ΤΕΧΝΟΛΟΓΙΚΟ ΕΚΠΑΙΔΕΥΤΙΚΟ ΙΔΡΥΜΑ ΠΑΤΡΑΣ
ΣΧΟΛΗ ΔΙΟΙΚΗΣΗ ΚΑΙ ΟΙΚΟΝΟΜΙΑΣ
ΤΜΗΜΑ ΔΙΟΙΚΗΣΗΣ ΕΠΙΧΕΙΡΗΣΕΩΝ

Πτυχιακή εργασία

ΘΕΜΑ:

«Μηχανές Turing και Γραμματικές»

ΕΙΣΗΓΗΤΡΙΑ:
ΑΝΤΩΝΟΠΟΥΛΟΥ ΗΡΑ

ΣΠΟΥΔΑΣΤΡΙΕΣ:
ΚΟΝΤΟΥ ΒΕΛΟΥΔΩ
ΖΑΦΕΙΡΑΚΗ ΠΑΝΑΓΙΩΤΑ

ΠΑΤΡΑ 1996

ΠΕΡΙΕΧΟΜΕΝΑ

Αφιέρωση

Στους γονείς μας που μας στήριξαν στις σπουδές μας και στην καθηγήτριά μας κυρία Αντωνοπούλου η οποία μας βοήθησε για να γίνει αυτή η εργασία

Η Θέση του Chrurch

Εισαγωγικά

Ο κύριος σκοπός του μαθήματος αυτού είναι να ορίσουμε - αυστηρά μαθηματικά - την ιδέα (και τα όρια) του υπολογισμού. Μέχρι τώρα, προσεγγίσαμε το θέμα θεωρώντας τον "υπολογισμό" σαν κάτι που γίνεται από "computers". Έτσι πρώτα τυποποιήσαμε (με αυστηρό τρόπο) την ιδέα της υπολγοιστικής μηχανής, με το μοντέλο της "μηχανής Turing" και δείξαμε αφ'ενός οι μηχανές turing μπορούν να σχεδιαστούν έτσι ώστε να κάνουν αρκετά πολύπλοκους υπολογισμούς και αφ΄ετέρου ότι δεν μποροούμε (εύκολα) να αυξήσουμε την δυναμή τους προσθέτοντας "πρόσθετα αρακτηριστικά. Σκοπός μας θα είναι τελικά να πείσουμε τους εαυτούς μας ότι οι μηχανές turing είναι το "τυπικό" ισοδύναμο της έννοιας του αλγόριθμου υπολογισμού, δηλαδή ότι τίποτα δεν μπορεί να θεωρηθεί αλγόριθμος αν δεν μπορεί να παρασταθεί σαν μηχανή turing. Αυτή η "άποψη" δεν είναι θεώρημα, είναι κάτι σαν "αξίωμα" του Chrurch ή σαν θέση ή βασική αρχή: Είναι βέβαια θεωρητικά "πιθανό" , στο μέλλον, κάποιος να καταργήσει το παραπάνω αξίωμα, προτείοντας ένα εναλλακτικό μοντέλο υπολογισμού που θα είναι μεν "δημόσια" αποδεκτό ότι ικανοιποιεί την απαίτηση "πεπερασμένης ποσότητας εργασίας σε κάθε υπολογιστικό βήμα" και παρόλα αυτά να μπορεί - αποδεδειγμένα - να κάνει υπολογισμούς που δεν μπορούν να γίνουν από καμία Turing machine. Κανείς δεν το θεωρεί το παραπάνω πιθανό σήμερα.

Αλλα οι μηχανές Turing δεν είναι η μόνη προσέγγιση στην ιδέα του υπολογισμού. Στο κεφάλαιο αυτό, εξετάζουμε 2 εναλλακτικές λύσεις: (1) τις γραμμικές και τις γλώσσες που δημιουργούν και (2) ένα ειδικό τύπο συναρτήσεων από αριθμούς σε αριθμούς (τις μ-αναδρομικές συναρτήσεις ή μερικές αναδρομικές συναρτήσεις). Θα δείξουμε ότι:

Α. Οι μηχανές Turing μπορούν να προσομοιωθούν από γραμματικές

Β. Οι γραμματικές μπορούν να προσομοιωθούν από μ-αναδρομικές συναρτήσεις

Γ. Οι μ-αναδρομικές συναρτήσεις μπορούν να προσομοιωθούν από μηχανές Turing.

Άρα τα 3 παραπάνω μοντέλα υπολογισμού θα είναι ισοδύναμα. Αυτό είναι μια μερική "απόδειξη" του αξιώματος του Crurch.

Γραμματικές

Ορισμέος: Μια γραμματική είναι μια τετράδα

G= (V, Σ, R, S) όπου

V Είναι ένα αλβάβητο

Σ C V είναι το σύνολο των terminals (Τερματικών συμβόλων)

V-Σ είναι τα μη - τερματικά σύμβολα

S ∈ (V-Σ) είναι το αρχικό σύμβολο

R = πεπερασμένο υποσύνολο του (V*(V-Σ)V*) x V* είναι το σύνολο κανόνων

Γράφουμε $u \rightarrow \upsilon$ αν $(u,\upsilon) \in R$

Γράφουμε $u \Rightarrow \upsilon$ αν για κάποια W_1, W_2, $\in V^{**}$ και κάποιον κανόνα $u' \rightarrow \upsilon'$ έχουμε ότι $u=W_1u'W_2$ και $\upsilon= W_1\upsilon'W_2$ (λέμε: το u "παράγει σε 1 βήμα" το υ).

Το $\Rightarrow$ είναι η ανακλαστική μεταβατική συμπλήρωση του $\Rightarrow$.

Μια λέξη $W \in \Sigma^*$ παράγεται από την G αν και μόνα αν $S \Rightarrow w$.

L(G) είναι το σύνολο όλων των λέξεων στο Σ* που παράγονται από την G.

Μια παραγωγή είναι μια ακολουθία της μορφής

$$w_0 \underset{G}{\Longrightarrow} w_1 \underset{G}{\Longrightarrow} w_2 \underset{G}{\Longrightarrow} \cdots \underset{G}{\Longrightarrow} w_n$$

(και έχει μήκος ή "βήματα")

Παράδειγμα

Η γραμματική G με

V=(S, A, B, C, a, b, c)

Σ= (a, b, c)

R= (S -> e, S-> ABCS,

AB -> BA, AC -> CA, BC -> CB,

BA -> AB, CA -> AC, CB -> BC,

A -> a, B -> b, C -> k)

(όπου e=άδεια λέξη) παράγει τη γλώσσα

L= (W ∈ (a, b, c)*: το w έχει ίσον αριθμό από a, b, c)

(Γιατί;)

Λήμμα 5.1. Έστω M= (K, Σ, δ, s) μια οποιαδήποτε μηχανή Turing.

Τότε υπάρχει μαι γραμματική G έτσι ώστε για κάθε δύο σχήματισμούς (q, u, a,) και (q', u', a', υ') της M:

(q, u, a, υ)

$$(q, u, a, v) \overset{*}{\underset{M}{\vdash}} (q', u', a', v')$$

αν και μόνο αν

$$[u\ q\ a\ v] \Rightarrow [u'\ q'\ a'\ v']$$

όπου [,] είναι νέα σύμβολα, που δεν ανήκουν στο Σ ή στο K (και φυσικά υποθέτουμε Σ K=∅).

<u>Αποδειξη</u> : Η γενική ιδέα είναι να αναπαραστήσουμε το σχηματισμό (q, u, a, v) με την λέξη [uqav], όπου η θέση του "q" μέσα στη λέξη υποδηλώνει συγχρόνως και την θέση της κεφαλής της M. Τυπικά,

$$G = (V, \Sigma, R, S)$$

και

V = K U Σ U {[,] , h, s}

και το R έχει τους εξής κανόνες:

(a) $\forall\ q \notin k$, $a \in \Sigma$, αν $\delta\ (q,\ a) = (p,\ b)$ με $b \in \Sigma$ τότε το R περιέχει τον κανόνα $qa \rightarrow pb$.

(b) Για κάθε $q \in K$ και $a \in \Sigma$, αν το $\delta\ (q,\ a) = (p,\ R)$ όπου $p \in k \cup \{h\}$, τότε το R περιέχει τον κανόνα

$qab \rightarrow apb$

για κάθε $b \in \Sigma$ και

$qa] \rightarrow ap\#]$

(c) Αν $\delta(q,\ a) = (p,\ L)$ τότε το R περιέχει τους κανόνες

$$bqac \rightarrow pbac$$

(για κάθε $b \in \Sigma$ και κάθε $c \in \Sigma \cup \{j\}$)

(τέτοια ώστε $a \neq \#$ ή $c\#j$)

και

$bq\#] \rightarrow pb]$, για κάθε $b^{\bullet} \in \Sigma$ αν $a = \#$.

Είναι εύκολο να διαπιστώσουμε ότι:

$(q,\ u,\ a,\ v) \xrightarrow[M]{} (q',\ u',\ a',\ v')$ αν και μόνο αν

$$[uqav] \Rightarrow [u',\ q',\ a',\ v']$$

Το αυτό ισχύει και για τα συμπληρώματα

? ο.ε.δ.

ΟΡΙΣΜΟΣ

(a) Έστω Σ_0 , Σ_1 δύο αλφάβητα που δεν περιέχουν το #, και έστω f μια οποιαδήποτε συνάρτηση από το Σ_0^* στο Σ_1^*.

H f θα καλείται "γραμματικά υπολογίσιμη" αν και μόνο αν υπάρχει μια γραμματική $G = (V, \Sigma, R, S)$ όπου Σ_0 , $\Sigma_1 \subseteq \Sigma$, και επίσης υπάρχουν λέξεις χ, ψ χ', $\psi' \in V^*$ έτσι ώστε για κάθε

$u \in \Sigma_0^*$, $v \in \Sigma_1^*$

$$f(u) = v \Leftrightarrow xu\psi \xrightarrow{*} \chi' v\psi'$$

(Λέμε ότι η G υπολογίζει την f)

(b) Έστω η συνάρτηση $f: N \to N$. H f ονομάζεται "γραμματικά υπολογίσιμη" αν η συνάρτηση $f^1 : \{I\}^* \to \{I\}^*$ είναι γραμματικά υπολογίσιμη, όπου

$$f^1 (I^n) = I^{f(n)} \quad \forall\, n \in N .$$

Θεώρημα 5.1 Κάθε Turing - υπολογίσιμη συνάρτηση από λέξεις σε λέξεις, ή από αριθμούς σε αριθμούς, είναι γραμματικά υπολογίσιμη.

Απόδειξη: Από το λήμμα 5.1, ό.ε.δ.

Οι πρωτόγονες αναδρομικές συναρτήσεις

Ας μιλήσουμε τώρα για συναρτήσεις από το N^K στο N (δηλ. συναρτήσεις φυσικών αριθμών με Κ - ορίσματα). Θα ορίσουμε πρώτα (για λόγους ευκολίας) ένα γνήσιο υποσύνολο των μ - αναδρομικών συναρτήσεων, τις πρωτόγονες αναδρομικές συναρτήσεις.

ΟΡΙΣΜΟΣ 5.2.1 (Οι Αρχικές Συναρτήσεις)

(α) Η μηδενική συνάρτηση με μηδέν ορίσματα

$\zeta: N^0 \rightarrow N$ είναι η $\zeta(\ ) = 0$

(β) Έστω $\kappa \geq 1$ και $1 \leq i \leq \kappa$. Η οστή συνάρτηση προβολής με κ - ορίσματα

Π^{κ}_{i} είναι η συνάρτηση από το $N^{\kappa} \rightarrow N$ έτσι ώστε

$$\Pi^{\kappa}_{i}\ (n_1,\ n_2,\ \ldots, n_k) = n_i$$

για κάθε $(n_1,\ n_2,\ \ldots, n_k) \in N^k$.

Σημείωση : Έστω $\vec{n} = (n_1,\ n_2,\ \ldots, n_k)$

Αρα $\Pi^{\kappa}_{i}\ (\vec{n}) = n_i\ \forall\ \vec{n} \in N^k$

(γ) Η συνάρτηση διαδοχής σ είναι η συνάρτηση από το N στο N έτσι ώστε:

$\sigma(\eta) = \eta+1$ για κάθε $\eta \in N$.

(δ) Μια αρχική συνάρτηση είναι μια των ζ ή σ, ή μια απ' τις Π^{κ}_{i}.

ΟΡΙΣΜΟΣ 5.2.2

(a) Έστω $l > 0$, $k \geq 0$ και έστω g μια συνάρτηση: $N^l \rightarrow N$ και οι συναρτήσεις $h_1,\ h_2, \ldots,\ h_l$ από το $N^{\kappa} \rightarrow N$.

Έστω η συνάρτηση $f: N^k \rightarrow N$ ώστε $\forall\ \vec{n} \in N^k$, $f(\vec{n}) = g(h_1(\vec{n}), \ldots, h_l(\vec{n}))$.

Λέμε ότι η f προέρχεται από τις g μια συνάρτηση από το $N^{\kappa} \rightarrow N$ και h μια συνάρτηση απ' το $N^{(\kappa+2)} \rightarrow N$.

Έστω f: $N^{(k+1)} \rightarrow N$ μια συνάρτηση, έτσι ώστε για κάθε $\vec{n} \in N$ $f(\vec{n},0) = g(\vec{n})$

και για κάθε $\vec{n} \in N^k$, $m \in N$,

$f(\vec{n}, m+1) = h(\vec{n}, m, f(\vec{n},m))$

Τότε λέμε ότι η f προέρχεται από τις g και h με την πράξη της πρωτόγονης αναδρομής.

ΟΡΙΣΜΟΣ 5.2.3

Μια συνάρτηση καλείται πρωτόγονη αναδρομική (primitive recursive) αν είτε είναι μια αρχική συνάρτηση, είτε μπορεί να παραχθεί από τις αρχικές συναρτήσεις με μια ακολουθία πράξεων σύνθεσης και πρωτόγονης αναδρομής.

(Δηλαδή οι πρωτόγονες αναδρομικές συναρτήσεις είναι η μικρότερη κατηγορία συναρτήσεων που περιέχει τις αρχικές συναρτήσεις και είναι κλειστή ως προς τις πράξεις της σύνθεσης και της πρωτόγονης αναδρομής)

Παράδειγμα 5.1

(α) Η συνάρτηση σ2:N-> N έτσι ώστε σ2 $(n)=n+2 \forall n \in N$, είναι πρωτόγονη αναδρομική. Διότι, αν θέσουμε l=κ=1 και g=h1=σ θα έχουμε σ2(n)=g(h1(n))=σ(σ(n)) (σ- συνάρτηση διαδοχής δηλ. αρχική συνάρτηση και σ2 προκύπτει με με σύνθεση από πρωτόγονες αναδρομικές συναρτήσεις.

(β) Η σ3:N3->N ορισμένη από σ3(n1, n2,n3)=n3+1, για κάθε n ∈ N3 είναι πρωτόγρονη αναδρομική.

Διότι αν l=1, κ=3, $h1=\pi_3^3$ και g=σ έχουμε:

$\sigma_3(n)=\sigma(\pi^3_3(n_1, n_2, n_3)) = n_3+1$

(γ) Έστω η plus $(n_1, n_2) = n_1+n_2$. Τότε η plus είναι πρωτόγονη αναδρομική. Έστω $g = \pi^1_1$ ταυτότητα - συνάρτηση) και h η σ_3 (του (β), άνω). Η plus παράγεται από τις g και h με πρωτόγονη αναδρομή, καθόσον

plus $(n,0) = g(n) = \pi_1^1(n) = n$

και

plus $(n,m+1) = \sigma_3(n, m, \text{plus}(n,m)) = \sigma(n+m) = n+m++1$ $\forall\ n, m \in N$.

(δ) Έστω η plus - 3 $(n_1, n_2, n_3) = n_1+n_2+n_3$

Η plus - 3 είναι πρωτόγονη αναδρομική, διότι plus - 3 (n_1, n_2, n_3) = plus $(\pi_1^3(n_1, n_2, n_3)$, plus $(\pi_2^3(\vec{n}), \pi_3^3(\vec{n}))$

(για $\vec{n}= (n_1,n_2, n_3)$).

