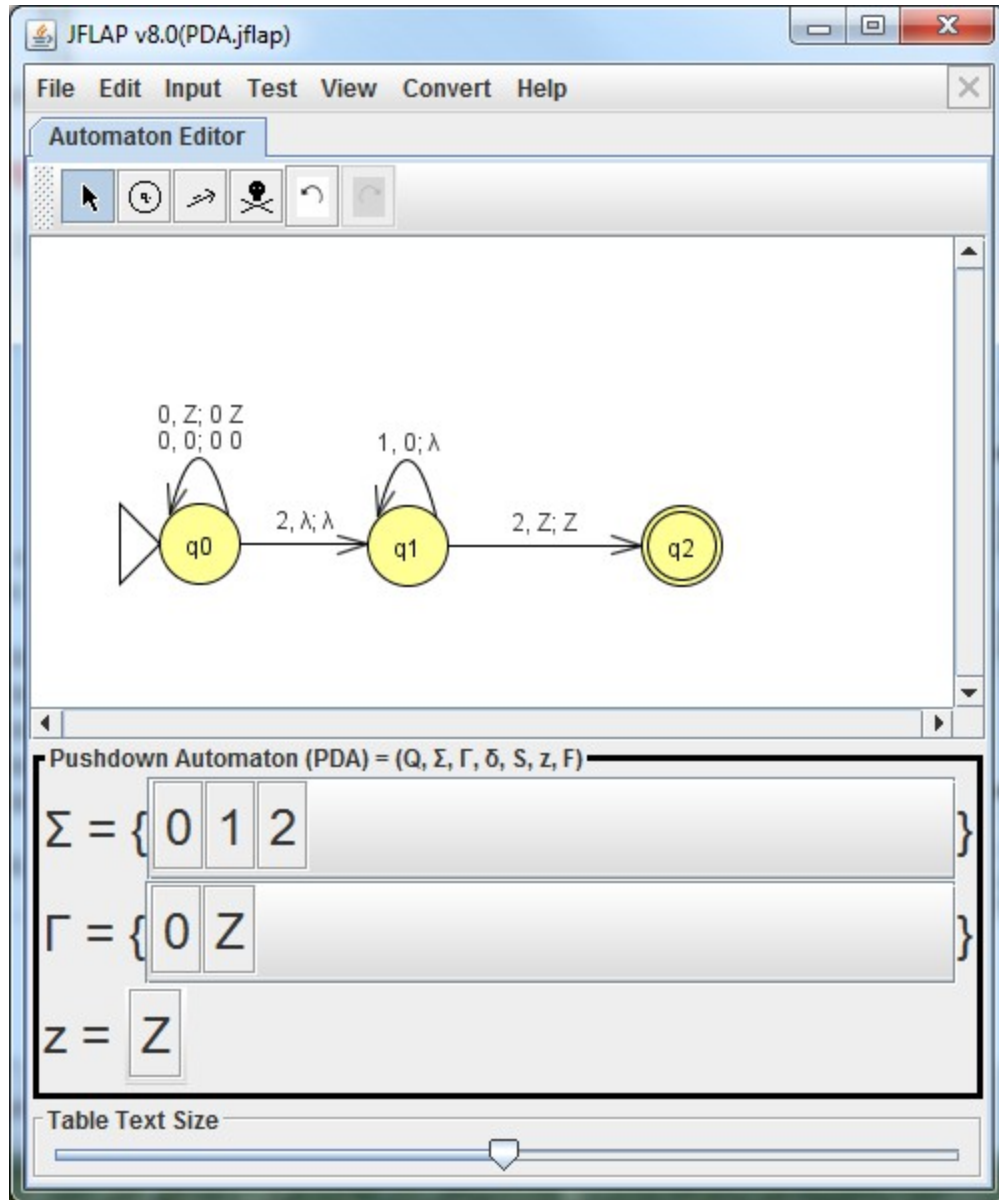Lets build up a PDA which accepts odd length palindromes consisting of strings of 0 or 1, with a 2 in the middle and a 2 at the end.

The first step, lets make one which accepts 0* 2 1* 2. So any number of 0s followed by a 2 with an equal number of ones terminated with a 2.


So in the initial state, we see a zero, and push a zero onto the top of the stack. You can see this, by the guard, [a,b,c] = [input value, b=pop(stack),push(c)]. So we read, [0,0,0 0] as, if the input symbol is zero, and we popped a zero from the stack, push the thing we popped, and the zero we read.

Pop of an empty stack, returns Z, so [0,Z, 0 Z] is our starting condition.

In state q1, on a 1 if we pop a zero we push nothing, and return to q1.   On a 2, if the stack is empty we accept.

