CS481F01 Solutions 7

A. Demers

7 Dec 2001

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.

Answer (a) By Parikh's Theorem, if L were context-free then it would be letter-equivalent to a regular set. Applying a homomorphism that erases 1, we conclude

$$L \text{ a CFL} \quad \Rightarrow \quad \{\ 0^{(i+2^i)} \mid i > 0\ \} \text{ is ultimately periodic}$$

Consider the function

$$F(L) = \lim_{n\to\infty} \frac{|\{w \in L|\ len(w) \leq n\}|}{n}$$

It is easy to see that, for any infinite ultimately periodic set, this limit exists and is at least $1/p$, where p is the least period for the set. It is also easy to see that

$$F(\{\ 0^{(i+2^i)} \mid i > 0\ \}) = 0$$

Thus, the set is not ultimately periodic, and L is not context-free.

(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.

Answer (b) A machine is

$$\begin{aligned} M &= (Q, \{0,1\}, \{\vdash, 0, 1, \bullet, \sqcup\}, \vdash, \sqcup, \delta, s, t, r\} \\ Q &= \{s, q_1, q_2, q_3, q_4, q_5, q_6, t, r\} \end{aligned}$$

The state transitions (explained below) are given by the following table.

	$\vdash$	0	1	$\bullet$	$\sqcup$
s	$q_1, \vdash, R$	$r, 0, R$	$r, 1, R$	$r, \bullet, R$	$r, \bullet, R$
q_1	ϕ	$q_2, \bullet, R$	$q_6, 1, R$	$q_1, \bullet, R$	$r, \sqcup, R$
q_2	ϕ	$q_2, 0, R$	$q_3, 1, R$	$q_2, \bullet, R$	$r, \sqcup, R$
q_3	ϕ	$q_4, \bullet, R$	$r, 1, R$	$q_3, \bullet, R$	$r, \sqcup, R$
q_4	ϕ	$q_3, 0, R$	$r, 1, R$	$q_4, \bullet, R$	$q_5, \sqcup, L$
q_5	$q_1, \vdash, R$	$q_5, 0, L$	$q_5, 1, L$	$q_5, \bullet, L$	ϕ
q_6	ϕ	$r, 0, R$	$q_6, 1, R$	$q_6, \bullet, R$	$t, \sqcup, R$
t	ϕ	ϕ	ϕ	ϕ	$t, \sqcup, R$
r	ϕ	$r, 0, R$	$r, 1, R$	$r, \bullet, R$	$r, \sqcup, R$

The operation is roughly as follows:

State s just moves off the left endmarker and transitions to state q_1.

State q_1 finds a 0 to the left of the 1 and changes it to $\bullet$.

State q_2 scans forward to the 1.

States q_3 and q_4 change every second 0 to $\bullet$.

State q_5 returns to the left end of the tape.

State q_6 checks that all $0's$ have been consumed (converted to $\bullet$).

States t and r are the usual accept and reject states.

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.

Answer (a) A *deterministic 1-counter automaton* is a 7-tuple

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

where

Q is a finite set of states
Σ is a finite input alphabet
$\vdash \in \Sigma$ is the left endmarker symbol
$\dashv \in \Sigma$ is the right endmarker symbol
$s \in Q$ is the start state
$t \in Q$ is the accept state
$\delta \; : \; (Q \times \Gamma \times \{+, 0\}) \; \rightarrow \; (Q \times \{L, \bullet, R\} \times \{1, 0, -1\})$
is the transition function

The first argument of δ is the current machine state; the second argument is the symbol under the input head; and the third argument represents the value of the counter – $+$ means the counter is positive, 0 means it is zero (recall the counter must be nonnegative). The result of applying δ yields a new state, a direction in which to move the head ($\bullet$ means the head remains stationary), and a number to add to the counter value.

