

---

## Discussion Section Notes – Commented Solution

### Discussion Section Notes

The problem we discussed on Friday Jan. 10, 2003 was the following:

Consider the language  $L$  of all strings over  $\{0,1\}$  containing 101 or 11 as substrings.

1. Give a NFA that recognizes this language. Include a pictorial representation of the NFA and a formal description (showing explicitly its components, namely  $Q, \Sigma, \delta, q_0$  and  $F$ )

**Solution:** First of all, remember that saying that “an automaton  $M$  recognizes  $L$ ” means two things: (1) that  $M$  on input a string from  $L$  always finishes its computation in an accepting state, and (2) that  $L$  does not recognize strings that don’t belong to  $L$  (therefore, you must assure it does not recognize *more* strings than what it should; this is a common mistake).

Figure 1 shows an NFA that recognizes language  $L$ . Let’s call it  $M$ .

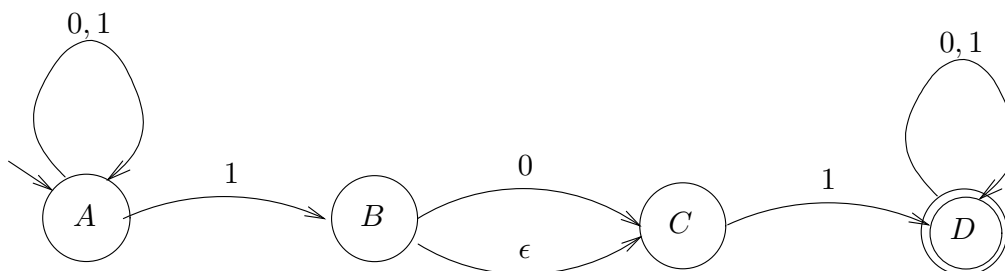


Figure 1: State diagram of the NFA that recognizes  $L$

Formally,  $M$  is the 5-tuple  $(Q, \Sigma, \delta, q_0, F)$  where

- the set of states  $Q$  is  $\{A, B, C, D\}$
- $\Sigma = \{0, 1\}$
- $q_0 = \{A\}$
- $F = \{D\}$

The transition function  $\delta$  is shown in the table below.

$\delta$	A	B	C	D
0	{ A }	{ C }	$\emptyset$	{ D }
1	{ A, B }	$\emptyset$	{ D }	{ D }
$\epsilon$	$\emptyset$	{ C }	$\emptyset$	$\emptyset$

We now proceed to transform  $M$  into a DFA  $M' = (Q', \Sigma', \delta', q'_0, F')$ . In order to fully specify  $M'$  we need to give explicit values for  $Q'$ ,  $\Sigma'$ ,  $\delta'$ ,  $q'_0$  and  $F'$  in terms of  $M$ , as follows:

- $Q'$  is the power set of  $Q$ , that is  $Q' = \mathcal{P}(\{A, B, C, D\}) = \{\emptyset, \{A\}, \{B\}, \{C\}, \{D\}, \{A, B\}, \dots, \{A, B, C, D\}\}$ .
- $\Sigma' = \Sigma = \{0, 1\}$
- $q'_0 = E(q_0) = E(\{A\}) = \{\text{all states reachable from } \{A\} \text{ by taking } 0, 1 \text{ or more } \epsilon\text{-transitions.}\}$
- $F' = \{Q \cap F \neq \emptyset\} = \{\{D\}, \{A, D\}, \{B, D\}, \dots, \{A, B, C, D\}\}$

The hardest part by far is building the transition function  $\delta'$ . Remember that the transformation from NFAs to DFAs says that, for each element  $S \in Q'$  and each input  $a \in \Sigma'$ , the value of  $\delta'(S, a)$  is given by

$$\delta'(S, a) = E\left(\bigcup_{t \in S} \delta(t, a)\right).$$

where function  $E(S)$  (called the  $\epsilon$ -closure of set  $S$ ) is defined as the set of all elements (states) in  $Q$  that can be reached from states in  $S$  by taking 0, 1 or more  $\epsilon$ -transitions. (Note that if for a state  $q$  there is no  $\epsilon$ -transition leaving  $q$ , then  $E(\{q\}) = \{q\}$ ).

