Θεωρία Υπολογισμού και Πολυπλοκότητα

Χρονική Πολυπλοκότητα

Στην ενότητα αυτή θα μελετηθούν τα εξής επιμέρους θέματα:

Μέτρηση της Πολυπλοκότητας (7.1)

Η κλάση P (7.2)

Η κλάση NP (7.3)

NP-πληρότητα (7.4)

Επιλυσιμότητα και Πολυπλοκότητα

- Δόγμα 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.
- Ανεπίλυτα προβλήματα: Προβλήματα για τα οποία δεν υπάρχει κανένας αλγόριθμος που να τα επιλύει
  - π.χ. το πρόβλημα του τερματισμού
- Επιλύσιμα Προβλήματα: Προβλήματα για τα οποία υπάρχει αλγόριθμος/Μηχανή Turing που τα επιλύει
  - Πολυπλοκότητα
    - Πόσος χρόνος/μνήμη απαιτείται για την επίλυση του προβλήματος;
    - Είναι η επίλυση του προβλήματος πρακτικά εφικτή;

Μέτρηση της Πολυπλοκότητας

- Η πιο κάτω μηχανή Turing διαγιγνώσκει τη γλώσσα
$$L_1 = \{w\#w \mid w \in \{a, b\}^*\}$$
- Ποια η χρονική της πολυπλοκότητα;

Μέχρι να συναντήσεις το σύμβολο #

Διάβασε και διάγραψε το επόμενο σύμβολο

Μετακινήσου δεξιά πέραν του # και όλων των x

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

Διαφορετικά

Γράψε x μετακινήσου αριστερά πέραν του # και σταμάτα δεξιά από το τελευταίο x

Αν βλέπεις μόνο # και μετά x να ακολουθούνται από ␣, αποδέξου τη λέξη

Εξαρτάται από:
- το μήκος της λέξης
- τη δομή της λέξης

Χρονική πολυπλοκότητα

ΟΡΙΣΜΟΣ

Έστω M μια ντετερμινιστική ΤΜ που τερματίζει σε κάθε είσοδο. Ο *χρόνος εκτέλεσης* ή η *χρονική πολυπλοκότητα* της M είναι η συνάρτηση $f: \mathbb{N} \to \mathbb{N}$ όπου $f(n)$ είναι το μέγιστο πλήθος βημάτων που είναι δυνατόν να πραγματοποιήσει η M όταν το μήκος της εισόδου της είναι n.

Εργαλεία Εκτίμησης Πολυπλοκότητας

- *Ασυμπτωτική ανάλυση*
  - Στόχος: εκτίμηση του χρόνου σε μεγάλα δεδομένα εισόδου
  - Δυνατότητα σύγκρισης του χρόνου εκτέλεσης διαφορετικών αλγορίθμων

- Παράδειγμα: Έστω αλγόριθμος με χρονική πολυπλοκότητα

$$f(n) = 6n^3 + 2n^2 + 20n + 45$$

Ασυμπτωτική ανάλυση: Μας ενδιαφέρει ο μεγιστοβάθμιος όρος (αγνοούμε τον συντελεστή του όρου όπως επίσης και τους όρους μικρότερου βαθμού).

Τότε λέμε ότι η πολυπλοκότητα του αλγορίθμου είναι ασυμπτωτικά το πολύ n^3:

$$f(n) = \mathrm{O}(n^3)$$

Ασυμπτωτική Ανάλυση – Η τάξη O

ΟΡΙΣΜΟΣ

Έστω f και g δύο συναρτήσεις από το σύνολο $\mathbb{N}$ στο σύνολο $\mathbb{R}^+$.

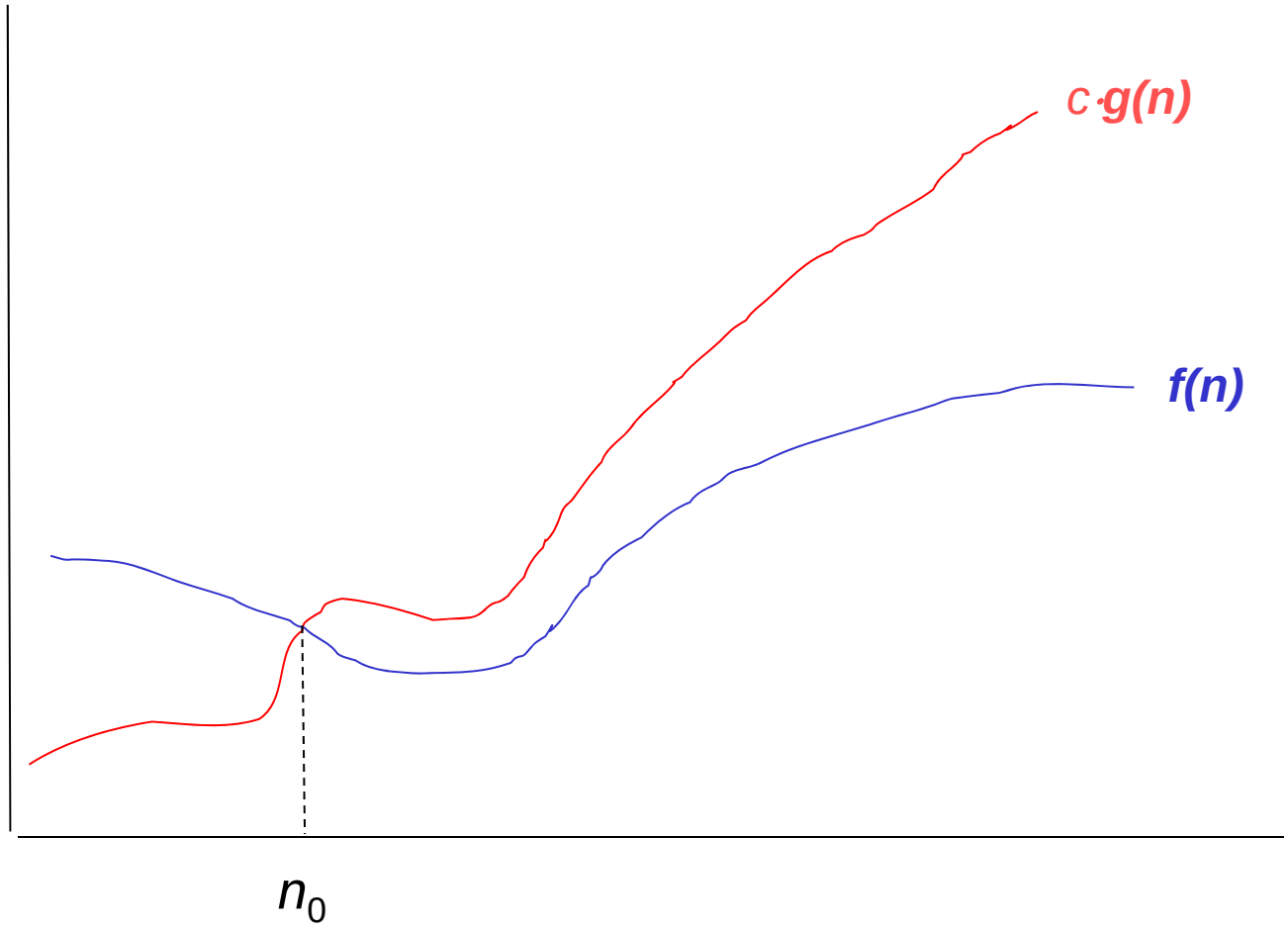
Ορίζουμε $f(n) = O(g(n))$, αν υπάρχουν ακέραιοι $c > 0$ και $n_0 \geq 0$ τέτοιοι ώστε για κάθε $n \geq n_0$ να ισχύει

$$f(n) \leq c \cdot g(n)$$

- Αν $f(n) = O(g(n))$, τότε λέμε πως η συνάρτηση f είναι *της τάξεως $g(n)$* ή της *τάξεως κεφαλαίο O του $g(n)$.*

- Αν $f(n) = O(g(n))$, τότε η $g(n)$ αποτελεί *ασυμπτωτικό, άνω φράγμα* της $f(n)$.

