Θεωρία Υπολογισμού

Το Δόγμα Church-Turing

Στην ενότητα αυτή θα μελετηθούν τα εξής επιμέρους θέματα:

Μηχανές Turing (3.1)

- *Τυπικό Ορισμός*
- *Παραδείγματα*

Παραλλαγές Μηχανών Turing (3.2)

- *Πολυταινιακές Μηχανές Turing*
- *Μη ντετερμινιστικές Μηχανές Turing*

Το Δόγμα των Church-Turing

- Αν υπάρχει κάποια μέθοδος (αλγόριθμος) μέσω της οποίας μπορούμε να διεκπεραιώσουμε κάποιο υπολογισμό, τότε ο ίδιος υπολογισμός μπορεί να διεκπεραιωθεί μέσω μιας μηχανής Turing (ή μέσω μιας συνάρτησης διατυπωμένης στο λ-calculus ή μέσω μίας αναδρομικής συνάρτησης).

Alonzo Church (1903-1995)

Alan Turing (1912-1954)

Τι είναι ένας υπολογιστής;

- Ένας υπολογιστής είναι μια μηχανή η οποία επεξεργάζεται την είσοδό της σύμφωνα με μια ακολουθία από εντολές.

Υπολογιστικά μοντέλα που έχουμε μελετήσει

- Πεπερασμένα Αυτόματα
  - Το πιο απλό υπολογιστικό μοντέλο
  - Πεπερασμένη μνήμη

- Αυτόματα στοίβας
  - Άπειρη μνήμη – προσπελάσιμη μόνο ως στοίβα

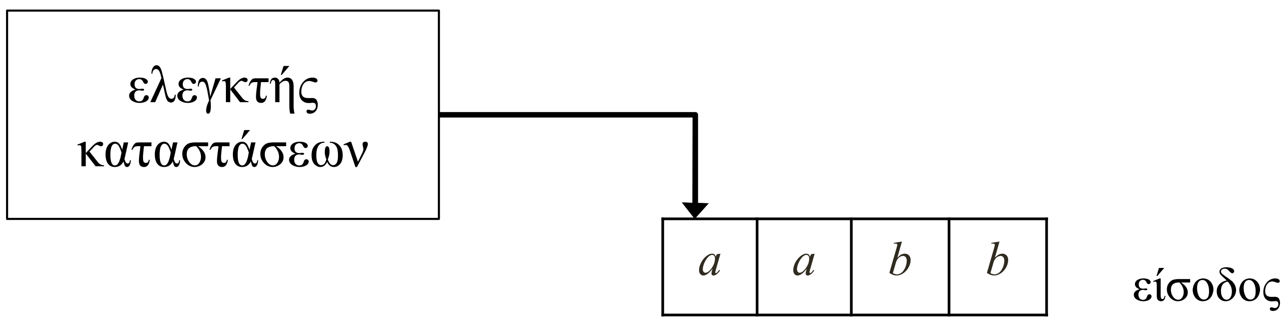
- Και οι δύο πιο πάνω μηχανές αδυνατούν να διεκπεραιώσουν πολύ απλές εργασίες
  - Δεν μπορούν να χρησιμοποιηθούν ως μοντέλα υπολογιστών γενικής χρήσης

Μηχανές Turing (Turing Machine, TM)

- Μηχανή Turing (Alan Turing 1936)
  - Πολύ πιο ισχυρό μοντέλο
- Διαθέτει άπειρη μνήμη
  - Μπορεί να την προσπελάσει χωρίς περιορισμούς
- Μπορεί να κάνει ακριβώς ότι και ένας υπολογιστής
- Εξακολουθούν να υπάρχουν προβλήματα που δεν μπορεί να επιλύσει
  - Αυτά τα προβλήματα βρίσκονται πέρα από τα όρια των υπολογιστικών δυνατοτήτων

Σύγκριση TM με DFA και PDA

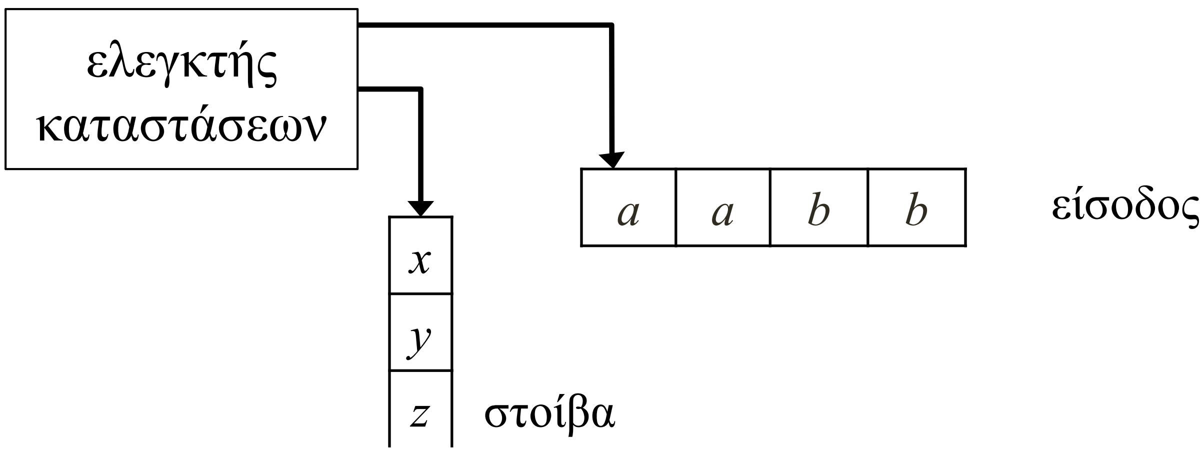
- Σχηματική αναπαράσταση πεπερασμένου αυτόματου
  - Ελεγκτής: Καταστάσεις και Μεταβάσεις Καταστάσεων
  - Ταινία: Λέξη Εισόδου
  - Βέλος : Κεφαλή Εισόδου



- Ο ελεγκτής διαβάζει το επόμενο σύμβολο και βάσει αυτού καθορίζει την επόμενη κατάσταση του αυτομάτου.

Σύγκριση TM με DFA και PDA

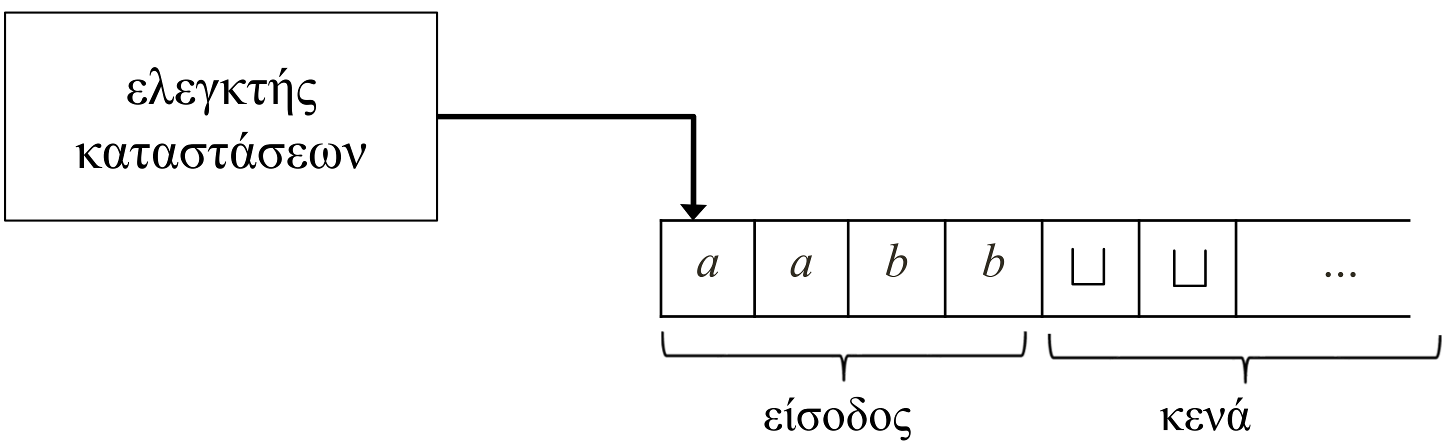
- Σχηματική Αναπαράσταση πεπερασμένου αυτόματου στοίβας
  - Ελεγκτής: Καταστάσεις και Μεταβάσεις Καταστάσεων
  - Ταινία: Λέξη Εισόδου
  - Βέλος : Κεφαλή Εισόδου
  - Στοίβα: Εγγραφή και διάβασμα συμβόλων



- Ο ελεγκτής διαβάζει (1) το επόμενο σύμβολο και (2) το στοιχείο στην κορυφή της στοίβας και, βάσει αυτής της δυάδας, αποφασίζει ποια θα είναι η επόμενη κατάσταση.

