Solutions to Problem Set 10

1. (a) Show that TQBF is complete for **PSPACE** also under logspace reductions.
   (*Hint:* The solution is not lengthy or tedious. Do not try to give the full logspace reduction. Instead, take a second look at the reduction done in class.)

   (b) Show that $TQBF \notin \mathbf{NL}$.

   [20 + 10 = 30 points]

   SOLUTION: We look at the proof of **PSPACE** hardness of TQBF and show that the reduction can be carried out in logspace. The reduction consists of the following steps

   (a) Start with $t = 2^{n^k}$ if the given machine uses space n^k.

   (b) Start with the formula expressing reachability of the final state from the starting state: $\phi_{c_{start},c_{accept},t}$.

   (c) Recursively simplify

   $$\phi_{c_1,c_2,t} = \exists m_1 \forall (c_3, c_4) \in \{(c_1, m_1), (m_1, c_2)\}[\phi_{c_3,c_4,t/2}]$$

   (d) Finally, express $\phi_{c_1,c_2,1}$ by the constraints that if c_1 and c_2 are two configurations then the transition function of the machine correctly leads from c_1 to c_2.

   We now see that each step can be performed in logspace:

   (a) In the first step, we simply need to write 1 followed by n^k zeros. But note that t is only needed to carry out the reduction so that we can check how much more do we need to simplify the formula. We can thus maintain $\log t$ on the scratch tape (which takes $k \log n$ bits) since we are reducing t by 1/2 at every step.

   (b) We do not need to explicitly write the formula for t and erase and replace by the one for $t/2$. We can just simplify "on the go" by simply writing $\exists m_1 \forall (c_3, c_4) \in \{(c_1, m_1), (m_1, c_2)\}$ and then decrementing the counter for $\log t$, since we know this must be followed by the formula for $t/2$.

   (c) $\phi_{c_1,c_2,1}$ can be written in logspace, since the action of the transition function (which is constant sized) on the symbol at a particular location on the tape. The location is between 1 and n^k and can be specified in logspace.

   For the second part, note that $TQBF \in \mathbf{NL}$ would imply that $\mathbf{PSPACE} \subseteq \mathbf{NL}$, since we showed that all problems in **PSPACE** reduce to TQBF through logspace reductions. However, by the hierarchy theorems, we know that

   $$\mathbf{NL} = SPACE(\log^2 n) \subsetneq \mathbf{PSPACE}$$

2. Consider the function $pad : \Sigma^* \times \mathbb{N} \to \Sigma^*\#^*$ defined as $pad(s, l) = s\#^j$, where $j = \min(0, l - |s|)$. Thus, $pad(s, l)$ just adds enough copies of the new symbol # to the end of the string s so that the length of the new string is at least l. For a language A and a function $f : \mathbb{N} \to \mathbb{N}$, define the language $pad(A, f(n))$ to be

   $$pad(A, f(n)) = \{pad(s, f(|s|)) \mid s \in A\}$$

(a) Prove that if $A \in \mathbf{TIME}(n^6)$, then $pad(A, n^2) \in \mathbf{TIME}(n^3)$.
(*Note:* This part will not be graded as we proved this in section. You need not submit the solution to this, but you can attempt this part to understand the definition.)

(b) (Sipser 9.14) Define $\mathbf{EXPTIME} = \mathbf{TIME}(2^{n^{O(1)}})$ and $\mathbf{NEXPTIME} = \mathbf{NTIME}(2^{n^{O(1)}})$. Use the function pad to prove that

$$\mathbf{NEXPTIME} \neq \mathbf{EXPTIME} \Rightarrow \mathbf{P} \neq \mathbf{NP}$$

[15 points]

SOLUTION:

(a) Let M be the machine that decides A in time n^6. Now, consider the machine M' for $pad(A, n^2)$ that on input x, check if x is of the format $pad(w, |w|^2)$ for some string $w \in \Sigma^*$. If not, reject. Otherwise, simulate M on w. The running time of M' is $O(|x|^3) + O(|w|^6) = O(|x|^3)$.

(b) We shall prove the contrapositive. Suppose that $\mathbf{P} = \mathbf{NP}$. Then, consider any language $L \in \mathbf{NEXPTIME}$, and let c be a positive integer such that $L \in \mathbf{NTIME}(2^{n^c})$. Then, it is easy to see that $pad(L, 2^{n^c}) \in \mathbf{NP}$. By assumption, $\mathbf{P} = \mathbf{NP}$, so $pad(L, 2^{n^c}) \in \mathbf{P}$ and therefore $L \in \mathbf{TIME}(2^{O(n^c)}) \subseteq \mathbf{EXPTIME}$. It follows that $\mathbf{EXPTIME} = \mathbf{NEXPTIME}$.

3. Recall that we defined $\mathbf{IP}$ as the class of languages A, such that for a polynomial time verifier V and provers P

$$w \in A \Rightarrow \exists P \; \mathbf{Pr}[V \leftrightarrow P \text{ accepts } w] = 1$$
$$w \notin A \Rightarrow \forall P \; \mathbf{Pr}[V \leftrightarrow P \text{ accepts } w] \leq 1/2$$

(a) Let $\mathbf{IP}'$ be the class of languages where we allow the prover to be probabilistic i.e. the prover can use randomness. Show that $\mathbf{IP}' = \mathbf{IP}$.

(b) Let $\mathbf{IP}'$ be the class of languages where we replace the $1/2$ in the definition above by 0 i.e. the verifier must surely reject in case $w \notin A$. Show that $\mathbf{IP}' = \mathbf{NP}$.

[7 + 8 = 15 points]

SOLUTION:

(a) Since we allow the prover to be computationally unbounded, a probabilistic prover can be easily simulated by a deterministic prover which considers all possible values of the provers randomness and the verifier's responses on each, and then chooses the best. Hence, a probabilistic prover is no more (and also no less!) powerful than a deterministic prover which implies that $\mathbf{IP}' = \mathbf{IP}$.

(b) Let r be the randomness used by the verifier. If the verifier accepts a correct proof with probability 1 and a wrong proof with probability 0, it must accept a correct proof for every r and reject a wrong proof for every fixed r. But then, the verifier is no more powerful than a deterministic verifier. However, we saw in class that the class of languages which can be checked by a deterministic polynomial time verifier equals $\mathbf{NP}$.