DEPARTMENT OF COMPUTER SCIENCE
CARLOS III UNIVERSITY OF MADRID

Computer Science
Language Processors

Rules

- The duration of the test is **30 minutes**
- Questions will not be answered during the test
- One cannot re-enter the classroom after leaving it
- The answers must be written using a pen (not a pencil)

1.- Describe the language denoted by the following regular expressions. The alphabet ∑ is {x, y}.

a) x(x|y)*y
Strings must start with x and end in y.

b) ((x|y)(x|y))$^+$
String must be of even length >= 2.

c) x*(yx+)*x*
Every y is followed by at least one x (can't contain substring yy & can't end with y).

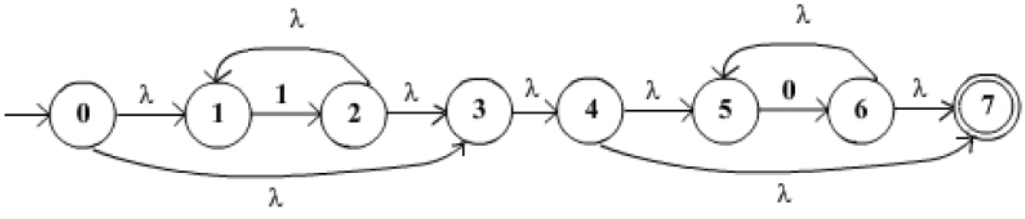
d) (x|y)*(xx|yy)*y*
Any string (i.e. regular expression matches 2*).

2.- Contruct a DFA given the language of all strings which do not contain the substring 01.

1*0* (0 can only be followed by another 0)

Given the regular expression, we construct the DFA following the same steps explained in Unit 2.

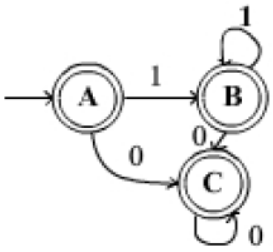
a) The associated NFA using Thompson's construction is:



David Griol Barres, Antonio Berlanga de Jesús, Jesús García Herrero, Juan Manuel Alonso Weber

b) Then, we apply the algorithm to construct the associated DFA:

ʎ-closure(0) = {0,1,3,4, 5,7} = A
ʎ-closure(Move(A,1)) = ʎ-closure({2}) = {1,2,3,4,5,7} = B
Transition(A,1) = B
ʎ-closure(Move(A,0)) = ʎ-closure({6}) = {5,6,7} = C
Transition(A,0) = C
ʎ-closure(Move(B,1)) = ʎ-closure({2}) = {1,2,3,4,5,7} = B
Transition(B,1) = B
ʎ-closure(Move(B,0)) = ʎ-closure({6}) = C
Transition(B,0) = C
ʎ-closure(Move(C,0)) = ʎ-closure({6}) = C
Transition(C,0) = C



c) Lastly, we apply the algorithm to minimize the number of states of the DFA:

It can be seen that states A and B are not distinguished using both 0 or 1 transitions