Θεωρία Υπολογισμού και Πολυπλοκότητα

Κλάσεις P, NP
NP-πληρότητα

15 Απριλίου 2008

Δρ. Παπαδοπούλου Βίκη

Υπολογίσιμα και Εφικτά Υπολογίσιμα Προβλήματα

Είδαμε ότι

1. Οτιδήποτε μπορούμε να περιγράψουμε με ένα αλγόριθμο μπορεί να υπολογιστεί με μια μηχανή Turing
2. υπάρχουν προβλήματα που δεν μπορούν να επιλυθούν (π.χ. πρόβλημα τερματισμού
3. υπάρχουν προβλήματα που μπορούν να επιλυθούν

- Επιλύσιμα προβλήματα
  - Πόσους υπολογιστικούς πόρους απαιτούν
    - χρόνο

Παράδειγμα

- Το πρόβλημα του Πλανώδιου πωλητή: Για ένα σύνολο από πόλεις και κόστη διαδρομών από πόλη σε πόλη, βρες μια διαδρομή
  - που να περνά από κάθε κόμβο του δικτύου ακριβώς μια φορά και
  - να ελαχιστοποιεί την συνολικό κόστος της διαδρομής που θα ακολουθηθεί.
- Μπορεί να λυθεί σε (n-1)! χρόνο.
- Είναι μη ρεαλιστικός!

- Συμπέρασμα. Ορισμένα επιλύσιμα προβλήματα είναι υπολογιστικά δύσκολα

Ασυμπτωματικός Ρυθμός Ανάπτυξης

- Μια συνάρτηση f(n) λέμε ότι αυξάνεται πιο αργά από την συνάρτηση g(n) και συμβολίζεται $f(n) \prec g(n)$ εάν

$$\lim_{n\to\infty} \frac{f(n)}{g(n)} = 0$$

- Γράφουμε $f(n)= o\,(g(n))$ εάν $f(n) \prec g(n)$
- $f(n)= O\,(g(n))$ εάν για όλα τα $n \geq 0$, $f(n) \leq K\, g(n)$.
- Γνωστές συναρτήσεις: (i=σταθερό)

1. Πολυλογαριθμική Σειρά: $\{ (\log(n))^i \mid i = 1,2,\ldots\}$
2. Πολυωνιμική Σειρά: $\{n^i \mid i = 1, 2,\ldots\}$
3. Υποεκθετική Σειρά: $\{ n^{\log n^i} \mid i = 1,2,\ldots\}$
4. Εκθετική Σειρά: $\{2^{i\,n} \mid i = 1,2,\ldots\}$
5. Υπερεκθετική Σειρά: $\{2^{n^i} \mid i = 1,2,\ldots\}$

Ιδιότητες

- Σε κάθε σειρά, εάν i < j τότε η i-οστή συνάρτηση αυξάνεται πιο αργά από την j-οστή.
  - Π.χ. , $n^{\log(n)^3} \prec n^{\log(n)^4}$.

- Για οποιεσδήποτε δύο σειρές, κάθε συνάρτηση στην προγενέστερη σειρά αυξάνεται πιο αργά από οποιαδήποτε συνάρτηση στην μεταγενέστερη σειρά.

  (εκτός απο την πρώτη συνάρτηση της τελευταίας σειράς).
  - Π.χ. , $(\log(n)^{64} \prec n^{10} \prec 2^{3n} \prec 2^{n^3}$.

Πολυπλοκότητα Χώρου και Χρόνου

- Υπολογιστική Πολυπλοκότητα μιας μηχανής Turing
  - Χρόνος

ΧΡΟΝΟΣ.

- Χρόνος εκτέλεσης μιας μηχανής Turing με είσοδο w, $Time_M(x)$:
  - Αριθμός βημάτων της μηχανής έως ότου η μηχανή τερματίσει.
  - Αν δεν τερματίσει ο χρόνος είναι άπειρος
- Φράγμα χρόνου της M, $t(n)$ $|x| \leq n$, (υποθέτουμε $t(n) \geq n+1$)
  - $Time_M(x) \leq \max\{n+1, t(n)\}$

Πολυπλοκότητα Χρόνου

- Ποιο μοντέλο μηχανής Turing θεωρούμε?