Γραφική Απεικόνιση $f(n) = O(g(n))$



Ασυμπτωτική ανάλυση – Η τάξη ο

ΟΡΙΣΜΟΣ

Έστω f και g δύο συναρτήσεις από το σύνολο $\mathbb{N}$ στο σύνολο $\mathbb{R}^+$.

Ορίζουμε $f(n) = o(g(n))$, αν ισχύει

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

- $f(n) = O(g(n)) \Rightarrow$ η f είναι ασυμπτωτικά μικρότερη ή ίση της g
- $f(n) = o(g(n)) \Rightarrow$ η f είναι ασυμπτωτικά μικρότερη της g

- Παραδείγματα
  - $f(n) = 6n^3 + 2n^2 + 20n + 45 \Rightarrow f(n) = o(n^4)$
  - $n = o(n \log n)$
  - $n \log n = o(n^2)$

Παράδειγμα 1

- Η πιο κάτω μηχανή Turing διαγιγνώσκει τη γλώσσα
$$L_1 = \{w\#w \mid w \in \{a, b\}^*\}$$
- Ποια η χρονική της πολυπλοκότητα;

Μέχρι να συναντήσεις το σύμβολο # — $O(n)$ φορές

- Διάβασε και διάγραψε το επόμενο σύμβολο
- Μετακινήσου δεξιά πέραν του # και όλων των x
- Αν το σύμβολο διαφέρει από αυτό που διαβάστηκε, απόρριψε τη λέξη
- Διαφορετικά
- Γράψε x μετακινήσου αριστερά πέραν του # και σταμάτα δεξιά από το τελευταίο x

$O(n)$ βήματα

Αν βλέπεις μόνο # και μετά x να ακολουθούνται από $\sqcup$, αποδέξου τη λέξη — $O(n)$ βήματα

Χρόνος εκτέλεσης: $O(n^2)$

Κλάση Χρονικής Πολυπλοκότητας

ΟΡΙΣΜΟΣ

Έστω $t : \mathrm{N} \rightarrow \mathrm{R}^+$ οποιαδήποτε συνάρτηση. Ορίζουμε ως *κλάση χρονικής πολυπλοκότητας* $\mathrm{TIME}(t(n))$ τη συλλογή όλων των γλωσσών που μπορούν να διαγνωστούν από κάποια μηχανή Turing χρόνου $\mathrm{O}(t(n))$.

- Η γλώσσα L_1 (Διαφάνεια 10-9) είναι τέτοια ώστε $L_1 \in \mathrm{TIME}(\mathrm{O}(n^2))$.

Παράδειγμα 2

$L_2 = \{0^n 1^n \mid n \geq 0\}$

M_1 = ‘Με είσοδο x,

1. Διατρέχουμε την ταινία και αν εντοπίσουμε κάποιο 0 στα δεξιά ενός 1 απορρίπτουμε. — $O(n)$ βήματα
2. Ενόσω η ταινία περιέχει και 0 και 1: — $O(n)$ φορές
3. Διατρέχουμε την ταινία και διαγράφουμε ένα 0 και ένα 1 — $O(n)$ βήματα
4. Εάν η ταινία εξακολουθεί να περιέχει 0 ή 1 απορρίπτουμε. Αλλιώς αποδεχόμαστε. — $O(n)$ βήματα

Χρόνος εκτέλεσης: $O(n^2)$

$(= O(n) + O(n^2) + O(n))$

Συμπέρασμα: $L_2 \in \text{TIME}(O(n^2))$.

Υπάρχει γρηγορότερος αλγόριθμος;

$$L_2 = \{0^n 1^n \mid n \geq 0\}$$

M_2 = ‘Με είσοδο x,

1. Διατρέχουμε την ταινία και αν εντοπίσουμε κάποιο 0 δεξιά κάποιου 1 απορρίπτουμε — $O(n)$ βήματα
2. Ενόσω η ταινία περιέχει και 0 και 1: — $O(\lg n)$ φορές
3. Διατρέχουμε την ταινία και απορρίπτουμε αν το σύνολο των συμβόλων είναι περιττό — $O(n)$ βήματα
4. Διατρέχουμε την ταινία και διαγράφουμε κάθε δεύτερο 0 και κάθε δεύτερο 1 (ξεκινώντας από το πρώτο). — $O(n)$ βήματα
5. Αν η ταινία είναι κενή αποδεχόμαστε. Αλλιώς απορρίπτουμε.

Χρόνος εκτέλεσης: $O(n \lg n)$

($= O(n) + O(\lg n) \cdot O(n) + O(n)$)

Συμπέρασμα: $L_2 \in \text{TIME}(O(n \lg n))$.

Υπάρχει ακόμη γρηγορότερος αλγόριθμος;

- Και αν έχουμε μια διταινιακή μηχανή Turing; $L_2 = \{0^n 1^n \mid n \geq 0\}$

M_3 = ‘Με είσοδο x,

1. Διατρέχουμε την ταινία και αν εντοπίσουμε κάποιο 0 δεξιά κάποιου 1 απορρίπτουμε — O(n) βήματα
2. Διατρέχουμε την ταινία μέχρι να συναντήσουμε το πρώτο 1. Ταυτόχρονα αντιγράφουμε τα 0 στη δεύτερη ταινία. — O(n) βήματα
3. Διατρέχουμε τα 1 της πρώτης ταινίας. Για κάθε 1 στην πρώτη ταινία, διαγράφουμε ένα 0 στην δεύτερη ταινία. — O(n) βήματα
4. Αν διαγραφούν όλα τα 0 και 1 αποδεχόμαστε. Αλλιώς απορρίπτουμε. — O(1) βήματα

Χρόνος Εκτέλεσης: O(n)

(= O(n) + O(n) + O(n) + O(1))

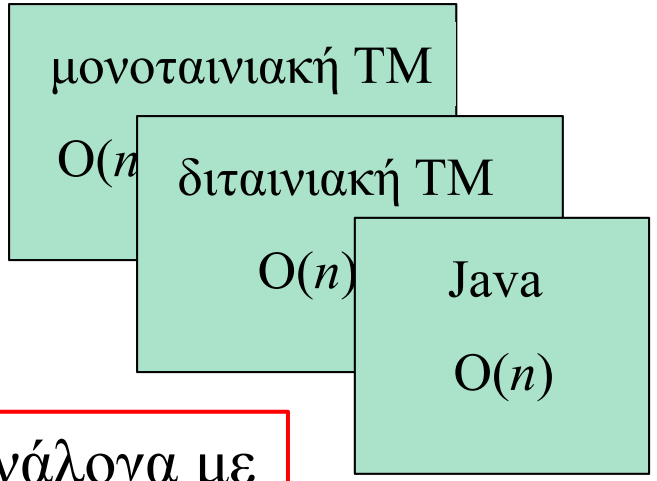
Συμπέρασμα: $L_2 \in$ TIME(O(n)).

