# CS375 Homework Assignment 8 Solution Set (40 points)
Due date: 4/9/2024

1. (6 points)

   Fill out the following blanks for the instructions of a Turing machine that moves an input string over {a, b} to the right one cell position. The read/write head of the control unit initially is at the left end of the input string. The rest of the tape cells are blank. The TM will move the entire string to the right one cell position and leave all remaining tape cells blank. The read/write head ends at the right end of the output string.

   | ( 0 , a, Λ, R, | 1 | found an a |
   |---|---|---|
   | ( 0 , Λ, R, | 2 | found a b |
   | ( 0 , Λ, S, halt) | | Done |

   | ( 1 , a, a, R, | 1 | found an a & to write an a |
   |---|---|---|
   | ( 1 , a, R, | 2 | found a b & to write an a |
   | ( 1 , a, S, halt) | | Done |

   | ( 2 , a, b, R, | 1 | found an a & to write a b |
   |---|---|---|
   | ( 2 , b, R, | 2 | found a b & to write a b |
   | ( 2 , b, S, halt) | | Done |

2. (7 points)

   The following is the instruction set of a TM that moves an input string over {a, b} to the right two cells position.

(0, a, Λ, R, 1)      found an a
(0, b, Λ, R, 2)      found a b
(0, Λ, Λ, S, halt)   Done

(1, a, Λ, R, 11)     found aa
(1, b, Λ, R, 12)     found ab
(1, Λ, Λ, R, 10)     found aΛ

(2, a, Λ, R, 21)     found ba
(2, b, Λ, R, 22)     found bb
(2, Λ, Λ, R, 20)     found bΛ

(11, a, a, R, 11)    write an a, aa to go
(11, b, a, R, 12)    write an a, ab to go
(11, Λ, a, R, 10)    write an a, aΛ to go

(12, a, a, R, 21)    write an a, ba to go
(12, b, a, R, 22)    write an a, bb to to
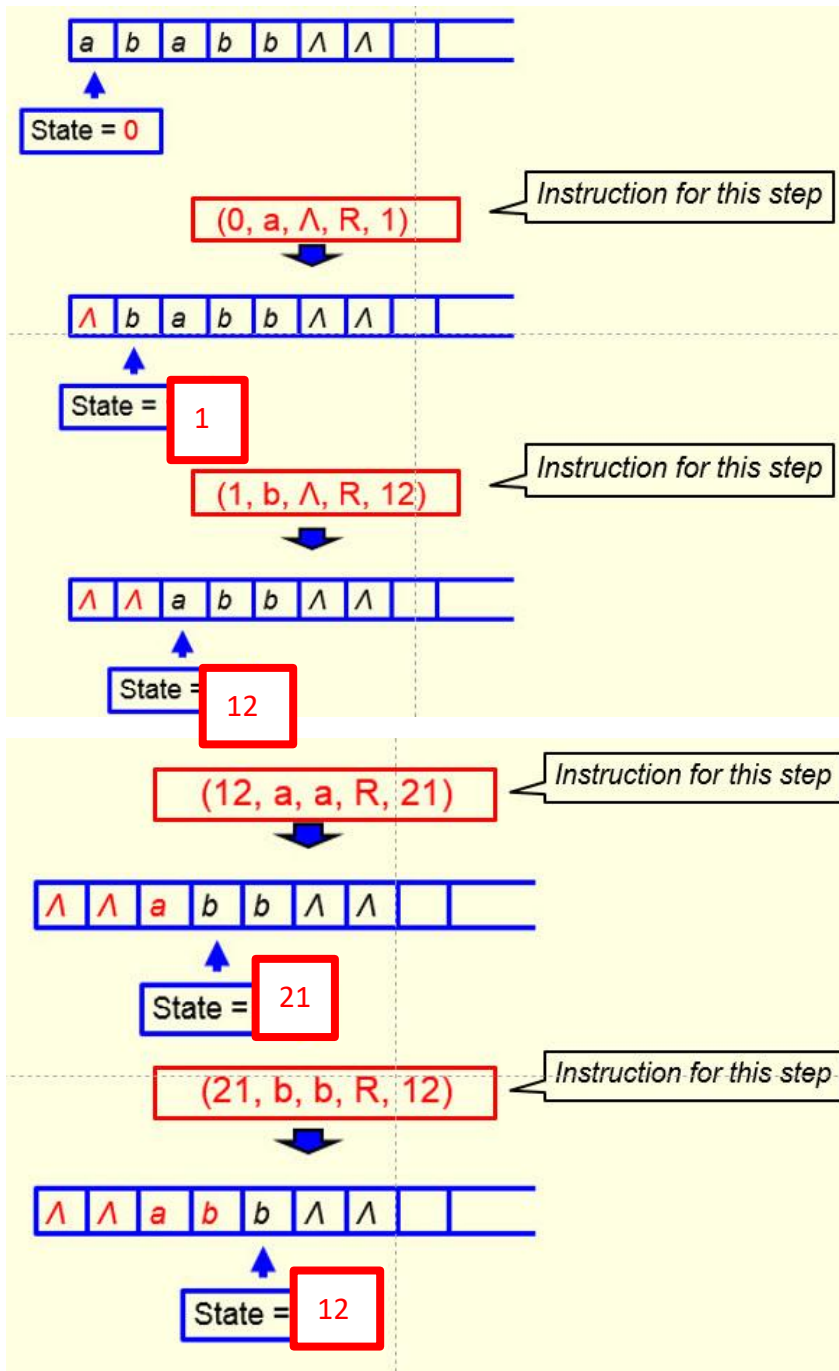(12, Λ, a, R, 20)    write an a, bΛ to go

