Αναδρομή και Μηχανές Turing

Δημάκης Δημήτριος
Σχολή Εφαμορμένων Μαθηματικών και Φυσικών Επιστημών

28 Μαΐου 2019

Διπλωματική Εργασία
Επιβλέπων:Αρβανιτάκης Αλέξανδρος-Επίκουρος Καθηγήτης Ε.Μ.Π.

Τριμελής Εξεταστική Επιτροπή:
1.Αρβανιτάκης Αλέξανδρος(Επίκουρος Καθηγητής)
2.Στεφανέας Πέτρος (Επίκουρος Καθηγήτης)
3.Κανελλόπουλος Βασίλειος (Αναπληρωτής Καθηγητής)

Περιεχόμενα

1 Η διαισθητική προσέγγιση του αλγορίθμου **6**
1.1 Αλγοριθμικές συναρτήσεις και σύνολα 6
1.2 Κωδικοποιήση-Αποκωδικοποιήση Συνόλων 14

2 Οι Αλγοριθμικές Συναρτήσεις μέσω Αναδρομής **17**
2.1 Γενικά περί Αναδρομής 17
2.2 Βασικές αναδρομικές συναρτήσεις ή Πρωτογενής Αναδρομικές(primitive recursive fuction) 17
2.3 Βασικά Αναδρομικά Σύνολα 21
2.4 Πέρα από τις βασικές συναρτήσεις.Η συνάρτηση Ackermann 23
2.5 Αναδρομικές συναρτήσεις 27
2.6 Αναδρομικά και αναδρομικά απαριθμήσιμα σύνολα 27
2.7 Αριθμητικοποίηση αναδρομικών Συναρτήσεων. Κανονική μορφή Kleene. 29

3 Turing Machines **33**
3.1 Γενική περιγραφή των μηχανών Turing 33
3.2 Turing υπολογίσιμες συναρτήσεις 38

4 Πρόβλημα Τερματισμού **46**

Εισαγωγή

Η Αναδρομική θεωρία προέρχεται από τη δεκαετία του 1930, με το έργο του Kurt Gödel, Alonzo Church, Alan Turing, Stephen Kleene.

Τα θεμελιώδη αποτελέσματα που αποκόμισαν οι ερευνητές εγκαθίδρυσαν την αναδιαρθρωτική υπολογισιμότητα ως σωστή επισημοποίηση της άτυπης ιδέας του αποτελεσματικού υπολογισμού και οδήγησαν τον Stephen Kleene (1952) για να πλάσει τα δύο ονόματα «Church's thesis» (Kleene 1952:300) και «Turing's Thesis» (Kleene 1952:376). Σήμερα αυτά συχνά θεωρούνται ως μια ενιαία υπόθεση η Church–Turing thesis η οποία ορίζει ότι κάθε λειτουργία που είναι υπολογίσιμη από εναν αλγόριθμο είναι μια υπολογίσιμη συνάρτηση. Αν και αρχικά σκεπτικός, από το 1946 ο Gödel τάχθηκε υπέρ αυτής της διατριβής:

«Ο Tarski τόνισε στην ομιλία του τη μεγάλη σημασία της έννοιας της γενικής αναδρομής (ή του υπολογιστικού περιβάλλοντος του Turing). Μου φαίνεται ότι η σημασία αυτή σε μεγάλο βαθμό οφείλεται στο γεγονός ότι με αυτήν την έννοια για πρώτη φορά κατόρθωσε κάποιος να δώσει μια απόλυτη έννοια σε μια ενδιαφέρουσα επιστημολογική αντίληψη, δηλαδή χωρίς να εξαρτάται από τον φορμαλισμό που επικρατούσε. (Gödel 1946 στο Davis 1965:84)».

Με τον ορισμό του αποτελεσματικού υπολογισμού ήρθαν οι πρώτες αποδείξεις ότι υπάρχουν προβλήματα στα μαθηματικά που δεν μπορούν να αποφασιστούν αποτελεσματικά. Ο Church (1936a, 1936b) και ο Turing (1936), εμπνευσμένοι από τις τεχνικές που χρησιμοποιούνταν από τον Gödel (1931) για να αποδείξουν τη μη πληρότητα των θεωρημάτων του, ανεξάρτητα κατέδειξαν ότι το Entscheidungs problem δεν λύνεται αποτελεσματικά. Το αποτέλεσμα έδειξε ότι δεν υπάρχει αλγοριθμική διαδικασία που μπορεί σωστά να αποφασίσει αν κάποια αυθαίρετη μαθηματική πρόταση είναι αληθής ή ψευδής.

Πολλά προβλήματα των μαθηματικών έχει αποδειχθεί ότι είναι άλυτα αφότου αυτά τα αρχικά παραδείγματα καθιερώθηκαν. Στο παρών σύγγραμα θα ασχοληθούμε με την αλγοριθμικότητα συνάρτησεων και συνόλων και πως αυτή μπορεί να υπολογιστεί μέσω αναδρομικών συνάρτησεων και μετά θα δείξουμε πως οι μηχανές Turing είναι ισοδύναμο μοντέλο με τις αναδρομικές συναρτήσεις.

Introduction

Recursion theory was established in 30s by the work of Kurt Gödel, Alonzo Church, Alan Turing and Stephen Kleene.

The fundamental results presented by those mathematicians gave restructuring computability the status of the solemn definition of the vague concept of efficient computation and led Stephen Kleene (1952) to name Church's and Turing's work as «Church's Thesis» (Kleene 1952:300) and «Turing's Thesis» respectively (1952:376). Today those two are perceived as a single solid thesis, the Church- Turing thesis, which defines that every process computable by an algorithm is a computable function. At 1946 Kurt Gödel sided with this approach, denouncing his skeptical scope which he had at first:

« Tarski emphasized in his speech the great importance of the concept of general recursion (or Turing's computational environment). As it seems that importance is based on the fact that for the first time someone made it possible to produce an absolute notion in regard of an interesting scientific belief, free of the formalism that was dictated. (Gödel to Davies 1965:84) »

With the definition of efficient computation came the first proof that some mathematic problems cannot be efficiently decidable. Church (1936a, 1936b) and Turing (1931) inspired by Gödel's techniques (1931) that were used to demonstrate the incompleteness of his theorems,independently showed that the Entscheindungs problem is not effectively solved.Based on the results there is none algorithmic procedure that can truly decide if a random mathematical proposition is true or false.

Many mathematical problems are proven to be unsolvable ever since those examples were first stated. At this thesis we will deal with the notion of when a function or set can be described by an algorithm, how it can be calculated by recursive functions and then we will demonstrate how Turing Machines can be a model equivalent model to recursive functions.

1 Η διαισθητική προσέγγιση του αλγορίθμου

1.1 Αλγοριθμικές συναρτήσεις και σύνολα

Αλγόριθμος είναι μια πεπερασμένη σειρά ενεργειών,αυστηρά καθορισμένων και εκτελέσιμων σε πεπερασμένο χρόνο, με σκοπό την επίλυση ενός προβλήματος.Ένας αλγόριθμος τροφοδοτείται με τιμές δεδομένων από ένα καλά ορισμένο σύνολο αντικειμένων πεπερασμένου τύπου. Αυτές οι τιμές λέγονται είσοδοι (inputs). Αφού δοθεί μία είσοδο σε έναν αλγόριθμο αυτός μέσω μιας πεπερασμένης διαδικασίας δίνει μια το πολύ μια έξοδο(output), η οποία είναι η απάντηση στο πρόβλημα και είναι επίσης πεπερασμένου τύπου. Λέμε το πολύ μία για να συμπεριλάβουμε την περίπτωση που ο αλγόριθμος δεν δίνει έξοδο στην συγκεκριμένη είσοδο. Αν ένας αλγόριθμος δίνει έξοδο για κάθε είσοδο, θα λέγεται ολικός αλγόριθμος(total). Οι ολικοί αλγόριθμοι δηλαδή είναι μόνο ένα κομμάτι των αλγορίθμων.

Αλγόριθμος μπορεί να είναι μια συνταγή μαγειρικής ή βήμα προς βήμα η επίλυση ενός μαθηματικού προβλήματος. Πλέον με την επιστήμη των υπολογιστών τόσο διαδεδομένη και επιδραστική στην καθημερινότητα του ανθρώπου, η πληρέστερη εικόνα ενός αλγορίθμου είναι αυτή ενός προγράμματος υπολογιστή. Παρακάτω θα προσπαθήσουμε να μελετήσουμε την συμπεριφορά των αλγορίθμων.

Θα παριστάνουμε τους αλγορίθμους με τα κεφαλαία γράμματα A, B, Γ κλπ, και με τα γράμματα X, Y, Z κλπ, κάποια υποσύνολα του $\mathbb{N}$ ή του $\mathbb{N}^k$. Τώρα για έναν αλγόριθμο A θα συμβολίζουμε με $in(A)$ την είσοδο του A ενώ με $out(A)$, την έξοδο του αλγορίθμου. Μπορούμε να υποθέσουμε ότι $in(A) = \mathbb{N}^k$ και $out(A) \subseteq \mathbb{N}^l$, για κάποια $k, l \geq 0$.Αν $in(A) = \mathbb{N}^k$ και $\vec{x} \in in(A)$, ο συμβολισμός $A(\vec{x}) \downarrow$ σημαίνει ότι ο A σταματά και δίνει έξοδο όταν έχει ως είσοδο το $\vec{x}$. Αν η έξοδος είναι ένα $\vec{y} \in out(A) \subseteq \mathbb{N}^l$ τότε θα λέμε ότι $A(\vec{x}) = \vec{y}$.Αν ο A δεν δίνει έξοδο στο $\vec{x}$ θα γράφουμε ότι $A(\vec{x}) \uparrow$. Ο A λέγεται ολικός όταν για κάθε $x \in \mathbb{N}^k, A(\vec{x}) \downarrow$.

Σε έναν αλγόριθμο σε κάθε input έχουμε το πολύ ένα output, προκύπτει ότι ο A με $in(A) = \mathbb{N}^k$ και $out(A) \subseteq \mathbb{N}^l$, παράγει μια συνάρτηση $f : \mathbb{N}^k \to \mathbb{N}^l$. Η διαφορά από τις συνηθισμένες συναρτήσεις είναι ότι για κάποια $\vec{x} \in \mathbb{N}^k$ μπορεί να μην ορίζεται αυτό σημαίνει ότι έχουμε να κάνουμε με μερικές συναρτήσεις. Άρα όταν έχουμε μια συνάρτηση $f : \mathbb{N}^k \to \mathbb{N}^l$ θα εννοούμε μερική συνάρτηση, δηλαδή για κάθε $\vec{x} \in \mathbb{N}^k$ έχουμε το πολύ μια αντιστοιχία σε ένα $\vec{y} \in \mathbb{N}^l$. Π.χ. η συνάρτηση $f : \mathbb{N} \to \mathbb{N}$ όπου $f(n) = \sqrt{n}$ είναι μερική.

Το σύνολο X που ορίζεται η $f(\vec{x})$ είναι το πεδιό ορισμού της f και συμβολίζεται $dom(f)$. Το πεδίο τιμών της f συμβολίζεται $rng(f)$.Γενικά ισχύει ότι $dom(f) \subseteq \mathbb{N}^k$ (καθώς έχουμε να κάνουμε με μερικές συναρτήσεις) αν $dom(f) = \mathbb{N}^k$, τότε προφανώς η f είναι ολική. Όσο αφορά την ισότητα των μερικών συναρτήσεων ορίζεται ως εξής: Οι $f, g : \mathbb{N}^k \to \mathbb{N}^l$ είναι ίσες αν $dom(f) = dom(g)$ και για κάθε $\vec{x} \in dom(f), f(\vec{x}) = g(\vec{x})$.

Ορισμος 1.1.1. *Έστω συνάρτηση $f : \mathbb{N}^k \to \mathbb{N}^l$. Η f θα λέγεται αλγοριθμική(computable) αν υπάρχει αλγόριθμος A τέτοιος ώστε κάθε $\vec{x} \in \mathbb{N}^k, \vec{x} \in dom(f) \iff A(\vec{x}) \downarrow$, και για κάθε $\vec{x} \in dom(f), f(\vec{x}) = A(\vec{x})$. Η f λέγεται ολική αλγοριθμική όταν είναι αλγοριθμική και $dom(f) = \mathbb{N}^k$, δηλαδή όταν ο A είναι ολικός.*

Ο αλγόριθμος δεν ταυτίζεται με την συνάρτηση που παράτει καθώς υπάρχουν άπειροι διαφορετικοί αλγόριθμοι που παράγουν την ίδια συνάρτηση.
Μια σημαντική κατηγορία αλγόριθμων είναι οι ναι/όχι αλγόριθμοι, αυτοί δίνουν έξοδο ναι ή όχι και ουσιαστικά μας βοηθούν να απαντήσουμε σε ερωτήματα τύπου $x \in X$; όπου X κάποιο σύνολο.Προφανώς έχουμε $out(A) \subseteq \{\text{ναι ή οχι}\}$, θα γράφουμε 1 αντί για "ναι" και 0 αντί για "όχι", δηλαδή έχουμε $out(A) \subseteq \{0, 1\}$, προφανώς οι ναι/όχι αλγόριθμοι είναι απλώς μια κατηγορία των αλγοριθμών όπως ορίστηκαν παραπάνω.

Ορισμος 1.1.2. *Ένα σύνολο $X \subseteq \mathbb{N}^k$ λέγεται αλγοριθμικό(decidable) αν υπάρχει ολικός ναι/όχι αλγόριθμος A για το ερώτημα $x \in X$; δηλαδή για κάθε $\vec{x} \in \mathbb{N}^k$,*

$$\vec{x} \in X \Longrightarrow A(\vec{x}) = \textit{ναι}$$

και

$$\vec{x} \notin X \Longrightarrow A(\vec{x}) = \textit{όχι}$$

Έστω V ένα βασικό συνόλο π.χ. $V = \mathbb{N}^k$, και $X \subseteq V$. Χαρακτηριστική συνάρτηση του X λέγεται η συνάρτηση $C_X : V \to \{0, 1\}$ που ορίζεται ως εξής:

$$C_X = \begin{cases} 1, & \text{αν } x \in X \\ 0, & \text{αν } x \notin X \end{cases}$$

Προφανώς κάθε χαρακτηριστική συνάρτηση είναι ολική. Η παρακάτω πρόταση είναι προφανής.

Πρόταση 1.1.3. *Ένα σύνολο $X \subseteq \mathbb{N}^k$ είναι αλγοριθμικό αν και μόνο αν η χαρακτηριστική του συνάρτηση είναι αλγοριθμική.*

Πρόταση 1.1.4. *Ένα σύνολο $X \subseteq \mathbb{N}^k$ είναι αλγοριθμικό αν και μόνο αν το $-X(\mathbb{N}^k - X)$ είναι αλγοριθμικό.*

Απόδειξη. Έστω Α ο ολικός ναι/όχι αλγόριθμος για το X, τότε ο αλγόριθμος Β ο οποίος δίνει "ναι" εκεί που ο Α δίνει "όχι", και "όχι" εκεί που ο Α δίνει "ναι" είναι ολικός αλγόριθμος για το $-X$. □

Παραδείγματα Αλγοριθμικών Συναρτήσεων

(1) Η σταθερή συνάρτηση $f(\vec{x}) = c$ για κάθε $\vec{x} \in \mathbb{N}^k$ είναι αλγοριθμική. Η συνάρτηση διαδοχής των φυσικών αριθμών $S(n) = n + 1$ είναι αλγοριθμική. Το ίδιο η πρόσθεση και ο πολλαπλασιασμός. Για κάθε n και κάθε $m \leq n$, ορίζονται οι προβολές $\Pi_{nm} : \mathbb{N}^n \to \mathbb{N}$, όπου $\Pi_{nm}(x_1, \cdots, x_n) = x_m$, οι οποίες είναι αλγοριθμικές.

(2) Κάθε πεπερασμένο σύνολο είναι αλγοριθμικό. Πράγματι έστω $X = \{\vec{x_1}, \cdots, \vec{x_n}\}$ Για το ερώτημα $\vec{x} \in X$; ο αλγόριθμος είναι: Εξέτασε βήμα-βήμα αν $\vec{x} = \vec{x_1}, \cdots, \vec{x} = \vec{x_n}$. Επειδή τα $\vec{x}, \vec{x_i}$ είναι πεπερασμένου τύπου παέι να πει ότι ο έλεγχος $\vec{x} = \vec{x_i}$ γίνεται σε πεπερασμένο χρόνο.

(3) Έστω $g : \mathbb{N}^{k+1} \to \mathbb{N}$ μια ολική συνάρτηση $k+1$ μεταβλητών. Γράφουμε $g(\vec{x}, y)$ αντί $g(x_1, \cdots, x_k, y)$. Αν δοθέντος $\vec{x}$, υπάρχει y τέτοιο ώστε π.χ. $g(\vec{x}, y) = 0$, προφανώς θα υπάρχει ένα τέτοιο ελάχιστο y, έστω y_0. Συμβολίζουμε τότε

$$y_0 = (\mu y)(g(\vec{x}, y) = 0)$$

Ο τελεστής μ λέγεται τελεστής ελαχιστοποιήσης(minimization) ή μ-τελεστής. Ορίζεται τότε η εξής (μερική εν γένει) συνάρτηση $f : \mathbb{N}^k \to \mathbb{N}$:

$$f(\vec{x}) = \begin{cases} (\mu y)(g(\vec{x}, y) = 0) \ \text{ αν } \ (\exists y)(g(\vec{x}, y) = 0), \\ \text{δεν ορίζεται αλλιώς} \end{cases}$$

Αν η g είναι ολική αλγοριθμική με ολικό αλγόριθμο Α, τότε και η f είναι αλγοριθμική(όχι κατάνάγκη ολική). Ο αλγόριθμος B για την f είναι: δοθέντος $\vec{x} \in \mathbb{N}^k$, τροφοδοτούμε διαδοχικκά τον Α με τις εισόδους $(\vec{x}, 0), (\vec{x}, 1), \cdots$. Λόγω της ολικότητας του Α για κάθε είσοδο $(\vec{x}, k)$ θα παίρνουμε μια έξοδο. Για το πρώτο k που θα βρούμε (αν υπάρχει) τέτοιο ώστε $A(\vec{x}, k) = 0$ θέτουμε $B(\vec{x}) = k$. Αν για κάθε $k \in \mathbb{N}. A(\vec{x}, k) \neq 0, B(\vec{x}) \uparrow$. Προφανώς ο B είναι αλγόριθμος για την f. Η ιδιότητα $g(\vec{x}, y) = 0$ λέγεται κανονική αν $(\forall \vec{x})(\exists y)(g(\vec{x}, y) = 0)$. Αν η $g(\vec{x}, y) = 0$ είναι κανονική τότε η f είναι ολική αλλιώς δεν ορίζεται σε κάποια σημεία.

(4) Ένα άλλο παράδειγμα μερικής αλγοριθμικής συνάρτησης είναι η συνάρτηση που δεν ορίζεται πουθενά, έχει δηλαδή πεδίο ορισμού το 0.Έστω Ω η παραπάνω συνάρτηση και την ορίζουμε ως εξής $\Omega(x) = (\mu y)(x + y + 1 = 0)$.Αυτός είναι ένας από τους πολλούς τρόπους που μπορούμε να ορίσουμε την Ω.

Παρατηρούμε ότι η διαφορά του αλγοριθμικού συνόλου με την αλγοριθμική συνάρτηση είναι είναι οτι η αλγοριθμική συνάρτηση απαιτεί απλώς την ύπαρξη ενός αλγορίθμου(όχι απαραίτητα ολικού), ενώ για το αλγοριθμικό σύνολο θέλουμε έναν ναι/όχι ολικό αλγόριθμο ή ισοδύναμα έναν ολικό αλγόριθμο για την χαρακτηριστική του συνάρτηση. Γενικότερα οι συναρτήσεις είναι ένα σύνολο ζευγών και δεχόμαστε ότι για κάθε συνάρτηση $f = \{(x, y) : y = f(x)\}$, αυτή είναι η εκτατική έννοια της συνάρτησης, από την σκοπιά των αλγορίθων όμως δεν ισχύει. Εδώ η συνάρτηση f δεν είναι σύνολο αλλά μια διαδικάσία αντιστοίχισης μιας εξόδους $f(x)$ σε μια έξοδο x, που καθορίζεται από έναν νόμο (αλγόριθμο). Έφαση δίνεται στην "περιγραφή" του νόμου αντιστοιχίας.Για κάθε f ορίζεται ένα σύνολο $\{(x, y) : y = f(x)\}$, όμως δεν το ταυτίζουμε με την f. Αυτό λέγεται γράφημα της f και το συμβολίζουμε με $G(f)$. Δηλαδή σε κάθε f αντιστοιχεί το σύνολο

$$G(f) = \{(x, y) : y = f(x)\}$$

Παρακάτω θα δούμε την σχέση αλγοριθμικότητας της f με το $G(f)$. Για απλότητα στα επόμενα θα θεωρούμε συναρτήσεις $f : \mathbb{N} \to \mathbb{N}$ αντί $f : \mathbb{N}^k \to \mathbb{N}^l$. Αυτό σε τίποτα δεν βλάπτει την γενικότητα και τα ίδια ισχύουν για τη γενικότερη περίπτωση.

