

Sheila Gallardo Redondo

Alexandra Perruchot-Triboulet Rodríguez

b) Inputs

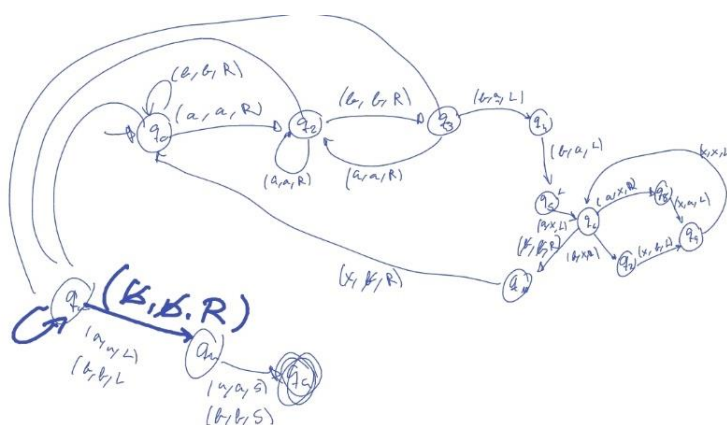
Input	Output	Result
abb	aa	Accept
aaaaabababb	aaaaababaa	Accept
aaabbaaaabbababa	aaaaaaaaaababa	Accept
abaa	abaa	Accept
aabba	aaaa	Accept
abb	aa	Accept
aaaaabababb	aaaaababaa	Accept
aaabbaaaabbababa	aaaaaaaaaababa	Accept
abaa	abaa	Accept
aabba	aaaa	Accept
aaybbba		Reject
abbabbabb	aaaaaa	Accept
bbabbab	bbaaab	Accept

c) Explanation of the algorithm:

At first, we created 3 states to detect the pattern 'abb'. If we detect this pattern, we change each 'b' for the corresponding 'a's' and we have marked the original a with an x to be able to read the rest of the word without leaving a blank space in between the symbols.

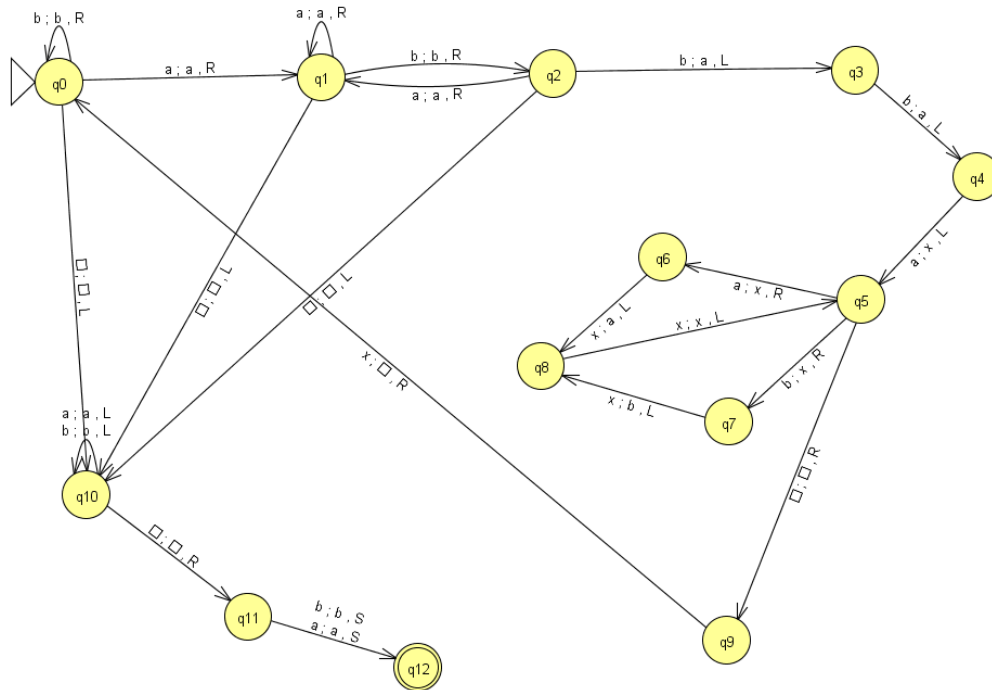
The idea is to move the x to the start of the string and so we have considered two cases if there is an ' a ' or a ' b ' before the ' x '. Then we interchange the value of the symbol we have detected and the ' x ', and so on until we arrive at the start of the word, then we delete the ' x ' replacing it with a blank space and we start to detect new patterns again in the string.

If there are no more patterns, then we set the head pointer to the beginning of the string and we stop the turing machine.



d) Formal definition of the Turing Machine:

TM = ({a, b}, {a, b, x, blank}, blank, {q0, q1, q2, q3, q4, q5, q6, q7, q8, q9, q10, q11, q12}, q0, f, q12)



e) Explanation of the alphabet symbols and states and transitions:

We have added an 'x' as an alphabet symbol to help us to make the changes without leaving any blank space in the string.

We have three main groups of states. The first one is to detect the pattern asked in the practice ('abb'), the second one is to be able to move the 'x' out of the string and so we do not leave blank spaces in between the string, and, the last group is to leave the head pointer of the tape at the beginning of the word and stop the machine.