15-750: Graduate Algorithms April 28, 2017

Lecture 37: Proving NP-Completeness via reductions

Lecturer: David Witmer *Scribe: Ilaï Deutel*

1 Review of Lecture 36

Definition 1.1. Language L_1 is **poly-time reducible** to language L_2, $L_1 \leq_p L_2$, if there exists poly-time compatible $f : \{0,1\}^* \to \{0,1\}^*$ such that $x \in L_1 \Leftrightarrow f(x) \in L_2$.

Definition 1.2. A language is **NP-complete** if:

1. $L \in \text{NP}$
2. $L' \leq_p L$ for all $L' \in \text{NP}$ (L is NP-hard)

Remark 1.3. If L is NP-hard and $L \leq_p L'$, then L' is NP-hard

Theorem 1.4. *CNF·SAT is NP-complete.*

Remark 1.5. Example of CNF·SAT problem: is there a tuple of boolean variables $(x_1, x_2, x_3, \dots)$ such that $(x_1 \vee x_2 \vee \bar{x_3}) \wedge (x_7 \vee x_{10} \vee x_{15}) \wedge \dots$ is true.

2 Outline of this lecture

First, we will show that 3 coloring is NP-complete. We will do this in two main steps: First, we will reduce from 3-SAT, CNF-SAT with three variables per clause, to a problem called NAE-SAT, which is a variant of SAT in which we require each clause to have both a true literal and a false literal. Then we will reduce from NAE-SAT to 3 coloring. We will also sketch a proof that 3-coloring planar graphs is NP-complete.

Second, we will show that the independent set problem is NP-complete. Unlike SAT, NAE-SAT, and 3-coloring, independent set is not a constraint satisfaction problem (CSP).

3 Preliminary definitions and results

Definition 3.1.

- **3·SAT**: Every clause has **at most** 3 literals.
- **E3·SAT**: Every clause has **exactly** 3 literals.

Proposition 3.2. *CNF·SAT* $\leq_p$ *3·SAT*

Proposition 3.3. *3·SAT* $\leq_p$ *E3·SAT*

For technical reasons, it will be easier for us to reduce from 3·SAT and E3·SAT.

4 NAE·SAT

Definition 4.1.

- **NAE·SAT** (not-all-equal SAT): Like CNF·SAT, except clause is satisfied if at least one literal is true and one is false
- **NAE-k·SAT**: All clauses have length at most k
- **NAE-Ek·SAT**: All clauses have length exactly k

Remark 4.2. NAE·SAT is clearly in NP.

Example 4.3.

$$(x, y, z) \wedge (x, \bar{y}) \wedge \ldots$$

is satisfied by $x = T, y = T, z = F$

Remark 4.4. If X satisfies NAE·SAT instance φ, then so does $\bar{X}$ (negate every x_i)

Theorem 4.5. *3·SAT $\leq_p$ NAE-3·SAT*

We will reduce via NAE-4·SAT.

Theorem 4.6. *3·SAT $\leq_p$ NAE-4·SAT*

Proof. Given 3·SAT instance φ, make NAE-4·SAT instance ϕ' by adding new variable S to every clause (clearly in poly-time):

Example: $(x_3 \vee \bar{x_5} \vee \bar{x_5}) \to (x_3, \bar{x_5}, \bar{x_5}, S)$

Let's prove the following claim: φ satisfiable as 3·SAT $\Leftrightarrow$ φ' satisfiable as NAE-4·SAT

($\Rightarrow$) Say φ has satisfying assignment X. Then $(X, S = F)$ satifies φ': since X satisfies φ, every clause has one T and one $F(S)$.

($\Leftarrow$) Say φ' has satisfying assignment (X, S).

- If $S = F$, set $Y = X$.
- If $S = T$, then $(\bar{X}, F)$ satisfies φ'. Set $Y = \bar{X}$.

Y satisfies φ. ∎

Theorem 4.7. *NAE-4·SAT $\leq_p$ NAE-3·SAT*

Proof. Create a new variable w_i for each input clause:

Convert the i^{th} NAE-4·SAT clause (a, b, c, d) to 2 clauses $(a, b, w_i), (\bar{w_i}, c, d)$ (clearly in poly-time).

