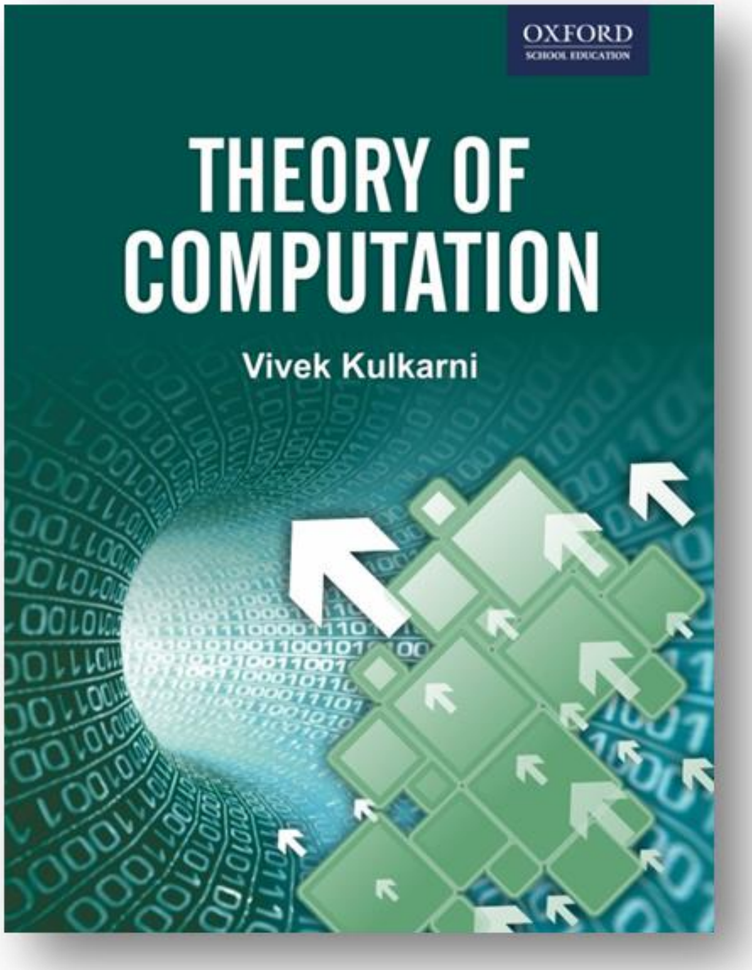


Author: Vivek Kulkarni
(vivek_kulkarni@yahoo.com)

Chapter-8: Post Machines

Solutions for Review Questions

Q.1 Write a short note on PMs.

Solution:

Refer to the sections 8.1 and 8.2.

Q.2 Compare TM, PDM, FSM, and PM (definition is not required).

Solution:

Refer to the sections 8.3 and 8.5.

Q.3 Write a short note on the power of PMs.

Solution:

Refer to the sections 8.1, 8.3 and 8.5.

Q.4 Show that a Post machine has more power than a PDA.

Solution:

Refer to the section 8.3.

Q.5 What are the different elements of a Post machine?

Solution:

Refer to the sections 8.2 and 8.4.

Q.6 Design a Post machine to check if the given parentheses are well-formed. Simulate it with the help of a suitable example.

Solution:

Refer to the example 8.4 from the book.

Q.7 Design a Post machine to accept palindrome strings over the alphabet $\sum = \{x, y\}$. Simulate its working with the help of a suitable example.

Solution:

Refer to the example 8.5 from the book; replace *a* by *x* and *b* by *y*.

Q.8 Design Post machines that accept the following languages:

(i) $L = \{a^n b^n c^n \mid n \geq 0\}$

(ii) $L = \{a^n b^n \mid n \geq 0\}$

Solution:

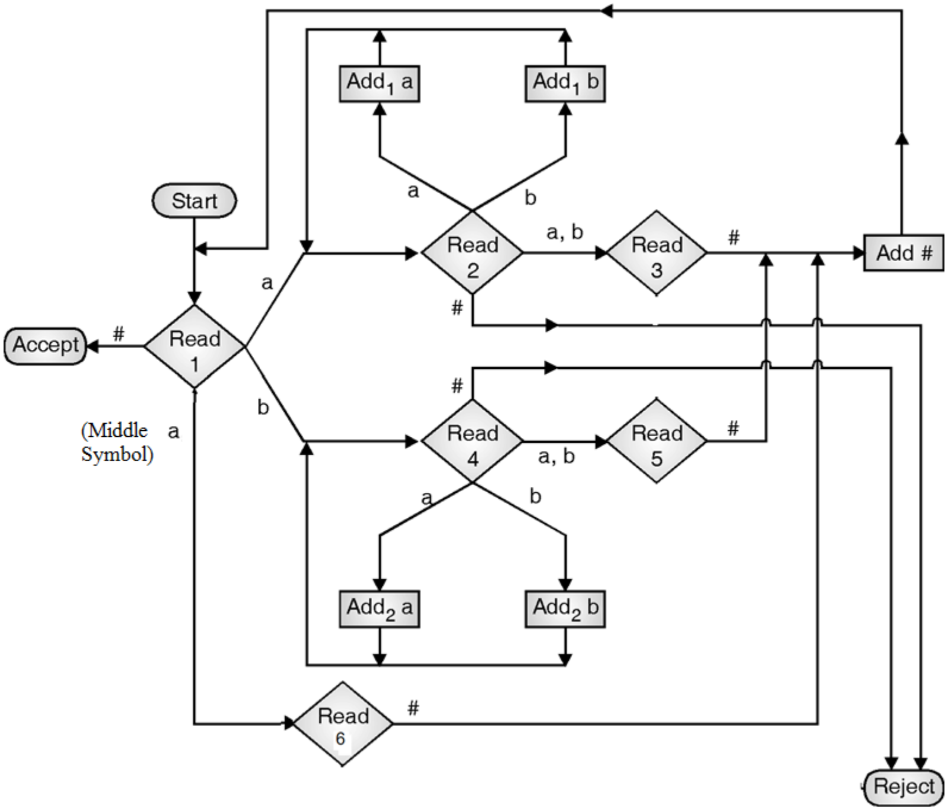
(i) Refer to the example 8.6 from the book.

(ii) Refer to the example 8.3 from the book.

Q.9 Design a Post machine which accepts the strings of *a*'s and *b*'s having odd length, and *a* as the middle element.

Solution:

We can draw the NPM as shown below.



Q.10 Design a Post machine that accepts the language described by the following grammar:

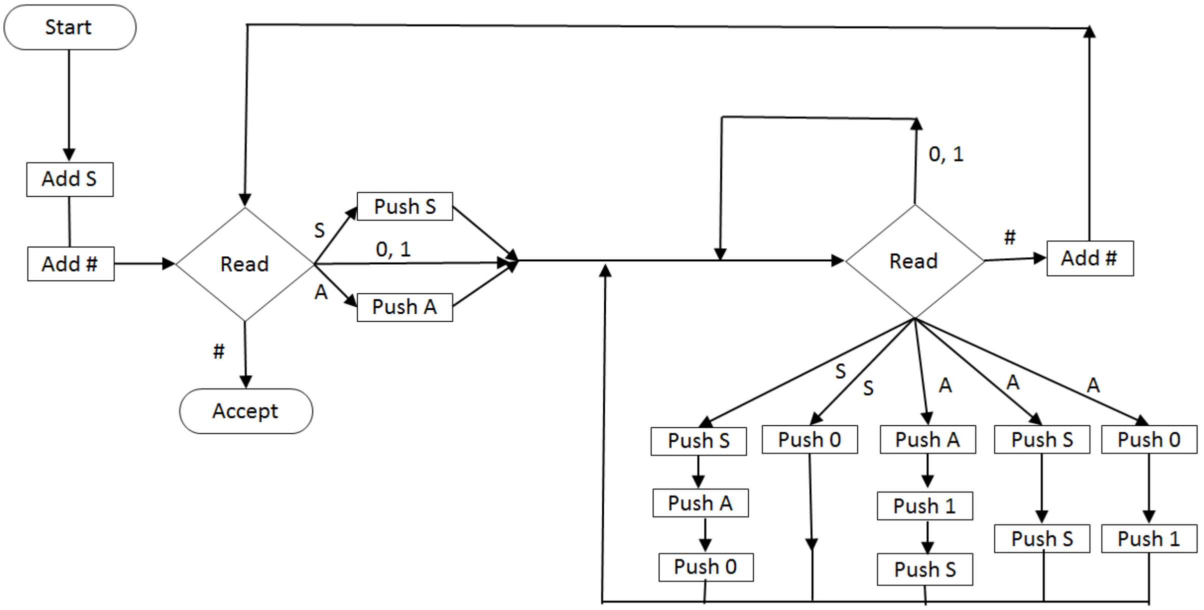
$G = (\{S, A\}, \{0, 1\}, P, S)$, where P consists of:

$S \to 0\,A\,S \mid 0$

$A \to S\,1\,A \mid S\,S \mid 1\,0$

Solution:

We can design PM as described in the figure below. Assume that the function ‘Push’ is implemented using a queue data structure. It involves adding the symbol to be pushed to the end of the queue and then the remaining of the queue to be added as it is till it reads ‘#’. Thus, ‘Push’ always adds the symbol to the beginning of the queue; in turn, treats the queue as stack.



