Κεφάλαιο 12

Μετασχηματισμοί προβλημάτων (transformations of problems)

12.1 Γενικά

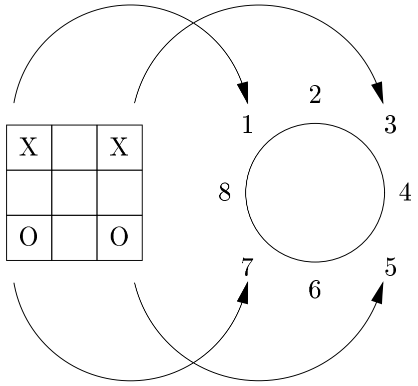
Πολλές φορές έχουμε να αντιμετωπίσουμε προβλήματα των οποίων η λύση, αν και απλή, δεν είναι προφανής. Τότε μετασχηματίζουμε το αρχικό πρόβλημα σε ένα άλλο πρόβλημα του οποίου μπορούμε εύκολα να βρούμε τη λύση. Έπειτα (με βάση το μετασχηματισμό) μεταφέρουμε τη λύση στο αρχικό μας πρόβλημα.

Παράδειγμα: Έστω το εξής πρόβλημα:
Δεδομένα: Μια σκακιέρα 3×3 με μαύρα (X) και λευκά (O) αλογάκια όπως φαίνεται στο σχήμα 12.1.
Ερώτηση: Πως μπορούμε να ανταλλάξουμε τις θέσεις των λευκών με τα μαύρα αλογάκια χρησιμοποιώντας νόμιμες κινήσεις;

X		X
O		O

Σχήμα 12.1: Πρόβλημα ίππων επί σκακιέρας

Λύση: Φτιάχνουμε έναν δακτύλιο όπως φαίνεται στο σχήμα 12.2. Τα μαύρα αλογάκια βρίσκονται στις θέσεις 1, 3 και τα λευκά στις θέσεις 5, 7. Η λύση του προβλήματος τώρα είναι προφανής. Με συνεχείς διαδοχικές κυκλικές μεταθέσεις, τα μαύρα αλογάκια θα ανταλλάξουν τελικά θέσεις με τα λευκά. Δηλαδή εκτελώντας μία περιστροφή κατά τη φορά των δεικτών του ρολογιού, το άλογο που βρίσκεται στη θέση 1 θα μετακινηθεί στη θέση 2, εκείνο που βρίσκεται στη



Σχήμα 12.2: Μετασχηματισμός προβλήματος ίππων επί σκακιέρας

θέση 3 θα πάει στη θέση 4, κ.ο.κ, έως ότου μετά από 4 μεταθέσεις καθενός αλόγου, τα μαύρα ανταλλάξουν θέσεις με τα λευκά.

Έτσι λοιπόν, αν ξαναγυρίσουμε στο αρχικό πρόβλημα, το μόνο που απομένει είναι να αριθμήσουμε τη σκακιέρα με βάση τις νόμιμες κινήσεις των αλόγων. Αυτό φαίνεται στο σχήμα 12.3. Δηλαδή η πορεία που θα ακολουθήσει το άλογο 1 θα είναι $1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5$ ενώ παράλληλα το άλογο 3 θα ακολουθεί την πορεία $3 \rightarrow 4 \rightarrow 5 \rightarrow 6 \rightarrow 7$, κ.ο.κ.

1	6	3
4		8
7	2	5

Σχήμα 12.3: Αρίθμηση σκακιέρας 3×3

Εδώ χρησιμοποιούμε τη μέθοδο της αναγωγής για να δείξουμε ότι ένα νέο πρόβλημα είναι δυσεπίλυτο: Ανάγουμε με εύκολο μετασχηματισμό ένα γνωστό δυσεπίλυτο πρόβλημα στο νέο μας πρόβλημα και έτσι δείχνουμε τη δυσεπιλυσιμότητά του (ειδάλλως αυτή η αναγωγή θα οδηγούσε σε εύκολη λύση του όμως γνωστού δυσεπίλυτου προβλήματος).

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 $x \in \Pi' \iff f(x) \in \Pi$.

