Chapter 4 solutions

Automata Theory (University of the Punjab)

Chapter 4, Problem 1P Bookmark Show all steps: ON

Let **r1, r2,** and r3 be three regular expressions. Show that the language associated with (**r1+ r2)r3**is the same as the language associated with **r1r3 + r2r3.** Show that **r1(r2 + r3)** is equivalent to **r1r2 + r2r3.** This will be the same as proving a "distributive law" for regular expressions.

Step-by-step solution

Step 1 of 2

The r_1, r_2, r_3 are three regular expressions and L_1, L_2, L_3 are three languages

If regular expressions r_1 generates the language L_1, regular expressions r_2 generates the language L_2, and regular expressions r_3 generates the language L_3, then the $(r_1+r_2)\,r_3$ generates to the $(L_1+L_2)\,L_3$ languages and $(r_1\,r_3+r_2\,r_3)$ generates to the $(L_1+L_2)L_3$ languages

$\therefore$ $(r_1+r_2)\,r_3$ and $(r_1\,r_3+r_2\,r_3)$ generates the same language then

$(r_1+r_2)\,r_3 = (r_1\,r_3+r_2\,r_3)$

language L_2, and regular expressions r_3 generates the language L_3, then the

Chapter 4, Problem 1P Bookmark Show all steps: ON

$\therefore$ $(r_1+r_2)\,r_3$ and $(r_1\,r_3+r_2\,r_3)$ generates the same language then

$(r_1+r_2)\,r_3 = (r_1\,r_3+r_2\,r_3)$

Comment

Step 2 of 2

Similarly $r_1(r_2+r_3)$ generates the $L_1(L_2+L_3)$ languages and $(r_1\,r_2+r_1\,r_3)$ generates the $L_1(L_2+L_3)$ so

$\therefore r_1(r_2+r_3) = (r_1\,r_2+r_1\,r_3)$

Comment

Was this solution helpful? 19 6

Chapter 4, Problem 2P

5 Bookmarks

Show all steps: ON

Construct a regular expression defining each of the following languages over the alphabet Σ = {*a* *b*}:

All words in which a appears tripled, if at all. This means that every clump of *a*'s contains 3 or 6 or 9 or 12 . . . *a*'s.

Step-by-step solution

Step 1 of 1

Construct regular expression

Input alphabet $\Sigma = \{a\ b\}$

Generate words as $'a'$ **tripled**

Every clump of a's contains 3 or 6 or 9 or 12...*a*'s $(aaa)^*$

In a language any occurrences of *a's* and *b's* are possible, whereas *a's* are always tripled or multiple of three.

Construct regular expression *R*,

$R = (b + aaa)^*$

Examples of strings generated $baaa, aaaaaab, bbaaaaaabbaaa, etc$

Hence, the regular expression is $\boxed{(b + aaa)^*}$ **.**

Comments (3)

Chapter 4, Problem 3P

1 Bookmark

Show all steps: ON

Construct a regular expression defining each of the following languages over the alphabet Σ = {*a* *b*}:

All words that contain at least one of the strings *s*1 *s*2, *s*3 or *s*4.

Step-by-step solution

Step 1 of 1

Assume Anystring(a+b)*

At least one of the substrings as s1, s2, s3 or s4:

(a+b)*(s1+s2+s3+s4) (s1+s2+s3+s4)*(a+b)*

Comments (4)

Equivalent solution: (a+b)* (s1+s2+s3+s4)+ (a+b)*

The above comment uses "+" in (s1+s2+S3+S4)+ to indicate "one or more" just as * indicates 0 or more.

Is (a+b)*(s1+s2+s3+s4)* true?

Chapter 4, Problem 4P

3 Bookmarks Show all steps: ON

Construct a regular expression defining each of the following languages over the alphabet Σ = {*a* *b*} :

All words that contain exactly two *b*'s or exactly three *b*'s, not more.

