Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών
Εθνικό Μετσόβιο Πολυτεχνείο

Θεμελιώδη Θέματα Επιστήμης Υπολογιστών

Μετασχηματισμοί Υπολογιστικών Προβλημάτων

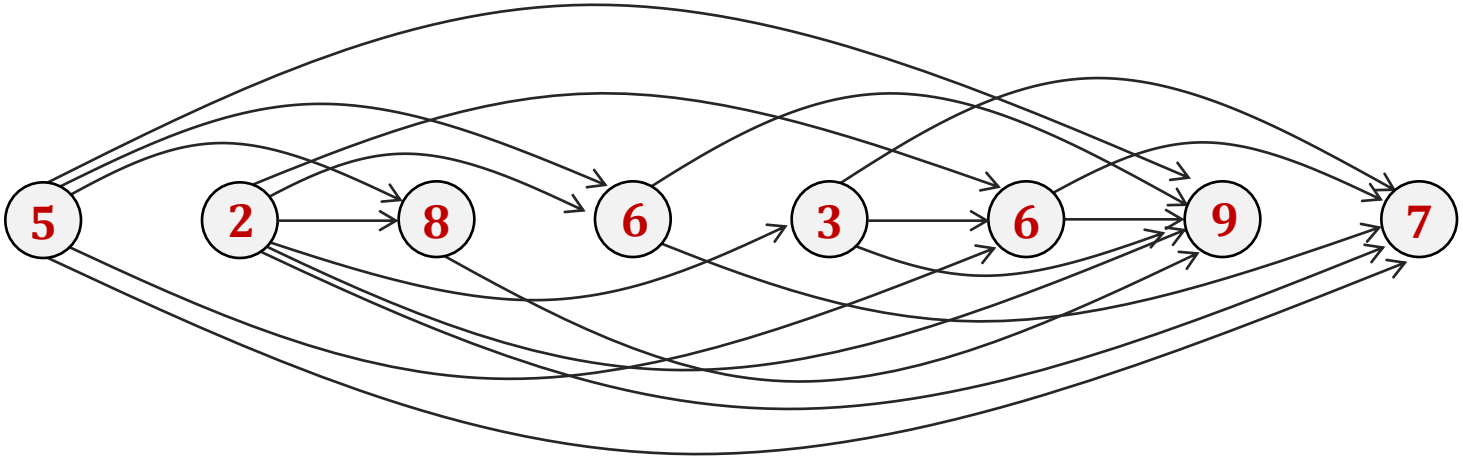
Αναγωγές και Πληρότητα

Προσαρμογή από διαφάνειες
Σταύρου Νικολόπουλου
(Παν. Ιωαννίνων)

Αναγωγές

- **Μεγιστη-Αυξουσα-Υπακολουθια ≤ Μεγιστη-Διαδρομη-DAG**

A = (5, 2, 8, 6, 3, 6, 9, 7)

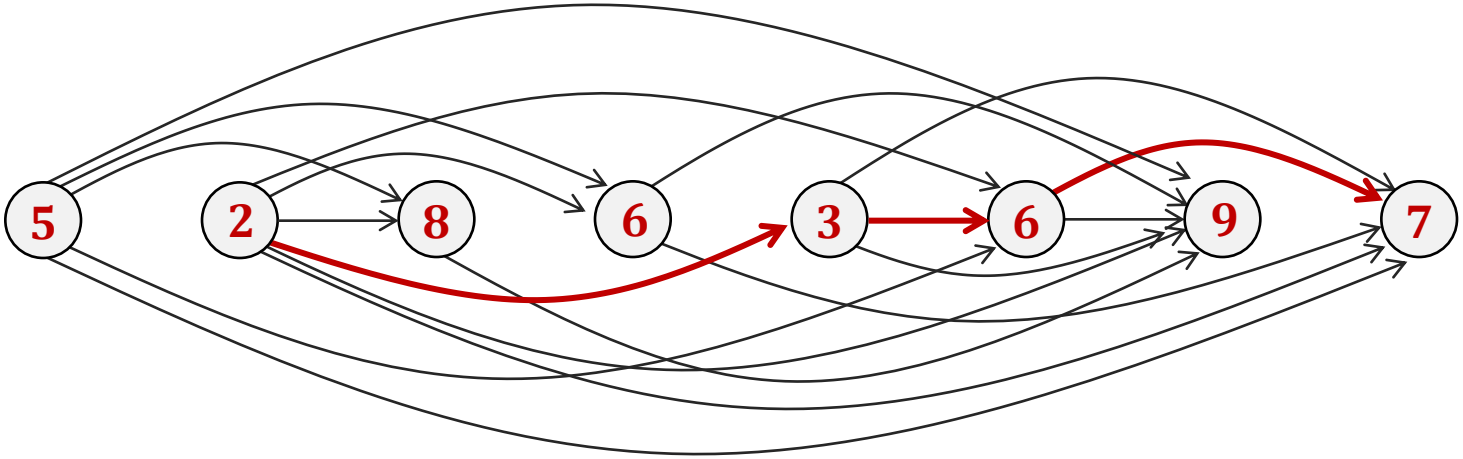


Δημιουργούμε ένα **G** με **ΟΛΕΣ** τις **επιτρεπτές μεταβάσεις** !!!

Αναγωγές

- **Μεγιστη-Αυξουσα-Υπακολουθια ≤ Μεγιστη-Διαδρομη-DAG**

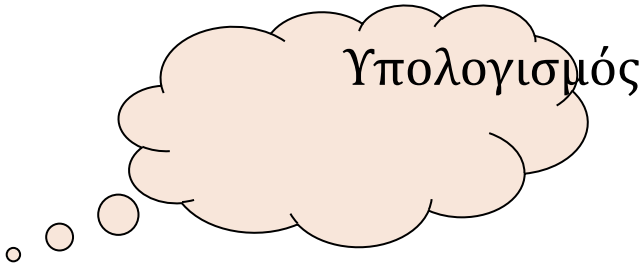
A = (5, 2, 8, 6, 3, 6, 9, 7)



Λύση: (2, 3, 6, 7)

Υπάρχει 1-1 αντιστοιχία ανάμεσα στις **αύξουσες υπακολουθίες** και στις **διαδρομές του DAG** !!!

Αναγωγές



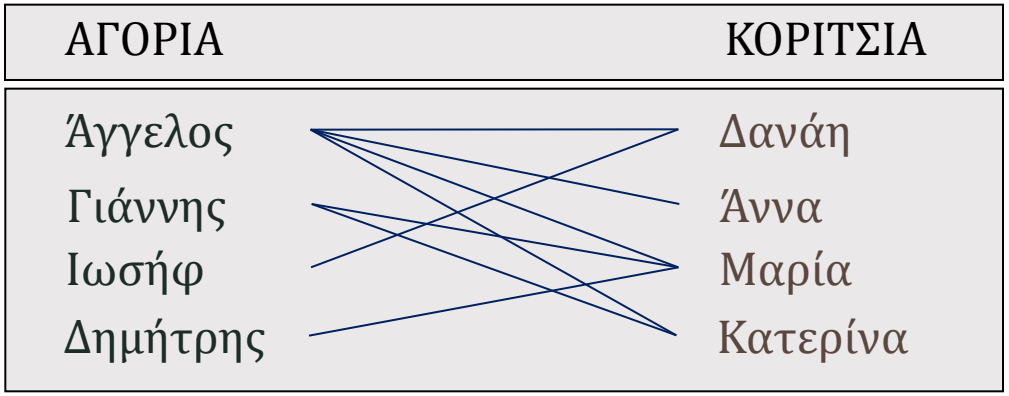
Τέλεια Αντιστοίχιση $\leq$ Μέγιστη Ροή

Πρόσφατα είδαμε ότι:

Μπορούμε να **«ανάγουμε»** το πρόβλημα της Τέλειας Αντιστοίχισης σε πρόβλημα Μεγίστης Ροής !!!

Αναγωγές

- **ΤΕΛΕΙΑ-ΑΝΤΙΣΤΟΙΧΙΣΗ ≤ ΜΕΓΙΣΤΗ-ΡΟΗ**



Πρόβλημα: Είναι δυνατόν να επιλεγούν ζευγάρια έτσι ώστε:

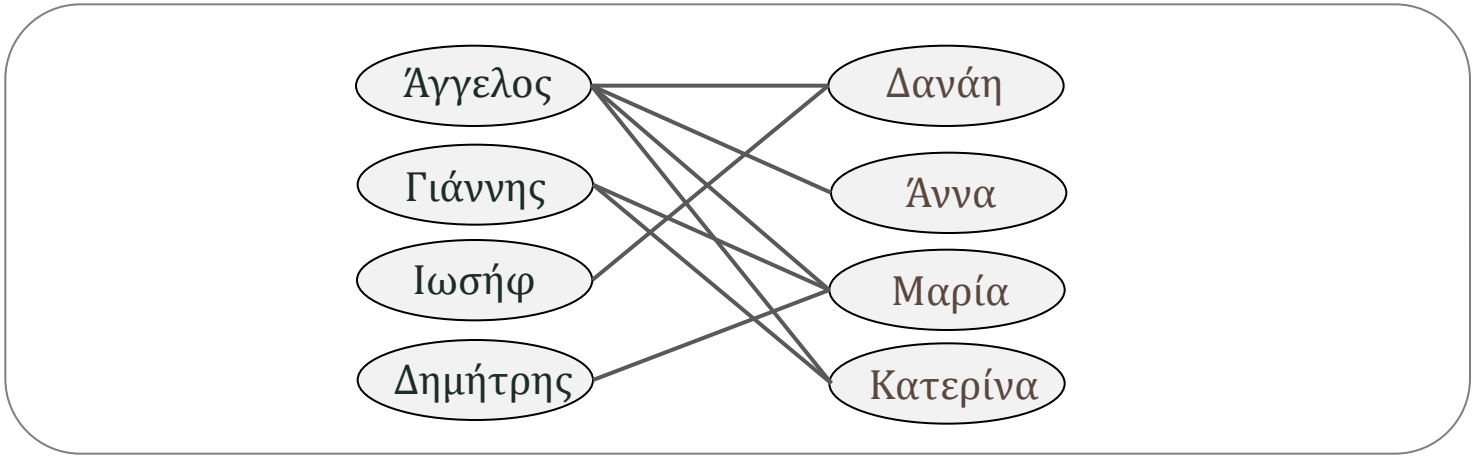
- ✓ όλοι να έχουν από ένα μόνο σύντροφο, και
- ✓ να είναι κάποιος που συμπαθούν !!!

Αναγωγές

Τελεια-Αντιστοιχιση ≤ Μεγιστη-Ροη

- Με όρους Γραφοθεωρίας το πρόβλημα διατυπώνεται ως εξής: Υπάρχει μια τέλεια αντιστοίχιση (perfect matching) ?

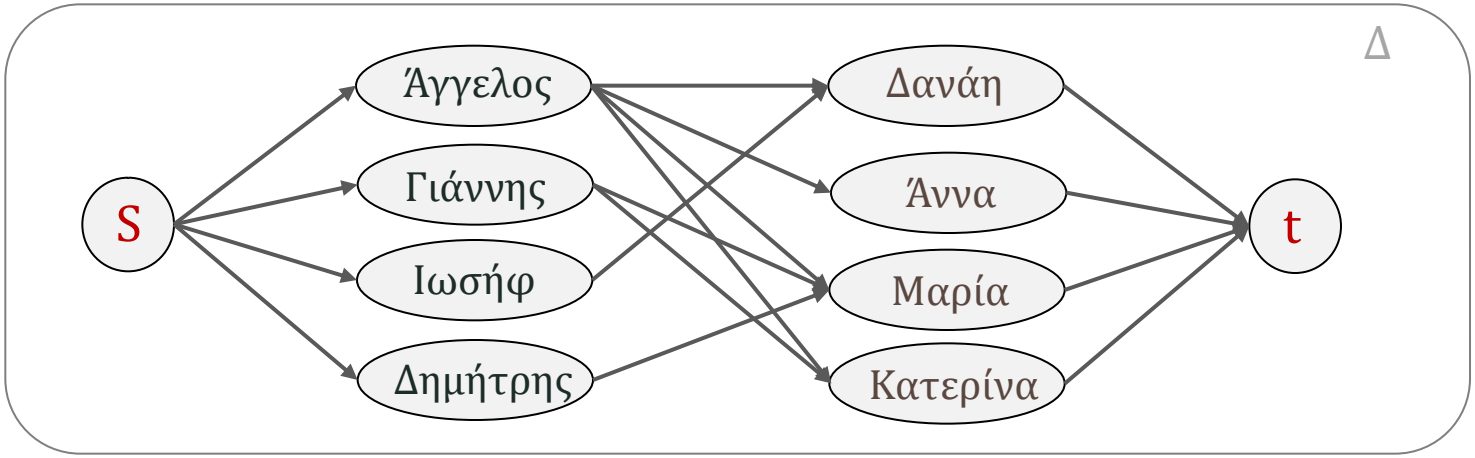
Αναγωγή !!!



Αναγωγές

- **Τελεια-Αντιστοιχιση ≤ Μεγιστη-Ροη**
  - Εισάγουμε 2 κόμβους **s** και **t**

    Εισάγουμε κατευθύνσεις και ορίζουμε το δίκτυο Δ !!!

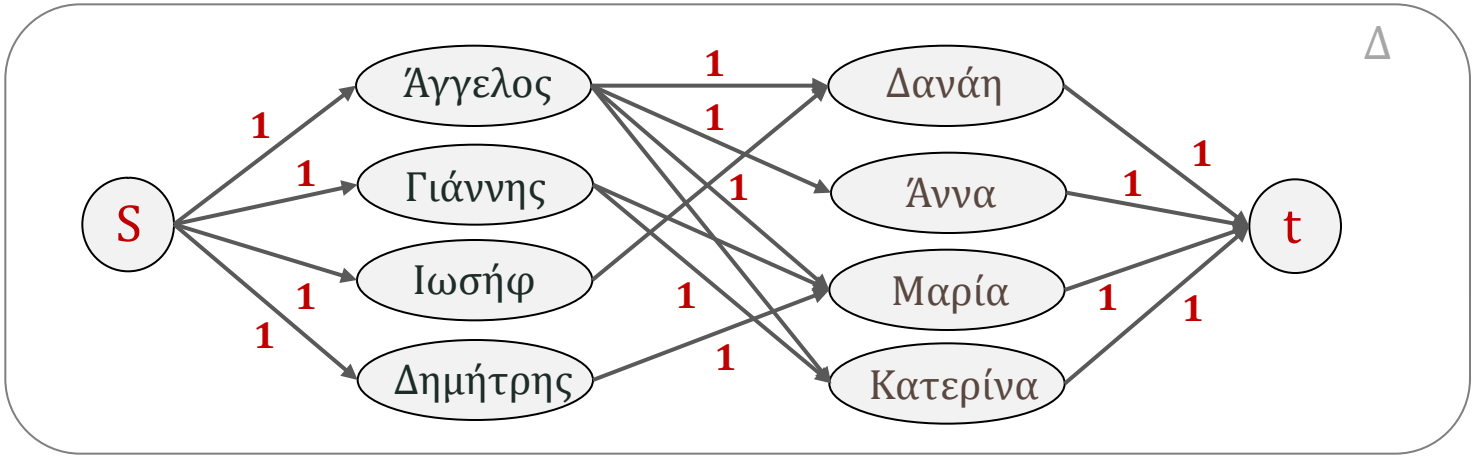


Αναγωγές

Τελεια-Αντιστοιχιση ≤ Μεγιστη-Ροη

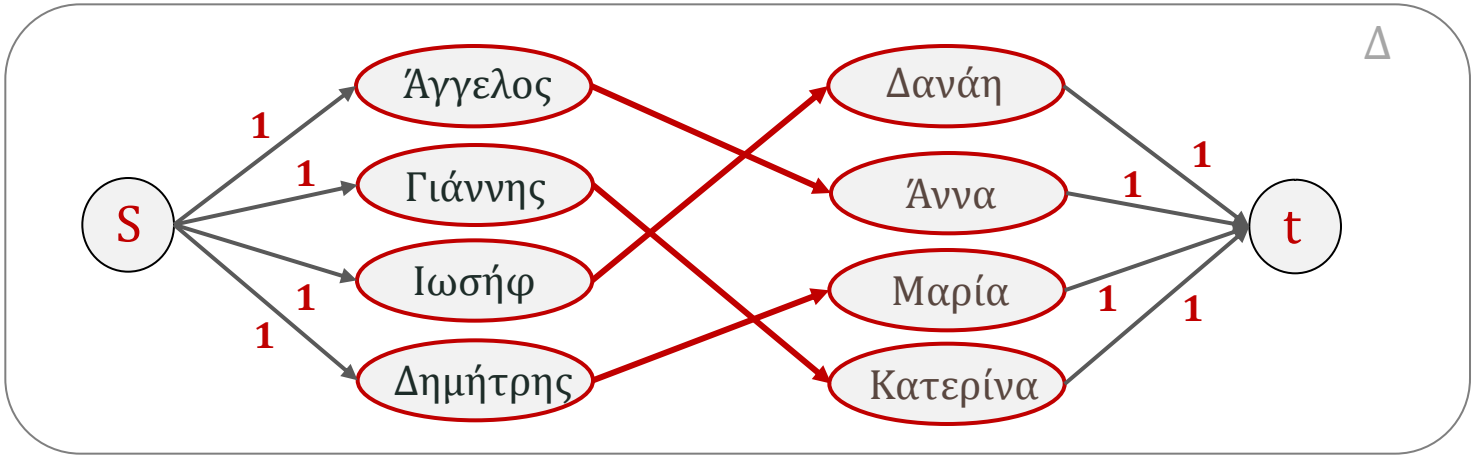
- Δίνουμε σε κάθε ακμή *χωρητικότητα* **1**

  Και τότε, υπάρχει τέλεια αντιστοίχιση **εάν-ν** το δίκτυο Δ δίνει μια ροή f με μέγεθος |f| = πλήθος των ζευγαριών !!!



Αναγωγές

- **Τελεια-Αντιστοιχιση ≤ Μεγιστη-Ροη**
  - Υπολογίζουμε μια Μέγιστη Ροή στο δίκτυο Δ

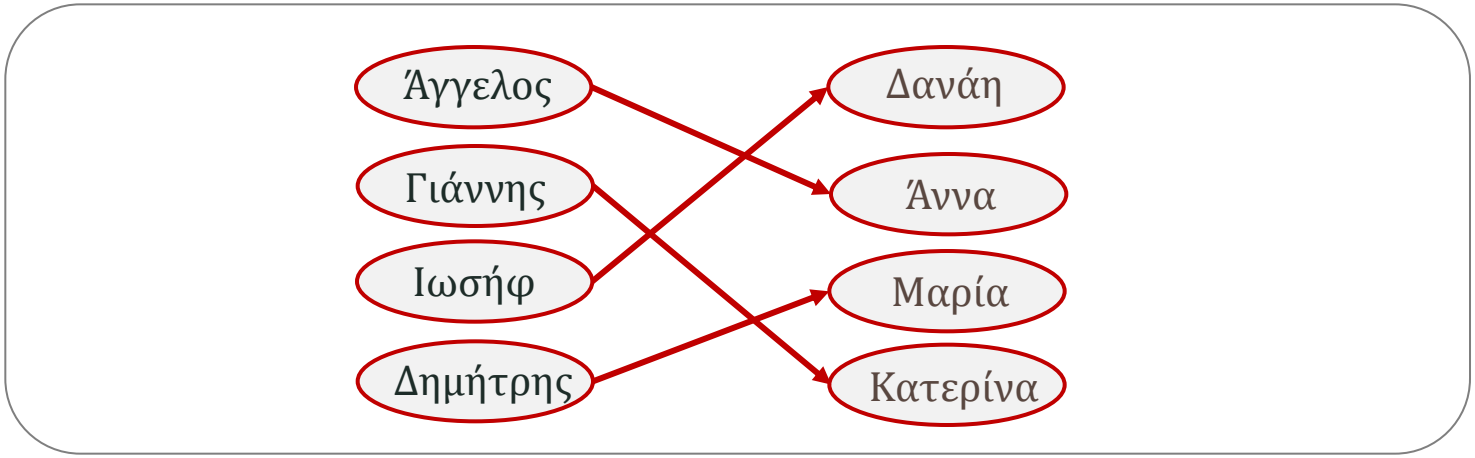


Αναγωγές

Τελεια-Αντιστοιχιση ≤ Μεγιστη-Ροη

- Υπολογίζουμε μια Μέγιστη Ροή στο δίκτυο Δ

  Και παίρνουμε μια τέλεια αντιστοίχιση !!!



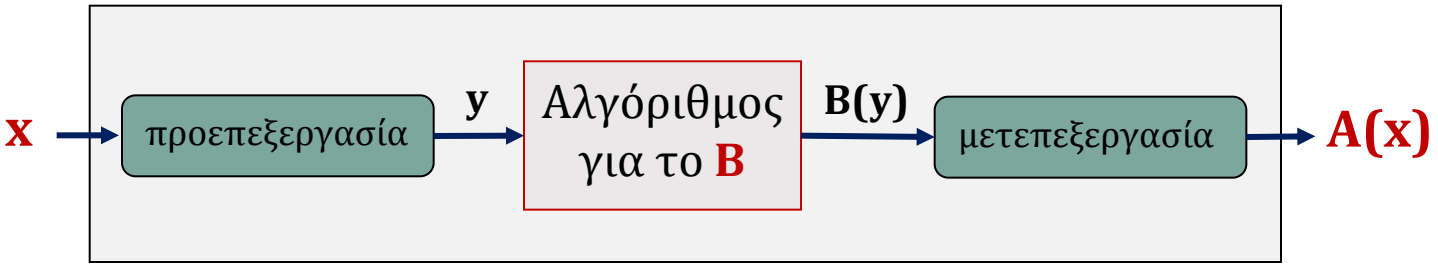
Ιδιότητα: Εάν όλες οι χωρητικότητες των ακμών είναι ακέραιες, τότε η βέλτιστη ροή είναι ακέραιη !!!

Αναγωγές

Μια πιο Τυπική Θεώρηση

- Λέμε ότι **A ≤ B** εάν μπορούμε να διατυπώσουμε έναν Αλγόριθμο πολυωνυμικού χρόνου για την επίλυση του **A** *χρησιμοποιώντας* έναν Αλγόριθμο για την επίλυση του **B** !!!

Αλγόριθμος για το A



Σημείωση: Στην πραγματικότητα, μπορεί να *έχουμε* ή και να *μην έχουμε* αλγόριθμο για το **B** !!!

Αναγωγές

Γιατί Αναγωγές ?

- **Δίνουν νέους αλγορίθμους**:

  εάν **A ≤ B** και έχουμε έναν αλγόριθμο για το **B**, τότε μπορούμε να τον χρησιμοποιήσουμε για να λύσουμε το πρόβλημα **A** !!!

- **Επιτρέπουν συγκρίσεις και κατηγοριοποιήσεις**:

  μια αναγωγή **A ≤ B** μας επιτρέπει να *συγκρίνουμε* τις πολυπλοκότητες των **A** και **B** και να τα *κατηγοριοποιήσουμε* ανάλογα με τη δυσκολία τους !!!

Αναγωγές:

- ✓ ΜΕΓΙΣΤΗ-ΑΥΞΟΥΣΑ-ΥΠΑΚΟΛΟΥΘΙΑ ≤ ΜΕΓΙΣΤΗ-ΔΙΑΔΡΟΜΗ-DAG
- ✓ ΤΕΛΕΙΑ-ΑΝΤΙΣΤΟΙΧΙΣΗ ≤ ΜΕΓΙΣΤΗ-ΡΟΗ
- ✓ ΜΕΓΙΣΤΗ-ΔΙΑΔΡΟΜΗ-DAG ≤ ΕΛΑΧΙΧΤΗ-ΔΙΑΔΡΟΜΗ-DAG

Αναγωγές

Γιατί Αναγωγές ?

- **Δίνουν νέους αλγορίθμους** :

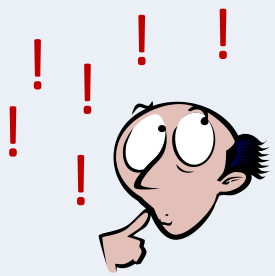
  εάν **A ≤ B** και έχουμε έναν αλγόριθμο για το **B,** τότε μπορούμε να τον χρησιμοποιήσουμε για να λύσουμε το πρόβλημα **A** !!!

- **Επιτρέπουν συγκρίσεις και κατηγοριοποιήσεις**:

  μια αναγωγή **A ≤ B** μας επιτρέπει να *συγκρίνουμε* τις πολυπλοκότητες των **A** και **B** και να *κατηγοριοποιήσουμε αυτά* ανάλογα με τη δυσκολία τους !!!

Γενικά, διευρύνουν:

- ✓ την αλγοριθμική μας ικανότητα !
- ✓ την επιστημονική μας σκέψη !!
- ✓ το πεδίο γνώσης της επιστήμης μας !!!



Αναγωγές

Γιατί Αναγωγές ?

- **Ειδικά, οι αναγωγές είναι χρήσιμες**:

  - ✓ Για *επινόηση* και *δημιουργία ΝΕΩΝ αλγορίθμων* για *ΝΕΑ προβλήματα* (άνω φράγματα)
  - ✓ Για απόδειξη ότι *ΚΑΠΟΙΑ προβλήματα δεν μπορούν να εκτελεστούν γρηγορότερα από ένα πολυωνυμικό χρόνο T(n)*

    Ή για εύρεση ενδείξεων ότι *δεν υπάρχουν αποτελεσματικοί αλγόριθμοι* για *ΚΑΠΟΙΑ άλλα προβλήματα* (κάτω φράγματα)

Αναγωγές

Άνω και Κάτω Φράγματα

- Έστω η αναγωγή **A ≤ B**

✓ **Άνω φράγμα** Εάν υπάρχει αλγόριθμος πολυωνυμικού χρόνου για το **B**, τότε υπάρχει ένας και για το **A !**

Το A είναι *το πολύ* ίδιας δυσκολίας με το B !

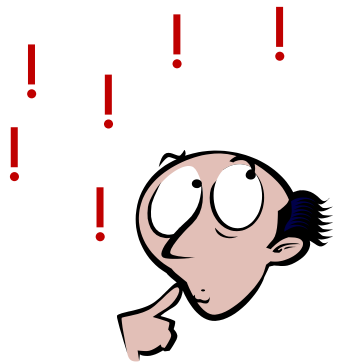
✓ **Κάτω φράγμα** Αντίστροφα, εάν δεν υπάρχει πολυωνυμικός αλγόριθμος για το **A,** τότε δεν υπάρχει ούτε για το **B !**

Το B είναι *τουλάχιστον* ίδιας δυσκολίας με το A !

Προβλήματα Αναζήτησης και Βελτιστοποίησης

Τι αλγορίθμους έχουμε δει έως τώρα ?

- Αναπτύξαμε αλγορίθμους για να βρίσκουμε:
  - ✓ ελάχιστες διαδρομές,
  - ✓ αντιστοιχίσεις σε διμερή γραφήματα,
  - ✓ μέγιστες ροές σε δίκτυα, και



γενικά, για να βρίσκουμε βέλτιστες (optimal) λύσεις !
με χρήση έξυπνων ***αλγοριθμικών τεχνικών*** και ***αναγωγών*** !!

- Όλοι αυτοί οι αλγόριθμοι είναι **αποδοτικοί** (efficient) !!!

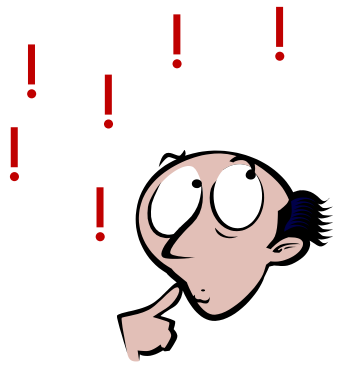
ο χρόνος εκτέλεσής τους $T(n)$ είναι πολυωνυμική συνάρτηση του *μεγέθους της εισόδου n* (όπως n, n^2, n^3,…)

Προβλήματα Αναζήτησης και Βελτιστοποίησης

Τι αναζητούμε σε όλα αυτά τα Προβλήματα ?

- Σε όλα **αναζητούμε μια λύση**, για παράδειγμα:
  - ✓ μια διαδρομή,
  - ✓ ένα δένδρο,
  - ✓ μια αντιστοίχιση, κλπ

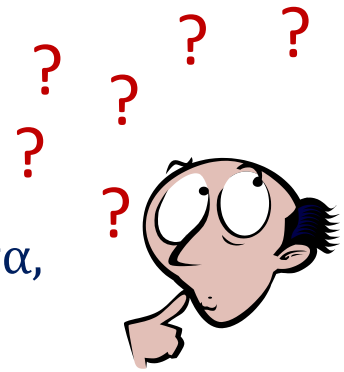
  ανάμεσα σε ένα **εκθετικό πλήθος** λύσεων !!!



- Πράγματι:
  - n αγόρια μπορούν να αντιστοιχηθούν με n κορίτσια με **n!** διαφορετικούς τρόπους !!!
  - ένα γράφημα τάξης n έχει $\mathbf{n^{n-2}}$ συνδετικά δένδρα
  - ένα τυπικό γράφημα τάξης n έχει **εκθετικό πλήθος** διαδρομών από τον κόμβο s στον t.

Προβλήματα Αναζήτησης και Βελτιστοποίησης

- **ΌΛΑ** αυτά τα **Προβλήματα Αναζήτησης και Βελτιστοποίησης** θα μπορούσαν να λυθούν σε **εκθετικό χρόνο !!!**
  - ✓ Με τον έλεγχο **όλων** των **υποψηφίων λύσεων**, μία προς μία !!!



Όμως, ένας αλγόριθμος με εκθετική πολυπλοκότητα, όπως $T(n) = 2^n$, είναι πρακτικά **άχρηστος !!!**

- **Τι Θέλουμε !!!**

Έξυπνες και Πρωτότυπες Ιδέες !!!

για να παρακάμψουμε την

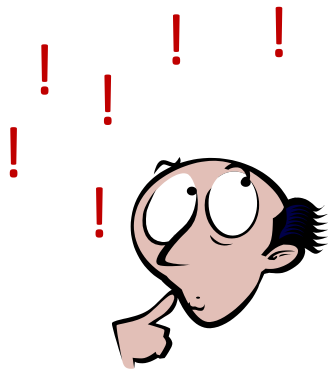
Εξαντλητική Αναζήτηση !!!

Προβλήματα Αναζήτησης και Βελτιστοποίησης

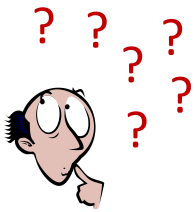
- **Υπάρχουν Έξυπνες και Πρωτότυπες Ιδέες ?**

  ΝΑΙ !!! έχουμε δει **αλγοριθμικές τεχνικές** που **«κατατροπώνουν»** τον **εφιάλτη** της **εκθετικότητας** !!!

  - ✓ Άπληστη τεχνική
  - ✓ Δυναμικός προγραμματισμός
  - ✓ Γραμμικός Προγραμματισμός



- **Κάποτε όμως έρχεται και η ώρα της…**

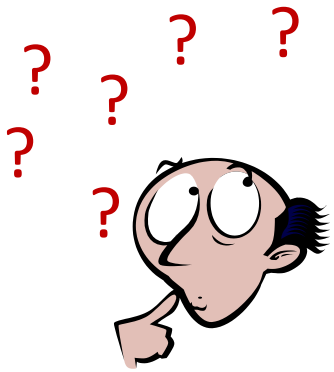


Αποτυχίας !!!

