

ΙΟΝΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ



– Πτυχιακή Εργασία –

Μηχανές **Turing**

Γιάννης Καρζόλας

Επιβλέπων: Θεόδωρος Αδρόνικος

12 Ιουνίου 2018

Επιβλέπων

Θεόδωρος Αδρόνικος, Μόνιμος Επίκουρος Καθηγητής,
Ιόνιο Πανεπιστήμιο

Τριμελής Επιτροπή

Θεόδωρος Αδρόνικος, Μόνιμος Επίκουρος Καθηγητής,
Ιόνιο Πανεπιστήμιο

Παναγιώτης Βλάμος, Καθηγητής-Πρόεδρος Τμήματος Πληροφορικής,
Ιόνιο Πανεπιστήμιο

Μιχαήλ Στεφανιδάκης, Επίκουρος Καθηγητής,
Ιόνιο Πανεπιστήμιο

Περίληψη

Η διπλωματική αυτή εργασία γίνεται στο πλαίσιο του προπτυχιακού προγράμματος σπουδών στο Τμήμα Πληροφορικής του Ιονίου Πανεπιστημίου. Στόχος είναι η μελέτη των Μηχανών Turing στη Θεωρία Υπολογισμού. Στο πρώτο κεφάλαιο γίνεται μια εισαγωγή στη Θεωρία του Υπολογισμού και μια ιστορική αναδρομή. Στο δεύτερο κεφάλαιο εισάγονται ορισμοί για τα μαθηματικά που χρησιμοποιούνται στην Επιστήμη των Υπολογιστών. Βασικός μαθηματικός ορισμός αποτελούν οι γλώσσες και οι ιδιότητες τους. Στο τρίτο κεφάλαιο εξετάζονται μοντέλα υπολογισμού που σχετίζονται με τη διαδικασία αναγνώρισης συγκεκριμένων γλωσσών ενώ παρουσιάζεται η Θέση των Church και Turing για τον τυπικό ορισμό αυτού που σήμερα αντιλαμβανόμαστε ως "μηχανιστικό υπολογισμό" ή αλγόριθμο. Στη συνέχεια στο τέταρτο κεφάλαιο εισάγονται οι βασικοί ορισμοί για τις διάφορες χρήσεις των Μηχανών Turing. Γίνεται επίσης αναφορά σε διάφορες παραλλαγές των Μηχανών Turing, που όμως έχουν αποδειχθεί ότι δεν αυξάνουν τις δυνατότητες περιγραφής υπολογισμών, που διαθέτουν οι βασικές Μηχανές Turing. Στο τελευταίο κεφάλαιο αποδεικνύεται ότι υπάρχουν προβλήματα, που δεν επιλύονται από κανένα υπολογιστικό μοντέλο. Ενώ γίνεται αναφορά στον υπολογισμό συναρτήσεων με οποιοδήποτε πεδίο τιμών και αναπτύσσεται μια θεωρία που μελετά το τι ακριβώς υπολογίζεται και τι όχι, καθώς και ιδιότητες συναρτήσεων διαφορετικών κλάσεων, ως προς την επιλυσιμότητά τους.

Ευχαριστίες

Πρώτα απ' όλους θα ήθελα να ευχαριστήσω την οικογένεια μου που μου στάθηκε στις σπουδές μου και στην σταδιοδρομία μου τόσα χρόνια. Στην συνέχεια θα ήθελα να ευχαριστήσω τους καθηγητές μου για την καθοδήγηση τους στα βήματά μου μέσα στο Πεδίο της Πληροφορικής. Θα ήθελα να ευχαριστήσω θερμά τον κύριο Θ. Αδρόνικο για τις πολύτιμες συμβουλές του στην πτυχιακή μου εργασία. Τέλος ευχαριστώ όλους τους φίλους μου που ο καθένας συνέβαλε με ένα μικρό λιθαράκι στην δουλειά μου αυτή!

Περιεχόμενα

A'	Εισαγωγή και Ιστορικά Στοιχεία	1
A'.1	ΕΙΣΑΓΩΓΗ	1
A'.2	ΙΣΤΟΡΙΚΑ ΣΤΟΙΧΕΙΑ	2
B'	Εισαγωγή στα Μαθηματικά των Υπολογιστών	6
B'.1	ΣΥΝΟΛΑ ΚΑΙ ΣΧΕΣΕΙΣ	7
B'.2	ΣΥΝΑΡΤΗΣΕΙΣ ΚΑΙ ΣΧΕΣΕΙΣ	9
B'.3	ΜΑΘΗΜΑΤΙΚΗ ΛΟΓΙΚΗ	10
B'.4	ΣΥΜΒΟΛΟΣΕΙΡΕΣ	11
B'.4.1	Πράξεις με συμβολοσειρές	12
B'.5	ΓΛΩΣΣΕΣ	14
B'.5.1	Πράξεις με Γλώσσες	15
Γ'	Θεωρία Υπολογισμού	17
Γ'.1	ΠΑΡΑΓΩΓΗ ΚΑΙ ΑΝΑΓΝΩΡΙΣΗ ΓΛΩΣΣΩΝ ΣΤΗ ΘΕΩΡΙΑ ΥΠΟΛΟΓΙΣΜΟΥ	17
Γ'.2	ΚΛΑΣΕΙΣ ΓΛΩΣΣΩΝ	19
Γ'.2.1	Κλάσεις γλωσσών ως προς τις μηχανές για την αναγνώρισή τους	19
Γ'.2.1.1	Πεπερασμένα Αυτόματα	20
Γ'.2.1.2	Αυτόματα Στοίβας	22
Γ'.2.1.3	Μηχανές Turing	23
Γ'.2.1.4	Λογισμός-λ	27
Γ'.2.2	Κλάσεις γλωσσών ως προς το βαθμό αποδοτικότητας της διαδικασίας απόφασης	29
Γ'.3	ΚΑΤΑΤΑΞΗ ΓΛΩΣΣΩΝ	29
Γ'.4	ΘΕΣΗ ΤΩΝ CHURCH ΚΑΙ TURING	31
Γ'.5	ΝΤΕΤΕΡΜΙΝΙΣΜΟΣ ΚΑΙ ΜΗ ΝΤΕΤΕΡΜΙΝΙΣΜΟΣ	33
Δ'	Μηχανές Turing	36
Δ'.1	ΟΡΙΣΜΟΣ ΜΗΧΑΝΩΝ TURING (MT)	36

Δ'.1.1	Ένας συνθετικός συμβολισμός για MT	43
Δ'.2	ΥΠΟΛΟΓΙΣΜΟΙ ΜΕ ΜΗΧΑΝΕΣ TURING	47
Δ'.2.1	Μηχανές Turing ως μηχανές αναγνώρισης γλωσσών	47
Δ'.2.2	Υπολογισμός συναρτήσεων με Μηχανές Turing	48
Δ'.3	ΠΟΛΛΕΣ ΤΑΙΝΙΕΣ ΚΑΙ ΜΗ ΝΤΕΤΕΡΜΙΝΙΣΜΟΣ	51
Δ'.3.1	Πολλές ταινίες	51
Δ'.3.2	Μη ντετερμινισμός	54
Δ'.4	ΠΡΟΣΟΜΟΙΩΣΗ ΥΠΟΛΟΓΙΣΤΗ ΜΕ MT	58
Δ'.5	ΕΝΑΛΛΑΚΤΙΚΟΙ ΟΡΙΣΜΟΙ MT	61
Δ'.5.1	MT με ταινία άπειρου μήκους προς μία κατεύθυνση	61
Δ'.5.2	Ο ρόλος της στοίβας και της ταινίας	63
Δ'.6	ΓΕΝΙΚΗ ΜΗΧΑΝΗ TURING	65
Δ'.6.1	Κωδικοποίηση MT ως συμβολοσειρά	65
Δ'.6.2	Η Γενική μηχανή Turing	68
Ε'	Υπολογίσιμες Συναρτήσεις και μηχανές Turing	71
Ε'.1	Το Πρόβλημα του Τερματισμού	71
Ε'.2	ΥΠΟΛΟΓΙΣΙΜΕΣ ΣΥΝΑΡΤΗΣΕΙΣ	72
Ε'.3	ΘΕΩΡΙΑ ΑΝΑΔΡΟΜΙΚΩΝ ΣΥΝΑΡΤΗΣΕΩΝ	76
Ε'.3.1	Στοιχειώδης Αναδρομικές Συναρτήσεις	77
Ε'.3.2	μ - αναδρομικές συναρτήσεις	79
Ε'.4	ΘΕΩΡΗΜΑ ΑΝΑΔΡΟΜΗΣ/ΣΤΑΘΕΡΟΥ ΣΗΜΕΙΟΥ	80
Γ'	Επίλογος	85
Παράρτημα Α'		87

Κατάλογος Σχημάτων

Γ.1 Ένα απλό αυτόματο	21
Γ.2 Ένα απλό αυτόματο στοίβας για τη γλώσσα $\{a^n b^n : n \geq 0\}$	22
Γ.3 Απεικόνιση της λειτουργίας μιας μηχανής Turing	24
Γ.4 μια μηχανή Turing για τη γλώσσα $\{a^n b^n c^n : n \geq 0\}$	25
Γ.5 Ιεραρχία γλωσσών κατά Chomsky	30
Γ.6 μη ντετερμινιστικό αυτόματο με δύο μεταβάσεις με 0 από την q_0	35
Δ.7 Απεικόνιση της λειτουργίας μιας μηχανής Turing	38
Δ.8 Γραφικός συμβολισμός για τη MT του Παραδείγματος	42
Δ.9 MT $S \rightarrow$ για τη μετατόπισησυμβολοσειράς μία θέση δεξιά.	47
Δ.10 MT C για την αντιγραφή συμβολοσειράς.	50
Δ.11 MT για τον υπολογισμό του «επόμενου» αριθμού.	50
Δ.12 Ντετερμινιστική MT M' για τη μη ντετερμινιστική MT M	58
Δ.13 Προσομοίωση MT δύο κατευθύνσεων από ταινίες μιας κατεύθυνσης	62
Δ.14 Προσομοίωση ΑΣ από MT	64
Δ.15 Προσομοίωση της ταινίας μιας MT από δύο στοίβες	65
Ε.16 Συνάρτηση με σταθερό σημείο	84
΄.17 JFLAP menu	87
΄.18 Παράθυρο JFLAP για Πεπερασμένο Αυτόματο	88
΄.19 Καθορισμός αρχικής κατάστασης στο JFLAP για Πεπερασμένο Αυτόματο	88
΄.20 Καθορισμός τελικής κατάστασης στο JFLAP για Πεπερασμένο Αυτόματο	89
΄.21 Δημιουργία μεταβάσης στο JFLAP για Πεπερασμένο Αυτόματο	89
΄.22 Δημιουργία μεταβάσης στο JFLAP για Πεπερασμένο Αυτόματο	90
΄.23 Δημιουργία μεταβάσης στο JFLAP για Πεπερασμένο Αυτόματο	90
΄.24 Ένα απλό Πεπερασμένο Αυτόματο στο JFLAP	91
΄.25 Ένα απλό Πεπερασμένο Αυτόματο στο JFLAP	91
΄.26 Έλεγχος Συμβολοσειρών σε ένα απλό Πεπερασμένο Αυτόματο στο JFLAP	92
΄.27 Αποτέλεσμα ελέγχου Συμβολοσειρών σε ένα απλό Πεπερασμένο Αυτόματο στο JFLAP	92

'.28 Σχεδιασμός ενός απλού αυτόματου στοιβάς στο JFLAP	92
'.29 Τελικός Σχεδιασμός ενός απλού αυτόματου στοιβάς στο JFLAP	93
'.30 Έλεγχος συμβολοσειρών για ένα απλό αυτόματο στοιβάς στο JFLAP	93
'.31 Σχεδιασμός μιας μηχανής Turing στο JFLAP	94
'.32 Έλεγχος Συμβολοσειρών μιας μηχανής Turing στο JFLAP	94
'.33 Έλεγχος Μη Ντετερμινιστικών καταστάσεων ενός αυτομάτου στο JFLAP	95
'.34 Μεταβάσεις ενός αυτομάτου στο JFLAP	95

Κεφάλαιο Α΄

Εισαγωγή και Ιστορικά Στοιχεία

Α΄.1 Εισαγωγή

Η Θεωρία Υπολογισμού είναι ο κλάδος της θεωρητικής πληροφορικής που πραγματεύεται το εάν και το κατά πόσο αποδοτικά είναι δυνατόν να λυθεί κάποιο πρόβλημα με χρήση κάποιου αλγορίθμου σ' ένα υπολογιστικό μοντέλο, μια αφηρημένη μαθηματική έννοια ορισμένη σε αυστηρούς κανόνες. Θα μπορούσε κάποιος να χωρίσει την Θεωρία Υπολογισμού σε τρεις μεγάλες ενότητες που προσπαθούν να απαντήσουν σε τρία μεγάλα ερωτήματα.

Στην ερώτηση: "Τι είναι υπολογιστής;" καλείται η Θεωρία Αυτομάτων να ασχοληθεί με τις ιδιότητες μαθηματικών μοντέλων υπολογισμού για να δώσει απάντηση. Ένα αυτόματο είναι μια στοιχειώδης υπολογιστική μηχανή η οποία έχει τη δυνατότητα να αναγνωρίζει τυπικές γλώσσες. Υπάρχουν πολλά διαφορετικά είδη αυτομάτων, με διαφορετικές υπολογιστικές δυνατότητες. Τα αυτόματα είναι πολύ χρήσιμα σε διάφορες περιοχές της Πληροφορικής (μεταγλωτιστές, hardware, γλώσσες προγραμματισμού κτλ.).

Στην ερώτηση: "Τι είναι και τι δεν είναι υπολογίσιμο;" καλείται να δώσει απάντηση η Θεωρία Υπολογισιμότητας, η οποία εξετάζει ποιά προβλήματα μπορούν να επιλυθούν από υπολογιστή και ποιά όχι. Η Θεωρία Υπολογισιμότητας μας βοηθάει να καταλάβουμε ποιά είναι τα όρια της επιστήμης μας. Στις περισσότερες επιστήμες ερευνούμε τι μπορούμε να κάνουμε με τους υπολογιστές, και όχι τι δεν μπορούμε. Η Θεωρία Υπολογισιμότητας μας προσφέρει εκείνα τα μαθηματικά εργαλεία για να αποδείξουμε ότι ένα πρόβλημα δεν είναι επιλύσιμο, ενώ έχει πολύ στενές σχέσεις με τη μαθηματική Λογική, όπως μπορεί να υποψιαστεί κάποιος. Στην ερώτηση: "Τι μπορεί να υπολογιστεί γρήγορα και τι όχι;" καλείται να απαντήσει η Θεωρία Υπολογιστικής Πολυπλοκότητας. Στη Θεωρία Πολυπλοκότητας το ζητούμενο είναι να ελεγχθούν τα προβλήματα ως "εύκολα" ή "δύσκολα". Η Θεωρία Πολυπλοκότητας συγγενεύει στενά με τη Θεωρία Υπολογισιμότητας έτσι φαίνεται αναγκαίο να σημειώσουμε ότι το ζητούμενο για την Θεωρία Υπολογισιμότητας είναι ο έλεγχος των προβλημάτων ως "επιλύσιμων" ή "μη επιλύσιμων". Να τονίσουμε ότι υπάρχουν προβλήματα για τα οποία δεν έχει καταφέρει κανείς να βρεί αποτελεσματικό αλγόριθμο επίλυσης.

Συνοψίζοντας τις προαναφερθέντες ερωτήσεις θα μπορούσαμε να θέσουμε μια ερώτηση που η Θεωρία Υπολογισμού θα πρέπει να δύναται να υπολογίσει: “Ποιές είναι οι δυνατότητες και ποιοί οι περιορισμοί της έννοιας του υπολογιστή;”

Οι επιστήμονες αξιοποιούν διάφορες διατυπώσεις υπολογιστικών μοντέλων, αλλά συνηθέστερη είναι η μηχανή Τούρινγκ. μπορεί κανείς να θεωρήσει τη μηχανή Τούρινγκ σαν έναν προσωπικό υπολογιστή με άπειρη μνήμη, αν και η μνήμη αυτή είναι προσβάσιμη μόνο κατά μικρά διακριτά τμήματα. Οι επιστήμονες υπολογιστών μελετούν τη μηχανή Turing γιατί έχει απλή μαθηματική διατύπωση και μπορούν να την αναλύσουν, να την χρησιμοποιήσουν σε αποδείξεις, ενώ κατά πολλούς αντιπροσωπεύει το ισχυρότερο εφικτό μοντέλο υπολογισμού. Αυτό σημαίνει ότι ένα πρόβλημα μπορεί να λυθεί αλγοριθμικά αν και μόνο αν υπάρχει μηχανή Τούρινγκ που το επιλύει. Αν και η άπειρη μνήμη μπορεί να θεωρηθεί ανέφικτη, στην πραγματικότητα κάθε πρόβλημα που λύνει η μηχανή Τούρινγκ χρειάζεται πάντα πεπερασμένη μνήμη, άρα κάθε πρόβλημα που μπορεί να λυθεί από μια μηχανή Τούρινγκ θα μπορούσε να λυθεί από έναν προσωπικό υπολογιστή με επαρκή μνήμη.

Α΄.2 Ιστορικά Στοιχεία

Η μελέτη μιας επιστημονικής/ μαθηματικής (και όχι μόνον) περιοχής, βοηθά στην κατανόηση της ιστορίας της, ισχύει όμως και το αντίστροφο: η ιστορία μιας περιοχής φωτίζει και την ίδια. Η ιστορία της θεωρίας του υπολογισμού όμως δεν έχει γραφεί ακόμα. Η σχετική διεθνής βιβλιογραφία σε αυτό το θέμα είναι ακόμα αποσπασματική, ίσως ακόμα και ρηχή. Δεν έχουμε λοιπόν να προσφέρουμε μια καλή σύνοψη του τί συνέβη και φτάσαμε να έχουμε μια θεωρία (και μια τεχνολογία, πια) του υπολογισμού. Αντ’ αυτού θα απαριθμήσουμε μια σειρά επεισοδίων σχετικά με αυτή την εξέλιξη, μέσω μιας σειράς από εξέχουσες προσωπικότητες. Ήδη από την αρχαιότητα είχε επισημανθεί ότι υπάρχουν λογικοί κανόνες, και ότι βάσει αυτών μπορεί και πρέπει να ασκούνται τα μαθηματικά. Είχε ήδη επισημανθεί η πρακτική και όχι μόνον, αξία των κατασκευαστικών μεθόδων, τις οποίες ο αρχαίος κόσμος σεβόταν πολύ και διαρκώς επεδίωκε.

Ο Αριστοτέλης, ο Σταγειρίτης (384–332)π.Χ., μεταξύ αναρίθμητων συνεισφορών του, έγραψε και ένα εγχειρίδιο για την λογική. Κινούταν στο επίπεδο του “προτασιακού λογισμού”, και όπως ο ίδιος έλεγε: *περί της συμπερασματικής λογικής δεν γνώριζε καμμία προηγούμενη πηγή* (αν και κάποιοι είχαν, προηγουμένως ασχοληθεί με τους κανόνες όχι της σκέψης αλλά της γλώσσας, και είχαν μιλήσει για το “συντακτικό” της). Η “Αριστοτελική Λογική” έμελε να παραμείνει σε εκείνο το αρχικό επίπεδο για πάνω από 2000 χρόνια.

Γύρω στα 300 π.Χ. –200 μ.Χ., Έλληνες και όχι μόνο, η σχολή των Στωικών είχε την δική της εκδοχή για την “λογική”, η οποία έδινε θεμελιώδη έμφαση στη σχέση “συνεπάγεται” ($p \rightarrow q$, αν και τότε δεν χρησιμοποιούσαν αυτόν τον συμβολισμό). Η εκδοχή τους νόμιζαν ότι διέφερε από εκείνη του Αριστοτέλη, αυτό όμως συνέβαινε μόνον επιφανειακά.

Ο Ευκλείδης, ο Αλεξανδρινός, στην Εποχή του Πτολεμαίου (323 – 283) π.Χ., ήταν έφορος της

βιβλιοθήκης της Αλεξανδρείας, και κατά γενική ομολογία ο κρισιμότερος μαθηματικός όλων των εποχών. με τα "Στοιχεία" του όχι μόνο θεμελίωσε την γεωμετρία, αλλά και την ίδια την μαθηματική μέθοδο: σαφείς *ορισμοί* και ακριβείς *αποδείξεις*. Τα "Στοιχεία" περιλαμβάνουν και τον περίφημο "αλγόριθμο του Ευκλείδη" (για την εύρεση του μέγιστου κοινού διαιρέτη), έναν από τους πρώτους ρητά διατυπωμένους αλγορίθμους (μαζί με την απόδειξη ότι είναι σωστός!)- αν και ούτε εκείνη την εποχή τον έβλεπαν ως αλγόριθμο, ούτε και ποτέ (προ 20^{ου} αιώνα) τον ονόμαζαν έτσι.

Κατά την μεσαιωνική Περίοδο, αν και οι φιλόσοφοι και οι μαθηματικοί διαπίστωναν προβλήματα και ελλείψεις στα διαθέσιμα εργαλεία της λογικής, δεν πέτυχαν καμμία ουσιαστική συνεισφορά. Έτσι η κατάσταση στο χώρο της λογικής και των κατασκευαστικών μαθηματικών παρέμεινε στάσιμη όσο αναφορά την κατανόηση των σχετικών εννοιών για εξαιρετικά μακρύ χρονικό διάστημα.

Κατά τον 19^ο αιώνα η μελέτη της λογικής επανήλθε με μαθηματικά πια εργαλεία, και έγιναν νέα καίρια, ιστορικού επιπέδου βήματα. Η αξιωματική και αποδεικτική μέθοδος του Ευκλείδη επανισχυροποιήθηκε και έγινε αποδεκτό ότι η απόδειξη όχι μόνον είναι μια "μηχανική" διαδικασία, αλλά καλύτερα που είναι τέτοια, διότι μόνον έτσι εξασφαλίζεται το αλάθητο και η βεβαιότητα για την ορθότητα της συλλογιστικής διαδικασίας (και άρα και των όποιων συμπερασμάτων).

Το 1847 ο Βρετανός George Boole (1815 – 1864), με το βιβλίο του *The laws of thought* επανέφερε το θέμα της λογικής, ακριβέστερα του προτασιακού λογικισμού, έδειξε την ενότητα των απόψεων του Αριστοτέλη και των Στωικών, και έδειξε τις δυνατότητες αλγεβρικού χειρισμού της λογικής. Η λογική δεν ήταν πια μόνον όργανο των μαθηματικών, αλλά και αντικείμενο των μαθηματικών.

Το 1879 μια φυσιογνωμία κλειδί, ο Γερμανός Gottlob Frege (1848 – 1925) με το βιβλίο του *"Εννοιολογία"* επεξέτεινε ρητά και συνειδητά- επί τέλους- την προτασιακή λογική, περιγράφοντας τον *κατηγορηματικό λογισμό*, εισάγοντας τους λογικούς ποσοδείκτες ($\forall$ και $\exists$), και δείχνοντας το πώς οι αποδείξεις μπορούν να γίνουν μέσω συμβολικών κατ' ουσίαν χειρισμών. Ο κατηγορηματικός λογισμός είναι σχεδόν αποκλειστικά το σύστημα λογικής που χρησιμοποιούμε τώρα στα μαθηματικά. Το 1893 (I τόμος) και 1903 (II τόμος) με το βιβλίο του *Βασικοί νόμοι της αριθμητικής* προσπάθησε να θεμελιώσει τους φυσικούς αριθμούς επί της λογικής. μια κρίσιμη λογική αδυναμία στο II τόμο, άναψε την φωτιά της αναζήτησης επαρκώς ευσταθών θεμελίων για τα μαθηματικά, και οδήγησε στην συστημική ανάλυση της μεθόδου των "αποδείξεων" και στη συνέχεια των υπολογισμών".

Το 1889 ο Ιταλός Giuseppe Peano (1858 – 1932), δημοσίευσε ένα συλλεκτικό έργο (με εργασίες από το 1860) επί της "αξιωματικής θεμελίωσης" των φυσικών αριθμών, για να δείξει ότι λίγες αρχικές παραδοχές αρκούν για να εξαχθεί αποδεικτικά όλη η θεωρία των αριθμών. Τα σχετικά αξιώματα καλούνται ακόμα "αξιώματα του Peano".

Επιφανείς μαθηματικοί, όπως οι Russel και Hilbert, καταθέτουν όλη τους την εμπιστοσύνη στις "πεπερασμένες" αξιωματικές και αποδεικτικές μεθόδους, και κηρύσσουν ως πρόγραμμα

των μαθηματικών την εύρεση μιας μεθόδου, για την διάγνωση της αλήθειας ή όχι, οποιουδήποτε μαθηματικού ισχυρισμού. Διαπιστώθηκε γρήγορα, όμως, ότι η ορθή χρήση των λογικών- αποδεικτικών εργαλείων όχι μόνον δεν είναι προφανής, αλλά ότι ακόμα και αυτά υπόκεινται σε σοβαρούς περιορισμούς.

Το 1900 ο Βρετανός Bertrand Russel (1872 – 1970), επεσήμανε το “παράδοξο του Russel”, μια καίρια αντίφαση στο υπό δημοσίευση έργο του Frege. Ο ίδιος εργάστηκε επί μακρά σειρά ετών στην “αξιωματική θεμελίωση” των μαθηματικών, εμποδώνοντας ότι ο λογισμός του Frege ήταν όντως, με σωστή χρήση, το κατάλληλο εργαλείο.

Ο Γερμανός David Hilbert (1862 – 1943) ήταν μια εμβληματική φυσιογνωμία των μαθηματικών του 20^{ου} αιώνα. Όχι μόνον έκανε πολλά και βαθιά μαθηματικά, αλλά έκανε τα ίδια μαθηματικά του 20^{ου} αιώνα, προσδιορίζοντας τόσο τα μείζονα προβλήματά τους, αλλά και τις μεθόδους τους. Ο ίδιος βάθυνε και στερέωσε την χρήση της ευκλείδιας αξιωματικής μεθόδου. Το σημείο κλειδί όσον αφορά στην ιστορία του υπολογισμού είναι ότι τις 10ετίες 1900 – 1920 υποστήριζε θερμά την “πεπερασμένη” εκδοχή των μαθηματικών εργαλείων, ότι δηλαδή με πεπερασμένες και διακριτές λογικές μεθόδους μπορούμε και θα είμαστε σε θέση να κρίνουμε την αλήθεια ή το ψεύδος κάθε δυνατής μαθηματικής πρότασης. Ως προς αυτό, διαφεύστηκε από έναν μαθητή του, τον K. Godel.

Ο Αυστριακός Kurt Godel (1906 – 1978), εργάστηκε στη λογική και έδωσε τα θεωρήματα της “πληρότητας” (του κατηγορηματικού λογισμού a-la Frege), και της “μη-πληρότητας” (της αριθμητικής, στην διδακτορική του διατριβή το 1931). Το 2^ο θεωρείται το πιο “shocking” θεώρημα του 20^{ου} αιώνα, αν όχι όλων των μαθηματικών μέχρι τώρα. Το θεώρημα αυτό λέει ότι η αριθμοθεωρία περιέχει προτάσεις που είναι μεν αληθινές, αλλά δεν μπορούμε να τις αποδείξουμε (με κανένα “εύλογο” σύστημα αξιωμάτων). Έδειξε λοιπόν με αυτόν τον τρόπο ότι η αξιωματική- αποδεικτική μέθοδος έχει κάποιου είδους όρια, και επομένως έθετε το ζωτικό και επείγον ζήτημα, του ποιά είναι αυτά.

Τέλος, προκειμένου να διαπιστωθεί με τον πιο σαφή και τελεσίδικο τρόπο τί είδους κατασκευαστικές διαδικασίες έχουμε στην διάθεσή μας, και τί είδους χειρισμούς είμαστε σε θέση να κάνουμε “μηχανικά”, ο Turing καταλήγει στις ομώνυμες μηχανές. Ακολουθούν και άλλοι, την ίδια εποχή οι Kleene, Church, Markov, Neuman, οι οποίοι διευκρινίζουν το θέμα των “μηχανών”, της “υπολογισιμότητας”, των “προγραμμάτων” κοκ., ανοίγουν εφεξής το πεδίο της κατασκευής υπολογιστικών συσκευών.

Ο Βρετανός Alan Turing (1912 – 1954) θεωρείται ο “πατέρας” της θεωρίας υπολογισμού. μετά το θεώρημα μη-πληρότητας του Godel ένωσε το εξής ερώτημα: “ΟΚ - οι αποδείξεις έχουν όρια, αλλά μήπως άλλοι συμβολικοί χειρισμοί είναι σε θέση να τα καταφέρουν καλύτερα”. Για την απάντησή του έπρεπε να εξετάσει τον ισχυρότερο κατά το δυνατόν τύπο μηχανής ικανής για συμβολικούς χειρισμούς- την λεγόμενη πλέον μηχανή Turing (1936). Ο Turing εξήγησε γιατί αυτός ο τύπος μηχανής που εισηγήθηκε είναι ο ισχυρότερος στον οποίο μπορούμε να ελπίζουμε, αλλά ότι ακόμα και αυτός δεν είναι σε θέση να επιλύσει όλα τα προβλήματα που θα θέλαμε να λύσουμε “αλγοριθμικά”. Κατά την διάρκεια του πολέμου

εργάστηκε σε ένα είδος υπολογιστή, εξειδικευμένου στην επίλυση κρυπταναλυτικών προβλημάτων. Δεν ήταν ακριβώς ένας υπολογιστής όπως τον θέλουμε τώρα, αλλά ήταν αρκετός για να δείξει ότι ο υπολογισμός ήταν -μεταξύ άλλων- και ένα όπλο.

Ένας εξαιρετικά ευφυής μαθηματικός, ο οποίος δεν άφησε κλάδο στον οποίο να μην προσφέρει κατά καίριο τρόπο: ανάλυση, άλγεβρα, κβαντική φυσική, θεωρία παιγνίων, θεωρία υπολογισμού κα., ήταν ο Ούγγρος, και τελικά πολίτης των ΗΠΑ, John von Neuman (1903 – 1957). Αμέσως μετά τον 2^ο παγκόσμιο πόλεμο, το 1945, ενεπλάκη στην ομάδα των ΗΠΑ για την κατασκευή του πρώτου “γνήσιου” υπολογιστή. Εκεί πρότεινε την υπολογιστική αρχιτεκτονική που φέρει το όνομά του, δηλαδή, της συσκευής με μία μνήμη, τόσο για το “πρόγραμμα” όσο και για τα “δεδομένα”. Εν πολλοίς, αυτή την αρχιτεκτονική χρησιμοποιούμε ακόμα.

μια προσωπικότητα όχι των μαθηματικών αλλά και της διεθνούς σκηνής, ο εν ζωή Αμερικανός Noam Chomsky (1928), έγραψε πολλά για τις “τυπικές γλώσσες” την 10–ετία ’50 – ’60. Η μορφή αυτού του μαθήματος οφείλεται σε δικές τους συνεισφορές, λόγου χάρη στη χαρακτηριστική εργασία *“Three models for the description of a language”*, 1956.

Στη δεκαετία του 1960 τα ηνία της θεωρίας υπολογισμού ανέλαβε ένας κλάδος αυτής, η “θεωρία πολυπλοκότητας”, η οποία δεν ενδιαφέρεται μόνον για το ποιά προβλήματα λύνονται υπολογιστικά, αλλά κυρίως για το πόσο γρήγορα ή οικονομικά αυτό επιτυγχάνεται.

Κεφάλαιο Β'

Εισαγωγή στα Μαθηματικά των Υπολογιστών

Θεμελιώδη σημασία σε κάθε υπολογισμό έχει η αναπαράσταση της πληροφορίας που επεξεργαζόμαστε. Στην καθημερινή ζωή, η πληροφορία που επικοινωνούμε και επεξεργαζόμαστε διατυπώνεται σε μία φυσική γλώσσα, όπως είναι τα Ελληνικά. Κάθε τέτοια φυσική γλώσσα στηρίζεται σ' ένα πεπερασμένο σύνολο βασικών φωνολογικών ενοτήτων για το σχηματισμό των λέξεων και προτάσεων. Οι λέξεις ορίζονται ως πεπερασμένες ακολουθίες τέτοιων φωνολογικών ενοτήτων (συλλαβές) και οι προτάσεις σχηματίζονται ως πεπερασμένες ακολουθίες λέξεων. Στο γραπτό λόγο, κάθε φυσική γλώσσα χρησιμοποιεί ένα πεπερασμένο σύνολο συμβόλων και λέξεις γράφονται ως πεπερασμένες ακολουθίες αυτών.

Αντίστοιχες προσεγγίσεις έχουν αναπτυχθεί για την αναπαράσταση στοιχείων άλλων συνόλων. Για παράδειγμα, ένας φυσικός αριθμός αναπαριστάται ως πεπερασμένη ακολουθία ψηφίων του δεκαδικού συστήματος αρίθμησης. Οι υπολογισμοί, όπως και οι φυσικές γλώσσες, επεξεργάζονται πληροφορίες, που μπορεί να περιλαμβάνουν ακέραιους, γραφήματα, προγράμματα ή άλλες οντότητες, που σε όλες τις περιπτώσεις θα πρέπει να αναπαριστώνται σε μία γενική μορφή. Στην πραγματικότητα οι υπολογισμοί τελικά επεξεργάζονται συμβολοσειρές, που αναπαριστούν τα αντικείμενα που περιλαμβάνει η πληροφορία.

Στη Θεωρία Υπολογισμού, κάθε συμβολοσειρά εκφράζει έναν υπολογισμό, δηλαδή την εκτέλεση μιας μηχανής, όπως η μηχανή του Turing. Αυτό ισχύει, επειδή κάθε τυπικά ορισμένη μηχανή ή μοντέλο υπολογισμού εκτελεί υπολογισμούς, που εξαρτώνται από το πρόγραμμα, που φορτώνεται και από την είσοδο που δέχεται μέσω μιας "ταινίας εισόδου". Στα περισσότερα συστήματα αρκεί να εξετάσουμε υπολογισμούς με πεπερασμένες συμβολοσειρές. Άπειρες σε μήκος συμβολοσειρές χρησιμοποιούνται στη μελέτη υπολογισμών αλληλεπιδραστικών συστημάτων (reactive systems), όπως είναι τα λειτουργικά συστήματα, που υποτίθεται ότι εκτελούνται- τουλάχιστο σε αντιληπτικό και μαθηματικό επίπεδο- για πάντα. Τέτοιου είδους συμπεριφορές τελικά δε μπορούν να μοντελοποιηθούν σ' ένα υψηλό επίπεδο χρησιμοποιώντας αποκλειστικά συμβολοσειρές πεπερασμένου μήκους.

Η μελέτη των συμβολοσειρών και των γλωσσών είναι σημαντική για τη θεωρία της Επιστήμης των Υπολογιστών. μετρώντας τους χρόνους εκτέλεσης ή τις απαιτήσεις αποθήκευσης των προγραμμάτων ως προς τα μήκη των συμβολοσειρών εισόδου τους, μπορούμε να χαρακτηρίσουμε την πολυπλοκότητά τους σε υπολογιστικό χρόνο και χώρο. μελετώντας πρότυπα που ακολουθούνται από σύνολα συμβολοσειρών, μπορούμε να ταξινομήσουμε το βαθμό στον οποίο επιλύονται τα διάφορα προβλήματα.

Έτσι στο πρώτο βασικό κεφάλαιο της παρούσας διπλωματικής εργασίας επικεντρωνόμαστε στον τυπικό ορισμό των γλωσσών που χρησιμοποιούμε στην Επιστήμη των Υπολογιστών και των ιδιοτήτων τους.

Οι εξής ορολογία και έννοιες θα είναι τα πρώτα θεμέλια της θεωρίας μας.

Β'.1 Σύνολα και Σχέσεις

Ορισμός Β'.1.1. *Σύνολο είναι μια συλλογή καλώς καθορισμένων και διακριτών αντικειμένων, που αποτελούν τα μέλη (ή στοιχεία) του.*

Η αναπαράσταση των συνόλων γίνεται απαριθμώντας τα μέλη τους μέσα σε άγκιστρα. Σημειώνουμε ότι αν το z είναι μέλος του συνόλου L , τότε γράφουμε $z \in L$, ενώ αν δεν είναι γράφουμε $z \notin L$.

Συχνά δε γίνεται να ορίσουμε ένα σύνολο με παράθεση των στοιχείων του, ακόμη και όταν αυτά είναι πεπερασμένα. Παράδειγμα αποτελεί ένα σύνολο με άπειρα στοιχεία, αυτό των φυσικών αριθμών $N = \{0, 1, 2, \dots\}$. Έτσι εναλλακτικά τα σύνολα αναπαριστώνται με αναφορά σε άλλα σύνολα και στις ιδιότητες που έχουν ή όχι τα μέλη $B = \{x : x \in A \text{ και } x \text{ έχει την ιδιότητα } P\}$. Για παράδειγμα το σύνολο των φυσικών αριθμών μπορεί να αναπαρασταθεί ως:

$$N = \{x : \text{το } x \text{ είναι φυσικός αριθμός}\} \quad (\text{B'.1})$$

Τονίζουμε ότι ένα σύνολο περιέχει μόνο διακριτά αντικείμενα, δεν επιτρέπονται δηλαδή επαναλήψεις των στοιχείων του. Επίσης η διάταξη των μελών δεν παίζει κανένα ρόλο. Ο ορισμός του συνόλου δεν αποκλείει την περίπτωση ύπαρξης ενός συνόλου χωρίς καθόλου στοιχεία, ένα τέτοιο σύνολο αποτελεί το *κενό σύνολο* $\{\}$ ή $\emptyset$.

Ορισμός Β'.1.2. *Ένα σύνολο A είναι υποσύνολο ενός συνόλου B αν κάθε στοιχείο του A είναι και στοιχείο του B .*

$$\{G, B\} \subseteq \{R, B, G\} \quad (\text{B'.2})$$

Φυσικά το κενό σύνολο θα είναι υποσύνολο κάθε συνόλου, ενώ κάθε σύνολο θα είναι υποσύνολο του εαυτού του. Το σύνολο A είναι κανονικό (γνήσιο) υποσύνολο του B , ($A \subset B$), αν είναι υποσύνολο του B αλλά όχι ίσο με αυτό.

Δύο σύνολα A και B είναι ίσα, αν και μόνο αν ($A \subseteq B$) και ($B \subseteq A$).

Ορισμός Β'.1.3. Το συμπλήρωμα A' ενός συνόλου A αποτελείται από όλα εκείνα τα στοιχεία που δεν ανήκουν στο A .

$$A' = \{x \in U | x \notin A\} \quad (\text{B'.3})$$

Ορισμός Β'.1.4. Για οποιαδήποτε δύο σύνολα A, B μπορούμε να σχηματίσουμε την ένωση τους $A \cup B$ ή την τομή τους $A \cap B$, με τις σχέσεις

$$A \cup B = \{x | x \in A \text{ ή } x \in B\} \quad (\text{B'.4})$$

$$A \cap B = \{x | x \in A \text{ και } x \in B\} \quad (\text{B'.5})$$

$$(\text{B'.6})$$

Είναι δυνατό να σχηματίσουμε την ένωση και την τομή περισσότερων από δύο συνόλων. Για $A_1, A_2, \dots$ σύνολα τότε η ένωση και η τομή των συνόλων αυτών είναι αντίστοιχα:

$$\bigcup_i A_i = \{x | x \in A_i \text{ για τουλάχιστον ένα } i \geq 1\}$$

$$\bigcap_{i=1}^N A_i = \{x | x \in A_i \text{ για κάθε } 1 \leq i \leq N\}$$

Ορισμός Β'.1.5. Η διαφορά δύο συνόλων A, B είναι το σύνολο των στοιχείων του A που δεν ανήκουν στο B

$$A - B = A \cap B' = \{x | x \in A \text{ και } x \notin B\} \quad (\text{B'.7})$$

Φυσικά για τον πλήρη ορισμό των συνόλων απαραίτητο είναι να σημειώσουμε τους νόμους που διέπουν τα συνόλα.

Αντιμεταθετικότητα	$A \cup B = B \cup A$
	$A \cap B = B \cap A$
Προσεταιριστικότητα	$(A \cup B) \cup C = A \cup (B \cup C)$
	$(A \cap B) \cap C = A \cap (B \cap C)$
Επιμεριστικότητα	$A \cup (B \cap C) = (A \cup B) \cap C$
	$A \cap (B \cup C) = (A \cap B) \cup C$
Νόμοι De Morgan	$(A \cup B)' = A' \cap B'$
	$(A \cap B)' = A' \cup B'$
Συμπλήρωμα και κενό σύνολο	$(A')' = A, A \cap A' = \emptyset, A \cup A' = U$
	$A \cup \emptyset = A, A \cap \emptyset = \emptyset$

Ορισμός Β'.1.6. Για κάθε σύνολο A , το σύνολο όλων των δυνατών υποσυνόλων του A συμβολίζεται με 2^A ή $\mathcal{P}(A)$ και ονομάζεται δυναμοσύνολο του A .

Ορισμός Β'.1.7. Το καρτεσιανό γινόμενο δύο συνόλων A, B αποτελείται από όλα τα διατεταγμένα ζεύγη της μορφής (a, b) , όπου $a \in A$ και $b \in B$, και συμβολίζεται με $A \times B$. Διατεταγμένο ζεύγος σημαίνει ότι το (a, b) είναι διαφορετικό από το (b, a) .

Β'.2 Συναρτήσεις και Σχέσεις

μαζί με τα σύνολα οι συναρτήσεις αποτελούν μερικά από τα σπουδαιότερα μαθηματικά αντικείμενα.

Ορισμός Β'.2.1. μια εμπησυνάρτηση f από το σύνολο A στο σύνολο B είναι ένας κανόνας που αντιστοιχεί σε κάθε στοιχείο του συνόλου A ένα μοναδικό στοιχείο του συνόλου B

$$f : A \rightarrow B \quad (\text{B'.8})$$

Το σύνολο A λέγεται πεδίο ορισμού της f . Αν $a \in A$, τότε το $f(a)$ είναι το μοναδικό στοιχείο του B που αντιστοιχεί στο a και λέγεται εικόνα του a .

Για μια συνάρτηση από το A στο B , με τα στοιχεία x να ανήκουν στο A και τα στοιχεία y να ανήκουν στο B , το πεδίο τιμών της συνάρτησης f είναι το σύνολο των εικόνων της f .

Ορισμός Β'.2.2. μια συνάρτηση $f : A \rightarrow B$ είναι ένα προς ένα ή $1-1$, αν για κάθε δύο διαφορετικά $a, a' \in A$ ισχύει $f(a) \neq f(a')$.

Ορισμός Β'.2.3. μια συνάρτηση $f : A \rightarrow B$ είναι επί, αν για κάθε στοιχείο του B είναι η εικόνα κάποιου στοιχείου του A .

Ορισμός Β'.2.4. μια συνάρτηση $f : A \rightarrow B$ θα λέγεται αντιστοιχία, αν είναι ένα προς ένα και επί.

