Context Free Grammars and Pushdown Automata

Module 2

Dr. A K Yadav

April 28, 2022

Outline

1. CFG: Formal Definition
2. Derivation and Syntax Trees
3. Ambiguous Grammar
4. Simplification Forms
5. Properties of CFL
6. Normal Forms (CNF and GNF)
7. Pumping Lemma for Context Free Language
8. Decision Algorithms
9. Linear Grammar
10. Pushdown Automata
11. Relationship between PDA and CFL

Context Free Grammar

- **Finite Automata** accepts all regular languages.
  - Simple languages such as
    - $a^n b^n : n = 0, 1, 2, \ldots.$
    - $w : w$ is a Palindrome

    are not regular and thus no finite automata accepts them.
- Context Free Languages are a larger class of languages that encompasses all regular languages and many others including above examples.
- Languages generated by context free grammar are called Context free languages.
- Context free grammar are more expressive than finite automata: If a language L is accepted by a finite automata, then L can be generated by a context-free grammar, *While opposite is not true*

Context-Free Grammars

- A Context-free grammar is a 4-tuple (V_n, Σ, P, S)
  - V_n is set of *Variables*.
  - Σ is set of terminals.
  - P is set of Productions.
  - S is the start symbol.
- A Grammar G is Context free, if every production is of the form $A \to \alpha$, where $A \in V_N$ and $\alpha \in (V_N \cup \Sigma)^*$

Example:CFG

- Contruct a CFG generating all integers (with sign).

Example:CFG

- Contruct a CFG generating all integers (with sign).

Solution: Let $G = (V, \Sigma, P, S)$

where $V = \{S, < sign >, < digit >, < integer >\}$

$\Sigma = \{0, 1, 2, 3, ...9, +, -\}$

P consists of:

S $\rightarrow < sign >< integer >$

$< sign > \rightarrow +|-$

$< integer > \rightarrow < digit >< integer > \mid < digit >$

$< digit > \rightarrow 0|1|2|\ldots|9$

Example:CFG

- Contruct a CFG generating all integers (with sign).

Solution: Let $G = (V, \Sigma, P, S)$

where $V = \{S, < sign >, < digit >, < integer >\}$

$\Sigma = \{0, 1, 2, 3, ...9, +, -\}$

P consists of:

S $\rightarrow < sign >< integer >$

$< sign > \rightarrow +|-$

$< integer > \rightarrow < digit >< integer > \mid < digit >$

$< digit > \rightarrow 0|1|2| \ldots |9$

Derivation for -42

S$\rightarrow < sign >< integer >$

$\Longrightarrow$ -$< integer >$

$\Longrightarrow$ -$< digit >< integer >$

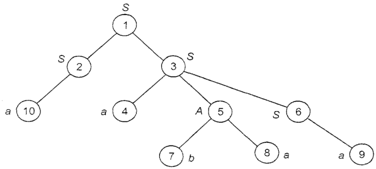
$\Longrightarrow$ -4$< digit >$

$\Longrightarrow$ -42

Derivation Trees

- Trees are used for derivation of CFG.
- Definition: A derivation tree (or a parse tree) for a CFG $G = (V, \Sigma, P, S)$ is a tree satisfying:
  - Every vertex has a label which is variable/terminal/$\wedge$.
  - The root has label S.
  - The label of the internal vertex is a variable.
  - If vertices $n_1, n_2, ...n_k$ written with labels $X_1, X_2, ...X_k$ are sons of vertex 'n' with label A, then $A \rightarrow X_1X_2...X_k$ is a production in P.
  - A vertex 'n' is a leaf if its label is $a \in \Sigma$ or $\wedge$; 'n' is the only son of its father if its label is $\wedge$
- Let G = ({S,A},{a,b},P,S) where P consists of

$$S \rightarrow aAS|a|SS, A \rightarrow SbA|ba$$



Derivation Trees

- Yield of Derivation Tree:is a concatenation of labels of the leaves without repetition in the left to right ordering.
  Example: aabaa
- Subtree of a Derivation Tree T is a tree:
  - whose root is some vertex 'v' of T,
  - whose vertices are descendants of 'v' together with their labels.
  - whose edges are those connecting the descendants of v.

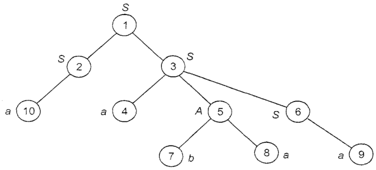


Figure 1: Derivation Tree T

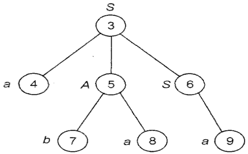


Figure 2: Sub tree of Tree T

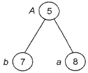


Figure 3: Sub tree of Tree T

Derivation Trees

Theorem 1:

Let G=(V,Σ, P, S) be a CFG. Then, $S \xRightarrow{*} \alpha$ if and only if there is a derivation tree for G with yield α.

- Example: Consider G whose productions are $S \to aAS|a, A \to SbA|SS|ba$. Show that $S \xRightarrow{*} aabbaa$ and Construct a derivation tree whose yield is $aabbaa$.

Derivation Trees

Theorem 1:

Let G=(V,Σ, P, S) be a CFG. Then, $S \stackrel{*}{\Longrightarrow} \alpha$ if and only if there is a derivation tree for G with yield α.

