Θεωρία Υπολογισμού και Εφαρμογή στην Αλγοριθμική Θεωρία Πληροφοριών

Μπουραντάνης Κωνσταντίνος
Διπλωματική Εργασία

1 Νοεμβρίου 2018

•

Η παρούσα διπλωματική εργασία γράφτηκε στα πλαίσια του προπτυχιακού προγράμματος ΣΕΜΦΕ-ΕΜΠ, υπό την επίβλεψη του καθηγητή Α. Αρβανιτάκη, τον οποίο θα ήθελα να ευχαριστήσω για την πολύτιμη καθοδήγηση και την άριστη συνεργασία.

Κ.Μπουραντάνης
Σεπτέμβρης, 2018

Περιεχόμενα

Abstract

> Algorithmic information theory is the result of putting Shannon's information theory and Turing's computability theory into a cocktail shaker and shaking vigorously..
>
> *Chaitin*

We often witness the existence of mathematical notions, some fundamental, that even though they are intuitively quite simple, providing a formalisation of them is not only disproportionately difficult, but also is usually considered a breakthrough when achieved.

Such are the notions of algorithms, computability, complexity and randomness which are processed in the current thesis. The main issue, is part of the field of algorithmic information theory and is heavily based on some profound results of computation theory.

At first Turing's computation model is presented, which is a conceptually approachable introduction to the general notion of computability and some basic tools that are used later on. Subsequently by means Church-Turing thesis, we continue to recursion theory, through which some of the main results that are required to the foundation of the notions of the last chapter, are presented. There is also an emphasis to the ability of computation theory to produce negative results, as such will be the final result of this thesis.

Since all theoretical requirements for the foundation of the notion of algorithmic complexity have been covered, some interesting properties are presented, and a formalisation of the notion of randomness is introduced. Finally, based on a question about the existence of "true random" objects that often rises in fields relevant to informatics, a negative result is presented, according to the context and the given formal system of course, that implies the incompatibility of today's computers and true randomness.

Εισαγωγή

> Η αλγοριθμική θεωρία πληροφοριών είναι το κοκτέϊλ που προκύπτει αν βάλουμε την υπολογισιμότητα του Turing και την θεωρία πληροφοριών του Shannon σε ένα σέηκερ.
>
> *Chaitin*

Υπάρχουν ενίοτε μαθηματικές έννοιες που, παρότι απτές για τον κάθε άνθρωπο, η δυσκολία της μαθηματικής τυποποίησής τους είναι όχι μόνον αντιστρόφως ανάλογη της διαισθητικής τους προσέγγισης, αλλά -αφότου επιτευχθεί- οδηγεί σε αποτελέσματα ριζικής σπουδαιότητας.

Τέτοιες είναι λόγου χάρη οι έννοιες της υπολογισιμότητας, της πολυπλοκότητας και της τυχαιότητας, που συνιστούν αντικείμενο της προκείμενης εργασίας. Το κυρίως αντικείμενο εμπίπτει στον τομέα της Αλγοριθμικής Θεωρίας Πληροφοριών (Algorithmic Information Theory) και ερείδεται σε θεμελιώδη αποτελέσματα της Θεωρίας Υπολογσμού (Computation Theory).

Αρχικά εκτίθεται το κλασσικό μοντέλο υπολογισμού κατά Turing, που χρησιμεύει ως μια εννοιλογικά βατή εισαγωγή στην γενικότερη θεωρία υπολογισιμότητας (computability) και των εργαλείων που θα παρουσιαστούν αργότερα. Εν συνεχεία, επικαλούμενοι την Θέση Church-Turing, περνάμε στην Θεωρία Αναδρομής (Recursion Theory), στο πλαίσιο της οποίας εκτίθενται όσα βασικά αποτελέσματα είναι απαραίτητα για την θεμελίωση των εννοιών του τελικού κεφαλαίου. Έμφαση δίνεται στις ενδιάθετες δυνατότητες της Θεωρίας Υπολογισμού να παράγει αρνητικά αποτελέσματα, όπως άλλωστε το τελικό αποτέλεσμα της προκείμενης εργασίας.

Αφότου καλυφθούν βάσει των παραπάνω οι προϋποθέσεις για την καλή θεμελίωση της έννοιας της αλγοριθμικής πολυπλοκότητας (algorithmic complexity), παρουσιάζονται έπειτα ορισμένα ειδικά χαρακτηριστικά της και εισάγεται ο κατάλληλος φορμαλισμός για την έννοια της τυχαιότητας (randomness). Τέλος, εκκινώντας από ένα ερώτημα που ανακύπτει συχνά σε τομείς συναφείς της πληροφορικής, αποδεικνύται -εντός του αναπτυχθέντος πλαισίου και φορμαλισμού- ένα αρνητικό αποτέλεσμα που υποδηλώνει την ασυμβατότητα των υπολογιστών ως έχουν με την πραγματική τυχαιότητα.

Κεφάλαιο 1

Μηχανιστική Υπολογισιμότητα

1.1 Μία διαισθητική προσέγγιση της έννοιας του αλγορίθμου

Ορισμός 1.1 *Αριθμητική ή Αριθμοθεωρητική συνάρτηση είναι μια κ-μελής συνάρτηση* $f : \mathbb{N}^k \to \mathbb{N}$.

Θα ασχοληθούμε κυρίως με τις αριθμοθεωρητικές συναρτήσεις. Υπάρχουν υπερβολικά πολλές τέτοιες συναρτήσεις, οι περισσότερες από τις οποιές συμπεριφέρονται με τελείως απρόβλεπτο τρόπο. Εδώ, θα μας απασχολήσουν και θα προσπαθήσουμε να προσδιορίσουμε, εκείνες τις αριθμοθεωρητικές συναρτήσεις, οι οποίες συμπεριφέρονται ¨ομαλά' ή ¨προβλέψιμα¨.
Θα ονομάσουμε αυτές τις συναρτήσεις υπολογίσιμες και στη συνέχεια θα ασχοληθούμε με την τυποποίηση αυτού του όρου.

Μία αριθμοθεωρητική συνάρτηση είναι ¨μηχανιστικά ή αλγοριθμικά υπολογίσιμη¨ (effective computable), αν υπάρχουν σαφείς, καθοριστικοί κανόνες/οδηγίες, τους οποίους αν ακολουθήσει κάποιος, θα μπορέσει κατ΄ αρχήν να υπολογίσει την τιμή/έξοδο της για οποιεσδήποτε κ-άδες φυσικών εισόδου.
Βάση των σημερινών δεδομένων η διαισθητική αντίληψη που έχει ο καθένας για την έννοια του αλγορίθμου είναι αυτή ενός προγράμματος υπολογιστή, ωστόσο αλγόριθμος μπορεί να θεωρηθεί και μια οποιουδήποτε είδους συνταγή.
Σάυτό το κεφάλαιο θα εισάγουμε μια πρώτη τυποποίηση της έννοιας του αλγορίθμου, βάσει μηχανών Turing και Register.

Υποθέτουμε οτι ένας άνθρωπος πρέπει να υπολογίσει την τιμή μιας συνάρτησης για ένα δοσμένο όρισμα, ακολουθώντας ένα πεπερασμένο πλήθος οδηγιών. Κατα την εκτέλεση του υπολογισμού θα χρησιμοποιήσει ένα πεπερασμένο αριθμό διαφορετικών συμβόλων κάποιου είδους, όπως επίσης κάθε στιγμή θα μπορεί να εντοπίσει πεπερασμένες εμφανίσεις των συμβόλων αυτών. Πεπερασμένες πρέπει τέλος να είναι και οι οδηγίες. Εκτελώντας αυτές τις οδηγίες μπορεί μέσω μιας κίνησης να αλλάξει την κατάσταση του υπολογισμού κατα πεπερασμένο τρόπο Κατά το δοκούν, πχ αλλαζοντας, αφαιρώντας ή προσθέτοντας σύμβολα. Έτσι με μία διαδοχή τέτοιων κινήσεων θα οδηγηθεί από μία συμβολική έκφραση που παριστάνει το όρισμα, σε μία άλλη συμβολική έκφραση που παριστάνει την τιμή της συνάρτησης.

Ο άνθρωπος αυτός, που είναι ένας φυσικός υπολογιστής, κάνει τον υπολογισμό εκτελώντας μια σειρά απο μοριακές κινήσεις. Είναι λοιπόν εύλογο να αναρωτηθούμε, αν κάθε εκτελέσιμος υπολογισμός είναι ισοδύναμος με μια σειρά τέτοιων κινήσεων.

Μία τέτοια ανάλυση πραγματοποιήθηκε απ' τον Turing(1936) που προσπάθησε να ορίσει μια υπολογιστική μηχανή. Μία παρόμοια ανάλυση προτάθηκε και απο τον Post(1936). Έτσι προέκυψε η πρώτη τυποποίηση των υπολογίσιμων συναρτήσεων, μέσω των μηχανών Turing.

1.2 Μηχανές Turing

Σημείωση 1.1 *Οι μηχανές Turing είναι μια εξιδανικευμένη προσπάθεια τυποποίησης ενός υπολογιστή. Υπερτερεί ως προς τους πραγματικούς υπολογιστές, ως προς το γεγονός ότι έχουν αρθεί οι περιορισμοί μνήμης και χρόνου υπολογισμού αλλά και αποδοτικότητας που εδώ δεν μας αφορούν.*

Η γενική περιγραφή μιας μηχανής Turing, μία απ' τις πολλές που υπάρχουν στη βιβλιογραφία, είναι αυτή μιας κορδέλας, άπειρου μήκους, χωρισμένης σε τετράγωνες κυψέλες, πάνω στην οποία κινείται μια κεφαλή με την ικανότητα να διαβάζει και να γράφει μέσα στις κυψέλες, κάποιο από τα σύμβολα ενός πεπερασμένου αλφαβήτου ή ένα ειδικό σύμωολο κενου "-", υπό τον περιορισμό ο αριθμός των κυψελών που περιέχουν τέτοια σύμβολα να είναι πεπερασμένος.
Η κεφαλή μπορεί να εκτελέσει μία από τις εξής κινήσεις κάθε στιγμή:

(i) Να αντικαταστήσει το σύμβολο στη θέση που βρίσκεται με κάποιο άλλο.

(ii) Να κινηθεί Κατά μία θέση δεξία.

(iii) Να κινηθεί αριστερά.

Κάθε τέτοια στιχειώδης πράξη αποτελεί ένα βήμα υπολογισμού, και οδηγεί τη ΜΤ σε μία νέα "κατάσταση".

Ορισμός 1.2 *($Turing$, 1936)*
Μία ντετερμινιστική μηχανή $Turing$ είναι μια τριάδα $\mathcal{M} = (S, Q, F)$ όπου:

(i) *S είναι ένα πεπερασμένο σύνολο συμβόλων, το αλφάβητο της μηχανής, ένα απο τα σύμβολα του οποίου είναι το κενό σύμβολο.*

(ii) *Q είναι ένα πεπερασμένο σύνολο καταστάσεων για την $\mathcal{M}$, τέτοιο ώστε $Q \cap S = \oslash$, με q_0 αρχική και q_f την τελική κατάσταση.*

(iii) *F η συνάρτηση μετάβασης*

$$F : (Q - q_f) \times S \to Q \times (S \cup (R, L)),$$

με $R, L \notin S \cup Q$.

Η εναλλακτικά ένα σύνολο από οδηγίες $I_1, ..., I_p$ που ανήκουν στις εξής κατηγορίες:

(α΄) $q_a s_b s_c q_d$ *: αν στην κατάσταση* q_d *η κεφαλή διαβάσει* s_b *το αντικαθιστά με το* s_c *και προχωρά στην κατάσταση* q_d*.*

(β΄) $q_a s_b R q_d$ *: αν στην* q_a *διαβάσει* s_b *προχωρά μία θέση δεξιά, και μπαίνει στην κατάσταση* q_d*.*

(γ΄) $q_a s_b L q_d$ *: αν στην* q_a *διαβάσει* s_b *προχωρά μία θέση αριστερά, και προχωρά στην κατάσταση* q_d*.*

Τα συγκεκριμένα χαρακτηριστικά του μοντέλου ΜΤ που ορίσθηκε πιο πάνω δεν έχουν μεγάλη σημασία, αφού μας απασχολεί η υπολογιστική ισχύς και όχι η αποδοτικότητα. Θα παραθέσουμε κάποια ενδιαφέροντα αποτελέσματα για τις ΜΤ.

- **Καταστάσεις**. *Μία μόνο κατάσταση δεν είναι γενικά αρκετή για να υπολογίσει κάθε αναδρομική συνάρτηση, δύο ωστόσο καταστάσεις αρκούν*(Shanon 1956, Wang 1957).

- **Αλφάβητο**. *Δύο σύμβολα αρκούν για να υπολογίσουμε κάθε αναδρομική συνάρτηση* (Shannon 1956). Σίγουρα πρέπει να έχουμε τουλάχιστον δύο σύμβολα, αφού θεωρούμε και το κενό σύμβολο. Γενικά μπορούμε να περιορίσουμε τον αριθμό των συμβόλων, αυξάνοντας τον αριθμό των καταστάσεων.

- **Διαγραφή**. *Όλες οι αναδρομικές συναρτήσεις μπορούν κάλλιστα να υπολογιστούν απο Μηχανές χωρίς λειτουργία διαγραφής* (Wang 1957). Βασικά, δεδομένης μιας ΤΜ, μπορούμε να προσομοιάσουμε τη λειτουργία της, με μια άλλη Μηχανή χωρίς λειτουργία διαγραφής. Το αποτέλεσμα αυτό υπονοεί, ότι δε χρειαζόμαστε κατ αρχήν, επανεγγράψιμο υλικό για την εξωτερική μνήμη των υπολογιστών.

- **Ταινίες και κεφαλές**.Εδώ η ελευθερία κατασκευής της ΜΤ είναι πρακτικά απόλυτη.*Μία Μηχανή Turing με πεπερασμένες ταινίες, καθε μία με την δική της (πεπερασμένη, η ακόμα και άπειρη αριθμήσιμη) διάσταση, και τις δικές της πεπερασμένες κεφαλές να την διαβάζουν ταυτόχρονα, μπορεί να προσομοιωθεί απο μια ΜΤ με ταινία άπειρη προς την μία μόνο κατεύθυνση, που την διαβάζει μία μόνο κεφαλή* (Hartmanis and Stearns 1965). Ωστόσο, χρειαζόμαστε την ελευθερία κίνησης και προς τις δύο κατευθύνσεις, καθώς ο περιορισμός της στη μία θα σήμαινε συμβατότητα με πεπερασμένη ή περιοδική κίνηση, σε κυψέλες εκτός αυτών που περιέχουν το input.

- **Ντετερμινισμός**. Η ΜΤ που ορίσαμε είναι ντετερμινιστική υπό την έννοια του ότι οι εντολές που δέχεται πρέπει να είναι συνεπείς (δλδ το πολύ μία να είναι εφαρμόσιμη σε κάθε κατάσταση). Αρκετά νωρίς προστέθηκαν χαρακτηριστικά τυχαιότητας στις ΜΤ, από τους Shannon(1948), De Leeuw, Moore και Shapiro(1956). Ουσιαστικά υπάρχουν δύο μοντέλα ΜΤ, οι ντετερμινιστικές και οι μη-ντετερμινιστικές οι οποίες συμπεριφέρονται με ένα διφορούμενο τρόπο, όπου δεχόμενες αλληλοσυγκρουόμενες εντολές, διαλέγουν τυχαία και εφαρμόζουν μία απ' αυτές. Η υπολογιστική τους ισχύς, δεν υπερβαίνει αυτή των ντετερμινιστικών ΤΜ. Υπάρχουν και οι πιθανοκρατικές ΤΜ, στις οποίες διαφορετικές οδηγίες έχουν μια πιθανότητα, να επιλεγούν απο την Μηχανή.

1.3 Μηχανές Register

Εδώ θα χρησιμοποίσουμε μια πιο ευέλικτη εκδοχή εξιδανικευμένης μηχανής, η οποία όμως είναι ισοδύναμη με μια ΤΜ, δηλαδή οποιονδήποτε υπολογισμό μπορούμε να κάνουμε με μια ΤΜ μπορούμε να τον κάνουμε και μ'αυτή τη μηχανή, τη μηχανή "$Register$", ή απλά άβακα.

Ορισμός 1.3 *Μια μηχανή Register αποτελείται από ένα πεπερασμένο σύνολο "μετρητών", $R_1, ..., R_s$. Κάθε μετρητής έχει τη δυνατότα να έχει πάνω του καταχωρημένο έναν οποιονδήποτε φυσικό $z_i \in \mathbb{N}$, όπου προφανώς $1 \leqslant i \leqslant s$. Μπορούμε να να φανταστούμε κάθε μετρητή R_i σαν ένα δοχείο που περιέχει z_i βόλους. Οι στοιχειώδείς πράξεις που μπορούμε να κάνουμε είναι να αφαιρέσουμε, ή να προσθέσουμε ένα βόλο σε κάθε κουτί. Ένα πρόγραμμα MR, είναι ένα flowchart που αποτελείται από τους εξής τύπους οδηγιών.*

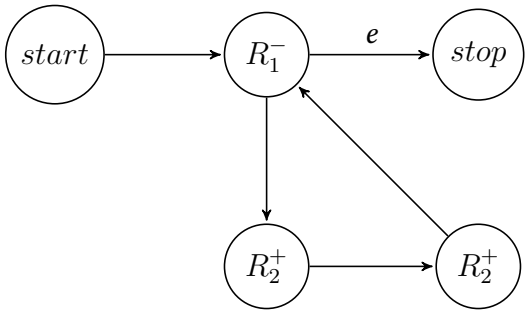
(i) Την εντολή προσαύξησης ενός μετρητή, R_i^+. Αυτή η εντολή αντικαθιστά το z_i με το $z_i + 1$ (δηλαδή προσθέτει ένα βόλο στο κουτί R_i) και μεταβαίνει στην επόμενη οδηγία μέσω ενός συμβόλου-βελους.

(ii) Την αφαίρεση, R_i^-, "αφαιρεί ένα βόλο από το αντίστοιχο κουτί" και προχωρά στην επόμενη οδηγία μέσω ενός βέλους. Αν ο μετρητής αδειάσει, δηλαδή $z_i = 0$, τότε προχωρά στην επόμενη οδηγία μέσω ενός βέλους με την ένδειξη e.

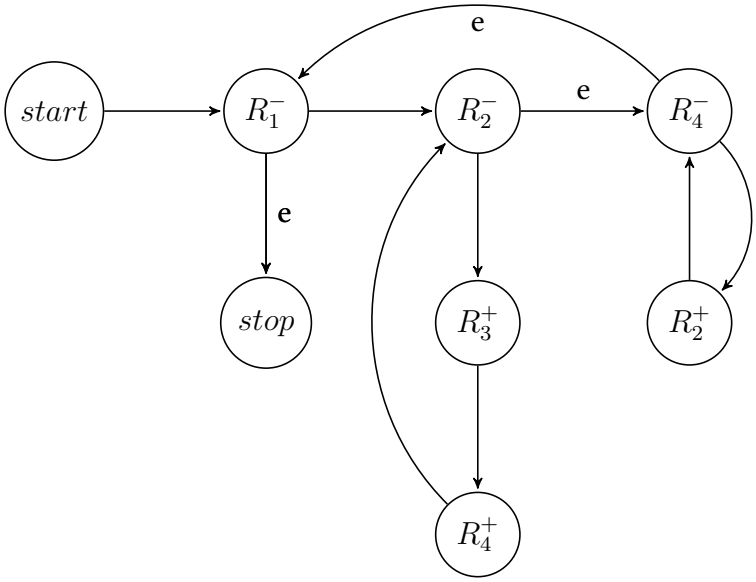
(iii) Εντολές έναρξης κ τερματισμού. Κάθε πρόγραμμα έχει μόνο μία εντολή έναρξης.

