ECS 120 Final Exam

Instructions:
Read the problems carefully and ask questions if anything is unclear. Make sure you understand what a problem asks of you before writing the solution.
You can use as known (i.e. refer to) all problems we have solved in class or homeworks.
Please answer the questions succinctly and thoughtfully. Use the back of the pages if needed.
Good luck!

Name: KEY
ID: KEY

<table>
<thead>
<tr>
<th>On section</th>
<th>you got</th>
<th>out of</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td>12</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>18</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td>24</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>15</td>
</tr>
<tr>
<td>5</td>
<td></td>
<td>15</td>
</tr>
<tr>
<td>6</td>
<td></td>
<td>16</td>
</tr>
<tr>
<td>Σ</td>
<td></td>
<td>100</td>
</tr>
</tbody>
</table>
1. (2 pts.) Finish the definition:

A pushdown automaton is a 6-tuple . . .

\[(Q, \Sigma, \Gamma, \delta, q_0, F)\], where \(Q, \Sigma, \Gamma, F\) are finite sets and

1. \(Q\) is a set of states
2. \(\Sigma\) is the input alphabet
3. \(\Gamma\) is the stack alphabet
4. \(\delta: Q \times \Sigma \times \Gamma \rightarrow P(Q \times \Gamma),\) transition function
5. \(q_0 \in Q\)
6. \(F \subseteq Q\)

2. (3 pts.) What does it mean for a decision problem \(A\) to be reducible to decision problem \(B\)?

\[A \text{ is not more difficult to solve than } B.\]

3. (4 pts.) Finish the definition:

A language \(L\) is \(NP\)-complete if . . .

\[(1) \ L \in NP, \text{ and} \]
\[(2) \ \nexists \ L' \in NP, \ L' \leq_p L\]

4. (3 pts.) The Church-Turing Thesis states that . . .

Effectively computable \(\iff\) TM computable
2 Short Problems

1. (9 pts.) Using the pumping lemma for CFLs show that the language
   \( L = \{0^i1^j2^i3^j \mid i \geq 1 \text{ and } j \geq 1 \} \) is not context-free.

   Let \( L \) be context free. Then, let \( p \) be the PL constant and
   pick \( w = 0^p1^p2^p3^p \). Clearly, \( w \in L \) \( |w| = 4p > p \). Then, \( w = uvxyz \) is
   possible, subject to \( |vxy| \leq p \) and \( |vx| > 0 \). But then \( vxy \) is
   either contained in the substring of one symbol (i.e. 0, 1, 2, or 3)
   or it straddles two adjacent symbols. Then,
   - if \( vxy \) consists of only one symbol \( u \), \( v \) and \( y \) \( z \) being in \( L \) gives
     a contradiction since it has more \( 0 \) that symbol than \( 2 \).
   - if \( vxy \) straddles two symbols, say 1 \& 2. Then, \( uxz \in L \) either
     has fewer 1's than \( p \) or fewer 2's than \( p \) or both. Either
     way \( uxz \in L \) since \( u \), z has exactly \( p \), 3's and \( p \), 0's.

2. (9 pts.) Describe a procedure that accepts the following language:
   \( L = \{\langle M \rangle \mid M \text{ is a TM that accepts some odd length string } \} \).

   A non-deterministic TM \( N \):

   \( N = " \) On input \( \langle M \rangle \)
   1. Guess non-deterministically an odd-length string \( x \) in \( L(M) \)
   2. Simulate \( M \) on \( x \).
   3. Accept if \( M \) accepts. "
3 Justified True or False

Put an $\times$ through the correct box. Provide a brief (but convincing) justification. No credit will be given to correct answers that lack a proper justification. Where appropriate, make your justification a counter-example. Each question in this section is worth 4 points.

1. If $A$ and $B$ are recursively enumerable languages then so is $A \setminus B$.

   Explain:
   
   r.e. languages are not closed under complement.

2. If $L$ is not co-r.e. then $L$ is not a regular language.

   Explain:
   
   $L \notin \text{co-r.e.} \Rightarrow L \notin \text{recursive} \Rightarrow L \notin \text{Regular.}$

3. $n^{\log_2 n} = O(n^k)$

   Explain:
   
   $\log_2 n$ is not bounded by any constant $k$.
   Thus
   
   $n^{\log_2 n} > n^k$, for some $n$. 

4. If \( A \) is a regular language and \( A \leq_p B \) then \( B \in \mathcal{P} \).

   **True**   **False**

   Explain:
   Although regular languages are decidable in poly-time, \( B \) need not be in \( \mathcal{P} \). The existence of \( \leq_p \) just means that \( A \) can be decided possibly in \( \mathcal{NP} \) or \( \mathcal{EXP} \) time.

5. If \( L_1 \subseteq L \subseteq L_2 \) and \( L_1 \) and \( L_2 \) are regular then \( L \) is Turing acceptable.

   **True**   **False**

   Explain:
   \[
   L_1 = \emptyset \\
   L_2 = \Sigma^* \\
   L = \overline{A_{TM}}
   \]

6. If \( SAT \) is NP-complete and \( SAT \leq_p L \) then \( L \) is NP-complete.

   **True**   **False**

   Explain:
   If \( L \) is not in \( NP \), the statement is false.
4 A Decision Procedure

[14 points]

You are given some CFG $G$ over the English alphabet, and are curious to find out if $G$ can derive your first name. Describe a decision procedure that does that. (You can assume that you have procedures that can convert PDAs into CFGs and CFGs into PDAs).

let your name be given by the string $w$.
Then, the decision procedure is as follows.

