Αλγόριθμοι και Πολυπλοκότητα

Εθνικό Μετσόβιο Πολυτεχνείο
7ο εξάμηνο
Σ.Η.Μ.Μ.Υ. & Σ.Ε.Μ.Φ.Ε.

http://www.corelab.ece.ntua.gr/courses/

10η ενότητα: Το Θεώρημα του Cook, Μετασχηματισμοί Προβλημάτων, NP-complete Προβλήματα, Κλάσεις Πολυπλοκότητας

Διδάσκοντες: Στάθης Ζάχος – Άρης Παγουρτζής

Hamilton Circuit $\leq^p$ TSP (i)

Μας δίνεται ένας γράφος $G(V,E)$. Θέλουμε να κατασκευάσουμε έναν πλήρη γράφο $G'(V',E')$ με βάρη $d(u,v), \forall (u,v) \in E'$ και έναν θετικό ακέραιο B, έτσι ώστε ο γράφος $G(V,E)$ να έχει κύκλο Hamilton αν και μόνο αν ο $G'(V',E')$ έχει tour με βάρος $\leq B$. Η κατασκευή γίνεται ως εξής:

- Σαν γράφο $G'(V',E')$, παίρνουμε τον γράφο $G(V,E)$, προσθέτοντας όλες τις ακμές που υπολείπονται για να γίνει πλήρης. Βάζουμε βάρη στις ακμές του G' ως εξής:

$$d(u,v) = \begin{cases} 1, & (u,v) \in G \\ 2, & (u,v) \notin G \end{cases}$$

- Τέλος, παίρνουμε $B = |V'| = |V|$. Η κατασκευή έχει τελειώσει και μπορεί, προφανώς να γίνει σε πολυωνυμικό χρόνο.

Hamilton Circuit $\leq^p$ TSP (ii)

Έστω ότι ο $G(V,E)$ έχει κύκλο Hamilton. Τότε παίρνοντας αυτόν τον κύκλο σαν tour στον γράφο G', προφανώς περνά μία ακριβώς φορά από κάθε κόμβο και έχει συνολικό βάρος $B = |V'| = |V|$ εφόσον κάθε πλευρά έχει βάρος 1 (αφού ανήκε στον G).

Αντίστροφα, έστω ότι ο γράφος G' έχει κάποιο tour με συνολικό βάρος $\leq B = |V'| = |V|$. Αφού όμως το G' έχει $|V'|$ κόμβους, το tour θα περνά από $|V'|$ πλευρές και συνεπώς το συνολικό βάρος θα είναι ακριβώς $B = |V'|$. Αυτό όμως μπορεί να συμβεί μόνο όταν κάθε μία από τις $|V'|$ πλευρές έχει βάρος 1. Άρα όλες αυτές οι πλευρές ανήκουν στον G και συνεπώς ο G έχει κύκλο Hamilton (είναι το tour του G'). $\square$

Ορισμός: CNF formula

Ορισμός 13.6.2. Αν $x_1, x_2, \ldots$ είναι προτασιακές μεταβλητές, ονομάζουμε:

- literals: όρους όπως $x_1, x_3, \neg x_1, \neg x_5$
- clauses: διαζεύξεις (disjunctions) από litetals, π.χ. $(x_1 \vee \neg x_2 \vee \neg x_5)$
- Conjuctive Normal Form (CNF) την ακόλουθη μορφή που μπορεί να έχει η boolean formula:

$$(clause_1 \wedge clause_2 \wedge \ldots \wedge clause_m)$$

Θεώρημα 13.6.3. *Κάθε λογική έκφραση είναι ισοδύναμη με μία σε CNF.*

Ορισμός: SAT

Ορισμός 13.6.4. Ικανοποιήσιμη (satisfiable) ονομάζεται μία boolean formula όταν υπάρχει απονομή αλήθειας (truth assignment) στις προτασιακές μεταβλητές που την αποτελούν, έτσι ώστε η έκφραση να παίρνει την τιμή True (T).

Το πρόβλημα SAT
Δεδομένα: Μία boolean formula σε CNF.
Ερώτηση: Είναι η boolean formula ικανοποιήσιμη;

Παράδειγμα 13.6.5. Η φόρμουλα $(x_1 \vee \neg x_2) \wedge (\neg x_1 \vee x_2)$ είναι ικανοποιήσιμη. Μία απονομή αλήθειας που την ικανοποιεί είναι η $(x_1, x_2) = (T, T)$. Η φόρμουλα, $(x_1 \vee x_2) \wedge (x_1 \vee \neg x_2) \wedge \neg x_1$ δεν είναι ικανοποιήσιμη.

Θεώρημα Cook

Το πρόβλημα SAT είναι NP-complete

Απόδειξη. Κατ' αρχάς πρέπει να αποδείξουμε ότι $SAT \in NP$. Αυτό όμως είναι εύκολο. Ένας μη-ντετερμινιστικός αλγόριθμος που λύνει το SAT είναι ο εξής:

1. μάντεψε μία απονομή αλήθειας

2. έλεγξε αν η έκφραση αποτιμάται σε True

Ο έλεγχος μπορεί να γίνει σε γραμμικό χρόνο ως προς το μέγεθος της φόρμουλας και συνεπώς $SAT \in NP$.

Θεώρημα Cook: Απόδειξη NP-hardness

Ενας οποιοσδήποτε υπολογισμός μιας NDTM πολυωνυμικού χρόνου M μπορεί να αναπαρασταθεί από μια boolean formula Φ πολυωνυμικού μήκους έτσι ώστε:

υπάρχει υπολογισμός της M που αποδέχεται το x αν και μόνο αν υπάρχει απονομή αλήθειας που ικανοποιεί την boolean formula Φ(x)

Θεώρημα Cook: Κατασκευή *Φ(x)*

Λογικές μεταβλητές της Φ(x)

- $Q[i,k]$, $0 \le i \le p(n)$, $0 \le k \le r$ κωδικοποιούν την κατάσταση q_k στην οποία βρίσκεται η T.M. τη χρονική στιγμή i . Δηλαδή η μεταβλητή $Q[i,k]$ θα είναι True αν και μόνο αν τη χρονική στιγμή i η T.M. βρίσκεται στην κατάσταση q_k. Ο αριθμός αυτών των μεταβλητών είναι $(r+1)\cdot(p(n)+1)$.

- $H[i,j]$, $0 \le i \le p(n)$, $-p(n) \le j \le p(n)$ κωδικοποιούν τη θέση j στην οποία βρίσκεται η κεφαλή τη χρονική στιγμή i . Δηλαδή η μεταβλητή $H[i,j]$ θα είναι True ανν τη χρονική στιγμή i η κεφαλή βρίσκεται στη θέση (κυψέλη) j. Ο αριθμός αυτών των μεταβλητών είναι $(p(n)+1)\cdot(2p(n)+1)$.

- $S[i,j,l]$, $0 \le i \le p(n)$, $-p(n) \le j \le p(n)$, $0 \le l \le v$ κωδικοποιούν το σύμβολο s_l, το οποίο περιέχεται στη θέση j τη χρονική στιγμή i. Δηλαδή η μεταβλητή $S[i,j,l]$ θα είναι True ανν τη χρονική στιγμή i, στη θέση j περιέχεται το σύμβολο s_l. Ο αριθμός αυτών των μεταβλητών είναι $(p(n)+1)\cdot(2p(n)+1)\cdot(v+1)$.

Θεώρημα Cook: Κατασκευή Φ(x)

Clauses της Φ(x): group G_1

G_1: Το group αυτό θα εκφράζει το γεγονός, ότι σε μία δεδομένη χρονική στιγμή η T.M. θα βρίσκεται ακριβώς σε μία κατάσταση. Δηλαδή μία δεδομένη χρονική στιγμή i, θα είναι True ακριβώς μία από τις μεταβλητές $Q[i,k], 0 \leq k \leq r$. Αυτό θα πρέπει να ισχύει για κάθε χρονική στιγμή. Το G_1 λοιπόν, θα λέει ότι μία δεδομένη χρονική στιγμή i, κάποια από τις μεταβλητές $Q[i,k]$ είναι True ενώ η σύζευξη οποιωνδήποτε δύο μεταβλητών απ' αυτές είναι False. Αυτό κωδικοποιείται εύκολα, όπως μπορεί να επαληθεύσει κανείς ως εξής:

$$\wedge_{i=0}^{p(n)}((\vee_{j=0}^{r} Q[i,j]) \wedge (\wedge_{j=0}^{r} \wedge_{k=0}^{j-1} (\neg Q[i,j] \vee \neg Q[i,k]))) \qquad (13.1)$$

Ο πρώτος όρος της παραπάνω έκφρασης εξασφαλίζει ότι σε μία δεδομένη χρονική στιγμή η μηχανή βρίσκεται τουλάχιστον σε μία κατάσταση και ο δεύτερος όρος ότι βρίσκεται το πολύ σε μία κατάσταση. Ο ολικός αριθμός των literals που περιέχονται σ' αυτά τα clauses είναι:

$$(p(n)+1)\cdot[(r+1)+\frac{r(r+1)}{2}\cdot 2] = (p(n)+1)\cdot(r+1)^2 = O(p(n))$$

Θεώρημα Cook: Κατασκευή *Φ(x)*

Clauses της Φ(x): group G_2

G_2: Το group αυτό θα εκφράζει το γεγονός ότι σε μία δεδομένη χρονική στιγμή, η κεφαλή θα βρίσκεται σε μία ακριβώς θέση. Δηλαδή μία δεδομένη χρονική στιγμή i, θα είναι True ακριβώς μία από τις μεταβλητές $H[i,j], -p(n) \leq j \leq p(n)$ και αυτό θα πρέπει να ισχύει για κάθε χρονική στιγμή. Εντελώς ανάλογα λοιπόν, με το group G_1, το G_2 κωδικοποιείται ως εξής:

$$\wedge_{i=0}^{p(n)}((\vee_{j=-p(n)}^{p(n)} H[i,j]) \wedge (\wedge_{j=-p(n)}^{p(n)} \wedge_{k=-p(n)}^{j-1} (\neg H[i,j] \vee \neg H[i,k]))) \quad (13.2)$$

Ο ολικός αριθμός των literals που περιέχονται σ' αυτά τα clauses είναι,

$$(p(n)+1) \cdot (2p(n)+1)^2) = O(p^3(n))$$

Θεώρημα Cook: Κατασκευή Φ(x)

Clauses της Φ(x): group G_3

G_3: Το group αυτό θα εκφράζει το γεγονός ότι σε μία δεδομένη χρονική στιγμή, σε κάθε θέση στην ταινία περιέχεται ακριβώς ένα σύμβολο. Δηλαδή μία δεδομένη χρονική στιγμή i, για μία συγκεκριμένη θέση j, θα είναι True ακριβώς μία από τις μεταβλητές $S[i,j,l], 0 \leq l \leq v$ και αυτό θα πρέπει να ισχύει για κάθε χρονική στιγμή και για κάθε θέση. Εντελώς ανάλογα λοιπόν, με τα groups G_1 και G_2, το G_3 θα αποτελείται από τα εξής clauses:

$$\wedge_{i=0}^{p(n)} \wedge_{j=-p(n)}^{p(n)} ((\vee_{l=0}^{v} S[i,j,l]) \wedge (\wedge_{l=0}^{v} \wedge_{k=0}^{l-1} (\neg S[i,j,l] \vee \neg S[i,j,k]))) \quad (13.3)$$

Συνολικά ο αριθμός των literals που περιέχονται στα clauses του G_3 είναι:

$$(p(n)+1) \cdot (2p(n)+1) \cdot (v+1)^2 = O(p^2(n))$$

Θεώρημα Cook: Κατασκευή $\Phi(x)$

Clauses της $\Phi(x)$: groups G_4, G_5

- G_4: Το group αυτό θα δηλώνει ότι τη χρονική στιγμή 0 η T.M. βρίσκεται στο αρχικό configuration. Δηλαδή: Η κατάσταση στην οποία βρίσκεται η T.M. είναι q_0: $Q[0,0]$, η κεφαλή βρίσκεται στη θέση 1: $H[0,1]$, ότι στις θέσεις 1 έως n είναι γραμμένο το input x: $S[0,1,l_1] \wedge S[0,2,l_2] \wedge \ldots \wedge S[0,n,l_n]$, αν υποθέσουμε ότι $x = S_{l_1}, S_{l_2}, \ldots S_{l_n}$ και από τη θέση $n+1$ έως τη θέση $p(n)$ έχουμε κενά ($blanks$):

  $$S[0,n+1,0] \wedge S[0,n+2,0] \wedge \ldots \wedge S[0,p(n),0]$$

  Ακόμα, οι θέσεις $-p(n)$ έως 0 τη χρονική στιγμή 0 περιέχουν τον κενό χαρακτήρα ($S[0,0,0]$, κ.τ.λ.).

  Η σύζευξη όλων αυτών των clauses αποτελεί το group G_4. Ο συνολικός αριθμός των literals είναι $2p(n)+3 = O(p(n))$.

- G_5: Το group αυτό αποτελείται μόνο από ένα clause το οποίο έχει ένα literal που δηλώνει, ότι τη χρονική στιγμή $p(n)$, η κατάσταση στην οποία βρίσκεται η T.M. είναι η $q_1 = q_Y$. Δηλαδή:

  $$Q[p(n),1]$$

Θεώρημα Cook: Κατασκευή *Φ(x)*

Clauses της Φ(x): group G_6 (i)

G_6: Το τελευταίο αυτό group θα δηλώνει πως το configuration της T.M. τη χρονική στιγμή $i+1$ προκύπτει από την εφαρμογή της συνάρτησης μετάβασης (transition function) δ, στο configuration της χρονικής στιγμής i. Κατ' αρχάς το G_6 θα δηλώνει πως το περιεχόμενο της θέσης j, δεν μπορεί να αλλάξει τη χρονική στιγμή $i+1$, αν τη χρονική στιγμή i η κεφαλή δεν βρισκόταν στη θέση j. Δηλαδή:

$$(\neg H[i,j] \wedge S[i,j,l]) \rightarrow S[i+1,j,l],$$

ή αλλιώς:

$$\begin{cases} (H[i,j] \vee \neg S[i,j,l]) \vee S[i+1,j,l] \\ 0 \leq i \leq p(n), -p(n) \leq j \leq p(n), 0 \leq l \leq v \end{cases}$$

Θεώρημα Cook: Κατασκευή *Φ(x)*

Clauses της Φ(x): group G_6 (ii)

Έχουμε δηλαδή $3(2p(n)+1) \cdot p(n) \cdot (v+1)$ literals. Επιπλέον το G_6 θα δηλώνει πως οι αλλαγές που γίνονται στο configuration τη χρονική στιγμή $i+1$ προκύπτουν από την εφαρμογή της συνάρτησης μετάβασης δ, στο configuration της χρονικής στιγμής i. Δηλαδή:

$$(H[i,j] \wedge Q[i,k] \wedge S[i,j,l]) \rightarrow$$
$$\bigvee_{m=1}^{|\delta(q_k,s_l)|} (H[i+1, j+\Delta_m] \wedge Q[i+1, k'_m] \wedge S[i+1, j, l'_m]) \quad (13.4)$$

Θεώρημα Cook: Κατασκευή *Φ(x)*

Clauses της Φ(x): group G_6 (iii)

Αυτό σημαίνει πως αν τη χρονική στιγμή i η T.M. βρίσκεται στην κατάσταση q_k με την κεφαλή στη θέση j να διαβάζει το σύμβολο s_l, τότε τη χρονική στιγμή $i+1$, το περιεχόμενο της θέσης j, η κατάσταση και η θέση της κεφαλής θα πρέπει να ικανοποιούν την μη ντετερμινιστική συνάρτηση μετάβασης. δηλαδή, αν $q_k \in Q\{q_Y, q_N\}$ τότε οι τιμές των Δ_m, k'_m, l'_m είναι τέτοιες ώστε να ισχύει:

$$(q_{k'_m}, s_{l'_m}, \Delta_m) \in \delta(q_k, s_l), \text{ όπου} \Delta_m \in \{-1, 0, 1\}.$$

Την παραπάνω έκφραση μπορούμε να την φέρουμε σε CNF κάνοντας μερικές πράξεις και τελικά ο ολικός αριθμός των literals για δεδομένα i, j, k, l είναι $(c+3)3^c$, όπου $c = |\delta(q_k, s_l)|$. Αν $q_k \in \{q_Y, q_N\}$ τότε $\Delta = 0, k' = k, l' = l$ (δηλαδή αν η T.M. έχει ήδη αποδεχθεί ή απορρίψει το x πριν τη στιγμή $p(n)$, διατηρεί το configuration όπως είναι μέχρι τη στιγμή $p(n)$). Ο ολικός αριθμός των literals που περιέχονται στα clauses του G_6 είναι: $O(p^2(n))$.

