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

REGULAR EXPRESSIONS

Definition

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

I. BASE: ε and each individual symbol of $\sum$ are regular expressions.

II. RECURSION: if r and s are regular expressions over $\sum$, then the following are also regular expressions over $\sum$:

- (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 $\sum$ 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 $\sum$ be an alphabet. Define a function L as follows:

$$L : \left[\begin{array}{lcl} \{\text{all r.e.'s over } \Sigma\} & \rightarrow & \{\text{all languages over } \Sigma\} \\ r & \mapsto & L(r) = \text{the language defined by r} \end{array} \right.$$

I. $L(\varepsilon) = \{\varepsilon\}$, $\forall a \in \sum$ $L(a)=\{a\}$

II. RECURSION: If L(r) and L(s) are the languages defined by the regular expressions r and s over $\sum$, 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 $L(\varnothing)=\varnothing$

PROPERTIES OF REGULAR EXPRESSIONS

Regular expressions can be simplified by applying the following properties:

For any regular expressions r, s, t,

Axiom	Description
$r \mid s = s \mid r$	| is commutative
$r \mid (s \mid t) = (r \mid s) \mid t = r \mid s \mid t$	| is associative
$(rs)t = r(st) = rst$	Concatenation is associative
$r(s\|t) = rs \mid rt$ and $(s\|t)r = sr \mid tr$	Concatenation is distributive over |
$r\varepsilon = \varepsilon r = r$	ε is the identity element for concatenation
$r^{**} = r^*$	* is idempotent
$r^* = (r\|\varepsilon)^*$	

NOTATIONAL SHORTHANDS

Here are some frequent constructs which have their own notation:

- $(r)^+$ means one or more instances of r.

  $L((r)+) = (L(r))+$
- (r)? means 0 or 1 instances of r. i.e. $(r)? = r|\varepsilon$

  $L((r)?) = (L(r|\varepsilon)) = L(r) \cup L(\varepsilon) = L(r) \cup \{\varepsilon\}$
- Character classes:

  [abc] = a|b|c

  [a-z] = a|b|…|z

REGULAR DEFINITIONS

Regular expressions can be broken down into regular definitions: sequences of expressions of the form

$d_1 \rightarrow r_1$

…

$d_n \rightarrow r_n$

where each d_i is a distinct name and

r_i is a regular expression over symbols in $\sum \cup \{d_1, d_2, \ldots d_{i-1}\}$