ΕΠΛ 211: Θεωρία Υπολογισμού και Πολυπλοκότητας

Διάλεξη 13: Παραλλαγές Μηχανών Turing και Περιγραφή Αλγορίθμων

Τι θα κάνουμε σήμερα…

- Εισαγωγή
- Πολυταινιακές Μηχανές Turing (3.2.1)
- Μη Ντετερμινιστικές Μηχανές Turing (3.2.2)
- Ορισμός του Αλγόριθμου
- Πρόβλημα Πολυωνύμων (3.3.1)
- Ορολογία για την περιγραφή μηχανών Turing (3.3.2)

Εισαγωγή

- Διάφοροι εναλακτικοί ορισμοί των ΤΜ
  - Πολυταινιακές
  - Μη ντετερμινιστικές
- Όλες οι παραλλαγές είναι **ισοδύναμες** με το αυθεντικό μοντέλο
  - Αναγνωρίζουν την ίδια κλάση γλωσσών
- Πως αποδεικνύουμε ότι δυο μοντέλα είναι ισοδύναμα;
  - Προσομοιώνουμε το ένα μέσω του άλλου

Πολυταινιακές TM…

- Τι είναι;
  - Μια μηχανή Turing με πολλές ταινίες
  - Κάθε ταινία έχει τη δική της κεφαλή

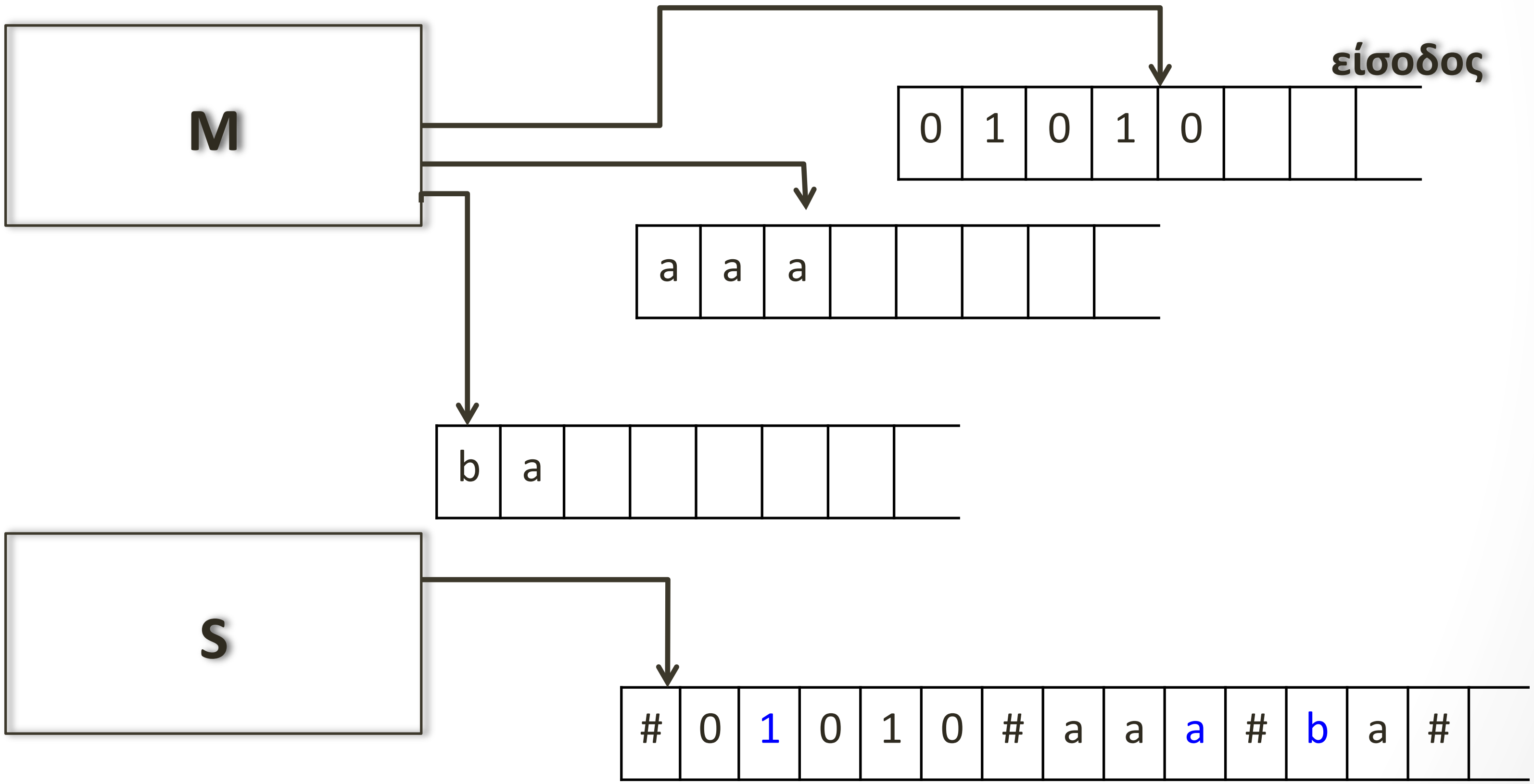
- Αρχικοποίηση ταινιών
  - Πρώτη ταινία περιέχει την λέξη εισόδου
  - Οι υπόλοιπες είναι κενές

