CP8201/CPS815 Advanced Algorithms

Assignment 5

Rohaan Ahmed
PhD Student
Department of Computer Science
rohaan.ahmed@ryerson.ca

December 10, 2020

Instructor: Dr. Javad (Jake) Doliskani
December 10, 2020

1. 3-SAT $\leq_P$ 5-SAT and 5-SAT $\leq_P$ 3-SAT

1.1 3-SAT $\leq_P$ 5-SAT

We will prove that 3-SAT $\leq_P$ 5-SAT in two parts:

1. 4-SAT $\leq_P$ 5-SAT
2. 3-SAT $\leq_P$ 4-SAT

1.1.1 4-SAT $\leq_P$ 5-SAT

An instance of the 4-SAT problem over variables $x_1, x_2, \ldots, x_n$ is given by the formula:

$$f_4 = c_1 \wedge c_2 \wedge c_3 \wedge \ldots \wedge c_k$$

Where each c_i is a disjunctive clause of exactly 4 literals for all $i = 1, \ldots, k$:

$$c_i = a \vee b \vee c \vee d$$

where a, b, c, d are distinct literals from $\{x_1, \overline{x_1}, x_2, \overline{x_2}, \ldots, x_n, \overline{x_n}\}$

Let's introduce a new variable, α, and add it to each clause, such that:

$$c_{i+} = a \vee b \vee c \vee d \vee \alpha$$
$$c_{i-} = a \vee b \vee c \vee d \vee \overline{\alpha}$$

Now, it follows from logic that:

- If c_i can be satisfied by some truth assignment, then $(c_{i+} \wedge c_{i-})$ is satisfied by the same assignment, plus $\alpha = 0$ or $\alpha = 1$.
- If $(c_{i+} \wedge c_{i-})$ can be satisfied by some truth assignment, the same assignment must satisfy c_i regardless of α since one of α or $\overline{\alpha}$ must be false.

Now, construct an instance of 5-SAT over variables $x_1, x_2, \ldots, x_n, \alpha$:

$$f_5 = c_{1+} \wedge c_{1-} \wedge c_{2+} \wedge c_{2-} \wedge c_{3+} \wedge c_{3-} \wedge \ldots \wedge c_{k+} \wedge c_{k-}$$

Then it follows that:

- If f_4 can be satisfied by some truth assignment, then f_5 is satisfied by the same assignment, plus $\alpha = 0$.
- If f_5 can be satisfied by some truth assignment, the same assignment must satisfy f_4 regardless of α.

1.1.2 3-SAT $\leq_P$ 4-SAT

We can prove that 3-SAT $\leq_P$ 4-SAT using the exact same procedure as the one described above to prove 4-SAT $\leq_P$ 5-SAT.

Let f_3 be the formula for a 3-SAT problem over variables $x_1, x_2, \ldots, x_n$:

$$f_3 = c_1 \wedge c_2 \wedge c_3 \wedge \ldots \wedge c_k$$

Where each c_i is a disjunctive clause of exactly 3 literals for all $i = 1, \ldots, k$:

$$c_i = a \vee b \vee c$$

Introduce α such that:

$$c_{i+} = a \vee b \vee c \vee \alpha$$
$$c_{i-} = a \vee b \vee c \vee \overline{\alpha}$$

Construct 4-SAT over variables $x_1, x_2, \ldots, x_n$:

$$f_4 = c_{1+} \wedge c_{1-} \wedge c_{2+} \wedge c_{2-} \wedge c_{3+} \wedge c_{3-} \wedge \ldots \wedge c_{k+} \wedge c_{k-}$$

Then it follows that:

- If f_3 can be satisfied by some truth assignment, then f_4 is satisfied by the same assignment, plus $\alpha = 0$.
- If f_4 can be satisfied by some truth assignment, the same assignment must satisfy f_3 regardless of α.

Result
The above proves the 3-SAT can be reduced to 5-SAT in polynomial time, i.e., 3-SAT $\leq_P$ 5-SAT.