Και στη Java;;

$L_2 = \{0^n 1^n \mid n \geq 0\}$

```
M(string x) {
  n = x.len;
  if n % 2 == 1 reject;
  else
   for (i = 1; i <= n/2; i++)
    if x[i] != 0 reject;
    if x[n-i+1] != 1 reject;
  accept;
}
```

Χρόνος Εκτέλεσης: $O(n)$



Ο χρόνος εκτέλεσης διαφοροποιείται ανάλογα με το μοντέλο υπολογισμού!

Σύνοψη

- Η γλώσσα L_2 μπορεί να τύχει διάγνωσης από μια μονοταινιακή ΤΜ σε χρόνο $O(n \lg n)$.
  - Μπορεί να δειχθεί ότι αυτό είναι και κάτω φράγμα για την επίλυση του προβλήματος (δεν υπάρχει ΤΜ που να μπορεί να πετύχει ταχύτερη εκτέλεση).

- Η γλώσσα L_2 μπορεί να τύχει διάγνωσης από μια διταινιακή ΤΜ σε χρόνο $O(n)$.
  - Αυτός είναι και ο βέλτιστος χρόνος για επίλυση του προβλήματος

- Συμπέρασμα: Η πολυπλοκότητα εξαρτάται από το μοντέλο!

Συμπεράσματα

- Θεωρία Υπολογισιμότητας
  - Δόγμα Church-Turing: Όλα τα εύλογα υπολογιστικά μοντέλα είναι ισοδύναμα (επιλύουν την ίδια κλάση από προβλήματα).

- Θεωρία Πολυπλοκότητας
  - Η επιλογή του μοντέλου επηρεάζει τη χρονική πολυπλοκότητα μιας γλώσσας
  - Ταξινόμηση προβλημάτων με βάση τη χρονική πολυπλοκότητα
  - Ποιο μοντέλο θα πρέπει να χρησιμοποιείται για τη μέτρηση του χρόνου;;

  - Τα καθιερωμένα ντετερμινιστικά μοντέλα δεν παρουσιάζουν μεγάλες διαφορές ως προς τις χρονικές τους απαιτήσεις.

Σχέσεις πολυπλοκότητας μεταξύ μοντέλων

ΘΕΩΡΗΜΑ

Έστω $t(n)$ κάποια συνάρτηση τέτοια ώστε $t(n) \geq n$. Για κάθε πολυταινιακή ΜΤ χρόνου $t(n)$ υπάρχει ισοδύναμη μονοταινιακή ΤΜ χρόνου $O(t^2(n))$

Σχέσεις πολυπλοκότητας μεταξύ μοντέλων

ΟΡΙΣΜΟΣ

Έστω N μια μη-ντετερμινιστική TM που τερματίζει σε κάθε είσοδο. Ο *χρόνος εκτέλεσης* της N είναι η συνάρτηση $f: \mathbb{N} \to \mathbb{N}$ όπου $f(n)$ είναι το μέγιστο πλήθος βημάτων που είναι δυνατόν να πραγματοποιήσει η N σε οποιοδήποτε κλάδο του υπολογισμού της για οποιαδήποτε είσοδο μήκους n.

ΘΕΩΡΗΜΑ

Έστω $t(n)$ κάποια συνάρτηση τέτοια ώστε $t(n) \geq n$. Για κάθε μη-ντετερμινιστική TM χρόνου $t(n)$ υπάρχει ισοδύναμη αιτιοκρατική μονοταινιακή TM χρόνου $2^{O(t(n))}$.

Πολυωνυμικός χρόνος

- Για κάθε πολυταινιακή ΤΜ υπάρχει μια ισοδύναμη μονοταινιακή ΤΜ η οποία είναι πολυωνυμικά πιο αργή
- Για κάθε μη-ντετερμινιστική ΤΜ υπάρχει μια ντετερμινιστική ΤΜ η οποία είναι εκθετικά πιο αργή
- Μια πολυωνυμική διαφορά ανάμεσα σε δύο χρόνους εκτέλεσης θεωρείται μικρή ενώ μια εκθετική διαφορά θεωρείται μεγάλη
  - Συγκρίνετε τις συναρτήσεις n^2 και 2^n για $n = 1000$.
- Αλγόριθμοι εκθετικού χρόνου εκτέλεσης είναι τόσο αργοί που σπάνια είναι χρήσιμοι
  - Συνήθως προκύπτουν όταν αναζητούμε τη λύση σε κάποιο πρόβλημα εκτελώντας μια εξαντλητική αναζήτηση στο πεδίο των πιθανών λύσεων.

Η Κλάση P

ΟΡΙΣΜΟΣ

Η κλάση γλωσσών P αποτελείται από τις γλώσσες που μπορούν να διαγνωστούν σε πολυωνυμικό χρόνο από κάποια ντετερμινιστική μηχανή Turing.

- Η κλάση P είναι αναλλοίωτη για μοντέλα που είναι πολυωνυμικώς ισοδύναμα με τη ντετερμινιστική TM
  - Το Δόγμα των Cobham-Edmonds: όλα τα «εύλογα» υπολογιστικά μοντέλο είναι πολυωνυμικά ισοδύναμα μεταξύ τους, και με τις TM (οποιοδήποτε μπορεί να προσομοιώσει οποιοδήποτε άλλο προκαλώντας μόνο πολυωνυμική αύξηση του χρόνου εκτέλεσης)

- Η κλάση P αντιστοιχεί χονδρικά στην κλάση των προβλημάτων που είναι πρακτικώς επιλύσιμα με υπολογιστές.

Παραδείγματα γλωσσών της κλάσης P

$L_{01} = \{0^n 1^n \mid n > 0\}$

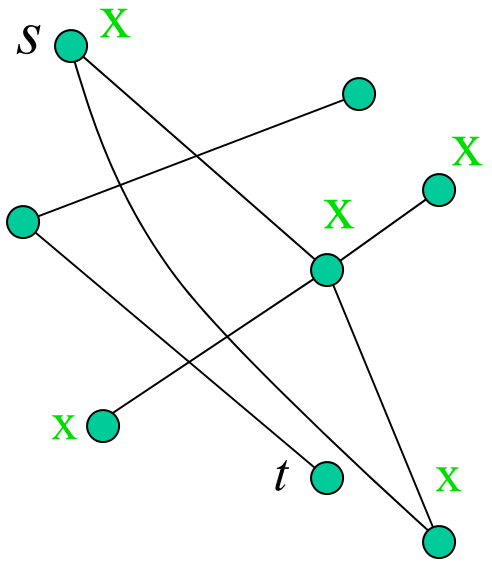
$L_G = \{x \mid \text{η λέξη } x \text{ παράγεται από τη CFG } G\}$ G κάποια CFG

ΔΙΑΔΡΟΜΗ = $\{(G, s, t) \mid G$ ένας γράφος με μονοπάτι από τον κόμβο s στον κόμβο $t\}$

ΑΜΟΙΒΑΙΑ_ΠΡΩΤΟΙ = $\{(n, m) \mid$ οι n και m δεν είναι αμοιβαία πρώτοι$\}$

ΔΙΑΔΡΟΜΗ ∈ P

ΔΙΑΔΡΟΜΗ =
{(*G*, *s*, *t*) | *G* ένας γράφος
με μονοπάτι από τον
κόμβο *s* στον κόμβο *t*}



Έστω ότι ο $G = (V, E)$, $|V| = n$, $|E| = |m|$

M := Με είσοδο ⟨*G*, *s*, *t*⟩,
όπου *G* ένας γράφος με κορυφές *s* και *t*

Σημάδεψε τον κόμβο *s*.