- Θεώρημα. Για οποιαδήποτε Ντετερμινιστική Μηχανή Turing (NMT) με πολλές ταινίες M, υπάρχει μια διπλής κατεύθυνσης, μονής ταινίας NMT M_1 που υπολογίζει την ίδια συνάρτηση τέτοια ώστε για όλες τις εισόδους x, έχουμε $\text{Time}_{M1}(x) \leq c \cdot (\text{Time}_M(x))^2$ για κάποια σταθερά $c > 0$.

Πολυπλοκότητα Χρόνου

- Θεώρημα 1. Για οποιαδήποτε NMT M μονής ταινίας εργασίας και οποιαδήποτε είσοδο x, $\text{Space}_M(x) \leq \text{Time}_M(x) + 1$.
- Συμπέρασμα. Ο χώρος είναι **πιο** ισχυρός από τον χρόνο, μπορούμε να επαναχρησιμοποιήσουμε τον χώρο ενώ τον χρόνο όχι.
- Ορισμός. Μια γλώσσα έχει **χρονική πολυπλοκότητα** f(n) εάν αποφασίζεται από μια NMT με πολλές ταινίες με φράγμα χρόνου f(n).

- Για οποιαδήποτε συνάρτηση f(n), ορίζουμε κλάσεις πολυπλοκότητας:
  - **DTIME(f(n))** = {L | L έχει χρονική πολυπλοκότητα f(n) }

Κλάσεις Πολυπλοκότητας

$$P = \bigcup_{c>0} DTIME(n^c)$$

Κλάση P (Polynomial Time)

- Η NMT (ντετερμινιστική μηχανή Turing) M είναι μια **NMT πολυωνυμικού χρόνου** εάν η M έχει φράγμα χρόνου t(n) για κάποια πολυωνυμική συνάρτηση $t(n) = n^c$.
- Μια γλώσσα L είναι **πολυνωνυμικά αποφασίσιμη** αν υπάρχει μηχανή NMT πολυωνυμικού χρόνου M που την αποφασίζει.
- Συμπερασματικά: Η κλάση P περιέχει προβλήματα που μπορούν να αποφασιστούν σε πολυωνυμικό χρόνο από μια Ντετερμινιστική μηχανή Turing.

Κλάση P

- Θεώρημα. Η κλάση P είναι κλειστή ως προς την πράξη της σύνθεσης, σύμπτυξης, ένωση, τομή, και συμπληρώματος, για γλώσσες.

Απόδειξη. (ένωση $A \cup B$)

- Έστω M_1 η ΝΜΤ που αποφασίζει για την γλώσσα A με φράγμα χρόνου p_1
- Έστω M_2 η ΝΜΤ που αποφασίζει για την γλώσσα την B με φράγμα χρόνου p_2

p_1, p_2: πολυωνυμικές συναρτήσεις

- Η μηχανή M' δύο ταινιών, για είσοδο w, αντιγράφει την w στην $2^η$ ταινία.
- Πρώτα προσομοιώνει την M_1 με είσοδο w στην 1η ταινία
- Μετά προσομοιώνει την M_2 με είσοδο w στην 2η ταινία
- Εάν είτε η M_1 είτε η M_2 αποδέχεται τη λέξη, η M' αποδέχεται τη λέξη
- Αλλιώς την απορρίπτει.

$\Rightarrow$ η M' αποδέχεται την w εάν και μόνο αν είτε $w \in A$ ή $w \in B$.

