COMP-170: Homework #8

Ben Tanen - April 09, 2017

Problem 3

Nobody's That Good 3SAT is the problem where given a boolean formula in conjunctive normal form with 3 literals you want to know if there is a satisfying assignment where at least one literal in *each and every* clause evaluates to false.

NTGSAT = $\{\langle\phi\rangle \mid \phi$ is in 3CNF and it is satisfiable with a formula where no clause evaluates to all true$\}$

Prove that NTGSAT is in NP-complete, do a reduction from 3SAT.

* * *

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

First, we will prove NTGSAT $\in$ NP. In order to prove this, we can construct a polynomial time verifier V for NTGSAT. Once we show that V does indeed verify in polynomial time, we will then be able to see that NTGSAT $\in$ 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 all three literals 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 $\phi \in$ NTGSAT such that ϕ is in conjunctive normal form with exactly 3 literals in each clause and no clause in ϕ has all three literals 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 $\phi \in$ NTGSAT 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 $\phi \notin$ NTGSAT 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 $\phi \notin$ NTGSAT such that ϕ has at least one clause where all three 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 NTGSAT. Thus, the existence and validity of V shows that NTGSAT $\in$ NP.

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

Given this, we can define $f : \Sigma^*_{\text{4SAT}} \to \Sigma^*_{\text{3SAT}}$ 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 $(a_i \vee b_i \vee x_{i1}) \wedge (c_i \vee \overline{x_{i1}} \vee x_{i2}) \wedge (\overline{x_{i1}} \vee \overline{x_{i2}} \vee \overline{x_{i3}}) \wedge (\overline{x_{i1}} \vee \overline{x_{i2}} \vee x_{i3})$, where x_{i1}, x_{i2}, and x_{i3} are dummy variables that do 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 four 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 $\Leftrightarrow f(\phi) \in$ NTGSAT. To show this, we can show that for each clause C_i of ϕ, we map C_i to a series of four clauses C_i' in ϕ' that maintains the same truth value. We can first consider the "in-cases" (i.e. when there is a satisfying assignment $\mathscr{A}$ for C_i in ϕ, there is a satisfying assignment $\mathscr{A}'$ for C_i' in $f(\phi)$). 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}, and x_{i3} for each satisfying assignment of C_i which makes C_i' evaluate to true. Thus, consider the following truth table:

a_i b_i c_i	x_{i1} x_{i2} x_{i3}	C_i	$C_i' = (a_i \vee b_i \vee x_{i1}) \wedge (c_i \vee \overline{x_{i1}} \vee x_{i2}) \wedge (\overline{x_{i1}} \vee \overline{x_{i2}} \vee \overline{x_{i3}}) \wedge (\overline{x_{i1}} \vee \overline{x_{i2}} \vee x_{i3})$
F F T	T F T	T	$T \wedge T \wedge T \wedge T = T$
F T F	F F T	T	$T \wedge T \wedge T \wedge T = T$
F T T	F F T	T	$T \wedge T \wedge T \wedge T = T$
T F F	F F T	T	$T \wedge T \wedge T \wedge T = T$
T F T	T F T	T	$T \wedge T \wedge T \wedge T = T$
T T F	F F T	T	$T \wedge T \wedge T \wedge T = T$
T T T	T F T	T	$T \wedge T \wedge T \wedge T = T$

We can next consider the "out-cases" (i.e. when there is no satisfying assignment $\mathscr{A}$ for C_i in ϕ, there is no satisfying assignment $\mathscr{A}'$ for C_i' in $f(\phi)$). No satisfying assignment for C_i exists when all of a_i, b_i, c_i evaluate to false. In order for no satisfying assignment of ϕ' to exist, we must consider all of the cases where a_i, b_i, c_i evaluate to false. Thus, consider the following truth table:

a_i b_i c_i	x_{i1} x_{i2} x_{i3}	C_i	$C_i' = (a_i \vee b_i \vee x_{i1}) \wedge (c_i \vee \overline{x_{i1}} \vee x_{i2}) \wedge (\overline{x_{i1}} \vee \overline{x_{i2}} \vee \overline{x_{i3}}) \wedge (\overline{x_{i1}} \vee \overline{x_{i2}} \vee x_{i3})$
F F F	F F F	F	$F \wedge T \wedge T \wedge T = F$
F F F	F F T	F	$F \wedge T \wedge T \wedge T = F$
F F F	F T F	F	$F \wedge T \wedge T \wedge T = F$
F F F	F T T	F	$F \wedge T \wedge T \wedge T = F$
F F F	T F F	F	$T \wedge F \wedge T \wedge T = F$
F F F	T F T	F	$T \wedge F \wedge T \wedge T = F$
F F F	T T F	F	$T \wedge T \wedge T \wedge F = F$
F F F	T T T	F	$T \wedge T \wedge F \wedge T = F$

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. We can also see that if there exists no assignment to make C_i evaluate to true (i.e., all of the literals of C_i evaluate to false), there exists no assignment to make C_i' evaluate to true. Thus, for all clauses C_i in ϕ, we can see that C_i and C_i' always evaluate to the same truth value. Therefore, we can see that f does indeed satisfy the claim $\phi \in$ 3SAT $\Leftrightarrow f(\phi) \in$ NTGSAT. Thus, we can see that 3SAT $\leq_m^p$ NTGSAT.

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