(10, Λ, a, S, halt)   write an a, then halt
(10, a, a, S, halt)   write an a, then halt
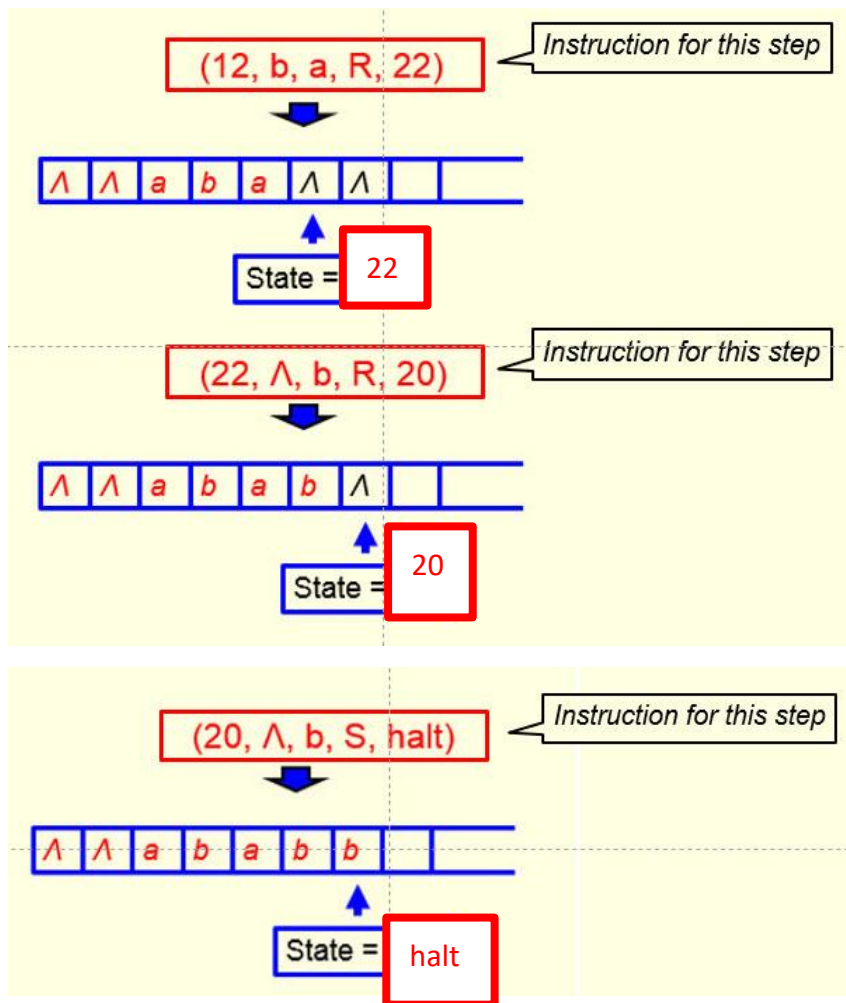(10, b, a, S, halt)   write an a, then halt