Σύγκριση TM με DFA και PDA

- Σχηματική αναπαράσταση Μηχανής Turing
  - Ελεγκτής: Καταστάσεις και Μεταβάσεις Καταστάσεων
    - Καταστάσεις αποδοχής και απόρριψης
    - Αν δεν φτάσουμε σε αποδοχή ή απόρριψη η TM δεν τερματίζει
  - Ταινία: Άπειρη
    - Αρχικά περιέχει τη λέξη εισόδου και σύμβολα διαστήματος στις υπόλοιπες θέσεις
  - Βέλος : Κεφαλή Εισόδου
    - Μετακινείται δεξιά και αριστερά
    - Δυνατότητα εγγραφής και ανάγνωσης



Παράδειγμα

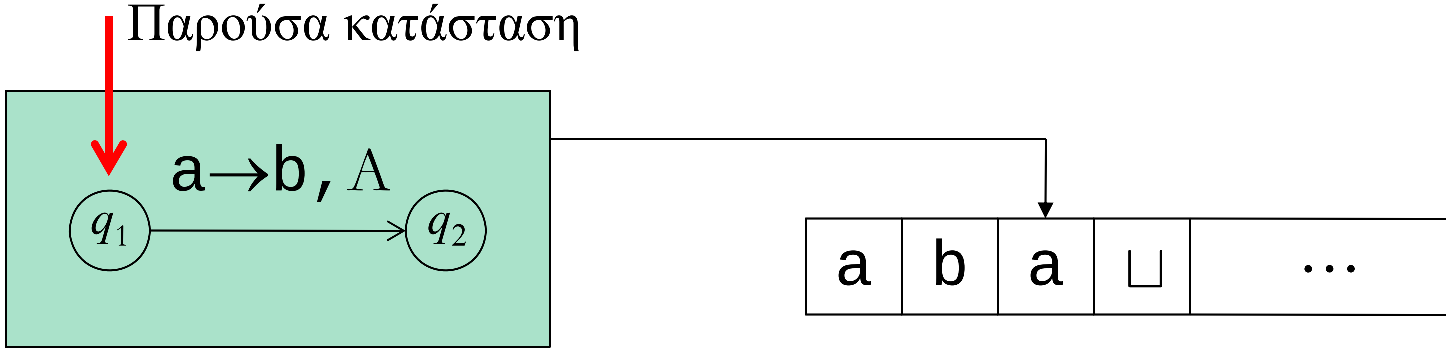
$L_1 = \{w\#w: w \in \{\mathrm{a}, \mathrm{b}\}^*\}$

<u>Στρατηγική</u>

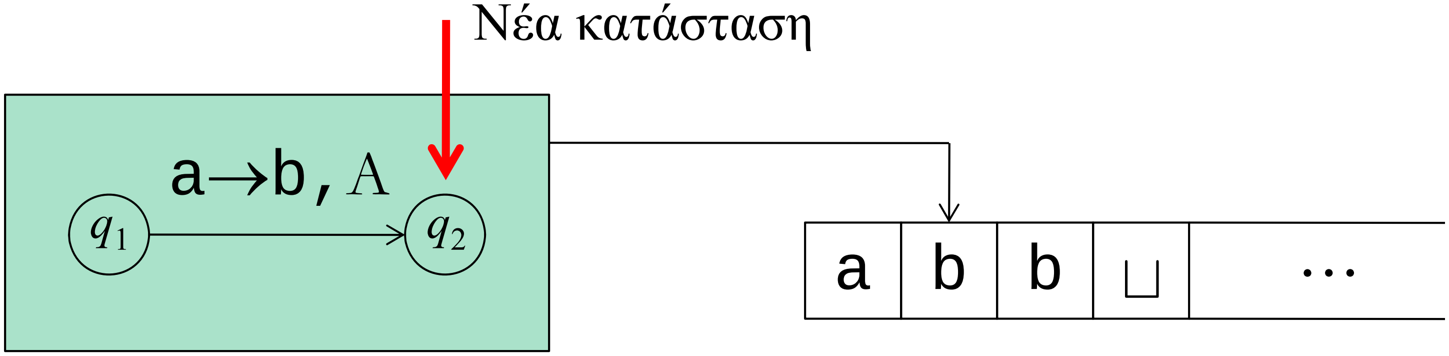
Διάβασε το πρώτο σύμβολο	<u>a</u>bbaa#abbaa
Διάγραψε το σύμβολο που διάβασες (x)	<u>x</u>bbaa#abbaa
Διάβασε το πρώτο σύμβολο μετά από το #	xbbaa#<u>a</u>bbaa
Αν τα δύο σύμβολα που διάβασες δεν είναι όμοια τότε απόρριψε τη λέξη	
Διαφορετικά, διάγραψε και το δεύτερο σύμβολο	xbbaa#<u>x</u>bbaa

Στο τέλος πρέπει να υπάρχουν μόνο x και #.
Αν όχι , απορρίπτουμε. Διαφορετικά αποδεχόμαστε. xxxxx#xxxxx

Λειτουργία Μηχανών Turing



Από την κατάσταση q_1 αν το επόμενο γράμμα είναι a, τότε το αντικαθιστούμε με b, και μετακινούμαστε στη ταινία μια θέση προς τα αριστερά.



Τυπικός ορισμός TM

ΟΡΙΣΜΟΣ

Μηχανή Turing είναι μια επτάδα $(Q, \Sigma, \Gamma, \delta, q_o, q_{acc}, q_{rej})$

1. Το Q είναι ένα πεπερασμένο *σύνολο καταστάσεων*
2. Το Σ είναι το *αλφάβητο εισόδου*, που δεν περιέχει το σύμβολο διαστήματος $\sqcup$
3. Το Γ είναι το *αλφάβητο ταινίας*, με $\sqcup \in \Gamma$, $\Sigma \subseteq \Gamma$
4. Η *συνάρτηση μετάβασης* $\delta: (Q - \{q_{acc}, q_{rej}\}) \times \Gamma \rightarrow Q \times \Gamma \times \{A, \Delta\}$
5. $q_o \in Q$ είναι η *εναρκτήρια κατάσταση*
6. $q_{acc} \in Q$ είναι η *κατάσταση αποδοχής*
7. $q_{rej} \in Q$ είναι η *κατάσταση απόρριψης*, με $q_{rej} \neq q_{acc}$

Σημείωση: Οι μηχανές Turing είναι ντετερμινιστικές

Πως υπολογίζει μια TM

- Μια TM υπολογίζει ως εξής:

- Με δεδομένο εισόδου μια λέξη *w* αρχικά:
  - Η λέξη *w* καταχωρείται στις αρχικές θέσεις της ταινίας
  - Η κεφαλή τοποθετείται στην αριστερότερη θέση
  - Το υπόλοιπο της ταινίας περιέχει το σύμβολο ⊔
    - Το πρώτο ⊔ σηματοδοτεί το τέλος της λέξης εισόδου

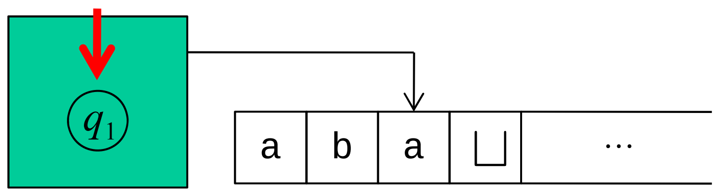
- Στη συνέχεια
  - Η μηχανή ακολουθεί τις μεταβάσεις
  - Συνεχίζουμε μέχρι να φτάσουμε σε κατάσταση αποδοχής ή απόρριψης.

Λειτουργία μιας μηχανής Turing

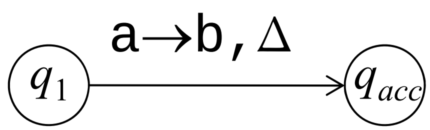
- Κατά την εκτέλεση μιας μετάβαση σε μια TM, αλλάζουν:
  - Η κατάσταση στην οποία βρίσκεται η μηχανή
  - Το περιεχόμενο της ταινίας
  - Η θέση της κεφαλής
- Κάθε συνδυασμός για αυτές τις τρεις οντότητες ονομάζεται φάση της μηχανής.
- Αναπαριστούμε μια φάση γράφοντας τα μη-κενά γράμματα της ταινίας παρεμβάλλοντας την κατάσταση του αυτομάτου αριστερά από το σημείο που βρίσκεται η κεφαλή.
  - Π.χ. Η φάση $1011q01111$ αναπαριστά την κατάσταση όπου
    - Το αυτόματο βρίσκεται στην κατάσταση q
    - Η ταινία έχει περιεχόμενο 101101111
    - Και η κεφαλή βρίσκεται στο δεύτερο 0