Γιατί λέμε ότι οι πρωτόγονες αναδρομικές συναρτήσεις είναι "υπολογίσιμες";

Σίγουρα οι αρχικές συναρτήσεις είναι εύκολα υπολογίσιμες. Αν η f αποκτάται από τις $g, h_1, \ldots ,h_l$ με σύνθεση και οι $g, h_1, \ldots ,h_l$ είναι ήδη γνωστό ότι είναι "υπολογίσιμες", τότε η τιμή $f(\vec{n})$ μπορεί να υπολογιστεί ως εξής: Πρώτα υπολογίζουμε τις τιμές

$m_1 = h_1(\vec{n}), \ldots , m_l = h_l(\vec{n})$

και μετά υπολογίζουμε την $g(m_1, m_2, \ldots ,m_l)$.

Ομοίως αν η f αποκτάται από τις g και h με πρωτόγονη αναδρομή και οι g και h είναι γνωστό ότι είναι υπολογίσιμες, τότε η $f(\vec{n}, m)$ μπορεί να βρεθεί ως εξής:

1) Πρώτα υπολογίζουμε την f $(\vec{n}, 0)$ (δηλαδή την $g(\vec{n})$). Αν m=0, τότε τελειώσαμε. Αλλοιώς, έστω f $(\vec{n}, 0) = m_0$

2) Μετά υπολογίζουμε την f $(\vec{n}, 1)$, δηλαδή την $h(\vec{n}, 0, p_0)$. Έστω p_1 το αποτέλεσμα. Αν m=1, τότε τελειώσαμε.

3) Μετά υπολογίζουμε την f $(\vec{n}, 1)$, δηλαδή την $h(\vec{n}, 1, p_1)$...

.

.

.

Χρησιμοποιώντας κάθε υπολογισθείσα τιμή p_i σαν βοήθεια στον επόμενο υπολογισμό, η τιμή της f(n, m) μπορεί τελικά να υπολογιστεί.

Γκεντελοποίηση

(Κωδικοποίηση κατά Godel)

Εδώ θα ορίσουμε μια σύμβαση για να κωδικοποιούμε λέξεις με χρήση φυσικών αριθμών.

Αυτό θα μας χρειαστεί για να προσωμοιώσουμε τις "γραμματικές συναρτήσεις" με "μερικές αναδρομικές" συναρτήσεις.

Αυτό θα μας βοηθήσει να αποδείξουμε την δεύτερη από τις 3 σχέσεις

(1) ΤΜ $\subseteq$ ΓΣ (δηλαδή ΤΜ=ΓΣ=ΜΑ)

(2) $\Gamma\Sigma \subseteq MA$

(3) $MA \subseteq TM$

όπου

TM = όλες οι "Turing - Υπολογίσιμες" συναρτήσεις.

ΓΣ = όλες οι "Υπολιγίσιμες με Γραμματικές" συναρτήσεις.

MA = όλες οι <u>μερικές</u> <u>αναδρομικές</u> συναρτήσεις.

Έστω Δ ένα τυχόν αλφάβητο.

Έστω $B = |\Delta|+1$

Έστω $d_1, d_2, \ldots, d_{\beta-1}$ μια διάταξη των συμβόλων του Δ.

(δηλ. $\Delta = \{d_1, d_2, \ldots, d_{\beta-1}\}$)

Κάθε λέξη στο Δ* μπορεί να ιδωθεί σαν ένας <u>ακέραιος</u> σε "βάση β".

Αν $w = di_1, di_2, \ldots, di_k \in \Delta^*$

και $1 \leq i_j \leq \beta-1$ για $1 \leq j \leq k$

τότε η συνάρτηση

$$gn: \Delta^* \rightarrow N$$

(<u>κωδικοποίηση κατά Godel</u>)

ορίζεται ως εξής:

$$gn(w) = \beta^{k-1} \cdot i_1 + \beta^{k-2} \cdot i_2 + \ldots + \beta^1 \cdot i_{k-1} + i_k$$

Αν e = άδεια η λέξη, τότε gn(e) = 0.

Το gn(w) καλείται <u>Ο ΑΡΙΘΜΟΣ GODEL</u> του w.

Λήμμα

(α) Το $gn(w)$ ορίζεται για κάθε $w \in \Delta^*$

(β) Αν $w_1 \neq w_2 \Rightarrow gn(w_1) \neq gn(w_2)$

(Απόδειξη προφανής)

Για να κάνουμε το gn "επί", πρέπει να εισαγάγουμε ένα νέο σύμβολο (d_0) που "κωδικοποιεί" το ψηφίο $\varnothing$. Τότε η gn μπορεί να "επεκταθεί" έτσι ώστε

$$gn: \{e\} \cup \Delta(\Delta \cup \{d_0\})^* \rightarrow N$$

και είναι αμφιμονότιμη

(δηλ. το $g_n^{-1}(k)$ ορίζεται $\forall k \in N$).

π.χ.

Έστω $\Delta = \{a, b, c\}$, $\beta=4$

$d_1=a$, $d_2=b$, $d_3=c$

τότε

$gn(a) = 1$

$g_n^{-1}(4) = d_1 d_0 = ad_0$

καθόσον $gn(ad0) = \beta^1 \cdot 1+\beta^0 \cdot 0 = 4$

Λήμμα

(α) Το μήκος μιας λέξης w που ανήκει στο $D = \{e\} \cup \Delta(\Delta\cup\{d_0\})^*$

είναι ο ελάχιστος φυσικός αριθμός l έτσι ώστε $\beta l > gn(w)$

(β) Αν $\chi, \psi \in \Delta^*$ και χ = υπολέξη του ψ τότε $gn(x) \leq gn(\psi)$.

(Απόδειξη εύκολη, παραλείπεται).

ΟΙ ΜΕΡΙΚΕΣ ΑΝΑΔΡΟΜΙΚΕΣ ΣΥΝΑΡΤΗΣΕΙΣ

ΟΡΙΣΜΟΣ

Έστω το $\kappa \geq 0$ και η g μια συνάρτηση με $(\kappa+1)$ ορίσματα. τότε η μη φραγμένη ελαχιστοποίηση της g είναι η συνάρτηση κ-ορισμάτων f έτσι ώστε $\forall \vec{n} \in N^k$,

$$f(\vec{n}) = \begin{cases} \text{ο ελάχιστος } m\text{: } g(\vec{n}, m) = 0 \\ \quad \text{αν ένα τέτοιο } m \text{ υπάρχει} \\ 0 \text{ αλλιώς} \end{cases}$$

Γράφουμε

$$f(\vec{n}) = \mu m[g(\vec{n}, m) = 0]$$

και λέμε ότι η f προκύπτει από την g με "μη φραγμένη ελαχιστοποίηση".

Ορισμός

(α) Μια συνάρτηση, g, $(\kappa+1)$ - ορισμάτων λέγεται "κανονική" αν και μόνο αν, για κάθε $\vec{n} \in N^{\kappa}$ υπάρχει ένα $m \in N$ έτσι ώστε

$$g(\vec{n}, m) = 0$$

(β) Μι συνάρτηση λέγεται μ-αναδρομική (μΑ) αν και μόνο αν μπορεί να αποκτηθεί από τις αρχικές συναρτήσεις ζ, Π_i^{κ} και σ με συνδυασμό των ακόλουθων πράξεων:

- σύνθεση

- πρωτόγονη αναδρομή

- εφαρμογή της "μη φραγμένης ελαχιστοποίησης" σε κανονικές συναρτήσεις.

Σημείωση : Αν μΑ=σύνολο ωρών των μΑ συναρτήσεων, τότε (προφανώς)

$$\Pi.A. \subseteq \mu A$$

Το "$\subseteq$" είναι "γνήσιο υποσύνολο".

Ο λόγος είναι ο εξής (διαισθητικά):

Έστω f_0

f_1

f_2

f_3

.

.

.

μια απαρίθμηση των ΠΑ συναρτήσεων.

(Οι ΠΑ συναρτήσεις μπορούν να απαριθμηθούν λεξικογραφικά, καθόσον οι ορισμοί τους προκύπτπυν "συντακτικά" από τα ζ( ), π_κ^i, σ με εφαρμογή "σύνθεσης" και Π.Α. που επίσης ορίζονται συντακτικά).

Τότε η συνάρτηση

$\psi(n) = f_n(n)+1$

ΔΕΝ είναι Π.Α.

(Αν ήταν θα υπήρχε ένα i έτσι ώστε $\psi=f_i$.

Αλλά τότε,

$$f_i(i) = \psi(i) = f_i(i)+1$$

δηλ. 0=1, άτοπο).

Όμως, η συνάρτηση

$$\psi = \psi(n) = f_n(n)+1$$

είναι μΑ. (Γιατί; αν δεν το καταλαβαίνετε τώρα, θα φανεί αργότερα).

ΒΑΣΙΚΟ ΣΧΟΛΙΟ για τις μΑ συναρτήσεις

Αν κάποιος μας δώσει έναν ορισμό μιας συνάρτησης και μας ρωτήσει "ορίζει ο αριθμός μου μιαν μΑ συνάρτηση", τότε δεν υπάρχει τρόπος να απαντήσουμε.

(Διότι, αν μπορούσαμε να απαριθμήσουμε συστηματικά τις μΑ συναρτήσεις, (σε μια λίστα ψ_0, ψ_1, ψ_2, ...), πάλι θα μας "ξέφευγε" η $\varphi(\chi) = \psi_\chi(\chi)+1$ που θα ήταν "υπολογίσιμη" αλλά ΟΧΙ μΑ).

Βασικά, (όπως θα φανεί και αργότερα), είναι αδύνατον να ποιοί ορισμοί παριστάνουν "κανονικές" συναρτήσεις και ποιοί όχι.

ΟΡΙΣΜΟΣ (Τελεστής μ)

Έστω κ≥0 και g μια συνάρτηση με κ+1 ορίσματα.

Τότε "$\mu m\ [g(\vec{n}, m) = 0]$"

είναι μια έκφραση που παριστάνει τον ελάχιστο m : $g(\vec{n}, m) = 0$.

ΑΝ ΤΕΤΟΙΑ ΥΠΑΡΧΟΥΝ, αλλιώς η έκφραση δεν ορίζεται (δεν επιστρέφει τιμή, "αποκλίνει").

Στην τελευταία αυτή περίπτωση γράφουμε:

$\mu m\ [g(\vec{n}, m) = 0] = \uparrow$

($\uparrow$= το σύμβολο της "απόκλισης").

***ΟΡΙΣΜΟΣ* (ΜΕΡΙΚΕΣ ΑΝΑΔΡ. ΣΥΝΑΡΤΗΣΕΙΣ)**

'Ολες οι (μερικές) συναρτήσεις που προκύπτουν (συντακτικά) από τις $\zeta(\ )$, π_i^k ,σ με εφαρμογή των πράξεων

"σύνθεση"

"πρωτόγονη αναδρομή"

και "μ"

ονομάζονται ΜΕΡΙΚΕΣ ΑΝΑΔΡΟΜΙΚΕΣ ΣΥΝΑΡΤΗΣΕΙΣ (Μ.Α.)

Παρατήρηση:

Οι Μ.Α. συναρτήσεις μπορούν να ταξινομηθούν (απαριθμηθούν) λεξικογραφικά και συστηματικά, σε μια λίστα ψ_0, ψ_1, ψ_2, ... (αρκεί να εφαρμόσουμε συνεχώς τις πράξεις "σύνθεση", "Π.Α.", "μ" στις αρχικές συναρτήσεις και να απαριθμούμε με μέθοδο ζίκ - ζάκ).

Σημείωση:

Λόγω του γεγονότος ότι οι ψ_0, ψ_1, ... είναι "μερικές" (δηλαδή μπορεί να "αποκλίνουν" για κάποια τιμή του ορίσματος), η διαγωνοποίηση του στύλ $\varphi(\chi)=\psi_{\chi}(\chi)+1$ <u>ΔΕΝ</u> μας δημιουργεί προβλήματα.

<u>Ορισμός</u>

Έστω Μ.Α. = σύνολο όλων των Μ.Α. συναρτήσεων.

<u>Λήμμα</u> $\mu A \subseteq MA$

<u>Απόδειξη</u>: Οι μΑ είναι οι "ολικές" εκ των ΜΑ συναρτήσεων.

Ο.Ε.Δ.

<u>ΟΡΙΣΜΟΣ</u> (<u>ΣΥΝΔΕΣΗΣ</u>)

Μια συνάρτηση $f: \Sigma_0^* \to \Sigma_1^*$ είναι μια μΑ (ΜΑ) αν και μόνο αν η συνάρτηση $f': N \to N$ είναι μΑ (ΜΑ) όπου $\forall\ n \in N$

$gn(f\ (g_n^{-1}(n))$ αν $g_n^{-1}(n) \in \Sigma_0^*$

$f'(n)=0$ αν $g_n^{-1}(n) \notin \Sigma_0^*$

<u>ΣΠΟΥΔΑΙΟ ΘΕΩΡΗΜΑ Θ.1</u>

ΚΑΘΕ "ΥΠΟΛΟΓΙΣΙΜΗ ΜΕ ΓΡΑΜΜΑΤΙΚΕΣ" ΣΥΝΑΡΤΗΣΗ ΑΠΟ ΛΕΞΕΙΣ ΣΕ ΛΕΞΕΙΣ ΕΙΝΑΙ μΑ.

<u>Απόδειξη</u>

Πρώτα θα εισάγουμε μια σειρά από χρήσιμες ΠΑ συναρτήσεις και προτασιακούς τύπους:

(1) <u>ΟΡΙΣΜΟΣ</u>

μήκος(n) = "μήκος της λέξης χ: gn(x) = n"

<u>ΙΔΙΟΤΗΤΑ Π.Α.</u>

$\mu\acute{\eta}\kappa o\varsigma\ (n) = \mu i \leq n : n < \beta^i$

(2) Concatenation (συγκόληση, "0")

<u>ΟΡΙΣΜΟΣ</u>

$concat\ (m, n) = gn(g_n^{-1}(m)\, 0\ g_n^{-1}(n))$

<u>Π.Α. ΟΡΙΣΜΟΣ</u>

$concat\ (m, n) = n + m \cdot \beta^{\mu\acute{\eta}\kappa o\varsigma(n)}$

(3) Μέλος του Δ* (Δ* n)

<u>ΟΡΙΣΜΟΣ</u> "Μέλος του Δ*" (n) = αληθές αν και μόνο αν $g_n^{-1}(n) \in \Delta^*$

<u>Π.Α. ΟΡΙΣΜΟΣ</u>

$\Delta^* n \Leftrightarrow \forall\ n_1 \leq n (\forall n_2 \leq n)\ (n \neq concat\ (n_1, n_2)$ <u>ή</u> $n_1 = 0$ <u>ή</u> $\forall n_3 \leq n (n_1 \neq n_3 \cdot \beta))$

Δηλαδή, το $g_n^{-1}(n)$ δεν μπορεί να γραφεί σαν uv με u που <u>τελειώνει</u> σε d_0.

(4) "Μέλος του V*" V*n

(όμοια)

(5) <u>Υποκατάσταση</u> SB $n_1\ n_2\ n_3\ n_4$

<u>ΟΡΙΣΜΟΣ</u>

$SB\ (n_1, n_2, n_3, n_4) =$ αληθές

$$\Leftrightarrow$$

(1) οι λέξεις $g_n^{-1}(n_1) = w_1$

$g_n^{-1}(n_2) = w_2$

$$g_n^{-1}(n_3) = w_3$$

$$g_n^{-1}(n_4) = w_4$$

ανήκουν στο Δ* και (2) για κάποιες λέξεις u, v έχουμε :

$w_1=uw_3v$ και $w_2=uw_4v$

(Δηλαδή $w_1 \Rightarrow w_2$ με $w_3 \rightarrow w_4$)

Το SB είναι ΠΑ διότι:

SB $n_1\ n_2\ n_3\ n_4 \Leftrightarrow$

$\exists\ m \leq$ [$\exists p \leq n_1$ [n_1=concat(m, n_3, p) <u>και</u> n_2=concat(m, n_4, p)]]

και Δ* n_1 και Δ* n_2 και Δ* n_3 και Δ* n_4.