Πρόταση 1.1.5. *Έστω $f : \mathbb{N} \to \mathbb{N}$. (i) Αν το $G(f)$ είναι αλγοριθμικό, η f είναι αλγοριθμική.Το αντίστροφο δεν ισχύει. (ii) Αν όμως η f είναι ολική τότε ισχύει*

και το αντίστροφο του (i)

$$f \ \ \textit{αλγοριθμική} \Longleftrightarrow G(f) \ \textit{αλγοριθμικό}$$

Απόδειξη. (i) Έστω $G(f)$ αλγοριθμικό με αλγόριθμο B. Έστω ο αλγόριθμος Α:

$$A(m) = \begin{cases} n, & \text{αν } B(m,n) = \text{ναι}, \\ \text{δεν ορίζεται αλλιώς} \end{cases}$$

Ο Α δουλεύει ως εξής: Δοθέντος m, εξετάζουμε διαδοχικά τις εξόδους $B(m,o)$, $B(m,1)\cdots$. Για το πρώτο n που θα βρούμε(αν υπάρχει) ώστε $B(m,n) = \text{ναι}$, θέτουμε $A(m) = n$. Αν για κάποιο m δεν υπάρχει n ώστε $B(m,n) = \text{ναι}$, τότε ο Α δεν είναι ολικός. Ο Α είναι αλγόριθμος για την f,έχουμε:
$f(m)\downarrow$ και $f(m) = n \Longleftrightarrow (m,n) \in G(f) \Longleftrightarrow B(m,n) = \text{ναι} \Longleftrightarrow A(m)\downarrow$ και $A(m) = n$. Άρα η f είναι αλγοριθμική.
(ii) Τώρα έστω f ολική συνάρτηση. Έστω Α ολικός αλγόριθμος για την f, δηλαδή $f(m) = n \Leftrightarrow A(m) = n$. Θεωρούμε B με $in(B) \subseteq \mathbb{N}^2$ τον εξής αλγόριθμο :

$$B(m,n) = \begin{cases} \text{ναι}, & \text{αν } A(m) = n, \\ \text{όχι}, & \text{αν } A(m) \neq n. \end{cases}$$

Αφού ο Α είναι ολικός αλγόριθμος τότε και ο B είναι αναγκαστικά ολικός και προφανώς έχουμε $B(m,n) = \text{ναι} \Leftrightarrow (m,n) \in G(f)$, άρα $G(f)$ αλγοριθμικό. Εδώ παρατηρούμε ότι αν η f δεν ήταν ολική τότε και ο Α δεν θα ήταν ολικός και κατά συνέπεια ούτε ο B οπότε δεν μπορούμε να βγάλουμε κάποιο συμπέρασμα για την αλγοριθμικότητα του $G(f)$. □

Ορισμος 1.1.6. *Ένα σύνολο $X \subseteq \mathbb{N}^k$ λέγεται αλγοριθμικά απαριθμήσιμο ή για συντομία α.α.(effectively enumerable ή listable), αν υπάρχει ναι/όχι αλγόριθμος Α (όχι κατάναγκη ολικός) τέτοιος ώστε για κάθε $\vec{x} \in \mathbb{N}^k$,*

$$\vec{x} \in X \Leftrightarrow A(\vec{x})\downarrow \textit{ και } A(\vec{x}) = \textit{ναι}$$

Παρατηρούμε ότι από την παραπάνω ισοδυναμία αν $\vec{x} \notin X$ δεν συνεπάγεται $A(\vec{x}) = $ όχι, αλλά $A(\vec{x})\uparrow$ ή $A(\vec{x}) = $ όχι. Από τον ορισμό 1.1.2 προκύπτει ότι κάθε αλγοριθμικό σύνολο είναι α.α. Η διαφορά ενός α.α. συνόλου με ένα αλγοριθμικό σύνολο είναι ότι στο αλγοριθμικό σύνολο έχουμε την απαίτηση ο αλγόριθμος Α να μας απαντάσει "όχι" αν το $\vec{x} \notin X$, ενώ στο α.α. δεν μας ενδιαφέρει καθώς αν $\vec{x} \notin X$ μπορεί ο Α να μην δίνει απάντηση.

Πρόταση 1.1.7. *Το $X \subseteq \mathbb{N}^k$ είναι αλγοριθμικό αν και μόνο αν το X και το $-X$ είναι α.α.*

Απόδειξη. Αν το X είναι αλγοριθμικό τότε 1.1.4 και το $-X$ είναι αλγοριθμικό άρα τα $X, -X$ είναι α.α. Αντίστροφα έστω τα $X, -X$ είναι α.α. με αλγορίθμους A, B αντίστοιχα. Θεωρούμε τον αλγόριθμο Γ :

$$\Gamma(\vec{x}) = \begin{cases} \text{ναι}, & \text{αν } A(\vec{x}) = \text{ναι}, \\ \text{όχι}, & \text{αν } B(\vec{x}) = \text{ναι} \end{cases}$$

Βλέπουμε εύκολα ότι ο Γ είναι ολικός αλγόριθμος για το X. □

Μια βασική έννοια στους αλγορίθμους είναι ο χρόνος αναμονής, δηλαδή ο χρόνος που περιμένουμε να πάρουμε έξοδο σε μια είσοδο. Αν ο αλγόριθμος είναι ένα πρόγραμμα, τότε ο χρόνος αναμονής για την έξοδο $A(\vec{x})$, είναι ο αριθμός των βημάτων που κάνει το πρόγραμμα μέχρι να υπολογίσει το $A(\vec{x})$ (υποθέτουμε ότι κάνει ένα βήμα ανα μονάδα χρόνου). Σε κάθε αλγόριθμο Α με $in(A) = \mathbb{N}^k$ και $out(A) \subseteq \mathbb{N}^l$, αντιστοιχεί ένας ολικός ναι/όχι αλγόριθμος T_A με $in(T_A) \subseteq \mathbb{N}^{n+l+1}$ που ορίζεται ως εξής :

$$T_A(\vec{x}, \vec{y}, z) = \begin{cases} \text{ναι, αν } A(\vec{x}) = y \text{ σε } \leq z \text{ βήματα,} \\ \text{όχι αλλιώς} \end{cases}$$

Όπου η έκφραση "$A(\vec{x}) = y$ σε $\leq z$ βήματα" σημαίνει: Ο αλγόριθμος Α με είσοδο $\vec{x}$ δίνει έξοδο $\vec{y}$ το πολύ σε z βήματα. Ο T_A λέγεται χρόνος αναμονής του Α. Ο T_A είναι φανερό ότι είναι ολικός από τον ορισμό του(ακόμα και όταν ο Α δεν είναι). Σε κάθε είσοδο $(\vec{x}, \vec{y}, z)$ του T_A, δίνουμε είσοδο στον Α το $\vec{x}$ και περιμένουμε το πολύ z μονάδες χρόνου. Αν σε αυτό το διάστημα δεν έχει έρθει απάντηση ή έρθει και είναι διάφορη του $\vec{y}$ ο T_A δίνει απάντηση "όχι",αλλιώς απαντά "ναι". Προφανώς ισχύει για κάθε $\vec{x}$

$$A(\vec{x}) = \vec{y} \Longleftrightarrow (\exists z)(T_A(\vec{x}, \vec{y}, z) = \text{ναι}). \quad (1)$$

Αν ο Α είναι ναι/όχι αλγόριθμος ο T_A γράφεται απλούστερα :

$$T_A(\vec{x}, z) = \begin{cases} \text{ναι, αν } A(\vec{x}) = \text{ναι σε } \leq z \text{ βήματα,} \\ \text{όχι αλλιώς} \end{cases}$$

Οπότε η (1) γίνεται

$$A(\vec{x}) = \text{ναι} \Longleftrightarrow (\exists z)(T_A(\vec{x}, z) = \text{ναι}). \quad (2)$$

Πρόταση 1.1.8. *Το $X \subseteq \mathbb{N}^k$ είναι α.α. αν και μόνο αν είναι πεδίο τιμών μιας ολικής αλγοριθμικής συνάρτησης $f : \mathbb{N} \to \mathbb{N}^k$.*

Απόδειξη. Για απλότητα έστω $X \subseteq \mathbb{N}$(η απόδειξη για $\mathbb{N}^k$ δεν έχει καμμιά διαφορά). Έστω $X = rng(f)$ όπου f ολική αλγοριθμική και $f : \mathbb{N} \to \mathbb{N}$. Αλλιώς έχουμε $X = \{f(0), f(1), \cdots\}$. Ο αλγόριθμος Α για το X είναι ο εξής :Δοθέντος $x \in \mathbb{N}$, έλεγξε αν $x = f(0)$, αν $x = f(1)$,$\cdots$,$x = f(n)$,$\cdots$, επειδή η f είναι ολική αλγοριθμική το $f(n)$ υπολογίζεται πάντα. Αν για κάποιο $n, f(n) = x$ ο Α απαντά "ναι", αλλιώς δεν απαντά.Ο $A(x)$ γράφεται ως εξής :

$$A(x) = \begin{cases} \text{ναι, αν } (\exists n)(x = f(n)) \\ \text{δεν ορίζεται αλλιώς} \end{cases}$$

Προφανώς $x \in X \Longleftrightarrow A(x) \downarrow$ και $A(x) = \text{ναι}$.. Άρα το X είναι α.α.
Για το αντίστροφο τώρα, έστω $X \subseteq \mathbb{N}$ α.α. και Α ένας αλγόριθμος του. Θέλουμε να βρούμε ολική αλγοριθμική $f : \mathbb{N} \to \mathbb{N}$ τέτοια ώστε $X = rng(f)$. Ένας τρόπος είναι ο εξής να εισάγουμε ένα-ένα τα $0,1,\cdots n,\cdots$ στον Α, και να πάρουμε τις εξόδους $A(0), A(1), \cdots, A(n), \cdots$. Κάθε φορά που έρχεται μια έξοδος $A(k) = \text{ναι}$, με τη σειρά που έρχεται βάζουμε το k στην αντίστοιχη θέση μιας

ακολουθίας $x_0, x_1, \cdots, x_n, \cdots$. Οπότε αν $f(n) = x_n, X = rng(f)$ και f ολική αλγοριθμική.Απλά εδώ έχουμε το πρόβλημα ότι αν για κάποιο $k \in \mathbb{N}$ μπορεί να μην πάρουμε ποτέ απάντηση οπότε δεν ξέρουμε γενικά πόσο χρόνο θα περιμένουμε για κάθε $k \in \mathbb{N}$. Γιάυτό θα χρησιμοποιήσουμε τον αλγόριθμο αναμονής του A,T_A ο οποίος είναι ο εξής $T_A(m, n) =$ ναι σημαίνει ότι $A(m) =$ ναι το πολυ σε n βήματα.Τροφοδοτούμε τον T_A με όλα τα ζεύγη (m, n) της μορφής $m \leq n$ με την εξής σειρά: $(0, 0), (0, 1), (1, 1), (0, 2), (1, 2), (2, 2)$, κλπ.Κάθε φορά που για το ζεύγος (m, n) ο T_A δίνει απάντηση "ναι", βάζουμε το m στην αντίστοιχη θέση της σχηματιζόμενης ακολουθίας $x_0, x_1, \cdots$. Έτσι αν το $m \in X$,όσο χρόνο και να πάρει για την απάντηση του $A(m)$, ο αλγόριθμος θα το συλλάβει. Αν η απάντηση έρχεται σε χρόνο k και $k \geq m$, το ζεύγος $(m, k) \in in(T_A)$ και $T_A(m, n) =$ ναι. Αν $k < m$,τότε $T_A(m, m) =$ ναι(γιατί η απάντηση έρχεται το πολύ σε $\leq m$ βήματα). Άρα σε κάθε περίπτωση το m θα συμπεριληφθεί στα στοιχεία της ακολουθίας. Συνεπώς $X = \{x_0, x_1, \cdots\}$ και αν $f(n) = x_n, f$ ολική αλγοριθμική και $X = rng(f)$.Δηλαδή η f ορίζεται επαγωγικά ως εξής :$f(n) = m$, αν υπάρχει k τέτοιο ώστε $T_A(m, k) =$ ναι και το ζεύγος (m, n) είναι n-οστό μ' αυτήν την ιδιότητα στην παραπάνω διάταξη των ζευγών. □

Πρόταση 1.1.9. *Έστω $f : \mathbb{N} \to \mathbb{N}$. Η f είναι αλγοριθμική αν και μόνον αν το $G(f)$ είναι α.α.*

Απόδειξη. Έστω $f : \mathbb{N} \to \mathbb{N}$ αλγοριθμική και έστω Α ένας αλγόριθμος της. Και B ο εξής αλγόριθμος:

$$B(m, n) = \begin{cases} \text{ναι, αν } A(m)\downarrow \text{ και } A(m) = n \\ \text{δεν ορίζεται αλλιώς} \end{cases}$$

Προφανώς ο B είναι αλγόριθμος για το $G(f)$, δηλαδή $(m, n) \in G(f) \iff B(m, n)\downarrow$ και $B(m, n) =$ ναι. Άρα $G(f)$ α.α.
Το αντίστροφο τώρα, έστω B αλγόριθμος για το $G(f)$ και T_B ο αλγόριθμος αναμονής του B, με εισόδους $((m, n), k)$.Τότε

$$f(m) = \begin{cases} n, \text{αν } (\exists k)(T_B((m, n), k) = \text{ναι}), \\ \text{δεν ορίζεται αλλιώς} \end{cases}$$

Άρα ένας αλγόριθμος Α για την f είναι ο εξής:

$$A(m) = \begin{cases} n, \text{αν } (\exists k)(T_B((m, n), k) = \text{ναι}), \\ \text{δεν ορίζεται αλλιώς} \end{cases}$$

Ο Α δουλεύει ως εξής: Έστω $(0, 0), (0, 1), (1, 1)$ κλπ η διάταξη των ζευγών (n, k) με $n \leq k$. Για κάθε $m \in \mathbb{N}$, εισάγουμε στον T_B μία-μία τις εισόδους $(m, 0, 0), (m, 0, 1), (m, 1, 1)$ κλπ. Για το πρώτο (n, k) για το οποίο $T_B((m, n), k)\downarrow$ και $T_B((m, n), k) =$ ναι, γράφουμε $A(m) = n$.Αλλιώς $A(m)\uparrow$. Προφανώς ο Α είναι αλγόριθμος για την f. □

Πρόταση 1.1.10. *Έστω $X \subseteq \mathbb{N}$ άπειρο. Τα παρακάτω είναι ισοδύνα:*
(a) Το X είναι α.α.

(β) Το X είναι πεδίο τιμών μιας αλγοριθμικής συνάρτησης.
(γ) Το X είναι πεδίο ορισμού μιας αλγοριθμικής συνάρτησης.
(δ) Το X είναι πεδίο τιμών μιας ολικής 1-1 αλγοριθμικής συνάρτησης.

Απόδειξη. (α)⇒(β): Συνέπεια από το 1.1.8

(β) ⇒(γ): Έστω $X = rng(f)$, όπου f αλγοριθμική με αλγόριθμο Α. Θέλουμε να βρούμε έναν αλγόριθμο B όπου για $A(n) = m$, να έχουμε $B(m) = n$. Θα χρησιμοποιήσουμε τον αλγόριθμο αναμονής του Α T_A. Ο B είναι ο εξής αλγόριθμος: Δοθέντος m, τροφοδοτούμε τον T_A διαδοχικά με τις τριάδες (n, m, k) για όλα τα ζεύγη (n, k) όπου $n \leq k$, τα οποία διατεταγμένα ως εξής $(0, 0), (0, 1), (1, 1)$ κλπ ας ονομάσουμε αυτή τη διάταξη $\leq_{DT}$. Αν (n, m) το ελάχιστο ζεύγος στην παραπάνω διάταξη ώστε $T_A(n, m, k) =$ ναι(αν υπάρχει τέτοιο ζεύγος), θέτουμε $B(m) = n$, αλλιώς $B(m) \uparrow$. Ο Β γράφεται:

$$B(m) = \begin{cases} n, \text{αν } (\exists k)(T_A(n, m, k) = \text{ναι}), \text{ και } (n, k) \text{ είναι το ελάχιστο ζεύγος} \\ \text{στην διάταξη } \leq_{DT} \text{ με αυτή την ιδίοτητα} \\ \text{δεν ορίζεται αλλιώς} \end{cases}$$

Προφανώς αν έχουμε $B(m) = n$ τότε $A(n) = m$. Άρα αν g η συνάρτηση που ορίζει ο αλγόριθμος B, τότε $dom(g) \subseteq rng(f)$. Αλλά και αντίστροφα, αν $m \in rng(f)$ τότε $T_A(x, m, y) =$ ναι για κάποιο ζεύγος (x, y). Άρα αν (n, k) είναι το ελάχιστο ζεύγος στη διάταξη $\leq_{DT}$,τότε $B(m) = n$, δηλαδή $g(m) = n$, συνεπώς $m \in dom(f)$. Οπότε $dom(g) = rng(f) = X$.
(γ)⇒(α):Έστω $X = dom(f)$, όπου f αλγοριθμική με αλγόριθμο Α και έστω B ο έξης αλγόριθμος :

$$B(n) = \begin{cases} \text{ναι, αν } A(n) \downarrow \\ \text{δεν ορίζεται αλλιώς} \end{cases}$$

Τότε $n \in X \iff n \in dom(f) \iff A(n) \downarrow \iff B(n) \downarrow$ και $B(n) =$ ναι. Άρα X α.α.

(δ)⇒(α): Άμεσο από 1.1.8.
(α)⇒(δ): Από 1.1.8 υποθέτουμε ότι $X = rng(f)$όπου f ολική αλγοριθμική. Θέλουμε να βρούμε 1-1 ολική αλγοριθμική g τέτοια ώστε $X = rng(g)$. Η g ορίζεται ως εξής :
$g(0) = f(0)$ και
$g(n+1) = f(m)$, όπου m είναι ο ελάχιστος x για τον οποίο $f(x) \notin \{g(0), \cdots, g(n)\}$.
Με την χρήση του τελεστή ελαχιστοποιήσης η παραπάνω σχέση γράφεται
$g(n + 1) = f((\mu x)[f(x) \notin \{g(0), \cdots, g(n)\}])$.
Επειδή το $X = rng(f)$ είναι άπειρο για κάθε n, $X - \{g(0), \cdots, g(n)\} \neq \emptyset$ άρα πάντα θα υπάρχει x τέτοιο ώστε $f(x) \notin \{g(0), \cdots, g(n)\}$, και συνεπώς η g ορίζεται σε όλο το $\mathbb{N}$. Προφανώς η g είναι αλγοριθμική ,1-1, και $rng(g) = rng(f) = X$. □

Το παραπάνω αποτελεί έναν χαρακτηρισμό για τα άπειρα α.α. $X \subseteq \mathbb{N}$ σύνολα ότι δηλαδή αποτελούν το πεδίο τιμών μιας ολικής, 1-1 αλγοριθμικής $f : \mathbb{N} \to \mathbb{N}$.

Πρόταση 1.1.11. *Αν $X \subseteq \mathbb{N}$ αλγοριθμικό, υπάρχει $f : \mathbb{N} \to \mathbb{N}$ ολική αλγοριθμική και αύξουσα, τέτοια ώστε $X = rng(f)$.Αν το X είναι άπειρο, η παραπάνω f είναι αυστηρά αύξουσα. (Δηλαδή,κάθε άπειρο αλγοριθμικό $X \subseteq \mathbb{N}$ έχει μια αλγοριθμική γνησίως αύξουσα απαρίθμηση των στοιχειών του.)*

Απόδειξη. Έστω $X \subseteq \mathbb{N}$ αλγοριθμικό. Ορίζουμε:
$f(0) =$ ελάχιστο στοιχείο του X,
$f(n+1) =$ ελάχιστο στοιχείο του$X - \{f(0), \cdots, f(n)\}$
αν $X - \{f(0), \cdots, f(n)\} \neq \emptyset$
$= f(n)$ αλλιώς.
Επειδή X αλγοριθμικό είναι εύκολο να δούμε ότι η f είναι ολική αλγοριθμική(αν X αλγοριθμικό η ιδιότητα "x ελάχιστο στοιχείο του X " είναι αλγοριθμική, δεν συμβαίνει το ίδιο αν το X είναι α.α.). Επίσης η f είναι αύξουσα και $X = rng(f)$. Τέλος ο μόνος λόγος για να μην είναι η f αυστηρά αύξουσα, θα ήταν το X να είναι πεπερασμένο. □

Λήμμα 1.1.12. *Αν η $f : \mathbb{N} \to \mathbb{N}$ είναι αυστηρά αύξουσα, τότε $f(n) \geq n$ για κάθε $n \in \mathbb{N}$*

Απόδειξη. Μία συνάρτηση $f : \mathbb{N} \to \mathbb{N}$ λέγεται άυξουσα αν $m < n \Rightarrow f(m) \leq f(n)$ και αυστηρά αύξουσα αν $m < n \Rightarrow f(m) < f(n)$.
Για να αποδείξουμε τον ισχυρισμό μας κάνουμε επαγωγή στο n.
$f(0) \geq 0$ είναι προφανές.
Έστω ότι ισχύει για n,$f(n) \geq n$.Έχουμε από υπόθεση $f(n+1) > f(n)$, άρα $f(n+1) > f(n) \geq n$ οπότε $f(n+1) \geq n+1$. Άρα αποδείχθηκε το ζητούμενο. □

Πρόταση 1.1.13. *Αν η $f : \mathbb{N} \to \mathbb{N}$ είναι αυστηρά αύξουσα, το $x = rng(f)$ είναι αλγοριθμικό.*

Απόδειξη. Έχουμε $n \in X \iff (\exists m)(f(m) = n)$. Όμως από το προηγούμενο Λήμμα 1.1.12 $f(m) \geq m$, δηλαδή αν $f(m) = n$ τότε $m \leq n$. Άρα έχουμε στην περίπτωση που η f είναι αύξουσα

$$n \in X \iff (\exists m \leq n)(f(m) = n)$$

Αυτό σημαίνει ότι για να ελέγξουμε αν $n \in X$, αρκεί να ελέγξουμε μόνο αν $n = f(0), n = f(1), \cdots, n = f(n)$ που σημαίνει ότι η αναζητησή μας είναι φραγμένη. Αυτό προφανώς συνιστά ολικό ναι/όχι αλγόριθμο για το ερώτημα "$x \in X$;" □

Άρα από τις προτάσεις 1.1.11 και 1.1.3 έχουμε το εξής Πόρισμα.

Πόρισμα 1.1.14. *Ένα άπειρο $X \subseteq \mathbb{N}$ είναι αλγοριθμικό αν και μόνο αν έχει μια αυστηρά αύξουσα απαρίθμηση (είναι πεδίο τιμών μιας αυστηρά αύξουσας $f : \mathbb{N} \to \mathbb{N}$).*

1.2 Κωδικοποιήση-Αποκωδικοποιήση Συνόλων

Έστω $X \subseteq \mathbb{N}^k$ και $Y \subseteq \mathbb{N}^l$. Μια ολική συνάρτηση $f : X \to Y$ λέγεται αλγοριθμικός ισομορφισμός(α.ι. για συντομία) αν είναι αλγοριθμική , 1-1 και επί. Αν περιοριστούμε σε συναρτήσεις $f : \mathbb{N}^k \to \mathbb{N}^l$ παρατηρούμε οτι η σύνθεση α.ι. μας δίνει α.ι. το ίδιο ισχύει και για τις αντίστροφες συναρτήσεις καθώς και για την ταυτοτική απεικόνιση. Άρα το σύνολο των α.ι. στο $\mathbb{N}^k$ είναι μια ομάδα. Η παρακάτω πρόταση οφείλεται στον Cantor και δίνει έναν α.ι. μεταξύ του $\mathbb{N}^2$ και του $\mathbb{N}$.

