**CPSC 323 Handout No.9, Constructing PDA for a given CFG**

In handout 8, you learned how to construct PDA for a given language. Lets go through one more example and then learn how to construct a PDA when the grammar of the language is given.

**Example:** Given L={ anbn| n=0,1,2,……. } . Construct PDA to accept L

pop

accept

read

popp

Push a

read

Start

a b a ∆

∆( n=0)

∆

To Construct PDA using CFG, the grammar must be in special form which is called **Chomsky Normal Form (CNF)**. A CFG is in CNF form if each production (rule) is in one of the following format:

1. One-nonterminal 🡪 exactly **two** non-terminals (example: A🡪BC )
2. One-nonterminal 🡪one terminal (example: B🡪b )

First, we practice how to convert CFG to CNF format, and then I show you how to construct PDA using the CNF form of a CFG.

**Example**. Convert the following CFG to CNF format

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1. **Convert S**🡪**aSb | λ to CNF**   first we need to remove λ to have S🡪aSb | ab  Now to convert it to CNF, lets start from S🡪aSb, we group the right-hand-side into two non-  Terminal, either S🡪AX, A🡪a, X🡪 Sb or S🡪XB, X🡪aS, B🡪 b ( choose one).   |  |  | | --- | --- | | S🡪AX, A🡪a, X🡪 Sb | S🡪XB, X🡪aS, B🡪 b | | X🡪Sb is not CNF yet, Let X🡪SB, B🡪 b  Now altogether we have:  **S🡪AX**  **X🡪SB**  **B🡪b**  **A🡪a** | X🡪aS is not CNF, Let X🡪AS, A🡪a  Put all pieces together to get  **S🡪XB**  **X🡪AS**  **A🡪a**  **B🡪b** | |
| 1. Convert E🡪E+E | E\*E | r to CNF   Production E🡪E+E in CNF is : E🡪EX, X🡪+E, E🡪r , convert X🡪+E to CNF to get  E🡪EX, X🡪 PE, E🡪r, P🡪+  Production E🡪E\*E in CNF is : E🡪EY, Y🡪\*E, E🡪r . convert Y🡪\*E to CNF to get  E🡪EY, Y🡪TE, T🡪\*, E🡪 r  Altogether, we have  E🡪EX, X🡪PE, P🡪+, E🡪r  E🡪EY, Y🡪TE, T🡪\* |
| 1. Convert S🡪aXbY, X🡪aX | λ, Y🡪bY| b to CNF   Before doing the conversion, we need to remove λ  S🡪aXbY |aλbY 🡺 S🡪aXbY | abY  X🡪aX | aλ 🡺 X🡪aX | a  Now convert each production to CNF  S🡪aXbY 🡺 S🡪PQ, P🡪aX, Q🡪 bY 🡺 S🡪PO, P🡪AX, Q🡪BY, A🡪a, B🡪b  S🡪abY 🡺 S🡪WY, W🡪ab 🡺 S🡪WY, W🡪AB, A🡪a, B🡪b  X🡪aX 🡺 X🡪AX, A🡪 a  Y🡪bY 🡺 Y🡪BY, B🡪b  Altogether, we have  S🡪PO, S🡪WY , W🡪AB, P🡪AX, Q🡪BY, X🡪AX, Y🡪BY  A🡪a, B🡪b, Y🡪b |

Now let’s construct PDA for each CFG

|  |  |
| --- | --- |
| **Given CFG** | **PDA for the given CFG** |
| i. **Construct PDA for S**🡪**aSb | λ**  a.remove λ  S🡪 aSb | ab  b.convert to CNF  (i)S🡪AX  (ii)S🡪AB (iii)X🡪SB  (iv) A🡪a  (v) B🡪b | a b  readdd  read  (iv)A🡪a (v) B🡪b  ∆  read  POP  Push S  start  ∆    (i)S🡪AX (ii)S🡪AB (iii)X🡪SB  Push A  Push B  Push A  Push X  Push S  Push B  Accept |

