Θέση των Church-Turing

Όλες οι προσπάθειες τις οποίες κάναμε προκειμένου να ισχυροποιήσουμε τις Μ.Τ., έπεσαν στο κενό.

Θέση των Church-Turing: Η μηχανή Turing που τερματίζει για όλες τις εισόδους αποτελεί την αυστηρή τυποποίηση της διαισθητικής έννοιας του «αλγορίθμου». Ισοδύναμα: τίποτα δεν είναι αλγόριθμος αν δεν μπορεί να εκφραστεί ως Μ. Τ. που τερματίζει για όλες τις εισόδους.

Η θέση των Church-Turing δεν είναι θεώρημα, και επομένως δεν μπορεί να αποδειχτεί. Είναι (θεωρητικά) δυνατό να βρεθεί στο μέλλον κάποιο ισχυρό μοντέλο υπολογισμού, το οποίο θα καταρρίψει τη θέση των Church-Turing. Κάτι τέτοιο θεωρείται σήμερα απίθανο.

Καθολικές Μηχανές Turing

Μέχρι τώρα, κάθε Μ. Τ. που εξετάσαμε ήταν σχεδιασμένη να λύνει ένα και μοναδικό πρόβλημα (να αποφασίζει μια συγκεκριμένη γλώσσα L, να υπολογίζει μια συγκεκριμένη συνάρτηση f κοκ).

Θα δείξουμε πως οι Μ. Τ. είναι «προγραμματίσιμες», μπορούν δηλ. να εκτελέσουν οποιονδήποτε αλγόριθμο.

Από τη θέση των Church-Turing, αλγόριθμος για μας είναι μια Μ. Τ. Άρα...

Καθολικές Μηχανές Turing

... θα ορίσουμε μια μηχανή Turing η οποία όταν παίρνει ως είσοδο την περιγραφή μιας μηχανής M, θα συμπεριφέρεται όπως η M.

Για να μπορούμε να αναπαραστήσουμε μια οποιαδήποτε μηχανή Turing, θα πρέπει να βρούμε τρόπο να κωδικοποιήσουμε τις καταστάσεις και τα σύμβολα ταινίας της χρησιμοποιώντας ένα προκαθορισμένο αλφάβητο.

Αναπαριστούμε καταστάσεις στη μορφή qw, όπου $w \in \{0,1\}^*$. Ομοίως, τα σύμβολα ταινίας στη μορφή aw, όπου $w \in \{0,1\}^*$.

Παράδειγμα Αναπαράστασης

$M = (K, \Sigma, \delta, s, \{h\})$
$K = \{s, q, h\}$
$\Sigma = \{a, \sqcup, \triangleright\}$

Κατάσταση q	**Σύμβολο** σ	**Μετάβαση** $\delta(q,\sigma)$
s	a	$(q, \sqcup)$
s	$\sqcup$	$(h, \sqcup)$
s	$\triangleright$	$(s, \rightarrow)$
q	a	(s, a)
q	$\sqcup$	$(s, \rightarrow)$
q	$\triangleright$	$(q, \rightarrow)$

Παράδειγμα Αναπαράστασης, συνέχεια

Αναπαριστούμε τις καταστάσεις και τα στοιχεία του αλφαβήτου της M ως εξής:

Κατάσταση	Αναπαράσταση
s	$q00$
q	$q01$
h	$q10$
$\sqcup$	$a000$
$\triangleright$	$a001$
$\leftarrow$	$a010$
$\rightarrow$	$a011$
a	$a100$

Η αναπαράσταση $\ll M \gg$ της μηχανής Turing είναι η συμβολοσειρά:

$$\ll M \gg = (q00, a100, q01, a000), \ldots, (q01, a001, q01, a011)$$

Υπολογισμοί με Καθολικές Μηχανές Turing

Μπορούμε τώρα να ορίσουμε μια καθολική μηχανή Turing U, η οποία έχει δύο ορίσματα: την περιγραφή $\ll M \gg$ μιας μηχανής M και την περιγραφή $\ll w \gg$ μιας συμβολοσειράς w.

