ECS 120 Final – Winter 1998

Hints for success:

Read the questions. Maybe they ask something different from what you expect! If you don't understand what a question means, please ask me.

Please make your writing legible, logical, and succinct. Definitions and theorem statements should be complete and rigorous to receive credit.

Watch the newsgroup to see when your final exam scores are ready. If you want to look over your final, check with me near the beginning of Spring quarter.

Have a good break!

—Phil Rogaway

Name:

Signature:

On problem	you got	out of
1		30
2		40
3		20
4		30
5		30
6		30
Σ		180

1 Recall [30 points]

---

1. *Complete the following definition:*

A function $f : \{0,1\}^* \rightarrow \{0,1\}^*$ is ***Turing-computable*** if ...

---

2. *Complete the following definition:*

A language L is in the class NP if ...

---

3. A Turing machine M is a 7-tuple $M = (Q, \Sigma, \Gamma, \delta, q_0, q_{\text{accept}}, q_{\text{reject}})$ where $\cdots$

---

4. Using the procedure given in class and in the book, convert the following regular expression into an NFA:

$\alpha = (0 \cup 11)^*.$

---

5. Exhibit a CFG for the language

$$L = \{w \in \{1, 2\}^* : \; w \text{ has twice the number of 1s as 2s}\}.$$

(For example, L contains ε, 111122, 211, and 212121111.)

---

6. Complete the statement of the **pumping lemma** for **context free languages**:

Theorem. If a language L is context free then there exists a number K such that ...

---

2 Justified True or False [40 points]

Put an **X** through the **correct** box. When it says "Explain:" provide a **brief** (but convincing) justification. *Where appropriate, make this justification a counterexample.*

---

1. Suppose L is a language for which every finite subset of L is decidable. Then L is decidable. **True** **False**

Explain:

---

2. If $A \leq_P B$, and $B \in \text{NP}$, then $A \in \text{NP}$. **True** **False**

Explain:

---

3. If C_1 and C_2 are context free, then $C_1 \cap C_2$ is context free.

Explain: **True** **False**

---

4. Let $M = (Q, \Sigma, \delta, q_0, F)$ be an NFA that accepts L. Then there is an NFA with $2|Q|$ states that accepts L^R (the reversal of L). **True** **False**

Explain:

---

5. The language $L = \{1^{a_1}\#1^{a_2}\#\cdots\#1^{a_n} :\ a_i \neq a_{i+1} \text{ for some } 1 \leq i < n\}$ is accepted by some PDA.

Explain: **True** **False**

---

6. Every regular language is accepted by some DFA with an equal number of final and non-final states. **True** **False**

Explain:

---

7. The name "NP" stands for *Not Polynomial* — since some languages in NP require *more* than any polynomial amount of time. **True** **False**

Explain:

---

8. $\{\langle G \rangle : G \text{ is an unrestricted grammar and } L(G) \neq \emptyset\}$ is decidable.

Explain: **True** **False**

---

3 Language Classification. [20 points]

Classify as:

decidable	decidable
r.e.	Turing-acceptable but not decidable
co-r.e.	co-Turing-acceptable but not decidable
neither	neither Turing-acceptable nor co-Turing-acceptable

No explanation is required.

---

1. $\{\langle M \rangle : M$ is a TM and $L(M)$ is Turing decidable$\}$

2. $\{\langle M \rangle : M$ is a TM which accepts some string of even length$\}$

3. $\{\langle G \rangle : G$ is a CFG and $L(G) = \emptyset\}$

4. $\{\langle G_1, G_2 \rangle : G_1$ and G_2 are CFGs and $L(G_1) = L(G_2)\}$

4 Mapping Reductions [30 points]

Let $L = \{\langle M_1, M_2\rangle : \ M_1$ and M_2 are TMs and $L(M_1) \subseteq L(M_2)\}$.

Part A. Prove that $A_{\mathrm{TM}} \leq_{\mathrm{m}} L$.

Part B. Prove that $\overline{A_{\mathrm{TM}}} \leq_{\mathrm{m}} L$.

5 NP-Completeness [30 points]

Let $D = \{\langle p \rangle : p$ is a polynomial in several variables having an integral root$\}$. Prove that *3SAT* $\leq_{\mathrm{P}} D$.

6 Cook-Levin Theorem [30 points]

*Complete the following "narrative" which describes the main idea in the proof of the Cook-Levin theorem which I presented in class. (**Note:** Don't be intimidated — this is an easy problem if you understood the main ideas in that proof.)*

Let the running time of a TM be measured in the length of the #–terminated prefix of its input. Given a TM $M = (Q, \Sigma, \Gamma, \delta, q_0, q_{\text{accept}}, q_{\text{reject}})$ which runs in time bounded by some polynomial $q(n)$, and given a string $w \in \Sigma^n$, we construct a Boolean formula ϕ such that ϕ is satisfiable if and only if there is some $c \in \{0,1\}^*$ such that M accepts $w\#c$.

To construct ϕ, envisage a computation "tableau" T — an array which is ______ cells wide and ______ cells high. Columns of T are indexed by numbers $i =$ ______ to $i =$ ______, and rows are indexed by numbers $t =$ ______ to $t =$ ______. Inside each cell is a character from the alphabet ______.

Intuitively, the contents of row t represents:

______

The crucial thing is that computation is a "local phenomenon." To know that the table is globally valid it is enough to check that it is valid in each ______ $\times$ ______ "window" of adjoining cells. There are only finitely many possibilities for valid window contents. Exactly what they are can be read off of the transition function δ.

Formula ϕ is constructed so that it is satisfiable if and only if there is a valid, accepting tableau for M which has the characters ______ beginning row $t = 0$. The variables used by ϕ will be of the form $X[i, t, \sigma]$, for each i, t, σ. Thus ϕ has a total of ______ variables. Intuitively, variable $X[i, t, \sigma]$ is `true` iff there is a ______ at cell ______ of the tableau.

Part of the formula ϕ will capture that there is exactly one character in each cell of the tableau. We can translate this into a formula by saying that there is at least one character in each cell, and also that there is at most one character in each cell. The former ("there's at least one character per cell") would give rise to the following Boolean expression, which we will AND with several other Boolean expressions to give us ϕ:

Another conjunct of ϕ would enforce that the bottom row of the tableau ($t = 0$) starts in the manner we have described. Another conjunct of ϕ would enforce that each "window" into the tableau (as described above) is valid. Another conjunct of ϕ would enforce that the machine M eventually accepts (given the right value of c). This last conjunct would look like this: