Turing Machines (TM)

Jay Bagga

1 Introduction

A Turing Machine (TM) is a powerful model which represents a general purpose computer. The Church-Turing thesis states that our intuitive notion of algorithms is equivalent to TM algorithms. In this module, we discuss the concept of a TM through an example and use JFLAP to simulate and test the TM.

A TM can be defined in terms of its components in a manner similar to DFA, NFA and PDA etc. Informally, a one-tape TM consists of a single infinite tape that is divided into cells. Each cell can contain a symbol. A read-write head can scan the tape one cell at a time by moving to the right or left (or staying on a cell). There are also variations of TMs with several tapes and heads. In this module we restrict our discussion to a TM with a single tape and single head. Initially the cells are all blanks and an input string is placed on contiguous cells. The tape head then scans the input and moves according to the TM's transition function. If the TM reaches an accept state the input is accepted. Otherwise a TM may reject an input string or may never halt.

2 An example TM

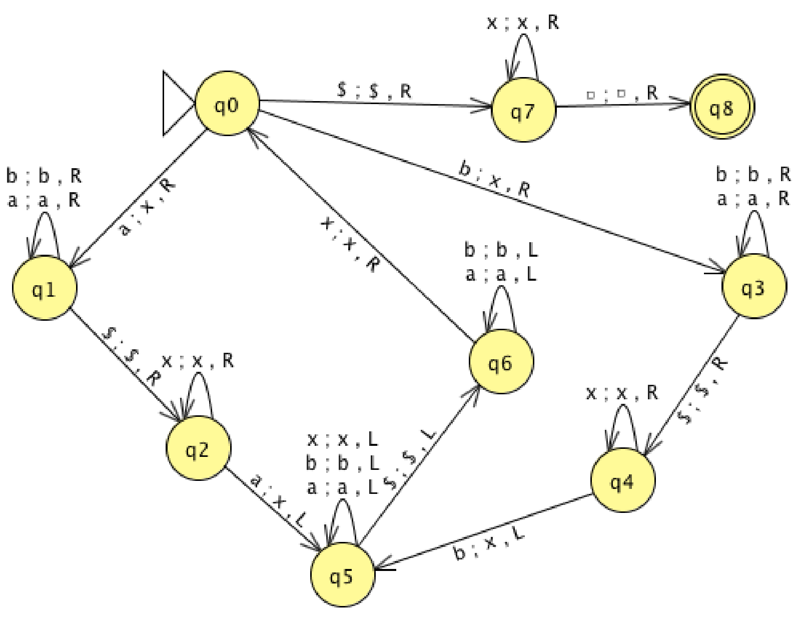
A TM can also be described by a state-transition diagram. Look at such a diagram of a TM below. We shall use JFLAP to test this TM with several inputs. We observe that this TM has nine states $q0$ through $q8$, with $q0$ as the initial state and $q8$ as the final state. Initially the tape head is at the leftmost symbol of the input. A transition such as

$$q0 \xrightarrow{a;x,R} q1$$

is interpreted as follows: If the TM is in state $q0$ and the tape head reads a in the current cell, then it writes x in that cell and moves one cell to the right (R). The little square symbols in the transition from $q7$ to $q8$ denote the special blank symbol.

Question 1.

1. Load the TM in the file TMv7.jff.
2. For each transition in this TM describe in words what the transition means.



For the alphabet $\Sigma = \{a, b, \$\}$, let us now consider the language $L = \{w\$w | w \in \{a, b\}^*\}$. Thus L is the set of words that have any string of a's and b's followed by one \$ sign followed by the same string. For example, $ab\$ab$, $aaa\$aaa$, and \$ are words in L, while $ab\$aa$, aba and \$\$ are not in L. We want to construct a TM that accepts precisely the words in L.