To make the PDA more organized, I tooke care of the productions: non-terminal🡪 only one terminal (i.e A🡪a, B🡪b) on the top of the “POP” state and the other type of productions under the “POP” state(S🡪AX, S🡪AB, X🡪SB).

iv A🡪a v. B🡪b

POP

i. S🡪AX ii. S🡪AB iii. X🡪SB

* When you pop the stack, if the popped item is A or B, we will go up and use branch (iv) or (v) respectively. In case of A, we enter state “read” to read an “a” from input string and then go back to the “POP” state. The same when the popped item is B.
* If the popped item is S or X, we go to branch (i),(ii), or (iii) respectively. At each branch, we push the right hand side of each production in reverse order in stack.
* When the popped item is ∆, that indicates stack is empty. Test to make sure you are done with the input string by reading ∆. When you enter state “Accept”, the machine accepts the string and machine halts, otherwise machine will reject the input string.

readd

POP

∆

∆

Accept

Trace the above PDA for input string w=aabb∆

a b

read

readdd

(iv) A🡪a (v) B🡪b

∆

read

POP

Push S

start

∆

(i)S🡪AX (ii)S🡪AB (iii)X🡪SB

Push A

Push B

Push A

Push X

Push S

Push B

Accept

|  |  |  |
| --- | --- | --- |
| **State** | **Stack** | **Tape** |
| start | ∆ (stack is empty) | aabb∆….. |
| Push S | ∆S (S is on the top of stack) | aabb∆….. |
| Pop S, go to (i) S🡪AX | ∆ | aabb∆….. |
| Push X and A | ∆XA | aabb∆….. |
| Pop A, go to (iv) to enter read | ∆X | aabb∆….. |
| Read a, go to pop state | ∆X | abb∆….. |
| Pop X, go to (iii)X🡪SB | ∆ | abb∆….. |
| Push B and S | ∆BS | abb∆….. |
| Pop S, go to (ii)S🡪AB | ∆B | abb∆….. |
| Push B and A | ∆BBA | abb∆….. |
| Pop A, go to (iv) and enter read | ∆BB | bb∆….. |
| Read a, go to pop state | ∆BB | bb∆….. |
| Pop B, go to (v) B🡪b to enter read | ∆B | b∆….. |
| Read b, go to pop | ∆B | b∆….. |
| Pop B , go to (v) to enter read | ∆ | b∆….. |
| Read b, go to pop | ∆ | ∆….. |
| Pop ∆ | Stack is empty | ∆….. |
| Read ∆, enter accept state |  | …….. |
| **Accept** |  |  |

ii.Construct PDF for CFG : E🡪E+E | E\*E | r

There are no λ to remove, convert the CFG to CNF

(i) E🡪EX, (ii) X🡪PE, (iii) E🡪EY, (iv) Y🡪TE

(v) P🡪+, (vi) E🡪r , (vii) T🡪\*

|  |
| --- |
| **PDA for the given CFG** |
| + r \*  read  readdd  read  (v) P🡪+ (vi) E🡪r (vii) T🡪\*    read  POP  Push E  start  ∆  ∆    (i)E🡪EX (ii)X🡪PE (iii)E🡪EY (iv) Y🡪TE  Push T  Push E  Accept  Push P  Push E  Push E  Push X  Push E  Push Y |

