studocu

# Chapter 5 Solutions

Automata Theory (University of the Punjab)

Chapter 5, Problem 1P

4 Bookmarks

Show all steps: ON

Problem

Write out the transition_tables for the FAs on pp. 56, 58 (both), 63, 64, and 69 that were defined by pictures.

Step-by-step solution

**Step 1** of 9

P: 56

Transition table:

| | a | b |
|---|---|---|
| Start x- | y | z+ |
| y | -x | z+ |
| Final z+ | z+ | z+ |

Comment

**Step 2** of 9

Chapter 5, Problem 1P

4 Bookmarks

Show all steps: ON

Comment

**Step 2** of 9

P: 58

Transition table:

| | a | b |
|---|---|---|
| Start - | + | + |
| Final + | + | + |

Comment

**Step 3** of 9

Transition table:

| | a | b |
|---|---|---|
| Start final ± | ± | ± |

Comment

**Step 4** of 9

P: 63

Transition table:

| | a | b |
|---|---|---|
| Start 1 - | 2 | 3 |
| 2 | 4 + | 3 |
| 3 | 2 | 4 + |
| Final 4 + | 4 + | 4 + |

Comment

**Step 5** of 9

Transition table:

| | a | b |
|---|---|---|
| Start 1 - | 2 | 2 |
| 2 | 3 | 3 |
| 3 | 4 | 5 + |
| 4 | 4 | 4 |
| Final 5 + | 5 + | 5 + |

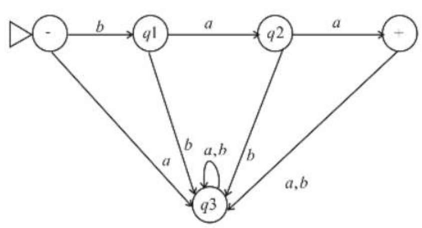
Comment

Chapter 5, Problem 1P

4 Bookmarks

Show all steps: ON

Transition graph:



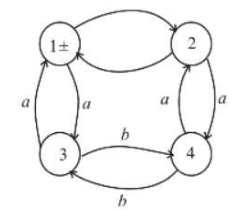

Comment

**Step 7** of 9

Transition table:

| | a | b |
|---|---|---|
| Start - | $q3$ | $q1$ |
| $q1$ | $q2$ | $q3$ |
| $q3$ | $q3$ | $q3$ |
| $q2$ | + | $q3$ |
| Final + | $q3$ | $q3$ |

P: 69:




Comment

Step 9 of 9

Transition table:

| | a | b |
|---|---|---|
| Start final 1± | 3 | 2 |
| 2 | 4 | 1± |
| 3 | 1± | 4 |
| 4 | 2 | 3 |

Comment



Build an FA that accepts only the language of all words with *b* as the second letter. Show both the picture and the transition table for this machine and find a regular expression for the language.

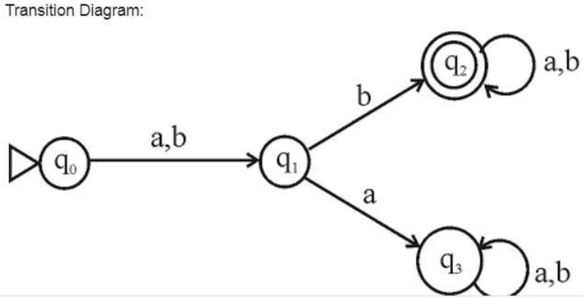
Step-by-step solution

Step 1 of 2

Regular expression for the words with 'b' as the second letter:

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

Transition Diagram:



**Step 2** of 2

Transition table:

| | a | b |
|---|---|---|
| Start $q0$ | $q1$ | $q1$ |
| $q1$ | $q3$ | $q2$ |
| Final $q2$ | $q2$ | $q2$ |
| $q3$ | $q3$ | $q3$ |

Comment

Was this solution helpful? 11 0



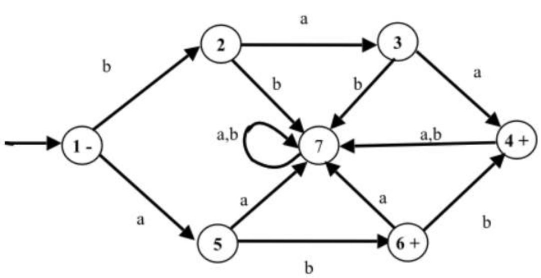
Build an FA that accepts only the words *baa*, *ab*, and *abb* and no other strings longer or shorter.

Step-by-step solution

**Step 1** of 3

**Finite automata:**

$M = \{Q, \Sigma, q_0, \delta, F\}$




Where

Chapter 5, Problem 3P | 3 Bookmarks | Show all steps: ON

**Transition table for above FA:** $\delta = Q \times \Sigma \rightarrow Q$

| State/Input | a | b |
|---|---|---|
| →1 | 5 | 2 |
| 2 | 3 | 7 |
| 3 | 4 | 7 |
| F 4 | 7 | 7 |
| 5 | 7 | 6 |
| F 6 | 7 | 4 |
| 7 | 7 | 7 |

Here state 7 is a trap state or dead state.

