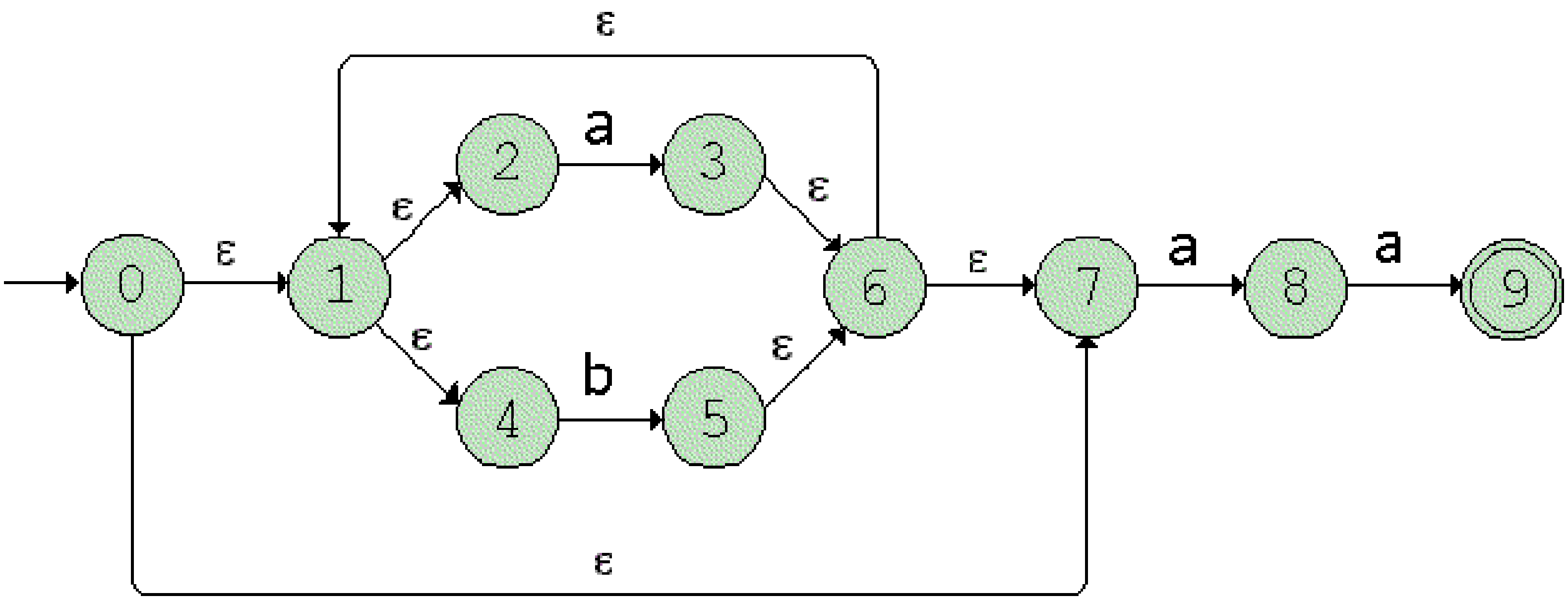
From NFA to DFA

Subset Construction

Converting an NFA to a DFA

- Let's convert the following NFA for the regular expression **(a|b)*aa** to a DFA.