(21, a, b, R, 11)    write a b, aa to go
(21, b, b, R, 12)    write a b, ab to go
(21, Λ, b, R, 10)    write a b, aΛ to go

(22, a, b, R, 21)    write a b, ba to go
(22, b, b, R, 22)    write a b, bb to to
(22, Λ, b, R, 20)    write a b, bΛ to go

(20, Λ, b, S, halt)   write a b, then halt
(20, a, b, S, halt)   write a b, then halt
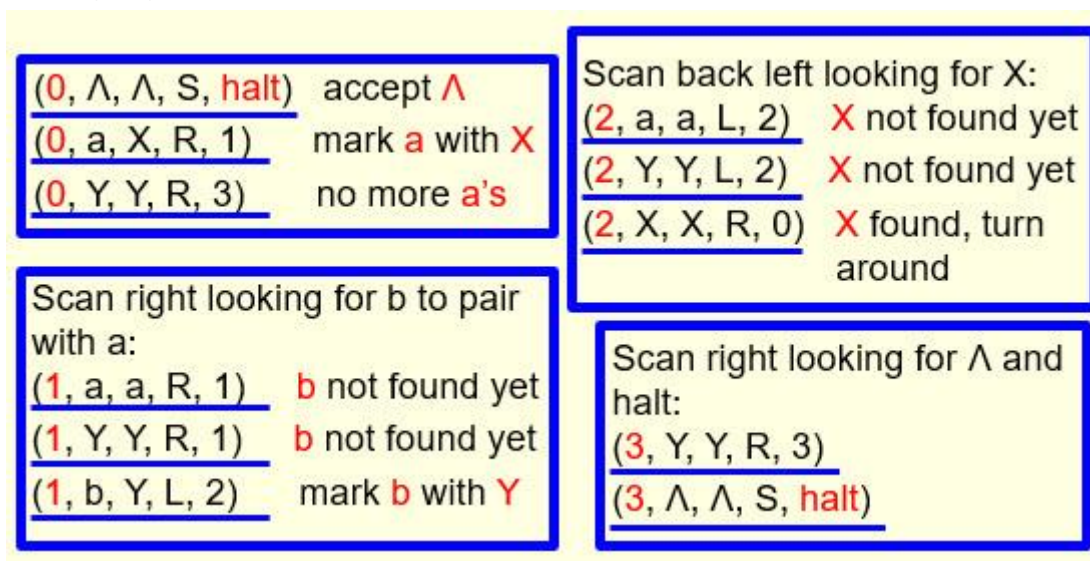(20, b, b, S, halt)   write a b, then halt

The following is the process of moving the string ababb to the right two cells position by this TM. Fill out the blanks below including instructions required for each step in the process.