Θεώρημα Cook: Κατασκευή *Φ(x)*

Τελική Φόρμουλα

$$\Phi(x) = G_1 \wedge G_2 \wedge G_3 \wedge G_4 \wedge G_5 \wedge G_6$$

Το μήκος της είναι: $O(p^3(n))$ (καθοριστικό είναι το μήκος του G_2).

Άρα το SAT είναι NP-complete.

Μετασχηματισμοί Προβλημάτων

Παράδειγμα: Έστω το εξής πρόβλημα:
Δεδομένα: Μια σκακιέρα 3×3 με μαύρα (X) και λευκά (O) αλογάκια όπως φαίνεται στο σχήμα 14.1.
Ερώτηση: Πως μπορούμε να ανταλλάξουμε τις θέσεις των λευκών με τα μαύρα αλογάκια χρησιμοποιώντας νόμιμες κινήσεις;

X		X
O		O

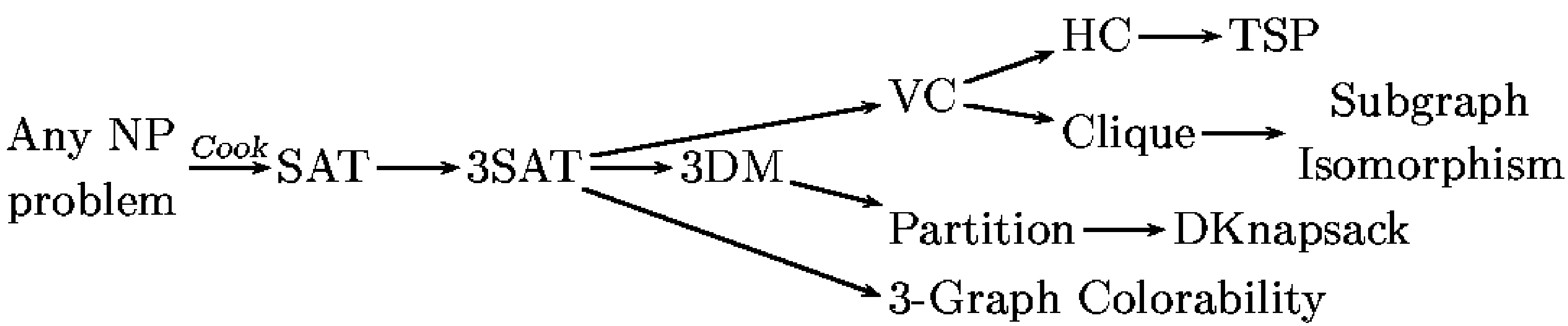
Απόδειξη NP-completeness

Για να δείξουμε ότι ένα πρόβλημα Π είναι NP-complete ακολουθούμε τα παρακάτω βήματα:

1. Δείχνουμε ότι $\Pi \in NP$.
2. Διαλέγουμε ένα γνωστό NP-complete πρόβλημα Π' και κατασκευάζοντας μια συνάρτηση f, το μετασχηματίζουμε στο πρόβλημα Π.
3. Δείχνουμε ότι ο μετασχηματισμός f γίνεται σε πολυωνυμικό χρόνο
4. Αποδεικνύουμε ότι $x \in \Pi' \iff f(x) \in \Pi$.

Αναγωγές μεταξύ NP-complete προβλημάτων

Οι αναγωγές προβλημάτων που θα δούμε στη συνέχεια, έγιναν με τη σειρά που φαίνεται στο σχήμα 14.4 και ιστορικά, οι περισσότερες από αυτές παρουσιάστηκαν από τον *Karp* (1972).



Ορισμοί Προβλημάτων Απόφασης

SAT (SATISFIABILITY)

Δεδομένα: Μία λογική έκφραση (boolean formula) σε κανονική συζευκτική μορφή (CNF).

Ερώτηση: Είναι η λογική έκφραση ικανοποιήσιμη; Δηλαδή, υπάρχει απονομή αλήθειας στις μεταβλητές τις τέτοια ώστε η boolean formula να αποτιμάται σε τιμή True.

3SAT

Δεδομένα: Μια boolean formula σε CNF, κάθε clause της οποίας έχει ακριβώς 3 literals.

Ερώτηση: Είναι η boolean formula ικανοποιήσιμη;

Ορισμοί Προβλημάτων Απόφασης

CLIQUE

Δεδομένα: Ένας γράφος $G(V,E)$ και ένας θετικός ακέραιος $j \leq |V|$.
Ερώτηση: Περιέχει ο γράφος G κλίκα μεγέθους $\geq j$; Δηλαδή υπάρχει $V' \subseteq V$, τέτοιο ώστε: $|V'| \geq j$ και $\forall u,v \in V' : (u,v) \in E$;
Η ερώτηση μπορεί να γίνει ως εξής: Περιέχει ο γράφος G πλήρη υπογράφο με πλήθος κόμβων $\geq j$;

VC (VERTEX COVER)

Δεδομένα: Ένας γράφος $G(V,E)$ και ένας θετικός ακέραιος $k \leq |V|$.
Ερώτηση: Υπάρχει ένα vertex cover όλων των ακμών του E, μεγέθους $\leq k$; Δηλαδή, υπάρχει ένα σύνολο $V' \subseteq V$ τέτοιο ώστε $|V'| \leq k$ και $\forall \{u,v\} \in E : \quad u \in V' \vee v \in V'$;

Ορισμοί Προβλημάτων Απόφασης

3DM (3-DIMENSIONAL MATCHING)
Δεδομένα: Ένα σύνολο $M \subseteq W \times X \times Y$, όπου W, X, Y είναι σύνολα ξένα μεταξύ τους (disjoint) με $|W| = |X| = |Y| = q$.
Ερώτηση: Περιέχει το M ένα ταίριασμα (matching); Δηλαδή, υπάρχει σύνολο $M' \subseteq M$ τέτοιο ώστε $|M'| = q$ και έτσι ώστε 2 οποιαδήποτε στοιχεία του M' να μην έχουν καμία κοινή συντεταγμένη;

GRAPH 3-COLORABILITY
Δεδομένα: Ένας γράφος $G(V, E)$.
Ερώτηση: Μπορούμε να βάψουμε τους κόμβους του γράφου G χρησιμοποιώντας **3 χρώματα** και έτσι ώστε 2 οποιοδήποτε γειτονικοί κόμβοι να έχουν διαφορετικό χρώμα; Δηλαδή, υπάρχει συνάρτηση $f : V \rightarrow \{1, 2, 3\}$ τέτοια ώστε, $\forall (u, v) \in E : f(u) \neq f(v)$;

Ορισμοί Προβλημάτων Απόφασης

HC (Hamilton Circuit)

Δεδομένα: Ένας γράφος $G(V, E)$.

Ερώτηση: Έχει ο γράφος κύκλο Hamilton; Δηλαδή, υπάρχει μία διάταξη των κόμβων του γράφου G, $< v_1, v_2, \ldots, v_n >, n = |V|$, τέτοια ώστε

$$(v_i, v_{i+1}) \in E, 1 \leq i \leq n-1, (v_n, v_1) \in E;$$

TSP (TRAVELING SALESMAN PROBLEM)

Δεδομένα: Δίνεται ένας πλήρης γράφος $G(V, E)$ με βάρη και ένας αριθμός B.

Ερώτηση: Υπάρχει μια κλειστή διαδρομή (tour) που να περνά απ' όλους τους κόμβους του G, $< v_{\pi(1)}, v_{\pi(2)}, \ldots, v_{\pi(m)} >$ έτσι ώστε:

$$\sum w(v_{\pi(i)}, v_{\pi(i+1)}) + w(v_{\pi(m)}, v_{\pi(1)}) \leq B;$$

Ορισμοί Προβλημάτων Απόφασης

SUBGRAPH ISOMORPHISM

Δεδομένα: Δύο γράφοι $G(V_1, E_1)$ και $H(V_2, E_2)$.

Ερώτηση: Έχει ο γράφος G υπογράφο ισομορφικό με τον γράφο H; Δηλαδή, υπάρχουν $V \subseteq V_1, E \subseteq E_1$ τέτοια ώστε $|V| = |V_2|, |E| = |E_2|$ και συνάρτηση $f : V_2 \rightarrow V$, «1-1» και «επί» (bijection) ώστε να ισχύει, $(u, v) \in E_2 \iff (f(u), f(v)) \in E$;

PARTITION

Δεδομένα: Ένα πεπερασμένο σύνολο A με βάρη, $w(a) \in Z^+, \forall a \in A$.

Ερώτηση: Είναι δυνατόν το σύνολο A να μοιραστεί σε δύο ισοβαρή υποσύνολα; Δηλαδή, υπάρχει $A' \subseteq A$ τέτοιο ώστε,

$$\sum_{a \in A'} w(a) = \sum_{a \in (A - A')} w(a);$$

Ορισμοί Προβλημάτων Απόφασης

DKNAPSACK (DISCRETE KNAPSACK)

Δεδομένα: Ένα πεπερασμένο σύνολο U, μια συνάρτηση βάρους $w(u) \in Z^+, \forall u \in U$, μια συνάρτηση κόστους $p(u) \in Z^+, \forall u \in U$ και δύο θετικοί ακέραιοι W, P.

Ερώτηση: Μπορούμε να πάρουμε μερικά αντικείμενα από το σύνολο U και να τα βάλουμε μέσα σε ένα σακίδιο, έτσι ώστε το ολικό βάρος του σακιδίου να είναι $\leq W$ και η ολική του αξία $\geq P$; Δηλαδή υπάρχει $U' \subseteq U$ τέτοιο ώστε,

$$\sum_{u \in U'} w(u) \leq W \text{ και } \sum_{u \in U'} p(u) \geq P;$$

Τι δεν ήξερε ο Karp;

- Linear Programming (LP)
- Primality
- Graph Isomorphism

Το 3-SAT είναι NP-complete

Απόδειξη. Κατ' αρχάς είναι εύκολο να δούμε ότι το 3SAT $\in NP$. Πράγματι, ένας μη-ντετερμινιστικός αλγόριθμος, αφού μαντέψει μια απονομή αλήθειας, μπορεί πάντα να ελέγξει σε πολυωνυμικό χρόνο, αν αυτή ικανοποιεί τη boolean formula που δίνεται (το 3SAT είναι ένα υποπρόβλημα του SAT).

Για να αποδείξουμε ότι το 3SAT είναι NP-complete θα ανάγουμε το SAT σ' αυτό (SAT $\leq_m^p$ 3SAT). Έστω ότι μας δίνεται ένα οποιοδήποτε στιγμιότυπο του SAT δηλαδή ένα σύνολο C από m clauses, $C = \{c_1, c_2, \ldots, c_m\}$ που χρησιμοποιούν μεταβλητές από ένα σύνολο από n μεταβλητές, $U = \{z_1, z_2, \ldots, z_n\}$. Θα κατασκευάσουμε ένα καινούριο σύνολο από clauses C' και ένα καινούργιο σύνολο μεταβλητών V', έτσι ώστε κάθε clause που ανήκε στο C' να αποτελείται από 3 ακριβώς literals. Η κατασκευή γίνεται ως εξής:

SAT $\leq^p$ 3-SAT: κατασκευή 3-SAT φόρμουλας

- Για κάθε clause $c \in C$ της αρχικής φόρμουλας που αποτελείται από 1 literal $c = z$, κατασκευάζουμε τα εξής 4 clauses (φυσικά οι μεταβλητές y_1 και y_2 είναι νέες, δεν περιέχονται στην C):

$$(z \vee y_1 \vee y_2) \wedge (z \vee y_1 \vee \neg y_2) \wedge (z \vee \neg y_1 \vee y_2) \wedge (z \vee \neg y_1 \vee \neg y_2)$$

  Είναι εύκολο να δούμε ότι η τιμή της παραπάνω έκφρασης, είναι πάντα ίδια με την τιμή του z. Ανεξάρτητη, δηλαδή, από τις τιμές των y_1, y_2 (*dummy variables*).

- Για κάθε clause $c \in C$ της αρχική φόρμουλας που αποτελείται από 2 literals $c = z_1 \vee z_2$, κατασκευάζουμε τα παρακάτω 2 clauses (νέα μεταβλητή y_1):

$$(z_1 \vee z_2 \vee y_1) \wedge (z_1 \vee z_1 \vee \neg y_1)$$

  Και εδώ είναι εύκολο να δούμε ότι η τιμή της παραπάνω παράστασης είναι πάντα ίδια με την τιμή της έκφρασης $(z_1 \vee z_2)$.

SAT ≤^{p} 3-SAT: κατασκευή 3-SAT φόρμουλας

- Κάθε clause της αρχικής φόρμουλας που αποτελείται από 3 literals το παίρνουμε όπως είναι στην καινούργια μας φόρμουλα.