Comment

**Step 3** of 3

The input 'baa' will go through the states: 1 – 2 – 3 – 4 and terminate.

The input 'ab' will go through the states: 1 – 5 – 6 and terminate.

The input 'abb' will go through the states: 1 – 5 – 6 – 4 and terminate.

Therefore, only these three strings are accepted by the above FA.

Chapter 5, Problem 4P | 5 Bookmarks | Show all steps: ON

(i) Build an FA with three states that accepts all strings.

(ii) Show that given any FA with three states and three +'s, it accepts all input strings.

(iii) If an FA has three states and only one +, must it reject some inputs?

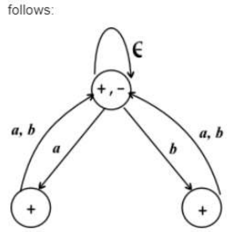
Step-by-step solution

**Step 1** of 4

Consider the set of alphabets = $\{a,b\}$

**(i)**

The Finite automaton that accepts the accepts all strings over the set $\{a,b\}$ using 3 states is as follows:



**Step 2** of 4

The following automaton accepts all strings of all lengths that is, strings of length one, two, three, four and so on are accepted.

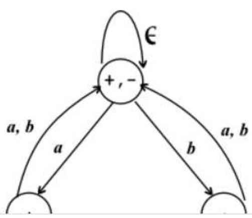
For example: ‘a’, ‘b’, ‘ba’, ‘bb’, ‘aa’, ‘ab’, ‘aaa’, ‘aba’, ‘bab’, ‘bbb’ are accepted.

Comment

**Step 3** of 4

**(ii)**

The Finite automaton that accepts the accepts all strings over the set $\{a,b\}$ using 3 states as **+** states is as follows:




Chapter 5, Problem 4P

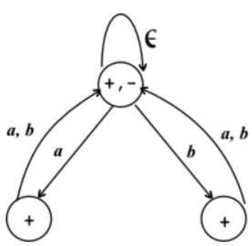
5 Bookmarks

Show all steps: ON

**Step 3** of 4

**(ii)**

The Finite automaton that accepts the accepts all strings over the set $\{a,b\}$ using 3 states as **+** states is as follows:




The following Finite Automata accepts strings of all lengths which can be listed as follows:

- **Length one:** The strings ‘a’, ‘b’ are accepted.
- **Length two**: The strings ‘ba’, ‘bb’, ‘aa’, ‘ab’ are accepted.
- **Length three**: The strings ‘aaa’, ‘aba’, ‘bab’, ‘bbb’ are accepted.

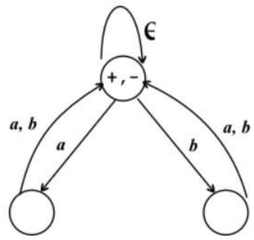
Similarly, the strings of length four, five and so on are accepted by this finite automaton.

Comment

**Step 4** of 4

**(iii)**

The Finite automaton with only + state among the 3 states is as follows:




Yes. When the automata consists of only one **+** state, then some strings are rejected.

**For example:**

• Consider the string ***aab***.

• The string *aab* is not acceptable using the above finite automata, because the string *aab* cannot reach its final state.

• The strings ‘a’, ‘b’ are not accepted because they are pointing to dead-end states and these states are unreachable.

• Thus, some strings are rejected when the automata consist of only one **+** state. In simple words, the strings with odd length are rejected using the above automata.



(i) Build an FA that accepts only those words that have more than four letters.

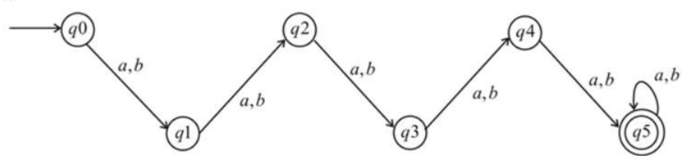
(ii) Build an FA that accepts only those words that have fewer than four letters.

(iii) Build an FA that accepts only those words with *exactly* four letters.

Step-by-step solution

**Step 1** of 3

I)

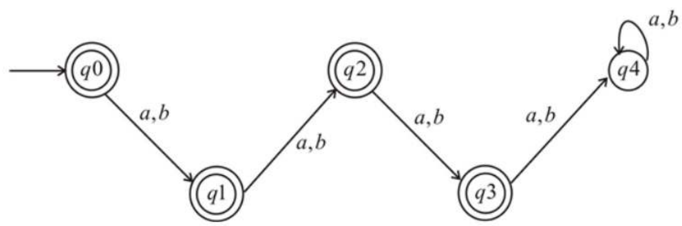



The above FA accepts four letters up to q4 state. After q4, q5 accepts another word while after q5, the kleene closure of {a, b} to q5 allows it to accept any set of words including null string. Hence, the above FA has more than four words.

