

# NP-Completeness of 3SAT and MAX 2SAT

## 3SAT

The 3SAT problem is the following.

INSTANCE : Given a boolean expression  $E$  in conjunctive normal form (CNF) that is the conjunction of clauses, each of which is the disjunction of three distinct literals.

QUESTION : Is  $E$  satisfiable ?

The CSAT problem is the following.

INSTANCE : Given a boolean expression  $E$  in CNF.

QUESTION : Is  $E$  satisfiable; that is, is there a truth assignment to the variables of  $E$  so that each clause in  $E$  becomes true ?

The Cook-Levin theorem asserts that SATISFIABILITY is NP-complete. Although 3-CNF expressions are a subset of the CNF expressions, they are complex enough in the sense that testing for satisfiability turns out to be NP-complete.

**Theorem :** 3SAT is NP-complete.

**Proof :** Evidently 3SAT is in NP, since SAT is in NP.

To determine whether a boolean expression  $E$  in CNF is satisfiable, nondeterministically guess values for all the variables and then evaluate the expression. If  $E$  turns out to be true, then accept. This can be carried out in nondeterministic polynomial time. Thus 3SAT is in NP.

To prove NP-completeness, we shall reduce CSAT to 3SAT. Let a given CNF expression be  $E = e_1 \wedge e_2 \wedge \dots \wedge e_k$  where each  $e_i$  is a disjunction of literals. We replace each clause  $e_i$  (as shown below) to create a new expression  $F$  such that

- i)  $F$  is a clause in 3SAT form.
- ii) time taken to construct  $F$  is linear in the length of  $E$ .
- iii) a truth assignment satisfies  $E$  if and only if it can be extended to a satisfying truth assignment for  $F$ .

The construction of  $F$  is as follows.

1. If an  $e_i$  is a single literal, say  $(x)$  or  $(\bar{x})$  we introduce two new variables  $u$  and  $v$ . We replace  $(x)$  by the conjunction of four clauses as  $(x+u+v)(x+u+\bar{v})(x+\bar{u}+v)(x+\bar{u}+\bar{v})$ . Since  $u$  and  $v$  appear in all combinations, the only way to satisfy all four clauses is to make  $x$  true.
2. Suppose an  $e_i$  is the disjunction of two literals,  $(x+y)$ . We introduce a new variable  $z$  and replace  $e_i$  by the conjunction of two clauses  $(x+y+z)(x+y+\bar{z})$ . As in case 1, the only way to satisfy both clauses is to satisfy  $(x+y)$ .

3. If an  $e_i$  is the disjunction of three literals it is already in the form required for 3-CNF, so we take  $e_i$  as such to construct  $F$ .
4. Suppose  $e_i = (x_1 + x_2 + \dots + x_m)$  for some  $m \geq 4$ . We introduce new variables  $y_1, y_2, \dots, y_{m-3}$  and replace  $e_i$  by the conjunction of clauses

$$(x_1 + x_2 + y_1)(x_3 + \overline{y_1} + y_2)(x_4 + \overline{y_2} + y_3) \dots (x_{m-2} + \overline{y_{m-4}} + y_{m-3})(x_{m-1} + x_m + \overline{y_{m-3}}). \quad (1)$$

If there is a truth assignment  $\tau$  that satisfies  $e_i$  then at least one literal (one of the  $x$ 's) in  $x_1 + x_2 + \dots + x_m$  should be true; say  $\tau$  makes  $x_j$  true; that is in

$$\dots (x_{j-1} + \overline{y_{j-3}} + y_{j-2})(x_j + \overline{y_{j-2}} + y_{j-1})(x_{j+1} + \overline{y_{j-1}} + y_j) \dots$$

Then in (1) above, if we make  $y_1, y_2, \dots, y_{j-2}$  true and make  $y_{j-1}, y_j, \dots, y_{m-3}$  false, we satisfy all the clauses of (1). Thus  $\tau$  can be extended to satisfy these clauses. On the other hand, if  $\tau$  makes all the  $x$ 's false we can reason (as follows) that it is not possible to extend  $\tau$  to make (1) true. The reason is that, since  $x_1$  and  $x_2$  are false,  $y_1$  must be true (otherwise this clause will become false and will collapse  $e_i$  to false) in the first clause; since  $x_3$  is false and  $\overline{y_1}$  is false,  $y_2$  must be true to keep the situation alive. By continuing the argument, we will reason that  $y_{m-3}$  is true. But, alas,  $x_{m-1}$  is false,  $x_m$  is false and  $\overline{y_{m-3}}$  is false and we see that  $e_i$  cannot be satisfied.

The above argument shows how to reduce an instance  $E$  of CSAT to an instance  $F$  of 3SAT, such that  $F$  is satisfiable if and only if  $E$  is satisfiable. The construction evidently requires time that is linear in the length of  $E$ , because none of the four cases above expand a clause by more than a factor 32/3 (that is the ratio of symbol counts in case 1) and it is possible to build  $F$  in polynomial time.

