμών της μεταβλητής που συμμετέχουν στην συμβολοσειρά, δηλαδή $C^*(x_1x_2\cdots x_n) = C(x_1)C(x_2)\cdots C(x_n)$.

***Επαγωγική Βάση:** Θα αποδείξουμε ότι η κωδική λέξη που προκύπτει από την παράθεση δύο κωδικών λέξεων που ανήκουν στον στιγμιαίο κώδικα $\mathcal{C}$ ανήκει στον στιγμιαίο και μη ιδιόμορφο κώδικα $\mathcal{C}^2$, όπου $\mathcal{C}^2 = \{C^*(x_ix_j)\}_{x_ix_j \in \mathcal{X}^2}$*

Μη Ιδιόμορφος

Έστω ότι δεν ισχύει ο ισχυρισμός μας, δηλαδή η παράθεση δύο κωδικών λέξεων δεν είναι αμφιμονοσήμαντη. Τότε θα πρέπει να υπάρχουν δύο συμβολοσειρές μήκους δύο, που να παράγουν την ίδια κωδική λέξη. Δηλαδή υπάρχουν $x_ix_j, x_kx_l \in \mathcal{X}^2$ για τις οποίες ισχύει $C^*(x_ix_j) = C^*(x_kx_l)$. Επειδή όμως $C^*(x_ix_j) = C(x_i)C(x_j)$ και $C^*(x_kx_l) = C(x_k)C(x_l) \Rightarrow C(x_i)C(x_j) = C(x_k)C(x_l)$.

• Έστω ότι $x_i = x_k$. Επειδή ο $\mathcal{C}$ είναι συνάρτηση θα πρέπει $C(x_i) = C(x_k)$, Τότε δεν γίνεται $C(x_i)C(x_j) = C(x_k)C(x_l)$ και $x_j \neq x_l$ καθώς τα $C(x_j), C(x_l) \in \mathcal{C}$ και ο κώδικας $\mathcal{C}$ είναι μη ιδιόμορφος. Οπότε αν $x_i = x_k$ τότε $C(x_i)C(x_j) = C(x_k)C(x_l)$ αν και μόνο αν $x_ix_j = x_kx_l$.

• Έστω ότι $x_j = x_l$. Επειδή ο $\mathcal{C}$ είναι συνάρτηση θα πρέπει $C(x_j) = C(x_l)$, τότε δεν γίνεται $C(x_i)C(x_j) = C(x_k)C(x_l)$ και $x_i \neq x_k$ καθώς τα $C(x_i), C(x_k) \in \mathcal{C}$ και ο $\mathcal{C}$ είναι μη ιδιόμορφος. Οπότε αν $x_j = x_l$ τότε $C(x_i)C(x_j) = C(x_k)C(x_l)$ αν και μόνο αν $x_ix_j = x_kx_l$.

• Έστω ότι $x_i \neq x_k$ και $x_j \neq x_l$. Χωρίς βλάβη της γενικότητας υποθέτουμε ότι $l_i = l(C(x_i)) < l(C(x_k)) = l_k$. Επειδή όμως $C(x_i)C(x_j) = C(x_k)C(x_l)$, έπεται ότι η κωδική λέξη $C(x_i)$ θα αποτελεί πρόθεμα της $C(x_k)$, το οποίο είναι άτοπο καθώς ο κώδικας $\mathcal{C}$ είναι μη προθεματικός. Άρα $C(x_i)C(x_j) = C(x_k)C(x_l)$ αν και μόνο αν $x_ix_j = x_kx_l$.

Ιδιότητα της μη προθεματικότητας

Έστω ότι ο $\mathcal{C}^2$ δεν είναι μη προθεματικός, δηλαδή υπάρχουν τουλάχιστον δύο συμβολοσειρές μήκους δύο που η κωδική λέξη της μίας αποτελεί πρόθεμα τη άλλης. Έστω ότι η $C^*(x_ix_j)$ αποτελεί πρόθεμα της $C^*(x_kx_l)$. Τότε είτε το $C(x_i)$ θα αποτελεί πρόθεμα της $C(x_k)$ αν $l_i < l_k$ ή το αντίστροφο αν $l_i > l_k$, το οποίο είναι αδύνατο καθώς ο κώδικας $\mathcal{C}$ είναι μη προθεματικός. Άρα καμία λέξη που ανήκει στον $\mathcal{C}^2$ δεν θα αποτελεί πρόθεμα κάποιας άλλης. Από τις παραπάνω περιπτώσεις έπεται ότι ο κώδικας $\mathcal{C}^2 = \{C^*(x_ix_j)\}_{x_ix_j \in \mathcal{X}^2}$ είναι στιγμιαίος και μη ιδιόμορφος.

*Στην **επαγωγική υπόθεση**, ισχυριζόμαστε ότι η κωδική λέξη που προκύπτει από την παράθεση k κωδικών λέξεων που ανήκουν στον στιγμιαίο κώδικα $\mathcal{C}$ ανήκει στον στιγμιαίο και μη ιδιόμορφο κώδικα $\mathcal{C}^k$, όπου $\mathcal{C}^k = \{C^*(x_i\cdots x_{i+k})\}_{x_i\cdots x_{i+k} \in \mathcal{X}^k}$.*

*Στο **επαγωγικό βήμα** θα δείξουμε ότι και η παράθεση $k+1$ κωδικών λέξεων που ανήκουν στον στιγμιαίο κώδικα $\mathcal{C}$ δημιουργεί τον στιγμιαίο και μη ιδιόμορφο κώδικα $\mathcal{C}^{k+1}$, όπου $\mathcal{C}^{k+1} = \{C^*(x_i\cdots x_{i+k+1})\}_{x_i\cdots x_{i+k+1}} \in \mathcal{X}^{k+1}$.*

Έστω ένα τυχαίο στοιχείο $C^*(x_1x_2\cdots x_{k+1})$ που ανήκει στον $\mathcal{C}^{k+1}$. Επειδή:

$$C^*(x_1x_2\cdots x_{k+1}) = \underbrace{C^*(x_1x_2\cdots x_k)}_{\in \mathcal{C}^k}\underbrace{C(x_{k+1})}_{\in \mathcal{C}}$$

Μη Ιδιόμορφος

Έστω ότι υπάρχει μία ακόμη συμβολοσειρά $x_l\cdots x_{l+k+1} \in \mathcal{C}^{k+1}$ τέτοια ώστε $C^*(x_l\cdots x_{l+k+1}) = C^*(x_1x_2\cdots x_{k+1}) \Rightarrow C^*(x_l\cdots x_{l+k})C(x_{l+k+1}) = C^*(x_1x_2\cdots x_k)C(x_{k+1})$.

•. Έστω $x_l\cdots x_{l+k} = x_1\cdots x_k$ και $C^*(x_l\cdots x_{l+k})C(x_{l+k+1}) = C^*(x_1x_2\cdots x_k)C(x_{k+1})$. Επειδή ο $\mathcal{C}^k$ είναι στιγμιαίος από επαγωγική υπόθεση έπεται ότι $x_l\cdots x_{l+k} = x_1\cdots x_k \Leftrightarrow C^*(x_l\cdots x_{l+k}) = C^*(x_1\cdots x_k)$. Τότε

δεν γίνεται $C^*(x_l \cdots x_{l+k})C(x_{l+k+1}) = C^*(x_1x_2 \cdots x_k)C(x_{k+1})$ *και* $x_{l+k+1} \neq x_{k+1}$ *καθώς και ο κώδικας* $\mathcal{C}$ *είναι μη ιδιόμορφος.*

•. *Έστω* $x_l \cdots x_{l+k+1} = x_1 \cdots x_{k+1}$ *και* $C^*(x_l \cdots x_{l+k})C(x_{l+k+1}) = C^*(x_1x_2 \cdots x_k)C(x_{k+1})$. *Επειδή ο* $\mathcal{C}$ *είναι στιγμιαίος από επαγωγική υπόθεση έπεται ότι* $x_{l+k+1} = x_{k+1} \Leftrightarrow C(x_{l+k+1}) = C(x_{k+1})$. *Τότε δεν γίνεται* $C^*(x_l \cdots x_{l+k})C(x_{l+k+1}) = C^*(x_1x_2 \cdots x_k)C(x_{k+1})$ *και* $x_l \cdots x_{l+k} \neq x_1 \cdots x_k$ *καθώς και ο κώδικας* $\mathcal{C}^k$ *είναι μη ιδιόμορφος.*

•. *Έστω* $x_l \cdots x_{l+k+1} \neq x_1 \cdots x_{k+1}, x_l \cdots x_{l+k} \neq x_1 \cdots x_k$ *και* $C^*(x_l \cdots x_{l+k})C(x_{l+k+1}) = C^*(x_1x_2 \cdots x_k)C(x_{k+1})$ *Χωρίς βλάβη της γενικότητα υποθέτουμε ότι* $l_{l\to l+k} = l(C^*(x_l \cdots x_{l+k})) < l(C^*(x_1x_2 \cdots x_k)) = l_{1\to k}$. *Επειδή όμως* $C^*(x_l \cdots x_{l+k})C(x_{l+k+1}) = C^*(x_1x_2 \cdots x_k)C(x_{k+1})$ *έπεται ότι η κωδική λέξη* $C^*(x_l \cdots x_{l+k})$ *θα αποτελεί πρόθεμα της* $= C^*(x_1x_2 \cdots x_k)$, *άτοπο καθώς ο κώδικας* $\mathcal{C}^k$ *είναι στιγμιαίος. Άρα για οποιεσδήποτε συμβολοσειρές* $x_l \cdots x_{l+k+1}, x_1 \cdots x_{k+1} \in \mathcal{C}^{k+1}$ *έπεται ότι:*

$$C^*(x_l \cdots x_{l+k+1}) = C^*(x_1x_2 \cdots x_{k+1}) \Leftrightarrow x_l \cdots x_{l+k+1} = x_1x_2 \cdots x_{k+1} \tag{3.4}$$

Ιδιότητα της μη προθεματικότητας

Έστω ότι ο $\mathcal{C}^{k+1}$ *δεν είναι μη προθεματικός, δηλαδή υπάρχουν τουλάχιστον δύο συμβολοσειρές μήκους* $k+1$ *που η κωδική λέξη της μίας αποτελεί πρόθεμα τη άλλης. Έστω ότι η* $C^*(x_i \cdots x_{i+k+1})$ *αποτελεί πρόθεμα της* $C^*(x_l \cdots x_{l+k+1})$. *Τότε είτε το* $C^*(x_i \cdots x_{i+k})$ *θα αποτελεί πρόθεμα της* $C^*(x_l \cdots x_{l+k})$ *αν* $l_{i\to i+k} < l_{l\to l+k}$ *ή το αντίστροφο αν* $l_{i\to i+k} > l_{l\to l+k}$, *το οποίο είναι αδύνατο καθώς ο κώδικας* $\mathcal{C}^k$ *είναι μη προθεματικός. Άρα καμία λέξη που ανήκει στον* $\mathcal{C}^k$ *δεν θα αποτελεί πρόθεμα κάποιας άλλης. Από τις παραπάνω περιπτώσεις έπεται ότι ο κώδικας* $\mathcal{C}^{k+1} = \{C^*(x_i \cdots x_{i+k+1})\}_{x_i \cdots x_{i+k+1}} \in \mathcal{X}^{k+1}$ *είναι στιγμιαίος και μη ιδιόμορφος .*

Τώρα ερχόμαστε στο δεύτερο ζητούμενο που είναι να αποδείξουμε ότι ο κώδικας $\mathcal{C}^* = \{C(x_1x_2 \cdots x_n | x_1x_2 \cdots x_n \in \mathcal{X}^*)\}$ *είναι μοναδικά αποκωδικοποιήσιμος. Για να ολοκληρώσουμε την απόδειξη αρκεί να δείξουμε ότι για οποιεσδήποτε συμβολοσειρές* $x_i \cdots x_{i+j}$ *και* $x_k \cdots x_{k+l}$ *ισχύει:*

$$x_i \cdots x_{i+j} = x_k \cdots x_{k+l} \Leftrightarrow C^*(x_i \cdots x_{i+j}) = C^*(x_k \cdots x_{k+l})$$

• *Από τον τρόπο συμβολισμού καταλαβαίνουνε ότι η πρώτη συμβολοσειρά έχει μήκος* j *και η δεύτερη* l. *Υποθέτουμε ότι* $l \neq j$ *και χωρίς βλάβη της γενικότητας διαλέγουμε* $l < j$. *Έστω ότι για* $l < j$ *ισχύει* $C^*(x_i \cdots x_{i+j}) = C^*(x_k \cdots x_{k+l})$. *Από τον παραπάνω ισχυρισμό έπεται ότι τα μήκη των δύο κωδικών λέξεων είναι ίδια,* $l(C^*(x_i \cdots x_{i+j})) = l(C^*(x_k \cdots x_{k+l}))$. *Αφού* $l < j$ *η κωδική λέξη* $C^*(x_i \cdots x_{i+j})$ *θα αποτελεί παράθεση περισσότερων κωδικών λέξεων από ότι η* $C^*(x_k \cdots x_{k+l})$, *οπότε μπορούμε να διαλέξουμε τις* l *πρώτες κωδικές λέξεις της* $C^*(x_i \cdots x_{i+j}) = C(x_i)C(x_{i+1}) \cdots C(x_{i+l})C(x_{i+l+1}) \cdots C(x_{i+j})$, *δηλαδή την κωδική υποσυμβολοσειρά* $C(x_i)C(x_{i+1}) \cdots C(x_{i+l})$. *Επειδή ισχύει* $l(C^*(x_i \cdots x_{i+j})) = l(C(x_k \cdots x_{k+l}))$ *και* $l < j$ *έπεται ότι* $l(C^*(x_i \cdots x_i + l)) < l(C(x_k \cdots x_{k+l}))$.

Όμως από την υπόθεση ισχύει ότι $C^*(x_i \cdots x_{i+j}) = C^*(x_k \cdots x_{k+l})$, *άρα συνεπάγεται ότι η κωδική λέξη* $C^*(x_i \cdots x_i + l)$ *αποτελεί πρόθεμα της* $C^*(x_k \cdots x_{k+l})$, *το οποίο είναι άτοπο καθώς και οι δύο κωδικές λέξεις προκύπτουν από μία παράθεση* l *κωδικών λέξεων που ανήκουν στο στιγμιαίο κώδικα* $\mathcal{C}$, *γεγονός που με τη σειρά του τις εντάσσει στον κώδικα* $\mathcal{C}^l$, *ο οποίος όπως δείξαμε στο πρώτο μέρος της απόδειξης είναι στιγμιαίος. Άρα δεν γίνεται ούτε η* $C^*(x_i \cdots x_i + l)$ *να αποτελεί πρόθεμα της* $C^*(x_k \cdots x_{k+l})$, *ούτε καμίας κωδικής λέξης της μορφής* $C^*(x_i \cdots x_i + m)$ *με* $m < l$. *Ολοκληρώνοντας βλέπουμε ότι δεν γίνεται για* $l \neq j$ *να ισχύει* $C^*(x_i \cdots x_{i+j}) = C^*(x_k \cdots x_{k+l})$, *οπότε αν* $l \neq j \Rightarrow C^*(x_i \cdots x_{i+j}) \neq C^*(x_k \cdots x_{k+l})$. *Άρα ο κώδικας* $\mathcal{C}^*$ *είναι μοναδικά αποκωδικοποιήσιμος*

Πόρισμα 3.2. *Ένας στιγμιαίος κώδικάς είναι και μοναδικά αποκωδικοποιήσιμος*

Απόδειξη

Στο παραπάνω θεώρημα δείξαμε ότι αν έχουμε μία τυχαία μεταβλητή X *που για τις τιμές της υπάρχει ένα στιγμιαίος κώδικας* $\mathcal{C} = \{C(x_i)\}_{x_i \in \mathcal{X}}$, *τότε υπάρχει μία αμφιμονοσήμαντη αντιστοιχία μεταξύ των συμβολοσειρών που ανήκουν στο σύνολο* $\mathcal{X}^*$ *και των κωδικών λέξεων που ανήκουν στο κώδικα* $\mathcal{C}^*$. *Από αυτό έπεται ότι ένας στιγμιαίος κώδικάς είναι και μοναδικά αποκωδικοποιήσιμος.*

Οι μη προθεματικοί κώδικες είναι ιδιαίτερα χρήσιμοι όταν απαιτείται γρήγορη αποκωδικοποίηση των δεδομένων. Μία εφαρμογή που χρειάζεται ταχείς κώδικες αποτελεί το διαδίκτυο και συγκεκριμένα η επικοινωνία των φυλλομετρητών ιστοσελίδων (browsers) με τους εξυπηρετητές (servers). Παρόλα αυτά οι μοναδικά αποκωδικοποιήσιμοι κώδικες μπορούν να είναι χρήσιμοι σε εφαρμογές που η ταχύτητα αποκωδικοποίησης μπορεί να μην παίζει τόσο μεγάλο ρόλο αλλά για παράδειγμα η απλότητα και η ευελιξία του αλγορίθμου. Ενώ όμως είδαμε είναι σχετικά εύκολο να ξεχωρίσουμε ένα μη προθεματικό κώδικα από οποιονδήποτε άλλο, καθώς οι στιγμιαίοι κώδικες δεν περιέχουν τιμές της εκάστοτε τυχαίας μεταβλητής στους ενδιάμεσους κόμβους των δυαδικών δένδρων που τους αναπαριστούν, δεν μπορούμε να πούμε το ίδιο και για τους μοναδικά αποκωδικοποιήσιμους κώδικες. Ήδη από τα δένδρα των κωδίκων 2 και 3 φαίνεται ότι ενώ και δύο περιλαμβάνουν τιμές της τυχαίας μεταβλητής στους ενδιάμεσους κόμβους τους, δεν ανήκουν στην ίδια κατηγορία.

3.3 Κριτήρια για μοναδικά αποκωδικοποιήσιμους και στιγμιαίους κώδικες

Στην ενότητα αυτή θα αναφέρουμε δύο βασικά κριτήρια με βάση τα οποία μπορούν να διαχωριστούν οι κώδικες σε μοναδικά αποκωδικοποιήσιμους και μη αλλά και να ελέγξουμε πότε ένας κώδικας είναι στιγμιαίος. Το πρώτο κριτήριο που θα παρουσιάσουμε είναι ένας αλγόριθμός των Sardinas και Patterson [Say12] που μας βοηθάει να αποφασίσουμε σε πολυωνυμικό χρόνο αν ένας κώδικας είναι μοναδικά αποκωδικοποιήσιμος ή όχι

Αλγόριθμος 3.1. *(Ο αλγόριθμός των Sardinas και Patterson [Say12]) Έστω X μία τυχαία μεταβλητή που παίρνει τιμές σε ένα πεπερασμένο σύνολο $\mathcal{X}$ και $\mathcal{C} = \{C(x_i)\}_{x_i \in \mathcal{X}}$ με $C(x_i) \in F^*$ ο κώδικας για τις τιμές της. Ο αλγόριθμος που εξετάζει αν ο $\mathcal{C}$ είναι μοναδικά αποκωδικοποιήσιμος αποτελείται από τα παρακάτω βήματα:*

Αλγόριθμος Sardinas και Patterson

Είσοδος: *Κώδικας* $\mathcal{C}$

1. ***Για κάθε*** *ζεύγος λέξεων που ανήκουν στον κώδικα* $\mathcal{C}$:
   (α′) *Έλεγξε αν η μία αποτελεί πρόθεμα της άλλης.*
   (β′) ***Αν*** *ο προηγούμενο έλεγχος για ένα ζεύγος βγει αληθής:*
      i. *Αφαίρεσε από την μεγαλύτερη σε μήκος κωδική λέξη την μικρότερη που αποτελεί πρόθεμα της και βρες την κατάληξη που προκύπτει*
      ii. *Εισήγαγε την κατάληξη στο σύνολο* $\mathcal{S}^1$
2. ***Αν*** *το* $\mathcal{S}^1 == \emptyset$*:*
   (α′) ***Επέστρεψε*** *True*
3. ***Αλλιώς αν*** *κάποια κωδική λέξη του* $\mathcal{C}$ *ανήκει στο* $\mathcal{S}^1$*:*
   (α′) ***Επέστρεψε*** *False*
4. ***Αλλιώς***
   (α′) *i=1*
   (β′) ***Για κάθε*** *κατάληξη που ανήκει στο σύνολο* $\mathcal{S}^i$:
      i. *Έλεγξε αν η κατάληξη αποτελεί πρόθεμα ή έχει ως πρόθεμα κάποια από τις κωδικές λέξεις του* $\mathcal{C}$
      ii. ***Αν*** *ο προηγούμενο έλεγχος για ένα ζεύγος βγει αληθής:*
         A′. *Βρες την καινούρια κατάληξη που προκύπτει*
         B′. *Εισήγαγε την κατάληξη στο σύνολο* $\mathcal{S}^{i+1}$
   (γ′) ***Αν*** *το* $\mathcal{S}^{i+1} == \emptyset$*:*
      i. ***Επέστρεψε*** *True*
   (δ′) ***Αλλιώς αν*** *κάποια κωδική λέξη του* $\mathcal{C}$ *ανήκει στο* $\mathcal{S}^{i+1}$*:*
      i. ***Επέστρεψε*** *False*
   (ε′) ***Αλλιώς****:*
      i. $i \rightarrow i+1$
      ii. *Επανέλαβε τα βήματα 4(β′) μέχρι 4(ε′)*

Έξοδος: *True* $\vee$ *False*

Πριν προχωρήσουμε στην απόδειξη του συγκεκριμένου αλγορίθμου θα προσπαθήσουμε να καταλάβουμε την διαδικασία του και γιατί πράγματι ελέγχει αν ένας κώδικας είναι μοναδικά αποκωδικοποιήσιμος. Ο αλγόριθμος ξεκινάει ελέγχοντας αν οποιοδήποτε ζεύγος κωδικών λέξεων αποτελούν η μία πρόθεμα της άλλης. Αφού βρει αυτά τα ζεύγη εξάγει τις καταλήξεις που δημιουργούνται και τις αποθηκεύει στο σύνολο $\mathcal{S}^1$*, το οποίο σύμφωνα με τους Sardinas και Patterson ονομάζεται segment 1. Στην συνέχεια αφού βρει το σύνολο των καταλήξεων ελέγχει δύο περιπτώσεις:*

1. *Αν το σύνολο* $\mathcal{S}^1$ *είναι κενό αυτόματα γνωρίζουμε πως ο αλγόριθμος δεν βρήκε κανένα ζεύγος λέξεων του κώδικα στο οποίο η μία αποτελεί πρόθεμα της άλλης. Άρα έπεται ότι ο κώδικας είναι στιγμιαίος οπότε και μοναδικά αποκωδικοποιήσιμος σύμφωνα με το Πόρισμα 3.2 για αυτό και ο αλγόριθμος σε αυτή την*

περίπτωση επιστρέφει True.

2. *Αν μέσα στο σύνολο $\mathcal{S}^1$ περιέχεται τουλάχιστον μια κωδική λέξη του $\mathcal{C}$ αυτό σημαίνει ότι υπάρχει κάποια κατάληξη που είναι ταυτόχρονα και κωδική λέξη. Έστω ότι το ζεύγος των κωδικών λέξεων που έδωσε την συγκεκριμένη κατάληξη είναι το C_i, C_j με $l_i > l_j$ ενώ η κατάληξη που δημιουργήθηκε αντιστοιχεί στην κωδική λέξη C_k. Αυτό πρακτικά σημαίνει ότι $C_i = C_jC_k = C^*$ γεγονός που παραβιάζει τον ορισμό του μοναδικά αποκωδικοποιήσιμου κώδικα καθώς βρήκαμε δύο συμβολοσειρές x_i και x_jx_k που αντιστοιχίζονται στην ίδια κωδική λέξη C^*. Για το λόγο αυτό ο αλγόριθμος σε αυτήν την περίπτωση επιστρέφει False.*

Αν δεν ισχύει καμία από τις δύο περιπτώσει τότε ο αλγόριθμός συνεχίζει βρίσκοντας το σύνολο $\mathcal{S}^2$ το οποίο δημιουργείται ελέγχοντας αν κάποια από τις καταλήξεις του προηγούμενου συνόλου καταλήξεων $\mathcal{S}^)$ αποτελεί πρόθεμα ή έχει ως πρόθεμα κάποια κωδική λέξη του $\mathcal{C}$. Ο λόγος που είναι απαραίτητο να συνεχίσουμε την διαδικασία στον υπολογισμό του $\mathcal{S}^2$ θα γίνει αντιληπτός με ένα παράδειγμα. Έστω ότι έχουμε μία κατάληξη S_1 που ανήκει στο $\mathcal{S}^1$. Αυτό σημαίνει ότι υπάρχουν δύο κωδικές λέξεις, έστω οι C_i, C_j που η μία αποτελεί πρόθεμα τις άλλης. Εμείς χωρίς βλάβη της γενικότητας υποθέτουμε ότι η C_j αποτελεί πρόθεμα της C_i. Πάλι χωρίς βλάβη της γενικότητας υποθέτουμε ότι η κατάληξη S_1 που σχηματίστηκε από τις κωδικές λέξεις C_i, C_j έχει ως πρόθεμα την κωδική λέξη C_k. Τότε η κατάληξη που σχηματίζεται αφαιρώντας την C_k από την S_1 είναι ένα στοιχείο του συνόλου $\mathcal{S}^2$. Θα ονομάσουμε την τελευταία κατάληξη που σχηματίστηκε S_2.

Για να δούμε τι σημαίνουν όλα αυτά που έχουμε βρει μέχρι στιγμής. Η σχέση που συνδέει τις κωδικές λέξεις C_i, C_j και την κατάληξη S_1 είναι η $C_i = C_jS_1$, δηλαδή η C_i δημιουργείται από την παράθεση της κωδικής λέξης C_j με την κατάληξη S_1. Επειδή όμως η S_1 έχει ως πρόθεμα την C_k έπεται ότι $C_i = C_jS_1 = C_jC_kS_2$. Καταλαβαίνουμε λοιπόν πως αν η S_2 είναι κάποια κωδική λέξη πάλι θα παραβιαστεί ο ορισμός της μοναδικής αποκωδικοποιησιμότητας. Άρα στην ουσία ο αλγόριθμός ελέγχει με αυτό τον τρόπο αν κάποιο ζεύγος πιθανών παραθέσεων κωδικών λέξεων παραβιάζει τον ορισμό της μοναδικής αποκωδικοποιησιμότητας. Άρα σε κάθε βήμα ο αλγόριθμος βρίσκει κάποιο υποσύνολο των πιθανών καταλήξεων των κωδικών λέξεων του $\mathcal{C}$. Επειδή το πλήθος των κωδικών λέξων είναι πεπερασμένο έπεται ότι και το πλήθος των καταλήξεων θα είναι πεπερασμένο, άρα και το δυναμοσύνολο των καταλήξεων θα είναι πεπερασμένο. Αυτό πρακτικά σημαίνει ότι σε κάποιο βήμα είτε ο αλγόριθμος θα βρει κάποια κωδική λέξη που ανήκει στο σύνολο των καταλήξεων $\mathcal{S}^i$ ή θα εξαντλήσει το δυναμοσύνολο των καταλήξεων, δηλαδή όλα τα πιθανά υποσύνολα οπότε σε κάποιο πεπερασμένο βήμα δεν θα μπορεί να βρει άλλες καταλήξεις και θα προκύψει ένα σύνολο $\mathcal{S}^i = \emptyset$ που θα τον αναγκάσει να τερματίσει επιστρέφοντας την τιμή True. Ότι θα εξαντλήσουμε τα υποσύνολα των καταλήξεων χωρίς να βρούμε κάποια κωδική λέξη να ανήκει σε αυτά πρακτικά σημαίνει ότι κανένα ζεύγος παραθέσεων κωδικών λέξεων δεν οδηγεί στην ίδια κωδική λέξη, δηλαδή δεν παραβιάζεται ο ορισμός της μοναδικής αποκωδικοποιησιμότητας. Αφού έχουμε καταλάβει την βασική λογική του αλγορίθμου ήρθε η ώρα να δώσουμε την απόδειξη του αλγορίθμου η οποία μας προσφέρει και μία γεωμετρική ερμηνεία στην έννοια της μοναδικής αποκωδικοποιησιμότητας.

Απόδειξη [Ban63]

Σύμφωνα με την θεωρία ένας κώδικας δεν είναι μοναδικά αποκωδικοποιήσιμος αν για κάποια κωδική συμβολοσειρά υπάρχει διφορούμενη μετάφραση. Άρα ένας κώδικας που δεν πληρεί την παραπάνω προϋπόθεση επιτρέπει μία κωδική συμβολοσειρά να προέρχεται από δύο διαφορετικές ακολουθίες τιμών της τυχαίας μεταβλητής. Έστω ότι έχουμε δύο σύνολα κωδικών λέξεων $\mathcal{C}_1, \mathcal{C}_2 \subset \mathcal{C}$, των οποίων η παράθεση των στοιχειών τους οδηγεί στην κοινή κωδική συμβολοσειρά C^*. Τα στοιχεία τις πρώτης ακολουθίας τα συμβολίζουμε με $C_1, C_2, \cdots, C_k$ και τις δεύτερης με $C'_1, C'_2, \cdots, C'_{k'}$. Να διευκρινίσουμε ότι δεν είναι αναγκαίο $k = k'$, δηλαδή η κάθε ακολουθία μπορεί να απαρτίζεται από διαφορετικό πλήθος κωδικών λέξεων. Επειδή όμως η κωδικοποιημένη συμβολοσειρά είναι η ίδια και για τις δύο ακολουθίες έπεται ότι:

1. $\sum_{i=1}^{k} l(C(x_i)) = \sum_{i=1}^{k'} l(C'(x_i))$
2. *Οι κωδικές λέξεις C_k και $C'_{k'}$ θα τελειώνουν στο ίδιο σημείο με το ίδιο σύμβολο.*

Για να οπτικοποιήσουμε την παραπάνω κατάσταση, κάθε κωδική λέξη την σχεδιάζουμε με βάση το μήκος της, δηλαδή το πλήθος των κωδικών συμβόλων από τα οποία αποτελείται, ως ένα ευθύγραμμο τμήμα σε μία ευθεία που έχει μήκος όσο η C^*. Τα σημεία που βρίσκονται στην ευθεία αναπαριστούν την αρχή ή το τέλος κάθε

κωδικής λέξης. Η κωδική λέξη C_1 της ακολουθίας $C_1, C_2, \cdots, C_k$ αναπαριστάται ως το ευθύγραμμο τμήμα AA_1, ενώ οι κωδικές λέξεις C_k, $C'_{k'}$ με τα σημεία A_k, $A'_{k'}$ που αποτελούν το τέλος της κωδικής συμβολοσειράς C^*. Επειδή οι συμβολοσειρές τελειώνουν με το ίδιο σύμβολο έπεται ότι $A_k = A'_{k'}$.

Σχήμα 3.2: Η γραφική αναπαράσταση μιας παράθεσης κωδικών λέξεων με ευθύγραμμα τμήματα

Για την δεύτερη ακολουθία η κωδική λέξη C'_1 της $C'_1, C'_2, \cdots, C'_k$ αναπαρίσταται με το ευθύγραμμο τμήμα AA'_1. Το σημείο A'_1 μπορεί να τοποθετηθεί στα αριστερά του A_1 αν η C_1 έχει μεγαλύτερο μήκος από τη C'_1 ή στα δεξιά του αν συμβαίνει το αντίστροφο.

Σχήμα 3.3: Η περίπτωση κατά την οποία η κωδική λέξη C_1 έχει μεγαλύτερο μήκος από τη C'_1

Σχήμα 3.4: Η περίπτωση κατά την οποία η κωδική λέξη C_1 έχει μικρότερο μήκος από τη C'_1

Στην πρώτη περίπτωση (Σχήμα 3.2) στο σύνολο $\mathcal{S}^1$ εισάγουμε τη κατάληξη A'_1A_1 καθώς η λέξη C'_1 αποτελεί πρόθεμα της C_1, το οποίο μεταφράζεται στο σχήμα με το να περιέχεται το ευθύγραμμο τμήμα AA'_1 στο AA_1. Ενώ αν ισχύσει η δεύτερη περίπτωση (Σχήμα 3.3), στο σύνολο $\mathcal{S}^1$ εισάγουμε την υποσυμβολοσειρά $A_1A'_1$ για τους αντίστοιχους λόγους. Χωρίς βλάβη της γενικότητας υποθέτουμε ότι το A'_1 βρίσκεται αριστερά του A_1. Θα δούμε τώρα πως μπορεί να υπάρξει ένα στοιχείο που ανήκει στο σύνολο $\mathcal{S}^2$.

Σχήμα 3.5: Η περίπτωση κατά την οποία η κωδική λέξη $C'_{2'}$ βρίσκεται πριν από τη C_1

Έστω ότι η κωδική λέξη C'_2 που αναπαριστάται με το ευθύγραμμο τμήμα $A'_1A'_2$ βρίσκεται πριν τη από τη κωδική λέξη C_1 (Σχήμα 3.4) τότε η κατάληξη A'_2A_1 θα ανήκει στο σύνολο $\mathcal{S}^2$.

Σχήμα 3.6: Η περίπτωση κατά την οποία η κωδική λέξη C'_2 βρίσκεται μετά από τη C_1

Σχήμα 3.7: Το πρώτο σημείο A'_r που βρίσκεται μετά το A_1

Αν όμως η κωδική λέξη C'_2 βρίσκεται μετά από τη C_1 (Σχήμα 3.5) τότε η κατάληξη $A_1A'_2$ θα ανήκει στο σύνολο $\mathcal{S}^2$. Χωρίς βλάβη της γενικότητας υποθέτουμε ότι ισχύει πάλι η περίπτωση όπου η κωδική λέξη C'_2 βρίσκεται πριν από τη C_1.

Στην ουσία αυτό που προσπαθούμε να κάνουμε σε αυτή την απόδειξη είναι να διατάξουμε την αρχή και το τέλος κάθε κωδικής λέξης, αναπαριστώντας τα με σημεία πάνω σε ένα ευθύγραμμο τμήμα. Αρχίζουμε σχεδιάζοντας μία ευθεία και βάζοντας τα σημεία $A, A_1, A_2, \cdots, A_k, A'_{1'}, \cdots, A'_{k'}$. Αν ανάμεσα στο ευθύγραμμο τμήμα AA_1 παρεμβάλλεται κάποιο από τα $A'_1, A'_2, \cdots, A'_i$ τότε σίγουρα δημιουργούνται προθέματα και καταλήξεις που καταλήγουν να εμπεριέχονται σε κάποιο σύνολο $\mathcal{S}^i$. Το πλήθος των συνόλων $\mathcal{S}^i$ που εμφανίζονται εξαρτάται από το πόσα σημεία $A'_1, A'_2, \cdots$ παρεμβάλλονται ανάμεσα στα σημεία A και A_1. Για παράδειγμα κατά την αρχή της απόδειξης είδαμε ότι μέσα στη κωδική λέξη C_1 παρεμβλήθηκε η C'_1 το οποίο οδήγησε στη δημιουργία μίας κατάληξης που εισήχθηκε στο $\mathcal{S}^1$. Επειδή υπήρχε και άλλη παρεμβολή που αναγκαστικά σημειώθηκε μετά το C'_1 και πριν τελειώσει η C_1 είδαμε ότι δημιουργήθηκε μία κατάληξη μέσα στην προηγούμενη κατάληξη, γεγονός που με τη σειρά του οδήγησε στην εισαγωγή της καινούριας κατάληξης στο $\mathcal{S}^2$.

Συνεχίζοντας με την ίδια λογική κάποια στιγμή θα συναντήσουμε το A'_r που είναι το πρώτο από τα τονισμένα σημεία μετά το A_1. Τότε θα εισάγουμε στο $\mathcal{S}^r$ την κατάληξη $A_1A'_r$ του ευθυγράμμου τμήματος $A'_{r-1}A'_r$. Έπειτα στο $r+1$ βήμα θα εισάγουμε στο $\mathcal{S}^{r+1}$ την κατάληξη $A_2A'_r$. Μετά από $l-1$ θα εισάγουμε στο $\mathcal{S}^{r+l-1}$ την κατάληξη A'_rA_l όπου A_l το πρώτο σημείο μετά το A'_r. Η κατάληξη A'_rA_l δημιουργείται γιατί η κατάληξη $A_{l-1}A'_r$ που προήλθε από το ακριβώς προηγούμενο βήμα αποτελεί πρόθεμα της κωδικής λέξης που αναπαρίσταται με το ευθύγραμμο τμήμα $A_{l-1}A_l$.

Σχήμα 3.8: Το σχήμα για την κατάληξη $A'_{r+1}A_l$ ή την $A_lA'_{r+1}$

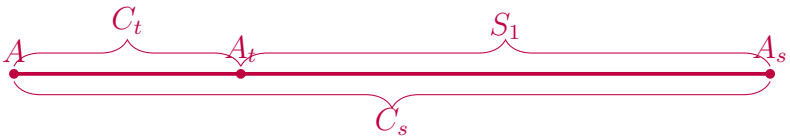
Στο επόμενο βήμα η κατάληξη που θα εισαχθεί στο σύνολο $\mathcal{S}^{r+l}$ θα είναι η $A'_{r+1}A_l$ αν το A'_{r+1} βρίσκεται πριν το A_l καθώς το τμήμα A'_rA_l αποτελεί κατάληξη ή το τμήμα $A_lA'_{r+1}$ αν συμβεί το αντίστροφο. Αν λοιπόν πάρουμε για βάση της επαγωγής το σχηματισμό των καταλήξεων A'_1A_1 ή $A_1A'_1$, σαν επαγωγική υπόθεση την κατάληξη A'_rA_l και σαν επαγωγικό βήμα τις καταλήξεις που μόλις εξάγαμε τότε με επαγωγικό τρόπο έχουμε αποδείξει ότι όλες οι καταλήξεις που μπαίνουν στα διάφορα σύνολα $\mathcal{S})$ έχουν την μορφή $A_iA'j$ ή A'_jA_i.

Καθώς εξελίσσεται η διαδικασία της διάταξης και προχωράει προς το τέλος της, φτάνουμε στο σημείο A_{k-1} το οποίο θεωρούμε ότι βρίσκεται πλησιέστερα στα τερματικά σημεία $A_k, A'_{k'}$. Αν το A_{k-1} αποτελεί το τέλος κάποιας κατάληξης τότε το ευθύγραμμο τμήμα που την αναπαριστά θα έχει ως αρχή το A'_{k-1}. Όμως τότε το τμήμα $A_{k-1}A'_{k'}$ θα αποτελεί κατάληξη της λέξης $C'_{k'}$ και για το λόγο αυτό θα ανήκει στο $\mathcal{S}^k$. Όμως $C_k = C'_{k'}$ από το οποίο συνεπάγεται ότι $A_{k-1}A_{k'} = A_{k-1}A_k$ που αναπαριστά τη κωδική λέξη C_k. Άρα βρήκαμε ότι αν ένας κώδικας δεν είναι μοναδικά αποκωδικοποιήσιμος τότε κάποια κατάληξη αποτελεί κωδική λέξη.

Σχήμα 3.9: Το σημείο A_{k-1} που βρίσκεται πλησιέστερα στα τερματικά A_k και A'_k.

Για να αποδείξουμε το αντίστροφο πρέπει να δείξουμε πως αν κάποια κατάληξη είναι και κωδική λέξη τότε ο κώδικας δεν είναι μοναδικά αποκωδικοποιήσιμος. Χωρίς βλάβη της γενικότητας υποθέτουμε ότι μία τέτοια

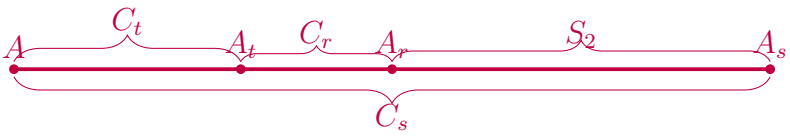
περίπτωση συμβαίνει στα στοιχεία του συνόλου $\mathcal{S}^3$, δηλαδή υπάρχει μία κατάληξη, έστω η S_3, που είναι και κωδική λέξη την οποία θα τη συμβολίσουμε με C_p ($S_3 = C_p$). Για να συμβαίνει αυτό θα πρέπει κατά την αρχή του αλγορίθμου να υπήρχαν δύο κωδικές λέξεις έστω οι C_s, C_t όπου η μία αποτέλεσε πρόθεμα της άλλης δημιουργώντας την κατάληξη $S_1 = C_s - C_t$. Χωρίς βλάβη της γενικότητας υποθέτουμε ότι η C_t αποτέλεσε το πρόθεμα.



Σχήμα 3.10: $C_s = C_t + S_1$

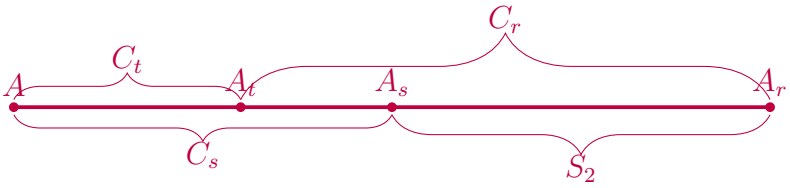
Επειδή υπάρχει $\mathcal{S}^1$ και $\mathcal{S}^2$ καταλαβαίνουμε ότι η κατάληξη S_1 θα αποτελεί πρόθεμα ή θα έχει η ίδια κάποια άλλη κωδική λέξη σαν πρόθεμα δημιουργώντας την κατάληξη S_2 με τους εξής δύο τρόπους:

1. $S_2 = S_1 - C_r$, όπου θεωρούμε ότι η κατάληξη S_1 έχει πρόθεμα ή:



Σχήμα 3.11: $C_s = C_t + C_r + S_2$

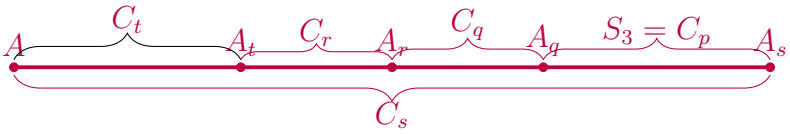
2. $S_2 = C_r - S_1$, όπου θεωρούμε ότι η κατάληξη S_1 είναι πρόθεμα.



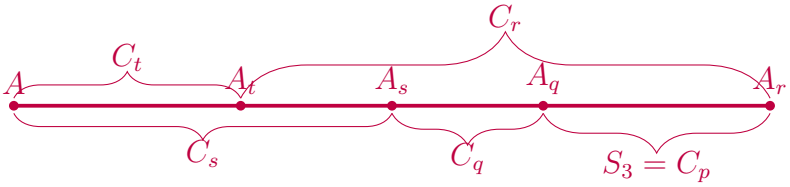
Σχήμα 3.12: $C_s + S_2 = C_t + C_r$

Όπως και πριν, η ύπαρξη του $\mathcal{S}^2$ μαζί με το $\mathcal{S}^3$ προϋποθέτουν ότι:

1. Η κατάληξη S_2 θα έχει ως πρόθεμα κάποια C_q ή:

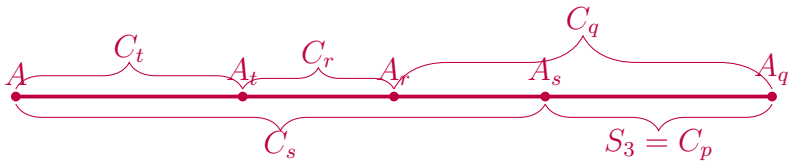


Σχήμα 3.13: $C_s = C_t + C_r + C_q + C_p$

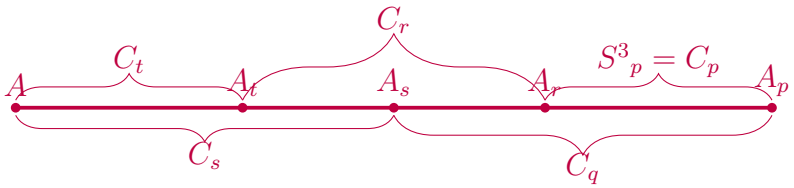


Σχήμα 3.14: $C_t + C_r = C_s + C_q + C_p$

2. Η κατάληξη S_2 θα αποτελεί πρόθεμα για κάποια C_q :



Σχήμα 3.15: $C_s + C_p = C_t + C_r + C_q$



Σχήμα 3.16: $C_s + C_q = C_t + C_r + C_p$

Χωρίς βλάβη της γενικότητας υποθέτουμε ότι ισχύουν οι εξισώσεις:
$S_1 = C_s - C_t$ *(1) (Σχήμα 3.10)*
$S_2 = S_1 - C_r$ *(2) (Σχήμα 3.11)*
$S_3 = C_p = C_q - S_2$ *(3) (Σχήμα 3.16)*

Χρησιμοποιώντας τις (1) και (2) απαλείφουμε την κατάληξη S_1 και έχουμε $S_2 = C_s - C_t - C_r$. Αντικαθιστώντας στην (3) έχουμε: $C_p = C_q - C_s + C_t + C_r \Rightarrow C_p + C_s = C_q + C_t + C_r$ (Σχήμα 1.13). Είναι εύκολο να φανταστούμε ότι η διαδικασία αυτή μπορεί να επεκταθεί και στην περίπτωση που ο αλγόριθμος αποφασίζει ότι ο κώδικας δεν είναι μοναδικά αποκωδικοποιήσιμος κατά τη $n-$οστή επανάληψη. Αυτό που θα δημιουργηθεί πάλι είναι ένα γραμμικό σύστημα n εξισώσεων, το οποίο όταν απαλείψουμε τις n καταλήξεις $S_1, \cdots, S_n$ θα προκύψει μία ισότητα μεταξύ δύο ακολουθιών κωδικών λέξεων γεγονός που αποδεικνύει ότι ο κώδικας μας δεν είναι μοναδικά αποκωδικοποιήσιμος.

Παράδειγμα 3.1. *Δίνονται οι κώδικές $\mathcal{C}_1 = \{01, 001, 000\}$, $\mathcal{C}_2 = \{0, 01, 11, 010\}$ και $\mathcal{C}_3 = \{00, 01, 11, 010, 0011\}$. Να εξελέγξετε κάνοντας χρήση του αλγορίθμου Sardinas - Patterson ποιος κώδικας είναι μοναδικά αποκωδικοποιήσιμος.*

Στον κώδικα $\mathcal{C}_1$ καμία λέξη δεν αποτελεί πρόθεμα κάποιας άλλης άρα είναι μοναδικά αποκωδικοποιήσιμος ως στιγμιαίος. Για τον κώδικα $\mathcal{C}_2$ η λέξη 0 αποτελεί πρόθεμα των κωδικών λέξεων $\{01, 010\}$, οπότε στο σύνολο $\mathcal{S}^1$ αποτελείται από τις καταλήξεις $\{1, 10\}$. Ελέγχοντας αν τα στοιχεία του $\mathcal{S}^1$ αποτελούν προθέματα των κωδικών λέξεων ή έχουν ως πρόθεμα κάποια κωδική λέξη, βλέπουμε ότι αυτό συμβαίνει για την λέξη 11. Επειδή όμως η κατάληξη της 11 με πρόθεμα τη λέξη 1 είναι το 1 που ήδη υπάρχει στο σύνολο $\mathcal{S}^1$, δεν εισέρχεται στο σύνολο $\mathcal{S}^2$ με αποτέλεσμα ο αλγόριθμός να μας απαντήσει καταφατικά καθώς το $\mathcal{S}^2 == \emptyset$.

Τέλος για τον κώδικα $\mathcal{C}_3$ οι λέξεις 00,01 αποτελούν προθέματα των κωδικών λέξεων 010,0011 αντίστοιχα με αποτέλεσμα να δημιουργηθεί το σύνολο $\mathcal{S}^1 = \{10, 11\}$. Επειδή όμως το $\mathcal{S}^1$ περιέχει κωδική λέξη ο αλγόριθμός δίνει αρνητική απάντηση.

Θεώρημα 3.2. ***(Η ανισότητα του McMillan)*** *Έστω μία τυχαία μεταβλητή X που παίρνει τιμές σε ένα περασμένο σύνολο $\mathcal{X}$, F το κωδικό αλφάβητο και l_i το μήκος της κωδική λέξης $C(x_i) \in F^*$ που αντιστοιχεί στην τιμή $x_i \in \mathcal{X}$. Τότε ο κώδικας $\mathcal{C} = \{C(x_i)\}_{x_i \in \mathcal{X}}$ για την τυχαία μεταβλητή X είναι μοναδικά αποκωδικοποιήσιμος αν:*

$$\sum_{x_i \in \mathcal{X}} |F|^{-l_i} \leq 1 \tag{3.5}$$

Αντιστρόφως δοθέντος ενός συνόλου με μήκη κωδικών λέξεων που ικανοποιούν την παραπάνω ανισότητα, είναι δυνατή η κατασκευή ενός μοναδικά αποκωδικοποιήσιμου κώδικα του οποίου οι κωδικές λέξεις έχουν τα αντίστοιχα μήκη.

Απόδειξη

Έχοντας σαν δεδομένο ότι ο κώδικας είναι μοναδικά αποκωδικοποιήσιμος, το ζητούμενο είναι να αποδείξουμε ότι:

$$\sum_{x_i \in \mathcal{X}} |F|^{-l_i} \leq 1$$

Από τον ορισμό της έννοιας του μοναδικά αποκωδικοποιήσιμου κώδικα, ξέρουμε ότι υπάρχει μία αμφιμονοσήμαντη αντιστοιχία μεταξύ των συμβολοσειρών που παράγονται από τις τιμές της τυχαίας μεταβλητής και των κωδικών συμβολοσειρών. Υπενθυμίζουμε ότι η κωδική συμβολοσειρά αποτελείται από την παράθεση των κωδικών λέξεων των αντίστοιχων τιμών της τυχαίας μεταβλητή, $C^*(x_1x_2\dots x_n) = C(x_1)C(x_2)\dots C(x_n)$. Άρα το μήκος της κωδικής λέξης για μία συμβολοσειρά $x_1x_2\dots x_n$ θα αποτελείται από το άθροισμα μηκών των κωδικών λέξεων των τιμών της τυχαίας μεταβλητής που συμμετέχουν στη συμβολοσειρά.

$$l(C^*(x_1x_2\dots x_n)) = \sum_{i=1}^{n} l_i$$

Η ανισοϊσότητα (3.5) αποτελεί ένα φράγμα για τα μήκη των κωδικών λέξεων που προήλθαν από συμβολοσειρές μήκους 1, δηλαδή για τα $l_i = l(C(x_i)),\ με\ x_i \in \mathcal{X}$. Αν έχουμε μία κωδική συμβολοσειρά τότε τα μήκη των κωδικών λέξεων που συμμετέχουν σε αυτή αποτελούν ένα υποσύνολο των $\{l_i\}, i = 1, 2, \dots |\mathcal{X}|$. Για το παραπάνω σύνολο η μη αρνητική ποσότητα $|F|^{-l_i},\ \forall i$ είναι φραγμένη καθώς το άθροισμα τους είναι φραγμένο.

Επειδή ισχύει:

$$l_i \leq \sum_{i=1}^{n} l_i \Rightarrow -l_i \geq -\sum_{i=1}^{n} l_i \Rightarrow |F|^{-l_i} \geq |F|^{-\sum_{i=1}^{n} l_i}\ \forall i.$$

, έπεται ότι και η ποσότητα $|F|^{-\sum_{i=1}^{n} l_i}\ \forall i$ θα είναι φραγμένη. Αυτό πρακτικά σημαίνει ότι για κάθε συμβολοσειρά πεπερασμένου μήκους η ποσότητα $\sum_{\mathbf{x_1^n} \in \mathcal{X}^n} |F|^{-\sum_{i=1}^{n} l_i}$ θα είναι φραγμένη. Με την έκφραση $\mathbf{x_1^n}$ συμβολίζουμε τις συμβολοσειρές μήκους n. Οπότε αντί να βρούμε ένα φράγμα για τις κωδικές λέξεις που προήλθαν από συμβολοσειρές μήκους 1 (τις τιμές της τυχαίας μεταβλητής), θα αποδείξουμε ότι η συγκεκριμένη ποσότητα, $\sum_{\mathbf{x_1^n} \in \mathcal{X}^n} |F|^{-\sum_{i=1}^{n} l_i}$, έχει φράγμα για οποιοδήποτε n άρα και για $n = 1$. Η έκφραση $\sum_{\mathbf{x_1^n} \in \mathcal{X}^n} |F|^{-l(C^*(\mathbf{x_1^n}))}$ αποτελεί την μετάφραση της ανισοϊσότητας $\sum_{x_i \in \mathcal{X}} |F|^{-l_i} \leq 1$ για συμβολοσειρές μήκους n

Αν αναπτύξουμε τον εκθέτη της τελευταίας σχέση έχουμε:

$$\sum_{\mathbf{x_1^n} \in \mathcal{X}^n} |F|^{-l(C^*(\mathbf{x_1^n}))} = \sum_{\mathbf{x_1^n} \in \mathcal{X}^n} |F|^{-\sum_{k=1}^{n} l_k} = \sum_{\mathbf{x_1^n} \in \mathcal{X}^n} |F|^{-(l_1+l_2+\dots+l_n)} = \sum_{\mathbf{x_1^n} \in \mathcal{X}^n} |F|^{-l_1-l_2-\dots-l_n} =$$
$$|F|^{-l_1} \sum_{\mathbf{x_1^n} \in \mathcal{X}^n} |F|^{-l_1} \times |F|^{-l_2} \times \dots \times |F|^{-l_n}$$

Για να βρούμε τον πληθάριθμο των συμβολοσειρών μήκους n, αρκεί να υπολογίσουμε το άθροισμα:

$$\sum_{\mathbf{x_1^n} \in \mathcal{X}^n} 1$$

.

Ένας άλλος τρόπος για να υπολογίσουμε τον ίδιο πληθάριθμο είναι να βρούμε το άθροισμα:

$$\sum_{x_1\in\mathcal{X}}\sum_{x_2\in\mathcal{X}}\cdots\sum_{x_n\in\mathcal{X}}1$$

.

Το τελευταίο άθροισμα στην ουσία μετράει πόσες επιλογές έχουμε για κάθε ένα από τα n σύμβολα. Αν για παράδειγμα η τυχαία μεταβλητή αναπαριστά μία πηγή χωρίς περιορισμούς για κάθε θέση της συμβολοσειράς έχουμε $|\mathcal{X}|$ επιλογές και τότε $\sum_{x_1\in\mathcal{X}}\sum_{x_2\in\mathcal{X}}\cdots\sum_{x_n\in\mathcal{X}}1=|\mathcal{X}|^n$. Αν όμως μιλάμε για μία πηγή με περιορισμούς τότε $\sum_{x_1\in\mathcal{X}}\sum_{x_2\in\mathcal{X}}\cdots\sum_{x_n\in\mathcal{X}}1<|\mathcal{X}|^n$. Μια πηγή με περιορισμούς αποτελεί η ελληνική γλώσσα όπου καμία λέξη της δεν αρχίζει με το τελικό σίγμα (ς).

Με βάση την παραπάνω καταμέτρηση των συμβολοσειρών μήκους n η ζητούμενη σχέση για συμβολοσειρές γράφεται ως:

$$\sum_{\mathbf{x_1^n}\in\mathcal{X}^n}|F|^{-l_1}\times|F|^{-l_2}\times\cdots\times|F|^{-l_n}=\sum_{x_1\in\mathcal{X}}\sum_{x_2\in\mathcal{X}}\cdots\sum_{x_n\in\mathcal{X}}|F|^{-l_1}\times|F|^{-l_2}\times\cdots\times|F|^{-l_n}=$$
$$=\sum_{x_1\in\mathcal{X}}|F|^{-l_1}\times\sum_{x_2\in\mathcal{X}}|F|^{-l_2}\times\cdots\times\sum_{x_n\in\mathcal{X}}|F|^{-l_n}=\left(\sum_{x_i\in\mathcal{X}}|F|^{-l_i}\right)^n$$

Θα αξιοποιήσουμε τον κλάδο της συνδυαστικής για να εκφράσουμε την ποσότητα $\left(\sum_{x_i\in\mathcal{X}}|F|^{-l_i}\right)^n$ και με έναν διαφορετικό τρόπο. Το παραπάνω άθροισμα είπαμε ότι μετράει πάνω σε όλες τις συμβολοσειρές μήκους n. Επειδή ο κώδικας είναι μοναδικά αποκωδικοποιήσιμος κάθε συμβολοσειρά της μορφής $x_1x_2\dots x_n$ θα αντιστοιχίζεται σε μία μοναδική κωδική λέξη $C(x_1)C(x_2)\dots C(x_n)$. Όμως ενώ όλες οι συμβολοσειρές έχουν το ίδιο μήκος n, δεν ισχύει το ίδιο για τις κωδικές λέξεις αφού το μήκος τους αποτελείται από το άθροισμα μηκών ενός διαφορετικού κάθε φορά υποσυνόλου κωδικών λέξεων. Άρα αντί να μετράμε με βάση το πλήθος των διαφορετικών συμβολοσειρών μήκους n, μπορούμε να μετράμε με βάση το πλήθος των διαφορετικών μηκών των κωδικών λέξεων. Αν διατάξουμε τα μήκη $\{l_i\},\ i=1,\dots,|\mathcal{X}|$ σίγουρα το $1\le l_{min}\le l_i\le l_{max}$. Άρα η πιο "σύντομη" κωδική λέξη θα έχει μήκος 1 και η μεγαλύτερη θα έχει μήκος $n\cdot l_{max}$. Οπότε το 1 και το $n\cdot l_{max}$ αποτελούν τα άκρα του αθροίσματος που μετράει με βάση το μήκος των κωδικών λέξεων. Επειδή όμως κάθε μήκος προέρχεται από ένα διαφορετικό σύνολο συμβολοσειρών, μπορούμε να πούμε ότι το μήκος επιβάλει μία διαμέριση στο σύνολο των συμβολοσειρών μήκους n.

Πρόταση 3.1. *Έστω A_k με $k=1,2,...,n\cdot l_{max}$ το σύνολο που περιέχει τις συμβολοσειρές $x_1x_2\dots x_n$ οι οποίες αντιστοιχίζονται σε κωδικές λέξεις μήκους k ενός μοναδικά αποκωδικοποιήσιμου κώδικα, δηλαδή $A_k=\{x_1x_2\dots x_n : l(C^*(x_1x_2\dots x_n))=k\}$. Τότε θα ισχύει $A_i\cap A_j=\emptyset,\ \forall i\neq j$ με $i,j=1,2,\dots,n\cdot l_{max}$.*

Απόδειξη

Έστω ότι υπάρχουν i,j για τα οποία ισχύει $A_i\cap A_j\neq\emptyset$. Τότε θα υπάρχει μία τουλάχιστον συμβολοσειρά $x_1x_2\dots x_n$ τέτοια ώστε $x_1x_2\dots x_n\in A_i\wedge x_1x_2\dots x_n\in A_j$. Αυτό όμως πρακτικά σημαίνει ότι η συμβολοσειρά θα αντιστοιχίζεται σε μία κωδική λέξη μήκους i αφού ανήκει στο A_i και σε μία κωδική λέξη μήκους j αφού ανήκει και στο A_j. Άρα η συμβολοσειρά θα αντιστοιχίζεται σε δύο κωδικές λέξεις γεγονός που παραβιάζει την αμφιμονοσήμαντη αντιστοιχία μεταξύ συμβολοσειρών και κωδικών λέξεων και κατ΄ επέκταση τον ορισμό του μοναδικού αποκωδικοποιήσιμου κώδικά. Επειδή η παραπάνω υπόθεση οδηγεί σε άτοπο έπεται ότι $A_i\cap A_j=\emptyset$

Από την παραπάνω πρόταση έπεται ότι $\sum_{k=1}^{n\cdot l_{max}}|A_k|=|\mathcal{X}^n|$, όπου $|\mathcal{X}^n|$ ο πληθάριθμος των συμβολοσειρών μήκους n, καθώς η οικογένεια συνόλων $\mathcal{A}=\{A_k\}_{k=1}^{n\cdot l_{max}}$ αποτελεί μία διαμέριση του συνόλων των συμβολοσειρών μήκους n. Με βάση τα παραπάνω μπορούμε να παρατηρήσουμε ότι ο όρος $|F|^{-l(C^*(\mathbf{x_1^n}))}$, για ένα συγκεκριμένο μήκος έστω k, όταν μετράμε με βάση της συμβολοσειρές $\mathbf{x_1^n}$ μήκους n θα παρουσιαστεί $|A_k|$ φορές, δηλαδή $||F|^{-k}|=|A_k|,\ \mu\epsilon\ k=1,2,\dots n\cdot l_{max}$. Οπότε το άθροισμα $\sum_{\mathbf{x_1^n}\in\mathcal{X}^n}|F|^{-l(C^*(\mathbf{x_1^n}))}=\left(\sum_{x_i\in\mathcal{X}}|F|^{-l_i}\right)^n$ αν μετρήσουμε με βάση το μήκος των κωδικών λέξεων διαμορφώνεται ως εξής:

$$\sum_{\mathbf{x_1^n}\in\mathcal{X}^n} |F|^{-l(C(\mathbf{x_1^n}))} = \left(\sum_{x_i\in\mathcal{X}} \|F|^{-l_i}\right)^n = \sum_{k=1}^{n\cdot l_{max}} |A_k|\cdot|F|^{-k} \tag{3.6}$$

Όμως το πλήθος των κωδικών λέξεων μήκους k δεν γίνεται να υπερβεί το $|F|^k$,δηλαδή $|A_k| \le |F|^k$ άρα

$$\begin{aligned}&\left(\sum_{x_i\in\mathcal{X}} \|F|^{-l_i}\right)^n = \sum_{k=1}^{n\cdot l_{max}} |A_k|\cdot|F|^{-k} \le \sum_{k=1}^{n\cdot l_{max}} |F|^k\cdot|F|^{-k} = \sum_{k=1}^{n\cdot l_{max}} |F|^{k-k} = \sum_{k=1}^{n\cdot l_{max}} 1 = n\cdot l_{max} \Rightarrow \\ &\left(\sum_{x_i\in\mathcal{X}} \|F|^{-l_i}\right)^n \le n\cdot l_{max}\end{aligned} \tag{3.7}$$

Αν ισχύει $\sum_{x_i\in\mathcal{X}} \|F|^{-l_i} > 1$ τότε η ποσότητα $\left(\sum_{x_i\in\mathcal{X}} \|F|^{-l_i}\right)^n$ θα αυξανόταν με εκθετικό ρυθμό, ενώ η ποσότητα $n\cdot l_{max}$ αυξάνεται γραμμικά, γεγονός που παραβιάζει την ανισότητα $\left(\sum_{x_i\in\mathcal{X}} \|F|^{-l_i}\right)^n \le n\cdot l_{max}$. Αναγκαστικά λλοιπόν πρέπει $\sum_{x_i\in\mathcal{X}} \|F|^{-l_i} \le 1$

Πριν αποδείξουμε το αντίστροφο του θεωρήματος και εντρυφήσουμε στη ερμηνεία του θα αποδείξουμε ότι το ακριβώς ίδιο κριτήριο ισχύει και για στιγμιαίους κώδικες και ονομάζεται ανισότητα του Kraft.

Θεώρημα 3.3. ***(Η ανισότητα του Kraft)*** *Έστω X μία τυχαία μεταβλητή που παίρνει τιμές σε ένα πεπερασμένο σύνολο $\mathcal{X}$, ένα κωδικό αλφάβητο F και $\{l_i\}_{i=1}^{|\mathcal{X}|}$ τα μήκη των κωδικών λέξεων που αντιστοιχούν στις τιμές της τυχαίας μεταβλητή X. Τότε για οποιονδήποτε στιγμιαίο κώδικα ισχύει:*

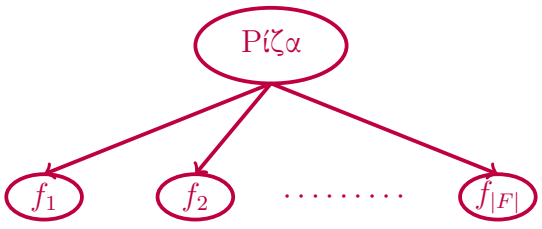
$$\sum_{x_i\in\mathcal{X}} |F|^{-l_i} \le 1 \tag{3.8}$$

Αντιστρόφως αν τα μήκη ενός σύνολου κωδικών λέξεων ικανοποιούν την παραπάνω ανισότητα, τότε μπορούμε να κατασκευάσουμε ένα στιγμιαίο κώδικα με βάση τα μήκη αυτά.

Απόδειξη

Ήδη από τα παραδείγματα, έχουμε δει ότι ένας στιγμιαίος κώδικας μπορεί να αναπαρασταθεί μέσω ενός δένδρου. Στο παράδειγμα της εισαγωγής επειδή το κωδικό αλφάβητό είχε πληθάριθμο 2, χρησιμοποιήσαμε ένα δυαδικό δένδρο, τώρα που ο πληθάριθμος του κωδικού αλφαβήτου είναι $|F|$, θα χρησιμοποιήσουμε ένα $|F|$-αδικό δένδρο[1]*.*

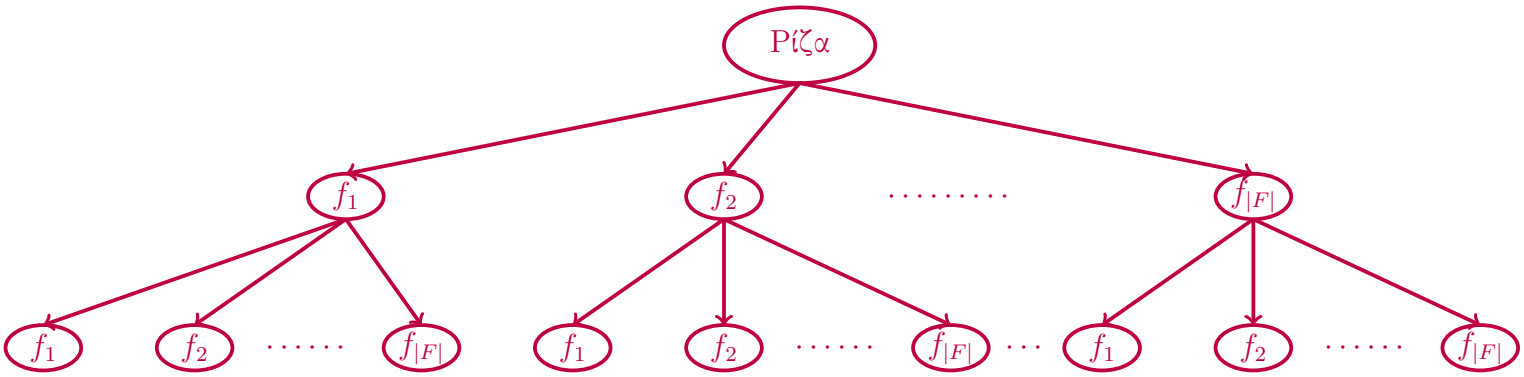
Η αναπαράσταση των στιγμιαίων κωδικών με δένδρα είναι απλά η οπτικοποίηση των πιθανών κωδικών λέξεων που μπορεί να προκύψουν. Για παράδειγμα μία κωδική λέξη μήκους k θα έχει γενική μορφή $f_1 f_2 \dots f_k$, με $f_i \in F\ \forall i = 1, 2, \dots k$. Γνωρίζουμε ότι για το πρώτο σύμβολο έχουμε $|F|$ επιλογές. Το γεγονός αυτό αναπαρίσταται με τη ρίζα ενός δένδρου από την οποία βγαίνουν $|F|$ παιδιά.



Σχήμα 3.17: Το πλήθος των επιλογών για το πρώτο σύμβολο μια κωδικής λέξης

Για το δεύτερο σύμβολο έχουμε πάλι $|F|$ επιλογές. Άρα κάθε κόμβος του πρώτου επιπέδου του δένδρου θα έχει $|F|$ παιδιά.

[1]Με την έννοια $|F|$-αδικό δένδρο εννοούμε ένα δένδρο του οποίου κάθε κόμβος μπορεί να έχει μέχρι $|F|$ παιδιά.



Σχήμα 3.18: Το πλήθος των επιλογών για το δεύτερο σύμβολο μια κωδικής λέξης

Άρα στο δεύτερο επίπεδο που συμβολίζει τις κωδικές λέξεις μήκους 2 έχουμε $|F|^2$ κόμβους γιατί κάθε κόμβος του πρώτου επιπέδου έχει $|F|$ παιδιά και το πρώτο επίπεδο έχει $|F|$ κόμβους, άρα:

$$\text{Πλήθος κόμβων δευτέρου επιπέδου} = \underbrace{|F| + |F| + \cdots |F|}_{|F| \text{φορές}} = |F| \cdot |F| = |F|^2$$

Με την ίδια λογική μπορούμε να καταλάβουμε ότι στο επίπεδο k, θα έχουμε $|F|^k$ κόμβους, αποτέλεσμα που συμφωνεί με τις γνώσεις μας από τον κλάδο της συνδυαστικής. Οπότε οι πιθανές κωδικές λέξεις μήκους k μπορούν να αναπαρασταθούν ως ένα πλήρες[2] $|F|$-αδικό δένδρο k επιπέδων ενώ το κάθε μονοπάτι από τη ρίζα ως το αντίστοιχο φύλλο (κόμβος που δεν έχει παιδιά) αναπαριστά μία συγκεκριμένη κωδική λέξη. Άρα το $|F|$-αδικό δένδρο που οπτικοποιεί έναν στιγμιαίο κώδικα αποτελείται από το σύνολο των μονοπατιών $\{p_i\}$ με $i = 1, \cdots |\mathcal{X}|$ που αποτελούν τις αναπαραστάσεις των κωδικών λέξεων των τιμών της τυχαίας μεταβλητής.

Το σημαντικότερο βήμα στην απόδειξη αυτή, είναι να καταλάβουμε τι σημαίνει για τη δομή του δένδρου μας ότι ο κώδικας είναι στιγμιαίος. Ο πυρήνας ενός στιγμιαίου κώδικα είναι ότι καμία κωδική λέξη δεν αποτελεί πρόθεμα κάποια άλλης. Προ ολίγου αναφέραμε ότι κάθε μονοπάτι στο $|F|$-αδικό δένδρο που ξεκινάει από την ρίζα και φτάνει μέχρι κάποιο από τα φύλλα του δένδρου αποτελεί μία κωδική λέξη για την τιμή της τυχαίας μεταβλητής που βρίσκεται στο αντίστοιχο φύλλο. Το σημαντικότερο στοιχείο στην προηγούμενη πρόταση είναι ότι η κωδική λέξη ολοκληρώνεται όταν φτάνουμε σε κάποιο φύλλο του δένδρου. Τι θα γίνει όμως αν παραβιαστεί η παραπάνω συνθήκη; Την απάντηση μας τη δίνει η παρακάτω πρόταση.

Πρόταση 3.2. *Έστω ότι έχουμε ένα $|F|$-αδικό δένδρο που αναπαριστά τον στιγμιαίο κώδικα για μια τυχαία μεταβλητή X που παίρνει τιμές σε ένα πεπερασμένο σύνολο $\mathcal{X}$ και ένα κωδικό αλφάβητο F. Τότε για κάθε κωδική λέξη ισχύει ότι :*

1. *Ξεκινάει από τον ρίζα του δένδρου.*
2. *Ολοκληρώνεται όταν φτάσουμε σε κάποιο φύλλο του δένδρου.*
3. *Η κάθε κωδική λέξη δεν αποτελεί πρόγονο ή απόγονο καμίας άλλης λέξης.*

Απόδειξη

Η απόδειξη του (1) προκύπτει εύκολα από τον τρόπο που κατασκευάζουμε το $|F|$-αδικό δένδρο ώς το σύνολο των μονοπατιών $\{p_i\}$ με $i = 1, \cdots |\mathcal{X}|$ που αποτελούν τις αναπαραστάσεις των κωδικών λέξεων. Για να αποδείξουμε τα (2) και (3) θα δουλέψουμε με απαγωγή σε άτοπο. Υποθέτουμε ότι δεν ισχύει το (2), δηλαδή ότι υπάρχει μία κωδική λέξη έστω η $C(x_k)$, η οποία δεν ολοκληρώνεται όταν φτάσουμε σε κάποιο φύλλο του δένδρου. Επειδή τα φύλλα αποτελούν τους τερματικούς κόμβους έπεται ότι η $C(x_k)$ θα τερματίζεται σε κάποιο εσωτερικό κόμβο του δένδρου, έστω τον N_k. Επειδή ο N_k είναι εσωτερικός κόμβος συνεπάγεται ότι θα έχει μέχρι $|F|$ παιδιά ή αλλιώς μέχρι $|F|$ μονοπάτια μήκους 1. Οι τελευταίοι το πολύ $|F|$ κόμβοι μπορεί να είναι είτε φύλλα είτε εσωτερικοί κόμβοι του δένδρου. Έστω l_k μήκος του μονοπατιού p_k από τη ρίζα μέχρι τον

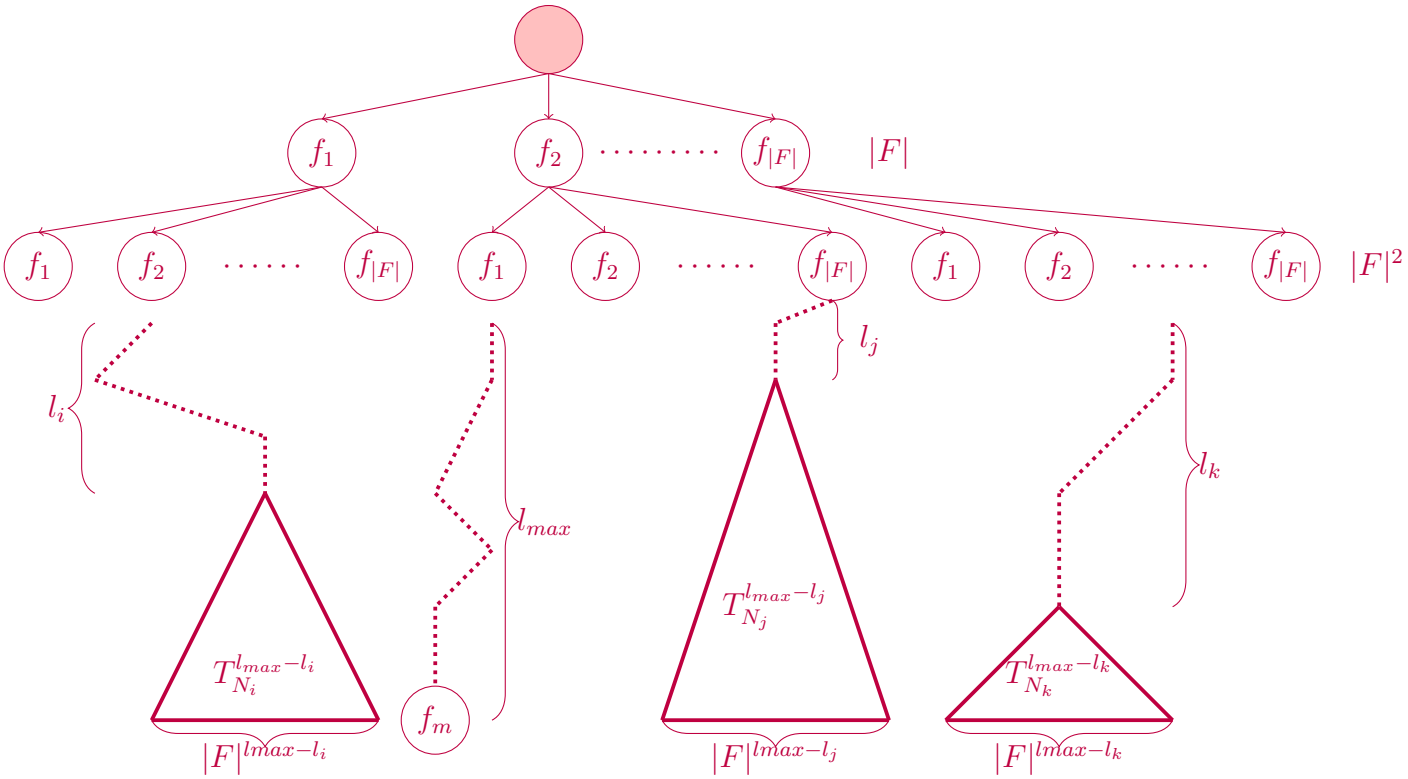
[2] Ένα $|F|$-αδικό δένδρο λέγεται πλήρες αν οι κόμβοι κάθε επιπέδου πλην του τελευταίου έχουν ακριβώς $|F|$ παιδιά.

εσωτερικό κόμβο (N_k) που τερματίζεται η κωδική λέξη και l_{max} το βάθος[3] *του δένδρου. Τότε στο επίπεδο l_{max} το δένδρο θά έχει το πολύ μέχρι $|F|^{l_{max}-l_k}$ απογόνους ή μονοπάτια μήκους το πολύ $l_{max}-l_k$. Όμως το δένδρο αποτελείται από τα μονοπάτια που ξεκινούν από τη ρίζα του δένδρου, ολοκληρώνονται στα φύλλα και αναπαριστούν τις κωδικές λέξεις, άρα έπεται ότι κάποιο από τα $|F|^{l_{max}-l_k}$ μονοπάτια, έστω το p_t, θα αποτελεί μέρος μίας άλλης κωδικής λέξης έστω της $C(x_t)$. Τότε όμως η $C(x_t)$ θα αποτελεί απόγονο της $C(x_k)$ γεγονός που είναι ισοδύναμο με το να είναι η $C(x_k)$ πρόθεμα της $C(x_t)$. Άτοπο γιατί ο κώδικας είναι στιγμιαίος. Άρα όλες οι κωδικές λέξεις ολοκληρώνονται όταν φτάσουμε σε κάποιο φύλλο του δένδρου.*

Έστω υπάρχει κάποια κωδική λέξη, η $C(x_k)$, η οποία δεν ικανοποιεί το(3), δηλαδή αποτελεί είτε πρόγονο είτε απόγονο κάποιας άλλης λέξης. Αν αποτελεί πρόγονο κάποιας άλλης λέξης τότε αυτό σημαίνει ότι δεν τερματίζεται σε φύλλο αλλά σε εσωτερικό κόμβο και την κάνει αυτόματα πρόθεμα της λέξης αυτής γεγονός που αντιφάσκει με τον ορισμό του στιγμιαίου κώδικα. Αν αποτελεί απόγονο κάποιας άλλης κωδικής λέξης τότε αυτόματα σημαίνει ότι έχει ως πρόθεμα την λέξη αυτή γεγονός που πάλι αντικρούεται από τον ορισμό του στιγμιαίου κώδικα. Οπότε και στις δύο περιπτώσεις οδηγούμαστε σε άτοπο.

Πόρισμα 3.3. *Έστω ότι έχουμε ένα $|F|$-αδικό δένδρο που αναπαριστά τον στιγμιαίο κώδικα για μια τυχαία μεταβλητή X που παίρνει τιμές σε ένα πεπερασμένο σύνολο $\mathcal{X}$ και ένα κωδικό αλφάβητο F. Τότε τα σύνολα των απογόνων των κωδικών λέξεων είναι ξένα μεταξύ τους.*

Απόδειξη



Σχήμα 3.19: Η οπτικοποίηση ενός τυχαίο στιγμιαίου κώδικα που αναπαριστάται από τα μονοπάτια ενός $|F|$-αδικού δένδρου. Τα τρίγωνα είναι τα υπόδενδρα που απενεργοποιούνται όταν τελειώνει κάποια κωδική λέξη σε ένας ύψος $l < l_{max}$. Τα υπόδενδρα αυτά έχουν ως ρίζα τον τερματικό κόμβο του μονοπατιού που αναπαριστά την λέξη

Από το προηγούμενο θεώρημα γνωρίζουμε ότι κάθε κωδική λέξη ξεκινάει από τη ρίζα, ολοκληρώνεται σε κάποιο φύλλο του δένδρου και δεν αποτελεί πρόγονο η απόγονο καμίας άλλης λέξης. Έστω l_{max} το μήκος της μεγαλύτερης κωδικής λέξης και συνάμα το βάθος του δένδρου. Υποθέτουμε ότι στο επίπεδο $k < l_{max}$,

[3]Με τον όρο βάθος ενός δένδρου εννοούμε το μεγαλύτερό μονοπάτι από τη ρίζα στα φύλλα του δένδρου

ολοκληρώνεται η κωδική λέξη $C(x_k)$ στον κόμβο N_k. Από το προηγούμενο θεώρημα γνωρίζουμε ότι ο N_k αποτελεί φύλλο του δένδρου.

Αν θέλαμε να βρούμε πόσοι θα ήταν οι απόγονοι του N_k αν δεν τερματιζόταν η λέξη στο επίπεδο k, θα έπρεπε να μετρήσουμε τους κόμβους του υποδένδρου που έχει ως ρίζα τον κόμβο N_k και τα φύλλα του φτάνουν μέχρι το επίπεδο l_{max}.

1. Στο επίπεδο $k+1$ θα είχε το πολύ $|F|$ απογόνους.

2. Στο επίπεδο $k+2$ θα είχε το πολύ $|F|^2$ απογόνους.

3. Στο επίπεδο $k+3$ θα είχε το πολύ $|F|^3$ απογόνους.

4. Με τη ίδια λογική στο επίπεδο l_{max} θα έχει το πολύ $|F|^{l_{max}-l_k}$ απογόνους.

Άρα το σύνολο των κόμβων ενός υποδένδρου με ρίζα τον τερματικό κόμβο k μια κωδικής λέξης $C(x_k)$ είναι το πολύ $S_k = \sum_{i=1}^{l_{max}-l_k} |F|^i = |D_k|$ με $D_k = \{f \in T_{N_k}^{l_{max}-l_k}\}$, όπου $T_{N_k}^{l_{max}-l_k}$ το υποδένδρο που έχει ως ρίζα τον N_k και τα φύλλα του φτάνουν μέχρι το επίπεδο l_{max}. Έστω τώρα ότι υπάρχει ένα σύνολο $D_j = \{f \in T_{N_j}^{l_{max}-l_j}\}$ τέτοιο ώστε $D_k \bigcap D_j \neq \emptyset$. Τότε θα υπήρχε ένα τουλάχιστον στοιχείο $f \in D_k \bigcap D_j$, δηλαδή ο κόμβος f θα είχε ως πρόγονο και τον κόμβο N_k και τον N_j από το οποίο έπεται ότι και οι τρεις κόμβοι θα άνηκαν στο ίδιο μονοπάτι. Χωρίς βλάβη της γενικότητάς υποθέτουμε ότι $k < i$, άρα ο N_i είναι απόγονος του N_k. Αυτό όμως σημαίνει ότι η λέξη $C(x_k)$ αποτελεί πρόθεμα της $C(x_i)$ το οποίο είναι άτοπο. Επομένως τα σύνολα των απογόνων κωδικών λέξεων είναι ξένα μεταξύ τους.

Όπως έχουμε ήδη δει, ο τερματικός κόμβος N_k, που βρίσκεται στο επίπεδο k του δένδρου, μιας κωδικής λέξης $C(x_k)$ θα έχει το πολύ $|F|^{l_{max}-l_k}$ απογόνους στο επίπεδο l_{max}, όπου l_{max} το μήκος της μεγαλύτερης κωδικής λέξης. Από την πρόταση 3.1 ξέρουμε ότι τα σύνολα των απογόνων είναι ξένα μεταξύ τους οπότε το άθροισμα τους πάνω σε όλες τις κωδικές λέξεις θα πρέπει να μας δίνει το μέγιστο πλήθος κόμβων που συναντάμε στο τελευταίο επίπεδο του δένδρου. Όπως έχουμε ήδη δείξει το δένδρο σε ένα τυχαίο επίπεδο k έχει μέχρι $|F|^k$ κόμβους, άρα στο επίπεδο l_{max} θα έχει μέχρι $|F|^{l_{max}}$ φύλλα. Οπότε:

$$\sum_{x_i \in \mathcal{X}} |F|^{l_{max}-l_i} \leq |F|^{l_{max}} \Rightarrow \sum_{x_i \in \mathcal{X}} |F|^{l_{max}} \cdot |F|^{-l_i} \leq |F|^{l_{max}} \Rightarrow \frac{1}{|F|^{l_{max}}} \sum_{x_i \in \mathcal{X}} |F|^{l_{max}} \cdot |F|^{-l_i} \leq 1 \Rightarrow$$
$$\sum_{x_i \in \mathcal{X}} |F|^{-l_i} \leq 1$$

Αντιστρόφως αν έχουμε μήκη κωδικών λέξεων που ικανοποιούν την ανισότητα του Kraft μπορούμε να φτιάξουμε ένα στιγμαίο κώδικα ως εξής:

1. Στο πρώτο βήμα ταξινομούμε κατά αύξουσα σειρά τα μήκη των κωδικών λέξεων $l_1, l_2, \cdots, l_{max}$.

2. Έπειτα κατασκευάζουμε ένα πλήρες $|F|$-αδικό δένδρο βάθους $l_{max}+1$ όπου στα κλαδιά του κάθε επιπέδου βρίσκονται τα σύμβολα του κωδικού αλφαβήτου F και ένα δένδρο T που περιέχει μόνο ένα κόμβο τη ρίζα του δένδρου.

3. Ξεκινώντας από το μήκος l_1 διαλέγουμε ένα μονοπάτι που ξεκινάει από τη ρίζα και φτάνει μέχρι κάποιο κόμβο του επιπέδου l_1. Στην συνέχεια διαγράφουμε του κόμβους του υποδένδρου $T_{N_k}^{l_{max}-l_k}$ και εισάγουμε το μονοπάτι στο δένδρο T ενώνοντας το πρώτο κόμβο του μονοπατιού με τη ρίζα, αφαιρώντας το από το πλήρες $|F|$-αδικό δένδρο ύψους l_{max}.

4. Για να βρούμε το μονοπάτι που αντιστοιχεί στο μήκος l_i μεταβαίνουμε στο επίπεδο l_i και διαλέγουμε κάποιον εσωτερικό κόμβου του δένδρου και διαγράφουμε τους κόμβους του υποδένδρου $T_{N_i}^{l_{max}-l_i}$. Κατόπιν εισάγουμε το μονοπάτι στο δένδρο T, αφαιρώντας το από το πλήρες $|F|$-αδικό δένδρο.

5. Η διαδικασία συνεχίζεται μέχρι να φτιάξουμε ένα μονοπάτι μήκους l_{max} και να το εισάγουμε στο δένδρο T.

Η ορθότητα της παραπάνω διαδικασίας στηρίζεται στην διάταξη των μηκών κατά αύξουσα σειρά. Έστω ότι θέλουμε να βρούμε ένα μονοπάτι για την λέξη $C(x_i)$ με μήκος l_i. Επειδή ήδη έχουμε φτιάξει τις κωδικές λέξεις που έχουν μήκος μικρότερο από τη $C(x_i)$ άρα στο δένδρο βρίσκονται σε επίπεδο μικρότερο του l_i, έπεται ότι οι κόμβοι των υποδένδρων $T_{N_j}^{l_{max}-l_j}$, $\forall j \leq i$ που έχουν διαγραφεί περιλαμβάνουν και τους κόμβους του επιπέδου

l_i. Άρα στο επίπεδο l_i είτε λείπουν κάποιοι κόμβοι που ανήκαν σε υπόδενδρα της μορφής $T_{N_j}^{l_{max}-l_j}$ με $j < i$ ή που έχουν μεταφερθεί στο δένδρο T, είτε έχουν μείνει εσωτερικοί κόμβοι που δεν ανήκουν σε κάποιο μονοπάτι κάποιας κωδικής λέξης και έτσι μπορούν να χρησιμοποιηθούν ελεύθερα εξασφαλίζοντας ότι η λέξη C_{x_i} δεν θα ανήκει στο ίδιο μονοπάτι με κάποια $C_{x_k} : k \leq i$, δηλαδή η C_{x_i} δεν θα έχει πρόθεμα κάποια άλλη κωδική λέξη.

Συνέχεια της απόδειξης για την ανισότητα McMillan

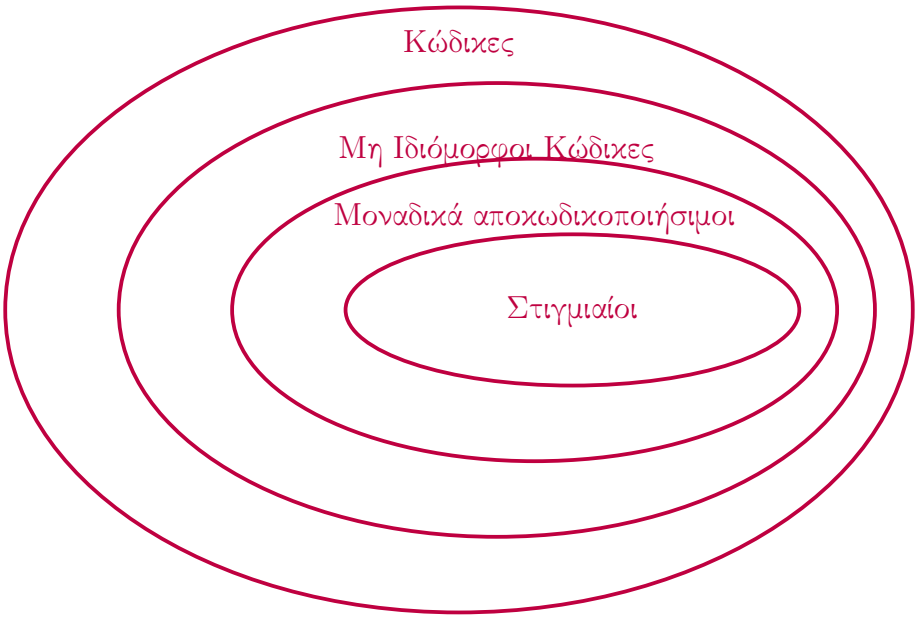
Η ολοκλήρωση της ανισότητας του Kraft, συμπληρώνει και την ημιτελή απόδειξη της ανισότητας του McMillan. Από το θεώρημα 3.1 ξέρουμε ότι ένας στιγμιαίος κώδικας είναι και μοναδικά αποκωδικοποιήσιμος, γεγονός που επιβεβαιώνεται και από τα παραπάνω κριτήρια καθώς τα μήκη των κωδικών λέξεων που επαληθεύουν την ανισότητα του Kraft επιβεβαιώνουν και την ανισότητα του McMillan. Βέβαια από τα παραδείγματα έχουμε δει ότι το αντίστροφο δεν ισχύει. Δηλαδή ένας κώδικας που είναι μοναδικά αποκωδικοποιήσιμος δεν είναι κατά ανάγκη στιγμιαίος. Η διαπίστωση αυτή επιβάλει την σχέση του υποσυνόλου ανάμεσα στα δύο σύνολα κωδίκων.

$$\{\text{Στιγμιαίοι Κώδικές}\} \subseteq \{\text{Μοναδικά αποκωδικοποήσιμοι Κώδικες}\}$$

Η παραπάνω ιδιότητα των δύο σύνολων μας επιτρέπει να ολοκληρώσουμε την απόδειξη της ανισότητας του McMillan. Το αντίστροφο της ανισότητας διαπίστωνε ότι "*αν τα μήκη ενός σύνολου κωδίκων λέξεων ικανοποιούν την παραπάνω ανισότητα, τότε μπορούμε να κατασκευάσουμε ένα μοναδικά αποκωδικοποιήσιμο κώδικα με βάση τα μήκη αυτά.*" Αφού λοιπόν τα μήκη ικανοποιούν την ανισότητα $\sum_{x_i \in \mathcal{X}} |F|^{-l_i} \leq 1$ που αποτελεί και την προϋπόθεση για την δημιουργία ενός στιγμιαίου κώδικα, μπορούμε να χρησιμοποιήσουμε τα παραπάνω μήκη ώστε να φτιάξουμε έναν στιγμιαίο κώδικα σύμφωνα με τη μέθοδο που αναπτύχθηκε κατά το θεώρημα 3.2. Επειδή όμως κάθε στιγμιαίο κώδικας είναι και μοναδικά αποκωδικοποιήσιμος, έπεται ότι θα έχουμε δημιουργήσει εν τέλει έναν μοναδικά αποκωδικοποιήσιμο κώδικα.

Ολοκληρώνοντας με βάση τους ορισμούς και τα θεωρήματα που έχουν παρουσιαστεί μέχρι στιγμής μπορούμε να διατάξουμε από συνολοθεωρητική άποψη τις διάφορες κατηγορίες κωδίκων. Το ευρύτερο σύνολο που περιέχει όλες τις προαναφερθείσες κατηγορίες είναι οι κώδικες που πληρούν τον ορισμό 3.1, δηλαδή είναι όλες οι απεικονίσεις $\{x \to C(x)$ με $x \in \mathcal{X}$ και $C(x) \in F^*\}$ μεταξύ των τιμών μιας τυχαίας μεταβλητής και των συμβολοσειρών που ανήκουν στην πεπερασμένη επέκταση ενός κωδικού αλφαβήτου F.[4]. Μέσα σε αυτό το σύνολο των γενικών κωδίκων ανήκουν οι μη ιδιόμορφοι κώδικες που απαιτούν σύμφωνα με τον ορισμό 3.3, η αντιστοιχία μεταξύ των τιμών της τυχαίας μεταβλητής και της πεπερασμένης επέκτασης του κωδικού αλφαβήτου F να είναι αμφιμονοσήμαντη. Μία ειδική κατηγορία των μη ιδιόμορφων κωδίκων είναι οι μοναδικά αποκωδικοποιήσιμοι κώδικες οι οποίοι προϋποθέτουν η αντιστοιχία μεταξύ της πεπερασμένης επέκτασης του συνόλου τιμών της τυχαίας μεταβλητής $\mathcal{X}^*$ και του κωδικού αλφαβήτου F^* να είναι αμφιμονοσήμαντη (ορισμός 3.4) και τέλος έπονται ως ειδική περίπτωση της προηγούμενης κατηγορίας οι μη προθεματικοί κώδικες. Το παρακάτω σχήμα περιγράφει πλήρως τις σχέσεις που περιγράψαμε.

[4]Με τον όρο πεπερασμένη επέκταση εννοούμε ένα σύνολο που περιέχει όλες τις συμβολοσειρές πεπερασμένου μήκους των οποίων όλα τα στοιχεία ανήκουν σε ένα συγκεκριμένο σύνολο $\mathcal{A}$, δηλαδή $\mathcal{A}^* = \{x_1 x_2 \cdots x_n : x_i \in \mathcal{A}\ \forall i = 1, \cdots, n\ \wedge n < \infty\}$



3.4 Βέλτιστο μήκος κώδικα

Στην προηγούμενη ενότητα αποδείξαμε ότι αν τα μήκη των κωδικών λέξεων ικανοποιούν την ανισοϊσότητα $\sum_{x_i \in \mathcal{X}} |F|^{=l_i} \leq 1$, τότε μπορούμε να φτιάξουμε έναν μοναδικά αποκωδικοποιήσιμο ή στιγμιαίο κώδικα με αυτά. Δεν απαντήσαμε όμως στα παρακάτω βασικά ερωτήματα:

1. Από ποιο θεωρητικό πλαίσιο προκύπτει το παραπάνω φράγμα για τα μήκη των κωδικών λέξεων;
2. Γιατί είναι ανισοϊσότητα και πότε επιτυγχάνεται η ισότητα;
3. Τα μήκη των κωδικών λέξεων που πληρούν την ανισότητα Kraft είναι τα βέλτιστα δυνατά;

Για χάριν ευκολίας θα υποθέσουμε για λίγο ότι το κωδικό μας αλφάβητο είναι το $\{0, 1\}$. Έστω ότι το σύνολο των δεδομένων προς συμπίεση μπορεί να μοντελοποιηθεί μέσω μίας τυχαίας μεταβλητής X που παίρνει n διακριτές τιμές με σ.μ.π $P_X(x) = \{Pr[X = x_1], \cdots, Pr[X = x_n]$. Τότε η ανισοϊσότητα $\sum_{x_i \in \mathcal{X}} |F|^{-l_i} \leq 1$ γίνεται:

$$\frac{1}{2^{l_1}} + \frac{1}{2^{l_2}} + \cdots + \frac{1}{2^{l_n}} \leq 1$$

Αν πάρουμε την περίπτωση της ισότητα τότε θα έχουμε:

$$\frac{1}{2^{l_1}} + \frac{1}{2^{l_2}} + \cdots + \frac{1}{2^{l_n}} = 1$$

Παρατηρούμε ότι κάθε όρος του αθροίσματος είναι θετικός και μικρότερος της μονάδας καθώς και ότι το άθροισμα τους μας κάνει ένα. Ακόμη όσο μικρότερο είναι ένα μήκος l_i τόσο μεγαλύτερο είναι το κλάσμα $\frac{1}{2^{l_i}}$. Με βάση τις παραπάνω διαπιστώσεις συνειδητοποιούμε ότι οι όροι που συμμετέχουν στο άθροισμα όταν επιτυγχάνεται η ισότητα μπορούν να απολέσουν μία συνάρτηση μάζας πιθανότητας που περιγράφει με λογικό τρόπο τα δεδομένα μας, αφού οι τιμές της μεταβλητής που κωδικοποιούνται με σύντομες κωδικές λέξεις θα αντιστοιχίζονται σε μεγάλες πιθανότητες της μορφής $\frac{1}{2^{l_i}}$.

Θα προσπαθήσουμε να αναπαραστήσουμε γραφικά το παραπάνω πρόβλημα για να καταλάβουμε πότε ελαχιστοποιείται το μέσο μήκος κώδικα δεδομένου ότι ικανοποιείται η ανισότητα Kraft. Για να είναι δυνατή η γραφική απεικόνιση του προβλήματος θα μελετήσουμε την περίπτωση που η X είναι μία δίτιμη τυχαία μεταβλητή. Προκειμένου να αναπαραστήσουμε το πρόβλημα θα χρησιμοποιήσουμε τρεις διακριτές διμετάβλητες απεικονίσεις.

1. Για το μέσο μήκος θα χρησιμοποιήσουμε την απεικόνιση:

$$f(l_1,l_2) = Pr[X=x_1]\cdot l_1 + Pr[X=x_2]\cdot l_2 \overset{Pr[X=x_1]+Pr[X=x_2]=1}{=} Pr[X=x_1]\cdot l_1 + (1-Pr[X=x_1])\cdot l_2$$
$$\Rightarrow f(l_1,l_2) = Pr[X=x_1](l_1-l_2)+l_2 \tag{3.9}$$

2. Ενώ για την ανισοϊσότητα του Kraft θα χρησιμοποιήσουμε τις παρακάτω δύο απεικονίσεις:

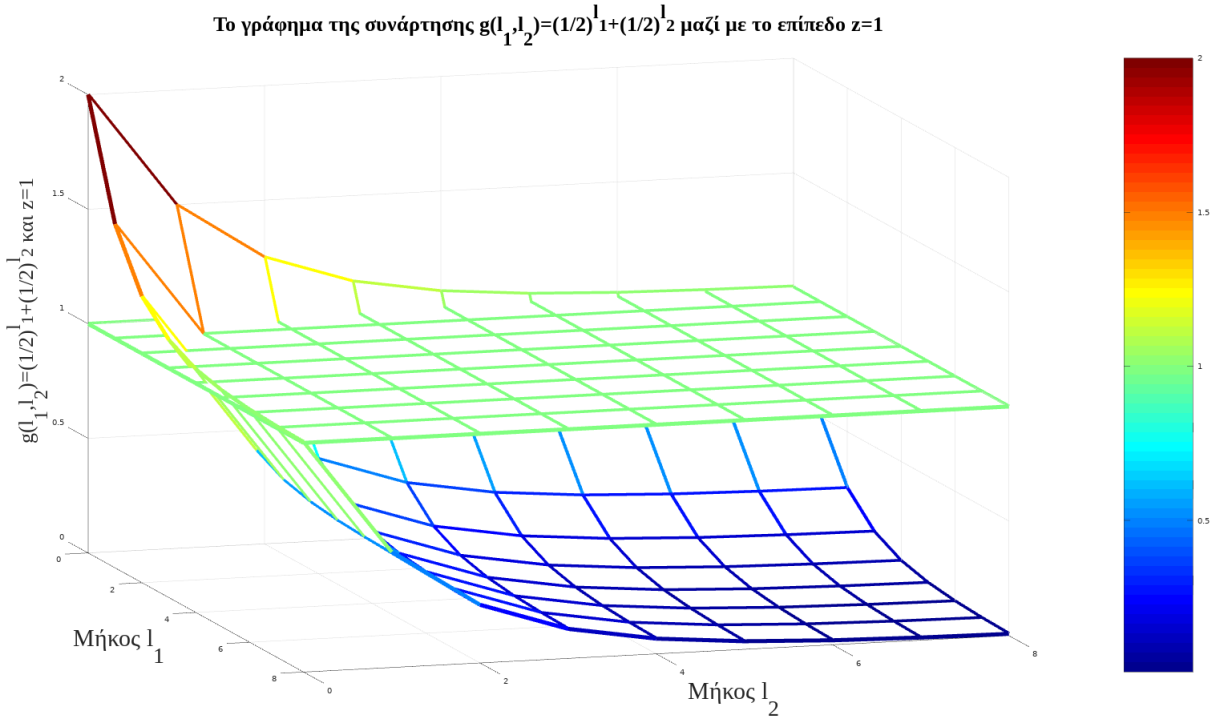
(α′)

$$g(l_1,l_2) = \left(\frac{1}{2}\right)^{l_1} + \left(\frac{1}{2}\right)^{l_2} \tag{3.10}$$

(β′)

$$z = 1 \tag{3.11}$$

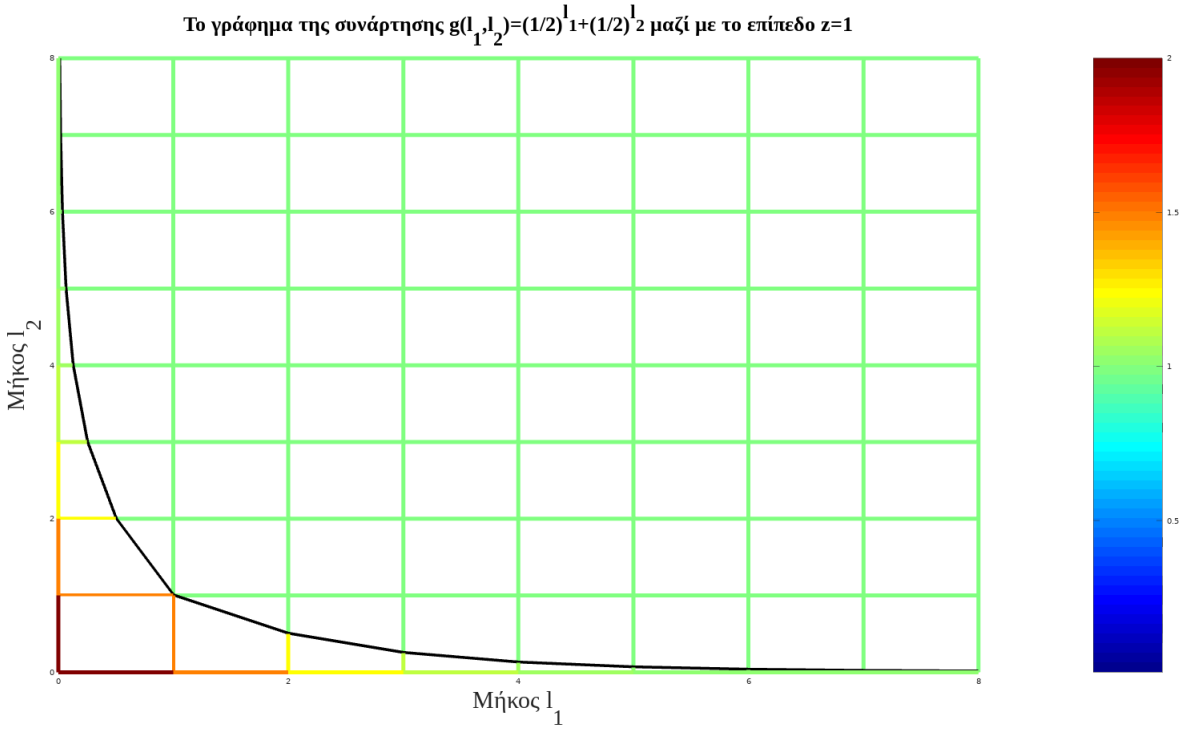
Με βάση τα παραπάνω ο περιορισμός μεταφράζεται γραφικά σαν το σύνολο των σημείων της συνάρτησης $g(l_1,l_2)$ που βρίσκονται κάτω από τη συνάρτηση $z=1$ ή έχουν κοινά σημεία με αυτή. Η γραφική αναπαράσταση των απεικονίσεων (3.10) και (3.11) θα αποτελείται από δύο τρισδιάστατα πλέγματα. Θεωρητικά τα μήκη l_1, l_2 μπορούν να πάρουν τιμές στον σύνολο των φυσικών αριθμών $\mathbb{N}$. Πρακτικά για χάρη ευκολίας θα θεωρήσουμε ότι τα μήκη l_1, l_2 παίρνουν τιμές στο διάστημα $[0,8]$ με τη λογική ότι 1 byte$= = 8$bits, έτσι ώστε να πλησιάσουμε ρεαλιστικά σενάρια που μπορούμε να συναντήσουμε στον ψηφιακό κόσμο. Κάθε σημείο της μορφής (l_1,l_2) θα αναπαριστάται ως το σημείο τομής δύο ευθειών που ανήκουν στο πλέγμα.



Σχήμα 3.20: Το πλέγμα της διακριτής συνάρτησης $g(l_1,l_2)$ που τέμνεται από το επίπεδο $z=1$.

Κώδικας στο Octave για τις γραφικές παραστάσεις $g(l_1, l_2), z = 1$

```
clear all;
clf;
pan on;
rotate3d on;
x=linspace(0,8,9);
y=linspace(0,8,9);
[xx, yy]=meshgrid(x,y);
z_1=(1/2).^ xx+(1/2).^ yy;
mesh(xx,yy,z_1,"LineWidth",5)
hold on;
z_2=1-0.*xx+0.*yy;
mesh(xx,yy,z_2,"LineWidth",5)
xlabel("Μήκος l_1","fontsize",30,"fontname","Times New Roman")
ylabel("Μήκος l_2","fontsize",30,"fontname","Times New Roman")
zlabel("g(l_1,l_2)=(1/2)^ {l_1}+(1/2) ^ {l_2} και z=1","fontsize",26,"fontname","Times New Roman")
title(" Το γράφημα της συνάρτησης g(l_1,l_2)=(1/2)^ {l_1}+(1/2)^ {l_2} μαζί με το επίπεδο z=1",
"fontsize",26,"fontname","Times New Roman")
colormap(jet)
colorbar
```



Σχήμα 3.21: Η κάτοψη της τομής των συναρτήσεων $g(l_1, l_2)$ που και $z = 1$. Η καμπύλη της τομής απεικονίζεται με μαύρο χρώμα

Από την κάτοψη του γραφήματος της τομής των συναρτήσεων $g(l_1, l_2)$ και $z = 1$ (σχήμα 3.21) προκύπτουν δύο συμπεράσματα. Πρώτον το μόνο σημείο που τέμνει η καμπύλη της τομής το πλέγμα της συνάρτησης g είναι το σημείο $(1, 1)$ και δεύτερον τα σημεία της g που βρίσκονται κάτω από τη $z = 1$ είναι τα σημεία του πλέγματος δεξιά της καμπύλης, τα οποία αναπαριστούν ζεύγη μηκών της μορφής (l_1, l_2) με l_1 και $l_2 \geq 1$. Από

τα παραπάνω συμπεράσματα έπεται ότι το $min\{(l_1, l_2) : g(l_1, l_2) \leq 1\} = (1, 1)$, δηλαδή οι κωδικές λέξεις με το μικρότερο μήκος προκύπτουν όταν ικανοποιείται η ισότητα στην ανισότητα Kraft

Αν ξεχάσουμε για λίγο ότι η συνάρτηση $g(l_1, l_2)$ ορίζεται στο $\mathbb{N} \times \mathbb{N}$ και υποθέσουμε ότι είναι μία συνεχής συνάρτηση με πεδίο ορισμό το $[0, +\infty) \times [0, +\infty)$. Τότε μπορούμε να πάρουμε τις μερικές παραγώγους της συνάρτηση ως προς l_1, l_2

1.
$$\frac{\partial g(l_1, l_2)}{\partial l_1} = \frac{\partial\left(\left(\frac{1}{2}\right)^{l_1} + \left(\frac{1}{2}\right)^{l_2}\right)}{\partial l_1} = ln\left(\frac{1}{2}\right) \cdot \left(\frac{1}{2}\right)^{l_1} < 0 \tag{3.12}$$

2.
$$\frac{\partial g(l_1, l_2)}{\partial l_2} = \frac{\partial\left(\left(\frac{1}{2}\right)^{l_1} + \left(\frac{1}{2}\right)^{l_2}\right)}{\partial l_2} = ln\left(\frac{1}{2}\right) \cdot \left(\frac{1}{2}\right)^{l_2} < 0 \tag{3.13}$$

Από τη σχέση 3.12 βλέπουμε ότι αν θεωρήσουμε το l_2 σταθερό οποιαδήποτε μεταβολή προς τη κατεύθυνση του $l_1 \to l_1 + dl_1$ έχει ως αποτέλεσμα τη μείωση της τιμής $g(l_1, l_2)$. Αντίστοιχα η σχέση 3.13 μας λέει ότι η μεταβολή $l_2 \to l_2 + dl_2$ προκαλεί μείωση της $g(l_1, l_2)$. Άρα συμπερασματικά μπορούμε να καταλήξουμε πως όταν αυξάνεται είτε το l_1 είτε το l_2 η συνάρτηση $g(l_1, l_2)$ θα μειώνεται. Άρα $\forall(l_1, l_2)$ και $(l_1{}', l_2{}')$ με $l_1{}' \geq l_1$ και $l_2{}' \geq l_2 \Rightarrow g(l_1{}', l_2{}') \leq g(l_1, l_2)$

$\forall(l_1, l_2) \in [0, +\infty) \times [0, \infty)$ με l_1 και $l_2 \geq 1 \Rightarrow \left(\frac{1}{2}\right)^{l_1} \leq \left(\frac{1}{2}\right)^{1} \wedge \left(\frac{1}{2}\right)^{l_2} \leq \left(\frac{1}{2}\right)^{1} \Rightarrow \left(\frac{1}{2}\right)^{l_1} \leq \frac{1}{2} \wedge \left(\frac{1}{2}\right)^{l_2} \leq \frac{1}{2} \Rightarrow$
$\left(\frac{1}{2}\right)^{l_1} + \left(\frac{1}{2}\right)^{l_2} \leq \frac{1}{2} + \frac{1}{2} \Rightarrow \left(\frac{1}{2}\right)^{l_1} + \left(\frac{1}{2}\right)^{l_2} \leq 1 \Rightarrow \left(\frac{1}{2}\right)^{l_1} + \left(\frac{1}{2}\right)^{l_2} - 1 \leq 0 \Rightarrow g(l_1, l_2) \leq g(1, 1).$

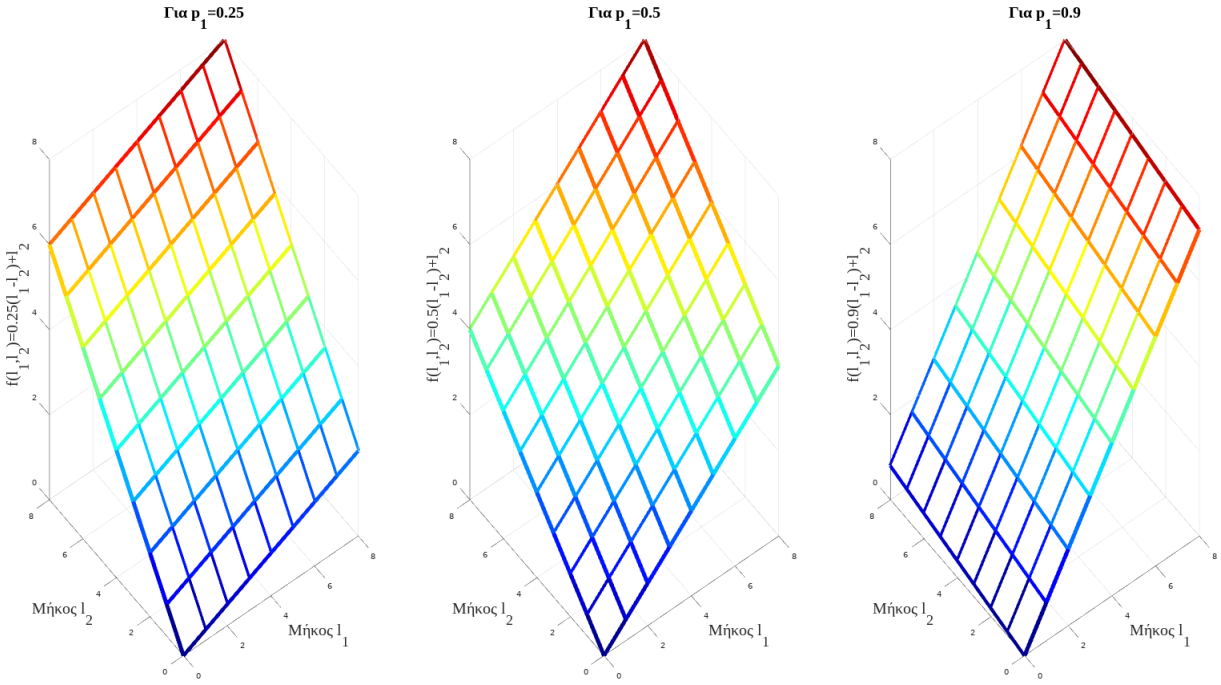
Άρα η συνάρτηση $g(l_1, l_2)$ γίνεται ένα για μία δίτιμη τυχαία μεταβλητή όταν και τα δύο μήκη πάρουν την τιμή 1 ενώ παίρνει αρνητικές τιμές μικρότερες της μονάδας όταν l_1 και $l_2 > 1$. Τι σημαίνει όμως για μία πρακτική εφαρμογή το παραπάνω αποτέλεσμα; Ας σκεφτούμε προς στιγμήν ότι έχουμε να κωδικοποιήσουμε τη δίτιμη τυχαία μεταβλητή X=$\{a, b\}$. Τότε αρκεί ένα μόνο *bit* για να κωδικοποιήσουμε την κάθε τιμή, λόγου χάρη $a \to 0$ και $b \to 1$. Αξίζει ακόμη να επισημανθεί ότι η συνάρτηση g δεν λαμβάνει καθόλου υπόψιν της τη σ.μ.π της τυχαίας μεταβλητή X. Απλά το μόνο που μας λέει είναι ότι όταν έχουμε μία δίτιμη τυχαία μεταβλητή, το βέλτιστο είναι να αναθέσουμε κωδικές λέξεις μήκους 1 σε κάθε τιμή. Οπότε ο περιορισμός $\sum_{x_i \in \mathcal{X}} |F|^{-l_1}$ γίνεται 1 όταν έχουν χρησιμοποιηθεί για τις τιμές της μεταβλητή X οι συντομότερες κωδικές λέξεις που είναι δυνατόν να υπάρξουν. Αλλιώς όταν χρησιμοποιούνται υποβέλτιστα μήκη κωδικών λέξεων τότε η ανισότητα του Kraft είναι μικρότερη της μονάδας. Με απλά λόγια η ανισοϊσότητα του Kraft μας δείχνει αν κινούμαστε στη σωστή κατεύθυνση κωδικοποίησης με βάση τα μήκη που έχουμε επιλέξει. Όσο πιο κοντά στο 1 είναι το αποτέλεσμα του αθροίσματος $\sum_{x_i \in \mathcal{X}} |F|^{-l_1}$ τόσο πιο κοντά στο βέλτιστο σύνολο μηκών είναι η επιλογή μας.

Ο δικός μας σκοπός βέβαια ήταν να ελέγξουμε αν το μέσο μήκος κώδικα που θα προκύψει αν χρησιμοποιήσουμε τα μήκη που προτείνει η ανισότητα Kraft είναι και το ελάχιστο. Στο μέσο μήκος κώδικα συμμετέχουν οι πιθανότητες και τα μήκη. Επειδή όμως ενδιαφερόμαστε για να βρούμε τα βέλτιστα μήκη των κωδικών λέξεων, μπορούμε να θεωρήσουμε τις πιθανότητες $Pr[X = x_1], Pr[X = x_2]$ σαν παραμέτρους που παίρνουν τιμές στο $[0, 1]$ και το μέσο μήκος σαν συνάρτηση των l_1, l_2 όπως φαίνεται και από την συνάρτηση 3.9, $f(l_1, l_2) = Pr[X = x_1] \cdot (l_1 - l_2) + l_2$.

Από τη μορφή της $f(l_1, l_2)$ συμπεραίνουμε ότι το μέσο μήκος μίας δίτιμης τυχαίας μεταβλητής μπορεί να αναπαρασταθεί με μία επίπεδη επιφάνεια (πλέγμα) στο τρισδιάστατο χώρο. Για να απεικονίσουμε γραφικά το παραπάνω επίπεδο και να υπάρχει μία συνοχή με τη γραφική παράσταση της $g(l_1, l_2)$ υποθέτουμε πάλι ότι τα μήκη l_1 και l_2 παίρνουν τιμές στο διάστημα $[0, 8] \subset \mathbb{N}$. Αν υποθέσουμε πάλι ότι η f παίρνει τιμές στο

$[0, +\infty) \times [0, \infty) \subset \mathbb{R}^2$ είναι εύκολο να δούμε ότι η συνάρτηση αναπαριστά ένα επίπεδο του $\mathbb{R}^2$ του οποίου η κλίση εξαρτάται από την πιθανότητα $Pr[X = x_1]$, καθώς:

$$\nabla f(l_1, l_2) = \begin{bmatrix} \dfrac{\partial f(l_1, l_2)}{\partial l_1} \\ \dfrac{\partial f(l_1, l_2)}{\partial l_2} \end{bmatrix} = \begin{bmatrix} \dfrac{\partial (Pr[X = x_1] \cdot (l_1 - l_2) + l_2)}{\partial l_1} \\ \dfrac{\partial (Pr[X = x_1] \cdot (l_1 - l_2) + l_2)}{\partial l_2} \end{bmatrix} = \begin{bmatrix} Pr[X = x_1] \\ 1 - Pr[X = x_1] \end{bmatrix} = \begin{bmatrix} Pr[X = x_1] \\ Pr[X = x_2] \end{bmatrix}$$

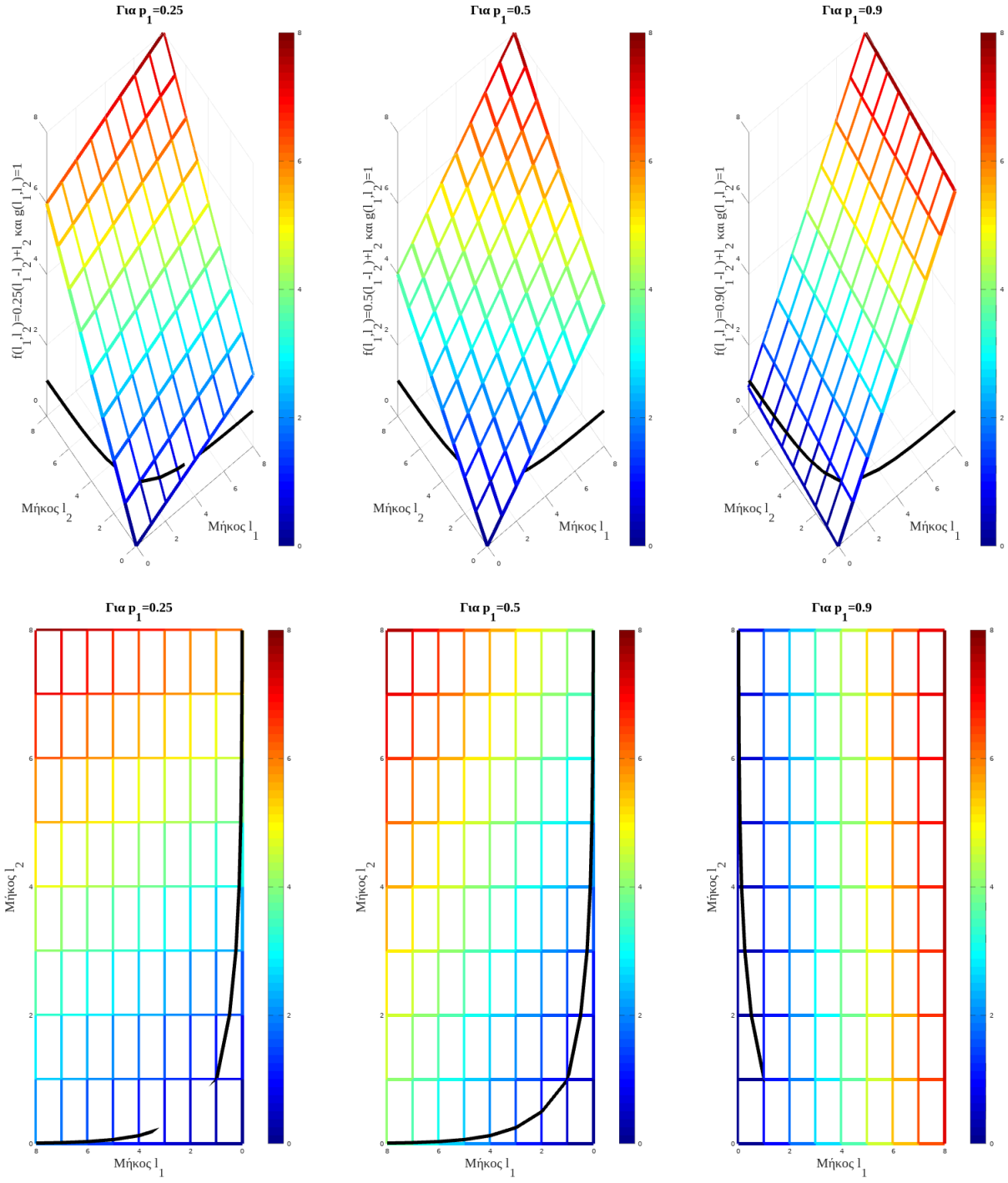


Σχήμα 3.22: Το γράφημα της συνάρτησης $f(l_1, l_2)$ για διαφορετικές τιμές του $Pr[X = x_1]$.

Κώδικας στο Octave για την γραφική παράσταση της $f(l_1, l_2)$ για $Pr[X = x_1] = p_1 = 0.25, 0.5, 0.9$

```
clf;
clear all;
pan on;
rotate3d on;
x=linspace(0,8,9);
y=linspace(0,8,9);
[xx, yy]=meshgrid(x,y);
subplot(1,3,1)
z_1=0.25.*(xx-yy)+yy;
mesh(xx,yy,z_1,"LineWidth",5)
xlabel("Μήκος l_1","fontsize",20,"fontname","Times New Roman")
ylabel("Μήκος l_2","fontsize",20,"fontname","Times New Roman")
zlabel("f(l_1,l_2)=0.25(l_1-l_2)+l_2","fontsize",20,"fontname","Times New Roman")
title("Για p_1=0.25","fontsize",20,"fontname","Times New Roman")
colormap(jet)
subplot(1,3,2)
z_2=0.5.*(xx-yy)+yy;
mesh(xx,yy,z_2,"LineWidth",5)
xlabel("Μήκος l_1","fontsize",20,"fontname","Times New Roman")
ylabel("Μήκος l_2","fontsize",20,"fontname","Times New Roman")
zlabel("f(l_1,l_2)=0.5(l_1-l_2)+l_2","fontsize",20,"fontname","Times New Roman")
title("Για p_1=0.5","fontsize",20,"fontname","Times New Roman")
colormap(jet)
subplot(1,3,3)
z_3=0.9.*(xx-yy)+yy;
mesh(xx,yy,z_3,"LineWidth",5)
xlabel("Μήκος l_1","fontsize",20,"fontname","Times New Roman")
ylabel("Μήκος l_2","fontsize",20,"fontname","Times New Roman")
zlabel("f(l_1,l_2)=0.9(l_1-l_2)+l_2","fontsize",20,"fontname","Times New Roman")
title("Για p_1=0.9","fontsize",20,"fontname","Times New Roman")
colormap(jet)
```

Αφού απεικονίσαμε και την συνάρτηση μέσου μήκους, το τελευταίο πράγμα που μένει είναι να δούμε πότε ελαχιστοποιείται η παραπάνω συνάρτηση δεδομένου ότι ικανοποιείται η ανισότητα του Kraft. Όπως αποδείξαμε προηγουμένως, η λύση της εξίσωσης $g(l_1, l_2) = 1$ είναι το σημείο $(1, 1)$ ή η καμπύλη C αν μιλάμε για συνεχές πεδίο ορισμού, που προκύπτει από την τομή των συναρτήσεων g και z και αποτελεί το βέλτιστο μήκος κωδικών λέξεων που μπορούμε να έχουμε για μία δίτιμη τυχαία μεταβλητή.



Σχήμα 3.23: Η τομή της καμπύλης C με την $f(l_1, l_2)$ για διάφορες τιμές του $Pr[X = x_1]$ και η κάτοψη της

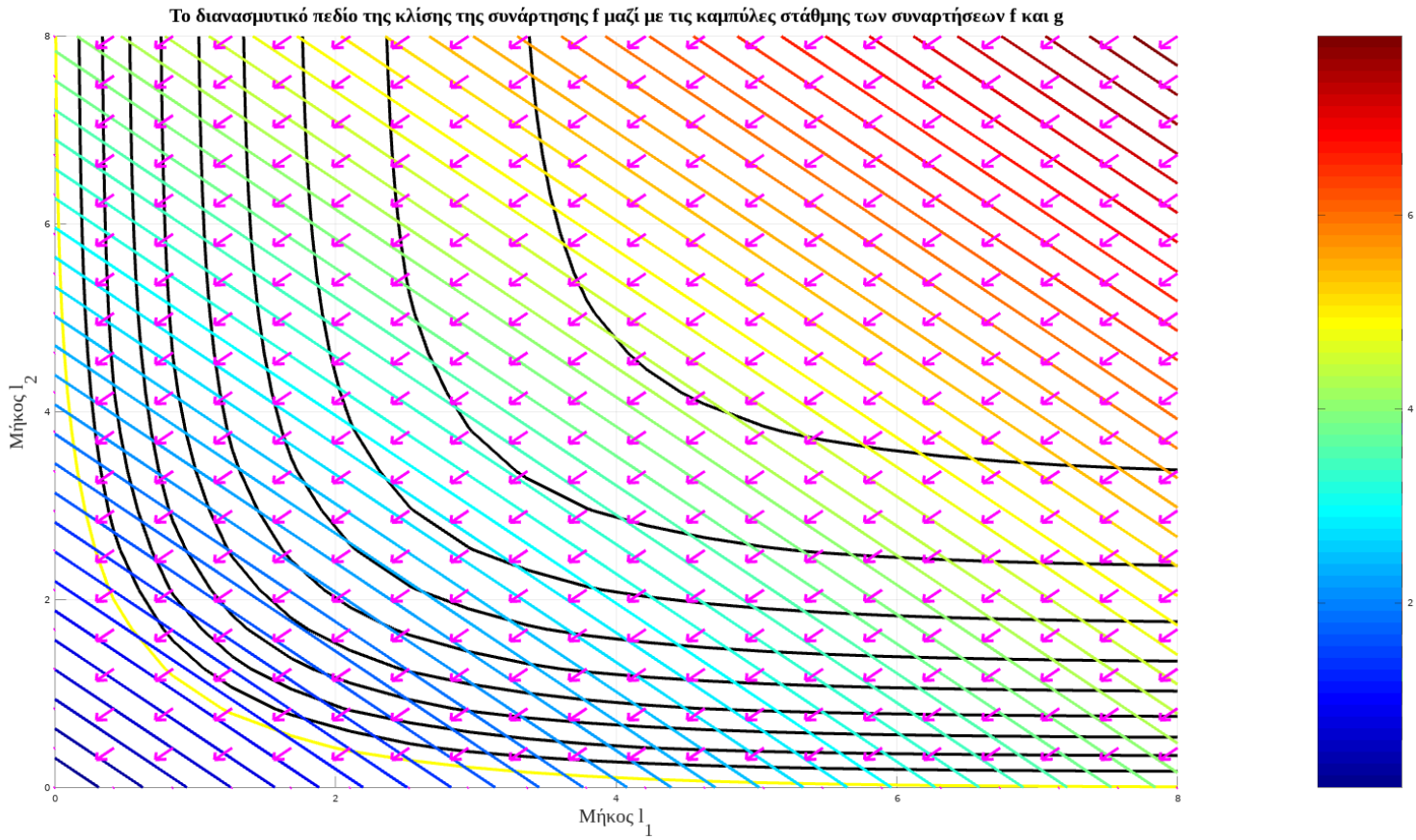
Κώδικας στο Octave για την γραφική παράσταση της τομής της $f(l_1, l_2)$ με την καμπύλη $C : \{g(l_1, l_2), z = 1\}$ για $Pr[X = x_1] = p_1 = 0.25, 0.5, 0.9$

```
clf;
clear all;
pan on;
rotate3d on;
x=linspace(0,8,9);
y=linspace(0,8,9);
[xx, yy]=meshgrid(x,y);
z= (1/2).^ xx+(1/2).^ yy;
subplot(1,3,1)
z_1=0.25.*(xx-yy)+yy;
mesh(xx,yy,z_1,"LineWidth",5)
xlabel("Μήκος l_1","fontsize",20,"fontname","Times New Roman")
ylabel("Μήκος l_2","fontsize",20,"fontname","Times New Roman")
zlabel("f(l_1,l_2)=0.25(l_1-l_2)+l_2 και g(l_1,l_2)=1","fontsize",20,"fontname","Times New Roman")
title("Για p_1=0.25","fontsize",20,"fontname","Times New Roman")
hold on;
contour3(xx,yy,z,[1, 1],"LineWidth",5,"k")
colormap(jet)
colorbar
subplot(1,3,2) z_2=0.5.*(xx-yy)+yy; mesh(xx,yy,z_2,"LineWidth",5) xlabel("Μήκος
l_1","fontsize",20,"fontname","Times New Roman")
ylabel("Μήκος l_2","fontsize",20,"fontname","Times New Roman")
zlabel("f(l_1,l_2)=0.5(l_1-l_2)+l_2 και g(l_1,l_2)=1","fontsize",20,"fontname","Times New Roman")
title("Για p_1=0.5","fontsize",20,"fontname","Times New Roman")
hold on;
contour3(xx,yy,z,[1, 1],"LineWidth",5,"k")
colormap(jet)
colorbar
subplot(1,3,3)
z_3=0.9.*(xx-yy)+yy;
mesh(xx,yy,z_3,"LineWidth",5)
xlabel("Μήκος l_1","fontsize",20,"fontname","Times New Roman")
ylabel("Μήκος l_2","fontsize",20,"fontname","Times New Roman")
zlabel("f(l_1,l_2)=0.9(l_1-l_2)+l_2 και g(l_1,l_2)=1","fontsize",20,"fontname","Times New Roman")
title("Για p_1=0.9","fontsize",20,"fontname","Times New Roman")
hold on;
contour3(xx,yy,z,[1, 1],"LineWidth",5,"k")
colormap(jet)
colorbar
```

Στο σχήμα 3.23 φαίνεται η τομή της καμπύλης C (μαύρο χρώμα) με τη συνάρτηση $f(l_1, l_2)$. Συμβουλευόμενοι τη γραμμή χρωμάτων παρατηρούμε ότι οι σκούρες μπλε αποχρώσεις αντιστοιχούν σε μικρές τιμές της f ενώ όσο πηγαίνουμε σε πιο θερμές αποχρώσεις η τιμή του μέσου μήκους αυξάνει. Η καμπύλη C αποτελεί τον οδηγό μας ώστε να βαδίζουμε πάνω σε ζεύγη μηκών που ικανοποιούν την ανισότητα Kraft καθώς κάθε σημείο του πλέγματος που βρίσκεται πάνω στην καμπύλη ή κάτω από αυτή θα είναι ένα ζεύγος μηκών που την επαληθεύει. Το θέμα είναι ότι χρειαζόμαστε και μία κατεύθυνση προκειμένου να βαδίσουμε προς την ελάχιστη τιμή της f που βρίσκεται πάνω στην καμπύλη. Από την ανάλυση πολυμετάβλητων συναρτήσεων γνωρίζου-

με ότι την κατεύθυνση για την πιο σύντομη μετάβαση σε μία χαμηλότερη ή ψηλότερη τιμή της συνάρτησης μας τη δίνει η κλίση της (gradient ∇f). Για το λόγο αυτό θα θεωρήσουμε παροδικά την $f(l_1, l_2)$ σαν μία πραγματική συνάρτηση δύο μεταβλητών που ορίζεται στο $[0, +\infty]) \times [0, +\infty]) \subset \mathbb{R}^2$ ώστε να υπολογίσουμε την κλίση της σε διάφορα σημεία του πεδίου ορισμού της και να σχεδιάζουμε το διανυσματικό της πεδίο $(l_1, l_2) \in [0, +\infty]) \times [0, +\infty]) \subset \mathbb{R}^2 \rightarrow \nabla f(l_1, l_2) \subset \mathbb{R}^2$. Θυμίζουμε ότι:

$$\nabla f(l_1, l_2) = \begin{bmatrix} \dfrac{\partial f(l_1, l_2)}{\partial l_1} \\ \dfrac{\partial f(l_1, l_2)}{\partial l_2} \end{bmatrix} = \begin{bmatrix} Pr[X = x_1] \\ -Pr[X = x_1] + 1 \end{bmatrix} = \begin{bmatrix} Pr[X = x_1] \\ Pr[X = x_2] \end{bmatrix} \tag{3.14}$$



Σχήμα 3.24: Το γράφημα ισοϋψών καμπύλων της $f(l_1, l_2)$ για $Pr[X = x_1] = 0.5$ μαζί με το διανυσματικό της πεδίο και τις ισοϋψείς καμπύλες της $g(l_1, l_2)$ για $z = 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0$. Η κίτρινη καμπύλη αναπαριστά την στάθμη της g για $z = 1$

.