Πρόταση 1.2.1. *Η συνάρτηση $J : \mathbb{N}^2 \to \mathbb{N}$ που ορίζεται από τη σχέση*

$$J(m,n) = \frac{(m+n)(m+n+1)}{2} + m$$

είναι α.ι. Επίσης υπάρχουν αλγοριθμικές $K : \mathbb{N} \to \mathbb{N}, L : \mathbb{N} \to \mathbb{N}$ έτσι ώστε για κάθε $q, (K(q), L(q)) = J^{-1}(q)$.

Απόδειξη. Γράφουμε τα στοιχεία του $\mathbb{N}^2$ υπό τη μορφή άπειρου πίνακα ως εξής:

$$\begin{array}{ccccc} (0,0) & (0,1) & (0,2) & (0,3) & \cdots \\ (1,0) & (1,1) & (1,2) & (1,3) & \cdots \\ (2,0) & (2,1) & (2,2) & (2,3) & \cdots \\ (3,0) & (3,1) & (3,2) & (3,3) & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots \end{array}$$

Κατόπιν απαριθμούμε τα στοιχεία του πίνακα κινούμενοι κατά μήκος των διαγωνίων ΒΑ προς ΝΔ,δηλαδή ως εξής:
(0,0), (0,1), (1,0), (0,2), (1,1) (2,0), $\cdots$. Κάθε διαγώνιος περιέχει ζεύγη με άθροισμα στοιχείων σταθερό. Αν σε μια διαγώνιο υπάρχουν ζεύγη (m,n) με $m+n = k$,τότε η διαγώνιος έχει $k+1$ στοιχεία. Συνεπώς αν το ζεύγος (m,n) κατέχει την q-οστή θέση στην παραπάνω ακολουθία, $q = 1+2+\cdots+(m+n)+m = \frac{(m+n)(m+n+1)}{2}+m$.Αυτό δείχνει ότι η απεικόνιση $J(m,n)$ παρέχει ακριβώς τη θέση του ζεύγους (m,n) στην πιο πάνω απαρίθμηση,άρα 1-1 και επί.

Αν $J(m,n) = q$ τότε $(m,n) = J^{-1}(q)$. Τα m,n μπορούν να ανακτηθούν από το q με αλγοριθμικό τρόπο. Αν θέσουμε $m+n = s$, τότε εύκολα βλέπουμε ότι πρέπει $\frac{s(s+1)}{2} \leq q < \frac{(s+1)(s+2)}{2}$. Δοθέντος q, υπάρχει μονάδικο s μ' αυτή την ιδιότητα, το οποίο προφανώς υπολογίζεται αλγοριθμικά. Τότε όμως $m = q - \frac{s(s+1)}{2}$ και $n = s-m$. Άρα αρκεί να θέσουμε $K(q) = q - \frac{s(s+1)}{2}$ και $L(q) = s - K(q)$. □

Με τη βοήθεια της J μπορούμε να ορίσουμε για κάθε $k \geq 2$ έναν α.ι. $J_k : \mathbb{N} \to \mathbb{N}$.
Οι J_k ορίζονται επαγωγικά ως εξής:
$J_2 = J$,
$J_{k+1}(x_1, \cdots, x_{k+1}) = J(x_1, J_k(x_2, \cdots, x_{k+1}))$.
Επαγωγικά αποδεικνύεται ότι κάθε J_k είναι α.ι. Τέλος αν $\mathbb{N}^{<\omega} = \bigcup_{k\geq 2} \mathbb{N}^k$ είναι

το σύνολο όλων των πεπερασμένων ακολουθιών φυσικών αριθμών με μήκος $\geq$ 2,ορίζουμε $J_\omega : \mathbb{N}^{<\omega} \to \mathbb{N}$ ως εξής: Για κάθε $n \geq 2$ και κάθε $(x_1, \cdots, x_n) \in \mathbb{N}^n$,

$$J_\omega(x_1, \cdots, x_n) = J(n, J_n(x_1, \cdots, x_n)).$$

Η J_ω είναι 1-1 και επί. Όπως και στην περίπτωση του J, για κάθε k υπάρχουν αλγοριθμικές $K_i : \mathbb{N} \to \mathbb{N}, 1 \leq i \leq k$, τέτοιες ώστε για κάθε $q \in \mathbb{N}, (K_1(q), \cdots, K_k(q) = J_k^{-1}(q)$.

Με τη βοήθεια των J_k, μια συνάρτηση $f : \mathbb{N}^k \to \mathbb{N}^l$ μπορεί να μετατραπεί σε μια συνάρτηση $g : \mathbb{N} \to \mathbb{N}$, θέτοντας $g = J_l \circ f \circ J_k^{-1}$ με την εξής έννοια: Η f είναι (ολική) αλγοριθμική αν και μόνον αν η g είναι ολική αλγοριθμική.

Αυτό που κάνει η συνάρτηση J αλλά και γενικότερα οι συναρτήσεις J_n είναι να στοιχίζουν ένα (x_1, x_2) ή γενικότερα ένα $(x_1, \cdots, x_n)$ σε ένα μοναδικό στοιχείο του $\mathbb{N}$, επίσης ισχύει και το αντίστροφο δηλαδή να επιστρέψουμε στην n-αδα από τον αριθμό. Ουσιαστικά "αποθηκεύουμε" την πληροφορία που υπάρχει στο $(x_1, \cdots x_n)$ σε έναν αριθμό του $\mathbb{N}$ με τέτοιο τρόπο ώστε να μπορούμε και από τον αριθμό να επιστρέψουμε στο $(x_1, \cdots, x_n)$. Τέτοιες αλγοριθμικές και αντιστρέψιμες διαδικασίες αντιπροσώπευσης ενός αντικειμένου πεπερασμένου τύπου από ένα άλλο λέγονται κωδικοποιήσεις. Κάθε αλγοριθμικός ισομορφισμός $f : X \to Y$ είναι μια κωδικοποίηση των στοιχείων του X μετα στοιχεία του Y. Το $f(x)$ λέγεται κωδικός του x. Η αντίστροφη αντιστοιχία $f(x) \longmapsto x$ λέγεται αποκωδικοποίηση. Οι κωδικοποιήσεις ζευγών φυσικών αριθμών με ένα φυσικό δηλαδή οι ολικές αλγοριθμικές 1-1 $f : \mathbb{N}^2 \to \mathbb{N}$ λέγονται συναρτήσεις ζεύγους. Η $J(m, n) = \frac{(m+n)(m+n+1)}{2} + m$ είναι μια συνάρτηση ζεύγους αλλά δεν είναι η μόνη, παραδείγματα τέτοιων συναρτήσεων είναι οι εξής :
$G_2(m, n) = 2^m \cdot 3^n$
$h(m, n) = 2^{m+n+2} + 2^{n+1}$
$e(m, n) = 2^m(2n + 1)$

Η G_2 γενικευέται στην $G_k : \mathbb{N}^k \to \mathbb{N}$ ως εξής: Αν p_n ο n-οστός πρώτος αριθμός,θέτουμε

$$G_k(x_1, \cdots, x_k) = p_1^{x_1} \cdots p_k^{x_k} = \prod_{i=1}^{k} p_i^{x_i}$$

Όπου $p_1 = 2, p_2 = 3$ κλπ. Η G_k είναι κωδικοποίηση (όχι επί) των στοιχείων του $\mathbb{N}^k$ και είναι καλύτερη από την J_k λόγω της απλότητας της. Η κωδικοποίηση G_k χρησιμοποιεί το θεμελιώδες θεώρημα της Θεωρίας Αριθμών ότι κάθε ακέραιος αριθμός έχει μονοσήμαντη ανάλυση σε γινόμενο πρώτων.

Επίσης για την κωδικοποίηση των στοιχείων του $\mathbb{N}^{<\omega}$, έχουμε την $G : \mathbb{N}^{<\omega} \to \mathbb{N}$ να ορίζεται ως εξής: Για κάθε $n \in \mathbb{N}$ και κάθε $x_1, \cdots, x_n$,

$$G(x_1, \cdots, x_n) = \prod_{i=1}^{n} p_i^{x_i+1}$$

Η συνάρτηση G οφείλεται στον K.Gödel και συχνά ο $G(x_1, \cdots x_n)$ αναφέρεται ως αριθμός Gödel της n-αδας $(x_1, \cdots x_n)$.
Ο λόγος που βάζουμε $p_i^{x_i+1}$ αντί $p_i^{x_i}$ είναι για να κάνουμε την G 1-1.Αλλιώς θα είχαμε π.χ. $G(2,5,1) = G(2,5,1,0) = G(2,5,1,0,0)$ κλπ. Συχνά για λόγους συντομίας γράφουμε αντί για $G(x_1, \cdots, x_n)$

$$\langle x_1, \cdots, x_n \rangle$$

Οι αντίστροφες συναρτήσεις της $\langle\rangle$ γράφονται $()_i$ δηλαδή,

$$\langle x_1, \cdots, x_n \rangle = y \Longleftrightarrow \forall i \leq n (y)_i = x_i$$

Θα μπορούσαμε να ταυτίσουμε τις κωδικοποιήσεις με τους αλγοριθμικους ισομορφισμούς αν περιοριζόμασταν σε στοιχεία του $\mathbb{N}^k$. Όμως η κωδικοποιήση είναι γενικοτέρη έννοια και εφαρμόζεται και σε άλλα σύνολα όπως π.χ. το σύστημα των σημάτων Morse που μετατρέπει τα γράμματα της αγγλικής αλφαβήτου σε ακολουθίες από τελείες και παύλες.

Κάθε μη κενό το πολύ αριθμίσιμο σύνολο Σ μπορεί να θεωρηθεί αλφάβητο μιας γλώσσας. Τα στοιχεία του Σ είναι τα σύμβολα του αλφάβητου. Οι λέξεις ή φράσεις είναι ορισμένες πεπερασμένες ακολουθίες συμβόλων που υπακούουν σε κάποιους κανόνες σχηματισμού της γλώσσας. Π.χ.το αλφάβητο της Ελληνικής γλώσσας είναι το $\Sigma = \{\alpha, \beta, \cdots, \omega\} \cup \{$κόμμα,τελεία,κλπ$\}$ και μία φράση του είναι η ακολουθία "Θα φάω σε λίγο,δεν πεινάω τώρα." Οι αριθμοί του δεκαδικού είναι λέξεις του αλφαβήτου $\Sigma = \{0, 1, 2, \cdots, 9\}$, ενώ οι αριθμοί του δυαδικού συστήματος είναι λέξεις από το αλφάβητο $\Sigma = \{0, 1\}$. Στο πρώτο παράδειγμα η γλώσσα έχει πολύπλοκους κανόνες σχηματισμού, δεν είναι όλες οι ακολουθίες των συμβόλων του Σ λέξεις ή φράσεις π.χ.η ακολουθία "ξγσαρφθ" δεν αποτελεί κάποια λέξη στο ελληνικό λεξιλόγιο. Αντίθετα στο δεκαδικό και στο δυαδικό σύστημα οποιαδήποτε πεπερασμένη ακολουθία $a_1, \cdots, a_n$ όπου $a_i \in \{0, 1, \cdots 9\}$ ή $a_i \in \{0, 1\}$, αποτελεί κάποιον αριθμό του δεκαδικού ή του δυαδικού συστήματος αντίστοιχα. Σάυτές τις περιπτώσεις οι λέξεις ταυτίζονται με τις πεπερασμένες ακολουθίες συμβόλων του Σ και το συνολό τους ταυτίζεται με το Σ^*, όπου Σ^* είναι η ελεύθερη ημιομάδα με γεννήτορες τα στοιχεία του Σ.

Αν τώρα $\Sigma = \{a_1, \cdots, a_n\}$ είναι μια αρίθμηση του Σ και θεωρήσουμε τον φυσικό αριθμό n κώδικα του a_n, οι λέξεις μετατρέπονται σε πεπερασμένες ακολουθίες του $\mathbb{N}$, δηλαδή στοιχεία του $\mathbb{N}^{<\omega}$. Έτσι η κωδικοποίηση των λέξεων του Σ, ανάγεται σε κωδικοποίηση των στοιχείων του $\mathbb{N}^{<\omega}$, η οποία όπως είδαμε πιο πάνω είναι εφικτή μέσω της συνάρτησης Gödel. Έτσι για κάθε λέξη $a_{i1}, \cdots, a_{in}$ μπορούμε να θέσουμε

$$G(a_{i1}, \cdots a_{in}) = G(i_1, \cdots, i_n) = \prod_{k=1}^{n} p_k^{ik+1}$$

Η κωδικοποίηση μη αριθμιτικών εννοιών, όπως οι λέξεις μια γλώσσας με αριθμούς λέγεται συχνά και αριθμητικοποίηση ή αριθμοποίηση.

2 Οι Αλγοριθμικές Συναρτήσεις μέσω Αναδρομής

Δύο είναι οι βασικές τυποποιήσεις(δηλαδή μαθηματικοποιήσεις) της εμπειρικής έννοιας του αλγορίθμου: Η μία μέσω αναδρομικών συναρτήσεων και η άλλη μέσω των μηχανών Turing. Σε αυτό το κεφάλαιο θα μιλήσουμε για τις αναδρομικές συναρτήσεις και για τις μηχανές Turing στο επόμενο.

2.1 Γενικά περί Αναδρομής

Αναδρομή, είναι η επαναληπτική εφαρμογή ενός κανόνα, όπου σε κάθε βήμα (εκτός τους πρώτου) χρησιμοποιούμε σαν είσοδο την έξοδο του προηγουμένου. Π.χ. υπάρχουν ακολουθίες που ορίζονται αναδορμικά, δηλαδή ο n-οστός όρος a_n της ακολουθίας είναι συναρτήση του προηγούμενου a_{n-1}, ή των a_{n-2} και a_{n-1} κλπ. Π.χ. η ακολουθία που ορίζεται ως εξής:

$$a_0 = 2, \quad a_1 = 3, \quad a_{n+1} = a_n + a_{n-1}$$

είναι αναδρομική με $a_0 = 2, a_1 = 3, a_2 = a_1 + a_0 = 5$ κτλ.
Μια ακολουθία είναι μια συνάρτηση $f(n) = a_n$, π.χ η παραπάνω ακολουθία γράφεται

$$f(0) = 2, \quad f(1) = 3, \quad f(n+1) = f(n) + f(n-1)$$

Εδώ για υπολογίσουμε κάποιο βήμα εκτός των δύο πρώτων, χρησιμοποιούμε τις τιμές που δίνουν τα προηγούμενα βήματα $n-1$,n σαν είσοδο για το βήμα $n+1$. Εδώ η f είναι αλγοριθμική καθώς η $f(n+1)$ παράγεται από τα $f(n-1), f(n)$ μέσω της πρόσθεσης η οποία ειναι αλγοριθμική. Η $f : \mathbb{N}^2 \to \mathbb{N}$,όπου $f(x, y) = x+y$,είναι ανδρομική και ορίζεται απότο σχήμα:

$$f(x, 0) = x, \quad f(x, y+1) = f(x, y) + 1$$

Θεωρούμε παράμετρο το x, βλεπουμε ότι το $f(x, y+1)$ υπολογίζεται μέσω του $f(x, y)$ και της συνάρτησης διαδοχής $S(x) = x + 1$. Η $S(x)$, είναι αλγοριθμική, δεν ανάγεται σε απλούστερες, γι'αυτό δεχόμαστε εξ'ορισμού ότι είναι αναδρομική. Παρατηρούμε ότι η αναδρομή έχει ως κύριο χαρακτηριστικό την επανάληψη.

2.2 Βασικές αναδρομικές συναρτήσεις ή Πρωτογενής Αναδρομικές(primitive recursive fuction)

Ορισμος 2.2.1 (Σχήμα βασικής αναδρομής ή πρωτογενή αναδρομή). *Έστω f:$\mathbb{N}^{k+1}\to\mathbb{N}$, όπου γράφουμε $f(\vec{x}, y)$ το $\vec{x}$ έχει μήκος k, και παίζει ρόλο παραμέτρων. Έστω επίσης g:$\mathbb{N}^k\to\mathbb{N}$ και h:$\mathbb{N}^{k+2}\to\mathbb{N}$.Θα λέμε ότι η f παράγεται από τις g και h με βασική αναδρομή αν για κάθε $\vec{x} \in \mathbb{N}^k$ και $y \in \mathbb{N}$ ισχύουν οι παρακάτω σχέσεις,*

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

Επίσης η κλάση των αλγοριθμικών συναρτήσεων είναι κλειστή ως προς την σύνθεση. Χονδρικά,αν f, g αλγοριθμικές και η $f \circ g$ είναι αλγοριθμική. Επειδή η σύνθεση πάντως παίρνει γενικότερες μορφές, θα την ορίσουμε με το παρακάτω Σχήμα Σύνθεσης.

Ορισμος 2.2.2 (Σχήμα Σύνθεσης). *Έστω $f : \mathbb{N}^k \to \mathbb{N}, h : \mathbb{N}^m \to \mathbb{N}$ και $g_i : \mathbb{N}^k \to \mathbb{N}, i = 1, ..., m$. Θα λέμε ότι η f είναι σύνθεση των h και g_i,αν $f(\vec{x}) = h(g_1(\vec{x}), ..., g_m(\vec{x}))$. Συμβολικά γράφουμε τότε $f = h \circ (g_1, ..., g_m)$.*

Είναι προφανές ότι οι παράκατω συναρτήσεις είναι αλγοριθμικές:
(α)Η συνάρτηση διαδοχής $S : \mathbb{N} \to \mathbb{N}$ με $S(x) = x + 1$,
(β)Οι σταθερές συναρτήσεις $C_k : \mathbb{N} \to \mathbb{N}$,για κάθε $k \in \mathbb{N}$,όπου $C_k(x) = k$ για κάθε $x \in \mathbb{N}$
(γ)Οι προβολές ,δηλαδή οι συναρτήσεις $\Pi_{nm} : \mathbb{N}^n \to \mathbb{N}$,για $1 \leq m \leq n$,όπου $\Pi_{nm}(x_1, ..., x_n) = x_m$.
Οι συναρτήσεις S,C_k και π_{nm} λέγονται αρχικές συναρτήσεις.

Ορισμος 2.2.3. *Το σύνολο PR των βασικών αναδρομικών συναρτήσεων (β.α. για συντομία) ορίζεται ως το ελάχιστο σύνολο C (δηλαδή η τομή των συνολών C) με τις ιδιότητες:*
(1) Το C περιέχει τις βασικές συναρτήσεις C_k, S, Π_{nm}.
(2) Το C είναι κλειστό ως προς την πρωτογενή αναδρομή δηλαδή αν $g, h \in C$ και η f ορίζεται με πρωτογενή αναδρομή απο τις g και h τότε $f \in C$. Είναι κλειστό ως προς την σύνθεση δηλαδή αν $h : \mathbb{N}^m \to \mathbb{N}$ και $g_i : \mathbb{N}^k \to \mathbb{N}$,$i = 1, ..., m$, ανήκουν στο C τότε η $f = h \circ (g_1, .., g_m)$ ανήκει στο C.

Απόδειξη. (Ύπαρξη ελαχίστου)
Έστω το σύνολο $PR = \bigcap\{A \in \bigcup_k(\mathbb{N}^k \to \mathbb{N}), \text{A: ικανοποιεί τις (1) και (2)}\}$. Αρκεί να δείξω ότι PR ικανοποιεί τις (1) και (2).
Για τις βασικές συναρτήσεις.
- $S \in A, \forall A \in \{A \in \bigcup_k(\mathbb{N}^k \to \mathbb{N}), \text{A: ικανοποιεί τις (1) και (2)}\}$, αφού Α ικανοποιεί τις (1) και (2)άρα και $S \in PR$ αφού είναι η τομή των Α παρομοιώς για την σταθερή C_k και για την προβολή Π_{nm}
Για την συνθέση.
-Έστω $h, g \in PR$ θα δείξω ότι και η $h \circ g \in PR$
$\forall A \in \{A \in \bigcup_k(\mathbb{N}^k \to \mathbb{N}), \text{A: ικανοποιεί τις (1) και (2)}\}$.
$PR \subseteq A \Rightarrow h, g \in A$ άρα $h \circ g \in A$ αφού το A είναι κλειστό ως προς την σύνθεση, άρα $\forall A \in \{A \in \bigcup_k(\mathbb{N}^k \to \mathbb{N}), \text{A: ικανοποιεί τις (1) και (2)}\}$, $h \circ g \in A$, άρα ανήκει στην τομή των A άρα $h \circ g \in PR$.
-Με παρόμοιο τρόπο δουλεύουμε για την πρωτογενή αναδρομή δηλαδή, έστω $h, g \in PR$ θα δείξουμε ότι η f που ορίζεται από πρωτογενή αναδρομή από τις g, h ανήκει στο PR.

□

Παρατήρηση 2.2.4. *(1)Από τον ορισμό 2.2.3 έπεται ότι μπορούμε να κάνουμε επαγωγή με την εξής έννοια. Έστω μια ιδιότητα $Property(f)$, $f : \mathbb{N}^k \to \mathbb{N}$ αν*

ισχύουν
(α) $Property(C_k)$, $Property(S)$ *και* $Property(\Pi_{nm})$
(β) Αν $Property(f)$, $Property(g_1)$,...,$Property(g_m) \Rightarrow Property(f \circ (g_1, ..., g_m))$
(γ) Αν $Property(g)$, $Property(h)$ *και η* f *παράγεται απο τις* g, h *με βασική αναδρομή και* $Property(f)$
τότε $\forall f \in PR$ *ισχύει* $Property(f)$.
(2)Ένας ισοδύναμος ορισμός των β.α. συναρτήσεων είναι: Η f *είναι β.α. αν υπάρχει μια πεπερασμένη ακολουθία συναρτήσεων* $f_1, \ldots, f_n$, *τέτοια ώστε :(α)* $f_n = f$ *και (β) για κάθε* $i \leq n$, *η* f_i *είτε είναι αρχική είτε προέρχεται από δύο προηγούμενες με* $f_k, f_j (k, j < i)$ *με βασική αναδρομή, είτε προέρχεται από άλλες προηγούμενες με σύνθεση.*
(3)Κάθε β.α.είναι ολική.Αυτό αποδεικνύεται επαγωγικά, δεδομένου ότι όλες οι αρχικές συναρτήσεις είναι είναι ολικές τα δε σχήματα βασικής αναδρομής και σύνθεσης οδηγούν από ολικές συναρτήσεις σε ολικές.
(4)Αντί για όλες τις σταθερές συναρτήσεις θα μπορούσαμε να θεωρήσουμε αρχική μόνο την $C_0 (C_0(x) = 0$ *για κάθε* $x)$ *αφού* $C_1(x) = S \circ C_0(x)$ *και γενικά* $C_k = S^k \circ C_0$.