Πρόταση 1.1 *Κάθε Register υπολογίσιμη συνάρτηση είναι Turing υπολογίσιμη. Μπορούμε εύκολα να προσομοιώσουμε την κάθε μηχανή Register αντστοιχίζοντας τα κουτια της και το περιεχόμενο τους, σε διαδοχικά μη-κενά σύμβολα μιας μηχανής Turing.*

Παράδειγμα 1.1 *Παρατηρούμε ότι το πρόγραμμα αυτό αν ξεκινήσει με x στο R_1 κ y στο R_2 θα τερματίσει κάποια στιγμή μετα από πεπερασμένο αριθμό βημάτων, έχοντας το αποτέλεσμα $2x + y$ στο R_2.*



Παράδειγμα 1.2 *Μία RM που υπολογίζει τη 2-μελή συνάρτηση $f(x, y) = xy$*



1.4 Υπολογίσιμες Συναρτήσεις και Κατηγορήματα

Ορισμός 1.4 *Μία k-μελής αριθμοθεωρητική συνάρτηση $f(\vec{x})$ θεωρείται υπολογίσιμη, εάν υπάρχει μηχανή Register η αλλιώς πρόγραμμα $\mathcal{P}$ τέτοιο ώστε, $\forall \vec{x} \in \mathbb{N}^k$ αν το $\mathcal{P}$ ξεκινήσει με τιμή x_i σε κάθε R_i θα τερματίσει κάποια στιγμή με την τιμή $f(\vec{x})$ στο R_{k+1}.*

Σημείωση 1.2 *Αργότερα θα δούμε ότι ο ενδεχομένως αρχικά ασαφής όρος της μηχανιστικής υπολογισιμότητας, θα ταυτιστεί όχι μόνο με τη Register και την Turing υπολογισιμότητα αλλά και με την κλάση των αναδρομικών συναρτήσεων. Έτσι η εγγενώς ασαφής έννοια της υπολογισιμότητας, αποκτά έναν πολύ ισχυρό φορμαλισμό.*

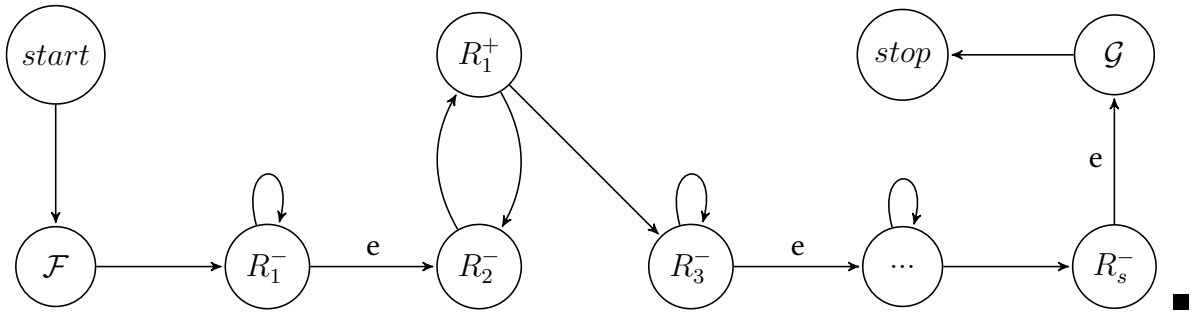
Μεγάλος αριθμός απ' τις πιο γνωστές μας αριθμοθεωρητικές συναρτήσεις, είναι υπολογίσιμες. Όμως για να αποδείξουμε τη Register υπολογισιμότητα μιας συνάρτησης, θα θέλαμε να αποφύγουμε το σχεδιασμό του προγράμματος που την υπολογίζει. Έτσι θα αναπτύξουμε κάποια απλά αλλα ισχυρά εργαλεία για να αποδεικνύουμε οτι δεδομένες συναρτήσεις είναι Register υπολογίσιμες.
Ξεκινάμε απ' το απλό αλλα βασικό θεώρημα.

Θεώρημα 1.2 *(Γενικευμένη Σύνθεση). Έστω g m-μελής αριθμοθεωρητική συνάρτηση και $f_1 ... f_m$ k-μελείς αριθμοθεωρητικές συναρτήσεις. Τότε υπάρχει μοναδική k-μελής h τέτοια ώστε*

$$h(x_1, ..., x_k) = g(f_1(x_1, ..., x_k), ..., f_m(x_1, ..., x_k)) \; \forall \vec{x} \in \mathbb{N}^k.$$

Επιπλέον αν οι g και $f_1, ..., f_m$ είναι υπολογίσιμες, τότε και η h είναι υπολογίσιμη.

Απόδειξη. Έστω $\mathcal{F}$ και $\mathcal{G}$ προγράμματα που υπολογίζουν τις f και g αντίστοιχα. Μπορούμε χωρίς βλάβη της γενικότητας να υποθέσουμε ότι οι $\mathcal{F}$ και $\mathcal{G}$ χρησιμοποιούν το ίδιο σύνολο μετρητών.



Ορισμός 1.5 *Ένα k-μελές αριθμοθεωρητικό κατηγόρημα, είναι ένα σύνολο $P \subseteq \mathbb{N}^k$. Βλέπουμε το P σαν πρόταση με k μεταβλητές.*

$$P(\vec{x}) \equiv \text{"η k-αδα } \vec{x} \text{ είναι στοιχείο του P"}$$

Ορισμός 1.6 *Η χαρακτηριστική συνάρτηση του P, είναι η k-μελής αριθμοθεωρητική συνάρτηση:*

$$\chi_P = \begin{cases} 1 & \text{αν η P είναι αληθής} \\ 0 & \text{αν η P είναι ψευδής} \end{cases}$$

$\forall \vec{x} \in \mathbb{N}^k$. Κάθε κατηγόρημα P είναι υπολογίσιμο αν και μόνο αν η χαρακτηριστική του είναι Register υπολογίσιμη.

Θα αναπτύξουμε τώρα κάποια εργαλεία για να αποδεικνύουμε ότι διάφορα κατηγορήματα και συναρτήσεις είναι υπολογίσιμες.

Θεώρημα 1.3 *(Λογικοί τελεστές) Έστω P, Q υπολογίσιμα k-μελή κατηγορήματα, τότε η κλάση των υπολογίσιμων κατηγορημάτων είναι κλειστή στους ακόλουθους λογικούς τελεστές:*

(i) $P \wedge Q \equiv$ *"P και Q" ή αλλιώς* $P \cap Q$.

(ii) $P \vee Q \equiv$ *"P ή Q" ή αλλιώς* $P \cup Q$.

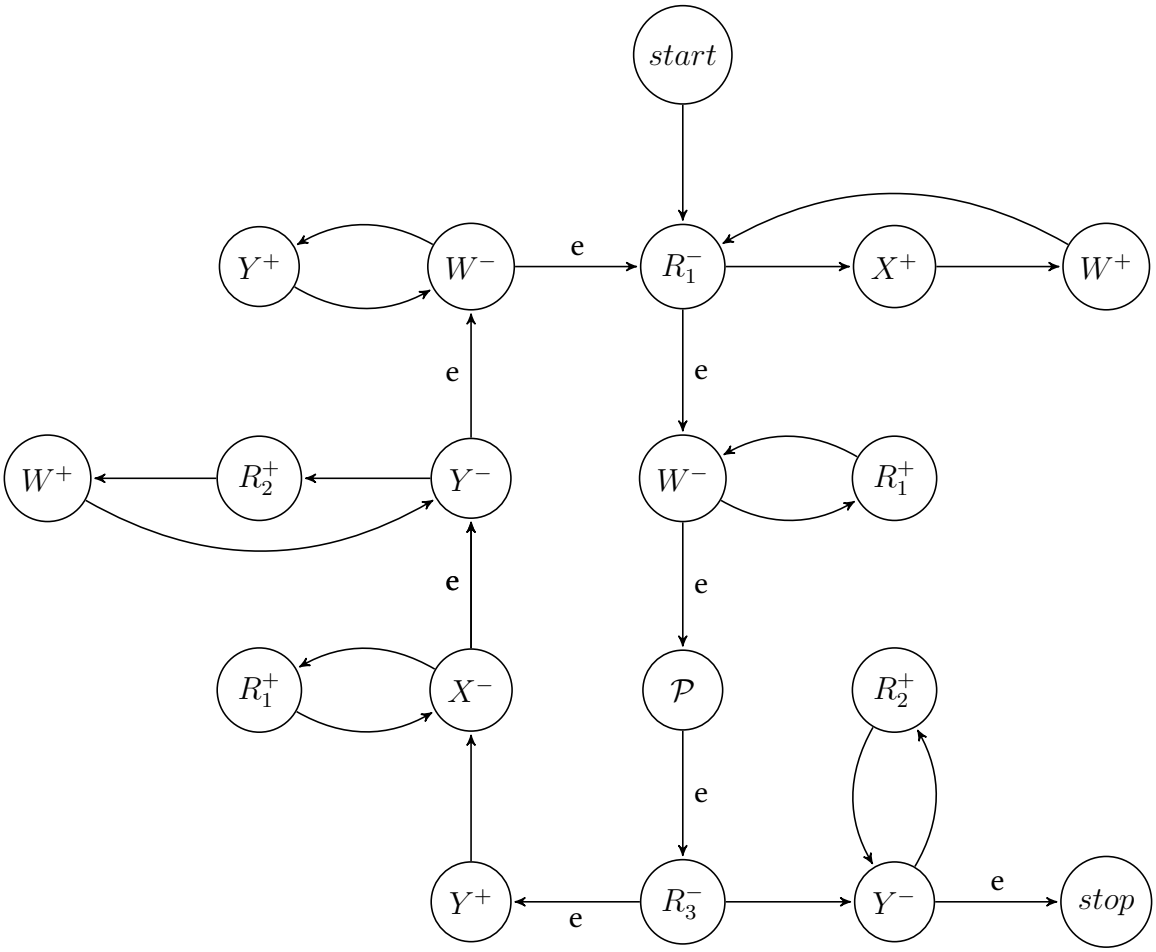
(iii) $\neg P \equiv$ *"όχι P" ή αλλιώς* $\mathbb{N}^k \backslash P$.

Θεώρημα 1.4 *(μ-τελεστής) Έστω P k-μελές αριθμοθεωρητικό κατηγόρημα. Υποθέτουμε πως $\forall$ k-αδα $\vec{x} \in \mathbb{N}^k$, $\exists$ τουλάχιστον ένα $y \in \mathbb{N}$ τέτοιο ώστε η $P(\vec{x})$ να ισχύει. Τότε υπάρχει μοναδική k-μελής f :*

$$f(\vec{x}) = \text{το ελάχιστο } y \text{ τέτοιο ώστε η } P(\vec{x}) \text{ να ισχύει.} = \mu y P(\vec{x}, y).$$

Επιπλέον αν P υπολογίσιμο, τότε και η f ειναι υπολογίσιμη.

Απόδειξη. Το ίδιο ουσιαστικά αποτέλεσμα θα αποδείξουμε στο επόμενο κεφάλαιο, ωστόσο θα δούμε μια αρχική απόδειξη μέσω μηχανών Register, που βοηθάει στην διαισθητική προσέγγιση, των εννοιών και εργαλείων που έχουμε δει μέχρι στιγμής.



Αρχικά υποθέτουμε ότι για κάθε k-άδα εισόδου $\vec{x}$ το σύνολο $\{y \in \mathbb{N} : P(\vec{x}, y)\}$ είναι μη κενό και συνεπώς έχει ελάχιστο στοιχείο. Το στοιχείο αυτό είναι το $\mu y P(\vec{x}, y)$.
Έστω τώρα ότι η P είναι υπολογίσιμη. Θα δείξουμε ότι και η f είναι υπολογίσιμη. Για απλότητα θα πάρουμε k=1. Έτσι το P είναι ένα 2-μελές κατηγόρημα, η $f(x)$ είναι μονομελής συνάρτηση και $f(x) = \mu y P(x, y)$ για κάθε x. Το $\mathcal{P}$ τώρα, είναι το πρόγραμμα που υπολογίζει την χ_P χρησιμοποιώντας τα R_i. Έτσι η Register που αντιστοιχεί στο $\mathcal{P}$ ξεκινά με όρισμα $x, y, 0, ...0$ στα $R_1, R_2, ..., R_n$ και τερματίζει με 1 στο R_3 αν η $P(x, y)$ ισχύει και με 0 στο R_3 αν δεν ισχύει. Προσθέτουμε τώρα τρεις ακόμα μετρητές, τους $X = R_{n+1}, Y = R_{n+2}, W = R_{n+3}$ και διαμορφώνουμε το πρόγραμμα $\mathcal{Q}$ που απεικονίζεται πιο πριν και ισχυριζόμαστε ότι το $\mathcal{Q}$ υπολογίζει την f. Πράγματι το πρόγραμμα αυτό τρέχει διαδοχικά τα $\mathcal{P}(x, 0), \mathcal{P}(x, 1), ..., \mathcal{P}(x, y)$ μέχρι να εντοπίσει ένα y ώστε να τερματίζει το $\mathcal{P}$ με μη κενό R_3, ενώ παράλληλα κρατά αντίγραφα των x, y στα X, Y. ■

1.5 Η Θέση Church-Turing

> Με αυτή την ιδέα (αναφερόμενος στη θεωρία αναδρομής), για πρώτη φορά κάποιος κατάφερε να προσδώσει έναν απόλυτο ορισμό σε μια ενδιαφέρουσα επιστημολογική έννοια, δηλαδή έναν ορισμό που δεν εξαρτάται από το φορμαλισμό που θα επιλέξουμε.
>
> *Gödel, 1946*

Θεώρημα 1.5 *Τα επόμενα είναι ισοδύναμα:*

(i) *Η f είναι αναδρομική.*

(ii) *Η f είναι ορίσιμη.*

(iii) *Η f είναι Herbrand-Gödel υπολογίσιμη.*

(iv) *Η f είναι Turing υπολογίσιμη.*

(v) *Η f είναι flowchart (ή while) υπολογίσιμη.*

(vi) *Η f είναι ορίσιμη μέσω λ-λογισμού.*

Κατά τη διάρκεια της δεκαετίας του '30 προσπάθειες με διαφορετικές αφετηρίες, κινούμενες σχεδόν παράλληλα, κατέληξαν στα ίδια αποτελέσματα. Αυτές οι προσπάθειες, των Kleene, Church, Gödel, Turing και Post, οδήγησαν σε διαφορετικούς φορμαλισμούς της ίδιας έννοιας. Η κλάση των συναρτήσεων που μελετούσαν, ήταν η ίδια και ουσιαστικά παρείχε ένα άνω φράγμα στην κάπως ασαφή διαισθητική έννοια του αλγορίθμου.

Αν και η ισοδυναμία της θεωρίας αναδρομής, των μηχανών Turing και του λ-λογισμού μπορεί να αποδειχθεί σε μορφή θεωρημάτων, η ισοδυναμία των αναδρομικών και των Turing-υπολογίσιμων συναρτήσεων με την έννοια της "μηχανιστικής υπολογισιμότητας" δεν μπορεί παρα να μείνει στη μορφή υπόθεσης, που υποστηρίζεται βέβαια από πλήθος ενδείξεων. Αυτό συμβαίνει όχι λόγω κάποιας έλλειψης επιχειρημάτων υπέρ της, άλλα λόγω της αοριστίας του δεύτερου σκέλους της υπόθεσης και της μη επαρκούς τυποποίησης της μηχανιστικής υπολογισιμότητας. Αυτά τα αποτελέσματα έχουν γίνει γνωστά ως Θέσεις του Church και του Turing αντίστοιχα.

Τα επιχειρήματα που υποστηρίζουν τη Θέση Church-Turing ανήκουν στις ακόλουθες κατηγορίες:

(α′) Ευρετικά επιχειρήματα: Το μεγαλύτερο μέρος των "μηχανιστικά υπολογίσιμων" συναρτήσεων, μας έιναι γνωστό και έχει διαπιστωθεί ότι είναι δυνατός ο μετασχηματισμός

τους σε αναδρομικές συναρτήσεις. Κάθε προσπάθεια να αναπτυχθεί κάποια μέθοδος που θα οδηγεί με μηχανιστική διαδικασία έξω από την κλάση των αναδρομικών συναρτήσεων, έχει βρεθεί σε αδιέξοδο.

(β′) Η ισοδυναμία των φορμαλισμών: Όπως αναφέρθηκε, τρείς έννοιες που είχαν τελικά το ίδιο αντικείμενο έρευνας, αναπτύχθηκαν σχεδόν ταυτόχρονα, οι γενικώς αναδρομικές συναρτήσεις και οι συναρτήσεις που μπορούν να παρασταθούν μέσω λ-λογισμού (Church,Kleene) και οι υπολογίσιμες συναρτήσεις (Turing, Post). Η ισοδύναμία των τριών φορμαλισμών αποδείχθηκε άμεσα.
Το γεγονός ότι οι διαφορετικές έννοιες, οδηγούν στην ίδια κλάση συναρτήσεων είναι ισχυρή ένδειξη ότι αυτή η κλάση είναι θεμελιώδης.

(γ′) Η Turing-υπολογισιμότητα: Η έννοια αυτή είναι εμφανώς μία άμεση προσπάθεια να τυποποιηθεί μαθηματικά η έννοια της μηχανιστικής υπολογισιμότητας. Η κεντρική ιδέα άλλωστε είναι η προσωμοίωση από μια μηχανή των υπολογισμών που μπορεί να κάνει ένας άνθρωπος ακολουθώντας δεδομένες οδηγίες.

Σ' αυτό το κεφάλαιο είδαμε μια πρώτη τυποποίηση της έννοιας της υπολογισιμότητας. Δεν θα ασχοληθούμε περισσότερο με αυτή τη μορφή υπολογισιμότητας, πέραν ίσως απο κάποιες αναφορές σε κάποιες από το πλήθος αναλογιών που υπάρχουν ανάμεσα στην θεωρία αναδρομής και στην θεωρία υπολογισμού μέσω μηχανών. Ωστόσο στόχος του κεφαλαίου ήταν να καταστήσει πιο οικεία την κλάση των συναρτήσεων με τις οποίες θα ασχοληθούμε αλλά και να εισάγει μια διαισθητικά πιο βατή τυποποίηση κάποιων εργαλιών που θα δούμε στη συνέχεια, καθώς επίσης και κάποιες από τις δυνατότητες γενίκευσης της θεωρίας μας πέραν των αριθμοθεωρητικών συναρτήσεων.

Κεφάλαιο 2

Θεωρία Αναδρομής

> Αν οι γενικώς αναδρομικές συναρτήσεις ειναι ο ισοδύναμος φορμαλισμός της μηχανιστικής υπολογισιμότητας, τότε η τυποποίηση τους μπορεί να παίξει ένα ρόλο στην ιστορία των συνδιαστικών μαθηματικών σχεδόν το ίδιο σημαντικό με αυτόν της διατύπωσης της γενικής ιδέας των φυσικών αριθμών.
>
> *Post, 1944*

Το κεφάλαιο αυτό περιέχει τον κορμό της θεωρίας αναδρομής, των βασικών εννοιών, αποτελεσμάτων και μεθόδων που χρησιμοποιούνται. Για παράδειγμα θα ορίσουμε τις μερικώς αναδρομικές συναρτήσεις και το συνολοθεωρητικό τους αντίστοιχο, τα αναδρομικώς απαριθμήσιμα σύνολα, στη συνέχεια θα δούμε καποιες θεμελιώδεις μεθόδους όπως η απαρίθμηση, η διαγωνοποιηση και τις έννοιες των βαθμών και των αναγωγών, με σκοπό να καταλήξουμε σε μερικά απ' τα πιο σημαντικά συμπεράσματα, όπως το θεώρημα κανονικής μορφής και το θεώρημα αναδρομής και τελικά να περάσουμε σε κάποια αποτελέσματα μη υπολογισιμότητας.

2.1 Κλάσεις αναδρομικών συναρτήσεων

2.1.1 Πρωτογενώς αναδρομικές συναρτήσεις, η κλάση PR

Η κλάση των πρωτογενώς αναδρομικών συναρτήσεων είναι πολύ κατανοητή, ως επακόλουθο των πολύ ισχυρών ιδιοτήτων κλειστότητας που έχει.

Ορισμός 2.1 *(Kleene, 1938) Η κλάση των πρωτογενώς αναδρομικών συναρτήσεων PR (primitive recursive) είναι η μικρότερη κλάση αριθμοθεωρητικών συναρτήσεων η οποία:*

(I) Περιέχει τις βασικές συναρτήσεις:

- $\mathcal{O}(x) = 0$ *την μηδενική,*
- $\mathcal{S}(x) = x + 1$ *την συνάρτηση επομένου,*
- $\mathcal{I}_i^n = x_i \ (1 \leqslant i \leqslant n)$ *και την προβολή.*

(II) Είναι κλειστή ως προς τη σύνθεση δηλαδή το σχήμα που δοσμένων $g_1, ..., g_m,$ παράγει:

$$f(\vec{x}) = h(g_1(\vec{x}), ..., g_m(\vec{x})),$$

όπου η αριστερή πλευρά της εξίσωσης δεν ορίζεται, όταν τουλάχιστον μία από τις τιμές $g_1, ..., g_m,$ δεν ορίζεται.