Trace w=r+r\*r∆

|  |  |  |
| --- | --- | --- |
| **State** | **Stack** | **Tape** |
| start | ∆ | r+r\*r∆……. |
| Push E | ∆E | r+r\*r∆ |
| Pop E | ∆ | r+r\*r∆ |
| (i)Push X, push E | ∆XE | r+r\*r∆ |
| Pop E, go to (vi) | ∆X | r+r\*r∆ |
| (vi) read r | X∆ | +r\*r∆ |
| Pop X, go (ii) | ∆ | +r\*r∆ |
| (ii)push E, push P | ∆EP | +r\*r∆ |
| Pop p: go to (v) | ∆E | +r\*r∆ |
| (v) read + | E∆ | r\*r∆ |
| Pop E, go to (iii) | ∆ | r\*r∆ |
| (iii) Push Y, push E | ∆YE | r\*r∆ |
| Pop E, go to (vi) | ∆Y | r\*r∆ |
| (vi) read r, | ∆Y | \*r∆ |
| Pop Y, go to (iv) | ∆ | \*r∆ |
| (iv) push E, push T | ∆ET | \*r∆ |
| Pop T, go to(vii) | ∆E | \*r∆ |
| (vii) read \* | ∆E | r∆ |
| Pop E, go to (vi) | ∆ | r∆ |
| (vi) read r | ∆ | ∆ |
| Pop, go to read |  | ∆ |
| Read ∆, go to accept |  |  |
| Accept |  |  |

1. Construct PDA for : S🡪aXbY, X🡪aX | λ, Y🡪bY| b
2. Remove λ. The grammar becomes:

S🡪aXbY | abY

X🡪aX | a

1. Convert CFG to CNF. The CFG becomes:

(i) S🡪PO, (ii)S🡪WY , (iii)W🡪AB, (iv) P🡪AX,(v) Q🡪BY, (vi)X🡪AX,(vii) Y🡪BY

(v)A🡪a, (vi)B🡪b, (vii) Y🡪b

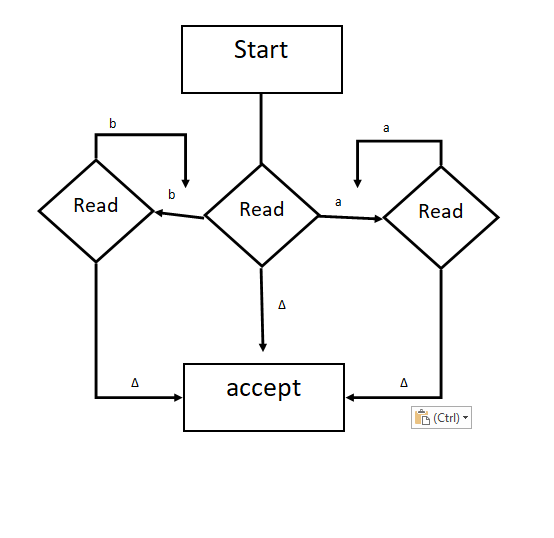
|  |
| --- |
| **PDA for the above CNF** |
| a b \*  read  readdd  read  (viii) A🡪a (ix) B🡪b (x) Y🡪b    Accept  read  Push E  POP  start  ∆ ∆      (i)S🡪PO (ii)S🡪WY (iii)W🡪AB (iv)P🡪AX (v)Q🡪BY (vi) X🡪AX (vii)Y🡪BY  Push B  Push Y  Push A  Push B  Push W  Push Y  Push A  Push X  Push X  O  PushY  PushA A  P  pushB |