Comment

**Step 2** of 3

II)




The above FA has four accepting or terminating states. If the control terminates at q0, no word will be present in the language. If termination takes place at q1, the language will have one word. Similarly, at q2, language has two words and at q3, the language has three words. So far, all possible words will have lengths less than four. Even if the control does not terminate at q3, the maximum word length up to q4 is four.

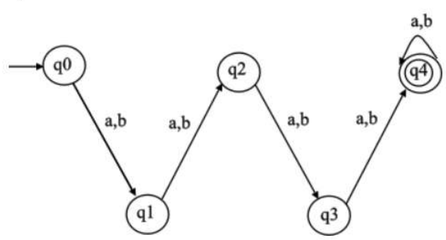
Comment

**Step 3** of 3



**Step 3** of 3

III)




The above FA accepts exactly four letters as, on reaching q4, the inputs to the FA are going to be terminated.

Comments (3)

Was this solution helpful? 2 2

Chapter 5, Problem 6P

5 Bookmarks

Show all steps: ON

Problem

Build an FA that accepts only those words that do *not* end with *ba*.

Step-by-step solution

**Step 1** of 2

To construct a finite automata *FA* that accepts only those words that do not end with "ba", consider the *FA* with the following things:

1. a finite set of states $Q = \{1,2,3\}$
2. an input letters $\Sigma = \{a,b\}$
3. a transition function $\delta : Q \times \Sigma \rightarrow Q$
4. start state $q_0 = \{1\}$
5. final states $F = \{1,2\}$

Comment

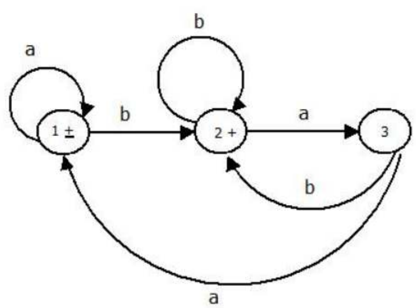
Chapter 5, Problem 6P

5 Bookmarks

Show all steps: ON

**Step 2** of 2

The finite automata $FA = (Q, \Sigma, \delta, q_0, F)$ to accept only those words that do not end with "ba" is given as follows:




**Explanation:**

• The sign '-' indicates the start state and '+' indicates the final state. Here, " $\pm$ " indicated the start as well as final state.

• The language accepts by a finite automaton is the set of words that should be start with start state "-" and end with a final state "+".

• To change from one state to another state, use only the required input transition letters a or b.

Chapter 5, Problem 7P

5 Bookmarks

Show all steps: ON

## Problem

Build an FA that accepts only those words that begin or end with a double letter.

## Step-by-step solution

**Step 1** of 2

Consider the language *L* over *{a, b}* accepts the set of strings that begin or end with a double letter. Here, the double letter means *aa* or *bb*. The language accepts the strings like *aa, abaa, bba, . . .*

Comment

**Step 2** of 2

The FA that accepts the language *L* is as follows:

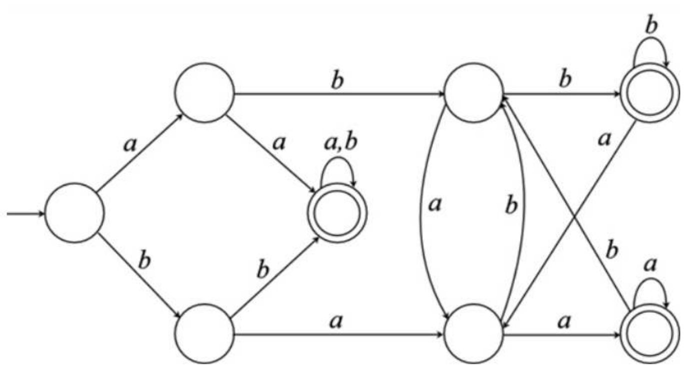
Chapter 5, Problem 7P

5 Bookmarks

Show all steps: ON

**Step 2** of 2

The FA that accepts the language *L* is as follows:




The above is a DFA that accepts the set of strings over *{a, b}* that begin or end with a double letter.

Comment

Chapter 5, Problem 8P

6 Bookmarks

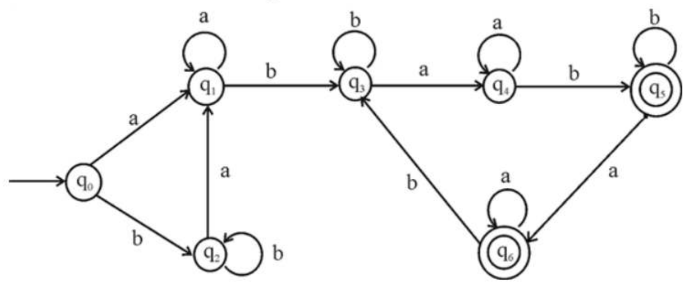
Show all steps: ON

Build an FA that accepts only those words that have an even number of substrings *ab*.

