CSE 105: Automata and Computability Theory **Autumn 2014**

Homework #4 Solutions

ANDREAS PRODROMOU A53049230

Collaborated with: NONE

Solution to Problem 1

Assume that language A is decidable. This means $\exists$ TM M which decides A. For all $w \in \Sigma^*$, if $w \in A$ then M(w) accepts, otherwise M(w) rejects.

By definition, a string W $\in$ A iff W = ε or W = $X_1 X_2 \ldots X_k$ for some k and $X_{1\ldots k}$ are non-empty and belong in A. To show that A^* is decidable, I will define a new TM N that decides A^*. This means that for a given string W $\in \Sigma^*$, N(W) accepts if $W \in A*$ and rejects otherwise.

N = On input W:
    If W = ε accept
    Let n be the length of W
    For k = 1 to n do
        For each possible $X_1, X_2, \ldots, X_k \in \Sigma^* - \{\varepsilon\}$, where $W = X_1 X_2 \ldots X_k$ do
            Run $M(X_i)$ for i = 1 to k
            If all computation accept, then accept
            If any computation rejects, reject

Solution to Problem 2

Assume languages A and B are recognizable and M_A and M_B are the TMs that recognize them. I will define a new TM N that recognizes the intersection of A and B.

N = On input w:
    Run M_A on w. If it rejects, reject.
    If M_A accepts w, Run M_B on w. If it rejects, reject. If it accepts, accept.

The new machine will accept a string iff M_A accepts it first and then M_B accepts it too. Consequently, N recognizes the language $A \cap B$.

Solution to Problem 3

We can observe that we can simulate a DFA using an always-right TM. We simply need to add transitions from the states in F to q_{accept} when blank is read and transitions from states outside of F to q_{reject} when blank is read. Consequently, DFAs, are a subset of always-right TMs.

Now assume a TM M = $(Q, \Sigma, \Gamma, \delta, q_0, q_{accept}, q_{reject})$. I will construct a DFA D = $(Q', \Sigma', \delta', q_0', F)$ that recognizes the same language, where: $Q' = Q$, $\Sigma' = \Sigma$ and $q_0' = q_0$. The transition function will be:

$$\delta'(q,s) = \begin{cases} q & \text{if q} \in \{q_{accept}, q_{reject}\} \\ q' & \text{where M has transition (q,s) to q'} \end{cases} \tag{1}$$

F has to be defined as the set containing q_{accept} and the set of states q $\in$ Q, q $\neq q_{accept}, q_{reject}$, suche that M starting at q and reading blanks, eventually enters q_{accept}.

I showed that DFAs are a subset of always-right TMs and that always-right TMs are a subset of DFAs. Consequently, always-right TMs are equivalent to DFAs.

Solution to Problem 4

In a machine that can never move left, we can observe that any symbols written by this machine will never be read again. I will define a new character which is not in Γ, which will be written on every transition. If that character is read at some state, I will know that the machine moved left and I will transition to the q_{reject} state. The new TM will be the same as the old, except the new symbol and the simple transition rule upon reading this symbol.

This new machine is a decider for L_4, since it will either accept if M accepts and only moves right, and it will reject if M moves left or attempts to loop.