- Τέλος, για κάθε clause της αρχικής φόρμουλας που έχει παραπάνω από 3 literals, έστω $c = (z_1 \vee z_2 \vee \ldots z_k)$, κατασκευάζουμε τα εξής clauses (y_i νέες μεταβλητές):

$$(z_1 \vee z_2 \vee y_1) \wedge (\neg y_1 \vee z_3 \vee y_2) \wedge (\neg y_2 \vee z_4 \vee y_3) \wedge \ldots$$
$$\wedge (\neg y_{k-4} \vee z_{k-2} \vee y_{k-3}) \wedge (\neg y_{k-3} \vee z_{k-1} \vee z_k)$$

SAT $\leq^p$ 3-SAT: Φ ικανοποιήσιμη <=> Φ' ικανοποιήσιμη

Όπως είπαμε κατά την διάρκεια της κατασκευής, αν ένα clause της αρχικής φόρμουλας με 1, 2 ή 3 literals ικανοποιείται, τότε θα ικανοποιούνται και τα αντίστοιχα clauses της Φ' και αντίστροφα. Αν έχουμε ένα clause στην αρχική φόρμουλα , $c = (z_1 \vee z_2 \vee \ldots \vee z_k)$, με $k \geq 4$ τότε για να ικανοποιείται το c θα πρέπει να έχει τιμή True τουλάχιστον ένα από τα literals του. Έστω λοιπόν $t(z_l) = True$. Θα δώσουμε μια απονομή αλήθειας που ικανοποιεί τα αντίστοιχα clauses στη φόρμουλα Φ':

$$t(y_i) = \begin{cases} \text{True}, & 1 \leq i \leq l-2 \\ \text{False}, & l-1 \leq i \leq k-3 \end{cases}$$

Και εδώ τα αντίστοιχα clauses της Φ' ικανοποιούνται (αυτά που βρίσκονται πριν από το clause i, από τα literals y_i και εκείνα που βρίσκονται μετά τα clause i, από τα literals $\neg y_i$).

Η απόδειξη του αντιστρόφου, ότι δηλαδή αν ικανοποιούνται τα clauses της Φ' τότε ικανοποιείται το αντίστοιχο clause της Φ γίνεται ως εξής: Έστω ότι δεν ικανοποιείται η $\Phi(x)$, δηλαδή $\forall i \; z_i = false$. Τότε μπορούμε εύκολα να δείξουμε (επαγωγή) ότι για να ικανοποιούνται τα n πρώτα clauses της $\Phi'(x)$ πρέπει οι μεταβλητές $y_1, y_2, \ldots, y_n$ να παίρνουν την αληθοτιμή true. Λόγω του παραπάνω για $n = k-3$ το τελευταίο clause παίρνει την τιμή false. □

Αναγωγή του 3-SAT σε άλλα προβλήματα

Γενικά, όταν ανάγουμε το 3SAT σε κάποιο άλλο πρόβλημα, τα δεδομένα μας είναι ένα σύνολο μεταβλητών $u_1, \ldots, u_n$ και ένα σύνολο από clauses $c_1, \ldots, c_m$. Τα στοιχεία που συνθέτουν την αναγωγή μας είναι:

- *Truthsetting*: Εξασφαλίζουμε ότι κάθε μεταβλητή έχει μία και μοναδική αληθοτιμή (truth value) σε όλα τα clauses.
- *Satisfaction*: Εξασφαλίζουμε ότι κάθε clause περιέχει τουλάχιστον ένα literal που ικανοποιείται (έχει τιμή True).
- *Remaining (interconnections)-garbage collection*: Εξασφαλίζουμε ότι έχουμε ένα σωστό πρόβλημα του καινούργιου τύπου.

Θεώρημα

Το πρόβλημα VERTEX COVER είναι NP-complete

3SAT $\leq^p_m$ VERTEX COVER

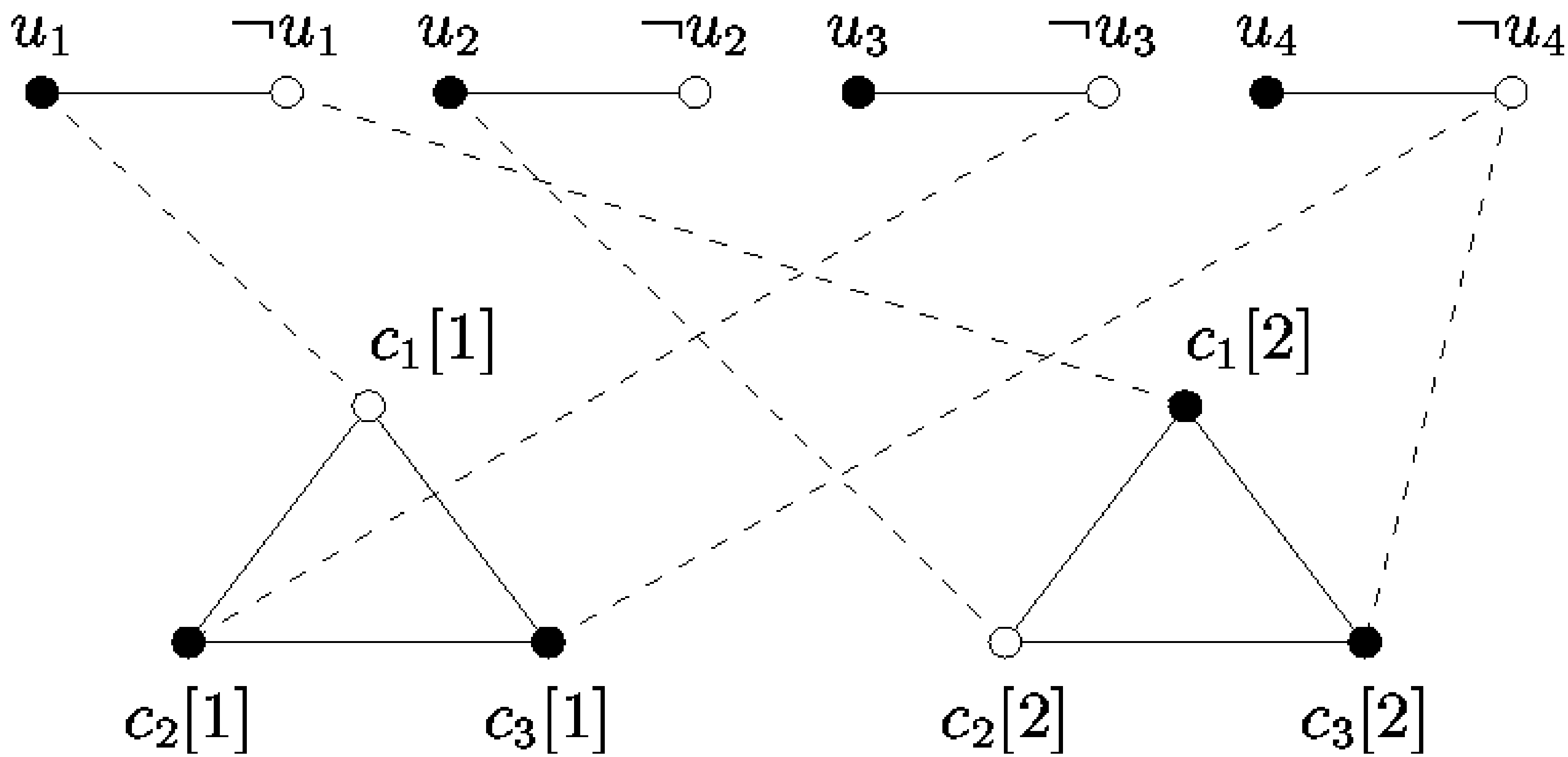
- Για κάθε μεταβλητή $u_i \in U$ εισάγουμε 2 κόμβους στο V, τους $u_i, \neg u_i$ και 1 πλευρά στο E, $(u_i, \neg u_i)$. Άρα συνολικά παίρνουμε $2n$ κόμβους και n πλευρές. Αυτό είναι το στάδιο truthsetting.

- Για κάθε clause $c_i \in C$ εισάγουμε 3 κόμβους στο V τους $c_1[i], c_2[i], c_3[i]$ και 3 πλευρές στο E, τις $(c_1[i], c_2[i]), (c_2[i], c_3[i])(c_3[i], c_1[i])$. Άρα συνολικά έχουμε $3m$ καινούργιους κόμβους και $3m$ πλευρές(αυτό είναι το στάδιο satisfaction).

- Τέλος προσθέτουμε τις πλευρές που χρειάζονται ώστε κάθε (satisfaction) τρίγωνο να συνδέεται με τρία αντίστοιχα (truthsetting) literals και το καινούριο πρόβλημα να είναι το VERTEX COVER (remaining interconnections). Για κάθε κόμβο $c_k[i], 1 \leq k \leq 3$ προσθέτουμε την πλευρά $(c_k[i], u_j)$ ή $(c_k[i], \neg u_j)$ ανάλογα με το αν στην $k-$οστή θέση του clause c_i εμφανίζεται το literal u_j ή $\neg u_j$ αντίστοιχα. Οι καινούριες πλευρές λοιπόν, είναι $3m$ (3 για κάθε clause). Η κατασκευή του γράφου έχει τελειώσει.

3SAT $^{p}_{m}$ VERTEX COVER

$$\Phi:\ (u_1 \vee \neg u_3 \vee \neg u_4) \wedge (\neg u_1 \vee u_2 \vee \neg u_4)$$



3SAT $^{p}_{m}$ VERTEX COVER

- Η κατασκευή γίνεται σε πολυωνυμικό χρόνο: ο γράφος *G* που κατασκευάζεται έχει πλήθος κόμβων $|V|=2n+3m$ και πλήθος ακμών $|E|=n+6m$, όπου n=αριθμός μεταβλητών της Φ, και m=αριθμός clauses της Φ.

- Ισχυρισμός: ο γράφος *G* έχει clique μεγέθους $k=n+m$ αν και μόνο αν η Φ είναι ικανοποιήσιμη.

3SAT ${}^{p}_{m}$ VERTEX COVER : απόδειξη "<="

Απόδειξη του ισχυρισμού: Έστω ότι η φόρμουλα Φ είναι ικανοποιήσιμη και έστω t μια απονομή αληθείας που ικανοποιεί την Φ, τότε βρίσκουμε ένα vertex cover ως εξής: Παίρνουμε εκείνους τους κόμβους που αντιστοιχούν σε literals οι οποίοι έχουν τιμή True. Αυτοί οι κόμβοι θα είναι ακριβώς n, αφού θα ισχύει $t(u_i) = True$, είτε $t(\neg u_i) = True$ αλλά όχι συγχρόνως (truthsetting). Συνεπώς αυτοί οι κόμβοι θα καλύπτουν $n + m$ πλευρές τουλάχιστον, αφού κάθε clause της αρχικής φόρμουλας Φ, θα έχει τουλάχιστον ένα literal που το ικανοποιεί (satisfaction). Στη χειρότερη περίπτωση λοιπόν, δηλαδή όταν κάθε clause ικανοποιείται από ένα ακριβώς literal, έχουν μείνει $5m$ πλευρές ακάλυπτες. Τις καλύπτουμε παίρνοντας 2 κόμβους από κάθε τριγωνάκι (συνολικά $2m$ κόμβους), εκείνους που συνδέονται με literals που έχουν τιμή False. Γενικώς, επειδή κάθε clause ικανοποιείται από τουλάχιστον ένα literal, τουλάχιστον μια σύνδεση (satisfaction) τρίγωνο με truthsetting είναι ήδη καλυμμένη. Οπότε παίρνοντας τις 2 άλλες κορυφές του τριγώνου καλύπτουμε σίγουρα όλες τις υπόλοιπες πλευρές. Έτσι καλύπτονται όλες οι πλευρές του γράφου με $n + 2m$ κόμβους.

3SAT $^{p}_{m}$ VERTEX COVER: απόδειξη "=>"

Αντίστροφα, αν ο G έχει vertex cover $V' \sqsubseteq V$ μεγέθους $|V'| \leq k = n+2m$, το V' περιλαμβάνει τουλάχιστον έναν κόμβο από κάθε πλευρά (truthsetting) της μορφής $u_i \;\; \neg u_i$ (ειδάλλως θα υπάρχει πλευρά αυτής της μορφής που δεν θα καλύπτεται) και τουλάχιστον δύο κόμβους από κάθε τρίγωνο (ειδάλλως δεν θα καλύπτονται όλες οι πλευρές του τριγώνου). Δηλαδή το V' θα περιέχει τουλάχιστον $n+2m$ κόμβους. Συνεπώς (αφού από την υπόθεση $V' \leq n+2m$, το V' θα περιέχει ακριβώς 1 κόμβο από κάθε πλευρά της μορφής $u_i \;\; \neg u_i$ και ακριβώς 2 κόμβους από κάθε τρίγωνο. Θα δώσουμε μια απονομή αλήθειας που ικανοποιεί την Φ: $t(u_i) = \begin{cases} \text{True}, & u_i \in V' \\ \text{False}, & \neg u_i \in V'. \end{cases}$

Οι δύο κόμβοι που έχουμε πάρει από κάθε τρίγωνο μπορούν να καλύπτουν μόνο δύο από τις πλευρές της μορφής $(c_k[i], u_j)$. Η τρίτη πλευρά πρέπει υποχρεωτικά να καλύπτεται από κάποιο u_i ή $\neg u_i$ που ανήκει στο V'. Δηλαδή, κάθε τριγωνάκι συνδέεται με κάποιο από τα u_i ή $\neg u_i$ που ανήκουν στο V'. Άρα κάθε clause της φόρμουλας Φ, έχει ένα literal u_i ή $\neg u_i$ με $t(u_i)$ ή $t(\neg u_i)$ True. Συνεπώς η Φ ικανοποιείται. $\square$

Θεώρημα

Το πρόβλημα
3-DIMENSIONAL MATCHING (3DM)
είναι NP-complete

3SAT $\leq^p_m$ 3DM

Μας δίνεται ένα σύνολο μεταβλητών $U = \{u_1, \ldots, u_n\}$ και ένα σύνολο από clauses $C = \{c_1, \ldots, c_m\}$. Θα κατασκευάσουμε 3 ξένα μεταξύ τους σύνολα W, X, Y με $|W| = |X| = |Y|$ και ένα σύνολο $M \subseteq W \times X \times Y$ έτσι ώστε το M να έχει matching αν και μόνο αν η φόρμουλα Φ: $c_1 \wedge c_2 \wedge \ldots \wedge c_m$ είναι ικανοποιήσιμη. Η κατασκευή γίνεται ως εξής:

Βάζουμε στο W όλα τα στοιχεία $u_i[j], \neg u_i[j], 1 \leq j \leq m, 1 \leq i \leq n$ ($2mn$ το πλήθος). Στο σύνολο X βάζουμε τα στοιχεία:

- $a_i[j], 1 \leq j \leq m,\ 1 \leq i \leq n$, ($nm$ το πλήθος, τα οποία θα χρησιμοποιηθούν στο truthsetting)
- $S_1[j], 1 \leq j \leq m$, (m το πλήθος, τα οποία θα χρησιμοποιηθούν στο satisfaction)
- $g_1[k], 1 \leq k \leq (n-1)m$ ($(n-1)m$ το πλήθος, τα οποία θα χρησιμοποιηθούν στο garbage collection).

