Θεωρία Υπολογισμού

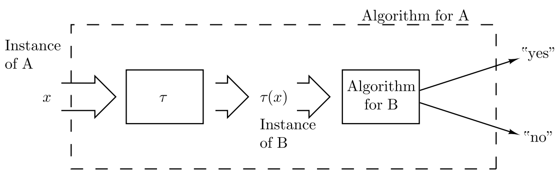
$\mathcal{NP}$-δύσκολα και $\mathcal{NP}$-πλήρη Προβλήματα
($\mathcal{NP}$-hard and $\mathcal{NP}$-complete Problems)

2003-04

Αναγωγές Πολυωνυμικού Χρόνου

- **Ορισμός.** Έστω Σ ένα αλφάβητο. Μια συνάρτηση $f : \Sigma^* \to \Sigma^*$ ονομάζεται **υπολογίσιμη σε πολυωνυμικό χρόνο** αν υπάρχει πολυωνυμικά φραγμένη μηχανή Turing M που υπολογίζει την f.
- **Ορισμός.** Έστω γλώσσες $L_1, L_2 \subseteq \Sigma^*$. Μια υπολογίσιμη σε πολυωνυμικό χρόνο συνάρτηση $\tau : \Sigma^* \to \Sigma^*$ λέγεται **πολυωνυμική αναγωγή από την L_1 στην L_2** αν για κάθε $x \in \Sigma^*$, $x \in L_1$ αν και μόνο αν $\tau(x) \in L_2$.

Αναγωγές Πολυωνυμικού Χρόνου



Αναγωγές Πολυωνυμικού Χρόνου

- Αν υπάρχει πολυωνυμική αναγωγή από ένα πρόβλημα A σε ένα πρόβλημα B τότε το **πρόβλημα B είναι τουλάχιστον τόσο δύσκολο όσο και το A.** Δηλ. αν το A χρειάζεται **εκθετικό χρόνο** για να λυθεί, τότε και το B χρειάζεται τουλάχιστον εκθετικό χρόνο για να λυθεί.
- Αν το B λύνεται σε **πολυωνυμικό χρόνο**, τότε το ίδιο ισχύει και για το A.

Παράδειγμα

- 3-SATISFIABILITY: Έστω τύπος του Boole F με την ιδιότητα ότι κάθε clause έχει το πολύ 3 literals. Είναι ο F ικανοποιήσιμος;
- Το πρόβλημα 3-SATISFIABILITY είναι μια **μερική περίπτωση** του προβλήματος SATISFIABILITY.
- INDEPENDENT SET: Έστω μη κατευθυνόμενος γράφος $G \subseteq V \times V$ και ακέραιος $K \geq 2$. Υπάρχει $C \subseteq V$ με $|C| \geq K$ τέτοιο ώστε για όλες τις κορυφές $v_i, v_j \in C$, δεν υπάρχει ακμή που να συνδέει τις v_i και v_j;

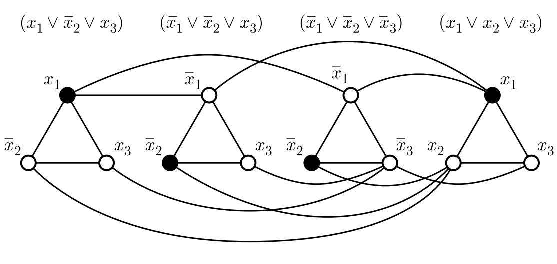
Από το 3-satisfiability στο independent set

- Έστω ότι έχουμε μια περίπτωση (instance) του προβλήματος 3-SATISFIABILITY δηλ. ένα τύπο του Boole F με clauses $C_1, \ldots, C_m$ που έχουν **ακριβώς 3 literals.**
- Μπορούμε να ορίσουμε ένα μη κατευθυνόμενο γράφο G και ένα ακέραιο K ώστε υπάρχει ένα σύνολο από K κορυφές στον G που δεν συνδέονται μεταξύ τους με κάποια ακμή (δηλ. υπάρχει ένα independent set με K στοιχεία) αν και μόνο αν ο τύπος F είναι ικανοποιήσιμος.

