Homework 1 solutions

1. (10 + 5 = 15 points)
a. Let $M_1 = (Q_1, \Sigma, \delta_1, {q_0}^1, F_1)$ and $M_2 = (Q_2, \Sigma, \delta_2, {q_0}^2, F_2)$ be two DFA.

Construct a DFA that accepts $L(M_1)$ - $L(M_2)$. (If A and B are two sets then
$A - B = \{ x \mid x \in A \text{ and } x \notin B \}$.

Give a precise description of how the new DFA is constructed in terms of the two original DFA's.

New constructed DFA = $M_3 = (Q_3, \Sigma, \delta_3, s_3, F_3)$,

where

$Q_3 = Q_1 \text{ X } Q_2 = \{(p, q) \mid p \in Q_1 \text{ and } q \in Q_2\}$
Σ = same Σ as in M_1 and M_2
$F_3 = F_1 \text{ X } (Q_2 - F_2) = \{(p, q) \mid p \in F_1 \text{ and } q \in Q_2 \text{ and } q \notin F_2\}$
$s_3 = (s_1, s_2)$,

and let:
δ_3: $Q_3 \text{ X } \Sigma \rightarrow Q_3$

be the transition function defined by:

$\delta_3((p, q), a) = (\delta_1(p, a), \delta_2(q, a))$

(Note that this construction is very similar to the product construction on page 22 of the book which is used for intersection; the only difference is the accepting states as defined in F_3)

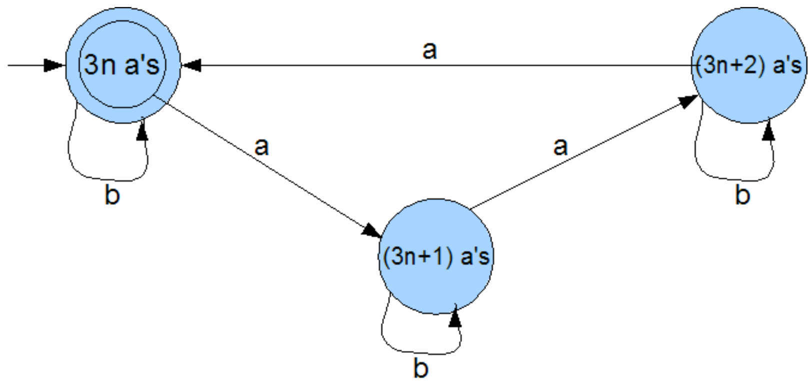
b. Give a DFA to accept the set of strings a's and b's where the number of a's is divisible by 3. Also give a DFA to accept the set of strings of a's and b's where the number of b's is even. Call these two DFA's M_1 and M_2 respectively.

(i) What is $L(M_1)$ - $L(M_2)$.

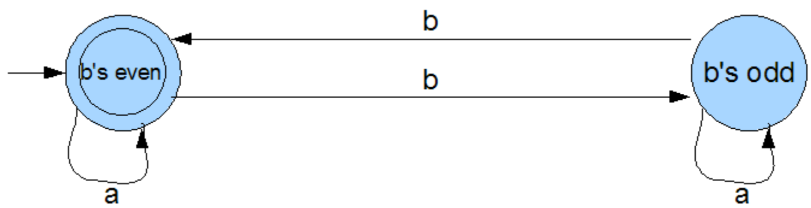
$L(M_1)$ - $L(M_2)$ represents all strings where the number of a's is divisible by three and the number of b's is NOT even (AKA: the number of b's is odd).

You will not receive any credit if you do not apply the construction you describe in your answer to part a.