| a | b | a | b | b | Λ | Λ | | | |

State = 0

(0, a, Λ, R, 1) — *Instruction for this step*

| Λ | b | a | b | b | Λ | Λ | | | |

State = 1

(1, b, Λ, R, 12) — *Instruction for this step*

| Λ | Λ | a | b | b | Λ | Λ | | | |

State = 12

(12, a, a, R, 21) — *Instruction for this step*

| Λ | Λ | a | b | b | Λ | Λ | | | |

State = 21

(21, b, b, R, 12) — *Instruction for this step*

| Λ | Λ | a | b | b | Λ | Λ | | | |

State = 12

**(12, b, a, R, 22)** — Instruction for this step

| Λ | Λ | a | b | a | Λ | Λ | | | |

State = 22

**(22, Λ, b, R, 20)** — Instruction for this step

| Λ | Λ | a | b | a | b | Λ | | | |

State = 20

**(20, Λ, b, S, halt)** — Instruction for this step

| Λ | Λ | a | b | a | b | b | | | |

State = halt

3. (2 points)

The following is the instruction set of a Turing machine (TM) that accepts the language { $a^n b^n$ | n ∈ **N** }.

(0, Λ, Λ, S, halt)   accept Λ
(0, a, X, R, 1)   mark a with X
(0, Y, Y, R, 3)   no more a's

Scan right looking for b to pair with a:
(1, a, a, R, 1)   b not found yet
(1, Y, Y, R, 1)   b not found yet
(1, b, Y, L, 2)   mark b with Y

Scan back left looking for X:
(2, a, a, L, 2)   X not found yet
(2, Y, Y, L, 2)   X not found yet
(2, X, X, R, 0)   X found, turn around

Scan right looking for Λ and halt:
(3, Y, Y, R, 3)
(3, Λ, Λ, S, halt)

This TM does not accept the string aaabb. Why? Put your reason in the following box. You cannot simply say the number of a's is not the same as the number of b's. The TM cannot tell if the number of a's is not the same as the number of b's. It would reject a string only if it reaches a point that it cannot proceed any further and the state is not the 'halt' state.

| X | X | X | Y | Y | Λ | |
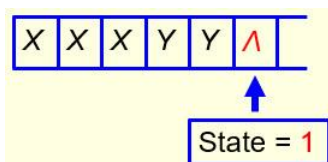
State = 1

Once the third 'a' is marked with 'X' (the first two a's are already marked with 'X' and the two b's are already marked with 'Y'), the state is changed to '1' and the read-write head moves rightward to look for a 'b' to mark with 'Y'. However, when it skips the two Y's already marked and enter the entry immediately to the right of the second 'Y', there is no 'b' there, but a 'Λ' (see the figure on the left), and there is no instruction for the TM to execute for such a combination (state=1, input=Λ), so the TM stops execution with the state being '1'. So the string 'aaabb' is rejected.

4. (9 points)
The following is the instruction set of a Turing machine (TM) that computes the sum of 2 and a given natural number represented as a binary string. The read/write head of the TM starts at the right end of the given natural number and halts at the left end of the output string. The start state of the TM is 0.

Move one cell left:
(0, 0, 0, L, 1)
(0, 1, 1, L, 1)

