Example: One Tape Turing Machine$_{\mathbf{JP}}$

Define a Turing Machine M that decides the language L = { *wcw*| *w* □ {a,b}* }

Recall that JFLAP defines a Turing Machine M as the septuple M = (Q, Σ, Γ, δ, q_s, □, F) where

- Q is the set of internal states {q_i | i is a nonnegative integer}
- Σ is the input alphabet
- Γ is the finite set of symbols in the tape alphabet
- δ : Q × Γ → Q × Γ × {L,R} is the transition function
- □ is the blank symbol.
- q_s (is member of Q) is the initial state
- F (is a subset of Q) is the set of final states

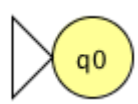
Sample Solution (see: `TM_wcw.jff`)

One approach to defining this Turing Machine (TM) takes advantage of the existence of a single symbol, **c**, marking the end of the first occurrence of *w* and the beginning of the second occurrence of *w*. The TM can determine if the initial symbol and the first symbol following **c** are identical. If so, they can be removed from further consideration and each subsequent character checked for a match. If the symbols are ever different, the string is not in language L. If the symbol **c** is reached before the end of the input string, then the string is not in language L. If the end of string is reached before **c**, then the string is not in language L. Otherwise, because all symbols have been matched and the c symbol marks the exact center of the string, the string is in language L.

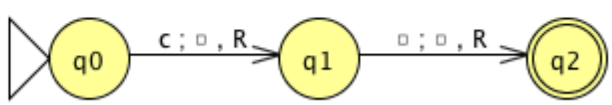
1. Identify the input alphabet: Σ = {a, b, c}

2. Identify the tape alphabet, which should include the input alphabet, the blank symbol, and another symbol to be used as a marker for symbols already processed: Γ = {a, b, c, 0, □}

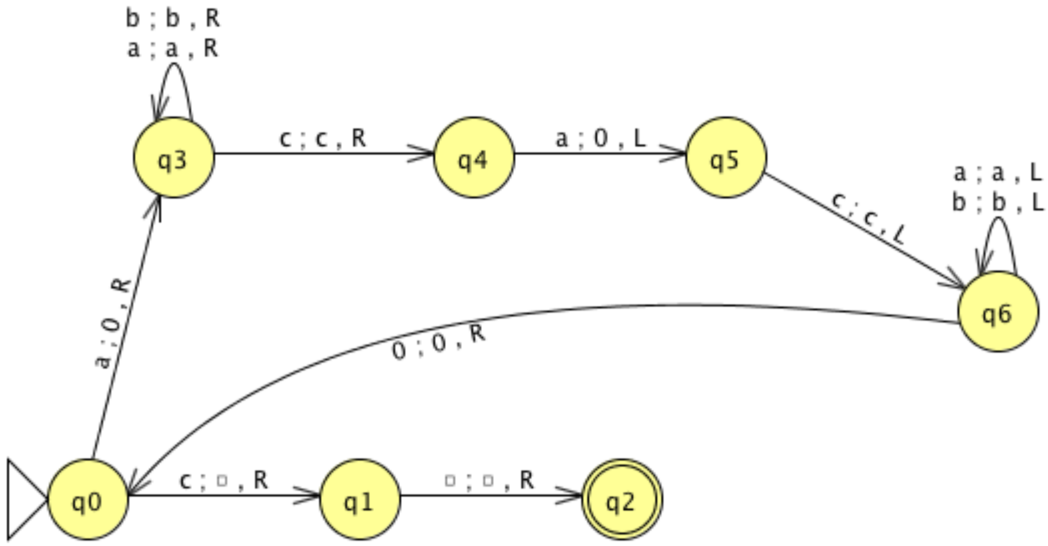
3. Create an initial state for the TM.



