

# Pushdown Automata (PDA) Exercise

Jay Bagga

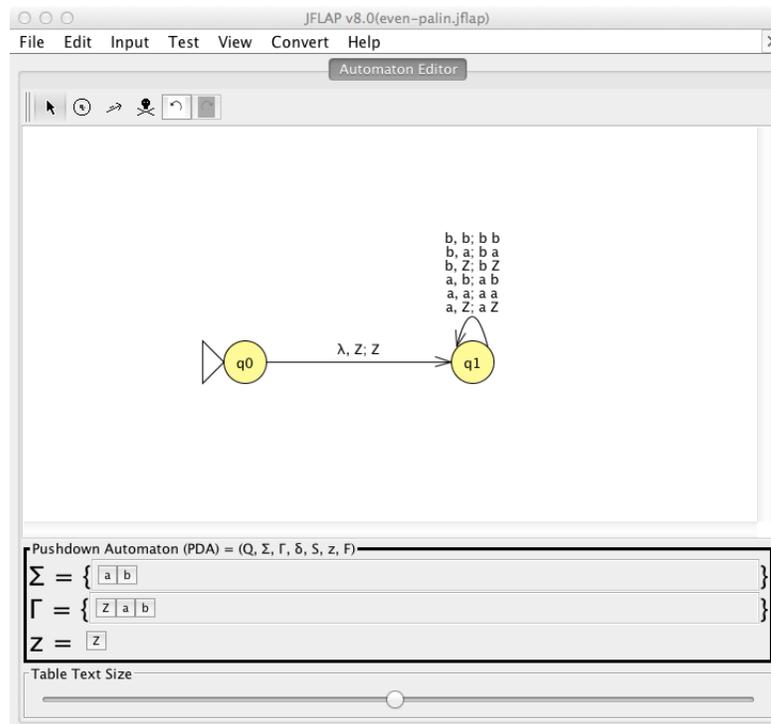
## 1 An example PDA

In this exercise, you will use JFLAP to build a PDA for a given language. Recall that a palindrome is a string of characters that reads the same forwards and backwards. Thus *aba*, *aaaa*, *MOM* and *RACECAR* are palindromes. Let us build a PDA for the following language  $L$  over the alphabet  $\Sigma = \{a, b\}$ .

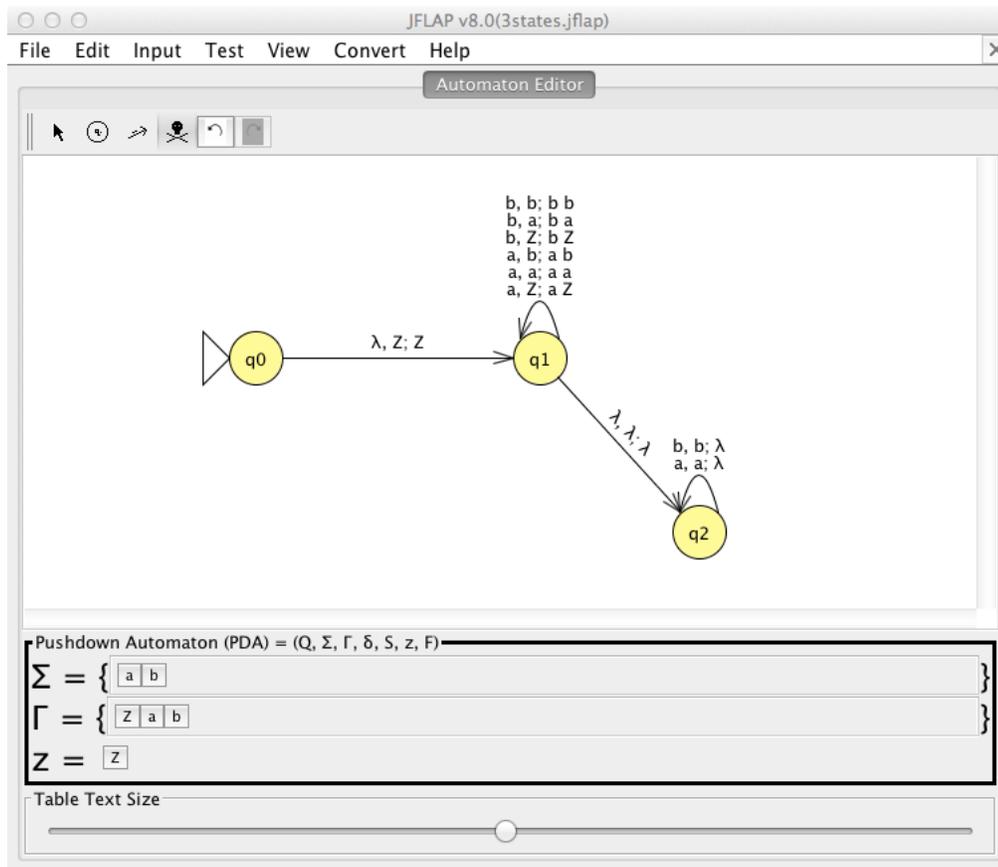
$$L = \{w \in \Sigma^* \mid w \text{ is a palindrome of even length}\}.$$