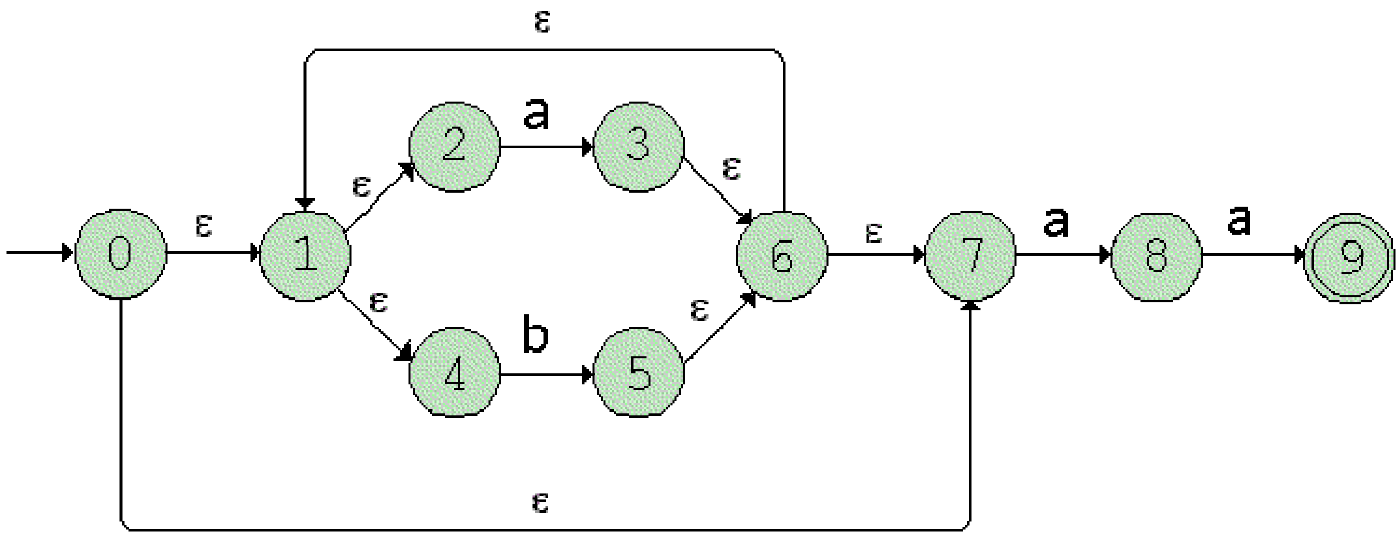


Subset Construction

- Subset construction is a technique for constructing a DFA from an NFA.
- Central to this algorithm is the calculation of the ε**-closure** of a state s.
- Simply put, the ε-closure of ***s*** is the set of all states reachable from ***s*** by following 0 or more ε-transitions.
- The ε-closure of a set of states is the union of the ε-closures of each state in the set.

Subset Construction

- We begin by calculating ε-closure(s_0), where s_0 is the start state of NFA N.
- In the NFA below, ε-closure(0) = {0,1,2,4,7}. This set is the start state for the DFA **D**.

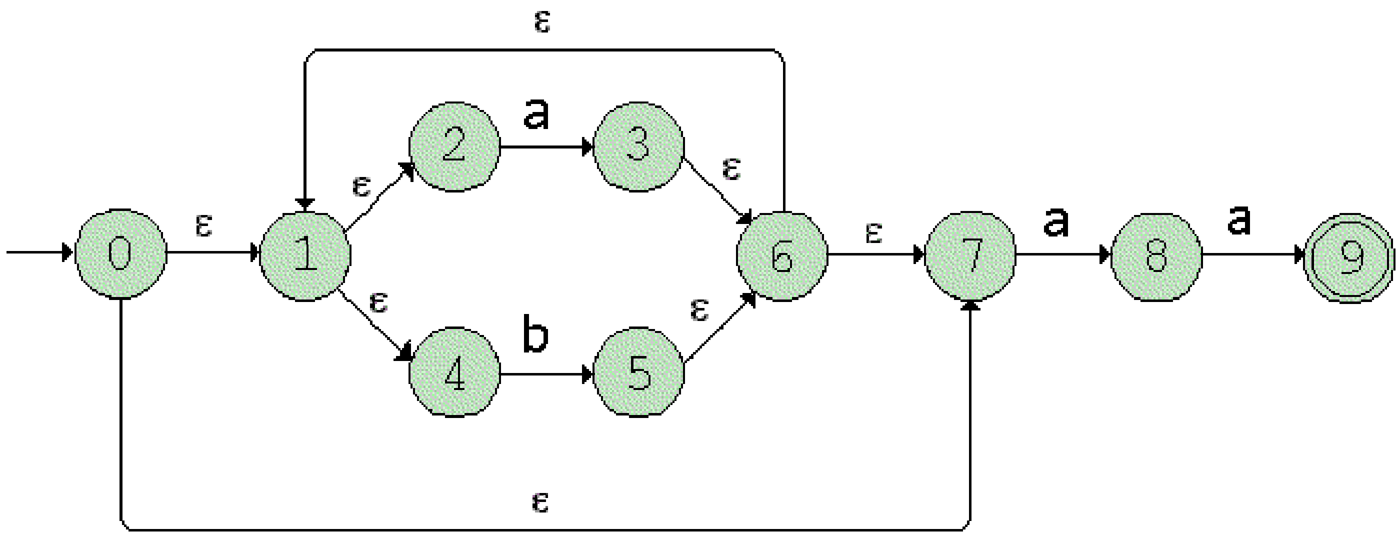


Subset Construction

- Another operation in subset construction is **move(S,a),** where ***S*** is a set of NFA states and ***a*** is an input character.
- **move(S,a)** is a set of NFA states to which there is a transition on input symbol ***a*** from some NFA state ***s*** in ***S***.

Subset Construction

- For our example below,
  - move({0,1,2,4,7},a) = {3,8}
  - move({0,1,2,4,7},b) = {5}



Subset Construction Algorithm

- Calculate ε-closure(s_0). Add this state to *DFAStates* (*DFAStates* is initially empty).
- While there is an unmarked state *T* in *DFAStates* do
  - mark *T*
  - for each $a \in \Sigma$ do
    - U = ε-closure(**move**(T,a))
    - if $U \notin$ *DFAStates* then

      add *U* as an unmarked state to *DFAStates*
    - add the transition $T \xrightarrow{a} U$ to DFA

Subset Construction Algorithm

- A DFA state which contains as a member the start state of the NFA will be the start state of the DFA.
- Any DFA state which contains as a member an accepting state of the NFA will be an accepting state of the DFA.

