COMP-170: Homework #8

Ben Tanen - April 09, 2017

Problem 4

Barely Legal 3SAT is similar to NTGSAT except that only one literal can be true in *each and every* clause.

BLSAT = $\{\langle\phi\rangle \mid \phi$ is 3CNF and is satisfiable with a formula where only one literal in each clause is true$\}$

Prove that BLSAT is in NP-complete. Do a reduction from 3SAT.

* * *

To prove BLSAT is NP-complete, we will show that BLSAT $\in$ NP and that 3SAT $\leq^p_m$ BLSAT. Given that 3SAT is NP-complete, we will then be able to see that BLSAT is NP-complete.

First, we will prove BLSAT $\in$ NP. In order to prove this, we can construct a polynomial time verifier V for BLSAT. Once we show that V does indeed verify in polynomial time, we will then be able to see that BLSAT $\in$ 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 more than one literal set to true in $\mathscr{A}$. If there does exist such a C_i, return *REJECT*.
3. 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 takes $O(m)$ 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 $\phi \in$ BLSAT such that ϕ is in conjunctive normal form with exactly 3 literals in each clause and no clause in ϕ has more than one literal set to 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 step 2 (by definition of ϕ). Thus, V would correctly accept $\langle\phi, \mathscr{A}\rangle$.
2. Suppose $\phi \in$ BLSAT 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 $\phi \notin$ BLSAT 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 $\phi \notin$ BLSAT such that ϕ has at least one clause where two or more literals evaluate to true. Given this, we can see that V would halt on either step 2 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 BLSAT. Thus, the existence and validity of V shows that BLSAT $\in$ NP.

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

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

f on input ϕ outputs ϕ', where ϕ' is defined as follows:

Suppose $\phi = C_1 \wedge C_2 \wedge \ldots \wedge C_m$ where $C_1, C_2, \ldots, C_m$ are all three-literal clauses. Given this, initially set $\phi' = \phi$.

For each clause C_i from ϕ', where $C_i = a_i \vee b_i \vee c_i$ for some literals a_i, b_i, c_i, replace C_i with $(\overline{a_i} \vee x_{i1} \vee x_{i2}) \wedge (b_i \vee x_{i2} \vee x_{i3}) \wedge (\overline{c_i} \vee x_{i3} \vee x_{i4})$, where x_{i1}, x_{i2}, x_{i3}, and x_{i4} are dummy variables that does not appear anywhere else in ϕ'.

Output ϕ'

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 ϕ' when given ϕ. M would iterate through each of our clauses and replace each three-literal clause with three three-literal clauses, which takes $O(1)$ time per clause and takes $O(m)$ time overall. Thus, we can see f is indeed computable in polynomial time.

Next, we must verify that f satisfies the condition that $\phi \in$ 3SAT $\Leftrightarrow f(\phi) \in$ BLSAT. We can first consider the "in-cases" (i.e. when there is a satisfying assignment $\mathscr{A}$ for ϕ, there is a satisfying assignment $\mathscr{A}'$ for $f(\phi)$). To show this, we can show that for each clause C_i of ϕ, we map C_i to a series of three clauses C_i' in ϕ' that maintains the same truth value. A satisfying assignment for C_i exists when at least one of a_i, b_i, c_i evaluate to true. In order for a satisfying assignment of ϕ' to exist, we must show only one arrangement of x_{i1}, x_{i2}, x_{i3}, and x_{i4} for each case where C_i evaluates to true. Thus, consider the following truth table:

$a_i\ b_i\ c_i$	$x_{i1}\ x_{i2}\ x_{i3}\ x_{i4}$	C_i	$C_i' = (\overline{a_i} \lor x_{i1} \lor x_{i2}) \land (b_i \lor x_{i2} \lor x_{i3}) \land (\overline{c_i} \lor x_{i3} \lor x_{i4})$
$F\ F\ T$	$T\ F\ T\ T$	T	$T \land T \land T = T$
$F\ T\ F$	$F\ F\ T\ T$	T	$T \land T \land T = T$
$F\ T\ T$	$F\ F\ T\ T$	T	$T \land T \land T = T$
$T\ F\ F$	$T\ F\ T\ T$	T	$T \land T \land T = T$
$T\ F\ T$	$T\ F\ T\ T$	T	$T \land T \land T = T$
$T\ T\ F$	$T\ F\ T\ T$	T	$T \land T \land T = T$
$T\ T\ T$	$T\ F\ T\ T$	T	$T \land T \land T = T$

Given this, we can see that if there exists an assignment to make C_i evaluate to true, there exists an assignment to make C_i' evaluate to true. Therefore, we can see that if $\phi \in \text{3SAT} \Rightarrow f(\phi) = \phi' \in \text{BLSAT}$.

In order to show the "out-case" (i.e., $\phi \notin \text{3SAT} \Rightarrow f(\phi) = \phi' \notin \text{BLSAT}$), we can use a proof by contradiction. Specifically, given some $\phi \notin \text{3SAT}$, we will assume $\phi' \in \text{BLSAT}$ and arrive at a contradiction.

Suppose $\phi \notin \text{3SAT}$ such that there exists no assignment that makes every clause in ϕ evaluate to true. This means given any assignment, there exists some clause C_i that will evaluate to false. Given ϕ, we will assume $f(\phi) = \phi' \in \text{BLSAT}$. Now, let $\mathscr{A}$ be any assignment for ϕ and C_i be any clause in ϕ that evaluates to false given $\mathscr{A}$ where $C_i = a_i \lor b_i \lor c_i$. Because C_i evaluates to false, we know that a_i, b_i, and c_i are all assigned to false in $\mathscr{A}$. Now consider C_i', the clause triple in ϕ' that C_i was transformed into, where $C_i' = (\overline{a_i} \lor x_{i1} \lor x_{i2}) \land (b_i \lor x_{i2} \lor x_{i3}) \land (\overline{c_i} \lor x_{i3} \lor x_{i4})$. Because C_i' comes from ϕ', which we've assumed to be in BLSAT, we know that each clause of C_i' contains exactly one literal that evaluates to true. Now, because a_i, b_i, and c_i are all false, we can see that the first clause of C_i' contains $\overline{a_i}$, which evaluates to true. Thus, we know, in order for $\phi' \in \text{BLSAT}$, x_{i1} and x_{i2} must evaluate to false. Likewise, we can see that the third clause of C_i' contains $\overline{c_i}$, which evaluates to true. Thus, we know, in order for $\phi' \in \text{BLSAT}$, x_{i3} and x_{i4} must also evaluate to false. Therefore, we are left with the middle clause of C_i' where b_i, x_{i2}, and x_{i3} have all previously been assigned to evaluate as false in order to guarantee $\phi' \in \text{BLSAT}$. However, this is impossible, because if b, x_2, and x_3 all evaluate to false, $b_i \lor x_{i2} \lor x_{i3}$ evaluates to false, which would make $\phi' \notin \text{BLSAT}$. Thus, we arrive at a contradiction, indicating that if $\phi \notin \text{3SAT}$, $f(\phi) = \phi' \notin \text{BLSAT}$.

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

Given that $\text{BLSAT} \in \text{NP}$, $\text{3SAT} \leq_m^p \text{BLSAT}$, and 3SAT is NP-complete, we can see that BLSAT is also NP-complete. $\boxtimes$