LEX3: Regexps are Trees

Lexical Analysis

CMPT 379: Compilers

Instructor: Anoop Sarkar

anoopsarkar.github.io/compilers-class

Regular Expressions are Trees

Regular Expressions: Definition

- Note that operators apply recursively and these applications can be ambiguous
  - E.g. is aa|bc equal to a(a|b)c or ((aa)|b)c?
- Avoid such cases of ambiguity - provide explicit arguments for each regexp operator
  - For convenience, for examples on this page, let us use the symbol '$\cdot$' to denote the operator for concatenation
- Remove ambiguity with an explicit regexp tree

Regular Expressions: Definition

- Remove ambiguity with an explicit regexp tree

  a(a|b)c is written as

  (· (· a(|ab))c)

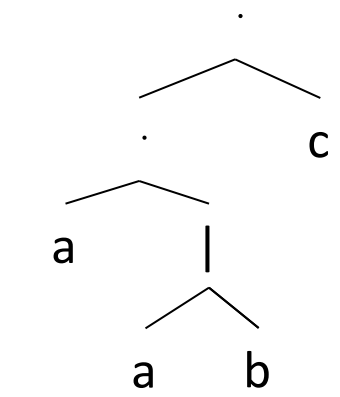
  or in postfix: aab| · c ·

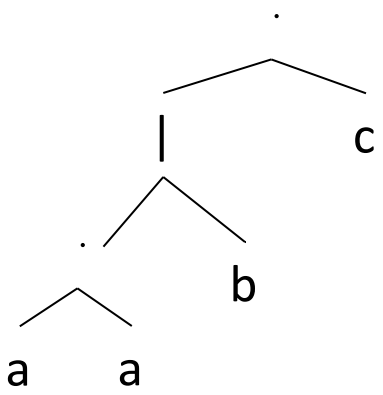
  ((aa)|b)c is written as

  (· (|(· aa)b)c)

  or in postfix: aa · b|c ·

- Does the order of concatenation matter?





Equivalence of Regexps

- (R|S)|T == R|(S|T) == R|S|T
- (RS)T == R(ST)
- (R|S) == (S|R)
- R*R* == (R*)* == R* == RR*| **ε**
- R** == R*
- (R|S)T = RT|ST
- R(S|T) == RS | RT
- (R|S)* == (R*S*)* == (R*S)*R* == (R*|S*)*
- RR* == R*R
- (RS)*R == R(SR)*
- R = R|R = R**ε**

0(10)*1|(01)* ??

Equivalence of Regexps

- 0(10)*1|(01)*
- (01)(01)*|(01)*
- (01)(01)*|(01)(01)*|ε
- (01)(01)*|ε
- (01)*

- (RS)*R == R(SR)*
- RS == (RS)
- R* == RR*|**ε**
- R == R|R
- R* == RR*| **ε**

There is a better way! To be revealed later …

(if you are impatient, web search for “DFA to regular expression”)