- Example: Consider G whose productions are $S \rightarrow aAS|a, A \rightarrow SbA|SS|ba$. Show that $S \stackrel{*}{\Longrightarrow} aabbaa$ and Construct a derivation tree whose yield is $aabbaa$.
- Case 1:
  $S \Longrightarrow aAS \Longrightarrow aSbAS \Longrightarrow aabAS \Longrightarrow a^2bbaS \Longrightarrow aabbaa$

Derivation Trees

Theorem 1:

Let G=(V,Σ, P, S) be a CFG. Then, $S \overset{*}{\Longrightarrow} \alpha$ if and only if there is a derivation tree for G with yield α.

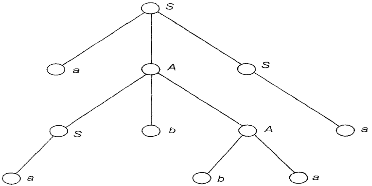
- Example: Consider G whose productions are $S \rightarrow aAS|a, A \rightarrow SbA|SS|ba$. Show that $S \overset{*}{\Longrightarrow} aabbaa$ and Construct a derivation tree whose yield is $aabbaa$.
- Case 1:
  $S \Longrightarrow aAS \Longrightarrow aSbAS \Longrightarrow aabAS \Longrightarrow a^2bbaS \Longrightarrow aabbaa$
- Case 2:
  $S \Longrightarrow aAS \Longrightarrow aAa \Longrightarrow aSbAa \Longrightarrow aSbbaa \Longrightarrow aabbaa$

Derivation Trees

Theorem 1:

Let G=(V,Σ, P, S) be a CFG. Then, $S \overset{*}{\Longrightarrow} \alpha$ if and only if there is a derivation tree for G with yield α.

- Example: Consider G whose productions are $S \rightarrow aAS|a, A \rightarrow SbA|SS|ba$. Show that $S \overset{*}{\Longrightarrow} aabbaa$ and Construct a derivation tree whose yield is $aabbaa$.
- Case 1: $S \Longrightarrow aAS \Longrightarrow aSbAS \Longrightarrow aabAS \Longrightarrow a^2bbaS \Longrightarrow aabbaa$
- Case 2: $S \Longrightarrow aAS \Longrightarrow aAa \Longrightarrow aSbAa \Longrightarrow aSbbaa \Longrightarrow aabbaa$
- Case 3: $S \Longrightarrow aAS \Longrightarrow aSbAS \Longrightarrow aSbAa \Longrightarrow aabAa \Longrightarrow aabbaa$



Derivation Trees

- Left most derivation: A derivation $A \xRightarrow{*} w$ is called left-most derivation, if we apply production only to the left most variable at every step.
- Right most derivation: A derivation $A \xRightarrow{*} w$ is a right most derivation, if we apply production to right most variable at each step.

Theorem

If $A \xRightarrow{*} w$ in G, then there is a left most derivation of w.

Derivation Trees

- Example: Let G be the grammar $S \to 0B|1A$, $A \to 0|0S|1AA$, $B \to 1|1S|0BB$. For the string 00110101, find:
  - the leftmost derivation
  - the rightmost derivation
  - The derivation tree

Derivation Trees

- Example: Let G be the grammar $S \to 0B|1A$, $A \to 0|0S|1AA$, $B \to 1|1S|0BB$. For the string 00110101, find:
  - the leftmost derivation
  - the rightmost derivation
  - The derivation tree
- Leftmost derivation:

  $S \implies 0B \implies 00BB \implies 001B \implies 0011S \implies 0^21^20B \implies$

  $0^21^201S \implies 0^21^2010B \implies 0^21^20101$

Derivation Trees

- Example: Let G be the grammar $S \to 0B|1A$, $A \to 0|0S|1AA$, $B \to 1|1S|0BB$. For the string 00110101, find:
  - the leftmost derivation
  - the rightmost derivation
  - The derivation tree
- Leftmost derivation:

  $S \implies 0B \implies 00BB \implies 001B \implies 0011S \implies 0^2 1^2 0B \implies$
  $0^2 1^2 01S \implies 0^2 1^2 010B \implies 0^2 1^2 0101$
- Rightmost derivation:

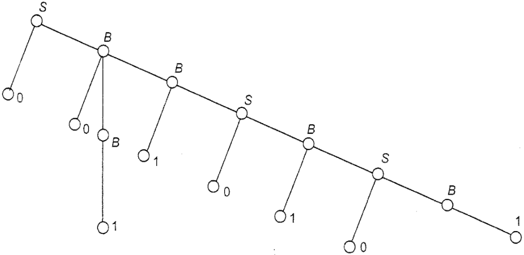
  $S \implies 0B \implies 00BB \implies 00B1S \implies 00B10B \implies 0^2 B101S \implies$
  $0^2 B1010B \implies 0^2 B10101 \implies 0^2 110101$

Derivation Trees

- Example: Let G be the grammar $S \to 0B|1A$, $A \to 0|0S|1AA$, $B \to 1|1S|0BB$. For the string 00110101, find:
  - the leftmost derivation
  - the rightmost derivation
  - The derivation tree
- Leftmost derivation:

  $S \implies 0B \implies 00BB \implies 001B \implies 0011S \implies 0^21^20B \implies$
  $0^21^201S \implies 0^21^2010B \implies 0^21^20101$
- Rightmost derivation:

  $S \implies 0B \implies 00BB \implies 00B1S \implies 00B10B \implies 0^2B101S \implies$
  $0^2B1010B \implies 0^2B10101 \implies 0^2110101$
- Derivation tree:



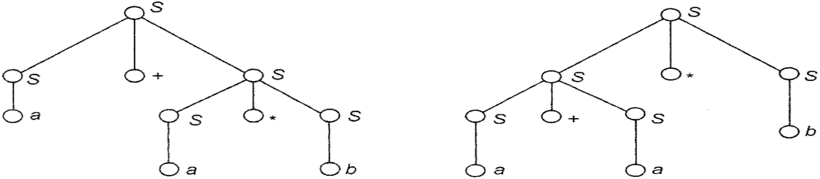
Ambiguity in CFG

In books selected information is given.