**CPSC 323 Name: Richard Gresham row: 5**



**Assignment No.9 Name:** **Sean McCarthy**

1. (10 points) Construct PDA for L=a\* + b\*

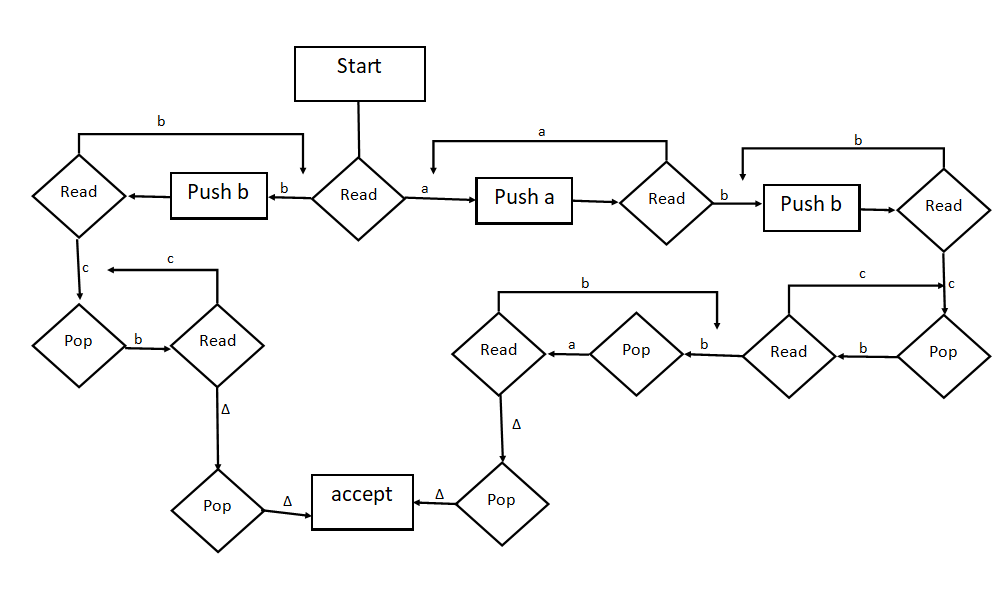


1. (10 points) Construct a PDA for L = { anbmcmbn| n=0,1,………; m=1,2,… }without using its CFG

anbmcmbn



bmcm anbmcmbn



1. (10 points) Convert this CFG to CNF

S🡪aSbb | λ

First remove λ

S 🡪 aSbb | abb

Then individually complete each side section.

S 🡪 aSbb | abb



Becomes CNF:

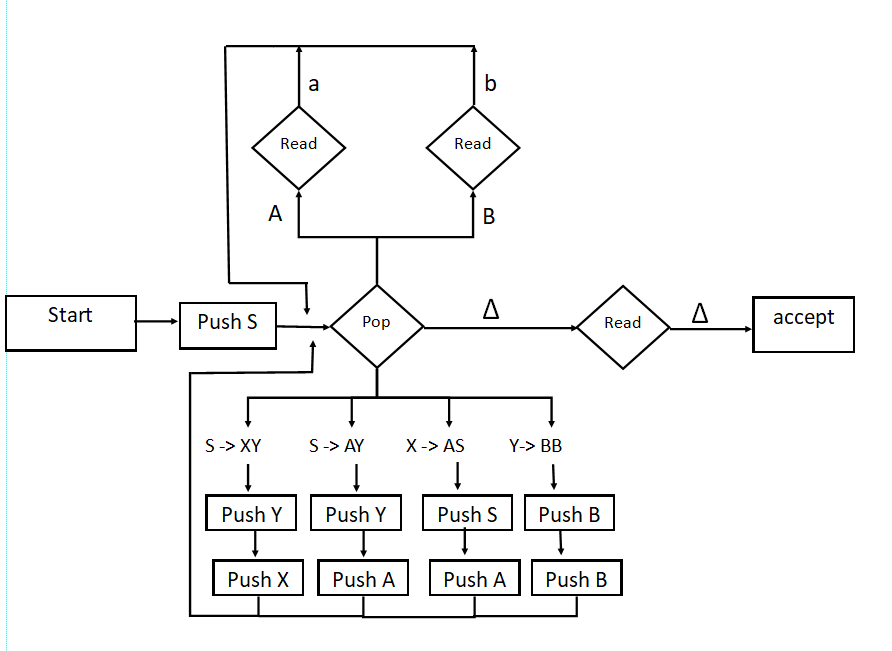
S 🡪 XY  
S 🡪 AY

X 🡪 AS

Y 🡪 BB

A 🡪 a

B 🡪 b



1. (10 points) Given CFG : S🡪XY, X🡪aX | a, Y🡪bY | b .

(a) convert the CFG to CNF

(b) construct a PDA for the CNF of the grammar.

