NP-completeness

Complexity Theory

Faculty of Information Technology
Brno University of Technology
Brno, Czech Republic

Ondřej Lengál

This material was created with the support of the Czech Ministry of Education, Youth and Sports (project FRVŠ 166/2013/G1).

Completeness

The concept of completeness is one of the most important in complexity theory.

Definition (Hardness, Completeness)

Let **C** be a complexity class. We call a language L

C-hard if for all $L' \in \mathbf{C}$, $L' \leq L$,

C-complete if L is **C**-hard and $L \in \mathbf{C}$.

Note: We use $L_1 \leq L_2$ to denote that there exists a polynomial reduction from L_1 to L_2, i.e. that there exists a **PTIME** Turing Machine computing a function $R : \Sigma^* \to \Sigma^*$ s.t. $w \in L_1 \iff R(w) \in L_2$.

This means that **C**-complete problems are the hardest problems of **C**.

Motivation

Proving that a problem A is **NP**-complete means that:

- there is probably no fast algorithm for solving A,
- naïve ways for solving A will probably not work,
- heuristics may be necessary for practical algorithms,
- or we may just try to find an approximate solution,
- Richard M. Karp. *Reducibility Among Combinatorial Problems.*

SAT

SAT: Is a given propositional formula ψ satisfiable?

Theorem (Cook-Levin)

SAT *is* **NP***-complete.*

Proof.

- SAT $\in$ **NP** — by constructing an **NPTIME** TM accepting SAT.
- SAT is **NP**-hard — by showing that for any **NPTIME** TM M and its input w, there is a **PTIME** reduction to a propositional formula ψ s.t. ψ is satisfiable iff $w \in L(M)$. □

CNF

CNF: Is a given propositional formula φ in the conjunctive normal form satisfiable?

Theorem

CNF *is* **NP***-complete.*

Proof.

- CNF is **NP**-hard — from SAT using Tseitin transformation
  - transforms ψ into an equisatisfiable formula φ in CNF,
  - the size of φ grows linearly with the size of ψ,
  - naïve transformation (using De Morgan's laws and distribution) yields exponentially larger formula in the worst case. □

Note: in practice, "SAT" is often used to mean "CNF".

k-CNF (k-SAT)

k-CNF: A restricted version of CNF where each clause has exactly k literals.

Theorem

2CNF $\in$ **P**.

Proof.

Clauses can be rewritten to implications which can be viewed as Horn clauses. There is a **PTIME** algorithm for solving HORNSAT. □

k-CNF (k-SAT)

Theorem

*k-CNF is **NP**-complete for $k \geq 3$.*

Proof.

- 3-CNF is **NP**-hard — by reduction from CNF (similarly for other k). We can transform every clause

$$(a \vee b \vee c \vee \cdots \vee f \vee g)$$

into the conjunction

$$(a \vee b \vee x) \wedge (\neg x \vee c \vee y) \wedge \cdots \wedge (\neg z \vee f \vee g)$$

which is equisatisfiable and only linearly larger. □

3-CNF (3-SAT) is interesting because it is the variant of k-CNF with the lowest k that is **NP**-complete.

CLIQUE

CLIQUE: Given a graph $G = (V, E)$ and $k \in \mathbb{N}$, does G contain a clique (a complete subgraph) of size k?

Theorem

CLIQUE *is* **NP**-*complete.*

Proof.

- CLIQUE is **NP**-hard — by reduction from CNF. For a formula $C_1 \wedge \cdots \wedge C_n$ we set $k = n$ and construct an undirected graph $G = (V, E)$ such that

$$V = \{(\sigma, i) \mid \sigma \text{ is a literal and occurs in } C_i\}$$
$$E = \{\{(\sigma, i), (\delta, j)\} \mid i \neq j \wedge \sigma \neq \neg\delta\}$$

□

INDEPENDENT SET

INDEPENDENT SET: Given a graph $B = (W, J)$ and $m \in \mathbb{N}$, does B contain an independent set of vertices (a set of vertices no two of which are adjacent) of size at least m?

Theorem

INDEPENDENT SET *is* **NP**-*complete.*

Proof.

- INDEPENDENT SET is **NP**-hard — by reduction from CLIQUE. For a graph $G = (V, E)$ and k, we set $m = k$ and construct

$$B = (V, V^2 \setminus E)$$

□

Note that cliques are independent sets in graphs' complements.