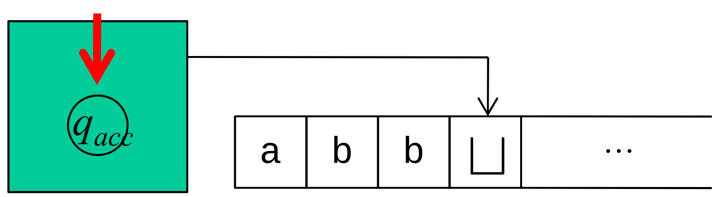
Παράδειγμα Φάσης

Φάση



$ab q_1 a$





$abb q_{acc}$

Φάσεις

- Μια φάση $Φ_1$ *αποδίδει* μια φάση $Φ_2$ αν μπορούμε να μεταβούμε από την $Φ_1$ στη $Φ_2$ με ένα μόνο βήμα
  - η $ua\, q_i\, bv$ αποδίδει την $u\, q_j\, acv$ αν υπάρχει μετάβαση $\delta(q_i, b) = (q_j, c, A)$

- Εναρκτήρια φάση για είσοδο w: q_0w

- Αποδεκτική φάση: Η μηχανή βρίσκεται στην κατάσταση αποδοχής

- Απορριπτική φάση: Η μηχανή βρίσκεται στην κατάσταση απόρριψης

- Τερματικές φάσεις: αποδεκτική και απορριπτική

Γλώσσα μιας Μηχανής Turing

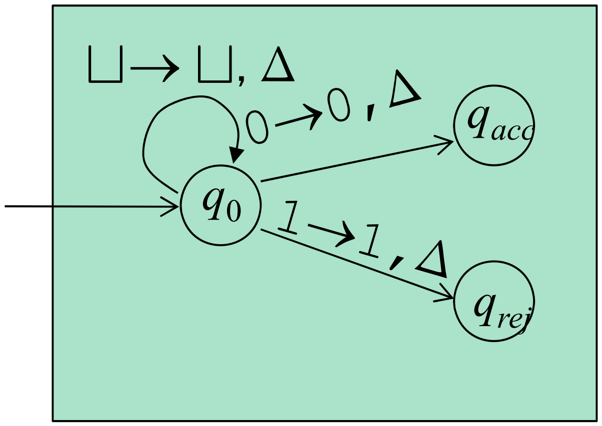
- Μια TM M αποδέχεται τη λέξη x αν υπάρχει μια ακολουθία φάσεων Φ_0, Φ_1, ..., Φ_k όπου

  Φ_0 αρχική Φ_i αποδίδει Φ_{i+1} Φ_k φάση αποδοχής

- L(M) η γλώσσα της M (M αναγνωρίζει την L):
  - L = σύνολο των λέξεων που αποδέχεται η M

Εγκλωβισμός

- Οι Μηχανές Turing δυνατόν να εμφανίσουν ένα ασυνήθιστο φαινόμενο

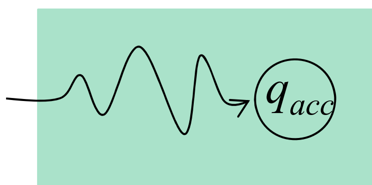


$\Sigma = \{0, 1\}$

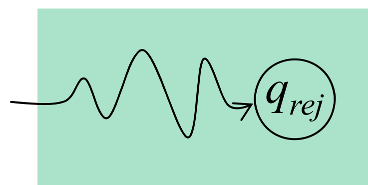
Είσοδος: ε

Η μηχανή δεν τερματίζει ποτέ

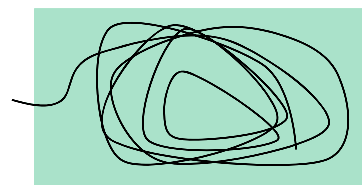
- Για οποιαδήποτε είσοδο υπάρχουν τρεις περιπτώσεις:



αποδοχή



απόρριψη



εγκλωβισμός

Αναγνωρίσιμες και Διαγνώσιμες Γλώσσες

<u>ΟΡΙΣΜΟΣ</u>

Μια γλώσσα λέγεται κατά Turing *αναγνωρίσιμη* (ή αναδρομικά απαριθμίσιμη) αν υπάρχει μηχανή Turing που την *αναγνωρίζει*, δηλαδή, που αποδέχεται ακριβώς τις λέξεις που ανήκουν στη γλώσσα.

<u>ΟΡΙΣΜΟΣ</u>

Μια γλώσσα λέγεται κατά Turing *διαγνώσιμη* (ή αναδρομική) αν υπάρχει μηχανή Turing που την *διαγιγνώσκει*, δηλαδή, που τερματίζει και αποδέχεται όλες τις λέξεις που ανήκουν στη γλώσσα και απορρίπτει όλες τις υπόλοιπες.

Παράδειγμα 1

- Να δώσετε μηχανή Turing που να αναγνωρίζει τη γλώσσα

$$L_1 = \{w\#w \mid w \in \{a, b\}^*\}$$

- Περιγραφή Μηχανής:

1 Μέχρι να συναντήσεις το σύμβολο #

2 Διάβασε το επόμενο σύμβολο — $\texttt{x\underline{b}baa\#xbbaa}$

3 Γράψε x στη θέση του — $\texttt{x\underline{x}baa\#xbbaa}$

4 Μετακινήσου δεξιά πέραν του # και όλων των x — $\texttt{xxbaa\#x\underline{b}baa}$

