Regular expression

Strings and languages

- Regular expression are suitable for describing tokens
- As string is a finite sequence of symbols such as “aabb” or “010”
- The length of a string is denoted by |x|
- The term language means a set of strings formed from some specific alphabet
- Simple sets
  - ϕ which denote an empty set
  - $\{\in\}$ which denotes containing only the empty string

Strings and languages

- Concatenation
  - If L and M are languages then
  - LM={xy| x is in L and y is in M}
- union
  - L U M={x|x is in L or x is in M}
  - Let L={0,01,110}
  - Let M=(00,111,1101,10}
  - LUM={0,01,110, 00,111,1101,10}
- ϕUL=LU ϕ=L
- ϕL=L ϕ= ϕ

Strings and languages

- closure
- $L^* = \bigcup_{i=0}^{\infty} L^i$

Definition of regular expression

- The regular expression over alphabet ∑ are exactly those expressions that can be constructed from the following rules
- Each regular expression denotes a language
- Any set represented by a regular expression is called a regular set

1. $\in$ is a regular expression denoting {$\in$ }, that is, the language containing the empty string
2. For each a in ∑, a is a regular expression denoting {a}
3. If R and S are regular expression denoting L_R and L_S respectively, then
   i. (R) | (S) is a regular expression denoting L_R U L_S
   ii. (R) . (S) is a regular expression denoting L_R . L_S
   iii. (R) * is a regular expression denoting L_R *

Definition of regular expression

- Identifier=letter (letter | digit)*
- (a| b)*
- a| ba*
- Two regular expression are considered equivalent
  - If?
- **Identities**
  - (PQ)*R=P(QP)*
  - (P+Q)R=PR+QR and R(Q+P)=RP+RQ
  - R*R*=R*

Class work

- Represent the following sets by regular expression
  1. {0,1,2}
  2. $\{1^{2n+1} | n>0\}$
  3. $\{a^2, a^5, a^8, ..\}$
  4. The set of all strings over {a,b} beginning with a and ends with b
  5. The set of all strings over {0,1} which has at most two zeros

Next classes topics

- Finite automata
- Deterministic automata
- Non deterministic automata
- Conversion a deterministic automata from a non deterministic automata
- Converting non deterministic finite automata from a regular expression
- Minimized the number of states of a DFA

solution

- 0+1+2
- 111(11)*
- aa(aaa)*
- a(a+b)*b
- 1*+1*01*+1*01*01*