(III) Είναι κλειστή ως προς το σχήμα της πρωταρχικής αναδρομής, δηλαδή:

$$f(\vec{x}, 0) = g(\vec{x})$$
$$f(\vec{x}, y + 1) = h(\vec{x}, y, f(\vec{x}, y))$$

Σημείωση 2.1 *Ένας ισοδύναμος ορισμός της PR είναι ο εξής: Η συνάρτηση f είναι στην PR αν υπάρχει πεπερασμένη ακολουθία $f_1, .., f_n$ τέτοια ώστε $f_n = f$ και $\forall i \leqslant n$, η f_i είναι είτε αρχική, είτε προέρχεται απο δύο προηγούμενες $f_k, f_j (k, j \leqslant i$ με πρωτογενή αναδρομή ή με σύνθεση.*

Πολλές από τις γνωστές μας συναρτήσεις ανήκουν στην PR και μάλιστα η πρωτογενής αναδρομή είναι είναι ένα βασικό εργαλείο για την απόδειξη της υπολογισιμότητας μιας συνάρτησης (που βέβαια στην ουσία είναι ο ορισμός με επαγωγή).

Παράδειγμα 2.1 *Ας δούμε μερικές οικείες συναρτήσεις:*

- *Μπορούμε να ορίσουμε για παράδειγμα τη συνάρτηση $f(y) = y!$ αναδρομικά:*
  $0! = 1$
  $(y + 1)! = y! \cdot (y + 1)$
- *Όπως επίσης και τη συνάρτηση ελαχίστου:*
  $min(a, b) = b - (b - a)$
  $min(a_1, ..., a_n) = min(...min(a_1, a_2), a_3)..., a_n)$

- *Ή τα φραγμένα αθροίσματα:*
  $\sum_{y\leqslant 0} f(\vec{x},y) = f(\vec{x},0)$
  $\sum_{y\leqslant z+1} f(\vec{x},y) = (\sum_{y=z} f(\vec{x},y)) + \sum_{y=z+1} f(\vec{x},z+1)$
  $\sum_{y<z} f(\vec{x},y) = \sum_{y\leqslant z-1} f(\vec{x},y)$

Ορισμός 2.2 *Ένα σύνολο $\subseteq \mathbb{N}^k$ λέγεται πρωτογενώς αναδρομικό ανν η χαρακτηριστική του συνάρτηση είναι πρωτογενώς αναδρομική.*

Σημείωση 2.2 *Ουσιαστικά, τα σύνολα που μας ενδιαφέρουν εδώ δεν είναι απλά υποσύνολα του $\mathbb{N}^k$ αλλα σύνολα της μορφής $A = \{\vec{x} : f(\vec{x})\}$, και αντιστοιχούν σε κάποια ιδιότητα ή σχέση που τα ορίζει. Έτσι συχνά θα αναφερόμαστε στα PR σύνολα, ως πρωτογενώς αναδρομικά κατηγορήματα, που είναι ισοδύναμο. Αργότερα θα χρησιμοποιήσουμε και τον όρο πρόβλημα για να αναφερθούμε σε διάφορα σύνολα.*

Πρόταση 2.1 *Έστω $\mathcal{R}$ k+1-μελές PR κατηγόρημα, τότε ορίζουμε $g : \mathbb{N}^{k+1} \to \mathbb{N}$ ως εξής: $g(x_1,...,x_k,z) = \mu y \leqslant z\mathcal{R}(x_1,...,x_k,y)$ όπου $\mu y \leqslant z\mathcal{R}(x_1,...,xk,y)$ ο φραγμένος μ-τελεστής πάνω στην $\mathcal{R}$. Η g ανήκει στην PR.*

Απόδειξη. Ορίζουμε την g ως εξής:

$$(1)\quad g(n_1,n_2,\ldots,n_k,0) = \begin{cases} 0 & : f(n_1,n_2,\ldots,n_k,0) = 0 \\ 1 & : \text{αλλιώς} \end{cases}$$

$$(2)\quad g(n_1,n_2,\ldots,n_k,z+1) = \begin{cases} g(n_1,n_2,\ldots,n_k,z) & : g(n_1,n_2,\ldots,n_k,z) \leq z \\ z+1 & : g(n_1,n_2,\ldots,n_k,z) = z+1 \text{ και } f(n_1,n_2,\ldots,n \\ z+2 & : \text{αλλιώς} \end{cases}$$

■

Πρόταση 2.2 *Τα PR κατηγορήματα είναι κλειστά στους λογικούς τελεστές, σύζευξης, διάζευξης και άρνησης.*

Παράδειγμα 2.2 *Το κατηγόρημα $Prime(x) \equiv$ "ο x είναι πρώτος αριθμός", είναι στην PR. Αρκεί να γράψουμε το κατηγόρημα ως εξής : $Prime(x) \equiv x > 1 \wedge \neg(\exists u < x)(\exists v < x)[x = u \cdot v]$, και το ζητούμενο έπεται άμεσα από τις ιδιότητς κλειστότητας της PR.*

2.1.2 Η συνάρτηση Ackermann

Εκτός από τις πρωτογενώς αναδρομικές συναρτήσεις, υπάρχει επίσης μεγάλος αριθμός συναρτήσεων "περιέχουν" αναδρομές, υπό την έννοια ότι μπορούν να οριστούν επαγωγικά. Είναι εύλογο να αναρωτηθούμε αν υπάρχουν αναδρομικές συναρτήσεις που δεν μπορούν να αναχθούν σε πρωτογενή αναδρομή.
Το πρόβλημα αυτό προέκυψε από τον Hilbert το 1926, κατα την ενασχόληση του με το πρόβλημα του συνεχούς και απαντήθηκε απο τον Ackermann το 1928.

Ορισμός 2.3 *(Συνάρτηση του Ackermann) Η συνάρτηση του Ackermann ορίζεται μέσω της λεγόμενης διπλής αναδρομής :*

$$A(x,y) = \begin{cases} y+1 & : x = 0 \\ A(x-1,1) & : x > 0, y = 0 \\ A(x-1, A(x,y-1)) & : \textit{αλλιώς} \end{cases}$$

Η συνάρτηση Ackermann, είναι μια γενίκευση της εκθετικής συνάρτησης, και οι τιμές της αυξάνονται πιο γρήγορα απο οποιαδήποτε PR συνάρτηση. Η παράμετρος x παριστάνει το ρυθμό αύξησης της συνάρτησης , η αλλιώς την τάξη των εκθετών.

Παράδειγμα 2.3

$A(m,n)$	$m=0$	$m=1$	$m=2$	$m=3$
$n=0$	1	$A(0,1)$	$A(1,1)$	$A(2,1)$
$n=1$	2	$A(0,A(1,0))$	$A(1,A(2,0))$	$A(2,A(3,0))$
$n=2$	3	$A(0,A(1,1))$	$A(1,A(2,1))$	$A(2,A(3,1))$
$n=3$	4	$A(0,A(1,2))$	$A(1,A(2,2))$	$A(2,A(3,2))$
$n=4$	5	$A(0,A(1,3))$	$A(1,A(2,3))$	$A(2,A(3,3))$

Ενδεικτικά η $A(4,4)$ παίρνει τιμή $2^{2^{2^{2^{2^2}}}} - 3$.

Πρόταση 2.3 *Η συνάρτηση Ackermann δεν είναι πρωτογενώς αναδρομική.*

Σημείωση 2.3 *Η συνάρτηση Ackermann αναπτύχθηκε για να αποδείξει οτι υπάρχει μεγαλύτερη κλάση συναρτήσεων που περιλαμβάνει τις PR, ωστόσο η αντίστροφη της έχει μεγάλες εφαρμογές στον καθορισμό της πολυπλοκότητας διαφόρων υπολογιστικών προβλημάτων.*

2.1.3 Ολικές και μερικές Αναδρομικές συναρτήσεις, η κλάση R

Θα εισάγουμε ένα νέο σχήμα εδώ την μ-ελαχιστοποίηση:

$$f(\vec{x}) = \mu y((g(\vec{x}, y) = 0)$$

Αν κλείσουμε την κλάση των PR ως προς το σχήμα της μ-ελαχιστοποίησης, προκύπτει μια νέα κλάση, αυτή των αναδρομικών συναρτήσεων.

Ορισμός 2.4 *(Kleene, 1938) Η κλάση των (μερικώς) αναδρομικών συναρτήσεων, είναι η μικρότερη κλάση αριθμοθεωρητικών συναρτήσεων, η οποία:*

(I) Περιέχει τις αρχικές συναρτήσεις $\mathcal{O}, \mathcal{S}, \mathcal{I}_i^n$

(II) Είναι κλειστή ως προς τη σύνθεση.

(III) Είναι κλειστή ως προς την πρωτογενή αναδρομή.

(IV) Είναι κλειστη ως προς την χωρίς περιορισμούς μ-αναδρομή, δηλαδή το σχήμα (με δοσμένη g):

$$f(\vec{x}) = \mu y[(\forall z \leqslant y)(g(\vec{x}, z) \downarrow) \wedge g(\vec{x}, y) = 0]$$

όπου η g δεν ορίζεται αν δεν υπάρχει τέτοιο y.

Αξίζει να σημειωθεί, ότι δεν χρησιμοποιούμε τον αρχικό απλό ορισμό του σχήματος της μ-ελαχιστοποίησης που αγνοεί τη σύγκλιση της συνάρτησης, αν και κάτι τετοιο θα ήταν αποδεκτό για να ορίσουμε τις ολικές αναδρομικές συναρτήσεις. Ο λόγος θα γίνει εμφανής αργότερα.

Πρόταση 2.4 *Η συνάρτηση Ackermann ανήκει στο R.*

2.2 Κάποια θεμελιώδη θεωρήματα και εργαλεία

2.2.1 Απαρίθμηση

Πιθανώς πρώτος ο Leibniz(1666), αναζήτησε μια σύνδεση της αριθμητικής, με τη φυσική γλώσσα και ονειρεύτηκε την υποκατάσταση των λογικών επιχειρημάτων στην γλώσσα από υπολογιστικές πράξεις. Μάλιστα, προχώρησε σε μια πρώτη κωδικοποιήση των απλούστερων εκφράσεων, και κάποιων απλών τρόπων σύνδεσης τους, ωστόσο το έργο του παρέμεινε αδημοσίευτο και δεν επηρρέασε τις σύγχρονες προσπάθειες.

Στη συνέχεια ο Hilbert το 1904 οραματίστηκε την απαρίθμηση μέσω της τυποποίησης των αποδείξεων συνέπειας στην Αριθμητική, ωστόσο ο Gödel μόλις το 1931, μπόρεσε να τη χρησιμοποιήσει στο έργο του. Την ίδια μέθοδο κατέληξε να χρησιμοποιήσει ανεξάρτητα και ο Tarski(1936) Κατά τις έρευνές του πάνω στην έννοια της αλήθειας.

Παραδόξως η απαρίθμηση δεν κατάφερε να θωρακίσει τη γλώσσα απέναντι στις "παράλογες" προτάσεις, αντιθέτως αποκάλυψε τα κενα στην Αριθμητική, κι έτσι τα γλωσσικά παράδοξα χρησιμοποιήθηκαν τελικά για να κατδείξουν τις αδυναμίες του φορμαλισμού.

Θα χρησιμοποιήσουμε πρώτους αριθμούς και εκθέτες, αφού αυτή η κωδικοποίηση είναι πολύ απλή. Να σημειώσουμε ότι το σύνολο των πρώτων αριθμών είναι PR. Για παράδειγμα για να κωδικοποιήσουμε την k-αδα $\langle x_0, ..., x_k \rangle$ ο πιο απλός τρόπος είναι να χρησιμοποιήσουμε τον αριθμό $p_0^{x_0} ... p_k^{x_k}$ μάλιστα για να είναι μοναδική η αποκωδικοποίηση ενός αριθμού μπορούμε να χρησιμοποιήσουμε έναν ακόμα πρώτο, με εκθέτη το μήκος της k-άδας.

Ορισμός 2.5 *(Αριθμοί Gödel) Κάθε σύστημα συναρτήσεων η ακολουθία f_n που παράγει μια μερικώς αναδρομική συνάρτηση (ομοιώς και για οδηγίες προγραμμάτων $\mathcal{P}$ μηχανών), μπορούμε να το κωδικοποιήσουμε με ένα μοναδικό Αριθμό Gödel, στον οποίο από δω και πέρα θα αναφερόμαστε ως πρόγραμμα e. Μάλιστα η συνάρτηση κωδικοποίησης είναι ένα-προς-ένα.*

Στη συνέχεια θα πρέπει να αναπτύξουμε ένα σύστημα αποκωδικοποίησης, που δίνεται από τις ακόλουθες συναρτήσεις και κατηγορήματα, οι οποίες είναι όλες PR.

$$(x)_n = exp(x, p_n)$$
$$len(x) = (x)_0$$
$$Seq(x) \Leftrightarrow (\forall n \leqslant x)[n \rangle 0 \wedge (x)_n \neq 0 \rightarrow n \leqslant len(x)].$$

Το $len(x)$ λέγεται μήκος του x, το $(x)_n$ n-ιοστό στοιχείο του x. Αν η $Seq(x)$ ισχύει, τοτε λέμε ότι το x είναι ακολουθιακός αριθμός. Σε αυτή την περίπτωση,

$$x = \langle (x)_1, ..., x_{len(x)} \rangle.$$

Χρειαζόμαστε επίσης την πράξη της παράθεσης:

$$\langle x_1, ..., x_n \rangle * \langle y_1, ..., y_m \rangle = \langle x_1, ..., x_n, y_1, ..., y_m \rangle$$

αν ισχύει η $Seq(x) \wedge Seq(y)$ και 0 αλλιώς.

Τέλος ορίζουμε την έννοια του αρχικού τμήματος, ως εξής:

$$x \sqsubseteq y \leftrightarrow Seq(x) \wedge Seq(y) \wedge (\exists u \langle y)(Seq(u) \wedge x * u = y)$$
$$x \sqsubset y \leftrightarrow x \sqsubseteq y \wedge x \neq y.$$

2.2.2 Θεώρημα Κανονικής Μορφής

Θα δούμε εδώ το πρώτο και μόνο πλήρες παράδειγμα απαρίθμησης, προσπαθώντας να ανάγουμε τις αναδρομικές συναρτήσεις σε μία "κανονική μορφή". Μπορούμε να κατασκευάσουμε κάθε ολική ή μερική αναδρομική συνάρτηση χρησιμοποιώντας τις βασικές συναρτήσεις, τη σύνθεση, την πρωτογενή αναδρομή και την μ-ελαχιστοποίηση, χρησιμοποιώντας το τελευταίο σχήμα το πολύ μία φορά.

Θεώρημα 2.5 *(Θεώρημα Κανονικής Μορφής, Kleene, 1936)*
Υπάρχει πρωτογενώς αναδρομική $\mathcal{U}$ και (για κάθε $k \geqslant 1$) πρωτογενώς αναδρομικά κατηγορήματα $\mathcal{T}_n$ τέτοιες ώστε για κάθε αναδρομική f, k μεταβλητών, υπάρχει αριθμός e (που λέγεται κωδικός της f) για τον οποίο ισχύουν τα ακόλουθα:

- $\forall x_1 ... \forall x_k \exists y \mathcal{T}_n(e, x_1, ..., x_n, y)$

- $f(x_1), ..., f(x_n) = \mathcal{U}(\mu y \mathcal{T}_n(e, x_1, ..., x_n, y))$.

Απόδειξη. Η κεντρική ιδέα της απόδειξης αυτού του θεωρήματος, η οποία μάλιστα μπορεί να επεκταθεί και σε κάθε άλλο μοντέλο υπολογισμού, είναι η εξής. Αρχικά μπορούμε να ορίσουμε την περιγραφή μιας αναδρομικής συνάρτησης f και να την κωδικοποιήσουμε μέσω ενός φυσικού. Στη συνέχεια θα πρέπει να εισάγουμε μια έννοια ακολουθίας υπολογισμού, την οποία επίσης θα κωδικοποιήσουμε. Η κωδικοποίηση θα πρέπει να είναι τέτοια ώστε, δεδομένου ενός κωδικού συνάρτησης και μιας τιμής εισόδου, να μπορούμε να προσομοιώσουμε τον υπολογισμό της συνάρτησης αποτελεσματικά.

Στη συνέχεια το κατηγόρημα $\mathcal{T}$ γνωστή και ως "κατηγόρημα Kleene" ελέγχει, αν είναι δόκιμη η κωδικοποίηση, αν κάθε στοιχείο της εισόδου αντιστοιχεί σε κάποιο βήμα υπολογισμού και τέλος αν δεδομένης της εισόδου, η υπολογιστική διαδικασία τερματίζεται κάποια στιγμή. Αν η απάντηση του $\mathcal{T}$ είναι σε όλα ναι (αλήθεια), τότε η συνάρτηση $\mathcal{U}$ αναλαμβάνει να εξάγει την τιμή του υπολογισμού.

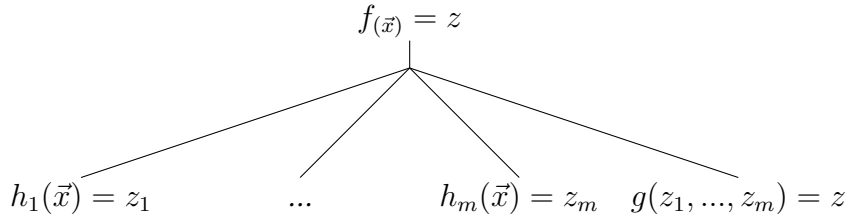
Ας προχωρήσουμε τώρα στην απόδειξη, ξεκινώντας απ' το να εισάγουμε ένα δόκιμο σύστημα κωδικοποίησης.
Η κωδικοποίηση, γίνεται με βάση την επαγωγική διαδικασία που γεννά τις αναδρομικές συναρτήσεις. Εδώ θα χρησιμοποιήσουμε την εξής αντιστοίχηση (μία από τις πολλές που μπορούν να πραγματοποιηθούν).

- $\langle 0 \rangle$ στο $\mathcal{O}$
- $\langle 1 \rangle$ στο $\mathcal{S}$
- $\langle 2, n, i \rangle$ στο $\mathcal{I}_i^n$
- $\langle 3, b_1, ..., b_n, a \rangle$ στο $g(h_1(\vec{x}), ..., h_n(\vec{x}))$, όπου b_i και a είναι οι κωδικοί των h_i και g αντίστοιχα.
- $\langle 4, a, b \rangle$ στο $f(\vec{x}, y)$ ορισμένο με πρωτογενή αναδρομή από τις g, h όπου a και b οι αντίστοιχοι κωδικοί.
- $\langle 5, a \rangle$ στο σχήμα μ-ελαχιστοποίησης $f(\vec{x}) = \mu y(g(\vec{x}, y) = 0)$ και το a είναι ο κωδικός της g.

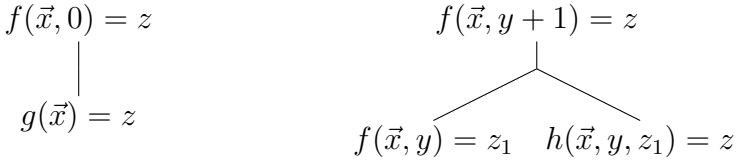
Γενικά υπάρχουν πολλοί τρόποι για να κατασκευάσουμε την ίδια αναδρομική συνάρτηση, έτσι είναι λογικό κάθε συνάρτηση να έχει πολλούς κωδικούς. Όπως επίσης και πολλοί αριθμοί δεν αποτελούν κωδικό αναδρομικής συνάρτησης είτε επειδή δεν έχουν τη σωστή "φόρμα" είτε επειδή τα επιμέρους στοιχεία τους δεν είναι δόκιμοι κωδικοί συναρτήσεων. Όπως θα δούμε αργότερα δεν υπάρχει τρόπος να ελέγξουμε αν ένας αριθμός είναι κωδικός αναδρομικής συνάρτησης.

Ένας φυσικός τρόπος να αναπαραστήσουμε τα βήματα του υπολογισμού, είναι μέσω δέντρων υπολογισμού. Κάθε κόμβος του δέντρου μας λέει πως μπορούμε βρούμε επαγώγικά κάθε τιμή που χρειάζεται στον υπολογισμό.

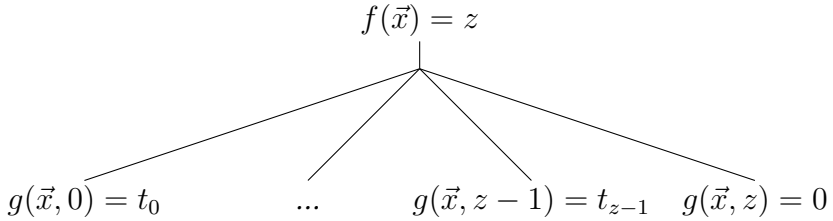
- κόμβοι χωρίς προγόνους:
  $f(x) = 0$ αν $f = \mathcal{O}$
  $f(x) = x + 1$ αν $f = \mathcal{S}$
  $f(x_1, ..., x_n) = x_i$ αν $f = \mathcal{I}_i^n$
- σύνθεση:
  Αν $f(\vec{x}) = g(h_1(\vec{x}), ..., h_m(\vec{x}))$ τότε ο κόμβος $f(\vec{x}) = z$ έχει m+1 προγόνους.