Προβλήματα Αναζήτησης και Βελτιστοποίησης

- **Τι σημαίνει «Αποτυχία» ?**
  - ✓ Υπάρχουν κάποια **«προβλήματα αναζήτησης»** για τα οποία :

    δεν φαίνεται να υπάρχει αποτελεσματικός τρόπος επίλυσης



- **Τι ξέρουμε για τα προβλήματα αυτά ?**

Οι **ταχύτεροι αλγόριθμοι** που γνωρίζουμε είναι όλοι **εκθετικοί !!!**

Ορίστε μερικά από αυτά !!!...

1 Ικανοποιησιμότητα

Πρόβλημα SAT

Μας δίδεται ένας λογικός τύπος (Boolean formula) σε κανονική συζευκτική μορφή (CNF), όπως για παράδειγμα:

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

και μας ζητείται είτε να βρούμε μια *ικανοποιούσα ανάθεση τιμών αληθείας* (satisfying truth assignment) ή να αναφέρουμε ότι δεν υπάρχει καμία !!!

Μια *ικανοποιούσα ανάθεση τιμών αληθείας* είναι μια ανάθεση τιμών **true** ή **false** σε κάθε μεταβλητή έτσι ώστε *ο λογικός τύπος να είναι* **true** **!!!**

2 Το πρόβλημα του Περιοδεύοντος Πωλητή

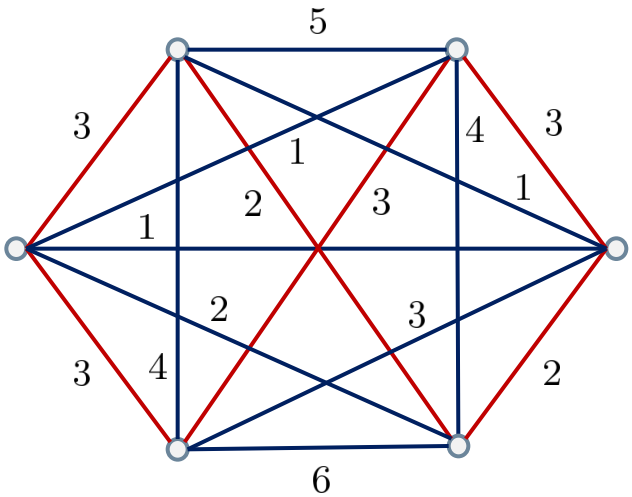
Πρόβλημα TSP

Μας δίδεται ένα πλήρες έμβαρο γράφημα **n** κόμβων, καθώς και ένας *προϋπολογισμός* **b**,

και μας ζητείται μια περιήγηση (tour), δηλαδή ένας κύκλος που διέρχεται από κάθε κόμβο μία μόνο φορά,

με συνολικό κόστος το πολύ **b**

ή να αναφέρουμε ότι δεν υπάρχει τέτοια περιήγηση !!!



Η βέλτιστη περιήγηση του παραδείγματος έχει κόστος 16 !!!

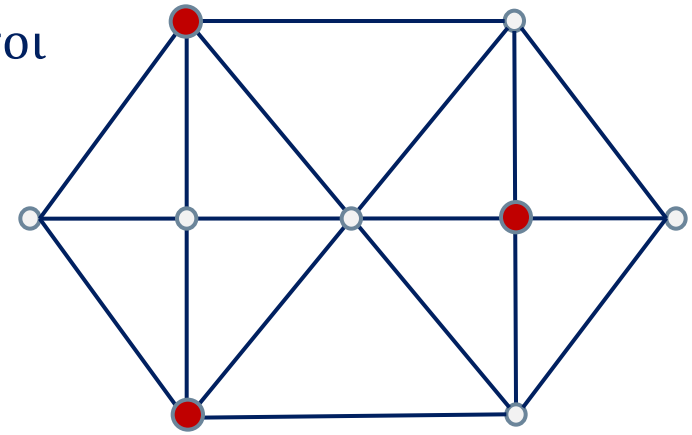
3 Ανεξάρτητα Σύνολα Γραφημάτων

Πρόβλημα ΑΝΕΞΑΡΤΗΤΟ-ΣΥΝΟΛΟ

Μας δίδεται ένα γράφημα **n** κόμβων και ένας *ακέραιος* **k**,

και μας ζητείται να βρούμε ένα σύνολο **IS** με **k** κόμβους οι οποίοι είναι ανεξάρτητοι (independent), δηλαδή ανά δύο δεν έχουν καμία ακμή μεταξύ τους,

ή να αναφέρουμε ότι δεν υπάρχει τέτοιο σύνολο κόμβων !!!



Το πρόβλημα λύνεται αποδοτικά σε δένδρα, αλλά όχι σε γενικά γραφήματα!!!

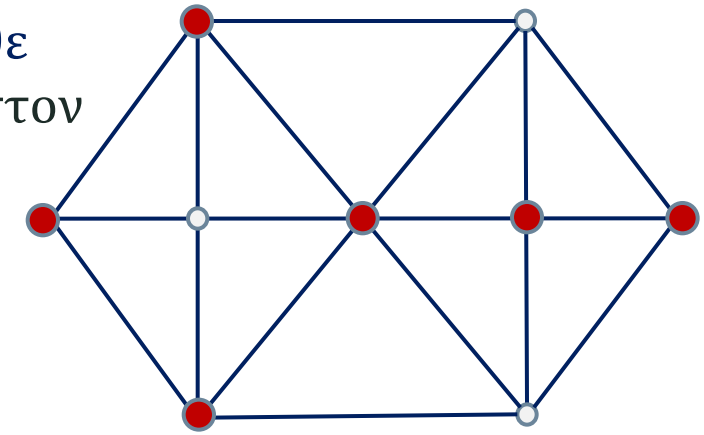
4 Κάλυψη Γραφήματος με Κόμβους

Πρόβλημα Κομβικη-Καλυψη

Μας δίδεται ένα γράφημα **n** κόμβων και ένας *ακέραιος* **k**,

και μας ζητείται να βρούμε ένα σύνολο **VC** με **k** κόμβους οι οποίοι καλύπτουν κάθε ακμή, δηλαδή κάθε ακμή έχει τουλάχιστον ένα κόμβο της στο σύνολο VC,

ή να αναφέρουμε ότι δεν υπάρχει τέτοιο σύνολο κόμβων !!!



Το πρόβλημα λύνεται αποδοτικά σε κάποιες κλάσεις γραφημάτων, αλλά όχι σε γενικά γραφήματα!!!

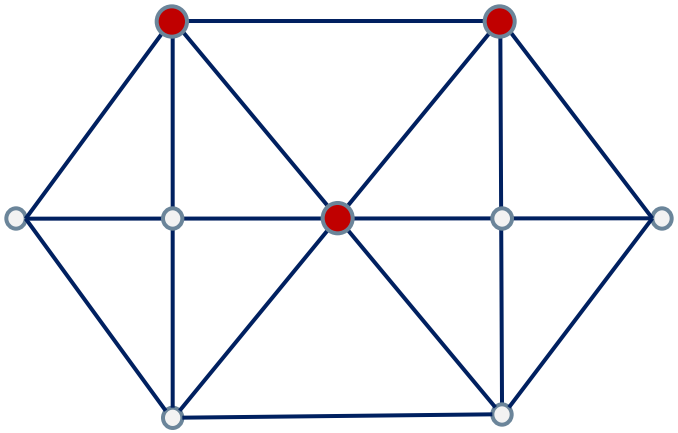
5 Κλίκες Γραφημάτων

Πρόβλημα Κλικα

Μας δίδεται ένα γράφημα **n** κόμβων και ένας *ακέραιος* **k**,

και μας ζητείται να βρούμε ένα σύνολο **CL** με **k** κόμβους οι οποίοι έχουν πλήρη σύνδεση, δηλαδή ανά δύο έχουν ακμή μεταξύ τους,

ή να αναφέρουμε ότι δεν υπάρχει τέτοιο σύνολο κόμβων !!!



Το πρόβλημα λύνεται αποδοτικά
σε κάποιες κλάσεις γραφημάτων, αλλά όχι σε γενικά γραφήματα!!!

6 Μέγιστες Διαδρομές

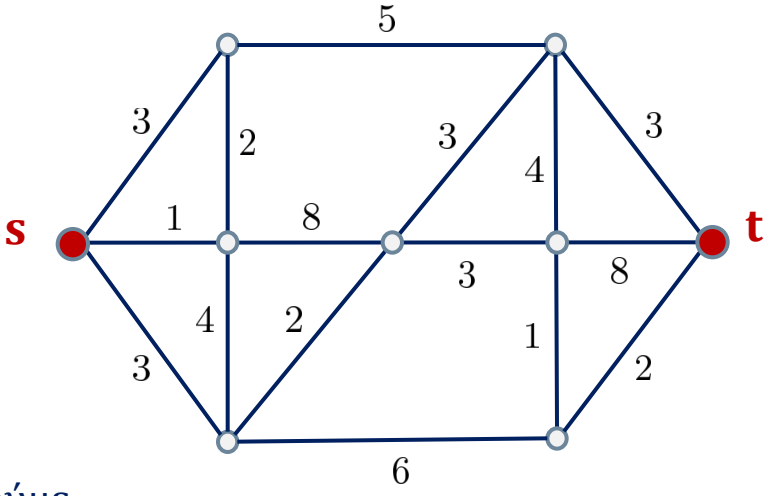
Πρόβλημα ΜΕΓΙΣΤΗ-ΔΙΑΔΡΟΜΗ

Μας δίδεται ένα έμβαρο γράφημα **n** κόμβων (με μη-αρνητικά βάρη) και δύο διακριτοί κόμβοι **s** και **t** μαζί με έναν *ακέραιο* **k**,

και μας ζητείται να βρούμε μια διαδρομή **Δ** από τον s στον t

με συνολικό κόστος τουλάχιστον **k**

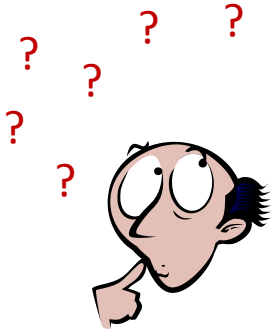
ή να αναφέρουμε ότι δεν υπάρχει τέτοια διαδρομή !!!



Για να αποφύγουμε τετριμμένες λύσεις απαιτούμε η διαδρομή να είναι *απλή*, δηλαδή να μην περιέχει επαναλαμβανόμενους κόμβους !!!

Κλάσεις Προβλημάτων

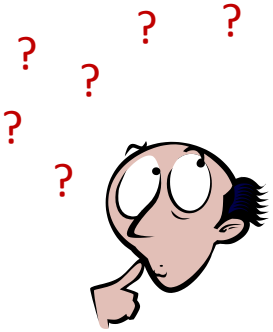
- **Δύσκολα & Εύκολα Προβλήματα !!!**



Δύσκολα Προβλήματα	Εύκολα Προβλήματα
3SAT	2SAT
TSP	MST
Μεγιστη Διαδρομη	Ελαχιστη Διαδρομη
Σακιδιο	Μοναδιαιο Σακιδιο
Ανεξαρτητο Συνολο	Ανεξαρτητο Συνολο σε Δενδρα
ILP	LP
Διαδρομη Hamilton	Διαδρομη Euler

Κλάσεις Προβλημάτων

Δύσκολα & Εύκολα Προβλήματα !!!



Δύσκολα Προβλήματα

3SAT
TSP
Μεγιστη Διαδρομη
Σακιδιο
Ανεξαρτητο Συνολο
ILP
Διαδρομη Hamilton

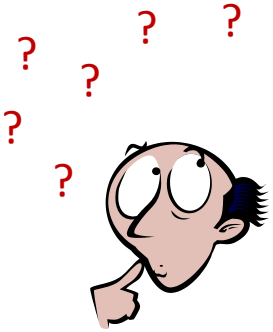
Εύκολα Προβλήματα

Όλα τα προβλήματα εδώ λύνονται **αποδοτικά** με αλγορίθμους:

- ✓ Αναζήτησης σε γραφήματα
- ✓ Άπληστων τεχνικών
- ✓ Ροών
- ✓ Δυναμικού προγραμματισμού
- ✓ Γραμμικού προγραμματισμού

Κλάσεις Προβλημάτων

Δύσκολα & Εύκολα Προβλήματα !!!



Δύσκολα Προβλήματα

Θλιβερή Αντίθεση !!!

Τα προβλήματα εδώ **είναι όλα δύσκολα**, όλα για τον ίδιο λόγο !!!

Στη βάση τους είναι **όλα το ίδιο πρόβλημα !!!**

με ***διαφορετικές μεταμφιέσεις*** !!!

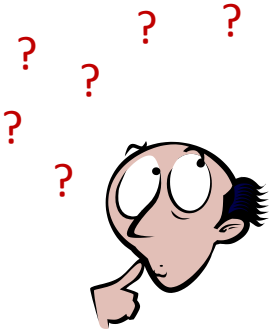
Εύκολα Προβλήματα

Όλα τα προβλήματα εδώ λύνονται **αποδοτικά** με αλγορίθμους:

- ✓ Αναζήτησης σε γραφήματα
- ✓ Άπληστων τεχνικών
- ✓ Ροών
- ✓ Δυναμικού προγραμματισμού
- ✓ Γραμμικού προγραμματισμού

Κλάσεις Προβλημάτων

Δύσκολα & Εύκολα Προβλήματα !!!



Δύσκολα Προβλήματα

Θλιβερή Αντίθεση !!!

Τα προβλήματα εδώ **είναι όλα δύσκολα**, όλα για τον ίδιο λόγο !!!

Στη βάση τους είναι **όλα το ίδιο πρόβλημα !!!**

NP-πλήρη !!!

Εύκολα Προβλήματα

Όλα τα προβλήματα εδώ λύνονται **αποδοτικά** με αλγορίθμους:

- ✓ Αναζήτησης σε γραφήματα
- ✓ Άπληστων τεχνικών
- ✓ Ροών
- ✓ Δυναμικού προγραμματισμού
- ✓ Γραμμικού προγραμματισμού

Κλάσεις Προβλημάτων

P και NP !!!

- **Γνωρίζουμε τι είναι ένα Πρόβλημα Αναζήτησης** !!!

Χαρακτηριστικό: Μια **προτεινόμενη λύση S** μπορεί να **ελεγχθεί γρήγορα** για την ορθότητά της !!!

Υπάρχει αποδοτικός (πολυωνυμικός) **αλγόριθμος ελέγχου C**

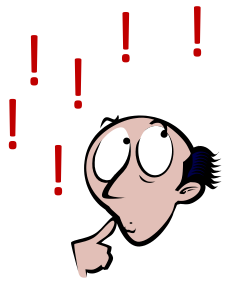
✓ Ο χρόνος εκτέλεσης του αλγόριθμου **C(I, S)** φράσσεται από ένα πολυώνυμο ως προς **|I|** , όπου I ένα στιγμιότυπο.

Συμβολίζουμε *την κλάση* ***Όλων*** *των* ***Προβλημάτων Αναζήτησης*** *που* ***μια Λύση τους Ελέγχεται*** *σε* ***Πολυωνυμικο Χρονο*** *με* **NP !!!**

Κλάσεις Προβλημάτων

P και NP !!!

- **Γνωρίζουμε τι είναι ένα Πρόβλημα Αναζήτησης** !!!

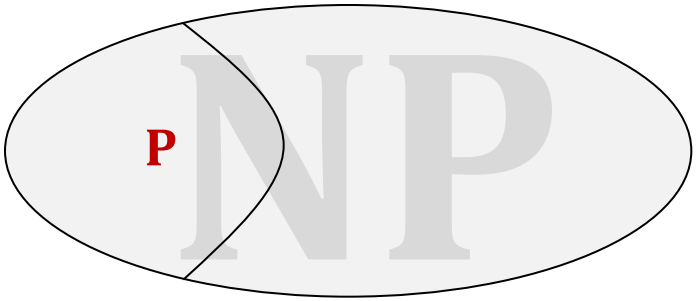


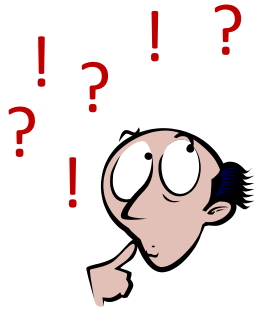
Συμβολίζουμε *την κλάση* ***Όλων*** *των* ***Προβλημάτων Αναζήτησης*** *που* ***μια Λύση τους Ελέγχεται*** *σε* ***Πολυωνυμικο Χρονο*** *με* **NP !!!**

Κλάσεις Προβλημάτων

P και NP !!!

- Γνωρίζουμε **ΠΟΛΛΑ πρόβλημα αναζήτησης NP** τα οποία μπορούν να λυθούν σε **πολυωνυμικό χρόνο** !!!



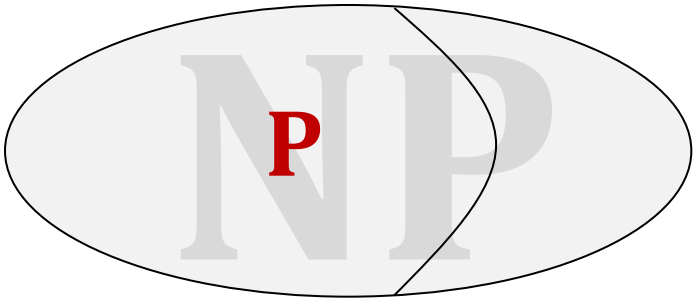


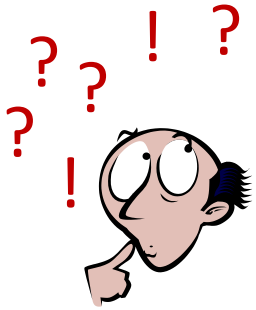
Συμβολίζουμε *την κλάση* ***Όλων*** *των* ***Προβλημάτων Αναζήτησης*** *που* ***Μπορούν να Λυθούν*** *σε* ***Πολυωνυμικο Χρονο*** *με* **P !!!**

Κλάσεις Προβλημάτων

Το Μεγάλο Ερώτημα !!!

- **Πόσα ΠΟΛΛΑ πρόβλημα αναζήτησης NP** υπάρχουν που μπορούν να λυθούν σε **πολυωνυμικό χρόνο ?**



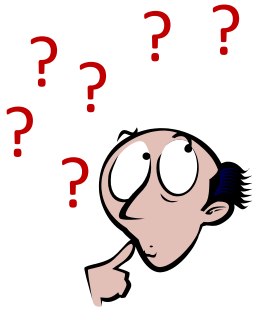


Συμβολίζουμε *την κλάση* ***Όλων*** *των* ***Προβλημάτων Αναζήτησης*** *που* ***Μπορούν να Λυθούν*** *σε* ***Πολυωνυμικο Χρονο*** *με* **P !!!**

Κλάσεις Προβλημάτων

Το Μεγάλο Ερώτημα !!!

- **Μήπως ΟΛΑ τα πρόβλημα αναζήτησης NP** μπορούν να λυθούν σε **πολυωνυμικό χρόνο ?**



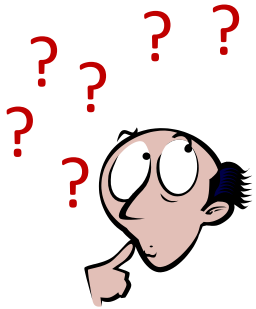
Συμβολίζουμε *την κλάση* ***Όλων*** *των* ***Προβλημάτων Αναζήτησης*** *που* ***Μπορούν να Λυθούν*** *σε* ***Πολυωνυμικο Χρονο*** *με* **P !!!**

Κλάσεις Προβλημάτων

Το Μεγάλο Ερώτημα !!!

- **Μήπως ΟΛΑ τα πρόβλημα αναζήτησης NP** μπορούν να λυθούν σε **πολυωνυμικό χρόνο ?**

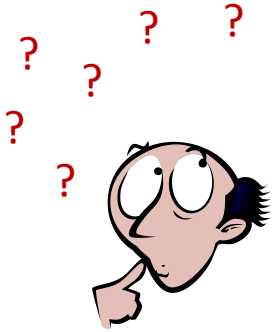
P = NP



Συμβολίζουμε *την κλάση* ***Όλων*** *των* ***Προβλημάτων Αναζήτησης*** *που* ***Μπορούν να Λυθούν*** *σε* ***Πολυωνυμικο Χρονο*** *με* **P !!!**

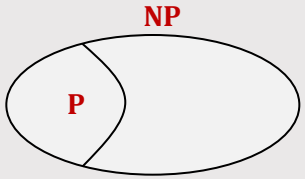
Κλάσεις Προβλημάτων

Το Μεγάλο Ερώτημα !!!



- Υπάρχουν στο **NP** **πρόβλημα αναζήτησης** τα οποία **δεν μπορούν** να λυθούν σε **πολυωνυμικό χρόνο ?**

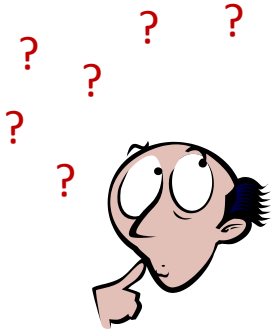
Ισχύει $\mathbf{P \neq NP}$ **?**



- Οι περισσότεροι ερευνητές αλγορίθμων πιστεύουν πως **Ναι** !!!
  - ✓ Είναι δύσκολο να πιστέψουμε ότι μπορεί πάντοτε να αποφευχθεί η εκθετική αναζήτηση !...
  - ✓ Ή, ότι μια τεχνική θα «σπάσει» όλα αυτά τα δύσκολα προβλ/τα !
  - ✓ Υπάρχει ένας καλός λόγος για να πιστεύουν οι μαθ/κοί ότι $P \neq NP$!

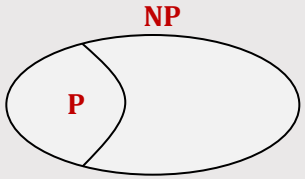
Κλάσεις Προβλημάτων

Το Μεγάλο Ερώτημα !!!



- Υπάρχουν στο **NP πρόβλημα αναζήτησης** τα οποία **δεν μπορούν** να λυθούν σε **πολυωνυμικό χρόνο ?**

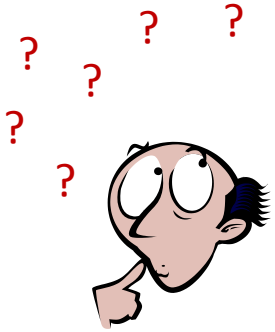
Ισχύει $\mathbf{P \neq NP}$ **?**



- Οι περισσότεροι ερευνητές αλγορίθμων πιστεύουν πως **Ναι** !!!
  - ✓ **Ωστόσο !... δεν υπάρχει απόδειξή** !... και η απόδειξη φαίνεται εξαιρετικά δύσκολη !!!
  - ✓ Το ερώτημα είναι ένα από τα **βαθύτερα** και **πιο σημαντικά αναπάντητα ερωτήματα** της επιστήμης μας **!!!**

Κλάσεις Προβλημάτων

Δύσκολα Προβλήματα !!!



- Εάν ισχύει: **P ≠ NP**

«Ζώνη Αβεβαιότητας» **NP-P**

Κλάσεις Προβλημάτων

Πάλι Αναγωγές !!!

- **Εάν δεχθούμε ότι ισχύει: P ≠ NP**

✓ Τι ισχύει για τα **προβλήματα αναζήτησης** του αριστερού πίνακα ?

✓ Υπάρχουν **ενδείξεις** ότι αυτά είναι ***ιδιαίτερα προβλήματα*** και ότι **δεν έχουν αποδοτικούς αλγορίθμους** επίλυσης !!!

3SAT	2SAT
TSP	MST
ΜΕΓΙΣΤΗ ΔΙΑΔΡΟΜΗ	ΕΛΑΧΙΣΤΗ ΔΙΑΔΡΟΜΗ
ΣΑΚΙΔΙΟ	ΜΟΝΑΔΙΑΙΟ ΣΑΚΙΔΙΟ
ΑΝΕΞΑΡΤΗΤΟ ΣΥΝΟΛΟ	ΑΝΕΞΑΡΤΗΤΟ ΣΥΝΟΛΟ ΣΕ ΔΕΝΔΡΑ
ILP	LP
ΔΙΑΔΡΟΜΗ HAMILTON	ΔΙΑΔΡΟΜΗ EULER

- **Τέτοιες ενδείξεις παρέχουν οι Αναγωγές**

Κλάσεις Προβλημάτων

Πάλι Αναγωγές !!!

- Θα δείξουμε, **μέσω αναγωγών**, ότι :

  όλα τα **προβλήματα του πίνακα**, κατά μία έννοια (υπολογιστικά),

  - ✓ είναι ακριβώς το ίδιο πρόβλημα !
  - ✓ είναι τα δυσκολότερα στην κλάση NP !
  - ✓ εάν ένα λυθεί πολυωνυμικά, τότε κάθε πρόβλημα στην NP λύνεται πολυωνυμικά !

3SAT
TSP
Μεγιστη Διαδρομη
Σακιδιο
Ανεξαρτητο Συνολο
ILP
Διαδρομη Hamilton

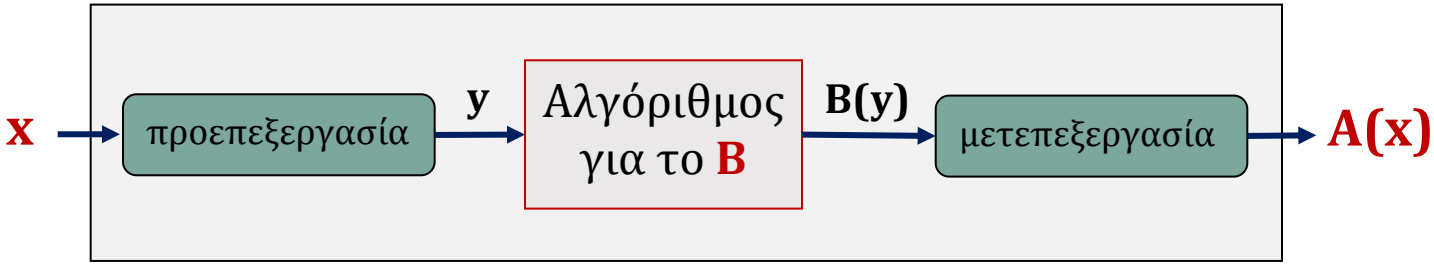
- Έτσι, εάν πιστεύουμε ότι **P≠NP** τότε όλα τα προβλήματα του πίνακα είναι **δύσκολα !**

Κλάσεις Προβλημάτων

Πάλι Αναγωγές !!!

- **Ας θυμηθούμε !!!...** οι αναγωγές **εκφράζουν** ένα **πρόβλημα αναζήτησης** σε κάποιο άλλο !!!
- **A ≤ B**

Αλγόριθμος για το A



ΤΕΛΕΙΑ-ΑΝΤΙΣΤΟΙΧΙΣΗ ≤ ΜΕΓΙΣΤΗ-ΡΟΗ

ΤΙΜΗ-ΚΥΚΛΩΜΑΤΟΣ ≤ LP

Κλάσεις Προβλημάτων

Πάλι Αναγωγές !!!

- **Ας θυμηθούμε !!!...** έως τώρα, **ο σκοπός της αναγωγής** ήταν **σαφής** και **έντιμος** !!!
- **A ≤ B**

Άνω Φράγματα

γνωρίζουμε έναν **αποδοτικό αλγόριθμο** για να **λύνουμε** το **B**, και τον ***χρησιμοποιούμε*** για να **λύσουμε αποδοτικά** το πρόβλημα **A** !!!

ΤΕΛΕΙΑ-ΑΝΤΙΣΤΟΙΧΙΣΗ ≤ ΜΕΓΙΣΤΗ-ΡΟΗ

ΤΙΜΗ-ΚΥΚΛΩΜΑΤΟΣ ≤ LP

ΜΕΓΙΣΤΗ-ΑΥΞΟΥΣΑ-ΥΠΑΚΟΛΟΥΘΙΑ ≤ ΜΕΓΙΣΤΗ-ΔΙΑΔΡΟΜΗ-DAG

Κλάσεις Προβλημάτων

Πάλι Αναγωγές !!!

- **Στη συνέχεια…** θα ***χρησιμοποιήσουμε*** τις **αναγωγές** για ένα *κάπως* ***διεστραμμένο*** *στόχο*:
- **A ≤ B**

Κάτω Φράγματα

γνωρίζουμε ότι το πρόβλημα **A** **είναι δύσκολο**,
και ***χρησιμοποιούμε*** την αναγωγή
για **να αποδείξουμε ότι** και το **B** είναι **δύσκολο** !!!

A ≤ B

→

Ροή Δυσκολίας

Κλάσεις Προβλημάτων

Πάλι Αναγωγές !!!

- **Στη συνέχεια…** θα ***χρησιμοποιήσουμε*** τις **αναγωγές** για ένα *κάπως* ***διεστραμμένο*** *στόχο*:
- **A ≤ B**

Κάτω Φράγματα

γνωρίζουμε ότι το πρόβλημα **A** **είναι δύσκολο**,
και ***χρησιμοποιούμε*** την αναγωγή
για **να αποδείξουμε ότι** και το **B** είναι **δύσκολο** !!!

Εάν **A ≤ B** και **B ≤ C** τότε **A ≤ C**

→

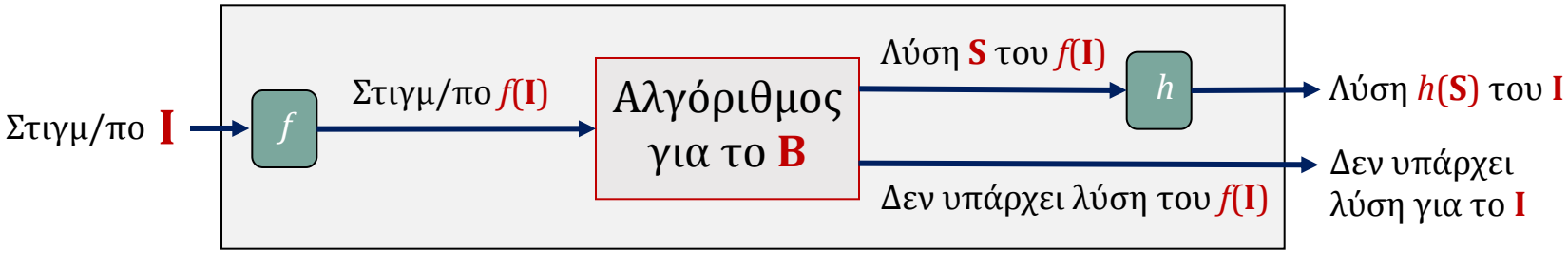
Ροή Δυσκολίας

Κλάσεις Προβλημάτων

Πάλι Αναγωγές !!!

