Automata (Fall 2010) Qual Exam

Answer ALL questions

Question 1 (25 points)

Consider the descriptions of Turing machines for inputs over $\{0, 1\}$. Suppose the Turing machines states are taken from the set of natural numbers. Note: a Turing machine still has only finite number of states. But there is no limit to how many states a Turing machine has.

Is the set of all Turing machines countably infinite? or, uncountably infinite?

Detailed and careful justifications are required.

Answers: *Let T_k be the set of Turing machines with state sets taken from $\{1, 2, \ldots, k\}$ and with description lengths at most k. It is clear that T_k is a finite set. For an arbitrary n-state Turing machine T with description length m, we have $T \in T_{\max(m,n)}$. Therefore, the set of all Turing machines is given by $\bigcup_{k=1,2,\ldots} T_k$, which is countably infinite.*

Question 2

Let w be a string over the alphabet Σ. Consider the languge $L_w = \Sigma^* w$.

(a) (5 points) Explain the design of the simplest NFA M_w for L_w.

Answers: *Let $w = a_1 a_2 \ldots a_n$ where $a_i \in \Sigma$ for $i = 1, 2, \ldots, n$. Then the simplest NFA consists of states $\{1, 2, \ldots, n+1\}$ with the state 1 the starting state and state $n + 1$ the final state. The transitions are: $\delta(1, a_1) = \{1, 2\}$, $\delta(1, a) = \{1\}$ for $a \neq a_1$, $\delta(k, a_k) = \{k + 1\}$ for $k = 2, 3, \ldots, n$.*

(b) (25 points) Consider the subset construction for converting the NFA M_w to a DFA A_w. Based on the specific design of the language L_w and an analysis of the subset construction, how many states does A_w have? Carefully justify your answer.

Answers: *Let S be a subset of states generated by subset construction. Let j be the highest numbered state in S. That is, the current input ends with the suffix $a_1 a_2 \ldots a_{j-1}$. For $1 \leq i < j$, whether i is in S is decided by whether the current input ends with the suffix $a_1 a_2 \ldots a_{i-1}$, which can be determined as we already know the last $j - 1$ symbols of the current input, where $i < j$.*

That is, given that the highest numbered state in S is known, the rest of the states in S are forced. Therefore, for each $j \in \{1, 2, \ldots, n+1\}$, there is exactly one subset of state generated by the subset construction which highest numbered state is j. Hence, the DFA has $n+1$ states.

Question 3

Let $L = \{s_k \mid k = 0, 1, 2, \ldots\}$ where $s_0 = \epsilon$ and $s_k = s_{k-1}a^k b^k$ for $k > 0$.

(a) (15 points) Show that L is the intersection of two context-free languages.

Answers: *We run two PDAs to process an input. If both PDAs accept, then the string is accepted. One PDA checks to see that every maximal substring of a's must be followed by the same number of b's. The other PDA checks to see that the input string, if non-empty, starts with ab, ends with b's, and every maximal substring of b's in between must be followed by a maximal substring of a's of length one longer than that of the length of the b's substring.*

(b) (15 points) Is L context-free? Prove your answer.

Answers: *No. Proof sketch: Let p be the pumping constant. Consider $w = aba^2b^2 \ldots a^p b^p \in L$. Next consider the different ways of breaking up w into u, v, x, y, z. For each case, we pump down the string by taking $i = 0$.*

(c) (15 points) Is the complement of L context-free? Prove your answer.

Answers: *We modify the construction from part (a). The two deterministic PDAs are modified to do the opposite of its previous missions. Moreover, the string is accepted if either one of the two PDAs accept. Thus, the new design gives rise to one PDA as it combines nondeterministically the design of two PDAs. Therefore, the complement of L is context-free.*