Problem 1. In your own words, give a definition of computation.

Solution 1. Computation is the transformation of inputs to outputs with a concrete machine of some sort to do the work.

Problem 2. Give a regular expression for each of the following languages. You may assume that $\Sigma = \mathbb{Z}_2$.

a) The language where every third symbol is a 1

b) The language where the length of every string is divisible by three or five.

Solution 2.

a) $(\Sigma \Sigma \Sigma)^*(\epsilon | 0 | 1)^2$

b) $(\Sigma \Sigma \Sigma)^* | (\Sigma \Sigma \Sigma \Sigma \Sigma)^*$

Problem 3. Give a right-regular grammar for the following languages. You may assume that $\Sigma = \mathbb{Z}_2$.

a) The language of binary encodings of natural numbers with no leading 0’s (e.g. include 4 = 100 but not 4 = 0100)

b) The language where the third to last symbol in every string is 1 (exclude strings of length less than three)

Solution 3.

a) With start symbol $S$,

$$S \rightarrow 1A \quad A \rightarrow \epsilon | 0A | 1A$$

b) With start symbol $S$,

$$S \rightarrow 0S \mid 1S \mid 1A \quad A \rightarrow 0B \mid 1B \quad B \rightarrow 0 \mid 1$$

Problem 4. Give a short description of the language of each of the following right-regular grammars. For each grammar, $S$ is the start symbol and $\Sigma = \mathbb{Z}_2$.

a)
\[ S \to 0A \quad A \to 0B \quad B \to 0A \mid \epsilon \]

b) 
\[
S \to A_{00} \quad A_{00} \to \epsilon \mid 0A_{10} \mid 1A_{01} \quad A_{10} \to 0A_{00} \mid 1A_{11} \\
A_{01} \to 0A_{11} \mid 1A_{00} \quad A_{11} \to 0A_{01} \mid 1A_{10}
\]

Solution 4.

a) \((00)^*\)

b) The language is the set of strings with an even number of 0’s and an even number of 1’s.

Problem 5. Define the reverse operator on a string \(\omega = \omega_1 \ldots \omega_n\) to be \(\omega^R = \omega_n \ldots \omega_1\). On a language \(L\), we can define
\[ L^R = \{ \omega^R \mid \omega \in L \}. \]
Prove that if \(L\) is regular, then \(L^R\) is regular.
(Hint: \(S \to \epsilon\))

Solution 5. If \(L\) is regular, then there must be a right-regular grammar \(G = (N, \Sigma, P, S)\). To reverse the language of \(G\), let \(S^R\) be the new start symbol. We will then reverse the production rules of \(P\) to create a new set of production rules \(P^R\) as follows.

- For each, \(A \to aB \in P\), we create a new production rule \(B \to aA\) and add it to \(P^R\).
- For each \(A \to a \in P\), we create a new production rule \(S^R \to aA\) and add it to \(P^R\).
- Add the production rule \(S \to \epsilon\) to \(P^R\).

Then we claim that \(G^R = (N \cup \{S^R\}, \Sigma, P^R, S^R)\) is a right-regular grammar whose language is \(L^R\).

Suppose \(\omega \in L(G)\). Then \(S \Rightarrow^* \omega\), so there is a sequence of productions such that
\[ S \Rightarrow a_1A_1 \Rightarrow \ldots \Rightarrow a_1 \ldots a_nA_n \Rightarrow a_1 \ldots a_na_{n+1}, \]
where \(a_1 \ldots a_{n+1} = \omega\). Then by construction, it follows that
\[ S^R \Rightarrow a_{n+1}A_n \Rightarrow \ldots \Rightarrow a_{n+1} \ldots a_2A_1 \Rightarrow a_{n+1} \ldots a_2a_1S \Rightarrow a_{n+1} \ldots a_1\epsilon, \]
so \(S^R \Rightarrow^* a_{n+1} \ldots a_1 = \omega^R\). Thus \(\omega^R \in L(G^R)\).

A similar argument in reverse shows that if \(\omega \in L(G^R)\), then \(\omega^R \in L(G)\). \(\square\)