(1) (20 marks) Consider the following language given below
\[ L = \{ w \in \{0, 1, 2\}^* \mid w \text{ contains an equal number of } 0s, 1s, \text{ and } 2s \}. \]
(a) (5 marks) Provide the transition diagram for a TM \( M = (Q, \Sigma, \Gamma, \delta, q_0, B, F) \) that decides \( L \).
(b) (5 marks) Describe how your TM works.
(c) (5 marks) Prove that your TM accepts 201210 by providing the intermediate IDs where \( q_0 \vdash \cdots \vdash \alpha q \beta \) for some \( \alpha, \beta \in \Gamma^* \) and \( q \in F \).
(d) (5 marks) Prove that your TM does not accept 10102 by providing the intermediate IDs where \( q_0 \vdash \cdots \vdash \alpha q \beta \vdash \text{halt} \) for some \( \alpha, \beta \in \Gamma^* \) and \( q \notin F \).

(2) (10 marks) Recall that the \( i \)th binary string is denoted \( w_i \) and is defined as follows
\[ w_1, w_2, w_3, w_4, w_5, w_6, w_7, w_8, \ldots = \epsilon, 0, 1, 00, 01, 10, 11, 000, \ldots \]
In other words, the binary strings are ordered so that \( 1w_i \) is the binary representation of the decimal value \( i \). Also recall that \( M_i \) denotes the \( i \)th Turing machine, and \( M_i \) is defined as the Turing machine whose binary representation is \( w_i \).
(a) (1 mark) What is the binary representation of the decimal value 11160850?
(b) (1 mark) What is the binary string \( w_{11160850} \)?
(c) (4 marks) Draw a transition diagram for \( M_{11160850} \). Assume \( q_2 \) is the only final state.
(d) (4 marks) Is \( w_4 \in L(M_{11160850}) \)? Justify your answer by showing the Turing machine’s IDs that lead it to halt on this input, or by arguing that the input causes the Turing machine to enter an infinite loop.

(3) (10 marks) Another variation of a Turing Machine that has equivalent power is a 2-Head Turing machine (2HTM). A 2HTM is a 7-tuple \( M = (Q, \Sigma, \Gamma, \delta, q_0, B, F) \) that is similar to a regular Turing machine except that it adds a second tape-head to the single infinite tape. The transition function \( \delta \) takes as input the current state \( q \in Q \), the current symbol \( X_i \in \Gamma \) being scanned by the first tape-head, and the current symbol \( X_j \in \Gamma \) being scanned by the second tape-head. If \( \delta(q, X_i, X_j) \) is defined, then it equals \( (p, Y, Z, D_1, D_2) \) where \( p \in Q \) is the new state, \( Y \in \Gamma \) overwrites \( X_i \), \( Z \in \Gamma \) overwrites \( X_j \), and \( D_1, D_2 \in \{L, R, S\} \) describe whether the first and second tape-heads move left, right or stay stationary, respectively. In this question you will practice the formal specification of machines by defining some aspects of a 2HTM. In your first two definitions assume that \( X_1X_2\ldots X_n \) is the area of interest on the tape, and that the first tape-head is scanning \( X_i \) and the second tape-head is scanning \( X_j \). (The area of interest on the tape stretches from the leftmost non-blank or tape-head position, to the rightmost non-blank or tape-head position.)
(a) (2 marks) Define an ID (instantaneous description) for a 2HTM.

(b) (4 marks) Define $\vdash$ for a single move between two IDs when $\delta(q, X_i, X_j) = (p, Y, Z, D_1, D_2)$. To limit the number of cases under consideration, you may assume $D_1 = R, D_2 = R, i > 1,$ and $j < n$.

(c) (2 marks) Define the language $L(M)$ of the 2HTM. Assume $\vdash^*$ is defined.

(d) (2 marks) Give an example of a language that could be decided more efficiently using a 2HTM than a regular TM. Here the term efficient refers to the number of moves required before halting.

Note: There is more than one “correct” solution for each of these questions.

(4) (10 marks) Give an example of a language $L \subseteq \Sigma^*$ that is not acceptable, and whose complement $\overline{L} \subseteq \Sigma^*$ is also not acceptable. You must justify that there is no Turing machine $M$ such that $L(M) = L$, and that there is no Turing machine $M'$ such that $L(M') = \overline{L}$, but you do not need to rigorously prove these results.

(5) (10 marks) An arc $(v_i, v_j)$ can be encoded in binary as $0^i10^j$. A directed graph with vertices $v_1, v_2, \ldots, v_n$ and arcs $a_1, a_2, \ldots, a_m$ can be encoded in binary as

$$b_111b_211 \cdots 11b_m$$

where $b_i$ is the binary encoding of $a_i$. For example, a directed graph and its encoding appears below

Consider the following two languages

$L_p = \{g1110^i10^j \in \{0,1\}^* \mid \text{there is a directed path from } v_i \text{ to } v_j \text{ in the directed graph with binary encoding } g\}$

$L_c = \{g1110^i10^j \in \{0,1\}^* \mid \text{there is a cycle that includes } v_i \text{ and } v_j \text{ in the directed graph with binary encoding } g\}$

For example, if $g$ is the binary encoding of the directed graph given above, then $g111000010 \in L_p$ since there is a directed path from $v_4$ to $v_1$ in $g$. On the other hand, $g111000010 \notin L_c$ since there is no directed cycle that includes both $v_4$ and $v_1$. In this question, you will show how an algorithm for finding directed cycles can be used to create an algorithm for finding directed paths, and vice versa.

(a) (5 marks) Reduce $L_p$ to $L_c$. In other words, describe a function $f : \{0,1\}^* \rightarrow \{0,1\}^*$ such that $b \in L_p \iff f(b) \in L_c$.

(b) (5 marks) Reduce $L_c$ to $L_p$. In other words, describe a function $f : \{0,1\}^* \rightarrow \{0,1\}^*$ such that $b \in L_c \iff f(b) \in L_p$. 