Οι αναγωγές προβλημάτων που θα δούμε στη συνέχεια, έγιναν με τη σειρά που φαίνεται στο σχήμα 12.4 και ιστορικά, οι περισσότερες από αυτές παρουσιάστηκαν από τον *Karp* (1972).

$$
\begin{tikzcd}
 & & & & & \text{HC} \arrow[r] & \text{TSP} \\
 & & & & \text{VC} \arrow[ur] \arrow[dr] & & \\
\text{Any NP problem} \arrow[r, "Cook"] & \text{SAT} \arrow[r] & \text{3SAT} \arrow[urr] \arrow[r] \arrow[ddrr] & \text{3DM} \arrow[dr] & & \text{Clique} \arrow[r] & \text{Subgraph Isomorphism} \\
 & & & & \text{Partition} \arrow[r] & \text{DKnapsack} & \\
 & & & & \text{3-Graph Colorability} & &
\end{tikzcd}
$$

Σχήμα 12.4: Αναγωγές προβλημάτων στην κλάση NP

Πριν προχωρήσουμε στις αναγωγές, θα ορίσουμε όλα τα προβλήματα απόφασης που αναφέρονται στο διάγραμμα.

12.2 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;

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

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, $\forall (u, v) \in E : f(u) \neq f(v)$;

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:

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

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 $\geq j$;

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,

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

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,

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

Από τα παραπάνω προβλήματα έχουμε ήδη αποδείξει πως το SAT είναι NP-complete.

12.3 Αναγωγή του SAT στο 3SAT

Θεώρημα 12.3.1. *Το 3SAT είναι* **NP-complete**.

Proof. 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:

- 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 $(z_1 \vee z_2)$.

- 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):

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

Οι y_i είναι dummy μεταβλητές, ενώ ο αριθμοί των καινούργιων clauses είναι $k-2$. Η κατασκευή της καινούργιας φόρμουλας Φ', η οποία είναι στιγμιότυπο του 3SAT έχει τελειώσει.

Αν το μήκος της αρχικής φόρμουλας Φ (μέγεθος του input) είναι $n \cdot m$ (m clauses με n το πολύ literals το καθένα), τότε το μήκος της φόρμουλας Φ', θα είναι $3m(n-2)$, δηλαδή $m(n-2)$ clauses με 3 literals το καθένα. Δηλαδή το μήκος Φ' θα είναι $O(m \cdot n)$, πολυωνυμικό ως προς το μήκος της Φ και συνεπώς ο μετασχηματισμός γίνεται σε πολυωνυμικό χρόνο. Αυτό που μένει να αποδείξουμε λοιπόν, είναι ότι:«η αρχική φόρμουλα Φ έχει απονομή αλήθειας που την ικανοποιεί», ανν η νέα φόρμουλα Φ' έχει απονομή αληθείας που την ικανοποιεί.

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. □

Παρατήρηση 12.3.2. Το πρόβλημα 2SAT ανήκει στο P. Θα μπορούσαμε να μετασχηματίσουμε το SAT, ή το ισοδύναμο του πλέον 3SAT στο 2SAT. Όμως ο αριθμός των clauses της καινούριας φόρμουλας που θα προκύψει, θα είναι εκθετικός ως προς την αρχική φόρμουλα.

Ένα άλλο υποπρόβλημα του SAT, το οποίο ανήκει στο P, είναι το *HORN-SAT*. Η δεδομένη boolean formula αποτελείται από *Horn clauses*. *Horn clause* ονομάζεται ένα clause το οποίο έχει το πολύ ένα θετικό literal. Δηλαδή, όλα τα literals εκτός πιθανόν από ένα, είναι αρνητικά, π.χ.:

$$(\neg x_2 \vee x_3), \quad (\neg x_1 \vee \neg x_2 \vee \neg x_3), \quad (\neg x_1 \vee \neg x_2 \vee \neg x_3 \vee x_4)$$

Τα clauses που έχουν ακριβώς ένα θετικό literal (όπως το πρώτο και το τελευταίο παράδειγμα), ονομάζονται *implications* διότι μπορούν να γραφτούν ως εξής: $(\neg x_2 \vee x_3) = (x_2 \rightarrow x_3)$, $(\neg x_1 \vee \neg x_2 \vee \neg x_3 \vee x_4) = ((x_1 \wedge x_2 \wedge x_3) \rightarrow x_4)$. Ο αλγόριθμος που λύνει το **HORNSAT** βασίζεται σ' αυτήν ακριβώς την *implicational* μορφή των clauses.

12.4 Αναγωγή του 3SAT 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.

