Lexical Analysis 2

Wednesday, February 10, 2021   10:22 AM

Regular Expressions

e --> epsilon -->string of length 0
L(e) = {e}

Operations
  1. Closure (Kleene *, Positive +)
  2. Concatenation
  3. Union

Regular definitions

$$d_i \notin \Sigma$$

$$r_i \in \{d_1, d_2, \ldots, d_{n-1}\}$$

Letter -> A|B….|Z|a|b|….
Digit -> 0|1|…|9
Id->letter(letter U digit)*

Unsigned numbers

23, 23E+10, 23E-10
23.10, 23.10E+10, 23.10E-10

Exponential part, fractional part

Digit -> 0|1|…|9
Digits -> (digit)+
Fraction -> .digits |e
Exponent -> (E(+|-|e)digits))|e
Number -> digit fraction exponent

? -> zero or one instance
[] -> character class
 0|1|…|9 -> [0123456789] -> [0-9]

Building a lexical analyzer
  1. Regular expression to be converted into a transition diagram
  - With nodes and edges

- Initial/start node, final/accepting node
- Forward pointer reads character by character

Nodes

edge

start

read i/p