Step-by-step solution

Step 1 of 3

A regular expression is a set of values of the algebraic formulas whose value is a pattern. Every pattern in a regular expression holds the set of strings. Simply, it is used to generate patterns of strings.

Comment

Step 2 of 3

Consider the following language over the alphabet $\sum = \{a, b\}$:

All words that contain exactly two b's or exactly three b's, not more.

The **regular expression** of the above language is as follows:

$$\boxed{\begin{array}{c}\mathbf{a^*ba^*ba^*} + \mathbf{a^*ba^*ba^*ba^*} \\ or \\ \mathbf{a^*ba^*ba^*} \mid \mathbf{a^*ba^*ba^*ba^*}\end{array}}$$

Chapter 4, Problem 5P

3 Bookmarks Show all steps: ON

Construct a regular expression defining each of the following languages over the alphabet Σ = {*a* *b*} :

(i) All strings that end in a double letter.

(ii) All strings that do not end in a double letter.

Step-by-step solution

Step 1 of 2

I) $(a+b)^*(aa+bb)$ -the string end with a double letter

Comment

Step 2 of 2

II) $(a+b)^*(ab+ba)+a+b+\in$ -the string do not end in a double letter

Comments (2)

Was this solution helpful? 32 6

Chapter 4, Problem 6P

7 Bookmarks

Show all steps: ON

Press Esc to exit full screen

Problem

Construct a regular expression defining each of the following languages over the alphabet Σ = {*a* *b*} :

All strings that have exactly one double letter in them.

Step-by-step solution

Step 1 of 1

$(b+\epsilon)(ab)^{*}aa(ba)^{*}(b+\epsilon)+(a+\epsilon)(ba)^{*}bb(ab)^{*}(a+\epsilon)$ -the string have exactly one double letter

Comment

Was this solution helpful? 9 2

Chapter 4, Problem 7P

3 Bookmarks

Show all steps: ON

Problem

Construct a regular expression defining each of the following languages over the alphabet Σ = {*a* *b*} :

All strings in which the letter *b* is never tripled. This means that no word contains the substring *bbb*.

Next

Step-by-step solution

Step 1 of 1

$(a+ba+bba)^{*}(bb+b+\varepsilon)$ - here no word contains the substring bbb

Comment

Was this solution helpful? 50 12

Chapter 4, Problem 8P

5 Bookmarks

Show all steps: ON

All words in which *a* is tripled or *b* is tripled, but not both. This means each word contains the substring aaa or the substring *bbb* but not both.

Step-by-step solution

Step 1 of 2

Consider the language *L* over the alphabet $\Sigma = \{a \ \ b\}$ that accepts the words in which *a* is tripled or *b* is tripled but not both. The words that can be accepted by the language are $\{aaa, bbb, bbaaa, aabbb, baaaa, \ldots\}$.

Comment

Step 2 of 2

The whole regular expression for the strings with *aaa* or *bbb* as substring is as follows:

$$(a+ba+bba)^*(bb+b+\varepsilon)aaa(a+ba+bba)^*(bb+b+\varepsilon)+$$
$$(b+ab+aab)^*(aa+a+\varepsilon)bbb(b+ab+aab)^*(aa+a+\varepsilon)$$

Comment

Chapter 4, Problem 9P

3 Bookmarks

Show all steps: ON

Construct a regular expression defining each of the following languages over the alphabet Σ = {*a* *b*} :

(i) All words that do not have the substring *ab*.

(ii) All words that do not have both the substrings *bba* and *abb*.

Step-by-step solution

Step 1 of 5

i)

A regular expression is a set of values of the algebraic formulas whose value is a pattern. Every pattern in a regular expression holds the set of strings. Simply, it is used to generate patterns of strings.