Όμοια με το σύνολο X φτιάχνουμε και το σύνολο Y βάζοντας τα στοιχεία:

- $b_i[j], 1 \leq j \leq m,\ 1 \leq i \leq n$.
- $S_2[j], 1 \leq j \leq m$.
- $g_2[k], 1 \leq k \leq (n-1)m$.

3SAT $\leq^p_m$ 3DM

Το σύνολο $M \subseteq W \times X \times Y$ το κατασκευάζουμε ως εξής: Για κάθε μεταβλητή u_i βάζουμε στο M τα σύνολα τριάδων T_i^t και T_i^f όπου:

$$T_i^t = \{(\neg u_i[j], a_i[j], b_i[j]), 1 \leq j \leq m\}$$

$$T_i^f = \{(u_i[j], a_i[j+1], b_i[j]), 1 \leq j \leq m\} \cup \{(u_i[m], a_i[1], b_i[m])\}$$

Όλες αυτές οι τριάδες είναι $2nm$ στο πλήθος και χρησιμοποιούνται για το truthsetting. Για κάθε clause c_j βάζουμε στο M το σύνολο τριάδων C_j όπου:

$$C_j = \{(u_i[j], s_1[j], s_2[j]) \mid u_i \in c_j \text{ clause}\} \cup \{(\neg u_i[j], s_1[j], s_2[j]) \mid \neg u_i \in c_j \text{ clause}\}, 1 \leq j \leq m\}$$

Συνολικά είναι $3m$ τριάδες και χρησιμοποιούνται για το satisfaction.

Τέλος βάζουμε στο M τα στοιχεία που λείπουν για να ανήκει το στιγμιότυπό μας στο πρόβλημα 3DM (garbage collection). Βάζουμε δηλαδή το σύνολο τριάδων G όπου:

$$G = \{(u_i[j], g_1[k], g_2[k]),\ (\neg u_i[j], g_1[k], g_2[k])\}, \quad 1 \leq k \leq m(n-1),\ 1 \leq i \leq n, 1 \leq j \leq m\}$$

3SAT $^{p}_{m}$ 3DM

Ένα οποιοδήποτε matching $M' \subseteq M$ θα πρέπει να περιλαμβάνει τουλάχιστον nm τριάδες από τα T_i^f, T_i^t ώστε να καλύψουμε όλα τα $a_i[j]$ και $b_i[j]$, τα οποία εμφανίζονται μόνο σε τριάδες στα T_i^f, T_i^t. Επίσης, επειδή υπάρχουν ακριβώς mn διαφορετικά $a_i[j]$ προκύπτει ότι το M' περιλαμβάνει ακριβώς mn τριάδες από τα T_i^f, T_i^t. Μάλιστα το M' θα περιλαμβάνει για κάθε u_i, είτε ολόκληρο το T_i^f είτε ολόκληρο το T_i^t. Επίσης το matching M' προτείνει μια απονομή αλήθειας που ικανοποιεί τη φόρμουλα Φ:

$$t(u_i) = \begin{cases} \text{True}, & T_i^t \subseteq M' \\ \text{False}, & T_i^f \subseteq M' \end{cases}$$

Ένα οποιοδήποτε matching M' θα πρέπει να περιέχει για κάθε clause μία ακριβώς τριάδα (σύνολο m τριάδες). Το literal $u_i[j]$ (ή $\neg u_i[j]$) που θα εμφανίζεται στην τριάδα C_j στο matching M' θα είναι ακριβώς εκείνο που ικανοποιεί το clause c_j, αφού για να υπάρχει στο M', σημαίνει ότι $T_i^t \subseteq M'(T_i^f \subseteq M')$.

Το matching M' έχει ως τώρα $nm + m$ τριάδες και συνεπώς του λείπουν $2mn - nm - m = m(n-1)$ τριάδες. Αυτές θα μπορεί πάντα να τις πάρει από το G, αφού θα υπάρχουν $m(n-1)$ literals που δεν έχει ως τώρα το M'. Η κατασκευή έχει τελειώσει και συνολικά το πλήθος των τριάδων που περιέχει το M είναι $|M| = 2nm + 3m + 2m^2n(n-1)$. Συνεπώς και εφόσον ο τρόπος κατασκευής του από το 3SAT είναι σχεδόν άμεσος, το M μπορεί να κατασκευαστεί σε πολυωνυμικό χρόνο.

3SAT $\leq^p_m$ 3DM

Ήδη αποδείξαμε πως αν το M περιέχει ένα matching, τότε η φόρμουλα Φ είναι ικανοποιήσιμη. Αν η φόρμουλα Φ είναι ικανοποιήσιμη, τότε θα κατασκευάσουμε ένα matching M' ως εξής:

- Διαλέγουμε όλα τα σύνολα T_i^t για τα οποία $t(u_i) = True$ και όλα τα σύνολα T_i^f για τα οποία $t(u_i) = False$. Δηλαδή, όλα τα literals που θα ανήκουν σ' αυτές τις τριάδες θα έχουν τιμή False και θα είναι ακριβώς mn.

- Επίσης διαλέγουμε m τριάδες της μορφής

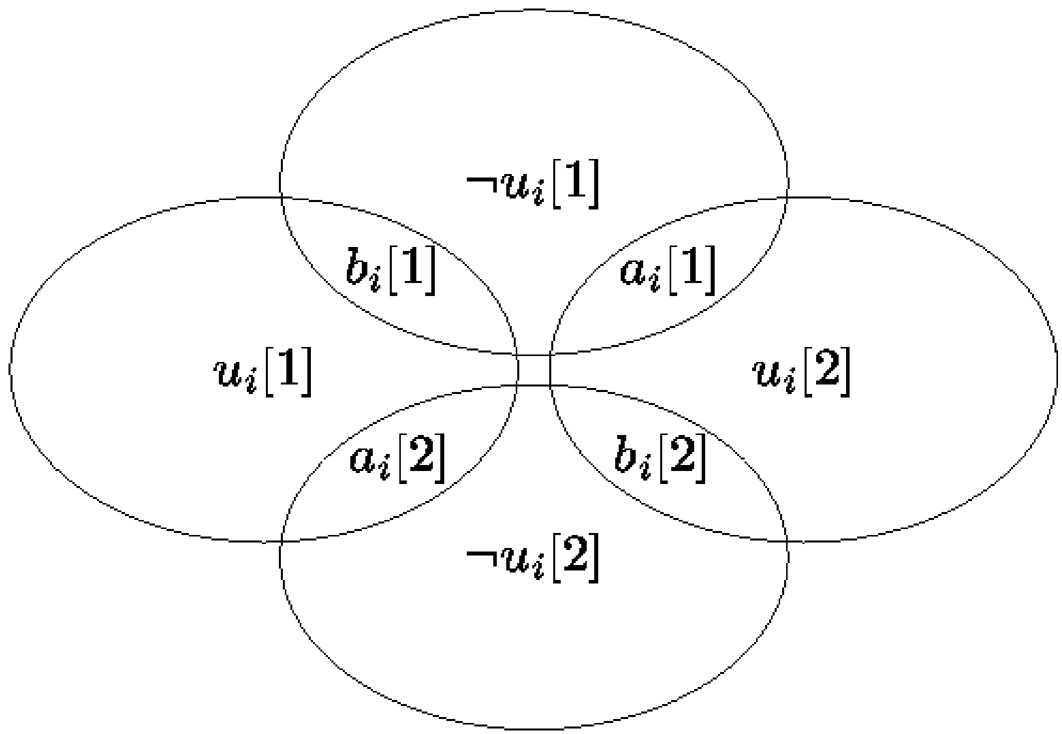
$$(u_i[j], s_1[j], s_2[j]) \text{ ή } (\neg u_i[j], s_1[j], s_2[j])$$

  οι οποίες προφανώς θα περιέχουν literals με τιμή True (ειδάλλως θα καταστρέφεται το matching). Θα υπάρχουν πάντα m τέτοιες τριάδες, αφού κάθε clause θα έχει ένα literal που το ικανοποιεί.

- Τέλος παίρνουμε ένα σύνολο $G' \subseteq G$ με $m(n-1)$ τριάδες οι οποίες θα περιέχουν literals με τιμή True. Ξέρουμε ότι υπάρχουν $m(n-1)$ τέτοιες τριάδες αφού έχουμε πάρει μόνο m με literals που έχουν τιμή True ως τώρα. Συνεπώς το M' περιέχει $2nm$ διαφορετικές τριάδες και άρα είναι ένα ταίριασμα του M.

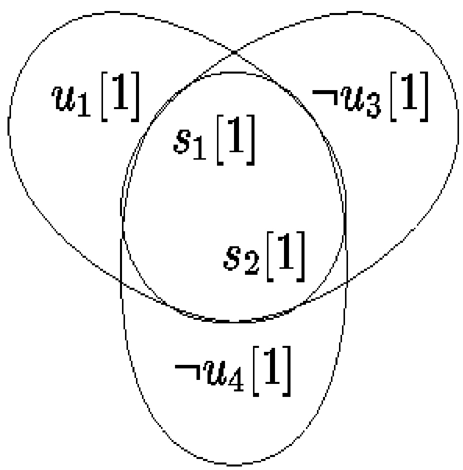
3SAT $^{p}_{m}$ 3DM

$T_i^t \cup T_i^f$:

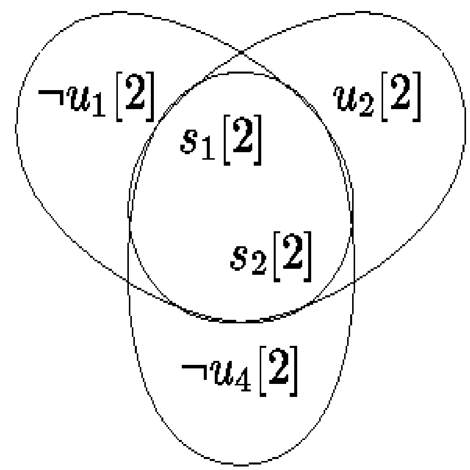


3SAT $^{p}_{m}$ 3DM

Παράδειγμα 14.6.2. Έχουμε τη φόρμουλα Φ: $(u_1 \vee \neg u_3 \vee \neg u_4) \wedge (\neg u_1 \vee u_2 \vee \neg u_4)$ η οποία είναι ικανοποιήσιμη. Μία απονομή αλήθειας που την ικανοποιεί είναι η $t(u_1, u_2, u_3, u_4) = (T, T, F, T)$. Για κάθε μεταβλητή u_i, το M θα περιλαμβάνει το σύνολο που φαίνεται στο σχήμα 14.6. Επίσης το M θα περιλαμβάνει τα σύνολα C_1 και C_2 που φαίνονται στο σχήμα 14.7. Τέλος το M θα περιλαμβάνει το σύνολο G με τα στοιχεία που απομένουν. Ένα matching $M' \subseteq M$ που προκύπτει από την απονομή (T, T, F, T) είναι το εξής:



C_1



C_2

3SAT $^{p}_{m}$ 3DM

$$
\begin{aligned}
M' = \{ & (\neg u_1[1], a_1[1], b_1[1]), \\
& (\neg u_1[2], a_1[2], b_1[2]), \\
& (\neg u_2[1], a_2[1], b_2[1]), \\
& (\neg u_2[2], a_2[2], b_2[2]), \\
& (u_3[1], a_3[1], b_3[1]), \\
& (u_3[2], a_3[2], b_3[2]), \\
& (\neg u_4[1], a_4[1], b_4[1]), \\
& (\neg u_4[2], a_4[2], b_4[2])\} \bigcup T' \bigcup G'
\end{aligned}
$$

όπου

$$T' = \{(u_1[1], s_1[1], s_2[1]), (u_2[2], s_1[2], s_2[2])\}$$

και

$$
\begin{aligned}
G' = \{ & (u_1[2], g_1[1], g_2[1]), (u_2[1], g_1[2], g_2[2]), \\
& (\neg u_3[1], g_1[3], g_2[3]), (\neg u_3[2], g_1[4], g_2[4]), \\
& (u_4[1], g_1[5], g_2[5]), (u_4[2], g_1[6], g_2[6])\}
\end{aligned}
$$

3SAT ${}^{p}_{m}$ 3DM

Παρατήρηση 14.6.3. Το πρόβλημα 2-DIMENSIONAL MATCHING ανήκει στο P. Δηλαδή, όταν έχουμε 2 σύνολα n στοιχείων και θέλουμε να βρούμε ένα ταίριασμα, μπορούμε να το κάνουμε σε πολυωνυμικό χρόνο με έναν ντετερμινιστικό αλγόριθμο. Το πρόβλημα $2DM$ αναφέρεται και σαν πρόβλημα γάμου (*marriage problem*) λόγω των προφανών προεκτάσεων που μπορεί να έχει στη ζωή (το ένα σύνολο περιέχει n άντρες και το άλλο n γυναίκες). Επίσης, άλλα παρόμοια προβλήματα είναι τα εξής:

- Μπορούν να τοποθετηθούν 8 βασίλισσες (queens) σε μια σκακιέρα (8×8) έτσι ώστε καμιά βασίλισσα να μην απειλεί κάποια άλλη;
- Μπορούν να τοποθετηθούν 8 πύργοι σε μια σκακιέρα (8×8) έτσι ώστε κανένας πύργος να μην απειλεί κάποιον άλλο; (εύκολο)
- Μπορούν να τοποθετηθούν 8 πύργοι όπως και πριν, σε ένα δεδομένο υποσύνολο της σκακιέρας; (δύσκολο, αλλά στο P: είναι το πρόβλημα γάμου)

Θεώρημα

Το πρόβλημα

GRAPH 3-COLORABILITY

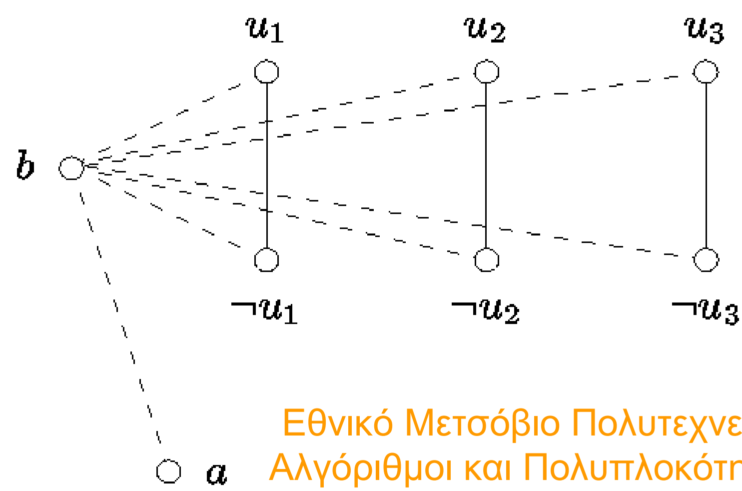
είναι NP-complete

3SAT $^{p}_{m}$ 3-COLORABILITY

Μας δίνεται ένα σύνολο μεταβλητών $U = \{u_1, \ldots, u_n\}$ και ένα σύνολο από clauses $C = \{c_1, \ldots, c_m\}$.