1.2 5-SAT $\leq_P$ 3-SAT

An instance of the 5-SAT problem over variables $x_1, x_2, \ldots, x_n$ is given by the formula:

$$f_5 = c_1 \wedge c_2 \wedge \ldots \wedge c_k$$

Where:

$$\text{and } c_i = a \vee b \vee c \vee d \vee e$$

and a, b, c, d, e are distinct literals from $\{x_1, \overline{x_1}, x_2, \overline{x_2}, \ldots, x_n, \overline{x_n}\}$

Introduce variables β_1 and β_2 and form a new clause z_i corresponding to each c_i as follows:

$$Z_i = (a \vee b \vee \beta_1) \wedge (\overline{\beta}_1 \vee c \vee \beta_2) \wedge (\overline{\beta}_2 \vee d \vee e)$$

The specific construction of Z_i of the form above satisfies the following statements (proven in Q3, can also be verified using a truth table in this case):

- If c_i is TRUE for some assignment of $x_1, x_2, ..., x_n$, then Z_i is satisfied by the same assignment for some values for β_1 and β_2.
- If c_i is FALSE for some assignment of $x_1, x_2, ..., x_n$, then Z_i will be FALSE for the same assignment, regardless of the values set for β_1 and β_2.

Now, we construct a 3-SAT problem over the variables $x_1, x_2, \ldots, x_n$ using z_i:

$$f_3 = z_1 \wedge z_2 \wedge \ldots \wedge z_k$$

It then follows that:

- If f_5 is TRUE for some assignment of $x_1, x_2, ..., x_n$, then f_3 is satisfied by the same assignment for some values for β_1 and β_2.
- If f_5 is FALSE for some assignment of $x_1, x_2, ..., x_n$, then f_3 will be FALSE for the same assignment, regardless of β_1 and β_2.

Result

The above proves the 5-SAT can be reduced to 3-SAT in polynomial time, i.e., 5-SAT $\leq_P$ 3-SAT.

2. Find 3-Coloring using the 3-Coloring Decision Problem Oracle

Let $G = (V, E)$ be a graph with vertices $V = \{v_1, v_2, \ldots, v_n\}$ and edges $E = \{v_1v_2, v_2v_3, v_1v_3, \ldots\}$

Step 1:

Check if G is 3-colourable by calling the Oracle O.
IF not, no 3-coloring exists: Output: "no 3-coloring"
ELSE, continue to Step 2.

Step 2:

Add new vertices $V \leftarrow r, g, b$ and add the edges $E \leftarrow rb, bg, br$ to the graph and call it G_0. By this construction, G_0 is 3-colorable, and in any 3-coloring of G_0, the three new vertices must receive different colours (we can also call them R, G, B).

We now inductively create G_i from G_{i-1} for $i = \{1, 2, \ldots, n\}$, using each vertex, v_i, in G:

- Set $G_i = G_{i-1} + \{rv_i, gv_i\}$. This is the new graph made by adding edges rv_i and gv_i to G_{i-1}.
  IF this graph is 3-colourable: v_i must be B (v_i is adjacent to both R and G).

- ELSE, set $G_i = G_{i-1} + \{gv_i, bv_i\}$.
  IF this graph is 3-colourable: v_i must be R.

- ELSE, set $G_i = G_{i-1} + \{bv_i, rv_i\}$. Since G_{i-1} was 3-colourable and v_i is not B or R, v_i must be G.

The above procedure visits each vertex of the graph and calls the Oracle, O, at most twice to determine if the new combined graph is still 3-colorable. Therefore, the algorithm above calls O a total of $(2n + 1)$ times to obtain the 3-colouring of the entire n-vertex graph G.

3. 3-SAT $\leq_P$ ℓ-SAT, and ℓ-SAT $\leq_P$ 3-SAT

We will prove this using 3 cases:

1. $\ell = 1$
2. $\ell = 2$
3. $\ell > 3$

3.1. 1-SAT ($\ell = 1$)

In this case, each c_i has one literal, $l = x_i$ or $l = \overline{x_i}$, such that:

$$c_i = l$$

Introduce two new variables, z_1, z_2 and clause Z_i such that:

$$Z_i = (l \vee z_1 \vee z_2) \wedge (l \vee \overline{z_1} \vee z_2) \wedge (l \vee z_1 \vee \overline{z_2}) \wedge (l \vee \overline{z_1} \vee \overline{z_2})$$

It can be proven, using a truth table, that $Z_i = c_i = l$. Thus, the 1-SAT problem is satisfiable iff the 3-SAT problem is satisfiable.