- πρωτογενής αναδρομή:
  Αν η f ορίζεται με πρωτογενή αναδρομή από τις g και h τότε υπάρχουν δύο περιπτώσης με ένα ή δύο προγόνους αντίστοιχα.

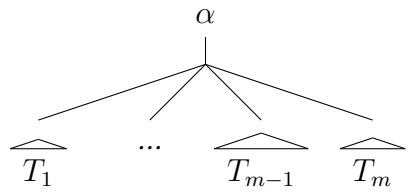


- μ-ελαχιστοποίηση:
  Προφανώς στην περίπτωση της μ-ελαχιστοποίησης $f(\vec{x}) = \mu y(g(\vec{x}, y))$, δεν υπάρχει δεδομένος αριθμός προπατόρων της $f(\vec{x}) = y$.



  όπου τα διάφορα $t_i \neq 0$.

- Κωδικοποίηση των δέντρων υπολογισμού. Η διαδικασία θα πραγματοποιηθεί επαγωγικά από τους κόμβους προς το σύνολο του δέντρου. Αρχικά κωδικοποιούμε τους κόμβους, με κωδικούς της μορφής $\langle e, \langle x_1, ..., x_n\rangle, z\rangle$ που προκύπτους απ την εξής τριάδα: τον κωδικό e της f, την είσοδο και την έξοδο. Κάθε δέντρο T αποτελείται από ένα κόμβο-ρίζα ή βεντάλια α πεπερασμένο αριθμό υποδέντρων, με ρίζες τους προγόνους του α $T_1, ..., T_m$. Άρα μπορούμε επαγωγικά να ορίσουμε τον κωδικό του δέντρου T, έστω $[T]$ και κατά συνέπεια του υπολογισμού ως $\langle [a], [T_1], ..., [T_m]\rangle$.



- Τέλος, μετατροπή σε κατηγόρημα $\mathcal{T}$ της ιδιότητας, ότι το y είναι δόκιμος κωδικός ενός δέντρου υπολογισμού.
  Η $\mathcal{T}$ θα είναι πρωτογενώς αναδρομικό κατηγόρημα. Η τυποποίηση της διαδικασίας της μετατροπής είναι αρκετά μακρά και δύσκολη, ωστόσο όχι σε εννοιολογικό επίπεδο, αφού η κεντρική ιδέα αυτού του τμήματος της απόδειξης συνίσταται αποκλειστικά στην αποκωδικοποίηση του y και τον έλεγχο αν όντως είναι δέντρο υπολογισμού. Για την αποκωδικοποιήση θα χρησιμοποιήσουμε τα εργαλεία που είχαμε εισάγει προηγουμένως.
  Για να διευκολύνουμε την ανάγνωση θα αντικαταστήσουμε τις αλλεπάλληλες παρενθέσεις με κόματα. Δηλαδή θα γράφουμε $a_{i,j,k}$ αντί για $(((a_i)_j)_k))$. Έτσι για $y = \langle v, T_1, ..., T_n\rangle\rangle$ έχουμε:

$$(y)_1 = \langle e, \langle x_1, ..., x_n\rangle$$

$$(y)_{1,1} = \text{ανάλογα με το } e.$$
$$(y)_{1,2} = \langle x_1, ..., xn \rangle$$
$$(y)_{1,3} = z$$
$$(y)_{i+1} = T_i$$
$$(y)_{i+1,1} = \text{αριθμός των κόμβων του } T_i.$$

Αρχικά έστω,

$$A(y) \Leftrightarrow Seq(y) \wedge Seq((y)_1) \wedge len((y)_1) = 3 \wedge Seq((y)_{1,1}) \wedge Seq((y)_{1,2}).$$

Αυτά αποτυπώνουν τις πιο τετριμμένες ιδιότητες του y. Στη συνέχεια θα εξετάσουμε τις περιπτώσεις που αντιστοιχούν στα σχήματα που εισήχθησαν νωρίτερα.
Για τις αρχικές συναρτήσεις έχουμε τρεις πιθανές περιπτώσεις για $v = (y)_1$:

$$\langle\langle 0\rangle, \langle x\rangle, 0\rangle$$
$$\langle\langle 1\rangle, \langle x\rangle, x+1\rangle$$
$$\langle\langle 2, n, i\rangle, \langle x_1, ..., x_n\rangle, x_i\rangle.$$

Στη συνέχεια ορίζουμε:

$$B(y) \Leftrightarrow len(y) = 1 \wedge \{[(y)_{1,1} = \langle 0\rangle \wedge len((y)_{1,2}) = 1 \wedge (y)_{1,3} = 0] \vee$$
$$[(y)_{1,1} = \langle 1\rangle \wedge len((y)_{1,2}) = 1 \wedge y_{1,3} = (y)_{1,2,1} + 1] \vee$$
$$[len((y)_{1,1} = 3 \wedge (y)_{1,1,1} = 2 \wedge (y)_{1,1,2} = len((y)_{1,2}) \wedge$$
$$1 \leqslant (y)_{1,1,3} \leqslant (y)_{1,1,2} \wedge (y)_{1,3} = ((y)_{1,2})_{(y)_{1,1,3}}]\}.$$

Για τη σύνθεση έστω:

$$C(y) = len(y)_{1,1} \geqslant 3 \wedge (y)_{1,1,1} = 3 \wedge len(y) = len((y)_{1,1}) \wedge$$
$$(\forall i)_2 \leq i \leq ln(y)[(y)_{i,1,1} = (y)_{1,1,i} \wedge (y)_{i,1,2} = (y)_{1,2}] \wedge$$
$$(y)_{len(y),1,1} = (y)_{1,1,len(y)} \wedge (y)_{i,1,2} = (y)_{1,2}] \wedge$$
$$(y)_{len(y),1,2} = \langle (y)_{2,1,3}, ...(y)_{len(y)-1,1,3}\rangle.$$

Για την πρωτογενή αναδρομή υπάρχουν δυο περιπτώσεις:

$$D(y) \Leftrightarrow len((y)_{1,1}) = 3 \wedge (y)_{1,1,1} = 4 \wedge$$
$$\{[(y)_{1,2,len(y)_{1,2}} = 0 \wedge len(y) = 2 \wedge (y)_{2,1,1} = (y)_{1,1,2} \wedge$$
$$(y)_{2,1,2} * \langle 0\rangle = (y)_{1,2} \wedge (y)_{2,1,3} = (y)_{1,3}] \vee$$
$$[(y)_{1,2,len((y)_{1,2})} > 0 \wedge len(y) = 3 \wedge$$
$$(y)_{2,1,1} = (y)_{1,1} \wedge len((y)_{2,1,2}) = len((y)_{1,2}) \wedge$$
$$(\forall i)_{1 \leq i < len((y)_1,2)} + 1 = (y)_{1,2,i}) \wedge$$
$$(y)_{2,1,2,len((y)_{1,2})} + 1 = (y)_{1,2,len((y)_{1,2})} \wedge$$
$$(y)_{3,1,1} = \langle (y)_{1,3,3}\rangle \wedge (y)_{3,1,3} = (y)_{1,3} \wedge$$
$$(y)_{3,1,2} = (y)_{2,1,2} * \langle (y)_{2,1,3}\rangle]\}.$$

Τέλος για τη μ-αναδρομή έχουμε:

$$E(y) \Leftrightarrow len((y)_{1,1}) = 2 \wedge (y)_{1,1,1} = 5 \wedge$$
$$len(y) \geqslant 2 \wedge (y)_{1,3} = len(y) - 2 \wedge$$
$$(\forall i)_{2 \leq i \leq len(y)}[(y)_{i,1,1} = (y)_{1,1,2} \wedge$$
$$(y)_{i,1,2} = (y)_{1,2} * \langle i - 2 \rangle] \wedge$$
$$(\forall i)_{2 \leq i \leq len(y)}[(y)_{i,1,3} \neq 0] \wedge (y)_{len(y)_{1,3}} = 0.$$

Εξαντήσαμε έτσι όλες τις περιπτώσεις κι έτσι μπορούμε τωρα να ορίσουμε επαγωγικά το $\mathcal{T}$

$$\mathcal{T}(y) \Leftrightarrow A(y) \wedge [B(y) \vee C(y) \vee D(y) \vee E(y)] \wedge$$
$$[len(y) > 1 \rightarrow (\forall i)_{2 \leq i \leq len(y)} \mathcal{T}((y)_i)].$$

Και το κατηγόρημα Kleene ανήκει στην PR.

Ορίζουμε τα $\mathcal{T}_n$ και $\mathcal{U}$.
Πλησιάζουμε στο τέλος της απόδειξης. Έτσι, για κάθε $n \leqslant 1$ έχουμε:

$$\mathcal{T}_n(e, \vec{x}, y) \Leftrightarrow \mathcal{T}(y) \wedge (y)_{1,1} = e \wedge (y)_{1,2} = \langle x_1, ..., x_n \rangle$$

και

$$\mathcal{U}(y) = (y)_{1,3}.$$

Προφανώς και αυτά τα κατηγορήματα είναι πρωτογενώς αναδρομικά.

Έστω τώρα f αναδρομική n-μελής συνάρτηση με κωδικό e. Αφού η f είναι ολική για κάθε n-άδα $\vec{x}$ θα υπάρχει ένα δέντρο υπολογισμού για την $f(\vec{x}$ σε αντιστοιχία με την διαδικασία υπολογισμού που κωδικοποιείται απ το e. Αυτό αποτυπώνεται τυπικά ως:

$$\forall x_1 ... \forall x_n \exists y \mathcal{T}_n(e, x_1, ..., x_n, y).$$

■

Το Θεώρημα Κανονικής Μορφής περιγράφει κατά κάποιον τρόπο τη διαδικασία κατά την οποία, δίνοντας ένα όρισμα-είσοδο, σε ένα υπολογιστικό πρόγραμμα, αν αυτό τερματίζει, επιστρέφεται η τιμή εξόδου. Είναι σημαντικό οτι δεν μας επηρρεάζει ο τερματισμός του προγράμματος, καθώς το ΘΚΜ επεκτείνεται και στις μερικές αναδρομικές συναρτήσεις. Επίσης μας παρέχει μια μηχανιστική απαρίθμηση $\{e : f_e$ αναδρομική $\}$ όλων των (μερικώς) αναδρομικών συναρτήσεων η οποία οδηγεί μάλιστα στον καθιερωμένο συμβολισμό για τις αναδρομικές συναρτήσεις: $\langle e, \vec{x} \rangle$.

Πρόταση 2.6 *Μία μηχανιστική απαρίθμηση (που εξαρτάται από τα συμφραζόμενα) είναι μια λίστα (μερικώς) αναδρομικών συναρτήσεων $f_0, f_1, f_2, ...$ τέτοια ώστε το κατηγόρημα $\langle e, x \rangle \mapsto f_e(x)$ είναι μερικώς αναδρομικό.*

2.2.3 Καθολικές αναδρομικές συναρτήσεις και σταθερά σημεία

Επακόλουθα της θεμελιώδους ιδέας της απαρίθμησης του συστήματος μας και της επιτυχούς απαρίθμησης των αναδρομικών συναρτήσεων, είναι το θεώρημα απαρίθμησης, και το θεώρημα παραμετροποίησης (γνωστό και ως θεώρημα s-m-n), καθώς και τα θεωρήματα σταθερού σημείου (η θεωρήματα αναδρομής) του Kleene τα οποία ωστόσο απλά θα παραθέσουμε, αφού πέραν του ότι μπορουν να θεωρηθούν επακόλουθα του ΘΜΝ, το ενδιαφέρον γεγονός που ενσωματώνουν είναι η ιδέα της ύπαρξης μιας καθολικής συνάρτησης (πιο συχνά συναντάται ο όρος καθολική μηχανή) ικανής να υπολογίσει οποιαδήποτε μερικώς αναδρομική συνάρτηση.

Όπως είδαμε εκτεταμένα στο ΘΚΜ, δεδομένης μιας αποδεκτής κωδικοποίησης των βασικών σχημάτων όλες οι συναρτήσεις μπορούν να γραφτούν σαν λέξεις από ένα αλφάβητο που περιέχει τους βασικούς κωδικούς. Και δύο βασικά χαρακτηριστικά που βοηθούν στην αποκωδικοποίηση των λέξεων είναι η πλειάδα της εισόδου, και ο αριθμός των βασικών συναρτήσεων που χρησιμοποιήθηκαν για να παράξουν την τελική συνάρτηση (όπως είδαμε σχηματικά μέσω δέντρων υπολογισμού). Έτσι έχουμε το εξής αποτέλεσμα.

Θεώρημα 2.7 *Θεώρημα S-m-n (παραμετροποίησης): Έστω m, n, τότε υπάρχει πρωτογενώς αναδρομική $s_n^m : \mathbb{N}^{m+1} \to \mathbb{N}$, τέτοια ώστε για κάθε $e, x_1, ..., x_m, y_1, ..., y_n$:*

$$f_e^{m+n}(x_1, ..., x_m, y_1, ..., y_n) \simeq f_{s_n^m}^{(n)}(y_1, ..., y_n).$$

Θεώρημα 2.8 *Θεώρημα Απαρίθμησης: Για κάθε $k \geqslant 1$ η $k+1$-μελής συνάρτηση*

$$f_e^{(k)}(x_1, ..., x_k)$$

είναι μερικώς αναδρομική. Σημειώνουμε ότι το όρισμα περιλαμβάνει τα $e, \vec{x}$.

Το επόμενο αποτέλεσμα υπονοεί την ύπαρξη "καθολικού υπολογιστή".

Πόρισμα 2.9 *Υπάρχει μερικώς αναδρομική $\psi : \mathbb{N}^2 \to \mathbb{N}$ τέτοια ώστε για όλα τα d, e και $\vec{x} \in \mathbb{N}^d$ έχουμε:*

$$f_e^d(\vec{x}) \simeq \psi(e, \langle \vec{x} \rangle).$$

Ορισμός 2.6 *Μια απαρίθμηση ονομάζεται αποδεκτή, αν για κάθε n, υπάρχουν ολικώς αναδρομικές h και g, τέτοιες ώστε:*

$$\psi_e^n \simeq f_{h(e)}^n \text{ και } f_e^n \simeq \psi_{g(e)}^n.$$

Πρόταση 2.10 *Μία απαρίθμηση είναι αποδεκτή ανν ικανοποιεί και την απαρίθμηση και την παραμετροποίηση.*

Απόδειξη.
$\Rightarrow$
Έστω ψ η οποία ικανοποιεί και τα δύο θεωρήματα, θα δείξουμε ότι η ψ είναι αποδεκτή.
Η ψ_e^n από την ιδιότητα της απαρίθμησης, είναι μια αναδρομική συνάρτηση με $n+1$ μεταβλητές. Αφού ένα σύστημα δεικτών e πρέπει να όλες τις μερικώς αναδρομικές συναρτήσεις, υπάρχει ένας δείκτης α σε σχέση με μια f, και έτσι

$$\psi_e^n(\vec{x}) \simeq f_\alpha^{(n+1)}(e, \vec{x}) \simeq f^n_{s_1^1(a,e)}(\vec{x}.$$

Έτσι θέτοντας $h(e) = s_1^1(\alpha, e)$, έχουμε $\psi_e^n \simeq f^n_{h(e)}$.
Η g του ορισμού παράγεται, χρησιμοποιώντας το θεώρημα απαρίθμησης για την f και το s-m-n για την ψ.
$\Leftarrow$
Για την άλλη κατεύθυνση, υποθέτουμε ότι η ψ είναι αποδεκτή.
Για την απαρίθμηση θα δείξουμε ότι,

$$\psi_e^n(\vec{x}) \simeq f^n_{h(e)}(\vec{x}).$$

Πράγματι, η $f^n_{h(e)}$ είναι μερικώς αναδρομική ως συνάρτηση των $e, \vec{x}$ (από το θεώρημα απαρίθμησης), και έτσι μπορεί να αντιστοιχεί σ'αυτή ένας δείκτης α του συστήματος ψ.
Για την παραμετροποίηση θα χρησιμοποιήσουμε την $\psi_e^{(n+m)}(\vec{x}, \vec{y})$, η οποία,αφού αποδείξαμε την ιδιότητα απαρίθμησης, είναι μια μερικώς αναδρομική συνάρτηση $m+n+1$ μεταβλητών μαζί με το e και έτσι υπάρχει δείκτης α σε σχέση με την f, τέτοιος ώστε:

$$\psi_e^{(n+m)}(\vec{x}, \vec{y}) \simeq f_\alpha^{(m+n+1)}(e, \vec{x}, \vec{y}) \simeq f_{s^m_{(n+1)}(a,e,\vec{x})}(\vec{y}).$$

Αφού η g μας επιτρέπει να γυρίσουμε στο σύστημα απαρίθμησης ψ, αν πάρουμε

$$s(e, \vec{x}) \simeq g(s^m_{(n+1)}(a, e, \vec{x}))$$

τότε:

$$\psi_e^{(n+m)}(\vec{x}, \vec{y}) \simeq \psi^m_{s(e,\vec{x})}(\vec{y}).$$

■

Σημείωση 2.4 *Αξίζει να παρατηρήσουμε, ότι η απαρίθμηση και η παραμετροποίηση εκτός των άλλων υπονοούν μια παλινδρόμηση μεταξύ των χώρων συναρτήσεων,*

$$\mathbb{N}^n \times \mathbb{N}^m \to \mathbb{N} \text{ και } \mathbb{N}^n \to (\mathbb{N}^m \to \mathbb{N}).$$

Πρόταση 2.11 *(Padding Lemma, Rogers 1958). Σε ένα αποδεκτό σύστημα απαρίθμησης, μπορούμε να παράξουμε άπειρους δείκτες για την ίδια συνάρτηση.*

Για τον επόμενο ορισμό θα δούμε για πρώτη φορά την έννοια της αναγωγής που θα αναπτύξουμε περισσότερο στα επόμενα κεφάλαια.

Ορισμός 2.7 *Καθολική αναδρομική συνάρτηση: Έστω $g(\vec{x})$ και $\psi(\vec{x})$ k-μελείς, μερικώς αναδρομικές συναρτήσεις. Λέμε ότι η ψ μπορεί να αναχθεί στην g, αν υπάρχει k-μελής ολικώς αναδρομική $h(\vec{x})$ τέτοια ώστε $\psi(\vec{x}) \simeq g(h(\vec{x}))\ \forall \vec{x} \in \mathbb{N}^k$. Λέμε ότι η g είναι καθολική αν κάθε k-μελής μερικώς αναδρομική συνάρτηση μπορεί να αναχθεί στην g.*

Στη συνέχεια θα δούμε τα τελευταία ενδιαφέροντα αποτελέσματα που προκύπτουν απο την προηγούμενη διαδικασία, θεωρήματα τα οποία έχουν μείνει γνωστά ως Θεώρημα Αναδρομής (ή θεώρημα σταθερού σημείου).
Σταθερό σημείο μιας συνάρτησης είναι μια τιμή που παραμένει αναλλοίωτη, αφού εφαρμόσουμε τη συνάρτηση.

Θεώρημα 2.12 *Έστω $f : \mathbb{N}^{d+1} \to \mathbb{N}$ μερικώς αναδρομική. Τότε e_0 τέτοιο ώστε για κάθε $\vec{x} \in \mathbb{N}^d$,*

$$f(e_0, \vec{x}) \simeq f^{(d)}_{e_0}(\vec{x}).$$

Απόδειξη. Ορίζουμε μερικώς αναδρομική $g : \mathbb{N}^{d+1} \to \mathbb{N}$ τέτοια ώστε $g(e, x) \simeq f(s^1_d(e, e), \vec{x})$. Παίρνουμε στη συνέχεια α τέτοιο ώστε $g = f^{(d+1)}_a$. Τότε για κάθε $\vec{x} \in \mathbb{N}^d$,

$$f(s^1_d(\alpha, \alpha), \vec{x}) \simeq g(\alpha, \vec{x}) \simeq f^{(d+1)}_a(\alpha, \vec{x}) \simeq f^{(d)}_{s^1_d}(\alpha, \alpha(\vec{x}).$$

Τότε για $e_0 = s^1_d(\alpha, \alpha)$ ισχύει. ■

Θεώρημα 2.13 *Έστω $h : \mathbb{N} \to \mathbb{N}$ αναδρομική. Τότε υπάρχει e_0 τέτοιο ώστε,*

$$f_{e_0} = f_{h(e_0)}.$$

Απόδειξη. Έστω $f : \mathbb{N} \to \mathbb{N}$ μερικώς αναδρομική $f(e, x) \simeq f_{h(e)}(x)$. Από το προηγούμενο θεώρημα έχουμε $e_0 \in \mathbb{N}$ τέτοιο ώστε για κάθε x,

$$f_{e_0}(x) \simeq f(e_0, x) \simeq f_{h(e_0)}(x),$$

και έτσι $f_{e_0} = f_{h(e_0)}$. ■

Αυτά τα θεωρήματα αποτυπώνουν την ικανότητα ενός προγράμματος (ή και κάθε μηχανής) να αναπαράγουν τον εαυτό τους, δίνοντας τους ως εντολή εισόδου την ίδια του την περιγραφή, αποτέλεσμα μη αναμενόμενο αφού διαισθητικά κανείς θα πίστευε ότι χρειάζεται μια μηχανή μεγαλύτερης πολυπλοκότητας για να παράγει μια άλλη μηχανή. Ωστόσο η ουσιαστική τους χρησιμότητα είναι η σημασία τους για την επίλυση πλήθους προβλημάτων στη θεωρία της υπολογισιμότητας. Μία άλλη πτυχή τους είναι η θεμελιώδης ιδιότητα των αναδρομικών (υπό την έννοια της αυτοαναφοράς και της επανάληψης) διαδικασιών, να οδηγούνται εν τέλει σε μια κατάσταση ισορροπίας.

