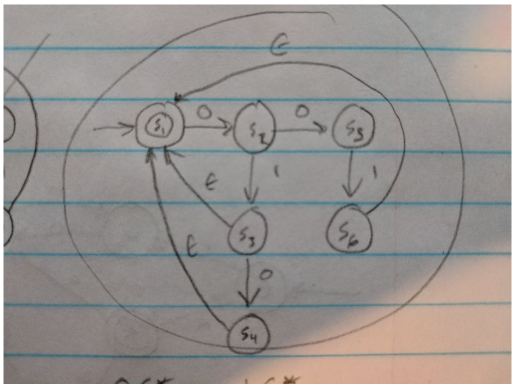
Ryan Darras, CS 5700 - HW03 1.(17, 19, 21, 40(b)) SP1.4
*I put some links to relevant work on imgur to increase readability.

Problem 1.17

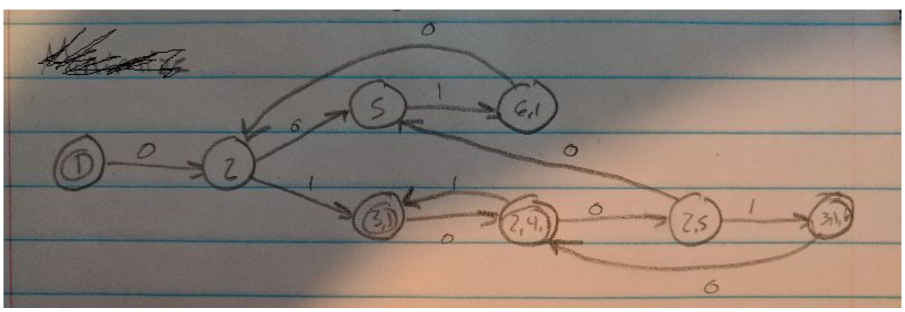
a) Give an NFA recognizing the language (01 ∪ 001 ∪ 010)*.
b) Convert this NFA to an equivalent DFA. Give only the portion of the DFA that is reachable from the start state.

