CMSC330 Fall 2009 Quiz #1 Solution & Grading Key

1. (4 pts) What is the output (if any) of the following Ruby program? Write FAIL if code does not execute.

```
a = [1,2,3 ]
a[4] = 5
a.each{ |x| puts "#{x}" }
```

Answer:

```
1
2
3          # 2 points for sequence 1 2 3 5
           # 2 points for gap (or nil) between 3& 5
5
```

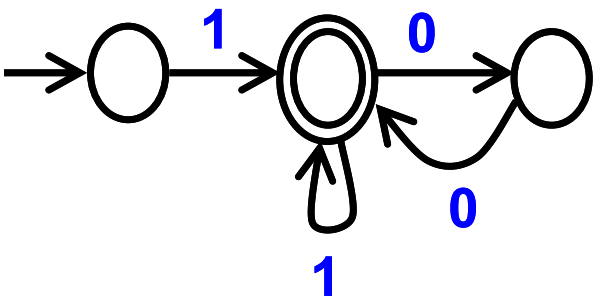
2. (4 pts) Is it true that every DFA is a NFA? Explain your answer.

Yes, since a DFA is simply a NFA without e-transitions or multiple edges from a state with the same label.

 # 2 points for answering yes
 # 2 points for understanding DFA is a simplified NFA

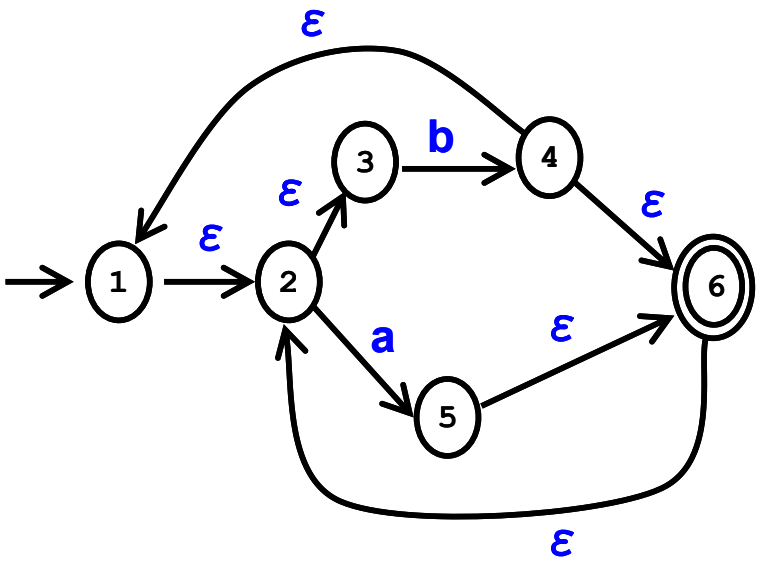
3. (4 pts) Write a DFA that accepts the language 1(00|1)*. You do not need to use the algorithms described in class.

Many possible DFAs. For instance the following DFA works.



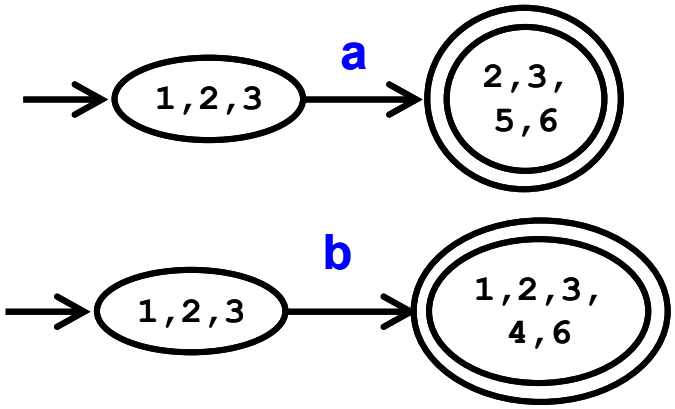
4 pts for correct answer
3 pts for 75% of the answer
(e.g., 1 minor mistake, like not accepting 1)
2 pts for 50% of the answer
(e.g., 1 major mistake or 2-3 minor ones, like not having a back edge)
1 pts for 25% of the answer
(e.g., far off, but there's some insight there, like it accepts 1 or 100)

4. (8 pts) Begin converting the following NFA into a DFA by applying the subset construction algorithm discussed in class. You do not need to produce the entire DFA. Create at least 2 DFA states (starting state & additional state) and the transition (with label) connecting them. Be sure to list the NFA states represented by each DFA state.



Answer

Either of the two following DFA is correct. May contain additional transitions.



 # 8 points for correct answer

 # partial credit
 # 2 points for starting state of DFA corresponding to 1,2,3
 # 2 points for 2nd state of DFA corresponding to 2,3,5,6 (for -> a)
 # OR for 2nd state of DFA corresponding to 1,2,3,4,6 (for -> b)
 # 2 points for correctly labeled transition
 # 2 points for realizing state containing 6 is final state