VERTEX COVER

VERTEX COVER: Given a graph $H = (U, F)$ and $l \in \mathbb{N}$, does H have a vertex cover of size at most l? I.e., is there a set of vertices $S \subseteq U$ of size $|S| \leq l$ such that all edges of H are incident with at least one vertex from S?

Theorem

VERTEX COVER *is* **NP***-complete.*

Proof.

- VERTEX COVER is **NP**-hard — by reduction from INDEPENDENT SET. For a graph $B = (W, J)$ and m, we set $l = |W| - m$ and $H = B$. □

Note that a set is independent iff its complement is a vertex cover.

GRAPH COLOURING

GRAPH COLOURING: Given a graph $M = (Y, L)$ and $p \in \mathbb{N}$, can the vertices of M be coloured using p colours such that no two adjacent vertices are assigned the same colour?

Theorem

GRAPH COLOURING $\in$ **P** *for* $p = 2$.

Proof.

- A graph is 2-colourable iff it is bipartite, which can be determined using BFS in linear time. □

GRAPH COLOURING

Theorem

GRAPH COLOURING *is* **NP***-complete for* $p \geq 3$*.*

Proof.

- GRAPH COLOURING for $p \geq 3$ is **NP**-hard — by reduction from 3-CNF. For a formula $\varphi_1 \wedge \cdots \wedge \varphi_k$ over variables $x_1, \ldots, x_r$, we set $p = r + 1$ and construct the graph $M = (Y, L)$ in the following way: *Assume the formula*

$$(x_1 \vee x_2 \vee x_3) \wedge (x_1 \vee \neg x_2 \vee \neg x_3) \wedge (\neg x_1 \vee x_2)$$

GRAPH COLOURING

Theorem

GRAPH COLOURING *is* **NP***-complete for* $p \geq 3$*.*

Proof.

- GRAPH COLOURING for $p \geq 3$ is **NP**-hard — by reduction from 3-CNF. For a formula $\varphi_1 \wedge \dots \wedge \varphi_k$ over variables $x_1, \dots, x_r$, we set $p = r + 1$ and construct the graph $M = (Y, L)$ in the following way: *Assume the formula*

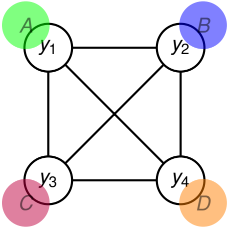
$$(x_1 \vee x_2 \vee x_3) \wedge (x_1 \vee \neg x_2 \vee \neg x_3) \wedge (\neg x_1 \vee x_2)$$

1. Make sure there are at least 4 variables ($r \geq 4$), otherwise add.
   - *we add* x_4 *to the set of variables* $\rightarrow \{x_1, x_2, x_3, x_4\}$*, and*
   - *set the number of colours* $p = 5$*, call them* $\{A, B, C, D, E\}$*.*

GRAPH COLOURING

Proof (cont).

2. Create a clique with a node y_i for every variable x_i.

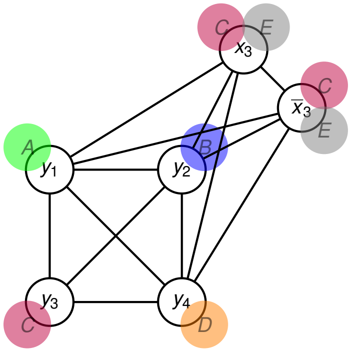


Each node of the clique needs to be coloured with a different colour.

GRAPH COLOURING

Proof (cont).

3. For every variable x_i, add nodes labelled with x_i and $\overline{x_i}$ and connect them with each other and with all y_j, $i \neq j$, from the clique.



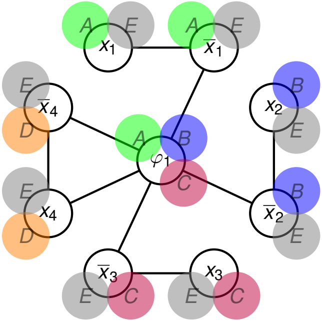
The node x_3 is coloured either by C (which stands for x_3 = *true*) or by E (for x_3 = *false*). The node $\overline{x}_3$ is coloured with the opposite colour.

GRAPH COLOURING

Proof (cont).