Θελουμε η U να τερματίζει με είσοδο $\ll M \gg\ll w \gg$ αν και μόνο αν η M τερματίζει με είσοδο w.

Είναι σχετικά απλό να σχεδιάσουμε τη U με τέτοιο τρόπο ώστε να προσομοιώνει τη λειτουργία της M (είναι απλούστερο αν η U χρησιμοποιεί τρείς ταινίες).

Stored Program Computer

Με όρους αρχιτεκτονικής υπολογιστών, αυτό που κάναμε ήταν να εισαγάγουμε την έννοια του stored program computer.

Μια από τις σπουδαιότερες επιστημονικές ανακαλύψεις του 20ου αιώνα.

Όρια των Μ. Τ. (άρα και των αλγορίθμων)

Μπορούμε να βρούμε προβλήματα τα οποία να μην επιλύονται από κανένα αλγόριθμο;

Τέτοια προβλήματα υπάρχουν: όλοι οι φορμαλισμοί που έχουμε εξετάσει (κανονικές γλώσσες, γλώσσες χωρίς συμφραζόμενα, μηχανές Turing, κλπ), μπορούν να αναπαραστήσουν (αναγνωρίσουν) αριθμήσιμα άπειρες το πλήθος γλώσσες. Το σύνολο όλων των γλωσσών είναι μη αριθμήσιμα άπειρο.

Πώς μπορούμε όμως να βρούμε μια τέτοια γλώσσα που δεν αποφασίζεται από μια μηχανή Turing;

Θυμηθείτε, έχουμε ήδη ορίσει μηχανές Turing που παίρνουν ως είσοδο κωδικοποιήσεις μηχανών Turing.

Το Πρόβλημα του Τερματισμού (Halting Problem)

Έστω ότι έχουμε καταφέρει να γράψουμε ένα πρόγραμμα `halts(P,X)` το οποίο μας επιστρέφει την απάντηση 'yes' αν, με είσοδο `X`, το πρόγραμμα `P` τερματίζει και 'no' διαφορετικά.

Θεωρείστε τώρα το εξής πρόγραμμα:

```
procedure diagonal(X)
L : if halts(X,X)
    then goto L
    else stop
```

Τερματίζει το `diagonal(diagonal)`;

Τερματίζει
αν και μόνο αν το `halts(diagonal,diagonal)` επιστρέφει 'no', δηλαδή
αν και μόνο άν το `diagonal` με είσοδο `diagonal` δεν τερματίζει (άτοπο).

Συμπέρασμα: δεν υπάρχει τέτοιο πρόγραμμα `halts`.

Το Πρόβλημα του Τερματισμού, συνέχεια

Με οδηγό τον προηγούμενο συλλογισμό, θα βρούμε μια γλώσσα που είναι αναδρομικά απαριθμήσιμη αλλά όχι αναδρομική. Έστω η γλώσσα:

$H = \{\ll \mathsf{M} \gg\ll \mathsf{w} \gg$: η μηχανή Turing M τερματίζει όταν έχει είσοδο w$\}$

Η γλώσσα H είναι προφανώς αναδρομικά απαριθμήσιμη καθώς ημιαποφασίζεται από τη μηχανή U. Είναι όμως η H αναδρομική;

Έστω ότι η H είναι αναδρομική. Τότε, το ίδιο θα ισχύει και για τη γλώσσα:

$H_1 = \{\ll \mathsf{M} \gg$: η μηχανή Turing M τερματίζει όταν έχει είσοδο $\ll \mathsf{M} \gg\}$

Αφού η H_1 είναι αναδρομική, τότε το ίδιο ισχύει και για το συμπλήρωμά της:

$\overline{H_1} =$ {w : η w δεν είναι κωδικοποίηση μηχανής Turing
ή είναι κωδικοποίηση μιας μηχανής Turing M
που δεν τερματίζει όταν έχει είσοδο $\ll \mathsf{M} \gg$}

