CS331 Practice Problem Set #3

1. *[5+5+15 marks]* Let $L_1, L_2 \subseteq \Sigma^*$. The *right quotient* of L_1 by L_2 is defined as $L_1/L_2 = \{w \in \Sigma^* \mid \exists u \in L_2 \text{ such that } wu \in L_1\}$. As an example, if $L_1 = \{0^n1^n \mid n \geq 1\}$ and $L_2 = \mathbf{1}^*$, then $L_3 = L_1/L_2 = \{0^n1^m \mid n \geq 1, 0 \leq m \leq n\}$.

   (a) Give a CFG G (*NOT a PDA*) generating L_3, as defined in the above example.

   (b) Prove that $L(G) = L_3$ for the grammar G in part (a).

   (c) Show that if L_1 and L_2 are recursively enumerable languages, then so is L_1/L_2.

2. *[10+10] marks* Let $\Sigma = \{a, b\}$ and $L = \{babaabaaab \ldots ba^{n-1}ba^n \mid n \geq 1\}$.

   (a) Is L context-free?

   (b) Is $L^c = \Sigma^* - L$ context-free?

   In each case, you must prove your claim.

3. *[10 + 15 marks]* In class, we have seen a flow-chart like notation for composing simple Turing Machines (TM) into more complex ones. Specifically, we'll assume the following notation:

   - $\mathbf{R}^{(i)}$ represents a TM that starts in state q_R, moves its head one square to the right on tape i regardless of the symbols read on the tapes. It keeps the heads on the other tapes (if any) stationary, and then halts in state h_R.
   - $\mathbf{L}^{(i)}$ is similar except that it starts in state q_L, moves its head one square to the left on tape i, and halts in state h_L.
   - $\mathbf{a}^{(i)}$ represents a TM that starts in state q_a, and writes a on the current square of tape i regardless of the symbols read on the tapes. It keeps the positions of all tape heads unchanged and then halts in state h_a. Here $a \in \Gamma$, where Γ is the tape alphabet of the complex TM we want to build by composing simpler ones.

- $\rightarrow M_1 \overset{a^{(i)}}{\rightarrow} M_2$ represents a composed TM M that starts in state q_{M_1} (starting state of M_1). It then simulates M_1 until either: (a) M_1 enters its halting state h_{M_1} on reading a symbol other than a on tape i, or (b) M_1 enters h_{M_1} on reading a on tape i. In the first case, the composed machine M halts. In the second case, however, M transitions to q_{M_2} (starting state of M_2) instead of going to h_{M_1}. M then simulates M_2 until it halts (if at all).
- The label $a^{(i)}$ in $M_1 \overset{a^{(i)}}{\rightarrow} M_2$ is omitted if the starting state of M_2 (q_{M_2}) is entered whenever M_1 halts regardless of the symbol scanned on tape i.

For example, Fig. 1 shows TMs that scan right until a particular symbol is/is not found. Using the above notation, draw flowcharts for TMs that

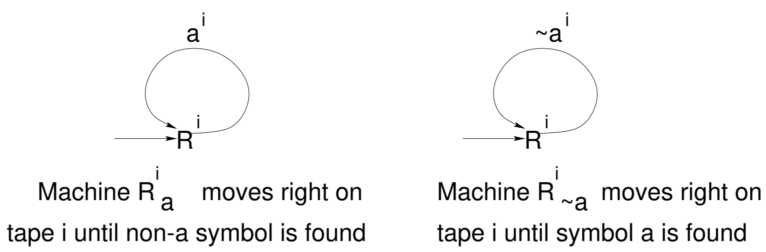

Figure 1:

do the following:

(a) Given a string 01^n0 ($n \geq 1$) on tape 1, the TM halts with the string $01^{Fib(n)}0$ on tape 2, where $Fib(n)$ is the n^{th} Fibonacci number.

(b) Given a string 01^n0 ($n \geq 1$) on tape 1, the TM halts if and only if $\exists p, q \geq 1$ such that $n^2 = p^2 + q^2$.

Clearly specify the number of tapes you use, any symbol other than $0, 1, B$ that you need and any other assumptions. All tapes other than tape 1 may be assumed to be blank to begin with. Similarly, all heads are assumed to be at the left end of the corresponding tapes at the beginning.