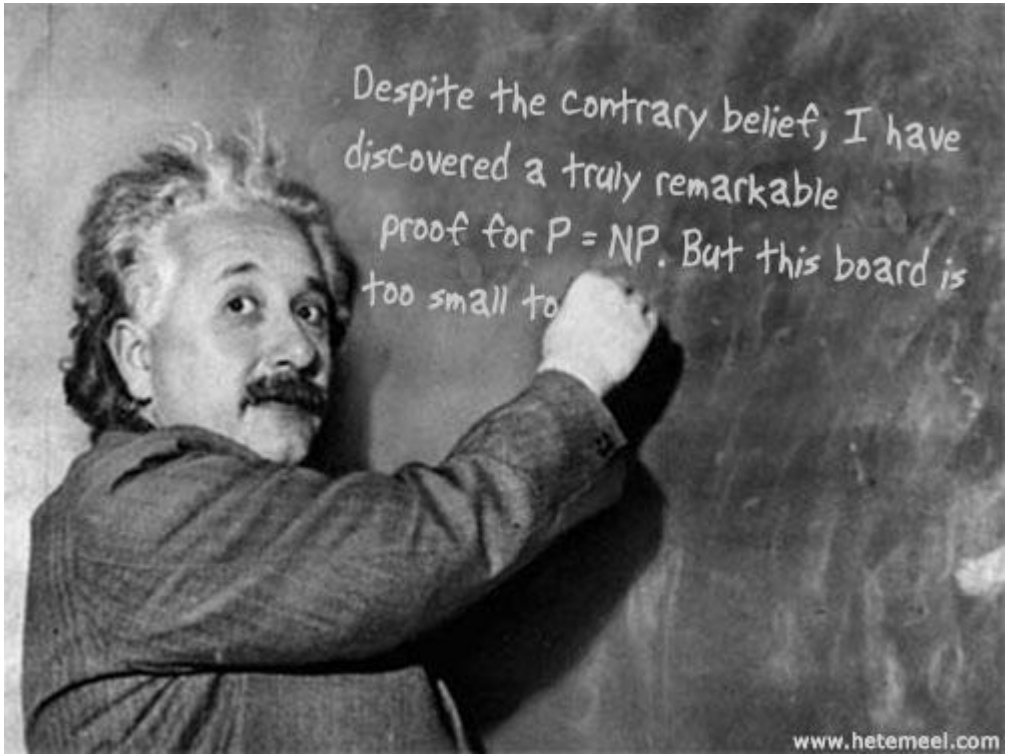
Πολυπλοκότητα

Οι Κλάσεις ***P*** και ***NP*** *και όλα τα υπόλοιπα θηρία*



Τσίχλας Κωνσταντίνος

Είδη Προβλημάτων

1. Προβλήματα Απόφασης (ΝΑΙ/ΌΧΙ)
2. Προβλήματα Αναζήτησης (επιστροφή δομής που πληρεί κάποιες απαιτήσεις)
3. Προβλήματα Βελτιστοποίησης (επιστροφή βέλτιστης δομής ως προς κάποια απαίτηση)
4. Προβλήματα Απαρίθμησης (όλες οι δομές που πληρούν κάποιες απαιτήσεις)
5. Προβλήματα Άθροισης (πλήθος δομών που πληρούν κάποιες απαιτήσεις)

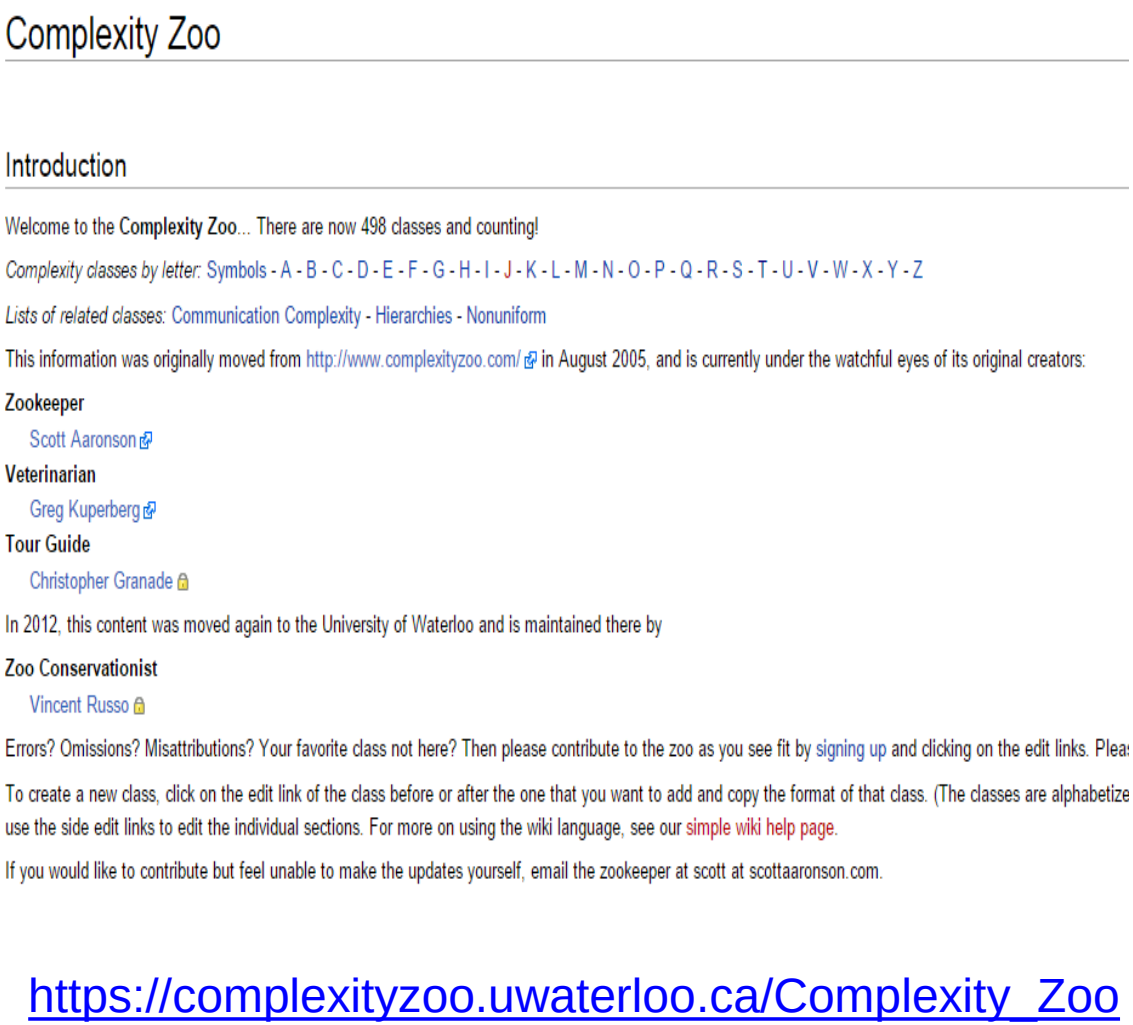
Κλάσεις Πολυπλοκότητας

- *Κλάσεις Υπολογισιμότητας*: σύνολα προβλημάτων που μπορούν να λυθούν (επιλυσιμότητα/αποδεκτικότητα) από έναν Η/Υ
- *Κλάσεις Πολυπλοκότητας*: σύνολα προβλημάτων που μπορούν να λυθούν από έναν Η/Υ σε περιορισμένο χρόνο και χώρο

Πόσες κλάσεις πολυπλοκότητας υπάρχουν;

Άπειρες! «Τα προβλήματα που μπορούν να λυθούν από έναν Η/Υ σε λιγότερα από 37 βήματα» είναι μία κλάση πολυπλοκότητας

Ενδιαφέρουσες Κλάσεις Πολυπλοκότητας



Complexity Zoo

Introduction

Welcome to the **Complexity Zoo**... There are now 498 classes and counting!

Complexity classes by letter: Symbols - A - B - C - D - E - F - G - H - I - J - K - L - M - N - O - P - Q - R - S - T - U - V - W - X - Y - Z

Lists of related classes: Communication Complexity - Hierarchies - Nonuniform

This information was originally moved from http://www.complexityzoo.com/ in August 2005, and is currently under the watchful eyes of its original creators:

Zookeeper

Scott Aaronson

Veterinarian

Greg Kuperberg

Tour Guide

Christopher Granade

In 2012, this content was moved again to the University of Waterloo and is maintained there by

Zoo Conservationist

Vincent Russo

Errors? Omissions? Misattributions? Your favorite class not here? Then please contribute to the zoo as you see fit by signing up and clicking on the edit links. Plea

To create a new class, click on the edit link of the class before or after the one that you want to add and copy the format of that class. (The classes are alphabetize use the side edit links to edit the individual sections. For more on using the wiki language, see our simple wiki help page.

If you would like to contribute but feel unable to make the updates yourself, email the zookeeper at scott at scottaaronson.com.

https://complexityzoo.uwaterloo.ca/Complexity_Zoo

498 (πέρυσι 496, πριν 2 χρόνια 495,πριν 3 χρόνια 493) «ενδιαφέρουσες» κλάσεις πολυπλοκότητας (και συνεχίζεται)!

Αιτιοκρατικός Χρόνος Εκτέλεσης

Έστω M ένας αλγόριθμος, και έστω:

$$t : N \rightarrow N$$

Λέμε ότι ο M εκτελείται σε χρόνο $t(n)$ αν:

Για κάθε είσοδο x μήκους n, το πλήθος των βημάτων που ο M κάνει είναι το μέγιστο $t(n)$.

Κλάση Χρονικής Πολυπλοκότητας

Έστω $t : N \rightarrow N$ μία συνάρτηση:

Ορισμός:

DTIME($t(n)$) = {L | L είναι ένα πρόβλημα που επιλύεται από έναν αλγόριθμο σε χρόνο $O(t(n))$}

Ανταιτιοκρατικός Χρόνος

Έστω N ένας ανταιτιοκρατικός αλγόριθμος και έστω:

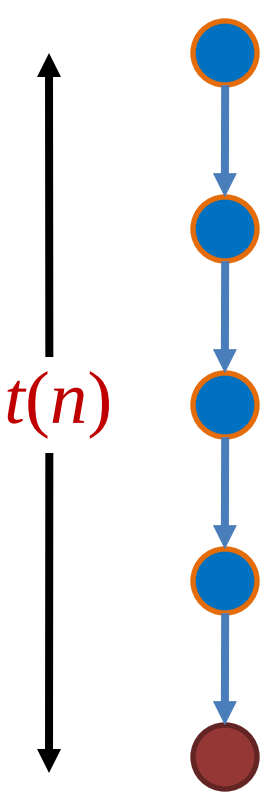
$$t : N \rightarrow N$$

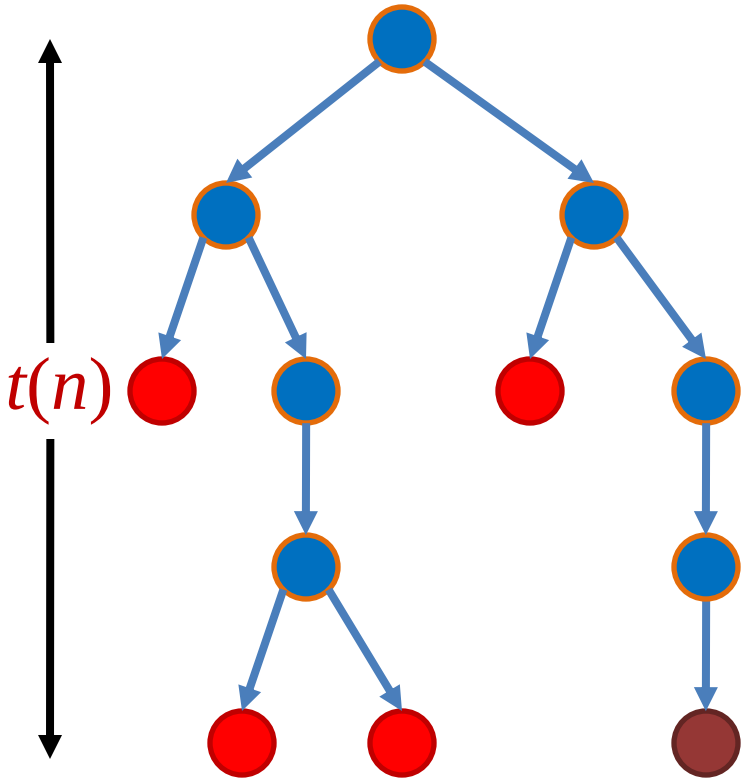
Θα λέμε ότι ο N εκτελείται σε χρόνο $t(n)$ αν

- για κάθε είσοδο x μήκους n,
- Το μέγιστο πλήθος βημάτων που ο N εκτελεί,
- σε οποιονδήποτε κλάδο του δέντρου υπολογισμού στο x,
- είναι το πολύ $t(n)$.

Αιτιοκρατικός vs Ανταιτιοκρατικός

Προσέξτε ότι απορριπτικά μονοπάτια απορρίπτουν μέσα σε το πολύ $t(n)$ βήματα.





Πολυπλοκότητα Ανταιτιοκρατικού Μοντέλου

Ισχυρισμός: Έστω N ένας ανταιτιοκρατικός αλγόριθμος που τρέχει σε χρόνο $t(n)$ και λύνει το πρόβλημα L.

Τότε: Υπάρχει ένας αιτιοκρατικός αλγόριθμος, D, που σε $2^{O(t(n))}$ βήματα λύνει το L.

Θεμελιώδης Διαφορές

- Το πολυωνυμικό χάσμα στον χρόνο μεταξύ διαφορετικών αιτιοκρατικών μοντέλων (RAM, δισδιάστατες, κτλ.)

- σε σύγκριση με

- το εκθετικό χάσμα στο χρόνο που υπάρχει μεταξύ αιτιοκρατικών και ανταιτιοκρατικών μοντέλων.

Προβλήματα Απόφασης

Θα ασχοληθούμε με προβλήματα στα οποία η απάντηση είναι ΝΑΙ/ΌΧΙ. Μήπως αυτό είναι κάπως περιοριστικό;

- Σε προβλήματα βελτιστοποίησης δεν είναι. (βρες το ελάχιστο πλήθος $\Leftrightarrow$ βρες αν υπάρχει λύση < όριο)

- Για υπολογισμό γενικών συναρτήσεων μπορεί να υπάρχει πρόβλημα (π.χ. $f(x)=2^x$).

Διαφορές Πολυπλοκοτήτων

- Ο πολυωνυμικός χρόνος είναι μικρός.
- Ο εκθετικός χρόνος είναι μεγάλος.

	10	20	30	40	50	60
n	.00001 second	.00002 second	.00003 second	.00004 second	.00005 second	.00006 second
n^2	.00001 second	.00004 second	.00009 second	.00016 second	.00025 second	.00036 second
n^3	.00001 second	.00008 second	.027 second	.064 second	.125 second	.216 second
n^5	.1 second	3.2 seconds	24.3 seconds	1.7 minute	5.2 minutes	13.0 minutes
2^n	.001 second	1.0 second	17.9 minutes	12.7 days	35.7 years	366 centuries
3^n	.059 second	58 minutes	6.5 years	3855 centuries	$2 \cdot 10^8$ centuries	$1.3 \cdot 10^{13}$ centuries

Πολυωνυμικός vs Εκθετικός

Ισχυρισμός: Όλα τα «λογικά» μοντέλα υπολογισμού είναι πολυωνυμικά ισοδύναμα.

Κάθε ένα μπορεί να εξομοιώσει το άλλο με το πολύ πολυωνυμική επιβάρυνση στον χρόνο εκτέλεσης.

Ερώτησεις:

Είναι ένα πρόβλημα επιλύσιμο σε γραμμικό χρόνο;

Εξαρτάται από το μοντέλο

Σε πολυωνυμικό χρόνο;

Ανεξάρτητο από το μοντέλο!!! (Μάλλον)

Η Κλάση *P*

Ορίζουμε μία κλάση αρκετά μεγάλη ώστε να μην επηρεάζεται από μικρές αλλαγές της ΤΜ.
μαθηματικώς ευσταθής κλάση

Ορισμός: *P* είναι το σύνολο των γλωσσών που είναι επιλύσιμες σε πολυωνυμικό χρόνο από μία ΤΜ.

$$P = \bigcup_{c \geq 0} DTIME\left(n^c\right)$$

Η κλάση αυτή είναι σημαντική αφού:

Όλες οι παραλλαγές της ΤΜ (λογικές) είναι πολυωνυμικά ισοδύναμες με μία απλή ΤΜ.

Σε γενικές γραμμές αντιστοιχεί στα ρεαλιστικώς επιλύσιμα προβλήματα (διαχειρίσιμα).

Η Κλάση P

- Η μετάβαση από εκθετικό σε πολυωνυμικό χρόνο για ένα πρόβλημα απαιτεί συνήθως εξαιρετική διαίσθηση.
- Αν βρεθεί ένας μη αποδοτικός πολυωνυμικός αλγόριθμος για ένα πρόβλημα, συνήθως μπορούμε να βρούμε έναν πιο αποδοτικό μετά.

Παραδείγματα: Προβλήματα στο P

Αριθμητική: Πρόσθεση, αφαίρεση, πολλαπλασιασμός, διαίρεση με υπόλοιπο.

Αλγόριθμοι σε Ακέραιους: Μέγιστος Κοινός Διαιρέτης.

Έρευνα Λειτουργίας: Μέγιστες ροές, γραμμικός προγραμματισμός.

Άλγεβρα: Πολλαπλασιασμός πινάκων, υπολογισμός οριζουσών, αναστροφή πίνακα, επίλυση γραμμικών συστημάτων.

Αλγόριθμοι Γράφων: BFS και DFS σε γραφήματα, Ελάχιστο ζευγνύον δένδρο, εύρεση μονοπατιού Euler.

Κωδικοποίηση

Για αριθμούς:

- Δυαδική αναπαράσταση (Καλή)
- Μοναδιαία αναπαράσταση μη-ρεαλιστική (εκθετικά μεγαλύτερη)

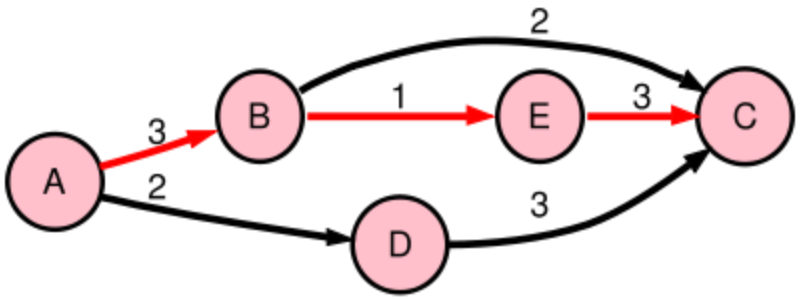
Για γραφήματα:

- Λίστα κόμβων και ακμών (Καλή)
- Πίνακας γειτνίασης (Καλή)

Το Πρόβλημα της Διαδρομής

Δοθέντος ενός κατευθυντού γραφήματος *G* και κόμβους *s* και *t*, υπάρχει διαδρομή από το *s* στο *t*;

ΔΙΑΔΡΟΜΗ = {⟨*G*, *s*, *t*⟩ | Το *G* έχει κατευθυντή διαδρομή από το *s* στο *t*.}



Πολυπλοκότητα Διαδρομής

Θεώρημα:

$$\text{ΔΙΑΔΡΟΜΗ} \in P$$

Όταν δεν ξέρουμε τι να κάνουμε : Εξάντληση

- Έστω m το πλήθος των κόμβων του G
- Κάθε διαδρομή από το s στο t δεν χρειάζεται να επαναλαμβάνει κόμβους
- Εξετάζουμε κάθε μονοπάτι του G μήκους $\leq m$
- Ελέγχουμε αν πάει από το s στο t.

Ερώτηση: Ποια η πολυπλοκότητα του αλγόριθμου;

m^m συνολικά μονοπάτια.
Ουπς, δεν ανήκει στο P!!

Πολυπλοκότητα Διαδρομής

Θεώρημα:

$$\text{ΔΙΑΔΡΟΜΗ} \in P$$

1. Μαρκάρουμε τον s
2. Επαναλαμβάνουμε μέχρι να μην μείνει καμία κορυφή:
   - Διαπέραση ακμών του G.
   - Αν η ακμή (a,b) από μαρκαρισμένο κόμβο a σε μη-μαρκαρισμένο κόμβο b,
   - Τότε μαρκάρισε το b.
3. Αν το t είναι μαρκαρισμένο τότε αποδεχόμαστε, αλλιώς απορρίπτουμε.

Ερώτηση: Ποια η πολυπλοκότητα;

Γραμμική…. Ως προς τι;;;;

Αμοιβαία Πρώτοι

Δύο αριθμοί είναι αμοιβαία πρώτοι αν το 1 είναι ο μεγαλύτερος ακέραιος που διαιρεί τέλεια και τους δύο αριθμούς.

- Το 10 και 21 είναι αμοιβαία πρώτοι
- Το 10 και 22 δεν είναι αμοιβαία πρώτοι

ΑΜΟΙΒΑΙΑ_ΠΡΩΤΟΙ = {$\langle x, y \rangle$ | οι x και y είναι αμοιβαία πρώτοι}

Αμοιβαία Πρώτοι(x,y)

Εξάντληση: δοκίμασε όλους τους αριθμούς μέχρι min(x,y) και έλεγξε αν διαιρούνται.

Πολυπλοκότητα;

Αν το x, y σε μοναδιαία μορφή:

- Το μέγεθος του $\langle x \rangle$, είναι x
- Έλεγχος των δυνατών διαιρετών του x, y είναι πολυωνυμικός

Αν το x, y σε δυαδική μορφή:

- Το μέγεθος του $\langle x \rangle$, είναι $\log x$
- Έλεγχος των δυνατών διαιρετών του x, y είναι εκθετικός

Τέτοιου τύπου αλγόριθμος ονομάζεται *ψευδο-πολυωνυμικός*.

Ο Αλγόριθμος του Ευκλείδη

Εύρεση Μέγιστου Κοινού Διαιρέτη, *E*:

Σε είσοδο $\langle x, y \rangle$

1. Επανέλαβε μέχρι $y = 0$
   - $x \leftarrow x \bmod y$
   - Αντάλλαξε το x με y
2. Έξοδος x

Πολυπλοκότητα;

1. Κάθε εκτέλεση του βήματος 1 μειώνει το x τουλάχιστον κατά το ήμισυ.
2. Πλήθος εκτελέσεων του 1 είναι $\min\{\log_2 x, \log_2 y\}$

Άρα πολυωνυμικός αλγόριθμος. Άρα: **ΑΜΟΙΒΑΙΑ_ΠΡΩΤΟΙ** $\in P$

Η Κλάση NTime

Έστω:

$$f : N \to N$$

μία συνάρτηση.

Ορισμός:

NTIME($f(n)$) = {L | Η L είναι μία γλώσσα επιλύσιμη από μία ανταιτιοκρατική ΤΜ σε $O(f(n))$ βήματα.}

Η Κλάση *NP*

Ορισμός:

Η κλάση *NP* είναι το σύνολο των γλωσσών που επιλύονται σε πολυωνυμικό χρόνο από μία ανταιτιοκρατική μηχανή.

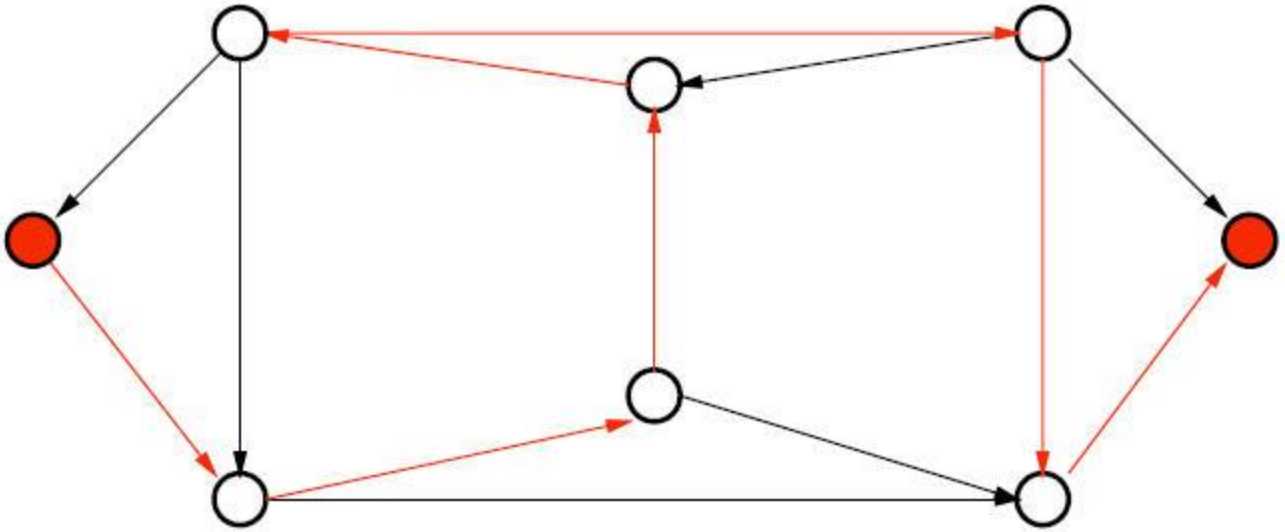
$$NP = \bigcup_{c \geq 0} NTIME\left(n^c\right)$$

Η κλάση *NP* είναι σημαντική αφού:

- Παραμένει ίδια ανεξαρτήτως πλήθους ταινιών.
- Η *NP* είναι αναλλοίωτη σε κάθε επιλογή «λογικού» ανταιτιοκρατικού μοντέλου υπολογισμού.
- Αντιστοιχεί σε προβλήματα των οποίων η λύση δεν μπορεί να παραχθεί αποδοτικά αλλά μπορεί να ***ελεγχθεί*** αποδοτικά.

Hamiltonian Μονοπάτι

Ένα Hamiltonian μονοπάτι σε ένα κατευθυντό γράφημα *G,* επισκέπτεται κάθε κόμβο ακριβώς μία φορά.