Ambiguity in CFG

In books selected information is given.

- A terminal string $w \in L(G)$ is ambiguous if there exist two or more derivation trees for 'w' (or there exist two or more left most derivation of w).
- Example: $G = (\{S\}, \{a, b, +, *\}, P, S)$, where P consists of $S \to S + S|S * S|a|b$. We have two derivation trees for a+a*b:



- $S \implies S + S \implies a + S \implies a + S * S \implies a + a * S \implies a + a * b$
- $S \implies S * S \implies S + S * S \implies a + S * S \implies a + a * S \implies a + a * b$

Ambiguity in CFG

Example:

- If G is grammar $S \rightarrow SbS|a$. Show that the given grammar is ambigious.

Ambiguity in CFG

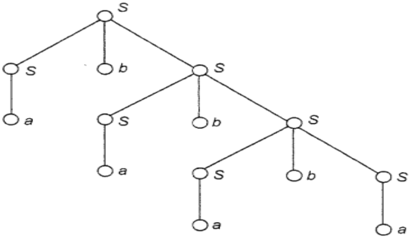
Example:

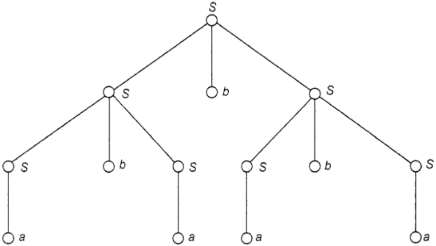
- If G is grammar $S \rightarrow SbS|a$. Show that the given grammar is ambigious.
- For $w = ababa ba$

Ambiguity in CFG

Example:

- If G is grammar $S \rightarrow SbS|a$. Show that the given grammar is ambigious.
- For $w = abababa$

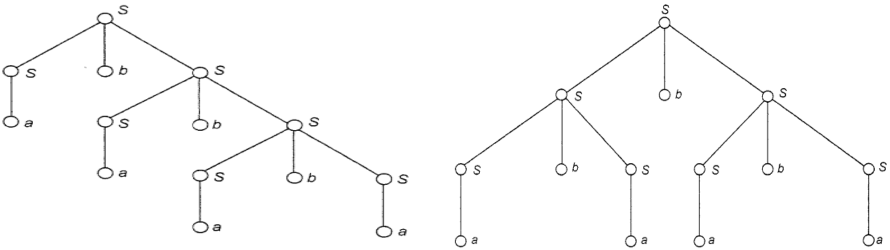




Ambiguity in CFG

Example:

- If G is grammar $S \rightarrow SbS|a$. Show that the given grammar is ambigious.
- For $w = abababa$



- Thus, G is ambiguous.

Simplification of CFG

- We eliminate following in order to simplify a grammar:
  - Useless variables
  - Unit productions
  - Null Productions

Eliminating Useless Variables

- Those variables which are not deriving any terminal string and which are not reachable are known as Useless Variables.
- Example: $S \rightarrow AB$
  $A \rightarrow a|B$
  $B \rightarrow b|C$
  $C \rightarrow aC$
  $D \rightarrow b$

Eliminating Useless Variables

- Those variables which are not deriving any terminal string and which are not reachable are known as Useless Variables.
- Example: $S \rightarrow AB$
  $A \rightarrow a|B$
  $B \rightarrow b|C$
  $C \rightarrow aC$
  $D \rightarrow b$
- C is not deriving any terminal, C is not deriving any terminal $\Longrightarrow$ useless variable, $\therefore$ remove C

Eliminating Useless Variables

- Those variables which are not deriving any terminal string and which are not reachable are known as Useless Variables.
- Example: $S \rightarrow AB$
  $A \rightarrow a|B$
  $B \rightarrow b|C$
  $C \rightarrow aC$
  $D \rightarrow b$
- C is not deriving any terminal, C is not deriving any terminal $\implies$ useless variable, $\therefore$ remove C
  $S \rightarrow AB$, $A \rightarrow a|B$, $B \rightarrow b$, $D \rightarrow b$
  D is not included in S $\implies$ remove D

Eliminating Useless Variables

- Those variables which are not deriving any terminal string and which are not reachable are known as Useless Variables.
- Example: $S \to AB$
  $A \to a|B$
  $B \to b|C$
  $C \to aC$
  $D \to b$
- C is not deriving any terminal, C is not deriving any terminal $\Longrightarrow$ useless variable, $\therefore$ remove C
  $S \to AB,\ A \to a|B,\ B \to b,\ D \to b$
  D is not included in $S \Longrightarrow$ remove D
  $\therefore S \to AB,\ A \to a|B,\ B \to b$

Eliminating Unit Productions

- Production of the form $A \rightarrow B$ is known as Unit Production

Eliminating Unit Productions

- Production of the form $A \rightarrow B$ is known as Unit Production
- Example: $S \rightarrow A$, $A \rightarrow B$, $B \rightarrow C$, $C \rightarrow D$, $D \rightarrow a$
  appears as chainlike process

Eliminating Unit Productions

- Production of the form $A \rightarrow B$ is known as Unit Production
- Example: $S \rightarrow A$, $A \rightarrow B$, $B \rightarrow C$, $C \rightarrow D$, $D \rightarrow a$
  appears as chainlike process
  Instead $S \rightarrow a$ will serve the purpose
  $\therefore$ required grammar is $S \rightarrow a$

Eliminating Unit Productions

- Example: $S \to Aa|B,\ B \to A|bb,\ A \to a|bc|B$

Eliminating Unit Productions

- Example: $S \rightarrow Aa|B$, $B \rightarrow A|bb$, $A \rightarrow a|bc|B$
  Starting from last to first
  $A \rightarrow B$ is a unit production, replace B by its R.H.S.