Επανέλαβε μέχρι να μην μπορείς να σημαδέψεις επιπλέον κόμβους: — O(*n*) φορές

Διάτρεξε τις ακμές του *G*. Αν εντοπίσεις ακμή (*a*, *b*), όπου η *a* είναι σημαδεμένη και η *b* όχι, σημάδεψε τη *b*. — O(*m*) βήματα

Αν ο *t* είναι σημαδεμένος αποδέξου, διαφορετικά απόρριψε.

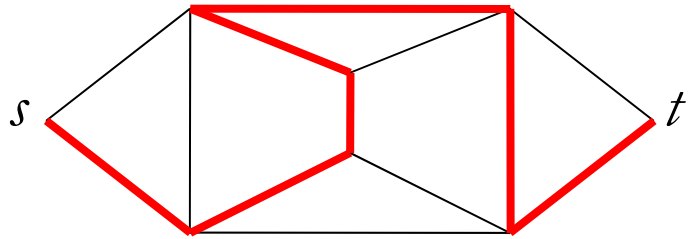
Χρόνος εκτέλεσης: O(*nm*) ✔

Η Κλάση NP

- Κλάση P: προβλήματα των οποίων η λύση μπορεί να επιτευχθεί σε πολυωνυμικό χρόνο, αποφεύγοντας την εξαντλητική αναζήτηση.
- Υπάρχουν όμως και πολλά προβλήματα για τα οποία όλες οι προσπάθειες να αποφευχθεί η εξαντλητική αναζήτηση έχει μέχρι στιγμής αποτύχει:
  - Δεν έχουν εντοπιστεί αλγόριθμοι πολυωνυμικού χρόνου εκτέλεσης που να τα επιλύουν.
- Που οφείλεται η αποτυχία αυτή;
  - Οι σχετικοί αλγόριθμοι βασίζονται σε άγνωστες αρχές;
  - Ή μήπως τα προβλήματα αυτά είναι εγγενώς δύσκολα;
- Θα μελετήσουμε τα πιο κάτω προβλήματα που ανήκουν σε αυτή την κατηγορία:
  - ΧΑΜΙΛΤΟΝΙΑΝΗ_ΔΙΑΔΡΟΜΗ, ΚΛΙΚΑ, SAT (ή ΕΠΑΛΗΘΕΥΣΙΜΟΤΗΤΑ)

Χαμιλτονιανή Διαδρομή

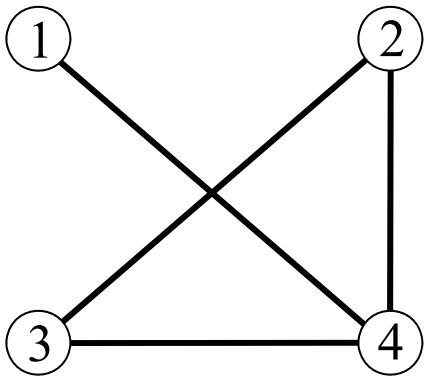
- Χαμιλτονιανή διαδρομή σε ένα κατευθυνόμενο γράφημα G είναι μια διαδρομή που διέρχεται από κάθε κόμβο του γραφήματος ακριβώς μια φορά.



ΧΑΜΙΛΤΟΝΙΑΝΗ_ΔΙΑΔΡΟΜΗ
$= \{\langle G, s, t\rangle |$ το G είναι ένα κατευθυνόμενο γράφημα που περιέχει μια Χαμιλτονιανή διαδρομή ανάμεσα στους κόμβους s και $t\}$

- Δεν γνωρίζουμε αν το πρόβλημα ΧΑΜΙΛΤΟΝΙΑΝΗ_ΔΙΑΔΡΟΜΗ $\in$ P

Το Πρόβλημα της Κλίκας



Το γράφημα *G*

Κλίκα σε ένα ακατεύθυντο γράφημα είναι ένα υπογράφημα στο οποίο οποιοιδήποτε δύο κόμβοι συνδέονται μέσω ακμής.

Οι {1, 4}, {2, 3, 4} και {1} αποτελούν κλίκες του G.

k-κλίκα σε ένα ακατεύθυντο γράφημα είναι μία κλίκα με k κόμβους.

Η {2, 3, 4} είναι μια 3-κλίκα του G.

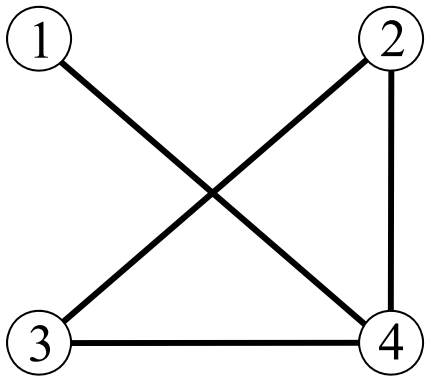
Το Πρόβλημα της Κλίκας: Να προσδιοριστεί αν ένα γράφημα περιέχει κάποια κλίκα δεδομένου μεγέθους.

Το Πρόβλημα της Κλίκας (2)

ΚΛΙΚΑ= $\{\langle G, k\rangle : G$ είναι ένας γράφος με k-κλίκα$\}$

M: "Με είσοδο$\langle G = (V,E).\ k\rangle$:

Για κάθε $S \subseteq V$, $|S| = k$:
Αν για κάθε ζεύγος κορυφών u, v in S
$(u, v) \in E$, αποδέξου
Διαφορετικά, απόρριψε."



Είσοδος: $\langle G, 3\rangle$

Υποσύνολα Μήκους 3: {1,2,3} {1,2,4} {1,3,4} {2,3,4}

Κλίκα;; ΟΧΙ ΟΧΙ ΟΧΙ ΝΑΙ

Το Πρόβλημα της Κλίκας (3)

- Ο αλγόριθμος εκτελεί μια εξαντλητική ανάλυση στο πεδίο των πιθανών λύσεων.
- Χρόνος Εκτέλεσης;;

M: "Με είσοδο$\langle G = (V,E).\ k\rangle$:

Για κάθε $S \subseteq V$, $|S| = k$:

Αν για κάθε ζεύγος κορυφών u, v in S

$(u, v) \in E$, αποδέξου

Διαφορετικά, απόρριψε."