Απόδειξη

- Ο γράφος G ορίζεται ως εξής:
  - Για κάθε clause C_i, $1 \leq i \leq m$ του F, ο G έχει τρεις κορυφές που παριστάνουν τα literals του C_i. Οι κορυφές αυτές συνδέονται με ακμές ώστε να σχηματίζεται ένα τρίγωνο.
    Συνολικά ο G έχει $3m$ κορυφές.
  - Δύο κορυφές του G συνδέονται με μια ακμή αν και μόνο αν παριστάνουν literals που το ένα είναι η άρνηση του άλλου.
- Ο ακέραιος K είναι ίσος με m.
- Η παραπάνω αναγωγή είναι υπολογίσιμη σε πολυωνυμικό χρόνο.
- **Ισχύει:** το F είναι ικανοποιήσιμο **ανν** το G έχει INDEPENDENT SET με μέγεθος m.

Απόδειξη



Σύνθεση Πολυωνυμικών Αναγωγών

- **Ορισμός.** Η **σύνθεση** δύο συναρτήσεων $f : A \to B$ και $g : B \to C$ είναι μια συνάρτηση $f \circ g : A \to C$ τέτοια ώστε για κάθε $x \in A$, $f \circ g(x) = g(f(x))$.
- **Λήμμα.** Αν τ_1 είναι μια πολυωνυμική αναγωγή από τη γλώσσα L_1 στη γλώσσα L_2, και τ_2 είναι μια πολυωνυμική αναγωγή από τη γλώσσα L_2 στη γλώσσα L_3, τότε η σύνθεση τους $\tau_1 \circ \tau_2$ είναι μια πολυωνυμική αναγωγή από την L_1 στην L_3.

9

Απόδειξη

- Υποθέτουμε ότι οι τ_1 και τ_2 υπολογίζονται από μηχανές Turing σε χρόνους p_1 και p_2 (πολυώνυμα). Τότε η $\tau_1 \circ \tau_2$ υπολογίζεται από τη μηχανή $M_1 M_2$.
- Με είσοδο x, η $M_1 M_2$ θα υπολογίσει το $(\tau_1 \circ \tau_2)(x)$ σε χρόνο $p_1(|x|) + p_2(p_1(|x|))$.
- Επίσης $x \in L_1$ ανν $\tau_1(x) \in L_2$ ανν $\tau_2(\tau_1(x)) \in L_3$ ανν $\tau_1 \circ \tau_2(x) \in L_3$.

10

$\mathcal{NP}$-δύσκολα και $\mathcal{NP}$-πλήρη Προβλήματα

- **Ορισμός.** Μια γλώσσα $L \subseteq \Sigma^*$ λέγεται **$\mathcal{NP}$-δύσκολη** (**$\mathcal{NP}$-hard**) αν και μόνο αν για κάθε γλώσσα $L' \in \mathcal{NP}$, υπάρχει μια πολυωνυμική αναγωγή από την L' στην L.
- **Ορισμός.** Μια γλώσσα $L \subseteq \Sigma^*$ λέγεται **$\mathcal{NP}$-πλήρης** (**$\mathcal{NP}$-complete**) αν και μόνο αν η L είναι $\mathcal{NP}$-δύσκολη και επίσης $L \in \mathcal{NP}$.

11

$\mathcal{NP}$-δύσκολα και $\mathcal{NP}$-πλήρη Προβλήματα

- Η έννοια των $\mathcal{NP}$-δύσκολων και $\mathcal{NP}$-πλήρων προβλημάτων είναι πολύ σημαντική στη θεωρία πολυπλοκότητας.
- Αν καταφέρουμε να λύσουμε ένα $\mathcal{NP}$-δύσκολο ή ένα $\mathcal{NP}$-πλήρες πρόβλημα σε **πολυωνυμικό χρόνο**, τότε μπορούμε να λύσουμε **όλα** τα προβλήματα της κλάσης $\mathcal{NP}$ σε πολυωνυμικό χρόνο. Τότε $\mathcal{P} = \mathcal{NP}$!!!