Eliminating Unit Productions

- Example: $S \rightarrow Aa|B$, $B \rightarrow A|bb$, $A \rightarrow a|bc|B$
  Starting from last to first
  $A \rightarrow B$ is a unit production, replace B by its R.H.S.
  $A \rightarrow a|bc|A|bb$
  $B \rightarrow a|bc|A|bb$
  $S \rightarrow Aa|a|bc|A|bb$

Eliminating Unit Productions

- Example: $S \rightarrow Aa|B$, $B \rightarrow A|bb$, $A \rightarrow a|bc|B$

  Starting from last to first

  $A \rightarrow B$ is a unit production, replace B by its R.H.S.

  $A \rightarrow a|bc|A|bb$

  $B \rightarrow a|bc|A|bb$

  $S \rightarrow Aa|a|bc|A|bb$

  Now, remove unit productions

  $A \rightarrow a|bc|bb$

  $B \rightarrow a|bc|bb$

  $S \rightarrow Aa|bc|a|bb$

Eliminating Unit Productions

- Example: $S \rightarrow Aa|B$, $B \rightarrow A|bb$, $A \rightarrow a|bc|B$
  Starting from last to first
  $A \rightarrow B$ is a unit production, replace B by its R.H.S.
  $A \rightarrow a|bc|A|bb$
  $B \rightarrow a|bc|A|bb$
  $S \rightarrow Aa|a|bc|A|bb$
  Now, remove unit productions
  $A \rightarrow a|bc|bb$
  $B \rightarrow a|bc|bb$
  $S \rightarrow Aa|bc|a|bb$
  Now, S does not has B, $\implies$ remove B
  $S \rightarrow Aa|bc|a|bb$
  $A \rightarrow a|bc|bb$

Eliminating Null Productions

- Any production of the form $A \to \epsilon$ is called Null Production.
- Example: $S \to aAb$, $A \to aAb|\epsilon$

Eliminating Null Productions

- Any production of the form $A \to \epsilon$ is called Null Production.
- Example: $S \to aAb$, $A \to aAb|\epsilon$
  Replace A with ϵ in each production containing A and add it to grammar without ϵ
  $S \to aAb|ab$
  $A \to aAb|ab$

Simplifying CFG

- Example: Construct reduced grammar equivalent to grammar G whose productions are:
  $S \rightarrow AB|CA$, $B \rightarrow BC|AB$, $A \rightarrow a$, $C \rightarrow aB|b$

Simplifying CFG

- Example: Construct reduced grammar equivalent to grammar G whose productions are:
  $S \rightarrow AB|CA$, $B \rightarrow BC|AB$, $A \rightarrow a$, $C \rightarrow aB|b$
  Here, B is not deriving any terminals $\therefore$ remove B

Simplifying CFG

- Example: Construct reduced grammar equivalent to grammar G whose productions are:
  $S \rightarrow AB|CA,\ B \rightarrow BC|AB,\ A \rightarrow a,\ C \rightarrow aB|b$
  Here, B is not deriving any terminals $\therefore$ remove B
  $S \rightarrow CA,\ A \rightarrow a,\ C \rightarrow b$
  $\therefore$ New Grammar $G' = (\{S, A, C\}, \{a, b\}, P, S)$

Simplifying CFG

- Example: Construct reduced grammar equivalent to grammar G whose productions are:
  $S \to AB|CA,\ B \to BC|AB,\ A \to a,\ C \to aB|b$
  Here, B is not deriving any terminals $\therefore$ remove B
  $S \to CA,\ A \to a,\ C \to b$
  $\therefore$ New Grammar $G' = (\{S, A, C\}, \{a, b\}, P, S)$
- Question: Find the reduced grammar equivalent to G.
  $S \to aAa,\ A \to bBB,\ B \to ab,\ C \to aB$

Normal forms of CFG

- Chomsky Normal Form
- Greibach Normal Form

Chomsky Normal Form

A CFG is in Chomsky normal form if all productions are of the form:
$A \rightarrow BC$ or $A \rightarrow a$ and $S \rightarrow \wedge$ if $\wedge \in L(G)$
where A,B,C are variables and a is a terminal. When $\wedge$ is in $L(G)$, we assume that S does not appear on the R.H.S. of any production.

Chomsky Normal Form

A CFG is in Chomsky normal form if all productions are of the form:
$A \rightarrow BC$ or $A \rightarrow a$ and $S \rightarrow \wedge$ if $\wedge \in L(G)$
where A,B,C are variables and a is a terminal. When $\wedge$ is in $L(G)$, we assume that S does not appear on the R.H.S. of any production.

- Example: The Grammar in the form:
  $S \rightarrow AS|a$, $A \rightarrow SA|b$

Chomsky Normal Form

A CFG is in Chomsky normal form if all productions are of the form:
$A \rightarrow BC$ or $A \rightarrow a$ and $S \rightarrow \wedge$ if $\wedge \in L(G)$
where A,B,C are variables and a is a terminal. When $\wedge$ is in $L(G)$, we assume that S does not appear on the R.H.S. of any production.

- Example: The Grammar in the form:
  $S \rightarrow AS|a$, $A \rightarrow SA|b$
- Reduction to Chomsky normal form
  1. Elimination of null productions and unit productions
  2. Elimination of terminals on R.H.S.
  3. Restricting the number of variables on R.H.S.