$\binom{n}{k}$ υποσύνολα

$k(k-1)$ ζεύγη

$O(\binom{n}{k} \cdot k \cdot (k-1))$

$O(2^n)$ για $k = n/2$

- Δεν γνωρίζουμε αν το πρόβλημα ΚΛΙΚΑ $\in$ P, όπως και για το πρόβλημα ΧΑΜΙΛΤΟΝΙΑΝΗ_ΔΙΑΔΡΟΜΗ.
- Τι κοινό έχουν αυτά τα προβλήματα;

Πολυωνυμική Επαληθευσιμότητα

- Αν και δεν γνωρίζουμε αποδοτική μέθοδο επίλυσης των προβλημάτων…
- Αν μας δοθεί μια λύση μπορούμε εύκολα να *επαληθεύσουμε* την ορθότητά της. Για παράδειγμα,
  - Αν κάποιος μας δώσει μια *k*-κλίκα, για να επιβεβαιώσουμε τη λύση αρκεί να επαληθεύσουμε ότι κάθε ζεύγος κορυφών συνδέεται μέσω ακμής
  - Αν κάποιος μας δώσει μια Χαμιλτονιανή διαδρομή για να επιβεβαιώσουμε τη λύση αρκεί να επαληθεύσουμε ότι περνά από κάθε κορυφή ακριβώς μια φορά.
- Παρατήρηση 1: Η επαλήθευση της ορθότητας μιας λύσης φαίνεται να είναι ευκολότερη από την εύρεσή της.
- Παρατήρηση 2: Ορισμένα προβλήματα ενδέχεται να μην είναι πολυωνυμικά επαληθεύσιμα, π.χ.
  - ΧΑΜΙΛΤΟΝΙΑΝΗ_ΔΙΑΔΡΟΜΗ : ένας γράφος δεν έχει Χαμιλτονιανή διαδρομή
  - ~~ΚΛΙΚΑ:~~ ένας γράφος δεν έχει *k*-κλίκα

Η έννοια του επαληθευτή

ΟΡΙΣΜΟΣ

Ονομάζουμε *επαληθευτή* μιας γλώσσας A οποιονδήποτε αλγόριθμο V τέτοιο ώστε

$w \in A$ αν και μόνο αν ο V αποδέχεται τη λέξη $\langle w,s \rangle$

για κάποιο s.

Λέμε ότι ο *V έχει πολυωνυμικό χρόνο* αν ο χρόνος εκτέλεσης του είναι πολυωνυμικός ως προς το μήκος της λέξης w.

Μια γλώσσα ονομάζεται *πολυωνυμικά επαληθεύσιμη* αν υπάρχει για αυτήν επαληθευτής πολυωνυμικού χρόνου.

- Για να επαληθεύσει ότι μια λέξη w ανήκει στην γλώσσα A, ο επαληθευτής χρησιμοποιεί κάποια επιπλέον πληροφορία, την οποία αναπαριστά η λέξη s στον ορισμό.
- Η πληροφορία αυτή ονομάζεται *πιστοποιητικό*, ή *απόδειξη*, της συμμετοχής της w στην A.

Η Κλάση NP

ΟΡΙΣΜΟΣ

Η *κλάση γλωσσών NP* αποτελείται από τις γλώσσες που επιδέχονται επαληθευτή πολυωνυμικού χρόνου.

NP = Non-deterministic Polynomial time
(μη ντετερμινιστικός πολυωνυμικός χρόνος)

ΘΕΩΡΗΜΑ

Μια γλώσσα ανήκει στη κλάση NP *αν και μόνο αν* υπάρχει μηχανή Turing μη ντετερμινιστικού πολυωνυμικού χρόνου που να τη διαγιγνώσκει.

ΚΛΙΚΑ $\in$ NP

Απόδειξη 1:

- Θα πρέπει να προτείνουμε αλγόριθμο V που να αποτελεί ένα επαληθευτή πολυωνυμικού χρόνου για το πρόβλημα

V := "Για είσοδο $\langle G, k \rangle$ και σύνολο από κόμβους C,

1. Ελέγχουμε αν το σύνολο C έχει μήκος k.
2. Ελέγχουμε αν το G περιέχει ακμή που συνδέει κάθε ζεύγος κορυφών από το C.
3. Αν και οι δύο έλεγχοι δώσουν θετικό αποτέλεσμα αποδέξου.
4. Διαφορετικά απόρριψε."

Χρόνος εκτέλεσης: $O(k^2)$ ✔

ΚΛΙΚΑ $\in$ NP

Απόδειξη 2:

- Η πιο κάτω μη ντετερμινιστική μηχανή Turing διαγιγνώσκει το πρόβλημα:

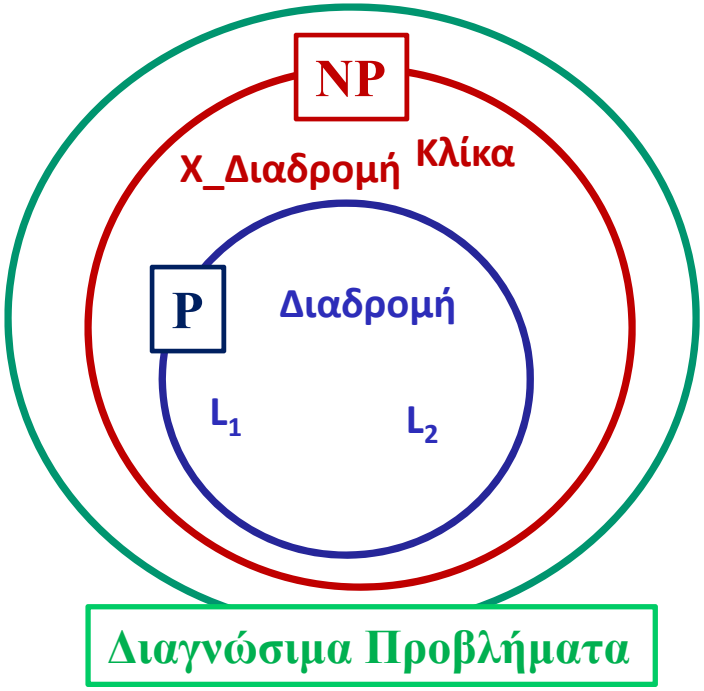
N := "Για είσοδο $\langle G, k \rangle$ όπου G ένα μη κατευθυνόμενο γράφημα και k ένας ακέραιος:

1. Επιλέγουμε μη ντετερμινιστικά ένα σύνολο k κόμβων του G, έστω C
2. Ελέγχουμε αν το G περιέχει όλες τις ακμές οι οποίες συνδέουν τους κόμβους του C.
3. Αν ναι, αποδεχόμαστε, διαφορετικά, απορρίπτουμε."