Το σύνολο PR αποτελεί μια πρώτη προσέγγιση της κλάσης των αλγοριθμικών συναρτήσεων. Στα παρακάτω παραδείγματα δείχνουμε πως μια πληθώρα συναρτήσεων που χρησιμοποιούμε στην πράξη ανήκουν σε αυτό.

ΠΑΡΑΔΕΙΓΜΑΤΑ
(1)Κάθε σταθερή συνάρτηση $f : \mathbb{N}^k \to \mathbb{N}$ είναι β.α..
Έστω $f(\vec{x}) = \alpha$ για κάθε $\vec{x} \in \mathbb{N}^k$. Τότε $f(\vec{x}) = C_\alpha(\Pi_{\kappa 1}(\vec{x}) = \alpha)$, άρα $f = C_a \circ \Pi_{\kappa 1}$ και αφού $C_\alpha, \Pi_{\kappa 1} \in PR$, $f \in PR$.
(2)Η ταυτοτική συνάρτηση είναι β.α..
Η $f(\vec{x}) = x = \Pi_{11}(\vec{x})$. Άρα $f = \Pi_{11}$.
(3)Η πρόσθεση είναι β.α..
Έστω $f(x, y) = x + y$.Όπως είδαμε ήδη η f ορίζεται ως εξής :
$f(x, 0) = x = \Pi_{11}(x)$ είναι β.α.
$f(x + 1, y) = x + 1 + y = f(x, y) + 1 = S \circ \Pi_{33}(x, y, f(x, y))$
Άρα η f παράγεται από πρωτογενή αναδρομή από τις $\Pi_{11}(x)$ και $h = S \circ \Pi_{33}$. Η h είναι β.α. ως σύνθεση β.α. συναρτήσεων(S, Π_{33}) η $\Pi_{11}(x)$ είναι β.α. συνεπάγεται και η f είναι β.α.
(4)Ο πολλαπλασιασμός είναι β.α.
Έστω $f(x, y) = x \cdot y$.
$f(x, 0) = 0 = C_0(x)$, είναι β.α.
$f(x, y + 1) = x \cdot (y + 1) = x \cdot y + y = f(x, y) + x = \Pi_{33}(x, y, f(x, y)) + \Pi_{13}(x, y, f(x, y))$.
Άρα η f παράγεται από πρωτογενή αναδρομή από τις C_0 που είναι β.α. και από την $h = \Pi_{33} + \Pi_{13} = + \circ (\Pi_{33}, \Pi_{31})$ που είναι β.α. ως σύθενση β.α. συναρτήσεων άρα και η f είναι β.α.
(5)Η x^y είναι β.α..
Έστω $f(x, y) = x^y$.
$f(x, 0) = x^0 = 1 = C_1(x)$ που είναι β.α.

$f(x,y+1) = x^{(y+1)} = x^y \cdot x = f(x,y) \cdot x = \Pi_{33}(x,y,f(x,y)) \cdot \Pi_{31}(x,y,f(x,y))$.
Άρα η f παράγεται με πρωτογενή αναδρομή από τις C_1 και $h = \Pi_{33} \cdot \Pi_{31} = \cdot \circ (\Pi_{33}, \Pi_{13})$ που είναι β.α. συναρτήσεις άρα είναι και η f.
(6)Η $f(x) = x!$ είναι β.α.
η f ορίζεται
$f(0) = 0! = 1 = C_1(x)$ Που είναι β.α.
$f(x+1) = (x+1)! = x \cdot (x+1) = f(x) \cdot (x+1)$
Έστω $h(x, f(x)) = f(x+1) = f(x)(x+1) = \Pi_{22}(x, f(x)) \cdot S_{\Pi_{21}}(x, f(x))$, άρα $h = \Pi_{21} \cdot S_{\Pi 21}$ που είναι β.α. ως γινόμενο β.α. συναρτήσεων. Άρα η f είναι β.α. αφού παράγεται με πρωτογενή αναδρομή από τις β.α. C_1, h.
(7)Η συνάρτηση του προηγουμένου $Pd(x)$ είναι β.α.
Ορίζεται ως εξής
$Pd(0) =$
$Pd(x+1) = x$
Άρα η Pd είναι β.α. αφού $Pd(0) = C_0$ και $Pd(x+1) = \Pi_{21}(x, Pd(x))$.
(8)Η συνάρτηση της διαφοράς $x \dot{-} y$ είναι β.α

$$x \dot{-} y = f(x,y) = \begin{cases} x - y, & \text{αν } x \geq y \\ 0, & \text{αν } x \leq y \end{cases}$$

$f(x,0) = x - 0 = x$ που είναι β.α. και $f(x, y+1) = x \dot{-} (y+1) = Pd(x \dot{-} y)$ που είναι β.α.
(9)Οι συναρτήσεις του προσήμου $sign(x)$ και συμπροσήμου $cosign(x)$ είναι β.α.
Οι $sign(x), cosign(x)$ ορίζονται ως εξής
$sign(0) = 0, sign(x+1) = 1$
$cosign(0) = 1, cosign(x+1) = 0$ που είναι προφανές ότι είναι β.α.
(10)Η $|x-y|$ είναι β.α.
$|x-y| = (x \dot{-} y) + (y \dot{-} x)$ άρα είναι β.α.
(11)Η $Sg(x) = \begin{cases} 1, & \text{αν } x > 0 \\ 0, & \text{αν } x = 0 \end{cases}$ είναι β.α.
$Sg(0) = 0$ είναι β.α
$Sg(x+1) = 1$ είναι β.α. άρα η $Sg(x)$ είναι β.α.
(12)Οι συναρτήσεις άθροισης και γινομένου είναι β.α.
Έστω $f : \mathbb{N}^{k+1} \to \mathbb{N}$ είναι β.α. και $g, h : \mathbb{N}^{k+1} \to \mathbb{N}$, όπου $g(\vec{x}, y) = \sum_{t=0}^{y} f(\vec{x}, t)$ και $h(\vec{x}, y) = \prod_{t=0}^{y} f(\vec{x}, t)$ οι συναρτήσεις άρθροισης και γινομένου αντίστοιχα.
Η g ορίζεται αναδρομικά ως εξής:
$g(\vec{x}, 0) = f(\vec{x}, 0)$ που είναι β.α.
$g(\vec{x}, y+1) = \sum_{t=0}^{y+1} f(\vec{x}, t) = g(\vec{x}, y) + f(\vec{x}, y+1)$
Η g παράγεται με πρωτογενή αναδρομή και σύνθεση από τις f, $+$ και τις αρχικές συναρτήσεις άρα είναι β.α.
Η h ορίζεται αναδρομικά ως εξής :
$h(\vec{x}, 0) = f(\vec{x}, 0)$ που είναι β.α.
$h(\vec{x}, y+1) = \prod_{t=0}^{y+1} f(\vec{x}, t) = h(\vec{x}, y) \cdot g(\vec{x}, y+1)$

Η h παράγεται με πρωτογενή αναδρομή από τις f, $\cdot$ και τις αρχικές συναρτήσεις άρα είναι β.α.
(13)Η φραγμένη ελαχιστοποίηση είναι φραγμένη
Έστω $h : \mathbb{N}^{k+1} \to \{0,1\}$ και $g : \mathbb{N}^{k+1} \to \mathbb{N}$ β.α. τέτοιες ώστε
$(\forall \vec{x} \in \mathbb{N}^k)(\forall y \in \mathbb{N})(\exists z \leq g(\vec{x},y))(h(\vec{x},z) = 0)$
Τότε η συνάρτηση $f : \mathbb{N}^{k+1} \to \mathbb{N}$
$f(\vec{x},y) = (\mu z \leq g(\vec{x},y))(h(\vec{x},z)) = 0$ είναι β.α.
Παρατηρούμε ότι αν k είναι το ελάχιστο $z \leq g(\vec{x},y)$ τέτοιο ώστε $h(\vec{x},z) = 0$, τότε

$$h(\vec{x},0) = 1, h(\vec{x},1) = 1, ...h(\vec{x},k-1) = 1, h(\vec{x},k) = 0$$

Αν για κάθε $i \leq g(\vec{x},y)$ φτιάξουμε το γινόμενο $\Pi_{z\leq i}h(\vec{x},z)$ τότε $\Pi_{z\leq i}h(\vec{x},z) = 1$ για $i < k$ ενώ $\Pi_{z\leq i}h(\vec{x},z) = 0$ για $k \leq i$. Επί πλέον το άθροισμα όλων αυτώ των γινομένων για $i \leq g(\vec{x},y)$ είναι k αν και μόνο αν $f(\vec{x},y) = k$, δηλαδή

$$f(\vec{x},y) = k \Longleftrightarrow \sum_{i\leq g(\vec{x},y)} \prod_{z\leq i} h(\vec{x},z) = k$$

Συνεπώς $f(\vec{x},y) = \sum_{i\leq g(\vec{x},y)} \prod_{z\leq i} h(\vec{x},z)$ και από το (10) έπεται ότι η f είναι β.α.

2.3 Βασικά Αναδρομικά Σύνολα

Ορισμος 2.3.1. *Ένα σύνολο $X \subseteq \mathbb{N}^k$ λέγεται βασικό αναδρομικό ή β.α. για συντομία, αν η χαρακτηριστική του συνάρτηση C_X είναι β.α.*

Πρόταση 2.3.2. *Αν A, B είναι β.α. τότε και τα σύνολα $-A, A\cap B, A\cup B$ είναι β.α. Επίσης αν $A \subseteq \mathbb{N}^{k+1}$, για $k \geq 1$, είναι β.α., τότε και τα $B = \{(\vec{x},z) : (\forall y \leq z)(\vec{x},y) \in A\}$, $Z = \{(\vec{x},z) : (\exists y \leq z)(\vec{x},y) \in A\}$.*

Απόδειξη. $X_{-X} = 1 \dot{-} X_A(x) = \begin{cases} 1, & \text{αν } x \notin A \\ 0, & \text{αν } x \in A \end{cases}$
Όπου $X_A(x)$ η χαρακτιριστική συνάρτηση του A, άρα και το $-A$ είναι β.α. αφού $x \dot{-} y$ είναι β.α.
$X_{A\cap B}(x) = X_A(x)\cdot X_B(x) = \begin{cases} 1, & \text{αν } x \in A \text{ ,και } x \in B \\ 0, & \text{αν } x \notin A, \text{και } x \notin B \end{cases}$
Όπου X_A, X_B είναι β.α. ο πολλαπλασιασμός ειναι β.α. άρα και το $X_{A\cap B}$ είναι β.α.
$X_{A\cup B} = Sg(X_A(x) + X_B(x)) = \begin{cases} 1, & \text{αν } x \in A \text{ ή } x \in B \\ 0, & \text{αν } x \notin A, \text{ και } x \notin B \end{cases}$
Άρα επειδή $Sg(x)$, X_A, X_B, $+$ είναι β.α. και το $X_{A\cup B}$ είναι β.α.
Έστω $B = \{(\vec{x},z) : (\forall y \leq z)(\vec{x},y) \in A\}$. Τότε για κάθε $(\vec{x},z) \in \mathbb{N}^{k+1}$,
$X_B(\vec{x},z) = 1 \Longleftrightarrow (\forall y \leq z)(X_A(\vec{x},y) = 1) \Longleftrightarrow \prod_{y\leq z} X_A(\vec{x},y) = 1$

Άρα η $\prod_{y\leq z} X_A(\vec{x},y)$ είναι η χαρακτηριστική συνάρτηση του B που είναι β.α. άρα B β.α. Για το $Z = \{(\vec{x},z) : (\exists y \leq z)(\vec{x},y) \in A\}$ το συπμλήρωμα του $-Z$ είναι

όπως το προήγουμενο B άρα είναι β.α. άρα και το Z είναι β.α. ως συπμλήρωμα β.α. συνόλου.

□

Πρόταση 2.3.3. *Η πρόσθεση και η ανισότητα στο* $\mathbb{N}$ *είναι β.α. σχέσεις.*

Απόδειξη. Έστω $X_=$ και $X_<$ οι χαρακτηριστικές συναρτήσεις των συνόλων $\{(x,y) : x = y\}$ και $\{(x,y) : x < y\}$
$X_= = Sg(|x-y|) = \begin{cases} 0, & \text{αν } |x-y| > 0 \\ 1, & \text{αν } x = y \end{cases}$
Η $Sg(x),\ |x-y|$ είναι β.α. άρα και η $X_=$ είναι β.α.
$X_< = Sg(x \dot{-} y) = \begin{cases} 0, & \text{αν } x \geq y \\ 1, & \text{αν } x < y \end{cases}$
Η $Sg(x),\ (x \dot{-} y)$ είναι β.α. οπότε είναι και η $X_<$.

□

Πρόταση 2.3.4. *(α) Το σύνολο* $Prime = \{n : n\ prime\}$ *είναι β.α*
(β) Για κάθε $n \in \mathbb{N}$*, έστω* p_n *ο* $n+1$*-οστός πρώτος, δηλαδή* $p_0 = 2, p_1 = 3, p_2 = 5$ *κλπ. Η συνάρτηση* $p : \mathbb{N} \to \mathbb{N}$*, όπου* $p(n) = p_n$ *είναι β.α.*

Απόδειξη. (α)Αρκεί να δείξουμε ότι το συμπλήρωμα του $Prime$ είναι β.α. Όμως ο $n \notin Prime$ αν και μόνο αν υπάρχει $1 < x < n$ τέτοιο ώστε $x|n$. Αυτό γράφεται ως εξής :

$$n \notin Prime \iff (\exists x \leq n)(\exists y \leq n)((x+2)(y+2) = n).$$

Αφού η ιδιότητα $(x+2)(y+2) = n$ είναι β.α. τότε και η $(\exists x \leq n)(\exists y \leq n)((x+2)(y+2) = n)$ είναι β.α. συνεπώς το $-Prime$ είναι β.α. άρα είναι και το $Prime$ ως συμπληρωματικό του.
(β)Η p_n ορίζεται αναδρομικά ως εξής:
$p(0) = 2$
$p(n+1) = (\mu x)(x > p(n)\ and\ x \in Prime)$.
Αλλά τότε, $p(n+1) = h(p(n))$, όπου h η συνάρτηση

$$h(x) = (\mu y)(y > x\ \kappa\alpha\iota\ y \in Prime). \quad (1)$$

Αρκεί η h να είναι β.α. Τώρα αφού οι ιδιότητες $y > x$ και $y \in Prime$ είναι β.α. όπως είδαμε πιο πριν αρκέι να δείξουμε στην (1) ότι ο τελέστης μ είναι φραγμένος (δηλαδή έχουμε φραγμένη αναζήτηση) οπότε από το παράδειγμα (13) η h θα είναι β.α. Οπότε αρκεί να βρούμε μία g τέτοια ώστε

$$h(x) = (\mu y \leq g(x))(y > x\ \kappa\alpha\iota\ y \in Prime). \quad (2)$$

Μια τέτοια συνάρτηση είναι η $g(x) = x! + 1$ η οποία είναι β.α. όπως δείξαμε στο παράδειγμα (6). Ότι η (1) συνεπάγεται την (2) αποδεικνύεται ως εξής: Έστω q ένας πρώτος διαιρέτης του $g(x) = x!+1$.Αν $q \leq x$, τότε $q|x!$, άρα $q|(x!+1-x!) = 1$, άτοπο.Άρα $q > x$. Εξάλλου προφανώς $q \leq g(x)$. Συνεπώς για κάθε x υπάρχει πρώτος q, τέτοιος ώστε $x < q \leq g(x)$. Αυτό δείχνει ότι η h που ορίζεται από την (1) ικανοποιεί την (2). □

2.4 Πέρα από τις βασικές συναρτήσεις.Η συνάρτηση Ackermann

Η κλάση PR των β.α. συναρτήσεων, μονολότι πολύ ευρεία δεν περιλαμβάνει όλες τις συναρτήσεις που έχουν έναν προφανή αλγόριθμο. Ένα παράδειγμα μια συναρτήσης με προφανή αλγόριθμο που δεν είναι β.α. είναι η συνάρτηση Ackermann που ορίζεται ως εξής:

$$\begin{cases} A(0,y) & = y+1 \\ A(x+1,0) & = A(x,1) \\ A(x+1,y+1) & = A(x,A(x+1,y)). \end{cases} \qquad (1)$$

Η συνάρτηση Ackerman δεν είναι μια διπλή βασική αναδρομή όπως μπορεί να φαίνεται, που θα οδηγούσε αυτόματα στο συμπέρασμα ότι είναι β.α. συνάρτηση. Η διαφορά είναι ότι στην πολλαπλή αναδρομή, η αναδρομή γίνεται μέσω μιας τρίτης συνάρτησης h ανεξάρτητη από την f ενώ εδώ γίνεται μέσω της ίδιας A. Αυτό μας βγάζει από την κλάση των β.α.
Έχουμε τις εξής ιδιότητες:

(α) $A(1,n) = n+2$
Απόδειξη: Η απόδειξη γίνεται με επαγωγή.
-Δείχνω ότι ισχύει για $n = 0$: $A(1,0) = A(0,1) = 2$ Από την (1).
και έχω από την σχέση $A(1,n) = n+2 = 0+2 = 2$. Άρα ισχύει.
- Δέχομαι ότι ισχύει για n, δηλαδή $A(1,n) = n+2$ και θα δείξω ότι ισχύει για $n+1$ δηλαδή $A(1,n+1) = n+1+2 = n+3$,όμως
$A(1,n+1) = A(0,A(1,n)) = A(0,n+2) = n+2+1 = n+3$ άρα ισχύει.

(β) $A(2,n) = 2n+3$
Η απόδειξη γίνεται με επαγωγή όπως παραπάνω:
- Για $n = 0$, έχω $A(2,0) = 2 \cdot 0 + 3 = 3$, από (1) $A(2,0) = A(1,1) = A(0,A(1,0)) = A(0,A(0,1)) = A(0,2) = 3$ άρα ισχύει
- Έστω ότι ισχύει για n, δηλαδή $A(2,n) = 2n+3$ θ.δ.ο. ισχύει για $n+1$ δηλαδή $A(2,n+1) = 2(n+1)+3 = 2n+5$
Από (1) έχω $A(2,n+1) = A(1,A(2,n)) = A(1,2n+3) = 2n+3+2 = 2n+5$ [από (α)] άρα ισχύει.

(γ) $A(3,n) = 2^{n+3} - 3$

-Για $n = 0, A(3,0) = 2^3 - 3 = 5$ και από την εξίσωση του Ackermann έχω $A(3,0) = A(2,1) = 2 \cdot 1 + 3 = 5$ από το(β)
-Έστω ότι ισχύει για n, $A(3,n) = 2^{n+3} - 3$ ϑ.δ.ο. ισχύει για $n+1$ δηλαδή $A(3,n+1) = 2^{n+4} - 3$
$A(3,n+1) = A(2,A(3,n)) = A(2,2^{n+3}-3) = 2 \cdot (2^{n+3}-3) + 3 = 2^{n+4} - 3$ άρα ισχυεί.

(δ) $A(4,n) = 2^{2^{\cdot^{\cdot^{\cdot^{2}}}}}$ ($n+3$ δυάρια)
-Για $n = 0, A(4,0) = 2^{2^2} - 3 = 16 - 3 = 13$, όμως $A(4,0) = A(3,1) = 2^4 - 3 = 16 - 3 = 13$ άρα ισχύει για $n = 0$

-Έστω ότι ισχύει για n, ϑ.δ.ο. ισχύει για $n+1$ δηλαδή $A(4,n+1) = 2^{2^{\cdot^{\cdot^{\cdot^{2}}}}} - 3$,($n+4$ δυάρια)

$A(4,n+1) = A(3,A(4,n)) = A(3,2^{2^{\cdot^{\cdot^{\cdot^{2}}}}} - 3)$ ($n+3$ δυάρια) από το (γ) είναι ίσο με $2^{2^{\cdot^{\cdot^{\cdot^{2}}}}} + 3 - 3$ ($n+4$ δυάρια) που είναι αυτό που ϑέλαμε να δείξουμε.

Στην συνάρτηση Ackermann $A(x,y)$, ο x καθορίζει το "ύψος" των εκθετών, άρα και την τάξη μεγέθους του $A(x,y)$. Π.χ. έχουμε δει ότι $A(2,n) = 2n+3, A(3,n) = 2^{n+3} - 3, A(4,n) = 2^{2^{\cdot^{\cdot^{\cdot^{2}}}}} - 3$. Δηλαδή η $A(2,n)$ είναι πολυωνυμική η $A(n,3)$ είναι εκθετική και η $A(n,4)$ είναι υπερεκθετική.

Επίσης για την A ισχύουν οι παρακάτω ιδιότητες των οποίων οι αποδείξεις ϑα γίνουν με βοήθεια της Αρχής της επαγωγής:
(α) $A(x,y) > y$.
Επαγωγή στο x:
- Για $x = 0, A(0,y) = y + 1 > y$ άρα ισχύει.
- Δέχομαι ότι ισχύει για x δηλαδή $A(x,y) > y$ (1), ϑ.δ.ο. ισχύει για $x+1$, $A(x+1,y) > y$. (3)
Κάνουμε επαγωγή στο y στην (3):
- Για $y = 0, A(x+1,0) > 0, A(x+1,0) = A(x,1) > 0$ που ισχύει καθώς η A είναι πάντα μεγαλύτερη του 0.
- Έστω ότι ισχύει για $y, A(x+1,y) > y$ (2), ϑ.δ.ο. $A(x+1,y+1) > y+1$
$A(x+1,y+1) = A(x,A(x+1,y)) > A(x+1,y)$ λόγω της (1), όμως λόγω της (2) έχω $A(x+1,y) \geq y+1$. Άρα ισχύει το ζητούμενο.

(β) $A(x,y+1) > A(x,y)$.
Επαγωγή στο x:
- Για $x = 0, A(0,y+1) > A(0,y) \Longrightarrow y+2 > y+1$ που ισχύει.
- Έστω ότι ισχύει για $x, A(x,y+1) > A(x,y)$ ϑ.δ.ο. $A(x+1,y+1) > A(x+1,y)$.
$A(x+1,y+1) = A(x,A(x+1,y)) > A(x+1,y)$, λόγω της ιδιότητας (α).

(γ) $y_1 < y_2 \Rightarrow A(x, y_1) < A(x, y_2)$.
Αρκεί να δείξω ότι $A(x, y) < A(x, y + 1)$, που ισχύει απο το (β).