Chomsky Normal Form

- Example: Convert the grammar G with Productions as: $S \rightarrow ABa$, $A \rightarrow aab$, $B \rightarrow Ac$ to chomsky normal form

Chomsky Normal Form

- Example: Convert the grammar G with Productions as:
  $S \rightarrow ABa$, $A \rightarrow aab$, $B \rightarrow Ac$ to chomsky normal form
  Let us assume few new variables:
  $X \rightarrow a$, $Y \rightarrow b$, $Z \rightarrow c$

Chomsky Normal Form

- Example: Convert the grammar G with Productions as:
  $S \rightarrow ABa$, $A \rightarrow aab$, $B \rightarrow Ac$ to chomsky normal form
  Let us assume few new variables:
  $X \rightarrow a$, $Y \rightarrow b$, $Z \rightarrow c$
  gives $S \rightarrow ABX$, $A \rightarrow XXY$, $B \rightarrow AZ$

Chomsky Normal Form

- Example: Convert the grammar G with Productions as:
  $S \rightarrow ABa$, $A \rightarrow aab$, $B \rightarrow Ac$ to chomsky normal form
  Let us assume few new variables:
  $X \rightarrow a$, $Y \rightarrow b$, $Z \rightarrow c$
  gives $S \rightarrow ABX$, $A \rightarrow XXY$, $B \rightarrow AZ$
  Now, Assuming $Q \rightarrow XX$, $P \rightarrow AB$

Chomsky Normal Form

- Example: Convert the grammar G with Productions as: $S \to ABa$, $A \to aab$, $B \to Ac$ to chomsky normal form
  Let us assume few new variables:
  $X \to a$, $Y \to b$, $Z \to c$
  gives $S \to ABX$, $A \to XXY$, $B \to AZ$
  Now, Assuming $Q \to XX$, $P \to AB$
  gives, $S \to PX$, $A \to QY$, $B \to AZ$, $X \to a$, $Y \to b$, $Z \to c$, $Q \to XX$, $P \to AB$

Chomsky Normal Form

- Example: Convert the grammar G with Productions as: $S \rightarrow ABa$, $A \rightarrow aab$, $B \rightarrow Ac$ to chomsky normal form
  Let us assume few new variables:
  $X \rightarrow a$, $Y \rightarrow b$, $Z \rightarrow c$
  gives $S \rightarrow ABX$, $A \rightarrow XXY$, $B \rightarrow AZ$
  Now, Assuming $Q \rightarrow XX$, $P \rightarrow AB$
  gives, $S \rightarrow PX$, $A \rightarrow QY$, $B \rightarrow AZ$, $X \rightarrow a$, $Y \rightarrow b$, $Z \rightarrow c$, $Q \rightarrow XX$, $P \rightarrow AB$
- Convert the given grammar to CNF
  - S $\rightarrow$ aSb |ab
  - S $\rightarrow$ aAB |Bb
    A $\rightarrow$ a , B $\rightarrow$ b
  - S $\rightarrow$ bA |aB
    A $\rightarrow$ bAA |aS |a
    B $\rightarrow$ aBB |bS |b

Greibach Normal Form

- A CFG is said to be in Greibach Normal form, if all the productions have the form $A \rightarrow aX$ (or $A \rightarrow a$ when X is $\wedge$), where $a \in \Sigma$ and $X \in V^*$ (X may be $\wedge$) and $S \rightarrow \wedge$ if $\wedge \in L(G)$ and S does not appear on the R.H.S. of any production
- Example: $S \rightarrow aAB|bBB|bB|a$
  $A \rightarrow aA|bB|b$, $B \rightarrow b$

Lemma 1:

Let $G = (V, \Sigma, P, S)$ be a CFG. Let $A \rightarrow B\gamma$ be an A-production in P. Let the B-productions be $B \rightarrow \beta_1|\beta_2|\ldots|\beta_k$. Define $P_1 = (P - \{A \rightarrow B\gamma\}) \cup \{A \rightarrow \beta_i\gamma | 1 \leq i \leq k\}$. Then $G_1 = (V, \Sigma, P_1, S)$ is a context-free grammar equivalent to G.

Greibach Normal Form

Lemma 2:

Let $G = (V, \Sigma, P, S)$ be a CFG. Let the set of A-productions be $A \rightarrow A\alpha_1 | A\alpha_2 | \dots | A\alpha_r | \beta_1 | \beta_2 | \dots | \beta_s |$ (β_i's do not start with A). Let Z be a new variable. Let $G_1 = (V \cup \{Z\}, \Sigma, P_1, S)$, where P_1 is defined as follows:

1. The set of A-productions in P_1 are
   $A \rightarrow \beta_1 | \beta_2 | \dots | \beta_s$
   $A \rightarrow \beta_1 Z | \beta_2 Z | \dots | \beta_s Z$
2. The set of Z-productions in P_1 are
   $Z \rightarrow \alpha_1 | \alpha_2 | \dots | \alpha_r$
   $Z \rightarrow \alpha_1 Z | \alpha_2 Z | \dots | \alpha_r Z$
3. The productions for the other variables are as in P.

Then G_1 is a CFG and equivalent to G.

