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

On section	you got	out of
1		12
2		18
3		24
4		15
5		15
6		16
Σ		100

1 Theory [12 points]

1. (2 pts.) Finish the definition:

A **pushdown automaton** is a 6-tuple . . .

$(Q, \Sigma, \Gamma, \delta, q_0, F)$, Q, Σ, Γ, F are finite sets and

1. Q is a set of states
2. Σ is the input alphabet
3. Γ is the stack alphabet
4. $\delta : Q \times \Sigma_\varepsilon \times \Gamma_\varepsilon \to P(Q \times \Gamma_\varepsilon)$, trans. func
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 is not more difficult to solve than B.

3. (4 pts.) Finish the definition:

A language L is ***NP-complete*** if . . .

(1) $L \in NP$, and

(2) $\forall L' \in NP$, $L' \leq_p L$

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

Effectively computible $\Leftrightarrow$ TM computible

2 Short Problems [18 pts.]

1. (9 pts.) Using the pumping lemma for CFLs show that the language $L = \{0^i 1^j 2^i 3^j | i \geq 1 \text{ and } j \geq 1\}$ is not context-free.

Let L be context free. Then, let p be the PL constant and pick $w = 0^p 1^p 2^p 3^p$. Clearly, $w \in L$ & $|w| = 4p \geq p$. 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 say 0 $uv^i xy^i z$ being in L gives a contradiction since it has more of that symbol than of 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 $uxz \notin L$ since uxz has exactly p, 3's and p, 0's.

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

A non-deterministic TM N:

N = " On input ⟨M⟩

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 [24 points]

Put an **X** 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 $A \setminus B$.

True **False**

Explain:

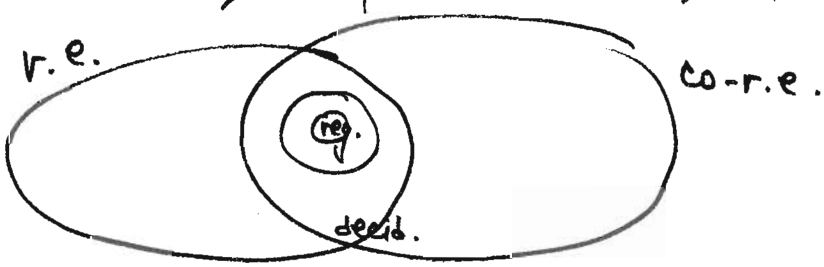
r.e. languages are not closed under complement.

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

True **False**

Explain:

$L \notin$ co-r.e $\Rightarrow L \notin$ recursive $\Rightarrow L \notin$ Regular.



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

True **False**

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 reg. languages are decidable in poly-time, B need not be in P. The existance of $\leq_p$ just means that A can be decided possibly in NP or EXPTIME.

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).
4. Convert P' into an equivalent grammar G'.
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 [15 pts.]

We showed in class that $E_{TM} = \{\langle M\rangle | M \text{ is a Turing machine and } L(M) = \emptyset\}$ 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. r.e. languages are not closed under complement.

(b) [**6 pts.**] Prove that $\overline{E_{TM}}$ is acceptable. $\overline{E_{TM}} = \{\langle M\rangle | M \text{ is a TM and } L(M) \neq \emptyset\}$

A non-deterministic TM N:

N = "On input ⟨M⟩,

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.)

Proof 1 Since E_{TM} is undecidable and $\overline{E_{TM}}$ is acceptable, E_{TM} cannot be acceptable (If it were then E_{TM} would be decidable).

Proof 2 We reduce $\overline{A_{TM}}$ to E_{TM} or equivalently $A_{TM} \leq_m \overline{E_{TM}}$. The following TM F computes the mapping $\leq_m$:

F = "On input ⟨M, w⟩

1. Construct the following TM
   M_1 = "On input x
   - if $x \neq w$, reject
   - if $x = w$, run M on w
     - accept if M does."
2. Output ⟨M_1⟩."

This works because of the same reasons the informal A_{TM} to E_{TM} reduction works (in your book.)

6 NP-completeness [16 pts.]

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:

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

Hint: Reduce $3SAT$ to INT_INEQ by following the directions bellow.

(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, $C = a \vee b \vee c$, we construct the inequality $a + b + c \geq 1$. This will ensure that at least one of $\{a, b, c\}$ is one.
If ϕ 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-SAT the $\exists$ an assignment of $\{T, F\}$ for the variables in ϕ 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 $\phi \in$ 3SAT also is a solution for the $2n+k$ integer inequalities consructed above as:

- each of the $2n$ variable ineq. hold since $v_i + \bar{v}_i = 1$.
- each of the K clause ineq. hold because $\exists$ a ~~variable~~ T literal in each clause, so the sum of the clause lit ≥ 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 $\{0,1\}$ ~~than~~ that solution is a satisfying assignment for ϕ (with $0 \equiv F$ and $1 \equiv T$) because there must be one variable in each of the K clause inequalities which is 1. In addition, since $V_i + \bar{V}_i = 1$ a variable and its complement in ϕ cannot both have the same value.

Both statements above, together, imply that there are variables (corresponding to literals) for each clause in ϕ that are T, thus satisfying the clauses and ϕ.