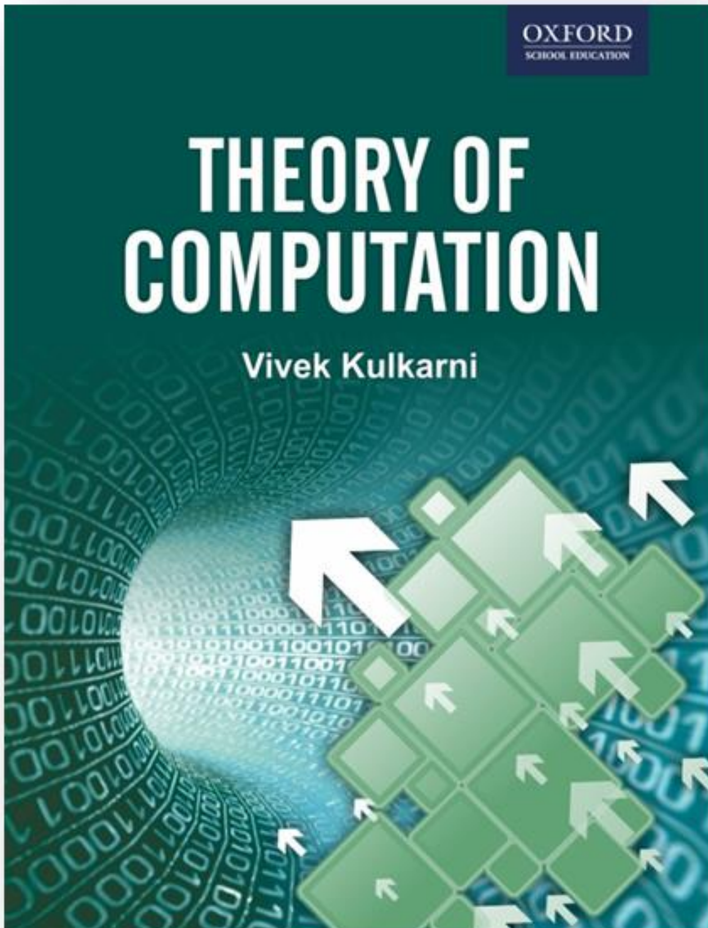


Author: Vivek Kulkarni
(vivek_kulkarni@yahoo.com)

Chapter-6: Pushdown Stack-memory Machine

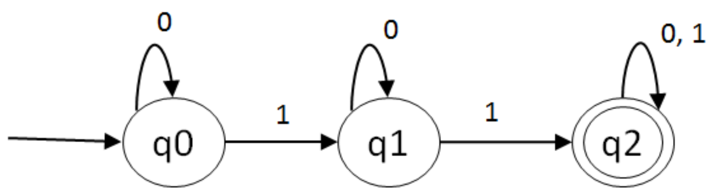
Solutions for Review Questions

Q.1 Convert the given FA, $M = \{(q_0, q_1, q_2), (0, 1), \delta, (q_0), (q_2)\}$ into its equivalent PDA; the transition function δ for M is defined as:

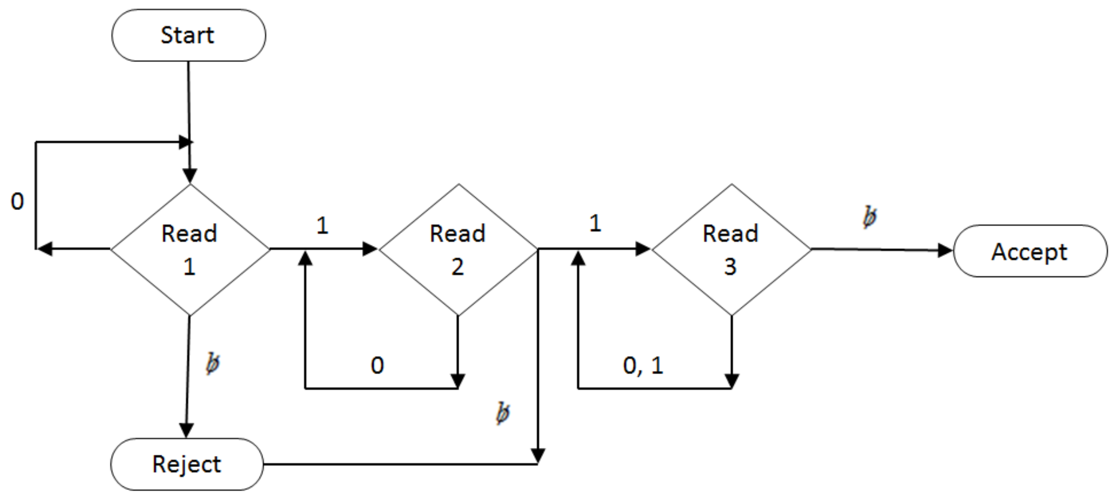
	0	1
q_0	q_0	q_1
q_1	q_1	q_2
q_2	q_2	q_2

Solution:

The transition diagram for the given DFA is,



We can draw the same DFA using PDA notations as below,



Q.2 Discuss the relative powers of DPDA and NPDA.

Solution:

Refer to the section 6.6.

Q.3 Construct a PDA that accepts the language defined by the following regular grammar:

$S \rightarrow 0\,A \mid 1\,B \mid 0$

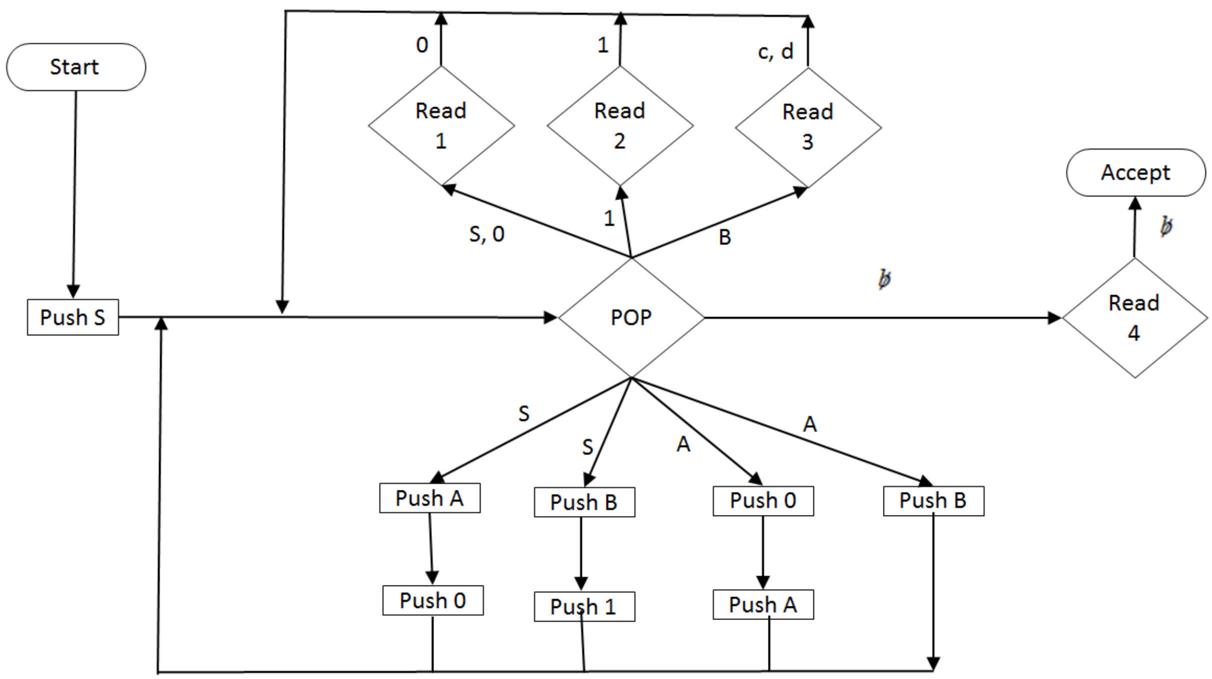
$A \rightarrow A\,0 \mid B$

$B \rightarrow c \mid d$

Here, $N = \{S, A, B\}$, T = {0, 1, c, d}, and S is the start symbol.

Solution:

PDA accepting the language generated by the above grammar is,



Q.4 With the help of PDAs, show that context-free languages are closed under union, concatenation, and Kleene closure.

Solution:

Refer to the section 6.8.

Q.5 Give the formal definition of PDA.

Solution:

Refer to the section 6.3.

Q.6 Construct a PDA (or NPDA) that accepts the language over $\sum = \{a, b\}$, and is defined as: $L = \{a^n b^n \mid n = 0, 1, 2 \ldots\}$.

Simulate the working of this PDA (or NPDA) for the inputs:

(i) *aaabbb* (ii) *aab* (iii) *aaa*

Solution:

Refer to the example 6.3 from the book.

Q.7 Which machine accepts the language of palindromes, FSM or PDM? Justify your answer.

Solution:

Refer to the example 6.6 from the book.

Q.8 Construct a PDA equivalent to following CFG:

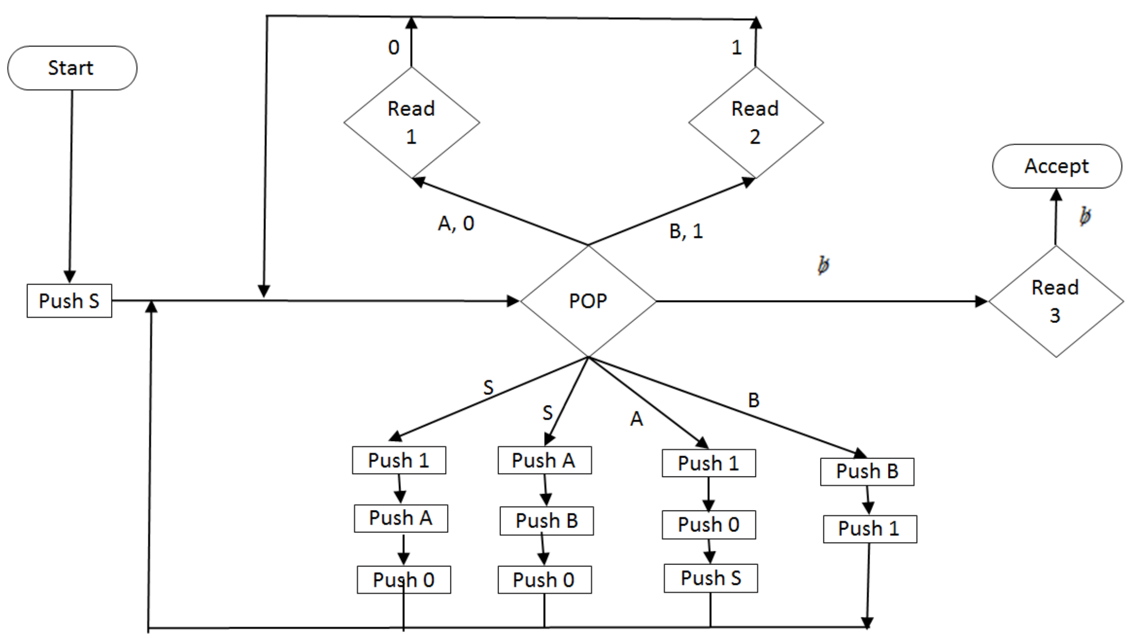
$S \rightarrow 0\,A\,1 \mid 0\,B\,A$

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

$B \rightarrow 1\,B \mid 1$

Solution:

The PDA can be drawn as,



Q.9 Give the graphical representation of the language generated by the following CFG:

$S \to S + S \mid S * S \mid 4$

Show the stack and tape contents for the expression: '4 + 4 * 4'.

Solution:

Refer to the example 6.8 from the book.

Q.10 Construct a PDA that accepts the following language by an empty stack:

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

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

Solution:

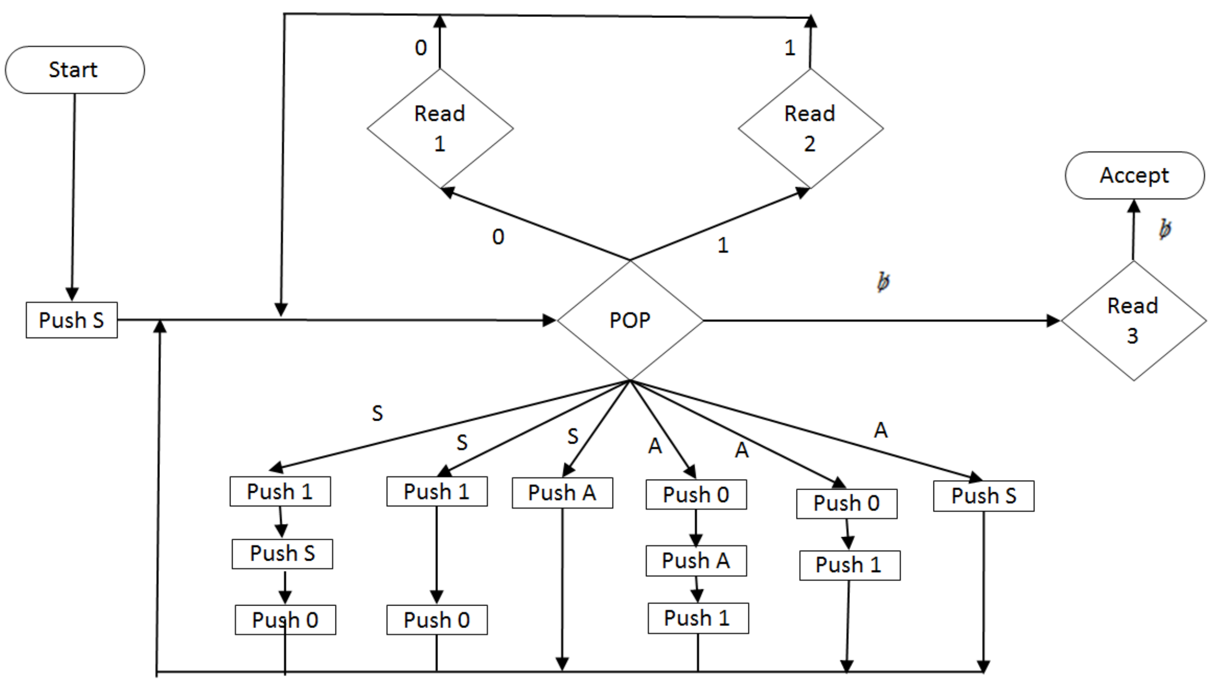
Let us first remove the $\in$ transitions. The non-terminals S and A both are nullable. The modified grammar is,

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

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

This grammar does not accept the empty string, $\in$.

The PDA now can be drawn as,



Q.11 Construct a PDA equivalent to the following grammar:

$S \rightarrow a\,A\,A$

$A \rightarrow a\,S \mid b\,S \mid a$

Solution:

Let us convert the given grammar to CNF so as to reduce the stack growth, as discussed in the section 6.7.1. The grammar in CNF can be written as,

$S \rightarrow P\,R$

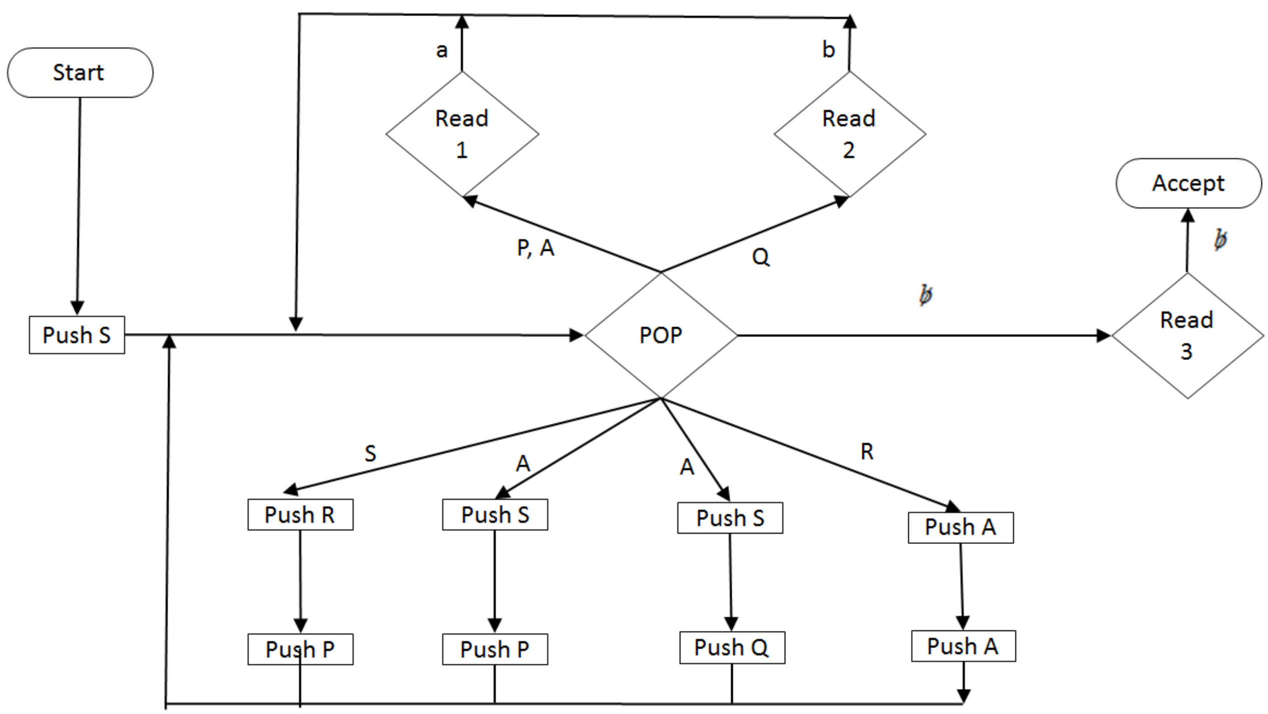
$A \rightarrow P\,S \mid Q\,S \mid a$

$R \rightarrow A\,A$

$P \rightarrow \text{a}$

$Q \rightarrow \text{b}$

Let us use this grammar in CNF to draw the PDA.



Q.12 Define pushdown automata. What are the different types of PDA? What are the applications of PDA?

Solution:

For definition of PDA refer to the section 6.3. Refer to the section 6.6 to know more about DPDA and NPDA. Section 6.7 puts light on using PDA as a parser.

Q.13 Let the grammar G be defined as:

$S \rightarrow a\,A\,B\,B \mid a\,A\,A$

$A \rightarrow a\,B\,B \mid a$

$B \rightarrow b\,B\,B \mid A$

Construct an NPDA that accepts the language generated by this grammar.

Solution:

Let us convert the grammar to CNF first. The modified grammar is,

$S \rightarrow P\,U \mid P\,T$

$A \rightarrow P\,R \mid a$

$B \rightarrow Q\,R \mid P\,R \mid a$

$R \rightarrow B\,B$

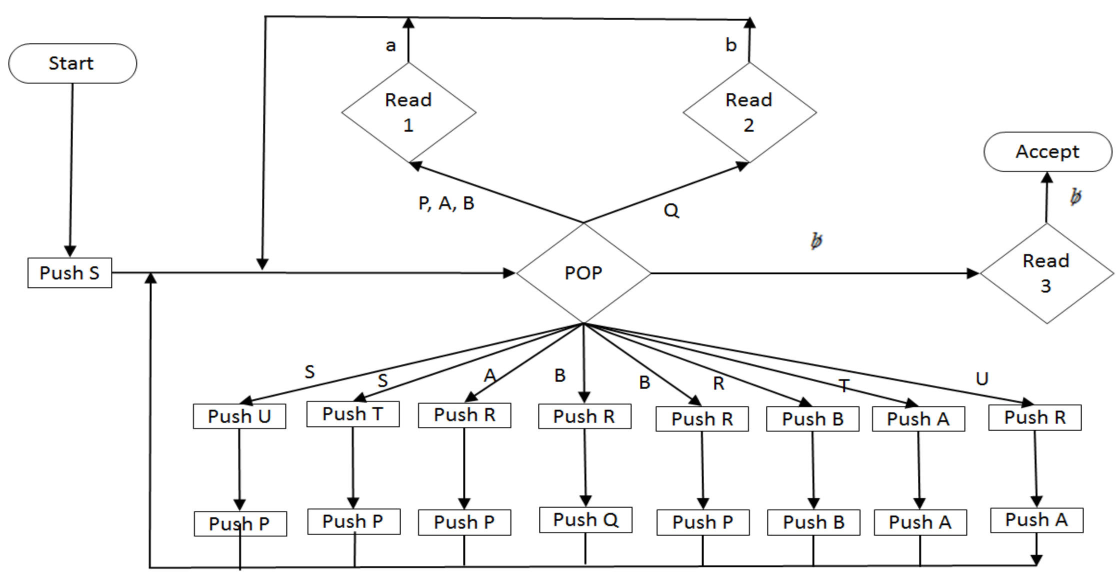
$T \rightarrow A\,A$

$U \rightarrow A\,R$

$P \rightarrow \text{a}$

$Q \rightarrow \text{b}$

The PDA now can be drawn as below,



Q.14 Construct pushdown automata for each of the following languages:

(1) The set of palindromes over alphabet $\{a, b\}$

(2) The set of all strings over alphabet $\{a, b\}$ with exactly twice many a's as b's

(3) $L = \{a^i b^j c^k \mid i \neq j \text{ or } j \neq k\}$

Solution:

(1) Refer to the example 6.10 from the book.

(2) As per example 5.25 from the book, the grammar can be written as,

$S \to A\,S \mid \in$

$A \to a\,a\,b \mid a\,b\,a \mid b\,a\,a$

Let us remove the $\in$-transitions and then convert the grammar to CNF as below,

