Lecture 3 Compilers Design

Tuesday, October 9, 2018 10:44 AM

- Contents of the lecture:-
  - Languages
    - Definition
    - Difference between ε and Φ
    - Types of languages
  - Operations on languages
    - Union
    - Concatenation
    - closure
  - Regular expressions
    - Conversion examples
    - Examples on what REs denote.
    - Priorities of calculations in REs.
    - Other examples on REs.
  - Algebraic laws for Regular Expressions.
    - Example

Languages

- A language is a set of strings all of which are chosen from some Σ*, whose Σ is a particular alphabet.
- If Σ is an alphabet, and L ⊆ Σ*"مجموعة جزئية من، أو هي نفسها", then L is said to be a language over alphabet Σ.
- Σ* is a language for any alphabet Σ
- Φ is the empty language, is a language over any alphabet (no strings).
- {ε} is the language consisting of only the empty string, is also a language over any alphabet(one string). So, Φ≠{ε}
- All alphabets are finite.
- Languages can have an infinite number of strings.
- An alphabet can be the empty language Φ.
  - What's the difference between ε and Φ ???
    - Φ is a string with no elements.
    - ε is the set of an empty character like "spaces, tabs".

- What're the types of languages ???
  - Natural language:-
    - it's normally spoken by people
    - It's like English, and Arabic, etc.
  - A formal language:-
    - It's specified precisely and is used with computers
    - Like the syntax of a programming language.
    - A formal language is a set of strings from a given alphabet.
  - Examples of languages from the alphabet {0,1}:
    - Finite sets such as {0,10,1011}, and { }
    - Infinite sets such as {ε, 0,00,000,0000,00000, ….}
    - The set of all strings of zeros and ones having an even number of ones.
      - Convert this description into mathematical expression.
        - L={ε, 000,011,01111,….}
- What are the most common operations on languages ???
  - Union:-
    - symbolized '+' or '|' or 'U'
    - Since a language is a set, the union of two sets is that set which contains all the element in each of the two sets.
    - For example:-
      - {abc, ab, ba}+{ba, bb} = {abc, ab, ba, bb}
    - The union of any language with the empty set is that language: L+ { } = L
  - Concatenation:-
    - symbolized '.' or even dot could be not written.
    - It's simply the juxtaposition"وضعهم جنباً الي جنب" of two strings forming a new string
    - Concatenation of L and M is different from M and L
    - For example:-
      - {ab, a, c}.{b, ε} = {ab.b, ab.ε, a.b, a.ε, c.b, c.ε}
      - ={abb ,ab, ab, a, cb, c}
  - Kleene Closure "L*" :-
    - If we have a language L = {0,1}
    - To get L*, we have to get :-
      - L0= {ε}
      - L1 = {0,1}
      - L2 = L.L1 = {0,1}.{0,1} = {00,01,10,11}
      - L3 = {000,001,010,011,100,101,110,111}
    - Then L* = L0 U L1 U L2 U L3 …….

- For Example:-
  - If we assume we have L and M languages, then

Operation	Definition and notation
Union of L and M	L U M = {s |"حيث" s is in L or s is in M} يعني اتحاد اللغة L مع اللغة M يدي s حيث s هي العناصر المتواجدة سواء ف L أو M.
Concatenation of L and M	LM = {st |s is in L and t is in M}
Kleene closure of L	$L^* = \bigcup_{i=0}^{\infty} L^i = \{L^0 \cup L^1 \cup L^2 \cup \ldots\}$
Positive closure of L	$L^+ = \bigcup_{i=1}^{\infty} L^i = \{L^1 \cup L^2 \cup \ldots\}$

- Another example:-
  - let L be the set of letters {A,B,…., Z,a,b, ….,z} and let D be the set of digits {0,1,2,…..,9}.
  - Some other languages can be constructed from languages L and D, using the above operators as:-
    - L U D is the set of letters and digits.
    - LD is the set of 520 strings of length two, each consisting of one letter followed by one digit.
    - L^4 is the set of all 4-letter strings
    - L* is the set of all strings of letters, including ε "the empty string".
    - L(L U D)* is the set of all strings of letters and digits beginning with a letter
    - D* is the set of all strings of one or more digits of the language D

