Undergraduate Complexity Theory
Lecture 12: NP-Completeness Reductions

Marcythm

July 15, 2022

1 Lecture Notes

Recap: poly-time mapping reduction $\le_m^P$, $\mathsf{3COL} \le_m^P \mathsf{4COL} \le_m^P \mathsf{SAT} \le_m^P \mathsf{CIRCUIT\text{-}SAT} \le_m^P \mathsf{3SAT}$. Cook-Levin: $\mathsf{CIRCUIT\text{-}SAT}$ is NP-complete = NP-hard + NP.

Today: $\mathsf{3SAT} \le_m^P \mathsf{3COL}$ (which is hw5 p4), steps there to show

1. $\mathsf{3SAT} \le_m^P \mathsf{E3SAT}$
2. $\mathsf{3SAT} \le_m^P \mathsf{NAE\text{-}3SAT}$
3. $\mathsf{NAE\text{-}3SAT} \le \mathsf{3COL}$

then show $\mathsf{3COL} \le_m^P \mathsf{INDEPENDENT\text{-}SET}$.

Remark 1.1. Here all the 5 problems shown above are CSPs (Constraints Satisfaction Problem), but $\mathsf{INDENPENDENT\text{-}SET}$ is not.

To show NP-completeness, given dicision problem L,

1. Show $L \in \mathsf{NP}$ (usually easy)
2. Pick NP-hard S (e.g. $\mathsf{3SAT}$), show $S \le_m^P L$.

Definition 1.2 ($\mathsf{E3SAT}$)**.** $\mathsf{3SAT}$ with every clause having **E**xactly 3 literals of distinct variables.

Theorem 1.3. $\mathsf{3SAT} \le_m^P \mathsf{E3SAT}$

Proof. Two cases to be considered:

1. In a clause there are two literal with the same variable, i.e. $x \vee \neg x$. These clauses are always satisfied, so just remove them.
2. In a clause there are only one or two literals. Add new variables a, b, c to ϕ, together with clauses $(\neg a \vee b \vee c), (a \vee \neg b \vee c), (a \vee b \vee \neg c), (\neg a \vee \neg b \vee c), (\neg a \vee b \vee \neg c), (a \vee \neg b \vee \neg c), (\neg a \vee \neg b \vee \neg c)$, which restricts $a = b = c = F$. Then add any of them into the clause to obtain the desired one.

$\square$

Definition 1.4 ($\mathsf{NAE\text{-}SAT}$)**.** same as $\mathsf{CNF\text{-}SAT}$ <u>except</u> instead requiring at least one true literal in each clause, we require that each clause has at least one literal assigned true and at least one literal assigned false. In other words, literals are **N**ot **A**ll **E**qual to each other.

Theorem 1.5. $\mathsf{3SAT} \le_m^P \mathsf{NAE\text{-}3SAT}$

Proof. Prove this by showing $\mathsf{3SAT} \le_m^P \mathsf{s3SAT} \le_m^P \mathsf{NAE\text{-}4SAT} \le_m^P \mathsf{NAE\text{-}3SAT}$. (reference)

First trivially convert $\mathsf{3SAT}$ into the form where each clause contains exactly 3 terms ($\mathsf{s3SAT}$).

The idea to show $\mathsf{s3SAT} \le_m^P \mathsf{NAE\text{-}4SAT}$: $(a \vee b \vee c) \mapsto (a \vee b \vee c \vee s)$, where all clauses sharing the same s. By symmetry, if $s = \mathit{true}$ then at least one of a, b, c is false, just negate all the assignments can have $(a \vee b \vee c)$ satisfiable.

The idea to show $\mathsf{NAE\text{-}4SAT} \le_m^P \mathsf{NAE\text{-}3SAT}$: $(a \vee b \vee c \vee d) \mapsto (a \vee b \vee s) \wedge (\neg s \vee c \vee d)$. $\square$

Corollary 1.6. *The proof above implies that* E3SAT $\leq_m^P$ NAE-E3SAT*.*

Remark 1.7. NAE-E3SAT is convenient for further reductions, especially for problems with "symmetry".

In NAE-SAT there are no difference between T and F (which is not the case in 3SAT). Also in 3COL there are no difference between colors.

Theorem 1.8. 3SAT $\leq_m^P$ 3COL

Proof. We do the proof in to steps.

1. Encode T/F assignment as a coloring.
   First fix a vertex, call it "Ground". By symmetry we WLOG assume it's colored Y. Then connect it with every pair of literals to form many triangles (which is also the NOT gate).

2. Encode NAE constraints in G_ϕ.
   For each claus C in ϕ, add gadget, subgraph that 3-colorable iff truth assignment induced by that 3-coloring satisfies C.

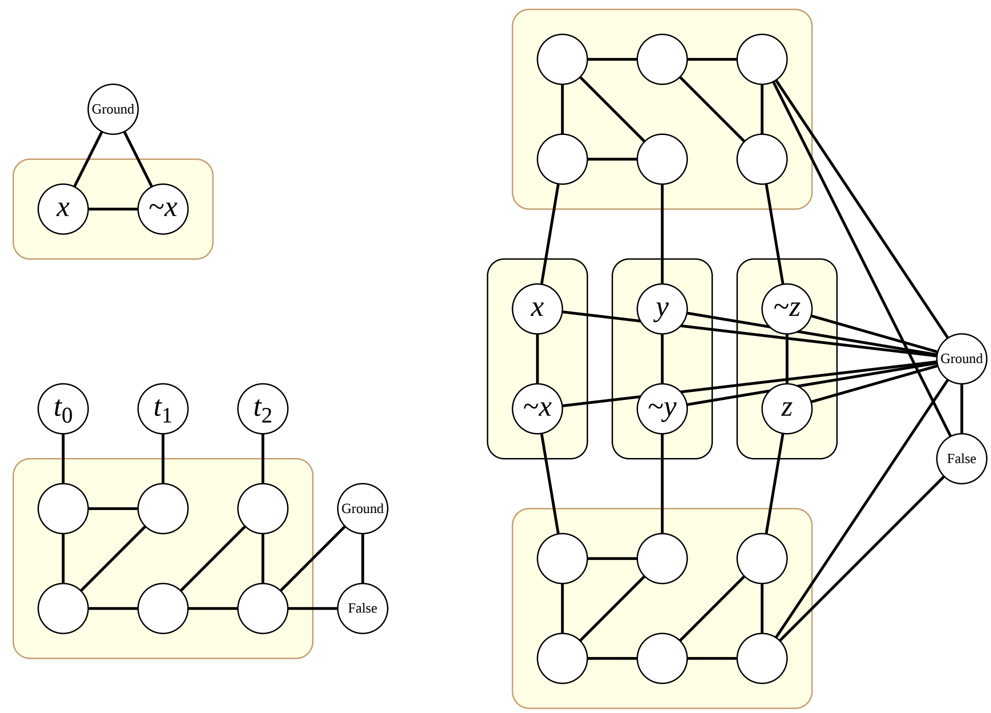

Figure 1: construction of gadgets from wikipedia (gadget)

□

Theorem 1.9. E3SAT $\leq_m^P$ INDENPENDENT-SET

Proof. Given E3SAT instance ϕ, construct G_ϕ, k_ϕ s.t. ϕ satisfiable iff G_ϕ has a independent set of size k_ϕ. Roughly speaking, one vertex for each literal in each clause ($3n$ vertices in total, where n is # of clauses), one edge for each pair of opposite literals (x and $\neg x$), and for each clause connect the three literals in it. Choose $k_\phi = n$. We show the idea by an example.

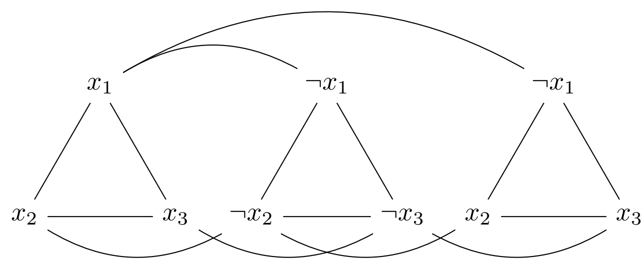

Figure 2: G_ϕ for formula $(x_1 \vee x_2 \vee x_3) \wedge (\neg x_1 \vee \neg x_2 \vee \neg x_3) \wedge (\neg x_1 \vee x_2 \vee x_3)$, with $k_\phi = 3$

Intuitively, that is to choose one literal in each clause to be true, guaranteeing no two opposite literals are chosen at the same time.

One thing should be noticed is that, cannot merge vertices for the same literal, since this would disable choosing one literal in multiple clause at the same time. $\square$

2 Reading

2.1 Sipser 7.5 (Additional NP-complete Problems)

1. VERTEX-COVER is NP-complete, by showing 3SAT $\leq_m^P$ VERTEX-COVER.
2. HAMPATH is NP-complete, by showing 3SAT $\leq_m^P$ HAMPATH.
   remark: truth assignment corresponds to direction of path.
3. UHAMPATH is NP-complete, by showing 3SAT $\leq_m^P$ UHAMPATH, constructs from directed case.
4. SUBSET-SUM is NP-complete, by showing 3SAT $\leq_m^P$ SUBSET-SUM.