CISC 4090
Theory of Computation

Midterm Exam

Prof. Daniel Leeds

The exam is closed book. No calculator is needed for the exam. Please show your work.

Name: ____________________________

Date: ____________________________
1. (6 points) Given the alphabet, $\Sigma = \{a, b\}$, provide a DFA or NFA that recognizes the language $L_1 = \{w | w$ contains an even number of $a$’s and odd number of $b$’s $\}$  
(Examples: b, aba, aab, bbaa, abaabab)

2. (8 points) Provide a regular expression to capture each language below given the alphabet $\Sigma = \{0, 1, 2, 3\}$

(a) $L_2 = \{w | w$ includes only even numbers or only odd numbers $\}$  
Examples: 222, 11, 02200, 1333  

$$(0U2)^* \cup (1U3)^*$$

(b) $L_3 = \{w | w$ contains exactly one 3 $\}$  
Examples: 3, 123, 00312, 3212  

$$(0U1U2)^*3(0U1U2)^*$$
3. (12 points) Consider the grammar G3
S -> nAnB
A -> Axx | ε
B -> ccBc | d

(a) What are the terminals?

x, n, c

(b) Convert G12 to Chomsky Normal Form.

S -> U_nC | U_nD
A -> AE | U_xU_x
B -> U_cF | d
C -> AD
D -> U_nB U_n->n
E -> U_xU_x U_x->x
F -> U_cG U_c->c
G -> BU_c

(c) Is L(G3) a regular language? Yes No NO

(b) What is the language produced by G4? (This is basically just the B rule from above)
G4: S -> ccSc | d

(cc)^idc^i i>=0
4. (10 points) Consider the DFA M4.

(a) Use a regular expression to describe the language recognized by M4.

\[ x^* y (xx)^* y^* \]

(b) List at least one state that can be removed from an NFA version of this DFA and explain why (e.g., you could draw an equivalent 3-state NFA).

\( q_3 \) can be removed. It is a permanent reject state, and permanent reject states don’t need to be included in NFAs (if you don’t list a transition, it goes to permanent reject by default for NFA).
5. (10 points) Consider the language $L_5 = \{(xyUy)^*(zz)^*(xxUyy)\}$

(a) What is the pumping length of $L_5$?

3 $yxx$ is the shortest pumpable string in the language, and it has length 3

(b) Draw a DFA to recognize $L_6 = \{(aUb)((bb)^*a)^*\}$
6. (6 points) Show that if M is a DFA that recognizes language B, swapping the accept and nonaccept states in M yields a new DFA recognizing the complement of B. Conclude that the class of regular languages is closed under complement.

Provide a GENERAL answer: If M recognizes B, every word in B leads to an accept state in M and every word NOT in B leads to a REJECT state in M. Defining M’ as M but FLIPPING the accept and reject states, every word in B will lead to a REJECT state in M’ (was accept in M) and every word not in B (that means, every word in B’) will lead to an ACCEPT state in M’ (was reject in M). Thus, M’ is a DFA that will accept B’, thus B’ is regular since there is a DFA that accepts it. This works for all regular languages B.

7. (7 points) Consider the NFA N7.

(a) Provide two strings accepted by N7.

00, 10, 0000, 10100000, 00111101

(b) Convert N7 to a DFA.
8. (8 points) The following PDA recognizes the language $0^{2n}1^n$.

Replace the ? with the proper symbols to make the PDA operate as specified.

(Reminder, $a,b\rightarrow c$ says: “When $a$ is next character on input and $b$ is top of stack, pop $b$ and push $c$”)

9. (6 points) Use the pumping lemma to show that the following language is not regular: $L_9 = \{0^k1^{2k} \mid k>0 \}$

Assume $L_9$ is regular. Will have pumping length $p$. $0^p1^{2p}$ is $w$ in $L_9$ with length $\geq p$, so must be pumpable.

Way to divide $w$ is: $x=0^m$ $y=0^n$ $z=0^{p-(m+n)}1^{2p}$ $n>0$

If we pump $w$: $xy^2z = 0^m0^{2n}0^{p-(m+n)}1^{2p} = 0^{p+n}1^{2p}$ -- number of 0’s more than half number of 1’s

Pumped $w$ not in $L_9$, thus $L_9$ not regular language.
10. (6 points) The parenthesis matching problem checks an input and ensures every open parenthesis is matched with a closed parenthesis. (An important thing to check when you are programming!)

Prove that the parenthesis matching problem is a context free language.

**Parenthesis matching is context free if we can write context free grammar to express the set of all strings with balanced parentheses. We can do this! Here it is:**

\[ S \rightarrow SS \mid (S) \mid \varepsilon \]

11. (6 points) Many web services require new account passwords contain at least one number and at least one “special” character (!, *, +, -, $). Prove that checking whether an account password is “acceptable” falls under the set of regular languages.

**Password rule checking is regular if we can make a FSM to recognize it or a regular expression to describe it. We can do that! Here is the regular expression:**

\[ S = U^* U + U - U \$; \quad N = 0 U 1 U \ldots U 8 U 9; \quad L = \Sigma^* S \Sigma^* N \Sigma^* U \Sigma^* N \Sigma^* S \Sigma^* \]

**Or, here is the FSM:**
12. (8 points)
A = \{apple, orange, pear\} \quad B = \{1, 2, 3, 4\} \quad C = \{red, blue, green\}

What is \(B \cdot C \cdot A\) ?
\{1\text{redapple}, 1\text{redorange}, 1\text{redpear}, 2\text{redapple}, \ldots 3\text{bluepear}, \ldots 3\text{greenorange}, \ldots 4\text{greenpear}\}

What is \((B \cup A)^*\) ?
\{\varepsilon, 1, 2, 3, 4, \text{apple}, \text{orange}, \text{pear}, 11, 12, 13, \ldots, 1\text{orange}, 1\text{pear}, \ldots 32, 33, \ldots 4\text{pear}, 1\text{apple}, 1\text{apple}, 1\text{apple}, 1\text{orange}, 1\text{orange}, \ldots \text{pearpear}, 111, 112, 113, \ldots 1\text{apple}, 2\text{orangeapple}, 34\text{pear}, \text{pearorange}, \ldots \}\