Exam 1 CPS 140 Feb. 2012

1. (12 pts) Complete or answer the following.

$L_1 = \{a, d\}$, $\Sigma = \{a, d\}$
$L_2 = \{b\}$, $\Sigma = \{b\}$
$L_3 = ab^*(a + b)$, $\Sigma = \{a, b\}$
$L_4 = b^*a^*$, $\Sigma = \{a, b\}$
$L_5 = \{w \in \Sigma^* \mid n_a(w) = n_b(w)\}$, $\Sigma = \{a, b\}$

(a) $L_1 \cap L_2 =$ $\emptyset$

(b) $L_1 - L_2 =$ $L_1 = \{a, d\}$

(c) $L_3 \circ L_4 =$ $ab^*(a+b)b^*a^*$

(d) $L_4 \cap L_5 =$ $\{b^n a^n \mid n \geq 0\}$

(e) $L_2 \times L_2 =$ $\{(b, b)\}$

(f) $2^{L_1} =$ $\{\emptyset, \{a\}, \{d\}, \{a, d\}\}$

2. (22 pts) Answer TRUE or FALSE to each of the statements below.

(a) If M is a DFA that does not have a trap state and that has only one cycle, which is of length 1, then $L(M)$ is an infinite language. (TRUE or FALSE?) TRUE

(b) If M is an NPDA with some transitions that push three or more symbols on the stack, then there exists an NPDA M' such that all of M's transitions push only 0, 1, or 2 symbols on the stack, and L(M)=L(M'). (TRUE or FALSE?) TRUE

(c) If M is an NPDA that has at most two stack symbols, then there exists a regular grammar G such that L(M) = L(G). (TRUE or FALSE?) FALSE $|\Gamma| \leq 2$

(d) Consider a CFG G and the parse tree for a string in L(G). All non-leaf nodes in the parse tree are variables from the grammar. (TRUE or FALSE?) TRUE

(e) If R is a regular expression, then there exists an NPDA M such that $L(R) = L(M)$. (TRUE or FALSE?) TRUE

(f) Consider the following statement involving regular expressions.
a*(b + a)* $= (a+b)^*$ (TRUE or FALSE?) TRUE

(g) The following grammar G is a regular grammar. (TRUE or FALSE?) TRUE

$$S \rightarrow Sa \mid Bba \mid d$$
$$B \rightarrow Sb \mid \lambda$$

(h) $L = \{a^{3n}c^{4m} \mid n > 0, m > 0\}$, $\Sigma = \{a, c\}$. L is regular. (TRUE or FALSE?) TRUE

(i) L=$\{w \in \Sigma^* \mid n_a(w) < n_b(w) + 10\}$ $\Sigma = \{a, b\}$. L is regular. (TRUE or FALSE?) FALSE

(j) $L = \{a^n b^p c^q \mid n > p, q > p, p > 0\}$, $\Sigma = \{a, b, c\}$. L is regular. (TRUE or FALSE?) FALSE

(k) L=$\{w \in \Sigma^* \mid n_a(w)$ is odd and *abc* is not a substring$\}$ $\Sigma = \{a, b, c\}$. L is regular. (TRUE or FALSE?) TRUE

3. (4 pts) Consider the following definition related to NPDA's.

$$L(M)=\{w \in \Sigma^* | (q_0, w, z) \overset{*}{\vdash} (p, \lambda, \lambda)\}$$

(a) Explain the general idea of what this definition means in words.
This is the definition of acceptance by empty stack. The string w is accepted after all symbols have been seen and the stack is empty.

(b) Explain these parts of the definition: $(q_0, w, z) \vdash (p, \lambda, \lambda)$

Starting in state q0 with string w on the tape and z on the stack, after 0 or more transitions, all symbols in w have been processed and the stack is empty and the current state is some state p.

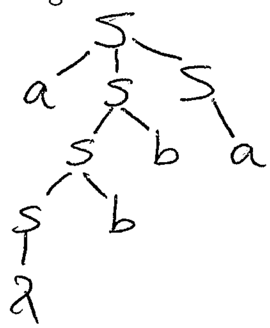
4. (4 pts) Consider the following grammar.

$$S \rightarrow aSS \mid Sb \mid a \mid \lambda$$

A) Give a left-most derivation for the string *abba*.