Κάνοντας μια σύνοψη, είδαμε σε αυτό το κεφάλαιο κάποια από τα βασικά στοιχεία της θεωρίας αναδρομής και κυρίως της διαδικασίας της απαρίθμησης, δηλαδή της κατασκευής ενός κατάλληλου συστήματος απαρίθμησης των μαθηματικών οντοτήτων με τις οποίες ασχολούμαστε. Αξίζει να σημειωθεί ότι αυτή η διαδικασία δεν περιορίζεται μόνο στη μελέτη των αναδρομικών συναρτήσεων και έχει τεράστιες δυνατότητες γενίκευσης.

Κεφάλαιο 3

Μη υπολογισιμότητα

> Αν απαιτείται από μια μηχανή να είναι
> αλάνθαστη, δεν μπορεί να είναι και ευφυής.
>
> *Turing, 1947*

Σκοπός του κεφαλαίου αυτού είναι να μελετήσουμε μία κλάση μαθηματικών προβλημάτων που είναι γνωστά ως προβλήματα απόφασης (decision problems). Θα καταλήξουμε στο συμπέρασμα ότι η συντριπτική πλειοψηφία αυτού του τύπου προβλημάτων είναι μη αποκρίσιμα. Θα δούμε πως επιχειρήματα της θεωρίας αναδρομής, μπορούν να χρησιμοποιηθούν για να παράξουν αποτελέσματα μη αποκρισιμότητας.

3.1 Halting Problem

Σημείωση 3.1 *Υπενθυμίζουμε ότι ένα σύνολο είναι αναδρομικό, ανν η χαρακτηριστική του συνάρτηση είναι αναδρομική.*

Ορισμός 3.1 *Έστω $A \subseteq \mathbb{N}$. Καλούμε πρόβλημα απόφασης που προκύπτει από το A το εξής πρόβλημα:*

Δεδομένου n, να αποφασίσουμε αν το n είναι στο A ή όχι.

Το πρόβλημα αυτό λέγεται αποκρίσιμο αν το A είναι αναδρομικό, αλλιώς λέγεται μη αποκρίσιμο.

Σημείωση 3.2 *Συμβολίζουμε το πεδίο σύγκλισης μιας μερικώς αναδρομικής συνάρτησης, βάση του κωδικού της, ως $\mathcal{W}_e$.*

Ας δούμε τώρα το πιο γνώστο μη αποκρίσιμο πρόβλημα.

Θεώρημα 3.1 *Halting Problem (Turing 1936): Το σύνολο $\mathcal{K}_0$ (Halting set) που ορίζεται ως εξής:*

$$\langle e, x\rangle \in \mathcal{K}_0 \Leftrightarrow x \in \mathcal{W}_e \Leftrightarrow f_e(x)\downarrow$$

δεν είναι αναδρομικό.

Απόδειξη. Υποθέτουμε το αντίθετο. Τότε το σύνολο $\{e : f_e(e)\downarrow\}$ είναι αναδρομικό. Ορίζουμε $h : \mathbb{N} \to \mathbb{N}$ ως εξής:

$$h(e) = \begin{cases} 0 & \text{αν} f_e(e)\downarrow \\ \downarrow & \text{αν} f_e(e)\uparrow \end{cases}$$

,

Η h είναι μερικώς αναδρομική αφού $f(e) = \mu y(y = 0 \wedge f_e(e)\uparrow)$. Έτσι έχουμε e_0 τέτοιο ώστε $h = f_{e_0}$ από το θεώρημα απαρίθμησης. Όμως $h(e_0)\uparrow \Leftrightarrow f_{e_0}(e_0)\downarrow$, δηλαδή

$$f_{e_0}(e_0)\uparrow \Leftrightarrow f_{e_0}(e_0)\downarrow,$$

άτοπο. ■

Πόρισμα 3.2 *Επίσης και το διαγώνιο Halting set $\mathcal{K} = \{e : f_e(e)\downarrow\}$ είναι μη αποκρίσιμο.*

Σημείωση 3.3 *Ουσιαστικά, η απόδειξη του Halting Problem, ειναι μια περίπτωση διαγωνιοποίησης, υπό την έννοια ότι χρησιμοποιούμε ένα επιχείρημα τύπου,*

$$a_{i,j} = \begin{cases} 1 & \alpha\nu\, j \in W_i \\ 0 & \alpha\lambda\lambda\iota\acute{\omega}\varsigma \end{cases}$$

και στη συνέχεια θέτουμε $d(a) = 1 - a$.

Σημείωση 3.4 *Μία πιο διαισθητική προσέγγιση της απόδειξης του Halting Problem είναι η εξής: Υποθέτουμε ότι υπάρχει πρόγραμμα που μπορεί να δει τον "κώδικα" οποιουδήποτε άλλου προγράμματος και να αποφασίσει αν το άλλο πρόγραμμα θα σταματήσει καποια στιγμή να τρέχει. Ωστόσο αν υπήρχε όντως τέτοιο πρόγραμμα, θα μπορούσαμε να το τρέξουμε σε μία εκδοχή του εαυτού του που τερματίζει αν το άλλο πρόγραμμα δεν σταματά ποτέ, ενώ συνεχίζει να τρέχει για πάντα αν το άλλο πρόγραμμα τερματίζει.*

Εξ'ορισμού το $\mathbb{N}$ είναι αριθμήσιμο και οποιοδήποτε σύνολο μπορεί να χρησιμοποιήσει ως δείκτες των στοιχείων του το $\mathbb{N}$ είναι επίσης αριθμήσιμο. Ωστόσο πολλά σύνολα αριθμοθεωρητικών συναρτήσεων όπως τα $\mathbb{N}^{\mathbb{N}} = \{f : \mathbb{N} \to \mathbb{N}\}$ ή το $\{0,1\}^{\mathbb{N}} = \{f : \mathbb{N} \to \{0,1\}\}$ είναι υπεραριθμήσιμα. Έτσι μπορούμε να υποθέσουμε ασφαλώς βάσει των πληθάριθμων αυτών των συνόλων, ότι σχεδόν όλες οι αναδρομικές συναρτήσεις, όπου "σχεδόν όλες" εννοούμε, όλες πέραν ενός αριθμήσιμου συνόλου δεν είναι αναδρομικές, αλλα και κατα συνέπεια σχεδόν όλα τα προβλήματα είναι μη αποκρίσιμα.

3.2 Άλλα γνωστά μη επιλύσιμα προβλήματα

Θα δούμε κάποια προβλήματα που αποδείχθηκε ότι είναι μη αποκρίσιμα και προέρχονται από διαφορετικούς τομείς των μαθηματικών. Δε θα μας απασχολήσουν κατ ουσίαν παρα μόνο για ιστορικούς λόγους και για να καταδείξουμε τη σύνδεση της θεωρίας υπολογισμού με διάφορα άλλα αντικείμενα.

3.2.1 Το 10ο Πρόβλημα του Hilbert

Στη διάσημη ομιλία του το 1900 ο Hilbert εξέδωσε τη λίστα με τα 23 προβλήματα που θα διαμόρφωναν τον κορμό στα μαθηματικά του 20ού αιώνα. Συμπεριλήφθηκαν αρκετά που αφορούσαν προβλήματα λογικής και θεμελιακά ζητήματα των μαθηματικών, όπως ας πούμε το πρώτο και το δεύτερο που αφορούσαν την υπόθεση του συνεχούς και το ζήτημα της συνέπειας της Αριθμητικής. Το δέκατο πρόβλημα ήταν το εξής:
Έστω μια Διοφαντική εξίσωση $f(x_1, ..., xk) = 0$, υπάρχει λύση της εξίσωσης στους ακέραιους $x_1, ..., x_k$; Το πρόβλημα είναι να κατασκευαστεί μια αλγοριθμική διαδικασία που να απαντά στην ερώτηση δεδομένου του πολυωνύμου f. (Εξ'ορισμού, μια Διοφαντική εξίσωση είναι ένα πολυώνυμο με ακέραιους συντελεστές.)

Το σύνολο των Διοφαντικών εξισώσεων ορίζεται ως,

$$\{\vec{a} \in \mathbb{N}^m : \exists \vec{x}(f(\vec{x}, \vec{a} = 0)\}.$$

Έτσι το αρχικό πρόβλημα, αντιμετωπίζεται ως ένα πρόβλημα απόφασης.

Θεώρημα 3.3 *(Matiyasevich,Davis,Robinson,Putnam-MRDP): Το 10ο Πρόβλημα του Hilbert είναι μη επιλύσιμο.*

3.2.2 Το Πρόβλημα των Λέξεων

Έστω ένα πεπερασμένο σύνολο από σύμβολα $A = \{a_1, ..., a_n\}$. Χρησιμοποιούμε τον όρο σχέση, με την έννοια που έχει στην θεωρία ομάδων, δηλαδή ως μια εξίσωση πάνω σε αυτό το αλφάβητο. Έτσι μια τέτοια σχέση γράφεται στη μορφή $W = 1$, όπου W λέξη, δηλαδή η παράθεση κάποιων συμβόλων του αλφαβήτου. Έστω και ένα πεπερασμένο σύνολο σχέσεων $R = \{W_1 = 1, ..., W_m = 1\}$, τότε υπάρχει μοναδική μέγιστη ομάδα $G = \langle A|R\rangle$ που επάγεται από το A και ικανοποιεί τις R.

Παράδειγμα 3.1 *Ας δούμε τους γεννήτορες $A = \langle a, b\rangle$και τις σχέσεις $R = \langle ab = ba, a^2 = 1, b^3 = 1\rangle$.*
Μπορούμε εύκολα να δούμε ότι η σχέση $ab = ba$ μπορεί να γραφεί και ως $aba^{-1}b^{-1} = 1$. Μπορούμε επομένως εν γένει να χρησιμοποιούμε σχέσεις για να απλοποιήσουμε λέξεις του αλφαβήτου μας. Για παράδειγμα

$$aba^2ba^{-1}b^{-1}a^{-1}b^2a^2ba^{-1} = a^2b^4 = b.$$

Η ομάδα $G = \langle A|R\rangle$ που ορίζεται από αυτό το ζεύγος γεννητόρων και κατηγόρημαεων είναι ισομορφική με το $C_2 \times C_3$ με στοιχεία τα $\langle 1, a, b, b^2, ab, ab^2\rangle$, όπου C_n είναι ο n-κύκλος.

Σημείωση 3.5 *Τέτοιες πεπερασμένα παραγόμενες ομάδες παρουσιάζονται συχνά στην αλγεβρική τοπολογία και γεωμετρία.*

Έτσι για μια πεπερασμένα παραγόμενη ομάδα $G = \langle A|R\rangle$, το πρόβλημα λέξεων για την G είναι το ακόλουθο:

Έστω μια λέξη W πάνω στο πεπερασμένο σύνολο γεννητόρων A, ισχύει η σχέση $W = 1$;
Το πρόβλημα είναι να κατασκευάσουμε έναν αλγόριθμο που να αποφασίζει αν η $W = 1$ ανήκει στην G.

Θεώρημα 3.4 *(Boone, Novikov, 1950) Μπορούμε να κατασκευάσουμε πεπερασμένα παραγόμενη ομάδα G, τέτοια ώστε το πρόβλημα λέξεων για την G να είναι μη αποκρίσιμο.*

3.2.3 Busy Beaver

Θα δούμε τέλος ένα πρόβλημα που προκύπτει στα πλαίσια της θεωρίας υπολογισμού. Οι αριθμοί Busy Beaver αφορούν το μέτρημα των βημάτων υπολογισμού μιας μηχανής Turing. Συγκεκιμένα, για δεδομένο n, είναι δυνατόν να διατρέξουμε όλες τις MT που έχουν ακριβώς n καταστάσεις. Από αυτές κάποιες θα τερματίζουν και κάποιες θα τρέχουν για πάντα. Μας ενδιαφέρουν μόνο αυτές που τερματίζουν, καθώς και ο μεγαλύτερος αριθμός βημάτων που θα εκτελέσουν πριν σταματήσουν.

Ορισμός 3.2 *Ο n-οστός αριθμός BB, BB(n), είναι ο μέγιστος αριθμός μη κενών συμβόλων που έχει στον τερματισμό η ταινία μιας MT με ακριβώς n καταστάσεις (συμπεριλαμβανομένης της halt), δύο σύμβολα στο αλφάβητο (κενό και |) και αρχικά κενή ταινία.*

Σημείωση 3.6 *Η BB(n) μεγαλώνει πιο γρήγορα από τη συνάρτηση Ackerman, μάλιστα μεγαλώνει πιο γρήγορα από οποιαδήποτε υπολογίσιμη συνάρτηση.*

Θεώρημα 3.5 *Η συνάρτηση Busy Beaver είναι μη υπολογίσιμη. Δηλαδή δεν υπάρχει MT που να δέχεται είσοδο n και να υπολογίζει την BB(n).*

3.3 Αναδρομικώς Απαριθμήσιμα Σύνολα

Έχουμε δει αρκετές φορές τον όρο μερικώς αναδρομική συνάρτηση και μέχρι τώρα δεν υπήρχε λόγος να τις διακρίνουμε ως προς τα αποτελέσματα που τις αφορούν από τις ολικές αναδρομικές συναρτήσεις. Τώρα όμως θα δούμε τον αντίστοιχο όρο για σύνολα και κατηγορήματα που παρουσιάζει μεγάλο ενδιαφέρον.

Είδαμε προηγουμένως κάποια σύνολα-προβλήματα τα οποία έχουν κάποιες ιδιομορφίες. Σύνολα τα οποία παρουσιάζουν μια δομική ασυμμετρία μεταξύ της ιδιότητας μέλους, η οποία μπορεί να εξακριβωθεί με μία αλγοριθμική διαδικασία μέσω πεπερασμένης ποσότητας πληροφοριών και της ιδιότητας όχι μέλους, η οποία για να εξακριβωθεί χρειάζεται άπειρη πληροφορία. Αυτά τα σύνολα είναι κατά κάποιον τρόπο ημι-αναδρομικά.

Ορισμός 3.3 *Ένα σύνολο λέγεται αναδρομικώς απαριθμήσιμο σύνολο (recursively enumerable), για συντομογραφία RE, αν είναι πεδίο ορισμού μιας n-μελούς μερικώς αναδρομικής συνάρτησης και συμβολίζεται ως*

$$\mathcal{W}_e^n \text{ που αντιστοιχεί στην } f_e^n.$$

Σημείωση 3.7 *Τα RE σύνολα είναι ακριβώς τα σύνολα μη αρνητικών ακέραιων λύσεων διοφαντικών εξισώσεων.*

Σημείωση 3.8 *Το Halting Set $\mathcal{K}_0$ όπως και το διαγώνιο Halting Set $\mathcal{K}$ είναι στην RE.*

Πρόταση 3.6 *Έστω $A \subseteq \mathbb{N}^m$, τα ακόλουθα είναι ισοδύναμα:*

(i) *Το A είναι στην κλάση RE.*

(ii) *Το A είναι Σ_1^0.*

(iii) *$A = Im(f)$ για κάποια μερικώς αναδρομική $f : \mathbb{N}^m \to \mathbb{N}$.*

(iv) *$A = \mathcal{W}_e^m = dom(f_e^m)$.*

(v) *$A = \emptyset$ ή $A = f(\mathbb{N})$ για κάποια $f \in$PR.*

(vi) *Υπάρχει PR $S \subseteq \mathbb{N}^{(m+1)}$ τέτοια ώστε για κάθε $\vec{x}$,*

$$x \in A \Leftrightarrow \exists y S(\vec{x}, y).$$

Θεώρημα 3.7 *(Post, 1943) Ένα σύνολο είναι αναδρομικό ανν είναι RE και το συμπλήρωμά του είναι RE.*

Απόδειξη. Αν το A έιναι αναδρομικό τότε και το A και το $\overline{A}$ έιναι RE, αφου οι συναρτήσεις με πεδίο ορισμού A και $\overline{A}$ είναι οι

$$f(x) \simeq \begin{cases} 1 & \alpha\nu \chi_A(x) = 1 \\ \uparrow & \alpha\lambda\lambda\iota\acute{\omega}\varsigma \end{cases}.$$

$$\psi(x) \simeq \begin{cases} 0 & \alpha\nu \chi_{\overline{A}}(x) = 0 \\ \uparrow & \alpha\lambda\lambda\iota\acute{\omega}\varsigma \end{cases}.$$

Για την αντίστροφη κατεύθυνση, υποθέτουμε ότι τόσο το A όσο και το $\overline{A}$ είναι RE, μη κενά, αφού αν το ένα είναι το $\emptyset$ το άλλο είναι το $\mathbb{N}$ οπότε είναι αναδρομικά.
Τότε υπάρχουν αναδρομικά κατηγορήματα R και Q τέτοιες ώστε:

$$x \in A \Leftarrow \exists y R(x, y)$$
$$x \in \overline{A} \Leftrightarrow \exists y Q(x, y).$$

και αφού ισχύει η

$$\forall x \exists y((R(x,y) \vee Q(x,y))$$

η συνάρτηση

$$f(x) = \mu y(R(x,y) \vee Q(x,y))$$

είναι αναδρομική και ακριβώς ένα από τα $R(x, f(x))$ και $Q(x, f(x))$ ισχύει. Έτσι το A είναι αναδρομικό, αφού

$$\chi_A(x) = \begin{cases} 1 & \text{αν } R(x, f(x)) \\ 0 & \text{αλλιώς} \end{cases}.$$

■

3.4 Αναγωγή many-one

Στην προηγούμενη ενότητα, είδαμε κάποια προβλήματα τα οποία είναι μη επιλύσιμα, όμως συχνά μια απόδειξη μη επιλυσιμότητας δεν είναι τετριμμένη. Έτσι σε αυτή την ενότητα θα εισάγουμε την κύρια μέθοδο απόδειξης ότι ένα πρόβλημα δεν έχει λύση, την αναγωγή.

Οι αναγωγές είναι ένας τρόπος να μετατρέπουμε ένα πρόβλημα σε ένα άλλο πρόβλημα με τέτοιο τρόπο, ώστε μια λύση για το δεύτερο πρόβλημα να μπορεί να χρησιμοποιηθεί για να λύσει το πρώτο πρόβλημα. Η αναγωγιμότητα (reducibility) περιστρέφεται πάντα γύρω απο δύο σύνολα A και B. Αν ένα πρόβλημα A μπορεί να αναχθεί (reduces to) στο B, μπορούμε να χρησιμοποιήσουμε μια λύση για το B για να λύσουμε το A. Αντίστροφα, αν ένα μη επιλύσιμο πρόβλημα A ανάγεται σε ενα πρόβλημα B, τότε και το B είναι μη επιλύσιμο.Για παράδειγμα το πρόβλημα εύρεσης ενός εμβαδού ανάγεται σε πρόβλημα μέτρησης δύο διαστάσεων, η το πρόβλημα επίλυσης ενός συστήματος εξισώσεων ανάγεται σε πρόβλημα αντιστροφής ενός πίνακα.

Οι αναγωγές είναι ένας τρόπος να ταξινομούμε προβλήματα βάσει της αποκρισιμότητας τους και επίσης παίζει σημαντικό ρόλο στη θεωρία πολυπλοκότητας, τόσο στην πολυπλοκότητα που αφορά περιορισμένους υπολογιστικούς πόρους, όσο και στην ιεράρχηση της πολυπλοκότητας μη επιλύσημων προβλημάτων.

Ορισμός 3.4 *Αναγωγή ενός συνόλου A σε ένα σύνολο B είναι η τυχαία ολικώς αναδρομική συνάρτηση f που ικανοποιεί την εξής ισοδυναμία:*

$$x \in A \Leftrightarrow f(x) \in B.$$

επιπλέον υιοθετούμε την παρακάτω ορoολογία:

1. $A \leq_m B \Leftrightarrow$ *υπάρχει πολλά-ένα (many-one) αναγωγή του A στο B.*

2. $A \leq_1 B \Leftrightarrow$ *υπάρχει ένα-προς-ένα αναγωγή του* A *στο* B.

