(*Subset construction*) Constructing a DFA from an NFA.[1]

Input. An NFA *N*.

Output. A DFA *D* accepting the same languange.

Method. Our algorithm constructs a transition table *Dtran* for *D*. Each DFA state is a set of NFA states and we construct *Dtran* so that *D* will simulate "in parallel" all possible moves *N* can make on a given input string.

We use the operations in Table 1 to keep track of sets of NFA states (*s* represents an NFA state and *T* a set of NFA states).

Operations	**Description**
Î -*closure*(*s*)	Set of NFA states reachable from NFA state *s* on Î -transitions alone.
Î -*closure*(*T*)	Set of NFA states reachable from some NFA state *s* in *T* on Î -transitions alone.
move(*T*,*a*)	Set of NFA states to which there is a transition on input symbol *a* from some NFA state *s* in *T*.

Table 1. Operations on NFA states

Before it sees the first input symbol, *N* can be in any of the states in the set $\in -closure(s_0)$, where s_0 is the start state of *N*. Suppose that exactly the states in set *T* are reachable from s_0 on a given sequence of input symbols, and let a, be the next input symbol. On seeing a, *N* can move to any of the states in the set $move(T,a)$. When we allow for $\in$-transitions, *N* can be in any of the states in $\in -closure(move(T,a))$, after seeing the a.

initially $\in -closure(s_0)$ is the only state in $Dstates$ and it is unmarked;
while there is an unmarked state T in $Dstates$ **do**
  mark T ;
  for each input symbol a **do**
    $U := \in -closure(move(T,a));$
    if U is not in $Dstates$ **then**
      add U as an unmarked state to $Dstates$;
    $Dtran[T,a] := U$
  end for
end while

[1] Excepted from Aho, Sethi, and Ullman *Compilers, principles, techniques, and tools*. Addison-Wesley, 1986, ISBN 0-201-10088-6

Example: Figure 1 shows an NFA *N* accepting the language $(a \mid b)^*abb$.