$S \to A\,S \mid R\,Q \mid P\,T \mid Q\,R$

$A \to R\,Q \mid P\,T \mid Q\,R$

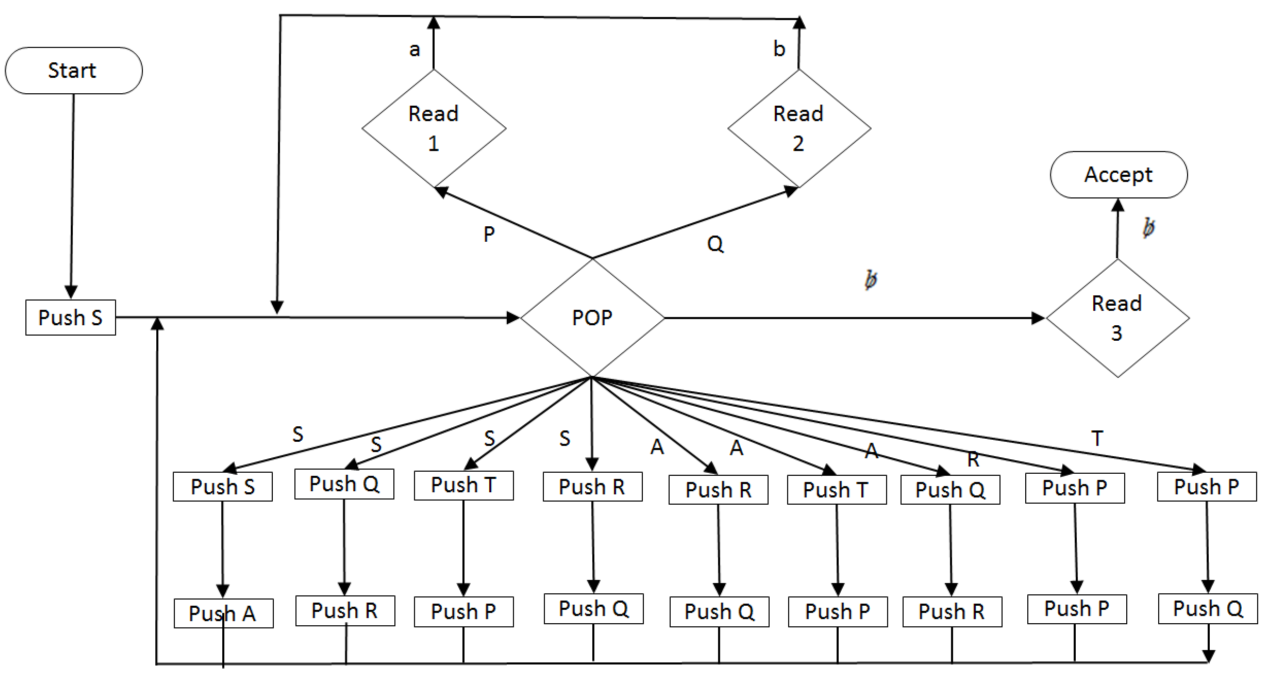
$R \to P\,P$

$T \to Q\,P$

$P \to$ a

$Q \to$ b

The PDA now can be drawn as,



(3) $i \neq j$ means, either $i > j$ or, $i < j$. Same holds true for $j \neq k$.

For the condition $i \neq j$ we can write the CFG as,

$X \rightarrow S\ C$

$S \rightarrow a\ S\ b \mid A \mid B$

$A \rightarrow a\ A \mid \in$

$B \rightarrow b\ B \mid \in$

$C \rightarrow c\ C \mid \in$

Similarly for the condition $j \neq k$ we can add the following productions,

$X \rightarrow A\ T$

$T \rightarrow b\ T\ c \mid B \mid C$

Therefore, the resultant grammar is,

$X \rightarrow S\ C \mid A\ T$

$S \rightarrow a\ S\ b \mid A \mid B$

$T \rightarrow b\ T\ c \mid B \mid C$

$A \rightarrow a\ A \mid \in$

$B \rightarrow b\ B \mid \in$

$C \rightarrow c\ C \mid \in$

Let us try removing the $\in$-transitions to simplify the grammar. We can see that, *A*, *B*, *C*, and hence, even *S*, *T* and *X* are nullable. The simplified grammar can be written as,

$X \rightarrow S\ C \mid A\ T \mid C \mid A \mid T$

$S \rightarrow a\ S\ b \mid a\ b \mid A \mid B$

$T \rightarrow b\ T\ c \mid b\ c \mid B \mid C$

$A \rightarrow a\ A \mid a$

$B \rightarrow b\ B \mid b$

$C \rightarrow c\ C \mid c$

The PDA now can be drawn as per the algorithm discussed in the section 6.7.

Q.15 Consider the following two languages:

$L_1 = \{a^n\ b^{2n}\ c^m \mid n, m \geq 0\}$

$L_2 = \{a^n\ b^m\ c^{2m} \mid n, m \geq 0\}$

Is $L_1 \cap L_2$ a context-free language? Justify your answer.

Solution:

Let us try to list the strings from the language L_1,

L_1 = {∈, c, abb, cc, ccc, aabbbb, cccc, ccccc, cccccc, … }

Let us now try to list the strings from the language L_2,

L_2 = {∈, a, bcc, aa, aaa, bbcccc, aaaa, aaaaa, aaaaaa, … }

$L_1 \cap L_2 = \{\in\}$ is a context-free language. Any empty string is a regular language and class of regular languages is a subset of CFLs.

Q.16 Construct a pushdown automaton to accept the language:

$L = \{WW^R \mid W \in \{a, b\}^*$, and W^R is the reverse of $W\}$

Show all possible states, transition inputs, and the contents of the stack.

Solution:

As we can understand the language L represents nothing but all even length palindromes over {a, b}. Refer to the example 6.6 from the book.

Q.17 WW^R is accepted by an NPDA but not by any DPDA. Justify.

Solution:

Refer to the example 6.6 from the book.

Q.18 Discuss the relative powers of FSM, PDM and Turing machine (TM).

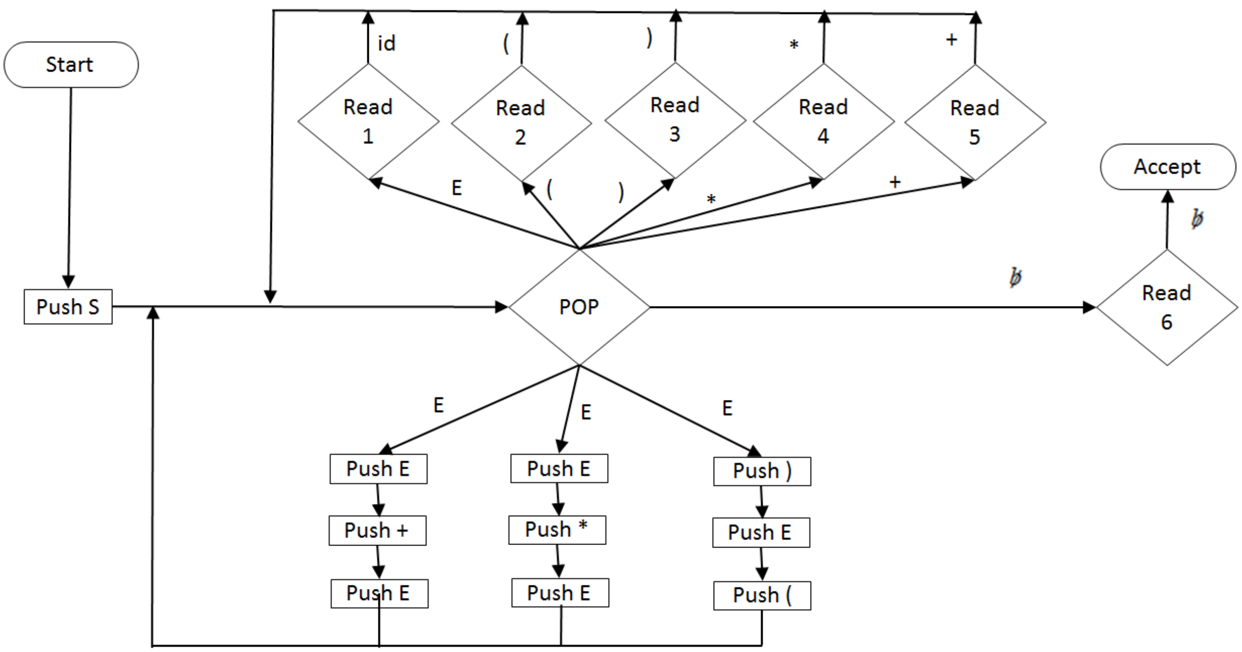
Solution:

Refer to the section 6.6.1.

Q.19 What is PDA? Construct a PDA for the grammar $E \rightarrow E + E \mid E * E \mid (E) \mid id$

Solution:

The PDA for the given grammar can be drawn as,



Q.20 Construct a DPDA that will recognize the language:

$L = \{WCW^R \mid W \in (a, b)^*$ and W^R is the reverse of $W\}$.

Simulate the working of this PDA for input string '*ababCbaba*'.

Solution:

The language L represents the odd length palindromes with the middle letter as *C*. Refer to the example 6.5; replace *X* by *C*.

Q.21 Design a PDA to accept the following languages for $n > 0$:

(1) $(bdb)^n C^n$

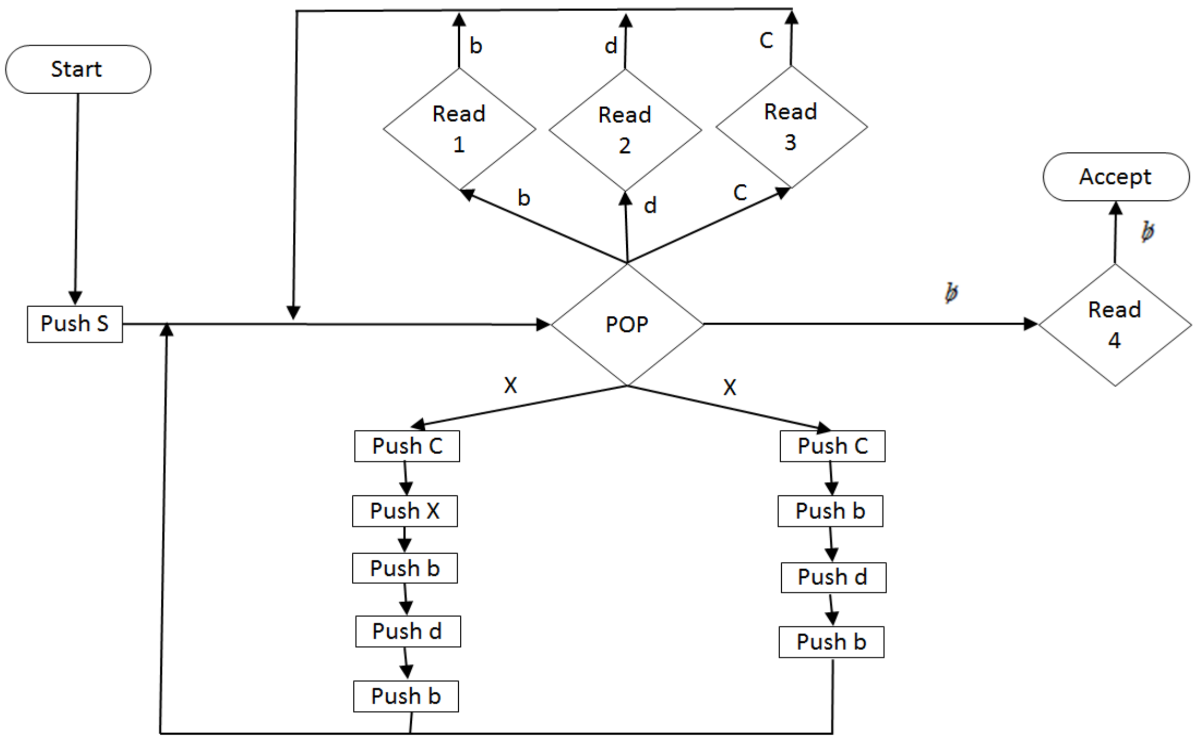
(2) $(ab)^n (cd)^n$

Solution:

(1) The language $(bdb)^n C^n$ is similar to language we saw in the example 6.3 from the book. We can write the grammar for the language as,

$X \rightarrow bdb\ X\ C \mid bdbC$

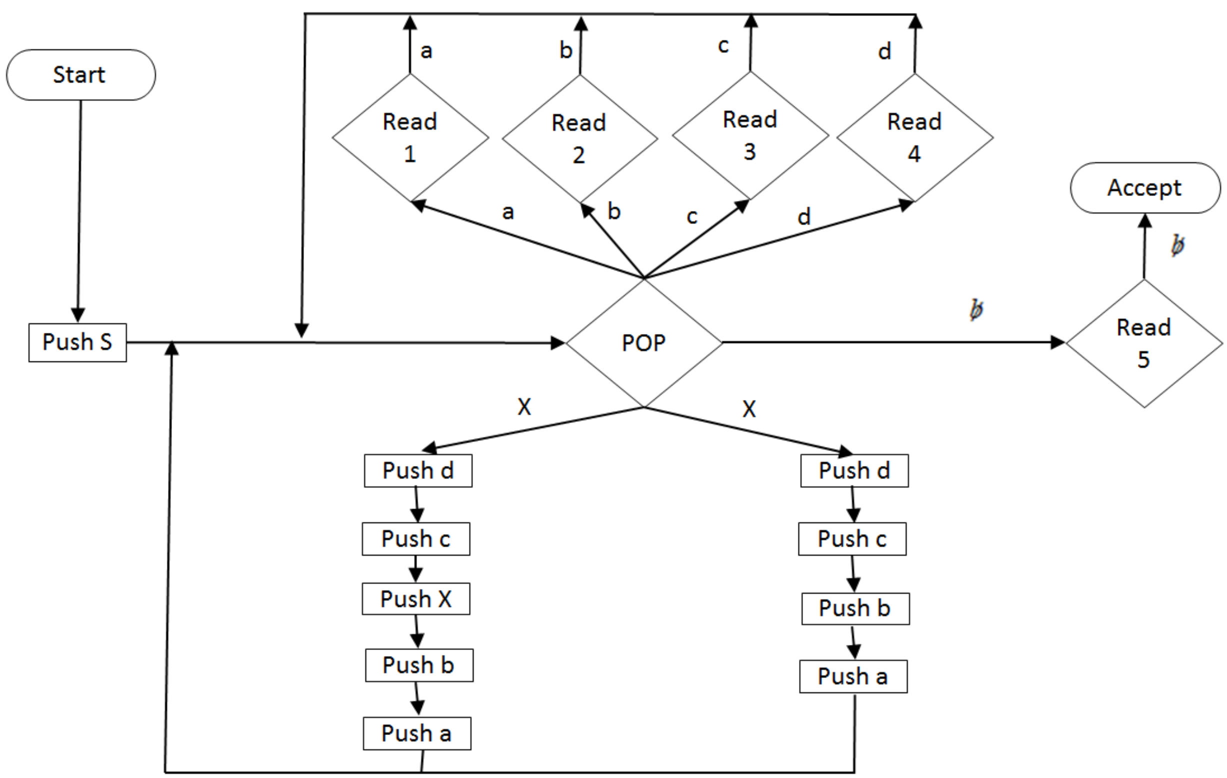
The PDA accepting the language is given as below.