Comment

Step 2 of 5

Consider the alphabet $\sum = \{a, b\}$

Comment

Step 2 of 5

Consider the alphabet $\sum=\{a,b\}$
Consider the language: "*All words that do not have the substring ab.*"

So, the regular expression of the language is as follows:

$$\boxed{\begin{array}{c}(\mathbf{a}+\mathbf{b})^{*}(\mathbf{a}+\mathbf{bb})\\ \text{(or)}\\ \mathbf{b}^{*}\mathbf{a}^{*}\end{array}}$$

Comments (1)

Step 3 of 5

Chapter 4, Problem 9P

3 Bookmarks

Show all steps: ON

Step 3 of 5

Explanation:

- **The symbol "+" represents the union of two strings. That means, if a and b are regular expressions, then a+b (also written as a | b or a∪b) is also a regular expression.**
- **The regular expression contains " (a+b) *" stands for a string of a's or b's. It is containing any length of a's or b's or null (∧) in the expression.**
- **The (a+b) represents word should end with "a'" or "b".**

Now take the above regular expression for check words:

$(\mathbf{a}+\mathbf{b})^{*}(\mathbf{a}+\mathbf{bb}) \Rightarrow aa, bba, ba, babb, abb, \ldots$

Here, observe that the word should not end with the substring ab. It should end with "a" or "bb".

Now take another regular expression for check words:
b*a*=ba, bba, baaa, bbbbba,.....

Here, observe that the word should not end with the substring ab.

Comment

Chapter 4, Problem 10P

4 Bookmarks

Show all steps: ON

Problem

Construct a regular expression defining each of the following languages over the alphabet Σ = {*a* *b*} :

All strings in which the total number of *a*'s is divisible by 3 no matter how they are distributed, such as *aabaabbaba*.

Step-by-step solution

Step 1 of 1

Regular Expression :

$(b^*ab^*ab^*ab^*)^*b^*$

The first expression has * around it, it can occur 0 or more times to give us any number of a's that is divisible by 3

Comment

Chapter 4, Problem 11P

4 Bookmarks

Show all steps: ON

Construct a regular expression defining each of the following languages over the alphabet Σ = {*a* *b*} :

(i) All strings in which any *b*'s that occur are found in clumps of an odd number at a time, such as *abaabbbab*.

(ii) All strings that have an even number of *a*'s and an odd number of *b*'s.

(iii) All strings that have an odd number of *a*'s and an odd number of *b*'s.

Step-by-step solution

Step 1 of 3

i) The regular expression for the strings for any b's that are found in clumps of an odd number at a time is,

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

Compulsory a after some number of odd b's are there. ODD+ODD = EVEN, so here needed to separate the odd clumps.

Comment

Chapter 4, Problem 11P

4 Bookmarks

Show all steps: ON

Step 2 of 3

ii) The regular expression for strings of even number of a's and odd number of b's

Divide this language to two groups:

a) When words that start with b and followed by even number of a's and even number of b's. It becomes odd number of b's and even number of a's

b) When words that start with a and followed by odd number of a's and odd number of b's. It also becomes odd number of b's and even number of a's

b[aa+bb+(ab+ba)(aa+bb)*(ab+ba)]*+

a[[aa+bb+(ab+ba)(aa+bb)*(ab+ba)]*(ab+ba)[aa+bb+(ab+ba)(aa+bb)*(ab+ba)]*]

Comment

Step 3 of 3

iii) The regular expression for the strings of odd number of a's and odd number of b's .

The small string is ab or ba we can add even letter string left or right or both.

[aa+bb+(ab+ba)(aa+bb)*(ab+ba)]*(ab+ba)[aa+bb+(ab+ba)(aa+bb)*(ab+ba)]*

Comments (1)

Chapter 4, Problem 12P

2 Bookmarks

Show all steps: ON

Problem

(i) Let us reconsider the regular expression

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

Show that this is equivalent to

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

in the sense that they define the same language.

(ii) Show that

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

(iii) Show that

$(\mathbf{a} + \mathbf{b})^*\mathbf{ab}[(\mathbf{a} + \mathbf{b})^*\mathbf{ab}(\mathbf{a} + \mathbf{b})^* + \mathbf{b}^*\mathbf{a}^*] + \mathbf{b}^*\mathbf{a}^* = (\mathbf{a} + \mathbf{b})^*$

(iv) Is (iii) the last variation of this theme or are there more beasts left in this cave?

