Lexical Analysis

Sudakshina Dutta

IIT Goa

2^{nd} February, 2022

Strings and Languages

- **Alphabet** Any finite set of symbols (generally denoted by $\sum$)
  - Example of symbols are letters, digits and punctuation
  - Example of alphabets : $\{0, 1\}$, $\{a, b, c\}$, ASCII code, etc.
- **String** A finite sequence of symbols drawn from alphabet
  - Sentence/word are synonyms of string
  - Example : 1001 is a string from alphabet $\{0, 1\}$. *aabbccc* is a string from the alphabet $\{a, b, c\}$
  - Length of string s is denoted as $|s|$
  - Empty string ϵ is a string of length 0
  - $\sum^*$ is the set of all strings over the alphabet $\sum$
- **Language** A countable set of strings over some fixed alphabet

- **Language** Any countable set of strings over some particular alphabet
  - Example : $\{0, 11, 1010\}$, $\{0\}$, $\{1, 11, 111\}$ are languages over the alphabet $\{0, 1\}$
  - $\emptyset$ represents the empty set and $\{\epsilon\}$ is the language containing empty string
  - Each language has a finite representation
  - The set of strings $\{01, 10, 111\}$ is a finite language

Regular expression denotes regular language

Construction of regular expressions

- Each regular expression r defines the language $L(r)$
- It is defined recursively from the languages denoted by the sub-expressions of r
- **Basis**
  1. ϵ is a regular expression and $L(\epsilon)$ is $\{\epsilon\}$
  2. If a is a symbol in $\sum$, then a is a regular expression and $L(a) = \{a\}$ i.e., the language of only one string a
  3. ϕ is a regular expression and the language is $L(\phi) = \phi$
- **Induction**
  1. $(r)|(s)$ is a regular expression denoting language $L(r) \cup L(s)$
  2. $(r)(s)$ is a regular expression denoting the language $L(r)L(s)$
  3. $(r)^*$ is a regular expression denoting $(L(r))^*$
  4. (r) is a regular expression denoting $L(r)$

Operations on Languages

- **Union** The set of all strings from both the languages
- **Concatenation** The set of all strings formed by taking a string from the first language and a string from the second language
- **Kleene Closure (L^*)** The set of strings obtained by concatenating the strings zero or more times
  —L^0 : Concatenation of L zero times and it is defined to be $\{\epsilon\}$
- **Positive Closure** Same as the Kleene Closure, but without the term L^0
  —ϵ will not be in L^+

Operations	Definition and Notation
Union of L and M	$L \cup M = \{\ s \mid s \text{ is in } L \text{ or } s \text{ is in } M\}$
Concatenation of L and M	$LM = \{\ st \mid s \text{ is in } L \text{ and } t \text{ is in } M\}$
Kleene closure of L	$L^* = \bigcup_{i=0}^{\infty} L^i$
Positive closure of L	$L^+ = \bigcup_{i=1}^{\infty} L^i$
? operator	$L(r?) = L(r) \cup \{\epsilon\}$

- **Examples**
- $L \cup D$ is the set of letters and digits
- LD is the set of strings of length two each consisting of one letter followed by one digit
- $L(L \cup D)^*$ is the set of all strings of letters and digit starting with a letter
- D^+ is the set of all strings of one or more digits

- **Examples of Regular Expression**
  1. L = set of all strings of 0's and 1's, $r = (0+1)^*$
  2. L = set of all strings of 0's and 1's, with at least two consecutive 0's, $r = (0+1)^*00(0+1)^*$
  3. $L = \{w \in \{0,1\}^* | w$ has two or three occurrences of 1, the first and second of which are not consecutive$\}$, $r = 0^*10^*010^*(10^* + \epsilon)$ r = 0*10+10* + 0*10+10*10*
  4. L = set of all strings of 0's and 1's beginning with 1 and not having two consecutive 0's, $r = (1+10)^*$
  5. L = set of all strings of 0's and 1's ending in 011, $r = (0+1)^*011$

Precedence and associativity of operators

- The unary operator $*$ has highest precedence and is left associative
- Concatenation has the second highest precedence and is left associative
- $|$ has the lowest precedence and is left associative
- Hence, $(a)|((b) * (c))$ can be replaced by $a|b^*c$
- **Example** Let $\sum = \{a, b\}$
  1. The regular expression $a|b$ denotes the language $\{a, b\}$
  2. $(a|b)a$ denotes the language $\{aa, ba\}$
  3. a^* denotes the set of string $\{\epsilon, a, aa, aaa, \cdots\}$

LAW	DESCRIPTION
$r \vert s = s \vert r$	$\vert$ is commutative
$r \vert (s \vert t) = (r \vert s) \vert t$	$\vert$ is associative
$r(st) = (rs)t$	Concatenation is associative
$r(s \vert t) = rs \vert rt;\ (s \vert t)r = sr \vert tr$	Concatenation distributes over $\vert$
$\epsilon r = r\epsilon = r$	ϵ is the identity for concatenation
$r^* = (r \vert \epsilon)^*$	ϵ is guaranteed in a closure
$r^{**} = r^*$	$*$ is idempotent

Lexical errors

- It occurs if the lexical analyzer is unable to proceed because none of the patterns match remaining input
- Sometimes the analyzer tries to do recovery by
  - Deleting characters from the end
  - Replace a character by other character
  - Insert a missing character
  - Transposing two adjacent characters

  Sometimes the approach becomes too expensive