1. Treat $w$ as a regular expression and convert it to an equivalent DFA $W$.

2. Convert $G$ to an equivalent PDA $P$.

3. Construct the automaton $P'$ for the language $L(P) \cap L(W)$ (using the procedure we studied in class).


5. Check $L(G')$ for emptiness (using the procedure studied in class).

- if $L(G')$ is empty $\Rightarrow$ your name is not derived by $G$.

- if $L(G')$ is not empty $\Rightarrow$ your name is derived by $G$. 
5 A Reduction

We showed in class that \( E_{TM} = \{(M) | M \text{ is a Turing machine and } L(M) = \emptyset \} \) is undecidable. In this problem you will determine where exactly in the Chomsky Hierarchy to place this language.

(a) [3 pts.] A warm-up question: if \( E_{TM} \) is not acceptable can \( \overline{E_{TM}} \) be acceptable? Why, or why not?

Sure it can. L.E. languages are not closed under complement.

(b) [6 pts.] Prove that \( \overline{E_{TM}} \) is acceptable.

\[ \overline{E_{TM}} = \{(M) | \text{M is a TM and } L(M) \neq \emptyset \} \]

A non-deterministic TM \( N \):

1. Non-deterministically guess a string \( x \), \( x \in L(M) \).
2. Simulate \( M \) on \( x \).
3. Accept if \( M \) accepts.

(c) [6 pts.] Prove that \( E_{TM} \) is not acceptable. (Note: to show this you can use part (b), but only if you've proven it correctly. Otherwise use a reduction.)

---

Note: The reduction proof is correct and as follows:

Proof 2: We reduce \( \overline{Atm} \) to \( E_{TM} \) or equivalently \( \overline{Atm} \leq_{m} E_{TM} \). The following TM \( F \) computes the mapping \( \leq_{m} \):

\[ F = \"\text{On input } \langle M, w \rangle \"

1. Construct the following TM \( M_{\phi} = \"\text{On input } x \"

   - if \( x \neq w \), reject
   - if \( x = w \), run \( M \) on \( w \)
   - accept if \( M \) does.

2. Output \( \langle M_{\phi} \rangle \).

This works because of the same reasons the informal \( \overline{Atm} \) to \( E_{TM} \) reduction works (in your book.)
6 NP-completeness

Let $V$ be a set of $\{0,1\}$ variables, and $I$ a set of linear inequalities over $V$. A solution to $I$ is an assignment of 0's and 1's to all $v_i \in V$ such that all the inequalities are true.

For example, $I = \{v_1 \geq 1, v_2 \geq 0, v_1 + v_2 \leq 3\}$ is a set of three inequalities over $V = \{v_1, v_2\}$, with solution $v_1 = 1$ and $v_2 = 1$.

Show that the following problem of solving integer inequalities is NP-complete:

$\text{INT\_INEQ} = \{(V, I) | V \text{ and } I \text{ are sets as above and there exists a solution to } I\}$

Hint: Reduce 3SAT to $\text{INT\_INEQ}$ by following the directions below.

(a) [3 pts.] Let TRUE correspond to 1 and FALSE to 0. For each variable in 3SAT construct two inequalities constraining a literal and its complement to have different values. One of them will be $v_i + \overline{v_i} \geq 1$. What’s the other inequality and why?

The other inequality is: $v_i + \overline{v_i} \leq 1$.

Together, $v_i + \overline{v_i} \geq 1$ and $v_i + \overline{v_i} \leq 1$ enforce the property that either ($v_i = 1$ and $\overline{v_i} = 0$) or ($v_i = 0$ and $\overline{v_i} = 1$).

(b) [4 pts.] For each clause in the 3SAT instance construct an inequality, which in order to be satisfied will require at least one variable to be 1. Finish the construction and calculate the total number of inequalities.

For each clause $C, C = a \lor b \lor c$, we construct the inequality $a + b + c \geq 1$. This will ensure that at least one of $\{a, b, c\}$ is one.

If $\phi$ has $n$ variables and $k$ clauses we'll have a total of $2n$ variable inequalities (from a) and $k$ clause ineq. for a total of $2n+k$ inequalities.
(c) [4 pts.] Show that any 3SAT instance gives an instance of INT_INEQ.

If $\phi \in 3\text{-SAT}$ there is an assignment of $\{T, F\}$ for the variables in $\phi$ which makes each of its clauses evaluate to $T$. Thus, each clause must contain a literal which is assigned $T$. Then, that assignment which makes $\phi \in 3\text{sat}$ also is a solution for the $2n+k$ integer inequalities constructed above as:
- each of the $2n$ variable inequalities hold since $v_i + \overline{v_i} = 1$.
- each of the $k$ clause inequalities hold because $\exists$ a $T$ literal in each clause, so the sum of the clause lit $\geq 1$.

(d) [5 pts.] Show that any INT_INEQ instance gives an instance of 3SAT.

If the $2n+k$ integer inequalities have a solution over $\{0, 1, \overline{1}\}$ then that solution is a satisfying assignment for $\phi$ (with $0 = F$ and $1 = T$) because there must be one variable in each of the $k$ clause inequalities which is 1. In addition, since $v_i + \overline{v_i} = 1$ a variable and its complement in $\phi$ cannot both have the same value. Both statements above together imply that there are variables (corresponding to literals) for each clause in $\phi$ that are $T$, thus satisfying the clauses and $\phi$. 