Computer Science 456/656 Fall 2020 Practice Examination October 7, 2018

The entire examination is 430 points.

1. True or False. [5 points each] T = true, F = false, and O = open, meaning that the answer is not known to science at this time.

   Review the 3 page list of true false questions on the web page.

2. [20 points] State the pumping lemma for regular languages. (Your answer must be correct in its structure, not just the words you use. Even if all the correct words are there, you could get no credit if you get the logic wrong.)

   I will write the quantifiers in boldface, for emphasis.

   For any regular language L
   there is a pumping length p such that
   for any $w \in L$ such that *barred* $\geq p$
   there exist strings x, y, z such that the following conditions hold

   1. $w = xyz$
   2. $|xy| \leq p$
   3. $|y| \geq 1$
   4. **for any** $i \geq 0$ $\; xy^iz \in L$

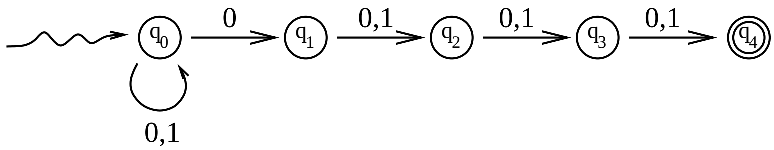
3. [20 points] State the pumping lemma for context-free languages. (Your answer must be correct in its structure, not just the words you use. Even if all the correct words are there, you could get no credit if you get the logic wrong.)

   I will write the quantifiers in boldface, for emphasis.

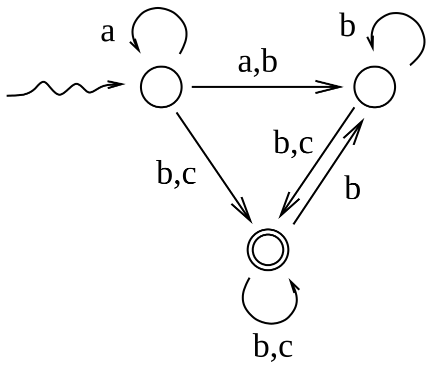
   For any context-free language L
   there is a pumping length p such that
   for any $w \in L$ such that *barred* $\geq p$
   there exist strings u, v, x, y, z such that the following conditions hold

   1. $w = uvxyz$
   2. $|vxy| \leq p$
   3. $|vy| \geq 1$
   4. **for any** $i \geq 0$ $\; uv^ixy^iz \in L$

4. [25 points] Draw an NFA with five states which accepts the language described by the regular expression $(0+1)^*0(0+1)(0+1)(0+1)$



5. [25 points] Write a regular expression for the language accepted by the following NFA. If your answer is unnecessarily long by a wide margin, I might mark it wrong even if it's right.



Concatenate a general path from start to the final state with the Kleene closu
of a general formula for looping at the final state,
$(a^*(b+c+(a+b)b^*(b+c)(b+c+bb^*)b+c)^*$

6. [20 points] Let G be the context-free grammar given below.

$S \to a$

$S \to wS$

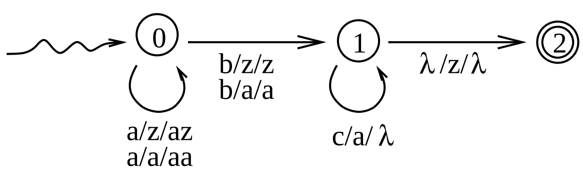
$S \to iS$

$S \to iSeS$

Prove that G is ambiguous by writing two different **leftmost** derivations for the string $iwiaea$. [If you simply show two different parse trees, you are not following instructions.]

$S \Rightarrow iSeS \Rightarrow iwSeS \Rightarrow iwiSeS \Rightarrow iwiaeS \Rightarrow iwiaea$
$S \Rightarrow iS \Rightarrow iwS \Rightarrow iwiSeS \Rightarrow iwiaeS \Rightarrow iwiaea$

7. [30 points] Design a PDA that accepts the language $L = \{a^n b c^n \ : \ n \geq 0\}$.



8. [30 points] Give a context-free grammar for the language of all strings over $\{0, 1\}$ of the form $0^m 1^n$ where $n \neq m$.