A standard method of building a TM is to use the Church-Turing thesis. Thus, we first describe an intuitive algorithm for recognizing words in L, and then we build a TM for this algorithm. Our intuitive algorithm is stated in terms of reading an input on a tape. So to recognize that an input is of the form $w\$w$, we look at the leftmost character of the first w and try to match it with the first symbol to the right \$. We then repeat this process for each symbol in w. To keep track of what symbols have already been read, we replace a symbol (a or b) with an x as soon as it is read for the next match. Thus the tape head zig-zags between the symbols to the left of the \$ sign and to the right of the \$ sign, matching them one at a time. If a match fails at any step, the input is rejected.

We now show that the TM shown above implements this algorithm. For example, starting at the initial state $q0$, if an a is read, it is replaced by an x and the TM moves to the state $q1$. Now we are looking to match the a just read to an a to the right of the \$ sign. We stay in $q1$ as we read through the other a's and b's. Once we reach the \$ sign, the TM moves to the state $q2$. We move though the x's and look for an a. If at any time this match does not occur, the TM halts in a non-accept state and the input is not accepted. If an a is found, the TM moves left through all the symbols until it finds the \$ sign and then keeps moving left until it is back to $q0$, at which point the TM repeats the process, taking a path toward $q3$ if a b is read. If all the a's and b's are matched and the TM returns to $q0$, it then moves right through the $q7$ and reads only x's to reach the accept state.

You will now simulate this TM in JFLAP and test it for several inputs.

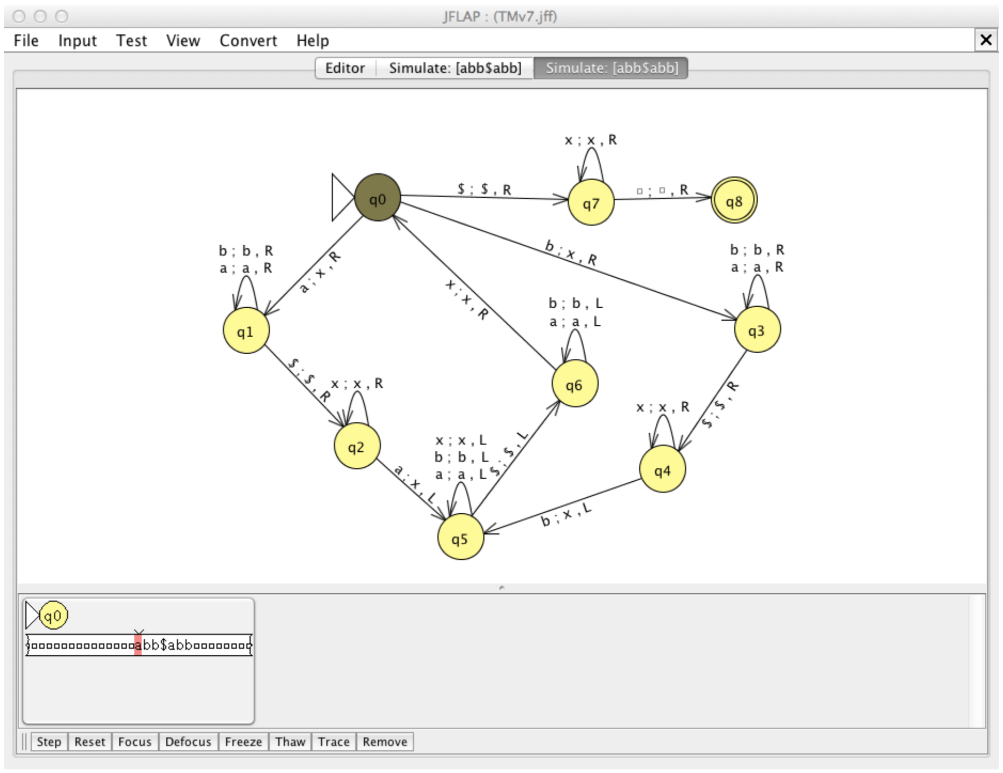
Question 2.