Κώδικας στο Octave για το γράφημα ισοϋψών καμπύλων της $f(l_1, l_2)$ για $Pr[X = x_1] = 0.5$ μαζί με το διανυσματικό της πεδίο και τις ισοϋψείς καμπύλες της $g(l_1, l_2)$ για $z = 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0$

```
clf;
clear all;
pan on;
rotate3d on;
x=linspace(0,8,20);
y=linspace(0,8,20);
[xx, yy]=meshgrid(x,y);
z= (1/2).^ xx+(1/2).^ yy;
z_1=0.5*(xx-yy)+yy;
z_2=11,2-0.*xx-0.*yy;
contour3(xx,yy,z,[0.1 : 0.1 : 1.0],"LineWidth",3,"k")
hold on;
contour3(xx,yy,z,[1, 1],"LineWidth",3,"y")
hold on;
u,v,w
= surfnorm(z_1);
quiver3(xx,yy,z_1,u,v,w,"m","LineWidth",3)
hold on;
contour3(xx,yy,z_1,50,"LineWidth",3)
colormap(jet)
colorbar
xlabel("Μήκος l_1","fontsize",20,"fontname","Times New Roman")
ylabel("Μήκος l_2","fontsize",20,"fontname","Times New Roman")
title("Το διανασμυτικό πεδίο της κλίσης της συνάρτησης f μαζί με τις καμπύλες στάθμης των συναρτήσεων f και g","fontsize",20,"fontname","Times New Roman")
```

Στην εικόνα 3.24 βλέπουμε το γράφημα ισοϋψών καμπυλών της $f(l_1, l_2)$ για $Pr[X = x_1] = 0.5$ μαζί με το διανυσματικό του πεδίο. Από το γράφημα μπορούμε να παρατηρήσουμε πρώτον πώς τα διανύσματα της κλίσης $\nabla f(l_1, l_2)$ τέμνουν κάθετα τις ισοϋψείς καμπύλες και δεύτερον μας δείχνουν της κατεύθυνση που πρέπει να ακολουθήσουμε ώστε να υπάρξει μεταβολή στη τιμής της f. Στο ίδιο γράφημα απεικονίζονται και οι ισοϋψείς καμπύλες της g για $g(l_1, l_2) \leq 1$. Οι καμπύλες που αντιστοιχούν σε μικρές τιμές τις g βρίσκονται στη θερμή περιοχή (υψηλές τιμές) του γραφήματος της f ενώ όσο προσεγγίζουμε την τιμή 1, οι καμπύλες πηγαίνουν προς την ψυχρή περιοχή (χαμηλές τιμές) του γραφήματος. Προφανώς την κατεύθυνση ώστε να μετακινηθούμε στις καμπύλες της g που προσεγγίζουν την τιμή 1, θα μας την δώσει η κλίση της g.

Γεωμετρικά συνειδητοποιούμε πως όταν φτάσουμε στο σημείο της C έστω το (l_1^*, l_2^*) που ελαχιστοποιεί την τιμή της f, τότε η ισοϋψής καμπύλη C με την αντίστοιχη της f που περιέχει το σημείο (l_1^*, l_2^*) θα εφάπτονται στο σημείο αυτό. Τότε όμως τα διανύσματα $\nabla f(l_1^*, l_2^*), \nabla g(l_1^*, l_2^*)$ θα είναι κάθετα και στις δύο καμπύλες και παράλληλα μεταξύ τους στο σημείο αυτό:

$$\nabla f(l_1^*, l_2^*) = \lambda \cdot \nabla g(l_1^*, l_2^*)$$

Ξέρουμε ότι:

$$\nabla g(l_1, l_2) = \begin{bmatrix} \frac{\partial g(l_1, l_2)}{\partial l_1} \\ \frac{\partial g(l_1, l_2)}{\partial l_2} \end{bmatrix} = \begin{bmatrix} \lambda \cdot \left(\frac{1}{2}\right)^{l_1} ln\left(\frac{1}{2}\right) \\ \lambda \cdot \left(\frac{1}{2}\right)^{l_2} ln\left(\frac{1}{2}\right) \end{bmatrix} \tag{3.15}$$

Εξισώνοντας την σχέση (3.15) έχουμε:

$$\left.\begin{aligned} Pr[X=x_1] &= \lambda\cdot\left(\frac{1}{2}\right)^{l_1} ln\left(\frac{1}{2}\right) \\ Pr[X=x_2] &= \lambda\cdot\left(\frac{1}{2}\right)^{l_2} ln\left(\frac{1}{2}\right) \end{aligned}\right\} \Rightarrow \left.\begin{aligned} \left(\frac{1}{2}\right)^{l_1} &= \frac{Pr[X=x_1]}{\lambda\cdot ln\left(\frac{1}{2}\right)} \\ \left(\frac{1}{2}\right)^{l_2} &= \frac{Pr[X=x_2]}{\lambda\cdot ln\left(\frac{1}{2}\right)} \end{aligned}\right\} \tag{3.16}$$

Από τον περιορισμό που επιβάλλει η ανισότητα Kraft για την περίπτωση της ισότητας γνωρίζουμε ότι:

$$\left(\frac{1}{2}\right)^{l_1} + \left(\frac{1}{2}\right)^{l_2} = 1 \tag{3.17}$$

Αντικαθιστώντας τα $\left(\frac{1}{2}\right)^{l_1}$, $\left(\frac{1}{2}\right)^{l_2}$ από τη σχέση (3.16) στην (3.17) βρίσκουμε το λ

$$\begin{aligned} &\left(\frac{1}{2}\right)^{l_1} + \left(\frac{1}{2}\right)^{l_2} = 1 \Rightarrow \frac{Pr[X=x_1]}{\lambda\cdot ln\left(\frac{1}{2}\right)} + \frac{Pr[X=x_2]}{\lambda\cdot ln\left(\frac{1}{2}\right)} = 1 \Rightarrow \frac{1}{\lambda\cdot ln\left(\frac{1}{2}\right)}\cdot(Pr[X=x_1]+Pr[X=x_2]) = 1 \\ &\overset{Pr[X=x_1]+Pr[X=x_2]=1}{\Rightarrow} \frac{1}{\lambda\cdot ln\left(\frac{1}{2}\right)} = 1 \Rightarrow \lambda = \frac{1}{ln\left(\frac{1}{2}\right)} \end{aligned} \tag{3.18}$$

Αντικαθιστώντας το λ στην (3.16) έχουμε:

$$\left.\begin{aligned} \left(\frac{1}{2}\right)^{l_1} &= \frac{Pr[X=x_1]}{\lambda\cdot ln\left(\frac{1}{2}\right)} \\ \left(\frac{1}{2}\right)^{l_2} &= \frac{Pr[X=x_2]}{\lambda\cdot ln\left(\frac{1}{2}\right)} \end{aligned}\right\} \Rightarrow \begin{aligned} \left(\frac{1}{2}\right)^{l_1} &= Pr[X=x_1] \\ \left(\frac{1}{2}\right)^{l_2} &= Pr[X=x_2] \end{aligned} \quad \begin{aligned} & l_1 = \log_2\left(\frac{1}{Pr[X=x_1]}\right) \\ \text{ή} \quad & l_2 = \log_2\left(\frac{1}{Pr[X=x_2]}\right) \end{aligned} \tag{3.19}$$

Το τελευταίο αποτέλεσμα (3.19) μπορεί να ερμηνευτεί με δύο τρόπους. Αν τα μήκη των κωδικών λέξεων είναι ίσα με την ιδιοπληροφορία των τιμών της τυχαίας μεταβλητής τότε το άθροισμα $\left(\frac{1}{2}\right)^{l_1} + \left(\frac{1}{2}\right)^{l_2}$ είναι ίσο με ένα. Αλλιώς αν οι πιθανότητες μπορούν να εκφραστούν σαν δυνάμεις με βάση το $\frac{1}{|F|}$, όπου F ο πληθάριθμος του κωδικού αλφαβήτου, τότε θα υπάρχει ένα σύνολο μηκών για το οποίο η ανισοϊσότητα του Kraft θα κάνει ένα. Από την προηγούμενη ανάλυση ξέρουμε ότι η ισότητα επιτυγχάνεται όταν έχουμε χρησιμοποιήσει τις πιο σύντομες κωδικές λέξεις για τις τιμές της τυχαίας μεταβλητής X, επομένως το μέσο μήκος κώδικά γίνεται βέλτιστο όταν ικανοποιείται η ισότητα στη σχέση $\sum_{x_i\in\mathcal{X}} |F|^{l_1} \leq 1$.

Για να λύσουμε το πρόβλημα βελτιστοποίησης μέσου μήκους για μία τυχαία μεταβλητή που παίρνει n διακριτές τιμές με σ.μ.π $P_X(x) = \{Pr[X=x]\}_{x\in\mathcal{X}}$ δεδομένου ότι ικανοποιείται η ανισότητα του Kraft θα χρησιμοποιήσουμε τους πολλαπλασιαστές Lagrange

Αν $f(l_1, l_2, \cdots, l_n) = \sum_{x_i\in\mathcal{X}} Pr[X=x_i]\cdot l_i$ δεδομένου ότι $\sum_{x_i\in\mathcal{X}} |F|^{-l_i} = 1$ με $g(l_1, l_2, \cdots, l_n) = \sum_{x_i\in\mathcal{X}} |F|^{-l_i}$, τότε σύμφωνα με τους πολλαπλασιαστές Lagrange θα ισχύει:

$$\nabla f(l_1, l_2, \cdots, l_n) = \lambda \cdot \nabla g(l_1, l_2, \cdots, l_n) \Rightarrow \frac{\partial f(l_1, l_2, \cdots, l_n)}{\partial l_i} = \lambda \cdot \frac{\partial g(l_1, l_2, \cdots, l_n)}{\partial l_i} \Rightarrow$$
$$\frac{\partial(\sum_{x_i \in \mathcal{X}} Pr[X = x_i] \cdot l_i)}{\partial l_i} = \lambda \cdot \frac{\partial \sum_{x_i \in \mathcal{X}} |F|^{-l_i}}{\partial l_i} \Rightarrow Pr[X = x_i] = \lambda \cdot |F|^{-l_i} ln(|F|) \Rightarrow |F|^{-l_i} = \frac{Pr[X = x_i]}{\lambda \cdot ln(|F|)} \tag{3.20}$$

Αντικαθιστώντας στον περιορισμό $\sum_{x_i \in \mathcal{X}} |F|^{-l_i} = 1$ τα $|F|^{=l_i}$ που βρήκαμε από τη σχέση 3.20 έχουμε :

$$\sum_{x_i \in \mathcal{X}} |F|^{-l_i} = 1 \Rightarrow \sum_{x_i \in \mathcal{X}} \frac{Pr[X = x_i]}{\lambda \cdot ln(|F|)} = 1 \Rightarrow \frac{1}{\lambda \cdot ln(|F|)} \sum_{x_i \in \mathcal{X}} Pr[X = x_i] = 1 \Rightarrow \frac{1}{\lambda \cdot ln(|F|)} = 1 \Rightarrow \lambda = \frac{1}{ln(|F|)} \tag{3.21}$$

Αντικαθιστώντας το λ που βρήκαμε από την 3.21 στην 3.20 έπεται:

$$|F|^{-l_i} = \frac{Pr[X = x_i]}{\frac{1}{ln(|F|)} \cdot ln(|F|)} \Rightarrow |F|^{-l_i} = \frac{Pr[X = x_i]}{\frac{1}{ln(|F|)} \cdot ln(|F|)} \Rightarrow Pr[X = x_i] = \frac{1}{|F|^{l_i}} \text{ ή } l_i = \log_{|F|} \frac{1}{Pr[X = x_i]} \tag{3.22}$$

Άρα και για την περίπτωση που η τυχαία μεταβλητή παίρνει τιμές σε ένα πεπερασμένο σύνολο, ο κώδικας που παράγεται για αυτήν είναι ο βέλτιστος δυνατός αν τα μήκη των κωδικών λέξεων που επιλέξουμε είναι ίσα με τη ιδιοπληροφορία των τιμών της, γεγονός ισοδύναμο με την περίπτωση να ισχύει η ισότητα στην ανισοϊσότητα Kraft καθώς:

$$\sum_{x_i \in \mathcal{X}} |F|^{-l_i} = \sum_{x_i \in \mathcal{X}} |F|^{-\log_{|F|} \frac{1}{Pr[X = x_i]}} = \sum_{x_i \in \mathcal{X}} |F|^{\log_{|F|} Pr[X = x_i]} = \sum_{x_i \in \mathcal{X}} Pr[X = x_i] = 1.$$

Στο σημείο αυτό φαίνεται ξεκάθαρα η σχέση που έχει η συμπίεση με την εντροπία. Αφού λοιπόν μόλις αποδείξαμε πως το καλύτερο μήκος που μπορεί να έχει η κωδική λέξη μίας τυχαίας μεταβλητής είναι η ιδιοπληροφορία της έπεται ότι το μέσο μήκος του βέλτιστο κώδικα θα είναι ο σταθμισμένος μέσο των βέλτιστων μηκών των τιμών της τυχαίας μεταβλητής. Ο σταθμισμένος μέσο όμως τον βέλτιστων μηκών δεν είναι παρά ο σταθμισμένος μέσος των ιδιοπληροφοριών των τιμών της τυχαίας μεταβλητής ή αλλιώς η εντροπία της. Από αυτή την παρατήρηση προκύπτει φυσιολογικά πως η εντροπία αποτελεί επί της ουσία το όριο της συμπίεσης μίας τυχαίας μεταβλητής. Η τελευταία διαπίστωση αποτελεί την μη φορμαλιστική διατύπωση του θεωρήματος κωδικοποίησης πηγής του Shannon. Στο κεφαλαίο αυτό θα δούμε μία απόδειξη που βασίζεται στην απόσταση Kullback-Leibler ενώ στο επόμενο θα αποδείξουμε πάλι το ίδιο θεώρημα κάνοντας χρήση του θεωρήματος ασυμπτωτικής ισοκατανομής για πηγές χωρίς μνήμη.

Θεώρημα 3.4. *Το μέσο μήκος $\overline{L}$ ενός στιγμιαίου κώδικα που ορίζεται σε ένα αλφάβητο F για μία τυχαία μεταβλητή X είναι μεγαλύτερο ή ίσο με την εντροπία της $H_F(X)$*

$$\overline{L} \geq H_F(X) \tag{3.23}$$

με την ισότητα να ισχύει μόνο για $Pr[X = x_i] = \frac{1}{|F|^{l_i}}$

Απόδειξη

$$\overline{L} - H_F(X) = \sum_{x\in\mathcal{X}} Pr[X=x]\cdot l_x - \sum_{x\in\mathcal{X}} Pr[X=x]\cdot \log_F \frac{1}{Pr[X=x]} =$$

$$-\sum_{x\in\mathcal{X}} Pr[X=x_i]\log_F F^{l_x} - \sum_{x\in\mathcal{X}} Pr[X=x]\cdot \log_F \frac{1}{Pr[X=x]} =$$

$$\sum_{x\in\mathcal{X}} Pr[X=x]\log_F \frac{Pr[X=x]}{|F|^{-l_x}} = \sum_{x\in\mathcal{X}} Pr[X=x]\log_F \frac{Pr[X=x]\cdot\sum_{i=1}^n |F|^{-l_x}}{|F|^{-l_x}\cdot\sum_{i=1}^n |F|^{-l_x}} \overset{r_x=\frac{|F|^{-l_x}}{\sum_{i=1}^n |F|^{-l_x}}}{=}$$

$$\sum_{x\in\mathcal{X}} Pr[X=x]\log_F \frac{Pr[X=x]}{r_x} + \log_F \frac{1}{\sum_{i=1}^n |F|^{-l_x}} = D(P_X(x)||R_X(X)) + \log_F \frac{1}{\sum_{i=1}^n |F|^{-l_x}}$$

όπου $P_X(x)$, $R_X(x)$ *δύο διαφορετικές σ.μ.π που περιγράφουν την* X *με* $P_X(x) = \{Pr[X=x]\}_{x\in\mathcal{X}}$ *και* $R_X(x) = \{\frac{|F|^{-l_x}}{\sum_{i=1}^n |F|^{-l_x}}\}_{x\in\mathcal{X}}$. *Όμως το* $D(P||R) \geq 0$ *από τις ιδιότητες της σχετικής εντροπίας. Επίσης από την ανισότητα του Kraft γνωρίζουμε πως* $\sum_{i=1}^n |F|^{-l_x} \leq 1 \Rightarrow \frac{1}{\sum_{i=1}^n |F|^{-l_x}} \geq 1 \Rightarrow \log_F \frac{1}{\sum_{i=1}^n |F|^{-l_x}} \geq 0$. *Άρα* $\overline{L} - H_F(X) \geq 0 \Rightarrow \overline{L} \geq H_F(X)$

Προφανώς έχουμε υποπτευθεί ότι δεν είναι δυνατό η ιδιοπληροφορία $\log(\frac{1}{Pr[X=x_i]})$ να είναι πάντα φυσικός αριθμός και αντίστροφα οι πιθανότητες $Pr[X=x_i]$ δεν μπορούν να εκφραστούν πάντοτε σαν κάποια δύναμη $(\frac{1}{|F|})^{l_i}$.

Όμως η ανισοϊσότητα του Kraft υποθέτει ότι τα μήκη $l_1, l_2, \cdots, l_n$ είναι φυσικοί αριθμοί. Οδηγούμαστε λοιπόν στο συμπέρασμα ότι τα μήκη l_i θα ισούνται στην πραγματικότητα με την ποσότητα $\lceil \frac{1}{Pr[X=x_i]} \rceil$, δηλαδή θα καταλήξουμε να χρησιμοποιούμε λίγο μεγαλύτερα μήκη από τα βέλτιστα $l_i{}^* = \log_{|F|}(\frac{1}{Pr[X=x_i]})$, γεγονός που προσδίδει ένα πλεόνασμα πληροφορίας στο τελικό κώδικα. Για το λόγο αυτό στη ανισοϊσότητα Kraft υπάρχει και επιλογή της καθαρής ανισότητας για την περίπτωση που η φύση των αριθμών δεν μας επιτρέπει να επιτύχουμε τα βέλτιστα μήκη.

Το επόμενο ερώτημα που αξίζει να θέσουμε στον εαυτό μας είναι κατά πόσο υπερβαίνει αυτό το πλεόνασμα πληροφορίας το μέσο μήκος του βέλτιστου κώδικα αλλά και τι μπορούμε να κάνουμε για να βελτιώσουμε την ανωτέρα κατάσταση. Για το ακέραιο μέρος ενός πραγματικού αριθμού ξέρουμε ότι ισχύει $x \leq \lceil x \rceil < x+1$. Άρα για τα μήκη l_i έπεται ότι:

$$l_i^* = \log_{|F|}(\frac{1}{Pr[X=x_i]}) \leq l_i = \lceil \log_{|F|}(\frac{1}{Pr[X=x_i]}) \rceil < \log_{|F|}(\frac{1}{Pr[X=x_i]}) + 1 \ \forall l_i$$

Επειδή τα $Pr[X=x_i] > 0 \ \forall i$ έπεται ότι :

$$Pr[X=x_i] \cdot \log_{|F|}(\frac{1}{Pr[X=x_i]}) \leq Pr[X=x_i] \cdot \lceil \log_{|F|}(\frac{1}{Pr[X=x_i]}) \rceil <$$

$$Pr[X=x_i] \cdot \log_{|F|}(\frac{1}{Pr[X=x_i]}) + 1 \ \forall \ x_i \in \mathcal{X}$$

Αθροίζώντας πάνω στη πιθανότητες της σ.μ.π έχουμε: (3.24)

$$\sum_{x_i \in \mathcal{X}} Pr[X=x_i] \cdot \log_{|F|}(\frac{1}{Pr[X=x_i]}) \leq \sum_{x_i \in \mathcal{X}} Pr[X=x_i] \cdot \lceil \log_{|F|}(\frac{1}{Pr[X=x_i]}) \rceil <$$

$$\sum_{x_i \in \mathcal{X}} Pr[X=x_i] \cdot \log_{|F|}(\frac{1}{Pr[X=x_i]}) + \sum_{x_i \in \mathcal{X}} Pr[X=x_i] \Rightarrow$$

$$H_F(X) \leq \sum_{x_i \in \mathcal{X}} Pr[X=x_i] \cdot l_i < H_F(X) + 1 \Rightarrow H_F(X) \leq \overline{L} < H_F(X) + 1$$

Από τη δεύτερη ανισότητα της σχέση (3.23), $\overline{L} < H_F(X) + 1$ βλέπουμε ότι το μέσο μήκος ενός κώδικα που ικανοποιεί την ανισοϊσότητα Kraft από αυτό του βέλτιστου κώδικά διαφέρουν κατά 1. Το πιο σημαντικό όμως συμπέρασμα κρύβεται στην πρώτη ανισότητα που δηλώνει ξεκάθαρα ότι το μέσο μήκος ενός κώδικα για μια τυχαία μεταβλητή X φράσσεται από την εντροπία της. Η διαπίστωση αυτή έρχεται να επιβεβαιώσει τον υπαινιγμό μας στο πρώτο κεφάλαιο ότι η εντροπία αποτελεί την συντομότερη περιγραφεί μίας τυχαίας μεταβλητής.

Ήρθε η ώρα να απαντήσουμε αν μπορεί να βελτιωθεί το πλεόνασμα πληροφορίας που εμφανίζεται όταν τα μήκη l_i δεν συμπίπτουν με την ιδιοπληροφορία των τιμών της τυχαίας μεταβλητής X. Πολλές φορές είναι χρήσιμο να κωδικοποιούμε ακολουθίες συμβόλων αντί για το κάθε σύμβολο ξεχωριστά. Για παράδειγμα, αν θέλουμε να συμπιέσουμε ένα σύνολο αιτήσεων που γίνονται προς κάποιο εξυπηρετητή $HTTPS$ είναι αποδοτικό να προβούμε σε μία συμπίεση συμβολοσειρών καθώς ορισμένα μέρη των συγκεκριμένων αιτήσεων παραμένουν σταθερά και επαναλαμβάνονται σε κάθε αίτηση. Αν λοιπόν θεωρήσουμε τις ακολουθίες συμβόλων σαν ανεξάρτητες και ισόνομες τυχαίες μεταβλητές που παράγονται από κάποια πηγή X, τότε το μέσο αν σύμβολο θα δίνεται από τη σχέση:

$$\overline{L}^n = \frac{1}{n} \sum_{\mathbf{x_1^n} \in \mathcal{X}^n} Pr[\mathbf{X_1^n} = \mathbf{x_1^n}] \cdot l(\mathbf{x_1^n}) \tag{3.25}$$

Κατά αναλογία με τη σχέση (3.22) το βέλτιστο μήκος $l^*(\mathbf{X_1^n})$ για την συμβολοσειρά $\mathbf{x_1^n}$ θα είναι η ιδιοπληροφορία της $\log(\frac{1}{Pr[\mathbf{X_1^n} = \mathbf{x_1^n}]})$ ενώ το μήκος που ικανοποιεί την ανισότητα Kraft θα ισούται με $l(\mathbf{X_1^n}) = \lceil \log(\frac{1}{Pr[\mathbf{X_1^n} = \mathbf{x_1^n}]}) \rceil$. Τότε το μέσο μήκος $\overline{L}^n$ θα φράσσεται από την κοινού εντροπία $H_F(\mathbf{X_1^n})$:

$$H_F(\mathbf{X_1^n}) \leq \frac{1}{n} \sum_{\mathbf{x_1^n} \in \mathcal{X}^n} Pr[\mathbf{X_1^n} = \mathbf{x_1^n}] \cdot l(\mathbf{x_1^n}) < H_F(\mathbf{X_1^n}) + 1 \tag{3.26}$$

Αν η συμβολοσειρά $x_1 \cdots, x_n$ δημιουργήθηκε από ανεξάρτητες και ισόνομες τυχαίες μεταβλητές τότε θα ισχύει:

$$H_F(\mathbf{X_1^n}) \leq \frac{1}{n} \sum_{\mathbf{x_1^n} \in \mathcal{X}^n} Pr[\mathbf{X_1^n} = \mathbf{x_1^n}] \cdot l(\mathbf{x_1^n}) < H_F(\mathbf{X_1^n}) + 1 \Rightarrow$$

$$\sum_{i=1}^{n} H_F(X_i) \leq \frac{1}{n} \sum_{\mathbf{x_1^n} \in \mathcal{X}^n} Pr[\mathbf{X_1^n} = \mathbf{x_1^n}] \cdot l(\mathbf{x_1^n}) < \sum_{i=1}^{n} H_F(X_i) + 1 \Rightarrow$$

$$n \cdot H_F(X_1) \leq \frac{1}{n} \sum_{\mathbf{x_1^n} \in \mathcal{X}^n} Pr[\mathbf{X_1^n} = \mathbf{x_1^n}] \cdot l(\mathbf{x_1^n}) < n \cdot H_F(X_1) + 1 \Rightarrow$$

$$H_F(X_1) \leq \sum_{\mathbf{x_1^n} \in \mathcal{X}^n} Pr[\mathbf{X_1^n} = \mathbf{x_1^n}] \cdot l(\mathbf{x_1^n}) < H_F(X_1) + \frac{1}{n}$$

Α συμβολοσειρά $x_1 \cdots, x_n$ δεν δημιουργήθηκε από ανεξάρτητες και ισόνομες τυχαίες μεταβλητές αλλά από κάποια στάσιμη και εργοδική πηγή πληροφορίας τότε από το θεώρημα 2.5 γνωρίζουμε ότι για μεγάλα n η εντροπία $H_F(\mathbf{X_1^n})$ θα συγκλίνει στον ρυθμό εντροπία της πηγής $H(\mathcal{X})$, άρα πάλι θα ισχύει

$$H_F(\mathbf{X_1^n}) \leq \sum_{\mathbf{x_1^n} \in \mathcal{X}^n} Pr[\mathbf{X_1^n} = \mathbf{x_1^n}] \cdot l(\mathbf{x_1^n}) < H_F(\mathbf{X_1^n}) \Rightarrow$$

$$\frac{H(X_1, \cdots, X_n)}{n} \leq \frac{1}{n} \sum_{\mathbf{x_1^n} \in \mathcal{X}^n} Pr[\mathbf{X_1^n} = \mathbf{x_1^n}] \cdot l(\mathbf{x_1^n}) < \frac{H(X_1, \cdots, X_n)}{n} + 1 \overset{n \to \infty}{\Rightarrow}$$

$$H(\mathcal{X}) \leq \frac{1}{n} \sum_{\mathbf{x_1^n} \in \mathcal{X}^n} Pr[\mathbf{X_1^n} = \mathbf{x_1^n}] \cdot l(\mathbf{x_1^n}) < H(\mathcal{X}) \Rightarrow \frac{1}{n} \sum_{\mathbf{x_1^n} \in \mathcal{X}^n} Pr[\mathbf{X_1^n} = \mathbf{x_1^n}] \cdot l(\mathbf{x_1^n}) \to H(\mathcal{X})$$

Με λίγα λόγια το πλεόνασμα που προκύπτει από την χρήση των μηκών $l_1 = \lceil \log_{|F|} \frac{1}{Pr[X = x_i]} \rceil$ μπορεί να αντισταθμιστεί αν κωδικοποιήσουμε αρκετά μεγάλες ακολουθίες συμβόλων αντί για μεμονωμένα σύμβολα.

Κεφάλαιο 4

Κωδικοποίηση πηγής

4.1 Εισαγωγή

Στις προηγούμενες ενότητες παρουσιάσαμε ενδελεχώς τα μέτρα που χρησιμοποιούνται για να μετρήσουμε την πληροφορία που εμπεριέχεται σε μία η περισσότερες τυχαίες μεταβλητές/πηγές. Πέραν της πληροφορίας αυτής καθαυτής που περιέχεται σε μία τυχαία μεταβλητή παρουσιάσαμε μέτρα που αφορούν στην πληροφορία που περιέχεται μεταξύ συσχετίσεων των διαφόρων τυχαίων μεταβλητών, όπως η δεσμευμένη εντροπία, η από κοινού πληροφορία καθώς και η απόσταση Kullback-Leibler. Επίσης εξετάζοντας τις διάφορες σχέσεις που αναπτύσσονται μεταξύ των διαφορετικών μέτρων εξάγαμε κάποια φράγματα όπου είναι εφικτό προκειμένου να καταλάβουμε τα όρια της πληροφορίας που υπάρχουν στο σύνολο των διακριτών μεταβλητών.

Ακόμη αναλύσαμε μαθηματικά την πηγή πληροφορίας όπως την εμπνεύστηκε και περιέγραψε ο Claude E. Shannon. Πέρα από την γενική περιγραφή της διακριτής πηγής πληροφορίας ως μία μαρκοβιανή αλυσίδα, σταθήκαμε στις πηγές που μπορούν να χαρακτηριστούν από στάσιμες και εργοδικές μαρκοβιανές αλυσίδες. Τα μαθηματικά χαρακτηριστικά (στασιμότητα και εργοδικότητα) της συγκεκριμένης κατηγορίας αλυσίδων μας επέτρεψαν να χωρίσουμε τις συμβολοσειρές που είναι δυνατόν να παραχθούν από την πηγή με βάση το αλφάβητο που διαθέτει σε δύο μεγάλες κατηγορίες. Τις τυπικές συμβολοσειρές, το σύνολο των οποίων συγκεντρώνει πιθανότητα που τείνει στο 1 όσο αυξάνεται το μήκος τους και το σύνολο των μη τυπικών συμβολοσειρών που η πιθανότητα του τείνει στο 0 για μεγάλα μήκη. Η παραπάνω κατάσταση διατυπώθηκε και αποδείχθηκε στο θεώρημα της ασυμπτωτικής ισοκατανομής για πηγές με και χωρίς μνήμη. Το σημαντικότερο αποτέλεσμα που προέκυψε από το προαναφερθέν θεώρημα ήταν η δυνατότητα που μας έδωσε για μία αξιόπιστη δειγματοληπτική προσέγγιση της εντροπίας χρησιμοποιώντας μεγάλες ακολουθίες συμβολοσειρών.

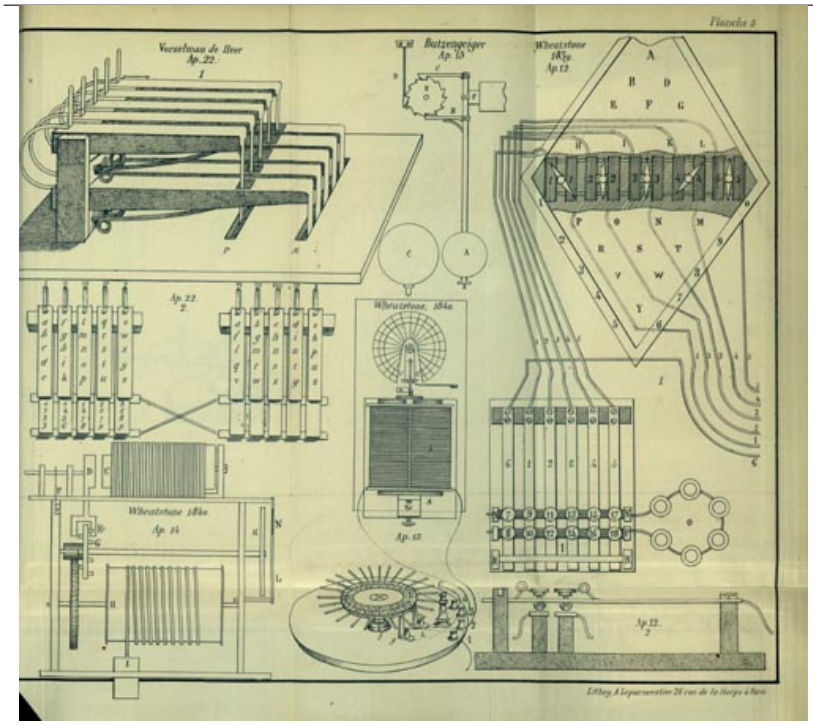
Στο κεφάλαιο της συμπίεσης αναλύσαμε το μαθηματικό υπόβαθρο στο οποίο μπορούμε να βασιστούμε προκειμένου να εκφράσουμε με τον πιο σύντομο και αντιστρέψιμο τρόπο τις τιμές μίας τυχαίας μεταβλητής/πηγής σε ένα άλλο αλφάβητο, δηλαδή να κωδικοποιήσουμε την πηγή. Είδαμε λοιπόν ότι η εντροπία της πηγής είναι αυτή που εκφράζει τον μέσο αριθμό πληροφορίας/σύμβολο που χρειαζόμαστε για να έχουμε μία συμπαγή κωδικοποίηση της πηγής στη γλώσσα ή στο αλφάβητο που επιβάλλουν οι ανάγκες κάποιας πρακτικής εφαρμογής.

Η καρδιά της θεωρίας πληροφορίας όμως δεν βασίζεται μόνο στην ανάλυση και κωδικοποίηση της πηγής αλλά και στην επιτυχία της αξιόπιστης μετάδοσης μεταξύ ενός πομπού και ενός δέκτη. Πως μπορεί όμως να μοντελοποιηθεί μαθηματικά ολόκληρη η διαδικασία της επικοινωνίας;

(α′) Sir William Fothergill Cooke

(β′) Sir Charles Wheatstone

Σχήμα 4.1: Οι εφευρέτες του βρετανικού ηλεκτρικού τηλέγραφου.



(α′) Η πατέντα του ηλεκτρικού τηλεγράφου πέντε βελονών που κατοχυρώθηκε το 1837.

(β′) Ο ηλεκτρικός τηλέγραφος που λειτουργεί με πέντε βελόνες.

Σχήμα 4.2: Ο βρετανικός ηλεκτρικός τηλέγραφος

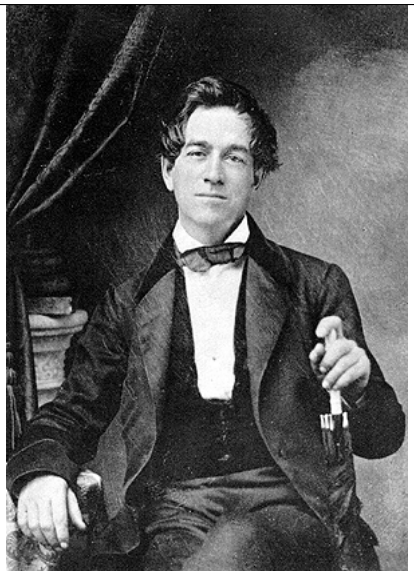
1

1

1. Πηγή εικόνας 4.1(α′):https://upload.wikimedia.org/wikipedia/commons/8/8e/Cooke_William_Fothergill.jpg
2. Πηγή εικόνας 4.1(β′):https://upload.wikimedia.org/wikipedia/commons/3/37/Wheatstone_Charles_drawing_1868.jpg
3. Πηγή εικόνας 4.2(α′):https://kingscollections.org/media/exh_spc/images/006845/C01Teleg_needle.jpg
4. Πηγή εικόνας 4.2(β′):https://web.mst.edu/~kosbar/ee3430/ff/telegraph/needletelegraphs/index.html

Ας γυρίσουμε πίσω στο χρόνο στην αυγή του 1800[2] κατά την οποία συνέβησαν μερικές από τις σημαντικότερες ανακαλύψεις στο πεδίο του ηλεκτρομγνητισμου. Ο νέος αυτός κλάδος της επιστήμης έφερε με την σειρά του στις ζωές των ανθρώπων τον ηλεκτρικό τηλέγραφο. Σχεδόν ταυτόχρονα το 1837 οι Βρετανοί ερευνητές Sir William Fothergill Cooke και Sir Charles Wheatstone καθώς και ο Αμερικάνος καθηγητής ζωγραφικής και γλυπτικής Samuel F.B. Morse κατοχύρωσαν από μια πατέντα για τον ηλεκτρικό τηλέγραφο. Ο τηλέγραφος των Sir William Fothergill Cooke και Sir Charles Wheatstone αποτελούνταν από πέντε μαγνητισμένες βελόνες που μετακινούνταν χάρη στην ηλεκτρομαγνητική επαγωγή όταν διαρρέοντα από ρεύμα συγκεκριμένης έντασης. Οι βελόνες στρέφονταν γύρω από τον άξονα τους δείχνοντας σε συγκεκριμένα σημεία ενός διαγράμματος που υπήρχαν σε μία πλακέτα πάνω στον τηλέγραφο τα οποία σημεία αντιστοιχούσαν σε συγκεκριμένα γράμματα.

(α′) Samuel F.B Morse

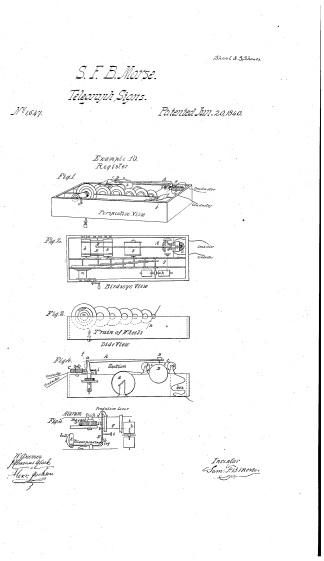


(β′) Alfred Vail

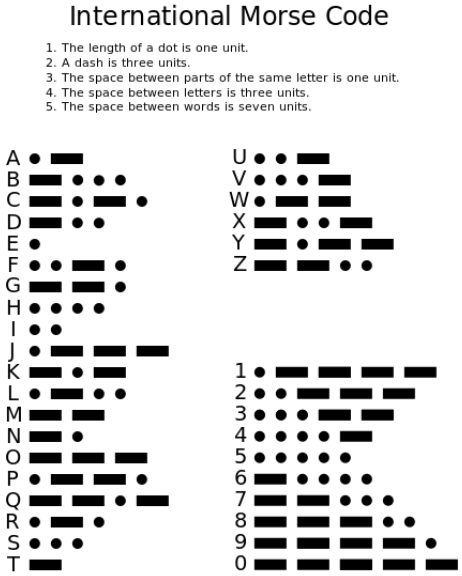
Σχήμα 4.3: Οι εφευρέτες του αμερικάνικου ηλεκτρικού τηλέγραφου.

Από την άλλη ο ηλεκτρικός τηλέγραφος που πρότεινε ο Morse μετά από σημαντικές παρεμβάσεις βελτιώσεις του μηχανικού Alfred Vail είναι ο ευρέως γνωστός μέχρι σήμερα τηλέγραφος που κωδικοποιεί τα ηλεκτρικά σήματα που δημιουργούνται χάρη στην βοήθεια ενός κλειδιού τηλεγράφησης σε τελείες, παύλες και κενά. Το κάθε σύμβολο διαχωρίζεται από τα υπόλοιπα βάση του χρόνου που κρατάμε το κύκλωμα του τηλεγράφου ανοιχτό και κλειστό. Τα γράμματα τις αλφαβήτου αντιστοιχίζονται σε ακολουθίες συμβόλων που είναι σήμερα γνωστές ως ο κώδικας Morse. Μπορούμε να πούμε ότι μία από τις πιο συγκινητικές ιστορικές στιγμές συναντάται το 1843 όταν ο Morse έχοντας εξασφαλίσει την οικονομική υποστήριξη του αμερικάνικου κράτους κάνει πραγματικότητα την εγκατάσταση της πρώτης γραμμής τηλεγράφησης μεταξύ Ουάσινγκτον και Βαλτιμόρης. Το σύστημα τηλεγράφησης γίνεται ανοικτό προς χρήση στο κοινό το Μάιο του 1844 με το πρώτο μήνυμα που μεταδίδεται να είναι το What hath god wrought, το οποίο είναι αρχαΐζων κείμενο της βίβλου από το βιβλίο των αριθμών και σημαίνει “ Θεέ μου τι έκανες;”. Τα συστήματα τηλεγράφησης ήταν από τα πρώτα αυτόματα συστήματα επικοινωνίας που αποτελούντα από τον πομπό, τον δέκτη και το κανάλι επικοινωνίας (καλώδιο τηλεγράφησης).

[2] **TelegraphHistory**.

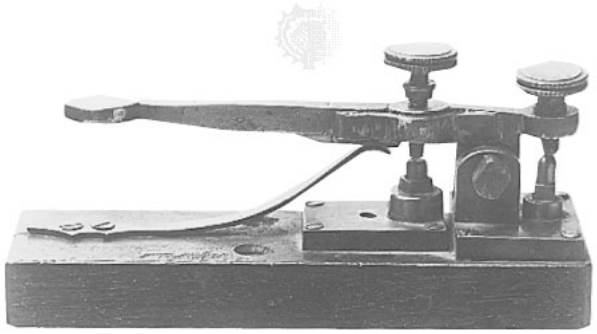


(α′) Η πατέντα του αμερικάνικου ηλεκτρικού τηλεγράφου που κατοχύρωσε ο Morse το 1837.



(β′) Ο κώδικας Morse.

Σχήμα 4.4: Οι Αμερικάνοι εφευρέτες του ηλεκτρικού τηλέγραφου



Σχήμα 4.5: Το ηλεκτρικό κλειδί τηλεγράφησης.

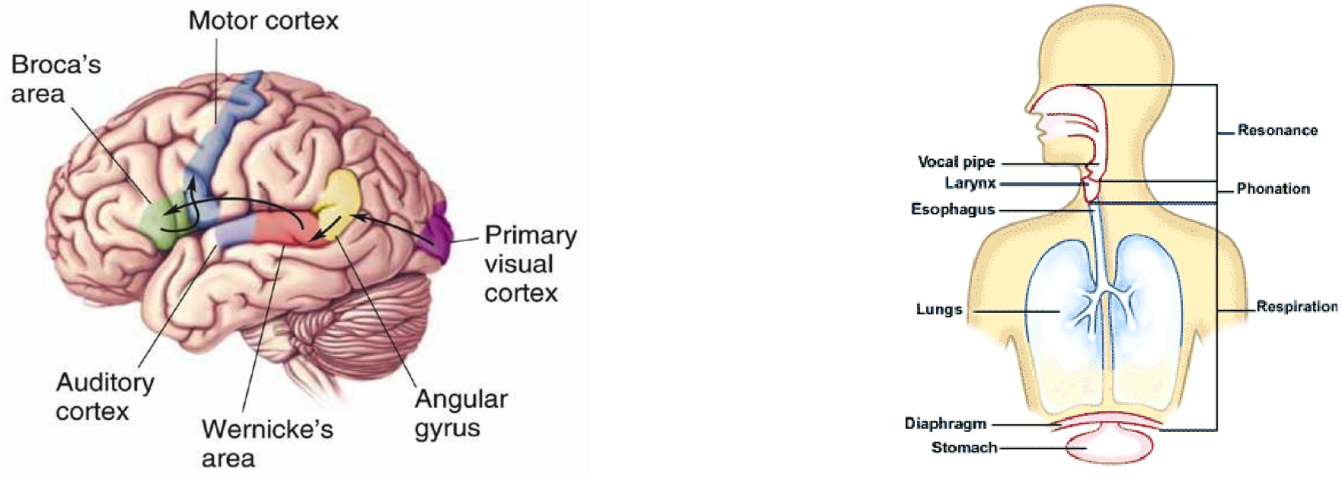
[3]

Ένα άλλο σύστημα επικοινωνίας που υπάρχει περίπου από την αρχή της ανθρωπότητα, συγκεκριμένα μετά την ανακάλυψη της ομιλίας είναι ο άνθρωπός. Οι άνθρωποι χρησιμοποιώντας σχεδόν όλο τους το σώμα καταφέρνουν την παραγωγή και κωδικοποίηση άπειρων μηνυμάτων καθημερινά. Για να εξηγήσουμε την διαδικασία

[3]

1. Πηγή εικόνας 4.3(α′):https://upload.wikimedia.org/wikipedia/commons/8/8d/Samuel_Morse_1840.jpg
2. Πηγή εικόνας 4.3(β′):https://upload.wikimedia.org/wikipedia/commons/c/c9/Alfred_Vail.GIF
3. Πηγή εικόνας 4.4(α′):https://patentimages.storage.googleapis.com/6b/04/d5/a2487df4a7b8ba/US1647-drawings-page-3.png
4. Πηγή εικόνας 4.4(β′):https://en.wikipedia.org/wiki/File:International_Morse_Code.svg
5. Πηγή εικόνας 4.5:https://cdn.britannica.com/25/60525-004-3A3EFC53/telegraph-transmitter-Morse.jpg

της επικοινωνίας στον άνθρωπο θα βασιστούμε στο μοντέλο Wernicke–Geschwind model[45] το οποίο μπορεί στον κλάδο της νευρολογίας να θεωρείτε ξεπερασμένο αλλά χρησιμοποιείται ακόμη σαν εισαγωγή στην κατανόηση των εγκεφαλικών λειτουργιών κατά τη διάρκεια της ομιλίας. Όταν θέλουμε να μιλήσουμε το πρώτο σημείο του εγκεφάλου που ενεργοποιείται είναι η περιοχή Broca που δίνει εντολές στο σώμα. Συγκεκριμένα[6] στην αρχή γεμίζουμε τα πνευμόνια μας με αέρα. Κατά την εκπνοή ο αέρας ανεβαίνει από τους πνεύμονες προς τον λάρυγγα όπου βρίσκονται οι φωνητικές χορδές. Η άνοδος του αέρα προκαλεί την δόνηση των φωνητικών χορδών η οποία δόνηση με τη σειρά της παράγει ένα ηχητικό κύμα. Τα ηχητικά κύματα που παράγονται με αυτό τον τρόπο έπειτα με τη χρήση της γλώσσας, της σιαγόνας, των δοντιών κ.λ.π μετασχηματίζονται σε φθόγγους (φωνήματα). Η επόμενη περιοχή του εγκεφάλου που έχει σχέση με την ομιλία και είναι αναγκαία για την επικοινωνία είναι η περιοχή Wernicke που συνδέεται μέσω νευρώνων με την περιοχή Broca. Η περιοχή Wernicke βοηθάει την επεξεργασία και την κατανόηση της γλώσσας.



(α′) Οι περιοχές Broca και Wernicke. (β′) Η ανατομία της ομιλίας.

Σχήμα 4.6: Τα βιολογικά συστατικά της ανθρώπινης ομιλίας

[7]

Καταλαβαίνουμε λοιπόν πως όταν θέλουμε να επικοινωνήσουμε είτε ανθρώπινα είτε μέσω μηχανημάτων οι οντότητες που εμπλέκονται στην διαδικασία απαρτίζονται από ένα πομπό, ένα μήνυμα, την κωδικοποίηση του σε κατάλληλη μορφή προκειμένου να μεταδοθεί στον προορισμό, ένα κανάλι επικοινωνίας που θα μεταδώσει το μήνυμα, ένα αποκωδικοποιητή και ένα δέκτη που θα το παραλάβει. Στον τηλέγραφο για παράδειγμα ο πομπός είναι ο χειριστής του τηλεγράφου, το μήνυμα είναι το "Θεέ μου τι έκανες", η κωδικοποίηση του έγκειται στην μετατροπή του σε μία ακολουθία από τελείες, παύλες και κενά ,το κανάλι επικοινωνίας είναι το ηλεκτρικό καλώδιο, ο αποκωδικοποιητής είναι είτε κάποιο αυτοματοποιημένο μηχάνημα ή ο ίδιο ο δέκτης.

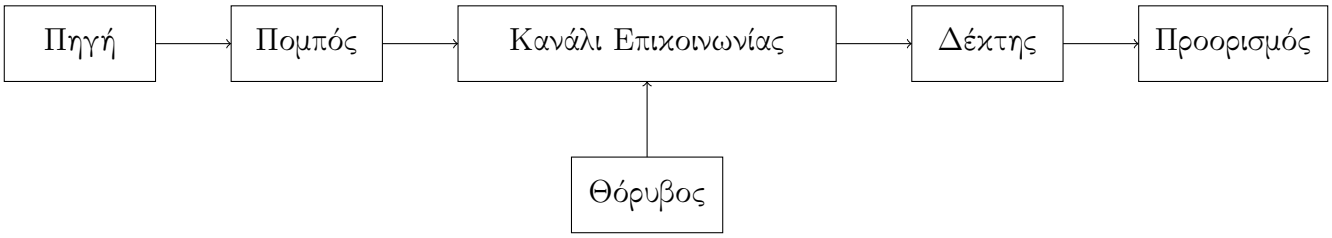
4.2 Το πρόβλημα της επικοινωνίας

Αν παρατηρήσουμε τα παραδείγματα επικοινωνίας που παρουσιάστηκαν στην προηγούμενη ενότητα μπορούμε να αντιληφθούμε ότι υπάρχει ένα μοτίβο που διαπερνάει τα παραπάνω συστήματα και είναι κοινό για όλα. Το

[4] **speechbrain2004Boeree**.
[5] **speechbrainLanguage**.
[6] **SpeechBrainNTUA**.
[7]

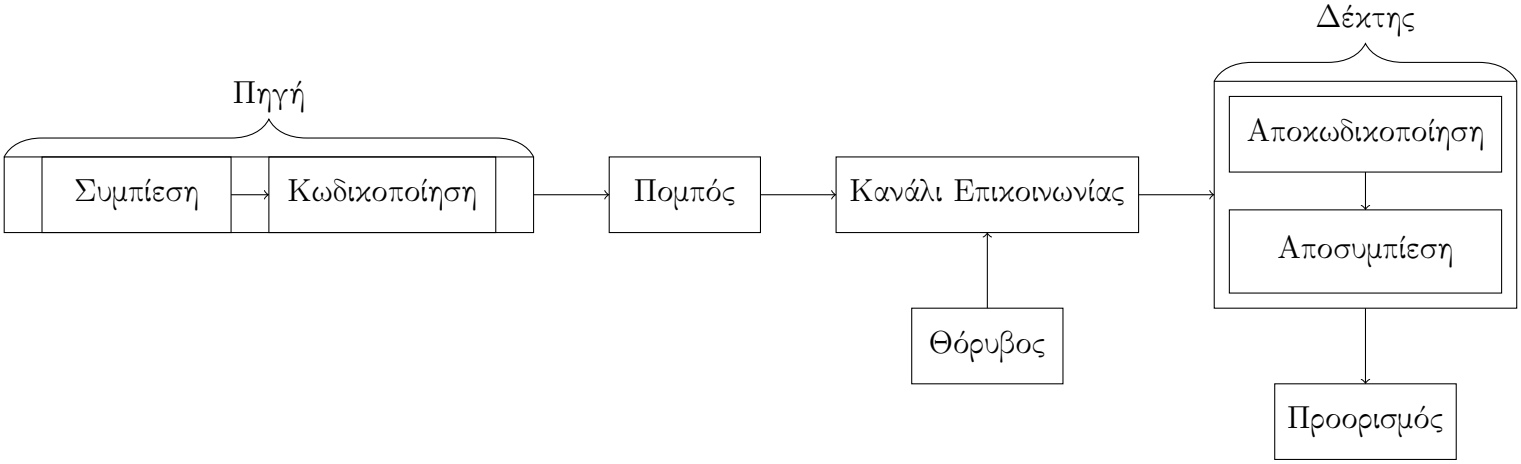
1. Πηγή εικόνας 4.6(α′):https://brainstuff.org/blog/what-is-the-wernicke-geschwind-model
2. Πηγή εικόνας 4.6(β′):https://www.researchgate.net/figure/Stages-of-voice-production-Source-Anatomy-and-Physi fig1_330546181

μοτίβο αυτό έχει να κάνει με την διαίρεση της επικοινωνίας σε τρία πολύ βασικά στάδια την παράγωγή, τη μετάδοση και την λήψη ενός μηνύματος. Εύκολα μπορούμε να αναγνωρίσουμε και τα τρία στάδια σε όλα τα παραπάνω συστήματα επικοινωνίας. Για παράδειγμα η παραγωγή του μηνύματος κατά τη διάρκεια της ανθρώπινης ομιλίας γίνεται με τη χρήση του εγκεφάλου και των μυών του λάρυγγα και της στοματικής κοιλότητας ενώ στον τηλέγραφο η παραγωγή γίνεται με τη χρήση του κλειδιού τηλεγράφησης. Η μετάδοση της ανθρώπινης ομιλίας γίνεται μέσω του αέρα ενώ στον τηλέγραφο χρησιμοποιούμε ένα καλώδιο που μεταδίδει το μήνυμα από την μία άκρη στην άλλη. Οι επιστήμονες που μελετούσαν την διαδικασία της επικοινωνίας από μια θεωρητική σκοπιά κατά τον $20^ο$ αιώνα προφανώς είχαν καταλάβει την επανάληψη του παραπάνω προτύπου και για το λόγο αυτό πρότειναν μία σχηματική διαμέριση και απεικόνιση των λειτουργιών που εκτελούνται κατά την διαδικασία της επικοινωνίας, η οποία ονομάζεται βασικό μοντέλο επικοινωνίας (Shannon-Weaver model[8]) και παρουσιάζεται στο παρακάτω σχήμα·



Σχήμα 4.7: Το βασικό μοντέλο επικοινωνίας

Η επιτυχία του βασικού μοντέλου επικοινωνίας έγκειται στην ευελιξία του να προσαρμόζεται και κάθε τμήμα του να υποδιαιρείται σε επιπρόσθετες λειτουργίες όταν το σύστημα γίνεται πιο σύνθετο. Αν χρειάζεται λόγου χάρη ένα μήνυμα να συμπιεστεί και να κωδικοποιηθεί τότε η πηγή και ο δέκτης επιφορτίζονται με τις λειτουργίες της συμπίεσης/αποσυμπίεσης και κωδικοποίησης/αποκωδικοποίησης αντίστοιχα. Γενικά ο κατακερματισμός μια σύνθετης διαδικασίας σε πιο απλές είναι το κλειδί της σχεδίασης και λειτουργίας όλων των σύγχρονων συστημάτων επικοινωνίας.



Σχήμα 4.8: Το μοντέλο επικοινωνίας επιφορτισμένο με τις διαδικασίες της συμπίεσης/αποσυμπίεσης και κωδικοποίησης/αποκωδικοποίησης

Ας δούμε λίγο πιο αναλυτικά τα μέρη που λαμβάνουμε μέρος στο βασικό μοντέλο της επικοινωνίας

• Η πηγή μπορεί να είναι είτε συνεχής είτε διακριτή. Μια συνεχής πηγή είναι η κεραία ενός πύργου που μεταδίδει μέσω ηλεκτρομαγνητικών κυμάτων την αγαπημένη μας ραδιοφωνική εκπομπή. Ως μία διακριτή πηγή

[8] **shannon1949mathematical**.

μπορούμε να θεωρήσουμε τα σύμβολα που πληκτρολογούμε όταν θέλουμε να γράψουμε κάποιο μήνυμα στο κινητό μας τηλέφωνο.

• Ο πομπός είναι επιφορτισμένος με την μετατροπή της πληροφορίας που παράγει η πηγή σε κατάλληλη μορφή προκειμένου να μεταδοθεί από το κανάλι επικοινωνίας. Για παράδειγμα στην ανθρώπινα ομιλία ο πομπός είναι η ταλάντωση των φωνητικών χορδών οι οποίες αλλάζουν την πίεση του ρεύματος αέρα που έρχεται από τους πνεύμονες παράγοντας έτσι ένα ηχητικό κύμα. Μία ακόμη περίπτωση διακριτού πομπού είναι η κωδικοποίηση κειμένου (ANSI, UTF-8, UTF-16) που μετατρέπει τα σύμβολα ενός πεπερασμένου και διακριτού αλφαβήτου σε συμβολοσειρές δυαδικών ακολουθιών.

• Το κανάλι επικοινωνίας μπορεί να είναι το φυσικό μέσο που χρησιμοποιούμε για τη μετάδοση τις πληροφορίας όπως ένα ομοαξονικό καλώδιο ή ένα καλώδιο οπτικών ινών ακόμα και τον νερό της θάλασσας ή ο ατμοσφαιρικός αέρας. Ανάλογα την εφαρμογή τα επιμέρους μέρη που συμμετέχουν στην επικοινωνία μπορούν να θεωρηθούν ώς μέρος του καναλιού. Για παράδειγμα το ράδιο που είναι μία εφαρμογή όπου το κανάλι επικοινωνίας είναι ο αέρας, μέρος του καναλιού μπορούν να θεωρηθούν οι ενισχυτές οι αναμεταδότες και οι κεραίες που χρησιμοποιούνται για μεταφορά του σήματος από τη μία άκρη στην άλλη. Αν θέλουμε όμως να ορίσουμε πιο αφαιρετικά και γενικά το κανάλι επικοινωνίας μπορούμε να πούμε πως αποτελεί το πιθανοκρατικό μοντέλο που περιγράφει τη σχέση μεταξύ των εισόδων και των εξόδων του. Αυτή τη προσέγγιση θα χρησιμοποιήσουμε αργότερα για να αναλύσουμε συστήματα επικοινωνίας όπου η πηγή και το σήμα αποτελούν διακριτές τυχαίες μεταβλητές.

• Ο δέκτης κάνει την αντίστροφη διαδικασία από ότι ο πομπός. Δηλαδή μετατρέπει το σήμα που παρέλαβε από το κανάλι επικοινωνίας στο μήνυμα αυτό καθαυτό ή σε μία προσέγγιση του.

• Ο προορισμός είναι το σημείο για το οποίο στάλθηκε το μήνυμα. Ο προορισμός μπορεί να είναι είτε μία ακόμη συσκευή, π.χ ένας αναμεταδότης ή ένας άνθρωπος, π.χ ο συνομιλητής μας σε μία τηλεφωνική κλήση.

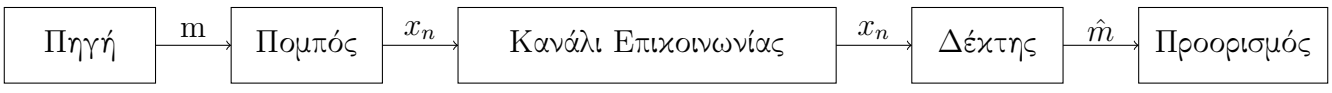
• Θόρυβος είναι κάθε διακριτή η συνεχής διαταραχή που μπορεί να επηρεάσει, να αλλάξει ή να καταστρέψει το σήμα κατά τη διάρκεια της μετάδοσης του.

Τα συστήματα επικοινωνίας ανάλογα τη φύση του μηνύματος και του σήματος μπορούν να κατηγοριοποιηθούν σε α) διακριτά, αν το μήνυμα και το σήμα αποτελούν διακριτές τυχαίες μεταβλητές όπως γίνεται στη περίπτωση του τηλέγραφου που το μήνυμα είναι ένας πεπερασμένος αριθμός λέξεων ενώ το σήμα είναι μία πεπερασμένη ακολουθία από τελείες και παύλες, β) συνεχή, αν το μήνυμα και το σήμα είναι συνεχείς τυχαίες μεταβλητές όπως στην περίπτωση του ραδιοφώνου που το μήνυμα είναι ένα ηχητικό κύμα και το σήμα είναι ένα ηλεκτρομαγνητικό κύμα ή γ) μικτά, όπου παρουσιάζονται διακριτές και συνεχείς τυχαίες μεταβλητές.

Το θεμελιώδες πρόβλημα στην επικοινωνία είναι η αξιόπιστη μεταφορά ενός μηνύματος από τη μία άκρη στη άλλη. Η ύπαρξη του θορύβου πολλές φορές έχει σαν αποτέλεσμα να αλλοιώσει το σήμα προς μετάδοση. Τα λάθη που εισάγονται στο σήμα κατά τη μετάδοση του από το θόρυβο μπορούν πολλές φορές να καταστήσουν αδύνατη την μετατροπή του στο αρχικό μήνυμα. Αυτό που κατάφερε να αποδείξει ο Claude Shannon το οποίο διαισθητικά δεν ήταν καθόλου προφανές είναι ότι παρουσία θορύβου μπορούμε κατόπιν μίας ‘‘έξυπνης’’ κωδικοποίησης να μεταδώσουμε την πληροφορία με πολύ μικρή πιθανότητα λάθους εάν ο ρυθμός μετάδοσης που επιλέγουμε δεν υπερβαίνει ένα ανώτατο όριο που ονομάζεται χωρητικότητα του καναλιού. Σκοπός λοιπόν αυτού του κεφαλαίου είναι να παρουσιάσει τα αποτελέσματα της θεωρίας του Shannon για την χωρητικότητα διακριτών καναλιών .

4.3 Διακριτά συστήματα Επικοινωνίας

4.3.1 Χωρητικότητα για κανάλια χωρίς θόρυβο



Σχήμα 4.9: Το βασικό μοντέλο επικοινωνίας για κανάλια χωρίς θόρυβο

Η μελέτη μας θα ξεκινήσει με την πιο απλή περίπτωση διακριτών συστημάτων επικοινωνίας, στα οποία απουσιάζει ο παράγοντας του θορύβου. Υπενθυμίζουμε ότι στα διακριτά συστήματα επικοινωνίας το μήνυμα και το σήμα μπορούν να αναπαρασταθούν από ακολουθίες διακριτών συμβόλων. Συγκεκριμένα η διακριτή πηγή μπορεί να μοντελοποιηθεί ως μία μαρκοβιανή αλυσίδα πεπερασμένης τάξης. Για να προχωρήσουμε ένα βήμα παραπάνω την μελέτη μας θα πρέπει να ορίσουμε μαθηματικά τη λειτουργία του πομπού και του δέκτη. Επειδή η λειτουργία του πομπού και του δέκτη σε διακριτά κανάλια χωρίς θόρυβο περιορίζεται στην μετατροπή των διακριτών συμβόλων της πηγής σε διακριτά σύμβολα κάποιου κωδικού αλφαβήτου και αντίστροφα κατά τη κωδικοποίηση και την αποκωδικοποίηση του μηνύματος, στη δημοσίευση του,ο Shannon ονόμασε τα παραπάνω μέρη του μοντέλου με την ενιαία ορολογία **διακριτός μετατροπέας** (discrete transducer).

Ορισμός 4.1. *Έστω $\mathcal{X}$ ένας πεπερασμένο αλφάβητο εισόδου, $\mathcal{S}$ ένα πεπερασμένο σύνολο καταστάσεων και $\mathcal{Y}$ το πεπερασμένο σύνολο συμβόλων εξόδου. Τότε ως διακριτός μετατροπέας ορίζεται το ζεύγος συναρτήσεων (y_n, a_{n+1}) με:*

- *$y_n = f(x_n, a_n)$, δηλαδή το σύμβολο εξόδου τη χρονική στιγμή n ορίζεται συναρτήσει του συμβόλου εισόδου x_n και της κατάστασης a_n που βρίσκεται ο μετατροπέας κατά την χρονική στιγμή n*
- *$a_{n+1} = g(x_n, a_n)$, δηλαδή η κατάσταση που θα μεταβεί κατά την χρονική στιγμή $n+1$ υπολογίζεται συναρτήσει του συμβόλου εισόδου x_n και της κατάστασης a_n που βρίσκεται ο μετατροπέας κατά την χρονική στιγμή n.*

Ο παραπάνω ορισμός αναφέρεται σε έναν διακριτό μετατροπέα χωρίς μνήμη καθώς το σύμβολο που θα παραχθεί από τον μετατροπέα και η κατάσταση στην οποία θα μεταβεί εξαρτάται μόνο από το σύμβολο εισόδου εκείνη τη στιγμή. Αν θέλουμε ο μετατροπέας να έχει κάποια πεπερασμένη μνήμη, έστω l, μπορούμε να επεκτείνουμε τον ορισμό ώστε να συμπεριλάβουμε μέσα στις συναρτήσεις τα προηγούμενα l σύμβολα από τα οποία θα εξαρτάται το $y_n = f(\{x_n, x_{n-1}, \cdots x_{n-l}\}, a_n)$ και η $a_{n+1} = g(\{x_n, x_{n-1}, \cdots x_{n-l}\}, a_n)$.

Ορισμός 4.2. *Η χωρητικότητα ενός διακριτού διαύλου χωρίς θόρυβο δίνεται από τη σχέση*

$$C = \lim_{T\to\infty} \frac{log_2 N(T)}{T} \; bits/\sec, \tag{4.1}$$

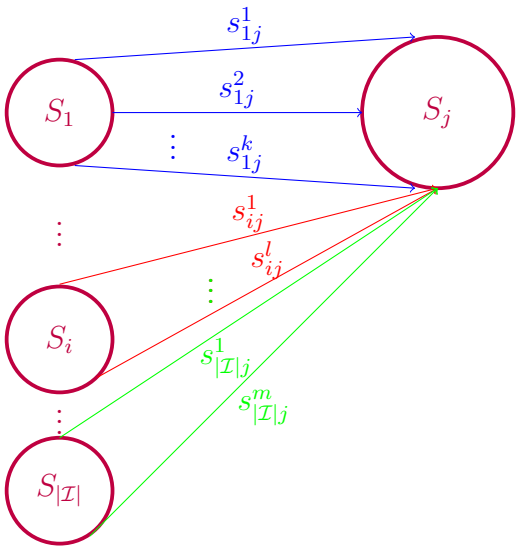
όπου N(T) το πλήθος των επιτρεπτών μηνυμάτων κατά τη χρονική διάρκεια T.

Όπως η διακριτή πηγή της πληροφορίας μπορεί να μοντελοποιηθεί σαν μία μαρκοβιανή αλυσίδα τάξης n έτσι και οι ακολουθίες συμβόλων εξόδου που παράγονται από τον διακριτό μετατροπέα-πομπό, δηλαδή η κωδικοποιημένη μορφή των μηνυμάτων της πηγής, μπορούν να παραχθούν σαν αποτέλεσμα κάποιας μαρκοβιανής αλυσίδα τάξης k, οι καταστάσεις της οποίας καθορίζουν τα σύμβολα που θα ακολουθήσουν.

▷ Αρχικά θα πρέπει να δούμε πως μπορούμε να καταμετρήσουμε το πλήθος των επιτρεπτών μηνυμάτων που παράχθηκαν στο χρονικό διάστημα $(0, T]$. Επειδή τα μηνύματα αποτελούν αποτέλεσμα μιας μαρκοβιανής αλυσίδας που εξελίσσεται στο χρονικό διάστημα $(0, T]$ καταλαβαίνουμε ότι κάθε μήνυμα όταν ολοκληρώνεται τερματίζει σε κάποια κατάσταση S_j του χώρου καταστάσεων $\mathcal{S}$. Το ίδιο μήνυμα μπορεί να μεταδοθεί παραπάνω από μία φορά στο χρονικό διάστημα $(0, T]$. Άρα μία αρχική προσέγγιση θα ήταν να μετρήσουμε το πλήθος των πιθανών μηνυμάτων που παράχθηκαν κατά τη χρονική διάρκεια $(0, T]$ και κατέληξαν στην κατάσταση S_j. Το πλήθος αυτών των μηνυμάτων το συμβολίζουμε με $N_j(T)$. Αν θέλουμε να φανταστούμε γραφικά την παραπάνω κατάσταση, μπορούμε να υποθέσουμε ένα πεπερασμένο διάγραμμα καταστάσεων στο οποίο κατά τη διάρκεια $(0, T]$ ενεργοποιούνται συγκεκριμένα μονοπάτια. Σκοπός μας είναι να μετρήσουμε το πλήθος των μονοπατιών που ενεργοποιήθηκαν στο διάσημα $(0, T]$

▷ Αντί να μετρήσουμε το πλήθος των μονοπατιών που κατέληξαν στην κατάσταση S_j από την αρχή του χρόνου, μπορούμε να μετρήσουμε το πλήθος των μονοπατιών που κατέληξαν από την κατάσταση S_i στην S_j κάνοντας χρήση κάποιου συμβόλου s το οποίο έχει διάρκεια t_{ij}^s ($S_i \xrightarrow{s} S_j$).

$$N_j(T) = \sum_{s\in S} \sum_{i\in\mathcal{I}} N_i(T - t_{ij}^s) \tag{4.2}$$



Σχήμα 4.10: Το μπλε συμβολίζει τα k διαφορετικά μονοπάτια που μας πάνε από την κατάσταση S_1 στην S_j, το κόκκινο τα l διαφορετικά μονοπάτια $S_i \to S_j$ και το πράσινο τα m διαφορετικά μονοπάτια $S_{|\mathcal{I}|} \to S_j$.

▷ Αν καταφέρουμε να μετρήσουμε το $N_j(T)$, τότε το πλήθος των επιτρεπτών μηνυμάτων είναι εύκολο να μετρηθεί αν αθροίσουμε πάνω σε όλες τις καταστάσεις S_j, δηλαδή:

$$N(T) = \sum_{j \in \mathcal{J}} N_j(T) \tag{4.3}$$

Βασικές γνώσεις στις γραμμικές εξισώσεις πεπερασμένων διαφορών[9]

Το θεώρημα που ακολουθεί βασίζεται σε αποτελέσματα των γραμμικών εξισώσεων πεπερασμένων διαφορών. Για το λόγο αυτό πριν περάσουμε στην απόδειξή του θα υπενθυμίσουμε κάποιες βασικές έννοιες στις πεπερασμένες διαφορές. Ο αναγνώστης που είναι εξοικειωμένος με το θέμα, μπορεί να παραλείψει την παρούσα ενότητα.

Η γενική μορφή μιας εξίσωσης πεπερασμένων διαφορών δίνεται από την σχέση:

$$X(n+k) = F(X(x+k-1),\ X(n+k-2), \cdots, X(n), n)$$

Παραδείγματα (1). $X(n-1) + X^2(n) = 2 \cdot X(n-2)$
(2). $X(n) - 2 \cdot X(n-1) + 3X(n-3) = 5$
(3). $sin(X(n)) = cos(X(n-1))$

Ορισμός 4.3. *Ως τάξη της εξίσωσης ορίζουμε τη διαφορά του μεγαλύτερου από τον μικρότερο δείκτη.*

Στα παραπάνω παραδείγματα η (1) έχει τάξη 2, η (2) έχει τάξη 3 και η (3) έχει τάξη 1.

Ορισμός 4.4. *Μία εξίσωση πεπερασμένων διαφορών λέγεται γραμμική αν έχει την παρακάτω μορφή:*

$$X(n+k) + a_1(n) \cdot X(x+k-1) + a_2(n) \cdot X(n+k-2) + \cdots + a_k(n) \cdot X(n) = R(n) \tag{4.4}$$

, όπου $a_i(n)$, $R(n)$ είναι συναρτήσεις του n.

[9] **FiniteDifferences**.

Στα παραδείγματα μόνο η (2) είναι γραμμική, οι υπόλοιπες περιέχουν συναρτήσεις του $X(n)$.

Η λύση των ομογενών ($R(n) = 0$) γραμμικών εξισώσεων με σταθερούς συντελεστές μοιάζει με την μέθοδο που ακολουθούσαμε για να λύσουμε συνήθεις γραμμικές διαφορικές εξισώσεις τάξης n με σταθερούς συντελεστές. Όπως στις συνήθεις διαφορικές έτσι και εδώ για να βρούμε την λύση της (4.4) υπολογίζουμε την χαρακτηριστική εξίσωση και βρίσκουμε τις ρίζες τις.

Αν $X(n+k) + a_1 \cdot X(x+k-1) + a_2 \cdot X(n+k-2) + \cdots + a_k \cdot X(n) = R(n)$ τότε η χαρακτηριστική εξίσωση είναι η:

$$r^k + a_1 \cdot r^{k-1} + a_2 \cdot r^{k-2} + \cdots + a_{k-1} r + a_k = 0$$

• Αν οι λύσεις της χαρακτηριστικής είναι όλες διαφορετικές μεταξύ τους, τότε για κάθε λύση r_i το σύνολο $\{c \cdot r^i\}$ θα αποτελεί ένα θεμελιώδες σύνολο λύσεων και η γενική λύση θα δίνεται από την σχέση:

$$X(n) = a_1 r_1^n + \cdots c_k r_k^n$$

Για την συγκεκριμένη περίπτωση αν $a_1 > a_2 \cdots > a_k$, τότε για $n \to \infty$, θα κυριαρχεί ο πρώτος όρος και

$$X(n) \approx a_1 r_1^n$$

• Αν κάποια ρίζα της έχει πολλαπλότητα m, τότε θα έχουμε:

$$X(n) = (a_1 + n \cdot a_2 + \cdots + n^{m-1} a_m) r^n + a_{m-1} r^{n-1} + \cdots + a_k$$

• Αν οι λύσεις της χαρακτηριστικής είναι μιγαδικές τότε θα έχουμε ζεύγη συζυγών r, $\overline{r}$.

Θεώρημα 4.1. *Έστω A ένας πίνακας με στοιχεία:*

$$a_{ij} = \sum_s X^{-t_{ij}^s} - \delta_{ij}, \tag{4.5}$$

όπου δ_{ij} είναι το σύμβολο kronocker ($\delta_{ij} = 1$, αν $i = j$ και $\delta_{ij} = 0$ για $i \neq j$) και t_{ij}^s η διάρκεια του συμβόλου s που οδηγεί την αλυσίδα από την κατάσταση S_i στην S_j. Τότε η χωρητικότητα ενός αθόρυβου διακριτού καναλιού με σύμβολα άνισης διάρκειας δίνεται από τη σχέση:

$$C = log X_0, \tag{4.6}$$

όπου x_0 είναι η μεγαλύτερη θετική λύση της ορίζουσας του πίνακα $A = \{a_{ij}\}$.

Απόδειξη

Το $N_j(T) = \sum_{i \in \mathcal{I}} \sum_{s \in S} N_i(T - t_{ij}^s)$ αποτελεί μία γραμμική εξίσωση πεπερασμένων διαφορών με γραμμικούς συντελεστές. Η λύση της εξίσωσης αυτή θα έχει την μορφή:

$$N_j(T) = a_j \cdot \mathcal{X}^T$$

, όπου $\mathcal{X}$ το αλφάβητο της πηγής. Η σχέση αυτή μπορεί να κατανοηθεί και διαισθητικά. Σε χρόνο T μία πηγή χωρίς περιορισμούς μπορεί να παράξει το πολύ $\mathcal{X}^T$ μηνύματα. Άρα το $a_j \mathcal{X}^T$ αποτελεί την μερίδα εκείνη των μηνυμάτων που τερματίζουν στην κατάσταση S_j

Άρα η εξίσωση $N_j(T) = \sum_{i \in \mathcal{I}} \sum_{s \in S} N_i(T - t_{ij}^s)$ μπορεί να γραφεί ως:

$$a_j \mathcal{X}^T = \sum_{i \in \mathcal{I}} \sum_{s \in S} a_i \mathcal{X}^{T - t_{ij}^s}$$

Γράφοντας το $a_j = a_i \cdot \delta_{ij}$, όπου δ_{ij} το δέλτα του Kronecker η εξίσωση γίνεται

$$\sum_{i \in \mathcal{I}} a_i \cdot \left(\sum_{s \in S} a_i \mathcal{X}^{T - t_{ij}^s} - \delta_{ij} \right) = 0$$

Επειδή υπάρχει η περίπτωση κάποιο a_i να είναι μηδέν, έπεται ότι οι ποσότητες μέσα στις αγκύλες είναι εξαρτημένες γεγονός που με τη σειρά του συνεπάγεται ότι η Wronksian πρέπει να είναι μηδέν.

Έστω X_0 η μεγαλύτερη ρίζα, τότε το $N_j(T) = a_j X_0^T$ και $N(T) = \sum_{j\in\mathcal{J}} X_0^T$.

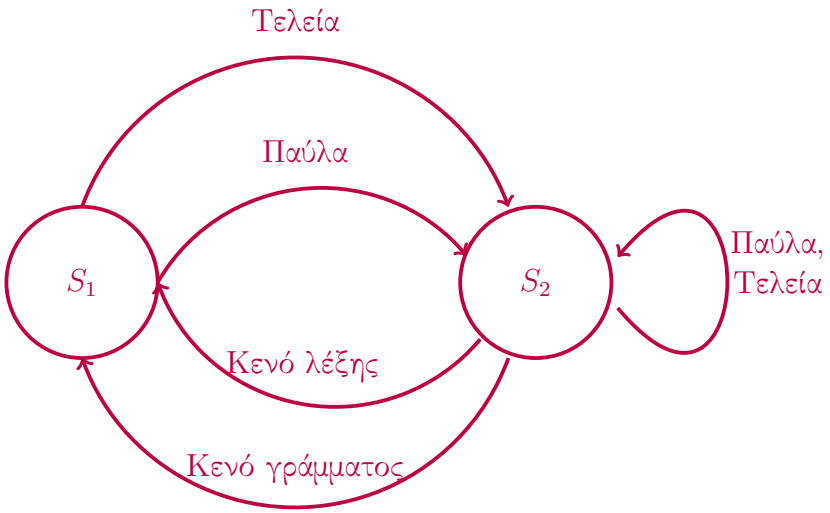
Τότε σύμφωνα με τον ορισμό της χωρητικότητα:

$$C = \lim_{T\to\infty} \frac{log_2 N(T)}{T} = \lim_{T\to\infty} \frac{log_2(\sum_{j\in\mathcal{J}} a_j X_0^T)}{T} = \frac{log_2(\sum_{j\in\mathcal{J}} a_j)}{T} + \frac{(T \cdot log X_0)}{T} = log_2 X_0.$$

Παράδειγμα 4.1. *Βρείτε την χωρητικότητα του κώδικα Morse, ο οποίος αποτελείται από από τέσσερα σύμβολα μια τελεία, μία παύλα, ένα κενό γράμματος και ένα κενό λέξης με τον περιορισμό ότι το ένα κενό δεν μπορεί να διαδέχεται το άλλο. Ο χρόνος μετάδοσης της τελείας είναι 2, της παύλα 4, του κενού γράμματος 3 και του κενού λέξης 6.*

Λύση

Η αναπαράσταση της πηγής που μεταδίδει ο κώδικας Morse μπορεί να γίνει μέσω των περιορισμών της. Από τη στιγμή που τα κενά δεν γίνεται να διαδέχονται το ένα το άλλο, οι επιτρεπτές καταστάσεις που μπορεί να υπάρξουν για την πηγή είναι είτε να παραχθεί κάποια τελεία ή παύλα και το προηγούμενο σύμβολο μπορεί να είναι ότι θέλει ή να παραχθεί κάποια τελεία η παύλα και το προηγούμενο σύμβολο να είναι κάποιο από τα κενά. Οι κατάστάσεις παρουσιάζονται στο παρακάτω διάγραμμα καταστάσεων.



Σχήμα 4.11: Το διάγραμμα καταστάσεων μίας πηγής που παράγει κώδικα Morse

Για να βρούμε τη χωρητικότητα πρέπει να κατασευάσουμε την ορίζουσα Wronskian.

$$A = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}$$

$a_{11} = \sum_{s\in S} a_i X^{-t_{11}^s} - 1 = -1$

$a_{12} = \sum_{s\in S} a_i X^{-t_{12}^s} = X^{-2} + X^{-4}$, *γιατί από την κατάσταση 1 στην 2 πάμε μόνο με τελεία και παύλα.*

$a_{22} = \sum_{s\in S} a_i X^{-t_{22}^s} - 1 = X^{-2} + X^{-4} - 1$, *γιατί από την κατάσταση 2 μένουμε στην 2 με τελεία και παύλα.*

$a_{21} = \sum_{s\in S} a_i X^{-t_{21}^s} = X^{-3} + X^{-6}$, *γιατί από την κατάσταση 2 πάμε στην 1 με το κενό λέξης και γράμματος.*

Τότε ο πίνακας που σχηματίζεται είναι ο:

$$A = \begin{bmatrix} -1 & X^{-2} + X^{-4} \\ X^{-3} + X^{-6} & X^{-2} + X^{-4} - 1 \end{bmatrix}$$

με ορίζουσα την $X^{-2} + X^{-4} + X^{-5} + X^{-7} + X^{-8} + X^{-10} - 1 = 0$. *Σύμφωνα με το Shannon πού έλυσε την παραπάνω εξίσωση στην δημοσίευση του η χωρητικότητα είναι* $C = 0.539$.*Η παραπάνω εξίσωση λύνεται με αριθμητικές η γραφικές μεθόδους.*

Αφού λοιπόν είδαμε και ποια είναι η χωρητικότητα ενός καναλιού χωρίς θόρυβος ήρθε η ώρα να διατυπώσουμε το θεώρημα κωδικοποίηση του Shannon για κανάλια χωρίς θόρυβο. Το συγκεκριμένο θεώρημα είναι γνωστό και ως θεώρημα κωδικοποιήσης πηγής. Το ξανασυναντήσαμε στο κεφάλαιο 3 όταν χρησιμοποιώντας την απόσταση Kullback-Leibler αποδείξαμε ότι το όριο της συμπίεσης μίας τυχαία μεταβλητής είναι η εντροπία της. Ο λόγος που το επαναδιατυπώνουμε και θα το αποδείξουμε πάλι είναι διότι αυτή τη φορά θα χρησιμοποιήσουμε το θεώρημα ασυμπτωτικής ισοκατανομής για πηγές χωρίς μνήμη. Πριν διατυπώσουμε το θεώρημα θα εισάγουμε ένα μέγεθός που ονομάζεται **ρυθμός μετάδοσης κώδικα** (R). Στην ουσία το μέγεθος αυτό ποσοτικοποιεί τον πλήθος των bits ανά σύμβολο που πρέπει να μεταδίδουμε σε κάθε χρήση του καναλιού ώστε μετά από n χρήσεις του να έχουμε μεταδώσει μία συμβολοσειρά μήκους n της πηγής.

Θεώρημα 4.2. ***(Θεώρημα Κωδικοποίησης Καναλιού)***
Έστω μία τυχαία μεταβλητή X *που παίρνει τιμές στο πεπερασμένο σύνολο* $\mathcal{X}$ *και έχει εντροπία* $H(X)$. *Τότε υπάρχει ένας κώδικας* $\mathcal{C}$ *ο οποίος κωδικοποιεί τις συμβολοσειρές μήκους* n *της τυχαίας μεταβλητής με αμελητέα πιθανότητα λάθους* P_e *αν ο ρυθμός μετάδοσης κώδικα* $R \geq H$. *Αντίστροφα αν* $R < H$ *τότε η πιθανότητα λάθους* $P_e \to 1$

Απόδειξη[10]

Ευθεία Κατεύθυνση: *Έστω* $\epsilon > 0$

Το σύνολο όλων των ακολουθιών μήκους n *που παράγονται από την πηγή έχει πληθάριθμο* $|\mathcal{X}|^n$. *Επειδή από τις* $|\mathcal{X}|^n$ *σύμφωνα με το θεώρημα ασυμπτωτικής ισοκατανομής μόνο οι* $|A_\epsilon^{(n)}| = M$ *θα εμφανιστούν με μεγάλη πιθανότητα θα φτιάξουμε ένα κώδικα ώς εξής:*

1. *Διατάσσουμε όλε τις τυπικές ακολουθίες σύμφωνα με τις πιθανότητες τους. Σε κάθε ακολουθία αντιστοιχίζουμε ένα μοναδικό ακέραιο* $\mathcal{I} \in \{1, 2, \cdots, M\}$ *έτσι ώστε η αντιστοίχιση να είναι αμφιμονοσήμαντη.*

2. *Κωδικοποιούμε κάθε ακέραιο με περίπου* $\log M$, *bits.*

3. *Σφάλμα συμβαίνει αν κατά την κωδικοποίηση μας τύχει κάποια ακολουθία που δεν ανήκει στο τυπικό σύνολο, δηλαδή* $P_e = Pr[\mathbf{x_1^n} \in \{A_\epsilon^n\}^c]$. *Τότε ο αποκωδικοποιητής επιστρέφει τον δείκτη 1 από το σύνολο* $\mathcal{I}$.

Από το θεώρημα ασυμπτωτικής ισοκατανομής γνωρίζουμε ότι:

$$(1-\epsilon)2^{n(H(X)-\epsilon)} \leq M = |A_\epsilon^n| \leq 2^{n(H(x)+\epsilon)}$$
$$\frac{1}{n}\log(1-\epsilon) + H(X) - \epsilon \leq \frac{1}{n}\log M \leq H(X) + \epsilon \Rightarrow$$
$$\frac{1}{n}\log(1-\epsilon) + H(X) - \epsilon \leq R \leq H(X) + \epsilon$$

Επίσης πάλι από το θεώρημα ασυμπτωτικής ισοκατανομής $P_e = Pr[\mathbf{x_1^n} \in A_\epsilon^{nc}] < \epsilon$.
Άρα για $n \to \infty \Rightarrow \epsilon \to 0 \Rightarrow R \to H(X)$ *με* $P_e \to 0$

Αντίστροφη Κατεύθυνση *Έστω ότι πάμε να φτιάξουμε έναν κώδικα με μέσο μήκος μίκροτερο από την εντροπία της πηγής. Δηλαδή:*

[10] (**DMCYeung**), pp 103-105

$$\frac{1}{n}\log M = H(X) - z \Rightarrow M = 2^{n(H(X)-\zeta)}$$

Άρα από τις $|A_\epsilon^n| \approx 2^{n(H(X)+\epsilon)}$ *τυπικές ακολουθίες εμείς επιλέγουμε να κωδικοποιήσουμε τις* $M = 2^{n(H(X)-\zeta)} < |A_\epsilon^n|$*.Συμβολίζουμε το σύνολο των τυπικών ακολουθιών που επιλέξαμε να κωδικοποιήσουμε με* $\mathcal{M}$

Επειδή για κάθε ακολουθία που ανήκει στο τυπικό σύνολο ισχύει ότι $Pr[\mathbf{x_1^n} \in A_\epsilon^n] \leq \frac{1}{2^{n(H(X)-\epsilon)}}$ *η συνολική πιθανότητα που θα περιέχει το σύνολο τυπικών ακολουθιών που επιλέξαμε να κωδικοποιήσουμε θα είναι* $Pr[\mathbf{x_1^n} \in M] = \sum_{\mathbf{x_1^n}\in\mathcal{M}} Pr[\mathbf{X_1^n} = \mathbf{x_1^n}] \overset{\mathcal{M}\subset A_\epsilon^n}{\leq} \sum_{\mathbf{x_1^n}\in\mathcal{M}} \frac{1}{2^{n\cdot(H(X)+\epsilon)}} = 2^{n(H(X)-\zeta)} \cdot \frac{1}{2^{n(H(X)-\epsilon)}} = \frac{1}{2^{n(\zeta-e)}}$

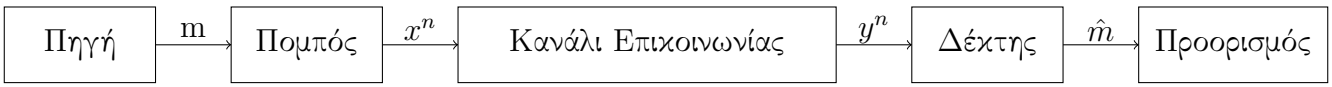
Άρα η πιθανότητα για να μην γίνει λάθος αυτή την φορά ορίζεται ως η πιθανότητα να ανήκει στο σύνολο M

$$1 - P_e = Pr[\mathbf{x_1^n} \in M] < Pr[\mathbf{x_1^n} \in M] + Pr[\mathbf{x_1^n} \notin A_\epsilon^n] < \frac{1}{2^{n(\zeta-\epsilon)}} + \epsilon \Rightarrow P_e > 1 - (\frac{1}{2^{n(\zeta-\epsilon)}} + \epsilon) \overset{n\to\infty}{\to} 1$$

Στην περίπτωση που θα θέλαμε το $P_e = 0$ καταλαβαίνουμε ότι θα έπρεπε να κωδικοποιήσουμε κάθε πιθανή ακολουθία που ανήκει στο $\mathcal{X}^n$.

4.4 Χωρητικότητα για κανάλια με θόρυβο

Όταν σε ένα κανάλι υπάρχει η παράμετρος του θορύβου, ποτέ δεν είμαστε σίγουροι ότι το μήνυμα που θα φτάσει στο δέκτη είναι το ίδιο με αυτό που στάλθηκε από τον πομπό. Σε αυτή την περίπτωση το μοντέλο επικοινωνίας διαμορφώνεται ως εξής:



Σχήμα 4.12: Το βασικό μοντέλο επικοινωνίας για ένα κανάλι με θόρυβο

Αν έχουμε ένα κανάλι που μεταδίδει ένα σύμβολο τη φορά, ο πιο λογικό τρόπος να το περιγράψουμε είναι μέσω των δεσμευμένων πιθανοτήτων $\{Pr[Y = y|X = x]\}$.

Ορισμός 4.5. *Ένα διακριτό κανάλι επικοινωνίας ορίζεται ως η τριάδα* $(\mathcal{X}, \{P[Y|X]\}, \mathcal{Y})$*, όπου* $\mathcal{X}$ *το πεπερασμένο σύνολο τιμών της τυχαίας μεταβλητής που μοντελοποιεί το αλφάβητο εισόδου,* $\mathcal{Y}$ *το πεπερασμένο σύνολο τιμών της τυχαίας μεταβλητής* Υ *που μοντελοποιεί το αλφάβητο εξόδου και* $\{P[Y = y|X = x]\}$ *το σύνολο συναρτήσεων μάζας πιθανότητας όπου* $\forall x \in \mathcal{X}, \forall y \in \mathcal{Y}$ *ισχύει* $\sum_{y\in\mathcal{Y}} Pr[Y = y|X = x] = 1$*.*

Επειδή όμως το μήνυμα m που μεταδίδεται από ένα κανάλι αποτελείται από μία ακολουθία συμβόλων εισόδου και κατ΄ επέκταση εξόδου υπάρχει η ανάγκη να επεκτείνουμε τον παραπάνω ορισμό καναλιού ώστε να συμπεριλάβουμε εισόδους και εξόδους μεγαλύτερες του 1.

Ορισμός 4.6. *Η επέκταση ενός διακριτού καναλιού επικοινωνίας που μεταδίδει μηνύματα μήκους* n*, ορίζεται ως η τριάδα* $(\mathcal{X}^n, Pr[\mathbf{Y^n} = \mathbf{y^n}|\mathbf{X^n} = \mathbf{x^n}], \mathcal{Y}^n)$*, όπου* $\mathcal{X}^n$ *το επεκταμένο πεπερασμένο σύνολο τιμών που μοντελοποιεί εισόδους μήκους* n *σύμβολα,* $\mathcal{Y}^n$ *το επεκταμένο πεπερασμένο σύνολο τιμών που μοντελοποιεί εξόδους μήκους* n *σύμβολα και* $Pr[\mathbf{Y^n} = \mathbf{y^n}|\mathbf{X^n} = \mathbf{x^n}]$*:*

$$Pr[\mathbf{Y^n} = \mathbf{y^n}|\mathbf{X^n} = \mathbf{x^n}] = Pr[Y_1 = y_1 \cdots, Y_n = y_n|X_1 = x_1, \cdots X_n = x_n]$$

και

$$\forall x^n \in \mathcal{X}^n,\ \forall y^n \in \mathcal{Y}^n,\ \sum_{y^n\in\mathcal{Y}^n} Pr[\mathbf{Y^n} = \mathbf{y^n}|\mathbf{X^n} = \mathbf{x^n}] = 1$$

- *Αν το σύμβολο εξόδου* y_k *δεν εξαρτάται από τα προηγούμενα* $k-1$ *σύμβολα εισόδου και εξόδου, δηλαδή*

$$Pr[Y_k = y_k|X_1 = x_1, \cdots X_k = x_k, Y_1 = y_1, \cdots Y_{k-1} = y_{k-1}] = Pr[Y_k = y_k|X_k = x_k]$$

τότε μιλάμε για ένα κανάλι χωρίς μνήμη(memoryless).

• *Αν το σύμβολο εισόδου x_k δεν εξαρτάται από τα προηγούμενα $k-1$ σύμβολα εξόδου, δηλαδή*

$$Pr[X_k = x_k|X_1 = x_1, \cdots X_{k-1} = x_{k-1}, Y_1 = y_1, \cdots Y_{k-1} = y_{k-1}] = Pr[X_k = x_k|X_1 = x_1, \cdots X_{k-1} = x_{k-1}]$$

, τότε μιλάμε για ένα κανάλι χωρίς σχόλια (Feedback)

4.5 Διακριτά Κανάλια χωρίς μνήμη και σχόλια

[11]

Σύμφωνα με του προηγούμενους ορισμούς αν το κανάλι επικοινωνίας δεν έχει μνήμη ξέρουμε ότι το y_i θα εξαρτάται μόνο από το παρών σύμβολο x_i που στέλνεται εκείνη τη στιγμή και είναι δεσμευμένα ανεξάρτητο από το μήνυμα m που επιλέχθηκε να κωδικοποιηθεί. Από την άλλη επειδή το κανάλι εκτός από αμνησία δεν περιέχει και σχόλια ξέρουμε ότι το x_n που θα σταλεί δεν θα εξαρτάται ούτε από τα προηγούμενα $y_1,\ y_2, \cdots, y_{n-1}$ παρά μόνο από το μήνυμα m που έχουμε επιλέξει να κωδικοποιήσουμε και να αποστείλουμε. Τέλος το ληφθέν μήνυμα $\hat{m}$ θα εξαρτάται μόνο από τα σύμβολα y_i που το συνθέτουν και κανένα σύμβολο προηγούμενου σταδίου Άρα η δεσμευμένη πιθανότητα μια ακολουθίας συμβόλων εξόδου όταν είναι γνωστή μία ακολουθία εισόδου δίνεται από τη σχέση.

$$Pr[\mathbf{Y^n} = \mathbf{y^n}|\mathbf{X^n} = \mathbf{x^n}] = \prod_{i=1}^{n} Pr[Y = y_i|X = x_i]$$

Επίσης όλα τα παραπάνω στάδια με τον τρόπο που περιγράφτηκαν και τις διαδικασίες τις οποίες αποτελούν βλέπουμε ότι σχηματίζουν μία μαρκοβιανή αλυσίδα της μορφής:

$$\mathcal{M} \to \mathcal{X}^n \to \mathcal{Y}^n \to \hat{\mathcal{M}}$$

Όπως και στην προηγούμενη ενότητα έτσι και σε αυτή αυτό που μας ενδιαφέρει είναι να βρούμε την χωρητικότητα του καναλιού, δηλαδή να βρούμε το μέγιστο αριθμό κωδικών συμβόλων που μπορούμε να μεταδώσουμε ανά χρήση του καναλιού. Η πρόταση "ανά χρήση του καναλιού" μπορεί να μεταφραστεί ως το πλήθος των συμβόλων που μεταδίδονται ανά μονάδα χρόνου ή το πλήθος των συμβόλων που μεταδίδονται ανά σύμβολο του μηνύματος m ή ανά μήνυμα m. Το τι θα επιλεγεί κάθε φορά να εννοείται "χρήση του καναλιού" εξαρτάται από την εφαρμογή την οποία μελετάμε. Πριν δώσουμε τον ορισμό για την χωρητικότητα σε κανάλια με θόρυβο θα φτάσουμε στη σχέση από την οποία περιγράφεται με μία απλή ανάλυση. Από την στιγμή που υπάρχει θόρυβος ξέρουμε ότι το σύμβολο που στέλνεται δεν είναι κατά ανάγκη ίδιο με αυτό που λαμβάνεται αλλά διέπεται από μία πιθανότητα $P[Y = y|X = x]$. Από την στιγμή που γεννάται αυτή η αβεβαιότητα στο δέκτη με βάση τα όσα έχουμε μάθει μέχρι στιγμής ξέρουμε ότι μπορεί να ποσοτικοποιηθεί μέσω της δεσμευμένης εντροπίας $H[X = x|Y = y]$. Με την ίδια λογική η αβεβαιότητα του πομπού μοντελοποιείται με την ποσότητα $H[Y = y|X = x]$. Αν ξέρουμε την εντροπία του πομπού και του δέκτη $H(X)$ και $H(Y)$, δηλαδή την ποσότητα πληροφορίας που παράγουν, τότε μπορούμε να καταλάβουμε ότι η ποσότητα της πληροφορίας που μεταδίδεται είναι η $I(X = x; Y = y) = H(Y = y) - H(Y = y|X = x) = H(X = x) - H(X = x|Y = y)$.

Η τελευταία ποσότητα που αποτελεί την αμοιβαία πληροφορία, μπορεί να ερμηνευθεί από την πλευρά του δέκτη $(H(X = x) - H(X = x|Y = y))$ ως η αβεβαιότητα της λήψης του συμβόλου x πριν λάβουμε το y μειωμένη από την αβεβαιότητα που απομένει όταν λάβαμε το σύμβολο y. Αντίστοιχα από τη μεριά του πομπού $(H(Y = y) - H(Y = y|X = x))$ η παραπάνω σχέση μπορεί να εκφρασθεί ως η αβεβαιότητα να ληφθεί το σύμβολο y μειωμένη από την αβεβαιότητα να ληφθεί το σύμβολο y όταν ξέρουμε ότι το σύμβολο που στάλθηκε είναι το x. Από τις ιδιότητες της αμοιβαίας πληροφορίας βλέπουμε ότι η αβεβαιότητα που υπάρχει στον πομπό

[11] Η αγγλική ορολογία είναι Discrete Memoryless channel (DMC) without feedback

είναι ίδια με αυτή του δέκτη. Θυμόμαστε ότι στο πρώτο κεφάλαιο είχαμε αποδείξει πως το I(X;Υ) είναι μία κοίλη συνάρτηση ως προς το $Pr[*]$. Καταλαβαίνουμε λοιπόν ότι κάθε φορά που θα αλλάζει η κατανομή της πηγής θα μεταβάλλεται και το $I(X,Y)$, δηλαδή το ποσό της πληροφορίας που μεταδίδεται μέσα από το κανάλι. Οπότε είναι λογικό να προσπαθήσουμε να βρούμε μία κατανομή που θα μεγιστοποιεί την αμοιβαία πληροφορία. Μία τέτοια κατανομή ξέρουμε ότι υπάρχει καθώς η συνάρτηση είναι κοίλη και είναι ορισμένη σε ένα κλειστό και φραγμένο διάστημα. Άρα η χωρητικότητα του καναλιού ορίζεται παρακάτω:

Ορισμός 4.7. *Η* ***χωρητικότητα*** *ενός καναλιού χωρίς μνήμη και σχόλια είναι:*

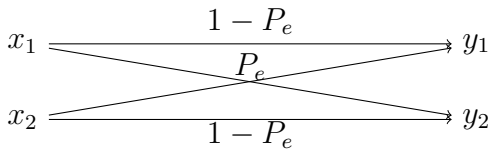
$$C = \max_{P_X(x)} I(X;Y) \tag{4.7}$$

Παραδείγματα καναλιών χωρίς μνήμη και σχόλια

Παράδειγμα 4.2. ***(Το δυαδικό συμμετρικό κανάλι)***

Το δυαδικό συμμετρικό κανάλι μεταδίδει κάθε bit λάθος με πιθανότητα P_e και σωστά με πιθανότητα $1-P_e$. Ο πίνακας που περιγράφει το δυαδικό συμμετρικό κανάλι είναι ο:

$$A = \begin{bmatrix} 1-P_e & P_e \\ P_e & 1-P_e \end{bmatrix}$$



$$\begin{aligned} Pr[Y=y_1|X=x_1] &= 1-P_e \\ Pr[Y=y_1|X=x_2] &= P_e \\ Pr[Y=y_2|X=x_1] &= P_e \\ Pr[Y=y_2|X=x_2] &= 1-P_e \end{aligned}$$

$$I(X;Y) = H(Y) - H(Y|X) = H(Y) - \sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} Pr[Y=y_i|X=xi]Pr[X=x_i]log\frac{1}{Pr[Y=y_i|X=x_i]} =$$

$$H(Y) - \sum_{x\in\mathcal{X}} Pr[X=x_i]\cdot\left(\sum_{y\in\mathcal{Y}} Pr[Y=y_i|X=xi]Pr[X=x_i]log\frac{1}{Pr[Y=y_i|X=x_i]}\right) =$$

$$H(Y) - \sum_{x\in\mathcal{X}} Pr[X=x_i]\cdot H(P_e,1-P_e) = H(Y) - H(P_e,1-P_e) \Rightarrow$$

$$I(X,Y) = H(Y) - H(P_e,1-P_e)$$

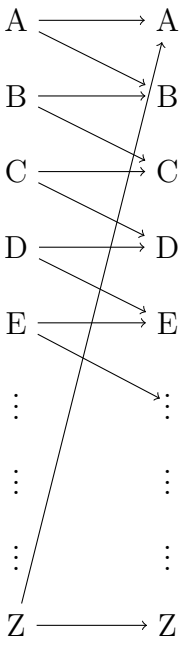
Προφανώς για να μεγιστοποιήσουμε αυτή τη συνάρτηση θα πρέπει να μεγιστοποιήσουμε την $H(Y)$. Η $H(Y)$ μεγιστοποιείται για ισοπίθανα ενδεχόμενα, άρα :

$$C = \max_{P_X(x)} I(X,Y)\} = \max_{P_X(x)}\{H(X) - H(X|Y)\} = \max_{P_Y(y)}\{H(Y) - H(Y|X)\} = 1 - H(P_e,1-P_e)$$

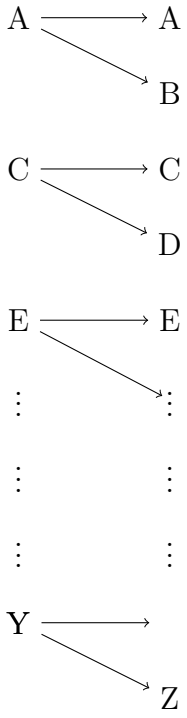
Η χωρητικότητα το δυαδικού συμμετρικού καναλιού παίρνει τη ελάχιστη τιμή της 0 όταν $H(P_e,1-P_e) = 1 \Leftrightarrow P_e = \frac{1}{2}$ ενώ παίρνει τη μεγαλύτερη 1 όταν $H(P_e,1-P_e) = 0 \Leftrightarrow P_e = 0$.

Παράδειγμα 4.3. ***(Γραφομηχανή με θόρυβο)***

Έστω ότι έχουμε μία πηγή που παράγει 26 γράμματα. Τότε το κανάλι κάθε φορά μεταδίδει την είσοδο σωστά με πιθανότητα $\frac{1}{2}$ *ή μεταδίδει το επόμενο γράμμα πάλι με πιθανότητα* $\frac{1}{2}$



(α′) Η αναπαράσταση μιας γραφομηχανής με θόρυβο



(β′) Η τροποποιημένη γραφομηχανή ώστε να είναι αθόρυβη

Αφού το κάθε σύμβολο μεταδίδεται σωστά με πιθανότητα $\frac{1}{2}$ *τότε :*

$$H[X|Y] = \sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} Pr[Y=y]\cdot Pr[X=x|Y=y]\cdot log\frac{1}{Pr[X=x|Y=y]} =$$

$$\sum_{y\in\mathcal{Y}} Pr[Y=y]\sum_{x\in\mathcal{X}} Pr[X=x|Y=y]\cdot log\frac{1}{Pr[X=x|Y=y]} \overset{(a)}{=} \sum_{y\in\mathcal{Y}} Pr[Y=y]\cdot H(\frac{1}{2},\ \frac{1}{2}) =$$

$$\sum_{y\in\mathcal{Y}} Pr[Y=y] = 1$$

Η ισότητα (a) ισχύει γιατί για οποιοδήποτε ληφθέν σύμβολο η πιθανότητα να έχει έρθει από άλλο γράμμα είναι $\frac{1}{2}$

$$I(X,Y) = H(X) - H(X|Y) = H(X) - 1 \Rightarrow C = \max_{P_X(x)}(H(X)-1) = log_2 6 - 1 = log\frac{26}{2} = log 13$$

Η χωρητικότητα του καναλιού, μας υποδεικνύει πως αν θέλουμε να μεταδώσουμε χωρίς σφάλματα θα πρέπει να αρκεστούμε στην μετάδοση 13 συμβόλων και όχι 26. Με αυτό τον τρόπο το κάθε σύμβολο εισόδου αντιστοιχίζεται σε ένα σύνολο συμβόλων εξόδου, το οποίο είναι ξένο ως προς κάθε άλλο σύνολο συμβόλων εξόδου που προέρχεται από ένα διαφορετικό σύμβολο εισόδου. Αυτή η λογική είναι πάρα πολύ σημαντική και θα χρησιμοποιηθεί και κατά την απόδειξη του θεωρήματος κωδικοποίησης καναλιού.

Συνεχίζοντας την μελέτη μας θα διερευνήσουμε λίγο βαθύτερα τη σχέση που υπάρχει ανάμεσα στις ακολουθίες εισόδου και εξόδου από το κανάλι. Θυμόμαστε στο κεφάλαιο 2, που συζητούσαμε για την εντροπία

των μαρκοβιανών πηγών πληροφορίας ότι είχαμε καταλήξει στο θεώρημα της ασυμπτωτικής ισοκατανομής. Το συγκεκριμένο θεώρημα έλεγε ότι αν αφήσουμε το n να γίνει πολύ μεγάλο, δηλαδή αν μιλήσουμε για ακολουθίες μεγάλους μήκους, τότε οι ακολουθίες αυτές ανήκουν σε ένα τυπικό σύνολο $A_\epsilon^{(n)}$ το οποίο συγκεντρώνει την περισσότερη πιθανότητα. Αυτό πρακτικά σήμαινε ότι οι ακολουθίες μεγέθους n που σχεδόν βεβαίως θα παράξει η πηγή ανήκουν σε αυτό το σύνολο και μάλιστα το να επιλέγει κάποιο από τα στοιχεία του συνόλου είχε πιθανότητα περίπου $\frac{1}{2^{n\cdot H(X)}}$. Τώρα όμως δεν έχουμε μόνο μία πηγή που παράγει ακολουθίες $x_1,\ x_2, \cdots, x_n$, υπάρχει και μία πηγή στην άλλη μεριά του καναλιού, ο αποκωδικοποιητής, που παράγει ακολουθίες $y_1,\ y_2, \cdots, y_n$. Είναι λογικό λοιπόν να θέλουμε να εξετάσουμε σε αντιστοιχία με το θεώρημα της ασυμπτωτικής ισοκατανομής, αν υπάρχει ένα αντίστοιχο τυπικό σύνολο που περιέχει τα ζεύγη $(x^n,\ y^n)$ που θα παραχθούν με μεγάλη πιθανότητα. Η απάντηση είναι θετική και το σύνολο περιγράφεται στον παρακάτω ορισμό.

Ορισμός 4.8. *Το τυπικό σύνολο $J_\epsilon^{(n)}$ των από κοινού ακολουθιών $\{(x^n,\ y^n)\}$ μήκους n που ακολουθούν τον κανόνα $Pr(X = x,\ Y = y)$ και οι δειγματικές εντροπίες $\frac{1}{n}logPr[\mathbf{X} = \mathbf{x^n}],\ \frac{1}{n}logPr[\mathbf{Y} = \mathbf{y^n}]$ συγκλίνουν κατά ϵ στις θεωρητικές, ορίζεται ως εξής:*

$$\begin{aligned} J_\epsilon^{(n)} = \{&(x^n,\ y^n) \in \mathcal{X}^n \times \mathcal{Y}^n : \\ &|\frac{1}{n}log\frac{1}{Pr[\mathbf{X} = \mathbf{x^n}]} - H(X)| < \epsilon, \\ &|\frac{1}{n}log\frac{1}{Pr[\mathbf{Y} = \mathbf{y^n}]} - H(Y)| < \epsilon, \\ &|\frac{1}{n}log\frac{1}{Pr[\mathbf{X} = \mathbf{x^n},\ \mathbf{Y} = \mathbf{y^n}]} - H(X,Y)| < \epsilon\} \end{aligned} \tag{4.8}$$

Το επόμενο βήμα είναι να διατυπώσουμε το θεώρημα ασυμπτωτικής ισοκατανομής για από κοινού ακολουθίες. Με αυτόν το τρόπο θα βρούμε το πλήθος του συνόλου που περιέχει τα ζεύγη $(x^n,\ y^n)$ τα οποία θα συμβούν με μεγάλη πιθανότητα καθώς και την πιθανότητα με την οποία συμβαίνει το κάθε ένα.

Θεώρημα 4.3. ***(Θεώρημα Ασυμπτωτική Ισοκατανομής για από κοινού ακολουθίες)*** *Έστω $(X^n,\ Y^n)$ ζεύγη ακολουθιών των τυχαίων μεταβλητών X και Y μήκους n, οι οποίες είναι ανεξάρτητες, ισόνομες και ακολουθούν τον κανόνα $Pr[\mathbf{X} = \mathbf{x^n},\ \mathbf{Y} = \mathbf{y^n}] = \prod_{i=1}^n Pr[X = x_i, Y = y_i]$. Τότε*

1. $\lim_{n\to\infty} Pr[(\mathbf{X} = \mathbf{x^n},\ \mathbf{Y} = \mathbf{y^n}) \in J_\epsilon^{(n)}] = 1$
2. $|J_\epsilon^{(n)}| \leq 2^{n\cdot H(X,Y)+\epsilon}$
3. *Αν $X^n \perp\!\!\!\perp Y^n$ με $X^n \sim Pr[\mathbf{X} = \mathbf{x^n}]$ και $Y^n \sim Pr[\mathbf{Y} = \mathbf{y^n}]$ τότε:*

$$Pr[\mathbf{X} = \mathbf{x^n},\ \mathbf{Y} = \mathbf{y^n} \in J_\epsilon^{(n)}] \leq 2^{-n\cdot(I(X,Y)-3\cdot\epsilon)}. \tag{4.9}$$

Επίσης για μεγάλο n

$$Pr[\mathbf{X} = \mathbf{x^n},\ \mathbf{Y} = \mathbf{y^n} \in J_\epsilon^{(n)}] \geq (1-\epsilon)\cdot 2^{-n\cdot(I(X,Y)+3\cdot\epsilon)}. \tag{4.10}$$

Απόδειξη

1. *Από την στιγμή που το ζεύγος $(X = x^n,\ Y = y^n)$ θέλουμε να ανήκει στο σύνολο τυπικών από κοινού ακολουθιών τότε θα υπάρχει:*

   (α΄) ένα n_1 για το οποίο θα ισχύει $Pr[|\frac{1}{n}log\frac{1}{Pr[|\mathbf{X} = \mathbf{x^n}]} - H(X)| \geq \epsilon] < \frac{\epsilon}{3}\ \forall n > n_1$

(β′) ένα n_2 για το οποίο θα ισχύει $Pr[|\frac{1}{n}log\frac{1}{Pr[|\mathbf{Y}=\mathbf{y^n}]}-H(Y)|\geq\epsilon]<\frac{\epsilon}{3}\ \forall n>n_2$

(γ′) ένα n_3 για το οποίο θα ισχύειι $Pr[|\frac{1}{n}log\frac{1}{Pr[|\mathbf{X}=\mathbf{x^n},\ \mathbf{Y}=\mathbf{y^n}]}-H(X,Y)|\geq\epsilon|]<\frac{\epsilon}{3}\ \forall n>n_3$

Τότε για $n=\max\{n_1,\ n_2,\ n_3\}$ *θα συμβαίνουν και τα τρία μαζί, δηλαδή*

$Pr[(X^n,\ Y^n)\notin J_\epsilon^{(n)}]=Pr[\{|\frac{1}{n}log\frac{1}{Pr[|\mathbf{X}=\mathbf{x^n}]}-H(X)|\geq\epsilon|\}\cup Pr[|\frac{1}{n}log\frac{1}{Pr[|\mathbf{Y}=\mathbf{y^n}]}-H(Y)|\geq\epsilon]\cup\{|\frac{1}{n}log\frac{1}{Pr[|\mathbf{X}=\mathbf{x^n},\ \mathbf{Y}=\mathbf{y^n}]}-H(X,Y)|\geq\epsilon|\}]\leq Pr[|\frac{1}{n}lo\frac{1}{gPr[|\mathbf{X}=\mathbf{x^n}]}-H(X)|\geq\epsilon|]+Pr[|\frac{1}{n}log\frac{1}{Pr[|\mathbf{Y}=\mathbf{y^n}]}-H(Y)|\geq\epsilon]+Pr[|\frac{1}{n}log\frac{1}{Pr[|\mathbf{X}=\mathbf{x^n},\ \mathbf{Y}=\mathbf{y^n}]}-H(X,Y)|\geq\epsilon|]<\frac{\epsilon}{3}+\frac{\epsilon}{3}+\frac{\epsilon}{3}=\epsilon\Rightarrow Pr[(X^n,\ Y^n)\notin J_\epsilon^{(n)}]<\epsilon\Rightarrow Pr[(X^n,\ Y^n)\in J_\epsilon^{(n)}]>1-\epsilon$

Τότε για μεγάλα n θα ισχύει ότι $\epsilon\to0\Rightarrow Pr[(X^n,\ Y^n)\in J_\epsilon^{(n)}]\to1$

2. $1=\sum_{(x^n,\ y^n)\in\mathcal{X}^n\times\mathcal{Y}^n}Pr[(\mathbf{X}=\mathbf{x^n},\ \mathbf{Y}=\mathbf{y^n})]\leq\sum_{J_\epsilon^{(n)}}Pr[(\mathbf{X}=\mathbf{x^n},\ \mathbf{Y}=\mathbf{y^n})]$

   Όμως για κάθε ζεύγος $(x^n,\ y^n)$ *που ανήκει στο τυπικό σύνολο των από κοινού ακολουθιών* $J_\epsilon^{(n)}$ *ισχύει:*

$$
\begin{aligned}
&|\frac{1}{n}log\frac{1}{Pr[|\mathbf{X}=\mathbf{x^n},\ \mathbf{Y}=\mathbf{y^n}]}-H(X,Y)|<\epsilon\Rightarrow\\
&H(X,Y)-\epsilon<\frac{1}{n}log\frac{1}{Pr[|\mathbf{X}=\mathbf{x^n},\ \mathbf{Y}=\mathbf{y^n}]}<H(X,Y)+\epsilon\Rightarrow\\
&n\cdot(H(X,Y)-\epsilon)<log\frac{1}{Pr[|\mathbf{X}=\mathbf{x^n},\ \mathbf{Y}=\mathbf{y^n}]}<n\cdot(H(X,Y)+\epsilon)\Rightarrow\\
&2^{n\cdot(H(X,Y)-\epsilon)}<\frac{1}{Pr[|\mathbf{X}=\mathbf{x^n},\ \mathbf{Y}=\mathbf{y^n}]}<2^{n\cdot(H(X,Y)+\epsilon)}\Rightarrow\\
&2^{-n\cdot(H(X,Y)+\epsilon)}<Pr[|\mathbf{X}=\mathbf{x^n},\ \mathbf{Y}=\mathbf{y^n}]<2^{-n\cdot(H(X,Y)-\epsilon)}
\end{aligned}
$$

   Οπότε με βάση την τελευταία ανισοτική σχέση το άθροισμα γίνεται:

$$
\begin{aligned}
&1=\sum_{(x^n,\ y^n)\in\mathcal{X}^n\times\mathcal{Y}^n}Pr[(\mathbf{X}=\mathbf{x^n},\ \mathbf{Y}=\mathbf{y^n})]\geq\sum_{J_\epsilon^{(n)}}Pr[(\mathbf{X}=\mathbf{x^n},\ \mathbf{Y}=\mathbf{y^n})]\geq\sum_{J_\epsilon^{(n)}}2^{-n\cdot(H(X,Y)+\epsilon)}\Rightarrow\\
&1\geq|J_\epsilon^{(n)}|\cdot2^{-n\cdot(H(X,Y)+\epsilon)}\Rightarrow|J_\epsilon^{(n)}|\leq2^{n\cdot(H(X,Y)+\epsilon)}
\end{aligned}
$$

   Αντίστοιχα μπορούμε να εξάγουμε ένα κάτω φράγμα για το πληθάριθμο του συνόλου των από κοινού ακολουθιών:

$$
\begin{aligned}
&1-\epsilon<Pr[(\mathbf{X}=\mathbf{x^n},\ \mathbf{Y}=\mathbf{y^n})\in J_\epsilon^{(n)}]=\sum_{J_\epsilon^{(n)}}Pr[(\mathbf{X}=\mathbf{x^n},\ \mathbf{Y}=\mathbf{y^n})]\leq\sum_{J_\epsilon^{(n)}}2^{-n\cdot(H(X,Y)-\epsilon)}\Rightarrow\\
&1-\epsilon\leq|J_\epsilon^{(n)}|\cdot2^{-n\cdot(H(X,Y)-\epsilon)}\Rightarrow|J_\epsilon^{(n)}|\geq(1-\epsilon)\cdot2^{n\cdot(H(X,Y)-\epsilon)}
\end{aligned}
$$

3. *Από την στιγμή που έχουμε υποθέσει πώς το X^n είναι ανεξάρτητο από το Y^n έπεται ότι* $Pr[\mathbf{X}=\mathbf{x^n},\ \mathbf{Y}=\mathbf{y^n}]=Pr[\mathbf{X}=\mathbf{x^n}]\cdot Pr[\mathbf{Y}=\mathbf{y^n}]$. *Ακόμη από την στιγμή που ένα ζεύγος* $(x^n,\ y^n)$ *ανήκει στο* $J_\epsilon^{(n)}$ *σύμβαινουν τρία πράγματα ταυτόχρονα:*

   (α′) $|\frac{1}{n}log\frac{1}{Pr[\mathbf{X}=\mathbf{x^n}]}-H(X)|<\epsilon\Rightarrow2^{-n\cdot(H(X)+\epsilon)}<Pr[\mathbf{X}=\mathbf{x^n}]<2^{-n\cdot(H(X)-\epsilon)}$

(β′) $|\frac{1}{n}log\frac{1}{Pr[\mathbf{Y}=\mathbf{y^n}]} - H(Y)| < \epsilon \Rightarrow 2^{-n\cdot(H(Y)+\epsilon)} < Pr[\mathbf{Y}=\mathbf{y^n}] < 2^{-n\cdot(H(Y)-\epsilon)}$

(γ′) $|\frac{1}{n}log\frac{1}{Pr[\mathbf{X}=\mathbf{x^n},\ \mathbf{Y}=\mathbf{y^n}]} - H(X,Y)| < \epsilon \Rightarrow 2^{-n\cdot(H(X,Y)+\epsilon)} < Pr[|\mathbf{X}=\mathbf{x^n},\ \mathbf{Y}=\mathbf{y^n}] < 2^{-n\cdot(H(X,Y)-\epsilon)}$

Άρα:

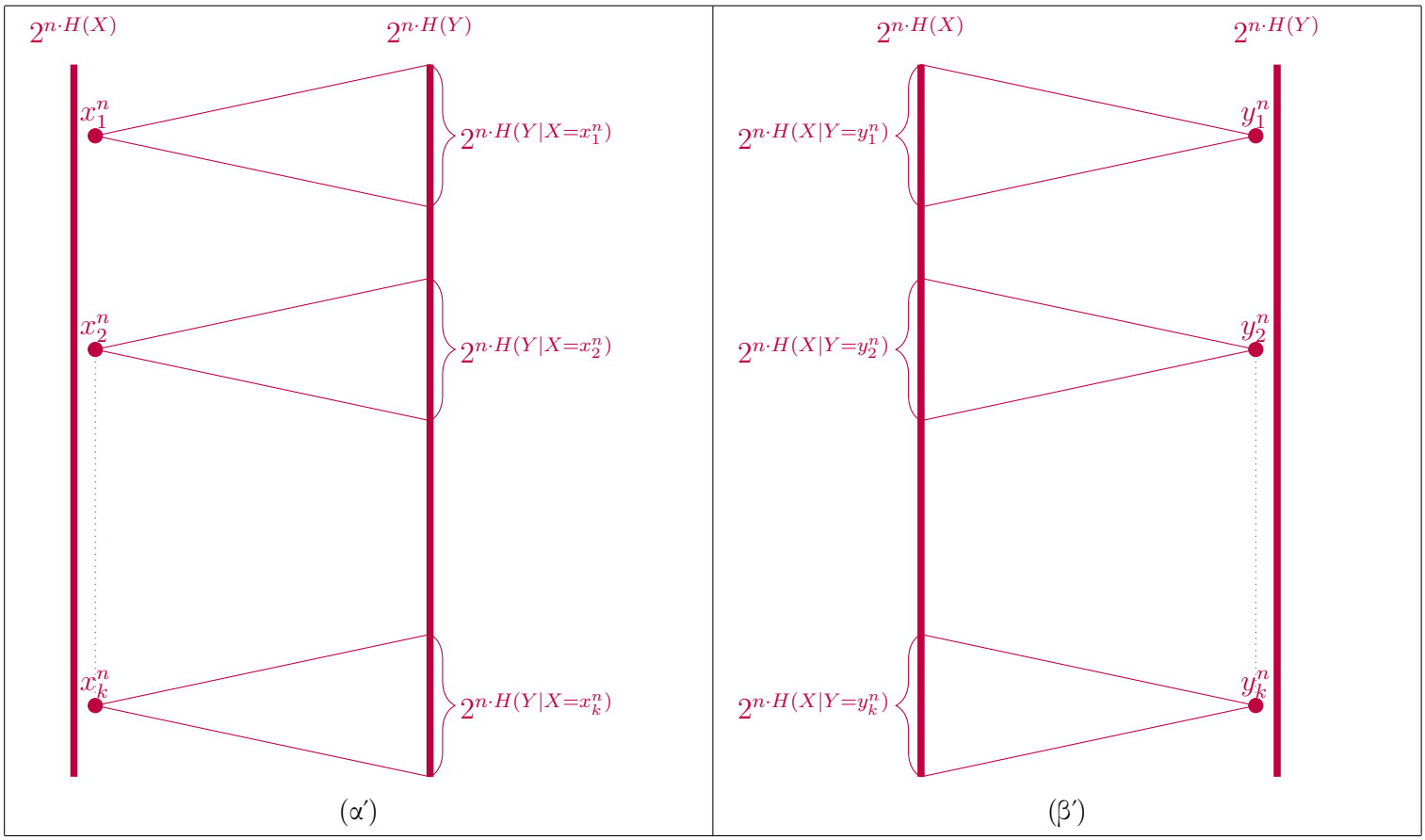
$$Pr[(\mathbf{X}=\mathbf{x^n},\ \mathbf{Y}=\mathbf{y^n}) \in J_\epsilon^{(n)}] = \sum_{J_\epsilon^{(n)}} Pr[\mathbf{X}=\mathbf{x^n},\ \mathbf{Y}=\mathbf{y^n}] = \sum_{J_\epsilon^{(n)}} Pr[\mathbf{X}=\mathbf{x^n}]\cdot Pr[\mathbf{Y}=\mathbf{y^n}] \leq$$
$$\sum_{J_\epsilon^{(n)}} 2^{-n\cdot(H(X)-\epsilon)}\cdot 2^{-n\cdot(H(Y)-\epsilon)} = 2^{-n\cdot(H(X)-\epsilon)}\cdot 2^{-n\cdot(H(Y)-\epsilon)}\cdot \sum_{J_\epsilon^{(n)}} 1 \leq$$
$$2^{-n\cdot(H(X)-\epsilon)}\cdot 2^{-n\cdot(H(Y)-\epsilon)}\cdot 2^{n\cdot(H(X,Y)+\epsilon)} = 2^{n\cdot(H(X,Y)-H(X)-H(Y)+3\cdot n\cdot\epsilon} = 2^{-n\cdot(I(X,Y)-3\dagger\epsilon)}$$

Ακόμη ένα άνω φράγμα για την $Pr[(\mathbf{X}=\mathbf{x^n},\ \mathbf{Y}=\mathbf{y^n}) \in J_\epsilon^{(n)}]$ *μπορεί να εξαχθεί με το ίδιο τρόπο:*

$$Pr[(\mathbf{X}=\mathbf{x^n},\ \mathbf{Y}=\mathbf{y^n}) \in J_\epsilon^{(n)} = \sum_{J_\epsilon^{(n)}} Pr[\mathbf{X}=\mathbf{x^n},\ \mathbf{Y}=\mathbf{y^n}] = \sum_{J_\epsilon^{(n)}} Pr[\mathbf{X}=\mathbf{x^n}]\cdot Pr[\mathbf{Y}=\mathbf{y^n}] \geq$$
$$\sum_{J_\epsilon^{(n)}} 2^{-n\cdot(H(X)+\epsilon)}\cdot 2^{-n\cdot(H(Y)+\epsilon)} = 2^{-n\cdot(H(X)+\epsilon)}\cdot 2^{-n\cdot(H(Y)+\epsilon)}\cdot \sum_{J_\epsilon^{(n)}} 1 \geq$$
$$2^{-n\cdot(H(X)+\epsilon)}\cdot 2^{-n\cdot(H(Y)+\epsilon)}\cdot(1-\epsilon)2^{n\cdot(H(X,Y)-\epsilon)} = 2^{n\cdot(H(X,Y)-H(X)-H(Y)-3\cdot n\cdot\epsilon} = (1-\epsilon)\cdot 2^{-n\cdot(I(X,Y)+3\cdot\epsilon)} \Rightarrow$$
$$Pr[(\mathbf{X}=\mathbf{x^n},\ \mathbf{Y}=\mathbf{y^n}) \in J_\epsilon^{(n)}] \geq (1-\epsilon)\cdot 2^{-n\cdot(I(X,Y)+3\cdot\epsilon)}$$

Το θεώρημα της ασυμπτωτικής ισοκατανομής λέει κάτι πολύ σπουδαίο που θα χρησιμοποιηθεί για την απόδειξη του θεωρήματος κωδικοποίησης για κανάλια. Αν σκεφτούμε λίγο προσεκτικά θα δούμε ότι ο κωδικοποιητής που περιγράφεται από την μεταβλητή X παράγει περίπου $2^{n\cdot H(X)}$ μηνύματα. Από την άλλη μεριά του καναλιού ο αποκωδικοποιητής παράγει περίπου $2^{n\cdot H(Y)}$ μηνύματα. Το πλήθος όμως των μηνυμάτων που παράγονται σε ζεύγη $(x^n,\ y^n)$ είναι το πολύ $2^{n\cdot H(X,Y)}$. Αυτό μας λέει ότι δεν αποτελούν όλα τα ζεύγη εισόδων-εξόδων τυπικές από κοινού ακολουθίες. Πρακτικά η παραπάνω διαπίστωση σημαίνει ότι κάθε είσοδος κάτω από την επίδραση του θορύβου μπορεί να οδηγήσει σε ένα σύνολο ακολουθιών εξόδου που είναι τυπικά συνδεδεμένες με εκείνη και προφανώς είναι πολύ λιγότερες από το $2^{n\cdot H(Y)}$.