- **Ας εξειδικεύσουμε…** τον ορισμό της **αναγωγής** για **προβλήματα αναζήτησης !!!**
- **A ≤ B**

Αλγόριθμος για το A



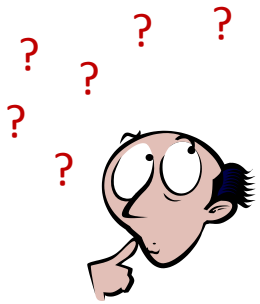
Μια αναγωγή A ≤ B είναι δύο πολυωνυμικοί αλγόριθμοι ***f*** και ***h***

f : ***μετασχηματίζει*** κάθε στιγμιότυπο I του A σε ένα στιγμιότυπο *f*(I) του B

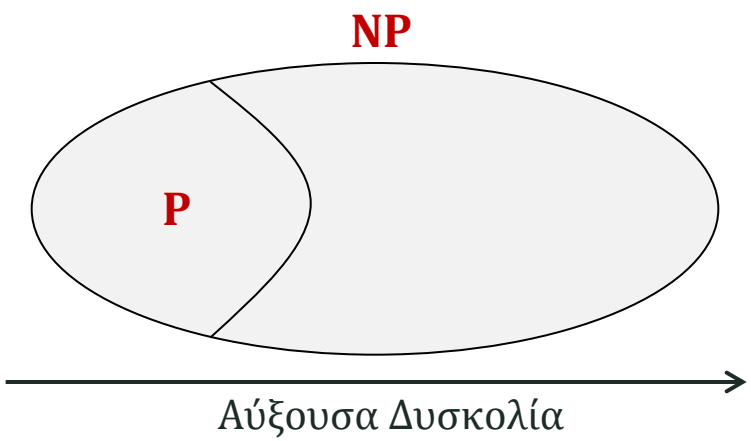
h : ***απεικονίζει*** μια λύση S του στιγμιότυπου *f*(I) σε μια λύση *h*(S) του I

Κλάσεις Προβλημάτων

NP και NP-πλήρη !!!



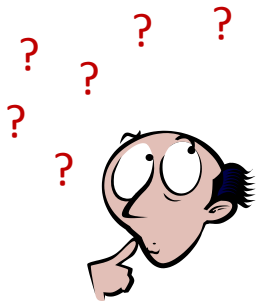
- **Και τώρα…** ορίζουμε την κλάση των **δυσκολότερων προβλημάτων αναζήτησης !**
- **A ≤ B**



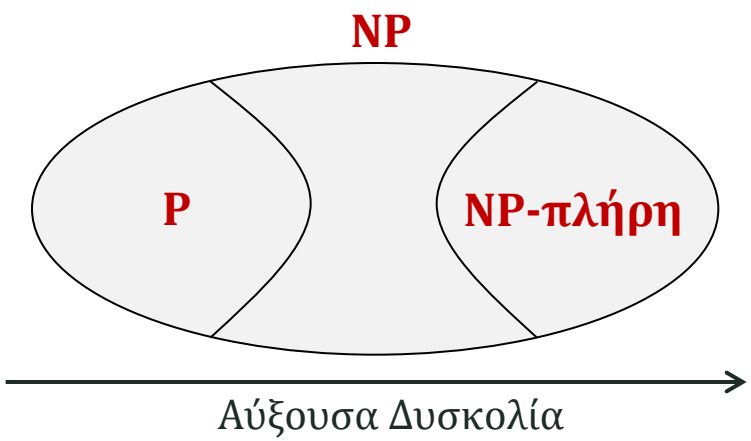
Ένα πρόβλημα αναζήτησης είναι **NP-πλήρες** (NP-complete) *εάν* ***ΌΛΑ ΤΑ ΆΛΛΑ ΠΡΟΒΛΗΜΑΤΑ ΑΝΑΖΗΤΗΣΗΣ ΣΤΟ*** **NP** ***ΑΝΑΓΟΝΤΑΙ*** *σε αυτό* **!!!**

Κλάσεις Προβλημάτων

NP και NP-πλήρη !!!



- **Και τώρα…** ορίζουμε την κλάση των **δυσκολότερων προβλημάτων αναζήτησης !**
- **Εάν P ≠ NP**

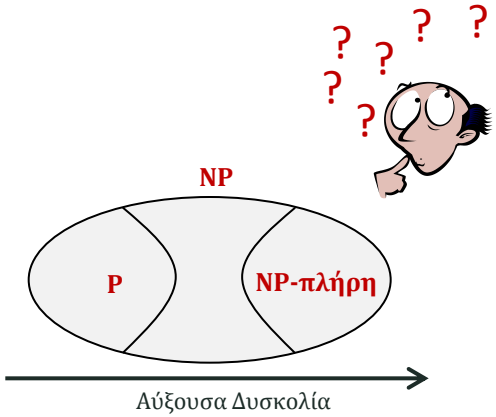


Ένα πρόβλημα αναζήτησης είναι **NP-πλήρες** (NP-complete) *εάν* ***ΌΛΑ ΤΑ ΆΛΛΑ ΠΡΟΒΛΗΜΑΤΑ ΑΝΑΖΗΤΗΣΗΣ ΣΤΟ*** **NP** ***ΑΝΑΓΟΝΤΑΙ*** *σε αυτό* **!!!**

Κλάσεις Προβλημάτων

NP και NP-πλήρη !!!

- **Αυτή…** είναι μια **πολύ ισχυρή απαίτηση !!!**



✓ Για να είναι ένα πρόβλημα **NP-πλήρες** πρέπει να είναι

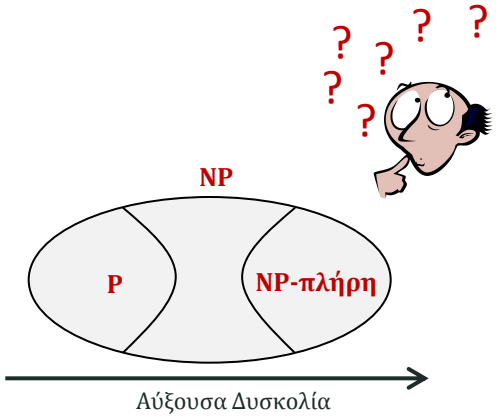
Χρήσιμο για την επίλυση ***κάθε προβλήματος* στον Κόσμο !!!**

✓ Είναι εντυπωσιακό, αλλά !!!…

Υπάρχουν *τέτοια προβλήματα* (3SAT, TSP, ΜΕΓΙΣΤΗ ΔΙΑΔΡΟΜΗ, …)

Κλάσεις Προβλημάτων

NP και NP-πλήρη !!!



- **Ισχύει:**

✓ Όλα τα **NP-πλήρη** πρόβλημα **ανάγονται** το ένα στο άλλο **!!!**

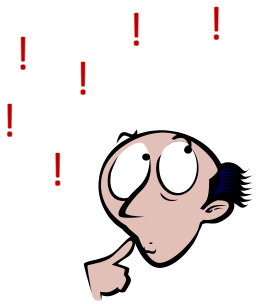
SAT ≤ ··· ≤ Q ≤ ··· ≤ R

✓ Και, όλα τα **NP** προβλήματα **ανάγονται** στα **NP-πλήρη !!!**

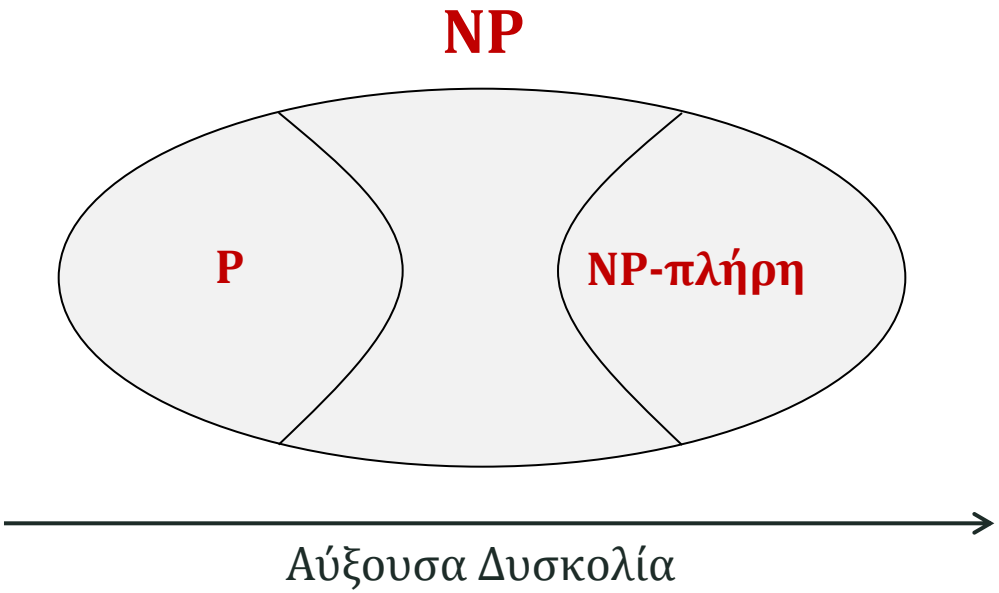
ΟΠΟΙΟΔΗΠΟΤΕ-ΠΡΟΒΛΗΜΑ-ΣΤΟ-NP ≤ SAT

Κλάσεις Προβλημάτων

P, NP και NP-πλήρη !!!



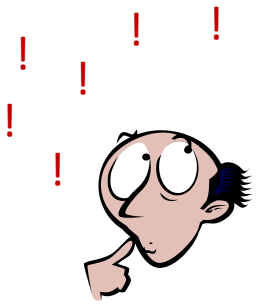
- Εάν **P ≠ NP**



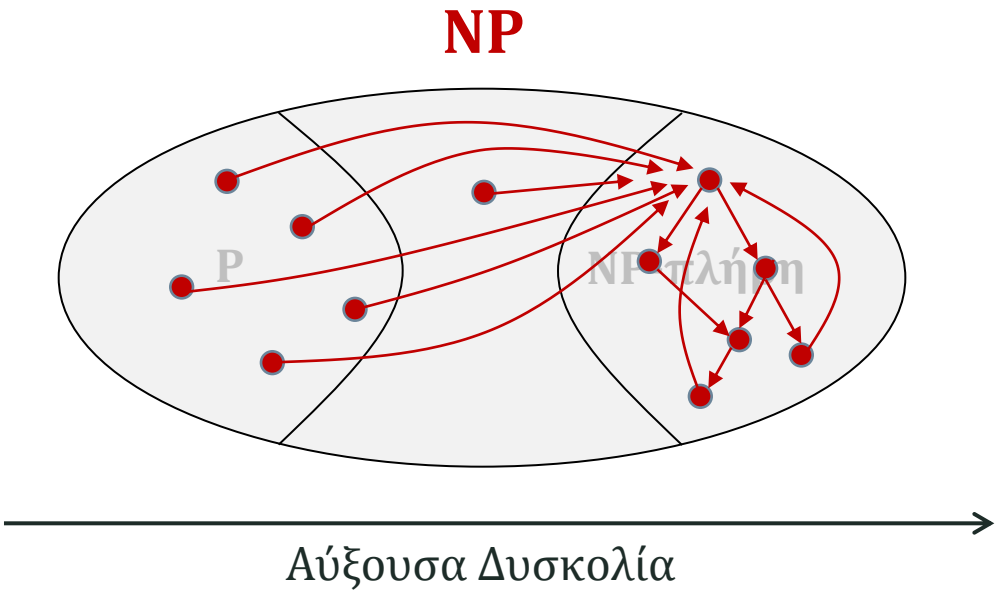
Μια Σχηματική Παρουσίαση !!!

Κλάσεις Προβλημάτων

P, NP και NP-πλήρη !!!



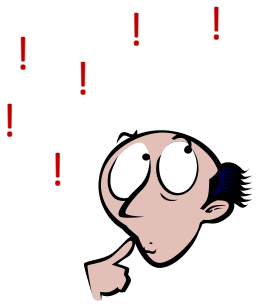
- Εάν **P ≠ NP**



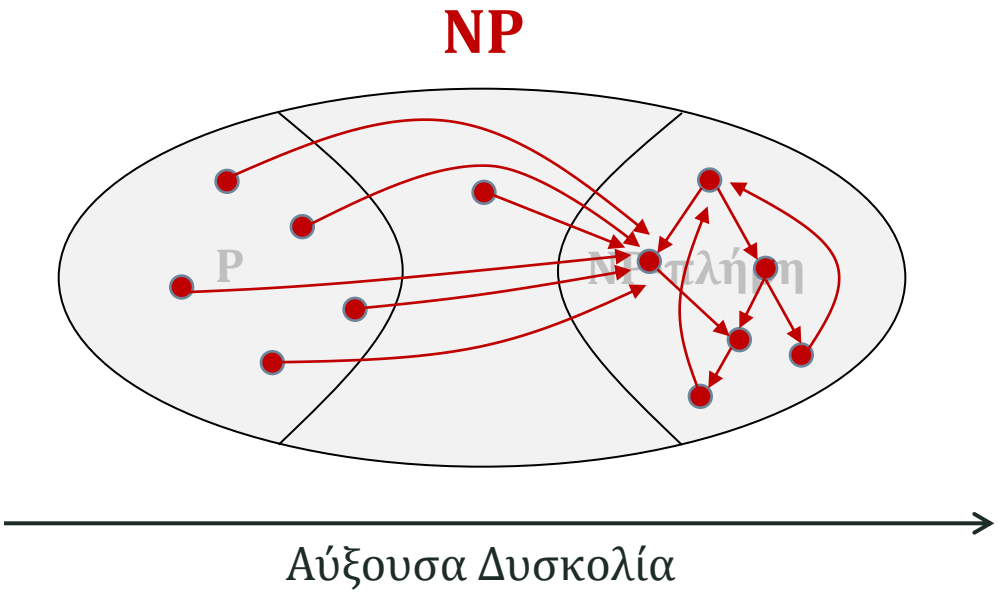
ΟΠΟΙΟΔΗΠΟΤΕ-ΠΡΟΒΛΗΜΑ-ΣΤΟ-NP ≤ SAT ≤ ··· ≤ Q ≤ ··· ≤ R

Κλάσεις Προβλημάτων

P, NP και NP-πλήρη !!!



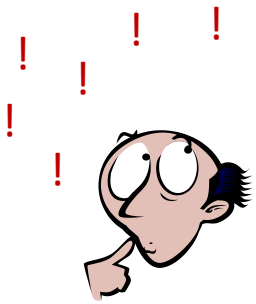
- Εάν **P ≠ NP**



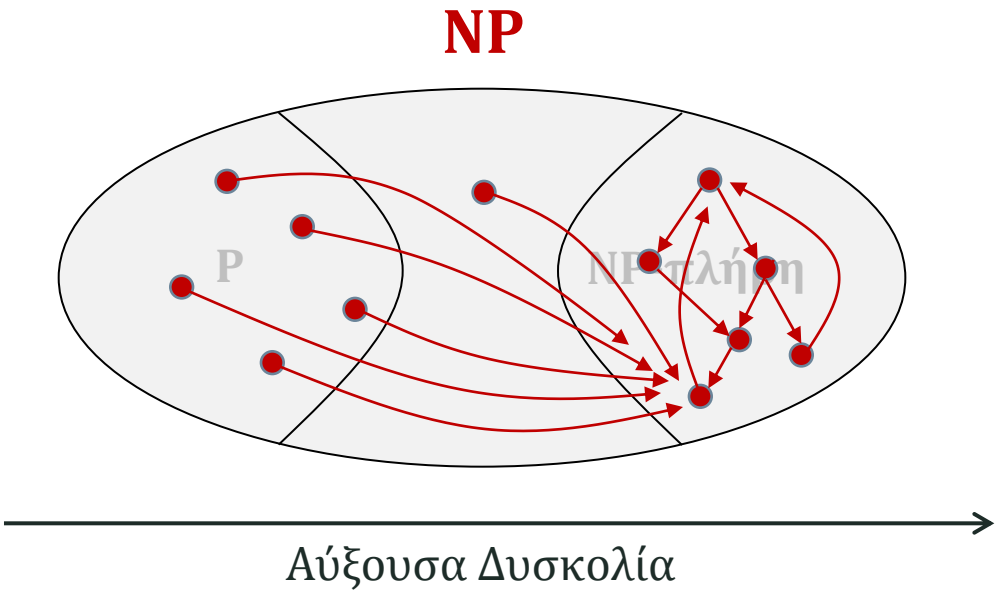
ΟΠΟΙΟΔΗΠΟΤΕ-ΠΡΟΒΛΗΜΑ-ΣΤΟ-NP ≤ SAT ≤ ··· ≤ Q ≤ ··· ≤ R

Κλάσεις Προβλημάτων

P, NP και NP-πλήρη !!!



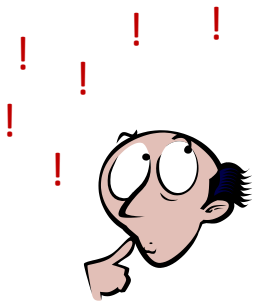
- Εάν **P ≠ NP**



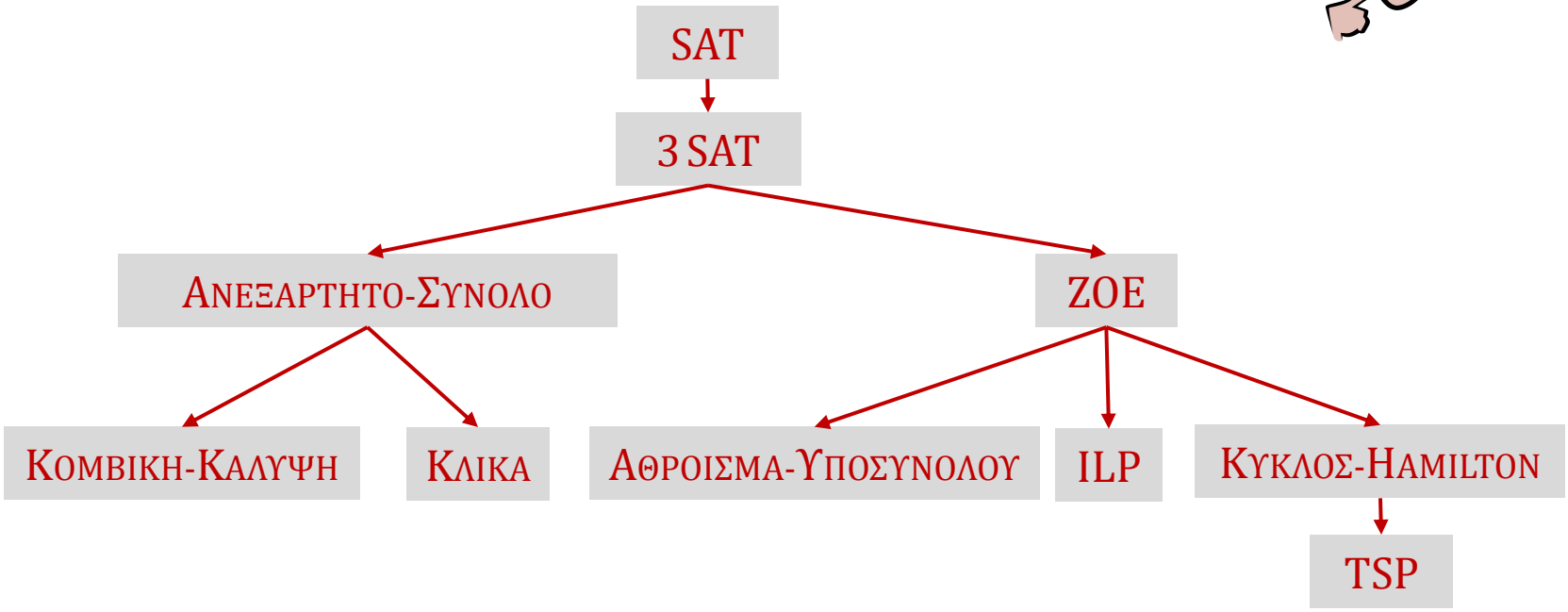
ΟΠΟΙΟΔΗΠΟΤΕ-ΠΡΟΒΛΗΜΑ-ΣΤΟ-NP ≤ SAT ≤ ··· ≤ Q ≤ ··· ≤ R

Αναγωγές Προβλημάτων NP

- $\mathbf{A \leq B \Leftrightarrow A \rightarrow B}$



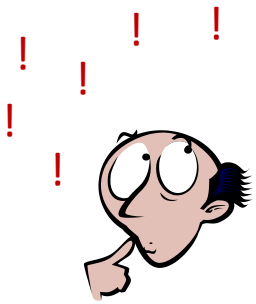
Οποιοδηποτε-Προβλημα-στο-**NP**



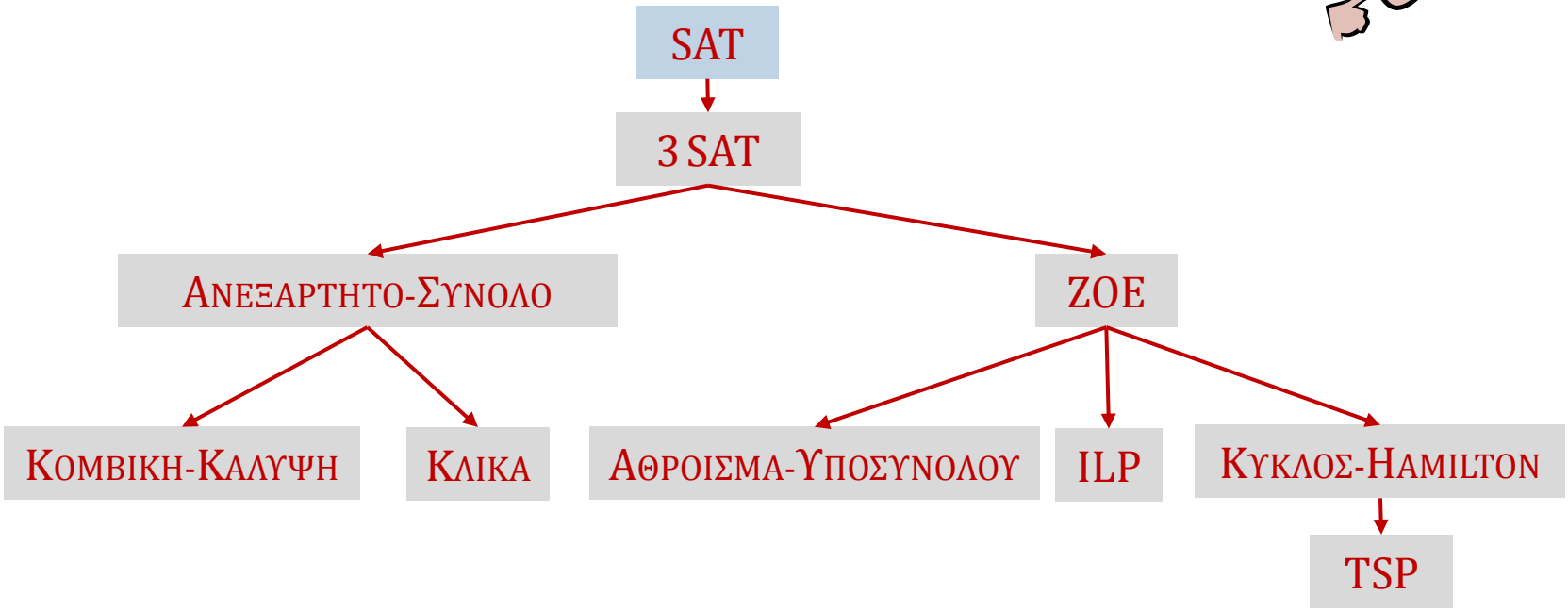
Θα Δείξουμε μέσω Αναγωγών !!!

Αναγωγές Προβλημάτων NP

- $\mathbf{A \leq B \Leftrightarrow A \rightarrow B}$



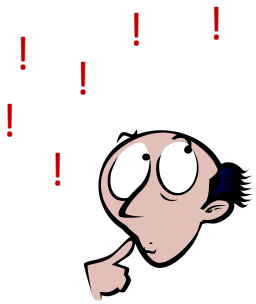
Οποιοδηποτε-Προβλημα-στο-**NP**



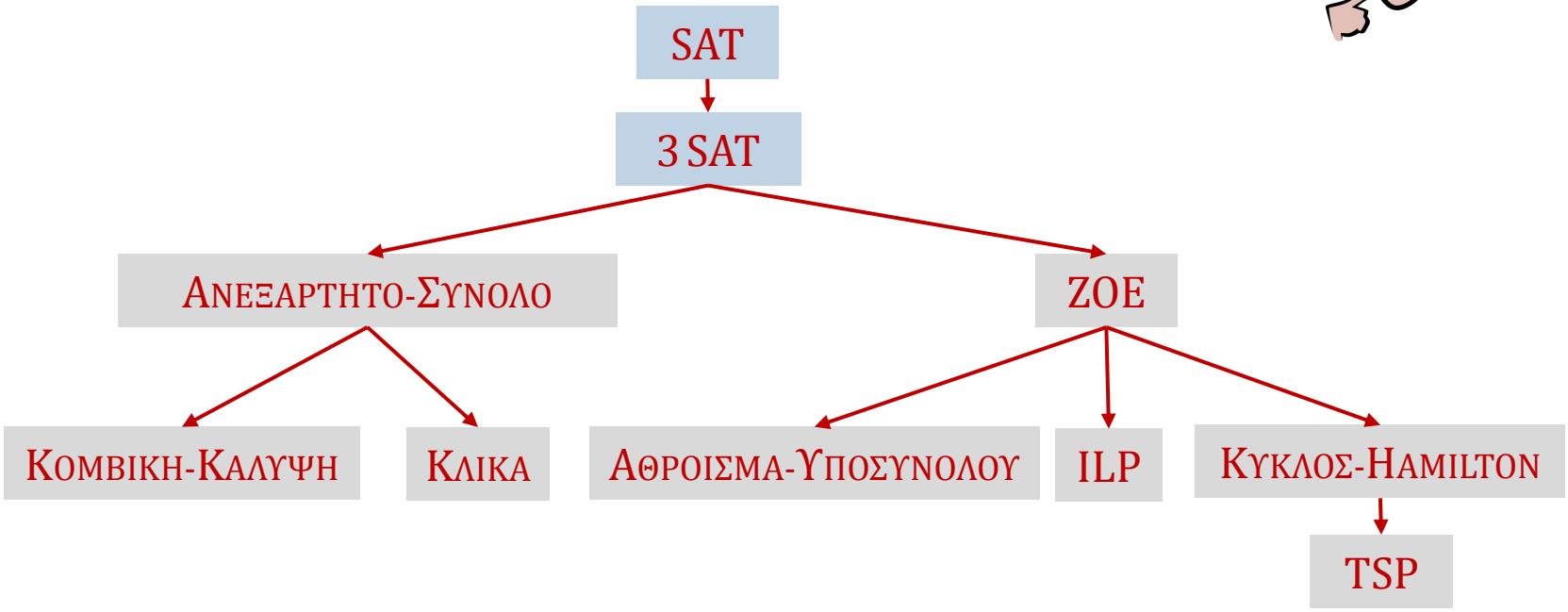
Θα δείξουμε μέσω Αναγωγών !!!

Αναγωγές Προβλημάτων NP

- $\mathbf{A \leq B \Leftrightarrow A \rightarrow B}$



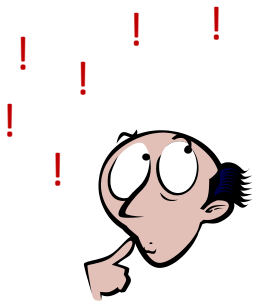
Οποιοδηποτε-Προβλημα-στο-**NP**



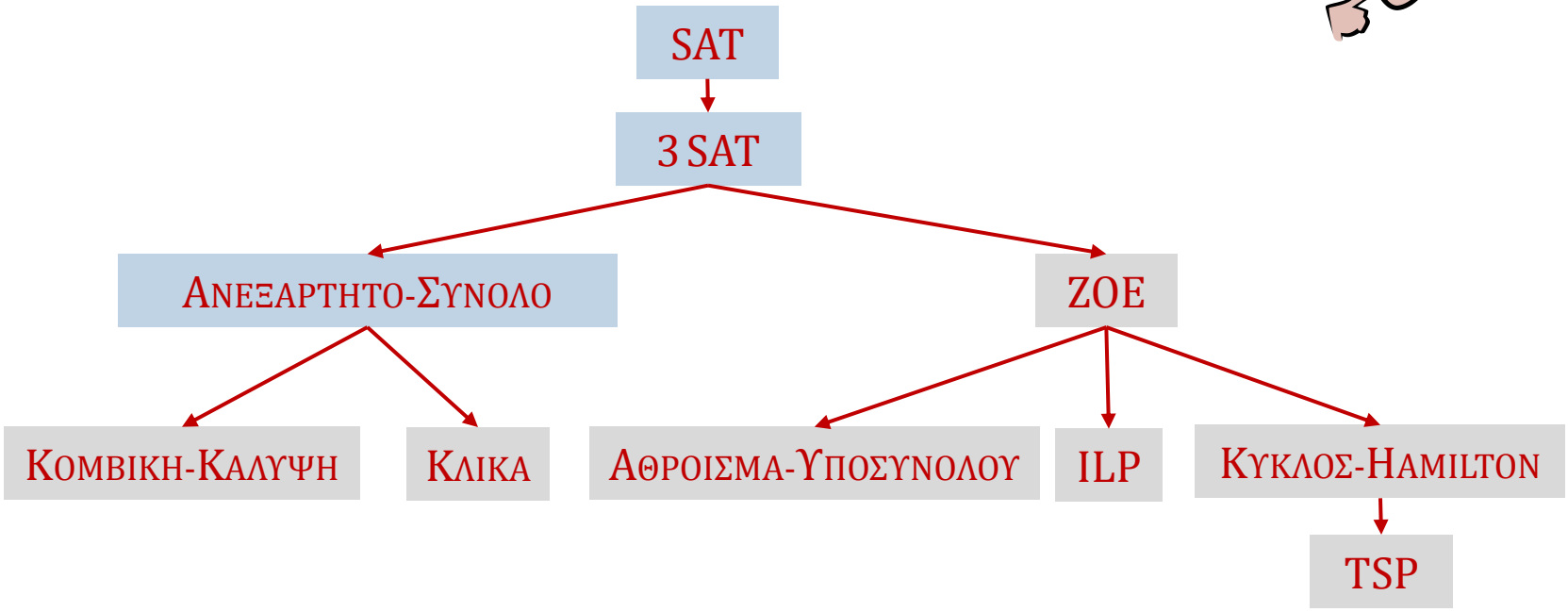
Θα δείξουμε μέσω Αναγωγών !!!