Step-by-step solution

Step 1 of 4

(i)

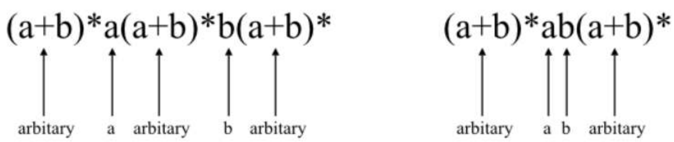
Step 1 of 4

(i)

Consider the regular expressions, $R1=(a+b)^*a(a+b)^*b(a+b)^*$ and $R2=(a+b)^*ab(a+b)^*$.

The two regular expressions are said to be equivalent if both R1 and R2 accept the same language.

The R1 accepts all the strings that contain at least one 'a' and one 'b'. The R2 accepts all the strings that contains the substring 'ab'. In both regular expressions 'a' comes before 'b'.



The regular expression $(a+b)^*$ accepts all the strings over the input alphabet {a,b}. Both R1 and R2 starts with $(a+b)^*a$ and ends with $b(a+b)^*$. In the middle part, the regular expression R1 contains an arbitrary $(a+b)^*$ followed by b whereas the regular expression R2 contains b. The arbitrary part can be ignored. Both the regular expressions accept an 'a' followed by 'b'. Therefore, the regular expression R1 is equivalent to R2.

Comment

Chapter 4, Problem 12P   2 Bookmarks   Show all steps: ON

Comment

Step 2 of 4

(ii)

Consider the regular expressions, $R1=(a+b)^*ab(a+b)^*+b^*a^*$ and $R2=(a+b)^*$.

The two regular expressions are said to be equivalent if both R1 and R2 accept the same language.

The regular expression $(a+b)^*$ accepts all the strings over the input alphabet {a,b}. The R1 accepts all the strings that contains the substring 'ab' whereas the R2 accepts all the strings that are accepted by the R1. Therefore, the regular expression R1 is equivalent to R2.

Comment

Step 3 of 4

(iii)

Consider the regular expressions, $R1=(a+b)^*ab[(a+b)^*ab(a+b)^*+b^*a^*]+b^*a^*$ and $R2=(a+b)^*$.

are accepted by the R1. Therefore, the regular expression R1 is equivalent to R2.

Comment

Step 3 of 4

(iii)

Consider the regular expressions, $R1=(a+b)^*ab[(a+b)^*ab(a+b)^*+b^*a^*]+b^*a^*$ and $R2=(a+b)^*$.

The two regular expressions are said to be equivalent if both R1 and R2 accept the same language.

$$\begin{aligned}R1&=(a+b)^*ab[(a+b)^*ab(a+b)^*+b^*a^*]+b^*a^*\\&=(a+b)^*ab(a+b)^*ab(a+b)^*+(a+b)^*abb^*a^*+b^*a^*\end{aligned}$$

The regular expression $(a+b)^*$ accepts all the strings over the input alphabet {a,b}. The regular expression R1 accepts all the strings including null string that are accepted by $(a+b)^*$ which is R2. Therefore, the regular expression R1 is equivalent to R2.

Comment

Step 4 of 4

language.

$$\begin{aligned}R1&=(a+b)^*ab[(a+b)^*ab(a+b)^*+b^*a^*]+b^*a^*\\&=(a+b)^*ab(a+b)^*ab(a+b)^*+(a+b)^*abb^*a^*+b^*a^*\end{aligned}$$

The regular expression $(a+b)^*$ accepts all the strings over the input alphabet {a,b}. The regular expression R1 accepts all the strings including null string that are accepted by $(a+b)^*$ which is R2. Therefore, the regular expression R1 is equivalent to R2.

Comment

Step 4 of 4

(iv)

The part (iii) is not the last variant to this theme. There are many other variant regular expressions exist which are equivalent to $(a+b)^*$.

Comment

Was this solution helpful? 1 0

Chapter 4, Problem 13P

Bookmark

Show all steps: ON

Problem

We have defined the product of two sets of strings in general. If we apply this to the case where both iactors are the same set, *S* = *T*, we obtain squares, *S*2. Similarly, we can define *S*3, *S*4, ... Show that it makes some sense to write:

(i) $S^* = \Lambda + S + S1 + S2 + S3 + S4 + \ldots$

(ii) $S+ = S + S1 + S2 + S3 + S4+ \ldots$

Step-by-step solution

Step 1 of 1

For two identical sets S=T, product of them is equal to S^2, for three of them - S^3 and so on. Now we have to prove that:

a) $S^* = \lambda + S + S^1 + S^2 + S^3 + \ldots$

Chapter 4, Problem 13P

Bookmark

Show all steps: ON

a) $S^* = \lambda + S + S^1 + S^2 + S^3 + \ldots$

OK, to understand this relation we need to think about the nature of Kleene closure. It is obtained by concatenaiting with each other a finite number of words belonging to a certain language + λ. It means that this concatenation can always be presented as a combination of basic words "multiplied" by each other.

In this theorem, the RHS is containing all possible sets generated by multiplying S by S finitely many times. In other words, by doing so we get all possible words of all lengths comprised of basic words belonging to S e.g.:

S = {aa bb}

Then S* = {λ aa bb aaaa aabb bbaa bbbb aaaaaa aaaabb ...}

RHS = λ + {aa bb} + {aaaa aabb bbaa bbbb} + {aaaaaa aaaabb ... bbbbbb} +... = S*

Chapter 4, Problem 13P

Bookmark

Show all steps: ON

b) In this case we need to prove that:

$S^+ = S + S^1 + S^2 + S^3 + \ldots$

This is a concatenation of two sets R and S, denoted RS, is {st|s ε S ad t ε R}. We will sometimes write $S^2$for SS, the concatenation of S with itself, and S^3 for SSS {SS^2}.This set does not contain λ.

By analogy we see that the only difference is that both sets does not contain λ, so the theorem from the first part still holds.

Comment

Was this solution helpful? 0 0

Chapter 4, Problem 14P

Bookmark

Show all steps: ON

Problem

If the only difference between *L* and *L** is the word Λ, is the only difference between *L*2 and *L** the word **Λ**?

Step-by-step solution

Step 1 of 1

The difference between L^2 and $L*$ is the word $\wedge$, and words of length more than 2. Because $L*$ contains all words of L^2, but L^2 does not contain all words of L*.

Comment

Was this solution helpful? 3 0

Chapter 4, Problem 15P

3 Bookmarks

Show all steps: ON

Problem

Show that the following pairs of regular expressions define the e language over the alphabet Σ = {*a b*}:

(i) (**ab**)*__a__ and **a**(**ba**)*

(ii) (**a*** + **b**)* and (**a** + **b**)*

(iii) (**a*** + **b***)* and (**a** + **b**)*

Step-by-step solution

Step **1** of 3

Two Regular expressions define the same language

i) $(ab)^* a$ and $a(ba)^*$

The language defined by the expression $(ab)^* a$

Chapter 4, Problem 15P

3 Bookmarks

Show all steps: ON

Step-by-step solution

Step **1** of 3

Two Regular expressions define the same language

i) $(ab)^* a$ and $a(ba)^*$

The language defined by the expression $(ab)^* a$

Set of all strings of a's and b's, that have at least 'a' as string and that have nothing but b's inside.

Language1 $(ab)^* a$,

$$(ab)^* = \{\wedge, ab, abab, ababab\}$$
$$(ab)^* a = \{a, aba, ababa, abababa\}$$

The language defined by the expression $a(ba)^*$

Set of all strings of a's and b's, that have at least 'a' as string and that have nothing but b's inside.