1. Load the TM in the file TMv7.jff. This is the same TM as shown above.

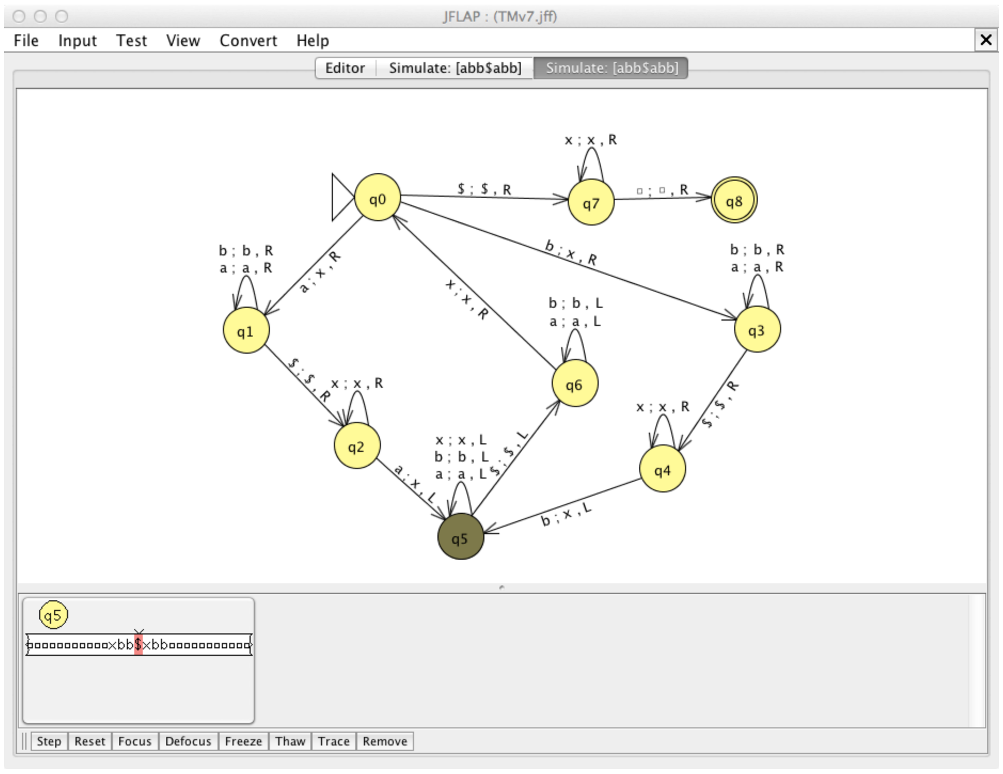
2. Study the path $q0 \to q1 \to q2 \to q5 \to q6 \to q0$ and the transitions along this path. Explain the actions of the TM along this path.