Θήκη Kleene

- Έστω μια γλώσσα $A \in P$. Δείχνουμε ότι τότε $A^* \in P$.
- w in A^* εάν και μόνο αν υπάρχει διαμερισμός της w ($|w|=n$) σε υποσυμβολισειρές $x = x_1 \cdots x_m$, $1 \leq m \leq n$, τέτοια ώστε $x_i \in A$, για κάθε $i = 1, \cdots m$.
- Θεώρημα. Η κλάση P είναι κλειστή ως προς την πράξη της θήκης Kleene για γλώσσες.

Απόδειξη. (Φροντιστήριο)

- Γενικά Αποδεκτή Άποψη. Η κλάση P αντιπροσωπεύει την κλάση των **εφικτών** επιλύσιμων γλωσσών, ή απλά **εφικτών** προβλημάτων.

Κλάση P

Προβλήματα στην κλάση P

- Μια κανονική γλώσσα L μπορεί να αποφασιστεί σε πολυωνυμικό χρόνο:
  - **Πώς?**: από το διάγραμμα καταστάσεων ενός αυτόματου που δέχεται τη γλώσσα προκύπτει ένας αλγόριθμος που ελέγχει εάν $w \in L$, σε χρόνο $|w|$.

  $\Rightarrow$ Σύνολο Κανονικών γλωσσών $\in$ P
- Μια γλώσσα χωρίς συμφραζόμενα L μπορεί να αποφασιστεί σε πολυωνυμικό χρόνο:
  - **Πώς?**: από το διάγραμμα καταστάσεων του αυτόματου στοίβας το οποίο δέχεται τη γλώσσα προκύπτει ένας αλγόριθμος που ελέγχει εάν $w \in L$, σε χρόνο $|w|$.(σύνθετος αλγόριθμος, επειδή τα αυτόματα στοίβας είναι μη ντετερμινιστικά)

  $\Rightarrow$ Σύνολο γλωσσών χωρίς συμφραζόμενα $\in$ P

Προβλήματα

- **Πρόβλημα Συνεκτικότητας.** Δεδομένου ενός κατευθυνόμενου γραφήματος G, υπάρχει ένα μονοπάτι μεταξύ δύο κόμβων v_i, v_j , όπου $v_{i,}$ $v_j \in V$?
- Το πρόβλημα ανήκει στο P?
- Η κλάση P αφορά γλώσσες !!

⇒ **Μετατρέπουμε το πρόβλημα έτσι ώστε να αφορά γλώσσες :**

- **Το Πρόβλημα Συνεκτικότητας σε μορφή γλώσσας:**
  - Γλώσσα **R = {κ(G) b(i) b(j) : υπάρχει μονοπάτι από τον v_i, v_j στο G(V, E) }**
    - κ(G) = κωδικοποίηση G : δίνεται με ένα πίνακα γειτνίασης a(), a(i,j) =1 εάν και μόνο εάν υπάρχει ακμή μεταξύ των κόμβων v_i και v_j, όπου $i, j \leq n$.
    - b(i) , b(j) = κωδικοποίηση των κόμβων v_i και v_j

Προβλήματα και Γλώσσες

- Έτσι, αναφερόμαστε σε προβλήματα και εννοούμε ότι τις αντίστοιχες γλώσσες τους.
- Το πρόβλημα συνεκτικότητας ανήκει P
  - λύνεται σε χρόνο $O(n^2)$ (n = πλήθος κόμβων του G)
  - Πώς?
    - Μέσω ψαξίματος κατά πλάτος αρχίζοντας από τον κόμβο v_i.

Προβλήματα..

- **Πρόβλημα Κύκλου Euler:** Δεδομένου ενός γραφήματος G, υπάρχει κλειστό μονοπάτι στο G, το οποίο χρησιμοποιεί κάθε ακμή του G ακριβώς μια φορά?
- **Σε μορφή γλώσσας:**
  - **L = {κ(G) : το G είναι ένα γράφημα Euler (=έχει ένα κύκλο Euler)}**
  - Η L ανήκει στο P:
    - Γιατί υπάρχει ένας χαρακτηρισμός
      - (της μορφής ο G είναι γράφος Euler εάν και μόνο αν ισχύει κάτι)
    - ο οποίος μπορεί να ελεγχθεί σε πολυωνυμικό χρόνο.