5 Αν το σύμβολο διαφέρει από αυτό που διαβάστηκε, απόρριψε τη λέξη

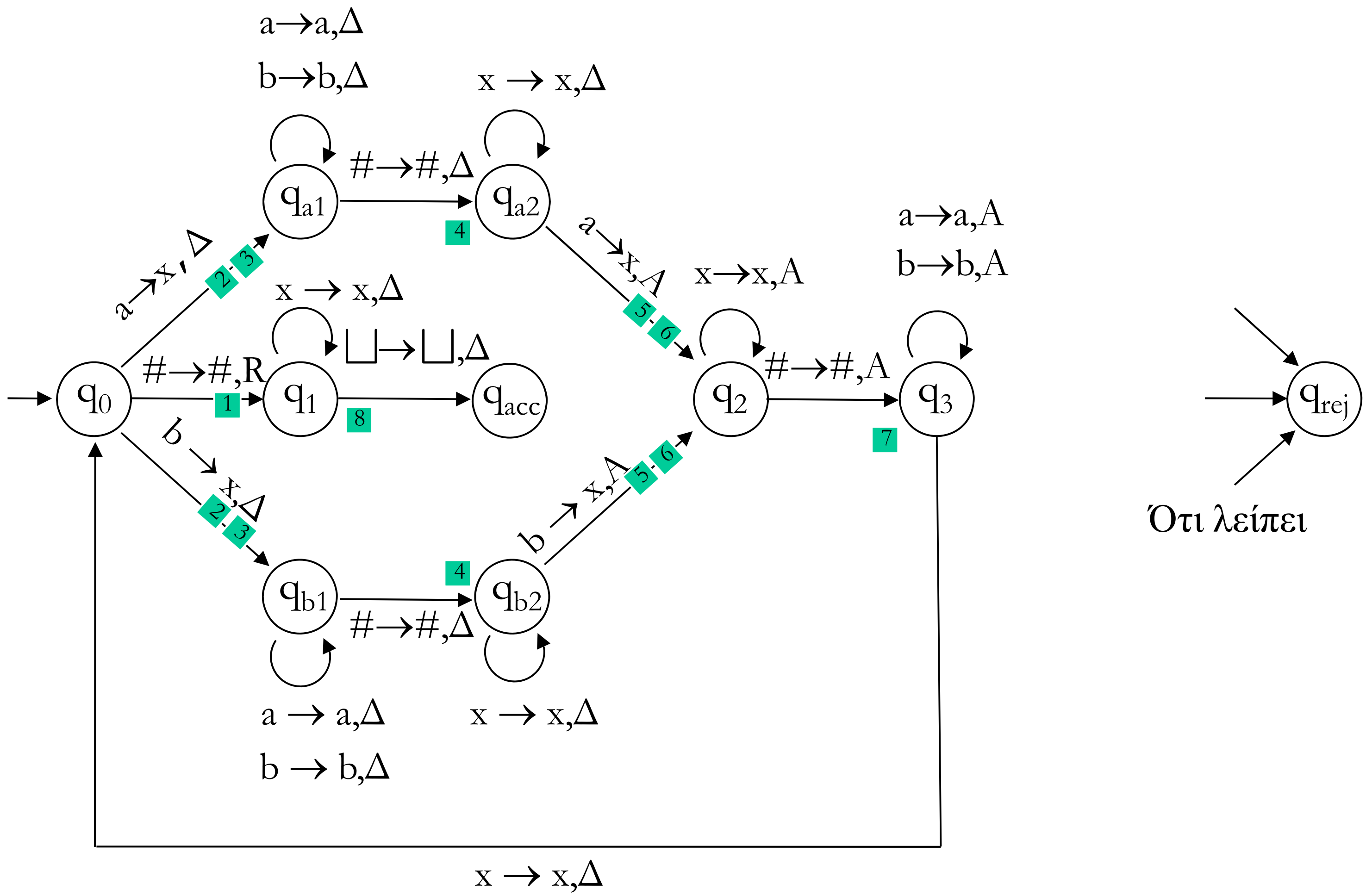
Διαφορετικά

6 Γράψε x — $\texttt{xxbaa\#x\underline{x}baa}$

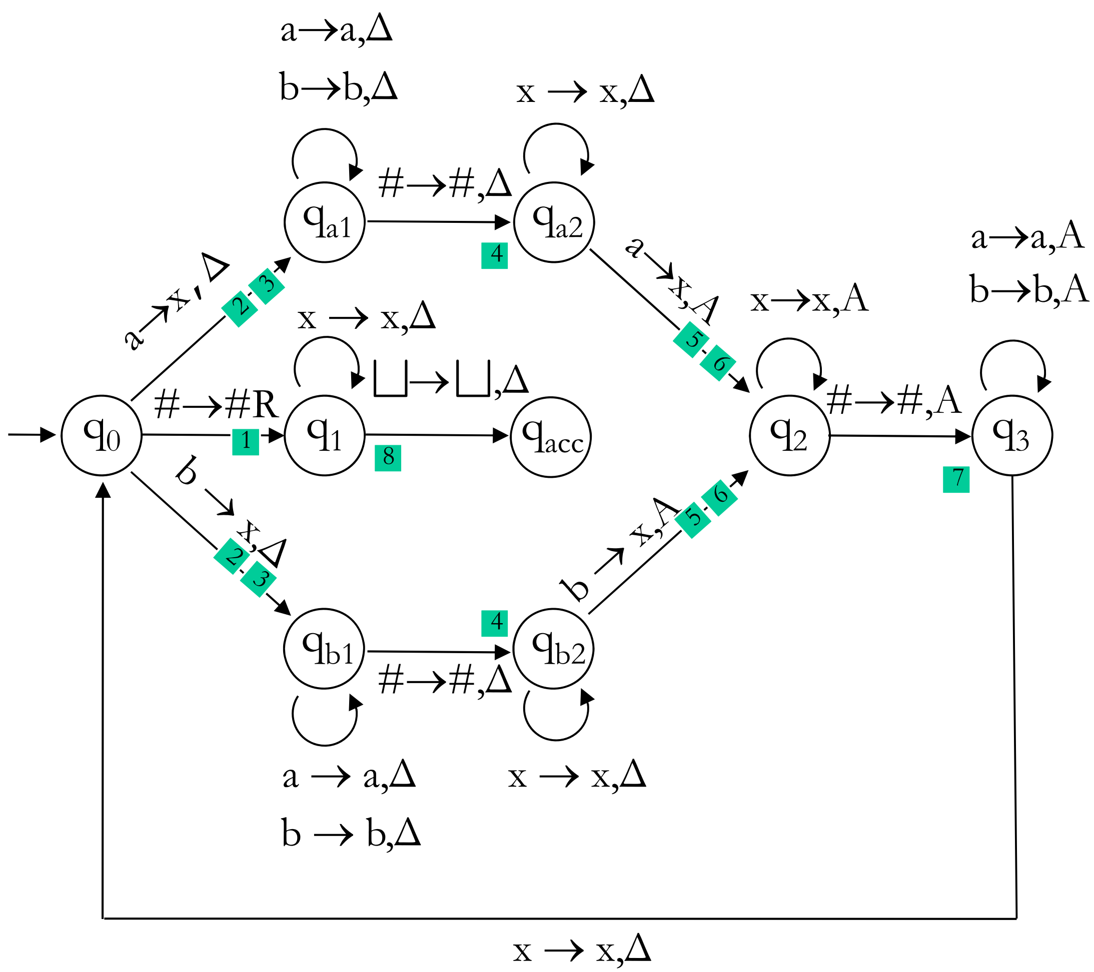
7 Μετακινήσου αριστερά πέραν του # και σταμάτα δεξιά από το τελευταίο x — $\texttt{xx\underline{b}aa\#xxbaa}$

8 Αν βλέπεις μόνο x να ακολουθούνται από ⊔, αποδέξου τη λέξη

Παράδειγμα 1 (συν.)



Παράδειγμα 1 (συν.)



Είσοδος: aab#aab

Φάσεις: $q_0$aab#aab

xq_{a1}ab#aab

xaq_{a1}b#aab

xabq_{a1}#aab

xab#q_{a2}aab

xabq_2#xab

xaq_3b#xab

x$q_3$ab#xab

$q_3$xab#xab

x$q_0$ab#xab

…

Παράδειγμα 2

- Να δώσετε μηχανή Turing που να αναγνωρίζει τη γλώσσα

  $L_2 = \{w \mid w$ έχει το ίδιο πλήθος από 0 και 1$\}$

Βασική Ιδέα TM:

Για κάθε 0:

Διέγραψε το 0

Εντόπισε ένα 1

Αν δεν βρεις απόρριψε τη λέξη

Διαφορετικά διέγραψε το 1

Αν έχουν μείνει 1 απόρριψε τη λέξη, διαφορετικά αποδέξου τη λέξη.

```
                001001

001001   x0x001   xxx00x
x01001   xxx001   xxxx0x

x01001   xxx001

                  Απόρριψη!

x0x001   xxx00x
```

Παράδειγμα 2 (συν.)

Υλοποίηση της Ιδέας: 001001

0. Αν η είσοδος είναι το ε αποδέξου.
1. Επανάλαβε τα εξής:
2. Προχώρησε δεξιά μέχρι να βρεις το πρώτο 0 001001
   Αν δεν βρεις προχώρησε στο Βήμα 7
3. Αντικατάστησε το 0 με x x01001
4. Επέστρεψε στην αρχή της ταινίας. Πως;; x01001
5. Προχώρησε δεξιά μέχρι να βρεις το πρώτο 1 x01001
   Αν δεν βρεις απόρριψε τη λέξη
   Διαφορετικά αντικατάστησε το 1 με x x0x001
6. Επέστρεψε στην αρχή της ταινίας και επανάλαβε από το Βήμα 2 x0x001
7. Αν έχουν μείνει 1 απόρριψε τη λέξη,
   διαφορετικά αποδέξου τη λέξη.

Παράδειγμα 2 (συν.)

Λεπτομέρειες Υλοποίησης:

`001001`

Για τη μετακίνηση της κεφαλής στην πρώτη θέση της ταινίας

Σημαδεύουμε το πρώτο σύμβολο της ταινίας
Και διατηρούμε το σημάδι αυτό ακόμη και όταν
το σύμβολο αυτό «διαγραφεί» (αντικατασταθεί από το x)

$\dot{0}01001$

$\Sigma = \{0,1\}$ $\Gamma = \{0,1,x,\dot{0},\dot{1},\dot{x}\}$

Παράδειγμα 3

- Να δώσετε μηχανή Turing που να αναγνωρίζει τη γλώσσα

$$L_3 = \{a^i b^j c^k \mid i \times j = k \text{ και } i, j, k > 0\}$$

Βασική Ιδέα TM:

aabbcccc

Για κάθε a:

Σημάδεψε όλα τα b και τον ίδιο αριθμό από c

Επανάφερε όλα τα b αλλά όχι τα c

Διέγραψε το a

Αν όλα τα a και τα c έχουν διαγραφεί, αποδέξου τη λέξη.

aabbcccc	~~a~~abb~~cc~~cc
aa~~bbcc~~cc	~~a~~a~~bbcccc~~
aabb~~cc~~cc	~~a~~abb~~cccc~~
~~a~~abb~~cc~~cc	~~aa~~bb~~cccc~~

$\Sigma = \{a, b, c\}$ $\Gamma = \{a, b, c, \cancel{a}, \cancel{b}, \cancel{c}, \sqcup\}$

Παράδειγμα 3 (συν.)

Υλοποίηση της ιδέας:

Ξεκινώντας διαβάζουμε τη λέξη μας για να ελέγξουμε ότι έχει τη ζητούμενη μορφή $aa^*bb^*cc^*$

Επιστρέφουμε στην αρχή της ταινίας. Πως;;

Για κάθε a:

Σημάδεψε όλα τα b και τον ίδιο αριθμό από c. Πως;;

Επανάφερε όλα τα b αλλά όχι τα c

Διέγραψε το a

Αν όλα τα a και τα c έχουν διαγραφεί, αποδέξου τη λέξη.

Παράδειγμα 3 (συν.)

Λεπτομέρειες Υλοποίησης:

aabbcccc

Για τη μετακίνηση της κεφαλής στην πρώτη θέση της ταινίας

Σημαδεύουμε το πρώτο σύμβολο της ταινίας

ȧabbcccc

Για την διαγραφή όλων των b και του ίδιου αριθμό από c.