Το Πρόβλημα του Τερματισμού, συνέχεια

Όμως, η $\overline{H_1}$ δεν είναι ούτε καν αναδρομικά απαριθμήσιμη: έστω M^* μια μηχανή Turing που ημιαποφασίζει την $\overline{H_1}$. Ανήκει η $\ll \mathsf{M}^* \gg$ στην $\overline{H_1}$;

Από τον ορισμό της $\overline{H_1}$, $\ll \mathsf{M}^* \gg \in \overline{H_1}$ αν και μόνο αν η M^* δεν δέχεται τη συμβολοσειρά $\ll \mathsf{M}^* \gg$.

Όμως, η M^* ημιαποφασίζει την $\overline{H_1}$, και έτσι $\ll \mathsf{M}^* \gg \in \overline{H_1}$ αν και μόνο αν η M^* δέχεται τη συμβολοσειρά $\ll \mathsf{M}^* \gg$

Καταλήξαμε σε άτοπο, και επομένως η H δεν είναι αναδρομική.

Το Πρόβλημα του Τερματισμού, συνέχεια

Θεώρημα: Η κλάση των αναδρομικών γλωσσών είναι ένα γνήσιο υποσύνολο της κλάσης των αναδρομικά απαριθμήσιμων γλωσσών.

Επιπλέον, όπως είδαμε, η γλώσσα H_1 είναι αναδρομικά απαριθμήσιμη ενώ το συμπλήρωμά της δεν είναι. Κατά συνέπεια:

Θεώρημα: Η κλάση των αναδρομικά απαριθμήσιμων γλωσσών δεν είναι κλειστή ως προς το συμπλήρωμα.

Μη Επιλύσιμα Προβλήματα

Το πρώτο συμπέρασμα που προκύπτει από την προηγούμενη συζήτηση είναι ότι δεν υπάρχει αλγόριθμος ο οποίος για κάθε μηχανή Turing M και συμβολοσειρά εισόδου w, να αποφασίζει αν η M δέχεται τη w ή όχι.

Προβλήματα όπως το παραπάνω θα τα ονομάζουμε μη επιλύσιμα.

Προσοχή: Το πρόβλημα του τερματισμού μπορεί να λύνεται σε κάποιες ειδικές περιπτώσεις. Αυτό που έχουμε αποδείξει είναι ότι δεν υπάρχει μια γενική μέθοδος η οποία να αποφασίζει σωστά σε όλες τις περιπτώσεις.

Μη Επιλύσιμα Προβλήματα, συνέχεια

Πώς μπορούμε να αποδείξουμε ότι ένα πρόβλημα είναι μη επιλύσιμο (χωρίς να χρησιμοποιήσουμε την τεχνική που ακολουθήσαμε για την H;)

Θα χρησιμοποιήσουμε την έννοια της αναγωγής.

Γενικά για τις αναγωγές

Μια αναγωγή είναι ένας τρόπος να μετατρέπουμε ένα πρόβλημα Π_1 σε ένα πρόβλημα Π_2, έτσι ώστε μια λύση του Π_2 να μπορεί να χρησιμοποιηθεί για να λύσουμε το Π_1.

Λέμε τότε πως ανάγουμε το Π_1 στο Π_2.

Η αναγωγή έχει κατεύθυνση: μια αναγωγή από το Π_1 στο Π_2, δεν σημαίνει απαραίτητα πως υπάρχει αναγωγή από το Π_2 στο Π_1.

Παράδειγμα αναγωγής

Έστω `Sort(A)` ένας *οποιοσδήποτε* αλγόριθμος που διατάσσει τα στοιχεία ενός array A.

Ο παρακάτω αλγόριθμος βρίσκει το ελάχιστο στοιχείο μέσα σε ένα array.

`Find-Min(array A)`

```
Sort(A);
return(A[1]);
```

Ο αλγόριθμος Find-Min ορίζει μια αναγωγή από το πρόβλημα της εύρεσης του ελάχιστου στοιχείου στο πρόβλημα της διάταξης.