Let's prove the following claim: (a, b, c, d) is NAE $\Leftrightarrow$ there exists $w_i \in \{F, T\}$ such that (a, b, w_i) is NAE and $(\bar{w_i}, c, d)$ is NAE.

($\Leftarrow$) If (a, b, w_i) and $(\bar{w_i}, c, d)$ are both NAE, then (a, b, c, d) is NAE (can't set w_i otherwise)

($\Rightarrow$) If (a, b, c, d) is NAE, then we can satisfy (a, b, w_i) and $(\bar{w_i}, c, d)$.

- Case 1: If $a \neq b$, (a, b, w_i) is NAE, make $(\bar{w_i}, c, d)$ NAE by setting $w_i = c$.
- Case 2: If $a \neq c$, set $w_i = c \neq a$.
- Other cases are similar

∎

5 Coloring

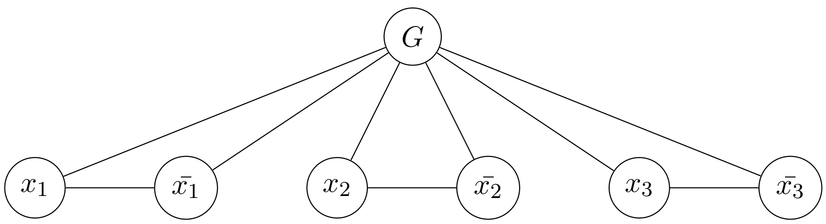
Definition 5.1. 3·COL: Decision problem: Given G, does there exist a valid 3-coloring (each edge has different-colored endpoints) of G?

Again, 3·COL is clearly in NP.

Theorem 5.2. *NAE-E3·SAT* $\leq_p$ *3·COL*

Proof. Given NAE-E3·SAT instance φ, we want to construct a 3-coloring instance (graph) G_φ.

1. Start with 1 vertex ground G.
2. Add two vertices $x_i, \bar{x_i}$ for each variable i.
3. Draw triangles $(x_i, \bar{x_i}, G)$ for each variable i.

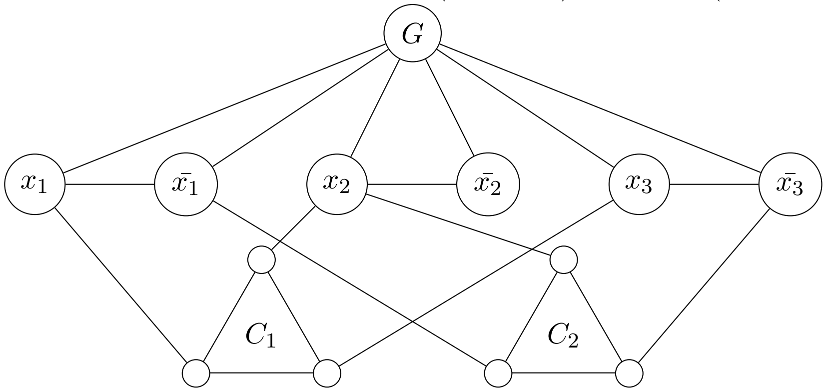


4. Without loss of generality, color Ground Y.
5. Force $x_i, \bar{x_i}$ to have different colors in $\{R, B\}$.

We can think of B = True, R = False. Any valid 3-coloring of G_φ induces a truth assignment to X.

We want to encode NAE constraint C as a "gadget": subgraph should be colorable $\Leftrightarrow$ the corresponding assignment satisfies C. It turns out that a triangle satisfies this property. For each clause C, we can add a triangle on three new vertices. Then we add edges connecting the three triangle vertices to the vertices corresponding to the literals in C. See Example 5.3.

Example 5.3. $C_1 \wedge C_2$, where $C_1 = (x_1, x_2, x_3)$ and $C_2 = (\bar{x_1}, x_2, \bar{x_3})$



It is then easy to see that the vertices of the triangle for C have a valid 3-coloring if and only if the truth assignment corresponding to this 3-coloring satisfies C. □