Language2 $a(ba)^*$,

$$(ba)^* = \{\wedge, ba, baba, bababa\}$$
$$a(ba)^* = \{a, aba, ababa, abababa\}$$

Hence Language1 and Language2 define the same set of language.

Step 2 of 3

ii) $\left(a^*+b\right)^*$ and $\left(a+b\right)^*$

The language defined by the expression $\left(a^*+b\right)^*$

Set of all strings of a's and b's

Language1 $\left(a^*+b\right)^*$.

$\left(a^*+b\right)=\{\wedge,a,b,aa,aaa,..\}$

$\left(a^*+b\right)^*=\{\wedge,a,b,aa,ab,ba,bb,aaa,..\}$

The language defined by the expression $\left(a+b\right)^*$

Set of all strings of a's and b's,

Language2 $\left(a+b\right)^*$.

$\left(a+b\right)^*=\{\wedge,a,b,aa,ab,ba,bb,aaa,..\}$

Hence Language1 and Language2 define the same set of language.

Comment

Chapter 4, Problem 15P
3 Bookmarks
Show all steps: ON

Comment

Step 3 of 3

iii) $\left(a^*+b^*\right)^*$ and $\left(a+b\right)^*$

The language defined by the expression $\left(a^*+b^*\right)^*$

Set of all strings of a's and b's

Language1 $\left(a^*+b^*\right)^*$.

$\left(a^*+b^*\right)=\{\wedge,a,b,aa,bb,aaa,bbb..\}$

$\left(a^*+b^*\right)^*=\{\wedge,a,b,aa,ab,ba,bb,aaa,aabbaa,..\}$

The language defined by the expression $\left(a+b\right)^*$

Set of all strings of a's and b's,

Language2 $\left(a+b\right)^*$.

$\left(a+b\right)^*=\{\wedge,a,b,aa,ab,ba,bb,aaa,aabbaa,..\}$

Hence Language1 and Language2 define the same set of language.

Comment

Chapter 4, Problem 16P

3 Bookmarks

Show all steps: ON

Problem

Show that the following pairs of regular expressions define the e language over the alphabet Σ = {*a b*}:

(i) Λ^* and **Λ**?

(ii) $(\mathbf{a}^*\mathbf{b})^*\mathbf{a}^*$ and $\mathbf{a}^*(\mathbf{ba}^*)^*$

(iii) $(\mathbf{a}^*\mathbf{bbb})^*\mathbf{a}^*$ and $\mathbf{a}^*(\mathbf{bbba}^*)^*$

Step-by-step solution

Step 1 of 3

I) $\wedge^*$ and $\wedge$

$\wedge^*$ consists of all strings $(\wedge)^n$ for $n = 0,1,2,.........$

1) $\wedge^0 = \wedge$

2) $\wedge^1 = \wedge$

Thus, $\wedge^*$ and $\wedge$ are equal

Chapter 4, Problem 16P

3 Bookmarks

Show all steps: ON

Step 2 of 3

(II) $\left(a^*b\right)^* a^*$ and $a^*\left(ba^*\right)^*$

$\left(a^*b\right)^* a^*$ consists of all strings $\left(a^*b\right)^n a^n$ for $n = 0,1,2,.........$

$a^*\left(ba^*\right)^*$ consists of all strings $a^n\left(ba^*\right)^n$ for $n = 0,1,2,......$

Using induction, we will prove $\left(a^*b\right)^n a^n = a^n\left(ba^*\right)^n$

Assume that $\left(a^*b\right)^{n-1} a^{n-1} = a^{n-1}\left(ba^*\right)^{n-1}$

Now consider

$(a^*b)^n a^n$ = (a b) $(a^*b)^{n-1}a^{n-1}$

= a b ($a^{n-1}(ba^*)^{n-1}$)

= $a^n(ba^*)^n$

Thus, both the regular expressions define the same language

Comment

Chapter 4, Problem 16P

3 Bookmarks Show all steps: ON

Step 3 of 3

(III) $\left(a^{*}bbb\right)^{*}a^{*}$ and $a^{*}\left(bbba^{*}\right)^{*}$

$\left(a^{*}bbb\right)^{*}a^{*}$ consists of all strings $\left(a^{*}bbb\right)^{n}a^{n}$ for $n=0,1,2.........$