Therefore, 1-SAT $\leq_P$ 3-SAT. There is no known way of performing reduction in the opposite direction: 3-SAT $\leq_P$ 1-SAT.

3.2. 2-SAT ($\ell = 2$)

In this case, each c_i has two literals, l_1, l_2, such that:

$$c_i = (l_1 \vee l_2)$$

Introduce a new variable, z_1 and clause Z_i such that:

$$Z_i = (l_1 \vee l_2 \vee z_1) \wedge (l_1 \vee l_2 \vee \overline{z_1})$$

It can be proven, using a truth table, that $Z_i = c_i$. Thus, the 2-SAT problem is satisfiable iff the 3-SAT problem is satisfiable.

Therefore, 2-SAT $\leq_P$ 3-SAT. There is no known way of performing reduction in the opposite direction: 3-SAT $\leq_P$ 2-SAT.

3.3. ℓ-SAT ($\ell > 3$)

We will show this in two steps:

1. ℓ-SAT $\leq_P$ 3-SAT
2. 3-SAT $\leq_P$ ℓ-SAT

3.3.1. ℓ-SAT $\leq_P$ 3-SAT

In this case, each c_i has $k > 3$ literals, $l_1, l_2, \ldots, l_k$ (we use k = ℓ to avoid confusion), such that:

$$c_i = (l_1 \vee l_2 \vee \ldots \vee l_k)$$

and f_ℓ is an ℓ-SAT problem of the form:

$$f_\ell = c_1 \wedge c_2 \wedge \ldots \wedge c_k$$

Introduce $(k-3)$ new variables, $z_1, z_2, \ldots, z_{k-3}$ and k clauses Z_i to form a 3-SAT problem:

$$f_3 = Z_1 \wedge Z_2 \wedge \ldots \wedge Z_k$$

where:

$$Z_i = (l_1 \vee l_2 \vee z_1) \wedge (l_3 \vee \overline{z_1} \vee z_2) \wedge (l_4 \vee \overline{z_2} \vee z_3) \wedge \ldots \wedge (l_{k-2} \vee \overline{z_{k-4}} \vee z_{k-3}) \wedge (l_{k-1} \vee l_k \vee \overline{z_{k-3}})$$

Unlike 1-SAT and 2-SAT, $c_i \neq Z_i$ for ℓ-SAT with $\ell > 3$. Therefore, we need to show that doing this replacement does not affect whether the formula is satisfiable.

Proof: f_3 is Satisfiable when f_ℓ is Satisfiable:

Suppose that f_ℓ is satisfiable. Select an assignment A of truth values to f_ℓ's variables that makes f_ℓ = TRUE.

Since assignment A makes f_ℓ = TRUE, it must make at least one of the literals in each clause c_i true. Let that literal be l_i.

For the corresponding Z_i, set z_j = TRUE for $j = 1, 2, ..., (i-2)$ and set z_j = FALSE for $j = (i-1), ..., (k-3)$. Then, since l_i = TRUE, the corresponding clause Z_i will also be TRUE.

Similarly, every clause Z_i in f_3 can be satisfied using the truth assignment A of the corresponding c_i in f_ℓ, plus some values for $z_1, z_2, ..., z_{k-3}$.

