Lecture #1: Introduction to CPSC 351
Alphabets, Strings, and Languages

Introduction

This document introduces ***alphabets***, ***strings*** and ***languages***. These will be used intensively, during the beginning and middle of this course, to model computational problems and algorithms or machines that can be used to solve them.

It is assumed that you have worked through at least the first two parts of the ***mathematics review*** for this course. Please work through this material, if you have not already done so, before you read the material that follows.

Alphabets

Definition 1. An ***alphabet*** is a finite non-empty set. The elements of an alphabet are often called ***symbols***.

In general, an "alphabet" includes the basic elements, or building blocks, in a system, which are combined to form more complex entities (which are described below).

- Alphabets in this course will often be given names Σ, Γ, or Σ_i or Γ_i, for some non-negative integer i, when more than one alphabet must be considered.
- Alphabets that will initially include simple ones like $\{0, 1\}$, $\{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$, the set of decimal ***digits*** $\{0, 1, 2, \ldots, 9\}$, or the set of lower-case letters in the Roman alphabet $\{\mathtt{a}, \mathtt{b}, \mathtt{c}, \ldots, \mathtt{z}\}$. More complicated alphabets, that are more useful for the problems being solved, will be considered later on.
- Another example — that will not be used in this course (because it is too large) is the set of possible values that can be represented using a single word of machine memory —

which could be modelled as the set of integers between 0 and $2^k - 1$, where k is the size (in bits) of a word of machine memory.

Strings

Definition 2. A ***string***, over an alphabet Σ, is a finite sequence of elements of Σ.

Strings are the "more complex entities", assembled using the symbols in an alphabet, that are mentioned above.

- ***Example (and Notation):*** If $\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ then the sequence

  $$\mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{a}$$

  is an example of a string over the alphabet Σ. However, strings are often written by listing the symbols in the string immediately after the other, without spaces or commas separating them, so that

  $$\mathtt{abca}$$

  is another (more frequently used) representation of the same string over Σ.

- ***Note:*** Since strings are *sequences* of symbols — and not *sets* of symbols — the order in which symbols appear in a string, and the number of copies of each symbol that appear, both matter.

  For example, the strings "$\mathtt{aba}$", "$\mathtt{aab}$", "$\mathtt{ab}$" and "$\mathtt{ba}$" are distinct (that is, no two are the same) even though the sets "$\{\mathtt{a}, \mathtt{b}, \mathtt{a}\}$", "$\{\mathtt{a}, \mathtt{a}, \mathtt{b}\}$", "$\{\mathtt{a}, \mathtt{b}\}$" and "$\{\mathtt{b}, \mathtt{a}\}$" are all the same (that is, each one is equal to each of the others, as a set).

Definition 3. The ***length*** of a string ω, over an alphabet Σ, is the same as its length when it is considered as a sequence. The length of a string ω is often written as $|\omega|$.

- ***Example:*** The length of the string "$\mathtt{abac}$" (over the alphabet $\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$) is 4.
- ***Special Case:*** The string of length zero (over any alphabet Σ) is called the ***empty string*** and is denoted by λ.[1]

Definition 4. For any alphabet Σ, $\boldsymbol{\Sigma^\star}$ is the set of *all* strings over Σ.

[1] Some other references denoted the empty string by "ε" instead.

Definition 5. Suppose that

$$\mu = a_1 a_2 \dots a_n \qquad \text{and} \qquad \nu = b_1 b_2 \dots b_m$$

are strings over an alphabet Σ (that is, suppose that $\mu, \nu \in \Sigma^\star$) with lengths n and m, respectively, so that

$$\alpha_1, \alpha_2, \dots, \alpha_n, \beta_1, \beta_2, \dots, \beta_m \in \Sigma.$$

Then the ***concatenation*** of μ and ν, denoted by $\mu \cdot \nu$, is the string

$$\alpha_1 \alpha_2 \dots \alpha_n \beta_1 \beta_2 \dots \beta_m$$

over Σ, with length $n + m$, obtained by listing the symbols in ν after the symbols in μ.

Definition 6. If $\mu, \omega \in \Sigma^\star$, for an alphabet Σ, then μ is a ***substring*** of ω if there exist strings $\nu, \varphi \in \Sigma^\star$ such that $\omega = \nu \cdot \mu \cdot \varphi$.

Another (equivalent) way to define "substring" is as follows: If

$$\omega = a_1 a_2 \dots a_n \qquad \text{and} \qquad \mu = b_1 b_2 \dots b_m$$

are strings in $\Sigma^\star$ with lengths n and m respectively (so that $a_1, a_2, \dots, a_n, b_1, b_2, \dots, b_m \in \Sigma$), then μ is a ***substring*** of ω if $m \leq n$ and there exists an integer i such that $0 \leq i \leq n - m$, and such that $b_j = a_{j+i}$ for every integer j such that $1 \leq j \leq m$.

Definition 7. Once again, suppose that $\mu, \omega \in \Sigma^\star$, for an alphabet Σ. Then μ is a ***prefix*** of ω if there exists a string $\nu \in \Sigma^\star$ such that

$$\omega = \mu \cdot \nu,$$

and μ is a ***suffix*** of ω if there exists a string $\nu \in \Sigma^\star$ such that

$$\omega = \nu \cdot \mu.$$

Another (equivalent) way to define "prefix" and "suffix" is as follows: If

$$\omega = a_1 a_2 \dots a_n \qquad \text{and} \qquad \mu = b_1 b_2 \dots b_m$$

are strings in $\Sigma^\star$ with lengths n and m respectively (so that $a_1, a_2, \dots, a_n, b_1, b_2, \dots, b_m \in \Sigma$), then μ is a ***prefix*** of ω if $m \leq n$ and $b_j = a_j$ for every integer j such that $1 \leq j \leq n$; μ is a ***suffix*** of ω if $m \leq n$ and $b_j = a_{j+n-m}$ for every integer j such that $1 \leq j \leq n - m$, instead.

Languages

Definition 8. A ***language*** over an alphabet Σ is a subset of $\Sigma^\star$.

That is, a ***language*** over an alphabet Σ is a set of (some of the) strings over Σ.

One kind of computational problem is a ***decision problem*** — one where the required answer is either "Yes" or "No". Strings of symbols over an alphabet Σ can be used to represent, or "encode", instances of a decision problem — so that the set of instances of a decision problem, for which the corresponding answer is "Yes", is encoded by a language over Σ.

We will, therefore, be considering various (kinds of) algorithms that decide membership in languages, as well as properties of languages, in much of this course