Αν θέλουμε μπορούμε να σκεφτούμε την ίδια ιδέα με τους όρους των δεσμευμένων εντροπιών. Σε κάθε είσοδο $x \in A_\epsilon^{(n)}$ που παράγεται αντιστοιχούν περίπου $2^{n\cdot H(Y|X=x)}$ έξοδοι. Η πιθανότητα να επιλεγεί κάποια από τις $2^{n\cdot H(Y|X=x)}$ εξόδους που συνδέονται τυπικά με το x είναι $\frac{2^{n\cdot H(Y|X=x)}}{2^{n\cdot H(X=x,Y)}} = 2^{-n\cdot I(X=x,Y)}$ Από την άλλη κάθε έξοδος $y \in B_\epsilon^{(n)}$ αντιστοιχεί σε $2^{n\cdot H(X|Y=y)}$ εισόδους. Στο παρακάτω σχήμα φαίνεται ξεκάθαρα αυτό το οποίο συμβαίνει με τα σύνολα των εισόδων και εξόδων,



Σχήμα 4.14

Η παραπάνω εικόνα αποτελεί την καρδιά του θεωρήματος κωδικοποίησης για κανάλια με θόρυβο και είναι ίδια με το σχήμα που συναντήσαμε όταν υπολογίζαμε την χωρητικότητα της θορυβώδους γραφομηχανής. Παρατηρούμε ότι κάθε ακολουθία εισόδου x_i^n αντιστοιχίζεται σε ένα σύνολο ακολουθιών εξόδου που είναι ξένο ως προς τα άλλα που αντιστοιχούν σε διαφορετικές ακολουθίες εισόδου όπως φαίνεται στο Σχήμα 4.14(α′). Το αντίστοιχο συμβαίνει και με τα σύμβολα εξόδου όπως φαίνεται στο Σχήμα 4.14(β′). Καταλαβαίνουμε πως όταν διαμορφώνεται μία τέτοια αντιστοιχία μεταξύ των ακολουθιών εισόδου και εξόδου τότε η επικοινωνία είναι αξιόπιστη. Το ερώτημα είναι αν μπορεί πάντα να διαμορφωθεί μία τέτοια κατάσταση. Η απάντηση προφανώς είναι όχι. Το ζήτημα είναι να βρούμε πότε μπορεί να διαμορφωθούν τέτοιες αντιστοιχίες ή ακόμη καλύτερα πόσα x_i^n μπορούν να μεταδοθούν έτσι ώστε να αντιστοιχίζονται σε ξένα σύνολα ακολουθιών εξόδου y_i^n. Το πλήθος των ακολουθιών εξόδου συνολικά είναι $2^{n\cdot H(Y)}$. Το πλήθος των ακολουθιών εξόδου που αντιστοιχούν σε μία συγκεκριμένη είσοδο x_i^n είναι $2^{n\cdot H[Y|X]}$. Άρα το πλήθος των διαφορετικών συνόλων που αντιστοιχίζονται οι ακολουθίες εισόδου x_i^n είναι: $2^{n\cdot(H(Y)-H(Y|X)} = 2^{n\cdot I(X,Y)}$. Πλέον μπορούμε να καταλάβουμε τη σχέση του ορισμού χωρητικότητας ως την μέγιστη αμοιβαία πληροφορία της αξιόπιστης επικοινωνίας που επιτυγχάνεται μόνο όταν μεταδίδουμε το πολύ $2^{n\cdot I(X,Y)}$ εισόδους. Η χωρητικότητα του καναλιού επί της ουσίας μας λέει το μέγιστο αριθμό εισόδων που μπορούμε να μεταδώσουμε αφήνοντας ένα μικρό περιθώριο λάθους για να μην υπάρξει σύγχυση κατά την αποκωδικοποίηση.

Κάτω από την επίδραση του θορύβου η ακολουθία x_i^n δεν αντιστοιχίζεται σε ένα μοναδικό y_i^n αλλά σε ένα σύνολο από ακολουθίες εξόδου που αποτελούν τις πιθανότερες εξόδους που αντιστοιχούν στο x_i^n δεδομένου ότι κάποια σύμβολα του μπορεί να έχουν επηρεαστεί από το θόρυβο. Την σύγχυση που ενδέχεται να προκληθεί κατά την αποκωδικοποίηση μπορούμε να τη καταλάβουμε αν σκεφτούμε προς στιγμήν ότι δύο από τα $2^{nI(X;Y)}$ σύνολα πλήθους $2^{n\cdot H(Y|X=x_i^n)}$ δεν είναι ξένα μεταξύ τους. Αυτόματα θα υπάρχει μία ακολουθία y_i^n που θα αντιστοιχεί στην είσοδο x_i^n και σε κάποια άλλη x_j^n. Τότε προφανώς η αποκωδικοποίηση δεν θα είναι μοναδική και θα πέσουμε σε λάθος.

Το γεγονός ότι χρειαζόμαστε $2^{n\cdot I(X,Y)}$ ακολουθίες είτε εισόδου είτε εξόδου για να έχουμε αξιόπιστη

επικοινωνία μπορούσαμε να το καταλάβουμε αν αντί για το Σχήμα 4.8(α′) αναλύαμε το Σχήμα 4.8(β′). Στο (β′) έχουμε συνολικά $2^{n\cdot H(X)}$ ακολουθίες εξόδου που είναι οργανωμένες σε σύνολα των $2^{n\cdot H(X|Y=y_i^n)}$ ακολουθιών. Πάλι το πλήθος των εξόδων θα είναι $2^{n\cdot(H(X)-H(X|Y)} = 2^{n\cdot I(X,Y)}$. Άρα όταν έχουμε $2^{n\cdot I(X,Y)}$ ακολουθίες είτε εισόδου είτε εξόδου, αυτές οργανώνονται με τέτοιο τρόπο σε σύνολα $2^{n\cdot H(X|Y=y_i^n)}$ και $2^{n\cdot H(Y|X=x_i^n)}$ αντίστοιχα ώστε να μην υπάρχει σύγχυση κατά την κωδικοποίηση και αποκωδικοποίηση ενός μηνύματος m

Σε όλη τη διάρκεια της ανάλυσης μιλάμε για κωδικοποίηση και αποκωδικοποίηση. Πως ορίζεται όμως ο κώδικας των μηνυμάτων της πηγής και πως οι διαδικασίες κωδικοποίησης και αποκωδικοποίησης. Την απάντηση στα παραπάνω ερωτήματα μας τις δίνουν οι ορισμοί που ακολουθούν.

Ορισμός 4.9. *Ως κώδικα $(\mathcal{M},\ n)$ για ένα κανάλι $(X, Pr[Y = y|X = x], Y)$ χωρίς μνήμη και σχόλια ορίζουμε:*

1. *Ένα σύνολο δεικτών $\{1,\ 2, \cdots, |\mathcal{M}|\}$*
2. *Μία ντετερμινιστική συνάρτηση κωδικοποίησης $f : \mathcal{M} \to \mathcal{X}^n$ με $f(i) = x_1(i)x_2(i)\cdots x_n(i)$. Το σύνολο των απεικονίσεων για κάθε $i = 1, \cdots |\mathcal{M}|$ λέγεται κώδικας της πηγής X.*
3. *Μία ντετερμινιστική συνάρτηση αποκωδικοποίησης $g : \mathcal{Y}^n \to \mathcal{M}$ με $g(y_1(i),\ y_2(i), \cdots, y_n(i)) = i$*

Το λάθος που μπορεί να δημιουργηθεί κατά την αποκωδικοποίηση ενός $(\mathcal{M},\ n)$ κώδικας είναι ο δείκτης i που λαμβάνουμε από την συνάρτηση αποκωδικοποίησης g να μην αντιστοιχεί στο δείκτη του μηνύματος που στάλθηκε, δηλαδή:

$$P_e(i) = Pr[g(\mathbf{Y^n}) \neq i|\mathbf{X^n} = \mathbf{x^n(i)}] = \sum_{\mathbf{y^n}\in\mathcal{Y^n}} Pr[\mathbf{Y^n} = \mathbf{y^n}|\mathbf{X^n} = \mathbf{x^n(i)}] \cdot \mathbb{1}_{\{g(\mathbf{Y^n})\neq i\}}$$

Επειδή οι ακολουθίες εισόδου-εξόδου συνδέονται πιθανοκρατικά έπεται ότι κάθε κωδική λέξη $\mathbf{x(i)}$ θα συσχετίζεται με μία διαφορετική πιθανότητα λάθους. Η **μέγιστη πιθανότητα λάθους** για τον κώδικα $(\mathcal{M},\ n)$ δίνεται από τη σχέση:

$$\boxed{P_m^{max}(\mathcal{C}) = \max_{i\in\{1,\cdots,|\mathcal{M}|\}} P_e(i)}$$

Ακόμη η **μέση πιθανότητα** λάθους ορίζεται ως:

$$\overline{P_e^n} = \sum_{\mathbf{y^n}\in\mathcal{Y^n}:g(\mathbf{y^n})\neq i} Pr[\mathbf{Y^n} = \mathbf{y^n}|\mathbf{X^n} = \mathbf{x^n(i)}] \cdot Pr[\mathbf{X^n} = \mathbf{x^n(i)}] =$$

$$\sum_{\mathbf{y^n}\in\mathcal{Y^n}:g(\mathbf{y^n})\neq i} \Pr[\mathbf{Y^n} = \mathbf{y^n}|\mathbf{X^n} = \mathbf{x^n(i)}] \cdot \frac{1}{|\mathcal{M}|} = \frac{1}{|\mathcal{M}|}\sum_{i=1}^{|\mathcal{M}|} P_e(i)$$

Η μέγιστη και η μέση πιθανότητα λάθους αποτελούν δύο μέτρα απόδοσης του κώδικα. Αν ένα κώδικας αποτελείται από $|\mathcal{M}|$ κωδικές λέξεις και υποθέτουμε ότι μία κάθε λέξη του κώδικα επιλέγεται με τυχαίο τρόπο, τότε ξέρουμε ότι $H(\mathcal{C}) = log(|\mathcal{M}|)$. Αν σε κάθε χρήση του καναλιού θέλουμε να μεταδίδεται ένα σύμβολο της κωδικής λέξης $X^n(i)$,τότε χρειάζεται σίγουρα $R = \dfrac{log(|\mathcal{M}|)}{n}$ bits ανά χρήση του καναλιού. Η παραπάνω ποσότητα ονομάζεται **ρυθμός μετάδοσης κώδικα** και αποτελεί επί της ουσίας τα απαραίτητα bits αν σύμβολο κωδικής λέξης που πρέπει να μεταδώσουμε ανά χρήση του καναλιού. Ο τελευταίος ορισμός που θα δώσουμε είναι της επιτευξιμότητας ενός ρυθμού μετάδοσης. Όπως είδαμε για να μεταδώσουμε ένα κώδικα χρειάζεται σε κάθε χρήση καναλιού να μεταδίδονται $\dfrac{log(|\mathcal{M}|)}{n}$ bits. Υπάρχουν όμως κανάλια που η χωρητικότητα τους δεν επιτρέπει τέτοιους ρυθμούς μετάδοσης όπως επίσης υπάρχουν και κώδικες που όταν μεταδοθούν με συγκεκριμένους ρυθμούς είναι ευάλωτοι σε πολλά λάθη.

Ορισμός 4.10. *Ένα ρυθμός R λέγεται* **ϵ-επιτεύξιμος** *αν υπάρχει μία ακολουθία κωδίκων $(\lceil 2^{n\cdot R}\rceil,\ n)$ για την οποία η $|P_m^{max}| < \epsilon$ καθώς το $n \to \infty$.*

Πρακτικά λοιπόν μπορούμε να πούμε πως η χωρητικότητα του καναλιού μπορεί να οριστεί ως το supremum όλων των ε-επιτεύξιμων κωδίκων.

4.5.1 Θεώρημα κωδικοποίησης Καναλιού-Απόδειξης της ευθείας κατεύθυνσης

Θεώρημα 4.4. ***(Θεώρημα κωδικοποίησης για κανάλια με θόρυβο)*** *Για ένα διακριτό κανάλι χωρίς μνήμη και σχόλια όλοι οι ρυθμοί μετάδοσης που είναι μικρότεροι από τη χωρητικότητα του καναλιού είναι επιτεύξιμοι, δηλαδή υπάρχει μία ακολουθία $(\lceil 2^{n\cdot R}\rceil,\ n)$ κωδίκων που το $P_m^{max} \to 0$ καθώς το $n \to \infty$.*

Αντίστροφα αν μία ακολουθία κωδίκων $(\lceil 2^{n\cdot R}\rceil,\ n)$ έχει $P_m^{max} \to 0$ καθώς το $n \to \infty$, τότε $R \leq C$.

Απόδειξη

Απόδειξη ευθείας κατεύθυνσης για ρυθμούς μετάδοσης μικρότερους της χωρητικότητας του καναλιού.

Η απόδειξη δεν είναι κατασκευαστική, δηλαδή δεν μας λέει πως να φτιάξουμε ένα κώδικα με επιτεύξιμο ρυθμό μετάδοσης, απλά μας δείχνει ότι υπάρχει ένας τέτοιος κώδικας. Η λογική που θα ακολουθήσουμε για να αποδείξουμε την ευθεία κατεύθυνση είναι αρχικά να πάρουμε όλους τους κώδικες που χρειάζονται ρυθμούς μετάδοσης χαμηλότερους της χωρητικότητας. Στην συνέχεια θα βρούμε τη μέση πιθανότητα σφάλματος για όλους τους κώδικες. Αν αποδείξουμε ότι αυτή η μέση πιθανότητα είναι μικρή (ϵ), τότε ξέρουμε ότι υπήρχε στο μέσο όρο τουλάχιστον ένας κώδικας με πιθανότητα μικρότερη από ϵ.

Θα διατυπώσουμε την παραπάνω λογική λίγο πιο χοντροκομμένα προκειμένου να γίνει κατανοητή. Σκεφτόμαστε ότι έχουμε n τσουβάλια αλεύρι και θέλουμε να βρούμε αν κάποιο ζυγίζει κάτω από m κιλά. Τότε αν ο μέσο όρος του βάρους των τσουβαλιών είναι μικρότερος από m, ξέρουμε ότι σίγουρα θα υπάρχει τουλάχιστον ένα τσουβάλι με βάρος μικρότερο του m. Έτσι και στο θεώρημα, αν πάρουμε το μέσο όρο της πιθανότητας του σφάλματος που δημιουργείτε από κάθε κώδικα και βρούμε ότι είναι μικρότερη από ϵ, τότε θα υπάρχει τουλάχιστον ένας κώδικα με πιθανότητα λάθους μικρότερη του ϵ.

1. *Αρχικά επιλέγουμε μία αυθαίρετη κατανομή πιθανότητας $P_X(x) = \{Pr[X = x]\}_{x\in\mathcal{X}}$ και με βάση αυτήν δημιουργούμε έναν τυχαίο κώδικα $\mathcal{C} = (\lceil 2^{n\cdot R}\rceil,\ n)$ με $Pr[\mathcal{C}] = \prod_{m=1}^{2^{n\cdot R}} \prod_{i=1}^{n} Pr[X = x_i(m)]$. Συγκεκριμένα δημιουργούμε έναν κώδικα που περιέχει $2^{n\cdot R}$ κωδικές λέξεις κάθε μία από τις οποίες έχει πιθανότητα $Pr[\mathbf{X^n} = \mathbf{x^n(m)}] = \prod_{i=1}^{n} Pr[X = x_i(m)]$.*

2. *Υποθέτουμε ότι ο κώδικας είναι γνωστός στον πομπό και στο δέκτη μαζί με τις πιθανότητες $Pr[Y = y_i|X = x_i]$ του καναλιού. Για να στείλουμε μία ολόκληρη κωδική λέξη $\mathbf{x^n(m)}$ χρειάζεται απλά να χρησιμοποιήσουμε το κανάλι n φορές.*

3. *Διαλέγουμε με τυχαίο τρόπο ένα μήνυμα m να στείλουμε που αντιστοιχεί στην κωδική λέξη $\mathbf{x^n(m)}$. Τότε μετά από n χρήσεις του καναλιού ο δέκτης λαμβάνει την κωδική λέξη y^n για την οποία ισχύει:*

$$Pr[\mathbf{Y^n} = \mathbf{y^n}|\mathbf{X^n} = \mathbf{x^n(m)}] = \prod_{i=1}^{n} Pr[Y = y_i|X = x_i(m)]$$

4. *Ο δέκτης αποκωδικοποιεί το μήνυμα στο $\hat{m}$ κοιτώντας για ποια κωδική λέξη $\mathbf{x^n}(\hat{\mathbf{m}})$ ισχύει ότι το ζεύγος $(\mathbf{x^n}(\hat{\mathbf{m}}),\ \mathbf{y^n})$ ανήκει στις τυπικές από κοινού ακολουθίες. Αν το m δεν υπάρχει ή υπάρχει και άλλο $\hat{m'}$ που είναι τυπικά συνδεδεμένο με τη ληφθείσα ακολουθία $\mathbf{y^n}$, τότε ο αποκωδικοποιητής έχει πέσει σε σφάλμα.*

5. *Το ενδεχόμενο λάθους είναι λοιπόν* $\mathcal{E} = \{g(\mathbf{Y} = \mathbf{y^n}) \neq m\}$. *Τώρα θα βρούμε το μέσο όρο λάθους που παράγεται από το σύνολο των κωδίκων* $\mathcal{C} = (\lceil 2^{n \cdot R} \rceil,\ n)$. *Ο μέσος όρος ορίζεται ως σταθμισμένος μέσος των μέγιστων πιθανοτήτων λάθους όλων των κωδίκων. Ο λόγος για τον οποίο ορίζεται έτσι το* $Pr[\mathcal{E}]$ *είναι πώς* $\forall\ \mathcal{C}\ \wedge \forall m \in \mathcal{M} \Rightarrow P_e(m) \leq P_m^{max}(\mathcal{C})$. *Άρα αν ο μέσος όρος των μέγιστων πιθανοτήτων όλων των κωδίκων είναι μικρότερος από* ϵ *τότε υπάρχει ένας τουλάχιστον κώδικας* $\mathcal{C}$ *που έχει* $P_m^{max}(\mathcal{C}) < \epsilon$.

$$Pr[\mathcal{E}] = \sum_{\mathcal{C}} Pr[\mathcal{C}] \cdot \overline{P_e^n}(C) = \sum_{\mathcal{C}} Pr[\mathcal{C}] \cdot \frac{1}{2^{n \cdot R}} \cdot \sum_{m=1}^{2^{n \cdot R}} P_m^{max}(\mathcal{C})$$

Το εσωτερικό όμως άθροισμα είναι ανεξάρτητο της λέξης που επιλέχθηκε, άρα

$$Pr[\mathcal{E}] = \sum_{\mathcal{C}} Pr[\mathcal{C}] \cdot \frac{1}{2^{n \cdot R}} \cdot \sum_{m=1}^{2^{n \cdot R}} P_e^{\mathcal{C}}(1) = \sum_{\mathcal{C}} Pr[\mathcal{C}] \cdot \frac{1}{2^{n \cdot R}} \cdot 2^{n \cdot R} \cdot P_1^{max}(\mathcal{C}) = Pr[\mathcal{E}|m = 1]$$

Όποτε έχουμε φτάσει στο σημείο να εκφράσουμε τη μέση πιθανότητα λάθους ως την πιθανότητα λάθους που θα προκύψει αν επιλέξουμε να κωδικοποιήσουμε την λέξη που αντιστοιχεί στο δείκτη 1 του κάθε κώδικα, ο οποίος δείκτης 1 αντιστοιχεί σε όλους στους κώδικες στο μήνυμα που δίνει την μέγιστη πιθανότητα λάθους. Ας συμβολίσουμε με $E_i = \{(\mathbf{x^n(i)},\ \mathbf{y^n}) \in J_\epsilon^n\}$, *τότε η πιθανότητα να γίνει λάθος κατά την κωδικοποίηση του 1 είναι να μην συμβεί του το* E_1, *δηλαδή το* $\mathbf{x^n(i)}$ *να μην είναι τυπικά συνδεδεμένο με την* $\mathbf{y^n}$ *ή να συμβεί κάποιο από τα άλλα* E_i, *δηλαδή η ληφθείσα ακολουθία* $\mathbf{y^n}$ *να είναι τυπικά συνδεδεμένη με κάποιο άλλο μήνυμα που δεν είναι το 1.*

$$Pr[\mathcal{E}|m = 1] = Pr[E_1^c \cup E_2 \cup E_3 \cup \cdots \cup E_n|m = 1 \Rightarrow]$$
$$Pr[\mathcal{E}|m = 1] \leq Pr[[E_1^c|m = 1] + \sum_{m=2}^{2^{n \cdot R}} Pr[E_i|m = 1]$$

Από το θεώρημα ασυμπτωτικής κατανομής για από κοινού ακολουθίες έχουμε ότι η πιθανότητα να μεταδοθεί μια ακολουθία εξόδου $\mathbf{y^n}$ *που δεν είναι τυπικά συνδεδεμένη με την κωδική λέξη που αντιστοιχεί στο* $m = 1$ *ενώ ξέρουμε ότι μεταδόθηκε είναι* $\leq \epsilon$. *Επίσης η πιθανότητα να συνδέεται η ακολουθία* $\mathbf{y^n}$ *με οποιαδήποτε άλλη κωδική λέξη* $\mathbf{x^n(i)}$ *με* $i \neq 1$ *είναι* $\leq 2^{-n \cdot (I(X,Y) - 3 \cdot \epsilon)}$, *άρα*

$$Pr[\mathcal{E}|m = 1] \leq \epsilon + \sum_{m=2}^{2^{n \cdot R}} 2^{-n \cdot (I(X,Y) - 3 \cdot \epsilon)} = \epsilon + (2^{n \cdot R} - 1) \cdot 2^{-n \cdot (I(X,Y) - 3 \cdot \epsilon)} \leq \epsilon + 2^{n \cdot R} \cdot 2^{-n \cdot (I(X,Y) - 3 \cdot \epsilon)} \Rightarrow$$
$$Pr[\mathcal{E}|m = 1] \leq \epsilon + 2^{-n \cdot (I(X,Y) - R - 3 \cdot \epsilon)}$$

Αν διαλέξουμε ένα ρυθμό μετάδοσης R *για το οποίο ισχύει* $I(X,Y) - R - 3 \cdot \epsilon > 0 \Rightarrow R < I(X,Y) - 3 \cdot \epsilon \Rightarrow R < I(X,Y)$, *τότε για αρκετά μεγάλο* n *η ποσότητα* $2^{-n \cdot (I(X,Y) - R - 3 \cdot \epsilon)}$ *γίνεται πολύ μικρή. Διαλέγουμε ένα τέτοιο* ϵ *ώστε* $2^{-n \cdot (I(X,Y) - R - 3 \cdot \epsilon)} < \epsilon$. *Τότε θα έχουμε*

$$Pr[\mathcal{E}|m = 1] \leq 2 \cdot \epsilon$$

Μόλις αποδείξαμε ότι αν ο ρυθμός μετάδοσης δεν ξεπερνάει την αμοιβαία πληροφορία, τότε ο μέσο όρος λάθους για όλους του κώδικες είναι μικρότερος από 2ϵ, *άρα σίγουρα θα υπάρχει τουλάχιστον ένας κώδικας που για* $n \to \infty$ *θα ισχύει ότι το* $Pr_m^{max}(\mathcal{C}) < 2 \cdot \epsilon$.

6. **Αυστηροποίηση του φράγματος για τους ρυθμούς μετάδοσης R**

Μέχρι στιγμής έχουμε αποδείξει ότι για $R < I(X,Y)$ έχουμε αμελητέα μέση πιθανότητα λάθους. Θα αυστηροποιήσουμε λίγο παραπάνω το άνω φράγμα για το R. Αντί για μία τυχαία κατανομή $P_X(x)$, διαλέγουμε εκείνη που μεγιστοποιεί την $I(X,Y)$ και παράγουμε πάλι κατά τον ίδιο τρόπο ($\lceil 2^{n\cdot R} \rceil$, n) κώδικες. Τότε η συνθήκη $R < I(X,Y)$ που καταλήξαμε στο βήμα 5 μπορεί να αντικατασταθεί από την $R < C$. Από την προηγούμενη ανάλυση ξέρουμε ότι η μέση πιθανότητα λάθους είναι μικρότερη του $2 \cdot \epsilon$. Σίγουρα λοιπόν υπάρχει ένας κώδικας $\mathcal{C}^*$ για το οποίο ισχύει $Pr[\mathcal{E}|\mathcal{C}^*] \leq 2 \cdot \epsilon$. Αυτός ο βέλτιστο κώδικας $\mathcal{C}^*$ που έχει αμελητέα μέγιστη πιθανότητα λάθους μπορεί να ευρεθεί με εξαντλητική αναζήτηση των $2^{n\cdot R}$ κωδίκων που κατασκευάσαμε. Διατάσσοντας τις κωδικές λέξεις του $\mathcal{C}$ σύμφωνα με την πιθανότητα λάθους της κάθε μίας μπορούμε να πετάξουμε τις μισές "κακές" κωδικές λέξεις που δημιουργούν μεγάλες πιθανότητες λάθους. Από την στιγμή που ξέρουμε ότι $P_m^{max}(\mathcal{C}^*) \leq 2 \cdot \epsilon$ έπεται ότι $P_e(i) \leq 2\epsilon \ \forall m \in \mathcal{M} \Rightarrow \overline{P_e^n} \leq 2 \cdot \epsilon$ Αφού λοιπόν η μέση πιθανότητα λάθους του κώδικα φράσσεται από την ποσότητα $2 \cdot \epsilon$ έπεται πώς όταν πετάξουμε τις μισές κακές λέξεις το $P_e' = \frac{1}{\frac{|\mathcal{M}|}{2}} \sum_{i=1}^{\frac{|\mathcal{M}|}{2}} P_e(i) = \frac{2}{|\mathcal{M}|} \sum_{i=1}^{\frac{|\mathcal{M}|}{2}} P_e(i) \leq \frac{2}{|\mathcal{M}|} \sum_{i=1}^{|\mathcal{M}|} P_e(i) \leq 2 \cdot 2 \cdot \epsilon \leq 4 \cdot \epsilon$. Αυτό με τη σειρά του συνεπάγεται ότι οι μισές "καλές" λέξεις που απομείναν θα έχουν $P_m^{max}(\mathcal{C}) \leq 4 \cdot \epsilon$. Τότε το $R = \frac{\log|\mathcal{M}|}{n}$ γίνεται $R' = \frac{\log\frac{|\mathcal{M}|}{2}}{n} = \frac{\log|\mathcal{M}| - 1}{n} = \frac{\log|\mathcal{M}|}{n} - \frac{1}{n} \Rightarrow R' = R - \frac{1}{n} \Rightarrow R' \overset{n\to\infty}{\to} R$. Άρα κατασκευάσαμε έναν κώδικα με $R' < C$ ο οποίος επιτυγχάνει $P_m^{max} \leq 4 \cdot \epsilon$

4.5.2 Βασικά αποτελέσματα για την απόδειξη της αντίστροφης κατεύθυνσης του θεωρήματος.

Πριν αποδείξουμε την αντίστροφη κατεύθυνση θα παρουσιάσουμε κάποια βασικά θεωρήματα που θα χρησιμοποιηθούν κατά τη διάρκεια της απόδειξης. Στην αρχή που δόθηκαν οι ορισμοί για ένα κανάλι χωρίς μνήμη και σχόλια, καταλήξαμε στο συμπέρασμα ότι η ολόκληρη η διαδικασία από την παραγωγή ενός μηνύματος m, στη κωδικοποίηση του σε x^n, την μετάδοση του σε y^n μέχρι και την αποκωδικοποίηση σε μία εκτίμηση του μηνύματος $\hat{m}$ συνιστά μία μαρκοβιανή αλυσίδα:

$$M \to X^n \to Y^n \to \hat{M}$$

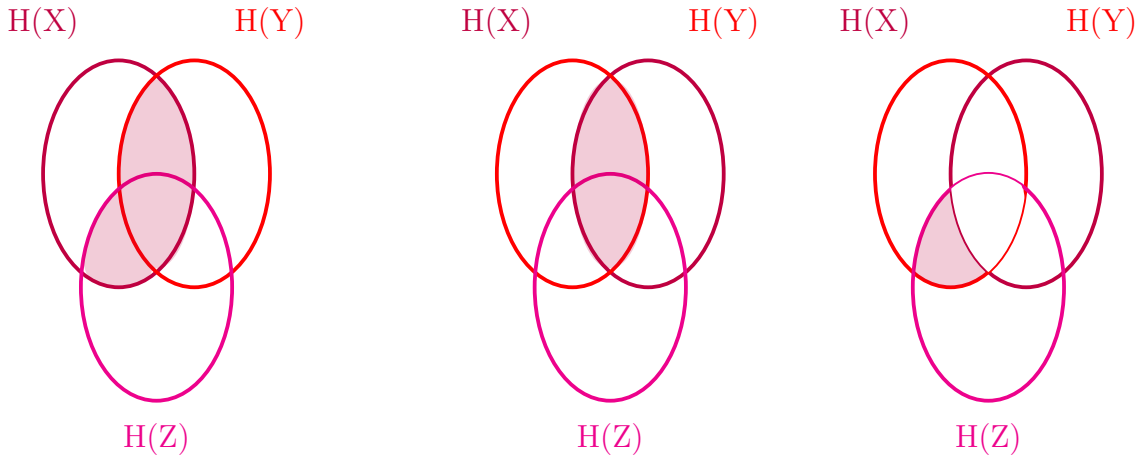
Αυτο σημαίνει ότι από κοινού κατανομή του καναλιού $Pr[M = m, X^n = x^n, Y^n = y^n, \hat{M} = \hat{m}]$ μπορεί να εκφραστεί ως:

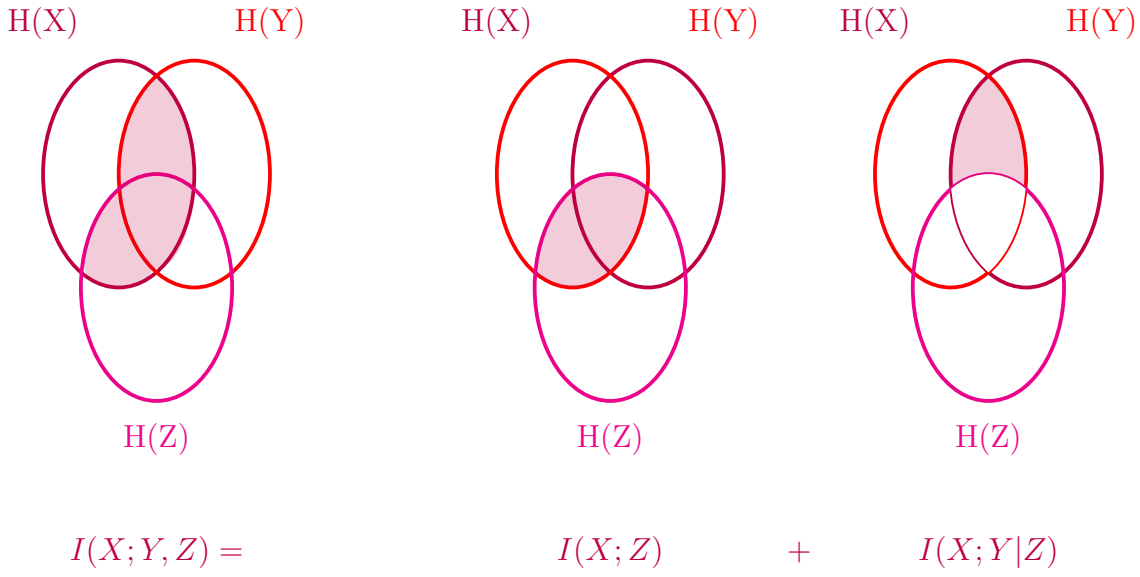
$$Pr[M = m, X^n = x^n, Y^n = y^n, \hat{M} = \hat{m}] =$$
$$Pr[M = m] \cdot Pr[X^n = x^n | M = m] \cdot Pr[Y^n = y^n | X^n = x^n] \cdot Pr[\hat{M} = \hat{m} | Y^n]$$

Η παραπάνω μαρκοβιανή αλυσίδα μας δείχνει κατά κάποιο τρόπο "πως ρέει" η πληροφορία. Ένα εύλογο ερώτημα είναι να δούμε τις σχέσεις που σχηματίζονται μεταξύ των αμοιβαίων μέτρων πληροφορίας ανάμεσα στις διάφορες μεταβλητές. Για παράδειγμα μπορούμε διαισθητικά να καταλάβουμε ότι η πληροφορία που εμπεριέχεται αρχικά στο μήνυμα m καθώς περνάει από τα διάφορα στάδια του καναλιού επικοινωνίας φθίνει λόγω της επίδρασης του θορύβου. Έτσι όταν φτάσει στο δέκτη μία "έκτίμηση" του αποσταλθέντος μηνύματος, τότε αυτή θα κατέχει πολύ λιγότερη πληροφορία για το αρχικό m από ότι περιέχει η κωδικοποίηση του $x^n(m)$. Άρα λογικά θα πρέπει $I(M; X^n) \geq I(M, \hat{M})$. Αυτή η διαισθητική αντίληψη για την φθίνουσα ποσότητα της πληροφορίας καθώς ρέει μέσα από μια μαρκοβιανή αλυσίδα ενσαρκώνεται μέσα από το παρακάτω θεώρημα που λέγεται η ανισότητα της επεξεργασίας δεδομένων (data processing inequality).

Θεώρημα 4.5. *(**Ανισότητα της επεξεργασίας δεδομένων**) Έστω $X \to Y \to Z$ μία μαρκοβιανή αλυσίδα που σχηματίζουν οι τυχαίες μεταβλητές X, Y, Z τότε:*

$$I(X;Y) \geq I(X;Z) \tag{4.11}$$





Από το σχήμα βλέπουμε ότι:

$$I(X;Y,Z) = I(X;Y) + I(X;Z|Y) = I(X;Z) + I(X;Y|Z)$$

Επειδή όμως τα X και Z είναι ανεξάρτητα δεδομένου του Y έπεται ότι $I(X,Z|Y) = 0$. Άρα:

$I(X;Y)+I(X;Z|Y)=I(X;Z)+I(X;Y|Z)\Rightarrow I(X;Y)=I(X;Z)+I(X;Y|Z)\geq I(X;Z)\Rightarrow$
$\boxed{I(X;Y)\geq I(X;Z)}$

Για να ισχύει η ισότητα θα πρέπει το $I(X;Y|Z)=0$, το οποίο σημαίνει ότι τα X και Y είναι ανεξάρτητα δεδομένου του Z, δηλαδή $X\rightarrow Y\rightarrow Z$.

Ένα ακόμη σημείο που πρέπει να εξετάσουμε είναι το φράγμα της αμφιβολίας που έχουμε για το απεσταλμένο μήνυμα όταν ξέρουμε το ειλημμένο, δηλαδή ζητάμε ένα άνω φράγμα για το $H[X|Y]$. Επειδή η αμφιβολία που έχουμε σχετίζεται με την πιθανότητα λάθους λόγω ύπαρξης θορύβου περιμένουμε το φράγμα για την $H[X|Y]$ να σχετίζεται με το P_e. Συγκεκριμένα αναμένουμε όσο μικρότερη πιθανότητα σφάλματος υπάρχει κατά την κωδικοποίηση τόσο μικρότερη να είναι και η αμφιβολία που έχουμε για την ακολουθία από την οποία προήλθε η Y. Η σχέση που αναπτύσσεται ανάμεσα στις δύο ποσότητες ονομάζεται ανισότητα του Fano

Θεώρημα 4.6. ***(Ανισότητα του Fano)*** *Για οποιαδήποτε εκτιμήτρια μεταβλητή $\hat{X}$ τέτοια ώστε $X\rightarrow Y\rightarrow\hat{X}$ με $P_e=Pr[X\neq\hat{X}]$*

1.
$$H(X|Y)\leq H(X|\hat{X})\leq H(P_e)+P_e\cdot log|\mathcal{X}| \tag{4.12}$$

2.
$$H(X|Y)\leq P_e\cdot log|\mathcal{X}|+1$$

Απόδειξη

1. *Το $I(X;Y)=H(X)-H(X|Y)$ και $I(X;\hat{X})=H(X)-H(X|\hat{X})$. Επειδή τα X, Y, $\hat{X}$ σχηματίζουν μία μαρκοβιανή αλυσίδα έπεται ότι $I(X;Y)\geq I(X,\hat{X})\Rightarrow\boxed{H(X|Y)\leq H(X|\hat{X})}$.*

   Το $H(X|\hat{X})$ είναι η αμφιβολία που έχουμε ότι εκτιμήσαμε σωστά τη μεταβλητή X. Έστω η δείκτρια συνάρτηση:
$$\mathbb{1}=\begin{cases}1 & X\neq\hat{X}\\ 0 & X=\hat{X}\end{cases}$$

   Τότε $H(\mathbb{1},X|\hat{X})=H(X|\hat{X})+H(\mathbb{1}|X,\hat{X})=H(\mathbb{1}|\hat{X})+H(X|\mathbb{1},\hat{X})\Rightarrow H(X|\hat{X})+H(\mathbb{1}|X,\hat{X})=H(\mathbb{1}|\hat{X})+H(X|\mathbb{1},\hat{X})\Rightarrow H(X|\hat{X})=H(\mathbb{1}|\hat{X})+H(X|\mathbb{1},\hat{X})\Rightarrow\boxed{H(X|\hat{X})\leq H(\mathbb{1})+H(X|\mathbb{1},\hat{X})}$(1)

   Η αμφιβολία να έχει γίνει κάποιο λάθος ποσοτικοποιείται από την ποσότητα $H(\mathbb{1})$ και είναι ίση με την $H(P_e)$, $\boxed{H(\mathbb{1})=H(P_e)}$(2). Επίσης το:

   $H(X|\mathbb{1},\hat{X})=H(X|\mathbb{1}=1,\hat{X})\cdot Pr[\mathbb{1}=1]+H(X|\mathbb{1}=0,\hat{X})\cdot Pr[\mathbb{1}=0]$

   Η αμφιβολία ότι η εκτίμηση $\hat{X}\neq X$ όταν ξέρουμε ότι δεν έχει γίνει λάθος($\mathbb{1}=0$), θα είναι μηδέν ενώ η αμφιβολία $\hat{X}\neq X$ όταν ξέρουμε ότι έχει γίνει λάθος($\mathbb{1}=1$) ισοδυναμεί με το να έχει επιλεγεί κάποια άλλη τιμή της τυχαία μεταβλητής X. Άρα τότε το $H(X|\mathbb{1}=1,\hat{X})$ θα ισοδυναμεί με την $H(X)$. Οπότε:

   $H(X|\mathbb{1},\hat{X})=H(X|\mathbb{1}=1,\hat{X})\cdot Pr[\mathbb{1}=1]+H(X|\mathbb{1}=0,\hat{X})\cdot Pr[\mathbb{1}=0]\Rightarrow H(X|\mathbb{1},\hat{X})\leq P_e\cdot H(X)\Rightarrow$
   $\boxed{H(X|\mathbb{1},\hat{X})\leq P_e log|\mathcal{X}|}$(3)

   Συνδυάζοντας τις σχέσεις (1) (2) και (3) θα έχουμε:

$$\boxed{H(X|\hat{X})\leq H(P_e)+P_e\cdot log|\mathcal{X}|}$$

   και επειδή $H(X|Y)\leq H(X|\hat{X})\Rightarrow\leq H(X|\hat{X})\leq H(P_e)+P_e\cdot log|\mathcal{X}|$

2. Επίσης $H(P_e) = H(P_e,\ 1-P_e) \leq log2 = 1 \Rightarrow H(X|Y) \leq 1 + P_e \cdot log|\mathcal{X}|$

Με απλά λόγια η ανισότητα του Fano μας λέει πως η αμφιβολία για το αν η εκτίμησή μας είναι σωστή φράσσεται από την αμφιβολία του λάθους $H(P_e)$ *και αν έχει γίνει λάθος τότε αυτό φράσσεται από το χειρότερο σενάριο να θεωρούνται όλα τα μηνύματα που οδηγούν στο* Υ *ισοπίθανα.*

4.5.3 Θεώρημα κωδικοποίησης Καναλιού- Απόδειξη της αντίστροφης κατεύθυνσης

Για να αποδείξουμε την αντίστροφη κατεύθυνση αρκεί να δείξουμε πως ένα κώδικας $(\lceil 2^{n\cdot R}\rceil,\ n)$ όπου το $P_m^{max}(\mathcal{C}) \overset{n\to\infty}{\to} 0$ τότε το $R \leq C$. Αφού λοιπόν απαιτούμε $P_m^{max}(\mathcal{C}) \overset{n\to\infty}{\to} 0 \Rightarrow \overline{P_e^n} \overset{n\to\infty}{\to} 0$ Σε ένα διακριτό κανάλι χωρίς μνήμη και σχόλια είδαμε ότι οι μεταβλητές M, x^n, $Y^n, \hat{M}$ σχηματίζουν μία μαρκοβιανή αλυσίδα. Επειδή οι διαδικασίες τις κωδικοποίησης και αποκωδικοποίησης είναι ντετερμινιστικές έπεται ότι $H(X^n|M) = H(M|X^n) = H(Y^n|\hat{M}) = H(\hat{M}|Y^n) = 0$. Επειδή η μέγιστη πιθανότητα λάθους πρέπει να τείνει στο μηδέν επίσης έπεται ότι τα M και $\hat{M}$ θα πρέπει να είναι σχεδόν τα ίδια, δηλαδή $H(M|\hat{M}) \approx H(\hat{M}|M) \approx 0$

Έστω ότι επιλέγουμε τυχαία ένα M να σταλθεί, τότε το $H(M) = log|\mathcal{M}| = log2^{n\cdot R} = n \cdot R$, άρα:

$$n \cdot R = H(M) = H(M|\hat{M}) + I(M;\hat{M}) \overset{\text{Ανισότητα Fano}}{\leq} 1 + \overline{P_e^n} \cdot n \cdot R + I(M;\hat{M}) \overset{\text{ανισότητα επεξεργασίας δεδομένων}}{\leq}$$

$$1 + \overline{P_e^n} \cdot n \cdot R + I(X^n;Y^n) \leq 1 + \overline{P_e^n} \cdot n \cdot R + n \cdot C \Rightarrow R \leq \frac{1}{n} + \overline{P_e^n} + C \overset{n\to\infty}{\Rightarrow} R \leq C$$

Θα αποδείξουμε γιατί ισχύει η τρίτη ανισότητα.

$$I(\mathbf{X^n},\ \mathbf{Y^n}) = H(\mathbf{Y^n}) + H(\mathbf{Y^n}|\mathbf{X^n}) = H(\mathbf{Y^n}) + \sum_{i=1}^{H(Y_i|X_i)} \overset{\text{Κανόνα της αλυσίδας}}{\leq} \sum_{i=1}^{n} H(Y_i) - \sum_{i=1}^{n} H(Y_i|X_i) =$$

$$\sum_{i=1}^{n} H(Y_i) - H(Y_i|X_i) = \sum_{i=1}^{n} I(X_i;Y_i) \leq \sum_{i=1}^{n} C = n \cdot C$$

Η σχέση $R \leq \frac{1}{n} + \overline{P_e^n} + C$ μπορεί να ξαναγραφτεί σαν $\overline{P_e^n} \geq 1 - \frac{C}{R} - \frac{1}{n\cdot R}$. Τότε στην περίπτωση που προσπαθήσουμε να μεταδώσομε με $R > C \Rightarrow \frac{C}{R} < 1 \Rightarrow \overline{P_e^n} \overset{n\to\infty}{\to} 1$

Για να δούμε όμως τι σημαίνει πρακτικά το αποτέλεσμα που βγάλαμε. Μόλις αποδείξαμε πως αν το $P_m^{max}(\mathcal{C}) \overset{n\to\infty}{\to} 0 \Rightarrow \overline{P_e^n} \overset{n\to\infty}{\to} 0$ τότε το $R < C$. Τι σημαίνει όμως το $\overline{P_e^n} \overset{n\to\infty}{\to} 0$; Από την ανισότητα του Fano γνωρίζουμε πως $H(X|\hat{X}) \leq H(P_e) + P_e \log|\mathcal{X}|$. Στην δική μας περίπτωση η ανισότητα μεταφράζεται σε $H(M|\hat{M}) \leq H(\overline{P_e^n}) + \overline{P_e^n} \cdot n \cdot R$. Όταν λοιπόν το $\overline{P_e^n} \overset{n\to\infty}{\to} 0 \Rightarrow H(M|\hat{M}) = 0$. Τότε όμως από την ανισότητα της απόδειξης που δημιουργείται χρησιμοποιώντας την ανισότητα Fano προκύπτει ότι $H(M) \leq I(M;\hat{M})$. Εδώ ακριβώς βρίσκεται όλη η ουσία. Όταν το $H(M) \leq I(M;\hat{M})$ αυτό σημαίνει ότι η κοινή πληροφορίας που περιέχει η εκτιμήτρια του μηνύματος $\hat{M}$ και το μήνυμα M είναι περισσότερη από την εντροπία του μηνύματος. Αφού λοιπόν έχουμε ορίσει την εντροπία μια τυχαίας μεταβλητής ως την ελάχιστη πληροφορία που είναι απαραίτητη ώστε να την περιγράψει έπεται ότι η πληροφορία $I(M;\hat{M})$ που φτάνει στο δέκτη μέσω της εκτιμήτριας είναι αρκετή για να περιγράψει το μήνυμα. Αυτό λοιπόν επιτυγχάνεται όταν στέλνουμε με $R < C$.

4.6 Διακριτά κανάλια χωρίς μνήμη που περιέχουν σχόλια

Όταν ένα κανάλι έχει σχόλια όπως έχουμε αναφέρει το σύμβολο εισόδου x_i στο κανάλι δεν εξαρτάται μόνο από το μήνυμα M που επιλέγεται να κωδικοποιηθεί εκείνη την στιγμή αλλά και από τα σύμβολα εξόδου

$y_1, \cdots, y_{i-1}$.Υποθέτουμε ότι στο κανάλι της επιστροφής που στέλνει τα σχόλια δεν υπάρχει θόρυβος. Το ερώτημα που γεννάται είναι αν η ύπαρξη σχολίων μπορεί να αυξήσει την χωρητικότητα του καναλιού. Απλοϊκά σκεπτόμενοι θα μπορούσαμε να πούμε πως από την στιγμή η πηγή μπορεί να δει τι έχει φτάσει στο δέκτη μέχρι στιγμής τότε αναπροσαρμόζοντας τις εισόδους της θα μπορέσει να στείλει τα κατάλληλα $x_i + 1, \cdots, x_n$ ώστε να ελαχιστοποιήσει την πιθανότητα να γίνει κάποια λάθος αποκωδικοποίηση. Η σκέψη αυτή θα είχε κάποιο νόημα αν το κανάλι είχε μνήμη, δηλαδή αν θυμόταν τι έχει σταλεί μέχρι εκείνη την στιγμή ίσως θα μπορούσε να παραμετροποιήσει τις εισόδους ώστε να αντιπαλέψει τα σφάλματα. Επειδή το κανάλι είναι αμνήμων δεν μπορεί να ξέρει, παρά τα σχόλια, αν αυτά που έχει στείλει μέχρι εκείνη την στιγμή οδηγούν σε σφάλμα η όχι. Άρα είτε με σχόλια είτε χωρίς η πιθανότητα λάθους παραμένει ανεπηρέαστη. Αυτό που αλλάζει σε σχέση με την προηγούμενη περίπτωση είναι ότι δεν ισχύει η μαρκοβιανή αλυσίδα $M \to X^n \to Y^n \to \hat{M}$, καθώς το X^n, δεν εξαρτάται μόνο από το M αλλά και από τις προηγούμενες εξόδους. Ισχύει όμως η εξής μαρκοβιανή αλυσίδα $M \to Y^n \to \hat{M}$ καθώς το κανάλι συνεχίζει να είναι αμνήμων.

Επιλέγοντας πάλι ένα μήνυμα M τυχαία έχουμε:

$$n \cdot R = H(M) = H(M|\hat{M}) + I(M;\hat{M}) \overset{\text{Ανισότητα Fano}}{\leq} 1 + \overline{P_e^n} \cdot n \cdot R + I(M;\hat{M}) \overset{\text{ανισότητα επεξεργασίας δεδομένων}}{\leq}$$

$$1 + \overline{P_e^n} \cdot n \cdot R + I(M;Y^n) = 1 + \overline{P_e^n} \cdot n \cdot R + H(Y^n) - H(Y^n|M) =$$

$$1 + \overline{P_e^n} \cdot n \cdot R + H(Y^n) - \sum_{i=1}^{n} H(Y_i|Y_1, \cdots, Y_{i-1}, W) = 1\overline{P_e^n} \cdot n \cdot R + H(Y^n) - \sum_{i=1}^{n} H(Y_i|Y_1, \cdots, Y_{i-1}, X_i, W)$$

$$\overset{X_i = f(W, Y_1, \cdots, Y_{i-1}), Y_i \perp\!\!\!\perp W | X_i}{=} 1 + \overline{P_e^n} \cdot n \cdot R + H(Y^n) - \sum_{i=1}^{n} H(Y_i|X_i) \leq 1 + \overline{P_e^n} \cdot n \cdot R + \sum_{i=1}^{n} H(Y_i) - \sum_{i=1}^{n} H(Y_i|X_i) =$$

$$1 + \overline{P_e^n} \cdot n \cdot R + \sum_{i=1}^{n} H(Y_i) - H(Y_i|X_i) = 1 + \overline{P_e^n} \cdot n \cdot R + \sum_{i=1}^{n} I(X_i;Y_i) \leq 1 + \overline{P_e^n} \cdot n \cdot R + n \cdot C$$

Άρα καταλήγουμε πάλι ότι $n \cdot R \leq 1 + P_e^{(n)} \cdot n \cdot R + n \cdot C \Rightarrow \boxed{R \leq \frac{1}{n} + P_e^{(n)} + C \overset{n \to \infty}{\Rightarrow} R \leq C}$

Άρα ένα κώδικας σε ένα κανάλι με σχόλια πάλι δεν ξεπερνάει την προκαθορισμένη χωρητικότητα του καναλιού.

Κεφάλαιο 5

Μέθοδοι συμπίεσης

5.1 Η μέθοδος κωδικοποίησης του Fano

5.2 Παρουσίαση της μεθόδου

Για να κωδικοποιήσουμε μία τυχαία μεταβλητή X με τιμές στο $\mathcal{X} = \{x_1,\ x_2, \cdots, x_n\}$ και $P_X(x) = \{Pr[X = x_1],\ Pr[X = x_2], \cdots, Pr[X = x_n]\}$ σύμφωνα με την κωδικοποίηση Fano αρχικά ταξινομούμε τις τιμές της κατά φθίνουσα σειρά ως προς τις πιθανότητες τους και ύστερα διαμερίζουμε αναδρομικά το αρχικό σύνολο σε δύο (σχεδόν) ισοπίθανα υποσύνολα μέχρι να προκύψουν μονοσύνολα. Γενικά οι αλγοριθμικές διαδικασίες που περιλαμβάνουν διαμερίσεις και αναδρομικές ή επαναληπτικές διαδικασίες τείνουν να αναπαριστώνται από δυαδικά δένδρα. Η διαδικασία της κωδικοποίησης Fano ακολουθεί μία άπληστη λογική[1] και παρουσιάζεται στα παρακάτω βήματα.

Διαδικασία κωδικοποίησης

1. Διατάσσουμε τις τιμές της X κατά φθίνουσα σειρά σύμφωνα με τις πιθανότητες εμφάνισης τους και τις ορίζουμε σαν ρίζα του δένδρου κωδικοποίησης.

2. Χωρίζουμε το διατεταγμένο σύνολο$\{x_1,\ x_2, \cdots, x_n\}$ σε δύο σχεδόν ισοπίθανα υποσύνολα. Αν $r > 0$ ο μεγαλύτερος θετικό ακέραιος για τον οποίο ισχύει:

$$\{x_1,\ x_2, \cdots, x_n\} = \begin{cases} \{x_1,\ x_2, \cdots, x_r\} \\ \{x_{r+1},\ x_{r+2}, \cdots, x_n\} \end{cases}$$

   με $P_1^r = \sum_{i=1}^{r} Pr[X = x_i]$ και $P_2^r = \sum_{i=r+1}^{n} Pr[X = x_i] = 1 - P_1^r$, τότε τυπικά το πρόβλημα αναγάγετε στην εύρεση του μεγαλύτερου θετικού ακεραίου $r > 0$, για το οποίο ισχύει ότι

$$|P_1^r - P_2^r| \leq |P_1^{r+1} - P_2^{r+1}|$$

3. Στο ένα υποσύνολο που παίζει το ρόλο του αριστερού παιδιού αντιστοιχίζουμε τον ακέραιο 0 και στο άλλο που θα αποτελέσει το δεξιό παιδί δίνουμε την τιμή 1.

4. Αν είτε το δεξιό είτε το αριστερό παιδί περιέχει παραπάνω από ένα στοιχεία, το διαμερίζουμε αναδρομικά επαναλαμβάνοντας τα βήματα 2 με 3. Επειδή κάθε εσωτερικός κόμβος που δεν είναι φύλο, άρα θα διαμεριστεί, περιέχει κάποιο υποσύνολο του $\mathcal{X}$ της μορφής $\{x_i,\ x_{i+1}, \cdots, x_j\}$ η συνθήκη διαμέρισης αναδιατυπώνεται ως εξής:

   "Βρείτε τον μεγαλύτερο θετικό ακέραιο $r > 0$ για τον οποίο ισχύει:

[1] Ως άπληστη λογική εννοούμε κάθε υπολογιστική διαδικασία η οποία σε κάθε βήμα παίρνει αποφάσεις που φαντάζουν βέλτιστες την δεδομένη στιγμή χωρίς να εξετάζονται καλύτερες λύσεις που μπορούν να παρουσιαστούν στο μέλλον.

$$|P_1^r - P_2^r| \leq |P_1^{r+1} - P_2^{r+1}|$$

,

με $P_1^r = \sum_{k=i}^{r} Pr[X = x_k]$ και $P_2^r = \sum_{k=r+1}^{j} Pr[X = x_k]$"

5. Αφού φτιάξουμε το δένδρο κωδικοποίησης εξάγουμε του κώδικες των συμβόλων ξεκινώντας από τη ρίζα και γράφοντας 0 ή 1 στην έξοδο του κωδικοποιητή ανάλογα το κλαδί του μονοπατιού το οποίο θα μας οδηγήσει από τη ρίζα στο φύλλο με το ζητούμενο σύμβολο.

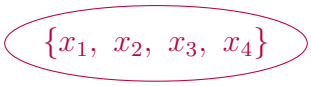
Παρατήρηση: Επειδή το δυαδικό δένδρο που προκύπτει έχει όλα τα σύμβολα στα φύλλα έπεται ότι θα παράγει ένα στιγμιαίο κώδικα. Η ιδιότητα αυτή μας διευκολύνει πολύ κατά την κωδικοποίηση καθώς μπορούμε να στέλνουμε την ακολουθία εξόδου του κωδικοποιητή χωρίς να χρειάζεται πρώτον επιπρόσθετη πληροφορία για τον διαχωρισμό της από τις επόμενες και δεύτερον δεν είναι αναγκαίο να περιμένουμε να ολοκληρωθεί η διαδικασία της κωδικοποίησης ώστε να αρχίσει η αποκωδικοποίηση.

Ένα από τα αρνητικά βέβαια αυτής της μεθόδου είναι ότι πρέπει πριν από όλα να στείλουμε το δένδρο κωδικοποίησης, το οποίο δημιουργεί προβλήματα ταχύτητας, χώρου και αποδοτικότητας. Η αποδοτικότητα έγκειται στην στατικότητα του δένδρου. Φανταστείτε στην πράξη οι δύο πλευρές του κωδικοποιητή και αποκωδικοποιητή να έχουν συμφωνήσει σε ένα προκαθορισμένο δένδρο ώστε να γλυτώσουν τον χρόνο που χρειάζεται για την μετάδοση του και να αυξήσουν την ταχύτητα της υπηρεσίας του. Αν οι πιθανότητες των συμβόλων δεν ταιριάζουν με αυτές που δημιουργήθηκε το δένδρο, τότε το αποτέλεσμα μπορεί να είναι από κακή συμπίεση μέχρι και μικρή διόγκωση του συνόλου δεδομένων σε εξαιρετικές περιπτώσεις. Από αυτές τις επιπτώσεις δεν πάσχει μόνο η κωδικοποίηση Fano αλλά κάθε κωδικοποίηση που μπορεί να αναπαρασταθεί με μία στατική δενδρική δομή.

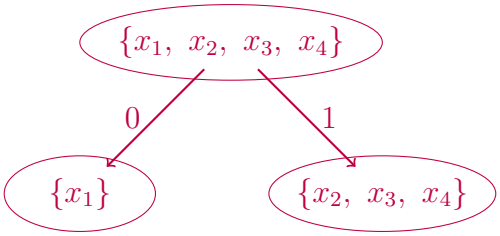
Παράδειγμα 5.1. *Έστω μία τυχαία μεταβλητή X με τιμές $\{s_1, s_2, s_3, s_4\}$ και συνάρτηση μάζας πιθανότητας $\{p_1 = 0.1, p_2 = 0.4, p_3 = 0.3, p_4 = 0.2\}$. Να κωδικοποιηθεί με βάση την κωδικοποίηση Fano.*

1. *Ταξινομούμε τις τιμές κατά φθίνουσα σειρά ως προς τις πιθανότητες εμφάνισης τους και δημιουργούμε ένα δένδρο που έχει ως ρίζα το σύνολο των ταξινομημένων τιμών*

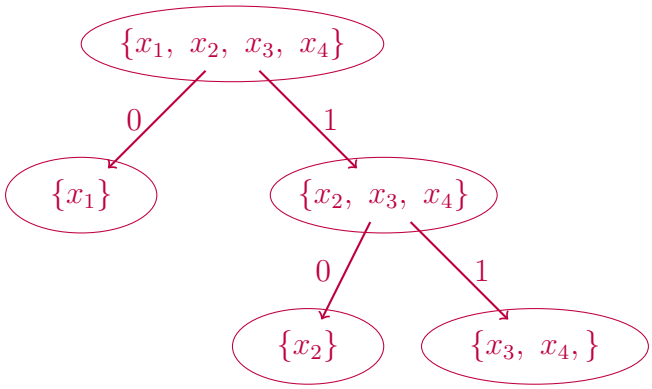
$$\{(s_2, 0.4), (s_3, 0.3), (s_4, 0.2), (s_1, 0.1)\} = \{x_1,\ x_2,\ x_3,\ x_4\}$$



2. *Χωρίζουμε το αρχικό σύνολο σε (σχεδόν)ισοπίθανα υποσύνολα βρίσκοντας τον μεγαλύτερο $|P_1^r - P_2^r| < |P_1^r - P_2^{r+1}|$. Το $r_0 = 1$*

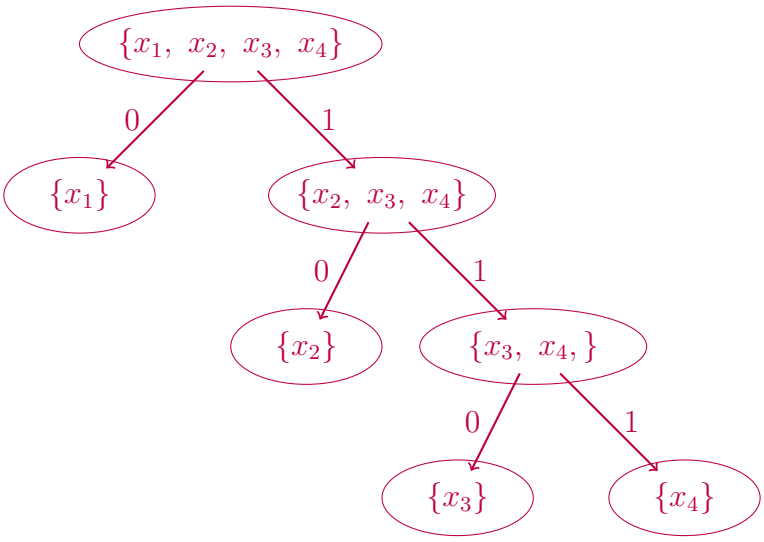


3. *Το δεξιό παιδί περιέχει μόνο ένα στοιχείο άρα η διαμέριση του έχει τελειώσει οπότε ο κώδικας που αντιστοιχεί στο $x_1 = (s_2,\ 0.4)$είναι το 0. Το αριστερό παιδί θα συνεχιστεί να διαμερίζεται αναδρομικά.*



Το δεξιό παιδί περιλαμβάνει μόνο ένα στοιχείο άρα ο κώδικας για το $x_2 = (s_3,\ 0.3)$ είναι το 10

4. *Το αριστερό παιδί συνεχίζει να περιλαμβάνει περισσότερα από ένα στοιχεία οπότε διαμερίζεται εκ νέου.*



Και τα δύο παιδιά περιλαμβάνουν από ένα στοιχείο άρα η κωδική λέξη που αντιστοιχεί στην $x_3 = (s_4,\ 0.2)$ είναι το 110 και για την $x_4 = (s_1,\ 0.1)$ είναι το 111

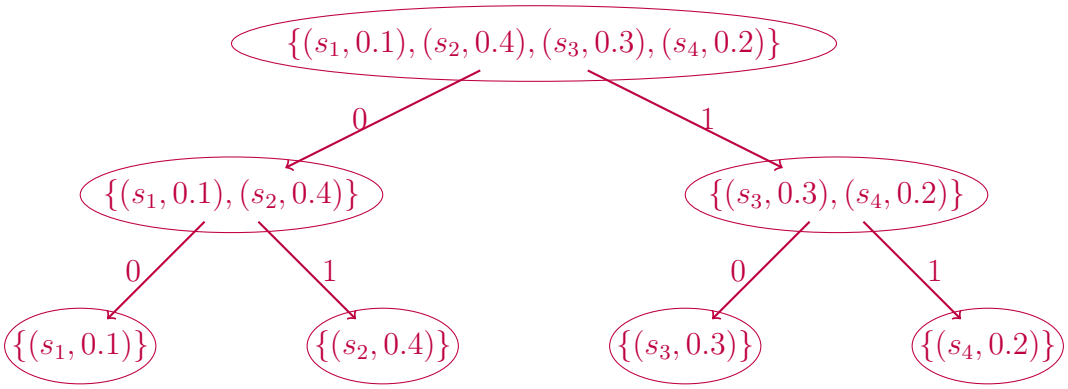
Άρα, $s_2 \to 0, s_3 \to 10, s_4 \to 110, s_1 \to 111$. Το μέσο μήκος κώδικα είναι $\overline{L} = \sum_{i=1}^{4} p_i \cdot l_i$ όπου p_i η πιθανότητα που αντιστοιχεί στο σύμβολο s_i, δηλαδή $p_i = Pr[s = s_i]$ και l_i είναι το μήκος της κωδικής λέξης που αντιστοιχεί στην τιμή s_i. Οπότε:

$$\overline{L} = \sum_{i=1}^{4} p_i \cdot l(s_1) = p_1 \cdot l(s_1) + p_2 \cdot l(s_2) p_3 \cdot l(s_3) + p_4 \cdot l(s_4) = 0.1 \cdot 3 + 0.4 \cdot 1 + 0.3 \cdot 2 + 0,2 \cdot 3 =$$
$$0.3 + 0.4 + 0.6 + 0.6 = 1.9 bits/\text{σύμβολο}$$

Αντίστοιχα αν υπολογίσουμε την εντροπία που περιέχεται στη συγκεκριμένη συνάρτηση μάζας πιθανότητα θα έχουμε:

$$H(p_1, p_2, p_3, p_4) = -\sum_{i=1}^{4} p_i \cdot log_2 p_i = -(p_1 \cdot log_2 p_1 + p_2 \cdot log_2 p_2 + p_3 \cdot log_2 p_3 + p_4 \cdot log_2 p_4) =$$
$$-\left(0.1 \cdot log_2 0.1 + 0.4 \cdot log_2 0.4 + 0.3 \cdot log_2 0.3 + 0.2 \cdot log_2 0.2\right) \approx 1.85 bits/\text{σύμβολο}.$$

Από ότι βλέπουμε το μέσο μήκος κώδικα που παράγεται από την κωδικοποίηση Fano είναι πολύ κοντά στη εντροπία, που όπως είπαμε στο κεφάλαιο της συμπίεσης αναπαριστά το ελάχιστο μέσο μήκος των κωδικών λέξεων ενός αλφαβήτου. Το κλειδί της επιτυχίας της κωδικοποίησης κατά Fano έγκειται στο πρώτο βήμα που γίνεται η διάταξη των συμβόλων κατά φθίνουσα πιθανότητα εμφάνισης. Ο Fano προφανώς είχε παρατηρήσει πως όταν έχουμε ένα πεπερασμένο σύνολο πιθανοτήτων είναι πιθανό να προκύψουν υποσύνολα με μεγάλες αποκλίσεις ως προς την συνολική πιθανότητα που κατέχει το κάθε ένα, γεγονός που καθιστά ανούσιο να μιλάμε για διαμέριση σε (σχεδόν) ισοπίθανα σύνολα. Για να αποφύγει λοιπόν το παραπάνω πρόβλημα αποφάσισε να διατάξει τις πιθανότητες έτσι ώστε στο κάθε υποσύνολο να βρίσκονται στοιχεία που έχουν κοντινά μεγέθη. Με αυτό τον τρόπο εξασφάλισε ότι και τα υποσύνολα που θα προκύψουν από τις τυχόν διαμερίσεις θα περιέχουν στοιχεία που θα επιτρέψουν την περεταίρω διαμέριση τους σε (σχεδόν) ισοπίθανα υποσύνολα. Για να καταλάβουμε καλύτερα την αναγκαιότητα και την σπουδαιότητα του πρώτου βήματος θα χωρίσουμε το αρχικό σύνολο πιθανοτήτων του παραδείγματος 5.1 με όποιον τρόπο θέλουμε χωρίς να έχει υπάρξει κάποια ταξινόμηση πριν.



1.

Οι κωδικές λέξεις των συμβόλων αυτή τη φορά θα είναι $s_1 \to 00, s_2 \to 01, s_3 \to 10, s_4 \to 11$. Σε αυτή την περίπτωση βλέπουμε ότι το μέσο μήκος κώδικα είναι 2 και απέχει περισσότερο από την εντροπία από ότι το μέσο μήκος του κώδικα που βρήκαμε με την κωδικοποίηση Fano. Ακόμη κατά τη διαμέριση του συνόλου $\{(s_1, 0.1), (s_2, 0.4)\}$ παρατηρούμε ότι τα δύο υποσύνολα που προκύπτουν διαφέρουν πολύ από το να είναι (σχεδόν) ισοπίθανα.

Διαδικασία αποκωδικοποίησης

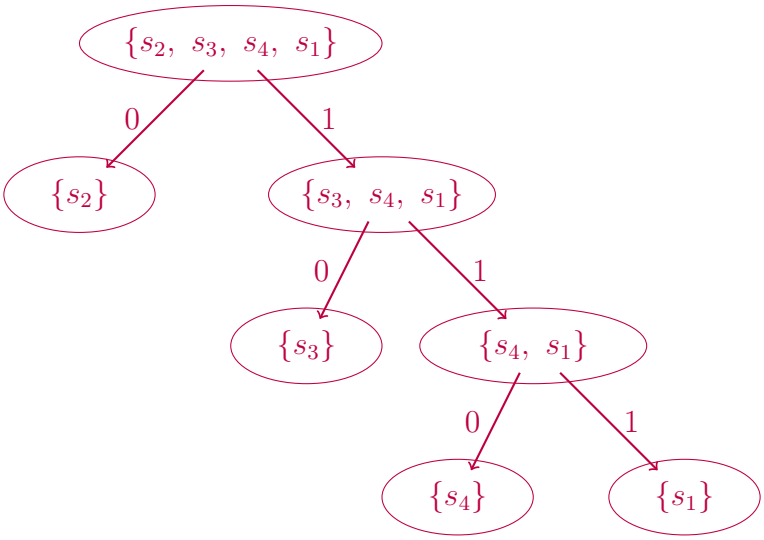
1. Όταν λάβουμε το πρώτο κωδικό σύμβολο πηγαίνουμε στη ρίζα του δένδρου και
   (α′) μεταβαίνουμε στο αριστερό υποδένδρο αν το κωδικό σύμβολο ήταν το 0 ή
   (β′) μεταβαίνουμε στο δεξί υποδένδρο αν το κωδικό σύμβολο ήταν το 1
2. Για κάθε επόμενο κωδικοποιημένο σύμβολο που συναντάμε από τον τρέχοντα κόμβο:
   (α′) Ακολουθούμε το αριστερό κλαδί αν το σύμβολο είναι 0
   (β′) Ακολουθούμε το δεξιό κλαδί αν το σύμβολο είναι 1
3. Αν φτάσουμε σε κάποιο φύλλο του δένδρου γράφουμε στην έξοδο του αποκωδικοποιητή το σύμβολο και μεταβαίνουμε πάλι στη ρίζα

Παράδειγμα 5.2. *Να κωδικοποιήσετε και να αποκωδικοποιήσετε την ακολουθία $s_3s_3s_1s_2s_2s_2$ σύμφωνα με τον κώδικα που παράχθηκε στο προηγούμενο παράδειγμα.*

Από το δένδρο κωδικοποίησης ξέρουμε:

Σύμβολο	*Κωδική λέξη*
s_1	*111*
s_2	*0*
s_3	*10*
s_4	*110*

Άρα $s_3s_3s_1s_2s_2s_2 \Rightarrow 1010111000$
Το δένδρο κωδικοποίησης είναι το:

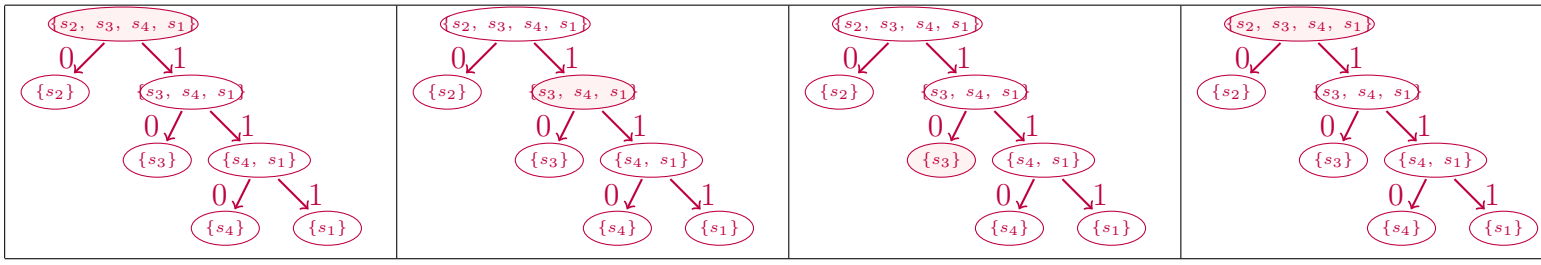


1. *(α′) Ξεκινάμε την αποκωδικοποίηση από τη ρίζα. Το πρώτο σύμβολο είναι το 1 οπότε μεταβαίνουμε στον δεξιό κόμβο.*

   (β′) Το επόμενο είναι το 0, οπότε μεταβαίνουμε από τον τρέχοντα κόμβο στο αριστερό παιδί, το οποίο αποτελεί φύλλο

   (γ′) Αντικαθιστούμε το 10 με το s_3 *και μεταβαίνουμε πάλι στην ρίζα.*

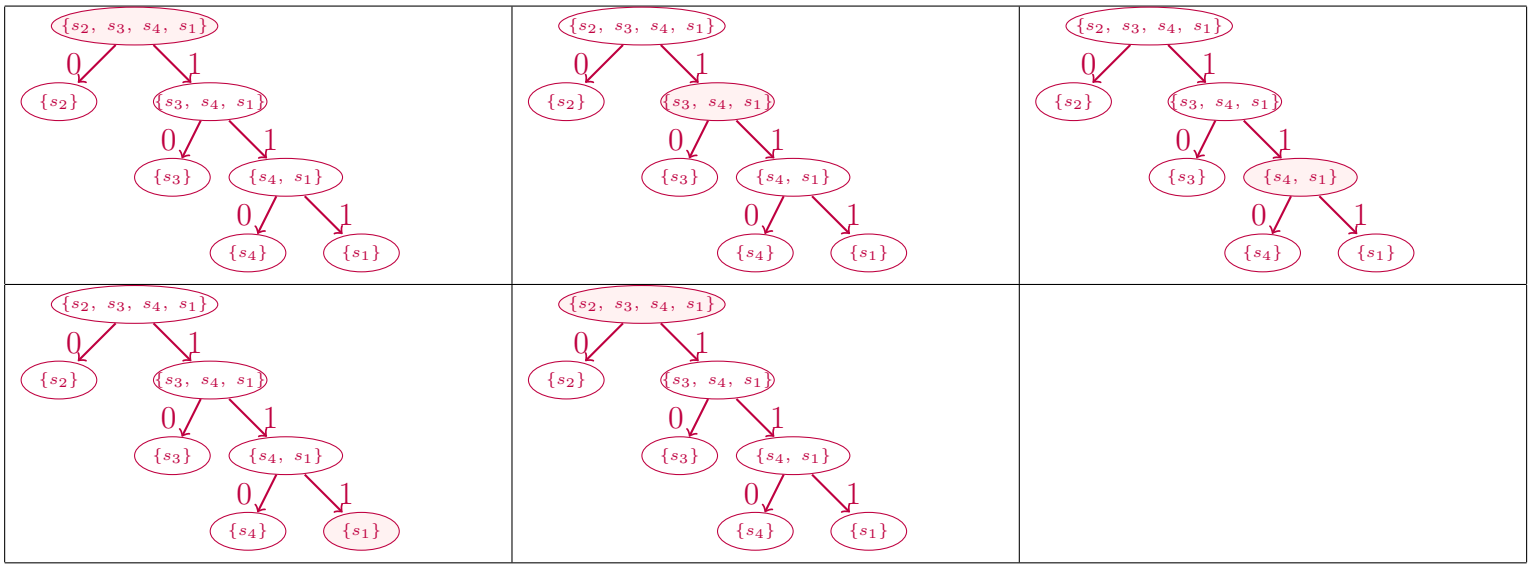
   Η αποκωδικοποίηση του πρώτου συμβόλου φαίνεται στο παρακάτω σχήμα:



Σχήμα 5.1

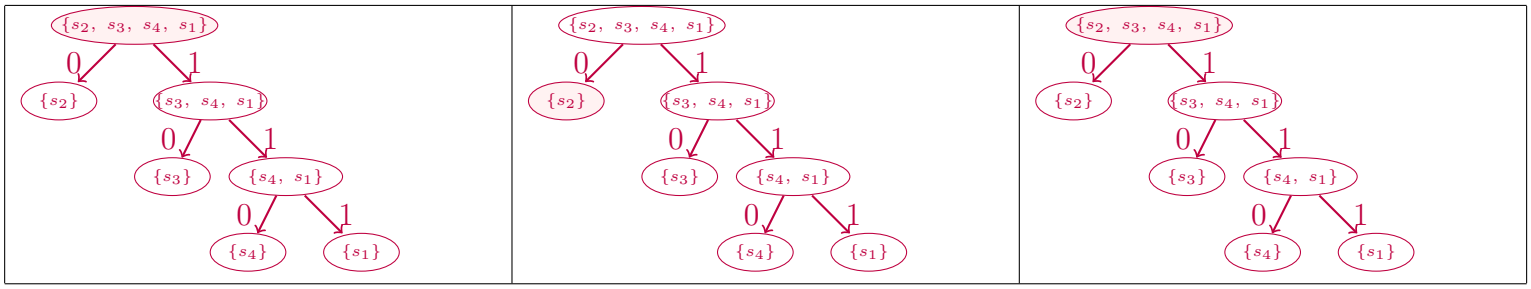
2. *Η διαδικασία συνεχίζει πανομοιότυπα με πριν, βλέπουμε το σύμβολο 1 μεταβαίνουμε στο δεξιό υποδένδρο και μετά βλέποντας το 0 μεταβαίνουμε στο αριστερό παιδί, το οποίο είναι φύλο. Αντικαθιστούμε το 10 με το* s_3 *και επανερχόμαστε στη ρίζα. Άρα* $s_3s_3s_1s_2s_2s_2 \Rightarrow s_3s_3111000$. *Το σχήμα αποκωδικοποίησης είναι ίδιο με το 5.2.*

3. $s_3s_3s_1s_2s_2s_2 \Rightarrow s_3s_3s_1000$.



Σχήμα 5.2

4. $s_3s_3s_1s_2s_2s_2 \Rightarrow s_3s_3s_1s_200$.



Σχήμα 5.3

5. $s_3s_3s_1s_2s_2s_2 \Rightarrow s_3s_3s_1s_2s_20$.

6. $s_3s_3s_1s_2s_2s_2 \Rightarrow s_3s_3s_1s_2s_2s_2$.

Στα βήματα 5 και 6 το σχήμα που αντιστοιχεί είναι ίδιο με το Σχήμα 5.3

5.2.1 Ανάλυση της κωδικοποίησης του Fano

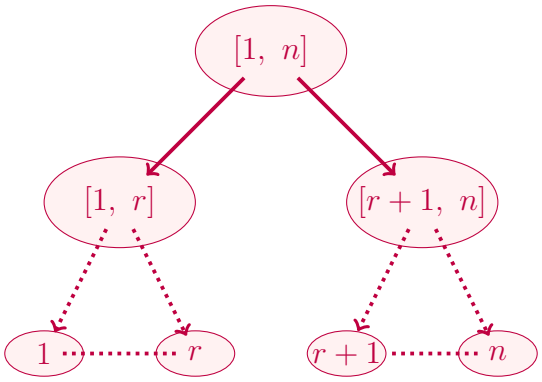
[2] Ήρθε η ώρα να εξετάσουμε πόσο "καλή" είναι η κωδικοποίηση του Fano. Η ανάλυση μας θα στηριχτεί στην ιδιότητα αθροιστικότητας της εντροπίας. Αν ανασύρουμε την συζήτηση που είχαμε στις αρχές του πρώτου κεφαλαίου για τον τρόπο με τον οποίο μπορούμε να κάνουμε μία επιλογή ανάμεσα σε N ενδεχόμενα, θα θυμηθούμε ότι η επιλογή μπορούσε να πραγματοποιηθεί σε ένα στάδιο δηλαδή να διαλέξουμε κάποιο από τα N ενδεχόμενα, είτε σε m στάδια. Για να έχουμε m στάδια έπρεπε να χωρίσουμε τις N επιλογές σε k σύνολα και μετά από m στάδια να καταλήξουμε σε συγκεκριμένο στοιχείο που ανήκει σε κάποιο από τα k σύνολα.

Αν είμαστε λίγο υποψιασμένοι θα καταλάβουμε ότι κάτι αντίστοιχο γίνεται κατά την κωδικοποίηση Fano, δηλαδή σε κάθε βήμα ο κωδικοποιητής επιλέγει ανάμεσα σε δύο σχεδόν ισοπίθανα σύνολα προκειμένου να φτάσει στην τελική απόφαση που είναι είτε η κωδικοποίηση ενός συμβόλου ή η αποκωδικοποίησή του.

Πριν συνεχίσουμε στο αποδεικτικό μέρος θα διευκρινίσουμε τους συμβολισμούς προκειμένου να εξασφαλιστεί η ομαλή διεξαγωγή της μελέτης μας. Επειδή μιλάμε για διακριτές τυχαίες μεταβλητές που παίρνουν

[2] Ολόκληρη η ανάλυση της παρούσας ενότητας βασίζεται στις δημοσιεύσεις [**horibe1977improved**] και [**rissanen1973bounds**]

πεπερασμένο πλήθος τιμών δηλαδή μπορούν να αριθμηθούν ώς $\{x_1,\ x_2, \cdots, x_n\}$ και να αντιστοιχηθούν στο διάστημα των φυσικών αριθμών $[1,\ n]$. Το δένδρο κωδικοποίησης θα συμβολίζεται με $T_{[1,n]}$ και η ρίζα με το διάστημα $[1,\ n]$. Επειδή κάθε εσωτερικός κόμβος του δένδρου επί της ουσίας αποτελεί ένα υποσύνολο του $\mathcal{X} = \{x_1,\ x_2, \cdots, x_n\}$ και κατά επέκταση ένα υποσύνολο του $[1,\ n]$ θα συμβολίζεται με $[i,\ j]$. Ο κόμβος $[i,\ j]$ θα περιέχει τις τιμές $\{x_i,\ x_{i+1}, \cdots, x_j\}$. Το υποδένδρο που έχει σαν ρίζα τον κόμβο $[i,\ j]$ και φύλλα του φυσικούς $i,\ i+1, \cdots, j$ που αντιστοιχούν στις τιμές $\{x_i,\ x_{i+1}, \cdots, x_j\}$ θα συμβολίζεται με $T_{[i,\ j]}$.

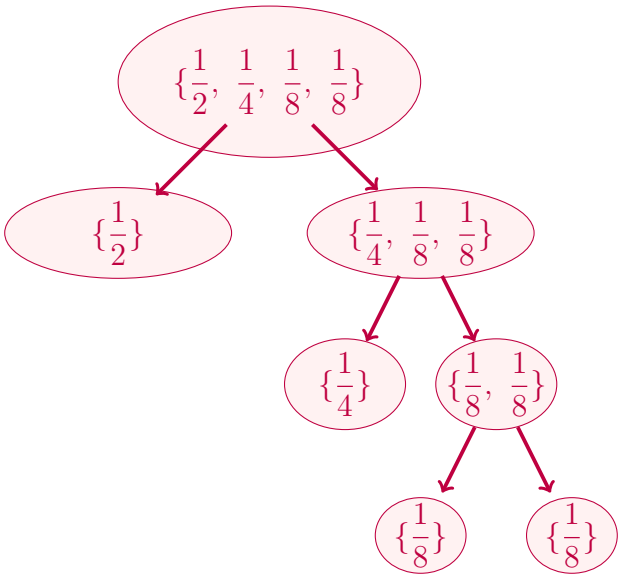


Σχήμα 5.4: Το δένδρο $T_{[1,\ n]}$ μαζί με τα υπόδεδνρα $T_{[1,\ r]}$ και $T_{[r+1,\ n]}$

Η κωδικοποίηση κατά Fano λειτουργεί αναδρομικά. Προκειμένου να βρούμε ένα άνω φράγμα της κωδικοποίησης σε σχέση με την εντροπία θα πρέπει να συγκρίνουμε το μέσο μήκος των μονοπατιών[3] που παράγεται κατά τη δημιουργία του δένδρου σε σχέση με αυτό που προτείνει η εντροπία.

Το δένδρο που παράγεται κατά τη διαδικασία κωδικοποίησης μπορεί να θεωρηθεί σαν ένας βεβαρημένος γράφος , όπου κάθε κόμβος έχει ως βάρος την συνολική πιθανότητα των τιμών που περιέχει. Τα φύλλα έχουν βάρος μηδέν. Για παράδειγμα στο σχήμα 5.5 το βάρος του κόμβου $[1,\ r]$ θα είναι ίσο με την $P_{[1,\ r]} = \sum_{i=1}^{r} Pr[X = x_i]$. Ένα κρίσιμο σημείο είναι να σκεφτούμε τους διαφορετικούς τρόπους με τους οποίους μπορούμε να μετρήσουμε το μέσο μήκος κώδικα ή αλλιώς το μέσο βάρος του δένδρου. Γνωρίζουμε ότι το μέσο μήκος των μονοπατιών δίνεται από την σχέση $\overline{L} = \sum_{i=1}^{n} Pr[X = x_i] \cdot l_i$. Αν παρατηρήσουμε την παραπάνω εξίσωση θα δούμε ότι ο όρος l_i δείχνει το πλήθος των διαμερίσεων που χρειάστηκαν για να συναντήσαμε τον κόμβο με πιθανότητα/βάρος $Pr[X = x_i]$. Στο Σχήμα 5.6 φαίνεται η διαμέριση της $P_X(x) = \{\frac{1}{2},\ \frac{1}{4},\ \frac{1}{8},\ \frac{1}{8}\}$.

[3]Το μέσο μήκος των μονοπατιών όπως έχουμε αναφέρει επανειλημμένα αποτελούν το μέσο μήκος του κώδικα. Στο θεωρητικό κεφάλαιο της συμπίεσης είδαμε ότι το μέσο μήκος κώδικα είναι μεγαλύτερο ή ίσο από την εντροπία. Άρα η σύγκριση μεταξύ του μέσου μήκους των μονοπατιών και της εντροπίας που γίνεται εδώ, αποτελεί το πλέον λογικό τρόπο για να συγκρίνουμε την αποτελεσματικότητα μίας μεθόδου σε σχέση με την συμπίεση που προσφέρει.



Σχήμα 5.5: Το δένδρο κωδικοποίησης Fano για την $P_X(x) = \{\frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \frac{1}{8}\}$

Το μέσο μήκος για το συγκεκριμένο δένδρο θα είναι:

$$\overline{L} = \frac{1}{2} \cdot 1 + \frac{1}{4} \cdot 2 + \frac{1}{8} \cdot 3 + \frac{1}{8} \cdot 3$$

Οι ακέραιοι 1,2 και 3 δηλώνουν ακριβώς αυτό που είχαμε προαναφέρει, το μήκος του μονοπατιού l_i μετράει το πλήθος των κόμβων που συναντήσαμε την πιθανότητα $Pr[X = x_i]$. Ένα σημείο ακόμη που αξίζει να δώσουμε βάση είναι ο αριθμός των εσωτερικών κόμβων που πρέπει να διασχίσουμε προκειμένου να προκύψει μήκος l_i. Συγκεκριμένα για να προκύψει μήκος 2 πρέπει να έχουμε περάσει από δύο εσωτερικούς κόμβους[4], για μήκος 3 από 3 εσωτερικούς κόμβους κ.ο.κ. Άρα καταλαβαίνουμε ότι το πλήθος των φορών που θα συναντήσουμε την πιθανότητα $Pr[X = x_i]$ εξαρτάται από το πλήθος των εσωτερικών κόμβων που την περιέχουν. Οπότε ένας άλλος τρόπος να μετρήσουμε το μέσο μήκος/βάρος του δένδρου είναι να αθροίσουμε τα βάρη των εσωτερικών κόμβων. Έτσι έχουμε:

$$\overline{W}_{[1,\ n]} = P_{[1,\ 4]} + P_{[2,\ 4]} + P_{[3,\ 4]} = (Pr[X = x_1] + Pr[X = x_2] + Pr[X = x_3] + Pr[X = x_4]) +$$
$$(Pr[X = x_2] + Pr[X = x_3] + Pr[X = x_4]) + (Pr[X = x_3] + Pr[X = x_4]) =$$
$$(\frac{1}{2} + \frac{1}{4} + \frac{1}{8} + \frac{1}{8}) + (\frac{1}{4} + \frac{1}{8} + \frac{1}{8}) + (\frac{1}{8} + \frac{1}{8}) =$$
$$\frac{1}{2} \cdot 1 + \frac{1}{4} \cdot 2 + \frac{1}{8} \cdot 3 + \frac{1}{8} \cdot 3 = \overline{L}$$

Αν $\mathcal{I}$ το σύνολο των εσωτερικών κόμβων, τότε το μέσο μήκος/βάρος γράφεται ώς:

$$\boxed{\overline{W}_{[i,\ n]} = \sum_{[i,\ j] \in \mathcal{I}} P_{[i,\ j]}}$$

Γενικά εναλλάσσουμε συχνά τους όρους μέσο μήκος και μέσο βάρος γιατί στη συγκεκριμένη περίπτωση είναι το ίδιο πράγμα. Απλά κάθε φορά επιλέγεται ο όρος που θα κάνει την ανάλυση μας διαισθητικά πιο κατανοητή. Επειδή το μέσο βάρος των φύλλων είναι μηδέν μπορούμε να βρούμε έναν αναδρομικό τρόπο προκειμένου να υπολογίσουμε το μέσο μήκος/βάρος. Για παράδειγμα το μέσο μήκος ενός δένδρου $T_{[1,\ n]}$ μπορεί να υπολογιστεί ως το άθροισμα των μέσων μηκών του δεξιού και αριστερού υποδένδρου σταθμισμένα

[4]Στους εσωτερικούς κόμβους συγκαταλέγεται και η ρίζα στην παρούσα σύμβαση

με την ανάλογη πιθανότητα συν 1, που εκφράζει την μετάβαση από την ρίζα είτε στο δεξιό είτε στο αριστερό υποδένδρο, δηλαδή

$$\overline{W}_{[1,\ n]} = 1 + P_{[1,\ r]}\overline{W}_{[1,\ r]} + P_{[r+1,\ n]}\overline{W}_{[r+1,\ n]}$$

Με τη σειρά του, το μέσο μήκος $\overline{W}_{[1,\ r]}$ μπορεί να αναλυθεί στο μέσο μήκος του αριστερού και δεξιού του υποδένδρου σταθμισμένα με την κατάλληλη πιθανότητα συν το βήμα της επιλογή υποδένδρου, δηλαδή 1.

$$\overline{W}_{[1,\ r]} = 1 + \frac{P_{[1,\ k]}}{P_{[1,\ r]}}\overline{W}_{[1,\ k]} + \frac{P_{[k+1,\ r]}}{P_{[1,\ r]}}\overline{W}_{[k+1,\ r]}$$

Η πιθανότητα με την οποία σταθμίζουμε τα μέσα μήκη των υποδένδρων αποτελεί μία κανονικοποίηση των βαρών τους. Η παραπάνω κανονικοποίηση είναι ισοδύναμη με τον υπολογισμό της δεσμευμένης πιθανότητας $Pr\{[i,\ r]|[i,\ j]\}$ και $Pr\{[r+1,\ j]|[i,\ j]\}$ αντίστοιχα. Από τα παραπάνω συνάγουμε ότι η αναδρομική μορφή για τον υπολογισμό του μέσου μήκους ενός υποδένδρου με ρίζα τον κόμβο $[i,\ j]$ θα είναι:

$$\boxed{\overline{W}_{[i,\ j]} = 1 + \frac{P_{[i,\ k]}}{P_{[i,\ j]}}\overline{W}_{[1,\ k]} + \frac{P_{[k+1,\ j]}}{P_{[i,\ j]}}\overline{W}_{[k+1,\ j]}}$$

Από την ιδιότητα αθροιστικότητας της εντροπίας γνωρίζουμε ότι η πληροφορία που κρύβεται στην επιλογή ενός ενδεχομένου από ένα σύνολο N, ισοδυναμεί με το άθροισμα της πληροφορίας που υπάρχει σε m επιλογές αν τα N ενδεχόμενα οργανωθούν σε k σύνολα. Για παράδειγμα αν $H(\frac{1}{2},\ \frac{1}{4},\ \frac{1}{8},\ \frac{1}{8})$ η εντροπία της $P_X(x) = \{\frac{1}{2},\ \frac{1}{4},\ \frac{1}{8},\ \frac{1}{8}\}$, τότε αν οι πιθανότητες οργανωθούν στα δύο ισοπίθανα σύνολα $\{\frac{1}{2}\}$ και $\{\frac{1}{4},\ \frac{1}{8},\ \frac{1}{8}\}$ η εντροπία της $P_X(x)$ θα ισούται με την εντροπία του κάθε συνόλου σταθμισμένο με την πιθανότητα επιλογής του συν την εντροπία της αρχικής επιλογής. Δηλαδή:

$$\begin{aligned} &H\left(\frac{1}{2},\ \frac{1}{4},\ \frac{1}{8},\ \frac{1}{8}\right) = H(P_{[1,\ 1]},\ P_{[2,\ 4]}) + P_{[1,\ 1]} \cdot H_{[1,\ 1]} + P_{[2,\ 4]} \cdot H_{[2,\ 4]} = \\ &H\left(\frac{1}{2}, \frac{1}{2}\right) + P_{[1,\ 1]} \cdot 0 + P_{[2,\ 4]} \cdot H_{[2,\ 4]} \end{aligned} \tag{5.1}$$

Προφανώς η εντροπία μίας πιθανότητας είναι 0, άρα $H_{[1,\ 1]} = 0$.Το $H_{[2,\ 4]}$ μπορεί να αναλυθεί αναδρομικά και πάλι σε

$$\begin{aligned} &H_{[2,\ 4]} = H\left(\frac{P[2,2]}{P[2,4]},\ \frac{P[3,4]}{P[2,4]}\right) + \frac{P[2,2]}{P[2,4]} \cdot H_{[2,\ 2]} + \frac{P[3,4]}{P[2,4]} \cdot H_{[3,\ 4]} = \\ &H\left(\frac{1}{2}, \frac{1}{2}\right) + \frac{Pr[2,\ 2]}{P[2,4]} \cdot 0 + \frac{Pr[3,\ 4]}{P[2,4]} \cdot H_{[3,\ 4]} \end{aligned} \tag{5.2}$$

Το $H_{[3,\ 4]}$ με τη σειρά του αναλύεται αναδρομικά σε:

$$\begin{aligned} &H_{[3,\ 4]} = H\left(\frac{Pr[X = x_3]}{P[3,\ 4]},\ \frac{Pr[X = x_4]}{P[3,\ 4]}\right) + \frac{Pr[X = x_3]}{P[3,4]} \cdot H_{[3,\ 3]} + \frac{Pr[X = x_4]}{P[3,4]} \cdot H_{[4,\ 4]} = \\ &H\left(\frac{1}{2}, \frac{1}{2}\right) + \frac{Pr[X = x_3]}{P[3,4]} \cdot 0 + \frac{Pr[X = x_4]}{P[3,4]} \cdot 0 = 1 \end{aligned} \tag{5.3}$$

Αντικαθιστώντας την (5.4) στην (5.3) έχουμε:

$$\begin{aligned} &H_{[2,\ 4]} = H\left(\frac{P[2,2]}{P[2,4]},\ \frac{P[3,4]}{P[2,4]}\right) + \frac{Pr[3,\ 4]}{P[2,4]} \cdot H_{[3,\ 4]} \Rightarrow \\ &H_{[2,\ 4]} = H\left(\frac{P[2,2]}{P[2,4]},\ \frac{P[3,4]}{P[2,4]}\right) + \frac{Pr[3,\ 4]}{P[2,4]} \cdot H\left(\frac{Pr[X = x_3]}{P[3,\ 4]},\ \frac{Pr[X = x_4]}{P[3,\ 4]}\right) \end{aligned} \tag{5.4}$$

Αντικαθιστώντας την (5.5) στην (5.2) έχουμε:

$$H_{[1,\ 4]} = H(P_{[1,\ 1]},\ P_{[2,\ 4]}) + P_{[2,\ 4]} \cdot H_{[2,\ 4]} \Rightarrow$$
$$H_{[1,\ 4]} = H(P_{[1,\ 1]},\ P_{[2,\ 4]}) + P_{[2,\ 4]} \cdot \left(H\left(\frac{P[2,2]}{P[2,4]},\ \frac{P[3,4]}{P[2,4]}\right) + \frac{Pr[3,\ 4]}{P[2,4]} \cdot H\left(\frac{Pr[X=x_3]}{P[3,\ 4]},\ \frac{Pr[X=x_4]}{P[3,\ 4]}\right)\right) \Rightarrow$$
$$H_{[1,\ 4]} = H(P_{[1,\ 1]},\ P_{[2,\ 4]}) + P_{[2,\ 4]} \cdot H\left(\frac{P[2,2]}{P[2,4]},\ \frac{P[3,4]}{P[2,4]}\right) + P_{[3,\ 4]} \cdot H\left(\frac{Pr[X=x_3]}{P[3,\ 4]},\ \frac{Pr[X=x_4]}{P[3,\ 4]}\right) \Rightarrow$$
$$H_{[1,\ 4]} = \sum_{[i,\ j]\in\mathcal{I}} P_{[i,\ j]} \cdot H\left(\frac{P_{[i,\ r]}}{P_{[i,\ j]}},\ \frac{P_{[r+1,\ j]}}{P_{[i,\ j]}}\right)$$

Λόγω της εξίσωσης (5.4) ξέρουμε ότι $H_{[3,\ 4]} = 1$. Αντικαθιστώντας στην (5.3) έχουμε:
$H_{[2,4]} = 1 + 0 + \frac{1}{2} \cdot 1 = \frac{3}{2}$. Αφού βρήκαμε το $H_{[2,4]}$ υπολογίζουμε από την (5.2) το:
$H_{[1,4]} = 1 + 0 + \frac{1}{2} \cdot \frac{3}{2} = 1 + \frac{3}{4} = 1.75$.

Καταλαβαίνουμε λοιπόν ότι για να υπολογίσουμε την εντροπία ενός υποσυνόλου $\{Pr[X = x_i], \cdots, , Pr[X = x_j]\}$ της $P_X(x)$ αρκεί να ακολουθήσουμε τον αναδρομικό τύπο

$$H_{[i,\ j]} = H(\frac{P_[i,r]}{P_[i,j]},\ \frac{P_[r+1,j]}{P_[i,j]}) + \frac{P_[i,r]}{P_[i,j]} \cdot H_{[i,\ r]} + \frac{P_[r+1,j]}{P_[i,j]} \cdot H_{[r+1,\ j]},\ \text{με } H_{[i,\ i]} = 0.$$

και για να υπολογίσουμε την εντροπία ολόκληρου του $P_X(x)$ αρκεί να υπολογίσουμε το:

$$\boxed{H_{[1,\ n]} = \sum_{[i,\ j]\in\mathcal{I}} P[i,\ j] \cdot H\left(\frac{P[i,\ r]}{P[i,\ j]},\ \frac{P[r+1,\ j]}{P[i,\ j]}\right)}$$

Από τον τρόπο που ορίστηκε το $\overline{L} = \overline{W}_{[1,\ n]}$ και το $H_{[1,\ n]}$ έχουμε

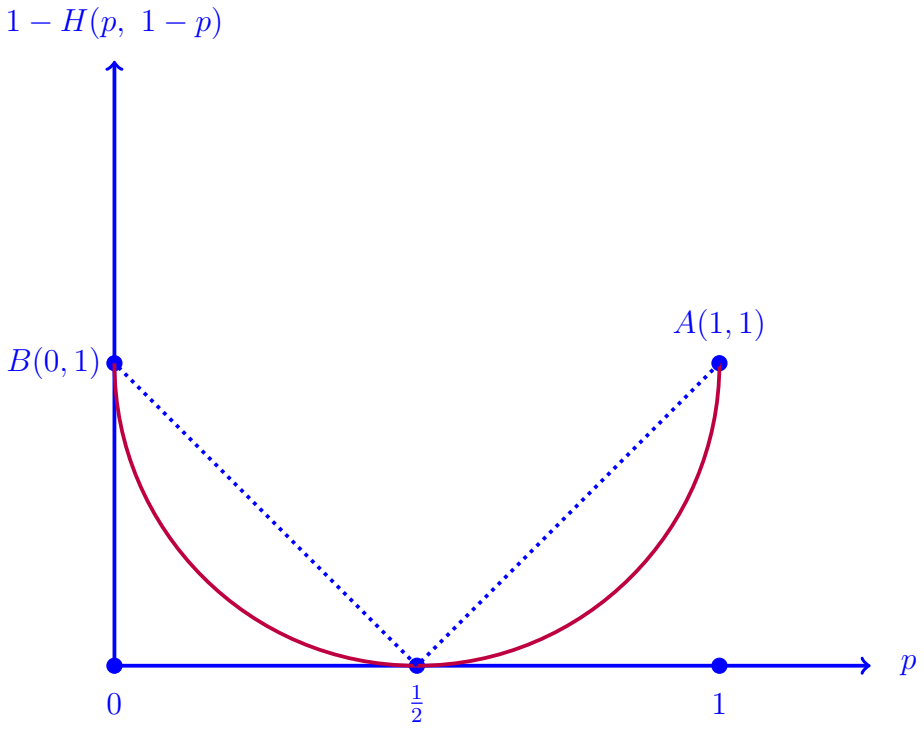
$$\boxed{\overline{L} - H = \overline{W}_{[1,\ n]} - H_{[1,\ n]} = \sum_{[i,\ j]\in\mathcal{I}} P_{[i,\ i]} \cdot \left(1 - H\left(\frac{P[i,\ r]}{P[i,\ j]},\ \frac{P[r+1,\ j]}{P[i,\ j]}\right)\right)} \tag{5.5}$$

Αν σε κάθε βήμα τα σύνολα $[i,\ r]$ και $[r{+}1,\ j]$ ήταν ισοπίθανα, τότε η εντροπία $H\left(\frac{P[i,\ r]}{P[i,\ j]},\ \frac{P[r+1,\ j]}{P[i,\ j]}\right) = H\left(\frac{1}{2},\ \frac{1}{2}\right) = 1 \Rightarrow 1 - H\left(\frac{P[i,\ r]}{P[i,\ j]},\ \frac{P[r+1,\ j]}{P[i,\ j]}\right) = 0 \Rightarrow \sum_{[i,\ j]\in\mathcal{I}} P_{[i,\ i]} \cdot \left(1 - H\left(\frac{P[i,\ r]}{P[i,\ j]},\ \frac{P[r+1,\ j]}{P[i,\ j]}\right)\right) = 0 \Rightarrow \overline{L} = H$ και η κωδικοποίηση θα ήταν βέλτιστη. Επειδή όμως πρακτικά κάτι τέτοιο δεν είναι δυνατόν να συμβεί με μεγάλη πιθανότητα θα προσπαθήσουμε να φράξουμε την ποσότητα $1 - H\left(\frac{P[i,\ r]}{P[i,\ j]},\ \frac{P[r+1,\ j]}{P[i,\ j]}\right)$.

Για την συνάρτηση $1 - H(p,\ 1-p)$ που είναι συνεχής και παραγωγίσιμη για κάθε p ισχύουν τα παρακάτω:

$$1 - H(p,\ 1-p) = 1 + p \cdot logp + (1-p) \cdot log(p-1)$$
$$\frac{d(1 - H(p,\ 1-p))}{dp} = logp + 1 - log(1-p) - 1 = log(\frac{p}{1-p})$$
$$\frac{d^2(1 - H(p,\ 1-p))}{dp^2} = \frac{1-p}{p} \cdot \frac{1}{(1-p)^2} = \frac{1-p}{p} > 0$$

Από την πρώτη παράγωγο βλέπουμε ότι η $1 - H(p,\ 1-p)$ είναι φθίνουσα για $0 \le p \le \frac{1}{2}$ και αύξουσα για $\frac{1}{2} < p \le 1$, δηλαδή παρουσιάζει ελάχιστο για $p = \frac{1}{2}$ που τα ενδεχόμενα είναι ισοπίθανα. Από την δεύτερη παράγωγο βλέπουμε ότι είναι κυρτή. Η μορφή της φαίνεται στο παρακάτω σχήμα:



Σχήμα 5.6: Η γραφική παράσταση της συνάρτησης $1-H(p,\ 1-p)$. Οι διακεκομμένες γραμμές δηλώνουν τις ευθείες $-2\cdot p+1$ και $2\cdot p+1$

Η ευθεία που ενώνει τα σημεία $(0,\ 1)$ και $(\frac{1}{2},\ 0)$ είναι η: $-2\cdot p+1$. Αντίστοιχα η ευθεία που ενώνει τα $(1,\ 1)$και $(\frac{1}{2},\ 0)$ είναι η: $2\cdot p-1$. Επειδή η ευθείες βρίσκονται πάνω από τη γραφική παράσταση της συνάρτησης έπεται ότι:

$$\left.\begin{array}{l} 1-H(p,\ 1-p)\leq -2p+1 \forall\ 0\leq p<\frac{1}{2} \\ 1-H(p,\ 1-p)\leq 2p-1 \forall\ \frac{1}{2}\leq p\leq 1 \end{array}\right\}. \Rightarrow 1-H(p,\ 1-p)\leq 2|p-\frac{1}{2}|\ \forall 0\leq p\leq 1$$

Αφού λοιπόν ισχύει η παραπάνω ανισότητα, ο όρος $1-H\left(\frac{P[i,\ r]}{P[i,\ j]},\ \frac{P[r+1,\ j]}{P[i,\ j]}\right)$ στην εξίσωση (5.6) φράσσεται από την ποσότητα:

$$2\cdot\left|\frac{P_{[i,\ r]}}{P_{[i,j]}}-\frac{1}{2}\right|=\left|2\cdot\frac{P_{[i,\ r]}}{P_{[i,j]}}-1\right|=\left|\frac{P_{[i,\ r]}}{P_{[i,j]}}+\frac{P_{[i,\ r]}}{P_{[i,j]}}-1\right|=\left|\frac{P_{[i,\ r]}}{P_{[i,j]}}-(1-\frac{P_{[i,\ r]}}{P_{[i,j]}})\right|=\left|\frac{P_{[i,\ r]}}{P_{[i,j]}}-\frac{P_{[r+1,\ j]}}{P_{[i,j]}}\right|$$

Οπότε το φράγμα ολοκληρωμένα γράφεται ως:

$$\boxed{1-H\left(\frac{P[i,\ r]}{P[i,\ j]},\ \frac{P[r+1,\ j]}{P[i,\ j]}\right)\leq\left|\frac{P_{[i,\ r]}}{P_{[i,j]}}-\frac{P_{[r+1,\ j]}}{P_{[i,j]}}\right|}$$

Από την τελευταία σχέση έχουμε:

$$1 - H\left(\frac{P[i,\ r]}{P[i,\ j]},\ \frac{P[r+1,\ j]}{P[i,\ j]}\right) \leq \left|\frac{P_{[i,\ r]}}{P_{[i,j]}} - \frac{P_{[r+1,\ j]}}{P_{[i,j]}}\right| \Rightarrow$$

$$P_{[i,\ j]} \cdot \left(1 - H\left(\frac{P[i,\ r]}{P[i,\ j]},\ \frac{P[r+1,\ j]}{P[i,\ j]}\right)\right) \leq P_{[i,\ j]} \cdot \left(\left|\frac{P_{[i,\ r]}}{P_{[i,j]}} - \frac{P_{[r+1,\ j]}}{P_{[i,j]}}\right|\right) \Rightarrow$$

$$P_{[i,\ j]} \cdot \left(1 - H\left(\frac{P[i,\ r]}{P[i,\ j]},\ \frac{P[r+1,\ j]}{P[i,\ j]}\right)\right) \leq |P_{[i,\ r]} - P_{[r+1,\ j]}| \Rightarrow$$

$$\sum_{[i,\ j]\in\mathcal{I}} P_{[i,\ j]} \cdot \left(1 - H\left(\frac{P[i,\ r]}{P[i,\ j]},\ \frac{P[r+1,\ j]}{P[i,\ j]}\right)\right) \leq \sum_{[i,\ j]\in\mathcal{I}} \left|P_{[i,\ r]} - P_{[r+1,\ j]}\right| \Rightarrow$$

$$\boxed{\overline{L} - H \leq \sum_{[i,\ j]\in\mathcal{I}} |P_{[i,\ r]} - P_{[r+1,\ j]}|}$$

Επειδή με την διαδικασία Fano ο κάθε εσωτερικό κόμβος διαμερίζεται σε δύο σύνολα που είναι προσεγγιστικά ισοπίθανα, έπεται ότι το r στο οποίο διαμερίζεται το σύνολο του κόμβου θα είναι αυτό που δίνει την ελάχιστη διαφορά $|P_{[i,\ r]} - P_{[r+1,\ j]}|$. Άρα $\Delta_{[i,\ j]} = |P_{[i,\ r]} - P_{[r+1,\ j]}| = \min_{i\leq k<j} |P_{[i,\ k]} - P_{[k+1,\ j]}|$. Το τελικό βήμα για να ολοκληρωθεί η εύρεση του φράγματος για τη κωδικοποίηση Fano, είναι να προσπαθήσουμε να βρούμε ένα άνω φράγμα για την ποσότητα $\Delta_{[i,\ j]}$. Ας εξετάσουμε τι συμβαίνει κατά τη διάρκεια μίας διαμέρισης του εσωτερικού κόμβου $[i,\ j]$.

1. Έστω $\Delta_{[i,\ j]} = |P_{[i,\ r]} - P_{[r+1,\ j]}|$ και $P_{[i,\ r]} \geq P_{[r+1,\ j]}\ \forall i < r < j$, τότε αφού το $\Delta_{[i,\ j]}$ εκφράζει την ελάχιστη διαφορά έπεται ότι:

   (α′) $P_{[i,\ r-1]} < P_{[r,\ j]}$ και

   (β′) $|P_{[i,\ r-1]} - P_{[r,\ j]}| > |P_{[i,\ r]} - P_{[r+1,\ j]}|$
   Από τις δύο αυτές διαπιστώσεις έπεται ότι το:

   $$\Delta_{[i,\ j]} \leq P_{[r,\ j]} - P_{[i,\ r-1]} \Rightarrow \Delta_{[i,\ j]} \leq Pr[X = x_r] + P_{[r+1,\ j]} - (P_{[i,\ r]} - Pr[X = x_r]) \Rightarrow$$
   $$\Delta_{[i,\ j]} \leq 2Pr[X = x_r] - \Delta_{[i,\ j]} \Rightarrow \boxed{\Delta_{[i,\ j]} \leq Pr[X = x_r]}$$

   (γ′) Στην περίπτωση που η διαμέριση γίνει χρησιμοποιώντας το πρώτο σύμβολο x_i του κόμβου $[i,\ j]$, δηλαδή $r = i$ έχουμε:

   $$\Delta_{[i,\ j]} = Pr[X = x_r] - P_{[r+1,\ j]} \leq Pr[X = x_r] - \min_{1\leq k\leq n} Pr[X = x_k] < Pr[X = x_k] \Rightarrow$$
   $$\boxed{\Delta_{[i,\ j]} < Pr[X = x_k]} \text{ και } \boxed{\Delta_{[i,\ j]} \leq Pr[X = x_r] - \min_{1\leq k\leq n} Pr[X = x_k]}$$

2. Έστω $\Delta_{[i,\ j]} = |P_{[i,\ r]} - P_{[r+1,\ j]}|$ και $P_{[i,\ r]} < P_{[r+1,\ j]}\ \forall i < r < j$, τότε αφού το $\Delta_{[i,\ j]}$ εκφράζει την ελάχιστη διαφορά έπεται ότι:

   (α′) $P_{[i,\ r+1]} \geq P_{[r+2,\ j]}$ και

   (β′) $|P_{[i,\ r]} - P_{[r+1,\ j]}| < |P_{[r+2,\ j]} - P_{[i,\ r+1]}|$

   $$\Delta_{[i,\ j]} \leq P_{[i,r+1]} - P_{[r+2,\ j]} \Rightarrow P_{[i,\ r]} + Pr[X = x_{r+1}] - (P_{[r+1,\ j]} - Pr[X = x_{r+1}]) \Rightarrow$$
   $$\Delta_{[i,\ j]} \leq -\Delta_{[i,\ j]} - 2 \cdot Pr[X = x_{r+1}] \Rightarrow \boxed{\Delta_{[i,\ j]} \leq Pr[X = x_{r+1}]}$$

   (γ′) Στην περίπτωση που η διαμέριση γίνει στο j θα έχουμε για $r + 1 = j$:

   $$\Delta_{[i,\ j]} = Pr[X = x_{r+1}] - P_{[i,\ r]} \leq Pr[X = x_{r+1}] - \min_{1\leq k\leq n} Pr[X = x_k] < Pr[X = x_{r+1}]$$
   $$\Rightarrow \boxed{\Delta_{[i,\ j]} \leq Pr[X = x_{r+1}] - \min_{1\leq k\leq n} Pr[X = x_k]}$$

Άρα σε κάθε εσωτερικό κόμβο η ποσότητα $\Delta_{[i,\ j]}$ φράσσεται είτε από το $Pr[X = x_r]$ ή από $Pr[X = x_{r+1}]$. Στις περιπτώσεις που η διαμέριση γίνεται στα άκρα του διαστήματος το $\Delta_{[i,\ j]}$ φράσσεται από τις ποσότητες $Pr[X = x_r] - \min_{1\leq k\leq n} Pr[X = x_k]$ και $Pr[X = x_{r+1}] - \min_{1\leq k\leq n} Pr[X = x_k]$. Άρα για το δέντρο $T_{[1,\ n]}$ που περιέχει $n-1$ εσωτερικούς κόμβους θα ισχύει

$$\boxed{\sum_{[i,\ j]\in\mathcal{I}} \Delta_{[i,\ j]} \leq \sum_{k=1}^{n-1} \max\{Pr[X = x_k],\ Pr[X = x_{k+1}]\}}$$

Επειδή όμως κάποια στιγμή κατά τη διαδικασία κατασκευής του δένδρου θα βρεθούμε σε μία από τις περιπτώσεις που η διαμέριση θα γίνει είτε στο πρώτο ή στο τελευταίο στοιχείο του κόμβου θα εμφανιστεί μέσα στο άθροισμα ένα όρος της μορφής $Pr[X = x_r] - \min_{1\leq k\leq n} Pr[X = x_k]$ ή $Pr[X = x_{r+1}] - \min_{1\leq k\leq n} Pr[X = x_k]$, άρα

$$\boxed{\sum_{[i,\ j]\in\mathcal{I}} \Delta_{[i,\ j]} \leq \sum_{k=1}^{n-1} \max\{Pr[X = x_k],\ Pr[X = x_{k+1}]\} - \min_{1\leq l\leq n} Pr[X = x_l]}$$

Ολοκληρώνοντας την ανάλυση διαπιστώνουμε πως το άνω φράγμα για την κωδικοποίηση Fano δίνεται από την σχέση

$$\boxed{\overline{L} - H \leq \sum_{[i,\ j]\in\mathcal{I}} |P_{[i,\ r]} - P_{[r+1,\ j]}|} \Rightarrow$$

$$\boxed{\overline{L} - H \leq \sum_{k=1}^{n-1} \max\{Pr[X = x_k],\ Pr[X = x_{k+1}]\} - \min_{1\leq l\leq n} Pr[X = x_l]}$$

5.3 Η μέθοδος κωδικοποίησης του Shannon

5.3.1 Παρουσίαση της Μεθόδου

Διαδικασία Κωδικοποίησης Για να κωδικοποιήσουμε μία τυχαία μεταβλητή X με συνάρτηση μάζας πιθανότητας $\{p_1, p_2, \cdots p_n\}$ χρησιμοποιώντας την κωδικοποίηση Shannon ακολουθούμε τα εξής βήματα:

1. Ταξινομούμε τα σύμβολα κατά φθίνουσα σειρά σύμφωνα με τις πιθανότητες εμφάνισης τους.

2. Υπολογίζουμε τις αθροιστικές πιθανότητες $P_k = \sum_{i<k} p_i = \sum_{i=1}^{k-1} p_k$, όπου p_k, η πιθανότητα εμφάνισης του συμβόλου s_k που ανήκει σε ένα αλφάβητο $\mathcal{A} = \{s_1, \cdots s_k, \cdots s_n\}$.

3. Έπειτα γράφουμε το δυαδικό ανάπτυγμα των P_k. Επειδή το δυαδικό ανάπτυγμα ενός αριθμού $x \in [0, 1]$ μπορεί να αποτελείται από άπειρους όρους χρησιμοποιούμε μόνο τα l_k πρώτα στοιχεία του αναπτύγματος έτσι ώστε $log_2\frac{1}{p_k} \leq l_k < log_2\frac{1}{p_k} + 1$, δηλαδή ορίζουμε ότι το μήκος $l_k = \lceil log_2\frac{1}{p_k}\rceil$.

Σημείωση: Η μέθοδος δουλεύει ακόμα και όταν το κωδικό αλφάβητο έχει μέγεθος μεγαλύτερο του 2. Οι μονές αλλαγές που πρέπει να πράξουμε είναι: 1) να γράψουμε τις αθροιστικές πιθανότητες σε ένα $r-$αδικό ανάπτυγμα, όπου r ο πληθάριθμος του κωδικού αλφαβήτου και 2) τα l_k στοιχεία του αναπτύγματος που θα χρησιμοποιήσουμε να υπακούν στον κανόνα $log_r\frac{1}{p_k} \leq l_k < log_r\frac{1}{p_k} + 1$ ή αλλιώς $l_k = \lceil log_r\frac{1}{p_k}\rceil$.

Παράδειγμα 5.3. *Έστω η τυχαία μεταβλητή X με τιμές $\{s_1, s_2, s_3, s_4, s_5\}$ και συνάρτηση μάζας πιθανότητας $\{p_1 = 0.25, p_2 = 0.25, p_3 = 0.125, p_4 = 0.125, p_5 = 0.25\}$. Να κωδικοποιηθεί με βάση την κωδικοποίηση Shannon για $k = 2$.*

Λύση

Θα δημιουργήσουμε ένα πίνακα όπου στην πρώτη στήλη θα αναγραφούν οι πιθανότητες εμφανίσεις των συμβόλων ταξινομημένες κατά φθίνουσα σειρά, στην δεύτερη στήλη θα υπολογίσουμε τις αθροιστικές πιθανότητες, στην τρίτη στήλη θα υπολογισθεί το μήκος $l_k = \lceil log_2 \frac{1}{p_k} \rceil$ της κάθε κωδικής λέξης και τέλος στην τέταρτη στήλη γράφουμε την δυαδική ανάπτυξη της αθροιστικής πιθανότητας που αποτελεί την κωδική λέξη του αντίστοιχου συμβόλου.

Σύμβολο	*Πιθανότητα* p_i	*Αθροιστική πιθανότητα* P_k	*μήκος* $l_k = \lceil log_2 \frac{1}{p_k} \rceil$	*Κωδική λέξη* C_k
$s_1^{(1)}$	$p^{(1)} = 0.25 = \frac{1}{4}$	$P_1 = 0$	$l_1 = \lceil log_2 4 = 2 \rceil$	$C_1 = (0)_2 = 00$
$s_2^{(2)}$	$p^{(2)} = 0.25 = \frac{1}{4}$	$P_2 = 0.25 = \frac{1}{4}$	$l_1 = \lceil log_2 4 = 2 \rceil$	$C_2 = (0.25)_2 = 01$
$s_5^{(3)}$	$p^{(3)} = 0.25 = \frac{1}{4}$	$P_3 = 0.5 = \frac{1}{2}$	$l_1 = \lceil log_2 4 = 2 \rceil$	$C_3 = (0.5)_2 = 10$
$s_3^{(4)}$	$p^{(4)} = 0.125 = \frac{1}{8}$	$P_4 = 0.75 = \frac{3}{4}$	$l_1 = \lceil log_2 8 = 3 \rceil$	$C_4 = (0.75)_2 = 110$
$s_4^{(1)}$	$p^{(5)} = 0.125 = \frac{1}{8}$	$P_1 = 0.875 = \frac{7}{8}$	$l_1 = \lceil log_2 8 = 3 \rceil$	$C_5 = (0.875)_2 = 111$

Σημείωση Ο εκθέτης στα σύμβολα s_i χρησιμοποιήθηκε για να δηλώσει τη καινούρια θέση που έχουν μετά τη μετάθεση που υπέστησαν λόγω της ταξινόμησης των πιθανοτήτων τους. Επίσης υπενθυμίζουμε ότι κάθε πραγματικός αριθμός $x \in [0,1] \subset \mathbb{R}$ μπορεί να γραφτεί ως $x = \sum_{i=1}^{\infty} \frac{a_i}{2^i}$ με $a_i \in \{0,1\}$. Λόγου χάρη το κλάσμα $\frac{1}{4} = \frac{1}{2^2} = 0 \cdot \frac{1}{2^1} + 1 \cdot \frac{1}{2^2} = 01$. Ακόμη να σημειώσουμε ότι μπορεί να αναφερόμαστε σε τυχαίες μεταβλητές αλλά αυτή η μαθηματική προσέγγιση των τεχνικών κωδικοποίησης δεν πρέπει να αδυνατεί τον πρακτικό ορίζοντα εφαρμογής τους. Με μία τυχαία μεταβλητή μπορούμε να μοντελοποιήσουμε από τη συχνότητα των γραμμάτων ενός αλφαβήτου μέχρι ολόκληρες συλλογές δεδομένων.

5.3.2 Ανάλυση της κωδικοποίησης Shannon

Η κωδικοποίηση που περιγράφτηκε παραπάνω παρουσιάστηκε και χρησιμοποιήθηκε από τον ίδιο τον Shannon ως ένας δεύτερος τρόπος απόδειξης του θεωρήματος "θεμελιώδες θεώρημα για κανάλια χωρίς θόρυβο"[5]. Αν παρατηρήσουμε λίγο καλύτερα τη διαδικασία κωδικοποίησης θα δούμε ότι ακολουθεί μία σχετικά απλή λογική. Με την ταξινόμηση των πιθανοτήτων καταφέρνουμε να αντιστοιχίζουμε μικρές αθροιστικές πιθανότητες στις συχνά εμφανιζόμενες τιμές της τυχαίας μεταβλητής. Επίσης αν $p_1 > p_j \Rightarrow \frac{1}{p_i} < \frac{1}{p_j} \Rightarrow log_2 \frac{1}{p_i} < log_2 \frac{1}{p_j} \Rightarrow \lceil log_2 \frac{1}{p_i} \rceil \leq \lceil log_2 \frac{1}{p_j} \rceil \Rightarrow$ αν $p_i > p_j \Rightarrow l_i \leq l_j$. Η ισότητα στην τελευταία σχέση μπορούμε να πούμε ότι εκφράζει την χειρότερη περίπτωση για δύο πιθανότητες που η μία είναι καθαρά μικρότερη από την άλλη. Συνοπτικά λοιπόν με τον παραπάνο τρόπο κωδικοποίησης ο Shannon εξασφάλισε ότι οι συχνές τιμές της τυχαίας μεταβλητής θα κωδικοποιούνται με σύντομες κωδικές λέξεις. Ένα ερώτημα που μένει να απαντηθεί είναι αν ο κώδικας που παράγεται από την παραπάνω διαδικασία είναι μοναδικά αποκωδικοποιήσιμος έτσι ώστε να μην δημιουργούνται λάθη κατά την αποκωδικοποίηση του.

Από την ανισότητα των Kraft γνωρίζουμε ότι αν τα μήκη ενός κώδικα ικανοποιούν την ανισότητα $\sum_{i=1}^{|\mathcal{X}|} 2^{-l_1} \leq 1$ τότε ο κώδικας είναι στιγμιαίος. Για μία τυχαία μεταβλητή X που παίρνει τιμές σε ένα πεπερασμένο σύνολο $\mathcal{X}$ με $|\mathcal{X}| = n$ έχουμε:

$$\sum_{i=1}^{n} 2^{-l_1} = \sum_{i=1}^{n} 2^{-\lceil log \frac{1}{p_i} \rceil} \overset{(a)}{\leq} \sum_{i=1}^{n} 2^{-log \frac{1}{p_i}} = \sum_{i=1}^{n} 2^{log_2 p_i} = \sum_{i=1}^{n} p_i = 1 \Rightarrow \sum_{i=1}^{n} 2^{-l_1} \leq 1$$

Το (α) ισχύει από την ανισότητα $x \leq \lceil x \rceil < x + 1 \Rightarrow -x \geq -\lceil x \rceil . - x - 1$

Άρα ο κώδικας είναι στιγμιαίος. Το επόμενο βήμα είναι να εξετάσουμε το μέσο μήκος του κώδικα για να δούμε την αποδοτικότητα του.

[5] Δημοσίευση [Sha48], σελίδες 401-402

$$\overline{L}(X) = \sum_{i=1}^{n} p_i l_i = \sum_{i=1}^{n} p_i \lceil log\frac{1}{p_i} \rceil$$

Επειδή $x \leq \lceil x \rceil < x+1 \Rightarrow log\frac{1}{p_i} \leq \lceil log\frac{1}{p_i} \rceil < log\frac{1}{p_i} + 1$ θα έχουμε:

$$log\frac{1}{p_i} \leq \lceil log\frac{1}{p_i} \rceil < log\frac{1}{p_i} + 1 \overset{p_i \geq 0}{\Rightarrow} p_i \cdot log\frac{1}{p_i} \leq p_i \cdot \lceil log\frac{1}{p_i} \rceil < p_i \cdot (log\frac{1}{p_i} + 1) \Rightarrow$$
$$\sum_{i=1}^{n} p_i \cdot log\frac{1}{p_i} \leq \sum_{i=1}^{n} p_i \cdot \lceil log\frac{1}{p_i} \rceil < \sum_{i=1}^{n} p_i \cdot (log\frac{1}{p_i} + 1) \Rightarrow$$
$$H(X) \leq \overline{L}(X) < H(X) + 1$$

Στο κεφάλαιο 3 που αναλύσαμε θεωρητικά τη συμπίεση αποδείξαμε πως αν έχουμε μία εργοδική και στάσιμη πηγή τότε για μεγάλες ακολουθίες συμβόλων της πηγής το μέσο μήκος της παραπάνω κωδικοποίησης τείνει στην εντροπία.

5.4 Η μέθοδος του Huffman

5.4.1 Παρουσίαση της Μεθόδου

Έστω μία τυχαία μεταβλητή X που παίρνει τιμές σε ένα πεπερασμένο σύνολο $\mathcal{X}$ και έχει συνάρτηση μάζας πιθανότητας $\{p_1, p_2, \cdots p_n\}$.

1. Κατατάσσουμε τις τιμές της X κατά αύξουσα σειρά σύμφωνα με τις πιθανότητες εμφάνισης τους και τις αποθηκεύουμε σε ένα σύνολο S.

2. Εξάγουμε από το S τις τιμές με τις μικρότερες πιθανότητες εμφάνισης συγχωνεύοντας τις σε μία βοηθητική τιμή η οποία έχει ως πιθανότητα το άθροισμα των πιθανοτήτων των τιμών. Στη πρώτη τιμή που επιλέχθηκε για την συγχώνευση δίνουμε την τιμή 0 και στη δεύτερη την τιμή 1.

3. Επανατοποθετούμε τη βοηθητική τιμή στο σύνολο S με τέτοιο τρόπο ώστε οι τιμές που περιέχονται στο S να παραμένουν διατεταγμένες κατά αύξουσα σειρά. Το σύνολο πλέον περιέχει $\mathcal{X} - 1$ στοιχεία.

4. Συνεχίζουμε την ίδια διαδικασία μέχρι να μείνει μόνο ένα στοιχείο στο σύνολο S με πιθανότητα 1.

Παράδειγμα 5.4. *Έστω μία τυχαία μεταβλητή X με τιμές $\{s_1, s_2, s_3, s_4, s_5, s_6, s_7, s_8, s_9, s_{10}\}$ και σ.μ.π $\{0.1, 0.2, 0.1, 0.05, 0.1, 0.05, 0.3, 0.025, 0.025, 0.05\}$. Να κωδικοποιηθεί με βάση τη μέθοδο Huffman.*

Λύση

$\left.\begin{matrix}(s_8, 0.025)\\(s_9, 0.025)\end{matrix}\right\} \rightarrow (s'_8, 0.05)$	$\left.\begin{matrix}(s_4, 0.05)\\(s_6, 0.05)\end{matrix}\right\} \rightarrow (s'_4, 0.1)$	$\left.\begin{matrix}(s_10, 0.05)\\(s'_8, 0.05)\end{matrix}\right\} \rightarrow (s''_8, 0.1)$	$\left.\begin{matrix}(s_1, 0.1)\\(s_3, 0.1)\end{matrix}\right\} \rightarrow (s'_1, 0.2)$
$(s_4, 0.05)$	$(s_{10}, 0.05)$	$(s_1, 0.1)$	$(s_5, 0.1)$
$(s_6, 0.05)$	$(s'_8, 0.05)$	$(s_3, 0.1)$	$(s'_4, 0.1)$
$(s_{10}, 0.05)$	$(s_1, 0.1)$	$(s_5, 0.1)$	$(s''_8, 0.1)$
$(s_1, 0.1)$	$(s_3, 0.1)$	$(s'_4, 0.1)$	$(s_2, 0.2)$
$(s_3, 0.1)$	$(s_5, 0.1)$	$(s_2, 0.2)$	$(s_7, 0.3)$
$(s_5, 0.1)$	$(s_2, 0.2)$	$(s_7, 0.3)$	
$(s_2, 0.2)$	$(s_7, 0.3)$		
$(s_7, 0.3)$			

$\left.\begin{matrix}(s_5, 0.1)\\(s'_4, 0.1)\end{matrix}\right\} \to (s''_4, 0.2)$	$\left.\begin{matrix}(s''_1, 0.1)\\(s_2, 0.2)\end{matrix}\right\} \to (s'''_1, 0.3)$	$\left.\begin{matrix}(s'_1, 0.2)\\(s''_4, 0.2)\end{matrix}\right\} \to (s_1^{(4)}, 0.4)$	$\left.\begin{matrix}(s_7, 0.3)\\(s_4^{(4)}, 0.3)\end{matrix}\right\} \to (s_1^{(5)}, 0.6)$
$(s''_8.0.1)$	$(s'_1, 0.2)$	$(s_7, 0.3)$	$s_1^{(4)}, 0.4)$
$(s_2, 0.2)$	$(s''_4, 0.2)$	$(s'''_1, 0.3)$	
$(s'_1, 0.2)$	$(s_7, 0.3)$		
$(s_7, 0.3)$			

$$\left.\begin{matrix}(s_1^{(4)}, 0.6)\\(s_4^{(5)}, 0.4)\end{matrix}\right\} \to (s_1^{(6)}, 1)$$

5.4.2 Η μέθοδος Huffman με τη χρήση δένδρου

Να υπενθυμίσουμε από το κεφάλαιο 3, ότι οι στιγμιαίοι κώδικες με σύμβολα που ανήκουν σε ένα αλφάβητο F μπορούν να αναπαρασταθούν από έν $|\mathcal{F}|$-αδικό δένδρο. Άρα η μέθοδος κωδικοποίησης Huffman μπορεί να επαναδιατυπωθεί ως εξής:

Διαδικασία κωδικοποίησης

1. Κατατάσσουμε τις τιμές της X κατά αύξουσα σειρά σύμφωνα με τις πιθανότητες εμφάνισης τους και τις αποθηκεύουμε σε ένα διατεταγμένο S.