$S \Rightarrow aSS \Rightarrow aSbS \Rightarrow aSbbS \Rightarrow abbS$
$\Rightarrow abba$

B) Give a parse tree for the string *abba*



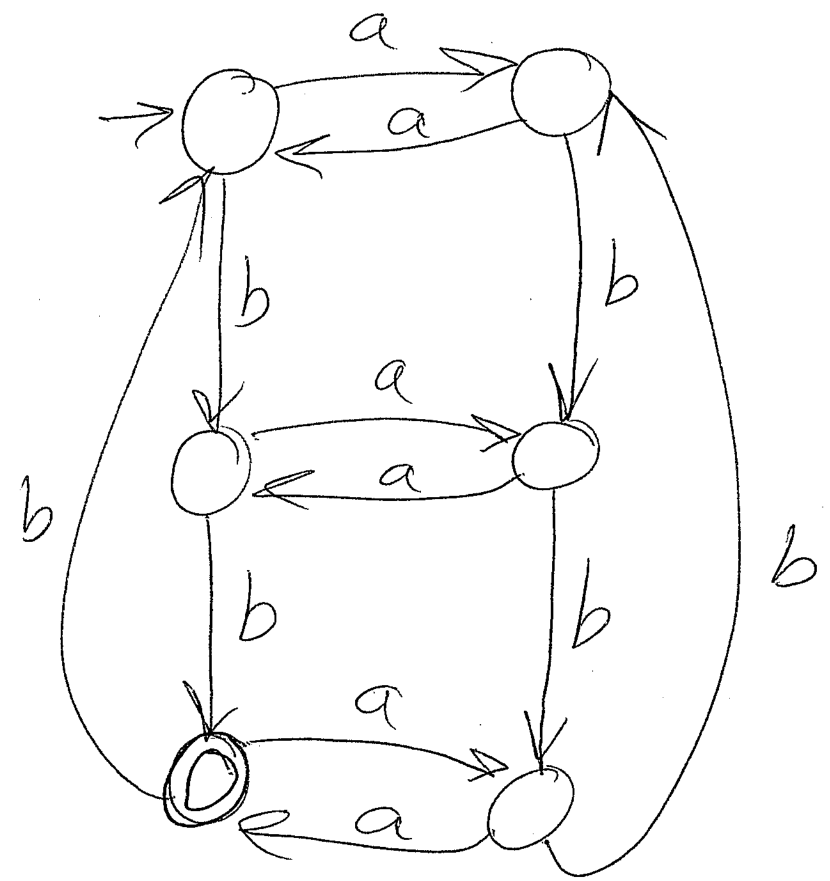
5. (5 pts) Write a CFG G for the following language:
$L = \{a^n b^p c^q \mid n > p + q, p \geq 0, q > 0\}$, $\Sigma = \{a, b, c\}$.

$S \rightarrow aSc \mid aBc$
$B \rightarrow aBb \mid aC$
$C \rightarrow aC \mid \lambda$

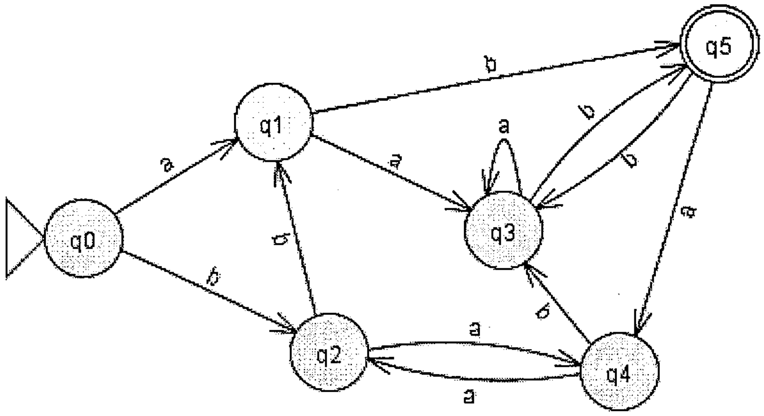
6. (8 pts) Draw a DFA for the following language. You do not have to show trap states (show the transition diagram, indicate the start state by a short arrow, and final states by double circles.)

   L=$\{w \in \Sigma^* \mid n_a(w)$ is even and $n_b(w) \bmod 3 = 2\}$, $\Sigma = \{a, b\}$.

   For example, *bbaa*, *bbbbb* and *bababbaab* are in L.



7. (6 pts) Consider the following DFA.



a) Show states $q0$ and $q2$ are distinguishable with an appropriate string. Explain.

ab $\delta(q0, ab) = q5$ $\delta(q2, ab) = q3$

q5 is final state q3 is a nonfinal state

b) Give the states in the minimal state DFA (you do not need to show the arcs). Each state should indicate which states it represents from the original DFA. For example you could list one state as 0,1,2 if states 0, 1 and 2 in the original DFA can be combined to form a state in the minimal state DFA.

012345

01234 5

b

13 024

a

0 24

4 states

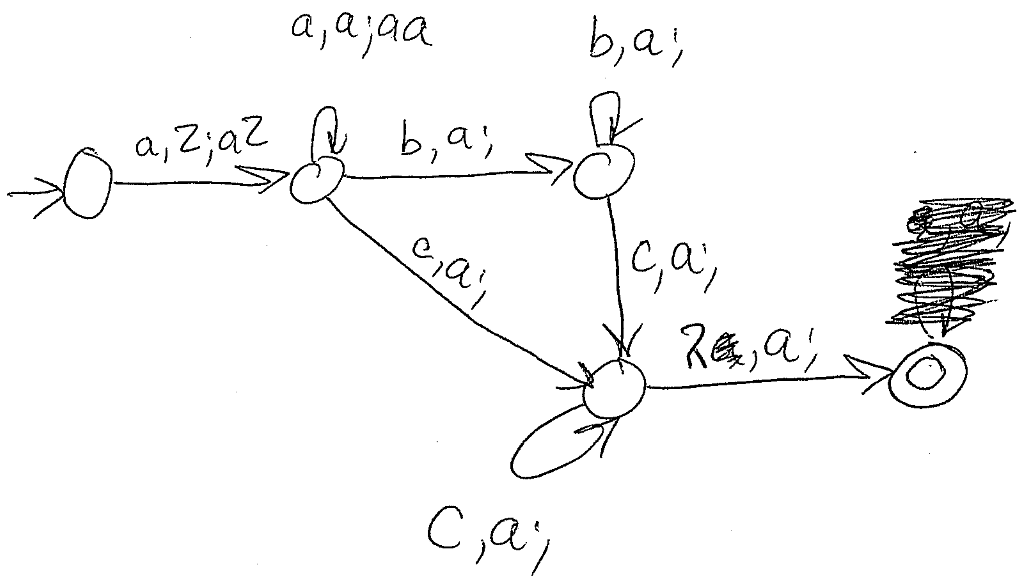
0 13 24 5

8. (10 pts) Consider $L = \{a^n b^p c^q \mid n > p+q, p \geq 0, q > 0\}$, $\Sigma = \{a, b, c\}$. Draw the transition diagram for a nondeterministic pushdown automaton M that accepts L by final state. (Remember to identify the start state by an arrow and final states by double circles. Format of labels are $a, b; cd$ where a is the symbol on the tape, b is the symbol on top of the stack that is popped, and cd are pushed onto the stack (with c on top of d). Z is on top of the stack when M starts.).

(a) First list 3 strings in L.

aac, aaabc, aaaabc

(b) Now draw the transition diagram.



you don't have to pop items off at the end.

9. (6 pts) **Pumping Lemma:** Let L be an infinite regular language. $\exists$ a constant $m > 0$ such that any $w \in L$ with $|w| \geq m$ can be decomposed into three parts as $w = xyz$ with

$$|xy| \leq m$$
$$|y| \geq 1$$
$$xy^iz \in L \quad \text{for all } i \geq 0$$

Use the Pumping Lemma to prove the language L below is not regular.

$L = \{a^n b^p c^q \mid n > p + q, p > 0, q \geq 0\}$, $\Sigma = \{a, b, c\}$.

Proof: (SHOW ALL STEPS! Some have been started for you.)

Assume L is regular

Choose w = $a^{m+1} b^m$

Show there is no way to partition this string w=xyz such that the properties of the pumping lemma hold.

$x = a^j \quad y = a^t \quad z = a^{m+1-j-t} b^m$

$i = 0 \quad xy^0z = xz = a^{m+1-t} b^m \quad t > 0$

$n_a(w) \leq n_b(w) + n_c(w)$

contradiction!