Από την αναγωγή συμπεραίνουμε πως αν το πρόβλημα της διάταξης είναι επιλύσιμο, τότε σίγουρα και πρόβλημα της εύρεσης ελάχιστου στοιχείου είναι επιλύσιμο.

Παράδειγμα αναγωγής (συνέχεια)

```
Find-Min(array A)

    Sort(A);
    return(A[1]);
```

Ο αλγόριθμος Find-Min ορίζει μια αναγωγή από το πρόβλημα της εύρεσης του ελάχιστου στοιχείου στο πρόβλημα της διάταξης.

Από την αναγωγή συμπεραίνουμε πως αν το πρόβλημα της διάταξης είναι επιλύσιμο, τότε σίγουρα και πρόβλημα της εύρεσης ελάχιστου στοιχείου είναι επιλύσιμο.

(Βεβαίως γνωρίζουμε πως και τα δύο προβλήματα είναι επιλύσιμα μέσω γρήγορων αλγορίθμων).

Αναγωγές μεταξύ γλωσσών

Μια αναγωγή από τη γλώσσα L_1 στη γλώσσα L_2 είναι ένας αλγόριθμος (δηλ. Μηχανή Turing) M_1 ο οποίος αποφασίζει την L_1 καλώντας σαν υπορουτίνα έναν *οποιονδήποτε* αλγόριθμο (δηλ. Μ. Τ.) M_2 ο οποίος αποφασίζει την L_2.

M_1(είσοδος w)

$\vdots$
$M_2(w')$;
$\vdots$
$M_2(w'')$;
$\vdots$
return(YES or NO accordingly); // $w \in L_1$ or not

Ειδική περίπτωση αναγωγής

Το βιβλίο των Lewis-Παπαδημητρίου χρησιμοποιεί μόνο αναγωγές της μορφής:

M_1(είσοδος w)

Compute "suitable" w';
If ($M_2(w')$== YES) then
    return(YES); // $w \in L_1$
else
    return(NO); // $w \notin L_1$;

δηλ. η M_1 καλεί τη M_2 μόνο μία φορά με είσοδο $w' = \tau(w)$, όπου η αναδρομική συνάρτηση $\tau : \Sigma^* \to \Sigma^*$ είναι τέτοια ώστε w $\in$ L_1 αν και μόνο αν τ(w) $\in$ L_2.

Σχεδόν όλες οι αναγωγές που θα δούμε στο μάθημα θα είναι αυτής της ειδικής μορφής. Η δυσκολία έγκειται στον υπολογισμό του κατάλληλου w', έτσι ώστε να ισχύει το «αν και μόνο αν».

Συνέπειες της ύπαρξης μιας αναγωγής

M_1(είσοδος w)

$\vdots$
$M_2(w')$;
$\vdots$
$M_2(w'')$;
$\vdots$
return(YES or NO accordingly); // $w \in L_1$ or not

Θεώρημα 1: Αν η L_2 είναι αναδρομική και υπάρχει μια αναγωγή από την L_1 στην L_2, τότε και η L_1 είναι αναδρομική.
Απόδειξη: Έστω ότι η L_2 είναι αναδρομική τότε υπάρχει μια μηχανή Turing M_2 που την αποφασίζει. Επομένως η Μ. Τ. M_1 της παραπάνω αναγωγής είναι καλά ορισμένη και αποφασίζει την L_1.

Συνέπειες της ύπαρξης μιας αναγωγής (συνέχεια)

Μια αναγωγή από τη γλώσσα L_1 στη γλώσσα L_2 είναι ένας αλγόριθμος (δηλ. Μηχανή Turing) M_1 ο οποίος αποφασίζει την L_1 καλώντας σαν υπορουτίνα έναν *οποιονδήποτε* αλγόριθμο (δηλ. Μ. Τ.) M_2 ο οποίος αποφασίζει την L_2.

Θεώρημα 1: Αν η L_2 είναι αναδρομική και υπάρχει μια αναγωγή από την L_1 στην L_2, τότε και η L_1 είναι αναδρομική.