$a^{*}\left(bbba^{*}\right)^{*}$ consists of all strings $a^{n}\left(bbba^{*}\right)^{n}$ for $n=0,1,2,........$

Using induction, we will prove $\left(a^{*}bbb\right)^{n}a^{n}$ = $a^{n}\left(bbba^{*}\right)^{n}$

Now consider

$\left(a^{*}bbb\right)^{n}a^{n}$ = a bbb ($a^{*}bbb)^{n-1}a^{n-1}$

= a bbb($a^{n-1}bbba^{*})^{n-1}$

= $a^{n}(bbba^{*})^{n}$ for n = 0, 1, 2,.......

Thus, both the regular expressions define the same language

Comment

Was this solution helpful? 0 1

Chapter 4, Problem 17P

4 Bookmarks Show all steps: ON

Problem

Show that the following pairs of regular expressions define the e language over the alphabet Σ = {*a b*}:

(i) ((**a** + **bb**)*aa)* and **Λ** + (**a** + **bb**)*aa

(ii) (**aa**)*(**Λ** + **a**) and **a***

(iii) **a**(**aa**)*(**Λ** + **a**)**b** + **b** and **a*b**

(iv) **a**(**ba** - **a**)*b and **aa*b**(**aa*b**)*

(v) **Λ** + **a**(**a** + **b**)* + (**a** + **b**)*aa(**a** + **b**)* and ((**b*****a**)*ab*)*

Step-by-step solution

Step 1 of 5

(i)

((a + bb) * aa) * and ∧ + (a + bb) * aa

((a + bb) * aa) *

This regular expression generates the strings the either start with ‘a’ or ‘bb’ and ends with ‘aa’ and also contains the empty string.

Step-by-step solution

Step 1 of 5

(i)

((a + bb) * aa) * and ∧ + (a + bb) * aa

((a + bb) * aa) *

This regular expression generates the strings the either start with 'a' or 'bb' and ends with 'aa' and also contains the empty string.

∧ + (a + bb) * aa

This also generates the strings the either start with 'a' or 'bb' and ends with 'aa' along with empty string.

Thus, both the regular expressions define the same language, and hence they are equal.

Comment

Step 2 of 5

(ii)

Chapter 4, Problem 17P

4 Bookmarks Show all steps: ON

string.

Thus, both the regular expressions define the same language, and hence they are equal.

Comment

Step 2 of 5

(ii)

(aa)* (∧ + a) and a*

(aa)* (∧ + a)

This regular expression generates the strings that contains all possible a's including empty string.

a*

This also generates the strings that contains all possible a's including empty string.

Thus, both the regular expressions define the same language, and hence they are equal.

Comment

Step 3 of 5

(iii)

Chapter 4, Problem 17P

4 Bookmarks

Show all steps: ON

Thus, both the regular expressions define the same language, and hence they are equal.

Comment

Step 3 of 5

(iii)

a(aa)* (∧ + a) b + b and a*b

a(aa)* (∧ + a) b + b

This regular expression generates the strings that contains only exactly one ‘b’ or starting with all possible combinations of ‘a’ that end with single ‘b’.

a*b

This also generates the strings that contains only exactly one ‘b’ or starting with all possible combinations of ‘a’ that end with single ‘b’.

Thus, both the regular expressions define the same language, and hence they are equal.

Comment

Step 4 of 5

Chapter 4, Problem 17P

4 Bookmarks

Show all steps: ON

Comment

Step 4 of 5

(iv)

a (ba + a) *b and aa*b (aa* b) *

a (ba + a) *b

This regular expression generates the strings that contains all combinations of ‘ba’ or ‘a’ that start with ‘a’ and end with ‘b’.

aa*b (aa* b) *

This also generates the strings that start with ‘a’ and end with ‘b’ and all possible combinations of a’s or ba’s.