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

$$\delta : Q \times \Gamma^k \rightarrow Q \times \Gamma^k \times \{A, \Delta, \Sigma\}^k$$
$$\delta(q_i, a_1, \ldots, a_k) = (q_j, b_1, \ldots, b_k, A, \Delta, \ldots, A)$$

Ισοδυναμία ΜΤ-ΤΜ με ΤΜ

Θεώρημα
Για κάθε πολυταινιακή μηχανή Turing υπάρχει ισοδύναμη μονοταινιακή μηχανή Turing



Ισοδυναμία ΜΤ-ΤΜ με ΤΜ

- Για είσοδο $w=w_1 \dots w_n$
  - Διαμορφώνουμε την ταινία μας να αναπαριστά και τις k ταινίες της M
  - Σαρώνουμε από το πρώτο # (αρχή λέξης) μέχρι το (k+1)-οστό # (τέλος λέξης) για να προσδιορίσουμε τα σύμβολα κάτω από τις κεφαλές
  - Με δεύτερη σάρωση ενημερώνουμε κάθε ταινία με τον τρόπο που υπαγορεύει η συνάρτηση μεταβάσεων

Μηχανή πολλαπλών κεφαλών

- Επιτρέπουμε πολλαπλές κεφαλές και μια ταινία

Θεώρημα
Για κάθε μηχανή Turing πολλαπλών κεφαλών υπάρχει ισοδύναμη μονοταινιακή μηχανή Turing

- Αποδειξη: Με προσομοίωση (αφήνεται για άσκηση)

Μη Ντετερμινιστικές ΤΜ

- Από μια συνολική κατάσταση είναι δυνατόν να προκύψουν πολλές εναλλακτικές συνολικές καταστάσεις.
  - $$\delta : Q \times \Gamma \rightarrow \mathcal{P}(Q \times \Gamma \times \{A, \Delta\})$$

Θεώρημα
Για κάθε μη ντετερμινιστική μηχανή Turing υπάρχει ισοδύναμη μονοταινιακή μηχανή Turing

- Αποδειξη: Με προσομοίωση (αφήνεται για άσκηση)

Αλγόριθμοι και ΤΜ -- Εισαγωγή

- Τι είναι ένας αλγόριθμος;
  - Συλλογή απλών οδηγιών για τη διεκπεραίωση μιας εργασίας.

- Ορισμός του αλγόριθμου
  - Ακριβής ορισμός αλγορίθμου εκφράστηκε τον 20ο αιώνα
  - 1900 Hilbert – Απαρίθμησε 23 μαθηματικά προβλήματα
  - Το 10ο αφορούσε τους αλγόριθμους
    - Εύρεση αν ένα πολυώνυμο έχει ακέραιες ρίζες
  - Αυτό το πρόβλημα ώθησε την ανάγκη για ακριβή ορισμό των αλγορίθμων

Ακέραια Ρίζα Πολυωνύμων

- Τι είναι πολυώνυμο;
  - Έστω το πολυώνυμο $6x^3yz^2 + 3xy^2 - x^3 - 10$
  - Όροι: γινόμενο μεταβλητών και συντελεστή
  - Ρίζα: μια ανάθεση τιμών στις μεταβλητές τέτοιων ώστε το πολυώνυμο να παίρνει τιμή 0
  - Π.χ. x=5, y=3, z=0
- Αν όλες οι ρίζες ενός πολυωνύμου είναι ακέραιοι τότε λέμε ότι το **πολυώνυμο έχει ακέραια ρίζα**
- **Πρόβλημα Hilbert**: Μπορούμε να επινοήσουμε μια διαδικασία (αλγόριθμο) που να μας λέει αν ένα πολυώνυμο έχει ακέραια ρίζα σε πεπερασμένο χρόνο;
  - Σήμερα ξέρουμε ότι **ΔΕΝ ΥΠΑΡΧΕΙ** τέτοια διαδικασία

Δόγμα Church - Turing

- Ορίζει επακριβώς την έννοια του αλγορίθμου
  - Church όρισε τον αλγόριθμο με λογισμό λ
  - Turing όρισε τον αλγόριθμο με τις «μηχανές» του
  - Ισοδύναμοι ορισμοί => Δόγμα Church-Turing
  - Τι μπορεί να υπολογιστεί και τι δεν μπορεί να υπολογιστεί?

