

# Converting a Regular Expression to DFA

Jay Bagga

## 1 Regular Expression to DFA

In this exercise we use JFLAP to construct a DFA that is equivalent to a given regular expression. We will use the regular expression  $(ab + a)^*$  for our exercise. Can you describe in words the language represented by this regular expression?

Choose Regular Expression in JFLAP and enter the regular expression  $(ab+a)^*$ . See Figure 1.

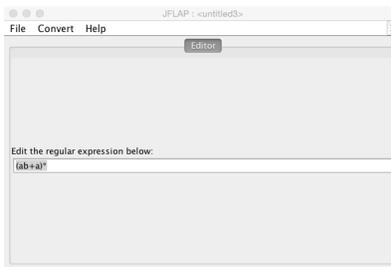


Figure 1: Input regular expression

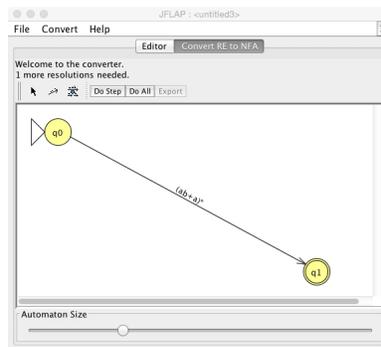
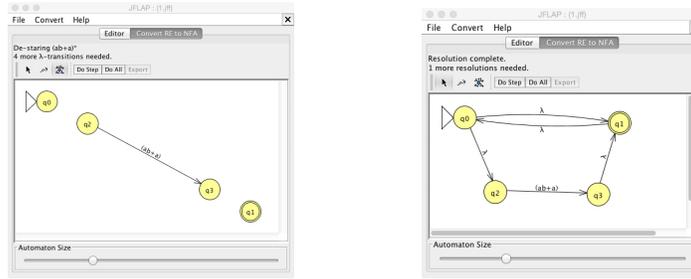


Figure 2: GTG from regular expression

Then select Convert: Convert to NFA. JFLAP creates a generalized transition graph as shown in Figure 2, with two states and a single transition labeled  $(ab + a)^*$ . We next need to break this transition into its simpler components.

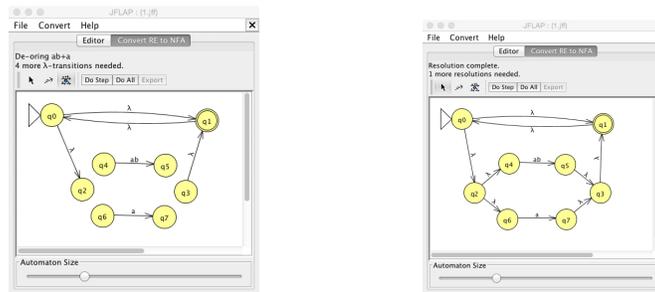
Click the “De-expressionify” transition button and then click on the transition label. This creates two new states and a transition labeled  $(ab + a)$ . See Figure 3. Since we are “de-starring” the regular expression, we need four  $\lambda$ -transitions to achieve the equivalent GTG. Where do these  $\lambda$ -transitions go?



A GTG needing  $\lambda$ -transitions    A GTG with  $\lambda$ -transitions

Figure 3: GTGs

Click the “Do Step” button to see the answer, as in Figure 3.



A GTG needing  $\lambda$ -transitions    A GTG with  $\lambda$ -transitions

Figure 4: GTGs

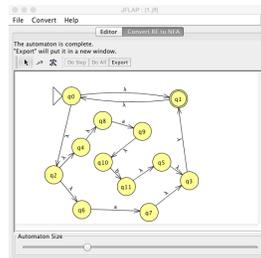
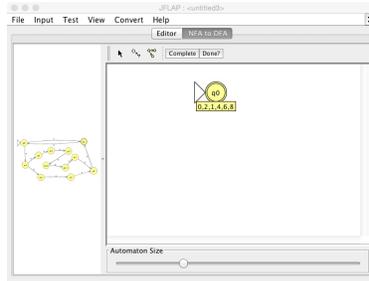


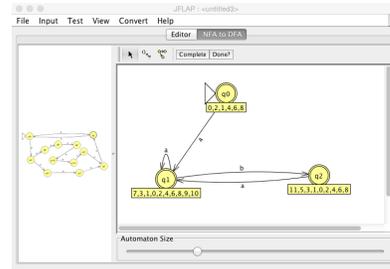
Figure 5: NFA

Next, repeat the De-expressionify transition process for the transition labeled  $ab+a$  to get the results shown in Figures 4. Repeat the same process for each of the remaining expressions on transitions. Think though the answers for placing  $\lambda$ -transitions before letting JFLAP do it for you. The NFA that results is shown in Figure 5.

We finally convert this NFA to a DFA. Review the appropriate lesson, the algorithm and exercises on how this is done. Here we review one intermediate step. Click on Convert to



NFA to DFA conversion - first step



Final DFA

Figure 6: NFA to DFA

DFA. As a first step, you get the result as in Figure 6. JFLAP creates a new state  $q_0$  by grouping the equivalent states  $q_0, q_1, q_2, q_4, q_6$  and  $q_8$  of the NFA. You should proceed by following the algorithm for converting an NFA to a DFA. The final result is shown in Figure 6. What language does this DFA accept? Is it the same as that for the regular expression  $(ab + a)^*$ ?

## 2 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