MAX2SAT is NP-Complete

Rafael Accorsi
WINS - ILLC
University of Amsterdam
accorsi@wins.uva.nl

Amsterdam, April 27, 1999

Abstract

This hand-out provides a proof (according to the one presented in [Pap94]) that the problem MAX2SAT is a NP-complete problem. To do attain such result we start by giving some previous account on satisfiability and motivating the problem of MAX2SAT . In the sequel we present the required proof.

1 The problem

Before presenting the proof that MAX2SAT belongs to NP we shall introduce the problem we are dealing with. Consider the SAT problem; it maybe understood as a large placeholder to which one is able to attach various instances. As an example we can point out that 3SAT , 4SAT and HORNSAT are problems encompassed within SAT in the sense that SAT generalises these problems.

In searching for 2SAT we require that *every* clause in the formula φ must be satisfied by at least one of its compound variables, i.e, the truth assignment $\mathcal{V}$ satisfies at least one variable in each clause. Another way to see the problem is the following: given a formula φ in 2SAT and a constant k ($k \in \mathbb{N}$), is there any truth assignment $\mathcal{V}$ that satisfies k clauses ? This problem is called MAX2SAT and from now on we devote to the proof that such problem is in NP-complete[1].

2 The proof

Outline - the proof is carried out by presenting a clause that has a set of interesting *features* from which we will see that a reduction from 3SAT to MAX2SAT is suitable. Specifically, given any instance φ of 3SAT , we will

[1]Note that MAX2SAT is a generalisation of 2SAT since the latter is a special case of the former where k is equal to the number of clauses.

construct an instance $R(\varphi)$ of MAX2SAT . This procedure will produce the desired outcome, showing that MAX2SAT is indeed in NP-complete.

Proof - take the formula below.

$$\underbrace{(x)\wedge(y)\wedge(z)\wedge(w)}_{I}\wedge\underbrace{(\neg x\vee\neg y)\wedge(\neg y\vee\neg z)\wedge(\neg x\vee\neg z)}_{II}\wedge\underbrace{(x\vee\neg w_1)\wedge(y\vee\neg w_1)\wedge(z\vee\neg w_1)}_{III}$$

Is this formula satisfiable via some truth assignment? The answer is **no** since any attempt to satisfy the first group will automatically disable the second group. This happens due to the symmetric nature of the clauses w.r.t the variables x, y, z (but not w – why?).

2.1 Detecting properties

Let us now inspect clause and analyse some of its features. We start by testing some truth assignments.

1. Assume that x, y and z are satisfied – thus it is the case that none of the formulas in II is satisfied. Regarding to the third group, we see that satisfying w entails that the whole III is satisfied. Hence, the maximum number of clauses satisfied simultaneously is seven (the whole group I and III).

2. Assume that just two of the x, y and z are satisfied – thus it is the case that two of the clauses in II are satisfied. In satisfying w, one gains two clauses on III. Again, we end up with seven clauses being satisfied. *What happen when we set w to false ?*

3. Assume that just one of the variables x, y and z is satisfied – this entails that the whole II is satisfied and by satisfying w, two clauses of III group are satisfied as well. Summing up, we are again with at most seven clauses satisfied at the same time.

4. Assume that *none* of x, y and z is satisfied – thus it follows that all the II group is satisfied and by setting w to false we satisfy the entire group III. Well, this gives us *six* clauses satisfied simultaneously. Let us try another value to w, namely, make it true now. It follows that we gain one clause in I but we lose the entire III. This ends-up with just *four* clauses.

Now it is time to take stock and draw a general pattern and reasoning. Note that whenever one of x, y and z is satisfiable (or more than one), it follows that we can satisfy exactly seven clauses of the formula φ above with the same truth assignment. This suggests a connection between 3SAT and MAX2SAT using some *gadget* to enable this translation. In fact, this is the idea used in the sequel.

