CS 172 — Spring 2000	**Final Exam**
Computability and Complexity	**May 19, 2000**

Problem 1. (100 points) Given a word w, the *stutter reduction* $[w]$ is the word that results from w by deleting repeated adjacent occurrences of the same letter. For example, $[aabcccbbabbbbb] = abcbab$. Given a language A, let $[A] = \{[w] \mid w \in A\}$ be the set of stutter reductions of words in A. If A is a regular language, does it necessarily follow that $[A]$ is also regular? Prove your answer.

Problem 2. (100 points) Let B_1 be the set of quantified boolean formulas whose operators are taken from the set $\{\wedge, \vee, \neg\}$ (arbitrarily nested) and whose variables are letters from the set $V_1 = \{x, y, z\}$. We require that every variable is bound by a quantifier. For example, $(\forall x)(\exists y)(x \vee y)$ is in B_1, whereas $(\forall x)(x \vee y)$ is not. You may assume that all quantifiers occur at the beginning of a formula, and you are free to choose the precise syntax of formulas (where to put parentheses etc.). Let B_2 be the set of quantified boolean formulas whose variables are words from the set $V_2 = \{x, y, z\}^*$. For example, $(\forall xx)(\exists xyx)(xx \vee xyx)$ is in B_2 (note that xx is one variable, and xyx is another one). Unlike the formulas in B_1, the formulas in B_2 have an unlimited supply of variables. Is B_1 context-free? What about B_2? Prove your answers.

Problem 3. (100 points) Let C_1 be the set of all pairs $\langle M, w \rangle$, where M is a deterministic Turing machine whose computation on input w visits at most half of the non-blank tape cells (i.e., the machine M accepts, rejects, or loops without moving past the midpoint of the input w). Let C_2 be the set of all pairs $\langle M, w \rangle$, where M is a deterministic Turing machine whose computation on input w visits at most half of the states of M. Is C_1 recursive or r.e. or co-r.e. or neither? What about C_2? Prove your answers.

Problem 4. (100 points) A regular expression is *star-free* iff it does not contain the * operator. Prove that the following language is NP-complete:

$$\overline{\text{StarfreeUniversality}} = \{\langle R, k \rangle \mid R \text{ is a star-free regular expression and } k \text{ is a nonnegative integer and } L(R) \neq \{\texttt{0}, \texttt{1}\}^k\}$$

Here $\{\texttt{0}, \texttt{1}\}^k$ is the language that contains all words of length k with letters from $\{\texttt{0}, \texttt{1}\}$. To prove containment in NP, give a certificate and show that it can be verified in polynomial time. To prove hardness for NP, reduce 3Sat to $\overline{\text{StarfreeUniversality}}$ in polynomial time.

Hint: Think of the truth-value assignment that assigns *false* to all variables as the word $\texttt{00}\ldots\texttt{0}$, and the truth-value assignment that assigns *true* to all variables as the word $\texttt{11}\ldots\texttt{1}$. Given a 3cnf formula ϕ, construct a star-free regular expression R and an integer k so that the truth-value assignments that satisfy ϕ correspond to words of length k that are rejected by R, and the truth-value assignments that do not satisfy ϕ correspond to words of length k that are accepted by R.