1. [3 points] Let $\Sigma = \{0, 1\}$. Prove that the set of all strings is *countable* but the set of all languages is *uncountable*. Recall: A set is said to be countable if it is finite or if there exists a bijection between that set and the set of Natural numbers. If a set is not countable, then it is said to be uncountable. [1] [CO4]

2. [2 points] *A grammar is defined as ambiguous if a string in its language has two or more distinct parse trees or, equivalently, two or more distinct leftmost derivations.*
The language $L = \{w \in \{0,1\}^* \mid w \text{ has an equal number of 0s and 1s}\}$ can be generated by the following Context-Free Grammar (CFG):

$$G : S \to SS \mid 0S1 \mid 1S0 \mid \epsilon$$

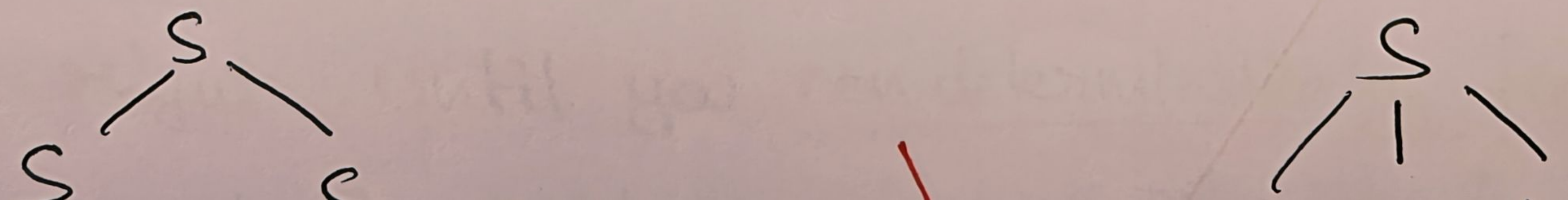
This grammar is ambiguous. Your task is to prove this.

(i) Find the shortest non-empty string $w \in L(G)$ that demonstrates the ambiguity of the grammar. *(1 point)*

(ii) Formally prove that the grammar G is ambiguous by providing two distinct parse trees for the string w you identified in part (i). *(1 point)* [CO1, CO2, CO3]

(i) take the string 0 1

(ii)



3. [4 points] Construct a Turing machine that decides the language

$$L = \{\, ww \mid w \in \{0,1\}^* \,\}.$$

You may provide a high level description without the transition function, but without skipping any of the algorithm details. [CO1, CO2, CO3, CO4]

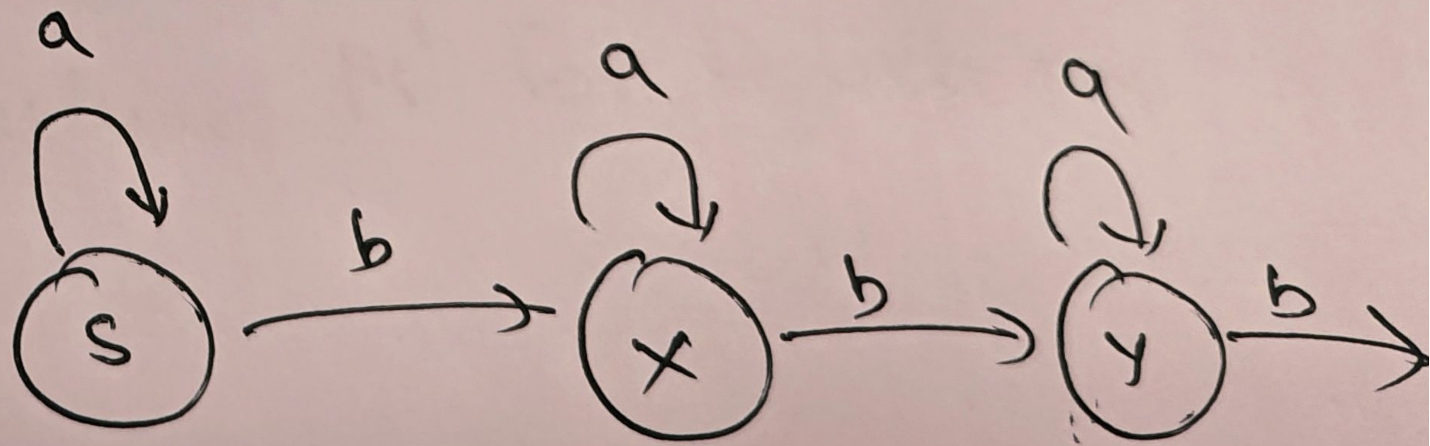
2

4. [2 points] Describe the language generated by this grammar:

- $S \longrightarrow aS \mid bX$
- $X \longrightarrow aX \mid bY$
- $Y \longrightarrow aY \mid bZ$
- $Z \longrightarrow aZ \mid \epsilon$

[CO1, CO2, CO3]

2



0.5

5. [4 points] Prove that the following problem is undecidable by reduction:

$$\text{REGULAR}_{TM} := \{\langle M \rangle \mid M \text{ is a TM and } L(M) \text{ is a regular language}\}$$

Hint: You can use the fact that the language $A_{TM} := \{\langle M, w\rangle \mid M \text{ is a TM and } M \text{ accepts } w\}$ *is undecidable.* **[CO3, CO4]**

4.5
+1.5
=6

6. [6 points] The language L consisting of all strings having an equal number of 0's and 1's is context-free. State whether the following languages are regular or context-free. Draw the corresponding automaton (DFA/PDA) to support your answer.