Ορισμός Β'.2.5. Η αντίστροφη συνάρτηση της f είναι η f^{-1} και έχει τις εξής ιδιότητες

$$\text{για κάθε } x \in A, f^{-1}(f(x)) = x$$

$$\text{για κάθε } y \in B, f(f^{-1}(y)) = y$$

Ορισμός Β'.2.6. Αν $f : A \rightarrow B$ και $g : B \rightarrow C$ τότε $f(x) \in B$ και τότε έχουμε τη σύνθεση δύο συναρτήσεων $g(f(x))$ ή $g \circ f$. Το πεδίο τιμών της f είναι υποσύνολο του B και η $g(f(x))$ παίρνει τιμές στο σύνολο C . Η σύνθεση των συναρτήσεων είναι η $h : A \rightarrow C$.

Ξεκάθαρα οι συναρτήσεις συσχετίζουν στοιχεία ενός συνόλου με στοιχεία κάποιου άλλου συνόλου. μπορούμε να αντιστοιχίσουμε σε κάθε στοιχείο του A πολλά ή κανένα στοιχείο του B , και σε κάθε στοιχείο του B πολλά ή κανένα στοιχεία του A . μια τέτοια αντιστοίχιση ίσως να μην είναι μια συνάρτηση. Αν το ζεύγος (x, y) ανήκει στο υποσύνολο $A \times B$, τότε το x "σχετίζεται" με το y . Ένα τέτοιο υποσύνολο R αποτελεί τη σχέση από το A στο B και για ένα ζεύγος (x, y) μπορούμε να γράφουμε xRy αντί για $(x, y) \in R$.

Ορίζουμε στο σημείο αυτό μερικά βασικά είδη σχέσεων. Έστω μία σχέση $R \subseteq A \times A$

- **Ανακλαστική.** Η R λέγεται ανακλαστική, αν κάθε στοιχείο σχετίζεται με τον εαυτό του, δηλαδή για κάθε $x \in A, xRx$.

- *Συμμετρική.* Η R είναι συμμετρική, αν για κάθε x και y στο A για τα οποία xRy , τότε επίσης yRx .
- *Αντισυμμετρική.* Η R λέγεται αντισυμμετρική, αν μόνο ένα εκ των δύο ζευγών (x, y) και (y, z) μπορεί να ανήκει στην R , δηλαδή αν xRy και $x \neq y$, τότε δεν μπορεί να ισχύει yRx .
- *Μεταβατική.* Η R λέγεται μεταβατική, αν για κάθε x, y, z στο A για τα οποία xRy και yRz , τότε επίσης xRz .
- *Σχέση Ισοδυναμίας.* Η R λέγεται σχέση ισοδυναμίας αν είναι ανακλαστική, συμμετρική και μεταβατική.

Έστω ότι έχουμε μια σχέση ισοδυναμίας $R \subseteq A \times A$. Αν x είναι ένα στοιχείο του A , δηλώνουμε με $[x]_R$ την *κλάση ισοδυναμίας* που περιέχει το x , δηλαδή το σύνολο

$$[x]_R = \{a \in A \mid aRx\} \quad (\text{B'.9})$$

όλων των στοιχείων a που σχετίζονται με το x . Οι κλάσεις ισοδυναμίας μιας σχέσης R σχηματίζουν μια *διαμέριση* του A , δηλαδή μία συλλογή από υποσύνολα του A ώστε:

1. κανένα υποσύνολο δεν είναι ίσο με το κενό σύνολο $\emptyset$
2. τα υποσύνολα δεν έχουν κοινά στοιχεία μεταξύ τους
3. η ένωσή τους περιέχει όλα τα στοιχεία του A .

Β'.3 Μαθηματική Λογική

Η διαδικασία της απόδειξης έχει ως εξής: πρώτα διατυπώνουμε αυτό που θέλουμε να δείξουμε και στη συνέχεια με *λογικά επιχειρήματα* αποδεικνύουμε την αλήθεια (ή όχι) του *ισχυρισμού μας*.

Τα βασικά αντικείμενα στη λογική είναι οι *ισχυρισμοί* που παίρνουν κάποια *τιμή αληθείας*, μπορούν δηλαδή να είναι αληθείς ή ψευδείς αλλά όχι και τα δύο. Το σύνολο των τιμών που καθιστούν έναν ισχυρισμό αληθή είναι το *σύνολο αληθείας*.

Αν p και q είναι δύο ισχυρισμοί, τότε ο p *συνεπάγεται* τον q ($p \Rightarrow q$), αν η αλήθεια του q προκύπτει από την αλήθεια του p . Δύο ισχυρισμοί p και q είναι *λογικά ισοδύναμοι* ($p \Leftrightarrow q$) αν ο καθένας συνεπάγεται τον άλλον.

Οι σύνδεσμοι *και* και *ή* σχηματίζουν *σύνθετους ισχυρισμούς*. Ο σύνθετος ισχυρισμός $p \wedge q$ είναι αληθής, αν και μόνο αν και ο p και ο q είναι αληθείς. Ο σύνθετος ισχυρισμός $p \vee q$

είναι αληθής, αν ένας τουλάχιστον από τους p και q είναι αληθής.

Η χρήση του *όχι* αντιστρέφει την τιμή αληθείας ενός ισχυρισμού. Ο ισχυρισμός $\neg p$ (ή $\bar{p}$) είναι αληθής, αν και μόνο αν ο p είναι ψευδής.

Ένας άλλος σύνδεσμος είναι ο $\rightarrow$ τέτοιος ώστε με $p \rightarrow q$ να συμβολίζουμε την έκφραση "αν p , τότε q ". Ο $p \rightarrow q$ είναι αληθής, αν και μόνο αν ο p είναι ψευδής ή ο q είναι αληθής. Στην ουσία διακρίνουμε τις εξής περιπτώσεις:

1. Έστω ότι ο p είναι αληθής. Τότε για να είναι ο $p \rightarrow q$ πρέπει και ο q να είναι αληθής.
2. Έστω ότι ο p είναι ψευδής. Τότε ο $p \rightarrow q$ είναι αληθής άσχετα από την τιμή του q .

Άλλος ένας σημαντικός ισχυρισμός είναι ο $p \leftrightarrow q$ που διαβάζεται " p αν και μόνο αν q ". Για να είναι ο ισχυρισμός αυτός αληθής, τα p, q πρέπει να είναι και τα δύο αληθή ή και τα δύο ψευδή.

Κάποιος μπορεί να βρεί την αλήθεια μιας πρότασης της μορφής $p \rightarrow q$ πέφτοντας σε *αντίφαση*, να υποθέσει δηλαδή ότι δεν ισχύει το q , ενώ ισχύει το p , και να καταλήξει σε κάποια ασυνέπεια.

Ένας άλλος τρόπος απόδειξης ισχυρισμών της μορφής $p \rightarrow q$ είναι η *άμεση απόδειξη*, όπου υποθέτουμε ότι το p είναι αληθές και δείχνουμε χρησιμοποιώντας αυτήν την υπόθεση ότι και το q είναι αληθές.

Στην *έμμεση απόδειξη* που χρησιμοποιεί το γεγονός ότι το $p \rightarrow q$ είναι λογικά ισοδύναμο με το $\neg q \rightarrow \neg p$. Η ισοδυναμία προκύπτει ως εξής:

$$\begin{aligned}
 p \rightarrow q &\Leftrightarrow \neg p \vee q && \text{ορισμός } \rightarrow \\
 &\Leftrightarrow q \vee \neg p && \text{αντιμεταθετικότητα} \\
 &\Leftrightarrow \neg(\neg p) \vee \neg q && \text{διπλή χρήση της άρνησης } \neg \\
 &\Leftrightarrow \neg q \rightarrow \neg p && \text{ορισμός } \rightarrow
 \end{aligned}$$

Β'.4 Συμβολοσειρές

Ορισμός Β'.4.1. Αλφάβητο Σ είναι κάθε πεπερασμένο μη κενό σύνολο συμβόλων ή γραμμάτων

$$\Sigma = \{\sigma_1, \sigma_2, \dots, \sigma_\nu\} \quad (\text{Β'.10})$$

Χρησιμοποιούμε το αλφάβητο για την παράσταση των οποιονδήποτε πληροφοριών, ή την περιγραφή των οποιονδήποτε αντικειμένων, σκοπεύουμε να χειριστούμε υπολογιστικά. Τα συνηθισμένα γράμματα, τα αριθμητικά ψηφία και μερικά σημεία στίξης είναι υπεραρκετά

για μια φυσική παράσταση οποιασδήποτε πληροφορίας. Θεωρητικά για παράδειγμα στα ολοκληρωμένα κυκλώματα ένα δυαδικό αλφάβητο δύο συμβόλων, "0" και "1", είναι επαρκές.

Ορισμός Β'.4.2. Συμβολοσειρά (string) ή αλφωώς λέξη λ ενός αλφαβήτου Σ είναι μια πεπερασμένη ακολουθία συμβόλων του Σ .

$$\lambda = \sigma_1 \sigma_2 \dots \sigma_\nu, \sigma_\kappa \in \Sigma, \kappa = 1, \dots, \nu \quad (\text{B'.11})$$

Το ν είναι το μήκος της λέξης και το συμβολίζουμε με $|\lambda|$ και αποτελεί τον αριθμό των συμβόλων της συμβολοσειράς. Επιτρέπεται η τιμή $\nu = 0$ στην οποία περίπτωση θεωρούμε ότι η λέξη μας είναι κενή και συμβολίζεται με το σύμβολο ϵ ή $\emptyset$. Το σύνολο όλων των πεπερασμένων συμβολοσειρών εξ ενός αλφαβήτου Σ συμβολίζεται με Σ^* .

Τι είναι ακριβώς όμως οι συμβολοσειρές; Όταν υπολογίζουμε, υπάρχουν πράγματα που θυμόμαστε, σημειώσεις που κρατάμε. Αυτά θα τα παριστάνουμε συμβολικά με μία λέξη. Επιλεγμένα σύμβολα που χρησιμοποιούνται ως σημεία στίξης είναι δυνατόν να ενώνουν συμβατικά πολλές λέξεις σε μια. Το παρόν κείμενο για παράδειγμα μπορεί να θεωρηθεί ως μια λέξη με σημεία στίξης το "διάστημα", την "τελεία", τις "παύλες", τις "παρενθέσεις", κα. με μια συμβολοσειρά παριστάνουμε τα δεδομένα που έχουμε, τη διαθέσιμη σε μας πληροφορία, δηλαδή την περιγραφή των αντικειμένων που χρειαζόμαστε. Το ότι αυτή είναι μια απλή μονοδιάστατη σειρά συμβόλων δεν μας οδηγεί σε απώλεια γενικότητας, διότι αυτή η σειρά δεν αντιστοιχεί παρά στη χρονική σειρά με τα οποία ούτως ή άλλως θα τα εξετάσουμε. Ό,τι εμείς, ή μια υπολογιστική συσκευή, εξετάζει "σε ένα βήμα" της, αυτό το θεωρούμε ως ένα σύμβολο, και η σειρά εξέτασης αντιστοιχεί στη συμβολοσειρά που θεωρούμε ως τα δεδομένα μας.

Για παράδειγμα αλφάβητο είναι το Ελληνικό αλφάβητο $\{\alpha, \beta, \dots, \omega\}$ με ϵ , *ενα, τρια* να είναι συμβολοσειρές του πεπερασμένου Ελληνικού αλφαβήτου και τα μήκη των συμβολοσειρών αυτών είναι αντίστοιχα 0, 3, 4. Το Λατινικό αλφάβητο $\{a, b, \dots, z\}$ έχει συμβολοσειρές όπως ϵ , *one, three* με μήκη 0, 3, 5 αντίστοιχα. Αν θεωρήσουμε το "δυαδικό αλφάβητο" $\{0, 1\}$ έχουμε συμβολοσειρές όπως ϵ , 0, 01011 με μήκη 0, 1, 5.

Β'.4.1 Πράξεις με συμβολοσειρές

- *Παράθεση (concatenation):* Η παράθεση δύο συμβολοσειρών x και y είναι η συμβολοσειρά xy και σχηματίζεται με απλή παράθεση της ακολουθίας συμβόλων της y αμέσως μετά από τα σύμβολα της x και την συμβολίζουμε με xy ή $x \circ y$. Σε αυτήν την περίπτωση θα έχουμε $|xy| = |x| + |y|$. Η κενή συμβολοσειρά είναι το ουδέτερο στοιχείο για την παράθεση συμβολοσειρών ενός αλφαβήτου Σ , δηλαδή ισχύει

$$\forall y \in \Sigma^*, y\epsilon = \epsilon y = y \quad (\text{B'.12})$$

Επίσης, η παράθεση ως συνάρτηση που ορίζεται στο σύνολο των συμβολοσειρών έχει την προσεταιριστική ιδιότητα, δηλαδή

$$\forall u, v, w \in \Sigma^*, ((uv)w) = (u(vw)) \quad (\text{B'.13})$$

- *Επανάληψη*: Αν w είναι μια συμβολοσειρά, τότε w^i για οποιονδήποτε φυσικό αριθμό i αποτελείται από την παράθεση i αντιγράφων του w , ορίζεται επαγωγικά ως εξής:

$$\begin{aligned} w^0 &= \epsilon \\ w^{i+1} &= w^i w \end{aligned} \quad (\text{B'.14})$$

- *Αντίστροφη*: Η αντίστροφη μιας συμβολοσειράς w συμβολίζεται με w^R και προκύπτει αν διαβάσουμε το w από το τέλος προς την αρχή. Πάλι επαγωγικά ορίζεται ως εξής:

$$\begin{aligned} \text{Αν } |w| = 0 \text{ τότε } w^R &= w = \epsilon \\ \text{Αν } |w| \geq 1 \text{ τότε } \exists \alpha \in \Sigma, u \in \Sigma^* \text{ ώστε } w &= u\alpha \text{ και } w^R = \alpha u^R \end{aligned} \quad (\text{B'.15})$$

μια συμβολοσειρά u είναι *υποσυμβολοσειρά* της w αν και μόνο αν η u εμφανίζεται σε συνεχόμενες θέσεις ως προς ένα μέρος της w . μια συμβολοσειρά είναι *κανονική συμβολοσειρά* της w αν και μόνο αν η u είναι υποσυμβολοσειρά της w και συγχρόνως $u \neq w$. Κάθε συμβολοσειρά είναι συμβολοσειρά (όχι όμως κανονική υποσυμβολοσειρά) του εαυτού της. Η κενή συμβολοσειρά ϵ είναι υποσυμβολοσειρά κάθε άλλης συμβολοσειράς. μια συμβολοσειρά u είναι *κατάληξη* της w αν και μόνο αν $\exists v \in \Sigma^*, w = vu$. μια συμβολοσειρά u είναι *κανονική κατάληξη* της w , αν και μόνο αν η u είναι κατάληξη της w και συγχρόνως $u \neq w$. Η κενή συμβολοσειρά ϵ είναι κατάληξη κάθε άλλης συμβολοσειράς. Από την άλλη, μια συμβολοσειρά u είναι *πρόθεμα* της w αν και μόνο αν $\exists v \in \Sigma^*, w = uv$. μια συμβολοσειρά u είναι *κανονικό πρόθεμα* της w , αν και μόνο αν η u είναι πρόθεμα της w και συγχρόνως $u \neq w$. Η κενή συμβολοσειρά ϵ είναι πρόθεμα κάθε άλλης συμβολοσειράς.

Η αναζήτηση πληροφορίας είναι μία από τις πιο σημαντικές επεξεργασίες και στις περισσότερες περιπτώσεις στηρίζεται στην ύπαρξη κάποιας σχέσης, που ορίζει μια διάταξη μεταξύ των οντοτήτων του συνόλου αναζήτησης. μία ευρέως χρησιμοποιούμενη σχέση μεταξύ συμβολοσειρών είναι η σύγκρισή τους αλφαριθμητικά με το ίδιο κριτήριο, που αυτές διατάσσονται σε κάθε ταξινομημένη λίστα όπως ο τηλεφωνικός κατάλογος. Η συγκεκριμένη σχέση και διάταξη των συμβολοσειρών ορίζεται ως εξής:

Ορισμός Β'.4.3. *Αλφαριθμητική διάταξη συμβολοσειρών.* Για δύο συμβολοσειρές u και w στο αλφάβητο Σ λέμε ότι η u είναι *αλφαριθμητικά μικρότερη* από τη συμβολοσειρά w στο Σ^* ή *ισοδύναμα* ότι η w είναι *αλφαριθμητικά μεγαλύτερη* από την u , αν ισχύει κάποια από τις παρακάτω δύο περιπτώσεις:

- η u είναι *κανονικό πρόθεμα* της w , ή

- για κάποια $v \in \Sigma^*$ και κάποια σύμβολα $\alpha, \beta \in \Sigma$ τέτοια ώστε το α να προηγείται του β στο Σ , η συμβολοσειρά να είναι πρόδεμα της u και η $v\beta$ να είναι πρόδεμα της w .

Ένα διατεταγμένο υποσύνολο του Σ^* λέμε ότι είναι σε αλφαβητική διάταξη, αν για κάθε δύο συμβολοσειρές u, w , με την u να προηγείται της w μέσα στο υποσύνολο, δε συμβαίνει η w να είναι αλφαβητικά μικρότερη από την u στο Σ^* . Η αλφαβητική διάταξη είναι ένα επαρκές κριτήριο για πεπερασμένα σύνολα, επειδή σ'ένα τέτοιο σύνολο μπορεί τελικά να προσεγγιστεί η κάθε συμβολοσειρά που περιέχεται σ'αυτό. Όμως στην περίπτωση άπειρων συνόλων συμβολοσειρών η αλφαβητική ταξινόμηση δεν είναι ικανοποιητική, επειδή μπορεί γι κάποιες συμβολοσειρές ο αριθμός όσων προηγούνται να μην είναι φραγμένος. μια τέτοια περίπτωση είναι για παράδειγμα αυτή της συμβολοσειράς 1 στο αλφαβητικά ταξινομημένο σύνολο $\{0, 1\}^*$, εκεί δηλαδή που οι συμβολοσειρές 0, 00, 000, ... προηγούνται της συμβολοσειράς 1. Το συγκεκριμένο πρόβλημα καταδεικνύει την αναγκαιότητα του παρακάτω ορισμού της λεξικογραφικής διάταξης των συμβολοσειρών. με τη συγκεκριμένη διάταξη, ο αριθμός των συμβολοσειρών που προηγούνται μιας δοθείσας συμβολοσειράς είναι πάντα πεπερασμένος.

Ορισμός Β'.4.4. *Κανονική ή λεξικογραφική διάταξη.* Για δύο συμβολοσειρές u και w στο αλφάβητο Σ λέμε ότι η u είναι κανονικά ή λεξικογραφικά μικρότερη από τη συμβολοσειρά w στο Σ^* ή ισοδύναμα ότι η w είναι λεξικογραφικά μεγαλύτερη από την u , αν ισχύει κάποια από τις παρακάτω δύο περιπτώσεις:

- η u έχει μικρότερο μήκος από την w , ή
- οι u και w έχουν το ίδιο μήκος, αλλά η u είναι αλφαβητικά μικρότερη από την w

Ένα διατεταγμένο υποσύνολο του Σ^* λέμε ότι είναι διατεταγμένο κανονικά ή λεξικογραφικά, αν για κάθε δύο συμβολοσειρές u, w , με την u να προηγείται της w μέσα στο υποσύνολο, δε συμβαίνει η w να είναι κανονικά μικρότερη από την u στο Σ^* .

Επιστρέφουμε τώρα στο ζήτημα της αναπαράστασης της πληροφορίας με βάση τις έννοιες των συμβολοσειρών ως προς ένα αλφάβητο, που ορίσαμε παραπάνω. Κάθε τέτοια αναπαράσταση μπορούμε να την ορίσουμε ως μία αντιστοίχιση αντικειμένων σε συμβολοσειρές με βάση κάποιους κανόνες. Τυπικά, μία αναπαράσταση ή κωδικοποίηση ενός συνόλου D σ'ένα αλφάβητο Σ είναι μία συνάρτηση f από το D στο 2^{Σ^*} , που ικανοποιεί την παρακάτω συνθήκη: για δύο οποιαδήποτε στοιχεία e_1 και e_2 του D , τα $f(e_1)$ και $f(e_2)$ είναι διακριτά μη κενά σύνολα. Σ αυτήν την περίπτωση, κάθε στοιχείο του $f(e)$ είναι μια αναπαράσταση ή κωδικοποίηση του e .

Β'.5 Γλώσσες

Οι συμβολοσειρές είναι ένα μέσο αναπαράστασης της πληροφορίας στο βαθμό όμως που υπάρχει μια συνάρτηση, που παρέχει την ερμηνεία των συμβολοσειρών ως προς την πληροφορία, που αυτές αναπαριστούν. μια ερμηνεία είναι το αντίστροφο της αντιστοίχισης, που

προκύπτει από μια αναπαράσταση, δηλαδή μια ερμηνεία είναι κάποια συνάρτηση g από το Σ^* στο σύνολο D των αντικειμένων της πληροφορίας, για κάποιο αλφάβητο Σ . Για παράδειγμα, η συμβολοσειρά 111, ερμηνεύεται ως ο αριθμός εκατόν έντεκα σε αναπαράσταση αριθμό στο δεκαδικό σύστημα ή ως αριθμός επτά σε αναπαράσταση αριθμού του δυαδικού συστήματος. Κατά την επικοινωνία δύο μερών η πληροφορία που μεταβιβάζεται είναι άρρηκτα δεμένη με την αναπαράσταση και την ερμηνεία της. Η αναπαράστασή της παρέχεται από τον αποστολέα και η ερμηνεία της από τον παραλήπτη. Κατά συνέπεια, από τη σκοπιά των δύο μερών της επικοινωνίας, μια γλώσσα μπορεί απλά να είναι μια συλλογή συμβολοσειρών, καθώς οι συναρτήσεις αναπαράστασης και ερμηνείας είναι γνωστές στα μέρη.

Ορισμός Β'.5.1. Γλώσσα είναι ένα πεπερασμένο ή άπειρο σύνολο συμβολοσειρών ενός πεπερασμένου αλφάβητου Σ , δηλαδή κάθε γλώσσα είναι υποσύνολο του Σ^* .

μια γλώσσα θα μπορούσε να οριστεί με τον ακόλουθο τρόπο: Έστω η γλώσσα

$$L = \{w \in \{0, 1\}^* : \text{τα } 0 \text{ προηγούνται όλων των } 1\}$$

Οι συμβολοσειρές $\epsilon, 0, 00, 00111$ και 11 ανήκουν στην L .

Β'.5.1 Πράξεις με Γλώσσες

- Ένωση δύο γλωσσών L_1, L_2 είναι η γλώσσα με τις συμβολοσειρές που ανήκουν στην L_1 ή στην L_2 .

$$L_1 \cup L_2 = \{w, w \in L_1 \text{ ή } w \in L_2\} \quad (\text{B'.16})$$

- Τομή της L_1 με την L_2 είναι η γλώσσα με όλες τις συμβολοσειρές της, που ανήκουν τόσο στην L_1 όσο και στην L_2 .

$$L_1 \cap L_2 = \{w, w \in L_1 \text{ και } w \in L_2\} \quad (\text{B'.17})$$

- Συμπλήρωμα μιας γλώσσας L με αλφάβητο Σ αναφέρεται στη γλώσσα που αποτελείται από όλες τις συμβολοσειρές του Σ , που δεν ανήκουν στην L .

$$\bar{L} = \{w, w \in \Sigma^* \text{ και } w \notin L\} \quad (\text{B'.18})$$

- Διαφορά δύο γλωσσών L_1 και L_2 αναφέρεται στη γλώσσα που αποτελείται από όλες τις συμβολοσειρές, που ανήκουν στην L_1 , αλλά δεν ανήκουν στην L_2 .

$$L_1 - L_2 = \{w, w \in L_1 \text{ και } w \notin L_2\} \quad (\text{B'.19})$$

- Παράθεση δύο γλωσσών L_1 και L_2 με αλφάβητο Σ είναι η γλώσσα

$$L_1 L_2 = \{w \in \Sigma^* : w = uv \text{ για } u \in L_1, v \in L_2\} \quad (\text{B'.20})$$

Η παράθεση γλωσσών είναι προσεταιριστική πράξη, δηλαδή για όλες τις γλώσσες L_1, L_2 και L_3 ισχύει:

$$(L_1 L_2) L_3 = L_1 (L_2 L_3) \quad (\text{B'.21})$$

- *Κλεισιότητα* ή αλλιώς *Kleene star* μίας γλώσσας L με αλφάβητο Σ είναι το σύνολο των συμβολοσειρών που σχηματίζεται με παράθεση μηδέν ή περισσότερων συμβολοσειρών της L :

$$L^* = \{\epsilon\} \cup \{w \in \Sigma^* : \exists k \geq 1, w = w_1 w_2 \dots w_k \text{ για } w_1, w_2, \dots, w_k \in L\} \quad (\text{B'.22})$$

Αν η L δεν είναι κενή και δεν είναι η $\{\epsilon\}$, τότε η L^* σίγουρα περιέχει έναν άπειρο αριθμό συμβολοσειρών. Αν $L = \emptyset$ τότε $L^* = \{\epsilon\}$, καθώς δεν υπάρχουν συμβολοσειρές που θα μπορούσαν να παρατεθούν στην ϵ . Αν τώρα $L = \{\epsilon\}$ τότε επίσης $L^* = \{\epsilon\}$.

- *Αντιστροφή* μιας γλώσσας L με αλφάβητο Σ , L^R είναι το σύνολο των συμβολοσειρών που σχηματίζονται από την αντιστροφή κάποιας συμβολοσειράς της L :

$$L^R = \{w \in \Sigma^* : w = v^R \text{ για κάποια } v \in L\} \quad (\text{B'.23})$$

Κεφάλαιο Γ'

Θεωρία Υπολογισμού

Γ'.1 Παραγωγή και Αναγνώριση Γλωσσών στη Θεωρία Υπολογισμού

Καθώς οι γλώσσες είναι σύνολα, από υπολογιστικής άποψης ο ορισμός μιας γλώσσας μπορεί να είναι:

- ένας *παραγωγός συμβολοσειρών*, που περιγράφει το πώς παράγονται όλα τα στοιχεία της γλώσσας ή
- ένας *αναγνωριστής συμβολοσειρών*, που αποφασίζει αν μια συμβολοσειρά ανήκει ή δεν ανήκει στην γλώσσα.

Στη Θεωρία Υπολογισμού, κάθε πρόβλημα που επιθυμούμε να επιλύσουμε το διατυπώνουμε πρώτα χρησιμοποιώντας μία αναπαράσταση με συμβολοσειρές. Τα αποτελέσματα που έχουμε δίνουν απάντηση στο ερώτημα: "Ανήκει η w στη γλώσσα L ?", που είναι ένα πρόβλημα απόφασης. Το ερώτημα που λογικά προκύπτει είναι αν μία θεωρία που περιορίζεται σ' αυτού του τύπου τα αποτελέσματα είναι πρακτικά χρήσιμη στη μελέτη πραγματικών υπολογιστικών προβλημάτων. Παρατηρούμε λοιπόν ότι με την κατάλληλη αναπαράσταση, πολλά προβλήματα της καθημερινότητας μπορούν να αναδιατυπωθούν ως πρόβλημα απόφασης του αν μια συμβολοσειρά ανήκει σε μια γλώσσα ή όχι. Θα δούμε πρόβληματα, που εντάσσονται σε δύο κατηγορίες:

- Προβλήματα που είναι ήδη διατυπωμένα ως προβλήματα απόφασης. Γιαυτό, χρειάζεται να βρούμε μια αναπαράσταση της εισόδου τους με συμβολοσειρές και στη συνέχεια να ορίσουμε μια γλώσσα, που θα περιέχει ακριβώς εκείνο το σύνολο των εισόδων, για τις οποίες η απάντηση στο ερώτημα της απόφασης είναι ΝΑΙ.
- Προβλήματα, που δεν έχουν εξ αρχής περιγραφεί ως ερωτήματα απόφασης. Γιαυτό, μπορεί να απαιτούνται διαφόρων ειδών αποτελέσματα, που οδηγούν στην αναδιατύπω-

ση του προβλήματος ως πρόβλημα απόφασης, που στη συνέχεια θα πρέπει να αναπα-
ρασταθεί ως διαδικασία αναγνώρισης γλώσσας.

Θα δώσουμε κάποια παραδείγματα προβλημάτων. Συμβολίζουμε με $\langle X \rangle$ μια αναπαρά-
σταση του αντικειμένου X με συμβολοσειρά. Συμβολίζουμε με $\langle X, Y \rangle$ την αναπαράσταση
δύο αντικειμένων X, Y με μια συμβολοσειρά.

Παράδειγμα Γ'.1.1. Αναγνώριση πρώτων αριθμών. Δοθέντος ενός μη αρνητικού ακεραίου
 n είναι αυτός πρώτος. Αναζητούμε δηλαδή αν υπάρχει τουλάχιστον ένας θετικός ακέραιος
παράγοντας διαφορετικός του αριθμού αυτού ή της μονάδας. Κωδικοποιώντας το πρόβλημα
αναπαριστούμε κάθε μη αρνητικό ακέραιο ως συμβολοσειρά από δυαδικά ψηφία. Σ' αυτήν
την περίπτωση, η γλώσσα που αποφασίζεται είναι η:

$$L_{\text{prime}} = \{w : \eta w \text{ είναι η δυαδική αναπαράσταση ενός πρώτου αριθμού} \} \quad (\Gamma'.1)$$

Παράδειγμα Γ'.1.2. Αναζήτηση προτύπου στον παγκόσμιο ιστό. Δοθείσης μιας συμβολοσειράς
αναζήτησης w και ενός πόρου d του παγκόσμιου ιστού ταιριάζει το δεύτερο με το πρώτο· μ'
άλλα λόγια, θα έπρεπε μια μηχανή αναζήτησης με είσοδο w να επιστρέφει το d . Σ' αυτήν την
περίπτωση, η γλώσσα που αποφασίζεται είναι η:

$$L_{\text{www}} = \{ \langle w, d \rangle : \text{το } d \text{ ταιριάζει στο ερώτημα } w \} \quad (\Gamma'.2)$$

Παράδειγμα Γ'.1.3. Το πρόβλημα του τερματισμού. Δοθέντος ενός προγράμματος p , που έχει
γραφεί στη γλώσσα προγραμματισμού C είναι σίγουρο ότι το πρόγραμμα p τερματίζει για όλες
τις εισόδους. Σ' αυτήν την περίπτωση, η γλώσσα που αποφασίζεται είναι η:

$$L_{\text{halt}} = \{p : \text{το πρόγραμμα } p \text{ τερματίζει για όλες τις εισόδους} \} \quad (\Gamma'.3)$$

Φυσικά θα θέλαμε πολύ να υπάρχει ένας αλγόριθμος που θα μπορούσε να αποφασίσει αν μία
συμβολοσειρά- πρόγραμμα- ανήκει στη γλώσσα L_{halt} γιατί έτσι θα αποδεικνύαμε την ορθότητα
των προγραμμάτων που συντάσσουμε. Δυστυχώς ένα θεώρημα της Θεωρίας Υπολογισμού έχει
δείξει ότι ένας τέτοιος αλγόριθμος δε μπορεί να υπάρξει.

Παράδειγμα Γ'.1.4. Ευθυγράμμιση ακολουθίας πρωτεΐνης/ Πρόβλημα Υπολογιστικής Βιο-
λογίας. Δοθέντος ενός μέρους f μιας πρωτεΐνης και ενός μορίου πρωτεΐνης pr μπορεί το f
να είναι μέρος του pr . Κωδικοποιώντας το πρόβλημα αναπαριστούμε κάθε μόριο πρωτεΐνης
ή ενός μέρους του ως μια ακολουθία στοιχείων αμινοξέων. Ορίζουμε έναν χαρακτήρα για το
καθένα από τα είκοσι πιθανά αμινοξέα. Έτσι θα μπορούσε για παράδειγμα η συμβολοσει-
ρά AGHTYWDNR να αναπαριστά μέρος κάποιας πρωτεΐνης. Σ' αυτήν την περίπτωση, η
γλώσσα που αποφασίζεται είναι η:

$$L_{\text{pr}} = \{ \langle f, p \rangle : \text{το } f \text{ θα μπορούσε να είναι μέρος του } pr \} \quad (\Gamma'.4)$$

Φυσικά θα θέλαμε πολύ να υπάρχει ένας αλγόριθμος που θα μπορούσε να αποφασίσει αν μία
συμβολοσειρά- πρόγραμμα- ανήκει στη γλώσσα L_{halt} γιατί έτσι θα αποδεικνύαμε την ορθότητα

των προγραμμάτων που συντάσσουμε. Δυστυχώς ένα θεώρημα της Θεωρίας Υπολογισμού έχει δείξει ότι ένας τέτοιος αλγόριθμος δε μπορεί να υπάρξει.

Πολλά προβλήματα που δεν είναι διατυπωμένα ως προβλήματα απόφασης μπορούν να μετασχηματιστούν σε τέτοια με μία αναδιατύπωσή τους ως προβλήματα αναγνώρισης γλώσσας. Η βασική ιδέα είναι η αναπαράσταση σε μία συμβολοσειρά τόσο των εισόδων, όσο και των εξόδων του αρχικού προβλήματος. Έτσι, για παράδειγμα, αν για το πρόβλημα Π έχουμε δύο εισόδους i_1 και i_2 και παράγεται ένα αποτέλεσμα r , τότε μπορούμε να σχηματίσουμε συμβολοσειρές της μορφής $i_1; i_2; r$. Σ αυτή την περίπτωση, η συμβολοσειρά $w = u; v; z$ θα ανήκει στη γλώσσα L για το πρόβλημα Π , αν και μόνο αν z είναι η συμβολοσειρά που αναπαριστά το αποτέλεσμα για δύο εισόδους στο πρόβλημα Π , που εκφράζονται από τις συμβολοσειρές x και y .

Παράδειγμα Γ'.1.5. Άθροιση ακεραίων ως πρόβλημα απόφασης. Το πρόβλημα συνίσταται στον υπολογισμό του αθροίσματος δύο μη αρνητικών ακεραίων. Για την κωδικοποίηση του προβλήματος αναπαριστούμε τον κάθε αριθμό ως συμβολοσειρά δεκαδικών ψηφίων και μετασχηματίζουμε το αρχικό πρόβλημα σ'ένα πρόβλημα ελέγχου του αν σε μια τριάδα αριθμών ο τρίτος είναι το άθροισμα των δύο πρώτων. Έτσι κάθε περίπτωση του προβλήματος διατυπώνεται με μια συμβολοσειρά της μορφής

$$\langle \text{ακέραιος}_1 \rangle + \langle \text{ακέραιος}_2 \rangle = \langle \text{ακέραιος}_3 \rangle$$

Τότε η γλώσσα που αποφασίζεται για το συγκεκριμένο πρόβλημα είναι η:

$$\begin{aligned} L_{\text{sum}} = \{ w : w \text{ είναι της μορφής } &\langle \text{ακέραιος}_1 \rangle + \langle \text{ακέραιος}_2 \rangle = \langle \text{ακέραιος}_3 \rangle \\ &\text{με } \langle \text{ακέραιος}_1 \rangle, \langle \text{ακέραιος}_2 \rangle, \langle \text{ακέραιος}_3 \rangle \text{ στοιχεία της} \\ &\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}^+ \text{ και ο } \langle \text{ακέραιος}_3 \rangle \text{ να είναι το άθροισμα των} \\ &\langle \text{ακέραιος}_1 \rangle \text{ και } \langle \text{ακέραιος}_2 \rangle \} \end{aligned}$$

Γ'.2 Κλάσεις γλωσσών

Γ'.2.1 Κλάσεις γλωσσών ως προς τις μηχανές για την αναγνώρισή τους

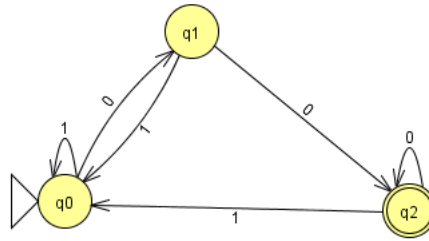
Το κάθε μοντέλο υπολογισμού έχει τη δυνατότητα εκτέλεσης προγραμμάτων, όπου ένα πρόγραμμα εδώ ορίζεται ως μια πεπερασμένη ακολουθία εντολών με ένα πεδίο ορισμού, έστω D . Το πεδίο ορισμού D λέμε ότι είναι το πεδίο τιμών των μεταβλητών και υποθέτουμε ότι αποτελείται από ένα σύνολο στοιχείων, από τα οποία ένα ξεχωρίζει, επειδή αντιστοιχεί στην αρχική τιμή των μεταβλητών του προγράμματος. Κάθε ένα από τα στοιχεία του συνόλου D υποθέτουμε ότι είναι ένας συνδυασμός ανάθεσης τιμών στις μεταβλητές. Κατά την εκτέλεση ενός τέτοιου προγράμματος διακρίνουμε δύο πιθανά ενδεχόμενα:

- *Αποδοχή*, που αντιστοιχεί σε κάποια εντολή τερματισμού μετά από επιτυχή ολοκλήρωση της εκτέλεσης και που μπορεί να συμβεί μόνο μετά από την προσέγγιση του τέλους της εισόδου.
- *Απόρριψη*, που αντιστοιχεί σε κάποια εντολή τερματισμού μετά από ανεπιτυχή ολοκλήρωση της εκτέλεσης.

μια ακολουθία βημάτων εκτέλεσης λέγεται *υπολογισμός αποδοχής* αν τερματίζει λόγω κάποιας εντολής αποδοχής. Αντίθετα, μια ακολουθία βημάτων εκτέλεσης για είσοδο, που δεν έχει υπολογισμούς αποδοχής, λέγεται *υπολογισμός απόρριψης*. Αν ένα πρόγραμμα έχει έναν υπολογισμό αποδοχής για μια είσοδο, τότε αυτή η είσοδος είναι *αποδεκτή* ή αλλιώς *αναγνωρίζεται*. Σε διαφορετική περίπτωση η είσοδος δεν είναι *αποδεκτή* ή αλλιώς απορρίπτεται από το πρόγραμμα. Ένα πρόγραμμα παράγει την έξοδο y για δοθείσα είσοδο x , αν αυτό έχει έναν υπολογισμό αποδοχής για τη x , που να παράγει την έξοδο y . Οι έξοδοι των υπολογισμών απόρριψης θεωρούμε ότι είναι απροσδιόριστες. Στο σημείο αυτό θα παρουσιάσουμε τα μοντέλα υπολογισμού με τη πιο σημαντική επίδραση στην εξέλιξη της Επιστήμης των Υπολογιστών. Θεωρούμε χωρίς βλάβη της γενικότητας ότι η τρέχουσα κατάσταση των προγραμμάτων κατά την εκτέλεσή τους καθορίζεται από μια μόνο μεταβλητή, που η τιμή της θα μεταβάλλεται με τρόπο, που θα απεικονίζεται σε διαγράμματα καταστάσεων.

Γ'.2.1.1 Πεπερασμένα Αυτόματα

Το πρώτο από τα μοντέλα υπολογισμού, που θα εξετάσουμε είναι οι *μηχανές πεπερασμένων καταστάσεων*, γνωστές επίσης και ως *πεπερασμένα αυτόματα*. Τα προγράμματα γι'αυτές τις μηχανές είναι γενικά εύκολα στην κατανόηση, εκτελούνται σε γραμμικά μεταβαλλόμενο χρόνο και υπάρχουν αποτελέσματα για σχεδόν όλες τις ερωτήσεις, σχετικά με τη συμπεριφορά τους. Το Σχήμα Γ'.1 δείχνει ένα απλό πεπερασμένο αυτόματο, που αποδέχεται/ αναγνωρίζει συμβολοσειρές από 0 και 1, που τελειώνουν σε 00. Η είσοδος σ'ένα πεπερασμένο αυτόματο είναι μια συμβολοσειρά με τους χαρακτήρες της να διαβάζονται ένας κάθε φορά, από τάρισερα προς τα δεξιά. Το αυτόματο έχει μια αρχική κατάσταση, όπως φαίνεται στο σχήμα με ένα τόξο χωρίς επιγραφή, που την υποδεικνύει, καθώς και έναν αριθμό από καταστάσεις αποδοχής ή τελικές καταστάσεις, που απεικονίζονται με ένα διπλό κύκλο. Ο υπολογισμός του πεπερασμένου αυτομάτου ξεκινά από την αρχική κατάσταση. με την ανάγνωση του κάθε χαρακτήρα, το αυτόματο αλλάζει κατάσταση με βάση τις μεταβάσεις, που φαίνονται στο σχήμα. Αν μετά από την ανάγνωση του τελευταίου χαρακτήρα της συμβολοσειράς w το αυτόματο M βρίσκεται σε μια κατάσταση αποδοχής, τότε η μηχανή M αποδέχεται την w , ενώ σε διαφορετική περίπτωση την απορρίπτει. Ο υπολογισμός για το αυτόματο του παραδείγματος παραμένει στην κατάσταση q_0 όσο διαβάζει χαρακτήρες 1 και μεταβαίνει στην κατάσταση q_1 όταν συναντήσει το πρώτο 0. Στην συνέχεια αρκεί ένας χαρακτήρας 0, για να προσεγγίσει την τελική κατάσταση q_2 , όπου και θα παραμείνει αν αυτός είναι ο τελευταίος χαρακτήρας.