Step-by-step solution

**Step 1** of 1

Many possibilities are there, we are giving one example. The FA that accepts only those words that have an even number of substrings 'ab'




Comments (1)

Chapter 5, Problem 9P

2 Bookmarks

Show all steps: ON

Problem

(i) Recall from Chapter 4 the language of all words over the alphabet {*a b*} that have both the letter *a* and the letter *b* in them, but not necessarily in that order. Build an FA that accepts this language.

(ii) Build an FA that accepts the language of all words with only *a*'s or only *b*'s in them. Give a regular expression for this language.

Step-by-step solution

**Step 1** of 2

I) The proper FA should be: All the words that have both the letter a and b. Where + denotes the final state and – denotes the start state.

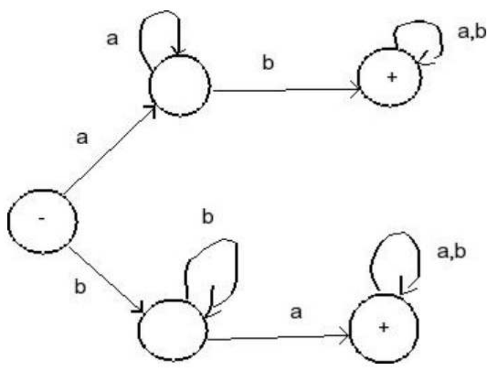
Chapter 5, Problem 9P

2 Bookmarks

Show all steps: ON

**Step 1** of 2

I) The proper FA should be: All the words that have both the letter a and b. Where + denotes the final state and – denotes the start state.




Comment

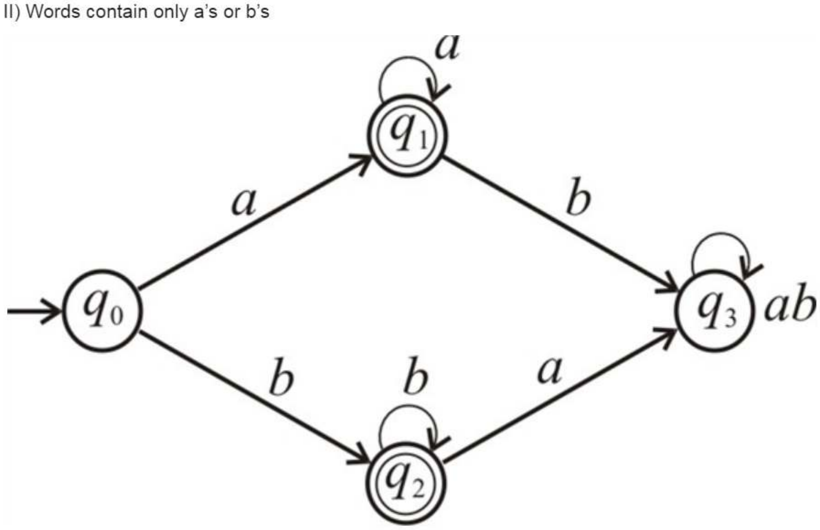
Chapter 5, Problem 9P

2 Bookmarks

Show all steps: ON

**Step 2** of 2

II) Words contain only a's or b's




Regular expression: (aa* + bb*)

Chapter 5, Problem 11P

1 Bookmark

Show all steps: ON

Problem

Show that there are exactly 5832 different finite automata with three states *x*,*y*, *z* over the alphabet {*a* *b*}, where *x* is always the start state.

Step-by-step solution

**Step 1** of 1

$2^3 \times 3^6 = 5832$ different automata on three states acting on two letters. Obvious symmetries such as permutation of states, permutation of letters, and inversion of states, together with minimization, reduces the number of automata that needs to checked to 194.

Comments (2)

Could someone explain with more depth?

Chapter 5, Problem 11P

1 Bookmark

Show all steps: ON

**Step 1** of 1

$2^3 \times 3^6 = 5832$ different automata on three states acting on two letters. Obvious symmetries such as permutation of states, permutation of letters, and inversion of states, together with minimization, reduces the number of automata that needs to checked to 194.

Comments (2)

Could someone explain with more depth?

2 represents the two letters given to us in the Sigma alphabet. The 3 making 2^3 represents the number of states. This product is then multiplied to a 3 that again represents the number of states and a 6 (for the 3^6) that represents the number of outputs each state (number out outputs is determine by how many letters in the alphabet).

Submit

Chapter 5, Problem 14P

5 Bookmarks

Show all steps: ON

## Problem

(i) Build an FA that accepts the language of all strings of *a*'s and *b*'s such that the next-to-last letter is an *a*.

(ii) Build an FA that accepts the language of all strings of length 4 or more such that the next-to-last letter is equal to the second letter of the input string.

## Step-by-step solution

**Step 1** of 2

(i)

The regular expression for the language of all strings of *a*'s and *b*'s such that contains *a* as the before last letter is $(a+b)*a(a+b)$.