(6) "παράγει σε ένα βήμα" $\Rightarrow$

<u>ΟΡΙΣΜΟΣ</u> Έστω G μια γραμματική με κανόνες

$u_1 \rightarrow v_1, \ldots, u_k \rightarrow v_k$

ΤΟΤΕ

$\Rightarrow$ (n, m) = αληθές
<u>αν και μόνο αν</u>
$g_n^{-1}(n) \in V^*$ <u>και</u> $g_n^{-1}(m) \in V^*$
και $g_n^{-1}(n) \Rightarrow g_n^{-1}(m)$

<u>Π.Α. ορισμός</u>

" $\Rightarrow$ (n, m)"

αν και μόνο αν

V^* n <u>και</u> V^* m <u>και</u>

Sb_{nm} gn(u_1) gn(v_2) ή

Sb_{nm} gn(u_2) gn(v_2) ή

.

.

. ή

Sb_{nm} gn(u_κ) gn(v_κ)

(7) Ο Π.Τ. D:

$D_n \Leftrightarrow g_n^{-1}(n) = \$ x_0 \$ x_1 \ldots \$ x_k \$$

όπου $\kappa \geq 0$ και κάθε $x_i \in V^*$

και $x_i \Rightarrow x_{i+1}$ για $i=0,\ldots,k-1$

(Το D παριστάνει μιαν "παραγωγή" 'η έναν "υπολογισμό" στη γραμματική G).

Το D είναι Π.Α. διότι

D_n = αληθές

$\Leftrightarrow$

Δ^*n και $\exists$ $m \leq n$ [n=concat (gn($), m, gn($))]

και $\forall p \leq n$ ($\forall q \leq n$ ($\forall r \leq n$ ($\forall s \leq n$

($n \neq$concat (p, gn ($), q, gn($), r, gn($),s)

ή ΟΧΙ (V^* q) ή ΟΧΙ (V^* r)

ή $\Rightarrow$ (q, r)

(Δηλαδή, αν το $g_n^{-1}(n)$ μπορεί να γραφεί σαν uvw$z όπου τα v και w δεν περιέχουν $, τότε αναγκαστικά $v \Rightarrow w$).

(7) "Αρχίζει μια παραγωγή" B

$Bpq \Leftrightarrow Dp$ και $q=gn(x_0)$

(8) "Τελειώνει μια παραγωγή" Epq

$Epq \Leftrightarrow Dp$ και $q=gn(X_k)$

(9) Μέλος του Σ_0^*, Σ_1^*

(10) "Βρές τον κωδικό της τελευταίας λέξης" (Extract) (σε μια παραγωγή)

(11) "Υπολογισμός" Cnp

$Cnp \Leftrightarrow n=gn(u)$ & $u \in \Sigma_0^*$

και $gn^{-1}(p)=X_0 -\!\!=> X_1 => \ldots => X_k$ (δηλ. $\$X_0\$X_1\ldots\$X_k\$$)

κσι $X_0=xuy$, $X_k=x'vy'$ και $n \in \Sigma_1^*$.

Τώρα ακολουθεί η "ουσία' της απόδειξης:

Επειδή το Cnp είναι Π.Α., η χαρακτηριστική του συνάρτηση γ είναι Π.Α.

Και επειδή η γραμματική G υπολογίζει την f για κάθε u $\in\Sigma_0^*$ υπάρχει ένα $v \in \Sigma_1^*$ έτσι ώστε

$xuy \Rightarrow x'vy'$ (μάλιστα υπάρχει ένα ακριβώς v, $v=f(u)$.

Άρα , $\forall \nu \in N\ \exists p$: $\gamma((\nu,\pi)=1$

(αν το $gn^{-1}(n)$ $\notin\Sigma_0^*$ τότε Cn0= αληθές και συνεπώς $\gamma(n,0)=1$)

Έστω $\gamma(n,p)=1—\gamma(n,p)$ $\forall n\forall p$. Η γ είναι κανονική συνάρτηση. Άρα έχει νόημα να γράψουμε $\mu p(\gamma(\nu,p)=0)$.

An $f'(n)=extract[(\mu p\ \gamma(n,p)=0])$ τότε $\forall u$ $\mathrm{E}\Sigma_0^*$ έχουμε $f'(gn(u))=gn(f(u))$.

Επίσης, για κάθε $n=gn^{-1}(n)$ $\notin\Sigma_0^*$ έχουμε $f'(h)=0$ (γιατί;). Άρα η f΄ σχετίζεται με την f όπως στον ορισμό σύνδεσης, και η f' είναι μΑ. Άρα η f είναι μΑ/

ΟΕΔ.

Τώρα ας θεωρήσουμε την συνάρτηση f:N-> N.

Αν η f υπολογίζεται με γραμματικές, τότε η συνάρτηση $f';(I)^*\text{->}(I)^*$ (ώστε $f'(I^n)=I^{f(n)}$) είναι επίσης υπολογίσιμη με γραμματικές.

Από την προηγούμενη απόδειδη, η f' είναι μΑ. Δηλαδή, η συνάρτηση

$gn(f'(gn^{-1}(n)))$ αν $gn^{-1}(n)$ $\in(I)^*$

$f''(n)=$ 0 αλλιώς είναι μΑ.

Ας ορίσουμε την ακόλουθη Π.Α. συνάρτηση από N σε N:

enc $(0)=gn_{,}(e)=0$

enc $(n+1)$=Conncat(gn(I), enc(n))

Δηλαδή, enc(n)= ο αριθμός Godel για την "μοναδιαία" αναπαράσταση του n

Θα δείξουμε ότι $f(n)$=μήκος $(f''(enc(n)))$ για κάθε $n \in N$.

Αυτό θα αποδίξη ότι η f είναι μΑ

Απόδειξη Έστω $n \in N \Rightarrow enc(n) = gn(I^n)$

$\Rightarrow gn\text{-}1(enc(n)) \in (I)^*$

$\Rightarrow f''(enc(n)) = f''(gn(I^n))$

$= gn(f'(I^n))$

$= gn(I^{f(n)})$

Άρα μήκος $(gn(If^{(n)}) = f(n)$

6.2. Παγκόσμιες μηχανές turing

Μέχρι τώρα, σχεδιάζαμε για κάθε πρόβλημα μια συγκεκριμένη TM. Θα δείξουμε τώρα ότι οι TM μπορούν να "προγραμματισθούν" δηλαδή ότι η ίδια μηχανή μπορεί να χρησιμοποιηθεί για όλες τις εφαρμογές (όπως οι "πραγματικοί" H/Y).

Για να αρχίσουμε, χρειαζόμαστε έναν γενικό τρόπο περιγραφής TM, έτσι ώστε οι περιγραφές τους να μπορούν να χρησιμοποιηθούν σαν είσοδος σε άλλες TM.

Κάθε TM είναι ένα 4-τιμο (k, Σ, δ, s) και επειδή τα k,Σ είναι πεπερασμένα, θα μπορούσαμε να παραστησουμε το (k, Σ, δ, s) ως μία λέξη. Αλλά το σύνολο αυτών των λέξεων δεν θα ήταν γλώσσα, διότι υπάρχουν TM με k, Σ που είναι αυθαίρετα μεγάλα.

Γι΄αυτό θα κωδικοποιήσουμε τις καταστάσεις και τα σύμβολα της ταινίας σαν λέξεις ενός καθορισμένου αλφαβήτου (πεπερασμένου) που έχει μόνο 2 σύμβολα (c, I) ως εξής:

Ας υποθέσουμε ότι υπάρχουν αριθμήσιμα σύνολα:

$K\infty = (q1, q2, q3\ldots)$

$\Sigma\infty = (a1, a2, a3\ldots)$

Έτσι ώστε για κάθε TM το Σ της είναι ένα πεπερασμένο υποσύνολο του Σ∞ και το K της ένα πεπερασμένο υποσύνολο του K∞.

Ας υιοθετήσουμε τώρα την ακόλουθη αντιστοιχία ρ μεταξύ των συμβόλων μιας TM και λέξεων στο αλφάβητο (I) του ενός γράμματος:

σ	ρ(σ)
q1	I^{i+1}
h	I
L	I
R	I I
ai	I^{i+2}

Σημειώστε ότι δεν υπάρχουν δύο μέλη του K∞U(h) με την ίδια αναπαράσταση. Το ίδιο ισχύει και για το (L,R)UΣ∞.

Έστω τώρα η TM M=(k, Σ, δ, s) με K C K∞ και Σ C Σ∞.

Γράψτε το K σαν $(q_{i1}, q_{i2}, \ldots q_{ik})$ όπου $i_1<i_2<\ldots<i_k$ και το Σ σαν $(a_{j1}, a_{j2}, \ldots, a_{jl})$ με $j_1<j_2<\ldots<j_l$. Επίσης το $s=q_{im}$ με $1<m<k$.

Τώρα θα ορίσουμε kl λέξεις spr με $1 \leq p \leq k$ και $1 \leq r \leq l$: Η κάθε λέξη κωδικοποιεί την τιμή της συνάρτησης μεταφοράς (μετάβασης) επί του ζεύγους (q_{ip}, a_{jr}).

Πιο συγκεκριμένα, έστω

$\delta(q_{ip}, a_{jr}) = (q', b)$

με $q' \in K \cup (h)$ και $b \in \Sigma \cup (L,R)$.

Τότε

$Spr = cw_1cw_2cw_3cw_4c$

όπου

$w1 = p(q_{ip})$

$w2 = p(a_{jr})$

$w3 = p(q')$

$w4 = p(b)$

Τέλος, το p(m) (όλης της μηχανής) είναι η λέξη:

p(m)

$cS_0cS_{11}S_{12}...S_{11}S_{21}S_{22}...S_2...Skl^c$

όπου $S_0=p(s)$, η αρχική κατάσταση

Σημειώστε ότι:

(1) δεν υπάρχουν 2 τέτοιες λέξεις (όπως άνω) που να αντιστοιχούν στην ίδια TM

(2) από το p(M) μπορούμε εύκολα να "κατασκευάσουμε" το M.

(3) Είναι εύκολο να ελέγξουμε αν μια οιαδήποτε λέξη στο (c, I) είναι πράγματι το p κάποιας μηχανής Turing.

Τώρα, θα ορίσουμε μια παγκόσμια μηχανή turing, U, που παίρνει 2 ορίσματα και έχει 3 ταινίες.

(1) Η πρώτη ταινία περιέχει το p(w) όπου w=b1…bn είναι μια είσοδος για μια μηχανή Turing M.

(2) Η δεύτερη ταινία περιέχει το p(M).

(3) Η τρίτη ταινία περιέχει την κωδικοποίηση της κατάστασης της M στην τρέεχουσα φάση του υπολογισμού.

Η U κάνει simulation (προσομοίωση) του υπολογισμού της M πάνω στο W, ως εξής:

(α) Η U δέχεται σαν είσοδο το #p(M)p(w)# στην πρώτη ταινία και οι άλλες δύο ταινίες είναι άδειες.

(β) Η U τοποθετεί το p(M) στην ταινία 2 και αφήνει το p(w) στην ταινία 1, και τοποθετεί το p(s) στην ταινία 3.

(γ) Η U συμβουλεύεται την ταινία 2 για να τροποποιήσει το w (της ταινίας 1) και να σημειώσει το p της νέας κατάστασης στην ταινία 3.

(δ) Η U επαναλαμβάνει το (γ) έως ότου η ταινία 3 περιέχει το p(h). Τότε η U σταματά, και το αποτέλεσμα είναι στην ταινία 1.

Δηλαδή, η U έχει τις εξής ιδιότητες: Για κάθε TM M=(k, Σ, δ, s) και κάθε λέξη $w \in \Sigma^*$.

(I) Αν η M σταματά με αρχική είσοδο w, τότε η σταματά με αρχική είσοδο p(M) p(w) και η ταινία 1 της U (όταν η U σταματήσει) περιέχει την κωδικοποίηση της λέξης που θα ήταν στην ταινία της M όταν η M σταματούσε.

(II) Αν η U σταματά, με αρχική είσοδο p(M)p(w), τότε η M σταματά, με αρχική είσοδο w. Επιπλέον το περιεχόμενο της ταινίας 1 της U όταν η U σταματά, είναι η κωδικοποίηση του περιεχομένου της ταινίας της M όταν η M σταματά.

(Σημείωση: την U μπορούμε να την φανταστούμε και σαν μία απλή TM, μιας μόνο ταινίας).

Άλυτα προβλήματα

Το πρόβλημα του "Αν σταματά μια TM"

(Halting Problem)

Ας θυμηθούμε 2 βασικούς ορισμούς:

Ορισμός Α

Έστω Σ_0 ένα αλφάβητο (που δεν περιέχει το #). Έστω (Y), (N) δύο σύμβολα που δεν ανήκουν στο Σ_0.

Μια γλώσσα $L \subset \Sigma_0^*$ καλείται Turing - αποφασίσιμη (Turing - decidable) an και μόνο αν η συνάρτηση

$XL: \Sigma 0^* \to \{(Y), (N)\}$

είναι Turing - υπολογίσιμη, όπου $\forall W \in \Sigma 0^*$:

$$XL(w) = \begin{cases} (Y) \text{ αν } w \in L \\ (N) \text{ αν } w \notin L \end{cases}$$

Αν M είναι η TM που υπολογίζει την XL τότε η M "αποφασίζει" την L. Επίσης λέμε ότι η M είναι μια διαδικασία απόφασης (ένας αλγόριθμος απόφασης) για την L.

Ορισμός Β.

(α) Λέμε ότι η Turing μηχανή M δέχεται μιαν λέξη w $\in \Sigma_0^*$, αν και μόνο αν η M σταματά με αρχική είσοδο w.

(β) Η TM M δέχεται μιαν γλώσσα $L \subset \Sigma 0^*$ αν και μόνο αν η $L=\{w \in \Sigma_0^*: \text{η M δέχεται το } w\}$.

(γ) Μια γλώσσα L καλείται Turing - αποδεκτή (Turing Acceptable) αν υπάρχει κάποια μηχανή Turing που την δέχεται.

Όμως οι TM είναι πεπερασμένα αντικείμενα και μπορούμε συστηματικά να απαριθμήσουμε τις περιγραφές τους. Άρα το σύνολο των γλωσσών που είναι Turing -αποφασίσιμες (ή Turing-αποδεχτές) είναι αριθμήσιμο. Αλλα τα υποσύνολα του Σ_∞* είναι μη αριθμήσιμα δηλαδή, υπάρχει ένα μη αριθμήσιμο πλήθος γλωσσών. Άρα θα υπάρχουν γλώσσες που δεν είναι Turing - αποφασίσιμες ή Turing - αποφδεχτές.

Το ενδιαφέρον μας για τέτοιες γλώσσες δεν είναι απλώς περιέργεια για την υπολογιστική δύναμη των TM. Έχουμε ήδη επιχειρηματολογήσει ότι οι TM είναι η μαθηματική περιγραφή των "αλγορίθμων". Έτσι, μια γλώσσα που δεν είναι Turing αποφασίσιμη. (Θα τις λέμε μη - αποφασίσιμες) έχει την ιδιότητα ότι δεν υπάρχει κανείς γενικός και συστηματικός τρόπος για να πούμε ποιες λέξεις ανήκουν στην γλώσσα και ποιες όχι.

Θα δούμε ότι πολλά "φυσικά" ερωτήματα πανω σε γλώσσες ή γραμματικές κ.λ.π. είναι μη - αποφασίσιμα (ή μη αποδεκτά). Τέτοια ερωτήατα αντιστοιχούν σε προβλήματα που ονομάζουμε Άλυτα προβλήματα.

Θα δώσούμε πολλά παραδείγματα τέτοιων άλυτων προβλημάτων, μθοδολογικά, το πρώτο που θα παρουσιάσουμε, έχει μια στριφνή και μάλλον, "τεχνητή" απόδειξη. Είναι το περίφημο "halting problem", το πρόβλημα του ανα μια TM M σταματά σε μία είσοδο.

Αλλά πρώτα, δύο βασικά θεωρήματα.

Θεώρημα 7.1.1. Κάθε turing- αποφασίσιμη γλώσσα είναι turing αποδεκτή

Απόδειξη

Αν η L αποφαασίζεται από την TM M τότε η L γίνεται δεκτή από την μηχανή M΄ του συστήματος

$$\begin{array}{l} \quad\quad N \\ >ML \rightarrow N \end{array}$$

Αυτή η μηχανή στην αρχή προσομοιώνει την M. Όταν η M σταματήσει, αν η απάντηση είναι (N), τότε η M' πάει σε άπειρο loop αλλιώς η M' σταματάει.

Θεώρημα 7.1.2.