The transition function must satisfy two restrictions. First, it may not move the head outside the endmarkers:

$$\begin{aligned} \delta(p, \vdash, c) \; &\neq \; (q, L, i) \\ \delta(p, \dashv, c) \; &\neq \; (q, R, i) \end{aligned}$$

Second, it may not decrement the counter below zero:

$$\delta(p, a, 0) \; \neq \; (q, d, -1)$$

A *configuration* is a 4-tuple $\langle w, q, i, c \rangle$ where w is the tape content, q is the state, i is the head position, and c is the counter value.

The *one step transition relation* is defined by

$$\begin{array}{lclll}
\langle w, p, i, 0 \rangle & \rightarrow & \langle w, q, i-1, d \rangle & \text{if} & \delta(p, w[i], 0) = \langle q, L, d \rangle \\
\langle w, p, i, 0 \rangle & \rightarrow & \langle w, q, i, d \rangle & \text{if} & \delta(p, w[i], 0) = \langle q, \bullet, d \rangle \\
\langle w, p, i, 0 \rangle & \rightarrow & \langle w, q, i+1, d \rangle & \text{if} & \delta(p, w[i], 0) = \langle q, R, d \rangle \\
\langle w, p, i, c \rangle & \rightarrow & \langle w, q, i-1, c+d \rangle & \text{if} & \delta(p, w[i], +) = \langle q, L, d \rangle \\
\langle w, p, i, c \rangle & \rightarrow & \langle w, q, i, c+d \rangle & \text{if} & \delta(p, w[i], +) = \langle q, \bullet, d \rangle \\
\langle w, p, i, c \rangle & \rightarrow & \langle w, q, i+1, c+d \rangle & \text{if} & \delta(p, w[i], +) = \langle q, R, d \rangle
\end{array}$$

Machine M *accepts* x if

$$\langle \vdash x \dashv, s, 0, 0\rangle \;\; \rightarrow^* \;\; \langle \vdash x \dashv, t, i, j\rangle$$

for some i, j.

(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.

Answer (b) The machine has 7 states, as follows:

	$\vdash 0$	$\vdash 1$	$a0$	$a1$	$b0$	$b1$	$c0$	$c1$	$\dashv 0$	$\dashv +$
s	q_1R0	ϕ	ϕ	ϕ	ϕ	ϕ	ϕ	ϕ	ϕ	ϕ
q_1	ϕ	ϕ	q_1R1	q_1R1	$r \bullet 0$	q_2R-1	$r \bullet 0$	$r \bullet 0$	$r \bullet 0$	$r \bullet 0$
q_2	ϕ	ϕ	$r \bullet 0$	$r \bullet 0$	$r \bullet 0$	q_2R-1	q_3R1	$r \bullet 0$	$r \bullet 0$	$r \bullet 0$
q_3	ϕ	ϕ	$r \bullet 0$	$r \bullet 0$	$r \bullet 0$	$r \bullet 0$	ϕ	q_3R1	$r \bullet 0$	q_4L0
q_4	ϕ	ϕ	$r \bullet 0$	$t \bullet 0$	$r \bullet 0$	q_4L-1	ϕ	q_4L0	ϕ	ϕ
t	$t \bullet 0$	$t \bullet 0$	$t \bullet 0$	$t \bullet 0$	$t \bullet 0$	$t \bullet 0$	$t \bullet 0$	$t \bullet 0$	$t \bullet 0$	$t \bullet 0$
r	$r \bullet 0$	$r \bullet 0$	$r \bullet 0$	$r \bullet 0$	$r \bullet 0$	$r \bullet 0$	$r \bullet 0$	$r \bullet 0$	$r \bullet 0$	$r \bullet 0$

In this machine:

The start state s just moves right and goes to q_1.
State q_1 moves to the right counting $a's$.
State q_2 moves right matching $b's$ against the counter value.
State q_3 moves right counting $c's$.
State q_4 moves left, ignoring $c's$ and matching $b's$.
The accept and reject states (t and r) behave in the usual way.