$\Rightarrow$ L is not regular

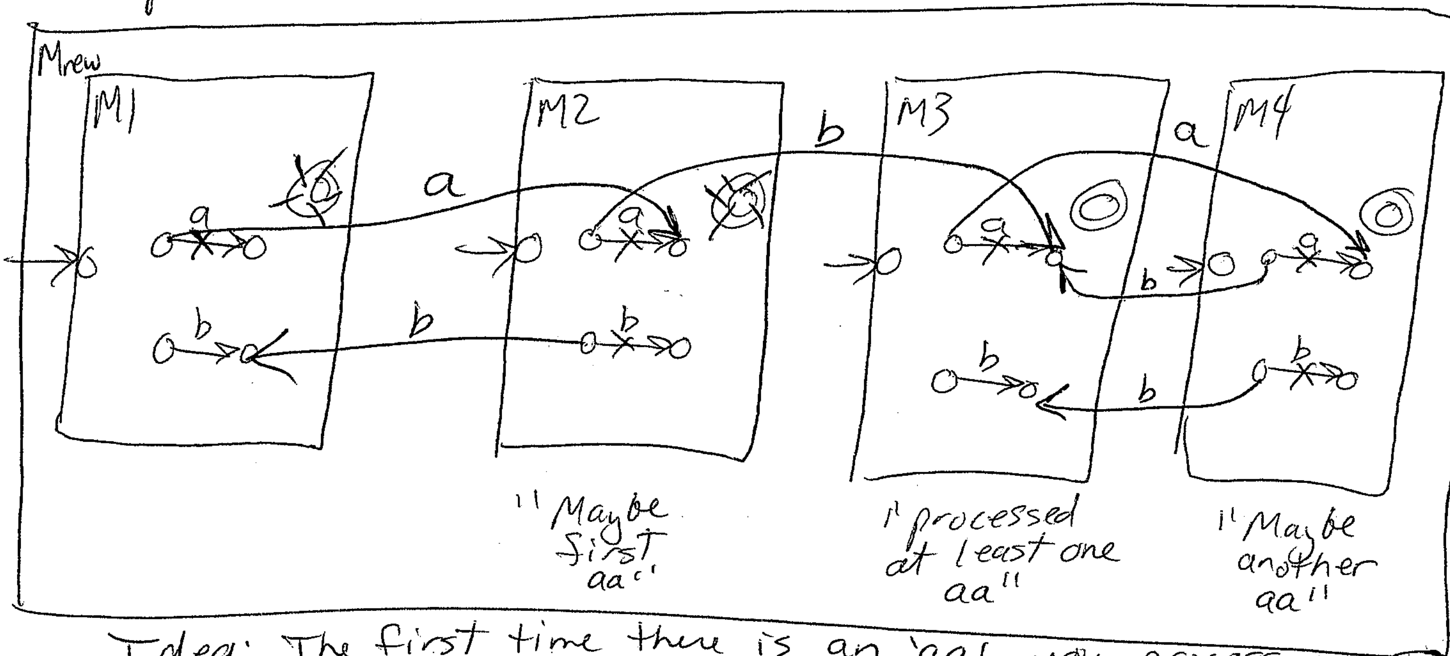
10. (8 pts) Consider the following property, ReplaceAllWith. If L is a regular language, then ReplaceAllWith(L)= strings from L that have every occurance of aa replaced with ab. If there is a string w in L that does not have the substring aa in the string, then that does not put w in ReplaceAllWith(L).

For example, if $aaaa$ is in L, then $abab$ is in ReplaceAllWith(L), the second a of each aa was replaced by b. If $bbaaaaaaab$ is in L, then $bbabababab$ is in ReplaceAllWith(L), with three as (all the second a of an aa) replaced with a b.

If ab is in L, then ab does not generate a string in ReplaceAllWith(L).

Prove that ReplaceAllWith(L) is a regular language.

L is regular. ∃ DFA M for L. Make 4 copies of M called M_1, M_2, M_3 & M_4.



Idea: The first time there is an 'aa', you process the first a into M2, then the second a (replacing w/ a b) into M3. At that point you can start accepting strings. In M3 every 'a' goes to M4 to see if a 'a' follows it (replace with a b) or a b follows it. Both go back to M3.

Changes: 1) No final states in M1 or M2.
2) all a arcs in M1 replace with a to corresponding state in M2
3) all a arcs in M2 replace with b arc to M3 to corresponding state
4) all b arcs in M2 replace with b arc to corresponding state in M1
5) all a arcs in M3 replace with a arcs to " " " M4
6) all a arcs in M4 " " b " " " " M3
7) all b arcs in M4 replace with b arc to " " " M3.