ȧaqbbcccc

Αντικατάστησε το πρώτο b με ~~b~~

ȧa~~b~~qbcccc

Εντόπισε το πρώτο c

ȧa~~b~~bqcccc

Αντικατέστησε το c με ~~c~~

ȧa~~b~~qb~~c~~ccc

Μετακινήσου αριστερά στην πρώτη θέση μετά από τα ~~b~~

ȧa~~bb~~q~~c~~ccc

Αν υπάρχουν και άλλα b επανέλαβε τη διαδικασία

ȧa~~bbc~~qccc

ȧa~~bb~~q~~cc~~cc

Σ = {a, b, c}   Γ = {a, b, c, ~~a~~, ~~b~~, ~~c~~, ȧ, ~~ȧ~~, $\sqcup$}

Παράδειγμα 4

- Να δώσετε μηχανή Turing που να αναγνωρίζει τη γλώσσα

  $L_4 = \{\#x_1\#x_2\ldots\#x_l \mid x_i \in \{0, 1\}^* \text{ και } x_i \neq x_j \text{ για κάθε } i \neq j\}$

Βασική Ιδέα της TM `#01#0011#1`

Για κάθε είσοδο w,

Για κάθε ζεύγος τμημάτων x_i και x_j στο w,

Συγκρίνουμε τα τμήματα x_i και x_j

Αν είναι τα ίδια, απορρίπτουμε.

Διαφορετικά αποδεχόμαστε.

Παράδειγμα 4 (συν.)

Υλοποίηση της Ιδέας: #01#0011#1

0. Αν η είσοδος είναι το ε ή το #, αποδέξου.
1. Σημάδεψε το πρώτο #, δηλαδή αντικατάστησε το # με $\dot{\#}$, και μετακινήσου δεξιά — $\dot{\#}$01#0011#1
2. Σημάδεψε το επόμενο μη-σημαδεμένο #. Αν δεν υπάρχουν άλλα τέτοια # τότε αποδέξου τη λέξη. — $\dot{\#}$01$\dot{\#}$0011#1
3. Σύγκρινε τις συμβολοσειρές δεξιά από τα $\dot{\#}$. Αν είναι οι ίδιες απέρριψε. — $\dot{\#}$01$\dot{\#}$0011#1
4. Σημάδεψε το επόμενο # δεξιά από το δεξιό $\dot{\#}$ — $\dot{\#}$01#0011$\dot{\#}$1

   Αν δεν υπάρχει άλλο # τότε μετακίνησε το πρώτο $\dot{\#}$ δεξιά στο επόμενο # και το δεύτερο $\dot{\#}$ στο αμέσως επόμενο #. Αν δεν είναι εφικτό αποδέξου. — $\dot{\#}$01#0011$\dot{\#}$1
5. Επέστρεψε στο Βήμα 3

Παράδειγμα 4 (συν.)

Συνέχεια Παραδείγματος:

3. Σύγκρινε τις συμβολοσειρές δεξιά από τα #. Αν είναι οι ίδιες απέρριψε.

   $\dot{\#}\texttt{01\#0011}\dot{\#}\texttt{1}$

4. Σημάδεψε το επόμενο # δεξιά από το δεξιό #

   $\texttt{\#01}\dot{\#}\texttt{0011}\dot{\#}\texttt{1}$

   Αν δεν υπάρχει άλλο # τότε μετακίνησε το πρώτο $\dot{\#}$ δεξιά στο επόμενο # και το δεύτερο $\dot{\#}$ στο αμέσως επόμενο #.
   Αν δεν είναι εφικτό αποδέξου.

Αποδοχή!

Πως παρουσιάζουμε τις Μηχανές Turing;

- Σε αντίθεση με τα DFA, NFA, PDA, σπάνια παρουσιάζουμε πλήρη συστήματα μεταβάσεων για TM.
- Πραγματική χρήση μηχανών Turing
  - Ακριβές μοντέλο για τον ορισμό αλγορίθμων
    - Δυνατότητα να αναγνωρίζουμε αν ένα πρόβλημα είναι επιλύσιμο
- Πως περιγράφουμε ένα αλγόριθμο;
  - Τυπική Περιγραφή: Αναλυτική παράθεση των καταστάσεων της μηχανής, των μεταβάσεων, κτλ.
  - Λεπτομερής Περιγραφή: Τρόπος που κινεί η μηχανή την κεφαλή της και πως αποθηκεύει στην ταινία της δεδομένα
  - Περιγραφή Υψηλού Επιπέδου: Περιγραφή αλγορίθμου χωρίς να αναφερθούμε στον τρόπο διαχείρισης της ταινίας ή της κεφαλής.
- Συνήθως παρουσιάζουμε μια TM μέσω μιας περιγραφής σε ψηλό επίπεδο.

Περιγραφή μηχανών Turing

- Είσοδος μηχανής: πάντοτε μια ΛΕΞΗ
  - Κωδικοποίηση: Μετατροπή οποιουδήποτε αντικειμένου σε λέξη
  - Συμβολισμός κωδικοποίησης αντικειμένου O: <O>
  - Συμβολισμός κωδικοποίησης έκφρασης $O_1, O_2, .., O_k$: $<O_1, O_2, ..., O_k>$

- Περιγραφή αλγορίθμου με μηχανή Turing
  - Είσοδος στην πρώτη γραμμή
    - w: αν η είσοδος είναι μια λέξη
    - <A>: αν η είσοδος είναι κωδικοποίηση του αντικειμένου
  - Περιγραφή σταδίων και βημάτων του υπολογισμού της μηχανής

Παράδειγμα 5

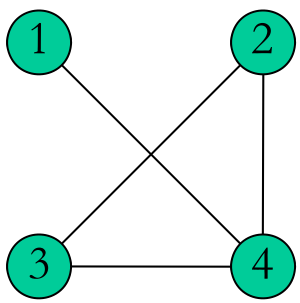
- Να δώσετε μηχανή Turing που να αναγνωρίζει τη γλώσσα

$$L_5 = \{\langle G \rangle : G \text{ ένα συνεκτικό ακατεύθυντο γράφημα}\}$$

Πως κωδικοποιούμε ένα γράφο;;

Ως μια λέξη!

Παράδειγμα:



`(1,2,3,4)((1,4),(2,3),(3,4)(4,2))`

Μέθοδος:

(κορυφές)(ακμές)

καμιά κορυφή δεν πρέπει να επαναλαμβάνεται

οι ακμές είναι ζεύγη: $(\text{κορυφή}_1, \text{κορυφή}_2)$

Παράδειγμα 5 (συν.)

Περιγραφή Υψηλού Επιπέδου της TM

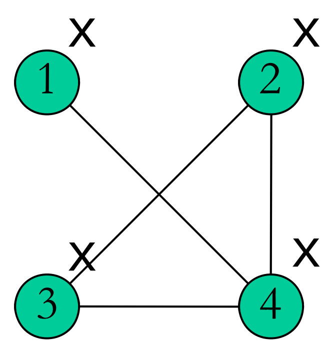
Για είσοδο$\langle G \rangle$:

0. Επιβεβαίωσε ότι η είσοδος $\langle G \rangle$ περιγράφει ένα γράφο
   (όχι επανάληψη κορυφών, οι ακμές αναφέρονται σε κορυφές)

1. Σημάδεψε την πρώτη κορυφή του G

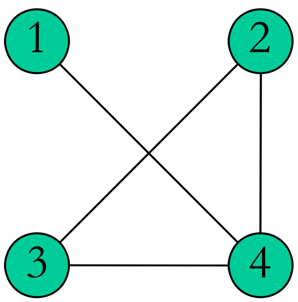
2. Επανάλαβε μέχρι να μην μπορείς να σημαδέψεις καινούριες κορυφές :

   Για κάθε κορυφή, σημάδεψέ την αν συνδέεται
   μέσω ακμής με μία ήδη σεσημασμένη κορυφή



3. Αν όλες οι κορυφές έχουν σημαδευτεί αποδέξου, διαφορετικά απέρριψε.

Παράδειγμα 5 (συν.)



$(\dot{1},2,3,4)((1,4)(2,3)(3,4)(4,2))$

$(\dot{1},2,3,4)((\underline{1},4)(2,3)(3,4)(4,2))$

$(\dot{1},2,3,4)((1,\underline{4})(2,3)(3,4)(4,2))$

$(\dot{1},2,3,\dot{4})((1,4)(2,3)(3,4)(4,2))$

$(\dot{1},2,3,\dot{4})((\underline{1},4)(2,3)(3,4)(4,2))$

$(\dot{1},2,3,\dot{4})((1,\underline{4})(2,3)(3,4)(4,2))$

$(\dot{1},2,3,\dot{4})((1,4)(\underline{2},3)(3,4)(4,2))$

$(\dot{1},2,3,\dot{4})((1,4)(2,\underline{3})(3,4)(4,2))$

$(\dot{1},2,3,\dot{4})((1,4)(2,3)(\underline{3},4)(4,2))$

$(\dot{1},2,\dot{3},\dot{4})((1,4)(2,3)(3,4)(4,2))$

...

Παραλλαγές των Μηχανών Turing

- Διάφοροι εναλλακτικοί ορισμοί των TM
  - Πολυταινιακές
  - Μη ντετερμινιστικές
- Όλες οι παραλλαγές είναι ισοδύναμες με το αυθεντικό μοντέλο
  - Αναγνωρίζουν την ίδια κλάση γλωσσών
  - Το μοντέλο είναι ευσταθές
- Πως αποδεικνύουμε ότι δυο μοντέλα είναι ισοδύναμα;
  - Προσομοιώνουμε το ένα μέσω του άλλου
- Παράδειγμα: Έστω η παραλλαγή των TM όπου

$$\delta: (Q - \{q_{acc}, q_{rej}\}) \times \Gamma \rightarrow Q \times \Gamma \times \{\mathrm{A}, \Delta, \Sigma\}$$

όπου το Σ συμβολίζει τη δυνατότητα η κεφαλή να μείνει στάσιμη κατά την εκτέλεση μιας μετάβασης.
Η παραλλαγή αυτή μπορεί να προσομοιωθεί από την αυθεντική TM και αντίστροφα.