**Σχήμα Γ'.1:** Ένα απλό αυτόματο

Αν αυτό δεν ισχύει, τότε ο μόνος τρόπος για να επιστρέψει στην τελική κατάσταση είναι η εμφάνιση στη συμβολοσειρά δύο διαδοχικών 00. Το αυτόματο του παραδείγματος αποδέχεται, για παράδειγμα, τη συμβολοσειρά 110100 και απορρίπτει την 010. Η κλάση των γλώσσών, που γίνονται αποδεκτές (αναγνωρίζονται) από κάποια μηχανή πεπερασμένων καταστάσεων είναι η κλάση των *καινουργιών γλώσσών*. Πολλές γλώσσες, απάυτες που χρησιμοποιούμε στην καθημερινή μας ζωή είναι κανονικές γλώσσες, όπως για παράδειγμα το σύνολο των συμβολοσειρών, που εκφράζουν αριθμούς κινητής υποδιαστολής, οι σωστές ακολουθίες κερμάτων σε μια αυτόματη μηχανή πώλησης, για την αγορά ενός αναψυκτικού, οι σωστές αλληλουχίες μηνυμάτων σε πρωτόκολλα επικοινωνίας συστημάτων και άλλα. Εκτός από τις γλώσσες, που περιγράψαμε, υπάρχουν απλές γλώσσες, που όμως δεν είναι κανονικές. Ας θεωρήσουμε για παράδειγμα τη γλώσσα :

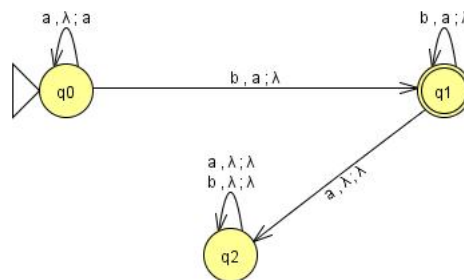
$$L_1 = \{a^n b^n : n \geq 0\} \quad (\Gamma.5)$$

Σε κάθε συμβολοσειρά της γλώσσας L_1 προηγούνται όλα τα a και ο αριθμός των a είναι ίσος με τον αριθμό των b , που έπονται. Θα μπορούσαμε να προσπαθήσουμε να αναπαραστήσουμε τη γλώσσα αυτή μ'ένα πεπερασμένο αυτόματο. Το πρόβλημα που θα αντιμετωπίσουμε είναι το πώς μπορούμε να καταμετρήσουμε τον αριθμό των a , ώστε να τον συγκρίνουμε στη συνέχεια με τον αριθμό των b . Η "μνήμη" ενός πεπερασμένου αυτόματου είναι οι καταστάσεις του και μόνο εκεί μπορεί να καταγραφεί πληροφορία, αλλά κάθε αυτόματο αποτελείται από ένα σταθερό αριθμό καταστάσεων, που αποφασίζεται κατά τον ορισμό της μηχανής. Από την άλλη στον ορισμό της γλώσσας L_1 δεν υπάρχει όριο στον αριθμό των a , που μπορεί να περιέχει μια συμβολοσειρά. Στην πραγματικότητα έχει αποδειχθεί ότι δεν υπάρχει πεπερασμένο αυτόματο, που να αποδέχεται τη γλώσσα L_1 . Η ίδια διαπίστωση ισχύει και για πολλές άλλες γλώσσες, απάυτες που χρησιμοποιούμε στην καθημερινή μας ζωή, όπως για παράδειγμα η γλώσσα με τις συμβολοσειρές, που έχουν ισορροπημένες (ένθετες) αριστερές και δεξιές παρενθέσεις. Οι γλώσσες αυτές είναι σημαντικές, γιατί θέλουμε να έχουμε διαδικασίες απόφασης, για να εντοπίζεται π.χ. μία λάθος χρήση παρενθέσεων στις αριθμητικές και λογικές εκφράσεις, για κάποια γλώσσα προγραμματισμού.

Γ'.2.1.2 Αυτόματα Στοίβας

Το πρόβλημα της αναπαράστασης των μη κανονικών γλωσσών μπορεί να αντιμετωπιστεί επεκτείνοντας τα πεπερασμένα αυτόματα, για να τους διθεί η δυνατότητα χρήσης μιας στοίβας. Κάθε μηχανή, που αποτελείται από ένα πεπερασμένο αυτόματο και μία στοίβα ονομάζεται *αυτόματο στοίβας* (*pushdown automaton*).

Ένα αυτόματο στοίβας M_{PA} αποδέχεται τη γλώσσα L_1 με τον παρακάτω υπολογισμό. Κάθε φορά, που διαβάζεται ένα a , το M_{PA} το εναποθέτει στη στοίβα. Στη συνέχεια, κάθε φορά που διαβάζει ένα b θα εξάγεται από τη στοίβα ένα a . Αν στο τέλος έχει διαβαστεί και ο τελευταίος χαρακτήρας της συμβολοσειράς και η στοίβα έχει αδειάσει, ενώ το αυτόματο βρίσκεται σε μια τελική κατάσταση, τότε η συμβολοσειρά γίνεται δεκτή. Σε κάθε άλλη περίπτωση η συμβολοσειρά απορρίπτεται. Οι καταστάσεις της μηχανής M_{PA} και οι μεταξύ τους μεταβάσεις θα είναι κατάλληλα σχεδιασμένες, ώστε να διασφαλίζουν ότι όλα τα a στη συμβολοσειρά θα εμφανίζονται πριν από τα b . Όταν ένα αυτόματο στοίβας απεικονίζεται με διάγραμμα, όπως αυτό του Σχήματος Γ'.1 έχουμε στα τόξα των μεταβάσεων επιγραφές της μορφής $x, y; z$ που έχουν την εξής σημασία: αν η είσοδος είναι ένα x και μπορούμε να εξάγουμε ένα y από τη στοίβα, τότε εκτέλεσε τη μετάβαση στη νέα κατάσταση, αφαίρεσε το y από τη στοίβα και βάλε σ'αυτήν το z . Αν αντί για y εμφανίζεται στη μεσαία θέση το λ , τότε για τη μετάβαση δε λαμβάνεται υπόψη το περιεχόμενο της στοίβας και αντίστοιχα αν στη θέση του y εμφανίζεται το λ , τότε η στοίβα δε μεταβάλλεται. Στο Σχήμα απεικονίζεται το αυτόματο στοίβας, που αποδέχεται τη γλώσσα L_1 .



Σχήμα Γ'.2: Ένα απλό αυτόματο στοίβας για τη γλώσσα $\{a^n b^n : n \geq 0\}$

Η κλάση των γλωσσών, που γίνονται αποδεκτές από ένα αυτόματο στοίβας είναι η κλάση των *γλωσσών χωρίς συμφραζόμενα*. Πολλές τεχνητές γλώσσες απάυτες που χρησιμοποιούμε καθημερινά στην Επιστήμη των Υπολογιστών είναι γλώσσες χωρίς συμφραζόμενα. Αυτές περιλαμβάνουν όλες τις γλώσσες προγραμματισμού, τις γλώσσες ερωτημάτων, τις γλώσσες επισημείωσης κ.α.

Τα αυτόματα στοιβάς όμως ως μοντέλο υπολογισμού έχει αποδειχθεί ότι δεν είναι αρκετά εκφραστικό για να αναπαραστήσει υπολογισμούς, όπως αυτούς των προγραμμάτων που γράφουμε. Ας εξετάσουμε για παράδειγμα την περίπτωση της γλώσσας

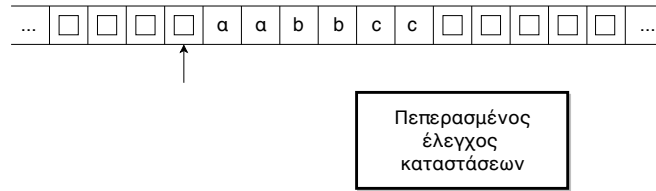
$$L_2 = \{a^n b^n c^n : n \geq 0\} \quad (\Gamma.6)$$

για την οποία εύκολα θα μπορούσαμε να γράψουμε ένα πρόγραμμα, που θα καταμετρά τον αριθμό των a , των b και των c στις συμβολοσειρές και αν αυτοί είναι ίσοι και οι χαρακτήρες είναι στη σωστή διάταξη, τότε να αποφαινεται ότι η συμβολοσειρά εισόδου είναι ένα στοιχείο της γλώσσας L_2 . Για να αντιληφθούμε το γιατί ένα αυτόματο στοιβάς είναι αδύνατο να εκτελέσει έναν τέτοιο υπολογισμό, ας δούμε το πώς θα μπορούσε να χρησιμοποιηθεί η μνήμη της στοιβάς, για να αναγνωρίζονται οι συμβολοσειρές της γλώσσας L_2 . Αν λοιπόν είχαμε ένα κατάλληλα ορισμένο αυτόματο στοιβάς, όπως αυτό του σχήματος Γ.2 που δέχεται τη γλώσσα L_2 , τότε θα μπορούσαμε να χρησιμοποιήσουμε τη στοιβα με ανάλογο τρόπο για την καταμέτρηση των a , όπως και στην περίπτωση της L_1 και θα μπορούσαμε να αφαιρούμε σύμβολα από τη στοιβα, καθώς διαβάζονται τα b που ακολουθούν και έτσι να τα συγκρίνουμε με τον αριθμό των a που προηγήθηκαν. Τότε όμως δε θα μπορούσαμε να εξετάσουμε τα c στις συμβολοσειρές της L_2 , καθώς αν ο αριθμός των a είναι ίσος με τον αριθμό των b , θα έχει χαθεί από τη στοιβα κάθε πληροφορία γι'αυτά, αφού αυτή θα είναι άδεια. Η L_2 είναι μια γλώσσα, για την οποία έχει αποδειχθεί ότι δεν υπάρχει αυτόματο στοιβάς, που αν την αναγνωρίζει.

Γ'.2.1.3 Μηχανές Turing

Το πρόβλημα της αναγνώρισης γλωσσών, όπως η L_2 μπορεί να αντιμετωπιστεί από ένα πιο εκφραστικό μοντέλο υπολογισμού, αυτό που πρότεινε ο Alan Turing το 1936 και είναι σήμερα γνωστό ως *μηχανή Turing*. Η πρόταση του Turing για ένα μοντέλο υπολογισμού ήταν ένα παράπλευρο αποτέλεσμα στην προσπάθεια του να δώσει απάντηση στο γνωστό τότε ως "πρόβλημα απόφασης", το Entscheidungsproblem, όπως είχε διατυπωθεί από τον David Hilbert το 1928 σ'ένα μικρό εγχειρίδιο λογικής, που είχε δημοσιευτεί μαζί με τον μαθητή του Wilhelm Ackermann. Το Entscheidungsproblem έθεσε το ζήτημα της ύπαρξης ενός αλγορίθμου με είσοδο έναν ισχυρισμό σε λογική πρώτης τάξης (και έναν πεπερασμένο αριθμό αξιωμάτων εκτός απάντά της λογικής), που να αποφαινεται με ένα "ΝΑΙ" ή "ΟΧΙ" για το αν ο ισχυρισμός είναι καθολικά έγκυρος, είναι δηλαδή αληθής για κάθε δομή που ικανοποιεί τα αξιώματα. Ο Turing δεν επικεντρώθηκε στα βήματα του ζητούμενου αλγορίθμου, αλλά σε μερικές βασικές ενέργειες, που γίνονται κατά την εκτέλεση κάθε τέτοιου υπολογισμού. Στη συνέχεια, απέδειξε ότι καμία μηχανή δε θα μπορούσε εκτελώντας μόνο αυτές τις βασικές ενέργειες να καθορίσει εάν ένα δοθέν συμπέρασμα προκύπτει από τις δοθείσες προϋποθέσεις. Έτσι, κατέληξε στο συμπέρασμα ότι δεν υπάρχει αλγόριθμος για το Entscheidungsproblem.

Το αποτέλεσμα αυτό σχετίζεται- αν και όχι εμφανώς- με το Πρώτο Θεώρημα της μη Πληρότητας που είχε αποδείξει με ένα λογικό παίγνιο ο Kurt Godel σε ηλικία μόλις 25 ετών το 1931. Οι μηχανές Turing δεν έχουν στοίβα, αλλά όπως φαίνεται στο Σχήμα Δ'.7 η λειτουργία της υποκαθίσταται από μια ταινία, που επεκτείνεται απείρως σε μήκος. Πάνω στην ταινία

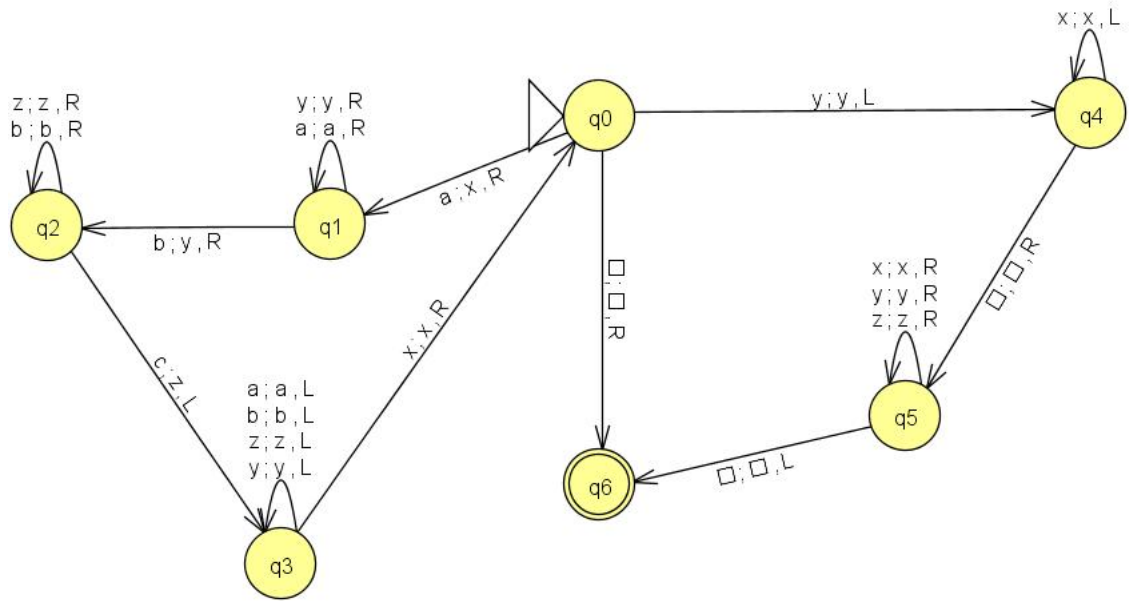


Σχήμα Γ'.3: Απεικόνιση της λειτουργίας μιας μηχανής Turing

μετακινείται μια κεφαλή ανάγνωσης/ εγγραφής και κατά τη μετάβαση από μια κατάσταση σε μια άλλη επιτρέπεται η μετακίνηση της κεφαλής κατά μια θέση προς την κατεύθυνση που περιγράφει η μετάβαση (δεξιά ή αριστερά). μόνο το σύμβολο που βρίσκεται στην τρέχουσα θέση της κεφαλής είναι δυνατό να προσπελάσσει και- σε αντίθεση με τα μοντέλα υπολογισμού που περιγράψαμε- μπορεί να αλλάξει, αν η μετάβαση περιγράφει μια εγγραφή την τρέχουσα θέση. Η συμβολοσειρά εισόδου βρίσκεται αρχικά τοποθετημένη πάνω στην ταινία και η μηχανή ξεκινάει τον υπολογισμό με την κεφαλή τοποθετημένη στη θέση του πρώτου της χαρακτήρα. Σε κάθε βήμα, μια μηχανή Turing M_T καθορίζει τη μετάβαση στην επόμενη κατάσταση με βάση την τρέχουσα μετάβαση και το χαρακτήρα στην τρέχουσα θέση της κεφαλής, που σημειώνεται με ένα τόξο. Η μετάβαση, που θα εκτελεστεί προσδιορίζει το χαρακτήρα που θα εγγραφεί στη θέση της κεφαλής, η οποία στη συνέχεια θα μετακινηθεί μια θέση (αριστερά ή δεξιά). Η συμβολοσειρά εισόδου μπορεί να καταλαμβάνει μόνο ένα πεπερασμένο σε μήκος μέρος της ταινίας, ενώ στις υπόλοιπες θέσεις της ταινίας βρίσκεται χαρακτήρας "κενό". Σημειώνουμε ότι η μηχανή M_T μπορεί να μετακινεί την κεφαλή έξω από τα όρια της συμβολοσειράς και βέβαια να εγγράφει στις κενές θέσεις της ταινίας. Όταν μια μηχανή Turing απεικονίζεται με διάγραμμα, όπως αυτό του Σχήματος Γ'.4, έχουμε στα τόξα των μεταβάσεων επιγραφές της μορφής $j : k, l$, που έχουν την εξής σημασία:

- το j αναφέρεται στο σύμβολο που βρίσκεται στην τρέχουσα θέση της κεφαλής ανάγνωσης/εγγραφής,
- το k αναφέρεται στο σύμβολο που θα αντικαταστήσει στην ταινία το πρώτο αναφερόμενο σύμβολο και
- το l εκφράζει την κατεύθυνση κατά την οποία θα κινηθεί η κεφαλή.

Ο χαρακτήρας R δηλώνει μετακίνηση μια θέση προς τα δεξιά, ενώ ο χαρακτήρας L δηλώνει μετακίνηση μια θέση προς τα αριστερά. Η μηχανή Turing του Σχήματος Γ'.4, αποδέχεται τις συμβολοσειρές της γλώσσας L_2 εκτελώντας τον εξής υπολογισμό. Ευρισκόμενη η κεφαλή



Σχήμα Γ'.4: μια μηχανή Turing για τη γλώσσα $\{a^n b^n c^n : n \geq 0\}$

στην πρώτη θέση της ταινίας από αριστερά με χαρακτήρα διάφορο του “κενού” εκτελεί μετάβαση αν ο χαρακτήρας αυτός είναι a αντικαθιστώντας τον με x και σπυρτή την περίπτωση μετακινείται δεξιά. Αν εκεί υπάρχει ένα ή στη συνέχεια και άλλα a , τότε τα προσπερνάει μέχρι να βρεί το πρώτο b . αντικαθιστά το b με το χαρακτήρα y και προσπερνά τα υπόλοιπα b - αν υπάρχουν- μέχρι να βρεί το πρώτο c . Αυτό αλλάζει σε z και η κεφαλή κινείται πλέον αριστερά, μέχρι να βρεί το x , που αρχικά έγραψε. Τότε χωρίς να αλλάξει το x μετακινείται μια θέση προς τα δεξιά. Εκεί είτε θα βρει ένα επιπλέον a και θα επαναλάβει τον κύκλο (όσες φορές χρειάζεται) αναζητώντας (για κάθε τέτοιο a) ένα b και ένα c στη συνέχεια της συμβολοσειράς, είτε θα βρεί ένα y απάντά που έχουν αντικαταστήσει τα b της συμβολοσειράς. Στη δεύτερη περίπτωση, η κεφαλή επανέρχεται στην αρχή και τελικά ολοκληρώνει τον υπολογισμό της κινούμενη προς τα δεξιά και προσπερνώντας όλους τους χαρακτήρες x, y, z . Τελικά προσεγγίζεται η κατάσταση αποδοχής μόνο αν η συμβολοσειρά εισόδου για κάθε a που εμφανίζεται στο πρώτο τμήμα έχει ένα b και ένα c στο δεύτερο και στο τρίτο τμήμα της αντίστοιχα. Τα πεπερασμένα αυτόματα και οι μηχανές στοιβάς εκτελούν υπολογισμούς, που τερματίζουν πάντα, αφού πρώτα έχουν καταναλώσει όλους τους χαρακτήρες συμβολοσειράς εισόδου. Από την άλλη, οι μηχανές Turing μπορεί και να μην τερματίζουν. Καθώς η συμβολοσειρά βρίσκεται στην ταινία σε όλη τη διάρκεια του υπολογισμού και η κεφαλή μπορεί να κινείται προς τα δεξιά και αντίθετα για πολλές φορές, θα μπορούσε αυτό να γίνεται για πάντα. Θα μπορούσε επίσης η κεφαλή να κινείται συνεχώς προς μια συγκεκριμένη κατεύθυνση και τελικά συνεχίζει την κίνησή της για πάντα στο μέρος της ταινίας, όπου υπάρχουν μόνο

“κενά”. Η ευελιξία, που χαρακτηρίζει τις μηχανές Turing στον τρόπο χρήσης της ταινίας είναι η αιτία της αυξημένης εκφραστικότητάς τους σε σχέση με τα πεπερασμένα αυτόματα και αυτόματα στοίβας. Στην πραγματικότητα έχει αποδειχθεί ότι *με μια κατάλληλη μηχανή Turing είναι δυνατό να αναπαρασταθεί ο οποιοσδήποτε υπολογισμός, που μπορεί να γραφεί σε μια γλώσσα προγραμματισμού*. Πρέπει όμως να έχουμε πάντα υπόψη μας ότι δεν υπάρχει εγγύηση τερματισμού του υπολογισμού μιας μηχανής Turing και επιπλέον ότι έχει δυστυχώς αποδειχθεί ότι *δεν υπάρχει αλγόριθμος, που να αποφασίζει αν μια μηχανή Turing τερματίζει για κάποια ή για όλες τις πιθανές εισόδους της*. Αυτό το θεμελιώδες αποτέλεσμα, που αποκαλύπτει τα όρια των εκτελέσιμων υπολογισμών είναι γνωστό ως *πρόβλημα του τερματισμού*. Με βάση τις μηχανές Turing ορίζονται δύο νέες κλάσεις γλωσσών:

- μια γλώσσα L είναι *αποφασίσιμη ή αναδρομική*, αν και μόνο αν υπάρχει μηχανή Turing M_T με υπολογισμό, που τερματίζει για όλες τις εισόδους, αποδέχεται όλες τις συμβολοσειρές της L και απορρίπτει όλες τις συμβολοσειρές, που δεν ανήκουν στην L .
- μια γλώσσα L είναι *ημιαποφασίσιμη ή αναδρομικά απαριθμήσιμη*, αν και μόνο αν υπάρχει μηχανή Turing M_T , που αποδέχεται όλες τις συμβολοσειρές της L , χωρίς να αποδέχεται τις συμβολοσειρές που δεν ανήκουν στην L . Ειδικότερα για τις συμβολοσειρές της δεύτερης περίπτωσης, η M_T μπορεί είτε να τις απορρίπτει, είτε να εκτελεί υπολογισμό χωρίς να τερματίζει.

Η L_2 είναι αναδρομική γλώσσα και ισχύει ότι κάθε αναδρομική γλώσσα είναι επίσης και αναδρομικά απαριθμήσιμη. Από την άλλη, υπάρχουν γλώσσες, που είναι αναδρομικά απαριθμήσιμες αλλά όχι αναδρομικές. μία τέτοια περίπτωση είναι η γλώσσα:

$$L_3 = \{ \langle p, w \rangle : p \text{ είναι ένα } C \text{ πρόγραμμα που τερματίζει για είσοδο } w \} \quad (\Gamma.7)$$

Η L_3 ημιαποφασίζεται από μία μηχανή Turing, που προσομοιώνει την εκτέλεση του προγράμματος p με είσοδο w . Υπάρχει περίπτωση η προσομοίωση να τερματίσει, οπότε ξέρουμε ότι το p τερματίζει για είσοδο w , αλλά υπάρχει και η περίπτωση να μην τερματίσει, οπότε δε θα ήμαστε σε θέση να ξέρουμε αν το p τερματίζει. Σε τελική ανάλυση, δεν υπάρχει μηχανή Turing, που να αναγνωρίζει ότι ένα πρόγραμμα p δεν τερματίζει για είσοδο w , όπως και δεν υπάρχει αλγόριθμος, που θα εξετάζει ένα C πρόγραμμα και θα καθορίζει αν αυτό τερματίζει.

Η εργασία του Alan Turing πάνω σε αφαιρετικά μοντέλα υπολογισμού είχε μεγάλη επίδραση στη σχεδίαση των ηλεκτρονικών υπολογιστών και αργότερα στη σχεδίαση γλωσσών προγραμματισμού γι' αυτούς τους υπολογιστές. Όλα τα βασικά συστατικά των μηχανών Turing υπάρχουν στην αρχιτεκτονική von Neumann, που υιοθετήθηκε στους σύγχρονους υπολογιστές: μία μνήμη, που μπορούμε να την αντιληφθούμε ως άπειρη σε μέγεθος αν σ' αυτή συμπεριλάβουμε εκτός από την RAM και το χώρο αποθήκευσης των δίσκων και των άλλων περιφερειακών και μία μονάδα ελέγχου, που διαχειρίζεται την εκτέλεση των προγραμμάτων στη μηχανή. Έτσι, μπορούμε να δούμε τους σύγχρονους υπολογιστές ως μία

υλοποίηση της Γενικής μηχανής Turing, που μπορεί να εκτελεί τον υπολογισμό οποιασδήποτε μηχανής Turing, για τυχαία είσοδο. Η μνήμη είναι ένα από τα βασικά συστατικά του υπολογιστή και αποθηκεύει εντολές και δεδομένα, ενώ το άλλο βασικό συστατικό του είναι ο επεξεργαστής. Οι πρώτες γλώσσες προγραμματισμού σχεδιάστηκαν με βάση τη συγκεκριμένη φυσική σχεδίαση για τη μηχανή υπολογισμού. Στις γλώσσες, που εξελίχθηκαν απ' αυτές και έχει επικρατήσει να λέγονται γλώσσες προστακτικού προγραμματισμού μπορεί κανείς να διακρίνει ακόμη και σήμερα την επιρροή της συγκεκριμένης αρχιτεκτονικής υπολογιστή. Οι προστακτικές γλώσσες είναι αφαίρεση της υποκείμενης αρχιτεκτονικής von Neumann με την έννοια ότι σ' αυτές διατηρούνται τα απαραίτητα μέρη της, ενώ ταυτόχρονα αγνοούνται οι περιττές λεπτομέρειες. Οι χαμηλού επιπέδου γλώσσες παρέχουν ένα περιορισμένο βαθμό αφαίρεσης, ενώ τις υψηλού επιπέδου γλώσσες μπορούμε να τις δούμε ως μία ιδεατή μηχανή, στην οποία η διαχείριση της μνήμης είναι γενικά διάφανη για τον προγραμματιστή και τα θεμελιώδη στοιχεία εισόδου/εξόδου δεν εξαρτώνται από το υλικό.

Παρόλο που το επίπεδο της αφαίρεσης, που παρέχουν οι προστακτικές γλώσσες διαφοροποιείται σημαντικά ξεκινώντας από τις συμβολικές γλώσσες μηχανής μέχρι και τις πιο εξελιγμένες γλώσσες όπως η Java, σήμερα μπορούμε να διακρίνουμε κοινά χαρακτηριστικά στη σχεδίαση όλων των προστακτικών γλωσσών, που τελικά αντανακλούν την υποκείμενη αρχιτεκτονική μηχανής. Η μνήμη και ο επεξεργαστής, τα κοινά συστατικά της κάθε υπολογιστικής μηχανής, αντιπροσωπεύονται σ' ένα αφαιρετικό επίπεδο μέσα σε μία υψηλού επιπέδου προστακτική γλώσσα από μεταβλητές, που αναπαριστούν το χώρο μνήμης, μαζί με εντολές εκχώρησης που μεταβάλλουν τα περιεχόμενά της, και δομές ελέγχου, που υποδεικνύουν τη σειρά εκτέλεσης των εντολών από τον επεξεργαστή.

Εκτός από τη σχεδίαση της αρχιτεκτονικής των υπολογιστών και των πρώτων προστακτικών γλωσσών, η επίδραση της δουλειάς του Turing επεκτείνεται και στις αφαιρετικές μηχανές καταστάσεων, που χρησιμοποιούνται για τον ακριβή ορισμό της σημασίας των γλωσσικών στοιχείων των προστακτικών γλωσσών. Αυτές οι αφαιρετικές μηχανές εκτέλεσης στηρίζονται στην έννοια της μετάβασης από κατάσταση σε κατάσταση. Είναι αξιοσημείωτο ότι η πρώτη προσέγγιση για μία τυπική περιγραφή της συμπεριφοράς των γλωσσικών στοιχείων μιας γλώσσας έγινε με τη χρήση μιας αφαιρετικής μηχανής καταστάσεων ή ακριβέστερα ενός συστήματος μεταβάσεων, που περιγράφει ένα διερμηνευτή για τη γλώσσα προγραμματισμού.

Γ'.2.1.4 Λογισμός-λ

Ο *Λογισμός-λ* είναι ένα μοντέλο υπολογισμού, που στηρίζεται στην ιδέα ότι μπορούμε να δούμε τους αλγορίθμους ως μαθηματικές συναρτήσεις, που αντιστοιχίζουν τις εισόδους στις εξόδους τους. Προτάθηκε από τον Alonzo Church στη δεκαετία του '30 ως συμβολισμός για μία θεωρία ανώνυμων συναρτήσεων. Το όνομα του λογισμού περιέχει τον ελληνικό

χαρακτήρα λ , που χρησιμοποιείται σε συναρτησιακές εκφράσεις. Ο Church παρατήρησε ότι όταν αναπαριστούμε μία συνάρτηση από μία έκφραση όπως η $x + y$, δεν είναι ξεκάθαρο ποια ακριβώς είναι η οριζόμενη συνάρτηση. μία τέτοια έκφραση θα μπορούσε για παράδειγμα να έχει την οποιαδήποτε από τις παρακάτω ερμηνείες:

1. ο αριθμός $x + y$, όπου x και y κάποιοι δοθέντες αριθμοί,
2. η συνάρτηση $f : x \mapsto x + y$, που αντιστοιχεί σ' έναν αριθμό x τον αριθμό $x + y$ για κάποια προκαθορισμένη τιμή y ,
3. η συνάρτηση $g : y \mapsto x + y$, που αντιστοιχεί σε μία είσοδο y τον αριθμό $x + y$ για μία προκαθορισμένη τιμή x ή
4. η συνάρτηση $h : x, y \mapsto x + y$, που δέχεται ως παραμέτρους τα x και y και παράγει ως αποτέλεσμα την τιμή $x + y$.

Από τις παραπάνω ερμηνείες είναι σαφές ότι η χρήση αριθμητικών εκφράσεων για την αναπαράσταση συναρτήσεων είναι πηγή ασάφειας και γι' αυτό ο Church πρότεινε τη χρήση ενός συμβολισμού, που δίνει έμφαση στη διάκριση μεταξύ των μεταβλητών, που χρησιμοποιούνται ως παράμετροι και των μεταβλητών, που αναπαριστούν προκαθορισμένες τιμές. Στο συμβολισμό αυτό, για μία συνάρτηση με μία παράμετρο x προηγείται το σύμβολο λ και η μεταβλητή x . Για παράδειγμα, η συνάρτηση $f : x \mapsto x + y$, που αντιστοιχεί σε μία είσοδο x τον αριθμό $x + y$ για κάποια προκαθορισμένη τιμή y γράφεται ως $\lambda x.x + y$. Πιο συγκεκριμένα, οι συναρτήσεις που αναφέρθηκαν παραπάνω διακρίνονται εύκολα μεταξύ τους με τη χρήση του λογισμού- λ ως εξής:

- ο αριθμός $x + y$ γράφεται ως $x + y$,
- η συνάρτηση $f : x \mapsto x + y$ γράφεται ως $\lambda x.x + y$,
- η συνάρτηση $g : y \mapsto x + y$ γράφεται ως $\lambda y.x + y$ και
- η συνάρτηση $h : x, y \mapsto x + y$ γράφεται ως $\lambda xy.x + y$

Το βασικό κίνητρο της δουλειάς του Church, όπως και στην περίπτωση του Turing, ήταν η τυπική περιγραφή της έννοιας του αλγορίθμου για να καταστεί εφικτή η επίλυση κάποιων από τα ανοικτά προβλήματα, που πρότεινε ο Hilbert στο συνέδριο των μαθηματικών το 1900. Εκτός όμως από ένα εργαλείο για την ανάλυση προβλημάτων υπολογισιμότητας, ο λογισμός- λ εξελίχθηκε πρόσφατα σ' ένα ιδιαίτερα χρήσιμο εργαλείο για τα εξής:

- τον ορισμό της σημασίας των γλωσσών προγραμματισμού,
- τη μελέτη τεχνικών υλοποίησης συναρτησιακών γλωσσών προγραμματισμού,
- την κωδικοποίηση αποδείξεων σε μια σειρά συστημάτων λογικής και

- τη σχεδίαση εργαλείων για την απόδειξη θεωρημάτων και την υποβοήθηση αποδείξεων.

Ο λογισμός-λ είναι ένα *κατά Turing πλήρες* μοντέλο υπολογισμού, που σημαίνει ότι έχει την ίδια υπολογιστική ισχύ με τις μηχανές Turing.

Γ'.2.2 Κλάσεις γλωσσών ως προς το βαθμό αποδοτικότητας της διαδικασίας απόφασης

Για τις αποφασίσιμες γλώσσες, όπως τις ορίσαμε, υπάρχει καταρχήν μηχανή Turing, που τερματίζει για όλες τις εισόδους αποδεχόμενη τις συμβολοσειρές της γλώσσας, αν και μπορεί για πολύ μεγάλες εισόδους να απαιτούνται περισσότερα βήματα απ' όσα είναι δυνατό να εκτελεστούν στο διάστημα από την εποχή της μεγάλης Έκρηξης για τη δημιουργία του σύμπαντος. Αξίζει επομένως να εξετάσουμε την κλάση των αποφασίσιμων γλωσσών από μία διαφορετική σκοπιά, που έχει να κάνει με τους πόρους σε χρόνο και χώρο, που απαιτούνται για τις πλέον προηγμένες διαδικασίες απόφασης, που έχουμε. Ορίζουμε τις εξής κλάσεις γλωσσών:

- P , που περιλαμβάνει τις γλώσσες που αποφασίζονται σε χρόνο που μεταβάλλεται σε πολυωνυμική συνάρτηση του μήκους της εισόδου,
- NP , που περιλαμβάνει τις γλώσσες που αποφασίζονται από μηχανή με πολλές επιλογές σε κάθε βήμα (μη ντετερμινιστικές), τέτοιες ώστε ο χρόνος που απαιτείται για τη διερεύνηση μιας ακολουθίας επιλογών (μο- νοπάτι) να μεταβάλλεται σε πολυωνυμική συνάρτηση του μήκους της εισόδου και
- $PSPACE$, που περιλαμβάνει τις γλώσσες, που αποφασίζονται από μηχανή με απαιτήσεις χώρου, που μεταβάλλονται σε πολυωνυμική συνάρτηση του μήκους της εισόδου.

Κλάσεις γλωσσών, όπως αυτές που ορίσαμε ως προς συγκεκριμένα είδη μηχανών μπορούν να διαταχθούν σε μία ιεραρχία. Έτσι, για τις γλώσσες που αναφέραμε ισχύει ότι:

$$P \subseteq NP \subseteq PSPACE \quad (\Gamma'.8)$$

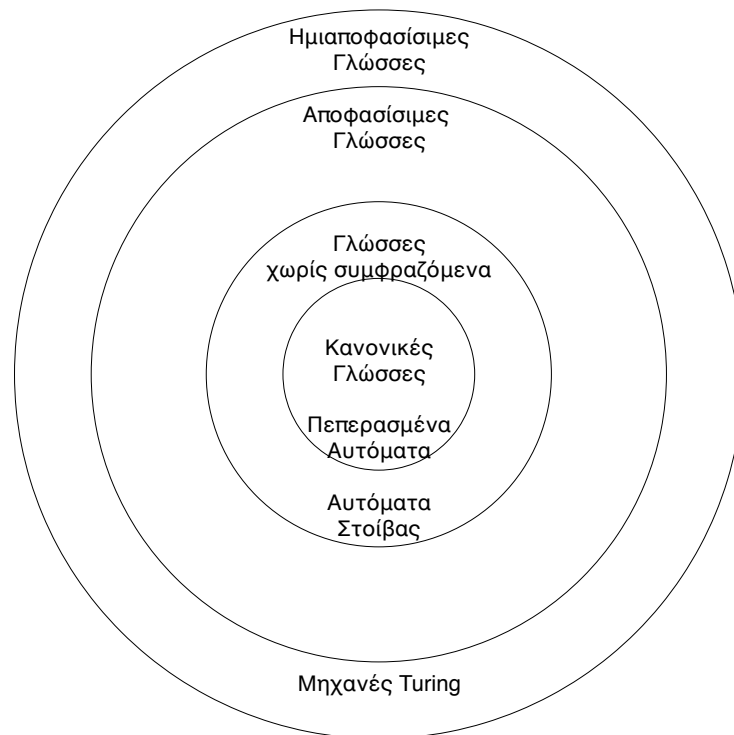
Τα θεωρητικά προβλήματα που σχετίζονται με τη συγκεκριμένη ιεραρχία γλωσσών συνεχίζουν να αποτελούν ένα σημαντικό πεδίο έρευνας. Για παράδειγμα, ένα σημαντικό ανοικτό ζήτημα είναι αν ισχύει η $P = NP$. Είναι δυνατό, αν και μάλλον απίθανο, ότι κάθε γλώσσα της κλάσης NP είναι επίσης και στην P .

Γ'.3 Κατάταξη γλωσσών

Έχουμε ορίσει τέσσερις διαφορετικές κλάσεις γλωσσών σε σχέση με τον τύπο μηχανής, που χρησιμοποιείται, για την αναγνώρισή τους. Αυτές είναι οι:

- Κανονικές γλώσσες, που γίνονται αποδεκτές από κατάλληλα σχεδιασμένες μηχανές πεπερασμένων καταστάσεων.
- Γλώσσες χωρίς συμφραζόμενα, που γίνονται αποδεκτές από κατάλληλα αυτόματα στοίβας.
- Αποφασίσιμες γλώσσες, που η κάθε μία αποφασίζεται από μία μηχανή Turing, που πάντα τερματίζει.
- Ημιαποφασίσιμες γλώσσες, που ημιαποφασίζονται από κάποια μηχανή Turing, που τερματίζει για όλες τις συμβολοσειρές της γλώσσας.

Κάθε μία από τις κλάσεις γλωσσών αυτής της λίστας είναι γνήσιο υποσύνολο της επόμενης κλάσης στη λίστα, όπως εξάλλου φαίνεται και στο Σχήμα της *Ιεραρχίας Γλωσσών κατά Chomsky*. Καθώς λοιπόν εξετάζουμε τις διαφορετικές κλάσεις γλωσσών στην ιεραρχία από



Σχήμα Γ'.5: *Ιεραρχία γλωσσών κατά Chomsky*

μέσα προς τα έξω έχουμε πρόσβαση σε εργαλεία με ολοένα και μεγαλύτερη εκφραστική ισχύ. Έτσι, μπορούμε για παράδειγμα να περιγράψουμε τη γλώσσα $\{a^n b^n : n \geq 0\}$ ως μία γλώσσα χωρίς συμφραζόμενα, αλλά δε μπορούμε να την περιγράψουμε ως κανονική γλώσσα. Επίσης η γλώσσα $\{a^n b^n c^n : n \geq 0\}$ μπορεί να περιγραφεί ως αποφασίσιμη γλώσσα, αλλά δε μπορεί να εκφραστεί ως γλώσσα χωρίς συμφραζόμενα ή ως κανονική γλώσσα. Η μεγαλύτερη εκφραστικότητα όμως συνοδεύεται από ένα σχετικό κόστος, που μπορεί να αφορά την:

- Υπολογιστική αποδοτικότητα. Οι μηχανές πεπερασμένων καταστάσεων εκτελούνται σε χρόνο, που είναι γραμμική συνάρτηση του μήκους της συμβολοσειράς εισόδου. Η εκτέλεση ενός αυτόματου στοιβάς απαιτεί χρόνο, που μεταβάλλεται ανάλογα προς τον κύβο του μήκους της συμβολοσειράς εισόδου. Η εκτέλεση μιας μηχανής Turing μπορεί να απαιτεί χρόνο, που μεταβάλλεται εκθετικά (ή πιο γρήγορα) ως προς το μήκος της συμβολοσειράς εισόδου.
- Αποφασισιμότητα. Υπάρχουν διαδικασίες που δίνουν απάντηση σε πολύ σημαντικά ζητήματα σχετικά με τις μηχανές πεπερασμένων καταστάσεων. Για παράδειγμα, μπορεί να απαντηθεί το ερώτημα «αν μία συγκεκριμένη μηχανή πεπερασμένων καταστάσεων αποδέχεται μία δοθείσα συμβολοσειρά» ή «αν μία μηχανή πεπερασμένων καταστάσεων είναι ελάχιστη, δηλαδή η απλούστερη δυνατή για την αναπαράσταση της γλώσσας την οποία ορίζει». Για τα αυτόματα στοιβάς μπορεί να απαντηθεί μόνο ένα υποσύνολο απ' αυτά τα ερωτήματα, ενώ για τις μηχανές Turing δεν υπάρχει απάντηση για κανένα απ' αυτά.
- Σαφήνεια. Σήμερα υπάρχουν πολλά εργαλεία, που υποστηρίζουν τη σχεδίαση και την ανάλυση μηχανών πεπερασμένων καταστάσεων, με το *JFLAP* να είναι ένα απ' αυτά. Κάθε κανονική γλώσσα μπορεί επίσης να περιγραφεί από έναν παραγωγό γλώσσας γνωστό ως κανονική έκφραση. Κάθε γλώσσα χωρίς συμφραζόμενα, εκτός του ότι γίνεται αποδεκτή από κάποιο αυτόματο στοιβάς, μπορεί επίσης να περιγραφεί από μία γραμματική χωρίς συμφραζόμενα. Για πολλές περιπτώσεις γλωσσών οι γραμματικές χωρίς συμφραζόμενα είναι ένα επαρκές εργαλείο τεκμηρίωσής τους. Δυστυχώς δεν υπάρχουν αντίστοιχα εργαλεία για τις ευρύτερες κλάσεις των αποφασίσιμων και ημιαποφασίσιμων γλωσσών.

Κατά συνέπεια, για θεωρητικούς και πρακτικούς λόγους επιδιώκουμε πάντα ένα συγκεκριμένο πρόβλημα αναπαράστασης γλώσσας να το περιγράψουμε με τον εκφραστικά ασθενέστερο φορμαλισμό, που επαρκεί για τη διατύπωσή του.

Γ'.4 Θέση των Church και Turing

Η Θέση των Church και Turing αφορά την έννοια μιας μηχανιστικής μεθόδου υπολογισμού, αυτού δηλαδή που διαισθητικά αντιλαμβανόμαστε ως αλγόριθμο. μία μέθοδος ή διαδικασία M , για την επίτευξη ενός επιθυμητού αποτελέσματος χαρακτηρίζεται ως «μηχανιστική» ή «αποτελεσματική» μόνο στην περίπτωση που :

1. Η M είναι ένα σύνολο από πεπερασμένο αριθμό καλά καθορισμένων εντολών (κάθε εντολή εκφράζεται από ένα πεπερασμένο αριθμό συμβόλων)
2. Αν η M εκτελεστεί χωρίς λάθος, τότε θα παράξει το επιθυμητό αποτέλεσμα μετά από ένα πεπερασμένο αριθμό βημάτων

3. Η M μπορεί καταρχήν και στην πράξη να εκτελεστεί από άνθρωπο χωρίς τη βοήθεια κάποιου μηχανισμού και χωρίς περιορισμούς πόρων
4. Η M δεν απαιτεί κάποια ιδιαίτερη γνώση ή εφευρετικότητα από τον άνθρωπο που την εκτελεί.

Ένα γνωστό παράδειγμα «μηχανιστικού» υπολογισμού είναι η δοκιμή για ταυτολογία με πίνακα τιμών αληθείας. Στην πράξη βέβαια αυτή η δοκιμή δε δουλεύει για τύπους που περιέχουν μεγάλο αριθμό προτασιακών μεταβλητών, αλλά βασικά είναι εφαρμόσιμη σε οποιονδήποτε τύπο του προτασιακού λογισμού, αν υπάρχει επαρκής χρόνος, υπομονή, χαρτί και μολύβια.

Η ύπαρξη μηχανιστικής μεθόδου για την επίτευξη ενός αποτελέσματος συνήθως εκφράζεται λέγοντας ότι υπάρχει ένας τρόπος εύρεσης των τιμών μιας υπολογίσιμης συνάρτησης. Για παράδειγμα, αν υπάρχει μηχανιστική μέθοδος που καθορίζει το αν ένας τύπος προτασιακής λογικής (δεν) είναι ταυτολογία, αυτό εκφράζεται λέγοντας ότι υπάρχει μηχανιστική μέθοδος υπολογισμού των τιμών μιας συνάρτησης T με πεδίο ορισμού το σύνολο των έγκυρων τύπων προτασιακής λογικής και τιμή για ένα δοθέν τύπο x , που γράφεται ως $T(x)$ και είναι 1 ή 0 ανάλογα με το αν το x είναι ταυτολογία ή όχι.