12

$\mathcal{NP}$-πλήρη Προβλήματα

- **Θεώρημα.** Έστω L μια $\mathcal{NP}$-πλήρης γλώσσα. Τότε $\mathcal{P} = \mathcal{NP}$ αν και μόνο αν $L \in \mathcal{P}$.
- **Απόδειξη.** **(Μόνο αν)** Υποθέστε ότι $\mathcal{P} = \mathcal{NP}$. Επειδή $L \in \mathcal{NP}$, έχουμε και $L \in \mathcal{P}$.

13

Απόδειξη

(Αν)

- Υποθέστε ότι L είναι μια $\mathcal{NP}$-πλήρης γλώσσα και $L \in \mathcal{P}$. Δηλ. η L αποφασίζεται από μια ντετερμινιστική μηχανή Turing M_1 σε πολυωνυμικό χρόνο p_1.
- Έστω $L' \in \mathcal{NP}$. Τότε υπάρχει μια πολυωνυμική αναγωγή τ από την L' στην L. Υποθέστε ότι η τ υπολογίζεται από μια μηχανή Turing M_2 σε πολυωνυμικό χρόνο p_2. Τότε η L' αποφασίζεται από τη ντετερμινιστική μηχανή $M_2 M_1$.
- Η μηχανή $M_2 M_1$ με είσοδο x τερματίζει πάντα σε χρόνο το πολύ

$$p_2(|x|) + p_1(p_2(|x|)).$$

14

Το Πρώτο $\mathcal{NP}$-πλήρες Πρόβλημα

- **Θεώρημα.** **(Cook, 1971)** Το πρόβλημα SATISFIABILITY είναι $\mathcal{NP}$-πλήρες.

15

Το Πρώτο $\mathcal{NP}$-πλήρες Πρόβλημα

- Για την συνεισφορά του στη θεωρία πολυπλοκότητας ο Steve Cook (University of Toronto) βραβεύτηκε με το Turing award το 1982 (το Turing award είναι ουσιαστικά το βραβείο Nobel της Επιστήμης Υπολογιστών).
- **Citation.** For his advancement of our understanding of the complexity of computation in a significant and profound way. His seminal paper, "The Complexity of Theorem Proving Procedures," presented at the 1971 ACM SIGACT Symposium on the Theory of Computing, laid the foundations for the theory of NP-Completeness. The ensuing exploration of the boundaries and nature of NP-complete class of problems has been one of the most active and important research activities in computer science for the last decade.

16

Απόδειξη

- Έχουμε ήδη αποδείξει ότι το πρόβλημα SATISFIABILITY ανήκει στην κλάση $\mathcal{NP}$.
- Επίσης το πρόβλημα SATISFIABILITY είναι $\mathcal{NP}$-δύσκολο όπως προκύπτει από τα δύο παρακάτω λήμματα:
- **Λήμμα 1.** Το πρόβλημα BOUNDED TILING είναι $\mathcal{NP}$-πλήρες.
- **Λήμμα 2.** Υπάρχει πολυωνυμική αναγωγή από το πρόβλημα BOUNDED TILING στο πρόβλημα SATISFIABILITY.

Το Πρόβλημα Φραγμένης Πλακόστρωσης

- Ένα **σύστημα πλακόστρωσης** είναι μια τετράδα $\mathcal{D} = (D, H, V)$, όπου D είναι ένα πεπερασμένο σύνολο (που δείχνει τι πλακάκια έχουμε στη διάθεση μας) και $H, V \subseteq D \times D$. Μας δίνεται επίσης ένας θετικός ακέραιος s και μια συνάρτηση
$$f_0 : \ \{0, \ldots, s-1\} \to D$$
- Μια $s \times s$ **πλακόστρωση** με το σύστημα $\mathcal{D}$ είναι μια συνάρτηση
$$f : \ \{0, \ldots, s-1\} \times \{0, \ldots, s-1\} \to D$$
τέτοια ώστε να ισχύουν τα παρακάτω:
$$f(m, 0) = f_0(m) \text{ για κάθε } m < s$$
$$(f(n, m), f(n+1, m)) \in H \text{ για κάθε } m < s-1, \ n < s$$
$$(f(n, m), f(n, m+1)) \in V \text{ για κάθε } m < s, \ n < s-1$$