Hamiltonian Μονοπάτι

HAMPATH = $\{\langle G, s, t\rangle \mid$ Το G έχει μονοπάτι Hamiltonian από το s στο $t\}$

Ερώτηση: Πόσο δύσκολη είναι η επίλυση του προβλήματος;

Εύκολη μία εκθετικού χρόνου λύση παράγοντας όλα τα δυνατά μονοπάτια και ελέγχοντας κάθε ένα από αυτά αν είναι Hamiltonian.
(Εξαντλητικό Ψάξιμο)

Επίλυση από μία NTM

Σε είσοδο $\langle G, s, t \rangle$,

1. Μάντεψε και γράψε μία λίστα από αριθμούς $p_1,\ldots, p_m$
2. Ελέγχουμε για επαναλήψεις
3. Ελέγχουμε αν $p_1 = s$ και $p_m = t$
4. Ελέγχουμε αν (p_i, p_{i+1}) είναι μία ακμή του G

Βήμα 1: πολυωνυμικός χρόνος

Βήματα 2 και 3: απλοί έλεγχοι σε πολυωνυμικό χρόνο.

Βήμα 4: απλός έλεγχος σε πολυωνυμικό χρόνο

Hamiltonian Μονοπάτι

Αυτό το πρόβλημα έχει μία πολύ ενδιαφέρουσα ιδιότητα:
πολυωνυμική επαληθευσιμότητα

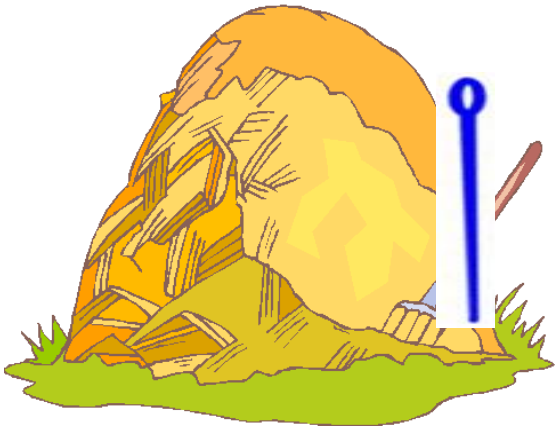
- Δεν ξέρουμε έναν γρήγορο τρόπο για να βρούμε ένα τέτοιο μονοπάτι
- αλλά μπορούμε να ελέγξουμε γρήγορα αν ένα δοθέν μονοπάτι είναι Hamiltonian.

Με άλλα λόγια:

- Η επαλήθευση ορθότητας ενός μονοπατιού είναι πιο εύκολη
- από το να βρούμε αν υπάρχει ένα τέτοιο

Εύρεση Βελόνας

Είναι η εξαντλητική αναζήτηση απαραίτητη;



Όχι αν έχεις μαγνήτη (ή vodafone)

Σύνθετοι Αριθμοί

Ένας φυσικός αριθμός είναι σύνθετος αν είναι γινόμενο δύο φυσικών αριθμών μεγαλύτερων του 1.

COMP = { $\langle x \rangle$ | $x = pq$ για ακέραιους $p, q > 1$}

Δεν έχουμε αποδοτικό πολυωνυμικό αλγόριθμο για να επιλύουμε το συγκεκριμένο πρόβλημα αλλά μπορούμε εύκολα να επαληθεύσουμε αν ένας αριθμός είναι σύνθετος

$2^{O\left(\sqrt[3]{n \log n}\right)}$ για αριθμούς με n bits

Στην πραγματικότητα, το καλοκαίρι του 2002, δύο φοιτητές (σαν και εσάς) και ο καθηγητής τους βρήκαν τρόπο να το κάνουν σε πολυωνυμικό χρόνο.

Επαληθευσιμότητα

Ένας επαληθευτής για την γλώσσα A είναι ένας αλγόριθμος V έτσι ώστε:

$$A = \{w \mid \text{Ο } V \text{ αποδέχεται τη λέξη } \langle w,c\rangle, \text{ για κάποια λέξη } c\}$$

- Ο επαληθευτής χρησιμοποιεί την επιπρόσθετη πληροφορία c για να επαληθεύσει το αν $w \in A$.
- Μετράμε το χρόνο της επαλήθευσης με βάση το μήκος του w.
- Η λέξη c καλείται πιστοποιητικό (ή απόδειξη) του w αν ο V αποδέχεται τη λέξη $\langle w,c\rangle$.
- Ένας πολυωνυμικός επαληθευτής εκτελείται σε πολυωνυμικό χρόνο ως προς το $|w|$ (άρα $|c| \leq |w|^{O(1)}$).
- Μια γλώσσα A είναι πολυωνυμικά επαληθεύσιμη αν έχει πολυωνυμικό επαληθευτή.

Πιστοποιητικά για HAMPATH

Για το HAMPATH, ένα πιστοποιητικό για

$$\langle G,s,t\rangle \in \text{HAMPATH}$$

είναι απλά το Hamiltonian μονοπάτι από το s στο t.

Μπορούμε να επαληθεύσουμε σε χρόνο πολυωνυμικό σε σχέση με το $|\langle G\rangle|$ αν το μονοπάτι είναι Hamiltonian.

Πιστοποιητικά για Συνθετότητα

Για τους σύνθετους αριθμούς ένα πιστοποιητικό για:

$$x \in \text{COMPOSITES}$$

είναι απλά κάποιος από τους διαιρέτες του.

Μπορούμε να επαληθεύσουμε σε χρόνο πολυωνυμικό ως προς $|x|$ αν ο δοθέν διαιρέτης πραγματικά διαιρεί το x.

Επαληθευσιμότητα

Δεν είναι όλα τα προβλήματα πολυωνυμικώς επαληθεύσιμα.

Δεν υπάρχει τρόπος (μάλλον) να επαληθεύσουμε σε πολυωνυμικό χρόνο το co-HAMPATH.

Θα δούμε πολλά παραδείγματα όπου η γλώσσα *L* είναι πολυωνυμικώς επαληθεύσιμη αλλά το συμπλήρωμά της, *co-L*, δεν είναι γνωστό αν είναι πολυωνυμικώς επαληθεύσιμο.

Η NP και Επαληθευσιμότητα

Θεώρημα: Μία γλώσσα ανήκει στην κλάση *NP* αν και μόνο αν έχει πολυωνυμικό επαληθευτή.

Απόδειξη – Διαίσθηση:

Η NTM εξομοιώνει τον επαληθευτή μαντεύοντας την απόδειξη.

Ο επαληθευτής εξομοιώνει την NTM, χρησιμοποιώντας ως απόδειξη τον αποδεκτικό κλάδο υπολογισμού (πολυωνυμικά μεγάλο).

$\Rightarrow$

Ισχυρισμός: Αν η A έχει πολυωνυμικό επαληθευτή τότε μπορεί να επιλυθεί από μία NTM σε πολυωνυμικό χρόνο.

Έστω V ένας πολυωνυμικός επαληθευτής της A.

- Μονοταινιακή TM
- Χρόνος εκτέλεσης n^k

N: σε είσοδο w μήκους n

- Ανταιοτιοκρατικά επέλεξε λέξη c μήκους n^k
- Τρέξε το V *στο* $\langle w,c \rangle$
- Αν η V αποδέχεται, αποδεχόμαστε αλλιώς απορρίπτουμε

$\Leftarrow$

Ισχυρισμός: Αν η A επιλύεται από μία NTM N σε πολυωνυμικό χρόνο n^k, τότε η A έχει πολυωνυμικό επαληθευτή.

Κατασκευάζουμε έναν πολυωνυμικό επαληθευτή V ως εξής:

V: σε είσοδο w μήκους n, και σε λέξη c μήκους n^k

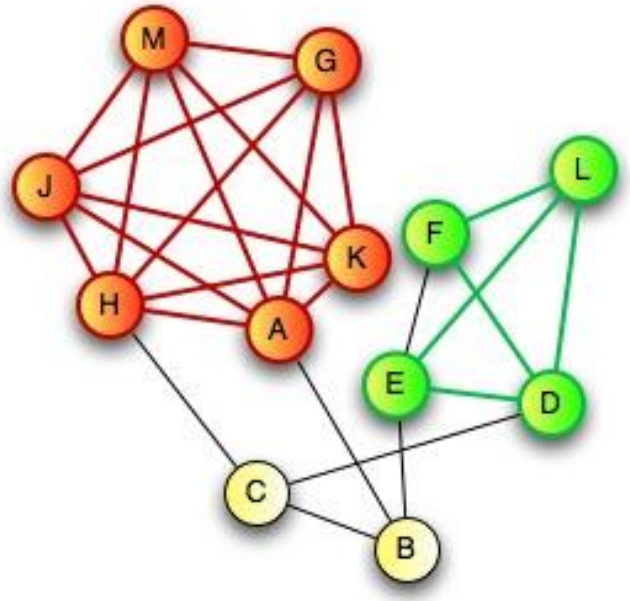
- Εξομοιώνουμε την N σε είσοδο w, χρησιμοποιώντας κάθε σύμβολο του c σαν περιγραφή της ανταιτιοκρατικής επιλογής σε κάθε βήμα της N.
- Αν αυτός ο κλάδος αποδέχεται, αποδεχόμαστε αλλιώς απορρίπτουμε.

Παραδείγματα: Κλίκα

Μία κλίκα σε ένα γράφημα είναι ένα υπογράφημα όπου κάθε ζευγάρι συνδέεται με μία ακμή.

Μία k-κλίκα είναι μία κλίκα τάξης k.

Ποια είναι η μεγαλύτερη k-κλίκα στο σχήμα;



Κλίκα

ΚΛΙΚΑ = {⟨G, k⟩ | Ο G είναι ένα γράφημα με k-κλίκα}

Θεώρημα:

$$\textit{ΚΛΙΚΑ} \in \textit{NP}$$

Η κλίκα είναι το πιστοποιητικό.

Ένας επαληθευτής V: σε είσοδο ⟨G, k, c⟩

- Αν c δεν είναι k-κλίκα, απορρίπτουμε
- Αν ο G δεν περιέχει όλες τις κορυφές του c, απορρίπτουμε
- Αποδεχόμαστε

Κλίκα $\in$ NP

Εναλλακτική Απόδειξη:

Θα φτιάξουμε μία NTM *N* που να επιλύει το συγκεκριμένο πρόβλημα.

Ομοιότητα με την προηγούμενη απόδειξη;;;;

N: για είσοδο $\langle G,k \rangle$

1. Επιλέγουμε ανταιτιοκρατικά ένα σύνολο *k* κόμβων του *G*, έστω *c*.
2. Ελέγχουμε αν το *G* περιέχει όλες τις ακμές που συνδέουν κόμβους του *c*
3. Αν ναι, αποδεχόμαστε αλλιώς απορρίπτουμε.

Άθροισμα Υποακολουθίας

Στιγμιότυπο του προβλήματος:

Μια συλλογή από αριθμούς $x_1, \ldots, x_k$

Αριθμός στόχος: t

Ερώτηση: Υπάρχει υποακολουθία της οποίας το άθροισμα να είναι ακριβώς t;

ΑΘ_ΥΠ = $\{\langle S, t\rangle \mid S = \langle x_1, \ldots, x_k\rangle, \exists\, y_1, \ldots, y_\lambda \subseteq x_1, \ldots, x_k: \Sigma y_i = t\}$

Άθροισμα Υποακολουθίας

Παράδειγμα:

- $(\{4, 11, 16, 21, 27\}, 25) \in$ ΑΘ_ΥΠ αφού $4 + 21 = 25$.
- $(\{4, 11, 16, 21, 27\}, 26) \notin$ ΑΘ_ΥΠ (γιατί;)

Θεώρημα:

$$\text{ΑΘ\_ΥΠ} \in NP$$

Η υποακολουθία είναι το πιστοποιητικό.

Ο επαληθευτής: V σε είσοδο $\langle S, t, c \rangle$

- Έλεγξε αν c είναι μία συλλογή αριθμών με άθροισμα t
- Έλεγξε αν το c είναι υποακολουθία του S
- Αν αποτύχουμε σε (1) ή (2) τότε απόρριψη αλλιώς αποδοχή

Συμπληρωματικά Προβλήματα

Η *co-ΚΛΙΚΑ* και *co-ΑΘ-ΥΠ* φαίνεται ότι δεν είναι μέλη της κλάσης *NP*.

Είναι πιο δύσκολη η αποδοτική επαλήθευση ότι κάτι δεν υπάρχει παρά η αποδοτική επαλήθευση ότι κάτι υπάρχει.

Ορισμός: Η κλάση co-NP:

Κλάση προβλημάτων για τα οποία υπάρχουν αποδοτικώς επαληθεύσιμα αντιπαραδείγματα

$$L \in \text{co-NP} \text{ αν } \text{co-}L \in \text{NP}$$

Μέχρι τώρα δεν ξέρουμε αν οι κλάσεις co-NP και NP είναι πράγματι διαφορετικές.

co-NP

- Έστω το *co-ΑΘ-ΥΠ*: δοθέντος ενός συνόλου ακεραίων, κάθε υποσύνολο να έχει άθροισμα που να μην είναι ίσο με το στόχο; έχουμε μία απόδειξη για αρνητική απάντηση (*αντιπαράδειγμα*) που είναι αποδοτικά επαληθεύσιμη, απλά ένα σύνολο που να έχει άθροισμα ίσο με το στόχο.

Παράδειγμα:

$(\{4, 11, 16, 21, 27\}, 25) \notin$ co-ΑΘ_ΥΠ αφού $4 + 21 = 25$

$(\{4, 11, 16, 21, 27\}, 26) \in$ co-ΑΘ_ΥΠ αφού ?????

Ασυμμετρία σε NP και coNP

Η Κλάση **NP κλειστή ως προς ένωση, τομή, και** σώρευση.

- Πιστεύουμε ότι κλάση **NP δεν είναι κλειστή ως** προς συμπλήρωμα (ασυμμετρία υπέρ αποδοχής).
- **co-NP: αντίστοιχη κλάση με ασυμμετρία υπέρ απόρριψης.**

NP = co-NP ?

- Έχουν τα αποδεκτικά στιγμιότυπα αποδοτικούς επαληθευτές αν έχουν και τα απορριπτικά στιγμιότυπα;

Μάλλον ΟΧΙ.

Αν NP ≠ co-NP, τότε P ≠ NP.

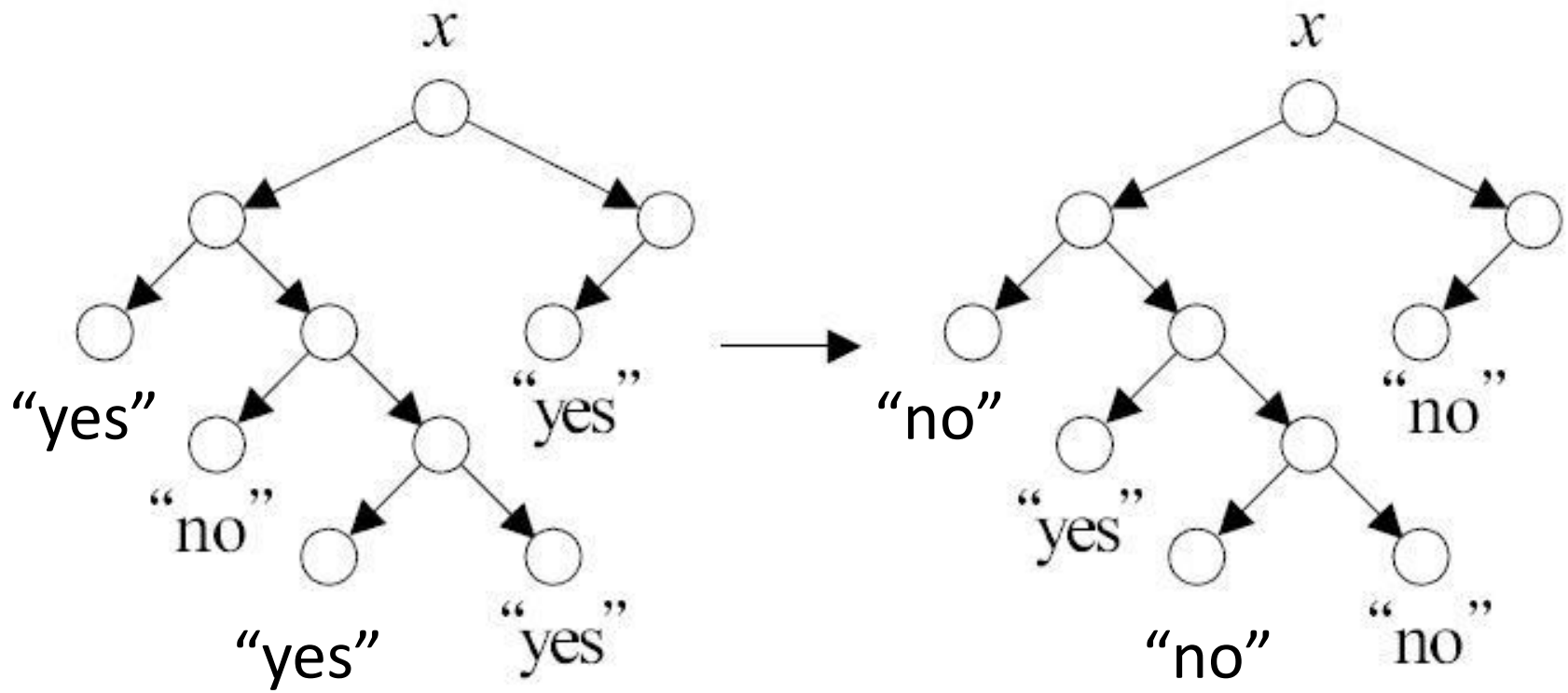
- Ιδέα απόδειξης:
  - Το P είναι κλειστό ως προς το συμπλήρωμα.
  - Αν P = NP, τότε και το NP είναι κλειστό ως προς το συμπλήρωμα.
  - Με άλλα λόγια, NP = co-NP.
  - Αυτό είναι το αντιθετοαντίστροφο του θεωρήματος.

Συμπληρώνοντας τις Εισόδους

Έστω *M* διαγιγνώσκει την *L*, και *M΄* είναι η *M* μετά την αντιστροφή εξόδου.

- Αν η *M* είναι μία αιτιοκρατική ΤΜ τότε η *M΄* αποφασίζει την *L΄*.
- Αν η *M* είναι μία ανταιτιοκρατική ΤΜ, τότε η *M΄* μπορεί να μην αποφασίζει την *L΄*.

Είναι πιθανό οι *M* και *M΄* να δέχονται και οι δύο μία είσοδο *x* και άρα δεν είναι συμπληρωματικές οι γλώσσες τους


x
"yes"
"no"
"yes"
"yes"
"yes"
x
"no"
"yes"
"no"
"no"
"no"

NP ∩ *co-NP*

IF ∈ P;;;;

- Μπορείτε να σκεφτείτε προβλήματα που ανήκουν στο *NP* ∩ *co-NP;;;*

Η κλάση προβλημάτων P

- Το πρόβλημα παραγοντοποίησης ακεραίων:

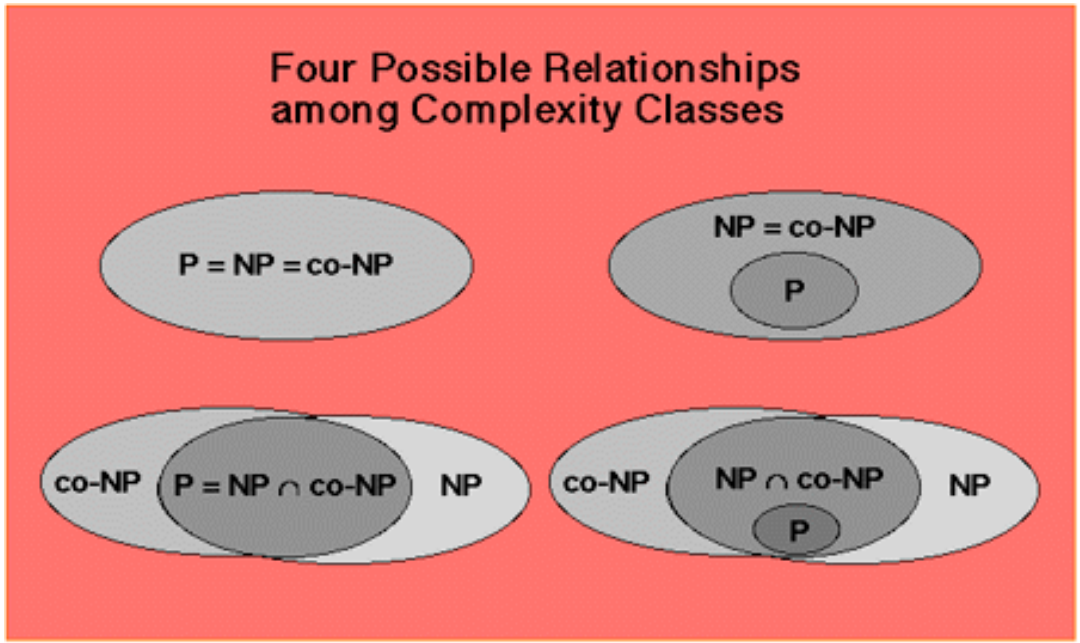
$$IF = \{\langle m, n\rangle \mid m = x \times q, x \in (1, n], q \in N\}$$

IF ∈ *NP*: το πιστοποιητικό είναι το *x*

IF ∈ co-*NP*: όλοι οι πρώτοι παράγοντες του *m* μαζί με ένα πιστοποιητικό για τον καθένα ότι είναι πρώτοι

P vs. **NP**
Ή Τρόπος vs Κόπος

- Το *P* είναι υποσύνολο του *NP*
- Είναι *P=NP*? Αν ναι, τότε κάθε απλοϊκό πρόβλημα εξαντλητικής αναζήτησης θα είχε ένα πολυωνυμικού χρόνου αλγόριθμο



$$NP \subseteq EXPTIME = \bigcup_k TIME\left(2^{n^k}\right)$$

Clay Mathematics Institute

MILLENNIUM PRIZE PROBLEMS

P versus NP

The Hodge Conjecture

The Poincaré Conjecture

The Riemann Hypothesis

Yang-Mills Existence and Mass Gap

Navier-Stokes Existence and Smoothness

The Birch and Swinnerton-Dyer Conjecture

Λύθηκε (Perelman)!

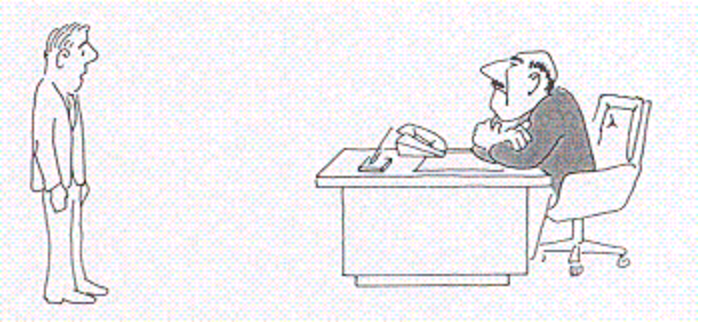
Announced 16:00, on Wednesday, May 24, 2000
Collège de France

P vs. **NP** και Μαθηματικά

- Αν *P=NP*, τότε θα αντικαθιστούσαμε τους μαθηματικούς από (πολύ πιο αξιόπιστους) υπολογιστές:

> *P=NP*: Υπάρχει αλγοριθμική διαδικασία που παίρνει ως είσοδο οποιαδήποτε τυπική μαθηματική πρόταση και πάντα παράγει την μικρότερη δυνατή απόδειξη σε χρόνο ανάλογο με το μήκος της απόδειξης.

- Αυτός είναι ένα λόγος που συνήθως θεωρούμε (ιδιαίτερα οι μαθηματικοί!) ότι οι κλάσεις *P* και *NP* είναι διαφορετικές.

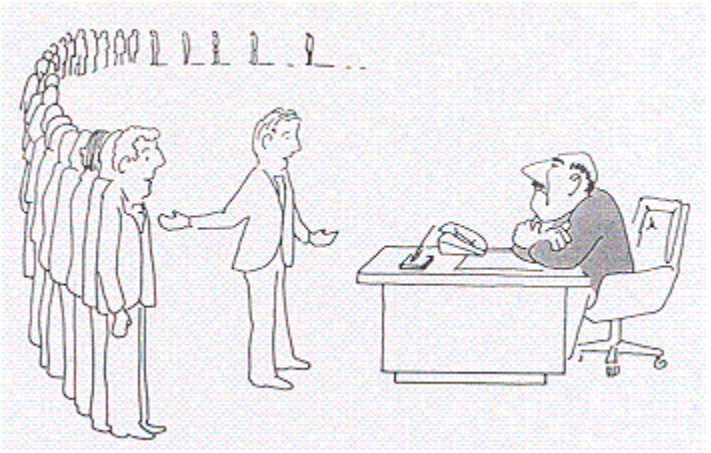


I can’t find an efficient algorithm, I guess I am just too dumb

Πολυπλοκότητα

NP-Πληρότητα

I can’t find an efficient algorithm, because no such algorithm is possible



I can’t find an efficient algorithm, but neither can all these famous people

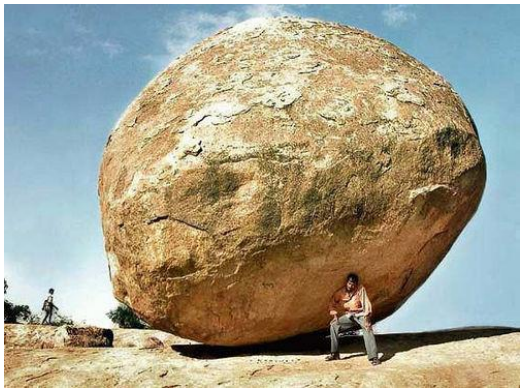
P

- Μερικά προβλήματα είναι αποδεδειγμένα επιλύσιμα σε πολυωνυμικό χρόνο σε έναν κανονικό υπολογιστή.
  - Αυτά τα προβλήματα ανήκουν στην κλάση ***P***
  - Στην ουσία είναι ένας Η/Υ με άπειρη μνήμη
  - *Πώς αποδεικνύουμε ότι ένα πρόβλημα είναι στο* ***P****?*

NP

- Μερικά προβλήματα είναι αποδεδειγμένα επιλύσιμα σε πολυωνυμικό χρόνο σε έναν ανταιτιοκρατικό Η/Υ
  - Αυτά τα προβλήματα ανήκουν στην κλάση ***NP***
  - Μπορούμε να φανταστούμε έναν ανταιτιοκρατικό Η/Υ σαν μία παράλληλη μηχανή που μπορεί να τρέχει παράλληλα άπειρες διεργασίες
  - *Πώς αποδεικνύουμε ότι ένα πρόβλημα είναι στο* ***NP;***

Η Έννοια της Αναγωγής



Το πρόβλημα της μετακίνησης βράχου.

Το πρόβλημα της εύρεσης μοχλού.

Man lifting a stone with a lever

Δυνατότητα επίλυσης μετακίνησης δεδομένης της εύρεσης μοχλού

Άλλο Παράδειγμα

- Το ταξίδι από την Αθήνα στο Παρίσι …
- *ανάγεται* στην αγορά αεροπορικού εισιτηρίου …

- το οποίο *ανάγεται* στην εύρεση χρημάτων για την αγορά του εισιτηρίου …

- το οποίο *ανάγεται* στην εύρεση εργασίας

- (ή τα πληρώνει ο μπαμπάς και η μαμά…)

Ακόμα Ένα Παράδειγμα (από την ανάποδη όμως)

Θέλουμε να μιλήσουμε με τον *Μήτσο* τον ντουντούκαρο αλλά δεν έχουμε το τηλ. του. Το έχει όμως ο *Μπάμπης* ο σουγιάς.

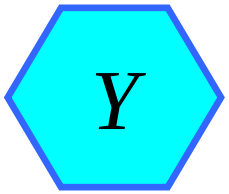
(εύρεση τηλ. *Μήτσου* ***ανάγεται*** στην επικοινωνία με τον *Μπάμπη*)

Αν όμως κάποιος μάντης μας έλεγε ότι δεν μπορούμε με τίποτα να βρούμε το τηλ. του *Μήτσου* τότε θα μπορούσαμε να επικοινωνήσουμε με τον *Μπάμπη*;

Απόδειξη με Αναγωγή

A ανάγεται στο *B*

1. Ξέρουμε ότι ο αποδοτικός *X* που λύνει το *A* δεν υπάρχει.

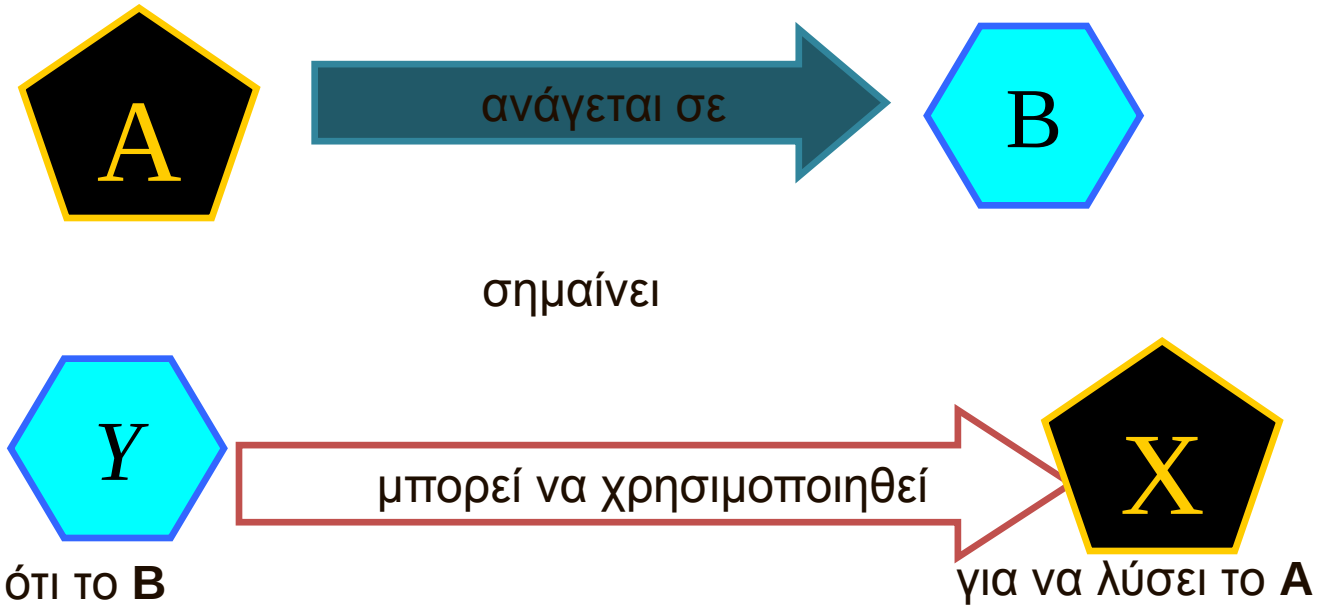


2. Έστω ότι ο *Y* υπάρχει.
(*Y* = ένας αποδοτικός αλγόριθμος που λύνει το *B*)

3. Δείξτε πως ο *Y* συνεπάγεται τον *X*.

4. Αφού ο *X* δεν υπάρχει, αλλά ο *Y* θα μπορούσε να χρησιμοποιηθεί για τον *X*, τότε ο *Y* δεν μπορεί να υπάρχει.

Αποδείξεις με Αναγωγή



Άρα, το **Α δεν είναι πιο δύσκολο πρόβλημα από το Β.**

Ο όρος “ανάγεται” είναι λίγο περίεργος: λειτουργεί προς την αντίστροφη φορά από την λύση.

Αναγωγιμότητα Πολυωνυμικού Χρόνου

Ορισμός: Μία συνάρτηση

$$f : \Sigma^* \to \Sigma^*$$

Είναι υπολογίσιμη σε πολυωνυμικό χρόνο αν υπάρχει αιτιοκρατικός αλγόριθμος που

- ξεκινά με είσοδο w, και
- τερματίζει έπειτα από ***πολυωνυμικό*** πλήθος βημάτων με έξοδο $f(w)$.

Πολυωνυμικού Χρόνου Αναγωγή

Ορισμός: Ένα πρόβλημα A είναι αναγώγιμο σε πολυωνυμικό χρόνο στο B:

$$A \leq_P B$$

αν υπάρχει συνάρτηση πολυωνυμικού χρόνου

$$f : \Sigma^* \to \Sigma^*$$

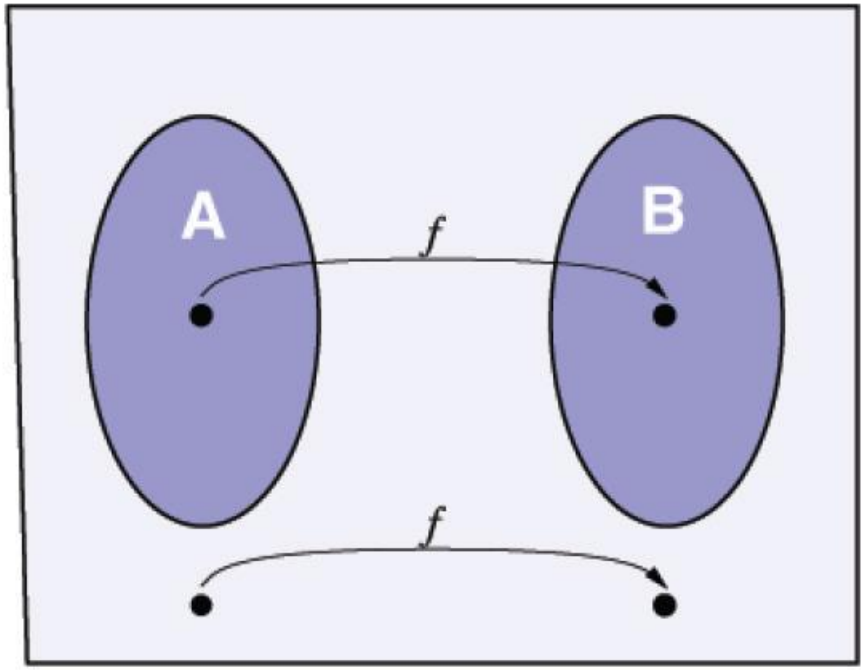
έτσι ώστε για κάθε w,

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

Η συνάρτηση f καλείται πολυωνυμικού χρόνου αναγωγή της A στο B.

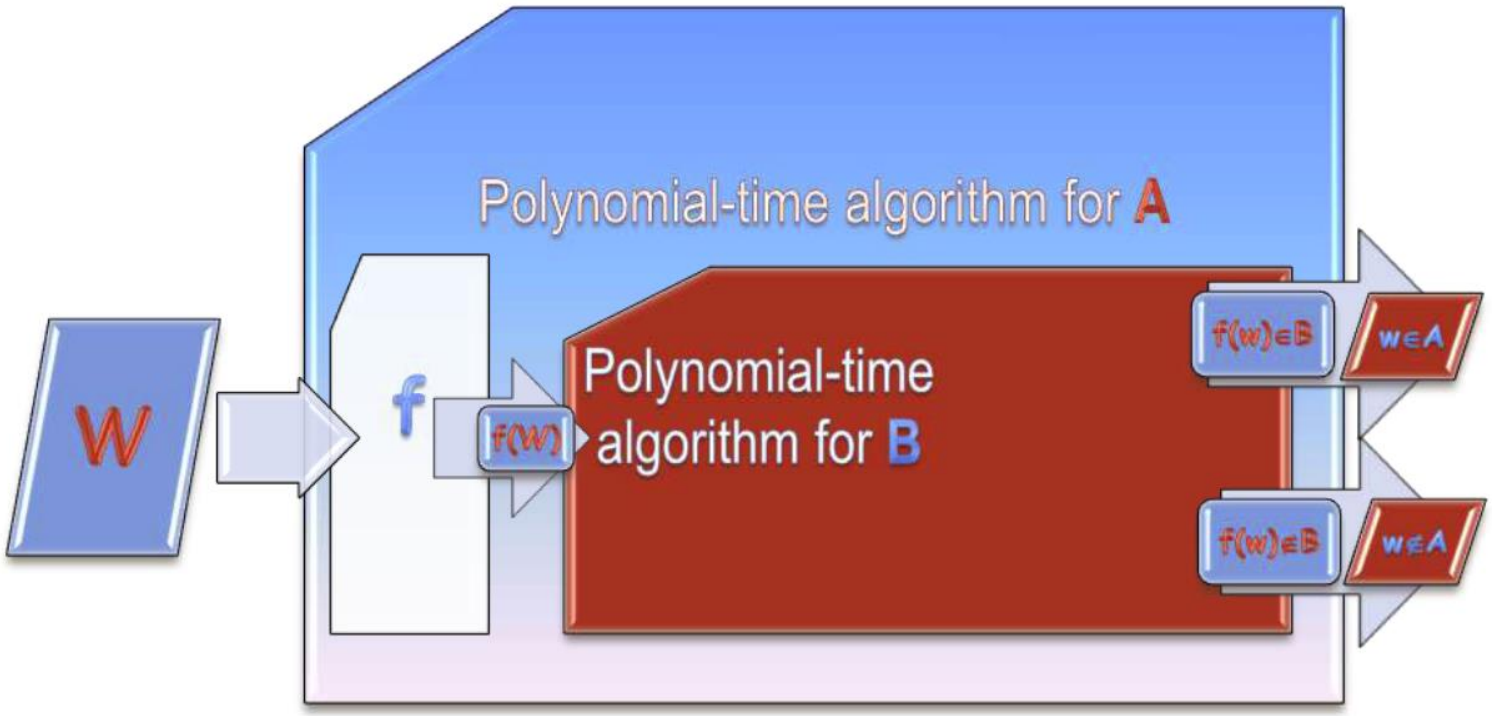
Πολυωνυμικού Χρόνου Αναγωγή

Μετατρέπει ερωτήσεις συμμετοχής στο *A* σε ερωτήσεις συμμετοχής στο *B*, με ***αποδοτικό*** τρόπο.



Αναγωγή για την Κλάση *P*

Θεώρημα: Αν $A \leq_p B$ και $B \in P$ τότε $A \in P$.



Αληθευσιμότητα

Μία λογική μεταβλητή παίρνει τιμές:

- ΑΛΗΘΕΣ (1), και ΨΕΥΔΕΣ (0).

Λογικές Πράξεις:

- ΚΑΙ: $\wedge$
- Η: $\vee$
- ΟΧΙ: $\neg$

Παραδείγματα:

- $0 \wedge 1 = 0$
- $0 \vee 1 = 1$
- $\neg 0 = 1$

Αληθευσιμότητα

Ένας λογικός τύπος είναι μία έκφραση των λογικών μεταβλητών και πράξεων.

$$\varphi = (x \wedge y) \vee (x \wedge z)$$

Ορισμός: Ένας λογικός τύπος είναι αληθεύσιμος αν υπάρχει κάποιος συνδυασμός από 0 και 1 έτσι ώστε η έκφραση να είναι 1.

Αληθευσιμότητα

$\varphi = (x \wedge y) \vee (x \wedge \neg z)$

Είναι αληθεύσιμος από την τιμοδοσία:

- $x = 0$
- $y = 1$
- $z = 0$

Αυτή η τιμοδοσία είναι αληθοποιός για την έκφραση φ.

Το Πρόβλημα της Αληθευσιμότητας

SAT = {$\langle\varphi\rangle$ | φ είναι ένας αληθεύσιμος λογικός τύπος}

Ασχολούμαστε με συγκεκριμένη μορφή:

- Ένα λεξίγραμμα είναι μία μεταβλητή ή η συμπληρωματική της: x ή $\neg x$.
- Μία φράση είναι λεξιγράμματα που συνδέονται με διάζευξη (V): $(x_1 \vee x_2 \vee x_3)$
- Ένας λογικός τύπος είναι σε Κανονική Συζευκτική Μορφή (CNF) αν αποτελείται από φράσεις συνδεόμενες με συζεύξεις ($\wedge$).
- Παράδειγμα: $(x_1 \vee x_2 \vee x_3 \vee x_4) \wedge (x_3 \vee x_5 \vee x_6) \wedge (x_3 \vee x_6)$

Αληθευσιμότητα

Ορισμός: Ένας λογικός τύπος είναι σε μορφή $_3$CNF αν είναι CNF μορφή, και όλες οι φράσεις έχουν ακριβώς 3 λεξιγράμματα.

$$(x_1 \vee x_2 \vee x_3) \wedge (x_3 \vee x_5 \vee x_6) \wedge (x_3 \vee x_6 \vee x_4)$$

$_3$SAT = $\{\langle \varphi \rangle \mid \varphi$ είναι αληθεύσιμος λογικός τύπος $_3$CNF$\}$

Αν ο φ είναι αληθεύσιμος $_3$CNF τύπος, τότε για κάθε τέτοια τιμοδοσία του φ, κάθε φράση θα περιέχει τουλάχιστον ένα λεξίγραμμα που είναι 1.

Αναγωγή

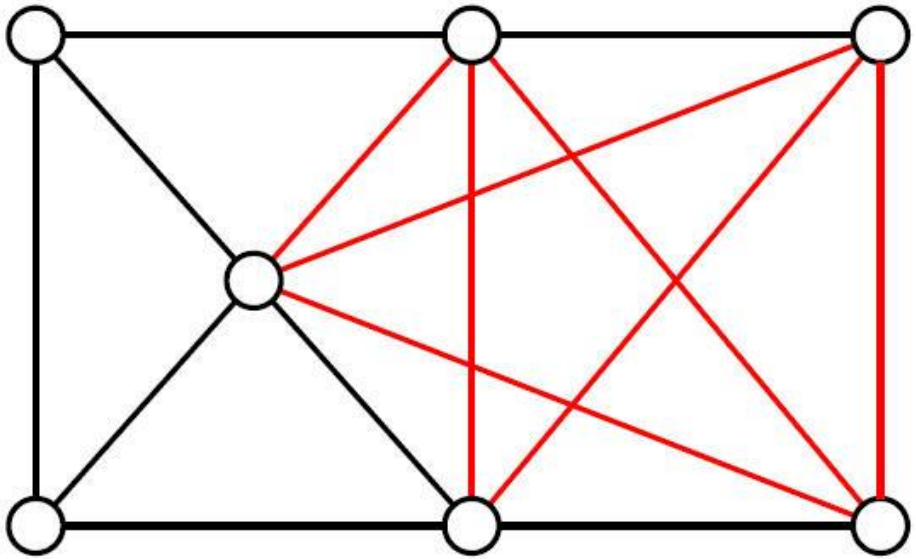
Ισχυρισμός: Υπάρχει πολυωνυμική αναγωγή από το $_3$SAT στην ΚΛΙΚΑ. Με άλλα λόγια:

$$_3\text{SAT} \leq_P \text{CLIQUE}$$

Θα κατασκευάσουμε μία πολυωνυμική αναγωγή f που απεικονίζει έναν $_3$CNF λογικό τύπο φ σε ένα γράφημα G και έναν αριθμό k.

Η συνάρτηση f έχει τη ιδιότητα ότι η φ είναι αληθεύσιμη αν και μόνο αν το γράφημα G έχει κλίκα μεγέθους k.

Παράδειγμα: Κλίκα



Η κλίκα σε ένα γράφημα είναι ένα υπογράφημα όπου μεταξύ κάθε ζευγαριού κόμβων υπάρχει μία ακμή.

Μία *k*-κλίκα είναι μία κλίκα μεγέθους *k*. Για παράδειγμα, το γράφημα παραπάνω έχει μία 5-κλίκα.

${}_3\text{SAT} \leq_P \text{ΚΛΙΚΑ}$

Έστω φ ένας ${}_3\text{CNF}$ λογικός τύπος με k φράσεις.

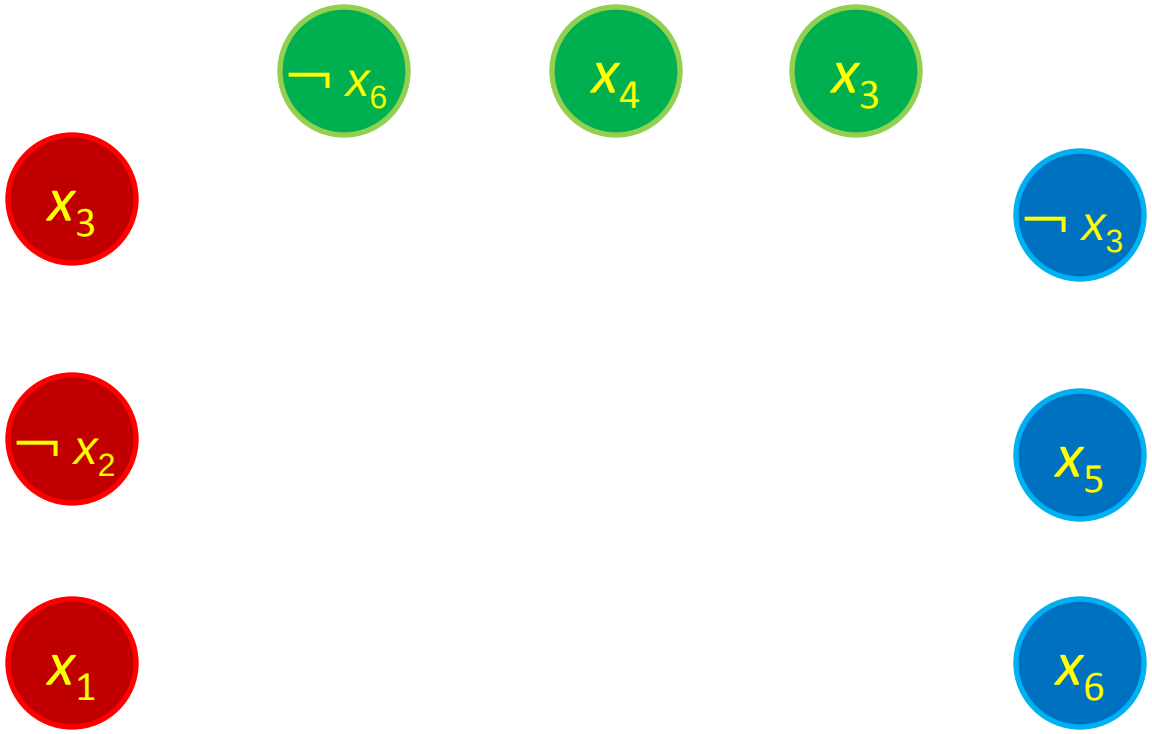
$(x_1 \vee \neg x_2 \vee x_3) \wedge (\neg x_3 \vee x_5 \vee x_6) \wedge (x_3 \vee \neg x_6 \vee x_4)$

Ορίζουμε το γράφημα ως εξής:

- Οι κόμβοι του G οργανώνονται σε τριάδες $t_1, \ldots, t_k$.
- Κάθε τριάδα αντιστοιχεί σε μία φράση
- Κάθε κόμβος σε μία τριάδα αντιστοιχεί σε ένα λεξίγραμμα.

Παράδειγμα

$(x_1 \vee \neg x_2 \vee x_3) \wedge (\neg x_3 \vee x_5 \vee x_6) \wedge (x_3 \vee \neg x_6 \vee x_4)$

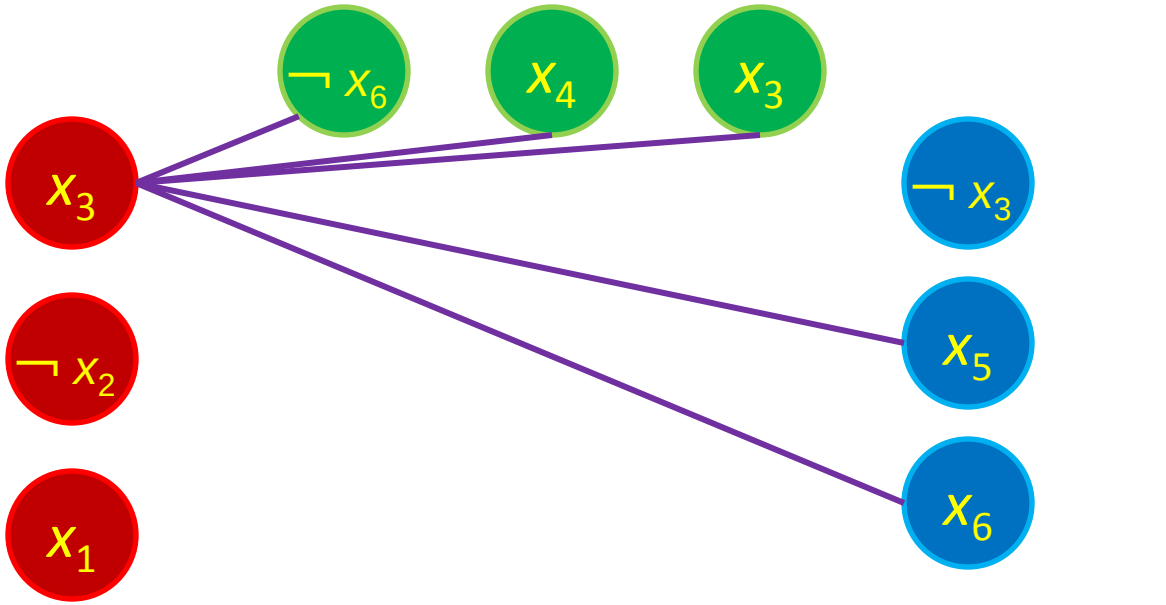


${}_3\text{SAT} \leq_P$ ΚΛΙΚΑ

Πρόσθεσε ακμές μεταξύ όλων των ζευγαριών κορυφών εκτός:

- Αν ανήκουν στην ίδια τριάδα
- Μεταξύ αντίθετων λεξιγραμμάτων

$(x_1 \vee \neg x_2 \vee x_3) \wedge (\neg x_3 \vee x_5 \vee x_6) \wedge (x_3 \vee \neg x_6 \vee x_4)$



$_3$SAT $\leq_P$ ΚΛΙΚΑ

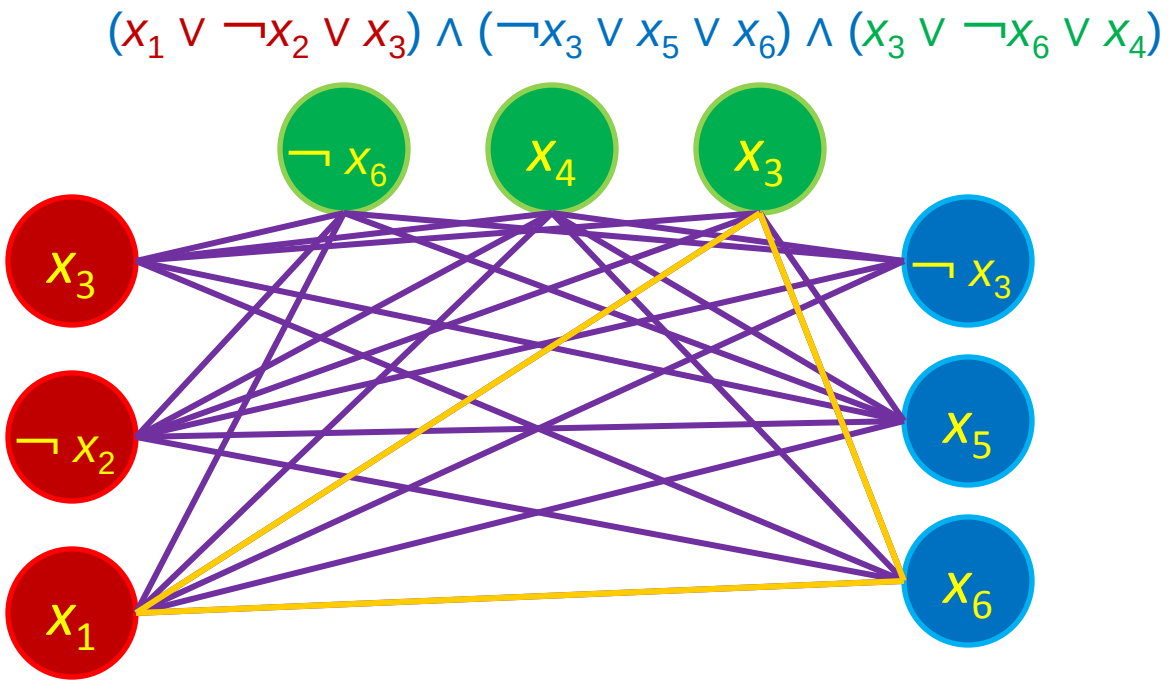
Αν η φ είναι αληθεύσιμη, τότε η G έχει μία k-κλίκα.

Έστω ότι η φ είναι αληθεύσιμη.

- Τουλάχιστον ένα λεξίγραμμα είναι ΑΛΗΘΕΣ σε κάθε φράση
- σε κάθε τριάδα επέλεξε ένα κόμβο με ΑΛΗΘΕΣ λεξίγραμμα
- Αυτές συνδέονται με ακμές που αποδίδουν μία k-κλίκα

${}_3\text{SAT} \leq_P \text{ΚΛΙΚΑ}$

Αν η φ είναι αληθεύσιμη τότε ο G έχει k-κλίκα.



$_3$SAT $\leq_P$ ΚΛΙΚΑ

Αν η *G* έχει μία *k*-κλίκα, η *φ* είναι αληθεύσιμη.

- Κανένα ζευγάρι κόμβων δεν είναι στην ίδια τριάδα.
- Έχει *k* κορυφές και *k* φράσεις, και άρα
- κάθε τριάδα έχει ακριβώς ένα κόμβο κλίκας
- Δίνουμε 1 σε κάθε κόμβο της κλίκας
- Δεν υπάρχει αντίφαση

${}_3\text{SAT} \leq_P \text{ΚΛΙΚΑ}$

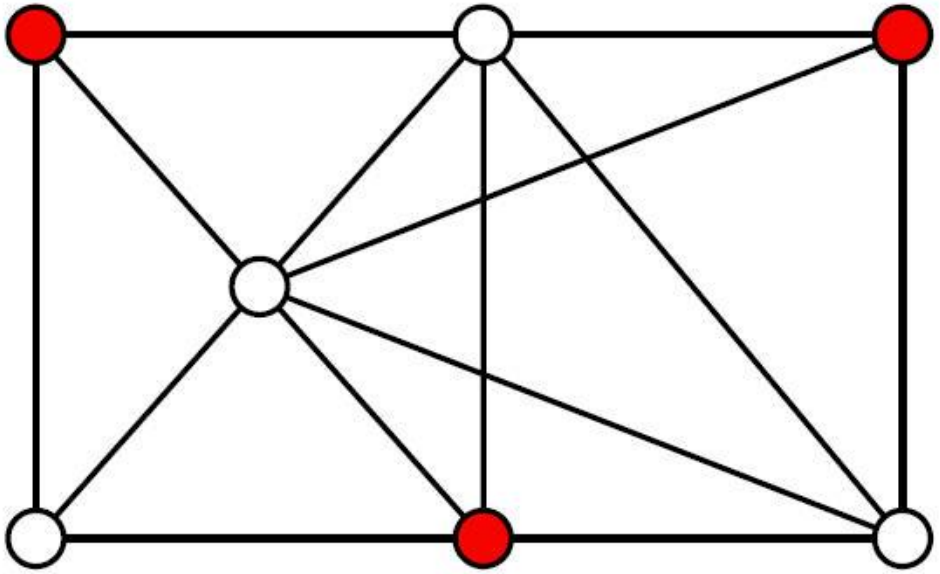
- Κατασκευάσαμε μία υπολογίσιμη σε πολυωνυμικό χρόνο συνάρτηση f.
- Δείξαμε ότι η συνάρτηση f έχει την ιδιότητα ότι ο $\varphi \in {}_3\text{SAT}$ αν και μόνο αν $f(\varphi) \in \text{ΚΛΙΚΑ}$.
- Άρα η f είναι μία αναγωγή από το ${}_3\text{SAT}$ στην ΚΛΙΚΑ και άρα ${}_3\text{SAT} \leq_P \text{ΚΛΙΚΑ}$

Αναγωγές

- Σκέψη για κατανόηση της δομής και των δύο προβλημάτων
- Εύρεση «εξαρτήματος» που εκμεταλλεύεται αυτή τη δομή (δομές που μπορούν να προσομοιώσουν τις φράσεις και μεταβλητές του $_3$SAT)
- Γενικά είναι δύσκολα προβλήματα

Ανεξάρτητο Σύνολο

- Ένα ανεξάρτητο σύνολο σε ένα γράφημα είναι ένα σύνολο κορυφών έτσι ώστε κανένα ζευγάρι κορυφών να μην είναι γειτονικές
- Υπάρχει ανεξάρτητο σύνολο μεγέθους k;



Ανεξάρτητο Σύνολο

ΑΝΕΞΑΡΤΗΤΟ_ΣΥΝ = {⟨*G*,*k*⟩ | *G* περιέχει ανεξάρτητο σύνολο μεγέθους *k*}

Το ΑΝΕΞΑΡΤΗΤΟ_ΣΥΝ είναι πολυωνυμικά αναγώγιμο στην ΚΛΙΚΑ

ΑΝΕΞΑΡΤΗΤΟ-ΣΥΝ $\leq_P$ ΚΛΙΚΑ

και αντίστροφα :

ΚΛΙΚΑ $\leq_P$ ΑΝΕΞΑΡΤΗΤΟ-ΣΥΝ

Ισχύει πάντα αυτό;

Ανεξάρτητο Σύνολο

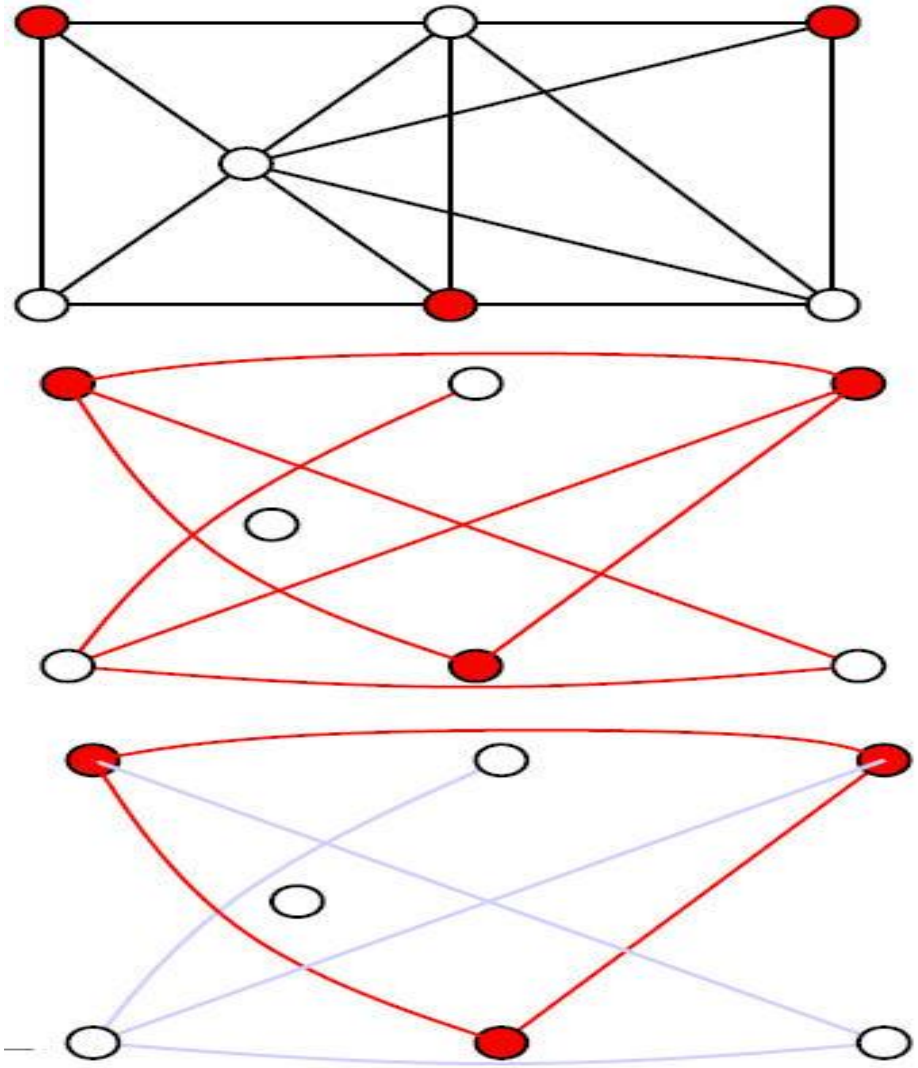
Το συμπλήρωμα του γραφήματος $G = (V,E)$ είναι ένα γράφημα:

$G^c = (V,E^c)$, όπου:

$$E^c = \{(v_1, v_2) \mid v_1, v_2 \in V \text{ και } (v_1, v_2) \notin E\}$$

Αν το V είναι ένα ανεξάρτητο σύνολο στο G, τότε ο V είναι κλίκα στο G^c. (και αντίστροφα)

Παράδειγμα



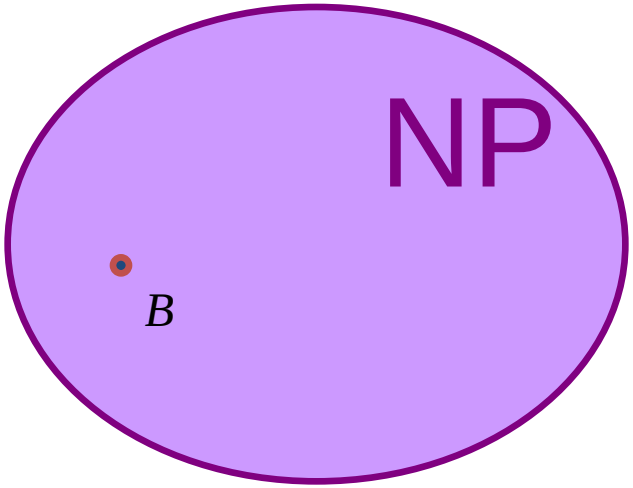
Cook–Levin (1971-1973)

Θεώρημα: Υπάρχει γλώσσα $S \in NP$ έτσι ώστε $S \in P$ αν και μόνο αν $P = NP$.

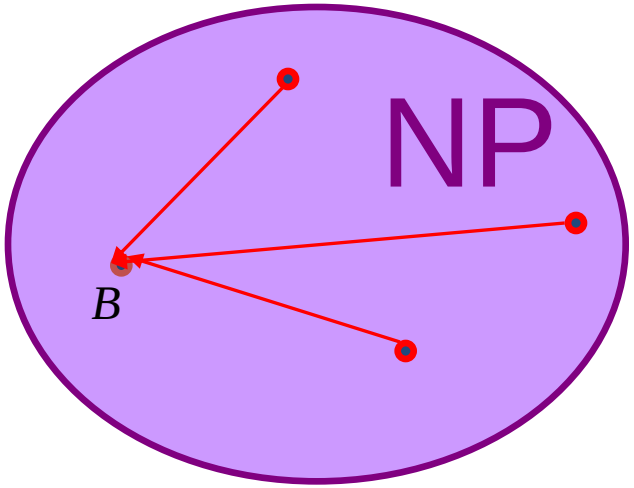
- Αυτό το θεώρημα καθορίζει την κλάση των *NP*-πλήρων γλωσσών.
- Αυτές οι γλώσσες, όπως και ο Frodo Baggins, «κουβαλάνε στην πλάτη τους» το βάρος όλης κλάσης *NP*.

NP-Πληρότητα

Ένα πρόβλημα *B* είναι NP-πλήρες αν:

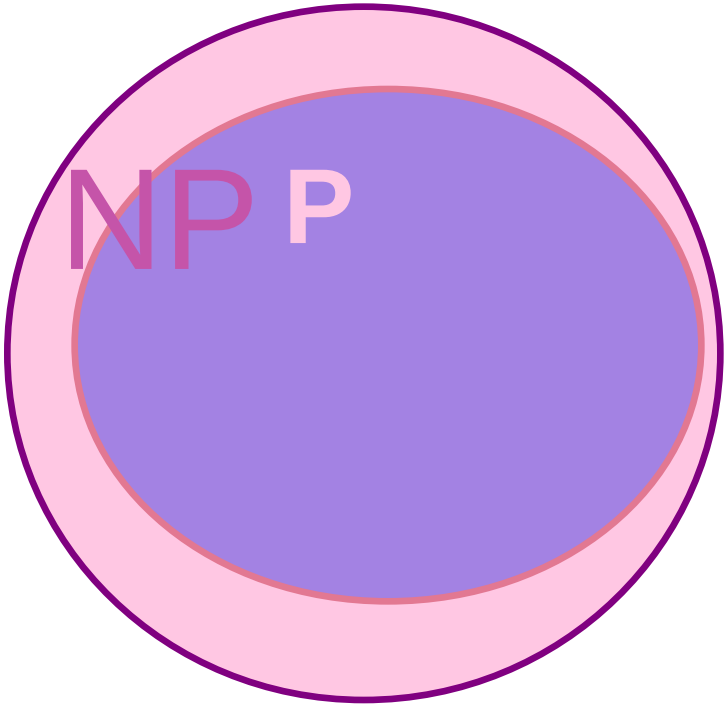


1. $B \in \mathbf{NP}$

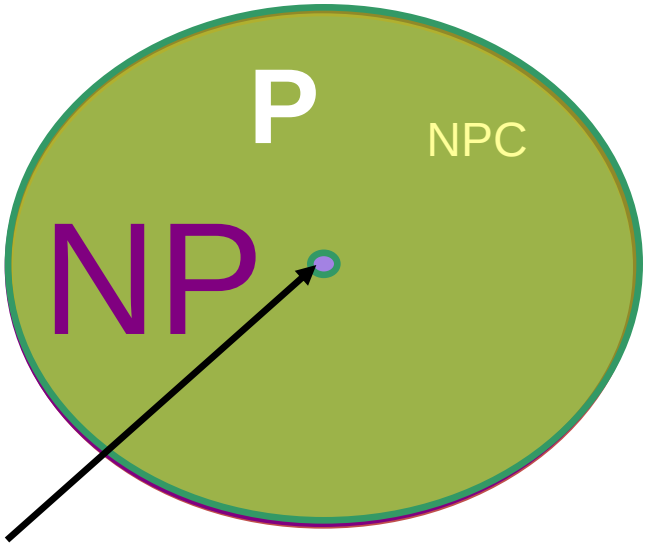


2. Υπάρχει πολυωνυμικού χρόνου αναγωγή από κάθε πρόβλημα $A \in \mathbf{NP}$ στο B.

NP-Πληρότητα



Περίπτωση 1: **P** $\subset$ **NP**

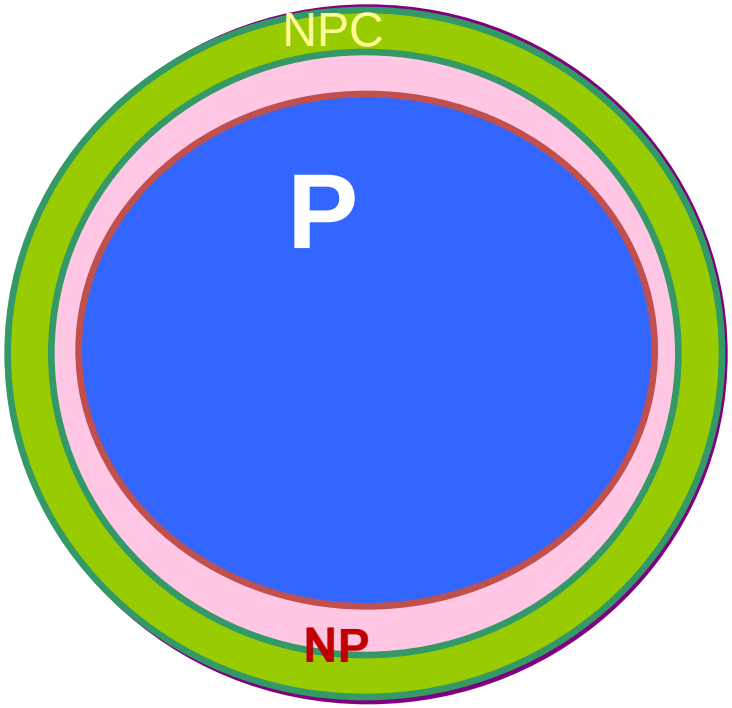


Τετριμμένα Προβλήματα
A = {}; A = Σ*

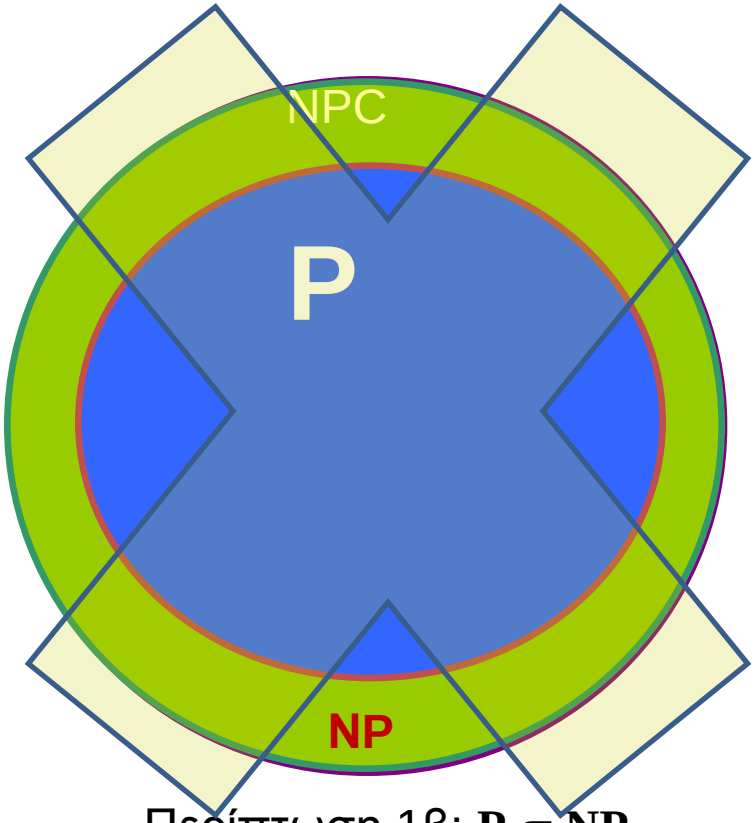
Περίπτωση 2: **P** = **NP**

= **NP**-Πλήρη ∪ Τετριμμένα

NP-Πληρότητα



Περίπτωση 1α: **P ⊂ NP,**
NPC ∪ P ⊂ NP

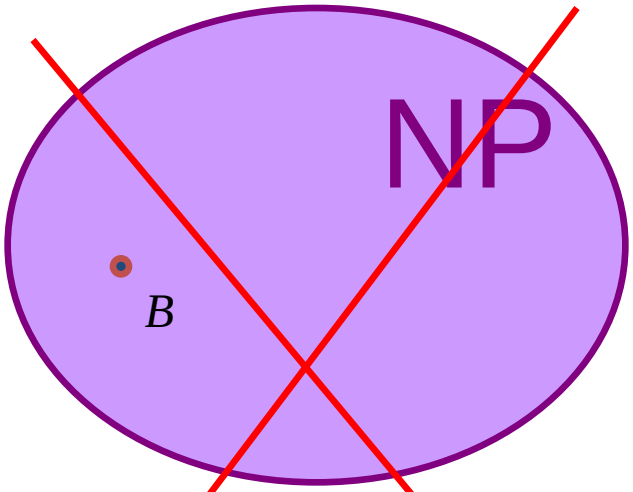


Περίπτωση 1β: **P ⊂ NP,**
NPC ∪ P = NP

Θεώρημα Ladner

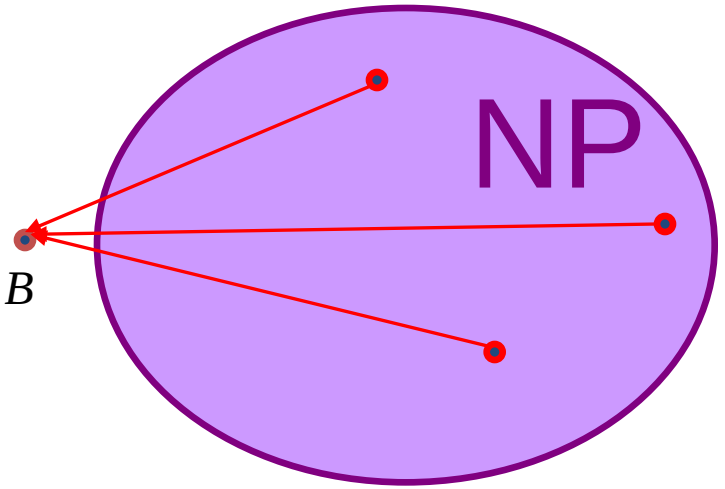
NP-~~Πλήρες~~ Δυσχερής (Hard)

Μία γλώσσα B είναι στην NPC αν:



1. $B \in \mathbf{NP}$

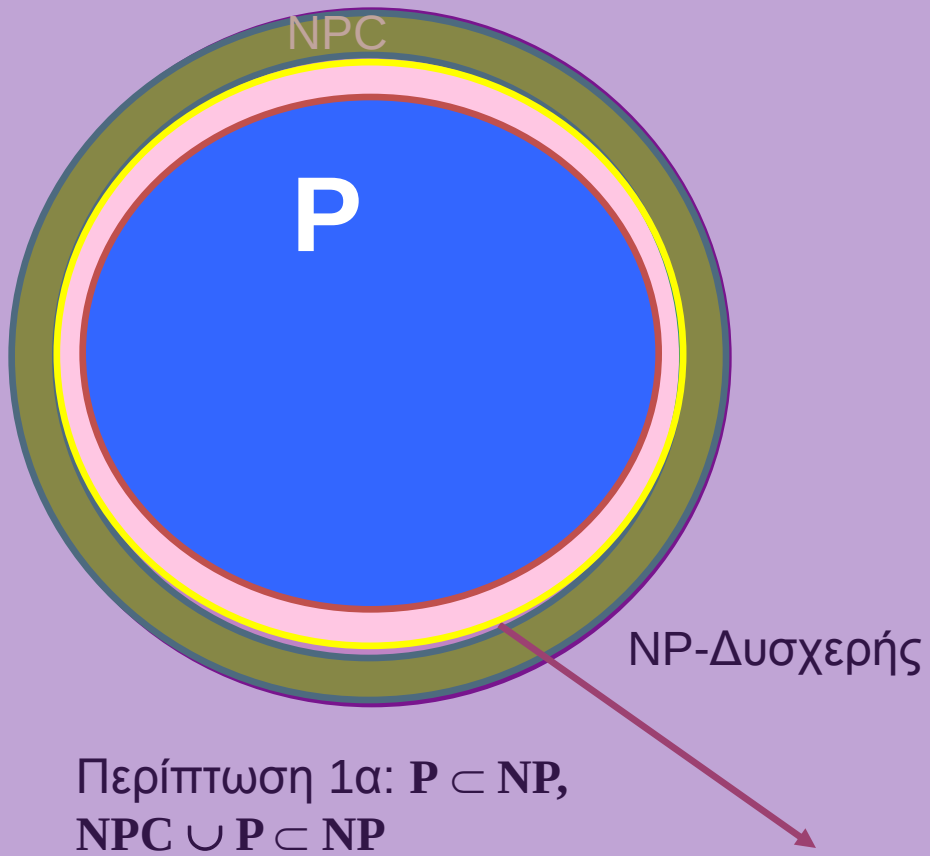
Όχι *απαραίτητο* για τα NP-δυσχερή



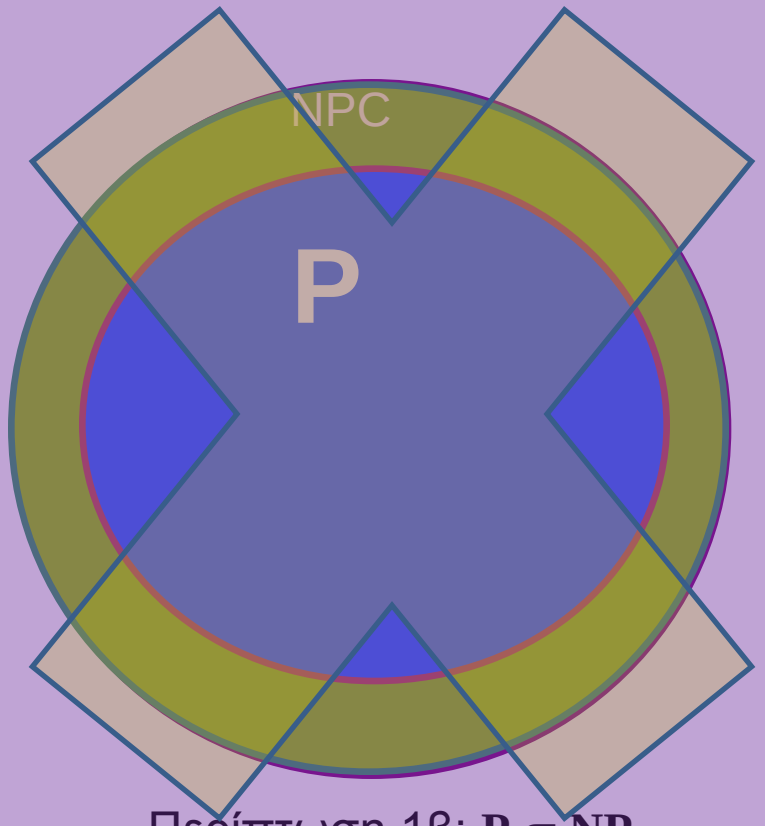
2. Υπάρχει πολυωνυμικού χρόνου αναγωγή από κάθε πρόβλημα $A \in \mathbf{NP}$ στο B.

Πώς μοιάζει η κλάση των NP-δυσχερών προβλημάτων;

NP-Δυσχερής (αν $\mathbf{P} \subset \mathbf{NP}$)

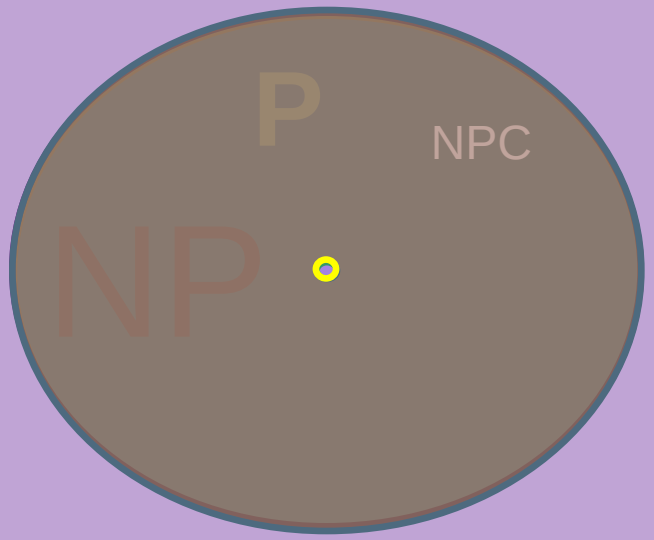


Περίπτωση 1α: $\mathbf{P} \subset \mathbf{NP}$, $\mathbf{NPC} \cup \mathbf{P} \subset \mathbf{NP}$



Περίπτωση 1β: $\mathbf{P} \subset \mathbf{NP}$, $\mathbf{NPC} \cup \mathbf{P} = \mathbf{NP}$

NP-Δυσχερής (αν **P** = **NP**)



Περίπτωση 2: **P** = **NP**

≈ **NP**-Πλήρες

NP-Δυσχερή = Όλα τα προβλήματα – {A = {}; A = Σ*}

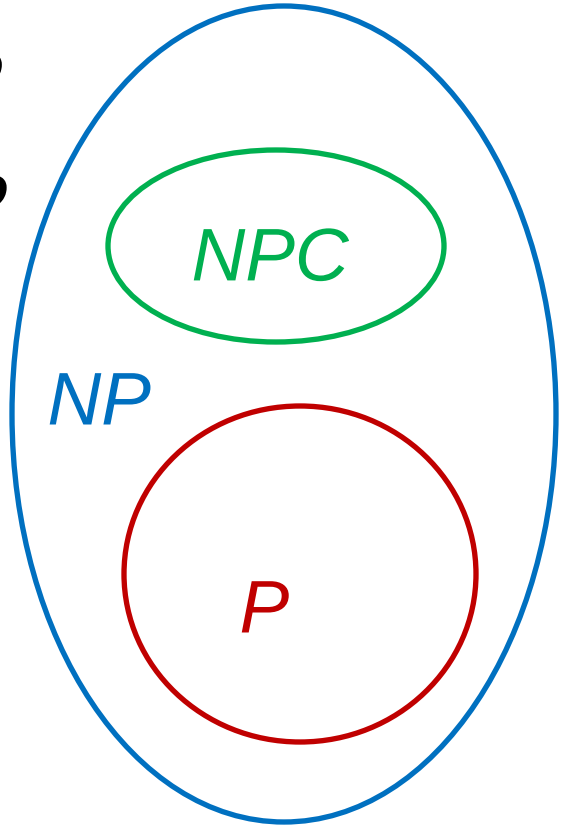
NP-Δυσχερής

- Αν P = NP:
  - Για να δείξετε ότι ένα πρόβλημα είναι NP-δυσχερές: δείξτε ότι για κάποια είσοδο δίνει ΝΑΙ και για κάποια ΟΧΙ
- Αν $P \subset NP$:
  - Για να δείξετε ότι ένα πρόβλημα είναι NP-δυσχερές: δείξτε ότι υπάρχει πολυωνυμική αναγωγή από κάποιο NP-πλήρες πρόβλημα σε αυτό
  - Αυτό σημαίνει ότι το πρόβλημα μάλλον δεν επιδέχεται πολυωνυμικής λύσης

NP Πληρότητα

Η κλάση των NP-πλήρων προβλημάτων είναι:

- «δυσκολότερες» γλώσσες στην *NP*
- «λιγότερο πιθανό» να είναι στην *P*
- Αν κάποιο NP-πλήρες $A \in P$, τότε $NP=P$.



Θεώρημα Αναγωγής

Αν το B είναι NP-πλήρες, $C \in NP$, και $B \leq_P C$, τότε και το C είναι NP-πλήρες.

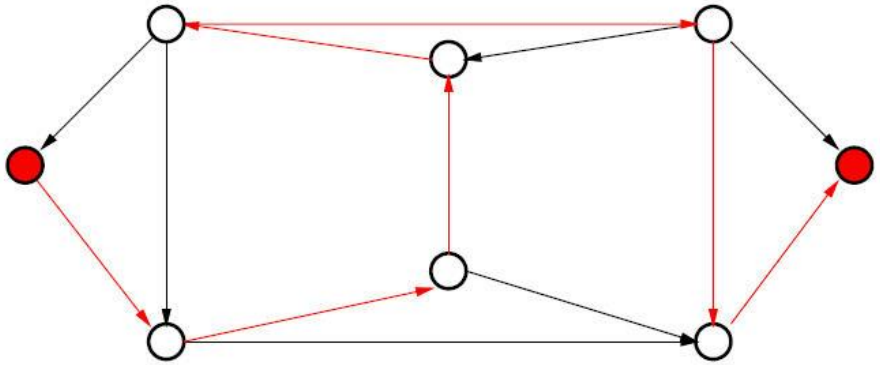
Γνωρίζουμε ότι $C \in NP$ – θα πρέπει να δείξουμε ότι κάθε πρόβλημα A στο NP είναι πολυωνυμικά αναγώγιμο στο C.

Αφού το B είναι NP-πλήρες, κάθε γλώσσα στην NP είναι πολυωνυμικά αναγώγιμη στη B. Επίσης η B είναι πολυωνυμικά αναγώγιμη στη C.

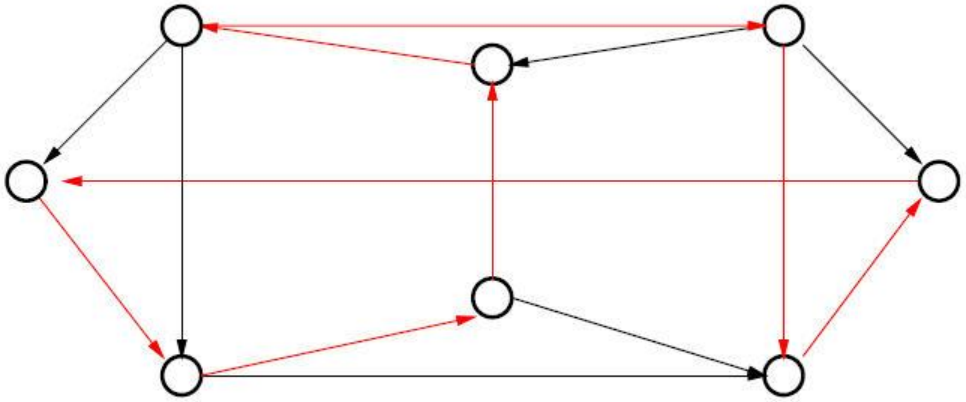
Άρα η A είναι πολυωνυμικά αναγώγιμη στη C.

Hamiltonian Μονοπάτι - Κύκλος

Ένα Hamiltonian μονοπάτι σε ένα κατευθυντό γράφημα *G* επισκέπτεται κάθε κόμβο μία φορά.



Ένας Hamiltonian κύκλος είναι ένα Hamiltonian μονοπάτι που η αρχή και το τέλος συμπίπτουν.



Hamiltonians

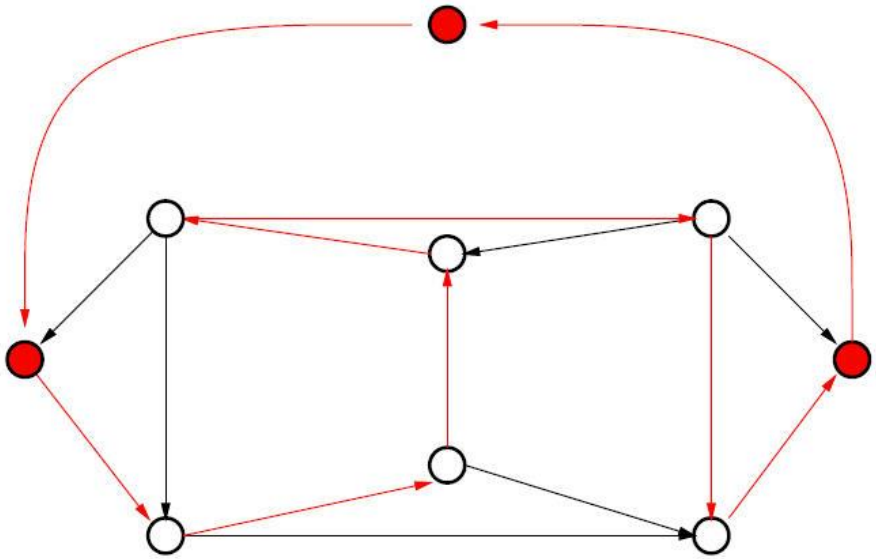
HAMPATH = {⟨*G*, *s*, *t*⟩ |ο *G* έχει Hamiltonian μονοπάτι από τον *s* στο *t*}

HAMCIRCUIT = {⟨*G*⟩ | ο *G* έχει Hamiltonian κύκλο}

Θεώρημα:

HAMPATH $\leq_P$ HAMCIRCUIT

HAMCIRCUIT $\leq_P$ HAMPATH



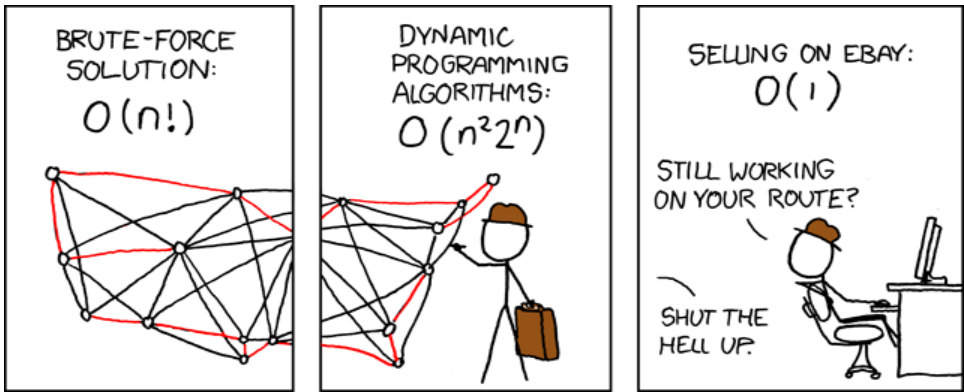
Πρόβλημα Περιοδεύοντος Πωλητή (TSP)

Παράμετροι:

- Σύνολο πόλεων *C*
- Σύνολο αποστάσεων μεταξύ πόλεων *D*
- Στόχος *k*

Τυπικά:

- Κατευθυντό γράφημα *G*=(*C*,*D*)
- Οι ακμές έχουν βάρη
- Πλήρες γράφημα *G*
- Στόχος *k*



Πρόβλημα Περιοδεύοντος Πωλητή (TSP)

TSP = $\{\langle C, D, k\rangle \mid$ το (C,D) έχει πορεία από όλες τις πόλεις συνολικής απόστασης $\leq k\}$

HAMCIRCUIT = $\{\langle G\rangle \mid$ ο G έχει ένα Hamiltonian κύκλο$\}$

$$\text{HAMCIRCUIT} \leq_P \text{TSP}$$

Αναγωγή: Δοθέντος ενός κατευθυντού γραφήματος $G=(V,E)$ κατασκευάζουμε στιγμιότυπο TSP. Οι πόλεις είναι ίδιες με τους κόμβους του G, $C = V$.
Η απόσταση από το v_1 στο v_2 είναι 1 αν $(v_1, v_2) \in E$, και 2 διαφορετικά.
Το φράγμα για την συνολική απόσταση είναι $k = |V|$.

HAMCIRCUIT $\leq_p$ TSP

Ορθότητα Αναγωγής

$\Rightarrow$ Έστω ότι ο G έχει ένα Hamiltonian κύκλο.

Η απόσταση που δίνεται λόγω της αναγωγής σε όλες τις πλευρές είναι 1. Άρα, στο (C,D) υπάρχει μία πορεία του πωλητή συνολικής απόστασης $|V|=k$.

$$(C,D,k) \in \text{TSP}$$

$\Leftarrow$ Έστω ότι το (C,D) έχει μία πορεία συνολικής απόστασης $|V|=k$.

Η πορεία δεν μπορεί να περιέχει καμία ακμή απόστασης 2. Άρα έχουμε ένα Hamiltonian κύκλο στο G.

Απόδοση: Η αναγωγή γίνεται σε τετραγωνικό χρόνο (αποστάσεις στο πλήρες γράφημα)

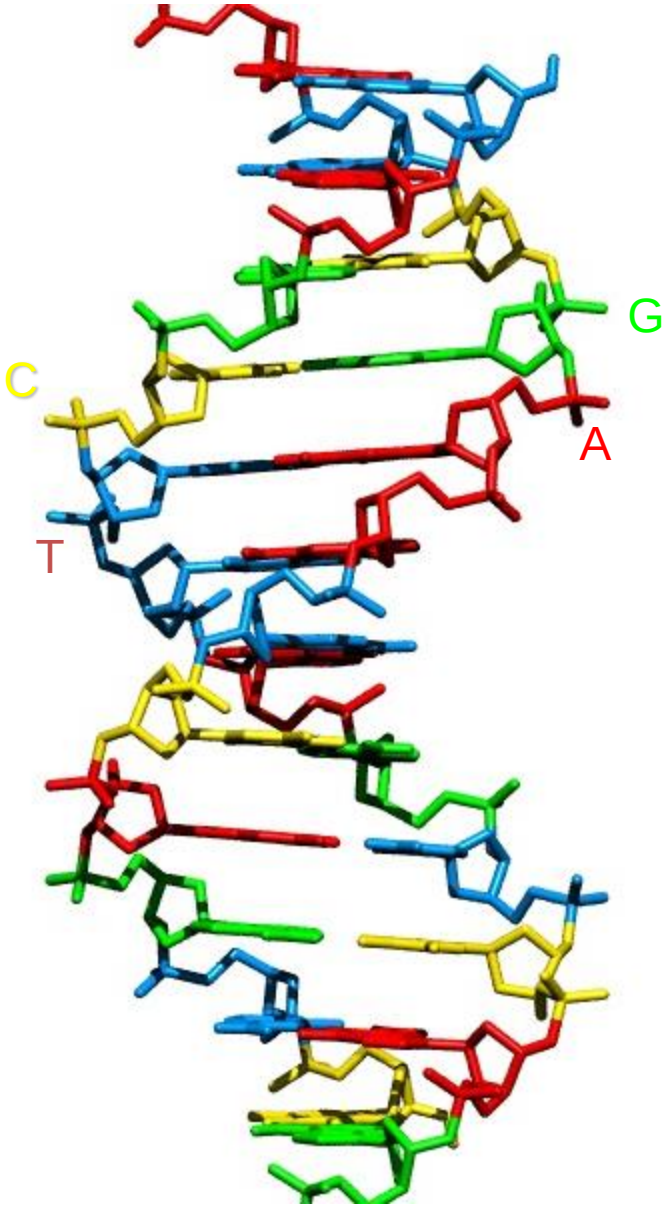
Στρατηγική

Από την στιγμή που θα έχουμε ένα «αποδεδειγμένο» NP-πλήρες πρόβλημα, μπορούμε να παράγουμε επιπλέον μέσω πολυωνυμικών αναγωγών.

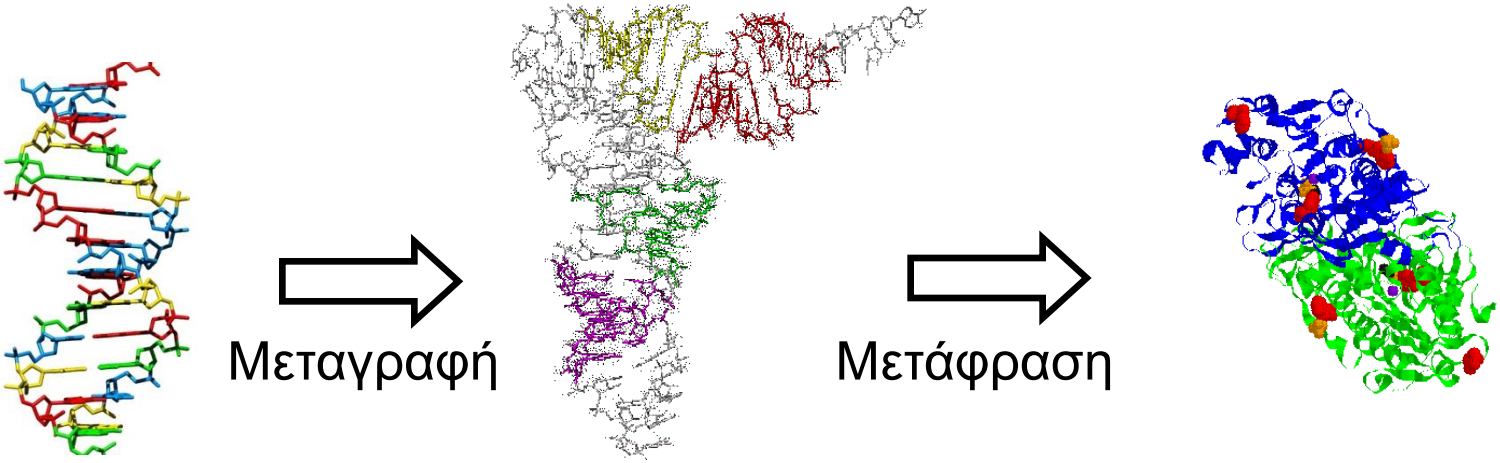
Το να φτιάξουμε όμως το πρώτο χρειάζεται αρκετή δουλειά. Αυτή τη δουλειά την έκαναν οι Steve Cook (τότε στο Berkeley, τώρα στο Τορόντο) και Leonid Levin (τότε στη Μόσχα, τώρα στην Βοστόνη) στις αρχές της δεκαετίας του 70.

DNA

- Ακολουθία νουκλεοτιδίων: αδενίνη (*A*), γουανίνη (*G*), κυτοσίνη (*C*), και θυμίνη (*T*)
- Μία έλικα, όπου η *A* συνδέεται με *T* και η *G* με *C*



Κεντρικό Δόγμα Βιολογίας



DNA RNA Πρωτεΐνη

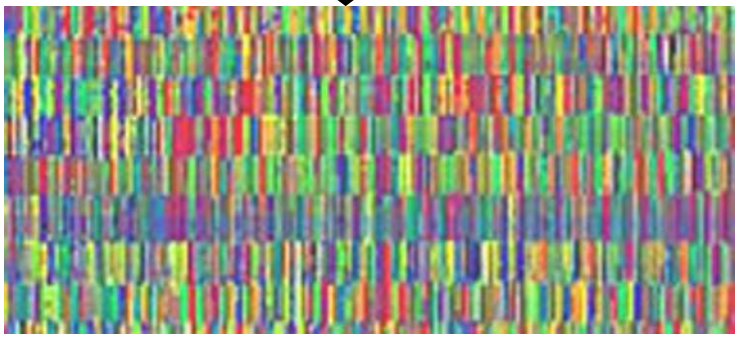
Εικόνα από http://www.umich.edu/~protein/

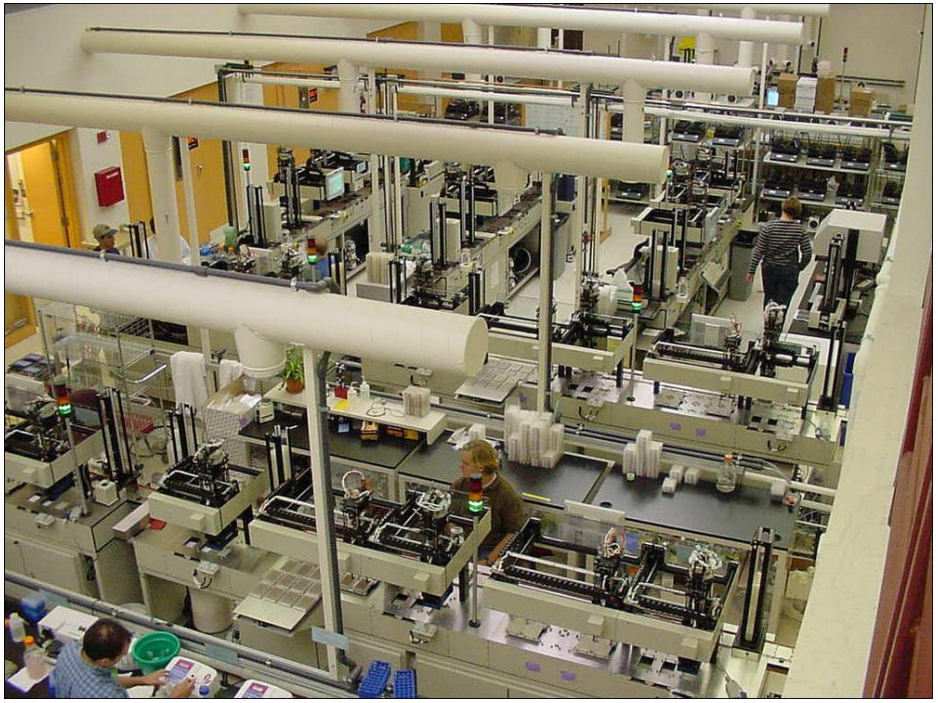
- Το RNA αντιγράφει κομμάτια του DNA
- Το RNA περιγράφει ακολουθίες αμινοξέων
- Αλυσίδες αμινοξέων κατασκευάζουν πρωτεΐνες
- Οι πρωτεΐνες φτιάχνουν εμάς

Ανθρώπινο Γονιδίωμα

- 3 Δις Ζευγάρια Βάσεων
  - Κάθε νουκλεοτίδιο είναι 2 bits (4 περιπτώσεις)
  - 3GB ζευγάρια * 1 byte/4 pairs = 750 MB
- Κάθε ακολουθία από 3 ζευγάρια βάσεων αντιστοιχεί σε 21 αμινοξέα
  - 21 πιθανά αμινοξέα, αλλά $4^3 = 64$ δυνατά
  - Άρα, μόνο 750MB * (21/64) ~ 250 MB
- Το περισσότερο από αυτό (> 95%) μάλλον είναι σκουπίδια (δεν αντιστοιχεί σε πρωτεΐνες αλλά μπορεί να έχει σημαντική λειτουργία)

Διαβάζοντας το Γονιδίωμα

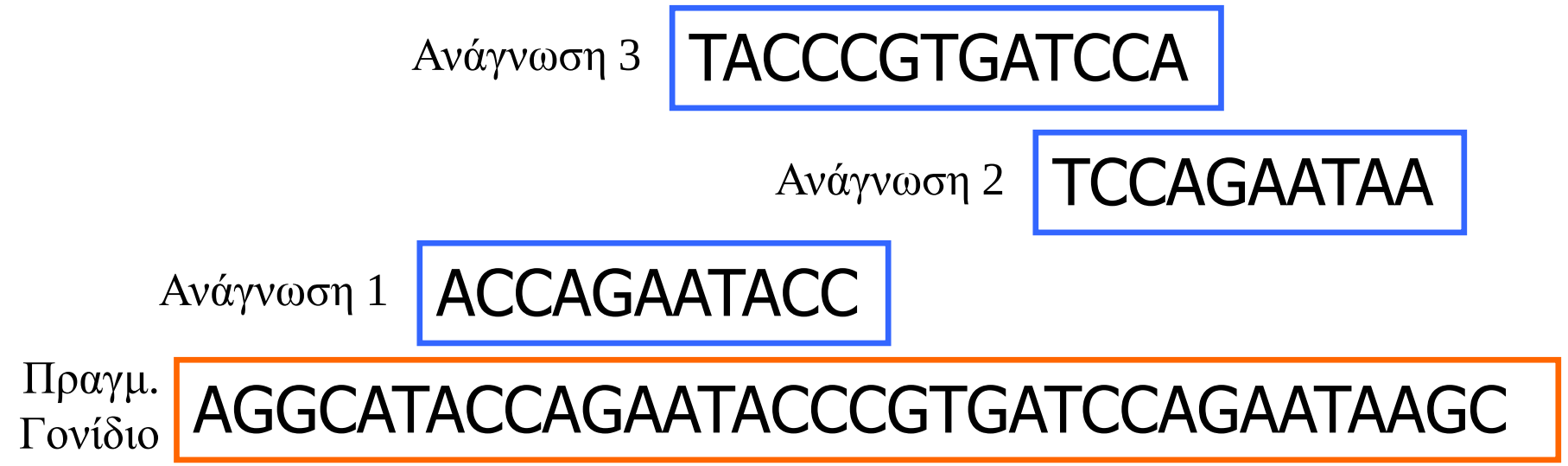




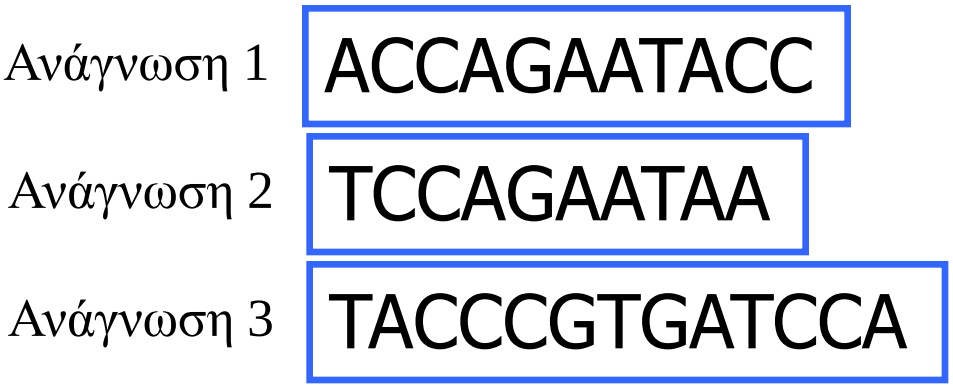
Whitehead Institute, MIT

Οι Μηχανές Ανάγνωσης

- Μία ανάγνωση: περίπου 700 ζευγάρια βάσεων
- Αλλά δεν ξέρουμε που είναι στο χρωμόσωμα!!!!



Το Πρόβλημα της Κατασκευής του Γονιδιώματος



Είσοδος: Κομμάτια του γονιδιώματος (χωρίς να ξέρουμε από που)

Έξοδος: Το πλήρες γονίδιο

Πρόβλημα Απόφασης

$GA = \{\langle \{ x_1, x_2, \ldots, x_n \}, m \rangle$ | όπου
κάθε x_i είναι μία λέξη
και υπάρχει μία λέξη X μήκους m
που περιλαμβάνει όλα τα x_i
σαν υπολέξεις$\}$

Αν λύναμε αυτό το πρόβλημα, μπορούμε να βρούμε το μήκος της μικρότερης υπερλέξης;

ΝΑΙ. Δοκίμασε όλες τις τιμές m από 1, 2, …, $\mathbf{\Sigma}|x_i|$.