$S \to 0S_1$
$S \to S_2 1$
$S_1 \to 0S_1$
$S_1 \to 0S_1 1$
$S_1 \to \lambda$

9. [30 points] The following context-free grammar G is ambiguous. Give an equivalent unambiguous grammar. **(This problem will not appear on Test 2.)**

1. $E \to E\ +\ E$	1. $E \to E\ +\ T$	There are many solutions, but we want a grammar which preserves the standard precedence rules. Let the variables E, T, F stand for *expression*, *term* and *factor*. An expression is one or more terms connected by $+$ and $-$, a term is one or more factors connected by $*$, while a factor is a variable, a parenthesized expression, or the negation of a factor.
2. $E \to E\ -\ E$	2. $E \to E\ -\ T$	
3. $E \to E\ *\ E$	3. $E \to T$	
4. $E \to -\ E$	4. $T \to T\ *\ F$	
5. $E \to (\ E\ )$	5. $T \to F$	
6. $E \to a$	6. $F \to -F$	
7. $E \to b$	7. $F \to (\ E\ )$	
8. $E \to c$	8. $F \to a$	
	9. $F \to b$	
	10. $F \to c$	

10. [30 points] Let L be the language generated by the Chomsky Normal Form (CNF) grammar given below. Use the CYK algorithm to prove that the string $a(a+a)$ is a member of L. Use the figure below for your work.

(a) $S \to a$

(b) $E \to a$

(c) $S \to LA$

(d) $E \to LA$

(e) $L \to ($

(f) $A \to ER$

(g) $R \to)$

(h) $S \to PE$

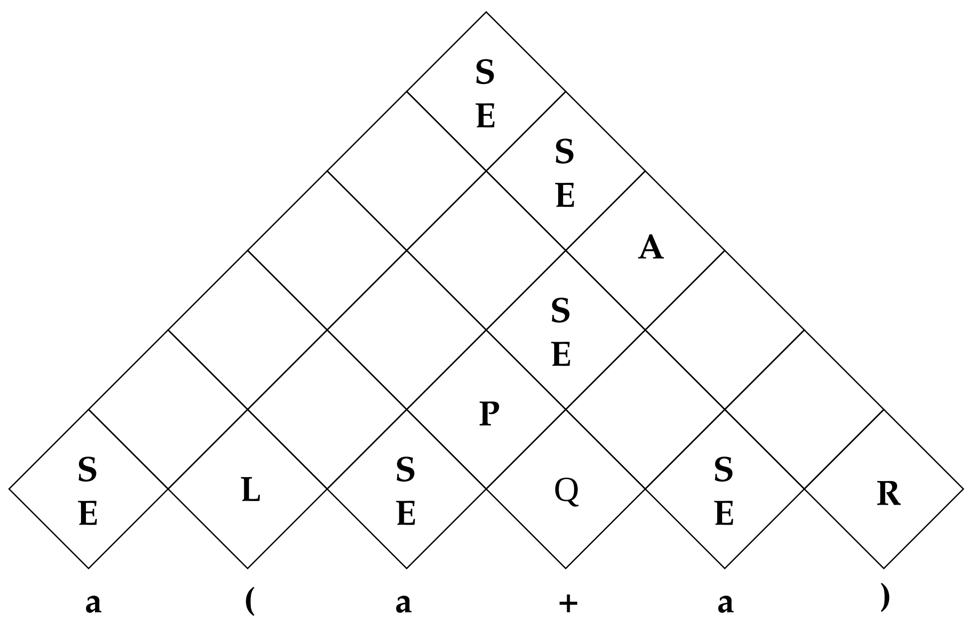
(i) $E \to PE$

(j) $S \to EE$

(k) $E \to EE$

(l) $P \to EQ$

(m) $Q \to +$



11. [30 points] Consider the NFA whose transition diagram is in Figure 1 below. where the input alphabet is $\{a, b, c\}$. Draw the transition diagram of an equivalent minimal DFA. Show your steps.

This was covered adequately on the first test.

12. [30 points] Let $L = \left\{w \in \{a,b\}^* \mid \#_a(w) = 2\#_b(w)\right\}$, here $\#_a(w)$ denotes the number of instances of the symbol a in the string w. For example, $aaababaaabba \in L$, because that string has the twice as many a's as b's. Give a context-free grammar for L. Your grammar may be ambiguous.

How to work this problem. Let $w \in L$. We ask, how can the grammar generate w?

Let $A = \left\{w \in \{a,b\}^* \mid \#_a(w) = 2\#_b(w) + 1\right\}$,

(a) If $w = \lambda$, use the production $S \to \lambda$. Otherwise, assume $w \neq \lambda$.