(δ) $A(x + 1, y) \geq A(x, y + 1)$
Επαγωγή στο y:
- Για $y = 0, A(x+1, 0) \geq A(x, 1)$ όμως $A(x+1, 0) = A(x, 1)$ άρα ισχύει η ισότητα.
- Έστω ότι ισχύει για y, $A(x+1, y) \geq A(x, y+1)$ θ.δ.ο. $A(x+1, y+1) \geq A(x, y+2)$
$A(x, y + 1) > y + 1$ από (α), άρα $A(x, y + 1) \geq y + 2$ άρα και $A(x + 1, y) \geq y + 2$
λόγω (γ)$\Rightarrow A(x, A(x+1, y)) \geq A(x, y+2) \Rightarrow A(x+1, y+1) \geq A(x, y+2)$. Άρα ισχύει το ζητούμενο.

(ε) $A(x, y) > x + y$.
Επαγωγή στο x:
- Για $x = 0,\ A(0, y) > y$ ισχύει αφού $A(0, y) = y + 1$
- Έστω ότι ισχύει για $x,\ A(x, y) > x + y$ (1) θ.δ.ο. $A(x + 1, y) > x + y + 1$
Κάνω επαγωγή στο y:
- Για $y = 0,\ A(x + 1, 0) > x + 1 \Rightarrow A(x, 1) > x + 1$ που ισχύει λόγω (1)
- Έστω ότι ισχύει για $y,\ A(x+1, y) > x+y+1$ θ.δ.ο $A(x+1, y+1) > x+y+2$
$A(x+1, y+1) = A(x, A(x+1, y)) > A(x+1, y) > x+y+1$ άρα $A(x+1, y+1) > x + y + 2$, και έτσι αποδείχθηκε το ζητούμενο.

(ζ) $x_1 < x_2 \Rightarrow A(x_1, y) < A(x_2, y)$
Αρκεί να δείξω ότι $A(x + 1, y) > A(x, y)$
$A(x + 1, y) \geq A(x, y + 1) > A(x, y)$ από τις (δ) και (β).

(η) $A(x + 2, y) > A(x, 2y)$
Επαγωγή στο y:
-Για $y = 0, A(x + 2, 0) > A(x, 0)$ που ισχύει αφού $x + 2 > x$
-Έστω ότι ισχύει για $y,\ A(x + 2, y) > A(x, 2y)$ (1) θ.δ.ο. $A(x + 2, y + 1) > A(x, 2y + 2)$
$A(x + 2, y + 1) = A(x + 1, A(x + 2, y)) > A(x + 1, A(x, 2y))$ από (1) και (γ)
$A(x + 1, A(x, 2y)) \geq A(x, A(x, 2y) + 1)$ από (δ), όμως $A(x, A(x, 2y) + 1) \geq A(x, 2y+2)$ αφού $A(x, 2y) > 2y \Rightarrow A(x, 2y) \geq 2y+1 \Rightarrow A(x, 2y)+1 \geq 2y+2 \Rightarrow A(x, A(x, 2y) + 1) \geq A(x, 2y + 2)$.

Ορισμος 2.4.1. *Έστω $f : \mathbb{N} \to \mathbb{N}$. Λέμε ότι η f είναι το πολύ τάξης r αν για κάθε $x \in \mathbb{N},\ f(x) \leq A(r, x)$, όπου A η συνάρτηση Ackermann. Γενικότερα, αν $f : \mathbb{N}^k \to \mathbb{N}$, λέμε ότι η f το πολύ τάξης r αν $f(\vec{x}) \leq A(r, max(\vec{x}))$.*

Δηλαδή η f είναι το πολύ τάξης r, αν η ταχύτητα αύξησης της είναι το πολύ όση και της $A(r, x)$.

Πρόταση 2.4.2. *Για κάθε β.α. συνάρτηση f, υπάρχει $r \in \mathbb{N}$ τέτοιο ώστε η f είναι το πολύ τάξης r.*

Απόδειξη

Δείχνουμε πρώτα ότι ισχύει για τις αρχικές συναρτήσεις που είναι τάξης 0:
(α) Για την συνάρτηση διαδοχής:$S(x) = x + 1 = A(0, x)$, άρα ισχύει η ισότητα.
(β)Για την σταθερά συνάρτηση: $C_0(x) = 0 < x + 1 = A(0, x)$, άρα ισχύει.
(γ)Για την προβολή: $\Pi_{nm}(\vec{x}) = x_m \leq max(\vec{x}) < max(\vec{x}) + 1 = A(0, max(\vec{x}))$.
Άρα η ιδιότητα ισχύει για τις αρχικές συναρτήσεις.
Για την σύνθεση:
Έστω τώρα ότι η f παράγεται με σύνθεση από τις $h, g_1, ..., g_m$,δηλαδή $f(\vec{x}) = h(g_1(\vec{x}), ...g_m(\vec{x}))$, και έστω η h είναι τάξης r και η g_i το πολύ τάξης $s_i, i \leq m$.
Για χάρη συντομίας $g_i(\vec{x}) = y_i$, καθώς επίσης και $max(\vec{x}) = a,\ max(\vec{y}) = b$.
$f(\vec{x}) = h(g_1(\vec{x}), ..., g_m(\vec{x})) = h(y_1, ...y_m) = h(\vec{y})$
Έχουμε ότι

$$h(\vec{y}) \leq A(r, b)\ \kappa\alpha\iota\ g_i(\vec{x}) = y_i \leq A(s_i, a) \quad (1)$$

Θέτουμε

$$s = max\{r, s_1, ..., s_m\} + 3$$

Θα δείξουμε ότι η f είναι το πολύ τάξης s, δηλαδή ότι για κάθε $\vec{x}$,

$$f(\vec{x}) \leq A(s, max(\vec{x})) = A(s, a)$$

Από τις ιδιότητες (α)-(η), έχουμε

$$A(s, a) \geq A(s - 1, a + 1) = A(s - 2, A(s - 1, a)) \quad (2)$$

Επίσης $s - 1 > s_i$, άρα από την (ζ) και την (1)

$$A(s - 1, a) > A(s_i, a) \geq g_i(\vec{x}) = y_i$$

Από τις (2) και (3) και την (γ) έχουμε

$$A(s, a) > A(s - 2, y_i)$$

για κάθε i, άρα από την (ζ), την $s - 2 > r_i$, και την (1)

$$A(s, a) > A(s - 2, b) > A(r, b) \geq h(\vec{y}) = f(\vec{x})$$

άρα αποδείχθηκε το ζητούμενο.
Παρόμοια δουλεύουμε με το σχήμα πρωτογενούς αναδρομής.

Πόρισμα 2.4.3. *Η συνάρτηση Ackermann δεν είναι β.α.*

Απόδειξη. Έστω ότι είναι.Τότε και η συνάρτηση $f : \mathbb{N} \to \mathbb{N}$, όπου $f(x) = A(x, x) + 1$ είναι β.α. Τότε από Πρόταση 2.4.2 η f είναι το πολύ τάξης r για κάποιο r, δηλαδή $f(x) \leq A(r, x)$ για κάθε x. Άρα για $x = r$ έχω $f(r) \leq A(r, r)$ όμως $f(r) = A(r, r) + 1 > A(r, r)$ άτοπο. □

2.5 Αναδρομικές συναρτήσεις

Αν έχουμε $f(\vec{x}) = \begin{cases} (\mu y)g(\vec{x},y) = 0, \text{ αν } (\exists y)(g(\vec{x},y)=0) \\ \text{δεν ορίζεται αλλιώς.} \end{cases}$
Και η g ολική, τότε λέμε ότι η f παράγεται από την g με ελαχιστοποιήση.

Ορισμος 2.5.1. *Το σύνολο R των (μερικών) αναδρομικών συναρτήσεων είναι το ελάχιστο σύνολο C με τι ιδιότητες:*
(1) Το C περιέχει τις βασικές συναρτήσεις $C_k,\ S,\ \Pi_{nm}$.
(2) Το C είναι κλειστό ως προς την πρωτογενή αναδρομή δηλαδή αν $g,\ h \in C$ και η f ορίζεται με πρωτογενή αναδρομή απο τις g και h τότε $f \in C$. Είναι κλειστό ως προς την σύνθεση δηλαδή αν $h : \mathbb{N}^m \to \mathbb{N}$ και $g_i : \mathbb{N}^k \to \mathbb{N}$, $i = 1, ..., m$,ανήκουν στο C τότε η $f : \mathbb{N}^k \to \mathbb{N}$, όπου $f(\vec{x}) = h(g_1(\vec{x}), ..., g_m(\vec{x}))$, ανήκει στο C.
(3)Το C είναι κλειστό ως προς την ελαχιστοποίηση, δηλαδή αν $g \in C$ και η f παράγεται με ελαστοποίηση από την g τότε $f \in C$.

Η απόδειξη για την ύπαρξη του ελαχίστου συνόλου είναι παρόμοια με την αυτή της 2.2.3

Πρόταση 2.5.2. *Η συνάρτηση Ackermann ανήκει στο R.*

Το σχήμα ελαχιστοποίησης μας δίνει μερικές συναρτήσεις καθώς το σχήμα δεν εγγυάται ολικότητα. Λέγοντας "άναδρομική" συνάρτηση θα εννοούμε μερική αναδρομική. Οι ολικές αναδρομικές θα είναι μέρος μόνον των αναδρομικών.

Θέση του Church.Μία συνάρτηση $f : \mathbb{N}^k \to \mathbb{N}$ είναι αλγοριθμική αν και μόνο αν είναι αναδρομική.

Δεν τίθεται θέμα απόδειξης της Θέσης, καθώς δεν είναι μια μαθηματική πρόταση αλλά βασίζεται στην έννοια της αλγοριθμικότητας. Ουσιαστικά αυτό που λέει είναι ότι δεν υπάρχει κάτι πέρα από αυτό που παράγει η σύνθεση, η αναδρομή και η ελαχιστοποίηση. Καθώς υπάρχουν και άλλες μορφές περιγραφής της αλγοριθμικότητας όπως οι Μηχανές Turing αλλά και αυτές αποδεικνύονται ότι είναι ισοδύναμες με την περιγραφή μέσω των αναδρομικών συναρτήσεων κι αυτό κάνει την θέση Church πειστική.

Ουσιαστικά η Θέση μας επιτρέπει να εμπιστευτούμε την διαισθησή μας. Όταν μία συνάρτηση έχει έναν προφανή αλγόριθμο αλλά όχι αυστηρά διατυπωμένο, μπορούμε να την κατάξουμε στις αναδρομικές, χωρίς να ανατρέχουμε στην λεπτομερή τεχνική απόδειξη του πως παράγεται από τα σχήματα του ορισμού της αναδρομής.

2.6 Αναδρομικά και αναδρομικά απαριθμήσιμα σύνολα

Ορισμος 2.6.1. *Ένα σύνολο $X \subseteq \mathbb{N}^k$ λέγεται αναδρομικό αν η χαρακτηριστική του συνάρτηση C_X είναι αναδρομική. Το X θα λέγεται αναδρομικά απαραρηθμήσι-*

μο ή α.α. για συντομία (recursive enumerable ή r.e.),αν $X = rng(f)$ όπου f ολική αναδρομική.

Πρόταση 2.6.2. *(α)Κάθε αναδρομικό σύνολο είναι α.α.*
(β)Το X είναι αναδρομικό αν και μόνο αν τα X και $-X$ είναι α.α.

Απόδειξη. (α) Έστω X αναδρομικό. Τότε η C_X είναι αναδρομική. Θέλουμε να δείξουμε ότι $X = rng(f)$ όπου f ολική αναδρομική. Αν $X = \emptyset$ το X είναι α.α. εξ ορισμού. Έστω $X \neq \emptyset$ και έστω $a \in X$. Ορίζουμε την $f : \mathbb{N} \to \mathbb{N}$ ως εξής:

$$f(n) = n \cdot C_X(n) + cosign(C_X(n)) \cdot a.$$

Προφανώς f ολική αναδρομική και είναι εύκολο να δούμε ότι $rng(f) = X$, αφού αν $n \in X,\ f(n) = n$, ενώ αν $n \notin X, f(n) = a$ όμως $a \in X$
(β)Αν X αναδρομικό τότε και το $-X$ αναδρομικό άρα λόγω του (α) είναι και α.α. Αντίστροφα, έστω $X = rng(f)$ και $-X = rng(g)$, όπου f και g ολικές αναδρομικές. Έστω η συνάρτηση h με $h(2n) = f(n)$ και $h(2n+1) = g(n)$. Προφανώς η h είναι ολική αναδρομική και$rng(h) = \mathbb{N}$. Έστω η συνάρτηση:

$$e(x) = \begin{cases} 1, \text{ αν } (\mu n)(h(n) = x) \text{ άρτιος} \\ 0, \text{ αν } (\mu n)(h(n) = x) \text{ περιττός} \end{cases}$$

Είναι εύκολο να διαπιστώσουμε ότι η e είναι αναδρομική (ορίζεται από περιπτώσεις αναδρομικών συναρτήσεων). Επίσης e είναι ολική και είναι η χαρακτηριστική του συνάρτηση του X. □

Πρόταση 2.6.3. *Έστω f ολική. Η f είναι αναδρομική αν και μόνο αν το σύνολο $G(f)$ είναι αναδρομικό.*

Απόδειξη. Αν f αναδρομική,τότε η συνάρτηση

$$g(m,n) = \begin{cases} 1, \text{ αν } f(m) = n \\ 0, \text{ αν } f(m) \neq n, \end{cases}$$

είναι ολική αναδρομική και $g = C_{G(f)}$. Αντίστροφα η $C_{G(f)}$ είναι αναδρομική, τότε η f γράφεται

$$f(m) = (\mu n)(C_{G(f)}(m,n) = 1)$$

Άρα f αναδρομική. □

Πρόταση 2.6.4. *Έστω $X \subseteq \mathbb{N}$. Το X είναι α.α. αν και μόνον αν υπάρχει αναδρομικό $Y \subseteq \mathbb{N}^2$, τέτοιο ώστε για κάθε $x \in \mathbb{N}$,*

$$x \in X \Longleftrightarrow (\exists n)((n,x)\exists Y)$$

.

Απόδειξη. Έστω $X \subseteq \mathbb{N}$ α.α. Από τον ορισμό, $X = rng(f)$ για μια ολική αναδρομική f. Άρα

$$x \in X \Longleftrightarrow (\exists n)((n,x) \in G(f)).$$

Από την προηγούμενη πρόταση το $G(f)$ είναι αναδρομικό, συνεπώς η ιδιότητα ισχυεί. Αντίστροφα, έστω το X έχει την παραπάνω ιδιότητα. Το Y είναι αναδραμικό,άρα α.α. και έστω $g : \mathbb{N} \to \mathbb{N}^2$ ολική αναδρομική τέτοια ώστε $Y = rng(g)$. Ορίζουμε $f : \mathbb{N} \to \mathbb{N}$ ως εξής: $f(n) =$το δεύτερο μέλος του ζεύγους $g(n)$. Αυτό αυστηρά γράφεται $f(n) = \Pi_{22}(g(n))$. Προφανώς η f είναι ολική αλγοριθμική και $rng(f) = \Pi_{22}(rng(g)) = X$. Άρα το X είναι α.α. □

Πρόταση 2.6.5. *(α)Το X είναι α.α. αν και μόνον αν υπάρχει $1-1$ ολική αναδρομική f τέτοια ώστε $X = rng(f)$.*
(β)Έστω X άπειρο. Το X είναι ανδρομικό αν και μόνον αν υπάρχει αυστηρά αύξουσα ολική αναδρομική f τέτοια ώστε $X = rng(f)$.

Απόδειξη. (α) Η απόδειξη είναι ίδια με της πρότασης 1.1.11.

(β) Η απόδειξη για την κατεύθυνση $\Rightarrow$ είναι ίδια με της πρότασης 1.1.12. Για την άλλη κατεύθυνση, έστω $X = rng(f)$, f αυστηρά αύξουσα αλγοριθμική. Όπως και στο 1.1.14, έχουμε ότι

$$m \in X \iff (\exists n \leq m)(f(n) = m).$$

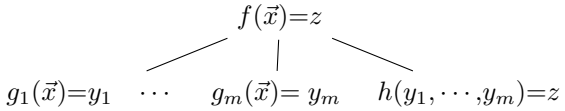
Από πρόταση 2.3.2(κλειστότητα προς φραγμένους ποσοδείκτες), το X ορίζεται από μια αναδρομική σχέση, άρα είναι αναδρομικό. □

2.7 Αριθμητικοποίηση αναδρομικών Συναρτήσεων. Κανονική μορφή Kleene.

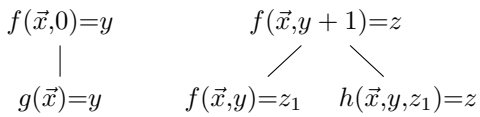
Όπως προκύπει από τον ισοδύναμο ορισμό της παρατήρησης 2.2.4 (2), για κάθε αναδρομική συνάρτηση f υπάρχει μια πεπερασμένη ακολουθία συναρτήσεων $f_1, ...f_n$, όχι μοναδική τέτοια ώστε: (α) $f_n = f$ και (β) για κάθε $i \leq n$, η f_i είτε είναι αρχική, είτε παράγεται από προηγούμενες με σύνθεση, είτε παράγεται από μια προηγούμενη με ελαχιστοποίηση, είτε παράγεται από δύο προηγούμενες $f_k, f_j, (k, j < i)$ με βασική αναδρομή.Η ακολουθία $f_1, ..., f_n$ περίεχει τα βήματα ενός αλγορίθμου της f (ενός από τους πολλούς δυνατούς αλγορίθμους) και όχι στην "έκτατική του οντότητα" της f, ως συνόλου ζευγών.Αν π.χ. $f(x, y, z) = x \cdot (y + z)$, το δεξιό μέλος της ισότητας υποδυλώνει μια διαδικάσία υπολογισμού της:Πρόσθεσε τα y, z και μετά πολλαπλασίασε το αποτέλεσμα με το x. Το ίδιο αποτέλεσμα προκύπτει από μια διαφορετική διαδικασία που περιγράφεται από τον εξής όρο $x \cdot y + x \cdot z$, καθώς έχουμε και διαφορετικό αλγόριθμο υπολογισμού: Πολλαπλασίασε το x πρώτα με το y μετά το x με το z και μετά πρόσθεσε τα εξαγόμενα. Συνεπώς με το σύμβολο f ή g στην πραγματικότητα θα εννοούμε πάντοτε κάποιο πρόγραμμα υπολογισμού της συνάρτησης και όχι το εκτατικό της ισοδύναμο.

Για να υπολογίσουμε την τιμή $f(\vec{x})$, ένας κανονικός τρόπος είναι να ακολουθήσουμε τα βήματα της διαδικασίας που υποδηλώνει ο ορισμός της, ανάγοντας τον υπολογισμό του $f(\vec{x})$ σε προηγούμενους υπολογισμούς, αυτούς σε ακόμα προηγούμενους κ.ο.κ. Τα βήματα αυτά είναι σαν κόμβοι ενός δέντρου υπολογισμού της f στο x, $T(f, \vec{x})$. Κάθε κόμβος αντιστοιχεί σε έναν υπολογισμό, δηλαδή σε μια ισότητα της μορφής $g(\vec{x}) = y$ ενώ οι προπάτορες (predecessors) κάθε κόμβου

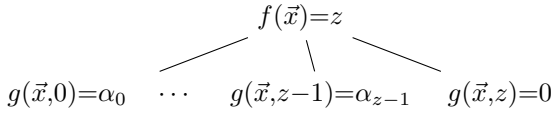
περιέχουν τους προηγούμενους υπολογισμούς. Συγκεκριμένα, το $T(f,\vec{x})$ ορίζεται με τα παρακάτω βήματα (1)-(5).
(1) Η ρίζα του δέντρου είναι ο τελικός υπολογισμός $f(\vec{x})=y$.
(2) Αν η f παράγεται με σύνθεση από τις $h,\ g_1,\cdots g_m$ και $f(\vec{x})=h(g_1(\vec{x}),\cdots g_m(\vec{x})$, τότε για να υπολογίσουμε το $f(\vec{x})$ πρέπει να έχουμε υπολογίσει πρώτα τα $y_1=g_1(\vec{x})\cdots y_m=g_m(\vec{x})$ και $z=h(y_1,\cdots y_m)$, για να θέσουμε τελικά $f(\vec{x})=z$. Έτσι ο κόμβος του $f(\vec{x})$ έχει τους $m+1$ παράγοντες που δείχνει το σχήμα παρακάτω:



(3) Αν η f παράγεται με βασική αναδρομή από τις g και h, δηλαδή $f(\vec{x},0)=g(\vec{x})$ και $f(\vec{x},y+1)=h(\vec{x},y,f(\vec{x},y))$, οι προπάτορες του $f(\vec{x},y)$ φαίνοντε στο παρακάτω σχήμα:



(4) Αν η f παράγεται από την g με ελαχιστοποίηση,δηλαδή
$f(\vec{x})=(\mu y)(g(\vec{x},y)=0)$,οι προπάτορες του $f(\vec{x},y)$ είναι οι παρακάτω:



όπου $\alpha_0\neq 0,\cdots,\alpha_{z-1}\neq 0$.

(5) Τέλος τα "φύλλα" του δέντρου, δηλαδή οι κόμβοι χωρίς προπάτορες, είναι ισότητες που περιέχουν μόνο τις αρχικές συναρτήσεις, δηλαδή ισότητες της μορφής $C_0(x)=0,\ S(x)=x+1$ και $\Pi_{ni}(x_1,\cdots,x_n)=x_i$.

Είναι τώρα φανερό ότι σε κάθε συνάρτηση f, σε κάθε αλγόριθμο που την ορίζει, και σε κάθε είσοδο $\vec{x}$, αντιστοιχεί(μονοσήμαντα) ένα δέντρο υπολογισμού(computation tree). Φυσικά ο υπολογισμός του $f(\vec{x})$ γίνεται από τα φύλλα προς την ρίζα. Η μόνη περίπτωση να μην τερματίσει ο υπολογισμός και να έχουμε δέντρο με άπειρους κόμβους και χωρίς ρίζα, είναι όταν σε κάποιο σημείο του υπολογισμού εφαρμόσουμε το σχήμα ελαχιστοποίησης και η ιδιότητα $g(\vec{x},y)=0$ δεν είναι κανονική, δηλαδή $(\exists\vec{x})(\forall y)(g(\vec{x},y)\neq 0)$.

Αυτό που ονομάζαμε εμπειρικά "βήμα" υπολογισμού ή "μονάδα χρόνου αναμονής" εδώ γίνεται σαφές ως "κόμβος στο δέντρο υπολογισμού" και το να "περιμένουμε k μονάδες χρόνου" εδώ μεταφράζεται στο να "περάσουμε από k κόμβους του δέντρου".