Το παρακάτω θεώρημα είναι ισοδύναμο (γιατί;) με το Θεώρημα 1:

Θεώρημα 2: Αν η L_1 δεν είναι αναδρομική και υπάρχει μια αναγωγή από την L_1 στην L_2, τότε και η L_2 δεν είναι αναδρομική.

Μη Επιλύσιμα Προβλήματα και αναγωγές

Έστω πως γνωρίζουμε πως ένα πρόβλημα (γλώσσα) L_1 (π.χ. το Πρόβλημα του Τερματισμού) είναι μη επιλύσιμο (μη αναδρομική).

Θέλουμε να αποδείξουμε ότι ένα καινούργιο πρόβλημα L_2 είναι μη επιλύσιμο (δηλ. η αντίστοιχη γλώσσα είναι μη αναδρομική).

Με εις άτοπο απαγωγή.
1) Υποθέτουμε πως το L_2 είναι επιλύσιμο.
2) Δείχνουμε αναγωγή από το L_1 στο L_2.
3) Από το Θεώρημα 1 έπεται πως και το L_1 είναι επιλύσιμο, άτοπο. Άρα και το L_2 είναι μη επιλύσιμο.

Μη Επιλύσιμα Προβλήματα Ι

Θεώρημα: Το παρακάτω πρόβλημα είναι μη επιλύσιμο:
'Δεδομένης μιας μηχανής Turing M, τερματίζει η M με κενή ταινία;'

Απόδειξη: Με αναγωγή από το πρόβλημα του τερματισμού. Έστω μια μηχανή Turing M και μια συμβολοσειρά w.

Η αναγωγή μας κατασκευάζει μια μηχανή Turing M_w η οποία ξεκινάει τη λειτουργία της με κενή ταινία. Αμέσως μετά, η M_w γράφει το w στην ταινία της και ξεκινάει να προσομοιώνει τη μηχανή M.

Αν είχαμε έναν αλγόριθμο ο οποίος αποφασίζει αν μια μηχανή Turing τερματίζει με κενή ταινία, τότε θα μπορούσαμε να αποφασίσουμε αν η M_w τερματίζει, και επομένως να αποφασίσουμε αν η M με είσοδο w τερματίζει.

Μη Επιλύσιμα Προβλήματα, συνέχεια

Θεώρημα: Το παρακάτω πρόβλημα είναι μη επιλύσιμο:
'Δεδομένης μιας μηχανής Turing M, υπάρχει έστω και μία συμβολοσειρά για την οποία η M τερματίζει;'

Απόδειξη: Αν το πρόβλημα αυτό ήταν επιλύσιμο, τότε το ίδιο θα ίσχυε και για το αμέσως προηγούμενο πρόβλημα.

Θα μπορούσαμε να αποφασίσουμε αν μια μηχανή Turing M τερματίζει με κενή συμβολοσειρά, ως εξής: τροποποιούμε την M ώστε να σβήνει οποιαδήποτε είσοδο της δίνεται και μετά την αφήνουμε να προχωρήσει όπως θα προχωρούσε αν η είσοδος της ήταν εξαρχής η κενή συμβολοσειρά.

Για τη νέα αυτή μηχανή M′ ισχύει το εξής: η M′ τερματίζει για μια δεδομένη συμβολοσειρά αν και μόνο αν η M′ τερματίζει για κάθε συμβολοσειρά αν και μόνο αν η M τερματίζει για την κενή συμβολοσειρά.

Επομένως, θα μπορούσαμε να ελέγξουμε αν η M τερματίζει με είσοδο την κενή συμβολοσειρά απλά ελέγχοντας αν υπάρχει συμβολοσειρά για την οποία η M′ τερματίζει.

Μη Επιλύσιμα Προβλήματα II

Θεώρημα: Το παρακάτω πρόβλημα είναι μη επιλύσιμο:
'Δεδομένης μιας μηχανής Turing M, τερματίζει η M για κάθε συμβολοσειρά εισόδου;'

Απόδειξη: Η απόδειξη που παρουσιάσαμε για το αμέσως προηγούμενο ερώτημα μεταφέρεται και εδώ (γιατί η M' δέχεται κάποια είσοδο αν και μόνο αν δέχεται κάθε είσοδο).