We can illustrate this with an example:

Example:
Suppose l_4 = TRUE for some clause c_i in a 7-SAT problem (i.e., i = 4, k = 7):

$$c_i = (l_1 \vee l_2 \vee l_3 \vee l_4 \vee l_5 \vee l_6 \vee l_7)$$

Reduce the above to a 3-SAT problem using the the method written above by setting:

$$z_1 = z_2 = \text{TRUE}$$
$$z_3 = z_4 = \text{FALSE}$$

We then get the corresponding clause Z_i:

$$Z_i = (l_1 \vee l_2 \vee \text{TRUE}) \wedge (\text{FALSE} \vee l_3 \vee \text{TRUE}) \wedge$$
$$\boldsymbol{(\text{FALSE} \vee l_4 \vee \text{FALSE})}$$
$$\wedge(\text{TRUE} \vee l_5 \vee \text{FALSE}) \wedge (\text{TRUE} \vee l_6 \vee l_7)$$

Since l_4 is TRUE, the sub-clause in **blue** is TRUE, and therefore, Z_i is TRUE.

Proof: f_ℓ is Satisfiable when f_3 is Satisfiable:

Suppose that f_3 is satisfiable. Select values for the variables that make f_3 = TRUE. We need to show that, regardless of the values for $z_1, z_2, ..., z_{k-3}$, each clause c_i in f_ℓ must have one true literal. That is, at least one of $l_1, l_2, ..., l_k$ is TRUE. We will show this is true by using Proof by Contradiction:

Assume $l_1, l_2, ..., l_k$ are all FALSE and Z_i = TRUE (corresponding to c_i = TRUE). The first sub-clause of Z_i will then be:

$$(l_1 \vee l_2 \vee z_1) = \text{TRUE}$$
$$\implies z_1 = \text{TRUE}$$

The second sub-clause in Z_i will be:

$$(\overline{z_1} \vee l_3 \vee z_2) = \text{TRUE}$$
$$\implies z_2 = \text{TRUE}$$

We proceed this way iteratively until the last two sub-clauses:

$$(\overline{z_{k-4}} \vee z_{k-3} \vee l_{k-2}) \wedge (\overline{z_{k-3}} \vee l_{k-1} \vee l_k)$$

Both these sub-clauses *cannot* be TRUE, since z_{k-4} = TRUE and $z_{k-3} \neq \overline{z_{k-3}}$. Therefore, our assumption that $l_1, l_2, ..., l_k$ are all FALSE and Z_i = TRUE cannot be true. In other words, at least one literal in $l_1, l_2, ..., l_k$ must be TRUE for Z_i = TRUE.

Thus, it does not matter what values we assign to $z_1, z_2, ..., z_{k-3}$, if c_i = TRUE, then Z_i = TRUE, and if c_i is FALSE, then Z_i =FALSE

Result

Above, we prove the following:

- If f_ℓ is TRUE for some assignment of $x_1, x_2, ..., x_n$, then f_3 is satisfied by the same assignment for some values for $z_1, z_2, ..., z_{k-3}$.

- If f_ℓ is FALSE for some assignment of $x_1, x_2, ..., x_n$, then f_3 will be FALSE for the same assignment, regardless of the values of $z_1, z_2, ..., z_{k-3}$.

This is the technique we used in Q1 to show that 5-SAT $\leq_P$ 3-SAT.

3.3.2. 3-SAT $\leq_P$ ℓ-SAT

In Q1, we showed that 3-SAT $\leq_P$ 4-SAT and 4-SAT $\leq_P$ 5-SAT, therefore, 3-SAT $\leq_P$ 5-SAT.

Without repeating the proof, it is trivial to show that for $l > 3$, $(\ell - 1)$-SAT $\leq_P$ ℓ-SAT using the same technique employed in Q1, namely:

- Generate new variable, α
- Break each clause c_i into:

$$c_{i+} = l_1 \vee l_2 \vee \ldots \vee l_k \vee \alpha$$
$$c_{i-} = l_1 \vee l_2 \vee \ldots \vee l_k \vee \overline{\alpha}$$

- Now c_i can be replaced by c_{i+} and c_{i-}

$$c_i = c_{i+} \wedge c_{i-}$$

Therefore, we can iteratively show that:

$$\text{3-SAT} \leq_P \text{4-SAT} \leq_P \text{5-SAT} \leq_P \ldots \leq_P (\ell-1)\text{-SAT} \leq_P \ell\text{-SAT}$$
$$\implies \text{3-SAT} \leq_P \ell\text{-SAT}$$

Thank you for a great semester, Professor Doliskani!