2. Εξάγουμε από το S τις τιμές με τις μικρότερες πιθανότητες εμφάνισης συγχωνεύοντας τις σε ένα κόμβο ο οποίος έχει ως τιμή το άθροισμα των πιθανοτήτων των τιμών. Στο αριστερό παιδί του κόμβου δίνουμε την τιμή 0 και στο δεξί παιδί την τιμή 1.

3. Επανατοποθετούμε τον κόμβο στο σύνολο S στην θέση που διατηρεί το S διατεταγμένο. Το σύνολο πλέον περιέχει $\mathcal{X} - 1$ στοιχεία.

4. Συνεχίζουμε την ίδια διαδικασία μέχρι να μείνει μόνο ένα στοιχείο στο σύνολο S με πιθανότητα 1. Το τελευταίο στοιχείο αποτελεί τη ρίζα του δένδρου.

Διαδικασία αποκωδικοποίησης

Το δένδρο κωδικοποίησης που παράγεται από τον Huffman χρησιμοποιείται και για την αποκωδικοποίηση ενός συνόλου δεδομένων.

1. Για να αποκωδικοποιήσουμε ένα κωδικοποιημένο σύνολο δεδομένων, ξεκινάμε από τη ρίζα του δένδρου και κάθε φορά που διαβάζουμε κάποιο από τα F κωδικά σύμβολα ακολουθούμε το κλαδί που αντιστοιχεί στο σύμβολο και μεταβαίνουμε στον ανάλογο κόμβο. Η αποκωδικοποίηση ενός στοιχείου του συνόλου δεδομένων τελειώνει όταν φτάσουμε σε κάποιο από τα φύλλα του δένδρου.

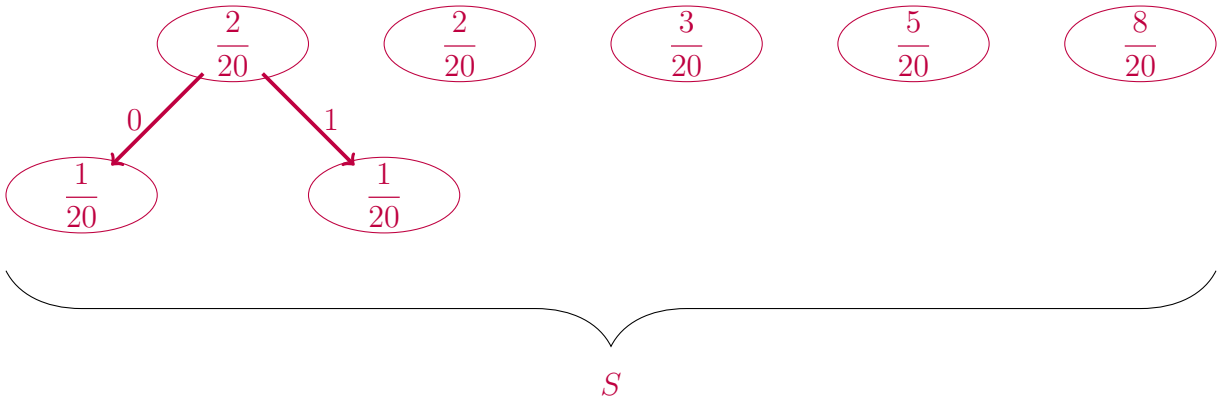
2. Όταν μεταβούμε σε κάποιο από τα φύλλα γράφουμε το αποκωδικοποιούμενο σύμβολο και μεταβαίνουμε πάλι στην ρίζα.

3. Η διαδικασία συνεχίζεται μέχρι να αποκωδικοποιηθεί κάθε στοιχείο του συνόλου δεδομένων

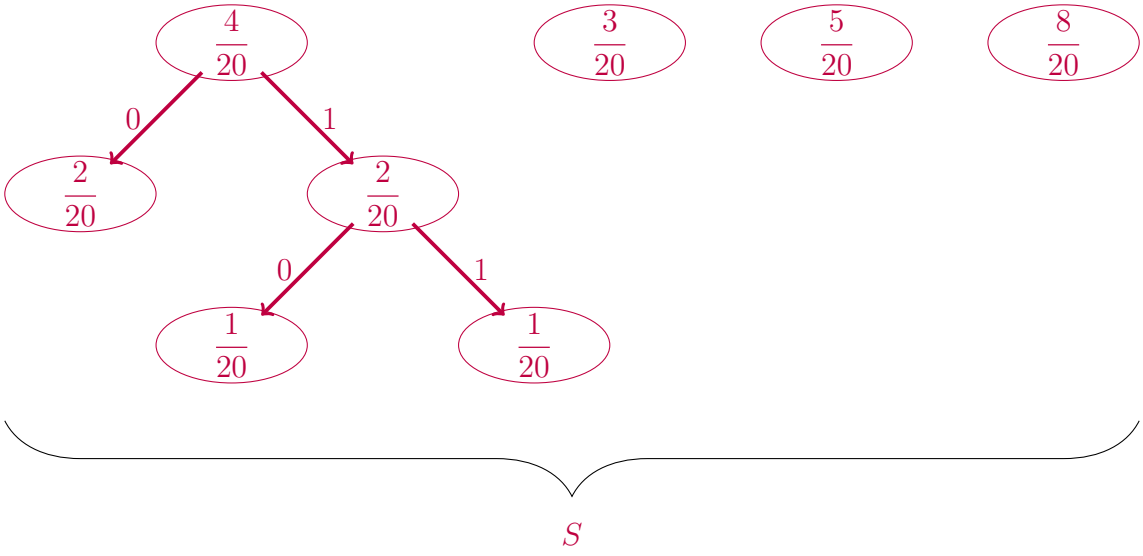
Παράδειγμα 5.5. *(**Κωδικοποίηση Huffman με τη χρήση δυαδικού δένδρου**) Έστω μία τυχαία μεταβλητή με τιμές* $\{x_1,\ x_2,\ x_3,\ x_4,\ x_5,\ x_6\}$ *και* $P_X(x) = \{\frac{1}{20},\ \frac{1}{20},\ \frac{2}{20},\ \frac{3}{20},\ \frac{5}{20},\ \frac{8}{20}\}$. *Να κωδικοποιηθεί το αλφάβητο που μοντελοποιείται από την τυχαία μεταβλητή X με ένα δυαδικό δένδρο Huffman. Στην συνέχεια κωδικοποιήστε και αποκωδικοποιήστε την ακολουθία* $x_5x_5x_2x_1$

Λύση

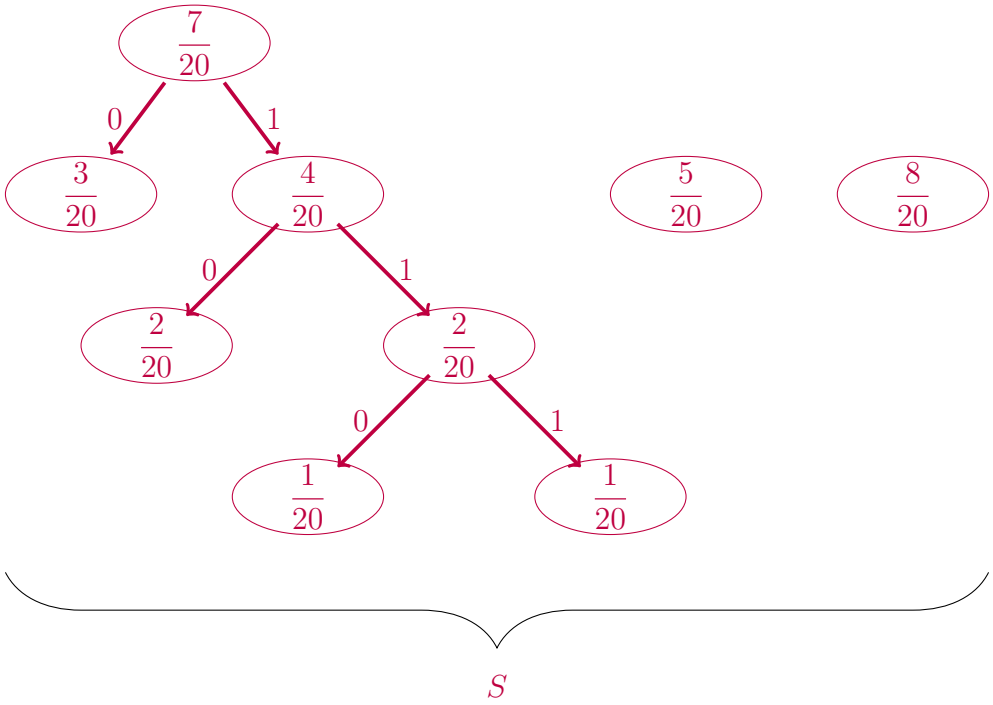
1. *Η κωδικοποίηση ξεκινάει με τις δύο μικρότερες πιθανότητες* $\frac{1}{20}$, $\frac{1}{20}$, *τις ενώνει σε ένα κόμβο και επανατοποθετεί το κόμβο με το άθροισμα των πιθανοτήτων τους στο διατεταγμένο σύνολο* S



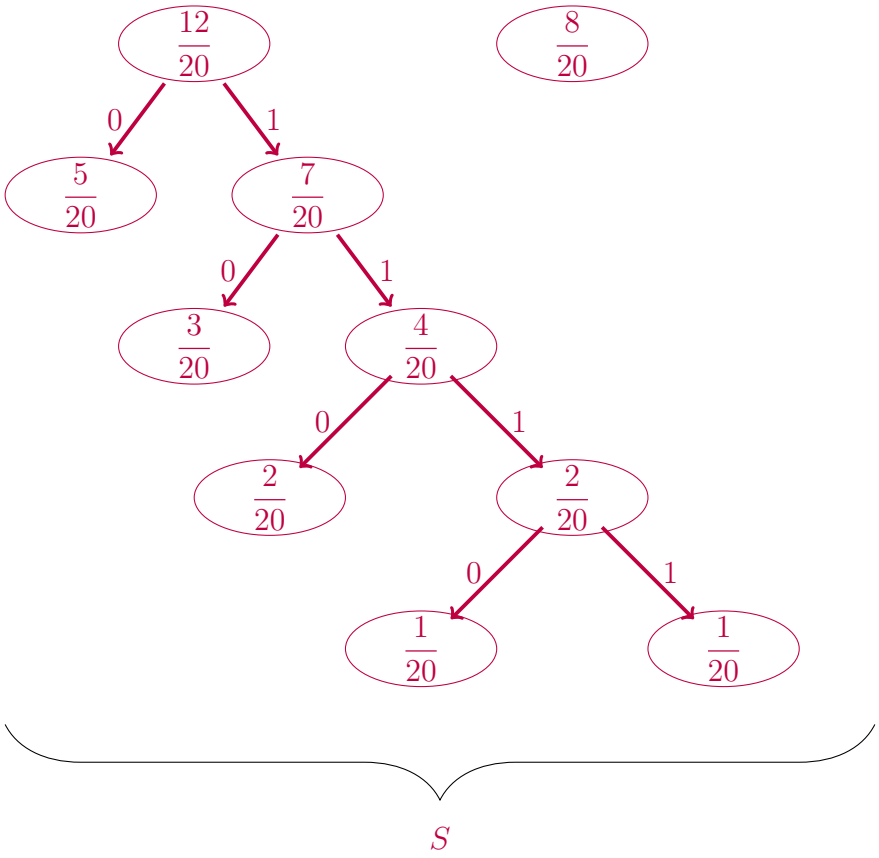
2. *Συνεχίζουμε συγχωνεύοντας τον κόμβο που μόλις δημιουργήθηκε με τον κόμβο* $\frac{2}{20}$ *που προϋπήρχε στο δένδρο.*



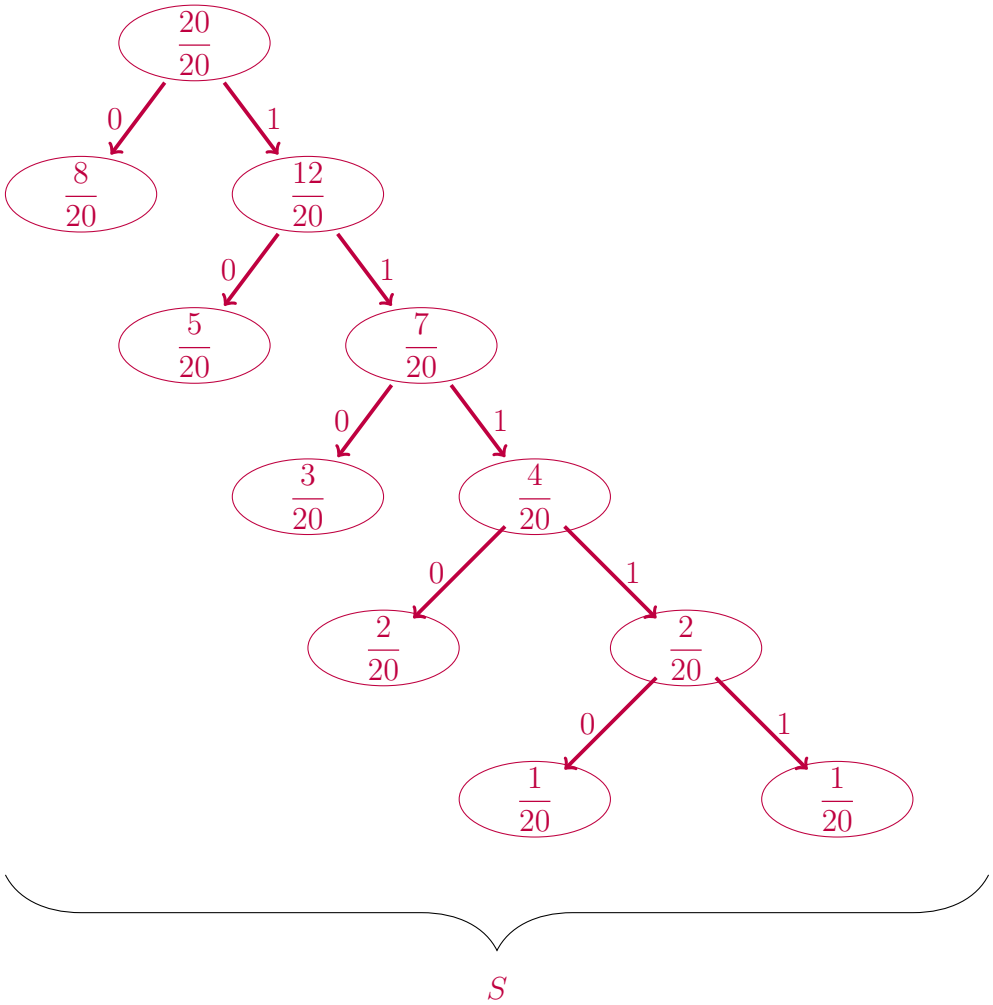
3. *Ενώνουμε τον κόμβο* $\frac{4}{20}$ *με τον κόμβο* $\frac{3}{20}$



4. *Συγχωνεύουμε τον κόμβο* $\frac{5}{20}$ *με τον* $\frac{7}{20}$



5. *Ολοκληρώνοντας συγχωνεύουμε τους δύο τελευταίους κόμβους που απέμειναν*



Το παράδειγμα αυτό αποτελεί την χειρότερη περίπτωση κωδικοποίησης που μπορεί να τύχει. Παρατηρήστε ότι οι αριθμητές των πιθανοτήτων αποτελούν μία ακολουθία Fibonacci, όπου οι δύο προηγούμενοι αριθμοί δίνουν το άθροισμα του επόμενου:

$$F(n) = F(n-1) + F(n-2)$$

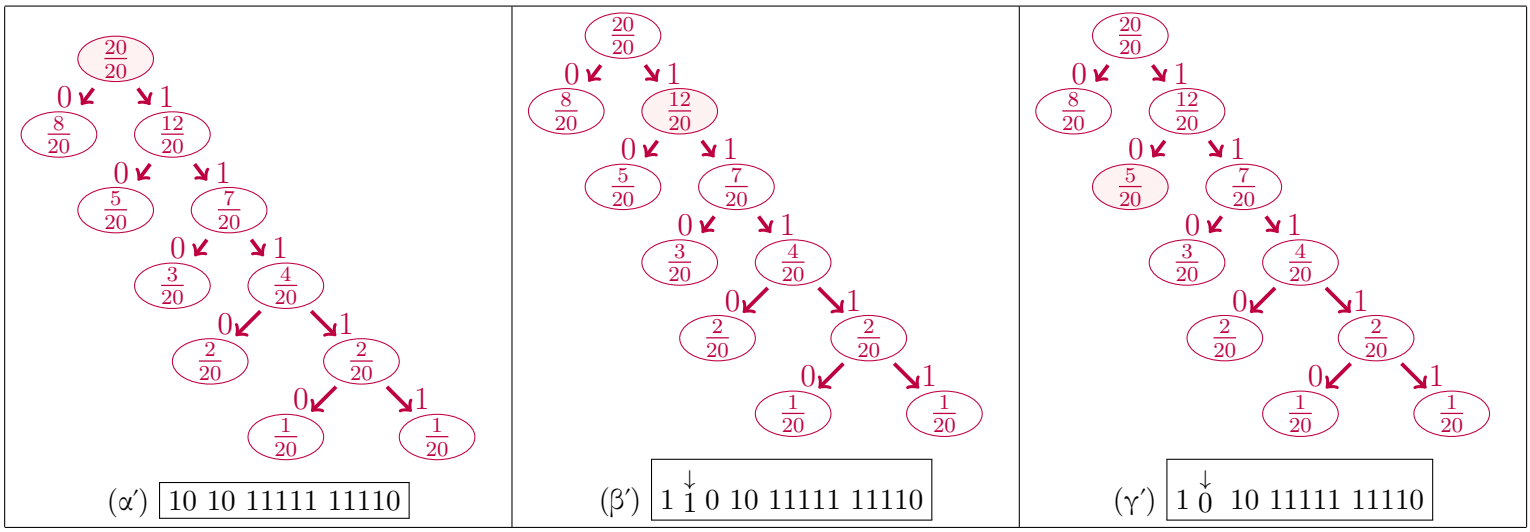
Γενικά όταν έχουμε μία κατανομή πιθανότητας όπου διαδοχικά το άθροισμα των δύο μικρότερων πιθανοτήτων είναι μικρότερο από τις υπόλοιπες θα οδηγούμαστε σε μία κακή κωδικοποίηση Huffman.

Οι κώδικες που προκύπτουν από τον δένδρο Huffman είναι

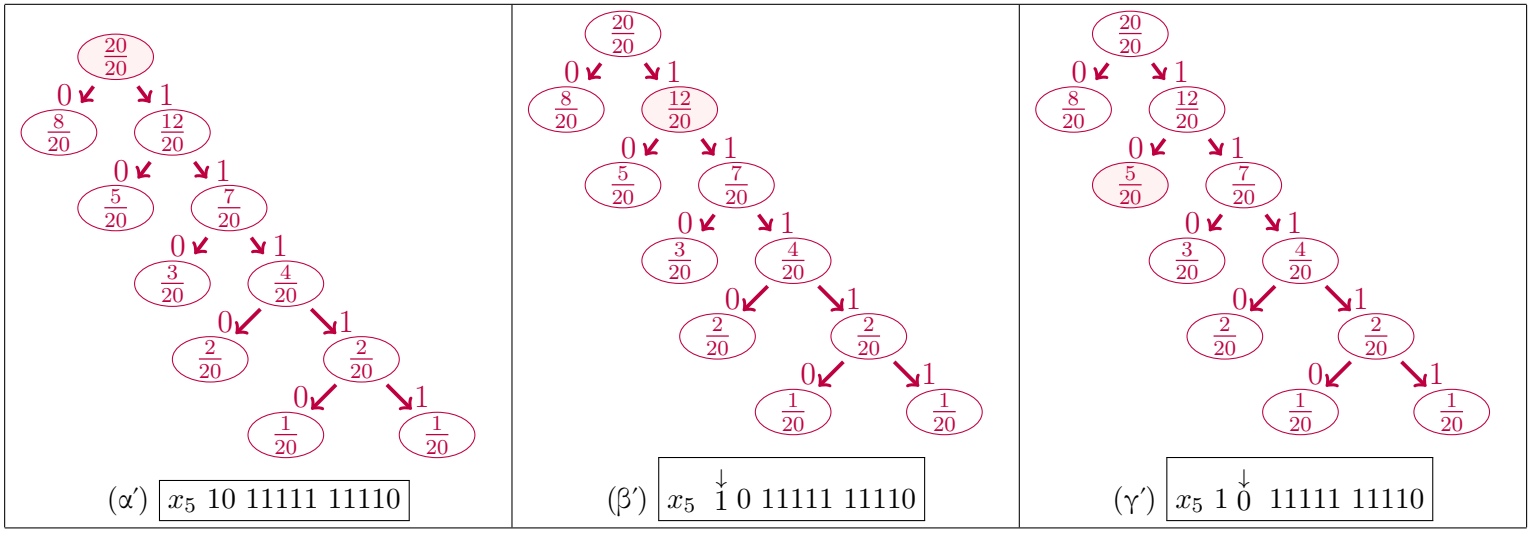
Σύμβολο	*Πιθανότητα*	*Κωδική λέξη*
x_1	$\frac{1}{20}$	11110
x_2	$\frac{1}{20}$	11111
x_3	$\frac{2}{20}$	1110
x_4	$\frac{3}{20}$	110
x_5	$\frac{5}{20}$	10
x_6	$\frac{8}{20}$	0

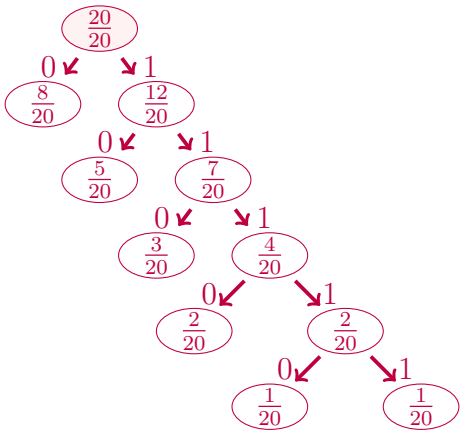
Η κωδικοποίηση της $x_5x_5x_2x_1$ θα είναι $\boxed{10\ 10\ 11111\ 11110}$. *Για να αποκωδικοποιήσουμε τη συμβολοσειρά ξεκινάμε από τη ρίζα και διασχίζουμε το δένδρο πηγαίνοντας δεξιά ή αριστερά ανάλογα το σύμβολο που συναντάμε. Όταν φτάσουμε σε κάποιο φύλλο γράφουμε το αποκωδικοποιημένο σύμβολο και μεταβαίνουμε πάλι στη ρίζα για να αποκωδικοποιήσουμε την επόμενη κωδική λέξη.*

Ξεκινάμε από τη ρίζα, διαβάζουμε 1 και μεταβαίνουμε στον κόμβο $\frac{12}{20}$, έπειτα διαβάζουμε 0 και μεταβαίνουμε στον κόμβο $\frac{8}{20}$. Ο κόμβος $\frac{8}{20}$ είναι φύλλο. Γράφουμε το σύμβολο x_5 και μεταβαίνουμε πάλι στη ρίζα.

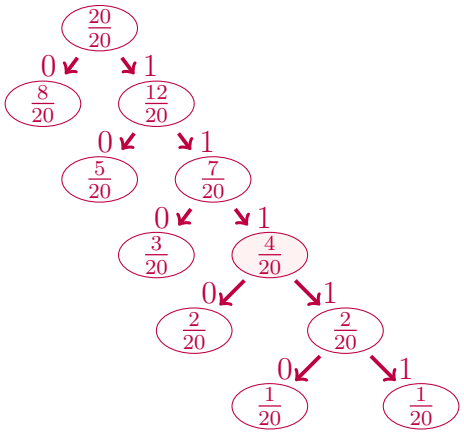


Η διαδικασία της αποκωδικοποιήσεως παρουσιάζεται στις επόμενες εικόνες.

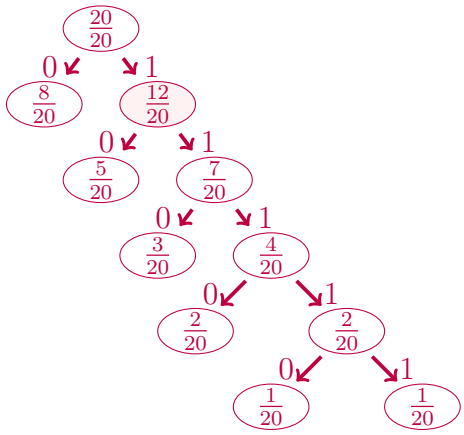




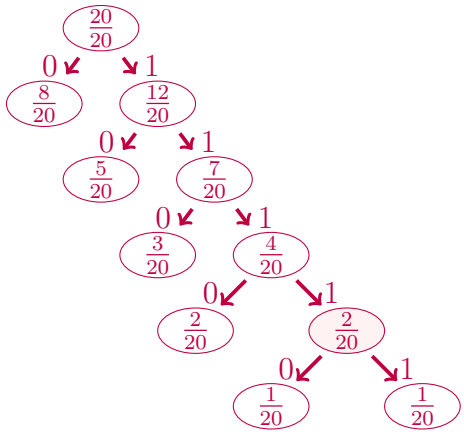
(α′) $x_5\ x_5\ 11111\ 11110$



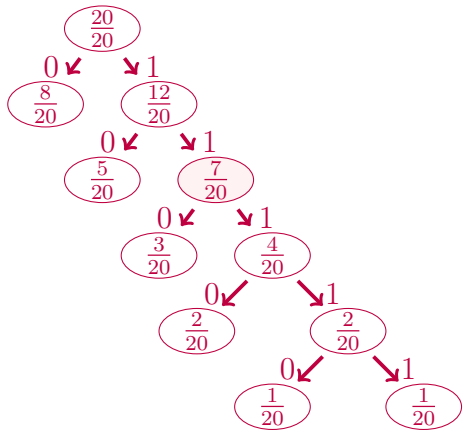
(δ′) $x_5\ x_5\ 11\ \underset{\downarrow}{1}\ 11\ 11110$



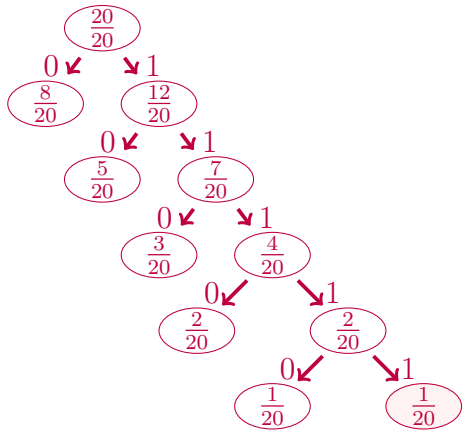
(β′) $x_5\ x_5\ \underset{\downarrow}{1}\ 1111\ 11110$



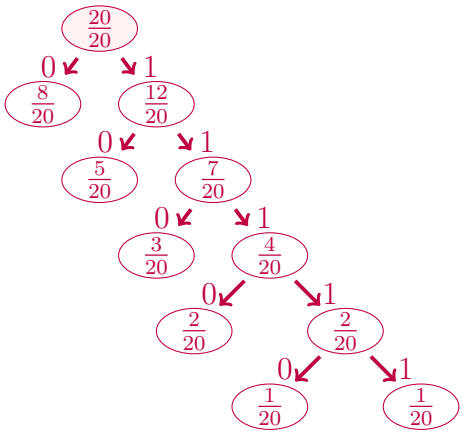
(ε′) $x_5\ x_5\ 11\ \underset{\downarrow}{1}\ 1\ 11110$



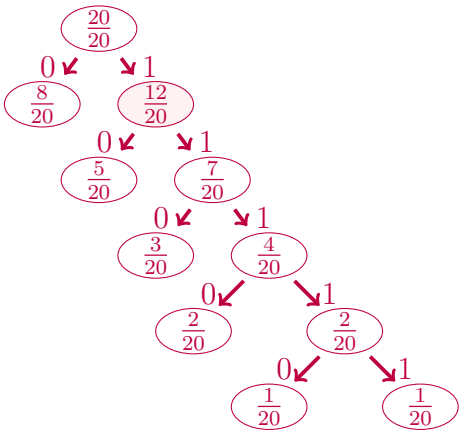
(γ′) $x_5\ x_5\ 1\ \underset{\downarrow}{1}\ 111\ 11110$



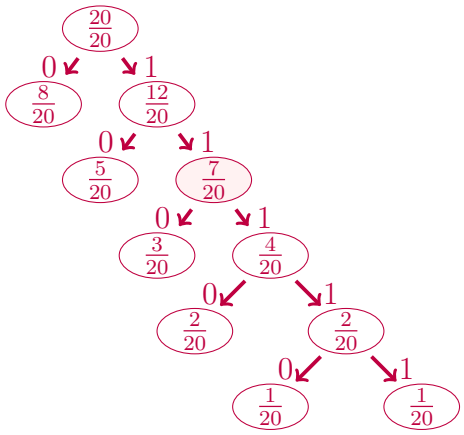
(ϛ′) $x_5\ x_5\ 1111\ \underset{\downarrow}{1}\ \ 11110$



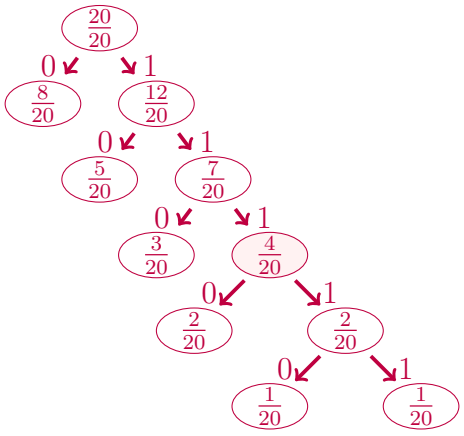
(α′) $x_5\ x_5\ x_2\ 11110$



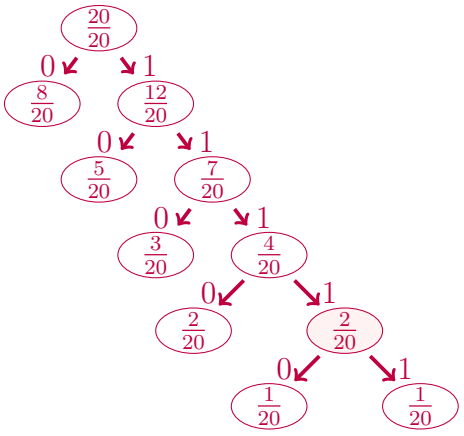
(β′) $x_5\ x_5\ x_2\ \underset{\downarrow}{1}\ 1110$



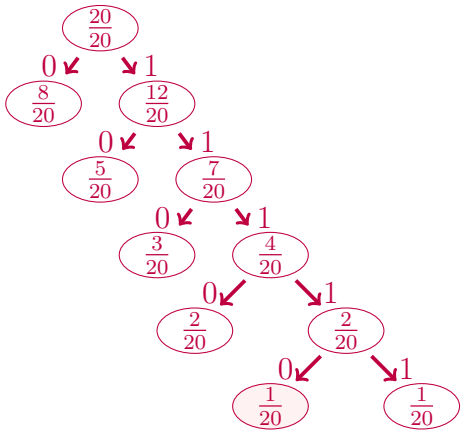
(γ′) $x_5\ x_5\ x_2\ 1\ \underset{\downarrow}{1}\ 110$



(δ′) $x_5\ x_5\ x_2\ 11\ \underset{\downarrow}{1}\ 10$



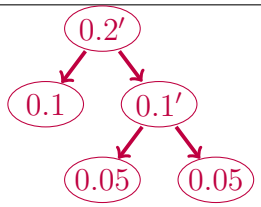
(ε′) $x_5\ x_5\ x_2\ 111\ \underset{\downarrow}{1}\ 0$



(τ′) $x_5\ x_5\ x_2\ 1111\ \underset{\downarrow}{0} \Rightarrow x_5\ x_5\ x_2\ x_1$

5.4.3 Κώδικες Huffman ελάχιστης διασποράς

Έστω μία τυχαία μεταβλητή X που παίρνει τιμές στο $\mathcal{X} = \{x_1, x_2, x_3, x_4, x_5, x_6\}$ με $P_X(x) = \{0.05, 0.05, 0.1, 0.1, 0.2, 0.5\}$. Τους κόμβους που αποτελούν φύλλα του δένδρου Huffman θα τους λέμε απλούς ενώ αυτούς που δημιουργούνται λόγω των συγχωνεύσεων θα του λέμε σύνθετους. Η διαδικασία αρχικά θα συγχώνευε τις πιθανότητες 0.05 και 0.05 στο κόμβο με πιθανότητα 0.1. Στην συνέχεια η διαδικασία κωδικοποίησης θα μπορούσε να επιλέξει να συγχωνεύσει είτε το σύνθετο κόμβο που δημιουργήθηκε στο πρώτο βήμα είτε του δύο απλούς με πιθανότητες 0.1 έκαστος. Παρακάτω απεικονίζεται το αποτέλεσμα των διαφορετικών επιλογών.

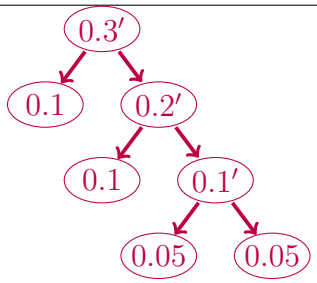


(α′) Το σύνολο των πιθανοτήτων $\{0.05, 0.05, 0.1, 0.1, 0.2, 0.5\}$ δέχθηκε σε σειρά τις εξής συγχωνεύσεις: $\{0.05, 0.05, 0.1, 0.1, 0.2, 0.5\} \Rightarrow \{0.1', 0.1, 0.1, 0.2, 0.5\} \Rightarrow \{0.1, 0.2' 0.2, 0.5\}$, όπου με τόνο σημειώνονται οι σύνθετοι κόμβοι

(β′) Το σύνολο των πιθανοτήτων $\{0.05, 0.05, 0.1, 0.1, 0.2, 0.5\}$ δέχθηκε σε σειρά τις εξής συγχωνεύσεις: $\{0.05, 0.05, 0.1, 0.1, 0.2, 0.5\} \Rightarrow \{0.1, 0.1, 0.1', 0.2, 0.5\} \Rightarrow \{0.1', 0.2, 0.2', 0.5\}$, όπου με τόνο σημειώνονται οι σύνθετοι κόμβοι

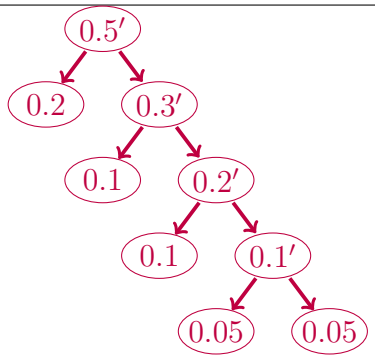
Στην περίπτωση (α′) οι σύνθετοι κόμβοι κατά την ταξινόμηση τους στο σύνολο εισέρχονται στην χαμη-

λότερη δυνατή θέση του διατεταγμένου συνόλου, ενώ στη περίπτωση (β′) οι σύνθετοι κόμβοι τοποθετούνται όσο ψηλότερα γίνεται στο διατεταγμένο σύνολο. Οι συγχωνεύσεις συνεχίζονται με την κάθε περίπτωση να ακολουθεί την δικιά της λογική.

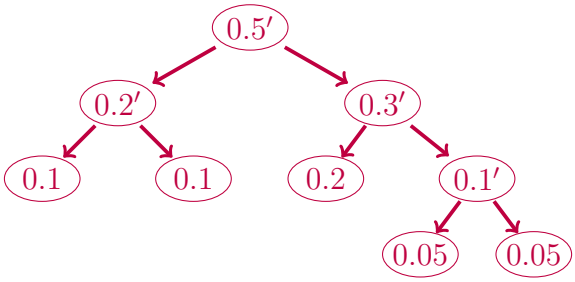


(α′) { 0.1, 0.2′ 0.2, 0.5} ⇒ {0.2, 0.3′, 0.5}, όπου με τόνο σημειώνονται οι σύνθετοι κόμβοι

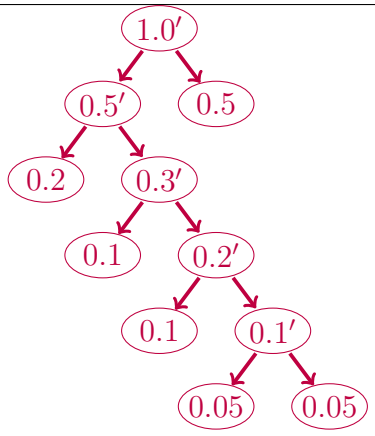
(β′) { 0.1′, 0.2, 0.2′, 0.5} ⇒ {0.2′, 0.3′, 0.5}, όπου με τόνο σημειώνονται οι σύνθετοι κόμβοι



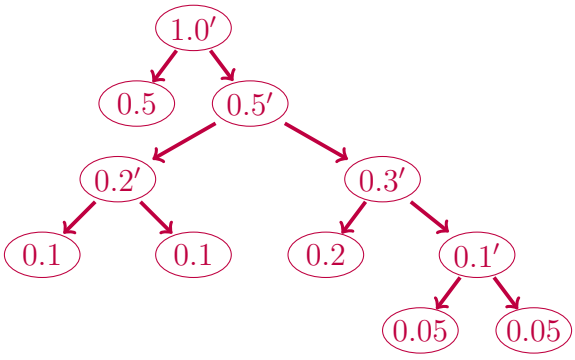
(α′) {0.2, 0.3′, 0.5} ⇒ {0.5′, 0.5}, όπου με τόνο σημειώνονται οι σύνθετοι κόμβοι



(β′) {0.2′, 0.3′, 0.5} ⇒ {0.5, 0.5′}, όπου με τόνο σημειώνονται οι σύνθετοι κόμβοι



(α′) {0.5′, 0.5} ⇒ {1.0′}, όπου με τόνο σημειώνονται οι σύνθετοι κόμβοι



(β′) {0.2′, 0.3′, 0.5} ⇒ {0.5, 0.5′}, όπου με τόνο σημειώνονται οι σύνθετοι κόμβοι

Παρατηρούμε ότι οι δύο λογικές οδήγησαν σε διαφορετικά δένδρα. Θα ονομάσουμε το δένδρο που παρασκευάστηκε με τη λογική (α′) T_1 και το άλλο T_2.Το μέσο μήκος κωδίκων που παράγονται από τα T_1 και T_2 είναι:

$$\overline{L}(T_1) = 1 \cdot 0.5 + 2 \cdot 0.2 + 3 \cdot 0.1 + 4 \cdot 0.1 + 5 \cdot 0.05 + 5 \cdot 0.05 = 0.5 + 0.4 + 0.3 + 0.4 + 0.25 + 0.25 = 2.1$$

$$\overline{L}(T_2) = 1 \cdot 0.5 + 3 \cdot 0.1 + 3 \cdot 0.1 + 3 \cdot 0.2 + 4 \cdot 0.05 + 4 \cdot 0.05 = 0.5 + 0.3 + 0.3 + 0.6 + 0.2 + 0.2 = 2.1$$

Όπως ήταν αναμενόμενο τα δύο δένδρα δεν διαφέρουν στο μέσο μήκος κώδικα που παράγουν μολονότι υπάρχει δομική διαφορά ανάμεσα τους. Συγκεκριμένα το T_2 φαίνεται και είναι περισσότερο ισορροπημένο από ότι το T_1. Αν υπολογίσουμε τις διασπορές των δύο δένδρων θα έχουμε:

$$Var[T_1] = \sum_{1}^{6} Pr[X = x_i](l_i - 2.1)^2 = 0.5 \cdot (1 - 2.1)^2 + 0.2 \cdot (2 - 2.1)^2 + 0.1 \cdot (3 - 2.1)^2 + 0.1 \cdot (4 - 2.1)^2 +$$
$$0.05 \cdot (5 - 2.1)^2 + 0.05 \cdot (5 - 2.1)^2 = 1.89$$
$$Var[T_2] = \sum_{1}^{6} Pr[X = x_i](l_i - 2.1)^2 = 0.5 \cdot (1 - 2.1)^2 + 0.2 \cdot (3 - 2.1)^2 + 0.1 \cdot (3 - 2.1)^2 + 0.1 \cdot (3 - 2.1)^2 +$$
$$0.05 \cdot (4 - 2.1)^2 + 0.05 \cdot (4 - 2.1)^2 = 1.21$$

Βλέπουμε πως το T_2 έχει πολύ μικρότερη διασπορά από το T_1. Η ιδιότητα αυτή είναι πολύ χρήσιμη σε εφαρμογές που χρειάζονται σταθερό ρυθμό μετάδοσης δεδομένων και μεγάλες αποκλίσεις μεταξύ των μηκών των κωδικών λέξεων δημιουργούν προβλήματα συγχρωτισμού και ταχύτητας.

5.4.4 Ανάλυση της κωδικοποίησης Huffman

Η παραπάνω μέθοδος κωδικοποίησης ανακαλύφθηκε από τον Huffman το 1951 κατά τη διάρκεια της παρακολούθησης ενός μαθήματος ηλεκτρολόγων μηχανικών που διδασκόταν από τον Robert Fano[6]. Ο Fano έδωσε στου μαθητές του την επιλογή να γράψουν το καθιερωμένο διαγώνισμα στο τέλος του μαθήματος ή μία εξαμηνιαία εργασία, το θέμα της οποίας ήταν να βρούνε τον βέλτιστο κώδικα συμπίεσης για τις τιμές μίας τυχαίας μεταβλητής. Τους απέκρυψε βέβαια ότι το παραπάνω θέμα αποτελούσε ανοιχτό πρόβλημα της εποχής πάνω στο οποίο εργαζόταν και ο ίδιος. Ο Huffman δέχτηκε την πρόκληση δίχως να ξέρει ότι το πρόβλημα ήταν ανοιχτό και επιβεβαίωσε την ύπαρξη της εξαίρεσης του μαθητή που ξεπέρασε τον δάσκαλο. Ο ίδιος αργότερα παραδέχτηκε πως αν ήξερε ότι ο καθηγητής του και ο ίδιος ο Shannon είχαν καταπιαστεί με το παραπάνω πρόβλημα ποτέ δεν θα είχε επιχειρήσει να το λύσει.

Η ειδοποιός διαφορά ανάμεσα στην λύση που προέβαλε ο Huffman σε σχέση με τους κώδικες που αναπτύχθηκαν από τους Fano και Shannon είναι ότι πετύχαινε τη δημιουργία ενός κώδικα ελάχιστου πλεονασμού πληροφορίας (minimum rendundancy) ακόμα και για πεπερασμένα αλφάβητα. Αν παρατηρήσουμε τις αναλύσεις των δύο προηγούμενων μεθόδων συμπεραίνουμε ότι οι κώδικες που παράγονται από αυτές τείνουν στην εντροπία όταν το σύνολο των μηνυμάτων N τείνει στο άπειρο. Για να δημιουργήσει ένα αποδοτικό κώδικα ο Huffman διετύπωσε δύο προυποθέσεις που έπρεπε να ισχύουν:

1. Κανένα μήνυμα δεν θα περιέχει την ίδια διάταξη κωδικών ψηφίων με κάποιο από τα υπόλοιπα
2. Τα μηνύματα πρέπει να κατασκευαστούν με τέτοιο τρόπο ώστε να μην χρειαστεί επιπλέον σήμανση, άρα και κωδικά σύμβολα, που να σηματοδοτούν την έναρξη και τη λήξη της κωδικής λέξης.

Με όσα έχουν επισημανθεί στα προηγούμενα κεφάλαια καταλαβαίνουμε ότι αυτό που θεώρησε απαραίτητο ο Huffman είναι να φτιάξει ένα στιγμιαίο κώδικα. Επίσης κατάφερε να διατυπώσει τρεις ακόμα προϋποθέσεις που αναγκαστικά πρέπει να ικανοποιεί ένα προθεματικός κώδικας για να είναι βέλτιστος (optimal), δηλαδή να επιτυγχάνει ελάχιστο μέσο μήκος. Οι προϋποθέσεις διατυπώνονται στο παρακάτω λήμμα:

Λήμμα 5.1. *Για οποιαδήποτε κατανομή πιθανοτήτων που αντιστοιχεί σε κάποια τυχαία μεταβλητή X, υπάρχει ένας βέλτιστος στιγμιαίος κώδικας ο οποίος ικανοποιεί τις παρακάτω προυποθέσεις:*

1. *Για οποιεσδήποτε πιθανότητες $Pr[X = x_i],\ Pr[X = x_j]$ της κατανομής με $Pr[X = x_i] > Pr[X = x_j]$ πρέπει να ισχύει $l_i \leq l_j$, όπου $l_i,\ l_j$ τα μήκη των κωδικών λέξεων με πιθανότητες $Pr[X = x_i],\ Pr[X = x_j]$ αντίστοιχα*

[6] **InventionofHuffmanCodes**.

2. *Οι δύο μεγαλύτερες κωδικές λέξεις πρέπει να έχουν το ίδιο μήκος*

3. *Οι δύο μεγαλύτερες κωδικές λέξεις πρέπει να αντιστοιχούν στις δύο λιγότερες πιθανές τιμές της τυχαίας μεταβλητής X και να διαφέρουν μόνο στο τελευταίο ψηφίο.*

Απόδειξη
Έστω ότι έχουμε ένα βέλτιστο κώδικα C^*

1. *Υποθέτουμε ότι η πρώτη προϋπόθεση δεν ισχύει, δηλαδή υπάρχουν δύο κωδικές λέξεις με* $Pr[X = x_k] > Pr[X = x_j]$ *και* $l_k \geq l_j$. *Τότε μπορούμε να δημιουργήσουμε ένα καινούργιο κώδικα* C' *ανταλλάσοντας τη κωδική λέξη* $C(x_k)$ *με την* $C(x_j)$ $(C(x_k) \leftrightarrow C(x_j))$. *Τότε:*

$$\overline{L}(C') - \overline{L}(C^*) = \sum Pr[X = x_i] \cdot l'_i - \sum Pr[X = x_i] \cdot l_i =$$
$$(Pr[X = x_1] \cdot l(x_1) + \cdots + Pr[X = x_k] \cdot l_j + Pr[X = x_j] \cdot l_k + \cdots + Pr[X = x_n] \cdot l(x_n)) -$$
$$((Pr[X = x_1] \cdot l(x_1) + \cdots + Pr[X = x_k] \cdot l_k + Pr[X = x_j] \cdot l_j + \cdots + Pr[X = x_n] \cdot l(x_n)) =$$
$$Pr[X = x_k] \cdot l_j + Pr[X = x_j] \cdot l_k - Pr[X = x_k] \cdot l_k - Pr[X = x_j] \cdot l_j =$$
$$Pr[X = x_k] \cdot (l_j - l_k) - Pr[X = x_j] \cdot (l_j - l_k) = (Pr[X = x_k] - Pr[X = x_j]) \cdot (l_j - l_k) \leq 0,$$
αφού $Pr[X = x_k] > Pr[X = x_j] \Rightarrow Pr[X = x_k] - Pr[X = x_j] > 0$ *και* $l_k \geq l_j \Rightarrow l_J - l_k \leq 0$. *Άρα έπεται :*
$\overline{L}(C') - \overline{L}(C^*) \leq 0 \Rightarrow \overline{L}(C') \leq \overline{L}(C^*)$

Το οποίο είναι άτοπο καθώς ο κώδικας C^* *είναι βέλτιστος όποτε δεν γίνεται να έχει μεγαλύτερο μέσο μήκος από οποιονδήποτε άλλο.*

2. *Έστω ότι οι δύο μεγαλύτερες κωδικές λέξεις δεν έχουν το ίδιο μήκος, δηλαδή* $l_N > l_{N-1}$ *κατά* k *σύμβολα. Επειδή ο κώδικας είναι στιγμιαίος έπεται ότι η* C_{N-1} *καθώς και οποιαδήποτε άλλη λέξη έχει μήκος ίσο με* l_{N-1} *δεν θα αποτελεί πρόθεμα της* C_N. *Για το λόγο αυτό μπορούμε να διαγράψουμε τα τελευταία* k *σύμβολα της* C_N *δημιουργώντας έτσι ένα καινούριο κώδικα* C'. *Τότε:*

$$\overline{L}(C') - \overline{L}(C^*) = \sum p_i \cdot l'_i - \sum p_i \cdot l_i =$$
$$(Pr[X = x_1] \cdot l_1 + Pr[X = x_2] \cdot l_2 + \cdots + Pr[X = x_{N-1}] \cdot l_{N-1} + Pr[X = x_N] \cdot l_{N-1}) -$$
$$(Pr[X = x_1] \cdot l_1 + Pr[X = x_2] \cdot l_2 + \cdots + Pr[X = x_{N-1}] \cdot l_{N-1} + Pr[X = x_N] \cdot l_N) =$$
$$Pr[X = x_{N-1}] \cdot l_{N-1} + Pr[X = x_N] \cdot l_{N-1} - Pr[X = x_{N-1}] \cdot l_{N-1} - Pr[X = x_N] \cdot l_N =$$
$$Pr[X = x_{N-1}] \cdot l_{N-1} + Pr[X = x_N] \cdot l_{N-1} - Pr[X = x_{N-1}] \cdot l_{N-1} - Pr[X = x_N] \cdot (l_{N-1} + k) =$$
$$- Pr[X = x_N] \cdot k \leq 0 \Rightarrow \overline{L}(C') - \overline{L}(C^*) \leq 0 \Rightarrow \overline{L}(C') \leq \overline{L}(C^*)$$

Το οποίο είναι άτοπο γιατί κώδικας C^* *είναι βέλτιστος.*

3. *Το συμπέρασμα πως οι δύο μεγαλύτερες λέξεις πρέπει να αντιστοιχούν στις δύο μικρότερες πιθανότητες είναι άμεση συνέπεια του (1). Έστω ότι έχουμε ένα κώδικα στον οποίο υπάρχουν δύο κωδικές λέξεις* C_i, C_j*μήκους* l_N *που δεν διαφέρουν μόνο στο τελευταίο σύμβολο αλλά και στο* $l_N - 1$ *κατά σειρά. Επειδή ο κώδικας είναι στιγμιαίος γνωρίζουμε ότι δεν υπάρχουν κώδικες μικρότερου μήκος που να αποτελούν προθέματα των παραπάνω κωδικών λέξεων και επίσης κανένα πρόθεμα των* C_i, C_j*δεν αποτελεί κωδική λέξη. Άρα μπορούμε να αφαιρέσουμε το τελευταίο σύμβολο χωρίς να επηρεαστεί κάποια από τις ιδιότητες (1) και (2) του κώδικα. Έτσι όμως προκύπτει ένας καινούργιος κώδικας* C' *με μέσο μήκος* $\overline{L}$ *μικρότερο από τον* C^* *το οποίο είναι άτοπο.*

Η κωδικοποίηση Huffman παράγει ένα βέλτιστο κώδικα

Αφού αναλύσαμε τα χαρακτηριστικά ενός βέλτιστου στιγμιαίου κώδικα όπως τα διατύπωσε ο Huffman ήρθε η ώρα να αποδείξουμε γιατί σε κάθε βήμα η μέθοδος του παράγει ένα βέλτιστο κώδικα.

Θεώρημα 5.1. *Έστω μία τυχαία μεταβλητή X που παίρνει τιμές σε ένα πεπερασμένο σύνολο $\mathcal{X}$ και ακολουθεί την σ.μ.π $P_X(x)$. Ο κώδικας Huffman παράγει έναν βέλτιστο στιγμιαίο κώδικα για την συνάρτηση μάζας πιθανότητας $P_X(x)$.*

Απόδειξη

• *Η απόδειξη θα γίνει με επαγωγή στο πληθάριθμο $|\mathcal{X}|$ του συνόλου τιμών της τυχαίας μεταβλητής. Για $|\mathcal{X}| = 1$ ο αλγόριθμος παράγει έναν βέλτιστο κώδικα μήκους 0, καθώς αν η τυχαία μεταβλητή αποτελείται μόνο από μία τιμή, έστω την x_i, τότε το $p_i = 1$ και άρα $H(X) = Pr[X = x_i] \cdot log\frac{1}{Pr[X = x_i]} = 1 \cdot log1 = 0$.*

•*(Επαγωγική υπόθεση). Έστω ότι η μέθοδος Huffman παράγει ένα βέλτιστο κώδικα $C^*(P')$ για ένα αλφάβητο μήκους n και την κατανομή P', η οποία έχει τη μορφή $P'_X(x) = \{p_1, p_2, \cdots, p_{n-1} + p_n\}$.*

• *(Επαγωγικό βήμα). Από τον βέλτιστο κώδικα $C^*(P')$ θα κατασκευάσουμε ένα κώδικα για την κατανομή P, που έχει τη μορφή $P_X(x) = \{p_1, p_2, \cdots, p_{n-1}, p_n\}$, σπάζοντας την κωδική λέξη c'_{n-1} που αντιστοιχεί στην πιθανότητα $p_{n-1} + p_n$ και έχει μήκος l_{n-1}στις δύο κωδικές λέξεις $c'_{n-1}0$ και $c'_{n-1}1$ με πιθανότητες p_{n-1} και p_n αντίστοιχα. Ο κώδικας $C(P)$ θα έχει μέσο μήκος:*

$$\begin{aligned}
&\overline{L}(C_P) = \sum_{i=1}^{n} p_i \cdot l_i = p_1 \cdot l_1 + \cdots + p_{n-1} \cdot (l_{n-1} + 1) + p_n \cdot (l_{n-1} + 1) = \\
&p_1 \cdot l_1 + \cdots + (p_{n-1} + p_n) \cdot l_{n-1} + (p_{n-1} + p_n) = \\
&\overline{L}(C^*(P')) + (p_{n-1} + p_n)
\end{aligned} \tag{5.6}$$

Αντίστοιχα υποθέτουμε ότι υπάρχει ένας βέλτιστος κώδικας $C^*(P)$ για μία συνάρτηση κατανομής $P_X(x) = \{p_1, p_2, \cdots, p_{n-1}, p_n\}$ και θα φτιάξουμε ένα κώδικα για την κατανομή $P'_X(x) = \{p_1, p_2, \cdots, p_{n-1} + p_n\}$. Για να το πράξουμε αυτό συγχωνεύουμε τις δύο τελευταίες κωδικές λέξεις σε μία διαγράφοντας το τελευταίο τους ψηφίο. Αυτό είναι εφικτό γιατί έχουμε υποθέσει ότι ο κώδικας για την κατανομή P θα είναι βέλτιστος οπότε ξέρουμε ότι οι δύο μεγαλύτερες κωδικές λέξεις έχουν ίδιο μήκος και διαφέρουν μόνο στο τελευταίο ψηφίο. Η καινούρια λέξη θα έχει πιθανότητα $p_{n-1} + p_n$ και μήκος $l_n - 1$. Τότε:

$$\begin{aligned}
&\overline{L}(C_{P'}) = \sum_{i=1}^{n} p_i \cdot l_i = p_1 \cdot l_1 + \cdots + p_{n-1} \cdot (l_n - 1) + p_n \cdot (l_n - 1) = \\
&p_1 \cdot l_1 + \cdots + p_{n-1} \cdot l_n + p_n \cdot l_n - (p_{n-1} + p_n) = \\
&\overline{L}(C^*(P)) - (p_{n-1} + p_n)
\end{aligned} \tag{5.7}$$

Από τις σχέσεις (5.2) και (5.3) έπεται:

$$\begin{aligned}
&\overline{L}(C_{P'}) + \overline{L}(C_P) = \overline{L}(C^*(P')) + (p_{n-1} + p_n) + \overline{L}(C^*(P)) - (p_{n-1} + p_n) \Rightarrow \\
&\overline{L}(C_{P'}) + \overline{L}(C_P) = \overline{L}(C^*_{P'}) + \overline{L}(C^*_P) \Rightarrow (\overline{L}(C_{P'}) - \overline{L}(C^*_{P'})) + (\overline{L}(C_{P'} - \overline{L}(C^*_{P'})) = 0
\end{aligned} \tag{5.8}$$

Όμως $\overline{L}(C_{P'}) - \overline{L}(C^*_{P'})$, $\overline{L}(C_P) - \overline{L}(C^*_P) \geq 0$ με άθροισμα μηδέν οπότε αναγκαστικά $\overline{L}(C_{P'}) = \overline{L}(C^*_{P'})$ και $\overline{L}(C_P) = \overline{L}(C^*_P)$. Το τελευταίο συμπέρασμα ολοκληρώνει την απόδειξη μας καθώς μόλις αποδείξαμε ότι ο κώδικας που παράγεται σε κάθε βήμα του αλγορίθμου (συγχώνευση πιθανοτήτων) είναι βέλτιστος και επίσης κάθε κώδικας που θα προέλθει από την επέκταση της μεγαλύτερης κωδικής λέξης θα είναι και αυτός βέλτιστος.

Ένα αυστηρότερο άνω φράγμα για την κωδικοποίηση Huffmnan

[7]

[7] Το φράγμα και η απόδειξη προέρχονται από τη δημοσίευση [**gallager1978variations**]

Θεώρημα 5.2. *Έστω P_1 η μεγαλύτερη πιθανότητα της κατανομής $P_X(x)$ της τ.μ X. Τότε το πλεόνασμα $\overline{L} - H(P) = r$ του κώδικα Huffman φράσσεται από την ποσότητα:*

$$\overline{L} - H(P) = r < P_1 + \sigma,\ \sigma = 1 - log_2 e + log_2(log_2 e) \approx 0.086.$$

Αν το $P_1 \geq \frac{1}{2} \Rightarrow r \leq 2 - H(P_1) - P_1 \leq P_1$

Απόδειξη

Σε κάθε βήμα της διαδικασίας κωδικοποίησης συγχωνεύονται δύο[8] κόμβοι του συνόλου S, δημιουργώντας δύο αδελφούς κόμβους[9]. Αν σε κάθε βήμα αποθηκεύουμε τους κόμβους που συγχωνεύονται σε ένα διάνυσμα, τότε:

1. Το διάνυσμα θα περιέχει κόμβους ταξινομημένους σε αύξουσα σειρά με βάση τις πιθανότητες τους.
2. Αρχικά έχουμε n ασυγχώνευτους κόμβους, τότε μετά τη δημιουργία του δένδρου έχουν δημιουργηθεί ακόμη $n-1$ λόγω των συγχωνεύσεων. Εξαιρώντας τη ρίζα από το διάνυσμα, έχουμε συνολικά $2n-2$ κόμβους
3. Οι κόμβοι $2k$ και $2k-1$ για $0 < k \leq n-1$ είναι αδέρφια.

Τα παραπάνω συμπεράσματα είναι απλό να γίνουν κατανοητά αν αναλογιστούμε την διαδικασία κωδικοποίησης. Έστω ότι ξεκινάμε με ένα κενό διάνυσμα V. Κατά το πρώτο βήμα της κωδικοποίησης επιλέγονται οι δύο μικρότερες πιθανότητες P_1 και P_2 από το S και συγχωνεύονται στον σύνθετο κόμβο $P_1 + P_2$ και γίνονται αδέλφια. Τότε στο V παίρνει τη μορφή:

$$V = \boxed{P_1,\ P_2}$$

και στο σύνολο S εισάγεται ο σύνθετος κόμβος στη κατάλληλη θέση ώστε το S να παραμένει ταξινομημένο.

Στο γενικό βήμα k τις κωδικοποίησης διαλέγουμε δύο κόμβους P_i και P_J με $P_i < P_j$ και τους συγχωνεύουμε. Ύστερα τους εισάγουμε στο V. Αν από τα στοιχεία του V, υπήρχε έστω ένα που έχει μεγαλύτερη πιθανότητα από κάποιον από τους δύο, έστω τον P_i αυτό θα σήμαινε ότι ο κόμβος θα έπρεπε να είχε επιλεχθεί σε νωρίτερο βήμα της κωδικοποίησης το οποίο είναι άτοπο καθώς σε κάθε βήμα ο Huffman διαλέγει τους δύο κόμβους με τις μικρότερες πιθανότητες. Τέλος επειδή οι δύο κόμβοι που εισάγονται στο V κάθε φορά είναι αδέρφια έπεται ότι οι γειτονικοί κόμβοι στο διάνυσμα θα είναι αδέρφια.

Κατά τη διαδικασία της κωδικοποίηση Huffman κάθε φορά που ένα απλός κόμβος συμμετέχει σε μία συγχώνευση αυξάνεται το μονοπάτι που οδηγεί από την τρέχουσα ρίζα του υποδένδρου στο οποίο ανήκει προς εκείνον. Άρα στη σχέση που περιγράφει το μέσο μήκος $\overline{L} = \sum_{i=1}^{n} Pr[X = x_i] l_i$ το μήκος l_i περιγράφει σε πόσες συγχωνεύσεις πήρε μέρος η πιθανότητα $Pr[X = x_i]$. Επειδή κάθε σύνθετος κόμβος περιλαμβάνει το άθροισμα των πιθανοτήτων των παιδιών του, ο πρόγονος κάθε φύλλου θα εμπεριέχει την πιθανότητα του φύλλου. Άρα ένας άλλος τρόπος να βρούμε το l_i είναι να αθροίσουμε τις πιθανότητες όλων του κόμβων εκτός της ρίζας. Άρα:

$$\overline{L} = \sum_{i=1}^{2 \cdot n - 1} Pr[X = x_i]^{10}$$

Επειδή όμως είπαμε οι κόμβοι $Pr[X = x_{2 \cdot k}]$ και $Pr[X = x_{2 \cdot k - 1}]$ είναι αδέρφια το άθροισμα $Pr[X = x_{2 \cdot k}] + Pr[X = x_{2 \cdot k - 1}]$ θα δίνει την πιθανότητα του εσωτερικού κόμβου από τον οποίο προήλθαν τα αδέρφια. Άρα η τελευταία σχέσει μπορεί να γραφεί και ως

[8] Αν μιλάμε για δυαδική κωδικοποίηση

[9] Αδελφοί κόμβοι λέγονται εκείνοι που έχουν κοινό πατέρα

[10] Αυτή η σχέση είναι ισοδύναμη με αυτή που εξέγαμε κατά την ανάλυση του Fano. Στο μόνο που διαφέρουν, είναι ότι στην ανάλυση του Fano μετρούσαμε όλους τους εσωτερικούς κόμβους, δηλαδή συμπεριλαμβάναμε τη ρίζα και εξαιρούσαμε τα φύλλα, ενώ εδώ συμπεριλαμβάνουμε τα φύλλα και εξαιρούμε τη ρίζα. Οι δύο καταστάσεις είναι ισοδύναμες καθώς η ρίζα αποτελεί το άθροισμα όλων των πιθανοτήτων των φύλλων, οπότε περιέχει τις πιθανοτητές τους.

$$\overline{L} = \sum_{i=1}^{2\cdot n-1} Pr[X = x_i] = \sum_{k=1}^{n-1} Pr[X = x_{2\cdot k}] + Pr[X = x_{2\cdot k-1}]$$

11

Όπως έχουμε ήδη αναφέρει κατά την ανάλυση του Fano η εντροπία μία κατανομής μπορεί να υπολογιστεί σε στάδια. Αν πάρουμε τα στάδια που επιβάλλει η κωδικοποίηση Huffman.τότε θα προκύψει η ίδια σχέση που είχαμε και στην ανάλυση του Fano με μία μικρή διαφορά που θα τη συζητήσουμε σε λίγο.

$$H(P)=\sum_{k=1}^{n-1}(Pr[X=x_{2\cdot k}]+Pr[X=x_{2\cdot k-1}])\cdot H\left(\frac{Pr[X=x_{2\cdot k}]}{Pr[X=x_{2\cdot k}]+Pr[X=x_{2\cdot k-1}]}+\frac{Pr[X=x_{2\cdot k-1}]}{Pr[X=x_{2\cdot k}]+Pr[X=x_{2\cdot k-1}]}\right)$$

Η μόνη διαφορά ανάμεσα στις δύο σχέσεις είναι ότι εδώ χρησιμοποιούμε την ιδιότητα πως δύο διαδοχικοί κόμβοι $2\cdot k-1,\ 2\cdot k$ είναι αδέρφια οπότε το άθροισμα τους σίγουρα μας δίνει την πιθανότητα του εσωτερικού κόμβου του οποίου είναι παιδιά.

Σκοπός μας και πάλι είναι να προσπαθήσουμε να φράξουμε την ποσότητα $\overline{L} - H$:

$$\overline{L}-H(P)=\sum_{k=1}^{n-1}(Pr[X=x_{2\cdot k}]+Pr[X=x_{2\cdot k-1}])\cdot\left(1-H\left(\frac{Pr[X=x_{2\cdot k}]}{Pr[X=x_{2\cdot k}]+Pr[X=x_{2\cdot k-1}]}+\frac{Pr[X=x_{2\cdot k-1}]}{Pr[X=x_{2\cdot k}]+Pr[X=x_{2\cdot k-1}]}\right)\right)$$

Σε ένα δένδρο Huffman επειδή η πιθανότητα των εσωτερικών κόμβων είναι ίση με το άθροισμα των πιθανοτήτων των παιδιών τους, σε κάθε επίπεδο που θα είναι πλήρες το άθροισμα των πιθανοτήτων των κόμβων του επιπέδου θα είναι 1. Αν λοιπόν πάρουμε σαν m τον μικρότερο ακέραιο για τον οποίο το l είναι το τελευταίο πλήρες επίπεδο και ο κόμβος $2\cdot m-1$ είναι ο πρώτος κόμβος του επιπέδου $l+1$, τότε μπορούμε να "σπάσουμε" την εντροπία στα εξής δύο κομμάτια:

$$H(P)=\sum_{k=1}^{n-1}(Pr[X=x_{2\cdot k}]+Pr[X=x_{2\cdot k-1}])\cdot H\left(\frac{Pr[X=x_{2\cdot k}]}{Pr[X=x_{2\cdot k}]+Pr[X=x_{2\cdot k-1}]},\ \frac{Pr[X=x_{2\cdot k-1}]}{Pr[X=x_{2\cdot k}]+Pr[X=x_{2\cdot k-1}]}\right)=$$
$$H(P)=\sum_{k=1}^{m-1}(Pr[X=x_{2\cdot k}]+Pr[X=x_{2\cdot k-1}])\cdot H\left(\frac{Pr[X=x_{2\cdot k}]}{Pr[X=x_{2\cdot k}]+Pr[X=x_{2\cdot k-1}]},\ \frac{Pr[X=x_{2\cdot k-1}]}{Pr[X=x_{2\cdot k}]+Pr[X=x_{2\cdot k-1}]}\right)+$$
$$H(P)=\sum_{k=m}^{n-1}(Pr[X=x_{2\cdot k}]+Pr[X=x_{2\cdot k-1}])\cdot H\left(\frac{Pr[X=x_{2\cdot k}]}{Pr[X=x_{2\cdot k}]+Pr[X=x_{2\cdot k-1}]},\ \frac{Pr[X=x_{2\cdot k-1}]}{Pr[X=x_{2\cdot k}]+Pr[X=x_{2\cdot k-1}]}\right)$$

Επειδή όμως οι πιθανότητες των κόμβων στο επίπεδο l έχουν άθροισμα ένα, θα αποτελούν μία κατανομή πιθανότητας. Οπότε η τελευταία σχέση ξαναγράφεται ως:

$$H(P)=\sum_{k=1}^{n-1}(Pr[X=x_{2\cdot k}]+Pr[X=x_{2\cdot k-1}])\cdot H\left(\frac{Pr[X=x_{2\cdot k}]}{Pr[X=x_{2\cdot k}]+Pr[X=x_{2\cdot k-1}]},\ \frac{Pr[X=x_{2\cdot k-1}]}{Pr[X=x_{2\cdot k}]+Pr[X=x_{2\cdot k-1}]}\right)=$$
$$H(Q[X=x_1],\cdots,Q[X=x_m])+$$
$$\sum_{k=m}^{n-1}(Pr[X=x_{2\cdot k}]+Pr[X=x_{2\cdot k-1}])\cdot H\left(\frac{Pr[X=x_{2\cdot k}]}{Pr[X=x_{2\cdot k}]+Pr[X=x_{2\cdot k-1}]},\ \frac{Pr[X=x_{2\cdot k-1}]}{Pr[X=x_{2\cdot k}]+Pr[X=x_{2\cdot k-1}]}\right)$$

,όπου $Q[X = x_i]$ οι πιθανότητες των κόμβων του επιπέδου l. Με βάση τα παραπάνω η διαφορά $\overline{L} - H(P)$ γράφεται ώς:

[11] Στην ουσία το $2n-1$ μετράει πάνω στο πλήθος των κόμβων εκτός της ρίζας ενώ το k μετράει πάνω στα βήματα που κάνει ο αλγόριθμος

$$\overline{L}-H(P)=\sum_{k=1}^{n-1}(Pr[X=x_{2\cdot k}]+Pr[X=x_{2\cdot k-1}])\cdot\left(1-H\left(\frac{Pr[X=x_{2\cdot k}]}{Pr[X=x_{2\cdot k}]+Pr[X=x_{2\cdot k-1}]},\ \frac{Pr[X=x_{2\cdot k-1}]}{Pr[X=x_{2\cdot k}]+Pr[X=x_{2\cdot k-1}]}\right)\right)=$$

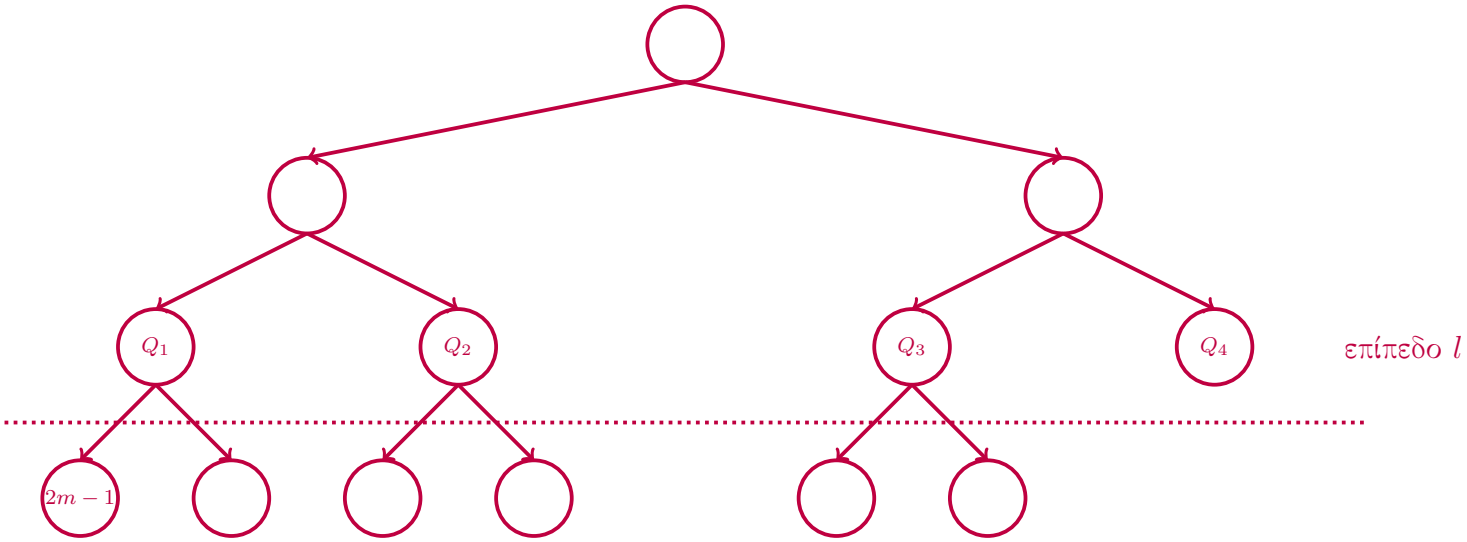
$$\sum_{k=1}^{n-1}(Pr[X=x_{2\cdot k}]+Pr[X=x_{2\cdot k-1}])\cdot 1-\sum_{k=1}^{n-1}(Pr[X=x_{2\cdot k}]+Pr[X=x_{2\cdot k-1}])\cdot H\left(\frac{Pr[X=x_{2\cdot k}]}{Pr[X=x_{2\cdot k}]+Pr[X=x_{2\cdot k-1}]},\ \frac{Pr[X=x_{2\cdot k-1}]}{Pr[X=x_{2\cdot k}]+Pr[X=x_{2\cdot k-1}]}\right)=$$

$$\sum_{k=1}^{m-1}(Pr[X=x_{2\cdot k}]+Pr[X=x_{2\cdot k-1}])+\sum_{k=m}^{n-1}(Pr[X=x_{2\cdot k}]+Pr[X=x_{2\cdot k-1}])-H(Q[X=x_1]),\cdots,Q[X=x_m])-$$

$$\sum_{k=m}^{n-1}(Pr[X=x_{2\cdot k}]+Pr[X=x_{2\cdot k-1}])\cdot H\left(\frac{Pr[X=x_{2\cdot k}]}{Pr[X=x_{2\cdot k}]+Pr[X=x_{2\cdot k-1}]},\ \frac{Pr[X=x_{2\cdot k-1}]}{Pr[X=x_{2\cdot k}]+Pr[X=x_{2\cdot k-1}]}\right)=$$

$$l-H(Q[X=x_1]),\cdots,Q[X=x_m])+\sum_{k=m}^{n-1}(Pr[X=x_{2\cdot k}]+Pr[X=x_{2\cdot k-1}])\cdot\left(1-H\left(\frac{Pr[X=x_{2\cdot k}]}{Pr[X=x_{2\cdot k}]+Pr[X=x_{2\cdot k-1}]},\ \frac{Pr[X=x_{2\cdot k-1}]}{Pr[X=x_{2\cdot k}]+Pr[X=x_{2\cdot k-1}]}\right)\right)$$

,όπου $\sum_{k=1}^{m-1}(Pr[X=x_{2\cdot k}]+Pr[X=x_{2\cdot k-1}])=l$, γιατί αθροίζουμε πάνω στους κόμβους που περιέχονται σε l πλήρη επίπεδα και όπως επισημάναμε το άθροισμα των πιθανοτήτων κάθε πλήρους επιπέδου δίνει 1.



Σχήμα 5.15: Στο σχήμα φαίνεται ξεκάθαρα η ιδέα της απόδειξης. Από την στιγμή που αποφασίσω να χωρίσω το δυαδικό δένδρο που παράγεται από την κωδικοποίηση Huffman στο επίπεδο l που είναι το τελευταίο πλήρως συμπληρωμένο επίπεδο, επί της ουσίας πετυχαίνουμε να χωρίσουμε το δένδρο σε δύο μέρη. Ένα πλήρες δυαδικό δένδρο που εκτείνεται από την ρίζα μέχρι το επίπεδο l και τους κόμβους που περισσεύουν. Το χαρακτηριστικό σε αυτό το χώρισμα είναι το πλήρες δένδρο ύψους l θα αποτελείται από φύλλα των οποίων τα βάρη αποτελούν κατανομή πιθανότητα. Τα φύλλα των υποδένδρων που απομένουν επίσης αποτελούν κατανομή πιθανότητας, αυτήν της τυχαίας μεταβλητής X. Άρα η συνολική εντροπία μπορεί να σπάσει σε δύο μέρη. Στην εντροπία του πλήρους δένδρου ύψους l ($H(Q[X=x_1],\cdots,Q[X=x_m])$) και στην σταθμισμένη εντροπία των υποδένδρων που απομένουν ($\sum_{k=m}^{n-1}(Pr[X=x_{2\cdot k}]+Pr[X=x_{2\cdot k-1}])\cdot H\left(\frac{Pr[X=x_{2\cdot k}]}{Pr[X=x_{2\cdot k}]+Pr[X=x_{2\cdot k-1}]},\ \frac{Pr[X=x_{2\cdot k-1}]}{Pr[X=x_{2\cdot k}]+Pr[X=x_{2\cdot k-1}]}\right)$). Αντίστοιχα το μέσο μήκος κώδικα μπορεί να σπάσει σε δύο μέρη. Σίγουρα από την στιγμή που υπάρχει ένα πλήρες δένδρο ύψους l το ένα μέρος του μέσου κώδικα θα είναι l. Το υπόλοιπο βγαίνει από το μέσο ύψος των υποδένδρων που απομένουν και είναι ίσο με $\sum_{k=m}^{n-1}(Pr[X=x_{2\cdot k}]+Pr[X=x_{2\cdot k-1}]$.

Από την ανάλυση Fano γνωρίζουμε ότι για την $1-H(p,\ 1-p)$ ισχύει

$$1-H(p,\ 1-p)\leq -2p+1\ \forall 0\leq p\leq\frac{1}{2}\Rightarrow\boxed{H(p,1-p)\geq 2p\ \forall 0\leq p\leq\frac{1}{2}}$$

Άρα για κάθε $0\leq p\leq\frac{1}{2}$ ο όρος $H\left(\frac{Pr[X=x_{2\cdot k-1}]}{Pr[X=x_{2\cdot k-1}]+Pr[X=x_{2\cdot k}]},\ \frac{Pr[X=x_{2\cdot k}]}{Pr[X=x_{2\cdot k-1}]+Pr[X=x_{2\cdot k}]}\right)$

φράσσεται από την ποσότητα $2 \cdot \frac{Pr[X = x_{2 \cdot k}]}{Pr[X = x_{2 \cdot k-1}] + Pr[X = x_{2 \cdot k}]}$. Από αυτό έπεται ότι:

$$\sum_{k=m}^{n-1} (Pr[X = x_{2 \cdot k-1}] + Pr[X = x_{2 \cdot k}]) \cdot \left(1 - H\left(\frac{Pr[X = x_{2 \cdot k-1}]}{Pr[X = x_{2 \cdot k-1}] + Pr[X = x_{2 \cdot k}]}, \frac{Pr[X = x_{2 \cdot k}]}{Pr[X = x_{2 \cdot k-1}] + Pr[X = x_{2 \cdot k}]}\right)\right)$$
$$\leq \sum_{k=m}^{n-1} (Pr[X = x_{2 \cdot k-1}] + Pr[X = x_{2 \cdot k}]) \cdot \left(1 - 2 \cdot \frac{Pr[X = x_{2 \cdot k}]}{Pr[X = x_{2 \cdot k-1}] + Pr[X = x_{2 \cdot k}]}\right) =$$
$$\leq \sum_{k=m}^{n-1} (Pr[X = x_{2 \cdot k-1}] + Pr[X = x_{2 \cdot k}]) \cdot \left(1 - \frac{Pr[X = x_{2 \cdot k}] - Pr[X = x_{2 \cdot k}]}{Pr[X = x_{2 \cdot k-1}] + Pr[X = x_{2 \cdot k}]}\right) -$$
$$= \sum_{k=m}^{n-1} (Pr[X = x_{2 \cdot k-1}] + Pr[X = x_{2 \cdot k}]) \cdot \left(\frac{Pr[X = x_{2 \cdot k-1}] - Pr[X = x_{2 \cdot k1}]}{Pr[X = x_{2 \cdot k-1}] + Pr[X = x_{2 \cdot k}]}\right) =$$
$$\sum_{k=m}^{n-1} Pr[X = x_{2 \cdot k-1}] - Pr[X = x_{2 \cdot k}]$$

Με βάση το παραπάνω φράγμα η σχέση $\overline{L} - H(P)$ φράσσεται και αυτή από την ποσότητα:

$$\overline{L} - H(P) \leq l - H(Q[X = x_1]), \cdots, Q[X = x_m]) + \sum_{k=m}^{n-1} Pr[X = x_{2 \cdot k-1}] - Pr[X = x_{2 \cdot k}]$$

Να πούμε σε αυτό το σημείο πως στην ανάλυση του Fano εξάγαμε την σχέση $H(p, 1-p) \leq |1 - 2 \cdot p| \; \forall 0 \leq p \leq 1$ με την ισότητα να ισχύει για $p = \frac{1}{2}$. Άρα η τελευταία σχέση υπό την μορφή:

$$\overline{L} - H(P) \leq l - H(Q[X = x_1]), \cdots, Q[X = x_m]) + \sum_{k=m}^{n-1} \left| Pr[X = x_{2 \cdot k-1}] - Pr[X = x_{2 \cdot k}] \right|$$

θα ισχύει για κάθε $0 \leq Pr[X = x] \leq 1$.

Οι κόμβοι πάνω στους οποίους μετράει το k είναι αυτοί που βρίσκονται από το επίπεδο $l + 1$ και κάτω. Έστω $Pr[X = x_{2 \cdot m-1}]$ ο κόμβος του $l + 1$. Οι κόμβοι λοιπόν που βρίσκονται από το επίπεδο $l + 1$ και κάτω αποτελούν μία φθίνουσα ακολουθία αριθμών. Αυτό ισχύει γιατί αν $x_{2 \cdot j-1}$ ένας εσωτερικός κόμβος τότε το βάρος του $Pr[X = x_{2 \cdot j-1}]$ θα ισούται με το άθροισμα των πιθανοτήτων των παιδιών του, το οποίο αυτόματα σημαίνει ότι τα παιδιά του θα έχουν μικρότερες πιθανότητες από εκείνον. Άρα όσο σε χαμηλότερο επίπεδο πάμε τόσο μικραίνουν οι αριθμοί $Pr[X = x_{2 \cdot j-1}], Pr[X = x_{2 \cdot j}]$. Καταλαβαίνουμε λοιπόν ότι κάθε διαφορά $\left| Pr[X = x_{2 \cdot k-1}] - Pr[X = x_{2 \cdot k}] \right|$ ή $Pr[X = x_{2 \cdot k-1}] - Pr[X = x_{2 \cdot k}]$ φράσσεται από τον πρώτο κόμβο του επιπέδου $l + 1$ δηλαδή από την ποσότητα $Pr[X = x_{2 \cdot m-1}]$.

Αυτό είναι λίγο δύσκολο προς το παρών να το καταλάβουμε διαισθητικά γιατί ο συγγραφέας της απόδειξης έχει ορίσει τον Huffman αντίθετα από ότι εμείς. Συγκεκριμένα εκείνος θέτει το πρώτο στοιχείο που τραβάει από το σύνολο S σαν δεξιό παιδί του κόμβου και το δεύτερο σαν αριστερό. Με αυτό τον τρόπο η μεγαλύτερη από τις δύο πιθανότητες που συγχωνεύει είναι αριστερό παιδί. Για τον λόγο αυτόν όμως έπεται ότι και ο πρώτος κόμβος $x_{2 \cdot m-1}$ του επιπέδου $l + 1$ θα αποτελεί το μεγαλύτερο παιδί του πατέρα του. Επειδή όμως μιλάμε για τον πρώτο κόμβο μετά το πλήρες δένδρου ύψους l έπεται ότι και ο πατέρα του θα είναι και αυτός αριστερό παιδί και ο πρώτο κόμβος του επιπέδου l οπότε θα είναι ο μεγαλύτερος από τα παιδιά του πατέρα του, δηλαδή του παππού του κόμβου $x_{2 \cdot m-1}$. Συνεχίζοντας να πηγαίνουμε προς τα πάνω σε επίπεδα θα φράσουμε κάποια στιγμή στο επίπεδο που βρίσκεται ακριβώς μετά την ρίζα. Επειδή όπως είδαμε οι πρόγονοι του κόμβου $x_{2 \cdot m-1}$ αποτελούν τα πρώτα παιδιά των επιπέδων τους, έπεται ότι ο πρόγονος του $x_{2 \cdot m-1}$ στο επίπεδο μετά τη ρίζα θα είναι το αριστερό παιδί που έχει το μεγαλύτερο βάρος. Άρα επαγωγικά σκεπτόμενοι

καταλαβαίνουμε ότι κόμβος $x_{2\cdot m-1}$ έχει το μεγαλύτερο βάρος από όλους του κόμβους του επιπέδου του. Για το λόγο αυτό θα ισχύει ότι $Pr[x_{2\cdot m-1}] \geq Pr[x_{2\cdot j-1}]$ $j = m+1, \cdots, n-1$. Άρα για αυτό το λόγο κάθε διαφορά $Pr[X = x_{2\cdot k-1}] - Pr[X = x_{2\cdot k}]$ θα φράσσεται από την $Pr[x_{2\cdot m-1}]$

$$\overline{L} - H(P) \leq l - H(Q[X = x_1], \cdots, Q[X = x_m]) + \sum_{k=m}^{n-1} Pr[X = x_{2\cdot k-1}] - Pr[X = x_{2\cdot k}] \leq$$
$$l - H(Q[X = x_1], \cdots, Q[X = x_m]) + Pr[X = x_{2\cdot m-1}]$$

Δηλαδή:

$$\overline{L} - H(P) \leq l - H(Q[X = x_1]), \cdots, Q[X = x_m]) + Pr[X = x_{2\cdot m-1}] \; \forall 0 \leq Pr[X = x] \leq 1$$

Τώρα θα ασχοληθούμε με το φράξιμο της ποσότητας $H(Q[X = x_1]), \cdots, Q[X = x_m])$ για τις διαφορετικές περιπτώσεις του P_1 που αποτελεί την μέγιστη πιθανότητα της σ.μ.π $P_X(x)$.

• Αν $P_1 \geq \frac{1}{2}$, τότε αναγκαστικά το δένδρο θα χωριστεί στο πρώτο επίπεδο, οπότε το $l = 1$ και $H(P_1,\ 1 - P_1)$.

$$\begin{aligned}
&\overline{L} - H(P) \leq 1 - H(P_1,\ 1 - P_1) + Pr[X = x_{2\cdot m-1}] \Rightarrow \\
&\overline{L} - H(P) \leq 1 - H(P_1,\ 1 - P_1) + (1 - P_1) \Rightarrow \\
&\overline{L} - H(P) \leq 2 - H(P_1,\ 1 - P_1) - P_1 \overset{H(P_1,\ 1-P_1)\leq 1}{\Rightarrow} \\
&\overline{L} - H(P) \leq 2 - 1 - P_1 \Rightarrow \\
&\overline{L} - H(P) \leq 1 - P_1 \overset{P_1 \geq 1-P_1}{\Rightarrow} \\
&\overline{L} - H(P) \leq P_1
\end{aligned}$$

• Στην περίπτωση που $P_1 = \frac{1}{2}$, έχουμε:

$$\begin{aligned}
&\overline{L} - H(P) = 1 - H(P_1,\ 1 - P_1) + Pr[X = x_{2\cdot m-1}] \Rightarrow \\
&\overline{L} - H(P) = 1 - 1 + P_1 \Rightarrow \\
&\overline{L} - H(P) = P_1
\end{aligned}$$

• Στην περίπτωση που $P_1 = 1$, έχουμε:

$$\begin{aligned}
&\overline{L} - H(P) = 0 - H(P_1,\ 1 - P_1) + Pr[X = x_{2\cdot m-1}] \Rightarrow \\
&\overline{L} - H(P) = 0 - 0 + P_1 \Rightarrow \\
&\overline{L} - H(P) = P_1 = 1
\end{aligned}$$

Άρα συνολικά μόλις αποδείξαμε ότι για $P_1 \geq \frac{1}{2}$ ισχύει $\overline{L} - H(P) \leq 2 - H(P_1,\ 1 - P_1) - P_1 \leq P_1$ με την ισότητα ισχύει μόνο στις περιπτώσεις $P_1 = \frac{1}{2}, 1$. Η ανισότητα θα ισχύει και ανάμεσα στο $\frac{1}{2}$ και 1 λόγω της κυρτότητας της συνάρτησης.

Στην περίπτωση που όλες οι κωδικές λέξεις έχουν το ίδιο μήκος n_1 παίρνουμε σαν ύψος διαμέρισης το $l = n_1 - 1$ ενώ διαφορετικά $l = n_1$. Και στις δύο περιπτώσεις πάντως θα ισχύει ότι $Pr[X = x_{2\cdot -1}] \leq P_1$ και οι κόμβοι που βρίσκονται μετά το επίπεδο l μπορούν να διαταχθούν σε μία αύξουσα ακολουθία.Από όλους του κόμβους εμείς παίρνουμε ένα σύνολο $\mathcal{Q}$ για τους οποίους ισχύει $Q'[X = x_1] \geq Q'[X = x_2] \geq \cdots \geq Q'[X = x_L] \geq \frac{Q'[X = x_1]}{2}$ και $\sum_{i=1}^{L} Q'[X = x_i] = 1$. Μέχρι στιγμής ξέρουμε ότι:

$$\overline{L} - H(P) \leq l - H(Q[X = x_1], \cdots, Q[X = x_m]) + Pr[X = x_{2 \cdot m-1}] \; \forall 0 \leq Pr[X = x] \leq 1 \Rightarrow$$
$$\overline{L} - H(P) \leq l - H(Q[X = x_1], \cdots, Q[X = x_m]) + P_1 \; \forall 0 \leq Pr[X = x] \leq 1$$

Στην προηγούμενη ενότητα παίρνοντας την περίπτωση που $\frac{1}{2} \leq Pr[X = x] \leq 1$ καταφέραμε να αποδείξουμε ότι $\overline{L} - H(P) \leq 2 - H(P_1, 1 - P_1) - P_1 \leq P_1$. Σε αυτή την περίπτωση έχοντας στο μυαλό μας την δομή που δημιουργεί το χώρισμα του δένδρου θα προσπαθήσουμε να δούμε πως φράσσεται η ποσότητα $H(Q[X = x_1], \cdots, Q[X = x_m])$ για οποιαδήποτε κατανομή μας τύχει.

Επειδή η H(*) είναι κοίλη έπεται ότι η -H(*) θα είναι κυρτή οπότε ξέρουμε ότι το ελάχιστο για την -H(*) θα εμφανίζεται όταν μεγιστοποιείται η H(*). Γνωρίζουμε όμως ότι η εντροπία μίας σ.μ.π μεγιστοποιείται όταν αποτελείται από ισοπίθανα ενδεχόμενα. Παίρνουμε λοιπόν ότι οι n από τις L διατεταγμένες πιθανότητες $(Q'[X = x_1] \geq Q'[X = x_2] \geq \cdots \geq \frac{Q'[X = x_1]}{2}$ ισούνται με την $(Q'[X = x_1]$ ενώ οι υπόλοιπες $n - L$ με την $\frac{Q'[X = x_1]}{2}$. Επειδή πρέπει να σχηματίζουν κατανομή οι πιθανότητες έπεται ότι $\sum_{i=1}^{L} Q'[X = x_i] = 1 \Rightarrow nQ'[X = x_1] + (L - n)\frac{Q'[X = x_1]}{2} = 1 \Rightarrow Q'[X = x_1] = \frac{2}{L + n}$. Τότε η εντροπία της συγκεκριμένη κατανομής θα είναι:

$$H(Q'1, \cdots, Q' + L) = nQ'[X = x_1] \log \frac{1}{Q'[X = x_1]} + (L - n)\frac{Q'[X = x_1]}{2} \log \frac{2}{Q'[X = x_1]} =$$
$$- nQ'[X = x_1] \log Q'[X = x_1] + (L - n)\frac{Q'[X = x_1]}{2} \log 2 - (L - n)\frac{Q'[X = x_1]}{2} \log Q'[X = x_1] =$$
$$- nQ'[X = x_1] \log Q'[X = x_1] - (L - n)\frac{Q'[X = x_1]}{2} \log Q'[X = x_1] + (L - n)\frac{Q'[X = x_1]}{2} =$$
$$\frac{-L - n'}{Q}[X = x_1] 2 \log Q'[X = x_1] + \frac{L - n}{L + n} = - \log Q'[X = x_1] + \frac{L - n}{L + n} = - \log \frac{2}{L + n} + \frac{L - n}{L + n}$$

Άρα το $\min_{Q'} H(Q'[X = x_1], \cdots, Q'[X = x_L]) = \min_{1 \leq n \leq L} \left[- \log \frac{2}{L + n} + \frac{L - n}{L + n} \right]$

Αν αφήσουμε το n να πάρει και άλλες τιμές πέρα από τις ακέραιες, τότε το παραπάνω πρόβλημα ελαχιστοποίηση λύνεται αναλυτικά.

Θεωρούμε την:

$$f(\frac{2}{L + n}) = \frac{L - n}{L + n} = - \log \frac{2}{L + n} + \frac{L - n}{L + n} = \frac{L - n}{L + n} = - \log \frac{2}{L + n} + \frac{L - n + L - L}{L + n} = \frac{L - n}{L + n} =$$
$$\log \frac{2}{L + n} - \frac{L + n}{L + n} + L \cdot \frac{2}{L + n} = log \frac{2}{L + n} - 1 + L \cdot \frac{2}{L + n} \Rightarrow$$
$$f(x) = -1 + L \cdot x - \log_2(x)$$

όπου $x = \frac{2}{L + n}$.

$$\frac{df(x)}{dx} = 0 \Rightarrow L + -\frac{1}{x} \cdot \log_2 e = 0 \Rightarrow x = \frac{\log_2 e}{L}$$

Αντικαθιστώντας πάνω την θέση ελαχίστου στην f έχουμε:

$$f(\frac{\log_2 e}{L}) = -1 + L \cdot \frac{\log_2 e}{L} - log_2 \left(\frac{\log_2 e}{L} \right) = -1 + \log_2 e - log_2(log_2 e) + log_2 L$$

$1 - \log_2 e + \log_2(\log_2 e) \approx 0.086$

5.5 Αριθμητική κωδικοποίηση

5.5.1 Παρουσίαση της μεθόδου

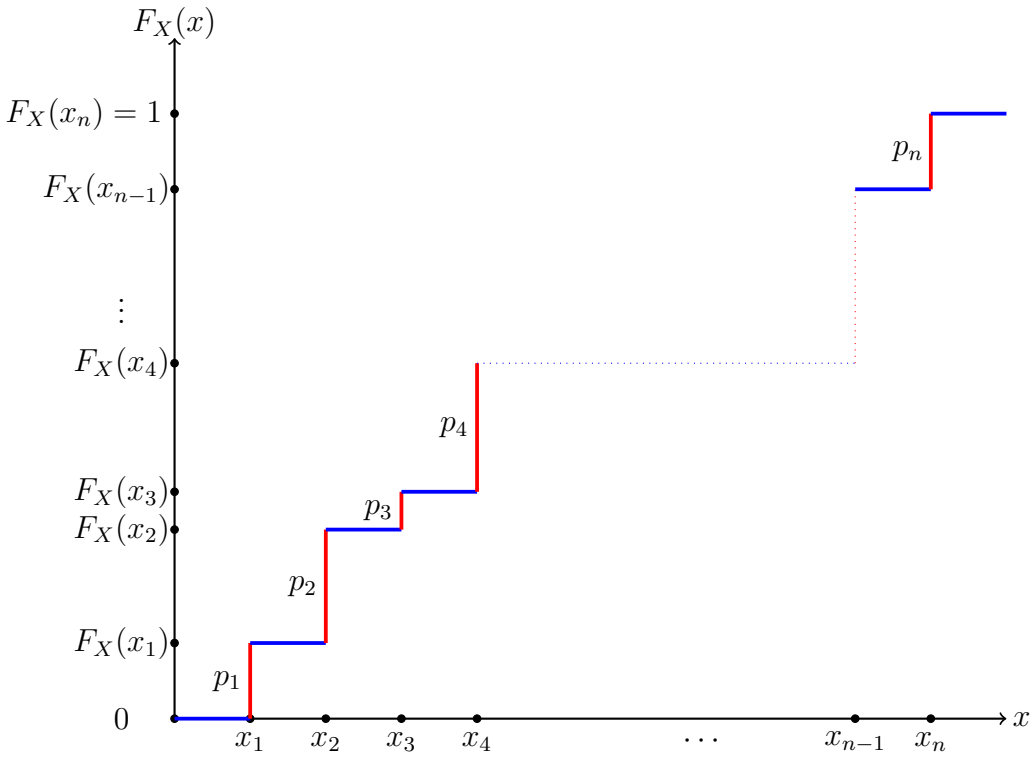
Όπως είδαμε στην κωδικοποίηση Huffman αν το μέγεθος του αλφαβήτου είναι μικρό ή η μέγιστη πιθανότητα είναι αρκετά μεγάλη, αρχίζει το μέσο μήκος κώδικα που παράγεται να απέχει σημαντικά από την εντροπία της πηγής. Το όριο που πλησιάζει αυτή η απόκλιση ξέρουμε ότι θεωρητικά είναι μέχρι 1 κωδικό σύμβολο. Αν όμως για κάθε τιμή της τυχαίας μεταβλητής αρχίζουμε και απέχουμε ένα κωδικό σύμβολο, τότε καταλαβαίνουμε ότι το συμπιεσμένο σύνολο δεδομένων θα απέχει κατά M κωδικά σύμβολα, όπου M το μέγεθος του συνόλου δεδομένων, από την βέλτιστη αναπαράστασή του. Μία λύση όπως έχουμε δει και θεωρητικά είναι να μην κωδικοποιούμε τις τιμές μίας τυχαίας μεταβλητής ξεχωριστά αλλά ακολουθίες αυτών. Κάτι τέτοιο όμως αυξάνει το μέγεθος του κώδικα που παράγει ο Huffman εκθετικά, γεγονός που καθιστά την προσέγγιση αυτή μη εφαρμόσιμη. Την λύση στο παραπάνω πρόβλημα έρχεται να μας τη δώσει η αριθμητική κωδικοποίηση.

Η αριθμητική κωδικοποίηση έχει τη λογική να αντιστοιχεί σε κάθε τιμή ή σε κάθε ακολουθία τιμών της τυχαίας μεταβλητής μία ετικέτα την οποία ύστερα κωδικοποιεί. Ας βάλουμε σε μαθηματικά πλαίσια το παραπάνω πρόβλημα. Το δεδομένο του προβλήματος είναι ότι έχουμε ένα σύνολο δεδομένων που θέλουμε να συμπιέσουμε το οποίο μοντελοποιείται με μία τυχαία μεταβλητή X που παίρνει τιμές σε ένα πεπερασμένο σύνολο $\mathcal{X}$ και ακολουθεί τον κανόνα

$$P_X(x) = \{Pr[X = x_1], Pr[X = x_2], \cdots \ Pr[X = x_{|\mathcal{X}|}]\}.$$

Για παράδειγμα αν το σύνολο δεδομένων είναι μία συλλογή κειμένων τότε με την τυχαία μεταβλητή X μπορούμε να μοντελοποιήσουμε τα σύμβολα που εμφανίζονται στα κείμενα και η σχετική συχνότητα εμφάνισής τους να αποτελεί τον κανόνα που ακολουθούν. Το ζητούμενο του προβλήματος είναι να καταφέρουμε να αντιστοιχίσουμε τις τιμές της τυχαίας μεταβλητής X σε κωδικές λέξεις με σύμβολα από ένα κωδικό αλφάβητο F έτσι ώστε η συμπιεσμένη αναπαράσταση του συνόλου δεδομένων να είναι η βέλτιστη.

Έστω μία τυχαία μεταβλητή X που παίρνει τιμές σε ένα πεπερασμένο σύνολο $\mathcal{X}$ και ακολουθεί τον κανόνα $P_X(x) = \{Pr[X = x_1], Pr[X = x_2], \cdots Pr[X = x_n]\}$. Τότε κάθε τιμή της $x_i \in \mathcal{X}$ με $Pr[X = x_i] > 0$ αντιστοιχίζεται κατά μοναδικό τρόπο στον αριθμό $F_X(x_i) = \sum_{k=0}^{i} Pr[X = x_k]$. Αν παρατηρήσουμε προσεκτικά το γράφημα της αθροιστικής κατανομής πιθανότητας μίας τυχαίας μεταβλητής θα δούμε ότι οι αριθμοί $F_X(x_i),\ x_i \in \mathcal{X}$ διαμερίζουν κατά μοναδικό τρόπο το διάστημα [0,1] των πραγματικών αριθμών. Άρα κάθε πραγματικός αριθμός που ανήκει στο διάστημα $(F_X(x_i - 1), F_X(x_i))$ μπορεί να χρησιμοποιηθεί για να αναπαραστήσει με μοναδικό τρόπο την τιμή x_i της τυχαίας μεταβλητής. Αυτό το συμπέρασμα αποτελεί τη βάση της αριθμητικής κωδικοποίησης.



Σχήμα 5.16: Η αθροιστική κατανομή μίας διακριτής τυχαίας μεταβλητής. Παρατηρείστε ότι οι αθροιστικές πιθανότητες χωρίζουν με μοναδικό τρόπο το διάστημα [0,1] για μία συγκεκριμένη κατανομή $P_X(x)$.

Μία πρώτη προσέγγιση κωδικοποίησης

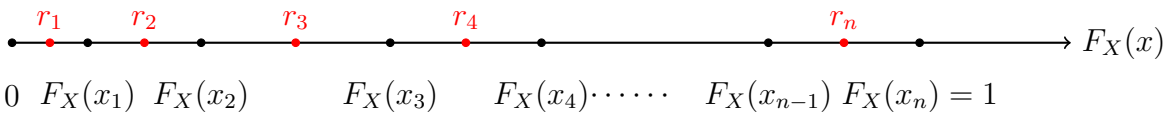
Με βάση τα όσα ειπώθηκαν παραπάνω για να κωδικοποιήσουμε τις τιμές μίας τυχαίας μεταβλητής ακολουθούμε τα παρακάτω βήματα:

Διαδικασία κωδικοποίησης

1. Υπολογίζουμε τις αθροιστικές πιθανότητες $\{F_X(x_i)\}$και διαμερίζουμε το διάστημα $I = [0, 1]$ στα ξένα υποδιαστήματα $I_i = [F_X(x_{i-1}), F_X(x_i))$.

2. Επιλέγουμε σαν ετικέτα της τιμής x_i της X, τον πραγματικό αριθμό που αντιστοιχεί στο μέσο του διαστήματος $I_i = [F_X(x_{i-1}), F_X(x_i))$: $r_i = \sum_{k=1}^{i-1} Pr[X = x_k] + \frac{1}{2} \cdot Pr[X = x_i] = F_X(x_{i-1}) + \frac{F_X(x_i) - F_X(x_{i-1})}{2}$

3. Γράφουμε τον πραγματικό αριθμό r_i σε ένα ανάπτυγμα με βάση τον πληθάριθμο του κωδικού αλφαβήτου. Το πλήθος των ψηφίων του αναπτύγματος που θα χρησιμοποιήσουμε υπολογίζεται από την σχέση $l_i = \lceil log\frac{1}{Pr[X = x_i]} \rceil + 1$



Σχήμα 5.17: Γραφική αναπαράσταση της μεθόδου

Διαδικασία αποκωδικοποίησης:

1. Βρίσκουμε τα διαστήματα $[F_X(x_{i-1}), F_X(x_i))$
2. Υπολογίζουμε το μέσο του διαστήματος r_i
3. Το μετατρέπουμε σε μία δυαδική ακολουθία με μήκος $l_i = \lceil log\frac{1}{Pr[X=x_i]} \rceil + 1$

Παράδειγμα 5.6. *Να κωδικοποιήσετε σε bits τις τιμές της τυχαίας μεταβλητής* $X = \{x_1, x_2, x_3\}$ *που ακολουθούν τον κανόνα* $P_X(x) = \{\frac{1}{2}, \frac{1}{4}, \frac{1}{4}\}$. *Με βάση τον κώδικα που παράγατε συμπιέστε την ακολουθία* $x_1x_1x_3$.

Λύση

1. *Υπολογίζουμε τις αθροιστικές πιθανότητες:*

$$F_x(x_1) = Pr[X = x_1] = \frac{1}{2}$$
$$F_x(x_2) = Pr[X = x_1] + Pr[X = x_2] = \frac{1}{2} + \frac{1}{4} = \frac{3}{4}$$
$$F_x(x_2) = Pr[X = x_1] + Pr[X = x_2] + Pr[X = x_3] = \frac{1}{2} + \frac{1}{4} + \frac{1}{4} = 1$$

Άρα το διάστημα $I = [0, 1]$ *χωρίζεται στα υποδιαστήματα* $I_1 = [0, \frac{1}{2}),\ I_2 = [\frac{1}{2}, \frac{3}{4}),\ I_3 = [\frac{3}{4}, 1]$

2. *Υπολογίζουμε τα* r_i*:*

$$r_1 = 0 + \frac{F_x(x_1) - F_x(x_0)}{2} = \frac{\frac{1}{2}}{2} = \frac{1}{4}$$
$$r_2 = F_x(x_1) + \frac{F_x(x_2) - F_x(x_1)}{2} = \frac{1}{2} + \frac{\frac{3}{4} - \frac{1}{2}}{2} = \frac{5}{8}$$
$$r_3 = F_x(x_2) + \frac{F_x(x_3) - F_x(x_2)}{2} = \frac{3}{4} + \frac{1 - \frac{3}{4}}{2} = \frac{7}{8}$$

3. *Αναπαριστούμε τα* r_i *δυαδικά. Για το* x_1 *θα χρησιμοποιήσουμε* $\lceil log\frac{1}{Pr[X=x_1]} \rceil + 1 = \lceil log2 \rceil + 1 = 2$ *κωδικά σύμβολα, για το* x_2 *και το* x_3 $\lceil log\frac{1}{Pr[X=x_2]} \rceil + 1 = \lceil log4 \rceil + 1 = 3$ *σύμβολα. Άρα έχουμε:*

$$r_1 = \frac{1}{4} = 0 \cdot \frac{1}{2^1} + 1 \cdot \frac{1}{2^2} = 01$$
$$r_2 = \frac{5}{8} = 1 \cdot \frac{1}{2^1} + 0 \cdot \frac{1}{2^2} + 1 \cdot \frac{1}{2^3} = 101$$
$$r_3 = \frac{7}{8} = 1 \cdot \frac{1}{2^1} + 1 \cdot \frac{1}{2^2} + 1 \cdot \frac{1}{2^3} = 111$$

Άρα η κωδικοποίηση της ακολουθίας $x_1x_1x_3$ *είναι η* 0101111.

Παράδειγμα 5.7. *Αποκωδικοποιήστε τη συμβολοσειρά του προηγούμενου παραδείγματος.*

Λύση

Επειδή το μήκος των κωδικών λέξεων $l_i = \lceil log\frac{1}{Pr[X=x_i]} + 1 \rceil$ *ικανοποιεί την ανισότητα Kraft ξέρουμε ότι ο κώδικας είναι στιγμιαίος. Άρα είμαστε σε θέση να καταλαβαίνουμε πότε τελειώνει μία κωδική λέξη χωρίς*

να έχουμε ανάγκη γνώσης της επόμενης. Με βάση την ιδιότητα αυτή κάθε ακολουθία κωδικών συμβόλων μπορεί ανά πάσα στιγμή να μετατραπεί σε ακολουθία κωδικών λέξεων. Άρα υπολογίζοντας τα όπως και στην διαδικασία της κωδικοποίησης τα διαστήματα $I_1 = [0, \frac{1}{2}),\ I_2 = [\frac{1}{2}, \frac{3}{4}),\ I_3 = [\frac{3}{4}, 1]$ *και υπολογίζοντας τις ετικέτες που παράγονται από αυτά κατασκευάζουμε τον κώδικα για την τυχαία μεταβλητή X. Έτσι* $0101111 \rightarrow 01 - 01 - 111 \Rightarrow x_1 = x_1 - x_3$.

Κωδικοποίηση ακολουθιών ανεξάρτητων και ισόνομων τυχαίων μεταβλητών

Στην αρχή της παραγράφου διατυπώσαμε τον τρόπο με τον οποίο μέσω της αριθμητικής κωδικοποίησης μπορούμε να αντιστοιχίσουμε με αποδοτικό τρόπο ετικέτες, δηλαδή πραγματικούς αριθμούς $x \in [0, 1]$ σε τιμές μια διακριτής τυχαίας μεταβλητής X. Τώρα θα προσπαθήσουμε να επεκτείνουμε την παραπάνω μέθοδο ώστε να μπορέσουμε να αντιστοιχίσουμε ακολουθίες ανεξάρτητων και ισόνομων τυχαίων μεταβλητών σε έναν πραγματικό αριθμό $x \in [0, 1]$. Για να καταφέρουμε το ζητούμενο απαιτείται να διευρύνουμε την προσέγγιση της κωδικοποίησης που αναλύσαμε στην προηγούμενη ενότητα.

Από τον κλάδο των πιθανοτήτων γνωρίζουμε ότι ισχύουν οι δύο παρακάτω σχέσεις για διακριτές τυχαίες μεταβλητές.

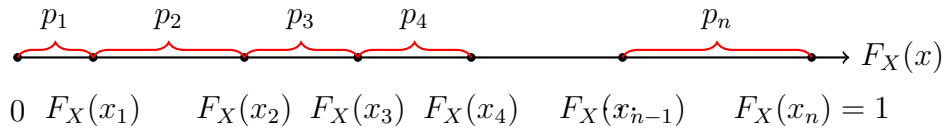
$$F_X(x) = \sum_{a \leq x} Pr[X = a]$$

$$Pr[X = x_i] = \sum_{k=1}^{i} Pr[X = k] - \sum_{k=1}^{i-1} Pr[X = k] = F_X(x_i) - F_X(x_i)$$

Σύμφωνα με τα όσα έχουν αναπτυχθεί μέχρι στιγμής μπορούμε να δούμε την πιθανότητα κάθε συμβόλου, $Pr[X = x_i]$, ως το μήκος του διαστήματος ή αλλιώς το ποσοστό μήκους που καταλαμβάνει η x_i στο διάστημα [0,1] με [12]$\lambda([0, 1]) = 1$. Αν έχουμε μία ακολουθία ανεξάρτητων και ισόνομων τυχαίων μεταβλητών τότε:

$$Pr[X = x_1, X = x_2, \cdots, X_n = x_n] = Pr[X = x_1] \cdot Pr[X = x_2] \cdots Pr[X = x_n]$$

Ας προσπαθήσουμε να δούμε σχηματικά τι μας λέει η παραπάνω σχέση. Όπως και πριν αρχίζουμε με ένα διάστημα $I = [0, 1]$ μήκους 1, το οποίο το χωρίζουμε σε n κομμάτια $[F_X(x_{i-1}, F_X(x_i))$ μήκους $\lambda([F_X(x_{i-1}, F_X(x_i))) = F_X(x_i) - F_X(x_{i-1}) = Pr[X = x_i]$ το κάθε ένα.

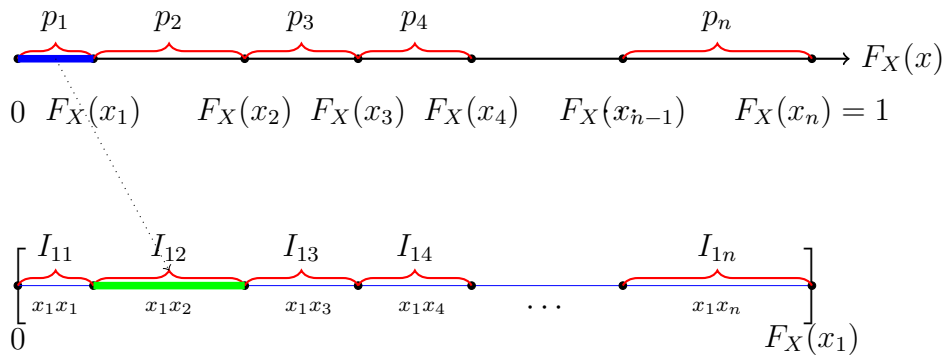


Σχήμα 5.18: Αρχικά το [0,1] διαμερίζεται σε n κομμάτια μήκους $p_i = Pr[X = x_i]$

.

Στην έκφραση $Pr[X = x_1] \cdot Pr[X = x_2] \cdots Pr[X_n = x_n]$. η θέση της $Pr[X = x_1]$ δείχνει ότι πρώτα επιλέχθηκε το διάστημα $I_1 = [0, F_X(x_1))$ με μήκος $Pr[X = x_1]$. Ο πολλαπλασιασμός $Pr[X = x_1] \cdot Pr[X = x_2]$ σχηματικά σημαίνει ότι από το μήκος $Pr[X = x_1]$ επιλέχθηκε μέρος ανάλογο του $Pr[X = x_2]$. Άρα η πιθανότητα να συμβεί η συμβολοσειρά x_1x_2 αντιστοιχίζεται σε μήκος $Pr[X = x_1] \cdot Pr[X = x_2]$ ή αλλιώς σε ένα κομμάτι του I_1 με μήκος ανάλογο του $Pr[X = x_2]$. Το μήκος όμως $Pr[X = x_1] \cdot Pr[X = x_2]$ αντιστοιχεί σε ένα διάστημα I_{12}. Για να βρούμε ποιο είναι το παραπάνω διάστημα παίρνουμε το διάστημα I_1 που επιλέχθηκε πρώτα και το χωρίζουμε σε n κομμάτια κάθε ένα από τα οποία έχει μήκος ανάλογο της $Pr[X = x_i]$. Έπειτα επιλέγουμε το δεύτερο διάστημα που δημιουργήθηκε.

[12]Όπου λ το μέτρο Lebesque



Σχήμα 5.19: Για να κωδικοποιηθεί η συμβολοσειρά x_1x_2 επιλέγεται πρώτα το I_1 (μπλε χρώμα) το οποίο διαμερίζεται σε n κομμάτια με μήκη ανάλογα των $Pr[X = x_i]$. Ύστερα επιλέγουμε το δεύτερο διάστημα I_{12} (πράσινο χρώμα) με μήκος p_1p_2

Τώρα θα εξάγουμε τη φόρμουλα που μας δίνει τα άκρα των διαστημάτων που παράγονται από την υποδιαίρεση του I_1. Το I_1 θα πρέπει να διαμεριστεί με τέτοιον τρόπο ώστε το συνολικό μήκος των υποδιαστημάτων να αθροίζει στο $Pr[X = x_1]$ καθώς

$$Pr[X = x_1] = \sum_{i=1}^{n} Pr[X_1 = x_1, X_2 = j] = \sum_{i=1}^{n} l(I_{1j}),$$

όπου $l(I_{1j})$ το μήκος του υποδιαστήματος I_{ij}.

Για να βρούμε τι ποσοστό του μήκους αντιστοιχεί στο I_1 σε κάθε πιθανότητα $Pr[X = x_j]$ πολλαπλασιάζουμε το μήκος του διαστήματος με την αντίστοιχη πιθανότητα:

$$l(I_{1j}) = (F_X(x_1) - F_X(0)) \cdot Pr[X = x_j] = Pr[X = x_1] \cdot Pr[X = x_j]$$

Αφού υπολογίσουμε τα $l(1j)$ διαμερίζουμε το I_1 ως εξής

1. Το I_{11} ξεκινάει από το $F_X(0)$ και τερματίζει στο σημείο $F_X(0) + l(I_{ij}) = F_X(0) + (F_X(x_1) - F_X(0)) \cdot Pr[X = x_1]$. Άρα το

$$I_{11} = [0, \quad F_X(0) + (F_X(x_1) - F_X(0) \cdot Pr[X = x_1]) = [0, \quad F_X(0) + (F_X(x_1) - F_X(0)) \cdot F_X(x_1))$$

2. Το I_{12} ξεκινάει από το $F_X(0) + (F_X(x_1) - F_X(0)) \cdot F_X(x_1)$ και τερματίζει στο σημείο $F_X(0) + (F_X(x_1) - F_X(0)) \cdot Pr[X = x_1] + l(I_{12})$. Όμως:

$$\begin{aligned}
&F_X(0) + (F_X(x_1) - F_X(0)) \cdot Pr[X = x_1] + l(I_{12}) = \\
&F_X(0) + (F_X(x_1) - F_X(0)) \cdot Pr[X = x_1] + (F_X(x_1) - F_X(0)) \cdot Pr[X = x_2] = \\
&F_X(0) + (F_X(x_1) - F_X(0)) \cdot (Pr[X = x_1] + Pr[X = x_2]) = F_X(0) + (F_X(x_1) - F_X(0)) \cdot F_X(x_2) \\
&\text{Άρα } I_{12} = [F_X(0) + (F_X(x_1) - F_X(0)) \cdot F_X(x_1), \quad F_X(0) + (F_X(x_1) - F_X(0)) \cdot F_X(x_2))
\end{aligned}$$

3. Γενικά το $i-$ οστό διάστημα I_{1i} έχει την μορφή:

$$I_{1i} = [F_X(0) + (F_X(x_1) - F_X(0)) \cdot F_X(i-1), \quad F_X(0) + (F_X(x_1) - F_X(0)) \cdot F_X(i-1))$$

Σημείωση Η τελευταία σχέση που μας δίνει τη γενική μορφή του $i-$οστού υποδιαστήματος δουλεύει ακόμα και αν τον πρώτο στοιχείο της ακολουθίας δεν είναι το x_1 αλλά κάποιο τυχαίο x_j. Τότε απλά αλλάζουν τα εξής πράγματα:

1. Το αρχικό διάστημα δεν είναι το $[F_X(0),\ \ F_X(x_1))$ αλλά το $[F_X(x_{j-1}),\ \ F_X(x_j))$

2. Το διάστημα I_{j1} δεν αρχίζει από το $F_X(0)$ αλλά από το $F_X(x_{j-1})$.

Άρα κατά αναλογία το διάστημα I_{ji} θα έχει την μορφή:

$$I_{ji} = [F_X(x_{j-1}) + (F_X(x_j) - F_X(x_{j-1})) \cdot F_X(i-1),\ \ F_X(x_{j-1}) + (F_X(x_j) - F_X(x_{j-1})) \cdot F_X(i))$$

Αν θέσουμε το άνω άκρο του $[F_X(x_{j-1}),\ \ F_X(x_j))$ με $U_j = F_X(x_j)$ και το κάτω άκρο με $L_{j-1} = F_X(x_j)$, τότε η σχέση που δίνει το διάστημα I_{ji} γράφεται στην κομψότερη μορφή:

$$I_{ji} = [L_j + (U_j - L_j) \cdot F_X(x_{i-1}),\ \ L_j + (U_j - L_j) \cdot F_X(x_i))$$

Συνεχίζοντας με τη ίδια λογική για να κωδικοποιήσουμε την ακολουθία $x_1x_2x_3$ θα χωρίσουμε το διάστημα I_{12} σε n κομμάτια με μήκος ανάλογο της $Pr[X = x_i]$. Τότε το διάστημα I_{123} είναι το τρίτο διάστημα κατά σειρά της καινούρια διαμέρισης. Θυμίζουμε ότι το I_{12} έχει άνω άκρο τον αριθμό $F_X(0) + (F_X(x_1) - F_X(0)) \cdot F_X(x_2)$ και κάτω άκρο τον αριθμό $F_X(0) + (F_X(x_1) - F_X(0)) \cdot F_X(x_1)$

Αν θέσουμε το κάτω άκρο του I_{12} ίσο με τον αριθμό L_{11} και το άνω άκρο με U_{12} τότε το πρώτο διάστημα μετά τη διαμέριση του I_{12} θα έχει τη μορφή:

$$\begin{aligned}
&I_{121} = [L_{12},\ \ L_{12} + l(I_{123})) = \\
&[L_{12},\ \ L_{12} + (U_{12} - L_{12})) \cdot Pr[X_3 = x_1]) = \\
&[L_{12},\ \ L_{12} + (U_{12} - L_{12}) \cdot F_X(x_1))
\end{aligned}$$

Ανάλογα το I_{122} θα έχει σαν κάτω άκρο το άνω άκρο του προηγούμενου διαστήματος και σαν άνω άκρο τον αριθμό

$$\begin{aligned}
&L_{12} + (U_{12} - L_{12}) \cdot Pr[X_3 = x_1]) + (U_{12} - L_{12}) \cdot Pr[X_3 = x_2] = \\
&L_{12} + (U_{12} - L_{12}) \cdot (Pr[X_3 = x_1] + Pr[X_3 = x_2]) = \\
&L_{12} + (U_{12} - L_{12}) \cdot F_X(x_2)
\end{aligned}$$

Άρα το $I_{122} = [L_{12} + (U_{12} - L_{12}) \cdot F_X(x_1),\ \ L_{12} + (U_{12} - L_{12}) \cdot F_X(x_2))$.

Ανάλογα το $I_{123} = [L_{12} + (U_{12} - L_{12}) \cdot F_X(x_2),\ \ L_{12} + (U_{12} - L_{12}) \cdot F_X(x_3))$ και το i−-οστό θα ακολουθεί την εξίσωση:

$$I_{12i} = [L_{12} + (U_{12} - L_{12}) \cdot F_X(x_{i-1}),\ \ L_{12} + (U_{12} - L_{12}) \cdot F_X(x_i))$$

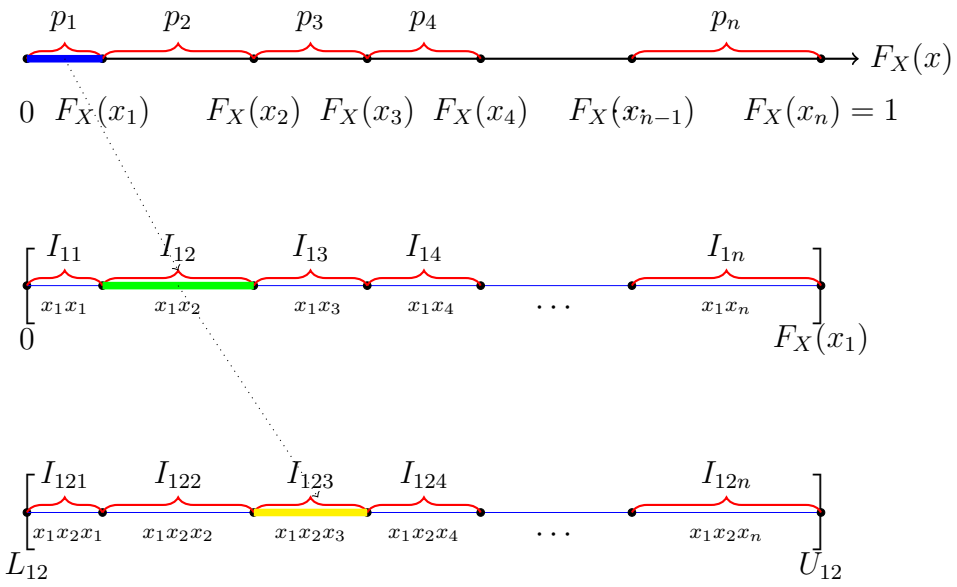
Σημείωση: Όπως και στην προηγούμενη διαμέριση έτσι και σε αυτή η σχέση που παράγαμε για το i−οστό υποδιάστημα δουλεύει ακόμα και όταν το πρώτο και δεύτερο σύμβολο είναι οι τυχαίως επιλαχούσες τιμές $x_j,\ x_kx$. Τότε το διάστημα στο οποίο ανήκει η ακολουθία $x_jx_kx_i$ δίνεται από τη σχέση:

$$I_{jki} = [L_{jk} + (U_{jk} - L_{jk}) \cdot F_X(x_{i-1}),\ \ L_{jk} + (U_{jk} - L_{jk}) \cdot F_X(x_i))$$

,

όπου $L_{jk},\ U_{jk}$ το κάτω και άνω άκρο αντίστοιχα του διαστήματος I_{jk}.

Στο επόμενο σχήμα απεικονίζεται γραφικά η τρίτη κατά σειρά διαμέριση που πραγματοποιούμε:



Σχήμα 5.20: Για να κωδικοποιηθεί η συμβολοσειρά $x_1x_2x_3$ επιλέγεται πρώτα το I_{12} (πράσινο χρώμα) το οποίο διαμερίζεται σε n κομμάτια με μήκη ανάλογα των $Pr[X = x_i]$. Ύστερα επιλέγουμε το τρίτο διάστημα I_{123} (κίτρινο χρώμα) με μήκος $p_1p_2p_3$

Η διαδικασία θα συνεχιστεί μέχρι να φτάσουμε στο διάστημα που κωδικοποιεί την ακολουθία $x_1x_2, \cdots x_n$. Η εξίσωση που ορίζει το διάστημα των πραγματικών αριθμών στον οποίο ανήκει η ω δίνεται από τη σχέση

$$I_{x_1z_2\cdots x_n} = [L_{12\cdot n-1} + (U_{12\cdot n-1} - L_{12\cdot n-1}) \cdot F_X(x_{n-1}),\quad L_{12\cdot n-1} + (U_{12\cdot n-1} - L_{12\cdot n-1}) \cdot F_X(x_n))$$

Τέλος για να κωδικοποιήσουμε την ακολουθία $x_1x_2 \cdots x_n$ θα επιλέξουμε το μέσο του διαστήματος $I_{12\cdots n}$ και να βρούμε το δυαδικό ανάπτυγμά του μήκους $\lceil log\frac{1}{\prod_{i=1}^{n} Pr[X = x_i]}\rceil + 1$[13]

Συνοψίζοντας, έστω μία τυχαία μεταβλητή X που παίρνει τιμές στο σύνολο $\mathcal{X} = \{x_1, x_2, \cdots x_n\}$ και ακολουθεί την $P_X(x) = \{Pr[X = x_1], \cdots Pr[X = x_n]\}$. Για να κωδικοποιήσουμε μία ακολουθία ανεξάρτητων και ισόνομων τυχαίων μεταβλητών της μορφής $X_iX_{i+1} \cdot X_{i+l}$ ακολουθούμε τα παρακάτω βήματα:

Διαδικασία κωδικοποίησης

1. Αρχικά χωρίζουμε το τμήμα των πραγματικών αριθμών [0,1] στα διαστήματα $I_j = [F_X(x_{j-1}), F_X(x_j))$ μήκους $p_j = Pr[X = x_j]$ και επιλέγουμε το διάστημα I_i στο οποίο ανήκει το x_i.

2. Για k από το 0 μέχρι το l:

   (α′) Υπολογίζουμε το διάστημα $I_{j\cdots j+k} = [L_{j\cdots j+k-1} + (U_{j\cdots j+k-1} - L_{j\cdots j+l-1}) \cdot F_X(x_{j-1}),\quad L_{j\cdots j+k-1} + (U_{j\cdots j+k-1} - L_{j\cdots j+k-1}) \cdot F_X(x_j))$

3. Επιλέγουμε το μέσο του διαστήματος $I_{j(j+1)\cdots j+l}$ και το αναπαριστούμε σε μία δυαδική συμβολοσειρά μήκους $\lceil log\frac{1}{\prod_{i=1}^{n} Pr[X = x_j]}\rceil + 1$. Η συμβολοσειρά αυτή αποτελεί την ετικέτα της ακολουθίας $x_jx_{j+1} \cdots x_{j+l}$.

Παράδειγμα 5.8. *Έστω μία τυχαία μεταβλητή X που παίρνει τις τιμές* $\{x_1,\ x_2,\ x_3\}$ *και ακολουθεί την κατανομή* $P_X(x) = \{\frac{1}{2},\ \frac{1}{4},\ \frac{1}{4}\}$. *Να βρείτε την ετικέτα της ακολουθίας* $x_2x_1x_1x_3$

Λύση

[13] Έχουμε το δικαίωμα να βάλουμε το γινόμενο των πιθανοτήτων στον παρονομαστή του κλάσματος της λογαριθμικής συνάρτησης γιατί έχουμε υποθέσει ότι οι πιθανότητες είναι ισόνομες και ανεξάρτητες.

1. *Βρίσκουμε τα* $F_X(x_1) = \frac{1}{2},\ F_X(x_2) = Pr[X = x_1] + Pr[X = x_2] = \frac{3}{4},\ F_X(x_2) = Pr[X = x_1] + Pr[X = x_2] + Pr[X = x_3] = 1$. *Έπειτα χωρίζουμε το διάστημα* $I_0 = [0, 1]$ *στα υποδιαστήματα* $I_1 = [F_X(0),\ F_X(x_1)) = [0,\ \frac{1}{2}),\ I_2 = [F_X(x_1),\ F_X(x_2)) = [\frac{1}{2},\ \frac{3}{4}),\ I_3 = [F_X(x_2),\ F_X(x_3)) = [\frac{3}{4}, 1)$.

