FORMAL LANGUAGES

Alphabets and Strings

- An alphabet Σ is a finite set of characters (or symbols).
- A word, or sequence, or string over Σ is any group of 0 or more consecutive characters of Σ.
- The length of a word is the number of characters in the word.
- The null string is the string of length 0. It is denoted ε or λ.
- A string of length n is really an ordered n-tuple of characters written without parentheses or commas.
- Given two strings x and y over Σ, the concatenation of x and y is the string xy obtained by putting all the characters of y right after x.

Languages over an alphabet

Let Σ be an alphabet. A formal language over Σ is a set of strings over Σ.

- $\varnothing$ is the empty language (over Σ)
- Σ^n = {all strings over Σ that have length n} where $n \in \mathbb{N}$
- Σ^+ = the positive closure of Σ ={all strings over Σ that have length ≥ 1}
- Σ^* = the Kleene closure of Σ = {all strings over Σ}

Operations on Languages

Let Σ be an alphabet. Let L and L′ be two languages defined over Σ.
The following operations define new languages over Σ:

- The concatenation of L and L′, denoted LL′, is $LL' = \{xy \mid x \in L \wedge y \in L'\}$
- The union of L and L′, denoted $L \cup L'$, is $L \cup L' = \{x \mid x \in L \vee x \in L'\}$
- The Kleene closure of L, denoted L^*, is $L^* = \{ x \mid x$ is a concatenation of any finite number of strings in $L\}$. Note that $\varepsilon \in L^*$.

REGULAR EXPRESSIONS

Definition

Let Σ be an alphabet. The following are regular expressions (r.e.) over Σ:

I. BASE: ε and each individual symbol of Σ are regular expressions.

II. RECURSION: if r and s are regular expressions over Σ, then the following are also regular expressions over Σ:
- (rs) the concatenation of r and s
- (r | s) r or s
- (r^*) the Kleene closure of r

III. RESTRICTION: The only regular expressions over Σ are the ones defined by I and II above.

Order of Precedence of Regular Expression Operations

- The order of precedence of r.e. operators are, from highest to lowest:
- Highest: () * concatenation | : lowest

Languages Defined by Regular Expressions

Let Σ be an alphabet. Define a function L as follows:

$$L: \begin{cases} \{\text{all r.e's over S}\} \rightarrow \{\text{all languages over S}\} \\ \quad r \;\mapsto L(r) = \text{the language defined by r} \end{cases}$$

I. BASE: $L(\varepsilon) = \{\varepsilon\}$, $\forall a \in \Sigma$ $L(a)=\{a\}$

II. RECURSION: If L(r) and L(s) are the languages defined by the regular expressions r and s over Σ, then
- $L(rs) = L(r)L(s)$
- $L(r|s) = L(r) \cup L(s)$
- $L(r^*) = (L(r))^*$

Variations

Some definitions of regular expressions and regular languages define $\varnothing$ to be a r.e. with $L(\varnothing)=\varnothing$

Shorthand:

- [a-c] = a|b|c
- [^a-c] = any letter other than a, b, c
- $r^+ = rr^*$
- r{n} = r is concatenated n times
- [a-c x-z]=[a-c,x-z] = a|b|c|x|y|z
- . means any character
- r? = (r|ε)
- r{n,m} r concatenated n to m times