(b) If w has a proper prefix in L, use $S \to SS$. Otherwise, assume w has no proper prefix in L.

(c) If w begins with aa and ends with b, use the production $S \to aaSb$

(d) If w begins with bb and ends with a, use the production $S \to bSaa$

(e) If w begins with a and ends with a, find a prefix $u \in A$ and a suffix $v \in A$ such that $w = ubv$. Use the production $S \to aSbSa$.

Our grammar is: $S \to SS \mid aaSb \mid bSaa \mid aSbSa \mid \lambda$

Fill in the blank. If L_1 is undecidable and if R is a reduction of L_1 to L_2 and if R is **recursive**, then L_2 is undecidable.

13. [20 points] Use the pumping lemma for context-free languages to prove that the language $L = \{a^n b^n c^n\}$ is not context-free.

Suppose L is context-free. Pick a positive pumping length p. Let $w = a^p b^p c^p$. Then $w = uvxyz$ where u, v, x, y, z satisfy the conditions given by the pumping lemma. Then $uxz \in L$. Since $|vxy| \le p$, vxz cannot contain all three symbols. hence has either no a's or no c's. Since $|vy| \ge 1$, $|uxz| < 3p$. If uxz has no a, then $\#_a(uxz) = p$, contradiction. If uxz has no c, then $\#_c(uxz) = p$, contradiction.

14. [20 points] Prove that a language is recursively enumerable if and only if it is accepted by some machine.

Suppose L is R.E. Let $w_1, w_2, \ldots$ be a recursively computed enumeration of L. The following program accepts L.

. Read w
. For (i from 1 to ∞, unless there is a HALT)
. If($w_i = w$) HALT

Conversely, suppose there is a machind M which accepts L. The following program enumerates L.

Let $w_1, w_2, \ldots$ be a canonical order enumeration of Σ^*.

. For t = 1 to ∞
. For $i = 1$ to t
. If M accepts w_i within t steps, WRITE w_i.

15. [20 points] Prove that the halting problem is undecidable.

Suppose HALT is decidable. Let $L_{\text{diag}} = \{\langle M\rangle : \langle M\rangle\langle M\rangle \notin \text{HALT}\}$, the diagonal language.. Since HALT is decidable, L_{diag} is decidable. Let M_{diag} be a machine which decides L_{diag}.

By the definition of M_{diag}: $\langle M_{\text{diag}}\rangle\langle M\rangle \in \text{HALT}$ if and only if $\langle M\rangle \in L_{\text{diag}}$
By the definition of L_{diag}: $\langle M\rangle\langle M\rangle \in \text{HALT}$ if and only if $\langle M\rangle \notin L_{\text{diag}}$
By universal instantiation, replacing each $\langle M\rangle$ by $\langle M_{\text{diag}}\rangle$
$\langle M_{\text{diag}}\rangle\langle M_{\text{diag}}\rangle \in \text{HALT}$ if and only if $\langle M\rangle \in L_{\text{diag}}$

$\langle M_{\text{diag}}\rangle\langle M_{\text{diag}}\rangle \in$ HALT if and only if $\langle M_{\text{diag}}\rangle \notin L_{\text{diag}}$

Contradiction.

16. [20 points] Prove that the context-free grammar equivalence problem is co-RE.

    Suppose G_1 and G_2 are context-free grammars. If the two grammars are not equivalent, then there is some string w generated by one grammar but not by the other. Let $w_1, w_2, \ldots = \Sigma^*$ in canonical order.
    For i = 1 to ∞
    If $w_i \in L(G_1)$ and $w_i \notin L(G_2)$ HALT   If $w_i \in L(G_2)$ and $w_i \notin L(G_1)$ HALT

17. [20 points] Give a definition of each of these $\mathcal{NP}$-complete languages/problems.

    (a) SAT All satisfiable Boolean expressions.

    (b) 3-SAT **not covered yet**

    (c) Independent Set **not covered yet**

    (d) Subset Sum **not covered yet**

    (e) Partition **not covered yet**

The remaining problems deal with material I have not covered, and hence will not be on the examination.

18. [20 points] Give a general (unrestricted) grammar for the language consisting of all strings of 1's of length a power of 2, that is, $\left\{1^{2^n}\right\}$

19. [20 points] Give one of these polynomial time reductions (your choice).

    (a) 3-SAT to Independent Set.

    (b) Independent Set to Subset Sum

    (c) Subset Sum to Partition