Αλγόριθμοι και Πολυπλοκότητα

Το Θεώρημα του Cook, Μετασχηματισμοί Προβλημάτων, NP-complete Προβλήματα, Κλάσεις Πολυπλοκότητας

Στάθης Ζάχος

Hamilton Circuit $\leq_m^p$ 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 $\leq_m^p$ 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'). □

Ορισμός: CNF formula

Ορισμός 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 *Κάθε λογική έκφραση είναι ισοδύναμη με μία σε CNF.*

Ορισμός: SAT

Ορισμός **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 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 $\Phi(x)$

Λογικές μεταβλητές της $\Phi(x)$

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

- $H[i,j],\ 0 \leq i \leq p(n),\ -p(n) \leq j \leq p(n)$ 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 $(p(n)+1)\cdot(2p(n)+1)$.

- $S[i,j,l],\ 0 \leq i \leq p(n),\ -p(n) \leq j \leq p(n),\ 0 \leq l \leq v$ 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 $\Phi(x)$

Clauses της $\Phi(x)$: 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:

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

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:

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

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 $\Phi(x)$

Clauses της $\Phi(x)$: 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:

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

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 Τ.Μ. βρίσκεται στο αρχικό configuration. Δηλαδή: Η κατάσταση στην οποία βρίσκεται η Τ.Μ. 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 litetal που δηλώνει, ότι τη χρονική στιγμή $p(n)$, η κατάσταση στην οποία βρίσκεται η Τ.Μ. 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 $\Phi(x)$

Clauses της $\Phi(x)$: 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:

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

Θεώρημα Cook: Κατασκευή $\Phi(x)$

Clauses της $\Phi(x)$: group G_6 (iii)

Αυτό σημαίνει πως αν τη χρονική στιγμή i η Τ.Μ. 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 Τ.Μ. 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 $\Phi(x)$

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
Ερώτηση: Πως μπορούμε να ανταλλάξουμε τις θέσεις των λευκών με τα μαύρα αλογάκια χρησιμοποιώντας νόμιμες κινήσεις;

X		X
O		O

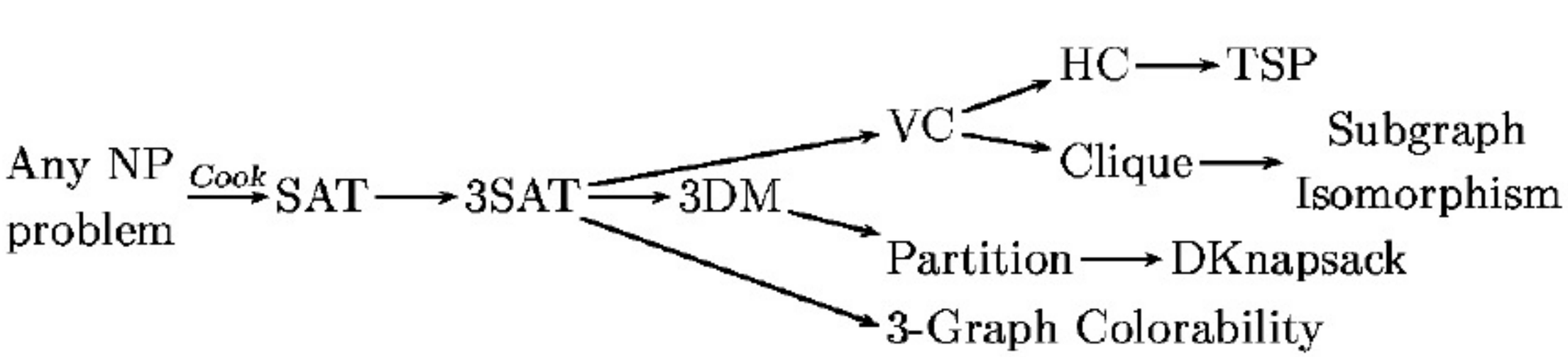
Απόδειξη NP-completeness

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 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 $\forall \{u, v\} \in E : \ u \in V' \vee v \in V'$;

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 $f : V \to \{1, 2, 3\}$ 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 $f : V_2 \to V$, «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 $\leq_m^p$ 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 $\leq_m^p$ 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 $\leq_m^p$ 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.