The Australian National University
School of Computing
Dirk Pattinson and Victor Rivera

Semester 2, 2021
Practical Session 10

Foundations of Computation

The practical contains a number of exercises designed for the students to practice the course content. During the practical session, the tutor will work through some of these exercises while students will be responsible for completing the remaining exercises in their own time. There is no expectation that all the exercises will be covered in the practical session.

Covers: Lecture Material Week 10

At the end of this tutorial, you will be able to design Turing Machines given a language.

Exercise 1 — Even Number of 1s

Design a Turing machine that accepts all strings over $\{0, 1\}$ that contain an *even* number of 1s. Assume that the head is initially on the *leftmost* bit of the input string.

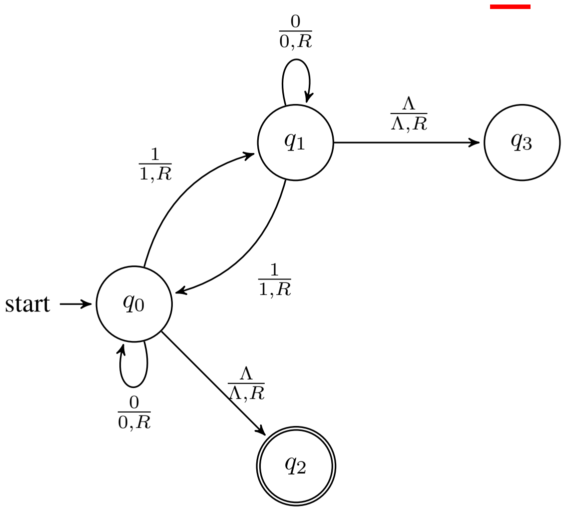
Represent the Turing machine with a state transition diagram, and give a brief description of the machine.

Solution.

The machine has two working states q_0 and q_1 and termination states q_2 and q_3.

q_0	an even number of 1s has been read
q_1	an odd number of 1s has been read
q_2	successful termination (a final state)
q_3	unsuccessful termination

Initially, when the head is on the leftmos 1s (precisely zero 1s which is an even number). We then sca q_1 (as we have now seen an odd number of 1s, i.e precisely on q_0 as we have now seen an even number of 1s. This conti Λ). We then terminate successfully from state q_0 (as we've seen an even number of 1s) and un (as we then have seen an odd number of 1s).

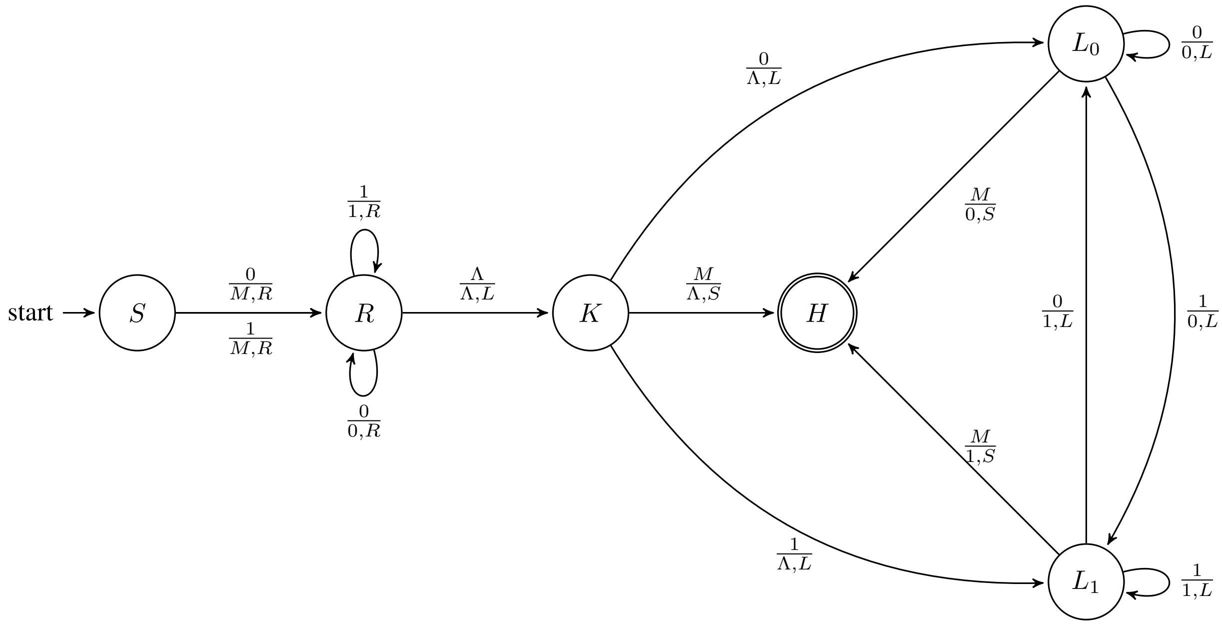


Exercise 2 — Deletion

Construct a Turing Machine that deletes the symbol at the current head position, by moving all the symbols to the right of this position one tape square to the left, and returns the head to the original position.