Αναγωγές Προβλημάτων NP

- $\mathbf{A \leq B \Leftrightarrow A \rightarrow B}$



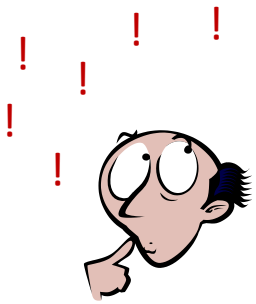
Οποιοδηποτε-Προβλημα-στο-**NP**



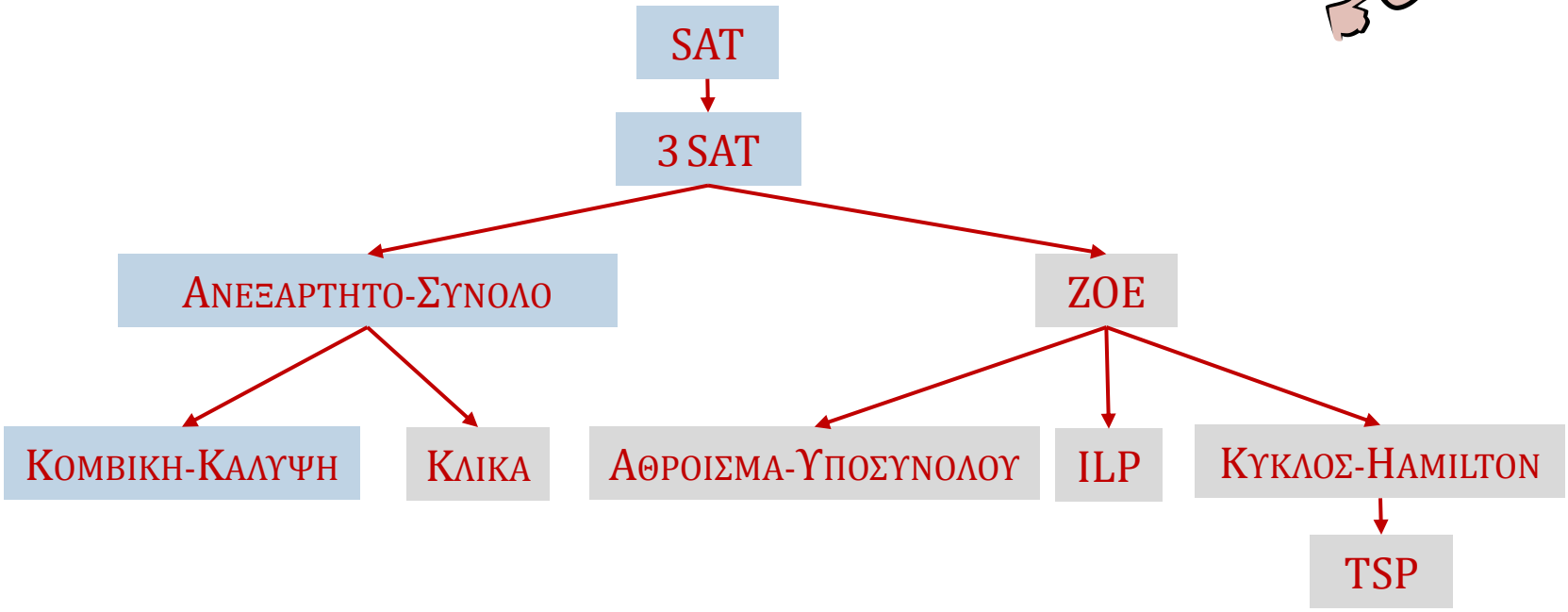
Θα δείξουμε μέσω Αναγωγών !!!

Αναγωγές Προβλημάτων NP

- $\mathbf{A \leq B \Leftrightarrow A \rightarrow B}$



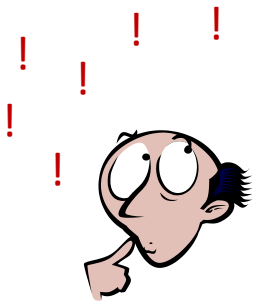
Οποιοδηποτε-Προβλημα-στο-**NP**



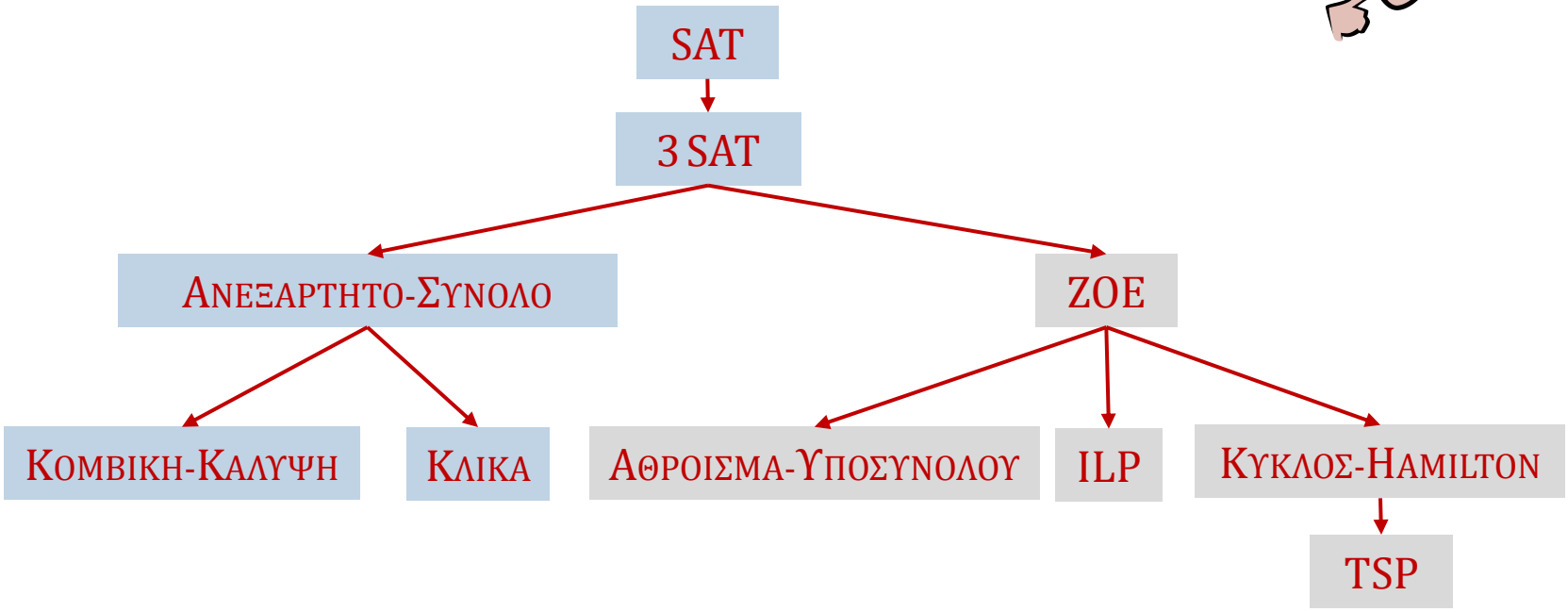
Θα δείξουμε μέσω Αναγωγών !!!

Αναγωγές Προβλημάτων NP

- $\mathbf{A \leq B \Leftrightarrow A \rightarrow B}$



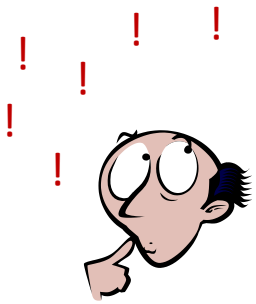
Οποιοδηποτε-Προβλημα-στο-**NP**



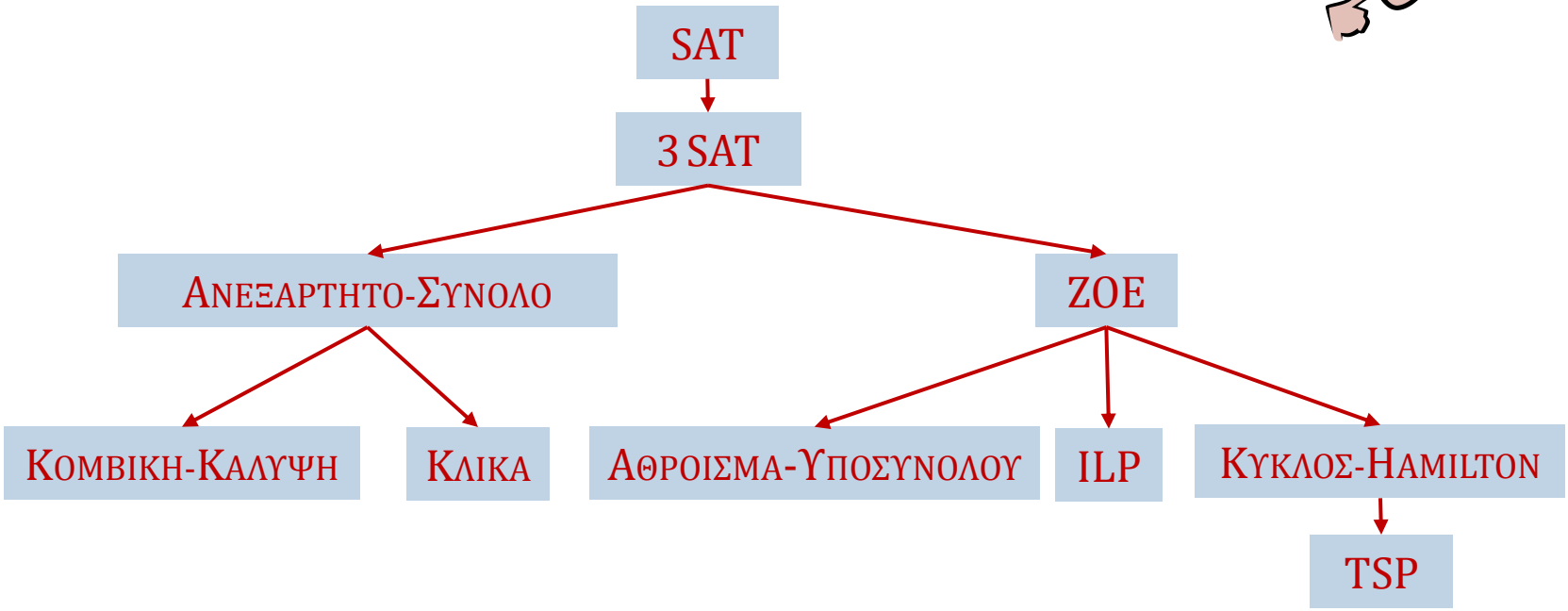
Θα δείξουμε μέσω Αναγωγών !!!

Αναγωγές Προβλημάτων NP

- $\mathbf{A \leq B \Leftrightarrow A \rightarrow B}$



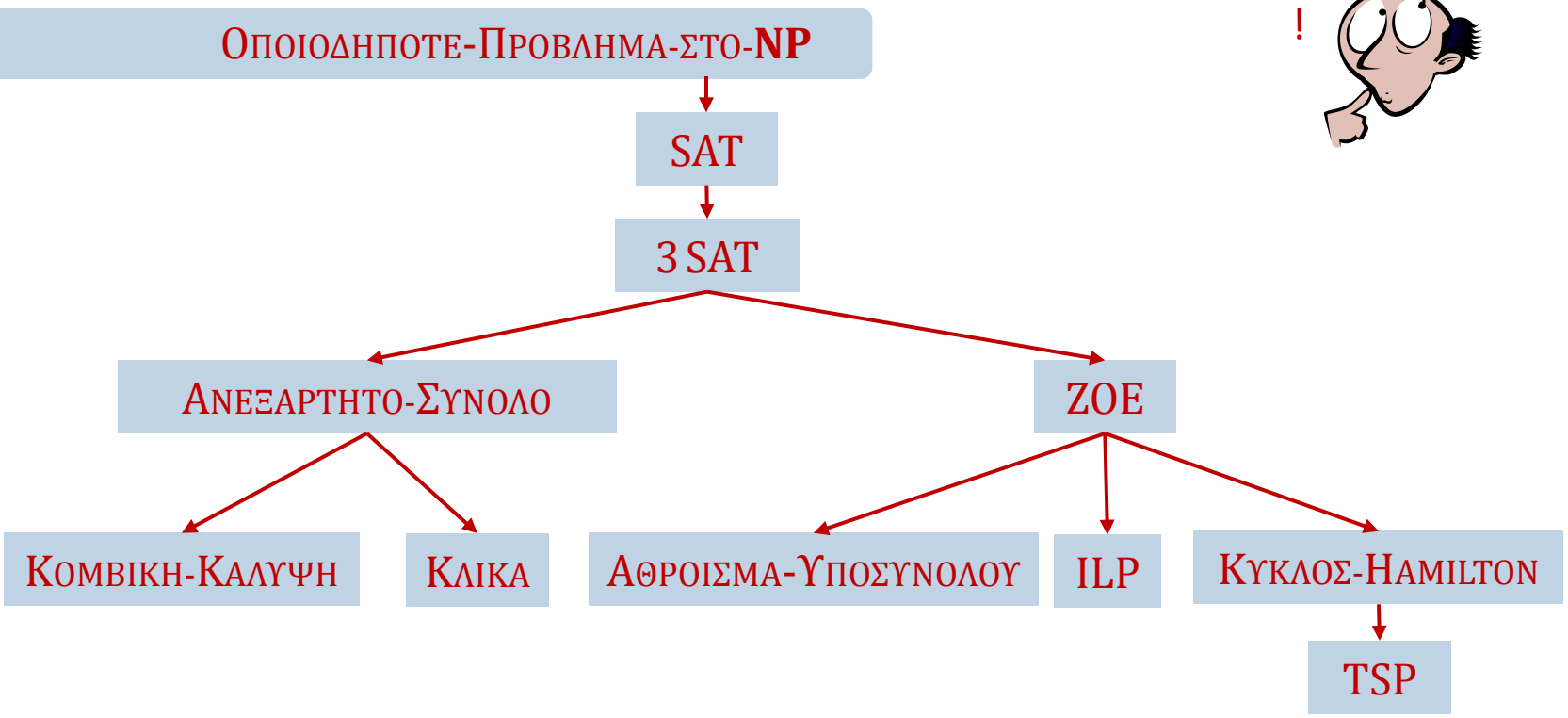
Οποιοδηποτε-Προβλημα-στο-**NP**



Θα δείξουμε μέσω Αναγωγών !!!

Αναγωγές Προβλημάτων NP

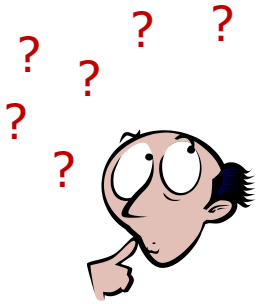
- $\mathbf{A \leq B \Leftrightarrow A \rightarrow B}$

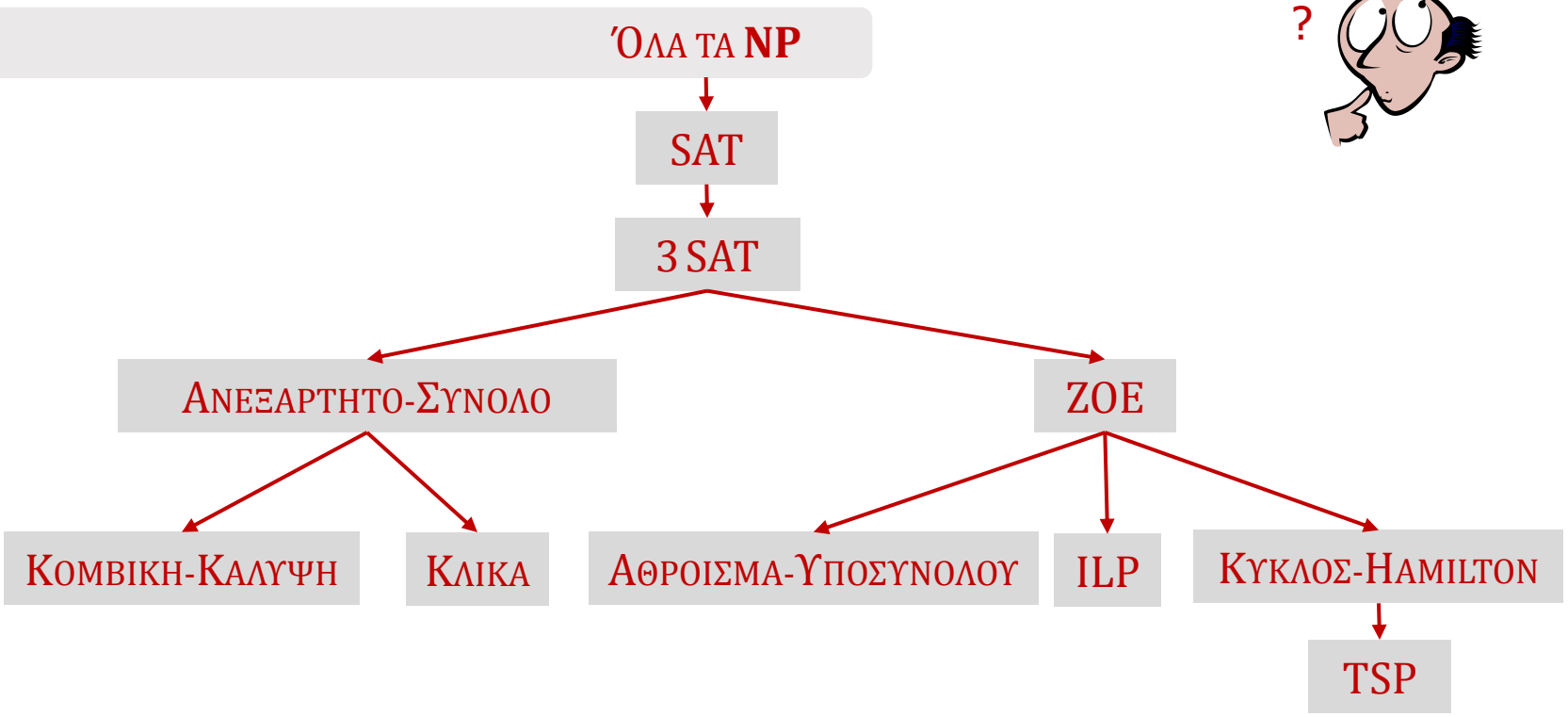


Θα δείξουμε μέσω Αναγωγών !!!

Αναγωγές Προβλημάτων NP

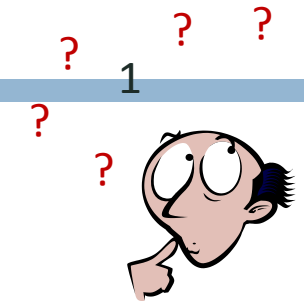
- $\mathbf{A \leq B \Leftrightarrow A \rightarrow B}$





ΟΠΟΙΟΔΗΠΟΤΕ-ΠΡΟΒΛΗΜΑ-ΣΤΟ-NP $\leq$ SAT $\leq \cdots \leq$ Q $\leq \cdots \leq$ R

Αναγωγές Προβλημάτων NP



1 SAT → 3SAT

- **Το πρόβλημα SAT**

  Μας δίδεται ένας λογικός τύπος (Boolean formula) σε συζευκτική μορφή:

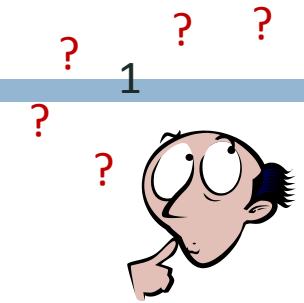
$$(x \vee y \vee z \vee \bar{w}) \wedge (\bar{x} \vee y) \wedge (z \vee \bar{x} \vee w \vee \bar{p} \vee \bar{y}) \wedge (x \vee \bar{y} \vee z)$$

  και μας ζητείται είτε να βρούμε μια *ικανοποιούσα ανάθεση τιμών αληθείας* (satisfying truth assignment) ή να αναφέρουμε ότι δεν υπάρχει καμία !!!

- **3SAT**

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

Αναγωγές Προβλημάτων NP



1 SAT → 3SAT

Τέχνασμα για την Αναγωγή του SAT στο 3SAT:

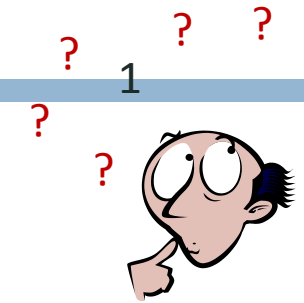
- Με δεδομένο ένα στιγμιότυπο **I** του SAT, χρησιμοποιούμε ακριβώς το ίδιο στιγμιότυπο για το 3SAT, με την διαφορά ότι ένας όρος με **k > 3** στοιχεία

$$(a_1 \vee a_2 \vee \ldots \vee a_k)$$

αντικαθίσταται από ένα σύνολο όρων

$$(a_1 \vee a_2 \vee y_1) \wedge (\bar{y}_1 \vee a_3 \vee y_2) \wedge (\bar{y}_2 \vee a_4 \vee y_3) \wedge \ldots \wedge (\bar{y}_{k-3} \vee a_{k-1} \vee a_k)$$

όπου τα $y_1, y_2, \ldots, y_{k-3}$ είναι νέες μεταβλητές.



1 SAT → 3SAT

Τέχνασμα για την Αναγωγή του SAT στο 3SAT:

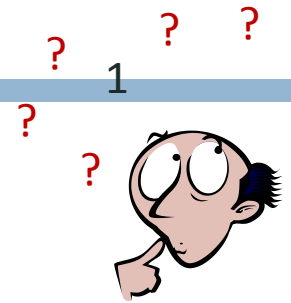
- Έστω **I′** το στιγ/πο του 3SAT που προκύπτει... προφανώς σε χρόνο T(|I|) !!!
- Στιγμιότυπο **I′** είναι ισοδύναμο με **I** ως προς την **ικανοποιησιμότητα** !!!

διότι για οποιαδήποτε ανάθεση τιμών στα $\mathbf{a_i}$ ($1 \le i \le k$), ισχύει:

$$\left\{ \underbrace{(a_1 \vee a_2 \vee \ldots \vee a_k)}_{\text{ικανοποιείται}} \right\} \Leftrightarrow$$

υπάρχει μια ανάθεση τιμών των $y_1, y_2, \ldots, y_{k-3}$ για την οποία οι όροι

$$\left\{ \underbrace{(a_1 \vee a_2 \vee y_1) \wedge (\bar{y}_1 \vee a_3 \vee y_2) \wedge \ldots \wedge (\bar{y}_{k-3} \vee a_{k-1} \vee a_k)}_{\text{ικανοποιούνται όλοι}} \right\}$$



1 SAT → 3SAT

Τέχνασμα για την Αναγωγή του SAT στο 3SAT:

- Για να το δείτε… υποθέστε ότι **ικανοποιούνται όλοι οι όροι στα δεξιά**.

Τότε, τουλάχιστον ένα από τα $\mathbf{a_1, a_2, \ldots, a_k}$ πρέπει να είναι **true** !!!

$$\left\{ \underbrace{(a_1 \vee a_2 \vee \ldots \vee a_k)}_{\text{ικανοποιείται}} \right\} \Leftrightarrow$$

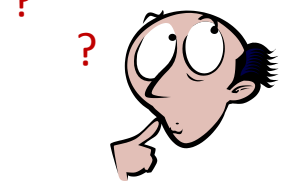
διαφορετικά το $\mathbf{y_1}$ θα πρέπει να είναι true, το οποίο αναγκάζει το $\mathbf{y_2}$ να είναι true, … κάνοντας false τον τελευταίο όρο !!!

Άρα, ικανοποιείται επίσης και ο όρος $(a_1 \vee a_2 \vee \ldots \vee a_k)$

υπάρχει μια ανάθεση τιμών των $y_1, y_2, \ldots, y_{k-3}$ για την οποία οι όροι

$$\left\{ \underbrace{(a_1 \vee a_2 \vee y_1) \wedge (\bar{y}_1 \vee a_3 \vee y_2) \wedge \ldots \wedge (\bar{y}_{k-3} \vee a_{k-1} \vee a_k)}_{\text{έστω ότι ικανοποιούνται όλοι}} \right\}$$

Αναγωγές Προβλημάτων NP



1 SAT → 3SAT

Τέχνασμα για την Αναγωγή του SAT στο 3SAT:

- Αντιστρόφως… υποθέστε ότι **ικανοποιείται ο όρος** $(a_1 \vee a_2 \vee \ldots \vee a_k)$

Τότε, κάποια a_i πρέπει να είναι **true** !!!

Θέτουμε τιμές **true** στις $y_1, y_2, \ldots, y_{i-2}$ και στις $y_{i-1}, y_i, \ldots, y_{k-3}$ τιμές **false** !!!

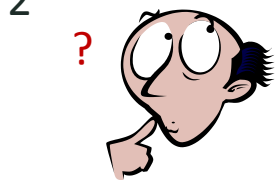
$$\left\{ \underbrace{(a_1 \vee a_2 \vee \ldots \vee a_k)}_{\text{ικανοποιείται}} \right\} \Leftrightarrow$$

Αυτό διασφαλίζει ότι **ικανοποιούνται** όλοι οι όροι δεξιά !!!

υπάρχει μια ανάθεση τιμών των $y_1, y_2, \ldots, y_{k-3}$ για την οποία οι όροι

$$\left\{ \underbrace{(a_1 \vee a_2 \vee y_1) \wedge (\bar{y}_1 \vee a_3 \vee y_2) \wedge \ldots \wedge (\bar{y}_{k-3} \vee a_{k-1} \vee a_k)}_{\text{ικανοποιούνται όλοι}} \right\}$$

Αναγωγές Προβλημάτων NP



2 3SAT → ΑΝΕΞΑΡΤΗΤΟ-ΣΥΝΟΛΟ

- **3SAT**

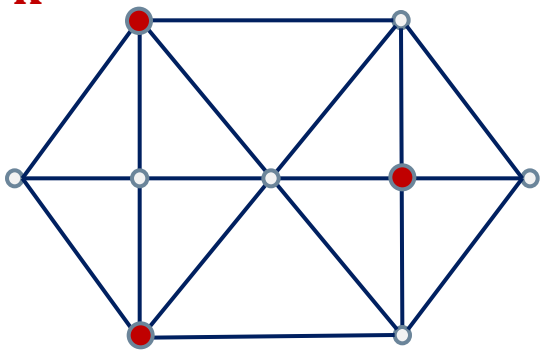
$$(\bar{x} \vee y \vee \bar{z}) \wedge (x \vee \bar{y} \vee z) \wedge (x \vee y \vee z) \wedge (\bar{x} \vee \bar{y})$$

- **ΑΝΕΞΑΡΤΗΤΟ-ΣΥΝΟΛΟ**

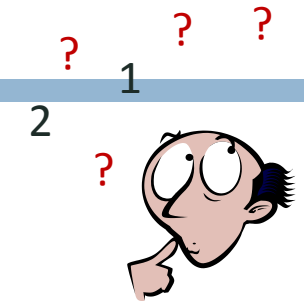
Μας δίδεται γράφημα **n** κόμβων και *ακέραιος* **k**

και μας ζητείται να βρούμε ένα σύνολο **IS** με **k** κόμβους οι οποίοι είναι ανεξάρτητοι, δηλ. ανά δύο δεν έχουν ακμή μεταξύ τους,

ή να αναφέρουμε ότι δεν υπάρχει τέτοιο σύνολο κόμβων !!!



Αναγωγές Προβλημάτων NP

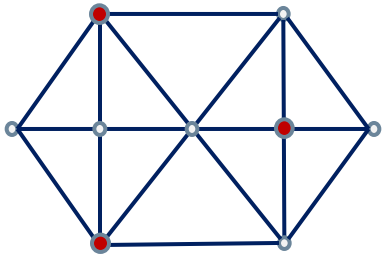


2 3SAT → ΑΝΕΞΑΡΤΗΤΟ-ΣΥΝΟΛΟ

- **Δύο πολύ διαφορετικά προβλήματα !!!**

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

- Συσχετισμός **Λογικής Boole** με **Γραφήματα** ?

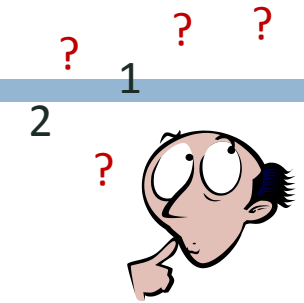


- **Ας σκεφτούμε…**

  Σε μια *ικανοποιούσα ανάθεση τιμών αληθείας* του 3SAT θα πρέπει **ένα στοιχείο** από **κάθε όρο** να είναι **true** !!!

- **Συνεπείς επιλογές…** εάν **επιλέξουμε** το **x** να είναι **true** σε **κάποιο όρο** δεν μπορούμε να **επιλέξουμε** το $\bar{x}$ να είναι **true** σε **κάποιο άλλο** !!!

Αναγωγές Προβλημάτων NP

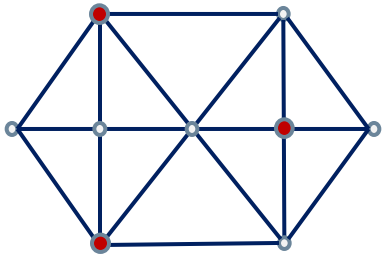


2 3SAT → ΑΝΕΞΑΡΤΗΤΟ-ΣΥΝΟΛΟ

- **Δύο πολύ διαφορετικά προβλήματα !!!**

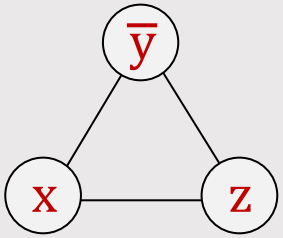
$$(\bar{x} \vee y \vee \bar{z}) \wedge (x \vee \bar{y} \vee z) \wedge (x \vee y \vee z) \wedge (\bar{x} \vee \bar{y})$$

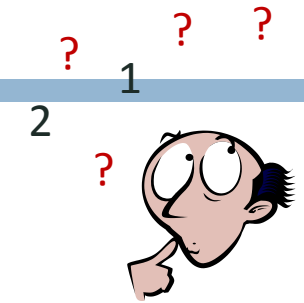
- Συσχετισμός **Λογικής Boole** με **Γραφήματα** ?



ΑΝΑΠΑΡΑΣΤΑΣΗ

$$(x \vee \bar{y} \vee z) \rightarrow$$

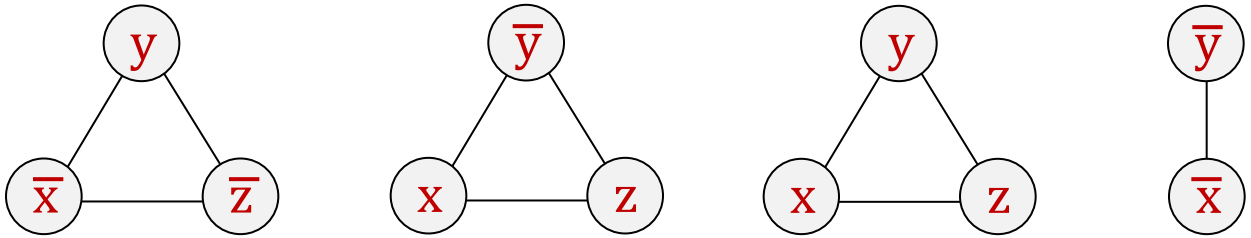




2 3SAT → ΑΝΕΞΑΡΤΗΤΟ-ΣΥΝΟΛΟ

- **Παράδειγμα**

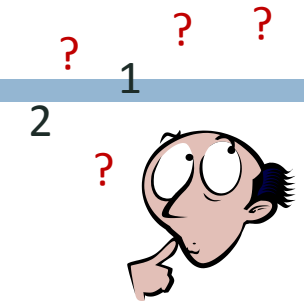
$$(\bar{x} \vee y \vee \bar{z}) \wedge (x \vee \bar{y} \vee z) \wedge (x \vee y \vee z) \wedge (\bar{x} \vee \bar{y})$$



- Στο **γράφημα G** που προκύπτει, ένα **IS** περιέχει *το πολύ ένα στοιχείο από κάθε **τρίγωνο–όρο*** !!!

