ECS 120 Lesson 1 – Alphabets, Languages and Grammars, Pt. 1

Oliver Kreylos

Friday, March 30th, 2001

1 Outline

Today's lesson will lay the foundations for all further material in this class. It will supply the definitions and techniques needed in the next couple of lessons to define a *formal language* – as opposed to a natural language like English – and introduce methods of defining specific languages. It will also reinforce the mathematical techniques of recursive definition and induction.

2 Alphabets

An *alphabet* $\Sigma = \{s_1, s_2, \ldots, s_n\}$ is a non-empty, finite set of symbols, also called *characters*. Examples:

- The latin alphabet $\{\mathrm{a}, \mathrm{b}, \ldots, \mathrm{z}, \mathrm{A}, \mathrm{B}, \ldots, \mathrm{Z}\}$.
- The greek alphabet $\{\alpha, \beta, \ldots, \omega, \mathrm{A}, \mathrm{B}, \ldots, \Omega\}$.
- The ASCII set.
- The Unicode set.

3 Words

A *word* x over an alphabet Σ is the concatenation of a finite number of symbols from Σ. In programming language terms, a word is a string of char-

acters. To handle words over an alphabet Σ mathematically, we recursively define Σ^n, the set of all words of length n over Σ:

1. Σ^0, the set of words of length zero over Σ, has exactly one element – the *empty word* ϵ: $\Sigma^0 := \{\epsilon\}$. (Note: ϵ is not a character, and cannot be an element of Σ.)

2. For $n > 0$, Σ^n is constructed by taking every word $x \in \Sigma^{n-1}$ and prefixing it with every character $s \in \Sigma$: $\Sigma^n := \left\{ sx \;\middle|\; s \in \Sigma, x \in \Sigma^{n-1} \right\}$.

3. Σ^*, the set of all words over Σ, is defined as the union of all Σ^n: $\Sigma^* := \bigcup_{n \geq 0} \Sigma^n$. The *-operator is often called "Kleene star."

From this definition, it follows that the sets Σ and Σ^1 are not identical! Σ is an alphabet, i. e., a set of characters, whereas Σ^1 is a set of words of length one. This might seem subtle, especially since in this course characters and one-character words will be written in the same way, but is exactly equivalent to the way strings are treated in the `C` programming language: `'a'` is the character a, whereas `"a"` is the one-character word a. From the above definition, the difference becomes obvious: The character a is just a single symbol, whereas the word a is the concatenation of the character a and the empty word ϵ. In the `C` language, this is the same, with the ASCII `NUL` character taking the role of the empty word. This means that `NUL` is, in `C`'s philosophy, not part of the ASCII alphabet – a `C` string cannot contain a `NUL` character.

The *length* $|x|$ of a word x is defined as its number of characters. If $x \in \Sigma^n$, then $|x| = n$.

3.1 Discussion Problems

- How many elements does Σ^n have?
- How many elements does $\bigcup_{n=0}^{N} \Sigma^n$, the set of all words over Σ up to and including length N, have?
- How many elements does Σ^* have?
- Give a proper, recursive definition for the length of a word.