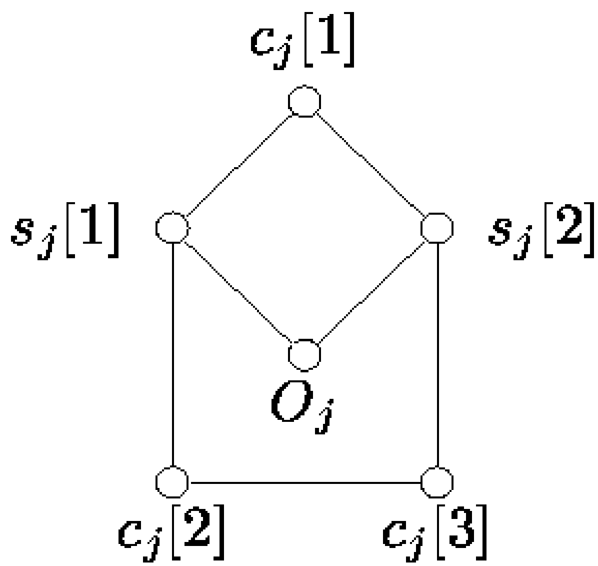
Θα κατασκευάσουμε έναν γράφο $G(V, E)$ έτσι ώστε η φόρμουλα Φ: $(c_1 \wedge c_2 \wedge \ldots \wedge c_m)$ να είναι ικανοποιήσιμη αν και μόνο αν ο γράφος G βάφεται με 3 χρώματα. Η κατασκευή του G γίνεται ως εξής:

- Για κάθε μεταβλητή $u_i \in U$ βάζουμε στο V τους κόμβους $u_i, \neg u_i$ και στο E τις πλευρές $(u_i, \neg u_i)$. Επίσης βάζουμε στο V τους κόμβους a, b και στο E τις πλευρές $(a, b), (u_i, b), (\neg u_i, b)$. Δηλαδή ως τώρα $|V| = 2n + 2, |E| = 3n + 1$. Ο γράφος G παίρνει τη μορφή που φαίνεται στο σχήμα 14.8. Αυτό είναι το στάδιο truthsetting.



3SAT $^{p}_{m}$ 3-COLORABILITY

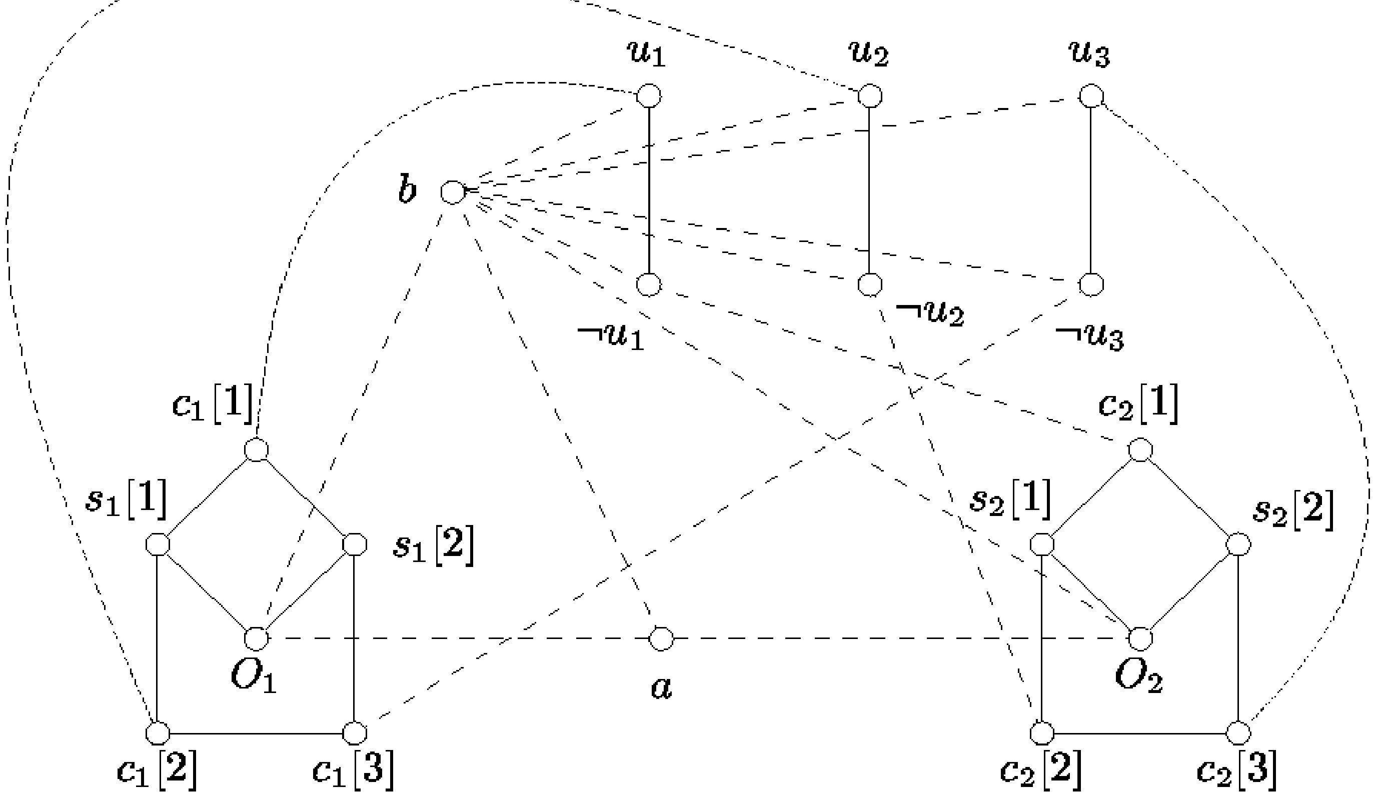
- Για κάθε clause c_j προσθέτουμε στο γράφο G, έναν υπογράφο σαν αυτόν που φαίνεται στο σχήμα 14.9. Επίσης βάζουμε στο E τις πλευρές $(c_j[k], u_i)$ ή $(c_j[k], \neg u_i)$ ανάλογα με το αν το k-οστό literal του c_j είναι το u_i ή $\neg u_i$ αντίστοιχα. Δηλαδή έχουμε $6m$ καινούργιους κόμβους για το V και $10m$ καινούριες πλευρές για το E. Αυτό είναι το στάδιο satisfaction.



3SAT $^{p}_{m}$ 3-COLORABILITY

- Τέλος κάνουμε τις συνδέσεις που απομένουν (remaining interconnections) για να είναι το πρόβλημά μας, του σωστού τύπου. Προσθέτουμε, δηλαδή, τις πλευρές (O_j, b) και τις πλευρές (O_j, a) ($2m$ στο πλήθος). Η κατασκευή μας έχει τελειώσει.

$$\Phi : (u_1 \vee u_2 \vee \neg u_3) \wedge (\neg u_1 \vee \neg u_2 \vee u_3)$$



3SAT $\leq^p_m$ 3-COLORABILITY

- Η κατασκευή γίνεται σε πολυωνυμικό χρόνο: ο γράφος *G* που κατασκευάζεται έχει πλήθος κόμβων $|V|=2n+6m+2$ και πλήθος ακμών $|E|=3n+12m+1$, όπου n=αριθμός μεταβλητών της Φ, m=αριθμός clauses της Φ.

- Ισχυρισμός: η Φ είναι ικανοποιήσιμη αν και μόνο αν ο *G* βάφεται με 3 χρώματα.

3SAT $\leq^p_m$ 3-COLORABILITY: απόδειξη "=>"

Εστω ότι η Φ ικανοποιείται:

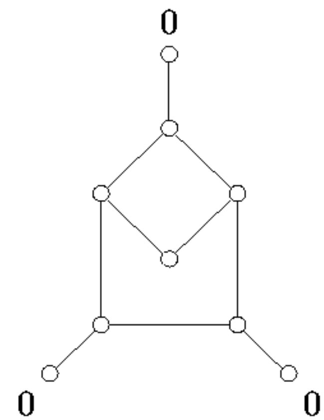
Βάφουμε τον κόμβο b με το χρώμα 2 και τον κόμβο a με το χρώμα 1. Συνεπώς, όλες οι κορυφές O_j πρέπει να βαφούν με το χρώμα 0. Επίσης βάφουμε τους κόμβους $u_i, \neg u_i$ ως εξής:

$$f(u_i) = \begin{cases} 1, & t(u_i) = True \\ 0, & t(u_i) = False \end{cases} \quad \text{και} \quad f(\neg u_i) = \begin{cases} 0, & t(u_i) = True \\ 1, & t(u_i) = False \end{cases}$$

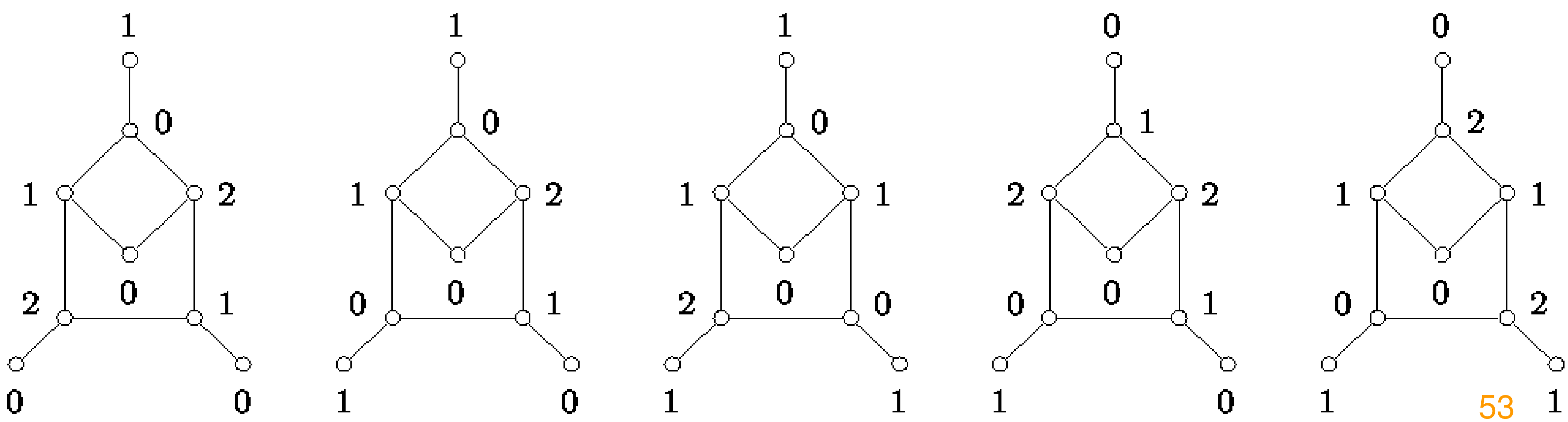
Δηλαδή, όλα τα literals που γίνονται True με την απονομή που ικανοποιεί την Φ τα βάφουμε με το χρώμα 1 και όλα τα literals που γίνονται False τα βάφουμε με το χρώμα 0. Απομένει να βάψουμε τους υπογράφους που αντιστοιχούν στα clauses. Επειδή η φόρμουλα Φ ικανοποιείται σημαίνει ότι υπάρχει τουλάχιστον ένας κόμβος u_i ή $\neg u_i$ γειτονικός ενός κόμβου $c_j[k]$, ο οποίος έχει βαφεί με το χρώμα 1. Είναι ο κόμβος που αντιστοιχεί στο literal το οποίο ικανοποιεί το clause c_j. Συνεπώς δε μπορεί να υπάρχει υπογράφος που έχει χρωματιστεί όπως στο σχήμα 14.10. Όλες οι δυνατές περιπτώσεις, είναι αυτές που φαίνονται στο σχήμα 14.11 και οι συμμετρικές τους. Άρα σε οποιαδήποτε περίπτωση το G βάφεται με 3 χρώματα.

3SAT $^{p}_{m}$ 3-COLORABILITY: απόδειξη "=>"

- Αποκλείεται:



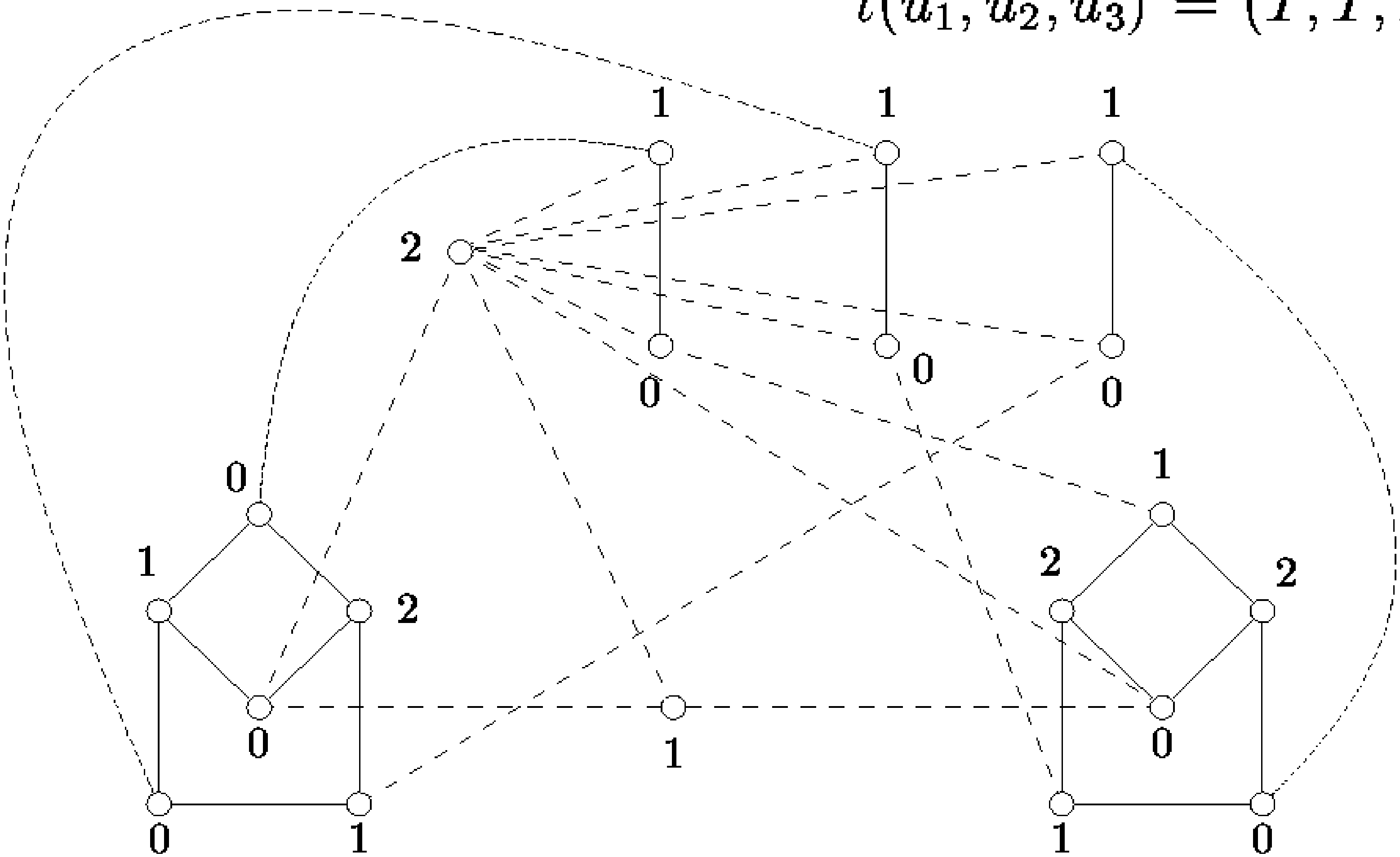
- Δυνατές περιπτώσεις:



3SAT $^{p}_{m}$ 3-COLORABILITY: απόδειξη "=>"

Παράδειγμα: $\Phi: (u_1 \vee u_2 \vee \neg u_3) \wedge (\neg u_1 \vee \neg u_2 \vee u_3)$

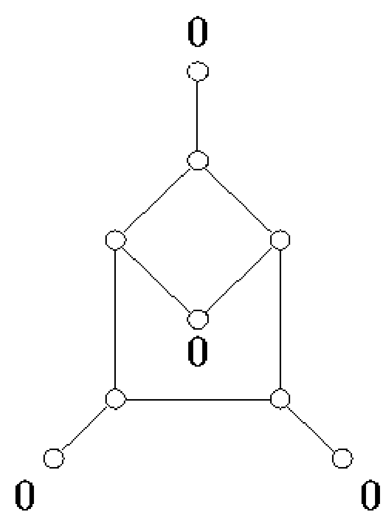
$$t(u_1, u_2, u_3) = (T, T, T)$$



3SAT $^{p}_{m}$ 3-COLORABILITY: απόδειξη "<="

Εστω ότι ο G βάφεται με 3 χρώματα:

Έστω ότι το b έχει βαφεί με το χρώμα 2, το a με το χρώμα 1 και το O_j με το χρώμα 0. Τότε οι κόμβοι $u_i, \neg u_i$ πρέπει να βαφούν με τα χρώματα 0, 1. Θεωρούμε την απονομή $t(u_i) = True$ όταν το u_i έχει χρώμα 1 και $t(u_i) = False$ όταν το u_i έχει χρώμα 0. Αυτή η απονομή θα ικανοποιεί τη φόρμουλα Φ, διότι αν δεν την ικανοποιεί, σημαίνει ότι θα υπάρχει clause του οποίου όλα τα literals θα είναι False. Δηλαδή στον γράφο G θα υπάρχει υπογράφος σαν αυτόν του σχήματος 14.14. Αυτός ο υπογράφος, όπως εύκολα μπορεί να διαπιστώσει κανείς, δεν βάφεται με 3 χρώματα. ΆΤΟΠΟ. □



Θεώρημα

Το πρόβλημα
HAMILTON CIRCUIT (HC)
είναι NP-complete

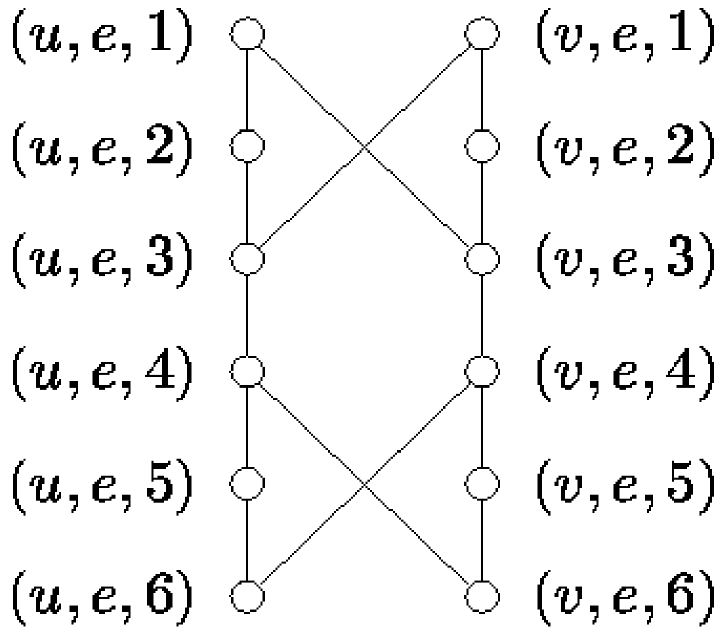
VERTEX COVER $\leq^p_m$ HC

Μας δίνεται ένας γράφος $G(V, E)$ και ένας θετικός ακέραιος $k \leq |V|$. Θα κατασκευάσουμε έναν γράφο $G'(V', E')$ έτσι ώστε ο γράφος $G(V, E)$ να έχει vertex cover μεγέθους $k \leq |V|$ αν και μόνο αν ο γράφος $G'(V', E')$ έχει κύκλο Hamilton. Η κατασκευή γίνεται ως εξής:

- Βάζουμε στο V', k κόμβους $a_1, a_2, \ldots, a_k$ οι οποίοι ονομάζονται και selector vertices διότι θα χρησιμοποιηθούν για να επιλέξουν k κόμβους από το σύνολο V του G.

- Για κάθε πλευρά (u, v), βάζουμε στο γράφο G' μια «γέφυρα», δηλαδή το component που φαίνεται στο σχήμα 14.15. Δηλαδή, προσθέτουμε στο V' $12|E|$ κόμβους και στο E' $14|E|$ πλευρές. Αυτές οι γέφυρες αποτελούν τα cover testing components, διότι θα χρησιμοποιηθούν για να εξασφαλίζουν ότι τουλάχιστον ένας από τους κόμβους u, v ανήκει στους k κόμβους που αποτελούν vertex cover για το G.

VERTEX COVER $\leq^p_m$ HC

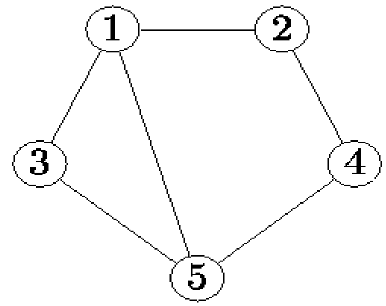


Σχήμα 14.15: Γέφυρα για την αναγωγή του VC στο HC

VERTEX COVER $\leq^p_m$ HC

- Ας θεωρήσουμε, $\forall v \in V$, μία μετάθεση όλων των πλευρών που ξεκινούν απ' το v:
$$\left\langle e_{v_{(1)}}, e_{v_{(2)}}, \ldots, e_{v_{(d(v))}} \right\rangle$$
όπου $d(v)$ είναι ο βαθμός της κορυφής v. Θεωρούμε, δηλαδή, n τέτοιες μεταθέσεις ($|V| = n$), μία για κάθε κόμβο του V. Η κάθε μία απ' αυτές θα έχει μήκος ίσο με το βαθμό της αντίστοιχης κορυφής ($n-1$ το πολύ). Για κάθε κόμβο v, ενώνουμε όλες τις γέφυρες που αντιστοιχούν σε μια τέτοια ακολουθία. Προσθέτουμε, δηλαδή, στο E' το σύνολο των πλευρών E'_v, όπου
$$E'_v = \{((v, e_{v_{(i)}}, 6), (v, e_{v_{(i-1)}}, 1))\}, 1 \leq i \leq d(v), \forall v \in V$$
Στη χειρότερη περίπτωση ο αριθμός πλευρών που προσθέτουμε είναι $O(n^2)$ ($\sum_{v \in V}(d(v) - 1)$). Έτσι λοιπόν, τώρα έχουν δημιουργηθεί στο G', n μονοπάτια που αντιστοιχούν στις n μεταθέσεις.

VERTEX COVER $\leq^p_m$ HC

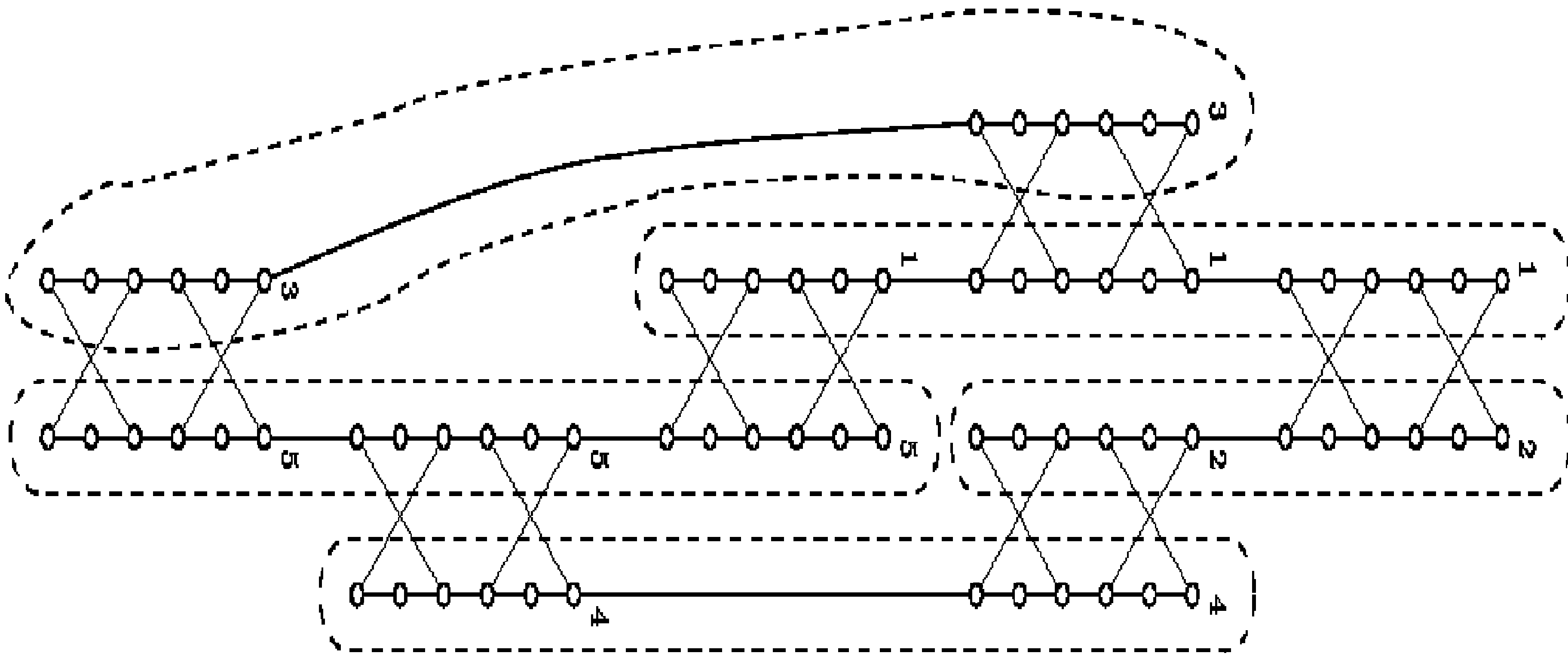


Σχήμα 14.16: Γράφος παραδείγματος αναγωγής VC στο HC

Παράδειγμα: Έστω ότι έχουμε το γράφο που φαίνεται στο σχήμα 14.16. Επιλέγουμε τις μεταθέσεις:

- Για τον κόμβο 1: $< (1,2),(1,3),(1,5) >$
- Για τον κόμβο 2: $< (2,1),(2,4) >$
- Για τον κόμβο 3: $< (3,1),(3,5) >$
- Για τον κόμβο 4: $< (4,2),(4,5) >$
- Για τον κόμβο 5: $< (5,1),(5,4),(5,3) >$

Τα n μονοπάτια που δημιουργούνται φαίνονται στο σχήμα 14.17.



3
3
1
1
1
5
5
5
2
2
4
4

VERTEX COVER $\leq^p_m$ HC

Τέλος, συνδέουμε τα δύο άκρα του κάθε μονοπατιού με όλα τα σημεία $a_1, a_2, \ldots, a_k$. Δηλαδή, προσθέτουμε στο E' το σύνολο E'' το οποίο αποτελείται από τις εξής $2kn$ πλευρές:

$$E'' = \{((v, e_{v_{(1)}}, 1), a_i), ((v, e_{v_{(d(v))}}, 6), a_i)\}, 1 \leq i \leq k, \forall v \in V$$

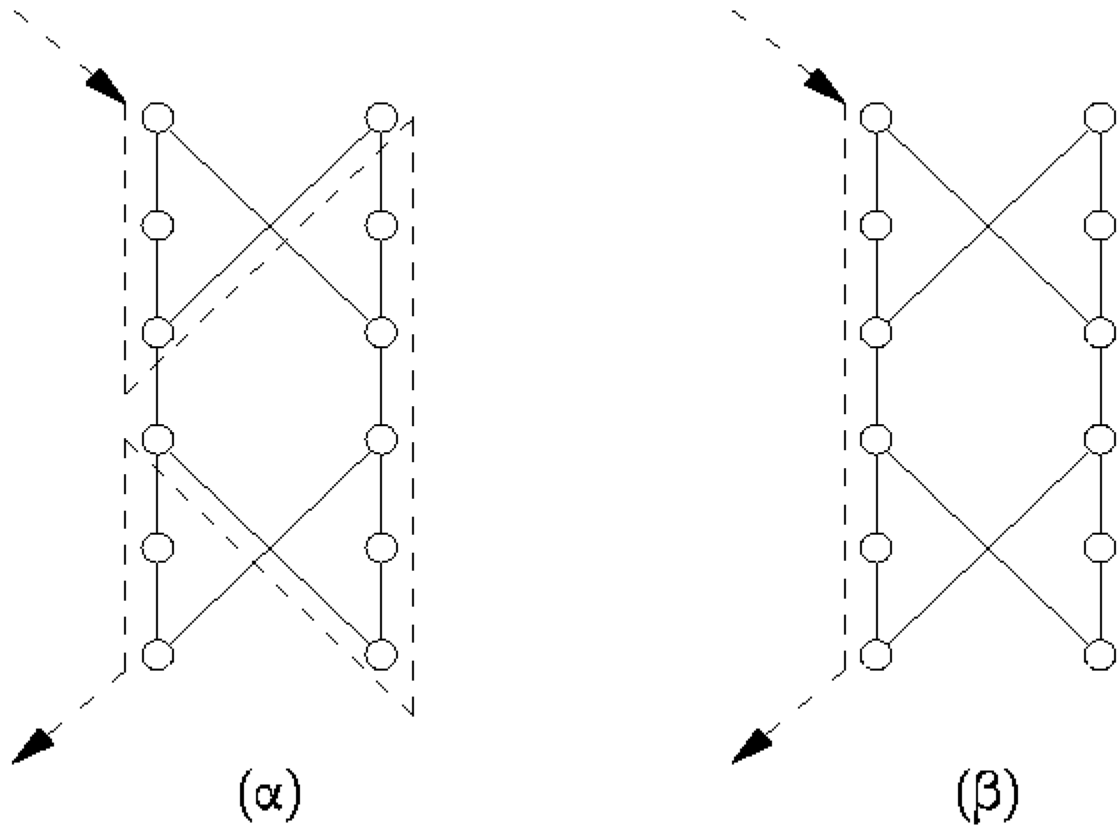
Η κατασκευή του G' έχει τελειώσει και εφόσον τα V', E' έχουν πολυωνυμικό μέγεθος ως προς τα V, E αντίστοιχα, το G' μπορεί να κατασκευαστεί σε πολυωνυμικό χρόνο.

