REGULAR EXPRESSIONS

Vasuki P
<VasukiP@ssn.edu.in>

Regular Expression and DFA in Compiler Design

- The scanner is the first stage in the front end
- Specifications can be expressed using regular expressions
- Build tables and code from a DFA

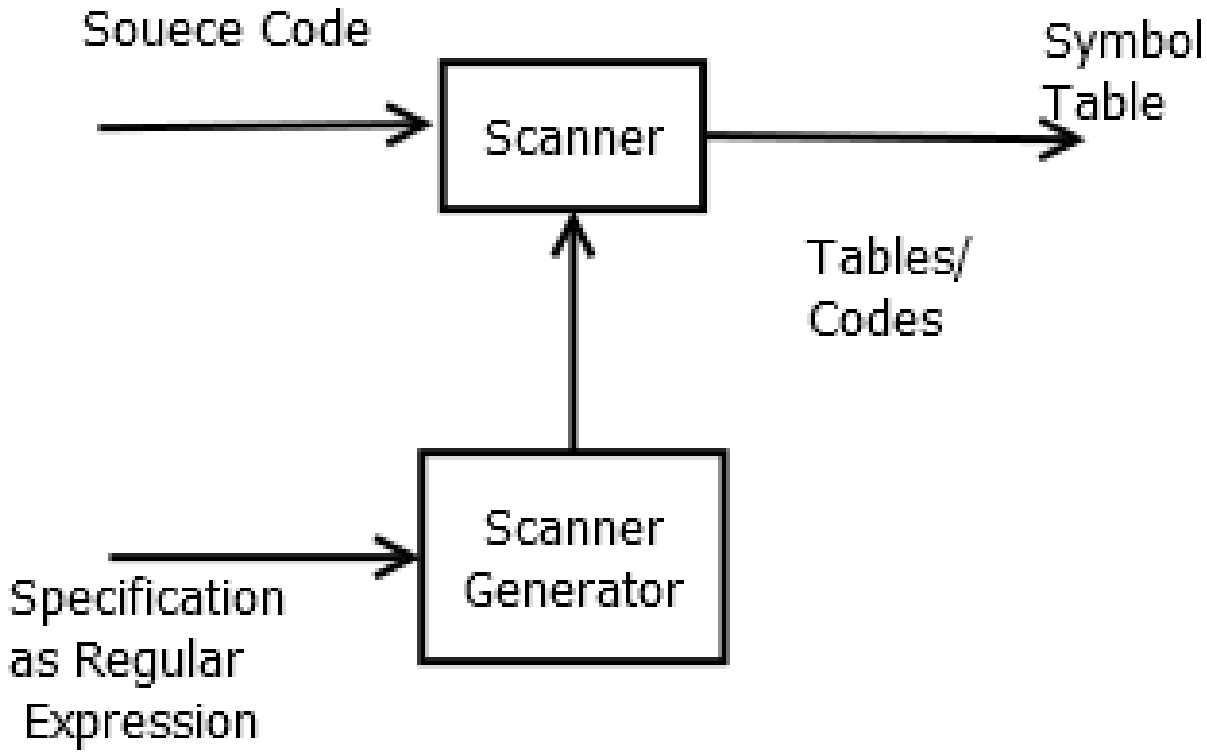

Figure: Lexical Analyzer

REGULAR EXPRESSION

- A pattern of special character s used to match strings in a search
- Typically made up from special characters called metacharacters
- Regular expressions are used thoughout UNIX:
  Editors: ed, ex, vi
  Utilities: grep, egrep, sed, and awk

RE Metacharacter	Matches…
.	**Any one character, except new line**
[a-z]	**Any one of the enclosed characters (e.g. a-z)**
*	**Zero or more of preceding character**
? or \?	**Zero or one of the preceding characters**
+ or \+	**One or more of the preceding characters**

Figure: Meta Charecters used in Regulare Expression

RE Metacharacter	Matches…
^	**beginning of line**
$	**end of line**
char	**Escape the meaning of *char* following it**
[^]	**One character not in the set**
\<	**Beginning of word anchor**
\>	**End of word anchor**
( ) or \(\)	**Tags matched characters to be used later (max = 9)**
| or |	**Or grouping**
x\{m\}	**Repetition of character x, m times (x,m = integer)**
x\{m,\}	**Repetition of character x, at least m times**
x\{m,n\}	**Repetition of character x between m and m times**

Figure: Meta Charecters used in Regulare Expression

Example: Classes

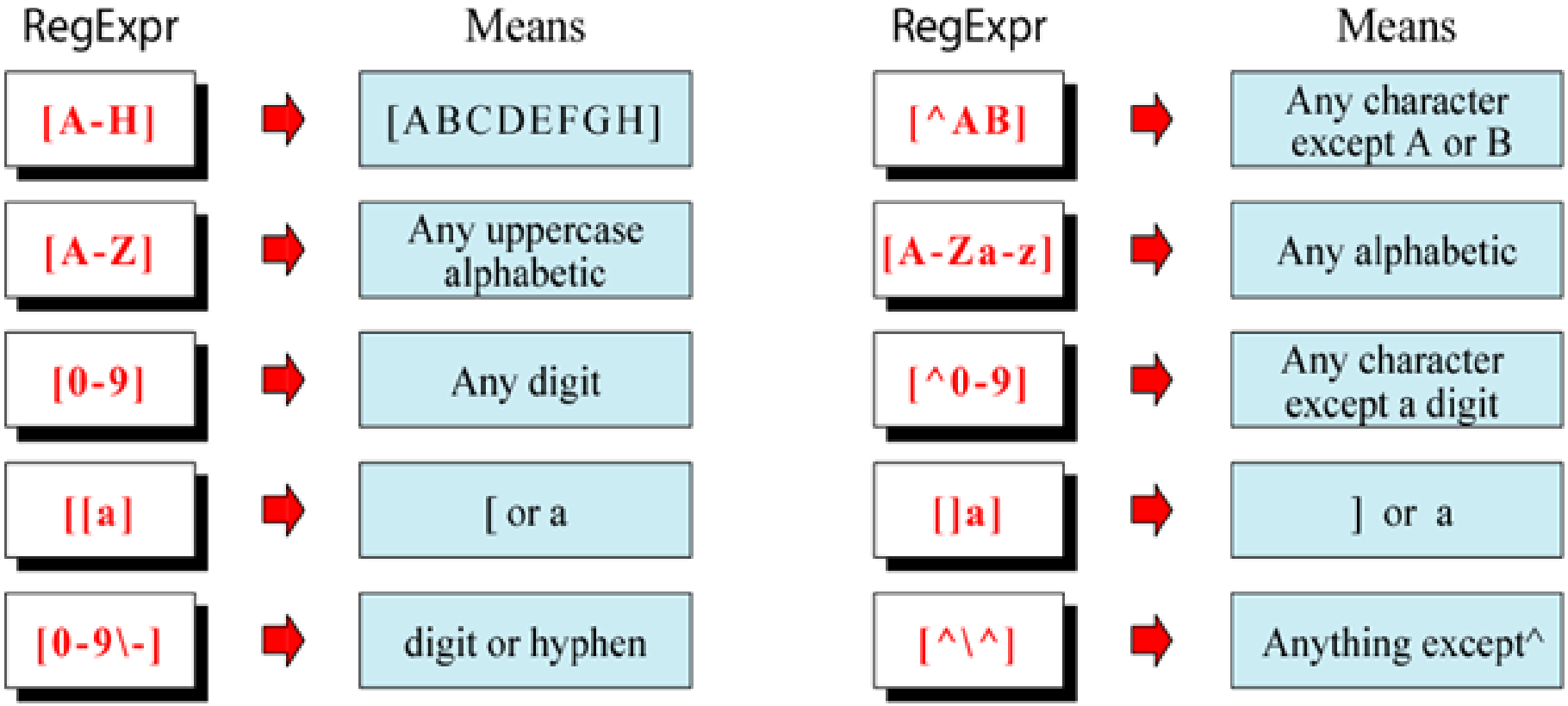

Figure: Classes of Meta charecters

Regular Grammer

- Regular expressions have the capability to express finite languages by defining a pattern for finite strings of symbols.
- The grammar defined by regular expressions is known as **regular grammar**.
- The language defined by regular grammar is known as **regular language**

Operations

- Union of two languages L and M is written as
  L U M = s | s is in L or s is in M
- Concatenation of two languages L and M is written as
  LM = st | s is in L and t is in M
- The Kleene Closure of a language L is written as L* = Zero or more occurrence of language L.

Notations

- If r and s are regular expressions denoting the languages L(r) and L(s), then
  Union : (r)|(s) is a regular expression denoting L(r) U L(s)
- Concatenation : (r)(s) is a regular expression denoting L(r)L(s)
- Kleene closure : (r)* is a regular expression denoting (L(r))*
- (r) is a regular expression denoting L(r)

Examples of Regular Expression

- All strings of 1s and 0s ending in a 1
  $(0|1) * 1$
- All strings of 1s and 0s that do not contain three 0s in a row:
  $(1 * (?|01|001)1*) * (?|0|00)$

Reference

Alfred V Aho, Monica S. Lam, Ravi Sethi and Jeffrey D Ullman, **Compilers - Principles, Techniques and Tools**, 2nd Edition, Pearson Education, 2007.