Name: ____________________________________________________________

Instructions:

1. This is a closed book exam. Do not use any notes or books. Do not confer with any other student.

2. Show your work. Partial credit will be given. Grading will be based on correctness, clarity and neatness.

3. I suggest that you read the whole exam before beginning to work any problem. Budget your time wisely—according to the point distribution.

4. There are 5 questions worth a total of 100 points, on 6 pages (including this page).

DO NOT BEGIN THE EXAM UNTIL INSTRUCTED TO DO SO. GOOD LUCK!
1. (20 pts total) Provide brief answers to the following questions:
   (a) (5 pts) What is the recursive definition of the set of regular languages?

   (b) (5 pts) True or False: If language $L_1$ is not regular and is a subset of language $L_2$, then $L_2$ is not regular. Explain.

   (c) (5 pts) True or False: If $L_1$ and $L_2$ are regular languages, then $L_1 \cap L_2$ is regular. Explain.

   (d) (5 pts) Write a regular expression to represent the set of all legal variable names in hypothetical programming language $X$, in which a variable name can consist of upper case letters (A to Z), lower case letters (a to z), digits (0 to 9), and underscore (_), but cannot start with a digit.
2. (20 pts) Let $F$ be the set of all functions $f$ from the natural numbers to the natural numbers such that $n < f(n) < 2n + 2$, for all natural numbers $n$.

For example, $f(n) = 2n$ is in $F$, but $f(n) = 3n$ is not in $F$.

Prove using diagonalization that $F$ is uncountably infinite.
3. (15 pts) Let $\Sigma = \{0, 1\}$. Consider the language $L$ over $\Sigma$ defined recursively as follows:

- **Basis step:** $\lambda$ is in $L$.
- **Recursive step:** If $w$ is in $L$, then $0w1$ is also in $L$.

For example, the string $000111$ is in $L$.

Prove that every string $w$ in $L$ is of the form $0^i1^i$ for some $i \geq 0$. *Use induction on the number of recursive steps applied to create $w$.*
4. (20 pts total) Consider the following NFA-λ \( M \).

(a) (5 pts) Give a regular expression for \( L(M) \). (You do not have to use the conversion algorithm.)

(b) (10 pts) Use the conversion algorithm to come up with a DFA \( M' \) that is equivalent to \( M \).
*Hint:* It should have 5 states.

(c) (10 pts) Use the conversion algorithm on your \( M' \) from part (b) to come up with a regular grammar that generates \( L(M) \).
5. (20 pts) Use the pumping lemma for regular languages to show that the following language is not regular:

\[ L = \{ww : w \in \{0, 1\}^*\} \]

For example, 01010101 is in \( L \), but 01011010 is not.