Now we need to allow 1 to be put onto the stack in state q0, and in state q2, need to check and pop off the complementary character, so if we read a 1, we need to have a 1 on the stack, which we pop off, and if we see a 0 we need to pop off a zero from the stack.
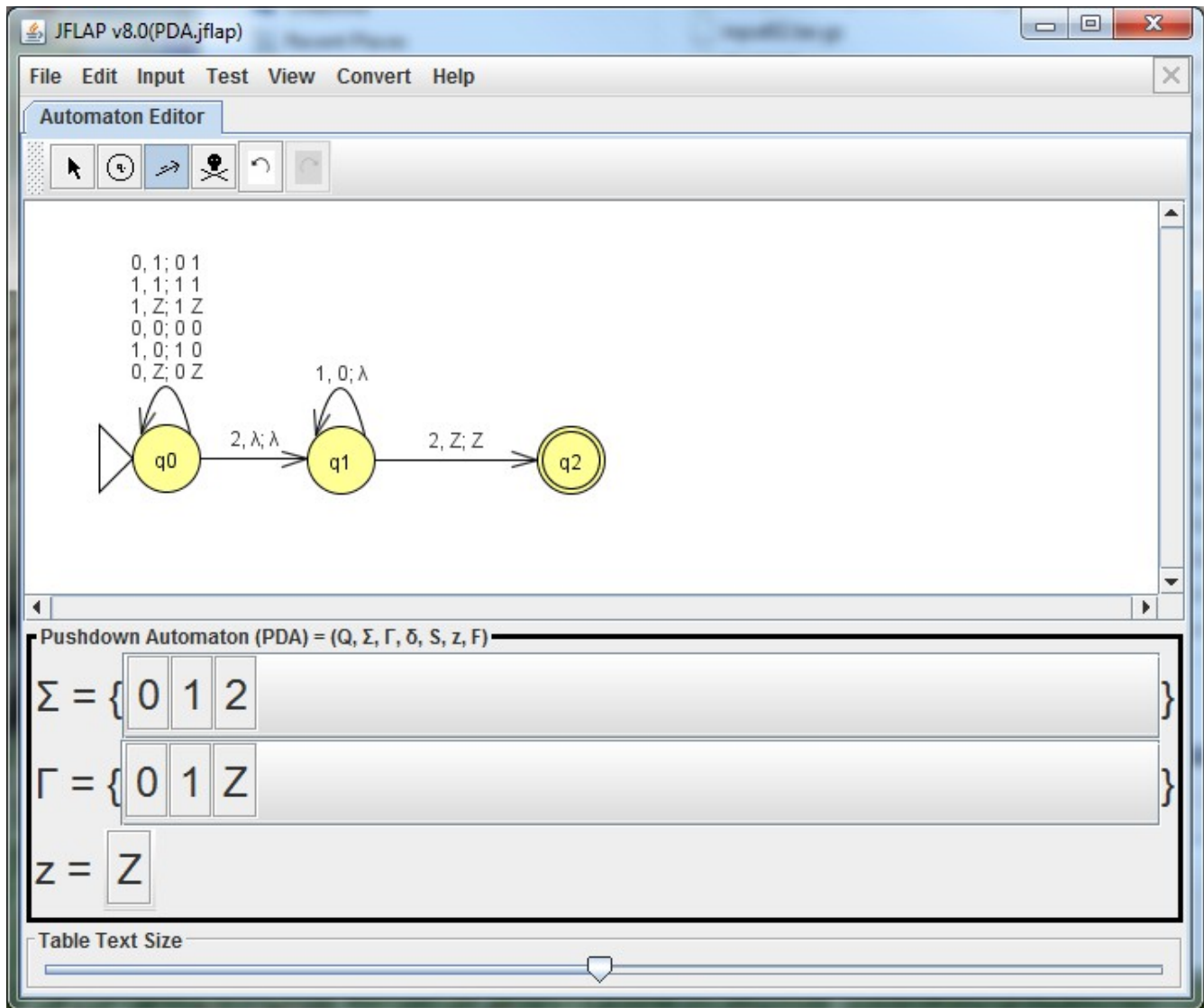
And in state q1, the item we read must equal the thing we popped, and we discard both.

File   Edit   Input   Test   View   Convert   Help

Automaton Editor

0, 1; 0 1
1, 1; 1 1
1, Z; 1 Z
0, 0; 0 0
1, 0; 1 0
0, Z; 0 Z

0, 0; λ
1, 1; λ

2, λ; λ   2, Z; Z

q0   q1   q2

Pushdown Automaton (PDA) = (Q, Σ, Γ, δ, S, z, F)

Σ = { 0 1 2 }

Γ = { 0 1 Z }

z = Z

Table Text Size