(2) The language $(ab)^n\,(cd)^n$ can be represented using the following grammar,

$$X \rightarrow ab\;X\;cd \mid abcd$$

The PDA accepting the language is given as below.



Q.22 Design a pushdown automaton to accept the language containing all odd length palindromes over $\Sigma = \{0, 1\}$.

Solution:

Refer to the examples 6.5 and 6.6 from the book.

Q.23 Construct a context-free grammar equivalent to the following PDA (described with the help of the given set of equations):

$\delta\ (q_0, b, Z_0) = \{(q_0, ZZ_0)\}$

$\delta\ (q_0, \in, Z_0) = \{(q_0, \in)\}$

$\delta\ (q_0, b, Z) = \{(q_0, ZZ)\}$

$\delta\ (q_0, a, Z) = \{(q_1, Z)\}$

$\delta\ (q_1, b, Z) = \{(q_0, \in)\}$

$\delta\ (q_1, a, Z_0) = \{(q_0, Z_0)\}$

Solution:

We can make the below observations –

1. In state q0 every symbol *b* makes us push *Z* onto the stack without changing the state.
2. In state q0 if we get symbol *a*, the move is made to state q1.
3. In state q1 if we read *b*, one *Z* is popped from the stack and the transition is made to state q0.
4. In state q1 if we read *a* and if the stack is empty (we read the same *b*'s in q1 as we read in q0; refer 1 and 3 above), it makes transition to state q0; otherwise it could be error. This means when we read *a* in state q1, stack needs to be empty.
5. In state q0 if we receive end of input, the stack needs to be empty; the input is accepted.

From the above observations it is clear that whatever number of *b*'s are read in q0, one needs to find the same number of '*ab*'s so as to reach to the stack empty condition – *a* takes you to state q1 followed by *b* that pops one *Z* out of the stack.

The CFG thus can be written as,

$$S \rightarrow b\ S\ a\ b \mid a \mid \in$$

The production rule '$S \rightarrow b\ S\ a\ b$' represents that every b read in q0 has corresponding 'ab' in q1. The rule '$S \rightarrow a$' indicates that stack needs to be empty when only a is read in state q1. Similarly, '$S \rightarrow \epsilon$' indicates the acceptance of the empty input in q0.

Q.24 Give a grammar for the language $L(M)$, where:

$M = (\{q_0, q_1\}, \{0, 1\}, \{z_0, x\}, \delta, q_0, z_0, \phi)$,

and δ is given by:

$\delta\ (q_0, 1, z_0) = (q_0, xz_0)$ $\delta\ (q_0, \epsilon, z_0) = (q_0, \epsilon)$

$\delta\ (q_0, 1, x) = (q_0, xx)$ $\delta\ (q_1, 1, x) = (q_1, \epsilon)$

$\delta\ (q_0, 0, x) = (q_1, x)$ $\delta\ (q_0, 0, z_0) = (q_0, z_0)$

Solution:

The PDA described is similar to the previous example Q.23 except that b = 1, a = 0 and Z = x. Thus, the CFG can be written as,

$S \rightarrow 1\ S\ 0\ 1 \mid 0 \mid \epsilon$

Q.25 Construct a PDA for the following language set:

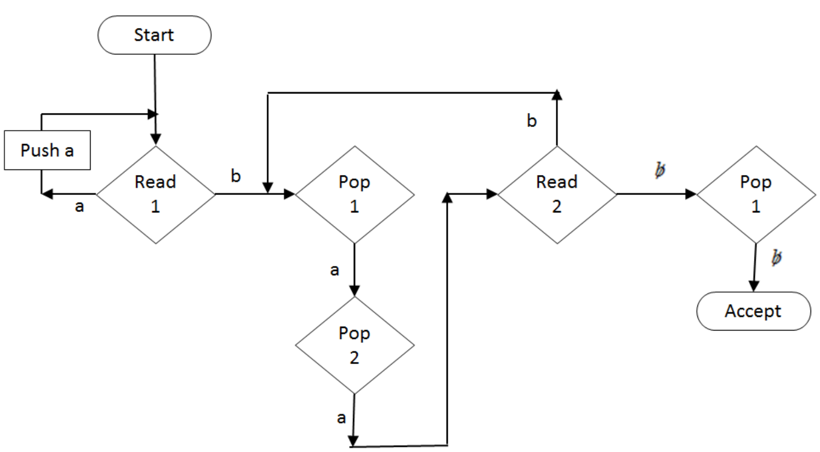
$L = \{a^{2n} b^n \mid n \geq 1\}$

Solution:

The CFG for the language L can be written as,

$S \rightarrow aa\ S\ b \mid aab$

Here, one b is matched with 2 a's. The only difference in this example and the example 6.3 from the book is that here after reading every b two a's are popped out (instead of one as in example 6.3).