Represent the Turing machine using a state transition diagram in the format used in the lectures, and give a brief description of the machine.

Solution.



Overview:
- Overwrite current symbol with M
- Move to right-hand end of string (state R)
- Move left, until reaching M; each step remember the symbol seen by going to state L_0 or L_1; write the symbol previously seen (which is 0 or 1 if you are in state L_0 or L_1).

State S: Initial: write M

State R: Go to right hand end of strin

State K: If still at symbol M … e write blank (you are now beyond the … L_0 or L_1.

State L_i: Write symbol remembered (i), remember current symbol j … epeat this until you see M; then halt.

State H: Having seen M, and overwritten it, halt.

Exercise 3 **Length of the input**

Implement a Turing machine that computes the length of its input. That is, construct a Turing machine that

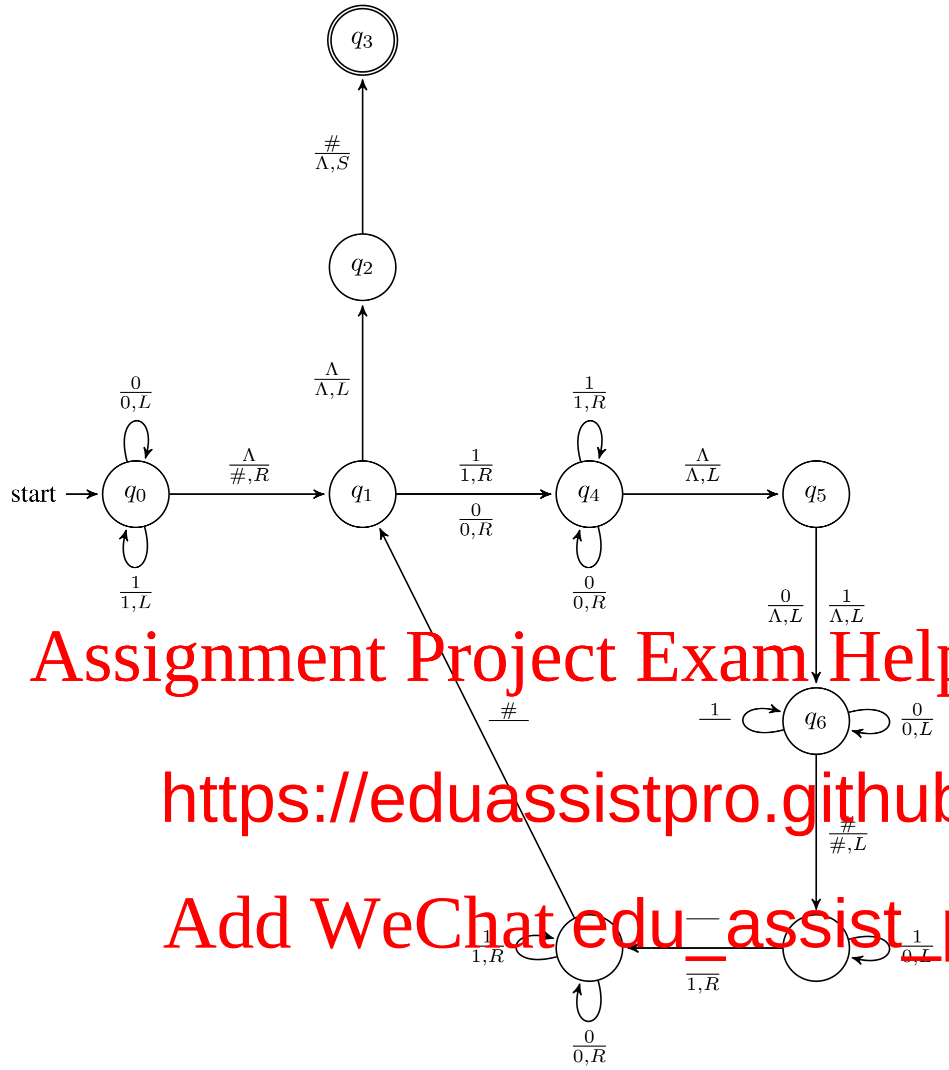
- replaces its tape content with the binary coding of the length of the input string.
- leaves on the tape the binary coding of the length of the input string.
- outputs the length of the input string. The tape should only contain the output.
- replaces its input string with the binary coding of the length of the input string. (The output does not necessarily need to be on the position as the input.)