- Για να επιβάλουμε **ΜΙΑ επιλογή από κάθε τρίγωνο-όρο**, θέτουμε:

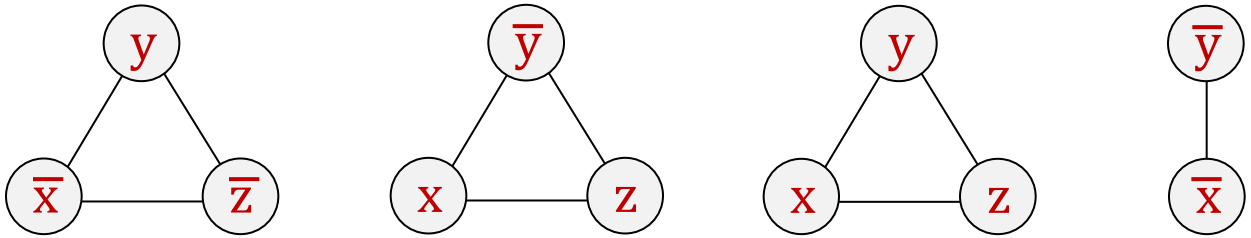
k = πλήθος όρων



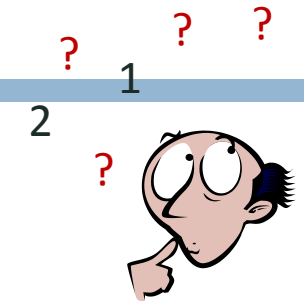
2 3SAT → ΑΝΕΞΑΡΤΗΤΟ-ΣΥΝΟΛΟ

- **Παράδειγμα**

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



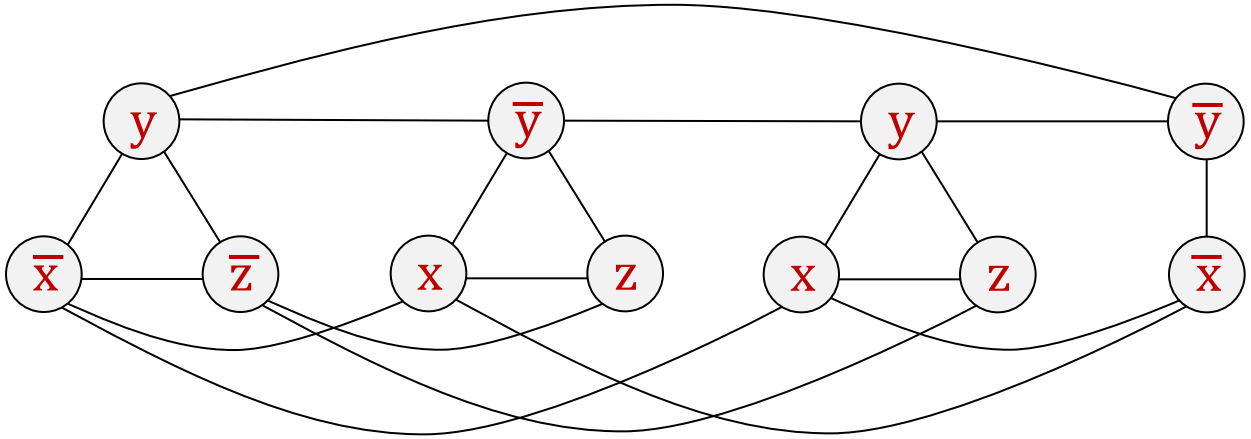
- **Λείπει κάτι ?...** **ΝΑΙ**, ένας τρόπος που θα μας **εμποδίζει να επιλέγουμε *αντίθετα στοιχεία*** - και το **x** και το $\bar{x}$!!!
- **Αυτό είναι πολύ εύκολο !!!**



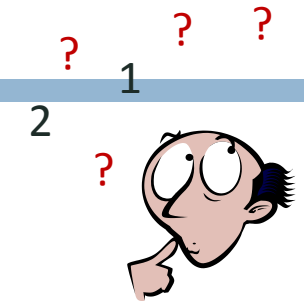
2 3SAT → Ανεξαρτητο-Συνολο

- **Παράδειγμα**

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



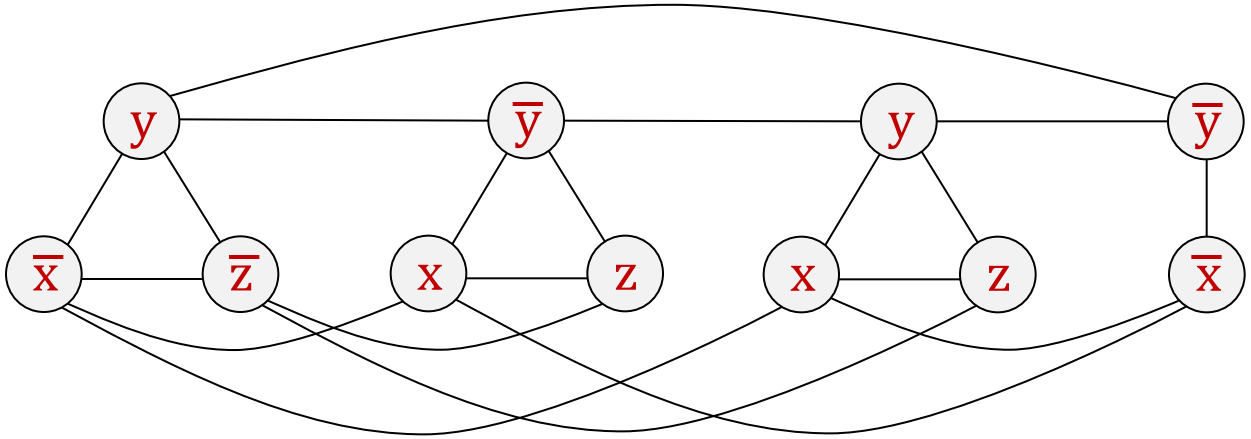
- **Αυτό είναι πολύ εύκολο !!!... Ορίστε και ο Τρόπος !!!**



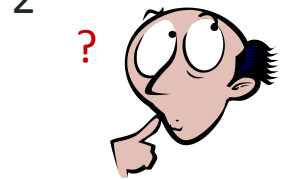
2 3SAT → ΑΝΕΞΑΡΤΗΤΟ-ΣΥΝΟΛΟ

- **Παράδειγμα**

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



- Με δεδομένο ένα στιγμιότυπο **I** του προβλήματος **3SAT** δημιουργήσαμε ένα στιγ/πο **(G, k)** του προβλήματος **ΑΝΕΞΑΡΤΗΤΟ-ΣΥΝΟΛΟ**

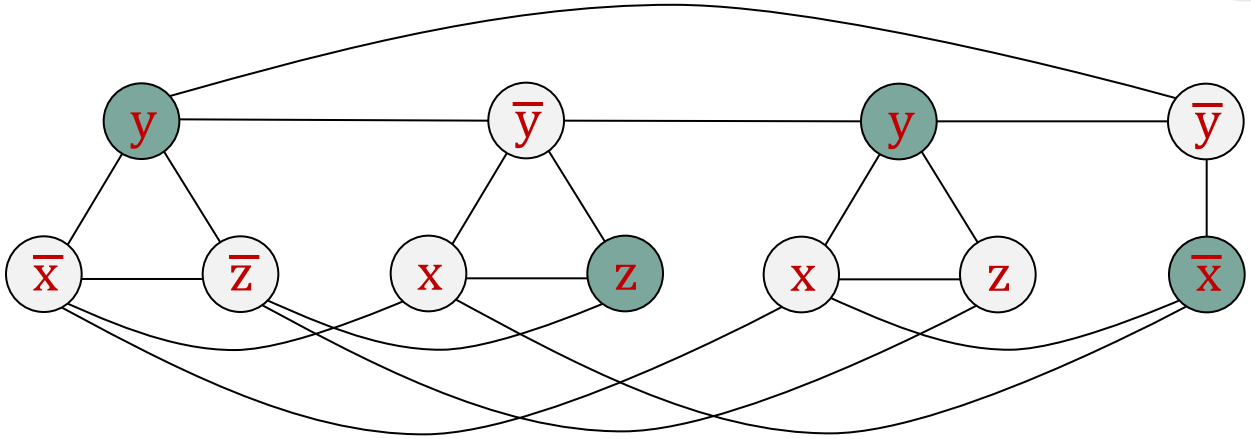


2 3SAT → ΑΝΕΞΑΡΤΗΤΟ-ΣΥΝΟΛΟ

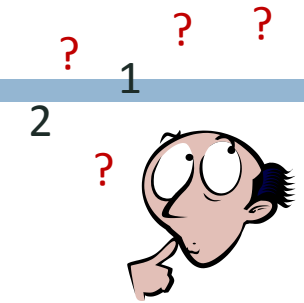
- **Παράδειγμα**

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

x = false
y = true
z = true



- Με δεδομένο ένα στιγμιότυπο **I** του προβλήματος **3SAT** δημιουργήσαμε ένα στιγ/πο **(G, k)** του προβλήματος **ΑΝΕΞΑΡΤΗΤΟ-ΣΥΝΟΛΟ**

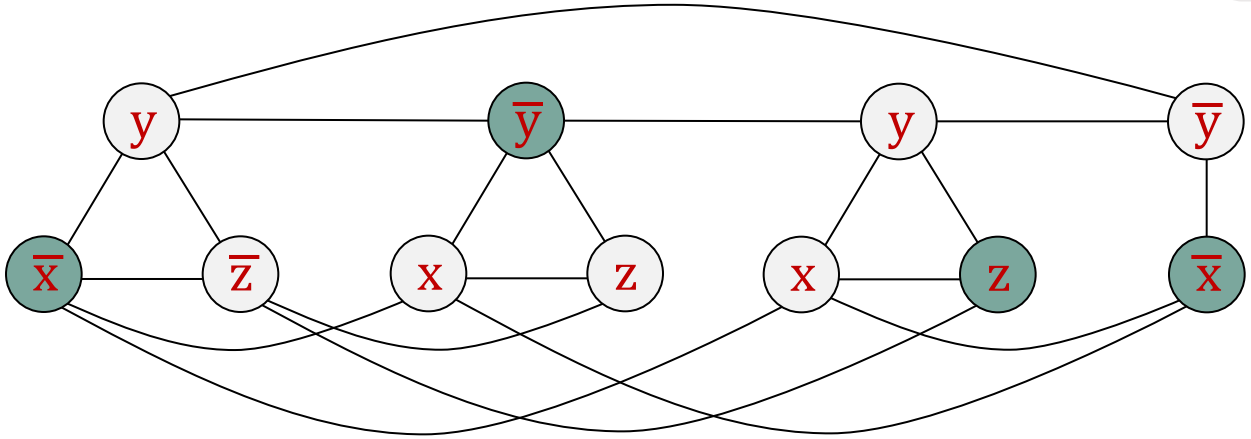


2 3SAT → ΑΝΕΞΑΡΤΗΤΟ-ΣΥΝΟΛΟ

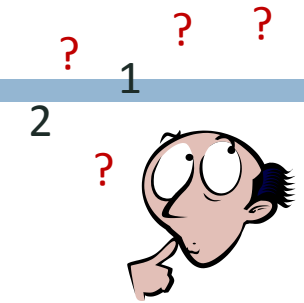
- **Παράδειγμα**

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

x = false
y = false
z = true



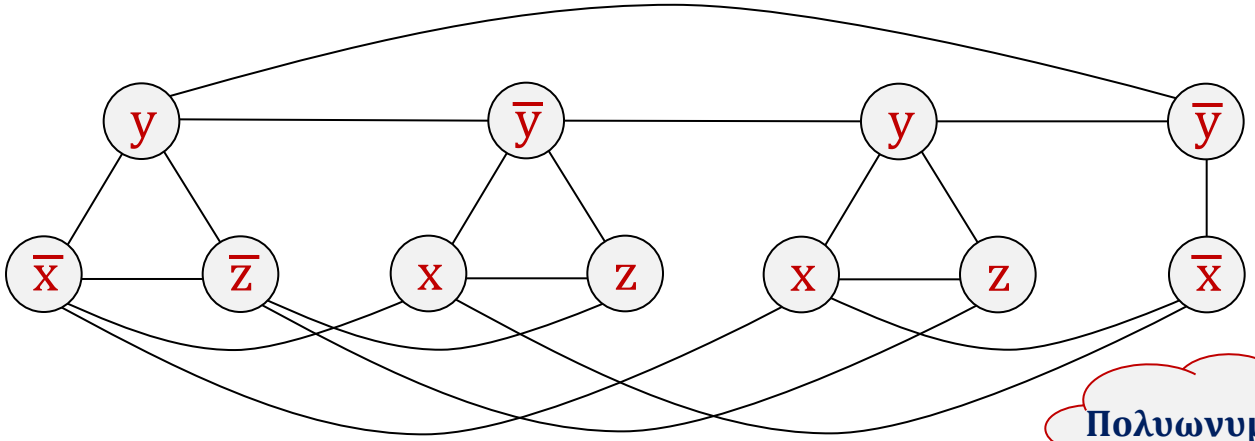
- Με δεδομένο ένα στιγμιότυπο **I** του προβλήματος **3SAT** δημιουργήσαμε ένα στιγ/πο **(G, k)** του προβλήματος **ΑΝΕΞΑΡΤΗΤΟ-ΣΥΝΟΛΟ**



2 3SAT → ΑΝΕΞΑΡΤΗΤΟ-ΣΥΝΟΛΟ

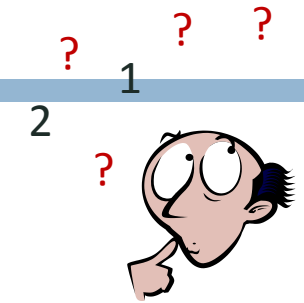
- **Παράδειγμα**

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



Πολυωνυμικό Χρόνο

- Με δεδομένο ένα στιγμιότυπο **I** του προβλήματος **3SAT** δημιουργήσαμε ένα στιγ/πο **(G, k)** του προβλήματος **ΑΝΕΞΑΡΤΗΤΟ-ΣΥΝΟΛΟ**



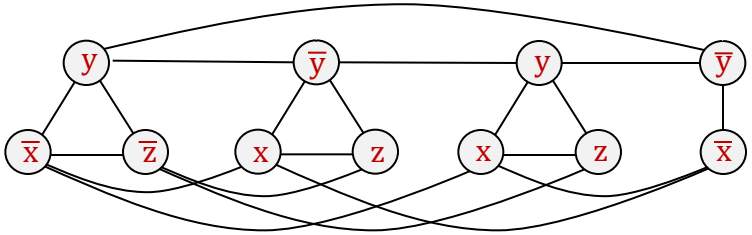
2 3SAT → Ανεξαρτητο-Συνολο

- **Θυμηθείτε !!!**

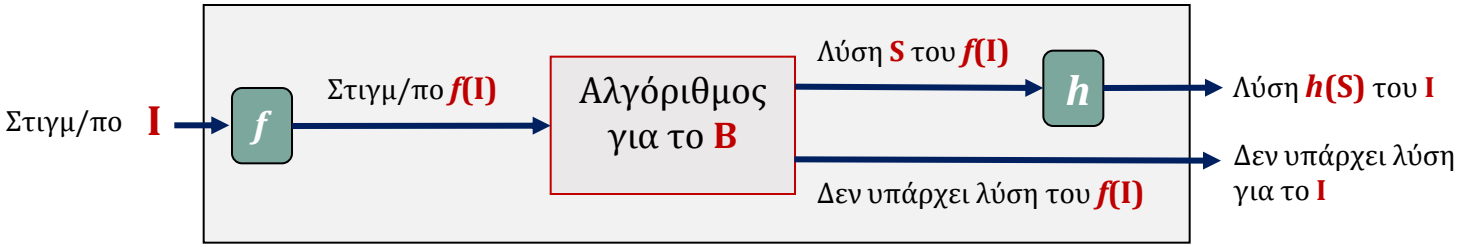
Στις αναγωγές **δεν χρειαζόμαστε μόνο** τρόπο απεικόνισης στιγμιότυπων **I** του **A** σε στιγμιότυπα ***f*(I)** του **B** !!!

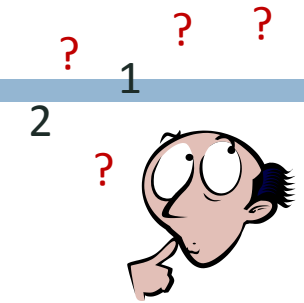
Αλλά και έναν τρόπο **ανακατασκευής** μιας λύσης ***h*(I)** του **I** από οποιαδήποτε λύση **S** του ***f*(I)** -- δηλαδή τη συνάρτηση ***h***

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



Αλγόριθμος για το A



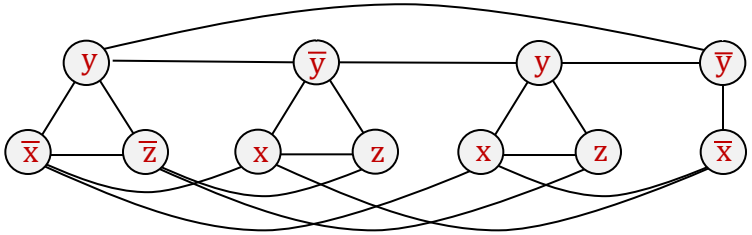


2 3SAT → ΑΝΕΞΑΡΤΗΤΟ-ΣΥΝΟΛΟ

- **Όπως πάντα !!!**

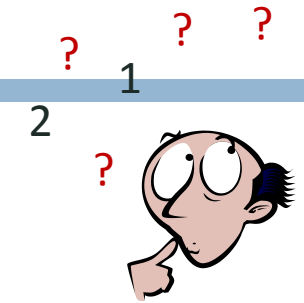
  Υπάρχουν δύο πράγματα για να **αποδείξουμε** !!!

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



(1) Με δεδομένο ένα ***ανεξάρτητο σύνολο* S** με **k** κόμβους στο γράφημα **G**, μπορούμε πάντα να ανακτήσουμε αποδοτικά μια ***ικανοποιούσα ανάθεση τιμών αληθείας*** στο **I.**

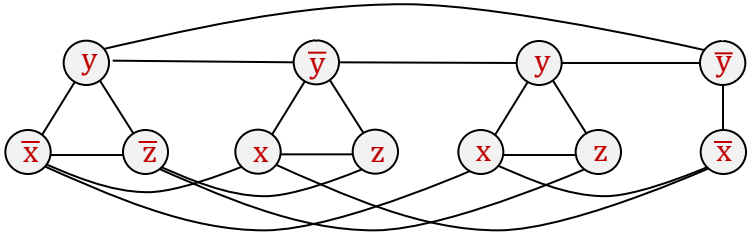
(2) Αν το γράφημα **G** δεν έχει ***ανεξάρτητο σύνολο* S** μεγέθους **k,** τότε ο λογικός (Boolean) τύπος **I *δεν είναι ικανοποιήσιμος*.**



2 3SAT → ΑΝΕΞΑΡΤΗΤΟ-ΣΥΝΟΛΟ

- **Απόδειξη (1)**

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



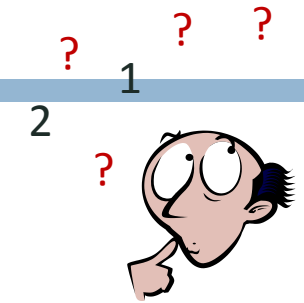
Για οποιαδήποτε μεταβλητή **x**, το **S δεν περιέχει κόμβους** με επιγραφές **x** και $\bar{x}$!!!

Αναθέτουμε: **x = true** εάν το **S** περιέχει κόμβο με επιγραφή **x**
x = false εάν το **S** περιέχει κόμβο με επιγραφή $\bar{x}$

Επειδή, **|S| = k** πρέπει να έχει **ένα κόμβο ανά όρο** !!!

Αυτή η ανάθεση ικανοποιεί όλους τους όρους !!!

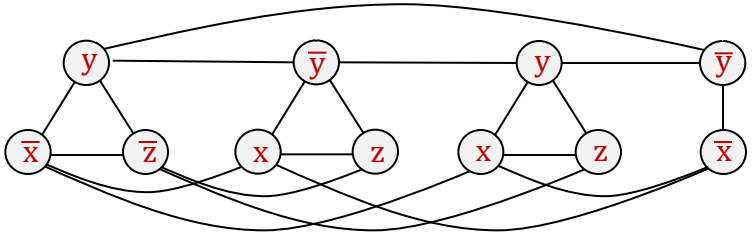
(1) Με δεδομένο ένα ***ανεξάρτητο σύνολο*** **S** με **k** κόμβους στο **G**, μπορούμε πάντα να ανακτήσουμε αποδοτικά μια ***ικανοποιούσα ανάθεση τιμών αληθείας*** στο **I.**



2 3SAT → ΑΝΕΞΑΡΤΗΤΟ-ΣΥΝΟΛΟ

- **Απόδειξη (2)**

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

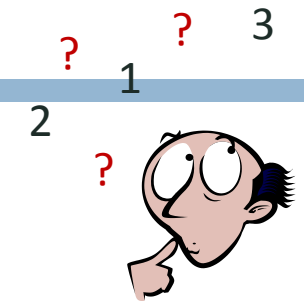


Αρκεί να αποδείξουμε ότι:
εάν ο λ.τ. **I** έχει μια *ικανοποιούσα ανάθεση τιμών αληθείας*,
τότε το **G** έχει ένα **IS** μεγέθους **k** !!!

Αυτό είναι εύκολο !!!... Για **κάθε όρο** του **3SAT**, ***επιλέγουμε ένα στοιχείο του οποίου η τιμή στην ικανοποιούσα ανάθεση είναι true*** και ***προσθέτουμε τον αντίστοιχο κόμβο στο S***... τότε το S είναι ανεξάρτητο... το βλέπετε ?

(2) Αν το γράφημα G δεν έχει ***ανεξάρτητο σύνολο S*** μεγέθους **k,** τότε ο λογικός (Boolean) τύπος **I** ***δεν είναι ικανοποιήσιμος***.

Αναγωγές Προβλημάτων NP

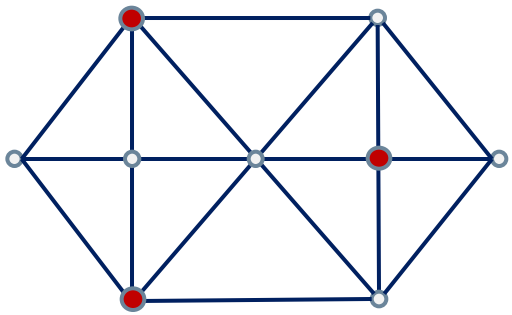


3 ΑΝΕΞΑΡΤΗΤΟ-ΣΥΝΟΛΟ → ΚΟΜΒΙΚΗ-ΚΑΛΥΨΗ

- **ΑΝΕΞΑΡΤΗΤΟ-ΣΥΝΟΛΟ**

  Μας δίδεται γράφημα **n** κόμβων και *ακέραιος* **k** και μας ζητείται να βρούμε ένα σύνολο **IS** με **k** κόμβους οι οποίοι είναι ανεξάρτητοι

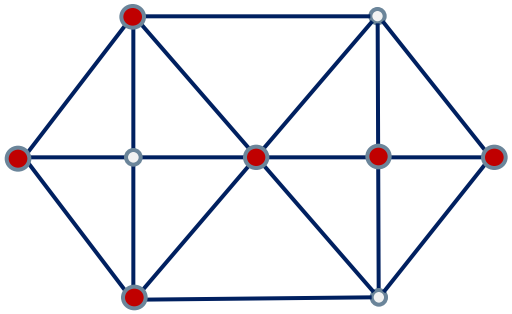
  ή να αναφέρουμε ότι δεν υπάρχει τέτοιο σύνολο κόμβων !!!



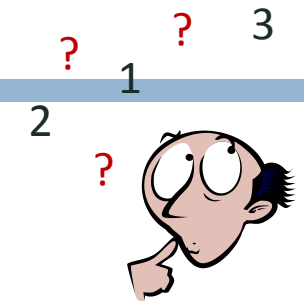
- **ΚΟΜΒΙΚΗ-ΚΑΛΥΨΗ**

  Μας δίδεται γράφημα **n** κόμβων και *ακέραιος* **k** και μας ζητείται να βρούμε ένα σύνολο **VC** με **k** κόμβους οι οποίοι καλύπτουν κάθε ακμή

  ή να αναφέρουμε ότι δεν υπάρχει τέτοιο σύνολο κόμβων !!!



Αναγωγές Προβλημάτων NP

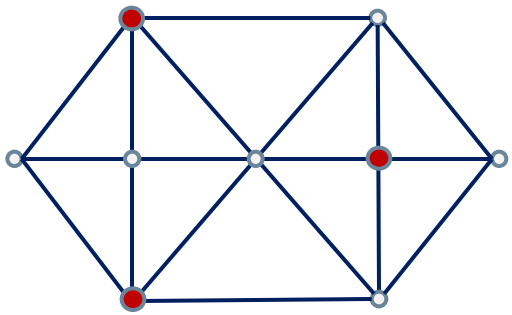


3 ΑΝΕΞΑΡΤΗΤΟ-ΣΥΝΟΛΟ → ΚΟΜΒΙΚΗ-ΚΑΛΥΨΗ

- **Ισχύει ότι…**

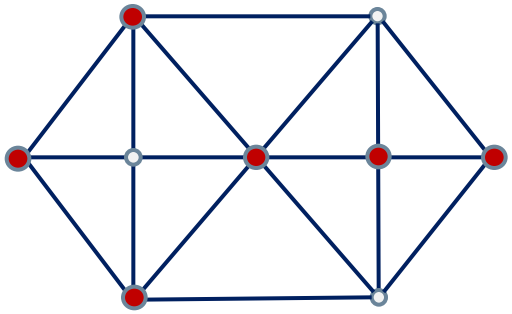
  Κάποιες αναγωγές βασίζονται στην **επινοητικότητα** του **συσχετισμού** ***δύο πολύ διαφορετικών προβλημάτων*** !!!

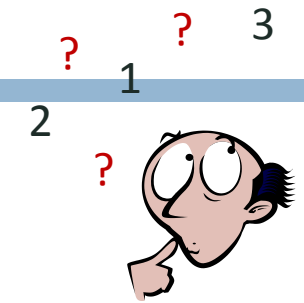
  Ενώ, άλλες απλώς **καταγράφουν** μια λεπτή ***μεταμφίεση*** του ***ενός προβλήματος*** στο ***άλλο*** !!!



- **Τι ισχύει εδώ ?**

  Τι μπορούμε να πούμε για τη **συσχέτιση** των δύο προβλημάτων **?**





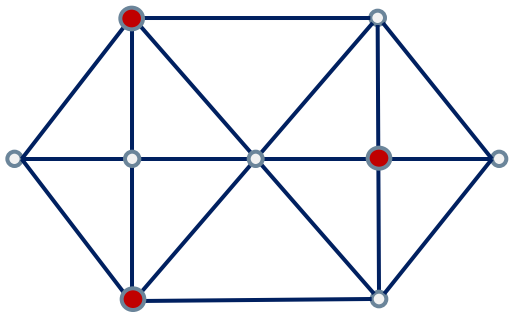
3 Ανεξαρτητο-Συνολο → Κομβικη-Καλυψη

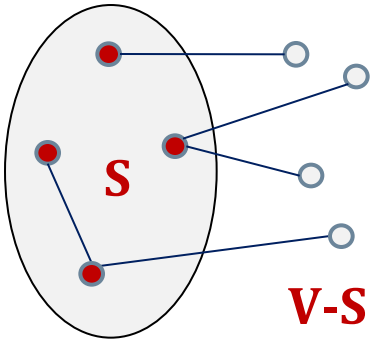
- **Παρατήρηση**

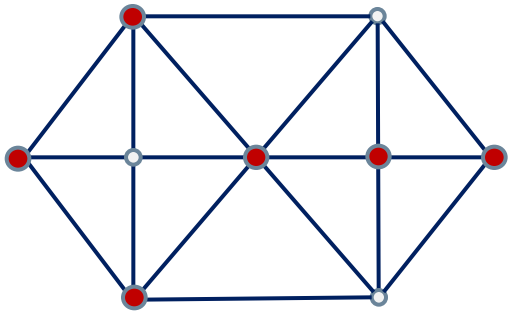
  Ένα σύνολο κόμβων **S** αποτελεί **κομβική κάλυψη** ενός γραφήματος **G = (V, E)**

  εάν και μόνο εάν

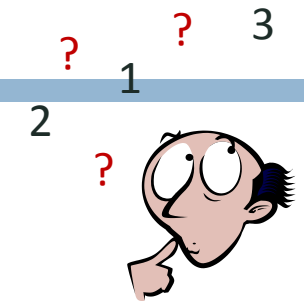
  οι υπόλοιποι κόμβοι στο σύνολο **V-S** είναι **ανεξάρτητο σύνολο** του **G !!!**







Αναγωγές Προβλημάτων NP

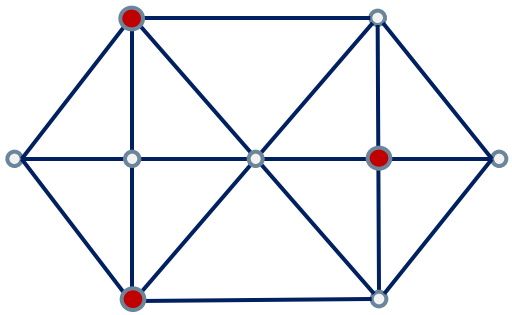


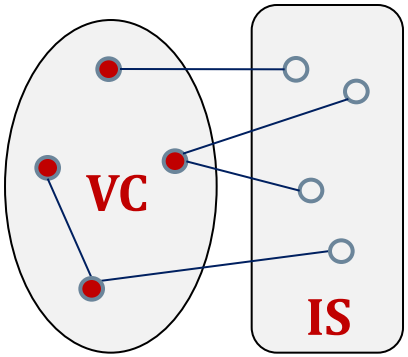
3 ΑΝΕΞΑΡΤΗΤΟ-ΣΥΝΟΛΟ → ΚΟΜΒΙΚΗ-ΚΑΛΥΨΗ

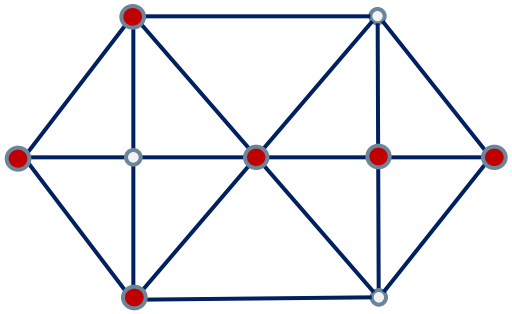
- **Αναγωγή**

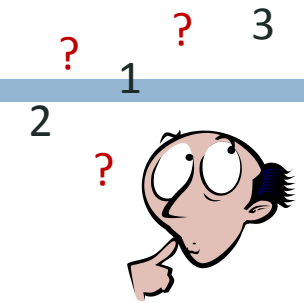
  Για να **λύσουμε** ένα στιγμιότυπο **(G, k)** του προβλήματος ΑΝΕΞΑΡΤΗΤΟ-ΣΥΝΟΛΟ

  απλώς **αναζητούμε** μια Κομβική Κάλυψη **VC** του **G** με **|V|-k** κόμβους !!!







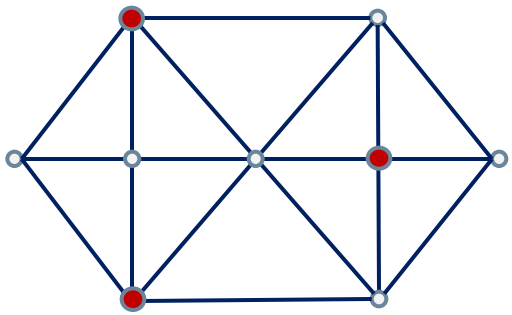


3 ΑΝΕΞΑΡΤΗΤΟ-ΣΥΝΟΛΟ → ΚΟΜΒΙΚΗ-ΚΑΛΥΨΗ

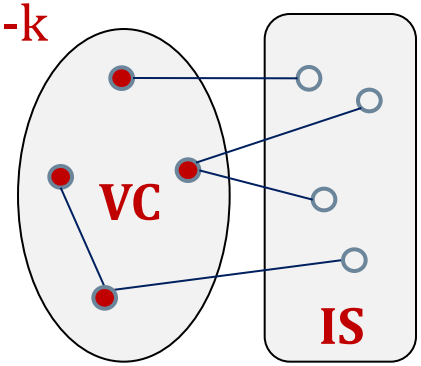
- **Αναγωγή**

  Εάν **υπάρχει** τέτοια **κομβική κάλυψη VC**

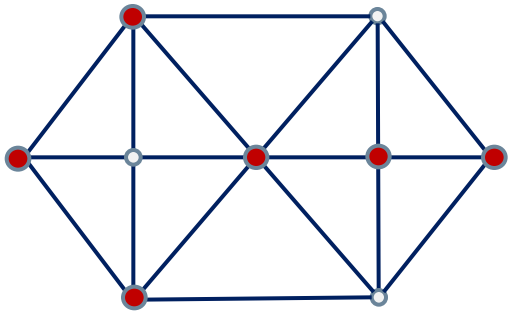
  τότε, *παίρνουμε όλους τους κόμβους* που **δεν ανήκουν** σε αυτή !!!