Greibach Normal Form

- Reduction to Greibach normal form
  1. Elimination of null productions and unit productions
  2. Elimination of terminals on R.H.S. except the first leftmost terminal.
  3. Make all production starting with a terminal if not.
- Example: Convert the grammar G into equivalent GNF:
  $S \rightarrow abSb|aa$
  Solution: Let $A \rightarrow a$, $B \rightarrow b$
  $\therefore S \rightarrow aBSB|aA$, $A \rightarrow a$, $B \rightarrow b$
- Convert the grammar S → ab |aS |aaS into GNF
  Solution: Let B → b, A → a, S → aB |aS |aAS
- Exercise:
  1. Convert the grammar $S \rightarrow AA|a, A \rightarrow SS|b$ into GNF.
  2. Convert the grammar $S \rightarrow AB, A \rightarrow BS|b, B \rightarrow SA|a$ into GNF.
  3. Convert the grammar $E \rightarrow E+T|T, T \rightarrow T*F|F, F \rightarrow (E)|a$ into GNF.

Pumping Lemma for Context Free Language

Let L be an infinite context free language. Then, there exists some positive integer n such that:

1. Every $z \in L$ with $|z| \geq n$ can be written as $uvwxy$ for some strings u, v, w, x, y.
2. $|vx| \geq 1$
3. $|vwx \leq n$
4. $uv^k wx^k y \in L$ for all $k \geq 0$

Application: We use the pumping lemma to show that a language L is not a context free language.

Procedure: We assume that L is context-free. By applying the pumping lemma we get a contradiction. The procedure can be carried out by using the following steps:

Step 1 Assume L is context-free. Let n be the natural number obtained by using the pumping lemma.

Pumping Lemma for Context Free Language

Step 2 Choose $z \in L$ so that $|z| \geq n$. Write $z = uvwxy$ using the pumping lemma.

Step 3 Find a suitable k so that $uv^kwx^ky \notin L$. This is a contradiction, and so L is not context-free.

Pumping Lemma for Context Free Language

Ques: Show that the language

L=$\{a^n b^n c^n$: n $\geq 0\}$ is not context free.

Pumping Lemma for Context Free Language

Ques: Show that the language
L=$\{a^n b^n c^n$: n $\geq$ 0$\}$ is not context free.

Solution: Let L be Context free

Let w be a string in L

For n=4, string becomes aaaabbbbcccc

Pumping Lemma for Context Free Language

Ques: Show that the language

L=$\{a^n b^n c^n$: n $\geq$ 0$\}$ is not context free.

Solution: Let L be Context free

Let w be a string in L

For n=4, string becomes aaaabbbbcccc

By Pumping Lemma, w=uvxyz

w=aaaabbbbcccc

Pumping Lemma for Context Free Language

Ques: Show that the language

L=$\{a^n b^n c^n$: n $\geq 0\}$ is not context free.

Solution: Let L be Context free

Let w be a string in L

For n=4, string becomes aaaabbbbcccc

By Pumping Lemma, w=uvxyz

w=aaaabbbbcccc

Case 1: If vxy contain only a, b or c, then on pumping. It won't be in L

Case 2: If string contains any two either ab or bc, then pumped string will contain $a^k b^l c^m$ with $k \neq l \neq m$ or it will not be in the order, so does not belong to L

Pumping Lemma for Context Free Language

Ques: Show that the language

L=$\{a^n b^n c^n$: n $\geq 0\}$ is not context free.

Solution: Let L be Context free

Let w be a string in L

For n=4, string becomes aaaabbbbcccc

By Pumping Lemma, w=uvxyz

w=aaaabbbbcccc

Case 1: If vxy contain only a, b or c, then on pumping. It won't be in L

Case 2: If string contains any two either ab or bc, then pumped string will contain $a^k b^l c^m$ with $k \neq l \neq m$ or it will not be in the order, so does not belong to L

$\therefore$ L is not context free.

Pumping Lemma for Context Free Language

Ques: Show that the language

L=$\{a^n b^n c^n$: n $\geq$ 0$\}$ is not context free.

Solution: Let L be Context free

Let w be a string in L

For n=4, string becomes aaaabbbbcccc

By Pumping Lemma, w=uvxyz

w=aaaabbbbcccc

Case 1: If vxy contain only a, b or c, then on pumping. It won't be in L

Case 2: If string contains any two either ab or bc, then pumped string will contain $a^k b^l c^m$ with $k \neq l \neq m$ or it will not be in the order, so does not belong to L

$\therefore$ L is not context free.

- Show that following L are not Context free
  1. L=$\{ww : w \in \{a, b\}*\}$
  2. L=$\{a^n b^j : n = j^2\}$
  3. L= $\{a^{n!} : n \geq 0\}$

Pumping Lemma for Context Free Language

Ques: Show that $L = \{a^p | p \text{ is a prime } \}$ is not a context-free language

Pumping Lemma for Context Free Language

Ques: Show that $L = \{a^p | p \text{ is a prime } \}$ is not a context-free language

Solution:

1. Let L be Context free, Let w be a string in L

Pumping Lemma for Context Free Language

Ques: Show that $L = \{a^p | p \text{ is a prime }\}$ is not a context-free language

Solution:

1. Let L be Context free, Let w be a string in L
2. Let n be the natural number obtained by using the pumping lemma.

Pumping Lemma for Context Free Language

Ques: Show that $L = \{a^p | p \text{ is a prime }\}$ is not a context-free language

Solution:

1. Let L be Context free, Let w be a string in L
2. Let n be the natural number obtained by using the pumping lemma.
3. Let p be a prime number greater than n, Then $z = a^p \in L$. We can write $z = uvwxy$.

Pumping Lemma for Context Free Language

Ques: Show that $L = \{a^p | p \text{ is a prime } \}$ is not a context-free language

Solution:

1. Let L be Context free, Let w be a string in L
2. Let n be the natural number obtained by using the pumping lemma.
3. Let p be a prime number greater than n, Then $z = a^p \in L$. We can write $z = uvwxy$.
4. Prove for some k that $uv^kwx^ky \notin L$ means $|uv^kwx^ky|$ is not prime.