Μη Επιλύσιμα Προβλήματα III

Θεώρημα: Το παρακάτω πρόβλημα είναι μη επιλύσιμο:
'Δεδομένων δύο μηχανών Turing M_1 και M_2, τερματίζουν με τις ίδιες συμβολοσειρές εισόδου;'

Απόδειξη: Έστω ότι μπορούμε να αποφασίσουμε αν δύο μηχανές τερματίζουν με τις ίδιες συμβολοσειρές εισόδου.

Θέτουμε τη μία από αυτές να είναι μια τετριμμένη μηχανή που τερματίζει για όλες τις πιθανές εισόδους.

Επομένως, τώρα έχουμε ένα αλγόριθμο ο οποίος αποφασίζει αν μια οποιαδήποτε μηχανή Turing τερματίζει με κάθε είσοδο.

Όμως, γνωρίζουμε ήδη ότι το πρόβλημα αυτό είναι μη επιλύσιμο.

Ιδιότητες των Αναδρομικών Γλωσσών

Θεώρημα: Μία γλώσσα L είναι αναδρομική αν και μόνο αν και αυτή και το συμπλήρωμά της είναι αναδρομικά απαριθμήσιμες.

Απόδειξη: Η κατεύθυνση ($\Rightarrow$) είναι προφανής (γιατί;)

Για την κατεύθυνση ($\Leftarrow$) θα κατασκευάσουμε μια μηχανή Turing M δύο ταινιών που αποφασίζει την L. Έστω ότι η L ημιαποφασίζεται από μια μηχανή Turing M_1 και η $\overline{L}$ από μία μηχανή Turing M_2.

Δεδομένης μιας συμβολοσειράς w, η M προχωράει στην πρώτη ταινία της όπως θα προχωρούσε η M_1 με είσοδο w, και στη δεύτερη ταινία της όπως η M_2 με είσοδο w.

Κάποια στιγμή ο υπολογισμός σε μία από τις δύο ταινίες θα τερματίσει. Τότε και η M τερματίζει στην κατάσταση y ή n (ανάλογα με το αν ο υπολογισμός ο οποίος τερμάτισε ήταν αυτός της πρώτης ταινίας ή της δεύτερης ταινίας αντίστοιχα).

Ιδιότητες των Αναδρομικών Γλωσσών, συνέχεια

Ορισμός: Μια μηχανή Turing M απαριθμεί τη γλώσσα L αν και μόνο αν, για κάποια προκαθορισμένη κατάσταση q της M,

$$L = \{w : (s, \triangleright\sqcup) \vdash_M^* (q, \triangleright\sqcup w)\}$$

Μία γλώσσα είναι απαριθμήσιμη κατά Turing αν και μόνο αν υπάρχει μία μηχανή Turing που την απαριθμεί.

Διαισθητικά: Όταν η M εισέρχεται στην κατάσταση q τότε σηματοδοτεί ότι η συμβολοσειρά που βρίσκεται αυτή τη στιγμή στην ταινία της, ανήκει στην L. Η M ακολούθως μπορεί να αφήσει την q και να επανέλθει αργότερα με ένα άλλο μέλος της L.

Προσοχή: Τα μέλη της L μπορούν να παρατεθούν σε οποιαδήποτε σειρά και με επαναλήψεις.

Ιδιότητες των Αναδρομικών Γλωσσών, συνέχεια

Θεώρημα: Μία γλώσσα L είναι αναδρομικά απαριθμήσιμη αν και μόνο αν είναι απαριθμήσιμη κατά Turing.

