Biçimsel Diller ve Otomata Teorisi

Sunu III
Düzenli İfadeler

İZZET FATİH ŞENTÜRK

Defining Languages by Another New Method

- We wish to be very careful how we define languages
  - We presumed that we all understood exactly which values n cloud be
    - L_1 = {x^n for n = 1 2 3 …}
    - L_2 = {x^n for n = 1 3 5 7 …}
    - L_5 = {x^n for n = 1 4 9 16 …}
  - Symbols are becoming more of an IQ test than a clear definition
    - L_6 = {x^n for n = 3 4 8 22 …} ?
    - More precision and less guesswork are required, especially where computers are concerned

A More Precise Language Defining Symbolism

- L_4 = {Λ x xx xxx xxxx …}
  - The closure of a smaller set
  - Let S = {x}, then L_4 = S*
  - As shorthand L_4 = {x}*

- The use of Kleene star applied not to a set but directly to the letter x,
  - Written as a superscript as if it were an exponent
  - **x***

The Star Operator

- **x***
  - Indicates some sequence of x’s (maybe none at all)
  - Written in boldface type to distinguish it from an alphabet character

  x* = Λ or x or x^2 or x^3 or x^4 …

  x* = x^n for some n = 0 1 2 3 4 …

    - Think the star as undetermined power
    - x* stands for a string of x’s but we do not specify how many
    - An arbitrary concatenation of copies of that letter (maybe none at all)
    - Can be used to define languages, L_4 = language(x*)

The Star Operator

- L_4 = language(x*)
  - **x*** is any string of x's
  - L_4 is the set of all possible strings of x's of any length including Λ
- Suppose that we wish to describe L over the Σ = {a b} where L = {a ab abb abbb abbbb …}
  - We could summarize this language by "all words of the form one a followed by some number of b's (maybe no b's at all)"
  - We can also write L = language(**ab***)
    - The language in which the words are the concatenation of an initial a with some or no b's
    - Not string can contain a blank unless a blank is a character in Σ

The Star Operator

- Apply the Kleene star to the whole string ab
  - (ab)* = Λ or ab or abab or ababab…
  - Parantheses are not letters in the alphabet. They are used to indicate factoring
- Define the language L_1
  - L_1 = language(**xx***)
  - We start each word in L_1 by writing down an x followed by some string of x’s (maybe no more x’s at all)
  - We may also use $^+$ notation, L_1 = language(**x**$^+$)

The Star Operator

- L_1 = language(**xx***) = language(**x**$^+$) can be defined by any of the following expressions:
  - xx*
  - x$^+$
  - xx*x*
  - x*xx*
  - x$^+$x*
  - x*x$^+$
  - x*x*x*xx*
- Remmber, x* can always be Λ

Example

- The language defined by the expression: ab*a
  - The set of all strings of a’s and b’s that have at least two letters, that begin and end with a’s, and that have nothing but b’s inside (if anything at all)
  - Language(**ab*a**) = {aa aba abba abbba abbbba …}
  - It would be a mistake to say only that this language is the set of all words that begin and end with an a and have only b’s in between
    - May also apply to the word a, which is not part of the language
    - Our symbolism eliminates this ambiguity

Example

- The language of the expression: a*b*
  - Contains all the strings of a’s and b’s in which all the a’s (if any) come before all the b’s (if any)
  - Language(**ab*a**) = {Λ a b aa ab bb aaa aab abb bbb aaaa …}
  - Notice that ba and aba are not in this language
  - Notice that there need not be the same number of a’s and b’s
  - Observe that a*b* ≠ (ab)*
    - Check the string abab
    - * is not an algebraic exponent

Example

- The following expressions both define the language $L_2 = \{x^{odd}\}$

  x(xx)* or (xx)*x

- But x*xx* does not..
  - Since it includes the word (xx)x(x)

Another Use for the Plus Sign

- The expression x+y where x and y are strings of characters form an alphabet, we mean “either x or y”
    - x+y offers a choice

- Consider the language T defined over Σ={a b c}
    - T = {a c ab cb abb cbb abbb cbbb abbbb cbbbb …}
    - All the words in T begin with one a or c and followed by some b’s
    - We can write this as
        - T = language(either a or c then some b’s)
        - T= language((a+c)b*)

Example

- Consider a finite language L that contains all the strings of a’s and b’s of length three exactly
    - L = {aaa aab aba abb baa bab bba bbb}
        - The first letter of each word in L is either an a or a b
        - The second letter of each word in L is either an a or a b
        - The third letter of each word in L is either an a or a b
    - So, we may write
        - L = language$((\mathbf{a}+\mathbf{b})(\mathbf{a}+\mathbf{b})(\mathbf{a}+\mathbf{b}))$
    - Or shortly
        - L = language$((\mathbf{a}+\mathbf{b})^3)$
- Define the set of all seven-letter strings of a’s and b’s
    - $(\mathbf{a}+\mathbf{b})^7$

Example

- The set of all possible strings of letters from Σ = {a b}
  - (**a**+**b**)*

- All words that begin with the letter a
  - **a**(**a**+**b**)*

- All words that begin with an a and end with a b
  - **a**(**a**+**b**)***b**

Formal Definition of Regular Expressions

- The name of these language-defining expression: **regular expressions**
- The corresponding languages that they define: **regular languages**
- Regular expressions are of limited capacity
    - There are many interesting languages that cannot be defined by regular expressions
- The symbols that appear in regular expressions are
    - Letters in Σ, Λ, parantheses, * and +

Formal Definition of Regular Expressions

- The set of REs is defined by the following rules
  - Rule 1, Every letter of Σ can be made into a regular expression by writing it in boldface; **Λ** itself is a regular expression
  - Rule 2, If $\mathbf{r}_1$ and $\mathbf{r}_2$ are regular expression, then so are:
    - (i) $(\mathbf{r}_1)$
    - (ii) $\mathbf{r}_1\mathbf{r}_2$
    - (iii) $\mathbf{r}_1+\mathbf{r}_2$
    - (iv) $\mathbf{r}_1{}^*$
  - Rule 3, Nothing else is a regular expression
- The plus sign as a superscript is not included since $\mathbf{r}_1{}^+ = \mathbf{r}_1\mathbf{r}_1{}^*$

Example

- Consider the language defined by: (**a**+**b**)***a**(**a**+**b**)*
  - (**a**+**b**)* stands for anything
  - Then comes an a
  - Then another anything
- The language is the set of all words over Σ={a b} that have an a in them somewhere
- The only words left out are those that have only b’s and the word Λ (**b***)
- All strings = (all strings with an a) + (all strings without an a)
  - (**a**+**b**)* = (**a**+**b**)***a**(**a**+**b**)* + **b***

Example

- The language of all words that have at least two a’s:

  (a+b)*a(a+b)*a(a+b)*

- Another description for the words with at least two a’s

  b*ab*a(a+b)*

- Other descriptions..

  (a+b)*ab*ab*

  b*a(a+b)*ab*

Example

- The language of all words that have exactly two a’s:

  b*ab*ab*

- The language of all words that have at least one a and at least one b is somewhat trickier..

  (a+b)*a(a+b)*b(a+b)*

  (arbitrary)a(arbitrary)b(arbitrary)

  We require that an a precede a b in the word

  Such words as ba and bbaaaa are not included in the set

- Either a comes before b or the b comes before the a

  (a+b)*a(a+b)*b(a+b)* + (a+b)*b(a+b)*a(a+b)*

Example

- There is a simpler expression that defines the same language
- We know that $(\mathbf{a}+\mathbf{b})^*\mathbf{a}(\mathbf{a}+\mathbf{b})^*\mathbf{b}(\mathbf{a}+\mathbf{b})^*$ does not include words of the form some b’s followed by some a’s

  It is sufficient to add this specific exception to the set: $\mathbf{bb}^*\mathbf{aa}^*$

- The language of all words over Σ={a b} that contain both an a and a b:

  $(\mathbf{a}+\mathbf{b})^*\mathbf{a}(\mathbf{a}+\mathbf{b})^*\mathbf{b}(\mathbf{a}+\mathbf{b})^* + \mathbf{bb}^*\mathbf{aa}^*$

Example

- The only words that do not contain both an a and a b in them somewhere are..
  - The words of all a’s and b’s or Λ
  - When these are included, we get everything
- Therefore, all possible strings of a’s and b’s are:

  $(\mathbf{a}+\mathbf{b})^*\mathbf{a}(\mathbf{a}+\mathbf{b})^*\mathbf{b}(\mathbf{a}+\mathbf{b})^* + \mathbf{bb}^*\mathbf{aa}^* + \mathbf{a}^* + \mathbf{b}^*$
- We can write..

  $(\mathbf{a}+\mathbf{b})^* = (\mathbf{a}+\mathbf{b})^*\mathbf{a}(\mathbf{a}+\mathbf{b})^*\mathbf{b}(\mathbf{a}+\mathbf{b})^* + \mathbf{bb}^*\mathbf{aa}^* + \mathbf{a}^* + \mathbf{b}^*$

Discovering Hidden Meaning

- Not every regular expression has a simple English description
  - The regular expresssion itself can be the simplest description of the language
- For example, the following RE has no concise characterization:

  $$(\mathbf{\Lambda} + \mathbf{ba}^*)(\mathbf{ab}^*\mathbf{a} + \mathbf{ba}^*)^*\mathbf{b}(\mathbf{a}^* + \mathbf{b}^*\mathbf{a})\mathbf{bab}^*$$

  - Even if it does reduce to something simple, there is no way of knowing this
  - There is no algorithm to discover hidden meaning

Example

- $(\mathbf{a} + \mathbf{b})^* = (\mathbf{a} + \mathbf{b})^* + (\mathbf{a} + \mathbf{b})^*$
- $(\mathbf{a} + \mathbf{b})^* = (\mathbf{a} + \mathbf{b})^* + \mathbf{a}^*$
- $(\mathbf{a} + \mathbf{b})^* = (\mathbf{a} + \mathbf{b})^*(\mathbf{a} + \mathbf{b})^*$
- $(\mathbf{a} + \mathbf{b})^* = \mathbf{a}(\mathbf{a} + \mathbf{b})^* + \mathbf{b}(\mathbf{a} + \mathbf{b})^* + \mathbf{\Lambda}$
- $(\mathbf{a} + \mathbf{b})^* = (\mathbf{a} + \mathbf{b})^*\mathbf{ab}(\mathbf{a} + \mathbf{b})^* + \mathbf{b}^*\mathbf{a}^*$

$\mathbf{b}^*\mathbf{a}^*$: all a’s, all b’s, Λ, or some b’s followed by some a’s

Finite Language

- If the language L over the alphabet Σ = {a b} contains the finite list of words L = {abba baaa bbbb}
  - L = language(**abba** + **baaa** + **bbbb**)

- If L includes the null word Λ
  - L = {Λ abba baaa bbbb}
  - L = language(**Λ** + **abba** + **baaa** + **bbbb**)

Example

- The language V of all strings of a’s and b’s in which either the strings are all b’s or else there is an a followed by some b’s. V also contains the word Λ
  - V = {Λ a b ab bb abb bbb abbb bbbb …}
- We can define V by the expression:
  - **b*** + **ab***
  - (**Λ** + **a**)**b***

Example

- The language T = {a c ab cb abb cbb …}
- T can be defined by:
  - (**a** + **c**)**b***
  - **ab*** + **cb***
  - Another example of the distributive law
- The distributive law must be used with extreme caution
  - Expressions may be distributed but operators cannot
  - (**ab**)* ≠ **a*****b***

The Product Set of Strings

- We define the operation of multiplication of sets of words
- If S and T are sets of strings of letters (finite or infinite sets)
  - ST = {all combinations of a string from S concatenated with a string from T in that order}

- If S = {a aa aaa}, T = {bb bbb}
  - ST = {abb abbb aabb aabbb aaabb aaabbb} !not given in lexicographic order!

Example

- If S = {a bb bab}, T = {a ab}
  - ST = {aa aab bba bbab baba babab}

- If P = {a bb bab}, Q = {Λ bbbb}
  - PQ = {a bb bab abbbb bbbbbb babbbbb}

- If L is any language
  - LΛ = ΛL = L

Languages Associated with REs

- The following rules define the language associated with any regular expression
  - Rule 1, If the RE is just a single letter, the language is that one-letter word. If the RE is Λ, the language is just {Λ}
  - Rule 2, If r_1 and r_2 are REs associated with L_1 and L_2 respectively
    - (i) language($\mathbf{r}_1\mathbf{r}_2$) = L_1L_2
    - (ii) language($\mathbf{r}_1 + \mathbf{r}_2$) = $L_1 + L_2$
    - (iii) language($\mathbf{r}_1$*) = L_1*
- These rules prove that there is some language associated with every regular expression

Finite Languages are Regular

- If L is a finite language (a language with only finitely many words), then L can be defined by a regular expression
  - All finite languages are regular
- For example
  - L = {aa ab ba bb}
  - The RE is **aa** + **ab** + **ba**+ **bb**
  - Another RE is (**a** + **b**)(**a** + **b**)
  - The RE need not be unique. We need only show that at least one RE exists

Example

- L = (Λ x xx xxx xxxx xxxxx)
- The RE is $\mathbf{\Lambda} + \mathbf{x} + \mathbf{xx} + \mathbf{xxx} + \mathbf{xxxx} + \mathbf{xxxxx}$
- A more elegant RE is $(\mathbf{\Lambda} + \mathbf{x})^5$
  - 5 is not a legal symbol in RE
  - It means $(\mathbf{\Lambda} + \mathbf{x})(\mathbf{\Lambda} + \mathbf{x})(\mathbf{\Lambda} + \mathbf{x})(\mathbf{\Lambda} + \mathbf{x})(\mathbf{\Lambda} + \mathbf{x})$

How Hard is it to Understand a RE

- (**a** + **b**)*(**aa** + **bb**)(**a** + **b**)*
  - The set of strings that contain a double letter
  - (arbitrary)(double letter)(arbitrary)
- What strings do not contain a double letter?
  - Λ a b ab ba aba bab abab baba …
  - The expression (**ab**)* covers all but those that begin with b or end with a
  - Add these cases: (**Λ** + **b**)(**ab**)*(**Λ** + **a**)
- Combine both: (**a** + **b**)*(**aa** + **bb**)(**a** + **b**)* + (**Λ** + **b**)(**ab**)*(**Λ** + **a**)

Example

- E = (**a** + **b**)* **a** (**a** + **b**)*   (**a** + **Λ**)   (**a** + **b**)* **a** (**a** + **b**)*
  - (arbitrary) a (arbitrary) (a or nothing) (arbitrary) a (arbitrary)
  - All the words in E must have at least two a’s
- Break the middle plus sign into two cases: a or Λ
  - E = (**a** + **b**)***a**(**a** + **b**)***a**(**a** + **b**)***a**(**a** + **b**)*

    + (**a** + **b**)***a**(**a** + **b**)***Λ**(**a** + **b**)***a**(**a** + **b**)*
  - The first term: All words with at least three a’s
  - The second term: (**a** + **b**)***a**(**a** + **b**)***a**(**a** + **b**)*
- E is the union of all strings that have three or more a’s with all strings that have two or more a’s
- E is associated with (**a** + **b**)***a**(**a** + **b**)***a**(**a** + **b**)*

Repeated Application of the Rules

- (**a** + **b***)* = (**a** + **b**)*
- (**a***)* = **a***

However

- (**aa** + **ab***)* ≠ (**aa** + **ab**)*
    - Check the word abbabb
    - The language defined by the RE on the right cannot contain any word with a double b

Example

- (**a** ***b***)*
  - Factors of the form **a*****b***
  - Both the signle letters a and b are words of the form **a*****b***
  - This language contains all strings of a’s and b’s
  - Therefore, (**a** ***b***)* = (**a** + **b**)*

- The possibility of finding a set of algebraic rules to reduce one regular expression to another equivalent one
  - Still unknown whether this can be done

Example

- **b***(**abb***)*(**Λ** + **a**)
  - The language of all words without a double a
  - The word starts with some b’s
  - Then come repeated factors of the **abb*** (a followed by at least one b)
  - Finish with a final a or leave the last b

- (**a** + **b**)* = (**a** + **b**)***aa**(**a** + **b**)* + **b***(**abb***)*(**Λ** + **a**)

EVEN-EVEN

- E = [**aa** + **bb** + (**ab** + **ba**)(**aa** + **bb**)*(**ab** + **ba**)]*
- This RE represents the collection of all words of three types
  - type$_1$ = aa
  - type$_2$ = bb
  - type$_3$ = (ab + ba)(aa + bb)*(ab + ba)
  - E = [type$_1$ + type$_2$ + type$_3$]*

- It has an even number of a’s and an even number of b’s
- EVEN-EVEN = {Λ aa bb aaaa aabb abab abba baab baba bbaa bbbb aaaaaa aaaabb aaabab …}