# Automata, Computability, and Complexity sheet 12

Hamza Bouhelal, Mohamed Reda Arsalane, Mahdi Ouchrahou

SHEET #10:

## Exercice 1:

doubleSAT, like any variant of SAT, is in NP since the truth assignment is the certificate,  we can check every clause in polynomial time to see if it is satisfied.

We now show a reduction from 3-SAT.
Given a 3-SAT formula f, we construct a doubleSAT formula

f' = f $\wedge$ (xk $\vee$ - xk),

where x  is a variable not used in f. Clearly this reduction is poly-time.

We now show that f has a satisfying assignment iff f' has two satisfying assignments.

=>
If f has a satisfying assignment , then f' has two satisfying assignments, corresponding to

- case 1: if f = true and xk = true ((xk $\vee$ - xk) always = 1)
- case 2: if f = true and xk =false ((xk $\vee$ - xk) always = 1)

<=
The clause (xk  xk) can be satisfied with either xk =true or xk =false. In either case, if we have a satisfying assignment  for f, this ensures DOUBLE-SAT is validated. In this case, then 3-SAT is also validated.

We can therefore conclude that doubleSAT is NP-complete.

## Exercice 3:

Assume that P=NP. Let B $\in$ P such that B $\neq \varnothing$ and B $\neq \Sigma^*$. This means that there is a string wIN$\in$ B and a string wOUT $\notin$ B. We want to show

that B is NP-complete. Surely, the language B is in NP=P (by our assumption). We have to show that B is NP-complete.
Let A be an arbitrary language from NP=P. Hence A has a polynomial time decider MA. We need to argue that $A \leq_p B$. Here is a poly-time reduction f from A to B:

```
”On input w:
    1. Run MA (decider for A) on w.
    2. If MA accepted then output wIN If MA rejected then output wOUT.”
```

This is a poly-time reduction from A to B, and hence B is NP-complete.

**<u>Exercice 4:</u>**
if we replace the 2x3 window by a 2x2 window the proof will fail since the encoding of configurations can change in 3 different places with a single transition. This is due to the fact that the head will arrive to a certain position i by either the next-left position i-1 or the nest-right position i+1.

We take the example of a non deterministic decider N with configuration UUVWUU,
The head tape is on symbol W as follows UUVqWUU.
We assume to have the transition δ(q,W) = {(q’, X, L)}, meaning that X is replacing W in the tape, head moves to the left and new state change to q’.
We get a new configuration UUqVXUU.
We can see that the change is in 3 consecutive places.
If we index the original configuration from 0 to 6, changes occure in index 2, 3 and 4.
We can illustrate the situation in these two configurations index tables :

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| | U | U | V | q | W | U | U |
| index | 0 | 1 | 2 | 3 | 4 | 5 | 6 |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| | U | U | q | V | X | U | U |
| index | 0 | 1 | 2 | 3 | 4 | 5 | 6 |