Solution. Overview.

- scan to the left and write a '#' immediately on the left of the input. State q_0 scans to left, and in state q_1 we are immediately right of the '#' symbol.
- if there is nothing on the right of '#', erase the '#' and terminate. This is accomplished using states q_2 and q_3.
- otherwise, scan to the right of the input. In state q_5, the head points to the rightmost bit of the input string.
- delete the rightmost bit of the input string, and scan back (to the left) until we have found '#'. In state q_7, the head points to the bit immediately to the left of '#'.
- increment the binary number to the left of '#'. This is as in the lectures, and we are in state q_8 after incrementing.

- search to the right until '#' is found, move to the square immediately right of '#' and repeat (starting again in state q_1).

A transition diagram looks as follows:



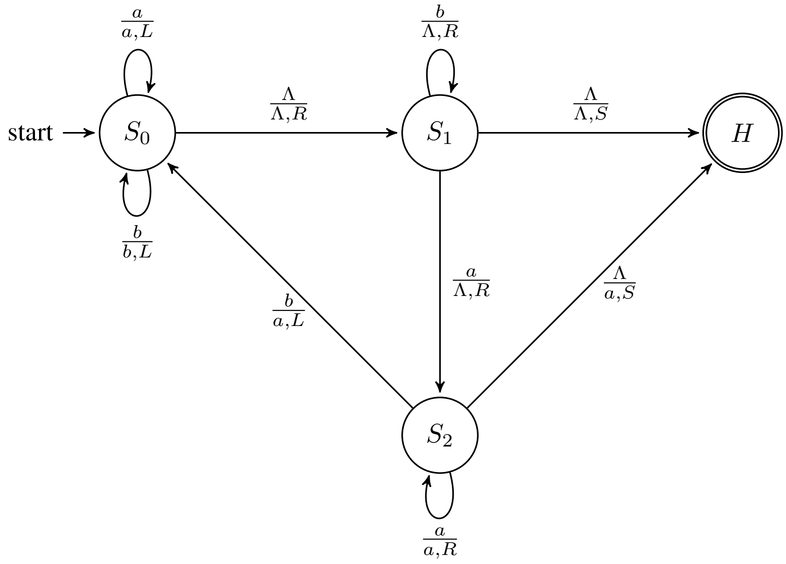
Exercise 4 **Selective Removal**

Give the state change diagram for a Turing machine that takes a tape containing a string of 'a's and 'b's and removes the 'b's. The initial string on the tape is terminated at each end by a blank symbol (Λ). If the initial string has n a symbols then the final string will be exactly n 'a's with no embedded blanks.

(Hint: the original 'a's are all identical, they don't have to be in the same order in the final string).

Solution.

- **Overview:** The TM repeatedly
  - moves to the left end of the string (state S_0)
  - deletes bs on the left end of the string (state S_1)
  - deletes the left-most a
  - scans right to find a b (or blank beyond the right-hand end) and overwrites it by a (state S_2)
- **State S_0:** This is the state that is occupied while the TM scans left to find the end.
- **State S_1:** This state indicates the TM is deleting leading bs while looking for the first a on the tape, if any.
- **State S_2:** In this state the TM is carrying a single a to the right. It replaces the first b it finds by this a, or drops it on the first blank symbol if no more bs are on the tape.



Exercise 5 **Adding Two to a binary Number**

The lecture notes give a Turing machine that adds one to a binary number. Construct a Turing machine that adds two instead. Assume the input data is represented on the tape with the least significant bit on the right. For example, the binary number for 139 (decimal) would be

on the tape. Assume the head is initiall e transition diagram, and give a brief description o

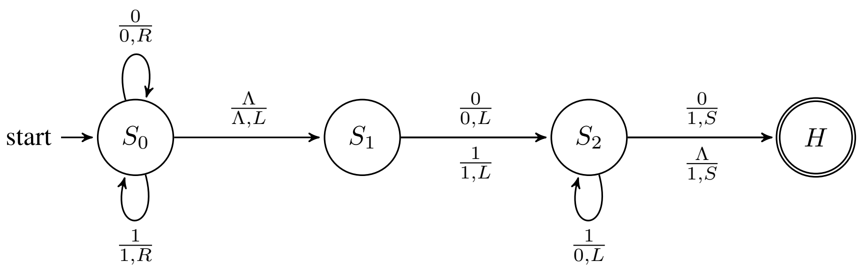
Solution. Recall from the lecture n e following things:

S_0**:** move the pointer head to the right most bit of the input;

S_1**:** add 1 to the current bit of the input, move the pointer left one cell, repeat if there is a ca

H**:** halt the Turing machine.