3. $A \equiv B \Leftrightarrow$ *υπάρχει αναγωγή του* A *στο* B *που είναι μετάθεση, δηλαδή η* f *είναι αμφιμονοσήμαντη.*

Πρόταση 3.8 *Αν* $A \leq_m B$ *και το* B *είναι αναδρομικό, τότε και το* A *είναι αναδρομικό.*

Απόδειξη. Αφού $A \leq_m B$ τότε ισχύει ότι $\chi_A(x) = \chi_B(f(x))$ για κάθε x, όπου f αναδρομική. Αν επιπλέον χ_B αναδρομική τότε προφανώς και η χ_A είναι αναδρομική. ■

Πρόταση 3.9 *Αν* $A \leq_m B$ *και το είναι μη-αναδρομικό, τότε και το* B *είναι μη-αναδρομικό.*

Απόδειξη. Ομοίως με την προηγούμενη. ■

Πόρισμα 3.10 *Αν* $\mathcal{K} \leq_m B$*, τότε το* B *είναι προφανώς μη αναδρομικό. Για την ακρίβεια το Halting Set, είναι βασικό πρόβλημα που χρησιμοποιείται στις αποδείξεις μη επιλυσιμότητας με αναγωγή.*

Πρόταση 3.11 *Η σχέση* $\leq_m$ *είναι μεταβατική και ανακλαστική.*

Θεώρημα 3.12 *Ένα σύνολο* A *είναι m-αναγώγιμο στο* $\mathcal{K}$*, ανν είναι RE.*

Απόδειξη. Εφόσων το $\mathcal{K}$ είναι RE, είναι το πεδίο ορισμού μίας μερικώς αναδρομικής g. Σε περίπτωση που $A \leq_m \mathcal{K}$ μέσω μιας αναδρομικής f, τότε το A αποτελεί πεδίο ορισμού της μερικώς αναδρομικής $x \to g(f(x))$. Άρα το A είναι RE.
Αν τώρα το A είναι RE, τότε μπορούμε να ορίσουμε την ακόλουθη 2-μελή συνάρτηση στο f:

$$f(e,x) = \begin{cases} 1 & \text{αν } e \in A \\ \uparrow & \text{αν } e \notin A \end{cases}.$$

■

Ορισμός 3.5 *Η κλάση* $\{B : B \equiv_m A\}$ *λέγεται many-one βαθμός του* A.

Έτσι τα RE σύνολα είναι ακριβώς, τα σύνολα στους κατώτερους many-one βαθμούς απ' το $\mathcal{K}$. Πέρα από το many-one βαθμό του $\mathcal{K}$ υπάρχουν και άλλοι m-βαθμοί. Αρχικά το $\emptyset$ και το $\mathbb{N}$ έχουν σίγουρα διαφορετικό βαθμό. Έπισης έχουμε δεί ότι τα αναδρομικά σύνολα είναι υποσύνολο των RE, και ο βαθμός τους είναι διαφορετικός από αυτόν του $\mathcal{K}$, αφού σίγουρα τα μη αναδρομικά σύνολα της RE, δεν μπορούν να αναχθούν σε αναδρομικά.

Είναι εύλογο να αναρωτηθούμε αν μπορεί να υπάρξει ένας χαρακτηρισμός των συνόλων που είναι RE και έχουν τον ίδιο many-one βαθμό με το $\mathcal{K}$. Έτσι εισήχθηκε ο όρος "δημιουργικά" σύνολα, από τον Post το 1944 και ο χαρακτηρισμός ολοκληρώθηκε από τον Myhill, έντεκα χρόνια μετά.

Σημείωση 3.9 *Συνηθίζεται να συμβολίζουμε τον βαθμό των αναδρομικών προβλημάτων με* $\boldsymbol{0}$ *και το βαθμό του* $\mathcal{K}$ *με* $\boldsymbol{0'}$.

Ορισμός 3.6 *Ένα σύνολο* A *λέγεται* $\mathcal{C}$*-δύσκολο (*$\mathcal{C}$*-hard) ώς προς την κλάση* $\mathcal{C}$*, αν για κάθε* $B \in \mathcal{C}$, $B \leq_m A$.

Ορισμός 3.7 *Ένα σύνολο* A *λέγεται* $\mathcal{C}$*-πλήρες (*$\mathcal{C}$*-complete), αν ανήκει στην κλάση* $\mathcal{C}$ *και είναι* $\mathcal{C} - hard$.

Ορισμός 3.8 *Ένα σύνολο* A *είναι RE-complete ή many-one complete αν είναι RE και ο m-βαθμός του είναι* $\boldsymbol{0'}_m$, *δηλαδή* $\mathcal{K} \leq_m A$.

Θεώρημα 3.13 *(Myhill, 1955) Τα δημιουργικά σύνολα είναι ακριβώς τα* $\{A : A \equiv_m \mathcal{K}\}$.

Στη βιβλιογραφία συχνά συναντάμε ένα άλλο είδος αναγωγής την Turing αναγωγή, συμβολιζόμενη με $\leq_T$. Πολλές φορές η Turing αναγωγή συνδέεται με την έννοια των μαντείων. Μπορούμε να φανταστούμε τα μαντεία σαν κάποιου είδους υπερφυσικές οντότητες που μπορούν να αποφασίσουν για οποιοδήποτε πρόβλημα, και αποτελούν κομμάτι μια μηχανής (πχ ένας μετρητής σε μια διάταξη Register μπορεί να παίζει το ρόλο του μαντείου. Έτσι βάζοντας ένα μη αποκρίσιμο πρόβλημα στο μαντείο μπορούμε να αποφασίσουμε για κάποιο άλλο ισοδύναμο ή ευκολότερο πρόβλημα. Η many one αναγωγή έιναι μια ειδική περίπτωση και ισχυρότερη μορφή της Turing αναγωγής, αφού μας περιορίζει στο να μπορούμε να χρησιμοποιήσουμε μαντείο μόνο μια φορά και στο τέλος, του υπολογισμού μας.
Έχουμε δει έτσι κάποια από τα βασικότερα αποτελέσματα και εργαλεία της θεωρίας αναδρομής, έχοντας περιοριστεί βέβαια μεταξύ αποκρίσιμων και μη αποκρίσιμων προβλημάτων χωρίς όμως να μας ενδιαφέρει ούτε η αποδοτικότητα των μεθόδων επίλυσης τους δεδομένων των περιορισμών που προκύπτουν στον πραγματικό κόσμο, αλλα ούτε και ο βαθμός μη επιλυσιμότητας από την άλλη πλευρά.

Θα συνεχίσουμε σε αυτό το επίπεδο ανάλυσης, χωρίς να εισάγουμε περιορισμούς, αλλά και χωρίς να επεκταθούμε στην πολυπλοκότητα που ορίζεται από την αριθμητική ιεραρχία. Θα ασχοληθούμε με μία δομική μορφή πολυπλοκότητας που αφορά τις κλάσεις συναρτήσεων που έχουμε δει μέχρι τώρα.

Κεφάλαιο 4

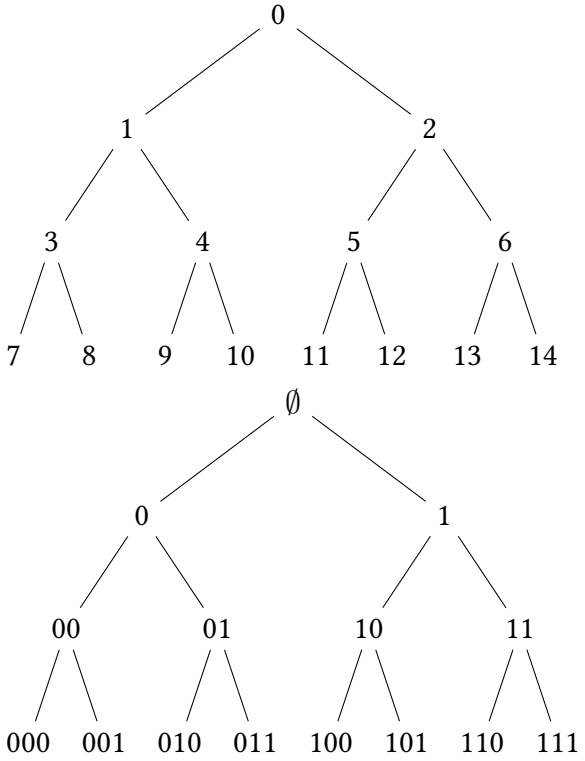
Αλγοριθμική Τυχαιότητα

> Όποιος επεξεργάζεται την ιδέα να χρησιμοποιήσει αριθμητικές μεθόδους για να παράξει τυχαίους αριθμούς διαπράτει σίγουρα κάποιο αμάρτημα. Δεν υπάρχουν τυχαίοι αριθμοί, παρα μόνο μέθοδοι για να παράξουμε τυχαίους αριθμούς και σίγουρα μια αυστηρή αριθμητική διαδικασία δεν αποτελεί μια τέτοια.
>
> *Von Neumann, 1951*

Το κεντρικό ζήτημα με το οποίο θα ασχοληθούμε σε αυτό το κεφάλαιο, είναι το ερώτημα "Τι είναι τυχαίο;".Η απάντηση σε αυτό εξαρτάται από το μαθηματικό αντικείμενο το οποίο μας απασχολεί αν είναι δηλαδή πεπερασμένο, ή άπειρο. Υπάρχουν δύο προσεγγίσεις σε αυτό το ζήτημα, η κλασσική και η αλγοριθμική. Την πρώτη μπορούμε να την εξηγήσουμε με απλό τρόπο βασιζόμενοι στο παραδειγμα των τυχερών παιχνιδιών. Ας φανταστούμε ένα αυτόματο που παράγει μια αλληλουχία αριθμών την οποία πρέπει να μαντέψει ένας παίκτης. Μία αλληλουχία η οποία επιτρέπει στον παίκτη να κερδίζει συνέχεια δεν είναι τυχαία. Έτσι οι αριθμοί που "διαλέγουν" τα μηχανήματα είναι τυχαίοι όταν επιτρέπουν στον καζίνο και όχι στον παίκτη να κερδίζει σε βάθος χρόνου. Η δεύτερη προσέγγιση έχει να κάνει με τον όγκο μη τετριμμένης πληροφορίας που μπορεί να εξάγει κανείς από μια αλληλουχία ή με το κατά πόσο μπορεί να περιγραφεί αυτή με μη τετριμμένο τρόπο. Ουσιαστικά η αλγοριθμική προσέγγιση προσπαθεί να συγκεράσει τις έννοιες της τυχαιότητας και της πολύ μεγάλης δομικής πολυπλοκότητας. Η έννοια της δομικής πολυπλοκότητας είναι διαισθητικά πολύ κατανοητή και θα τη δούμε μέσα από ένα παράδειγμα. Ο αριθμός 2^{1000} παρά το μήκος του, είναι σχετικά απλός, ωστόσο μπορούμε υποθέσουμε αρκετά ασφαλώς ότι υπάρχουν αριθμοί αντίστοιχου μεγέθους, που δεν μπορουν να περιγραφούν με αντίστοιχα απλό τρόπο. Θα δούμε τελικά ότι και οι δύο προσεγγίσεις συνδέονται και η μελέτη τους αποτελεί μέρος του αντικειμένου που ονομάζεται Αλγοριθμική Τυχαιότητα ή αλλιώς Πολυπλοκότητα Kolmogorov.

4.1 Εισαγωγικές Έννοιες

Αρχικά πρέπει να σημειώσουμε ότι αντι για το $\mathbb{N}$ θα χρησιμοποιήσουμε σε αυτό το κεφάλαιο κυρίως δυαδικές ακολουθίες (bitstrings), χωρίς αυτό να επηρεάζει κάπως την προηγούμενη θεωρία μας αφού θα βασιστούμε στην δυαδική αμφιμονοσήμαντη κωδικοποίηση $tree : 2^{\leq \mathbb{N}} \to \mathbb{N}$ που απεικονίζεται παρακάτω:



Σημείωση 4.1 *Θα συμβολίζουμε τα bitstrings με p, r, s, t και το μήκος τους $len(s) = |s|$. Επίσης $tree(s) \geqslant tree(r)$ αν $|s| \geqslant |r|$ ή το s προηγείται του r βάσει λεξικογραφικής διάταξης.*

Μέχρι στιγμής έχουμε δει ότι υπάρχουν πάρα πολλοί τρόποι να για να περιγράψεις ένα μαθηματικό αντικείμενο. Ας τους ανατρέξουμε όλους. Αν ονομάσουμε περιεχόμενη πληροφορία του αντικειμένου αυτού την μικρότερη περιγραφή αυτου, σε μορφή bitstring, καταλήγουμε σε μία παραλλαγή του Berry's Paradox, αφού η έννοια μας δεν είναι καλά ορισμένη.

Έστω W η μικρότερη δυαδική λέξη (με λεξικογραφική διάταξη) που δεν μπορεί να περιγραφεί από οποιαδήποτε δυαδική λέξη μήκους μικρότερου του 3000.

Όμως η περιγραφή αυτή, με μια τυχαία περιγραφή (πχ κωδικούς utf-8), μπορεί να γραφτεί σαν δυαδική λέξη μήκους $2072 \leq 3000$ (μαζί με τα κενά).

Το ασθενές κομμάτι του ορισμού που οδηγεί σε αυτού του τύπου τα παράδοξα είναι το σημείο της περιγραφής. Έτσι ο Kolmogorov έκανε μία μεγαλοφυή κίνηση: αντικατέστησε τον

όρο *περιγραφή* με τον όρο *πρόγραμμα*, εκμεταλλευόμενος έτσι τον επιτυχημένο φορμαλισμό του ασαφούς αυτού όρου που είχε εισαχθεί ήδη τη δεκαετία του '30.

Έτσι η βασική ιδέα της Πολυπλοκότητας Kolmogorov μπορεί να συμπυκνωθεί στην εξής σχέση:

περιγραφή ≡ πρόγραμμα

Έτσι θα χρησιμοποιήσουμε τη θεωρία που έχουμε μέχρι τώρα και το μοντέλο κωδικοποιήσης των αναδρομικών συναρτήσεων που είχαμε εισάγει (με μετατροπή σε bitstrings) για να παρουσιάσουμε τα επόμενα αποτελέσματα.

4.2 Απλή πολυπλοκότητα Kolmogorov

Αρχικά, ας δούμε μια γενικευμένη εκδοχή της πολυπλοκότητας Kolmogorov. Έστω λοιπόν το σύνολο S τον αντικειμένων που μας αφορούν και έστω μια κωδικοποίηση $n(x)$ του αντικειμένου x. Μας ενδιαφέρει το γεγονός ότι το $n(x)$ μπορεί να μην είναι ο πιο οικονομικός τρόπος για να προσδιορίσουμε το x. Για να συγκρίνουμε τις μεθόδους προσδιορισμού, θα αντιμετωπίζουμε μια μέθοδο σαν μία μερική συνάρτηση πάνω στους φυσικούς $f(p)$. Δεν θα υποθέσουμε ακόμα ότι η f είναι αναδρομική για να διατηρήσουμε τη γενικότητα και να τονίσουμε σε ποιό σημείο είναι αναγκαία η εισαγωγή της αναδρομής. Έτσι η πολυπλοκότητα ενός αντικειμένου x δεδομένης της μεθόδου κωδικοποίησης f είναι ορίζεται ως:

$$C_f = min\{len(p) : f(p) = n(x)\}$$

και $C_f = \infty$ αν δεν υπάρχει τέτοιο p. Γενικά μπορούμε να φανταστούμε την p σαν πρόγραμμα και την f σαν υπολογιστή και αναζητούμε το μικρότερο πρόγραμμα που υπολογίζει ο x, με αδιάφορη ή κενή είσοδο.

Πρόταση 4.1 *Ανεξάρτητα από τη μέθοδο κωδικοποίησης η πολυπλοκότητα ενός αντικειμένου δεν μπορεί να διαφέρει παρα κατά μία σταθερά:* $C_f(x) \leqslant C_g(x) + c_{f,g}$.

Γενικά στη βιβλιογραφία συναντάμε διάφορους ορισμούς της πολυπλοκότητας Kolmogorov, ωστόσο σημείο κλειδί στην ανάπτυξη της θεωρίας, είναι ο περιορισμός της στην κλάση των αναδρομικών συναρτήσεων. Μας δίνεται έτσι η δυνατότητα να χρησιμοποιήσουμε μια συγκεκριμένη μέθοδο κωδικοποίησης, αφού όπως είδαμε και πιο πριν η κλάση αυτή περιέχει ένα καθολικό στοιχείο. Έτσι, στη συνέχεια θα χρησιμοποιούμε μόνο αναδρομικές κωδικοποιήσεις.

4.2.1 Θεώρημα αναλλοίωτου

Το πρόβλημα με τον προηγούμενο ορισμό είναι ότι εξαρτάται υπερβολικά από τη συνάρτηση f. Αυτό το πρόβλημα έρχεται να ξεπεράσει το θεώρημα αναλλοίωτου.

Ορισμός 4.1 *Καθολική λέγεται μία συνάρτηση $U : \{0,1\}^* \to \{0,1\}^*$ τέτοια ώστε για κάθε μερικώς αναδρομική f, υπάρχει κωδικός e:*

$$\forall x[U(e,x) = f(x)].$$

μάλιστα για κάθε f, ισχύει ότι

$$C_U(x) \leqslant C_f(x) + O(1)$$

Η καθολική συνάρτηση U ένα αντικείμενο μεγαλύτερης πολυπλοκότητας σε σχέση με τα προγράμματα που εκτελεί. Μπορούμε σε μία αναλογία με τους πραγματικούς υπολογιστές, να έχουμε στο μυαλό μας αυτές τις συναρτήσεις σαν μηχανές που περιλαμβάνουν μια γλώσσα προγραμματισμού και έναν compiler και τα απαραίτητα μηχανικά μέρη για να τρέξουν το e με κάποιο input.

Σημείωση 4.2 *Υπάρχει απόλυτη ταύτιση με το αποτέλεσμα που προέκυψε στο δεύτερο κεφάλαιο, απλά το προσαρμόσαμε τα συμφραζόμενα στη θεωρία της πολυπλοκότητας Kolmogorov.*

Είναι αρκετά βασικό, το γεγονός ότι με τον όρο περιγραφή αναφερόμαστε σε έναν διμερή κώδικα, $\langle e, y\rangle$ όπου e ο κωδικός της αναδρομικής συνάρτησης, ή της μηχανής και y η ακολουθία εισόδου.
Έτσι ως περιγραφή στην πραγματικότητα ονομάζουμε την παράθεση των δύο ακολουθιών. Πολλές φορές στη βιβλιογραφία μπορεί να συναντήσουμε και τον ορισμό $K(x) = \mu e(f_e(0) \simeq x)$ όπου αναζητούμε το μικρότερο πρόγραμμα που υπολογίζει το x χωρίς είσοδο.

Πρόταση 4.2 *Έστω $s \geqslant 1$, υπάρχει απεικόνιση $\langle\ \rangle : (\{0,1\}^*)^{s+1} \to \{0,1\}^*$, ένα-προς-ένα και αναδρομική, τέτοια ώστε για κάθε $u_1, ..., u_s, v \in \{0,1\}^*$, $len(\langle u_1, ..., u_s, v\rangle) = 2(len(u_1) + ... + len(u_s)) + len(v) + 1$.*

Ορισμός 4.2 *Η κλάση των στοιχειωδών συνόλων παράγεται ως εξής:*

- *Περιλαμβάνει το $\mathbb{N}$ και τα A^*, όπου A πεπερασμένο ή αριθμήσιμο αλφάβητο, και $*$ το Kleene star.*
- *Είναι κλειστό στο πεπερασμένο γινόμενο, στο γινόμενο με οποιοδήποτε μη κενό σύνολο και στο Kleene star.*

Θεώρημα 4.3 *(Invariance theorem, Kolmogorov, 1965) Έστω $\mathcal{O}$ ένα στοιχειώδες σύνολο. Ανάμεσα στις C_f όπου f ανήκει στην κλάση των μερικώς αναδρομικών συναρτήσεων $f : \{0,1\}^* \to \mathcal{O}$, υπάρχει μία ελάχιστη ως προς μια σταθερά, η οποία λέγεται βέλτιστη.*
Επιπλέον κάθε καθολική συνάρτηση U είναι βέλτιστη.

