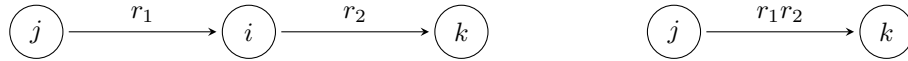


The following is an alternative algorithm for converting a non-deterministic finite automaton (acceptor) into a regular expression.

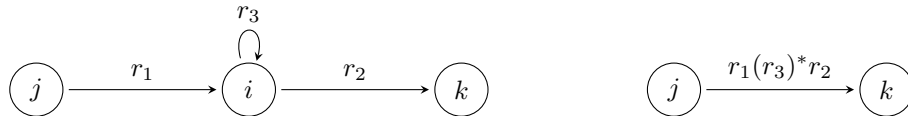
1. Convert the NFA into an NFA that has an initial state with no arrows coming into it. This can be done by creating a new initial state and  $\lambda$ -jumping to the old initial state.
2. Convert the NFA into an NFA that has a single final (favorable) state with no arrows coming out of it. This can be done by creating a new final state and  $\lambda$ -jumping from the old final states to it and then converting all of the old final states to non-final states.
3. Change the state names to numbers starting with the initial state as 1 and the final state as  $n$  where  $n$  is the number of states. That is, if there are 7 total states then the initial state will be labeled 1 and the final state will be labeled 7. The way that the states are labeled in between does not matter as long as they have different numbers between 2 and  $n - 1$ .
4. This portion of the algorithm will eliminate each state, 2 to  $n - 1$ , one by one, until the only states left are the initial state and the final state. In this algorithm, the arrows will contain regular expressions, not just alphabetic characters. When the algorithm is complete the single arrow from the initial state to the final state will have a label that is a regular expression associated with the automaton.

For  $i = 2, 3, 4, \dots, n - 1$ , do the following. For every pair of states  $j$  and  $k$ ,  $j$  may equal  $k$  but they are both different from  $i$ . If there is an arrow from  $j$  to  $i$  and from  $i$  to  $k$  add a new arrow from  $j$  to  $k$  with a label under the following conditions.

- (a) If there is no arrow from  $i$  to  $i$ , add an arrow from  $j$  to  $k$  with the concatenation of the labels of the arrows from  $j$  to  $i$  and from  $i$  to  $k$ .



- (b) If there is an arrow from  $i$  to  $i$ , add an arrow from  $j$  to  $k$  with the concatenation of the label of the arrow from  $j$  to  $i$ , the label of the arrow from  $i$  to  $i$  starred, and label of the arrow from  $i$  to  $k$ .



- (c) Once this is done for all pairs  $j$  and  $k$ , remove state  $i$  along with all arrows that are coming into it and all the arrows going out of it.
- (d) If there are multiple arrows from  $j$  to  $k$ , replace them with a single arrow with the label as the sum (union) of the labels.