3. Repeat the above step for the path $q0 \to q3 \to q4 \to q5 \to q6 \to q0$.

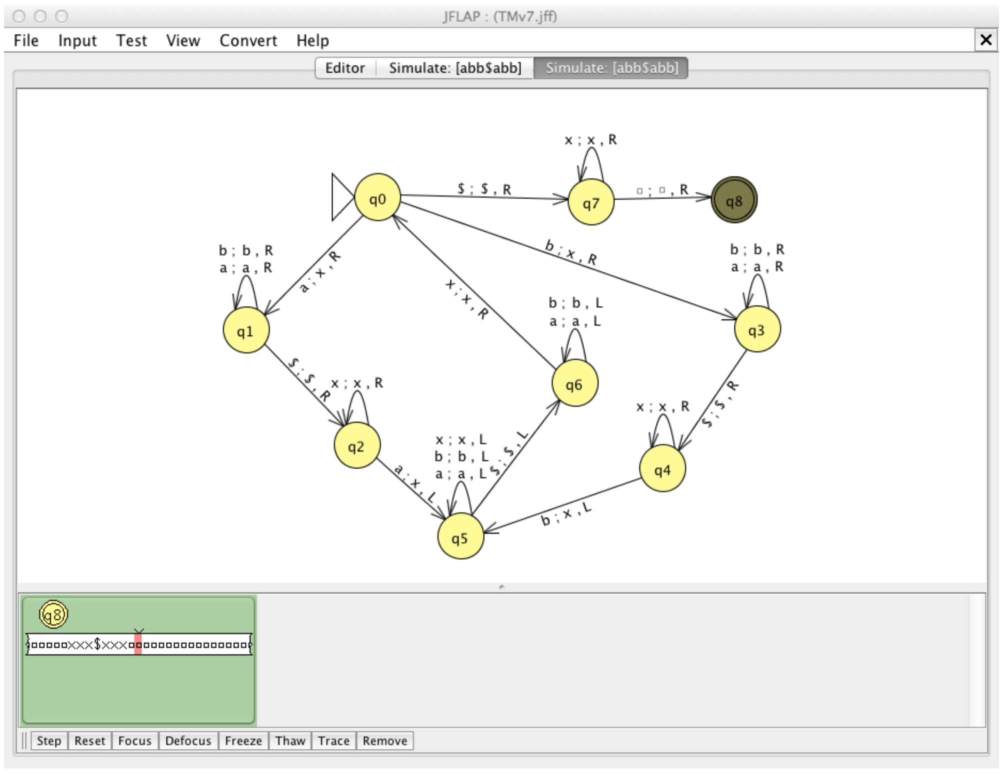
4. Choose Input from the menu and click on Step... Enter the input *abb$abb* and click OK. Verify that the you have the window as shown below. At this stage the initial state $q0$ is highlighted. In the lower left corner, you see the current state, the tape and the current cell highlighted. See the diagram below.



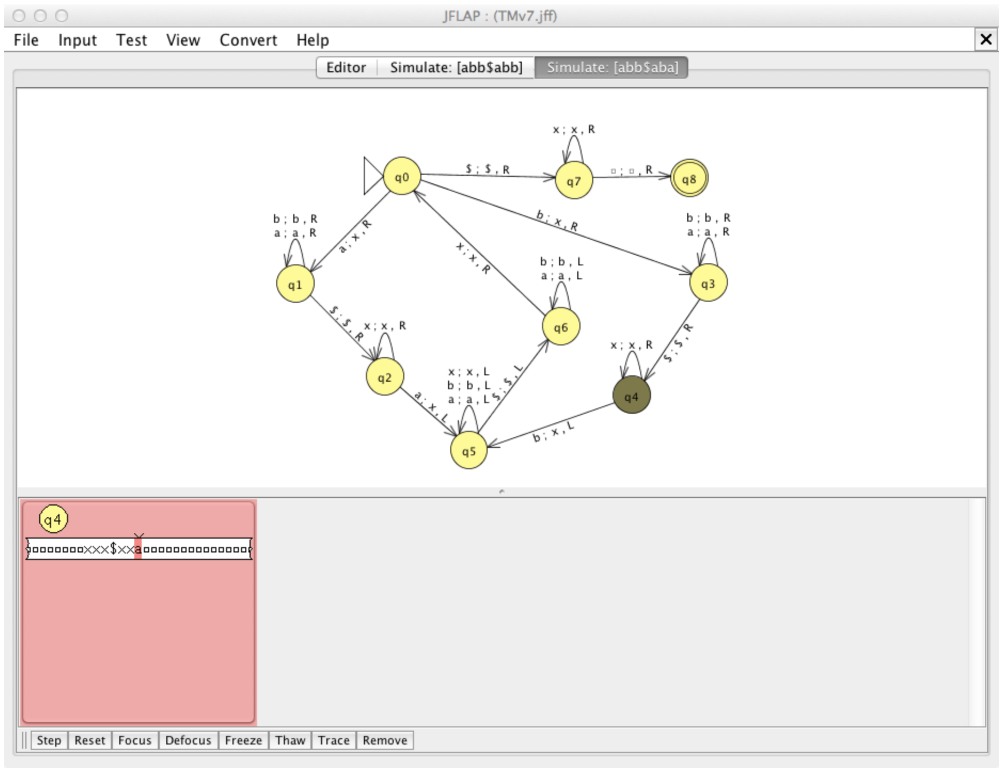
5. Click on the button $\boxed{\text{Step}}$ in the lower left corner. Describe the changes you see on the tape and the current state. Explain those changes. Continue to step through until the state in again at $q0$. Compare this with you answer in part 2. above. The diagram below shows an intermediate step.



