AUTO

**Comprehensive Exam:** **Autumn 2000-01**

## Automata and Formal Languages (60 points)

**Problem 1.** [10 points]
Consider the language $L$ defined by the regular expression $00^*10$. Provide a PDA $M$ for this language using *as few states as possible*. Note that there is a PDA with only 1 state and that the number of points you get will depend on the number of states used in your solution.

**Problem 2.** [18 points]
Decide whether the following statements are TRUE or FALSE. *You will receive 3 points for each correct answer and -2 points for each incorrect answer.*

1. If $L_1$ and $L_2$ are both non-regular, then $L_1 \cap L_2$ must be non-regular.
2. $L = \{w \in \{a, b, c\}^* \mid w$ does not contain an equal number of occurrences of $a$, $b$, and $c\}$ is context-free.
3. Let $L$ represent the language of a non-deterministic finite-state automaton $N$; then, swapping the final and non-final states of $N$ gives a machine $N'$ whose language is the complement of $L$.
4. Assume that P $\neq$ NP. If $L_1$ is in P and $L_2$ is in NP, then $L_1 \cap L_2$ must be in $P$.
5. If $L_1$ and $L_2$ are both in NP, then $L_1.L_2$ must be in NP.
6. If $L_1$ is context-free and $L_2$ is NP-complete, then $L_1 \cup L_2$ must be NP-complete.

**Problem 3.** [12 points]
Classify each of the following languages as being in one of the following classes of languages: *empty, finite, regular, context-free, recursive, recursively enumerable.* You must give the *smallest* class that contains *every possible language* fitting the following definitions. For example, the language of a DFA $M$ could be *empty* or *finite*, and must always be *context-free*, but the smallest class that is appropriate is *regular*.

1. The language $L = \{a^i b^j c^k d^l \mid i = k \text{ and } j = l\}$.
2. The set of strings from $\{0, 1\}^*$ which, when viewed as integers written in binary, are divisible by 3.
3. The language of a non-deterministic finite state automaton (NFA) with only two states.
4. The language of a non-deterministic push-down automaton (NPDA) with only one

5. The complement of a language $L$ that belongs to P (polynomial time) but is not context-free.

6. A language $L$ to which we can give a polynomial-time reduction from an undecidable language.

**Problem 4.** [10 points]
Using a reduction from a known undecidable problem, prove that the following problem is undecidable: Determine whether a Turing machine $M$ halts on all inputs from $\{0, 1\}^*$ that represent a valid encoding of some Turing machine. (You may assume any standard scheme for encoding a Turing machine into a string of 0's and 1's.)

**Problem 5.** [10 points]
Recall the decision problems called 2-SAT and 3-SAT. These are the versions of the satisfiability problems for 2-CNF and 3-CNF boolean formulas, respectively.

**a).** Prove that 2-SAT is polynomial-time reducible to 3-SAT. (Describe a reduction and justify its correctness.)

**b).** Given that 3-SAT is NP-complete, is the result in part (a) sufficient to prove the NP-completeness of 2-SAT? Explain.