Assignment 3 solutions

Exercise 5.2

The computation of an ATM can be modelled by a tree, where each node at distance i from the root corresponds to a snapshot of the machine after i steps, the node is labeled by $\exists$ or $\forall$, and there are two children corresponding to the two possible choices made by the ATM. We can arrange the tree nodes into layers, where each layer has the same label ($\exists$ or $\forall$). An accepting computation corresponds to a subtree that includes one child of each $\exists$ node and all children of every $\forall$ node, and all its leaves are in the accepting state of the ATM. Then it is clear that $L \in \cup_c \Sigma_i TIME(n^c)$ ($\cup_c \Pi_i TIME(n^c)$) iff there is an ATM whose computation for every x corresponds to a tree of polynomial height, with the root labeled $\exists(\forall)$, and exactly $i-1$ alternations of the label on its layers.

On the other hand, $L \in \Sigma_i^p$ can be computed by an ATM that *'guesses'* certificates $u_1, u_2, \ldots, u_i$ using its transition function choices, by recording whether the choice was 0 or 1. While guessing u_1 it is in $\exists$ states, while guessing u_2 it is in $\forall$ states, etc. After recording $u_1, u_2, \ldots, u_i$ on its working tape, it runs $M(x, u_1, u_2, \ldots, u_i)$ for a polynomial time keeping the state label of u_i. (Symmetric for Π_i^p.)

(a) The computation of the ATM for $L \in \Sigma_i^p$ above corresponds to a labeled tree, with the root labeled $\exists$, the first $|u_1|$ layers labeled $\exists$, followed by $|u_2|$ layers labeled $\forall$, etc. After $|u_1|+|u_2|+\ldots+|u_i|$ layers, follow polynomial many layers *without transition branchings*, labeled with the same label as the u_i layers, that correspond to the computation of the deterministic $M(x, u_1, u_2, \ldots, u_i)$. Obviously the tree has polynomial height. This tree corresponds exactly to the computation of an ATM that decides a $\cup_c \Sigma_i TIME(n^c)$ language, i.e., $L \in \cup_c \Sigma_i TIME(n^c)$.

(b) To show that $\cup_c \Sigma_i TIME(n^c) \subseteq \Sigma_i^p$, we can assign to the j-th block of consecutive layers with the same label of the ATM M computation tree for $\cup_c \Sigma_i TIME(n^c)$ a *certificate* u_j with the quantifier that labels the block. This certificate records the transition function choices made inside this block of layers. Then there is an acceptance subtree of this ATM tree for input x iff the statement $\exists u_1 \forall u_2 \ldots Q u_i : \ M(x, u_1, u_2, \ldots, u_i) = 1$ is true.

Exercise 5.12

Assuming that $\Sigma_i^p = \Pi_i^p$, we use induction on j to prove that $\Sigma_j^p, \Pi_j^p \subseteq \Sigma_i^p$.

$j = i$**:** True by hypothesis.

$j = k-1$**:** We assume that $\Sigma_{k-1}^p, \Pi_{k-1}^p \subseteq \Sigma_i^p$.

$j = k$**:** We show that $\Sigma_k^p \subseteq \Sigma_i^p$. Let $L \in \Sigma_k^p$. Then

$$x \in L \Leftrightarrow \exists u_1 \forall u_2 \ldots Q_{k-i+1} u_{k-i+1} Q_{k-i+2} u_{k-i+2} \ldots Q_k u_k : \ M(x, u_1, \ldots, u_k) = 1$$

The statement $Q_{k-i+1} u_{k-i+1} Q_{k-i+2} u_{k-i+2} \ldots Q_k u_k : \ M(x, u_1, \ldots, u_k) = 1$ is in Σ_i^p if $Q_{k-i+1} = \exists$ or Π_i^p if $Q_{k-i+1} = \forall$, so it can replaced by an equivalent statement

$\bar{Q}_{k-i+1}u_{k-i+1}\bar{Q}_{k-i+2}u_{k-i+2}\ldots\bar{Q}_k u_k : M'(x,u_1,\ldots,u_k) = 1$ in Π_i^p or Σ_i^p respectively, by hypothesis. Since $\bar{Q}_{k-i+1} = Q_{k-i}$ (i.e., $Q_{k-i}u_{k-i}, \bar{Q}_{k-i+1}u_{k-i+1}$ collapse to $Q_{k-i}u_{k-i}u_{k-i+1}$), the new (equivalent) statement is in $\Sigma_{k-1}^p \subseteq \Sigma_i^p$.