Q.26 Convert the following CFG into PDA:

$S \to a\,B \mid b\,A$

$A \to a \mid a\,S \mid b\,A\,A$

$B \to b \mid b\,S \mid a\,B\,B$

Solution:

Let us convert the grammar to CNF which is suitable for drawing the PDA. The modified grammar is,

$S \to P\,B \mid Q\,A$

$A \to a \mid P\,S \mid Q\,R$

$B \to b \mid Q\,S \mid P\,T$

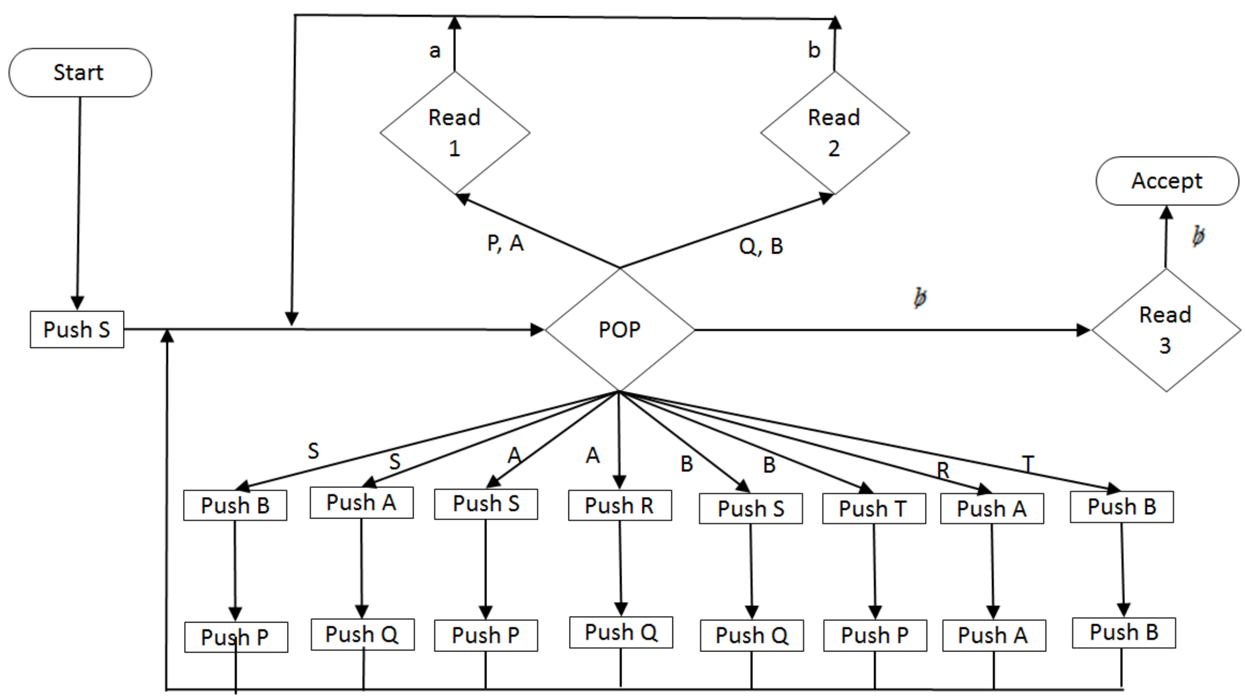
$R \to A\,A$

$T \to B\,B$

$P \to a$

$Q \to \text{b}$

The PDA can be drawn as below,



Q.27 Prove the following:

(i) CFLs are not closed under intersection.

(ii) CFLs are closed under Kleene closure.

Solution:

(i) Actually, CFLs are closed under intersection.

(ii) Refer to the section 6.8, theorem 6.3.

Q.28 Construct an NPDA defined over $\sum = \{a, b, c\}$ that accepts the language:

$L = \{\omega_1 c \omega_2 \mid \omega_1, \omega_2 \in \{\text{a, b}\}^*, \omega_1 \neq \omega_2\}$

Solution:

Let us write CFG for the language given,

$$L = \{\omega_1 c \omega_2 : \omega_1, \omega_2 \in \{a, b\}^*; \omega_1 \neq \omega_2\}$$

As 'ω_1' and 'ω_2' cannot be same, we can consider the scenario when 'ω_1' starts with a, 'ω_2' starts with b and vice versa. Same can be the case that they end with different symbols. We can write CFG with the above considerations as below,

$$S \to a\,A\,c\,b\,A \mid b\,A\,c\,a\,A \mid A\,a\,c\,A\,b \mid A\,b\,c\,A\,a$$

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

We can obtain the NPDA from the above CFG as,

Use algorithm discussed in the section 6.7 to convert the above CFG to NPDA.

Q.29 Write short notes on:

(1) Deterministic pushdown automata

(2) Equivalence of PDA and CFG

(3) Relative powers of NFA/DFA and NPDA/DPDA

(4) Use of CNF in PDA construction

(5) Closure properties of CFLs

Solution:

(1) Deterministic pushdown automata: Refer to the section 6.3.

(2) Equivalence of PDA and CFG: Refer to the section 6.7.

(3) Relative powers of NFA/DFA and NPDA/DPDA: Refer to the section 6.6.1.

(4) Use of CNF in PDA construction: Refer to the section 6.7.1.

(5) Closure properties of CFLs: Refer to the section 6.8.

Q.30 Construct deterministic PDA recognizing the following language:

$L = \{x \in (ab)^* \mid$ number of a's is more than number of b's$\}$

Solution:

The grammar below generates the language, $L = \{x \in (ab)^* \mid$ number of a's is more than number of b's$\}$

$X \rightarrow Y\ S \mid S\ Y$

$S \rightarrow a\ B \mid b\ A$

$A \rightarrow a \mid a\ S \mid b\ A\ A$

$B \rightarrow b \mid b\ S \mid a\ B\ B$

$Y \rightarrow a\ Y \mid a$

Use algorithm discussed in the section 6.7 to convert the above CFG to PDA.

