MATH 4LT/6LT3 Assignment #3 Solutions
Due: Tuesday, 21 October by 11:59pm.

1. Let $T = (\{q_0, q_1, q_2, q_{accept}, q_{reject}\}, \{a\}, \{a, \sqcup\}, q_0, q_{accept}, q_{reject}, \delta)$ be the DTM whose transition function is given by the following table:

$Q \times \Gamma$	δ
$(q_0, \sqcup)$	$(q_{accept}, \sqcup, +1)$
(q_0, a)	$(q_1, a, +1)$
$(q_1, \sqcup)$	$(q_2, \sqcup, +1)$
(q_1, a)	$(q_0, \sqcup, +1)$
$(q_2, \sqcup)$	$(q_2, \sqcup, +1)$
(q_2, a)	$(q_2, \sqcup, +1)$

(a) Show that T accepts the string $aaaa$ by writing down the sequence of configurations that T occupies, after being started in the initial configuration $[q_0, aaaa, 1]$.

Solution:

$$\begin{aligned}[q_0, aaaa\sqcup\ldots, 1] \quad &\vdash_T [q_1, aaaa\sqcup\ldots, 2] \vdash_T [q_0, a\sqcup aa\sqcup\ldots, 3]\\ &\vdash_T [q_1, a\sqcup aa\sqcup\ldots, 4] \vdash_T [q_0, a\sqcup a\sqcup\sqcup\ldots, 5]\\ &\vdash_T [q_{accept}, a\sqcup a\sqcup\sqcup\sqcup\ldots, 6]\end{aligned}$$

(b) Describe the sequence of configurations that T occupies after being started in the configuration $[q_0, aaa, 1]$. Does T accept aaa?
Solution:

$$\begin{aligned}[q_0, aaa\sqcup\ldots, 1] \quad &\vdash_T [q_1, aaa\sqcup\ldots, 2] \vdash_T [q_0, a\sqcup a\sqcup\ldots, 3]\\ &\vdash_T [q_1, a\sqcup a\sqcup\ldots, 4] \vdash_T [q_2, a\sqcup a\sqcup\sqcup\ldots, 5] \vdash_T \ldots\end{aligned}$$

T does not accept aaa, since after four steps, it reaches state q_2 and then continues to move to the right, in that state, forever.

(c) For $n \geq 0$, carefully describe what T does when started in the configuration $[q_0, a^n, 1]$.

Solution: T will move to the right, erasing every other a, starting with the second one it finds. If n is even, then once T has reached the end of the input string, it will halt in state q_{accept}, in the configuration $[q_{accept}, (a\sqcup)^{n/2} \ldots, n+2]$. If n is odd, then T will erase every other a, starting with the second one, and once it has reached the end of the input string, will continue to move to the right, in state q_2, forever.

(d) Describe the language accepted by T.

Solution: $L(T) = \{aa\}^*$, a regular language. Note that even though this language is computable, the DTM T is not halting and so does not decide it.

2. Consider the language L over the alphabet $\{0, 1\}$ consisting of all words w that contain an equal number of 0's and 1's.

(a) Describe a halting DTM that accepts L. Your description should describe how your DTM operates on an input word w without going into too much detail.

Solution: When started in the configuration $[s, w \sqcup \ldots, 1]$ (where s is its start state), the TM will move to the right until it finds a 0 or a 1. When it does so, it replaces the symbol with an x. It then continues to move to the right until it finds a 1, if it just replaced a 0 with an x, or a 0, otherwise. It then replaces that symbol with an x. It then moves to the left end of the tape and repeats the above procedure.

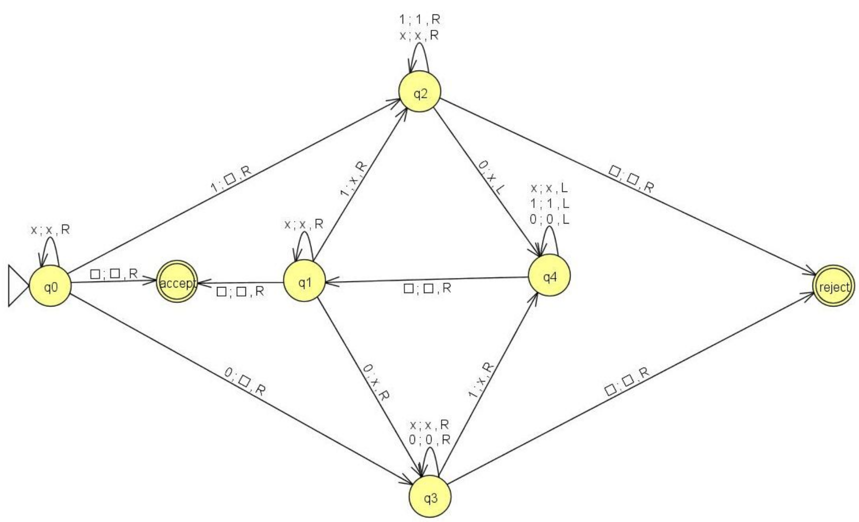
If when searching for a new 0 or 1 at the start of the above procedure, a blank symbol is encountered, then the TM will halt in the accept state. If, when searching for a matching 1 or 0, and a blank is encountered, then the TM will halt in the reject state.

