

Lecture 8

NFA ? DFA Construction

The algorithm is called *subset construction*. In the transition table of an NFA, each entry is a set of states. In DFA, each entry is a single state. The general idea behind NFA-to-DFA construction is that each DFA state corresponds to a set of NFA states. The DFA uses its state to keep track of all possible states the NFA can be in after reading each input symbol.

We will use the following operations.

- *e-closure(T)*:
set of NFA states reachable from some NFA state s in T on ϵ -transitions alone.
- *move(T,a)*:
set of NFA states to which there is a transition on input a from some NFA state s in set of states T .

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

Subset Construction

Algorithm:

Input: NFA N with state set S , alphabet Σ , start state s_0 , final states F

Output: DFA D with state set S' , alphabet Σ , start states,
 $s_0' = \epsilon$ -closure(s_0), final states F' and transition table: $S' \times \Sigma \rightarrow S'$

// initially, ϵ -closure(s_0) is the only state in D states S' and it is unmarked

$s_0' = \epsilon$ -closure(s_0)

$S' = \{s_0'\}$ (unmarked)

while (there is some unmarked state T in S')

 mark state T

for all a in Σ **do**

$U = \epsilon$ closure($move(T,a)$);

if U not already in S'

 add U as an unmarked state to S'

$Dtran(T,a) = U$;

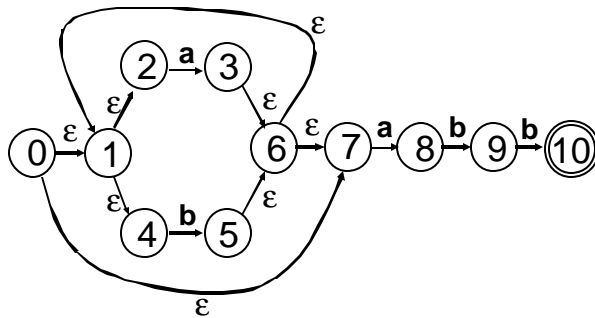
end for

end while

F' :
 for each DFA state S
 if S contains an NFA final state
 mark S as DFA final state
end algorithm

Example

Let us apply the algorithm to the NFA for $(a | b)^*abb$. Σ is $\{a, b\}$.



The start state of equivalent DFA is ϵ -closure(0), which is $A = \{0,1,2,4,7\}$; these are exactly the states reachable from state 0 via ϵ -transition. The algorithm tells us to mark A and then compute ϵ -closure(move(A, a))

move(A, a), is the set of states of NFA that have transition on 'a' from members of A . Only 2 and 7 have such transition, to 3 and 8. So, ϵ -closure(move(A, a)) = ϵ -closure($\{3,8\}$) = $\{1,2,3,4,6,7,8\}$. Let $B = \{1,2,3,4,6,7,8\}$; thus $Dtran[A, a] = B$

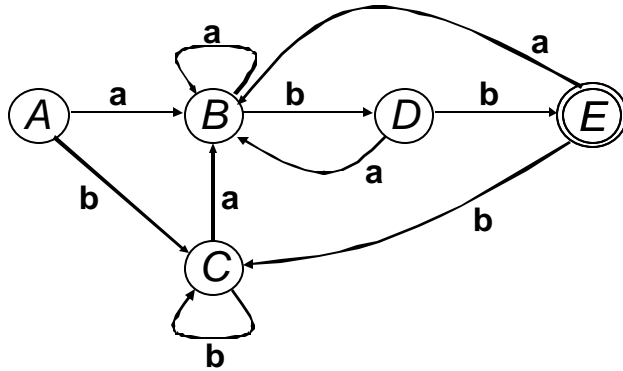
For input b , among states in A , only 4 has transition on b to 5. Let $C = \epsilon$ -closure($\{5\}$) = $\{1,2,4,5,6,7\}$. Thus, $Dtran[A, b] = C$

We continue this process with the unmarked sets B and C , i.e., ϵ -closure(move(B, a)), ϵ -closure(move(B, b)), ϵ -closure(move(C, a)) and ϵ -closure(move(C, b)) until all sets and states of DFA are marked. This is certain since there are *only* 2^{11} (!) different subsets of a set of 11 states. A set, once marked, is marked forever. Eventually, the 5 sets are:

1. $A = \{0,1,2,4,7\}$
2. $B = \{1,2,3,4,6,7,8\}$
3. $C = \{1,2,4,5,6,7\}$
4. $D = \{1,2,4,5,6,7,9\}$
5. $E = \{1,2,4,5,6,7,10\}$

A is start state because it contains state 0 and E is the accepting state because it contains state 10.

The subset construction finally yields the following DFA



The Resulting DFA can be encoded as the following transition table

State	Input symbol	
	a	b
A	B	C
B	B	D
C	B	C
D	B	E
E	B	C