Algorithms and Complexity April 11, 2017
UT Austin CS 331
Prof. Dana Moshkovitz Problem Set 7

Problem Set 7

This problem set is due at **10:00 am** on **Tuesday, April 18th**.

Problem 7- 1: Reductions

In *SAT*(10) the input is a Boolean formula φ in which each variable appears in at most 10 clauses. The problem is to decide whether φ is satisfiable. Show a polynomial-time reduction from *3SAT* to *SAT*(10).
Solution:
Reduction:

Given 3 CNF boolean formula S.

Let $C = \{c_1, c_2, c_3, \cdots, c_n\}$ be the set of clauses in S.

Let $X = \{x_1, x_2, x_3, \dots\ x_n\}$ be the set of variables in S.

Each clause c_i is of the form $(t_1 \vee t_2 \vee t_3)$ where $t_i \in \{x_j, \neg x_j\}$

Let $X' = \{x'_i \in X |\ |\{c_i \in C | c_i \ni x'_i\}|\ > 10\}$ be the set of all variables in X that appear in more than 10 clauses.

Let $S' = S$. Let all changes to the boolean formula described below now be performed on S'.

For each variable x'_i in X' take all clauses (except the first 6) in which x'_i appears and replace it with a newly introduced variable x_{n+j}. Let j start at 1 and increment for each new variable created. Add two additional clauses $(\neg x'_i \vee x_{n+j}) \wedge (x'_i \vee \neg x_{n+j})$. If x_{n+j} is contained in more than 10 clauses in S', add x_{n+j} to $X^{'}$, and make sure not to replace the clauses of the form $(\neg A \vee B) \wedge (A \vee \neg B)$

S' is now in $SAT(10)$ form.

S is satisfiable $\iff$ S' is satisfiable:

S is satisfiable $\Rightarrow$ S' is satisfiable.

Given a satisfiable assignment of each element of X^S to boolean values, there is a satisfying assignment of all elements of $X^{S'}$.

First assign all corresponding variables of X^S to $X^{S'}$ such that $x_i^s = x_i^{s'}$.

For all remaining variables, find all clauses of the form $(\neg A \lor B) \land (A \lor \neg B)$. Consider the clauses with B unassigned and A assigned. Assign to B the same boolean value of A.

S' is now satisfied.

For clauses that were unchanged from S to S': they are satisfied because x_i^s was assigned to $x_i^{s'}$.

For clauses in which the variables were replaced with a new variable: they are satisfied because the two variables were assigned the same value.

For the added clauses: they are satisfied because they are of the form $(\neg A \lor B) \land (A \lor \neg B)$ which has the following truth table:

A	B	$(\neg A \lor B) \land (A \lor \neg B)$
0	0	1
0	1	0
1	0	0
1	1	1

Thus as long as A and B have the same value, they are satisfied.

S is satisfiable $\Leftarrow$ S' is satisfiable.

Given a satisfiable assignment of each element of $X^{S'}$ to boolean values, there is a satisfying assignment of all elements of X^S.

Assign all corresponding variables of $X^{S'}$ to X^S such that $x_i^s = x_i^{s'}$.

S is now satisfied by a similar argument as the case above.

This reduction takes polynomial time because

1. finding the variables that appear in more than 10 clauses takes linear time with respect to the number of clauses

2. replacing a variable takes linear time. (Need to iterate through the clauses in which to replace).

3. The number of times a variable is replaced is bounded by the square of number of times the variable appears in a clause.

4. The number of times the variable appears in a clause is bounded the number of clauses. (We do not replace the additional clauses we created.)

$\approx O(N^3)$ where N is the number of clauses

Problem 7- 2: Reductions and algorithms

Suppose that one can decide *3SAT* in polynomial time. Show that given a Boolean formula φ one can find the satisfying assignment for φ, if such exists, in polynomial time. If φ is unsatisfiable, the algorithm should declare so.

Hint: Given a Boolean formula and a variable, consider the formula where the variable is substituted TRUE and another formula where the same variable is substituted FALSE.

Solution:

Let $SAT(\varphi)$ be an algorithm that determines if φ is satisfiable. Let $x_1, x_2, \ldots, x_n$ be the variables in φ. In this algorithm, 0 will indicate false and 1 will indicate true. To determine the assignment that satisfies φ, first call $SAT(\varphi)$. If $SAT(\varphi) = 0$, then φ is not satisfiable, so declare it so.

Otherwise, φ is satisfiable. Compute φ_1^1, substituting 1 for all x_1 in φ. This means that every clause containing x_1 reduces to 1, and can therefore be eliminated from the set of clauses. If there are no other clauses, then the $\varphi_1^1 = 1$.

Also compute φ_1^0, substituting 0 for all x_1 in φ. This means that every clause containing x_1 can simply remove x_1 from the clause. If the clause has no more variables, then the whole clause can be set to 0, and $\varphi_1^0 = 0$.

Next, call $SAT(\varphi_1^1)$ and $SAT(\varphi_1^0)$. Note that one of these must return 1, because φ itself is satisfiable. If $SAT(\varphi_1^1)$ is 1, then set $\varphi_1 = \varphi_1^1$, and assign $x_1 = 1$. Similarly, if $SAT(\varphi_1^0) = 1$, then set $\varphi_1 = \varphi_1^0$, and assign $x_1 = 0$.

Continue making assignments to x_k by computing φ_k^1 and φ_k^0 by substituting 1 and 0 respectively for x_k into φ_{k-1}, until all x_k are assigned. This will be a valid assignment for the x_ns to satisfy φ.

If $SAT(\varphi)$ takes $O(n^{O(1)})$, each iteration of the algorithm takes $O(n^{O(1)})$ time; this implies the entire algorithm runs in $O(n \times n^{O(1)}) = O(n^O(1))$ time.