Thus, both the regular expressions define the same language, and hence they are equal.

Comment

Step 5 of 5

(v)

Thus, both the regular expressions define the same language, and hence they are equal.

Comment

Step 5 of 5

(v)

$\wedge$ + a (a + b) * + (a + b) * aa (a + b) * and ((b*a) ab*)*

$\wedge$ + a (a + b) * + (a + b) * aa (a + b) *

This regular expression generates all possible words that contains a's or b's.

((b*a) ab*) *

This also generates all possible words that contains a's or b's.

Thus, both the regular expressions define the same language, and hence they are equal.

Comment

Was this solution helpful? 8 0

Problem

Describe (in English phrases) the languages associated with the following regular expressions:

(i) (**a** + **b**)****a**(**Λ** + **bbbb**)

(ii) (**a**(**a** + **bb**)*)*

(iii) (**a**(**aa**)****b**(**bb**)*)*

(iv) (**b**(**bb**)*)*(**a**(**aa**)****b**(**bb**)*)*

(v) (**b**(**bb**)*)*(**a**(**aa**)****b**(**bb**)*)*(**a**(**aa**)*)*

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

Step-by-step solution

Step 1 of 6

(i) Any string that ends in *a* or *abbbb*.

Comment

Chapter 4, Problem 18P

5 Bookmarks

Show all steps: ON

Step 1 of 6

(i) Any string that ends in *a* or *abbbb*.

Comment

Step 2 of 6

(ii) All words that do not begin with *b* and in which *b*'s appear in clumps of even length

Comment

Step 3 of 6

(iii) All words that contains odd number of *a*'s followed by any odd number of *b*'s or a null string. The series begins with *a*'s and ends with *b*'s.

Comment

Chapter 4, Problem 18P

5 Bookmarks

Show all steps: ON

Step 4 of 6

(iv) The strings of all letters of *a*'s and *b*'s where both letters appear in clumps of odd length and strings always end with *b*. This language also includes the null string.

Comment

Step 5 of 6

(v) The strings of all letters of *a*'s and *b*'s where both letters appear in clumps. This language also includes the null string.

Comment

Step 6 of 6

(vi) All words with even lengths and which start in *a*'s or *b*'s and end in *a*'s. This language also includes null string.

Problem

(i) Explain why we can take any pair of equivalent regular expressions and replace the letter a in both with any regular expression **R** and the letter *b* with any regular ex-pression S and the resulting regular expressions will have the same language. For example, 16(ii), which says

$(\mathbf{a}^*\mathbf{b})^*\mathbf{a}^* = \mathbf{a}^*(\mathbf{ba}^*)^*$

becomes the identity

$(\mathbf{R}^*\mathbf{S})^*\mathbf{R}^* = \mathbf{R}^*(\mathbf{SR}^*)^*$

which is true for all regular expressions **R** and **S**. In particular, **R** = **a** + **bb**, **S** = **ba*** results in the complicated identity

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

(ii) What identity would result from using

$R = (ba^*)^*$ $S = (\mathbf{\Lambda} + \mathbf{b})$

Example, 16(ii)

$(\mathbf{a}^*\mathbf{b})^*\mathbf{a}^*$ and $\mathbf{a}^*(\mathbf{ba}^*)^*$

Step-by-step solution

Step 1 of 2

Step 1 of 2

I) Because the regular expressions are closed under the following properties, there will not be any change in replacing letter by regular expression

1) The distributive property

2) The associative property

3) The commutative property

4) The identity property

5) The order of operations including exponents and nested parentheses

6) The combination of like terms

Comment

Step 2 of 2

II)

$(((ba^*)^*)^*(\wedge+b))^*((ba^*)^*)^* = ((ba^*)^*)^*((\wedge+b)((ba^*)^*)^*)^*$

$((ba^*)(\wedge+b))^*(ba^*) = ((ba^*)((\wedge+b)(ba^*))$