CS481F01 HW 7 – TMs

A. Demers

14 Nov – due 21 Nov

Please remember to turn in each problem on a separate page, *put your name on each page*, and turn in the pages in three separate piles!

1. Consider the language

$$L = \{0^i 1 0^{2^i} \mid i > 0\}$$

L is the exponentiation function "represented by pairs."

(a) Prove that L is not a CFL.

(b) Give a formal description of a deterministic total TM recognizing L. Give all the components of the 9-tuple, including a complete specification of the tape alphabet and the transition function. Describe informally how your machine works. A formal proof of correctness is not necessary.

2. Part (a) of this question is from Q. 100 on p. 341 of the text.

A *deterministic 1-counter automaton* (D1CA) is a deterministic automaton with a finite set of states Q, a 2-way read-only input head, and a separate counter that can hold any nonnegative integer. The input $x \in \Sigma^*$ is enclosed in endmarkers $\vdash$ and $\dashv$ that are not in Σ, and the input head may not go outside the endmarkers. The machine starts in its start state s with its counter set to 0 and with its input head on the left endmarker $\dashv$. In each step, it can test its counter for zero. Based on this information, the current state, and the symbol its input head is currently reading, the machine updates its counter value (by adding or subtracting one, or leaving the counter unchanged), moves its head (left, right, or stationary), and enters a new state. The machine accepts by entering a distinguished final state.

(a) Give a rigorous formal definition of these machines, including a definition of acceptance. Your definition should begin as follows:
"A *deterministic 1-counter automaton* is a 7-tuple

$$M \quad = \quad (Q, \Sigma, \vdash, \dashv, s, t, \delta)$$

where ..."

(b) Using your definition from part (a), give a formal description of a D1CA that recognizes the (non-context-free) language

$$L_3 \quad = \quad \{ \ a^i b^i c^i \mid i \geq 0 \ \}$$

Describe informally how your machine works. A formal proof of correctness is not necessary.

(c) Describe informally how a D1CA can recognize the language

$$\{ \ ww^r \mid w \in \Sigma^* \ \}$$

of even-length palindromes. Your description should be informal but complete, roughly at the level of the descriptions in Examples 29.1 and 29.2 on pp. 216-219 of the textbook. In particular, you need not give a list of all the transitions.

3. Consider the set of DFAs with input alphabet $\{0, 1\}$. We can describe such a DFA by a string over the alphabet $\{f, n, 0, 1\}$ as follows. Assume the states are numbered from 1 to N, and by convention the start state is q_1. Each state q is completely described by specifying whether q is final or not, and giving $\delta(q, 0)$ and $\delta(q, 0)$. We use the string $f0^i1^j$ to describe a final state q such that $\delta(q, 0) = q_i$ and $\delta(q, 1) = q_j$. The string $n0^i1^j$ describes a nonfinal state analogously. To describe a DFA, we simply concatenate the descriptions of its states q_1, q_2, ..., in order.

(a) Let

$$R \quad = \quad \{ \ w \in \{0, 1\}^* \mid \sharp 0(w) \text{ and } \sharp 1(w) \text{ are both even } \}$$

Describe a DFA recognizing R and give its encoding as a string as described above.

(b) Now consider the language

$$\begin{aligned} L \;=\; & \{ m\$w \mid m \in \{f, n, 0, 1\}^* \;\wedge\; w \in \{0,1\}^* \\ & \wedge\, m \text{ is the description of a DFA } M \\ & \wedge\, w \;\in\; L(M) \;\} \end{aligned}$$

Describe a deterministic TM that recognizes L. Your description should be informal but complete, as discussed in question 2(c) above. Describe the action of your TM on your DFA description from part (a) on input 0110.

Hint: Your TM should simulate the DFA described by m computing on input w. We have argued that multiple tapes do not increase the power of the TM model; so if you wish you may describe a multi-tape machine to recognize L. This should make your life somewhat easier.

Danger Will Robinson: Suppose the input string is the description of a DFA M followed by an input that is not in $L(M)$. Your TM should reject in this case.

Suppose the input string is *not* a DFA description followed by an input. Your TM had better reject in this case as well, right? So it should perform some sort of "sanity check" on its input.

Suppose there is a state description like $n0^2 1^{17}$ in (the description of) a DFA with only 16 states. You may treat this as a "run-time error" – you don't need to detect this situation unless sometime during the simulated computation you actually try to change to the nonexistent state 17.

This is a judgement call in the interpretation of

"m is the description of a DFA M"

above. I choose to interpret an out-of-range state transition as an explicit reject action, and still consider m a well-formed DFA description. Using this interpretation, the set of well-formed DFA descriptions is regular.