

## Solution to Homework 25

**Problem.** Similar to what we did in the class, illustrate the general algorithm of reducing NP problems to satisfiability on the example of the following problem:

- given a bit  $x$ ,
- find a bit  $y$  for which the following formula is true:  $\neg x \vee y$ .

**Solution.**

**Computational device for checking the desired property.** In accordance with the above proof, we need to start with a computational device that, given  $x$  and  $y$ , checks whether  $x \vee \neg y$  is true. In the beginning, we have two cells: an  $x$ -cell that contains the input bit  $x$  and a  $y$ -cell which contains the bit  $y$ .

We also need a wire to transmit the information. We will thus send the content of the  $y$ -cell to the  $x$ -cell, and then use the  $x$ -cell to compare its original content with what is sent by wire. Once the  $y$ -signal is sent, we no longer need it, so we can simply erase it (i.e., replace it with 0).

The whole computation process takes 3 moments of time:

- at moment  $t = 1$ , the  $x$ -cell contains  $x$ , the  $y$ -cell contains  $y$ , and the wire is inactive;
- at moment  $t = 2$ , the  $x$ -cell still contains  $x$ , the  $y$ -cell now contains 0, and the wire transmits the  $y$  signal;
- at moment  $t = 3$ , the  $x$ -cell contains 1 if  $\neg x \vee \neg y$  and 0 otherwise, the  $y$ -cell contains 0, and the wire is again inactive.

Similarly to the example from the lecture, in this computation process, we have 3 cells: the  $x$ -cell, the  $y$ -cell, and the wire. The  $x$ -cell has 2 possible states: 0 and 1, so one bit is sufficient to describe its state. According to the general notation, we will denote the state of this bit at moment  $t$  by  $s_{1,1,t}$ . Similarly, to describe the state of the  $y$ -cell, we need one bit  $s_{2,1,t}$ .

The wire can be in 3 possible states: inactive, sending 0, and sending 1. Thus, to describe the state of the wire, we will need 2 bits. Let the first bit describe whether the wire is active or not, and the second bit describe the signal sent via an active wire. So, the state  $S_3$  of the wire is either 00 (inactive), or 10 (sending 0), or 11 (sending 1).

In this case,  $S = 3$ , and the number of bits  $B$  needed to describe the state of each of the cells is  $B = 2$ .

**Corresponding dynamics of states.** Let us describe the above computations in terms of changing states.

At the first moment of time, the wire is inactive:  $s_{3,1,1} = s_{3,2,1} = 0$ .

At the second moment of time, the first cell retains its state, i.e.,  $s_{1,1,2} = s_{1,1,1}$ . The second cell becomes 0:  $s_{2,1,2} = 0$ . The wire becomes active:  $s_{3,1,2} = 1$ , and the signal it transmits is exactly the bit originally stored in the  $y$ -cell:  $s_{3,2,2} = s_{2,1,1}$ .

At the third moment of time, the  $x$ -cell gets the value 1 if the property  $\neg x \vee y$  is true, where:

- $x$  is the same initial  $x$ -state (since we did not change it), i.e.,  $x = s_{1,1,2}$ , and
- $y$  is the state passed through the wire, i.e.,  $y = s_{3,2,2}$ .

Thus,  $s_{1,1,3} = 1 \Leftrightarrow (\neg s_{1,1,2} \vee s_{3,2,2})$ . The  $y$ -cell still contains 0:  $s_{2,1,3} = 0$ , and the wire is again inactive:  $s_{3,1,3} = s_{3,2,3} = 0$ .

**Describing the dynamics in CNF terms.** The above formulas have the form  $a = 0$ , etc., for some variables  $a$ . So, to describe the above formulas in the CNF terms, we need to translate the following general formulas into CNF:  $a = 0$ ,  $a = 1$ ,  $a = b$ , and  $a = 1 \Leftrightarrow (b \vee \neg c)$ . Once we do that, we will be able to translate specific formulas by plugging the specific name of the variable  $a$  into the corresponding CNF formula.

We already know, from the example presented in the handout, that:

- the CNF form of the formula  $a = 0$  is  $\neg a$ ;
- the CNF form of the formula  $a = 1$  is  $a$ ; and
- the CNF form of the formula  $a = b$  is  $(a \vee \neg b) \& (\neg a \vee b)$ .