Applying the Algorithm 1

- First, calculate ε-closure(0) = {0,1,2,4,7}. Let's call this set **A**.
  - move(A, a) = {3,8}
  - ε-closure({3,8}) = {1,2,3,4,6,7,8} = **B**
  - Add $A \xrightarrow{a} B$ to the DFA

  - move(A, b) = {5}
  - ε-closure({5}) = {1,2,4,5,6,7} = **C**
  - Add $A \xrightarrow{b} C$ to the DFA

Applying the Algorithm 2

- Next, let's consider **B** = {1,2,3,4,6,7,8}
  - move(B, a) = {3,8,9}
  - ε-closure({3,8,9}) = {1,2,3,4,6,7,8,9} = **D*** (an accepting state)
  - Add $B \xrightarrow{a} D$ to the DFA

  - move(B, b) = {5}
  - ε-closure({5}) = {1,2,4,5,6,7} = **C**
  - Add $B \xrightarrow{b} C$ to the DFA

Applying the Algorithm

- Now, let's consider **C** = {1,2,4,5,6,7}
  - move(C, a) = {3,8}
  - ε-closure({3,8}) = **B**
  - Add $C \xrightarrow{a} B$ to the DFA

  - move(C, b) = {5}
  - ε-closure({5}) = **C**
  - Add $C \xrightarrow{b} C$ to the DFA

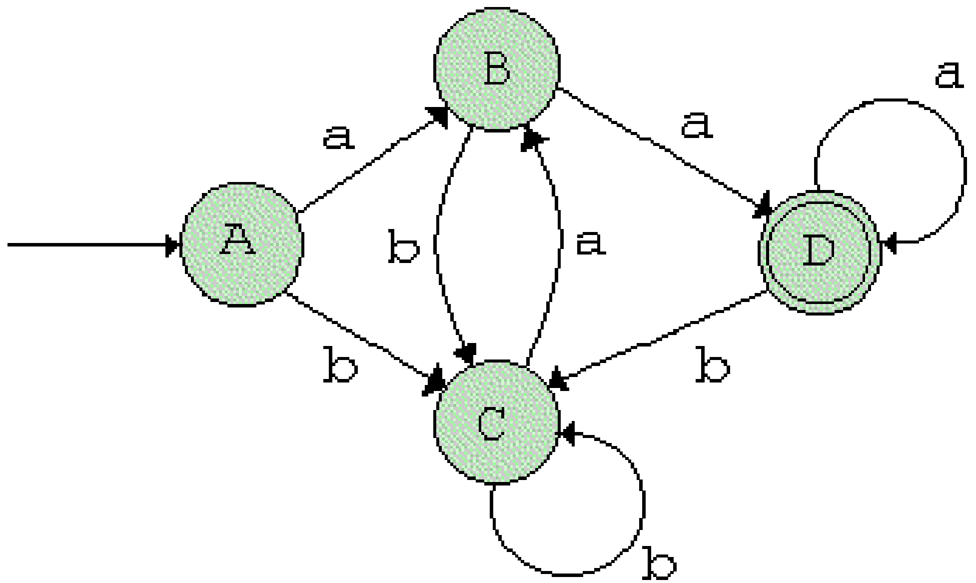
Applying the Algorithm 4

- Finally, let's consider **D** = {1,2,3,4,6,7,8,9}
  - move(D, a) = {3,8,9}
  - ε-closure({3,8,9}) = **D**
  - Add $\mathrm{D} \xrightarrow{\mathrm{a}} \mathrm{D}$ to the DFA

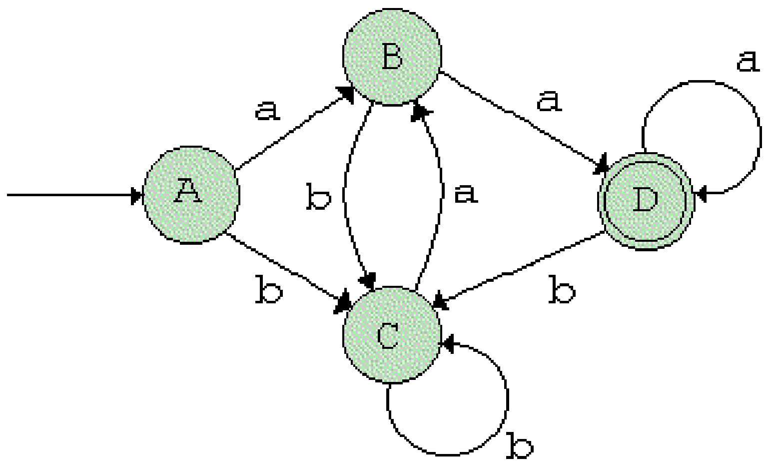
  - move(D, b) = {5}
  - ε-closure({5}) = **C**
  - Add $\mathrm{D} \xrightarrow{\mathrm{b}} \mathrm{C}$ to the DFA

Applying the Algorithm 5

- Since there are no remaining unmarked DFA states, the process terminates. Here is the resulting DFA:



The DFA as a Table



	a	b	
A	B	C	start
B	D	C	
C	B	C	
D	D	C	accept

- The DFA can be represented as a table