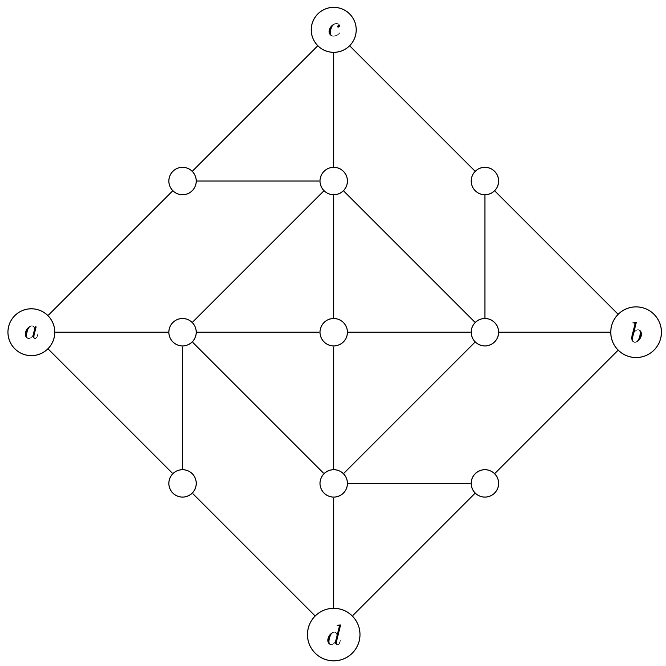
We can also consider the 3-coloring problem restricted to planar graphs.

Definition 5.4. PLANAR-3·COL: 3·COL, where G is planar

Planar 3-coloring is also NP-hard.

Theorem 5.5. *3·COL $\leq_p$ PLANAR-3·COL*

Proof. Given 3·COL instance G, we want to construct a PLANAR-3·COL instance G'. Draw G in the plane (with edge crossing), replace edge crossings with "crossover gadget"! ☐



Properties 5.6.

1. *Every valid 3-coloring χ has $\chi(a) = \chi(b)$, $\chi(c) = \chi(d)$*

2. *Given $c_1, c_2 \in \{R, B, Y\}$, $\exists$ valid 3-coloring χ such that $c_1 = \chi(a) = \chi(b)$, $c_2 = \chi(c) = \chi(d)$*

Theorem 5.7. *Any PLANAR graph can be colored with 4 colors*

Remark 5.8.

- PLANAR-3·COL is NP-hard
- PLANAR-4·COL is easy: always answer yes

6 Independent Set

So far, we have talked about constraint satisfaction problems (CSPs). Let's talk about a problem that is not a CSP.

Definition 6.1. Recall: An **independent set** is a subset S of vertices with no edge between any pair in S.

IND-SET: Decision problem: Given G, k, does G have an independent set of size $\geq k$?

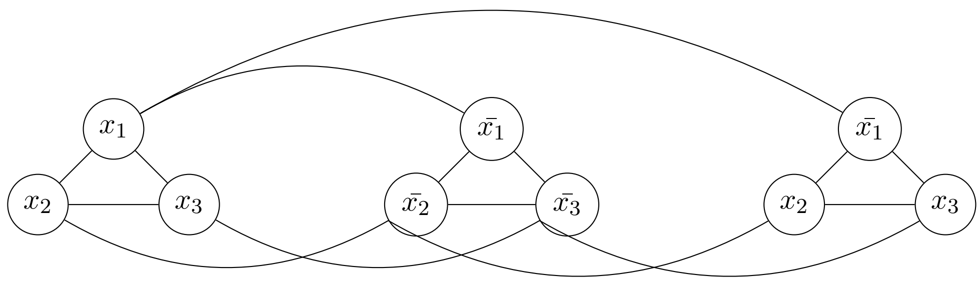
IND-SET is clearly in NP.

Theorem 6.2. *E3·SAT $\leq_p$ IND-SET*

Proof. Given E3·SAT instance, we construct an IND-SET instance.

Say our E3·SAT instance has m clauses. In this reduction, we again use a triangle as our gadget. For each clause, we add a vertex for each literal and all three edges on these vertices. We then have m disjoint, disconnected triangles. Note that any independent set has size at most m: one vertex per triangle. Then add an edge between each pair of vertices corresponding to opposite literals. Consider Example 6.3.

Example 6.3. $(x_1 \vee x_2 \vee x_3) \wedge (\bar{x_1} \vee \bar{x_2} \vee \bar{x_3}) \wedge (\bar{x_1} \vee x_2 \vee x_3)$.



We then claim that there is an independent set of size $m \Leftrightarrow$ there is a satisfying assignment for the E3·SAT instance. ☐