Χρόνος εκτέλεσης: $O(k^2)$ ✔

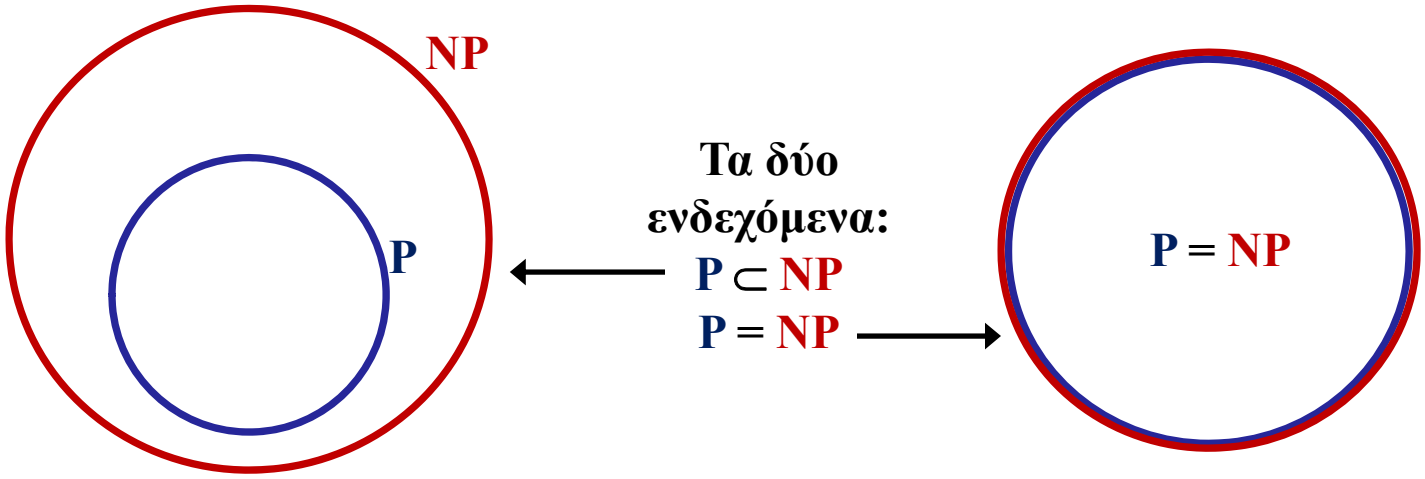
Το ερώτημα P έναντι NP

- **P** = η κλάση των γλωσσών στις οποίες η συμμετοχή μπορεί να *διαγνωστεί* γρήγορα
- **NP** = η κλάση των γλωσσών στις οποίες η συμμετοχή μπορεί να *επαληθευτεί* γρήγορα (και να διαγνωστεί "αργά").
- Η κλάση P είναι υποσύνολο της κλάσης NP.



Το ερώτημα P έναντι NP

- P = NP;
  - Ένα από τα σημαντικότερα άλυτα της θεωρητικής επιστήμης υπολογιστών και των σύγχρονων μαθηματικών.
- Επικρατέστερη άποψη : P ≠ NP
  - Έχει καταβληθεί τεράστια προσπάθεια για εύρεση αλγορίθμων πολυωνυμικού χρόνου για ορισμένα προβλήματά της χωρίς αποτέλεσμα.
- Για να αποδειχθεί ότι οι δύο κλάσεις διαφέρουν θα πρέπει να αποδειχθεί ότι δεν υπάρχει «ταχύς» αλγόριθμος για τα προβλήματα.



Millenium prize problems

- Το 2000, το Clay Mathematical Institute όρισε 7 προβλήματα για τον 21ο αιώνα.

1. P έναντι NP
2. Η εικασία του Hodge
3. Η εικασία του Poincaré ← Perelman 2003
4. Η υπόθεση Riemann
5. Yang–Mills existence and mass gap
6. Navier–Stokes existence and smoothness
7. Η εικασία των Birch και Swinnerton-Dyer

$1,000,000

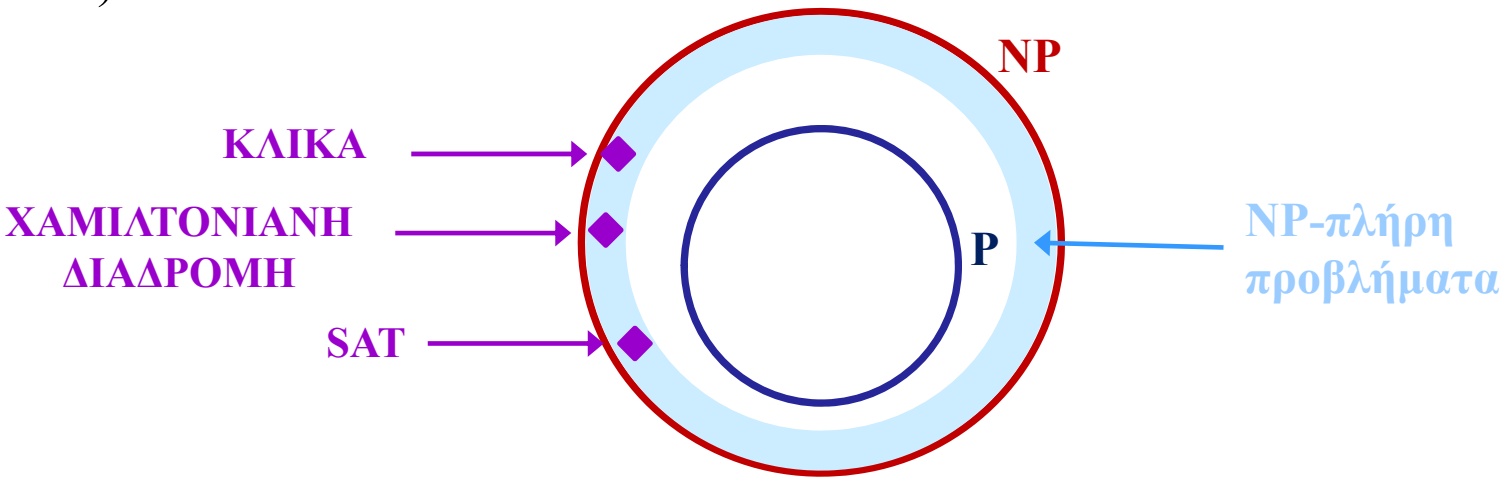
NP-Πληρότητα

- Εντατικές προσπάθειες για ταχεία επίλυση NP προβλημάτων έχουν μέχρι στιγμής αποτύχει.
- Αυτό οδηγεί στην υποψία ότι οποιοσδήποτε αλγόριθμος επίλυσης των προβλημάτων αυτών απαιτεί χρόνο της τάξης $O(2^n)$.
- Αυτό δεν έχει αποδειχθεί. Έχει όμως αποδειχθεί ότι:

Cook & Levin: Υπάρχουν κάποια προβλήματα στην κλάση NP για τα οποία ισχύει ότι
αν κάποιο από αυτά μπορεί να λυθεί σε πολυωνυμικό χρόνο τότε όλα τα προβλήματα της NP μπορούν να επιλυθούν σε πολυωνυμικό χρόνο.
Τα προβλήματα αυτά ονομάζονται *NP-πλήρη* (NP-complete).

NP-Πληρότητα

- Τα NP-πλήρη προβλήματα είναι εξίσου δύσκολα το ένα με το άλλο.
- Ταυτόχρονα είναι εξίσου ή περισσότερο δύσκολα από όλα τα υπόλοιπα NP προβλήματα
  - Αν κάποιο από αυτά μπορεί να επιλυθεί γρήγορα τότε όλα τα υπόλοιπα μπορούν να επιλυθούν γρήγορα.
- Παραδείγματα NP προβλημάτων: το Πρόβλημα της Κλίκας, το Πρόβλημα της Χαμιλτονιανής Διαδρομής και το πρόβλημα της Αληθευσιμότητας (SAT)…



Το Πρόβλημα της Αληθευσιμότητας

- *Λογικός τύπος* είναι οποιαδήποτε έκφραση περιλαμβάνει λογικές μεταβλητές και λογικές πράξεις, π.χ.

$$(x_1 \vee \neg x_2) \wedge (x_2 \vee \neg x_3 \vee x_4) \wedge (x_1)$$

- Ένας λογικός τύπος ονομάζεται *αληθεύσιμος* αν υπάρχει κάποιος συνδυασμός τιμών T (True) και F (False) στις μεταβλητές για τον οποίο ο τύπος να παίρνει την τιμή T.

- Ο πιο πάνω τύπος είναι αληθεύσιμος αφού, για παράδειγμα, για x_1=T, x_2 = F, x_3 = F, x_4 = F, παίρνει την τιμή T.

- Το Πρόβλημα της Αληθευσιμότητας

SAT = {⟨φ⟩ | ο φ είναι ένας αληθεύσιμος τύπος}

ΘΕΩΡΗΜΑ ΤΩΝ COOK-LEVIN

SAT ∈ P αν και μόνο αν P = NP.

