Logic - Notes

Dom Hutchinson

January 27, 2020

Contents

1 Introduction

1.1 Alphabets & Strings

Definition 1.1 - *Alphabet*
An *Alphabet* is a set of symbols from which *Strings* can be created.

Definition 1.2 - *String*
A *String* over a set $\mathcal{A}$ is any sequence $\alpha := \langle a_1, \dots, a_n \rangle$ where $a_1, \dots, a_n \in \mathcal{A}$.
N.B. Here we say α has *length* n and $\alpha \in \mathcal{A}^n$.

Remark 1.1 - *Concatenating Strings*
Define *Strings* $\alpha := \langle a_1, \dots, a_n \rangle \in \mathcal{A}^n$ and $\beta := \langle b_1, \dots, b_m \rangle \in \mathcal{A}^m$.
We define *Concatenation* of α & β as $\alpha\beta := \langle a_1, \dots, a_n, b_1, \dots, b_m \rangle$ Note that

$$\alpha\beta \neq \langle \alpha, \beta \rangle = \langle \langle a_1, \dots, a_n \rangle, \langle b_1, \dots, b_m \rangle \rangle$$

N.B. Sometimes the following notation is used $\alpha * \beta$.
Example 1.1 - *English Alphabet*
If we define an alphabet $\mathcal{A} := \{`a`, \dots, `z`\}$ then $\langle `t`, `h`, `i`, `s` \rangle$ is a *String* of $\mathcal{A}$.

Remark 1.2 - *Ambiguity when using multiple Alphabets*
Consider the *Alphabets* $\mathcal{A}_1 := \{0, 1, \dots, 9\}$ & $\mathcal{A}_2 := \mathbb{N}$.
Then we are unsure which of the following definitions of 123 is valid

$$\langle 123 \rangle,\ \langle 12, 3 \rangle,\ \langle 1, 23 \rangle, \langle 1, 2, 3 \rangle$$

Remark 1.3 - $\mathcal{A} := \{0, 1\}$ *is sufficient to describe any language - binary*

Remark 1.4 - *Describing Formal Languages*
When describing a *Formal Language* we need to provide two things

1. An *Alphabet* which defines what symbols are allowed.

2. A *Grammar* which defines what combinations of symbols are allowed.