12.5 Η αναγωγή του 3SAT στο VERTEX COVER

Θεώρημα 12.5.1. *Το πρόβλημα VERTEX COVER (VC) είναι NP-complete.*

Proof. Ισχύει $VC \in NP$, διότι ένας μη-ντετερμινιστικός αλγόριθμος μαντεύει ένα σύνολο κόμβων V' και μετά ελέγχει αν το V' περιέχει τουλάχιστον ένα endpoind από κάθε πλευρά και αν $|V'| \leq k$ σε πολυωνυμικό χρόνο. Θα δείξουμε ότι το VC είναι NP-complete ανάγοντας το 3SAT σ' αυτό ($3SAT \leq_m^p VC$).

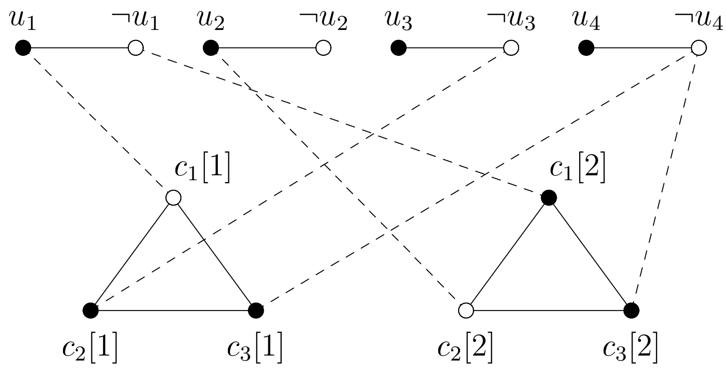
Μας δίνεται ένα σύνολο από clauses $C = \{c_1, \ldots, c_m\}$ που περιέχει μεταβλητές από ένα σύνολο μεταβλητών $U = \{u_1, \ldots, u_n\}$. Θα κατασκευάσουμε έναν γράφο $G(V, E)$ και έναν θετικό ακέραιο $k \leq |V|$ τέτοια ώστε ο G να έχει vertex cover μεγέθους $\leq k$ αν και μόνο αν η φόρμουλα $(c_1 \wedge c_2 \wedge \ldots \wedge c_m)$ είναι ικανοποιήσιμη. Η κατασκευή γίνεται ως εξής:

- 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.

Το πλήθος των κόμβων του γράφου είναι $|V| = 2n + 3m$, ενώ το πλήθος των πλευρών είναι $|E| = n + 6m$. Το μέγεθος του γράφου είναι πολυωνυμικό ως προς το μέγεθος της φόρμουλας ($3m$) και η κατασκευή μπορεί να γίνει σε πολυωνυμικό χρόνο. Ορίζουμε $k = n + 2m$. Υποστηρίζουμε πως η φόρμουλα του προβλήματος 3SAT ικανοποιείται αν και μόνο αν υπάρχει vertex cover βαθμού $\leq k$ στο γράφο $G(V, E)$. Πριν προχωρήσουμε στην απόδειξη του ισχυρισμού, δίνουμε ένα παράδειγμα.



Σχήμα 12.5: Αναγωγή 3SAT στο VERTEX COVER

Παράδειγμα: Έστω ότι έχουμε τη φόρμουλα

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

Η Φ είναι ικανοποιήσιμη. Μια απονομή αλήθειας που την ικανοποιεί είναι η $t(u_1, u_2, u_3, u_4) = (T, T, T, T)$. Συνεπώς ο γράφος $G(V, E)$ που προκύπτει απ' αυτήν τη φόρμουλα και φαίνεται στο σχήμα 12.5, πρέπει να έχει vertex cover μεγέθους $\leq n + 2m = 4 + 2 \cdot 2 = 8$.

Οι 8 κόμβοι που αποτελούν ένα vertex cover του γράφου είναι οι κόμβοι που σημειώνονται με κουκκίδες.

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.

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. □

12.6 Η αναγωγή του 3SAT στο 3-DIMENSIONAL MATCHING

Θεώρημα 12.6.1. *Το πρόβλημα 3-DIMENSIONAL MATCHING είναι NP-complete.*

Proof. Ισχύει 3DM $\in NP$, διότι ένας μη-ντετερμινιστικός αλγόριθμος μαντεύει ένα σύνολο $M' \subseteq M = W \times X \times Y$ από q τριάδες και μετά ελέγχει σε πολυωνυμικό χρόνο ότι ανά 2 δεν έχουν κοινές συντεταγμένες. Θα δείξουμε ότι το 3DM είναι NP-complete ανάγοντας το 3SAT σ' αυτό (3SAT $\leq_m^p$ 3DM).