

Automata Theory

Homework 1: due 14 October 2008

1. Constructing a DFA from an NFA

- Input: An NFA $N = (Q_N, \Sigma, \Delta_N, q_0, F_N)$. The NFA is represented by a transition table (hw1.in).

state	final	input symbol		
		0	1	ϵ
0	0	0,1	0	-
1	0	-	2	-
2	0	-	3	3
3	1	-	-	-

A set of states in the table should be implemented by a pointer to the set of states.

- Output: a DFA $D = (Q_D, \Sigma, \delta_D, E(q_0), F_D)$ equivalent to N . The DFA is also represented by a transition table.

2. Running an NFA

- Input: input string x and an NFA $N = (Q, \Sigma, \Delta, q_0, F)$
- Output: yes if N accepts x ; no otherwise

```
Run( $N, x$ )
   $C \leftarrow E(q_0)$ 
  for  $i \leftarrow 1$  to  $n$  do
     $C' \leftarrow \Delta(C, x_i)$ 
     $C \leftarrow E(C')$ 
  od
  if  $C$  contains a final state
    then print "yes"
  else print "no" fi
end
```

3. Implementation

Given an NFA (table), your program should construct an equivalent DFA (print the DFA table) and run the NFA and the DFA on various inputs.

- Assume that $\Sigma = \{0, 1\}$.
- Run your program with at least two NFAs. For each NFA, run your NFA and DFA with at least two "yes" strings and at least two "no" strings.

- Hand in your programs, executable files, and an example running by email to `tyjeong@theory.snu.ac.kr`.
- Write down the environment you run your program.
- Write comments appropriately in your program.