- What are the REs "Regular Expressions" ???
  - Regular expression is the mathematical way for descripting a token "string"
  - They are expressions consisting of the three possible operations on language:-
    - Union, concatenation, and Kleene star "closure".
  - Examples on converting to REs:-
    1. An integer constant token might be defined in English as:
       - At least one decimal digit, followed by zero or more additional digits
       - "Zero or more" is recognized in Kleene star notation as *
       - So, an integer constant maybe represented as "dd*" where d represents a digit.
         - This means a constant is one digit followed by zero or more digits.
    2. Similarly, an identifier may consist of one letter (represented as a) followed by zero or more of either letters or digits:-
       - If we use a vertical bar "|" to mean "either/or," then we can express the idea ===> either an "a" or a "d", but not both by writing "a | d".
       - So, an identifier succinctly "بعد الصياغة المختصرة" can be represented as "a(a | d)*".
  - Such compact notation "الصياغة الموجزة" is called a Regular expression.
    - It's used to make an expression for the token.

  - The rules that define the regular expressions over some alphabet Σ and the languages that those expressions denote are :
    1. if ε is a regular expression, and L(ε) is {ε}.
       - It's the language of empty string.
    2. If a is a symbol in Σ, then a is a regular expression, and L(a) = {a}.
       - It's the language with one string, of length one, with a in its one position

- Examples for what an REs denote:-
  - Suppose r and s are regular expressions denoting L(r) and L(s).
  - (r) | (s) is a regular expression denoting the language L(r) U L(s).
  - (r) (s) is a regular expression denoting the language L(r) L(s).
  - (r)* is a regular expression denoting (L(r))*
  - (r) is a regular expression denoting L(r). Adding additional parentheses around expressions without changing the language they denote.

- What are the priorities of calculations in REs ???
  - Parentheses.
  - Closure.
  - Concatenation
  - Union
  - For example:-
    - L(L U D)*
      - Union is evaluated first as it's within parentheses,
      - Then closure,
      - Then concatenation.

Examples

1. Σ={a,b}
   - (a|b) (a|b) denotes {aa,ab,ba,bb}, the language of all strings or length two over the alphabet Σ.
   - Another regular expression for the same language is aa |ab | ba | bb

2. Σ = {a} ===> r = a
   - $\Sigma^* = \Sigma^0 \cup \Sigma^1 \cup \Sigma^2 \cup \ldots..$
   - $r^* = r^0 \cup r^1 \cup r^2 \cup \ldots..$
   - r* = {ε, a , aa, ……..}

3. 1.(0+1)*0 = {10,100,110,1000,1010,1100,1110,….}
   - It's the set of all strings of zero and ones which begin with a 1 and end with a 0.

- *Note:-*
  - We do not need to be concerned with the order of evaluation of several concatenations in one regular expression, since it is an associative operation. The same is true of union.

Algebraic laws for Regular Expressions

(r, s, and t are any regular expressions)

1.	$r \mid s = s \mid r$	(commutativity for alternation)
2.	$r \mid (s \mid t) = (r \mid s) \mid t$	(associativity for alternation)
3.	$r \mid r = r$	(absorption for alternation)
5.	$r(st) = (rs)t$	(associativity for concatenation)
6.	$r(s \mid t) = rs \mid rt$	(left distributivity)
7.	$(s \mid t)r = sr \mid tr$	(right distributivity)
8.	$r\epsilon = \epsilon r = r$	(identity for concatenation)
9.	$r^*r^* = r^*$	(closure absorption)
10.	$r^* = \epsilon \mid r \mid rr \mid \ldots$	(Kleene closure)
11.	$(r^*)^* = r^*$	
12.	$rr^* = r^*r$	
13.	$(r^* \mid s^*)^* = (r^*s^*)^*$	
14.	$(r^*s^*)^* = (r \mid s)^*$	
15.	$(rs)^*r = r(sr)^*$	
16.	$(r \mid s)^* = (r^*s)^*r^*$	

- Consider the regular expression a(ba|ca)|(ac|ab)a

=(aba|aca) | (aca|aba)
=(aba|(aca|(aca|aba))
=aba|((aca|aca)|aba)
=aba|(aca|aba)
=(aba|(aba|aca)
=(aba|aba)|aca
=aba|aca
=a(b|c)a