Η έννοια της μηχανιστικής μεθόδου δεν ορίζεται αυστηρά και όλες οι προσπάθειες προς αυτή την κατεύθυνση αφήνουν δίχως ερμηνεία τη βασική απαίτηση ότι δεν απαιτείται κάποια ιδιαίτερη γνώση ή εφευρετικότητα.

Ένα από τα επιτεύγματα των Church και Turing ήταν ότι με τις εργασίες τους παρουσίασαν δύο μαθηματικά διατυπωμένες προτάσεις, που μπορούν να αντικαταστήσουν την άτυπη δήλωση «μπορεί να υπολογιστεί από μία μηχανιστική μέθοδο». Βέβαια, οι δύο προτάσεις ήταν πολύ διαφορετικές μεταξύ τους, αλλά τελικά αποδείχθηκε ότι είναι ισοδύναμες, με την έννοια ότι η κάθε μία τελικά αναφέρεται στο ίδιο σύνολο υπολογίσιμων συναρτήσεων.

Η Θέση των Church και Turing είναι ένας ισχυρισμός ότι το σύνολο αυτών των συναρτήσεων περιλαμβάνει κάθε συνάρτηση, που ικανοποιεί τις παραπάνω προϋποθέσεις για να χαρακτηρίζεται μηχανιστική (βεβαίως αν υπήρχαν συναρτήσεις για τις οποίες οι άτυπες προϋποθέσεις ήταν αληθείς αλλά όχι οι μαθηματικά διατυπωμένες προτάσεις, τότε οι δεύτερες θα ήταν λιγότερο γενικές και δε θα μπορούσαν να αντικαταστήσουν τις άτυπες προϋποθέσεις).

Η τυπικά ορισμένη έννοια, που προτάθηκε από τον Turing ήταν η υπολογισιμότητα από μία μηχανή Turing. Υποστήριζε (Θέση του Turing) ότι «*όταν υπάρχει μηχανιστική μέθοδος για την εύρεση των τιμών μιας υπολογίσιμης συνάρτησης, η συνάρτηση αυτή μπορεί να υπολογιστεί από μία μηχανή Turing*». μπορεί βεβαίως εύκολα να θεμελιωθεί και ο αντίστροφος ισχυρισμός

για το ότι ένα πρόγραμμα μηχανής Turing είναι το ίδιο του μία προδιαγραφή μηχανιστικής μεθόδου: χωρίς να επιστρατεύεται ιδιαίτερη γνώση ή εφευρετικότητα μπορεί ένας άνθρωπος να επεξεργαστεί τις εντολές ενός προγράμματος και να εκτελέσει τις απαιτούμενες ενέργειες. Αν η Θέση του Turing είναι σωστή, τότε η συζήτηση για την (μη) ύπαρξη μηχανιστικών μεθόδων στο χώρο των μαθηματικών, της Λογικής και της Επιστήμης των Υπολογιστών μπορεί να αντικατασταθεί από την (μη) ύπαρξη προγραμμάτων μηχανής Turing.

Ο Church χρησιμοποίησε την άτυπη έννοια του «μηχανιστικά υπολογίσιμου» για να δείξει ότι υπάρχει μία μηχανιστική μέθοδος υπολογισμού των τιμών μιας συνάρτησης. Πρότεινε τον ορισμό μιας «μηχανιστικά υπολογίσιμης» συνάρτησης θετικών ακεραίων με την έννοια της αναδρομικής συνάρτησης μ θετικών ακεραίων.

Η έννοια της αναδρομικής συνάρτησης είχε βεβαίως οριστεί μαθηματικά από τον Godel το 1934 και τον Herbrand το 1931, ενώ αργότερα ο Church έδειξε ότι για τους θετικούς ακεραίους οι συγκεκριμένες συναρτήσεις δε διαφέρουν από αυτές που ο ίδιος είχε περιγράψει ως λ-ορισμένες. Όταν ο Turing έμαθε για την πρόταση του Church, άμεσα θεμελίωσε ότι η έννοια λ-ορισμένη και η δική του έννοια υπολογισιμότητας είναι ισοδύναμες. Έτσι, κατά την πρόταση του Church, οι λέξεις «αναδρομική συνάρτηση θετικών ακεραίων» μπορούν να αντικατασταθούν από τις λέξεις «συνάρτηση θετικών ακεραίων υπολογίσιμη από μηχανή Turing». Τελικά η Θέση του Church είναι ότι «μία συνάρτηση θετικών ακεραίων είναι μηχανιστικά υπολογίσιμη μόνο αν αυτή είναι αναδρομική».

Ο όρος «Θέση των Church-Turing» φαίνεται ότι για πρώτη φορά εισήχθη από τον Kleene, με μία μικρή εύννοια για τη συνεισφορά του Church: «Επομένως οι Θέσεις των Church και Turing είναι ισοδύναμες. Θα αναφερόμαστε σ' αυτές χρησιμοποιώντας συνήθως τον όρο Θέση του Church ή ως Θέση των Church - Turing αν πρόκειται για μία από τις εκδόσεις της, που συνδέεται με τις μηχανές Turing».

Η σύνδεση μιας μαθηματικής έννοιας με την έννοια του αλγορίθμου μέσω της Θέσης των Church - Turing μας δίνει τη δυνατότητα μεταξύ άλλων να *μπορούμε να αποδείξουμε ότι συγκεκριμένα υπολογιστικά προβλήματα δε μπορούν να επιλυθούν από κανένα αλγόριθμο*. Το να βρούμε συγκεκριμένα παραδείγματα υπολογιστικών προβλημάτων που δε μπορούν να επιλυθούν στα πλαίσια ενός μοντέλου υπολογισμού είναι βεβαίως πολύ ενδιαφέρον, αλλά αυτό αποδεικνύεται με διαφορετικό τρόπο στην περίπτωση των μηχανών Turing από ότι στην περίπτωση γλωσσών μη αναγνωρίσιμων από πεπερασμένα αυτόματα και αυτόματα στοίβας.

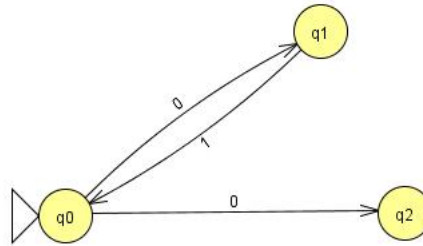
Γ'.5 Ντετερμινισμός και μη ντετερμινισμός

Ένα *μη ντετερμινιστικό* πρόγραμμα ή αλγόριθμος ενσωματώνει ένα είδος επιλογής σε κάποια στάδια της εκτέλεσής του. Τα προγράμματα που εκτελούνται στους υπολογιστές μας είναι *ντετερμινιστικά*, που σημαίνει ότι κάθε εκτελέσιμο βήμα προσδιορίζεται με μοναδικό

τρόπο. Από την άλλη μεριά, στα μη ντετερμινιστικά προγράμματα οι ενέργειες που αντιστοιχούν σ' ένα ή περισσότερα βήματα εκτέλεσης εξαρτώνται από έναν αριθμό εναλλακτικών περιπτώσεων. Χρησιμοποιούμε το μη ντετερμινισμό ως ένα μέσο, που διευκολύνει τον ορισμό μιας μηχανιστικής μεθόδου υπολογισμού και όχι για την υλοποίηση μιας ρεαλιστικής εκτελέσιμης λύσης για ένα πρόβλημα.

Για καλύτερη κατανόηση της συμπεριφοράς ενός μη ντετερμινιστικού προγράμματος ως προς ένα ντετερμινιστικό, υποθέτουμε ότι κάθε φορά που πρόκειται να γίνει μία επιλογή ενεργοποιούνται πολλά αντίγραφα του προγράμματος με το καθένα να εκτελεί μία διαφορετική επιλογή. Το αντίγραφο του προγράμματος, που πρώτο επιτυγχάνει τη λύση του προβλήματος τερματίζει όλα τ' άλλα ενεργά αντίγραφα και έτσι μία μη επιτυχής εκτέλεση σ' ένα αντίγραφο δεν επηρεάζει τις άλλες εκτελέσεις. Απλουστευτικά μπορούμε να υποθέσουμε ότι ένα μη ντετερμινιστικό πρόγραμμα θα μπορούσε να «εκτελείται» ακολουθώντας πάντα τη «σωστή» επιλογή, εφόσον βέβαια αυτή συμπεριλαμβάνεται στις εναλλακτικές που έχουν οριστεί. Επιπλέον, όταν ένας επιτυχής τερματισμός είναι εφικτός, τότε το μη ντετερμινιστικό πρόγραμμα «εκτελεί» μία ακολουθία επιλογών, που είναι η πιο σύντομη από τις ακολουθίες, που οδηγούν σε επιτυχή τερματισμό. Σε μία τέτοια ανάλυση μπορούμε να αγνοήσουμε τις ακολουθίες επιλογών, που δεν οδηγούν στο επιδιωκόμενο αποτέλεσμα.

Το ερώτημα είναι αν η προσθήκη της δυνατότητας μη ντετερμινιστικής επιλογής σ' ένα πρόγραμμα μας επιτρέπει να ορίσουμε λύσεις για προβλήματα, που διαφορετικά δε θα λύνονταν. Η απάντηση σ' αυτό το ερώτημα τελικά εξαρτάται από τις άλλες δυνατότητες που έχει η γλώσσα προγραμματισμού ή το μοντέλο υπολογισμού που χρησιμοποιούμε. Ας υποθέσουμε, για παράδειγμα, ότι τα προγράμματά μας περιγράφονται με μηχανές πεπερασμένων καταστάσεων. Ένας τρόπος για να ενσωματώσουμε τη μη ντετερμινιστική επιλογή είναι να επιτρέψουμε τον ορισμό δύο ή περισσότερων μεταβάσεων με τον ίδιο χαρακτήρα από μία κατάσταση. Αυτό επιδεικνύεται με το απλό παράδειγμα του Σχήματος Γ.6. Θα λέμε ότι ένα *μη Ντετερμινιστικό Πεπερασμένο Αυτόματο* (μΝΠΑ) αποδέχεται μία συμβολοσειρά, αν και μόνο αν υπάρχει τουλάχιστο ένα μονοπάτι από την αρχική κατάσταση σε μία κατάσταση αποδοχής. Από την άλλη, η συμβολοσειρά απορρίπτεται αν και μόνο αν δεν υπάρχει μονοπάτι που οδηγεί σε κατάσταση αποδοχής. Ο μη ντετερμινισμός διευκολύνει την αναπαράσταση μιας γλώσσας με μηχανή πεπερασμένων καταστάσεων και ταυτόχρονα ισχύει ότι όταν ένα μΝΠΑ αποδέχεται κάποια γλώσσα L , τότε υπάρχει επίσης ένα πολύ πιο μεγάλο και σύνθετο *Ντετερμινιστικό Πεπερασμένο Αυτόματο* (ΝΠΑ), που αποδέχεται την L . Άρα η προσθήκη της μη ντετερμινιστικής επιλογής δε μεταβάλλει την κλάση των γλωσσών, που γίνονται αποδεκτές από τις μηχανές πεπερασμένων καταστάσεων. Ας υποθέσουμε τώρα ότι τα προγράμματά μας περιγράφονται με αυτόματα στοιβάς. Ενσωματώνοντας τη μη ντετερμινιστική επιλογή, επιτρέπουμε την ύπαρξη ανταγωνιστικών μεταβάσεων από μία κατάσταση. Η προσθήκη της μη ντετερμινιστικής επιλογής προσδίδει μεγαλύτερη υπολογιστική ισχύ: υπάρχουν γλώσσες που μπορεί να γίνονται αποδεκτές από αυτόματα στοιβάς με μη ντετερμινιστική επιλογή, όμως δε γίνονται αποδεκτές από κανένα αυτόματο στοιβάς με μόνο ντετερμινιστικές μεταβάσεις. Τέλος, ας



Σχήμα Γ'.6: μη ντετερμινιστικό αυτόματο με δύο μεταβάσεις με 0 από την q_0

υποθέσουμε ότι τα προγράμμάτα μας περιγράφονται ως μηχανές Turing ή ως κώδικας σε κάποια σύγχρονη γλώσσα προγραμματισμού. Τότε, επανερχόμαστε σε όσα ισχύουν για τις μηχανές πεπερασμένων καταστάσεων. Ο μη ντετερμινισμός είναι ένα πολύ χρήσιμο σχεδιαστικό εργαλείο, που μας επιτρέπει να περιγράψουμε σύνθετα προγράμματα χωρίς να μας απασχολούν οι λεπτομέρειες για το πώς γίνεται η αναζήτηση των ακολουθούμενων επιλογών. Αν όμως για ένα πρόβλημα υπάρχει μία μη ντετερμινιστική μηχανή Turing, τότε υπάρχει και μία ντετερμινιστική, που επίσης επιλύει το πρόβλημα.

Στις δύο περιπτώσεις (πεπερασμένα αυτόματα και μηχανές Turing), που ο μη ντετερμινισμός δεν προσθέτει επιπλέον υπολογιστικές δυνατότητες στη μηχανή, χρησιμοποιείται λόγω της μεγαλύτερης περιγραφικής ισχύος του. Αυτά τα αποτελέσματα καθιστούν το μη ντετερμινισμό χρήσιμο ως ένα αποτελεσματικό εργαλείο σχεδίασης της συμπεριφοράς ενός συστήματος, ενώ η εργασία της δημιουργίας ενός ντετερμινιστικού προγράμματος αυτοματοποιείται από ένα μεταφραστή. Ο μη ντετερμινισμός παίζει επίσης σημαντικό ρόλο κατά την ανάλυση της πολυπλοκότητας μιας λύσης σε ένα πρόβλημα: πόσος είναι ο απαιτούμενος χρόνος εκτέλεσης και μνήμη; Σε αυτού του τύπου την ανάλυση, ο μη ντετερμινισμός μας επιτρέπει να διαχωρίσουμε την ανάλυση σε δύο μέρη:

1. Την πολυπλοκότητα ενός μόνο μονοπατιού μέσα στο δέντρο αναζήτησης με τις εναλλακτικές περιπτώσεις στις μη ντετερμινιστικές επιλογές του προγράμματος. Κάθε τέτοιο μονοπάτι στην τυπική περίπτωση αντιστοιχεί στον έλεγχο μιας πλήρους ακολουθίας επιλογών προκειμένου να διαπιστωθεί αν αυτή αποτελεί λύση για το πρόβλημα.
2. Τη συνολική πολυπλοκότητα όλης της διαδικασίας αναζήτησης της λύσης.

Παρότι λοιπόν ο μη ντετερμινισμός φαίνεται να μη διευρύνει την έννοια του μηχανιστικού υπολογισμού ή αλγορίθμου, είναι ένα πολύ χρήσιμο εργαλείο κατά την ανάλυση των διαφόρων προβλημάτων και στο να βρούμε το πώς αυτά αναπαριστώνται με τα μοντέλα υπολογισμού που περιγράφουμε.

Κεφάλαιο Δ΄

Μηχανές **Turing**

Οι μηχανές Turing (MT) δεν είναι απλά μία ακόμη μηχανή αναγνώρισης για κάποια ευρύτερη οικογένεια γλωσσών από τις γλώσσες, που γίνονται δεκτές από τα Αυτόματα Στοίβας. Σύμφωνα με τη Θέση των Church και Turing, οι MT είναι ένα γενικό μοντέλο υπολογισμού, που μπορεί δυνητικά να εκτελέσει οποιονδήποτε αλγόριθμο. Στο κεφάλαιο αυτό εισάγουμε τους βασικούς ορισμούς και παρουσιάζουμε παραδείγματα, που επιδεικνύουν τις διάφορες χρήσεις των MT: αποδοχή μιας γλώσσας, υπολογισμός μιας συνάρτησης, καθώς και εκτέλεση υπολογισμών διαφόρων άλλων τύπων. Επίσης αναφερόμαστε εν συντομία σε παραλλαγές των MT, όπως οι MT πολλών ταινιών. Εισάγουμε τις μη Ντεντερμινιστικές MT, καθώς και τη Γενική MT, που λειτουργεί όπως ακριβώς ο σύγχρονος υπολογιστής, όταν αυτός εκτελεί ένα πρόγραμμα αποθηκευμένο στη μνήμη του.

Δ΄.1 Ορισμός μηχανών Turing (MT)

Τόσο τα Πεπερασμένα Αυτόματα (ΠΑ) όσο και τα Αυτόματα Στοίβας (ΑΣ) δέχονται μία συμβολοσειρά στην είσοδο και εκτελούν έναν αλγόριθμο για να παράξουν ένα αποτέλεσμα ακολουθώντας το σύνολο κανόνων, που αντιστοιχεί στον τύπο της μηχανής. Στις δύο αυτές περιπτώσεις, η μηχανή μπορεί να εκτελέσει μόνο συγκεκριμένης μορφής αλγορίθμους: στην περίπτωση ενός ΠΑ εκτελούνται αλγόριθμοι, που χαρακτηρίζονται από ένα απόλυτο όριο στο μέγεθος της πληροφορίας, που απομνημονεύεται, ενώ στην περίπτωση των ΑΣ εκτελούνται μόνο αλγόριθμοι, για τους οποίους αρκεί η προσπέλαση πληροφορίας με βάση τους κανόνες της στοίβας.

Όπως είδαμε είναι εύκολο να βρούμε περιπτώσεις γλωσσών, που δε γίνονται δεκτές από τις συγκεκριμένες μηχανές εξαιτίας των προαναφερόμενων περιορισμών.

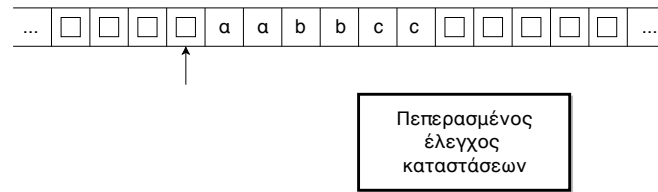
Το αφαιρετικό μοντέλο υπολογισμού, που επινόησε ο Turing δεν προήλθε από την επέκταση ενός ΠΑ με μία επιπλέον δομή δεδομένων και δε θα μπορούσε να συμβεί αυτό, καθώς είναι προγενέστερο από τα ΠΑ. Ο αρχικός στόχος του Turing ήταν να επιδείξει τους εγγενείς περιορισμούς των αλγοριθμικών μεθόδων και γι' αυτό το λόγο, όπως είδαμε επικεντρώθηκε

στη μελέτη του Entscheidungsproblem. Έτσι, ξεκίνησε με την ιδέα να σχηματοποιήσει ένα μοντέλο αρκετά γενικό ώστε να μπορεί καταρχήν να εκτελέσει έναν οποιοδήποτε αλγόριθμο, ώστε να οδηγηθεί στο συμπέρασμα ότι οι όποιες αδυναμίες του μοντέλου είναι βασικά περιορισμοί της αλγοριθμικής μεθόδου. Ο σκοπός λοιπόν της MT ήταν να παίξει το ρόλο ενός θεωρητικού μοντέλου και όχι μιας πρακτικής προσέγγισης για την εκτέλεση υπολογισμών. Παρόλα αυτά, η MT επιδεικνύει πολλά από τα χαρακτηριστικά των σύγχρονων ηλεκτρονικών υπολογιστών.

Ο Turing υπέθεσε έναν άνθρωπο - υπολογιστή, ο οποίος εργάζεται με μολύβι και χαρτί. Αποφάσισε λοιπόν, χωρίς βλάβη της γενικότητας, ότι αυτός ο υπολογιστής λειτουργεί με κάποιους απλούς κανόνες. Πρώτον, ο υπολογιστής μπορεί να εγγράψει στο χαρτί μόνο σύμβολα από ένα καθορισμένο πεπερασμένο αλφάβητο. Δεύτερον, κάθε βήμα, που εκτελεί ο υπολογιστής, εξαρτάται μόνο από το σύμβολο που διαβάζει και την «κατάσταση της μνήμης» εκείνη τη στιγμή. Τρίτον, παρόλο που η «κατάσταση της μνήμης» μπορεί να μεταβάλλεται ως αποτέλεσμα ανάγνωσης διαφορετικών συμβόλων, μόνο ένας πεπερασμένος αριθμός διακριτών καταστάσεων είναι επισκέψιμος. με βάση αυτές τις αρχές, ο Turing απομόνωσε τις ενέργειες, που εκείνος θεώρησε ως στοιχειώδη βήματα στην εκτέλεση ενός υπολογισμού:

1. Ανάγνωση ενός συμβόλου από το χαρτί.
2. Διαγραφή ενός συμβόλου ή αντικατάστασή του από ένα άλλο.
3. μετακίνηση της προσοχής από ένα σύμβολο σε ένα άλλο γειτονικό σύμβολο.

Περιγράφουμε τώρα με περισσότερες λεπτομέρειες τη λειτουργία μιας MT, με σκοπό να εισάγουμε τον τυπικό ορισμό της. μία MT λειτουργεί με βάση δύο πεπερασμένα αλφάβητα: ένα αλφάβητο εισόδου και ένα δεύτερο, πιθανόν μεγαλύτερο, που χρησιμοποιείται κατά τον υπολογισμό. Χρησιμοποιεί επίσης ένα πεπερασμένο σύνολο καταστάσεων, που αντιστοιχεί στις διακριτές «καταστάσεις της μνήμης» του ανθρώπου - υπολογιστή, κατά τον Turing. Για την εγγραφή των υπολογισμών, ένα φύλλο χαρτί θα εκτείνονταν σε δύο διαστάσεις, αλλά ο Turing υιοθέτησε μία απλουστευτική προσέγγιση υποκαθιστώντας το με μία ταινία, που εκτείνεται απείρως προς τις δύο κατευθύνσεις. Οι θέσεις της ταινίας απεικονίζονται με τετράγωνα, που στο καθένα μπορεί να εγγραφεί ένα σύμβολο. Αν σε ένα τετράγωνο δεν έχει καταχωρηθεί κάποιο σύμβολο, τότε λέμε ότι αυτό περιέχει το κενό σύμβολο. Σε κάποιες περιπτώσεις, βολεύει να αναφερόμαστε στα τετράγωνα της ταινίας με αύξουσα αρίθμηση ξεκινώντας από το 0 και με κατεύθυνση προς τα δεξιά, αλλά αυτή η αρίθμηση δεν είναι μέρος του βασικού ορισμού του μοντέλου και βεβαίως οι αριθμοί δεν είναι απαραίτητοι για την περιγραφή της λειτουργίας της μηχανής. Η ανάγνωση και η εγγραφή στην ταινία οπτικοποιείται με την κίνηση μιας κεφαλής ανάγνωσης/εγγραφής, που σε μία δεδομένη στιγμή θα βρίσκεται σε κάποιο από τα τετράγωνα της ταινίας. Στο Σχήμα Δ'.7 ανακαλούμε την απεικόνιση της λειτουργίας μιας μηχανής Turing, που παρουσιάσαμε, με την κεφαλή ανάγνωσης/εγγραφής να εμφανίζεται με ένα τόξο. μία κίνηση της MT καθορίζεται από την τρέχουσα κατάσταση και το σύμβολο



Σχήμα Δ'.7: Απεικόνιση της λειτουργίας μιας μηχανής Turing

της ταινίας, στο οποίο δείχνει η κεφαλή και προσδιορίζεται από τρεις πιθανές μεταβολές:

1. Αλλαγή της τρέχουσας κατάστασης σε μία επόμενη κατάσταση πιθανώς διαφορετική.
2. Αντικατάσταση του συμβόλου στο τετράγωνο της τρέχουσας θέσης της κεφαλής με ένα άλλο πιθανώς διαφορετικό σύμβολο.
3. μετακίνηση της κεφαλής ανάγνωσης/εγγραφής μία θέση προς τα δεξιά ή προς τα αριστερά.

Η ταινία παίζει το ρόλο της συσκευής εισόδου (η είσοδος είναι η πεπερασμένη συμβολοσειρά από μη κενά σύμβολα, που αρχικά βρίσκεται στην ταινία), της διαθέσιμης μνήμης για χρήση κατά τη διάρκεια του υπολογισμού, αλλά και της συσκευής εξόδου. Η έξοδος, αν ενδιαφέρει, είναι η συμβολοσειρά με τα σύμβολα, που τελικά μένουν στην ταινία μετά από τον υπολογισμό.

Μία σημαντική διαφορά μεταξύ της MT και των δύο απλούστερων τύπων μηχανής, με τους οποίους ασχοληθήκαμε, είναι το ότι στις MT δεν περιοριζόμαστε μόνο σε ένα πέρασμα επεξεργασίας της εισόδου από τα αριστερά προς τα δεξιά και εκτέλεσης του υπολογισμού κατά την ανάγνωση. Η συμβολοσειρά εισόδου βρίσκεται αρχικά στην ταινία πριν από την έναρξη του υπολογισμού. μπορεί ενίοτε να αντιλαμβανόμαστε τη λειτουργία της μηχανής ως ανάγνωση της εισόδου, καθώς η κεφαλή κινείται πάνω στην ταινία, αλλά συχνά το πέρασμα της κεφαλής από τετράγωνα που περιέχουν σύμβολα εισόδου γίνεται απλά για να μεταβεί σε ένα άλλο μέρος της ταινίας. Αυτό είναι συμβατό με την επεξεργασία από έναν άνθρωπο - υπολογιστή, που μπορεί να εξετάζει ένα μέρος της εισόδου, να την τροποποιεί ή να τη διαγράφει και στη συνέχεια να εκτελεί κάποιους υπολογισμούς σε μία διαφορετική περιοχή της ταινίας, χωρίς να αποκλείεται η επιστροφή του για να επανεξετάσει την είσοδο και να επαναλάβει οποιεσδήποτε από αυτές τις ενέργειες ή ενδεχομένως να σταματήσει τον υπολογισμό ακόμη και πριν εξαντλήσει όλα τα περιεχόμενα της εισόδου, αφού όλα τα παραπάνω είναι πιθανά.

Οι δύο βασικές χρήσεις μιας MT, στις οποίες θα επικεντρωθούμε είναι η *αποδοχή μιας γλώσσας* και ο *υπολογισμός μιας συνάρτησης*. Η πρώτη χρήση μοιάζει με τον υπολογισμό ενός ΠΑ ή ενός ΑΣ, μόνο που οι γλώσσες των MT μπορεί να είναι πιο γενικές. Η δεύτερη χρήση

είναι εφικτή, εξαιτίας των επιπλέον δυνατοτήτων εξόδου των MT.

Ένα ΠΑ απαιτείται να έχει περισσότερες από μία τελικές καταστάσεις, αν για δύο συμβολοσειρές u και v που γίνονται αποδεκτές διαφέρει η πληροφορία, που πρέπει να απομνημονεύεται. Επιπλέον, μπορεί οι u και v να γίνονται αποδεκτές αμφότερες, αλλά η μηχανή να δέχεται επιπλέον είσοδο και στην περίπτωση αυτή οι ενέργειές της να καθορίζονται από το αν αρχικά βρήκε την u ή την v . μία MT δε χρειάζεται να αποφανθεί αν κάθε πρόθεμα της συμβολοσειράς εισόδου είναι αποδεκτό ή όχι. μπορεί να αρχίσει τον υπολογισμό της με ολόκληρη τη συμβολοσειρά στην είσοδο και όταν αποφασίσει να αποδεχτεί την είσοδο ή να την απορρίψει, τότε ο υπολογισμός τερματίζεται. Γι' αυτό το λόγο, μία MT δε χρειάζεται περισσότερες από δύο τελικές καταστάσεις, μία που θα εκφράζει αποδοχή και μία άλλη για την απόρριψη. Αν τώρα μία MT εκτελεί υπολογισμούς άλλου τύπου, τότε όταν ο υπολογισμός ολοκληρώνεται κανονικά τερματίζεται σε μία κατάσταση αποδοχής, ενώ η κατάσταση απόρριψης εκφράζεται από κάποιου είδους «αποτυχία» ή μη αναμενόμενο τερματισμό. μία τέτοια περίπτωση είναι οι υπολογισμοί της MT του Σχήματος Γ.4, όπου μόνο οι συμβολοσειρές της μορφής $\{a^n b^n c^n : n \geq 0\}$ οδηγούν τελικά τη μηχανή στην εικονιζόμενη κατάσταση τερματισμού.

με τις MT υπάρχει ένα επιπλέον ζήτημα σε σύγκριση με τους υπολογισμούς στα ΠΑ και τα ΑΣ. Αν μία MT αποφασίσει την αποδοχή ή την απόρριψη της συμβολοσειράς εισόδου, τότε τερματίζει την εκτέλεσή της. Αν όμως δεν αποφασίσει ένα από τα δύο ενδεχόμενα, τότε, όπως είδαμε μπορεί να συνεχίζει την εκτέλεσή της για πάντα. Τυπικά, μία MT ορίζεται ως εξής:

Ορισμός Δ'.1.1. *μία MT M ορίζεται ως εξάδα $(K, \Sigma, \Gamma, \delta, s, H)$ με:*

- K να είναι ένα πεπερασμένο σύνολο καταστάσεων,
- Σ να είναι το αλφάβητο εισόδου στο οποίο δεν περιλαμβάνεται το κενό σύμβολο $\square$,
- Γ να είναι το αλφάβητο με όλα τα σύμβολα, που μπορεί να εμφανίζονται στην ταινία, συμπεριλαμβανομένων του $\square$ και των συμβόλων του Σ ,
- $s \in K$ να είναι η αρχική κατάσταση,
- $H \subseteq K$ να είναι το σύνολο των καταστάσεων τερματισμού και
- δ η συνάρτηση μετάβασης, που αντιστοιχίζει διατεταγμένα ζεύγη από το $(K - H) \times \Gamma$ στο $K \times \Gamma \times \{\rightarrow, \leftarrow\}$.

Αν η συνάρτηση δ περιλαμβάνει τη μετάβαση $(p, a), (q, b, d)$ με $p, q \in K, a, b \in \Gamma$ και $d \in \{\rightarrow, \leftarrow\}$, τότε όταν η M βρίσκεται στην κατάσταση p με το σύμβολο a στην κεφαλή ανάγνωσης/εγγραφής, θα μεταβεί στην κατάσταση q και θα εγγράψει b στη θέση του a , ενώ θα μετακινήσει την κεφαλή μία θέση προς την κατεύθυνση d , δηλαδή μία θέση προς τα δεξιά ή μία θέση προς τα αριστερά.

Αρχικά σε όλες τις θέσεις της ταινίας εκτός από εκείνες της συμβολοσειράς εισόδου υπάρχει το κενό σύμβολο $\square$. Υποθέτουμε ότι η συμβολοσειρά εισόδου στην ταινία δε μπορεί να περιλαμβάνει το $\square$. Ασφαλώς η MT μπορεί να εγγράψει το $\square$ σε οποιαδήποτε θέση της ταινίας, όπως ακριβώς ισχύει και για οποιοδήποτε άλλο σύμβολο του Γ . Αυτό σημαίνει ότι χρειάζεται προσοχή στον ορισμό μιας MT, που εκτελεί αυτή την ενέργεια, καθώς σε κάθε περίπτωση θα πρέπει η μηχανή να διακρίνει το τέλος της εισόδου από τα κενά, που εισάγονται κατά την επεξεργασία της εισόδου.

Η συνάρτηση δ ορίζεται για όλα τα ζεύγη (q, σ) με q μία μη τερματική κατάσταση και $\sigma \in \Gamma$ και αφού πρόκειται για συνάρτηση μετάβασης - όχι σχέση- είναι σαφές ότι ο Ορισμός που δώσαμε αναφέρεται σε *Ντεντερμινιστικές μηχανές Turing*.

Παράδειγμα Δ'.1.1. *Καρκινικές συμβολοσειρές άρτιου μήκους* Επιδεικνύουμε τη λειτουργία και τον ορισμό μιας MT, που αποδέχεται το σύνολο των συμβολοσειρών

$$\{w \in \{0, 1\}^* : w = w^R \text{ και } w \text{ έχει άρτιο μήκος}\}$$

.

Η συγκεκριμένη MT διαβάζει το πρώτο μη κενό σύμβολο της ταινίας (0 ή 1), το αντικαθιστά με $\square$ και αλληλάζει κατάσταση (q_1 ή q_2). Στη συνέχεια σαρώνει το υπόλοιπο της συμβολοσειράς προς τα δεξιά χωρίς να την αλληλάζει, μέχρι να εντοπίσει $\square$. Ακολούθως η κεφαλή ανάγνωσης/εγγραφής μετακινείται προς τα αριστερά. Αν το σύμβολο στη δεξιότερη θέση της συμβολοσειράς ταυτίζεται με το σύμβολο στην αριστερότερη θέση (που αντικαταστάθηκε με $\square$ έχοντας όμως καταγραφεί στη «μνήμη»), τότε εγγράφεται $\square$ στην τελευταία από δεξιά θέση. Διαφορετικά η MT τερματίζει. μετά η κεφαλή ανάγνωσης/εγγραφής μετακινείται προς τα αριστερά μέχρι να συναντήσει $\square$. Τα βήματα του υπολογισμού επαναλαμβάνονται, ενώ μεταβάλλεται η κατάσταση της μηχανής με κατάλληλο τρόπο. Τυπικά η ζητούμενη MT ορίζεται ως εξής:

$$M = (\{q_0, q_1, q_2, q_3, q_4, q_5, q_6, h_a, h_r\}, \{0, 1\}, \{0, 1, \square\}, \delta, q_0, \{h_a, h_r\}) \quad (\Delta'.1)$$

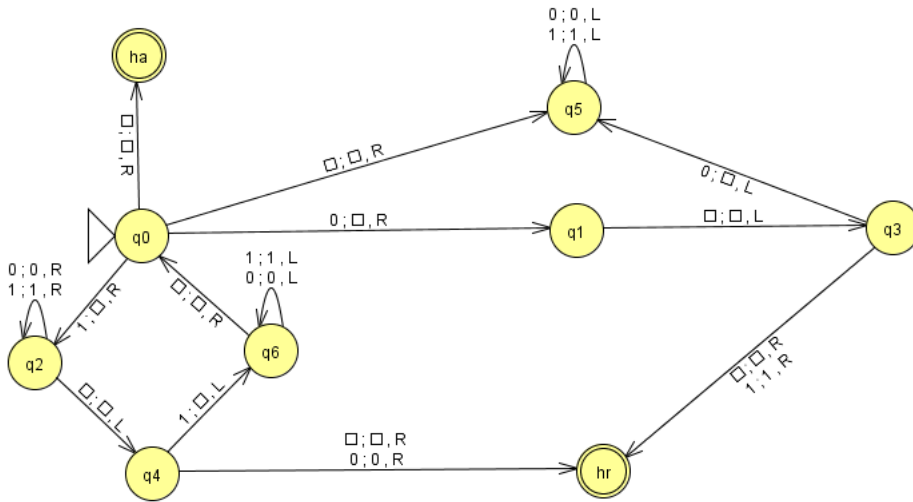
όπου

$$\begin{aligned} \delta = & (((q_0, 0), (q_1, \square, \rightarrow)) , \\ & ((q_0, 1), (q_2, \square, \rightarrow)) , \\ & ((q_0, \square), (h_a, \square, \rightarrow)) , \\ & ((q_1, 0), (q_1, 0, \rightarrow)) , \\ & ((q_1, 1), (q_1, 1, \rightarrow)) , \\ & ((q_1, \square), (q_3, \square, \leftarrow)) , \\ & ((q_2, 0), (q_2, 0, \rightarrow)) , \\ & ((q_2, 1), (q_2, 1, \rightarrow)) , \\ & ((q_2, \square), (q_4, \square, \leftarrow)) , \\ & ((q_3, 0), (q_5, \square, \leftarrow)) , \\ & ((q_3, 1), (h_r, 1, \rightarrow)) , \\ & ((q_3, \square), (h_r, \square, \rightarrow)) , \\ & ((q_4, 0), (h_r, 0, \rightarrow)) , \\ & ((q_4, 1), (q_6, \square, \leftarrow)) , \\ & ((q_4, \square), (h_r, \square, \rightarrow)) , \\ & ((q_5, 0), (q_5, 0, \leftarrow)) , \\ & ((q_5, 1), (q_5, 1, \leftarrow)) , \\ & ((q_5, \square), (q_0, \square, \rightarrow)) , \\ & ((q_6, 0), (q_6, 0, \leftarrow)) , \\ & ((q_6, 1), (q_6, 1, \leftarrow)) , \\ & ((q_6, \square), (q_0, \square, \rightarrow))) \end{aligned}$$

Η περιγραφή της συνάρτησης μετάβασης δ μιας MT με πεντάδες, όπως στο Παράδειγμα μας δεν είναι ιδιαίτερα πρακτική και για αυτό το λόγο έχει υιοθετηθεί στη βιβλιογραφία η γραφική αναπαράσταση, που έχουμε χρησιμοποιήσει και συγκεκριμένα στο Σχήμα Γ'.4. Οι τριάδες της μορφής $j : k, l$, με τις οποίες επιγράφονται τα τόξα του Σχήματος Γ'.4 περιέχουν όλη την απαιτούμενη πληροφορία για τον ορισμό των πεντάδων της συνάρτησης μετάβασης δ μιας MT. Το j αναφέρεται στο σύμβολο που βρίσκεται στην τρέχουσα θέση της κεφαλής ανάγνωσης/εγγραφής όταν η MT βρίσκεται στην κατάσταση από την οποία εξέρχεται το τόξο της μετάβασης. Το k αναφέρεται στο σύμβολο που θα αντικαταστήσει στην ταινία εκείνο που διαβάστηκε και το l εκφράζει την κατεύθυνση κίνησης της κεφαλής με το R να αντιστοιχεί στο $\rightarrow$ και το L να αντιστοιχεί στο $\leftarrow$. Η νέα κατάσταση του πεπερασμένου ελέγχου της MT είναι αυτή, στην οποία εισέρχεται το τόξο της μετάβασης.

Παρατηρούμε ότι στη MT του Σχήματος Γ'.4 έχουμε μία κατάσταση τερματισμού, που μπορεί να προσεγγιστεί μόνο από υπολογισμούς για συμβολοσειρές της μορφής $\{a^n b^n c^n : n \geq 0\}$.

Για όλες τις άλλες συμβολοσειρές ο υπολογισμός «αποτυγχάνει» χωρίς να προσεγγίζεται η κατάσταση τερματισμού. Στην περίπτωση του Παραδείγματος έχουμε ορίσει δύο καταστάσεις τερματισμού, με την h_a να εκφράζει αποδοχή και την h_r να εκφράζει απόρριψη της συμβολοσειράς. Τέλος είναι σημαντικό να επισημάνουμε ότι όταν χρησιμοποιείται ο γραφικός συμβολισμός μπορεί για λόγους αναγνωσιμότητας να παραλείπονται μεταβάσεις, που δε χρησιμοποιούνται, αλλά η δήλωσή τους σε έναν τυπικό ορισμό της MT είναι απαραίτητη, για να είναι η δ συνάρτηση. Έτσι, στο Σχήμα Γ'.4 δεν απεικονίζεται πλήρως η συνάρτηση μετάβασης της MT, ενώ στο Παράδειγμα έχουμε έναν πλήρη ορισμό συνάρτησης μετάβασης, που όμως δεν είναι εύκολα αναγνώσιμος και χρηστικός. Στο Σχήμα απεικονίζεται γραφικά η MT του Παραδείγματος.



Σχήμα Δ'.8: Γραφικός συμβολισμός για τη MT του Παραδείγματος

Ορίζουμε ως ενεργό μέρος της ταινίας μιας MT το ελάχιστο μέρος της ταινίας, που περιέχει το σύμβολο στη θέση της κεφαλής ανάγνωσης/εγγραφής και όλες τις μη κενές θέσεις της ταινίας. μία διαμόρφωση για κάποια MT $M = (K, \Sigma, \Gamma, s, H)$ είναι μία τετράδα, που δίνεται ως στοιχείο του

$$K \times ((G? \{\square\})\Gamma^*) \cup \{s\} \times \Gamma \times (\Gamma^*(\Gamma? \{\square\})) \cup \{s\} \quad (\Delta'.2)$$

Το πρώτο μέρος εκφράζει μία κατάσταση του πεπερασμένου ελέγχου, το δεύτερο μέρος τη συμβολοσειρά, που βρίσκεται στο ενεργό μέρος της ταινίας στα αριστερά της κεφαλής ανάγνωσης/εγγραφής, το τρίτο μέρος αναφέρεται στο σύμβολο στη θέση της κεφαλής και το τελευταίο μέρος στη συμβολοσειρά, που βρίσκεται στο ενεργό μέρος της ταινίας στα δεξιά της κεφαλής. Σημειώνουμε ότι αν και η ταινία μιας MT είναι άπειρη, η περιγραφή μιας οποιασδήποτε διαμόρφωσης είναι πεπερασμένη, καθώς περιλαμβάνεται σε αυτή το μικρότερο

συνεχές μέρος της ταινίας από όλες τις μη κενές θέσεις μέχρι και το τετράγωνο στη θέση της κεφαλής ανάγνωσης/εγγραφής.

Κάθε διαμόρφωση της μορφής (q, u, a, v) θα τη γράφουμε με τη συντόμευση $(q, \underline{a}uv)$, με το υπογραμμισμένο σύμβολο να εκφράζει το περιεχόμενο της ταινίας στη θέση της κεφαλής ανάγνωσης/εγγραφής. Η *αρχική διαμόρφωση* μιας οποιασδήποτε MT M με αρχική κατάσταση s και είσοδο w είναι η $(s, \square w)$. Οποιαδήποτε διαμόρφωση, που η κατάστασή της είναι ένα στοιχείο του H είναι *τελική διαμόρφωση*.