Let us use the general algorithm to translate the remaining formula  $a = 1 \Leftrightarrow (\neg b \vee \neg c)$  into CNF.

**Translating  $a = 1 \Leftrightarrow (\neg b \vee \neg c)$  into CNF.** For the formula  $a = 1 \Leftrightarrow (\neg b \vee \neg c)$ , the truth tables for formula  $F$  itself and for its negation  $\neg F$  take the form

$a$	$b$	$c$	$F$	$\neg F$
0	0	0	0	1
0	0	1	0	1
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	1	0
1	1	1	0	1

The corresponding DNF form for  $\neg F$  is

$$(\neg a \ \& \ \neg b \ \& \ \neg c) \vee (\neg a \ \& \ \neg b \ \& \ c) \vee (\neg a \ \& \ b \ \& \ c) \vee (a \ \& \ b \ \& \ \neg c),$$

so its negation  $F$  takes the CNF form

$$(a \vee b \vee c) \ \& \ (a \vee b \vee \neg c) \ \& \ (a \vee \neg b \vee \neg c) \ \& \ (\neg a \vee \neg b \vee c).$$

This means that the formula  $s_{1,1,3} = 1 \Leftrightarrow (\neg s_{1,1,2} \ \& \ \neg s_{3,2,2})$  takes the form

$$\begin{aligned} & (s_{1,1,3} \vee s_{1,1,2} \vee s_{3,2,2}) \ \& \ (s_{1,1,3} \vee s_{1,1,2} \vee \neg s_{3,2,2}) \ \& \\ & (s_{1,1,3} \vee \neg s_{1,1,2} \vee \neg s_{3,2,2}) \ \& \ (\neg s_{1,1,3} \vee \neg s_{1,1,2} \vee s_{3,2,2}). \end{aligned}$$

**The resulting long formula.** The resulting formula should include:

- the CNF forms of all the formulas describing the state's dynamics,
- the fact that the initial value  $x$  is given; for example, for  $x = 0$ , it should be  $s_{1,1,1} = 0$ , i.e.,  $\neg s_{1,1,1}$ ; and
- the fact that the result of checking the property  $C(x, y)$  is “true”; according to our computation scheme, this result is stored in the  $x$ -cell at moment 3, so this requirement takes the form  $s_{1,1,3} = 1$ , i.e., in CNF form, as  $s_{1,1,3}$ .

Thus, the corresponding long formula takes the following form:

$$\begin{aligned} & \neg s_{3,1,1} \ \& \ \neg s_{3,2,1} \ \& \\ & (s_{1,1,2} \vee \neg s_{1,1,1}) \ \& \ (\neg s_{1,1,2} \vee s_{1,1,1}) \ \& \\ & \neg s_{2,1,2} \ \& \ s_{3,1,2} \ \& \\ & (s_{3,2,2} \vee \neg s_{2,1,1}) \ \& \ (\neg s_{3,2,2} \vee s_{2,1,1}) \ \& \\ & (s_{1,1,3} \vee \neg s_{1,1,2} \vee s_{3,2,2}) \ \& \ (s_{1,1,3} \vee s_{1,1,2} \vee s_{3,2,2}) \ \& \\ & (s_{1,1,3} \vee s_{1,1,2} \vee s_{3,2,2}) \ \& \ (s_{1,1,3} \vee s_{1,1,2} \vee \neg s_{3,2,2}) \ \& \\ & (s_{1,1,3} \vee \neg s_{1,1,2} \vee \neg s_{3,2,2}) \ \& \ (\neg s_{1,1,3} \vee \neg s_{1,1,2} \vee s_{3,2,2}) \ \& \\ & \neg s_{1,1,1} \ \& \ s_{1,1,3}. \end{aligned}$$

This formula says that for given  $x = 0$  and for some  $y$ , we performed the checking of the property  $C(x, y) \equiv (x \vee \neg y)$  and concluded that the result of checking is “true”. Once the formula is satisfied, we can find  $y$  as the original value of the  $y$ -cell, i.e., as  $y = s_{2,1,1}$ .