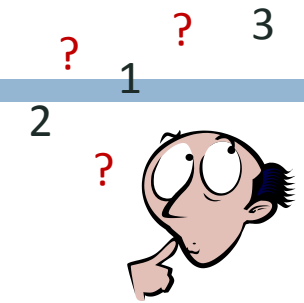


| VC | = |V|-k



| IS | = k

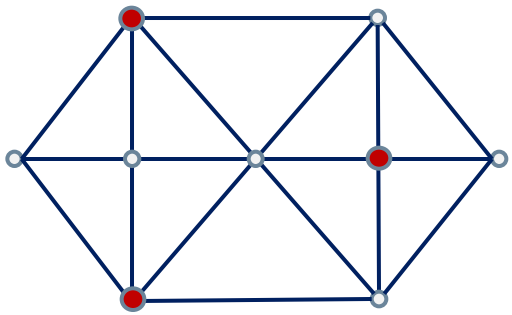




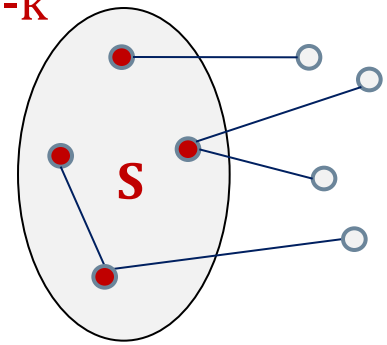
3 ΑΝΕΞΑΡΤΗΤΟ-ΣΥΝΟΛΟ → ΚΟΜΒΙΚΗ-ΚΑΛΥΨΗ

- **Αναγωγή**

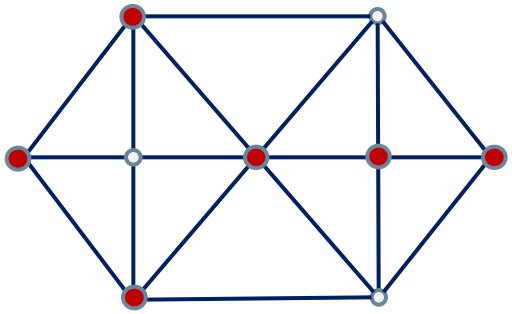
  Εάν **ΔΕΝ υπάρχει** τέτοια **κομβική κάλυψη VC τότε,** το **G** δεν έχει **ανεξάρτητο σύνολο** μεγέθους **k** !!!



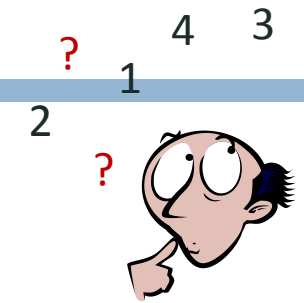
| VC | ≠ |V|-k



| IS | ≠ k



Αναγωγές Προβλημάτων NP

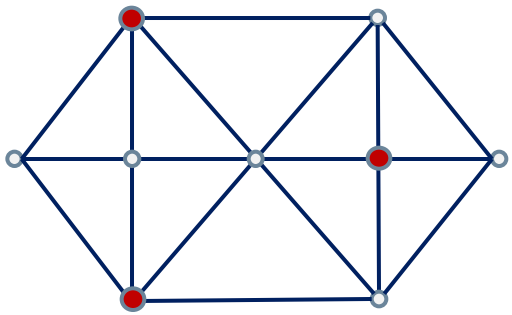


4 ΑΝΕΞΑΡΤΗΤΟ-ΣΥΝΟΛΟ → ΚΛΙΚΑ

- **ΑΝΕΞΑΡΤΗΤΟ-ΣΥΝΟΛΟ**

  Μας δίδεται γράφημα **n** κόμβων και *ακέραιος* **k** και μας ζητείται να βρούμε ένα σύνολο **IS** με **k** κόμβους οι οποίοι είναι ανεξάρτητοι,

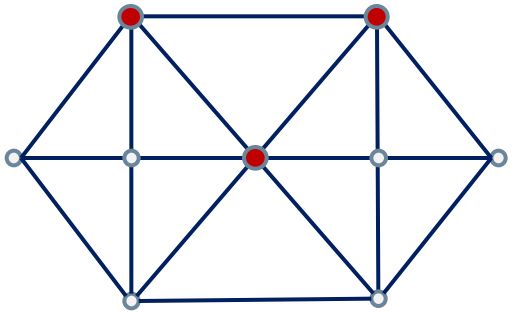
  ή να αναφέρουμε ότι δεν υπάρχει τέτοιο σύνολο κόμβων !!!



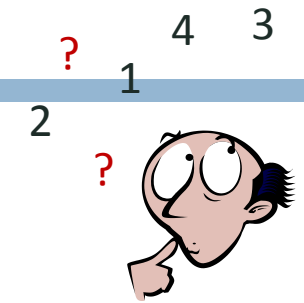
- **ΚΛΙΚΑ**

  Μας δίδεται γράφημα **n** κόμβων και *ακέραιος* **k** και μας ζητείται να βρούμε ένα σύνολο **CL** με **k** κόμβους οι οποίοι έχουν πλήρη σύνδεση

  ή να αναφέρουμε ότι δεν υπάρχει τέτοιο σύνολο κόμβων !!!



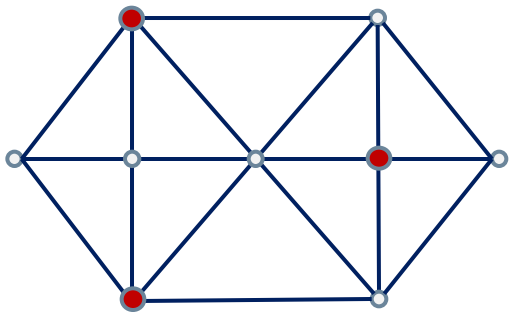
Αναγωγές Προβλημάτων NP



4 ΑΝΕΞΑΡΤΗΤΟ-ΣΥΝΟΛΟ → ΚΛΙΚΑ

- **Παρατήρηση**

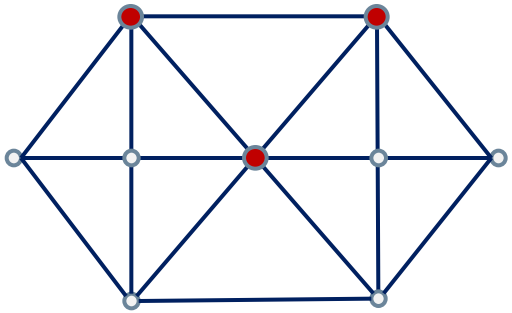
Πολύ εύκολη αναγωγή του ενός προβλήματος στο άλλο !!!



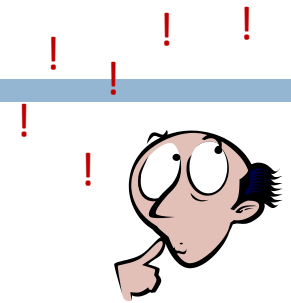
Ένα σύνολο κόμβων **S** αποτελεί **ανεξάρτητο σύνολο** ενός γραφήματος **G = (V, E)**

εάν και μόνο εάν

το **S** αποτελεί μια **κλίκα** του *συμπληρώματος (complement)* $\overline{\mathbf{G}} = (\mathbf{V}, \overline{\mathbf{E}})$ του γραφήματος **G !!!**

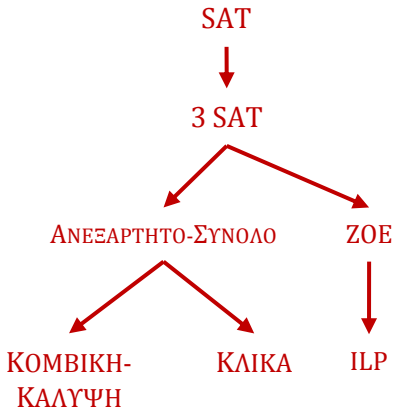


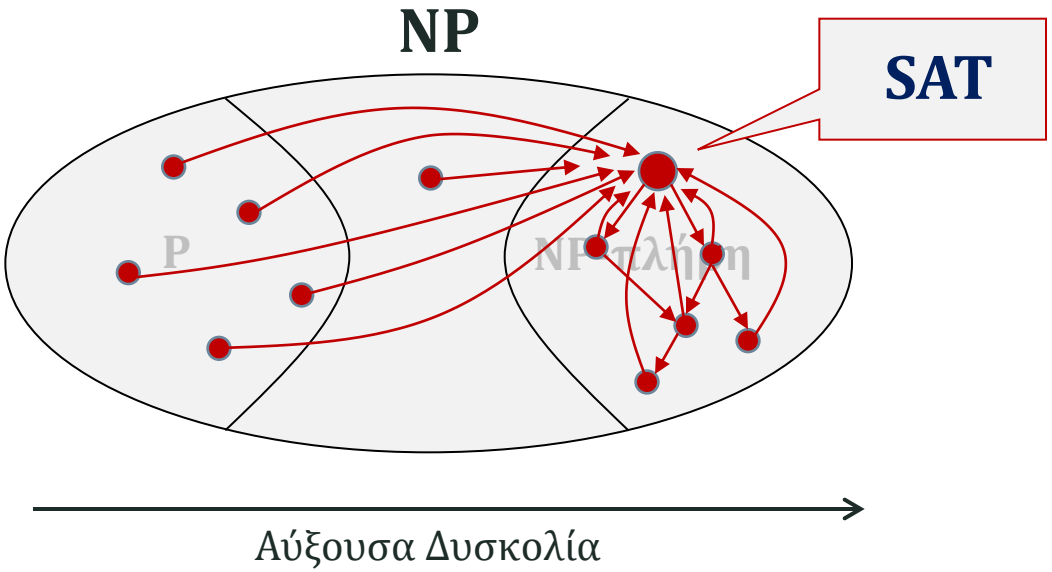
Αναγωγές Προβλημάτων NP



Οποιοδηποτε-Προβλημα-NP → SAT

- **Προβλήματα NP**

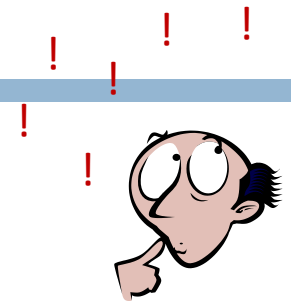




- **SAT**

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

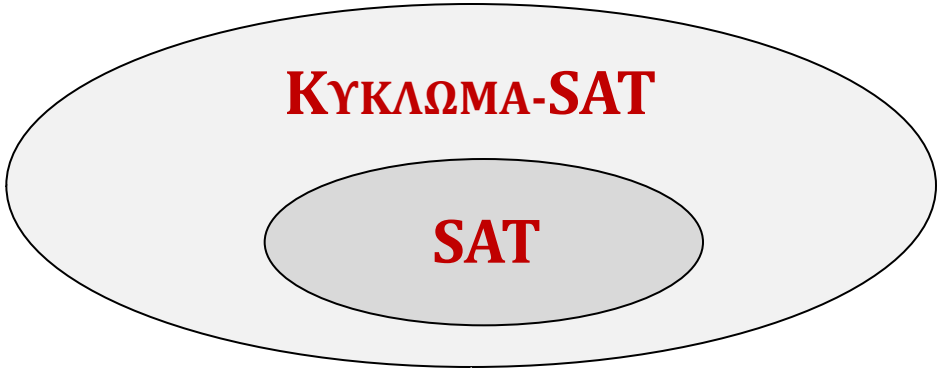
Αναγωγές Προβλημάτων NP



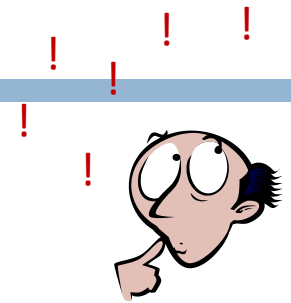
Οποιοδηποτε-Προβλημα-NP → SAT

- **Προβλημα C-SAT**

  Θα αποδείξουμε ότι *όλα τα προβλήματα στο NP μπορούν να αναχθούν* σε μία γενίκευση του SAT που ονομάζουμε **Κυκλωμα-SAT** ή **C-SAT** (Circuit-SAT) !!!



Αναγωγές Προβλημάτων NP



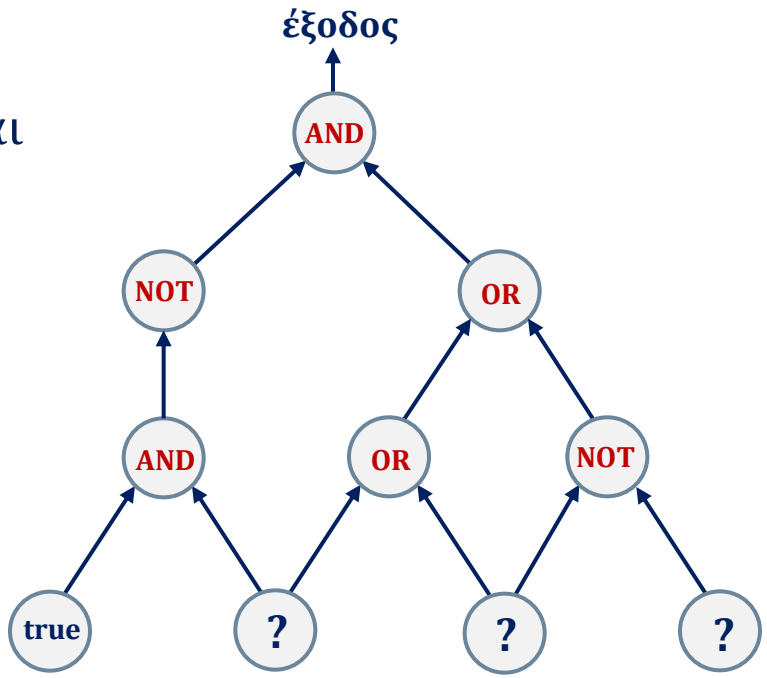
! Οποιοδηποτε-Προβλημα-NP → SAT

- **Πρόβλημα C-SAT**

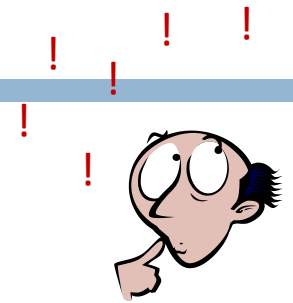
Ένα *λογικό (Boolean) κύκλωμα* -- ένα DAG του οποίου οι κόμβοι είναι **πύλες** (gates) 5 διαφ/κών τύπων :

- ✓ Πύλες **AND** και **OR** με βαθμό εισόδου 2
- ✓ Πύλες **NOT** με βαθμό εισόδου 1
- ✓ Πύλες ***γνωστής εισόδου*** με βαθμό 0
- ✓ Πύλες ***άγνωστης εισόδου*** με βαθμό 0

Ένας τερματικός κόμβος του DAG είναι η **πύλη εξόδου** (output gate)



Αναγωγές Προβλημάτων NP



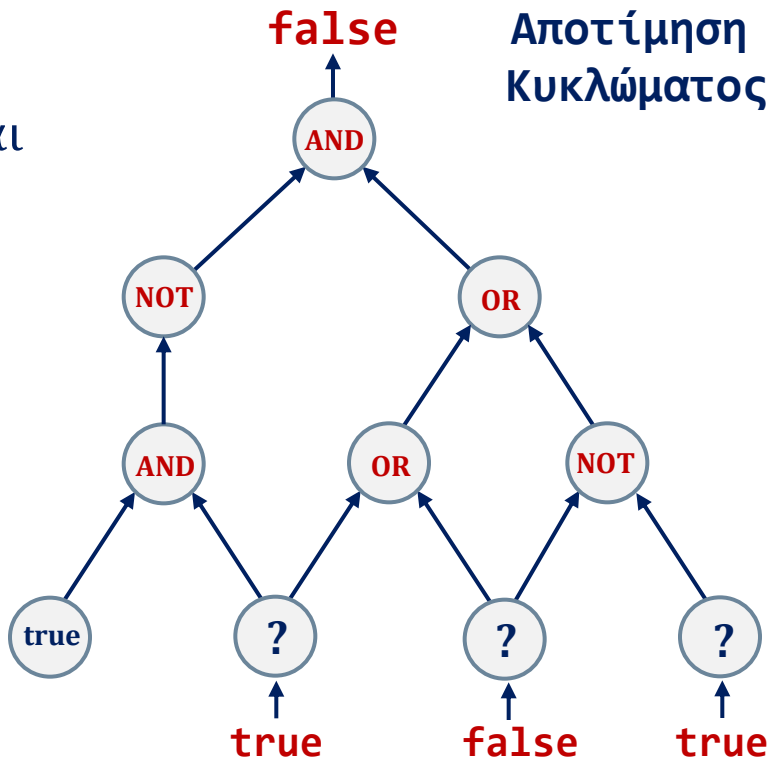
Οποιοδηποτε-Προβλημα-NP → SAT

- **Πρόβλημα C-SAT**

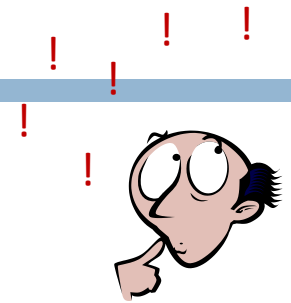
Ένα *λογικό (Boolean) κύκλωμα* -- ένα DAG του οποίου οι κόμβοι είναι **πύλες** (gates) 5 διαφ/κών τύπων :

- ✓ Πύλες **AND** και **OR** με βαθμό εισόδου 2
- ✓ Πύλες **NOT** με βαθμό εισόδου 1
- ✓ Πύλες ***γνωστής εισόδου*** με βαθμό 0
- ✓ Πύλες ***άγνωστης εισόδου*** με βαθμό 0

Ένας τερματικός κόμβος του DAG είναι η **πύλη εξόδου** (output gate)



Αναγωγές Προβλημάτων NP



! Οποιοδηποτε-Προβλημα-NP → SAT

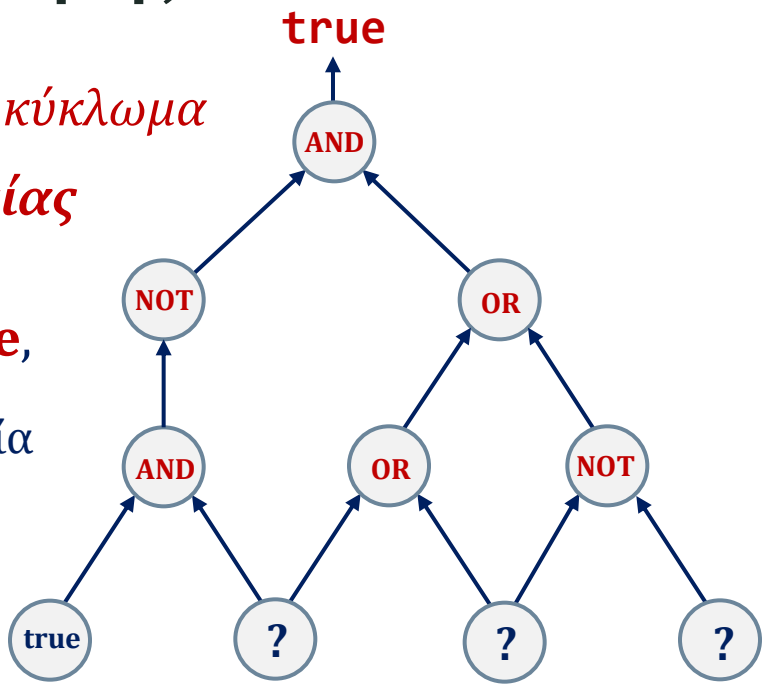
- **C-SAT ως Πρόβλημα Αναζήτησης**

Με δεδομένο ένα *λογικό (Boolean) κύκλωμα*

βρείτε μια ***ανάθεση τιμών αληθείας*** για τις άγνωστες εισόδους :

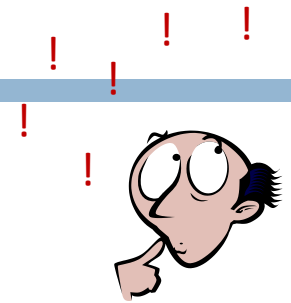
η **πύλη εξόδου** να αποτιμάται **true**,

ή να αναφέρουμε ότι δεν υπάρχει καμία τέτοια ανάθεση τιμών !!!



`(false, true, true)` ⇒ `true`
`(true, false, true)` ⇒ `false`

Αναγωγές Προβλημάτων NP

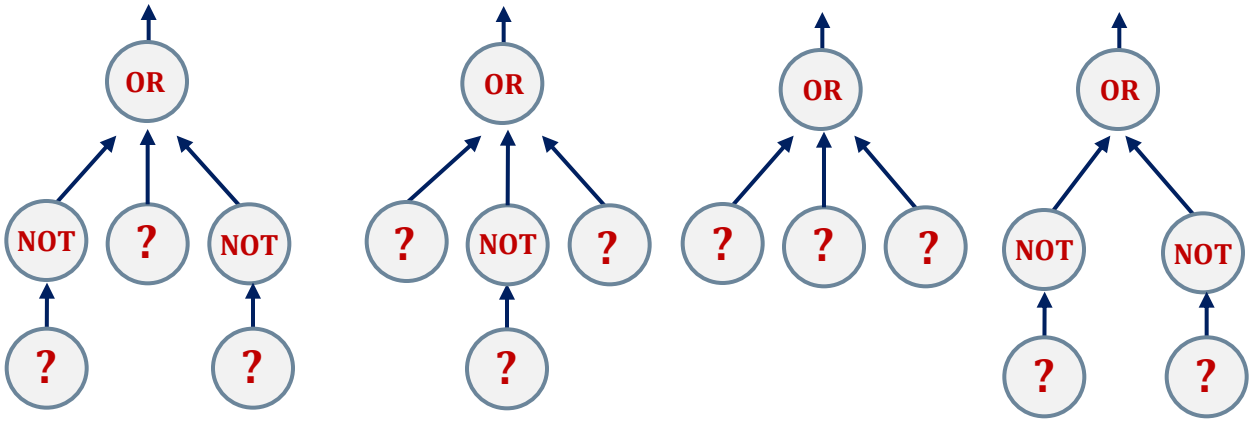


! Οποιοδηποτε-Προβλημα-NP → SAT

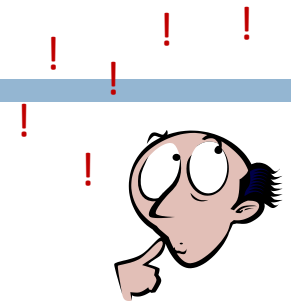
- **Κυκλωμα-SAT είναι γενίκευση του SAT**

Το SAT **αναζητά** μια *ικανοποιούσα ανάθεση τιμών αληθείας* για ***ένα λογικό κύκλωμα*** που έχει την εξής ***απλή δομή*** !!!

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



Αναγωγές Προβλημάτων NP



! Οποιοδηποτε-Προβλημα-NP → SAT

- **Κυκλωμα-SAT → SAT**

Μπορούμε να ξαναγράψουμε οποιοδήποτε λογικό κύκλωμα σε *κανονική συζευκτική μορφή* (CNF) :

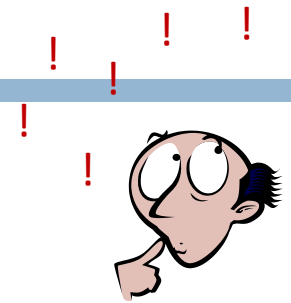
Για ***κάθε πύλη g*** του λογικού κυκλώματος **δημιουργούμε** μια

μεταβλητή g

και μοντελοποιούμε την **επίδραση της πύλης** χρησιμοποιώντας μερικούς **όρους** !!!

Να πως !!!...

Αναγωγές Προβλημάτων NP



Οποιοδηποτε-Προβλημα-NP → SAT

- **Κυκλωμα-SAT → SAT**

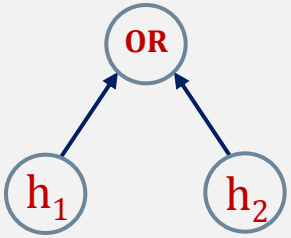
πύλη **g**

(g)

g

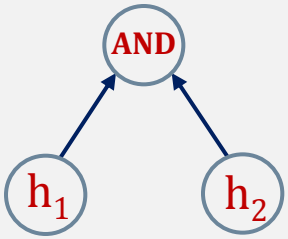
$(\overline{g})$

g



$(g \vee \overline{h}_1)$
$(g \vee \overline{h}_2)$
$(g \vee h_1 \vee h_2)$

g



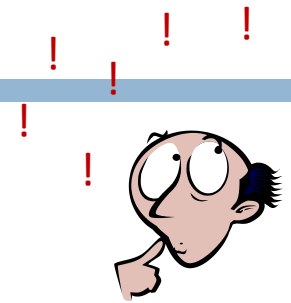
$(\overline{g} \vee h_1)$
$(\overline{g} \vee h_2)$
$(g \vee \overline{h}_1 \vee \overline{h}_2)$

g

$(g \vee h)$
$(\overline{g} \vee \overline{h})$

Βλέπετε γιατί αυτοί οι όροι επιβάλουν ακριβώς το επιθυμητό αποτέλεσμα !!!!

Αναγωγές Προβλημάτων NP



! ΟΠΟΙΟΔΗΠΟΤΕ-ΠΡΟΒΛΗΜΑ-NP → SAT

- **ΚΥΚΛΩΜΑ-SAT → SAT**

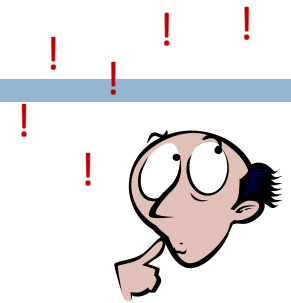
Για να ολοκληρώσουμε !!!

Εάν **g** είναι η **πύλη εξόδου**, την **αναγκάζουμε** να είναι **true** με την προσθήκη του όρου **(g)** !!!

[το **στιγμιότυπο** του **SAT** που προκύπτει] ⇔ [με το δεδομένο **στιγμιότυπο** του **SAT-ΚΥΚΛΩΜΑΤΟΣ**]

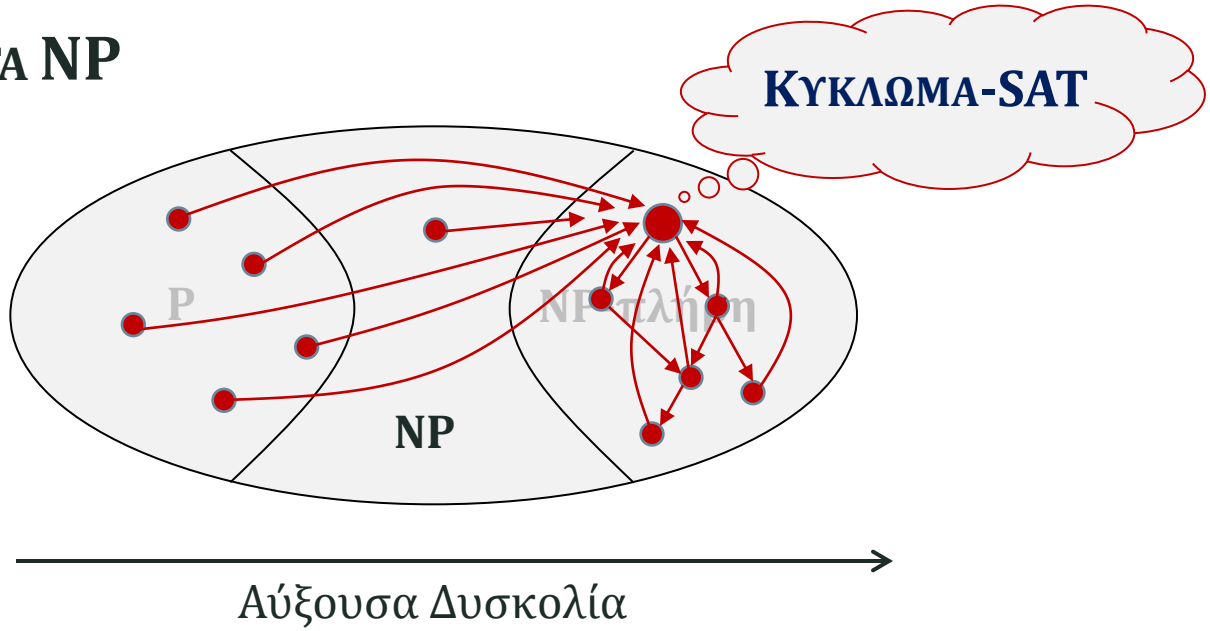
Πράγματι: οι ***ικανοποιούσες αναθέσεις τιμών αληθείας*** αυτής της κανονικής συζευκτικής μορφής (CNF) βρίσκονται σε **αμφιμονοσήμαντη αντιστοιχία** με ***εκείνες του κυκλώματος*** !!!

Αναγωγές Προβλημάτων NP



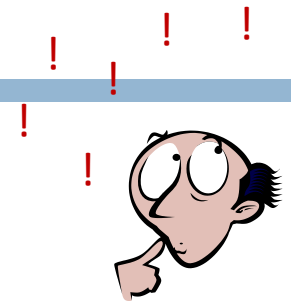
Οποιοδηποτε-Προβλημα-NP → SAT

- **Προβλήματα NP**



Κυκλωμα-SAT → SAT

Αναγωγές Προβλημάτων NP



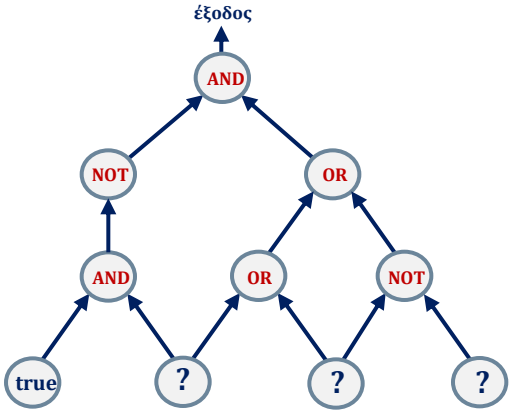
! Οποιοδηποτε-Προβλημα-NP → SAT

- **Αναγωγή**

Έστω **A** ένα πρόβλημα στο **NP** !!!

Θέλουμε μια αναγωγή

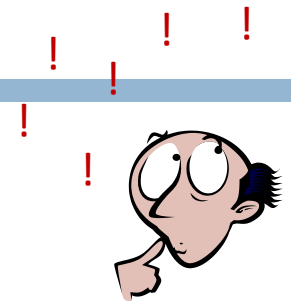
A → Κυκλωμα-SAT



Αυτό φαίνεται δύσκολο !!!...

καθώς *δεν γνωρίζουμε σχεδόν τίποτε* για το πρόβλημα A !!!

