Problem

Give informal English descriptions of PDAs for the languages in Exercise 2.6.

Step-by-step solution

Step 1 of 4

a.

Consider the following language:

"The set of strings over the alphabet $\{a,b\}$ with more a's than b's"

This is the language where the number of a's is greater then the number of b's. Construct the PDA (Push Down Automata), that accepts the above language. It uses its stack to count the number of a's minus the number of b's. It enters an accepting state whenever this count is positive or top of the stack is a.

In more detail, it operates as follows:

• The PDA scans across the input.

• If it sees a a and its top of the stack symbol is either a or empty than, push a into the stack.

• If it scans a a and its top stack symbol is a b, it pops the stack.

• If it sees a b and its top of the stack symbol is either b or empty than, push b into the stack.

• If it sees a b and its top stack symbol is a, it pops the stack.

After the PDA finishes the input, if a is on top of the stack, it accepts. Otherwise, it rejects.

Therefore, given language is recognized by the above PDA.

Comment

Step 2 of 4

b.

Consider the following language:

"The compliment of the language $\{a^n b^n : n \geq 0\}$"

First, we will construct the compliment of the given language. That is number of a's is not equal to the number of b's. The compliment of this language can obtain by breaking the given language into the union of several simpler languages.

The PDA uses its stack to count the number of a's and the number of b's. It enters an accepting state whenever this count is not equal.

In more detail, it operates as follows:

• The PDA scans across the input.

• If it sees a a and its top of the stack symbol is either a or empty than, push a into the stack.

• If it sees a b and its top stack symbol is a, it pops the stack.

• If it sees a b and its top of the stack symbol is either b or empty than, push b into the stack.

After the PDA finishes the input, if a or b is on top of the stack, it accepts. Otherwise, it rejects.

Therefore, if the end of the input is reached and stack is not empty than, it accepts. Otherwise it rejects because number of a's and the number of b's in the input are equal.

Comments (2)

Step 3 of 4

c.

Consider the following language:

$\left\{w\#x : w^R \text{ is a substring of } x \text{ for } w,x \in \{0,1\}^*\right\}$

The language is obtained by constructing the strings w^R which is a substring x. The PDA scans across the input string and finds the input is a substring of x.

In more detail, it operates as follows:

• The PDA scans across the input string and pushes every symbol it reads until it reads until it reads a $\#$.

• If $\#$ is never encountered, it rejects.

• Then the PDA skips over part of the input, nondeterministically deciding when to stop skipping.

• At that point, it compares the next input finishes while the stack is nonempty, this branch of the computations rejects.

After the PDA finishes the input, if the stack becomes empty, the machine reads the rest of the input and accepts.

Therefore, given language is recognized by the above PDA.

Comment

Step 4 of 4

d.

Consider the following language:

$$\left\{x_1 \# x_2 \# \cdots \# x_k \mid k \geq 1, \text{ each } x_i \in \{a,b\}^*, \text{ and for some } i \text{ and } j, x_i = x_j^R\right\}$$

The language can obtain by constructing matching pair of strings with at least one $\#$ between them. Before, after and between the matching pairs there can be any number of strings of a's and b's separated by $\#$. The PDA scans across the input string and finds the input is with at least one $\#$ is between them.

In more detail, it operates as follows:

• The PDA scans across the input string and pushes every symbol it reads until it reads until it reads a $\#$.

• If # is never encountered, it rejects.

• If an input $\#$ is read, the nondeterministically skips to another # in the input and continue reading input symbols, now comparing them with symbols popped off the stack. If all agree and stack becomes empty at the same time as either end of the input of # symbol is reached, accept, otherwise reject.

After the PDA finishes the input, if the stack becomes empty, the machine reads the rest of the input and accepts.

Therefore, given language is recognized by the above PDA.

Comment