Since CSAT is NP-complete, it follows that 3SAT is also NP-complete. □

## MAX 2SAT

The following is the maximum satisfiability with at most two literals per clause (MAX 2SAT) problem.

INSTANCE : Given a set  $C_1, \dots, C_p$  of clauses, each being a disjunction of at most two literals; and an integer  $k$  where  $1 \leq k < p$ .

QUESTION : Is there a truth assignment to the variables so that  $k$  or more clauses are satisfied ?

Note that if  $k = p$ , the above problem reduces to the 2SAT problem which can be solved efficiently in polynomial time. The following theorem relies on reducing 3SAT to MAX 2SAT. The proof rewritten slightly essentially follows [2].

**Theorem :** MAX 2SAT is NP-complete.

**Proof :** First we note that if a truth setting of variable makes at least  $k$  clauses true, then by substituting these truth values we can efficiently check the truth or falsity clause by clause, keeping a counter to hold the number of clauses found to be true so far. Hence MAX 2SAT is in NP.

The following is a reduction from 3SAT to MAX 2SAT. That is, given an instance of 3SAT we construct an instance of MAX 2SAT so that a satisfying truth assignment of 3SAT can be extended to a satisfying truth assignment of MAX 2SAT. Note that a satisfying truth assignment

is one that makes at least  $k$  clauses true in the MAX 2SAT instance.

Let  $S$  be the instance of 3SAT where the clauses are  $C_1, \dots, C_m$  where  $C_i = \{x_i, y_i, z_i\}$ , where each  $x_i, y_i$ , and  $z_i$  represents either a variable or its negation and  $1 \leq i \leq m$ . From  $S$  we build an instance  $S'$  of MAX 2SAT as follows. Each  $C_i$  in  $S$  corresponds to a clause group  $C'_i$  in  $S'$  where

$$C'_i = \{(w_i), (x_i), (y_i), (z_i), (\overline{x_i} + \overline{y_i}), (\overline{y_i} + \overline{z_i}), (\overline{x_i} + \overline{z_i}), (\overline{w_i} + x_i), (\overline{w_i} + y_i), (\overline{w_i} + z_i)\}$$

where  $w_i$  is a new variable and  $1 \leq i \leq m$ . We set  $k = 7m$ .

Each clause in  $S'$  as constructed above has at most two literals. It can be seen that the clauses in  $S'$  can be efficiently generated from the clauses in  $S$  in polynomial time. We now argue that a satisfying truth assignment of  $S$  exists if and only if it can be extended to a satisfying truth assignment for  $S'$  appropriately.

Assume that  $S$  is satisfiable. Then in a typical clause  $C_i = \{x_i, y_i, z_i\}$  either one or two or all the three variables are true.

- a) Let  $x_i = T, y_i = F, z_i = F$ . With this truth assignment in  $S'$ , if  $w_i = T$ , six clauses of  $C'_i$  become true; if  $w_i = F$ , seven clauses of  $C'_i$  become true.
- b) Let  $x_i = T, y_i = T, z_i = F$ . This truth assignment in  $S'$  together with  $w_i = T$  or  $w_i = F$ , makes seven clauses of  $C'_i$  true.
- c) Let  $x_i = T, y_i = T, z_i = T$ . With this truth assignment in  $S'$ , if  $w_i = T$ , seven clauses of  $C'_i$  become true; if  $w_i = F$ , six clauses of  $C'_i$  become true.

In summary, a satisfying truth assignment of  $S$  can be extended to a satisfying truth assignment of  $S'$  where exactly seven clauses in each clause group get satisfied. Moreover no setting of  $w_i$  causes more than seven of the ten clauses to be true in each clause group in  $S'$ .

Now assume that  $S$  is not satisfiable. Then in at least one clause  $C_i = \{x_i, y_i, z_i\}$  in  $S$ , we have  $x_i = F, y_i = F, z_i = F$ . With this truth assignment in  $S'$ , if  $w_i = T$ , four clauses of  $C'_i$  become true; if  $w_i = F$ , six clauses of  $C'_i$  become true. That is, if  $S$  is not satisfiable, no truth setting can make at least seven clauses true in each clause group  $C'_i$ .

Since 3SAT is NP-complete, we conclude that MAX 2SAT is also NP-complete. □

## References

- [1] J. E. Hopcroft, R. Motwani, and J. D. Ullman. *Introduction to Automata Theory, Languages, and Computation*, third edition. Pearson Addison-Wesley, 2007.
- [2] M. R. Garey, D. S. Johnson, and L. Stockmeyer. Some simplified NP-complete graph problems. *Theo. Comp. Sci.*, 1(3):237-267, 1976.