Note the behavior of q_4 ignoring $c's$ is okay, since the left-to-right scan has already verified that the input is in $a^*b^*c^*$.

(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.

Answer (c) This is not really too difficult. Suppose the head is at some position on the tape, and the counter is 0. To find the "matching" symbol you can

- move to the left end of the tape, counting the required steps,
- move to the right end of the tape without changing the counter,
- move left as many steps as the counter contains.

That enables the machine to compare two symbols at corresponding locations. if they do not match, the machine rejects.

To recognize even-length palindromes, a 1-counter machine first makes a left-to-right pass across the tape, counting the symbols and checking that the total length is even. It then moves left halfway across the tape (decrementing the counter by 2 at each step). This leaves the head on the rightmost symbol of the left half of the tape. The machine then performs the "checking" procedure described above, leaving the head on the leftmost symbol of the right tape half. It then advances the head one step to the right (to the next-to-leftmost symbol of the right tape half) and performs the "checking" procedure again. (For efficiency, it could interchange "left" and "right" in the description, but this is not necessary). This leaves the tape head on the next-to-rightmost symbol of the left tape half. The machine then advances one step to the left, and repeats the "checking" procedure. This continues until the endmarker is encountered in one of the "advance" steps, at which point the machine accepts.

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 $\delta(q, 1)$. 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, $\ldots$, 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.

Answer (a) Gosh – this would have been slightly easier if the states had been numbered from 0 to $n-1$ rather than from 1 to n in the problem description. But so be it. A succinct description of such a machine: the states are indexed by numbers of the form $1+ab$ where ab is a 2-bit number; after reading w the machine is in state $1+ab$ if

$$a \;=\; (\sharp 0(w) \bmod 2) \qquad \text{and} \qquad b \;=\; (\sharp 1(w) \bmod 2)$$

The only accepting state is q_1, that is, q_{1+00}. The δ function is

$$\begin{array}{ll}
\delta(q_1,0)=q_2 & \delta(q_1,1)=q_3 \\
\delta(q_2,0)=q_1 & \delta(q_2,1)=q_4 \\
\delta(q_3,0)=q_4 & \delta(q_3,1)=q_1 \\
\delta(q_4,0)=q_3 & \delta(q_4,1)=q_2
\end{array}$$

The string encoding this machine is:

$$f00111n01111n00001n00011$$

(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.

Answer (b) As suggested in the hint, we describe a multi-tape machine to recognize L.

The machine has two tapes in addition to its input tape, One of these, which we'll call the *description tape*, will contain the description of the DFA being simulated. So the machine's first sequence of moves will be to copy the initial portion of its input (up to the first \$ symbol) onto the description tape. At the same time it will check (using states of its finite control) that the string being copied is in fact a well-formed DFA description terminated by \$; otherwise the machine rejects.

The machine then moves the description tape head to the left end, and moves the input head to the position just to the right of the \$ (which terminated the machine description in the input).

While simulating the DFA, the machine maintains the invariant that at the beginning of a simulated step the input tape head is at the input symbol about to be read by the simulated DFA, the description tape head is at the first symbol (n or f) of the description of the state of the simulated DFA, and the work tape head (remember the work tape?) is at the left end. It simulates a move by

- Examine the next input symbol, in the process moving the input tape head one cell to the right
- If the input symbol is blank, then accept (if the symbol under the description tape head is f) or reject (if the symbol under the description tape head is n).
- If the input symbol is not 0 or 1, then reject.
- Copy the destination state number onto the work tape.
- Move the description tape head to the left end, then move it to the beginning of the destination state by repeatedly decrementing the state number on the work tape while moving the description tape head one state to the right. If the description tape head runs off the right end of the description (i.e. encounters a blank symbol) then reject.

Note this reestablishes the invariant as required: the input tape head is at the correct position, the description tape head is at the beginning of the current state description, and the work tape head is at the left end.