(c) Trace the PDA for w = aaab∆

(a) convert the CFG to CNF

S 🡪 XY Already in CNF form

X 🡪 aX becomes

X 🡪 AX

A 🡪 a

X 🡪 a Already in CNF form

Y 🡪bY becomes

Y 🡪 BY

B 🡪 b

Y 🡪 b Already in CNF form

Full CNF:

S 🡪 XY

X 🡪 AX

Y 🡪 BY

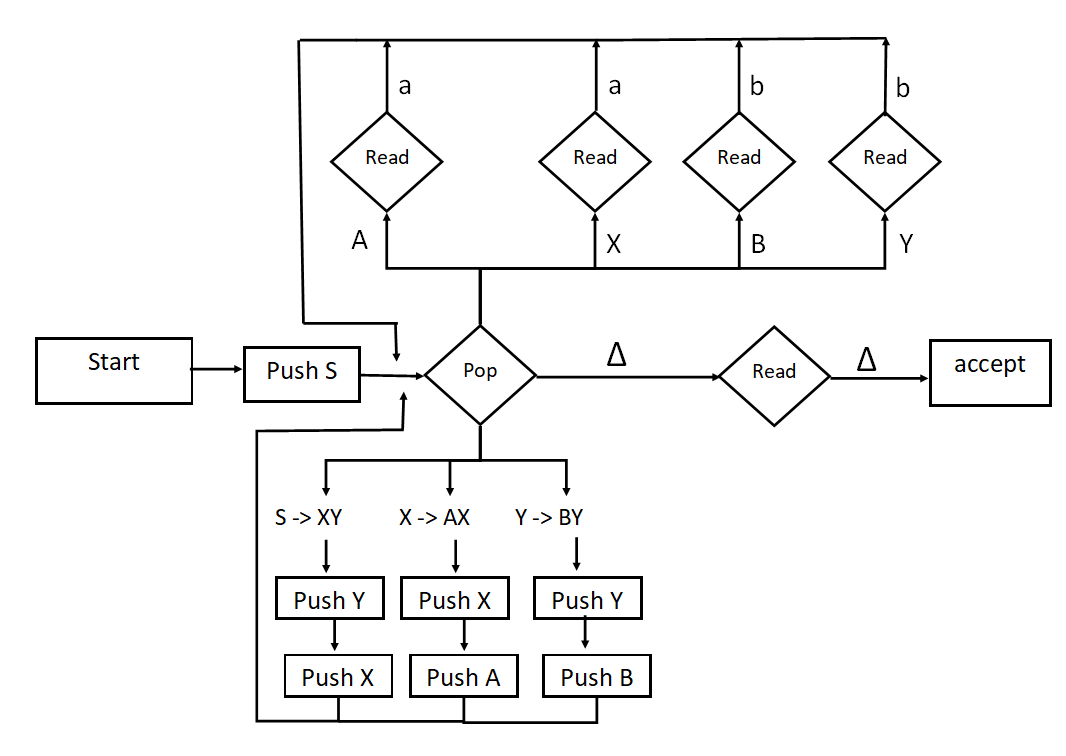
X 🡪 a

Y 🡪 b

A 🡪 a

B 🡪 b

(b) construct a PDA for the CNF of the grammar.