Ανήκει το GA στο NP;

$GA = \{\langle \{ x_1, x_2, \ldots, x_n \}, m \rangle$ | όπου
κάθε x_i είναι μία λέξη
και υπάρχει μία λέξη X μήκους m
που περιλαμβάνει όλα τα x_i
σαν υπολέξεις}

ΝΑΙ. Η λέξη X αποτελεί πιστοποιητικό.
- Έλεγχος αν περιέχει κάθε x_i.
- Έλεγχος αν το μήκος είναι m

Είναι το GA NP-πλήρες;

$GA = \{\langle \{ x_1, x_2, \ldots, x_n \}, m \rangle$ | όπου
κάθε x_i είναι μία λέξη
και υπάρχει μία λέξη X μήκους m
που περιλαμβάνει όλα τα x_i
σαν υπολέξεις}

Αναγωγή *HAMPATH* σε *GA*

- Τυχαία είσοδο σε *HAMPATH*:

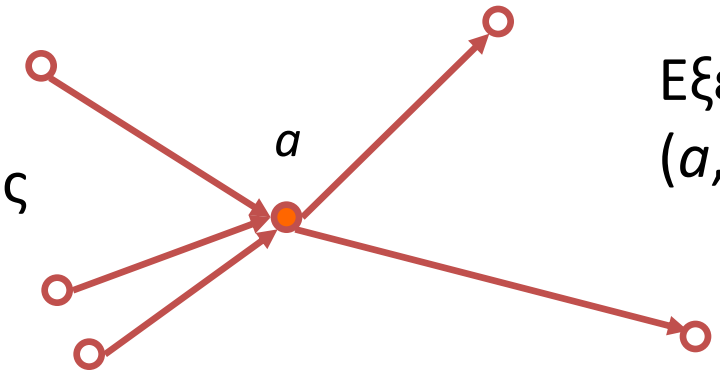
  $HAMPATH = \{\langle G, s, t\rangle \mid G$ είναι ένα γράφημα, και υπάρχει μονοπάτι μεταξύ s και t στο G που να περιλαμβάνει όλους τους κόμβους μία φορά$\}$

- Κατασκευή αντίστοιχης εισόδου στο GA έτσι ώστε η είσοδος ανήκει στο GA αν και μόνο αν η αρχική είσοδος ανήκει στο *HAMPATH*.

Άρα θα πρέπει να απεικονίσουμε τους κόμβους και τις ακμές του *G* σε είσοδο αλφαριθμητικών στο *GA*.

Ο Κόμβος

Εισερχόμενες Ακμές
(*x*, *a*)

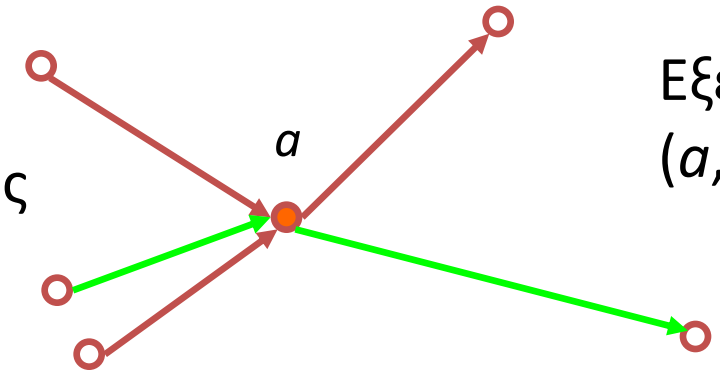


Εξερχόμενες ακμές
(*a*, *y*)

Σε ένα Hamiltonian μονοπάτι, σε κάθε κόμβο (εκτός της αρχής και του τέλους), ***ακριβώς*** μία εισερχόμενη και μία εξερχόμενη μπορεί να χρησιμοποιηθεί.

Ο Κόμβος

Εισερχόμενες Ακμές
(x, a)

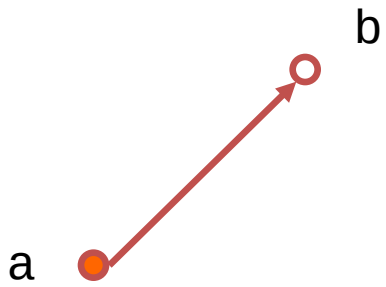


Εξερχόμενες ακμές
(a, y)

Οι GA λέξεις θα πρέπει να αναπαριστούν κάθε ακμή, αλλά με τρόπο ώστε μόνο μία να μπορεί να χρησιμοποιηθεί. Από την άλλη η υπερλέξη θα πρέπει να περιλαμβάνει όλες τις GA λέξεις.

Ιδέα: να τα φτιάξουμε έτσι ώστε οι μη επιλεγμένες ακμές να επιστρέφουν πίσω

Απλοί Κόμβοι



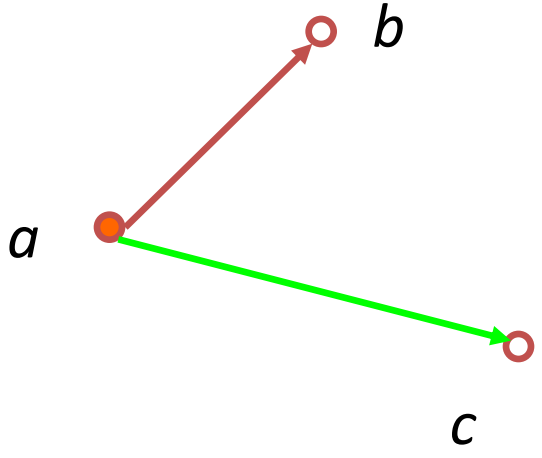
Αν υπάρχει μόνο μία εξερχόμενη ακμή (*a*, *b*) από έναν κόμβο, τότε η ακμή αυτή θα πρέπει να χρησιμοποιηθεί. Προσθέτουμε τη λέξη: **ab**

Κατασκευή Αλφαριθμητικών

- Για κάθε ακμή (a, y_i) προσθέτουμε δύο λέξεις:

  ay_ia Η «πίσω» ακμή.

  y_iay_{i+1} Αυτή συνδέει του δυνατούς προορισμούς.



aba ***aca***

bac ***cab***

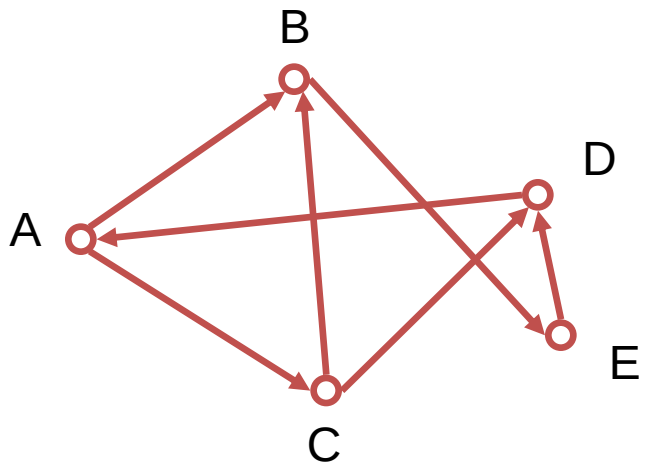
Δυνατές (μήκους 4) στοιχίσεις:

aba ***aca***

bac ***cab***

Αν υπάρχει Hamiltonian μονοπάτι, μία από αυτές τις στοιχίσεις θα χρησιμοποιηθεί.

Αρχή και Τέλος



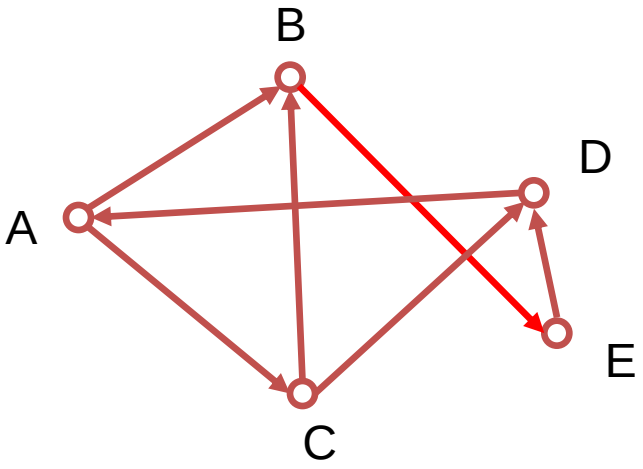
$<G, E, B>$

Ο αρχικός κόμβος θα πρέπει να είναι πρώτος:

Πρόσθεσε Λέξη: **@E** (όπου @ δεν χρησιμοποιείται αλλού)

Πρόσθεσε Λέξη: **B$** (όπου $ δεν χρησιμοποιείται αλλού)

Τι είναι το *m*?



$<G, E, B>$

@E
ED
DA
ACA
CAB
ABA
BAC
CBC
BCD
CDC
DCB
B$

{***@E***, ***ABA***, ***BAC***, ***ACA***, ***CAB***, ***BE***, ***ED***, ***DA***, ***CBC***, ***CDC***, ***BCD***, ***DCB***, ***B$***}

m = 2 + κόμβοι μίας εξ. ακμής * 1
+ πολλαπλών ακμών κόμβοι * 2 για κάθε εξ. ακμή
+ πολλαπλών εξ. ακμών κόμβοι
+ 1 για στοίχιση

HAMPATH $\leq_p$ *GA*

Ορθότητα Αναγωγής

$\Rightarrow$ Έστω ότι ο *G* έχει ένα Hamiltonian μονοπάτι από *s* σε *t*.

Η κατασκευή των λέξεων γίνεται με τέτοιο τρόπο ώστε να κωδικοποιούν ένα τέτοιο μονοπάτι σε μία υπερλέξη. Επομένως,

$$(\{ x_1, x_2, \ldots, x_n \}, m) \in GA$$

$\Leftarrow$ Έστω ότι το $(\{ x_1, x_2, \ldots, x_n \}, m) \in GA$. Τότε η υπερλέξη κωδικοποιεί ένα Hamiltonian μονοπάτι. Για λέξεις μήκους 2 ακολουθούμε την αντίστοιχη ακμή ενώ για λέξεις μήκους 3, αφού εξαντλήσουμε όλες τις περιπτώσεις ακολουθούμε την ακμή που δίνεται από το τελευταίο γράμμα. Λόγω κατασκευής αυτό θα δημιουργεί μονοπάτι Hamilton.

Απόδοση: Η αναγωγή γίνεται σε $O(n+m)$ χρόνο

Πολυπλοκότητα

Το Θεώρημα των Cook-Levin και άλλες Αναγωγές για NP-Πληρότητα

Τσίχλας Κωνσταντίνος

Το Πρόβλημα της Αληθευσιμότητας

SAT = {$\langle\varphi\rangle$ | φ είναι ένας αληθεύσιμος λογικός τύπος}

Ασχολούμαστε με συγκεκριμένη μορφή:

- Ένα λεξίγραμμα είναι μία μεταβλητή ή συμπληρωματική της: x ή $\neg x$.
- Μία φράση είναι λεξιγράμματα που συνδέονται με διάζευξη (V): $(x_1 \vee x_2 \vee x_3)$
- Ένας λογικός τύπος είναι σε Κανονική Συζευκτική Μορφή (CNF) αν αποτελείται από φράσεις συνδεόμενες με συζεύξεις ($\wedge$).
- Παράδειγμα: $(x_1 \vee x_2 \vee x_3 \vee x_4) \wedge (x_3 \vee x_5 \vee x_6) \wedge (x_3 \vee x_6)$

Αληθευσιμότητα

Ορισμός: Ένας λογικός τύπος είναι σε μορφή $_3$CNF αν είναι CNF μορφή, και όλες οι φράσεις έχουν ακριβώς 3 λεξιγράμματα.

$$(x_1 \vee x_2 \vee x_3) \wedge (x_3 \vee x_5 \vee x_6) \wedge (x_3 \vee x_6 \vee x_4)$$

$_3$SAT = $\{\langle \varphi \rangle \mid \varphi$ είναι αληθεύσιμος λογικός τύπος $_3$CNF$\}$

Αν ο φ είναι αληθεύσιμος $_3$CNF τύπος, τότε για κάθε τέτοια τιμοδοσία του φ, κάθε φράση θα περιέχει τουλάχιστον ένα λεξίγραμμα που είναι 1.

SAT και $_3$SAT

Θα δείξουμε μία πολυωνυμική αναγωγή που αντιστοιχεί λογικούς τύπους μορφής CNF σε λογικούς τύπους μορφής $_3$CNF.

Τι γίνεται αν έχουμε μία φράση με 1–2 λεξιγράμματα;

Μία φράση με x λεξιγράμματα αντιστοιχείται σε x-2 φράσεις με τα αρχικά λεξιγράμματα καθώς και με x-3 καινούργια.

Παράδειγμα: $(x_1 \vee x_2 \vee x_3 \vee x_4 \vee x_8)$

$\Rightarrow (x_1 \vee x_2 \vee y_1) \wedge (\neg y_1 \vee x_3 \vee y_2) \wedge (\neg y_2 \vee x_4 \vee x_8)$

$SAT \leq_P {}_3SAT$

Η φ έχει αληθοποιός τιμοδοσία αν και μόνο αν η φ_3 έχει.

Απόδειξη:

$\Leftarrow$ Μία τιμοδοσία που ικανοποιεί την φ_3 δεν μπορεί να στηρίζεται μόνο σε νέα λεξιγράμματα – τουλάχιστον ένα αρχικό λεξίγραμμα σε κάθε φράση πρέπει να ικανοποιείται.

$\Rightarrow$ Μία τιμοδοσία που ικανοποιεί την φ κάνει τουλάχιστον ένα λεξίγραμμα αληθές για κάθε φράση. Στην αντίστοιχη φράση φ_3 με αυτό το λεξίγραμμα οι νέες μεταβλητές μπορούν να πάρουν οποιαδήποτε τιμή. Αυτό μας επιτρέπει μία διάδοση κατάλληλων τιμών στις νέες μεταβλητές έτσι ώστε όλες οι αντίστοιχες φράσεις να ικανοποιούνται.

Η αναγωγή είναι πολυωνυμική και άρα $SAT \leq_P {}_3SAT$.

Θεώρημα Cook-Levin

Το SAT είναι NP-πλήρες.

Απόδειξη:

- Εύκολο να δείξετε ότι $SAT \in NP$
- Πρέπει να δείξουμε ότι κάθε NP πρόβλημα ανάγεται στο SAT σε πολυωνυμικό χρόνο.

Ιδέα: Έστω $L \in NP$, και M η NTM που την επιλύει.

Σε είσοδο w μήκους n, η M τρέχει σε χρόνο $t(n) = n^c$.

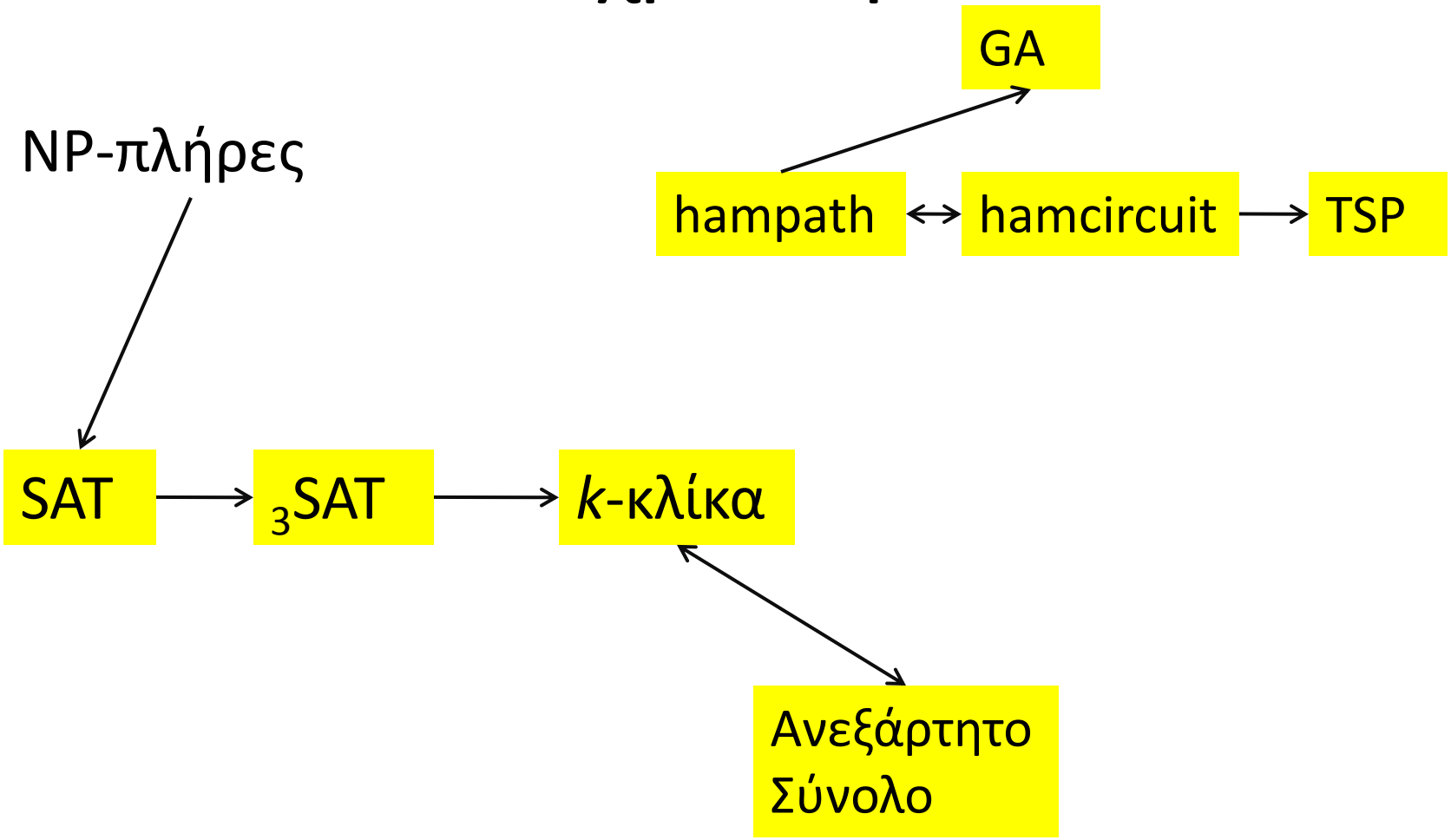
Ορίζουμε το μητρώο της μηχανής ως έναν πίνακα $n^c \times n^c$ που περιγράφει τον υπολογισμό της M σε είσοδο w.

Πολυπλοκότητα

Περισσότερες Αναγωγές ☺

Τσίχλας Κωνσταντίνος

Μέχρι Τώρα…


GA
NP-πλήρες
hampath
hamcircuit
TSP
SAT
3SAT
k-κλίκα
Ανεξάρτητο Σύνολο

$_2$SAT $\in$?

$_2$SAT = {$\langle\varphi\rangle$ | ο φ ικανοποιήσιμος $_2$CNF τύπος}

$$(x \vee y) \wedge (\bar{x} \vee z) \wedge (\bar{z} \vee \bar{w}) \wedge (w \vee x)$$

- Είναι το $_2$SAT *NP*-πλήρες; Είναι στο *P*;
- Ή μήπως δεν ξέρουμε;;
- Τελικά, το $_2$SAT ανήκει στο *P*.

$_2$SAT $\in$ P

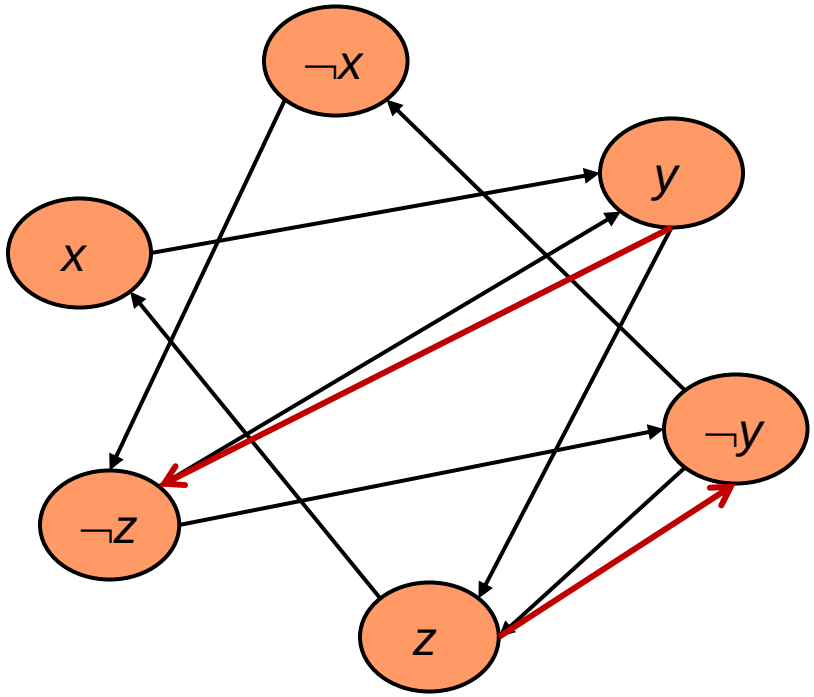
- Κατασκευάζουμε γράφημα G_φ από το στιγμιότυπο φ του $_2$SAT:
  - Οι κορυφές είναι οι μεταβλητές του φ
  - Υπάρχει ακμή από a σε b αν και μόνο αν η φράση $(\neg a \vee b)$ ανήκει στο φ. $(a \Rightarrow b)$
  - Αν η (a,b) είναι ακμή τότε και η $(\neg b,\neg a)$ είναι ακμή.

Πρόταση: *η φ είναι μη αληθεύσιμη αν και μόνο αν υπάρχει μεταβλητή x έτσι ώστε να υπάρχει μονοπάτι από το x στο $\neg x$ και από το $\neg x$ στο x στο G_φ.*

Παράδειγμα

$$(\neg x \vee y) \wedge (\neg y \vee z) \wedge (x \vee \neg z) \wedge (z \vee y)$$

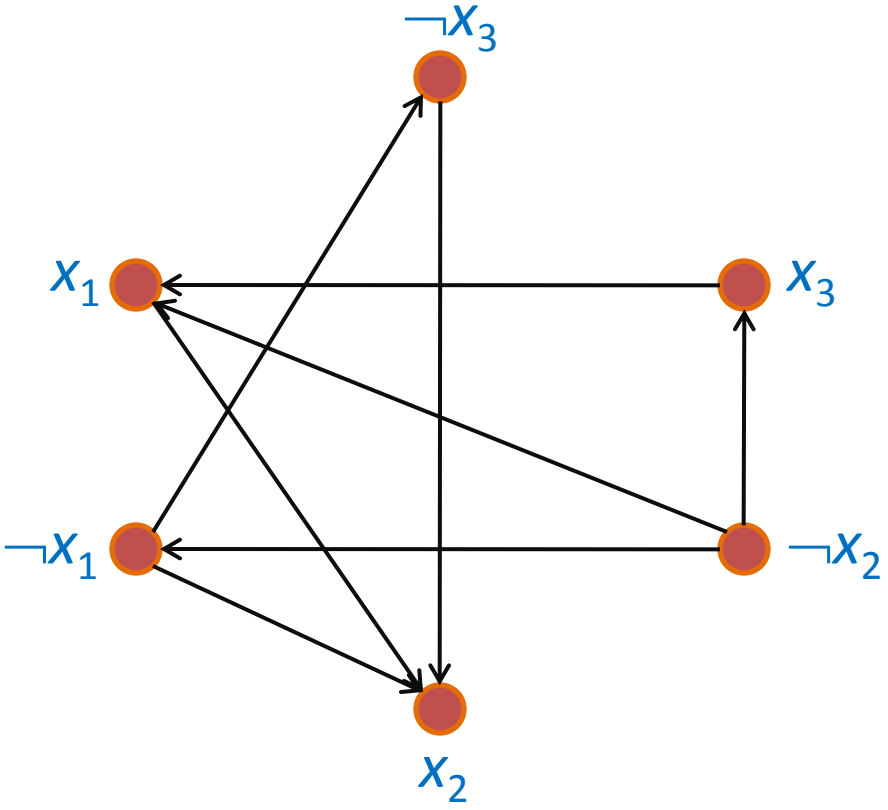
$\wedge(\neg z \vee \neg y)$



Άλλο Παράδειγμα

$$(x_1 \vee x_2)(x_1 \vee \neg x_3)(\neg x_1 \vee x_2)(x_2 \vee x_3) \Leftrightarrow$$

$$(\neg x_1 \Rightarrow x_2)(\neg x_2 \Rightarrow x_1)(\neg x_1 \Rightarrow \neg x_3)(x_3 \Rightarrow x_1)(x_1 \Rightarrow x_2)(\neg x_2 \Rightarrow \neg x_1)(\neg x_2 \Rightarrow x_3)(\neg x_3 \Rightarrow x_2)$$



Παρατήρηση

Ισχυρισμός: Αν το γράφημα περιέχει μονοπάτι από το α στο β, τότε περιέχει και μονοπάτι από το $\neg\beta$ στο $\neg\alpha$.

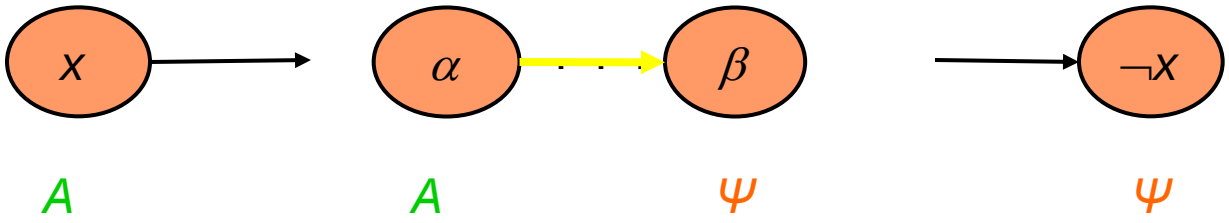
Απόδειξη: Αν υπάρχει ακμή (α,β), τότε υπάρχει και η ακμή $(\neg\beta,\neg\alpha)$.