SAT και NP

$$f = (x_1 \vee \neg x_2) \wedge (x_2 \vee x_3 \vee \neg x_4) \wedge (\neg x_1)$$

Εύρεση λύσης:

Δοκίμασε όλους τους δυνατούς συνδυασμούς

FFFF	FTFF	TFFF	TTFF
FFFT	FTFT	TFFT	TTFT
FFTF	FTTF	TFTF	TTTF
FFTT	FTTT	TFTT	TTTT

Για n μεταβλητές υπάρχουν 2^n διαφορετικοί συνδυασμοί

Χρόνος εκτέλεσης: Εκθετικός

Επαλήθευση λύσης:

FFTT

Αντικατέστησε
$x_1 = \text{F}\ x_2 = \text{F}\ x_3 = \text{T}\ x_4 = \text{T}$

Και αποτίμησε την τιμή του f για τη διδόμενη λύση

$f = (\text{F} \vee \text{T}) \wedge (\text{F} \vee \text{T} \vee \text{F}) \wedge (\text{T})$

Χρόνος εκτέλεσης: Γραμμικός

Το Πρόβλημα 3SAT

- *Λεξίγραμμα*: οποιαδήποτε λογική μεταβλητή ή η άρνηση μιας λογικής μεταβλητής (γράφουμε $\overline{x}$ για $\neg x$).
- *Φράση*: ένα σύνολο λεξιγραμμάτων που συνδέονται μέσω της πράξης της διάζευξης, π.χ. $\overline{x_1} \vee x_3 \vee x_5 \vee \overline{x_6}$
- Ένας λογικός τύπος βρίσκεται σε *συζευκτική κανονική μορφή* (ΣΚΜ) αν αποτελεί τη σύζευξη ενός συνόλου από φράσεις, π.χ.
$$(\overline{x_1} \vee x_3 \vee x_5 \vee \overline{x_6}) \wedge (x_2 \vee \overline{x_5}) \wedge (x_3 \vee \overline{x_4})$$
- *Τύπος 3ΣΚΜ*: ένας τύπος σε ΣΚΜ κάθε φράση του οποίου περιέχει ακριβώς τρία λεξιγράμματα, π.χ.
$$(\overline{x_1} \vee x_5 \vee \overline{x_6}) \wedge (x_2 \vee x_3 \vee \overline{x_5}) \wedge (\overline{x_2} \vee x_3 \vee \overline{x_4})$$
- Το Πρόβλημα 3SAT

  3SAT = {⟨φ⟩ | ο φ είναι ένας αληθεύσιμος 3ΚΣΜ τύπος}

NP-Πληρότητα και το ερώτημα "P έναντι NP"

- NP-πλήρη προβλήματα αποτελούν μια σημαντική εξέλιξη στο ερώτημα "P έναντι NP":
  - Αν πράγματι $P \subset NP$ που συνεπάγεται ότι η κλάση NP περιέχει προβλήματα που απαιτούν υπερπολυωνυμικό χρόνο τότε κάθε NP-πλήρες πρόβλημα είναι ένα από αυτά.
  - Αν όμως $P = NP$ τότε είναι αρκετό να δείξουμε ότι κάποιο NP-πλήρες πρόβλημα μπορεί να επιλυθεί σε πολυωνυμικό χρόνο.

- Η έννοιας της NP-πληρότητας χρησιμοποιείται επίσης για να δείξουμε ότι κάποιο πρόβλημα είναι δύσκολο να λυθεί:
  - Επιδεικνύουμε ότι είναι NP-πλήρες, επομένως τουλάχιστον τόσο δύσκολο όσο κάθε πρόβλημα στην NP.

Αναγωγιμότητα Πολυωνυμικού Χρόνου

- Τι εννοούμε λέγοντας π.χ. ότι "Το Πρόβλημα ΚΛΙΚΑ είναι τουλάχιστον τόσο δύσκολο όσο και το Πρόβλημα ΧΑΜΙΛΤΟΝΙΑΝΗ_ΔΙΑΔΡΟΜΗ";

- Εννοούμε ότι

  Αν το πρόβλημα ΧΑΜΙΛΤΟΝΙΑΝΗ_ΔΙΑΔΡΟΜΗ δεν επιλύεται μέσω κάποιου πολυωνυμικού αλγορίθμου τότε και το πρόβλημα ΚΛΙΚΑ δεν επιλύεται μέσω κάποιου πολυωνυμικού αλγόριθμου.

- Ισοδύναμα, μπορούμε να πούμε ότι οποιαδήποτε μηχανή Turing πολυωνυμικού χρόνου για το πρόβλημα ΚΛΙΚΑ μπορεί να μετασχηματιστεί σε μια μηχανή Turing πολυωνυμικού χρόνου για το πρόβλημα ΧΑΜΙΛΤΟΝΙΑΝΗ_ΔΙΑΔΡΟΜΗ.

Αναγωγιμότητα Πολυωνυμικού Χρόνου

- Πώς μπορούμε να περιγράψουμε μαθηματικά ότι ένα πρόβλημα Α ανάγεται σε ένα άλλο πρόβλημα Β σε πολυωνυμικό χρόνο?
  - Υπάρχει μια *υπολογίσιμη* συνάρτηση *πολυωνυμικού χρόνου εκτέλεσης* που μετατρέπει τα στιγμιότυπα του Α σε στιγμιότυπα του Β

- Αν έχουμε μια τέτοια συνάρτηση μετατροπής, τότε μια ταχεία λύση για το Β μπορεί να χρησιμοποιηθεί για να επιλυθεί ταχέως το Α..

ΟΡΙΣΜΟΣ

Μια συνάρτηση $f : \Sigma^* \rightarrow \Sigma^*$ λέγεται *υπολογίσιμη σε πολυωνυμικό χρόνο* αν υπάρχει ΤΜ πολυωνυμικού χρόνου εκτέλεσης που, για κάθε είσοδο w, τερματίζει έχοντας στην ταινία της μόνο τη λέξη $f(w)$.

Τυπικός Ορισμός της Αναγωγιμότητας Πολυωνυμικού Χρόνου

ΟΡΙΣΜΟΣ

Μια γλώσσα A είναι *απεικονιστικά αναγώγιμη σε πολυωνυμικό χρόνο* στη γλώσσα B, $A \leq_p B$, αν υπάρχει υπολογίσιμη συνάρτηση πολυωνυμικού χρόνου $f : \Sigma^* \to \Sigma^*$, τέτοια ώστε, για κάθε w,

$$w \in A \Leftrightarrow f(w) \in B.$$

Η συνάρτηση f λέγεται *πολυωνυμικού χρόνου αναγωγή* της A στη B.

- Χρήση: Για να δείξουμε ότι η NP γλώσσα B είναι NP-πλήρης:
  - Προσδιορίζουμε μια γλώσσα A που είναι γνωστό ότι είναι NP-πλήρης.
  - Εντοπίζουμε αναγωγή πολυωνυμικού χρόνου της A στη B.
  - Τότε κάποιος αλγόριθμος για τη B μπορεί να μετασχηματιστεί σε αλγόριθμο για την A με κόστος τη μετατροπή των στιγμιοτύπων της A σε στιγμιότυπα της B (πολυωνυμικός χρόνος εκτέλεσης)
  - Επομένως το B είναι τουλάχιστον τόσο δύσκολο όσο το A και άρα είναι και αυτό ένα NP-πλήρες πρόβλημα.