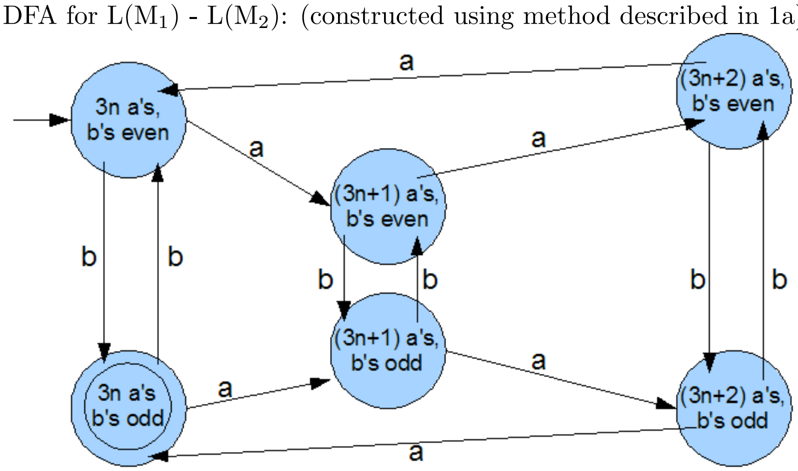
DFA for M1:



DFA for M2:

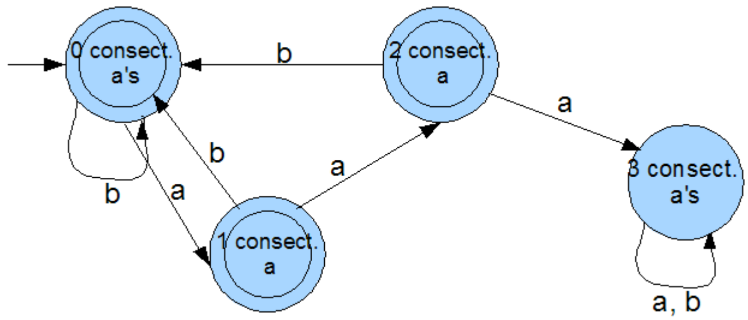


DFA for $L(M_1)$ - $L(M_2)$: (constructed using method described in 1a)

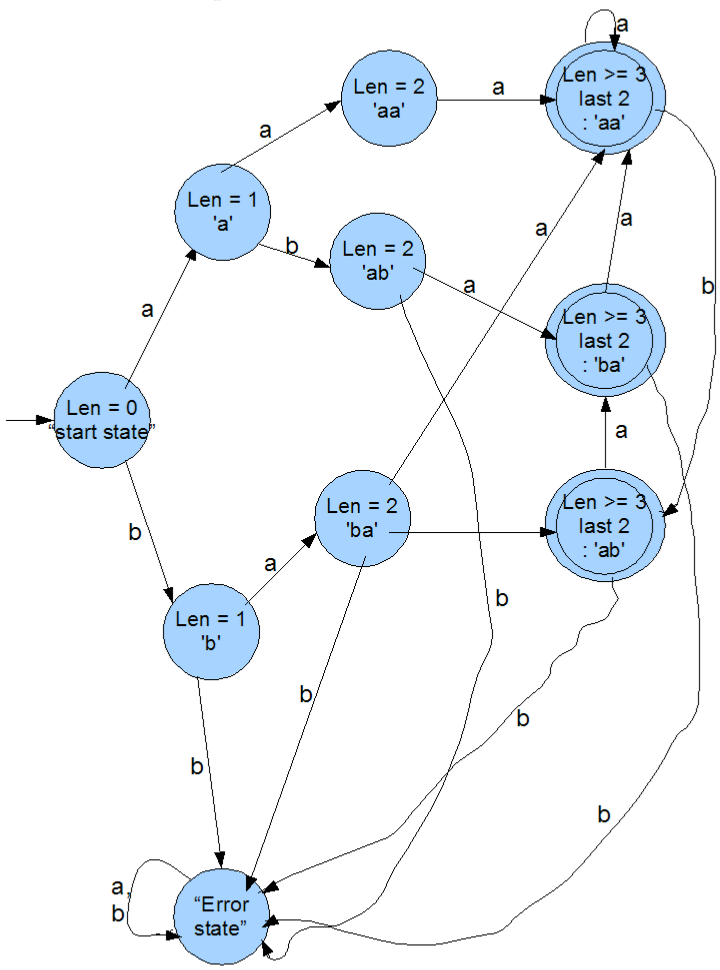


2. ($7 + 8 + 7 + 8 = 30$ points).
For each of the following four languages, give a finite state automaton to accept it. Assume $\Sigma = \{a, b\}$ for the first two languages, $\Sigma = \{a, b, c\}$ for the third language and for the fourth language, assume $\Sigma = \{0, 1, \ldots, 9, .\}$.

a. The set of strings that do not contain three consecutive a's.