Η συνάρτηση μετάβασης δ ορίζει τη λειτουργία της M σε ένα βήμα υπολογισμού. μέσω αυτής περιγράφεται η ακολουθία διαμορφώσεων από τις οποίες περνάει η εκτέλεση της M . Ορίζουμε τη σχέση «μεταβαίνει σε ένα βήμα» που γράφεται με το σύμβολο $\vdash_M$ και συνδέει τις διαμορφώσεις c_1 και c_2 αν και μόνο αν η μηχανή M μπορεί να περάσει από τη διαμόρφωση c_1 στη διαμόρφωση c_2 σε ένα βήμα :

$$(q_1, w_1) \vdash_M (q_2, w_2) \text{ αν η } (q_2, w_2) \text{ παράγεται μέσω της } \delta \text{ σε ένα βήμα} \quad (\Delta'.3)$$

Τώρα ορίζουμε τη σχέση «μεταβαίνει», που γράφεται με το σύμβολο $\vdash_M^*$, ως την ανακλαστική, μεταβατική κλειστότητα της σχέσης $\vdash_M$. Έτσι, από τη διαμόρφωση C_1 η μηχανή μεταβαίνει στην C_2 αν:

$$C_1 \vdash_M^* C_2 \quad (\Delta'.4)$$

Ένα *μονοπάτι* από τη MT M είναι μία ακολουθία διαμορφώσεων $C_0, C_1, \dots$ τέτοια ώστε η C_0 είναι η αρχική διαμόρφωση και

$$C_0 \vdash_M C_1 \vdash_M C_2 \vdash_M \dots \quad (\Delta'.5)$$

Ένας *υπολογισμός* της MT M είναι ένα μονοπάτι, που τερματίζει, είναι δηλαδή μία ακολουθία διαμορφώσεων $C_0, C_1, \dots, C_n$ για κάποιο $n \geq 0$, τέτοια ώστε η C_0 να είναι μία αρχική διαμόρφωση της M και η C_n μία τελική διαμόρφωση και επίσης:

$$C_0 \vdash_M C_1 \vdash_M C_2 \vdash_M \dots \vdash_M C_n \quad (\Delta'.6)$$

Αν ένας υπολογισμός τερματίζει σε n βήματα, θα λέμε ότι έχει μήκος n και θα τον γράφουμε ως:

$$C_0 \vdash_M^n C_n \quad (\Delta'.7)$$

Δ'.1.1 Ένας συνθετικός συμβολισμός για MT

Ο ορισμός ακόμη και μιας απλής MT με τον τρόπο, που περιγράψαμε, είναι γενικά χρονοβόρος, όπως επίσης η ανάγνωση και κατανόηση της λειτουργίας της. Στη σχετική βιβλιογραφία είναι επίσης διαδεδομένος ο ορισμός MT ως σύνθεση απλούστερων MT, που εκτελούν βασικές λειτουργίες, όπως η εγγραφή στην ταινία, η μετακίνηση της κεφαλής ανάγνωσης/εγγραφής και ο τερματισμός:

- μηχανές εγγραφής συμβόλων Για κάθε $x \in \Gamma$ ορίζεται M_x , που θα γράφεται απλά ως x , η MT που εγγράφει το x στην τρέχουσα θέση της ταινίας και στη συνέχεια τερματίζει. Σύμφωνα με το Ορισμό κάθε τέτοια MT εκτελεί δύο κινήσεις. Στην πρώτη κίνηση εγγράφει το νέο σύμβολο στην ταινία και μετακινεί την κεφαλή προς τα δεξιά. Στη δεύτερη κίνηση, επανεγγράφει το χαρακτήρα, που θα βρει η κεφαλή στη θέση που βρίσκεται και στη συνέχεια την μετακινεί προς τα αριστερά. Οι κινήσεις αυτές όμως είναι λεπτομέρειες, που δε χρειάζεται να μας απασχολούν κατά τον ορισμό MT μέσω σύνθεσης απλούστερων MT.
- μηχανές μετακίνησης κεφαλής Έχουμε δύο τέτοιες μηχανές, την R που επανεγγράφει το σύμβολο που βρίσκεται στη θέση της κεφαλής, η οποία στη συνέχεια μετακινείται προς τα δεξιά και την L , που επανεγγράφει το σύμβολο στη θέση της κεφαλής, που στη συνέχεια μετακινείται προς τα αριστερά.
- μηχανές τερματισμού Όλες οι MT, που αναφέραμε, τερματίζουν όταν δεν υπάρχει κάτι άλλο που μπορούν να κάνουν, δηλαδή μεταβαίνουν σε μία κατάσταση για την οποία δεν ορίζεται η συνάρτηση δ . Παρόλα αυτά, σε πολλές περιπτώσεις χρειάζεται να δηλώσουμε τον τερματισμό ρητά. Θα χρησιμοποιούμε για το σκοπό αυτό τρεις διαφορετικές μηχανές τερματισμού:
 - την h , που απλά τερματίζει. Η h χρησιμεύει στο να δείξουμε ρητά ότι τερματίζει κάποιο μονοπάτι της MT χωρίς να μας ενδιαφέρει αν αυτή αποδέχεται ή απορρίπτει
 - την y , που τερματίζει με αποδοχή
 - την n , που τερματίζει με απόρριψη

Στη συνέχεια θα περιγράψουμε το πώς ακριβώς γίνονται:

- Έλεγχος ταινίας και υπό συνθήκη διακλάδωση ανάλογα με το σύμβολο, που διαβάζεται
- Συνδυασμοί των παραπάνω λειτουργιών.

Για το σκοπό αυτό υιοθετούμε ένα συμβολισμό όμοιο με εκείνο, που χρησιμοποιήσαμε για τον ορισμό μηχανών καταστάσεων, όπως τα πεπερασμένα αυτόματα και τα αυτόματα στοίβας. Θα χρησιμοποιήσουμε δύο βασικές μορφές σύνθεσης:

- M_1M_2 Από την αρχική κατάσταση της M_1 , εκτελούνται μεταβάσεις σύμφωνα με τη συνάρτηση μετάβασης μέχρι η μηχανή να τερματίσει. Όταν αυτό συμβεί και χωρίς να μετακινηθεί η κεφαλή ανάγνωσης/εγγραφής ξεκινάει η εκτέλεση μεταβάσεων για την M_2 από την αρχική της κατάσταση μέχρι να τερματίσει. Σε αυτή την περίπτωση τερματίζει και η M_1M_2 . Αν οποιαδήποτε εκ των M_1 και M_2 αποτύχει να τερματίσει, τότε αποτυγχάνει και η M_1M_2 .

- $M_1 \xrightarrow{\text{συνθήκη}} M_2$ Από την αρχική κατάσταση της M_1 , εκτελούνται μεταβάσεις σύμφωνα με τη συνάρτηση μετάβασης μέχρι η μηχανή να τερματίσει. Αν συμβεί αυτό ελέγχεται η συνθήκη και αν αληθεύει, τότε ξεκινάει η εκτέλεση μεταβάσεων της M_2 από την αρχική της κατάσταση χωρίς να έχει μετακινηθεί η κεφαλή ανάγνωσης/εγγραφής. Η απλούστερη συνθήκη είναι η εμφάνιση ενός συγκεκριμένου συμβόλου στη θέση της κεφαλής ανάγνωσης/εγγραφής, αλλά δεν αποκλείεται και η δήλωση άλλων συνθηκών. μία μηχανή με αυτή τη δομή αποτυγχάνει να τερματίσει αν:

- αποτύχει να τερματίσει η M_1 ή
- η συνθήκη είναι αληθής και η M_2 αποτύχει να τερματίσει

Η MT από την οποία αρχίζει η εκτέλεση μεταβάσεων για τη σύνθετη MT επιγράφεται με $>$. Εισάγουμε κάποιες συντομεύσεις, που αποσκοπούν στο να γίνει ο συμβολισμός πιο χρηστικός κατά τον ορισμό σύνθετων MT:

$$M_1 \xrightleftharpoons[b]{a} M_2 \text{ γίνεται } M_1 \xrightarrow{a,b} M_2$$

$$M_1 \xrightarrow[\text{εκτός από το } a]{\text{όλα τα στοιχεία του } \Gamma} M_2 \text{ γίνεται } M_1 \xrightarrow{\neg a} M_2$$

Η εκφραστικότητα του συμβολισμού βελτιώνεται σημαντικά, όταν διατίθεται ένας μηχανισμός αποθήκευσης τιμών για μεταβλητές. Κάθε μεταβλητή μπορεί να δέχεται ως τιμή ένα μόνο σύμβολο. Σε μία τυπική MT υπάρχει δυνατότητα απομνημόνευσης τιμών για ένα πεπερασμένο σύνολο τέτοιων μεταβλητών, είτε εγγράφοντας τις κάπου στην ταινία, είτε χρησιμοποιώντας τις για διακλάδωση σε μία διαφορετική κατάσταση, για κάθε πιθανή τιμή. με τη δεύτερη λύση αποφεύγονται οι μετακινήσεις μπρος - πίσω πάνω στην ταινία, αλλά μπορεί τελικά να έχουμε έκρηξη του αριθμού των καταστάσεων, αφού πρακτικά απαιτείται ένα διαφορετικό αντίγραφο μηχανής για κάθε συνδυασμό τιμών, που μπορεί να έχει ένα σύνολο μεταβλητών. Για τον ορισμό MT δεν θα είναι σημαντικός ο μηχανισμός, που χρησιμοποιείται για τις μεταβλητές, αλλά το να μπορούμε να τις ονοματίζουμε και να αναφερόμαστε σε αυτές ρητά στις συνθήκες, που αναγράφονται στα τόξα. Έτσι, έχουμε τις παρακάτω συντομεύσεις:

$$M_1 \xrightarrow[\text{όλα τα στοιχεία του } \Gamma]{\text{εκτός από το } a} M_2 \text{ γίνεται } M_1 \xrightarrow{x \leftarrow \neg a} M_2$$

και το x παίρνει την τιμή της τρέχουσας θέσης

$$M_1 \xrightarrow{a,b} M_2 \text{ γίνεται } M_1 \xrightarrow{x \leftarrow a,b} M_2$$

και το x παίρνει την τιμή της τρέχουσας θέσης

Η τιμή μιας μεταβλητής χρησιμοποιείται με δύο τρόπους. Καταρχήν χρησιμοποιείται ως συνθήκη σε μία μετάβαση, όπως στο παρακάτω παράδειγμα:

$$M_1 \xrightarrow{x=y} M_2$$

αν $x = y$ τότε εκτέλεσε την μετάβαση

μία άλλη χρήση είναι όταν ρητά γράφουμε την τιμή της μεταβλητής, όπως στο παρακάτω παράδειγμα.

Παράδειγμα Δ'.1.2. μεταβλητές σε MT 'Εστω η MT

$$M \Rightarrow \frac{x \rightarrow \neg \square}{Rx} \quad (\Delta'.8)$$

Αν στην κεφαλή δεν υπάρχει το κενό σύμβολο, η M αποθηκεύει το σύμβολο στη μεταβλητή x , μετακινεί την κεφαλή δεξιά και αντιγράφει στη νέα θέση την τιμή της x . Αν όμως στην κεφαλή βρίσκεται το κενό σύμβολο, τότε δεν ορίζει κάποια ενέργεια και έτσι τερματίζει.

Στη συνέχεια παραθέτουμε τους ορισμούς κάποιων πολύ χρήσιμων MT:

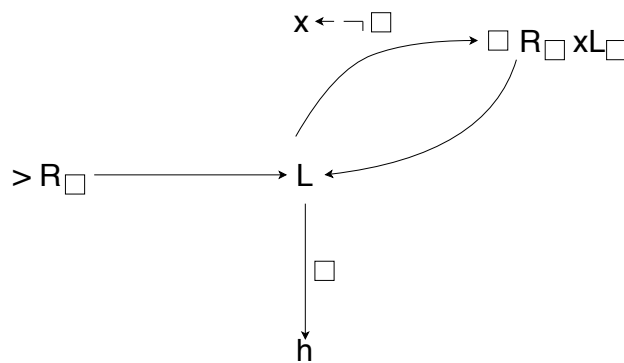
- $> R \leftrightarrow \neg \square$ μετακίνηση δεξιά. Αν το σύμβολο στην κεφαλή είναι διάφορο του $\square$ επανέλαβε. Αν είναι $\square$, δεν ορίζεται ενέργεια και άρα η εκτέλεση τερματίζεται. Συνοψίζοντας, η συγκεκριμένη MT βρίσκει την πρώτη θέση στα δεξιά της κεφαλής με το κενό σύμβολο. Εν συντομία η μηχανή γράφεται ως $R_{\square}$.
- $> L \leftrightarrow \neg \square$ μετακίνηση αριστερά. Αν το σύμβολο στη θέση της κεφαλής είναι διάφορο του $\square$ επανέλαβε. Αν το σύμβολο είναι $\square$, δεν ορίζεται ενέργεια και άρα η εκτέλεση τερματίζεται. Συνοψίζοντας, βρίσκει την πρώτη θέση στα αριστερά της κεφαλής με το κενό σύμβολο. Εν συντομία η μηχανή γράφεται ως $L_{\square}$.
- $> R \leftrightarrow \square$ Βρίσκει την πρώτη θέση στα δεξιά της κεφαλής με μη κενό σύμβολο. Εν συντομία η μηχανή γράφεται ως $R_{\neg \square}$.
- $> L \leftrightarrow \square$ Βρίσκει την πρώτη θέση στα αριστερά της κεφαλής με μη κενό σύμβολο. Εν συντομία η μηχανή γράφεται ως $L_{\neg \square}$.

Οι ίδιες λειτουργίες, που σχετίζονται με το $\square$, ορίζονται επίσης για οποιοδήποτε άλλο σύμβολο του Γ . Έτσι, έχουμε για παράδειγμα:

- L_a . Εύρεση της πρώτης εμφάνισης του a στα αριστερά της τρέχουσας θέσης.
- $R_{a,b}$. Εύρεση της πρώτης εμφάνισης του a ή του b στα δεξιά της τρέχουσας θέσης.
- $L_{a,b} \xrightarrow{a} M_1$
 $\downarrow b$. Εύρεση της πρώτης εμφάνισης του a ή του b στα αριστερά της M_2 τρέχουσας θέσης και στη συνέχεια εκτέλεση της M_1 , αν το σύμβολο στη θέση της κεφαλής είναι a . Αν το σύμβολο είναι b , τότε εκτελείται η M_2 .

- $L_{x \leftarrow a, b}$. Εύρεση της πρώτης εμφάνισης του a ή του b στα αριστερά της τρέχουσας θέσης και εκχώρηση του συμβόλου, που θα βρει, ως τιμή στη μεταβλητή x .
- $L_{x \rightarrow a, b} R x$. Εύρεση της πρώτης εμφάνισης του a ή του b στα αριστερά της τρέχουσας θέσης, εκχώρηση του συμβόλου, που θα βρει, στη μεταβλητή x . Στη συνέχεια η κεφαλή μετακινείται προς τα δεξιά και εγγράφει σε εκείνη τη θέση την τιμή της x (a ή b).

Παράδειγμα Δ'.1.3. μετατόπιση συμβολοσειράς μία θέση δεξιά Η $MT S_{\rightarrow}$ του Σχήματος Δ'.9 μετατοπίζει τη συμβολοσειρά εισόδου πάνω στην ταινία μία θέση προς τα δεξιά. Έτσι, για οποιαδήποτε συμβολοσειρά $w \in \Sigma^*$, από την αρχική διαμόρφωση $(s, \sqcup w)$ μεταβαίνει στην τελική διαμόρφωση $(h, \sqcup \sqcup w)$.



Σχήμα Δ'.9: $MT S_{\rightarrow}$ για τη μετατόπιση συμβολοσειράς μία θέση δεξιά.

Δ'.2 Υπολογισμοί με μηχανές Turing

Δ'.2.1 Μηχανές Turing ως μηχανές αναγνώρισης γλωσσών

Δοθείσης μιας γλώσσας L , μας ενδιαφέρει να σχεδιάσουμε μία $MT M$, που όταν δέχεται ως είσοδο στην ταινία μία συμβολοσειρά w αποκρίνεται με το αν η $w \in L$ είναι αληθής ή όχι. Υπάρχουν πολλές γλώσσες, για τις οποίες αυτό είναι εφικτό, ανάμεσα στις οποίες όλες οι κανονικές γλώσσες και οι γλώσσες χωρίς συμφραζόμενα, αλλά ακόμη και γλώσσες που δεν είναι χωρίς συμφραζόμενα. Από την άλλη μεριά, έχουμε περιπτώσεις γλωσσών, για τις οποίες δεν υπάρχει MT , που να διακρίνει αν μία συμβολοσειρά ανήκει στη γλώσσα ή όχι. Έχουμε όμως και περιπτώσεις γλωσσών, που το πρόβλημα λύνεται μερικώς. Για αυτή την τελευταία κατηγορία γλωσσών, μπορεί να σχεδιαστεί μία $MT M$, που ψάχνει αν μία ιδιότητα P ισχύει - για οποιαδήποτε ιδιότητα P - ώστε η συμβολοσειρά εισόδου να ανήκει στη γλώσσα L . Αν η M διαπιστώσει ότι η P ισχύει για την είσοδο στην ταινία, τότε τερματίζει και αποδέχεται την είσοδο. Αν όμως για μία είσοδο w συμβαίνει να μην ισχύει η P , τότε η M μπορεί να συνεχίζει να εκτελείται για πάντα.

Ορισμός Δ'.2.1. Έστω μία MT M με αρχική κατάσταση s και δύο τελικές καταστάσεις, που θα τις ονομάζουμε y και n . Έστω επίσης w ένα στοιχείο του Σ^* . Τότε θα λέμε ότι:

- η M αποδέχεται την w αν και μόνο αν $(s, \sqcup w) \vdash_M^* (y, w')$ για κάποια συμβολοσειρά w' . Κάθε διαμόρφωση (y, w') θα είναι διαμόρφωση αποδοχής.
- η M απορρίπτει την w αν και μόνο αν $(s, \sqcup w) \vdash_M^* (n, w')$ για κάποια συμβολοσειρά w' . Κάθε διαμόρφωση (n, w') θα είναι διαμόρφωση απόρριψης.

Τα περιεχόμενα της ταινίας της M , όταν αυτή τερματίζει δε μας ενδιαφέρουν. Σημειώνουμε επίσης ότι αν η M δεν τερματίσει, τότε δεν αποδέχεται, αλλά ούτε και απορρίπτει τη συμβολοσειρά.

Ορισμός Δ'.2.2. Έστω Σ το αλφάβητο εισόδου μιας MT M . Η M αποφασίζει μία γλώσσα $L \subseteq \Sigma^*$, αν και μόνο αν, για κάθε συμβολοσειρά $w \in \Sigma^*$ ισχύει:

- Αν $w \in L$, τότε η M αποδέχεται την w .
- Αν $w \notin L$, τότε η M απορρίπτει την w .

Αφού κάθε συμβολοσειρά του Σ^* είτε θα ανήκει, είτε δε θα ανήκει στην L , συμπεραίνουμε ότι η οποιαδήποτε μηχανή M , που αποφασίζει τη γλώσσα θα πρέπει να τερματίζει για όλες τις εισόδους. Όπως προαναφέραμε, αν για μία γλώσσα υπάρχει μία τέτοια MT, τότε αυτή είναι αποφασίσιμη ή αναδρομική. Το σύνολο όλων των αποφασίσιμων ή αναδρομικών γλωσσών συμβολίζεται με D .

Ορισμός Δ'.2.3. Έστω Σ το αλφάβητο εισόδου μιας MT M . Η M ημιαποφασίζει μία γλώσσα $L \subseteq \Sigma^*$, αν και μόνο αν, για κάθε συμβολοσειρά $w \in \Sigma^*$ ισχύει:

- Αν $w \in L$, τότε η M αποδέχεται την w .
- Αν $w \notin L$, τότε η M δεν αποδέχεται την w . Σε αυτή την περίπτωση, μπορεί η M να την απορρίπτει ρητά ή μπορεί και να μην τερματίζει ποτέ.

Αν για μία γλώσσα υπάρχει μία τέτοια MT, τότε αυτή λέμε ότι είναι ημιαποφασίσιμη ή αναδρομικά αποφασίσιμη. Το σύνολο όλων των ημιαποφασίσιμων γλωσσών συμβολίζεται με SD .

Δ'.2.2 Υπολογισμός συναρτήσεων με Μηχανές Turing

Όταν μία MT τερματίζει, τότε αφήνει πάνω στην ταινία μία συμβολοσειρά. Στις περιπτώσεις των MT, που αποφασίζουν ή ημιαποφασίζουν γλώσσες, η συμβολοσειρά που σχηματίζεται στην ταινία αγνοείται, αλλά υπάρχουν και MT για τις οποίες δε συμβαίνει αυτό. Σε αυτή την ενότητα θα ασχοληθούμε με MT, που υπολογίζουν μία συνάρτηση. Πρώτα θα ορίσουμε τη σημασία μιας MT, που υπολογίζει συνάρτηση, της οποίας το πεδίο ορισμού και το πεδίο τιμών

είναι σύνολα συμβολοσειρών. Στη συνέχεια χρησιμοποιώντας κατάλληλες κωδικοποιήσεις άλλων τύπων δεδομένων, όπως και κωδικοποιήσεις πολλαπλών τιμών, θα ορίσουμε MT, που υπολογίζουν πολλές διαφορετικές μορφές συναρτήσεων.

Έστω M μία MT με αρχική κατάσταση s , τελική κατάσταση h και αλφάβητο εισόδου Σ . Η αρχική διαμόρφωση της M θα είναι $(s, \sqsubseteq w)$, όπου $w \in \Sigma^*$. Ορίζουμε τη συνάρτηση $M(w) = z$, αν και μόνο αν $(s, \sqsubseteq w) \vdash_M^* (h, \sqsubseteq z)$, δηλαδή όταν η M ξεκινάει με είσοδο $w \in \Sigma^*$ τερματίζει αφήνοντας τη z στην ταινία με την κεφαλή ανάγνωσης/εγγραφής στη θέση ακριβώς αριστερά της z .

Ορισμός Δ'.2.4. Έστω $\Sigma' \subseteq \Gamma$ το αλφάβητο εξόδου της MT M , δηλαδή το σύνολο των συμβόλων των συμβολοσειρών, που αφήνει η M στην ταινία όταν τερματίζει. Έστω επίσης ότι f είναι μία συνάρτηση από το Σ^* στο Σ'^* . Η MT M υπολογίζει τη συνάρτηση f , αν και μόνο αν για όλες τις $w \in \Sigma^*$:

- Αν w είναι μία είσοδος για την οποία ορίζεται η f , $M(w) = f(w)$, δηλαδή η M τερματίζει αφήνοντας πάνω στην ταινία την $f(w)$.
- Διαφορετικά η $M(w)$ δεν τερματίζει.

μία συνάρτηση f είναι αναδρομική ή υπολογίσιμη αν υπάρχει MT M , που την υπολογίζει και τερματίζει σε όλες τις περιπτώσεις. Στη σχετική βιβλιογραφία μπορεί μία συνάρτηση f να αποκαλείται υπολογίσιμη ακόμη και αν υπάρχει μία MT M , που την υπολογίζει χωρίς απαραίτητα να τερματίζει σε όλες τις περιπτώσεις. Πιο συγκεκριμένα, αν υπάρχουν τιμές για τις οποίες η f δεν ορίζεται, τότε η M αποτυγχάνει να τερματίσει για αυτές τις τιμές. Σε αυτή την περίπτωση η f είναι μερικώς υπολογίσιμη και θα κρατήσουμε τον όρο υπολογίσιμη μόνο για τις συναρτήσεις του παραπάνω ορισμού.

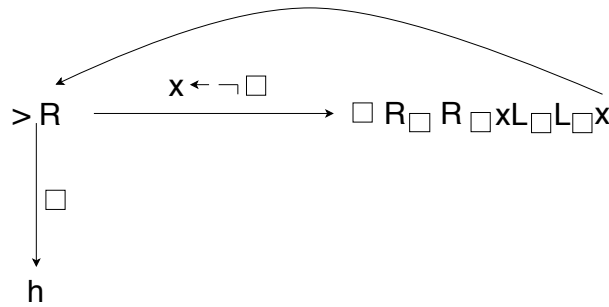
Υπάρχει μία αναλογία μεταξύ των MT, που υπολογίζουν συναρτήσεις και των MT, που αποφασίζουν γλώσσες. μία γλώσσα είναι αποφασίσιμη, αν και μόνο αν η χαρακτηριστική της συνάρτηση είναι υπολογίσιμη, δηλαδή αν υπάρχει μία MT, που τερματίζει σε όλες τις περιπτώσεις με έξοδο True αν η είσοδός της ανήκει στην L και έξοδο False αν δεν ανήκει.

Παράδειγμα Δ'.2.1. Συνάρτηση αντιγραφής συμβολοσειράς Θα ορίσουμε μία MT M , που υπολογίζει τη συνάρτηση $copy(w) = ww$, για οποιαδήποτε συμβολοσειρά w , που δεν περιέχει το $\square$. Πρώτα ορίζουμε τη μηχανή αντιγραφής C , που απεικονίζεται στο Σχήμα. Για τη C έχουμε $(s, \sqsubseteq w) \vdash_M^* (h, w \sqsubseteq w)$. Η MT M , που υπολογίζει τη συνάρτηση $copy$ είναι η

$$> CS \leftarrow L \square \quad (\Delta'.9)$$

όπου $S \leftarrow$ είναι η μηχανή μετατόπισης συμβολοσειράς προς τα αριστερά, αντίστοιχη προς την $S \rightarrow$, που περιγράψαμε σ'αλλο παράδειγμα.

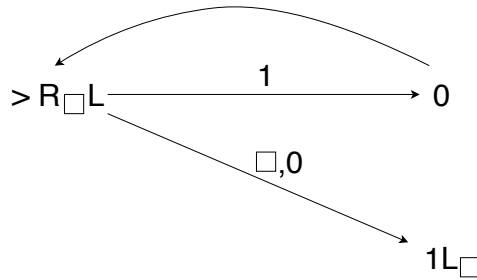
Επικεντρωνόμαστε τώρα στον υπολογισμό συναρτήσεων με τιμές, που δεν είναι συμβολοσειρές. Σε αυτή την περίπτωση χρειαζόμαστε μία κωδικοποίηση των τιμών των συναρτήσεων



Σχήμα Δ'.10: $MT C$ για την αντιγραφή συμβολοσειράς.

ως συμβολοσειρές. Ορίζουμε λοιπόν μία οικογένεια συναρτήσεων $val_k(w)$, τέτοιες ώστε για οποιονδήποτε ακέραιο k η $val_k(w)$ επιστρέφει τον μη αρνητικό ακέραιο, που κωδικοποιείται με βάση k από τη συμβολοσειρά w . Για παράδειγμα, $val_2(11) = 3$ και $val_8(11) = 9$.

Παράδειγμα Δ'.2.2. Συνάρτηση «επόμενος αριθμός» Σχεδιάζουμε μία $MT M$, που θα υπολογίζει τη συνάρτηση $succ(n) = n + 1$. Ο αριθμός n θα αναπαριστάται στο δυαδικό σύστημα χωρίς να προηγούνται περιττά 0. Επομένως, για $n \in 0 \cup 1\{0, 1\}^*$ θα έχουμε $succ(n) = m$, όπου $val_2(m) = val_2(n) + 1$. Στο Σχήμα Δ'.11 απεικονίζεται η ζητούμενη $MT M$, που λειτουργεί ως



Σχήμα Δ'.11: MT για τον υπολογισμό του «επόμενου» αριθμού.

εξής:

1. Σαρώνει την ταινία δεξιά μέχρι να εντοπίσει $\square$. Στη συνέχεια μετακινεί την κεφαλή μία θέση προς τα αριστερά, έτσι ώστε αυτή να βρεθεί στο τελευταίο ψηφίο της δυαδικής αναπαράστασης του n .
2. Επανάλαβε τα παρακάτω βήματα:
 - (α') Αν το ψηφίο στη θέση της κεφαλής είναι το 0, τότε εγγράφεται 1, η κεφαλή μετακινείται στη θέση του πρώτου κενού προς τα αριστερά και εκεί τερματίζει.
 - (β') Αν το ψηφίο στη θέση της κεφαλής είναι το 1, τότε αυτό πρέπει να μεταφερθεί προς τα αριστερά. Έτσι, εγγράφεται στη θέση εκείνη το 0, η κεφαλή μετακινείται μία θέση προς τα αριστερά και η εκτέλεση συνεχίζεται με τον επόμενο κύκλο της επανάληψης.

(γ') Αν το ψηφίο στη θέση της κεφαλής είναι το $\square$, τότε η κεφαλή βρίσκεται τέρμα αριστερά. Εγγράφεται 1 και στη συνέχεια μετακινείται μία θέση προς τα αριστερά, όπου και τερματίζει.

Η ΜΤ M υπολογίζει μία συνάρτηση f από το $\mathbb{N}_i$ στο $\mathbb{N}$ εφόσον για κάποιο

$$k, \text{val}_k(M(w_1; w_2; \dots w_i)) = f(\text{val}_k(w_1), \dots, \text{val}_k(w_i))$$

Μπορούμε δηλαδή να σχεδιάσουμε ΜΤ, που υπολογίζουν συναρτήσεις δύο ή περισσότερων μεταβλητών, κωδικοποιώντας κάθε μεταβλητή ως συμβολοσειρά και παραθέτοντας στην ταινία τη μία μετά την άλλη με το διαχωριστικό ”;”.

Δ'.3 Πολλές Ταινίες και μη Ντετερμινισμός

Δ'.3.1 Πολλές ταινίες

Μία ΜΤ k ταινιών ορίζεται από μια εξάδα $M = (K, \Sigma, \Gamma, \delta, s, H)$, όπως και μία απλή ΜΤ μιας ταινίας. μία διαμόρφωση σε ΜΤ k ταινιών ορίζεται ως $k + 1$ -άδα (κατάσταση, ταινία₁, ..., ταινία_k). Η αρχική διαμόρφωση μιας ΜΤ M k ταινιών θα δίνεται ως $(s, \square w, \square, \dots, \square)$, δηλαδή η είσοδος θα βρίσκεται τοποθετημένη στην πρώτη ταινία, ενώ οι άλλες ταινίες θα είναι αρχικά κενές με τις κεφαλές ανάγνωσης/εγγραφής να δείχνουν σε κάποιες θέσεις με το κενό σύμβολο. Αν η M τερματίζει, τότε θεωρούμε ότι η έξοδος της δίνεται από τα περιεχόμενα της πρώτης ταινίας, ενώ τα περιεχόμενα όλων των άλλων ταινιών αγνοούνται.

Σε κάθε βήμα, η M θα εξετάζει τη θέση στην οποία βρίσκεται κάθε μία από τις k κεφαλές ανάγνωσης/εγγραφής και ανάλογα με την τρέχουσα κατάσταση θα καθορίζει την επόμενη μετάβαση. Θα εγγράφει και θα μετακινεί την κεφαλή σε κάθε μία από τις ταινίες ταυτόχρονα. μερικές φορές μπορεί η M να μετακινεί τις κεφαλές σε μία ή περισσότερες ταινίες, χωρίς να μετακινεί τις κεφαλές στις υπόλοιπες. Για αυτό το λόγο, επιτρέπουμε στις ΜΤ πολλών ταινιών μία επιπλέον ενέργεια εκτός από τις $\leftarrow$ και $\rightarrow$, που είναι η παραμονή στη ίδια θέση οριζόμενη με το σύμβολο $\uparrow$. Έτσι η συνάρτηση μετάβασης δ για μία ΜΤ k ταινιών θα ορίζεται ως συνάρτηση:

$$\begin{array}{ll} \text{από το } ((K?H) & \times \Gamma_1 \quad \text{προς το } (K & \times \Gamma_1 \times \{\leftarrow, \rightarrow, \uparrow\} \\ & \times \Gamma_2 & \times \Gamma_2 \times \{\leftarrow, \rightarrow, \uparrow\} \\ & \times \dots & \times \dots \\ & \times \dots & \times \dots \\ & \times \Gamma_k) & \times \Gamma_k \times \{\leftarrow, \rightarrow, \uparrow\}) \end{array}$$

Θεώρημα Δ'.3.1. Έστω $M = (K, \Sigma, \Gamma, \delta, s, H)$ μία ΜΤ k ταινιών, για κάποιο $k > 1$. Θα υπάρχει μία απλή ΜΤ $M' = (K', \Sigma', \Gamma', \delta', s', H')$ με $\Gamma \subseteq \Gamma'$, τέτοια ώστε να ισχύουν οι παρακάτω συνθήκες:

- Για οποιαδήποτε συμβολοσειρά εισόδου w , η M τερματίζει με έξοδο z στην πρώτη ταινία, αν και μόνο αν η M' τερματίζει με είσοδο w στην ίδια τελική κατάσταση (y, n ή h) με έξοδο z στην ταινία της.
- Αν η M τερματίζει με είσοδο w μετά από n βήματα, τότε η M' τερματίζει το πολύ σε $O(n^2)$ βήματα.

Απόδειξη: Η

απόδειξη στηρίζεται σε κατάλληλη κατασκευή. Θα επιδιώξουμε τη σχεδίαση της M' , ώστε αυτή να προσομοιώνει τις k ταινίες της M χρησιμοποιώντας τη μία και μοναδική ταινία της ως αν αυτή να έχει διαιρεθεί σε γραμμές. Αν υποθέσουμε ότι η M έχει k ταινίες, τότε μία διατεταγμένη k -άδα τιμών περιγράφει το περιεχόμενο της κάθε ταινίας σε μία συγκεκριμένη θέση της. Χρειάζεται επίσης να καταγράφεται η θέση της κάθε μιας από τις k κεφαλές ανάγνωσης/εγγραφής. Αυτό το επιτυγχάνουμε αφιερώνοντας δύο γραμμές για κάθε μία από τις ταινίες της M . Η πρώτη γραμμή περιέχει τα σύμβολα σε κάθε θέση μιας ταινίας και η δεύτερη γραμμή περιέχει 1, στη θέση στην οποία βρίσκεται η κεφαλή ανάγνωσης/εγγραφής και 0 σε όλες τις υπόλοιπες θέσεις. Επειδή όλες οι ταινίες της M είναι άπειρες θα χρειαστούμε μία προσέγγιση στοίχισης των ταινιών, έτσι ώστε να αναπαριστούμε μόνο το μέρος εκείνο των ταινιών, που μας ενδιαφέρει. Αυτό επιτυγχάνεται ξεκινώντας με την αρχική διαμόρφωση της M και στη συνέχεια στοιχίζοντας όλες τις ταινίες, έτσι ώστε οι κεφαλές ανάγνωσης/εγγραφής να βρίσκονται όλες στην ίδια στήλη.

...	□	□	a	b	a	a	□	□	□	□	...
		↑									

...	□	□	□	□	□	□	□	□	□	□	...
		↑									

□	□	a	b	a	a	□	□	□	□	...
□	1	0	0	0	0	0	0	0	□	□
□	□	□	□	□	□	□	□	□	□	□
□	1	0	0	0	0	0	0	□	□	□

Για να γίνει ευκολότερα αντιληπτή η κατασκευή, θεωρούμε ότι $k = 2$. Τότε η αρχική διαμόρφωση της M είναι αυτή, που φαίνεται στο πάνω μέρος της αναπαράστασης, η M' θα αναπαριστά τις δύο ταινίες της M διαιρώντας τη μοναδική της ταινία σε τέσσερις γραμμές, όπως φαίνεται στο κάτω μέρος της αναπαράστασης.

Η ταινία της M' , όπως ακριβώς και η ταινία της οποιασδήποτε άλλης MT, θα περιέχει □ σε όλες τις θέσεις εκτός από έναν πεπερασμένο αριθμό θέσεων, που αρχικά θα είναι ίσος με το μήκος της συμβολοσειράς εισόδου w . Όταν όμως οποιαδήποτε από τις κεφαλές ανάγνωσης/εγγραφής της M μετακινείται αριστερά ή δεξιά στην περιοχή με τα κενά σύμβολα, η

M' θα πρέπει να περιλάβει τη νέα θέση της κεφαλής στην αναπαράσταση, που χρησιμοποιεί στην ταινία της με τις πολλές γραμμές.

Όπως όλες οι απλές MT, αρχικά η M' θα βρει στην ταινία της τη συμβολοσειρά εισόδου. Θα ξεκινήσει αναδιαμορφώνοντας την ταινία της, ώστε αυτή να κωδικοποιεί τις k γραμμές. Ο υπολογισμός συνεχίζει με την αναδιαμορφωμένη ταινία μέχρι η MT να τερματίσει. Στο τέλος η μηχανή αναδιαμορφώνει πάλι την ταινία της, έτσι ώστε το αποτέλεσμα, δηλαδή η συμβολοσειρά της πρώτης από τις ταινίες που προσομοιώνονται, να εγγραφεί στην ταινία χωρίς την κωδικοποίηση των γραμμών. Κατά συνέπεια, η M' χρειάζεται ένα κατάλληλο αλφάβητο ταινίας, που να μπορεί να κωδικοποιεί τις φάσεις εκείνες στις οποίες αποθηκεύεται μόνο ένα σύμβολο σε κάθε θέση της ταινίας (αρχική και τελική), καθώς και τη φάση κατά την οποία σε κάθε θέση της ταινίας έχουμε k τιμές μαζί με k bits για τις κεφαλές ανάγνωσης/εγγραφής των ταινιών, που προσομοιώνονται. Έτσι, η M' χρειάζεται για την ταινία της ένα αλφάβητο με ένα σύμβολο για κάθε στοιχείο του $\Gamma \cup (\Gamma \times \{0, 1\})^k$. Συνολικά ο αριθμός των συμβόλων του Γ' θα είναι $|\Gamma'| = |\Gamma| + (2 \cdot |\Gamma|)^k$.

Με την κατασκευή, που παρουσιάσαμε, αποδεικνύεται ότι ένας οποιοσδήποτε υπολογισμός MT k ταινιών μπορεί τελικά να εκτελεστεί από μία MT μιας μόνο ταινίας. Αυτό σημαίνει ότι η προσθήκη ενός πεπερασμένου αριθμού ταινιών δεν προσθέτει επιπλέον εκφραστική ισχύ στο μοντέλο υπολογισμού των MT.

Υπάρχει όμως μία διαφορά: η MT μιας ταινίας πρέπει να εκτελεί πολλά διαδοχικά βήματα, για ένα μόνο βήμα, που εκτελεί η MT k ταινιών. Το ερώτημα του προσδιορισμού του αριθμού των βημάτων, που χρειάζονται, μπορεί να απαντηθεί μόνο αν η M και επομένως και η M' τερματίζουν. Έτσι, αν η M τερματίζει, ας υποθέσουμε ότι αυτό γίνεται για μία συμβολοσειρά εισόδου w σε n βήματα. Υποθέτουμε ότι πριν από κάθε βήμα η κεφαλή ανάγνωσης/εγγραφής της M' βρίσκεται στο $\square$, που υπάρχει στη θέση ακριβώς δεξιά από το διαιρεμένο σε γραμμές μέρος της ταινίας. Κάθε φορά, που η M' προσομοιώνει ένα από τα n βήματα στον υπολογισμό της M θα πρέπει να σαρώσει δύο φορές το μη κενό μέρος της ταινίας: στην πρώτη σάρωση με κατεύθυνση προς τα αριστερά διαβάζονται τα σύμβολα για την k -άδα, που αντιστοιχεί στις θέσεις των k κεφαλών ανάγνωσης/εγγραφής και στη δεύτερη σάρωση προς τα αριστερά ενημερώνεται η κάθε γραμμή, όπως προβλέπεται από τη σχέση μετάβασης της M . Όμως πόσο μεγάλο είναι το μήκος του μη κενού τμήματος της ταινίας, που σαρώνεται δύο φορές; Αρχικά το μέρος αυτό είναι μήκους $|w|$, αλλά στη συνέχεια εφόσον η M εγγράφει έξω από την αρχική περιοχή της εισόδου θα πρέπει η M' να επεκτείνει το κωδικοποιημένο μέρος της ταινίας της. Παρατηρούμε ότι η M και κατά συνέπεια η M' μπορούν να εγγράψουν σε το πολύ μία επιπλέον θέση για κάθε ένα από τα n βήματα και επομένως ένα άνω φράγμα στο μήκος του κωδικοποιημένου μέρους της ταινίας είναι το $|w| + n$.

Με αυτά τα δεδομένα, μπορούμε να υπολογίσουμε ότι ο αριθμός των βημάτων, που απαιτούνται για την M' στην προσομοίωση του υπολογισμού της M με είσοδο w είναι $\mathcal{O}(n \cdot (|w| + n))$. Γενικά ισχύει $n \geq |w|$, ακόμη και αν η M επισκέπτεται κάθε θέση της εισόδου τουλάχιστο μία φορά, οπότε ο συνολικός αριθμός των βημάτων, που εκτελεί η M' είναι τελικά $\mathcal{O}(n^2)$.

Δ'.3.2 Μη ντετερμινισμός

Μέχρι στιγμής αναφερθήκαμε μόνο σε ντετερμινιστικές MT. Σχετικά λοιπόν με τις MT έχουμε:

- Ο μη ντετερμινισμός δεν προσθέτει επιπλέον υπολογιστικές δυνατότητες, με την έννοια ότι οποιοσδήποτε υπολογισμός μιας μη ντετερμινιστικής MT, μπορεί να εκτελείται επίσης και από μία αντίστοιχη ντετερμινιστική MT.
- Ένας υπολογισμός σε ντετερμινιστική MT μπορεί να χρειάζεται εκθετικά περισσότερα βήματα για την επίλυση ενός προβλήματος από τα βήματα, που απαιτούνται σε μία μη ντετερμινιστική MT.

Ορισμός Δ'.3.1. Μία μη ντετερμινιστική MT περιγράφεται από μία εξάδα $(K, \Sigma, \Gamma, \Delta, s, H)$, όπου K, Σ, Γ, s και H ορίζονται όπως στις απλές MT και Δ είναι ένα υποσύνολο του $((K \cup H) \times \Gamma) \times (K \times \Gamma \times \{\leftarrow, \rightarrow\})$.

Αντικαταστήσαμε δηλαδή τη συνάρτηση μετάβασης δ από μία σχέση μετάβασης Δ . Η βασική διαφορά με το μη ντετερμινισμό στις MT είναι ότι εφόσον η λειτουργία της μηχανής δεν είναι απλά η ανάγνωση και κατανάλωση των συμβόλων εισόδου ένα προς ένα, η έννοια των μεταβάσεων-ε είναι πλέον περιττή.

Ορισμός Δ'.3.2. Έστω $M = (K, \Sigma, \Gamma, \Delta, s, H)$ μία μη ντετερμινιστική MT. Έστω επίσης w ένα στοιχείο του Σ^* . Τότε θα λέμε ότι:

- η M αποδέχεται την w αν και μόνο αν τουλάχιστον ένας από τους υπολογισμούς της πράγματι την αποδέχεται και
- η M απορρίπτει την w αν και μόνο αν όλοι οι υπολογισμοί της πράγματι την απορρίπτουν.

Ορισμός Δ'.3.3. Μία μη ντετερμινιστική MT M αποφασίζει μία γλώσσα $L \subseteq \Sigma^*$, αν και μόνο αν, για κάθε συμβολοσειρά $w \in \Sigma^*$:

- Υπάρχει ένας πεπερασμένος αριθμός μονοπατιών, που μπορεί να ακολουθήσει η M με είσοδο w .
- Όλα αυτά τα μονοπάτια είναι υπολογισμοί, δηλαδή τερματίζουν.
- Ισχύει $w \in L$, αν και μόνο αν η M αποδέχεται την w .

Ορισμός Δ'.3.4. Μία μη ντετερμινιστική MT $M = (K, \Sigma, \Gamma, \Delta, s, H)$ ημιαποφασίζει μία γλώσσα $L \subseteq \Sigma^*$, αν και μόνο αν, για κάθε συμβολοσειρά $w \in \Sigma^*$:

- Ισχύει $w \in L$, αν και μόνο αν από την αρχική διαμόρφωση $(s, \sqcup w)$ υπάρχει τουλάχιστον ένας υπολογισμός, που οδηγεί σε τελική διαμόρφωση, δηλαδή υπάρχει τουλάχιστον ένα μονοπάτι, που τερματίζει και η M αποδέχεται την w .

Θα δώσουμε ένα παράδειγμα, με την περίπτωση μηχανών, που οι συμβολοσειρές εισόδων τους είναι περιγραφές MT! Θα χρησιμοποιούμε το συμβολισμό $\langle M \rangle$ για να εκφράσουμε τη συμβολοσειρά, που περιγράφει μία MT M .

Παράδειγμα Δ'.3.1. Ημιαπόφαση με προσομοίωση

• Έστω η γλώσσα

$$L = \{ \langle M \rangle : \eta M \text{ είναι μία MT, που τερματίζει για τουλάχιστο μία συμβολοσειρά} \}$$

Το πώς μία MT μπορεί να προσομοιώσει μία άλλη MT θα περιγραφεί στις επόμενες ενότητες. Αν υποθέσουμε ότι κάτι τέτοιο είναι εφικτό, η MT S , που ημιαποφαίζει την L θα πρέπει να λειτουργεί ως εξής, για μία είσοδο $\langle M \rangle$:

1. Επιλέγεται μη ντετερμινιστικά μία συμβολοσειρά $w \in \Sigma^*$, που εγγράφεται στην ταινία.
2. Εκτελείται η M για την w .
3. Γίνεται αποδοχή.

Αν υπάρχει τουλάχιστο μία συμβολοσειρά για την οποία η M τερματίζει, τότε θα υπάρχει τουλάχιστο ένα μονοπάτι της μηχανής S , που θα τερματίζει και θα αποδέχεται την $\langle M \rangle$.

Για κάποια προβλήματα, όπως εδώ το πρόβλημα αποδοχής της L , η ημιαπόφαση είναι ότι καλύτερο είμαστε σε θέση να έχουμε. Η προσέγγιση, που ακολουθήσαμε στη σχεδίαση της S , δηλαδή η προσομοίωση μιας άλλης μηχανής για να διαπιστωθεί αν αυτή τερματίζει, χρησιμοποιείται ευρέως, όταν δε μπορούμε να έχουμε καλύτερη λύση από την ημιαπόφαση.

Σε ότι αφορά τον υπολογισμό συναρτήσεων από μη ντετερμινιστικές MT, ο ορισμός πρέπει να αντιμετωπίσει το ζήτημα της ύπαρξης μονοπατιών για μία είσοδο w , που επιστρέφουν διαφορετικές τιμές στην έξοδο. Δεδομένου ότι κάθε συνάρτηση παίρνει μόνο μία τιμή στο πεδίο των τιμών της, ποια από τις τιμές των διαφορετικών μονοπατιών θα πρέπει να είναι το αποτέλεσμα της συνάρτησης; Για να ορίσουμε τον υπολογισμό συνάρτησης από μία μη ντετερμινιστική MT ακολουθούμε την παρακάτω αυστηρή προσέγγιση.

Ορισμός Δ'.3.5. μία μη ντετερμινιστική MT $M = (K, \Sigma, \Gamma, \Delta, s, H)$ υπολογίζει μία συνάρτηση f , αν και μόνο αν για κάθε $w \in \Sigma^*$:

- Όλα τα μονοπάτια, που μπορεί να ακολουθήσει η M για την είσοδο w τερματίζουν, δηλαδή είναι υπολογισμοί.
- Όλοι οι υπολογισμοί της M δίνουν ως αποτέλεσμα την τιμή $f(w)$.

Το θεώρημα, που αναφέρουμε στη συνέχεια αποδεικνύει το βασικό αποτέλεσμα, που τονίσαμε στην αρχή αυτής της ενότητας: ο μη ντετερμινισμός δεν προσθέτει επιπλέον υπολογιστικές δυνατότητες στο βασικό μοντέλο της MT. μπορεί η σχεδίαση μη ντετερμινιστικών MT να είναι πιο εύκολη και η εκτέλεσή τους να είναι σημαντικά πιο γρήγορη, αλλά δεν υπάρχει κάτι, που δε μπορεί να εκτελεστεί από μια ισοδύναμη ντετερμινιστική MT.

Θεώρημα Δ'.3.2. *μη ντετερμινιστικές MT που (ημι)αποφασίζουν. Αν μία μη ντετερμινιστική $MT\ M = (K, \Sigma, \Gamma, \Delta, s, H)$ αποφασίζει μία γλώσσα L , τότε υπάρχει επίσης μία ντετερμινιστική $MT\ M'$, που αποφασίζει την L . Αν μία μη ντετερμινιστική $MT\ M$ ημιαποφασίζει μία γλώσσα L , τότε υπάρχει επίσης μία ντετερμινιστική $MT\ M'$, που ημιαποφασίζει την L .*

Απόδειξη: Η

απόδειξη στηρίζεται σε κατάλληλη κατασκευή. μία κατασκευή της ντετερμινιστικής μηχανής M' από σύνολα καταστάσεων της μη ντετερμινιστικής μηχανής M , προσέγγιση που ακολουθήσαμε για την απόδειξη της ισοδυναμίας των ΝΠΑ με τα ΜΝΠΑ, δεν είναι κατάλληλη. Στην περίπτωση των MT είμαστε αναγκασμένοι να λαμβάνουμε υπόψη και το περιεχόμενο της ταινίας, καθώς κάθε πιθανό μονοπάτι της μη ντετερμινιστικής μηχανής χρειάζεται να χρησιμοποιεί το δικό του αντίγραφο της ταινίας. Επιπλέον, η κατασκευή του Θεωρήματος που σημειώσαμε με την κωδικοποίηση πολλών ταινιών σε μία ταινία με πολλές γραμμές δεν είναι κατάλληλη, αφού σε κείνη την περίπτωση ο αριθμός των ταινιών, που πρέπει να κωδικοποιηθούν είναι περιορισμένος και εκ των προτέρων γνωστός, κάτι που δεν ισχύει για τον αριθμό των μονοπατιών της μη ντετερμινιστικής MT M .

Προσανατολιζόμαστε λοιπόν σε μία κατασκευή, που θα υλοποιεί μία προσέγγιση αναζήτησης κατά μήκος των διαφορετικών μονοπατιών της μη ντετερμινιστικής MT. Δεδομένης μιας εισόδου w , η M' θα επιχειρήσει να εκτελέσει συστηματικά όλα τα δυνατά μονοπάτια της M , αναζητώντας κάποιο, που τερματίζει. Όταν και εάν ανακαλύψει έναν τέτοιο υπολογισμό, θα τερματίσει και αυτή. Έτσι, η M' θα τερματίσει αν και μόνο αν η M τερματίζει. Η αναζήτηση σε βάθος (depth-first search) δεν είναι κατάλληλη, επειδή μπορεί να αρχίζει από μονοπάτια, που δεν τερματίζουν, αν βέβαια μιλάμε για MT, που ημιαποφασίζει.

Προχωράμε λοιπόν στην κατασκευή μιας ντετερμινιστικής MT M' , που θα εκτελεί αναζήτηση σε εύρος (breadth-first search) μέσα στο δέντρο των μονοπατιών, που δημιουργεί η εκτέλεση της M . Εδώ είναι σημαντικό να παρατηρήσουμε το εξής: για κάθε διαμόρφωση C της M , το πλήθος των διαμορφώσεων C' για τις οποίες ισχύει $C \vdash_M C'$ είναι σταθερό και φραγμένο με τέτοιο τρόπο, που εξαρτάται από την M και όχι από την C . Αυτό συμβαίνει, επειδή το πλήθος των διατεταγμένων ζευγών $((p, a), (q, b, d)) \in \Delta$, που μπορεί να εφαρμοστεί σε κάθε βήμα είναι πεπερασμένο και δε μπορεί να είναι μεγαλύτερο από $|K| \cdot |\Sigma| \cdot 2$, που εκφράζει το μέγιστο πλήθος δυνατών συνδυασμών (q, b, d) με $q \in K, b \in \Gamma$ και $d \in \{\rightarrow, \leftarrow\}$. Ας ονομάσουμε r το μέγιστο πλήθος διατεταγμένων ζευγών, που μπορεί να εφαρμοστεί κάθε στιγμή, όπου r μπορεί να εκτιμηθεί με παρατήρηση της M . Ας υποθέσουμε επίσης ότι το μήκος του μεγαλύτερου μονοπατιού, που θα πρέπει να ακολουθήσει η M πριν να αποδεχθεί την είσοδο είναι n . Τότε η M' μπορεί να χρειάζεται $O(r^{n+1})$ κινήσεις, για να βρει μία λύση, αφού ίσως χρειαστεί να εξερευνήσει όλο το δέντρο των πιθανών μονοπατιών ύψους n .

Υποθέτουμε λοιπόν ότι για κάθε συνδυασμό κατάστασης - συμβόλου (p, a) και για κάθε ακέραιο $i \in \{1, 2, \dots, r\}$ υπάρχει ένα σαφώς ορισμένο διατεταγμένο ζεύγος $((p, a), (q_i, b_i, d_i))$. Αν για κάποιο συνδυασμό κατάστασης - συμβόλου (p, a) υπάρχουν λιγότερα από r σχετικά

διατεταγμένα ζεύγη στη Δ , τότε χωρίς βλάβη της γενικότητας μπορεί κάποια να επαναληφθούν (μας ενδιαφέρει η εξαντλητική αναζήτηση).

Η M είναι μη ντετερμινιστική και επομένως δεν είναι καθορισμένος ο τρόπος με τον οποίο η μηχανή θα επιλέγει σε κάθε βήμα ανάμεσα στις r διαθέσιμες επιλογές. Έστω λοιπόν M_d μία ντετερμινιστική παραλλαγή της M , που στηρίζεται στο ίδιο σύνολο καταστάσεων, αλλά χρησιμοποιεί δύο ταινίες. Η πρώτη ταινία είναι η ταινία της M , που αρχικά περιέχει την είσοδο w , ενώ η δεύτερη ταινία αρχικά περιέχει μία συμβολοσειρά από n ακέραιους στο διάστημα $1, \dots, r$, έστω τους $i_1 i_2 \dots i_n$. Η M_d λοιπόν θα λειτουργεί ως εξής για n βήματα: στο πρώτο βήμα, ανάμεσα στους r πιθανούς συνδυασμούς $(q_1, b_1, d_1), \dots, (q_r, b_r, d_r)$, που μπορούν να εφαρμοστούν στην αρχική διαμόρφωση, η M_d επιλέγει τον i_1 -οστό συνδυασμό, που υποδεικνύεται από το σύμβολο i_1 , που μόλις διαβάστηκε στη δεύτερη ταινία και η νέα διαμόρφωση σχηματίζεται εκτελώντας τη μετάβαση $(q_{i_1}, b_{i_1}, d_{i_1})$. Ταυτόχρονα, η M_d κινεί την κεφαλή της δεύτερης ταινίας προς τα δεξιά, έτσι ώστε στη συνέχεια να διαβάσει το i_2 . Στο επόμενο βήμα, η M_d παίρνει τον i_2 -οστό συνδυασμό, μετά τον i_3 -οστό, κ.ο.κ. Όταν η M_d συναντήσει ένα κενό στη δεύτερη ταινία της, που σημαίνει ότι της έχουν τελειώσει οι υποδείξεις, τότε τερματίζει.

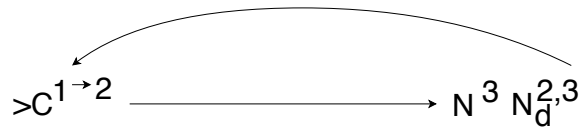
Θα περιγράψουμε τη ντετερμινιστική MT M' , ως μία MT τριών ταινιών, αφού γνωρίζουμε από το πρώτο Θεώρημα ότι η M' μπορεί να μετατραπεί σε μία ισοδύναμη MT μια ταινίας. Οι τρεις ταινίες της M' χρησιμοποιούνται ως εξής:

1. Η πρώτη ταινία περιέχει πάντα την αρχική είσοδο w , έτσι ώστε κάθε προσομοιούμενος υπολογισμός της M να μπορεί να ξεκινήσει εκ νέου με την ίδια είσοδο.
2. Η δεύτερη και η τρίτη ταινία χρησιμοποιούνται, για να προσομοιώνουν την M_d , με όλες τις συμβολοσειρές της τρίτης ταινίας να ανήκουν στη $\{1, 2, \dots, r\}^*$. Πριν από την προσομοίωση κάθε νέου μονοπατιού από την M αντιγράφεται η είσοδος από την πρώτη ταινία στη δεύτερη. Αρχικά η τρίτη ταινία περιέχει την ϵ .
3. Ανάμεσα σε δύο προσομοιώσεις της M_d , η M' χρησιμοποιεί μία MT N , για να παράξει την επόμενη λεξικογραφικά συμβολοσειρά του $\{1, 2, \dots, r\}^*$. Έτσι, αρχίζοντας από την ϵ , η N θα παράξει τις συμβολοσειρές

$$1, 2, \dots, r, 11, 12, \dots, rr, 111, \dots$$

.

Η M' είναι η MT, που απεικονίζεται στο Σχήμα Δ'.12 Η $C^{1 \rightarrow 2}$ συμβολίζει μία απλή MT, η οποία διαγράφει τα περιεχόμενα της δεύτερης ταινίας και αντιγράφει τα περιεχόμενα της πρώτης στη δεύτερη. Η N^3 παράγει την επόμενη λεξικογραφικά συμβολοσειρά στην τρίτη ταινία και η $M^{2,3}$ είναι η ντετερμινιστική παραλλαγή της M , η οποία λειτουργεί στη δεύτερη και στην τρίτη ταινία.



Σχήμα Δ'.12: Ντετερμινιστική MT M' για τη μη ντετερμινιστική MT M

Ισχυριζόμαστε ότι η M' τερματίζει με μία είσοδο w , αν και μόνο αν κάποιο μονοπάτι της M τερματίζει. Ας υποθέσουμε ότι η M' πράγματι τερματίζει με είσοδο w . Τότε, αυτό σημαίνει ότι για κάποια συμβολοσειρά $i_1 i_2 \dots i_n \in \{1, 2, \dots, r\}^*$ της τρίτης ταινίας, όταν η M_d ξεκινάει με την w στη δεύτερη ταινία, τερματίζει πριν η κεφαλή της τρίτης ταινίας οδηγηθεί στα κενά σύμβολα. Αυτό σημαίνει ότι υπάρχει ένα μονοπάτι της M με είσοδο w , που τερματίζει (υπολογισμός).

Αντιστρόφως, αν υπάρχει μονοπάτι της M με είσοδο w που τερματίζει, έστω σε n βήματα, τότε στην M' η συμβολοσειρά του $\{1, 2, \dots, r\}^*$, που αντιστοιχεί στις επιλογές του μονοπατιού της M , που τερματίζει, θα παραχθεί μια κάποια στιγμή από την N^3 και η M_d θα τερματίσει, όταν θα διαβάσει το τελευταίο σύμβολο αυτής. Επομένως, θα τερματίσει και η M' .

Μη ντετερμινιστικές Μηχανές Turing που υπολογίζουν. Αν μία μη ντετερμινιστική MT $M = (K, \Sigma, \Gamma, \Delta, s, H)$ υπολογίζει μία συνάρτηση f , τότε υπάρχει επίσης μία ντετερμινιστική MT M' , που υπολογίζει την f .

Απόδειξη: Η

απόδειξη στηρίζεται σε κατάλληλη κατασκευή, που είναι ανάλογη με την κατασκευή στην απόδειξη του Θεωρήματος 3.2.

Δ'.4 Προσομοίωση Υπολογιστή με MT

Είδαμε ότι η προσθήκη ταινιών στις απλές MT δεν προσθέτει επιπλέον υπολογιστικές δυνατότητες και το ίδιο συμβαίνει επίσης όταν μία MT χρησιμοποιεί μη ντετερμινιστική επιλογή.

• Έχει όμως ενδιαφέρον να διερευνήσουμε αν η προσθήκη πανομοιότυπων χαρακτηριστικών με αυτά ενός πραγματικού υπολογιστή θα δημιουργούσε ένα ακόμη πιο ισχυρό μοντέλο υπολογισμού. Τα χαρακτηριστικά ενός απλού υπολογιστή, που θα μπορούσαν να προστεθούν σε μία MT είναι τα παρακάτω:

- Απεριόριστος αριθμός κελιών μνήμης, που διευθυνσιοδοτούνται με ακέραιους αριθμούς, που αρχίζουν από το 0. Αυτά τα κελιά μνήμης μπορεί να χρησιμοποιούνται, τόσο για την αποθήκευση εντολών προγράμματος, όσο και για την αποθήκευση δεδομένων. Υποθέτουμε ότι δεν υπάρχει όριο στον αριθμό των bits, που αποθηκεύονται σε κάθε κελί.
- Ένα σύνολο εντολών για τις βασικές πράξεις, που περιλαμβάνουν την ανάγνωση (R), τη μετακίνηση του δείκτη εισόδου προς τα αριστερά (MR) ή τα δεξιά (ML), τη φόρτωση

(L), την αποθήκευση (STO), την πρόσθεση (A), την αφαίρεση (S), την αλλαγή γραμμής ($JUMP$), την υπό προϋπόθεση αλλαγή γραμμής ($CJUMP$) και τον τερματισμό (H).

- Έναν μετρητή προγράμματος.
- Έναν καταχωρητή διεύθυνσης.
- Ένα συσσωρευτή με τον οποίο εκτελούνται πράξεις.
- Ένα μικρό αριθμό καταχωρητών ειδικού σκοπού.
- Ένα αρχείο εισόδου.
- Ένα αρχείο εξόδου.

Μία MT με τα παραπάνω χαρακτηριστικά θα μπορούσε να λειτουργήσει ως ένας πραγματικός υπολογιστής. Θα μπορούσε για παράδειγμα να εκτελέσει το πρόγραμμα :

R	11	/*	<i>Read 3 bits from the input tape</i>
		/*	<i>and put them into the accumulator.</i>
MR	11	/*	<i>Move the input pointer 3bits to</i>
		/*	<i>the right.</i>
$CJUMP$	1011	/*	<i>If the value in the accumulator</i>
		/*	<i>is 0, jump to location 1011.</i>
A	11001	/*	<i>Add to the accumulator value</i>
		/*	<i>the value at location 11001.</i>
STO	11001	/*	<i>Store the result back in location</i>
		/*	<i>11001.</i>

Θεώρημα Δ'.4.1. Προσομοίωση υπολογιστή με MT Ένας υπολογιστής άμεσης προσπέμβασης με πρόγραμμα στη μνήμη μπορεί να προσομοιωθεί από μία απλή MT. Αν ο υπολογιστής απαιτεί n βήματα, για να εκτελέσει μία λειτουργία, τότε η προσομοίωση της MT θα απαιτεί $O(n^6)$ βήματα.

Απόδειξη: Η

απόδειξη στηρίζεται σε κατάλληλη κατασκευή προσομοίωσης. Ο προσομοιωτής θα χρησιμοποιεί επτά ταινίες :

- Η Ταινία 1 θα διατηρεί τη μνήμη του υπολογιστή. Θα είναι οργανωμένη ως μία σειρά από ζεύγη (δ/νση, μνήμη), που θα διαχωρίζονται μεταξύ τους με ";". Οι διευθύνσεις και οι τιμές θα αναπαριστώνται στο δυαδικό σύστημα. Αυτό σημαίνει ότι χρειάζεται μία δυαδική κωδικοποίηση των προγραμμάτων.

- Η Ταινία 2 θα διατηρεί τον μετρητή του προγράμματος, που θα είναι η τιμή ενός δείκτη στη μνήμη, που βρίσκεται στην Ταινία 1.
- Η Ταινία 3 θα διατηρεί τον καταχωρητή διεύθυνσης.
- Η Ταινία 4 θα διατηρεί το συσσωρευτή.
- Η Ταινία 5 θα διατηρεί τον κωδικό πράξης της τρέχουσας εντολής.
- Η Ταινία 6 θα διατηρεί το αρχείο εισόδου.
- Η Ταινία 7 θα διατηρεί το αρχείο εξόδου, που αρχικά θα είναι κενό.

Όπως όλες οι ΜΤ, ο προσομοιωτής θα ξεκινήσει με την είσοδό του στην Ταινία 1, ενώ όλες οι άλλες ταινίες θα είναι κενές. Η προσομοίωση χρειάζεται δύο εισόδους, το πρόγραμμα που θα προσομοιωθεί και την είσοδο για την οποία τρέξει η προσομοίωση. Οι δύο εισοδοί κωδικοποιούνται στην Ταινία 1, όπου βρίσκονται διαχωρισμένες από τον ειδικό χαρακτήρα %.

Θα υποθέσουμε ότι το πρόγραμμα βρίσκεται αποθηκευμένο στην περιοχή της μνήμης, που αρχίζει από τη διεύθυνση 0 και άρα ο μετρητής του προγράμματος πρέπει να αρχικοποιηθεί με την τιμή 0. Παραλείπουμε μία αναλυτική περιγραφή της λειτουργίας της προσομοίωσης, εστιάζοντας στη σημαντική παρατήρηση ότι ο αριθμός των βημάτων, που απαιτούνται για την εκτέλεση των πράξεων που ορίσαμε, δίνεται στη χειρότερη περίπτωση ως μία γραμμική συνάρτηση του μήκους της Ταινίας 1.

Αν θεωρήσουμε ότι η προσομοίωση αρχίζει χρησιμοποιώντας μία περιοχή μήκους k στην Ταινία 1, κάθε εντολή μπορεί να αυξάνει τον αριθμό των διευθύνσεων μνήμης κατά ένα και αυτό γιατί μία πράξη αποθήκευσης μπορεί να εγγράψει σε διεύθυνση, που δε χρησιμοποιούνταν προηγουμένως. Επιπλέον, κάθε εντολή μπορεί να αυξάνει το μήκος μιας «λέξης» της μηχανής κατά ένα, αφού η πράξη A μπορεί να δημιουργεί τιμές, που είναι κατά ένα bit μεγαλύτερες σε μήκος από τα ορίσματα. Άρα, μετά από n βήματα προσομοίωσης το μήκος της ταινίας θα μπορούσε να είναι $k + n^2$ αν δημιουργούνται συνεχώς νέες «λέξεις» και το μήκος των «λέξεων» διαρκώς μεγαλώνει. Αν υποθέσουμε ότι $n \geq k$, τότε μπορούμε να πούμε ότι το μήκος της ταινίας μετά από n βήματα είναι $\mathcal{O}(n^2)$. Επομένως για ένα βήμα του αρχικού προγράμματος, η ΜΤ των επτά ταινιών μπορεί να χρειάζεται το πολύ $\mathcal{O}(n^2)$ βήματα, ενώ για όλα τα n βήματα του προγράμματος θα χρειάζεται συνολικά $\mathcal{O}(n^3)$ δικά της βήματα.

Η προσομοίωση, που περιγράψαμε χρησιμοποιεί επτά ταινίες. Από το Θεώρημα 3.1 γνωρίζουμε ότι μία ΜΤ πολλών ταινιών, που εκτελεί n βήματα μπορεί να προσομοιωθεί από μία απλή ΜΤ με μία ταινία, που θα εκτελεί $\mathcal{O}(n^2)$ βήματα, ανεξαρτήτως του αριθμού των ταινιών. Κατά συνέπεια, ο αριθμός των βημάτων, που χρειάζεται μία απλή ΜΤ μιας ταινίας, για να προσομοιώσει ένα πρόγραμμα από n βήματα, θα είναι $\mathcal{O}(n^6)$. Αυτό συνιστά μία αύξηση, που είναι πολυωνυμική συνάρτηση του αριθμού των βημάτων του προγράμματος, σε αντίθεση με

την εκθετική αύξηση, που χαρακτηρίζει την προσομοίωση της μη ντετερμινιστικής MT στην απόδειξη του Θεωρήματος 3.2.

Κάθε πρόγραμμα, που γράφεται σε γλώσσα προγραμματισμού υψηλού επιπέδου, μπορεί να μεταφραστεί σε κώδικα για μία απλή μηχανή τυχαίας προσπέλασης, όπως αυτή που περιγράψαμε. Αφού λοιπόν δείξαμε ότι οποιαδήποτε μηχανή αυτού του τύπου μπορεί να προσομοιωθεί από μία απλή MT, θα μπορούμε στο εξής να ορίζουμε MT χρησιμοποιώντας ψευδοκώδικα.

Δ'.5 Εναλλακτικοί ορισμοί MT

Στη συνέχεια εξετάζουμε έναν τρόπο ορισμού MT, καθώς και το ρόλο της στοίβας, που υπάρχει στα Αυτόματα Στοίβας, αλλά δεν υπάρχει στις μηχανές Turing.

Δ'.5.1 MT με ταινία άπειρου μήκους προς μία κατεύθυνση

Σε πολλά βιβλία προτείνεται ο ορισμός MT με μία ταινία, που εκτείνεται σε άπειρο μήκος μόνο ως προς τα δεξιά, ενώ σε άλλα βιβλία υιοθετείται ένας ορισμός κατά τον οποίο η ταινία εκτείνεται σε άπειρο μήκος ως προς και τις δύο κατευθύνσεις.

Θεώρημα Δ'.5.1. *Ταινία άπειρου μήκους προς μία κατεύθυνση* Οποιοσδήποτε υπολογισμός MT με ταινία άπειρου μήκους ως προς τις δύο κατευθύνσεις προσομοιώνεται από μία MT με ταινίες, που εκτείνονται σε άπειρο μήκος ως προς μία μόνο κατεύθυνση.

Απόδειξη: Έ

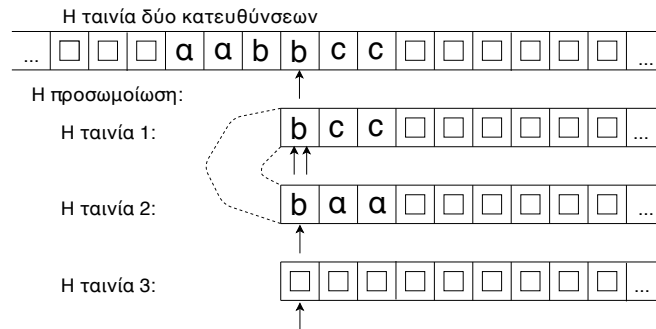
στω M μία MT με ταινία άπειρου μήκους ως προς τις δύο κατευθύνσεις. Θα περιγράψουμε μία ισοδύναμη MT M' με ταινίες άπειρου μήκους, που εκτείνεται ως προς μία μόνο κατεύθυνση. Η M' θα χρησιμοποιεί τρεις ταινίες.

Η πρώτη ταινία διατηρεί το μέρος της ταινίας της M από το τετράγωνο, που βρίσκεται η κεφαλή ανάγνωσης/εγγραφής και εκτείνεται προς τα δεξιά. Η δεύτερη ταινία της M' διατηρεί το μέρος της ταινίας της M , που εκτείνεται από τη θέση της κεφαλής ανάγνωσης/εγγραφής προς τα αριστερά. Τέλος, η τρίτη ταινία θα καταμετρά σε μοναδιαίο σύστημα με βάση το 1 τον αριθμό των κινήσεων, που έχει κάνει μέχρι στιγμής η M . Ένα παράδειγμα της συγκεκριμένης αναπαράστασης φαίνεται στο Σχήμα Δ'.13.

Η M' έχει τρεις κεφαλές ανάγνωσης/εγγραφής, μία κεφαλή (με μη διακεκομμένη γραμμή) σε κάθε ταινία της. Η M' χρησιμοποιεί τον πεπερασμένο έλεγχο καταστάσεων, για να κρατά λογαριασμό σχετικά με το αν η προσομοιωμένη κεφαλή ανάγνωσης/εγγραφής της M (διακεκομμένη γραμμή) βρίσκεται στην ταινία 1 ή στην ταινία 2. Αν η προσομοιωμένη κεφαλή ανάγνωσης/εγγραφής βρίσκεται στο τετράγωνο t της ταινίας 1, τότε η κεφαλή ανάγνωσης/εγγραφής της ταινίας 2 της M' θα βρίσκεται στο άκρο της ταινίας από αριστερά, ενώ η

κεφαλή της ταινίας 1 θα βρίσκεται στο τετράγωνο t . μία αντίστοιχη λειτουργία θα έχουμε αν η προσομοιωμένη κεφαλή ανάγνωσης/εγγραφής βρίσκεται στην περιοχή της ταινίας 2.

Αρχικά η ταινία 1 της M' θα είναι ακριβώς ίδια με την ταινία της M , η ταινία 2 θα είναι κενή και η ταινία 3 θα είναι επίσης κενή, αφού δεν έχει γίνει ακόμη καμία κίνηση από την M' . Οι κινήσεις της MT M προσομοιώνονται ως εξής. Αν η κεφαλή της M επιχειρήσει να



Σχήμα Δ'.13: Προσομοίωση MT δύο κατευθύνσεων από ταινίες μιας κατεύθυνσης

κινήθει προς το αριστερό άκρο της ταινίας, η M' θα αρχίσει να εγγράφει στο αριστερό άκρο της ταινίας 2. Αν η M συνεχίσει να κινείται προς τα αριστερά, η κεφαλή της M' στη ταινία 2 θα κινηθεί προς τα δεξιά. Αν η κεφαλή της M επιστρέψει στην αρχική της θέση κινούμενη προς τα δεξιά, τότε η κεφαλή της M' στην ταινία 1 θα αρχίσει να κινείται προς τα δεξιά. Όταν τερματίσει η M , τότε τερματίζει η προσομοίωση της από την M' .

Στην περίπτωση που η M' πρέπει να υπολογίσει μία συνάρτηση, είναι απαραίτητο να εξασφαλίσει ότι όταν θα τερματίσει περιέχει στην ταινία 1 ακριβώς τη συμβολοσειρά, που θα έπρεπε να περιέχει η ταινία της M . μέρος αυτής της συμβολοσειράς μπορεί να βρίσκεται στην ταινία 2 και γι' αυτό τα περιεχόμενα της ταινίας 1 θα πρέπει να μετακινηθούν προς τα δεξιά ώστε να δημιουργηθεί επαρκής χώρος για τα περιεχόμενα της ταινίας 2, που θα μεταφερθούν. Ο μέγιστος αριθμός συμβόλων, που μπορεί να έχει εγγράψει η M' στην ταινία 2 είναι n , όπου n είναι ο αριθμός των κινήσεων, που έχει εκτελέσει η M και βρίσκεται καταγεγραμμένος στην ταινία 3. Έτσι, η M' μετακινείται n τετράγωνα προς τα δεξιά στην ταινία 2. Στη συνέχεια μετακινείται ένα τετράγωνο κάθε φορά προς τα αριστερά για όσο διαβάσει μόνο κενά και κάθε φορά, που εκτελεί μία τέτοια κίνηση σβήνει ένα 1 από την ταινία 3. Όταν βρει το πρώτο μη κενό σύμβολο στην ταινία 2, τότε η ταινία 3 θα περιέχει την αναπαράσταση του αριθμού (σε μοναδιαίο σύστημα με βάση το 1) των φορών, που η M' θα πρέπει να μετακινήσει τα περιεχόμενα της ταινίας 1 προς τα δεξιά (κατά μία θέση), για να αντιγράψει ένα σύμβολο από την ταινία 2 στην ταινία 1. Η M' εκτελεί αυτή τη μεταφορά συμβόλων όσες φορές χρειάζεται και τότε τερματίζει.

Είναι προφανές ότι ένας οποιοσδήποτε υπολογισμός MT με ταινία άπειρου μήκους προς μία κατεύθυνση μπορεί εύκολα να προσομοιωθεί από MT με ταινία άπειρου μήκους ως προς τις δύο κατευθύνσεις και έτσι θεμελιώνεται η εκφραστική ισοδυναμία των δύο παραλλαγών

ορισμού MT.

Δ'.5.2 Ο ρόλος της στοίβας και της ταινίας

Σε σχέση με το υπολογιστικό μοντέλο του ΑΣ, οι MT δε χρησιμοποιούν στοίβα. Η άπειρη σε μήκος ταινία των MT μας έχει δώσει περισσότερες υπολογιστικές δυνατότητες από τη χρήση της στοίβας. Για να αποσαφηνίσουμε το ρόλο της στοίβας ως προς το ρόλο της ταινίας στις MT προσπαθούμε να απαντήσουμε στα εξής ερωτήματα:

- Ευνοώντας τη χρήση μιας ταινίας, που εκτείνεται σε άπειρο μήκος αντί μιας στοίβας, μήπως έχουν χαθεί κάποιες υπολογιστικές δυνατότητες, που είχαμε με τη χρήση της στοίβας;
- Θα μπορούσαμε να εξασφαλίσουμε την ισχύ, που μας δίνει η ταινία της MT, χρησιμοποιώντας μόνο στοίβες;

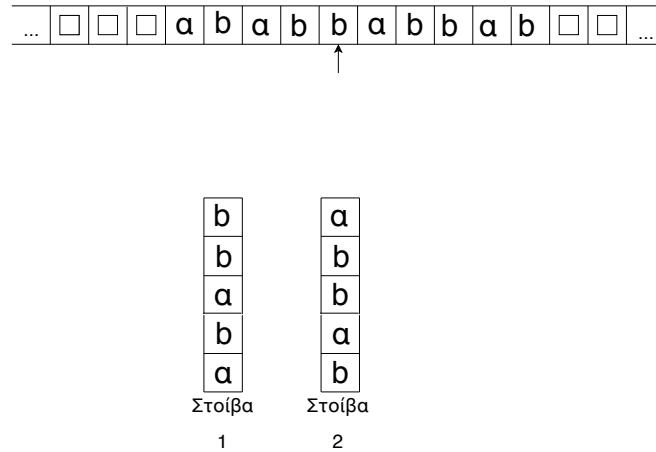
Θεώρημα Δ'.5.2. Προσομοίωση Αυτόματων Στοίβας από μηχανές Turing Η λειτουργία οποιουδήποτε ΑΣ P μπορεί να προσομοιωθεί από μία MT M .

Απόδειξη: Η

απόδειξη γίνεται με κατάλληλη κατασκευή. Δοθέντος ενός ΑΣ P κατασκευάζουμε μία (πιθανώς) μη ντετερμινιστική MT M , που προσομοιώνει τη λειτουργία του P . Εφόσον στο P έχουμε ένα πεπερασμένο αριθμό καταστάσεων, η M μπορεί να παρακολουθεί την τρέχουσα κατάσταση του P με το δικό της πεπερασμένο έλεγχο καταστάσεων.

Όπως φαίνεται στο Σχήμα, κάθε διακλάδωση εκτέλεσης της MT M , θα χρησιμοποιεί δύο ταινίες, μία για την είσοδο και μία για τη στοίβα. Η ταινία 1 θα λειτουργεί ως μία ροή εισόδου, που τροφοδοτεί το ΑΣ και θα χρησιμοποιείται αποκλειστικά για ανάγνωση. Η M θα μετακινεί την κεφαλή στην ταινία 1 μόνο προς τα δεξιά κατά ένα τετράγωνο κάθε φορά. Η ταινία 2 θα μιμείται τη συμπεριφορά της στοίβας της M , με την κεφαλή ανάγνωσης/εγγραφής να κινείται πίσω και μπρος, καθώς θα εισάγονται και θα αφαιρούνται σύμβολα από τη στοίβα. Η M θα λειτουργεί ως εξής:

1. *Αρχικοποίηση.* Γράψε στη θέση της κεφαλής ανάγνωσης/εγγραφής της ταινίας 2, που θα υποδεικνύει το κάτω όριο της στοίβας. Στην προσομοιωμένη κατάσταση S_{sim} τίθεται η τιμή s , της αρχικής κατάστασης του P .
2. *Προσομοίωση.* Έστω ς το σύμβολο στη θέση της κεφαλής ανάγνωσης/εγγραφής της ταινίας 1. Για κάθε βήμα της λειτουργίας του P εκτέλεσε:
 - (α') Αν $c = \square$, τερμάτισε με αποδοχή αν η S_{sim} είναι μία κατάσταση αποδοχής του P , ενώ διαφορετικά τερμάτισε με απόρριψη.



Σχήμα Δ'.15: Προσομοίωση της ταινίας μιας MT από δύο στοίβες

1. Αν η κεφαλή της M μετακινηθεί στο κενό μέρος από τη δεξιά πλευρά της ταινίας, τότε το σύμβολο που εγγράφεται απλά εισάγεται στην κορυφή της στοίβας 1.

Δ'.6 Γενική μηχανή Turing

Στα προβλήματα, που επεξεργαστήκαμε μέχρι τώρα, οι MT που περιγράψαμε, είχαν σχεδιαστεί για να κάνουν αποκλειστικά και μόνο μία συγκεκριμένη λειτουργία. μήπως είναι πιο επωφελής η σχεδίαση μιας MT, που «προγραμματίζεται», όπως οι ηλεκτρονικοί μας υπολογιστές; μία τέτοια MT θα δέχεται ως είσοδο ένα ζεύγος αποτελούμενο από την περιγραφή μιας οποιασδήποτε MT M και μιας συμβολοσειράς εισόδου w και θα παράγει ως αποτέλεσμα την έξοδο της μηχανής M με είσοδο w . Πράγματι, μία τέτοια κατασκευή είναι δυνατή και είναι η *Γενική μηχανή Turing* U .

Για να ορίσουμε την U χρειαζόμαστε τα εξής:

1. Να ορίσουμε μία κωδικοποίηση, που θα χρησιμοποιείται για να περιγράφεται στην U ένα ζεύγος οποιασδήποτε MT M και συμβολοσειράς w .
2. Να περιγράψουμε τη λειτουργία της U για ένα ζεύγος M και w .

Δ'.6.1 Κωδικοποίηση MT ως συμβολοσειρά

μία οποιαδήποτε MT M θα πρέπει να κωδικοποιείται ως συμβολοσειρά $\langle M \rangle$ αλφάβητου σταθερού μεγέθους, για οποιοδήποτε αριθμό καταστάσεων και αλφάβητο της M ενός οποιουδήποτε μεγέθους. Η προφανής λύση είναι να κωδικοποιούνται τα σύνολα καταστάσεων και τα αλφάβητα ως δυαδικές συμβολοσειρές.

Ας ξεκινήσουμε με την κωδικοποίηση του χώρου καταστάσεων K . Ανάλογα με το μέγεθος του K καθορίζουμε τον αριθμό i των δυαδικών ψηφίων, που απαιτούνται για την αρίθμηση

των καταστάσεων από 0 μέχρι $|K| - 1$ και εκχωρούμε σε κάθε κατάσταση τη δυαδική συμβολοσειρά μήκους i , που αντιστοιχεί στην αρίθμηση της. Κατά σύμβαση, η αρχική κατάσταση είναι η 0. Οι άλλες καταστάσεις μπορεί να αριθμούνται με οποιαδήποτε σειρά. Έστω t' η δυαδική συμβολοσειρά για την κατάσταση t . Υιοθετούμε τις εξής συμβάσεις:

- Αν t είναι η κατάσταση τερματισμού y , τότε αυτή κωδικοποιείται με τη συμβολοσειρά yt' .
- Αν t είναι η κατάσταση τερματισμού n , τότε αυτή κωδικοποιείται με τη συμβολοσειρά nt' .
- Αν t είναι οποιαδήποτε άλλη κατάσταση, τότε αυτή κωδικοποιείται με τη συμβολοσειρά qt' .

Στη συνέχεια κωδικοποιούμε το αλφάβητο της ταινίας. Θα ξεκινήσουμε αποφασίζοντας τον αριθμό j των δυαδικών ψηφίων, που απαιτούνται για την κωδικοποίηση των αριθμών από 0 μέχρι $|\Gamma| - 1$, όπου $|\Gamma|$ το μέγεθος του αλφάβητου. Σε κάθε σύμβολο εκχωρείται η δυαδική συμβολοσειρά μήκους j , που αντιστοιχεί στην αρίθμηση του. Τέλος, αν c' είναι η δυαδική συμβολοσειρά, που αντιστοιχεί στο σύμβολο c , τότε το σύμβολο κωδικοποιείται με τη συμβολοσειρά ac' .

Περιγράφουμε τώρα την κωδικοποίηση της συνάρτησης δ , που ορίζεται από πεντάδες της μορφής (κατάσταση, σύμβ. εισόδου, κατάσταση, σύμβ. εξόδου, κίνηση). Οι καταστάσεις και τα σύμβολα κωδικοποιούνται σύμφωνα με όσα συζητήθηκαν, οπότε απομένει η κωδικοποίηση των κινήσεων $\rightarrow$ και $\leftarrow$ της κεφαλής, που γίνεται απλά με τη χρήση του ίδιου συμβόλου. Κάθε μετάβαση κωδικοποιείται με μία συμβολοσειρά, που περιέχει όλες τις κωδικοποιήσεις των καταστάσεων, των συμβόλων και της κίνησης της κεφαλής, ενώ οι διαφορετικές μεταβάσεις της δ διαχωρίζονται μεταξύ τους με κόμμα.

Παράδειγμα Δ'.6.1. Κωδικοποίηση ενός πλήρους ορισμού MT. Ας θεωρήσουμε MT M , που ορίζεται ως εξής:

$$M = (\{q_0, q_1, h\}, \{a, b, c\}, \{\square, a, b, c\}, \delta, q_0, \{h\})$$

όπου

$$\begin{aligned} \delta = & ((q_0, \square), (q_1, \square, \rightarrow)), \\ & ((q_0, a), (q_0, b, \rightarrow)), \\ & ((q_0, b), (q_1, a, \leftarrow)), \\ & ((q_0, c), (q_1, b, \leftarrow)), \\ & ((q_1, \square), (q_0, a, \rightarrow)), \\ & ((q_1, a), (q_1, b, \rightarrow)), \\ & ((q_1, b), (q_1, b, \rightarrow)), \\ & ((q_1, c), (h, a, \leftarrow)) \end{aligned}$$

Ξεκινούμε κωδικοποιώντας πρώτα την κάθε κατάσταση και το καθένα από τα σύμβολα της ταινίας:

κατάσταση/σύμβολο	κωδικοποίηση
q_0	$q00$
q_1	$q01$
h	$q10$
$\square$	$a00$
a	$a01$
b	$a10$
c	$a11$

Η πλήρης κωδικοποίηση της $MT M$ είναι η συμβολοσειρά $\langle M \rangle$, ως εξής:

$$\begin{aligned} & (q00, a00, q01, a00, \rightarrow), (q00, a01, q00, a10, \rightarrow), (q00, a10, q01, a01, \leftarrow), \\ & (q00, a11, q01, a10, \leftarrow), (q01, a00, q00, a01, \rightarrow), (q01, a01, q01, a10, \rightarrow), \\ & (q01, a10, q01, a10, \leftarrow), (q01, a11, q10, a01, \leftarrow) \end{aligned}$$

Η Γενική MT θα δέχεται στην είσοδο μία $MT M$ και μία συμβολοσειρά w και θα προσομοιώνει την εκτέλεση της M με είσοδο w . Καταλαβαίνουμε λοιπόν ότι χρειάζεται η κωδικοποίηση δύο ή ενδεχομένως και περισσότερων παραμέτρων ως μία μόνο συμβολοσειρά. Αυτό γίνεται εύκολα ως εξής: κωδικοποιούμε πρώτα κάθε παράμετρο ξεχωριστά και στη συνέχεια παίρνουμε την παράθεση όλων των παραμέτρων χρησιμοποιώντας ένα σύμβολο διαχωρισμού τους, που δεν ανήκει στο αλφάβητο κάποιας συμβολοσειράς. Για παράδειγμα, θα μπορούσαμε να κωδικοποιήσουμε το ζεύγος $\langle M \rangle, \langle abba \rangle$ ως $\langle M \rangle; \langle abba \rangle$. Θα χρησιμοποιούμε το συμβολισμό $\langle w_1, w_2, \dots, w_n \rangle$, για να αναπαραστήσουμε μία μόνο συμβολοσειρά, που θα κωδικοποιεί την ακολουθία των συμβολοσειρών $w_1, w_2, \dots, w_n$.