**The Finite Automata is as follows:**

Chapter 5, Problem 14P

5 Bookmarks

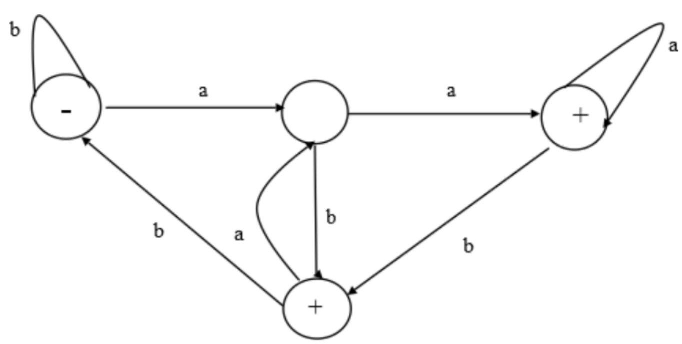
Show all steps: ON

**Step 1** of 2

(i)

The regular expression for the language of all strings of *a*'s and *b*'s such that contains *a* as the before last letter is $(a+b)*a(a+b)$.

**The Finite Automata is as follows:**




Comment

Chapter 5, Problem 15P

1 Bookmark

Show all steps: ON

## Problem

Build a machine that accepts all strings that have an even length that is not divisible by 6.

## Step-by-step solution

**Step 1** of 2

Consider the language *L* that accepts all strings that have an even length that is not divisible by 6 over the alphabet $\{a,b\}$. The strings that can accepted are $ab,\ aaba,\ aabbabab\ldots$

Comment

**Step 2** of 2


Chapter 5, Problem 15P

1 Bookmark

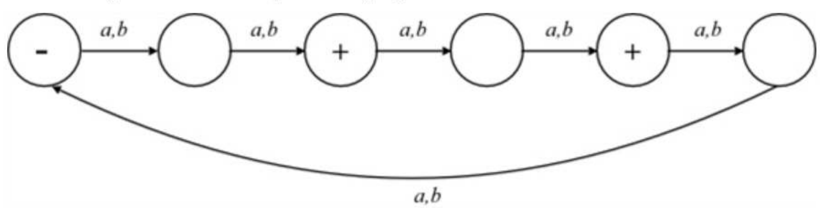
Show all steps: ON

Consider the language *L* that accepts all strings that have an even length that is not divisible by 6 over the alphabet $\{a,b\}$. The strings that can accepted are $ab,\ aaba,\ aabbabab\ldots$

Comment


**Step 2** of 2

The following is an FA that accepts the language *L*.




Comment

Chapter 5, Problem 16P

3 Bookmarks

Show all steps: ON

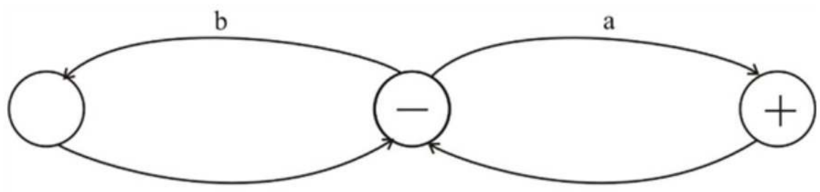
Problem

Build an FA such that when the labels a and b are swapped the new machine is different from the old one but equivalent (the language defined by these machinesITthe same).

Step-by-step solution

**Step 1** of 3

Old FA:




Chapter 5, Problem 16P

3 Bookmarks

Show all steps: ON

Comment

**Step 2** of 3

The language generated by this FA is, L = {a, aba, aaa, bba, ababa, …….}

This DFA has three states such that the alphabet ‘a’ is accepted from the initial state to final state.

There is one state which is known as the dead-end state for which the transition of ‘b’ takes place. If the labels ‘a’ and ‘b’ are swapped, then the new DFA will be as follows:

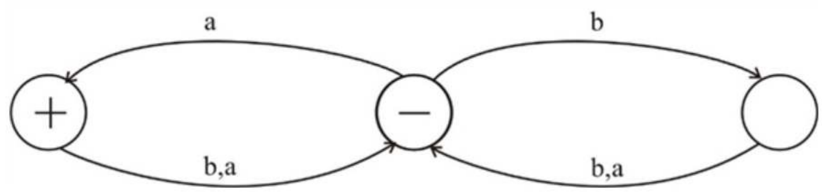
Comment

**Step 3** of 3

New FA (after swapping the labels and also the states):

**Step 3** of 3

New FA (after swapping the labels and also the states):




The machine has been changed by changing the positions of states and labels. But, the output of the new FA is same as the output of the old FA.

Here, the labels are changed but not the states and the strings accepted by this DFA is the same as the original one.

The language generated by this new FA is, L = {a, aba, aaa, bba,…}

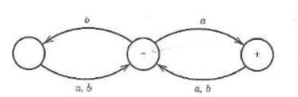
Therefore, the two FA's generate the same language

Comment

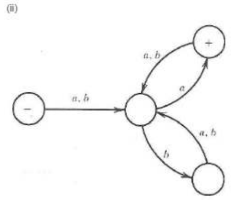


Describe in English the languages accepted by the following FAs:

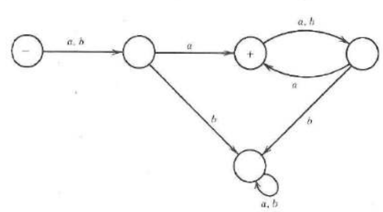
(i)




(ii)




(iii)




(iv) Write regular expressions for the languages accepted by these three machines.