Αν η L είναι Turing - αποφασίσιμη γλώσσα τότε το συμπλήρωμα της $L=\Sigma_0{}^*- L$ είναι turing αποφασίσιμο.

Απόδειξη

Η L αποφασίζεται από την M που τρέχει την M (που αποφασίζει την L) και στο τέλος εναλλάσσει τις απαντήσεις (Y) και (N).

Θα δείξουμε (αργότερα) ότι οι αντίστοιχες (των 7.1.1., 7.1.2) προτάσεις ΔΕΝ ισχύουν γενικά για Turing - αποδεκτές γλώσσες.

Ένας λόγος γι΄αυτό είναι και ο εξής:

Αν ήταν δυνατόν για κάθε TM και κάθε είσοδο να προβλέψουμε αν η TM σταματά τελικά γι'αυτή την είσοδο, τότε κάθε Turing αποδεκτή γλώσσα θα ήταν και Turing -

αποφασίσιμη. Διότι, αν η TM M1 δεχόταν τη γλώσσα L τότε θα σχεδιάζαμε την M2 που (α) προβλέπει αν η M1 σταματα με είσοδο w ή όχι και (β) σταματά και η ίδια με αποτέλεσμα (Y) ή (N) εξαρτώμενο από το αν η M1 σταματά ή όχι

Άρα, τελικά, το αν κάθε turing - αποδεκτή γλώσσα είναι και turing - αποφασίσιμη, ανάγεται στο αν υπάρχει κάποια μέθοδος, κάποιος αλγόριθμός, κάποιας (σούπερ) TM, που να προβλέπει αν μια TM M σταματά όταν η αρχική είσοδος είναι w.

Θεώρημα 7.1.3. (Halting Problem)

Η γλώσσα

Ko={p(M)p(w): η TM M δέχεται το w, δηλαδή σταμτά με είσοδο w).

δεν είναι Turing -αποφασίσιμη.

Απόδειξη

Έστω ότι ("χάριν της εις άτοπον απαγωγής") η Ko ήταν Turing αποφασίσιμη.

Αλλά τότε, και η

K1={p(M): Η TM M δέχεται την λέξη p(M)}

θα ήταν turing αποφασίσιμη. Διότι, αν η K0 αποφασίζεται από την M0 τότε κατασκευάζουμε την M1 που αποφασίζει την K1 ως εξής: πρωτα μετασχηματίζει την είσοδή της #w# σε #wp(w)# και μετά τρέχει την Mo. Εξ ορισμού, η Mo θα αποφάσιζε (Y) με είσοδο #wp(w)# αν και μόνο αν

(α) w=p(M) για κάποια TM M και

(β) η M δέχεται το w, δηλαδή το p(M).

Αλλά αυτός είναι ο αριθμός της K1. Άρα, αρκεί να δείξουμε ότι η K1 είναι μη - αποφασίσιμη.

Αλλά αν η K1 ήταν αποφασίσιμη, το ίδιο θα ίσχυε και για το συμπλήρωμα της K1 (λόγω 7.1.2.):

όπου

K1=w ∈ I, *: είτε το w δεν είναι η κωδικοποίηση p καμμιάς TM M είτε w=p(M) για κάποια M η οποία δεν δέχεται το p(M).

Αλλά η K1 δεν είναι καν Turing - αποδεκτή: Γιατί, αν υπήρχε μια TM M* που αποδέχεται την K1, τότε για να δούμε τι θα έκανε η M* για είσοδο (M*).

Έχουμε

(1) p(M*) ∈ K1 ⇔ η M* δέχεται το p(M*), εξ ορισμού της M*

(2) p(M*) ∈ K1 ⇔ η M* δεν δέχεται το p(M*) λόγω ορισμού της K1.

Δηλαδή, η M* δέχεται το p(M*) ⇔ η M* δεν δέχεται το p(M*) Αλλά αυτό είναι αντίφαση.

Άρα, η K_0 δεν είναι turing - αποφασίσιμη.

Σημείωση

Η K_0 είναι Turing - αποδεκτή. Πράγματι, την K_0 δέχεται μία απλή τροποποίηση της παγκόσμια μηχανής Turing U, η οποία προσομοιώνει την M (για είσοδο p(M)p(w)) σε είσοδο w και όταν η M πάει να σταματήσει, σταματά και η U.

Είναι εύκολο να δούμε (πως;) ότι και η K_1 είναι turing αποδεκτή.

Αλλά η K δεν είναι. Συμπερασματικά μέχρι τώρα

Γλώσσα	Turing αποφασίσιμη	Turing αποδεκτή
K0	ΟΧΙ	ΝΑΙ
K1	ΟΧΙ	ΝΑΙ
K1	ΟΧΙ	ΟΧΙ

Με άλλα λόγια, υπάρχουν προβλήματα που δεν επιδέχονται αλγοριθμική λύση. Τέτοια προβλήματα καλούνται άλυτα προβλήματα. Προβλήματα που επιδέχονται αλγοριθμική λύση λέγονται επιλύσιμα και οι αλγόριθμοι που τα λύνουν λέγονται διαδικασίες απόφασης (decision procedures).

Άλυτα προβλήμτα για μηχανές turing

(α) δοθείσας μιας TM M και μιας εισόδου w, σταματά η M με είσοδο w;

(β) Για μια ορισμένη TM M (συγκεκριμένη αν δοθεί μια αυθαίρετη λεξη, σταματά η M με είσοδο;

(γ) Δοθείσας μιας TM M, σταματά η M όταν η είσοδος είναι η άδεια ταινία;

(δ) Δοθείσας μιας TM M, υπάρχει τουλάχιστον μια λέξη για την οποία η.M σταματά.

(ε) Δοθείσας μιας TM M, σταματά η M για κάθε λέξη;

(στ) Δοθέντων δύο TM M1, M2, σταματάνε στις ίδιες λέξεις εισόδου;

(η) Δοθείσας μιας TM M, είναι η γλώσσα, που η M δέχεται, κανονική; είναι άνευ συμφραζομένων; είναι Turing - αποφασίσιμη;

Αλυτα προβλήματα για αναδρομικές συναρτήσεις

(α) Δοθείσας μια πρωτόγονης αναδρομικής συνάρτησης, είναι αυτή η συνάρτηση κανονική;

Άλυτα προβλήματα για Γενικές γραμματικές

(α) Για μια αυθαίρετη (δοθείσα) γραμματική G και λέξη w, ανήκει το στη γλώσσα L(G);

(β) Για αυθαίρετες (δοθείσες) γραμματικές G1 και G2, ισχύει ότι L(G1)=L(G2);

(γ) Για δοθείσα γραμματική G, ισχύει ότι L(G)=0; (usolvability of emptiness).

Συστήματα thue (θούε)

Ορισμός: Ένα σύστημα thue είναι ένα πεπερασμένο σύνολο από μη διατεταγμένα ζεύγη λέξεων. Μπορούμε να το παραστήσουμε σαν μια λίστα.

W_1, u_1

W_2, u_2

.

.

.

Wn, un

αλλά να θυμόμαστε ότι η διάταξη των ζευγών και της λίστας δεν παίζουν ρόλο.

Αν ζ είναι ένα σύστημα Thue, τότε αυτό ορίζει μια σχέση ισοδυναμίας ανάμεσα σε λέξεις ως εξής

Έστω Σ το αλφάβητο των συμβόλων των λέξεων του ζ. Δύο λέξεις x, y του Σ* λέγονται ισοδυναμίες άμεσα (λόγω ζ) (και γράφουμε x ζ y) αν και μόνο αν ∃I, $1 \leq i \leq n$ ώστε για κάποια Z_1, Z_2, ∈ Σ* είτε ($x=Z_1 W_i Z_2$ και $y=Z_1 u_i Z_2$), είτε

($\chi=Z_1 u_1 Z_2$ και $y=Z_1 W_i Z_2$)

Έστω ζ η σχέση που είναι το μεταβατικό, ανακλαστικό συμπλήρωμα της ζ. Η ζ είναι μία σχέση ισοδυναμίας.

Το πρόβλημα των συστημάτων Thue είναι:

"δοθέντος ενός συστήματος Thue ζ και 2 λέξεων x,y είναι xζ y;"

Το πρόβλημα αυτό είναι άλυτο (μη αποφασίσιμο).

Το πρόβλημα της αντιστοιχεία του Post

Ορισμός. Ένα σύστημα αντιστοιχιών είναι ένα πεπερασμένο σύνολο p από δοατεταγμένα ζεύγη μη κενών λέξεων.

Ένα ταίριασμα στο P είναι μια λέξη w έτσι ώστε για κάποιο n>0 και κάποια (όχι κατ'ανάγκη διακεκριμμένα ζεύγη (u_1, v_1), $(u_2, v_2), \ldots, (u_n, v_n)$ του P, έχουμε:

$w= u_1 u_2 \ldots u_n = v_1 v_2 \ldots v_n$

Για παράδειγμα: Αν p= (a,ab), (b,ca), (ca,a), (abc,c) τότε το w=abcaaabc είναι ένα ταίριασμα του P, και να η ακολουθία που το δημιουργεί.

(a,ab) (b,ca) (ca,a) (a,ab) (abc,c)

Το πρόβλημα αντιστοιχίας του Post είναι:

"Δοθέντος ενός συστήματος αντιστοιχιών P, έχει το P ένα ταίριασμα;"

Θεώρημα 7.3.1 Το πρόβλημα αντιστοιχίας του Post είναι άλυτο.

Άλυτα προβλήματα για γραμματικές χωρίς συμφραζόμενα

(α) Το πρόβλημα του να αποφασίσουμε, δοθέντων 2 Context free γραμματικών G_1 και G_2, αν η γλώσσα που παράγεται από την G_1 και η γλώσσα που παράγεται από την G_2 έχουν κάποια κοινή λέξη ή όχι (αν δηλαδή $L(G_1)$ $L(G_2)=0$) είναι άλυτο.

Ορισμός

Μια γραμματική G χωρίς συμφραζόμενα είναι αμφίβολα (ambiguous) αν υπάρχει κάποια λέξη $w \in L(G)$ που έχει 2 διαφορετικά δέντρα παραγωγής.

Π.χ. η γραμματική

$S \rightarrow N$

$S \rightarrow S+S$

$S \rightarrow S*S$

$N \rightarrow x$

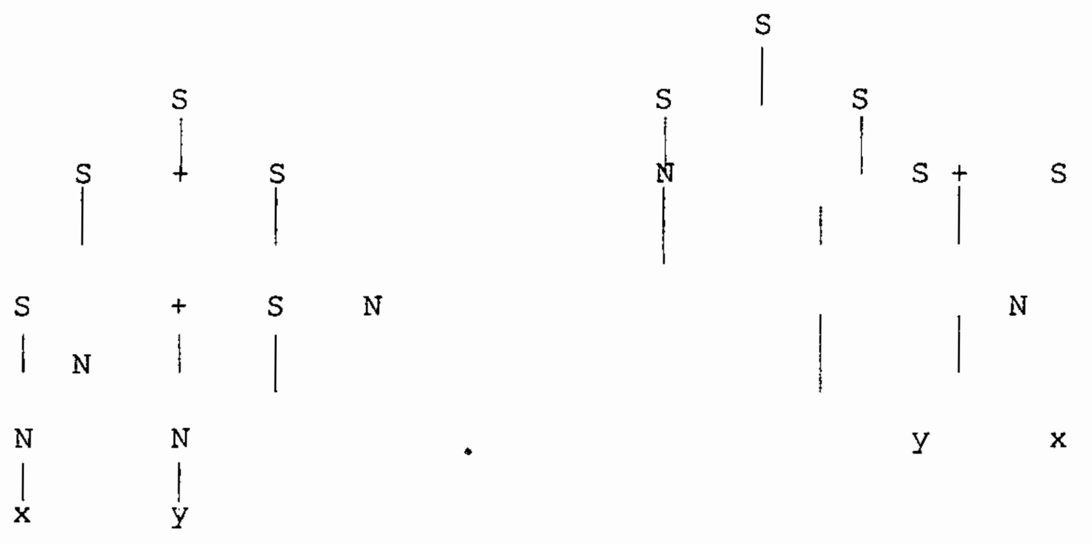
$N \rightarrow y$

είναι αμφίβολη.

Η λέξη x*y+x έχει 2 δέντρα παραγωγής και 2 δυνατές τιμές:

```
     *                              +
x            +          και      *           x
     y            x          x       y
```

Δέντρα παραγωγής της x*y+x

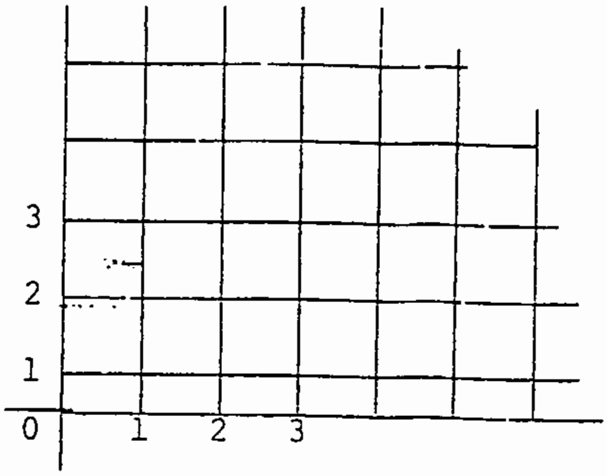


(β) Το πρόβλημα του να αποφασίσουμε, δοθείσας γραμαμτικής χωρίς συμφραζόμενα G αν η G είναι αμφίβολη (ή όχι), είναι άλυτα.

Το πρόβλημα του Ντόμινο

Για να βοηθηθεί ο αναγνώστης στην κατανόηση της μεθόδου της αναγωγής, θα αποδείξουμε τώρα ότι το περίφημο πρόβλημα του "Ντόμινο" είναι άλυτο. Το πρόβλημα αυτό έχει ως εξής:

Μας δίνεται ένα πεπεασμένο σύνολο από ψηλίδες (ντόμινο), το καθένα υπό μορφή ενός τετραγώνου πλευράς 1. Μας ζητείται να καλύψουμε όλο το πρώτο τεταρτημόριο του επιπέδου με τέτοιες αψίδες. Το τεταρτημόριο είναι χωρισμένο σε τετραγωνικές θέσεις πλευράς 1.



Μπορούμε να χρησιμοποιήσουμε όσα αντίγραφα θέλουμε, από τις ψηφίδες που μας δίνονται. Οι μόνοι περιορισμοί είναι ότι (α) μια ειδική αρχική ψηφίδα πρέπει να τοποθετηθεί στο τετράγωνο (0,0) (0,1) (1,0) (1,1) και (β) οι ψηφίδες που γειτονεύουν οριζόντια πρέπει να "ταιριάζουν" στην πλευρά που γειτονεύουν. Το ίδιο και για κατακόρυφη γειτνίαση (γ) οι ψηφίδες δεν πρέπει να περιστραφούν η να χρησιμοποιηθούν ανεστραμμένες.

Αυτό το πρόβλημα μπορεί να διατυπωθεί αυστηρά μαθηματικά ως εξής:

Ένα σύστημα ψηφιδωτών είναι μια τατράδα.

D= (D, do, H, V)

όπου D ένα πεπερασμένο σύνολο, do ∈ D

H, V, C DxD

Μια κάλυψη (ντόμινο) του D είναι μια συνάρτηση f: N N→D ώστε τα εξής να ισχύουν:

$f(0,0) = do$

$(f(n,m), f(n+1,m)) \in H \ \forall \ m,n \in N$

$(f(n.m), f(n,m+1)) \in V \ \forall \ m,n \in N$

Θεώρημα Το πρόβλημα δοθέντος ενός συστήματος ψηφιδωτών D, αν υπάρχει μια κάλυψη του D είναι άλυτο.

Απόδειξη

Θα ανάγουμε στο παραπάνω πρόβλημα, το πρόβλημα του να αποφασισθεί, δοθείσας μιας μηχανής Turing M, εάν η M τελικά σταματά όταν αρχίσει από το αριστερότερο τετράγωνο μιας άδειας ταινίας.

Η βασική ιδέα είναι να κατασκευάσουμε, από οποιαδήποτε μηχανή Turing, ένα σύστημα ψηφιδωτών D, έτσι ώστε μια κάλυψη από το D (αν μια τέτοια υπάρχει) να παριστάνει έναν απείρου διαρκείας υπολογισμό της M που να αρχίζει από άδεια ταινία.