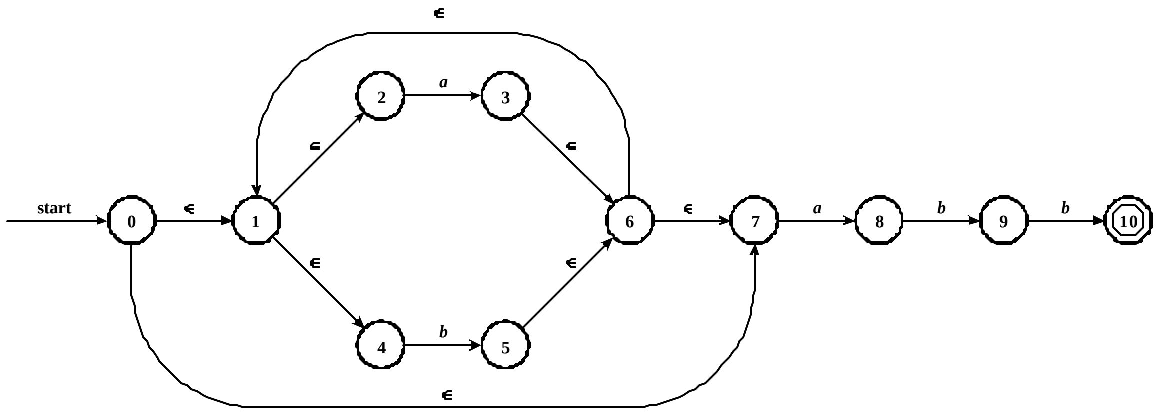

Figure 1. NFA *N* accepting the language $(a \mid b)^*abb$

$A = \in -closure(0) = \{0,1,2,4,7\}$

STATE	INPUT SYMBOL	
	a	*b*
A		

$B = \in -closure(move(A, a))$

$B = \in -closure(move(\{0,1,2,4,7\}, a))$

$B = \in -closure(\{3,8\})$

$B = \{1,2,3,4,6,7,8\}$

$Dtran[A, a] = B$

STATE	INPUT SYMBOL	
	a	*b*
A	*B*	

$C = \in -closure(move(A, b))$

$C = \in -closure(move(\{0,1,2,4,7\}, b))$

$C = \in -closure(\{5\})$

$C = \{1,2,4,5,6,7\}$

$Dtran[A, b] = C$

STATE	INPUT SYMBOL	
	a	*b*
A	*B*	*C*

$D = \in -closure(move(B, a))$

$D = \in -closure(move(\{1,2,3,4,6,7,8\}, a))$

$D = \in -closure(\{3,8\})$

$D = B$

$Dtran[B, a] = B$

STATE	INPUT SYMBOL	
	a	*b*
A	*B*	*C*
B	*B*	

$D = \in -closure(move(B, b))$
$D = \in -closure(move(\{1,2,3,4,6,7,8\}, b))$
$D = \in -closure(\{5,9\})$
$D = \{1,2,4,5,6,7,9\}$
$Dtran[B, b] = D$

STATE	**INPUT SYMBOL**	
	a	*b*
A	*B*	*C*
B	*B*	*D*

$E = \in -closure(move(C, a))$
$E = \in -closure(move(\{1,2,3,4,5,6,7\}, a))$
$E = \in -closure(\{3,8\})$
$E = B$
$Dtran[C, a] = B$

STATE	**INPUT SYMBOL**	
	a	*b*
A	*B*	*C*
B	*B*	*D*
C	*B*	

$E = \in -closure(move(C, b))$
$E = \in -closure(move(\{1,2,4,5,6,7\}, b))$
$E = \in -closure(5)$
$E = C$
$Dtran[C, b] = C$

STATE	**INPUT SYMBOL**	
	a	*b*
A	*B*	*C*
B	*B*	*D*
C	*B*	*C*

$E = \in -closure(move(D, a))$
$E = \in -closure(move(\{1,2,4,5,6,7,9\}, a))$
$E = \in -closure(\{3,8\})$
$E = B$
$Dtran[D, a] = B$

STATE	**INPUT SYMBOL**	
	a	*b*
A	*B*	*C*
B	*B*	*D*
C	*B*	*C*
D	*B*	

$E = \in -closure(move(D, b))$
$E = \in -closure(move(\{1,2,4,5,6,7,9\}, b))$
$E = \in -closure(\{5,10\})$
$E = \{1,2,4,5,6,7,10\}$
$Dtran[D, b] = E$

STATE	**INPUT SYMBOL**	
	a	*b*
A	*B*	*C*
B	*B*	*D*
C	*B*	*C*
D	*B*	*E*

$F = \in -closure(move(E, a))$
$F = \in -closure(move(\{1,2,4,5,6,7,10\}, a))$
$F = \in -closure(\{3,8\})$
$F = B$
$Dtran[E, a] = B$

STATE	**INPUT SYMBOL**	
	a	*b*
A	*B*	*C*
B	*B*	*D*
C	*B*	*C*
D	*B*	*E*
E	*B*	

$F = \in -closure(move(E, b))$
$F = \in -closure(move(\{1,2,4,5,6,7,10\}, b))$
$F = \in -closure(\{5\})$
$F = C$
$Dtran[E, b] = C$

STATE	**INPUT SYMBOL**	
	a	*b*
A	*B*	*C*
B	*B*	*D*
C	*B*	*C*
D	*B*	*E*
E	*B*	*C*

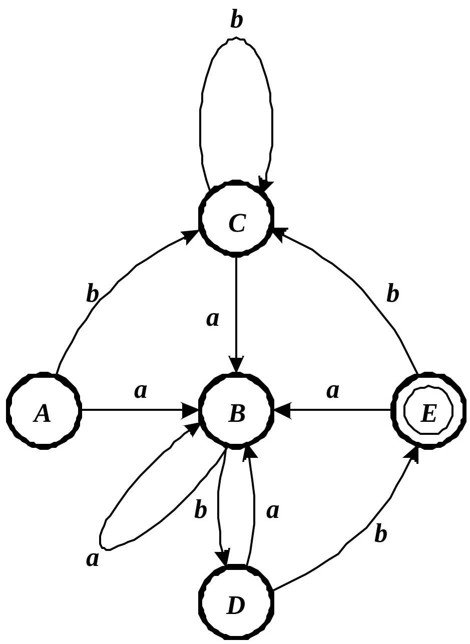

Figure 2. DFA accepting $(a \mid b)^* abb$.