

# NFA to DFA Conversion Algorithm

Jay Bagga

## 1 Introduction

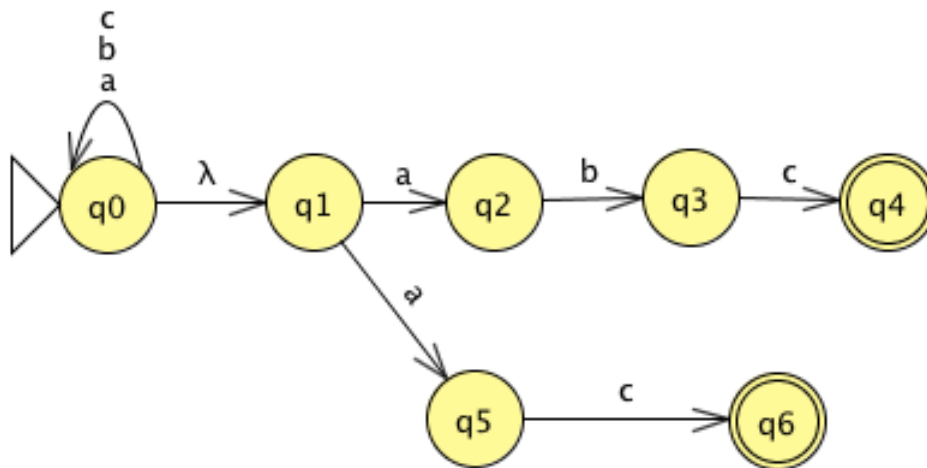
In the module on Nondeterministic Finite Automata (NFA) you learned that an NFA is a generalization of a DFA. However, it turns out that every NFA can be converted to an equivalent DFA. What this means is that if we have an NFA  $M_1$ , then we can construct a DFA  $M_2$  such that  $M_1$  and  $M_2$  accept the same set of words. In this module we describe an algorithm for this conversion. After studying this algorithm, you will be able to use JFLAP to run the algorithm on your own, or have JFLAP convert the NFA to an equivalent DFA.

The basic idea of the conversion algorithm is simple. Since the state transition function of an NFA maps a pair of state and alphabet symbol to a subset of states, we create new states in the DFA that represent these groups of states in the NFA. If the set of states  $Q$  of the NFA has  $n$  elements, then there are  $2^n$  subsets of  $Q$ . Thus the most number of states in our DFA to be constructed is  $2^n$ . In practice, some of the  $2^n$  states may be unreachable, so the actual number of states may be lower.

## 2 NFA to DFA Conversion Algorithm

We shall use the NFA shown below as an example to describe our conversion algorithm. Load the file NFAtoDFA-1.jflap to follow the guided steps of the simulation as described below.

Note that this is the same NFA that was described in the NFA module, except that the state names have been changed. Recall that this NFA accepts words that contain  $abc$  as a substring or that end in  $ac$  (or both).

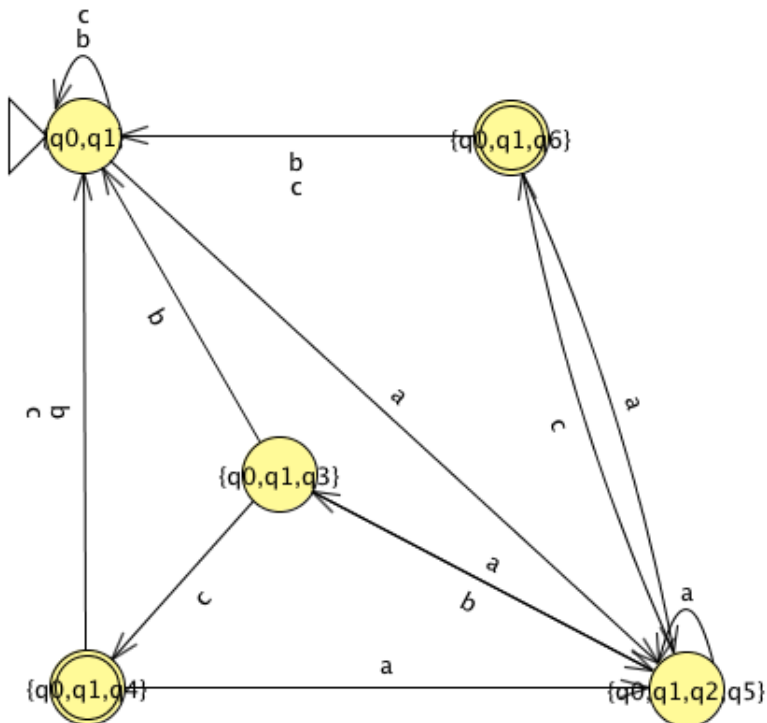


**Initial STEP:** The first step in construction of the DFA is to create the initial state of the DFA. For this initial state to have the same behavior as that of the NFA, look for all the states in the NFA that can be reached from the initial state without reading any input. Recall that in an NFA one can jump from one state to another with  $\lambda$ -transitions, without reading an input symbol. Hence we seek the group of states in the NFA that contains the initial state of the NFA and all those states that can be reached from this initial state by one or more  $\lambda$ -transitions.