Έχοντας υιοθετήσει μία καλά ορισμένη κωδικοποίηση $\langle M \rangle$ για οποιαδήποτε $MT M$ μπορούμε επίσης να περάσουμε την $\langle M \rangle$ ως είσοδο σε προγράμματα διαφορετικά από τη

γενική MT U , που θα μπορούν να εφαρμόσουν μία λειτουργία στην M . Έτσι, θα μπορούμε να θεωρήσουμε ότι ορίζεται μία MT T , που δέχεται ως είσοδο την περιγραφή μιας άλλης MT, έστω της M_1 και τη μετασχηματίζει στην περιγραφή μιας διαφορετικής MT, έστω της M_2 .

Δ'.6.2 Η Γενική μηχανή Turing

Σύμφωνα με το εισαγωγικό σημείωμα αυτής της ενότητας, η MT U που θα ορίσουμε είναι η Γενική MT με την έννοια ότι δοθείσης μιας οποιασδήποτε MT M και μιας εισόδου w , η U θα προσομοιώσει τη λειτουργία της M για είσοδο w . Πιο συγκεκριμένα, για είσοδο $\langle M, w \rangle$ η U θα πρέπει:

- Να τερματίζει, αν και μόνο αν η M τερματίζει με είσοδο w .
- Αν η M είναι μία μηχανή απόφασης ή ημιαπόφασης, τότε:
 - Αν η M αποδέχεται, να αποδέχεται και η U την είσοδο.
 - Αν η M απορρίπτει, να απορρίπτει και η U την είσοδο.
- Αν η M υπολογίζει μία συνάρτηση, τότε η $U(\langle M, w \rangle)$ να είναι ίση με $M(w)$.

Για να μπορεί η U να προσομοιώσει τη λειτουργία της M για είσοδο w θα χρησιμοποιεί τρεις ταινίες:

- Η ταινία 1 θα αντιστοιχεί στην ταινία της M .
- Η ταινία 2 θα περιέχει την $\langle M \rangle$, δηλαδή το «πρόγραμμα» που πρόκειται να εκτελέσει η U .
- Η ταινία 3 θα περιέχει την κωδικοποίηση της κατάστασης στην οποία βρίσκεται η M σε μία οποιαδήποτε στιγμή της προσομοίωσης. μπορούμε να θεωρήσουμε ότι η ταινία 3 δίνει την τιμή του «μετρητή προγράμματος».

Αρχικά η U θα έχει στην ταινία 1 τη συμβολοσειρά $\langle M, w \rangle$, με τις άλλες δύο ταινίες κενές και θα πρέπει να αρχικοποιήσει τις ταινίες της:

1. μεταφέροντας την $\langle M \rangle$ από την ταινία 1 στην ταινία 2 και σβήνοντας την στη συνέχεια από την ταινία 1.
2. Εξετάζοντας την $\langle M \rangle$, για να ανιχνεύσει τον αριθμό των καταστάσεων της M και για να καθορίσει το i , δηλαδή τον αριθμό των δυαδικών ψηφίων, που απαιτεί η κωδικοποίηση των καταστάσεων της M . Στην ταινία 3 γράφεται η $q0^i$, που αντιστοιχεί στην αρχική κατάσταση της M .

Η U λοιπόν ξεκινάει την προσομοίωση της M με τις τρεις κεφαλές ανάγνωσης/εγγραφής στις τρεις ταινίες, όπως περιγράψαμε προηγουμένως. Γενικά, κάθε βήμα της προσομοίωσης θα αρχίζει με τις κεφαλές τοποθετημένες ως εξής:

- Η κεφαλή ανάγνωσης/εγγραφής της ταινίας 1 θα βρίσκεται στο χαρακτήρα a , που είναι ο πρώτος χαρακτήρας της κωδικοποίησης του τρέχοντος συμβόλου στην ταινία της M .
- Η κεφαλή ανάγνωσης/εγγραφής της ταινίας 2 θα βρίσκεται στην αρχή της $\langle M \rangle$.
- Η κεφαλή ανάγνωσης/εγγραφής της ταινίας 3 θα βρίσκεται στο χαρακτήρα q του μετρητή προγράμματος.

μετά λοιπόν από την αρχικοποίηση της, όπως παραπάνω, η U λειτουργεί ως εξής:

1. μέχρι να τερματίσει η M :

- (α') Σάρωσε την ταινία 2 για μια πεντάδα, που ταιριάζει με το ζεύγος της τρέχουσας κατάστασης και συμβόλου εισόδου.
- (β') Εκτέλεσε την αντίστοιχη ενέργεια μεταβάλλοντας τις ταινίες 1 και 3. Αν είναι απαραίτητο, επέκτεινε την προσομοιωμένη ταινία, που βρίσκεται κωδικοποιημένη στην ταινία 1.
- (γ') Αν δε βρεθεί πεντάδα που ταιριάζει, τότε τερμάτισε.

2. Ανάφερε το ίδιο αποτέλεσμα, που θα έδινε η M :

- Αν η M είναι μία μηχανή απόφασης ή ημιαπόφασης και η προσομοιωμένη κατάσταση της M είναι y , τότε έχουμε αποδοχή. Αν η προσομοιωμένη κατάσταση είναι n , τότε έχουμε απόρριψη.
- Αν η M είναι μία μηχανή που υπολογίζει μία συνάρτηση, τότε αναδιαμόρφωσε την ταινία, έτσι ώστε να απομένει τελικά μόνο η τιμή της ταινίας 1.

Αν η MT M τερματίζει σε k βήματα, τότε η U θα εκτελέσει τον προαναφερόμενο βρόγχο k φορές και κάθε φορά, που εκτελείται ο βρόγχος θα πρέπει να σαρωθεί η $\langle M \rangle$, για να προσδιοριστεί η επόμενη ενέργεια. Έτσι, η U θα εκτελέσει συνολικά $O(|M| \cdot k)$ βήματα.

μετά από την περιγραφή της U είναι πλέον σαφές ότι οι MT είναι ένα γενικό μοντέλο υπολογισμού: αντί για την κατασκευή πραγματικών MT μπορούμε να κατασκευάσουμε μία μηχανή, που θα δέχεται ως είσοδο την περιγραφή οποιασδήποτε άλλης MT επιθυμούμε αυτή να εκτελέσει.

Θεώρημα Δ'.6.1. Προσομοίωση MT από άλλη MT. Δοθείσας μιας MT M και μιας συμβολοσειράς w , υπάρχει μία MT M' , που προσομοιώνει την εκτέλεση της M για είσοδο w , έτσι ώστε:

- να τερματίζει αν και μόνο αν η M τερματίζει για είσοδο w και
- αν τερματίζει, να επιστρέφει ακριβώς το ίδιο αποτέλεσμα, που επιστρέφει η M .

Απόδειξη: Δ

οθείσας μιας συγκεκριμένης MT M και μιας συμβολοσειράς w ορίζουμε τη MT M' , η οποία θα λειτουργεί ως εξής:

$$M'(x) = \text{Κλήση της Γενικής MTU με είσοδο } \langle M, w \rangle \quad (\Delta.10)$$

Η περιγραφή της M' ορίζει μία σταθερή συνάρτηση, αφού η M' αγνοεί την είσοδό της x . Η M' τερματίζει αν και μόνο αν τερματίζει η U και όταν αυτό συμβαίνει θα επιστρέφει το αποτέλεσμα της εκτέλεσης της M με είσοδο w .

Κεφάλαιο Ε'

Υπολογίσιμες Συναρτήσεις και μηχανές Turing

Στο κεφάλαιο αυτό έχοντας αναπτύξει τη θεωρία γύρω από τις μηχανές Turing θα δώσουμε έμφαση στην ισοδυναμία των μηχανών Turing και των μερικών αναδρομικών συναρτήσεων.

Σύμφωνα με τη θέση των Church-Turing είναι το πιο ισχυρό υπολογιστικό μοντέλο, που μπορεί να υπάρξει. Το γεγονός αυτό δεν οδηγεί σε επίλυση οποιουδήποτε προβλήματος, το οποίο να μπορεί να περιγραφεί σε τυπική γλώσσα, από κατάλληλη μηχανή Turing. Υπάρχουν καλώς ορισμένα προβλήματα, για τα οποία δεν είναι δυνατό να κατασκευαστεί MT, που να τα επιλύει.

Ε'.1 Το Πρόβλημα του Τερματισμού

Θεώρημα Ε'.1.1. *Πρόβλημα Τερματισμού - ημιαπόφαση. Η γλώσσα του Προβλήματος Τερματισμού:*

$$H = \{ \langle M, w \rangle : \text{η } MTM \text{ τερματίζει για την είσοδο } w \}$$

Απόδειξη: Η

απόδειξη στηρίζεται σε κατάλληλη κατασκευή μιας MT M_{SH} ημιαπόφασης. Το μόνο, που χρειάζεται να κάνει η μηχανή αυτή είναι η εκτέλεση της M με είσοδο w και εφόσον η M τερματίσει, τότε η M_{SH} αποδέχεται η είσοδο. Η περιγραφείσα μηχανή M_{SH} αποδέχεται την είσοδο, αν και μόνο αν η M τερματίζει όταν δέχεται είσοδο w . Επομένως η M_{SH} ημιαποφασίζει την H .

Θεώρημα Ε'.1.2. *Πρόβλημα Τερματισμού - μη αποφασίσιμο. Η γλώσσα του Προβλήματος Τερματισμού:*

$$H = \{ \langle M, w \rangle : \text{η } MTM \text{ τερματίζει για την είσοδο } w \}$$

δεν είναι αποφασίσιμη.

Απόδειξη: Η

απόδειξη γίνεται με εις άτοπο απαγωγή. Αν η H ήταν αποφασίσιμη, τότε θα υπήρχε μία MT M_H , τέτοια ώστε :

$$M_H (< M, w >) = Av < M >$$

είναι η κωδικοποίηση μιας MT , που τερματίζει για είσοδο w , τότε είναι αποδεκτή, αλλιώς απορρίπτεται.

Ορίζουμε τώρα μια άλλη MT M_{Tr} ως εξής :

$$M_{Tr} (x \in \Sigma^*) = Av \eta M_H$$

αποδέχεται την

$< x, x >$, τότε εκτελείται ένας επαναληπτικός

βρόχος για πάντα, αλλιώς τερματίζει.

Αν υπάρχει MT M_H , τότε θα υπάρχει επίσης και η MT M_{Tr} . Όμως σε αυτήν την περίπτωση, πώς θα συμπεριφέρεται η M_{Tr} για την είσοδο $< M_{Tr} >$;

Τότε θα εκτελέσει την $M_H (< M_{Tr}, M_{Tr} >)$ και αν αυτή τερματίσει με αποδοχή, τότε η M_{Tr} θα εκτελέσει έναν επαναληπτικό βρόχο για πάντα. Από την άλλη, αν η M_H απορρίψει την είσοδο, τότε η M_{Tr} θα τερματίσει.

Επομένως βρέθηκε μια είσοδος για την οποία η M_H αποτυγχάνει να δώσει το σωστό αποτέλεσμα σχετικά με τη συμπεριφορά της M_{Tr} , ανεξάρτητα από το πώς υλοποιείται η M_H .

Συμπεραίνουμε λοιπόν ότι δεν υπάρχει ορθή υλοποίηση για την M_H , κάτι που είναι άτοπο ως προς την υπόθεση που κάναμε αρχικά ότι υπάρχει M_H . Επομένως η γλώσσα H δεν είναι αποφασίσιμη.

Ε'.2 Υπολογίσιμες Συναρτήσεις

Ορισμός Ε'.2.1. Έστω $f : X \rightarrow Y$ μία οποιαδήποτε συνάρτηση.

- Η f είναι μια πλήρης συνάρτηση στο πεδίο X , αν και μόνο αν η συνάρτηση ορίζεται για όλα τα στοιχεία του X .
- Η f είναι μια μερική συνάρτηση στο πεδίο X , αν και μόνο αν η συνάρτηση ορίζεται για μηδέν ή περισσότερα στοιχεία του X .

Παράδειγμα Ε'.2.1. Ας θεωρήσουμε τη συνάρτηση *steps*, που ορίζεται σε ένα πεδίο αποτελούμενο με στοιχεία της μορφής $< M, w >$ και επιστρέφει τον αριθμό των βημάτων, που απαιτούνται για να τερματίσει η MT M όταν εκτελείται με είσοδο w . Η *steps* ορίζεται ως μερική συνάρτηση στο πεδίο $\{< M, w >\}$, αφού δεν ορίζεται για τις τιμές $< M, w >$ με τις περιπτώσεις

εκείνες, που η M δεν τερματίζει με είσοδο w . Αν όμως ορίζαμε τη *steps* σε ένα πιο περιορισμένο πεδίο, δηλαδή στο

$$\{ \langle M, w \rangle : \text{η } MTM \text{ τερματίζει με είσοδο } w \}$$

τότε θα ήταν πλήρης.

Η διαφορά από την έννοια της συνάρτησης στα μαθηματικά είναι ότι στη Θεωρία Υπολογισμού ενδιαφερόμαστε να γράφουμε προγράμματα, δηλαδή MT , που να μπορούν να υπολογίζουν συναρτήσεις. Έτσι θα μπορούσαμε να κάνουμε τη σύνδεση τονίζοντας ότι το σύνολο τιμών στο οποίο αναγκαστικά ορίζεται η *steps* είναι η γλώσσα του προβλήματος τερματισμού H η οποία αποδείξαμε ότι δεν είναι αποφασίσιμη γλώσσα. Άρα ασχέτως με τη MT , που θα σχεδιάσουμε για τον υπολογισμό της *steps*, δεν υπάρχει MT , που θα εξετάζει μια τιμή και θα μπορεί να αποφασίσει αν θα μπορούσε να εκτελεστεί η MT για τον υπολογισμό της *steps*. Έτσι λοιπόν η *steps* θα είναι μερική και όχι πλήρης συνάρτηση στο πεδίο εκείνο, στο οποίο εκτελείται το πρόγραμμα που την υλοποιεί. Αυτό σημαίνει ότι ένα οποιοδήποτε τέτοιο πρόγραμμα θα αποτυγχάνει να τερματίσει για κάποιες εισόδους!

Έστω M μια MT με αρχική κατάσταση s , τελική κατάσταση h , αλφάβητο εισόδου Σ και αλφάβητο ταινίας Γ . Έστω η αρχική διαμόρφωση της M είναι $(s, \sqcup w)$, όπου $w \in \Sigma^*$. Ορίζουμε τη συνάρτηση $M(w) = z$, αν και μόνο αν $(s, \sqcup w) \vdash_M^* (h, \sqcup z)$, δηλαδή όταν η M ξεκινάει με είσοδο $w \in \Sigma^*$ τερματίζει αφήνοντας τη z στην ταινία με την κεφαλή ανάγνωσης/εγγραφής στη θέση ακριβώς αριστερά της z .

Έχουμε διευκρινίσει ότι όταν η f ορίζεται μόνο σε ένα υποσύνολο του Σ^* υπάρχει MT M , που τερματίζει με τη σωστή τιμή για όλες τις εισόδους, που η f ορίζεται, τότε λέμε την f *μερικώς υπολογίσιμη*. Για όλες τις άλλες εισόδους η M αποτυγχάνει να τερματίσει.

Έστω f μια μερικώς υπολογίσιμη συνάρτηση με τιμές, που ορίζονται μόνο για ένα γνήσιο υποσύνολο του Σ^* . μια οποιαδήποτε MT , που υπολογίζει την f θα αποτυγχάνει σε κάποιες εισόδους. Ας θεωρήσουμε τώρα μόνο τις συναρτήσεις f , που το πεδίο X στο οποίο ορίζονται είναι αποφασίσιμο, δηλαδή τις συναρτήσεις, που είναι πλήρεις στο αποφασίσιμο σύνολο X . Για μια τέτοια f , ορίζουμε τη νέα συνάρτηση f' , ίδια με την f , εκτός από το πεδίο τιμών που θα περιλαμβάνει μια επιπλέον τιμή την *Error*. Για οποιαδήποτε είσοδο z , που συμβαίνει η f να μην ορίζεται, έχουμε $f' = \text{Error}$. Άρα για μια MT , που υπολογίζει την f μπορούμε να κατασκευάσουμε μια νέα MT M' , που υπολογίζει την f' και τερματίζει πάντα. Αν η f ορίζεται στο σύνολο X , εφόσον το ίδιο δεν είναι αποφασίσιμη γλώσσα, θα υπάρχει μια MT M_X , που θα το αποφασίζει.

Ορισμός Ε'.2.2. Έστω μια συνάρτηση f , που ορίζεται σε ένα σύνολο $X \subset \Sigma^*$, που αποφασίζεται από τη MT M_X . Η f είναι υπολογίσιμη, αν και μόνο αν υπάρχει MT M' , που τερματίζει

για όλες τις εισόδους υπολογίζοντας μια συνάρτηση f' ως εξής:

$M'(x) = 1$. Εκτέλεσε την M_X με είσοδο x .

2. Αν απορρίπτεται, τότε *Error*.

3. Αν είναι αποδεκτή, τότε για μια κατάληψη $MTM(x) = f(x)$.

Ισοδύναμα μια συνάρτηση f είναι υπολογίσιμη, αν και μόνο αν είναι μερικώς υπολογίσιμη και το πεδίο ορισμού της είναι ένα αποφασίσιμο σύνολο.

Ο ορισμός μπορεί να επεκταθεί κατάλληλα και για τις συναρτήσεις f , που τα πεδία ορισμού και τιμών δεν είναι σύνολα συμβολοσειρών, όπως η συνάρτηση *succ*. Σε αυτήν την περίπτωση η συνάρτηση f είναι υπολογίσιμη, αν και μόνο αν ισχύουν οι παρακάτω συνθήκες:

- Υπάρχουν αλφάβητα Σ και Σ^* , με μια κωδικοποίηση των στοιχείων του πεδίου ορισμού της f ως συμβολοσειρές του Σ^* και μια κωδικοποίηση των στοιχείων του πεδίου τιμών ως συμβολοσειρές του Σ'^* .
- Υπάρχει μια υπολογίσιμη συνάρτηση f' , τέτοια ώστε για κάθε $w \in \Sigma'^*$:
 - Αν $w = \langle x \rangle$ και x είναι ένα στοιχείο του πεδίου ορισμού της f τότε $f'(w) = \langle f(x) \rangle$
 - Αν η w δεν είναι κωδικοποίηση κάποιου στοιχείου του πεδίου ορισμού της f , τότε $f'(w) = \text{Error}$

Παράδειγμα Ε'.2.2. μερικώς Υπολογίσιμη Συνάρτηση. Η συνάρτηση *steps* είναι πλήρης στο σύνολο $\{ \langle M, w \rangle : \text{η } MTM \text{ τερματίζει με είσοδο } w \}$, αλλιώς είναι μερική συνάρτηση στο σύνολο $\{ \langle M, w \rangle \}$. Η *steps* διαπιστώνουμε πως είναι μερικώς υπολογίσιμη συνάρτηση αφού υπολογίζεται από την παρακάτω *MT* τριών ταινιών:

$$M_{steps}(x) =$$

1. Αν η x δεν είναι συντακτικά μια καλώς ορισμένη συμβολοσειρά $\langle M, w \rangle$, τότε επανέλαβε.
2. Αν η x είναι συντακτικά καλώς ορισμένη συμβολοσειρά $\langle M, w \rangle$, τότε:
 - (α) Αντέγραψε την M στην Ταινία 3.
 - (β) Αντέγραψε την w στην Ταινία 2.
 - (γ) Γράψε 0 στην Ταινία 1.
 - (δ) Προσομοίωσε την M για είσοδο w στην Ταινία 2, διατηρώντας ένα μετρητή των βημάτων της M στην Ταινία 1.

Η M_{steps} τερματίζει, όταν η είσοδός της είναι καλώς ορισμένη και συμβαίνει η M να τερματίζει για είσοδο w . Αν η M_{steps} τερματίσει, τότε βρίσκει την τιμή $steps (< M, w >)$ στην Ταινία 1. Όμως από Θεώρημα υπάρχει μια $MT M'_{steps}$ μιας ταινίας, που η έξοδός της είναι ίδια με την τιμή της M_{steps} , που τελικά καταγράφεται στην Ταινία 1. Επομένως η M'_{steps} θα είναι μια απλή MT , που υπολογίζει τη $steps$ και αυτό αποδεικνύει ότι η $steps$ είναι μερικώς υπολογίσιμη. Όμως η $steps$ δεν είναι υπολογίσιμη. Πιο συγκεκριμένα, αποδεικνύεται ότι δεν υπάρχει MT , που υπολογίζει τη συνάρτηση $steps'$, η οποία ορίζεται ως εξής:

$steps'(x) = \text{Αν } x \text{ δεν είναι συντακτικά καλώς ορισμένη } (< M, w >) \text{ τότε Error}$

$\text{Αν } x \text{ είναι συντακτικά καλώς ορισμένη αλλιώς } η steps (< M, w >) \text{ δεν ορίζεται}$

$(\text{δηλαδή } η μ \text{ δεν τερματίζει}), \text{ τότε Error}$

$\text{Αν } η steps (< M, w >) \text{ ορίζεται (δηλαδή } η μ \text{ τερματίζει), τότε } steps (< M, w >).$

Αποδεικνύουμε ότι δεν υπάρχει τέτοια MT με αναγωγή από το H . Υποθέτουμε ότι υπάρχει και την ονομάζουμε $M_{steps'}$. Τότε θα μπορούσαμε να ορίσουμε μια $MT D_H$, που θα αποφασίζει τη γλώσσα $H = \{ < M, w > : η MT M \text{ τερματίζει για την είσοδο } w \}$:

$D_H (< M, w >) = \text{Εκτέλεσε την } M_{steps'} (< M, w >).$

$\text{Αν το αποτέλεσμα είναι Error τότε απορρίπτεται διαφορετικά γίνεται}$
 δεκτή

Γνωρίζουμε όμως ότι δεν υπάρχει MT , που να αποφασίζει την H και άρα καταλήγουμε σε άτοπο. Επομένως δε μπορεί να υπάρχει η $MT M_{steps'}$ και επομένως η $steps$ δεν είναι υπολογίσιμη.

Θεώρημα Ε'.2.1. μη Υπολογίσιμες Συναρτήσεις. Υπάρχουν συναρτήσεις, που δεν είναι μερικώς υπολογίσιμες.

Έχουμε μια αντιστοιχία μεταξύ τριών διαφορετικών κλάσεων γλωσσών, που είναι οι $\mathcal{D}$, $\mathcal{SD}/\mathcal{D}$ και $\neg\mathcal{SD}$, με τρεις διακριτές κλάσεις υπολογισιμότητας συναρτήσεων, τις υπολογίσιμες, τις μερικώς υπολογίσιμες και τις μη υπολογίσιμες συναρτήσεις.

Παράδειγμα Ε'.2.3. Γλώσσες και υπολογισιμότητα συναρτήσεων. Εξετάζουμε τρία ενδεικτικά προβλήματα και τα διατυπώνουμε με περιγραφή γλώσσας και συνάρτησης. Καθένα από τα προβλήματα αντιστοιχεί σε γλώσσα/συνάρτηση διαφορετικής κλάσης.

1. Πρόβλημα: Δοθέντων τριών φυσικών αριθμών x, y, z ισχύει $z = x \cdot y$;

Γλώσσα στο $\mathcal{D}$ (αποφασίσιμη):

$$\{ \langle x \rangle \star \langle y \rangle = \langle z \rangle : x, y, z \in \{0, \dots, 9\}^+ \text{ τέτοιες } x \cdot y = z \}$$

Συνάρτηση υπολογίσιμη:

$$f : \mathbb{N} \times \mathbb{N} \rightarrow \mathbb{N}, f(x, y) = x \cdot y$$

2. Πρόβλημα: Δοθείσης μιας $MT\ M$ και μιας συμβοληοσειράς w , τερματίζει η M σε n βήματα ;

Γλώσσα στο $SD/\mathcal{D}$ (ημιαποφασίσιμη αλλιώς όχι αποφασίσιμη):

$$\{ \langle M, w, n \rangle : \text{η } MT\ M \text{ τερματίζει για την } w \text{ σε } n \text{ βήματα} \}$$

Συνάρτηση μερικώς υπογίσιμη:

$$f : \{ \langle M, w \rangle \} \rightarrow \mathcal{N}, f(\langle M, w \rangle) = \begin{array}{l} \text{αν η } MT\ M \text{ τερματίζει για την} \\ \text{ω τότε ο αριθμός βημάτων της} \\ \text{M πριν τερματίσει αλλιώς} \\ \text{δεν ορίζεται} \end{array}$$

3. Πρόβλημα: Δοθείσης μιας $MT\ M$ τερματίζει η M για όλες τις συμβοληοσειρές σε όχι περισσότερα από n βήματα ;

Γλώσσα στο $\neg SD$ (όχι ημιαποφασίσιμη):

$$\{ \langle M, n \rangle : \text{η } MT\ M \text{ τερματίζει για κάθε στοιχείο του } \Sigma^* \\ \text{σε όχι περισσότερα από } n \text{ βήματα} \}$$

Συνάρτηση όχι (μερικώς) υπογίσιμη:

$$f : \{ \langle M, w \rangle \} \rightarrow \mathcal{N}, f(\langle M, w \rangle) = \begin{array}{l} \text{αν η } M \text{ τερματίζει για όλες} \\ \text{της συμβοληοσειρές, τότε} \\ \text{ο μέγιστος αριθμός βημάτων της} \\ \text{M πριν τερματίσει αλλιώς} \\ \text{δεν ορίζεται} \end{array}$$

Ε'.3 Θεωρία Αναδρομικών Συναρτήσεων

Οι οικογένειες γλωσσών και συναρτήσεων αναφέρονται αλλιώς ως εξής:

- αναδρομικές γλώσσες αντι για αποφασίσιμες
- αναδρομικά απαριθμήσιμες γλώσσες αντί για ημιαποφασίσιμες
- μερικώς αναδρομικές συναρτήσεις αντί για μερικώς υπολογίσιμες
- αναδρομικές συναρτήσεις αντί για υπολογίσιμες

Στην προσπάθεια να είμαστε συνεπείς τονίζουμε ότι αναδρομική συνάρτηση είναι μια συνάρτηση που υπολογίζεται απο μια MT που τερματίζει για όλες τις εισόδους της. μερικώς αναδρομική συνάρτηση είναι μια συνάρτηση που μπορεί να υπολογίζεται από κάποια MT αλλά μπορεί να πέφτει σε αέναη επανάληψη, αν υπάρχουν εισοδοι για τις οποίες η συνάρτηση δεν ορίζεται.

Στο σημείο αυτό αναδύεται η ερώτηση: μπορούμε να ορίσουμε τις ίδιες κλάσεις συναρτήσεων ανεξάρτητα από ένα μοντέλο υπολογισμού, αλλά στηριζόμενοι σε καθιερωμένα μαθηματικά εργαλεία; Θα μπορούσαμε δηλαδή να έχουμε τον ορισμό ενός μικρού συνόλου αρχέγονων συναρτήσεων, μαζί με την δυνατότητα να δημιουργούμε νέες χρησιμοποιώντας τις πράξεις της σύνθεσης και της αναδρομής; Η απάντηση θα φανεί αμέσως και είναι θετική.

Το πεδίο ορισμού της κάθε συνάρτησης είναι μια διατεταγμένη n -άδα φυσικών αριθμών και το πεδίο τιμών είναι το σύνολο των φυσικών αριθμών. Καθώς οι αριθμοί μπορούν να αναπαρασταθούν από συμβολοσειρές και οι συμβολοσειρές μπορούν να αναπαρασταθούν από αριθμούς, δεν υπάρχει κάποια θεμελιώδης ασυμβατότητα μεταξύ της θεωρίας που θα αναφερθούμε και της θεωρίας των ΜΤ, που επεξεργάζονται συμβολοσειρές.

Ε'.3.1 Στοιχειώδης Αναδρομικές Συναρτήσεις

Ορισμός Ε'.3.1. Ορίζουμε ως αρχέγονες ή αρχικές τις παρακάτω συναρτήσεις από το $\mathcal{N} \times \mathcal{N} \dots \times \mathcal{N}$ στο $\mathcal{N}$:

- τις σταθερές συναρτήσεις:

$$\text{για } k \geq 1 \text{ και } a \geq 0, C_a^k(X) = a \text{ για κάθε } X \in \mathcal{N}^k$$

- τη συνάρτηση $\text{succ} : \mathcal{N} \rightarrow \mathcal{N}$:

$$\text{succ}(n) = n + 1$$

- την οικογένεια των συναρτήσεων προβολής:

$$\text{για } 0 < j \leq k, p_j^k(x_1, x_2, \dots, x_k) = x_j$$

Ορισμός Ε'.3.2. Ορίζουμε τις παρακάτω πράξεις μεταξύ συναρτήσεων:

- σύνθεση της $f : \mathcal{N}^k \rightarrow \mathcal{N}$ με τις $g_1, g_2, \dots, g_k : \mathcal{N}^m \rightarrow \mathcal{N}$

$$h(X) = f(g_1(X), \dots, g_k(X)) \text{ για κάθε } X \in \mathcal{N}^m$$

- στοιχειώδης αναδρομική της $f : \mathcal{N}^{n+1} \rightarrow \mathcal{N}$ ως προς τις $g : \mathcal{N}^n \rightarrow \mathcal{N}$ και $h : \mathcal{N}^{n+2} \rightarrow \mathcal{N}$, για $n \geq 0$ που ορίζεται από τους τύπους:

$$- f(X, 0) = g(X)$$

$$- f(X, k+1) = h(X, k, f(X, k))$$

για κάθε $X \in \mathcal{N}^n$ και κάθε $k \geq 0$.

Ορισμός Ε'.3.3. Το σύνολο των στοιχειωδών αναδρομικών συναρτήσεων ορίζεται ως εξής:

1. Όλες οι αρχέγονες συναρτήσεις είναι στοιχεία του.

2. Για κάθε $k \geq 0$ και $m \geq 0$, αν $f : \mathcal{N}^k \rightarrow \mathcal{N}$ και $g_1, \dots, g_k : \mathcal{N}^m \rightarrow \mathcal{N}$ είναι στοιχεία του συνόλου, τότε η συνάρτηση $f(g_1, g_2, \dots, g_k)$, που λαμβάνεται από τη σύνθεση της f με τις $g_1, g_2, \dots, g_k$ είναι επίσης ένα στοιχείο του συνόλου.
3. Για κάθε $n \geq 0$ και κάθε $g : \mathcal{N}^n \rightarrow \mathcal{N}$ και $h : \mathcal{N}^{n+2} \rightarrow \mathcal{N}$ του συνόλου των στοιχειωδών αναδρομικών συναρτήσεων, η συνάρτηση $f : \mathcal{N}^{n+1} \rightarrow \mathcal{N}$, που ορίζεται ως προς τις g, h με στοιχειώδη αναδρομή να ανήκει επίσης στο σύνολο των στοιχειωδών αναδρομικών συναρτήσεων.

Στην ουσία το σύνολο των στοιχειωδών αναδρομικών συναρτήσεων είναι το μικρότερο σύνολο συναρτήσεων, που περιέχει τις αρχέγονες συναρτήσεις και είναι κλειστό ως προς τις πράξεις της σύνθεσης και της στοιχειώδους αναδρομής.

Παράδειγμα Ε'.3.1. Αριθμητική με στοιχειώδους αναδρομικές συναρτήσεις. Οι συναρτήσεις $Add : \mathcal{N}^2 \rightarrow \mathcal{N}$ και $Mult : \mathcal{N}^2 \rightarrow \mathcal{N}$ ορίζονται από τους τύπους $Add(x, y) = x + y$, $Mult(x, y) = x \cdot y$ αντίστοιχα. Τόσο για την πρώτη όσο και για την δεύτερη μπορούμε να έχουμε μια παραγωγή εργαζόμενοι αντίστροφα με σκοπό να βρίσκουμε απλούστερες συναρτήσεις από τις οποίες μπορούμε να πάρουμε τη συνάρτηση, που μας ενδιαφέρει με στοιχειώδη αναδρομή.

$$Add(x, 0) = p_1^1(x) = x$$

$$Add(x, k + 1) = (x + k) + 1 = succ(Add(x, k)) = succ(p_3^3(x, k, Add(x, k)))$$

Επομένως μιας παραγωγή στοιχειώδους αναδρομής για την Add σχηματίζεται από τις τρεις αρχέγονες συναρτήσεις $p_1^1, succ, p_3^3$, καθώς και μια συνάρτηση f_3 που λαμβάνεται ως σύνθεση των $succ, p_3^3, Add$ που προκύπτει από τις p_1^1, f_3 με στοιχειώδη αναδρομή. Σχετικά με την $Mult$

$$Mult(x, 0) = 0 = C_0^1(x)$$

$$Mult(x, k + 1) = x \cdot k + x = Add(x, Mult(x, k))$$

Στην τελευταία έκφραση το πρώτο από τα ορίσματα της συνάρτησης προκύπτει ως $p_1^3(x, k, Mult(x, k))$ και το δεύτερο ως $p_3^3(x, k, Mult(x, k))$. Χρησιμοποιώντας λοιπόν αυτήν την προσέγγιση, ξεκινάμε με την παραγωγή της Add και συνεχίζουμε με την C_0^1 και τη συνάρτηση f που δίνεται ως σύνθεση της Add με τις $p_1^3, p_3^3, 3$ και ολοκληρώνουμε με την $Mult$, που προκύπτει από τις C_0^1 και f με στοιχειώδη αναδρομή.

Είναι κατάφωρο πλέον ότι οι αρχέγονες συναρτήσεις είναι υπολογίσιμες, καθώς επίσης και ότι το σύνολο των υπολογίσιμων συναρτήσεων είναι κλειστό ως προς τις πράξεις της σύνθεσης και της στοιχειώδους αναδρομής. Αυτό συμβαίνει διότι μπορούμε να κατασκευάσουμε MT, οι οποίες υπολογίζουν τις συναρτήσεις, που προκύπτουν από αυτές τις πράξεις με τη προϋπόθεση βέβαια ότι ξεκινάμε με υπολογίσιμες συναρτήσεις.

Ως συνέπεια της θέσης των Church -Turing έχουμε το εξής θεώρημα:

Θεώρημα Ε'.3.1. Υπολογισιμότητα Στοιχειώδους Αναδρομικών Συναρτήσεων. Κάθε στοιχειώδους αναδρομική συνάρτηση είναι πλήρης και υπολογίσιμη

Το ανάποδο δεν ισχύει.

Ε'.3.2 μ - αναδρομικές συναρτήσεις

Αφού λοιπόν υπάρχουν υπολογίσιμες συναρτήσεις που δεν είναι στοιχειώδεις αναδρομικές το επόμενο βήμα είναι να συνεχίσουμε την προσπάθεια για να ορίσουμε ακριβώς εκείνες τις συναρτήσεις που υπολογίζονται από τις ΜΤ.

Ορισμός Ε'.3.4. Ελαχιστοποίηση συνάρτησης. Η ελαχιστοποίηση μιας συνάρτησης $f : \mathbb{N} + \infty \rightarrow \mathbb{N}$ δίνεται από τη συνάρτηση $\min_f : \mathbb{N}^k \rightarrow \mathbb{N}$ που ορίζεται ως εξής:

$$\min_f(X) = z \stackrel{\text{def}}{\iff} f(z, X) = 0 \text{ και} \\ f(i, X) > 0 \text{ για } i = 0, \dots, z - 1$$

Διαισθητικά η ελαχιστοποίηση αναζητά ξεκινώντας από το 0 αυξητικά, το μικρότερο όρισμα της συνάρτησης, για το οποίο επιστρέφεται η τιμή 0. Αν δεν υπάρχει τέτοιο όρισμα, τότε η αναζήτηση δεν τερματίζει ποτέ. Αν λοιπόν η f είναι υπολογίσιμη, τότε μπορούμε να κατασκευάσουμε μια ΜΤ $M_{\min_f}$ για τον υπολογισμό της $\min_f$ που θα λειτουργεί ως εξής:

$$M_{\min_f}(X) =$$

1. $i = 0$
2. while $f(i, X) > 0$ do
 - $i = i + 1$
3. return i

Το πρόβλημα είναι ότι αν δεν υπάρχει κατάλληλο i , η $M_{\min_f}$ δε θα τερματίσει ποτέ. Οπότε επικεντρωνόμαστε σε μια κλάση συναρτήσεων για την οποία η $M_{\min_f}$ τερματίζει πάντα: μια συνάρτηση f είναι *ελαχιστοποιήσιμη*, αν και μόνο αν για κάθε $X \in \mathbb{N}^k$ υπάρχει ένα z τέτοιο ώστε $f(z, X) = 0$.

Ορισμός Ε'.3.5. μ -αναδρομικές συναρτήσεις. Ορίζουμε το σύνολο μ -αναδρομικών συναρτήσεων να είναι το μικρότερο σύνολο συναρτήσεων από το $\mathbb{N}^k$ στο $\mathbb{N}$, που περιλαμβάνει τις αρχέγονες συναρτήσεις και είναι κλειστός ως προς:

- τη σύνδεση συναρτήσεων
- τη στοιχειώδη αναδρομή και
- την ελαχιστοποίηση συναρτήσεων ελαχιστοποιήσιμων.

Η διαφορά μεταξύ των στοιχειωδών αναδρομικών συναρτήσεων και των μ -αναδρομικών συναρτήσεων είναι:

- Η επανάληψη στον υπολογισμό μιας στοιχειώδους αναδρομικής συνάρτησης είναι πάντα φραγμένη. Ένας τέτοιος υπολογισμός μπορεί να υλοποιηθεί με ένα βρόχο *for*, που θα εκτελείται για x_k βήματα, όπου x_k είναι η τιμή του τελευταίου ορίσματος της συνάρτησης. Έτσι για να υπολογιστεί η $Mult(2, 3)$ θα πρέπει να κληθεί η Add τρεις φορές.
- Ο υπολογισμός μιας μ -αναδρομικής συνάρτησης απαιτεί την εκτέλεση ενός βρόχου *for*, όπως αυτός για τη ΜΤ M_{min_f} , για τον οποίο δεν υπάρχει εκ των προτέρων γνωστό φράγμα, που περιορίζει τον αριθμό των απαιτούμενων βημάτων.

Θεώρημα Ε'.3.2. *Ισοδυναμία μ -αναδρομής και υπολογισιμότητας. μια συνάρτηση είναι μ -αναδρομική αν και μόνο αν είναι υπολογίσιμη.*

Θεώρημα Ε'.3.3. *μερικώς μ -αναδρομικών συναρτήσεων και υπολογισιμότητα. μια συνάρτηση είναι μερικώς μ -αναδρομική αν και μόνο αν είναι μερικώς υπολογίσιμη.*

Ε'.4 Θεώρημα Αναδρομής/Σταθερού Σημείου

Ένα άλλο ερώτημα που μπορεί να προκύψει είναι το κατά πόσο ένα πρόγραμμα μπορεί να γνωρίζει τον εαυτό του. Το πρόβλημα αυτό καλείται ως *ανάκληση* και μια από τις εφαρμογές του είναι η σχεδίαση ιών.

Οι ιοί είναι προγράμματα που αντιγράφουν τον εαυτό τους, εγγράφουν δηλαδή μια εικόνα του εκτελέσιμου κώδικα τους από έναν υπολογιστή σε ένα άλλο. Ας θεωρήσουμε μια απλή περίπτωση ενός προγράμματος που πρέπει να εκτυπώσει ένα αντίγραφο του εαυτού του. Τα προγράμματα αυτού του είδους είναι γνωστά με τον όρο *quines* προς τιμή του σημαντικού φιλοσόφου και επιστήμονα της Λογικής Willard Van Orman Quine.

μια άμεση προσέγγιση θα ήταν να έχουμε μια μεταβλητή δείκτη στη διεύθυνση του πρώτου byte από τον κώδικα του προγράμματος και στη συνέχεια να μετακινείται ο δείκτης με ένα επαναληπτικό βρόχο διατρέχοντας και τυπώνοντας τα περιεχόμενα των διευθύνσεων, στις οποίες είναι αποθηκευμένος ο κώδικας. Τα λειτουργικά συστήματα καταγράφουν πληροφορίες για το αν μια περιοχή μνήμης είναι εκτελέσιμη ή αν είναι μνήμη ανάγνωσης/εγγραφής. Έτσι σε μια περιοχή της μνήμης μπορούμε είτε να έχουμε ένα εκτελέσιμο πρόγραμμα, είτε εκεί να υπάρχουν δεδομένα. Για να εμποδιστεί λοιπόν ένα πρόγραμμα ιού από το να διαβάσει τον εαυτό του, θα μπορούσε η περιοχή του κώδικα του να χρησιμοποιείται αποκλειστικά ως εκτελέσιμη περιοχή. με αυτό τον τρόπο απαγορεύεται η εκτέλεση εντολών, που προσπελαίνουν μια περιοχή του προγράμματος, για να ανακτήσουν τα περιεχόμενα της ως

δεδομένα, που στη συνέχεια μπορεί να αντιγράφονται. Έτσι, είναι δυνατό τα προγράμματα να εμποδιστούν από το να διαβάσουν τα bytes του κώδικα τους.

Στο σημείο αυτό αναρωτιόμαστε: οι ιοί των υπολογιστών μπορούν να προσπελαύνουν τον εκτελέσιμο κώδικά τους με άλλο τρόπο;

Ας ξεκινήσουμε την προσπάθεια να γράψουμε ένα quine για ένα πολύ απλό πρόγραμμα:

```
PRINT'Hello'
```

Θέλουμε δηλαδή ένα πρόγραμμα που θα εκτυπώνει το παραπάνω πρόγραμμα. Ας εξετάσουμε την εξής λύση:

```
PRINT'PRINT'Hello''
```

Αυτο όμως δεν είναι ένα quine γιατί δεν εκτυπώνει τον εαυτό του, αλλά είναι ένα πρόγραμμα μεγαλύτερο από το αρχικό πρόγραμμα. Είναι λοιπόν σαφές ότι με μια επιπλέον PRINT τελικά παίρνουμε ένα διαφορετικό πρόγραμμα από που θέλουμε να εκτυπώσουμε και δεν είναι ένα quine.