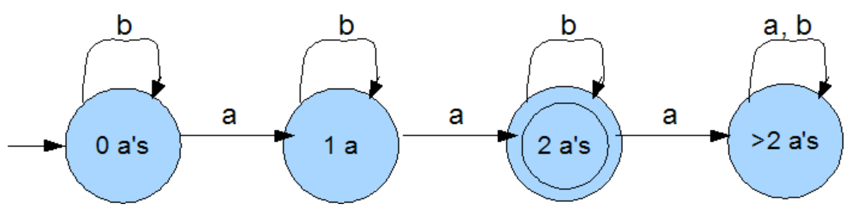
b. The set of strings (of length at least three) where any contiguous string of length three contains at least 2 a's. For example, aababa should not be accepted.



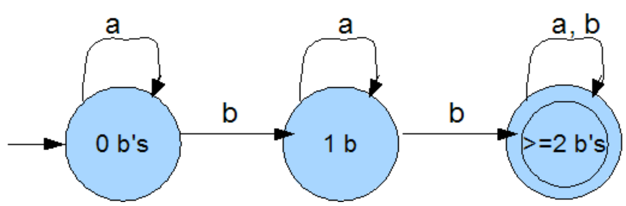
c. set of strings that contain exactly 2 occurences of ‘a’ and 2 or more occurences of ‘b’.

This is done using the product construction described on page 22 of book, starting with finding the DFAs for the set with exactly 2 occurences of ’a’ and the set with 2 or more occurences of ’b’, then using the product construction to find the intersection between the two sets:

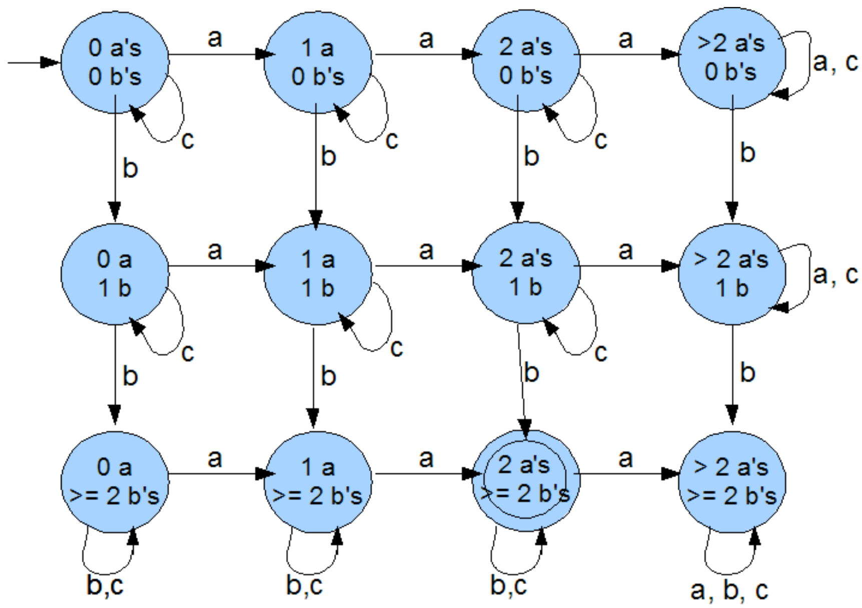
DFA for set of strings with exactly 2 occurences of ’a’:



DFA for set of strings with 2 or more occurences of ’b’:



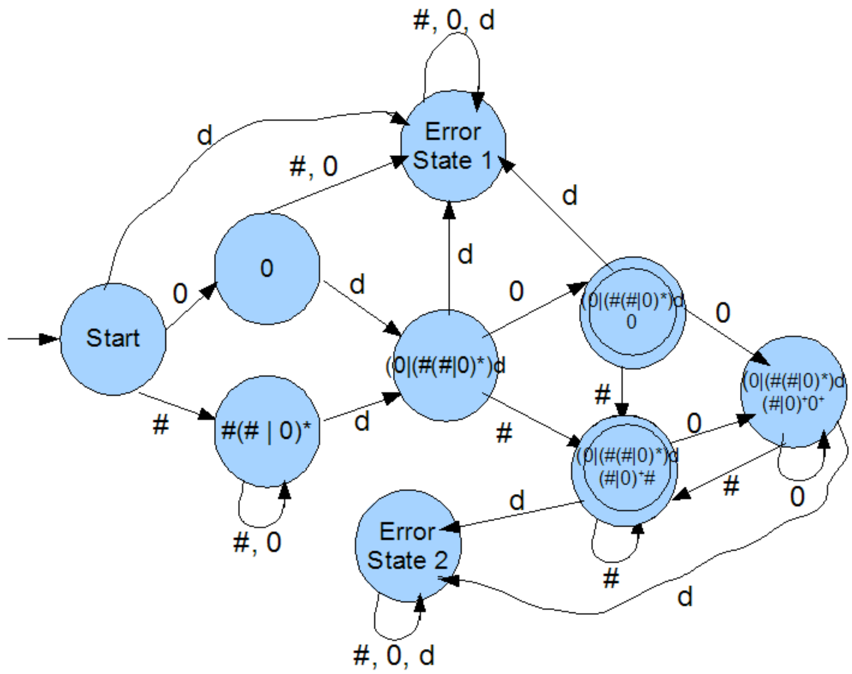
DFA for ’product’/intersection of above two DFAs...this results in the DFA for the set of strings that contains exactly 2 occurences of ’a’ and 2 or more occurences of ’b’:



d. Fixed-decimal literals with no superfluous leading or trailing zeros. Every literal has at least one digit before and after the decimal point. Thus, for example, 0.0, 1.0, 0.1, 123.01, and 123005.0 are legal, but 0, .12, 23., 01.0, 1.000, and 002345.1000 are not.

In the below FSA: '0' represents the number 0, '#' represents a number 1-9, and 'd' represents a 'decimal point'. In addition, 'Error State 1' and 'Error State 2' could be combined into a single state; they are split here to make the FSA more clear...

The text in each state represents the regular expression that corresponds to strings in the state.

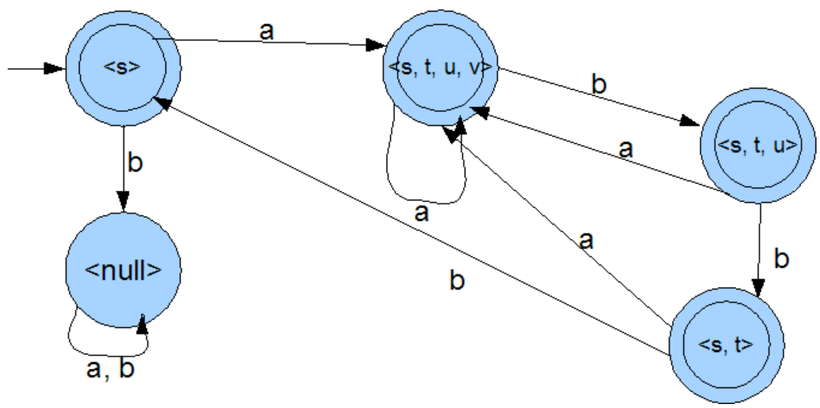


3. (3 + 6 + 6 = 15 points)
a. Exercise 3 on Page 316 of text:

Consider the following nondeterministic finite automaton (shown on page 316 of book...):

(a): Give a string beginning with 'a' that is NOT accepted: abbbb

(b): Construct an equivalent deterministic automaton using subset construction. Assuming the states are named s, t, u, v from left to right, show clearly which subset of {s, t, u, v} corresponds to each state of the deterministic automaton. Omit inaccessible states.



b. Exercise 5b (you don't need to answer 5a) on Pages 316-317 of the textbook:

Convert the nondeterministic finite automata on the top of page 317 in the text to an equivalent deterministic one using subset construction.