DFA - Exercise

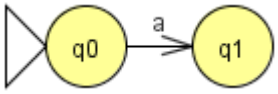
Problem:

Construct a DFA for L = {$ab^n a^m$: n ≥ 2, m ≥ 3}.

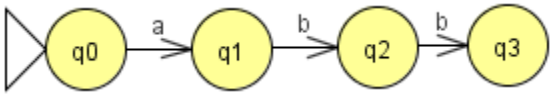
Solution:

We start by analyzing the type of strings accepted by this language. The string must start with an 'a'. This is followed by 2 or more 'b's. It is ended with three or more 'a's. Example strings include *abbaaa*, *abbbaaa*, and *abbbbbbbbbbaaaa*. The string *abb* is not in the language because there is no trailing a's. Similarly with *aaaa*, it is not accepted because there has to be two or more b's in the middle of the string.

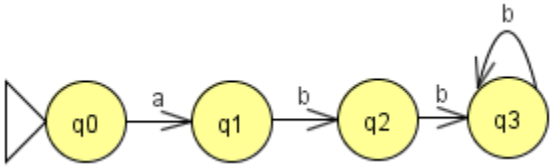
We start building the DFA with an initial state and a second state remembering that the first *a* is read. Note that we will focus on the strings to be accepted by the DFA. We will add a trap state to take care of all rejected strings at the end.



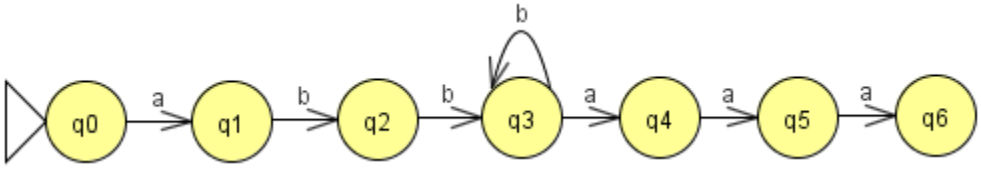
We build on this by adding two more states to remember the fact that two *b*'s have been read.



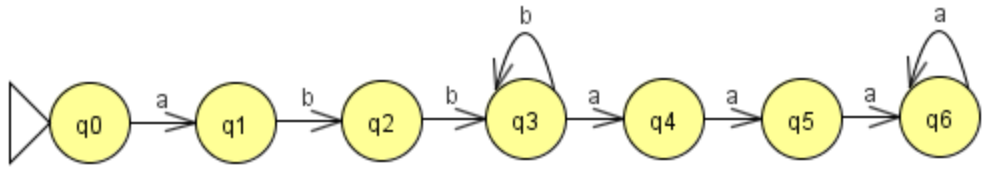
Next, we add a loop back on the last state to any number of *b*'s after the first two *b*'s.



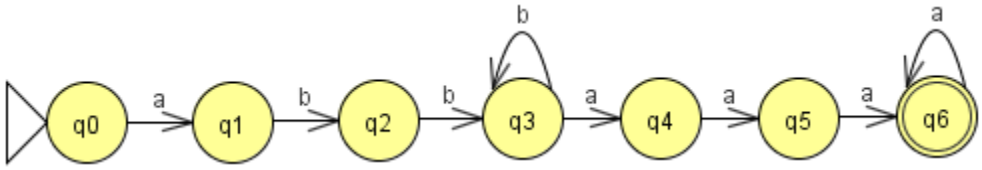
As we did with processing the two b's, we will now add three more states to remember three consecutive *a*'s read after the *b*'s.



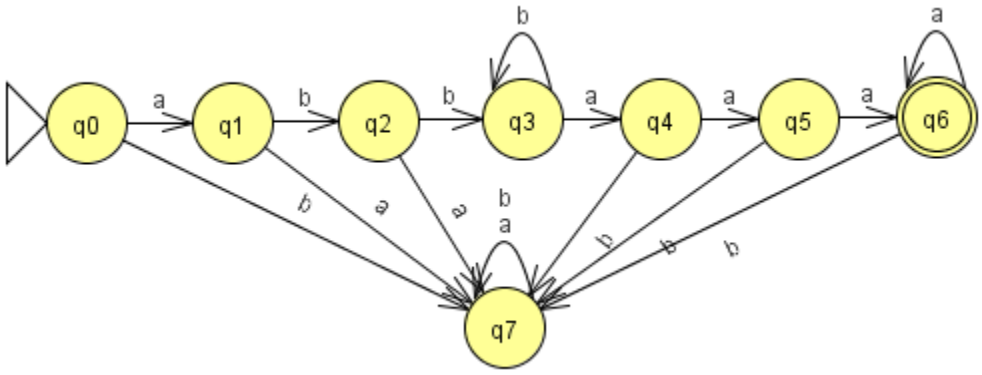
We add a loop back on the last state to accept any string of trailing *a*'s after the three *a*'s.



