CPS 140  
Exam 1  
Spring 2009  
Dr. Rodger

NOTE: \( n_a(w) \) means the number of a's in the string \( w \).

1. (14 pts) Complete or answer the following. If the resulting answer has less than 10 items, just list the items. Otherwise give a formal description of the language.

   \[
   L_1 = \{a, c\}, \Sigma = \{a, c\}  \\
   L_2 = \{a, bb\}, \Sigma = \{a, b\}  \\
   L_3 = b^*ba^*, \Sigma = \{a, b\}  \\
   L_4 = \{w \in \Sigma^* | n_a(w) > n_b(w)\}, \Sigma = \{a, b\}
   \]

(a) \( L_1 \times \{a\} = \{a(a), a(c), c(a)\} \)
(b) \( 2L_1 = 4 \{\emptyset, \{a\}, \{c\}, \{a, c\}\} = 2L_1 \)
(c) \( L_1 \cup L_2 = \{a, c, a_b, c_b\} \)
(d) \( L_1 \cap L_3 = \emptyset \)
(e) \( L_3 \cap L_4 = \{b^m a^n | m > 0, n > 0\} \)
(f) \( \overline{L_3} = \{b^* | a \notin b^*\} \)
(g) \( L_1 \cap L_3 = a^* b^* a^* b^* + c^* a^* b^* \)

2. (20 pts) Answer TRUE or FALSE to each of the statements below.

(a) \( \emptyset \in \{\emptyset, \{a\}, \{a, b\}\}\) \( \) (TRUE or FALSE?)
(b) If there is an NPDA \( M \) then there exists a regular grammar \( G \) such that \( L(M) = L(G) \).  \( \) (TRUE or FALSE?)
(c) If \( L \) is regular then \( L \cap \{a^n b^n | n > 0\} \) is regular. \( \) (TRUE or FALSE?)
(d) If \( G \) is a regular grammar then there is not always a unique parse tree for every string \( w \in L(G) \). \( \) (TRUE or FALSE?)
(e) If there is an NFA with only 1 final state, then there exists an equivalent DFA with only 1 final state. \( \) (TRUE or FALSE?)
(f) If \( r \) is a regular expression, then there exists a CFG \( G \) such that \( L(r) = L(G) \). \( \) (TRUE or FALSE?)
(g) \( L = \{w \in \Sigma^* | n_a(w) = n_b(w) \text{ and } n_b(w) \text{ is even}\} \Sigma = \{a, b\}, L \) is regular. \( \) (TRUE or FALSE?)
(h) \( L = \{a^{2n+1} | n > 0\}, \Sigma = \{a\} \). \( L \) is regular. \( \) (TRUE or FALSE?)
3. (4 pts) Explain what each of the parts of a 5-tuple for a deterministic finite automaton represent, $M = (Q, \Sigma, \delta, q_0, F)$. Give the formal definition for $\delta$.

4. (8 pts) Draw a DFA for the following language. You can show trap states if you want. (show the transition diagram, indicate the start state by a short arrow, and final states by double circles.)

$L = \{ w \in \Sigma^* \mid w \text{ has the substring } bba \text{ and } w \text{ does NOT have the substring } aa \}$

$\Sigma = \{a, b\}$.

For example, $bbbab$, $bba$ and $abba$ are in $L$.

5. (6 pts) Write an NFA that is equivalent to the following regular expression: $(bc+a)^*b$.

6. (6 pts) Write a context-free grammar for the following language.

$L = \{a^{2m+n}b^n c^m \mid m > 0, n > 0\}$

7. (6 pts) Consider the following DFA.

![DFA Diagram]

a) Show states $q_0$ and $q_3$ are distinguishable with an appropriate string. Explain.

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.
3. see lecture notes

4. 

5. 

6. 

S → aSc | aBc
B → aaBb | aab

7. a) b on b goes to q2 which is non-final. q3 on b goes to q4 which is final
b) \[ 0 1 2 3 4 5 6 \]
\[ 0 1 2 3 5 \quad 4 \quad 6 \quad 0 \quad 4 \quad 6 \]

4 states: \[ 0 1 2 3 5 \quad 4 \quad 6 \]

8. a) \( a b b a b c, \ a a b, \ a a a a a b b c \)

b) \[ a, a, a \quad b, a, \lambda \quad c, a, \lambda \]
\[ 0, a, 2, a, \lambda \quad 0, b, a, \lambda \quad 0, c, a, \lambda \quad 0, c, a, \lambda \]

9. Assume \( L \) is regular

Choose \( w = a^{2m+2} b^m \)

\( w = x y z \) show there is no way to partition

so \( RL \) holds

\( y = a^k, \quad x = a^i, \quad z = a^{2m+2-k-j} b^m \)

if \( i = 0 \quad x y z = x z = a^{2m+2-k-j} b^m \)

then \( n_a(w) \leq 2 \times n_b(w) + 1 \)

Contradiction! \( \Rightarrow L \) is not regular
10. \( L \) is regular \( \Rightarrow \) \( \exists \) DFA \( M \) for \( L \).

Make 4 copies of \( M \): \( M_1, M_2, M_3, M_4 \).

In \( M_1, M_2, M_3 \) all final states are made nonfinal.

In \( M_1 \), for each \( a \) arc, add an \( \lambda \) arc to the corresponding result state in \( M_2 \).

In \( M_1 \), for each \( b \) arc add a \( \lambda \) arc to the corresponding result state in \( M_3 \).

In \( M_2 \), for each \( b \) arc add a \( \lambda \) arc to the corresponding result state in \( M_4 \).

In \( M_3 \), for each \( a \) arc add a \( \lambda \) arc to the corresponding result state in \( M_4 \).

The start state in \( M_1 \) is the new start state. Exactly one \( a \) and one \( b \) will be replaced to reach \( M_4 \) which has final states.

See picture ⇒