Ορθότητα

<u>Πρόταση:</u> η φ είναι μη αληθεύσιμη αν και μόνο αν υπάρχει μεταβλητή x έτσι ώστε να υπάρχει μονοπάτι από το x στο $\neg x$ και από το $\neg x$ στο x στο G_φ (ισχυρά συνδεδεμένο γράφημα).

- Έστω ότι υπάρχουν μονοπάτια $x \ldots \neg x$ and $\neg x \ldots x$ για κάποια μεταβλητή x, αλλά υπάρχει και μία αληθοποιός τιμοδοσία ρ.
- Αν $\rho(x)=A$ (παρόμοια για $\rho(x)=\Psi$):

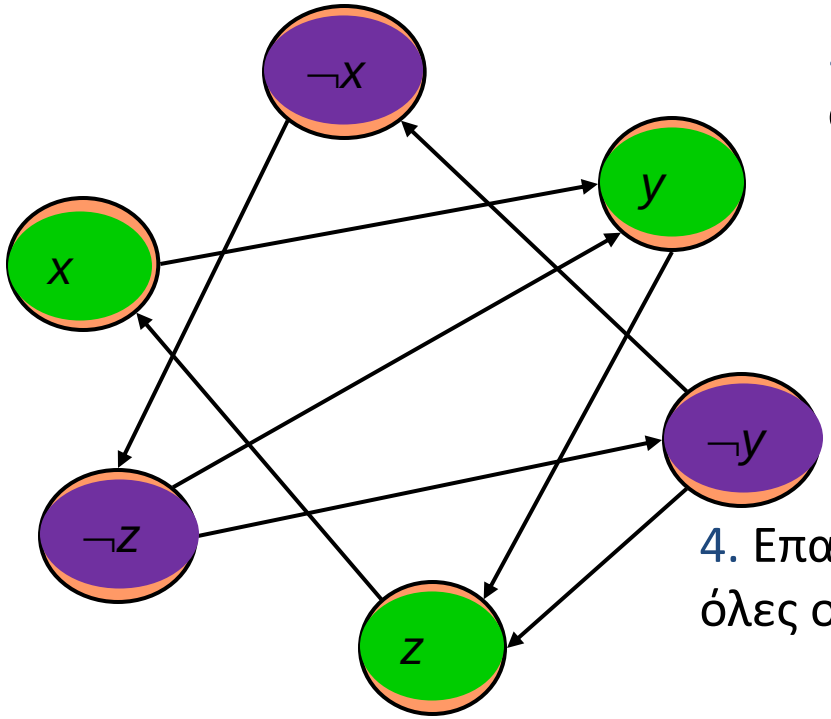
$(\neg\alpha \vee \beta)$ είναι ψευδές!



Ορθότητα (2)

- Έστω ότι δεν υπάρχουν τέτοια μονοπάτια.
- Κατασκευάζουμε μία τιμοδοσία ως εξής:

1. Επέλεξε μία μεταβλητή α χωρίς τιμή και χωρίς μονοπάτι από α σε $\neg\alpha$, και δώσε τιμή A



2. Δώσε τιμή A σε όλες τις γειτονικές κορυφές

3. Δώσε τιμή Ψ σε όλα τα συμπληρώματά τους

4. Επανέλαβε διαδικασία μέχρι όλες οι κορυφές να έχουν τιμή

Ορθότητα (2)

Ισχυρισμός: Ο αλγόριθμος είναι σωστός

Απόδειξη: Αν υπήρχε μονοπάτι από το x (A) σε y και $\neg y$, τότε θα υπήρχε και μονοπάτι από το x στο $\neg y$ και από το $\neg y$ στο $\neg x$. Αντίστοιχα και από $\neg x$ (Ψ).

2SAT $\in$ P

Έχουμε περιγράψει τον εξής αποδοτικό αλγόριθμο για το 2SAT:

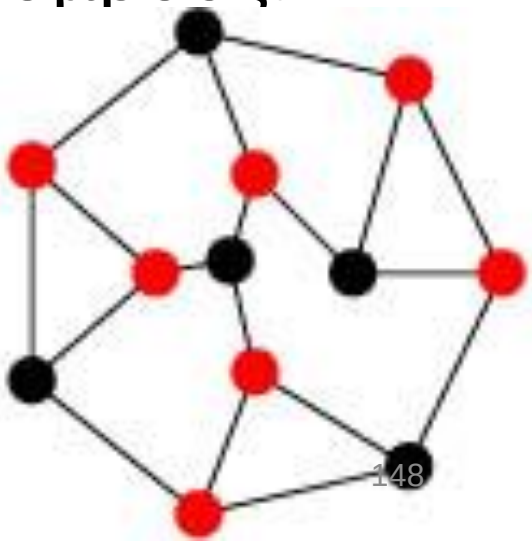
- Για κάθε μεταβλητή x δες αν υπάρχει μονοπάτι από x σε $\neg x$ και αντίστροφα.
- Απορρίπτουμε αν υπάρχει.
- Αποδεχόμαστε διαφορετικά

$\Rightarrow$ 2SAT$\in$P

Κομβικό Κάλυμμα

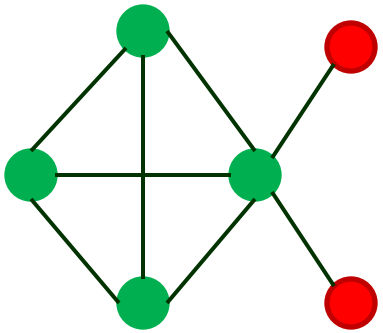
ΚΚ = {⟨*G*,*k*⟩ / το *G* είναι έναν ακατεύθυντο γράφημα και περιέχει κομβικό κάλυμμα *k* κόμβων}

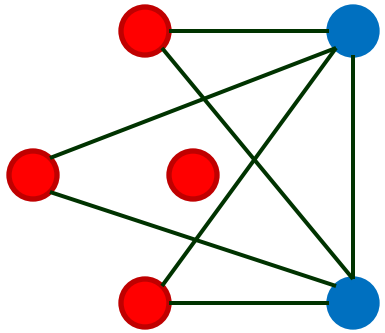
Σύνολο *k* κόμβων έτσι ώστε κάθε ακμή να καταλήγει σε έναν από αυτούς τους κόμβους.



Κλίκα $\leq_P$ Κομβικό Κάλυμμα

- Πρώτα αποδείξτε ότι το Κομβικό Κάλυμμα ανήκει στο **NP**
- Έπειτα, ανάγουμε την k-κλίκα στο κομβικό κάλυμμα
  - Υπολόγισε το συμπλήρωμα G_C του γραφήματος G σε πολυωνυμικό χρόνο
  - Το G έχει κλίκα μεγέθους k αν και μόνο αν το G_C έχει κομβικό κάλυμμα μεγέθους $|V| - k$





Κλίκα $\leq_P$ Κομβικό Κάλυμμα

- Αν ο G έχει κλίκα μεγέθους k, το G_C έχει κομβικό κάλυμμα μεγέθους $|V| - k$
  - Έστω V' η k-κλίκα
  - Τότε το $V - V'$ είναι ένα κομβικό κάλυμμα στο G_C
    - Έστω (u,v) μία οποιαδήποτε ακμή στο G_C
    - Τότε οι u και v δεν μπορεί να ανήκουν ταυτόχρονα στο V' (Γιατί;)
    - Άρα τουλάχιστον ένα από τα u ή v είναι στο $V-V'$, και επομένως η ακμή (u, v) καλύπτεται από το $V-V'$
    - Αφού αυτό είναι αληθές για κάθε ακμή στο G_C, το $V-V'$ είναι ένα κομβικό κάλυμμα

Κλίκα $\leq_p$ Κομβικό Κάλυμμα

- Αν το G_C έχει κομβικό κάλυμμα $V' \subseteq V$, με $|V'|=|V| - k$, τότε το G έχει κλίκα μεγέθους k
  - Για κάθε $u,v \in V$, αν $(u,v) \in G_C$ τότε $u \in V'$ ή $v \in V'$ ή και τα δύο (Γιατί;)
  - Αντιθετοαντίστροφο: αν $u \notin V'$ και $v \notin V'$, τότε $(u,v) \in E$
  - Με άλλα λόγια, όλες οι κορυφές στο $V-V'$ είναι συνδεδεμένες με μία ακμή και άρα το $V-V'$ είναι κλίκα
  - Αφού $|V| - |V'| = k$, το μέγεθος της κλίκας είναι k

Αναγωγές από SAT

- Θα πρέπει να αναζητήσουμε δομές στην άλλη γλώσσα που να αντιστοιχούν στις μεταβλητές και φράσεις.
  - Οι δομές αυτές αποκαλούνται *εξαρτήματα*

$_3$SAT$\leq_P$ΚΛΙΚΑ

- Κόμβος $\rightarrow$ Λεξίγραμμα
- Τριάδες $\rightarrow$ Φράσεις
- Τιμοδοσία Λεξιγράμματος $\rightarrow$ Συμμετοχή στην κλίκα
- Φράση με ΑΛΗΘΕΣ Λεξιγραμμα $\rightarrow$ Κόμβος από τριάδα που συμμετέχει στην κλίκα (για να πάρουμε το k)

Κάλυμμα Συνόλου

Κάλυμμα Συνόλου: Δοθέντος ενός συνόλου στοιχείων U, μίας συλλογής $S_1, S_2, \ldots, S_m$ υποσυνόλων του U, και έναν ακέραιο k, υπάρχει συλλογή από $\leq k$ από αυτά τα σύνολα των οποίων η ένωση να δίνει U;

- Εφαρμογή:
  - m κομμάτια λογισμικού
  - Σύνολο U από n δυνατότητες που θα θέλαμε να έχει το σύστημά μας.
  - Το i-οστό κομμάτι κώδικα παρέχει ένα υποσύνολο $S_i \subseteq U$ δυνατοτήτων.
  - Στόχος: να επιτύχουμε τις n δυνατότητες με το μικρότερο πλήθος κομματιών λογισμικού.

- π.χ:

$U = \{ 1, 2, 3, 4, 5, 6, 7 \}$

$k = 2$

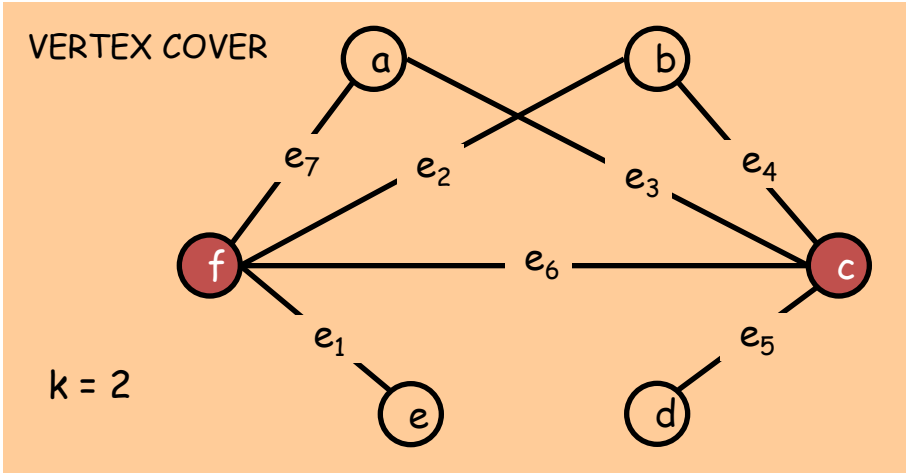
$S_1 = \{3, 7\}$ $S_4 = \{2, 4\}$

$S_2 = \{3, 4, 5, 6\}$ $S_5 = \{5\}$

$S_3 = \{1\}$ $S_6 = \{1, 2, 6, 7\}$

Κομβικό Κάλυμμα $\leq_p$ Κάλυμμα Συνόλου

- Απόδειξη: Δοθέντος ενός στιγμιοτύπου κομβικού καλύμματος, $\langle G=(V,E), k\rangle$, κατασκευάζουμε ένα κάλυμμα συνόλου ίδιου μεγέθους
- Κατασκευή:
  - Παράγουμε ένα στιγμιότυπο Καλύμματος Συνόλου:
    - $k = k$, $U = E$, $S_v = \{e \in E : e \text{ είναι προσκείμενη στην } v\}$
  - Κάλυμμα συνόλου μεγέθους $= k$ αν και μόνο αν κομβικό κάλυμμα $= k$.



SET COVER

U = { 1, 2, 3, 4, 5, 6, 7 }
k = 2

S_a = {3, 7} S_b = {2, 4}
S_c = {3, 4, 5, 6} S_d = {5}
S_e = {1} S_f = {1, 2, 6, 7}

MAX2SAT

MAX2SAT(k) – δοθέντος ενός λογικού τύπου με 2 λεξιγράμματα ανά φράση, υπάρχει τιμοδοσία έτσι ώστε τουλάχιστον k φράσεις να ικανοποιούνται;

Θεώρημα: Το *MAX2SAT* είναι NP-πλήρες

- Έστω οι παρακάτω 10 φράσεις:

$$(x)(y)(z)(w)(\neg x \vee \neg y)(\neg y \vee \neg z)(\neg z \vee \neg x)(x \vee \neg w)(y \vee \neg w)(z \vee \neg w)$$

- Αν το $(x \vee y \vee z)$ είναι ΑΛΗΘΕΣ, τότε 7 φράσεις ικανοποιούνται ενώ αν το $(x \vee y \vee z)$ είναι ΨΕΥΔΕΣ, τότε το πολύ 6 φράσεις ικανοποιούνται.

${}_3SAT \leq_p MAX2SAT$

- Έστω φ ο λογικός τύπος με k φράσεις και 3 λεξιγράμματα ανά φράση. Για κάθε φράση $(x \vee y \vee z)$ του φ, $R(\varphi)$ θα περιέχει 10 νέες φράσεις όπως το παράδειγμα πριν, όπου w είναι μία νέα μεταβλητή (διαφορετική για κάθε φράση)
- Η φ είναι ικανοποιήσιμη αν και μόνο αν τουλάχιστον $7k$ φράσεις της $R(\varphi)$ ικανοποιούνται.
- Το MAX2SAT ανήκει στο NP
- **Παρατήρηση:** Κατασκευάσαμε ένα εξάρτημα (οι 10 νέες φράσεις) που είχαν κάποια ενδιαφέρουσα ιδιότητα και επέτρεπε την αναγωγή.

Άθροισμα Υποακολουθίας

Μας δίνεται μία ακολουθία αριθμών $x_1,\ldots, x_k$, και ένας αριθμός στόχος t και μας ζητείται να βρούμε αν υπάρχει υποακολουθία που να έχει άθροισμα t.

Το πρόβλημα είναι NP-πλήρες:

- Δείξτε ότι ανήκει στο NP
- ${}_3\text{SAT} \leq_P$ Άθροισμα Υποακολουθίας

Εξαρτήματα

- Μεταβλητή $\rightarrow$ Ζεύγος αριθμών στην ακολουθία
- Φράση $\rightarrow$ Σε συγκεκριμένη θέση στις δεκαδικές αναπαραστάσεις των αριθμών

- Η μεταβλητή $x_i \rightarrow$ σε δύο αριθμούς y_i και z_i. Κάθε υποακολουθία θα περιλαμβάνει ή το y_i ή το z_i ανάλογα με την τιμοδοσία της x_i.
- Φράση $\rightarrow$ η θέση που αντιστοιχεί στην αναπαράσταση του στόχου t, περιέχει μία τιμή που επιβάλλει ένα από τα λεξιγράμματα της φράσης να παίρνει την τιμή ΑΛΗΘΕΣ.

Κατασκευή Πίνακα

	1	2	…	l	c_1	c_2	…	c_k
y_1	1	0	0	0	1	0	0	0
z_1	1	0	0	0	0	0	0	0
y_2		1	0	0	0	0	0	0
z_2		1	0	0	1	0	0	0
…								
g_1					1	0	0	0
h_1					1	0	0	0
g_2						1	0	0
h_2						1	0	0
…								
t	1	1	…	1	3	3	…	3

Πολυωνυμικός Χρόνος Κατασκευής της Αναγωγής

Απόδειξη

$\Rightarrow$ Έστω φ αληθεύσιμη: επιλέγουμε με βάση την τιμοδοσία τις κατάλληλες γραμμές και βρίσκουμε τον αριθμό στόχο

$\Leftarrow$ Έστω ότι κάποια υποακολουθία έχει άθροισμα τον αριθμό στόχο:

- Κάθε ψηφίο είναι 0 ή 1
- Κάθε στήλη περιέχει το πολύ πέντε 1

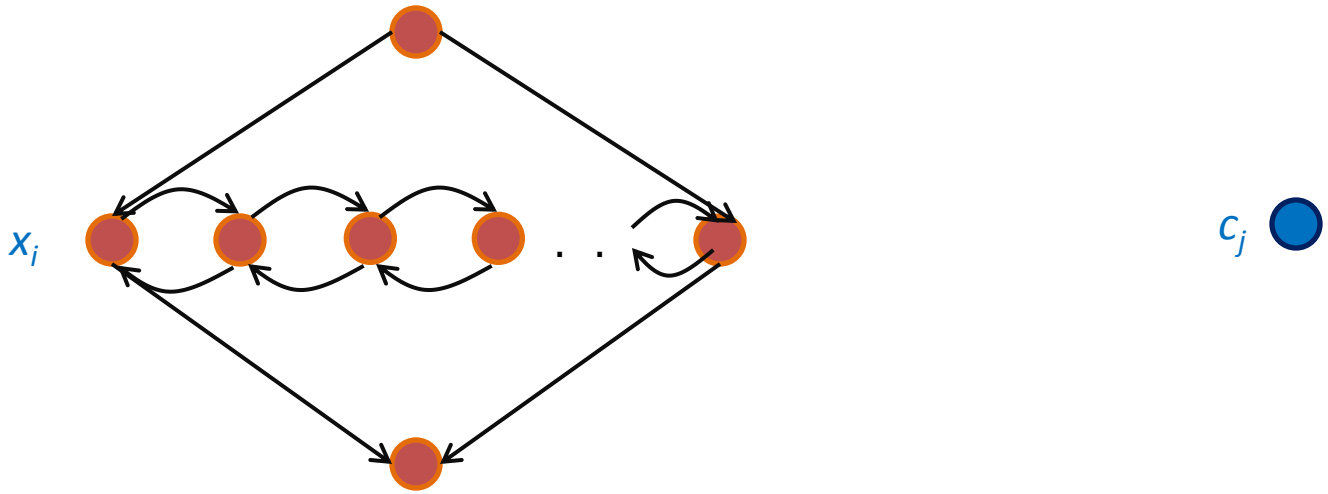
  Άρα δεν παράγεται κρατούμενο
- Πρέπει να υπάρχουν ή y_i ή z_i που συμμετέχουν
- Υπάρχει τουλάχιστον ένα λεξίγραμμα που συμμετέχει σε φράση

3SAT $\leq_P$ HAMPATH

HAMPATH = {$\langle G, s, t\rangle$ | το *κατευθυντό G* έχει Hamiltonian μονοπάτι από τον *s* στο *t*}

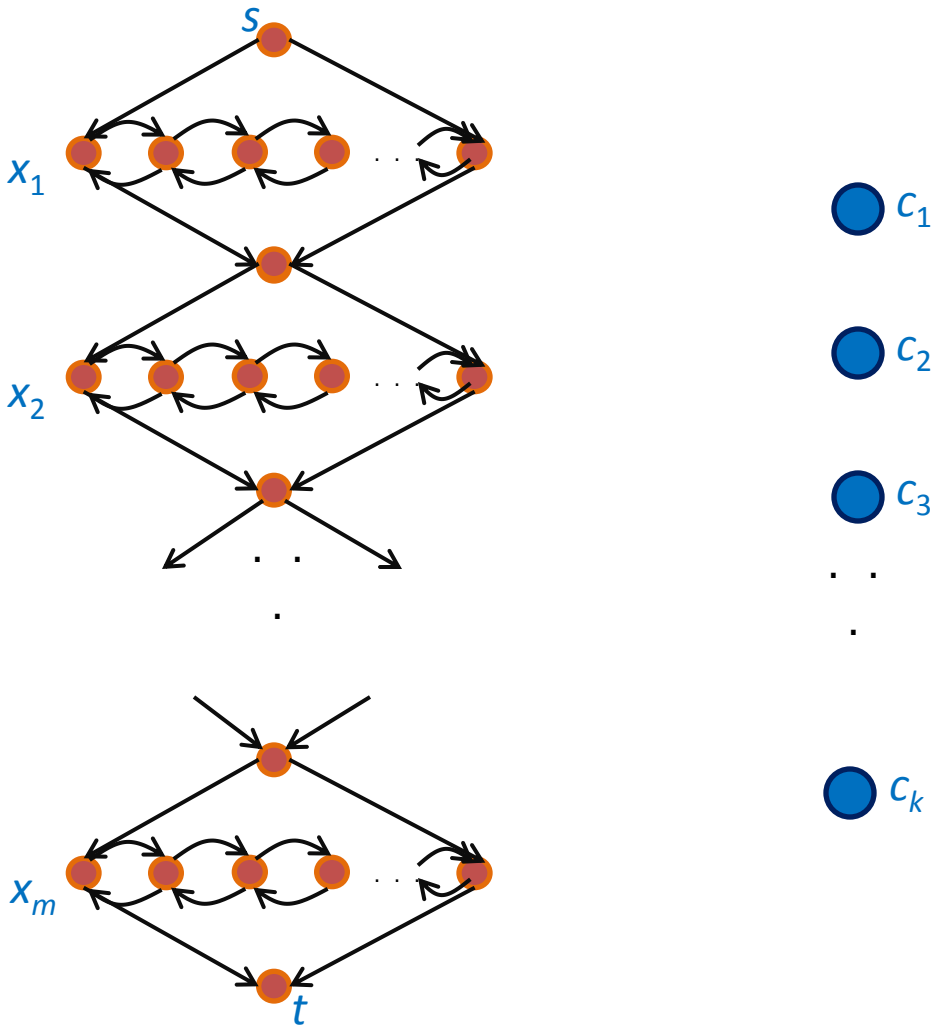
Εξαρτήματα:

- Μεταβλητή $\rightarrow$ Ρομβοειδής δομή που μπορεί να διανυθεί κατά δύο τρόπους ανάλογα με την τιμοδοσία της μεταβλητής
- Φράση $\rightarrow$ Ένας κόμβος από τον οποίο η διέλευση αντιστοιχεί στην ικανοποιησιμότητα της συγκεκριμένης φράσης

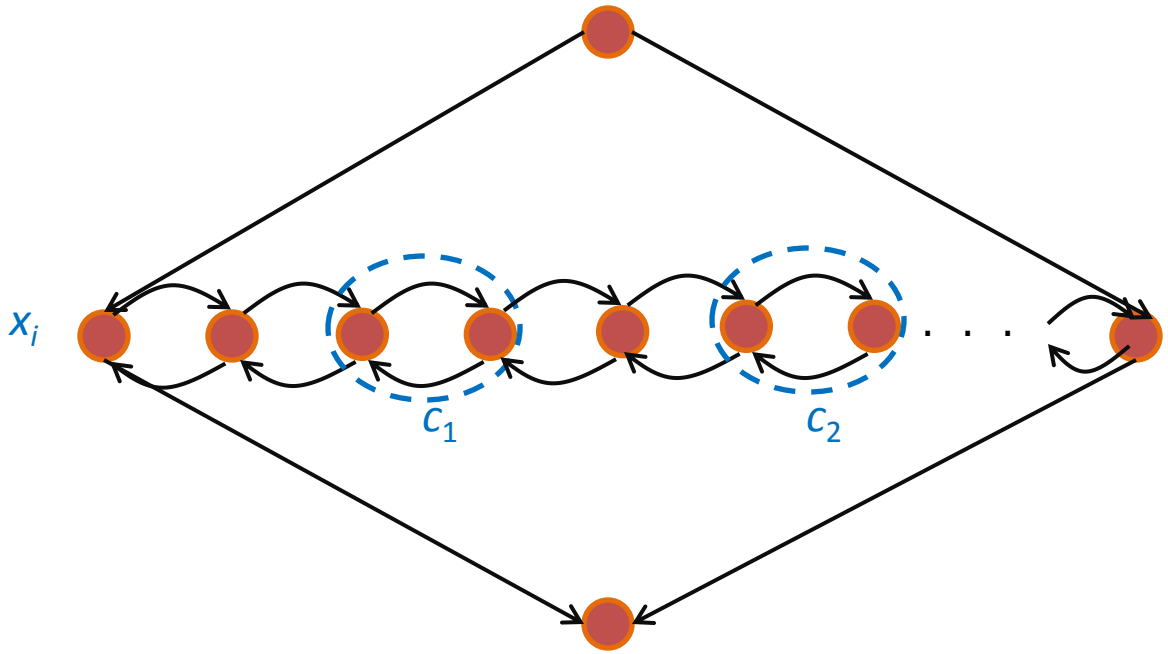


Γενική Δομή Γραφήματος

(Λείπουν κάποιες ακμές)

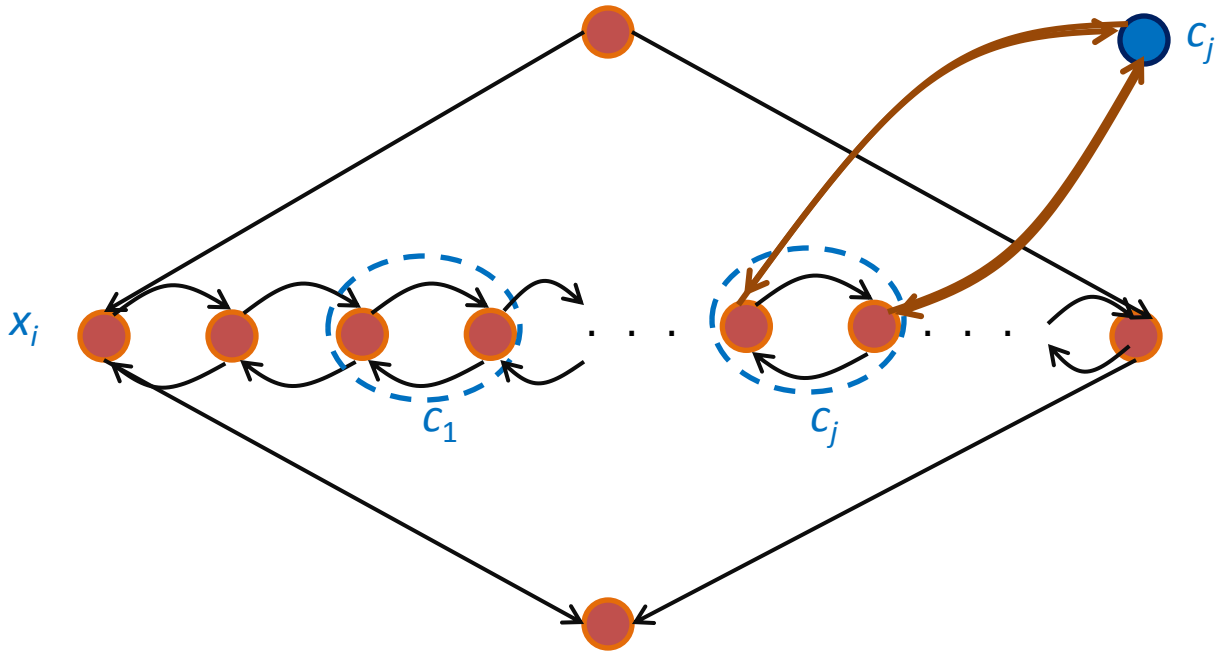


Το Λεξίγραμμα



Συνολικά έχουμε $3k+1$ κόμβους μεταξύ των δύο ακραίων κόμβων.