Let's compute the transition function  $\delta'$ . There are two ways to do it.

One possibility (the “brute force approach”) is to evaluate the above formula on *each possible* of  $Q'$  and each possible input: that is, using the formula we could compute  $\delta'(S, a)$  for each  $S \in Q' = \{\emptyset, \{A\}, \{B\}, \dots, \{A, B, C, D\}\}$ , and each  $a \in \Sigma'$

$$\begin{aligned} \delta'(\emptyset, 0) &= E(\emptyset) = \emptyset \\ \delta'(\emptyset, 1) &= E(\emptyset) = \emptyset \\ \delta'(\{A\}, 0) &= E(\delta(A, 0)) = \dots \\ \delta'(\{A\}, 1) &= \dots \\ \delta'(\{B\}, 0) &= \dots \\ \delta'(\{B\}, 1) &= \dots \\ &\vdots \\ \delta'(\{A, B, C, D\}, 0) &= \dots \\ \delta'(\{A, B, C, D\}, 1) &= \dots \end{aligned}$$

but this is not recommended, since we may end up computing  $\delta'$  for many *useless* values  $S$  and therefore we may waste time if the set  $Q'$  is big. (More explanations about why this is usually wasteful is given later.)

The second (and recommended) approach is to start computing  $\delta'$  from the starting state  $q'_0$ . In our case  $q'_0$  equals  $\{A\}$  and then,

$$\begin{aligned}\delta'(\{A\}, 0) &= E(\delta(A, 0)) = E(\{A\}) = \{A\} \\ \delta'(\{A\}, 1) &= E(\delta(A, 1)) = E(\{A, B\}) = \{A, B, C\}\end{aligned}$$

(why is  $E(\{A, B\}) = \{A, B, C\}$ ? because from  $B$  we can reach  $C$  by taking one  $\epsilon$  transition).

And now? in which value do we compute  $\delta'$ ? Simple, we just continue with the *new* values of  $\delta'$  we just obtained. **In other words, if we compute  $\delta'(S, a) = T$  and we have not computed  $\delta'(T, a)$ , we continue with set  $T$ .** In our case, the above calculations yielded sets  $\{A\}$  and  $\{A, B, C\}$ . Since the value for  $\{A\}$  is already computed, we compute the value of  $\delta'$  on  $\{A, B, C\}$ :

$$\begin{aligned}\delta'(\{A, B, C\}, 0) &= E(\delta(A, 0) \cup \delta(B, 0) \cup \delta(C, 0)) = E(\{A\} \cup \{C\} \cup \emptyset) \\ &= E(\{A, C\}) = \{A, C\} \\ \delta'(\{A, B, C\}, 1) &= E(\delta(A, 1) \cup \delta(B, 1) \cup \delta(C, 1)) = E(\{A, B\} \cup \emptyset \cup \{D\}) \\ &= E(\{A, B, D\}) = \{A, B, C, D\}\end{aligned}$$

which new sets (states) did we get?  $\{A, C\}$  and  $\{A, B, C, D\}$ . We then continue using those.

The rest of the values of  $\delta'$  are computed following the same strategy:

$$\begin{aligned}\delta'(\{A, C\}, 0) &= E(\delta(A, 0) \cup \delta(C, 0)) = E(\{A\} \cup \emptyset) \\ &= E(\{A\}) = \{A\} \\ \delta'(\{A, C\}, 1) &= E(\delta(A, 1) \cup \delta(C, 1)) = E(\{A, B\} \cup \{D\}) \\ &= E(\{A, B, D\}) = \{A, B, C, D\} \\ \delta'(\{A, B, C, D\}, 0) &= E(\delta(A, 0) \cup \delta(B, 0) \cup \delta(C, 0) \cup \delta(D, 0)) = E(\{A\} \cup \{C\} \cup \emptyset \cup \{D\}) \\ &= E(\{A, C, D\}) = \{A, C, D\} \\ \delta'(\{A, B, C, D\}, 1) &= E(\delta(A, 1) \cup \delta(B, 1) \cup \delta(C, 1) \cup \delta(D, 1)) = E(\{A, B\} \cup \emptyset \cup \{D\} \cup \{D\}) \\ &= E(\{A, B, D\}) = \{A, B, C, D\} \\ \delta'(\{A, C, D\}, 0) &= E(\delta(A, 0) \cup \delta(C, 0) \cup \delta(D, 0)) = E(\{A\} \cup \emptyset \cup \{D\}) \\ &= E(\{A, D\}) = \{A, D\} \\ \delta'(\{A, C, D\}, 1) &= E(\delta(A, 1) \cup \delta(C, 1) \cup \delta(D, 1)) = E(\{A, B\} \cup \{D\} \cup \{D\}) \\ &= E(\{A, B, D\}) = \{A, B, C, D\} \\ \delta'(\{A, D\}, 0) &= E(\delta(A, 0) \cup \delta(D, 0)) = E(\{A\} \cup \{D\}) \\ &= E(\{A, D\}) = \{A, D\} \\ \delta'(\{A, D\}, 1) &= E(\delta(A, 1) \cup \delta(D, 1)) = E(\{A, B\} \cup \{D\}) \\ &= E(\{A, B, D\}) = \{A, B, C, D\}\end{aligned}$$

How do we know when to finish? when we have computed the value of  $\delta'$  on all the sets we have obtained starting from  $q'_0$ .

The resulting transition function  $\delta'$  is shown in the table below (the table is transposed due to space constraints).

$\delta$	0	1
$\{A\}$	$\{A\}$	$\{A, B, C\}$
$\{A, C\}$	$\{A\}$	$\{A, B, C, D\}$
$\{A, D\}$	$\{A, D\}$	$\{A, B, C, D\}$
$\{A, B, C\}$	$\{A, C\}$	$\{A, B, C, D\}$
$\{A, C, D\}$	$\{A, B, C, D\}$	$\{A, D\}$
$\{A, B, C, D\}$	$\{A, C, D\}$	$\{A, B, C, D\}$

A pictorial description of the resulting DFA  $M'$  is shown in Figure 2.

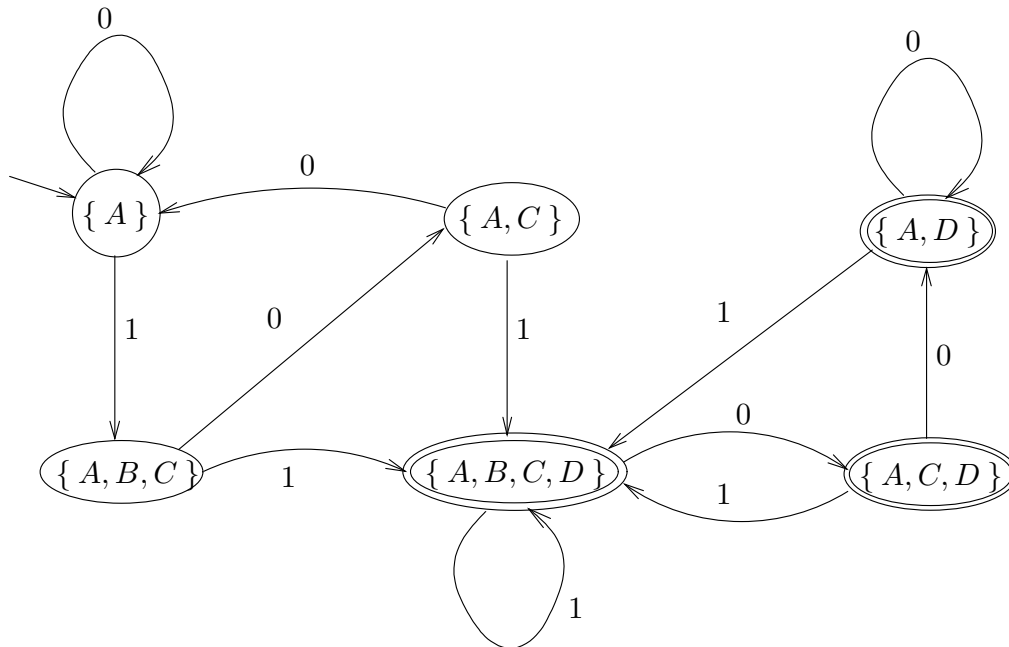


Figure 2: Resulting DFA  $M'$  equivalent to NFA  $M'$

In this point, we could say we are done. However, a final issue is: why did we neglect (discard) the values of the transition function  $\delta'$  on other states (like  $\emptyset$ ,  $\{B\}$ ,  $\{C\}$ ,  $\{D\}$ ,  $\{B, D\}$ ,  $\{B, C, D\}$ , and others) ?

The answer can be easily seen by computing some of the “missing” values:

$$\delta'(\emptyset, 0) = E(\emptyset) = \emptyset$$

$$\delta'(\emptyset, 1) = E(\emptyset) = \emptyset$$

$$\delta'(\{B\}, 0) = E(\delta(B, 0)) = E(\{C\}) = E(\{C\}) = \{C\}$$

$$\begin{aligned} \delta'(\{B\}, 1) &= E(\delta(B, 1)) = E(\emptyset) = E(\emptyset) = \emptyset \\ \delta'(\{C\}, 0) &= E(\delta(C, 0)) = E(\emptyset) = E(\emptyset) = \emptyset \\ \delta'(\{C\}, 1) &= E(\delta(C, 1)) = E(\{D\}) = E(\{D\}) = \{D\} \\ \delta'(\{D\}, 0) &= E(\delta(D, 0)) = E(\{D\}) = E(\{D\}) = \{D\} \\ \delta'(\{D\}, 1) &= E(\delta(D, 1)) = E(\{D\}) = E(\{D\}) = \{D\} \end{aligned}$$

these states and transitions are depicted in Figure 3.

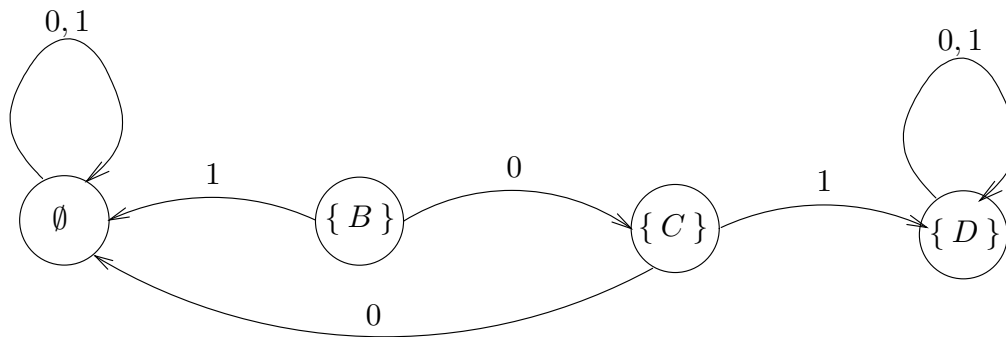


Figure 3: Unreachable states

Why can this part of the DFA be discarded from DFA  $M'$ ? **Simply because this part of the DFA (comprising states  $\{B\}$ ,  $\{C\}$ , and  $\{D\}$ ) cannot be reached from the start state  $q'_0 = \{A\}$ .** Hence, those states will never be used, since no input can make the DFA go into one of those states, and they can be “cut off” from the DFA without loss.

It is easy to verify that states  $\{A, B\}$ ,  $\{B, C\}$ ,  $\{B, D\}$ ,  $\{B, C, D\}$ ,  $\{C, D\}$ ,  $\{B, C, D\}$  are also not reachable from the start state, and can also be discarded.

This concludes the solution.

---