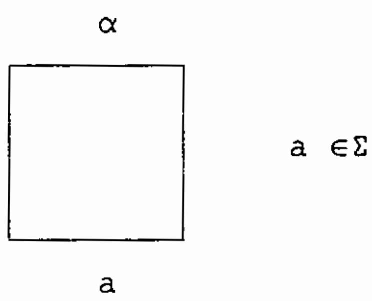
Οι σχηματισμοί της M παριστάνονται "οριζόντια" στο ψηφιδωτό: διαδοχικοί σχηματισμοί εμφανίζονται ο ένας πάνω στον άλλον. Αν η M λειτουργεί επ'άπειρον, διαδοχικές γραμμές μπορούν να "ψηφιδώνονται' επ'άπειρο; αντίθετα, αν η M σταματά μετά από κ βήματα, τότε είναι δυνατόν να καλύψουμε μόνο κ γραμμές.

Ας φανταστούμε ότι οι πλευρές των ψηφίδων είναι "μαρκαρισμένες" με πληροφορία. Θα επιτρέπουμ,ε 2 ψηφίδες να είναι άμεσοι γείτονες (οριζόντια ή κατακόρυφα) μόνο αν τα "μαρκαρίσματα" των πλευρών σε επαφή είναι ταυτόσημα.

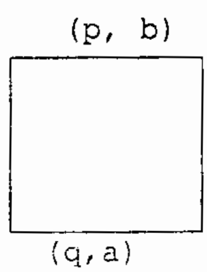
Για να γίνουμε πιο συγκεκριμένοι:

Έστω ότι μας δίνεται η M=(κ, Σ, δ, s). Τότε το D=(D, do, H, V) όπου το D περιέχει τις εξής ψηφίδες:

(α) Για κάθε α ∈ Σ, τις ψηφίδες του σχήματος Α, που απλώς μεταβιβάζουν τα μη αλλαγμένα σύμβολα (της ταινίας) προς τα πάνω, από σχηματισμό σε σχηματισμό.

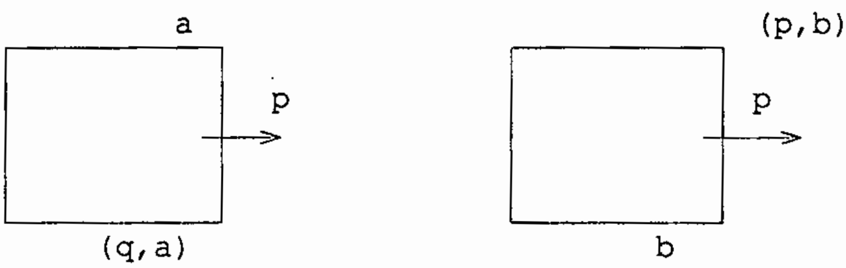


(β) για κάθε α ∈Σ καί q ∈ k έτσι ώστε δ(q, a)=(p,b) όπου p ∈k και b ∈Σ την ψηφίδα του σχήματος Β.

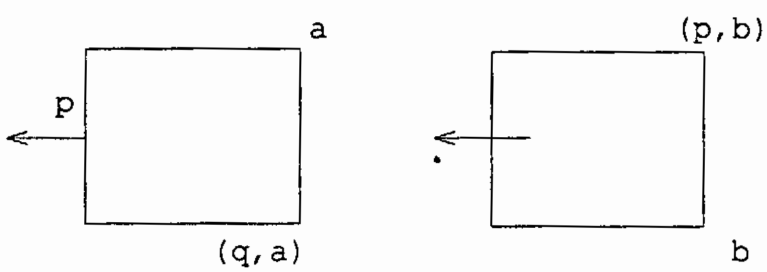


Αυτή η ψηφίδα μεταβιβάζει την θέση της κεφαλής προς τα πάνω και ακόμα αλλάζει την κατάσταση της M και το περιεχόμενο του τετραγώνου υπό εξέταση, σύμφωνα με την δ.

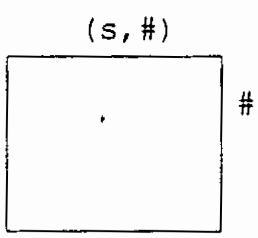
(γ) Για κάθε $q \in k$ και $a \in \Sigma$ έτσι ώστε $\delta(q,a)=(p,R)$, για κάποιο $p \in k$, αντιστοιχούμε την ψηφίδα της εικόνας Γ1 και επίσης, για κάθε $b \in \Sigma$, την ψηφίδα του σχήματος Γ2. Αυτές οι ψηφίδες μεταβιβάζουν την κίνηση της κεφαλής 1 τετράγωνο δεξιά ή αριστερά, και αλλάζουν την κατάσταση της M αντίστοιχα.



(δ) ψηφίδες παρόμοιες με το (γ) όταν έχουμε L αντί R



(ε) η αρχική ψηφίδα do



Επίσης, οι ψηφίδες της άδειας ταινίας:

#

d1: # $\boxed{}$ #

(Με την do και αντίγραφα της d1 "γεμίζουμε" την 1η γραμμή του τεταρτημόριου μπορούν να καλυφθούν.

Σημείωση

Το περίφημο 10ο πρόβλημα του Hilbert ρωτα αν υπάρχει μέθοδος που να βρίσκει ακέραιες λύσεις σε αυθαίρετα πολυώνυμα πολλών μεταβλητών. (D. Hilbert, 1990).

Θεώρημα

Το 10ο πρόβλημα του Hilbert είναι άλυτο

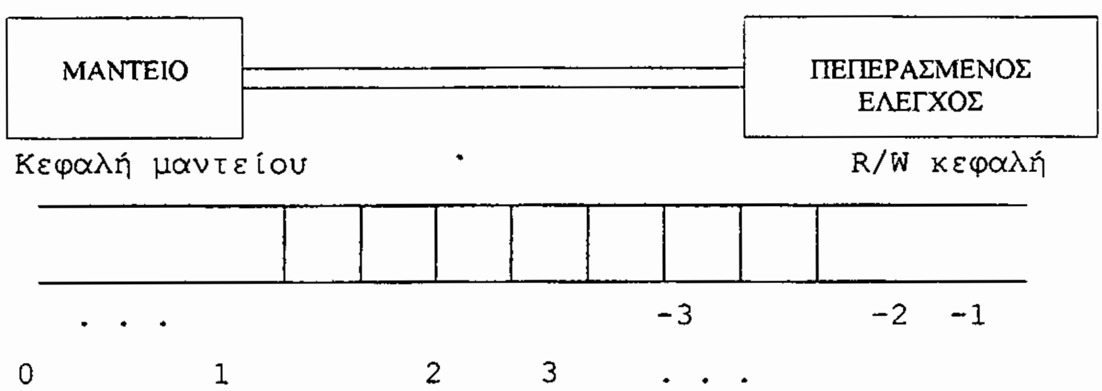
(Η απόδειξη έγινε από τον Σοβιετικό YU. Matiyasevich, Doklady Acad Nauk SSSR, 1970, 279-182).

Δες επίσης και την αγγλική περίληψη στο M. Davis "Hilbert's 10^{th} problem is unsolvable" Amer. Math. Monthly, 80 (1973) 233-269.

NP- Πλήρη προβλήματα

Μη ντετερμινιστικές μηχανές Turing (NDTM)

Θα χρησιμοποιήσουμε εδώ μια παραλλαγή των μη-ντετερμινιστικών μηχανών Turing, ισοδύναμη με τον σταντάρντ ορισμό. Η μηχανή αυτή Non -Deterministic Turing Machine - NDTM) έχει ακριβώς την ίδια δομή όπως και μια ντετερμινιστική Μηχανή Turing, με την διαφορά ότι έχει επιπλέον μια "μονάδα μαντέματος (μαντείο)" το οποίο έχει τη δικιά του κεφαλή, που μπορεί μόνο να γράψει στην ταινία (και όχι να διαβάσει - write only). Το μαντείο προμηθεύει το μέσον για την καταγραφή στην ταινία των μη - ντετερμινιστικών επιλογών, και θα χρησιμοποιηθεί αποκλειστικά για το σκοπό αυτό.



Η περιγραφή μιας τέτοιας NDTM (με ταινία άπειρη και προς τις 2 κατευθύνσεις) περιλαμβάνει το αλφάβητο ταινίας + εισόδου Σ, το "άδειο" σύμβολο #, το σύνολο καταστάσεων K, την αρχική κατάσταση s, δύο καταστάσεις παύσης q_Y και q_N και μια συνάρτηση μετάβασης.

δ: $(K - q_Y, q_N)\ \Sigma \rightarrow K x \Sigma x\ L,R$

Ο υπολογισμός της NDTM M πάνω σε μία λέξη εισοδου x ∈ Σ* εκτελείται σε δύο φάσεις:

Φάση 1: (φάση μαντέματος)

Η λέξη εισόδου γράφεται στα τετράγωνα 1 έως |χ|, η R/W κεφαλή ερευνά το τετράγωνο 1, η κεφαλή μαντέματος ειναι στο τετράγωνο -1 και το "κουτί" του πεπερασμένου ελέγχου είναι αδρανές. Το μαντείο τότε, αρχίζει να γράφει σύμβολα στην ταινία, ένα σύμβολο σε κάθε βήμα, και μετά να κινεί την κεφαλή μαντέματος 1 τετράγωνοο αριστερά. Κάποτε σταματά, γίνεται αδρανές, και ενεργοποιείται η μονάδα ελέγχου στην κατάσταση s.

Όσο δουλεύει το μαντείο, οι επιλογές του τι σύμβολο θα γραφεί και του αν θα σταματήσει το "μάντεμα" ή όχι, είναι τελείως αυθαίρετες. Έτσι, το μαντείο μπορεί να γράψει μαι οποιαδήποτε λέξη του Σ* και ίσως να μη σταματήσει ποτέ.

Φάση 2: (Φάση απολογισμού)

Η φάση αυτή αρχίζει όταν η μονάδα ελέγχου είναι στην κατάσταση s. Ο υπολογισμός προχωρεί τότε κάτω από τις οδηγίες της δ (όπως σε μια DTM). Βέβαια η "μαντεμένη" λέξη συνήθως εξετάζεται και χρησιμοποιείται κατά τη διάρεκια του υπολογισμού. Ο υπολογισμός σταματά εάν και όταν η μονάδα ελέγχου εισέλθει σε μία από τις καταστάσεις qY ή qN, και λέγεται "υπολογισμός που καταλήγει σε αποδοχή" αν σταματά στην κατάσταση qY.

Όλοι οι άλλοι υπολογισμοί (ανεξάρτητα από το εάν σταματούν ή όχι) λέγονται "υπολογισμοί που δεν καταλήγουν σε αποδοχή"

Ορισμός: Η NDTM M δέχεται την λέξη χ, αν υπάρχει τουλάχιστο ένας υπολογισμός (πάνω στη χ) που να καταλήγει σε αποδοχή.

Ορισμός

Η γλώσσα που αναγνωρίζεται από την M είναι η γλώσσα LM όπου

LM= x ∈ Σ*: Η M δέχεται το x

Ορισμός

Ο χρόνος που απαιτείται από μιαν NDTM M για να δεχθή την λέξη x είναι ο ελάχιστος (επί όλων των υπολογισμών που καταλήγουν σε αποδοχή) αριθμός βημάτων που γίνονται στις 2 φάσεις, εώς ότου η M εισέρχεται στην κατάσταση qY.

Ορισμός (συνάρτηση υπολογιστικής πολυπλοκότητας)

Η συνάρτηση υπολογιστικής πολυπλοκότητας

TM: Z+ → Z+ της (NDTM) M

είναι ∀ n ∈ N :

TM(n)= max 1 U m: ∃ x ∈L_M με x=n

έτσι ώστε ο χρόνος που απαιτείται από την M για να γίνει δεκτό το χ είναι m.

Ορισμός

Η (NDTM) Μ είναι πολυωνυμικού χρόνου αν υπάρχει ένα πολυώνυμο p( ) έτσι ώστε ∀ n 1

$T_M(n)$ p(n)

Ορισμός Η κλάση γλωσσών NP είναι

NP= L: υπάρχει μια NDTM Μ πολυωνυμικού χρόνου για την οποία

L_M=L C Σ*

Σημείωση

Εννοείται ότι μια NDTM χωρίς μαντείο είναι απλά μια DTM δηλαδή οι ντερερμ. TM είναι ειδικές περιπτώσεις των NDTM.

Προβλήματα απόφασης και κωδικοποίησης

Προβλήμτα απόφασης ειναι εκείνα που έχουν δύο πιθανές απαντήσεις (λύσεις), είτε "ναι" είτε "όχι". Γενικά, ένα πρόβλημα απόφασης Π αποτελείται, απλά από ένα σύνολο Dπ από συγκεκριμεννοποιήσεις (instances) και ένα υποσύνολο YπCDπ από συγκεκριμενοποίηση που έχουν απάντηση "ναι".

Μια μέθοδος κωδικοποίησης για ένα πρόβλημα Π είναι ένας τρόπος για να περιγράψουμε κάθε συγκεκριμενοποίηση του Π με μια "κατάλληλη" λέξη σε κάποιο αλφάβητο Σ.

Έτσι, το πρόβλημα Π και η μέθοδος κωδικοποίησης του e, χωρίζουν το Σ* σε τρία ξένα υποσύνολα: το σύνολο λέξεων που δεν είναι κωδικοποιήσεις συγκεκριμενοποιήσεων του Π, ένα σύνολο λέξεων που κωδικοποποιούν προβλήματα (συγκεκριμενοποιήσεις) στις οποίες η απάντηση είναι "όχι" και ένα σύνολο λέξεων που είναι συγκεκριμενοποιήσεις στις οποίες η απάντηση είναι "ναι".

Αυτό το τρίτο σύνολο είναι η γλώσσα L P, e

L Π, e = x ∊ Σ*: χ= κωδικοποίηση (με την e) μιας συγκεκριμενοποίησης I ∊Yπ.

Θα αναφέρουμε εδώ την "στάνταρντ" (κανονική) κωδικοποίηση, e0, που αντιστοιχεί σε κάθε συγκεκριμενοποίηση I, μια λέξη χ στο αλφάβητο Ψ=0,1 -, , ,(,) , , (που λέγεται δομημένη λέξη).

Ορίζουμε τις δομημένες λέξεις, αναδρομικά, ως ακολούθως:

(1) Η δυαδική αναπαράσταση ενός ακέραιου κ είναι μια δομημένη λέξη, eο(κ)

(2) Αν το χ είναι δομημένη λέξη που παριστάνει τον ακέραιο κ, τότε χ είναι μια δομημένη λέξη που χρησιμοποιείται σαν όνομα 9π.χ. για να ονομασθούν οι κορυφές ενός γράφου).

(3) Αν χ1,χ2,…,χm είναι δομημένες λέξεις που παριστάνουν τα αντιμείμενα χ1,…,χm, τότε (χ1,…,χm) είναι μια δομημένη λέξη που παριστάνει την ακολουθία <χ1,…,χm>.

Η σχέση μεταξύ P και NP

Ορισμός

P= L: υπάρχει μια ντετερμινιστική (χωρίς μαντείο) TM M για την οποία η L είναι η γλώσσα που αναγνωρίζεται από την M.

Έχουμε ήδη πει ότι κάθε ντετερμιν. TM είναι και NDTM (με "άδειο" μαντείο) άρα $P \subset NP$.

Θεώρημα Αν $\Pi \in NP$ τότε υπάρχει ένα πολυώνυμο $p(\ )$ έτσι ώστε η Π να μπορεί να λυθεί από μια ντετερμινιστική TM που έχει πολυπλοκότητα χρόνου της τάξεως του $2^{p(n)}$.

Απόδειξη Έστω A μια NDTM που λύνει το Π και έστω $q(n)$ η (πολυωνυμική) υπολογιστική πολυπλοκότητα της A.

(Χωρίς βλάβη της γενικότητας υποθέτουμε ότι η $q(\ )$ μπορεί να υπολογιστεί σε πολυωνυμικό χρόνο).

Τότε, για κάθε είσοδο μήκους n, που γίνεται δεκτή, υπάρχει ένα μάντεμα (μια μαντεμένη λέξη) μήκους το πολύ $q(n)$ η οποία βοηθά την φάση 2 να καταλήξει σε "ναι", σε (το πολύ) $q(n)$ βήματα.