Pumping Lemma for Context Free Language

Ques: Show that $L = \{a^p | p \text{ is a prime }\}$ is not a context-free language

Solution:

1. Let L be Context free, Let w be a string in L
2. Let n be the natural number obtained by using the pumping lemma.
3. Let p be a prime number greater than n, Then $z = a^p \in L$. We can write $z = uvwxy$.
4. Prove for some k that $uv^kwx^ky \notin L$ means $|uv^kwx^ky|$ is not prime.
5. By pumping lemma, $uv^0wx^0y = uwy \in L$. So $uxy|$ is a prime number, say q.

Pumping Lemma for Context Free Language

Ques: Show that $L = \{a^p | p \text{ is a prime }\}$ is not a context-free language

Solution:

1. Let L be Context free, Let w be a string in L
2. Let n be the natural number obtained by using the pumping lemma.
3. Let p be a prime number greater than n, Then $z = a^p \in L$. We can write $z = uvwxy$.
4. Prove for some k that $uv^k wx^k y \notin L$ means $|uv^k wx^k y|$ is not prime.
5. By pumping lemma, $uv^0 wx^0 y = uwy \in L$. So $uxy|$ is a prime number, say q.
6. Let $|vx| = r$. Then, $|uv^q wx^q y| = q + qr$.

Pumping Lemma for Context Free Language

Ques: Show that $L = \{a^p | p \text{ is a prime }\}$ is not a context-free language

Solution:

1. Let L be Context free, Let w be a string in L
2. Let n be the natural number obtained by using the pumping lemma.
3. Let p be a prime number greater than n, Then $z = a^p \in L$. We can write $z = uvwxy$.
4. Prove for some k that $uv^kwx^ky \notin L$ means $|uv^kwx^ky|$ is not prime.
5. By pumping lemma, $uv^0wx^0y = uwy \in L$. So $uxy|$ is a prime number, say q.
6. Let $|vx| = r$. Then, $|uv^qwx^qy| = q + qr$.
7. As $q(1 + r)$ is not a prime, means $uv^qwx^qy \notin L$.

Pumping Lemma for Context Free Language

Ques: Show that $L = \{a^p | p \text{ is a prime } \}$ is not a context-free language

Solution:

1. Let L be Context free, Let w be a string in L
2. Let n be the natural number obtained by using the pumping lemma.
3. Let p be a prime number greater than n, Then $z = a^p \in L$. We can write $z = uvwxy$.
4. Prove for some k that $uv^kwx^ky \notin L$ means $|uv^kwx^ky|$ is not prime.
5. By pumping lemma, $uv^0wx^0y = uwy \in L$. So $uxy|$ is a prime number, say q.
6. Let $|vx| = r$. Then, $|uv^qwx^qy| = q + qr$.
7. As $q(1 + r)$ is not a prime, means $uv^qwx^qy \notin L$.
8. This is a contradiction. Therefore, L is not context-free.

Pumping Lemma for Context Free Language

- Show that following L are not Context free
  1. L$=\{ww : w \in \{a, b\}^*\}$
  2. L$=\{a^n b^j : n = j^2\}$
  3. L$= \{a^{n!} : n \geq 0\}$

Decision Algorithms

Some decision algorithms for context-free languages and regular sets.

1. Algorithm for deciding whether a context free language L is empty.
2. Algorithm for deciding whether a context-free language L is finite.
3. Algorithm for deciding whether a regular language L is empty.
4. Algorithm for deciding whether a regular language L is infinite.

Linear Grammar

- A grammar in which atmost one variable can occur on right side of any production without restriction on the size of this grammar, is known as Linear Grammar.
- Right Linear Grammar- A grammar $G = (V.T, P, S)$ is said to be right linear, if all the productions are of the form:
  $A \to xB$, $A \to x$
  where $A, B \in V, x \in T^*$
- Left Linear Grammar- A grammar is said to be left linear if all the productions are of the form:
  $A \to Bx$, $A \to x$
  where $A, B \in V, x \in T^*$
- Linear Grammar- A grammar is said to be linear grammar if all the productions are of the form:
  $A \to vBw$, $A \to w$
  where $A, B \in V; v, w \in T^*$

Linear Grammar

- A regular grammar is always linear but not all linear grammars are regular.
- A regular grammar is one that is either right linear or left linear
- In a regular grammar, atmost one variable appears on right side of any production. Further, that variable must consistently be either on rightmost or leftmost symbol of right side of any production.
- Example: $G_1 = (\{S\}, \{a, b\}, P_1, S)$ where P_1 given as $S \rightarrow abS|a$ is right linear.
- Example: $G_2 = (\{S, S_1, S_2\}, \{a, b\}, P_2, S)$ with productions $S \rightarrow S_1ab, S_1 \rightarrow S_1ab, S_1 \rightarrow S_2, S_2 \rightarrow a$ is left linear

Linear Grammar

- $G = (\{S, A, B\}, \{a, b\}, P, S)$ with productions $S \to A$, $A \to aB|\wedge$, $B \to Ab$ is not regular $\because$ even if each production is right linear or left linear, but grammar itself is neither right linear nor left linear $\therefore$ not regular

Linear Grammar

1. Construct a finite automata that accepts the language generated by grammar
   $V_0 \to aV_1,\ V_1 \to abV_0\ |$b
2. Construct a right linear grammar for L(aab^*a)
3. Convert the following regular expression into equivalent regular grammar
   - $(a+b)^*a$
   - a^*+b+b^*