Q.11 Design a Post machine that accepts the language described by the following grammar:

$S \to A\,B\,A$

$A \to a\,A \mid \in$

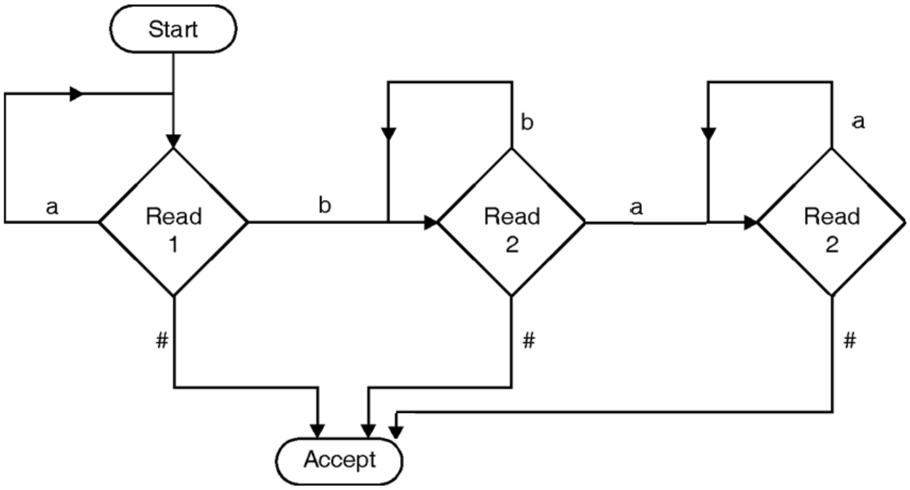
$B \to b\,B \mid \in$

Comment on the type of language.

Solution:

As we can see symbols A and B accepts the regular languages denoted as, a^* and b^* respectively. Hence, the language generated by the grammar given is also regular and can be denoted by the regular expression, ($a^* b^* a^*$).

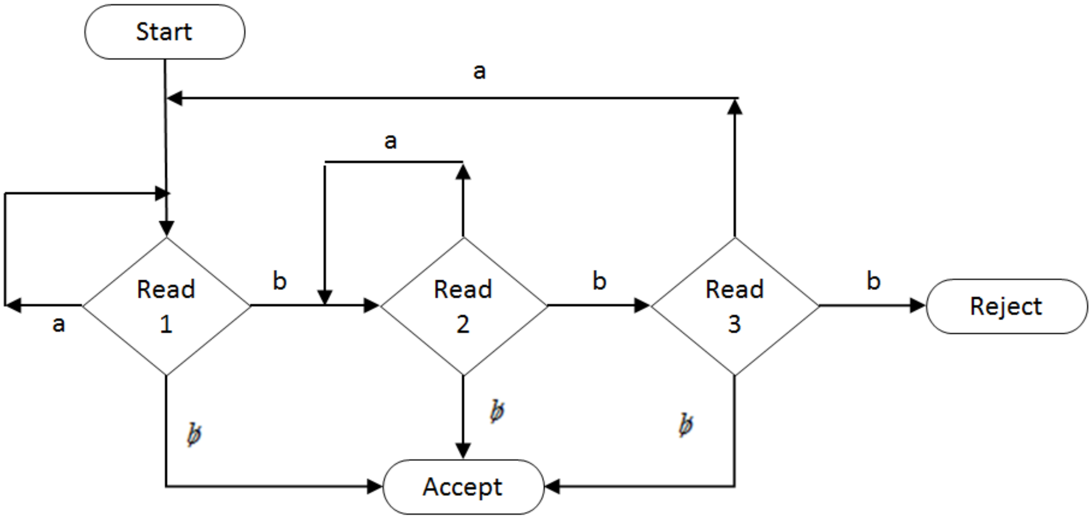
The PM can be drawn as below,



Q.12 Design a Post machine the accepts the language *L*, over ∑ = {*a*, *b*}, such that *L* contains words in which the letter *b* does not appear consecutively three times.

Solution:

This is a regular language and can be denoted by the regular expression, (a* b a* b a^+)*. The post machine accepting this language can be drawn as,


Start
a
a
Read 1
a
b
Read 2
b
Read 3
b
Reject
b̸
b̸
b̸
Accept