Έτσι, ο αριθμός των δυνατών μαντεμένων λέξεων είναι το πολύ $|\Sigma|^{q(n)}$ (γιατί;).

Μπορούμε να κατασκευάσουμε μια ντετερμινιστική TM που, για δοθείσα είσοδο μήκους n, μπορεί να κάνει ότι και η A, εφαρμόζοντας την φάση 2 της A έως να σταματήσει ή να κάνει $q(n)$ βήματα, σε κάθε ένα από τα $|\Sigma|^{q(n)}$ "μαντέματα". Η προσομοίωση θα απαντήσει "ναι" να βρει ένα "μάντεμα" που οδηγεί σε αποδοχή της λέξης εισόδου σε q (n) βήματα. Αλλιώς θα απαντήση "όχι". Η υπολογιστική πολυπλοκότητα χρόνου της νέας μηχανής είναι βασικά q(n) $|\Sigma|^{q(n)}$, δηλαδή $O(^{2p(n)})$' για ένα καταλλήλως εκλεγμένο πολυώνυμο p(m).

Μολονότι η παραπάνω προσομοίωση θα μπορούσε να επιταχυνθή λίγο με κόλπα, δεν υπάρχει μέχρι σήμερα γνωστός τρόπος για να γίνει σε λιγότερο από εκθετικό χρόνο.

Έτσι, οι Computer Science Theorists πιστεύουν ότι P≠NP.

NP

P

Πολυωνυμικοί μετασχηματισμοί και NP -πληρότητα

Ορισμός Ένας πολυωνυμικός μετασχηματισμός από μια γλώσσα $L_1 C \Sigma_2^*$

$f: \Sigma_1^* \rightarrow \Sigma_2^*$

που ικανοποιεί τις ακόλουθες δύο συνθήκες:

1. Υπάρχει μια ντετερμινιστική TM πολυωνυμικού χρόνου, που υπολογίζει την f
2. $\forall x \in \Sigma_1^*$, $x \in L_1$ $f(x) \in L_2$

Γράφουμε "L1 L2" και διαβάζουμε "η L1 μετασχηματίζεται στην L2" (σε πολυωνυμικό χρόνο).

Θεώρημα 4.1.

Αν L1 L2 και L2 ∈ P. Έστω Σ1, Σ2 τα αλφάβητα των L1, L2 και f η συνάρτηση μετασηματισμού. Έστω Mf και M_2 οι ντετερμιν. Τμ που υπολογίζουν την f και δέχονται την L_2 σε πολυωνυμικό χρόνο. Μπορούμε να κατασκευάσουμε μια ντετερμιν. ΤΜ πολυωνυμικού χρόνου που να δέχεται την L_1 συνθέτοντας την M_f με την M_2:

$\forall x \in \Sigma 1^*$ (1) χρησιμοποίησε την Mf για τον υπολογισμό της $f(x) \in \Sigma_2^*$. (2) Τρέξτε το M2 με είσοδο $f(x)$.

Θεώρημα 4.2. Η "α" είναι μεταβατική: δηλαδή αν L1 L2 και L2 L3 τοτε L1 L3.

(Απόδειξη παραλείπεται)

Ορισμός (NP πληρότητα)

Μία γλώσσα L ονομάζεται NP - πληρης αν

(1) η L ∈NP

(2) και ∀ L ∈ NP, η L′ L

Το θεώρημα 4.1. μας λέει ότι τα NP-πλήρη προβλήματα είναι τα "δυσκολότερα" προβλήματα στο NP. Αν ένα και μόνο από αυτά μπορούσε να λυθεί σε πολυωνυμικό χρόνο, τότε όλα τα προβλήματα του NP θα μπορούσαν να λυθούν σε πολυωνυμικό χρόνο. Αντίθετα, αν ένα μόνο από αυτά δεν λύνεται γρήγορα, τότε κανένα NP-πλήρες πρόβλημα δεν λύνεται γρήγορα.

Υποψία NP

NP - πλήρη

P

Θεώρημα 4.3. Αν L1, L2 ∈ NP και L1 L2 και L1 είναι NP-πλήρης τότε και η L2 είναι NP-πλήρης (απόδειξη: αφήνεται στον αναγνώστη).

Το θεώρημα του COOK

Το θεώρημα 4.3. μας δίνει μία μέθοδο για να αποδείξουμε ότι κάποιο πρόβλημα Τ είναι NP-πλήρες:

(1) Δείχνουμε ότι η L Π, e ∈ NP και

(2) Δείχνουμε ότι κάποιο γνωστό πρόβλημα (που είναι NP-πλήρες) " " στο π.

Για να χρησιμοποιήσουμε την παραπάνω μέθοδο, χρειαζόμαστε ένα NP - πλήρες πρόβλημα.

O Cook (1971) μας έδωσε το πρώτο ευρεθέν NP-πλήρες πρόβλημα. Πρόκειται για ένα πρόβλημα απόφασης από την (δίτιμη) προτασιακή λογική, που είναι γνωστό σαν "πρόβλημα της Ικανομοιησιμότητας" (Satisfiability), πιο σύντομα "SAT".

Έστω u=u1, u2,…,um ένα σύνολο από Boolean μεταβλητές.

Αν u ∈u, τότε τα u, u λέγονται "γράμματα" (litterals) του.

Μια αντιστοιχία τιμών αληθείας στο u είναι μια συνάρτηση t: u → true, false. Αν t(u)="true" τότε λέμε ότι η u είναι αληθής ως προς την t. (Εννοείται, τότε η t(u)= "false").

Μία πρόταση, c, επί του u είναι για συλλογή γραμμάτων του u π.χ. c=u1. U38, u15.

Μια πρόταση παριστάνει το λογικό Η (OR) των γραμμάτων, δηλ. Η c=u1Vu38Vu15.

Η c είναι αληθής αν και μόνο αν τουλάχιστον ένα από τα γραμματά της είναι αληθές.

Ορισμός

Μια συλλογή προτάσεων c είναι ικανοποιήσιμη αν και μόνο αν υπάρχει μια αντιστοιχεία τιμών αληθείας στο u που ταυτόχρονα κάνει αληθείς (ικανοποιεί) ΟΛΕΣ τις προτάσεις του c.

Πρόβλημα Ικανμοποίησιμότητας (SAT)

Συγκεκριμενοποίηση. Ένα σύνολο από Boolean μεταβλητές και μια συλλογή c προτάσεων επί του u.

Ερώτηση είναι η c ικανοποιήσιμη;

Θεώρημα (Cook 1971)

Το πρόβλημα της ικανοποιησιμότητας είναι NP-πλήρες.

ΠΑΡΑΔΕΙΓΜΑ

Έστω Σ_0 = {α, β} και έστω L = {ω ∈ Σ_0* : το ω περιέχει τουλάχιστον ένα α}.

Τότε η L αναγνωρίζεται από τη μηχανή Turing M = (K, Σ, δ, s) όπου K = {α_0}, Σ = {α, b, #}, s= q_0 και η δ δίνεται από τον παρακάτω πίνακα:

q	σ	δ (q, σ)
q_0	α	(h, α)
q_0	b	(q_0, L)
q_0	#	(q_0, L)

Η μηχανή αυτή, όταν ξεκινά με την συνολικά κατάσταση $(q_o, \#\omega\underline{\#})$ για κάποιο $\omega \in \Sigma^*_0$, απλώς σαρώνει αριστερά ώσπου να συναντήσει ένα α και τότε τερματίζει. Αν δεν βρέθει κανένα α, η μηχανή κρεμάει, οπότε δεν τερματίζει. Έτσι η L είναι ακριβώς το σύνολο των συμβολοσειρών ω του Σ^*_0, με τις οποίες η M τερματίζει, και συνεπώς η M αναγνωρίζει την L.

Συνδυασμοί μηχανών Turing

Κανένα από τα παραδείγματα των δύο τελευταίων ενοτήτων δεν είναι ιδιαίτερα εντυπωσιακό ως παράδειγμα υπολογιστικής διαδικασίας. Παρόλα αυτά, στην αρχή του κεφαλαίου ισχυριστήκαμε ότι οι μηχανές Turing θα αποδεικνύονταν υπολογιστές ύψιστης γενικότητας, ικανοί να εκτελούν υπερβολικά σύνθετους υπολογισμούς, παρά τον αυστηρά περιορισμένο τρόπο λειτουργίας τους. Προφανώς, έχουμε κάποια απόσταση να καλύψουμε για να αποδείξουμε τον ισχυρισμό αυτόν.

Στοχός μας στην ενότητας αυτή είναι να παρουσιάσουμε μια μέθοδο συνδυασμού απλούστερων μηχανών Turing για την κατασκευή πιο σύνθετων, έτσι ώστε οι μηχανές Turing να μπορούν να χρησιμοποιηθούν ως «τμήματα (modules) ή ως «υπορουτίνες» (subroutines) άλλων μηχανών Turing. Η κατασκευή πολύπλοκων μηχανών θα διευκολυνθεί εφόσον μπορούμε να δανειστούμε ολόκληρες μεγάλες υπομηχανές από προηγούμενες κατασκευές. Ταυτόχρονα, θα αναπτύξουμε έναν γραφικό συμβολισμό γι'αυτές τις μεγαλύτερες μηχανές, ανάλογα με τα διαγράμματα ροής για προγράμματα υπολογιστών. Ο συμβολισμός αυτός θα χρησιμοποιείται στο εξής αποκλειστικά: δεν θα χρειάζεται πλέον να προσδιορίζουμε μηχανές Turing παρουσιάζοντας με λεπτομέρειες τις συναρτήσεις μετάβασης.

Κατ'αρχήν θα χρειαστούμε κάποια τυπική επιβεβαίωση της ιδέας ότι δύο υπολογισμοί από την ίδια μηχανή Turing μπορούν, υπό ορισμένες προϋποθέσεις, να συνδυαστούν.

για κάποια συμβολοσειρά ω, και

$$(q_2\omega_2\underline{\alpha}_2u_2)\vdash^*M(q_3\omega_3\underline{\alpha}_3u_3),$$

τότε

$$(q_1,\omega_1\underline{\alpha}_1u_1)\vdash^*M(q_3\omega\omega_3\alpha_3u_3).$$

Απόδειξη. Δείτε το Πρόβλημα και τη συζήτηση που ακολουθεί.

Το λήμμα αυτό, καθορίζει ουσιαστικά ότι αν μια μηχανή Turing μπορεί να εκτελέσει κάποιον υπολογισμό χωρίς ποτέ να κουνήσει την κεφαλή της αριστερότερα από κάποιο συγκεκριμένο τετράγωνο, τότε θα εκτελέσει τον ίδιο υπολογισμό, ανεξάρτητα από το αν αυτό το τετράγωνο είναι πράγματι το αριστερότερο τετράγωνο της ταινίας ή είναι κάποιο τετράγωνο πιο πέρα στα δεξιά της ταινίας. Στην τελευταία περίπτωση, τα περιεχόμενα του αριστερού τμήματος της ταινίας παραμένουν απαράλλακτα.

Το κλειδί της απόδειξης είναι ότι, εφόσον η M δεν κρεμάει κατά τη διάρκεια του υπολογισμού $(q_2, \omega_2 \underline{\alpha_2} u_2) \vdash^{*}_{M} (q_3, \omega_3 \underline{\alpha_3} u_3)$ και εφόσον η M δεν έχει κάποιον άμεσο τρόπο για να ανιχνεύσει το αριστερό άκρο της ταινίας δεν θα πρέπει να κάνει καμία προσπάθεια κατά τη διάρκεια του υπολογισμού αυτού να μετακινήσει την κεφαλή της στα αριστερά του πρώτου συμβόλου του $\omega_2 \alpha_2 u_2$. Τότε όμως αν ξεκινήσει με την συνολική κατάσταση $(q_2, \omega\omega_2 \underline{\alpha_2} u_2)$, θα φτάσει στην $(q_3, \omega\omega_3 \underline{\alpha_3} u_3)$ χωρίς ποτέ να περάσει στα πρώτα (ω) τετράγωνα της ταινίας.

Το Λήμμα θα μας χρησιμεύσει στο να συνδυάσουμε μηχανές Turing ως υπορουτίνες. Ας υποθέσουμε ότι μια μηχανή Turing M_1 δεν κρεμάει όταν η είσοδός της παρουσιάζεται σύμφωνα με τις συμβάσεις που έχουμε περιγράψει. Τότε η M_1 μπορεί να γίνει τμήμα μιας μεγαλύτερης μηχανής Turing M_2 ως εξής: η M_2 προετοιμάζει κάποια συμβολοσειρά ως είσοδο της M_1 τοποθετώντας την κοντά στο δεξιό άκρο του μη κενού τμήματος της ταινίας, περνά τον έλεγχο στην M_1 με την κεφαλή εγγραφής /ανάγνωσης ακριβώς πέρα από το τέλος της συμβολοσειράς αυτής και τελικά ανακτά τον έλεγχο από την M_1 όταν η M_1 έχει τελειώσει τον υπολογισμό. Τα αποτελέσματα οποιουδήποτε μερικού υπολογισμού που εκτελέσθηκε από τη M_2 προτού περάσει τον έλεγχο στη M_1 θα διατηρηθούν κατά τη διάρκεια της λειτουργίας της M_1, υπό την προϋπόθεση ότι εκείνα τα μερικά αποτελέσματα διατηρήθηκαν στην ταινία στα αριστερά του τετραγώνου της ταινίας όπου άρχισε η είσοδος για τη M_1. Είναι σίγουρο ότι η M_1 δεν θα εισβάλει στα μερικά αποτελέσματα που φυλάχθηκαν από τη M_2, αφού είναι γνωστό ότι η M_1 ποτέ δεν επιχειρεί να κινήσει την κεφαλή της στα αριστερά του κενού που σημαδεύει το αριστερό άκρο της εισόδου της. Συνεπώς, θα σχεδιάζουμε πάντα τις μηχανές Turing με τέτοιον τρόπο, ώστε, όταν χρησιμοποιούνται σωστά, να μην κρεμούν αν επιθυμούμε αργότερα να τις κάνουμε τμήματα μεγαλύτερων μηχανών.

Πριν προχωρήσουμε, χρειάζεται μια ακόμη παρατήρηση. Όπως στην απόδειξη του Θεωρήματος 2.4.1, θα χρειαστεί μερικές φορές να υποθέσουμε ότι μηχανές οι οποίες συνδυάζονται δεν έχουν καμιά κοινή κατάσταση. Όπως ισχύει για όλα τα αυτόματα που έχουμε μελετήσει, η λειτουογία μιας μηχανής turring δεν εξαρτάται καθόλου από τη συγκεκριμένη επιλογή των αντικειμένων που χρησιμοποιήθηκαν ως καταστάσεις, αλλά μόνο από τον τρόπο με τον οποίο σχετίζονται αυτά τα αντικείμενα το ένα με το άλλο από τη συνάρτηση μετάβασης ή από τον προσδιορισμό της αρχικής κατάστασης. Συνεπώς είναι δυνατόν σε οποιοδήποτε σημείο να παράγουμε ένα αντίγραφο μιας μηχανής Turing απαράλλακτο με την M ή με οποιαδήποτε άλλη μηχανή Turing που θεωρούμε την ίδια στιγμή.

Είμαστε τώρα έτοιμοι να περιγράψουμε την κατασκευή συγκεκριμένων απλών μηχανών Turing και τον τρόπο με τον οποίο μπορούν να συνδυαστούν για να σχηματίσουν πιο ισχυρές μηχανές. Θεωρούμε κάποιο σταθερό αλφάβητο Σ.

Οι βασικές μηχανές. Αυτές είναι δύο ειδών: εγγραφής συμβόλου (symbol - writing) και μετακίνησης κεφαλής (head - moving).

1. Υπάρχουν (Σ) μηχανές εγγραφής συμβόλου, μια για κάθε σύμβολο στο Σ. Η κάθε μία απλώς γράφει ένα καθορισμένο σύμβολο στο τετράγωνο της ταινίας που σαρώνεται κάθε φορά και τερματίζει. Τα προηγούμενα περιεχόμενα του τετραγώνου της ταινίας αγνοούνται και η κεφαλή δεν μετακινείται. Τυπικά, η μηχανή turing που γράφει το σύμβολο α είναι η $W\alpha=(K,\Sigma,\delta,\varsigma)$ όπου $K=\{q\}$ για κάποια τυχαία επιλεγμένη θέση $q, s=q$ και $\delta(q,b)=(h,\alpha)$ για κάθε $b \in \Sigma$. Επειδή οι μηχανές

αυτές χρησιμοποιούνται τόσο συχνά, συντομεύουμε τα ονόματά τους και γράφουμε α αντί για Wα.