To add two instead, we only need to modify the Turing machine in lecture notes such that it adds one to the two's digit instead:

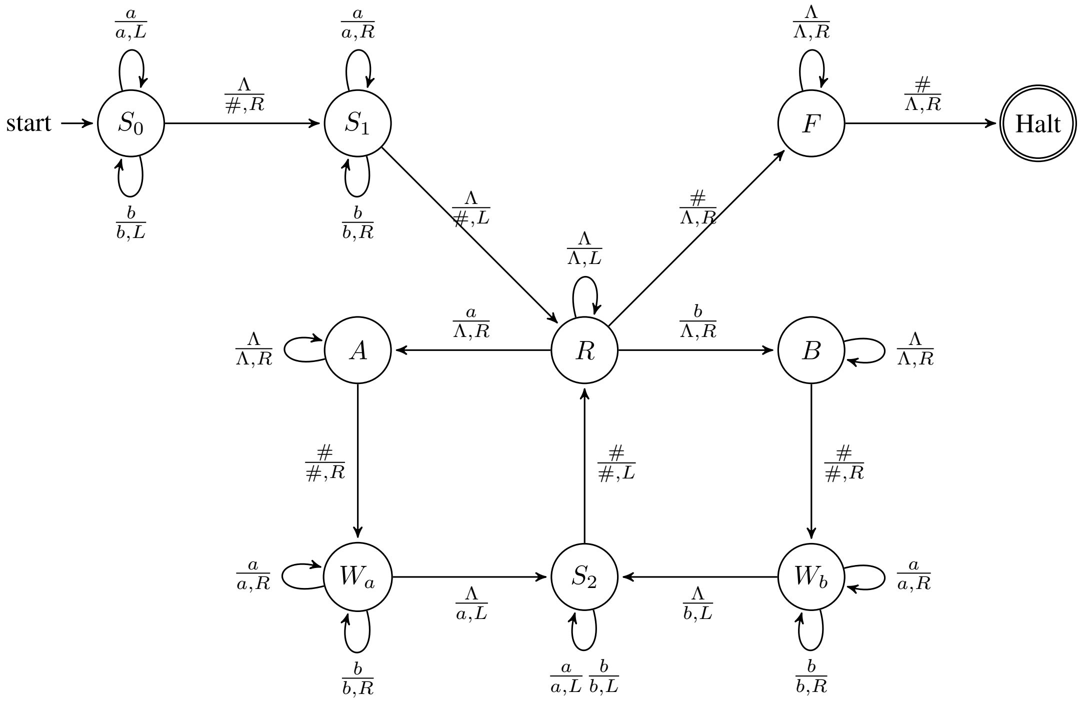


Exercise 6 **String Reversal**

Construct a Turing machine which, given a string over the alphabet $\{a, b\}$, replaces it by the reverse of that string. For example, if the input is $aaababb$ it will be replaced by $bbabaaa$. Assume that the read/write head is anywhere over the string to be reversed initially.

Represent the Turing machine using a state transition diagram in the format used in the lectures, and give an description of the machine in plain English.

Solution.



Description of the machine.

- In the first phase (marking) of this machines operation (states $S0$ and $S1$) an input of $aaababb$ becomes $\#aaababb\#$.
- In the main (transfer) phase, su ied to the right end of the reversed stri $\#aaab\Lambda\Lambda\Lambda\#bba$ and the final result of the phase is

- Last is the cleanup phase (state $bbabaaa$

Details of the meanings we associate with the various states is given in the following ta

S_0	Scan left to plant left marker
S_1	Scan right to plant right marker
R	Scan left to find letter to be moved
A	Scan right (over blanks) to marker, remembering a
W_a	Scan right (over $\{a, b\}$) to end and write a
B	Scan right (over blanks) to marker, remembering b
W_b	Scan right (over $\{a, b\}$) to end and write b
S_2	Scan left (to hash mark) over reversed string
F	Scan over blanked-out source and remove 2nd marker

Exercise 7 **Palindromes**

Construct a Turing Machine that recognises palindromes over the alphabet $\{a, b\}$. In the initial state the tape of the TM will have a string made up of a's and b's with no embedded spaces and with the read head somewhere over the string.

Your machine should halt in an accepting state if (and only if) the string is a palindrome. Your answer should consist of two parts:

- A description in plain English that simply describes how your machines works.
- A diagram for your Turing Machine, in the notation used in lectures.

Solution. The machine works by iteratively checking that the leftmost and rightmost symbols on the tape are the same. After each successful match those two symbols are erased. The looping terminates successfully when either

- At the start of an iteration the leftmost symbol is also the rightmost one.

- After erasure of matching symbols the tape is blank.

On successful termination, the machine writes the symbol T onto the tape – this is strictly optional.

The state transition diagram of the machine is the following.