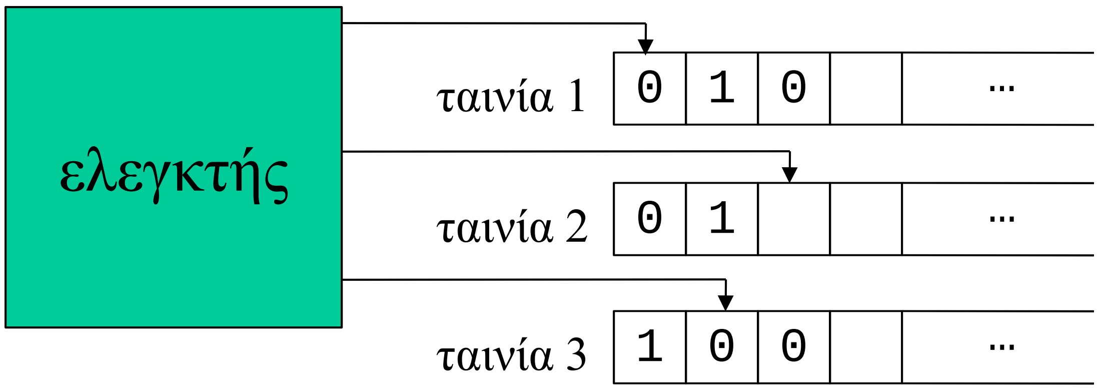
Πολυταινιακές Μηχανές Turing

- Multi-tape Turing machine (MT-TM)
  - Μια μηχανή Turing με πολλές ταινίες
  - Κάθε ταινία έχει τη δική της κεφαλή

- Αρχικοποίηση ταινιών
  - Πρώτη ταινία περιέχει την λέξη εισόδου
  - Οι υπόλοιπες είναι κενές

- Οι πολλαπλές ταινίες είναι βοηθητικές
  - Για παράδειγμα, μπορούν να χρησιμεύσουν για προσωρινή αποθήκευση

Σχηματική Αναπαράσταση



- Κάθε μετάβαση δυνατόν να εξαρτάται από τις τιμές που δείχνονται από τα βέλη πολλών ή και όλων των ταινιών.
- Τα διάφορα βέλη κινούνται ανεξάρτητα το ένα από το άλλο

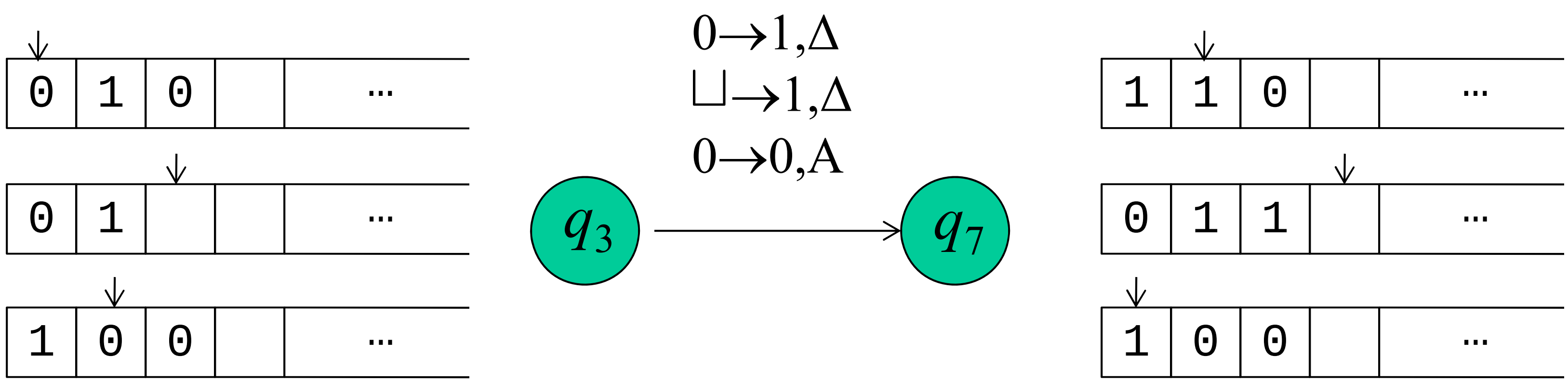
Συνάρτηση μεταβάσεων

- Συνάρτηση μεταβάσεων
  - Επιτρέπει την εγγραφή, ανάγνωση, και μετακίνηση της κεφαλής σε κάποιες ή όλες τις ταινίες

$$\delta: Q \times \Gamma^k \rightarrow Q \times \Gamma^k \times \{\mathrm{A},\Delta,\Sigma\}^k$$

$$\delta(q_i,a_1,\ldots,a_k) = (q_j,b_1,\ldots,b_k,\Delta,\ldots\mathrm{A})$$

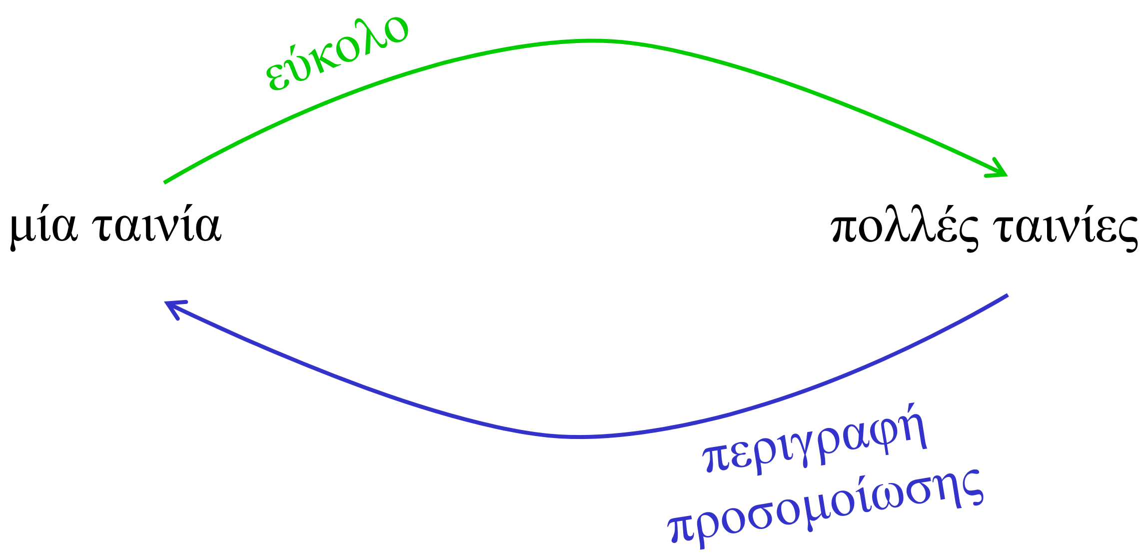
- Παράδειγμα: $\delta(q_3,0,\sqcup,0) = (q_7,1,1,0,\Delta,\Delta,\mathrm{A})$



