Andrew Blum
Prof. Woo
CS 385
8/26/2024

Homework #1

Time:
Problems: 35 min
Proofs: 75 min

1) a) 0001 is accepted

Proof: We start at q_0 and loop 3 times back to q_0 using one 0 each time and then use the 1 to transition to q_1 which means the string is then accepted (no more string input) because $q_1 \in F$.

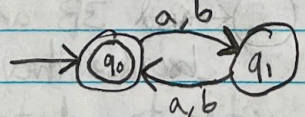
b) 01101 is accepted

Proof: First we start at q_0 and loop back to q_0 consuming one 0, next we transition from q_0 to q_1 using a 1 and then from q_1 to q_2 using another 1 and then we loop from q_2 to q_2 using a 0 and then transition from q_2 to q_1 using a 1 which is an end of input at an accept state ($q_1 \in F$). ■

c) 00001101 is accepted

Proof: First we start at q_0 and loop back to q_0 four times consuming four 0's and then we use a 1 to transition from q_0 to q_1 and then use another 1 to go from q_1 to q_2 and then loop once from q_2 to q_2 consuming a 0 and then transitioning from q_2 to q_1 using a 1 and then we run out of input at at an accept state ($q_1 \in F$). ■

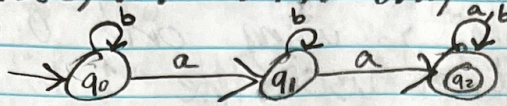
3) a)



Proof: No input is regarded as even for this DFA. When we start at q_0 there is no input used and when we transition to q_1 from q_0 one letter is used. q_1 is not an accept state only q_1 is. This means to get back to an accept state another letter must be used meaning 2 letters were used to cycle back to q_0. Effectively, we always leave and enter q_0 with an even number of letters and leave and enter q_1 with an odd number

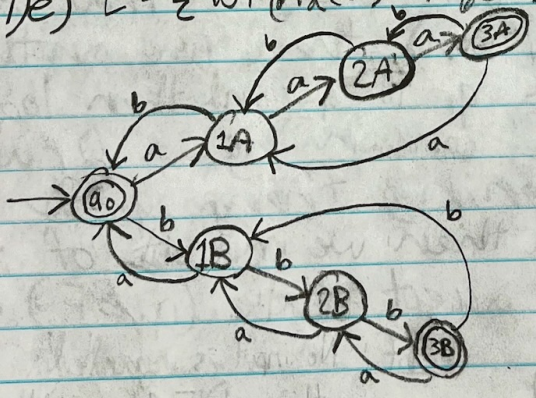
of characters. Because q_2 is the only final state, this DFA only accepts strings of even length.

4)b)



Proof: We start at q_0 where 0 a's have been read if a b is read we return to q_0 because q_0 is the state that 0 a's have been read. If an a is read from q_0 then we transition to q_1 to signify one a has been read; if a b is read at q_1 then we transition to q_1 because it represents the state saying one a has been read in. If an a is read in while at q_1, we transition to q_2 which is the state that represents that at least 2 a's have been read in. Since q_2 is the condition of accepting all strings with at least two a's that our DFA is trying to meet, $q_2 \in F$. The input after reaching this state does not matter because the condition having been met, is unchanging

7)e) $L = \{w : (n_a(w) - n_b(w)) \bmod 3 = 0\}$



Note: We need the strings with the diff. #a's ; # b's to be divisible by 3.

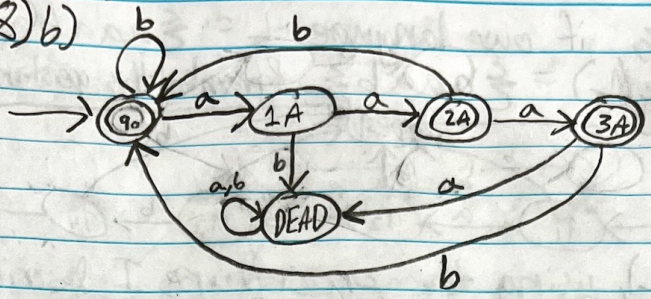
Proof: Each state of the DFA represents the number of more A's than B's or vice versa. q_0, 3A and 3B are all final states because they are all divisible by 3. At any given state we transition to whatever state is the result of the mod 3. For instance at state 1A if we read in an a then we go to 2A because $n_a(aa) - n_b(aa) = 2$ and $2 \bmod 3 = 2$. Take the string aaaabbbb, the states we go to starting at q_0 are 1A, 2A, 3A, 1A, q_0, 1B, 2B, 3B meaning the string is accepted which is what we want.

Next Page

Andrew Blum
Prof. Woo
CS 385
8/26/2024

Homework #1 Cont.

8) b)

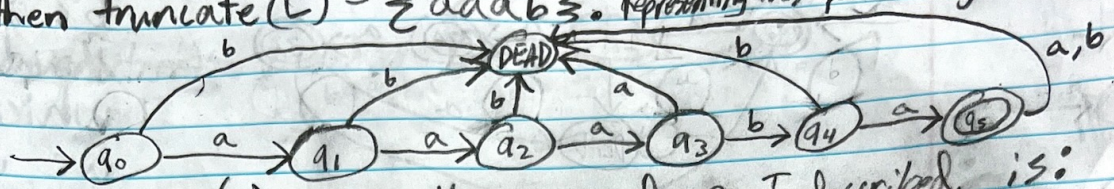


Proof: Strings of just b's and nothing should be accepted which is what we want. If an a is read in at q_0 then we transition to 1A. If a b is read in at 1A then we go to the state called DEAD which is not a final state because the condition the string is supposed to meet is already violated and can't be unviolated. If we read an a at 1A then we go to 2A which is a final state. From 2A we go back to q_0 if we read a b to "reset" the count of the number of a's. If we read an a from 2A, we go to 3A which is also a final state because a run of 3 a's meets the condition. If another a is read from 3A then we go to DEAD because we can't have a run of 4 a's or more and we are stuck at DEAD. If we read a b at 3A then we go back to q_0 to reset the count of a's just like we did when we went from 2A to q_0 using a b.

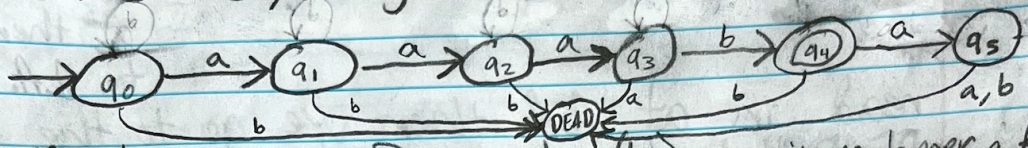
26) Truncate () just removes the right most character of a string and we can demonstrate the same can be done for language L by changing the final states in the DFA representing L to non-final states and make the nodes that were previously traveling to the old final states the new final state nodes.

Back Side

For instance, if our language $L = \{aaaba\}$ then truncate$(L) = \{aaab\}$. Representing this operation using DFA's L is:



Truncate (L), using the procedure I described, is:



Effectively, in Truncate (L) q_5 is no longer a final state and q_4 was made a final state because it traveled into q_5. None of the transitions need to change; only the membership of certain states changed.

A DFA can always be constructed for L making it regular and because a DFA can always be constructed for truncate (L), truncate (L) is also regular.