Χρησιμοποιώντας την παραπάνω ιδέα ο S.Kleene (1936) μπόρεσε να δώσει μια

κανονική περιγραφή των αναδρομικών συναρτήσεων. Συγκεκριμένα έδειξε :

Θεώρημα 2.7.1. *(Θεώρημα Κανονικής Μορφής του Kleene) Υπάρχουν μια β.α. συνάρτηση $U : \mathbb{N} \to \mathbb{N}$ και, για κάθε $n \geq 1$, ένα β.α. σύνολο/ιδιότητα $T_n(e, x_1, \cdots, x_n, y) \subseteq \mathbb{N}^{n+2}$ τέτοια ώστε: Για κάθε αναδρομική συνάρτηση $f : \mathbb{N}^n \to \mathbb{N}$, υπάρχει αριθμός e(δείκτης της f) έτσι ώστε να ισχύουν τα ακόλουθα:*
(α) $(x_1, \cdots, x_n) \in dom(f) \iff \exists y\ T_n(e, x_1, \cdots, x_n, y)$,
(β) $f(x_1, \cdots, x_n) = U(\mu y T_n(e, x_1, \cdots, x_n, y))$.

Η ιδιότητα $T_n(e, x_1, \cdots, x_n, y)$ σημαίνει το εξής: Ο y είναι ο κώδικας ενός (δέντρου) υπολογισμού της τιμής που δίνει η συνάτηση με κώδικα e όταν τροφοδοτηθεί με είσοδο τα $x_1, \cdots, x_n$. Αν f συνάρτηση με κώδικα e, παίρνοντας το ελάχιστο κώδικα $y_0 = \mu y T_n(e, x_1, \cdots, x_n, y)$ το $U(y_0)$ είναι η τιμή της f στο $(x_1, \cdots, x_n)$.
Αριθμητικοποίηση των αναδρομικών συναρτήσεων. Αντιστοιχούμε σε κάθε αναδρομική συνάρτηση f, που ορίζεται έναν συγκεκριμένο αλγόριθμο, έναν αριθμητικό κώδικα $\lceil f \rceil$(που εξαρτάται από τον συγκεκριμένο αλγόριθμο) ως εξής:
(1) Για την σταθερή συνάρτηση C_0, $\lceil C_0 \rceil = 0$.
(2) Για την συνάρτηση διαδοχής S, $\lceil S \rceil = 1$.
(3) Για την προβολή,Π_{ni}, $i \leq n$, $\lceil \Pi_{ni} \rceil = \langle 2, n, i \rangle$.
(4) Αν $f = h \circ (g_1, \cdots, g_m)$, $\lceil f \rceil = \langle 3, \lceil h \rceil, \lceil g_1 \rceil, \cdots, \lceil g_m \rceil \rangle$.
(5) Αν η f παράγεται με βασική αναδρομή από τις g και h, $\lceil f \rceil = \langle 4, \lceil h \rceil, \lceil g \rceil \rangle$.
(6) Αν η f παράγεται με ελαχιστοποίηση από την g, $\lceil f \rceil = \langle 5, \lceil g \rceil \rangle$.

Με τα παραπάνω βήματα προφανώς κάθε αναδρομική συνάρτηση f αποκτά έναν αριθμητικό κώδικα $\lceil f \rceil$. Ο $\lceil f \rceil$ λέγεται δείκτης (index) της f.

Παρατήρηση 2.7.2. *1) Είναι σημαντικό να παρατηρήσουμε ότι ο δείκτης e μιας συνάρτησης δεν είναι μοναδικός, αφού μια συνάρτηση μπορεί να οριστεί από πολλούς διαφορετικούς αλγόριθμους, και ο e κωδικοποιεί έναν συγκεκριμένο αλγόριθμο της f. Πιο πρακτικά μπορούμε να θεωρούμε έναν δείκτη της f σαν ένα πρόγραμμα που την υπολογιζεί.*

2) Προφανώς δεν είναι κάθε $x \in \mathbb{N}$ δείκτης μιας αναδρομικής συνάρτησης. Όμως το σύνολο των δεικτών, δηλαδή το σύνολο

$$Ind = \{x : \exists\ \text{αναδρομική } f \text{ τέτοια ώστε } \lceil f \rceil = x\}$$

είναι αναδρομικό. Από την άλλη μεριά θα δούμε ότι αυτό το σύνολο των δεικτών των ολικών συναρτήσεων δεν είναι αλγοριθμικό.

Παραδείγματα Αριθμητικοποίησης

(1) Ο κώδικας $\lceil + \rceil$ για τη συνάρτηση $+$.
Έστω $f(x, y) = x + y$.Έχουμε δει ότι η f ορίζεται με αναδρομή ως εξής:
$f(x, 0) = x = \Pi_{11}(x)$,

$f(x, y+1) = f(x, y) + 1 = S \circ \Pi_{33}(x, y, f(x, y))$,
Όπου $g = \Pi_{11},\ h = S \circ \Pi_{33}$. Άρα έχουμε

$$\ulcorner + \urcorner = \langle 4, \ulcorner g \urcorner, \ulcorner h \urcorner \rangle = \langle 4, \ulcorner \Pi_{11} \urcorner, \ulcorner S \circ \Pi_{33} \urcorner \rangle.$$

Τώρα έχουμε την $S \circ \Pi_{33}$ η οποία παράγεται με σύνθεση από τις $S,\ \Pi_{33}$ άρα ο κωδικός της είναι :

$$\ulcorner S \circ \Pi_{33} \urcorner = \langle 3, \ulcorner S \urcorner, \ulcorner \Pi_{33} \urcorner \rangle.$$

Όμως

$$\ulcorner S \urcorner = 1, \quad \ulcorner \Pi 33 \urcorner = \langle 2, 3, 3 \rangle \ kaι \ \ulcorner \Pi_{11} \urcorner = \langle 2, 1, 1 \rangle.$$

Άρα τελικά έχουμε,

$$\ulcorner + \urcorner = \langle 4, \langle 3, 1, \langle 2, 3, 3 \rangle \rangle, \langle 2, 1, 1 \rangle \rangle.$$

(2) Ο κώδικας $\ulcorner \cdot \urcorner$ την συνάρτηση του πολλαπλασιαμού $\cdot$.
Έχουμε $f(x, y) = x \cdot y$, οπότε ορίζεται με το σχήμα βασικής αναδρομής ως εξής:
$f(x, 0) = 0 = C_0(x)$.
$f(x, y+1) = f(x, y) + x = \Pi_{33}(x, y, f(x, y) + \Pi_{31}(x, y, f(x, y))$,
$g = C_0$ και $h = \Pi_{33} + \Pi_{31} = + \circ (\Pi_{33}, \Pi_{31})$, άρα

$$\ulcorner \cdot \urcorner = \langle 4, \ulcorner C_0 \urcorner, \ulcorner h \urcorner \rangle$$

Έχω τώρα τους κωδικούς των C_0, h
$\ulcorner C_0 \urcorner = 0$, $\ulcorner h \urcorner = \langle 3, \ulcorner + \urcorner, \ulcorner \Pi_{33} \urcorner, \ulcorner \Pi_{31} \urcorner \rangle$, όμως γνωρίζουμε ότι
$\ulcorner + \urcorner = \langle 4, \langle 3, 1, \langle 2, 3, 3 \rangle \rangle, \langle 2, 1, 1 \rangle \rangle, \ulcorner \Pi_{31} \urcorner = \langle 2, 3, 1 \rangle, \ulcorner \Pi_{33} \urcorner = \langle 2, 3, 3 \rangle$, άρα

$$\ulcorner \cdot \urcorner = \langle 4, \ulcorner C_0 \urcorner, \ulcorner h \urcorner \rangle = \langle 4, 0, \langle 3, \ulcorner + \urcorner, \ulcorner \Pi_{33} \urcorner, \ulcorner \Pi_{31} \urcorner \rangle \rangle =$$

$$\langle 4, 0, \langle 3, \langle 4, \langle 3, 1, \langle 2, 3, 3 \rangle \rangle, \langle 2, 1, 1 \rangle \rangle, \langle 2, 3, 3 \rangle, \langle 2, 3, 1 \rangle \rangle.$$

Το θεώρημα 2.7.1 μας λέει ότι όλες οι n-μελείς συναρτήσεις ορίζονται ομοιόμορφα από μία β.α. σχέση $T_n,\ n+2$ θέσεων όπου μία θέση έχει ο κωδικός της συνάρητης, και μια β.α. συνάρτηση U. Το θεώρημα αυτό είναι η βάση για την διαδικασία απαρίθμισης των αναδρομικών συναρτήσεων και των γεγονότων που απορρέουν από αυτήν, δηλαδή την χρήση των αριθμών ως ορίσματα αλλά και σαν κώδικες συναρτήσεων.

Επίσης δίνει την δυνατότητα να χρησιμοποιούμε τα α.α. σύνολα σαν πεδία τιμών ή πεδία ορισμού αναδρομικών συναρτήσεων. Στη σχέση $T_n(e, \vec{x}, y)$, το y παριστά το "μήκος του υπολογισμού" δηλαδή είναι ανάλογο του χρόνου αναμονής το κατηγόρημα $T_n(e, \vec{x}, y)$ είναι ανάλογο του αλγορίθμου αναμονής. Αν η f είναι μια (μερική εν γένει) συνάρτηση με δείκτη e τότε προφανώς

$$f(\vec{x}) = u \Longleftrightarrow (\exists y)[T_n(e, \vec{x}, y) \ \& \ U(y) = u] \quad (1)$$

και η σχέση $T_n(e, \vec{x}, y) \ \& \ U(y) = u$ είναι β.α.

Πρόταση 2.7.3. *Έστω $X \subseteq \mathbb{N}$. Τα παρακάτω είναι ισοδύναμα:*
(α) Το X είναι α.α.
(β) Το X είναι πεδίο τιμών μιας αναδρομικής συνάρτησης.
(γ) Το X είναι πεδίο ορισμού μιας αναδρομικής συνάρτησης.

Απόδειξη. Χωρίς περιορισμό της γενικότητας ας πάρουμε το X άπειρο.
(α)⇒(β): Ισχύει αφού κάθε ολική αναδρομική είναι αναδρομική συνάρτηση άρα ισχύει από 1.1.10
(β)⇒(α): Έστω $X = rng(f)$ και έστω $\lceil f \rceil = e$ ένας δείκτης της f. Τότε $u \in X \iff (\exists n)(f(n) = u)$ τότε από (1) έχουμε

$$u \in X \iff (\exists n)(\exists y)[T_1(e, n, y) \& U(y) = u]$$

Αφού η σχέση $T_1(e, n, y) \& U(y) = u$ είναι αναδρομική από 2.6.4 το X είναι α.α.
(α)⇒(γ): Έστω X α.α. άρα $X = rng(f)$ όπου f ολική αναδρομική. Ορίζουμε $h : \mathbb{N} \to \mathbb{N}$, ως εξής :
$h(n) = (\mu m)(f(m) = n)$.
προφανώς η h είναι (μερική) αναδρομική και $dom(h) = rng(f) = X$.
(γ)⇒(β): Έστω $X = dom(f)$ όπου f αναδρομική. Ορίζουμε την g ως εξής: $g(x) = x + 0 \cdot f(x)$. Προφανώς η g είναι αναδρομική και $dom(g) = dom(f)$. Και για κάθε $x \in \mathbb{N}$,

$$x \in X \iff x \in dom(f) \iff x \in dom(g) \iff g(x) = x \iff x \in rng(g),$$

δηλαδή $X = rng(g)$. □

3 Turing Machines

Η πρώτη κύρια τυποποίηση των αλγοριθμικών συναρτήσεων, είναι μέσω των αναδρομικών συναρτήσεων. Μια δεύτερη μορφή τυποποίησης είναι μέσω των μηχανών Turing.

Ο Alan Turing το 1936 περιέγραψε μια θεωριτική υπολογιστική μηχανή που πήρε το ονομά του και υπήρξε πρόδρομος των πραγματικών υπολογιστών που εμφανίστηκαν 20 χρόνια αργότερα.

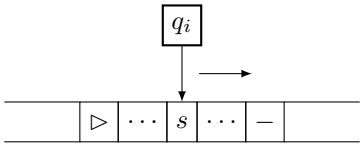
3.1 Γενική περιγραφή των μηχανών Turing

Οι μηχανές Turing είναι ένα αφηρημένο μοντέλο υπολογισμού, παρέχουν έναν ακριβή, επίσημο ορισμό για το τι σημαίνει για να μπορεί μια λειτουργεία να υπολογιστεί. Πολλοί άλλοι ορισμοί έχουν προταθεί με την πάροδο των χρόνων, για παράδειγμα θα μπορούσε ένας να προσπαθήσει να επισημοποίσει ακριβώς τι σήμαινει να τρέχεις ένα πρόγραμμα σε Java σε έναν υπολογιστή με άπειρη μνήμη, αλλά καταλήγουμε ότι όλοι οι γνωστοί ορισμοί υπολογισμού συμφωνούν στο τι είναι υπολογίσιμο και τι όχι. Ο ορισμός των μηχανών Turing μοιάζει να είναι ο πιο απλός. Τα κύρια χαρακτηριστικά του μοντέλου υπολογιστικής μηχανής Turing είναι :

1. Ένα πεπερασμένο ποσό εσωτερικών καταστάσεων.
2. Μια άπειρη ποσότητα αποθήκευσης, εξωτερικών δεδομένων.
3. Ένα πρόγραμμα που καθορίζεται από ένα πεπερασμένο αριθμό οδηγιών σε μια προκαθορισμένη γλώσσα.
4. Αυτο αναφορά: η γλώσσα προγραμματισμού είναι αρκετά εκφραστική για να γράψει έναν διερμηνέα για τα δικά της προγράμματα.

Τα μοντέλα υπολογισμού με αυτά τα βασικά χαρακτηριστικά τείνουν να είναι ισοδύναμα με τις μηχανές Turing, με την έννοια ότι η διάκριση μεταξύ υπολογίσιμων και μη υπολογίσιμων λειτουργειών είναι ίδια σε όλα αυτά τα μοντέλα.

Μια μηχανή Turing μπορεί να θεωρηθεί ως μια πεπερασμένη μηχανή κατάστασης που κάθεται σε μια απείρως μεγάλη ταινία που είναι χωρισμένη σε τετράγωνα κελιά (cells) και κάθε κελί περιέχει από ένα σύμβολο από κάποιο πεπερασμένο αλφάβητο S. Υπάρχει ένας κινούμενος κέρσορας ο οποίος με βάση το το σύμβολο που διαβάζει αυτή τη στιγμή και την τρέχουσα κατάσταση του, η μηχανή Turing γράφει ένα σύμβολο σε αυτή τη θέση (μπορεί και να παραμείνει ίδιο) και ο κέρσορας μετακινείται είτε δεξία, είτε αριστερά είτε μένει στη θέση του και εισέρχεται σε νέα κατάσταση(όπως φαίνεται στο σχήμα πιο κάτω).



Η μετάβαση της μηχανής είναι ένα "πρόγραμμα" που καθορίζει κάθε μια από αυτές τις ενέργειες, (αντικατάσταση συμβόλου, μετακίνηση, τερματισμός) την τρέχουσα κατάσταση και το σύμβολο που η μηχανή διαβάζει αυτή τη στιγμή. Η μηχανή λειτουργεί με ένα "πρόγραμμα", δηλαδή ένα πεπερασμένο σύνολο εντολών. Κάθε εντολή είναι μια τετράδα της μορφής $q_i x y q_j$, όπου q_i, q_j καταστάσεις,$x \in S$ και $y \in S \cup \{L, R\}$(τα L και R είναι από τις λέξεις left και right αντίστοιχα). Δηλαδή οι εντολές έχουν μια από τις εξής μορφές:

$$q_i s_k s_l q_j,\ q_i s_k L q_j, q_i s_k R q_j.$$

Η σημασία τους είναι η εξής:

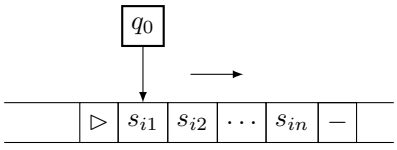
• $q_i s_k s_l q_j$: Όταν η μηχανή είναι σε κατάσταση q_i και διαβάζει το σύμβολο s_k, τότε γράφει σε αυτό το σύμβολο s_l και μεταβαίνει σε κατάσταση q_j.
• $q_i s_k L q_j$: Όταν η μηχανή είναι σε κατάσταση q_i και διαβάζει ένα τετράγωνο που περιέχει το σύμβολο s_k, τότε μετακινείται μια θέση αριστερά και μεταβαίνει σε κατάσταση q_j.
• $q_i s_k R q_j$: Όταν η μηχανή είναι σε κατάσταση q_i διαβάζει s_k τότε μετακινείται μια θέση δεξιά και μπαίνει σε κατάσταση q_j.

Ορισμος 3.1.1. *(Turing 1936) Μια (ντετερμινιστική) μηχανή Turing (MT για συντομία) είναι μαι τριάδα $M = (S, Q, F)$, όπου:*
(α) S είναι ένα πεπερασμένο σύνολο συμβόλων, το αλφάβητο της μηχανής. Ένα από τα σύμβολα του S είναι το $-$ το σύμβολο του κενού και $\triangleright$ το σύμβολο της αρχής.
(β) Q=$\{q_0, \cdots, q_f\}$ είναι ένα πεπερασμένο σύνολο καταστάσεων για την M, $Q \cap S = \emptyset$, όπου q_0 η αρχική κατάσταση, και q_f η τελική.
(γ) F είναι η συνάρτηση μετάβασης(transition function)

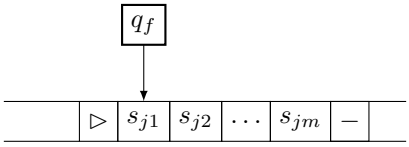
$$F : (Q - \{q_f\}) \times S \to Q \times (S \cup \{R, L\})$$

όπου τα R, L, yes, no, είναι ειδικά σύμβολα που δεν ανήκουν στο S και στο Q.

Η συνάρτηση F προσδιορίζει τις εντολές του προγράμματος. Η μηχανή ξεκινά πάντα σε κατάσταση q_0, και σταματά όταν περιέλθει στην τελική κατάσταση q_f(halt) ή σταματά προσωρινά όταν περιέλθει στις καταστάσεις yes,no.
Οι είσοδοι και οι έξοδοι της μηχανής είναι λέξεις του αλφαβήτου S. Λόγω του συμβόλου -(κενό), οι είσοδοι και οι έξοδοι είναι λέξεις που δεν τελιώνουν με -. Τέτοιες λέξεις θα τις λέμε γνήσιες. Όταν η μηχανή πάρει για είσοδο μία γνήσια λέξη $s_{i1}, \cdots, s_{in}$, θέτουμε την μηχανή σε κατάσταση q_0, και τον κέρσορα να διαβάζει το αριστερό άκρο της λέξης s_{i1} έχουμε δηλαδή το παρακάτω σχήμα



Η μηχανή ξεκινά μέσω του προγράμματος F και άμα σταματήσει η έξοδος είναι η γνήσια λέξη $s_{j1} \cdots s_{jm}$ που είναι γραμμένη στην ταινία και στης οποίας στο αριστερό άκρο(δεξιά από το $\triangleright$) είναι σταματημένος ο κέρσορας σε κατάσταση g_f



Σε κάθε στιγμή η μηχανή βρίσκεται σε μια υπολογιστική φάση ή απλώς φάση. Μια φάση περιγράφεται πλήρως από το περιεχόμενο της ταινίας(τη δεδομενή στιγμή), την εσωτερική κατάσταση q της μηχανής, και από τη θέση του κέρσορα(πιο κελί διαβάζει δηλαδή). Το περιεχόμενο της ταινίας είναι μια απεικόνηση $\sigma : \mathbb{Z} \to S$. Άρα η μια υπολογιστική φάση είναι μια τριάδα της μορφής (q, σ, n), όπου q η κατάσταση, σ το περιεχόμενο του κελιού και $n \in \mathbb{Z}$ ο αριθμός που δείχνει την θέση του κέρσορα. Ο κέρσορας όμως επειδή σχεδόν πάντα δείχνει κάποιο γράμμα της πεπερασμένης γνήσιας λέξης που υπάρχει στην ταινία ή κάποιο από τα διπλανά -, πιο απλά η φάση γράφεται

$$(q, s_{i1} \cdots \underline{s_{ik}} \cdots s_{in})$$

όπου $s_{i1} \cdots s_{in}$ η γνήσια λέξη της ταινίας και το υπογραμμισμένο σύμβολο δείχνει την θέση του κέρσορα. Μπορούμε να παραστήσουμε την φάση ως εξής :

$$s_{i1} \cdots q s_{ik} \cdots s_{in}$$

το σύμβολο της κατάστασης μπαίνει αριστερά του συμβόλου που εξετάζει η μηχανή.

Επειδή για κάθε κατάσταση q_i εκτός της q_f και για κάθε $x \in S$ η F προσδιορίζει ντετερμινιστικά την επόμενη κατάσταση q_j και το βήμα της μηχανής δηλαδή το $F(q_i, x) = (q_j, y)$, από κάθε φάση α που δεν περιέχει την τελική q_f, προχωράει αμέσως στην επόμενη φάση $\acute{\alpha}$ μέσω μια εντολής $q_i x y q_j$. Αν συμβολίσουμε γ την εντολή αυτή, αυτή τη μετάβαση την συμβολίζουμε $\alpha \xrightarrow{\gamma} \acute{\alpha}$ ή $\alpha \to \acute{\alpha}$. Η αρχική φάση είναι της μορφής $q_0 s_{i1} \cdots s_{in}$ ενώ η τελική είναι της μορφής $q_f s_{i1} \cdots s_{in}$. Ένας υπολογισμός είναι μια ακολουθία φάσεων,πεπερασμένη ή άπειρη,

$$\alpha_0 \to \alpha_1 \to \cdots \to \alpha_f,$$

ή

$$\alpha_0 \to \alpha_1 \to \cdots \alpha_i \to \alpha_{i+1} \to \cdots,$$

όπου α_0η αρχική και α_f η τελική φάση, η α_{i+1} είναι η διαδοχή της α_i. Ο πρώτος είναι ένας τερματισμένος υπολογισμός ενώ ο δεύτερος όχι.

Κάθε ΜΤ M ορίζει με φυσικό τρόπο μια (μερική) συνάρτηση $\widehat{M} : S^* \to S^*$, από το σύνολο των γνήσιων λέξεων του S στον εαυτό του, που ορίζεται ως εξής: Έστω $x = s_{i1} \cdots s_{in}$, μια γνήσια λέξη του S. Αν ο υπολογισμός που αρχίζει με την φάση $\alpha_0 = q_0 s_{i1} \cdots s_{in}$ είναι τερματισμένος, και $\alpha_f = q_f s_{j1} \cdots s_{jn}$ είναι η τελική του φάση, τότε

$$\widehat{M}(s_{i1} \cdots s_{in}) = s_{j1} \cdots s_{jn}.$$

Αν ο υπολογισμός δεν έιναι τερματιζόμενος το $\widehat{M}(s_{i1} \cdots s_{in})$ δεν ορίζεται.

Θα δώσουμε παρακάτω μερικά παραδείγματα μηχανών Turing.

Παράδειγμα 1

Ως πρώτο παράδειγμα θα κατασκευάσουμε μια μηχανή Turing που θα λαμβάνει ως είσοδο μία δυαδική συμβολοσειρά και θα προσθέτει 0 στο αριστερό της άκρο. Εδώ η μηχανή έχει 4 καταστάσεις :$q_0, q_1, q_2, \text{halt}$.
q_0: Η q_0 κινείται δεξιά και ετοιμάζεται να γράψει 0.
q_1: Η q_1 κινείται δεξιά και ετοιμάζεται να γράψει 1.
q_2: Η q_2 κινείται αριστερά και αφήνει ως έχει τα σύμβολα μέχρι να φτάσει στην αρχή και να σταματήσει.
Με αυτό τον τρόπο η μηχανή ξεκινάει σε κατάσταση q_0 διαβάζει το σύμβολο της αρχής το αφήνει ίδιο και κινείται δεξιά μέχρι να συναντήσει τον αριθμό 1, τον αλλάξει σε 0 και μπαίνει σε κατάσταση q_1. Η q_1 με την σειρά της κινείται δεξιά μέχρι να συναντήσει κάποιο 0, το κάνει 1 και μπαίνει σε κατάσταση q_0.Με αυτό τον τρόπο μετατοπίζουμε όλα τα ψηφία της συμβολοσειράς μία θέση δεξιά έχοντας βάλει στην αρχή ένα 0. Τέλος όταν η μηχανή διαβάσει το πρώτο - (που σημαίνει

ότι έχει τελειώσει η συμβολοσειρά) μπαίνει σε κατάσταση q_2 και επιστρέφει στην αρχή μπαίνοντας στην τελική κατάσταση halt όπου και σταματάει.

		$F(q,s)$		
q	s	state	symbol	direction
q_0	$\triangleright$	q_0	$\triangleright$	$\rightarrow$
q_0	0	q_0	0	$\rightarrow$
q_0	1	q_1	0	$\rightarrow$
q_0	-	q_2	0	$\leftarrow$
q_1	0	q_0	1	$\rightarrow$
q_1	1	q_1	1	$\rightarrow$
q_1	-	q_2	1	$\leftarrow$
q_2	0	q_2	0	$\leftarrow$
q_2	1	q_2	1	$\leftarrow$
q_2	$\triangleright$	halt	$\triangleright$	-

Παράδειγμα 2

Τώρα θα κατασκευάσουμε μία μηχανή Turing που έχοντας σαν είσοδο μια δυαδική συμβολοσειρά θα δίνει έξοδο τον προηγούμενο της. Όταν η είσοδος θα είναι το 0 θα θεωρήσουμε ως προηγούμενο του τον εαυτό του. Εδωπέρα έχουμε 5 καταστάσεις:$q_0,\ q_1,\ q_2,\ q_3$, halt.

		$F(q,s)$		
q	s	state	symbol	direction
q_0	$\triangleright$	q_0	$\triangleright$	$\rightarrow$
q_0	0	q_0	0	$\rightarrow$
q_0	1	q_1	1	$\rightarrow$
q_0	-	q_3	-	$\leftarrow$
q_1	0	q_1	0	$\rightarrow$
q_1	1	q_1	1	$\rightarrow$
q_1	-	q_2	-	$\leftarrow$
q_2	0	q_2	1	$\leftarrow$
q_2	1	q_3	0	$\leftarrow$
q_3	0	q_3	0	$\leftarrow$
q_3	1	q_3	1	$\leftarrow$
q_3	$\triangleright$	halt	$\triangleright$	-

Εδώ το προγραμμά μας δουλεύει ως εξής. Η q_0 είναι ουσιαστικά ο έλεχγος για να δούμε αν η εισοδό μας είναι ο 0, κινείται δεξιά αν συναντήσει 1 τότε μπαίνει σε κατάσταση q_1, αλλιώς φτάνει στο τέλος της τανιάς μπαίνει στην τελική κατάσταση

και επιστρέφει 0. Η q_1 κινείται δεξιά μέχει να συναντήσει - και να μπει σε κατάσταση q_2. Η q_2 είναι αυτή που ουσιαστικά γράφει τον προηγούμενο, κινείται αριστερά αν συναντήσει 0 τότε το κάνει 1 και παραμένει σε q_2 μέχρι να συναντήσει κάποιο 1 και να το κάνει 0 μπαίνοντας στην τελική κατάσταση q_3 που απλά κινείται αριστερά χωρίς να αλλάζει τα σύμβολα μέχρι να φτάσει στην αρχή και να μπει στην κατάσταση halt και να σταματήσει.

3.2 Turing υπολογίσιμες συναρτήσεις

Παραπάνω είδαμε την γενική μορφή και λειτουργία των μηχανών Turing. Τώρα θέλουμε η μηχανή να επεξεργάζεται μόνο τους φυσικούς αριθμούς, οπότε μπορούμε να χρησιμοποιήσουμε σαν αλφάβητο το $S = \{-, |\}$, όπου $-$ είναι το σύμβολο του κενού και με το $|$, θα αντιπροσωπεύουμε τους φυσικούς αριθμούς με τον εξής τρόπο: Ο 0 γράφεται $|$, ο 1 γράφεται $||$, ο n $\underbrace{|\cdots|}_{n+1}$. Γενικότερα η n-άδα $(n_1, \cdots n_k) \in \mathbb{N}^k$, παρίσταται από την λέξη:

$$\underbrace{|\cdots|}_{n_1+1} - \underbrace{|\cdots|}_{n_2+1} - \cdots - \underbrace{|\cdots|}_{n_k+1}$$

Ορισμος 3.2.1. *Έστω* $f : \mathbb{N}^k \to \mathbb{N}$ *μια συνάρτηση. Η* f *λέγεται Turing-υπολόγιση,*
αν υπάρχει ΜΤ M *με αλφάβητο* $S = -, |$, *τέτοια ώστε για κάθε* $n_1, \cdots, n_k, m \in \mathbb{N}, f(n_1, \cdots, n_k) = m$ *αν και μόνο αν*

$$\widehat{M}(\underbrace{|\cdots|}_{n_1+1} - \underbrace{|\cdots|}_{n_2+1} - \cdots - \underbrace{|\cdots|}_{n_k+1}) = \underbrace{|\cdots|}_{m+1}$$

δηλαδή ο υπολογισμός που ξεκινά με την φάση $q_0 \underbrace{|\cdots|}_{n_1+1} - \underbrace{|\cdots|}_{n_2+1} - \cdots - \underbrace{|\cdots|}_{n_k+1}$, *τερματίζει με την φάση* $q_f \underbrace{|\cdots|}_{m+1}$.

Είναι προφανές ότι κάθε Turing-υπολογίσιμη συνάρτηση είναι αλγοριθμική, με αλγόριθμο το "πρόγραμμα" της μηχανής. Άρα οι Turing-υπολογίσιμες συναρτήσεις είναι μια άλλη εναλακτική προσέγγιση της αλγοριθμικότητας. Παρακάτω θα δούμε και την σχέση τους με τις αναδρομικές συναρτήσεις.

Πρόταση 3.2.2. *Οι αρχικές συναρτήσεις είναι Turing υπολογίσιμες.*

Απόδειξη. Θα κατασκευάσουμε ΜΤ για τον υπολογισμό της S, της σταθεράς $C_0(x)$ και της συνάρτησεις προβολής Π_{nm}.

(α) Για την συνάρτηση του επομένου $S(x) = x + 1$,έχουμε το εξής λαμβάνουμε σαν είσοδο έναν αριθμό της μορφής $\underbrace{|\cdots|}_{n+1}$ και θέλουμε σαν έξοδο τον $\underbrace{|\cdots|}_{n+2}$.
Θεωρούμε πρόγραμμα με τρεις καταστάσεις $q_0,\ q_1,\ q_f$.

		$F(q,s)$		
q	s	state	symbol	direction
q_0	$\triangleright$	q_0	$\triangleright$	$\rightarrow$
q_0	$\|$	q_0	$\|$	$\rightarrow$
q_0	$-$	q_1	$\|$	$\leftarrow$
q_1	$\|$	q_1	$\|$	$\leftarrow$
q_1	$\triangleright$	q_f	$\triangleright$	$-$

Οι καταστάσεις q_0, q_1, q_f συμπεριφέρονται ως εξής, η q_0 ξεκινάει διαβάζει το σύμβολο της αρχής το αφήνει ίδιο και κινείται δεξιά όσπου να συναντήσει το πρώτο κενό σύμβολο $-$,το κάνει $|$ και μπαίνει σε κατάσταση q_1 αλλάζοντας την κατεύθυνση του (κινείται αριστερά), οπότε η q_0 μας αυξάνει τον αριθμό κατά ένα $|$,που είναι αυτό που θέλουμε. Η q_1 κινείται αριστερά μέχρι να συναντήσει το σύμβολο της αρχής $\triangleright$ όπου και μπαίνει στην τελική κατάσταση q_f, που μας δίνει το τελικό επιθυμητό αποτέλεσμα.

(β) Για την σταθερή συνάρτηση $C_0(x)$, παίρνουμε μια είσοδο της μορφής $\underbrace{|\cdots|}_{n+1}$
και θέλουμε έξοδο το $|$. Εδώ θα χρησιμοποιήσουμε τρεις καταστάσεις $q_0,\ q_1,\ q_f$ που λειτουργούν ως εξής:

		$F(q,s)$		
q	s	state	symbol	direction
q_0	$\triangleright$	q_0	$\triangleright$	$\rightarrow$
q_0	$\|$	q_0	$-$	$\rightarrow$
q_0	$-$	q_1	$\|$	$\leftarrow$
q_1	$-$	q_1	$-$	$\leftarrow$
q_1	$\triangleright$	q_f	$\triangleright$	$-$

Εδώ οι καταστάσεις $q_0,\ q_1,\ q_f$ συμπεριφέρονται ως εξής. Η q_0 ξεκινάει διαβάζοντας το σύμβολο της αρχής το αφήνει ίδιο και κινείται δεξιά διαγράφοντας όλα τα $|$, όταν συναντήσει το πρώτο $-$ (που σημαίνει ότι ο αριθμός έχει διαγραφτεί) το κάνει $|$ και μπαίνει σε κατάσταση q_1 αλλάζοντας την κατεύθυνση του. Η q_1 κινείται αριστερά χωρίς να αλλάξει τα σύμβολα μέχρι να πάει στην αρχή όπου και μπάινει στην κατάσταση q_f, όπου μας δίνει για έξοδο το $|$.

(γ) Για την προβολή $\Pi_{nm} : \mathbb{N}^n \to \mathbb{N}$, όπου $\Pi_{nm}(x_1, \cdots, x_n) = x_m$.Εδω θέλουμε να φτιάξουμε ένα πρόγραμμα που να έχει σαν είσοδο

$$\underbrace{|\cdots|}_{x_1+1} - \underbrace{|\cdots|}_{x_2+1} - \cdots - \underbrace{|\cdots|}_{x_n+1}$$

και θέλουμε να σβήνει όλες τις μονάδες εκτός αυτές του m-block , δηλαδή να δείνει έξοδο $\underbrace{|\cdots|}_{x_m+1}$. Αυτό μπορεί χρησιμοποιώντας τις εξής $2n+5$ καταστάσεις, $q_0,\ q_0',\ q_1,\ q_1',\cdots,\ q_n,\ q_n',\ q_{n+1},\ q_{n+1}',\ q_f$. Κάθε ζεύγος $q_i,\ q_i',\ i \leq n, i \neq m$, θα χρησιμεύει για να σβηστούν τα i-blocks,για $i \neq m$,όπως έγινε στο (β).Το ζεύγος $q_m,\ q_m'$ θα χρησιμεύει για την διατήρηση του m-block, ενώ οι υπόλοιπες θα χρειαστούν για να επιστρέψει ο κέρσορας στην αρχή του m-block, και να σταματήσει. □

Παράδειγμα 3

Εδώ θα κάνουμε μια MT για την πρόσθεση, εδώ πρέπει να φτιάξουμε ένα πρόγραμμα που έχει σαν είσοδο δύο αριθμούς της μορφής $n = \underbrace{|\cdots|}_{n+1},\ m = \underbrace{|\cdots|}_{m+1}$ και να μας δείνει έξοδο $\underbrace{|\cdots|}_{n+m+1}$. Οι καταστάσεις μας είναι οι εξής:

		$F(q,s)$		
q	s	state	symbol	direction
q_0	$\triangleright$	q_0	$\triangleright$	$\rightarrow$
q_0	$\|$	q_0	$\|$	$\rightarrow$
q_0	$-$	q_1	$\|$	$\rightarrow$
q_1	$-$	q_2	$-$	$\leftarrow$
q_1	$\|$	q_1	$\|$	$\rightarrow$
q_2	$\|$	q_3	$-$	$\leftarrow$
q_2	$-$	q_2	$-$	$\leftarrow$
q_3	$\|$	q_4	$-$	$\leftarrow$
q_3	$-$	q_3	$-$	$\leftarrow$
q_4	$\|$	q_4	$\|$	$\leftarrow$
q_4	$\triangleright$	q_f	$\triangleright$	$-$

Σε αυτό το πρόγραμμα πρέπει να πάρουμε υπόψη μας ότι θέλουμε $m+n+1$ | και όχι $m+n+2$ που θα πάρουμε σαν αποτελέσμα αν προσθέσουμε όλες τις γραμμές. Οι καστάσεις μας λειτουργούν ως εξής, η q_0 διαβάζει το σύμβολο της αρχής το αφήνει ίδιο και συνεχίζει δεξιά μέχρι να συναντήσει $-$ που είναι ουσιαστικά το κενό ανάμεσα στους δύο αριθμούς στην τανία, το κάνει | και μπαίνει σε κατάσταση q_1 κρατόντας την ίδια κατεύθυνση(δεξιά). Η q_1 προχωράει δεξιά μέχρι να συναντήσει $-$ δηλαδή το τέλος του αριθμού m, το αφήνει $-$ και μπαίνει σε κατάσταση q_2 αλλάζοντας την κατευθυνσή του αριστερά, η q_2 όταν συναντήσει | τον κάνει $-$ και μπαίνει σε καταστάση q_3 κρατόντας την ίδια κατεύθυνση. Η q_3 ουσιαστικά κάνει το ίδιο πράγμα με την q_2 μας διαγράφει την επόμενη μονάδα(για να έχουμε $m+n+1$ μονάδες) και μπαίνει στην σε κατάσταση q_4 για να πάει στην

αρχή και μας δώσει έξοδο τον αριθμό $m+n+1$.

Παράδειγμα 4

Εδώ θα κατασκευάσουμε μια ΜΤ που θα δέχεται σαν είσοδο μία λέξη s, και θα φτιάχνει ένα αντίγραφο της λέξης προς τα δεξιά με δύο κενά αναμεσά τους. Για απλότητα θα δέχεται μια είσοδο της μορφής $\underbrace{|\cdots|}_{n+1}$, και θα δίνει σαν έξοδο

$\underbrace{|\cdots|}_{n+1} - - \underbrace{|\cdots|}_{n+1}$.

Περιγραφή αλγορίθμου. Έστω $s = \underbrace{|\cdots|}_{n+1}$ η είσοδος.

Βήμα 1. Πήγαινε στο δεξί άκρο της s , άφησε δύο κενά και γράψε ένα |.
Βήμα 2. Πήγαινε στο δεξί άκρο της s και σβήστο.
Βήμα 3. Έλεγξε αν το | που σβήστηκε ήταν το τελευταίο αν όχι τότε:
Βήμα 3.1. Πήγαινε γράψε άμεσως μετά το | που έγραψες προηγουμένως ένα ακόμη |. Μετά επανέλαβε το Βήμα 2. Αν ναι τότε:
Βήμα 3.2.Ξαναγράψε την είσοδο s (γράφοντας | προς τα δεξιά μέχρι να συναντήσεις | και μετά σβήσε δύο για να έχουμε 2 κενα ανάμεσα στις λέξεις).

Θα υλοποιήσουμε αυτόν τον αλγόριθμο με μια ΜΤ με 17 καταστάσεις. Θα δούμε με τι έντολες υλοποιείται το κάθε βήμα, ξεκινάμε με την κατάσταση q_0, και στην πορεία εισαγουμε καταστάσεις και ανάλογες εντολές καθοδηγούμενοι από τον αλγόριθμο. Θα πάρουμε για είσοδο το $s = |||$. Το πρόγραμμα όμως που θα ορίσουμε θα δουλεύει για κάθε $\underbrace{|\cdots|}_{n+1}$.

Βήμα 1.Ορίζουμε:
$q_0||| \xrightarrow{q_0|Rq_0} |q_0|| \xrightarrow{q_0|Rq_0} ||q_0| \xrightarrow{q_0|Rq_0} |||q_0- \xrightarrow{q_0-Rq_1} ||| - q_1- \xrightarrow{q_1-Rq_2} ||| - -q_2- \xrightarrow{q_2-|q_3} ||| - -q_3|$.
Βήμα 2.Ορίζουμε:
$|||--q_3| \xrightarrow{q_3|Lq_4} |||-q_4-| \xrightarrow{q_4-Lq_4} |||q_4--| \xrightarrow{q_4-Lq_4} ||q_4|--| \xrightarrow{q_4|-q_5} ||q_5---|$.
Βήμα 3.Ορίζουμε:
$||q_5 - - - | \xrightarrow{q_5-Lq_6} |q_6| - - - |$.

Σε αυτό το σημείο αν ο κέρσορας διαβάζει |, είμαστε στο Βήμα 3.1, ενώ αν διαβάζει −, είμαστε στο Βήμα 3.2. Εδώ είμαστε στο 3.1 άρα:

Βήμα 3.1.Ορίζουμε:
$|q_6| - - - | \xrightarrow{q_6|Rq_7} ||q_7 - - - | \xrightarrow{q_7-Rq_7} || - q_7 - -| \xrightarrow{q_7-Rq_7} || - -q_7 - | \xrightarrow{q_7-Rq_7} || - - - q_7| \xrightarrow{q_7|Rq_8} || - - - |q_8- \xrightarrow{q_8-|q_8} || - - - |q_8| \xrightarrow{q_8|Lq_9} || - - - q_9|| \xrightarrow{q_9|Lq_9} || - -q_9 - || \xrightarrow{q_9--q_4} || - q_4 - ||$.

Βλέπουμε ότι με την τελευταία φάση έχουμε επανέλθει στο βήμα 2 (βρόχος). Αφού εκτελέσουμε δύο φορές τον βρόχο, θα σβηστούν όλα τα | της αρχικής λέξης και θα έρθουμε στη φάση $q_6 - - - - - |||$. Τώρα είμαστε στο
Βήμα 3.2.Ορίζουμε:
$q_6 - - - - - ||| \xrightarrow{q_6 - |q_{10}} q_{10}| - - - -||| \xrightarrow{q_{10}|Rq_{11}} |q_{11} - - - ||| \xrightarrow{q_{11}-|q10} |q_{10}| - - - ||| \xrightarrow{q_{10}|Rq11} ||q_{11} - - - ||| \xrightarrow{q_{11}-|q_{10}} ||q_{10}| - -||| \xrightarrow{q_{10}|Rq_{11}} |||q_{11} - -||| \xrightarrow{q_{11}-|q_{10}} |||q_{10}| - ||| \xrightarrow{q_{10}|Rq_{11}} ||||q_{11} - ||| \xrightarrow{q_{11}-|q_{10}} ||||q_{10}|||| \xrightarrow{q_{10}|Rq_{11}} |||||q_{11}||| \xrightarrow{q_{11}|Lq_{12}} ||||q_{12}|||| \xrightarrow{q_{12}|-q_{13}} ||||q_{13} - ||| \xrightarrow{q_{13}-Lq_{12}} |||q_{12}| - ||| \xrightarrow{q_{12}|-q_{13}} |||q_{13} - -||| \xrightarrow{q_{13}-Lq_{14}} ||q_{14}| - -||| \xrightarrow{q_{14}|Lq_{14}} |q_{14}|| - -||| \xrightarrow{q_{14}|Lq_{14}} q_{14}||| - -||| \xrightarrow{q_{14}|Lq_{15}} q_{15} - ||| - -||| \xrightarrow{q_{15}-Rq_f} q_f||| - -|||.$

Θα δείξουμε τώρα ότι κάθε αναδρομική συνάρτηση είναι Turing-υπολογίσιμη. Αν έχουμε n μηχανές Turing $M_1, \cdots M_n$, τότε μπορούμε να πάρουμε μία M που να προσομοιώνει τη λειτουργία κάθε μιας από τις M_i. Αυτό ουσιαστικά έχει την εξής λειτουργεία είναι σαν να έχουμε n ανεξάρτητα προγράμματα και να κάνουμε ένα που να τα περιέχει σαν υποπρογράμματα. Π.χ. αν $Q_i = \{q_{i0}, \cdots, q_{if}\}$ είναι το σύνολο των καταστάσεων και Γ_i το σύνολο των εντολών της M_i, μπορούμε να θεωρήσουμε ότι τα Q_i είναι ξένα σύνολα (αλλιώς τα θεωρούμε ξένα αντίγραφα), και να πάρουμε ως σύνολο καταστάσεων της M ένα Q τέτοιο ώστε $\bigcup_i Q_i \subseteq Q$. Επίσης τροποποιώντας κατάλληλα τις εντολές Γ_i(ειδικά εκείνες που αφορούν τις τελικές καταστάσεις q_{fi}, ώστε να μην είναι τώρα τελικές), μπορούμε να διαμορφώσουμε το πρόγραμμα Γ της M , ώστε να καλεί ανάλογα με τον σκοπό που επιδιώκουμε τα υποπρογράμματα Γ_i.

Για διευκόλυνση θα θεωρήσουμε το σύμβολο της αρχής $\triangleright$ σαν $-$ (το κενό σύμβολο) και όταν ένας υπολογισμός ξεκινά με είσοδο $s_{i1} \cdots s_{im}$ δηλαδή $-q_0 s_{i1} \cdots s_{im}$, ο κέρσορας δεν κινέται ποτέ αριστερά του $-$ που είναι αριστερά του s_{i1}, δηλαδή διεξάγονται υπολογισμοί μόνο στο δεξιό τμήμα της ταινίας που ορίζει η αρχή της λέξης.