Αναγωγές Προβλημάτων NP



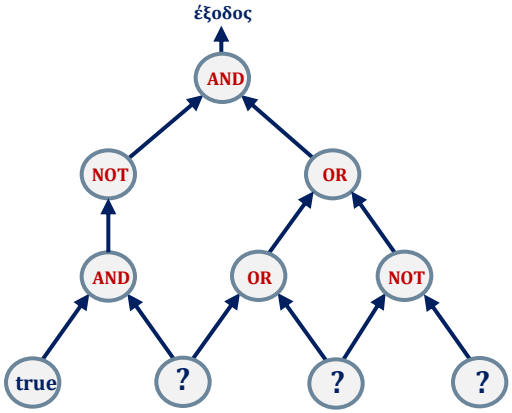
! Οποιοδηποτε-Προβλημα-NP → SAT

- **Αναγωγή**

Έστω **A** ένα πρόβλημα στο **NP** !!!

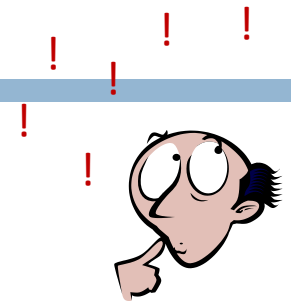
Το **ΜΟΝΟ** που γνωρίζουμε για το **A**

Πρόβλημα Αναζήτησης



Χαρακτηριστικό: Οποιαδήποτε **προτεινόμενη λύση** μπορεί να **ελεγχθεί γρήγορα** για την ορθότητά της !!!

Αναγωγές Προβλημάτων NP



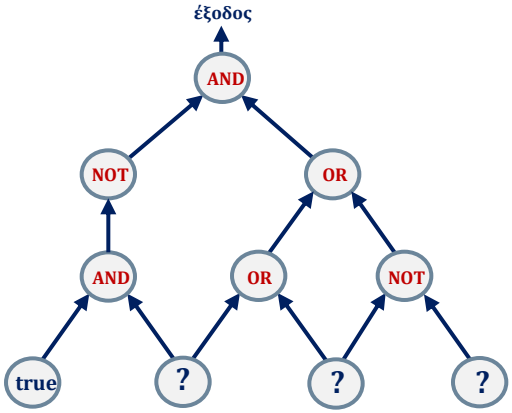
! ΟΠΟΙΟΔΗΠΟΤΕ-ΠΡΟΒΛΗΜΑ-NP → SAT

- **Αναγωγή**

Έστω **A** ένα πρόβλημα στο **NP** !!!

Το **ΜΟΝΟ** που γνωρίζουμε για το **A**

Πρόβλημα Αναζήτησης

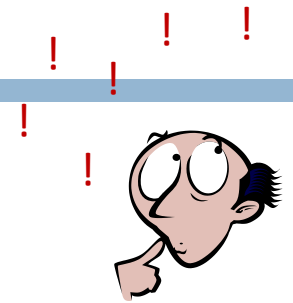


Πολυωνυμικός αλγόριθμος ελέγχου **C(I, S)** με μήκος εισόδου **|I|** !!!

✓ Υποθέτουμε : `I = (1001110010101001)`

`S = (1101010011101101)` και **|S|=*O*(|I|)**

Αναγωγές Προβλημάτων NP

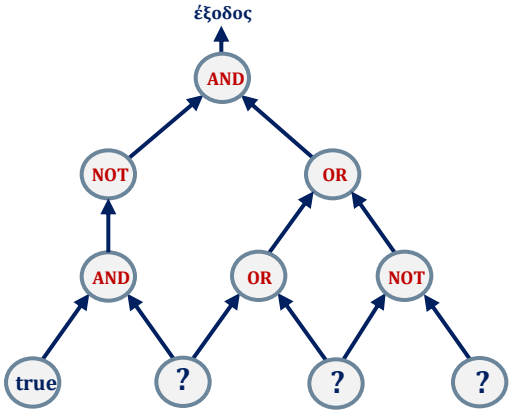


Οποιοδηποτε-Προβλημα-NP → SAT

- **Αναγωγή**

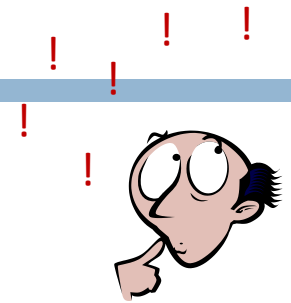
Γνωρίζουμε ότι:

Οποιοσδήποτε πολυωνυμικός αλγόριθμος μπορεί να *μετατραπεί* σε κύκλωμα !!!



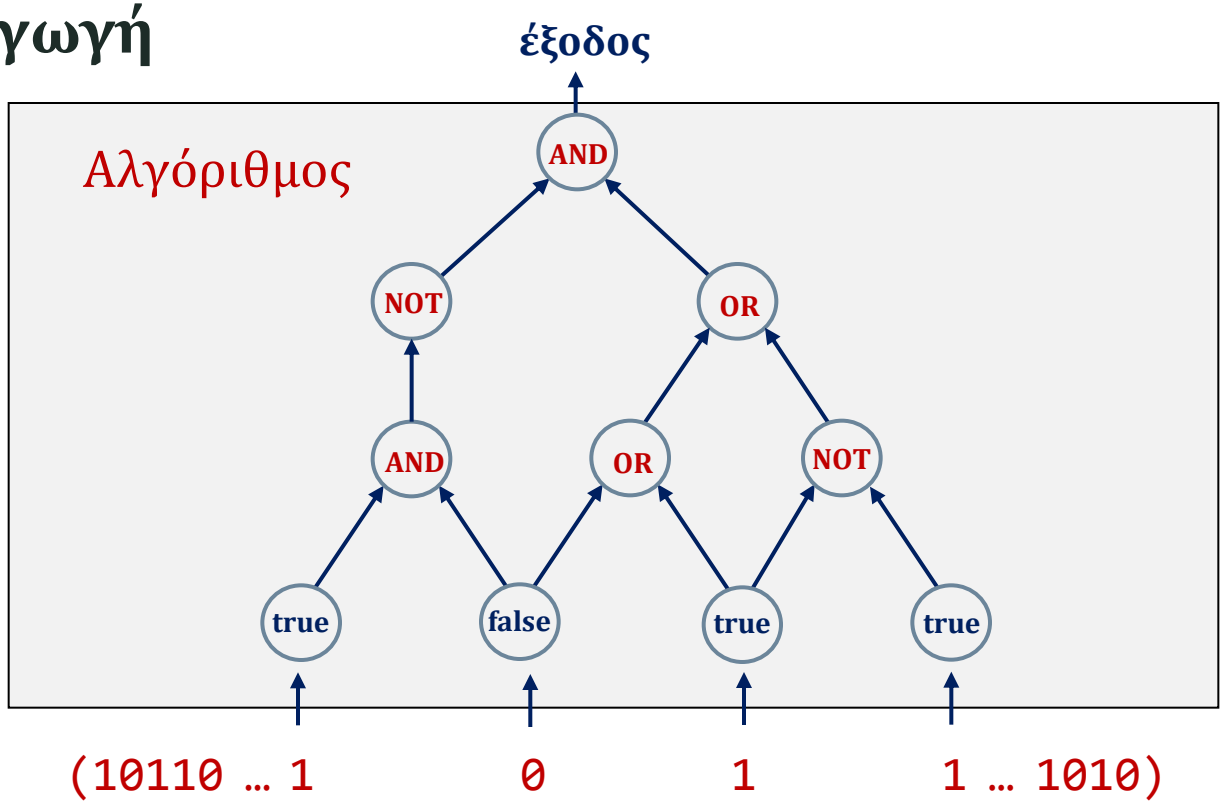
Οι **πύλες εισόδου του κυκλώματος** κωδικοποιούν την είσοδο του αλγόριθμου `I = (1001110010101101 10)`

Αναγωγές Προβλημάτων NP

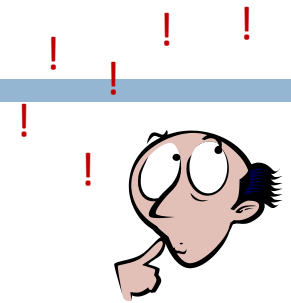


Οποιοδηποτε-Προβλημα-NP → SAT

- **Αναγωγή**

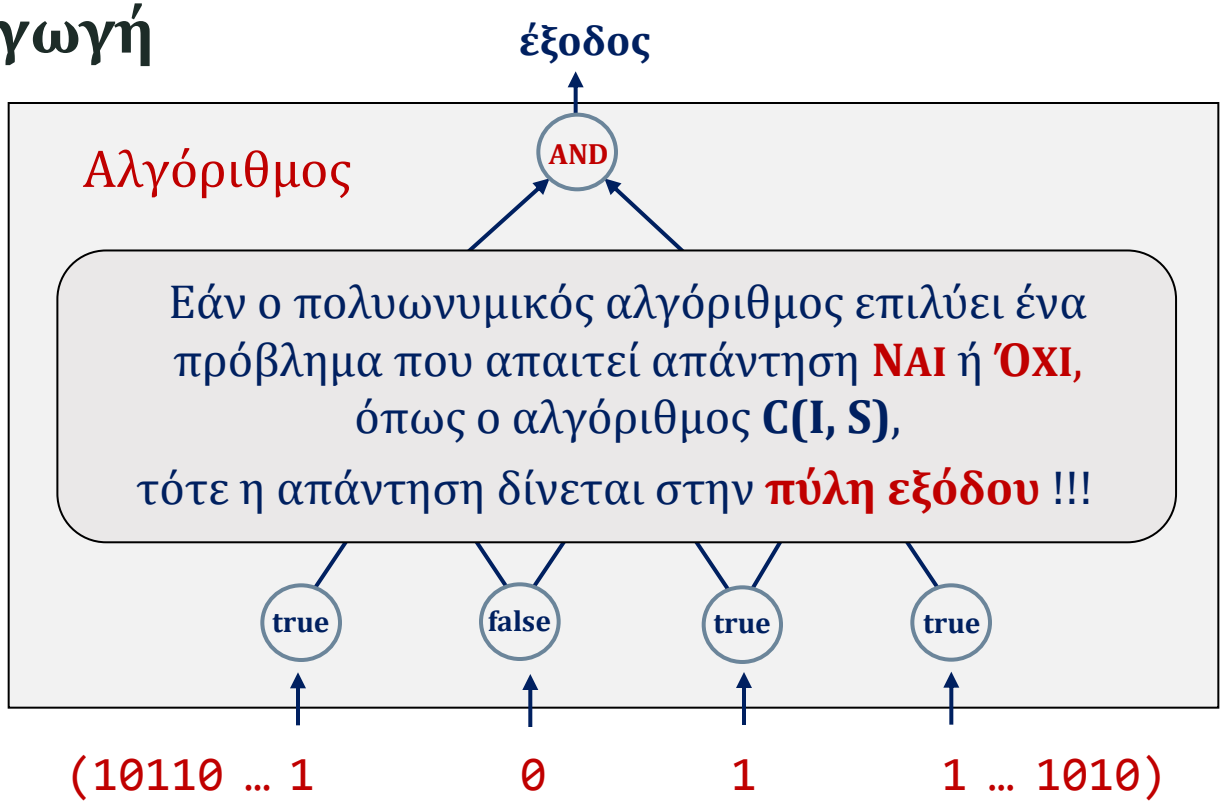


Αναγωγές Προβλημάτων NP

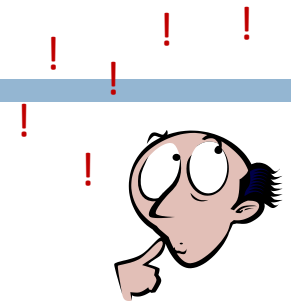


Οποιοδηποτε-Προβλημα-NP → SAT

- **Αναγωγή**



Αναγωγές Προβλημάτων NP

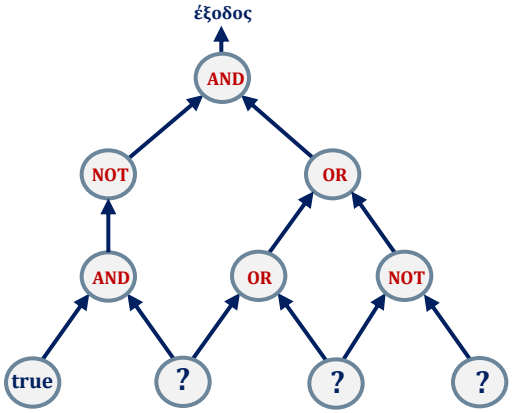


! Οποιοδηποτε-Προβλημα-NP → SAT

- **Αναγωγή**

Τι συμπεράνουμε ?

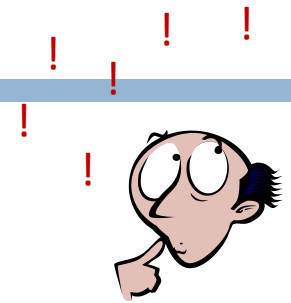
Για οποιοδήποτε στιγμιότυπο **I** του προβλήματος αναζήτησης **A**



Μπορούμε να **κατασκευάσουμε** σε *πολυωνυμικό χρόνο* ένα **κύκλωμα** του οποίου οι ***γνωστές είσοδοι*** είναι τα bit του **I** και οι ***άγνωστες είσοδοι*** είναι τα bit του **S** :

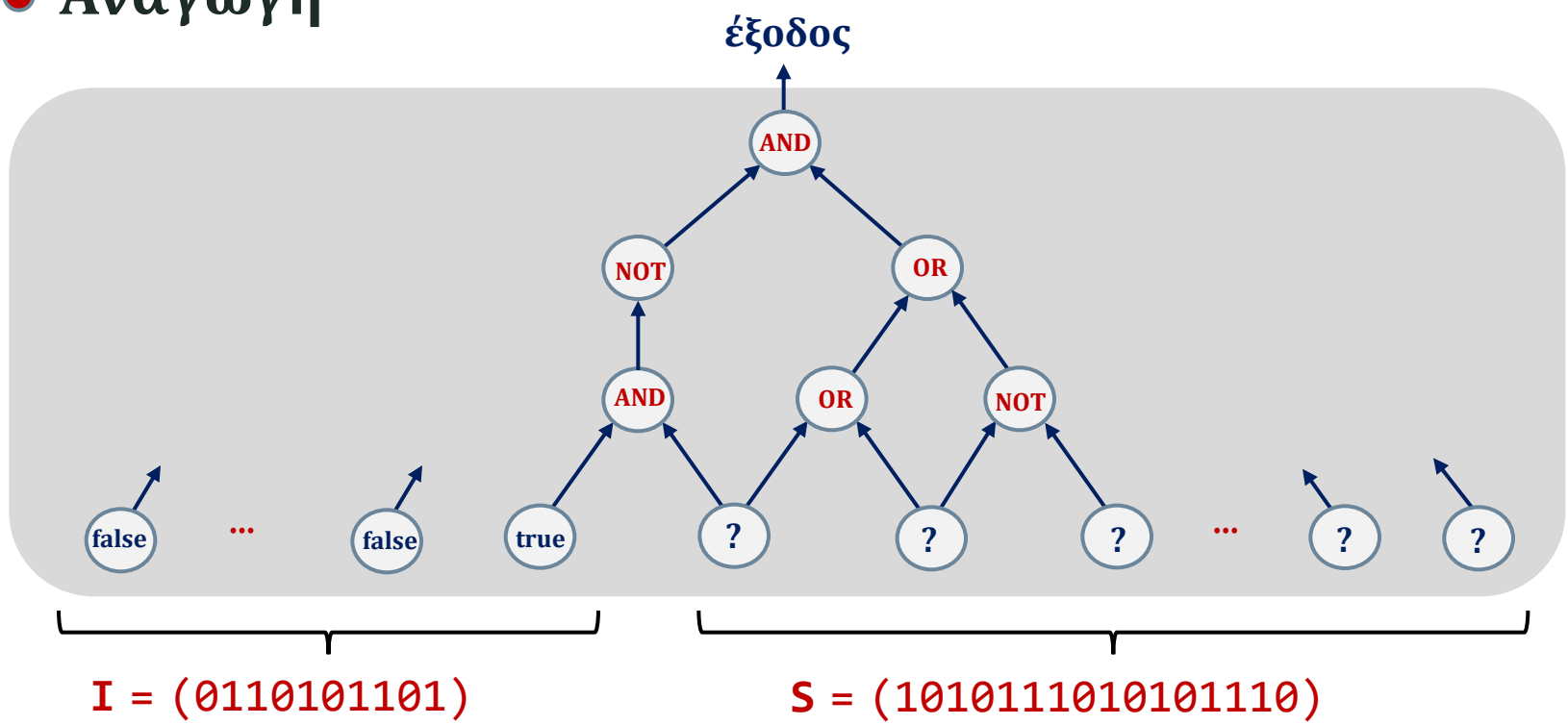
Η έξοδός του να είναι **true εάν-ν** οι *άγνωστες είσοδοι* αποτελούν μια λύση **S** του **I** !!!

Αναγωγές Προβλημάτων NP

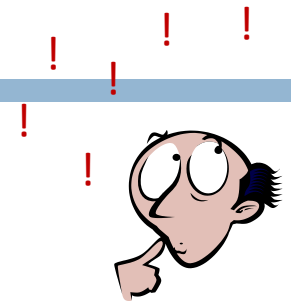


Οποιοδηποτε-Προβλημα-NP → SAT

- **Αναγωγή**

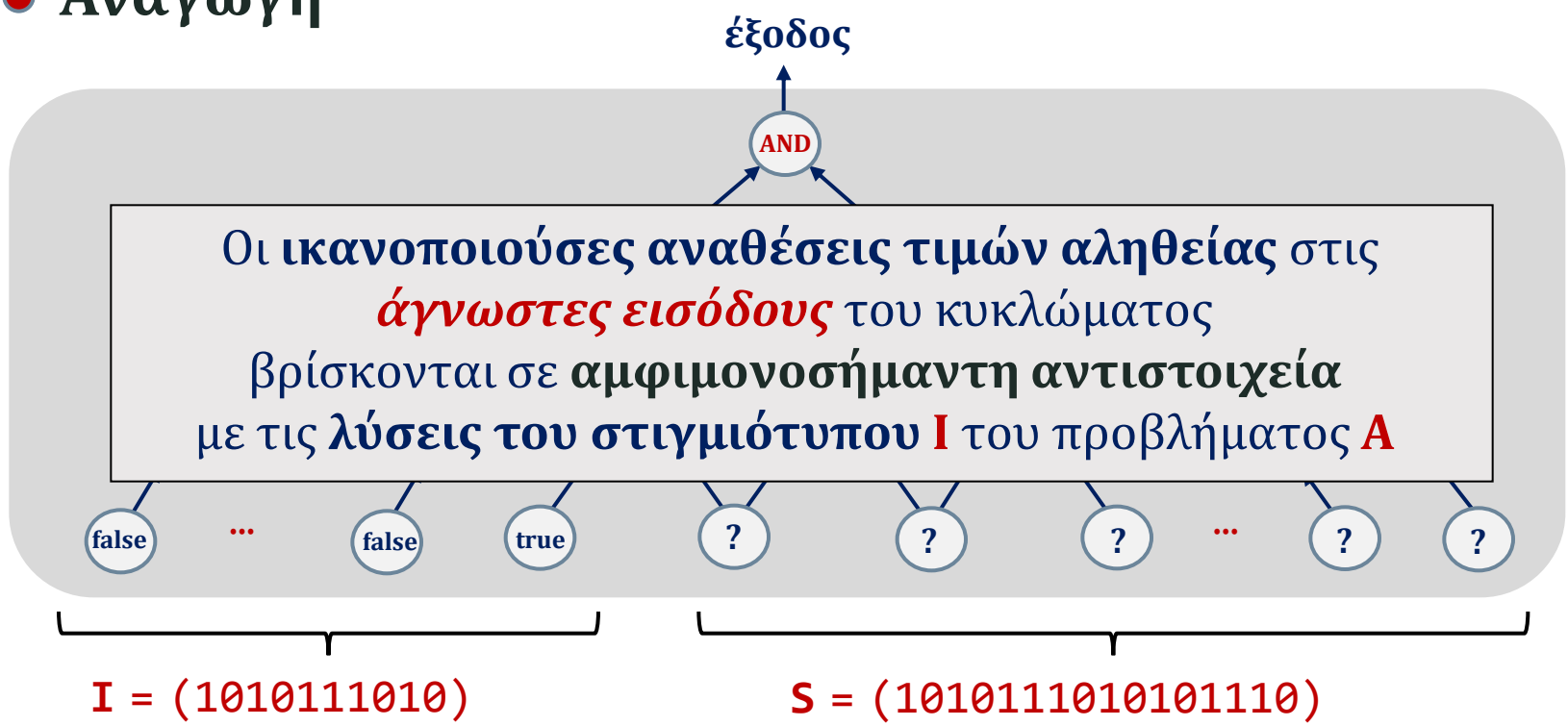


Αναγωγές Προβλημάτων NP

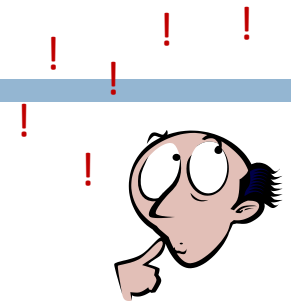


! Οποιοδηποτε-Προβλημα-NP → SAT

- **Αναγωγή**

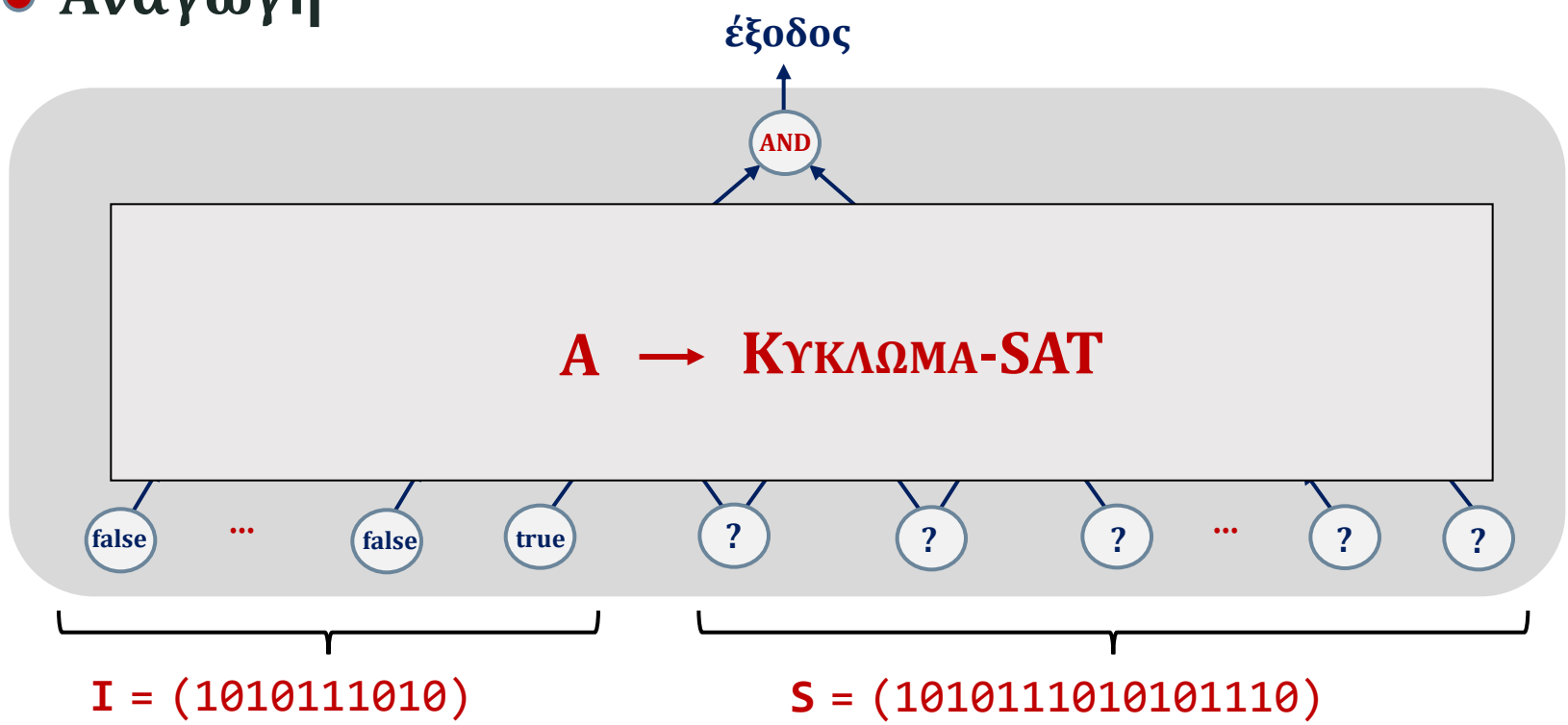


Αναγωγές Προβλημάτων NP

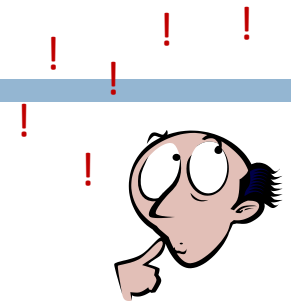


Οποιοδηποτε-Προβλημα-NP → SAT

- **Αναγωγή**

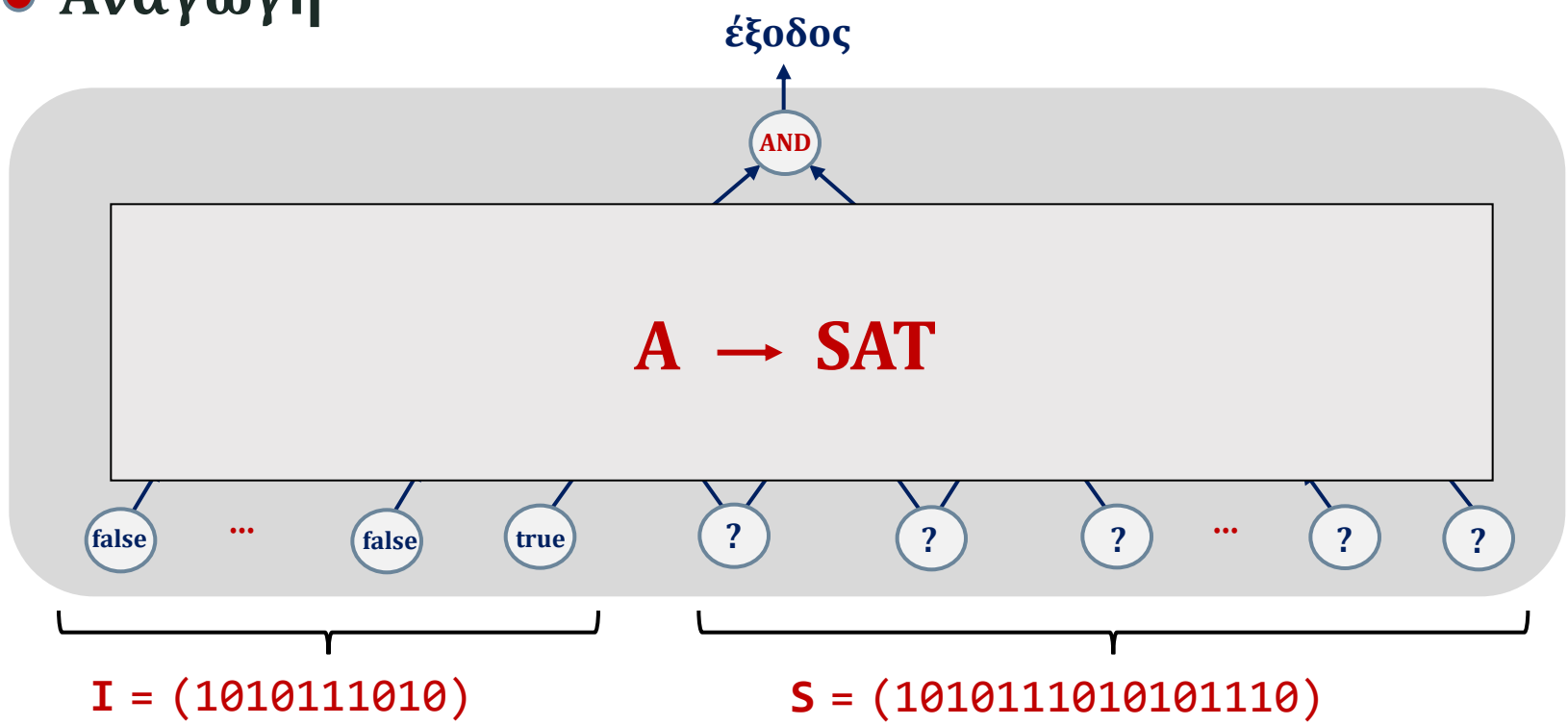


Αναγωγές Προβλημάτων NP

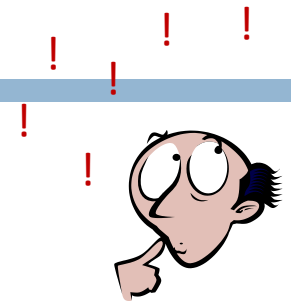


Οποιοδηποτε-Προβλημα-NP → SAT

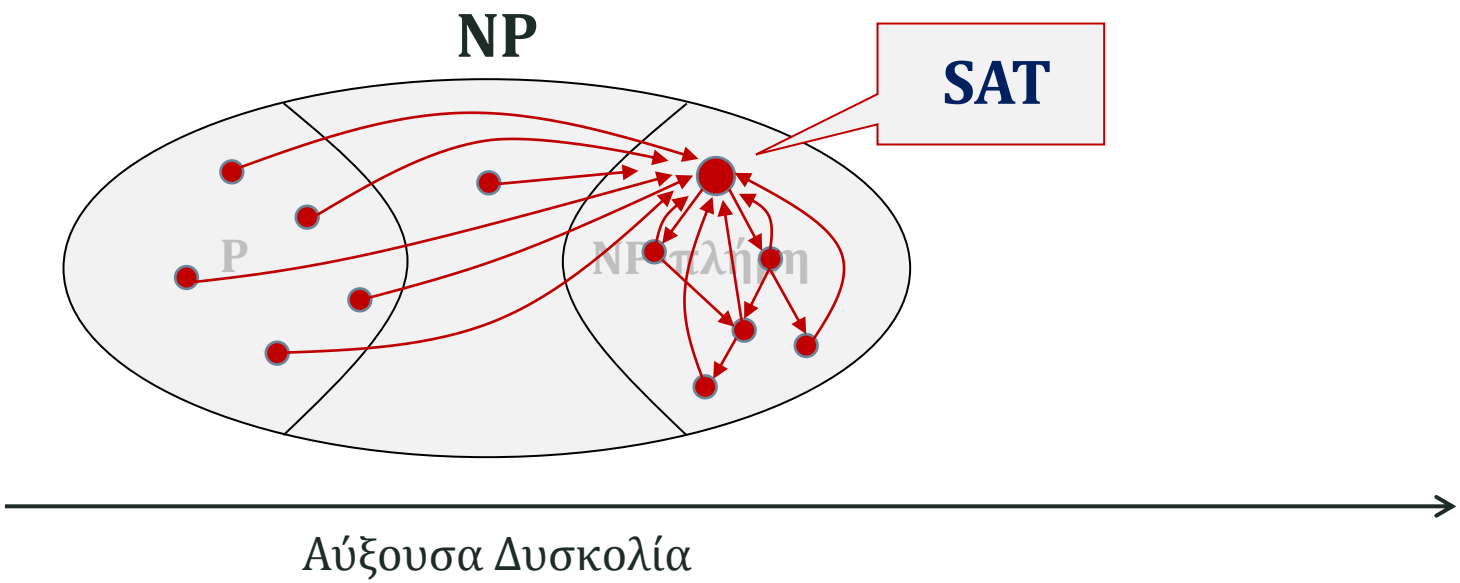
- **Αναγωγή**



Αναγωγές Προβλημάτων NP



Οποιοδηποτε-Προβλημα-NP → SAT



NP-πλήρη, NP-δύσκολα, Μη-επιλύσιμα

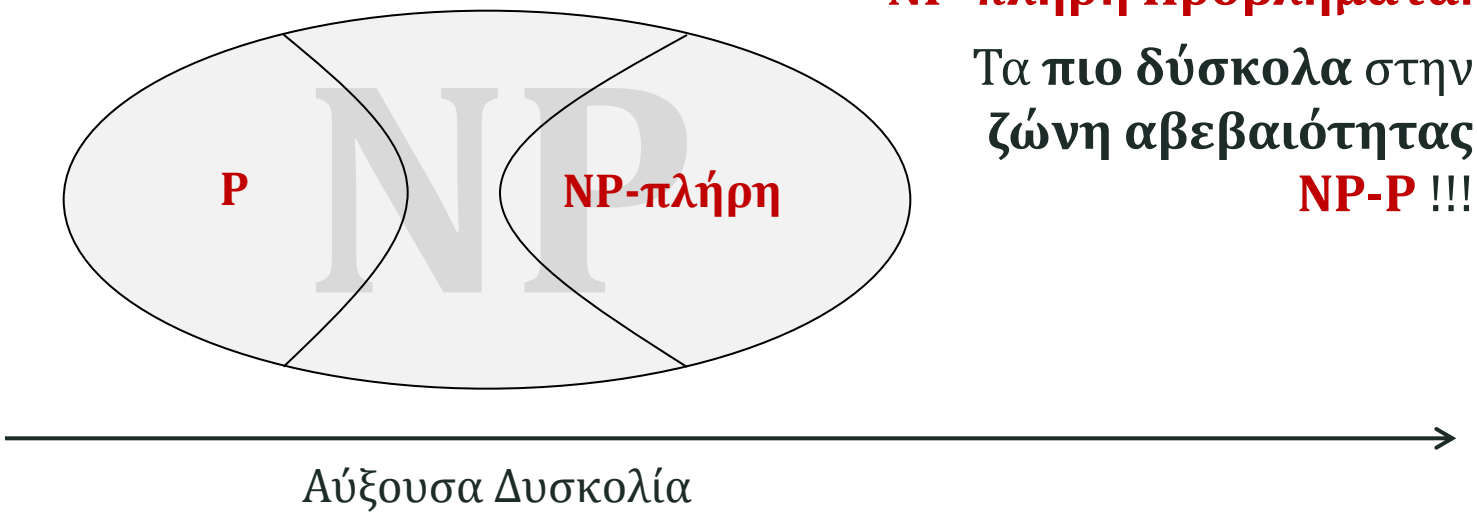
Προβλήματα NP-πλήρη !!!

