Guidelines:

Please submit your assignment to the TA/lecturer at the beginning of the appropriate lecture. Solutions should be typed, drawings can be done by hand.

You are encouraged to work in groups of up to three people, however we ask that you dedicate enough time to think about each problem by yourself before consulting others. Moreover, you must write up each problem solution by yourself without assistance. You are asked to identify your collaborators on problem sets. If you did not work with anyone, please write “Collaborators: none”.

You may consult the course notes and the text (Sipser). When this problem set references a section or a problem, we mean the corresponding section or problem of the textbook Introduction to the Theory of Computation (3rd Edition) by Michael Sipser.

The assignment questions may be similar to questions on problem sets from past offerings of this course or courses at other universities. Using any preexisting solutions from these or any other sources is strictly prohibited.

Each of the Questions 1-6 is worth 10 points, please attempt all problems. Please provide full proofs, unless stated otherwise.

Question 1.

The cardinality of a set measures the set’s size. Two sets $A, B$ are said to have the same cardinality if there exists a bijection (one-to-one and onto) mapping $f: A \rightarrow B$.

A set $A$ is called countably infinite, or has cardinality $\aleph_0$ (pronounced “Aleph zero”), if it has the same cardinality as the set of natural numbers $\mathbb{N} = \{1, 2, 3, \ldots\}$. Equivalently, $A$ is countably infinite, if its elements can be enumerated, e.g., $A = \{a_1, a_2, \ldots\}$.

A set $A$ has the cardinality of the continuum, denoted $\mathfrak{c}$, if it has the same cardinality as the unit interval $[0, 1]$. It can be shown that the set $\mathbb{R}$ of real numbers has cardinality $\mathfrak{c}$.

1. (4 points) Prove that the cardinality of a union of $\aleph_0$ many sets, each of cardinality $\aleph_0$, is $\aleph_0$. In other words, assume that $A_1, A_2, \ldots$ are sets of cardinality $\aleph_0$. Show that $\bigcup_{i \in \mathbb{N}} A_i$ has cardinality $\aleph_0$.

Remark (Hilbert’s paradox of the Grand Hotel): Consider a hypothetical hotel with a countably infinite number of rooms, all of which are occupied. One might be tempted
to think that the hotel would not be able accommodate any newly arriving guests, as would be the case with a finite number of rooms, where the pigeonhole principle would apply.

Suppose a new guest arrives and wishes to be accommodated in the hotel. We can (simultaneously) move the guest currently in room 1 to room 2, the guest currently in room 2 to room 3, and so on, moving every guest from his current room \( n \) to room \( n + 1 \). After this, room 1 is empty and the new guest can be moved into that room. By repeating this procedure, it is possible to make room for any finite number of new guests.

Now suppose that an infinite set of guests arrive and wishes to be accommodated in the hotel. We can move every guest from his current room \( n \) to room \( 2n \). The free rooms can accommodate all new guests.

What if infinitely many infinite hotels just closed because of hurricane Jose, and all their guests wish to be accommodated in our hotel. Can we accommodate all new guests? The above gives an affirmative answer.

2. (4 points) Prove that \( \mathfrak{c} \neq \aleph_0 \).

*Hint:* Assume for contradiction that \( a_1, a_2, \ldots \) is an enumeration of the elements in \([0, 1]\). Write each \( a_i \) as an infinite binary fraction \( a_i = 0.x_1x_2x_3x_4\ldots \), where \( x_j \in \{0, 1\} \). Construct a real number \( b \in [0, 1] \) by selecting its binary digits one-by-one, while ensuring that \( b \) is different from all the \( a_i \)'s.

*Remark:* This shows that “some infinities are larger than others”.

3. (2 points) Let \( \mathcal{P}(\{0, 1\}^*) \) be the power set of \( \{0, 1\}^* \), that is, the set of all subsets of binary strings. Prove that the cardinality of \( \mathcal{P}(\{0, 1\}^*) \) is \( \mathfrak{c} \).

*Remark:* Observed that the set \( \{0, 1\}^* \) can be enumerated (say lexicographically), thus has cardinality \( \aleph_0 \). Parts (2) and (3) shows that the cardinality of \( \mathcal{P}(\{0, 1\}^*) \) is strictly greater than the cardinality of \( \{0, 1\}^* \). Cantor’s Theorem (dated back to 1892!) uses similar arguments to show that for any set \( A \), the cardinality of \( \mathcal{P}(A) \) is strictly greater than the cardinality of \( A \).