Προβλήματα που φαίνεται να ΜΗΝ ανήκουν στην κλάση P

Προβλήματα

- **Πρόβλημα Κύκλου Hamilton:** Δεδομένου ενός γραφήματος G, υπάρχει ένας κύκλος που να περνά από κάθε κορυφή του G ακριβώς μια φορά?
- Ένας αλγόριθμος για το πρόβλημα:
  - εξέτασε όλες τα πιθανά μονοπάτια κόμβων
  - έλεγξε εάν είναι Hamilton κύκλος.
- Απαιτεί εκθετικό χρόνο!
- Δεν είναι γνωστός πολυωνυμικός αλγόριθμος για το πρόβλημα ☹

Προβλήματα Βελτιστοποίησης

- **Το πρόβλημα του Πλανώδιου πωλητή:** Για ένα δεδομένο δίκτυο με n κόμβους και αποστάσεις d_{ij} που συμβολίζει την απόσταση των κόμβων v_i v_j , βρες μια διαδρομή π:
  - που να περνά από κάθε κόμβο του δικτύου ακριβώς μια φορά και
  - ελαχιστοποιεί την συνολική απόσταση που θα διανυθεί.

  Δηλ.

  - Βρες μια μετάθεση π του {1, 2,…n} (δηλ. π(i) είναι η σειρά που περνούμε από την πόλη i στην διαδρομή) που να ελαχιστοποιεί το άθροισμα: $c(\pi) = d_{\pi(1)\pi(2)} + d_{\pi(1)\pi(2)} + \ldots + d_{\pi(1)\pi(2)}$
    - Μας ενδιαφέρει ΟΧΙ ΜΟΝΟ να βρουμε ‘κάτι’ αλλά
    - υπάρχει μια συνάρτηση κόστους c(π) ΚΑΙ επιθυμούμε να ικανοποιείται από το ‘κατι’.
- **Ως γλώσσα:** Δεδομένου ενός δικτύου με n κόμβους και αποστάσεις d_{ij} και ενός ακεραίου B, υπάρχει μια μετάθεση του δικτύου τέτοια ώστε $c(\pi) \leq B$?

Το πρόβλημα του Πλανώδιου πωλητή

- ΠΑΡΑΤΗΡΗΣΗ. Αν μπορούσαμε να λύσουμε το πρόβλημα βελτιστοποίησης, θα μπορούσαμε να λύσουμε και το πρόβλημα ως γλώσσα.
  - δηλ. το πρόβλημα βελτιστοποίησης είναι πιο δύσκολο από την έκδοση του προβλήματος ως γλώσσα
- Δεν είναι γνωστός πολυνωνυμικός αλγόριθμος για το πρόβλημα ☹

Το πρόβλημα Μέγιστου Ανεξάρτητου Συνόλου

- Το πρόβλημα μέγιστου ανεξάρτητου συνόλου: Για ένα δεδομένο γράφημα G(V, E) βρες ένα υποσύνολο $I \subseteq V$ τέτοιο ώστε
  - για κάθε v_i , $v_j \in I$ ισχύει $(v_i, v_j) \notin E$
  - το $|I|$ είναι το μέγιστο δυνατό για το γράφημα G.

Σε μορφή γλώσσας:

- Το πρόβλημα μέγιστου ανεξάρτητου συνόλου: Για ένα δεδομένο γράφημα G(V, E) και ενός ακεραίου K, υπάρχει ένα υποσύνολο $I \subseteq V$ τέτοιο ώστε
  - για κάθε v_i , $v_j \in I$ $(v_i, v_j) \notin V$
  - $|I| \geq K$?