- **Εάν P ≠ NP**

«Ζώνη Αβεβαιότητας» NP-P

NP-πλήρη Προβλήματα:

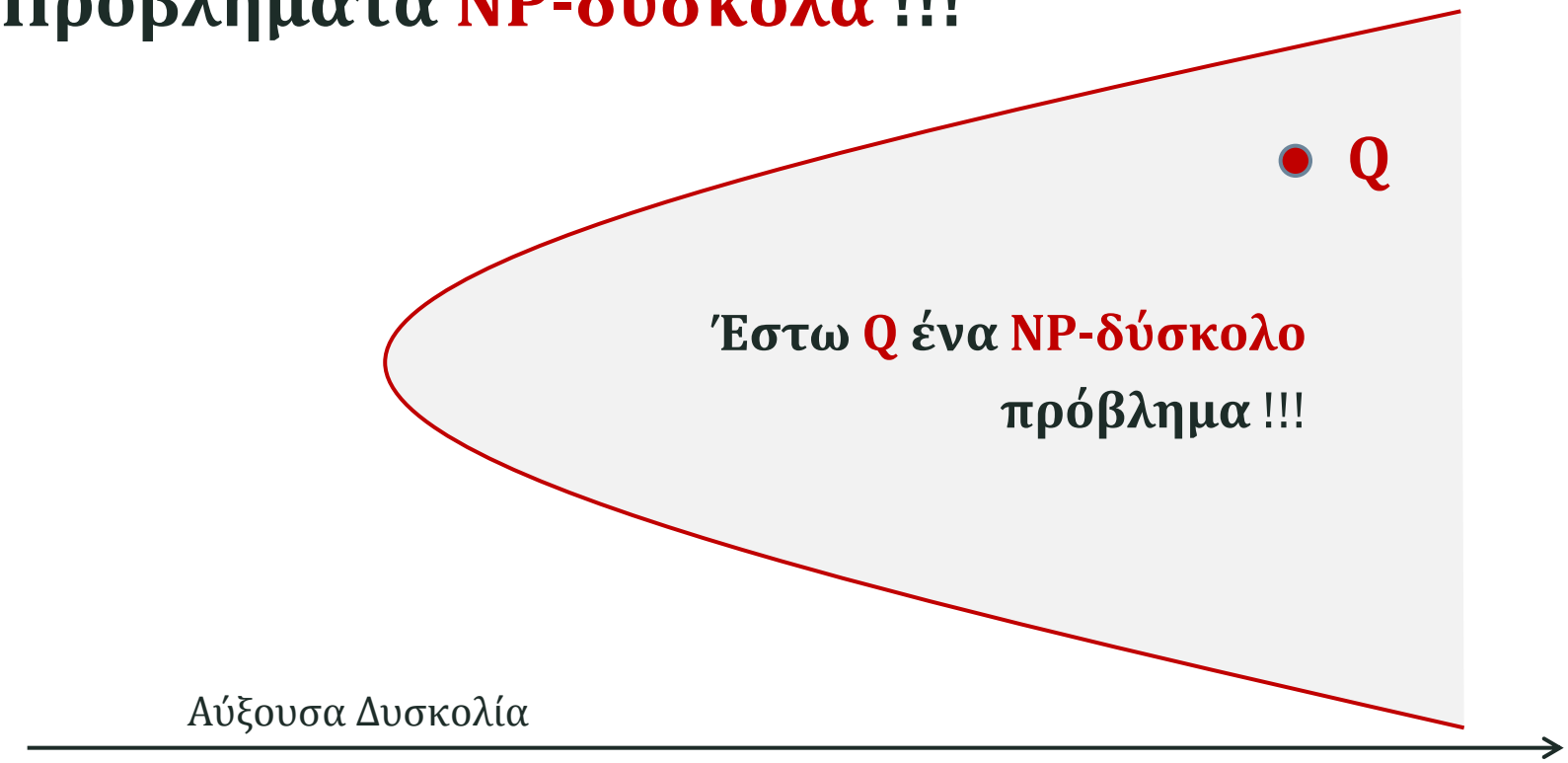
Τα **πιο δύσκολα** στην **ζώνη αβεβαιότητας NP-P** !!!



Μια Σχηματική Παρουσίαση !!!

NP-πλήρη, NP-δύσκολα, Μη-επιλύσιμα

- **Προβλήματα NP-δύσκολα !!!**



Μια Σχηματική Παρουσίαση !!!

NP-πλήρη, NP-δύσκολα, Μη-επιλύσιμα

Προβλήματα NP-δύσκολα !!!

Δεν έχουν όλα τα **Q πολυωνυμικό αλγόριθμο ελέγχου C(I,S)** !!!

Q

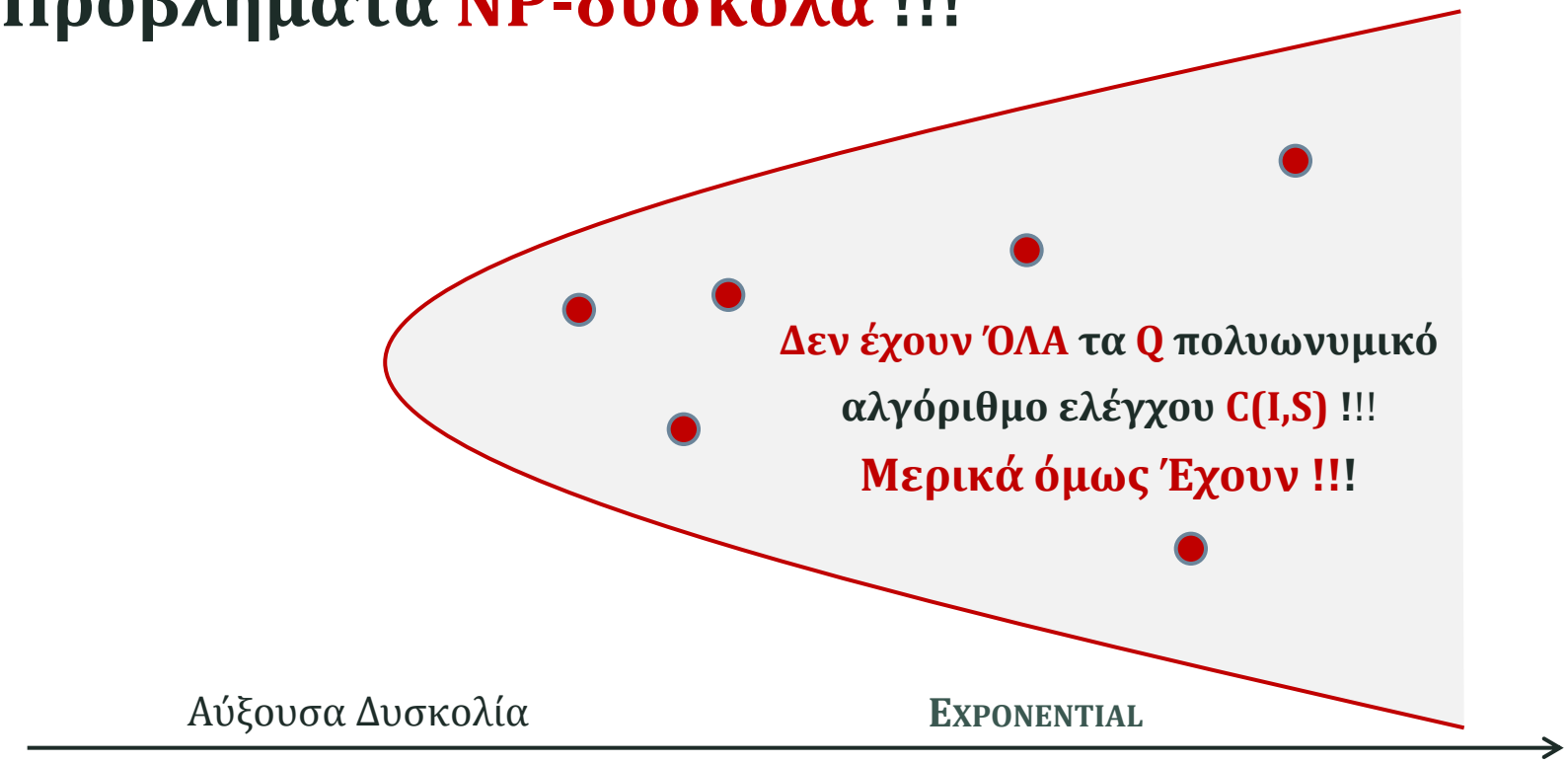
- ✓ Για κάθε $A \in NP \Rightarrow \mathbf{A \leq Q}$
- ✓ $\mathbf{Q \in P} \Rightarrow P = NP$
- ✓ Η **ύπαρξη** ενός **πολύωνυμικού αλγόριθμου** για το **Q** συνεπάγεται την ύπαρξη ενός πολυωνυμικού αλγόριθμου ***για κάθε πρόβλημα*** στο **NP** !

Αύξουσα Δυσκολία → **EXPONENTIAL**

Μια Σχηματική Παρουσίαση !!!

NP-πλήρη, NP-δύσκολα, Μη-επιλύσιμα

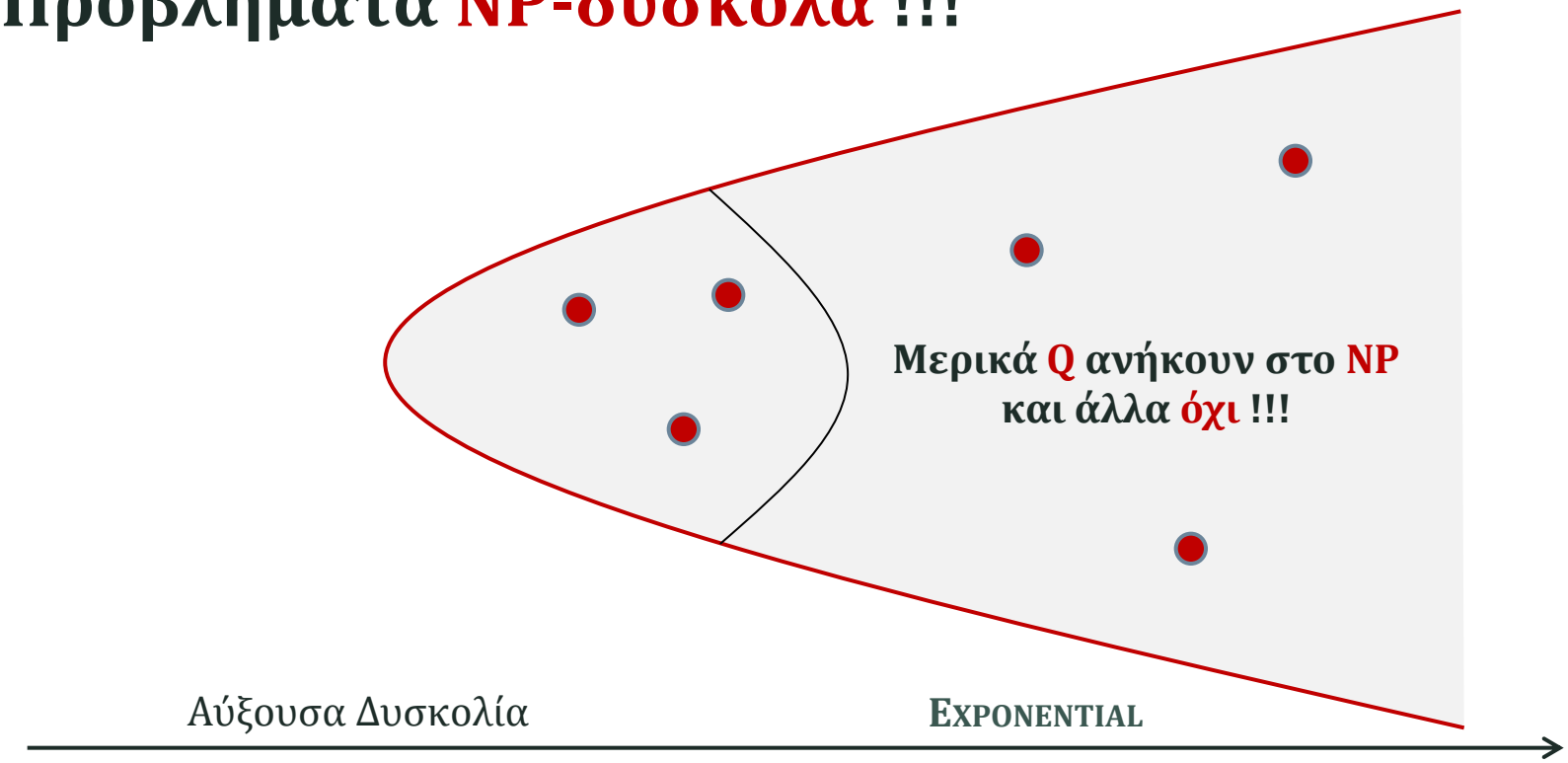
- **Προβλήματα NP-δύσκολα !!!**



Μια Σχηματική Παρουσίαση !!!

NP-πλήρη, NP-δύσκολα, Μη-επιλύσιμα

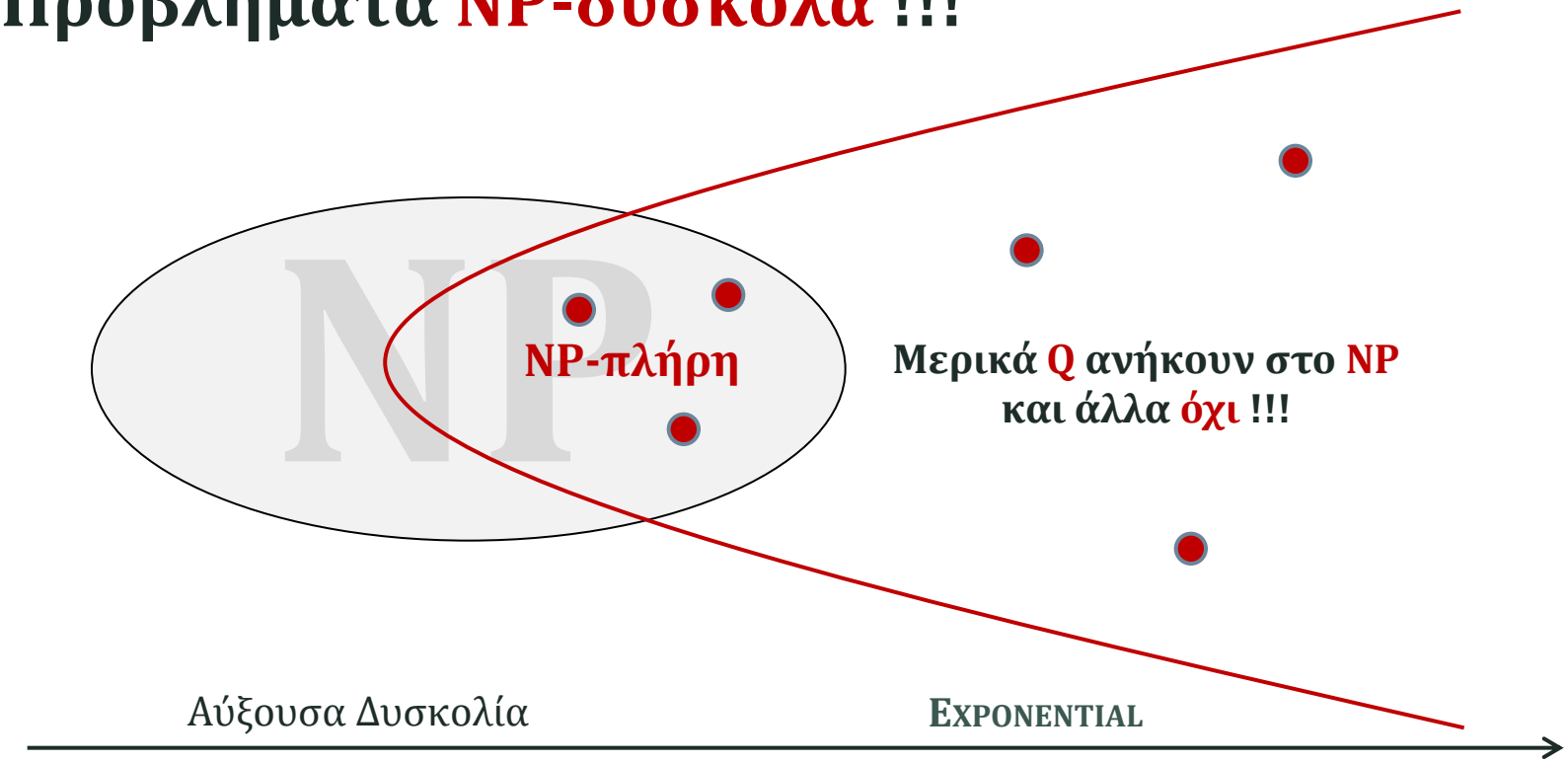
- **Προβλήματα NP-δύσκολα !!!**



Μια Σχηματική Παρουσίαση !!!

NP-πλήρη, NP-δύσκολα, Μη-επιλύσιμα

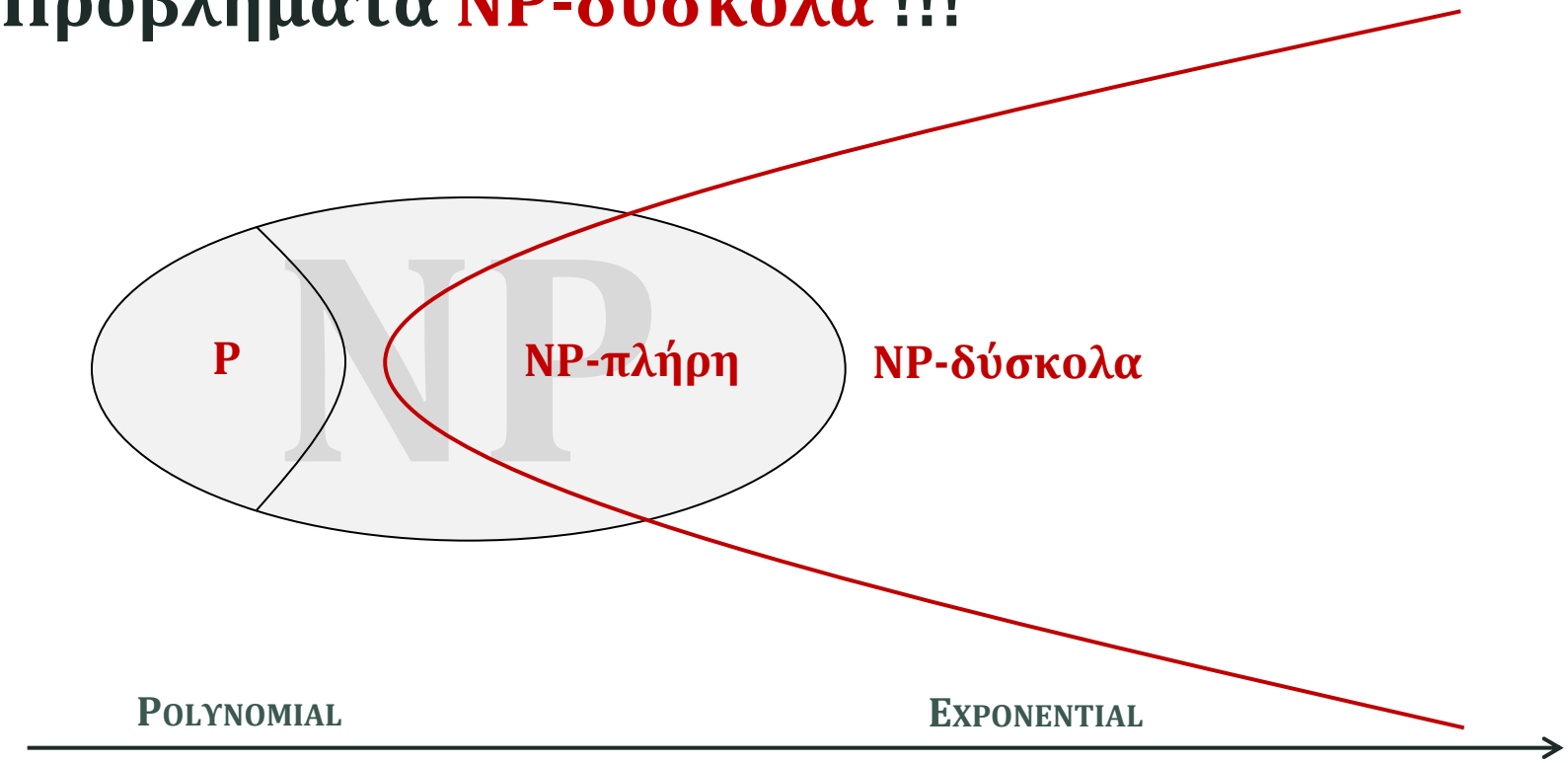
- **Προβλήματα NP-δύσκολα !!!**



Μια Σχηματική Παρουσίαση !!!

NP-πλήρη, NP-δύσκολα, Μη-επιλύσιμα

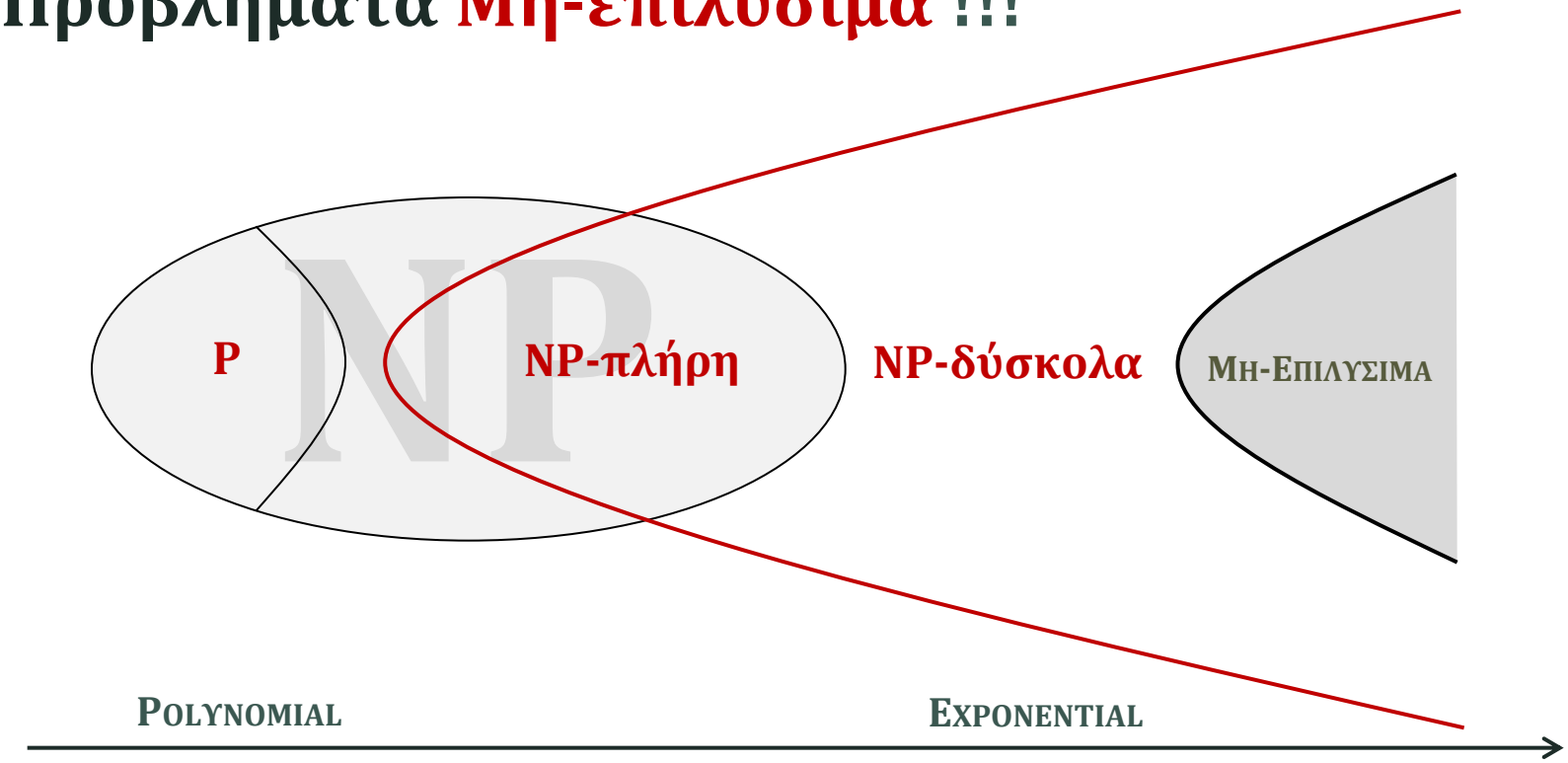
- **Προβλήματα NP-δύσκολα !!!**



Μια Σχηματική Παρουσίαση !!!

NP-πλήρη, NP-δύσκολα, Μη-επιλύσιμα

- **Προβλήματα Μη-επιλύσιμα !!!**



Μια Σχηματική Παρουσίαση !!!

Μη-επιλύσιμα Προβλήματα

Αριθμητική Εκδοχή του SAT !!!

- Ένα διάσημο πρόβλημα ***αυτού του είδους*** (μη-επιλύσιμο) είναι η ***αριθμητική εκδοχή*** του **SAT**

Με δεδομένη μια πολυωνυμική εξίσωση πολλών μεταβλητών, όπως για παράδειγμα

$$x^3yz + 2y^4z^2 - 7xy^5z = 6$$

υπάρχουν ακέραιες τιμές των μεταβλητών ***x, y, z*** οι οποίες να την ικανοποιούν ?

Μη-επιλύσιμα Προβλήματα

Το Πρόβλημα του Τερματισμού !!!

- Το **πρώτο μη-επιλύσιμο** πρόβλημα ανακαλύφθηκε το **1936** από τον **Alan Turing**, τότε φοιτητή των μαθηματικών στο Cambridge της Αγγλίας !!!

Το Πρόβλημα του Τερματισμού - HP

- ✓ Όταν το διατύπωσε ο Turing δεν υπήρχαν Η/Υ ή γλώσσες προγραμματισμού.
- ✓ Εμφανίστηκαν αργότερα ακριβώς επειδή είχε αυτή την ευφυή ιδέα !!!

Μη-επιλύσιμα Προβλήματα

Το Πρόβλημα του Τερματισμού !!!

- Έστω ένα *πρόγραμμα* **p** σε μια γλώσσα προγραμματισμού, μαζί με μια *είσοδο* **x** !!!

  Μπορείτε να απαντήσετε εάν το **p** με την είσοδο **x** θα μπορέσει ποτέ να **τερματίσει** !!!

Τι θα θέλαμε ?

Έναν Αλγόριθμο, ας τον πούμε **`terminates(p,x)`**, ο οποίος μετά από υπολογισμούς θα μας έλεγε εάν το **p** με είσοδο **x** θα τερματίσει την εκτέλεσή του !!!

Σημείωση: Υπάρχουν προγράμματα (αλγόριθμοι) που δέχονται ως είσοδο προγράμματα, π.χ., οι μεταγλωττιστές !

Μη-επιλύσιμα Προβλήματα

Το Πρόβλημα του Τερματισμού !!!

- Λοιπόν, μην προσπαθήσετε… δεν μπορείτε!…

Τέτοιος αλγόριθμος δεν υπάρχει !!!

Να και η απόδειξη !!!

Εάν υποθέσουμε ότι είχατε το πρόγραμμα **`terminates(p,x)`**, τότε θα μπορούσατε να κατασκευάσετε το ακόλουθο πρόγραμμα:

```
paradox(z: file)
if terminates(z,z) then loop_forever else halt
```

Μη-επιλύσιμα Προβλήματα

Το Πρόβλημα του Τερματισμού !!!

```
paradox(z: file)
if terminates(z,z) then loop_forever else halt
```

- Το **paradox** ***τερματίζει*** **εάν-ν** το **z** ***δεν τερματίζει*** !!! όταν του δοθεί ως είσοδος ο δικός του κώδικας.

  Εάν εκτελούσαμε το πρόγραμμα **paradox(paradox)** ?

  Τότε,

  το **paradox** **τερματίζει εάν-ν** το **paradox** **δεν τερματίζει !!!**

  Αντίφαση !!!

Μη-επιλύσιμα Προβλήματα

Το Πρόβλημα του Τερματισμού !!!

- Καταλήξαμε σε αντίφαση !!!

Επομένως, το πρόβλημα HP δεν μπορεί να λυθεί με *κανένα αλγόριθμο* !!!

Αυτό μας λέει κάτι σημαντικό:

- ✓ **Ο προγραμματισμός ποτέ δεν θα αυτοματοποιηθεί** !!!
- ✓ Και πάντοτε θα βασίζεται στην **επινοητικότητα** !!!.. και τη **σοβαρή μελέτη** !!!

Τέλος Ενότητας