Chapter 5, Problem 17P

6 Bookmarks

Show all steps: ON

**Step 1** of 4

i) Second state is initial, the FA accepts the language L of all odd length words over {a, b} that end with a.

Comment

**Step 2** of 4

ii) First state is initial, the FA that accepts words of at least length two, even length strings that end with ‘a’.

Comment

**Step 3** of 4

iii) First state is initial, the FA will accept words of at least length two, even length of strings that ends with ‘a’ and will not accept consecutive b’s.

Chapter 5, Problem 17P

6 Bookmarks

Show all steps: ON

**Step 3** of 4

iii) First state is initial, the FA will accept words of at least length two, even length of strings that ends with ‘a’ and will not accept consecutive b’s.

Comment

**Step 4** of 4

iv) i. (aa + ab + ba + bb)*a

ii. (a+b) [(b(a+b))*(a(a+b))*]*a.

iii. (a+b)a((a+b)a)*

Comment

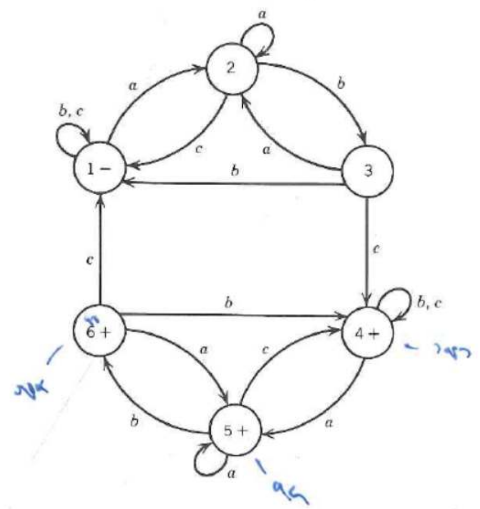
Was this solution helpful? 21 4

Chapter 5, Problem 18P

2 Bookmarks

Show all steps: ON

The following is an FA over the alphabet Σ = {*a b c*}. Prove that it accepts all strings that have an odd number of occurrences of the substring *abc*.

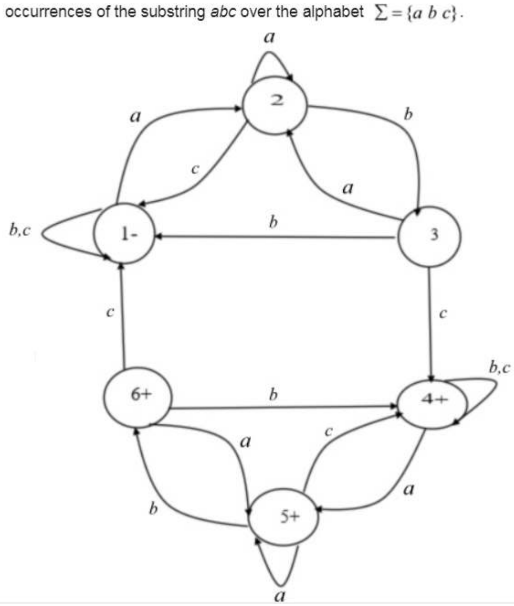


Chapter 5, Problem 18P

2 Bookmarks

Show all steps: ON

The following Finite automaton (FA) that accepts all strings that have an odd number of occurrences of the substring *abc* over the alphabet $\Sigma = \{a\ b\ c\}$.



Chapter 5, Problem 18P

2 Bookmarks Show all steps: ON

**Step 2** of 2

In the FA, '1' is the start state (initial sate). The states '4', '5' and '6' are final states. The FA accepts the strings like *abc*, *aabc*, *abcbabcaabc*, …

Consider the string *baabca*. This string contains only one occurrence of the substring *abc*.

• The state '1' is the start state.

• Read the first character from the string. The character '*b*' is the first one. For the input '*b*' from the state '1', it will not be changed because for the inputs *b* and *c* there is a self-loop to the state '1'.

• Read the next character from the string. The character '*a*' is the second one. For the input '*a*' from the state '1', it is changed to '2'.

• Read the next character from the string. The next character is '*a*'. The state will not be changed for the input '*a*' from the state '2'.

• Read the next character from the string. The next character is '*b*'. The state is changed to '3' for the input '*b*' from the state '2'.

• Read the next character from the string. The next character is '*c*'. The state is changed to '4' for the input '*c*' from the state '3'.

• Read the next character from the string. The next character is '*a*'. The state is changed to '5' for the input '*a*' from the state '4'.