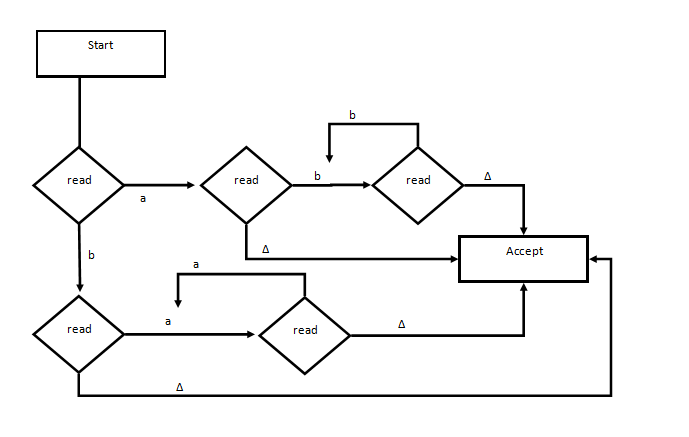


(c) Trace the PDA for w = aaab∆

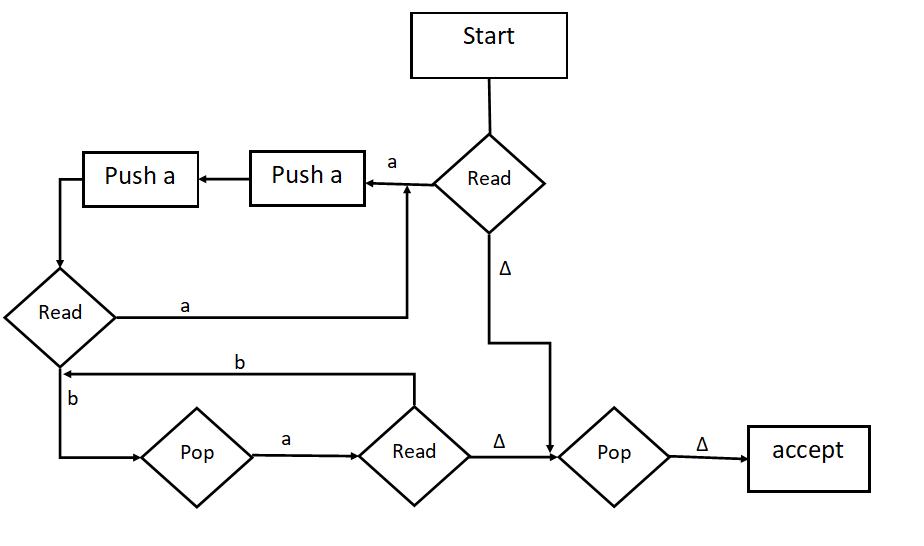
|  |  |  |
| --- | --- | --- |
| State | Stack | input |
| Start | ∆ | aaab∆ |
| Push S | ∆S | aaab∆ |
| Pop S | ∆ | aaab∆ |
| Push Y | ∆Y | aaab∆ |
| Push X | ∆YX | aaab∆ |
| Pop X | ∆Y | aaab∆ |
| Push X | ∆YX | aaab∆ |
| Push A | ∆YXA | aaab∆ |
| Pop A | ∆YX | aaab∆ |
| Read a | ∆YX | aab∆ |
| Pop X | ∆Y | aab∆ |
| Push X | ∆YX | aab∆ |
| Push A | ∆YXA | aab∆ |
| Pop A | ∆YX | aab∆ |
| Read a | ∆YX | ab∆ |
| Pop X | ∆Y | ab∆ |
| Read a | ∆Y | b∆ |
| Pop Y | ∆ | b∆ |
| Read b | ∆ | ∆ |
| Pop ∆ | - | ∆ |
| Read ∆ | - | - |
| Accept! |  |  |
|  |  |  |

**CPSC 323, Quiz No.9 Name: Richard Gresham row:5**

1. (5 points) Given L=ab\* + ba\*. Construct a PDA to accept L



1. ( 5 points) Given L={anb2n| n=0,1,2,…..}. Construct a PDA to accept L without using its Grammar.



1. (10 points )The CFG of the language in problem No. 2 is : S🡪aSbb | abb
2. Convert the CFG to CNF

S 🡪 aSbb | abb

Then individually complete each side section.

