Θεωρία Υπολογισμού

3ο Σετ Ασκήσεων

Σταυρόπουλος Αλέξανδρος Ανδρέας
2019030109

Διδάσκων:
Μιχαήλ Λαγουδάκης

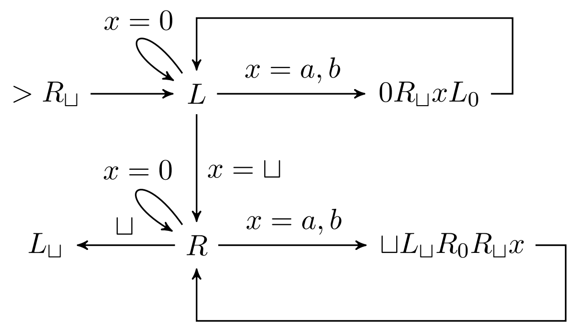
Υπεύθυνος εργαστηρίου:
-

ΗΜΜΥ
Πολυτεχνείο Κρήτης
Εαρινό εξάμηνο 2022-2023

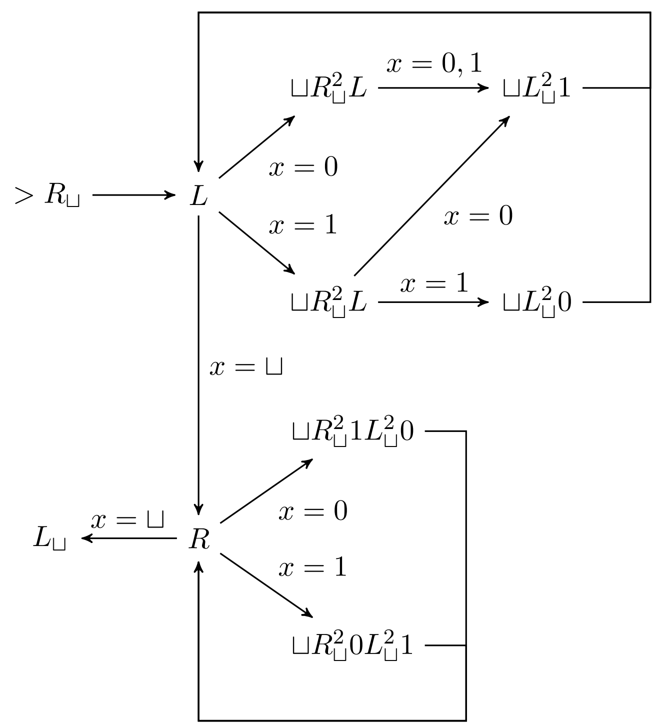
Πίνακας Περιεχομένων

1 Μηχανές Turing

1.1 Μηχανή Turing που μετασχηματίζει είσοδο $\triangleright \underline{\sqcup}\, w \sqcup$ σε $\triangleright \underline{\sqcup}\, w^R w^R \sqcup$ με $w \in \{a, b\}^*$



1.2 Μηχανή Turing που μετασχηματίζει είσοδο $\triangleright \underline{\sqcup}\, x \sqcup y \sqcup$ σε $\triangleright \underline{\sqcup}\, z \sqcup w \sqcup$ με $x, y, z, w \in \{0, 1\}^*, |x| = |y| = |z| = |w|$ και $z = x(NAND)y, W = x(AND)y$



2 Αναδρομικές και αναδρομικά απαριθμήσιμες γλώσσες

2.1 Το συμπλήρωμα μιας Turing-απαριθμήσιμης γλώσσας είναι πάντα αναδρομική γλώσσα

Σωστό

Κάθε γλώσσα που γίνεται αποδεκτή από μηχανή Turing είναι αναδρομική γλώσσα και επίσης το συμπλήρωμα μίας αναδρομικής γλώσσας είναι και αυτή αναδρομική γλώσσα λόγω κλειστότητας της πράξης της συμπλήρωσης. Άρα, μία Turing-απαριθμήσιμη γλώσσα είναι πάντα αναδρομική.

2.2 Για κάθε ντετερμινιστικό αυτόματο στοίβας μπορεί να κατασκευασθεί ισοδύναμη μηχανή απόφασης Turing

Σωστό

Κάθε αυτόματο στοίβας μπορεί να υλοποιηθεί μέσω μίας μηχανής Turing όπου η στοίβα αναπαρίσταται μέσω της ταινίας της μηχανής και για τις πράξεις της στοίβας μπορούν να χρησιμοποιηθούν τα επιπλέον κελιά της ταινίας.