2. *Επειδή το* x_2 *ανήκει στο* $I_2 = [\frac{1}{2},\ \frac{3}{4})$, *το επιλέγουμε και το διαμερίζουμε εκ νέου σε τρία υποδιαστήματα* $I_{21},\ I_{22},\ I_{23}$. *Θέτοντας* $L_2 = \frac{1}{2},\ U_2 = \frac{3}{4}$, *έχουμε:*

$$I_{21} = [L_2 + (U_2 - L_2) \cdot F_X(0),\ \ L_2 + (U_2 - L_2) \cdot F_X(x_1)) = [\frac{1}{2} + \frac{1}{4} \cdot 0,\ \ \frac{1}{2} + \frac{1}{4} \cdot \frac{1}{2}) = [\frac{1}{2},\ \ \frac{5}{8})$$
$$I_{22} = [L_2 + (U_2 - L_2) \cdot F_X(x_1),\ \ L_2 + (U_2 - L_2) \cdot F_X(x_2)) = [\frac{1}{2} + \frac{1}{4} \cdot \frac{1}{2},\ \ \frac{1}{2} + \frac{1}{4} \cdot \frac{3}{4}) = [\frac{5}{8},\ \ \frac{11}{16})$$
$$I_{23} = [L_2 + (U_2 - L_2) \cdot F_X(x_2),\ \ L_2 + (U_2 - L_2) \cdot F_X(x_3)) = [\frac{1}{2} + \frac{1}{4} \cdot \frac{3}{4},\ \ \frac{1}{2} + \frac{1}{4} \cdot 1) = [\frac{11}{16},\ \ \frac{3}{4})$$

3. *Επειδή* $x_2x_1 \in I_{21}$, *διαμερίζουμε το* I_{21} *εκ νέου θέτοντας* $L_{21} = \frac{1}{2}$ *και* $U_{12} = \frac{5}{8}$

$$I_{211} = [L_{21} + (U_{21} - L_{21}) \cdot F_X(0),\ \ L_{21} + (U_{21} - L_{21}) \cdot F_X(x_1)) = [\frac{1}{2} + \frac{1}{8} \cdot 0,\ \ \frac{1}{2} + \frac{1}{8} \cdot \frac{1}{2}) = [\frac{1}{2},\ \ \frac{9}{16})$$
$$I_{212} = [L_{21} + (U_{21} - L_{21}) \cdot F_X(x_1),\ \ L_{21} + (U_{21} - L_{21}) \cdot F_X(x_2)) = [\frac{1}{2} + \frac{1}{8} \cdot \frac{1}{2},\ \ \frac{1}{2} + \frac{1}{8} \cdot \frac{3}{4}) = [\frac{9}{16},\ \ \frac{19}{32})$$
$$I_{213} = [L_{21} + (U_{21} - L_{21}) \cdot F_X(x_2),\ \ L_{21} + (U_{21} - L_{21}) \cdot F_X(x_3)) = [\frac{1}{2} + \frac{1}{8} \cdot \frac{3}{4},\ \ \frac{1}{2} + \frac{1}{8} \cdot 1) = [\frac{19}{32},\ \ \frac{5}{8})$$

4. *Επειδή* $x_2x_1x_1 \in I_{211}$, *διαμερίζουμε το* I_{211} *εκ νέου θέτοντας* $L_{211} = \frac{1}{2}$ *και* $U_{12} = \frac{9}{16}$

$$I_{2111} = [L_{211} + (U_{211} - L_{211}) \cdot F_X(0),\ \ L_{211} + (U_{211} - L_{211}) \cdot F_X(x_1)) = [\frac{1}{2} + \frac{1}{16} \cdot 0,\ \ \frac{1}{2} + \frac{1}{16} \cdot \frac{1}{2}) = [\frac{1}{2},\ \ \frac{9}{16})$$
$$I_{2112} = [L_{211} + (U_{211} - L_{211}) \cdot F_X(x_1),\ \ L_{211} + (U_{211} - L_{211}) \cdot F_X(x_2)) = [\frac{1}{2} + \frac{1}{16} \cdot \frac{1}{2},\ \ \frac{1}{2} + \frac{1}{16} \cdot \frac{3}{4}) = [\frac{17}{32},\ \ \frac{35}{64})$$
$$I_{2113} = [L_{211} + (U_{211} - L_{211}) \cdot F_X(x_2),\ \ L_{211} + (U_{211} - L_{211}) \cdot F_X(x_3)) = [\frac{1}{2} + \frac{1}{16} \cdot \frac{3}{4},\ \ \frac{1}{2} + \frac{1}{16} \cdot 1) = [\frac{35}{64},\ \ \frac{9}{16})$$

5.

6. *Επειδή* $x_2x_1x_1x_3 \in I_{2113}$, *με* $L_{2113} = \frac{35}{64}$ *και* $U_{2113} = \frac{9}{16}$, *επιλέγουμε το μέσο του διαστήματος και το κωδικοποιούμε δυαδικά. Η δυαδική αναπαράσταση του μέσου αποτελεί την ετικέτα της συμβολοσειράς και θα έχει μήκος:*

$$l = \lceil log_2 \frac{1}{Pr[X = x_2] \cdot Pr[X = x_1] \cdot Pr[X = x_1] \cdot Pr[X = x_3]} \rceil + 1 = \lceil log_2 \frac{1}{\frac{1}{4} \cdot \frac{1}{2} \cdot \frac{1}{2} \cdot \frac{1}{4}} \rceil + 1 =$$

$$\lceil log_2 64 \rceil + 1 = \lceil 6 \rceil + 1 = 7$$

$$r = L_{2113} + \frac{U_{2113} - L_(2113)}{2} = \frac{35}{64} + \frac{\frac{9}{16} - \frac{35}{64}}{2} = \frac{35}{64} + \frac{1}{128} = \frac{71}{128}$$

$$tag = \ (r)_2 = \left(\frac{71}{127}\right)_2 = 1 \cdot \frac{1}{2^1} + 0 \cdot \frac{1}{2^2} + 0 \cdot \frac{1}{2^3} + 0 \cdot \frac{1}{2^4} + 1 \cdot \frac{1}{2^5} + 1 \cdot \frac{1}{2^6} + 1 \cdot \frac{1}{2^7} = (1000111)_2$$

Παρατήρηση: Στο παράδειγμα σε κάθε βήμα υπολογίζαμε όλα τα νέα διαστήματα στα οποία διαμεριζόταν το διάστημα της προηγούμενης επανάληψης. Αυτό έγινε για λόγους εξάσκησης και εξοικείωσης με τη μέθοδο και όχι γιατί ήταν πραγματικά απαραίτητο για την επίλυση του προβλήματος. Συγκεκριμένα αν μελετήσουμε ξανά το παράδειγμα παρατηρούμε ότι από την στιγμή που ξέρουμε πως το πρώτο σύμβολο είναι το x_2 μπορούμε απευθείας να υπολογίσουμε το $I_2 = [F_X(x_1), \ \ F_X(x_2))$. Αφού βρήκαμε το I_2 αρά και τα $L_2, \ U_2$ μπορούμε άμεσα να υπολογίσουμε το I_{21} στο οποίο ανήκει η υπακολουθία x_2x_1 με την χρήση της σχέσης $I_{21} = [L_2 + (U_2 - L_2)F_X(0), \ \ L_2 + (U_2 - L_2)F_X(x_1))$. Μετά τον υπολογισμό του I_{21} γνωρίζουμε το άνω και κάτω άκρο $U_{21}, \ L_{21}$ αντίστοιχα και μπορούμε απευθείας να υπολογίσουμε το I_{211}. Τέλος μετά την εύρεση του I_{211} και άρα των $L_{211}, \ U_{211}$ υπολογίζουμε το I_{2113}. Μέσα από αυτή την παρατήρηση αναδεικνύεται και η υπερτέρηση της αριθμητικής κωδικοποίησης έναντι του Huffman. Για να βρούμε την κωδική λέξη που αντιστοιχεί σε μία συμβολοσειρά $x_ix_{i+1} \cdots x_{i+l}$ με την αριθμητική κωδικοποίηση χρειάζεται να εκτελέσουμε $O(l)$ πράξεις, ενώ με το Huffman για να "χτίσουμε" το δένδρο που περιέχει την ακολουθία $x_ix_{i+1} \cdots x_{i+l}$ θα έπρεπε πρώτα να βρούμε όλες τις δυνατές κωδικές λέξεις μήκους l και έπειτα να βρούμε τον κώδικα για την ακολουθία που μας αφορά. Αυτή η διαδικασία στην χειρότερη περίπτωση, δηλαδή αν μιλάμε για πηγή χωρίς περιορισμούς, σημαίνει ότι θα χρειαστούν $O(l!)$ πράξεις!!!.

Για να αποκωδικοποιήσουμε μία ακολουθία που κωδικοποιήθηκε με την διαδικασία της αριθμητικής κωδικοποίησης θα προσπαθήσουμε να μιμηθούμε τα βήματα που κάνει ο κωδικοποιητής. Αρχικά μετατρέπουμε την ακολουθία κωδικών λέξεων που θα μας δοθεί σε ένα πραγματικό αριθμό r που να ανήκει στο [0,1]. Έπειτα διαμερίζουμε το $[0, 1]$ σε n διαστήματα της μορφής $[F_X(x_{i-1}), \ \ F_X(x_i))$ και ελέγχουμε σε ποιο υποδιάστημα ανήκει ο πραγματικό αριθμός r. Στην συνέχεια αφού επιλέξουμε το κατάλληλο διάστημα I_i που αντιστοιχεί στην τιμή x_i, σημειώνουμε το πρώτο αποκωδικοποιημένο σύμβολο x_i.Το I_i διαμερίζεται εκ νέου ελέγχοντας σε ποιο καινούργιο υποδιάστημα ανήκει ο αριθμός r. Αφού βρούμε το κατάλληλο διάστημα $I_{ij} : \ r \ \in I_{ij}$, σημειώνουμε το δεύτερο αποκωδικοποιημένο σύμβολο x_j. Η διαδικασία συνεχίζεται μέχρι να αποκωδικοποιηθεί όλη η συμβολοσειρά.

Ένα εύλογο ερώτημα είναι πότε σταματάει η διαδικασία καθώς οι διαμερίσεις μπορούν να συνεχιστούν επ΄ άπειρον αφού βρισκόμαστε στο σύνολο των πραγματικών αριθμών. Πρώτον αυτό το πρόβλημα πρακτικά δεν πρέπει να μας ανησυχεί αφού οι υπολογιστές έχουν πεπερασμένη ακρίβεια οπότε οι διαμερίσεις δεν μπορούν να συνεχιστούν επ΄άπειρον. Ακόμη και αν φοβόμαστε ότι οι διαμερίσεις θα συνεχιστούν μετά την αποκωδικοποίηση της αρχικής ακολουθίας, προσθέτοντας σύμβολα που δεν ανήκουν σε αυτή μπορούμε να επιλύσουμε αυτή την προβληματική κατάσταση προσθέτοντας στο τέλος κάθε ακολουθίας ένα σύμβολο που να σηματοδοτεί το τέλος της διαδικασίας.

Ένα ακόμη σημείο που μπορούμε να βελτιώσουμε στη διαδικασία της κωδικοποίησης είναι οι έλεγχοι που πραγματοποιούνται σε κάθε βήμα για να αποφανθούμε σε ποιο διάστημα ανήκει ο αριθμός r. Παρατηρούμε ότι κάθε διάστημα έχει τη μορφή

$$I_{ij\cdots kl} = [L_{ij\cdots k} + (U_{ij\cdots k} - L_{ij\cdots k}) \cdots F_X(x_{l-1}), \ \ L_{ij\cdots k} + (U_{ij\cdots k} - L_{ij\cdots k}) \cdots F_X(x_l))$$

Προκειμένου να γίνει πιο απλή η σύγκριση μπορούμε να αφαιρέσουμε από τα άκρα του διαστήματος το κάτω άκρο $L_{ij\cdots k}$. Η πράξη αυτή ισοδυναμεί με μία μετάθεση του διαστήματος προς τα αριστερά κατά $L_{ij\cdots k}$. Για να είναι ορθή η σύγκριση θα πρέπει να μεταθέσουμε και τον αριθμό r, άρα έχουμε:

$$I'_{ij\cdots kl} = [(U_{ij\cdots k} - L_{ij\cdots k}) \cdots F_X(x_{l-1}), \quad (U_{ij\cdots k} - L_{ij\cdots k}) \cdots F_X(x_l))$$
$$r' = r - L_{ij\cdots k}$$

Τέλος μπορούμε να διαιρέσουμε το διάστημα $I'_{ij\cdots kl}$ με το μήκος $U_{ij\cdots k} - L_{ij\cdots k}$. Αυτό αποτελεί μία μεγέθυνση του διαστήματος καθώς $U_{ij\cdots k} - L_{ij\cdots k} < 1$. Μεγενθύνουμε αντίστοιχα και την r' και έχουμε:

$$I^*_{ij\cdots kl} = [F_X(x_{l-1}), \quad F_X(x_l))$$
$$r^* = \frac{r'}{(U_{ij\cdots k} - L_{ij\cdots k})}$$

Διαδικασία αποκωδικοποίησης

1. Μετατρέπουμε την ακολουθία των κωδικών συμβόλων στον πραγματικό αριθμό r
2. Χωρίζουμε το $I = [0, 1]$ στα διαστήματα $I_i = [F_X(x_{i-1}), \quad F_X(x_i)]$
3. Θέτουμε με $L^{(0)} = 0$ και $U^{(0)} = 1$.
4. Για j από 0 μέχρι το $n - 1$
   (α′) Υπολογίζουμε το $r^* = \dfrac{r - L^{(j)}}{U^{(j)} - L^{(j)}}$
   (β′) Ελέγχουμε σε ποιο διάστημα $[F_X(x_{i-1}), \quad F_X(x_i)]$ ανήκει το r^*
   (γ′) Σημειώνουμε το x_i στο αποκωδικοποιημένο αρχείο
   (δ′) Βρίσκουμε τα άκρα $L^{(j+1)} = L^{(j)} + (U^{(j)} - L^{(j)}) \cdot F(x_j)$
   $U^{(j+1)} = L^{(j)} + (U^{(j)} - L^{(j)}) \cdot F(x_{j+1})$

Παράδειγμα 5.9. *Να αποκωδικοποιήσετε την ετικέτα του προηγούμενου παραδείγματος.*
Λύση

1. *Αρχικά μετατρέπουμε την δυαδική αναπαράσταση του r, δηλαδή την ετικέτα σε δεκαδική:*

$$(1000111)_{10} = \left(\frac{71}{127}\right)_2 = 1 \cdot \frac{1}{2^1} + 0 \cdot \frac{1}{2^2} + 0 \cdot \frac{1}{2^3} + 0 \cdot \frac{1}{2^4} + 1 \cdot \frac{1}{2^5} + 1 \cdot \frac{1}{2^6} + 1 \cdot \frac{1}{2^7} = \frac{71}{128}$$

2. *Έπειτα χωρίζουμε το [0,1] στα διαστήματα* $[F_X(x_{i-1}, \quad F_X(x_i)]$ *και έχουμε:*

$$I_1 = [F_X(0), \quad F_X(x_1)] = [0, \quad \frac{1}{2})$$
$$I_2 = [F_X(x_1), \quad F_X(x_2)] = [\frac{1}{2}, \quad \frac{3}{4})$$
$$I_3 = [F_X(x_2), \quad F_X(x_3)] = [\frac{3}{4}, \quad 1)$$

3. *Για* $j = 0$ *υπολογίζουμε*

(α′) $r^* = \dfrac{r - L^{(0)}}{U^{(0)} - L^{(0)}} = \dfrac{r - 0}{1} = r$

(β′) $r^* \in [F_X(x_1), \quad F_X(x_2)]$

(γ′) Γράφουμε το x_2 στο αποκωδικοποιημένο αρχείο

(δ′) Υπολογίζουμε το $L^{(1)} = L^{(0)} + U^{(0)} - L^{(0)} \cdot F_X(x_1) = \frac{1}{2}$ *και* $U^{(1)} = L^{(0)} + U^{(0)} - L^{(0)} \cdot F_X(x_2) = \frac{3}{4}$

4. *Για* $j = 1$ *υπολογίζουμε*

(α′) $$r^* = \frac{r - L^{(1)}}{U^{(1)} - L^{(1)}} = \frac{r - \frac{1}{2}}{\frac{1}{4}} = \frac{\frac{71}{128} - \frac{1}{2}}{\frac{1}{4}} = \frac{28}{128}$$

(β′) $r^* \in [F_X(0), \ F_X(x_1)]$

(γ′) Γράφουμε το x_1 στο αποκωδικοποιημένο αρχείο

(δ′) Υπολογίζουμε το $L^{(2)} = L^{(1)} + U^{(1)} - L^{(1)} \cdot F_X(0) = \frac{1}{2}$ *και* $U^{(2)} = L^{(1)} + U^{(1)} - L^{(1)} \cdot F_X(x_1) = \frac{5}{8}$

5. *Για* $j = 2$ *υπολογίζουμε*

(α′) $$r^* = \frac{r - L^{(2)}}{U^{(2)} - L^{(2)}} = \frac{r - \frac{1}{2}}{\frac{1}{8}} = \frac{\frac{71}{128} - \frac{1}{2}}{\frac{1}{8}} = \frac{56}{128}$$

(β′) $r^* \in [F_X(0), \ F_X(x_1)]$

(γ′) Γράφουμε το x_1 στο αποκωδικοποιημένο αρχείο

(δ′) Υπολογίζουμε το $L^{(3)} = L^{(2)} + U^{(2)} - L^{(2)} \cdot F_X(0) = \frac{1}{2}$ *και* $U^{(3)} = L^{(2)} + U^{(2)} - L^{(2)} \cdot F_X(x_1) = \frac{9}{16}$

6. *Για* $j = 3$ *υπολογίζουμε*

(α′) $$r^* = \frac{r - L^{(3)}}{U^{(3)} - L^{(3)}} = \frac{r - \frac{1}{2}}{\frac{1}{16}} = \frac{\frac{71}{128} - \frac{1}{2}}{\frac{1}{8}} = \frac{112}{128}$$

(β′) $r^* \in [F_X(x_2), \ F_X(x_3)]$

(γ′) Γράφουμε το x_3 στο αποκωδικοποιημένο αρχείο

(δ′) Υπολογίζουμε το $L^{(4)} = L^{(3)} + U^{(3)} - L^{(3)} \cdot F_X(x_2) = \frac{35}{64}$ *και* $U^{(4)} = L^{(3)} + U^{(3)} - L^{(3)} \cdot F_X(x_3) = \frac{9}{16}$

Γιατί η αριθμητική κωδικοποίηση παράγει μοναδικά αποκωδικοποιήσιμους κώδικες.

Όταν ολοκληρώνεται η κωδικοποίηση μιας ακολουθίας τιμών $x_i x_j \cdots x_l$ της τυχαίας μεταβλητής X, ξέρουμε ότι η διαδικασία έχει τερματίσει στο διάστημα $I_{ij\ldots k}$. Για να βρούμε την ετικέτα που αντιστοιχεί στην ακολουθία επιλέγουμε το μέσο του διαστήματος και το αναπαριστούμε δυαδικά. Αυτό που πρέπει να παρατηρήσουμε είναι ότι σε κάθε βήμα της κωδικοποίησης δημιουργούνται διαστήματα που είναι ξένα μεταξύ τους και η υπακολουθία που έχουμε επεξεργαστεί μέχρι εκείνη την στιγμή ανήκει σε κάποιο από αυτά. Επίσης κάθε υποδιάστημα στο οποίο περιορίζεται η υπακολουθία σε κάθε βήμα του αλγορίθμου περιέχεται στο $[F_X(x_{i-1}), \ F_X(x_i)]$ που αντιστοιχεί στο πρώτο στοιχείο της ακολουθίας.

Για παράδειγμα έστω ότι έχουμε να κωδικοποιήσουμε τις ακολουθίες $a_1 = x_1 x_2 \cdots x_k$ και $a_2 = x_2 x_1 \cdots x_l$. Η κωδικοποίηση της a_1 ξεκινάει από το διάστημα $[F_X(0). \ F_X(x_1))$ ενώ της a_2 από το $[F_X(x_1) \ \ F_X(x_2))$. Οποιοδήποτε διάστημα παραχθεί από αυτό το σημείο και μετά θα εμπεριέχεται εξόλοκλήρου στα αρχικά διαστήματα. Άρα αναγκαστικά τα διαστήματα στα οποία ανήκει η ετικέτα r_1 της a_1 και r_2 της a_2 θα είναι και αυτά ξένα μεταξύ τους. Αφού $r_1 \neq r_2$ έπεται ότι και $(r_1)_2 \neq (r_2)_2$ καθώς ξέρουμε ότι η δυαδική αναπαράσταση ενός πραγματικού αριθμού είναι μοναδική.

Ακόμα και αν δύο ακολουθίες ξεκινήσουν από την ίδια τιμή x_i και διαφοροποιηθούν μετά από j σύμβολα, πάλι τα διαστήματα που τις περιέχουν θα είναι ξένα μεταξύ τους γιατί από το σύμβολο $j+1$ και μετά η διαδικασία θα συνεχίσει σε ξένα μεταξύ τους υποδιαστήματα..

Ο αριθμός που επιλέγουμε για να αναπαραστήσουμε τη ακολουθία μετά τη δυαδική αναπαράσταση του στρογγυλοποιείται στα $\lceil log\frac{1}{\prod Pr[X=x_i]}\rceil+1$ πρώτα ψηφία. Πρέπει λοιπόν να εξετάσουμε ότι η συγκεκριμένη αποκοπή δεν επηρεάζει τη μοναδική αντιστοιχία μεταξύ ακολουθιών και πραγματικών αριθμών που περιγράψαμε παραπάνω. Το πρόβλημα αυτό είναι ισοδύναμο με τον να εξετάσουμε αν ο αποκομμένος αριθμός παύει να ανήκει στο διάστημα που κατέληξε η διαδικασία κωδικοποίησης. Υπενθυμίζουμε ότι ως κατάλληλο αριθμό επιλέγουμε το μέσο του τελικού διαστήματος.

Έστω μία ακολουθία $a_i = x_i(1)x_i(2)\cdots x_i(n)$ που μετά την κωδικοποίηση της κατέληξε στο διάστημα I_a με άνω και κάτω άκρα τα $U_a,\ L_a$ αντίστοιχα και r το μέσο του διαστήματος που χρησιμοποιούμε για να αναπαραστήσουμε την ακολουθία. Προφανώς r θα ανήκει στο διάστημα $[U_a, L_a) = [F_X(\mathbf{x_1^n}-1),\ F_X(\mathbf{x_1^n}))$ Από τη στιγμή που ο αποκομμένος αριθμός έχει μικρότερη ακρίβεια από τον πραγματικό θα ισχύει:

$$\lfloor r\rfloor_{l(\mathbf{x_1^n})} \leq r$$

Συγκεκριμένα από το σφάλμα αποκοπής σε $l(\mathbf{x_1^n})$ ψηφία ξέρουμε ότι

$$r - \lfloor r\rfloor_{l(\mathbf{x_1^n})} \leq \frac{1}{2^{l(\mathbf{x_1^n})}}$$

Επειδή όμως το r επιλέχθηκε κατά τέτοιον τρόπο ώστε $r < U_a < F_X(\mathbf{x_1^n})$ έπεται ότι και $\lfloor r\rfloor_{l(\mathbf{x_1^n})} < U_a < F_X(\mathbf{x_1^n})$.

Αρκεί λοιπόν να δείξουμε ότι $\lfloor r\rfloor_{l(\mathbf{x_1^n})} > L_a$.

$$\frac{1}{2^{l(\mathbf{x_1^n})}} = \frac{1}{2^{\lceil log\frac{1}{Pr[\mathbf{X}=\mathbf{a}]}\rceil+1}} \leq \frac{1}{2^{log\frac{1}{Pr[\mathbf{X}=\mathbf{a}]}+1}} = \frac{1}{2\cdot 2^{log\frac{1}{Pr[\mathbf{X}=\mathbf{a}]}}} = \frac{1}{2\cdot\frac{1}{Pr[\mathbf{X}=\mathbf{a}]}} = \frac{Pr[\mathbf{X}=\mathbf{a}]}{2}$$[14]

Επειδή όμως το r είναι το μέσο του διαστήματος $[U_a,\ L_a)$ έχουμε ότι:

$$\frac{Pr[\mathbf{X}=\mathbf{a}]}{2} = r - L_a \leq \lfloor r\rfloor_{l(\mathbf{x_1^n})} - L_a \Rightarrow \lfloor r\rfloor_{l(\mathbf{x_1^n})} \geq L_a + \frac{Pr[\mathbf{X}=\mathbf{a}]}{2} > L_a$$

Ολοκληρώνοντας έπεται ότι το $\boxed{\lfloor r\rfloor_{l(\mathbf{x_1^n})} \in [L_a,\ U_a) \subset [F_X(\mathbf{x_1^n}-1),\ F_X(\mathbf{x_1^n}))}$

*Μέχρι στιγμής έχουμε δείξει ότι ο αποκομμένος αριθμός r, άρα και η δυαδική αναπαράσταση του εμπεριέχονται εξόλοκλήρου στο διάστημα που τερματίζει η κωδικοποίηση. Οπότε η αναπαράσταση της ακολουθίας δια μέσου του αριθμού $\lfloor r\rfloor_{l(\mathbf{x_1^n})}$ είναι μοναδική και ο κώδικας μη ιδιόμορφος. Μένει να δείξουμε ότι ο κώδικας είναι μοναδικά αποκωδικοποιήσιμος. Για να το επιτύχουμε αυτό θα δείξουμε ότι ο κώδικας που σχηματίζεται από τις αποκομμένες δυαδικές ακολουθίες είναι στιγμιαίος. Από την στιγμή που αποκόπτουμε τους αριθμούς r_i, των ακολουθιών $a_i = x_i(1)\cdots x_i(n)$ στα $l(\mathbf{x_1^n})$ πρώτα ψηφία καταλαβαίνουμε ότι οι αριθμοί που ανήκουν στο διάστημα $[\lfloor r\rfloor_{l(\mathbf{x_1^n})}, \lfloor r\rfloor_{l(\mathbf{x_1^n})} + \frac{1}{2^{l(\mathbf{x_1^n})}})$ όταν αναπαρασταθούν με δυαδικές ακολουθίες μήκους $n > l(\mathbf{x_1^n})$ θα ταυτίζονται στα πρώτα $l(\mathbf{x_1^n})$ ψηφία. Για το λόγο αυτό πρέπει τα διαστήματα $[\lfloor r\rfloor_{l(\mathbf{x_1^n})}, \lfloor r\rfloor_{l(\mathbf{x_1^n})} + \frac{1}{2^{l(\mathbf{x_1^n})}})$ που αντιστοιχούν σε διαφορετικές ακολουθίες $\mathbf{a_i}$ να είναι ξένα μεταξύ τους. Η απαίτηση αυτή είναι ισοδύναμη με το να αποδείξουμε ότι τα διαστήματα $[\lfloor r\rfloor_{l(\mathbf{x_1^n})}, \lfloor r\rfloor_{l(\mathbf{x_1^n})} + \frac{1}{2^{l(\mathbf{x_1^n})}})$ εμπεριέχονται εξολοκλήρου στα διαστήματα $[F_X(\mathbf{x_1^n}-1),\ F_X(\mathbf{x_1^n}))$ καθώς για κάθε $\mathbf{x_1^n} \neq \mathbf{x_1^{n\prime}} \Rightarrow [F_X(\mathbf{x_1^n}-1),\ F_X(\mathbf{x_1^n})) \cap [F_X(\mathbf{x_1^{n\prime}}-1),\ F_X(\mathbf{x_1^{n\prime}})) = \emptyset$

[14] Με τα έντονο $X = a$ δηλώνουμε ότι πρόκειται για ακολουθία τυχαίων μεταβλητών

Ήδη έχουμε δείξει ότι $\lfloor r_i \rfloor_{l(\mathbf{x_1^n})} \geq F_X(\mathbf{x_1^n} - 1)$. *Αρκεί να δείξουμε ότι* $F_X(\mathbf{x_1^n}) - \lfloor r_i \rfloor_{l(\mathbf{x_1^n})} > \frac{1}{2^{l(\mathbf{x_1^n})}}$
Επειδή $\lfloor r_i \rfloor_{l(\mathbf{x_1^n})} < r \Rightarrow -\lfloor r_i \rfloor_{l(\mathbf{x_1^n})} > -r \Rightarrow F_X(\mathbf{x_1^n}) - \lfloor r_i \rfloor_{l(\mathbf{x_1^n})} < F_X(\mathbf{x_1^n}) - r$, *άρα*

$$F_X(\mathbf{x_1^n}) - r = \frac{Pr[\mathbf{X_1^n} = \mathbf{x_1^n}]}{2} > \frac{1}{2^{l(\mathbf{x_1^n})}} \Rightarrow 2^{l(\mathbf{x_1^n})} > \frac{2}{Pr[\mathbf{X_1^n} = \mathbf{x_1^n}]} \Rightarrow 2^{l(\mathbf{x_1^n})-1} > \frac{1}{Pr[\mathbf{X_1^n} = \mathbf{x_1^n}]} \Rightarrow$$
$$l(\mathbf{x_1^n}) - 1 > \log \frac{1}{Pr[\mathbf{X_1^n} = \mathbf{x_1^n}]} \Rightarrow l(\mathbf{x_1^n}) > \log \frac{1}{Pr[\mathbf{X_1^n} = \mathbf{x_1^n}]} + 1$$

Άρα για $l(\mathbf{x_1^n}) = \lceil log \frac{1}{Pr[\mathbf{X} = \mathbf{a_i}]} \rceil + 1$ *ο κώδικας είναι στιγμιαίος καθώς όλα τα προθέματα θα ανήκουν στο* $[L_a,\ U_a)$

Από την θεωρία $H(X_1, \cdots, X_n) \leq \sum_{\mathbf{x_1^n} \in \mathcal{X}^n} Pr[\mathbf{X_1^n} = \mathbf{x_1^n}] l(\mathbf{x_1^n})]$. *Επειδή:* $l(\mathbf{x_1^n}) = \lceil \log \frac{1}{Pr[\mathbf{X_1^n} = \mathbf{x_1^n}]} \rceil + 1 \Rightarrow l(x_1, \cdots, x_n) < \log \frac{1}{Pr[\mathbf{X_1^n} = \mathbf{x_1^n}]} + 2 \Rightarrow Pr[\mathbf{X_1^n} = \mathbf{x_1^n}] \cdot l(\mathbf{x_1^n}) < Pr[\mathbf{X_1^n} = \mathbf{x_1^n}] \cdot \log \frac{1}{Pr[\mathbf{X_1^n} = \mathbf{x_1^n}]} + 2Pr[\mathbf{X_1^n} = \mathbf{x_1^n}] \Rightarrow \overline{L}(\mathbf{x_1^n}) < \sum_{\mathbf{x_1^n} \in \mathcal{X}^n} Pr[\mathbf{X_1^n} = \mathbf{x_1^n}] \cdot \log \frac{1}{Pr[\mathbf{X_1^n} = \mathbf{x_1^n}]} + 2Pr[\mathbf{X_1^n} = \mathbf{x_1^n}] \Rightarrow$
$\overline{L}(\mathbf{x_1^n}) < H(X_1, \cdots, X_n) + 2 \Rightarrow$

$$H(X_1, \cdots, X_n) \leq \overline{L}(\mathbf{x_1^n}) < H(X_1, \cdots, X_n) + 2 \overset{i.i.d}{\Rightarrow} H(X) \leq \frac{1}{n} \cdot \overline{L}(\mathbf{x_1^n}) < H(X) + \frac{2}{n}$$

5.6 Η κωδικοποίηση LZ77

5.6.1 Παρουσίαση τς μεθόδου

Η κωδικοποίηση LZ77 αναπτύχθηκε από τους J. Jiv και A. Lempel το 1977, εξ΄ ου και το όνομα, στη δημοσίευση A universal algorithm for sequential data compression[**ziv1977universal**]. Η κωδικοποίηση LZ77 έφερε επανάσταση στον τρόπο που οι επιστήμονες σχεδίαζαν τις τεχνικές συμπίεσης. Είναι εύκολο να παρατηρήσουμε πως ότι τεχνική έχουμε παρουσιάσει μέχρι σήμερα βασίστηκε στην ύπαρξη μία σ.μ.π πιθανοτήτων και στον τρόπο που εκείνη επιβάλει την βέλτιστη κωδικοποίηση. Η διαφορά που έκαναν οι Lempel και Ziv είναι ότι δεν απαίτησαν την ύπαρξη κανενός πιθανοκρατικού μοντέλου για την πηγή. Η μέθοδος τους βασίστηκε στην λογική που λέει πως εάν μία πηγή έχει μια στοιχειώδη δομή τότε σε ένα κείμενο που θα παραχθεί από κείνη θα υπάρχουν πολλές εκφράσεις που θα επαναλαμβάνονται σε τακτά χρονικά διαστήματα. Εάν καταφέρουμε να κωδικοποιηθούν αυτές οι εκφράσεις με έναν έξυπνο τρόπο ώστε να γλυτώσουμε περιττά bits τότε θα επιτύχουμε συμπίεση. Ο "έξυπνο τρόπος" που σκαρφίστηκαν οι J. Jiv και A. Lempel βασίζεται στην ιδέα της αντιγραφής. Παρατηρήστε την παρακάτω έκφραση:

to be or not to be

Η φράση to be βλέπουμε ότι επαναλαμβάνεται δύο φορές. Αν πούμε ότι η έκφραση αποτελείται από 18 σύμβολα συμπεριλαμβανομένων και των κενών, τότε αντί να επαναλάβουμε τη φράση to be μπορούμε να την αντικαταστήσουμε με το ζεύγος (1,5) όπου το 1 συμβολίζει την αρχή της έκφρασης στο κείμενο και το 5, το μήκος του επαναλαμβανόμενου τμήματος του κειμένου. Έτσι έχουμε:

$$\underbrace{t}_{1}\underbrace{o}_{2}\underbrace{\ }_{3}\underbrace{b}_{4}\underbrace{e}_{5}\underbrace{\ }_{6}\underbrace{o}_{7}\underbrace{r}_{8}\underbrace{\ }_{9}\underbrace{n}_{10}\underbrace{o}_{11}\underbrace{t}_{12}\underbrace{\ }_{13}\underbrace{t}_{14}\underbrace{o}_{15}\underbrace{\ }_{16}\underbrace{b}_{17}\underbrace{e}_{18}$$
$$\underbrace{t}_{1}\underbrace{o}_{2}\underbrace{\ }_{3}\underbrace{b}_{4}\underbrace{e}_{5}\underbrace{\ }_{6}\underbrace{o}_{7}\underbrace{r}_{8}\underbrace{\ }_{9}\underbrace{n}_{10}\underbrace{o}_{11}\underbrace{t}_{12}\underbrace{\ }_{13}(1,5)$$

Γιατί όμως επιτύχαμε συμπίεση με αυτόν το τρόπο; Σκεφτείτε το παραπάνω κομμάτι κειμένου να ήταν γραμμένο σε κωδικοποίηση ANSI όπου κάθε σύμβολο αναπαριστάται με 8 bits, τότε θα αποτελούταν από

$8 \cdot 18 = 144$bits. Όμως ο αριθμός 1 χρειάζεται 1bit και το 5 3bits για αν αναπαρασταθούν άρα η καινούρια έκφραση θα αποτελείται από $8 \cdot 13 + 4 = 104 + 4 = 108$bits. Προφανώς σε ένα μεγάλο κείμενο οι εκφράσεις που επαναλαμβάνονται θα είναι μακρύτερες και η συμπίεση θα έχει μεγαλύτερο αποτέλεσμα από ότι σε αυτό το παράδειγμα. Η κεντρική όμως ιδέα της αναπαράστασης επαναλαμβανόμενων εκφράσεων από ζεύγη αριθμών που καταδεικνύουν τη θέση και το μήκος της συμβολοσειράς που επαναλαμβάνεται αποτελεί την βάση πάνω στην οποία οι παραπάνω ερευνητές δημιούργησαν την lZ77 κωδικοποίηση. Μπορεί το παράδειγμα να δόθηκε με κείμενο για να γίνει αντιληπτή η τεχνική, αλλά ο LZ77 χρησιμοποιείται για να βρίσκει μέγιστα ταιριάσματα σε ακολουθίες από bits. Θα παρουσιάσουμε την διαδικασία κωδικοποίησης και αποκωδικοποίησης όπως περιγράφτηκε στο πρωτότυπο κείμενο[15].

• Έστω ότι έχουμε ένα αλφάβητο $\mathcal{A}$ και μία συμβολοσειρά $S = s_1 s_2 \cdots s_{l(S)}$ με σύμβολα που ανήκουν στο $\mathcal{A}$ που έχει μήκος $l(s)$. Σκοπός μας είναι να επεξεργαστούμε το S με τέτοιο τρόπο ώστε να βρούμε τις επαναλαμβανόμενες συμβολοσειρές που υπάρχουν και να τις κωδικοποιήσουμε με ένα πιο σύντομο τρόπο. Για το λόγο αυτό θα πρέπει τις υποσυμβολοσειρές που συναντάμε για πρώτη φορά, καθώς επεξεργαζόμαστε το S, να τις αποθηκεύουμε σε μία δομή ώστε να τις θυμηθούμε όταν βρεθούμε σε μία επανάληψη τους. Για παράδειγμα στην έκφραση "to be or not to be" μέχρι να ξανασυναντήσουμε την υποσυμβολοσειρά "to be", η οποία είναι η μέγιστη υποσυμβολοσειρά που επαναλαμβάνεται, έχουμε συναντήσει για πρώτη φορά τις υποσυμβολοσειρές {t,o,to, to, o, b, tob, ob, e, tobe, $\cdots$, }. Βλέπουμε λοιπόν ότι η συγκεκριμένη πρόταση χωρίζεται με φυσικό τρόπο σε δύο μέρη:

$$\underbrace{\text{to be or not}}_{\text{Γνωστό μέρος}} \mid \underbrace{\text{to be}}_{\text{επαναλαμβανόμενο μέρος}}$$

Αυτή την λογική θα προσπαθήσουμε να ακολουθήσουμε και σε συμβολοσειρές μεγαλύτερου μήκους. Επειδή όμως σαν S μπορεί να θεωρηθεί ακόμη και ένα ολόκληρο κείμενο γίνεται κατανοητό ότι μία τέτοιου μεγέθους συμβολοσειρά δεν μπορούμε να την επεξεργαστούμε μονομιάς. Προκειμένου λοιπόν το S να το επεξεργαστούμε τμηματικά θα επιστρατεύσουμε τη δομή της ενδιάμεσης μνήμης (buffer). Την συγκεκριμένη δομή που θα φιλοξενεί κάποιο μέρος της S θα την χωρίσουμε σε δύο κομμάτια, την **ενδιάμεση μνήμη αναζήτησης** (search buffer) στην οποία θα αποθηκεύεται το κομμάτι που θεωρούμε ήδη επεξεργασμένο άρα και τις υποσυμβολοσειρές του γνωστές και την **ενδιάμεση μνήμη πρόβλεψης** (look-ahead buffer) που περιέχει το ακατέργαστο μέρος του συγκεκριμένου κομματιού του S στο οποίο αναζητούμε επαναλαμβανόμενες συμβολοσειρές. Για το παράδειγμα με τη φράση ""to be or not to be" η ενδιάμεση μνήμη θα είχε τη μορφή:

$$\underbrace{\underbrace{\text{to be or not}}_{\text{ενδιάμεση μνήμη αναζήτησης}} \mid \underbrace{\text{to be}}_{\text{ενδιάμεση μνήμη πρόβλεψης}}}_{\text{ενδιάμεση μνήμη}}$$

Πριν αρχίσουμε την παρουσίαση της διαδικασίας κωδικοποίησης θα αποσαφηνίσουμε το συμβολισμό που θα χρησιμοποιηθεί.

▷ Το αλφάβητο της συμβολοσειράς συμβολίζεται με το καλλιγραφικό $\mathcal{A}$. Χάριν δικής μας ευκολίας μπορούμε να σκεφτόμαστε ότι το αλφάβητο $\mathcal{A}$ είναι το {0,1}.

▷ Μια συμβολοσειρά είναι μία ακολουθία που έχει ως σύμβολα στοιχεία του αλφαβήτου $\mathcal{A}$, $S = s_1 s_2 \cdots s_{l(S)}$ με $s_i \in \mathcal{A}$ $\forall i = 1, 2, \cdots l(S)$

▷ Το μήκος της συμβολοσειράς S ορίζεται ως το πλήθος των συμβόλων της και συμβολίζεται με $l(S)$

▷ Η υποσυμβολοσειρά της S που αρχίζει από τον δείκτη i και τελειώνει στον δείκτη j με $1 \leq i \leq j \leq l(S)$ συμβολίζεται με $S(i, j)$.

▷ Κάθε υποσυμβολοσειρά της μορφής $S(1, j)$ με $j < l(S)$ ονομάζεται **τυπικό πρόθεμα** της S.

Σκοπός της κωδικοποίησης $LZ77$ είναι να βρει το μεγαλύτερο ταίριασμα μεταξύ των ακολουθιών συμβόλων (συμβολοσειρών) που περιέχονται στην ενδιάμεση μνήμη αναζήτησης και στην ενδιάμεση μνήμη πρόβλεψης.

[15] **ziv1977universal**.

Δηλαδή θέλουμε να βρούμε ένα δείκτη i που να ανήκει στην ενδιάμεση μνήμη αναζήτησης, η οποία ας πούμε ότι έχει μέγεθος j, τέτοιο ώστε:

$$S(i, i+l-1) = S(j+1, j+l), \text{δηλαδή σχηματικά:}$$

$$\underbrace{\underbrace{s_1 s_2 \cdots \overbrace{s_i s_{i+1} \cdots s_{i+l-1}}^{\text{μέγιστο ταίριασμα}} s_{i+l} \cdots s_j}_{\text{ενδιάμεση μνήμη αναζήτησης}} | \underbrace{\overbrace{s_{j+1} \cdots s_{j+l}}^{=s_i s_{i+1} \cdots s_{i+l-1}} s_{j+l+1} \cdots s_{l(S)}}_{\text{ενδιάμεση μνήμη πρόβλεψης}}}_{\text{ενδιάμεση μνήμη}} \tag{5.9}$$

,

με l το μεγαλύτερο μήκος για το οποίο ισχύει αυτή η ισότητα.

▷ Ως δείκτη του μέγιστου ταιριάσματος ορίζεται ο ακέραιος p που δηλώνει τη θέση του i που πετυχαίνει το μέγιστο ταίριασμα:

$$p = max_{1 \le i \le j}\{i : l(S(i, i+l-1)_{)} = l\}, \text{ όπου } l \text{ το μήκος του μέγιστου ταιριάσματος}$$

▷ Το μέγιστο ταίριασμα στην πρωτότυπη δημοσίευση αναφέρετε ως **αναπαραγωγίσιμη επέκταση** (reproducible extension) του προθέματος $S(1, j)$ στην συμβολοσειρά S. Εμείς βέβαια για λόγους απλότητας θα το λέμε μέγιστο ταίριασμα. Αφού λοιπόν ξεκαθαρίσαμε τους συμβολισμούς και τις διάφορες έννοιες που συσχετίζονται με τον LZ77 ήρθε η ώρα να περιγράψουμε τη διαδικασία της κωδικοποίησης.

Κωδικοποίηση LZ77

• Η κωδικοποίηση αρχίζει με μία άδεια ενδιάμεση μνήμη μεγέθους n. Η μνήμη χωρίζεται σε δύο μέρη, την ενδιάμεση μνήμη αναζήτησης που έχει μέγεθος $n - L_s$ και την ενδιάμεση μνήμη πρόβλεψης με μέγεθος L_s. Το L_s αποτελεί το μεγαλύτερό ακατέργαστο κομμάτι το οποίο μπορούμε να επεξεργαστούμε. Γενικά στις πρακτικές εφαρμογές ισχύει ότι $n - L_s >> L_s$.

$$\underbrace{[\quad n - L_s \quad |}_{\text{ενδιάμεση μνήμη αναζήτησης}} \underbrace{\quad L_s \quad]}_{\text{ενδιάμεση μνήμη πρόβλεψης}} \to B_0$$

• Η μνήμη γεμίζει με $n - L_s$ μηδενικά και ακολουθούν τα L_s πρώτα στοιχεία της συμβολοσειράς S.

$$[\underbrace{000 \cdots\cdots 0}_{n-L_s} | s_1 s_2 \cdots\cdots s_{L_s}] \to B_1 \tag{5.10}$$

Αναζητούμε αν υπάρχει κάποιο i με $1 \le i \le L_s - 1$ ώστε η υποσυμβολοσειρά $S(1, i)$ να αποτελεί το μέγιστο ταίριασμα για το κομμάτι $B_1 = \underbrace{000 \cdots 0}_{n-L_s} S(1, L_s) = 0^{n-L_s} S(1, L_s)$.

✓ Αν υπάρχει μέγιστο ταίριασμα S_1 έστω μήκους $l_1 = i + 1$ τότε κωδικοποιούμε το $S_1 = S(1, i+1)$ στην κωδική λέξη C_1 και μετακινούμε την ενδιάμεση μνήμη κατά l_1 σύμβολα. Έτσι η ενδιάμεση μνήμη τώρα περιέχει το κομμάτι $B_2 = B_1(l_1 + 1, n)S(L_s + 1, L_s + l_1)$. Εδώ υπάρχει ένα λεπτό σημείο που χρήζει προσοχής· Ενώ το μέγιστο ταίριασμα αναφέρεται στην υποσυμβολοσειρά $S(1, i)$, το κομμάτι των συμβόλων που κωδικοποιούμε αποτελείται από το μέγιστο ταίριασμα $S(1, i)$ και το επόμενο σύμβολο s_{i+1}. Για αυτό τον λόγο λέμε ότι το $S_1 = S(1, i+1)$ και ότι το $l_1 = i + 1$.

$$[\underbrace{000 \cdots 0}_{n-L_s-l_1} \underbrace{s_1 s_2 \cdots s_{l_1}}_{S_1} | s_{l_1+1} \cdots\cdots s_{L_s+l_1}] \to B_2$$

. ✓ Αν δεν υπάρχει μέγιστο ταίριασμα τότε απλά μετακινούμε την ενδιάμεση μνήμη κατά ένα σύμβολο το οποίο παραμένει ακωδικοποίητο.

$$[\underbrace{000\cdots0}_{n-L_s-1}s_1|s_2s_3\cdots\cdots s_{L_s+1}]\rightarrow B_2$$

Επειδή και στις δύο περιπτώσεις, είτε υπάρχει ταίριασμα είτε όχι η ενδιάμεση μνήμη κυλίεται κατά 1 ή l_i σύμβολα ο $LZ77$ αλλά και οι παραλλαγές του που έχουν αυτό το χαρακτηριστικό ονομάζεται και **τεχνική κυλιόμενου παραθύρου** (sliding window technique).

• Χωρίς βλάβη της γενικότητας υποθέτουμε ότι στο πρώτο βήμα υπάρχει μέγιστο ταίριασμα μεγέθους l_1 και η ενδιάμεση μνήμη περιέχει το κομματι B_2. Στο δεύτερο βήμα ψάχνουμε για ένα ακόμη μέγιστο ταίριασμα. Έστω ότι υπάρχει και έχει μέγεθος l_2. Τότε το παράθυρο θα κυλιστεί κατά l_2 σύμβολα και πλέον θα έχει μία από τις παρακάτω μορφές

$$(1)\ [\underbrace{000\cdots0}_{n-L_s-l_1-l_2}\underbrace{s_1s_2\cdots s_{l_1}}_{S_1}\underbrace{s_{l_1+1}s_{l_1+2}\cdots s_{l_1+l_2}}_{S_2}|s_{l_1+l_2+1}\cdots\cdots s_{L_s+l_1+l_2}]\rightarrow B_3,$$

αν $n-L_s-l_1-l_2>0,$ δηλαδή αν περισσεύουν $n-L_s-l_1-l_2$ μηδενικά από το προηγούμενο βήμα μέσα στην ενδιάμεση μνήμη αναζήτησης

ή

$$(2)\ [s_is_{i+1}\cdots s_{l_1}\underbrace{s_{l_1+1}s_{l_1+2}\cdots s_{l_1+l_2}}_{S_2}|s_{l_1+l_2+1}\cdots\cdots s_{L_s+l_1+l_2}]\rightarrow B_3$$

αν μετά τη κύλιση κατά l_2 σύμβολα η ενδιάμεση μνήμη αναζήτησης αρχίζει από το i-οστό σύμβολο

Προσοχή! Η ενδιάμεση μνήμη αναζήτησης που αντιστοιχεί στο κομμάτι B_3 δεν μπορεί να αρχίζει από το s_{l_1+1} και μετά γιατί τότε θα ίσχυε ότι το l_2 είναι μεγαλύτερο από το μέγεθος της ενδιάμεσης μνήμης αναζήτησης, το οποίο είναι άτοπο καθώς υποθέσαμε ότι τα ταιριάσματα έχουν μέγιστο μήκος L_s-1 και το $L_s<<n-L_s$.

• Γενικά στο $i-$οστό βήμα η ενδιάμεση μνήμη περιέχει το κομμάτι $B_i=B_{i-1}(l_{i-1}+1,n)S(\sum_{k=1}^{i-1}l_k+1,L_s+\sum_{k=1}^{i-1}l_k)$. Έστω ότι το μέγιστο ταίριασμα στο παρών βήμα έχει μήκος l_i, τότε το μέγιστο ταίριασμα αντιστοιχεί στην συμβολοσειρά $S_i=B_i(n-L_s+1,n-L_s+l_i)$. Στη συνέχει κωδικοποιούμε το S_i στη κωδική λέξη C_i ως εξής: $C_i=C_{i1}C_{i2}C_{i3}$, όπου C_{i1} είναι η αναπαράσταση του p_i-1 , η οποία γίνεται με βάση τον πληθάριθμο του αλφαβήτου μας. Από την στιγμή που ο δείκτης p_i κυμαίνεται από το ακέραιο 1 μέχρι τον ακέραιο $n-L_s$ για να τον αναπαραστήσουμε χρειαζόμαστε $\lceil log_{|\mathcal{A}|}(n-L_s)\rceil$ σύμβολα. Το C_{i2} εκφράζει την αναπαράσταση του l_i-1. Επειδή το $1\leq l_i\leq L_s-1$ επαρκούν $\lceil log_{|\mathcal{A}|}(L_s)\rceil$ για την κωδικοποίηση του. Τέλος C_{i3} είναι το τελευταίο σύμβολο του S_i και το συνολικό μήκος του $C_i=\lceil log_{|\mathcal{A}|}(n-L_s)\rceil+\lceil log_{|\mathcal{A}|}(L_s)\rceil+1$.

Άρα συνοπτικά κάθε κομμάτι S_i που αποτελείται από το μέγιστο ταίριασμα και το επόμενο σύμβολο κωδικοποιείται με μία τριπλέτα τη μορφής (θέση, μήκος, τελευταίο σύμβολο) ή αλλιώς (p_i-1,l_i-1,s_{l_i}). Αφού γίνει η κωδικοποιήσει η ενδιάμεση μνήμη θα ολισθήσει κατά l_i θέσεις για να ξεκινήσει το $i+1$-βήμα της διαδικασίας της κωδικοποίησης. Πλέον μέσα στην ενδιάμεση μνήμη υπάρχει το κομμάτι $B_{i+1}=B_i(l_i+1,n)S(\sum_{k=1}^{i}l_k+1,L_s+\sum_{k=1}^{i}l_k)$. Όταν τελειώσει η διαδικασία της κωδικοποίησης το S θα έχει αναλυθεί στα κομμάτια $000\cdots0S_1S_2\cdots S_N$ και κάθε κομμάτι θα έχει κωδικοποιηθεί στην αντίστοιχη λέξη C_i σταθερού μήκους.

Παράδειγμα 5.10. *Δίνεται η δυαδική συμβολοσειρά "11101001101111111011111"μήκους 21 που αναπαριστά τη λέξη too με βάση την κωδικοποίηση ASCII. Να κωδικοποιηθεί η συμβολοσειρά σύμφωνα με τον LZ77 για* $L_s=10$ *και* $n-L_s=20$

Λύση

Στο αρχικό βήμα γεμίζουμε την ενδιάμεση μνήμη αναζήτησης $n-L_s$ *με 20 μηδενικά και τη μνήμη πρόβλεψης με τα πρώτα 10 ψηφία της δοσμένης δυαδικής συμβολοσειράς (1110100110*11111101111*).*

$$[00000000000000000000|1110100110] \qquad (1)$$

Για να βρούμε το μέγιστο ταίριασμα συγκρίνουμε το τελευταίο σύμβολο της μνήμης αναζήτησης και το πρώτο σύμβολο της μνήμης πρόβλεψης. Βλέπουμε ότι τα δύο σύμβολα δεν ταυτίζονται. Συνεχίζουμε την αναζήτηση για ταιριάσματα μετακινώντας τον κόκκινο δείκτη μία θέση αριστερά. Βλέπουμε πως όσο πίσω και αν πάμε στην μνήμη αναζήτησης δεν θα βρούμε κάποιο ταίριασμα οπότε μετακινούμε την ενδιάμεση μνήμη κατά 1 σύμβολο και ο LZ77 παράγει την τριπλέτα (0,0,1).

$$[000000000000000000\underbrace{1}_{S_0}|\overset{\downarrow}{1}101001101] \tag{2}$$

Πάλι συγκρίνουμε τα δύο γειτονικά σύμβολα και βλέπουμε ότι υπάρχει ταίριασμα. Για να εξετάσουμε αν συνεχίζεται το ταίριασμα μετακινούμε και τους δύο δείκτες (κόκκινο και μπλε) μία θέση δεξιά

$$[000000000000000000\underbrace{1}_{S_0}|\overset{\downarrow}{1}\overset{\downarrow}{1}01001101] \tag{3}$$

Συγκρίνουμε τα δύο σύμβολα που έχουν τους δείκτες και βλέπουμε ότι πάλι συμπίπτουν. Αν οι δείκτες μετακινηθούν μία ακόμη θέση προς τα δεξιά, τα σύμβολα παύουν να ταυτίζονται. Άρα μέχρι στιγμής το ταίριασμα που έχουμε βρει εκφράζεται από την τριπλέτα (19,2,0). Για να εξακριβώσουμε αν υπάρχει άλλο ταίριασμα με μεγαλύτερο μήκος επιστρέφουμε τον μπλε δείκτη στην αρχική του θέση (2)και τον κόκκινο μία θέση αριστερότερα από την αρχική του

$$[00000000000000000\overset{\downarrow}{0}\underbrace{1}_{S_0}|\overset{\downarrow}{1}101001101] \tag{4}$$

Συγκρίνουμε τα σύμβολα και βλέπουμε ότι δεν ταυτίζονται, Συνεχίζουμε να μετακινούμε τον κόκκινο δείκτη αριστερά μέχρι να βρούμε είτε μία θέση όπου τα σύμβολα που αναφέρονται στον κόκκινο και μπλε δείκτη να ταυτιστούν είτε μέχρι να φτάσουμε στην αρχή της ενδιάμεσης μνήμης αναζήτησης χωρίς να βρούμε κάποιο επιπλέον ταίριασμα. Στο παρών βήμα επιβεβαιώνεται η δεύτερη περίπτωση άρα το μέγιστό ταίριασμα είναι η συμβολοσειρά $S_1 = 110$ που αντιστοιχεί στην τριπλέτα (19,2,0). Μετά τη κωδικοποίηση της τριπλέτας μετακινούμε το παράθυρο κατά 3 θέσεις και έχουμε:

$$[0000000000000000\underbrace{1}_{S_0}\underbrace{11\overset{\downarrow}{0}}_{S_1}|\overset{\downarrow}{1}001101111] \tag{5}$$

Πλέον μέσα στην ενδιάμεση μνήμη βρίσκεται το σημειωμένο με μπλε κομμάτι της συμβολοσειράς (1110100110111111101111). Ξεκινάμε πάλι την σύγκριση μεταξύ των σημειωμένων συμβόλων. Επειδή το $(0 \neq 1)$, ο κόκκινος δείκτης μετακινείται μία θέση αριστερότερα. Οι επόμενες εικόνες δείχνουν τη διαδικασία της κωδικοποίησης στο βήμα αυτό.

$$[0000000000000000\underbrace{1}_{S_0}\underbrace{11\overset{\downarrow}{0}}_{S_1}|\overset{\downarrow}{1}001101111] \tag{6}$$

$$[0000000000000000\underbrace{1}_{S_0}\underbrace{11\overset{\downarrow}{0}}_{S_1}|\overset{\downarrow}{1}001101111] \tag{7}$$

$$[0000000000000000\underbrace{1}_{S_0}\underbrace{110}_{S_1}|\overset{\downarrow}{1}\overset{\downarrow}{0}01101111] \tag{8}$$

Άρα το μέγιστο ταίριασμα είναι το $S_2 = 100$. Αν παρατηρήσουμε θα δούμε ότι δεν υπάρχει μεγαλύτερο ταίριασμα από αυτό οπότε κωδικοποιούμε την τριπλέτα (18,2,0) και ολισθαίνουμε το παράθυρο κατά 3 θέσεις.

$$[0000000000000\underbrace{1}_{S_0}\underbrace{110}_{S_1}\underbrace{10\overset{\downarrow}{0}}_{S_2}|\overset{\downarrow}{1}101111110] \quad (9)$$

$$[0000000000000\underbrace{1}_{S_0}\underbrace{110}_{S_1}\underbrace{\overset{\downarrow}{1}00}_{S_2}|\overset{\downarrow}{1}101111110] \quad (10)$$

$$[0000000000000\underbrace{1}_{S_0}\underbrace{110}_{S_1}\underbrace{1\overset{\downarrow}{0}0}_{S_2}|\overset{\downarrow}{1}101111110] \quad (11)$$

Άρα το πρώτο υποψήφιο ταίριασμα είναι το $S_3 = 11$ που αντιστοιχεί στην τριπλέτα (17,1,1). Ο αλγόριθμός συνεχίζει το ψάξιμο για να βρει και άλλα ταιριάσματα.

$$[0000000000000\underbrace{1}_{S_0}\underbrace{1\overset{\downarrow}{1}0}_{S_1}\underbrace{100}_{S_2}|\overset{\downarrow}{1}101111110] \quad (12)$$

$$[0000000000000\underbrace{1}_{S_0}\underbrace{11\overset{\downarrow}{0}}_{S_1}\underbrace{100}_{S_2}|\overset{\downarrow}{1}101111110] \quad (13)$$

Ο αλγόριθμός βρίσκει ακόμα ένα ταίριασμα που είναι πανομοιότυπο με το προηγούμενο και για αυτό κρατάει το πιο πρόσφατο που αντιστοιχεί στην τριπλέτα (15,1,1). Συνεχίζοντας την αναζήτηση βρίσκουμε ακόμη ένα ταίριασμα:

$$[0000000000000\underbrace{1}_{S_0}\underbrace{\overset{\downarrow}{1}10}_{S_1}\underbrace{100}_{S_2}|\overset{\downarrow}{1}101111110] \quad (14)$$

$$[0000000000000\underbrace{1}_{S_0}\underbrace{1\overset{\downarrow}{1}0}_{S_1}\underbrace{100}_{S_2}|\overset{\downarrow}{1}101111110] \quad (15)$$

$$[0000000000000\underbrace{1}_{S_0}\underbrace{11\overset{\downarrow}{0}}_{S_1}\underbrace{100}_{S_2}|\overset{\downarrow}{1}101111110] \quad (16)$$

$$[0000000000000\underbrace{1}_{S_0}\underbrace{110}_{S_1}\underbrace{\overset{\downarrow}{1}00}_{S_2}|1\overset{\downarrow}{1}01111110] \quad (17)$$

$$[0000000000000\underbrace{1}_{S_0}\underbrace{110}_{S_1}\underbrace{10\overset{\downarrow}{0}}_{S_2}|11\overset{\downarrow}{0}1111110] \quad (18)$$

Το τρίτο κατά σειρά ταίριασμα που βρήκε ο αλγόριθμός είναι το $S_3 = 11011$ που αντιστοιχεί στην τριπλέτα (14,4,1) και επειδή έχει το μεγαλύτερο μήκος μέχρι στιγμής ο LZ77 θυμάται αυτό. Το τελευταίο ταίριασμα που θα βρει και φαίνεται στα επόμενα τέσσερα βήματα θα το απορρίψει γιατί δεν έχει μεγαλύτερο μέγεθος από το ταίριασμα που βρήκαμε σε αυτό το βήμα.

$$[000000000000\underbrace{\overset{\downarrow}{1}}_{S_0}\underbrace{110}_{S_1}\underbrace{100}_{S_2}|\overset{\downarrow}{1}101111110] \quad (19)$$

$$[0000000000000\underbrace{\overset{\downarrow}{1}}_{S_0}\underbrace{110}_{S_1}\underbrace{100}_{S_2}|\overset{\downarrow}{1}101111110] \quad (20)$$

$$[0000000000000\underbrace{1}_{S_0}\underbrace{\overset{\downarrow}{1}10}_{S_1}\underbrace{100}_{S_2}|\overset{\downarrow}{1}101111110] \quad (21)$$

$$[0000000000000\underbrace{1}_{S_0}\underbrace{110}_{S_1}\underbrace{100}_{S_2}|1101111110] \tag{22}$$

Άρα το τελευταίο ταίριασμα είναι το $S_3 = 110$ *που αντιστοιχεί στην τριπλέτα (13,2,0) και απορρίπτεται καθώς δεν είναι το μέγιστο. Κυλίουμε το παράθυρο κατά σύμβολα και έχουμε:*

$$[00000000\underbrace{1}_{S_0}\underbrace{110}_{S_1}\underbrace{100}_{S_2}\underbrace{11011}_{S_3}|111101111] \tag{23}$$

Ακολουθώντας την διαδικασία που περιγράψαμε αναλυτικά βρίσκουμε ότι το μέγιστο ταίριασμα αντιστοιχεί στο $S_4 = 11110$ *με τριπλέτα (18,4,0)*

$$[00000000\underbrace{1}_{S_0}\underbrace{110}_{S_1}\underbrace{100}_{S_2}\underbrace{11011}_{S_3}|111111101111] \tag{24}$$

Ολισθαίνουμε το παράθυρο κατά 4 θέσεις και κωδικοποιούμε το τελευταίο κομμάτι που απέμεινε μέσα στην ενδιάμεση μνήμη πρόβλεψης.

$$[000\underbrace{1}_{S_0}\underbrace{110}_{S_1}\underbrace{100}_{S_2}\underbrace{11011}_{S_3}\underbrace{11110}_{S_4}|1111] \tag{25}$$

$$\underbrace{1}_{S_0}[\underbrace{110}_{S_1}\underbrace{100}_{S_2}\underbrace{11011}_{S_3}\underbrace{11110}_{S_4}\underbrace{1111}_{S_5}|\text{␣}] \tag{25}$$

Η τριπλέτα που αντιστοιχεί στο $S_5 = 1111$ *είναι η (3,3,1). Πολλές πρακτικές υλοποιήσεις του LZ77 συνηθίζουν μέχρι να κωδικοποιηθεί και το τελευταίο κομμάτι του* S*, να συμπληρώνουν την ενδιάμεση μνήμη πρόβλεψης με μηδενικά (padding) έτσι ώστε να έχει πάντα μέγεθος* L_s*. Επειδή η τακτική αυτή έχει αμελητέο αντίκτυπο στην κωδικοποίηση της συμβολοσειράς* S *αποφασίσαμε να την παραλείψουμε ώστε το παράδειγμα να είναι περισσότερο απλό και κατανοητό. Το ίδιο ισχύει και για τα μηδενικά που μπαίνουν στην αρχή της κωδικοποίησης. Ο μόνος λόγος που τα διατηρήσαμε ήταν για να είμαστε τυπικοί ως προς την πρωτότυπη δημοσίευση.*

Ολοκληρώνοντας το παράδειγμα βλέπουμε ότι η αρχική συμβολοσειρά "11101001101111110111" αναλύθηκε στην $S_0S_1S_2S_3S_4S_5$ *και κωδικοποιήθηκε στην* $C_0C_1C_2C_3C_4C_5$ *σύμφωνα με τον κανόνα* $C_i = (p_i - 1)_2(l_i - 1)_2(s)_2$

Συμβολοσειρά S_i	*Τριπλέτα*	*Κωδική λέξη* C_i
$S_0 = 1$	*(0,0,1)*	*00000 0000 1*
$S_1 = 110$	*(19,2,0)*	*10011 0010 0*
$S_2 = 100$	*(18,2,0)*	*10010 0010 0*
$S_3 = 11011$	*(14,4,1)*	*01110 0100 1*
$S_4 = 11110$	*(18,4,0)*	*10010 0100 0*
$S_5 = 1111$	*(3,3,1)*	*00011 0011 1*

Προφανώς σε αυτό το παράδειγμα δεν πετύχαμε συμπίεση. Έτσι κι αλλιώς δεν ήταν αυτός ο σκοπός αλλά η κατανόηση της διαδικασίας κωδικοποίησης του LZ77. Γενικά για να πετύχουμε συμπίεση με τον συγκεκριμένο αλγόριθμο πρέπει η συμβολοσειρά[16] να είναι πολύ μεγαλύτερη από την ενδιάμεση μνήμη. Στον DEFLATE που χρησιμοποιείται μία παραλλαγή του LZ77 μαζί με τον στατικό Huffman χρησιμοποιούμε κάτι χιλιάδες bytes για την ενδιάμεση μνήμη αναζήτησης ενώ μόνο μερικά δεκάδες bytes για την μνήμη πρόβλεψης και τα αρχεία που συμπιέζονται είναι κάποια Mbytes σε μέγεθος.

[16]Όταν λέμε συμβολοσειρά, χρησιμοποιούμε την έννοια με έναν πιο γενικό τρόπο. Συμβολοσειρά θεωρούμε και ένα αρχείο κειμένου ή ένα αρχείο μουσικής ή οποιαδήποτε συλλογή δεδομένων επιθυμούμε να συμπιέσουμε.

Αποκωδικοποίηση του LZ77

Η αποκωδικοποίηση ακολουθεί την ακριβώς αντίστροφη διαδικασία από την κωδικοποίηση. Η λογική όμως παραμένει ίδια, δηλαδή πάλι θα έχουμε μία ενδιάμεση μνήμη στην οποία αυτή τη φορά θα είναι αποθηκευμένα τα αποκωδικοποιημένα σύμβολα και με βάση αυτά θα μεταφράζουμε και τα υπόλοιπά. Η ενδιάμεση μνήμη θα έχει μέγεθος $n - L_s$

• Αρχικά γεμίζουμε την ενδιάμεση μνήμη αποκωδικοποίησης (decoding buffer) με μηδενικά.

$$\underbrace{[0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0 \cdots\cdots\cdots\ 0]}_{\text{Αρχικοποίηση της μνήμης αποκωδικοποίησης}}\ c_1^1 c_2^1 \cdots c_{\lceil log(n-L_S)\rceil}^1 c_{\lceil log(n-L_S)\rceil+1}^1 \cdots c_{\lceil log(n-L_S)\rceil+\lceil log(L_s)\rceil}^1 c_{\lceil log(n-L_S)\rceil+\lceil log(L_s)\rceil+1}^1$$ [17]

• Στο πρώτο βήμα διαβάζουμε τα πρώτα $\lceil log(n - L_S)\rceil$ σύμβολα ($c_1^1 c_2^1 \cdots c_{\lceil log(n-L_S)\rceil}^1$) που συναντάμε μετά τη μνήμη τα οποία αναπαριστούν το δείκτη $p_1 - 1$ που δείχνει σε ποιο σημείο της μνήμης πρέπει να μεταβούμε ώστε να αρχίσει η διαδικασία τη αποκωδικοποίησης.

$$[0\ 0\ 0\ 0\ 0 \cdots \overset{p_i-1}{0} \cdots 0] c_1^1 c_2^1 \cdots c_{\lceil log(n-L_S)\rceil}^1 c_{\lceil log(n-L_S)\rceil+1}^1 \cdots c_{\lceil log(n-L_S)\rceil+\lceil log(L_s)\rceil}^1 c_{\lceil log(n-L_S)\rceil+\lceil log(L_s)\rceil+1}^1$$

Στην συνέχεια διαβάζοντας τα επόμενα $\lceil log(L_S)\rceil$ σύμβολα $c_{\lceil log(n-L_S)\rceil+1} \cdots c_{\lceil log(n-L_S)\rceil+\lceil log(L_s)\rceil}$ αποφαινόμαστε για το μήκος της αντιγραφής $(l_1 - 1)$ και αρχίζουμε να γράφουμε μετά την μνήμη τα σύμβολα της μνήμης από το σημείο $p_1 - 1$ μέχρι το $p_1 - 1 + (l_1 - 1)$. Κάθε φορά που γράφουμε ένα αποκωδικοποιημένο σύμβολο μετακινούμε τη μνήμη μία θέση δεξιότερα.

$$[0\ 0 \cdots \overset{p_i-1}{0} \cdots \overset{p_i-1+(l_1-i)}{0} \cdots 0 \cdots \underset{1}{0}\ \underset{2}{0}\ \underset{3}{0} \cdots \underset{l_1-1}{0}] c_{\lceil log(n-L_S)\rceil+\lceil log(L_s)\rceil+1}^1$$

Τέλος μετακινούμε τη μνήμη μια ακόμη θέση δεξιά ώστε να συμπεριλάβει το $c_{\lceil log(n-L_S)\rceil+\lceil log(L_s)\rceil+1}^1$ που αναπαριστά το επόμενο σύμβολο, δηλαδή το τρίτο στοιχείο της τριπλέτας, που αποθηκεύαμε κατά τη κωδικοποίηση.

$$[0\ 0\ 0\ 0\ 0 \cdots \overset{p_i-1}{0} \cdots \overset{p_i-1+(l_1-i)}{0} \cdots 0 \cdots \underset{1}{0}\ \underset{2}{0}\ \underset{3}{0} \cdots \underset{l_1-1}{0}\ \underset{l_1}{c_{\lceil log(n-L_S)\rceil+\lceil log(L_s)\rceil+1}^1}]$$

$$[\underbrace{0\ 0 \cdots}_{n-L_s-l_1} \underbrace{0\ 0\ 0 \cdots c_{\lceil log(n-L_S)\rceil+\lceil log(L_s)\rceil+1}^1}_{l_1}] c_1^2 c_2^2 \cdots c_{\lceil log(n-L_S)\rceil}^2 c_{\lceil log(n-L_S)\rceil+1}^2 \cdots c_{\lceil log(n-L_S)\rceil+\lceil log(L_s)\rceil}^2 c_{\lceil log(n-L_S)\rceil+\lceil log(L_s)\rceil+1}^2$$

•Γενικά στο $i-$οστό βήμα θα υπάρχουν μέσα στην μνήμη τα στοιχεία που θα έχουν προκύψει από την αποκωδικοποίηση του $C_{i-1} \to S_{i-1}$:

$$[d_1\ d_2 \cdots d_{N-L_s}] c_1^i c_2^i \cdots c_{\lceil log(n-L_S)\rceil}^i c_{\lceil log(n-L_S)\rceil+1}^i \cdots c_{\lceil log(n-L_S)\rceil+\lceil log(L_s)\rceil}^i c_{\lceil log(n-L_S)\rceil+\lceil log(L_s)\rceil+1}^i$$

• Η αποκωδικοποίηση κατά το $i-$οστό βήμα ακολουθεί τη λογική που αναφέραμε και στο πρώτο. Δηλαδή διαβάζοντας τα πρώτα $\lceil log(n - L_S)\rceil$ σύμβολα που βρίσκονται με την μνήμη ανακτάμε τη θέση $p_i - 1$ του μέγιστου ταιριάσματος. Έπειτα διαβάζοντας τα επόμενα $\lceil log(L_S)\rceil$ βρίσκουμε και το μήκος του $l_i - 1$. Έχοντας αυτά τα δύο στοιχεία στα χέρια μας πηγαίνουμε στην θέση $p_i - 1$ της μνήμης και αντιγράφουμε τα επόμενα $l_i - 1$ στοιχεία που θα συναντήσουμε στη θέση των κωδικών λέξεων $c_{\lceil log(n-L_S)\rceil+\lceil log(L_s)\rceil}^i$ που βρίσκονται μετά την ενδιάμεση μνήμη. Κάθε φορά που αντιγράφουμε ένα σύμβολο μετακινούμε τη μνήμη μία θέση δεξιότερα. Τέλος αφού έχουν προηγηθεί $l_i - 1$ μεταθέσεις κάνουμε και μία τελευταία ώστε να εισάγουμε το σύμβολο $c_{\lceil log(n-L_S)\rceil+\lceil log(L_s)\rceil+1}^i$ στη μνήμη.

[17] Το 1 στο εκθέτη σημαίνει ότι μιλάμε για την πρώτη κωδική λέξη

Παράδειγμα 5.11. *Να αποκωδικοποιηθεί η συμβολοσειρά που κωδικοποιήθηκε στο προηγούμενο παράδειγμα/* ***Λύση***

Από το προηγούμενο παράδειγμα ξέρουμε ότι η ενδιάμεση μνήμη πρέπει να έχει μέγεθος $n - L_s = 20$ *και η κωδική συμβολοσειρά είναι η:*

$$C = \underbrace{00000000001}_{C_0}\underbrace{1001100100}_{C_1}\underbrace{1001000100}_{C_2}\underbrace{0111001001}_{C_3}\underbrace{1001001000}_{C_4}\underbrace{0001100111}_{C_5}$$

Αρχικά γεμίζουμε την μνήμη με 20 μηδενικά. Επειδή το $n-L_s = 20$ και το $L_s = 10$ ξέρουμε ότι μία κωδική λέξη C_i θα έχει μήκος $\lceil log_2(n-L_s)\rceil + \lceil log_2(L_s)\rceil + 1 = \lceil log_2 20\rceil + \lceil log_2 10\rceil + 1 = 5+4+1 = 10$. Από αυτά τα 10 σύμβολα υπενθυμίζουμε ότι τα 5 πρώτα ($\lceil log_2(n-L_s)\rceil$) αναπαριστούν τον δείκτη $p_i - 1$, τα 4 επόμενα ($\lceil log_2(L_s)\rceil$) το μήκος του μέγιστου ταιριάσματος $l_i - 1$ και το τελευταίο είναι απλά το επόμενο σύμβολο.

$$\underbrace{[00000000000000000000]}_{\text{Αρχικοποίηση μνήμης}}\underbrace{00000000001}_{C_0}\underbrace{1001100100}_{C_1}\underbrace{1001000100}_{C_2}\underbrace{0111001001}_{C_3}\underbrace{1001001000}_{C_4}\underbrace{0001100111}_{C_5}$$

Οπότε για να ξεκινήσουμε τη κωδικοποίηση μεταφράζουμε στο δεκαδικό την πρώτη κωδική λέξη $C_0 = 00000\ 0000\ 1 = (00000)_{10}(0000)_{10}(1)_{10} = (0,0,1)$. Η C_0 μας πληροφορεί ότι δεν υπάρχει κανένα ταίριασμα, αφού $p_0 - 1 = 0$, άρα απλά μετακινούμε τη μνήμη μία θέση δεξιότερα για να μπει το 1.

$$[00000000000000000001]\underbrace{1001100100}_{C_1}\underbrace{1001000100}_{C_2}\underbrace{0111001001}_{C_3}\underbrace{1001001000}_{C_4}\underbrace{0001100111}_{C_5}$$

Γενικά κατά τη διάρκεια της αποκωδικοποίησης κάθε φορά αποκωδικοποιούμε μία τριπλέτα της μορφής (p_i-1, l_i-1, s). Από το πρώτο στοιχείο της τριπλέτας βρίσκουμε το p_i δηλαδή τη θέση που πρέπει να μεταβούμε στην μνήμη κωδικοποίησης[18]. Αφού τοποθετήσουμε τον (κόκκινο) δείκτη στη σωστή θέση συμβουλευόμαστε το δεύτερο στοιχείο της τριπλέτας, το $l_i - 1$, για να δούμε πόσες αντιγραφές θα κάνουμε. Καθώς μελετάτε το παράδειγμα μπορείτε να παρατηρήσετε ότι πράγματι το πλήθος των αντιγραφών είναι όσο το $l_i - 1$. Το στοιχείο που αντιγράφεται σημειώνεται με έναν (μπλε) δείκτη. Κάθε φορά που ολοκληρώνεται μία αντιγραφή η μνήμη μετακινείται μία θέση δεξιά για να παραλάβει το καινούριο αποκωδικοποιημένο σύμβολο. Μία θέση δεξιότερα μετακινείται επίσης και ο δείκτης αποκωδικοποίησης (κόκκινος) που υποδεικνύει το αμέσως επόμενο σύμβολο που πρέπει να αντιγραφεί. Όταν γίνουν τόσες αντιγραφές όσες ορίζει το $l_i - 1$ η μνήμη κινείται μία θέση δεξιότερα για να παραλάβει το τρίτο στοιχείο της τριπλέτας που στο παράδειγμα σημαίνεται με πράσινο χρώμα.

Συνεχίζουμε μεταφράζοντας την δεύτερη κωδική λέξη $C_1 = 10011\ 0010\ 0 = (10011)_{10}(0010)_{10}(0)_{10} = (19,2,0)$. Αφού το $p_1 - 1 = 19 \Rightarrow p_1 = 20$. Μεταβαίνουμε λοιπόν στη θέση 20 της μνήμης και πραγματοποιούμε 2 αντιγραφές όπως φαίνεται παρακάτω.

[18] Η μέτρηση ξεκινάει από το 1 όχι από το 0, δηλαδή η θέση μίας μνήμης της μορφής [000] θα είναι $[\underset{1}{0}\underset{2}{0}\underset{3}{0}]$. Αν η αρίθμηση άρχιζε από το 0 τότε θα χρησιμοποιούσαμε τον δείκτη $p_i - 1$ και όχι τον p_i. Εν κατακλείδι η χρήση του p_i ή του $p_i - 1$ είναι θέμα αρίθμησης που δεν επηρεάζει την διαδικασία αποκωδικοποίησης. Εμείς σε αυτό το παράδειγμα επιλέξαμε τη σύμβαση η αρίθμηση να ξεκινάει από το 1 και για το λόγο αυτό χρησιμοποιούμε το p_i αντί του $p_i - 1$

$$[00000000000000000001\downarrow\!1]\downarrow\!10\ \underbrace{1001000100}_{C_2}\ \underbrace{0111001001}_{C_3}\ \underbrace{1001001000}_{C_4}\ \underbrace{0001100111}_{C_5}$$ (Αντιγραφή)

$$[000000000000000000011\downarrow]0\ \underbrace{1001000100}_{C_2}\ \underbrace{0111001001}_{C_3}\ \underbrace{1001001000}_{C_4}\ \underbrace{0001100111}_{C_5}$$ (Μετακίνση μία θέση δεξιά)

$$[00000000000000000011\downarrow]\downarrow\!10\ \underbrace{1001000100}_{C_2}\ \underbrace{0111001001}_{C_3}\ \underbrace{1001001000}_{C_4}\ \underbrace{0001100111}_{C_5}$$ (Αντιγραφή)

$$[000000000000000000111\downarrow]0\ \underbrace{1001000100}_{C_2}\ \underbrace{0111001001}_{C_3}\ \underbrace{1001001000}_{C_4}\ \underbrace{0001100111}_{C_5}$$ (Μετακίνση μία θέση δεξιά)

$$[000000000000000001111\downarrow0]\ \underbrace{1001000100}_{C_2}\ \underbrace{0111001001}_{C_3}\ \underbrace{1001001000}_{C_4}\ \underbrace{0001100111}_{C_5}$$ (Μετακίνση μία θέση δεξιά)

Από τον κώδικα για την λέξη $C_2 = 10010\ 0010\ 0 = (10010)_{10}(0010)_{10}0 = (18, 2, 0)$ έχουμε:

$$[0000000000000000011\downarrow\!10]\downarrow\!10\ \underbrace{0111001001}_{C_3}\ \underbrace{1001001000}_{C_4}\ \underbrace{0001100111}_{C_5}$$ (Αντιγραφή)

$$[0000000000000000011\downarrow\!01]0\ \underbrace{0111001001}_{C_3}\ \underbrace{1001001000}_{C_4}\ \underbrace{0001100111}_{C_5}$$ (Μετακίνση μία θέση δεξιά)

$$[0000000000000000111\downarrow\!01]\downarrow\!00\ \underbrace{0111001001}_{C_3}\ \underbrace{1001001000}_{C_4}\ \underbrace{0001100111}_{C_5}$$ (Αντιγραφή)

$$[0000000000000000111\downarrow\!01\downarrow\!0]0\ \underbrace{0111001001}_{C_3}\ \underbrace{1001001000}_{C_4}\ \underbrace{0001100111}_{C_5}$$ (Μετακίνση μία θέση δεξιά)

$$[0000000000000000111\downarrow\!01\downarrow\!00]\ \underbrace{0111001001}_{C_3}\ \underbrace{1001001000}_{C_4}\ \underbrace{0001100111}_{C_5}$$ (Μετακίνση μία θέση δεξιά)

Από τον κώδικα για την $C_3 = 01110\ 0100\ 1 = (01110)_{10}(0100)_{10}(1)_{10} = (14, 4, 0)$ έχουμε:

$[000000000000011\downarrow 10100]\downarrow 11\ \underbrace{1001001000}_{C_4}\ \underbrace{0001100111}_{C_5}$ (Αντιγραφή)

$[0000000000001\downarrow 1101001\downarrow]1\ \underbrace{1001001000}_{C_4}\ \underbrace{0001100111}_{C_5}$ (Μετακίνση μία θέση δεξιά)

$[000000000000\downarrow 11010011\downarrow]11\ \underbrace{1001001000}_{C_4}\ \underbrace{0001100111}_{C_5}$ (Αντιγραφή)

$[00000000000\downarrow 10100111\downarrow]1\ \underbrace{1001001000}_{C_4}\ \underbrace{0001100111}_{C_5}$ (Μετακίνση μία θέση δεξιά)

$[0000000000\downarrow 01001111\downarrow]01\ \underbrace{1001001000}_{C_4}\ \underbrace{0001100111}_{C_5}$ (Αντιγραφή)

$[000000000\downarrow 10011110\downarrow]1\ \underbrace{1001001000}_{C_4}\ \underbrace{0001100111}_{C_5}$ (Μετακίνση μία θέση δεξιά)

$[00000000\downarrow 01001101\downarrow]11\ \underbrace{1001001000}_{C_4}\ \underbrace{0001100111}_{C_5}$ (Αντιγραφή)

$[0000000\downarrow 10011010\downarrow]1\ \underbrace{1001001000}_{C_4}\ \underbrace{0001100111}_{C_5}$ (Μετακίνση μία θέση δεξιά)

$[000000\downarrow 01001101\downarrow]11\ \underbrace{1001001000}_{C_4}\ \underbrace{0001100111}_{C_5}$ (Μετακίνση μία θέση δεξιά)

Από τον κώδικα για την $C_4 = 10010\ 0100\ 0 = (10010)_{10}(0100)_{10}(0)_{10} = (18, 4, 0)$ έχουμε:

$$[000000001110100110\overset{\downarrow}{1}1]\overset{\downarrow}{1}0\underbrace{0001100111}_{C_5} \text{ (Αντιγραφή)}$$

$$[0000000111010011010\overset{\downarrow}{1}1\overset{\downarrow}{1}]0\underbrace{0001100111}_{C_5} \text{ (Μετακίνση μία θέση δεξιά)}$$

$$[000000011101001101\overset{\downarrow}{1}1]\overset{\downarrow}{1}0\underbrace{0001100111}_{C_5} \text{ (Αντιγραφή)}$$

$$[0000001110100110101\overset{\downarrow}{1}1\overset{\downarrow}{1}]0\underbrace{0001100111}_{C_5} \text{ (Μετακίνση μία θέση δεξιά)}$$

$$[000000111010011010\overset{\downarrow}{1}1]\overset{\downarrow}{1}0\underbrace{0001100111}_{C_5} \text{ (Αντιγαφή)}$$

$$[0000011101001101011\overset{\downarrow}{1}1\overset{\downarrow}{1}]0\underbrace{0001100111}_{C_5} \text{ (Μετακίνση μία θέση δεξιά)}$$

$$[000001110100110101\overset{\downarrow}{1}1]\overset{\downarrow}{1}0\underbrace{0001100111}_{C_5} \text{ (Αντιγραφή)}$$

$$[0000111010011010111\overset{\downarrow}{1}1\overset{\downarrow}{1}]0\underbrace{0001100111}_{C_5} \text{ (Μετακίνση μία θέση δεξιά)}$$

$$[000111010011010111\overset{\downarrow}{1}1\overset{\downarrow}{1}0]\underbrace{0001100111}_{C_5} \text{ (Μετακίνση μία θέση δεξιά)}$$

Ολοκληρώνοντας τη διαδικασία αποκωδικοποιούμε το $C_5 = 00011\ 0011\ 1 = (00011)_{10}(0011)_{10}(1)_{10} = (3, 3, 1)$.

$$[000\overset{\downarrow}{1}1101001101111110]\overset{\downarrow}{1}1 \text{ (Αντιγραφή)}$$

$$[00\overset{\downarrow}{1}11010011011111101\overset{\downarrow}{]}1 \text{ (Μετακίνση μία θέση δεξιά)}$$

$$[001\overset{\downarrow}{1}1010011011111101]\overset{\downarrow}{1}1 \text{ (Αντιγραφή)}$$

$$[01\overset{\downarrow}{1}10100110111111011\overset{\downarrow}{]}1 \text{ (Μετακίνση μία θέση δεξιά)}$$

$$[011\overset{\downarrow}{1}0100110111111011]\overset{\downarrow}{1}1 \text{ (Αντιγραφή)}$$

$$[11\overset{\downarrow}{1}01001101111110111\overset{\downarrow}{]}1 \text{ (Μετακίνηση μία θέση δεξιά)}$$

$$1[1\overset{\downarrow}{1}0100110111111011\overset{\downarrow}{1}1] \text{ (Μετακίνηση μία θέση δεξιά)}$$

Άρα η αποκωδικοποιημένη ακολουθία είναι η11101001101111110111 1 που προφανώς είναι ίδια με αυτήν που δίνεται στο παράδειγμα 5.3.

5.6.2 Ανάλυση της μεθόδου

Στο κομμάτι αυτό θα αποδείξουμε πως αν μία πηγή είναι στάσιμη και εργοδική τότε το μέσο μήκος κώδικα που παράγεται από τον $LZ77$ συγκλίνει στην εντροπία της πηγής. Μπορεί ο LZ77 να μην υποθέτει κανένα πιθανοκρατικό μοντέλο για την πηγή που κωδικοποιεί αλλά μπορούμε να πούμε πως το crash test για κάθε τεχνική συμπίεσης είναι μία στάσιμη και εργοδική πηγή καθώς μοντελοποιεί τη δομή που περιμένουμε να έχει

μια πηγή.Η δομή αυτή δεν θα εξαρτάται ούτε από το χρόνο στον οποίο παράγονται οι συμβολοσειρές, ούτε από συγκεκριμένες συμβολοσειρές. Άρα είναι διαισθητικά εύκολο να σκεφτούμε πως αν η τεχνική συμπίεσης όταν εφαρμόζεται σε μία στάσιμη και εργοδική πηγή παράγει ένα μέσο μήκος κώδικα που δεν συγκλίνει στον ρυθμό εντροπίας της πηγής, δεν θα τα πηγαίνει καλά για την πλειοψηφία των πηγών που υπάρχουν στον πραγματικό κόσμο. Η μόνη σύμβαση που θα γίνει στην απόδειξη είναι ότι το μήκος του παραθύρου είναι άπειρο. Δηλαδή ο κωδικοποιητής έχει πρόσβαση σε όλο το παρελθόν της ακολουθίας.

Λήμμα 5.2. [19] *Υπάρχει ένας στιγμιαίος κώδικας για τους ακεραίους έτσι ώστε η κωδική λέξη για τον ακέραιο k να έχει μήκος $logk + 2 \cdot log(logk) + \mathcal{O}(1)$*[20]

Απόδειξη

Αν γνωρίζαμε ένα φράγμα για τον k θα τον κωδικοποιούσαμε με βάση τον λογάριθμο του άνω φράγματος. Επειδή όμως δεν γνωρίζουμε το μήκος του ακέραιου επιλέγουμε να το κωδικοποιήσουμε ως εξής:

$$\underbrace{0 \ \cdots 0}_{\lceil logk \rceil} 1 \underbrace{b_1 b_2 \cdots b_n}_{\lceil logk \rceil}$$

, όπου το $b_1 b_2 \cdots b_n$ αποτελεί τη δυαδική αναπαράσταση του k. Το μήκος της συγκεκριμένης κωδικοποίησης είναι $2 \cdot \lceil logk \rceil + 1 < 2 \cdot (logk + 1)1 = 2 \cdot logk + 3$. Αν χρησιμοποιήσουμε την παραπάνω κωδικοποίηση για το $logk$ θα έχουμε:

$$\underbrace{0 \ \cdots 0}_{\lceil log(logk) \rceil} \ 1 \underbrace{b_1 b_2 \cdots x_n}_{\lceil log(logk) \rceil}$$

Άρα εν τέλει το μήκος της κωδικοποίησης θα είναι $2\lceil log(logk) \rceil + 1 < 2(log(logk) + 1) + 1 < log(logk) + log(logk) + 3 < logk + 1 + log(logk) + 3 < logk + log(logk) + 4 < logk + 2 \cdot log(logk) + 4$

Ορίζουμε ως $R_n(X_0, X_1, \cdots X_{n-1}) = max\{j < 0 : (X_{-j},\ X_{-j-1}, \cdots, X_{-j-n+1}) = (X_0,\ X_1,\ \cdots, X_{n-1})\}$. Το R_n είναι ο χρόνος που είδαμε για τελευταία φορά την ακολουθία $X_0,\ X_1,\ \cdots, X_{n-1}$. Πρέπει σε αυτό το σημείο να κατανοήσουμε ότι το μέσο μήκος που παράγεται από τον κώδικα είναι ίσο με το μέσο μήκος της κωδικοποίησης του R_n που στέλνουμε στον αποκωδικοποιητή. Επειδή έχουμε πρόσβαση σε όλο το παρελθόν της διαδικασίας και ξέρουμε το μήκος του ταιριάσματος που ψάχνουμε το μόνο που δεν ξέρουμε είναι αν υπάρχει ταίριασμα και που ακριβώς στο παρελθόν βρίσκεται. Για το λόγο αυτό το ζητούμενο στην προκειμένη απόδειξη είναι να δούμε που τείνει η ποσότητα $\frac{1}{n}E[log(R_n)]$. Οι υπόλοιποι όροι της κωδικοποίησης του R_n μπορούν να φραχτούν από το όριο της ποσότητας $\frac{1}{n}E[log(R_n)]$.

Στην απόδειξη του LZ77 θα χρησιμοποιήσουμε το λήμμα του Kac που διατυπώνεται ως εξής:

Λήμμα 5.3. ***(Kac)*** *Έστω $U_{-\infty}, \cdots, U_2, U_1, U_0, U_1, U_2, \cdots, U_{+\infty}$ μία στάσιμη και εργοδική διαδικασία ορισμένη σε ένα αριθμήσιμο αλφάβητο. Ορίζουμε την πιθανότητα της τελευταίας συνάντησης ενός συμβόλου $u, Pr[U = u] > 0$ πριν τον χρόνο 0 με:*

$$Q_U(i) = Pr[U_{-i} = u, U_{-j} \neq u \forall -i < j < 0 | U_0 = u]$$

Τότε:

$$E[R_1(U)|X_0 = u] = \sum_i i \cdot Q_U(i) = \frac{1}{Pr[U = u]}$$

[19] **cover2012elements**.

[20] Ο στιγμιαίος κώδικας που αναφέρεται στο λήμμα είναι ο κώδικας γάμμα του Elias

Για να καταλάβουμε γιατί ισχύει το λήμμα του Kac μπορούμε να σκεφτούμε ως εξής. Όσο πιο πιθανό είναι ένα σύμβολο τόσο πιο σύντομα θα το συναντήσουμε οπότε η μέση τιμή θα είναι μικρότερη, οπότε η αντίστροφη πιθανότητα μεγαλύτερη. Ένας άλλος τρόπος να σκεφτούμε το παραπάνω λήμμα είναι μέσω του πλήθους των φορών που θα συναντήσουμε ένα σύμβολο u σε μία μεγάλη συμβολοσειρά μήκους n που παρήγαγε η πηγή. Αφού η πηγή είναι στάσιμη και εργοδική τότε το πλήθος των φορών που θα συναντήσουμε το u είναι $nPr[U=u]$. άρα η μέση απόσταση ενδιάμεσα των συναντήσεων θα είναι $\frac{n}{nPr[U=u]} = \frac{1}{Pr[U=u]}$

Πόρισμα 5.1. *Έστω $X_{-\infty}, \cdots, X_2, X_1, X_0, X_1, X_2, \cdots, X_{+\infty}$ μία στάσιμη και εργοδική διαδικασία με $R_n(X_0, \cdots X_n)$ ο χρόνος τελευταίας συνάντησης της συμβολοσειράς $(X_0, \cdots X_n)$ πριν τη χρονική στιγμή 0. Τότε:*

$$E[R_n(X_0, \cdots X_n)|(X_0, \cdots X_n)] = \frac{1}{Pr[\mathbf{X_0^{n-1}} = \mathbf{x_0^{n-1}}]}$$

Απόδειξη

Με βάση την διαδικασία $\{X_{-\infty}^{\infty}\}$ φτιάχνουμε την $\{U_{-\infty}^{\infty}\}$ με $U_i = (X_i, \cdots X_{i+n-1})$. Η διαδικασία $\{U_{-\infty}^{\infty}\}$ είναι και αυτή στάσιμη και εργοδική με πιθανότητες $Pr[U_i = u_i] = Pr[\mathbf{X_i^{n-1}} = \mathbf{x_i^{n-1}}]$. Τότε από το λήμμα του Kac θα έχουμε ότι

$$E[R_1(U)|U_0 = u] = \frac{1}{Pr[U=u]} \Rightarrow E[R_n(X_0, \cdots X_n)|(X_0, \cdots X_n)] = \frac{1}{Pr[\mathbf{X_0^{n-1}} = \mathbf{x_0^{n-1}}]}$$

Θεώρημα 5.3. *Έστω $L_n(X_0,\ X_1,\ \cdots, X_{n-1}) = logR_n + 2log(logR_n) + \mathcal{O}(1)$, το μήκος του ταιριάσματος για τον LZ77 με άπειρο παράθυρο. Τότε:*

$$\lim_{n\to\infty} \frac{1}{n} E(L_n(X_0,\ X_1,\ \cdots, X_{n-1})) = H(\mathcal{X})$$

, με $H(\mathcal{X})$ ο ρυθμός εντροπίας της πηγής.

Απόδειξη

Αρχικά θα δείξουμε ότι $\lim_{n\to\infty}\{sup_{m\geq n}\frac{1}{m}\cdot E[log(R_m)]\} \leq H$

$$\frac{1}{n}\cdot E[log(R_n)] =$$

$$\frac{1}{n}\cdot \sum_{\mathbf{x_0^{n-1}}\in\ \mathcal{X}^n} E[log(R_n(X_0, \cdots, X_{n-1}))|X_0, \cdots, X_{n-1} = x_0, \cdots, x_{n-1}]\cdot Pr[X_0 = x_0, \cdots X_n = x_n] \overset{Ανισότητα\ Jensen}{\leq}$$

$$\frac{1}{n}\cdot \sum_{\mathbf{x_0^{n-1}}\in\ \mathcal{X}^n} log(E[(R_n(X_0, \cdots, X_{n-1}))|X_0, \cdots, X_{n-1} = x_0, \cdots, x_{n-1}])\cdot Pr[X_0 = x_0, \cdots X_n = x_n]$$

Από το λήμμα του Kac για εργοδικές πηγές όμως γνωρίζουμε ότι ο μέσος χρόνος για να συναντήσουμε μία συμβολοσειρά $x_0, x_1 \cdots x_n$ θα είναι ίσος με το αντίστροφο της πιθανότητας της συμβολοσειράς, οπότε

$$\frac{1}{n}\cdot \sum_{\mathbf{x_0^{n-1}}\in\ \mathcal{X}^n} log(E[(R_n(X_0, \cdots, X_{n-1}))|X_0, \cdots, X_{n-1} = x_0, \cdots, x_{n-1}])\cdot Pr[X_0 = x_0, \cdots X_n = x_n] \overset{ΛήμμαKac}{=}$$

$$\frac{1}{n}\cdot \sum_{\mathbf{x_0^{n-1}}\in\ \mathcal{X}^n} log(\frac{1}{Pr[X_0 = x_0, \cdots X_n = x_n]})\cdot Pr[X_0 = x_0, \cdots X_n = x_n] = \frac{1}{n}H(X_0 = x_0, \cdots X_n = x_n)$$

Επειδή όμως η πηγή είναι εργοδική και στάσιμη από το θεώρημα ασυμπτωτική ισοκατανομής γνωρίζουμε πώς

$$\lim_{n\to\infty}\{sup_{m\geq n}\frac{1}{m}H(X_0=x_0,\cdots X_m=x_m)\}=H(\mathcal{X})$$

Χρησιμοποιώντας την ανισότητα Jensen για να φράξουμε και τον δεύτερο μέρος του μήκους έχουμε:

$$\frac{1}{n}E[log(logR_n((X_0=x_0,\cdots X_n=x_n))]\leq\frac{1}{n}logE[log(R_n((X_0=x_0,\cdots X_n=x_n))]\leq$$
$$\frac{1}{n}logH(X_0=x_0,\cdots X_n=x_n)$$

Επειδή όμως η πηγή είναι εργοδική έπεται ότι για μεγάλα n το $\frac{1}{n}H(X_0=x_0,\cdots X_n=x_n)\to H(\mathcal{X})\Rightarrow \frac{1}{n}H(X_0=x_0,\cdots X_n=x_n)<H(\mathcal{X})+\epsilon\Rightarrow H(X_0=x_0,\cdots X_n=x_n)<n\cdot(H(\mathcal{X})+\epsilon)\Rightarrow\frac{1}{n}logH(X_0=x_0,\cdots X_n=x_n)<\frac{1}{n}log(n\cdot(H+\epsilon))$

Άρα το $\lim_{n\to\infty}\frac{1}{n}E[L(X_0,\cdots X_n)]=\lim_{n\to\infty}\frac{1}{n}E[logR_n(X_0=x_0,\cdots X_n=x_n)]+\frac{1}{n}E[log(logR_n(X_0=x_0,\cdots X_n=x_n))]+\frac{1}{n}\mathcal{O}(1)\leq\lim_{n\to\infty}\frac{1}{n}\cdot H(X_0=x_0,\cdots X_n=x_n)+\frac{1}{n}log(n\cdot(H+\epsilon))+\frac{\mathcal{O}(1)}{n}=H(\mathcal{X}).$

Από την θεωρία ξέρουμε ότι $E[L(X_0,\cdots X_n)]\geq nH(\mathcal{X})\Rightarrow\lim_{n\to\infty}\frac{1}{n}E[L(X_0,\cdots X_n)]\geq H(\mathcal{X})$

Από το κριτήριο παρεμβολής έπεται ότι:

$$\frac{1}{n}E[L(X_0,\cdots X_n)]\to H(\mathcal{X})$$

Για να καταλάβουμε γιατί το συγκεκριμένο φράγμα μπορεί να επεκταθεί και στον πραγματικό αλγόριθμο σκεφτόμαστε ότι από την τριπλέτα (p_i-1,l_i-1,s) το στοιχείο που αναπαριστάτε με τον μεγαλύτερο ακέραιο άρα συνεισφέρει και περισσότερο στο μέσο μήκος του κώδικα είναι το p_i-1. Άρα αφού το μέσο μήκος του συγκεκριμένου ακεραίου φράσσεται από τον $H(\mathcal{X})$ έπεται ότι και τα υπόλοιπα θα φράσσονται από την ίδια ποσότητα καθώς το $n\to\infty$.

5.7 Η κωδικοποίηση LZ78

5.7.1 Παρουσίαση της μεθόδου

Ένα από τα κύρια προβλήματα του LZ77 είναι ότι προσφέρει μια τοπικά βέλτιστη συμπίεση και απαιτεί παραμετροποίηση για κάθε ξεχωριστό σύνολο δεδομένων που κωδικοποιεί. Προκειμένου να γίνει κατανοητή η παραπάνω κατάσταση φανταστείτε ένα κείμενο που περιέχει τεχνικούς όρους. Για παράδειγμα στο παρών κεφάλαιο σίγουρα αναφέρονται πλειάδα φορών οι λέξεις "κωδικοποίηση", "συμπίεση" κ.λ.π. Σύμφωνα με την λογική της συμπίεσης θα έπρεπε οι συχνά εμφανιζόμενες λέξεις του κειμένου να κωδικοποιούνται με μικρό αριθμό κωδικών συμβόλων. Αν όμως δεν έχουμε χρησιμοποιήσει τις σωστές παραμέτρους για το παρών κεφάλαιο, δηλαδή δεν υπολογίσαμε σωστά το μέγεθος του κυλιόμενου παραθύρου που δίνει βέλτιστη συμπίεση, τότε όταν συναντήσουμε μία επαναλαμβανόμενη έκφραση η μνήμη αναζήτησης, που αποτελεί το δυναμικό λεξικό συμπίεσης, μπορεί να μην περιέχει την επίμαχη έκφραση με αποτέλεσμα να οδηγηθούμε σε χειρότερη συμπίεση από την θεωρητικά βέλτιστη που επιτρέπει η εντροπία του κειμένου. Το πρόβλημα αυτό έρχεται να λύσει μια παραλλαγή του LZ77, ο LZ78 που αναπτύχθηκε πάλι από τους Lempel και Ziv[21]. Το πρωτότυπο με την παραπάνω μέθοδο κωδικοποίησης είναι ότι περιγράφτηκε στα πλαίσια της απόδειξης ενός θεωρήματος κωδικοποίησης αντίστοιχου με το αυτό του Shannon, που είδαμε στο κεφάλαιο 3, το οποίο χρησιμοποιούσε ως φράγμα για την συμπίεση μίας άπειρης ακολουθίας ($\boldsymbol{x}=x_1x_2\cdots,\ x_i\in A$) μία ποσότητα διαφορετική από

[21] **ziv1978compression**.

την κλασσική εντροπία. Μία παραλλαγή της συγκεκριμένης μεθόδους, ο LZW, έμελλε να γίνει ένας από του πιο συχνά χρησιμοποιούμενους αλγορίθμους συμπίεσης μέχρι και σήμερα.

Η περιγραφή του LZ78 θα ακολουθήσει την πρωταρχική παρουσίαση του με κάποιες μικρές διαφοροποιήσεις που θα συμπεριληφθούν κατά την ανάλυση της κωδικοποίησης. Πριν παρουσιάσουμε την μέθοδο θα δώσουμε κάποιους ορισμούς που θα φανούν χρήσιμοι στην περιγραφή της μεθόδους σύμφωνα με το πρωτότυπο κείμενο.

Ορισμός 5.1. *Ως* ***κωδικοποιητή*** $\mathcal{E}$ *ορίζουμε την πεντάδα* $(S,\ A,\ B,\ g,\ f)$*, όπου:*

1. S *είναι το πεπερασμένο σύνολο καταστάσεων του κωδικοποιητή* $\mathcal{E}$*.*
2. A *είναι το αλφάβητο εισόδου του κωδικοποιητή* $\mathcal{E}$
3. B *είναι το αλφάβητο εξόδου του κωδικοποιητή* $\mathcal{E}$
4. g *είναι η συνάρτηση μετάβασης καταστάσεων* $g : S \times A \to S$ *με* $s_{i+1} = g(s_i,\ x_i)$ *και*
5. f *είναι η συνάρτηση εξόδου* $f : S \times A \to S \to B$ *με* $y_{i+1} = f(s_i, x_i)$

Παρατηρήσεις:

1. Ανάλογα με τη μορφή της εισόδου και εξόδου ο κωδικοποιητής μπορεί να υλοποιήσει μία από τις παρακάτω μορφές συμπίεσης:

   (α′) Σταθερό μήκος εισόδου σε μεταβλητό μήκος εξόδου, τότε η είσοδος και έξοδος απαρτίζονται από ακολουθίες της μορφής $input = \{x_1x_2\cdots x_m :\ x_i \in A\}$ και $output = \{y_1y_2\cdots y_n :\ n < \infty,\ y_i \in B\}$

   (β′) Μεταβλητό μήκος εισόδου σε σταθερό μήκος εξόδου, όπου $input = A^* = \{x_ix_2\cdots x_m :\ m < \infty, x_i \in A\}$ και $output = \{y_1y_2\cdots y_n :\ y_i \in B\}$

   (γ′) Σταθερό μήκος εισόδου σε σταθερό μήκος εξόδου, όπου $input = \{x_1x_2\cdots x_m :\ x_i \in A\}$ και $output = \{y_1y_2\cdots y_n :\ y_i \in B\}$

   (δ′) Μεταβλητό μήκος εισόδου σε μεταβλητό μήκος εξόδου, όπου $input = \{x_1x_2\cdots x_m :\ m < \infty,\ x_i \in A\}$ και $output = \{y_1y_2\cdots y_n :\ n < \infty,\ y_i \in B\}$

   Για παράδειγμα ένας κωδικοποιητής σταθερού μήκους εισόδου σε μεταβλητό μήκος εξόδου είναι ο Shannon που αντιστοιχεί τα σύμβολα του αλφαβήτου Α σε κωδικές λέξεις που ανήκουν στο B^* και έχουν μήκος $\lceil log\frac{1}{Pr[X = x_i]}\rceil$

2. Ο παραπάνω κωδικοποιητής όπως ορίστηκε αποτελεί μια μηχανή πεπερασμένων καταστάσεων όπως αυτή παρουσιάζεται στην θεωρία υπολογισιμότητας.

3. Για να επεκτείνουμε τον κωδικοποιητή ώστε να κωδικοποιεί ακολουθίες συμβόλων του αλφαβήτου Α, πρέπει να επεκτείνουμε τους ορισμούς των συναρτήσεων μετάβασης και εξόδου ως εξής:

   (α′) $\overline{g} : A^n \times S \to S$ με $s_{n+1} = g(s_1, x_1^n)$ και

   (β′) $\overline{g} : A^n \times S \to B$ με $y_1^n = g(s_1, x_1^n)$

Ορισμός 5.2. *Ένας κωδικοποιητής* $\mathcal{E}$ *λέγεται* ***κωδικοποιητής χωρίς απώλειες πληροφορίας*** *(Information lossless encoder- IL) αν για κάθε αρχική κατάσταση* $s_1 \in S$ *και για κάθε ακολουθία* $x_1^n \in A^n,\ n \geq 1$ *η τριάδα* $(s_1,\ f(s_1,\ x_1^n),\ g(s_1,\ x_1^n)) = (s_1,\ y_1^n,\ z_{n+1})$ *προσδιορίζει μοναδικά την* x_1^n

Ορισμός 5.3. *Ένας κωδικοποιητής $\mathcal{E}$ λέγεται* **κωδικοποιητής χωρίς απώλειες πληροφορίας πεπερασμένης διάστασης** *(Information lossless finite order encoder- ILF) αν υπάρχει κάποιος θετικός ακέραιος $m > 0$ τέτοιος ώστε για κάθε αρχική κατάσταση $s_1 \in S$ και για κάθε ακολουθία πεπερασμένου μήκους $x_1^m \in A^m,\ n \geq 1$ το ζεύγος $(s_1,\ f(s_1,\ x_1^m)) = (s_1,\ y_1^n)$ προσδιορίζει μοναδικά την x_1^n*

Οι Lempel-Ziv όρισαν τον LZ78 σαν ένα κωδικοποιητή $\mathcal{E}$ χωρίς απώλειες πληροφορίας πεπερασμένης διάστασης, ο οποίος υλοποιεί μία κωδικοποίηση σταθερής εισόδου σε μεταβλητή έξοδο και περιέχει ένα εσωτερικό κώδικα που πραγματοποιεί μία κωδικοποιήσει μεταβλητής εισόδου σε μεταβλητή έξοδο για το εκάστοτε σταθερό κομμάτι εισόδου (input block). Κάθε φορά που τερματίζει η κωδικοποίηση μιας εισόδου ο κωδικοποιητής επανέρχεται στην αρχική του κατάσταση "σβήνοντας" το παρελθόν της πρότερης κωδικοποίησης. Ο εσωτερικός κώδικας του $\mathcal{E}$ "παραγοντοποιεί" κάθε είσοδο x_1^n σε μία ακολουθία διαφορετικών λέξεων[22]

Διαδικασία κωδικοποίησης

1. Για να αρχικοποιήσουμε την διαδικασία κωδικοποίησης μετατρέπουμε την είσοδο x_1^n στην ϵx_1^n, όπου ϵ το κενό σύμβολο, το οποίο το θεωρούμε σαν την αρχική λέξη $x_{n_{-1}+1}^{n_0} = \epsilon$
2. $x_1^n = \epsilon x_1^n = x_{n_{-1}}^{n_0} x_{n_0+1}^{n_1} x_{n_1+1}^{n_2} \cdots x_{n_{k-1}+1}^{n_k} x_{n_k+1}^{n}$
3. Κάθε λέξη $x_{n_{i-1}+1}^{n_i}$ παράγεται ως εξής:
   (α′) Έστω $d(x_{n_{j-1}+1}^{n_j})$ η λέξη που παράγεται από την $x_{n_{j-1}+1}^{n_j}$ αν διαγράψουμε το τελευταίο σύμβολό της.
   (β′) ξεκινώντας από την θέση $n_{j-1}+1$ της εισόδου x_1^n βρίσκουμε τον μεγαλύτερο ακέραιο n_j για τον οποίο η λέξη $d(x_{n_{j-1}+1}^{n_j})$ υπάρχει ήδη στο επεξεργασμένο μέρος της εισόδου x_1^n, δηλαδή $d(x_{n_{j-1}+1}^{n_j}) = x_{n_{i-1}+1}^{n_i}$.
4. Αφού βρούμε το $i < j$ για το οποίο $d(x_{n_{j-1}+1}^{n_j}) = x_{n_{i-1}+1}^{n_i}$, κωδικοποιούμε την $x_{n_{j-1}+1}^{n_j}$ ως εξής:
   (α′) Θέτουμε $\pi(j) = i$ και
   (β′) Η κωδικοποίηση γίνεται αντιστοιχίζοντας στην λέξη $x_{n_{j-1}+1}^{n_j}$ τον ακέραιο $I(x_{n_{j-1}+1}^{n_j}) = \pi(j) \cdot |A| + I(x_{n_j}) = i \cdot |A| + z_j$, όπου $I(x_{n_j}) = z_j$ ένας προκαθορισμένος ακέραιος $0 \leq z_j \leq |A| - 1$ που αντιστοιχεί στο σύμβολο x_{n_j}. Επειδή το $i < j \leq j - 1$ και $z_j \leq |A| - 1$ έπεται ότι το $I(x_{n_{j-1}+1}^{n_j}) \leq (j-1) \cdot |A| + |A| - 1 = \ \cdot|A| - 1 < j \cdot |A|$. Άρα το μήκος της κωδική λέξη $L_j = \lceil log_{|B|}(j \cdot |A|) \rceil$, όπου $|B|$ ο πληθάριθμος του αλφαβήτου εξόδου.
   (γ′) Αποθηκεύουμε στο λεξικό την εγγραφή $(i, C(x_{n_j}))$
5. Όταν τελειώσει η πρώτη είσοδος x_1^n εισάγουμε στον $\mathcal{E}$ τις εισόδους $x_{n+1}^{2\cdot n},\ x_{2\cdot n+1}^{3\cdot n}, \cdots x_{(l-1)\cdot n+1}^{l\cdot n}$ κατά σειρά.

Παρατήρηση:

• Μπορεί στην δημοσίευση να υποστηρίζουν οι συγγραφείς ότι μετά από κάθε είσοδο ο κωδικοποιητής ξεχνάει το παρελθόν κωδικοποίησης, αλλά μία τέτοια απόφαση άπτεται των αναγκών της κάθε εφαρμογής αν θα εφαρμοσθεί η όχι. Για παράδειγμα αν σαν είσοδο θεωρούμε ένα ολόκληρο αρχείο από ένα σύνολο αρχείων καθορισμένου μήκους, τότε ίσως θα έπρεπε μετά την κωδικοποίηση του να διαγράφεται η μνήμη του $\mathcal{E}$ καθώς το περιεχόμενο των αρχείων μπορεί να διαφέρει σημαντικά με αποτέλεσμα οι λέξεις που έχουν βρεθεί σε μία είσοδο να μην εμφανιστούν σε κάποια άλλη. Αν ως είσοδο θεωρήσουμε ένα κομμάτι του αρχείου ίσως το παρελθόν κωδικοποίησης δεν θα έπρεπε να σβήνεται γιατί λέξεις που θα απαντηθούν στο μέλλον έχουν ήδη εμφανιστεί με μεγάλη πιθανότητα και στο παρελθόν.

[22]Με την έννοια λέξη εννοούμε κάθε υπακολουθία της μορφής x_i^j με $1 \leq i < j \leq n$.

Παράδειγμα 5.12. *Να κωδικοποιηθεί σύμφωνα με τον LZ78 η συμβολοσειρά "AABBBAA", που παράγεται από ένα σύνολο εισόδου* $\{A, B, \}$. *Το σύνολο εξόδου είναι το* $\{0, 1\}$.

Λύση

1. *Σαν είσοδο παίρνουμε όλη την συμβολοσειρά "AABBBAA" και την μετατρέπουμε σε "ϵAABBBAA" αρχικοποιώντας το λεξικό με την εγγραφή:*

Δείκτης	*Εγγραφή*	$\pi(j)$	*Λέξη*	*Κωδική λέξη*
0	*(0, ϵ)*	*0*	ϵ	*null*

Δηλαδή $x_{n-1}^{n_0} = \epsilon$ *με* $I(x_{n-1}^{n_0}) = 0 \cdot |A| + I(x_{n-1}^{n_0}) = 0 + 0 = 0$. *Άρα* $L_0 = \lceil log(j \cdot |A|) \rceil = 1$, *λόγω σύμβασης και* $C(x_{n-1}^{n_0} = \epsilon) = 0$

2. *Επειδή το A συναντάται για πρώτη φορά εισάγουμε στο λεξικό την εγγραφή (0,A), ο ακέραιος I που αντιστοιχεί στην* $x_{n_0+1}^{n_1} = x_1^1$ *είναι* $I(x_1^1) = \pi(1) \cdot 2 + I(x_1^1) = 0 \cdot 2 + 0 = 0$.

Δείκτης	*Εγγραφή*	$\pi(j)$	*Λέξη*	*Κωδική λέξη*
0	*(0, ϵ)*	*0*	ϵ	*null*
1	*(0, A)*	*0*	*A*	*0*

Άρα $L_1 = \lceil log(j \cdot |A|) \rceil = \lceil log_2(1 \cdot 2) \rceil = 1$, $C(x_1^1) = 1$ *και "ϵ|A|ABBBAA"*

3. *Το A υπάρχει στο λεξικό και το B όχι αρά η επόμενη εγγραφή είναι η (1,B) με* $x_{n_1+1}^{n_2} = x_2^3$, $\pi(j) = 1$ $I(x_2^3) = 1 \cdot 2 + 1 = 3$

Δείκτης	*Εγγραφή*	$\pi(j)$	*Λέξη*	*Κωδική λέξη*
0	*(0, ϵ)*	*0*	ϵ	*null*
1	*(0, A)*	*0*	*A*	*0*
2	*(1, B)*	*1*	*AB*	*11*

Άρα $L_2 = \lceil log(j \cdot |A|) \rceil = \lceil log_2(2 \cdot 2) \rceil = 2$, $C(x_2^3) = 11$ *και "ϵ|A|AB|BBAA"*

4. *Το B δεν υπάρχει στο λεξικό εισάγουμε την εγγραφή (0,B) με* $x_{n_2+1}^{n_3} = x_4^4$, $\pi(3) = 0$ *και* $I(x_4^4) = 0 \cdot 2 + 1 = 1$.

Δείκτης	*Εγγραφή*	$\pi(j)$	*Λέξη*	*Κωδική λέξη*
0	*(0, ϵ)*	*0*	ϵ	*null*
1	*(0, A)*	*0*	*A*	*0*
2	*(1, B)*	*1*	*AB*	*11*
3	*(0, B)*	*0*	*B*	*001*

Άρα $L_3 = \lceil log(j \cdot |A|) \rceil = \lceil log_2(3 \cdot 2) \rceil = 3$, $C(x_4^4) = 001$ *και "ϵ|A|AB|B|BAA"*

5. *Το B υπάρχει στο λεξικό και το BA όχι, οπότε η εγγραφή είναι η (3,A) με* $x_{n_3+1}^{n_4}$, $\pi(4) = 3$ $I(x_5^6) = 3 \cdot 2 + 0 = 6$

Δείκτης	Εγγραφή	$\pi(j)$	Λέξη	Κωδική λέξη
0	(0, ϵ)	0	ϵ	null
1	(0, A)	0	A	0
2	(1, B)	1	AB	11
3	(0, B)	0	B	001
4	(3, A)	3	BA	110

Άρα $L_4 = \lceil log(j \cdot |A|) \rceil = \lceil log_2(4 \cdot 2) \rceil = 3$, $C(x_4^4) = 110$ *και* "$\epsilon|A|AB|B|BA|A$"

6. *Το A υπάρχει ήδη στο στο λεξικό οπότε απλά γράφουμε τη κωδική λέξη για το A στην έξοδο του κωδικοποιητή. Οπότε:*

$$\text{"}\epsilon|A|AB|B|BA|A\text{"} \overset{\mathcal{E}}{\Rightarrow} null|0110011100$$

Η αποκωδικοποίηση του LZ78 ακολουθεί παρόμοια διαδικασία με την κωδικοποίηση. Δηλαδή παίρνει την κωδικοποιημένη είσοδο μήκους m, $\mathbf{y}$, την κατακερματίζει στα κομμάτια $\mathbf{y} = y_{k_{-1}}^{k_0} y_{k_0+1}^{k_1} \cdots y_{k_{l+1}+1}^{m}$ και αποκωδικοποιεί κάθε κομμάτι ξεχωριστά χτίζοντας παράλληλα το λεξικό αποκωδικοποίησης. Η διάσπαση της ακολουθίας εξόδου $\mathbf{y}$ σε l κομμάτια και η τμηματική ανάκτηση των αρχικών συμβόλων εισόδου είναι εφικτή γιατί ο LZ78 είναι ένας ILF κωδικοποιητής. Το σημαντικό και στις δύο περιπτώσεις είναι ότι το λεξικό δεν υπάρχει από πριν, ούτε χρειάζεται να μεταφερθεί αλλά δημιουργείται καθώς κωδικοποιείται/αποκωδικοποιείται η είσοδος. Το μόνο στο οποίο πρέπει να έχουν συμφωνήσει οι δύο πλευρές είναι η μέθοδος που παράγονται οι ακέραιοι για το αλφάβητο εισόδου.

Μέθοδος αποκωδικοποίησης

1. Αρχικοποιούμε την διαδικασία θέτοντας $j = 0,\ k_j = 0,\ n_j = 0$, και εισάγοντας στο κενό λεξικό αποκωδικοποίησης την εγγραφή $(0,\ \epsilon)$.
2. Υπολογίζουμε το $k_{j+1} = k_j + \lceil log(j+1) \cdot |A| \rceil$
3. Παίρνουμε το κομμάτι $y_{k_j+1}^{k_j}$ της κωδικοποιημένης συμβολοσειράς και το μετατρέπουμε στον ακέραιο $I(x_{n_j+1}^{n_{j+1}})$
4. Βρίσκουμε τα μοναδικά i και r για τα οποία ισχύει $I(x_{n_j+1}^{n_{j+1}}) = i \cdot |A| + r$[23]
5. Χρησιμοποιώντας το i πηγαίνουμε στην εγγραφή i στο λεξικό και βρίσκουμε τη λέξη $x_{n_{i-1}+1}^{n_i}$. Από το r αποκωδικοποιούμε το σύμβολο x του αλφαβήτου A για το οποίο ισχύει $I(x) = r$.
6. Αντικαθιστούμε την κωδική λέξη $y_{k_j+1}^{k_{j+1}}$ με την $x_{n_{i-1}+1}^{n_i}$ και αυξάνουμε το j κατά 1.

Παράδειγμα 5.13. *Να αποκωδικοποιηθεί η ακολουθία εξόδου που προέκυψε από το προηγούμενο παράδειγμα:* $null|0110011100$.

Λύση

1. *Αρχικοποιούμε το λεξικό κωδικοποίησης :*

Δείκτης	Εγγραφή	$\pi(j)$	Λέξη	Κωδική λέξη
0	(0, ϵ)	0	ϵ	null

[23] Ξέρουμε ότι τα i και r που θα βρούμε είναι μοναδικά καθώς δεν πραγματοποιούμε τίποτα περισσότερο από μία ευκλείδεια διαίρεση του $I(x_{n_j+1}^{n_{j+1}})$ με το $|A|$.

2. *Για* $j = 0$ *έχουμε:*

   (α′) *Για* $j = 0,\ k_0 = 0,\ n_0 = 0$, *υπολογίζουμε* $k_{j+1} = k_j + \lceil log(j+1) \cdot |A| \rceil \Rightarrow k_{0+1} = k_0 + \lceil log(0+1) \cdot 2 \rceil \Rightarrow k_1 = k_0 + \lceil log_2(2) \rceil \Rightarrow k_1 = 1$. *Άρα το* $y_{k_0+1}^{k_1} = y_1^1 = 0$.

   (β′) *Η δεκαδική αναπαράσταση του 0 είναι το 0 (*$(0)_{10} = 0$*).*

   (γ′) $0 = 0 \cdot |A| + 0$, *άρα* $i = 0$ *και* $r = 0$.

   (δ′) *Η εγγραφή 0 του πίνακα είναι το κενό σύμβολο και* $x \in A : I(x) = 0$ *είναι το A.*

   (ε′) *Αντικαθιστούμε το 0 με το A και έχουμε:*

$$null|0110011100 \Rightarrow null|A110011100$$

   (ϛ′) *Εισάγουμε στο λεξικό την εγγραφή (0, A)*

Δείκτης	*Εγγραφή*	$\pi(j)$	*Λέξη*	*Κωδική λέξη*
0	*(0,* ϵ*)*	*0*	ϵ	*null*
1	*(0, A)*	*0*	*A*	*0*

   (ζ′) *Ανανεώνουμε το* j.

3. *Για* $j = 1$ *έχουμε:*

   (α′) *Για* $j = 1,\ k_1 = 1,\ n_1 = 1$, *υπολογίζουμε* $k_{j+1} = k_j + \lceil log(j+1) \cdot |A| \rceil \Rightarrow k_{1+1} = k_1 + \lceil log(1+1) \cdot 2 \rceil \Rightarrow k_2 = 1 + log_2 4 = 3$. *Άρα το* $y_{k_1+1}^{k_2} = y_2^3 = 11$.

   (β′) *Η δεκαδική αναπαράσταση του 11 είναι το 3 (*$(11)_{10} = 3$*).*

   (γ′) $3 = 1 \cdot |A| + 1$, *άρα* $i = 1$ *και* $r = 2$.

   (δ′) *Η εγγραφή 1 του πίνακα είναι το σύμβολο A και* $x \in A : I(x) = 1$ *είναι το B.*

   (ε′) *Αντικαθιστούμε το 11 με το AB και έχουμε:*

$$null|A110011100 \Rightarrow null|AAB0011100$$

   (ϛ′) *Εισάγουμε στο λεξικό την εγγραφή (1, B)*

Δείκτης	*Εγγραφή*	$\pi(j)$	*Λέξη*	*Κωδική λέξη*
0	*(0,* ϵ*)*	*0*	ϵ	*null*
1	*(0, A)*	*0*	*A*	*0*
2	*(1, B)*	*1*	*AB*	*11*

   (ζ′) *Ανανεώνουμε το* j.

4. *Για* $j = 2$ *έχουμε:*

   (α′) *Για* $j = 2,\ k_2 = 3,\ n_2 = 3$, *υπολογίζουμε* $k_{j+1} = k_j + \lceil log(j+1) \cdot |A| \rceil \Rightarrow k_3 = k_2 + \lceil log(3 \cdot 2) \rceil = 3 + 32 \Rightarrow k_3 = 6$. *Άρα το* $y_{k_2+1}^{k_3} = y_4^6 = 001$.

   (β′) *Η δεκαδική αναπαράσταση του 001 είναι το 1 (*$(001)_{10} = 1$*).*

   (γ′) $1 = 0 \cdot |A| + 1$, *άρα* $i = 0$ *και* $r = 1$.

   (δ′) *Η εγγραφή 0 του πίνακα είναι το κενό σύμβολο* ϵ *και* $x \in A : I(x) = 1$ *είναι το B.*

   (ε′) *Αντικαθιστούμε το 001 με το B και έχουμε:*

$$null|AAB0011100 \Rightarrow null|AABB1100$$

(ς΄) Εισάγουμε στο λεξικό την εγγραφή (0, B)

Δείκτης	*Εγγραφή*	$\pi(j)$	*Λέξη*	*Κωδική λέξη*
0	*(0, ϵ)*	*0*	ϵ	*null*
1	*(0, A)*	*0*	*A*	*0*
2	*(1, B)*	*1*	*AB*	*11*
3	*(0, B)*	*1*	*B*	*001*

(ζ΄) Ανανεώνουμε το j.

5. Για $j = 3$ έχουμε:

(α΄) Για $j = 3,\ k_3 = 6,\ n_4 =$, υπολογίζουμε $k_{j+1} = k_j + \lceil log(j+1) \cdot |A| \rceil \Rightarrow k_4 = k_3 + \lceil log(3+1) \cdot 2 \rceil \Rightarrow k_4 = 6 + log_2(8) \Rightarrow k_4 = 9$. Άρα το $y_{k3+1}^{k_4} = y_7^9 = 111$.

(β΄) Η δεκαδική αναπαράσταση του 110 είναι το 6 ($(111)_{10} = 6$).

(γ΄) $6 = 3 \cdot |A| + 0$, άρα $i = 3$ και $r = 0$.

(δ΄) Η εγγραφή 3 του πίνακα είναι το σύμβολο B και $x\ \in A : I(x) = 0$ είναι το A.

(ε΄) Αντικαθιστούμε το 110 με το BA και έχουμε:

$$null|AABB1100 \Rightarrow null|AABBBA0$$

(ς΄) Εισάγουμε στο λεξικό την εγγραφή (0, B)

Δείκτης	*Εγγραφή*	$\pi(j)$	*Λέξη*	*Κωδική λέξη*
0	*(0, ϵ)*	*0*	ϵ	*null*
1	*(0, A)*	*0*	*A*	*0*
2	*(1, B)*	*1*	*AB*	*11*
3	*(0, B)*	*0*	*B*	*001*
4	*(3, B)*	*3*	*BA*	*110*

(ζ΄) Ανανεώνουμε το j.

6. Για $j = 4$ έχουμε

(α΄) Για $j = 4,\ k_4 = 9,\ n_4 = 6$ υπολογίζουμε $k_{j+1} = k_j + \lceil log(j+1) \cdot |A| \rceil \Rightarrow k_5 = k_4 + \lceil log(5 \cdot 2) \rceil \Rightarrow k_5 = 9 + 4 = 13$

(β΄) Επειδή το $k_5 = 13 > 7 = k_m$ παίρνουμε για $y_{k_4+1}^m = y_1 0^1 0 = 0$.

(γ΄) Το $(0)_1 0 = 0$ και $0 - 0 \cdot |A| + 0 \Rightarrow i = 0,\ r = 0$.

(δ΄) Άρα αντικαθιστούμε το 0 με το A. Στο λεξικό δεν γίνεται καμία εγγραφή γιατί υπάρχει ήδη εγγραφή για το A.

$$null|AABBBA0 \Rightarrow null|AABBBAA$$

5.7.2 Η ανάλυση του LZ78

Υπενθυμίζουμε ότι ο LZ78 χώριζε μία συμβολοσειρά s στις λέξεις $s_1 s_2 \cdots s_i s_{i+1} \cdots s_n$ ώστε κάθε λέξη να έχει μήκος n_i με $\sum_{i=1}^{n} = |s|$ και το πρόθεμα μήκους $n_i - 1$ να βρίσκεται ήδη στο λεξικό. Αντιλαμβανόμαστε ότι όταν έχουμε μία συμβολοσειρά που παράχθηκε από μία πηγή με μεγάλη εντροπία τότε οι λέξεις από τις οποίες απαρτίζεται θα είναι περισσότερες σε σχέση με κάποια που παράχθηκε από μία πηγή με μικρή εντροπία. Για να το καταλάβουμε αυτό ας πάρουμε δύο ακραίες περιπτώσεις, μία συμβολοσειρά που αποτελείται από επαναλήψεις τις ίδιας λέξης και μία άλλη που κατασκευάστηκε από μία πηγή με εντροπία $log|\mathcal{X}|$. Στην πρώτη καταλαβαίνουμε ότι ο συμπιεστής θα χωρίσει την λέξη μία φορά και τις υπόλοιπες θα στείλει στο δέκτη τον

δείκτη στο λεξικό με την επίμαχη λέξη. Στην άλλη ακραία περίπτωση όσο επεξεργαζόμαστε τη λέξη θα γεμίζουμε το λεξικό με εγγραφές μήκους n_i, όπου το n_i δηλώνει το βήμα της συμπίεσης.

Οπότε αν καταφέρουμε να εξάγουμε ένα γενικό φράγμα για το πλήθος των λέξεων που χωρίζει ο LZ78 την συμβολοσειρά s θα καταφέρουμε να έχουμε και ένα φράγμα για το μέγεθός του λεξικού οπότε και για το μήκος του κώδικα. Αυτό ακριβώς έκανε και ο Ziv με τον Wyner για να αποδείξουν την βελτιστότητα του LZ78 στην περίπτωση που έχει να συμπιέσει μία ακολουθία που παράχθηκε από μία εργοδική και στάσιμη πηγή.