• All the characters are read from the string. The current state is '5' which is the final state. The list of states that are visited is $1 \xrightarrow{b} 1 \xrightarrow{a} 2 \xrightarrow{a} 2 \xrightarrow{b} 3 \xrightarrow{c} 4 \xrightarrow{a} 5$. Thus, the string *baabca* is accepted by the FA.

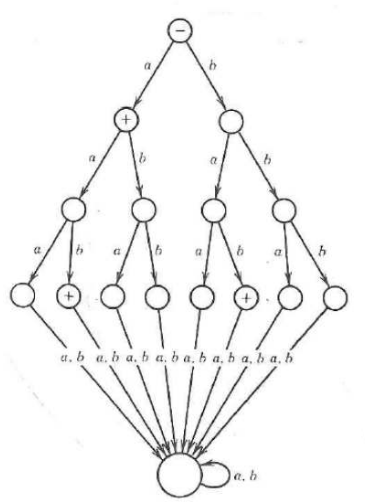
**Therefore, the FA accepts all strings that have an odd number of occurrences of the substring *abc* .**

Chapter 5, Problem 19P

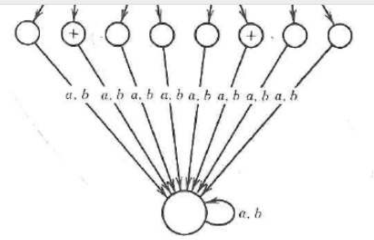
1 Bookmark Show all steps: ON

Problem

Consider the following FA:




(i) Show that any input string with more than three letters is not accepted by this FA.




(i) Show that any input string with more than three letters is not accepted by this FA.

(ii) Show that the only words accepted are *a, aab, and bab*.

(iii) Show that by changing the location of + signs alone, we can make this FA accept the language {*bb aba bba*}.

(iv) Show that any language in which the words have fewer than four letters can be accepted by a machine that looks like this one with the + signs in different places.

(v) Prove that if *L* is a finite language, then there is some FA that accepts *L* extending the binary-tree part of this machine several more layers if necessary.

Step-by-step solution

**Step 1** of 6

Chapter 5, Problem 19P
1 Bookmark
Show all steps: ON

**Step 1** of 6

I) Taking input of 'abab' will take us to state q15 which is a dummy state and the string is not accepted by the FA

Comment

**Step 2** of 6

II) Starting from q0 and reading a will we go to q1 which is a final stat so the word 'a' is accepted by this language similarly 'aab' and 'bab' words area also accepted by this language, passing any other word will take us to a state which will be either non-final (or) dummy state, and the word will not be accepted.

Comment

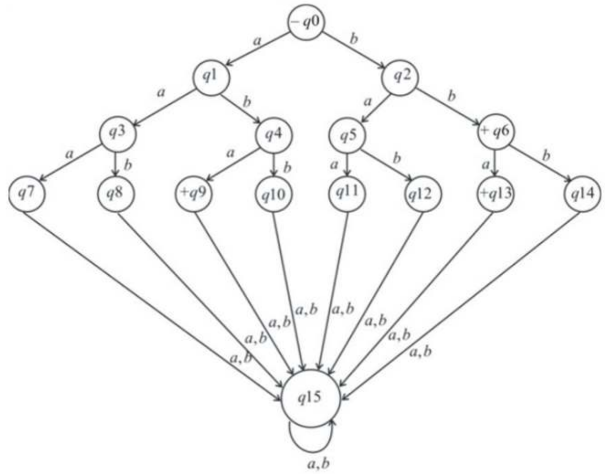
**Step 3** of 6

III) As this FA is change now it accepts the only words bb, bba, aba,

No other word than these will be accepted by this FA

Comment

**Step 4** of 6

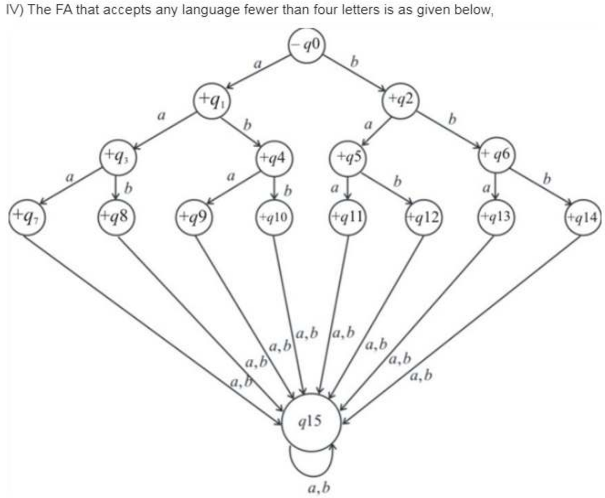



Comment



**Step 5** of 6

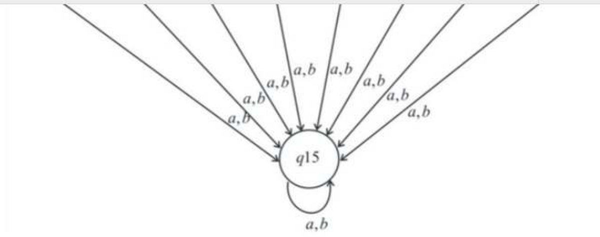
IV) The FA that accepts any language fewer than four letters is as given below,




Comment

Chapter 5, Problem 19P

1 Bookmark Show all steps: ON




Comment

**Step 6** of 6

V) As the language accepts finite no. of words in a finite language. So there can be another FA by extending more layers and mentioning length of words accepted (or) just giving words along with.