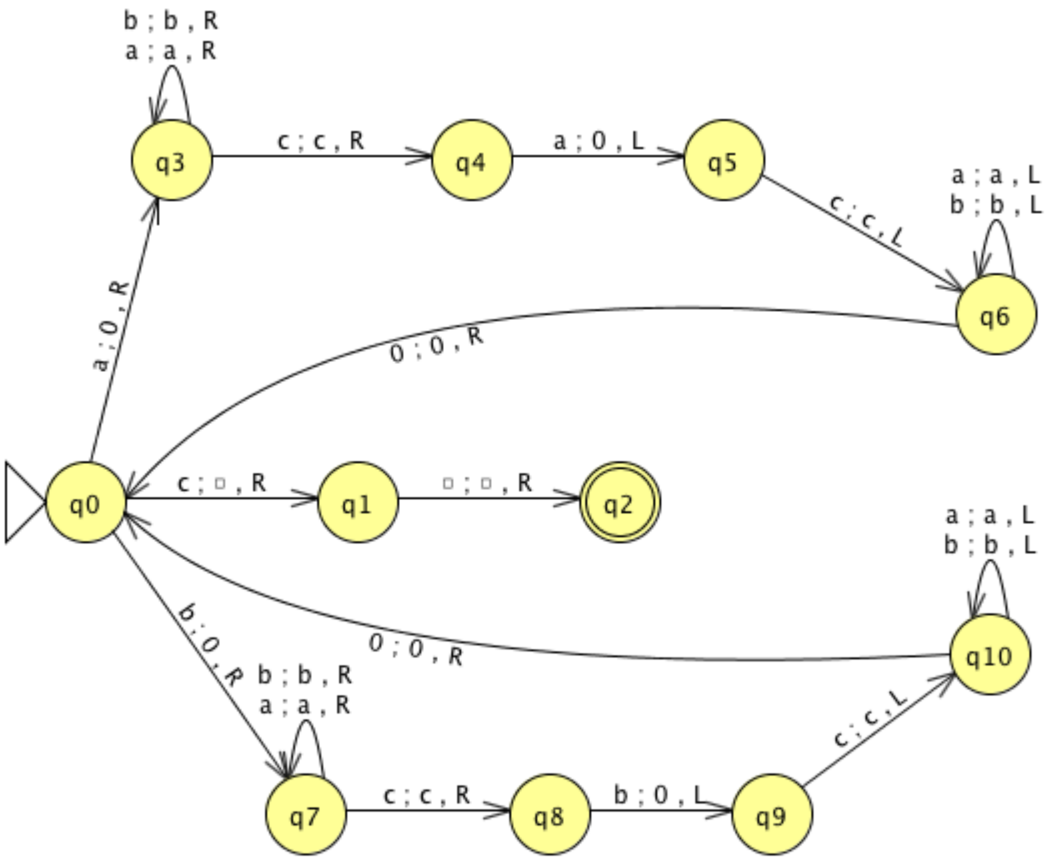
4. Address and accept the string "c".



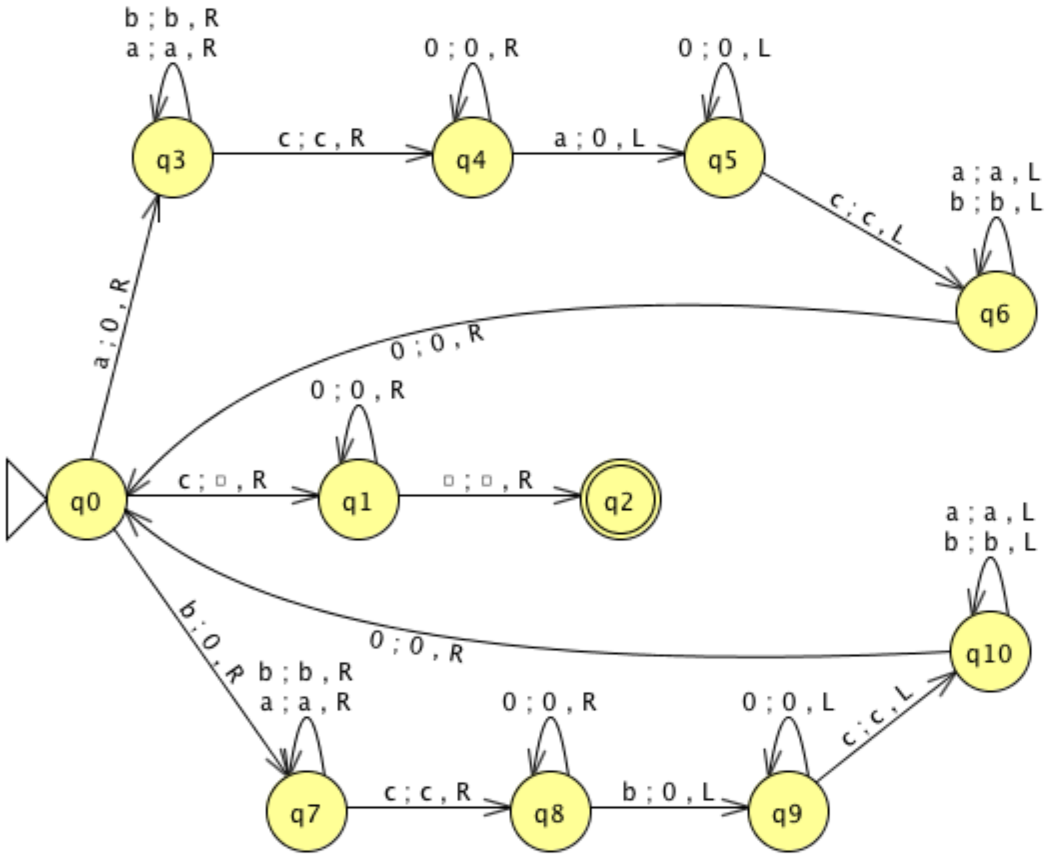
5. Address the case in which the current symbol is **a** by overwriting it with **0**, scanning right until the **c** symbol is found, verifying that the next symbol is also **a**, overwriting that with **0**, then scanning left to process the next symbol of the string.