For the NFA as shown above, there is only one  $\lambda$ -transition, from state  $q_0$  to state  $q_1$ . Hence the initial state of the DFA to be constructed is represented by the group of states  $\{q_0, q_1\}$ .

**Question 1.**

1. Load the NFA in the file NFAtoDFA-1.jflap.
2. Choose Convert in the menu and click on Convert to DFA. A new window opens with the NFA on the left. Notice that the DFA on the right already shows the initial state  $\{q_0, q_1\}$ . JFLAP does this step for you.
3. Before we see the next details of the conversion algorithm, let us have JFLAP build the DFA for us. Click on  on the right and the DFA is shown. Click on  and JFLAP responds with a message: The DFA is fully built. Click on  to place this DFA in a new window. Save this DFA as FinalDFA-1.jflap. Verify that this DFA is as shown below.



4. Test six different inputs on NFAtoDFA-1.jflap, three of which are accepted and three are rejected. Test the same inputs on FinalDFA-1.jflap and verify that you get the same results.
5. Open NFAtoDFA-1.jflap and FinalDFA-1.jflap in two windows. We will test their equivalence. In the window of NFAtoDFA-1.jflap choose Test from the menu and click on Compare Equivalence. Select FinalDFA-1.jflap. Verify that the message says: They ARE equivalent.

We now look at the main step of the conversion algorithm in more detail.

**Main STEP:** As mentioned in the Introduction, we find, for each state and alphabet symbol in the DFA, the next state which consists of the group of all states in the NFA that can be reached.

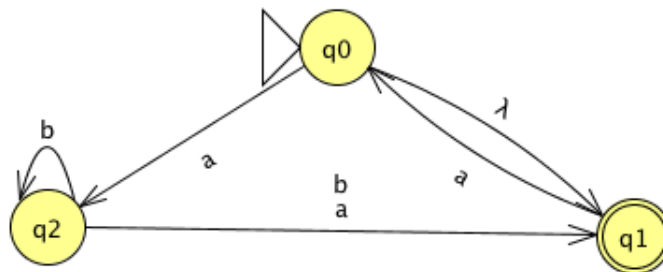
- Load the NFA in the file NFAtoDFA-1.jflap.
- Choose Convert in the menu and click on Convert to DFA. A new window opens with the NFA on the left. Notice that the DFA on the right already shows the initial state  $\{q_0, q_1\}$ . JFLAP does this step for you.
- Click on the “Expand Group on (T)erminal” button on the right side. We will expand from the initial state on symbol a. Click on the initial state on the right and drag to a blank area in that window. You are presented with a message “Expand on what terminal?”.
- Enter a and click OK. Another message asks: Which group of NFA states will that go on a? Look at the NFA on the left. What states can be reached from  $q_0$  or  $q_1$  with an a? From  $q_0$  we can reach  $q_0, q_2$  (by jumping to  $q_1$  and reading an a, and similarly  $q_5$ . From  $q_1$  we can reach  $q_2$  and  $q_5$ . Hence the group is  $\{q_0, q_1, q_2, q_5\}$ . Enter 0, 1, 2, 5 (or  $q_0, q_1, q_2, q_5$ ) and click OK. Verify that a new state with name  $\{q_0, q_1, q_2, q_5\}$  is drawn with a transition labeled a from the initial state.
- Now repeat the the above step at the initial state for the label b. With b, verify that the states that are reachable from  $q_0$  are  $q_0$  (with the b on the loop) and  $q_1$  with the b on loop and then a jump. No states can be reached from  $q_1$  on b. Hence this group is  $\{q_0, q_1\}$ , which is the initial state itself. Verify that a loop is drawn at the initial state with label b.
- Repeat the above step at the initial state for the label c, and verify that a loop is drawn at the initial state with label c. This completes all the labels (a, b, and c) out of the initial state.
- Repeat this for all the states until all states have all labels done. At any stage in this process, you can click on Done? to check if any states remain.
- What happens when you enter a wrong group of states?

- Alternatively, also experiment with the “State Expander” button. With this button selected, when you click on a state in the DFA, it is expanded immediately to include all labels from the clicked state. If you click on a state that is fully expanded, a message indicating this is displayed.
- Finally when the DFA is fully built, you can export it to a new window by clicking on the button . Try it!

Experiment with the steps above to gain practice.

### Question 2.

1. Load the NFA in the file NFAtoDFA-2.jflap as shown below.
2. Convert this NFA to a DFA by following the manual steps above, by expanding groups for each label.
3. Convert the NFA to DFA again by using the  button.
4. Verify that the DFAs you get in the above two steps are the same.
5. What language does this NFA accept? Why?



## 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