Το Πρόβλημα Φραγμένης Πλακόστρωσης

- BOUNDED TILING. Έστω ένα σύστημα πλακόστρωσης $\mathcal{D} = (D, H, V)$, ένας θετικός ακέραιος s, και μια συνάρτηση $f_0 : \ \{0, \ldots, s-1\} \to D$ που παριστάνεται από την ακολουθία τιμών $(f_0(0), \ldots, f_0(s-1))$. Υπάρχει μια $s \times s$ πλακόστρωση με το σύστημα $\mathcal{D}$;

Το Πρόβλημα bounded tiling Είναι $\mathcal{NP}$-πλήρες

- Το πρόβλημα BOUNDED TILING ανήκει στην κλάση $\mathcal{NP}$. Μπορούμε να "μαντέψουμε" τις s^2 τιμές της συνάρτησης f που περιγράφει την πλακόστρωση. Το μήκος αυτού του πιστοποιητικού είναι $s^2 \cdot k$ όπου k είναι ο αριθμός των συμβόλων που χρειάζονται για να παραστήσουμε ένα πλακάκι.
- **Σημαντικό:** Ο αριθμός s είναι μικρότερος ή ίσος από το μήκος της εισόδου επειδή η είσοδος περιέχει την ακολουθία των τιμών της f.
- Οπότε το "μάντεμα" μπορεί να γίνει σε πολυωνυμικό χρόνο.

Το Πρόβλημα bounded tiling Είναι $\mathcal{NP}$-πλήρες

- Μπορούμε επίσης να δείξουμε ότι για κάθε γλώσσα $L \in \mathcal{NP}$, υπάρχει μια πολυωνυμική αναγωγή τ από τη γλώσσα L στο πρόβλημα BOUNDED TILING.
- Έστω μια μη ντετερμινιστική μηχανή Turing M που αποφασίζει την L σε πολυωνυμικό χρόνο. Για κάθε $x \in \Sigma^*$, $\tau(x)$ είναι η αναπαράσταση ενός προβλήματος φραγμένης πλακόστρωσης ώστε: υπάρχει πλακόστρωση αν και μόνο αν $x \in L$.
- Οι λεπτομέρειες της αναγωγής είναι παρόμοιες με αυτές στην απόδειξη του ότι το (μη φραγμένο) πρόβλημα πλακόστρωσης δεν είναι επιλύσιμο.

Από το bounded tiling στο satisfiability

- Έστω μια περίπτωση του προβλήματος BOUNDED TILING δηλ. ένα σύστημα πλακόστρωσης $\mathcal{D} = (D, H, V)$, πλευρά s και κάτω γραμμή f_0. Έστω επίσης ότι $D = \{d_1, \ldots, d_k\}$.
- Θα κατασκευάσουμε ένα τύπο του Boole $\tau(\mathcal{D}, s, f_0)$ τέτοιο ώστε υπάρχει μια $s \times s$ πλακόστρωση με το σύστημα $\mathcal{D}$ αν και μόνο αν ο τύπος $\tau(\mathcal{D}, s, f_0)$ είναι ικανοποιήσιμος.

Από το bounded tiling στο satisfiability

- **Μεταβλητές.** Οι μεταβλητές του $\tau(\mathcal{D}, s, f_0)$ είναι
$$x_{mnd} \text{ για κάθε } 0 \le m, n < s \text{ και } d \in D.$$
- Θα φροντίσουμε κάθε μεταβλητή x_{mnd} να είναι $\top$ αν και μόνο αν $f(m, n) = d$.

Από το bounded tiling στο satisfiability

- Θα έχουμε τα ακόλουθα **clauses:**
  - Για κάθε $0 \le m, n < s$, το clause
  $$x_{mnd_1} \vee x_{mnd_2} \vee \cdots \vee x_{mnd_k}.$$
  Δηλ. καθένα από τα $s \times s$ τετράγωνα μπορεί να περιέχει τουλάχιστον ένα πλακάκι.
  - Για κάθε $0 \le m, n < s$ και $d, d' \in D$ με $d \neq d'$, το clause
  $$\bar{x_{mnd}} \vee \bar{x_{mnd'}}.$$
  Δηλ. καθένα από τα $s \times s$ τετράγωνα μπορεί να περιέχει το πολύ ένα πλακάκι.