Problem 1.17 a Answer

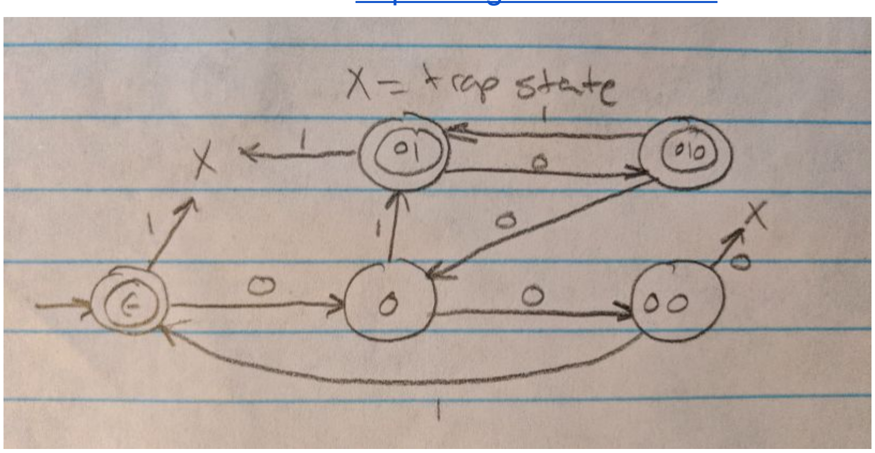


Problem 1.17 b Answer

DFA - Work = https://imgur.com/a/bh3dV



Minimized DFA - Work = https://imgur.com/a/7Xxlh

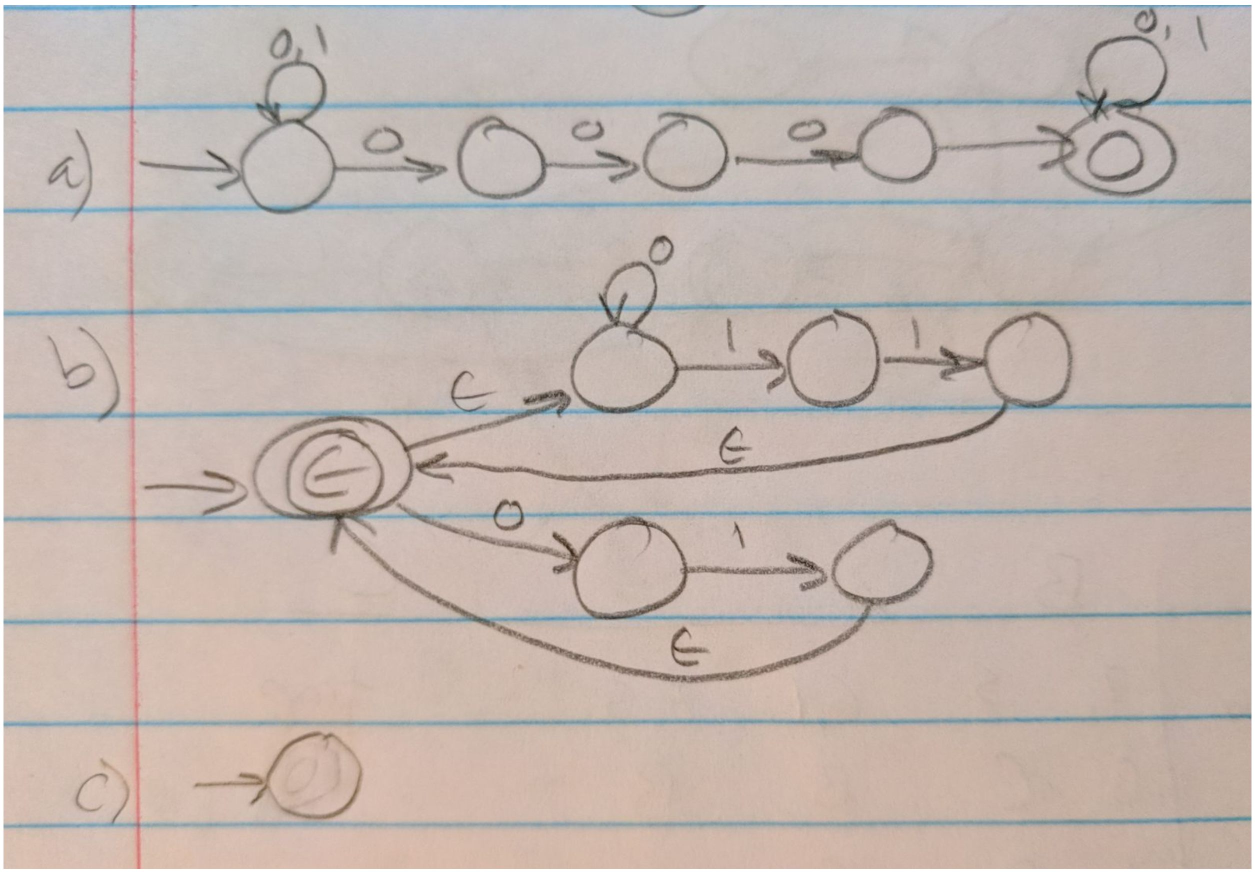


Problem 1.19

Use the procedure described in Lemma 1.55 to convert the following regular expressions to nondeterministic finite automata.

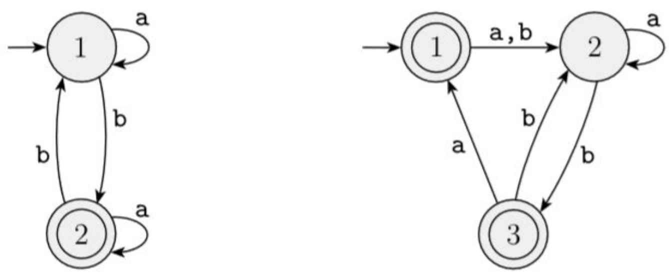
a) (0 ∪ 1)*000(0 ∪ 1)*
b) (((00)*(11)) ∪ 01)*
c) ∅*

Problem 1.19 a, b, c Answer



Problem 1.21

Use the procedure described in Lemma 1.60 to convert the following finite automata to regular expressions.



(a) (b)

Problem 1.21 Answer

a) a*b((bb)* ∪ a*)* ∪ a*b(((bb* ∪ a*)ba*b)*
b) (a∪b)(a*∪(bb)*)ba ∪ (a∪b)(a*∪(bb)*)b ∪ ε

Problem 1.40 b

Recall that string x is a prefix of string y if a string z exists where xz = y, and that x is a proper prefix of y if in addition x/= y. In each of the following parts, we define an operation on a language A. Show that the class of regular languages is closed under that operation.

b) NOEXTEND(A)={w ∈ A|w is not the proper prefix of any string in A}.

Problem 1.40 b Answer

Assume there is DFA M that accepts A. We can prove that language L = NOEXTEND(A) is regular by showing that there is DFA N that recognizes it. Because the nature of NOEXTEND means that once we reach an accept state, we cannot add any more characters to reach any other accept state. This means that there cannot exist a path from any accept state to any other accept state (including itself). Say F is the set of accept states for M. Take F’ ⊆ F where F’ is the set of all states where such a path doesn’t exist. Change F in M to F’ and you get N, showing that N is a DFA, hence L = NOEXTEND(A) is regular, which means that A is closed under the NOEXTEND operation.

Problem SP1.4

L = { w ∈ {0,1}* | w is a base-2 integer divisible by 3 but w^r is base-3 number not divisible by 2}

As usual, w is read msd first.

Example: w = 1100

$w = 1100_2 = 12_{10}$

$w^r = 0011_3 = 4_{10}$

Thus w is not in the language because w^r is divisible by 2.

Design a minimal DFA that establishes whether or not any strings are in L. If there are, find the first string, in a shortlex sense, that is.

Problem SP1.4 Answer

Doing some thinking, we realize w in base-2 is divisible by 3 if groups of 1's are even (ie: 1100 or 0110) and w^r in base-3 is divisible by 2 if the number of 1's is even. By this logic, we know that there are no strings in L because the two requirements contradict each other. The following DFA shows this because for the first requirement, states 1 and 4 would be the accept state, but these cannot be accept states for the second requirement because it would result in the w^r in base-3 to be divisible by 2.