To build this PDA we will first try to describe the algorithm to accept even palindromes informally. We will then build the states and transitions.

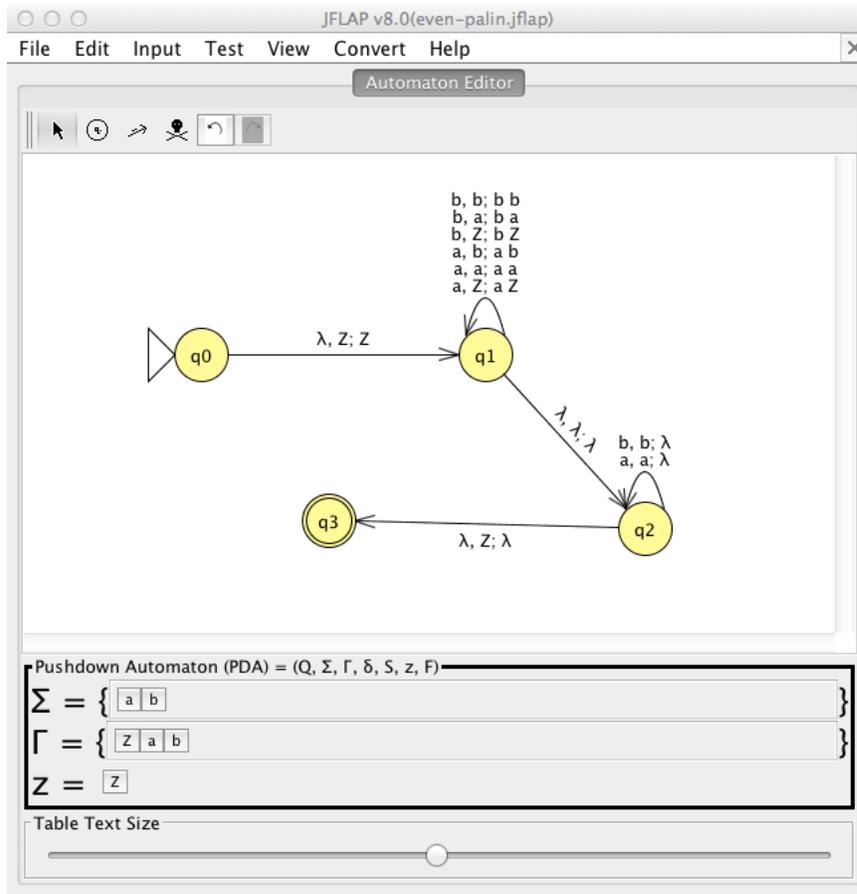
1. Let us look at an example of an even palindrome, say,  $w = aabbaabbaa$ . Observe that  $w$  can be written as  $w = uu^R$ , where  $u^R$  is the string  $u$  written in reverse. Here  $u = aabba$  and  $u^R = abbaa$ . In other words, a word is an even palindrome if and only if the first half of the word is the reverse of the second half. So for a PDA to recognize such a word, we can push the first half of the word on the stack and then pop it (so its pops in reverse order) and then match it to the second half. Of course, a PDA has no way to tell when you reach half-way in the input. So this is done nondeterministically. That is, we should stop pushing and start popping at any step and expect that at least one configuration will find the match.



