CS 722 Fall 2016
Homework Assignment #1
Solutions

In all questions "_" is the blank symbol.

1. Consider the DTM to decide { w#w | w ∈ {0, 1}* } in Example 3.9 in the book.
   a. Give the transition sequence on each of the following inputs:
      i. 00#00

         $q_1$00#00 |-
         x$q_2$0#00 |-
         x0q_2#00 |-
         x0#$q_4$00 |-
         x0q_6#x0 |-
         x$q_7$0#x0 |-
         q_7x0#x0 |-
         x$q_1$0#x0 |-
         xxq_2#x0 |-
         xx#q_4x0 |-
         xx#x$q_4$0 |-
         xx#$q_6$xx |-
         xxq_6#xx |-
         xq_7x#xx |-
         xxq_1#xx |-
         xx#$q_8$xx |-
         xx#xq_8x |-
         xx#xxq_8_ |-
         xx#xx_q_{accept}_

      ii. 01#00

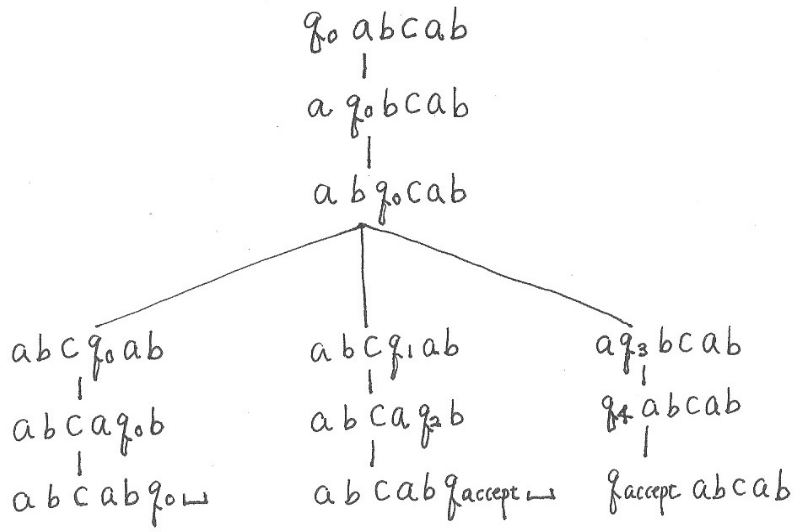
         $q_1$01#00 |-
         x$q_2$1#00 |-
         x1q_2#00 |-
         x1#$q_4$00 |-
         x1q_6#x0 |-
         x$q_7$1#x0 |-
         q_7x1#x0 |-
         x$q_1$1#x0 |-
         xxq_3#x0 |-
         xx#q_5x0 |-
         xx#x$q_5$0 |-
         xx#x0q_{reject}_

   b. Analyze the worst-case time complexity function of this DTM and express it in O(g(n)) asymptotic notation.

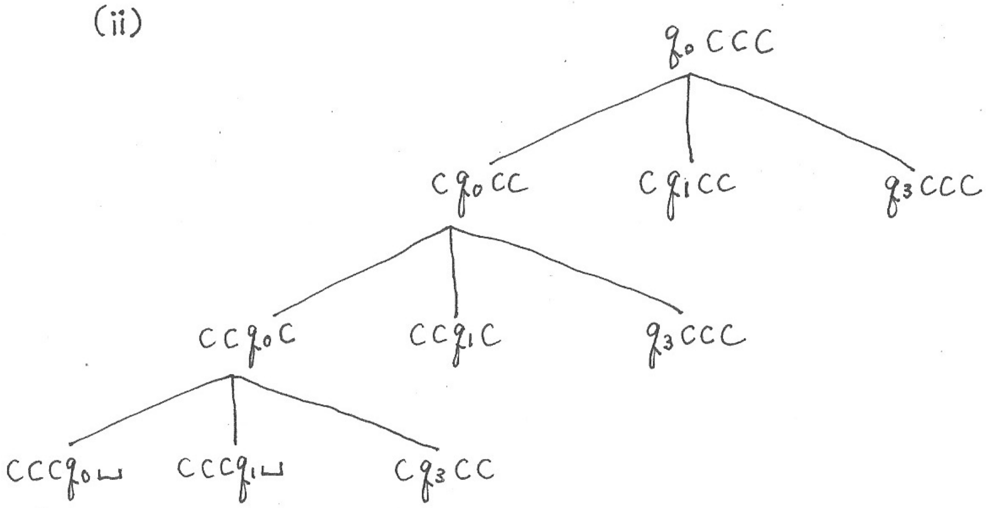
      The worst case occurs when the input is of the form w#wu, w ∈ {0, 1}*, u ∈ Σ*. If the input does not take this form, the DTM halts in q_{reject} as soon as a mismatch of two symbols in the substrings before and after # is detected. Each scan to cross off two matching 0's or 1's ($q_1 \to q_2 \to q_4 \to q_6$, or $q_1 \to q_3 \to q_5 \to q_6$) takes O(n/2) steps. Moving the tape head back ($q_6 \to q_7 \to q_1$) similarly takes O(n/2) steps. Since each scan crosses off 2 matching symbols, the total number of iterations is O(n/2). The last stage to check if u is empty ($q_1 \to q_8 \to q_{accept}$) takes O(n/2) steps. In total, (O(n/2)+O(n/2))×O(n/2) + O(n/2) = O(n^2).