- Δεν είναι γνωστός πολυωνυμικός αλγόριθμος για το πρόβλημα ☹

Ικανοποιησιμότητα Τύπων Bool

Ορισμός. **Τύπος Bool**

- **$X=\{x_1, \cdots, x_n\}$: Δυαδικές Μεταβλητές Bool**
- $\overline{X} = \{\overline{x_1}, \cdots, \overline{x_n}\}$: **αρνήσεις των** $x_1 \cdots, x_n$ ή literals
- **Συνθήκη** $C \subseteq X \cup \overline{X}$ $= (x_i \vee \cdots \vee x_j) = (x_i, \cdots, x_j)$

  $= x_1$ ή … ή x_j (x_i : **στοιχεία** της **C**)
- **Τύπος Bool** σε κανονική διαζευκτική μορφή:
  - **F** $= \{C_1, \cdots, C_m\} = C_1$ και C_2 … και C_m
- Παράδειγμα

  $$X = \{x_1, x_2, x_3\}, \overline{X} = \{\overline{x_1}, \overline{x_2}, \overline{x_3}\}$$

  - $$F = \{(\overline{x_1} \vee \overline{x_2} \vee x_3), (\overline{x_1}), (x_1 \vee \overline{x_2})\}$$

Ικανοποιησιμότητα Τύπων Bool

- **Ορισμός** (συνέχεια). **Απόδοση τιμών αληθείας** (truth assignment) T για τον F είναι μια συνάρτηση $T: X \to \{\top, \bot\}$, όπου
  - $\top$ = αληθές
  - $\bot$ = ψευδές
- Η T ***ικανοποιεί*** τον F εάν:
  - **για κάθε** συνθήκη $C \in F$ υπάρχει τουλάχιστον μια μεταβλητή x_i, τέτοια ώστε είτε
    1. $T(x_i) = \top$ και $x_i \in C$ (x_i : αληθές στοιχείο) είτε
    2. $T(x_i) = \bot$ και $\overline{x_i} \in C$ ($\overline{x_i}$: αληθές στοιχείο)
- Ο F είναι **ικανοποιήσιμος εάν υπάρχει μια απόδοση τιμών αληθείας που τον ικανοποιεί.**

- Παράδειγμα.

$$X = \{x_1, x_2, x_3\}, \overline{X} = \{\overline{x_1}, \overline{x_2}, \overline{x_3}\}$$

$$F = \{(\overline{x_1} \vee \overline{x_2} \vee x_3), (\overline{x_1}), (x_1 \vee \overline{x_2})\}$$

$$T(x_1) = \bot, T(x_2) = \top, T(x_3) = \top$$

- Η T είναι αληθής για την F.

Παράδειγμα.

- Παράδειγμα 2.

$$X = \{x_1, x_2, x_3\}, \overline{X} = \{\overline{x_1}, \overline{x_2}, \overline{x_3}\}$$

$$F = \{(x_1 \vee x_2 \vee x_3), (x_1 \vee \overline{x_2}), (\overline{x_1} \vee x_3), (\overline{x_3})\}$$

- Ο F δεν είναι ικανοποιήσιμος!
- Γιατί?

Πρόβλημα Ικανοποιησιμότητας

- Ορισμός. **Ικανοποιησιμότητα** (Satisfiability) ή **SAT** : Δεδομένου ενός τύπου Bool F σε κανονική διαζευκτική μορφή, είναι ικανοποιήσιμος?

- Δεν υπάρχει γνωστός πολυωνυμικός αλγόριθμος για το πρόβλημα☹

Ειδικές περιπτώσεις δύσκολων προβλημάτων

- Περιορισμός SAT σε **δύο στοιχεία για κάθε συνθήκη : 2-SAT**
  - Παράδειγμα.

$$X = \{x_1, x_2, x_3\}, \overline{X} = \{\overline{x_1}, \overline{x_2}, \overline{x_3}\}$$

$$F = \{(x_1 \vee x_2), (x_1 \vee \overline{x_2}), (\overline{x_1} \vee x_3), (\overline{x_3})\}$$

Κλάση NP

Μη-Ντετερμινιστικές Μηχανές Turing:

- **Είναι δυνατόν σε μια συνολική κατάσταση να υπάρχουν πολλές δυνατές επόμενες συνολικές καταστάσεις (μεταβάσεις).**
- Ο υπολογισμός σε μια ΝΜΤ είναι ένα μονοπάτι από συνολικές καταστάσεις.
- Ο υπολογισμός σε μια **ΜΜΤ** είναι ένα δένδρο από συνολικές καταστάσεις.
- Μια ΜΜΤ M **δέχεται** μια είσοδο x εάν υπάρχει τουλάχιστον ένα πεπερασμένο μονοπάτι υπολογισμού $(a_0, \cdots, a_m)$, a_i: συνολική κατάσταση, στο δέντρο υπολογισμού της M με ρίζα $a_0 = (s, B\underline{x}B)$ και φύλλο $a_m = (h, uyv)$, όπου $a_i \vdash a_{i+1}$.
- Για κάποιες εισόδους η M το δένδρο υπολογισμού μπορεί να είναι άπειρο.

Λειτουργία ΜΜΤ

- Ορισμός. Για μια ΜΜΤ Μ, L(M) να είναι το σύνολο όλων των λέξεων που δέχεται η Μ.
- Στάδιο Εικασίας (μαντέματος, guessing)
  - **Μαντεύει** ποια είναι η κατάλληλη μεταβίβαση για την είσοδο που έχει και την εκτελεί.
- Στάδιο επαλήθευσης:
  - **Ελέγχει ντετερμινιστικά** αν το μάντεμα ήταν σωστό για την είσοδο που έχει.

Πολυπλοκότητα Χρόνου σε μια ΜΜΤ

Πολυπλοκότητα Χρόνου σε μια ΜΜΤ **Μ**:

- Για $x \in L(M)$, $\mathbf{Time_M(x)}$ είναι ο αριθμός βημάτων στο πιο σύντομο μονοπάτι υπολογισμού αποδοχής της x.
  - Δηλ. στο στάδιο του μαντέματος, μαντεύει με την πρώτη προσπάθεια το σωστό (αν υπάρχει).
- Εάν η Μ απορρίπτει την x, θέτουμε $Time_M(x) = \infty$.
- Η Μ έχει **φράγμα χρόνου t(n)** εάν $Time_M(x) \leq \mathbf{max}\{|x| + 1, t(|x|)\}$ για όλα τα $\mathbf{x \in L(M)}$.
- Ορισμός. Μια γλώσσα έχει **χρονική πολυπλοκότητα** f(n) εάν **αποφασίζεται** από μια ΜΜΤ σε φράγμα χρόνο f(n).

Κλάσεις Πολυπλοκότητας σε ΜΜΤ

- **NTIME(t(n))** = {L(M) | M είναι μια ΜΜΤ με φράγμα χρόνου t(n)}

- **NP** = $\bigcup_{c>0}$ NTIME(n^c) (**Non-Deterministic Polynomial Time**)

Γνωστά Θεωρήματα

- Παρατηρείστε:
  - $P \subseteq NP$

Πλεονέκτημα Μη Ντετερμινισμού

- Μια **μη-ντετερμινιστική** μηχανή Turing **μπορεί να μαντεύει**
  - τα υποψήφια μονοπάτια μπορεί να είναι εκθετικού πλήθους
    - Μια ντετερμινιστική μηχανή Turing θα έπρεπε να τα ελέγχει όλα για να βρει το σωστό $\Rightarrow$ εκθετικός χρόνος
  - Μια μη-ντετερμινιστική μηχανή μπορεί να μαντέψει το σωστό και να ελέγξει ότι όντως είναι.
- **Κλάση NP**
  - Περιλαμβάνει τα προβλήματα για τα οποία μια μη-ντετερμινιστική μηχανή μπορεί να μαντέψει ένα μονοπάτι υπολογισμού του προβλήματος (μια υποψήφια λύση) και να ελέγξει εάν είναι σωστό σε πολυωνυμικό χρόνο.

