Name ______________________________ ID No. ______________

There are 200 points possible.

1. (40 pts.) Indicate whether true or false. Beware of guessing:

   correct answer +5pts. incorrect answer −3pts. no answer 0pts

   (a) **T** If L is a regular language, then $\overline{L}$ must be a regular language.

   (b) **F** If L is a context free language, then $\overline{L}$ must be a context free language.

   (c) **T** If L is decidable, then $\overline{L}$ must be decidable.

   (d) **F** If L is Turing recognizable, then $\overline{L}$ must be Turing recognizable.

   (e) **F** Personal computers can recognize languages that Turing machines cannot.

   (f) **T** It is possible to build a "universal simulator"; that is, a Turing machine that, when given the description and input for any Turing machine, will be able to simulate that Turing machine on that input.

   (g) **T** A language L in NP is "NP-complete" if every other language in NP is polynomial time reducible to L.

   (h) **F** We can build a Turing machine that takes as input the description of a 2-stack PDA plus the input to the PDA and decides if the PDA will stop.

2. (45 pts.) Do each of the following or explain why you cannot. If you give an example, explain why it is correct.

   (a) Two regular languages whose intersection is *not* regular.

   Ans. Impossible. Regular languages are closed under intersection.

   (b) Two context free languages whose intersection is *not* context free.

   Ans. One possibility is $L_1 = \{a^n b^n c^k \mid n, k \geq 0\}$ and $L_2 = \{a^n b^k c^k \mid n, k \geq 0\}$. Then $L_1 \cap L_2 = \{a^n b^n c^n \mid n \geq 0\}$. It was shown in the pumping lemma section that this is not a CFL.

   (c) A language which is in NP but is *not* in P.

   Ans. It can't be done at the present time because it is not known if P=NP. If P=NP, the problem is impossible. If P≠NP, any NP-complete language would work.

3. (45 pts.) Let the alphabet be $\Sigma = \{0, 1\}$.
   - Let A be set of strings in Σ^* which contain no adjacent ones.
   - Let B be the set of strings in Σ^* which contain an even number of zeros.
   - Let $C = (A \cap \overline{B}) \cup (\overline{A} \cap B)$.

   For example, 011000 and 100101 are in C but 01100 and 0100101 are not in C.

   (a) Write a regular expression for A.

Ans. One possibility is $0^*(100^*)^* \cup (00^*1)^*0^* \cup 1(00^*1)$.
Another is $0^*(100^*)^*(\epsilon \cup 1)$.

   (b) Prove that B is a regular language.

Ans. It's probably easiest to give a DFA. Let q_{even} be the start state and the accept state. There is one other state called q_{odd}. If you see a zero, change from your current state to the other. If you see a one, stay in your current state.

   (c) Prove that C is a regular language.
   HINT: You may use the results in (a) and (b) even if you have not done those parts.

Ans. Since regular languages are closed under complementation, intersection and union and since A and B are regular, it follows that C is regular.

4. (20 pts.) Construct a context free grammar for the language generated by the regular expression $(a^* \cup b) \circ (ba^*)^*$.

Ans. Here is the simplest solution. The start variable is S. The rules are

$$S \to XY \qquad X \to A \mid b \qquad A \to aA \mid \epsilon \qquad Y \to bAY \mid \epsilon$$

5. (20 pts.) Suppose L and M are Turing-recognizable languages. Prove that $L \cup M$ and $L \cap M$ are Turing-recognizable.

Ans. Let T_L and T_M be Turing machines that recognize L and M. Here is the algorithm for $L \cup M$ (resp. $L \cap M$).

For $k = 1, 2, 3, \ldots$:

Simulate T_L *and* T_M *for* k *steps each.*

If either (resp. both) `accept` *then* `accept`.

If both (resp. either) `reject` *then* `reject`.

Note: The last statement is not necessary.

6. (30 pts.) Recall that EQ_{CFG} is the set of pairs G_1, G_2 of CFGs such that G_1 and G_2 generate the same language. It was remarked on page 158 that EQ_{CFG} is *not* decidable.

(a) Let $EQ_{\mathrm{CFG,n}}$ be the set of pairs G_1, G_2 of CFGs such that the languages generated by G_1 and G_2 contain exactly the same strings of length n. Prove that $EQ_{\mathrm{CFG,n}}$ is decidable. **HINT**: Use Chomsky normal form.

Ans. Suppose w is a string of length n. When a grammar is in Chomsky normal form, production of w requires less than $2n$ applications of rules from the grammar. This tells us that there are only a finite number of possibilities to test to see if G_1 can produce w. Likewise for G_2. Since there are only a finite number of strings of length n, we are done.

(b) What is wrong with the following proof that EQ_{CFG} is Turing decidable?

For each n*, run* $EQ_{\mathrm{CFG,n}}$*. If* $EQ_{\mathrm{CFG,n}}$ *stops with* `reject` *for some* n*, then* `reject`*; otherwise,* `accept`.

Ans. If the grammars are the same, we will never stop with a reject and so we will run forever as n goes through all positive integer values.