COMP-170: Homework #8

Ben Tanen - April 09, 2017

Problem 2

"A little bit of everything 3SAT" is the problem where given a boolean formula in conjunctive normal form with exactly 3 literals in each clause you want to know if the formula is satisfiable and at least one clause has all three literals evaluate to true, at least one clause has two literals evaluate to true, and at least one clause has exactly one literal true.

Prove that "A little bit of everything 3SAT" is NP-complete, do a reduction from 3SAT.

* * *

To prove "A little bit of everything 3SAT" (or ALBESAT) is NP-complete, we will show that ALBESAT $\in$ NP and that 3SAT $\leq_m^p$ ALBESAT. Given that 3SAT is NP-complete, we will then be able to see that ALBESAT is NP-complete.

First, we will prove ALBESAT $\in$ NP. In order to prove this, we can construct a polynomial time verifier V for ALBESAT. Once we show that V does indeed verify in polynomial time, we will then be able to see that ALBESAT $\in$ NP. We can define V as follows:

V on input $\langle \phi, \mathscr{A} \rangle$:

// $\mathscr{A}$ is an assignment for ϕ to be verified

1. Let V_{3SAT} be a polynomial time verifier for 3SAT (we know this exists because 3SAT is NP-complete). Using V_{3SAT}, verify that $\phi \in$ 3SAT using $\mathscr{A}$. If V_{3SAT} accepts, continue. If V_{3SAT} rejects, return *REJECT*.
2. Scan through the clauses of ϕ and check if there exists some $C_i \in C(\phi)$ (where $C(\phi)$ is the set of clauses in ϕ) such that C_i has exactly one literal set to true in $\mathscr{A}$. If there does not exist such a C_i, return *REJECT*.
3. Check if there exists some $C_j \in C(\phi)$ such that C_i has exactly two literals set to true in $\mathscr{A}$. If there does not exist such a C_j, return *REJECT*.
4. Check if there exists some $C_k \in C(\phi)$ such that C_i has exactly three literals set to true in $\mathscr{A}$. If there does not exist such a C_k, return *REJECT*.
5. Return *ACCEPT*.

Given our definition of V, we claim V runs in polynomial time. Because 3SAT is NP-complete, we know there exists a polynomial time verifier for 3SAT. Thus, using V_{3SAT} in step 1 takes polynomial time. Steps 2 - 4 each take $O(m)$ time because we are simply scanning through the m clauses of ϕ. Thus, V takes polynomial time overall.

Finally, to show that V correctly verifies solutions, consider the following cases:

1. Suppose $\phi \in$ ALBESAT such that ϕ is in conjunctive normal form with exactly 3 literals in each clause and has at least one clause with all three literals evaluate to

true, at least one clause with two literals evaluate to true, and at least one clause with exactly one literal true. Also suppose $\mathscr{A}$ is a satisfying assignment for ϕ. Given ϕ, we know that V_{3SAT} would accept $\langle \phi, \mathscr{A} \rangle$ and we know that ϕ would pass steps 2 - 4 since $\phi \in$ ALBESAT. Thus, V would correctly accept $\langle \phi, \mathscr{A} \rangle$.

2. Suppose $\phi \in$ ALBESAT but $\mathscr{A}$ is not a satisfying assignment for ϕ. Because of this, we can see that V_{3SAT} would reject $\langle \phi, \mathscr{A} \rangle$, causing V to correctly reject $\langle \phi, \mathscr{A} \rangle$.

3. Suppose $\phi \notin$ ALBESAT such that ϕ is not in conjunctive normal form with exactly 3 literals in each clause. Given ϕ, we know that V_{3SAT} would reject $\langle \phi, \mathscr{A} \rangle$ so V would correctly reject ϕ.

4. Suppose $\phi \notin$ ALBESAT such that ϕ doesn't have at least one clause where all three literals evaluate to true, at least one clause where two literals evaluate to true, or at least one clause where exactly one literal true. Given this, we can see that V would halt on either step 2, 3, or 4 and correctly reject $\langle \phi, \mathscr{A} \rangle$.

Given these cases, we can see that V correctly verifies solutions, thus showing V is indeed a polynomial time verifier for ALBESAT. Thus, the existence and validity of V shows that ALBESAT $\in$ NP.

Next, we will prove 3SAT $\leq_m^p$ ALBESAT. In order to prove this, we can construct a function $f : \Sigma^*_{\text{3SAT}} \to \Sigma^*_{\text{ALBESAT}}$ such that $\phi \in$ 3SAT $\Leftrightarrow f(\phi) \in$ ALBESAT. Once we show that f satisfies this condition, we will have proven that 3SAT $\leq_m^p$ ALBESAT.

Given this, we can define $f : \Sigma^*_{\text{3SAT}} \to \Sigma^*_{\text{ALBESAT}}$ as follows:

$$f(\phi) = \phi' = \phi \wedge (x_1 \vee x_2 \vee x_3) \wedge (x_1 \vee x_2 \vee \overline{x_3}) \wedge (x_1 \vee \overline{x_2} \vee \overline{x_3})$$
$$\text{where } x_1,\ x_2,\ x_3 \text{ do not appear anywhere in } \phi$$

Given our definition of f, we claim that f is computable in polynomial time. We can see that we could construct a Turing machine M that outputs $f(\phi) = \phi'$ when given ϕ. M would simply walk through ϕ, append three clauses to the end of ϕ, and output ϕ'. This would take $O(m) + O(1)$ time overall, so we can see that f is indeed computable in polynomial time.

Next, we must verify that f satisfies the condition that $\phi \in$ 3SAT $\Leftrightarrow f(\phi) \in$ ALBESAT. In order to do this, consider the following:

1. ($\Rightarrow$) Suppose $\phi \in$ 3SAT such that there exists a satisfying assignment for ϕ, which is in conjunctive normal form with exactly 3 literals in each clause. Let $\mathscr{A}$ be a satisfying assignment for ϕ. Because $f(\phi) = \phi' = \phi \wedge (x_1 \vee x_2 \vee x_3) \wedge (x_1 \vee x_2 \vee \overline{x_3}) \wedge (x_1 \vee \overline{x_2} \vee \overline{x_3})$, where x_1, x_2, and x_3 do not appear in ϕ, we can see that we can add assignments for x_1, x_2, and x_3 to $\mathscr{A}$ to get $\mathscr{A}'$. If we assign x_1, x_2, and x_3 all to be true, we can see that $\mathscr{A}'$ still satisfies ϕ as well as the three additional clauses added to ϕ to get ϕ'. Thus, $\mathscr{A}'$ is a satisfying assignment for ϕ', showing that $\phi' \in$ 3SAT. Furthermore, when x_1, x_2, and x_3 are all assigned to true (as they are in $\mathscr{A}'$), we can see that ϕ' does have at least one clause with all three literals evaluating to true ($x_1 \vee x_2 \vee x_3$ contains three literals

that evaluate to true), at least one clause with exactly two literals evaluating to true ($x_1 \vee x_2 \vee \overline{x_3}$ contains two literals that evaluate to true), and at least one clause with exactly one literal evaluating to true ($x_1 \vee \overline{x_2} \vee \overline{x_3}$ contains one literal that evaluates to true). Thus we can see that $f(\phi) = \phi' \in$ ALBESAT.

2. ($\Leftarrow$) Suppose $f(\phi) = \phi' \in$ ALBESAT such that there exists a satisfying assignment for ϕ', which is in conjunctive normal form with exactly 3 literals in each clause and has at least one clause with all three literals evaluating to true, at least one clause with exactly two literals evaluating to true, and at least one clause with exactly one literal evaluating to true. Given this, let $\mathscr{A}'$ be any satisfying assignment for ϕ'. Given the definition of f, we know that ϕ' contains exactly three clauses that contain the variables x_1, x_2, and x_3. Thus, if we were to remove these three clauses from ϕ', we would be left with our original ϕ and we would no longer need assignments for x_1, x_2, or x_3 in $\mathscr{A}'$. Because x_1, x_2, and x_3 appear nowhere in ϕ, we know that $\mathscr{A}'$ without assignments for x_1, x_2, and x_3 (a.k.a $\mathscr{A}$) is still a satisfying assignment for ϕ. This shows there exists a satisfying assignment for ϕ, which is still in conjunctive normal form with exactly 3 literals in each clause. Thus, we can see that if $f(\phi) = \phi' \in$ ALBESAT, then $\phi \in$ 3SAT.

Given this, we can thus see that f does indeed satisfy the claim $\phi \in$ 3SAT $\Leftrightarrow f(\phi) \in$ ALBESAT. Therefore, we can see that 3SAT $\leq_m^p$ ALBESAT.

Given that ALBESAT $\in$ NP, 3SAT $\leq_m^p$ ALBESAT, and 3SAT is NP-complete, we can see that ALBESAT is also NP-complete. ⊠