NP-Completeness

1 Polynomial time reductions

Problem X polynomially reduces to problem Y if an arbitrary instance of problem X can be

- Transformed to an instance of problem Y in a polynomial number of steps and then
- Solved using a polynomial number of calls to an oracle that solve problem Y

Notation: X is polynomial time reducible to Y is written as $X \leqslant Y$

If $X \leqslant Y$ and $Y \leqslant X$, then X and Y are equivalent

2 NP completeness proofs

To prove that a problem Π is NP complete we now have to perform two steps

1. Show that Π belongs to NP
2. Find a known NP complete problem and show $\Pi' \leqslant \Pi$

If we can complete step 2 but not step 1, then we say that Π is NP hard

3 Proof techniques

Restriction

- Show that Π′ is a subproblem of Π

Local replacement

- Show that every basic unit in an instance of Π′ can be replaced by a different structure in a uniform way to obtain an instance of Π

Component design

- Show that the constituents of an instance of Π can be used to ”design” components that can be combined to ”realise” instanced of Π′

4 NP completeness of vertex cover

Problem: Vertex cover

Instance - A graph G=(V,E), and a natural number k
Question - Is there a set $W \subseteq V$, which $|W| \leqslant k$, such that for each edge $(i, j) \in E$

$$\{i, j\} \cap W \neq \emptyset$$

To show that Vertex Cover is NP complete we shall reduce satisfiability to vertex cover

Given a formula f with n variables and clauses $C_1, C_2, ..., C_m$

- For each variable x create two adjacent vertices x^t and x^f to represent the literals x and $\neg x$
- For each clause C_j of size n_j create a complete subgraph G_j with vertices connected to corresponding literals
- Set $k = n + \sum_{j=1}^{m}(n_j - 1)$

Theorem 1 *There exists a truth assignment that satisfies the formula f iff there exists a vertex cover of the constructed graph with size at most k*

Proof in ⇒ direction

- At least one of each pair (x^f, x^t) must be in the cover
- At least $n_j - 1$ vertices from each complete graph G_j must be in the cover
- If the formula is satisfiable, then choose the cover by choosing each literal assigned True plus all but one vertex in each G_j (omit a vertex which is connected to a satisfied literal)

Proof in $\Leftarrow$ direction

- If a vertex cover exists, assign each boolean variable according to whether x^t or x^f is in M
- By the choice of k, there must be exactly one vertex in each clique which is not in M. This vertex must be adjacent to a literal-vertex in M, hence clause is satisfied.

5 NP completeness of clique

Problem: Clique

Instance - A finite graph $G = (V, E)$ and an integer k
Question - Does G have a clique of size k?

A set of vertices W is a vertex cover in G iff V-W is a clique in the complement of G

6 NP completeness of Hitting Set

Problem: Hitting Set

Instance - Collection C of subsets of a set S and a positive integer k
Question - Does S contain a hitting set for C of size k or less. i.e. a subset $S' \subseteq S$ with $|S'| \leqslant k$ such that S' contains at least one element from each subset from C

To show that Hitting Set is NP complete restrict it to instances with $|c| = 2$ for all $c \in C$ and you get vertex cover

7 3 Satisfiability

To show that 3 satisfiability is NP complete we reduce Satisfiability to 3-Satisfiability

Proof
Replace every clause

$$C = x_1 \vee x_2 \vee ... \vee x_k$$

with $k > 3$ by:

$$C' = (x_1 \vee x_2 \vee y_1) \wedge (\neg y_1 \vee x_3 \vee y_2) \vee ... \vee (\neg y_{k-3} \vee x_{k-1} \vee x_k)$$

C is satisfiable iff C′ is, since at least one of the literals other than the y's must be true

8 NP completeness of Graph 3 colouring

Problem: Graph 3-Colouring

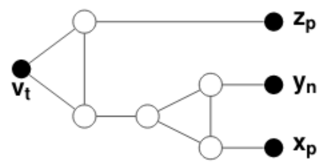
Instance - A graph G=(V,E)
Question - Is there a colouring of the vertices of G in 3 colours such that adjacent vertices are all different colours

To show completeness, reduce 3 satisfiability to graph 3 colouring

9 Encoding

Given a 3CNF formula f, encode it as a graph G_f such that the graph has a (proper) 3 colouring iff the formula is satisfiable

- Lets introduce 3 colours: ground, true and false
- Introduce a 3-clique of designated vertices, v_g, v_t and v_f. By symmetry, assume without loss of generality that they are always coloured ground, true and false, respectively
- For each variable x, introduce two vertices x_p and x_n (for x and $\neg x$), and add all edges between x_p, x_n and v_g. Hence one of x_p and x_n must be true and the other false
- For each clause C, say $C = (x \vee \neg y \vee z)$ connect vertices x_p,y_n and z_p via the below gadget consisting of five clauses vertices which we call the C-vertices



9.1 Proof

First suppose that G_f has a 3-colouring c

- As v_g, v_t v_f form a triangle, we may assume without loss of generality that $c(v_g) - 1$, $c(v_t) = 2$ and $c(v_f) = 3$
- For every variable x, $c(x_p) \in \{2, 3\}$ and $c(x_n) \in \{2, 3\}$, as x_p and x_n are both adjacent to v_g
- As x_p and x_n are adjacent, this means that if $c(x_p) = 2$ then $c(x_n) = 3$ and vice versa
- Let τ be the truth assignment of f that set x to be True if $c(x_p) = 2$ and false if it equals 3
- We claim that τ is satisfying. Suppose not and C is a clause whose three literals ℓ_1, ℓ_2 and ℓ_3 are all false.
  - Without loss of generality, let $C = (x \vee \neg y \vee z)$
  - Then $c(x_p) = c(y_n) = c(z_p) = 3$
  - As $c(v_t) = 1$, the C-vertex adjacent to z_p has colour 2, which means that it's C-neighbour has colour 3
  - But now as $c(x_p) = c(y_n) = 3$, the tree remaining C-vertices all have colour 1 or 2. As they form a triangle this is a contradiction as c is a 3-colouring of G_f

Now suppose that f has a satisfying truth assignment τ. We define a mapping c as follows

- Set $c(v_g) = 1$, $c(v_t) = 2$ and $c(v_f) = 3$
- If x is True, then set $c(x_p) = 2$ and $c(x_n) = 3$ and vice versa for False
- For every clause C do as follows
  - Without loss of generality, let $C = (x \vee \neg y \vee z)$ As τ is satisfying, at least one of $x, \neg y$ or z is True.
  - This means that at least one of x_p, y_n, z_p gets colour 2. Then we can colour the five C-vertices with colours 1,2,3 in such a way that no two adjacent vertices are coloured alike
- From above we conclude that c is a 3-colouring of G_f