S 🡪 aSbb | abb



S 🡪 XY X 🡪 AS A 🡪 a Y 🡪 BB B 🡪 b

S 🡪 AY A 🡪 a Y 🡪 BB B 🡪 b

Becomes CNF:

S 🡪 XY  
S 🡪 AY

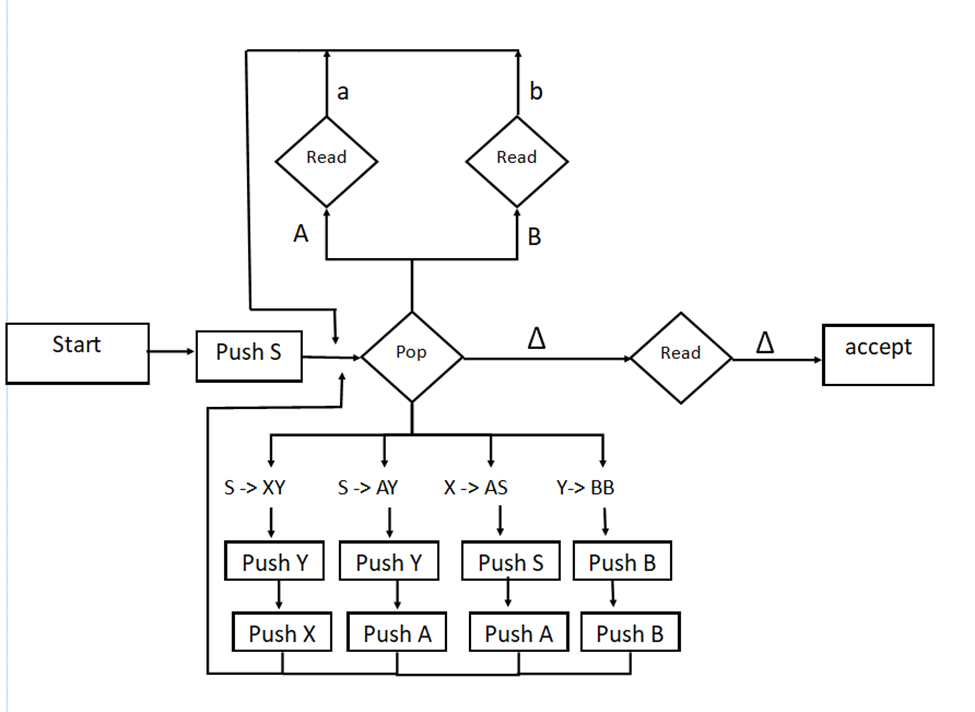
X 🡪 AS

Y 🡪 BB

A 🡪 a

B 🡪 b

1. Construct a PDA for the CFG in part (i).



1. Trace the PDA for input string w=aabbbb∆

|  |  |  |
| --- | --- | --- |
| State | Stack | Input |
| Start | ∆ | aabbbb∆ |
| Push S | ∆S | aabbbb∆ |
| Pop S | ∆ | aabbbb∆ |
| Push Y | ∆Y | aabbbb∆ |
| Push X | ∆YX | aabbbb∆ |
| Pop X | ∆Y | aabbbb∆ |
| Push S | ∆YS | aabbbb∆ |
| Push A | ∆YSA | aabbbb∆ |
| Pop A | ∆YS | aabbbb∆ |
| Read a | ∆YS | abbbb∆ |
| Pop S | ∆Y | abbbb∆ |
| Push Y | ∆YY | abbbb∆ |
| Push A | ∆YYA | abbbb∆ |
| Pop A | ∆YY | abbbb∆ |
| Read a | ∆YY | bbbb∆ |
| Pop Y | ∆Y | bbbb∆ |
| Push B | ∆YB | bbbb∆ |
| Push B | ∆YBB | bbbb∆ |
| Pop B | ∆YB | bbbb∆ |
| Read b | ∆YB | bbb∆ |
| Pop B | ∆Y | bbb∆ |
| Read b | ∆Y | bb∆ |
| Pop Y | ∆ | bb∆ |
| Push B | ∆B | bb∆ |
| Push B | ∆BB | bb∆ |
| Pop B | ∆B | bb∆ |
| Read b | ∆B | b∆ |
| Pop B | ∆ | b∆ |
| Read b | ∆ | ∆ |
| Pop ∆ | - | ∆ |
| Read ∆ | - | - |
| ACCEPT!!! |  |  |