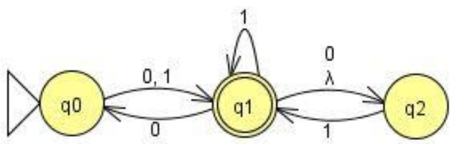
Homework 4 CS3110

Section 2.3 On page 64, #2, #4, #14

<u>Section 2.3</u>

#2) (3pts) Convert the nfa in exercise 13, section 2.2, into an equivalent dfa.

From exercise 13, 2.2:

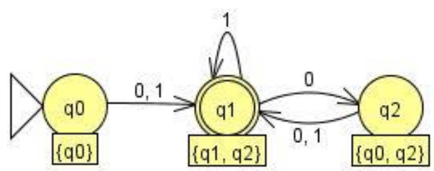


Answer:

NFA δ * table:

	0	1
q0	{q1, q2}	{q1, q2}
q1	{q0, q2}	{q1, q2}
q2	{}	{q1, q2}

dfa:



As long as the final DFA graph is correct (accepting the same language), then they can get full credits. If some states or transitions are wrong. You can give them partial points based on the correct parts of DFA graph and the table.

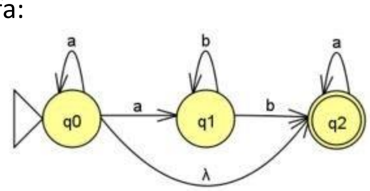
#4) (4 pts) Convert the nfa defined by

$\delta(q0, a) = \{q0, q1\}$
$\delta(q1, b) = \{q1, q2\}$
$\delta(q2, a) = \{q2\}$
$\delta(q0, \lambda) = \{q2\}$

with initial state q0 and final state q2 into an equivalent dfa.

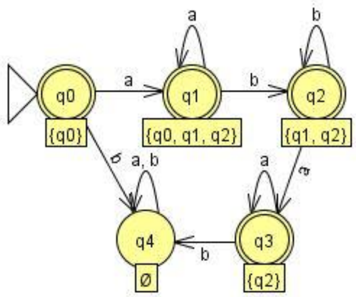
Answer:

nfa:



NFA δ * table:

	a	b
q0	{q0, q1, q2}	{}
q1	{}	{q1, q2}
q2	{q2}	{}



As long as the final DFA graph is correct (accepting the same language), then they can get full credits. If some states or transitions are wrong. You can give them partial points based on the correct part of DFA graph, table and the NFA graph.

#14) (3 pts) Given a regular language L, how can we show that L^R is regular?

Answer:

If a language is regular, there exists an accepting nfa for the language. To create an nfa for L^R, we reverse all the transitions and change initial state to accepting state. Add a new initial state with lambda transitions to all of the original accepting states of nfa for L. We have an accepting nfa for L^R, so L^R is regular.

Explaining the general ideas of how to construct the NFA for L^R would be enough. Don't need to prove with formal math notations. If only working on one example language without generalization, take 1 point off.