VERTEX COVER $\leq^p_m$ HC

Έτσι λοιπόν, το μόνο που απομένει να δείξουμε είναι ότι το $G(V,E)$ έχει vertex cover μεγέθους $\leq k$ αν και μόνο αν το $G'(V',E')$ έχει κύκλο Hamilton. Έστω ότι ο γράφος G έχει vertex cover μεγέθους $\leq k$. Τότε σίγουρα υπάρχουν και ακριβώς k κόμβοι που καλύπτουν όλες τις πλευρές του G. Συμβολίζουμε το σύνολο αυτών των k κόμβων με V_c. Θα περιγράψουμε έναν κύκλο Hamilton στο γράφο G'.

Ξεκινάμε από έναν κόμβο a_i, έστω χωρίς βλάβη της γενικότητας, τον a_1 και από εκεί πηγαίνουμε στον κόμβο $(v, e_{v_{(1)}}, 1)$ όπου $v \in V_c$. Για να διατρέξουμε την γέφυρα που ξεκινά με τον κόμβο $(v, e_{v_{(1)}}, 1)$, περνώντας μία ακριβώς φορά από κάθε σημείο της, πρέπει να επιλέξουμε έναν από τους δύο τρόπους που φαίνονται στο σχήμα 14.18.

VERTEX COVER $\leq^p_m$ HC



Σχήμα 14.18: Οι δύο τρόποι διάσχισης της κάθε γέφυρας

VERTEX COVER $\leq^p_m$ HC

Αν $u \notin V_c$ περνάμε την γέφυρα με τον τρόπο (α). Αν $u \in V_c$ περνάμε την γέφυρα με τον τρόπο (β). Μετά πηγαίνουμε στη γέφυρα που ξεκινά με τον κόμβο $(v, e_{v_{(2)}}, 1)$ και αντιστοιχεί στην πλευρά (v, u') του G. Την περνάμε κι αυτή με τον ίδιο τρόπο (ανάλογα με το αν ($u' \in V_c$ ή $u' \notin V_c$). Έτσι περνάμε και την τελευταία γέφυρα του κόμβου v καταλήγοντας σε έναν άλλο κόμβο από τους k selectors, έστω a_2.

Από το a_2 περνάμε στα testing components ενός άλλου κόμβου που ανήκει στο V_c, κ.ο.κ., έως ότου περάσουμε k τέτοια μονοπάτια καταλήγοντας τελικά στο σημείο a_1 από το οποίο ξεκινήσαμε. Αυτή η διαδρομή είναι κύκλος Hamilton, διότι αν υπήρχε γέφυρα από την οποία δεν περνούσαμε, θα σήμαινε ότι η αντίστοιχη πλευρά στο G δεν είχε καλυφθεί. ΆΤΟΠΟ.

Επίσης, αν δεν έφταναν τα k selectors για να πάρουμε όλα τα testing components, τότε ο G δεν θα είχε vertex cover μεγέθους k. ΆΤΟΠΟ.

Αντίστροφα, αν το G' έχει κύκλο Hamilton, τότε επειδή διατρέχουμε k μονοπάτια, σημαίνει ότι κάθε γέφυρα πρέπει να ανήκει σε ένα τουλάχιστον απ' αυτά τα μονοπάτια. Συνεπώς κάθε πλευρά του G (που αντιστοιχεί σε κάποια γέφυρα του G') καλύπτεται τουλάχιστον από 1 κόμβο του G (που αντιστοιχεί σε κάποιο από τα k μονοπάτια). $\square$

Θεώρημα

Το TRAVELING SALESMAN PROBLEM (TSP)
είναι NP-complete

Απόδειξη: HC $\leq^p_m$ TSP (i)

Απόδειξη. Το TSP ανήκει στο NP, διότι ένας μη-ντετερμινιστικός αλγόριθμος μαντεύει μια διάταξη των κόμβων και ελέγχει σε πολυωνυμικό χρόνο αν το άθροισμα των βαρών των αντίστοιχων πλευρών είναι $\leq B$. Θα δείξουμε ότι το TSP είναι NP-complete ανάγοντας το HC σ' αυτό ($HC \leq^p_m TSP$).

Μας δίνεται ένας γράφος $G(V, E)$. Θέλουμε να κατασκευάσουμε έναν πλήρη γράφο $G'(V', E')$ με βάρη $d(u, v), \forall (u, v) \in E'$ και έναν θετικό ακέραιο B, έτσι ώστε ο γράφος $G(V, E)$ να έχει κύκλο Hamilton αν και μόνο αν ο $G'(V', E')$ έχει tour με βάρος $\leq B$. Η κατασκευή γίνεται ως εξής:

- Σαν γράφο $G'(V', E')$, παίρνουμε τον γράφο $G(V, E)$, προσθέτοντας όλες τις ακμές που υπολείπονται για να γίνει πλήρης. Βάζουμε βάρη στις ακμές του G' ως εξής:

$$d(u, v) = \begin{cases} 1, & (u, v) \in G \\ 2, & (u, v) \notin G \end{cases}$$

- Τέλος, παίρνουμε $B = |V'| = |V|$. Η κατασκευή έχει τελειώσει και μπορεί, προφανώς να γίνει σε πολυωνυμικό χρόνο.

Απόδειξη: HC $\leq_m^p$ TSP (ii)

Έστω ότι ο $G(V, E)$ έχει κύκλο Hamilton. Τότε παίρνοντας αυτόν τον κύκλο σαν tour στον γράφο G', προφανώς περνά μία ακριβώς φορά από κάθε κόμβο και έχει συνολικό βάρος $B = |V'| = |V|$ εφόσον κάθε πλευρά έχει βάρος 1 (αφού ανήκε στον G).

Αντίστροφα, έστω ότι ο γράφος G' έχει κάποιο tour με συνολικό βάρος $\leq B = |V'| = |V|$. Αφού όμως το G' έχει $|V'|$ κόμβους, το tour θα περνά από $|V'|$ πλευρές και συνεπώς το συνολικό βάρος θα είναι ακριβώς $B = |V'|$. Αυτό όμως μπορεί να συμβεί μόνο όταν κάθε μία από τις $|V'|$ πλευρές έχει βάρος 1. Άρα όλες αυτές οι πλευρές ανήκουν στον G και συνεπώς ο G έχει κύκλο Hamilton (είναι το tour του G'). $\square$

Θεώρημα

Το πρόβλημα CLIQUE
είναι NP-complete

Απόδειξη: in NP + αναγωγή VC $^{p}_{m}$ CLIQUE

VC $^{p}_{m}$ CLIQUE

Μας δίνεται ένας γράφος $G(V, E)$ και ένας θετικός ακέραιος $k \leq |V|$. Θα κατασκευάσουμε έναν γράφο $G'(V', E')$ και έναν θετικό ακέραιο $j \leq |V'|$ έτσι ώστε ο γράφος G να έχει vertex cover μεγέθους $\leq k$ αν και μόνο αν ο γράφος G' έχει κλίκα μεγέθους $\geq j$. Η κατασκευή γίνεται ως εξής:

- Παίρνουμε σαν G' το συμπληρωματικό γράφο του G. Δηλαδή $V' = V$ και $E' = \{(u, v) \mid (u, v) \notin E\}$. Επίσης παίρνουμε $j = |V| - k = |V'| - k$. Η κατασκευή γίνεται προφανώς σε πολυωνυμικό χρόνο.

VC $\leq^p_m$ CLIQUE: απόδειξη "<=>"

Έστω ότι ο γράφος $G(V,E)$ έχει ένα σύνολο κόμβων $V_c \subseteq V$ με $|V_c| \leq k$ το οποίο καλύπτει όλες τις πλευρές του G. Αυτό σημαίνει ότι όλοι οι κόμβοι που ανήκουν στο $V - V_c$ είναι ανά δύο μη-γειτονικοί (διότι αν υπήρχε πλευρά που ένωνε δύο τέτοιους κόμβους, τότε αυτή η πλευρά δεν θα είχε καλυφθεί από το V_c). Άρα λοιπόν, όλοι αυτοί οι κόμβοι συνδέονται ανά δύο μεταξύ τους στον γράφο G'. Συνεπώς ο G' έχει κλίκα μεγέθους $|V| - |V_c| \geq |V| - k = j$. Αντίστροφα, αν ο γράφος $G'(V',E')$ έχει κλίκα μεγέθους $\geq j = |V'| - k$ τότε στον γράφο $G(V,E)$ θα υπάρχουν τουλάχιστον $|V'| - k = |V| - k$ κόμβοι που δεν θα συνδέονται μεταξύ τους. Συνεπώς οι υπόλοιποι (το πολύ k) κόμβοι θα φρουρούν υποχρεωτικά όλες τις ακμές του G. Άρα ο γράφος G έχει vertex cover μεγέθους $\leq k$. □

Θεώρημα

Το πρόβλημα SUBGRAPH ISOMORPHISM
είναι NP-complete

Απόδειξη: in NP + αναγωγή
CLIQUE $\ {}^{p}_{m}$ SUBGRAPH ISOMORPHISM

Απόδειξη. Το SUBGRAPH ISOMORPHISM ανήκει στο NP, διότι ένας μη-ντετερμινιστικός αλγόριθμος μπορεί να ελέγξει σε πολυωνυμικό χρόνο ότι η f που έχει μαντέψει, απεικονίζει το V_2 (βλ. ορισμό του προβλήματος) σε ένα υποσύνολο του V τέτοιο ώστε,$\forall u, v[(u, v) \in E_2 \iff (f(u), f(v)) \in E]$. Θα δείξουμε ότι το SUBGRAPH ISOMORPHISM είναι NP-complete ανάγοντας το CLIQUE σ' αυτό ($CLIQUE \leq_m^p SUBGRAPH\ ISOMORPHISM$).

Μας δίνεται ένας γράφος $G(V, E)$ και ένας θετικός ακέραιος $k \leq |V|$. Θα κατασκευάσουμε δύο γράφους $G_1(V_1, E_1)$ και $H(V_2, E_2)$ έτσι ώστε ο γράφος $G(V, E)$ να έχει κλίκα μεγέθους $\geq k$ αν και μόνο αν ο γράφος $G_1(V_1, E_1)$ έχει υπογράφο ισομορφικό με τον $H(V_2, E_2)$. Παίρνουμε σαν γράφο G_1 τον γράφο G και σαν H, ένα πλήρη γράφο με k κόμβους.

Έστω ότι ο γράφος G έχει κλίκα με $V_c = k$. Αυτό σημαίνει ότι ο G_1 έχει υπογράφο ισομορφικό με τον H (εξ ορισμού). Αντίστροφα, έστω ότι ο G_1 έχει υπογράφο ισομορφικό με τον H που είναι πλήρης γράφος με k κόμβους. Συνεπώς ο G έχει κλίκα μεγέθους k τουλάχιστον. $\square$

Θεώρημα

Το πρόβλημα PARTITION
είναι NP-complete

Απόδειξη: in NP + αναγωγή
3DM $\;^{p}_{m}$ PARTITION

3DM $\leq_m^p$ PARTITION

Μας δίνονται τρία ξένα μεταξύ τους σύνολα W, X, Y με $|W| = |X| = |Y|$ και ένα σύνολο $M \subseteq W \times X \times Y$. Θα κατασκευάσουμε ένα σύνολο A και κάποια συνάρτηση βάρους $w(a) \in Z^+, \forall a \in A$, έτσι ώστε το σύνολο M να έχει ένα matching $M' \subseteq M$ αν και μόνο αν το σύνολο A μπορεί να χωριστεί σε δύο ισοβαρή υποσύνολα. Η κατασκευή γίνεται ως εξής:

- Έστω

$$\begin{aligned} W &= \{w_1, w_2, \ldots, w_q\} \\ X &= \{x_1, x_2, \ldots, x_q\} \\ Y &= \{y_1, y_2, \ldots, y_q\} \\ M &= \{m_1, m_2, \ldots, m_q\}, \quad \text{όπου } m_i = (w_r, x_l, y_n) \end{aligned}$$

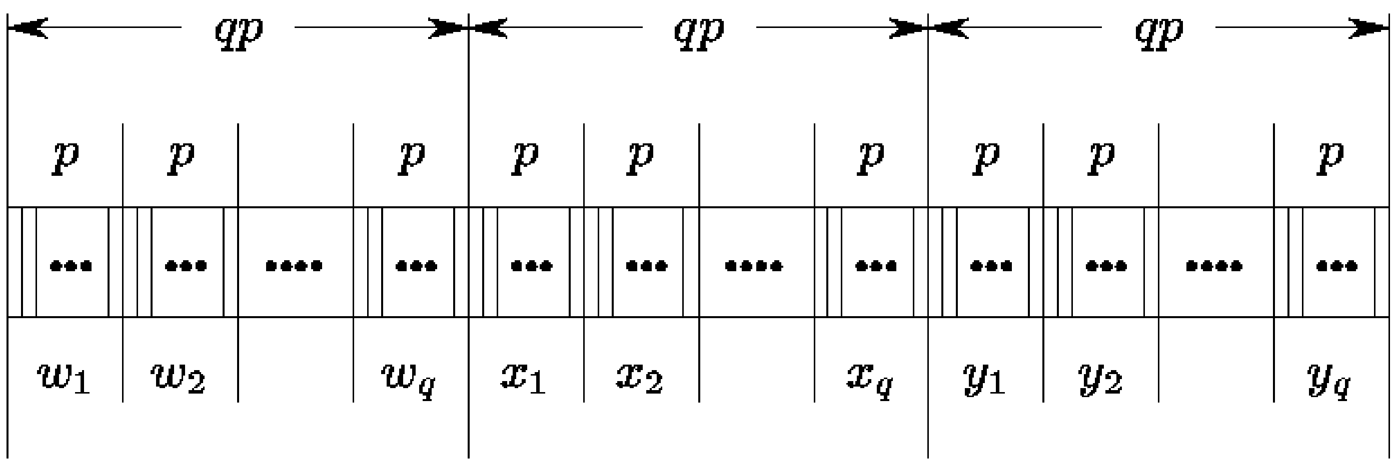
Βάζουμε στο σύνολο A, k στοιχεία $a_1, a_2, \ldots, a_k$ καθένα από τα οποία αντιστοιχεί σε κάποιο m_i. Το βάρος $w(a_i)$ θα οριστεί ως εξής:

3DM $_{m}^{p}$ PARTITION

– Αναπαριστούμε το κάθε $w(a_i)$ με μία ακολουθία δυαδικών ψηφίων. Η ακολουθία αυτή αποτελείται από $3q$ ζώνες και η κάθε ζώνη από $p = \lceil \log_2(k+1) \rceil$ δυαδικά ψηφία (σχήμα 14.19). Αν βάλουμε κάτω από τις ζώνες σαν labels τα

$$w_1, w_2, \ldots, w_q \qquad x_1, x_2, \ldots, x_q \qquad y_1, y_2, \ldots, y_q$$

όπως φαίνεται στο σχήμα 14.19, έχουμε μια αντιστοίχηση μεταξύ a_i και m_i.