Note that in order to keep track of the end of the tape, the TM will write the symbol $\sqcup$ when replacing the first 0 or 1 on the tape. We could use a "marked" version of x, such as $x_\vdash$ instead.

(b) Draw a state diagram for the DTM described in part a). To save time and effort you need not include any transitions that do not matter for the operation of your DTM. You may use the JFLAP software or some other Turing Machine simulator for this part of the problem.

Solution: The following diagram was created using JFLAP and the transition notation is slightly different from the one used in the textbook. The □ symbol is used in place of the ⊔ symbol to denote the blank symbol, a ; is used in place of /, and L and R are used in place of -1 and $+1$.

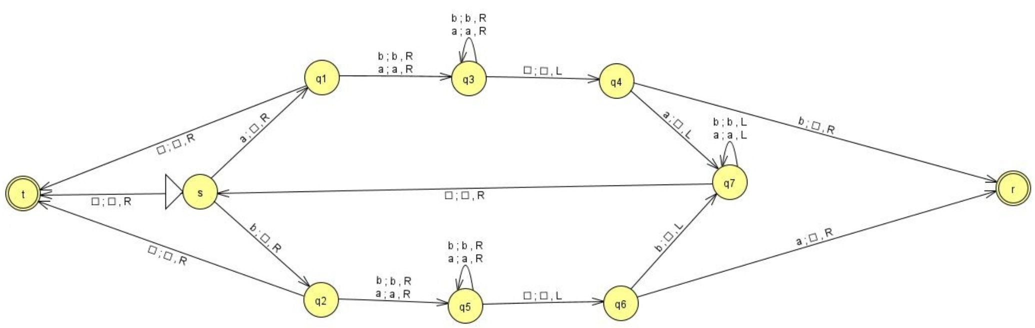
In this DTM, $q0$ is the start state, the input alphabet is $\{0, 1\}$, and the tape alphabet is $\{0, 1, x, \sqcup\}$.Note that the transition from state $q3$ to state $q4$ is incorrectly labelled. The correct label is: 1;x,L.



3. Exercise 2.10.5.

   Solution: The following is a diagram for a DTM whose language is the set of palindromes over the alphabet $\{a, b\}$. The start state is s, and the accepting and rejecting states are t and r respectively.

   Here is a high-level description of what it does: On input the string x, if the first symbol is an a (b), it will delete it and move to the right end of the word. If a b (a) is found, x will be rejected; if an a (b) is found, it will be erased, and the DTM will move left to the first non-blank symbol on the tape. If none are found, x will be accepted. Otherwise, the above procedure will be repeated on the remaining string. Note that if x has length one, then it will be accepted.



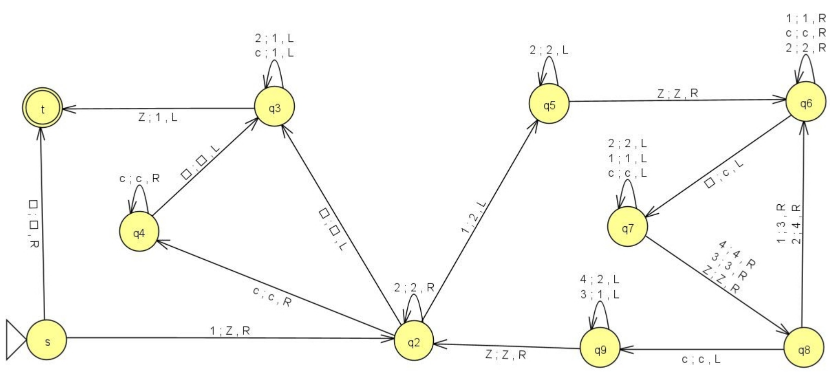
4. Exercise 2.10.9, part 2.

   Solution: The following diagram is of a DTM M that computes the function $f(1^n) = 1^{n^2}$. Here is a high level description of how M operates on the input string 1^n:

   - First it marks the left end of the tape with the symbol Z (unless the tape is empty). [see state s.]
   - For each remaining 1 in the input string, M replaces it with the symbol 2 and then moves to the end of the string and writes out the symbol c n-times. To keep track of this copying phase of the procedure, M uses additional symbols 3 and 4 to keep track of which symbols of the input string have been copied. When a 1 is copied, it is temporarily changed to a 3 and when a 2 is copied, it is temporarily changed to a 4. [see states q6, q7, and q8.]
   - Once the n c's have been written, the symbols 3 and 4 on the tape are changed back to 1's and 2's. [see states q8 and q9.]
   - When all of the 1's from the input string have been changed to 2's (or Z), then the number of non-blank symbols on the tape is n^2 (since for each 1 in the input string, except for the first one, n c's have been written on the tape). Finally, all non-blank symbols

are changed to 1's and the DTM halts in the accept state. [see states q3 and q4.]

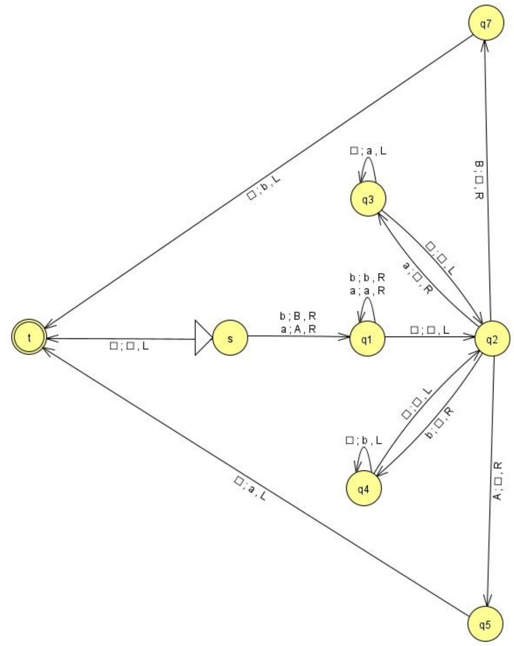
Note that this DTM diagram only includes transition that actually get used during a computation. For example, there is no transition from state $q4$ when reading the symbol 3, since this situation will never occur while M is running on input 1^n. Also, there is no reject state, but one could be added.



5. Exercise 2.10.10.

Solution: The following diagram is of a DTM that shifts its input string one cell to the right. It assumes that the input alphabet is $\{a, b\}$, but can be easily modified to handle other alphabets. If the input is empty, the DTM halts, otherwise it capitalizes the first symbol and then moves to the end of the input string. Next, it deletes the symbol in the cell and copies it one cell to the right. It then moves to the left two cells and repeats this step. This continues until the left most symbol (now capitalized) is encountered. This is deleted and copied into the cell to the right. The DTM moves to the left and halts.

For larger alphabets, some elements of the given diagram just need to be reproduced, using the other symbols in the alphabets as labels for the transitions, in place of a and b.



The following questions are for students enrolled in MATH 6LT3. Students in MATH 4LT3 can treat them as bonus questions.

B1 A common definition of a DTM allows for the machine to not move after reading a symbol, instead of always having to move one cell to the left or to the right. Consider a variant DTM whose transition function may also produce a triple $\delta(q, s) = (q', a', 0)$, where the 0 indicates that the read/write head of the DTM does not move.

Describe how a standard DTM can be used to simulate the running of one of these variant machines. More precisely, given a specification $T = (Q, \Sigma, \Gamma, q_s, q_{accept}, q_{reject}, \delta)$ of a variant DTM, give a definition of a standard DTM T' which essentially does the same things that T does.

Solution: To simulate a step in which the read/write head does not move, we can introduce a new, intermediate state for each such transition, and simulate the standing still transition by having the TM transition to the new state and move to the right one cell. The TM will next immediately move left one cell and transition to the state dictated by the original transition.

In more detail, if $\delta(q, s) = (q', a', 0)$ is a transition of T, we introduce a new state q^*, remove this transition, and add the following transitions:

- $\delta(q, s) = (q^*, a', +1)$ and
- for each symbol $r \in \Gamma$, we add the transition $\delta(q^*, r) = (q', r, -1)$.

The resulting TM will not have any stationary transitions and will be equivalent to the original TM.

B2 Exercise 2.10.14, part 1.

Solution: Let $M = (Q, \Sigma, \Gamma, s, t, r, \delta)$ be a right-moving DTM. Define $M' = (Q', \Sigma, s', T', \delta')$ to be the DFA with:

- $Q' = Q$,
- $s' = s$,
- $T' = \{t\} \cup \{q \in Q' \mid [q, \sqcup \ldots, 1] \vdash_M^* [t, x \sqcup \ldots, l]$ for some string x and $l \geq 1\}$, and
- for $q \in Q'$, with $q \neq t$, $a \in \Sigma$, $\delta'(q, a) = q'$, where $\delta(q, a) = (q', a', +1)$ for some $a' \in \Sigma$. For $a \in \Sigma$, $\delta'(t, a) = t$ and $\delta'(r, a) = r$.

We claim that a string $x \in \Sigma^*$ is accepted by M if and only if it is accepted by M'. This will establish that $L(M) = L(M')$ and hence that $L(M)$ is regular.

Let x be a string of length n. We claim that for $i \leq n$, M will be in state q, after i steps in its computation if and only if M' is as well. This is clear when $i = 0$. Suppose that it is true for $0 \leq i < n$ and consider the next step in the computations. Using the definition of δ', it can be seen that both devices enter the same next state. The only exceptional states are t and r, but in that case, once either device enters either of those states, it remains in that state afterwards.

After n steps in the computation of M on input x, the head will be in cell $n+1$, reading a blank symbol and in some state q, or at some prior step, M will have entered one of the states t or r. In the case that it enters t at some earlier step, then M accepts x, and by construction,

M' will as well, since $t \in T'$. On the other hand, if the computation of M takes at least n steps, then M will only accept x, if the state t is eventually reached, when starting in state q, reading a blank. But the collection of such states is defined to be T', the set of accept states of M', and so we see that in this case, x is accepted by M if and only if it is accepted by M'.