Ας δοκιμάσουμε μια άλλη προσέγγιση. Χρειαζόμαστε ένα πρόγραμμα, που θα αποτελείται από μια εντολή εκχώρησης και μια εντολή PRINT:

$$x \leftarrow ' \dots '$$

```
PRINT x
```

Ένα quine για αυτό το πρόγραμμα θα είχε τη μορφή:

$$x \leftarrow ' \dots '$$

```
PRINT'x ←''
```

```
PRINT x
```

```
PRINT'''
```

```
PRINT x
```

όπου οι τρεις πρώτες PRINT τυπώνουν την εντολή εκχώρησης, δηλαδή ένα μέρος του προγράμματος και η τελευταία PRINT τυπώνει την τιμή της x , που μέχρι στιγμής δεν έχουμε ορίσει. Το πρόγραμμα αυτό θα τυπώνει τον εαυτό του, μόνο αν η τιμή της μεταβλητής x είναι η συμβολοσειρά, που αποτελείται από τις τέσσερις εντολές PRINT. Αν λοιπόν στην εντολή εκχώρησης γράψουμε στη θέση των $\dots$ τις τέσσερις εντολές PRINT τότε θα έχουμε ένα quine.

Ας εξετάσουμε τώρα το πρόβλημα της σχεδίασης μιας MT quine. Ονομάζουμε P_w μία απλή MT, που απλά τυπώνει τη συμβολοσειρά $w \in \Sigma^*$ στην ταινία της και συμβολίζουμε με

$\langle P_w \rangle$ μία κωδικοποίηση της P_w ως συμβολοσειρά. Η σχεδίαση της P_w είναι εύκολη: αν η w έχει μήκος n , τότε αρκεί μία ακολουθία από n μεταβάσεις, που η κάθε μία εγγράφει στην ταινία το αντίστοιχο σύμβολο της w . Η τελευταία κατάσταση θα πρέπει να είναι τελική. Κατά συνέπεια, η συνάρτηση $g : \Sigma^* \rightarrow \Sigma^*$ με $g(w) = \langle P_w \rangle$, που για δοθείσα συμβολοσειρά w , επιστρέφει ως αποτέλεσμα μία κωδικοποίηση $\langle P_w \rangle$ της MT P_w είναι μία υπολογίσιμη συνάρτηση. Η MT quine θα λειτουργεί σε δύο βήματα. Στο πρώτο βήμα εκτελείται ο υπολογισμός $g(x)$ εγγράφοντας πάνω στην ταινία τη συμβολοσειρά x , όπου x είναι η κωδικοποίηση της MT, που εφαρμόζεται στο δεύτερο βήμα. Όταν αρχίζει η εκτέλεση του δεύτερου βήματος, υπάρχει ήδη πάνω στην ταινία η συμβολοσειρά x , που αντιγράφεται στις θέσεις της ταινίας δεξιά από την x . Στη συνέχεια καλείται η $g(x)$, που εγγράφει στη θέση της πρώτης x την κωδικοποίηση της MT του πρώτου βήματος, δηλαδή της MT που εγγράφει x στην ταινία. Τελικά ποια πρέπει να είναι η συμβολοσειρά x , για να λειτουργεί η περιγραφείσα MT ως ένα quine; Η απάντηση είναι εύκολη: η x θα είναι η κωδικοποίηση της MT του δεύτερου βήματος. Έτσι, ως αποτέλεσμα του υπολογισμού, που περιγράψαμε, θα έχουμε πάνω στην ταινία στο αριστερό μέρος την κωδικοποίηση της MT του πρώτου βήματος και ακριβώς στα δεξιά την κωδικοποίηση της MT του δεύτερου βήματος. Θα έχουμε δηλαδή μία MT, που λειτουργεί σε δύο βήματα, εγγράφοντας πάνω στην ταινία την κωδικοποίηση του εαυτού της.

μία MT που γνωρίζει τον εαυτό της, όπως το quine που περιγράψαμε, μπορεί να χρησιμοποιηθεί για προβλήματα όπως:

- Καταμέτρηση του αριθμού των καταστάσεων της MT.
- Έλεγχος του αν η τελική κατάσταση προσεγγίζεται από την αρχική κατάσταση της MT.
- Έλεγχος του αν η MT αποδέχεται μία συμβολοσειρά w .

Το τελευταίο πρόβλημα είναι μη αποφασίσιμο και για αυτό χρειάζεται προσοχή στην προσομοίωση μιας τέτοιας MT. Θα μπορούσαμε βέβαια να διατυπώσουμε και πολλά άλλα προβλήματα, αλλά σε όλες τις περιπτώσεις χρειάζεται ο υπολογισμός μιας συνάρτησης της μορφής:

$$t(\langle M \rangle, w)$$

και επομένως η σχεδίαση μιας MT, που θα δέχεται ως είσοδο την κωδικοποίηση μιας MT M και ενδεχομένως επιπλέον πληροφορίες, όπως για παράδειγμα μία συμβολοσειρά w . Αν υπήρχε μία τέτοια MT, τότε θα μπορούσαμε να περάσουμε ως είσοδο σε αυτή τη μηχανή την κωδικοποίηση του εαυτού της, για να υπολογιστεί η:

$$t(\langle t \rangle, w)$$

και να δοθεί λύση σε κάποιο πρόβλημα, όπως αυτά που αναφέραμε. Υπάρχει περίπτωση να επιτευχθεί αυτό με άλλο τρόπο, χωρίς να χρειαστεί η MT να δεχθεί ως είσοδο μία κωδικοποίηση του εαυτού της; Η απάντηση είναι καταφατική και δίνεται από το Θεώρημα της

Αναδρομής, που στην ουσία μας λέει ότι αν υπάρχει μία MT για τη συνάρτηση t , τότε θα υπάρχει και μία άλλη MT, που κάνει ότι και η t , αλλά υπολογίζει την κωδικοποίηση του εαυτού της αντί απλά να δέχεται την κωδικοποίηση ως είσοδο.

Θεώρημα Ε'.4.1. *Θεώρημα της Αναδρομής. Έστω μία MT T , που υπολογίζει μία συνάρτηση $t : \Sigma^* \times \Sigma^* \rightarrow \Sigma^*$. Για κάθε τέτοια MT υπάρχει μία άλλη MT R , που κάνει ότι ακριβώς κάνει η t , όταν αυτή εφαρμόζεται σε μία κωδικοποίηση του εαυτού της. Η νέα αυτή MT, έστω R θα υπολογίζει τη συνάρτηση $r : \Sigma^* \rightarrow \Sigma^*$:*

$$r(w) = t(< R >, w)$$

για οποιοδήποτε $w \in \Sigma^*$.

Αυτό λοιπόν που μας λέει το Θεώρημα της Αναδρομής είναι ότι στον ψευδοκώδικα μιας MT μπορούμε βάσιμα να συμπεριλάβουμε μία εντολή $< SELF >$, η οποία θα επιστρέφει μία κωδικοποίηση της ίδιας MT ως συμβολοσειρά.

Εφόσον λοιπόν υποστηρίζεται από την ψευδογλώσσα αυτή η δυνατότητα, το πρόγραμμα quine, που μελετήσαμε θα μπορούσε να γραφεί με τον εξής απλό τρόπο:

$x \leftarrow < SELF >$

PRINT x

που σύμφωνα με το Θεώρημα της Αναδρομής είναι μία καθόλα έγκυρη περιγραφή MT.

Το Θεώρημα της Αναδρομής έχει διατυπωθεί επίσης και σε μία άλλη μορφή, που αναφέρεται σε σταθερό σημείο μιας συνάρτησης.

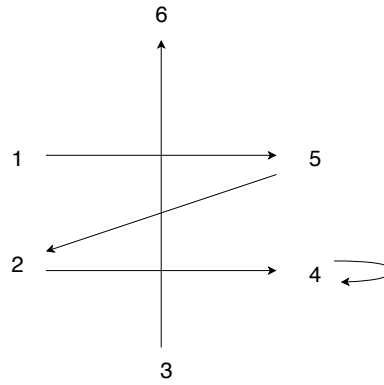
Ορισμός Ε'.4.1. *Σταθερό σημείο συνάρτησης. Ένα σταθερό σημείο για μία συνάρτηση είναι κάποια τιμή, που παραμένει αμετάβλητη όταν έχουμε επαναλαμβανόμενη εφαρμογή της συνάρτησης για την τιμή αυτή.*

Παράδειγμα Ε'.4.1. *Σταθερό σημείο συνάρτησης. Θεωρήστε τη συνάρτηση f , που ορίζεται από τον πίνακα:*

x	1	2	3	4	5	6
$f(x)$	5	4	6	4	2	4

και απεικονίζεται στο γράφημα του Σχήματος Ε.16 Η τιμή 4 είναι ένα σταθερό σημείο της συνάρτησης f .

Το Θεώρημα Αναδρομής Σταθερού Σημείου αναφέρεται σε υπολογίσιμες συναρτήσεις, που ορίζονται και παίρνουν τιμές στο σύνολο των συμβολοσειρών, που κωδικοποιούν MT. Κάθε συνάρτηση της οποίας το πεδίο ορισμού και το πεδίο τιμών είναι το ίδιο σύνολο λέγεται *μετασχηματισμός*. Το θεώρημα λέει ότι για οποιονδήποτε μετασχηματισμό κωδικοποιήσεων MT υπάρχει πάντα μία MT, που παραμένει αναλλοίωτη από το μετασχηματισμό.



Σχήμα Ε'.16: Συνάρτηση με σταθερό σημείο

Θεώρημα Ε'.4.2. Θεώρημα Αναδρομής Σταθερού Σημείου. Έστω μία MTT , που υπολογίζει μία συνάρτηση $t : \Sigma^* \times \Sigma^* \rightarrow \Sigma^*$. Υπάρχει μία κατάληψη MTF , τέτοια ώστε η $t(<F>)$ να είναι ισοδύναμη με την F .

Κεφάλαιο Γ'

Επίλογος

Στην παρούσα διπλωματική εργασία τονίσαμε ότι ούτε τα πεπερασμένα αυτόματα, ούτε τα αυτόματα στοιβάς μπορούν να θεωρηθούν γενικά μοντέλα υπολογισμού, αφού δεν μπορούν να αναγνωρίσουν γλώσσες όπως την $\{a^n b^n c^n : n \geq 0\}$. Τα πεπερασμένα αυτόματα μπορούν να "θυμούνται" μόνο ένα πεπερασμένο ποσό πληροφορίας, ενώ τα αυτόματα στοιβάς μπορούν να επεξεργαστούν την πληροφορία σύμφωνα με τους κανόνες που καθορίζουν τη δομή του σωρού.

Ο Άγγλος μαθηματικός, ο Alan Turing προσπάθησε να ορίσει μια μηχανή που να αποτελεί ένα "μοντέλο υπολογισμού". Το πρότυπό του ήταν ένας "ανθρώπινος υπολογιστής", δηλαδή ένας άνθρωπος με μολύβι και χαρτί που προσπαθεί να λύσει ένα πρόβλημα με κάποιο συστηματικό τρόπο. Ο Turing εντοπίζοντας τα στοιχειώδη βήματα που απαιτούσε η διαδικασία της επίλυσης από τον άνθρωπο, προσπάθησε να ορίσει μια μηχανή που θα ήταν ικανή να λειτουργήσει με παρόμοιο τρόπο. Ήθελε έτσι να καταλήξει στο συμπέρασμα ότι κάθε αλγοριθμική διαδικασία θα μπορούσε να εκτελεστεί και από μια μηχανή με ένα βασικό ρεπερτόριο εντολών.

Οι στοιχειώδεις λειτουργίες ενός υπολογισμού από έναν άνθρωπο σύμφωνα με τον Turing είναι η εξέταση κάποιου συμβόλου, η αντικατάστασή του με κάποιο άλλο και η μεταβίβαση της προσοχής σε κάποιο άλλο σημείο του χαρτιού. Όλα αυτά όμως με την προϋπόθεση ότι η επιλογή μεταξύ αυτών των λειτουργιών εξαρτάται από το τρέχον σύμβολο και την προηγούμενη κατάσταση στην οποία είχε περιέλθει ο "ανθρώπινος υπολογιστής".

Έτσι η μηχανή που προτάθηκε από τον Turing αποτελούταν αρχικά από ένα πεπερασμένο σύνολο καταστάσεων, όπως ακριβώς τα πεπερασμένα αυτόματα και τα αυτόματα στοιβάς. Η μηχανή μπορούσε να διαβάζει και να γράφει ένα σύμβολο, όχι όμως σε ένα κομμάτι χαρτί, αλλά σε μια ταινία που ήταν χωρισμένη σε τετράγωνα και εκτεινόταν στο άπειρο από το ένα άκρο. Η διαδικασία της ανάγνωσης γινόταν με μια κεφαλή, η οποία μπορούσε να κινηθεί

αριστερά ή δεξιά και κάθε στιγμή άλλαζε τα περιεχόμενα ενός μόνο τετραγώνου. Η απόφαση της μετακίνησης εξαρτιόταν από το σύμβολο κάτω από την κεφαλή και από την αντικατάσταση που βρισκόταν η μηχανή. Η είσοδος στη μηχανή δινόταν γράφοντας τη συμβολοσειρά πάνω στην ταινία, η οποία χρησίμευε επίσης ως μνήμη για την εκτέλεση επιμέρους υπολογισμών αλλά και ως έξοδος, αφού πάνω σ' αυτή μπορούσε να γραφεί μια απάντηση στο τέλος του υπολογισμού.

Τι είναι αυτό ακριβώς που μας κάνει να πιστεύουμε ότι μια τέτοια μηχανή θα μπορούσε να αποτελεί ένα γενικό μοντέλο υπολογισμού; Η απάντηση είναι ότι οποιαδήποτε επέκταση της βασικής μηχανής δε φαίνεται να προσθέτει σε *υπολογιστική ισχύ*, αλλά είναι *ισοδύναμη* με την αρχική. Φυσικά, όλες οι επεκτάσεις θα πρέπει να κινούνται στο ίδιο κλίμα, δηλαδή οι λειτουργίες της μηχανής θα πρέπει να μπορούν να εκτελεστούν μόνο με ένα πεπερασμένο πόσο δουλειάς σε κάθε βήμα. Αν επιτρέπαμε τη δυνατότητα η μηχανή μας να μπορεί να απαντήσει άπειρες ερωτήσεις σ' ένα μόνο βήμα, τότε τα μοντέλα δεν θα ήταν ίδια.

Ορίσαμε τον υπολογισμό ως κάτι που μπορεί να εκτελεστεί από μια τέτοια μηχανή. μελετώντας τις δυνατότητες των μηχανών αυτών μπορούμε να καταλάβουμε καλύτερα την έννοια του υπολογισμού και γενικότερα τι μπορεί να επιλυθεί από οποιαδήποτε αλγοριθμική διαδικασία. Τονίσαμε ότι υπάρχουν πρόβλήματα που μπορούν να λυθούν όσο δύσκολο και να είναι κάτι τέτοιο, αλλά δυστυχώς δεν έχουν όλα τα προβλήματα λύσεις. Υπάρχουν προβλήματα, των οποίων οι λύσεις θα ήταν εξαιρετικά χρήσιμες, τα οποία όμως δεν μπορούν να λυθούν από καμία μηχανή Turing και άρα από οποιαδήποτε αλγοριθμική διαδικασία.

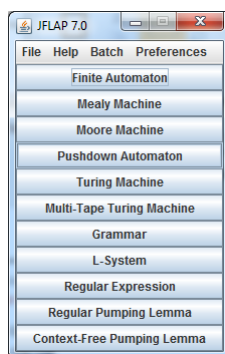
Παράρτημα Α΄

Σχεδίαση και Προσομοίωση Αυτομάτων με χρήση του JFLAP Για την σχεδίαση και την προσομοίωση των αυτομάτων που περιγράψαμε στα διάφορα παραδείγματα χρησιμοποιήσαμε ένα διαδραστικό εκπαιδευτικό λογισμικό το JFLAP (Java Formal Languages and Automata Package). Το λογισμικό αυτό είναι γραμμένο σε Java και χρησιμεύει στην προσομοίωση με θέματα στον τομέα των Επιστημών Πληροφορικής και της Θεωρίας των Αυτομάτων.

Σχεδίαση και Προσομοίωση Πεπερασμένων Αυτομάτων με χρήση του JFLAP

Στο Σχήμα Γ΄.1 σχεδιάσαμε ένα απλό πεπερασμένο αυτόματο που αποδέχεται συμβολοσειρές από 0 και 1, που τελειώνουν σε 00.

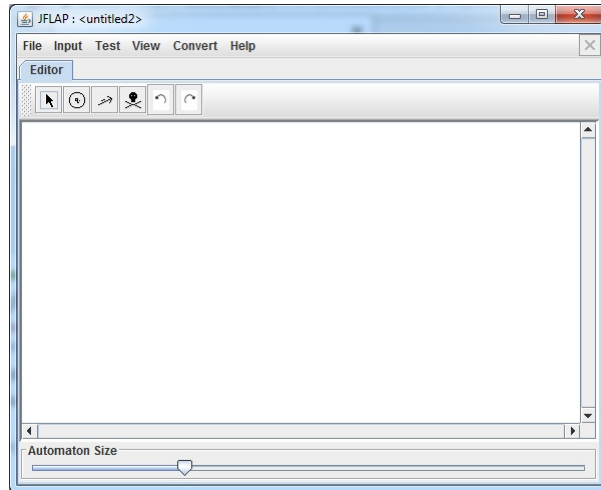
Στο λογισμικό του JFLAP επιλέγουμε από το μενού *Finite Automaton*. Η επιλογή αυτή ανοί-



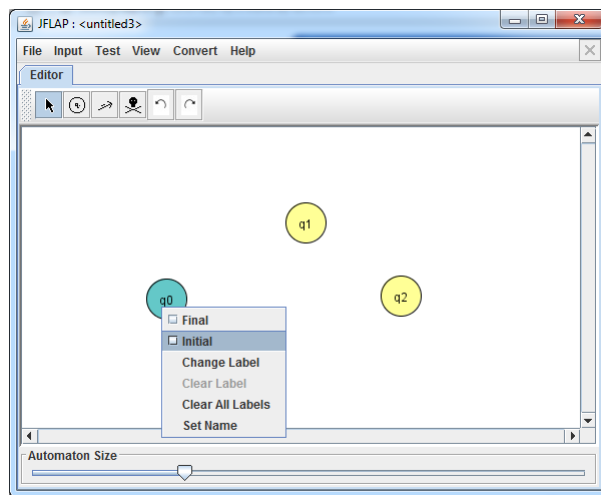
Σχήμα ΄.17: JFLAP menu

γει ένα νέο παράθυρο για τον σχεδιασμό του πεπερασμένου αυτόματου. Παρατηρούμε ότι ο editor είναι χωρισμένος σε δύο περιοχές. Στην μια βρίσκονται τα εργαλεία με τις επιλογές δημιουργίας καταστάσεων και μεταβάσεων, καθώς και διαγραφής ή αναίρεσης πράξης. Στην άλλη υπάρχει ένας καθαρός "καμβάς" όπου μπορούμε να σχεδιάσουμε το αυτόματο.

Δημιουργούμε τις καταστάσεις και καθορίζουμε ποιά είναι η αρχική και τελική κατάσταση με δεξί κλικ πάνω στην κατάσταση που μας ενδιαφέρει. Έπειτα δημιουργούμε τις μεταβάσεις με την βοήθεια της κατάλληλης εντολής από τα εργαλεία. Σημειώνουμε ότι το κενό συμβολίζεται με λ. Και τελικά έχουμε το πεπερασμένο αυτόματο του Σχήματος Γ΄.1. Έπει-

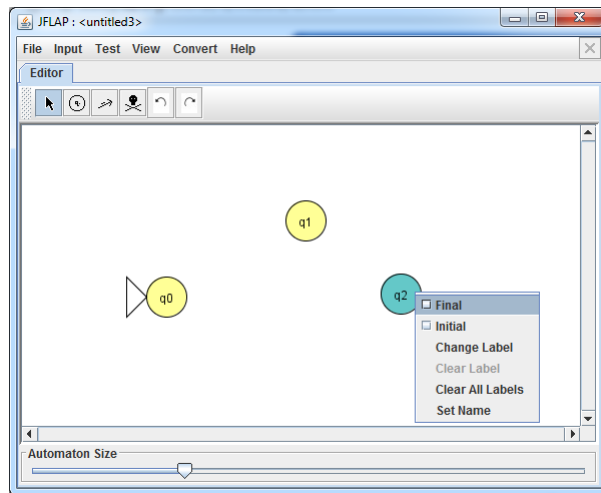


Σχήμα ' .18: Παράθυρο JFLAP για Πεπερασμένο Αυτόματο

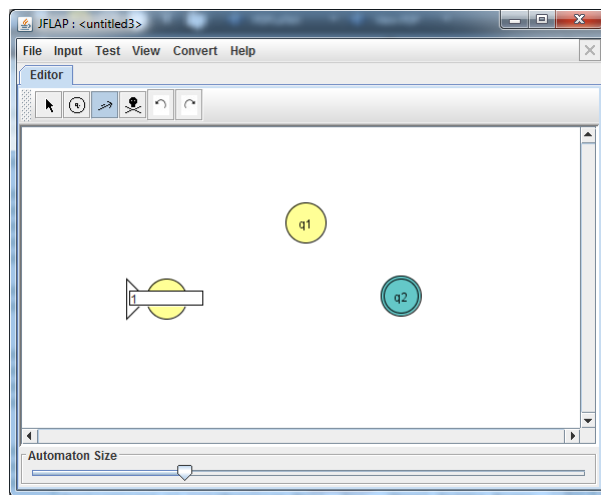


Σχήμα ' .19: Καθορισμός αρχικής κατάστασης στο JFLAP για Πεπερασμένο Αυτόματο

τα για τον έλεγχο του σχεδιασμού του Πεπερασμένου Αυτόματου που σχεδιάσαμε μπορούμε να εισάγουμε συμβολοσειρές και να "τρέξουμε" το αυτόματο για να δούμε την απόδοξη ή την απόρριψη των συμβολοσειρών. Ενώ με τον τρόπο αυτόν γενικά μπορούμε να ελέγξουμε οποιαδήποτε συμβολοσειρά αν είναι αποδεκτή ή όχι από το Πεπερασμένο Αυτόματο. Στην πράξη επιλέγουμε την εντολή "Multiple Run" από την καρτέλα "Input" στην γραμμή του μενού του λογισμικού. Τότε ο καμβάς χωρίζεται σε δύο περιοχές. Η νέα περιοχή έχει εισόδους για τις συμβολοσειρές που θέλουμε να ελέγξουμε πατώντας την εντολή "Run Inputs". Τελικά έχουμε το αποτέλεσμα δίπλα στις συμβολοσειρές ως "Accept" ή "Reject".



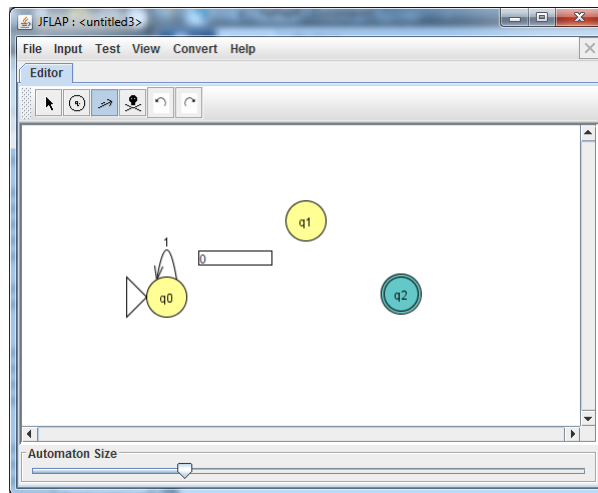
Σχήμα ' .20: Καθορισμός τελικής κατάστασης στο JFLAP για Πεπερασμένο Αυτόματο



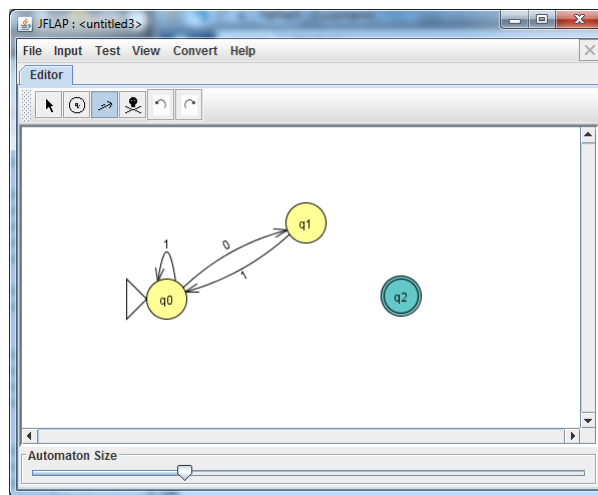
Σχήμα ' .21: Δημιουργία μεταβάσης στο JFLAP για Πεπερασμένο Αυτόματο

Σχεδίαση και Προσομοίωση Αυτομάτων Στοίβας με χρήση του JFLAP

Για τον σχεδιασμό ενός αυτόματου στοίβας επιλέγουμε την εντολή Pushdown Automato από το μενού του JFLAP. Όπως και στα πεπερασμένα αυτόματα έτσι και εδώ εμφανίζεται ένα παράθυρο με τα εργαλεία και τον καμβά που μπορούμε να σχεδιάσουμε. Στο Σχήμα Γ'.2 φαίνεται ένα αυτόματο στοίβας που αποδέχεται τη γλώσσα $L_1 = \{a^n b^n : n \geq 0\}$. Σε κάθε συμβολοσειρά της γλώσσας αυτής προηγούνται όλα τα a και ο αριθμός των a είναι ίσος με τον αριθμό των b , που έπονται. Έτσι σχεδιάζουμε το αυτόματο στοίβας όπως φαίνεται. Όπως παρατηρούμε στις επιγραφές της μορφής $x, y; z$ το αυτόματο στοίβας διαβάζει ένα x και εξάγει ένα y από τη στοίβα και στη συνέχεια αφαιρεί ένα y από τη στοίβα και βάζει σε αυτήν ένα z . Αν στο τέλος έχει διαβαστεί και ο τελευταίος χαρακτήρας της συμβολοσειράς και η στοίβα έχει αδειάσει, ενώ το αυτόματο βρίσκεται σε μια τελική κατάσταση τότε η συμβολοσειρά



Σχήμα 1.22: Δημιουργία μεταβάσης στο JFLAP για Πεπερασμένο Αυτόματο

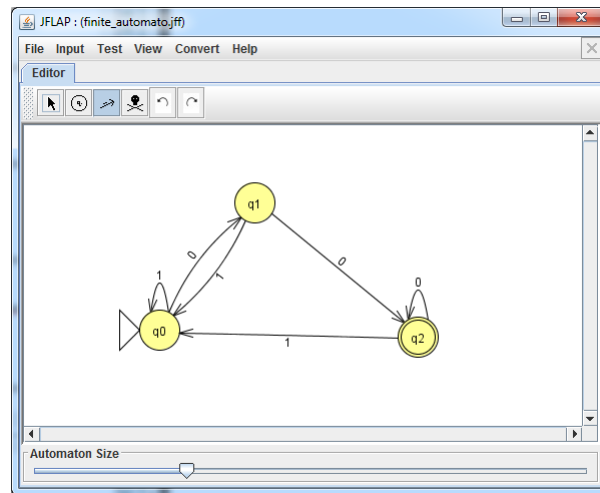


Σχήμα 1.23: Δημιουργία μεταβάσης στο JFLAP για Πεπερασμένο Αυτόματο

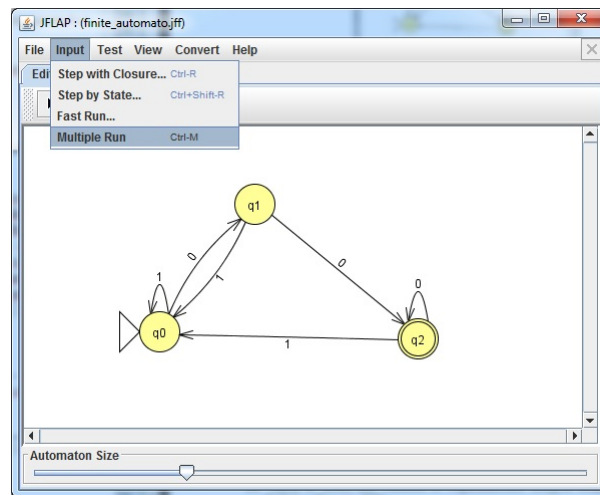
γίνεται δεκτή. Σε κάθε άλλη περίπτωση η συμβολοσειρά απορρίπτεται. Εξίσου μπορούμε να εισάγουμε οποιαδήποτε συμβολοσειρά και να ελέγχουμε αν είναι αποδεκτή ή όχι από το αυτόματο στοίβας που σχεδιάσαμε.

Σχεδίαση και Προσομοίωση μηχανών Turing με χρήση του JFLAP

Για τον σχεδιασμό μιας μηχανής Turing επιλέγουμε την εντολή Turing Machine από το μενού του JFLAP. Όπως και στα άλλα αυτόματα έτσι κ εδώ σχεδιάζουμε μια μηχανή Turing που όπως φαίνεται στο Σχήμα Γ.4 αποδέχεται την γλώσσα $\{a^n b^n c^n : n \geq 0\}$. Εξίσου μπορούμε να εισάγουμε οποιαδήποτε συμβολοσειρά και να ελέγχουμε αν είναι αποδεκτή ή όχι από τη μηχανή Turing που σχεδιάσαμε.



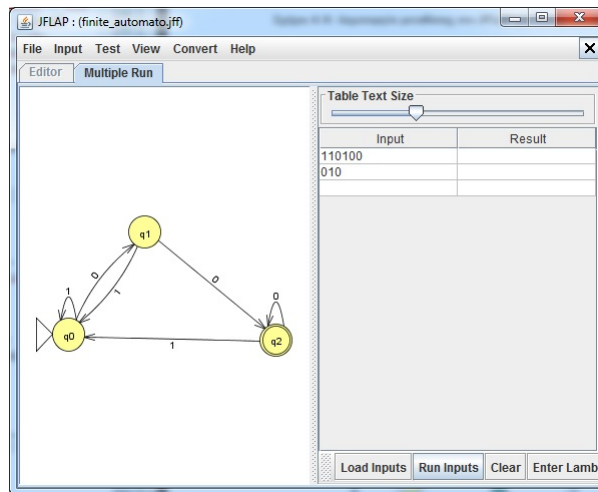
Σχήμα 7.24: Ένα απλό Πεπερασμένο Αυτόματο στο JFLAP



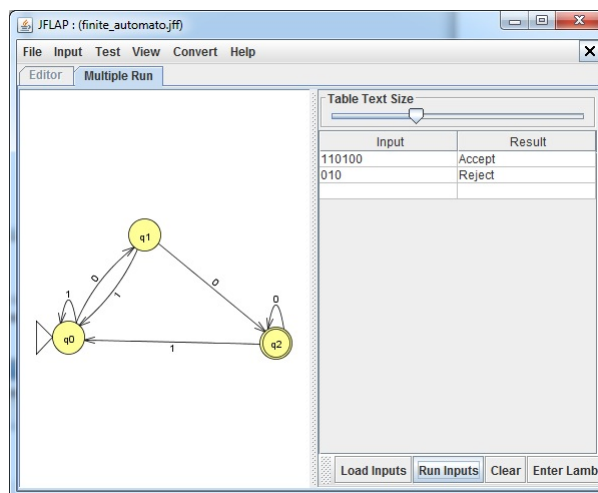
Σχήμα 7.25: Ένα απλό Πεπερασμένο Αυτόματο στο JFLAP

Σχεδίαση και Προσομοίωση Μη Ντετερμινιστικών Πεπερασμένων Αυτομάτων με χρήση του JFLAP

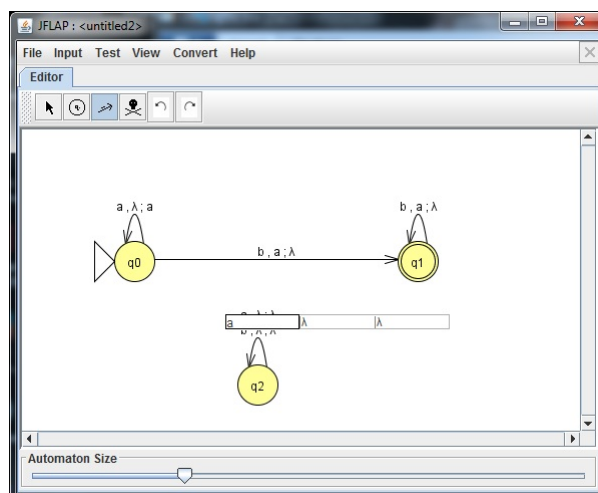
Στο Σχήμα 7.6 σχεδιάσαμε ένα μη ντετερμινιστικό αυτόματο με δύο μεταβάσεις με 0 από την q_0 . Από το μενού στις επιλογές "Test" από το μενού μπορούμε να επιλέξουμε την εντολή "Highlight Nondeterminism" και τότε φαίνεται να "φωτίζονται" οι καταστάσεις q_1, q_2 οι οποίες αποτελούν τις μη ντετερμινιστικές καταστάσεις, από την κατάσταση q_0 οι μη ντετερμινιστικές καταστάσεις είναι και οι δύο καταστάσεις μετάβασης. Έπειτα μπορούμε με την επιλογή της εντολής "Step by State" να δούμε αναλυτικά σε ποιά κατάσταση βρίσκεται το αυτόματο για τον έλεγχο μιας συμβολοσειράς. Παραθέτουμε μια εικόνα που φαίνεται το αυτόματο να διαβάζει τον δεύτερο χαρακτήρα μιας συμβολοσειράς και να βρίσκεται σε δύο καταστάσεις "ταυτόχρονα".



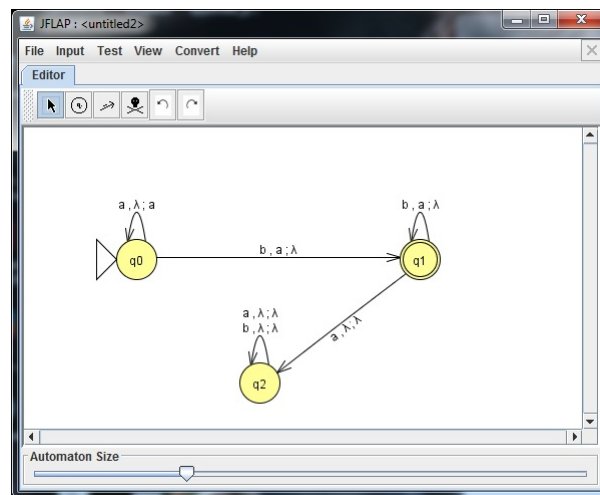
Σχήμα 1.26: Έλεγχος Συμβολοσειρών σε ένα απλό Πεπερασμένο Αυτόματο στο JFLAP



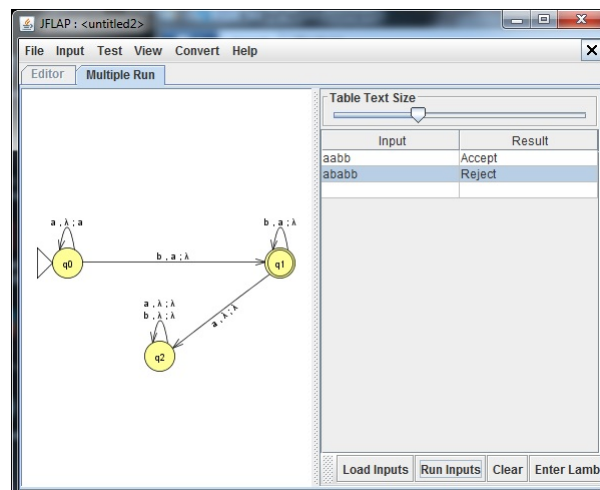
Σχήμα 1.27: Αποτέλεσμα ελέγχου Συμβολοσειρών σε ένα απλό Πεπερασμένο Αυτόματο στο JFLAP



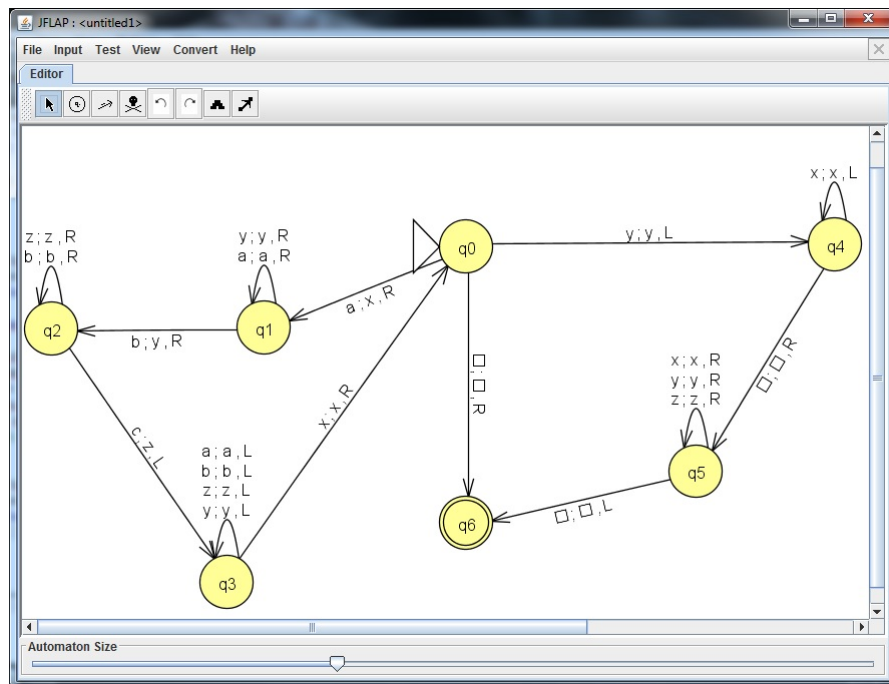
Σχήμα 1.28: Σχεδιασμός ενός απλού αυτόματου στοιβάς στο JFLAP



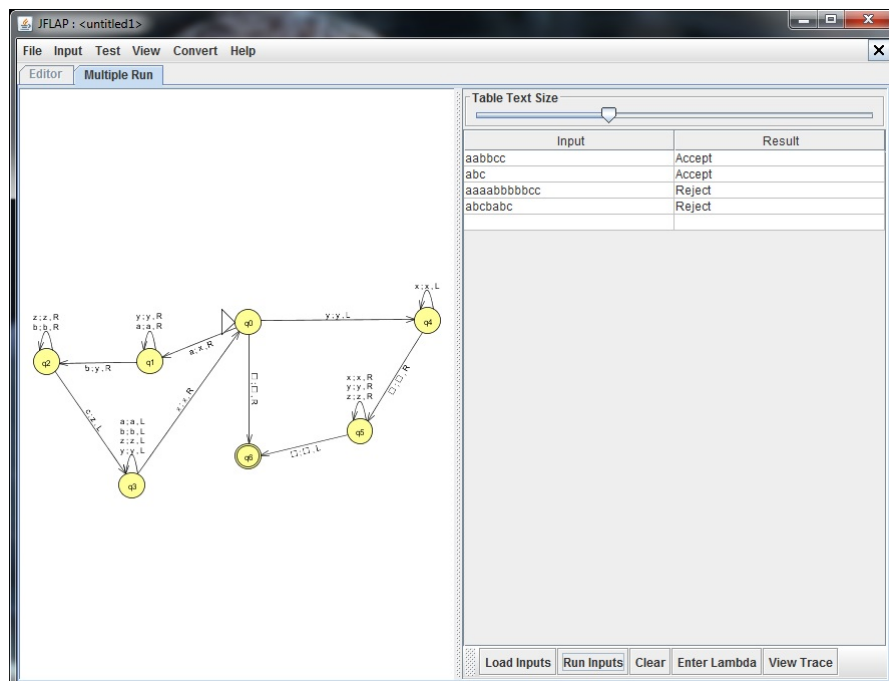
Σχήμα 1.29: Τελικός Σχεδιασμός ενός απλού αυτόματου στοιδας στο JFLAP



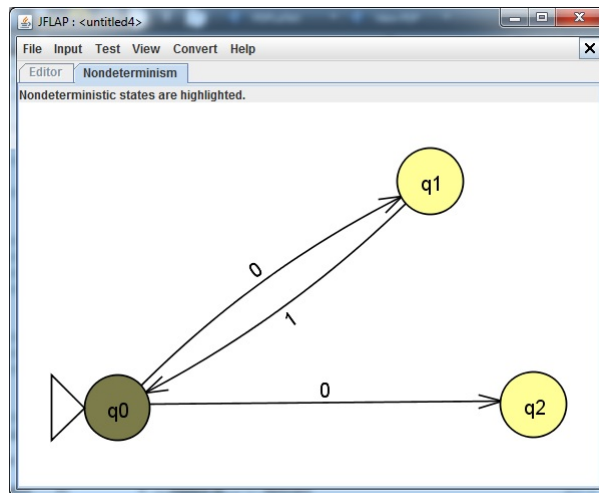
Σχήμα 1.30: Έλεγχος συμβολισμών για ένα απλό αυτόματο στοιδας στο JFLAP



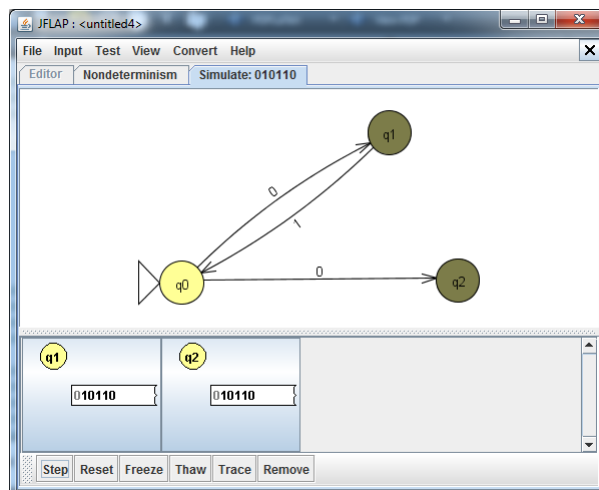
Σχήμα '31: Σχεδιασμός μιας μηχανής Turing στο JFLAP



Σχήμα '32: Έλεγχος Συμβολοσειρών μιας μηχανής Turing στο JFLAP



Σχήμα 1.33: Έλεγχος Μη Ντετερμινιστικών καταστάσεων ενός αυτομάτου στο JFLAP



Σχήμα 1.34: Μεταβάσεις ενός αυτομάτου στο JFLAP

Βιβλιογραφία

- [1] Harry R. Lewis and Christos H. Papadimitriou. *Στοιχεία Θεωρίας Υπολογισμού*. Εκδόσεις Κριτική, 2005.
- [2] Elaine A. Rich. *Automata Theory and Applications*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 2007. ISBN 0132288060.
- [3] J. Hopcroft, R. Motwani, J. Ullman. *Introduction to Automata Theory, Languages, and Computation* (Addison-Wesley, 2000).
- [4] M. Sipser. *Introduction to the theory of computation* (PWS Publishing Company, 1997; 2nd Edition: Course Technology, 2005).
- [5] Susan H. Rodger and Thomas W. Finley *JFLAP: An Interactive Formal Languages and Automata Package* 2006(c) Jones Bartlett Publishers, Sudbury, MA ISBN 0763738344