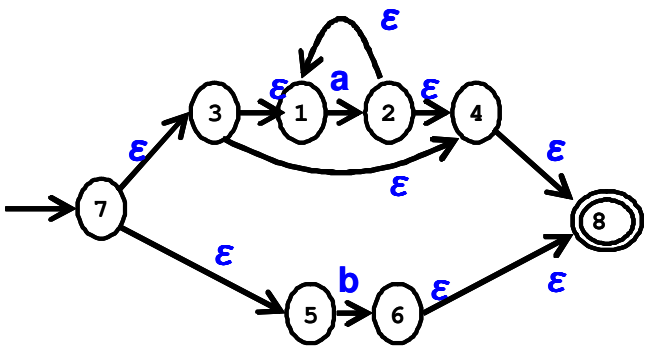
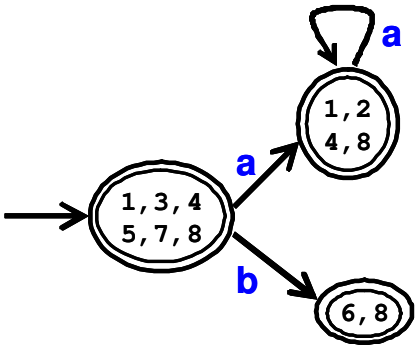
CMSC430 Spring 2009 Quiz 1 (Solutions)

1. (15 pts) Regular expressions and finite automata
   a. (4 pts) Using Thompson's construction method (or the alternative closure used in the practice problems), create a NFA for the regular expression a* | b.

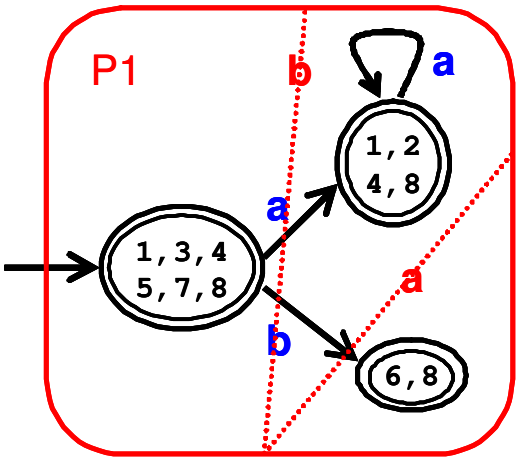


   b. (6 pts) Apply the subset construction algorithm to convert the resulting NFA to a DFA. Show the NFA states associated with each state in your DFA.



   c. (5 pts) Start applying Hopcroft minimization to the resulting DFA. List the DFA states in each initial partition created. Pick a partition and explain why the partition will or will not be split by the minimization algorithm, and list the DFA states in each resulting partition if the partition is split.

      Initially all in same partition (final states). Partition will be split because behavior is different. For input=a {6,8} is split because it rejects a. For input=b starting state is split because it does not reject b.



2. (25 pts) Top down parsing
   Consider the following grammar: $S \rightarrow AB$ $A \rightarrow aAb \mid \varepsilon$ $B \rightarrow c \mid \varepsilon$

   a. (3 pts) Compute FIRST for AB

      { ε, a, c}

   b. (8 pts) Compute FOLLOW for S, A, and B

      Follow(S) = { $ }, Follow(A) = { b, c, $ }, Follow(B) = { $ }

   c. (8 pts) Construct the LL(1) parse table for the grammar

	a	b	c	$
S	**S → AB**		**S → AB**	**S → AB**
A	**A → aAb**	**A → ε**	**A → ε**	**A → ε**
B			**B → c**	**B → ε**

   d. (2 pts) Prove the grammar is or is not LL(1)

      It is LL(1) since there are no multiple entries in the LL(1) parse table.

   e. (4 pts) Using the LL(1) parse table, parse the input "abc". You only need to show the stack and remaining input at each stage of the parse.

Stack	Input
$ S	**abc$**
$ B A	**abc$**
$ B b A a	**abc$**
$ B b A	**bc$**
$ B b	**bc$**
$ B	**c$**
$ c	**c$**
$	**$**