- **Δόγμα Church-Turing (1936):**
  - ότι μπορεί να υπολογιστεί (υπάρχει αλγόριθμος) μπορεί να υπολογιστεί από μια μηχανή Turing.
  - Δηλ. 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. (είναι **μη επιλύσιμα**).

Διατύπωση Προβλήματος Hilbert

- **D = {p|το p είναι ένα πολυώνυμο με ακέραια ρίζα}**

- Θεωρήστε ότι «υπάρχει τρόπος» να εισάγουμε ένα πολυώνυμο ως είσοδο σε μια μηχανή Turing
  - Θέλουμε να απαντήσουμε το ερώτημα:
    «Είναι η D διαγνώσιμη;»

Η D είναι αναγνωρίσιμη

- Θεωρείστε το πιο απλό πρόβλημα D_1:
  D_1 = {p|το p είναι ένα πολυώνυμο της x με ακέραια ρίζα}

- M1 = ‘ 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 M1 αναγνωρίζει την D_1: αν δεν υπάρχει ακέραια ρίζα δεν τερματίζει

- M=διέτρεξε όλους τους πιθανούς συνδυασμούς ακεραίων στις μεταβλητές του πολυωνύμου
  - **Η M αναγνωρίζει την D**

Η D είναι διαγνώσιμη;

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

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

  - Η Μ1 μπορεί να γίνει διαγνώστης αν ελέγχει μόνο τις ακέραιες τιμές μεταξύ των φραγμάτων

- Ο υπολογισμός φραγμάτων στην περίπτωση των πολλαπλών μεταβλητών είναι αδύνατος
  - Η Μ **ΔΕΝ ΜΠΟΡΕΙ** να είναι διαγνωστής => Η D **ΔΕΝ ΕΙΝΑΙ ΔΙΑΓΝΩΣΙΜΗ**

Περιγραφή μηχανών Turing

- 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, 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 στο τρόπο διαχείρισης της ταινίας ή της κεφαλής.

Περιγραφή μηχανών Turing

- Είσοδος μιας μηχανής: πάντοτε μια **ΛΕΞΗ**
  - **Κωδικοποίηση:** Μετατροπή οποιουδήποτε αντικειμένου σε λέξη
  - Συμβολισμός κωδικοποίησης αντικειμένου O: <O>
  - Συμβολισμός κωδικοποίησης έκφρασης O1,O2,..,Oκ: <O1,O2,...,Oκ>

- 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 1

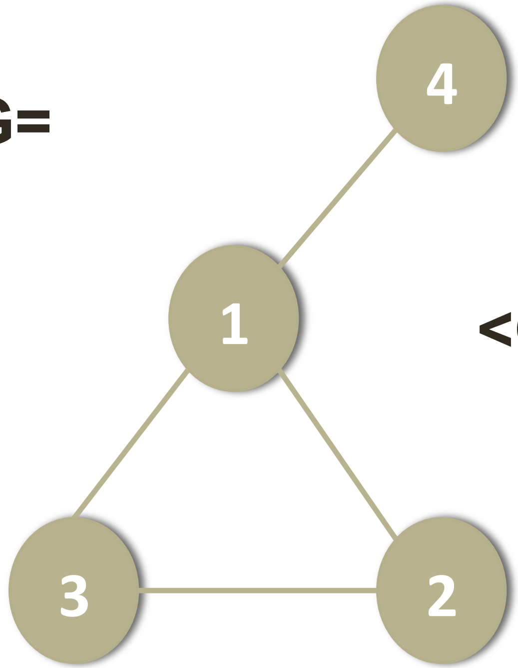
- A={<G> | το G είναι συνδεδεμένο ακατεύθυντο γράφημα}

- M=‘Για είσοδο <G>, την κωδικοποίηση κάποιου ακατεύθυντου γραφήματος G:
  1. Επιλέγουμε και σημαίνουμε τον πρώτο κόμβο
  2. Επαναλαμβάνουμε μέχρι να μην σημαίνεται κανένας:
     1. Για κάθε κόμβο u: ελέγχουμε αν υπάρχει ακμή που να τον συνδέει με κάποιον ήδη σημασμένο κόμβο. Αν ναι τότε σημαίνουμε και τον u.
  3. Ελέγχουμε αν είναι όλοι οι κόμβοι σημασμένοι. Αν ναι αποδεχόμαστε αλλιώς απορρίπτουμε

Παράδειγμα 1

- Κωδικοποίηση G:

G=



<G>=(1,2,3,4)((1,2),(2,3),(3,1),(1,4))

Ερωτήσεις;