Complexity Theory
Problem Set 1

1. Design TMs for functions $f(n_1, n_2) = n_1 + n_2$ and $f(n) = n^2$.

2. Prove that functions n^2 and $n\lfloor \log n \rfloor + 1$ are time-constructible.

3. Define a bidirectional TM to be a TM whose tapes are infinite in both directions. For every $f : \{0,1\}^* \to \{0,1\}^*$ and time-constructible $T : \mathbb{N} \to \mathbb{N}$, if f is computable in time $T(n)$ by a bidirectional TM M, then it is computable in time $O(T(n))$ by a standard (unidirectional) TM M'. *(Claim 1.8 from Arora-Barak)*

4. Define a TM M to be oblivious if M halts after the same number of steps on inputs of same length and for every input $x \in \{0,1\}^*$ and $i \in \mathbb{N}$, the location of each of M's heads at the ith step of execution on input x is only a function of $|x|$ and i. Show that for every time-constructible $T : \mathbb{N} \to \mathbb{N}$, if $L \in \mathrm{DTIME}(T(n))$, then there is an oblivious TM that decides L in time $O(T(n)^2)$. *(Exercise 1.5 from Arora-Barak.)*

5. Show that the following languages are undecidable:

a) *AcceptTM* = $\{\langle M, x \rangle \mid M$ is a TM that accepts* $x\}$

b) *ReverseTM* = $\{\langle M \rangle \mid M$ is a TM that accepts $reverse(w)$ if it accepts $w\}$

c) *ExactOneTM* = $\{\langle M \rangle \mid M$ is a TM that accepts exactly one input$\}$

*We say a TM M accepts x if M on input x halts with a 1 on the output tape. Similarly, we say a TM M rejects x if M on input x halts with a 0 on the output tape.)

6. Let $L_1, L_2 \in \mathsf{NP}$. Are $L_1 \cup L_2$ and $L_1 \cap L_2$ also in NP? Prove your answer.

Solutions

1. Skipped as the solutions are easy but tedious. You can search online for TMs that add or multiply.

2. Skipped. It is easy to see that the functions are time-constructible if we know that $f(n) = n$ is time-constructible.

3. Read the proof of Claim 1.8 from the book.

4. Suppose L has a TM M that decides it and runs in time $T(n)$, where T is time-constructible. We will construct an oblivious TM M' that decides L and runs in time $T(n)^2$. M' will have as many tapes as M and a few more. Alphabet Γ' of M' will contain symbols c and $\hat{c}$ for every symbol c of alphabet of M. The presence of $\hat{c}$ for every c will help in keeping track of tape-heads.

M' on input x will work in the following manner:

1. M' will first compute $T(|x|)$ on a work tape. (Since T is time-constructible it can be done in time $O(T(|x|))$.)
2. M' will simulate one step of M in one back and forth sweeps from the left-most cell to the $T(|x|)$th cell. Current cells have symbols $\hat{c}$ instead of just c to indicate the tape-head in over them.
    1. While going from left-most cell to the $T(|x|)$th cell, M' will collect the current symbols. M' will need the computed value of $T(|x|)$ to do this operation.
    2. Then it will use δ of M to determine the modifications, the head movements, and the next state (we can assume the states are stored in some other tape).
    3. Finally, M' will come back to the left-most cell while making necessary head changes using the ˆ symbol.

5. a) We can reduce *HALT* to *AcceptTM*.
The function f maps $\langle M, x\rangle$ to the $\langle M', x'\rangle$, such that M' on input x' starts simulating M on x. If M ever halts on x, then M' accepts x', otherwise it keeps simulating M on x.

Clearly, if M halts on x, then M' accepts x'. Else, M' does not accept x'.

b) We can reduce *HALT* to *ReverseTM*.
The function f maps $\langle M, x\rangle$ to the TM M', such that M' rejects all the inputs immediately except when the input is 10 and 01. When the input is 10 it accepts it immediately. When the input is 01 it starts simulating M on x. If M ever halts on x, M' will accept 01. Else, M' will continue the simulation

Clearly, if M halts on x, then M' accepts both 01 and 10. If M does not halt on x, then M' accepts only 10.

c) We can reduce *HALT* to *ExactOneTM*.
The function f maps $\langle M, x \rangle$ to the TM M', such that M' rejects all the inputs immediately except when the input is 1. When the input is 1, M' starts simulating M on x. If M ever halts on x it accepts 1, otherwise it keep simulating M on x.

Clearly, if M halts on x, then M' accepts only 1. If M does not halt on x, then M' accepts no inputs.

6. Let M_1 and M_2 be NTMs for L_1 and L_2, respectively. We can construct NTM M for $L_1 \cap L_2$ in the following way. M on input x will first simulate M_1 on x. If M_1 rejects x along some computation paths during the simulation, then M will also halt immediately and reject x. If M_1 accepts x along some computation paths during the simulation, then M starts simulating M_2 on x. Finally, M will accept only along those computation paths where M_2 accepts x.

Construction of NTM for $L_1 \cup L_2$ is similar.