We can similarly prove that $\Pi_k^p \subseteq \Sigma_i^p$, since $\Pi_{k-1} = \overline{\Sigma_{k-1}^p} \subseteq \overline{\Sigma_i^p} = \Pi_i^p = \Sigma_i^p$.

Exercise 5.3

(a) If $R(x)$ is the polynomial-time reduction $L \leq_P \bar{L}$, we have $x \in L \Leftrightarrow R(x) \in \bar{L}$, or, equivalently, $x \notin L \Leftrightarrow R(x) \notin \bar{L}$, or, equivalently, $x \in \bar{L} \Leftrightarrow R(x) \in L$, i.e., $R(x)$ is also the reduction $\bar{L} \leq_P L$.

(b) Since we have assumed that $3SAT \leq_p \overline{3SAT}$, (a) implies that we have also $\overline{3SAT} \leq_p 3SAT$. Let $L \in \Sigma_2^p$. Then

$$x \in L \Leftrightarrow \exists u_1 \forall u_2 : M(x,u_1,u_2) = 1. \tag{1}$$

For a fixed certificate u_1, the language L' defined by

$$x \in L' \Leftrightarrow \forall u_2 : M(x,u_1,u_2) = 1$$

is in $\Pi_1^p = coNP$, and since $\overline{3SAT}$ is $coNP$-complete, we have $L' \leq_p \overline{3SAT} \leq_p 3SAT$, i.e., there is polynomial TM M' such that

$$\forall u_2 : M(R(x),u_1,u_2) = 1 \Leftrightarrow x \in L' \Leftrightarrow \exists u_2 : M'(S(R(x)),u_1,u_2) = 1,$$

where R, S are the two polynomial reductions, and (1) becomes

$$x \in L \Leftrightarrow \exists u_1 \exists u_2 : M'(x,u_1,u_2) = 1,$$

which implies that $L \in \Sigma_1^p$.

(c) The proof goes exactly like in (b), except that we use the hypothesis $3SAT \leq_p \overline{3SAT}$.

(d) We have $\Pi_1^p \subseteq \Sigma_2^p = \Sigma_1^p$ and $\Sigma_1^p \subseteq \Pi_2^p = \Pi_1^p$, and, therefore, $\Pi_1^p = \Sigma_1^p$. Theorem 5.4 implies the exercise result.

Exercise 5.7

To show $APSPACE \subseteq EXP$, it is enough to see that we can construct the configuration graph for the ATM for input x, and then check if there is an accepting computation by applying the rules of Definition 5.7.

To show $EXP \subseteq APSPACE$, first note that the computation of a TM for an $L \in EXP$ uses $2^{p(n)}$ time and $2^{p(n)}$ space, for a polynomial p, and can be represented by a $2^{p(n)} \times 2^{p(n)}$ matrix. Like in the Cook-Levin theorem, or every ($\forall$) cell (i,j) of this matrix, we can 'guess' ($\exists$) its content σ, and the contents of cells $(i-1,j-1),(i-1,j),(i-1,j+1)$ (let's say $\sigma_1,\sigma_2,\sigma_3$), and verify that σ is compatible with $\sigma_1,\sigma_2,\sigma_3$ going from time step $i-1$ to time step i (using constant space and time). *After* this verification is done, we verify *recursively* the computation for all ($\forall$) three $\sigma_1, \sigma_2, \sigma_3$. For the verification of $\sigma_1, \sigma_2, \sigma_3$, we *reuse* the space used for σ to keep σ_1, then σ_2, then σ_3 in each recursive call. Extending this reuse all the way down the recursion tree, the only space we need is to index cell (i, j) (which takes $O(\log 2^{p(n)}) = O(p(n))$ space) and some constant space for verification computations.

Note that this description corresponds to an ATM computation tree, which, due to space reuse, takes polynomial space, proving the result.