**Ground Rules:**

- This assignment contains only written problems. You should first attempt the problems by yourself and start working in groups after a few days of thinking.
- Typeset your solution (a good excuse to learn/master LaTeX) or write legibly. You're handing in your work electronically. We only accept PDF files. Name the file whatever you want as long as it's a single PDF.
- Exercise common sense when collaborating with others or looking things up online. Even if you work together on a problem, the writeup should be your own. This is the only way I know for you to master this kind of subject.

## Task 1: Eh? They Have The Same Cardinality? (4 points)

Remember that $\mathbb{R}$ is the set of real numbers. Let

- the close-open interval $[a, b)$ denote the set $\{x \in \mathbb{R} \mid a \le x < b\}$.
- the open-open interval $(a, b)$ denote the set $\{x \in \mathbb{R} \mid a < x < b\}$.

Prove the following statements using rigorous mathematical reasoning:

(1) $|[0, \frac{1}{2})| = |[0, 1)|$ (*Hint:* Establish a bijection)

(2) $|[0, 1)| = |(-1, 1)|$ (*Hint:* You can use two functions, instead of one.)

(3) $|[0, 1)| = |\mathbb{R}|$.

## Task 2: The Power Set of *A* (2 points)

Let $A$ be a nonempty set, though it is potentially countably infinite. Your goal in this problem is to prove that $|A| < |2^A|$. You will do it in two steps:

(1) Prove that $|2^A| = |\{0,1\}^A|$. Remember that the power set of a set $A$, denoted by $2^A$, is the set $\{X \mid X \subseteq A\}$. Furthermore, $\{0,1\}^A$ is a binary array of length $|A|$. Because $A$ is potentially infinite, it helps to reason about each $w \in \{0,1\}^A$ as a function $w : A \to \{0,1\}$. This view isn't all that outlandish: For example, if you consider an $x \in \{0,1\}^{100}$, this is a binary string of length 100; each position is 0 or 1. Another way to view $x$ is as a function: $x : \{0, 1, \ldots, 99\} \to \{0,1\}$, where $x(k)$ returns $x[k]$.

(2) Prove that $|A| < |\{0,1\}^A|$ and conclude that $|A| < |2^A|$.

## Task 3: Hamming Code (4 points)

Consider applying the Hamming coding scheme to send 8 bits of data. This will require 4 parity bits, so an encoded codeword in this scheme is 12 bits long. You will answer the following questions:

(1) If the data bits are $d_1 d_2 d_3 \ldots d_8$, what is $\beta_2$ in terms of $d_i$'s?

(2) Encode the following 8-bit data: `01101010`.

(3) Assuming that at most a single single bit flip, decide the following codewords (indicate also whether there was any error):

(i) `010011111000` (ii) `011101010010`

## Task 4: Same Number of 0s and 1s (2 points)

Consider the language $L = \{w \in \{0,1\}^* \mid w$ contains an equal number of 0s and 1s$\}$. You already know that $L$ is not regular. In this problem, show that $L$ is (Turing) decideable by providing a TM that decides it (a medium-level detail is preferred).

## Task 5: Infinite DFA (2 points)

Show that the following language is (Turing) decideable:

$$\mathsf{IDFA} = \left\{ \langle M \rangle \mid M \text{ is a DFA and } L(M) \text{ is an infinite language} \right\}.$$

## Task 6: Lucky 9 (4 points)

The number 9 is considered by many cultures to be a lucky number. Following this trend, we'll deal with languages that involve only 9s. Let $\Sigma = \{9\}$. For this problem, a high-level TM description is sufficient. **Also: you do not need any deep knowledge of number theory or analysis to solve the following.**

(1) Let $L_1 \subseteq \Sigma^*$ be defined as

$$L_1 = \begin{cases} \emptyset & \text{if } 2^{74207281} - 1 \text{ is prime} \\ \{99\} & \text{if } 2^{74207281} - 1 \text{ is } \textbf{not} \text{ prime} \end{cases}$$

Prove that $L_1$ is (Turing) decideable.

(2) Let $L_2 \subseteq \Sigma^*$ be defined as follows:

$w \in L_2 \iff w$ appears somewhere (not necessarily consecutively) in the decimal expansion of $\pi$.

For example, $\varepsilon \in L_2$, $9 \in L_2$, and $99 \in L_2$, and so is $9999 \in L_2$ because

$$\pi = 3.14159265358979323846264338327950288419716939937 51\ldots$$

Prove that $L_2$ is (Turing) decideable.

## Task 7: $\beta$-reduction (2 points)

Apply $\beta$-reduction to the following $\lambda$-expressions as much as possible:

- $(\lambda z.z)(\lambda z.zz)(\lambda z.zy)$
- $(((\lambda x.\lambda y.(xy))(\lambda y.y))w)$

## Task 8: Fibonacci (2 points)

The $n$-th Fibonacci number is given by the recurrence $f(n) = f(n-1) + f(n-2)$, where $f(0) = 1$ and $f(1) = 1$. Using the functions we have developed (e.g., `pred`, `if_then_else`, `mult`, `add`, etc.), write down an explicit $\lambda$-term $\overline{\mathsf{fib}}$ such that

$$\overline{\mathsf{fib}}\ \overline{n} =_\beta \overline{f(n)}.$$

Here, $\overline{n}$ is the Church numeral $\lambda$-term for the number $n$.

## Task 9: Power Of 2 (2 points)

Without using the recursion or fixed-point theorem, implement a $\lambda$-term for the function $pow(n) = 2^n$. You can use the standard functions we developed in class.