Θεώρημα 3.2.3. *(Turing1936)Κάθε αναδρομική συνάρτηση είναι Turing-υπολογίσιμη.*

Απόδειξη. Θα δείξουμε ότι ισχύει για τις αρχικές συναρτήσεις, την σύνθεση , το σχήμα της προτωγενής αναδρομής και σχήμα της ελαχιστοποιήσης.
(α) Για τις αρχικές συναρτήσεις το δείξαμε στην πρόταση 3.2.2.
(β) Για την σύνθεση. Έστω η f παράγεται με σύνθεση από τις $h, g_1, \cdots, g_m$ δηλαδή

$$f(x_1, \cdots, x_k) = h(g_1(x_1, \cdots, x_k), \cdots, g_m(x_1, \cdots, x_k)),$$

και οι $h, g_1, \cdots g_m$ είναι Turing-υπολογίσιμες με της μηχανές $M_0,\ M_1, \cdots, M_m$ αντίστοιχα. Θεωρούμε μια μηχανή M που προσομοιώνει τις $M_0,\ M_1 \cdots M_m$, με τον τρόπο που περιγράψαμε πιο πριν και όταν δοθεί μια είσοδος

$$s = \underbrace{|\cdots|}_{x_1+1} - \underbrace{|\cdots|}_{x_2+1} - \cdots - \underbrace{|\cdots|}_{x_k+1}$$

κάνει τα εξής:
(1) Ξεκινά με την αρχική φάση $q_0 s$, αντιγράφει την είσοδο μια φορά προς τα δεξιά αφήνοντας δύο κενά αναμεσά τους (όπως κάναμε στο παράδειγμα 4), και πηγένει στην αρχή του αντιγράφου σε κατάσταση q_{m0}(αρχική της M_m), δηλαδή στη φάση

$$s - -q_{m0} s.$$

(2) Τώρα αρχίζει να δουλεύει το υποπρόγραμμα M_m με είσοδο s, και θα δώσει έξοδο t_m(που παριστά την τιμή $(g_m(x_1, \cdots, x_k))$. Επειδή οι υπολογισμοί γίνονται προς τα δεξιά όπως συμφωνήσαμε παραπάνω, η ύπαρξη του αντιγράφου της s προς τα αριστερά του κέρσορα δεν επηρεάζει τον υπολογισμό. Άρα η M είναι σε φάση

$$s - -q_{mf} t_m.$$

(3) Επόμενο βήμα είναι να αντιγράψει τη λέξη t_m, αριστερά της s αφήντοντας δύο κενά αναμεσά τους, αντιγράφει την s πάλι προς τα δεξιά, και πάει στην αρχή του αντιγράφου σε κατάσταση $q_{m-1}, 0$(αρχική της M_{m-1}), δηλαδή σε φάση

$$t_m - -s - -q_{m-1,0} s.$$

(4) Τώρα δουλεύει το υποπρόγραμμα M_{m-1} με είσοδο s και δίνει έξοδο t_{m-1}(η τιμή $g_{m-1}(x_1, \cdots, x_k)$), την οποία μεταφέρει αριστερά της t_m, με ένα κενά αναμεσά τους ,αντιγράφει την είσοδο και έρχεται(όπως στο (3» σε φάση

$$t_{m-1} - t_m - -s - -q_{m-2,0} s.$$

(5) Συνεχίζοντας στο ίδιο μοτίβο καταλήγουμε στη φάση

$$t_1 - t_2 - \cdots - t_m - -q_p s,$$

όπου q_p κάποια κατάσταση.

(6) Τώρα σβήνει τη λέξη s και έρχεται στο αριστερό άκρο της λέξης $t_1 - t_2 - \cdots - t_m$, σε κατάσταση q_{00}(αρχική της M_0),δηλαδή σε φάση

$$q_{00} t_1 - t_2 - \cdots - t_m.$$

(7) Όμως η λέξη $t_1 - t_2 - \cdots - t_m$ παριστά την m-άδα

$$(g_1(x_1, \cdots, x_k), \cdots, g_m(x_1, \cdots, x_k)),$$

η M_0 υπολογίζει τη συνάρτηση h, άρα όταν τερματίσει ο υπολογισμός που ξεκινά με τη φάση $q_{00} t_1 - t_2 - \cdots - t_m$ και δώσει τελική φάση

$$q_{0f} r,$$

η λέξη r παριστά την τιμή $h(g_1(x_1, \cdots, x_k), \cdots g_m(x_1, \cdots, x_k))$ δηλαδή $f(x_1, \cdots, x_k)$. Συνεπώς η M είναι μηχανή Turing για την f.

(γ) Έστω ότι η f παράγεται με το σχήμα πρωτογενής αναδρομή από τις $g,\ h$ δηλαδή
$f(\vec{x}, y) = g(\vec{x})$
$f(\vec{x}, y+1) = h(\vec{x}, y, f(\vec{x}, y))$,
και ότι οι g, h είναι Turing-υπολογίσιμες με τις μηχανές M_0, M_1 αντίστοιχα. Έστω M μια μηχανή που προσομοιώνει τις $M_0,\ M_1$. Έστω s παριστά το διάνυσμα $\vec{x}$ και t_y παριστά τον αριθμό y. Τότε μια είσοδος έχει την μορφή $s - ty$. Ξεκινάμε τη μηχανή με τη φάση

$$q_0 s - t_y.$$

Η M ελέγχει πρώτα τη μορφή του t_y. Αν $y = 0$, δηλαδή $t_y = |$, τότε ο κέρσορας σβήνει το t_0, και επιστρέφει στην αρχή σε κατάσταση q_{00} (αρχική της M_0), δηλαδή μπαίνει σε φάση $q_{00}s$, και εκτελεί τον υπολογισμό που αντιστοιχεί στο $g(\vec{x})$. Αν $y > 0$, τότε η μηχανή γράφει δεξιά της εισόδου, διαδοχικά τις λέξεις $s - t_{y-1},\ s - t_{y-2}, \cdots,\ s - t_0,\ s$, αφήνοντας δύο κενά αναμεσά τους,σβήνει την αρχική είσοδο $s - t_y$, και έρχεται στην αρχή της λέξης s σε κατάσταση q_{00}, δηλαδή σε φάση:

$$s - t_{y-1} - -s - t_{y-2} - - \cdots - -s - t_0 - -q_{00}s.$$

Τώρα μιμείται την M_0, εκτελεί τον υπολογισμό, δίνει έξοδο r_0, μετατοπίζει την r_0 μια θέση αριστερά,και κινούμενος αριστερά, έρχεται στην αρχή της προηγούμενης λέξης σε κατάσταση q_{10}(αρχική της M_1). Είναι δηλαδή σε φάση

$$s - t_{y-1} - -s - t_{y-2} - - \cdots - -q_{10}s - t_0 - r_0.$$

Η λέξη $s - t_0 - r_0$ παριστά το διάνυσμα $(\vec{x}, 0, f(\vec{x}, 0))$,συνεπώς είναι η είσοδος για τη μηχανή M_1 που αντιπροσωπεύει την h. Αν r_1 είναι η έξοδος αυτού του υπολογισμού, ξανά ο κέρσορας μετατοπίζει το r_1 μια θέση αριστερά και έρχεται στην αρχή της προηγούμενης λέξης (αν υπάρχει) σε κατάσταση πάλι q_{10}, δηλαδή στη φάση:

$$s - t_{y-1} - -s - t_{y-2} - - \cdots - -q_{10}s - t_1 - r_1.$$

Προχωράμε με τον ίδιο τρόπο και καταλήγουμε στη φάση:

$$q_{10} - t_{y-1} - r_{y-1}$$

Εκτελεί τον υπολογισμό και δίνει r_y, ψάχνει να δει αν υπάρχει άλλη λέξη αριστερά, δεν βρίσκει και σταματά. Η τελική έξοδος του υπολογισμού το r_y, που το παραστά το $h(\vec{x}, y-1, f(\vec{x}, y-1)) = f(\vec{x}, y)$
(δ) Έστω ότι η f παράγεται με ελαχιστοποιήση από την g δηλαδή

$$f(\vec{x}) = \begin{cases} (\mu y)(g(\vec{x}, y) = 1) \ \text{ αν } \ (\exists y)(g(\vec{x}, y) = 1), \\ \text{δεν ορίζεται αλλιώς,} \end{cases}$$

και η g είναι Turing-υπολογίσιμη με τη μηχανή M_0. Έστω M τώρα μια μηανή που εξομοιώνει την M_0, και η λέξη s παριστά την είσοδο $\vec{x}$ της M. Η M ξεκινά με τη φάση $q_0 s$ και αφήνοντας δύο κενά δεξιά γράφει τη λέξη $s - |$ και έρχεται στο αριστερό της άκρο σε κατάσταση q_{00}, άρα είμαστε σε φάση:

$$s - -q_{00}s - |.$$

Εκτελεί τον υπολογισμό μιμούμενη τη μηχανή M_0. Αν η έξοδος είναι η λέξη $||$(δηλαδή ο αριθμός 1), ο κέρσορας αφήνει μόνο τη λέξη $|$ και σταματά. Αν η έξοδος είναι $\neq ||$, τότε σβήνει την έξοδο και γράφει στα της s τη λέξη $s - ||$, έρχεται στη φάση:

$$s - -q_{00}s - ||,$$

και εκτελεί τον υπολογισμό αν η απάντηση είναι $||$ τότε σταμτά και αφήνει την λέξη $||$ μόνο. Αν είναι $\neq ||$ τότε σβήνει πάλι την έξοδο και συνεχίζει με τον ίδιο τρόπο (μπορεί και επ΄άπειρον), γενικά όταν από την φάση:

$$s - -q_{00}s - \underbrace{|\cdots|}_{n+1},$$

εκτελέσει τον υπολογισμό και η έξοδος είναι $||$, γράφει την λέξη $\underbrace{|\cdots|}_{n+1}$ και σταματά, αλλιώς συνεχίζει με τον ίδιο τροπο.

Η μηχανή M που περιγράψαμε είναι η ζητούμενη. Άρα αποδείξαμε το ζητούμενο. □

Θεώρημα 3.2.4. *(Turing 1936)Κάθε Turing-υπολογίσιμη συνάρτηση είναι αναδρομική.*

Απόδειξη. Στην παράγραφο 2.7 είδαμε ότι μπορούμε να αριθμητικοποιήσουμε τις αναδρομικές συναρτήσεις και τους υπολογισμούς που αναλογούν σε αυτές. Με τον ίδιο τρόπο μπορούμε να αριθμιτικοποιήσουμε τις ΜΤ και τους υπολογισμούς τους, αν και οι ΜΤ δεν ορίζονται επαγωγικά όπως οι αναδρομικές συναρτήσεις. Μια ΜΤ είναι ένα πεπερασμένο σύνολο τετράδων q_ixyq_j, όπου τα $q_i,\ q_j$ ανήκουν στο σύνολο των καταστάσεων της ΜΤ έστω Q_M, και τα x, y στο σύνολο των συμβόλων της ΜΤ έστω S_M όπου τα $Q_M,\ S_M$ είναι ξένα μεταξύ τους δηλαδή $Q_M \cap S_M = \emptyset$. Έστω τώρα $Q,\ S$ δύο ξένα άπειρα αριθμήσιμα σύνολα, το Q περιέχει όλες τις καταστάσεις όλων των ΜΤ και το S περιέχει τα σύμβολα όλων των ΜΤ. Έχουμε δηλαδή $S = \{s_0, s_1, \cdots\},\ Q = \{q_0, q_1, \cdots\}$, όπου ειδικά $s_0 = -,\ s_1 = L,\ s_2 = R,$ και $s_k = 2k$, ενώ $q_i = 2i + 1$. Μια ΜΤ M κωδικοποιείται με έναν αριθμό e τέτοιο ώστε για κάποιο k, τα $(e)_1, \cdots (e)_k$ είναι οι εντολές ενός προγράμματος της M, δηλαδή για κάθε $i \leq k$, η τετράδα

$$((e)_i)_1((e)_i)_2((e)_i)_3((e)_i)_4$$

είναι μια εντολή, τα $((e)_i)_1, ((e)_i)_4 \in Q,\ ((e)_i)_2((e)_i)_3 \in S$ και τα$((e)_i)_j$ πληρούν τις συνθήκες συμβατότητας που απαιτούνται για τις εντολές.

Ανάλογα μπορούμε να ορίσουμε κώδικα για έναν τερματισμένο υπολογισμό, δηλαδή την ακολουθία μιας πεπερασμένης ακολουθίας φάσεων $a_0 \to a_1 \to \cdots \to a_n$, όπου a_0 είναι η αρχική φάση, a_{i+1} είναι επόμενη φάση από την a_i, και η a_n

είναι η τελική φάση. Όπως ορίζει το θεώρημα του Kleene για κάθε $n \geq 1$ ορίζεται μια β.α. αναδρομική ιδιότητα $K_n(e, x_1, \cdots, x_n, y)$ που σημαίνει:

"Το y είναι κώδικας ενός υπολογισμού που πραγματοποιεί η ΜΤ με κωδικά e και είσοδο $x_1 \cdots, x_n$."

Ορίζεται τότε η β.α. συνάρτηση $W : \mathbb{N} \to \mathbb{N}$:

" Αν y ο κώδικας ενός υπολισμού ,$W(y)$ είναι ο αριθμός που παριστά η έξοδος του υπολογισμού. Αλλιώς $W(y) = 0$"

Έστω τώρα μια f μια n-μελής Turing-υπολογίσιμη συνάρτηση. Έστω M μια ΜΤ που την υπολογίζει και έστω e ο κώδικας της M. Τότε προφανώς για κάθε $x_1, \cdots, x_n \in \mathbb{N}$

$$f(x_1, \cdots, x_n) = W(\mu y K_n(e, x_1, \cdots, x_n, y)).$$

Άρα η f είναι αναδρομική. □

4 Πρόβλημα Τερματισμού

Μια συνέπεια της κωδικοποίησης των αναδρομικών συναρτήσεων με φυσικούς αριθμούς είναι η Κανονική μορφή του Kleene, που λέει ότι κάθε n-μελής αναδρομική συνάρτηση ορίζεται ομοιόμορφα από μια β.α. σχέση T_n $n+2$ θέσεων και μια β.α. συνάρτηση U,δηλαδή για κάθε n-μελή αναδρομική f,υπάρχει $e \in \mathbb{N}$ τέτοιο ώστε,για κάθε $x_1, \cdots x_n \in \mathbb{N}$

$$f(x_1, \cdots, x_n) = U(\mu y T_n(e, x_1, \cdots, x_n, y)). \quad (1)$$

Με αυτή ακριβώς την ιδέα αποδείχθηκε ότι κάθε Turing-υπολογίσιμη συνάρτηση είναι αναδρομική.Γενικά η (1) μας δίνει την δυνατότητα να αναφερόμαστε στην f μέσω κάποιου από τους κωδικές της e,ο οποίος λέγεται και δείκτης της f.

Ορισμος 4.0.1. *Συμβολίζουμε με φ_e^n την n-μελή αναδρομική συνάρτηση που έχει δείκτη e, δηλαδή*

$$\varphi_e^n(x_1, \cdots, x_n) = U(\mu y T_n(e, x_1, \cdots, x_n, y))$$

Επίσης συμβολίζουμε με W_e^n το πεδίο ορισμού της φ_e^n.

Για $n = 1$,γράφουμε απλώς φ_e, W_e.Αλλά και όταν ο αριθμός των ορισμάτων είναι ένας δεδομένος n,συχνά γράφουμε φ_e,αντί για φ_e^n.

Πρόταση 4.0.2. *Ένα σύνολο $X \subseteq \mathbb{N}^n$ είναι α.α. αν και μόνον αν υπάρχει $e \in \mathbb{N}$ τέτοιο ώστε $X = W_e^n$*

Απόδειξη. Άμεση από την πρόταση 2.7.3 □

Πρόταση 4.0.3. *(Θεώρημα Απαρίθμησης) Για κάθε n υπάρχει (μερική) αναδρομική συνάρτηση $\Phi : \mathbb{N}^{n+1} \to \mathbb{N}$ που απαριθμεί όλες τις αναδρομικές n-μελείς συναρτήσεις, δηλαδή $\Phi(e, \vec{x}) = \varphi_e(\vec{x})$,για κάθε $\vec{x} \in \mathbb{N}^n$.*

Απόδειξη. Αρκεί να θέσουμε $\Phi(e, \vec{x}) = U(\mu y\, T_n(e, \vec{x}, y))$.Άρα από την (1) $\Phi(e, \vec{x}) = \varphi_e(\vec{x})$, και η Φ είναι προφανώς αναδρομική. □

Πρόταση 4.0.4. *(Θεώρημα Καθολικής Συνάρτησης) Υπάρχει (μερική) αναδρομική συνάρτηση $\Psi : \mathbb{N}^2 \to \mathbb{N}$ που παράγει όλες τις φ_e^n(άρα όλες τις n-μελείς αναδρομικές,για κάθε n),με την εξής έννοια:Για κάθε $e, n, x_1, \cdots, x_n \in \mathbb{N}$,*

$$\Psi(e, \langle x_1, \cdots, x_n \rangle) = \varphi_e^n(x_1, \cdots, x_n).$$

Η Ψ λέγεται καθολική αναδρομική.

Απόδειξη. Αρκεί να θέσουμε

$$\Psi(x,y) = \begin{cases} z, \text{ αν ο } x \text{ είναι δείκτης μιας } n-\text{μελούς συνάρτησης } \varphi_x^n \text{ για} \\ \text{κάποιο } n \text{ ,και αν για κάποια } x_1, \cdots, x_n \ y = \langle x_1, \cdots, x_n \rangle \\ \text{και } \varphi_x^n(x_1, \cdots, x_n) = z \\ \text{δεν ορίζεται αλλιώς} \end{cases}$$

Προφανώς η Ψ είναι αναδρομική και έχει την ζητούμενη ιδιότητα. □

Καθώς η παραπάνω καθολική συνάρτηση Ψ είναι αναδρομική,σημαίνει ότι είναι και Turing-υπολογίσιμη,μια μηχανή M που την υπολογίζει λέγεται καθολική μηχανή Turing(universal Turing machine).Μια τέτοια μηχανή είναι πιο κοντά στην σημερινή έννοια του υπολογιστή,καθώς δεν εκτελεί ένα και μοναδικό πρόγραμμα,όπως οι κοινές MT αλλά μπορεί να τροφοδοτηθεί με οποιοδήποτε πρόγραμμα και αν το εκτελέσει,δηλαδή μπορούμε να την προγραμματίσουμε.

Πρόταση 4.0.5. *Δεν υπάρχει αναδρομική απαρίθμηση των μονομελών,ολικών αναδρομικών συναρτήσεων,δηλαδή το σύνολο $\{x : \varphi_x \text{ είναι ολική}\}$ δεν είναι α.α.*

Απόδειξη. Έστω ότι το $X = \{x : \varphi_x \text{ είναι ολική}\}$ είναι α.α.Τότε υπάρχει ολική αναδρομική σ τέτοια ώστε $X = rng(\sigma)$, δηλαδή μια f είναι ολική αν και μόνον αν $f = \varphi_{\sigma(x)}$ για κάποιο x.Θέτουμε

$$g(x) = \varphi_{\sigma(x)}(x) + 1$$

Προφανώς η g είναι ολική αναδρομική,άρα για κάποιο $e, g = \varphi_{\sigma(e)}$.Τότε για κάθε x,

$$\varphi_{\sigma(x)}(x) = g(x) = \varphi_{\sigma(x)}(x) + 1$$

Η τελευταία για $x = e$ δίνει $\varphi_{\sigma(e)}(e) = \varphi_{\sigma(e)}(e) + 1$,αντίφαση.

□

Όπως είπαμε στην παρατήρηση 2.7.2 (2),το σύνολο Ind των δεικτών αναδρομικών συναρτήσεων είναι αναδρομικό,άρα τίθεται το ερώτημα:Είναι το σύνολο

$$\mathcal{H} = \{(x,y) : y \in W_x\},$$

δηλαδή των (x, y) έτσι ώστε $\varphi_x(y) \downarrow$, καθώς επίσης και το σύνολο

$$\mathcal{K} = \{x : x \in W_x, \}$$

αναδρομικά;Το ερώτημα αυτό είναι γνωστό σαν Πρόβλημα Τερματισμού(Halting Problem),καθώς στην ορολογία των MT,$\varphi_x(y) \downarrow$ σημαίνει ότι η μηχανή που υπολογίζει την φ_x,τερματίζει όταν λάβει την είσοδο y.

Πρόταση 4.0.6. *(Το πρόβλημα Τερματισμού)Το πρόβλημα τερματισμού έχει αρνητική απάντηση,δηλαδή τα $\mathcal{H}$ και $\mathcal{K}$ είναι α.α. αλλά όχι αναδρομικά.*

Απόδειξη. Αν φ είναι η αναδρομική συναρτήση που απαριθμεί τις συναρτήσεις φ_x^1 σύμφωνα με την 4.0.4, τότε $\varphi(x,y) = \varphi_x(y)$ και συνεπώς $\mathcal{H} = dom(\varphi)$,άρα το $\mathcal{H}$ είναι α.α. ως πεδίο ορισμού αναδρομικής συνάρτησης.Όμοια το $\mathcal{K}$ είναι α.α. ως πεδίο ορισμού της συνάρτησης $f(x) = \varphi_x(x)$.Έστω ότι το $\mathcal{H}$ είναι αναδρομικό.Τότε προφανώς και το $\mathcal{K}$ είναι αναδρομικό,άρα και το συμπληρωμά του $-\mathcal{K}$ είναι α.α.,συνεπώς $-\mathcal{K} = W_s$ για κάποιο δείκτη s.Τότε για κάθε x έχουμε

$$x \in W_s \Longleftrightarrow x \in -\mathcal{K} \Longleftrightarrow x \notin \mathcal{K} \Longleftrightarrow x \notin W_x.$$

Για $x = s$ η προηγούμενη δίνει την αντίφαση

$$s \in W_s \Longleftrightarrow s \notin W_s.$$

Άρα τα $\mathcal{H}, \mathcal{K}$ δεν είναι αναδρομικά.

□

Βιβλιογραφία

1 Αναδρομή και Υπολογισημότητα - Γιάννης Ν.Μοσχοβάκης

2 Θεωρία Αναδρομικών Συναρτήσεων και Υπολογισιμότητας - Αθανάσιος Τζουβάρας

3 Definition of a Turing machine - Cornell University(http://www.cs.cornell.edu/courses/cs4820/2012sp/handouts/turingm.pdf)