Reduction from SAT to 3SAT

Swagato Sanyal

We describe a polynomial time reduction from SAT to 3SAT. The reduction takes an arbitrary SAT instance ϕ as input, and transforms it to a 3SAT instance ϕ', such that satisfiability is preserved, i.e., ϕ' is satisfiable if and only if ϕ is satisfiable. Recall that a SAT instance is an AND of some clauses, and each clause is OR of some literals. A 3SAT instance is a special type of SAT instance in which each clause has exactly 3 literals.

Example of a SAT instance
$x_1 \wedge (x_1 \vee \overline{x_2}) \wedge (x_2 \vee x_3 \vee x_5) \wedge (x_1 \vee x_4 \vee \overline{x_6} \vee \overline{x_7}) \wedge (x_1 \vee x_2 \vee \overline{x_3} \vee x_5 \vee x_7)$.

Example of a 3SAT instance
$(x_1 \vee \overline{x_2} \vee \overline{x_4}) \wedge (x_2 \vee x_3 \vee x_5) \wedge (x_1 \vee x_4 \vee \overline{x_6})$.

The reduction replaces each clause in ϕ with a set of clauses, each having exactly three literals. Assume that ϕ involves n variables $x_1, \ldots, x_n$. The new formula ϕ' will have some new variables in addition to the x_i's.
We now describe how we replace each clause in ϕ. Let C is an arbitrary clause in ϕ.

case 1: C **has one literal** : Let C consist of a single literal ℓ. ℓ is either x_i or $\overline{x_i}$ for some i. Let z_1 and z_2 be two new variables. We replace C by the following four clauses: $(\ell \vee z_1 \vee z_2), (\ell \vee \overline{z_1} \vee z_2), (\ell \vee z_1 \vee \overline{z_2}), (\ell \vee \overline{z_1} \vee \overline{z_2})$.
Please verify for yourself that the logical AND of the above four clauses is equal to ℓ. Thus, the new formula obtained by replacing C by these four clauses computes the same Boolean function as the original formula. Hence, the new formula is satisfiable if and only if the old formula is satisfiable.

case 2: C **has two literals** : Let $C = \ell_1 \vee \ell_2$. Each of ℓ_1 and ℓ_2 is either a variable x_i or a negated variable $\overline{x_i}$. Let z_1 be a new variable. Replace C by the following two clauses: $(\ell_1 \vee \ell_2 \vee z_1), (\ell_1 \vee \ell_2 \vee \overline{z_1})$.
Please verify for yourself that the logical AND of the above two clauses is equal to $C = \ell_1 \vee \ell_2$. Thus, the new formula obtained by replacing C by these two clauses computes the same Boolean function as the original formula. Hence, the new formula is satisfiable if and only if the old formula is satisfiable.

case 3: C **has three literals** : In this case leave C unchanged.

case 4: C **has more than three literals** : Let $k > 3$ and $C = \ell_1 \vee \ell_2 \vee \ldots \vee \ell_k$, where each l_i either a variable x_i or a negated variable $\overline{x_i}$. Let $z_1, z_2, \ldots, z_{k-3}$ be $k-3$ new variables. We replace C by the following $k-3$ clauses:
$(\ell_1 \vee \ell_2 \vee z_1), (\ell_3 \vee \overline{z_1} \vee z_2), (\ell_4 \vee \overline{z_2} \vee z_3), \ldots, (\ell_{k-2} \vee \overline{z_{k-4}} \vee z_{k-3}), (\ell_{k-1} \vee \ell_k \vee \overline{z_{k-3}})$.
Unlike cases 1 and 2, the logical AND of the above $k-2$ clauses is not the same as C. However, the following two statements can be verified to be true. Let Ψ denote the AND of the above $k-3$ clauses.

1. Given an assignment to $x_1, \ldots, x_n$ in which C is TRUE, there is a way of setting the new variables $z_1, \ldots, z_{k-3}$ such that Ψ is TRUE.
2. Given an assignment to $x_1, \ldots, x_n$ in which C is FALSE, there is no way of setting $z_1, \ldots, z_{k-3}$ such that Ψ is TRUE.

In the class, we showed that the statements 1 and 2 above are true for the special cases of $k = 4$ and 5.

Exercise: Prove that statements 1 and 2 are true for every $k > 3$.

From statements 1 and 2 it follows that performing the above replacement preserves the satisfiability of the original formula, i.e., the formula after the replacement is satisfiable if and only if the formula before the replacement is satisfiable.

The reduction is simply applying the appropriate replacement to each clause in ϕ. We use a fresh set of new variables z_i's for each clause. The procedure can be easily verified to be polynomial time.

Exercise: Prove the correctness of the reduction, i.e., that it preserves satisfiability.