Από το bounded tiling στο satisfiability

- Για κάθε $i = 0, \ldots, s-1$, το clause

$$x_{i0f_0(i)}.$$

Τα παραπάνω clauses περιορίζουν τις δυνατές πλακοστρώσεις ώστε να ισχύει $f(i,0) = f_0(i)$ (δηλ. η πρώτη γραμμή της πλακόστρωσης είναι όπως περιγράφει η συνάρτηση f_0).

- Για κάθε $0 \leq n < s$, $0 \leq m < s-1$ και $(d, d') \in D^2 \setminus H$, το clause

$$\bar{x}_{mnd} \vee \bar{x}_{m+1,n,d'}.$$

Δηλ. δύο πλακάκια που δεν "καλύπτονται" από τη σχέση H δεν μπορεί να είναι στην ίδια γραμμή και το ένα δίπλα στο άλλο.

Από το bounded tiling στο satisfiability

- Για κάθε $0 \leq n < s-1$, $0 \leq m < s$ και $(d, d') \in D^2 \setminus V$, το clause

$$\bar{x}_{mnd} \vee \bar{x}_{m,n+1,d'}.$$

Δηλ. δύο πλακάκια που δεν "καλύπτονται" από τη σχέση V δεν μπορεί να είναι στην ίδια στήλη και το ένα δίπλα στο άλλο.

- Η παραπάνω αναγωγή μπορεί να υπολογιστεί σε πολυωνυμικό χρόνο.
- Επίσης, είναι εύκολο να δούμε ότι υπάρχει μια $s \times s$ πλακόστρωση με το σύστημα $\mathcal{D}$ αν και μόνο αν ο τύπος $\tau(\mathcal{D}, s, f_0)$ είναι ικανοποιήσιμος.

Από το bounded tiling στο satisfiability

(Αν)

- Υποθέτουμε ότι υπάρχει μια $s \times s$ πλακόστρωση f με το σύστημα $\mathcal{D}$. Τότε ο τύπος $\tau(\mathcal{D}, s, f_0)$ είναι ικανοποιείται από την ανάθεση τιμών αλήθειας T που είναι τέτοια ώστε: $T(x_{mnd}) = \top$ αν και μόνο αν $f(m,n) = d$.

(Μόνο Αν)

- Υποθέστε ότι ο τύπος $\tau(\mathcal{D}, s, f_0)$ ικανοποιείται από την ανάθεση τιμών αλήθειας T. Η ανάθεση T ουσιαστικά περιγράφει μια πλακόστρωση.

Μεθοδολογία

- Για να αποδείξουμε ότι ένα πρόβλημα είναι $\mathcal{NP}$-δύσκολο, κάνουμε το εξής:
  - Δίνουμε μια πολυωνυμική αναγωγή από ένα πρόβλημα που ήδη γνωρίζουμε ότι είναι $\mathcal{NP}$-δύσκολο, στο πρόβλημα μας.
- Πρέπει λοιπόν να ξέρουμε πολλά $\mathcal{NP}$-δύσκολα προβλήματα, και να έχουμε παραγωγική φαντασία!

Μεθοδολογία

- Για να αποδείξουμε ότι ένα πρόβλημα είναι $\mathcal{NP}$-πλήρες, κάνουμε τα εξής:
  - Αποδεικνύουμε ότι ανήκει στην κλάση $\mathcal{NP}$, παρουσιάζοντας μια μη ντετερμινιστική μηχανή Turing που λύνει το πρόβλημα.
  - Δίνουμε μια πολυωνυμική αναγωγή από ένα πρόβλημα που ήδη γνωρίζουμε ότι είναι $\mathcal{NP}$-πλήρες, στο πρόβλημα μας.