3 Γραμματικές χωρίς περιορισμούς

3.1 Κατασκευάστε γραμματική χωρίς περιορισμούς για την γλώσσα $L = \{a^{n(n+1)} : n \leq 0\}$

Η γλώσσα για την οποία κατασκευάζεται η γραμματική είναι η εξής:

$$L = \{a^{n(n+1)} : n \leq 0\}$$

όμως κάνοντας τις απαραίτητες πράξεις καταλήγει στην εξής μορφή:

$$L = \{a^{n(n+1)} : n \leq 0\} \Rightarrow L = \{a^{n^2}a^n : n \leq 0\}$$

Η ζητούμενη γραμματική χωρίς περιορισμούς είναι η εξής:

$$\begin{aligned}
&G = (V, \Sigma, R, S)\\
&V = \{S, S_1, A, B, C, D, E, a\}\\
&\Sigma = \{a\}\\
&R = \{S \to AS_1E\\
&\qquad S_1 \to BS_1YZ\\
&\qquad S_1 \to e\\
&\qquad DC \to CD\\
&\qquad DE \to Ea\\
&\qquad BC \to CaB\\
&\qquad Ba \to aB\\
&\qquad Ca \to aC\\
&\qquad Aa \to aA\\
&\qquad AC \to A\\
&\qquad BE \to E\\
&\qquad AE \to e\\
&\quad\}
\end{aligned}$$

Η γλώσσα αποτελείται από 12 κανόνες οι οποίοι χωρίζουν τη διαδικασία σε 4 φάσεις. Ξεκινώντας γίνεται αρχικοποίηση τοποθετώντας τους χαρακτήρες A και E ώστε να γνωρίζουμε ποιά είναι τα όρια της έκφρασης.

Πρώτη φάση

Κατά την πρώτη φάση και εφόσον έχουν τοποθετηθεί τα διαχωριστικά A, E, ενδιάμεσα τους τοποθετούνται n τριάδες B C D με τελική μορφή την $B^n(CD)^n$ λόγω της μορφής του κανόνα 2 ενώ είναι εμφανές πως αυτή η μορφή μπορεί εύκολα να αντιστοιχηθεί στην ζητούμενη. Για την λήξη της φάσης, εφόσον δηλαδή έχουν τοποθετηθεί n τριάδες, αξιοποιείται ο τρίτος κανόνας μέσω του οποίου τοποθετείται κενός χαρακτήρας.

Δεύτερη φάση

Εφόσον πλέον έχει τοποθετηθεί ο απαραίτητος αριθμός B, C και D αξιοποιούνται ο κανόνας 4 ώστε να τοποθετηθεί ο απαραίτητος αριθμός συμβόλων α. Πιο συγκεκριμένα, κάθε φορά που αξιοποιείται ο 4ος κανόνας, δηλαδή εντοπίζεται C, D με την ανάποδη φορά (DC), γίνεται αντιστροφή των συμβόλων, διαδικασία η οποία επαναλαμβάνεται έως ότου η τελική μορφή είναι $B^nC^nD^n$, δηλαδή τα B, C, D βρίσκονται συνεχόμενα.

Τρίτη φάση

Κατά την τρίτη φάση και εφόσον τα σύμβολα B, C, D βρίσκονται ταξινομημένα, αξιοποιώντας το 5ο κανόνα, όταν ανιχνευτεί ζεύγος DE, το D αντικαθίσταται από α στα δεξιά του E, ώστε να κατασκευαστεί το δεύτερο μέρος της ζητούμενης σχέσης. Αντίστοιχα, για τα ζεύγη BC, αξιοποιώντας τον κανόνα 6, τοποθετείται ένα α ενδιάμεσα τους και τα B,C αντιστρέφονται με αποτέλεσμα, στο τέλος της διαδικασίας να έχουν συμβεί στο σύνολο n^2 εισαγωγές α, δημιουργώντας έτσι τον πρώτο όρο της ζητούμενης σχέσης.

Τέταρτη φάση

Τέλος, αξιοποιώντας τους υπόλοιπους κανόνες, αφαιρούνται όλα τα μη τερματικά σύμβολα ώστε να μείνει στο τέλος η ζητούμενη έκφραση.

