Start time: 6:15 pm
End time: 8:53
Total Time: 2:38

Andrew Plum
Prof Who
CS 385
9/3/2024

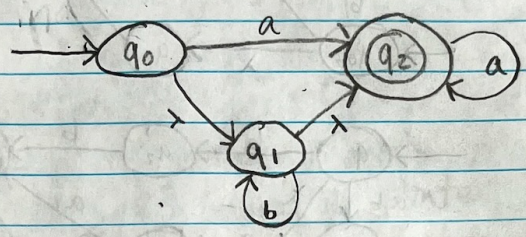
Homework 2

Section 2.2:

5) a) $\delta^*(q_0, 1011) = \delta^*(q_1, 011) \cup \delta^*(q_2, 011)$
$= \delta^*(q_0, 11) \cup \delta^*(q_2, 11)$
$= \delta^*(q_1, 1) = \{q_2\}$

b) $\delta^*(q_1, 01) = \delta^*(q_0, 1) \cup \delta^*(q_2, 1) = \{q_1\}$
Proof is procedure shown in class

11) a)

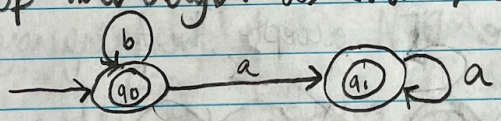


Proof: q_0 is the initial and by transitioning to q_2 from q_0 you cover $\{a^n : n \geq 1\}$ in the languag because you can loop infinitely at q_2.

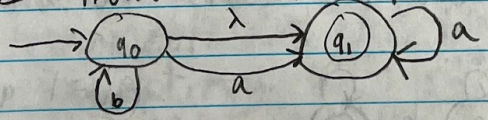
The λ transition from q_0 to q_1 covers the $\{b^m a^k : m \geq 0, k \geq 0\}$ part of the language because at q_1 you loop through as many b's as desired and then λ transition to q_2 where you loop through as many a's as desired. $q_2 \in F$

This NFA has less than 3 nodes and works
Proof: It exists and works

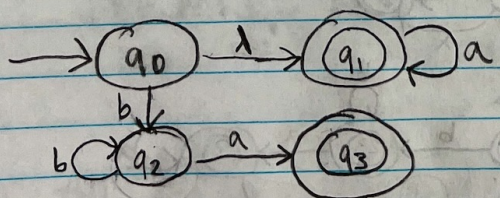
b)



I think this also works:



12)

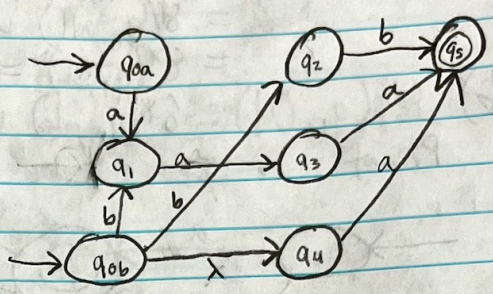


Proof: q_1 covers $\{a^n : n \geq 0\}$ and $q_2 \& q_3$ cover $\{b^n a : n \geq 1\}$. The transition made from q_0 determines which subset of the language we are using to satisfy our string which is why we have $q_1, q_3 \in F$

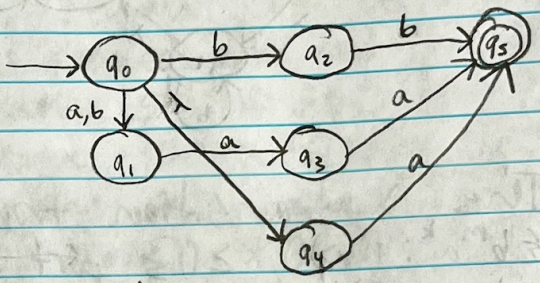
18) All you have to do to show this is to choose one $q_0 \in Q_0$ as you initial state & put all of the transitions which go from the initial state you are deleting as transitions which go from q_0 to $\delta(q_0, l)$ for

any $l \in \Sigma \cup \{\lambda\}$. Here is an example of this.