Αναγωγή 3SAT στην ΚΛΙΚΑ

ΘΕΩΡΗΜΑ

Αν η γλώσσα ΚΛΙΚΑ μπορεί να διαγνωστεί σε πολυωνυμικό χρόνο τότε και η γλώσσα 3SAT μπορεί να διαγνωστεί σε πολυωνυμικό χρόνο.

Απόδειξη:

- Ας υποθέσουμε ότι το πρόβλημα της ΚΛΙΚΑΣ μπορεί να διαγνωστεί σε πολυωνυμικό χρόνο.
  - Δηλαδή υπάρχει αλγόριθμος ο οποίος με δεδομένα εισόδου ένα γράφο και ένα ακέραιο k μπορεί να αποφασίσει κατά πόσο υπάρχει κλίκα μεγέθους k.
- Έστω ένας 3ΚΣΜ τύπος:

$$(a_1 \vee b_1 \vee c_1) \wedge (a_2 \vee b_2 \vee c_2) \wedge \ldots \wedge (a_k \vee b_k \vee c_k)$$

Αναγωγή 3SAT στην ΚΛΙΚΑ

- Ο τύπος αυτός μπορεί να μεταφραστεί σε πολυωνυμικό χρόνο σε ένα γράφο, έτσι ώστε ο γράφος να περιέχει κλίκα μεγέθους k αν και μόνο αν ο τύπος είναι αληθεύσιμος.

- Συνεπώς, μπορούμε να τρέξουμε τον αλγόριθμο ύπαρξης κλίκας στον γράφο που παράγεται:
  - Αν ο αλγόριθμος απαντήσει θετικά τότε ο τύπος είναι αληθεύσιμος
  - Αν ο αλγόριθμος απαντήσει αρνητικά τότε ο τύπος είναι μη αληθεύσιμος

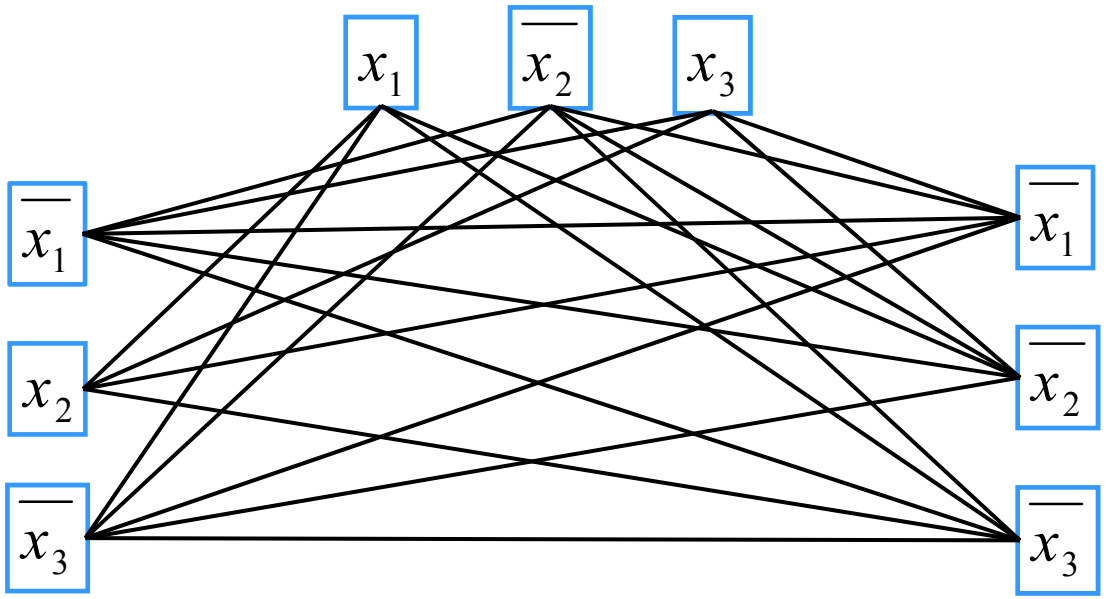
- Πως μπορεί να υλοποιηθεί αυτή η αναγωγή;
  - Δημιουργείται ένας γράφος του οποίου οι κόμβοι είναι όλα τα λεξικογράμματα του τύπου και ακμή ανάμεσα σε δύο κόμβους υπάρχει αν τα σχετικά λεξικογράμματα ανήκουν σε διαφορετικές φράσεις και δεν είναι αντίθετα μεταξύ τους

Αναγωγή 3SAT στην ΚΛΙΚΑ

- Για παράδειγμα, ο τύπος

$$(\overline{x_1} \vee x_2 \vee \overline{x_3}) \wedge (x_1 \vee \overline{x_2} \vee x_3) \wedge (\overline{x_1} \vee \overline{x_2} \vee \overline{x_3})$$

μεταφράζεται στον γράφο



Αναγωγή 3SAT στην ΚΛΙΚΑ

- Αν η φ είναι αληθεύσιμη τότε και ο σχετικός γράφος περιέχει κλίκα μεγέθους k:
  - Αφού υπάρχει συνδυασμός τιμών που κάνει τη φ True, κάθε φράση περιέχει (τουλάχιστον) ένα αληθές λεξίγραμμα. Παρατηρούμε ότι κάθε ζεύγος τέτοιων λεξιγραμμάτων συνδέεται με ακμή στον γράφο αφού δεν μπορούν να είναι αντίθετα (δεν είναι δυνατόν μια μεταβλητή να πάρει ταυτόχρονα τιμές Τ και F)

- Αν ο σχετικός γράφος περιέχει κλίκα μεγέθους k τότε και η φ είναι αληθεύσιμη:
  - Αν πάρουμε κάθε λεξίγραμμα της κλίκας και του αποδώσουμε την τιμή Τ τότε ο τύπος παίρνει επίσης την τιμή Τ.

- Συμπέρασμα: Αν το πρόβλημα ΚΛΙΚΑ επιλύεται σε πολυωνυμικό χρόνο τότε το ίδιο ισχύει και για το πρόβλημα 3SAT.

NP-Πληρότητα

ΟΡΙΣΜΟΣ

Μια γλώσσα Β είναι *NP-πλήρης* αν ικανοποιεί τις πιο κάτω συνθήκες:

1. Η Β ανήκει στην κλάση NP
2. Κάθε γλώσσα $A \in NP$ ανάγεται στη Β σε πολυωνυμικό χρόνο.

ΘΕΩΡΗΜΑ

Αν η γλώσσα Β είναι NP-πλήρης και $B \in P$ τότε $P = NP$.

ΘΕΩΡΗΜΑ

Αν η γλώσσα Β είναι NP-πλήρης, η γλώσσα $\Gamma \in NP$ και επιπλέον η Β μπορεί να αναχθεί σε πολυωνυμικό χρόνο στη Γ, τότε η Γ είναι επίσης NP-πλήρης.

NP-πλήρεις Γλώσσες

- Η γλώσσα SAT (πόρισμα Θεωρήματος Cook-Levin)
- Η γλώσσα 3SAT
- Η γλώσσα ΧΑΜΙΛΤΟΝΙΑΝΗ_ΔΙΑΔΡΟΜΗ
- Χρωματισμός Γράφων
- …

- Περισσότερα από τα διαγνώσιμα προβλήματα είτε ανήκουν στην P είτε είναι NP-πλήρη.