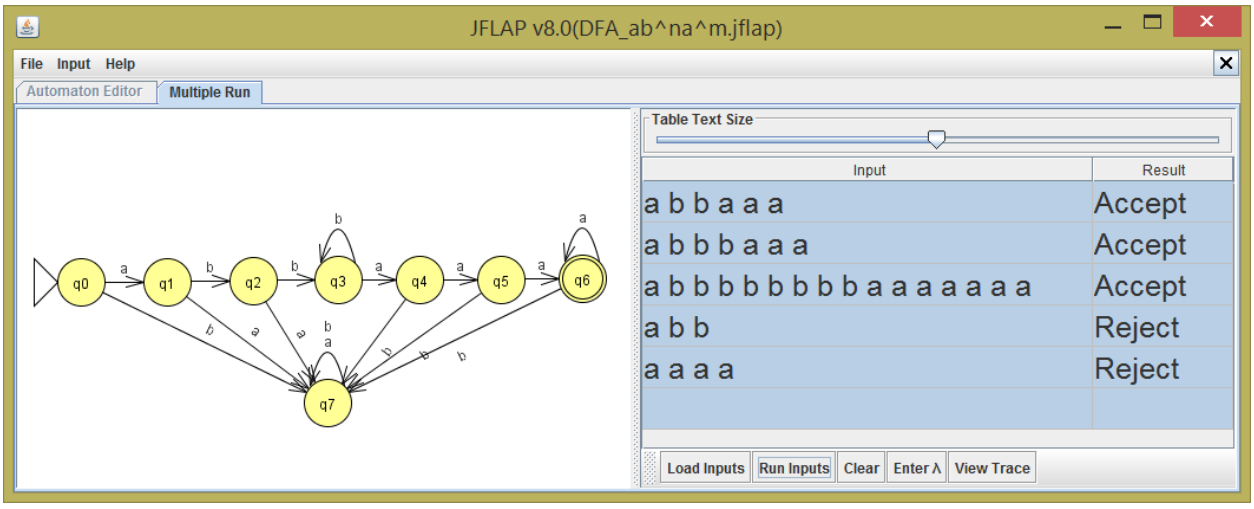
Next, we make the last state the final state to signify acceptance of the string.

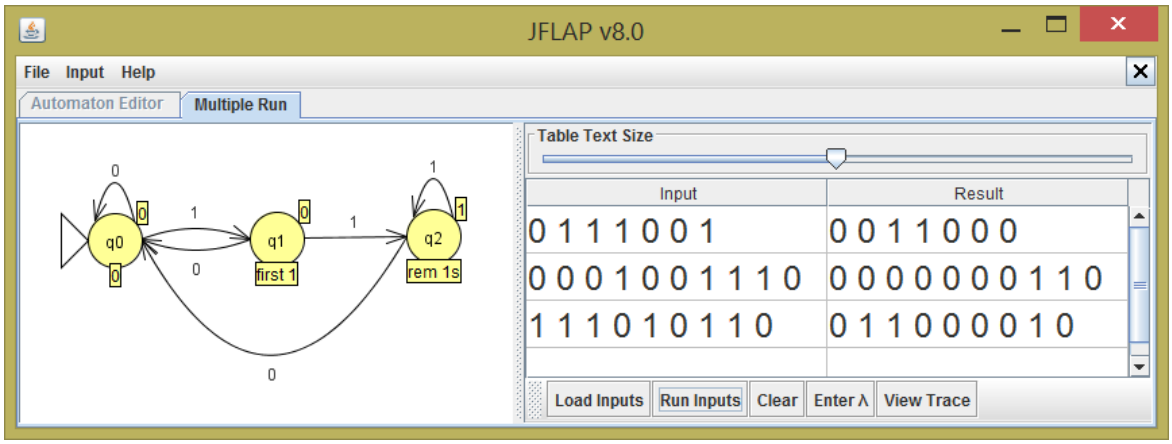


Finally, we add a reject state to send input sequences that are not in the language.



Test the DFA with sample strings.




JFLAP v8.0
File Input Help
Automaton Editor
Multiple Run
q0
q1
first 1
q2
rem 1s
Table Text Size
Input
Result
0 1 1 1 0 0 1
0 0 1 1 0 0 0
0 0 0 1 0 0 1 1 1 0
0 0 0 0 0 0 0 1 1 0
1 1 1 0 1 0 1 1 0
0 1 1 0 0 0 0 1 0
Load Inputs
Run Inputs
Clear
Enter λ
View Trace