Exercise 1 : Machines with certificates

We consider a variant of Turing Machines that are like usual deterministic TMs except that they also carry an extra read-only input tape (not a worktape) called a certificate tape that is read-once (on this tape the reading head may only move rightwards or stay motionless). We write $M(x, u)$ for the output of such a machine $M$ started with $x$ on input tape and $u$ on the certificate tape. We say $M$ runs in space $f(n)$ if for any input $x$ of size $n$ and any certificate $u$, the machine halts and uses at most $f(n)$ cells on its worktapes.

We now consider the class $C$ of all languages $L$ such that there exists a polynomial $p$ and a deterministic TM-with-certificate running in logspace such that

$$x \in L \iff \exists u : |u| \leq p(|x|) \text{ and } M(x, u) \text{ accepts.}$$

(* ) 1. Show that $C$ coincides with $NL$.

(**) 2. Let us now assume that TMs with certificate are allowed to move leftwards on the certificate tape (hence $u$ can be read several times but the certificate tape is still read-only) and define $C$ as previously.

Prove that $C$ coincides with one of $NL$, $PTIME$, $NP$ or $PSPACE$ (and tell which one).

Exercise 2 : Odd satisfiability?

Recall that ParitySat asks, given a list $I = \langle \varphi_1, \ldots, \varphi_m \rangle$ of boolean formulas, whether the number of satisfiable $\varphi_i$’s is odd. (The size $n \eqdef |I|$ of an instance is $\sum_{i=1}^m |\varphi_i|$.)

We are interested in a variant problem : ParityFirstSat asks, given $I$ as above, whether the smallest index $i \in \{1, \ldots, m\}$ such that $\varphi_i$ is satisfiable exists and is odd.

(* ) 3. Show $\text{ParityFirstSat} \leq \text{ParitySat}$. (As usual “$\leq$” denotes logspace reducibility.)
4. Does there exist a total function $\pi$ which, given a pair $(I, k)$ where $I$ is a list $\langle \varphi_1, \ldots, \varphi_m \rangle$ of boolean formulas and $k \in \mathbb{N}$ is some number written in binary, returns a boolean formula $\pi(I, k) = \psi$ such that $\psi$ is satisfiable if, and only if, at least $k$ formulas in $I$ are satisfiable?

5. Show $\text{ParitySat} \leq \text{ParityFirstSat}$.

6. Show that $\text{ParitySat}$ and $\text{ParityFirstSat}$ are in $\mathbb{P}^{\mathbb{NP}[O(\log n)]}$ (or in $\mathbb{P}^{\mathbb{NP}[O(\log n)]}$ if you prefer : we saw that these two classes coincide).

We now consider a new problem, $\text{SeqSatProgram}$, where one has to solve a sequence of dependent satisfiability problems. Formally, an instance is a “list” of the form

$$x_1 := \exists Y_1 : \varphi_1(Y_1)$$
$$x_2 := \exists Y_2 : \varphi_2(x_1, Y_2)$$
$$x_3 := \exists Y_2 : \varphi_3(x_2, Y_3)$$
$$\vdots$$
$$x_m := \exists Y_m : \varphi_m(x_{m-1}, Y_m)$$

where each $Y_i$ is a set of boolean variables disjoint from $X = \{x_1, \ldots, x_m\}$, and each $\varphi_i$ is a boolean formula with all its variables in $Y_i \cup \{x_{i-1}\}$. The meaning is that the $x_i$’s should be computed in turn. When computing $x_i$, one has to solve a satisfiability problem that depends on the value of $x_{i-1}$.

Seen as a decision problem, the question is whether $x_m$, the last computed variable, evaluates to true or false. It is clear that $\text{SeqSatProgram}$ is in $\mathbb{P}^{\mathbb{NP}}$.

7. Show that $\text{SeqSatProgram}$ is in $\mathbb{P}^{\mathbb{NP}[O(\log n)]}$.