4. Add a node for every clause φ_i. For every x_j, connect φ_i with x_j if $x_j \notin \varphi_i$, and with $\overline{x}_j$ if $\neg x_j \notin \varphi_i$.

$$\underbrace{(x_1 \vee x_2 \vee x_3)}_{\varphi_1} \wedge \underbrace{(x_1 \vee \neg x_2 \vee \neg x_3)}_{\varphi_2} \wedge \underbrace{(\neg x_1 \vee x_2)}_{\varphi_3}$$



φ_1 can be coloured only if the colour of at least one of $\overline{x}_1, \overline{x}_2, \overline{x}_3$ is E.

$\rightarrow$ M is p-colourable iff

$$\varphi_1 \wedge \cdots \wedge \varphi_k$$

is satisfiable. □

SUBSET SUM

SUBSET SUM: Let S be a finite set of elements and w be the weight function $w : S \to \mathbb{Z}$. Is there a subset S' of elements of S, $S' \subseteq S$, s.t. the total weight of elements from S' is W, i.e.

$$\sum_{s \in S'} w(s) = W \; ?$$

Theorem

SUBSET SUM *is* **NP**-*complete.*

Proof.

- SUBSET SUM is **NP**-hard — by reduction from 3-SAT. For a formula $\varphi_1 \wedge \cdots \wedge \varphi_k$ over variables $x_1, \ldots, x_n$, we set $S = \{t_1, \ldots, t_n, f_1, \ldots, f_n, c_1, \ldots, c_k, c'_1, \ldots, c'_k\}$ and assign values to w and W in the following way: *(next slide)*

SUBSET SUM

Proof (cont).

Assume the formula

$$\underbrace{(x_1 \vee x_2 \vee x_3)}_{\varphi_1} \wedge \underbrace{(x_1 \vee \neg x_2 \vee \neg x_3)}_{\varphi_2} \wedge \underbrace{(\neg x_1 \vee x_2)}_{\varphi_3}$$

- We consider decimal encoding of w and W of length $n + k$.
- Each variable x_i is assigned a pair of elements t_i and f_i.
- Each clause φ_j is assigned a pair of elements c_j and c'_j.

SUBSET SUM

Proof (cont).

Assume the formula

$$\underbrace{(x_1 \vee x_2 \vee x_3)}_{\varphi_1} \wedge \underbrace{(x_1 \vee \neg x_2 \vee \neg x_3)}_{\varphi_2} \wedge \underbrace{(\neg x_1 \vee x_2)}_{\varphi_3}$$

- We consider decimal encoding of w and W of length $n + k$.
- Each variable x_i is assigned a pair of elements t_i and f_i.
- Each clause φ_j is assigned a pair of elements c_j and c'_j.

	x_1	x_2	x_3	φ_1	φ_2	φ_3
t_1	1	0	0	1	1	0
f_1	1	0	0	0	0	1
t_2	0	1	0	1	0	1
f_2	0	1	0	0	1	0
t_3	0	0	1	1	0	0
f_3	0	0	1	0	1	0
c_1	0	0	0	1	0	0
c'_1	0	0	0	1	0	0
c_2	0	0	0	0	1	0
c'_2	0	0	0	0	1	0
c_3	0	0	0	0	0	1
c'_3	0	0	0	0	0	1
W	1	1	1	3	3	3

□

PARTITION

PARTITION: Let T be a finite set of elements and v be the weight function $v : T \to \mathbb{Z}$. Can T be partitioned into two sets T' and $T \setminus T'$ of equal total weight, i.e.

$$\sum_{t \in T'} v(t) = \sum_{t \in T \setminus T'} v(t) \ ?$$

Theorem

PARTITION *is* **NP***-complete.*

Proof.

- PARTITION is **NP**-hard — by reduction from SUBSET SUM. For the elements S, weight function w and target weight W, we set $T = S \cup \{z\}$ where $z \notin S$, and $v = w \cup \{z \mapsto (w(S) - 2W)\}$ where $w(S) = \sum_{s \in S} w(s)$. □

KNAPSACK

KNAPSACK: Let R be a finite set of elements, u be the weight function $u : R \to \mathbb{Z}$, and v be the value function $v : R \to \mathbb{Z}$. Is there a subset R' of elements of R, $R' \subseteq R$, s.t. the total weight of elements from R' is at most U and their total value is at least V, i.e.

$$\sum_{r \in R'} u(r) \leq U \;\wedge\; \sum_{r \in R'} v(r) \geq V\ ?$$

Theorem

KNAPSACK *is* **NP***-complete.*

Proof.

- KNAPSACK is **NP**-hard — by reduction from SUBSET SUM. For the elements S, weight function w and target weight W, we set $R = S$, $u = w$, $v = w$, $U = W$, and $V = W$. □