2. Οι μηχανές μετακίνησης κεφαλής μετακινούν απλώς την κεφαλή ή ένα τετράγωνο στα δεξιά ή στα αριστερά και μετά τερματίζουν. Έστω πάλι q κάποια σταθερή κατάσταση. Έστω $V_L=(\{q\}, \Sigma, \delta_L, q)$, όπου $\delta_L(q,\alpha)=(h,L)$ για κάθε α ε Σ. Όπως παραπάνω, συντομεύουμε τα ονόματα αυτών των μηχανών γράφοντας L και R αντί για V_L και V_R.

Οι κανόνες για το συνδυασμό μηχανών. Οι μηχανές Turing θα συνδυάζονται με τρόπο που θυμίζει τη δομή ενός πεπερασμένου αυτομάτου. Οι επιμέρους μηχανές είναι σαν τις καταστάσεις ενός πεπερασμένου αυτομάτου και οι μηχανές μπορούν να συνδεθούν μεταξύ τους με τον τρόπο που συνδέονται μεταξύ τους οι καταστάσεις ενός πεπερασμένου αυτομάτου. Ωστόσο, η μετάβαση από την μια μηχανή στην άλλη δεν γίνεται μέχρι να τερματίσει η πρώτη μηχανή. Η άλλη μηχανή ξεκινάει τότε από την αρχική της κατάσταση με την ταινία και τη θέση της κεφαλής όπως είχαν αφεθεί από την πρώτη μηχανή. Έτσι, αν M_1 και M_2 είναι μηχανές Turing, η μηχανή στο Σχήμα 4-3 λειτουργεί ως εξής: Ξεκινάει στην αρχική κατάσταση της M1 λειτουργεί όπως θα λειτουργούσε η M_1 έως ότου αυτή θα τερμάτιζε κατόπιν ξεκινάει από την αρχική κατάσταση της M_2 και λειτουργεί όπως θα λειτουργούσε η M_2.

$$\rangle M_1 \to M_2$$

Η σύνδεση μεταξύ δύο μηχανών μπορεί να εξαρτάται από το σύμβολο που σαρώνεται τη στιγμή που η πρώτη μηχανή θα τερμάτιζε (όπως ακριβώς η σύνδεση μεταξύ καταστάσεων ενός πεπερασμένου αυτομάτου μπορεί να εξαρτάται από το επόμενο σύμβολο εισόδου που πρόκειται να διαβαστεί). Συνεπώς εάν M_1, M_2 και M_3 είναι μηχανές Turing, τότε η μηχανή του Σχήματος ξεκινά μιμούμενη την M_1. Όταν η M_1 θα τερμάτιζε, θα αρχίσει η μίμηση της λειτουργίας είτε της M_2 είτε της M_3, ανάλογα με το αν το τετράγωνο που θα σαρώνεται εκείνη τη στιγμή περιέχει α ή b.

$$
\begin{array}{l}
\rangle M_1 \xrightarrow{a} M_2 \\
\;\downarrow b \\
\;M_3
\end{array}
$$

Πριν δώσουμε οποιαδήποτε πιο σύνθετα παραδείγματα ας γίνουμε πιο ακριβείς.

Ορισμός

Ένα σχήμα μηχανής (machine schema) είναι μια τριάδα (M, η, Mo), όπου M είναι ένα πεπερασμένο σύνολο από μηχανές turing με κοινό αλφάβητο Σ και σύνολα καταστάσεων ξένα μεταξύ τους.

Mo E M είναι η αρχική μηχανή (initial machine) και η είναι μια συνάρτηση από ένα υποσύνολο του MxM στο M.

Ενα σχήμα μηχανής M,η,Mo) παριστάνει μια μόνο μηχανή Turing, κατασκευασμένη από όλες τις επιμέρους μηχανές MεM. Αυτή η σύνθετη μηχανή - ας την πούμε M

- ξεκινά τη λειτουργία της μιμούμενη τη λειτουργία της αρχικής μηχανής Mo. Ωστόσο, όταν και εάν η Mo τερματίσει, η M μπορεί να αρχίσει τη λειτουργία μιας άλλης μηχανής του M. Συγκεκριμένα, αν η Mo τερματίσει ενώ σαρώνει το σύμβολο α, τότε:

- αν η (Mo,α) δεν έχει οριστεί, τότε η M τερματίζει αλλά
- αν η (Mo,α)=MεM, τότε η M συνεχίζει να λειτουργεί από την αρχική κατάσταση της M.

Ο ίδιος τρόπος λειτουργίας ακολουθείται ενόσω η M μιμείται την M όταν η M θα τερμάτιζε σε κάποιο σύμβολο b, η M θα τερματίσει αν το n (M,b) δεν είναι ορισμένο. Αλλιώς θα συνεχίσει από την αρχική κατάσταση του M΄=η(M,b). Οπωσδήποτε είναι αρκετά πιθανό ότι η M′ είναι ίδια με την M ή την Mo, και έτσι είναι πιθανό για την M να έχει άπειρους υπολογισμούς ακόμα και αν οι συνιστώσες μηχανές δεν έχουν.

Ο παρακάτω ορισμός παρέχει τις τυπικές λεπτομέρειες που αντιστοιχούν σε αυτήν την άτυπη καταγραφή.

Ορισμός

Έστω M={Mo,....,Mm} ($m \geq 0$), όπου για i=0,....,m, Mi=(Ki,Σ,δi,Si). Έστω q_0,.....,q_m νέες καταστάσεις που δεν περιέχονται σε κανένα από τα Ki. Τότε αν (M,η,Mo) είναι ένα σχήμα μηχανής, θα λέμε ότι παριστάνει τη μηχανή Turing M, όπου M=(K,Σ,δ,ς), K=KoU......UKmU{q_o,....,q_m}

s=s_o

και η δ ορίζεται ως εξής:

(α) Αν $0 \leq i \leq m, q \in Ki, \alpha \in \Sigma$, $και δi(p,b)$ όπου p δεν είναι h, τότε δ(q,α)=δi(q,α)=(p,b).

(β) Αν $0 \leq i \leq m, q \in Ki, \alpha \in \Sigma$, και δi (q,α) = (h,b), τότε δ(q,α)=(qi,b).

(γ) Αν $0 \leq i \leq m, \alpha \in \Sigma$ και η η(Mi, α) δεν είναι ορισμένη, τότε δ(q_iα)=(h,α).

(δ) Αν $0 \leq i \leq m, \alpha \in \Sigma$ και η (Mi, α)=Mj, τότε έστω δj(Sj, α)=(p,b). Τότε

$$\delta(q_i,\alpha)=\begin{cases}(p,b)\ \alpha\nu\ p \neq h \\ (q_j,b)\ \alpha\nu p = h\end{cases}$$

Οι καταστάσεις q_o,........,q_m εισάγονται ως προσωρινά σημεία παύσης ανάμεσα στις ποικίλες μηχανές. Η q_i παίρνει τη θέση της κατάστασης τερματισμού της Mi. Στον ορισμό του δ, το μέρος (α) δηλώνει ότι κάθε μηχανή λειτουργεί όπως ανεμενόταν, ώσπου να είναι έτοιμη να τερματίσει. Το μέρος (β) λέει ότι η Mi πηγαίνει στη κατάσταση qi αντί να τερματίσει και τα μέρη (γ) και (δ) λένε ότι από την κατάσταση qi, η σύνθετη μηχανή είτε τερματίζει είτε εκτελεί το πρώτο βήμα της κατάλληλης μηχανής Mj, ανάλογα με τα περιεχόμενα του τετραγώνου που σαρώνεται. Σημειώστε ότι τα σύνολα καταστάσεων Ki είναι ξένα μεταξύ τους λόγω του ορισμού του προηγούμενου και συνεπώς η δ είναι μια καλώς ορισμένη συνάρτηση.

ΠΑΡΑΔΕΙΓΜΑ

Έστω ότι το M αποτελείται μόνο από τη μηχανή R που ορίστηκε παραπάνω: M= {Mo}, όπου Mo=R=(q, Σ, δ_R,q). Θεωρείστε το σχήμα μηχανής (M, η, Mo), όπου

$$n(Mo,\alpha)=\begin{cases} Mo & \alpha\nu\ \alpha \neq \# \\ \mu\eta\ \ o\rho\iota\sigma\mu\varepsilon\nu\eta & \alpha\nu\alpha = \# \end{cases}$$

Τότε (M,η, Mo) παριστάνει τη μηχανή που συμβολίζουμε με R#, όπου ·

$$R\#=(\{q,q_o\},\ \Sigma,\ \delta,\ q),$$

και δ(q,α)=(q_o, R) για κάθε α ε Σ

$$\delta(q_o,a)=\begin{cases} (q_o,R) & \alpha\nu\ \alpha \neq \# \\ (h,\alpha) & \alpha\nu\ \alpha = \# \end{cases}$$

Η μηχανή αυτή, όταν ξεκινά από οποιοδήποτε τετράγωνο της ταινίας, μετακινεί την κεφαλή της ένα τετράγωνο προς τα δεξιά, συνεχίζει να κινεί την κεφαλή της προς τα δεξιά ώσπου να βρει ένα κενό τετράγωνο, και τότε τερματίζει.

Αν Σ είναι το αλφάβητο (#, α, b, c), τότε η μηχανή R # του Παραδείγματος θα εικονιζόταν διαγραμματικά όπως φαίνεται στο σχήμα.

Οι βασικοί κανόνες για την αναπαράσταση ενός σχήματος μηχανής (M,η,Mo) είναι οι εξής:

1. Κάθε μηχανή Turing M ε M απεικονίζεται μία και μόνο φορά.

2. Η αρχική μηχανή Mo σημαδεύεται με ένα >.

3. Οποτεδήποτε α ε Σ και η(M,α) ορίζεται, για παράδειγμα, αν η(M,α)=M', σχεδιάζεται ένα βέλος επιγραφόμενο α από το M στο M'. (Είναι δυνατό, όπως στο Σχήμα παραπάνω, ότι M=M'.)

Ακόμα κι αυτοί οι τρεις κανόνες είναι πολύπλοκοι όταν εφαρμόζονται και έτσι κάνουμε κι άλλες συντομεύσεις. Κατ'αρχήν μπορεί να υπάρχουν δύο ή περισσότερες εμφανίσεις της ίδιας μηχανής Turing (ή μάλλον, του συμβόλου που παριστάνει αυτή τη μηχανή Turing) στο ίδιο διάγραμμα. Αυτές δεν παριστάνουν το ίδιο μέλος του M, αλλά αντίγραφα της ίδιας μηχανής με σύνολα καταστάσεων που είναι ανά δύο ξένα και που είναι ξένα από τα σύνολα καταστάσεων οποιασδήποτε άλλης απεικονιζόμενης μηχανής. Όλες αυτές οι μηχανές είναι μέλη του M.

ΠΑΡΑΔΕΙΓΜΑ

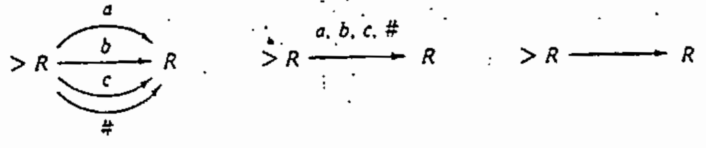
Το Σχήμα εικονίζει ένα σχήμα μηχανής (M, η, Mo), στο οποίο το M αποτελείται από δύο αντίγραφα της R. Η μηχανή που αναπαρίσταται από το διάγραμμα αυτό μετακινεί την κεφαλή της προς τα δεξιά κατά ένα τετράγωνο. Κατόπιν, αν το τετράγωνο αυτό περιέχει ένα α, μετακινεί την κεφαλή της ένα ακόμα τετράγωνο προς τα δεξιά.

$$\rangle R \xrightarrow{a} R$$

Πολλαπλά βέλη και παρόμοια υποδιαγράμματα μπορούν να περιοριστούν με ποικίλους τρόπους. Ένα βέλος επιγραφόμενο με πολλά σύμβολα παίρνει τη θέση πολλών βελών, εκ των οποίων το καθένα επιγράφεται με ένα σύμβολο. Ένα βέλος χωρίς επιγραφή αντικαθιστά μια δέσμη από βέλη, που επιγράφονται με όλα τα σύμβολα του Σ.

ΠΑΡΑΔΕΙΓΜΑ

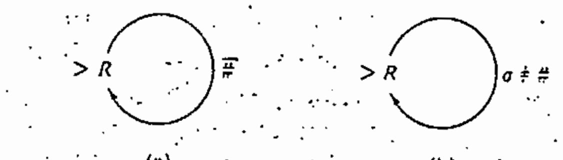
Αν Σ={a,b,c, #}, το διάγραμμα στο σχήμα (α) μπορεί να συντομευτεί όπως στο Σχήμα (β) ή ©. Η μηχανή μετακινεί την κεφαλή της προς τα δεξιά κατά δύο τετράγωνα και τερματίζει.



(a) (b) (c)

Μερικές φορές το βέλος που συνδέει δύο μηχανές μπορεί να παραλειφθεί πλήρως και οι αναπαραστάσεις των δύο μηχανών να παρατεθούν δίχως βέλος μεταξύ τους. Σύμφωνα με τη σύμβαση αυτή, η μηχανή του Παραδείγματος γίνεται απλώς RR. Αυτή η μηχανή μπορεί επίσης να γραφεί ως R^2 γενικά, το M* παριστάνει την ίδια μηχανή με το MM....M, με k εμφανίσεις του M.

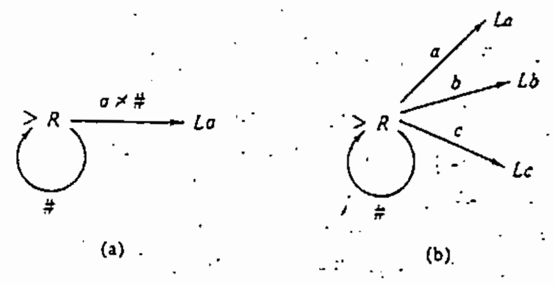
Αν σ είναι οποιοδήποτε σύμβολο, μπορούμε γενικά μερικές φορές να περιορίσουμε πολλαπλά βέλη χρησιμοποιώντας το σ με τη σημασία «οποιοδήποτε σύμβολο εκτός από το σ». Με τον συμβολισμό αυτόν, η μηχανή R# του Παραδείγματος θα γινόταν όπως φαίνεται στο σχήμα. Παρακάτω (α). Μια άλλη συντομευμένη μορφή της ίδιας μηχανής φαίνεται στο Σχήμα παρακάτω (b).



Εδώ το $\sigma \neq \#$ διαβάζετάι «οποιοδήποτε σύμβολο σ εκτός από το #». Το πλεονέκτημα του συμβολισμού αυτού είναι ότι το σ μπορεί κατόπιν να χρησιμοποιηθεί οπουδήποτε αλλού στο διάγραμμα ως όνομα μιας μηχανής.

ΠΑΡΑΔΕΙΓΜΑ

Το Σχήμα (α) εικονίζει μια μηχανή που σαρώνει προς τα δεξιά ώσπου να βρει ένα μη κενό τετράγωνο, κατόπιν αντιγράφει το σύμβολο εκείνου του τετραγώνου στο τετράγωνο που βρίσκεται αμέσως στα αριστερά από εκεί που βρέθηκε. Μια πιο πλήρης αναφορά για αυτή τη μηχανή, στην περίπτωση όπου $\Sigma=\{\alpha,b,c,\#\}$, φαίνεται στο Σχήμα (b).

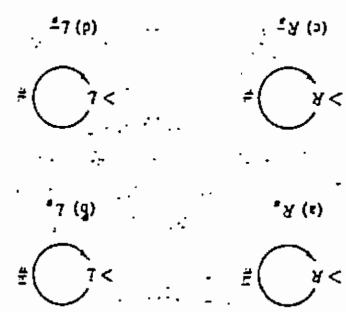


Μερικά παραδείγματα πιο ισχυρών μηχανών Turing

Ας δούμε τώρα πως οι Τεχνικές του τελευταίου κεφαλαίου μας δίνουν τη δυνατότητα να κατασκευάζουμε μηχανές Turing οι οποίες επιτελούν όλο και πιο πολύπλοκες λειτουργίες.