Comment

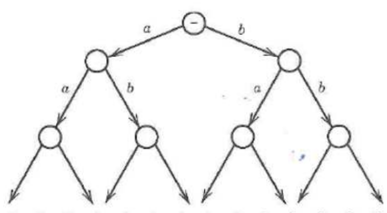
Was this solution helpful? 12 3

Chapter 5, Problem 20P

Bookmark Show all steps: ON

Problem

Let us consider the possibility of an infinite automaton that starts with this infinite binary tree:




Let $L$ be any infinite language of strings of $a$'s and $b$'s whatsoever. Show that by the judicious placement of +'s, we can turn the picture above into an infinite automaton to accept the language $L$. Show that for any given finite string, we can determine from this machine, in a finite time, whether it is a word in $L$. Discuss why this machine would not be a satisfactory language-definer for $L$.

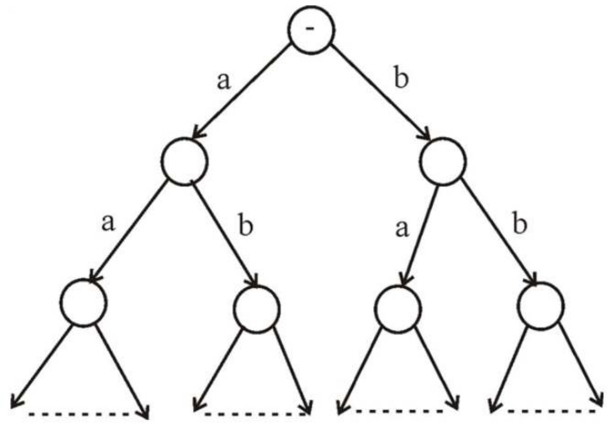
Next

Step-by-step solution

**Step 1** of 4

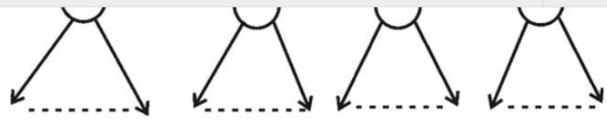
Step-by-step solution

**Step 1** of 4




Comment

Chapter 5, Problem 20P
Bookmark
Show all steps: ON



Comment

**Step 2** of 4

Above tree energy state contains 2 – edges.

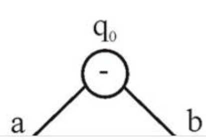
Suppose $q_0, q_1, q_2$................$q_n$ all states contains the edges a,b

L be infinite language of strings a's and b's

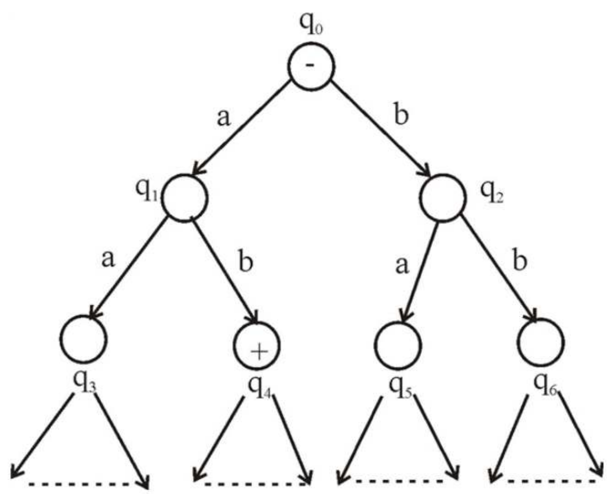
Let us assume to place '+' final state in $q_4$ state in the infinite tree

Comment

**Step 3** of 4

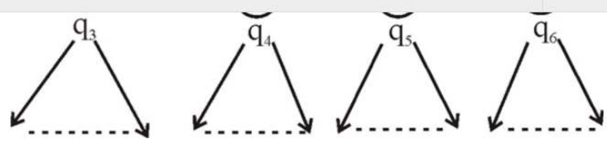


**Step 3** of 4




Comment

Chapter 5, Problem 20P
Bookmark
Show all steps: ON




Comment

**Step 4** of 4

Now, the infinite tree accepts only the string 'ab'.

Suppose, we place final state in (+) $q_3$ position the tree accepts only the string 'aa'.

In this infinite tree the edges are going one state other states like.

$q_0 \to q_1 \to q_3$ like this way.

So, we can't place final states in all states in the infinite tree.

That the reason, this machine would not be a satisfactory language definer for 'L'

Comment

Was this solution helpful? 0 0