Μεθοδολογία

- Η σημασία των $\mathcal{NP}$-πλήρων προβλημάτων αναπτύχθηκε από τον Richard Karp (University of Washington) το 1972. Όπως και ο Cook, ο Karp βραβεύτηκε επίσης με το Turing award το 1985.
- **Citation.** For his continuing contributions to the theory of algorithms including the development of efficient algorithms for network flow and other combinatorial optimization problems, **the identification of polynomial-time computability with the intuitive notion of algorithmic efficiency, and, most notably, contributions to the theory of NP-completeness. Karp introduced the now standard methodology for proving problems to be NP-complete which has led to the identification of many theoretical and practical problems as being computationally difficult.**

Το Πρόβλημα 3-satisfiability

- 3-SATISFIABILITY: Έστω τύπος του Boole F με την ιδιότητα ότι κάθε clause έχει το πολύ 3 literals. Είναι ο F ικανοποιήσιμος;
- Το πρόβλημα 3-SATISFIABILITY είναι μια **μερική περίπτωση** του προβλήματος SATISFIABILITY.
- **Θεώρημα.** Το πρόβλημα 3-SATISFIABILITY είναι $\mathcal{NP}$-πλήρες.

Απόδειξη

- Το πρόβλημα 3-SATISFIABILITY ανήκει στην κλάση $\mathcal{NP}$.
- Για να δείξουμε ότι είναι $\mathcal{NP}$-δύσκολο θα παρουσιάσουμε μια αναγωγή από το πρόβλημα SATISFIABILITY.
- Για ένα τύπο του Boole F, η αναγωγή τ δίνει ένα **ισοδύναμο** τύπο $\tau(F)$ που έχει την ιδιότητα ότι κάθε clause έχει το πολύ 3 literals.

Απόδειξη

- Ο τύπος $\tau(F)$ κατασκευάζεται ως εξής:
  - Για κάθε clause
    $$C \;=\; l_1 \vee l_2 \vee \cdots \vee l_k$$
    του F που έχει $k > 3$ literals, αντικαθιστούμε τον C με το ακόλουθο **ισοδύναμο** σύνολο από clauses:
    $$l_1 \vee l_2 \vee y_1, \quad \bar{y_1} \vee l_3 \vee y_2,$$
    $$\bar{y_2} \vee l_3 \vee y_3, \ldots, \bar{y_{k-4}} \vee l_{k-2} \vee y_{k-3},$$
    $$\bar{y_{k-3}} \vee l_{k-1} \vee l_k$$
  - Όλα τα clauses με $k \leq 3$ literals μένουν όπως είναι.
- Η παραπάνω αναγωγή μπορεί να υπολογιστεί σε πολυωνυμικό χρόνο.

Απόδειξη

- Έστω ότι ο F ικανοποιείται από την ανάθεση τιμών αλήθειας T. Τότε η T μπορεί να επεκταθεί σε μια ανάθεση τιμών αλήθειας που ικανοποιεί τον $\tau(F)$ ως εξής.
- Για κάθε clause
  $$C \;=\; l_1 \vee l_2 \vee \cdots \vee l_k$$
  του F με $k > 3$ literals, ας είναι j ο μικρότερος δείκτης τέτοιος ώστε $T(l_j) = \top$. Τότε ορίζουμε τιμές αλήθειας για τις νέες μεταβλητές $y_1, y_2, \ldots, y_{k-3}$ ως εξής:
  - $T(y_i) = \top$ αν $i \leq j - 2$
  - $T(y_i) = \bot$ αν $i > j - 2$
- Η παραπάνω επέκταση της T ικανοποιεί τον τύπο $\tau(F)$.

Απόδειξη

- Έστω ότι ο $\tau(F)$ ικανοποιείται από την ανάθεση τιμών αλήθειας T. Τότε η T επίσης ικανοποιεί τον τύπο F διότι:
  - Προφανώς ικανοποιεί τα clauses του F με $k \leq 3$ literals.
  - Αν η T αντιστοιχεί όλα τα $k > 3$ literals ενός clause στο $\bot$, τότε οι μεταβλητές y_i δεν μπορούν από μόνες τους να ικανοποιήσουν τα υπόλοιπα clauses.
- Εχουμε αποδείξει λοιπόν ότι ο F είναι ικανοποιήσιμος αν και μόνο αν ο $\tau(F)$ είναι ικανοποιήσιμος.