(i) $L_1 = \{w \mid |\#0's - \#1's| \leq 1\}$ *(3 points)*

(ii) $L_2 = \{w \mid |\#0's - \#1's| \leq 1, \forall \text{ prefixes of } w\}$ *(3 points)*

Example: The string 00001111 will belong to L_1 but not L_2 since 00001, a prefix of 00001111, does not satisfy $|\#0's - \#1's| \leq 1$. **[CO1, CO2, CO3, CO4]**

7. [5 points] We say that M is an Oblivious Turing Machine if it has a single tape whose head movements do not depend on the input. The head movements depend only the input length. That is, for any input $x \in \{0,1\}^*$, and $i \in \mathbb{N}$, the location of M's head at the i^{th} step of execution on input x is only a function of $|x|$ and i. Prove that the Oblivious Turing Machine model is computationally equivalent to the ordinary single tape Turing Machine model. **[CO3, CO4]**

Hint: What this means is that for cell c, no matter which input symbol we give, it will always move the head either always one step to the right, or one step to the left.

8. [4 points] Let

$$\Sigma_3 = \left\{ \begin{bmatrix}0\\0\\0\end{bmatrix}, \begin{bmatrix}0\\0\\1\end{bmatrix}, \begin{bmatrix}0\\1\\0\end{bmatrix}, \ldots, \begin{bmatrix}1\\1\\1\end{bmatrix} \right\}$$

Σ_3 contains all size 3 columns of 0s and 1s. A string of symbols in Σ_3 gives three rows of 0s and 1s. Consider each row to be a binary number and let

$$B = \{w \in \Sigma_3^* \mid \textit{the bottom row of } w \textit{ is the sum of the top two rows}\}.$$

For example, $\begin{bmatrix}0\\0\\1\end{bmatrix}\begin{bmatrix}1\\0\\0\end{bmatrix}\begin{bmatrix}1\\1\\0\end{bmatrix} \in B$ but $\begin{bmatrix}0\\0\\1\end{bmatrix}\begin{bmatrix}1\\0\\1\end{bmatrix} \notin B$

Show that B is regular. **[CO1, CO2, CO3, CO4]**

Hint: Working with B^R is easier. Here L^R is defined as: $L^R = \{w^R \mid \forall\, w \in L\}$, where w^R is the string w, but in reverse. You can assume that if A^R is regular, then A is also regular.

w^R

[0,1,0]
[1,0,0]

9. [5 points] Let

$$f(x) = \begin{cases} 3x+1 & \text{if } x \text{ is odd,} \\ x/2 & \text{if } x \text{ is even,} \end{cases}$$

for any natural number x. Define $C(x)$ as the sequence $x, f(x), f(f(x)), \ldots,$ which terminates if and when it hits 1. For example, if $x = 7$, then

$$C(x) = (7, 22, 11, 34, 17, 52, 26, 13, 40, 20, 10, 5, 16, 8, 4, 2, 1).$$

Computer tests have shown that $C(x)$ hits 1 eventually for x ranging from 1 to 87×2^{60}. But the question of whether $C(x)$ ends at 1 for all $x \in \mathbb{N}$ is not proven. This is believed to be true and is known as the Collatz conjecture.

Suppose that A_{TM}, the membership problem

$$A_{TM} = \{ \langle M, w \rangle \mid M \text{ is a Turing machine that accepts input } w \},$$

were decidable by a Turing machine A. Use A to describe a Turing machine that is guaranteed to prove or disprove the Collatz conjecture. [CO4]

BONUS **(Non-mandatory Bonus Question)** - Answer any ONE of the following:

(i) We define a Write Twice Turing Machine as a single tape Turing Machine that can alter each tape cell at most twice (including the input portion of the tape). Show that this variant of the single tape Turing Machine is computationally equivalent to the ordinary single tape Turing Machine model.

Basically, the Write Twice Turing Machine starts off with the input on the tape, and with blanks on every other cell. Once execution starts, the machine can write on top of any cell only twice. Note that any cell can be read any number of times. For example, if the tape started off as $\Delta \mid a \mid b \mid c \mid \Delta \mid \Delta$, then in one transition the tape can be changed to $\Delta \mid d \mid b \mid c \mid \Delta \mid \Delta$, and then in the next transition change it to $\Delta \mid e \mid b \mid c \mid \Delta \mid \Delta$. However, after this the Turing Machine cannot write over the cell that contained a at the beginning.

(ii) For a fixed binary k and regular languages A and B, we define the language L_k as follows:

$$L_k = \{x \oplus y \oplus k \mid \ x \in A \ , \ y \in B, \ |x| = |y| = |k|\}$$

where A and B are regular binary languages and $\oplus$ is the bit-wise XOR operator.

For a given k = "0101" and regular languages A and B, let's say A has the string "0011" in its language and B has the string "1100" in its language. We can see that L_k will have the string $(0011 \oplus 1100 \oplus 0101) = 1010$ in its language.

Is L_k a regular language? If your answer is yes, prove it by constructing a DFA/NFA for it. If you answer No, use the pumping lemma to disprove it.

N.B: Answering a bonus question is not mandatory. Correct answers to any of the two bonus questions will fetch you extra points. This will be added to the total score you get in the course as a "top-up".

Nb