Θεώρημα 5.4. *Το πλήθος των λέξεων $c(n)$ στις οποίες χωρίζεται μία δυαδική ακολουθία $X_1, \cdots X_n$ φράσσεται από την ποσότητα:*

$$c(n) \leq \frac{n}{(1-\epsilon_n)\log n}$$

, όπου $\epsilon_n = \min\{1, \frac{\log(\log n)+4}{\log n}\} \overset{n\to\infty}{\to} 0$.

Απόδειξη

Έστω ότι έχουμε το μήκος $n_k = \sum_{i=1}^{k} i \cdot 2^k$ που συμβολίζει το άθροισμα των μηκών όλων των πιαθανών λέξεων που έχουν μέγεθος το πολύ k.

$$\sum_{i=1}^{k} i \cdot 2^i = 2^1 + 2 \cdot 2^2 + \cdots k \cdot 2^k \text{ και } 2\sum_{i=1}^{k} i \cdot 2^i = \cdot 2^2 + 2 \cdot 2^3 + \cdots (k-1) \cdot 2^k + k \cdot 2^{k+1}, \text{ άρα}$$

$$\sum_{i=1}^{k} i \cdot 2^i - 2 \cdot \sum_{i=1}^{k} i \cdot 2^i = \underbrace{2 + 2^2 + 2^3 + \cdots + 2^n}_{\sum_{i=1}^{k} 2^i} + n \cdot 2^{k+1} \Rightarrow -\sum_{i=1}^{k} i \cdot 2^i = 2^{k+1} - 2 + k \cdot 2^{k+1} \Rightarrow$$

$$\sum_{i=1}^{k} i \cdot 2^i = (k-1)2^{k+1} + 2$$

Το πλήθος τον εκφράσεων μέχρι μήκος k θα είναι:

$$c(n_k) = \sum_{i=1}^{k} 2^i = 2^{k+1} - 2 < 2^{k+1} < \frac{n_k}{k-1}$$

Το πλήθος φράσσεται από την παραπάνω ποσότητα καθώς όσες περισσότερες λέξεις έχουμε τόσο μικρότερο απαιτούμε να είναι το μήκος τους. Οποιοδήποτε άλλο μήκος $n_k \leq n < n_k + 1$ μπορεί να εκφραστεί ως το μήκος $n_k + \Delta$ με $\Delta < (k+1)2^{k+1}$, τότε το πλήθος των λέξεων θα φράσσεται από την ποσότητα:

$$c(n) \leq \frac{n_k}{k-1} + \frac{\Delta}{k+1} < \frac{n_k}{k-1} + \frac{\Delta}{k-1} = \frac{n_k + \Delta}{k-1} = \frac{n}{k+1}$$

Επί της ουσίας εμείς θέλουμε να βρούμε ποιο είναι το εκείνο το k που θα μας δώσει τον μεγαλύτερο αριθμό λέξεων. Προφανώς αυτό εξαρτάται από το μήκος της συμβολοσειράς, επειδή λοιπόν:

$$n_k \leq n < n_{k+1} \Rightarrow (k-1)\cdot 2^{k+1} + 2n_< k \cdot 2^{k+2} + 2 \Rightarrow (k-1)2^{k+1} \leq (k-1)\cdot 2^{k+1} + 2n_< k \cdot 2^{k+2} + 2 \leq (k+2)\cdot 2^{k+2}$$

Από την πρώτη ανίσωση έχουμε $2^k \leq n \Rightarrow k \leq \log n$ ενώ από την δεύτερη ανίσωση ξέρουμε ότι $n < k \cdot 2^{k+2} + 2 \leq (k+2)2^{k+2} \overset{k\leq \log n}{\leq} (\log n + 2)2^{k+2} \Rightarrow k + 2 \geq \log \frac{n}{(\log n + 2)}$.

$$k + 2 \geq \log \frac{n}{(\log n + 2)} \Rightarrow k + (3-1) \geq \log \frac{n}{(\log n + 2)} \Rightarrow k - 1 \geq \log n - \log(\log n + 2) - 3$$

Βγάζοντας κοινό παράγοντα τον όρο $\log n$ *έχουμε:*

$k-1 \geq \log n - \log(\log n + 2) - 3 \Rightarrow k-1 \geq \left(1 - \frac{\log(\log n+2)+3}{\log n}\right)\log n$. *Όμως* $\forall n \geq 4 \Rightarrow \log n \geq 2$, *άρα:*
$k-1 \geq \left(1 - \frac{\log(\log n+2)+3}{\log n}\right)\log n \geq \left(1 - \frac{(\log(\log n+\log n)+3)}{\log n}\right)\log n \geq \left(1 - \frac{(\log(2\log n)+3)}{\log n}\right)\log n =$
$\left(1 - \frac{(\log(2)+\log(\log n)+3)}{\log n}\right)\log n = \left(1 - \frac{(1)+\log(\log n)+3)}{\log n}\right)\log n \Rightarrow k-1 \geq \left(1 - \frac{\log(\log n)+4)}{\log n}\right)\log n$
$\Rightarrow k-1 \geq (1-\epsilon_n)\log n$

$\mu\epsilon\ \epsilon_n = \min\{1, \frac{\log(\log n)+4)}{\log n}\}$.

Με αυτό το θεώρημα οι Ziv και Lempel απέδειξαν ότι το πλήθος των λέξεων στις οποίες διασπάται μία συμβολοσειρά s φράσσεται μία ποσότητα $c(n) \leq \frac{n}{\log n}$ για μεγάλα n.

Όπως στην θεωρία της συμπίεσης προσπαθούσαμε να προσεγγίσουμε τον ρυθμό εντροπίας μία πηγής με μαρκοβιανές αλυσίδες $k-$ τάξης έτσι και εδώ θα προσπαθήσουμε να προσεγγίσουμε την διαμέριση της συμβολοσειράς s σε λέξεις. Πράγματι αν μπορούμε να προσεγγίσουμε το πλήθος των φράσεων που παράγει η ανάλυση του LZ78 με αυτόν που παράγονται από μια μαρκοβιανή αλυσίδα $k-$ τάξης, τότε θα καταφέρουμε να φράξουμε το πλήθος των φράσεων από το πλήθος των εκφράσεων της προσεγγιστικής πηγής. Τότε όμως θα μπορέσουμε να φράξουμε το μέγεθος του λεξικού και κατά επέκταση το μέσο μήκος κώδικα ανά σύμβολο από τον ρυθμό εντροπίας της προσεγγιστικής πηγής

Έστω λοιπόν ότι δίνεται μία συμβολοσειρά s που ο LZ78 την έχει επεξεργαστεί μέχρι το σημείο n. Πρακτικά αυτό σημαίνει ότι η αρχική υποσυμβολοσειρά x_1^n έχει αναλυθεί στις λέξεις $y_1 \cdots y_c$, με $y_i = x_{\nu_i}^{\nu_{i+1}-1}$. Τότε το πρόθεμα μεγέθους k κάθε λέξης θα είναι η συμβολοσειρά $s_i = x_{\nu_i-k}^{\nu_i-1}$. Ορίζουμε το $c_{l,s}$ να είναι το πλήθος των y_i μήκους l που προηγήθηκαν και έχουν πρόθεμα $s_i = s$. Τότε αν αθροίσουμε πάνω σε όλα τα προθέματα και τα μήκη των λέξεων θα πάρουμε το πλήθος των λέξεων y_i και αν αθροίσουμε τα μήκη τους θα έχουμε το μήκος της επεξεργασμένης υποσυμβολοσειράς.

$$\sum_{l,s} c_{l,s} = c \text{ και } \sum_{l,s} l(c_{l,s}) = n$$

Λήμμα 5.4. ***(Ανισότητα του Ziv).*** *Για οποιαδήποτε ανάλυση του LZ78 μια συμβολοσειρά* $x_1 \cdots x_n$ *ισχύει:*

$$\log Q_k(x_1 \cdots x_n | \mathbf{s_i}) \leq -\sum_{l,s} c_{l,s} \log c_{l,s}$$

, όπου

$\log Q_k(x_1 \cdots x_n|\mathbf{s_i}) = \sum_{i=1}^n \log Pr[X_i = x_i|\mathbf{s_i}]$

Απόδειξη

Έστω ότι η συμβολοσειρά αναλύεται από τον LZ78 σε c *το πλήθος εκφράσεις, τότε:*

$\log Q_k(x_1 \cdots x_n|\mathbf{s_i}) = \log Q_k(y_1 \cdots y_c|\mathbf{s_i}) = \prod_{i=1}^c Pr[\mathbf{Y_i} = \mathbf{y_i}|\mathbf{S_i} = \mathbf{s_i}] \Rightarrow \log Q_k(y_1 \cdots y_c|\mathbf{s_i}) = \log \prod_{i=1}^c Pr[\mathbf{Y_i} = (y_1 \cdots y_c|\mathbf{s_i}) = \sum_{i-1}^c \log Pr[\mathbf{Y_i} = \mathbf{y_i}|\mathbf{S_i} = \mathbf{s_i}] \Rightarrow \log Q_k(y_1 \cdots y_c|\mathbf{s_i}) = \sum_{l,s}\sum_{i:L(y_i)=l,s_i=s} \log Pr[\mathbf{Y_i} = \mathbf{y_i}|\mathbf{S_i} = \mathbf{s_i}] \Rightarrow$
$\log Q_k(y_1 \cdots y_c|\mathbf{s_i}) = \sum_{l,s} c_{l,s} \sum_{i:L(y_i)=l,s_i=s} \frac{1}{c_{l,s}} \log Pr[\mathbf{Y_i} = \mathbf{y_i}|\mathbf{S_i} = \mathbf{s_i}] \overset{Jensen}{\leq}$

$\sum_{l,s} c_{l,s} \sum_{i:L(y_i)=l,s_i=s} \log(\frac{1}{c_{l,s}} Pr[\mathbf{Y_i} = \mathbf{y_i}|\mathbf{S_i} = \mathbf{s_i}]) \Rightarrow \log Q_k(x_1 \cdots x_n|\mathbf{s_i}) \leq \sum_{l,s} c_{l,s} \log \frac{1}{c_{l,s}}$

Θεώρημα 5.5. *Έστω μία στάσιμη εργοδική πηγή με ρυθμός εντροπίας $H(\mathcal{X})$ και $c(n)$ το πλήθος των λέξεων στις οποιες αναλύεται μία συμβολοσειρά μήκους n της πηγής. Τότε*

$$\lim_{n\to\infty} \frac{c(n)\log c(n)}{n} \leq H(\mathcal{X}) \tag{5.11}$$

Απόδειξη

Ξεκινάμε από την ανισότητα του ziv: $logQ_k(x_1\cdots x_n|\mathbf{s_i}) \leq -\sum_{l,s} c_{l,s}\log c_{l,s} \Rightarrow \log Q_k(x_1\cdots x_n|\mathbf{s_i}) \leq -\sum_{l,s} c_{l,s}\log\frac{c_{l,s}\cdot c}{c} \Rightarrow \log Q_k(x_1\cdots x_n|\mathbf{s_i}) \leq -(\sum_{l,s} c_{l,s}\log\frac{c_{l,s}}{c}+\sum_{l,s} c_{l,s}\log c) \overset{\sum_{l,s} c_{l,s}=c}{\Rightarrow} \log Q_k(x_1\cdots x_n|\mathbf{s_i}) \leq -\sum_{l,s} c_{l,s}\log\frac{c_{l,s}}{c} - c\log c \Rightarrow \log Q_k(x_1\cdots x_n|\mathbf{s_i}) \leq -c\cdot\sum_{l,s}\frac{c_{l,s}}{c}\log\frac{c_{l,s}}{c} - c\log c$

Αν συμβολίσουμε με $\pi_{l,s}$ την ποσότητα $\frac{c_{l,s}}{c}$, τότε το $\pi_{l,s}$ είναι το πλήθος των λέξεων μήκους l και προθέματος s προς τις συνολικές λέξεις. Αν πάρουμε όλα τα μήκη l υπόψιν μας τότε το $\pi_{l,s}$ είναι η συχνότητα εμφάνισης του προθέματος s στην ανάλυση των $c-$λέξεων. Η ποσότητα έχει τις ιδιότητες

$\sum_{l,s}\pi_{l,s} = \sum_{l,s}\frac{c_{l,s}}{c} = \frac{c}{c} =$ *και* $\sum_{l,s} l\pi_{l,s} = \sum_{l,s}\frac{lc_{l,s}}{c} = \frac{n}{c}$

Ορίζουμε τις τυχαίες μεταβλητές U,V όπου $Pr[U=l,V=s]=\pi_{l,s}$, τότε $E[U]=\sum_l lPr[U=u] = \sum_l l\sum_s Pr[U=l,V=s] = \sum_l l\sum_s \pi_{l,s} = \sum_{l,s} l\frac{c_{l,s}}{c} = \frac{n}{c}$. Τότε

$\log Q_k(x_1\cdots x_n|\mathbf{s_i}) \leq -c\cdot\sum_{l,s}\frac{c_{l,s}}{c}\log\frac{c_{l,s}}{c} - c\log c \Rightarrow \log Q_k(x_1\cdots x_n|\mathbf{s_i}) \leq -c\pi_{l,s}\log\pi_{l,s} - c\log c \Rightarrow \log Q_k(x_1\cdots x_n|\mathbf{s_i}) \leq cH(U,V) - c\log c \Rightarrow$

$$\boxed{-\frac{1}{n}\log Q_k(x_1\cdots x_n|\mathbf{s_i}) \geq \frac{c}{n}\log c - \frac{c}{n}H(U,V)}.$$

Από τις ιδιότητες τις εντροπίας γνωρίζουμε ότι $H(U,V)\leq H(U)+H(V)$

Το $H(V)\leq |X|^k = k$ γιατί έχουμε υποθέσει ότι τα προθέματα θα έχουν μήκος k.

Λήμμα 5.5. *Έστω Z μία μη αρνητική τυχαία μεταβλητή που παίρνει ακέραιες τιμές με μέση τιμή μ. Τότε*

$$H(Z) \leq (\mu+1)\log(\mu+1) - \mu\log\mu$$

Με βάση το παραπάνω λήμμα το $H(U)\leq (E[U]+1)\log(E[U]+1) - E[U]\log E[U] \Rightarrow$

$H(U) \leq (\frac{n}{c}+1)\log(\frac{n}{c}+1) - \frac{n}{c}\log\frac{n}{c} \Rightarrow H(U) \leq (\frac{n}{c}+1)\log(\frac{n}{c}+1) - \frac{n}{c}\log\frac{n}{c} + \log\frac{n}{c} - \log\frac{n}{c} \Rightarrow H(U) \leq (\frac{n}{c}+1)\log(\frac{n}{c}+1) - (\frac{n}{c}+1)\log\frac{n}{c} + \log\frac{n}{c} \Rightarrow H(U) \leq (\frac{n}{c}+1)\left[\log\frac{\frac{n}{c}+1}{\frac{n}{c}}\right] + \log\frac{n}{c} \Rightarrow H(U) \leq (\frac{n}{c}+1)\log(\frac{c}{n}+1) + \log\frac{n}{c}$

Άρα το $\frac{c}{n}H(U,V) \leq \frac{c}{n}(H(U)+H(V)) \leq \frac{c}{n}((\frac{n}{c}+1)\log(\frac{c}{n}+1) + \log\frac{n}{c} + k) = \frac{c}{n}(\frac{n}{c}+1)\log(\frac{c}{n}+1) + \frac{c}{n}\log\frac{n}{c} + \frac{c}{n}k = \log(\frac{c}{n}+1) + \frac{c}{n}\log(\frac{c}{n}+1) + \frac{c}{n}\log\frac{n}{c} + \frac{c}{n}k \leq \frac{c}{n}\log\frac{n}{c} + \frac{c}{n}k + O(1) \Rightarrow$

$$\boxed{\frac{c}{n}H(U,V) \leq \frac{c}{n}\log\frac{n}{c} + \frac{c}{n}k + O(1)}$$

Επειδή η συνάρτηση $\frac{c}{n}\log\frac{n}{c}(\frac{1}{x}\log x)$ παρουσιάζει μέγιστο στην τιμή $\frac{c}{n} = \frac{1}{e}$, δηλαδή $\frac{c}{n}\log\frac{n}{c} \leq -e$. Εμείς όμως από το θεώρημα 5.4 βρήκαμε ότι $c \leq \frac{n}{\log n}(1+O(1)) \Rightarrow \frac{c}{n}\log\frac{n}{c} \leq O(\frac{\log\log n}{\log n})$

Οπότε αφού ο όρος $\frac{c}{n}\log\frac{n}{c}$ φράσσεται από μία ποσότητα που τείνει στο 0 για μεγάλα n έπεται ότι $\frac{c}{n}H(U,V) \overset{n\to\infty}{\to} 0$

Άρα από την σχέση $-\frac{1}{n}\log Q_k(x_1\cdots x_n|\mathbf{s_i}) \geq \frac{c}{n}\log c - \frac{c}{n}H(U,V)$, έπεται ότι $\frac{c(n)}{n}\log c(n) \leq$

$-\frac{1}{n}\log Q_k(x_1\cdots x_n|\mathbf{s_i}) + \epsilon_k(n)$, $\mu\epsilon$ $\epsilon_k(n) \overset{n\to\infty}{\to} 0$.

Αφού η παραπάνω ανισότητα ισχύει για κάθε μήκος n, *έπεται:*

$\lim_{n\to\infty} sup\frac{c(n)}{n}\log c(n) \le \lim_{n\to\infty} -\frac{1}{n}\log Q_k(x_1\cdots x_n|\mathbf{s_i}) = H(X_0|X_{-1},\cdot X_{-k}) \overset{k\to\infty}{\to} H(\mathcal{X})$

Θεώρημα 5.6. *Έστω Έστω μία στάσιμη εργοδική πηγή* $\{Q_n\}_{-\infty}^{\infty}$ *με ρυθμός εντροπίας* $H(\mathcal{X})$. *Έστω* $l(X_1,\cdots,X_n)$ *το μήκος της συμπιεσμένης συμβολοσειρά που παράγεται από τον LZ78, τότε*

$$\frac{1}{n}\lim_{n\to\infty} supl(X_1,\cdots,X_n) \overset{\sigma.\beta}{\le} H(\mathcal{X}). \tag{5.12}$$

Από την προηγούμενη ανάλυση βρήκαμε ότι ο LZ78 διαμερίζει την συμβολοσειρά $X_1,\cdots,X_n$ *σε* $c(n)$ *εκφράσεις. Άρα κάθε έκφραση θα έχει μήκος το πολύ* $\lceil\log c(n)\rceil < \log c(n)+1$. *Οπότε το μήκος της κωδικοποιημένης ακολουθία θα είναι το πολύ* $c(n)(\log c(n)+1)$ *Τότε*

$\frac{1}{n}\lim_{n\to\infty} supl(X_1,\cdots,X_n) = \frac{1}{n}\lim_{n\to\infty} sup(\frac{c(n)(\log c(n)+1)}{n}) = \frac{1}{n}\lim_{n\to\infty} sup(\frac{c(n)(\log c(n))}{n}+\frac{c(n)}{n})$

Από το λήμμα 5.2 ξέρουμε ότι $c(n) \le \frac{n}{\log n}(1+O(1)) \Rightarrow \frac{c(n)}{n} \le \frac{1+O(1)}{\log n} \Rightarrow \frac{c(n)}{n} \overset{n\to\infty}{\to} 0$.

Άρα από αυτή την παρατήρηση έχουμε

$\frac{1}{n}\lim_{n\to\infty} sup(\frac{c(n)(\log c(n))}{n}+\frac{c(n)}{n}) = \frac{1}{n}\lim_{n\to\infty} sup(\frac{c(n)(\log c(n))}{n}) \overset{\sigma.\beta}{\to} H(\mathcal{X})$

5.8 Η μέθοδος κωδικοποίησης LZW

Η μέθοδος LZW αποτελεί μία παραλλαγή της LZ78. Αν θυμηθούμε κατά την διαδικασία κωδικοποίησης της LZ78 τα μέρη (bolcks) της κωδικοποιημένης ακολουθίας $y_{kj-1+1}^{k_j}$ αντιστοιχούσαν σε κομμάτια της ακολουθίας εισόδου της μορφής $x_{nj-1+1}^{n_j} = Ix$, όπου I ήταν μία συμβολοσειρά που υπήρχε ήδη στο λεξικό και x το τελικό γράμμα. Η κύρια βελτίωση που πρότεινε ο Terry Welch[24] με την μέθοδο LZW ήταν να καταστήσει περιττή την μετάδοση του τελικού συμβόλου x της ακολουθίας Ix και να καταλήξει να μεταδίδει μόνο το αριθμό εγγραφής που αναφέρεται στο I. Αυτό επιτεύχθηκε αρχικοποιώντας το λεξικό με τα σύμβολα του αλφαβήτου εισόδου. Αν για παράδειγμα το αλφάβητο εισόδου ήταν τα σύμβολα που περιέχονται στην κωδικοποίηση ASCII τότε θα αρχικοποιούσαμε το λεξικό με τα 256 σύμβολα που περιλαμβάνει η ASCII.

Διαδικασία κωδικοποίησης

1. Αρχικοποιούμε το λεξικό με όλες τις εγγραφές που περιλαμβάνουν τα σύμβολα του αλφαβήτου εισόδου A
2. Ξεκινάμε εισάγοντας το πρώτο σύμβολο σε μία αρχικά κενή συμβολοσειρά I.
3. Για κάθε επόμενο σύμβολο x που εισάγεται εξετάζουμε αν το Ix ανήκει στο λεξικό.
   (α′) Αν ανήκει το αποθηκεύουμε στην συμβολοσειρά I και εισάγουμε το επόμενο
   (β′) Αν δεν ανήκει εισάγουμε την καινούργια εγγραφή Ix στο λεξικό, μεταδίδουμε τη θέση της εγγραφής του I και θέτουμε το $I = x$
   (γ′) Επαναλαμβάνουμε το 2 μέχρι να εξαντληθεί η είσοδος.

Παράδειγμα 5.14. *Να κωδικοποιήσετε την είσοδο "TO BE OR NOT TO BE" σύμφωνα με την κωδικοποίηση LZW*

Λύση

[24] **Welch1984ATF**.

Θέση	Σύμβολο
0	␣
1	B
2	E
3	N
4	O
5	R
6	T

1. Εισάγουμε το πρώτο σύμβολο T, και το αποθηκεύουμε στο διάνυσμα I. Το T υπάρχει στο πίνακα οπότε εισάγουμε το επόμενο σύμβολο

$$\overset{\downarrow}{T}O\ BE\ OR\ NOT\ TO\ BE$$

2. Το επόμενο σύμβολο είναι το O, η συμβολοσειρά IO=TO δεν υπάρχει στον πίνακα οπότε την εισάγουμε στον πίνακα, μεταδίδουμε την θέση του T (6) και θέτουμε I=O

Θέση	Σύμβολο
0	␣
1	B
2	E
3	N
4	O
5	R
6	T
7	TO

$$T\overset{\downarrow}{O}\ BE\ OR\ NOT\ TO\ BE \rightarrow (6)$$

3. Το επόμενο σύμβολο είναι το ␣, η συμβολοσειρά IA=O␣ δεν υπάρχει στον πίνακα οπότε την εισάγουμε στον πίνακα, μεταδίδουμε την θέση του I (4) και θέτουμε I=␣

Θέση	Σύμβολο
0	␣
1	B
2	E
3	N
4	O
5	R
6	T
7	TO
8	O␣

$$TO\overset{\downarrow}{␣}BE\ OR\ NOT\ TO\ BE \rightarrow (6)(4)$$

4. Το επόμενο σύμβολο είναι το B, η συμβολοσειρά IB=␣B δεν υπάρχει στον πίνακα οπότε την εισάγουμε στον πίνακα, μεταδίδουμε την θέση του I (0) και θέτουμε I=B

Θέση	*Σύμβολο*
0	␣
1	*B*
2	*E*
3	*N*
4	*O*
5	*R*
6	*T*
7	*TO*
8	0␣
9	␣*B*

$$TO\text{␣}\overset{\downarrow}{B}E\ OR\ NOT\ TO\ BE \rightarrow (6)(4)(0)$$

Συνεχίζοντας με την ίδια λογική είναι απλό να διαπιστώσουμε ότι ο πίνακας που δημιουργείται είναι ο εξής:

Θέση	*Σύμβολο*
0	␣
1	*B*
2	*E*
3	*N*
4	*O*
5	*R*
6	*T*
7	*TO*
8	0␣
9	␣*B*
10	*BE*

Θέση	*Σύμβολο*
11	*E*␣
12	␣*O*
13	*OR*
14	*R*␣
15	␣*N*
16	*NO*
17	*OT*
18	*T*␣
19	␣*T*
20	*TO*␣
21	␣*BE*

Η κωδικοποιημένη συμβολοσειρά έχει τη μορφή:(6)(4)(0)(1)(2)(0)(4)(5)(0)(3)(4)(6)(0)(7)(9)(2)

Διαδικασία Αποκωδικοποίησης

1. Δημιουργούμε τον πίνακα του αλφαβήτου εισόδου
2. Διαβάζουμε την πρώτη κωδική λέξη που ξέρουμε ότι είναι στο λεξικό, την αποκωδικοποιούμε και εισάγουμε το σύμβολο x στο διάνυσμα Ι.
3. Για κάθε επόμενη κωδική λέξη J που εισάγεται
   (α′) Την αποκωδικοποιούμε
   (β′) Συγχωνεύουμε το Ι με το πρώτο γράμμα x της αποκωδικοποιημένης J και εισάγουμε στο λεξικό την συμβολοσειρά Ιx(αν δεν υπάρχει ήδη)
   (γ′) Θέτουμε το $I = J$.

Παράδειγμα 5.15. *Να αποκωδικοποιήσετε την έξοδο του προηγούμενου παραδείγματος.*

Λύση

Θέση	Σύμβολο
0	␣
1	B
2	E
3	N
4	O
5	R
6	T

1. *Το πρώτο σύμβολο που εισέρχεται είναι το (6). Υπάρχει στον πίνακα άρα το αποκωδικοποιούμε στο T και θέτουμε I=T.*

2. *Το επόμενο σύμβολο που εισάγεται είναι το J =(4) που αντιστοιχεί στο 0, το TO δεν υπάρχει στο πίνακα άρα εισάγεται και θέτουμε I = O*

Θέση	Σύμβολο
0	␣
1	B
2	E
3	N
4	O
5	R
6	T
7	TO

$$\overset{\downarrow}{TO}\ BE\ OR\ NOT\ TO\ BE \rightarrow (6)$$

3. *Συνεχίζουμε με το J = (0), το οποίο αποκωδικοποιείται στο κενό σύμβολο, το I = O␣ δεν υπάρχει στο λεξικό οπότε την εισάγουμε και θέτουμε I = ␣*

Θέση	Σύμβολο
0	␣
1	B
2	E
3	N
4	O
5	R
6	T
7	TO
8	O␣

4. *J = (1) → J = B, εισάγουμε το I = ␣B στο λεξικό και θέτουμε I=B*

5. *J = (2) → J = E, εισάγουμε το I = BE στο λεξικό και θέτουμε I=E*

6. *J = (0) → J = ␣, εισάγουμε το I = ␣B στο λεξικό και θέτουμε I=␣*

7. *J = (4) → J = O, εισάγουμε το I = ␣O στο λεξικό και θέτουμε I=O*

8. *J = (5) → J = R, εισάγουμε το I = OR στο λεξικό και θέτουμε I=R*

9. $J = (0) \rightarrow J\text{␣}$, *εισάγουμε το* $I = R\text{␣}$ *στο λεξικό και θέτουμε* $I{=}R$

10. $J = (3) \rightarrow J = N$, *εισάγουμε το* $I = \text{␣}N$ *στο λεξικό και θέτουμε* $I{=}N$

11. $J = (4) \rightarrow J = O$, *εισάγουμε το* $I = NO$ *στο λεξικό και θέτουμε* $I{=}O$

12. $J = (6) \rightarrow J = T$, *εισάγουμε το* $I = OT$ *στο λεξικό και θέτουμε* $I{=}T$

13. $J = (0) \rightarrow J = \text{␣}$, *εισάγουμε το* $I = T\text{␣}$ *στο λεξικό και θέτουμε* $I{=}\text{␣}$

14. $J = (7) \rightarrow J = TO$, *εισάγουμε το* $I = \text{␣}T$ *στο λεξικό και θέτουμε* $I{=}TO$

15. $J = (9) \rightarrow J = \text{␣}B$, *εισάγουμε το* $I = TO\text{␣}$ *στο λεξικό και θέτουμε* $I{=}\text{␣}B$

16. $J = (2) \rightarrow J = E$, *εισάγουμε το* $I = \text{␣}BE$ *στο λεξικό και θέτουμε* $I{=}E$

    Επειδή δεν υπάρχει άλλη είσοδος σταματάμε στο $I = E$, *Παρατηρούμε ότι το λεξικό που δημιουργήθηκε είναι πανομοιότυπο με το λεξικό κωδικοποίησης.*

Θέση	*Σύμβολο*
0	␣
1	B
2	E
3	N
4	O
5	R
6	T
7	TO
8	0␣
9	␣B
10	BE

Θέση	*Σύμβολο*
11	E␣
12	␣O
13	OR
14	R␣
15	␣N
16	NO
17	OT
18	T␣
19	␣T
20	TO␣
21	␣BE

Η μόνη περίεργη περίπτωση εμφανίζεται όταν έχουμε συμβολοσειρές της μορφής $ABABABAB$ ή $KcKcKc$, όπου K συμβολοσειρά και c ένας χαρακτήρας. Πάμε να δούμε που κρύβεται η παγίδα σε αυτή την περίπτωση. Θα κωδικοποιήσουμε και θα αποκωδικοποιήσουμε την συμβολοσειρά $ABABABAB$. Ξεκινάμε με έναν πίνακα

Θέση	Σύμβολο
1	A
2	B

1. $ABABABAB \Rightarrow 1BABABAB$

Θέση	Σύμβολο
1	A
2	B
3	AB

2. $1BABABAB \Rightarrow 12ABABAB$

Θέση	Σύμβολο
1	A
2	B
3	AB
4	BA

3. $12ABABAB \Rightarrow 123ABAB$

Θέση	Σύμβολο
1	A
2	B
3	AB
4	BA
5	ABA

4. $123ABAB \Rightarrow 1235B$

Θέση	Σύμβολο
1	A
2	B
3	AB
4	BA
5	ABA
6	ABAB

5. $1235B \Rightarrow 12352$

Θέση	Σύμβολο
1	A
2	B
3	AB
4	BA
5	ABA
6	ABAB

Ας προσπαθήσουμε να αποκωδικοποιήσουμε την συγκεκριμένη συμβολοσειρά.

1. $12352 \Rightarrow A2352$

Θέση	Σύμβολο
1	A
2	B

2. $A2352 \Rightarrow AB352$

Θέση	Σύμβολο
1	A
2	B
3	AB

3. $AB352 \Rightarrow ABAB52$

Θέση	Σύμβολο
1	A
2	B
3	AB
4	BA

Παρατηρήστε τι γίνεται όταν φτάνουμε στην κωδική λέξη (5). Δεν έχει εισαχθεί ακόμη στον πίνακα άρα φαινομενικά δεν ξέρουμε πως να τη αποκωδικοποιήσουμε. Σκεφτόμαστε ότι για να μην υπάρχει στον πίνακα σημαίνει ότι μόλις παράχθηκε. Επίσης παρατηρούμε ότι κάθε καινούρια λέξη που εισάγεται στο λεξικό θα έχει ως πρόθεμα την μόλις προηγούμενη κωδική λέξη που έχει αποκωδικοποιηθεί και σαν κατάληξη το πρώτο γράμμα της καινούρια κωδικής λέξης. Άρα μπορεί να μην ξέρουμε ολόκληρη την κωδική λέξη 5 αλλά ξέρουμε την αρχή της που είναι η προηγούμενη κωδική λέξη που μόλις αποκωδικοποιήθηκε, δηλαδή η AB. Αφού όμως η κωδική λέξη (5) αρχίζει με AB αυτό αυτόματα σημαίνει ότι η κατάληξη της κωδικής λέξης (3) που είναι το πρώτο γράμμα της επόμενης είναι επί της ουσία το A. Σκεπτόμενοι με αυτόν τον τρόπο μπορούμε να παράγουμε την κωδική λέξη (5) που είναι η ABA και να συνεχίσουμε την αποκωδικοποίηση.

4. $ABAB52 \Rightarrow ABABABA2$

Θέση	Σύμβολο
1	A
2	B
3	AB
4	BA
5	ABA

5. $ABABABA2 \Rightarrow ABABABAB$

Θέση	Σύμβολο
1	A
2	B
3	AB
4	BA
5	ABA

Το μόνο που αξίζει να πούμε για την ανάλυση του LZW είναι πως επειδή παράγει μικρότερες κωδικές λέξεις από τον LZ78 έπεται ότι το μέσο μήκος του κώδικα που παράγει θα φράσσεται από το μέσο μήκος του κώδικα που παράγεται από τον LZ78, δηλαδή $\frac{1}{n}\overline{L}_{LZW} \leq \frac{1}{n}\overline{L}_{LZ78}$ Όμως στην προηγούμενη ενότητα αποδείξαμε ότι για μία στάσιμη και εργοδική πηγή με ρυθμός εντροπίας $H(\mathcal{X})$ έπεται ότι $\frac{1}{n}\overline{L}_{LZ78} \overset{n\to\infty}{\to} H(\mathcal{X}) \Rightarrow \frac{1}{n}\overline{L}_{LZW} \overset{n\to\infty}{\to} H(\mathcal{X})$

5.9 Ο μετασχηματισμός Burrows-Wheeler

Όπως έχουμε δει και σε αυτό το κεφάλαιο και στα προηγούμενα, όσο περισσότερο ασσύμετρη είναι μία κατανομή τόσο λιγότερη εντροπία έχει με αποτέλεσμα τα σύνολα δεδομένων που παράγονται με βάση αυτή να δίνουν καλύτερα ποσοστά συμπίεσης. Ας πάρουμε τις κατανομές $P_X^1(x) = \{Pr[X_1 = x_1] = \frac{1}{4},\ Pr[X_2 = x_2] = \frac{1}{4},\ Pr[X_3 = x_3] = \frac{1}{4},\ Pr[X_4 = x_4] = \frac{1}{4}\}$ και $P_X^2(x) = \{Pr[X_1 = x_1] = \frac{1}{2},\ Pr[X_2 = x_2] = \frac{1}{4},\ Pr[X_3 = x_3] = \frac{1}{8},\ Pr[X_4 = x_4] = \frac{1}{8}\}$, τότε $H(P^1) = 2$ και $H(P^2) = 1.75$ bits/σύμβολο. Καταλαβαίνουμε λοιπόν ότι μας συμφέρει να συμπιέζουμε σύνολα δεδομένων των οποίων τα στοιχεία δεν είναι ισοπίθανα, αλλά υπάρχει κάποια δομή που επάγει με τη σειρά της μία προβλεψιμότητα σε κάποιες ακολουθίες στοιχείων.

Για παράδειγμα, από μετρήσεις γνωρίζουμε ότι στην αγγλική γλώσσα το πιο συχνό εμφανιζόμενο γράμμα είναι το e, ενώ λιγότερα συχνά συναντάται το z. Οι μετρήσεις αυτές αποτελούν μέσους όρους και δεν είναι

πανάκεια για την συμπίεση καθώς μπορεί σε ένα σύνολο δεδομένων το γράμμα e να μην εμφανιστεί τόσο συχνά ή οι σχετικές συχνότητες των γραμμάτων αν υποθέσουμε ότι είναι ανεξάρτητα μεταξύ τους να μην δώσουν μία κατανομή που θα οδηγήσει σε καλή συμπίεση. Το σκηνικό μπορούμε να το αλλάξουμε αν δεν βλέπουμε το κάθε γράμμα μεμονωμένα αλλά σε σχέση με το περιεχόμενο ή τα συμφραζόμενα (context) που υπάρχουν γύρω του. Αν σκεφτούμε διάφορες λέξεις το μυαλό μας, θα παρατηρήσουμε ότι η πιθανότητα να συναντήσουμε το γράμμα h αυξάνεται αν έχει προηγηθεί τα γράμματα t ή s. Με την ίδια λογική είναι πολύ πιθανότερο να συναντήσουμε το γράμμα t να έχει προηγηθεί το περιεχόμενο tha, παρά το e και ας έχει μεγαλύτερη πιθανότητα από το t.

Καταλήγουμε στο συμπέρασμα πως είναι περισσότερο αποδοτικό να δημιουργούμε συμπιεστές που δεν βλέπουν το κείμενο σαν παράγωγο μία ακολουθίας ανεξάρτητων και ισόνομων τυχαίων μεταβλητών αλλά σαν το αποτέλεσμα μία μαρκοβιανής αλυσίδας k -τάξης. Η παραπάνω τεχνική συμπίεσης λέγεται συμπίεση με συμφραζόμενα και ξεχωριστή θέση σε αυτή κατέχει ο μετασχηματισμός Burrows-Wheeler.[25] Η βασική ιδέα πίσω από τον μετασχηματισμό στον ρόλο της ασύμμετρης κατανομής που εξηγήσαμε παραπάνω. Ο Wheeler είδε ότι πρίν και μετά από κάποια γράμματα ή ακολουθίες γραμμάτων ακολουθεί ένα περιορισμένο σύνολο συμβόλων, λόγου χάρη μετά την ακολουθία th συνήθως ακολουθεί το e ή a. Ενώ πριν από την ακολουθία he, συνήθως βρίσκεται το γράμμα t ή s. Παρατήρησε λοιπόν πως αν έχουμε μία ακολουθία γραμμάτων μήκους N και εφαρμόσουμε N-1 κυκλικές μεταθέσεις προς τα δεξιά, τότε:

1. Θα δημιουργηθεί ένας $N \times N$ πίνακας όπου κάθε στήλη του θα περιέχει όλα τα γράμματα που υπάρχουν στη συμβολοσειρά.

2. Αν ταξινομήσουμε λεξικογραφικά τις μεταθέσεις με βάση το πρώτο γράμμα τότε η τελευταία στήλη, που αποτελεί μία μετάθεση των συμβόλων της αρχικής ακολουθίας, θα περιέχει ίδια σύμβολα στην σειρά που μπορούν να συμπιεστούν πολύ πιο αποδοτικά από ότι η αρχική ακολουθία. Το αποτέλεσμα αυτό βασίζεται σε όσα αναλύσαμε παραπάνω, δηλαδή πριν και μετά από κάποιες ακολουθίες γραμμάτων είναι περισσότερο πιθανό να εμφανιστούν συγκεκριμένα γράμματα. Να σημειώσουμε ότι ο μετασχηματισμός δεν συμπιέζει την ακολουθία εισόδου αλλά την μετατρέπει σε μία συμπιέσιμη μορφή για να "αναλάβει την δουλειά" κάποια άλλη μέθοδος.

Διαδικασία Κωδικοποίησης

1. Δημιουργούμε $N-1$ μεταθέσεις προς τα δεξιά της ακολουθίας εισόδου μήκους N

2. Ταξινομούμε λεξικογραφικά τις μεταθέσεις με βάση το πρώτο γράμμα της κάθε μετάθεσης

3. Συμπιέζουμε την τελευταία στήλη του ταξινομημένου πίνακα αφού πρώτα της εφαρμόσουμε προς τα εμπρός κωδικοποίηση και στέλνουμε στον αποκωδικοποιητή την συμπιεσμένη συμβολοσειρά μαζί με ένα δείκτη I, που υποδεικνύει σε ποιά γραμμή του πίνακα βρίσκεται η αρχική ακολουθία εισόδου.

Είπαμε προηγουμένως ότι η διαδικασία Burrows-Wheeler μετατρέπει την αρχική ακολουθία σε μία συμπιέσιμη. Πριν προχωρήσουμε στη συμπίεση της τελευταίας στήλης την κωδικοποιούμε πρώτα αποδοτικά με μία μέθοδο που ονομάζεται **προς τα εμπρός κωδικοποίηση** (move to front coding).

5.9.1 Προς τα εμπρός κωδικοποίηση

Η μέθοδος της προς τα εμπρός κωδικοποίησης δέχεται ένα διάνυσμα συμβόλων και εξάγει ένα διάνυσμα αριθμών το οποίο συμπιέζεται αποδοτικά.

Διαδικασία κωδικοποίησης

1. Η διαδικασία κωδικοποίησης ξεκινάει με ένα διάνυσμα Υ που περιέχει το αλφάβητο της ακολουθίας εισόδου ταξινομημένο σε αλφαβητική σειρά και ένα κενό διάνυσμα R.

[25] **burrows1994block**.

2. Για κάθε σύμβολο i της τελευταίας στήλης στον ταξινομημένο πίνακα μεταθέσεων βλέπουμε την θέση του στο διάνυσμα αλφαβήτου Υ και την αποθηκεύουμε στο $R[i]$. Μετά μετακινούμε το σύμβολο στο διάνυσμα Y στην πρώτη θέση.

3. Η διαδικασία συνεχίζεται μέχρι να τελειώσουν τα σύμβολα της τελευταίας στήλης.

Το διάνυσμα R που προκύπτει συμπιέζεται με κάποια από τις μεθόδους που έχουμε ήδη μάθει, όπως Huffman ή αριθμητική κωδικοποίηση.

Παράδειγμα 5.16. *Να κωδικοποιήσετε την συμβολοσειρά baaaacbbccc σύμφωνα με την προς τα εμπρός κωδικοποίηση.*

1. *Το διάνυσμα του αλφαβήτου είναι $Y = [a,\ b,\ c]$. Οι θέσεις στο διάνυσμα ξεκινούν να μετράνε από το μηδέν. Το διάνυσμα R προς το παρόν είναι κενό.*

2. *Το πρώτο γράμμα είναι το b, $\underline{b}$aaaacbbccc, που αντιστοιχεί στο $Y[1]$, οπότε το πρώτο στοιχείο του R θα είναι το 1 γράφουμε $R = [1]$ και μετακινούμε το b στην πρώτη θέση του $Y = [b,\ a,\ c]$*

3. *Το δεύτερο γράμμα είναι το a, b$\underline{a}$aaacbbccc ,βρίσκεται στη θέση 1, άρα γράφουμε $R = [1,\ 1]$ και το μετακινούμε στην πρώτη θέση του $Y = [a,\ b,\ c]$.*

4. *Το τρίτο γράμμα είναι το a, ba$\underline{a}$aacbbccc ,βρίσκεται στη θέση 0, άρα γράφουμε $R = [1,\ 1,\ 0]$ και αφήνουμε το Υ αμετάβλητο, $Y = [a,\ b,\ c]$.*

5. *Το τέταρτο γράμμα είναι το a, baa$\underline{a}$acbbccc ,βρίσκεται στη θέση 0, άρα γράφουμε $R = [1,\ 1,\ 0,\ 0]$ και αφήνουμε το Υ αμετάβλητο, $Y = [a,\ b,\ c]$.*

6. *Το πέμπτο γράμμα είναι το a, baaa$\underline{a}$cbbccc ,βρίσκεται στη θέση 0, άρα γράφουμε $R = [1,\ 1,\ 0,\ 0,\ 0]$ και αφήνουμε το Υ αμετάβλητο, $Y = [a,\ b,\ c]$.*

7. *Το έκτο γράμμα είναι το c, baaaa$\underline{c}$bbccc ,βρίσκεται στη θέση 2, άρα γράφουμε $R = [1,\ 1,\ 0,\ 0,\ 0,\ 2]$ και το μετακινούμε στην πρώτη θέση του $Y = [c,\ a,\ b,]$.*

8. *Το έβδομο γράμμα είναι το b, baaaac$\underline{b}$bccc ,βρίσκεται στη θέση 2, άρα γράφουμε $R = [1,\ 1,\ 0,\ 0,\ 0,\ 2,\ 2]$ και το μετακινούμε στην πρώτη θέση του $Y = [b,\ c,\ a]$.*

9. *Το όγδοο γράμμα είναι το b, baaaacb$\underline{b}$ccc ,βρίσκεται στη θέση 0, άρα γράφουμε $R = [1,\ 1,\ 0,\ 0,\ 0,\ 2,\ 2,\ 0]$ και αφήνουμε αμετάβλητο το $Y = [b,\ c,\ a]$*

10. *Το ένατο γράμμα είναι το c, baaaacbb$\underline{c}$cc ,βρίσκεται στη θέση 1, άρα γράφουμε $R = [1,\ 1,\ 0,\ 0,\ 0,\ 2,\ 2,\ 0,\ 1]$ και το μετακινούμε στην πρώτη θέση του $Y = [c,\ b,\ a]$.*

11. *Το δέκατο γράμμα είναι το c, baaaacbbc$\underline{c}$c ,βρίσκεται στη θέση 0, άρα γράφουμε $R = [1,\ 1,\ 0,\ 0,\ 0,\ 2,\ 2,\ 0,\ 1,\ 0]$ και αφήνουμε αμετάβλητο το $Y = [c,\ b,\ a]$.*

12. *Το ενδέκατο γράμμα είναι το c, baaaacbbcc$\underline{c}$,βρίσκεται στη θέση 0, άρα γράφουμε $R = [1,\ 1,\ 0,\ 0,\ 0,\ 2,\ 2,\ 0,\ 1,$ και αφήνουμε αμετάβλητο το $Y = [c,\ b,\ a]$.*

    Άρα η κωδικοποιημένη μορφή του baaaacbbccc είναι το διάνυσμα $R = [1,\ 1,\ 0,\ 0,\ 0,\ 2,\ 2,\ 0,\ 1,\ 0,\ 0]$ που το συμπιέζουμε με όποιο τρόπο θέλουμε.

Η αποκωδικοποίηση της προς τα εμπρός κωδικοποίησης υποθέτει ότι ο αποκωδικοποιητής γνωρίζει το διάνυσμα αλφαβήτου Υ και ακολουθεί τα ίδια βήματα με τον κωδικοποιητή.

Διαδικασία Αποκωδικοποίησης

1. Η διαδικασία ξεκινάει με το διάνυσμα αλφαβήτου Y και το αποσυμπιεσμένο διάνυσμα R και ένα κενό διάνυσμα συμβόλων D.

2. Αρχικά διαβάζουμε το πρώτο στοιχείο του R το οποίο αποτελεί τη θέση του πρώτου σύμβολου στο Y. Αν $R[0] = k$ ο δείκτης του πρώτο συμβόλου, τότε πηγαίνωντας στην θέση k στο Υ, βρίσκουμε το πρώτο σύμβολο, το αποθηκεύουμε στην πρώτη θέση του διανύσματος D και μετακινούμε το αποκωδικοποιημένο σύμβολο στην αρχή του Υ

3. Στην συνέχεια για το $i-$οστό στοιχείο του R βρίσκουμε σε ποιο σύμβολο αντιστοιχεί στο διάνυσμα Y, το αποθηκεύουμε στο διάνυσμα D στη θέση το i και μετακινούμε το $Y[i]$ στην αρχή του διανύσματος Y.

4. Η διαδικασία συνεχίζεται μέχρι να τελειώσει το διάνυσμα R.

Παράδειγμα 5.17. *Να αποκωδικοποιήσετε το διάνυσμα R που βρήκατε στο προηγούμενο παράδειγμα*

1. *Ξεκινάμε με το $Y = [a,\ b,\ c]$, το $R = [1,\ 1,\ 0,\ 0,\ 0,\ 2,\ 2,\ 0,\ 1,\ 0,\ 0]$ και ένα κενό διάνυσμα D.*

2. *Το πρώτο στοιχείο του R είναι το 1,* $\underline{1}1000220100$*, που αντιστοιχεί στο σύμβολο $Y[1] = b$. Αποθηκεύουμε το b στο διάνυσμα $D = [b]$ και το μετακινούμε στην αρχή του $Y = [b,\ a,\ c]$.*

3. *Το δεύτερο στοιχείο του R είναι το 1,* $1\underline{1}000220100$*, που αντιστοιχεί στο σύμβολο $Y[1] = a$. Αποθηκεύουμε το b στο διάνυσμα $D = [b,\ a]$ και το μετακινούμε στην αρχή του $Y = [a,\ b,\ c]$.*

4. *Το τρίτο στοιχείο του R είναι το 0,* $11\underline{0}00220100$*, που αντιστοιχεί στο σύμβολο $Y[0] = a$. Αποθηκεύουμε το a στο διάνυσμα $D = [b,\ a,\ a]$ και αφήνουμε αμετάβλητο το $Y = [a,\ b,\ c]$.*

5. *Το τέταρτο στοιχείο του R είναι το 0,* $110\underline{0}0220100$*, που αντιστοιχεί στο σύμβολο $Y[0] = a$. Αποθηκεύουμε το a στο διάνυσμα $D = [b,\ a,\ a,\ a]$ και αφήνουμε αμετάβλητο το $Y = [a,\ b,\ c]$.*

6. *Το πέμπτο στοιχείο του R είναι το 0,* $1100\underline{0}220100$*, που αντιστοιχεί στο σύμβολο $Y[0] = a$. Αποθηκεύουμε το a στο διάνυσμα $D = [b,\ a,\ a,\ a,\ a]$ και αφήνουμε αμετάβλητο το $Y = [a,\ b,\ c]$.*

7. *Το έκτο στοιχείο του R είναι το 2,* $11000\underline{2}20100$*, που αντιστοιχεί στο σύμβολο $Y[2] = c$. Αποθηκεύουμε το c στο διάνυσμα $D = [b,\ a,\ a,\ a,\ a,\ c]$ και το μετακινούμε στην αρχή του $Y = [c,\ a,\ b]$.*

8. *Το έβδομο στοιχείο του R είναι το 2,* $11000\underline{2}0100$*, που αντιστοιχεί στο σύμβολο $Y[2] = b$. Αποθηκεύουμε το b στο διάνυσμα $D = [b,\ a,\ a,\ a,\ a,\ c,\ b]$ και το μετακινούμε στην αρχή του $Y = [b,\ c,\ a,]$.*

9. *Το όγδοο στοιχείο του R είναι το ,* $11000220\underline{1}00$*, που αντιστοιχεί στο σύμβολο $Y[0] = b$. Αποθηκεύουμε το b στο διάνυσμα $D = [b,\ a,\ a,\ a,\ a,\ c,\ b,\ b]$ και αφήνουμε αμετάβλητο το $Y = [b,\ c,\ a,]$.*

10. *Το ένατο στοιχείο του R είναι το ,* $11000220\underline{1}00$*, που αντιστοιχεί στο σύμβολο $Y[1] = c$. Αποθηκεύουμε το b στο διάνυσμα $D = [b,\ a,\ a,\ a,\ a,\ c,\ b,\ b,\ c]$ και το μετακινούμε στην αρχή του$Y = [c,\ b,\ a,]$.*

11. *Το δέκατο στοιχείο του R είναι το ,* $110002201\underline{0}0$*, που αντιστοιχεί στο σύμβολο $Y[0] = c$. Αποθηκεύουμε το c στο διάνυσμα $D = [b,\ a,\ a,\ a,\ a,\ c,\ b,\ b,\ c,\ c]$ και αφήνουμε αμετάβλητο το $Y = [c,\ b,\ a,]$.*

12. *Το ενδέκατο στοιχείο του R είναι το ,* $110002201\underline{0}0$*, που αντιστοιχεί στο σύμβολο $Y[0] = c$. Αποθηκεύουμε το c στο διάνυσμα $D = [b,\ a,\ a,\ a,\ a,\ c,\ b,\ b,\ c,\ c,\ c]$ και αφήνουμε αμετάβλητο το $Y = [c,\ b,\ a,]$.*

Παράδειγμα 5.18. *Να κωδικοποιήσετε με τον μετασχηματισμό Burrows-Wheeler την συμβολοσειρά:*
“”

1. *Στο πρώτο βήμα δημιουργούμε όλες τις μεταθέσεις*

$$
\begin{array}{c}
abracadabra \\
aabracadabr \\
raabracadab \\
braabracada \\
abraabracad \\
dabraabraca \\
adabraabrac \\
cadabraabra \\
acadabraabr \\
racadabraab \\
bracadabraa
\end{array}
$$

2. *Ταξινομούμε λεξικογραφικά τον παραπάνω* 11×11 *πίνακα και προκύπτει ένας ταξινομημένος πίνακας μεταθέσεων.*

$$
\begin{array}{c}
aabracadabr \\
abraabracad \\
abracadabra \\
acadabraabr \\
adabraabrac \\
braabracada \\
bracadabraa \\
cadabraabra \\
dabraabraca \\
raabracadab \\
racadabraab
\end{array}
$$

3. *Αποθηκεύουμε την τελευταία στήλη σε ένα διάνυσμα μήκους 11 καθώς και την γραμμή που βρίσκεται η αρχική ακολουθία:* $([r,\ d,\ a,\ r,\ c,\ a,\ a,\ a,\ a,\ b,\ b], I = 2)$.

4. *Εφαρμόζουμε προς τα εμπρός κωδικοποίηση στην ακολουθία συμβόλων*

   (α′) *Αρχίζουμε με το αλφάβητο* $Y = [a,\ b,\ c,\ d,\ r]$, *την ακολουθία των συμβόλων* $rdarcaaaabb$ *και ένα κενό διάνυσμα* R.

   (β′) *Το πρώτο γράμμα είναι το* r,$\underline{r}darcaaaabb$, *που αντιστοιχεί στο* $Y[4] = r$, *οπότε αποθηκεύουμε τον δείκτη στο* $R = [4]$ *και μετακινούμε το σύμβολο* r *στην αρχή του* $Y = [r,\ a,\ b,\ c,\ d]$

   (γ′) *Το δεύτερο γράμμα είναι το* d ,$r\underline{d}arcaaaabb$, *που αντιστοιχεί στο* $Y[4] = d$, *οπότε αποθηκεύουμε τον δείκτη στο* $R = [4,\ 4]$ *και μετακινούμε το σύμβολο* r *στην αρχή του* $Y = [d,\ r,\ a,\ b,\ c]$

   (δ′) *Το τρίτο γράμμα είναι το* a ,$rd\underline{a}rcaaaabb$, *που αντιστοιχεί στο* $Y[2] = a$, *οπότε αποθηκεύουμε τον δείκτη στο* $R = [4,\ 4,\ 2]$ *και μετακινούμε το σύμβολο* a *στην αρχή του* $Y = [a,\ d,\ r,\ b,\ c]$

   (ε′) *Το τέταρτο γράμμα είναι το* r ,$rda\underline{r}caaaabb$, *που αντιστοιχεί στο* $Y[2] = r$, *οπότε αποθηκεύουμε τον δείκτη στο* $R = [4,\ 4,\ 2,\ 2]$ *και μετακινούμε το σύμβολο* r *στην αρχή του* $Y = [r,\ a,\ d,\ b,\ c]$

   (ς′) *Το πέμπτο γράμμα είναι το* r ,$rdar\underline{c}aaaabb$, *που αντιστοιχεί στο* $Y[4] = c$, *οπότε αποθηκεύουμε τον δείκτη στο* $R = [4,\ 4,\ 2,\ 2,\ 4]$ *και μετακινούμε το σύμβολο* c *στην αρχή του* $Y = [c,\ r,\ a,\ d,\ b]$

   (ζ′) *Το έκτο γράμμα είναι το* a ,$rdarc\underline{a}aaabb$, *που αντιστοιχεί στο* $Y[2] = a$, *οπότε αποθηκεύουμε τον δείκτη στο* $R = [4,\ 4,\ 2,\ 2,\ 4,\ 2]$ *και μετακινούμε το σύμβολο* a *στην αρχή του* $Y = [a,\ c,\ r,\ d,\ b]$

(η′) *Το έβδομο γράμμα είναι το a ,rdarca$\underline{a}$aabb, που αντιστοιχεί στο* $Y[0] = a$*, οπότε αποθηκεύουμε τον δείκτη στο* $R = [4,\ 4,\ 2,\ 2,\ 4,\ 2,\ 0]$ *και αφήνουμε αμετάβλητο το* $Y = [a,\ c,\ r,\ d,\ b]$

(θ′) *Το όγδοο γράμμα είναι το a ,rdarcaa$\underline{a}$abb, που αντιστοιχεί στο* $Y[0] = a$*, οπότε αποθηκεύουμε τον δείκτη στο* $R = [4,\ 4,\ 2,\ 2,\ 4,\ 2,\ 0,\ 0]$ *και αφήνουμε αμετάβλητο το* $Y = [a,\ c,\ r,\ d,\ b]$

(ι′) *Το ένατο γράμμα είναι το a ,rdarcaaa$\underline{a}$bb, που αντιστοιχεί στο* $Y[0] = a$*, οπότε αποθηκεύουμε τον δείκτη στο* $R = [4,\ 4,\ 2,\ 2,\ 4,\ 2,\ 0,\ 0,\ 0]$ *και αφήνουμε αμετάβλητο το* $Y = [a,\ c,\ r,\ d,\ b]$

(ια′) *Το δέκατο γράμμα είναι το b ,rdarcaaaa$\underline{b}$b, που αντιστοιχεί στο* $Y[4] = a$*, οπότε αποθηκεύουμε τον δείκτη στο* $R = [4,\ 4,\ 2,\ 2,\ 4,\ 2,\ 0,\ 0,\ 0,\ 4]$ *και μετακινούμε το σύμβολο b στην αρχή του* $Y = [b,\ a,\ c,\ r,\ d,]$

(ιβ′) *Το ενδέκατο γράμμα είναι το b ,rdarcaaaab$\underline{b}$, που αντιστοιχεί στο* $Y[0] = a$*, οπότε αποθηκεύουμε τον δείκτη στο* $R = [4,\ 4,\ 2,\ 2,\ 4,\ 2,\ 0,\ 0,\ 0,\ 4,\ 0]$ *και αφήνουμε αμετάβλητο το* $Y = [b,\ a,\ c,\ r,\ d,]$

.

5. *Συμπιέζουμε την ζεύγος* $([4,\ 4,\ 2,\ 2,\ 4,\ 2,\ 0,\ 0,\ 0,\ 4,\ 0],\ I = 2)$ *με μία μέθοδο της επιλογής μας και το αποστέλλουμε στον αποκωδικοποιητή.*

Το πιο ενδιαφέρον σημείο του μετασχηματισμού Burrows-Wheeler είναι η διαδικασία αποκωδικοποίησης. Φαντάζει δύσκολο να σκεφτούμε πως από την τελευταία στήλη και ένα δείκτη θα μπορέσουμε να ανακατασκευάσουμε ολόκληρη την ακολουθία. Η μαγεία της συγκεκριμένης κωδικοποίησης κρύβεται στην συσχέτιση που έχουν μεταξύ τους η πρώτη και η τελευταία στήλη. Ας σκεφτούμε τι γίνεται για παράδειγμα κατά τις πρώτες μετάθεσεις προς τα δεξιά της αρχικής λέξης abracadabra. Στην αρχική ακολουθία το πρώτο σύμβολο αντιστοιχεί ταυτοτικά στο πρώτο γράμμα της λέξης και το τελευταίο σύμβολο αντίστοιχα στο τελευταίο γράμμα της λέξης. Μετά την πρώτη μετάθεση η λέξη abracadabra γίνεται aabracadabr και το πρώτο σύμβολο της μετάθεσης αντιστοιχεί στην τελευταίο γράμμα της αρχικής λέξης ενώ το τελευταίο σύμβολο στο προτελευταίο γράμμα. Μετά από μία ακόμη μετάθεση raabracadab, το πρώτο γράμμα της μετάθεσης αντιστοιχεί στο προτελευταίο της αρχικής λέξης και το τελευταίο γράμμα της μετάθεσης αντιστοιχεί στο τρίτο από το τέλος γράμμα της αρχικής λέξης. Η διαδικασία συνεχίζεται με την ίδια λογική μέχρι την $N-1$ μετάθεση που το πρώτο γράμμα της τελευταίας μετάθεσης αντιστοιχεί στο δεύτερο της αρχική λέξης και το τελευταίο σύμβολο στο πρώτο γράμμα της αρχικής λέξης.

Καταλαβαίνουμε λοιπόν πως αν αποθηκεύσουμε σε δύο διανύσματα την πρώτη και την τελευταία λέξη των γραμμάτων που προκύπτουν από τις μεταθέσεις μπορούμε να ανακατασκευάσουμε ολόκληρη τη λέξη ως εξής: Αν πρώτη στήλη είναι η $F = [a,\ a,\ r,\ b,\ a,\ d,\ a,\ c,\ a,\ r,\ b]$ και τελευταία η
$L = [a,\ r,\ b,\ a,\ d,\ a,\ c,\ a,\ r,\ b,\ a]$, τότε ξέρουμε ότι το $L[0] = a$ αποτελεί το τελευταίο στοιχείο της λέξης. Το γράμμα που βρίσκεται πριν από το α, θα είναι στη γραμμή που έχει σαν αρχικό γράμμα το α, δηλαδή στην γραμμή δύο. Το τελευταίο γράμμα της γραμμής 2 είναι το r οπότε η λέξη που έχει σχηματιστεί μέχρι στιγμής είναι η ra. Το γράμμα πριν το r θα βρίσκετε στην γραμμή που έχει σαν αρχικό γράμμα το r, δηλαδή την 3. Το $L[3] = b$ και η λέξη που έχει ανακατασκευαστεί μέχρι στιγμή η bra. Γενικά αν διαβάσουμε το διάνυσμα L από το τέλος προς την αρχή θα δούμε ότι μάς δίνει την λέξη $abracadabra$. Τα πράγματα είναι εύκολα πριν ανακατευτούν οι μεταθέσεις προκειμένου να προκύψει ένας λεξικογραφικά ταξινομημένος πίνακας.

Τι γίνεται όμως μετά; Σίγουρα το πρώτο βήμα είναι εύκολο, αρχίζουμε από τον δείκτη που μεταδόθηκε μαζί με την συμπιεσμένη τελευταία στήλη, ας την πούμε L. Ξέρουμε λοιπόν ότι το $L[I]$ θα αντιστοιχεί στο τελευταίο σύμβολο της λέξης $abracadabra$. Για να βρούμε το προτελευταίο θα πρέπει να μεταβούμε στη γραμμή που έχει σαν πρώτο σύμβολο το $L[I]$ και να βρούμε το τελευταίο της σύμβολο $L[L[I]]$. Το πρώτο πρόβλημα στο οποίο σκοντάφτουμε είναι το εξής: το $L[I] = a$ και υπάρχουν στον ταξινομημένο πίνακα μεταθέσεων τέσσερις γραμμές, αν εξαιρέσουμε την I, που αρχίζουν με a. Σε ποια από όλες αντιστοιχίζεται λοιπόν το L[I]. Για να ξεπεράσουμε το παραπάνω πρόβλημα θα πρέπει με κάποιο τρόπο να θυμόμαστε την διάταξη των ίδιων γραμμάτων μέσα στη λέξη. Έτσι αντί να δημιουργήσουμε μεταθέσεις στην ακολουθία $abracadabra$ μπορούμε να χρησιμοποιήσουμε τη λέξη $a_0b_0r_0a_1ca_2da_3b_1r_1a_4$, όπου έχουμε δώσει ένα αύξοντα αριθμό στα όμοια γράμματα. Τότε οι μεταθέσεις έχουν ώς:

$$a_0b_0r_0a_1ca_2da_3b_1ra_4$$
$$a_4a_0b_0r_0a_1ca_2da_3b_1r_1$$
$$r_1a_4a_0b_0r_0a_1ca_2da_3b_1$$
$$b_1ra_4a_0b_0r_0a_1ca_2da_3$$
$$a_3b_1ra_4a_0b_0r_0a_1ca_2d$$
$$da_3b_1ra_4a_0b_0r_0a_1ca_2$$
$$a_2da_3b_1ra_4a_0b_0r_0a_1c$$
$$ca_2da_3b_1ra_4a_0b_0r_0a_1$$
$$a_1ca_2da_3b_1ra_4a_0b_0r_0$$
$$r_0a_1ca_2da_3b_1ra_4a_0b_0$$
$$b_0r_0a_1ca_2da_3b_1ra_4a_0$$

Ταξινομώντας τις παραπάνω μεταθέσεις παίρνουμε τον ταξινομημένο πίνακα:

$$a_4a_0b_0r_0a_1ca_2da_3b_1r_1$$
$$a_3b_1ra_4a_0b_0r_0a_1ca_2d$$
$$a_0b_0r_0a_1ca_2da_3b_1ra_4$$
$$a_1ca_2da_3b_1ra_4a_0b_0r_0$$
$$a_2da_3b_1ra_4a_0b_0r_0a_1c$$
$$b_1ra_4a_0b_0r_0a_1ca_2da_3$$
$$b_0r_0a_1ca_2da_3b_1ra_4a_0$$
$$ca_2da_3b_1ra_4a_0b_0r_0a_1$$
$$da_3b_1ra_4a_0b_0r_0a_1ca_2$$
$$r_1a_4a_0b_0r_0a_1ca_2da_3b_1$$
$$r_0a_1ca_2da_3b_1ra_4a_0b_0$$

Η σημαντικότερη παρατήρηση που πρέπει να γίνει στο σημείο αυτό είναι πως η σειρά με την οποία εμφανίζονται τα σύμβολα στην τελευταία στήλη διατηρείται και στην πρώτη. Για παράδειγμα η σειρά με την οποία εμφανίζονται τα α στην τελευταία στήλη ($a_4,\ a_3,\ a_0,\ a_1,\ a_2$) είναι ίδια με την σειρά που συναντώνται στην πρώτη. Αυτό γίνεται γιατί τα σύμβολα και των δύο στηλών έχουν μία ταξινόμηση που συμβαίνει σύμφωνα με τα γράμματα που έπονται (δεξιό περιεχόμενο). Για παράδειγμα μετά το γράμμα a_4 έπεται το a_0 ή το σύμβολο a_3 ακολουθείται από το σύμβολο b_0. Έτσι στην πρώτη στήλη το τρίτο α είναι το a_0 που ταξινομήθηκε με βάση το b_0 που βρίσκεται δεξιά του και το a_1 με βάση το c. Επειδή το b_0 λεξικογραφικά προηγείται του c τότε και το σύμβολο που βρίσκεται πριν το b_0 θα προηγείται από το σύμβολο που βρίσκεται πριν από το c. Τα σύμβολα αυτά δεν είναι άλλα από τα $a_0,\ a_1$ που βρίσκονται στην τελευταία στήλη στις γραμμές που έχουν ως πρώτο γράμμα το b_0 και c αντίστοιχα. Συμπερασματικά επειδή τα όμοια γράμματα διατάσσονται με βάση το περιεχόμενο που βρίσκεται προς τα δεξιά του έπεται ότι και τα γράμματα της τελευταίας στήλης που αποτελούν τα σύμβολα πριν το δεξιό περιεχόμενο θα υπακούν στην ίδια διάταξη.

Διαδικασία αποκωδικοποίησης

1. Η αποκωδικοποίηση ξεκινάει με την τελευταία στήλη L και τον δείκτη I. Ταξινομούμε λεξικογραφικά τα σύμβολα της στήλης L σεβόμενοι την διάταξη των όμοιων συμβόλων. Αποθηκεύουμε την αντιστοιχία μεταξύ των συμβόλων της πρώτης και τελευταίας στήλης σε ένα διάνυσμα T.

2. Ανακτούμε την λέξη από το τέλος προς την αρχή ως εξής:

   (α′) Αν το σύμβολο $L[I]$ αποτελεί το τελευταίο γράμμα της αρχικής λέξης, τότε το τελευταίο σύμβολο της γραμμής που έχει ως πρώτο στοιχείο το $L[I]$ θα αποτελεί το προτελευταίο γράμμα της λέξης. Βρίσκουμε μέσω του μετασχηματισμού T ποια γραμμή αντιστοιχεί στο $L[I]$ και έχουμε ότι το ζητούμενο γράμμα θα είναι το $L[T[I]]$. Με την ίδια λογική μέσω του T βρίσκουμε την γραμμή που ξεκινάει με τον προτελευταίο γράμμα θα είναι η T[T[I]], άρα το δεύτερο γράμμα πριν το τέλος θα είναι το $L[T[T[I]]]$. Γενικά το $i-$οστό γράμμα πριν το τέλος θα είναι το $L[T^i[I]]$ με $T^0[I] = I$ και $T^{i+1}[I] = T[T^i[I]]$.

Παράδειγμα 5.19. *Να αποκωδικοποιήσετε το διάνυσμα R του προηγούμενο παραδείγματος και να αντιστρέψετε τον μετασχηματισμό Burrows-Wheeler*

1. *Αρχικά πρέπει να αντιστρέψουμε τη προς τα εμπρός κωδικοποίηση. Η όλη διαδικασία ξεκινάει με το διάνυσμα $R = [4,\ 4,\ 2,\ 2,\ 4,\ 2,\ 0,\ 0,\ 0,\ 4,\ 0]$ και το αλφάβητο $Y = [a,\ b,\ c, d,\ r]$*

   (α′) *Το πρώτο στοιχείο του R είναι το 4,* $\underline{4}4224200040$*, που αντιστοιχεί στο σύμβολο $Y[4] = r$. Το αποθηκεύουμε στο διάνυσμα $L = [r]$ και το μετακινούμε στην αρχή του $Y = [r,\ a,\ b,\ c, d,]$*

   (β′) *Το δεύτερο στοιχείο του R είναι το 4 ,*$4\underline{4}224200040$*, που αντιστοιχεί στο σύμβολο $Y[4] = d$. Το αποθηκεύουμε στο διάνυσμα $L = [r,\ d]$ και το μετακινούμε στην αρχή του $Y = [d,\ r,\ a,\ b,\ c]$*

   (γ′) *Το τρίτο στοιχείο του R είναι το 2,* $44\underline{2}24200040$*, που αντιστοιχεί στο σύμβολο $Y[2] = a$. Το αποθηκεύουμε στο διάνυσμα $L = [r,\ d,\ a]$ και το μετακινούμε στην αρχή του $Y = [a,\ d,\ r,\ b,\ c]$*

   (δ′) *Το τέταρτο στοιχείο του R είναι το 2,* $442\underline{2}4200040$*, που αντιστοιχεί στο σύμβολο $Y[2] = r$. Το αποθηκεύουμε στο διάνυσμα $L = [r,\ d,\ a,\ r]$ και το μετακινούμε στην αρχή του $Y = [r,\ a,\ d,\ b,\ c]$*

   (ε′) *Το πέμπτο στοιχείο του R είναι το 4,* $4422\underline{4}200040$*, που αντιστοιχεί στο σύμβολο $Y[4] = c$. Το αποθηκεύουμε στο διάνυσμα $L = [r,\ d,\ a,\ r,\ c]$ και το μετακινούμε στην αρχή του $Y = [c,\ r,\ a,\ d,\ b]$*

   (ϛ′) *Το έκτο στοιχείο του R είναι το 2,* $44224\underline{2}00040$*, που αντιστοιχεί στο σύμβολο $Y[2] = a$. Το αποθηκεύουμε στο διάνυσμα $L = [r,\ d,\ a,\ r,\ c,\ a]$ και το μετακινούμε στην αρχή του $Y = [a,\ c,\ r,\ d,\ b]$*

   (ζ′) *Το έβδομο στοιχείο του R είναι το 0,* $44224\underline{2}0040$*, που αντιστοιχεί στο σύμβολο $Y[0] = a$. Το αποθηκεύουμε στο διάνυσμα $L = [r,\ d,\ a,\ r,\ c,\ a,\ a]$ και αφήνουμε αμετάβλητο το $Y = [a,\ c,\ r,\ d,\ b]$*

   (η′) *Το όγδοο στοιχείο του R είναι το 0,* $44224\underline{2}0040$*, που αντιστοιχεί στο σύμβολο $Y[0] = a$. Το αποθηκεύουμε στο διάνυσμα $L = [r,\ d,\ a,\ r,\ c,\ a,\ a,\ a]$ και αφήνουμε αμετάβλητο το $Y = [a,\ c,\ r,\ d,\ b]$*

   (θ′) *Το ένατο στοιχείο του R είναι το 0,* $44224200\underline{0}40$*, που αντιστοιχεί στο σύμβολο $Y[0] = a$. Το αποθηκεύουμε στο διάνυσμα $L = [r,\ d,\ a,\ r,\ c,\ a,\ a,\ a,\ a]$ και αφήνουμε αμετάβλητο το $Y = [a,\ c,\ r,\ d,\ b]$*

   (ι′) *Το ένατο στοιχείο του R είναι το 0,* $44224200\underline{4}0$*, που αντιστοιχεί στο σύμβολο $Y[4] = b$. Το αποθηκεύουμε στο διάνυσμα $L = [r,\ d,\ a,\ r,\ c,\ a,\ a,\ a,\ a,\ b]$ και το μετακινούμε στην αρχή του $Y = [b,\ a,\ c,\ r,\ d]$*

   (ια′) *Το ενδέκατο στοιχείο του R είναι το 0,* $44224200\underline{4}0$*, που αντιστοιχεί στο σύμβολο $Y[0] = b$. Το αποθηκεύουμε στο διάνυσμα $L = [r,\ d,\ a,\ r,\ c,\ a,\ a,\ a,\ a,\ b, b]$ και αφήνουμε αμετάβλητο το $Y = [b,\ a,\ c,\ r,\ d]$*

2. *Το διάνυσμα που αποκωδικοποιήσαμε δεν έχει κάποια διάταξη ως προς τα όμοια γράμματα. Αυτό δεν πρέπει να μας προβληματίζει για θα μπορούσαμε πολύ απλά αντί για το ζεύγος:*
   ($R = [4,\ 4,\ 2,\ 2,\ 4,\ 2,\ 0,\ 0,\ 0,\ 4,\ 0]$, $I = 2$) να στείλουμε την συμπιεσμένη τριπλέτα:
   ($R = [4,\ 4,\ 2,\ 2,\ 4,\ 2,\ 0,\ 0,\ 0,\ 4,\ 0], Order = [1, \llcorner,\ 4,\ 0, \llcorner,\ 3,\ 0,\ 1,\ 2\ ,1,\ 0]$ $I = 3$) και έτσι να έχουμε και την τάξη των όμοιων συμβόλων διαθέσιμη.

3. *Αφού λοιπόν αποκωδικοποιήσαμε το R σε $L = r_1 d a_4 r_0 c a_3 a_0 a_1 a_2 b_1 b_0$ μπορούμε να ταξινομήσουμε τα παραπάνω σύμβολα σεβόμενοι την διάταξη ώστε να πάρουμε την πρώτη γραμμή. Αποθηκεύουμε την μετάθεση των συμβόλων που δίνουν την ταξινόμηση σε ένα διάνυσμα T.*

   Άρα έχουμε $L = r_1 d a_4 r_0 c a_3 a_0 a_1 a_2 b_1 b_0 \to a_4 a_3 a_0 a_1 a_2 b_1 b_0 c d r_1 r_0$ με:
   $T = \left\{ \begin{array}{cccccccccccc} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \\ 9 & 8 & 0 & 10 & 7 & 1 & 2 & 3 & 4 & 5 & 6 \end{array} \right\}$.

4. *Με βάση τον μετασχηματισμό T και τον δείκτη I ξεκινάμε την ανάκτηση της αρχικής ακολουθίας από το τέλος προς την αρχή.*

(α') $L[T^0[I]] = L[I] = L[2] = a_4$

(β') $L[T^1[I]] = L[T[2]] = L[0] = r_1$

(γ') $L[T^2[I]] = L[T[0]] = L[9] = b_1$

(δ') $L[T^3[I]] = L[T[9]] = L[5] = a_3$

(ϵ') $L[T^4[I]] = L[T[5]] = L[1] = d$

(ς') $L[T^5[I]] = L[T[1]] = L[8] = a_2$

(ζ') $L[T^6[I]] = L[T[8]] = L[4] = c$

(η') $L[T^7[I]] = L[T[4]] = L[7] = a_1$

(ϑ') $L[T^8[I]] = L[T[7]] = L[3] = r_0$

(ι') $L[T^9[I]] = L[T[3]] = L[10] = b_0$

($\iota\alpha'$) $L[T^10[I]] = L[T[10]] = L[6] = a_0$

5.9.2 Η ανάλυση του μετασχηματισμού Burrows-Wheeler

Η ανάλυση του μετασχηματισμού Burrows-Wheeler θα βασιστεί στην δουλειά τών Kaplan, Landau και Verbin[26]. Συγκεκριμένα θα ακολουθήσουμε την λογική της δημοσίευσης A simpler analysis of Burrows-Wheeler based information για να εξάγουμε ένα άνω φράγμα για τον μετασχηματισμό. Υπενθυμίζουμε ότι η συμπίεση με βάση τον Burrows-Wheeler αρχικά μετασχηματίζει μία συμβολοσειρά s σε μία $\hat{s} = BWT(s)$. Στην συνέχει η $\hat{s}$ μετατρέπεται σε μία ακολουθία ακεραίων με την προς τα εμπρός κωδικοποίηση και τέλος η ακολουθία που παράχθηκε συμπιέζεται με την βοήθεια ενός συμπιεστή τάξης 0 (OrderO). Με την έννοια "συμπιεστής τάξης μηδέν" εννοούμε οποιοδήποτε συμπιεστή δημιουργεί κώδικες για τα σύμβολα ενός αλφαβήτου Σ και κατ' επέκταση των συμβολοσειρών που παράγονται από αυτό βασιζόμενος μόνο στην συχνότητα των συμβόλων αυτών καθεαυτών και όχι στα σύμβολα που προηγήθηκαν ή ακολούθησαν μετά από αυτά. Παραδείγματα τέτοιων συμπιεστών είναι η κωδικοποίηση Huffman, Shannon καθώς και η αριθμητική κωδικοποίηση για τις τιμές μια τυχαίας μεταβλητής.

Ο Burrows-Wheeler ανήκει στην κατηγορία τεχνικών συμπίεσης που χρησιμοποιούν το περιεχόμενο των συμβόλων που συμπιέζουν ώστε να πετύχουν καλύτερα αποτελέσματα. Ως περιεχόμενο (content) αναφέρονται τα σύμβολα που προηγήθηκαν η ακολούθησαν το σύμβολο προς συμπίεση. Τα φράγματα σε τέτοιους είδους συμπιεστές είναι λογικό να βασίζονται στην εντροπία k-τάξης που συναντήσαμε κατά την θεωρητική ανάλυση της συμπίεσης στο κεφάλαιο 3. Βέβαια επειδή κατά την μετάβαση από τη θεωρία στη πράξη πολλές από τις υποθέσεις που κάνουμε δεν ισχύουν, είναι λογικό και αναγκαίο να ορίσουμε πρακτικά μέτρα που σχετίζονται με την θεωρία αλλά επιβεβαιώνονται και από πειραματικά δεδομένα. Ένα τέτοιο μέτρο είναι η δειγματική εντροπία:

Ορισμός 5.4. *Ως **δειγματική εντροπία τάξης-0** μίας συμβολοσειράς s που έχει μήκος n και προέρχεται από ένα αλφάβητο Σ με πληθάριθμο $|\Sigma| = h$ ορίζουμε την ποσότητα:*

$$H_0(s) = \sum_{i=0}^{h-1} \frac{n_1}{n} \log \frac{n}{n_1},$$

όπου n_i το πλήθος των εμφανίσεων του συμβόλου $s_i \in \Sigma$ στην συμβολοσειρά s.

[26] **kaplan2007simpler**.

Αν παρατηρήσουμε τον παραπάνω ορισμό θα δούμε ότι μοιάζει πάρα πολύ με τον αυτόν της εντροπίας. Συγκεκριμένα η μόνη διαφορά με την θεωρητική εντροπία είναι ότι έχουμε αντικαταστήσει τις πιθανότητες των συμβόλων με τις σχετικές τους συχνότητες. Ο λόγος που γίνεται αυτό είναι πώς δεν ξέρουμε αν η πηγή από την οποία προέρχεται η s είναι εργοδική και στάσιμη ώστε να μπορούμε με βεβαιότητα να πούμε πως θα συγκλίνει σε κάποια στάσιμη κατανομή ώστε να υπάρχει η πιθανότητα του κάθε συμβόλου. Αν όμως η υποβόσκουσα πηγή είναι εργοδική και στάσιμη, τότε ξέρουμε πώς για μεγάλα n θα ισχύει ο νόμος των μεγάλων αριθμών και η σχετική συχνότητα $\frac{n_1}{n} \overset{n\to\infty}{\to} Pr[X = x_i] = \pi_i$, όπου π_i η πιθανότητα του συμβόλου s_i από την στάσιμη κατανομή της αλυσίδας. Άρα σε αυτή την περίπτωση για μεγάλα n η δειγματική εντροπία θα συγκλίνει στην θεωρητική. Να σημειώσουμε στο σημείο αυτό μία λεπτομέρεια για τη λέξη "μέτρο". Όταν μιλάμε για πειραματικά μέτρα δεν χρησιμοποιούμε τη λέξη με την αυστηρή μαθηματική έννοια του όρου αλλά πιο πολύ την εννοούμε ως στατιστική εκτιμήτρια μία ποσότητας.

Μία ακόμη στατιστική εκτιμήτρια που χρησιμοποιήσουμε είναι η δειγματική εντροπία k-τάξης. Πριν δώσουμε τον ορισμό της θα εξηγήσουμε τη λογική με την οποία αναπτύχθηκε από τους συγγραφείς του [**kaplan2007simpler**].

Πριν την ορίσουμε όμως θα διατυπώσουμε την εκδοχή που χρησιμοποίησαν οι συγγραφείς για τον μετασχηματισμό. Η εκδοχή αυτή απαντάται συχνά στην βιβλιογραφία αλλά δεν αλλάζει την ουσία του μετασχηματισμού που περιγράψαμε. Πάλι αρχίζουμε με μία συμβολοσειρά μήκους N. Στο τέλος της συμβολοσειράς παραθέτουμε το ειδικό σύμβολο \$ και δημιουργούμε τον γνωστό πίνακα των $N + 1 \times N + 1$ δεξιών μεταθέσεων, το οποίον έπειτα τον ταξινομούμε λεξικογραφικά και παίρνουμε την τελευταία στήλη του διαγράφοντας από αυτήν το ειδικό σύμβολο \$.

abracadabra\$
\$abracadabra
a\$abracadabr
ra\$abracadab
bra\$abracada
abra\$abracad
dabra\$abraca
adabra\$abrac
cadabra\$abra
acadabra\$abr
racadabra\$ab
bracadabra\$a

Ορισμός 5.5. *Έστω μία λέξη $w \in \Sigma^k$ μήκους k. Ορίζουμε ως $\mathbf{w_s}$ την συμβολοσειρά που περιέχει όλα τα σύμβολα που προηγούνται της λέξης w στο s.*

Για παράδειγμα αν $s = abracadabra$, τότε το $a_s = rcdr, b_s = aa$. Είναι εύκολο να δούμε πως αν ξέρουμε το μήκος (πλήθος συμβόλων) της w_s στην ουσία γνωρίζουμε και το πλήθος των φορών που εμφανίστηκε στο κείμενο. Αυτός ο τρόπος βολεύει παρά πολύ στον να μετράμε τις φορές που συναντήσαμε την w_s καθώς ο μετασχηματισμός βασίζεται στην ασύμμετρη πιθανότητα εμφάνισης που έχει ένα σύμβολο s αν ξέρουμε ότι ήδη έχει συμβεί το $r \in w_s$. Έστω μία λέξη $w_k \in \Sigma^k$ μήκους k που ανήκει στην συμβολοσειρά s. Αν w_{w_k} είναι το πλήθος των συμβόλων που προηγούνται της w_k στην s, τότε ξέρουμε πρώτον ότι θα αρχίζουν $|w_{w_k}|$ συνεχόμενες (λόγω της λεξικογραφικής διάταξης) γραμμές. Με αυτό τον τρόπο ο πίνακας του μετασχηματισμού χωρίζεται σε κομμάτια το οποίο επάγει και μία διαμέριση στην τελευταία γραμμή. Συγκεκριμένα το πλήθος των κομματιών t στα οποία χωρίζεται ο πίνακας θα φράσσεται από τον αριθμό $t \leq h^k + k$ καθώς το h^k περιλαμβάνει όλες τις συμβολοσειρές μήκους k, ενώ ο όρος k αναφέρεται σε εκείνες τις συμβολοσειρές μήκους k που μέσα στα συμβολά του περιέχουν το ειδικό σύμβολο \$.

Ορισμός 5.6. *Έστω μία αρχική συμβολοσειρά s και η μετασχηματισμένη $\hat{s} = BWT(S)$. Τότε η s διαμερίζεται το πολύ σε $t \leq h^k + k$ κομμάτια μήκους k, $s = s_1 \cdots s_t$. Τότε η δειγματική εντροπία k-τάξης ορίζεται ως*

$$|s|H_k(s)\sum_{i=1}^{t}|\hat{s}_i|H_0(\hat{s}_i)$$

Η δειγματική εντροπία μήκους k μετράει το μέσο αριθμός bits/σύμβολο που μπορούμε να λάβουμε αν συμπιέσουμε ένα σύμβολο έχοντας υπόψιν μας τα k προηγούμενα.

Η τελευταία στατιστική εκτιμήτρια που θα οριστεί και σχετίζεται με τον μετασχηματισμό είναι η τοπική εντροπία(LE(s) -local entropy). Γνωρίζουμε ότι στην προς τα εμπρός κωδικοποίηση, σε κάθε βήμα κωδικοποιούμε το σύμβολο της τελευταία στήλης σύμφωνα με τη θέση που κατέχει σε μία λίστα συμβόλων του αλφαβήτου Σ. Μετά την κωδικοποίηση του, το σύμβολο έρχεται στην πρώτη θέση. Αν λοιπόν ο μετασχηματισμός κατάφερε να δημιουργήσει ομαδούλες από συνεχόμενα όμοια σύμβολα τότε αναμένουμε το διάνυσμα των ακεραίων να αποτελείται από πολλούς μικρούς ακέραιους. Η τοπική εντροπία ορίζεται πάνω σε αυτό ακριβώς το σκεπτικό. Η προς τα εμπρός κωδικοποίηση θα ονομάζεται $MTF(MoveToFront)$.

Ορισμός 5.7. *Έστω μία αρχική συμβολοσειρά s και η μετασχηματισμένη $\hat{s} = BWT(s)$. Ορίζουμε σαν* ***τοπική εντροπία*** *της s την ποσότητα:*

$$LE_\pi(s) = \sum_{i=1}^{n}\log(MTF_\pi(s)+1)$$

και σαν τοπική εντροπία της μετασχηματισμένης $\hat{s} = BWT(S)$ την ποσότητα:

$$\hat{LE}_\pi(s) = \sum_{i=1}^{n}\log(MTF_\pi(\hat{s})+1)$$

Η τοπική εντροπία αποτελεί το πλήθος των bits που χρειαζόμαστε προκειμένου να κωδικοποιήσουμε την αρχική συμβολοσειρά ή την μετασχηματισμένη αν μας δίνεται μία αρχική μετάθεση π της κωδικοποίησης.

Με βάση αυτές τις στατιστικές εκτιμήτριες που εμφανίστηκαν πρώτα στην δουλεία του Manzini[27] θα αναλύσουμε την συμπίεση με βάση τον Burrows-Wheeler χρησιμοποιώντας την τεχνική της ανταγωνιστικής ανάλυσης. Η μέθοδος αυτή διαφέρει από τον τρόπο της χειρότερης περίπτωσης καθώς εφαρμόζεται περισσότερο για την πρακτική και όχι θεωρητική ανάλυση των αλγορίθμων. Η ανταγωνιστική ανάλυση βρίσκει ένα φράγμα για το μήκος ενός κώδικα που θα παραχθεί από ένα αλγόριθμό συμπίεσης A, συναρτήσει ενός βέλτιστου αλγορίθμου.

Ορισμός 5.8. *Ένας αλγόριθμος συμπίεσης είναι (μ, C)- f ανταγωνιστικός αν για κάθε είσοδο s ισχύει ότι:*

$$|A(s)| \leq \mu \cdot f + C \cdot n + o(n)$$

Όπου C το κόστος του βέλτιστου αλγορίθμου. Για παράδειγμα στον Huffman το $C = 1$ καθώς:
$H_0(s) \leq \overline{L}(s)_{\text{HUFFMAN}} \leq H_0(s) + 1$

Η πραγματική στιγμή της συμπίεσης των τεχνικών που χρησιμοποιούν τον μετασχηματισμό Burrows-Wheeler είναι κατά τη διάρκεια της συμπίεσης του διανύσματος ακεραίων που προκύπτει από την προς τα εμπρός κωδικοποίηση. Έχουμε ήδη πει ότι για να συμπιέσουμε το διάνυσμα ακεραίων χρησιμοποιούμε κάποιον συμπιεστή τάξης 0. Άρα το πρώτο κομμάτι της ανάλυσης του Burrows-Wheeler έγκειται στο να βρούμε ένα φράγμα για το συμπιεσμένο διάνυσμα ακεραίων. Οι Kaplan, Landau και Verbin έδειξαν ότι ο καλύτερος συμπιεστής που μπορούμε να έχουμε για ακεραίους, ο οποίος θα παράγει μήκη συγκρίσιμα με την ποσότητα $SL = \sum_{i=1}^{n}\log(s[i]+1)$[28] είναι ένας συμπιεστής τάξης 0. Η ποσότητα $\sum_{i=1}^{n}\log(s[i]+1)$ είναι επί της ουσίας το πλήθος των ψηφίων που χρειαζόμαστε για να κωδικοποιήσουμε τους ακεραίους στους οποίους αντιστοιχίζονται τα σύμβολα της ακολουθίας s προς συμπίεση.

[27] **manzini2001analysis**.
[28] Το SL είναι ακρωνύμιο της φράσης sum of logarithms, δηλαδή άθροισμα λογαρίθμων.

Θεώρημα 5.7. *Για οποιαδήποτε σταθερά $\mu > 0$ ένας αλγόριθμός τάξης-0 (Order(0)) είναι$(\mu, \log \zeta_h(\mu) + C_{Order(0)})$ - SL ανταγωνιστικός, όπου $\zeta_h(\mu) = \frac{1}{1^m} + \frac{1}{2^m} + \cdots \frac{1}{h^m}$*

Απόδειξη

Από τον ορισμός της ανταγωνιστικής ανάλυσης γνωρίζουμε ότι για να είναι ένας συμπιεστής A τάξης-0, $(\mu, \log \zeta_h(\mu))$ - SL ανταγωνιστικός αρκεί να ισχύει ότι:

$$|A(s)| \leq \mu \cdot SL(s) + n \cdot (\log \zeta_h(\mu) + C_{Order_0}) + o(n).$$

Από την θεωρία της πληροφορίας γνωρίζουμε ότι το μέσο μήκος κώδικα που παράγει ένας οποιοσδήποτε συμπιεστής τάξης-0 φράσσεται κάτω από την ποσότητα $n \cdot H_0(s)$ από το οποίο έπεται ότι $|A(s)| \geq nH_0(s)$. Επίσης επειδή η ποσότητα $\mu \cdot SL(s) + n \log \zeta_h(\mu) < \mu \cdot SL(s) + n \cdot (\log \zeta_h(\mu) + C_{Order_0}) + o(n)$ καταλαβαίνουμε ότι για να αποδείξουμε το θεώρημα αρκεί να δείξουμε ότι:

$$nH_0(s) \leq \mu \cdot SL(s) + n \log \zeta_h(\mu)$$

$$n \cdot \sum_{i=0}^{h-1} \frac{n_i}{n} \cdot \log \frac{n}{n_i} \leq \mu \cdot \sum_{i=1}^{n} \log(s[i]+1) + n \log \zeta_h(\mu)$$

Αν το κάθε σύμβολο του αλφαβήτου Σ εμφανίζεται n_i φορές τότε $\sum_{i=1}^{n} \log(s[i]+1) = \sum_{i=1}^{h-1} n_i \log(i+1)$. Άρα:

$$n \cdot \sum_{i=0}^{h-1} \frac{n_i}{n} \cdot \log \frac{n}{n_i} \leq \mu \cdot \sum_{i=1}^{n} \log(s[i]+1) + n \log \zeta_h(\mu) \Rightarrow n \cdot \sum_{i=0}^{h-1} \frac{n_i}{n} \cdot \log \frac{n}{n_i} \leq \mu \cdot \sum_{i=1}^{h-1} n_i \log(i+1) + n \log \zeta_h(\mu) \Leftrightarrow$$
$$\sum_{i=0}^{h-1} n_i \log \frac{n}{n_i} - \mu \cdot \sum_{i=1}^{h-1} n_i \log(i+1) \leq n \log \zeta_h(\mu) \Leftrightarrow \sum_{i=1}^{h-1} n_i \log \frac{n}{n_i \cdot (i+1)^\mu} \leq n \log \zeta_h(\mu) \Leftrightarrow$$
$$\sum_{i=1}^{h-1} p_i \log \frac{1}{p_i \cdot (i+1)^\mu} \leq \log \zeta_h(\mu) \Leftrightarrow \sum_{i=1}^{h-1} p_i \log \frac{1}{p_i \cdot (i+1)^\mu} \leq \log \zeta_h(\mu)$$

Από την ανισότητα Jensen έχουμε ότι:

$$\sum_{i=1}^{h-1} p_i \log \frac{1}{p_i \cdot (i+1)^\mu} \leq \log \left(\sum_{i=1}^{h-1} p_i \frac{1}{p_i \cdot (i+1)^\mu} \right) = \log \left(\sum_{i=1}^{h-1} \frac{1}{\cdot (i+1)^\mu} \right) = \log \zeta_h(\mu)$$

Οι συγγραφείς για να δείξουν ότι το παραπάνω φράγμα είναι αυστηρό για οποιονδήποτε συμπιεστή τάξης 0, απέδειξαν ένα άνω φράγμα για τους συμπιεστές τάξης 0.

Θεώρημα 5.8. *Για οποιαδήποτε σταθερά $\mu > 0$ και $C < \log \zeta_h(\mu)$ δεν υπάρχει αλγόριθμός τάξης-0 (Order0) που να είναι $(\mu, \log \zeta_h(\mu) + C_{Order_0})$ - SL ανταγωνιστικός.*

Απόδειξη

Για να είναι ένας αλγόριθμός A $(\mu, C) - f$ ανταγωνιστικός είπαμε ότι πρέπει:

$$|A(s| \leq \mu f(s) + nC + o(n)$$

, όπου το $o(n)$ συμβολίζει μία συνάρτηση g για την οποία ισχύει $\lim n \to \infty \frac{g(n)}{n} = 0$, ή αλλιώς λέμε ότι $f(n) \in o(g(n))$ αν υπάρχει μία σταθερά $c : f(n) < cg(n)$

Άρα για να αποδείξουμε ότι ένα αλγόριθμός δεν είναι $(\mu, \log \zeta_h(\mu) + C_{Order_0})$ - SL ανταγωνιστικός, αρκεί να δείξουμε ότι υπάρχει μία είσοδος s για την οποία ισχύει:

$$|A(s)| \leq \mu \cdot LS(s) + |s| \quad \underbrace{(\log(\zeta_h(\mu)) - \epsilon)}_{\text{Από την υπόθεση } C < \log \zeta_h(\mu)} \quad + \quad \underbrace{|s| f(|s|)}_{\text{Από τον ορισμό του } o(n)}$$

Θα κατασκευάσουμε λοιπόν μία κλάση συμβολοσειρών s με μήκος S, για τις οποίες δεν ισχύει το άνω φράγμα όταν $C < \log \zeta_h(\mu)$ για οποιοδήποτε $\mu > 0, \epsilon > 0$ και οποιαδήποτε συνάρτηση f.

Έστω το πλήθος των φορών που συναντούμε το σύμβολο $h \in \Sigma$ στο s ισούται με τον ακέραιο $a_i = \frac{n}{\zeta_h(\mu)(i+1)^\mu}$ τότε το μήκος του κώδικας θα είναι $\sum_{i=1}^{h-1} a_i \log(i+1) = \sum_{i=1}^{h-1} \frac{n}{\zeta_h(\mu)(i+1)^\mu \log(i+1)}$. Τότε το πλήθος της κλάσης αυτών των ειδικών συμβολοσειρών θα είναι $N(n) = \frac{n!}{\prod_{i=1}^{h-1} a_i!}$. Άρα το πλήθος των bits που θα χρειαστούμε για να κωδικοποιήσουμε μία συμβολοσειρά που ανήκει σε αυτή την κάση θα είναι το λιγότερο $\log N(n)$. Από τον τύπο του Stirling ξέρουμε ότι $\frac{1}{2}\sqrt{2\pi n}\left(\frac{n}{e}\right)^n \leq n! \leq \frac{3}{2}\sqrt{2\pi n}\left(\frac{n}{e}\right)^n$. Με βάση την παραπάνω προσέγγιση:

$$\log N(n) = \log \frac{n!}{\prod_{i=1}^{h-1} a_i!} \geq \log \frac{\frac{1}{2}\sqrt{2\pi n}\left(\frac{n}{e}\right)^n}{\prod_{i=1}^{h-1} \frac{3}{2}\sqrt{2\pi a_i}\left(\frac{a_i}{e}\right)_i^a} = \log \frac{2^{h-1}}{3^h} + \log \frac{\sqrt{2\pi n}}{\prod_{i=1}^{h-1}\sqrt{2\pi a_i}} + \log \frac{n}{e}^n - \log \frac{a_i}{e}^n$$

Σημείωση: Λέμε ότι μία συνάρτηση $f(n) \in O(g(n))$ αν υπάρχει μία σταθερά $c : f(n) \leq c \cdot g(n)$.

$$\log N(n) = \log \frac{n!}{\prod_{i=1}^{h-1} a_i!} \geq \log \frac{\frac{1}{2}\sqrt{2\pi n}\left(\frac{n}{e}\right)^n}{\prod_{i=1}^{h-1} \frac{3}{2}\sqrt{2\pi a_i}\left(\frac{a_i}{e}\right)^{a_i}} = \log \frac{2^{h-1}}{3^h} + \log \frac{\sqrt{2\pi n}}{\prod_{i=1}^{h-1}\sqrt{2\pi a_i}} + \log \left(\frac{n}{e}\right)^n - \log \prod_{i=1}^{h-1}\left(\frac{a_i}{e}\right)^{a_i} =$$

$$\log \frac{2^{h-1}}{3^h} + \log \frac{\sqrt{2\pi n}}{\prod_{i=1}^{h-1}\sqrt{2\pi \frac{n}{\zeta_h(\mu)(i+1)^\mu}}} + n\log n - n - \sum_{i=1}^{h-1} \log a_i \log a_i + \sum_{i=1}^{h-1} a_i =$$

$$\log \frac{2^{h-1}}{3^h} + \log(2\pi n)^{\frac{1-h}{2}} + \log \frac{1}{\prod_{i=1}^{h-1} \zeta_h(\mu)^{\frac{1}{2}}} + \log \frac{1}{\prod_{i=1}^{h-1}(i+1)^{\frac{\mu}{2}}} + n\log n - n - \sum_{i=1}^{h-1} a_i \log a_i + n =$$

$$\log \frac{2^{h-1}}{3^h} + \frac{1-h}{2}\log(2\pi n) + \frac{h}{2}\log \frac{1}{\zeta_h(\mu)} + \sum_{i=1}^{h-1} \frac{\mu}{2}\frac{1}{(i+1)} + \sum_{i=1}^{h-1} a_i \log \frac{n}{a_i}$$

Επειδή μιλάμε για ασυμπτωτική ανάλυση οποιοσδήποτε όρος δεν περιέχει το n θα σταματήσει να είναι σημαντικός για μεγάλες εισόδους, οπότε:

$$\log N(n) \geq \log \frac{2^{h-1}}{3^h} + \frac{1-h}{2}\log(2\pi n) + \frac{h}{2}\log \frac{1}{\zeta_h(\mu)} + \sum_{i=1}^{h-1} \frac{\mu}{2}\frac{1}{(i+1)} + \sum_{i=1}^{h-1} a_i \log \frac{n}{a_i} \geq -O(\log n) + \sum_{i=1}^{h-1} a_i \log \frac{n}{a_i}$$

Οπότε το ελάχιστο μήκος της ακολουθίας θα διαφέρει από το $\mu L(n)$ κατά:

$$logN(n) - \mu L(n) \geq -O(\log n) + \sum_{i=1}^{h-1} a_i \log \frac{n}{a_i} - \mu \sum_{i=1}^{h-1} a_i \log(i+1) = -O(\log n) + \sum_{i=1}^{h-1} a_i \log \frac{n}{a_i (i+1)^\mu} =$$

$$- O(\log n) + \sum_{i=1}^{h-1} a_i \log \frac{n}{\frac{n}{\zeta_h(\mu)(i+1)^\mu}(i+1)^\mu} = -O(\log n) + \sum_{i=1}^{h-1} a_i \log \zeta_h(\mu) = -O(\log n) + \log \zeta_h(\mu) \sum_{i=1}^{h-1} a_i =$$

$$- O(\log n) + n \log \zeta_h(\mu) \Rightarrow \log N(n) - \mu L(n) \geq -O(\log n) + n \log \zeta_h(\mu) \Rightarrow |A(s)| \geq \log N(n) \geq \mu L(n) + n \log \zeta_h($$

Άρα για οποιαδήποτε $f \in O(\log n)$ υπάρχει μία σταθερά $\epsilon > 0$ για την οποία ισχύει $f(n) \leq \epsilon \log n \leq \epsilon \cdot n$ για μεγάλα n. Άρα για οποιαδήποτε f:

$$|A(s)| \geq \log N(n) \geq \mu L(n) + n \log \zeta_h(\mu) - O(\log n) \geq \mu L(n) + n \log \zeta_h(\mu) - \epsilon \log n \geq \mu L(n) + n \log \zeta_h(\mu) - \epsilon n \Rightarrow$$
$$|A(s)| \geq \log N(n) \geq \mu L(n) + n(\log \zeta_h(\mu) - \epsilon)$$

Το παραπάνω φράγμα ισχύει και στην περίπτωση που οι a_i δεν είναι ακέραιοι. Τότε παίρνουμε το $\lceil a_i \rceil < a_i + 1$. Άρα το πολύ να απέχουμε από τα πραγματικά μήκη $\pm h$, ποσότητα αμελητέα για την ασυμπτωτική ανάλυση αφού δεν είναι τάξης n.

Αυτό που αποδείξαμε με τα παραπάνω θεωρήματα είναι πως αν θέλουμε να κωδικοποιήσουμε ακεραίους τότε ο αλγόριθμος τάξης 0 που θα επιλέξουμε είναι $SL-$ ανταγωνιστικός. Αν όμως στο $SL(s)$ αντικαταστήσουμε το s με το διάνυσμα ακεραίων που προέρχεται από την προς τα εμπρός κωδικοποίησης της μετασχηματισμένης $\hat{s} = BTW(s)$, τότε θα έχουμε ότι $SL(s) = SL(MTF(BTW(s))) = \sum_{i=1}^{n} \log(MTF_\pi(\hat{s})[i] + 1) = \hat{LE}(s) = LE(\hat{s})$. Οπότε η συμπίεση στο τελευταίο βήμα της κωδικοποίησης είναι $(\mu, \log \zeta_h(\mu) + C_{Order0}) - \hat{LE}$ ανταγωνιστική. Άρα σύμφωνα με το τελευταίο θεώρημα μόλις αποδείχτηκε ότι:

$$|BWT(s)| \leq \mu \hat{LE}(s) + (\log \zeta_h(\mu) + C_{Order0}) + o(n)$$

Εμείς όμως επιθυμούμε να βρούμε ένα φράγμα για τον Burrows-Wheeler που θα σχετίζεται με την δειγματική εντροπία $k-$τάξης, οπότε το επόμενο βήμα είναι να εξάγουμε τη σχέση μεταξύ της τοπικής εντροπίας $\hat{LE}(s)$ και της δειγματικής $H_k(s)$.

Το διάνυσμα των ακεραίων της μετασχηματισμένης σειράς $\hat{s} = BWT(s)$ ξέρουμε ότι κατά κύριο λόγο θα περιλαμβάνει μικρούς ακεραίους που θα διακόπτονται από μεγαλύτερους κάθε φορά που μεταβαίνουμε στην κωδικοποίηση ενός κομματιού s_i του s. Αν δεν πάρουμε υπόψιν μας της πρώτη φορά που συναντάμε ένα καινούργιο γράμμα τότε καταλαβαίνουμε πως από το διάνυσμα θα αφαιρεθούν πολύ μεγάλοι ακέραιοι. Την τοπική αυτή εντροπία την συμβολίζουμε με LE^1. Τότε η συνεισφορά των εναπομεινάντων συμβόλων σ του s θα: είναι:

$$\sum_{i:s[i]=\sigma} \log(MTF^1(s)[i] + 1)$$

Επειδή όμως το διάνυσμα των ακεραίων που προέρχεται από την MTF^1 κωδικοποίηση θα έχει μέγεθός μικρότερο του n και οι μεγάλοι ακέραιοι έχουν αφαιρεθεί θα ισχύει ότι:

$$\sum_{i:s[i]=\sigma} (MTF^1(s)[i] + 1) \leq n$$

Οπότε αν n_σ το πλήθος των φορών που συναντάμε το σ στο s, τότε για να κωδικοποιήσουμε όλες τις εμφανίσεις που του απέμειναν όταν αποκόψαμε τις πρώτες του θα χρειαστούμε $n_\sigma \log(\frac{\sum_{i:s[i]=\sigma} \log(MTF^1(s)[i]+1)}{n_\sigma})$ bits. Τότε συνολικά θα χρειάζονται:

$$LE^1(s) = \sum_{i:s[i]=\sigma} \log(MTF^1(s)[i]+1) \leq \sum_{\sigma} n_\sigma \log(\frac{\sum_{i:s[i]=\sigma} \log(MTF^1(s)[i]+1)}{n_\sigma}) \leq \sum_{\sigma} n_\sigma \log(\frac{n}{n_\sigma}) = n \cdot H_0(s)$$

Η χειρότερη περίπτωση που μπορεί να μας τύχει κατά τη μετατροπή ης τελευταία στήλης σε διάνυσμα ακεραίων είναι η αρχική μετάθεση να είναι τέτοια ώστε να παράξει τους μεγαλύτερους δυνατούς ακέραιους. Τότε η τοπική εντροπία χειρότερης περίπτωσης $LE_{\text{worst case}}$ θα είναι όση η LE^1 μαζί με την κωδικοποίηση των ακεραίων που συναντήσαμε για πρώτη φορά. Στην χειρότερη περίπτωση που έχουμε μία συμβολοσειρά s που περιλαμβάνει όλα τα γράμματα του αλφαβήτου Σ, ξέρουμε ότι θα έχουμε h πρώτες εμφανίσεις που κάθε μία κωδικοποιείται με το πολύ $logh$ σύμβολα. Άρα

$$LE_{\text{worst case}} \leq LE^1 + h \log h \Rightarrow LE_{\text{worst case}} \leq nH_0(s) + h \log h$$

Αφού βρήκαμε ένα φράγμα για την στατιστική εκτιμήτρια της τοπικής εντροπία χειρότερης περίπτωσης πάμε να δούμε αν θα είναι και το μοναδικό εξετάζοντας την κυρτότητα της. Ας υποθέσουμε ότι η π_1 είναι η αρχική μετάθεση που προκαλεί την χειρότερη τοπική εντροπία. Όπως είπαμε στους ορισμούς στην αρχή το s μπορεί να σπάσει στα κομμάτια $s_1 \cdots s_t$, με $t \leq h^k + k$. Τότε αν σε κάθε κομμάτι s_i που κωδικοποιείται με την προς τα εμπρός κωδικοποίηση εφαρμοστεί η χειρότερη μετάθεση π είναι εύκολο να δούμε ότι $LE_{\text{worst case}}(s) \leq \sum_{i=1}^{t} LE_{\text{worst case}}(s_i)$ Με βάση όλη την προηγηθείσα ανάλυση είμαστε σε θέση να συνδέσουμε την τοπική εντροπία χειρότερη περίπτωσης με την δειγματική εντροπία $k-$τάξης.

Θεώρημα 5.9. *Για κάθε $k \geq 0$ και κάθε συμβολοσειρά s ισχύει:*

$$|s|H_k(s) \geq \hat{LE}_{worst\ case} - (h^k + k)h \log h$$

Απόδειξη
Από τον ορισμό της $k-$οστής εντροπίας ξέρουμε ότι:

$$H_k(s) = \sum_{i=1}^{t} |\hat{s_i} H_0(\hat{s_i})|$$

Από την κυρτότητα της εντροπίας ξέρουμε ότι $LE_{worst\ case}(s) = \hat{LE}worst\ case(s) \leq \sum_{i=1}^{t} \hat{LE}worst\ case(s_i)$ Επειδή όμως $LE_{worst\ case} \leq nH_0(s) + h \log h$, έχουμε

$$LE_{worst\ case}(s) = \hat{LE}worst\ case(s) \leq \sum_{i=1}^{t} \hat{LE}worst\ case(s_i) \leq \sum_{i=1}^{t} (|s_i|H_0(\hat{s_i}) + h \log h) = |s|H_k(s) + th \log h.$$

Επειδή έχουμε ήδη αποδείξει ότι ο Burrows-Wheeler είναι LE ανταγωνιστικός και $LE \leq LE_{\text{worst case}}$ έπεται ότι:

$$|BWT(s)| \leq \mu \hat{LE}(s) + (\log \zeta_h(\mu) + C_{Order0}) + o(n) \Rightarrow$$
$$|BWT(s)| \leq \mu \hat{LE}_{\text{worst case}}(s) + (\log \zeta_h(\mu) + C_{Order0}) + o(n) \Rightarrow$$
$$\boxed{|BWT(s)| \leq \mu H_k(s) + (\log \zeta_h(\mu) + C_{Order0}) + o(n)}$$

Κεφάλαιο 6

Όταν η συμπίεση συνάντησε την μουσική ...

6.1 Περιγραφή Πειράματος

Στο κεφάλαιο αυτό θα παρουσιάσουμε την πειραματική διαδικασία που ακολουθήσαμε προκειμένου να αναλύσουμε την συμπεριφορά διάφορων συμπιεστών που παρουσιάστηκαν στην προηγούμενη ενότητα. Από την θεωρία συμπίεσης γνωρίζουμε πως ένα κείμενο που περιέχει επαναλαμβανόμενες εκφράσεις συμπιέζετε πολύ καλύτερα από ότι ένα τυχαίο κείμενο. Αντίστοιχα ένα μουσικό κομμάτι που περιέχει επαναλαμβανόμενους στίχους θα συμπιέζετε πολύ περισσότερο από ότι ένα κομμάτι χωρίς. Για το λόγο αυτό σκεφτήκαμε πως τα ελληνικά τραγούδια αποτελούν ένα από τα καλύτερα σύνολα δεδομένων προκειμένου να εξετάσουμε την συμπεριφορά των συμπιεστών ανάμεσα σε διαφορετικά μεγέθη αρχείων και περιεχόμενα. Επίσης είναι ένας ευφάνταστος τρόπος προκειμένου να πάρουμε μία εικόνα ως προς τον πλούτο τον στίχων για τα διάφορα είδη τραγουδιών καθώς ένα τραγούδι που πετυχαίνει χαμηλά ποσοστά συμπίεσης σημαίνει ότι δεν περιέχει πολλές επαναλαμβανόμενες λέξεις, εκφράσεις, κ.λ.π

Αν φέρουμε στο μυαλό μας ένα μέρος των τραγουδιών που ξέρουμε θα δούμε ότι είναι το ιδανικό δείγμα για συμπίεση καθώς παρουσιάζουν επαναλαμβανόμενους στίχους ή ακόμα επαναλήψεις κομματιών από στίχους σε επόμενους ή σε επόμενες στροφές. Για το λόγο αυτό ένας μέρος της συμπίεσης οφείλεται στη δομή των τραγουδιών, δηλαδή σε ομοιοκαταληξίες και στις επαναλήψεις στίχων που προείπαμε. Για παράδειγμα συχνά στα παραδοσιακά ή ρεμπέτικα τραγούδια ο ίδιος στίχος επαναλαμβάνετε με διαφορετική σειρά λέξεων ή επαναλαμβάνεται μέρος αυτού.

Απάνω στην μαύρα μου μάτια
Απάνω στην τριανταφυλλιά
Χτίζει η πέρδικα φωλιά
Χτίζει η πέ- μαύρα μου μάτια
Χτίζει η πέρδικα φωλιά

Οι παραπάνω στίχοι είναι από το παραδοσιακό τραγούδι "Απάνω στην τριανταφυλλιά" που είναι ενδεικτικό της επανάληψης ίδιων στίχων ή μεγάλο μέρος αυτών. Στο παρακάτω χρωματισμένο κείμενο φαίνονται τα ταιριάσματα που μπορεί να βρει ένας συμπιεστής σαν το LZ77. Παρατηρούμε ότι σχεδόν όλο το κομμάτι είναι υπογραμμισμένο.

Απάνω στην μαύρα μου μάτια
Απάνω στην τριανταφυλλιά
Χτίζει η πέρδικα φωλιά
Χτίζει η πέ- μαύρα μου μάτια
Χτίζει η πέρδικα φωλιά

6.2 Πειραματική Διαδικασία

Για να συλλέξουμε ένα δείγμα στίχων από διαφορετικά είδη μουσικής χρησιμοποιήσαμε την ιστοσελίδα stixoi.info. Συγκεκριμένα φτιάξαμε κάποιες ρουτίνες συλλογής δεδομένων(scraping) προκειμένου να κατεβάσουμε ένα επαρκές, σε μέγεθος, σύνολο στίχων από διάφορους στιχουργούς. Η επιλογή των στιχουργών και των στίχων έγινε με βάση τις προβολές που είχαν συλλέξει από τους επισκέπτες της ιστοσελίδας. Για να ξεκινήσουμε τη συλλογή αρχικά ζητήσαμε από τον server τους πίνακες που περιέχουν τους στιχουργούς ομαδοποιημένους σε αλφαβητική σειρά ανά γράμμα. Στην κάτω εικόνα φαίνεται ο πίνακας των στιχουργών που ξεκινάνε από Α. Κάθε γράμμα έχει μόνο ένα πίνακα οπότε για να κατεβάσουμε όλους τους στιχουργούς που υπάρχουν αποθηκευμένοι στην βάση δεδομένων του stixoi.info άρκεσαν 24 αιτήσεις στον server. Ο μέσος χρόνος που απαντούσε η ιστοσελίδα στις αιτήσεις μας ήταν $0.33ms$

Σχήμα 6.1: Ο πίνακας των στιχουργών που αρχίζουν από το γράμμα Α.

Το κατέβασμα έγινε με την χρήση ενός προγράμματος γραμμένο σε python χρησιμοποιώντας τα πακέτα request και beautiful soup 4. Πριν παρουσιάσουμε τον κώδικα να διευκρινίσουμε ότι σκοπός δεν ήταν να γράψουμε έναν βέλτιστο ή ταχύ web scrapper, αλλά κυρίως έναν ασφαλή για την ιστοσελίδα. Ήδη μπορούμε να διαβεβαιώσουμε πως αν το ζητούμενο είναι η ταχύτητα,ο συνδυασμός των πακέτων requests και beautiful soup 4 δεν είναι καλή επιλογή. Καλύτερα να προτιμηθούν πακέτα όπως το scrapy και το selenium. Επίσης να επισημανθεί πως κατά την διάρκεια του scraping δεν χρησιμοποιήθηκαν ψευδή headers ούτε ψευδείς ip και browsers agents. Η ιστοσελίδα γνώριζε καθολη την διάρκεια της διαδικασίας ότι τα requests γίνονται από την python. Ακόμη πριν ξεκινήσουμε την συλλογή δεδομένων συμβουλευτήκαμε το robots.txt του server. Το συγκεκριμένο αρχείο βρίσκεται σε κάθε server και περιλαμβάνει την πολιτική που έχει η ιστοσελίδα απέναντι σε προγράμματα περιήγησης και συλλογής δεδομένων. Η πρόσβαση σε αυτό το αρχείο γίνεται πολύ εύκολα πληκτρολογώντας την διεύθυνση της ιστοσελίδας ακολουθούμενη από μία πλάγια γραμμή και τον όρο robots.txt

Σχήμα 6.2: Τα αρχεία robots.txt για τις ιστοσελίδες youtube και stixoi.info αντίστοιχα.

Στην παραπάνω εικόνα βλέπουμε πως η ιστοσελίδα stixoi.info επιτρέπει όλα τα προγράμματα περιήγησης (User-agent:*) ενώ το youtube μόνο αυτά που συνεργάζονται με την google. Η λέξη Disallow σημαίνει ότι απαγορεύεται να εξερευνήσουμε τους καταλόγους και τα αρχεία που ακολουθούν. Οι δύο φωτογραφίες αποτελούν ακραία παραδείγματα ανάμεσα σε μία ιστοσελίδα που δεν αφήνει να γίνει sraping σχεδόν σε τίποτα και το stixoi.info που έχει μία πολύ ελαστική πολιτική για web scrapers.

Στο παρακάτω αρχείο κώδικα (Lyricist_Scraping.py) φαίνεται το πρόγραμμα που χρησιμοποιήσαμε προκειμένου να συλλέξουμε τους πίνακες των στιχουργών που είναι ομαδοποιημένοι κατά γράμμα. Η ρουτίνα ξεκινάει εφοδιάζοντας το πρόγραμμα με το γράμμα της αλφαβήτου από το οποίο επιθυμούμε να συλλέξουμε στιχουργούς. Ο κατασκευαστής (__init__) της κλάσης (Fetch_Lyricist_Table) κατασκευάζει ένα κατάλογο (make_letter_dir) με το όνομα Lyricists_From_{Alphabet_Letter} μέσα στον τρέχον φάκελο που εργαζόμαστε (os.getcwd()). Αν ο φάκελος αποτύχει να δημιουργηθεί το πρόγραμμα μας ειδοποιεί με ένα σφάλμα.

Στην συνέχεια στέλνουμε ένα request σε url της αρεσκείας μας με τη χρήση της μεθόδου get_source_code. Αν ο server αργήσει παραπάνω από 5 δευτερόλεπτα να απαντήσει γίνεται timeout. Επειδή σε αυτό το κομμάτι της συλλογής δεδομένων ο αριθμός αιτήσεων προς το server είναι μικρός, 24 συγκεκριμένα, δεν έχουμε πάρει ιδιαίτερα μέτρα πέρα του timeout σε περίπτωση αποτυχίας. Στις επόμενες ρουτίνες που αυξάνονται πολύ οι αιτήσεις προς το server, ο σχεδιασμός αποστολής τους περιλαμβάνει δικλίδες ασφαλείας σε περιπτώσεις που ο server αρχίζει να δίνει απαντήσεις με πιο αργούς ρυθμούς. Πάντως σε όλα τα προγράμματα έχουμε φροντίσει το πρόγραμμα να ενημερώνει για την επιτυχή επιστολή των αιτήσεων και την επιτυχή δημιουργία αρχείων και καταλόγων στο σύστημα. Αφού λοιπόν κατεβάσουμε τον πηγαίο κώδικα, τον επεξεργαζόμαστε με το πακέτο beautiful soup (soup = bs(request.content, features="lxml")) και τον αποθηκεύουμε σε ένα αρχείο με όνομα page_source_code.txt.

Lyricist_Scraping.py

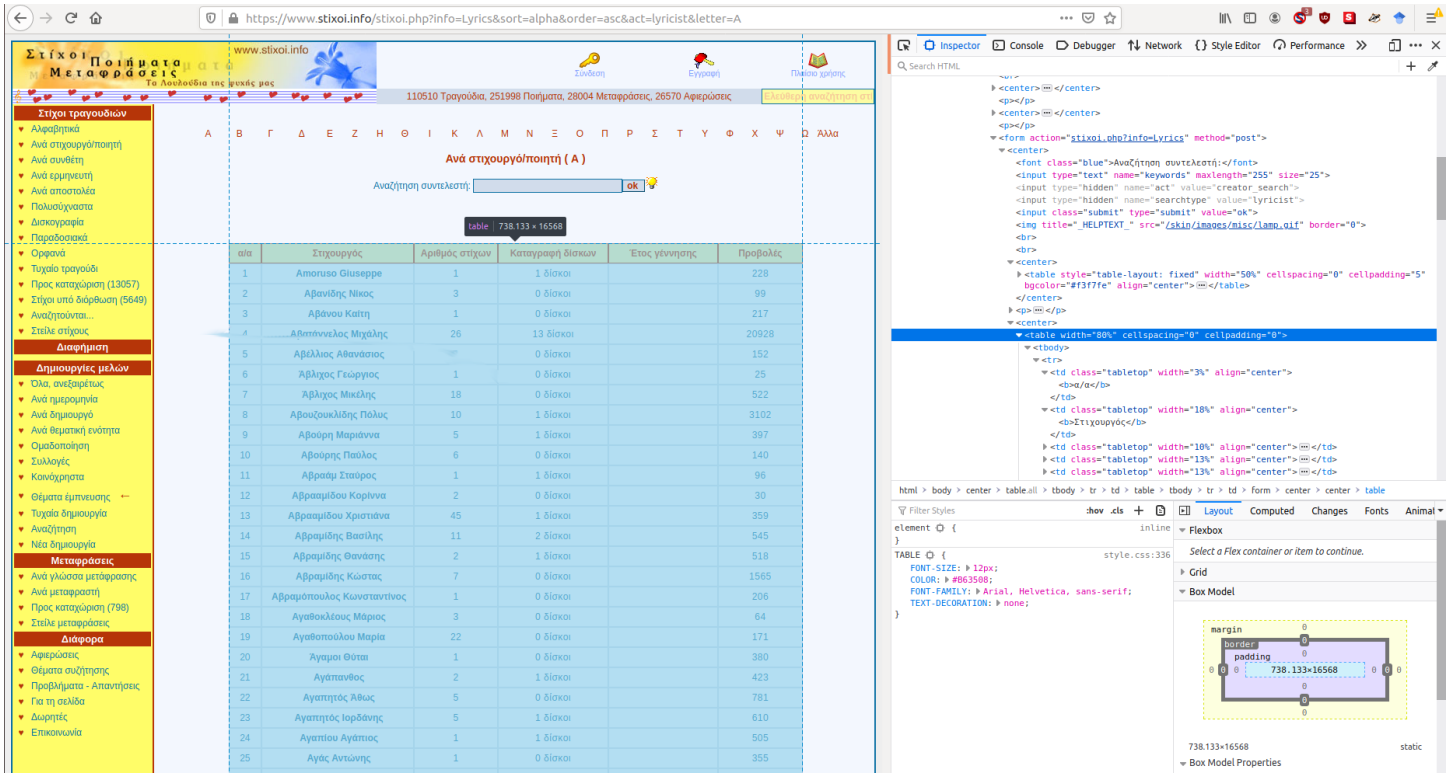
```
1  import requests
2  from bs4 import BeautifulSoup as bs
3  import re
4  import lxml
5  import os
6  import csv
7
8  class Fetch_Lyricist_Table:
9      url = None
10     html = None
11     search_keys = None
12     letter = None
13     dir_path = None
14     html_path=None
```

```
15      table_file =None
16
17
18      def __init__(self, letter):
19          self.letter = letter
20          path = os.getcwd()
21          self.dir_path = path + '/Second_Round/Lyricists_From_' + self.letter
22
23      def get_dir_path(self):
24          return self.dir_path
25
26      def make_letter_dir(self):
27          try:
28              os.mkdir(self.dir_path)
29          except OSError:
30              print("Creation of the directory %s failed" % self.dir_path)
31          else:
32              print("Successfully created the directory %s " % self.dir_path)
33
34      def get_source_code(self, url):
35          self.url = url
36          request = requests.get(self.url)
37          print(request.status_code)
38          respond_time = str(round(request.elapsed.total_seconds(), 2))
39          print(respond_time)
40          soup = bs(request.content, features="lxml")
41          self.html_path=self.dir_path + '/page_source_code.txt'
42          html = open(self.html_path, 'w')
43          html.write(str(soup))
44          html.close()
45          self.html = html
46
47      def get_html(self):
48          return self.html
49
50
51      def create_table(self,keys):
52          source_code=open(self.html_path, 'r')
53          html =source_code.read()
54          source_code.close()
55          soup=bs(html,features="lxml")
56          table = soup.find('table',keys)
57          links_list = table.find_all('a', href=re.compile('info=Lyrics&act=index&sort=alpha&lyricist_id'))
58          id_list = []
59          for link in links_list:
60              if 'href' in link.attrs:
61                  s = (str(link.attrs['href']))
62                  id_list.append(s[s.rfind('lyricist='):len(s)])
63          id_list.insert(0, 'Lyricist_id')
64          print(id_list)
65          self.html_path=self.dir_path+'/Lyricist_Table.csv'
66          print(self.html_path)
67          table_file = open(self.html_path, mode='w')
68          table_writer = csv.writer(table_file, delimiter=',', quotechar='"', quoting=csv.QUOTE_MINIMAL)
69          j = 0
70          for row in table.find_all('tr'):
71
72              cells = row.find_all('td')
73              cell_list = [1, 2, 5]
74              row_list = []
75              for i in range(0, len(cell_list)):
76                  cell = cells[cell_list[i]]
77                  row_list.append(str(cell.text))
78              row_list.insert(0, id_list[j])
79              j = j + 1
80              table_writer.writerow(row_list)
81
82          table_file.close()
```

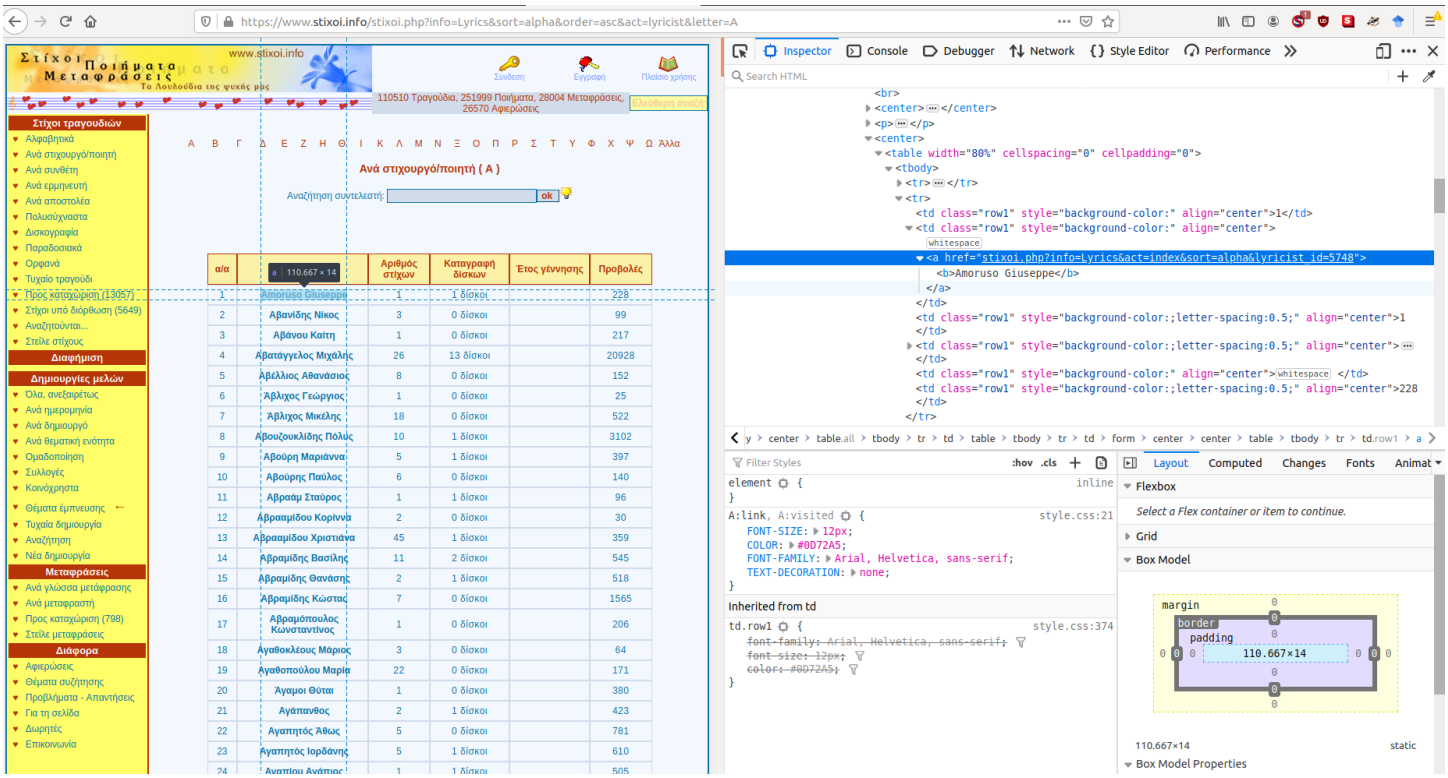
Στο τελικό στάδιο της συγκεκριμένης ρουτίνας επεξεργαζόμαστε τον αποθηκευμένο πηγαίο κώδικα του προηγούμενου βήματος ώστε να πάρουμε όλα τα μοναδικά χαρακτηριστικά των στιχουργών (lyricist_id). Τα μοναδικά αναγνωριστικά είναι επί της ουσίας ο τρόπος που είναι αποθηκευμένοι οι στιχουργοί στην βάση δεδομένων της ιστοσελίδας. Επειδή βρίσκονται στα url των στιχουργών τα χρειαζόμαστε ώστε να τα χρησιμοποι-

ήσουμε σε επόμενες αιτήσεις προς τον server όταν χρειαστεί να κατεβάσουμε τους πίνακες των στιχουργών. Η διαδικασία εξαγωγής του πίνακα των στιχουργών από τον πηγαίο κώδικα γίνεται χρησιμοποιώντας τον αναλυτή html, beautiful soup. Οι αναλυτές μας βοηθού να εντοπίσουμε μέσα στο πηγαίο κώδικα ετικέτες (tags) με τις οποίες σημαίνονται τα διάφορα στοιχεία της ιστοσελίδας όπως πίνακες, λίστες, ευρετήρια, φόρμες εισόδου ή συμπλήρωσης στοιχείων και πολλά άλλα. Εμείς από την συγκεκριμένη σελίδα επιθυμούμε να φτιάξουμε έναν πίνακα ο οποίος θα περιέχει μία στήλη με το μοναδικό αναγνωριστικό του στιχουργού lyricist_id, το οποίο θα χρησιμοποιηθεί σε επόμενες αιτήσεις και άλλες 3 στήλες που θα περιέχουν το όνομα του στιχουργού, τον αριθμός των στίχων και τις προβολές του.