Το Πρόβλημα max sat

- MAX SAT: Έστω ένα σύνολο F αποτελούμενο από m clauses και ακέραιος αριθμός $K \leq m$. Υπάρχει ανάθεση τιμών αληθείας που ικανοποιεί τουλάχιστον K από τα clauses του F;
- Το πρόβλημα MAX SAT είναι μια **γενίκευση** του προβλήματος SATISFIABILITY.
- **Θεώρημα.** Το πρόβλημα MAX SAT είναι $\mathcal{NP}$-πλήρες.

Απόδειξη

- Είναι φανερό ότι το πρόβλημα MAX SAT ανήκει στην κλάση $\mathcal{NP}$.
- Για να δείξουμε ότι είναι $\mathcal{NP}$-δύσκολο θα παρουσιάσουμε μια αναγωγή από το πρόβλημα SATISFIABILITY.
- Έστω μια περίπτωση του προβλήματος SATISFIABILITY δηλ. ένας τύπος F που αποτελείται από m clauses. Η αναγωγή κατασκευάζει το $\tau(F)$ που είναι ένα ζεύγος (F, K) τέτοιο ώστε $K = m$ (δηλ. μια περίπτωση του προβλήματος MAX SAT).

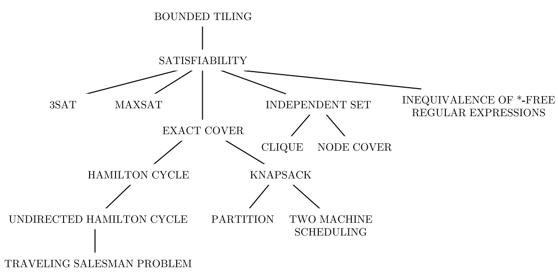
Το Πρόβλημα independent set

- INDEPENDENT SET: Έστω μη κατευθυνόμενος γράφος $G \subseteq V \times V$ και ακέραιος $K \geq 2$. Υπάρχει $C \subseteq V$ με $|C| \geq K$ τέτοιο ώστε για όλες τις κορυφές $v_i, v_j \in C$, δεν υπάρχει ακμή που να συνδέει τις v_i και v_j;
- **Θεώρημα.** Το πρόβλημα INDEPENDENT SET είναι $\mathcal{NP}$-πλήρες.

Απόδειξη

- Είναι φανερό ότι το πρόβλημα INDEPENDENT SET ανήκει στην κλάση $\mathcal{NP}$.
- Έχουμε ήδη παρουσιάσει μια πολυωνυμική αναγωγή από το πρόβλημα 3-SATISFIABILITY στο INDEPENDENT SET.
- Αρα το πρόβλημα INDEPENDENT SET είναι $\mathcal{NP}$-πλήρες.

Αλλα $\mathcal{NP}$-πλήρη Προβλήματα



Η Ζωή με την $\mathcal{NP}$-πληρότητα

- Αφού δείξουμε ότι ένα πρόβλημα είναι $\mathcal{NP}$-πλήρες, δεν σημαίνει ότι θα σηκώσουμε τα χέρια ψηλά!
- Μπορούμε να έχουμε ικανοποιητικές και αποδοτικές λύσεις του προβλήματος χρησιμοποιώντας τις ακόλουθες τεχνικές:
  - Ανάλυση ειδικών περιπτώσεων.
  - Χρήση προσεγγιστικών αλγόριθμων
  - Χρήση backtracking ή branch-and-bound.
  - Χρήση αλγόριθμων local search (simulated annealing, genetic algorithms κλπ.)
- Η μελέτη των παραπάνω τεχνικών θα ταίριαζε σε ένα δεύτερο μάθημα Θεωρίας Αλγορίθμων.

Μελέτη

- Κεφάλαια 7.1-7.3 του βιβλίου
  Harry R. Lewis και Χρίστος Χ. Παπαδημητρίου, Elements of the Theory of Computation, 2nd edition, Prentice Hall, 1998.