successfully terminates in the state. The language of the machine is the union of these sets. By Lemma 5.5.2, we can convert an arbitrary finite automaton to an equivalent NFA-λ with a single accepting set. To simplify the generation of a regular expression from a finite automaton, we will assume that the machine has only one accepting state.

The numbering of the states of the NFA-λ will be used in the node deletion algorithm to identify paths in the state diagram. The label of an arc from state q_i to state q_j is denoted $w_{i,j}$. If there is no arc from node q_i to q_j, $w_{i,j} = \emptyset$.

Algorithm 6.2.2
Construction of a Regular Expression from a Finite Automaton

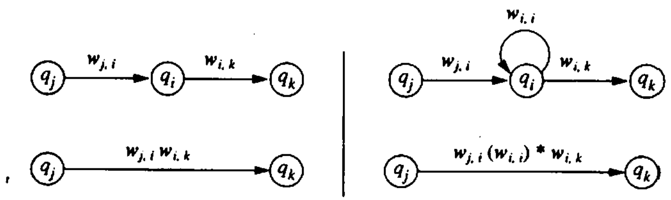
input: state diagram G of a finite automaton with one accepting state

Let q_0 be the start state and q_t the accepting state of G.

1. **repeat**
   1.1. choose a node q_i that is neither q_0 nor q_t
   1.2. delete the node q_i from G according to the following procedure:
      1.2.1 for every j, k not equal to i (this includes $j = k$) **do**
         i) **if** $w_{j,i} \neq \emptyset$, $w_{i,k} \neq \emptyset$ and $w_{i,i} = \emptyset$, **then** add an arc from node j to node k labeled $w_{j,i}w_{i,k}$
         ii) **if** $w_{j,i} \neq \emptyset$, $w_{i,k} \neq \emptyset$ and $w_{i,i} \neq \emptyset$, **then** add an arc from node q_j to node q_k labeled $w_{j,i}(w_{i,i})^*w_{i,k}$
         iii) **if** nodes q_j and q_k have arcs labeled $w_1, w_2, \ldots, w_s$ connecting them, **then** replace the arcs by a single arc labeled $w_1 \cup w_2 \cup \cdots \cup w_s$
      1.2.2 remove the node q_i and all arcs incident to it in G

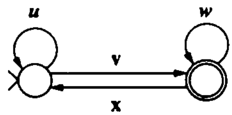
   until the only nodes in G are q_0 and q_t
2. determine the expression accepted by G

The deletion of node q_i is accomplished by finding all paths q_j, q_i, q_k of length two that have q_i as the intermediate node. An arc from q_j to q_k is added, bypassing the node q_i. If there is no arc from q_i to itself, the new arc is labeled by the concatenation of the expressions on each of the component arcs. If $w_{i,i} \neq \emptyset$, then the arc $w_{i,i}$ can be traversed any number of times before following the arc from q_i to q_k. The label for the new arc is $w_{j,i}(w_{i,i})^*w_{i,k}$. These graph transformations are illustrated as follows:



Step 2 in the algorithm may appear to be begging the question; the objective of the entire algorithm is to determine the expression accepted by G. After the node deletion process is completed, the regular expression can easily be obtained from the resulting graph. The reduced graph has at most two nodes, the start node and the accepting node. If these are the same node, the reduced graph has the form

accepting u^*. A graph with distinct start and accepting nodes reduces to

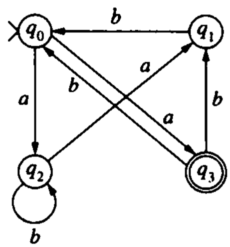


and accepts the expression $u^*v(w \cup xu^*v)^*$. This expression may be simplified if any of the arcs in the graph are labeled $\emptyset$.

Algorithm 6.2.2 can also be used to construct the language of a finite state machine with multiple accepting states. For each accepting state, we can produce an expression for the strings accepted by that state. The language of the machine is simply the union of the regular expressions obtained for each accepting state.

Example 6.2.1

The reduction technique of Algorithm 6.2.2 is used to generate a regular expression for the language of the NFA with state diagram



Deleting node q_1 yields