2.2 Using the gadget

Suppose that we have an instance of 3SAT , namely φ . We construct an instance MAX2SAT (called $R(\varphi)$) as described below. For each clause in φ we apply the following translation:

$$c_i = (\alpha \vee \beta \vee \gamma)$$
$$\Downarrow \text{goes to}$$
$$\underbrace{(\alpha)\wedge(\beta)\wedge(\gamma)\wedge(w')}_{I} \wedge \underbrace{(\neg\alpha\vee\neg\beta)\wedge(\neg\beta\vee\neg\gamma)\wedge(\neg\alpha\vee\neg\gamma)}_{II} \wedge \underbrace{(\alpha\vee\neg w')\wedge(\beta\vee\neg w')\wedge(\gamma\vee\neg w')}_{III}$$

where w' is a new variable which will be evaulated as specified later in the proof. One may easily notice that w' does not interfere in the satisfiability of x, y and z since we add clauses "$v \vee \neg w'$" for $v \in \{\alpha, \beta, \gamma\}$.

A piece of nomenclature shall be added: since each clause in φ is replaced by ten, we call the translation of each clause of φ a *group*. For instance, the formula depicted above constitutes a group extracted from the previous 3SAT clause. The fact that each clause is translated into ten entails that the final formula will have $10 \times n$ clauses where n is the original number of clauses in the 3SAT formula.

To satisfy a formula in nCNF we have to satisfy at least one variable in each clause. As our translation takes into account, say, m clauses and we have concluded that to satisfy x, y and z the number of clauses satisfied simultaneously have to be seven; the intention is to set $k = 7m$ (where k is the constant referred above).

2.3 Setting the objective

The *objective* is the following – $k = 7m \Leftrightarrow \varphi$ is satisfiable. We proceed as follows.

$\Rightarrow$ Assume that $7m$ clauses can be satisfied in $R(\varphi)$. We concluded before that it is the case that in each group, at least one of x, y and z is satisfied. It clearly implies that at least one variable is satisfied in each clause of φ . It follows that the whole φ is satisfied.

$\Leftarrow$ Conversely, any truth assignment that satisfies φ can be easily translated into one assignment that satisfies $7m$ clauses of $R(\varphi)$. To attain such result one has just to define the truth value to w_i' according to how many variables are satisfiable.

Once we have defined the translation, it is clear that MAX2SAT is NP-complete since it may be reduced in polyonomial time to 3SAT , fullfilling the requirements to establish its inclusion in this class.

3 An example

We finish this exposition by witnessing an example showing that we indeed attain the result being claimed.

Consider the following formula φ .

$$\varphi = \underbrace{(x \vee y \vee \neg z)}_{Group_1} \wedge \underbrace{(x \vee \neg y \vee z)}_{Group_2}$$

One translates it into the following $R(\varphi)$

$$\underbrace{(x)\wedge(y)\wedge(\neg z)\wedge(w_1)\wedge(\neg x\vee\neg y)\wedge(\neg y\vee z)\wedge(\neg x\vee z)\wedge(x\vee\neg w_1)\wedge(y\vee\neg w_1)\wedge(\neg z\vee\neg w_1)}_{Group_1}$$

$$\underbrace{(x)\wedge(\neg y)\wedge(z)\wedge(w_2)\wedge(\neg x\vee y)\wedge(y\vee\neg z)\wedge(\neg x\vee\neg z)\wedge(x\vee\neg w_2)\wedge(\neg y\vee\neg w_2)\wedge(z\vee\neg w_2)}_{Group_2}$$

Now we show that the claim is fullfilled.

$\Rightarrow$ φ is satisfiable $\rightarrow k = 7m$ Suppose that x is satisifed and that y and z are **not** satisfied:
First group – In setting w_1 to 1, the group one presents exactly seven clauses being satisfied.$\surd$
Second group – It is clear that the second group also presents seven clauses being satisfied.$\surd$
Conclusion – we have exactly 14 clauses being satisfied.

$\Leftarrow$ suppose that we have 14 clauses being satisfied in $R(\varphi)$[2]. Our previous experience tell us that in each group at least one of x, y, or z is satisfied. Thus each clause of φ is satisfied and also the formula as a whole.

References

[Pap94] Papadimitriou, C. H.; *Computational Complexity.*Addison-Wesley, USA. 1994.

[2]Seven in each group.