Ισοδυναμία ΜΤ-ΤΜ με ΤΜ

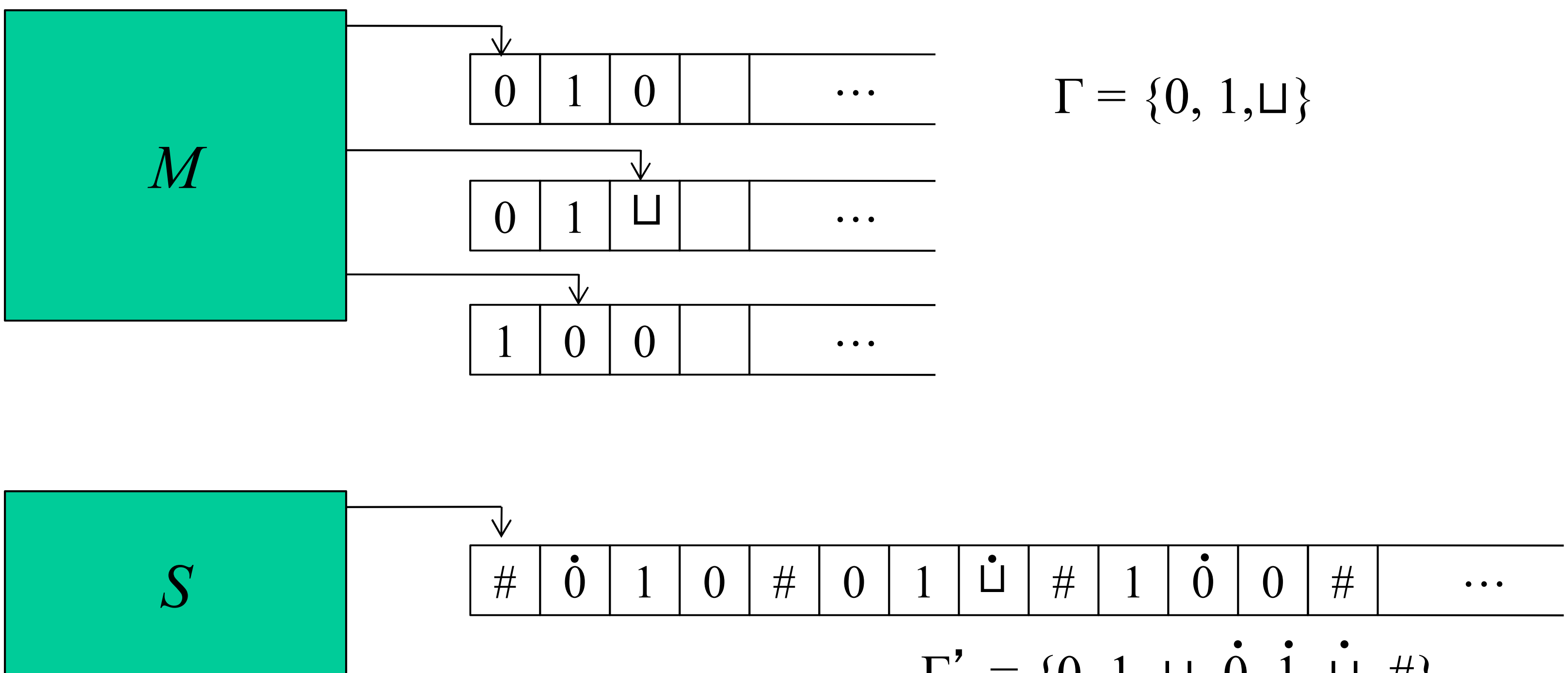
<u>ΘΕΩΡΗΜΑ</u>

Για κάθε πολυταινιακή μηχανή Turing υπάρχει ισοδύναμη μονοταινιακή μηχανή Turing και αντίστροφα.



Προσομοίωση ΜΤ-ΤΜ από ΤΜ

- Αποθηκεύουμε τα περιεχόμενα όλων των ταινιών σε μία ταινία
  - Χρησιμοποιούμε το σύμβολο # ως οριοθέτη
  - Χρησιμοποιούμε την κουκκίδα $\dot{}$ για να δείξουμε τη θέση των κεφαλών

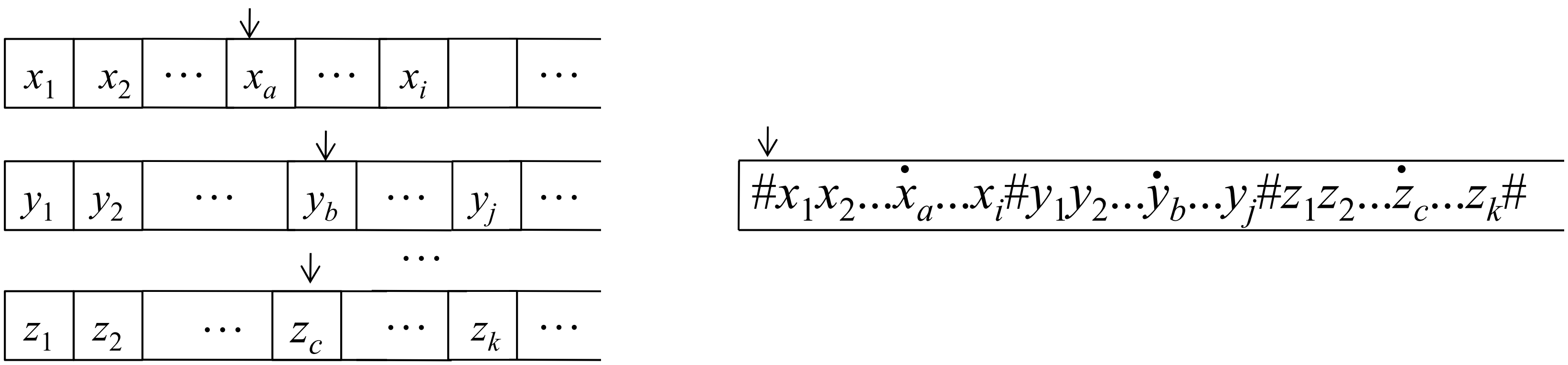


$\Gamma = \{0, 1, \sqcup\}$

$\Gamma' = \{0, 1, \sqcup, \dot{0}, \dot{1}, \dot{\sqcup}, \#\}$

Προσομοίωση ΜΤ-ΤΜ από ΤΜ (συν.)

- Παράδειγμα με τρεις ταινίες



MT – TM → TM

Προσομοίωση ΜΤ-ΤΜ από ΤΜ (συν.)

Αρχικοποίηση

$w_1w_2\ldots w_n$ $\Rightarrow$ $\#\dot{w}_1w_2\ldots w_n\#\dot{\sqcup}\#\dot{\sqcup}\#$

S: Για είσοδο $w_1\ldots w_n$:

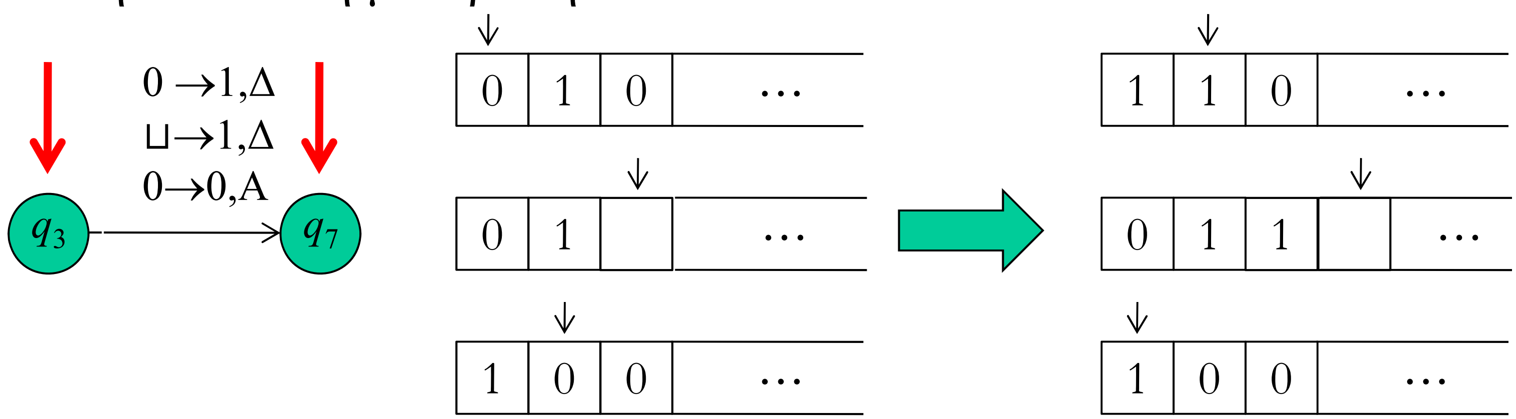
1. Γράφουμε στην ταινία τα περιεχόμενα

$$\#\dot{w}_1w_2\ldots w_n\#\dot{\sqcup}\#\dot{\sqcup}\#$$

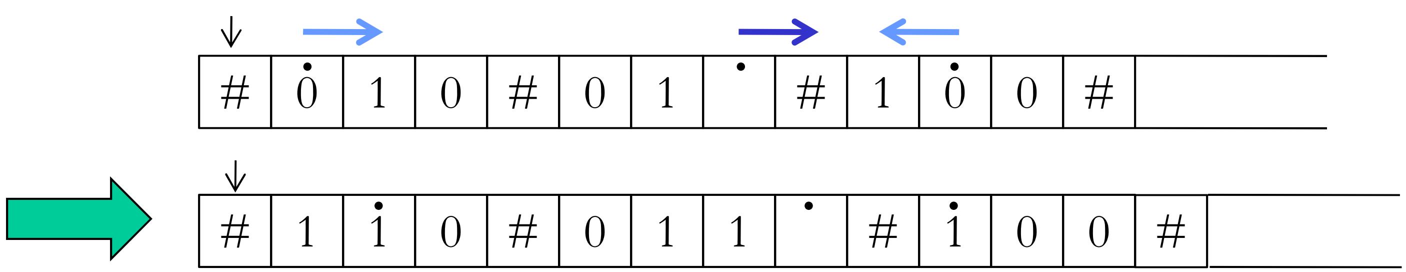
Προσομοίωση ΜΤ-ΤΜ από ΤΜ (συν.)