Add 1
(1, 0, 1, L, 2)       Move left
(1, 1, 0, L, 1)       Carry
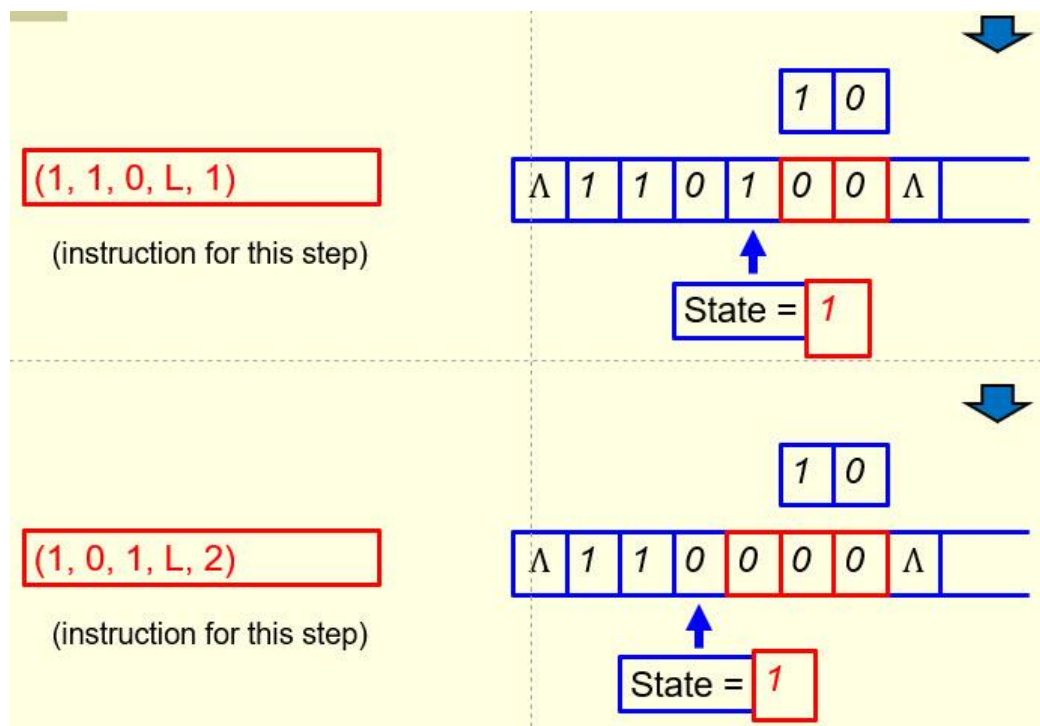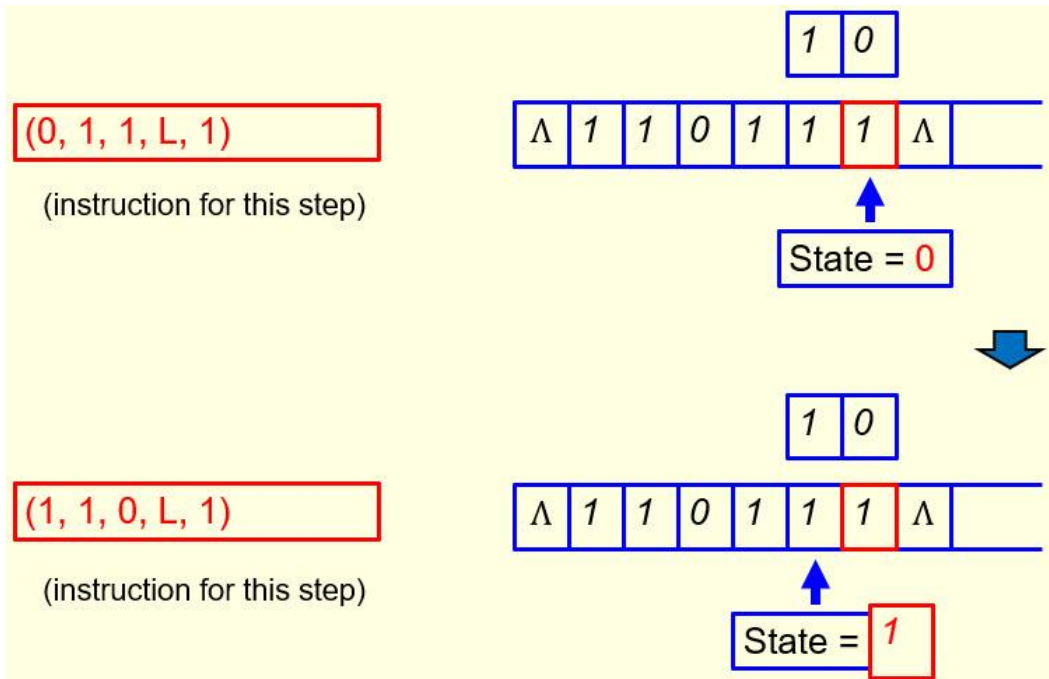(1, Λ, 1, S, halt)    Done
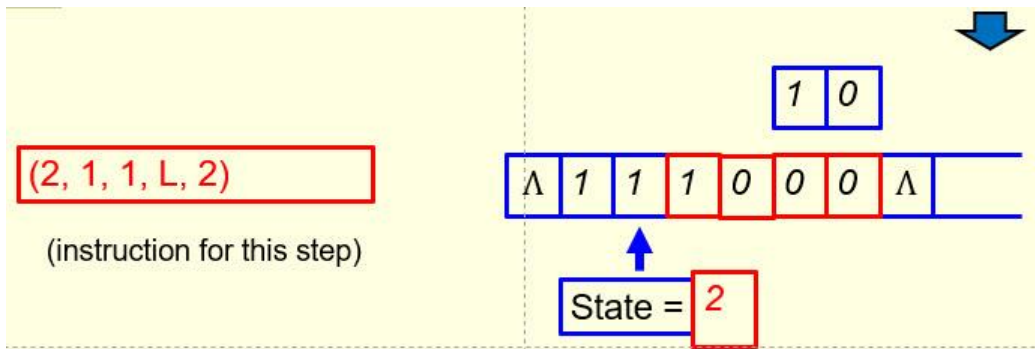
Find left end of the string
(2, 0, 0, L, 2)
(2, 1, 1, L, 2)
(2, Λ, Λ, R, halt)     Done