3DM $\leq_m^p$ PARTITION

Αν $m_i = (w_r, x_l, y_n)$, τότε στις αντίστοιχες θέσεις w_r, x_l, y_n του $w(a_i)$ κάνουμε το δεξιότερο απ' όλα τα bits 1 και όλα τ' άλλα bits 0. Δηλαδή κάθε αναπαράσταση των $w(a_i)$ θα έχει τρεις άσσους, ένα σε κάθε ζώνη μήκους q, και όλα τ' άλλα ψηφία 0. Συνεπώς αν μετατρέψουμε αυτόν τον δυαδικό αριθμό σε δεκαδικό έχουμε:

$$w(a_i) = 2^{p(3q-r)} + 2^{p(2q-l)} + 2^{p(q-n)}$$

Θέλουμε λοιπόν, k τέτοιες ακολουθίες των $3qp$ bits. Συνεπώς μπορούμε να το κάνουμε σε πολυωνυμικό χρόνο.

3DM ${}^{p}_{m}$PARTITION

– Πριν προχωρήσουμε θα εξηγήσουμε γιατί επιλέχθηκε το p ίσο με $\lceil \log_2(k+1) \rceil$. Αν λύσουμε ως προς k έχουμε $2^p \geq k+1 \Rightarrow k \leq 2^p - 1$. Δηλαδή, ακόμα και αν όλα τα στοιχεία m_i είχαν μία κοινή συντεταγμένη, αν προσθέταμε όλα τα $w(a_i)$, η άθροιση στην ζώνη που αντιστοιχεί σ' αυτήν την κοινή τους συντεταγμένη θα είναι:

$$\begin{array}{cccccc} & & p & & & \\ 0 & 0 & \cdots & 0 & 1 & \\ & & \vdots & & & k \\ 0 & 0 & \cdots & 0 & 1 & \end{array}$$

Το άθροισμα, δηλαδή, σ' αυτή τη ζώνη θα είναι ο αριθμός $k \leq 2^p - 1$, ένας αριθμός ο οποίος μπορεί πάντα να παρασταθεί με p bits. Αυτό σημαίνει πως δεν είναι δυνατόν να έχουμε μεταφορές κρατουμένων από κάποια ζώνη στην επόμενη. Δηλαδή αν για παράδειγμα στις m_k τριάδες δεν περιλαμβάνεται καθόλου κάποια συντεταγμένη, έστω η w_r, τότε η ζώνη με label w_r θα έχει μόνο μηδενικά και δε θα μπορούσε να αποκτήσει ποτέ 1 αν αθροίζαμε όλα τα $w(a_i)$.

3DM $\leq_m^p$ PARTITION

– Ορίζουμε τον αριθμό B ως εξής:

$$B = 2^{3qp-p} + 2^{3qpp-2p} + \ldots + 2^0 = \sum_{j=0}^{3q-1} 2^{pj}$$

Δηλαδή, η δυαδική αναπαράσταση του B έχει έναν άσσο στη δεξιότερη θέση κάθε ζώνης p. Παρατηρούμε ότι σε ένα matching, κάθε συντεταγμένη θα λαμβάνεται ακριβώς μία φορά. Συνεπώς, όπως εύκολα μπορεί να διαπιστώσει κανείς,

$\exists A' \subseteq A\colon \sum_{a_i \in A'} w(a_i) = B \qquad \Longleftrightarrow$

$\exists M' \subseteq M : M'$ είναι matching του M με $M' = \{m_i \mid a_i \in A'\}$ $(*)$

3DM $\leq_m^p$ PARTITION

- Τέλος, προσθέτουμε δύο ακόμα στοιχεία στο A. Το στοιχείο b_1 με βάρος $w(b_1) = 2\sum_{i=1}^{k} w(a_i) - B$ και το στοιχείο b_2 με βάρος $w(b_2) = \sum_{i=1}^{k} w(a_i) + B$. Αυτά τα δύο αθροίσματα μπορούν να υπολογιστούν επίσης σε πολυωνυμικό χρόνο.

3DM $\leq_m^p$ PARTITION

Αυτό που μένει να αποδείξουμε είναι ότι το M έχει matching αν και μόνο αν το A μπορεί να χωριστεί σε δύο ισοβαρή σύνολα. Έστω ότι το M έχει ένα matching M'. Τότε λόγω της $(*)$, $\exists A' \subseteq A$: $\sum_{a_i \in A} w(a_i) = B$.

Όλο το βάρος του A είναι:

$$W_{tot} = \sum_{i=1}^{k} w(a_i) + w(b_1) + w(b_2) = 4\sum_{i=1}^{k} w(a_i).$$

Αν πάρουμε στο ένα σύνολο τα $a_i \in A'$ και το b_1 θα έχουμε βάρος

$$2\sum_{i=1}^{k} w(a_i) - B + B = 2\sum_{i=1}^{k} w(a_i).$$

Συνεπώς το άλλο σύνολο που περιέχει τα $a_i \in (A - A')$ και το b_2 θα έχει βάρος

$$W_{tot} - 2\sum_{i=1}^{k} w(a_i) = 2\sum_{i=1}^{k} w(a_i).$$

3DM $\leq_m^p$ PARTITION

Αντίστροφα, έστω ότι το A χωρίζεται σε δύο ισοβαρή σύνολα. Ο καθένας θα έχει βάρος $2\sum_{i=1}^{k} w(a_i)$. Συνεπώς δεν μπορεί στο ίδιο σύνολο να υπάρχουν

τα b_1, b_2. Το σύνολο στο οποίο θα ανήκει το b_1 θα περιέχει και κάποια $a_i \in A'$ έτσι ώστε:

$$w(b_1) + \sum_{a_i \in A'} w(a_i) = 2\sum_{i=1}^{k} w(a_i) \Rightarrow \sum_{a_i \in A'} w(a_i) = B$$

Όμως, λόγω της $(*)$, αυτό σημαίνει ότι $\exists M' \subseteq M : M' = \{m_i \mid a_i \in A'\}$ και M' είναι matching. $\square$

Θεώρημα

Το πρόβλημα DISCRETE KNAPSACK
είναι NP-complete

Απόδειξη: in NP + αναγωγή
PARTITION $\leq^p_m$ DISCRETE KNAPSACK

PARTITION $\leq^p_m$ DISCRETE KNAPSACK

Μας δίνεται ένα σύνολο A και μία συνάρτηση βάρους $w(a_i), \forall a_i \in A$. Θα κατασκευάσουμε ένα σύνολο U, μια συνάρτηση βάρους $w'(u_i), \forall u_i \in U$, μια συνάρτηση κόστους $p(u_i), \forall u_i \in U$ και δύο θετικούς ακέραιους W, P έτσι ώστε το σύνολο A να χωρίζεται σε δύο ισοβαρή σύνολα αν και μόνο αν υπάρχει, $U' \subseteq U : \sum_{u \in U'} w'(u) \leq W,\ \sum_{u \in U'} p(u) \geq p$. Η κατασκευή γίνεται ως εξής:

- Σαν σύνολο U παίρνουμε το σύνολο A
- Σαν $w'(u), \forall u \in U$ παίρνουμε το $w(a), \forall a \in A$
- Σαν $p(u), \forall u \in U$ παίρνουμε το $w(a), \forall a \in A$
- Δηλαδή, παίρνουμε το ίδιο σύνολο όπως έχει με τα ίδια βάρη και το κόστος κάθε αντικειμένου το παίρνουμε ίσο με το βάρος του. Τέλος, τους αριθμούς W, P τους παίρνουμε ως εξής:

$$W = P = \frac{1}{2} \sum_{a \in A} w(a) = \frac{1}{2} \sum_{u \in U} w'(u)$$

PARTITION $\leq^p_m$ DISCRETE KNAPSACK: απόδειξη "<=>"

Έστω λοιπόν, ότι το σύνολο A χωρίζεται σε δύο ισοβαρή υποσύνολα, στον A' και στον $A - A'$. Επειδή το ολικό βάρος του A είναι $\sum_{a \in A} w(a)$, σημαίνει ότι το βάρος του υποσυνόλου A', για παράδειγμα, είναι $\frac{1}{2} \sum_{a \in A} w(a)$. Άρα αν πάρουμε σαν U' το σύνολο A' έχουμε, $\sum_{u \in U'} w'(u) = W \leq W$ και $\sum_{u \in U'} p(u) = P \geq P$

Αντίστροφα, έστω ότι $\exists U' \subseteq U$: $\sum_{u \in U'} w'(u) \leq W$ και $\sum_{u \in U'} p(u) \geq p$. Όμως $\sum_{u \in U'} w'(u) = \sum_{u \in U'} p(u) = B = P = \frac{1}{2} \sum_{u \in U} w'(u)$. Συνεπώς, αν πάρουμε σαν A' το U', τότε το A χωρίζεται σε δύο ισοβαρή υποσύνολα, A' και $A - A'$. $\square$

Κλάσεις Πολυπλοκότητας

- $P = PTIME = \bigcup_{i \geq 1} DTIME(n^i)$
- $NP = NPTIME = \bigcup_{i \geq 1} NTIME(n^i)$
- $PSPACE = \bigcup_{i \geq 1} DSPACE(n^i)$
- $NPSPACE = \bigcup_{i \geq 1} NSPACE(n^i)$
- $L = DSPACE(\log n)$
- $NL = NSPACE(\log n)$

Κλάσεις Πολυπλοκότητας

Αν η f είναι συνάρτηση πολυπλοκότητας:

- $DSPACE(f(n)) \subseteq NPSPACE(f(n))$
- $DTIME(f(n)) \subseteq NTIME(f(n))$
- $DTIME(f(n)) \subseteq DSPACE(f(n))$
- $NTIME(f(n)) \subseteq DSPACE(f(n))$
- $NSPACE(f(n)) \subseteq DTIME(k^{\log n + f(n)})$

Κλάσεις Πολυπλοκότητας

Αν f(n) > logn:

- $DSPACE(f(n)) \subseteq DTIME(c^{f(n)})$
- $NTIME(f(n)) \subseteq DTIME(c^{f(n)})$
- $NSPACE(f(n)) \subseteq DSPACE(f^2(n))$ [Savitch]
  συνεπώς $PSPACE = NPSPACE$
- $L \subset PSPACE$

Τι (δεν) ξέρουμε

- Γνωρίζουμε ότι:

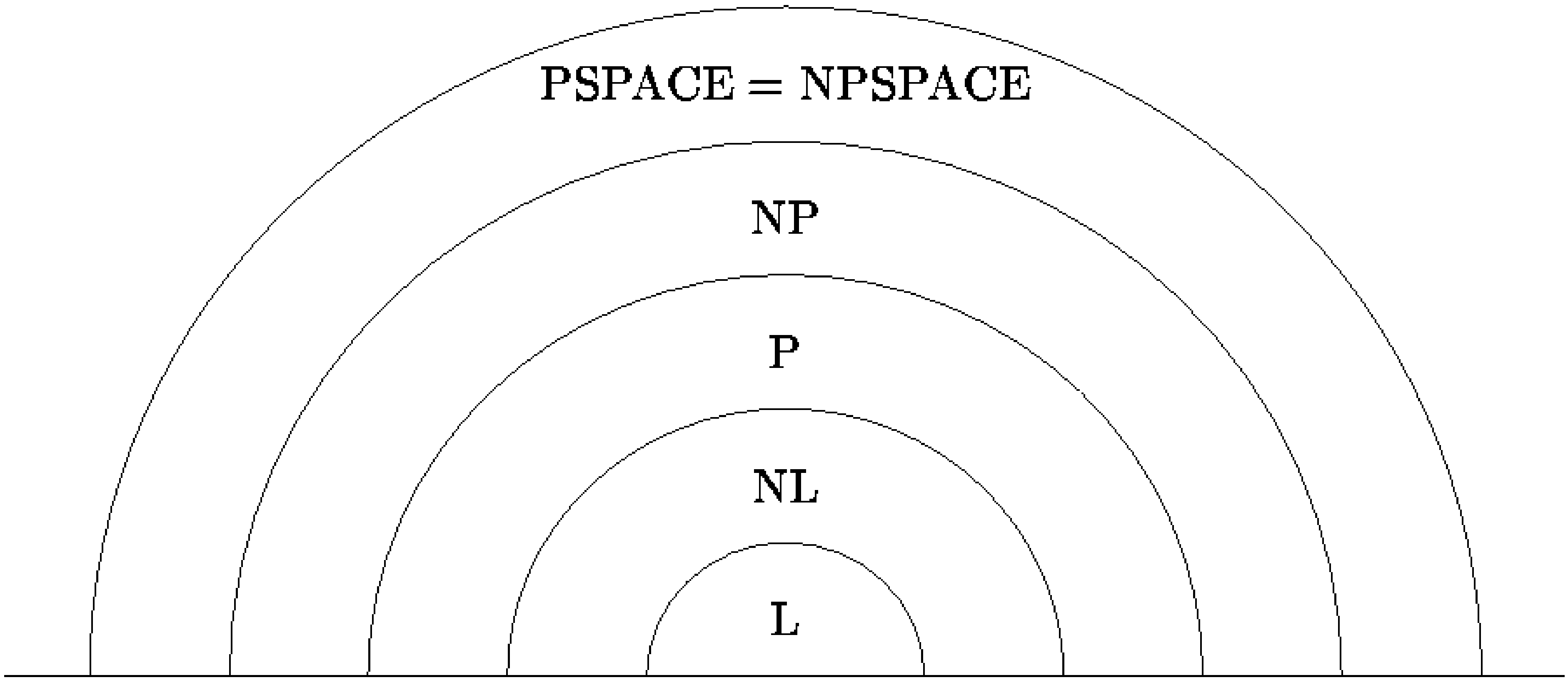
$$L \subseteq NL \subseteq P \subseteq NP \subseteq PSPACE = NPSPACE$$

$$L \neq PSPACE \text{ και } NL \neq PSPACE$$

- Δεν γνωρίζουμε αν:

$$L = NL = P = NP = PSPACE$$

Τι πιστεύουμε



Καλή επιτυχία στις εξετάσεις!!!