Απόδειξη. Στην απόδειξη θα χρησιμοποιήσουμε μόνο το θεώρημα απαρίθμησης.
Έστω $U : \{0,1\}^* \times \{0,1\}^* \to \mathcal{O}$ μερικώς αναδρομική και καθολική για τις μερικώς αναδρομικές $\{0,1\}^* \to \mathcal{O}$. Όπως είδαμε προηγουμένως υπάρχει $c : \{0,1\}^* \times \{0,1\}^* \to \{0,1\}^*$ τέτοια ώστε $|c(e,x)| = 2len(e) + len(x) + 1$. Έστω $V : \{0,1\}^* \to \mathcal{O}$ τέτοια ώστε:

$$\forall e \in \{0,1\}^* \forall x \in \{0,1\}^* V(c(e,x)) = U(e,x)$$

όπου είτε ορίζονται και τα δύο μέρη της εξίσωσης είτε όχι. Έτσι για κάθε μερικώς αναδρομική $f : \{0,1\}^* \to \mathcal{O}$ και κάθε $y \in \mathcal{O}$, αν $f = U_e \Leftrightarrow f(x) = U(e,x) \forall x$ έχουμε:

$$\begin{gathered} C_V(y) = min\{|p| : V(p) = y\} \\ \leqslant min\{c(e,x) : V(c(e,x)) = y\} \\ \text{(ας παρατηρήσουμε εδώ ότι το min εξαρτάται από το } x \text{ αφού το } e \text{ είναι δεδομένο)} \\ = min\{|c(e,x)| : U(e,x) = y\} \\ = min\{len(x) + 2len(e) + 1 : f(x) = y\} \\ = min\{len(x) : f(x) = y\} + 2len(e) + 1 \\ = C_f(y) + 2len(e) + 1 \end{gathered}$$

■

Το θεώρημα αναλλοίωτου έχει μια μεγάλη σημασία που προεκτείνεται στους πραγματκούς υπολογιστές. Δείχνει ότι για κάθε γλώσσα περιγραφής-απαρίθμηση, υπάρχει μια βέλτιστη γλώσσα τουλάχιστον τόσο αποδοτική όσο η αρχική. Αν για παράδειγμα θεωρήσουμε μια γλώσσα προγραμματισμού όπως η Python, κωδικοποιημένη στο δυαδικό, τη γλώσσα περιγραφής, τότε το $K_{Python}(x)$ θα είναι το μήκος του μικρότερου προγράμματος που επιστρέφει το x.

Η συνάρτηση U, ουσιαστικά είναι ο υπολογιστής ο οποίος μπορεί να μεταφράσει ένα πρόγραμμα από Python σε γλώσσα μηχανής. Μάλιστα η περιγραφή του x σε γλώσσα μηχανής, είναι μόνο κατα πεπερασμένο τρόπο μεγαλύτερη από την περιγραφή του x στην Python και το παραπάνω μήκος προκύπτει από τον interpreter της Python την U.

Ορισμός 4.3 *Είναι φυσικό από εδώ και στο εξής να ορίσουμε την (απλή) πολυπλοκότητα Kolmogorov ως*

$$C(x) := C_U(x).$$

Θεώρημα 4.4 *Για κάθε n, υπάρχει x με $len(x) = n$ και $C(x) \geqslant n$.*

Απόδειξη. Υπάρχουν το πολύ $2^n - 1$ δυαδικές ακολουθίες μήκους μικρότερου του n, οπότε υπάρχουν το πολύ $2^n - 1$ δυαδικές ακολουθίες με πολυπλοκότητα μικρότερη του n. ■

Άρα είδαμε οτι η πολυπλοκότητα Kolmogorov είναι ένας ακέραιος που προσδιορίζεται ως προς μια σταθερά. Στην πραγματικότητα αυτή η σταθερά αντικατοπτρίζει την ποικιλία των μοντέλων υπολογισμού. Μορφές καθολικότητας υπάρχουν τόσο στη θεωρία αναδρομής, όσο και στις μηχανές Turing ή στα συστήματα εξισώσεων Herbrand-Gödel. Αν θεωρήσουμε κάποιο από από αυτά το σωστό, μπορούμε ίσως να συγκεκριμενοποιήσουμε την πολυπλοκότητα Kolmogorov, ωστόσο η σημασία του θεωρήματος αναλλοίωτου παραμένει τεράστια αφού ως αποτέλεσμα θα εμφανίζεται συνεχώς στη θεωρία μας. Αντί να μετρήσουμε την ποσότητα της πληροφορίας στο $y \in \mathcal{O}$, την μετράμε δεδομένου ενός δεύτερου αντικειμένου z, το οποίο θα μας βοηθήσει να υπολογίσουμε το y. Τετριμμένη είναι η περίπτωση που καλούμαστε να υπολογίσουμε το y δεδομένου ενός $z = y$.

4.2.2 Άνω Φράγματα

Πρόταση 4.5 *Τα ακόλουθα είναι ισοδύναμα:*

- $C(x) \leqslant len(x) + O(1)$
- $C(x, x) \leqslant len(x) + O(1)$
- *Αν* $h : \{0,1\}^* \to \{0,1\}^*$ *αναδρομική, τότε* $C(h(x)) \leqslant C(x) + O(1)$.

Μέχρι τώρα αν και από το θεώρημα απαρίθμησης, θεωρούσαμε συναρτήσεις της μορφής *(προγραμμα,είσοδος)* $\to$ *έξοδος*, ωστόσο στον ορισμό της πολυπλοκότητας Kolmogorov όπως είδαμε μπορούσαμε να παραβλέψουμε την είσοδο, αφού μας ενδιέφερε κυρίως το πρόγραμμα και η κωδικοποίηση αυτού.
Υπάρχει όμως και η εξαρτημένη πολυπλοκότητα Kolmogorov που λαμβάνει υπόψην της μια βοηθητική είσοδο.

Ορισμός 4.4 *Εξαρτημένη πολυπλοκότητα Kolmogorvo* $C_f(y|z) = min\{|p| : f(p, z) = y$.

Πόρισμα 4.6 *Υπάρχει και η αντίστοιχη εκδοχή του θεωρήματος αναλλοίωτου, δηλαδή,*

$$C_U(y|z) \leqslant C_f(y|z) + O(1)$$

Βάσει αυτών μπορούμε να ορίσουμε μία ακόμα έννοια αυτήν της περιεχόμενης ποσότητας πληροφορίας, που ουσιαστικά εκφράζει την δυσκολία του να παράξουμε ένα bitstring x, αν έχουμε στη διάθεση μας ένα y.

Ορισμός 4.5 *Η (απλή) περιεχόμενη πληροφορία σε ένα bitstring* x*, δοσμένης της* y *ορίζεται ως:*

$$I_C(x : y) := C(x) - C(x|y).$$

Θεώρημα 4.7 *(Συμμετρία της πληροφορίας, Levin-Kolmogorov).* $I_C(x : y) = I_C(y : x) \pm O(\log C(x, y))$.

Πόρισμα 4.8 *Είναι άμεσο ότι* $I_C(x : y) = I_C(y : x) \pm O(\log n)$*, όπου* $n = max\{len(x), |y|\}$.

Λήμμα 4.9 *(Martin-Löf) Έστω* k*, και* z *με αρκετά μεγάλο μήκος ώστε να υπάρχει αρχικό τμήμα* x *του* z *τέτοιο ώστε* $C(x) < len(x) - k$*. Έτσι για οποιοδήποτε* d *έχουμε* $z = xy$ *τέτοιο ώστε* $C(z) > C(x) + C(y) + d$.

Απόδειξη. Έστω v αρχικό τμήμα του z και n τέτοιο ώστε το v να είναι η n-οστό bitstring βάσει της λεξικογραφικής διάταξης του $2^{<\mathbb{N}}$. Έστω r τα επόμενα n στοιχεία του z που διαδέχονται το v και $x = vr$. Για να παράξουμε το x πρέπει να γνωρίζουμε μόνο το r, αφού το μήκος του μπορεί να μας δώσει το v. Έτσι υπάρχει μια σταθερά c τέτοια ώστε, $C(x) \leqslant |r| + c$, όπου το c δεν εξαρτάται από την επιλογή του v. Όμως $len(x) = len(v) + len(r)$ κι έτσι αν $len(v) > c + k$, τότε $C(x) < len(x) - k$.

Για το δεύτερο κομμάτι, έστω c τέτοιο ώστε $C(y) \leqslant |y| + c$ για όλα τα y. και $k = c + d$. Έστω z αρκετά μεγάλο bitstring, τέτοιο ώστε $C(z) \geqslant len(z)$. Από το πρώτο μέρος του λήμματος έχουμε ότι $z = xy$ και $C(x) < len(x) - k$. Τότε $C(z) \geqslant |z| = len(x) + len(y) > C(x) + k + C(y) - c = C(x) + C(y) + d$. ■

Αναζητώντας άνω φράγματα για την $C(x)$, προκύπτει όπως είναι φυσικό η ανάγκη να φράξουμε την ποσότητα $C(x, y)$. Σίγουρα θα ήταν πολύ βολικό να ισχύει $C(x, y) \leqslant C(x) + C(y) + c$. Όμως προκύπτει το εξής πρόβλημα. Έστω ότι υπάρχουν προγράμματα g και q που να υπολογίζουν τα x και y. Η δυσκολία που θα αντιμετωπίσουμε είναι ότι αν θελήσουμε να κωδικοποιήσουμε τα δύο προγράμματα μαζί θα πρέπει να έχουμε τη δυνατότητα να τα αποκωδικοποιήσουμε ακέραια αργότερα. Έτσι αν απλά κωδικοποιήσουμε το pq, δεν θα ξέρουμε που αρχίζει το q και που τελειώνει το p. Αν προσθέσουμε παραπάνω στοιχεία στην κωδικοποίηση πχ $0^{len(p)}1pq$ τότε καταναλώνουμε πολύ "χώρο". Το αρχικό πρόβλημα ανακύπτει και αν χρησιμοποιήσουμε το $len(p)pq$. Ένας τρόπος για να ξεπεράσουμε το πρόβλημα αυτό είναι να "αυτο-οριοθετήσουμε" τα bitstrings διπλασιάζοντας τα ψηφία του ενός και χρησιμοποιώντας μια ένδειξη για το τέλος του, δηλαδή αν το $|p|$ είναι το 1010 τότε το κωδικοποιούμε ως 1100110001, με το 01 να είναι η προαναφερθείσα ένδειξη. Φυσικά μπορούμε να επαναλάβουμε αυτή τη διαδικασία όσες φορές θέλουμε, χρησιμοποιώντας το $len(p')p'pq$ όπου $p' = |p|$ και κωδικοποιώντας το $|p'|$ με τον προηγούμενο τρόπο.
Έτσι προκύπτει το ακόλουθο φράγμα:

$$C(x, y) \leqslant C(x) + C(y) + 2\log C(x) + c$$
$$C(x, y) \leqslant C(x) + C(y) + \log C(x) + 2\log\log C(x) + c$$
$$C(x, y) \leqslant C(x) + C(y) + \log C(x) + \log\log C(x) + 2\log\log\log C(x) + c$$
κ.ο.κ.

Πόρισμα 4.10 $C(xy) \leqslant C(x) + C(y) + 2\log len(x) + O(1)$.

Πόρισμα 4.11 $C(xy) \leqslant C(x, y) \leqslant C(x) + C(y) + 2C(x) + O(1)$.

Ας δούμε τώρα δύο μικρές εφαρμογές της αρχικής θεωρίας μας.

Παράδειγμα 4.1 *Πόσο μεγάλος είναι ο i-οστός πρώτος; Έστω $m \in \mathbb{N}$ τέτοιο ώστε ο p_i να είναι ο μέγιστος πρώτος διαιρέτης του m. Για να περιγράψουμε το m χρειαζόμαστε μόνο το i και το $\frac{m}{p_i}$. Οπότε από προηγούμενο λήμμα έχουμε,*

$$C(m) \leqslant C(i, \tfrac{m}{p_i}) + O(1) \leqslant C(i) + C(\tfrac{m}{p_i}) + 2\log^{(2)} i + O(1) \leqslant \log i + \log\tfrac{m}{p_i} + 2\log^{(2)} i + O(1)$$

(όπου $\log^{(2)} i$ το μήκος της δυαδικής μορφής του i) Επειδή υπάρχουν άπειρες επιλογές m, ώστε $C(m) \geqslant \log m - O(1)$ έχουμε

$$\log m \leqslant \log i + \log\tfrac{m}{p_i} + 2\log^{(2)} i + O(1) = \log i + \log m - \log p_i + O(1),$$

που υπονοεί ότι,

$$\log p_i \leqslant \log i + 2\log^{(2)} i + O(1).$$

Έτσι $p_i \leqslant O(i \log^2 i)$ που είναι αρκετά κοντά στην πραγματική απάντηση $i \log i$.

Παράδειγμα 4.2 *Μία άλλη περίπτωση, προερχόμενη από την θεωρία αναδρομής, είναι η κατασκευή ενός απρόσβλητου (immune) συνόλου, δηλαδή ενός συνόλου που δεν περιέχει άπειρο, RE, υποσύνολο. Έστω*

$$A = \{x : C(x) \geqslant \tfrac{len(x)}{2}\}.$$

Τότε το A είναι απρόσβλητο. Πράγματι, έστω ότι το A έχει ένα άπειρο, RE υποσύνολο B. Τότε

$$C(h(n)) \geqslant \tfrac{|h(n)|}{2} > \tfrac{n}{2},$$

αφού $h(n) \in A$, αλλά μπορούμε να παράξουμε το $h(n)$ από το n απλά τρέχοντας τη συνάρτηση απαρίθμησης του B μέχρι να βρούμε στοιχείο με μήκος μεγαλύτερο του n, έτσι

$$C(h(n)) \leqslant C(n) + O(1) \leqslant \log n + O(1).$$

Το οποίο για αρκετά μεγάλο n, είναι άτοπο.

4.2.3 Εξαρτημένη Πολυπλοκότητα

Μέχρι τώρα αν και από το θεώρημα απαρίθμησης, θεωρούσαμε συναρτήσεις της μορφής *(προγραμμα,είσοδος)* $\rightarrow$ *έξοδος*, ωστόσο στον ορισμό της πολυπλοκότητας Kolmogorov όπως είδαμε μπορούσαμε να παραβλέψουμε την είσοδο, αφού μας ενδιέφερε κυρίως το πρόγραμμα και η κωδικοποίηση αυτού.
Υπάρχει όμως και η εξαρτημένη πολυπλοκότητα Kolmogorov που λαμβάνει υπόψην της μια βοηθητική είσοδο.

Ορισμός 4.6 *Εξαρτημένη πολυπλοκότητα Kolmogorov $C_f(y|z) = min\{len(e) : f_e(z) = y\}$.*

Πόρισμα 4.12 *Υπάρχει και η αντίστοιχη εκδοχή του θεωρήματος αναλλοίωτου, δηλαδή,*

$$C_U(y|z) \leqslant C_f(y|z) + O(1)$$

Βάσει αυτών μπορούμε να ορίσουμε μία ακόμα έννοια αυτήν της περιεχόμενης ποσότητας πληροφορίας, που ουσιαστικά εκφράζει την δυσκολία του να παράξουμε ένα bitstring x, αν έχουμε στη διάθεση μας ένα y.

Ορισμός 4.7 *Η (απλή) περιεχόμενη πληροφορία σε ένα bitstring x, δοσμένης της y ορίζεται ως:*

$$I_C(x : y) := C(x) - C(x|y).$$

Θεώρημα 4.13 *(Συμμετρία της πληροφορίας, Levin-Kolmogorov).*

$$I_C(x : y) = I_C(y : x) \pm O(\log C(x, y)).$$

Πόρισμα 4.14 *Είναι άμεσο ότι $I_C(x : y) = I_C(y : x) \pm O(\log n)$, όπου $n = max\{len(x), |y|\}$.*

4.3 Πολυπλκότητα Kolmogorov χωρίς πρόθεμα

Σε αυτή την ενότητα θα δούμε μια παραλλαγή της πολυπλοκότητας Kolmogorov, την χωρίς-πρόθεμα εκδοχή της. Το κύριο κίνητρο για την ανάπτυξη αυτής της εκδοχής, ήταν εν τέλει να μελετηθεί η αλγοριθμική τυχαιότητα των άπειρων δυαδικών ακολουθιών, ωστόσο πολλοί τη θεωρούν την σωστή εκδοχή περιγραφικής πολυπλοκότητας ακόμα και για τα πεπερασμένα bitstrings.

Το κύριο επιχείρημα για την υποστήριξη αυτής της θέσης, είναι ότι κατά κάποιον τρόπο τα προγράμματα πρέπει να έχουν ικανότητα αυτοπεριορισμού. Αυτό σημαίνει ότι αν κάποιο x παράγεται μέσω μιας καθολικής συνάρτησης/μηχανής από το πρόγραμμα y, η πληροοφορία που περιέχεται στο y δεν είναι μόνο τα ψηφία του, αλλά και το μήκος του. Έτσι αν για παράδειγμα μια καθολική μηχανή Turing U έχει τη δυνατότητα να χρησιμοποιεί ως πληροφορία και το μήκος του προγράμματος, θα πρέπει στο αλφάβητό της να περιλαμβάνεται και κάποιο σύμβολο τερματισμού, το οποίο εγγυάται τον αυτοπεριοριζόμενο χαρακτήρα, αφού έτσι ο κωδικός ενός προγράμματος αν και μπορεί να αποτελεί αρχικό τμήμα άλλου κωδικού προγράμματος, υπάρχει τρόπος να γίνονται διακριτά.

Για να αντιμετωπιστεί αυτό το ζήτημα, θεωρούμε ότι η απαρίθμηση των προγραμμάτων είναι χωρίς πρόθεμα, δηλαδή κανένας κωδικός δεν αποτελεί αρχικό τμήμα άλλου κωδικού.

Ορισμός 4.8 *Λέμε ότι δύο ακολουθίες s, t είναι συγκρίσιμες, αν η μία αποτελεί αρχικό τμήμα της άλλης. Καλούμε ένα σύνολο $A \subseteq \{0,1\}^*$ σύνολο χωρίς πρόθεμα, αν κάθε ζεύγος $s, t \in A$ είναι μη συγκρίσιμες.*

Ορισμός 4.9 *Ορίζουμε ως πολυπλοκότητα Kolmogorov χωρίς πρόθεμα και συβολίζουμε $K(x)$ την $C_U(x)$ με πεδίο ορισμού της U (κωδικούς) ένα σύνολο χωρις πρόθεμα.*

Θεωρούμε ότι $K(x, y) = K(\langle x, y\rangle)$. Το γεγονός ο αλγόριθμος αποκωδικοποίησης είναι χωρίς πρόθεμα έχει μεγάλο πλεονέκτημα, αφού μπορούμε πλέον να παραθέτουμε περιγραφές χωρίς να μας ενδιαφέρει που τελειώνει η μία και που αρχίζει η επόμενη. Προηγουμένως, αυτό το στοιχείο χανόταν, προσθέτοντας έτσι τους λογαριθμικούς όρους στους τύπους μας. Έτσι σε αντίθεση με την απλή $C(x)$ ίσχύει ότι

$$K(x, y) = K(x) + K(y) + O(1),$$

και επίσης

$$C(x|y) \leqslant K(x|y) \leqslant C(x|y) + 2\log C(x|y).$$

4.4 Τυχαιότητα

Πολλές προσπάθειες έχουν γίνει, για να αναπτυχθεί ένας φορμαλισμός για την έννοια της τυχαιότητας. Στην θεωρία πιθανοτήτων, αν ρίξουμε ένα αμερόληπτο νόμισμα εκατό φορές, τότε το να έρθουν εκατό φορές γράμματα, είναι το ίδιο πιθανό με οποιοδήποτε άλλο αποτέλεσμα, απορρίπτοντας έτσι αξιωματικά την έννοια του τυχαίου. Στην κρυπτογραφεία

αντιθέτως γίνεται συνεχώς χρήση τυχαίων αντικειμένων, που όμως δεν είναι πραγματικά τυχαία, απλά παράγονται από έναν αρκετά μεγάλης πολυπλοκότητας μηχανισμό, τέτοιον ώστε η ανακάλυψη του να μην είναι δυνατή σε εύλογο χρονικό διάστημα. Έτσι βλέπουμε αρχικά ότι υπάρχει μια ταύτιση της τυχαιότητας και της μεγάλης δομικής πολυπλοκότητας, των αντικειμένων που εξετάζουμε. Ψευδής τυχαιότητα είναι αυτή η μορφή τυχαιότητας που παράγεται από αλγόριθμους και στόχος είναι να "περνάει" κάποια στατιστικά τεστ. Υπάρχει όμως πραγματική τυχαιότητα;

Καμία απο τις προσπάθειες να δομηθεί μια μαθηματική θεωρία για τα τυχαία αντικείμενα δεν ήταν αρκετά ικανοποιητική, εως ότου ο Kolmogorov το '60 βάσισες μια τέτοια θεωρία την πολυπλοκότητα Kolmogorov και κατά συνέπεια στην θεωρία υπολογισμού. Ανεξάρτητα από αυτό και ο Chaitin το 1965 δημοσίευσε σχετικά αποτελέσματα των οποίων η κεντρική ιδέα ήταν η εξής:

- Όσο μεγαλύτερη είναι η πολυπλοκότητα Kolmogorov ενός κειμένου, τόσο πιο τυχαίο είναι το κείμενο.
- Όσο μεγαλύτερη είναι η περιεχόμενη πληροφορία του, τόσο πιο συμπιεσμένο είναι το κείμενο.