Ακολουθεί παράδειγμα χρήσης της γραμματικής για την είσοδο aaaaaa∈ L

$$S \Rightarrow AS_1E \Rightarrow ABS_1CDE \Rightarrow ABBS_1CDCDE \Rightarrow ABBCDCDE \Rightarrow ABBCCDDE \Rightarrow ABBCCDEa \Rightarrow$$
$$\Rightarrow ABBCCEaa \Rightarrow ABCaBCEaa \Rightarrow ACaBaBCEaa \Rightarrow ACaBaCaBEaa \Rightarrow ACaaBCaBEaa \Rightarrow$$
$$\Rightarrow ACaaCaBaBEaa \Rightarrow ACaCaaBaBEaa \Rightarrow ACCaaaBaBEaa \Rightarrow ACCaaaaBBEaa \Rightarrow$$
$$\Rightarrow ACaaaaBBEaa \Rightarrow AaaaaBBEaa \Rightarrow AaaaaBEaa \Rightarrow AaaaaEaa \Rightarrow ... \Rightarrow AEaaaaaa \Rightarrow aaaaaa$$

4 Μη επιλυσιμότητα

4.1 Δύο μηχανές Turing M1, M2, υπάρχει έστω και μία συμβολοσειρά για την οποία τερματίζουν και οι δύο μηχανές;

Για την απάντηση της ερώτησης, μπορεί να χρησιμοποιηθεί το εξής γνωστό πρόβλημα:

$$L = \{"M'" : M' \text{ τερματίζει για κάποια είσοδο}\}$$

το οποίο πρόβλημα μπορεί πολύ εύκολα να κωδικοποιηθεί αξιοποιώντας δύο μηχανές Turing:

$$L_1 = \{"M_1""M_2" : M_1, M_2 \text{ τερματίζουν για την ίδια ακριβώς συμβολοσειρά εισόδου}\}$$

Υποθέτοντας πως το πρόβλημα είναι επιλύσιμο, θα πρέπει η γλώσσα L_1 να είναι αναδρομική, οπότε υπάρχει μηχανή Turing M_a η οποία αποφασίζει γι αυτή την γλώσσα μέσω της εξής αναγωγής:

$$"M'" \to "M'""M_2" \text{ όπου η } M_2 \text{ τερματίζει για κάθε είσοδο}$$

Η διαδικασία εκτέλεσης ξεκινά με την μηχανή M, να υπολογίζει την αναδρομική σχέση $M'""M_2"$, δίνει το αποτέλεσμα ως είσοδο στην μηχανή M_a η οποία με την σειρά της δέχεται την συμβολοσειρά $"M'""M_2"$ δεδομένου του ότι οι εκάστοτε μηχανές τερματίζουν. Την τελική απόφαση για την γλώσσα την παίρνει η μηχανή M κάτι το οποίο όμως δεν ισχύει άρα, καταλήγουμε σε άτοπο, οπότε το πρόβλημα είναι μη επιλύσιμο.

4.2 Δύο μηχανές Turing M1, M2, τερματίζει η μηχανή M2 για αυστηρά λιγότερες εισόδους από την M1

Για την απάντηση της ερώτησης, μπορεί να χρησιμοποιηθεί το εξής γνωστό πρόβλημα:

$$L = \{"M" : M \text{ τερματίζει για κάποια είσοδο}\}$$

Αντίστοιχα με την M, όλες οι "M" αναπαραστάσεις ανήκουν στην γλώσσα L και ημιαποφασίζουν για κάποια μη κενή γλώσσα, οπότε το πρόβλημα μπορεί πολύ εύκολα να κωδικοποιηθεί αξιοποιώντας δύο μηχανές Turing:

$$L_1 = \{"M_1""M_2" : M_1 \text{ τερματίζει για περισσότερες εισόδους σε σχέση με την } M_2\}$$

Υποθέτοντας πως το πρόβλημα είναι επιλύσιμο, θα πρέπει η γλώσσα L_1 να είναι αναδρομική, οπότε υπάρχει μηχανή Turing M_a η οποία αποφασίζει γι αυτή την γλώσσα μέσω της εξής αναγωγής:

$$"M" \to "M""M_3" \text{ όπου η } M_3 \text{ δεν τερματίζει (infinte loop)}$$

Η παραπάνω αναπαράσταση είναι αναδρομική καθώς η M_3 χρησιμοποιείται εύκολα σε συνέχεια της M. Αντίστοιχα με την προηγούμενη περίπτωση, η M_3 δέχεται ως είσοδο την M η οποία με την σειρά της υπολογίζει την παραπάνω αναδρομική σχέση. Στην συνέχεια, αξιοποιώντας την μηχανή M_a με είσοδο $"M"M_3"$ την οποία είσοδο αποδέχεται στην περίπτωση όπου η M τερματίζει για περισσότερες εισόδους από την M_3. Και πάλι, την τελική απόφαση για την γλώσσα την παίρνει η μηχανή M_3 κάτι το οποίο όμως δεν ισχύει άρα, καταλήγουμε σε άτοπο, οπότε το πρόβλημα είναι μη επιλύσιμο.