Προσομοίωση μετάβασης

Έστω η ακόλουθη μετάβαση



Προσομοιώνουμε τη μετάβαση σε κάθε εικονική ταινία.
Αν χρειαστεί μετακινούμε σύμβολα προς τα δεξιά για να διαθέσουμε περισσότερο χώρο σε μια εσωτερική ταινία



Προσομοίωση ΜΤ-ΤΜ από ΤΜ (συν.)

Για είσοδο $w_1 \ldots w_n$:

1. Γράφουμε στην ταινία τα περιεχόμενα

   $\#\dot{w}_1 w_2 \ldots w_n \#\dot{\sqcup}\#\dot{\sqcup}\#$

2. Για να προσομοιώσουμε ένα βήμα τη ΜΤ-ΤΜ:

   Σαρώνουμε την ταινία για να προσδιορίσουμε τις εικονικές κεφαλές
   Με μια δεύτερη σάρωση ενημερώνουμε τις εικονικές ταινίες σύμφωνα με τη συνάρτηση μεταβάσεων την ΜΤ-ΤΜ

3. Αν σε κάποιο σημείο κάποια από τις εικονικές κεφαλές μετακινηθεί προς τα δεξιά και βρεθεί σε κάποιο #, καταχωρούμε σε αυτό το σημείο ένα σύμβολο διαστήματος αφού μετακινήσουμε όλα τα επόμενα σύμβολα μια θέση προς τα δεξιά

Μη Ντετερμινιστικές TM

- Από μια κατάσταση είναι δυνατόν να προκύψουν πολλές εναλλακτικές συνολικές καταστάσεις.
  – Συνάρτηση μεταβάσεων

$$\delta: Q \times \Gamma \rightarrow \mathcal{P}(Q \times \Gamma \times \{\mathrm{A}, \Delta\})$$

<u>ΘΕΩΡΗΜΑ</u>

Για κάθε μη ντετερμινιστική μηχανή Turing υπάρχει ισοδύναμη ντετερμινιστική μηχανή Turing και αντίστροφα.

εύκολο

ντετερμινιστική TM μη ντετερμινιστική TM

περιγραφή προσομοίωσης

παραλείπεται

Αλγόριθμοι και TM

- **Τι είναι ένας αλγόριθμος;**
  - Αλγόριθμος είναι μια πεπερασμένη ακολουθία εντολών οι οποίες αν ακολουθηθούν επιτυγχάνεται κάποιο επιθυμητό αποτέλεσμα.
- Ορισμός του αλγόριθμου
  - Ακριβής ορισμός αλγορίθμου εκφράστηκε τον $20^{ο}$ αιώνα
  - Το $10^{ο}$ πρόβλημα του Hilbert
    - Εύρεση αν ένα πολυώνυμο έχει ακέραιες ρίζες
  - Σήμερα ξέρουμε ότι το πρόβλημα είναι αλγοριθμικά ανεπίλυτο
  - Η αντίληψη της εποχής δεν παρείχε μέσα για να επιδειχθεί ότι δεν υπάρχει αλγόριθμος για κάποιο πρόβλημα
  - Αναγκαία προϋπόθεση: ακριβής ορισμός της έννοιας του *αλγόριθμου*

Τα 23 προβλήματα του Hilbert

David Hilbert
(1862-1943)

- Κορυφαίος Μαθηματικός της εποχής του
- Στη διάλεξή του στο Διεθνές Συνέδριο Μαθηματικών 1900, παρουσίασε 23 προβλήματα τα οποία πρότεινε ως στόχο για τον 20ο αιώνα
- Το 10ο πρόβλημα αναφέρεται σε πολυώνυμα και τις ρίζες τους
  - Πολυώνυμο: $6x^3yz^2 + 3xy^2 - x^3 - 10$
  - (Ακέραια) Ρίζα του πολυωνύμου: $x = 5$, $y = 3$, $z = 0$
    - $6 \cdot 5^3 \cdot 3 \cdot 0^2 + 3 \cdot 5 \cdot 3^2 - 5^3 - 10 = 0 + 135 - 125 - 10 = 0$

Το 10ο πρόβλημα του Hilbert

Να επινοηθεί μια πεπερασμένη μέθοδος που να ελέγχει αν ένα πολυώνυμο έχει ακέραια ρίζα

Δόγμα Church - Turing

- Ακριβής ορισμός της έννοιας του αλγόριθμου
  - Church: λ-calclulus
  - Turing: Μηχανές Turing
  - Ισοδύναμοι ορισμοί
  - Αυτή η συσχέτιση της άτυπης έννοιας του αλγόριθμου με τους ισοδύναμους ακριβείς ορισμούς οδήγησε στο…

Δόγμα Church-Turing (1936)

Αν υπάρχει κάποια μέθοδος (αλγόριθμος) μέσω της οποίας μπορούμε να διεκπεραιώσουμε κάποιο υπολογισμό, τότε ο ίδιος υπολογισμός μπορεί να διεκπεραιωθεί μέσω μιας μηχανής Turing (ή μέσω μιας συνάρτησης διατυπωμένης στο λ-calculus).

- Επομένως: Αλγόριθμος είναι μια μηχανή Turing που τερματίζει για όλες τις εισόδους του προβλήματος
  - Τα προβλήματα που δεν μπορούν να επιλυθούν από μια μηχανή Turing δεν μπορούν να επιλυθούν από οποιοδήποτε μοντέλο υπολογισμού γνωρίζουμε.

Διατύπωση 10^{ου} Προβλήματος Hilbert

- Να επινοηθεί μια Μηχανή Turing που να ελέγχει αν ένα πολυώνυμο έχει ακέραια ρίζα

- Ισοδύναμα, έστω
  - $D = \{p \mid$ το p είναι ένα πολυώνυμο με ακέραια ρίζα$\}$

- Υπάρχει TM που να διαγιγνώσκει την D;

ή

Είναι η D διαγνώσιμη;

- Βασισμένος στην δουλειά των Gödel, Church, Turing, Davis, Putnam, Robinson, το 1970 ο Yuri Matijasievič έδειξε ότι:

Δεν υπάρχει τέτοια μηχανή!

Διαγνωσιμότητα και Αναγνωρισιμότητα

- Θεωρήστε το πιο απλό πρόβλημα D_1:
  D_1 = {*p* | το *p* είναι ένα πολυώνυμο της *x* με ακέραια ρίζα}
- Μια μηχανή που *αναγνωρίζει* το πρόβλημα είναι η εξής:

M_1 = ‘ Για είσοδο ένα πολυώνυμο *p* της μεταβλητής *x*:

- Θέτουμε διαδοχικά την *x* ίση με 0,1,-1,2,-2,3,-3, ... και για κάθε τιμή υπολογίζουμε την αντίστοιχη τιμή του πολυωνύμου. Εάν το *p* γίνει 0, αποδεχόμαστε. ’
- Η M_1 αναγνωρίζει την D_1
  - Περίπτωση 1: Έστω υπάρχει ρίζα *k*. Θα την εντοπίσουμε μετά από το πολύ $2k + 1$ επαναλήψεις
  - Περίπτωση 2: Έστω ότι δεν υπάρχει ρίζα. Τότε η TM δεν θα τερματίσει ποτέ.

Αναγνωρισιμότητα

- Συμπέρασμα: Η D_1 είναι αναγνωρίσιμη

- Παρόμοια : Η D είναι αναγνωρίσιμη!

M = ‘Για κάθε συνδυασμό των δυνατών τιμών των μεταβλητών του πολυωνύμου:
  - Υπολογίζουμε την αντίστοιχη τιμή του πολυωνύμου. Εάν το *p* γίνει 0, αποδεχόμαστε. ’

  - Η M αναγνωρίζει την D

Διαγνωσιμότητα;

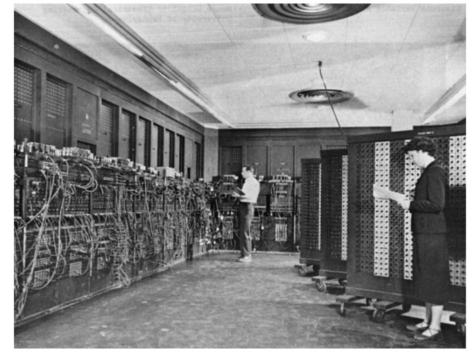
- Για πολυώνυμο μιας μεταβλητής
  - Άνω και κάτω φράγματα για τις ρίζες
  - Εάν k το πλήθος των όρων, c_{max} η απόλυτη τιμή του μέγιστου συντελεστή και c_1 ο συντελεστής του μεγιστοβάθμιου όρου τότε οι ρίζες μεταξύ των φραγμάτων

$$\pm k \frac{c_{max}}{c_1}$$

  - Η M_1 μπορεί να γίνει διαγνώστης αν ελέγχει μόνο τις ακέραιες τιμές μεταξύ των φραγμάτων

- Matijasevič (1970) Ο υπολογισμός φραγμάτων στην περίπτωση των πολλαπλών μεταβλητών είναι αδύνατος
  - Η M δεν μπορεί να διαγνώσει τη D.

Το Δόγμα Church-Turing



Turing Machine

quantum computing

DNA computing