# Solutions Excercise 2

Maksimov, Dmitrii
dmitrii.maksimov@fau.de

Ilia, Dudnik
ilia.dudnik@fau.de

November 19, 2021

## Exercise 1

The *k-SAT* problem is defined as follows:

- Given: Let $\mathcal{F}$ be a Boolean CNF formula where each clause contains exactly $k$ literals.
- Question: Is $\mathcal{F}$ satisfiable?

(a) Show: 3-SAT is $\mathcal{NP}$-complete.

3-SAT is $\mathcal{NP}$-complete if it is a $\mathcal{NP}$-hard and also 3-SAT $\in \mathcal{NP}$.

1. Proof of 3-SAT $\in \mathcal{NP}$

   3-SAT is a subset of SAT and we know that SAT is $\mathcal{NP}$.

   $$\text{3-SAT} \subseteq \text{SAT} \in \mathcal{NP} \Longrightarrow \text{3-SAT} \in \mathcal{NP}$$

2. Proof of 3-SAT is a $\mathcal{NP}$-hard

   3-SAT is a $\mathcal{NP}$-hard if for every $L' \in \mathcal{NP}$, there is a Karp reduction from $L$' to 3-SAT. Let us see that SAT can be reduced to 3-SAT in polynomial time:

   $$\text{SAT} \leq_p \text{3-SAT}.$$

   The original SAT problem is:

   $$\phi = C_1 \wedge C_2 \wedge \cdots \wedge C_n,$$

   where $\phi$ is satisfiable if every clause $C_i$ is satisfiable and $C_i = (v_1 \vee \cdots \vee v_m)$, $m = k$. Let us find an alogorithm $R$ for $k = 1$, $k = 2$, $k > 3$.

   - $k = 1$

     Let $z_1$ and $z_2$ be two new variables. Replace $C_i$ by the following four clauses:

     $$C_i' = (v \vee z_1 \vee z_2) \cdot (v \vee \overline{z_1} \vee z_2) \cdot (v \vee z_1 \vee \overline{z_2}) \cdot (v \vee \overline{z_1} \vee \overline{z_2}).$$

The truth table for this function:

| $v$ | $z_1$ | $z_2$ | $v \vee z_1 \vee z_2$ | $v \vee \overline{z_1} \vee z_2$ | $v \vee z_1 \vee \overline{z_2}$ | $v \vee \overline{z_1} \vee \overline{z_2}$ | $C'_i$ | $C_i$ |
|---|---|---|---|---|---|---|---|---|
| $T$ | $T$ | $T$ | $T$ | $T$ | $T$ | $T$ | $T$ | $T$ |
| $T$ | $T$ | $F$ | $T$ | $T$ | $T$ | $T$ | $T$ | $T$ |
| $T$ | $F$ | $T$ | $T$ | $T$ | $T$ | $T$ | $T$ | $T$ |
| $T$ | $F$ | $F$ | $T$ | $T$ | $T$ | $T$ | $T$ | $T$ |
| $F$ | $T$ | $T$ | $T$ | $T$ | $T$ | $F$ | $F$ | $F$ |
| $F$ | $T$ | $F$ | $T$ | $F$ | $T$ | $T$ | $F$ | $F$ |
| $F$ | $F$ | $T$ | $T$ | $T$ | $F$ | $T$ | $F$ | $F$ |
| $F$ | $F$ | $F$ | $F$ | $T$ | $T$ | $T$ | $F$ | $F$ |

Hence, $C_i$ can be reduced to $C'_i$ where each clause in $C'_i$ contains exactly 3 literals and $C_i \iff C'_i$

- $k = 2$

  Let $z$ be the new variable. Replace $C_i$ by the conjunction of clauses $C'_i$, where:

$$C'_i = (v_1 \vee v_2 \vee z) \cdot (v_1 \vee v_2 \vee \overline{z}).$$

The truth table for this function:

| $v_1$ | $v_2$ | $z$ | $v_1 \vee v_2 \vee z$ | $v_1 \vee v_2 \vee \overline{z}$ | $C'_i$ | $C_i$ |
|---|---|---|---|---|---|---|
| $T$ | $T$ | $T$ | $T$ | $T$ | $T$ | $T$ |
| $T$ | $T$ | $F$ | $T$ | $T$ | $T$ | $T$ |
| $T$ | $F$ | $T$ | $T$ | $T$ | $T$ | $T$ |
| $T$ | $F$ | $F$ | $T$ | $T$ | $T$ | $T$ |
| $F$ | $T$ | $T$ | $T$ | $T$ | $T$ | $T$ |
| $F$ | $T$ | $F$ | $T$ | $T$ | $T$ | $T$ |
| $F$ | $F$ | $T$ | $T$ | $F$ | $F$ | $F$ |
| $F$ | $F$ | $F$ | $F$ | $T$ | $F$ | $F$ |

Hence, $C_i$ can be reduced to $C'_i$ where each clause in $C'_i$ contains exactly 3 literals and $C_i \iff C'_i$