Ένα Μεγάλο Ερώτημα

- Η κλάση NP φαίνεται ότι περιέχει μεγαλύτερη πληθώρα προβλημάτων
- Υπάρχει ανάλογο του Πορίσματος 1 (PSPACE=NPSPACE) για τις (χρονικές) κλάσεις P και NP ? Δηλ.
  - **P = NP** ?
- **Το ερώτημα παραμένει ανοικτό από το 1970 και θεωρείται το πιο σημαντικό ανοικτό ερώτημα στη Θεωρία Πολυπλοκότητας…**
- **co-NP** = κλάση συνόλων A τα οποία τα συμπληρώματα ανήκουν στην NP.
  - Οι μη ντετερμινιστικές κλάσεις πολυπλοκότητας που είναι χρονικά φραγμένες δεν είναι γνωστό ότι είναι κλειστές ως προς το συμπλήρωμα.

Προβλήματα που ανήκουν στην κλάση NP

- **Το πρόβλημα SAT ανήκει στην κλάση NP.**

Απόδειξη.

- Σχεδιάζουμε μια ΜΜΤ M η οποία αποφασίζει σε πολυωνυμικό μη ντετερμινιστικό χρόνο για οποιαδήποτε στιγμιότυπο I=(X,F) του προβλήματος SAT εάν είναι ικανοποιήσιμο.
  - **1η φάση.** Έστω F το στιγμιότυπο του SAT στην είσοδο της μηχανής. Μετρά τις μεταβλητές του F (=n) και γράφει σε μια δεύτερη ταινία τη λέξη $\triangleright I^n$.
  - **2η φάση (Μη-ντετερμινιστική φάση).** Μη ντετερμινιστικά (μαντεύοντας), αντικαθιστά τη λέξη $\triangleright I^n$ με μια λέξη $w' \in \{\top, \bot\}$.
  - **3η φάση (Ντετερμινιστική φάση).** Ελέγχει εάν w' ικανοποιεί την F.
- Ο αλγόριθμος τρέχει σε πολυωνυμικό μη ντετερμινιστικό χρόνο. Γιατί?

Παράδειγμα. (Φροντιστήριο)

- Να δείξετε ότι το πρόβλημα του πλανώδιου πωλητή ανήκει στην κλάση NP.

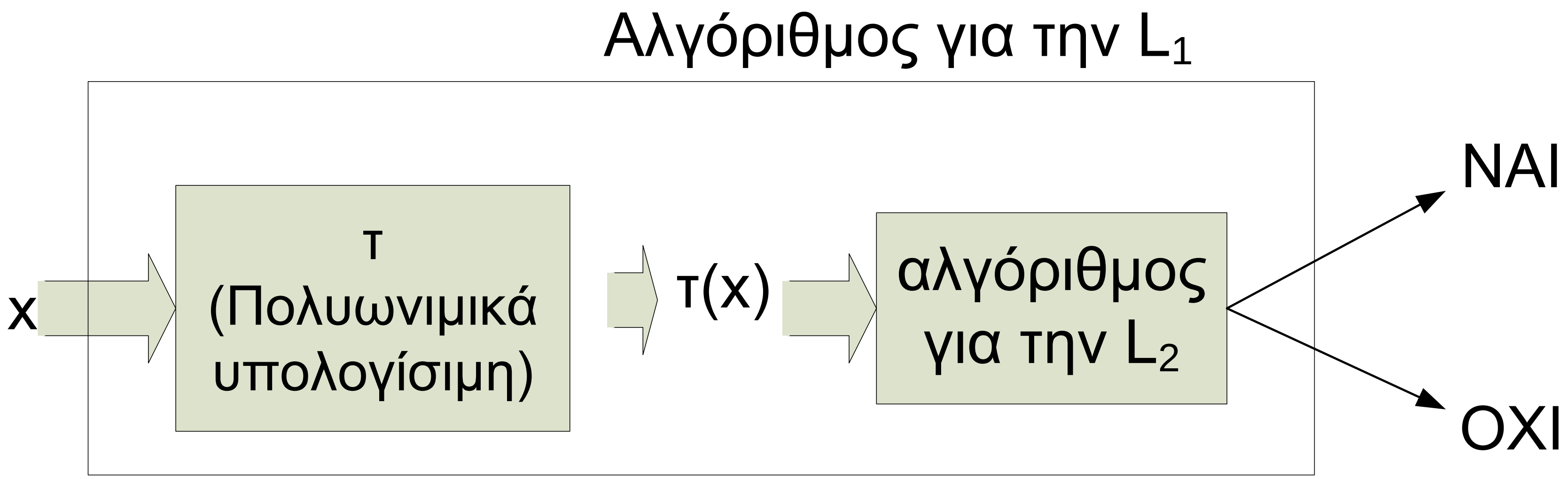
Απόδειξη.