Για να βρούμε το μοναδικό αναγνωριστικό που υπάρχει στον πηγαίο κώδικα καθώς και την ετικέτα που μας δίνει τον πίνακα χρησιμοποιήσαμε την λειτουργία inspect element του firefox. Αφού εντοπίσαμε τις επίμαχες ετικέτες, χτίσαμε τον κώδικα με βάση αυτές προκειμένου να συλλέξουμε τις πληροφορίες που μας ενδιαφέρουν. Συγκεκριμένα με την ετικέτα table και τις λέξεις κλειδιά keys εντοπίζουμε μέσα στον πηγαίο κώδικα τον πίνακα των στιχουργών. Στις κάτω εικόνες 1.3 και 1.4 φαίνεται ότι ο πίνακας που μας ενδιαφέρει, έχει ως παραμέτρους τις τιμές{width:80%, cellpanding: 0}. Τα γνωρίσματα αυτά αντικαθιστώνται στην θέση keys του κώδικα στην μέθοδο create_table. Το κομμάτι του προγράμματος που εντοπίζει τον πίνακα δίνεται στην γραμμή 56. Στο σημείο αυτό φαίνεται και ο τρόπος με τον οποίο αναζητούμε τα μοναδικά αναγνωριστικά lyricist_id των στιχουργών. Συγκεκριμένα αφού εντοπίσουμε τον πίνακα, ζητάμε από την beautiful soup να βρει μέσα στον πίνακα όλες τις ετικέτες a με λέξη κλειδί το href που υποδηλώνει κάποιο σύνδεσμο μαζί με την παραμετροποίηση το href να περιέχει την έκφραση info=Lyrics&act=index&sort=alpha&lyricist_id (γραμμή κώδικα 56) που μας οδηγεί στη εξαγωγή του μοναδικού αναγνωριστικού με το οποίο οι στιχουργοί είναι αποθηκευμένη στη βάση δεδομένων της ιστοσελίδας (γραμμές κώδικα 58-63). Τέλος στις γραμμές κώδικα 66-81 φαίνεται ο τρόπος με τον οποίο συλλέγουμε από συγκεκριμένες στήλες του πίνακα (cell_list=[1,2,5]) τις πληροφορίες που μας ενδιαφέρουν . Το πρόγραμμα ολοκληρώνεται γράφοντας σε ένα αρχείο csv τις στήλες του πίνακα μαζί με την στήλη του μοναδικού αναγνωριστικού.

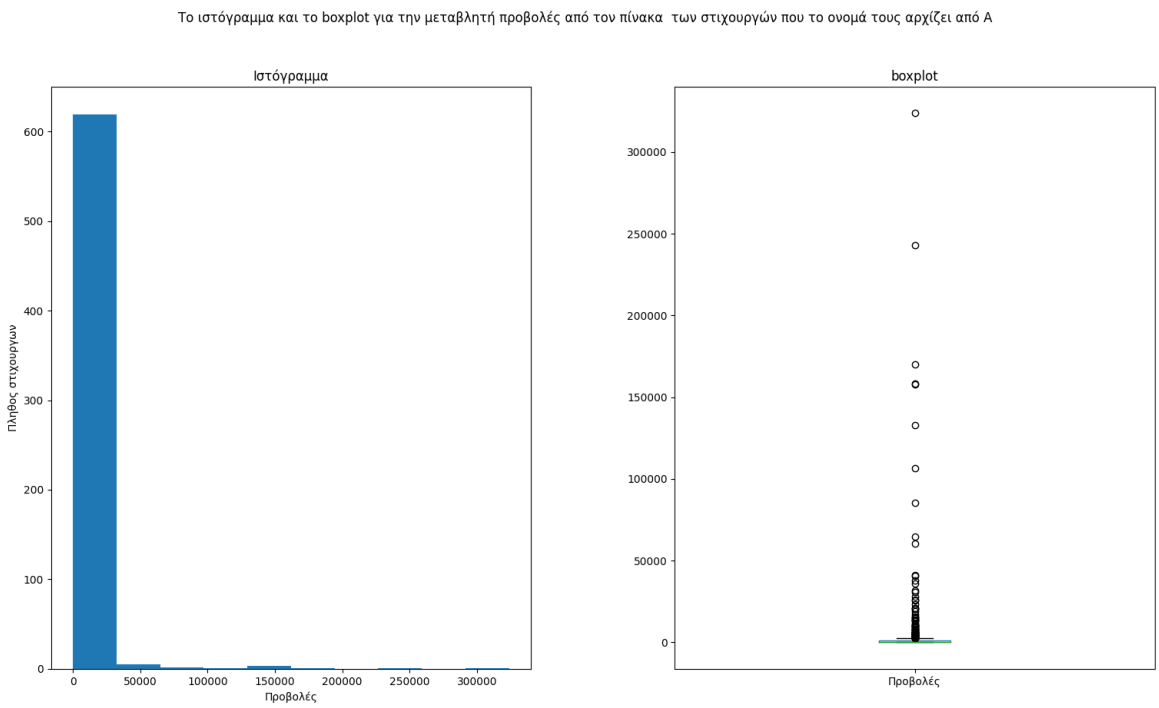


Σχήμα 6.3: Οι ετικέτες του πηγαίου κώδικα που αντιστοιχούν στον πίνακα των στιχουργών



Σχήμα 6.4: Οι ετικέτες του πηγαίου κώδικα που αντιστοιχούν στο μοναδικό αναγνωριστικό των στιχουργών,

Αφού ολοκληρώσουμε το κατέβασμα των πινάκων των στιχουργών διαλέγουμε τους πιο δημοφιλείς. Για να το κάνουμε αυτό πρέπει πρώτα να μεταφράσουμε την δημοφιλία στην γλώσσα της στατιστικής. Σύμφωνα με τα στατιστικά μέτρα που γνωρίζουμε καταλαβαίνουμε ότι δημοφιλής είναι αυτός που απέχει πολύ από το μέσο όρων των προβολών των στιχουργών με τους οποίου ανήκει στον ίδιο πίνακα. Άρα πούμε ότι δημοφιλής είναι ο στιχουργός που αποτελεί ακραία τιμή για την κατανομή της τυχαία μεταβλητής προβολές. Για καταλάβουμε το είδος της κατανομής που ακολουθεί η τυχαία μεταβλητή προβολές αλλά και τι ακραίες τιμές έχει, αν έχει θα δημιουργήσουμε σε πρώτη φάση το ιστόγραμμα και το boxplot των τιμών της. Μετά την δημιουργία ιστογραμμάτων και boxplots για την μεταβλητή ΄προβολές΄ σε κάθε πίνακα είδαμε ότι η μεταβλητή σε πολλές περιπτώσεις ακολουθεί μία κατανομή με πολύ μακριά ουρά καθώς και πολλές ακραίες παρατηρήσεις. Για το λόγο αυτό χρησιμοποιώντας τα z-scores των προβολών για τις εγγραφές κάθε πίνακα, μπορέσαμε να βρούμε εκείνους του στιχουργούς που αποτελούν ακραίες παρατηρήσεις.



Σχήμα 6.5: Στο σχήμα απεικονίζεται το ιστόγραμμα και το boxplot της μεταβλητής προβολές των στιχουργών που το επώνυμο τους ξεκινάει από Α.

Επαναλαμβάνοντας την ίδια διαδικασία για όλα τα γράμματα του αλφαβήτου καταφέραμε με αυτόν τον τρόπο να βρούμε όλους τους δημοφιλείς στιχουργούς όλων των γραμμάτων. Παρακάτω παρατίθεται ο πίνακας των δημοφιλών στιχουργών καθώς και ο κώδικας βάση του οποίου υλοποιήσαμε την διαδικασία.

Lyricist_Scraping.py

```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
import os
from scipy import stats

path=os.getcwd()+'/Lyricists_From_A/Lyricist_Table.csv'
table=pd.read_csv(path)
data=table['Προβολές']
mean_value = np.mean(data)
std = np.std(data)
z = np.abs(stats.zscore(data))
threshold = 3
pointer = np.where(z >=3)
floor = pointer[0][len(pointer[0]) - 1]
table_2 = table.loc[table['Προβολές'] >= data[floor]]
print(table_2)
fig ,axes=plt.subplots(1,2)
table.hist('Προβολές',bins=10,grid=False,ax=axes[0])
axes[0]. set_title ('Ιστόγραμμα')
axes[0]. set_xlabel ('Προβολές')
axes[0]. set_ylabel ('Πληθος στιχουργων')
table.boxplot(column='Προβολές',grid=False,ax=axes[1])
axes[1]. set_title ('boxplot')
fig . suptitle ('Το ιστόγραμμα και το boxplot για την μεταβλητή προβολές από τον πίνακα των στιχουργών που το ονομά τους αρχίζει από Α')
plt .show()
```

	Lyricist_id	Στιχουργός	Αριθμός στίχων	Προβολές
1	lyricist_id=72	Αλκαίος Άλκης	224	323718
2	lyricist_id=1128	Αγγελάκας Γιάννης	159	243026
3	lyricist_id=392	Αλεξίου Χαρούλα	76	169981
4	lyricist_id=63	Αλιβιζάτος Σαράντης	280	158068
5	lyricist_id=110	Άσιμος Νικόλας	89	157764
6	lyricist_id=625	Αποστολάκης Δημήτρης	122	133109
7	lyricist_id=73	Ανδρικάκης Αντώνης	117	106680
8	lyricist_id=361	Αξιώτης Άγγελος	318	85419
9	lyricist_id=383	Βραχάλη Ελεάνα	245	454764
10	lyricist_id=4	Βίρβος Κώστας	705	305061
11	lyricist_id=23	Βαμβακάρης Μάρκος	187	243271
12	lyricist_id=64	Βασιλειάδης Χαράλαμπο	290	170920
13	lyricist_id=489	Βαξαβανέλης Νίκος	220	138384
14	lyricist_id=366	Βλαχοπούλου Όλγα	75	100054
15	lyricist_id=421	Βουγιατζής Τάσος	250	78954
16	lyricist_id=210	Βάρναλης Κώστας	259	78878
17	lyricist_id=745	Βήτα Κωνσταντίνος	122	68924
18	lyricist_id=92	Γκάτσος Νίκος	374	604884
19	lyricist_id=360	Γιαννατσούλια Ελένη	452	487014
20	lyricist_id=309	Γιαννόπουλος Βασίλης	435	393065
21	lyricist_id=311	Γιατράς Σπύρος	526	265738
22	lyricist_id=358	Γερμανού Ναταλία	336	222615
23	lyricist_id=705	Γεροθόδωρου Βίκυ	288	209547
24	lyricist_id=447	Γρίτσης Νίκος	163	178443
25	lyricist_id=120	Γκανάς Μιχάλης	179	141619
26	lyricist_id=884	Γονίδης Σταμάτης	294	132901
27	lyricist_id=139	Γράψας Φίλιππος	133	106498
28	lyricist_id=333	Δρούτσα Εύη	557	222169
29	lyricist_id=5	Δασκαλόπουλος Άκος	308	160673
30	lyricist_id=735	Δόξας Γιάννης	93	93705
31	lyricist_id=224	Δεληβοριάς Φοίβος	101	93567
32	lyricist_id=75	Δαβαράκης Άρης	117	93118
33	lyricist_id=181	Δημοπούλου Λίνα	129	67806
34	lyricist_id=774	Δήμας Βασίλης	145	66690
35	lyricist_id=15	Ελευθερίου Μάνος	513	391313
36	lyricist_id=54	Ελύτης Οδυσσέας	153	318581
37	lyricist_id=889	Ευαγγελάτος Γεράσιμος	122	271948
38	lyricist_id=253	Ζιώγα Ελένη	75	162431
39	lyricist_id=182	Ζούδιαρης Νίκος	102	139502
40	lyricist_id=677	Ζερβουδάκης Δημήτρης	71	56222
41	lyricist_id=254	Ζιώγαλας Νίκος	63	43886
42	lyricist_id=96	Ζήκας Γιώργος	101	42785
43	lyricist_id=874	Ημισκούμπρια	42	52469
44	lyricist_id=319	Θεοφάνους Γιώργος	259	359439
45	lyricist_id=44	Θεοδωράκης Μίκης	77	80821
46	lyricist_id=265	Θωμαΐδου Τασούλα	174	61865
47	lyricist_id=539	Θεοδώρου Φώντας	136	50181
48	lyricist_id=229	Ιωαννίδης Αλκίνοος	81	267879
49	lyricist_id=156	Ιωάννου Οδυσσέας	193	198175
50	lyricist_id=6	Ιατρόπουλος Δημήτρης	256	79020
51	lyricist_id=339	Καρβέλας Νίκος	562	291081
52	lyricist_id=101	Καββαδίας Νίκος	62	227254
53	lyricist_id=371	Κραουνάκης Σταμάτης	204	179955
54	lyricist_id=440	Κριεζή Μαριανίνα	104	151511
55	lyricist_id=115	Κατσιμίχας Πάνος	105	149569
56	lyricist_id=722	Κατσιμίχας Χάρης	79	144338
57	lyricist_id=161	Κωνσταντινίδης Βαγγέλη	253	139905
58	lyricist_id=168	Κατσούλης Ηλίας	289	136745
59	lyricist_id=65	Κολοκοτρώνης Χρήστος	309	120326
60	lyricist_id=74	Καραλής Γιάννης	328	107889
61	lyricist_id=9	Καλδάρας Απόστολος	198	105250
62	lyricist_id=166	Καρασούλος Παρασκευά	134	100625
63	lyricist_id=500	Καμπανέλλης Ιάκωβος	51	93542
64	lyricist_id=230	Καρράς Βασίλης	154	90152
65	lyricist_id=29	Κοφινιώτης Κώστας	185	80284
66	lyricist_id=579	Κορνάρος Βιτσέντζος	39	78944
67	lyricist_id=568	Κηλαηδόνης Λουκιανός	55	73770
68	lyricist_id=285	Καρυωτάκης Κώστας	109	66871
69	lyricist_id=3	Κουγιουμτζής Σταύρος	50	63968
70	lyricist_id=71	Λειβαδίτης Τάσος	333	157355
71	lyricist_id=432	Λειβαδάς Κώστας	100	113334
72	lyricist_id=231	Λυμπερόπουλος Ηλίας	198	84704
73	lyricist_id=397	Λαζόπουλος Λάκης	44	69808
74	lyricist_id=687	Λιόντος Γιάννης	173	62324
75	lyricist_id=256	Μωραΐτης Νίκος	377	376124
76	lyricist_id=282	Μουκίδης Γιώργος	192	262389
77	lyricist_id=272	Μάλαμας Σωκράτης	76	194921
78	lyricist_id=212	Μουσαφίρης Τάκης	499	156949
79	lyricist_id=2	Μητσάκης Γιώργος	354	151628
80	lyricist_id=18	Μπουρμπούλης Μιχάλης	219	129303
81	lyricist_id=368	Μπαλτζή Sunny	117	81610
82	lyricist_id=257	Μηλιώκας Γιάννης	84	74562
83	lyricist_id=26	Μάνεσης Κώστας	129	66883
84	lyricist_id=104	Νικολακοπούλου Λίνα	480	586782
85	lyricist_id=293	Ντούμος Κυριάκος	100	82163
86	lyricist_id=226	Νεγρεπόντης Γιάννης	83	56261
87	lyricist_id=403	Νικολάου Φίλιππος	222	51214
88	lyricist_id=145	Ξυδούς Μάνος	102	151517
89	lyricist_id=496	Οικονομίδης Γιώργος	81	42796

Σχήμα 6.6: Το πρώτο μέρος του πίνακα με τους πιο διάσημους στιχουργούς σύμφωνα με τους επισκέπτες της ιστοσελίδας stixoi.info.

	Lyricist_id	Στιχουργός	Αριθμός στίχων	Προβολές
91	lyricist_id=214	Οικονόμου Τάσος	111	34996
92	lyricist_id=37	Παραδοσιακό	2453	1652878
93	lyricist_id=1	Παπαδόπουλος Λευτέρης	806	758422
94	lyricist_id=198	Παπακωνσταντίνου Θανά	134	408469
95	lyricist_id=12	Πυθαγόρας	770	373060
96	lyricist_id=236	Πάριος Γιάννης	343	258930
97	lyricist_id=433	Παυλίδης Παύλος	132	250416
98	lyricist_id=4184	Παντελίδης Παντελής	70	205137
99	lyricist_id=146	Πλιάτσικας Φίλιππος	140	190663
100	lyricist_id=219	Πορτοκάλογλου Νίκος	195	190549
101	lyricist_id=10	Παπαγιαννοπούλου Ευτυ	217	188152
102	lyricist_id=21	Πάνου Άκης	196	181792
103	lyricist_id=659	Παπανικολάου Θάνος	278	178415
104	lyricist_id=752	Παππάς Αντώνης	285	119668
105	lyricist_id=195	Πασχαλίδης Μιλτιάδης	49	117083
106	lyricist_id=289	Παπαδόπουλος Βασίλης	263	100857
107	lyricist_id=148	Περίδης Ορφέας	74	89751
108	lyricist_id=511	Πανούσης Τζίμης	60	81544
109	lyricist_id=53	Ρασούλης Μανώλης	275	231765
110	lyricist_id=14	Ρίτσος Γιάννης	353	213981
111	lyricist_id=709	Ρόκκος Στέλιος	186	175135
112	lyricist_id=527	Ρακιντζής Μιχάλης	191	72876
113	lyricist_id=801	Ρούσση Ρεβέκκα	130	65127
114	lyricist_id=97	Σαββόπουλος Διονύσης	139	241301
115	lyricist_id=89	Σακελλάριος Αλέκος	148	208665
116	lyricist_id=157	Σούσης Ισαάκ	96	137278
117	lyricist_id=528	Σιγανός Χριστόδουλος	163	103191
118	lyricist_id=445	Σιδηρόπουλος Παύλος	76	97842
119	lyricist_id=472	Σεφέρης Γιώργος	234	82508
120	lyricist_id=429	Στόκας Μπάμπης	60	74658
121	lyricist_id=220	Σπανουδάκης Σταμάτης	91	65043
122	lyricist_id=661	Σαρρής Νίκος	136	60415
123	lyricist_id=459	Σκαρβέλης Κώστας	135	58983
124	lyricist_id=19	Τσιτσάνης Βασίλης	441	421407
125	lyricist_id=7	Τσώτου Σώτια	420	191401
126	lyricist_id=52	Τριπολίτης Κώστας	113	132424
127	lyricist_id=275	Τουρνάς Κώστας	243	100081
128	lyricist_id=264	Τούντας Πάνος	131	90707
129	lyricist_id=113	Τσακνής Διονύσης	117	87684
130	lyricist_id=946	Τσάφας Δημήτρης	158	78627
131	lyricist_id=578	Τραϊφόρος Μίμης	70	53942
132	lyricist_id=412	Υπόγεια Ρεύματα	88	56459
133	lyricist_id=316	Φοίβος	466	513458
134	lyricist_id=357	Φιλίππου Ηλίας	1120	421583
135	lyricist_id=327	Φαλάρας Πάνος	793	294923
136	lyricist_id=183	Φάμελλος Μανώλης	162	78073
137	lyricist_id=242	Φασουλάς Κώστας	136	71344
138	lyricist_id=95	Χατζιδάκις Μάνος	111	138120
139	lyricist_id=11	Χριστοδούλου Δημήτρης	166	117589
140	lyricist_id=61	Χαψιάδης Λευτέρης	259	90058
141	lyricist_id=25	Χιώτης Μανώλης	239	82034
142	lyricist_id=400	Χικμέτ Ναζίμ	31	50230
143	lyricist_id=4274	Χατζηφραγκέτα	53	47743
144	lyricist_id=215	Ψυχογιός Κώστας	186	50936

Σχήμα 6.7: Το δεύτερο μέρος του πίνακα με τους πιο διάσημους στιχουργούς.Συνολικά τα z-scores αξιολόγησα 142 στιχουργούς ως δημοφιλείς.

Στην συνέχεια με το αναγνωριστικό του στιχουργού φπου βρήκαμε από τον προηγούμενο κώδικα συνεχίζουμε την εξαγωγή δεδομένων ώστε να κατεβάσουμε τον πίνακα που περιέχει τα τραγούδια. Ο τρόπος που

το πετυχαίνουμε φαίνεται στο παρακάτω κομμάτι κώδικα.

Lyrics_Table_Scraping.py

```
import requests
from bs4 import BeautifulSoup as bs
import re
import lxml
import os
import csv
import chardet
import time
import random


def countdown(t):
    while t > 0:
        min, secs = divmod(t, 60)
        timer = '{:02d}:{:02d}'.format(min, secs)
        print('\r{0}'.format(timer), end='')
        time.sleep(1)
        t = t - 1


class Fetch_Lyricist_Table:
    html = None
    search_keys = None
     artist = None
    dir_path = None
     song_table_file = None
     lyricist_id = None

    # Takes the name of he artist and path we want the artist directory to be made
    # and it construct the path for the dir of the artist
    def __init__(self, artist, path, lyricist_id):
        self.artist = artist
        self.dir_path = path + '/' + artist
        self. lyricist_id = lyricist_id

    # Returns the artist's directory path name
    def get_dir_path(self):
        return self.dir_path

    # Makes a directory for the artist if it doesen't already exist or else
    # the construcrtion fails
    def make_lyricist_dir(self):
        try:
            os.mkdir(self.dir_path)
        except OSError:
            print("Creation of the directory %s failed" % self.dir_path)
        else:
            print("Successfully created the directory %s " % self.dir_path)

    # Gets the url and the number of the lyrics. If the number of songs are under 100
    # there is only one table to download. If the numbers of songs are over 100 the scraper
    # downloads multiple files.

    def get_source_code(self, lyrics_no):
        if lyrics_no <= 100:
            print('The number of song is under 100. Only 1 table will be downloaded')
            url = 'https://www.stixoi.info/stixoi.php?info=Lyrics&act=index&sort=alpha&' + str(self.lyricist_id)
            print('Sending the request for artist:', self.artist)
            request = requests.get(url, timeout=5)
            if request.status_code == 200:
                print(request.status_code)
                respond_time_1 = float(round(request.elapsed.total_seconds(), 2))
                print('Time elapses is:', respond_time_1)
                soup = bs(request.content, features="lxml")
                html_path_1 = self.dir_path + '/page_source_code.txt'
                html = open(html_path_1, 'w')
                html.write(str(soup))
                html.close()
                print('Source code saved to file:' + html_path_1)
                self.html = html
                print('Source code text file has been successfully created')
```

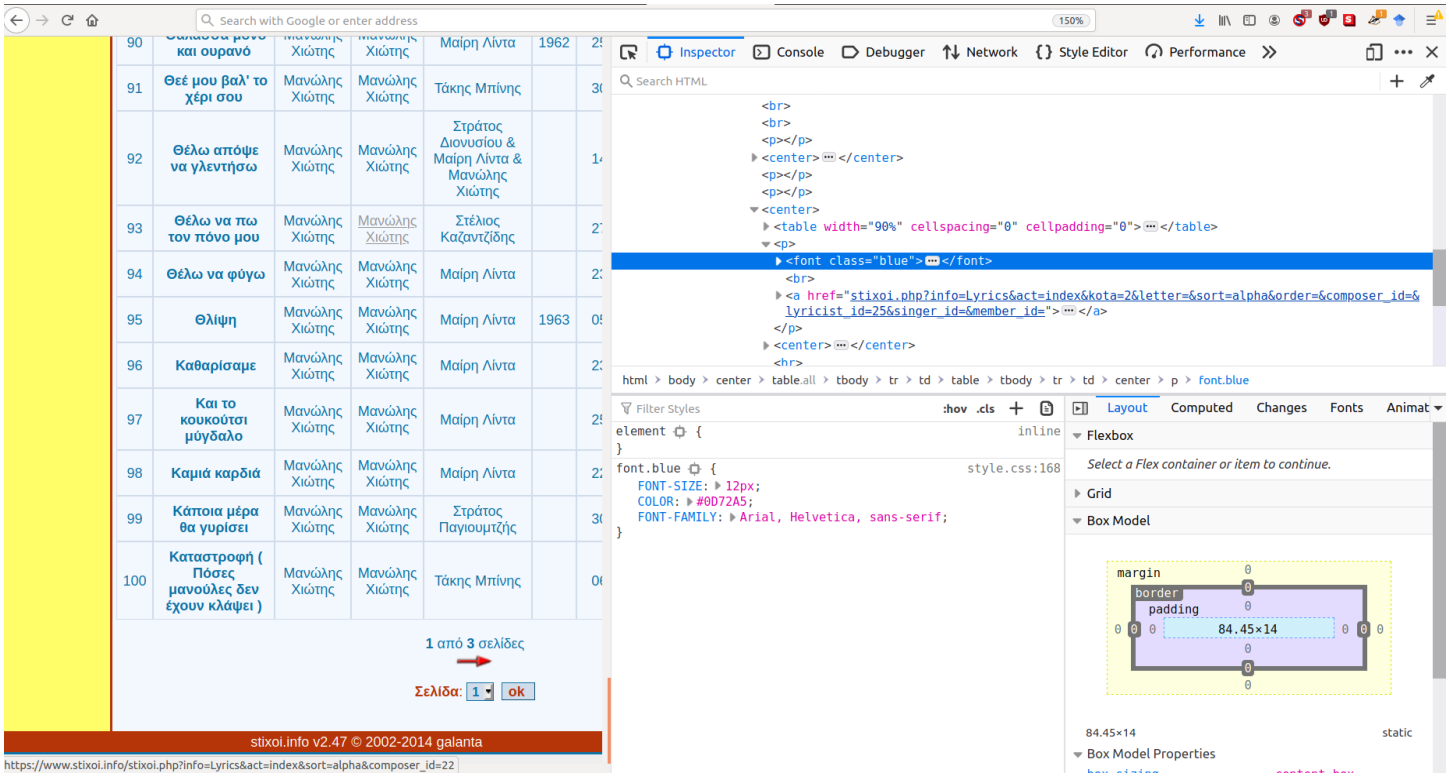
```
72                  if respond_time_1 > 1.0:
73                      wait_time = int(respond_time_1 * random.uniform(300, 420))
74                      print('Time out is:')
75                      countdown(wait_time)
76                  else:
77                      wait_time = int(respond_time_1 * random.uniform(120, 130))
78                      print('Time left for the next request')
79                      countdown(wait_time)
80          if lyrics_no > 100:
81              print('number of lyrics exceeded 100, multiple tables will be downloaded')
82              url_2 = 'https://www.stixoi.info/stixoi.php?info=Lyrics&act=index&sort=alpha&' + str(self.lyricist_id)
83              print('Sending the request for page 1, for artist:', self.artist)
84              request_2 = requests.get(url_2, timeout=5)
85              if request_2.status_code == 200:
86                  print(request_2.status_code)
87                  respond_time_2 = float(round(request_2.elapsed.total_seconds(), 2))
88                  print('Time elapses is:', respond_time_2)
89                  soup = bs(request_2.content, features="lxml")
90                  html_path_2 = self.dir_path + '/source_code_page_1.txt'
91                  html_2 = open(html_path_2, 'w')
92                  html_2.write(str(soup))
93                  html_2.close()
94                  print('File' + html_path_2 + 'was successfully created')
95                  fonts = soup.find_all('font', {'class': 'blue'})
96                  font = fonts[0]
97                  s = re.findall('\d+', font.text)
98                  pages = int(s[1])
99                  if respond_time_2 > 1.0:
100                     wait_time = int(respond_time_2 * random.uniform(300, 420))
101                     print('Time out is:')
102                     countdown(wait_time)
103                 else:
104                     wait_time = int(respond_time_2 * random.uniform(120, 130))
105                     print('Time left for the next request')
106                     countdown(wait_time)
107                 for page in range(1, pages):
108                     url_3 = 'https://www.stixoi.info/stixoi.php?info=Lyrics&act=index&kota=' + str(
109                         page + 1) + '&letter=&sort=alpha&order=&composer_id=&' + str(
110                         self.lyricist_id) + '&singer_id=&member_id='
111                     print('Sending request for artist:' + ' ' + self.artist + ', ' + 'page' + str(page))
112                     request_3 = requests.get(url_3, timeout=5)
113                     if request_3.status_code == 200:
114                         print(request_3.status_code)
115                         respond_time_3 = float(round(request_3.elapsed.total_seconds(), 2))
116                         print('Time elapses is:', respond_time_3)
117                         soup = bs(request_3.content, features="lxml")
118                         html_path_3 = self.dir_path + '/source_code_page_' + str(page + 1) + '.txt'
119                         html_3 = open(html_path_3, 'w')
120                         html_3.write(str(soup))
121                         html_3.close()
122                         print('File' + html_path_3 + 'was successfully created')
123                         if respond_time_3 > 1.0:
124                             wait_time = int(respond_time_3 * random.uniform(300, 420))
125                             print('Time out is:')
126                             countdown(wait_time)
127                         else:
128                             wait_time = int(respond_time_3 * random.uniform(120, 130))
129                             print('Time left for the next request')
130                             countdown(wait_time)
```

Με τον παραπάνω κώδικα αρχικά κατασκευάζουμε ένα φάκελο (γραμμές 31-48) με το όνομα του στιχουργού μέσα στον οποίο θα αποθηκεύσουμε το πηγαίο κώδικα που θα περιέχει τον πίνακα με τα τραγούδια. Επειδή αντίθετα με του στιχουργούς οι πίνακες των τραγουδιών όταν οι εγγραφές ξεπερνάνε τις 100 χωρίζονται σε σελίδες πρέπει να ελέγξουμε τον αριθμό των τραγουδιών πριν στείλουμε την αίτηση στον server. Για να βρούμε το πλήθος των τραγουδιών εξάγουμε από τον πίνακα των στιχουργών που φτιάξαμε στο προηγούμενο βήμα τον αριθμό των στίχων.

Παρατηρήστε ότι στην αρχή του κώδικα έχουμε ορίσει μία συνάρτηση countdown. Αυτή δεν είναι τίποτα άλλο παρά ένας μετρητής που δείχνει το χρόνο που απομένει για το επόμενο request. Ο κώδικας που "τραβάει" τους πίνακες των τραγουδιών είναι χωρισμένος σε δύο περιπτώσεις (γραμμή κώδικα 55 και 80) με βάση αν τα τραγούδια υπερβαίνουν τα 100.

Ο κώδικας αρχίζει με την πρώτη περίπτωση όπου οι στίχοι είναι λιγότεροι από 100 (γραμμή 55) κατασκευάζοντας το url στο οποίο θα γίνει η αίτηση. Πάλι για να καταλάβουμε την δομή του url παρατηρούμε το πλαίσιο διευθύνσεων του φυλλομετρητή που χρησιμοποιούμε. Αν περιηγηθούμε στις σελίδες διάφορων στιχουργών θα δούμε ότι το μοναδικό δυναμικό κομμάτι της διεύθυνσης είναι το αναγνωριστικό lyricist_id= που εξάγαμε στη προηγούμενη διαδικασία. Για του λόγου το αληθές μπορείτε μόνοι σας να επιβεβαιώσετε ότι ο Μάρκος Βαμβακάρης έχει αναγνωριστικό lyricist_id=23 ενώ ο Μανώλης Χιώτης το lyricist_id=25. Αρχικά ο το πρόγραμμα ενημερώνει ότι θα κατέβει μόνο μία σελίδα και στέλνει την αίτηση στο server.Όπως και στην προηγούμενη διαδικασία έτσι και σε αυτή αν αργήσει να απαντήσει ο server γίνεται timeout. Αφού βεβαιωθούμε ότι η επικοινωνία με τον server ήταν επιτυχής τυπώνοντας τον κωδικό κατάστασης της αίτησης(if respond.status.code==200) αποθηκεύουμε την ιστοσελίδα στον τρέχοντα κατάλογο σε ένα αρχείο με όνομα "page_source_code.txt". Κάθε φορά που στέλνουμε μία αίτηση αποθηκεύουμε τον χρόνο απάντησης και με βάση αυτό υπολογίζουμε τον χρόνο αναμονής που θα περιμένει το πρόγραμμα για να στείλει την επόμενη αίτηση. Εμείς επιλέξαμε ο χρόνος αναμονής να είναι ένα τυχαίος αριθμός ο οποίο είναι δεκαπλάσιος του χρόνου απάντησης. Δοκιμάσαμε πειραματικά και μικρότερα χρονικά διαστήματα αλλά ο server άρχισε να αυξάνει τον χρόνο απάντησης και καταλήξαμε ότι ο καλύτερος χρόνος αναμονής είναι ανά 30 με 40 περίπου δευτερόλεπτα ενδιάμεσα στις αιτήσεις.Οπότε προτιμήσαμε μεγάλο χρόνο αναμονής αλλά βάλαμε πολλούς crawlers να δουλεύουνε ταυτόχρονα.

Αν το πλήθος των στίχων είναι πάνω από 100 (γραμμή 80) τότε θα πρέπει να κατεβάσουμε μία σειρά σελίδων από τον εξυπηρετητή. Για να βρούμε σε ποιο σημείο του κώδικα αναγράφεται το πλήθος των σελίδων χρησιμοποιούμε πάλι τα εργαλεία του firefox.



Σχήμα 6.8

Βλέπουμε ότι οι σελίδες υπάρχουν στην κλάση font με παράμετρο blue. Ακόμη ακριβώς από κάτω είναι και ο σύνδεσμός που θα χρησιμοποιήσει ο χρήστης για να μεταβεί στη σελίδα νούμερο 2. Το μόνο δυναμικό κομμάτι του συνδέσμου είναι η φράση "kota=2" όπου ο αριθμός αλλάζει ανάλογα τη σελίδα στην οποία θέλουμε να μεταβούμε. Αυτά τα στοιχεία χρησιμοποιούμε αν προσέξετε και στον κώδικα (γραμμες 95-98) προκειμένου να βρούμε τον αριθμό των σελίδων που πρέπει να κατεβάσουμε.

Για κάθε σελίδα στέλνουμε και ένα καινούργιο αίτημα στον server. Ο σύνδεσμος του αιτήματος ανανε-

ώνεται δυναμικά καθώς εξελίσσεται ο βρόγχος "for" (γραμμή 107). Κάθε καινούργια σελίδα πηγαίου κώδικα που κατεβαίνει την αποθηκεύουμε σε ένα αρχείο στον τρέχοντα κατάλογο με όνομα "source_code_page + str(page+1) +'.txt'" (γραμμές 114-121), όπου το str(page+1) δηλώνει τον αύξοντα αριθμό της σελίδας.

Πλέον έχουμε δημιουργήσει φακέλους από όλα τα γράμματα της αλφαβήτου και μέσα σε κάθε φάκελο του αντίστοιχού γράμματος υπάρχουν φάκελοι με τα αρχεία των στιχουργών που αποφασίσαμε να κατεβάσουμε. Ο φάκελος κάθε στιχουργού περιέχει όλους του πηγαίους κώδικες που αφορούν στους στίχους που έχει γράψει. Το προτελευταίο βήμα πριν αρχίσουμε να κατεβάζουμε στίχους είναι να καταφέρουμε να συνθέσουμε όλο τον πηγαίο κώδικα σε ένα csv αρχείο με τις στήλες των πινάκων που μας ενδιαφέρουν καθώς και το αναγνωριστικό song_id με το οποίο είναι αποθηκευμένα τα τραγούδια στην βάση δεδομένων της ιστοσελίδας, βάση του οποίου θα αναζητήσουμε στίχους στο τελικό βήμα. Η διαδικασία αυτή θα γίνει αυτόματα με την χρήση του παρακάτω κώδικα

Create_Song_Table.py

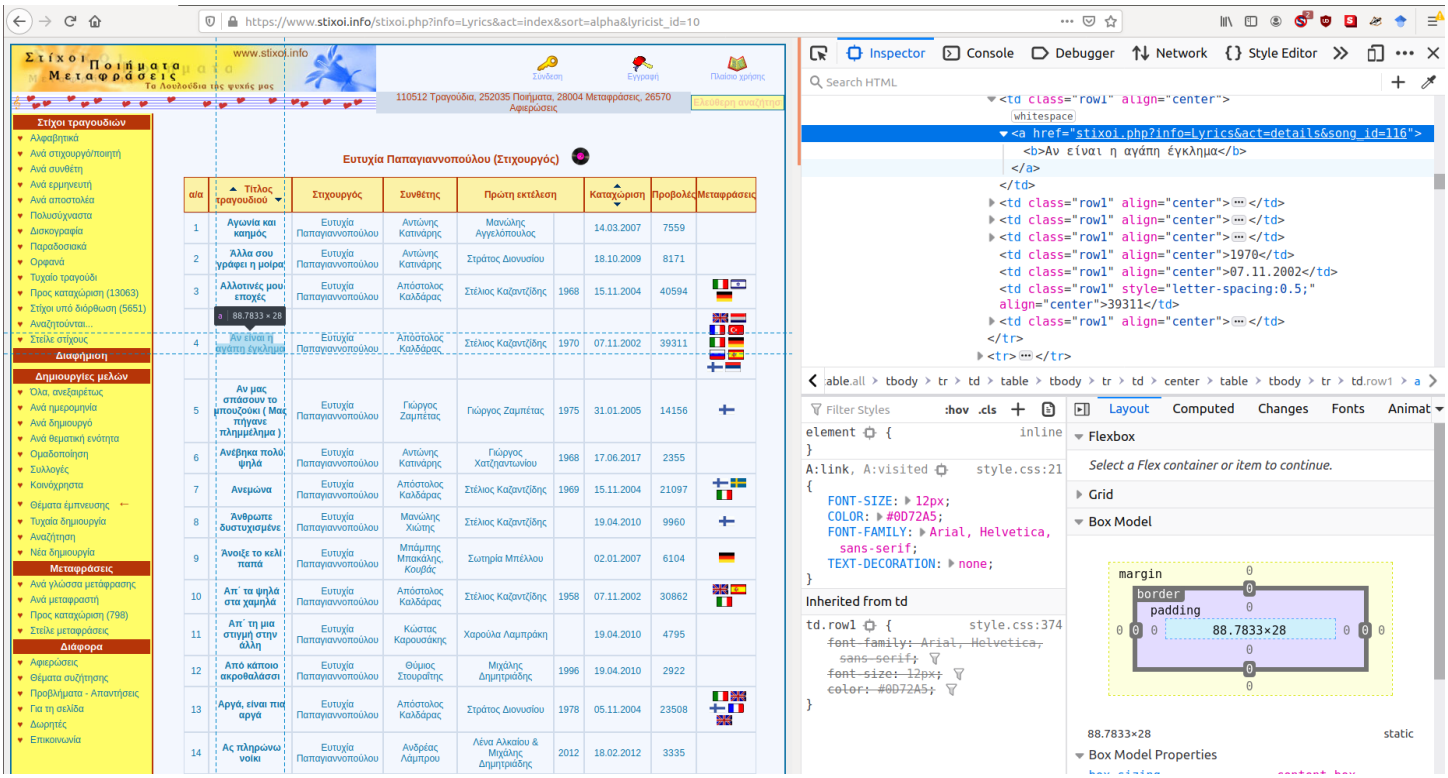
```
1
2  import os
3  import re
4  from bs4 import BeautifulSoup as bs
5  import lxml
6  import csv
7
8  path = os.getcwd()
9   dir_list  = []
10   lyricist_path_list  = []
11 for subdir in os.scandir(path):
12     if os.path.isdir(subdir.path) == True and len(re.findall('Lyricists', str(subdir.path))) == 1:
13         dir_list.append(str(subdir.path))
14 for path in sorted(dir_list):
15     for subdir in os.scandir(path):
16         if os.path.isdir(subdir.path):
17             lyricist_path_list.append(subdir.path)
18
19
20 for lyricist in sorted(lyricist_path_list):
21     page = 0
22     for file in os.scandir(lyricist):
23         if len(re.findall('txt', str(file.path)))==1:
24             print(file.path)
25             source_code = open(file.path, 'r')
26             html = source_code.read()
27             source_code.close()
28             soup = bs(html, features="lxml")
29             table = soup.find('table', {'width': '90%'})
30              links_list  = table('a', href=re.compile('song_id'))
31              id_list  = []
32             for link in links_list:
33                 if 'href' in link.attrs:
34                     s = (str(link.attrs['href']))
35                      id_list.append(s[s.rfind('song_id'):len(s)])
36             table_path = lyricist + '/Song_Table.csv'
37
38              table_file  = open(table_path, mode='a+')
39             print('The file was created succesfully', table_path)
40             table_writer = csv.writer(table_file, delimiter=',', quotechar='"', quoting=csv.QUOTE_MINIMAL)
41             if page == 0:
42                 first_row = ['Song_Id', 'Τίτλος Τραγουδιού', 'Πρώτη εκτέλεση', 'Χρονολογία', 'Προβολές']
43                 table_writer.writerow(first_row)
44             j = 0
45             for row in table.find_all('tr'):
46                 if j > 0:
47                     cells = row.find_all('td')
48                      cell_list  = [1, 4, 5, 7]
49                     row_list = []
50                     for i in range(0, len(cell_list)):
51                         cell = cells[cell_list[i]]
52                         row_list.append(str(cell.text))
53                     row_list.insert(0, id_list[j - 1])
54                     print(row_list)
55
56                     table_writer.writerow(row_list)
```

```
57
58                  j = j + 1
59              table_file . close()
60              print('File'+ table_path+'was successfully created')
61              page = page + 1
```

Η σύνθεση των πινάκων που θα περιέχουν τα στοιχεία των τραγουδιών που χρειαζόμαστε σε ένα ενιαίο csv αρχείο που θα υπάρχει στο φάκελο κάθε καλλιτέχνη ξεκινάει με την εύρεση του μονοπατιού που οδηγεί στον τρέχοντα κατάλογο (os.getcwd()). Στην συνέχει με ένα βρόγχο for εξερευνούμε τον κατάλογο και ψάχνουμε να βρούμε φακέλους (if os.path.isdir(subdir.path) == True) (γραμμή 12) που περιέχουν στο μονοπάτι τους την συμβολοσειρά "Lyricists" (len(re.findall('Lyricists', str(subdir.path))) == 1) (γραμμή 12). Με αυτό τον τρόπο εντοπίζουμε όλα τα μονοπάτια των φακέλων που περιέχουν τους στιχουργούς το επίθετο των οποίων ξεκινάει από ένα συγκεκριμένο γράμμα. Στην συνέχεια διατρέχουμε τα μονοπάτια των φακέλων της προηγούμενης ενότητας με έναν ακόμη βρόγχο for (γραμμή 14) προκειμένου να βρούμε τα μονοπάτια που οδηγούν στου φακέλους των καλλιτεχνών (for subdir in os.scandir(path))(γραμμή 15). Ολοκληρώνοντας αποθηκεύουμε το μονοπάτι που οδηγεί στον φάκελο του κάθε καλλιτέχνη σε μία λίστα (lyricist_path_list.append(subdir.path)) (γραμμή 17).

Παίρνοντας τη λίστα που δημιουργήσαμε με την προηγούμενη διαδικασία, διατρέχουμε τον φάκελο του κάθε καλλιτέχνη με ένα βρόγχο for (for file in os.scandir(lyricist):) (γραμμή 22) προκειμένου να βρούμε τα αρχεία που περιέχουν τον πηγαίο κώδικα για τους πίνακες των στίχων. Κάθε αρχείο πηγαίου κώδικα το αναλύουμε με την Beautiful Soup (soup = bs(html, features="lxml")) (γραμμή 28) και χρησιμοποιώντας την ετικέτα "table" με τα κλειδιά 'width': '90%') εξάγουμε τον πίνακα των στίχων της συγκεκριμένης σελίδας του πηγαίου κώδικα. Εν συνεχεία από τον πίνακα βρίσκουμε όλους του συνδέσμους που περιέχουν το αναγνωριστικό των στίχων "song_id" (links_list = table('a', href=re.compile('song_id'))) (γραμμή 30), το οποίο εξάγεται και αποθηκεύεται σε μία λίστα . Ελέγχουμε αν μελετάμε την πρώτη σελίδα πηγαίου κώδικα (γραμμή 41) ώστε η πρώτη γραμμή του αρχείου "Song_Table.csv να είναι οι επικεφαλίδες των στηλών του πίνακα.

Η διαδικασία ολοκληρώνεται δημιουργώντας γραμμές που περιέχουν ως πρώτο στοιχείο το αναγνωριστικό "song_id" (row_list.insert(0, id_list[j - 1])) (γραμμή 53) ενώ τα υπόλοιπα στοιχεία του πίνακα είναι οι πληροφορίες που επιλέξαμε να εξάγουμε από συγκεκριμένες στήλες (cell_list = [1, 4, 5, 7]) (γραμμή 48).



Σχήμα 6.9: Στην εικόνα φαίνεται ο σύνδεσμος που υπάρχει αποθηκευμένος στον πίνακα των στίχων και περιέχει το μοναδικό αναγνωριστικό "song_id" του τραγιουδιού.

Το τελευταίο βήμα πριν τη συλλογή στίχων είναι να φτιάξουμε έναν δεύτερο αρχείο csv από τον πίνακα που δημιουργήσαμε με την προηγούμενη διαδικασία ο οποίο θα περιέχει τα 30 πιο προβεβλημένα τραγούδια του κάθε στιχουργού. Ο πίνακας αυτός θα ονομαστεί Song_Table_30.csv

Table_30.py

```
1
2  import os
3  import pandas as pd
4  import re
5
6  path = os.getcwd()
7   lyricist_from_list =[]
8
9  for subdir in os.scandir(path):
10      if os.path.isdir(subdir.path) == True and len(re.findall('Lyricists', str(subdir.path))) == 1:
11          lyricist_from_list .append(str(subdir.path))
12
13 for i in range(0, len( lyricist_from_list )):
14     name_path=sorted(lyricist_from_list)[i]
15      lyricist_list =[]
16     for lyricist in os.scandir(name_path):
17         if os.path.isdir( lyricist ) == True:
18             lyricist_list .append(lyricist.path)
19
20
21     for lyricist_path in sorted( lyricist_list ):
22         for lyricist_table in os.scandir( lyricist_path ):
23             if len(re. findall ('Song_Table.csv',str( lyricist_table )))==1:
24                 table = pd.read_csv( lyricist_table .path)
25                 table.sort_values('Προβολές', inplace=True, ascending=False)
26                 table.drop_duplicates(subset=['Τίτλος Τραγουδιού'], keep='first', inplace=True)
27                 table_30 = table.head(30)
28                 save_path=lyricist_path+'/Song_Table_30.csv'
29                 table_30.to_csv(save_path)
30                 print('Song Table saved in', save_path)
```

Ο κώδικας αρχίζει βρίσκοντας τα μονοπάτια προς το φάκελο του κάθε στιχουργού. Η διαδικασία είναι πανομοιότυπη με αυτήν που ακολουθήσαμε κατά την δημιουργία του αρχείου Song_Table.csv. Αφού έχουμε διαθέσιμο το μονοπάτι του κάθε φακέλου με ένα βρόγχο for το επισκεπτόμαστε και ανοίγουμε το αρχείο Song_Table.csv που υπάρχει στο φάκελο (len(re.findall('Song_Table.csv',str(lyricist_table)))==1:) (γραμμή 22). Χρησιμοποιώντας την βιβλιοθήκη pandas της python ανοίγουμε το αρχείο που περιέχει τον πίνακα των τραγουδιών (γραμμή 23), ύστερα τον ταξινομούμε με βάση τη στήλη "Προβολές" (γραμμή 24) και διαγράφουμε τα διπλότυπα (γραμμή 25). Τέλος αποθηκεύουμε τις 30 πρώτες γραμμές του ταξινομημένου πίνακα που δεν περιέχει διπλότυπα σε ένα αρχείο csv με όνομα Song_Table_30.csv (γραμμή 28).

Αφού επιλέξαμε τα 30 δημοφιλέστερα τραγούδια του κάθε στιχουργού είμαστε έτοιμοι να συλλέξουμε τον πηγαίο κώδικα που περιέχει τους στίχους του κάθε τραγουδιού. Τα αρχεία που περιέχουν τους στίχους θα δημιουργηθούν σε δύο στάδια. Πρώτα κατεβάζουμε και αποθηκεύουμε τον πηγαίο κώδικα που τους περιέχει και ύστερα χρησιμοποιώντας την Beautiful Soup επεξεργαζόμαστε τους αντίστοιχους ληφθέντες πηγαίους κώδικες ώστε να τους κατεβάσουμε.

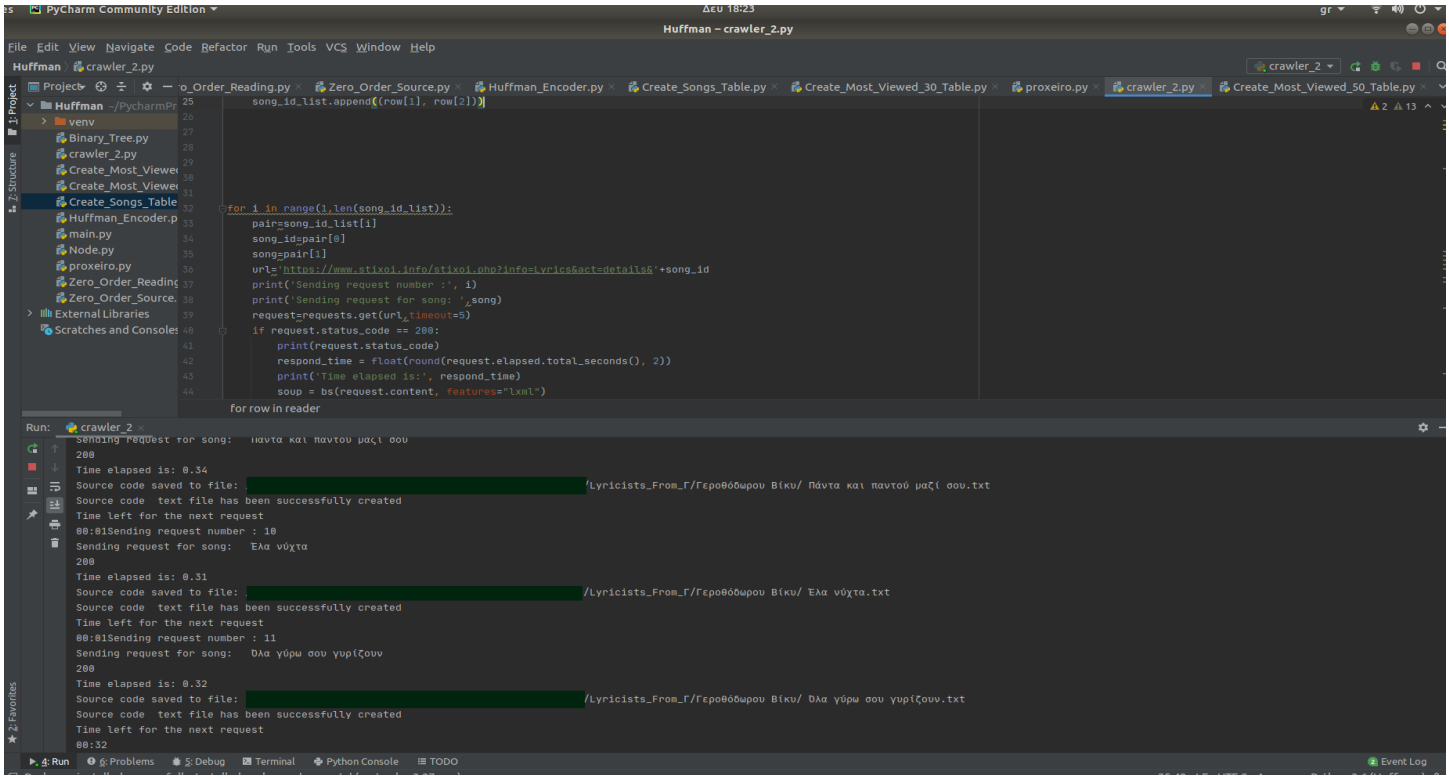
Song_Scraping.py

```
1  import requests
2  from bs4 import BeautifulSoup as bs
3  import re
4  import lxml
5  import os
6  import csv
7  import time
8  import random
9
10
11 def countdown(t):
12     while t > 0:
13         min, secs = divmod(t, 60)
14         timer = '{:02d}:{:02d}'.format(min, secs)
15         print('\r{0}'.format(timer), end='')
16         time.sleep(1)
17         t = t - 1
18
19
20 song_30 = 'hidden path Lyricists_From_Ψ/Ψυχογιός Κώστας/Song_Table_30.csv'
21  song_id_list = []
22  file = open(song_30, 'r')
23 reader = csv.reader( file , delimiter=',')
24 for row in reader:
25      song_id_list .append((row[1], row[2]))
26
27 for i in range(1,len( song_id_list )):
28     pair=song_id_list [i]
29     song_id=pair[0]
30     song=pair[1]
31     url='https://www.stixoi.info/stixoi .php?info=Lyrics&act=details&'+song_id
32     print('Sending request number :', i)
33     print('Sending request for song: ',song)
34     request=requests.get(url,timeout=5)
35     if request.status_code == 200:
36         print(request.status_code)
37         respond_time = float(round(request.elapsed.total_seconds (), 2))
38         print('Time elapsed is:', respond_time)
39         soup = bs(request.content, features="lxml")
40         lyric_path = 'hidden path Lyricists_From_Ψ/Ψυχογιός Κώστας' + '/' + str(song) + '.txt'
41         html = open(lyric_path, 'w')
42         html.write(str(soup))
43         html.close()
44         print('Source code saved to file : ' + lyric_path)
45         print('Source code text file has been successfully created')
46         if respond_time > 1.0:
47             wait_time = int(respond_time * random.uniform(300, 420))
48             print('Time out is:')
49             countdown(wait_time)
50         else:
51             wait_time = int(respond_time * random.uniform(120, 130))
52             print('Time left for the next request')
```

```
53              countdown(wait_time)
54
55  os.system('spd-say "Your program has finished"')
```

Η διαδικασία αρχίζει δίνοντας στο πρόγραμμα το μονοπάτι στο οποίο βρίσκεται το αρχείο Song_Table_30.csv το οποίο περιέχει τα 30 δημοφιλέστερα τραγούδια του στιχουργού (γραμμή 20) για τον οποίο συλλέγουμε μαζί με τα αναγνωριστικά τους. Στην συνέχεια εξερευνώντας το αρχείο αποθηκεύουμε σε μία λίστα το ζεύγος (Αναγνωριστικό,Τίτλος Τραγουδιού) ((song_id, song_title)) (γραμμή 22-25).

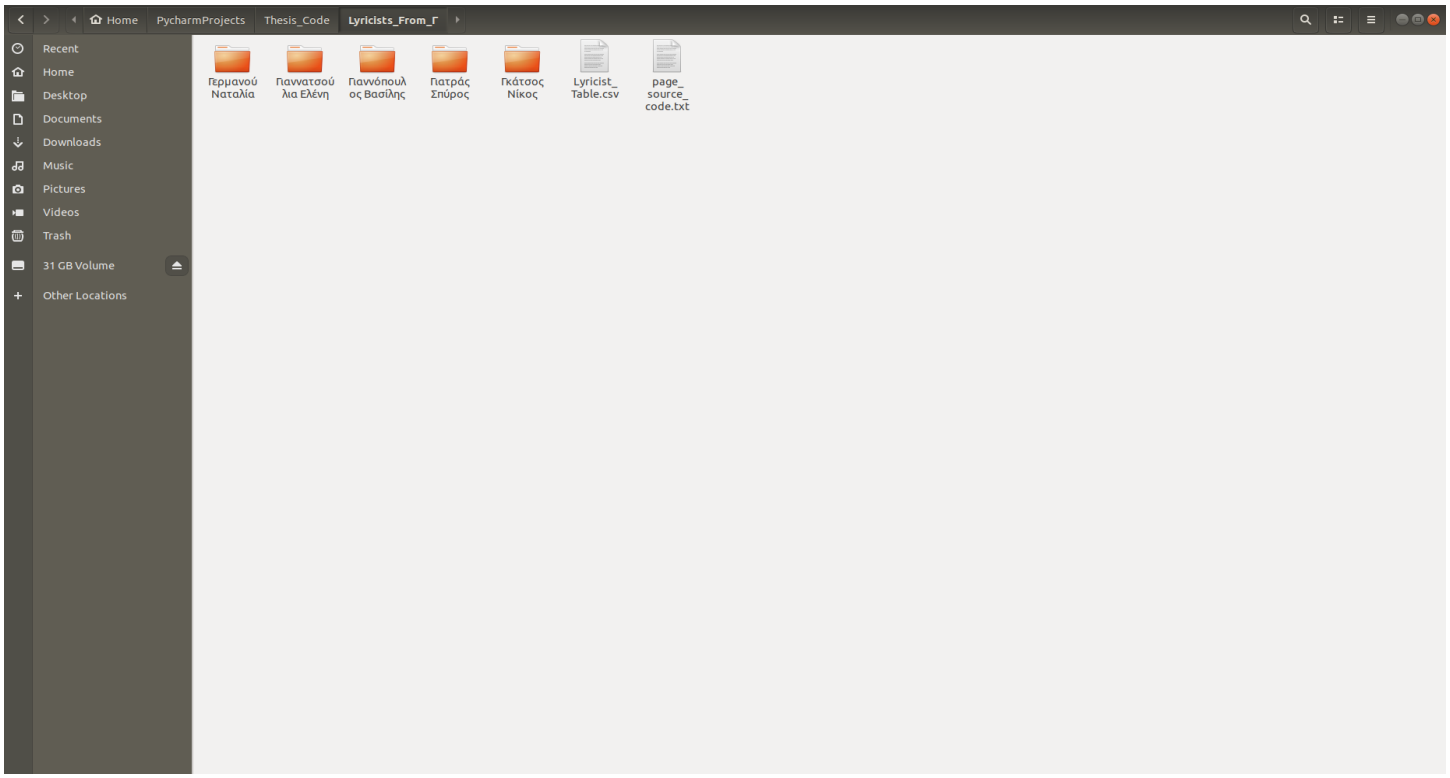
Με ένα βρόγχο for (γραμμή 27) πάνω τα στοιχεία της λίστας που δημιουργήθηκε προηγουμένως φτιάχνουμε το σύνδεσμό, url='https://www.stixoi.info/stixoi.php?info=Lyrics&act=details&'+song_id (γραμμή 31), με τον οποίο θα αναζητήσουμε τον πηγαίο κώδικα που περιέχει τους στίχους. Στέλνουμε την αίτηση και όταν βεβαιωθούμε ότι ολοκληρώθηκε επιτυχώς (γραμμή 34-35) δημιουργούμε ένα μονοπάτι (lyric_path) (γραμμή 40) για το αρχείο που θα φιλοξενήσει το πηγαίο κώδικα. Το όνομα του αρχείου αποτελείται από τον τίτλο του τραγουδιού και την κατάληξη .txt. Καθ' όλη την εκτέλεση του βρόγχου φροντίζουμε να ενημερώνουμε τον χρήστη για την κατάσταση των αιτήσεων, τους χρόνους αναμονής καθώς και τα αρχεία που δημιουργούνται (γραμμές 32,33,38,44,45,48,52). Ένα στιγμιότυπο της διαδικασίας φαίνεται στην παρακάτω εικόνα.



Σχήμα 6.10: Στιγμιότυπο από την διαδικασία συλλογής στίχων. Το IDE που χρησιμοποιήθηκε είναι το pycharm

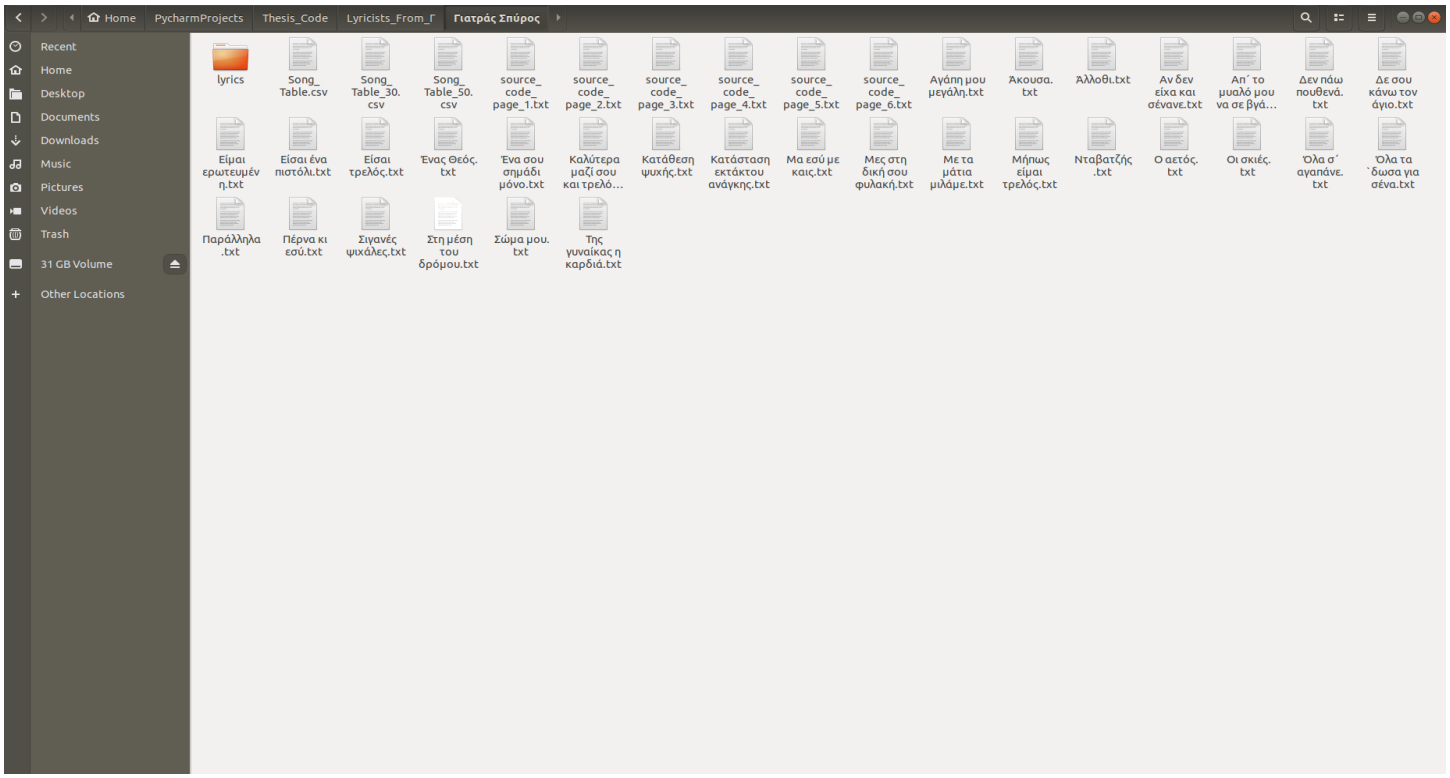
.

Παραθέτουμε δύο εικόνες που δείχνουν πως είναι οργανωμένοι οι φάκελοι και τα αρχεία από τις ρουτίνες που έχουμε τρέξει μέχρι στιγμής.



Σχήμα 6.11: Η εικόνα δείχνει τα αρχεία που περιέχονται μέσα στο κατάλογο Lyricist_From_Γ

.



Σχήμα 6.12: Η εικόνα δείχνει τα αρχεία που περιέχονται μέσα στο κατάλογο Lyricist_From_Γ/Γιατράς_Σπύρος

.

Πλέον είμαστε έτοιμοι να μετατρέψουμε όλους του πηγαίους κώδικες που περιέχουν στίχους σε αρχεία στίχων. Για να το κάνουμε αυτό αρχικά θα δημιουργήσουμε σε όλους τους καταλόγους των καλλιτεχνών ένα

φάκελο lyrics στον οποίο θα αποθηκεύσουμε τους στίχους που θα εξάγουμε από τους πηγαίους κώδικες. Η δημιουργία θα γίνει αυτόματα με την χρήση του παρακάτω κώδικα.

Create_Lyrics_Folder.py

```
import os
import re
path=os.getcwd()+'/Third_Round'
for subdir in os.scandir(path):
    if os.path.isdir(subdir.path) == True and len(re.findall('Lyricists', str(subdir.path))) == 1:
        for letter_path in os.scandir(subdir.path):
            if os.path.isdir(letter_path.path):
                lyric_path=letter_path.path+'/lyrics'
                j=0
                lyric_dict = dict()
                for file in os.scandir(lyric_path):
                    lyric_dict[str(file.path)] = j
                    j=j+1
                if j != 30:

                    print(lyric_path, j)
```

Αφού δημιουργήσαμε του φακέλους "lyrics" στους καταλόγους όλων των στιχουργών ήρθε η ώρα να φτιάξουμε τα πραγματικά αρχεία στίχων. Η δουλειά αυτή επιτυγχάνεται με τον παρακάτω κώδικα.

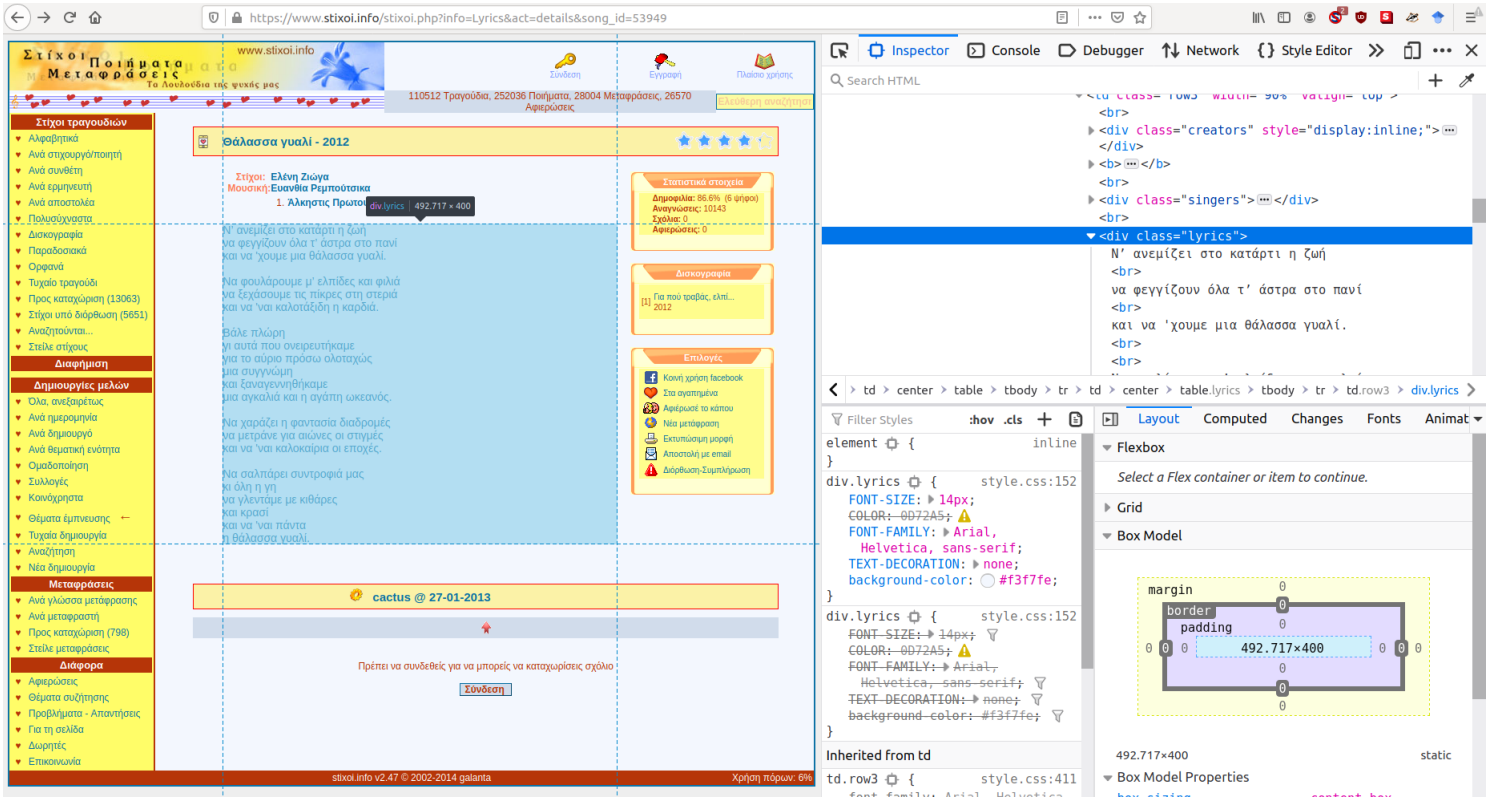
Create_Lyrics.py

```
import os
import re
from bs4 import BeautifulSoup as bs
path = os.getcwd()
for subdir in os.scandir(path):
    if os.path.isdir(subdir.path) == True and len(re.findall('Lyricists', str(subdir.path))) == 1:
        for lyricist in os.scandir(subdir.path):
            if os.path.isdir(lyricist.path) == True:

                for lyrics in os.scandir(lyricist.path):
                    if os.path.isdir(lyrics.path) == False and len(
                            re.findall('source_code|Song_Table', str(lyrics.path))) == 0:
                        r_file = open(lyrics.path, 'r',encoding='utf-8')
                        print(lyrics.path)
                        html = r_file.read()
                        r_file.close()
                        soup = bs(html, features='lxml')
                        lyrics_text = soup.find('div', {'class': "lyrics"})
                        lyric_path = lyricist.path + '/lyrics/' + str(
                            lyrics.path[len(lyricist.path) + 1:len(lyrics.path) - 4]) + '.txt'


                        if lyrics_text is None:
                            pass
                        else:
                            print(lyric_path)
                            w_file = open(lyric_path, 'w',encoding='utf-8')
                            w_file.write(lyrics_text.text)
                            w_file.close()
```

Αρχικά βρίσκουμε όλα τα μονοπάτια που οδηγούν στους φακέλους των καλλιτεχνών. Επειδή ο κώδικας που το πετυχαίνει έχει εξηγηθεί στις προηγούμενες διαδικασίες δεν θα επαναλάβουμε την ερμηνεία του. Διατρέχουμε τον φάκελο κάθε καλλιτέχνη ψάχνοντας για αρχεία κειμένου που έχουν τον πηγαίο κώδικα που περιέχει στίχους. Επειδή ο φάκελος του στιχουργού περιλαμβάνει και άλλα αρχεία κειμένου ή csv αρχεία από τις προηγούμενες διαδικασίες που δεν μας ενδιαφέρουν, χρησιμοποιούμε μία κανονική έκφραση ώστε το πρόγραμμα να μπορέσει να διαχωρίσει τα επιθυμητά αρχεία. Αυτό γίνεται με την γραμμή κώδικα:
if os.path.isdir(lyrics.path) == False and len(re.findall('source_code—Song_Table', str(lyrics.path))) == 0:(γραμμή 12) .

Ο παραπάνω κώδικας λέει στο πρόγραμμα να μην πάρει υπόψιν του μονοπάτια αρχείων που περιέχουν τις εκφράσεις Source_Code Song_Table. Αφού εντοπίσουμε τους πηγαίους κώδικες που μας ενδιαφέρουν, επεξεργαζόμαστε την html που περιέχουν με την Beautiful Soup. Πάλι εξερευνώντας με τα εργαλείο inspect

element του Firefox, βέλπουμε ότι οι στίχοι βρίσκονται στην ετικέτα div με κλειδιά τα {'class': "lyrics"}. Χρησιμοποιώντας αυτές τις πληροφορίες (γραμμή 18) εξάγουμε το μέρος του κώδικας που μας ενδιαφέρει. Ολοκληρώνοντας φτιάχνουμε το μονοπάτι (γραμμή 19-20) ώστε το αρχείο να αποθηκευτεί στον φάκελο "lyrics" και να έχει ως όνομα, το όνομα του τραγουδιού.



Σχήμα 6.13: Στην εικόνα φαίνεται το σημείο της html που περιέχει τους στίχους, δηλαδή η ετικέτα div με κλειδιά την έκφραση class="lyrics".

Σε κάποιο σημείο του κώδικα έχουμε μία δικλείδα ασφαλείας σε περίπτωση που δεν βρεθούν στίχοι να μην σταματήσει ο κώδικας (if lyrics_text is None:) (γραμμή 23-24). Αυτή η γραμμή εισήχθηκε γιατί διαπιστώσαμε πως υπήρξαν λάθη κατά την αποθήκευση των αρχείων στο προηγούμενο βήμα λόγο της κωδικοποίησης. Έτσι ορισμένοι πηγαίοι κώδικες δεν μπόρεσαν να αποθηκευτούν διότι το λειτουργικό σύστημα λόγων σφαλμάτων κωδικοποίησης δεν αναγνώρισε το μονοπάτι που οδηγούσε στο αρχείο τους. Επειδή το πάθημα γίνεται μάθημα, το μόνο που μπορούμε να δώσουμε σαν συμβουλή είναι να ελέγχουμε την κωδικοποίηση των πινάκων που δημιουργούνται από το πρόγραμμα που φτιάχνει του πίνακες με τα καλύτερα 30 τραγούδια πριν προχωρήσουμε στο κατέβασμα στίχων. Προκειμένου να διορθώσουμε αυτό το λάθος φτιάξαμε μία ρουτίνα που θα εντοπίζει σε ποιους στιχουργούς συνέβηκε το σφάλμα και θα βρίσκει τα τραγούδια όπου παρουσιάστηκε το συγκεκριμένο πρόβλημα.

Check_Lyrics_Folder.py

```
1  import os
2  import re
3  import csv
4
5  path = os.getcwd()
6  problem_paths = []
7  for subdir in os.scandir(path):
8      if os.path.isdir(subdir.path) == True and len(re.findall('Lyricists', str(subdir.path))) == 1:
9          for letter_path in os.scandir(subdir.path):
10             if os.path.isdir(letter_path.path):
11                 lyric_path = letter_path.path + '/lyrics'
12                 j = 0
13                 lyric_dict = dict()
```

```
14                  for file in os.scandir(lyric_path):
15                      lyric_dict[str(file.path)] = j
16                      j = j + 1
17                  if j != 30:
18                      problem_paths.append(letter_path.path)
19                      print(lyric_path, j)
20
21  for path in problem_paths:
22      read_file_path = path + '/Song_Table_30.csv'
23      title_path = path + '/lyrics'
24      title_dict = []
25      for file in os.scandir(title_path):
26          title = file.path[len(title_path) + 1:len(file.path) - 4]
27          title_dict.append(str(title))
28
29      read_file = open(read_file_path, 'r')
30      csv_reader = csv.reader(read_file, delimiter=',')
31      j = 0
32      for row in csv_reader:
33          j = j + 1
34          if str(row[2]) in title_dict:
35              pass
36          else:
37
38              print(path, row[1], row[2], j)
39              lyric_path = path + '/' + row[2] + '.txt'
40              print(lyric_path)
41      read_file.close()
```

Ο παραπάνω κώδικας αποτελεί μία απλή ρουτίνα με την οποία μπαίνουμε μέσα στο φάκελο lyrics κάθε στιχουργού (γραμμές 5-11) και μετράμε τα αρχεία χρησιμοποιώντας μία μεταβλητή-μετρητή j. Αν βρούμε ότι το πλήθος των αρχείων δεν είναι 30 (if j != 30:) (γραμμή 17), τότε αποθηκεύουμε το προβληματικό μονοπάτι σε μία λίστα problem_paths και τυπώνουμε στο χρήστη το πλήθος των αρχείων μαζί με το μονοπάτι που αποτέλεσε την εξαίρεση (print(lyric_path, j)) (γραμμές 18-20). Αφού βρούμε τα μονοπάτια, έπειτα για το κάθε μονοπάτι τρέχουμε τον κώδικα που βρίσκει σε ποιο τραγούδι έχει παρουσιαστεί το σφάλμα (γραμμές 21-41).

Ο κώδικας αρχίζει με ένα βρόγχο for για τον κάθε στιχουργό που περιέχει σφάλματα ο φάκελος lyrics. Αρχικά μπαίνουμε στο φάκελο lyrics (for file in os.scandir(tittle_path)) (γραμμή 25) και από κάθε μονοπάτι αρχείου που υπάρχει στο φάκελο εξάγουμε τον τίτλο του τραγουδιού (title = file.path[len(title_path) + 1:len(file.path) - 4]) (γραμμή 26). Ύστερα αποθηκεύουμε τον κάθε τίτλο σε μία δομή λεξικού (title_dict.append(str(ti 27). Στη συνέχεια ανοιγουμε το csv αρχείο Song_Table_30.csv και ελέγχουμε αν η τρίτη στήλη κάθε γραμμής που περιλαμβάνει τον τίτλο του τραγουδιών υπάρχει μέσα στο λεξικό των τίτλων. Αν δεν βρεθεί η εγγραφή (γραμμές 34-36) τυπώνουμε στο χρήστη το αναγνωριστικό "song_id", τον τίτλο του τραγουδιού, την γραμμή που έγινε το σφάλμα καθώς και το μονοπάτι του αρχείου που λείπει (γραμμές 38-41).

Επειδή στην περίπτωση μας τα σφάλματα ήταν λίγα σε αριθμό (κάτω από 10) η διαδικασία διόρθωσης ήταν ημιαυτόματη. Όμως χρησιμοποιώντας κάποιους βρόγχους for καθώς και κάποιες δομές δεδομένων ώστε να αποθηκεύσουμε τα προβληματικά μονοπάτια η διαδικασία εύρεσης και διόρθωσης σφαλμάτων μπορεί να γίνει πλήρως αυτοματοποιημένη. Πλέον αφού έχουμε εξασφαλίσει ότι κάθε φάκελος περιέχει 30 τραγούδια είμαστε έτοιμοι να επεξεργαστούμε τα δεδομένα μας ώστε να αναλυθούν σε δεύτερο στάδιο.

Σκοπός μας είναι να συμπιέσουμε του στίχους κάθε στιχουργού ώστε να μπορέσουμε να εξάγουμε κάποια συμπεράσματα την συμπεριφορά του συμπιεστή ανάλογα το περιεχόμενο και το μέγεθος που συμπιέζει. Για τον λόγο αυτό διαλέξαμε να προχωρήσουμε με τρεις συμπιεστές που έχουν αναλυθεί στο προηγούμενο κεφάλαιο. Οι συμπιεστές που διαλέξαμε είναι οι Huffman, gzip(LZ77)[1] και bzip2(Burrows-Wheeler),[2] που υπάρχουν σε βιβλιοθήκες της python εκτός από τον Huffman ο οποίος υλοποιήθηκε ξεχωριστά. Επειδή τα αρχεία των στίχων είναι πολύ μικρά σε μέγεθος επιλέξαμε μία υλοποίηση του Huffman που κάνει επί τόπου συμπίεση (on the fly). Ανάλογες τέτοιες μεθόδους περιέχουν και οι άλλοι δύο συμπιεστές που αναφέραμε.

[1] https://tools.ietf.org/html/rfc1951, https://tools.ietf.org/html/rfc1952
[2] https://www.sourceware.org/bzip2/manual/manual.pdf

Huffman.py

```
1
2
3  from Node import Node
4  from queue import  PriorityQueue
5
6
7  class  Huffman:
8      path=None
9       file_dict =None
10     huffman_tree=None
11     lyrics =None
12     code_dict=None
13     compressed_stream=None
14     padding=None
15
16     def  __init__ ( self ,path):
17          read_file  = open(path, 'r')
18          self . lyrics  =  read_file .read()
19          read_file . close ()
20          file_dict =dict()
21          for  i  in  range(0,  len( self . lyrics )):
22               if  self . lyrics [ i ]  in   file_dict :
23                    file_dict  [ self . lyrics [ i ]]  += 1
24               else :
25                    file_dict  [ self . lyrics [ i ]]  = 1
26          self .  file_dict = file_dict
27
28          h = PriorityQueue(-1)
29          for  (key,  value)  in  self .  file_dict  .items():
30               pair  = [value,key]
31               h.put(Node(pair))
32
33           leaf_list   = []
34          while  h. qsize ()  != 1:
35               left  = h.get()
36               right  = h.get()
37               root  = Node([left.get_data ()[0]  + right.get_data ()[0],  None])
38               root . add_right_child ( right )
39               root . add_left_child ( left )
40               right . add_parent(root)
41               left . add_parent(root)
42               if  left . get_data ()[1]  is  not  None:
43                    leaf_list  .append(left)
44               if  right . get_data ()[1]  is  not  None:
45                    leaf_list  .append(right)
46               h.put(root)
47          huffman_root=h.get()
48          self . huffman_tree=huffman_root
49
50          code_dict  = dict()
51          for  leaf  in   leaf_list  :
52               letter = leaf. get_data ()[1]
53               code =''
54               while  leaf . get_parent()  is  not  None:
55                    if  leaf .  is_left_child  ():
56                         code=code+'0'
57                    else :
58                         code=code+'1'
59                    leaf  = leaf. get_parent ()
60               code_dict [ letter ]  = code[::-1]
61          self . code_dict=code_dict
62
63
64     def  compress(self ):
65          residual  = ''
66          bitstring  = ''
67          for  i  in  range(0,  len( self . lyrics )):
68               letter  = self . lyrics [ i ]
69               code = self . code_dict [ letter ]
70               bitstring  = bitstring  + code
71
72          output = ''
73          if  len( bitstring )  % 8 == 0:
```

```
74              output = bitstring
75          elif len(bitstring) % 8 != 0 and len(bitstring) < 0:
76              output = bitstring.ljust(8, '0')
77          else:
78              residual = bitstring[len(bitstring) - len(bitstring) % 8:len(bitstring)]
79              self.padding=int(8-len(bitstring) % 8)
80              output = bitstring[0:len(bitstring) - len(bitstring) % 8] + residual.ljust(8, '0')
81
82          byte_no = 0
83          output_array = bytearray()
84          for i in range(0, int(len(output) / 8)):
85              byte = bitstring[byte_no:byte_no + 8]
86              output_array.append(int(byte, base=2))
87              byte_no=byte_no+8
88          self.compressed_stream=output_array
89
90      def decompress(self):
91          decomp_string=''
92          decomp_text=''
93          for i in range(0,len(self.compressed_stream)):
94              byte_to_int=self.compressed_stream[i]
95              byte_string='{0:08b}'.format(byte_to_int)
96              decomp_string=decomp_string+byte_string
97          decomp_string=decomp_string[0:len(decomp_string)-self.padding]
98          node=self.huffman_tree
99          for i in range(0,len(decomp_string)):
100             if decomp_string[i]=='1':
101                 if node.get_right_child() is not None:
102                     if node.get_right_child().get_data()[1] is not None:
103                         decomp_text=decomp_text+node.get_right_child().get_data()[1]
104                         node=self.huffman_tree
105                     else:
106                         node=node.get_right_child()
107             if decomp_string[i] == '0':
108                 if node.get_left_child() is not None:
109                     if node.get_left_child().get_data()[1] is not None:
110                         decomp_text = decomp_text + node.get_left_child().get_data()[1]
111                         node = self.huffman_tree
112                     else:
113                         node = node.get_left_child()
114         return decomp_text
```

Ο κώδικας αρχίζει με τον κατασκευαστή (γραμμές 16-61), τον οποίο προμηθεύουμε με τον μονοπάτι του αρχείου που επιθυμούμε να συμπιέσουμε. Η διαδικασία ξεκινάει ανοίγοντας το αρχείο και αποθηκεύοντας το περιεχομένο του στην μεταβλητή κλασης (self.lyrics, self.lyrics=read_file.read()) (γραμμές 17-19) . Στην συνέχεια διαβάζοντας τα δεδομένα που αποθηκεύσαμε σύμβολο προς σύμβολο δημιουργούμε μία δομή λεξικού η οποία αποθηκεύει τα σύμβολα μαζί με τον αριθμό εμφανίσεων τους στο κείμενο (γραμμές 21-25). Αφού ολοκληρωθεί η κατασκευή του λεξικού το αποθηκεύουμε στην μεταβλητή κλάσης self.file_dict (self.file_dict=file_dict)(γραμμή 26). Χρησιμοποιώντας το λεξικό που μόλις δημιουργήθηκε, με ένα βρόγχο (for) πάνω στις εγγραφές του αποθηκεύουμε το ζεύγος (σειρά εμφανίσεων, σύμβολο) μέσα σε μία ουρά προτεραιότητας ελαχίστου υπό τη μορφή κόμβου (Node) (γραμμές 28-31) που υλοποιεί τον κόμβο ενός δυαδικού δένδρου. Η κλάση Node θα παρουσιαστεί αφού τελειώσουμε την ανάλυση του Huffman.

Ο κατασκευαστής συνεχίζει φτιάχνοντας το δένδρο κωδικοποίησης - αποκωδικοποίησης Huffman. Μέχρι να μείνει μόνο ένα στοιχείο στη ουρά ελαχίστου σε κάθε επανάληψη βγάζουμε από την ουρά δύο κόμβους. Ο πρώτος θεωρείται ως αριστερό παιδί (left) και ο δεύτερος ως δεξιό (right). Στην συνέχεια κατασκευάζουμε ένα τρίτο κόμβο (root) που έχει ως δεδομένα το άθροισμα των εμφανίσεων του δύο προηγούμενων κόμβων και το σύμβολο None. Συνδέουμε του τρεις κόμβους βάζοντας τον κόμβο root να έχει αριστερό παιδί τον left και δεξί παιδί τον right και αντίστοιχα τους κόμβους left, right να έχουν σαν πατέρα τον root (γραμμές 34-41). Στην συνέχεια ελέγχουμε αν οι κόμβοι left και right έχουν αποθηκευμένα σύμβολα στα δεδομένα τους έτσι ώστε να τους αποθηκεύσουμε ανάλογα σε μία λίστα που περιέχει μόνο τα φύλλα του δένδρου (γραμμές 33,42-45). Όταν τελειώσει η παραπάνω επαναληπτική διαδικασία η ουρά ελαχίστου περιέχει μόνο ένα κόμβο ο οποίος είναι η ρίζα του δένδρου κωδικοποίησης. Το δένδρο σώζεται στην μεταλητή κλάσης self.huffman_tree (γραμμή 48).

Αφού έχουμε το δένδρο και τα φύλλα μπορούμε να εξάγουμε του κώδικες. Χρησιμοποιώντας ένα βρόγχο

for που τρέχει πάνω στα φύλλα του δένδρου ακολουθούμε το μονοπάτι που περιέχει το εκάστοτε φύλλο μέχρι να φτάσουμε στην ρίζα. Σε κάθε βήμα αν είμαστε δεξιό παιδί εμπλουτίζουμε τον κώδικα του συμβόλου με ένα ¨1¨, ενώ αν είμαστε αριστερό αποθηκεύουμε στον υπάρχων κώδικα το ¨0¨. Ολοκληρώνοντας την διαδικασία αποθηκεύουμε τους κώδικες ανεστραμμένους σε μία δομή λεξικού code_dict (γραμμές 50-61). Στο σημείο αυτό ολοκληρώνεται ο κατασκευαστής της κλάσης.

Η μέθοδος compress εκτελεί την συμπίεση του αρχείου. Η διαδικασία αρχίζει φτιάχνοντας μία κενή συμβολοσειρά (bitstring) (γραμμή 66). Για κάθε σύμβολο των στίχων, βρίσκουμε την κωδική λέξη του συμβόλου και επεκτείνουμε την κωδικοποιημένη συμβολοσειρά επικολλώντας την (γραμμές 67-70). Στην συνέχεια ελέγχουμε αν το μήκος της είναι πολλαπλάσιο του 8 (γραμμή 73). Στην περίπτωση που είναι την αποθηκεύουμε στην μεταβλητή output (γραμμή 74). Αν δε είναι πολλαπλάσιο του 8 και έχει μήκος μικρότερο του 8 τότε προσθέτοντας ένα padding από ΄0΄, τη επεκτείνουμε σε μήκους 8 και την αποθηκεύουμε στην μεταβλητή output (γραμμές 75-76). Αν δεν έχει μήκος πολλαπλάσιο του 8 και το μήκος της ξεπερνάει το 8, αποθηκεύουμε το υπόλοιπο στην μεταβλητή residual, το επεκτείνουμε με padding από ΄0΄ ώστε να έχει μήκος 8 και στην συνέχεια το επικολλάμε στην κωδικοποιημένη συμβολοσειρά (γραμμές 77-80). Ολοκληρώνοντας την συμπίεση βρίσκουμε από πόσα byte αποτελείται η συμβολοσειρά και τα αποθηκεύουμε σε ένα διάνυσμα από bytes και σώζουμε το διάνυσμα στην μεταβλητή self.compressed που αποτελεί το κωδικοποιημένο μας κείμενο (γραμμές 82-88).

Η αποσυμπίεση ξεκινάει με το συμπιεσμένο διάνυσμα. Κάθε byte του διανύσματος το μετατρέπουμε σε μία συμβολοσειρά από ΄0΄ και ΄1΄, τα οποία στην συνέχεια τα επικολλούμε στην κωδικοποιημένη συμβολοσειρά. Όταν ολοκληρωθεί η μετάφραση των στοιχείων του διανύσματος και κατασκευαστεί ολόκληρη η αποκωδικοποιημένη συμβολοσειρά αφαιρούμε το padding (γραμμές 93-97). Για κάθε στοιχείο της κωδικοποιημένης συμβολοσειράς ελέγχουμε αν είναι ΄0΄ ή ΄1΄ (γραμμές 100,107). Αν είναι ΄1΄ και δεν βρισκόμαστε σε φύλλο βλέπουμε μήπως το δεξί παιδί του τρέχοντα κόμβου είναι φύλλο. Αν είναι αντικαθιστούμε στην αποκωδικοποιημένη συμβολοσειρά το σύμβολο του φύλου και μεταβαίνουμε πάλι στη ρίζα του δένδρου αλλιώς προχωράμε στον επόμενο δεξί κόμβο (γραμμές 101-106). Η διαδικασία είναι πανομοιότυπη στην περίπτωση που το στοιχείο είναι ΄0΄. Οι επαναλήψεις συνεχίζονται μέχρι να εξαντληθεί η κωδικοποιημένη συμβολοσειρά.

Node.py

```
1
2  class Node:
3      parent = None
4       left_child  = None
5       right_child  = None
6      data = None
7
8      def __init__ (self,value):
9          self.parent = None
10         self. left_child  = None
11         self. right_child  = None
12         self.data = value
13
14     def get_parent(self):
15         return self.parent
16
17     def get_left_child (self):
18         return self. left_child
19
20     def get_right_child (self):
21         return self. right_child
22
23     def get_data(self):
24         return  self.data
25
26     def add_left_child (self, child_node):
27             self. left_child  = child_node
28
29     def add_right_child (self, child_node):
30             self. right_child  = child_node
31
32     def add_parent(self,parent_node):
33         self.parent=parent_node
```

```
34
35
36      def add_data(self, data):
37          if self.data is not None:
38              self.data = data
39
40      def remove_left_child(self):
41          self.left_child.parent = None
42          self.left_child = None
43
44      def remove_right_child(self):
45          self.right_child.parent = None
46          self.right_child = None
47
48      def remove_data(self):
49          self.data=None
50
51      def is_left_child(self):
52          return self.get_parent().get_left_child()==self
53
54      def is_right_child(self):
55          return self.get_parent().get_right_child()==self
56
57      def __lt__(self, other):
58          return self.get_data()[0] <other.get_data()[0]
```

Η κλάση Node είναι μία πολύ απλή κλάση που υλοποιεί ένα κόμβο ενός δυαδικού δένδρου στον οποίο μπορούμε να προσθέσουμε ή να αφαιρέσουμε παιδιά και πατέρα καθώς και να αλλάξουμε τα δεδομένα του. Το μόνο που αξίζει να επισημανθεί γι αυτή την κλάση είναι η μέθοδος __lt__ με την οποία ορίζουμε πως θα συγκρίνονται δύο κόμβοι. Εμείς σχεδιάζοντας τον κώδικα huffman ξέραμε από πριν ότι οι κόμβοι θα αποθηκεύουν δύο στοιχεία σαν δεδομένα από τα οποία μόνο το πρώτο χρησιμοποιείτε για την σύγκριση. Αν δεν συμπεριλάβουμε την μέθοδο σύγκρισης στον κώδικα η python θα βγάζει σφάλμα κάθε φορά που πάμε να συγκρίνουμε δύο κόμβους, γιατί δεν θα ξέρει πως να τους συγκρίνει.

6.3 Επεξεργασία-Παρουσίαση Αποτελεσμάτων-Πείραμα 1

Αφού πλέον έχουμε συλλέξει όλα τα δεδομένα μας και τους συμπιεστές με τους οποίους θα δουλέψουμε μπορούμε να περάσουμε στην ανάλυση τους. Οι συνολικοί στίχοι θα αναλυθούν σε δύο φάσεις, αρχικά για κάθε στιχουργό θα φτιάξουμε ένα αρχείο κειμένου που θα περιέχει και τα 30 τραγούδια του, το οποίο και θα συμπιέσουμε. Αυτή η διαδικασία θα μας βοηθήσει να καταλάβουμε την συμπεριφορά των συμπιεστών για μεγάλα αρχεία με ένα ομογενές (ίδια πηγή) περιεχόμενο προκειμένου να διαπιστώσουμε αν στην πράξη ισχύουν όλα αυτά που μαθαίναμε στην θεωρία για την ασυμπτωτική συμπεριφορά στάσιμων και εργοδικών πηγών. Στην συνέχεια οι στίχοι κάθε τραγουδιού θα συμπιεστούν ξεχωριστά βοηθώντας μας έτσι να κατανοήσουμε πως θα αντιδράσουν οι συμπιεστές σε μικρά αρχεία με πολύ διαφορετικά περιεχόμενα. Οι δύο φάσεις θα οργανωθούν σε δύο ξεχωριστά πειράματα.

Mega_File.py

```
1
2   import os
3   import re
4   path=os.getcwd()
5
6    letter_dir_list =[]
7   for letter_dir in os.scandir(path):
8       if os.path.isdir(letter_dir.path) and len(re.findall('Lyricist', letter_dir.path))==1:
9           letter_dir_list.append(letter_dir.path)
10   lyricist_list =[]
11  for lyricist_path in sorted(letter_dir_list):
12      for dir in os.scandir(lyricist_path):
13          if os.path.isdir(dir.path):
14              lyricist_list.append(dir.path)
15  print(lyricist_list)
16  print(len(lyricist_list))
```

```
17  for i in range(0,len( lyricist_list )):
18      lyricist =sorted( lyricist_list )[i]
19      lyrics_file_path =lyricist+'/lyrics'
20      i=0
21      for file in os.scandir( lyrics_file_path ):
22          write_file =open(lyricist+'/Mega_Lyrics.txt',mode='a')
23          read_file =open(file.path,'r')
24          lyrics=read_file.read()
25          write_file.write( lyrics )
26          read_file.close()
27          i+=1
28      write_file.close()
29      print(i, lyrics_file_path )
```

Για να κατασκευάσουμε το αρχείο όλων των στίχων, αρχικά στον κατάλογο εργασίας μας αναζητούμε όλους του φακέλους που περιέχουν στο μονοπάτι τους τη λέξη Lyricist . Αφού τους βρούμε, τους αποθηκεύουμε σε μία λίστα (γραμμές 6-9). Στην συνέχεια μπαίνουμε σε κάθε φάκελο και εξερευνώντας τα αρχεία που περιέχει, βρίσκουμε τους φακέλους των καλλιτεχνών και τους αποθηκεύουμε σε μία άλλη λίστα (γραμμές 10-14). Στην συνέχεια χρησιμοποιούμε το κάθε μονοπάτι του εκάστοτε καλλιτέχνη που είναι αποθηκευμένο στην τελευταία λίστα ώστε να χτίσουμε το μονοπάτι που οδηγεί στο φάκελο lyrics (γραμμές 17-19). Ολοκληρώνοντας κάθε αρχείο κειμένου του φακέλου lyrics το αντιγράφουμε στο αρχείο Mega_lyrics.txt (γραμμές 20-28). Αφού έχουμε κατασκευάσει το συνολικό αρχείο στίχων σε κάθε στιχουργό μπορούμε πλέον να περάσουμε στην συμπίεση τους προκειμένου να διεξάγουμε κάποια πρώτα συμπεράσματα για το δείγμα μας.

Mega_File_Compression.py

```
1
2   import os
3   import csv
4   import sys
5   from Huffman_2 import Huffman
6   import gzip
7   import bz2
8   write_file =open(os.getcwd()+'/Summary_Table_Compression.csv','w')
9   csv_writer=csv.writer( write_file ,delimiter=',',quotechar='"',quoting=csv.QUOTE_MINIMAL)
10  first_row =['Lyricist_id ','Στιχουργός','Αριθμός στίχων','Προβολές','Gzip','Bzip2','Huffman']
11  csv_writer.writerow(first_row)
12  read_file =open(os.getcwd()+'/Summary_Table.csv','r')
13  csv_reader=csv.reader( read_file ,delimiter=',',quotechar='"',quoting=csv.QUOTE_MINIMAL)
14  row=next(csv_reader)
15  for row in csv_reader:
16      name=row[1][1:len(row[1])]
17      if name=='Ασιμος Νικόλας':
18          letter ='A'
19      else:
20          letter =row[1][1]
21      path=os.getcwd()+'/Lyricists_From_'+letter+'/'+name+'/Mega_Lyrics.txt'
22      i=0
23
24      lyric_text = open(path, 'rb')
25      content = lyric_text.read()
26      g_file = gzip.compress(content, compresslevel=9)
27      g_data_savings = round(float((sys.getsizeof(content) - sys.getsizeof( g_file )) / sys.getsizeof(content)), 4) * 100
28
29      b_file = bz2.compress(content, compresslevel=9)
30      b_data_savings = round(float((sys.getsizeof(content) - sys.getsizeof( b_file )) / sys.getsizeof(content)), 4) * 100
31      lyric_text.close()
32      h_file = Huffman(path)
33      h_file.compress()
34      h_data_savings = round(float((sys.getsizeof( h_file.lyrics ) - sys.getsizeof( h_file.compressed_stream))
35                                   / sys.getsizeof( h_file.lyrics )),4) * 100
36      list = [row[0],row[1],row[2],row[3], g_data_savings, b_data_savings, h_data_savings]
37      csv_writer.writerow( list )
38  read_file.close()
39  write_file.close()
```

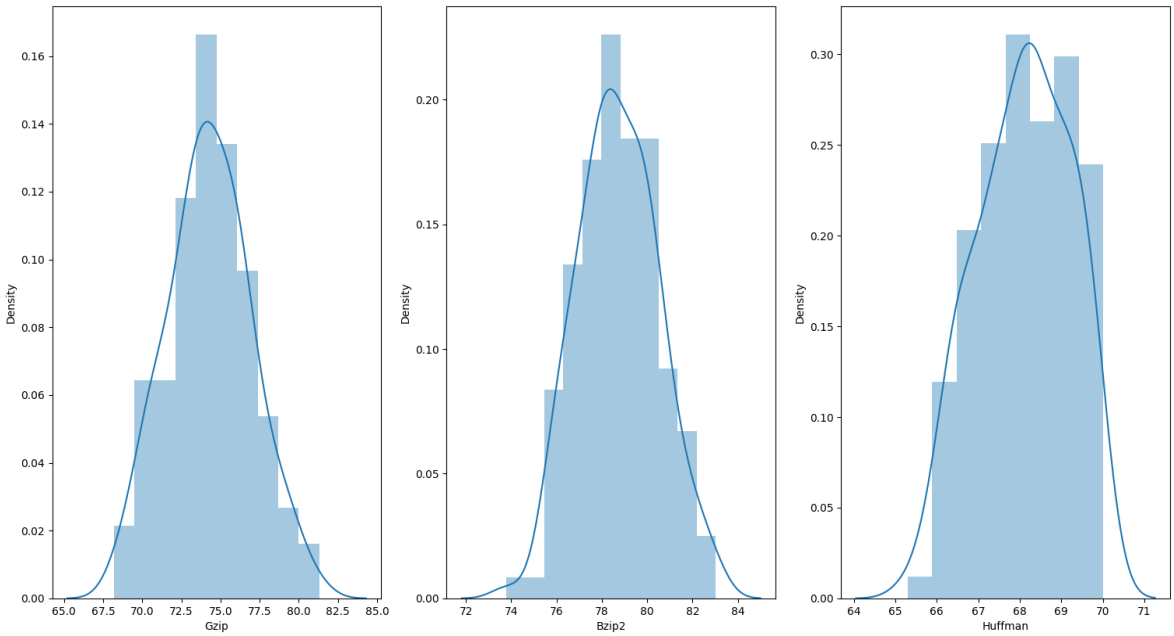
Για να συμπιέσουμε τα αρχεία Mega_Lyrics.txt αρχικά ανοίγουμε τον συγκεντρωτικό πίνακα των στιχουργών και διαβάζουμε κάθε γραμμή ώστε να βρούμε το όνομα και το αρχικό γράμμα κάθε καλλιτέχνη, εξαιρείτε ο Άσιμος γιατί το πρώτο γράμμα του ονόματος είναι τονισμένο. Στην συνέχεια με βάση τα δύο αυτά στοιχεία

κατασκευάζουμε το μονοπάτι που θα μας οδηγήσει στο επιθυμητό αρχείο (γραμμές 12-21). Συμπιέζουμε τους στίχους κάθε αρχείου με τους συμπιεστές Gzip, Bzip2 και Huffman (γραμμές 26-35) κατά σειρά. Μετά την κάθε συμπίεση υπολογίζουμε το ποσοστό ελάττωσης του αρχικού μεγέθους του αρχείου:

$$data\ savings = (1 - \frac{Compressed\ Size}{Uncompressed\ Size}) \cdot 100\%$$

6.3.1 Η κατανομή των ποσοστών συμπίεσης

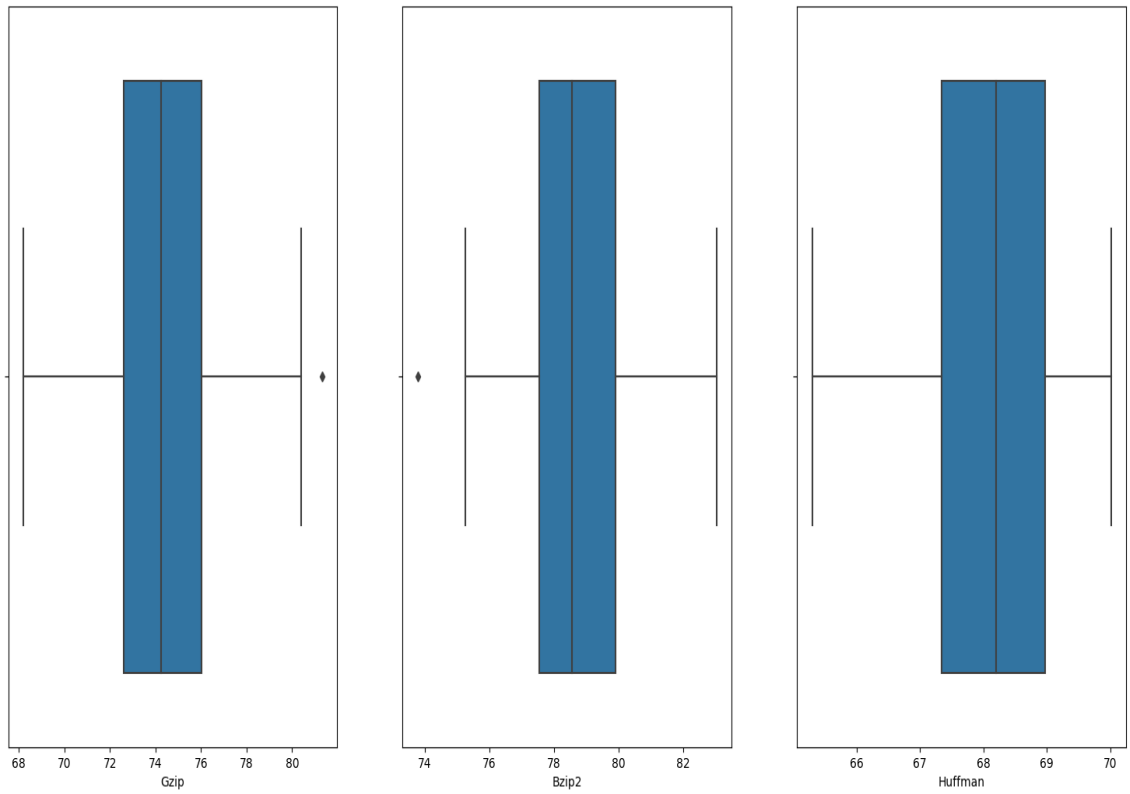
Έχοντας το αρχείο στα χέρια μας θα εξετάσουμε γραφικά αν υπάρχουν διαφοροποιήσεις ανάμεσα στους διάφορους στιχουργούς αλλά και να εκτιμήσουμε αν υπάρχουν διαφορές στον τρόπο λειτουργίας και την συμπεριφορά των συμπιεστών. Αρχικά για να βρούμε το μέσο ποσοστό συμπίεσης του κάθε συμπιεστή θα δημιουργήσουμε κάποια ιστογράμματα τα οποία απεικονίζονται μαζί με την εμπειρική κατανομή που ακολουθούν τα ποσοστά συμπίεσης του κάθε συμπιεστή.



Σχήμα 6.14: Τα ιστογράμματα και των ποσοστών συμπίεσης για κάθε ένα από τους συμπιεστές μαζί με τις εμπειρικές κατανομές.

Με μία πρώτη ματιά μπορούμε να δούμε ότι την καλύτερη απόδοση στην συμπίεση των κειμένων κατέχει ο Bzip2 που υλοποιεί τον μετασχηματισμό Burrow-Wheeler. Τα ποσοστά συμπίεσης είναι συγκεντρωμένα γύρω από το 80% και παρουσιάζουν μικρή διασπορά από το κέντρο. Ακολουθεί ο Gzip που υλοποιεί τον LZ77 ο οποίος επιτυγχάνει ποσοστά συμπιέσεων με ένα μέσο όρο που κινείται γύρω από το 75%, ο οποίος είναι χαμηλότερος από τον Bzip. Επίσης εμφανίζει ελαφρώς μεγαλύτερη διασπορά στις τιμές του πάλι σε σχέση με τον Bzip. Ο χειρότερος από όλους φαίνεται να είναι ο Huffman που υλοποιήσαμε, ο οποίος έχει ένα μέσο όρο γύρω στο 68% ποσοστό πολύ χαμηλότερο από τους προηγούμενους δύο. Ενώ στο σχήμα φαίνεται πιο πλατύς από τους άλλους αν παρατηρήσουμε τις τιμές του οριζόντιου άξονα θα δούμε ότι η διασπορά που εμφανίζει είναι η μικρότερη.

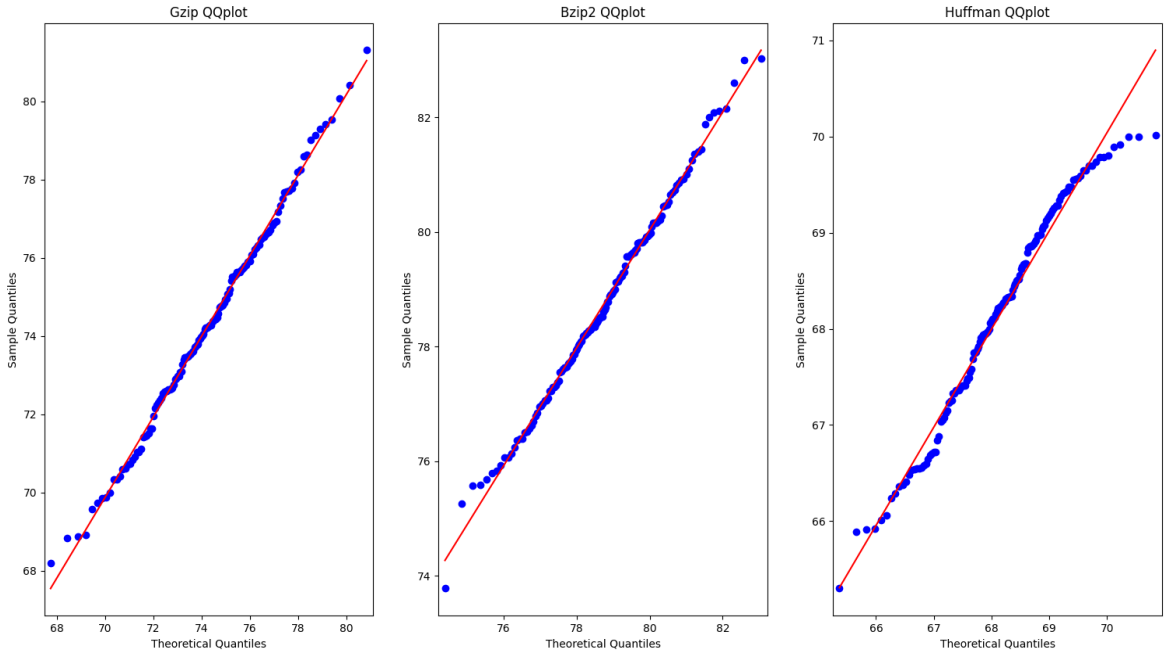
Από τις γραφικές παραστάσεις παρατηρούμε ότι οι παραπάνω εμπειρικές κατανομές θυμίζουν τη κανονική κατανομή $N(\mu,\ \sigma^2)$. Θα επιβεβαιώσουμε την υπόθεση μας με το στατιστικό τεστ Kolmogorov-Smirnov αλλά και γραφικά τα QQ plots και Boxplots των δεδομένων μας.



Σχήμα 6.15: Τα boxplots για τους συμπιεστές Gzip, Bzip2 και Huffman.

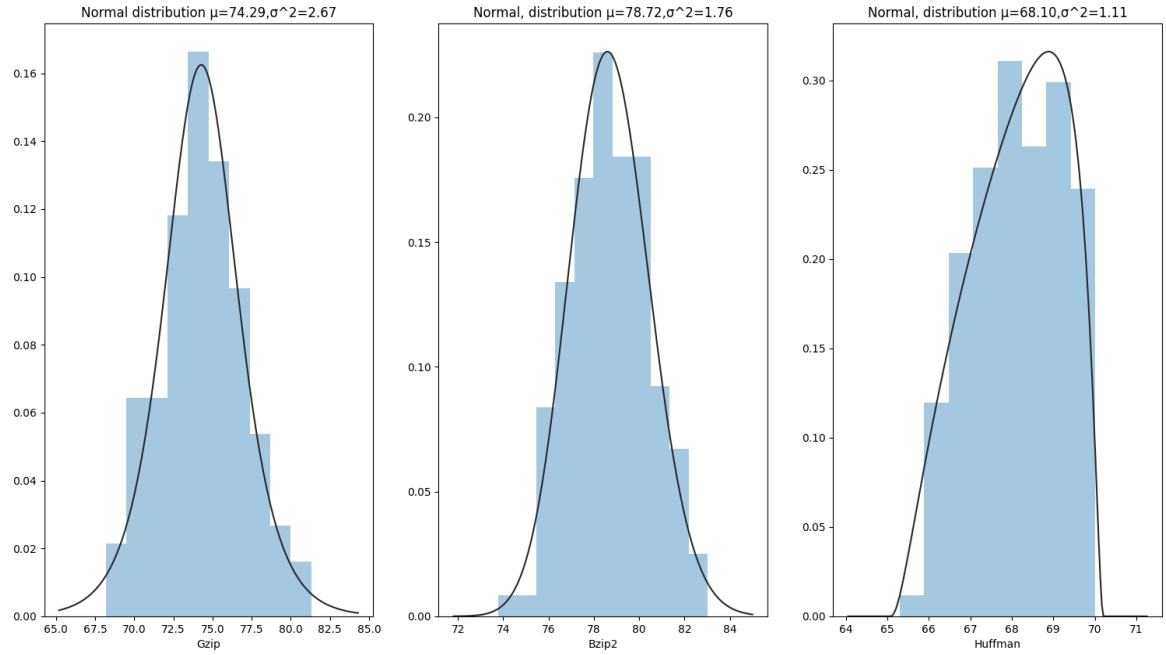
Συμπιεστής	Μέση τιμή	Διασπορά	Τιμή Kolmogorov-Smirnov	p-value
Gzip	74.29274647887324	2.6688054890800754	0.034997421841913234,	0.9926317497899979
Bzip2	78.72232394366198	1.7610262954795313	0.04603708201215251,	0.9104968240145357
Huffman	68.10338028169015	1.1135705087335284	0.06275348629676825	0.6082013473361361

Από τα αποτελέσματα του τεστ φαίνεται ότι οι συμπιεστές ακολουθούν την κανονική κατανομή ($p-value > 0.05$) και επιβεβαιώνουν όλα όσα παρατηρήσαμε γραφικά προηγουμένως στα ιστογράμματα. Συνεχίζουμε με τις γραφικές παραστάσεις προκειμένου να δούμε αν πληρούνται οι υποθέσεις τι· κανονικότητας



Σχήμα 6.16: Τα QQ plots για τους συμπιεστές Gzip, Bzip2 και Huffman.

Από τις δύο παραπάνω εικόνες (Σχήμα 1.15, Σχήμα 1.16) φαίνεται ότι ικανοποιούνται οι υποθέσεις τις κανονικότητας. Παρακάτω παρουσιάζονται τα ιστογράμματα των συμπιεστών μαζί με τις προσεγγιστικές κανονικές κατανομές που προέκυψαν



Σχήμα 6.17: Τα ιστογράμματα μαζί με τις προσεγγιστικές κανονικές κατανομές για τους συμπιεστές Gzip, Bzip2 και Huffman.

Συμπερασματικά λοιπόν η πρώτη εικόνα που έχουμε από το δείγμα μας είναι ότι τα συρραμμένα κείμενα στίχων που κατασκευάσαμε συμπιέζονται σε ποσοστό πάνω από 60% ο καλύτερος συμπιεστή αναδεικνύεται ο Bzip2 καθώς έχει την μεγαλύτερη μέση τιμή και μικρή διασπορά ενώ τη μεγαλύτερη ευαισθησία κρίνοντας από την διασπορά φαίνεται να τη έχει ο gzip. Τελευταίος έρχεται ο Huffman που παρουσιάζει τον μικρότερο μέσο όρο αλλά και την μικρότερη διασπορά. Ο λόγος που Huffmna πετυχαίνει ποσοστά συμπίεσης τόσο κοντά στον μέσο θα φανεί στην περεταίρω ανάλυση. Ο κώδικας που χρησιμοποιήθηκε για το κομμάτι αυτό φαίνεται στην παραπάνω εικόνα.

Experiment_1_Normality.py

```
1
2  import pandas as pd
3  import matplotlib.pyplot as plt
4  import os
5  from scipy import stats
6  import seaborn as sns
7  import statsmodels.api as sm
8  table = pd.read_csv(os.getcwd() + '/Experiment_1.csv', index_col=0)
9
10 g_data = table['Gzip']
11 b_data = table['Bzip2']
12 h_data = table['Huffman']
13
14 data=[g_data,b_data,h_data]
15 mu=[]
16 var=[]
17 for data in data:
18     params=stats.norm.fit(data)
19     result=stats.kstest(data,'norm',params)
20     mu.append(params[0])
21     var.append(params[1])
22     print(params,result)
23
24 f, axes=plt.subplots(1,3)
25 sns.distplot(g_data, fit=stats.logistic ,kde=False,ax=axes[0])
26 axes[0].set_title ('Normal distribution μ=74.29,σ^2=2.67')
27 sns.distplot(b_data, fit=stats.powerlognorm,kde=False,ax=axes[1])
28 axes[1].set_title ('Normal, distribution μ=78.72,σ^2=1.76')
29 sns.distplot(h_data, fit=stats.johnsonsb,kde=False,ax=axes[2])
30 axes[2].set_title ('Normal, distribution μ=68.10,σ^2=1.11')
31
32
33 f_2, axes_2=plt.subplots(1,3)
34 sns.boxplot(g_data,ax=axes_2[0])
35 sns.boxplot(b_data,ax=axes_2[1])
36 sns.boxplot(h_data,ax=axes_2[2])
37 plt.show()
38 f_3, axes_3=plt.subplots(1,3)
39 sm.qqplot(g_data,dist=stats.norm,loc=mu[0],scale=var[0],line='r' ,ax=axes_3[0])
40 axes_3[0].set_title ('Gzip QQplot')
41 sm.qqplot(b_data,dist=stats.norm,loc=mu[1],scale=var[1],line='r',ax=axes_3[1])
42 axes_3[1].set_title ('Bzip2 QQplot')
43 sm.qqplot(h_data,dist=stats.norm,loc=mu[2],scale=var[2],line='r',ax=axes_3[2])
44 axes_3[2].set_title ('Huffman QQplot')
45 plt.show()
46
47 f_4, axes_4=plt.subplots(1,3)
48 sns.distplot(g_data,kde=True,ax=axes_4[0])
49 sns.distplot(b_data,kde=True,ax=axes_4[1])
50 sns.distplot(h_data,kde=True,ax=axes_4[2])
51 plt.show()
```

6.3.2 Η συμπίεση σε σχέση με το είδος των τραγουδιών

Αφού πήραμε μία πρώτη ιδέα για την συμπεριφορά των συμπιεστών ήρθε η ώρα να εξετάσουμε πως αντιδρούν σε σχέση με άλλες μεταβλητές του συγκεντρωτικού πίνακας. Πριν ξεκινήσουμε την ανάλυση θα παρουσιάσουμε πρώτα κάποια γραφήματα που θα μας δώσουν μία γενική εικόνα για το σύνολο δεδομένων μας. Υπενθυμίζουμε

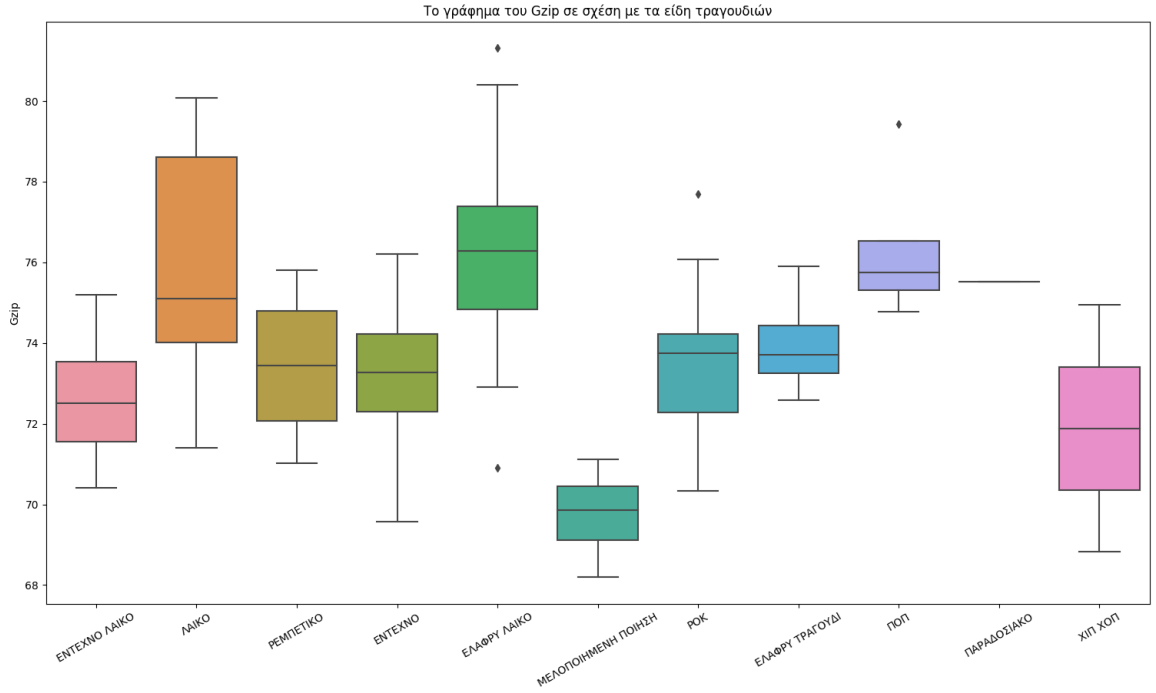
ότι έχουμε συλλέξει συνολικά 141 στιχουργούς[3] και 30 τραγούδια από τον κάθε ένα, άρα έχουμε κατεβάσει 4230 στίχους. Για να γίνει καλύτερη μελέτη προβήκαμε σε μία επεξεργασία των δεδομένων. Αρχικά προσπαθήσαμε να κατατάξουμε του στιχουργούς σε γενικές κατηγορίες ανάλογα το είδος των τραγουδιών. Τα είδη αποφασίστηκαν κατόπιν μελέτης από το διαδίκτυο[4] για τα διάφορα είδη ελληνική μουσικής που δημιουργήθηκαν στο πέρα της ιστορίας. Συγκεντρωτικά τα είδη είναι:

1. ΠΑΡΑΔΟΣΙΑΚΟ
2. ΡΕΜΠΕΤΙΚΟ
3. ΛΑΙΚΟ
4. ΕΛΑΦΡΥ ΤΡΑΓΟΥΔΙ
5. ΕΝΤΕΧΝΟ
6. ΜΕΛΟΠΟΙΗΜΕΝΗ ΠΟΙΗΣΗ
7. ΕΝΤΕΧΝΟ ΛΑΙΚΟ
8. ΡΟΚ
9. ΕΛΑΦΡΥ ΛΑΙΚΟ
10. ΠΟΠ
11. ΧΙΠ ΧΟΠ

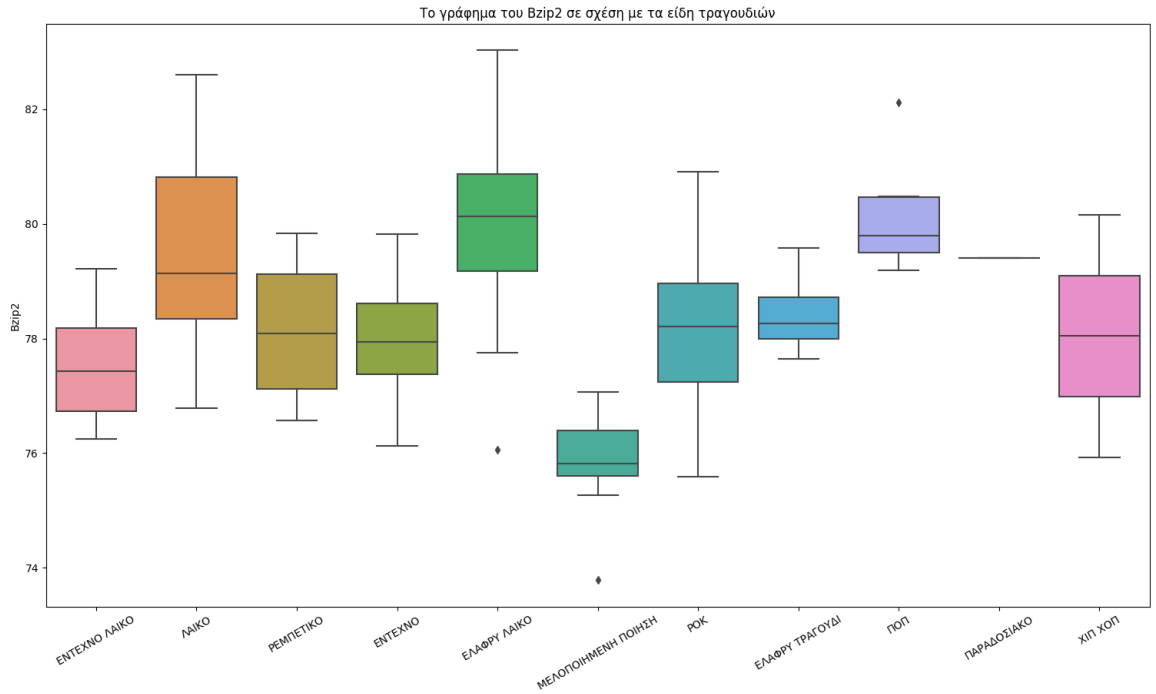
Θα ξεκινήσουμε σχεδιάζοντας boxplots του κάθε συμπιεστή σε σχέση με το είδος προκειμένου να εξετάσουμε αν υπάρχει κάποια διαφοροποίηση ως προς την συμπίεση σε σχέση με το είδος που συμπιέζεται.

[3]Είχαμε συλλέξει 142 αλλά μετά από έλεγχο των δεδομένων καταλάβαμε ότι ένα στιχουργός είχε απαγορέψει την δημοσίευση στίχων στη ιστοσελίδα, οπότε και τον αφαιρέσαμε.

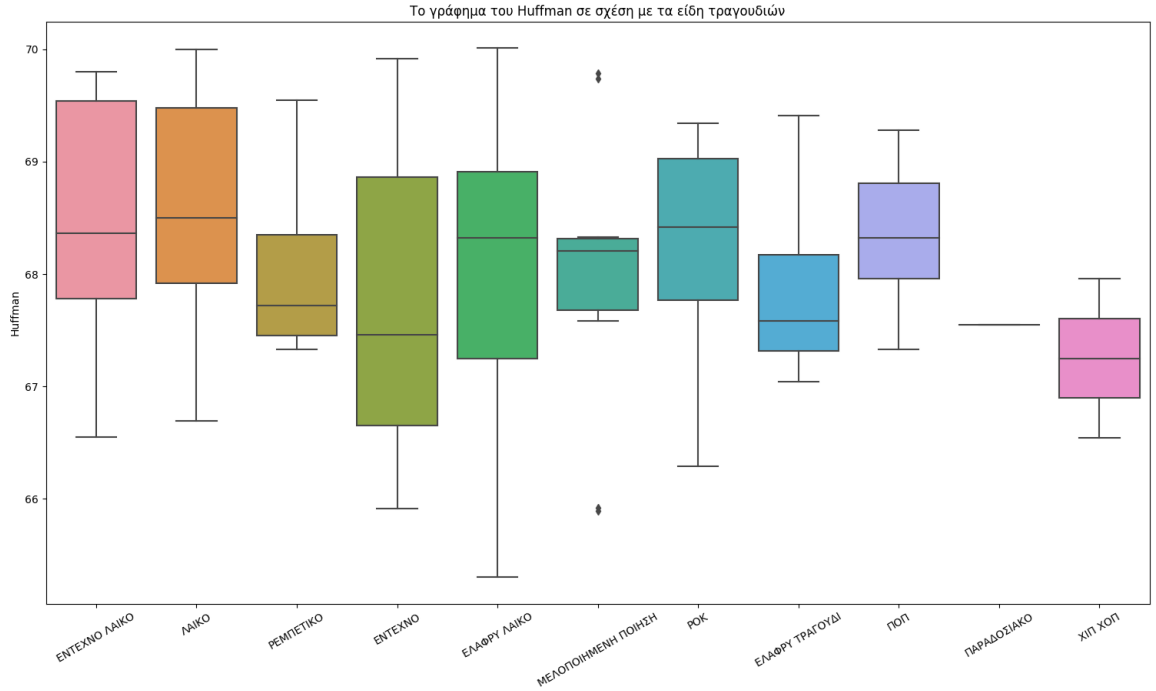
[4]
1. `https://www.musicportal.gr/music?lang=el`
2. `https://ecourse.uoi.gr/course/view.php?id=1468`
3. `https://www.slideshare.net/filippos_chatziandreas/ss-76493864`



Σχήμα 6.18: Το γράφημα που δείχνει την σχέση μεταξύ των ποσοστών συμπίεσης του Gzip και το είδος του τραγουδιού.

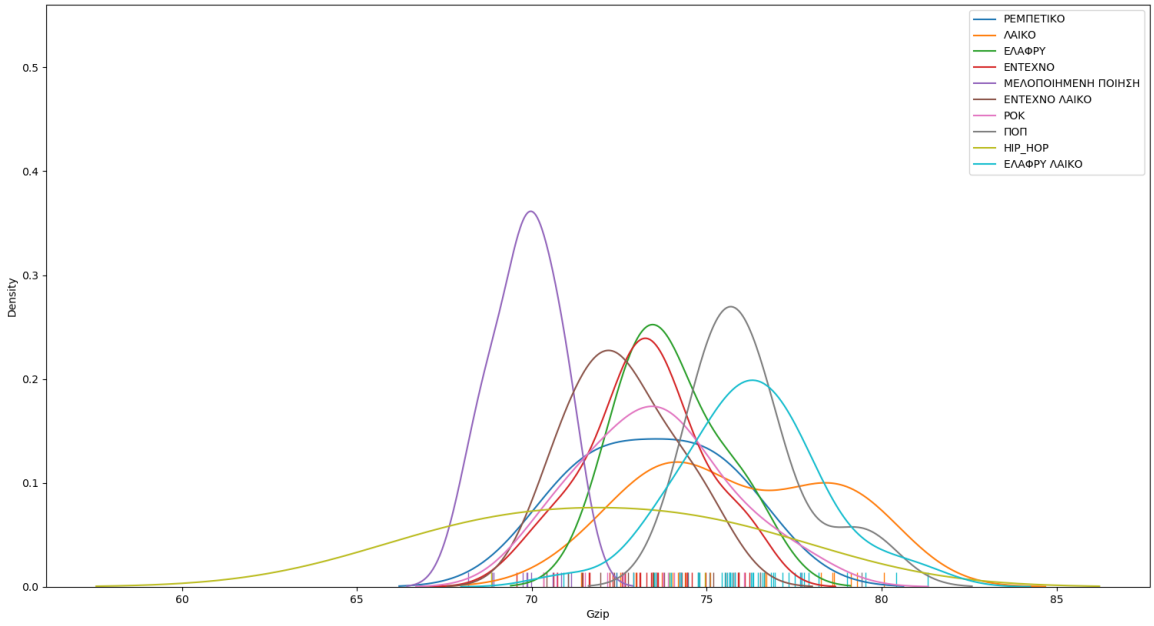


Σχήμα 6.19: Το γράφημα που δείχνει την σχέση μεταξύ των ποσοστών συμπίεσης του Bzip2 και το είδος του τραγουδιού.