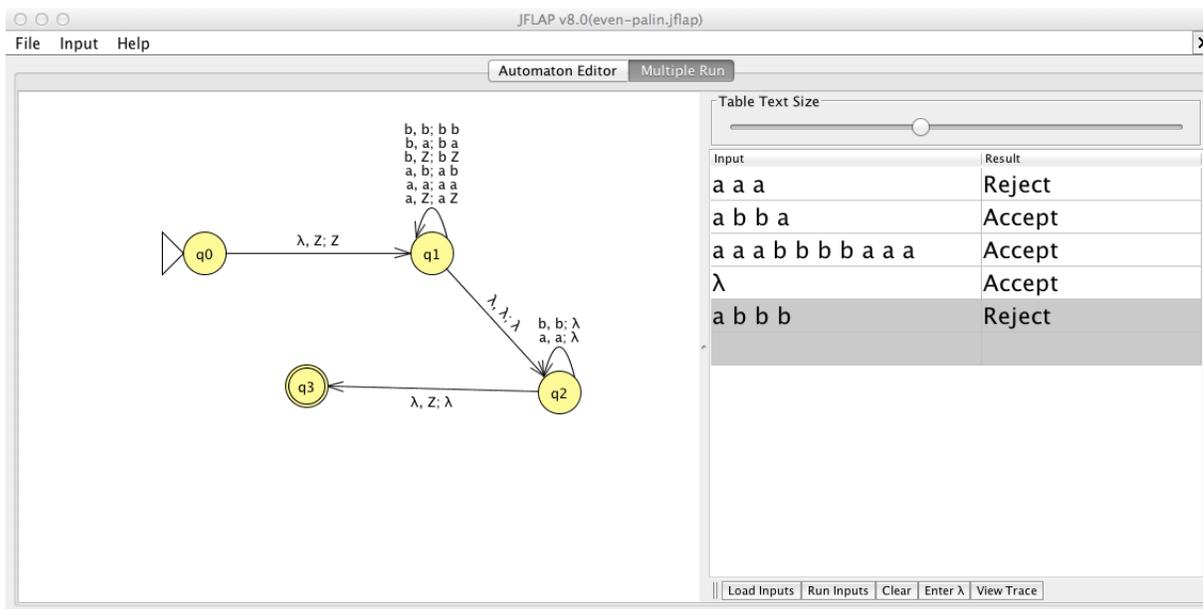
2. With the above informal description in mind, you'll now build the PDA. Open JFLAP and choose Pushdown Automaton.
3. Create an initial state  $q_0$  and with a transition  $(\lambda, Z; Z)$  move to a state  $q_1$ .  $q_1$  is the state where we try to push the first half of the input as it is read. At this stage, your PDA should be as above. Study each loop transition at  $q_1$  and explain its meaning.
4. Now make a nondeterministic jump to a new state  $q_2$  where you start popping the stack to match with the second half. See the diagram below and explain the meaning of the transition from  $q_1$  to  $q_2$  and the loops at  $q_2$ .



5. Finally, add an accept state  $q_3$ . When we reach the end of input, a  $Z$  should pop. This means that all the pushed symbols matched with the popped symbols. See below.



6. Now, you will test this NFA with several inputs as below. Verify that your results are the same as those shown. Test some more inputs.



## 2 Additional practice

1. Repeat the above steps to create a PDA for the language of all odd palindromes.
2. Repeat the above steps to create a PDA for the language of all palindromes.

## 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