ΠΑΡΑΔΕΙΓΜΑ

Μηχανές που βρίσκουν κενά ή μη κενά τετράγωνα φαίνονται στο σχήμα παρακάτω. Είναι οι εξής:



(α) Η R #, η οποία βρίσκει το πρώτο κενό τετράγωνο στα δεξιά του τρέχοντος τετραγώνου που σαρώνεται.

(β) η L#, η οποία βρίσκει το πρώτο κενό τετράγωνο στα αριστερά του τρέχοντος τετραγώνου που σαρώνεται.

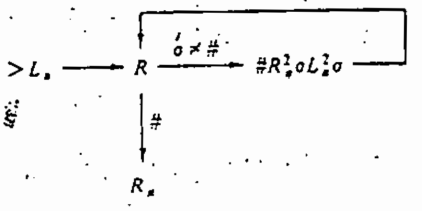
(γ) η R#, η οποία βρίσκει το πρώτο μη κενό τετράγωνο στα δεξιά του τρέχοντος τετραγώνου που σαρώνεται.

(δ) η L #, η οποία βρίσκει το πρώτο μη κενό τετράγωνο στα αριστερά του τρέχοντος τετραγώνου που σαρώνεται.

ΠΑΡΑΔΕΙΓΜΑ

Η μηχανή αντιγραφής C επιτελεί την παρακάτω λειτουργία: αν μια συμβολοσειρά ω, η οποία περιέχει μόνο μη κενά σύμβολα αλλα ενδεχομένως άδεια, τοποθετηθεί σε μια ταινία κατά τα άλλα κενή με ένα κενό τετράγωνο στα αριστερά της, και η κεφαλή τοποθετηθεί στο κενό τετράγωνο που σημαδεύει το τέλος της συμβολοσειράς ω, τότε η μηχανή τελικά θα σταματήσει με το ω#ω σε μια κατά τα άλλα κενή ταινία, με ένα κενό τετράγωνο στα αριστερά και την κεφαλή στο κενό τετράγωνο που σημαδεύει το άκρο της όλης συμβολοσειράς ω#ω. Τυπικά, αν ς είναι η αρχική κατάσταση της C, τότε (ς,#ω#) ⊢*C (h,#ω#ω#). Για να αποφύγουμε την άμεση αναφορά στην αρχική κατάσταση της C, λέμε ότι η C μετασχηματίζει το #ω# σε #ω#ω#, και χρησιμοποιούμε παρόμοιες εκφράσεις όταν περιγράφουμε άλλες μηχανές.

Ένα διάγραμμα για τη C δίνεται στο σχήμα παρακάτω.



Ας παρακολουθήσουμε τη λειτουργία της C με είσοδο το abc#. Μετά την L#, η είσοδος έχει μετασχηματισθεί σε #abc#. Η κεφαλή μετακινείται ένα τετράγωνο προς τα δεξιά για να σχηματίσει το #abc#. Το a δεν είναι κενό, για αυτό γράφεται ένα κενό σύμβολο (##bc#). Το δεύτερο κενό σύμβολο στα δεξιά αυτού που σαρώνεται βρίσκεται (##bc##) και εκεί γράφεται ένα α εκεί (##bc#a). Το δεύτερο κενό σύμβολο στα αριστερά αυτού που σαρώνεται βρίσκεται (##bc#a) και το α ξαναγράφεται (#abc#a). Μερικές από τις επόμενες συνολικές καταστάσεις της ταινίας και της κεφαλής μπορούν να απεικονιστούν ως εξής:

#abc#a

#a#c#a

#a#c#a#

#a#c#ab

#a#c#ab

#abc#ab τέλος άλλου ενός κύκλου

#abc#ab

.

.

.

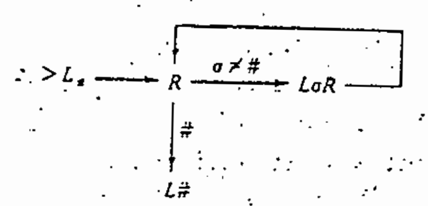
#abc#abc τέλος ενός τρίτου κύκλου

Στο σημείο αυτό, όταν η κεφαλή μετακινείται προς τα δεξιά, βρίσκεται ένα κενό σύμβολο (#abc#abc). Βρίσκεται το επόμενο κενό (#abc#abc#) στα δεξιά αυτού που βρέθηκε, και η μηχανή τερματίζει.

Σημειώστε ότι, από το Λήμμα, η C θα αντιγράψει την τελευταία ομάδα από μη κενά σύμβολα στην ταινία, αν υπάρχουν περισσότερες από μία ομάδες. Για παράδειγμα, αν u, υ και ω είναι συμβολοσειρές από μη κενά σύμβολα, τότε η C μετασχηματίζει το #u#υ#ω# σε #u#υ#ω##. Φυσικά, η C δεν θα λειτουργήσει σωστά αν ξεκινήσει με μερικά μη κενά σύμβολα στα δεξιά της κεφαλής.

ΠΑΡΑΔΕΙΓΜΑ

Η μηχανή αριστερής μετατόπισης S_L μετασχηματίζει το #ω#, όπου το ω δεν περιέχει κενά, σε ω#. Εικονίζεται στο Σχήμα.



Για να διευκρινίσουμε την λειτουργία της S_R μετασχηματίζει το #ω#, όπου ω δεν περιέχει κενά, σε ##ω#. Η κατασκευή της αφήνεται ως άσκηση.

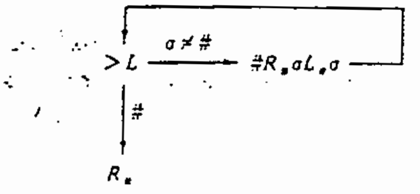
ΠΑΡΑΔΕΙΓΜΑ

Έστω Σ ένα αλφάβητο που περιέχει το # και έστω $\Sigma_0=\Sigma-\{\#\}$. Η συνάρτηση $f:\Sigma_0\text{->}\Sigma_0^*$ υπολογίζεται από τη μηχανή Turing CS_L. Η μηχανή αυτή απλώς χρησιμοποιεί C για να αντιγράψει τη συμβολοσειρά εισόδου αφήνοντας ένα κενό ανάμεσα στα δύο αντίγραφα.

ΠΑΡΑΔΕΙΓΜΑ

Έστω ότι τα Σ και Σ_0 είναι όπως στο Παράδειγμα και έστω ότι η $f:\Sigma_0^*\text{->}\Sigma_0^*$ ορίζεται από τη σχέση f(ω)=

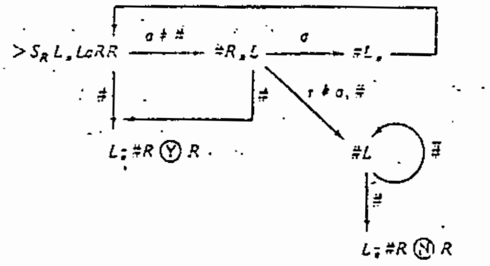
$\omega\omega^R$. Τότε η f υπολογίζεται από τη μηχανή Turing του Σχήματος.



Αυτή η μηχανή μετασχηματίζει την είσοδο $\#\sigma_1....\sigma_n\underline{\#}$ ($\sigma_1,...,\sigma_n\in \Sigma$) διαδοχικά σε $\#\sigma_1....\underline{\sigma_i}...\sigma_n\sigma_n\sigma_{n-1}...\sigma_{i+1}$ για $i=n-1, n-2,...,0$. Στην περίπτωση $i=0$, έχουμε $\underline{\#}\sigma_1...\sigma_n\sigma_n...\sigma_1$, το οποίο τελικά μετασχηματίζεται σε $\#\sigma_1...\sigma_n\sigma_n...\sigma_1\underline{\#}$.

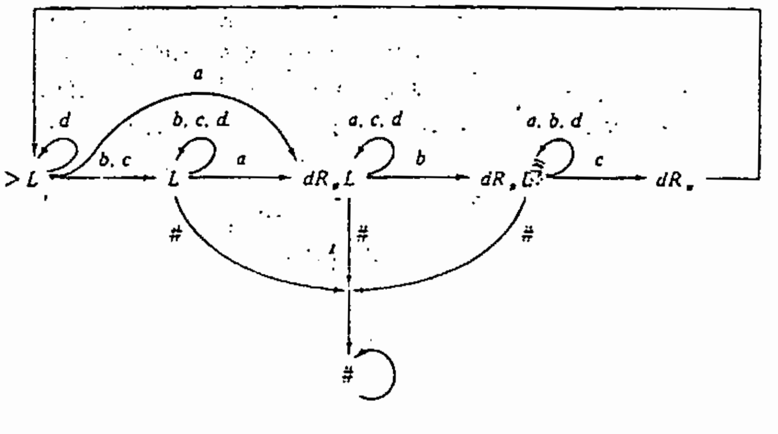
ΠΑΡΑΔΕΙΓΜΑ

Έστω $\Sigma_0 = \{\alpha,b\}$ και έστω $L=\{\omega\in\sigma_0^*:\omega=\omega^R\}$. Τότε η L αποφασίζεται από τη μηχανή Tyring του Σχήματος 4-14. Η μηχανή αυτή μετασχηματίζει πρώτα το $\#\omega\underline{\#}$ σε $\alpha\underline{\#}\omega$. Κατόπιν σβήνει εναλλάξ σύμβολα από τις άκρες του ω, ελέγχοντας αν το σύμβολο που σβήστηκε από τη δεξιά άκρη είναι το ίδιο με αυτό που σβήστηκε λίγο πριν από την αριστερή. Αν όλα τα σύμβολα έχουν σβηστεί, η μηχανή επιστρέφει στην αριστερή άκρη της ταινίας και το αντικαθιστά με $\#(Y)\underline{\#}$. Αν έχει παρουσιαστεί αναντιστοιχία, το υπόλοιπο του ω σβήνεται και στην ταινία αήνεται το $\#(N)\underline{\#}$.



ΠΑΡΑΔΕΙΓΜΑ

Έστω $\Sigma_0=\{\alpha,b,c\}$ και έστω $L=\{\omega\in\Sigma_0^*$: το ω έχει ίσο αριθμό από α, b και c$\}$. Τότε η L γίνεται δεκτή από τη μηχανή Turing που εικονίζεται στο Σχήμα. Σε έναν κύριο κύκλο της λειτουργίας της μηχανής αυτής, αλλάζουν α, και b και ένα c στην ταινία μετατρέπονται σε τρία d. Αυτό συντελείται με σάρωμα του μη κενού μέρους της ταινίας τρεις φορές από δεξιά προς τα αριστερά, μια φορά για ένα α, μια φορά για ένα b και μια φορά για ένα c, και με επιστροφή κάθε φορά προς το δεξιό άκρο της ταινίας προτού αρχίσει το σάρωμα. Η λειτουργία αυτή επαναλαμβάνεται ώσπου να εξαντληθεί το απόθεμα ενός από τα σύμβολα (α,b ή c). Αν ψάχνουμε για α και μένουν μόνο d στο μη κενό τμήμα της ταινίας, η μηχανή τερματίζει κι έτσι αναγνωρίζει τη συμβολοσειρά εισόδου. Σε όλες τις άλλες περιπτώσεις, μπαίνει σε έναν ατέρμονα υπολογισμό και συνεπώς δεν αναγνωρίζει τη συμβολοσειρά εισόδου.

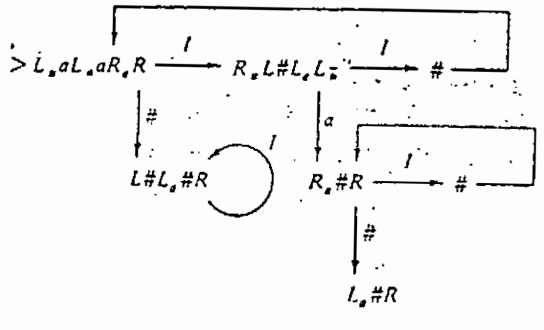


ΠΑΡΑΔΕΙΓΜΑ

Η αφαίρεση δεν ορίζεται για όλα τα ζεύγη των φυσικών αριθμών, αφού η διαφορά μεταξύ δύο φυσικών αριθμών μπορεί να είναι αρνητική. Ωστόσο, η κατά πολύ παρόμοια συνάρτηση ∸ (monus) έχει έναν φυσικό αριθμό ως τιμή της για κάθε ζεύγος φυσικών αριθμών ως ορισμάτων. Ο ορισμός της monus είναι:

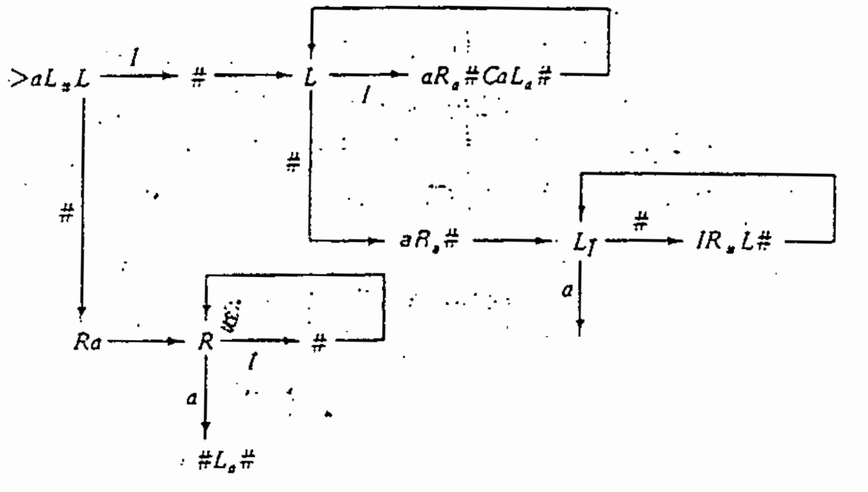
$$\chi \dot{-} \psi = \begin{cases} \chi - \psi & \text{αν } \chi \geq \psi \\ 0 & \text{αν } \chi < \psi \end{cases}$$

Η συνάρτηση monus υπολογίζεται από τη μηχανή Turing που εικονίζεται στο Σχήμα 4-16. Εδώ το α είναι κάποιο σταθερό σύμβολο και L_α και R_α είναι μηχανές που ανιχνεύουν για την πρώτη εμφάνιση του α στα αριστερά και στα δεξιά, αντίστοιχα, της τρέχουσας θέσης της κεφαλής. Η εικονιζόμενη μηχανή πρώτα μετασχηματίζει τη συμβολοσειρά εισόδου $\#I^m\#I^n\underline{\#}$ σε $\underline{\alpha}I^m\alpha I^n$. Ύστερα μετακινεί επανειλημμένα ένα I από την άκρη κάθε ομάδας από I, ώσπου να μη μείνουν άλλα στη μία ομάδα ή στην άλλη. Τελικά σβήνει τα α, σβήνει τα I που απέμειναν αν προέκυψε ότι n>m, επανατοποθετεί την κεφαλή σωστά και τερματίζει.



ΠΑΡΑΔΕΙΓΜΑ

Η συνάρτηση πολλαπλασιασμού f (n,m)=n.m υπολογίζεται από τη μηχανή Turing που εικονίζεται στο Σχήμα4-17. Η μηχανή αυτή ελέγχει πρώτα αν κάποιο όρισμα είναι μηδέν και, αν συμβαίνει αυτό, τερματίζει με την κενή συμβολοσειρά ως έξοδο. Αν όχι, αντιγράφει τη δεύτερη συμβολοσειρά τόσες φορές όσες δίνονται από την πρώτη συμβολοσειρά τόσες φορές όσες δίνονται από την πρώτη συμβολοσειρά και τελικά μετακινεί αυτά τα αντίγραφα στα αριστερά για να αφαιρέσει όλα τα κενά.



ΒΙΒΛΙΟΓΡΑΦΙΑ

Στοιχεία Θεωρίας υπολογισμού Harry Lewis - Χρήστος Παπαδημητρίου Βιβλιοθήκη Πληροφορικής Τεχνικού Επιμελητηρίου Ελλάδος.

Σημειώσεις Θεωρίας Υπολογισμού, Παύλος Σπυράκης, Πανεπιστήμιο Πατρών, Καθηγήτρια Δρ. . Αντωνοπούλου Ηρα.