- Είσοδος: I= (D, B) πίνακας n× n, στοιχεία : d_{ij}
- Μάντεμα (μη ντετερμινιστική φάση): Μαντεύει μια λύση για το I: Η ΜΜΤ γράφει σε μια 2η ταινία μια συμβολοσειρά από 0, 1 και ⊔ μήκους |I|.
- Έλεγχος (ντετερμινιστική φάση):
  1. Ελέγχει αν η λέξη που έγραψε είναι μια δυαδική κωδικοποίηση μιας μετάθεσης n αριθμών διαχωρισμένων με ένα κενό, δηλ. της μορφής $\pi(1) \sqcup \pi(2) \cdots \sqcup \pi(n)$.
     - A. Εάν ναι, τότε υπολογίζει το κόστος της μετάθεσης π, c(π) με βάση των πίνακα με τα κόστη διαδρομών D.
       - a) Εάν $c(\pi) \leq B$ αποφασίζει YES.
  2. Αλλιώς αποφασίζει NO.
- Ο αλγόριθμος ολοκληρώνεται σε πολυωνυμικό μη ντετερμινιστικό χρόνο.

Αναγωγές Πολυωνυμικού Χρόνου

- Ορισμός. f: $\Sigma^* \rightarrow \Sigma^*$ ονομάζεται **υπολογίσιμη σε πολυνωνυμικό χρόνο** αν υπάρχει μια πολυωνυμικά φραγμένη μηχανή Turing που την υπολογίζει.
- Έστω L_1, $L_2 \subseteq \Sigma^*$ δύο γλώσσες. Μια συνάρτηση υπολογίσιμη σε πολυωνυμικό χρόνο ονομάζεται **πολυωνυμική αναγωγή από την L_1 στην L_2** , $L_1 \leq^p L_2$, αν για κάθε $x \in \Sigma^*$ ισχύει:
  - $x \in L_1$ αν και μόνο αν $\tau(x) \in L_2$.
- **Συμπέρασμα**:
  - Η L_2 είναι τουλάχιστον όσο δύσκολη όσο η L_1.
    - Αν μπορούσαμε να λύσουμε την L_2 σε πολυωνυμικό χρόνο, θα μπορούσαμε να λύσουμε και την L_1 σε πολυωνυμικό χρόνο.
    - Αν δεν μπορούμε να λύσουμε την L_1 σε πολυωνυμικό χρόνο, δεν μπορούμε να λύσουμε ούτε την L_2 σε πολυωνυμικό χρόνο.

Σχηματικά


Αλγόριθμος για την L1
x
τ
(Πολυωνιμικά υπολογίσιμη)
τ(x)
αλγόριθμος για την L2
ΝΑΙ
ΟΧΙ

Παράδειγμα

- Θεώρημα 1. ΚΥΚΛΟΣ HAMILTON $\leq^{p}$ ΜΗ ΚΑΤΕΥΘΥΝΟΜΕΝΟΣ ΚΥΚΛΟΣ HAMILTON.