Σχήμα 6.20: Το γράφημα που δείχνει την σχέση μεταξύ των ποσοστών συμπίεσης του Huffman και το είδος του τραγουδιού.

Από τα γραφήματα καταλαβαίνουμε ότι οι συμπιεστές που επηρεάζονται περισσότερο από το είδος του τραγουδιού είναι οι Gzip και Bzip2 γεγονός αναμενόμενο από την θεωρία που κατέχουμε καθώς ο Huffman σε σχέση με του υπόλοιπους δύο είναι ένα συμπιεστής μηδενικής τάξης που δεν παίρνει υπόψιν του πέρα από τις συχνότητες των γραμμάτων τι έχει προηγηθεί κατά την συμπίεση του τρέχοντος συμβόλου. Για το λόγο αυτό δεν έχει την ικανότητα να επηρεάζεται από το περιεχόμενο που συμπιέζει. Στα γραφήματα η ευθεία γραμμή που βλέπουμε στο είδος ΠΑΡΑΔΟΣΙΑΚΟ, υπάρχει γιατί το συγκεκριμένο είδος αποτελείται από μία μόνο παρατήρηση. Για να κατανοήσουμε λίγο καλύτερα συμβαίνει παραθέτουμε και το γράφημα των εμπειρικών κατανομών της κάθε κατηγορίας.



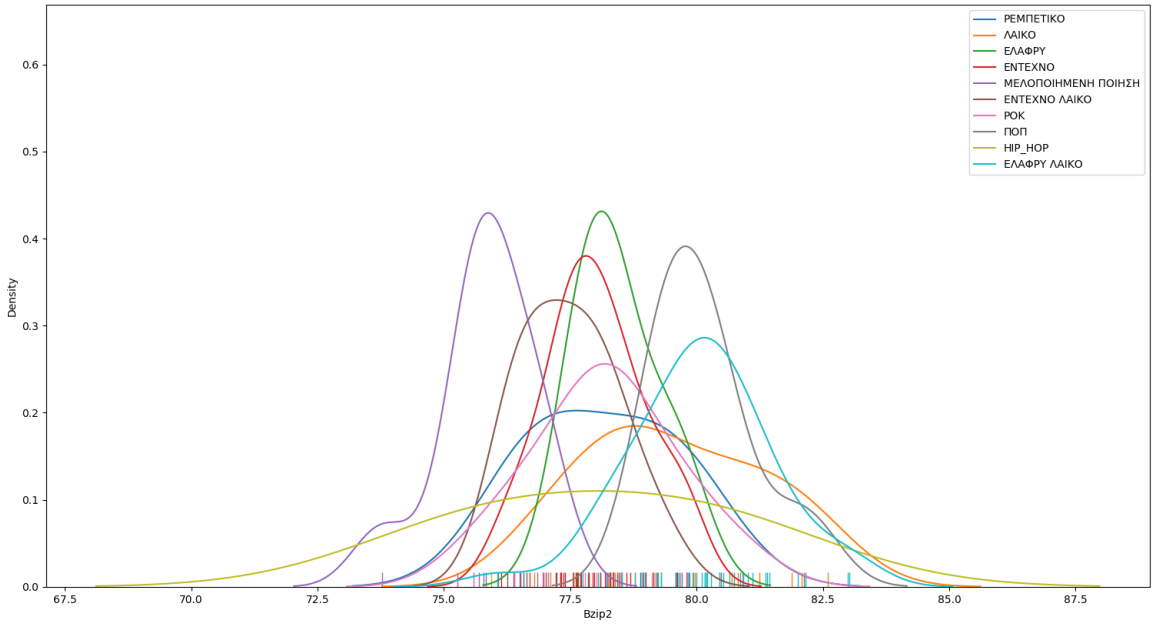
Σχήμα 6.21: Το γράφημα των εμπειρικών κατανομών που δείχνει την σχέση μεταξύ των ποσοστών συμπίεσης του Gzip και το είδος του τραγουδιού.

Στο γράφημα των εμπειρικών κατανομών φαίνεται ξεκάθαρα η διαφορά που υπάρχει ανάμεσα στα διαφορετικά είδη τραγουδιών σε σχέση με τα ποσοστά συμπίεσης που επιτυγχάνονται. Το είδος που ξεχωρίζει είναι η ΜΕΛΟΠΟΙΗΜΕΝΗ ΠΟΙΗΣΗ που επιτυγχάνει τα χαμηλότερα ποσοστά συμπίεσης ενώ προς τα υψηλότερα ποσοστά βαδίζει το ΛΑΙΚΟ , το ΕΛΑΦΡΥ ΛΑΙΚΟ και η ΧΙΠ ΧΟΠ. Αυτά που φαίνονται να είναι προσεγγίζουν τιμές κοντά στους μέσους όρους είναι το ΕΝΤΕΧΝΟ, το ΕΝΤΕΧΝΟ ΛΑΙΚΟ και το ΕΛΑΦΡΥ. Επειδή οι παρατηρήσεις για τα διάφορα είδη έχουν διαφορετικό μέγεθος καθώς και από τα γραφήματα των boxplots αλλά και της εμπειρικής κατανομής φαίνεται ότι δεν ισχύει η υπόθεση της ομοσκεδαστικότητας θα προχωρήσουμε σε μία μη παραμετρική ανάλυση one-way ANOVA χρησιμοποιώντας το Kruskal Wallis test. Οι πληθυσμοί που θα εξαιρεθούν είναι το ΠΑΡΑΔΟΣΙΑΚΟ, η ΧΙΠ ΧΟΠ, η ΠΟΠ και το ΡΕΜΠΕΤΙΚΟ λόγω έλλειψης παρατηρήσεων. Βέβαια τα συμπεράσματα που αναμένουμε από το τεστ μπορούν να γενικευτούν με ασφάλεια και στα άλλα είδη κρίνοντας από τις γραφικές παραστάσεις που έχουν προκύψει.

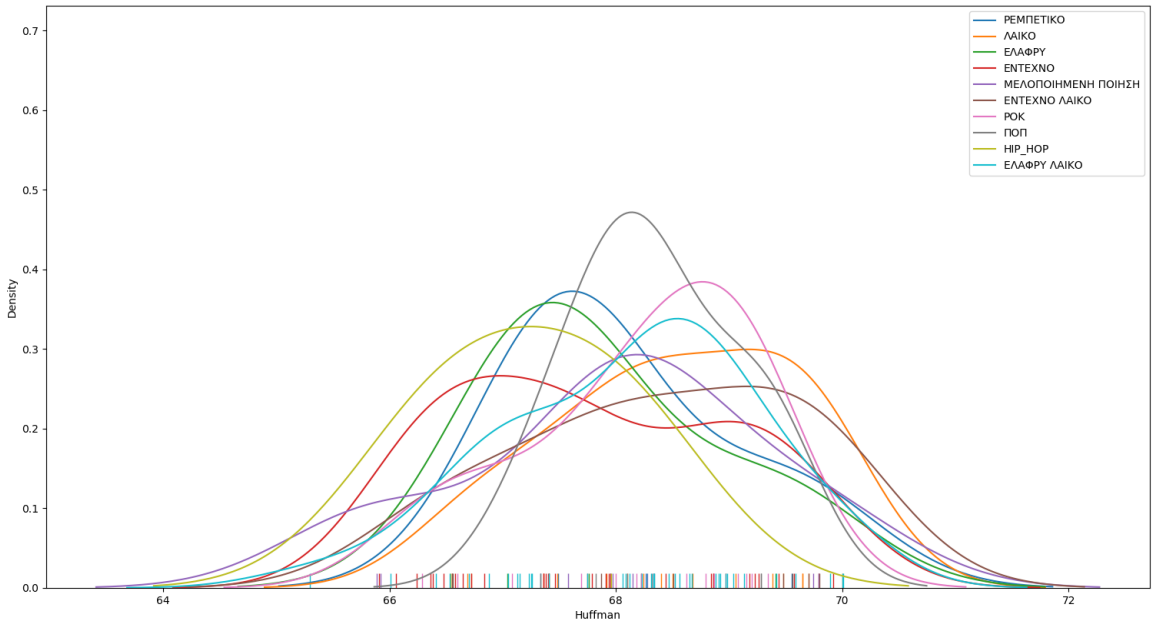
Συμπιεστής	Kruskal-Wallis Test	p-value
Gzip	62.68280677004247	3.3865515911404223e-12

Κάνοντας έναν ακόμη μη παραμετρικό ελέγχο διαπιστώνουμε ότι το ΕΝΤΕΧΝΟ από το ΕΝΤΕΧΝΟ ΛΑΙΚΟ δεν παρουσιάζουν κάποια σημαντική στατιστική διαφορά. Συμπερασματικά λοιπόν παρατηρούμε ότι ο Gzip επηρεάζεται αρκετά από το περιεχόμενο που συμπιέζει και είναι σε θέση να προσαρμόζεται στον ίδιο βαθμό σε παραπλήσια περιεχόμενα.

Συμπιεστής	Kruskal-Wallis Test	p-value
Gzip	1.3224138996244899	0.2501592902832977



Σχήμα 6.22: Το γράφημα των εμπειρικών κατανομών που δείχνει την σχέση μεταξύ των ποσοστών συμπίεσης του Bzip2 και το είδος του τραγουδιού.



Σχήμα 6.23: Το γράφημα των εμπειρικών κατανομών που δείχνει την σχέση μεταξύ των ποσοστών συμπίεσης του Huffman και το είδος του τραγουδιού.

Συνεχίζουμε την ανάλυση μας με τον Bzip2 . Στον Bzip2 (Σχήμα 1.55) βλέπουμε ότι υπάρχει και εδώ μία ευαισθησία ως προς το περιεχόμενο αλλά δεν είναι τόσο μεγάλη όσο στον Gzip. Αυτή μάλλον η ταχύτητα με την οποία προσαρμόζεται είναι που τον κάνει να έχει μία καλύτερη απόδοση από τον Gzip. Μπορούμε να πούμε

ότι για τα δεδομένα μας ο Bzip2 είχε το ιδανικό μείγμα ευαισθησίας και προσαρμοστικότητας. Διεξάγοντας πάλι την μη παραμετρική ANOVA έχουμε:

Συμπιεστής	Kruskal-Wallis Test	p-value
Bzip2	62.49322004109512	3.706932181228923ε-12

Επειδή και σε αυτό το γράφημα βλέπουμε ότι ΕΝΤΕΧΝΟ και το ΕΝΤΕΧΝΟ ΛΑΙΚΟ είναι αρκετά κοντά ως προς τις μέσες τιμές τους διεξάγοντας ένα ακόμη τεστ βλέπουμε ότι και γι αυτόν το συμπιεστή δεν διαφέρουν ιδιαίτερα ως προς την συμπίεση που τους εφαρμόζει.

Ολοκληρώνοντας με την ανάλυση ως προς το είδος βλέπουμε από γράφημα των εμπειρικών κατανομών αυτό που υποψιαζόμασταν ήδη από τα boxplots ότι δηλαδή ο Huffman δεν επηρεάζεται ιδιαίτερα από το περιεχόμενο που συμπιέζει γιατί πολύ απλά δεν το καταλαβαίνει. Στον huffman δοκιμάσαμε πρώτα το τεστ levene προκειμένου να δούμε αν υπάρχει ομοσκεδαστικότητα στις διασπορές. Το τεστ αποφάνθηκε θετικά, οπότε προχωρήσαμε σε μία παραμετρική ανάλυση ANOVA. Επειδή ο αριθμός παρατηρήσεων διαφέρει μεταξύ των ομάδων αποφασίσαμε να τρέξουμε και τον μη παραμετρικό έλεγχο για να είμαστε σίγουροι. Και τα δύο τεστ έδειξαν το ίδιο πράγμα, ότι ο συμπιεστής δεν βλέπει διαφορές ανάμεσα στις διάφορες κατηγορίες τραγουδιών.

Συμπιεστής	Levene's Test	p-value
Huffman	0.40080386405664603	0.847441272193825
Συμπιεστής	One-way ANOVA	p-value
Huffman	1.1419566726923998	0.34218098115455325
Συμπιεστής	Kruskal - Wallis	p-value
Huffman	5.419804866489881	0.36681971789644185

Άρα το σημαντικότερο συμπέρασμα που μπορούμε να βγάλουμε από αυτή την ενότητα είναι πώς όση μικρότερη τάξη έχει ένας συμπιεστής τόσο γρηγορότερα τείνει να συγκλίνει στην εντροπία. Απλά εδώ υπάρχει η εξής διαφοροποίηση. Οι συμπιεστές μηδενικής τάξης όπως ο Huffman που κωδικοποιούν μόνο τις τιμές της τυχαίας μεταβλητής όταν έχουν ένα μεγάλο δείγμα (θεωρητικά $n \to \infty$) να συμπιέσουν τείνουν να συγκλίνουν στις πιθανότητες εμφάνισης των γραμμάτων της γλώσσας και κατ΄ επέκταση στην εντροπία της. Το ενδιαφέρον που προτείνουν τα μαθηματικά είναι πως οι συχνότητες των γραμμάτων δεν έχουν σχέση με το περιεχόμενο. Συγκεκριμένα αν έχουμε μία μεγάλη ακολουθία (όπως τα mega files του πειράματος) $x_1x_2\cdots x_n$ και το κάθε γράμμα έχει θεωρητική πιθανότητα $p_1, p_2, \cdots p_n$ αντίστοιχα τότε από το νόμο των μεγάλων αριθμών ξέρουμε ότι κάθε γράμμα θα το συναντήσουμε μέσα στην συμβολοσειρά $n \cdot p_1, \cdots, n \cdot p_n$ φορές. Η θεωρία λοιπόν δεν διευκρινίζει κάποιο περιεχόμενο για την συμβολοσειρά αλλά μας μιλάει για κάποια συμβολοσειρά μεγάλου μεγέθους που παράχθηκε από την πηγή. Προφανώς τα mega files είναι συμβολοσειρές συμβολοσειρές που παράχθηκαν από την ίδια πηγή, δηλαδή την ελληνική γλώσσα. Αυτό λοιπόν συμβαίνει με τον Huffman καταφέρνει από το αρχείο να εξάγει τις σωστές πιθανότητες εμφάνισης των γραμμάτων της ελληνικής γλώσσας και έτσι φαίνεται ότι δεν επηρεάζεται από το περιεχόμενο που συμπιέζει.

Το ακριβώς ανάποδο συμβαίνει με τους άλλους δύο συμπιεστές. Επειδή ο Gzip και Bzip2 έχουν μνήμη γνωρίζουμε πως θεωρητικά θα πρέπει να συγκλίνουν στον ρυθμό εντροπία της πηγής αν η μνήμη τους επαρκεί η σε μία εντροπία k-τάξης. Αυτό πρακτικά σημαίνει το εξής· Όπως μάθαμε από το θεώρημα ασυμπτωτικής ισοκατανομής για στάσιμες και εργοδικές πηγές με μνήμη οι δυνατές ακολουθίες μήκους n που μπορούν να παραχθούν από μία πηγή (ελληνική γλώσσα) χωρίζονται σε δύο μεγάλες κατηγορίες, αυτές που παράγονται με πολύ μεγάλη πιθανότητα (τυπικό σύνολο) από την πηγή και αυτές που δεν παράγονται σχεδόν ποτέ. Οι συμβολοσειρές μεγάλης πιθανότητας δεν είναι τίποτα παραπάνω από τις εκφράσεις που συναντάμε στην ελληνική γλώσσα. Πάλι με την ίδια λογική αν είχαμε οποιαδήποτε συμβολοσειρά άπειρου μεγέθους αυτές οι ακολουθίες του τυπικού συνόλου θα εμφανίζονταν μέσα στην συμβολοσειρά με συγκεκριμένη συχνότητα ανάλογη της πιθανότητα τους ($\approx \frac{1}{2^{n\cdot H(\mathcal{X})}}$). Τώρα ο κάθε συμπιεστής ανάλογα με το πως είναι υλοποιημένος, δηλαδή με

βάση το μέγεθος της μνήμης του ή καλύτερα τι τάξης είναι, προσεγγίζει την πηγή ως ένα βαθμό.[5] Θυμηθείτε στην απόδειξη του Shannon-McMillan-Breiman φράζαμε την πηγή ανάμεσα σε μία εργοδική διαδικασία μνήμης k και σε μία διαδικασία άπειρης μνήμης. Τότε βλέπαμε πως ασυμπτωτικά οι εντροπίες των δύο διαδικασιών είχαν το ίδιο όριο. Αυτό λοιπόν που συμβαίνει στη πράξη και δεν εφαρμόζεται το αποτέλεσμα της θεωρίας είναι πολύ απλά ότι το μέγεθος των Mega files δεν επαρκεί για να εξάγουν σωστά τα στατιστικά στοιχεία της πηγής, δηλαδή αυτό που λέμε θεωρητικά "όταν το n γίνεται πολύ μεγάλο" φαίνεται ότι το μέγεθος των Mega files δεν ήταν αρκετά μεγάλο με αποτέλεσμα οι συμπιεστές να μην μπορούν να βρουν όλες τις τυπικές ακολουθίες που παράγει η πηγή οπότε να εξαρτώνται από αυτό που λέμε περιεχόμενο.

Το περιεχόμενο όμως ενός κειμένου είναι απλά κάποιες λέξεις, γράμματα ή εκφράσεις που εμφανίζονται πολύ συχνά σε ένα συγκεκριμένο κείμενο άρα αποτελεί στην ουσία κάποια προσέγγιση του τυπικού συνόλου. Με τον όρο "προσέγγιση του τυπικού συνόλου" εννοούμε πρακτικά το υποσύνολο που περιέχει υποσυμβολοσειρές που εμφανίζονται στις συμβολοσειρές του τυπικού συνόλου. Για παράδειγμα αν η ελληνική γλώσσα παράγει με μεγάλη πιθανότητα κάποιες συγκεκριμένες εκφράσεις μεγάλου μήκους τότε οι λέξεις που εμφανίζονται σε αυτές τις εκφράσεις ή ακολουθίες λέξεων θα αποτελούν υποσυμβολοσειρές των τυπικών ακολουθιών. Παρόλα αυτά είδαμε ότι οι συμπιεστές πέτυχαν μεγάλους βαθμούς ποσοστών συμπίεσης. Αυτό έγινε γιατί μπόρεσαν να προσεγγίσουν σε πολύ μεγάλο βαθμό κάποιο υποσύνολο του τυπικού συνόλου δηλαδή το περιεχόμενο που συμπίεζαν ίσως ακόμα και κάποιες από τις ακολουθίες του τυπικού συνόλου και επειδή κάποιες υποσυμβολοσειρές φαίνεται ότι είναι κοινές για όλα τα αρχεία το αποτέλεσμα ήταν τα ποσοστά συμπίεσης να συγκεντρώνονται γύρω από το μέσο όρο. Ένα ακόμη ωραίο σημείο που αξίζει να παρατηρηθεί είναι ότι ο Bzip2 που υλοποιείται με περισσότερη μνήμη (400 Kbytes-900Kbytes από τον Gzip 32Kbytes πέτυχε καλύτερη συμπίεση και μικρότερη διασπορά (ευαισθησία) στην αλλαγή περιεχομένου. Για τα mega files πρακτικά το μέγεθος μνήμης του Bzip2 είναι συνώνυμο με το άπειρη μνήμη.

Ολοκληρώνοντας αυτή την ενότητα παραθέτουμε τον κώδικα με τον οποίο δημιουργήθηκαν τα παραπάνω διαγράμματα και διενεργήθηκαν οι έλεγχοι.

Compression_1_Vs_Genre.py

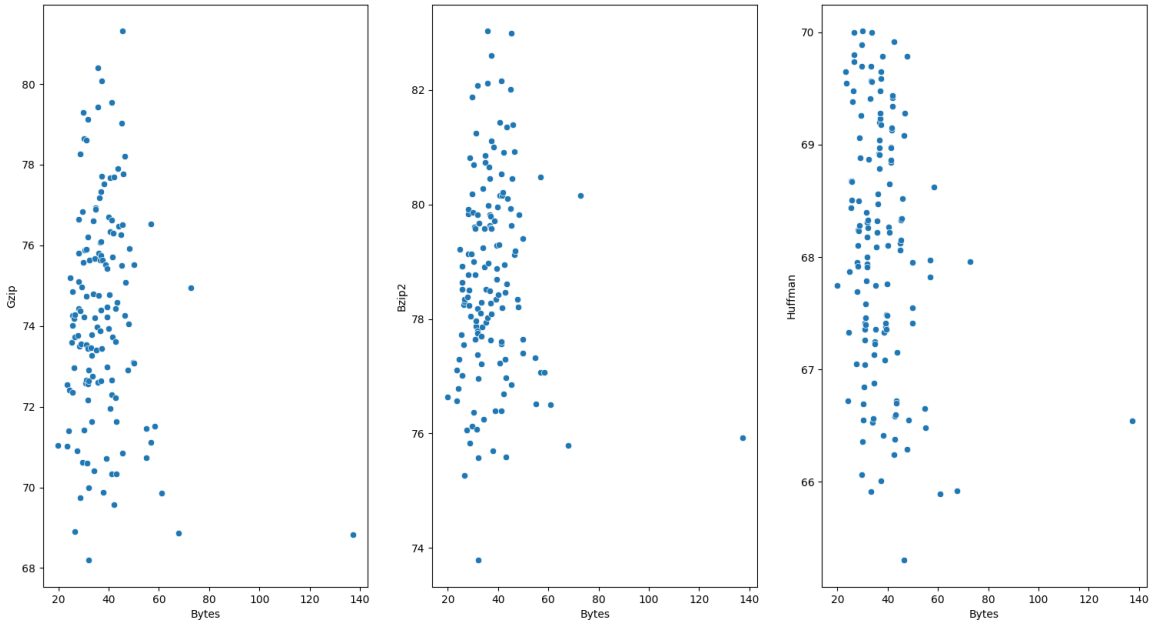
```
1
2
3  import pandas as pd
4  import matplotlib.pyplot as plt
5  import os
6  from scipy import stats
7  import seaborn as sns
8  table = pd.read_csv(os.getcwd() + '/Experiment_1.csv', index_col=0)
9
10 g_data = table['Gzip']
11 b_data = table['Bzip2']
12 h_data = table['Huffman']
13 size=table['Bytes']
14
15
16 plot_1=sns.boxplot(data=table,x='Είδος',y='Gzip')
17 plt.setp(plot_1.get_xticklabels(), rotation=30)
18 ax=plt.axes
19 plot_1.set_title('Το γράφημα του Γζιπ σε σχέση με τα είδη τραγουδιών')
20
21 plt.show()
22
23 plot_2=sns.boxplot(data=table,x='Είδος',y='Bzip2')
24 plt.setp(plot_2.get_xticklabels(), rotation=30)
25 plot_2.set_title('Το γράφημα του Βζιπ2 σε σχέση με τα είδη τραγουδιών')
26
27 plt.show()
28 plot_3=sns.boxplot(data=table,x='Είδος',y='Huffman')
29 plt.setp(plot_3.get_xticklabels(), rotation=30)
30 plot_3.set_title('Το γράφημα του Hυφφμαν σε σχέση με τα είδη τραγουδιών')
31 plt.show()
32
33
34 table_PARADOSIAKO=table.loc[table['Είδος']=='ΠΑΡΑΔΟΣΙΑΚΟ']
```

[5] Για να τη προσέγγιζε τέλεια έπρεπε η μνήμη του να ήταν άπειρη.

```
table_REMPETIKO=table.loc[table['Είδος']=='ΡΕΜΠΕΤΙΚΟ']
table_LAIKO=table.loc[table['Είδος']=='ΛΑΙΚΟ']
table_ELAFRI=table.loc[table['Είδος']=='ΕΛΑΦΡΥ ΤΡΑΓΟΥΔΙ']
table_ENTEXNO=table.loc[table['Είδος']=='ΕΝΤΕΧΝΟ']
table_MEL_P=table.loc[table['Είδος']=='ΜΕΛΟΠΟΙΗΜΕΝΗ ΠΟΙΗΣΗ']
table_ENTEXNO_LAIKO=table.loc[table['Είδος']=='ΕΝΤΕΧΝΟ ΛΑΙΚΟ']
table_ROK=table.loc[table['Είδος']=='ΡΟΚ']
table_EL_LAIKO=table.loc[table['Είδος']=='ΕΛΑΦΡΥ ΛΑΙΚΟ']
table_POP=table.loc[table['Είδος']=='ΠΟΠ']
table_HIP_HOP=table.loc[table['Είδος']=='ΧΙΠ ΧΟΠ']

sns.distplot(table_REMPETIKO['Gzip'],hist=False,kde=True,rug=True,label='ΡΕΜΠΕΤΙΚΟ')
sns.distplot(table_LAIKO['Gzip'],hist=False,kde=True,rug=True,label='ΛΑΙΚΟ')
sns.distplot(table_ELAFRI['Gzip'],hist=False,kde=True,rug=True,label='ΕΛΑΦΡΥ')
sns.distplot(table_ENTEXNO['Gzip'],hist=False,kde=True,rug=True,label='ΕΝΤΕΧΝΟ')
sns.distplot(table_MEL_P['Gzip'],hist=False,kde=True,rug=True,label='ΜΕΛΟΠΟΙΗΜΕΝΗ ΠΟΙΗΣΗ')
sns.distplot(table_ENTEXNO_LAIKO['Gzip'],hist=False,kde=True,rug=True,label='ΕΝΤΕΧΝΟ ΛΑΙΚΟ')
sns.distplot(table_ROK['Gzip'],hist=False,kde=True,rug=True,label='ΡΟΚ')
sns.distplot(table_POP['Gzip'],hist=False,kde=True,rug=True,label='ΠΟΠ')
sns.distplot(table_HIP_HOP['Gzip'],hist=False,kde=True,rug=True,label='HIP_HOP')
sns.distplot(table_EL_LAIKO['Gzip'],hist=False,kde=True,rug=True,label='ΕΛΑΦΡΥ ΛΑΙΚΟ')
plt.legend()
plt.show()
sns.distplot(table_REMPETIKO['Bzip2'],hist=False,kde=True,rug=True,label='ΡΕΜΠΕΤΙΚΟ')
sns.distplot(table_LAIKO['Bzip2'],hist=False,kde=True,rug=True,label='ΛΑΙΚΟ')
sns.distplot(table_ELAFRI['Bzip2'],hist=False,kde=True,rug=True,label='ΕΛΑΦΡΥ')
sns.distplot(table_ENTEXNO['Bzip2'],hist=False,kde=True,rug=True,label='ΕΝΤΕΧΝΟ')
sns.distplot(table_MEL_P['Bzip2'],hist=False,kde=True,rug=True,label='ΜΕΛΟΠΟΙΗΜΕΝΗ ΠΟΙΗΣΗ')
sns.distplot(table_ENTEXNO_LAIKO['Bzip2'],hist=False,kde=True,rug=True,label='ΕΝΤΕΧΝΟ ΛΑΙΚΟ')
sns.distplot(table_ROK['Bzip2'],hist=False,kde=True,rug=True,label='ΡΟΚ')
sns.distplot(table_POP['Bzip2'],hist=False,kde=True,rug=True,label='ΠΟΠ')
sns.distplot(table_HIP_HOP['Bzip2'],hist=False,kde=True,rug=True,label='HIP_HOP')
sns.distplot(table_EL_LAIKO['Bzip2'],hist=False,kde=True,rug=True,label='ΕΛΑΦΡΥ ΛΑΙΚΟ')
plt.legend()
plt.show()

sns.distplot(table_REMPETIKO['Huffman'],hist=False,kde=True,rug=True,label='ΡΕΜΠΕΤΙΚΟ')
sns.distplot(table_LAIKO['Huffman'],hist=False,kde=True,rug=True,label='ΛΑΙΚΟ')
sns.distplot(table_ELAFRI['Huffman'],hist=False,kde=True,rug=True,label='ΕΛΑΦΡΥ')
sns.distplot(table_ENTEXNO['Huffman'],hist=False,kde=True,rug=True,label='ΕΝΤΕΧΝΟ')
sns.distplot(table_MEL_P['Huffman'],hist=False,kde=True,rug=True,label='ΜΕΛΟΠΟΙΗΜΕΝΗ ΠΟΙΗΣΗ')
sns.distplot(table_ENTEXNO_LAIKO['Huffman'],hist=False,kde=True,rug=True,label='ΕΝΤΕΧΝΟ ΛΑΙΚΟ')
sns.distplot(table_ROK['Huffman'],hist=False,kde=True,rug=True,label='ΡΟΚ')
sns.distplot(table_POP['Huffman'],hist=False,kde=True,rug=True,label='ΠΟΠ')
sns.distplot(table_HIP_HOP['Huffman'],hist=False,kde=True,rug=True,label='HIP_HOP')
sns.distplot(table_EL_LAIKO['Huffman'],hist=False,kde=True,rug=True,label='ΕΛΑΦΡΥ ΛΑΙΚΟ')

plt.legend()
plt.show()


g_results=stats.kruskal(table_LAIKO['Gzip'],table_EL_LAIKO['Gzip'],table_ENTEXNO['Gzip'],table_MEL_P['Gzip'],
                        table_ENTEXNO_LAIKO['Gzip'],table_ROK['Gzip'])
print(g_results)
b_results=stats.kruskal(table_LAIKO['Bzip2'],table_EL_LAIKO['Bzip2'],table_ENTEXNO['Bzip2'],table_MEL_P['Bzip2'],
                        table_ENTEXNO_LAIKO['Bzip2'],table_ROK['Bzip2'])
print(b_results)
h_results=stats.kruskal(table_LAIKO['Huffman'],table_EL_LAIKO['Huffman'],table_ENTEXNO['Huffman'],
                        table_MEL_P['Huffman'],table_ENTEXNO_LAIKO['Huffman'],table_ROK['Huffman'])
print(h_results)
EN_ENLA_results=stats.kruskal(table_ENTEXNO['Gzip'],table_ENTEXNO_LAIKO['Gzip'])
print(EN_ENLA_results)
EN_ENLA_results_2=stats.kruskal(table_ENTEXNO['Bzip2'],table_ENTEXNO_LAIKO['Bzip2'])
print(EN_ENLA_results_2)
h_results_var=stats.levene(table_LAIKO['Huffman'],table_EL_LAIKO['Huffman'],table_ENTEXNO['Huffman'],
                           table_MEL_P['Huffman'],table_ENTEXNO_LAIKO['Huffman'],table_ROK['Huffman'])
print(h_results_var)
h_results_2=stats.f_oneway(table_LAIKO['Huffman'],table_EL_LAIKO['Huffman'],table_ENTEXNO['Huffman'],
                           table_MEL_P['Huffman'],table_ENTEXNO_LAIKO['Huffman'],table_ROK['Huffman'],table_REMPETIKO['Huffman'])
print(h_results_2)
```

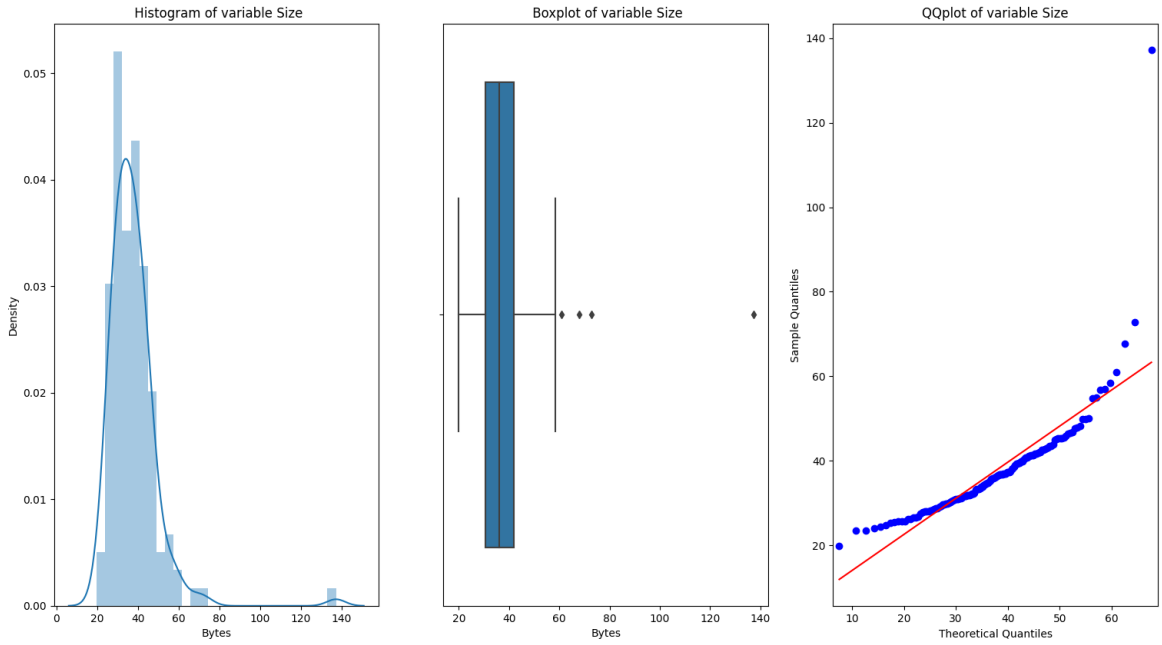
6.3.3 Η συμπίεση σε σχέση με το μέγεθος των αρχείων

Επειδή το μέγεθος και τα ποσοστά συμπίεσης αποτελούν αριθμητικές μεταβλητές αρχικά για να δούμε αν υπάρχει κάποια συσχέτιση μεταξύ των δύο θα προχωρήσουμε στην κατασκευή ενός scatterplot.



Σχήμα 6.24: Τα διαγράμματα σκέδαση μεταξύ του μεγέθους των αρχείων και των συμπιεστών.

Από τα διαγράμματα σκέδαση δεν φαίνεται ότι υπάρχει κάποια συσχέτιση μεταξύ του μεγέθους των αρχείων και το ποσοστό συμπίεσης. Μάλιστα για το ίδιο διάστημα μεγέθους μεταξύ 40000 και 60000 bytes παρατηρούμε ότι μπορούμε να συναντήσουμε όλα τα ποσοστά συμπίεσης. Για να είμαστε σίγουροι για την υπόθεση που κάνουμε θα ελέγξουμε την ύπαρξη συσχέτισης και με ένα στατιστικό τεστ.



Σχήμα 6.25: Τα διαγράμματα ελέγχου κανονικότητας της μεταβλητής μέγεθος τραγουδιών (Size).

Μεταβλητή	Kolmogorov-Smirnov	p-value
Size	=0.13120499689409915	0.013638390366182095

Από τα διαγράμματα ελέγχου κανονικότητας και το στατιστικό τεστ φαίνεται ότι η μεταβλητή Size δεν ακολουθεί την κανονική κατανομή. Για τον λόγο αυτό προκειμένου να ελέγξουμε αν υπάρχει συσχέτιση με τα ποσοστά συμπίεσης θα χρησιμοποιήσουμε τον μη παραμετρικό συντελεστή συσχέτισης Spearman

Συμπιεστής	Μεταβλητή	Spearman's	p-value
Gzip	Size	0.07591355599418342	0.36923066289650663
Bzip2	Size	0.13196655782439073	0.11746136899520368
Huffman	Size	-0.17654837404755386	0.03557425804483475

Επειδή οι συντελεστές συσχέτισης βρίσκονται μακριά από το +1,-1 , μπορούμε με ασφάλεια να δεχθούμε ότι οι συμπιεστές στο συγκεκριμένο πείραμα δεν επηρεάζονται από το μέγεθος των αρχείων. Να σημειώσουμε πως παρόλο που το p-value του Huffman είναι > 0.05 αν κοιτάξουμε τον συντελεστή και το γράφημα θα δούμε πως δεν υπάρχει κάποια συσχέτιση μεταξύ του Huffman και του μεγέθους των αρχείων. Έτσι κι αλλιώς το documentation για τον συγκεκριμένο συντελεστή συσχέτισης μας προειδοποιεί να μη έχουμε μεγάλη εμπιστοσύνη στο p-value που δίνει.

Το παραπάνω αποτέλεσμα μπορεί φαινομενικά να έρχεται σε σύγκρουση με την προηγούμενη ενότητα αλλά δεν είναι έτσι. Αυτό που συμβαίνει σε αυτό το πείραμα είναι ότι το μέγεθος των mega files επαρκεί ώστε οι συμπιεστές να εξάγουν κάποια από τα στατιστικά στοιχεία της πηγής (ελληνική γλώσσα) ώστε το ποσοστό συμπίεση του κάθε συμπιεστή να συγκεντρώνεται γύρω από ένα μέσο όρο ανάλογο της τάξης του συμπιεστή. Σε αυτό που δεν επαρκεί είναι στο να συγκλίνει ο συμπιεστής στον θεωρητικό ρυθμό εντροπίας της πηγής. Προσοχή όμως!! Ακόμα και τα αρχεία μας να ήταν πολύ μεγάλα μπορεί η τάξη του κάθε συμπιεστή να μην είναι η ασυμπτωτική τάξη της διαδικασίας από την οποία μπορούμε να προσεγγίσουμε τον ρυθμό εντροπίας της πηγής. Για παράδειγμα τα 32Kbytes του Gzip που αν μιλάμε για κείμενο γραμμένο σε κωδικοποίηση ASCII αντιστοιχούν σε 32000 σύμβολα του παρελθόντος. Πάντως αυτό που θα συνέβαινε αν είχαμε δύο συμπιεστές που η τάξης τους ταυτίζονταν με την ασυμπτωτική τάξη της πηγής και κείμενα πολύ μεγάλου μεγέθους θα ήταν

αυτοί οι δύο συμπιεστές να πετυχαίνουν τους ίδιους μέσους όρους συμπίεσης. Αλλά όλα αυτά τα ζητήματα που αφορούν στα μεγέθη αρχείων και στις μνήμες των συμπιεστών από τις οποίες μπορούμε να εξάγουμε την εντροπία μία γλώσσας είναι ερευνητικά και δεν άπτονται τον σκοπό αυτού του πειράματος και αυτής της διπλωματικής.

Ανακεφαλαιώνοντας από το πρώτο πείραμα συμπεράναμε:

1. Τα ποσοστά συμπίεσης και των τριών συμπιεστών ακολουθούν την κανονική κατανομή

2. Την καλύτερη επίδοση την είχε ο Bzip2, ακολούθησε ο Gzip και ήρθε τελευταίος ο Huffman το οποίο με τη σειρά του συνεπάγεται πως όσο μεγαλύρτη τάξη έχει ένας συμπιεστής τόσο καλύτερα ποσοστά συμπίεσης πετυχαίνει.

3. Ο πιο ευαίσθητος στις αλλαγές περιεχομένου φαίνεται ήταν ο Gzip καθώς είχε και την μεγαλύτερη διασπορά. Στην άλλη άκρη βρίσκεται ο Huffman που το περιεχόμενο δεν επηρεάζει καθόλου την συμπίεση του. item Τέλος είδαμε πως κανένας από τους συμπιεστές δεν επηρεάστηκε από το μέγεθος σε αυτό το πείραμα. Για το συγκεκριμένο συμπέρασμα διατηρούμε μία επιφύλαξη και θα το αναδιατυπώσουμε όταν ολοκληρωθεί κα το δεύτερο πείραμα. Ολοκληρώνοντας το πείραμα δίνουμε τον κώδικα με τον οποίο κατασκευάστηκαν οι έλεγχοι και τα γραφήματα της τελευταία ενότητας.

Compression_1_Vs_Size.py

```
1
2  import pandas as pd
3  import matplotlib.pyplot as plt
4  import os
5  from scipy import stats
6  import seaborn as sns
7
8  table = pd.read_csv(os.getcwd() + '/Experiment_1.csv', index_col=0)
9  import statsmodels.api as sm
10
11 g_data = table['Gzip']
12 b_data = table['Bzip2']
13 h_data = table['Huffman']
14 size = table['Bytes']
15
16 params = stats.norm.fit(size)
17 result = stats.kstest(size, 'norm', params)
18 print(result)
19 mu = params[0]
20 var = params[1]
21
22 g = stats.spearmanr(size, g_data)
23 print(g)
24 b = stats.spearmanr(size, b_data)
25 print(b)
26 h = stats.spearmanr(size, h_data)
27 print(h)
28
29 fig_2, axes_2 = plt.subplots(1, 3)
30 sns.distplot(size, kde=True, hist=True, ax=axes_2[0])
31 axes_2[0].set_title('Histogram of variable Size')
32 sns.boxplot(data=table, x='Bytes', ax=axes_2[1])
33 axes_2[1].set_title('Boxplot of variable Size')
34 sm.qqplot(size, dist=stats.norm, loc=mu, scale=var, line='r', ax=axes_2[2])
35 axes_2[2].set_title('QQplot of variable Size')
36 plt.show()
37
38 fig, axes = plt.subplots(1, 3)
39 sns.scatterplot(data=table, x='Bytes', y='Gzip', ax=axes[0])
40 sns.scatterplot(data=table, x='Bytes', y='Bzip2', ax=axes[1])
41 sns.scatterplot(data=table, x='Bytes', y='Huffman', ax=axes[2])
42 plt.show()
43
44
45 table_PARADOSIAKO=table.loc[table['Είδος']=='ΠΑΡΑΔΟΣΙΑΚΟ']
```

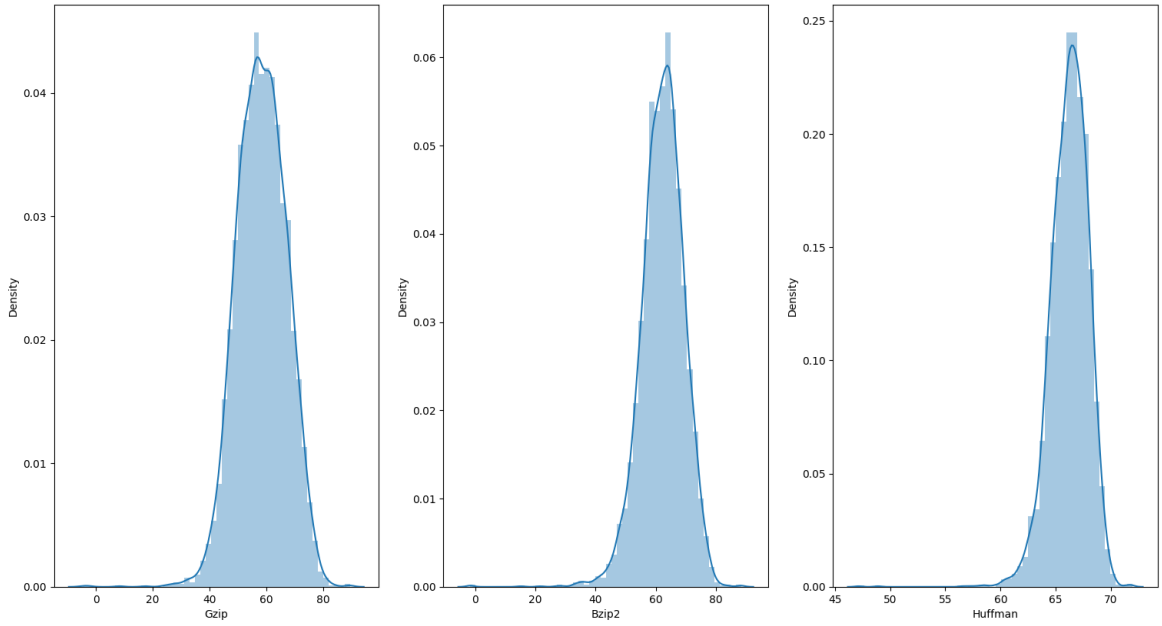
```
table_REMPETIKO=table.loc[table['Είδος']=='ΡΕΜΠΕΤΙΚΟ']
table_LAIKO=table.loc[table['Είδος']=='ΛΑΙΚΟ']
table_ELAFRI=table.loc[table['Είδος']=='ΕΛΑΦΡΥ ΤΡΑΓΟΥΔΙ']
table_ENTEXNO=table.loc[table['Είδος']=='ΕΝΤΕΧΝΟ']
table_MEL_P=table.loc[table['Είδος']=='ΜΕΛΟΠΟΙΗΜΕΝΗ ΠΟΙΗΣΗ']
table_ENTEXNO_LAIKO=table.loc[table['Είδος']=='ΕΝΤΕΧΝΟ ΛΑΙΚΟ']
table_ROK=table.loc[table['Είδος']=='ΡΟΚ']
table_EL_LAIKO=table.loc[table['Είδος']=='ΕΛΑΦΡΥ ΛΑΙΚΟ']
table_POP=table.loc[table['Είδος']=='ΠΟΠ']
table_HIP_HOP=table.loc[table['Είδος']=='ΧΙΠ ΧΟΠ']


sns.distplot(table_REMPETIKO['Bytes'],hist=False,kde=True,rug=True,label='ΡΕΜΠΕΤΙΚΟ')
sns.distplot(table_LAIKO['Bytes'],hist=False,kde=True,rug=True,label='ΛΑΙΚΟ')
sns.distplot(table_ELAFRI['Bytes'],hist=False,kde=True,rug=True,label='ΕΛΑΦΡΥ')
sns.distplot(table_ENTEXNO['Bytes'],hist=False,kde=True,rug=True,label='ΕΝΤΕΧΝΟ')
sns.distplot(table_MEL_P['Bytes'],hist=False,kde=True,rug=True,label='ΜΕΛΟΠΟΙΗΜΕΝΗ ΠΟΙΗΣΗ')
sns.distplot(table_ENTEXNO_LAIKO['Bytes'],hist=False,kde=True,rug=True,label='ΕΝΤΕΧΝΟ ΛΑΙΚΟ')
sns.distplot(table_ROK['Bytes'],hist=False,kde=True,rug=True,axlabel='ΡΟΚ')
sns.distplot(table_POP['Bytes'],hist=False,kde=True,rug=True,label='ΠΟΠ')
sns.distplot(table_POP['Bytes'],hist=False,kde=True,rug=True,label='ΗΙΡ_ΗΟΡ')
plt.legend()
plt.show()

plot_1=sns.boxplot(data=table,x='Είδος',y='Bytes')
plt.setp(plot_1.get_xticklabels(), rotation=30)
ax=plt.axes
plot_1.set_title('Το γράφημα του μεγέθους των αρχείων σε σχέση με τα είδη τραγουδιών')
plt.show()
```

6.4 Επεξεργασία-Παρουσίαση Αποτελεσμάτων-Πείραμα 2

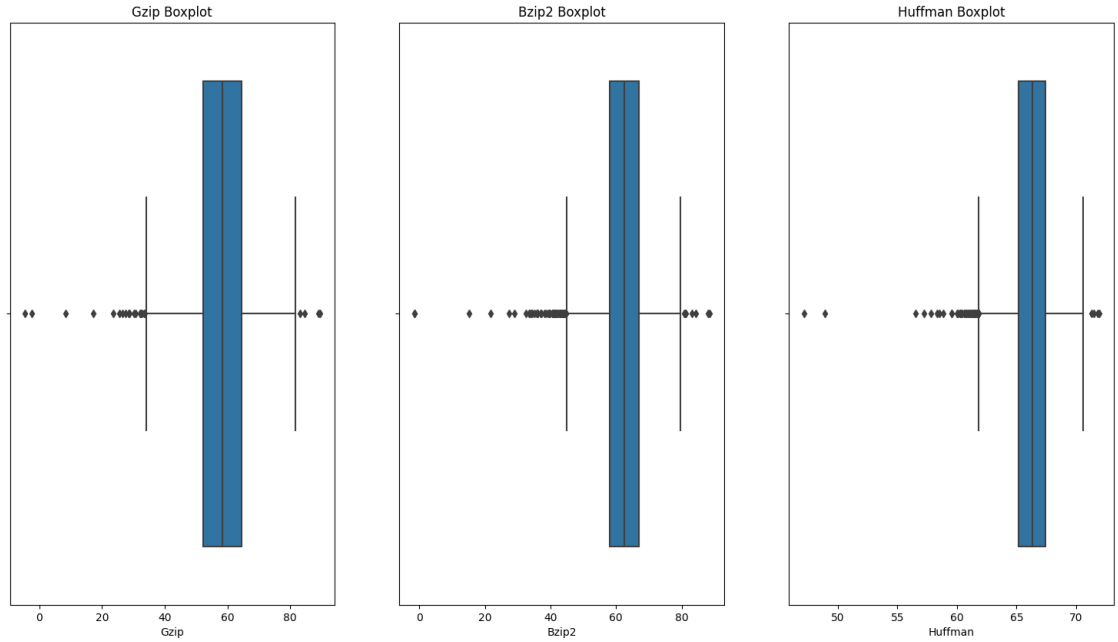
Στο δεύτερο πείραμα θα ασχοληθούμε με τα τραγούδια των στιχουργών τα οποία συμπιέστηκαν το κάθε ένα ξεχωριστά ώστε να εξετάσουμε την συμπεριφορά των συμπιεστών απέναντι σε μικρά αρχεία και εναλλαγές περιεχομένου. Επειδή για να αποδώσουμε το είδος βασιστήκαμε στο ύφος που είχε το μεγαλύτερο μέρος των τραγουδιών ενδέχεται σε αυτό το πείραμα να υπάρχουν ορισμένα τραγούδια τα οποία δεν αντιστοιχούν στο είδος που έχουμε προσδώσει στον στιχουργό. Αυτά όμως αποτελούν μία μικρή μειοψηφία μέσα στο δείγμα μας που δεν πρόκειται να αλλάξει την εγκυρότητα της γενικής απόδοσης των ειδών.

6.4.1 Η κατανομή των ποσοστών συμπίεσης

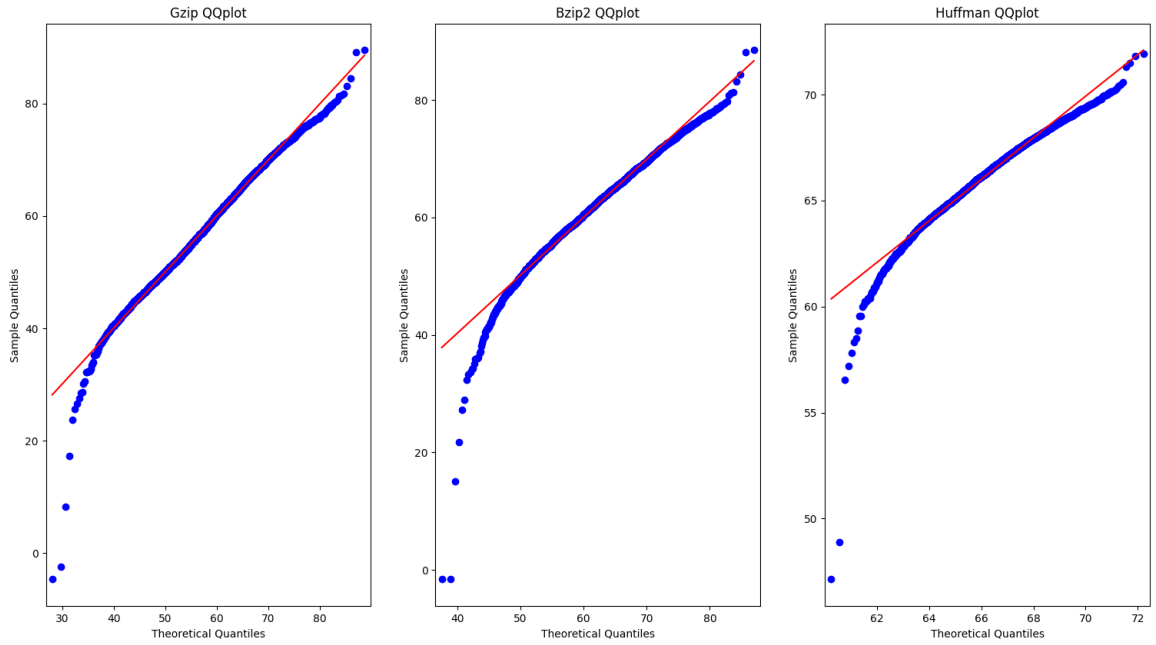
Θα ξεκινήσουμε πάλι προσπαθώντας να προσδιορίσουμε την κατανομή τον ποσοστών συμπίεσης των δεδομένων. Αρχίζουμε πάλι παρουσιάζοντας κάποια γραφήματα.



Σχήμα 6.26: Τα ιστογράμματα των ποσοστών συμπίεσης μαζί με τις εμπειρικές κατανομές



Σχήμα 6.27: Τα boxplots των ποσοστών συμπίεσης

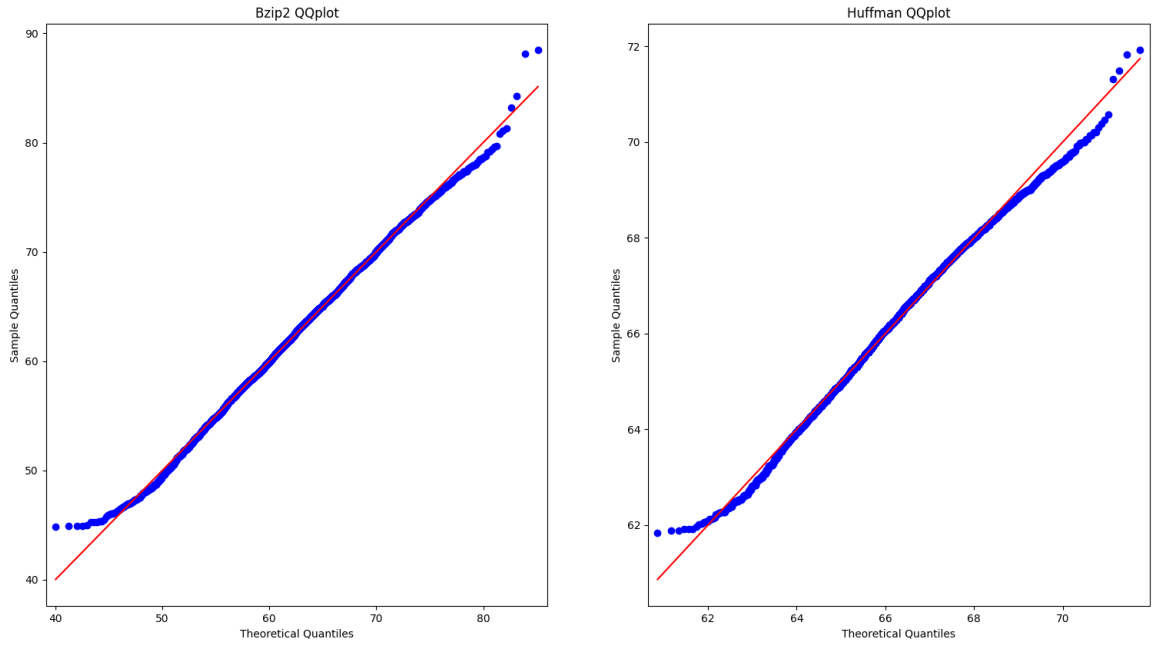


Σχήμα 6.28: Τα QQ plots των ποσοστών συμπίεσης

Συμπιεστής	Kolmogorov-Smirnoff	p-value
Gzip	0.013967437900664237,	0.3735938201754728
Bzip2	0.029177692786945153	0.001389240096863522
Huffman	0.036640568081122316	2.102522280548852e-05

Αυτό που βλέπουμε αρχικά σύγκριση με το πρώτο πείραμα με την βοήθεια του τεστ Kolmogorov-Smirnoff είναι πως χάνεται η υπόθεσης της κανονικότητας για δύο από του τρεις συμπιεστές, συγκεκριμένα τους Bzip και Huffman. Αν παρατηρήσουμε λίγο καλύτερα τα boxplots και QQ plots για τους συμπιεστές που σταμάτησαν να ακολουθούν την κανονική κατανομή, θα δούμε πως υπάρχει μία πληθώρα από ακραίες παρατηρήσεις που φαίνεται πως κάνουν την κατανομή να απομακρύνεται από την κανονική. Για το λόγο αυτό χρησιμοποιώντας τον κανόνα του ενδοτεταρτουμοριακού εύρους (Q1-1.5IQR, Q3+1.5IQR) θα βρούμε τις ακραίες παρατηρήσεις για του συγκεκριμένους συμπιεστές και θα τις απαλείψουμε από το δείγμα μας προκειμένου να δούμε αν αυτό θα φτιάξει τις κατανομές.

Συμπιεστής	Kolmogorov-Smirnoff	p-value
Bzip2	0.012280944677804273	0.5458923890034872
Huffman	0.02626625036275887	0.005900824915349666



Σχήμα 6.29: Τα QQ plots των συμπιεστών Bzip2, Huffman μετά την απαλοιφή των ακραίων τιμών

Από τα καινούριο τεστ βλέπουμε ότι ο Bzip2 μετά την απαλοιφή των ακραίων τιμών άρχισε να ακολουθεί και πάλι την κανονική κατανομή. Για το Huffman το τεστ δείχνει ότι η κατάσταση διορθώθηκε οριακά, το p-value είναι λίγο πιο πάνω από το 0.05 και το γράφημα δείχνει μία συμμόρφωση προς την κανονικότητα . Αυτό λοιπόν που μπορούμε να πούμε συμπερασματικά είναι ότι λόγω ακραίων παρατηρήσεων σταμάτησαν να ακολουθούν την κανονική κατανομή με εξαίρεση τον Gzip που δείχνει να είναι ο πιο ανθεκτικός στις ακραίες παρατηρήσεις. Από την άλλη έχουμε τον Bzip2 ο οποίος ναι μεν επηρεάστηκε από τις ακραίες παρατηρήσεις αλλά ήταν δυνατόν να διορθωθεί ενώ αυτός που δεν φαίνεται είναι ο Huffman.

Για τα ποσοστά συμπίεσης κοιτάζοντας τα γραφήματα έχουμε ότι οι μέσοι όροι των Gzip και Bzip2 κυμαίνονται γύρω στο 60% αλλά παρουσιάζουν και οι δύο μεγάλη διασπορά, ενώ τη μικρότερη διασπορά και καλύτερο μέσο όρο, γύρω στο 65% δείχνει να έχει ο Huffman. Ολοκληρώνοντας όπως κάθε φορά δίνουμε τον κώδικας που χρησιμοποιήσαμε.

Experiment_2_Normality.py

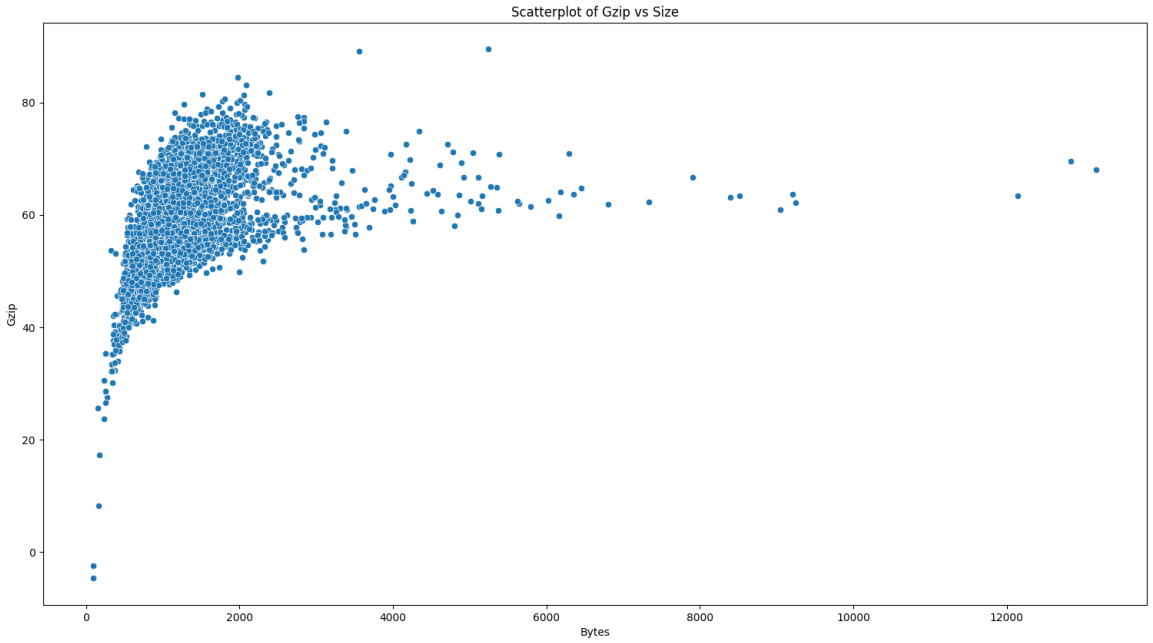
```
1
2  import pandas as pd
3  import matplotlib.pyplot as plt
4  import os
5  from scipy import stats
6  import seaborn as sns
7  import statsmodels.api as sm
8
9  table = pd.read_csv(os.getcwd() + '/Experiment_2.csv', index_col=0)
10 print(len(table.index))
11 indexes = table.loc[table['Στιχουργός'] == 'Αξιώτης Άγγελος'].index
12 table.drop(indexes, inplace=True)
13 print(len(table.index))
14 g_data = table['Gzip']
15 b_data = table['Bzip2']
16 h_data = table['Huffman']
17 size = table['Bytes']
18 mu = []
19 var = []
20 for data in [g_data, b_data, h_data]:
21     params = stats.norm.fit(data)
22     results = stats.kstest(data, 'norm', params)
```

```
23      mu.append(params[0])
24      var.append(params[1])
25      print(results)
26
27  f, axes = plt.subplots(1, 3)
28  sns.distplot(g_data, kde=True, ax=axes[0])
29  sns.distplot(b_data, kde=True, ax=axes[1])
30  sns.distplot(h_data, kde=True, ax=axes[2])
31  plt.show()
32
33  f_2, axes_2 = plt.subplots(1, 3)
34  sns.boxplot(g_data, ax=axes_2[0])
35  axes_2[0].set_title('Gzip Boxplot')
36  sns.boxplot(b_data, ax=axes_2[1])
37  axes_2[1].set_title('Bzip2 Boxplot')
38  sns.boxplot(h_data, ax=axes_2[2])
39  axes_2[2].set_title('Huffman Boxplot')
40  plt.show()
41
42  f_3, axes_3 = plt.subplots(1, 3)
43  sm.qqplot(g_data, dist=stats.norm, loc=mu[0], scale=var[0], line='r', ax=axes_3[0])
44  axes_3[0].set_title('Gzip QQplot')
45  sm.qqplot(b_data, dist=stats.norm, loc=mu[1], scale=var[1], line='r', ax=axes_3[1])
46  axes_3[1].set_title('Bzip2 QQplot')
47  sm.qqplot(h_data, dist=stats.norm, loc=mu[2], scale=var[2], line='r', ax=axes_3[2])
48  axes_3[2].set_title('Huffman QQplot')
49  plt.show()
50
51  mu = []
52  var = []
53  gq_1 = g_data.quantile(0.25)
54  gq_3 = g_data.quantile(0.75)
55  giqr = gq_3 - gq_1
56  g_data_norm = table.loc[table['Gzip'] <= gq_3 + 1.5 * giqr]
57  g_data_norm = table.loc[table['Gzip'] >= gq_1 - 1.5 * giqr]
58
59  bq_1 = b_data.quantile(0.25)
60  bq_3 = b_data.quantile(0.75)
61  biqr = bq_3 - bq_1
62  b_data_norm = table.loc[table['Bzip2'] <= bq_3 + 1.5 * biqr]
63  b_data_norm = table.loc[table['Bzip2'] >= bq_1 - 1.5 * biqr]
64
65  hq_1 = h_data.quantile(0.25)
66  hq_3 = h_data.quantile(0.75)
67  hiqr = hq_3 - hq_1
68  h_data_norm = table.loc[table['Huffman'] <= hq_3 + 1.5 * hiqr]
69  h_data_norm = table.loc[table['Huffman'] >= hq_1 - 1.5 * hiqr]
70
71  mu_2 = []
72  var_2 = []
73  i = 0
74  for data in [g_data_norm['Gzip'], b_data_norm['Bzip2'], h_data_norm['Huffman']]:
75      params = stats.norm.fit(data)
76      results = stats.kstest(data, 'norm', params)
77      mu_2.append(params[0])
78      var_2.append(params[1])
79      i += 1
80      print(results)
81
82  f_4, axes_4 = plt.subplots(1, 2)
83  sm.qqplot(b_data_norm['Bzip2'], dist=stats.norm, loc=mu_2[1], scale=var_2[1], line='r', ax=axes_4[0])
84  axes_4[0].set_title('Bzip2 QQplot')
85  sm.qqplot(h_data_norm['Huffman'], dist=stats.norm, loc=mu_2[2], scale=var_2[2], line='r', ax=axes_4[1])
86  axes_4[1].set_title('Huffman QQplot')
87  plt.show()
```

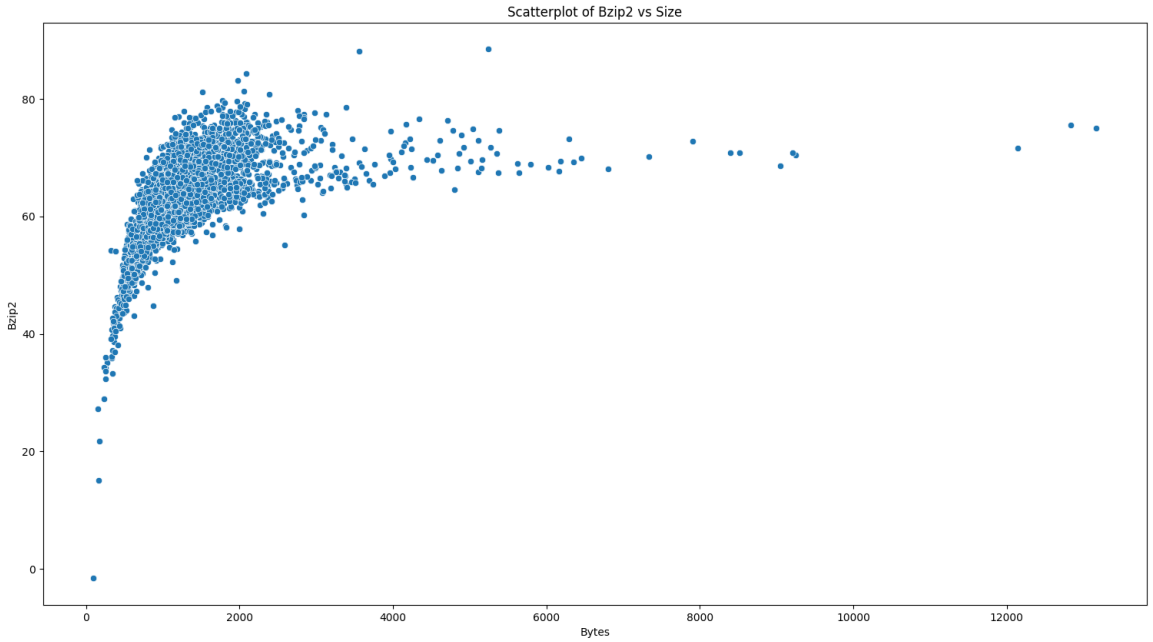
6.4.2 Η συμπίεση σε σχέση με το μέγεθος των αρχείων

Από την προηγούμενη ενότητα βλέπουμε ότι παρουσιάζεται μία πολύ διαφορετική εικόνα σε σχέση με το προηγούμενο πείραμα για την κατανομή των ποσοστών συμπίεσης σε σχέση με το μέγεθος των αρχείων. Υποψιαζόμαστε ότι ο λόγος που συμβαίνει αυτό είναι τα μικρά μεγέθη αρχείων που συναντάμε στην περίπτωση

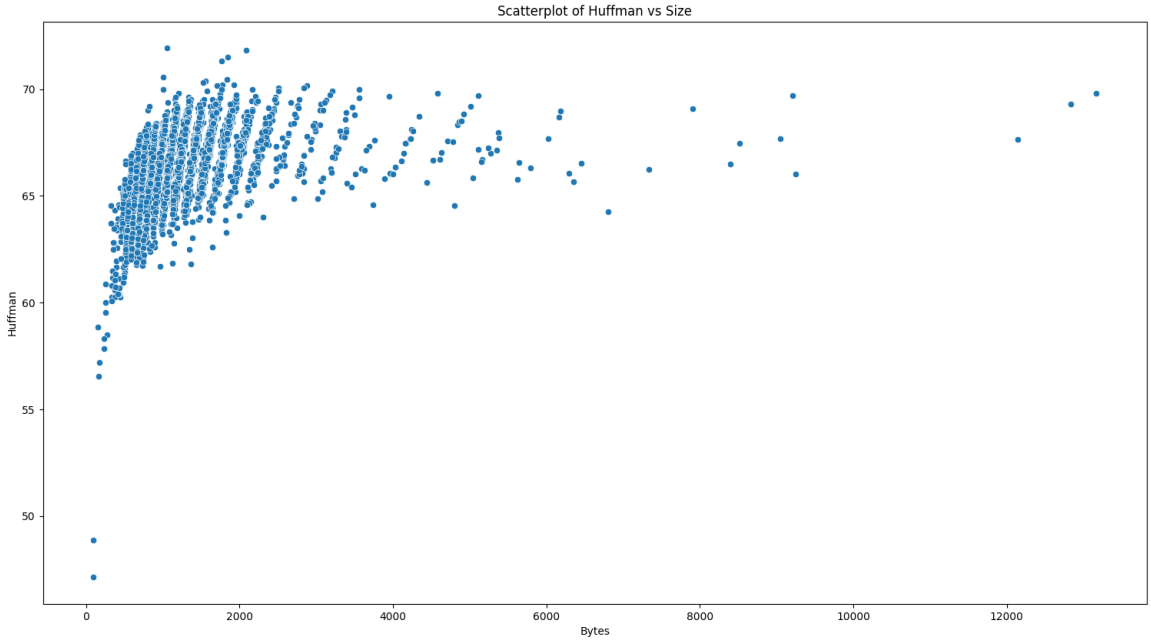
που μελετάμε. Ξεκινάμε την μελέτη μας δημιουργώντας κάποια γραφήματα σκέδασης για την ανεξάρτητη μεταβλητή μέγεθος και την εξαρτημένη ποσοστό συμπίεσης.



Σχήμα 6.30: Το scatterplot της μεταβλητής μέγεθος αρχείου σε σχέση με τον συμπιεστή Gzip



Σχήμα 6.31: Το scatterplot της μεταβλητής μέγεθος αρχείου σε σχέση με τον συμπιεστή Bzip2



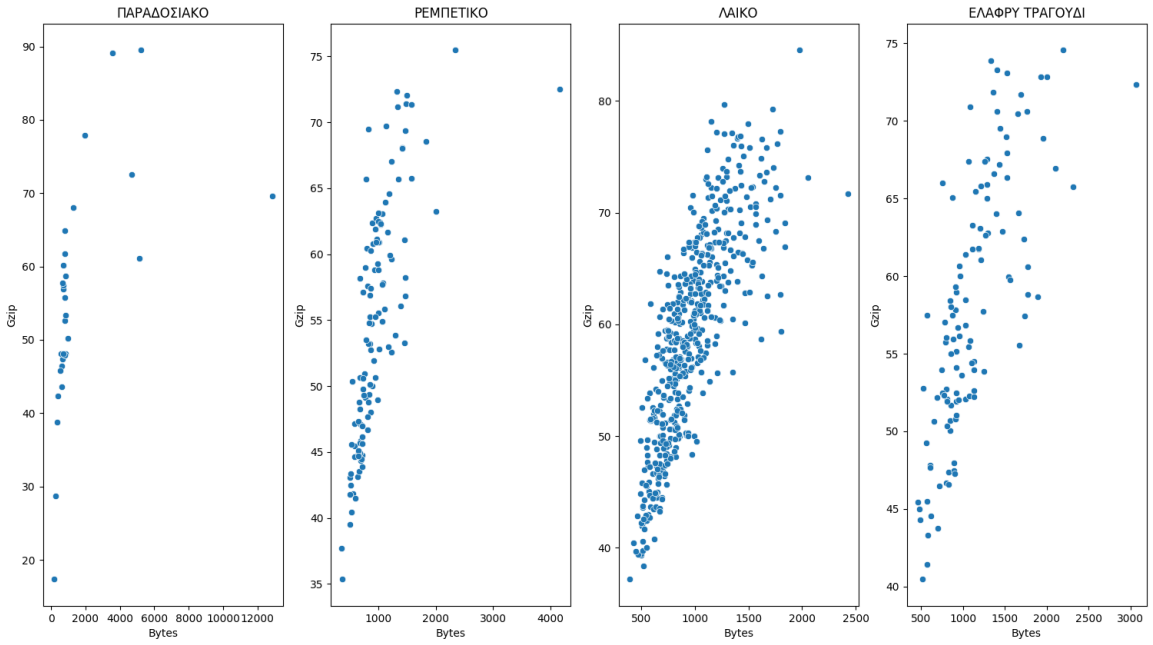
Σχήμα 6.32: Το scatterplot της μεταβλητής μέγεθος αρχείου σε σχέση με τον συμπιεστή Huffman

Από τα γραφήματα φαίνεται ότι υπάρχει ξεκάθαρη σχέση μεταξύ του μεγέθους των αρχείων και των ποσοστών συμπίεσης. Συγκεκριμένα βλέπουμε ότι υπάρχει μία λογαριθμική αύξηση του ποσοστού συμπίεσης η οποία τείνει να ισορροπήσει και να γίνει ευθεία γραμμή μετά τα 2000 bytes. Η έκρηξη στην διασπορά που βλέπουμε στα ποσοστά συμπίεσης υποθέτουμε ότι έχει να κάνει με το είδος, δηλαδή το περιεχόμενο, το οποίο συμπιέζεται. Ακόμη ενδιαφέρον έχει να παρατηρήσουμε την μείωση ακόμα και στην διασπορά στα ποσοστά συμπίεσης καθώς μεγαλώνει το μέγεθος. Μεταξύ του Gzip και του Bzip2 για την ίδια κλάση μεγέθους [1000-3000] bytes μεγαλύτερη διασπορά στα ποσοστά ξεκάθαρα εμφανίζει ο Gzip. Για να εξασφαλίσουμε και στατιστικά την συσχέτιση που υπάρχει ανάμεσα στις δύο μεταβλητές θα υπολογίσουμε τους μη παραμετρικούς συντελεστές Spearmans και Kendal Tau.

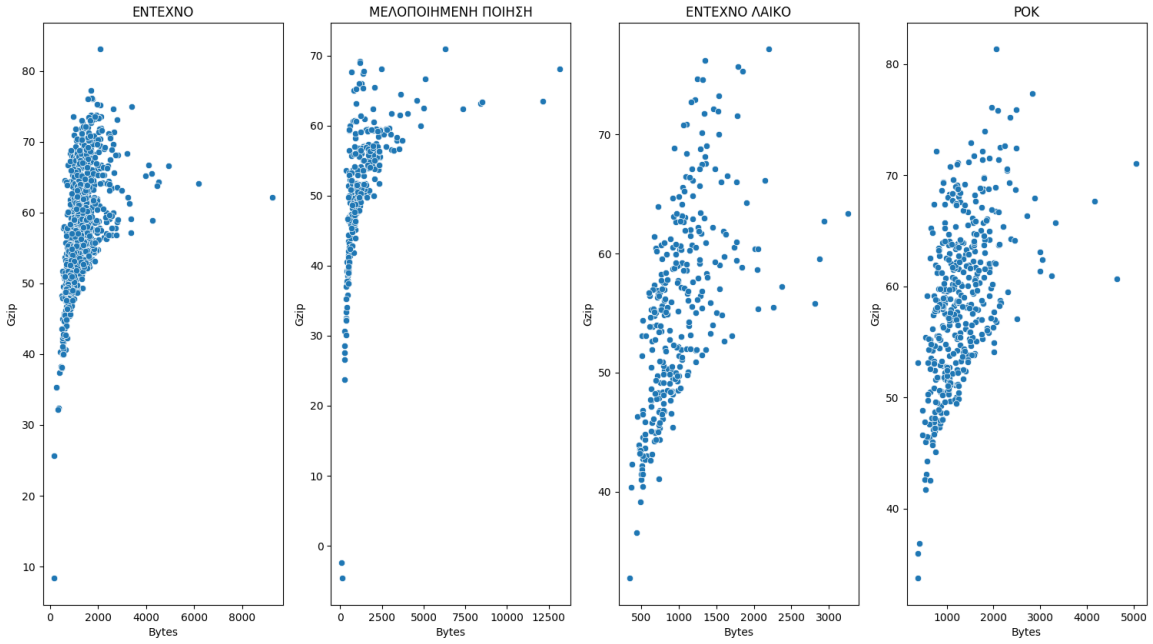
Συμπιεστής	Spearmans	Spearmans p-value	Kendall Tau	Kendall Tau p-value
Gzip	0.6716627490425969	0.0	0.4868100755619484	0.0
Bzip2	0.7981611522248032	0.0	0.6065994333704182,	0.0
Huffman	0.5697960026197212	0.0	0.4129423747341343	0.0

Οι μη παραμετρικοί συντελεστές συσχέτισης δείχνουν ακριβώς αυτό που περιγράψαμε από τα γραφήματα, δηλαδή την ύπαρξη μίας αύξουσας μονοτονικής σχέσης μεταξύ των συμπιεστών και του μεγέθους που συμπιέζεται. Αυτό που συμβαίνει στην προκειμένη περίπτωση είναι πως καθώς μεγαλώνει το μέγεθος μπαίνουν σε εφαρμογή τα ασυμπτωτικά θεωρήματα που παρουσιάστηκαν και αναλύθηκαν λεπτομερώς στο προηγούμενο μέρος, δηλαδή οι συμπιεστές αρχίζουν καθώς μεγαλώνει το μέγεθος να προσεγγίζουν την πηγή έως μια τάξη k που είναι η τάξη του συμπιεστή επί της ουσίας, οπότε είναι σε θέση να τη προβλέπουν με αποτέλεσμα να την συμπιέζουν καλύτερα. Άρα αυτό που προκύπτει από τα αποτελέσματα είναι πως όταν το μέγεθος γίνεται πολύ μικρό τότε δεν επαρκεί ώστε οι συμπιεστέ να καταφέρουν να εξάγουν τα σωστά στατιστικά στοιχεία της πηγής που τους επιτρέπει η τάξη τους.

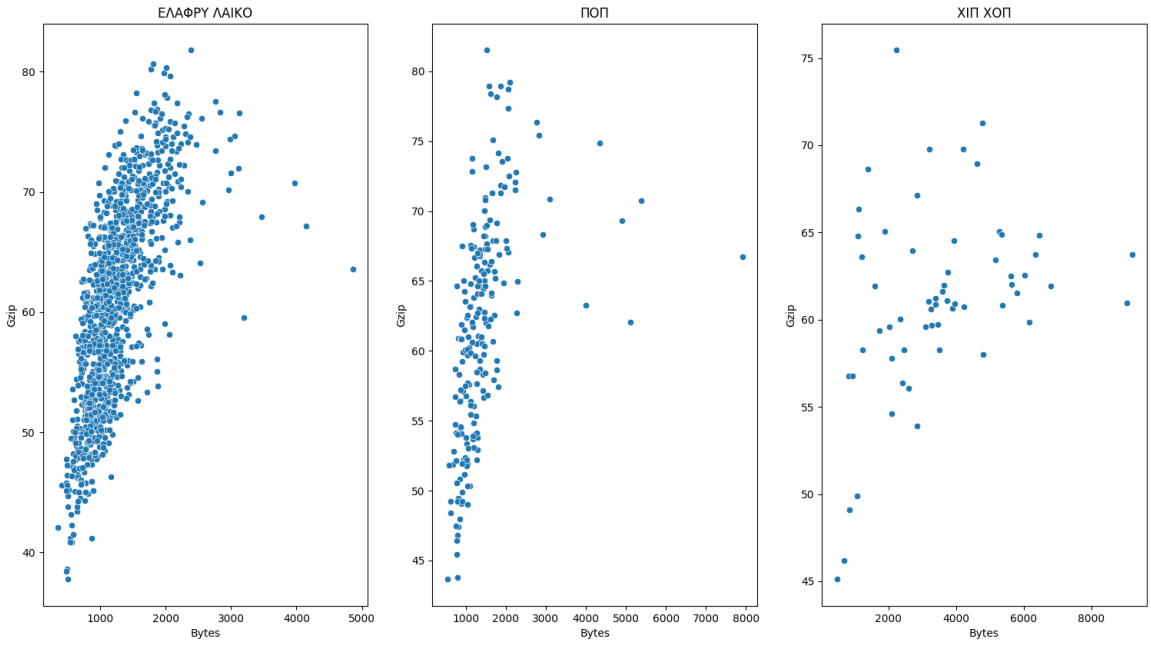
Επειδή από την προηγούμενο πείραμα βρήκαμε ότι το είδος του τραγουδιού επηρεάζει το ποσοστό συμπίεσης και σε αυτό το πείραμα βλέπουμε πως για το ίδιο μέγεθος υπάρχει διασπορά στα ποσοστά συμπίεσης δεν γίνεται να μην μπούμε στον πειρασμό να δημιουργήσουμε τα γραφήματα σκέδασης για κάθε είδος τραγουδιού.



Σχήμα 6.33

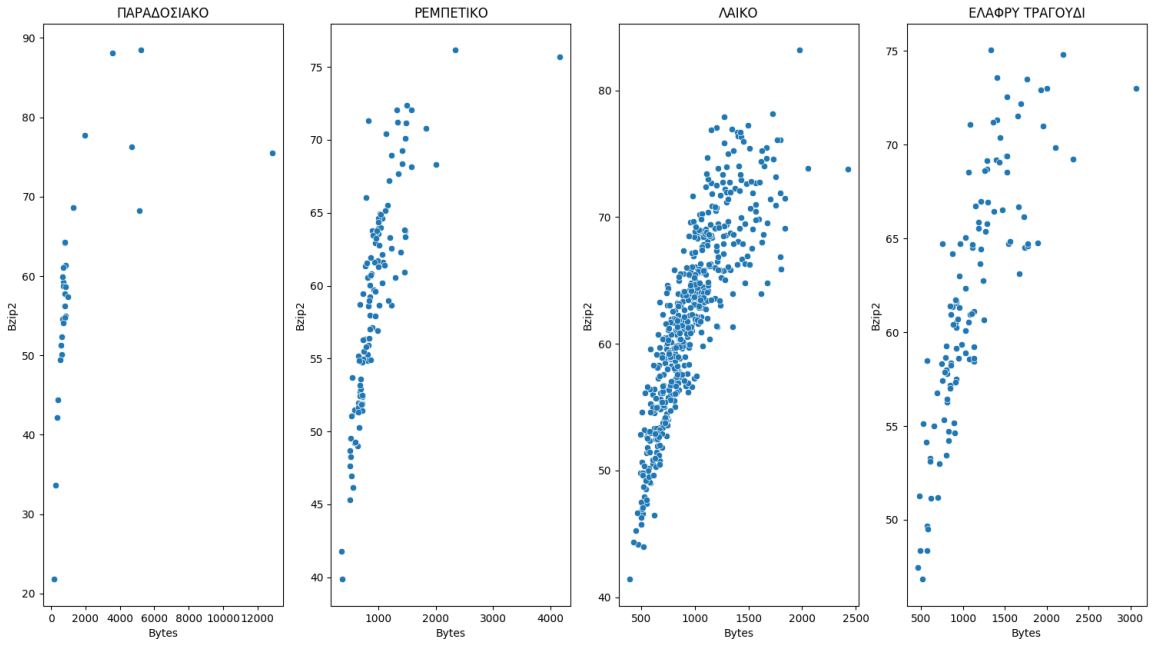


Σχήμα 6.34

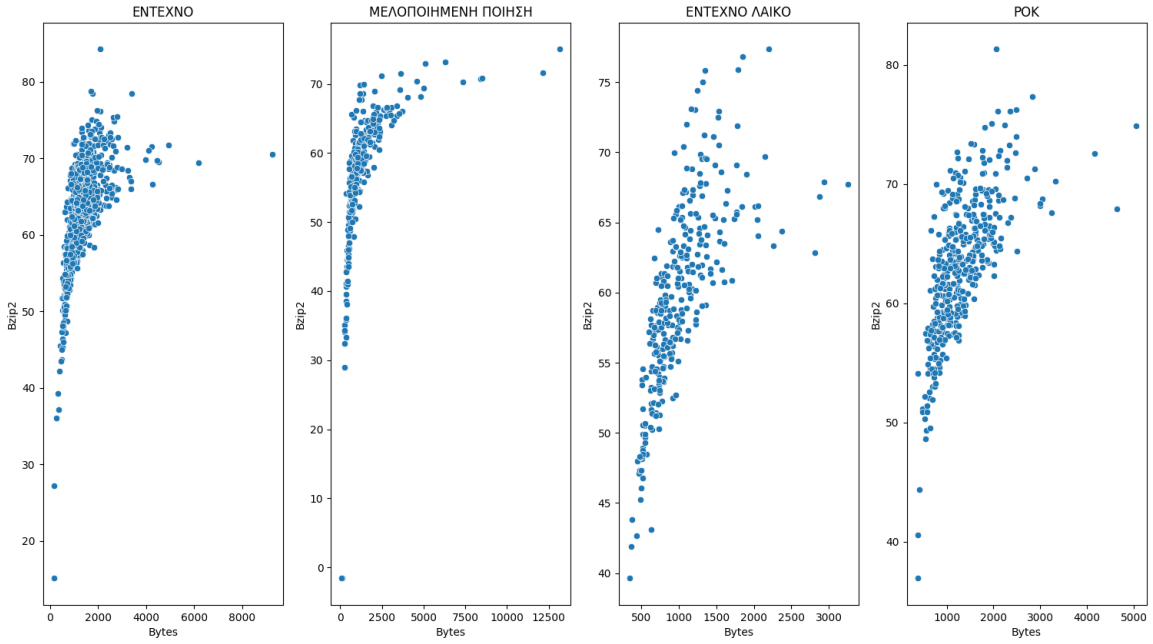


Σχήμα 6.35

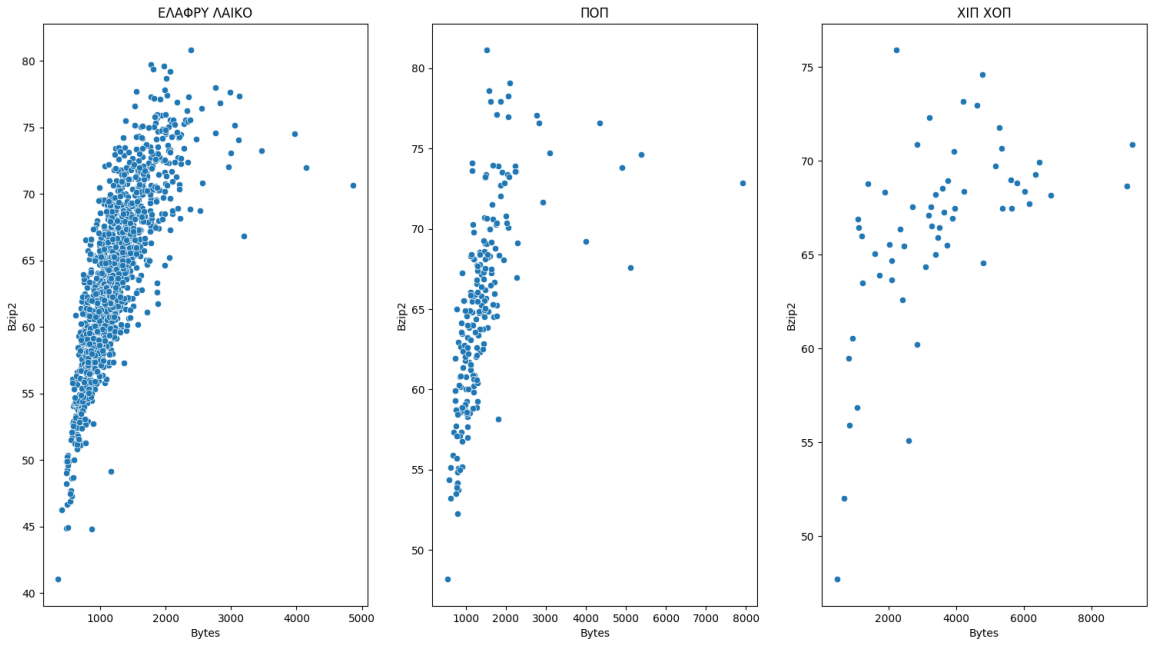
Από τα γραφήματα βλέπουμε ότι η σχέση του Gzip με το μέγεθος των αρχείων όταν παίρνουμε υπόψιν και τι είδος το οποίο συμπιέζεται είναι στις περισσότερες περιπτώσεις σχεδόν γραμμική αλλά πάντα μονότονα αύξουσα. Την χειρότερη συμπεριφορά βλέπουμε ότι την έχει στο ΧΙΠ ΧΟΠ. Μικρότερη διασπορά φαίνεται να παρουσιάζει στο ΠΟΠ, το ΠΑΡΑΔΟΣΙΑΚΟ και το ΡΕΜΠΕΤΙΚΟ τραγούδι, το οποίο είναι λογικό καθώς αν αναλογιστούμε τραγούδια του κάθε είδους από τα παραπάνω θα δούμε ότι περιέχουν επαναλαμβανόμενους στίχους σε κοντινές αποστάσεις μεταξύ τους.



Σχήμα 6.36

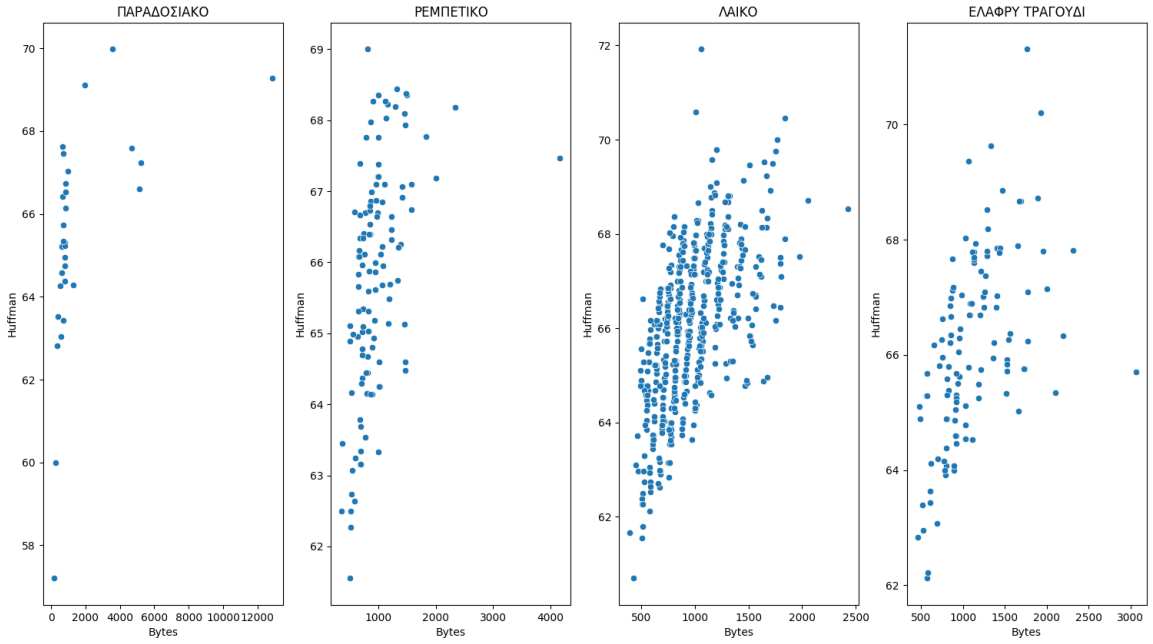


Σχήμα 6.37

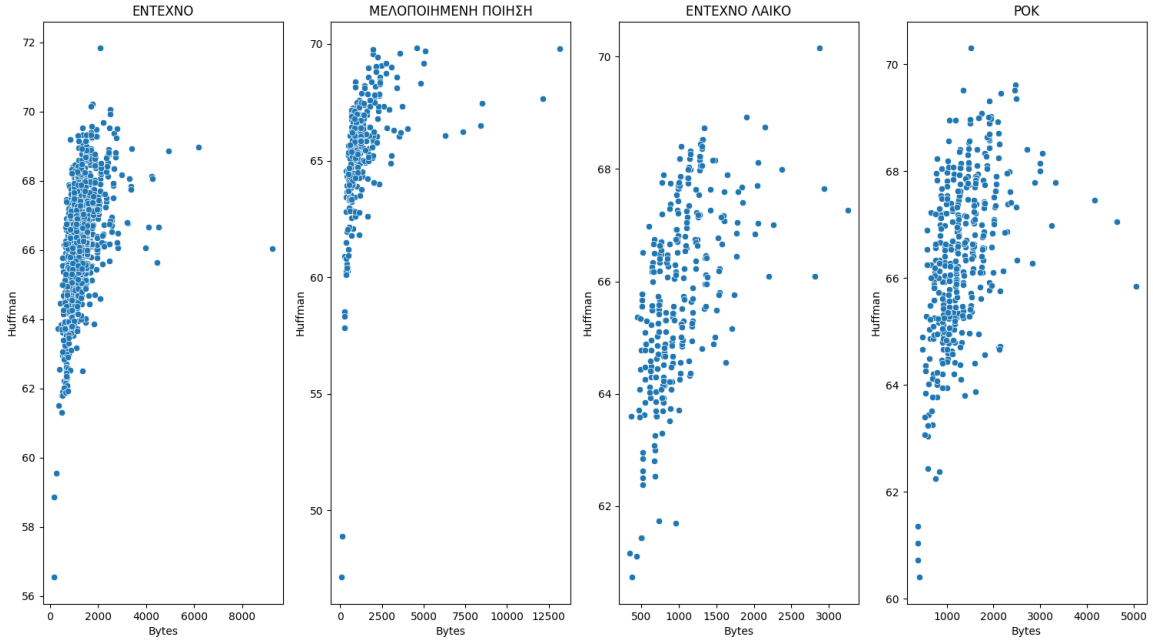


Σχήμα 6.38

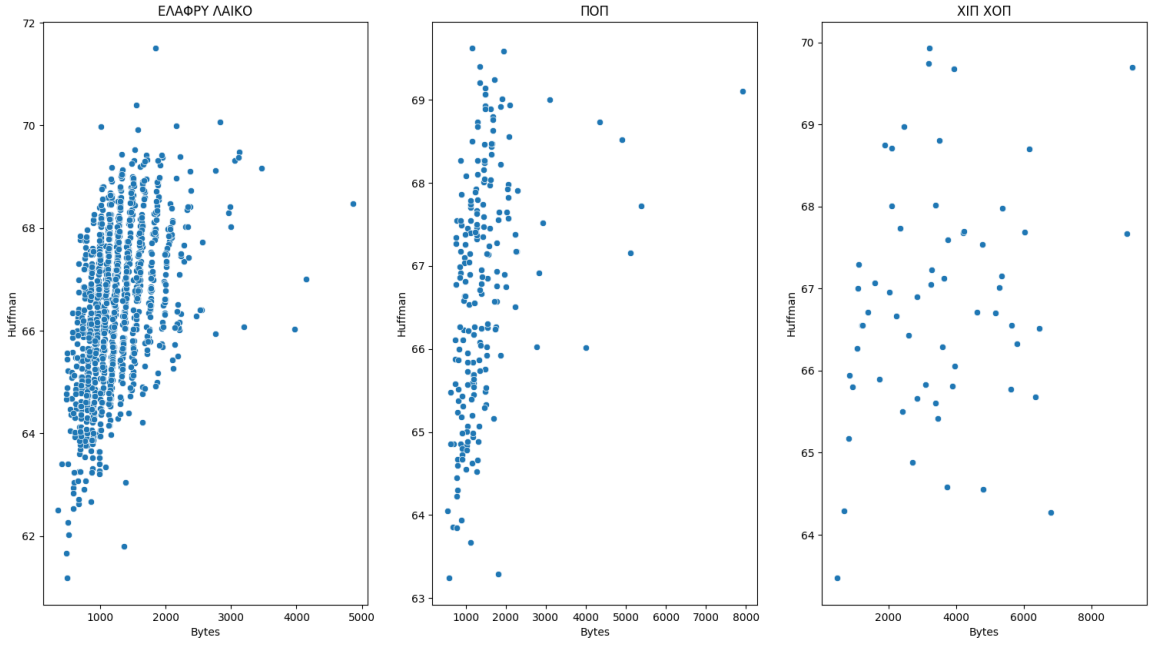
Στα γραφήματα για τον Bzip2 και τα είδη των τραγουδιών βλέπουμε πάλι την ίδια μονότονη σχέση που βρήκαμε στον Gzip. Και εδώ βλέπουμε μία αδυναμία προσαρμογής του συμπιεστή στο ΧΙΠ ΧΟΠ εν αντιθέσει με το ΠΟΠ, το ΡΕΜΠΕΤΙΚΟ κα το ΠΑΡΑΔΟΣΙΑΚΟ. Όσον αφορά στην διασπορά υπάρχει μία ελαφρά διαφορά αν συγκρίνουμε τα γραφήματα των δύο συμπιεστών για τα είδη ΧΙΠ ΧΟΠ, ΡΟΚ και ΜΕΛΟΠΟΙΜΕΝΗ ΠΟΙΗΣΗ όπου φαίνονται λίγο καλύτερα.



Σχήμα 6.39



Σχήμα 6.40



Σχήμα 6.41

Ο Huffman φαίνεται σε αυτό το πείραμα να επηρεάζεται από το είδος που συμπιέζει καθώς παρουσιάζει όμοιες εικόνες με τα προηγούμενα γραφήματα, όμως εξαιτίας της μικρής διασποράς του φαίνεται σαν δημιουργεί κλάσεις καθώς μεταβαίνει σε όλο και μεγαλύτερα μεγέθη. Παρατηρούμε ότι δεν παρουσιάζει την ίδια αυξητική διάθεση ως προς τη συμπίεση σε σχέση με τους Gzip καιBzip2, δηλαδή βελτιώνεται η συμπίεση καθώς αυξάνει το μέγεθος αλλά όχι στον βαθμό που επιτυγχάνεται από τους άλλους δύο. Αυτό φαίνεται εμφανώς

αν συγκρίνουμε τα γραφήματα της ΠΟΠ και του ΕΛΑΦΡΥ ΛΑΙΚΟ του Huffman σε σχέση με τα γραφήματα των Gzip καιBzip2.

Συμπερασματικά λοιπόν μπορούμε να πούμε πως και οι τρεις συμπιεστές εξαρτώνται από το μέγεθος των αρχείων. Εμφανίζουν διαφοροποιήσεις στα ποσοστά που επιτυγχάνουν ανάλογα το κείμενο που συμπιέζουν για μικρά μεγέθη ενώ τείνουν να επιτυγχάνουν ποσοστά συμπίεσης γύρω από το μέσο όταν το μέγεθος αρχίζει και μεγαλώνει. Την αύξηση του μεγέθους την εκμεταλλεύονται καλύτερα οι Gzip καιBzip2 σε σχέση με τον Huffman αυξάνοντας έτσι τα ποσοστά συμπίεσης του. Την μεγαλύτερη ευαισθησία σε αλλαγές μεγέθους και περιεχομένου φαίνεται να έχει ο Gzip ακολουθεί ο Bzip2 και έπεται τελευταίος ο Huffman. Το είδος που φαίνεται να δυσκόλεψε και τους τρεις περισσότερο είναι το ΧΙΠ ΧΟΠ, η ΜΕΛΟΠΟΙΗΜΕΝΗ ΠΟΙΗΣΗ και το ΕΝΤΕΧΝΟ ΛΑΙΚΟ ενώ φαίνεται να προσαρμόζονται εύκολα σε είδη όπως το ΠΟΠ και το ΕΛΑΦΡΥ ΛΑΙΚΟ. Τα ίδια είδη να πούμε ότι δυσκόλεψαν τους συμπιετές και στο προηγούμενο κώδικα. Ολοκληρώνοντας παραδίδουμε τον κώδικα βάση του οποίου κατασκευάστηκαν τα γραφήματα και οι έλεγχοι.

Experiment_2_Vs_Size.py

```
import pandas as pd
import matplotlib.pyplot as plt
import os
from scipy import stats
import seaborn as sns
import statsmodels.api as sm

table = pd.read_csv(os.getcwd() + '/Experiment_2.csv', index_col=0)
print(len(table.index))
indexes = table.loc[table['Στιχουργός'] == 'Αξιώτης Άγγελος'].index
table.drop(indexes, inplace=True)
print(len(table.index))
g_data = table['Gzip']
b_data = table['Bzip2']
h_data = table['Huffman']
size = table['Bytes']

sns.scatterplot(data=table, x='Bytes', y='Gzip')
plt.title('Scatterplot of Gzip vs Size')
plt.show()
sns.scatterplot(data=table, x='Bytes', y='Bzip2')
plt.title('Scatterplot of Bzip2 vs Size')
plt.show()
sns.scatterplot(data=table, x='Bytes', y='Huffman')
plt.title('Scatterplot of Huffman vs Size')
plt.show()

print(stats.spearmanr(size, g_data))
print(stats.spearmanr(size, b_data))
print(stats.spearmanr(size, h_data))

print(stats.kendalltau(size, g_data))
print(stats.kendalltau(size, b_data))
print(stats.kendalltau(size, h_data))

table_PARADOSIAKO=table.loc[table['Είδος']=='ΠΑΡΑΔΟΣΙΑΚΟ']
table_REMPETIKO=table.loc[table['Είδος']=='ΡΕΜΠΕΤΙΚΟ']
table_LAIKO=table.loc[table['Είδος']=='ΛΑΙΚΟ']
table_ELAFRI=table.loc[table['Είδος']=='ΕΛΑΦΡΥ ΤΡΑΓΟΥΔΙ']
table_ENTEXNO=table.loc[table['Είδος']=='ΕΝΤΕΧΝΟ']
table_MEL_P=table.loc[table['Είδος']=='ΜΕΛΟΠΟΙΗΜΕΝΗ ΠΟΙΗΣΗ']
table_ENTEXNO_LAIKO=table.loc[table['Είδος']=='ΕΝΤΕΧΝΟ ΛΑΙΚΟ']
table_ROK=table.loc[table['Είδος']=='ΡΟΚ']
table_EL_LAIKO=table.loc[table['Είδος']=='ΕΛΑΦΡΥ ΛΑΙΚΟ']
table_POP=table.loc[table['Είδος']=='ΠΟΠ']
table_HIP_HOP=table.loc[table['Είδος']=='ΧΙΠ ΧΟΠ']

fig, axes = plt.subplots(1, 4)
sns.scatterplot(data=table_PARADOSIAKO, x='Bytes', y='Gzip', ax=axes[0])
axes[0].set_title('ΠΑΡΑΔΟΣΙΑΚΟ')
sns.scatterplot(data=table_REMPETIKO, x='Bytes', y='Gzip', ax=axes[1])
axes[1].set_title('ΡΕΜΠΕΤΙΚΟ')
sns.scatterplot(data=table_LAIKO, x='Bytes', y='Gzip', ax=axes[2])
axes[2].set_title('ΛΑΙΚΟ')
```

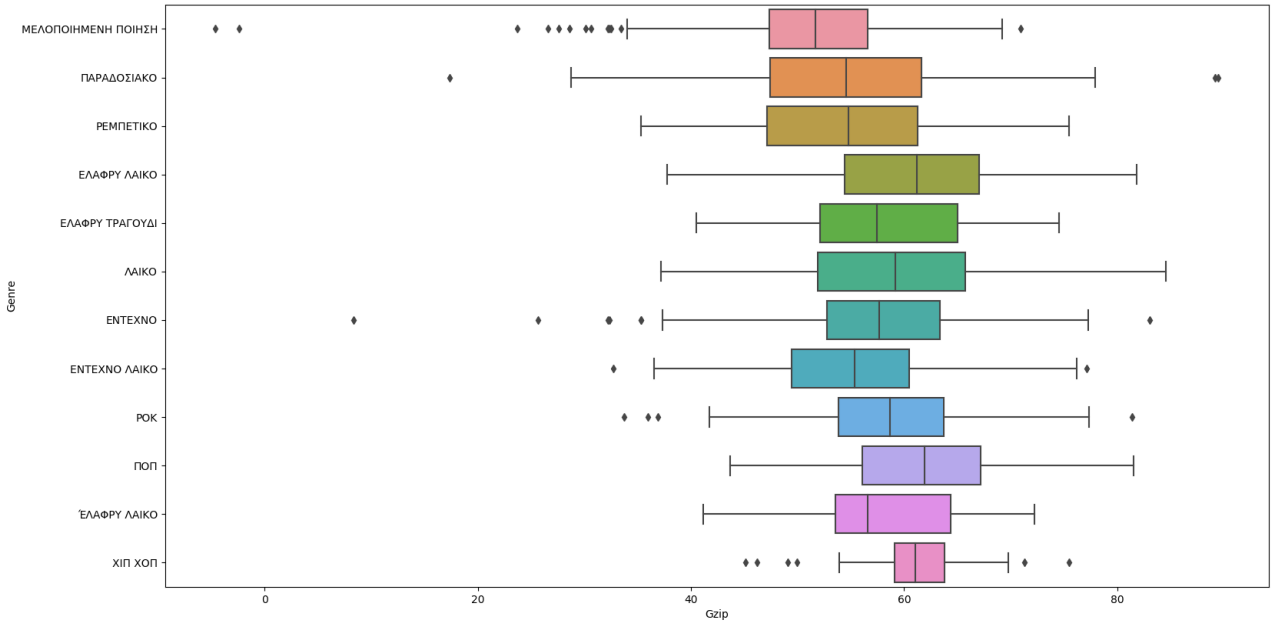
```
sns.scatterplot(data=table_ELAFRI, x='Bytes', y='Gzip', ax=axes[3])
axes[3].set_title('ΕΛΑΦΡΥ ΤΡΑΓΟΥΔΙ')
plt.show()
fig_2, axes_2 = plt.subplots(1, 4)
sns.scatterplot(data=table_ENTEXNO, x='Bytes', y='Gzip', ax=axes_2[0])
axes_2[0].set_title('ENTEXNO')
sns.scatterplot(data=table_MEL_P, x='Bytes', y='Gzip', ax=axes_2[1])
axes_2[1].set_title('ΜΕΛΟΠΟΙΗΜΕΝΗ ΠΟΙΗΣΗ')
sns.scatterplot(data=table_ENTEXNO_LAIKO, x='Bytes', y='Gzip', ax=axes_2[2])
axes_2[2].set_title('ENTEXNO ΛΑΙΚΟ')
sns.scatterplot(data=table_ROK, x='Bytes', y='Gzip', ax=axes_2[3])
axes_2[3].set_title('POK')
plt.show()

fig_3, axes_3 = plt.subplots(1, 3)
sns.scatterplot(data=table_EL_LAIKO, x='Bytes', y='Gzip', ax=axes_3[0])
axes_3[0].set_title('ΕΛΑΦΡΥ ΛΑΙΚΟ')
sns.scatterplot(data=table_POP, x='Bytes', y='Gzip', ax=axes_3[1])
axes_3[1].set_title('ΠΟΠ')
sns.scatterplot(data=table_HIP_HOP, x='Bytes', y='Gzip', ax=axes_3[2])
axes_3[2].set_title('ΧΙΠ ΧΟΠ')
plt.show()

fig, axes = plt.subplots(1, 4)
sns.scatterplot(data=table_PARADOSIAKO, x='Bytes', y='Huffman', ax=axes[0])
axes[0].set_title('ΠΑΡΑΔΟΣΙΑΚΟ')
sns.scatterplot(data=table_REMPETIKO, x='Bytes', y='Huffman', ax=axes[1])
axes[1].set_title('ΡΕΜΠΕΤΙΚΟ')
sns.scatterplot(data=table_LAIKO, x='Bytes', y='Huffman', ax=axes[2])
axes[2].set_title('ΛΑΙΚΟ')
sns.scatterplot(data=table_ELAFRI, x='Bytes', y='Huffman', ax=axes[3])
axes[3].set_title('ΕΛΑΦΡΥ ΤΡΑΓΟΥΔΙ')
plt.show()
fig_2, axes_2 = plt.subplots(1, 4)
sns.scatterplot(data=table_ENTEXNO, x='Bytes', y='Huffman', ax=axes_2[0])
axes_2[0].set_title('ENTEXNO')
sns.scatterplot(data=table_MEL_P, x='Bytes', y='Huffman', ax=axes_2[1])
axes_2[1].set_title('ΜΕΛΟΠΟΙΗΜΕΝΗ ΠΟΙΗΣΗ')
sns.scatterplot(data=table_ENTEXNO_LAIKO, x='Bytes', y='Huffman', ax=axes_2[2])
axes_2[2].set_title('ENTEXNO ΛΑΙΚΟ')
sns.scatterplot(data=table_ROK, x='Bytes', y='Huffman', ax=axes_2[3])
axes_2[3].set_title('POK')
plt.show()

fig_3, axes_3 = plt.subplots(1, 3)
sns.scatterplot(data=table_EL_LAIKO, x='Bytes', y='Huffman', ax=axes_3[0])
axes_3[0].set_title('ΕΛΑΦΡΥ ΛΑΙΚΟ')
sns.scatterplot(data=table_POP, x='Bytes', y='Huffman', ax=axes_3[1])
axes_3[1].set_title('ΠΟΠ')
sns.scatterplot(data=table_HIP_HOP, x='Bytes', y='Huffman', ax=axes_3[2])
axes_3[2].set_title('ΧΙΠ ΧΟΠ')
plt.show()
```

6.4.3 Η συμπίεση σε σχέση με το είδος τραγουδιών

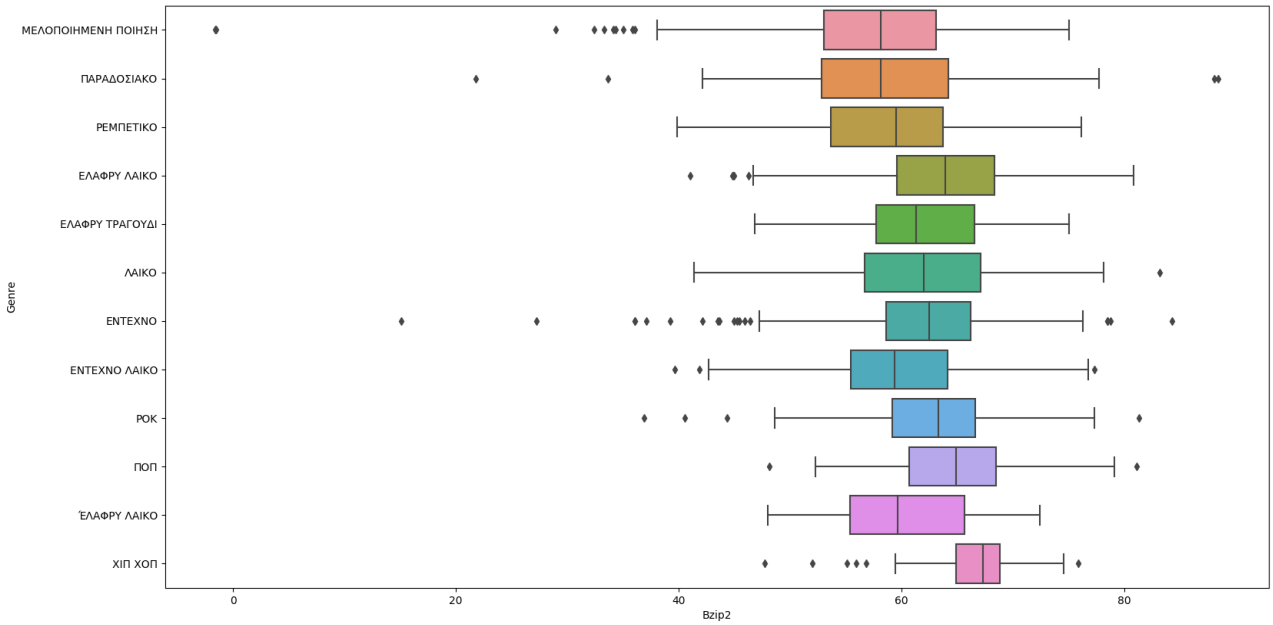
Από την στιγμή που είδαμε στο προηγούμενο πείραμα και την προηγούμενη ενότητα υπήρχε συσχέτιση μεταξύ τους είδους και της συμπίεσης ήρθε η ώρα να αναλύσουμε και σε αυτό το πείραμα την σχέση αυτή. Για μία οπτική σύγκριση θα παραθέσουμε τα boxplots των ποσοστών συμπίεσης για κάθε είδος προκειμένου να συγκρίνουμε την θέση των διάμεσων των διαφόρων ειδών ενώ για στατιστικό έλεγχο θα χρησιμοποιήσουμε το τεστ Kruskal-Wallis

Κοιτώντας τα διαγράμματα και τα αποτελέσματα των ελέγχων Kruskall Wallis βλέπουμε ότι υπάρχει διαφορά στο ποσοστό συμπίεσης ανάλογα το είδος. Αυτό όμως που έχει ενδιαφέρον είναι ότι για μικρά μεγέθη ακόμα και ο Huffman φαίνεται να δείχνει μία διαφοροποίηση στο είδος. Στην ουσία αυτό που συμβαίνει είναι πως τα μικρά σε μέγεθος αρχεία δεν αφήνουν τους συμπιεστές να κατανοήσουν την πηγή ώστε να την συμπιέσουν αποδοτικά και μάλιστα αυτό γίνεται σε τέτοιο βαθμό που μέχρι και στον Huffman σε ένα συμπιεστή μηδενική τάξης δεν του αρκούν τα δεδομένα προκειμένου να εξάγει το σωστό πιθανοκρατικό μοντέλο για το αρχείο που πρόκειται να συμπιέσει. Εν αντιθέσει με το προηγούμενο πείραμα που τα δεδομένα του αρκούσαν

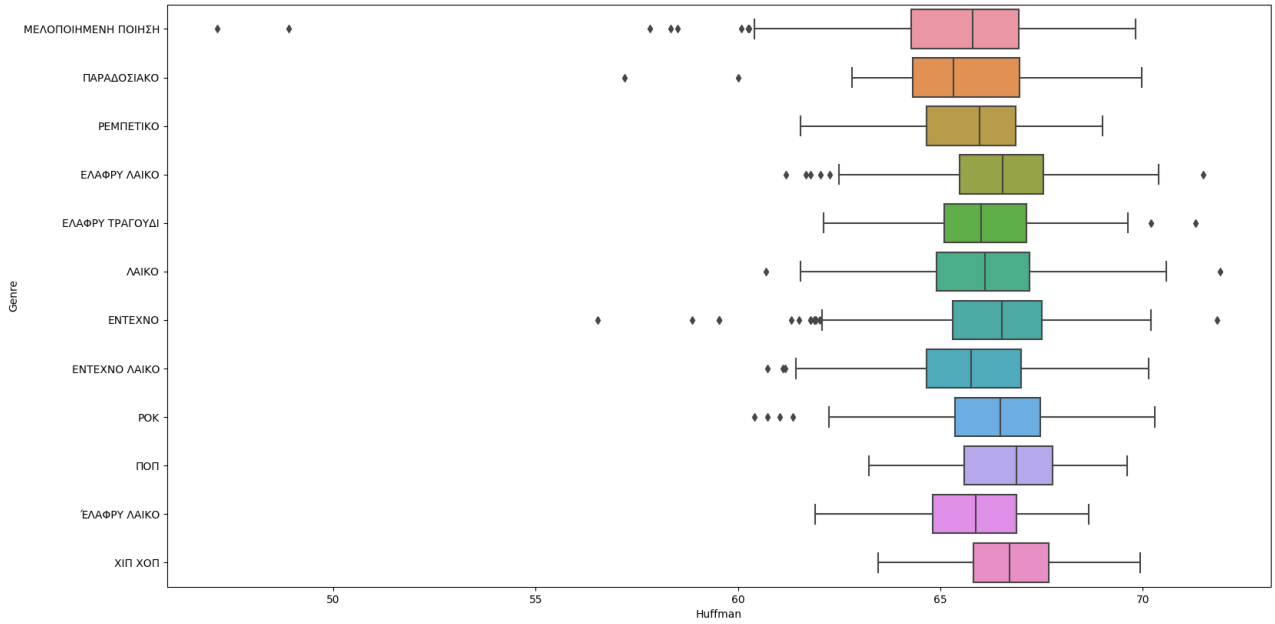
σε τέτοιο βαθμό ώστε να μη τον επηρεάζει καθόλου το περιεχόμενο.



Σχήμα 6.42



Σχήμα 6.43



Σχήμα 6.44

Συμπιεστής	Kruskal-Wallis	p-value
Gzip	356.60590250101114	1.578455435872303ε-70
Bzip2	318.91727339185945	1.5464546293902522ε-62
Huffman	134.3077074160327	6.162743528291311ε-24

Ο κώδικας με τον οποίο υλοποιήθηκαν όλα τα παραπάνω φαίνεται στις παρακάτω εικόνες:

Experiment_2_Vs_Genre.py

```
1
2  import pandas as pd
3  import matplotlib.pyplot as plt
4  import os
5  from scipy import stats
6  import seaborn as sns
7  import statsmodels.api as sm
8
9  table = pd.read_csv(os.getcwd() + '/Experiment_2.csv', index_col=0)
10 print(len(table.index))
11 indexes = table.loc[table['Στιχουργός'] == 'Αξιώτης Άγγελος'].index
12 table.drop(indexes, inplace=True)
13 print(len(table.index))
14 g_data = table['Gzip']
15 b_data = table['Bzip2']
16 h_data = table['Huffman']
17 size = table['Bytes']
18
19 table_PARADOSIAKO=table.loc[table['Είδος']=='ΠΑΡΑΔΟΣΙΑΚΟ']
20 table_REMPETIKO=table.loc[table['Είδος']=='ΡΕΜΠΕΤΙΚΟ']
21 table_LAIKO=table.loc[table['Είδος']=='ΛΑΙΚΟ']
22 table_ELAFRI=table.loc[table['Είδος']=='ΕΛΑΦΡΥ ΤΡΑΓΟΥΔΙ']
23 table_ENTEXNO=table.loc[table['Είδος']=='ΕΝΤΕΧΝΟ']
24 table_MEL_P=table.loc[table['Είδος']=='ΜΕΛΟΠΟΙΗΜΕΝΗ ΠΟΙΗΣΗ']
25 table_ENTEXNO_LAIKO=table.loc[table['Είδος']=='ΕΝΤΕΧΝΟ ΛΑΙΚΟ']
26 table_ROK=table.loc[table['Είδος']=='ΡΟΚ']
27 table_EL_LAIKO=table.loc[table['Είδος']=='ΕΛΑΦΡΥ ΛΑΙΚΟ']
28 table_POP=table.loc[table['Είδος']=='ΠΟΠ']
29 table_HIP_HOP=table.loc[table['Είδος']=='ΧΙΠ ΧΟΠ']
30
```

```
31 sns.boxplot(data=table, y='Genre', x='Gzip')
32 plt.show()
33
34 sns.boxplot(data=table, y='Genre', x='Bzip2')
35 plt.show()
36
37 sns.boxplot(data=table, y='Genre', x='Huffman')
38 plt.show()
39
40 g = stats.kruskal(table_PARADOSIAKO['Gzip'], table_REMPETIKO['Gzip'], table_LAIKO['Gzip'], table_ELAFRI['Gzip'],
41                   table_ENTEXNO['Gzip'], table_MEL_P['Gzip'], table_ENTEXNO_LAIKO['Gzip'], table_ROK['Gzip'],
42                   table_EL_LAIKO['Gzip'], table_POP['Gzip'], table_HIP_HOP['Gzip'])
43 print(g)
44
45 b = stats.kruskal(table_PARADOSIAKO['Bzip2'], table_REMPETIKO['Bzip2'], table_LAIKO['Bzip2'], table_ELAFRI['Bzip2'],
46                   table_ENTEXNO['Bzip2'], table_MEL_P['Bzip2'], table_ENTEXNO_LAIKO['Bzip2'], table_ROK['Bzip2'],
47                   table_EL_LAIKO['Bzip2'], table_POP['Bzip2'], table_HIP_HOP['Bzip2'])
48 print(b)
49 h = stats.kruskal(table_PARADOSIAKO['Huffman'], table_REMPETIKO['Huffman'], table_LAIKO['Huffman'],
50                   table_ELAFRI['Huffman'],
51                   table_ENTEXNO['Huffman'], table_MEL_P['Huffman'], table_ENTEXNO_LAIKO['Huffman'],
52                   table_ROK['Huffman'], table_EL_LAIKO['Huffman'], table_POP['Huffman'], table_HIP_HOP['Huffman'])
53 print(h)
```

Στο σημείο αυτό το πείραμα ολοκληρώνεται τα βασικά σημεία που πρέπει να κρατήσουμε είναι:

1. Από τους τρεις συμπιεστές πιο επιρρεπής σε μεταβολές ως προς το περιεχόμενο που συμπιέζει για μεγάλα μεγέθη αρχείων φάνηκε να είναι ο Gzip, ακολούθησε ο Bzip ο οποίος επηρεαζόταν από το περιεχόμενο αλλά εμφάνιζε μικρότερη διασπορά. Αυτός που παρέμεινε αμετάβλητος ως προς την απόδοση του ήταν ο Huffman του οποίου η μικρή τάξη του (συμπιεστής μηδενική τάξης) δεν του επιτρέπει να καταλάβει καν την έννοια περιεχόμενο. Άρα αυτό που καταλάβαμε είναι πώς όσο μεγαλύτερη τάξη έχει ένα συμπιεστεί τόσο περισσότερο θα επηρεάζεται από μειώσεις μεγέθους των αρχείων που συμπιέζει.

2. Από ένα μέγεθος και ύστερα οι συμπιεστές δείχνουν να μην επηρεάζονται από το μέγεθος του αρχείου καθώς μπαίνουν σε εφαρμογή τα ασυμπτωτικά θεωρήματα και ο συμπιεστής συγκλίνει στον ρυθμό εντροπίας της πηγή.

3. Όταν η συμπίεση γίνεται για αρχεία μικρού μεγέθους είδαμε ότι και οι τρεις συμπιεστές είναι επιρρεπής στις μεταβολές περιεχόμενου και μεγέθους. Εξηγήσαμε ότι αυτό συμβαίνει γιατί το μικρό δείγμα δεν επιτρέπει ούτε στους συμπιεστές της πιο μικρής τάξης να προσαρμοστούν στην πηγή. Παρόλα αυτά στα μικρά αρχεία φάνηκε ότι ο συμπιεστής με την μικρότερη διασπορά ήταν ο Huffman του οποίου το διάγραμμα σκέδασης έτεινε σε μία ισορροπία σε σχέση με την αύξηση του μεγέθους των αρχείων πιο γρήγορα από τους άλλους.

4. Ολοκληρώνοντας παρατηρήσαμε ότι οι συμπιεστές είναι ένα πολύ καλό εργαλείο προκειμένου να παρατηρήσουμε ποιοτικές διαφορές ανάμεσα σε γραπτά κείμενα διαφορετικού περιεχομένου. Ο καλύτερος για αυτή την δουλειά αποδείχτηκε ο Gzip που όπως προείπαμε έδειξε μια ευμεταβλητότητα στις αλλαγές περιεχομένου. Με βάση τα αποτελέσματα συμπίεση είδαμε πως το είδος μουσικής που συμπιέστηκε λιγότερο ήταν η ΜΕΛΟΠΟΙΗΜΕΝΗ ΠΟΙΗΣΗ, η ΧΙΠ ΧΟΠ και το ΕΝΤΕΧΝΟ ΛΑΙΚΟ ενώ στην αντίπερα όχθη με μεγάλα ποσοστά συμπίεσης βρίσκετια το ΛΑΙΚΟ και το ΕΛΑΦΡΥ ΛΑΙΚΟ ενώ τα υπόλοιπα είδη κυμαίνονται κάπου ενδιάμεσα.

Βιβλιογραφία

[o15] Μιχαήλ Λουλάκης. "Στοχαστικές Διαδικασίες". In: (2015), pp. 1–33, 72–12.

[AC88] Paul H Algoet and Thomas M Cover. "A sandwich proof of the Shannon-McMillan-Breiman theorem". In: *The annals of probability* (1988), pp. 899–909.

[Ban63] G Bandyopadhyay. "A simple proof of the decipherability criterion of Sardinas and Patterson". In: *Information and Control* 6.4 (1963), pp. 331–336.

[Bia] William Bialek. *Some background on information theory*. URL: `https://www.princeton.edu/~wbialek/rome/info_background.pdf`.

[Boe] Dr. C. George Boeree. *Speech and brain*. URL: `https://webspace.ship.edu/cgboer/speechbrain.html`.

[BW94] Michael Burrows and David J Wheeler. "A block-sorting lossless data compression algorithm". In: (1994).

[Cal] University of California San Francisco. *Speech and Language*. URL: `https://memory.ucsf.edu/symptoms/speech-language`.

[CT12] Thomas M Cover and Joy A Thomas. *Elements of information theory*. John Wiley & Sons, 2012, pp. 1–157, 183–243, 427–463, 644–650.

[Fan61] Robert M Fano. "Transmission of information: A statistical theory of communications". In: *American Journal of Physics* 29.11 (1961), pp. 793–794.

[Gal78] Robert Gallager. "Variations on a theme by Huffman". In: *IEEE Transactions on Information Theory* 24.6 (1978), pp. 668–674.

[Har28] Ralph VL Hartley. "Transmission of information 1". In: *Bell System technical journal* 7.3 (1928), pp. 535–563.

[Hor77] Yasuichi Horibe. "An improved bound for weight-balanced tree". In: *Information and Control* 34.2 (1977), pp. 148–151.

[Huf52] D. A. Huffman. "A Method for the Construction of Minimum-Redundancy Codes". In: *Proceedings of the IRE* 40.9 (1952), pp. 1098–1101.

[KLV07] Haim Kaplan, Shir Landau, and Elad Verbin. "A simpler analysis of Burrows–Wheeler-based compression". In: *Theoretical Computer Science* 387.3 (2007), pp. 220–235.

[Ko03] Χ Κουκουβίνος and Α Παπαϊωάννου. *Θεωρία Πληροφοριών και Κωδίκων*. 2003.

[K99] Γ Κοκολάκης and Ι Σπηλιώτης. "*Εισαγωγή στη θεωρία πιθανοτήτων και στατιστική*". In: *Εκδόσεις Συμεών, Αθήνα* (1999), pp. 49–89, 195–217.

[Lan] Ben Langmead. *Burrows-Wheeler Transform & FM Index*. URL: `https://www.cs.jhu.edu/~langmea/resources/lecture_notes/10_bwt_and_fm_index_v2.pdf`.

[Mak] Myrsini Makropoulou. *Introduction to Medical Physics - Sound and Speech*. URL: `http://www.physics.ntua.gr/~mmakro/index_files/Kef10_Hxos_Omilia_red.pdf`.

[Man01] Giovanni Manzini. "An analysis of the Burrows–Wheeler transform". In: *Journal of the ACM (JACM)* 48.3 (2001), pp. 407–430.

[McG15] Clare D McGillem. "Telegraph". In: *Encyclopædia Britannica Ultimate Reference Suite.–Chicago: Encyclopædia Britannica* (2015).

[McM56] Brockway McMillan. "Two inequalities implied by unique decipherability". In: *IRE Transactions on Information Theory* 2.4 (1956), pp. 115–116.

[Nyq24] Harry Nyquist. "Certain factors affecting telegraph speed". In: *Transactions of the American Institute of Electrical Engineers* 43 (1924), pp. 412–422.

[Pap03] Alexandros X Papaioanou. *Discrete Mathematics*. National Technical University of Athens, 2003, pp. 1–46.

[Piv] Inna Pivkina. *Discovery of Huffman Codes*. URL: `https://www.maa.org/press/periodicals/convergence/discovery-of-huffman-codes`.

[Ris73] Jorma Rissanen. "Bounds for weight balanced trees". In: *IBM Journal of Research and Development* 17.2 (1973), pp. 101–105.

[SM10] David Salomon and Giovanni Motta. *Handbook of data compression*. Springer Science & Business Media, 2010, pp. 45–48, 211–233, 264–275.

[Say12] Khalid Sayood. *Introduction to data compression*. Newnes, 2012, pp. 30–33, 46–51, 81–96, 141–143, 152–157.

[Sha48] Claude E Shannon. "A mathematical theory of communication". In: *Bell system technical journal* 27.3 (1948), pp. 379–423.

[SW49] Claude Elwood Shannon and Warren Weaver. *The Mathematical Theory of Communication, by CE Shannon (and Recent Contributions to the Mathematical Theory of Communication), W. Weaver*. University of illinois Press, 1949.

[Tam] Christopher K W Tam. *Finite Difference Equations*. URL: `https://assets.cambridge.org/97805218/06787/excerpt/9780521806787_excerpt.pdf`.

[Wel84] T. Welch. "A Technique for High-Performance Data Compression". In: *Computer* 17 (1984), pp. 8–19.

[Yeu08] Raymond W Yeung. *Information theory and network coding*. Springer Science & Business Media, 2008, pp. 99–111, 133–179.

[Yeu12] Raymond W. Yeung. *Chapter 7 DIscrete Memoryless Channels*. Feb. 2012.

[ZL77] Jacob Ziv and Abraham Lempel. "A universal algorithm for sequential data compression". In: *IEEE Transactions on information theory* 23.3 (1977), pp. 337–343.

[ZL78] Jacob Ziv and Abraham Lempel. "Compression of individual sequences via variable-rate coding". In: *IEEE transactions on Information Theory* 24.5 (1978), pp. 530–536.