2. Consider the NTM in [Question 2 here](). Give the computation tree of this NTM on each of the following inputs:
   i. abcab
   ii. ccc

(i)



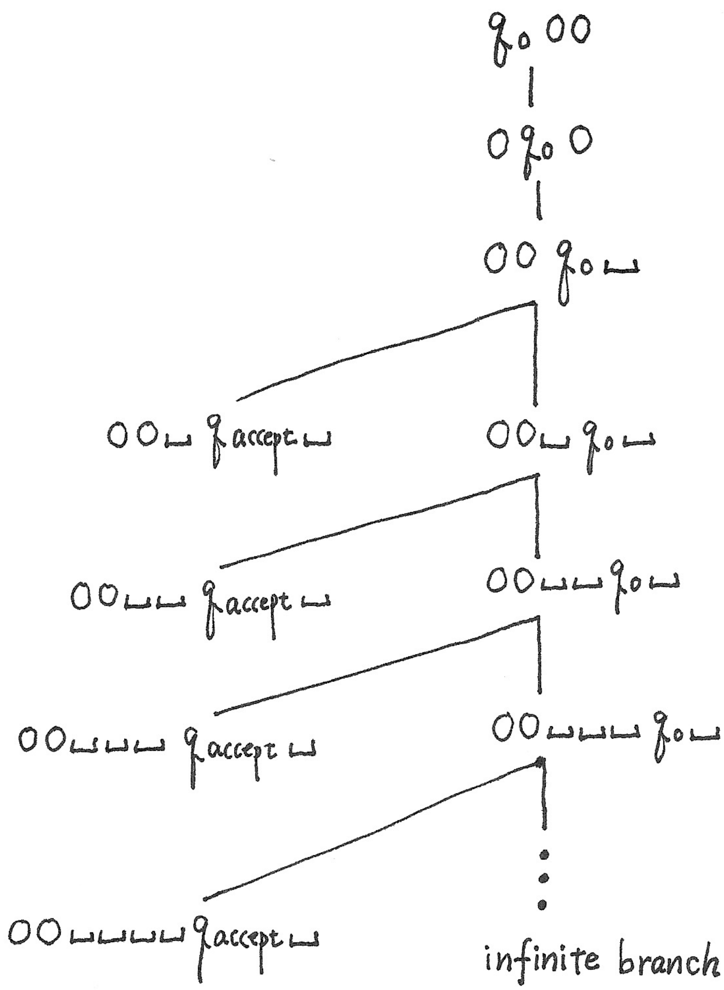
(ii)



3. Consider the NTM with Q = {q_0, q_{accept}}, Σ = {0}, Γ = {0, _},

   δ(q_0, 0) = {(q_0, 0, R)}
   δ(q_0, _) = {(q_0, _, R), (q_{accept}, _, R)}

a. Give the computation tree on the input 00. How many accepting branches are in the computation tree?



There are infinitely many accepting branches leading to the accepting configurations of the form $00\_^k q_{accept\_}$, $k \geq 1$.

b. Specify the set of all input strings accepted by this NTM.

