Theory of Computation

النظرية الاحتسابية
المحاضرة الخامسة والسادسة

كلية التربية للعلوم الصرفة / جامعة ديالى

اعداد
م.د. محمد سامي محمد

قسم علوم الحاسوب – المرحلة الثانية

REGULAR EXPRESSIONS

EXAMPLE

Another expression that denotes all the words with at least two a's is:

$$b^*ab^*a(a + b)^*$$

We scan through some jungle of b's (or no b's) until we find the first a, then more b's (or no b's), then the second a, then we finish up with anything.

In this set are *abbbabb* and aaaaa.

We can write:

$$(a + b)^*a(a + b)^*a(a + b)^* = b^*ab^*a(a + b)^*$$

where by the equal sign we do not mean that these expressions are equal algebraically in the same way as

$$x+x=2x$$

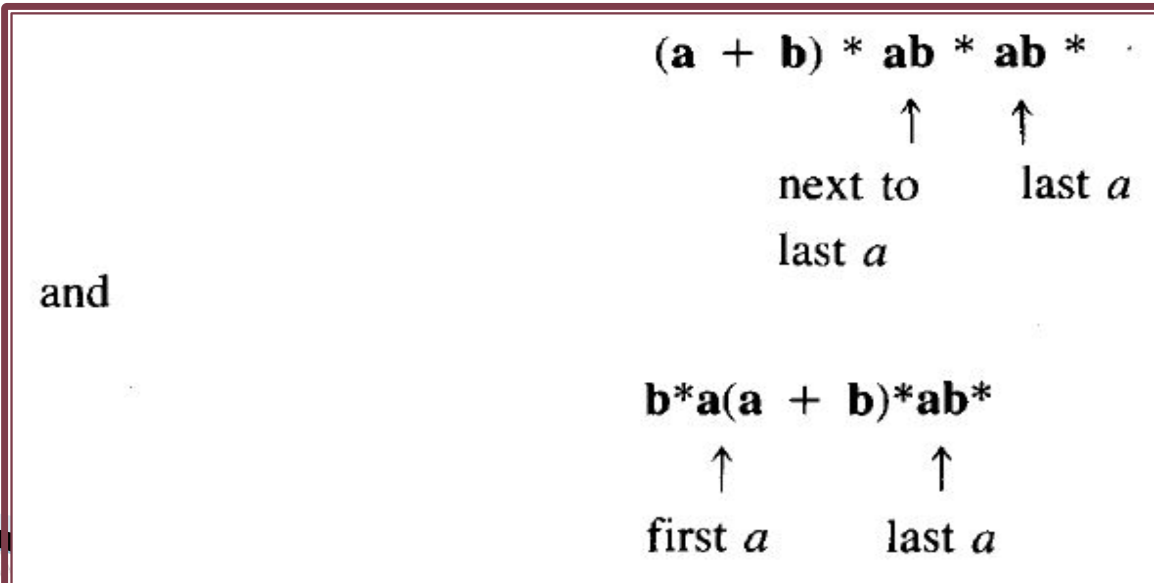
$$(a + b)^*a(a + b)^*a(a + b)^* = b^*ab^*a(a + b)^*$$

Are they equal ?

REGULAR EXPRESSIONS

We could write
language $((a + b)^*a(a + b)^*a(a + b)^*)$
$= $ language $(b^*ab^*a (a + b)^*)$
= all words with at least two a's.

To be careful about this point, we say that two regular expressions are equivalent if they describe the same language.



REGULAR EXPRESSIONS

EXAMPLE

If we wanted all the words with *exactly* two a's, we could use the expression

b*ab*ab*

which describes such words as *aab, baba,* and *bbbabbbab.*

To make the word *aab,* we let the first and second b* become Λ and the last becomes b.

b*ab*ab*

Prove is these one of the solutions and how?

aab

baba

and *bbbabbbab*

REGULAR EXPRESSIONS

EXAMPLE

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

$$(a + b)^*a(a + b)^* b(a + b)^*$$

$$= (\text{arbitrary})\ a(\text{arbitrary})\ b(\text{arbitrary})$$

we could define this set by the expression:

$$(a+b)^*a(a+b)^*b(a+b)^* + (a+b)^*b(a+b)^*a(a+b)^*$$

Here we are still using the plus sign in the general sense of disjunction (or). We are taking the union of two sets,

(a + b)*a(a + b)* b(a + b)* One a and one b always

(a+b)*a(a+b)*b(a+b)* + (a+b)*b(a+b)*a(a+b)* Are they equal ?