NFA with Q_0:

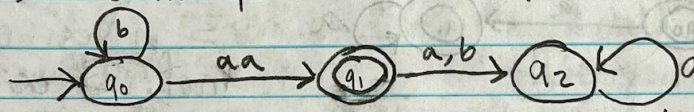


NFA with q_0:



Proof Cont.:
Notice both NFAs accept the language $\{aaa, baa, bb, a\}$
All the transitions going from q_{0b} are now coming from q_0 and going to wherever q_{0b} was going to.

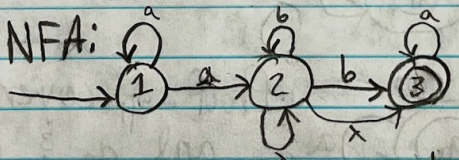
22) The incomplete DFA accepts the language $\{b^n aa : n \geq 0\}$



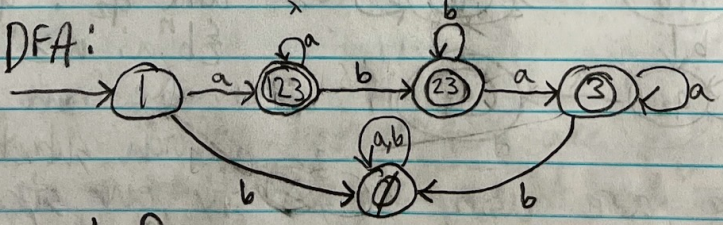
Proof: This standard DFA accepts the same language

Section 2.3: 5) NFA:

$q_0 = 1$ $q_1 = 2$ $q_2 = 3$



DFA:

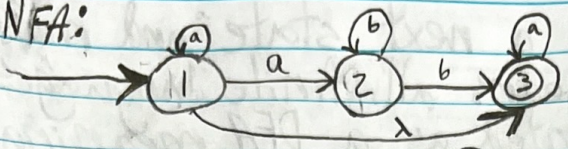


4) Next Page

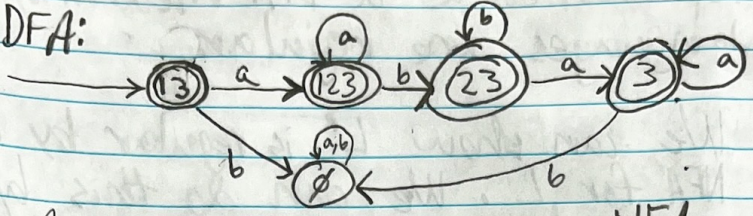
Andrew Alum
Prof. Woo
CS 395
4/3/2024

Homework 2 Cont.

5) NFA:



DFA:



9) Assume we have an NFA with multiple final states called M and an NFA M' with the same states and transitions except it has one extra state. This extra state in M' is the only final state whereas all of the final states in M are made nonfinal states in M'. All of the final states in M have λ transitions to the final state in M'. This so that when a final state in M is reached, the final state is reached in M'. The final state in M' should be thought of as a dead final state because you want it to be the last thing accepted meaning it should have no transitions. All we are simply doing is now accepting all strings $\{w\lambda : w$ is a string accepted by $M\}$ for M' by adding the extra λ transitions.

13) All finite languages are regular because they can be hardcoded as a DFA from the initial state. For all strings w in L, you take l in $w = lw'$ use that as the transition input for each transition from the previous state to the next state starting from the initial state. Once w' in $w = lw'$ is λ meaning l is the last letter, we transition with l from the previous

state to a next state and make it final. This means all finite languages can be represented as a DFA meaning all finite languages are regular.

14) We can show L^R is regular by creating a NFA for L. We can do this by first constructing a NFA M of L. From this, we create M', by taking the final state(s) in M & making them the nonfinal state(s) in M' and having a new λ transition from them to a new final state in M'. Then we construct M^R by taking the final state in M' and making it the nonfinal initial state in M^R and taking the initial state in M' and making it a final state in M^R. Using M^R we create a equivalent DFA using the theorem of DFA-NFA equivalency. Since we have made a DFA which accepts L^R, L^R is a regular language. Note we can only do this if L is a regular language.