Έτσι η θεωρία της δομικής πολυπλοκότητας είναι επίσης και θεωρία της τυχαιότητας.

Ορισμός 4.10 *Ένα bitstring x καλείται τυχαίο, αν $C(x) \geqslant len(x)$.*

Σημείωση 4.3 *Υπενθυμίζουμε ότι υπάρχει c τέτοιο ώστε για κάθε $x \in \{0,1\}^*, C(x) \leqslant len(x)+c$. Αυτό συμβαίνει επειδή υπάρχει κάποιο "χαζό" πρόγραμμα μήκους περίπου $len(x)$ που υπολογίζει το x υπαγορεύοντας ένα-ένα τα στοιχεία του. Από αυτό το γεγονός προκύπτει η έννοια της ασυμπιεστότητας. Το x είναι ασυμπίεστο αν δεν υπάρχει πιο σύντομος τρόπος από τον προηγούμενο για να το παράξουμε ως έξοδο κάποιου προγράμματος. Η έννοια του τυχαίου και του ασυμπίεστου είναι άρρηκτα συνδεδεμένες, αφού η ασυμπιεστότητα είναι αναγκαία συνθήκη για την τυχαιότητα.*

Ορισμός 4.11 *Μια λέξη x είναι c-ασυμπίεστη, αν $C(x) \geqslant len(x) - c$.*

Έχουμε δει ότι η περιγραφή μιας λέξης δεν μπορεί να είναι παρά λίγο μεγαλύτερη από την ίδια τη λέξη. Ωστόσο συχνά μια λέξη μπορεί να έχει μικρότερη περιγραφή από το μήκος της δηλαδή μπορεί να συμπιεστεί. Ωστόσο, εμείς θα ασχοληθούμε με τις ασυμπίεστες λέξεις, οι οποίες μοιάζουν δυαδικές ακολουθίες που συμβολίζουν το αποτέλεσμα διαδοχικών ρίψεων ενός αμερόληπτου νομίσματος.

Πρόταση 4.15 *Για κάθε n υπάρχει τυχαίο x με $len(x) = n$. Συμπέρασμα που βγαίνει εύκολα από την αρχή του περιστεριώνα:*

$$|\{s : len(s) < n\}| = 2^0 + 2^1 + ... + 2^{n-1} = 2^n - 1.$$

και αφού $|s : len(s) = n| = 2^n$, υπάρχει τυχαίο x μήκους n για όλα τα n.

Πρόταση 4.16 *Για κάθε n και για κάθε $k \in \{0, ..., n\}$ ισχύει ότι:*

$$|x : len(x) = n, C(x) \geqslant n - k\}| > 2^n(1 - \tfrac{1}{2^k}).$$

Έτσι από τις λέξεις μήκους 100, το $99,9\%$ έχει $C(x) \geqslant 90$.

4.4.1 Μη υπολογισιμότητα της C(x)

Θα εξετάσουμε σε αυτή την ενότητα την $C(x)$, σαν αριθμοθεωρητική συνάρτηση. Έχουμε ήδη βρει ένα άνω φράγμα, το οποίο μάλιστα είναι το κατώτατο άνω φράγμα, το $C(x) \leqslant len(x) + O(1)$. Θα δούμε ότι αν και το φράγμα αυτό είναι υπολογίσιμο, δεν μπορούμε να βρούμε ένα ανώτατο, μονότονο κάτω φράγμα που να είναι υπολογίσιμο.

Θεώρημα 4.17 *Τα επόμενα είναι ισοδύναμα:*

(i) Η συνάρτηση $C(x)$ είναι μη φραγμένη.

(ii) Έστω συνάρτηση m, τέτοια ώστε $m(x) = min\{C(y) : y \geqslant x\}$. Δηλαδή η m είναι η μεγαλύτερη μονότονη συνάρτηση που φράσσει την C από κάτω. Τότε η m δεν είναι φραγμένη.

(iii) Για οποιαδήποτε μερικώς αναδρομική $f(x)$ που τείνει στο άπειρο από κάποιο x_0 και μετά, έχουμε $m(x) < f(x)$ για όλα εκτός πεπερασμένου αριθμού x. Με άλλα λόγια, μπορεί η m να τείνει στο άπειρο, ωστόσο το κάνει πιο αργά από οποιαδήποτε μη φραγμένη μερικώς αναδρομική συνάρτηση.

Απόδειξη.

(i) Το i προκύπτει άμεσα από το ii.

(i) Για κάθε i υπάρχει ένα ελάχστιο x_i, τέτοιο ώστε για κάθε $x > x_i$, το μικρότερο πρόγραμμα e που επιστρέφει x έχει μήκος μεγαλύτερο ή ίσο με i. Αυτό έπεται άμεσα από το γεγονός ότι υπάρχουν μόνο πεπερασμένος αριθμός προγραμμάτων για κάθε μήκος i. Βέβαια για κάθε i υπάρχει $x_{i+1} > x_i$ όμως τότε η m έχει την ακόλουθη ιδιότητα, ότι $m(x) = i$ για $x_i < x < x_{i+1}$.

(iii) Έστω ότι δεν ισχύει: Υπάρχει δηλαδή μονότονη, μη-φθίνουσα, μη-φραγμένη μερικώς αναδρομική $f(x) < m(x)$ για άπειρα x. Τότε το $A = \{x : f(x) < \infty\}$ είναι ένα άπειρο, RE σύνολο. Άρα το A περιέχει ένα άπειρο αναδρομικό υποσύνολο B. Έστω

$$\psi(x) = \begin{cases} f(x) & \text{για } x \in B \\ f(y) & \text{με } y = max\{z : z \in B \wedge z < x\} \text{ αλλιώς} \end{cases}.$$

Η ψ έιναι ολική, αναδρομική, τείνει μονότονα στο άπειρο και $\psi(x) \leqslant m(x)$ για άπειρα x.
Ορίζουμε τώρα $M(\alpha) = max\{x : C(x) \leqslant \alpha\{$. Τότε $M(\alpha) + 1 = min\{x : m(x) > \alpha\}$. Εύκολα βλέπουμε ότι

$$max\{x : \psi(x) \leqslant \alpha + 1\} \geqslant min\{x : m(x) > \alpha\{ > M(\alpha),$$

για άπειρα α και η συνάρτηση $F(\alpha) = max\{x : \psi(x) \leqslant \alpha + 1\}$ είναι εμφανώς ολική αναδρομική. Έτσι $F(\alpha) > M(\alpha)$ για άπειρα α. Αλλά

$$C(F(\alpha)) \leqslant C_F(F(\alpha)) + O(1) \leqslant |\alpha| + O(1).$$

Αυτό όμως υπονοεί την ύπαρξη ενός c, τέτοιου ώστε $|\alpha| + c \geqslant \alpha$ για άπειρα α, το οποίο είναι αδύνατο.

■

Θεώρημα 4.18 *Η $C(x)$ δεν έιναι αναδρομική.*

Απόδειξη. Για απλούστευση θα χρησιμοποιήσουμε το $\mathbb{N}$. Έστω συνάρτηση $L : \mathbb{N} \to \mathbb{N}$, τέτοια ώστε

$$L(n) = \mu k C(k) \geqslant 2n.$$

Έτσι $C(L(n)) \geqslant 2n \forall n \in \mathbb{N}$. Αν η C ήταν αναδρομική, τότε θα ήταν ανδρομική και η L. Όμως τότε, C_U καθολική, υπάρχει c τέτοιο ώστε, $K \leqslant C_L + c$. Επίσης $C_L(L(n)) \leqslant n$ εξ' ορισμού. Έτσι

$$2n \leqslant C(L(n)) \leqslant C_L(L(n)) + c \leqslant n + c.$$

άτοπο για $n > c$. ■

Σημείωση 4.4 *Μπορούμε να δούμε τη μη υπολογισιμότητα της $C(x)$, ως μια εκδοχή του Halting problem. Πράγματι, αν μπορούμε να αποφασίσουμε για το $\mathcal{K}$, μπορούμε να υπολογίσουμε την $C(x)$. Ουσιαστικά, αφού ολοκληρώσουμε μια αποδεκτή απαρίθμηση, μπορούμε για κάθε πρόγραμμα e να ελέγξουμε αν η $U(e)$ τερματίζει και αν η απάντηση είναι θετική υπολογίζουμε την τιμή της. Τέλος σε περίπτωση που $U(e) = x$, τερματίζουμε τη διαδικασία δίνοντας στην έξοδο το $len(e)$.*
Μάλιστα και το αντίστροφο ισχύει, αφού μπορούμε βάζοντας το C στο μαντείο να υπολογίσουμε το Halting problem.

Πρόταση 4.19 *Το Halting problem και η πολυπλοκότητα Kolmogorov έχουν ακριβώς τον ίδιο βαθμό Turing.*

Το αποτέλεσμα αυτό υπονοεί, ότι δεν μπορούμε να πούμε με ακρίβεια πόσο τυχαίο είναι ένα αντικείμενο. Ξέρουμε ότι υπάρχουν τέτοιου είδους αντικείμενα ωστόσο δεν μπορούμε να πούμε αν όντως ένα αντικείμενο είναι τυχαίο.

Θεώρημα 4.20 *Το σύνολο $RAND = \{x : C(x) \geqslant len(x)\})$ των τυχαίων αριθμών είναι απρόσβλητο.*

Απόδειξη. Αρχικά, για να δείξουμε ότι το $RAND$ είναι άπειρο, για οποιδήποτε n, παίρνουμε τις συγκλίνουσες τιμές από τα,

$$f_0(0), f_1(0), ..., f_n(0)$$

και αν υπάρχει x διαφορετικό από αυτές, τότε $n + 1 \leqslant C(x)$ από τον ορισμό. Αν το x είναι μάλιστα ο μικρότερος αριθμός που δεν περιέχεται σε αυτές τις τιμές, τότε $x \leqslant n + 1$ (αφού έχουμε $n + 1$ τιμές και στη χειρότερη περίπτωση είναι τιμές από το 0 ως το n). Έτσι $x \leqslant C(x)$ και το x είναι τυχαίο. Έτσι υπάρχει τυχαίο αριθμός με πολυπλοκότητα $n + 1$ και αφού ισχύει για κάθε n, το RAND είναι άπειρο.
Για να δείξουμε ότι το RAND δεν περιέχει άπειρο RE υποσύνολο, αρχικά προσέχουμε ότι ο κωδικός e ενός RE συνόλου $\mathcal{W}_e$ παρέχει μια ομοιόμορφη περιγραφή των στοιχείων του συνόλου και έτσι αν $\mathcal{W}_e = \{x_0, ..., x_n, ...\}$, από το θεώρημα S-m-n, υπάρχει ένα προς ένα αναδρομική h, τέτοια ώστε $f_{h(e,n)}(0) \simeq x_n$. Αν μπορούσαμε να το δείξουμε αυτό για κάποιο n, θα είχαμε $h(e, n) < x_n$ και άρα το x_n δεν θα ήταν τυχαίο. Αυτό ισχύει για αρκετά μεγάλο n και x_n μεγαλύτερο από

$$t(n) = max_{e \leq n} h(e, n)$$

και έτσι το $t(n)$ φράσσει το $h(e, n)$ σχεδόν παντού.
Έτσι δεδομένου του $\mathcal{W}_e$ μπορούμε να πάρουμε ένα RE υποσύνολο $\mathcal{W}_{g(e)}$ του οποίου το n-οστό στοιχείο είναι μεγαλύτερο από $t(n)$, περιμένοντας απλά να παραχθεί ένα τέτοιο από το $\mathcal{W}_e$. Επίσης, αν το $\mathcal{W}_e$ είναι άπειρο, τότε και το $\mathcal{W}_{g(e)}$ είναι άπειρο και περιέχει ένα μη τυχαίο στοιχείο, άτοπο. ■

Σημείωση 4.5 *Μια πιο γρήγορη απόδειξη μπορεί να δωθεί μέσω του θεωρήματος αναδρομής, ως εξής. Έστω A άπειρο RE σύνολο τυχαίων αριθμών και h αναδρομική τέτοια ώστε,*

$$f_{h(e)}(0) \simeq \textit{το μικρότερο } x > e \textit{ που προκύπτει απαριθμώντας το } A.$$

Από το θεώρημα αναδρομής υπάρχει $f_e \simeq f_{h(e)}$. Εξ' ορισμού το $f_e(0)$ δεν μπορεί να είναι μεγαλύτερο από το e αφού ανήκει στο A, όμως έτσι ακριβώς το ορίσαμε, άτοπο.

Το απρόσβλητο των τυχαίων αριθμών είναι ένα πολύ ισχυρό αποτέλεσμα, που υπονοεί όχι μόνο την μη αποκρισιμότητα του Halting problem και ότι το σύνολο των τυχαίων αριθμών είναι RE-πλήρες, αλλα και μια εκδοχή του θεωρήματος μη πληρότητας. Σε οποιονδήποτε

συνεπή φορμαλισμό μπορούμε να αποδείξουμε ότι ένας αριθμός είναι τυχαίος σε πεπερασμένες το πολύ περιπτώσεις (Chaitin, 1974).

Έτσι απαντήσαμε με έμμεσο τρόπο στο ερώτημα που τέθηκε στην αρχή της ενότητας, σχετικά με το κατά πόσο υπάρχει πραγματική τυχαιότητα. Σήμερα, για την παραγωγή τυχαίων αντικειμένων χρησιμοποιούνται hardwired συσκευές που εκμεταλλεύονται την κίνηση των ρευστών. Ωστόσο η ιδιότητά τους αυτή δεν μπορεί θεωρηθεί βέβαιη. Είδαμε σίγουρα ότι τυχαία αντικείμενα υπάρχουν, ωστόσο καμία αλγοριθμική διαδικασία δεν είναι δυνατό να τα παράξει και καμία αλγοριθμική διαδικασία δεν είναι ικανή να επαληθεύσει την ιδιότητα της τυχαιότητας.

Κεφάλαιο 5

Επίλογος

Λίγα λόγια για τις εφαρμογές της πολυπλοκότητας Kolmogorov...

Όπως είδαμε η μαθηματική θεωρία της αλγοριθμικής πολυπλοκότητας περιέχει κάποια πολύ βαθιά και εκλεπτυσμένα μαθηματικά. Η θεμελίωση της βασίστηκε στα κυριότερα αποτελέσματα της θεωρίας υπολογισμού. Ωστόσο οι προαπαιτούμενες γνώσεις που απαιτούνται για την εφαρμογή της σε ένα πλήθος διαφορετικούς τομείς είναι πραγματικά λίγες. Η αλγοριθμική πολυπλοκότητα προέκυψε αρχικά μέσα από τη θεωρία υπολογισμού, ωστόσο η αποδεικτική της ισχύς -που δεν είναι απλά ένα τέχνασμα, όπως ένα επιχείρημα διαγωνοποίησης, αλλά κάτι πολύ πιο ουσιαστικό- απέκτησε γρήγορα μεγάλη σημασία και προσέφερε πρωτοφανή αποτελέσματα, όχι μόνο στην ίδια τη θεωρία υπολογισμού αλλά και σε όλους τους υπόλοιπους τομείς της θεωρίας της επιστήμης των υπολογιστών. Μάλιστα συχνά αποτελεί κλειδί για την προσέγγιση των διάφορων ανοιχτών ζητημάτων της θεωρίας πολυπλκότητας, της θεωρίας γραφημάτων, της ανάλυσης σχεδιασμού αλγορίθμων κ.ο.κ.

Εκτός αυτών, την έννοια αυτή την συναντάμε και σε πλήθος άλλων εφαρμογών σε περισσότερο "ξένα" αντικείμενα. Για παράδειγμα χρήση της έννοιας της δομικής πολυπλοκότητας συναντάμε σε απαντήσεις σε ζητήματα χρηματοοικονομικών, μοντελοποίησης της ανθρώπινης συμπεριφοράς και την ασφάλεια πληροφοριακών συστημάτων. Μάλιστα κάποιοι πιστεύουν ότι υπάρχει -ακόμα μη επαρκώς κατανοητή- σύνδεση με πλήθος τομέων της φυσικής.

Γενικώς, έχουμε να κάνουμε με μία συναρπαστική έννοια που αποτυπώνει πώς η πολυπλοκότητα ενός αντικειμένου μπορεί να περιγραφεί με αυστηρές επιστημονικές μεθόδους. Τέλος, το πεδίο της πολυπλοκότητας Kolmogorov είναι ώριμο και παράλληλα αρκετά δραστήριο ερευνητικά και το ενδιαφέρον που προκαλεί έγκειται όχι μόνο στα αποτελέσματα που προκύπτουν σε τομή με άλλα αντικείμενα, αλλά και στον μεγάλο αριθμό ανοιχτών προβλημάτων που υπάρχουν στη θεωρία μας.

Βιβλιογραφία

[1] S.C.Kleene, *Introduction to Metamathematics*, North-Holland. 1952.

[2] Y.N.Moschovakis, *Θεωρία Αναδρομής*, Διδακτικές σημειώσεις, 2011.

[3] Α.Τζουβάρας, *Θεωρία αναδρομικών συναρτήσεων και υπολογισιμότητα*, Διδακτικές σημειώσεις, 2007.

[4] E.L.Post, *Recursively enumerable sets and their decision problems*, Bull. Am. Math. Soc, 1944.

[5] P.Odifreddi, *Classical recursion theory, The theory of functions and sets of natural numbers*, Elsevier, 1999.

[6] H.Rogers, *Theory of recursive functions and effective computability*, The MIT press, 1987.

[7] N.J.Cutland, *Computability: An introduction to recursive function theory*, Cambridge university press, 1980.

[8] S.G.Simpson, *Studies in logic and foundation of mathematics*, Elsevier, 1977.

[9] S.G.Simpson, *Computability, Unsolvability, and Randomness*, lecture notes, 2007.

[10] R.I.Soare, *Recursively enumerable sets and degrees: A study of computable functions and computably generated sets*, Springer-Verlag, 1987.

[11] R.I.Soare, *Computability and recursion*, The bulletin of symbolic logic, 1996.

[12] A.M.Turing, *On computable numbers, with applications to the Entscheidungsproblem*, Proceedings of the London mathematical society, 1936.

[13] G.Boolos, J.P.Burgees, R.Jeffrey , *Computability and logic*, Harvard university press, 1998.

[14] M.Sipser, *Introduction to the theory of computation*, The MIT press, 1999.

[15] G.Lolli, *Recursion theory and computational complexity*, Springer, 2010.

[16] R.I.Soare, *Computability and recursion*, Bulletin of Symbolic Logic, 1996.

[17] R.I.Soare, *The History and Concept of Computability*, North-Holland, 1999.

[18] F.Stephan, *Recursion Theory*, Lecture notes, 2012.

[19] L.van den Dries, *Recursion Theory Notes*, Lecture Notes 2011.

[20] Y.N.Moschovakis, *Recursion and complexity*, Springer, 2005.

[21] V.Becher, S.Figueira, A.Nies, S.Picchi, P.Vitanyi, *Program size com- plexity for possibly infinite computations*, Notre Dame Journal of Formal Logic, 2005.

[22] C.Calude and H.Jürgensen, *Is complexity a source of incompletenes?*, Advances in Applied Mathematics, 2005.

[23] M.Li and P.Vitanyi, *An introduction to Kolmogorov complexity and its applications*, Springer, 1997.

[24] R.G.Downey and D.R.Hirschfeld , *Algorithmic randomness and complexity*, Springer, 2010.

[25] A.Nies, *Computability and randomness*, Oxford university press, 2009.

[26] G.J.Chaitin, *Algorithmic Information Theory*, Cambridge University Press, 1987.

[27] Y.V.Matiyasevich, *Hilbert's Tenth Problem*, The MIT press, 1993.

[28] M.Ferbus-Zanda and S.Gregorieff, *Kolmogorov complexity in perspective*, Springer-Verlag, 2010.

[29] A.Shen, *Algorithmic information theory and kolmogorov complexity*, Lecture notes, 2007.

[30] R.G.Downey and D.R.Hirschfeldt, *Randomness and Reducibility*, Elsevier, 2004.

[31] L.Fortnow, *Kolmogorov Complexity*, Gruyter Series in Logic and Its Applications. de Gruyter, 2000.

[32] L.Fortnow, *Kolmogorov Complexity and Computational Complexity*, Quaderni di Matematica, 2004.

[33] M.Li and P.Vitanyi *Algorithms and Complexity*, Elsevier, 1990.

[34] G.Chaitin, *META MATH!*, Vintage, 2005.