The NTM accepts $\{ 0^n \mid n \geq 0 \}$. The computation tree on 0^n has a pattern similar to the above tree, containing one infinite branch and infinitely many accepting branches leading to $0^n\_^k q_{accept\_}$, $k \geq 1$.

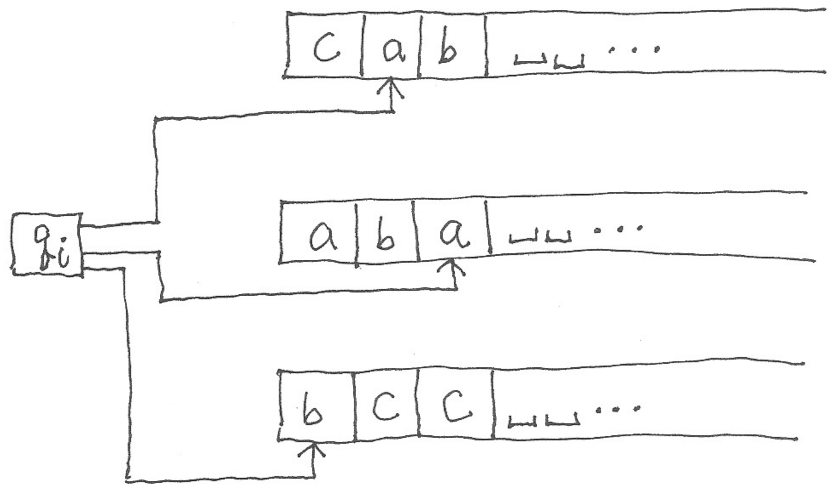
c. Specify the set of all input strings rejected by this NTM.

The empty set.

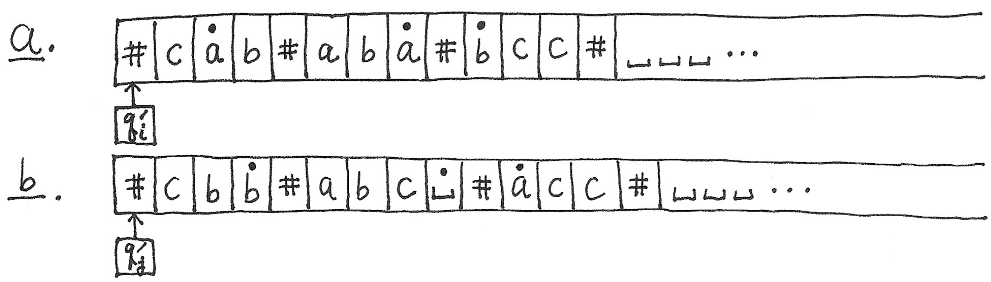
d. Is this NTM a decider? Why or why not?

It is not a decider because the computation tree on 0^n, $n \geq 0$, has an infinite branch.

4. This question is about the simulation of multi-tape DTMs by 1-tape DTMs. Consider the following configuration of a 3-tape DTM with $\Gamma = \{a, b, c, \_\}$:

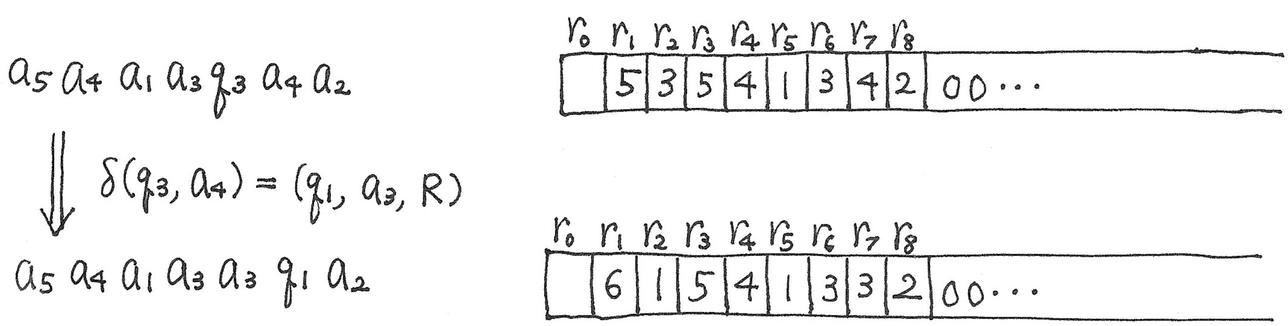


a. Give the 1-tape DTM configuration simulating the above configuration.
b. Give the 1-tape DTM configuration simulating the configuration obtained by the 3-tape DTM transition $\delta(q_i, a, a, b) = (q_j, b, c, a, R, R, L)$ from the above configuration.