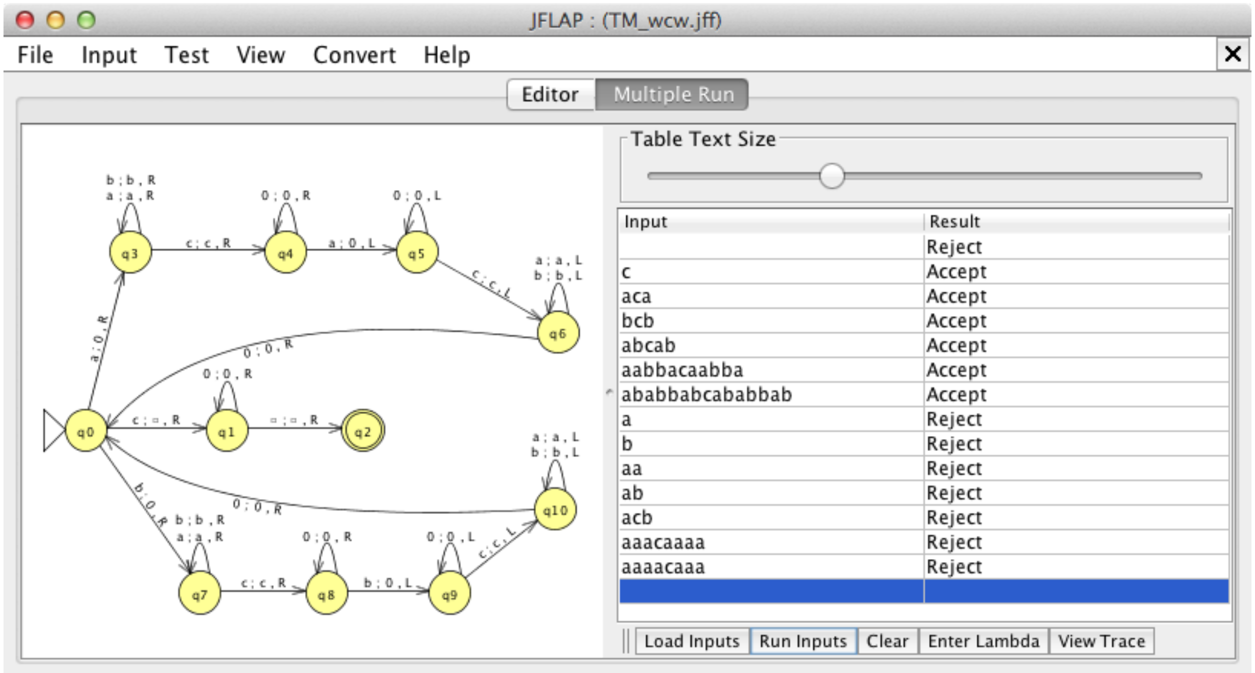
6. Address the analogous case in which the current symbol is **b**.



7. Handle subsequent symbols by skipping over markers in the second instance of *w* and unprocessed symbols in the first instance of *w*.


b ; b , R
a ; a , R
0 ; 0 , R
0 ; 0 , L
q3
c ; c , R
q4
a ; 0 , L
q5
c ; c , L
a ; a , L
b ; b , L
q6
a ; 0 , R
0 ; 0 , R
0 ; 0 , R
q0
c ; □ , R
q1
□ ; □ , R
q2
a ; a , L
b ; b , L
b ; 0 , R
0 ; 0 , R
q10
b ; b , R
a ; a , R
0 ; 0 , R
0 ; 0 , L
c ; c , L
q7
c ; c , R
q8
b ; 0 , L
q9

8. Check your TM by running multiple inputs and comparing with expected results.



Here is a formal definition of this Turing Machine that decides language L:

$$(Q, \Sigma, \Gamma, \delta, q_s, \square, F) = (\ \{q0, q1, q2, q3, q4, q5, q6, q7, q8, q9, q10\}, \{a, b, c\}, \{a, b, c, 0, \square\},$$
$$\delta \textit{ as defined in the preceding state diagram, } q0, \square, \{q2\}\ )$$

Note that states q6 and q10 could be collapsed into a single state. Likewise, states q5 and q9 could be collapsed into a single state. The result would be a TM with nine states instead of eleven.