*Remark:* Parts (2) and (3) allow us to prove that there exists a boolean function \( f : \{0, 1\}^* \rightarrow \{0, 1\} \) that cannot be computed by any JAVA program: The set of JAVA programs has cardinality \( \aleph_0 \), as they can be enumerated (say lexicographically). The set of boolean function has the same cardinality as \( \mathcal{P}(\{0, 1\}^*) \) (we choose the strings that are mapped to 1 by \( f \)). By parts (2) and (3), the cardinality of \( \mathcal{P}(\{0, 1\}^*) \) is strictly greater than \( \aleph_0 \).
Question 2.

Let \( f : \Sigma^* \rightarrow \Sigma^* \) be an arbitrary function. You are asked to (1) define a related language \( L_f \subseteq \Gamma^* \), and (2) describe how a computer program could use a procedure that decides \( L_f \) to compute \( f \), and vice-versa. Here \( \Gamma \) is an alphabet that may or may not be the same as \( \Sigma \). Ideally, on input \( x \), your program should invoke the procedure that decides \( L_f \) at most polynomially many times in the length of \( x \) and the length of \( f(x) \), (but attaining this quantitative goal is not required to get full credit for this problem).

Remark. This justifies our use of languages (equivalently, of decision problems or boolean functions) rather than the more general notion of function problems.

Question 3.

1. (4 points) Solve Exercise 1.16, Part (a) (page 86) in the textbook.
2. (4 points) Solve Exercise 1.21, Part (b) (page 86) in the textbook.
3. (2 points) Consider the following (false) claim and proof:
   
   Claim: Every language \( L \subseteq \{0,1\}^* \) is regular.

   Proof: For every string \( w \in \{0,1\}^* \), the language \( \{w\} \) is regular. Since the union of regular languages is regular, we get that \( L = \bigcup_{w \in L} \{w\} \) is regular.

   Find the bug in the proof.

Question 4.

A language over an alphabet \( \Sigma \) with only one symbol is still meaningful. It is called a unary language. In this problem, the set \( \Sigma = \{0\} \).

1. (5 points) For a fixed positive integer \( n \), define the language \( L_n \subseteq \Sigma^* \) to be the set of all strings whose length is not divisible by \( n \).

   Prove that for all \( n \geq 1 \), \( L_n \) is a regular language. You do not have to provide a formal proof, just show the construction and explain the main ideas.

2. (5 points) Prove that the language \( \text{primes} \subseteq \Sigma^* \), consisting of all strings whose length is a prime number, is not regular.
Question 5.

1. (6 points) For a word $x = x_1x_2\ldots x_n$, the reverse of $x$, denoted $x^R$, is the word $x$ written in reverse. That is, $x^R = x_n\ldots x_2x_1$. For a language $L$, let

$$L^R = \{x^R : x \in L\}.$$ 

Show that if $L$ is regular, so is $L^R$. You do not have to provide a formal proof, just show the construction and explain the main ideas.

*Hint:* Construct an NFA, reverse edges.

2. (4 points) Prove that, if a language $L$ is regular, so is $L_{1/2} = \{w : ww^R \in L\}$.

Again, you do not have to provide a formal proof, just show the construction and explain the main ideas.

*Hint:* Construct an NFA. The states of this NFA can be of the form $(p, q)$, where $p$ is a state of the DFA recognizing $L$ and $q$ is a state of the NFA you constructed in Part (1). What are the accepting states? What is the transition function?

Question 6.

A *palindrome* is a string that reads the same forwards as backwards (ignoring spaces); an example is “do geese see god”. Formally, let $\Sigma$ be the 26 letter English alphabet. Define the set of palindrome as $L = \{w \in \Sigma^* : w = w^R\}$.

1. (5 points) Prove that $L$ is not regular.

2. (5 points) Read Section 2.2 in the textbook up to the heading “Equivalence with Context-Free Grammars” (pages 111-116) describing pushdown automata. These automata are like nondeterministic finite automata, but have an extra component called a stack. The stack provides additional memory beyond the finite amount available in the automata states. The languages recognized by a pushdown automata are called *context free* languages.

Show that $L$ is context free. You do not have to provide a formal proof, just show the construction and explain the main ideas.