5. Recall the simulation of DTMs by RAMs described in class.

   a. Give the RAM memory configuration simulating $a_5a_4a_1a_3q_3a_4a_2$.
   b. Give the RAM memory configuration simulating the configuration obtained by the transition $\delta(q_3, a_4) = (q_1, a_3, R)$ from the above configuration.



6.

   a. Give a RAM program segment to compute the value of $(r_1-r_2)+(r_3*r_4)$. The result will be placed in the accumulator r_0.

   The following is just an example program.

```
Load 3   // r0 ← r3
Mult 4   // r0 ← r0 * r4
```

```
Store 5 // r5 ← r0
Load 1  // r0 ← r1
Sub 2   // r0 ← r0 - r2
Add 5   // r0 ← r0 + r5
```

b. Recall the simulation of RAMs by the 5-tape DTMs described in class. Presuming that, initially, $r_1 = 4$, $r_2 = 3$, $r_3 = 1$, $r_4 = 2$, show the contents of tape 1 (simulating the RAM memory) and tape 2 (simulating the accumulator r_0) after the execution of each instruction in your RAM program. Use a format similar to Question 5 on this page, and abbreviate integers in unary notation to decimal notation.

initial tape configuration

tape 1: ##1#4##2#3##3#1##4#2##
tape 2: has some value

Load 3

tape 1: ##1#4##2#3##3#1##4#2##
tape 2: 1

Mult 4

tape 1: ##1#4##2#3##3#1##4#2##
tape 2: 2

Store 5

tape 1: ##1#4##2#3##3#1##4#2##5#2##
tape 2: 2

Load 1

tape 1: ##1#4##2#3##3#1##4#2##5#2##
tape 2: 4

Sub 2

tape 1: ##1#4##2#3##3#1##4#2##5#2##
tape 2: 1

Add 5

tape 1: ##1#4##2#3##3#1##4#2##5#2##
tape 2: 3

7. Informally but in sufficient detail, describe how to simulate an arbitrary 1-tape DTM by a 1-tape DTM with $\Sigma = \{0, 1\}$ and $\Gamma = \{0, 1, \_\}$.

The following is an example simulation method.

Let M be an arbitrary 1-tape DTM with $\Gamma = \{a_0, \ldots, a_{n-1}\}$ where a_{n-1} is "_". Encode each of these n tape symbols by a p-bit string where $p = \lceil \log_2 n \rceil$. Let $\alpha(a_i)$ be the p-bit string encoding a_i.

The simulating 1-tape DTM, M', encodes each symbol *a* on the tape of M by $\alpha(a)$, except for infinitely many "_"s that have not been accessed by the tape head. For example, suppose n = 8. Let $\alpha(a_i)$ be the 3-bit binary code of i. Then the following string on the tape of M:

a_2 _ a_6 _ a_3 _ _ _ _ …

would be encoded by the following bit string on the tape of M':

010111110111011 _ _ _ _ …

This way, the tape string of M' is segmented into p-bit parts (p-bit "words"); no separator symbol is necessary since p is a constant determined by Γ of M. In simulation of one step-transition $\delta(q_i, a) = (q_j, b, L/R)$ of M, M' reads the p-bit code $\alpha(a)$, move the tape head back to the start of the p-bit segment, replaces $\alpha(a)$ by $\alpha(b)$, and move the tape head to the start of the preceding or succeeding p-bit segment according as M moves the tape head to L or R. This process can be implemented by introducing additional control states to M'. The exception is if M' reads the leftmost "_" not accessed by the tape head so far. In this case M' replaces the "_" by $\alpha(b)$ where $\delta(q_i, \_) = (q_j, b, L/R)$.

Another simulation method is to encode the tape symbols of M in unary notation 0^k separated by 1's.