Το Λεξίγραμμα



Τρεις περιπτώσεις:

- Το x_i δεν ανήκει στην φράση c_j
- Το x_i ανήκει στην φράση c_j
- Το $\neg x_i$ ανήκει στην φράση c_j

Η κατασκευή του γραφήματος G ολοκληρώθηκε.

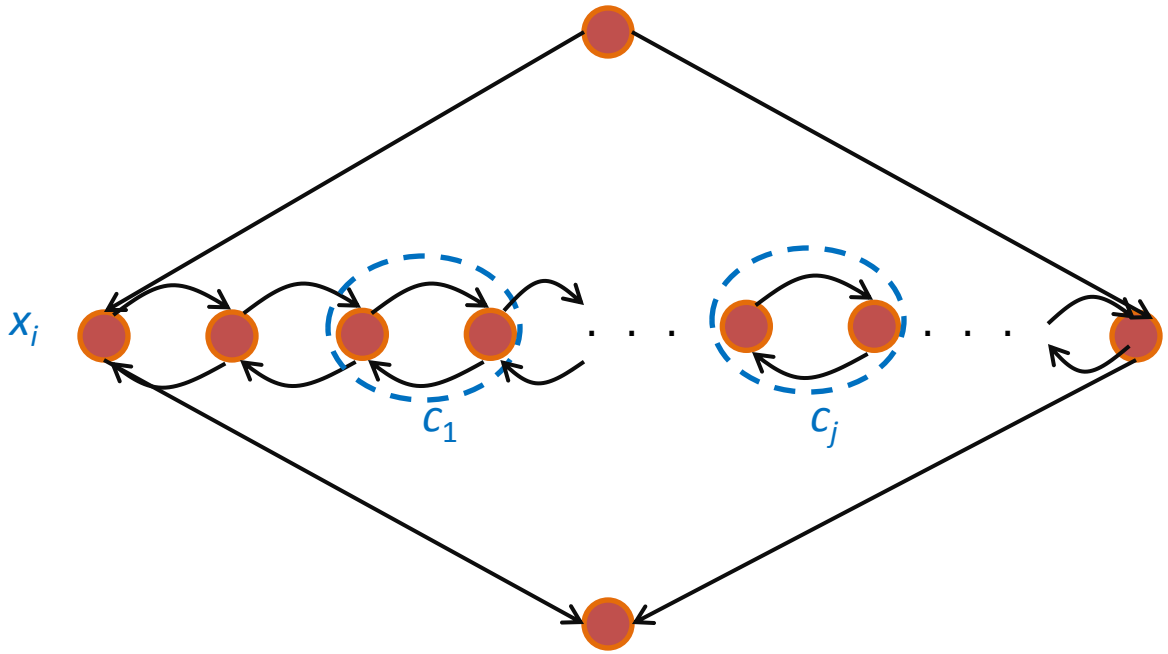
Αναγωγή

Λήμμα: Η φ είναι ικανοποιήσιμη αν και μόνο αν ο G έχει Hamiltonian μονοπάτι από το s στο t.

$\Rightarrow$ Στρατηγική:

- Στην αρχή δεν θα πάρουμε καθόλου υπόψη τους κόμβους φράσεις
- Θα διαπεράσουμε τα ρομβοειδή υπογραφήματα

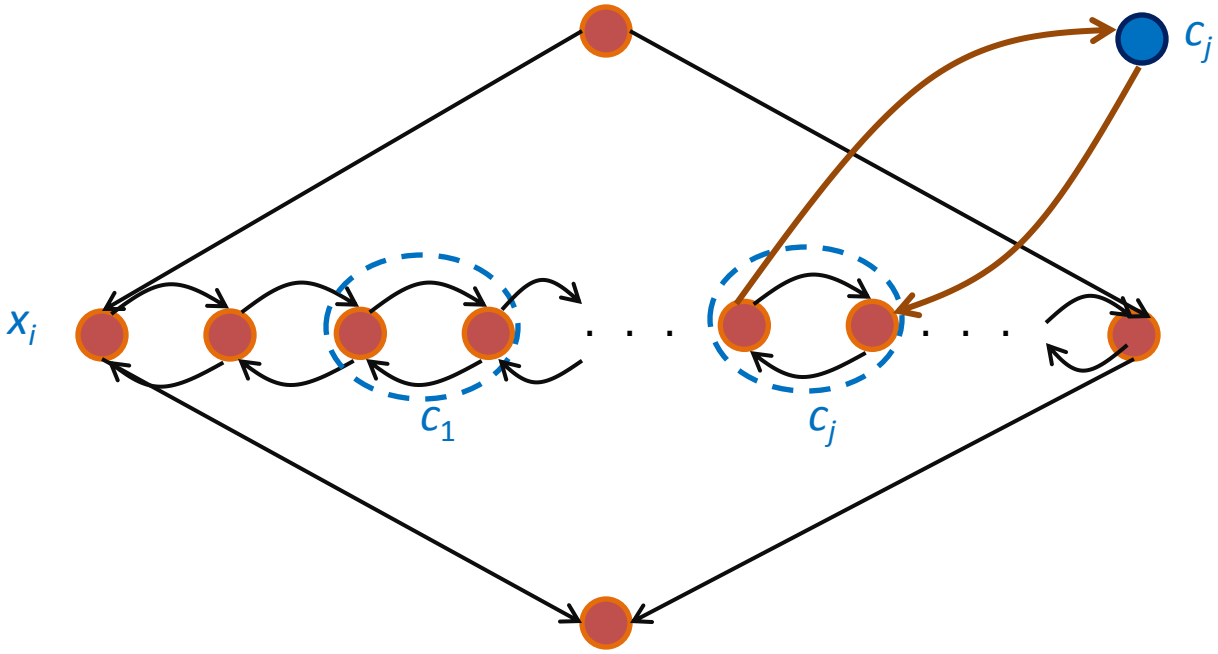
Διαπέραση Ρόμβων



Δύο περιπτώσεις:

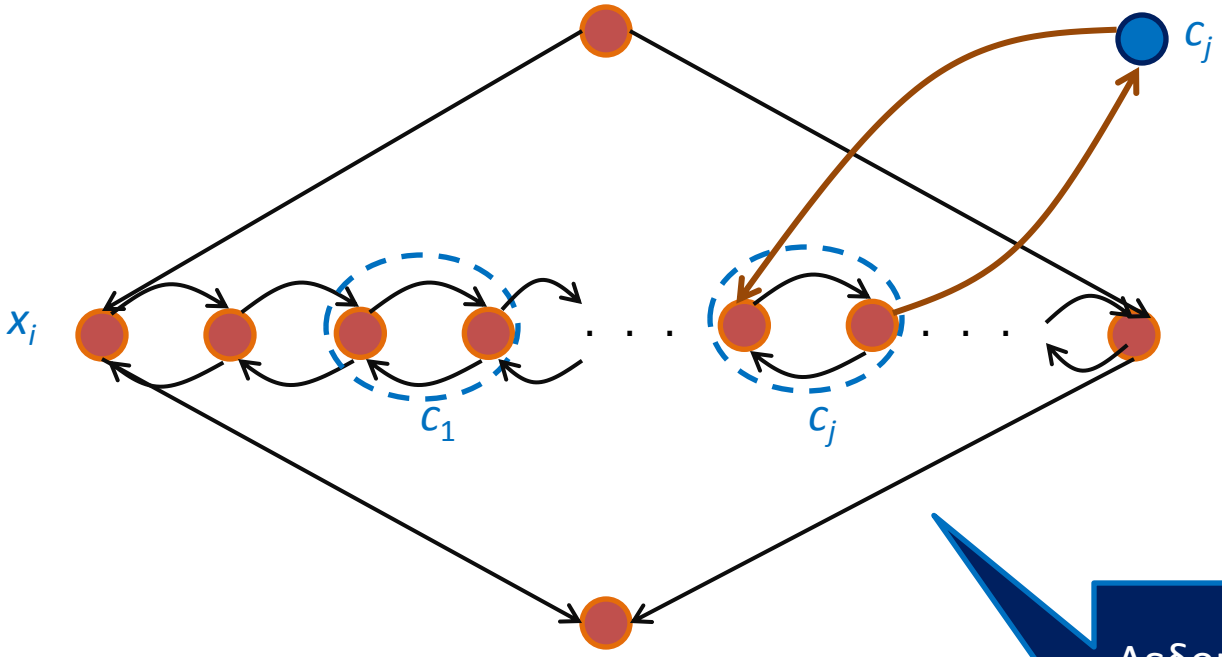
- x_i είναι ΑΛΗΘΕΣ
- x_i είναι ΨΕΥΔΕΣ

Οι Κόμβοι Φράσεων



Το x_i είναι ΑΛΗΘΕΣ και ανήκει στο c_j

Οι Κόμβοι Φράσεων (2)



Το $\neg x_i$ είναι ΑΛΗΘΕΣ και ανήκει στο c_j

Δεδομένου ότι για κάθε κόμβο φράσης περνάμε από μία μεταβλητή μόνο, η απόδειξη ολοκληρώθηκε.

Αναγωγή - $\Leftarrow$

Ένα *κανονικό* Hamiltonian μονοπάτι διαπερνά τις ρομβοειδής δομές με δύο τρόπους:

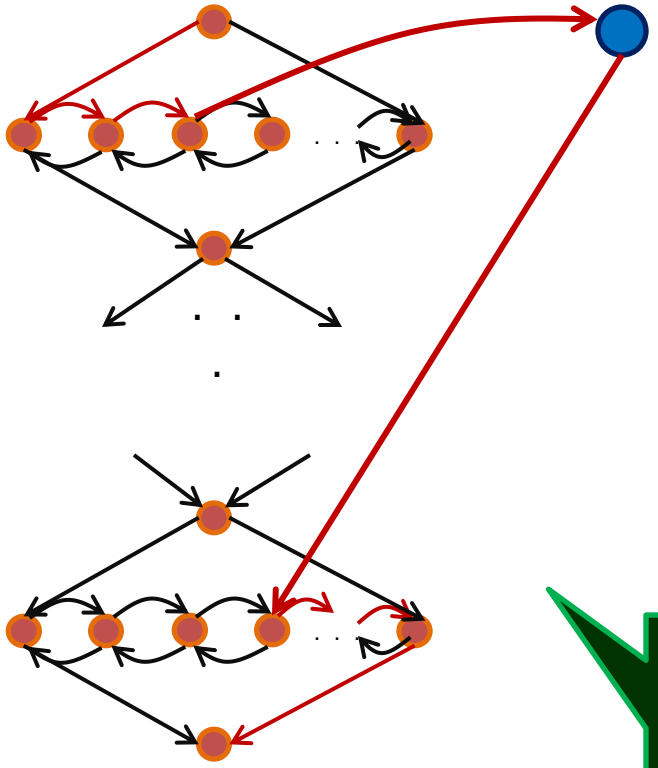
- Αν το x_i διαπερνά από αριστερά προς δεξιά, τότε δίνουμε τιμή ΑΛΗΘΕΣ
- Αν το x_i διαπερνά από δεξιά προς αριστερά, τότε δίνουμε τιμή ΨΕΥΔΕΣ

Επίσης:

- Κάθε κόμβος φράση εμφανίζεται μόνο μία φορά
- Η πηγή της παράκαμψης προς κόμβο-φράση καθορίζει ποιο λεξίγραμμα είναι ΑΛΗΘΕΣ

Κανονικά Hamiltonian Μονοπάτια

Λήμμα: Όλα τα Hamiltonian μονοπάτια είναι *κανονικά*.



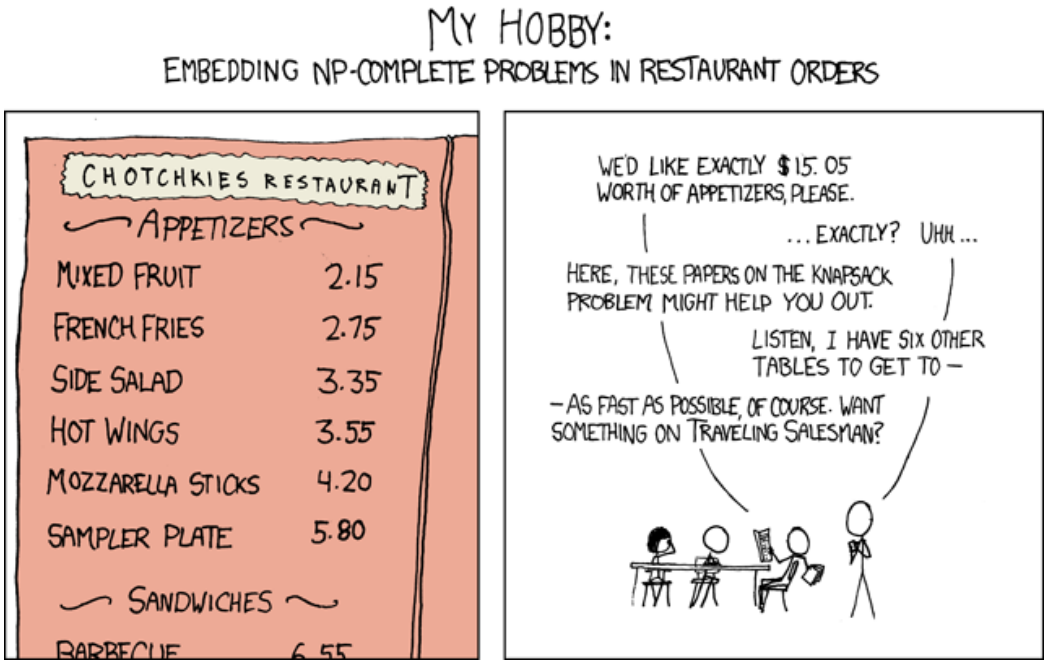
Όλοι οι ενδιάμεσοι κόμβοι δεν είναι δυνατόν να προσπελασθούν στην συνέχεια και άρα δεν έχουμε Hamiltonian μονοπάτι.

Η αναγωγή γίνεται σε πολυωνυμικό χρόνο. Άρα, το Hamiltonian μονοπάτι είναι NP-πλήρες.

Το Πρόβλημα του Σάκου (Knapsack)

Δοθέντος ενός συνόλου αντικειμένων, κάθε ένα με βάρος και τιμή και ενός σάκου που χωρά βάρος k, υπάρχουν αντικείμενα που να χωρούν στο σάκο και των οποίων η συνολική τιμή να είναι μεγαλύτερη του W;

Θεώρημα: Το πρόβλημα του Σάκου είναι NP-πλήρες.



ΑΘ_ΥΠ$\leq_p$KNAPSACK

- Εύκολη αναγωγή με ***περιορισμό***

KNAPSACK:

Ένα σύνολο αντικειμένων $I = \{i_1, \ldots, i_n\}$ όπου κάθε αντικείμενο i_j έχει τιμή v_j και βάρος w_j (θετικοί ακέραιοι) καθώς και μία τιμή στόχο p και βάρος στόχο k.

Ερώτηση: Υπάρχει υποσύνολο T του I με συνολική τιμή τουλάχιστον p και βάρος το πολύ k;

ΑΘ_ΥΠ$\leq_P$KNAPSACK

Θέτουμε σε στιγμιότυπα του KNAPSACK έτσι ώστε

$$v_j = w_j (= \alpha_j)$$

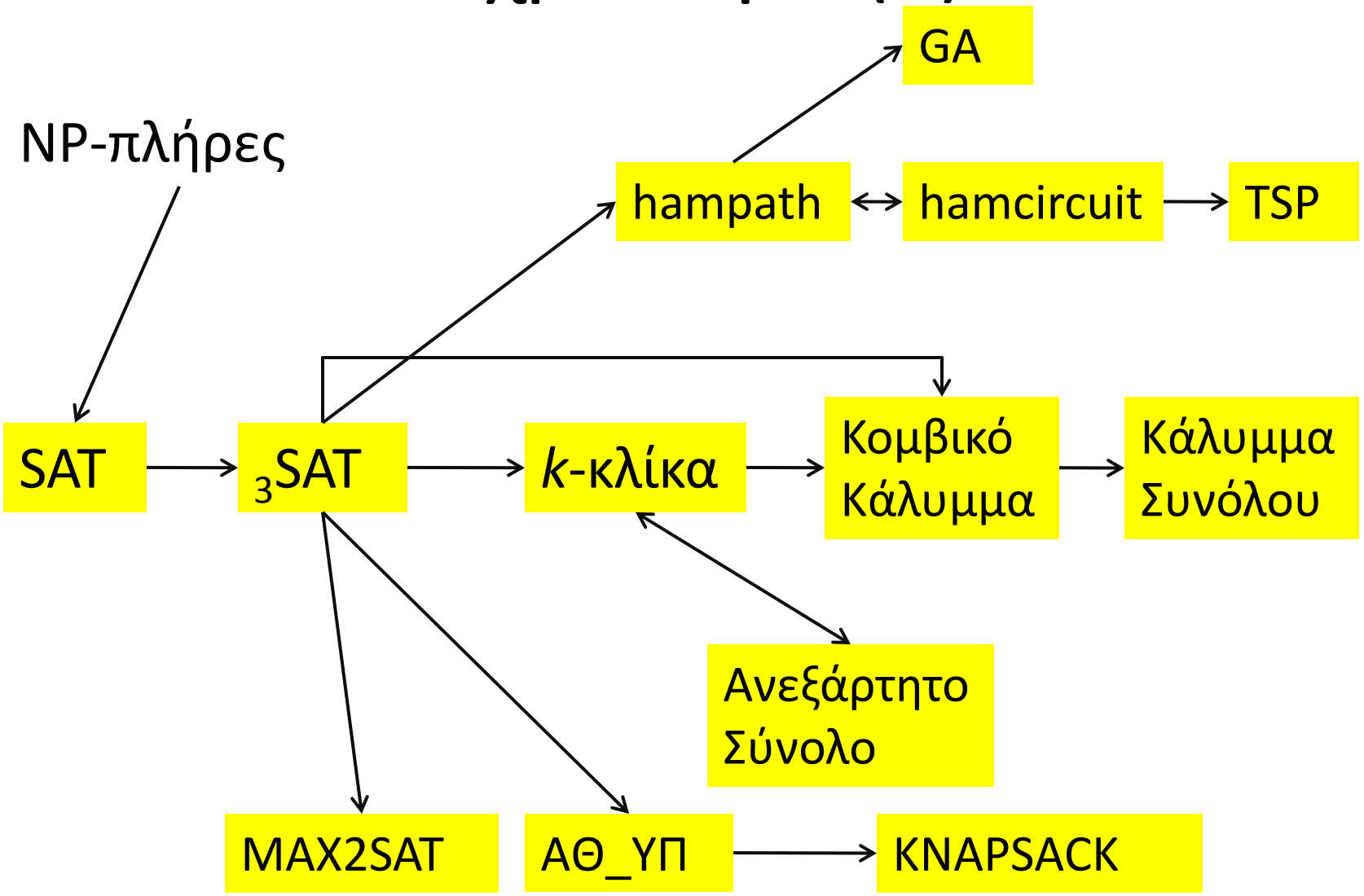
για όλα τα j, και $p = k(=t)$.

Το πρόβλημα που έχουμε είναι το ΑΘ_ΥΠ.

Πράγματι (I είναι ένα υποσύνολο των ακεραίων για το ΑΘ_ΥΠ):

$$\sum_{j \in I} v_j \geq p \qquad \sum_{j \in I} w_j \leq k \qquad \sum_{j \in I} n_j = t$$

Μέχρι Τώρα (2) …


GA
NP-πλήρες
hampath
hamcircuit
TSP
SAT
3SAT
k-κλίκα
Κομβικό Κάλυμμα
Κάλυμμα Συνόλου
Ανεξάρτητο Σύνολο
MAX2SAT
ΑΘ_ΥΠ
KNAPSACK

3-διάστατο Πρόβλημα Ταιριάσματος

Δοθέντων ξένων συνόλων X, Y και Z, κάθε ένα με μέγεθος n, και δοθέντος ενός συνόλου $T \subseteq X \times Y \times Z$ διατεταγμένων τριάδων, υπάρχει σύνολο n τριάδων στο T έτσι ώστε κάθε στοιχείου του συνόλου $X \cup Y \cup Z$ να περιέχεται σε ακριβώς μία τριάδα;

Θεώρημα: Το 3-διάστατο πρόβλημα ταιριάσματος είναι NP-πλήρες.

k-χρωματισμός

Δοθέντος ενός γραφήματος G και ενός φράγματος k, ο G είναι k-χρωματίσιμος;

Θεώρημα: Το πρόβλημα του 3-χρωματισμού είναι NP-πλήρες.

Μερικά NP-πλήρη Προβλήματα

- Έξι βασικές κατηγορίες NP-πλήρων προβλημάτων και μερικά παραδείγματα.
  - Προβλήματα Συλλογών: SET-PACKING, INDEPENDENT SET.
  - Προβλήματα Καλύμματος: SET-COVER, VERTEX-COVER.
  - Προβλήματος Ικανοποίησης Περιορισμών: SAT, 3-SAT.
  - Προβλήματα Ακολουθίας: HAMILTONIAN-CYCLE, TSP.
  - Προβλήματα Διαμέρισης: 3D-MATCHING 3-COLOR.
  - Αριθμητικά Προβλήματα: SUBSET-SUM, KNAPSACK.
- Στην πράξη: Τα περισσότερα NP προβλήματα είτε ανήκουν στην P ή είναι NP-πλήρη.
- Εξαιρέσεις: Παραγοντοποίηση, ισομορφισμός γραφημάτων, Nash equilibrium.

Άσκηση

Είστε σύμβουλος μίας μικρής εταιρείας που διατηρεί ένα υπολογιστικό σύστημα ασφάλειας. Το σύστημα κρατάει logs που κρατά IP διευθύνσεις των χρηστών που προσπελαύνουν. Ας υποθέσουμε ότι σε κάθε λεπτό ένας χρήστης προσπελαύνει μόνο μία IP διεύθυνση. Στο log θα υπάρχει μία καταγραφή $I(u,m)$ που σημαίνει ότι ο χρήστης u το λεπτό m προσπέλασε την $I(u,m)$ IP. Αν ο u δεν προσπέλασε καμία IP την θέτει σε NIL.

Η εταιρεία έμαθε ότι κάποια μέρα το σύστημα χρησιμοποιήθηκε για μία επίθεση σε κάποια απομακρυσμένα συστήματα. Η επίθεση έγινε με προσπέλαση t διαφορετικών IP διευθύνσεων σε t διαδοχικά λεπτά. Στο λεπτό 1, η IP ήταν η i_1, στο λεπτό 2, η IP ήταν η i_2,…, στο λεπτό t, η IP ήταν η i_t. Ποιος έκανε την επίθεση;

Η εταιρεία έλεγξε το Log και βρήκε ότι δεν υπήρχε χρήστης με αυτά τα χαρακτηριστικά. Άρα, η επίθεση έγινε από μία ομάδα k χρηστών. Θα λέμε ότι μία ομάδα χρηστών είναι *ύποπτη* αν σε κάθε λεπτό της επίθεσης υπάρχει τουλάχιστον ένας που να προσπέλασε την αντίστοιχη IP.

Δοθέντων των $I(u,m)$ και ενός αριθμού k, υπάρχει ύποπτη ομάδα μεγέθους k;

Αντίκτυπος NP-Πληρότητας

- [Papadimitriou 1995]
  - Prime intellectual export of CS to other disciplines.
  - 6,000 citations per year (title, abstract, keywords).
    - more than "compiler", "operating system", "database"
  - Broad applicability and classification power.
  - "Captures vast domains of computational, scientific, mathematical endeavors, and seems to roughly delimit what mathematicians and scientists had been aspiring to compute feasibly."

- NP-completeness can guide scientific inquiry.
  - 1926: Ising introduces simple model for phase transitions.
  - 1944: Onsager solves 2D case in tour de force.
  - 19xx: Feynman and other top minds seek 3D solution.
  - 2000: Istrail proves 3D problem NP-complete.

Παραδείγματα NP-Πληρότητας

- Aerospace engineering: optimal mesh partitioning for finite elements.
- Biology: protein folding.
- Chemical engineering: heat exchanger network synthesis.
- Civil engineering: equilibrium of urban traffic flow.
- Economics: computation of arbitrage in financial markets with friction.
- Electrical engineering: VLSI layout.
- Environmental engineering: optimal placement of contaminant sensors.
- Financial engineering: find minimum risk portfolio of given return.
- Game theory: find Nash equilibrium that maximizes social welfare.
- Genomics: phylogeny reconstruction.
- Mechanical engineering: structure of turbulence in sheared flows.
- Medicine: reconstructing 3-D shape from biplane angiocardiogram.
- Operations research: optimal resource allocation.
- Physics: partition function of 3-D Ising model in statistical mechanics.
- Politics: Shapley-Shubik voting power.
- Pop culture: Minesweeper consistency.
- Statistics: optimal experimental design.