Απόδειξη: ($\Leftarrow$) Πρώτα αποδεικνύουμε ότι αν έχουμε μια μηχανή Turing E που απαριθμεί την L, τότε υπάρχει μια μηχανή Turing M που ημιαποφασίζει την L. Η M δουλεύει ως εξής:

Για συμβολοσειρά εισόδου w, η M αρχίζει να «τρέχει» τη μηχανή E. Κάθε φορά που η E εισέρχεται στην ειδική κατάστασή της, η M συγκρίνει τη w με τη συμβολοσειρά που μόλις έχει παραχθεί από την E. Αν είναι ίδιες, τότε η M αποδέχεται τη w. Διαφορετικά, συνεχίζει την παραπάνω διαδικασία.

Είναι προφανές ότι η M δέχεται ακριβώς εκείνες τις συμβολοσειρές οι οποίες εμφανίζονται στην απαρίθμηση της E.

Ιδιότητες των Αναδρομικών Γλωσσών, συνέχεια

Απόδειξη (συνέχεια): ($\Rightarrow$) Έστω ότι έχουμε μια μηχανή Turing M που ημιαποφασίζει την L. Μπορούμε να κατασκευάσουμε μια μηχανή Turing E που απαριθμεί την L. Έστω $\mathsf{s}_1, \mathsf{s}_2, \ldots$ μια ακολουθία από όλες τις συμβολοσειρές του Σ^*. Η E λειτουργεί ως εξής:

Επανέλαβε τα ακόλουθα βήματα για $i = 1, 2, 3, \ldots$.

1. «Τρέξε» την M για i βήματα για τις συμβολοσειρές $\mathsf{s}_1, \mathsf{s}_2, \ldots, \mathsf{s}_i$.
2. Αν κάποια συμβολοσειρά s_j γίνεται δεκτή, η E πηγαίνει στην ειδική κατάσταση q με s_j στην ταινία.

Αν η M αποδέχεται μια συμβολοσειρά w, τότε αυτή θα εμφανιστεί αργά ή γρήγορα στην ακολουθία των συμβολοσειρών που απαριθμεί η E.

Ιδιότητες των Αναδρομικών Γλωσσών, συνέχεια

Ορισμός: Μια μηχανή Turing M απαριθμεί λεξικογραφικά τη γλώσσα L αν,

1. απαριθμεί την L περνώντας από μια προκαθορισμένη κατάσταση q και
2. Όποτε έχουμε:
$$(\mathsf{q}, \triangleright\sqcup\mathsf{w}) \vdash_{\mathsf{M}}^{+} (\mathsf{q}, \triangleright\sqcup\mathsf{w}')$$
τότε η w′ βρίσκεται λεξικογραφικά μετά από την w.

Μία γλώσσα είναι λεξικογραφικά απαριθμήσιμη κατά Turing αν και μόνο αν υπάρχει μία μηχανή Turing που την απαριθμεί λεξικογραφικά.

Ιδιότητες των Αναδρομικών Γλωσσών, συνέχεια

Θεώρημα: Μία γλώσσα L είναι αναδρομική αν και μόνο αν είναι λεξικογραφικά απαριθμήσιμη κατά Turing.

Απόδειξη: ($\Rightarrow$) Αφού η L είναι αναδρομική, τότε υπάρχει μηχανή Turing M που την αποφασίζει. Κατασκευάζουμε μία μηχανή Turing M′ η οποία παράγει λεξικογραφικά όλες τις συμβολοσειρές του αλφαβήτου της L. Για κάθε μία από τις συμβολοσειρές «τρέχει» την M. Όποτε η M δέχεται την είσοδο, η M′ παρουσιάζει τη συμβολοσειρά και συνεχίζει στην επόμενη.

($\Leftarrow$) Αφού η L είναι λεξικογραφικά απαριθμήσιμη, μπορούμε να κατασκευάσουμε μια μηχανή Turing M′ που την αποφασίζει. Η M′ με είσοδο w ξεκινάει την απαρίθμηση της L και περιμένει έως ότου είτε η w παρουσιαστεί είτε ξεπεραστεί λεξικογραφικά. Στην πρώτη περίπτωση δέχεται την w ενώ στη δεύτερη την απορρίπτει.