In the following, fill out the blanks in the following sum computation process

for the given natural number 54 (= $110110_2$). For each step, put the instruction to be used for that step in the blank on the left side, and fill out the blank(s) in the I/O tape on the right side. (1 point each blank)



(0, 1, 1, L, 1)

(instruction for this step)

State = 0



(1, 1, 0, L, 1)

(instruction for this step)

State = 1



(1, 1, 0, L, 1)

(instruction for this step)

State = 1



(1, 0, 1, L, 2)

(instruction for this step)

State = 1

(2, 1, 1, L, 2)

(instruction for this step)

From this point on, every digit read by the TM will be kept the same until the left end of the string is reached.

5. (16 points)

Given an integer say 45, to find the sum of 45 with 8 in binary form (see the figure below), we can use the TM designed in slides 79-90 of the notes "Turing Machines and Equivalent Models I" twice to find the result. A more effective way is to design a TM to do the addition with 8 directly.
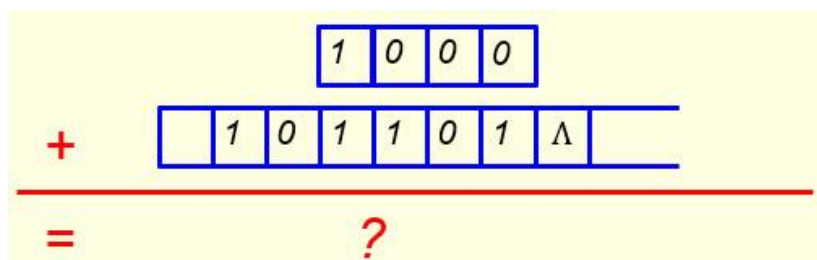


Such a TM can be designed by extending the TM designed in slides 79-90 of the notes "Turing Machines and Equivalent Models I". The TM has 14 instructions and 6 states: 0, 1, 2, 3, 4, and halt. Six instructions of such a TM are given in the first and the second tables below. Fill out the remaining blanks in the first and the second tables and also blanks in the third table to show the remaining instructions of the TM (you have to use your own text boxes here)

and then fill out the blanks in the following chart to make it a complete state transition diagram for this TM. Make sure the instructions you choose fit into the diagram naturally and logically. In the diagram, state H means the 'Halt' state. You have to use your own text boxes here too.   (1 point each blank)