6. Continue to step through until you see the state $q8$ highlighted and the lower left portion of the window is green. See the diagram below. Is the input accepted?

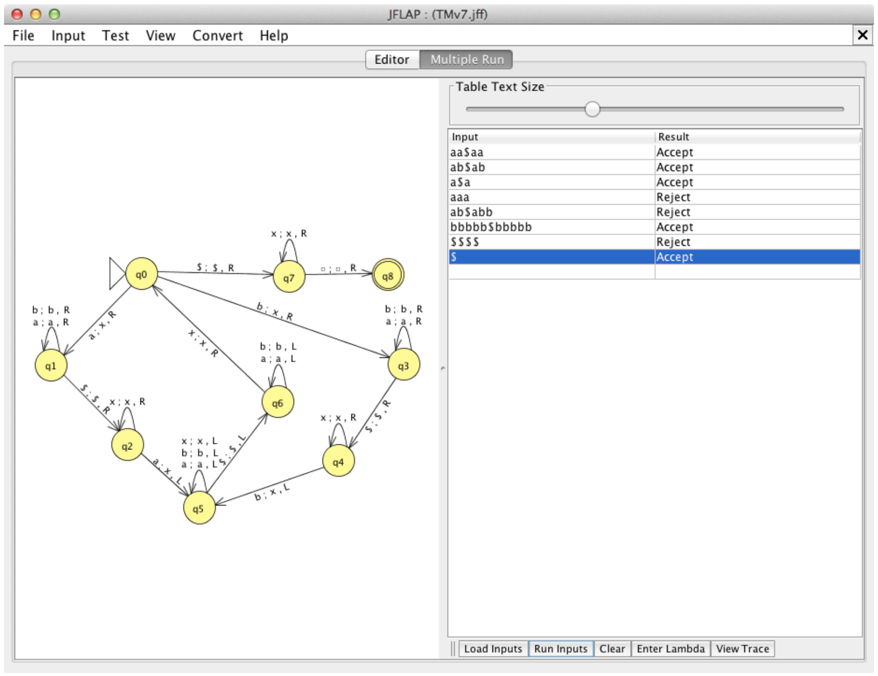


7. Repeat the above steps with a new input: *abb$aba*. See the diagram below. Is this input accepted? Why or why not? What happens after the TM reaches *q*4?



Question 3.

1. You will now test the TM with multiple inputs.


JFLAP : (TMv7.jff)
File Input Test View Convert Help
Editor
Multiple Run
Table Text Size
Load Inputs Run Inputs Clear Enter Lambda View Trace

Input	Result
aa$aa	Accept
ab$ab	Accept
a$a	Accept
aaa	Reject
ab$abb	Reject
bbbbb$bbbbb	Accept
$$$$	Reject
$	Accept

2. Load the TM in the file TMv7.jff.
3. Choose Input in the menu and click on Multiple Run. See the diagram above. Enter inputs as shown in the diagram. Explain why each input is accepted or rejected. Enter four more different inputs, two of which should be accepted and two rejected.
4. If $q7$ is also made an accept state, how would it change the language of words that are accepted by the modified TM? Explain your answer. Test your answer with several inputs in JFLAP.

3 References

1. Introduction to the Theory of Computation (Third Edition), Michael Sipser. Cengage Learning. 2013.
2. JFLAP - An Interactive Formal Languages and Automata Package, Susan H. Rodger and Thomas W Finley. Jones and Bartlett Publishers. 2006