- $k > 3$

  Let $A = v_3 \vee \cdots \vee v_k$. So, $C_i = v_1 \vee v_2 \vee A$. Define $C'_i$ as following:

$$C'_i = (v_1 \vee v_2 \vee z) \cdot (\overline{z} \vee A),$$

where $z$ is a new variable. The first clause contains exactly 3 literals and the same procedure can be applied repeatedly to the second clause. Hence, we need to proof that $C_i'$ and $C_i$ are equisatisfiable. The truth table:

| $v_1$ | $v_2$ | $z$ | $A$ | $v_1 \vee v_2 \vee z$ | $\overline{z} \vee A$ | $C_i'$ | $C_i$ |
|---|---|---|---|---|---|---|---|
| $T$ | $T$ | $T$ | $T$ | $T$ | $T$ | $T$ | $T$ |
| $T$ | $T$ | $T$ | $F$ | $T$ | $F$ | $F$ | $T$ |
| $T$ | $T$ | $F$ | $T$ | $T$ | $T$ | $T$ | $T$ |
| $T$ | $T$ | $F$ | $F$ | $T$ | $T$ | $T$ | $T$ |
| $T$ | $F$ | $T$ | $T$ | $T$ | $T$ | $T$ | $T$ |
| $T$ | $F$ | $T$ | $F$ | $T$ | $F$ | $F$ | $T$ |
| $T$ | $F$ | $F$ | $T$ | $T$ | $T$ | $T$ | $T$ |
| $T$ | $F$ | $F$ | $F$ | $T$ | $T$ | $T$ | $T$ |
| $F$ | $T$ | $T$ | $T$ | $T$ | $T$ | $T$ | $T$ |
| $F$ | $T$ | $T$ | $F$ | $T$ | $F$ | $F$ | $T$ |
| $F$ | $T$ | $F$ | $T$ | $T$ | $T$ | $T$ | $T$ |
| $F$ | $T$ | $F$ | $F$ | $T$ | $T$ | $T$ | $T$ |
| $F$ | $F$ | $T$ | $T$ | $T$ | $T$ | $T$ | $T$ |
| $F$ | $F$ | $T$ | $F$ | $T$ | $F$ | $F$ | $F$ |
| $F$ | $F$ | $F$ | $T$ | $F$ | $T$ | $F$ | $T$ |
| $F$ | $F$ | $F$ | $F$ | $F$ | $T$ | $F$ | $F$ |

Hence, from this table we can we conclude that $C_i'$ is satisfied if and only if $C_i$ is satisfied.

Taking into account that all procedures above perform in polynomial time - 3-SAT is a $\mathcal{NP}$-hard.

Since 3-SAT is a $\mathcal{NP}$-hard and 3-SAT $\in \mathcal{NP}$, 3-SAT is $\mathcal{NP}$-complete.

**P.S. We are not sure that we have used Karp reduction rather than Cook-Levin Theorem.**

(b) Next, show that 2-SAT is in $\mathcal{P}$.

(i) Give an example of a non-satisfiable 2-SAT formula and the corresponding auxiliary graph $\widetilde{G}$.

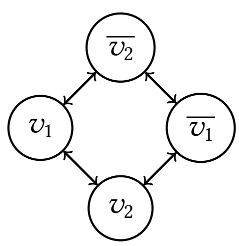
An example of non-satisfiable2 2-SAT fromula:

$$\mathcal{F} = (v_1 \vee v_2) \cdot (\overline{v_1} \vee \overline{v_2}) \cdot (\overline{v_1} \vee v_2) \cdot (v_1 \vee \overline{v_2}).$$

The directed graph will contain the following vertices and edges:

$\overline{v_1} \implies v_2, \overline{v_2} \implies v_1; v_1 \implies \overline{v_2}, v_2 \implies \overline{v_1}; v_1 \implies v_2, \overline{v_2} \implies \overline{v_1}; \overline{v_1} \implies \overline{v_2}, v_2 \implies v_1.$

The corresponding graph:



(ii) Show: If there exists a directed path from $v$ to $u$ in $\widetilde{G}$ for $u, v \in V$, then there also exists a path from $\overline{u}$ to $\overline{v}$.

Let the path from $v$ to $u$ be $v \implies p_1 \implies p_2 \implies \cdots \implies p_n \implies u$. By construction of $\widetilde{G}$, if there is a $l_1 \implies l_2$, then there is also $\overline{l_2} \implies \overline{l_1}$. Therefore, there is a path: $\overline{v} \Leftarrow \overline{p_1} \Leftarrow \overline{p_2} \Leftarrow \cdots \Leftarrow \overline{p_n} \Leftarrow \overline{u}$. Hence, there also exists a path from $\overline{u}$ to $\overline{v}$.

(iii) Show: 2-SAT is <u>not</u>-satisfiable if and only if there is a node $v \in V$ for which there exists both a directed path from $v$ to $\overline{v}$ as well as a directed path from $\overline{v}$ to $v$.

Let the path from $v$ to $\overline{v}$ be $v \implies p_1 \implies p_2 \implies \cdots \implies l_1 \implies l_2 \implies \cdots \implies p_n \implies \overline{v}$. If there is $l_1 \implies l_2$, then there is a claus $(\overline{l_1} \vee l_2)$. Consider the posiible values of $v$:

- $v$ = TRUE

  The edge from $l_i$ to $l_k$ represents that if $l_i$ is TRUE, then $l_k$ must be also TRUE. Let us divide our path into 2 paths: from $v$ to $l_1$ and from $l_2$ to $\overline{v}$. Hence, all literals in first path must be TRUE and in the second one must be FALSE. Since, $(\overline{l_1} \vee l_2)$ is a claus if it becomes FALSE $\mathcal{F}$ becomes not-satisfiable.

- $v$ = FALSE

  Exactly the same proof that $\mathcal{F}$ becomes not-satisfiable, but for a path from $\overline{v}$ to $v$.

(iv) Show by i)-iii): 2-SAT $\in \mathcal{P}$

In order to show that 2-SAT$\in \mathcal{P}$ we need to show that we can check for the existance of a path from $\overline{v}$ to $v$ and from $v$ to $\overline{v}$ in a polynomial time. The search for this path is $O(N^2)$, since for every variable (N) are considered (N) clauses, which means the algorithm is polynomial.