REGULAR EXPRESSIONS

يمكن التعبير عن الصيغ بصيغ اخرى مشابهة

EXAMPLE

All temptation to treat these language-defining expressions as if they were algebraic polynomials should be dispelled by these equivalences:

$(a+b)^* = (a+b)^* + (a+b)^*$

$(a+b)^* = (a+b)^* (a+b)^*$

$(a+b)^* = a(a+b)^* + b(a+b)^* + \Lambda$

$(a+b)^* = (a+b)^* ab(a+b)^* + b^*a^*$

$(a+b)^* = (a+b)^* + (a+b)^*$

$(a+b)^* = (a+b)^* (a+b)^*$

Check these?

$(a+b)^* = a(a+b)^* + b(a+b)^* + \Lambda$

$(a+b)^* = (a+b)^* ab(a+b)^* + b^*a^*$

REGULAR EXPRESSIONS

Usually when we employ the star operator, we are defining an infinite language.
We can represent a finite language by using the plus (union sign) alone.
If the language L over the alphabet X = {a, *b}* contains only the finite list of words given below,

$$L = \{abba\ baaa\ bbbb\}$$

then we can represent L by the symbolic expression

$$L = \text{language } (abba + baaa + bbbb)$$

If L is a finite language that includes the null word Λ, then the expression that defines L must also employ the symbol Λ.
For example, if

$$L = \{\Lambda\ a\ aa\ bbb\}$$

then the symbolic expression for L must be

$$L = \text{language } (\Lambda + a + aa + bbb)$$

L = {abba baaa bbbb}

What is Regular Expression of this language?

REGULAR EXPRESSIONS

Let us reconsider the language
$T = \{a\ c\ ab\ cb\ abb\ cbb\ \ldots\}$.

T can be defined as above by

$$(a + c)b^*$$

but it can also be defined by

$$ab^* + cb^*$$

This is another example of the distributive law.

T = {a c ab cb abb cbb ... }.

The distributive law

(a + c)b*

REGULAR EXPRESSIONS

If r1 = aa + b then the expression r1* technically refers to the expression

r1* = aa + b*

which is the formal concatenation of the symbols for r, with the symbol *, but what we generally mean when we write r1* is actually (r1)*

(r1)* = (aa + b)*

r1 = aa + b

r1*=

REGULAR EXPRESSIONS

DEFINITION

If S and T are sets of strings of letters (whether they are finite or infinite sets), we define the product set of strings of letters to be

$$ST = \{\text{all combinations of a string from S concatenated with a string from T}\}$$

EXAMPLE

If

$$S = \{a\ aa\ aaa\} \text{ and } T = \{bb\ bbb\}$$

then

$$ST = \{abb\ abbb\ aabb\ aabbb\ aaabb\ aaabbb\}$$

EXAMPLE

If $S = \{a\ bb\ bab\}$ $T = \{a\ ab\}$

Then $ST = \{aa\ aab\ bba\ bbab\ baba\ babab\}$

$S = \{a\ aa\ aaa\}$ and $T = \{bb\ bbb\}$

Then $ST = \{abb\ abbb\ aabb\ aabbb\ aaabb\ aaabbb\}$

If $S = \{a\ bb\ bab\}$ $T = \{a\ ab\}$

Then ST =

REGULAR EXPRESSIONS

EXAMPLE

If $P = \{a\ bb\ bab\}$ *and* $Q = \{\Lambda\ bbbb\}$

then

$PQ = \{\ a\ bb\ bab\ abbbb\ bbbbbb\ babbbbb\}$

ملاحظة :- فقط في حالة الصفر موجود في المجموعتين يكون في الناتج صفر

EXAMPLE

If $M = \{\Lambda\ x\ xx\}\ N\text{-}=\{\Lambda\ y\ yy\ yyy\ yyyy\ldots\}$

then

$MN = \{\Lambda$ y yy *yyy yyyy* **...**

x xy xyy xyyy xyyyy...

xx **xxy** xxyy xxyyy xxyyyy.. .}.

If $P = \{a\ bb\ bab\}$ *and* $Q = \{\mathbf{\Lambda}\ bbbb\}$

Then

$PQ =$

If $M = \{\mathbf{\Lambda}\ \mathrm{x}\ xx\}$

$N=\{\mathbf{\Lambda}\ \mathrm{y\ yy\ yyy\ yyyy}...\}$

Then MN =

REGULAR EXPRESSIONS

Using regular expressions, these four examples can be written as:

(a + aa + aaa)(bb + bbb) = abb + abbb + aabb + aabbb + aaabb + aaabbb

(a + bb + bab)(a + ab) = aa + aab + bba + bbab + baba + babab

(a + bb + bab)(Λ + bbbb) = a+bb+bab+abbbbb +bbbbbb + babbbbb

(Λ + x + xx)(y*) = y* + xy* + xxy*

REGULAR EXPRESSIONS

THEOREM 5
If L is a finite language (a language with only finitely many words), then L can be defined by a regular expression.

PROOF

For example, the regular expression that defines the language

L = {baa abbba bababa}

is

baa + abbba + bababa

Another example If

L = {aa ab ba bb}

the algorithm described above gives the regular expression

aa + ab + ba + bb

Another regular expression that defines this language is

(a + b)(a + b)

L = {baa abbba bababa}

R.E.= baa + abbba + bababa

L = {aa ab ba bb}

R.E.= aa + ab + ba + bb

REGULAR EXPRESSIONS

EXAMPLE

Let

$$L = \{\Lambda \ x \ xx \ xxx \ xxxx \ xxxxx\}$$

The regular expression we get from the theorem is

$$\Lambda + x + xx + xxx + xxxx + xxxxx$$

Λ more elegant regular expression for this language is

$$(\Lambda + x)^5$$

Of course the 5 is, strictly speaking, not a legal symbol for a regular expression although we all understand it means

$$(\Lambda + x)(\Lambda + x)(\Lambda + x)(\Lambda + x)(\Lambda + x)$$

$L = \{\Lambda\ x\ xx\ xxx\ xxxx\ xxxxx\}$

$\Lambda + x + xx + xxx + xxxx + xxxxx$

$(\Lambda + x)^5$

REGULAR EXPRESSIONS

EXAMPLE

Consider the expression:

$$(a + b)^*(aa + bb)(a + b)^*$$

This is the set of strings of a's and b's that at some point contain a double letter. We can think of it as

(arbitrary)(double letter)(arbitrary)

Example are: Λ a *b ab ba aba bab abab baba* .... The expression (ab)* covers all of these except those that begin with b or end in a. Adding these choices gives us the regular expression

$$(\Lambda + b)(ab)^*(\Lambda + a)$$

REGULAR EXPRESSIONS

EXAMPLE

Consider the regular expression below:

$$E = (a + b)^*a(a + b)^* (a + \Lambda) (a + b)^*a(a + b)^*$$

= (arbitrary) a (arbitrary) [a or nothing] (arbitrary) a (arbitrary).

$$= (a+b)^*a(a+b)^*a(a+b)^*a(a+b)^*+(a + b)^*a(a + b)^* \Lambda(a + b)^*a(a + b)^*$$

Before we analyze the second term let us make the observation that

$$(a + b)^* \Lambda(a + b)^*$$

Simplify it?

$$(a + b^*)^* (aa + ab^*)^* ((a + bbba^*) + ba^*b)^*$$

REGULAR EXPRESSIONS

EXAMPLE

Consider the regular expression:

$$(a^*b^*)^*$$

The language defined by this expression is all strings that can be made up of factors of the form a*b*, but since both the single letter a and the single letter b are words of the form a*b*, this language contains all strings of a's and b's. It cannot contain more than everything, so

$$(a^*b^*)^* = (a + b)^*$$

Prove it?

$$(a^*b^*)^* = (a + b)^*$$

REGULAR EXPRESSIONS

EXAMPLE

One very interesting example, which we consider now in great detail is

$$E = [aa + bb + (ab+ba)(aa+bb)^*(ab+ba)]^*$$

This regular expression represents the collection of all words that are made up of "syllables" of three types:

typeI = aa

type2 = bb

type3 = (ab + ba)(aa + bb)*(ab + *ba)*

$$E = [\text{type}, + \text{type2} + \text{type 3}]^*$$

REGULAR EXPRESSIONS

EXAMPLE

Consider the language defined by the regular expression:

$$b^*(abb^*)^*(\Lambda + a)$$

This is the language of all words without a double a.