Pushdown Automata

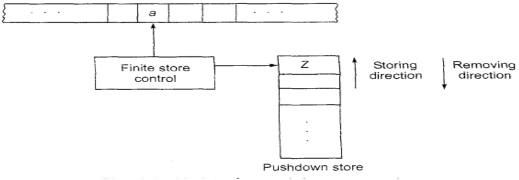


Figure 4: Model of Pushdown Automata

- A Non-deterministic Pushdown Automata is a 7-tuple $(Q, \Sigma, \tau, \delta, q_0, Z_0, F)$
  where , $Q=$ finite non-empty set of states
  $\Sigma=$ finite non-empty set of input symbols
  $\tau=$ finite non-empty set of pushdown symbols
  $q_0=$ initial state
  $Z_0=$ initial symbol on pushdown store
  $F=$ set of final states
  $\delta \implies Q \times (\Sigma \cup \{\wedge\}) \times \tau \rightarrow Q \times \tau^*$

Pushdown Automata

$\delta \implies Q \times (\Sigma \cup \{\wedge\}) \times \tau \to Q \times \tau^*$

- Each move of the control unit is determined by the current input symbol as well as by the symbol currently on the top of the stack.
- The result of the move is a new state of control unit and a change in the top of the stack.

Instantaneous Description (ID) Let $A = (Q, \Sigma, \tau, \delta, q_0, Z_0, F)$ be a pda. An instantaneous description (ID) is (q, w, α), where $q \in Q, w \in \Sigma^*$ and $\alpha \in \tau^*$.

- An initial ID is $({}_q o, w, Z_0)$. This means that initially the pda is in the initial state q_0, the input string to be processed is w and the PDS has only one symbol, namely Z_0.
- In an ID $(q, \wedge, Z)$, In this case the pda makes a $\wedge$-move.

Pushdown Automata

- A move relation, denoted by $\vdash$ between IDs is defined as

$$(q, a_1 a_2 \ldots a_n, Z_1 Z_2 \ldots Z_m) \vdash (q', a_2 \ldots a_n, \beta Z_2 \ldots Z_m)$$

  if $\delta(q, a_1, Z_1) = (q', \beta)$

- if $(q_1, x, \alpha) \vdash^* (q_2, \wedge, \beta)$ then for every $y \in \Sigma^*$, $(q_1, xy, \alpha) \vdash^* (q_2, y, \beta)$
- Conversely, if $(q_1, xy, \alpha) \vdash^* (q_2, y, \beta)$ for some $y \in \Sigma^*$, then $(q_1, x, \alpha) \vdash^* (q_2, \wedge, \beta)$
- if $(q_1, x, \alpha) \vdash^* (q_2, \wedge, \beta)$ then for every $\gamma \in \tau^*$, $(q_1, x, \alpha\gamma) \vdash^* (q_2, \wedge, \beta\gamma)$

NPDA: Example

- Consider a NPDA as under:
  $Q = \{q_0, q_1, q_2, q_3\}$, $\Sigma = \{a, b\}$
  $\tau = \{a, Z_0\}$, Z_0, $F = \{q_3\}$ and
  $\delta(q_0, a, Z_0) = (q_0, aZ_0)$
  $\delta(q_0, a, a) = (q_0, aa)$
  $\delta(q_0, b, a) = (q_1, \wedge)$
  $\delta(q_1, b, a) = (q_1, \wedge)$
  $\delta(q_1, \wedge, Z_0) = (q_f, Z_0)$
  What can we say about the action of this automaton?

Language accepted by a PDA

Acceptance of input strings by pda is of two way:

1. Acceptance by Final State
2. Acceptance by Null Store

Let M $=(Q, \Sigma, \tau, \delta, q_0, Z_0, F)$ be a non-deterministic push-down automata. The language accepted by M is the set
$L(M) = \{w \in \Sigma^* | (q_0, w, Z_0) \vdash_M^* (q', \Lambda, \alpha)\}$
where $q' \in F$ and $\alpha \in \tau^*$

Example: Construct a NPDA for the language
$L = \{w \in \{a, b\}^* | n_a(w) = n_b(w)\}$

Solution: $Q = \{q_0, q_f\}, \Sigma = \{a, b\}, \tau = \{a, b, Z\}, F = \{q_f\}$

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

$\delta(q_0, a, Z) = (q_0, aZ)$

$\delta(q_0, b, Z) = (q_0, bZ)$

$\delta(q_0, a, a) = (q_0, aa)$

Language accepted by a PDA

$\delta(q_0, b, b) = (q_0, bb)$
$\delta(q_0, a, b) = (q_0, \Lambda)$
$\delta(q_0, b, a) = (q_0, \Lambda)$
$\delta(q_0, \Lambda, Z) = (q_f, Z)$
Let us assume $w = baab$ to process
$(q_0, baab, Z) \vdash$
$(q_0, aab, bZ) \vdash$
$(q_0, ab, Z) \vdash$
$(q_0, b, aZ) \vdash$
$(q_0, \Lambda, Z) \vdash$
(q_f, Λ, Z)

Language accepted by PDA

Let A $=(Q, \Sigma, \tau, \delta, q_0, Z_0, F)$ be a non-deterministic push-down automata. The language accepted by null store or empty store A is the set $N(A) = \{w \in \Sigma^* | (q_0, w, Z_0) \vdash_A^* (q, \Lambda, \Lambda)\}$
where $q \in Q$

Theorem

If $A = (Q, \Sigma, \tau, \delta, q_0, Z_0, F)$ is a pda accepting L by empty store. we can find a pda $B = (Q', \Sigma, \tau', \delta', q_0', Z_0, F')$ accepting L by final state: i.e. $L = N(A) = T(B)$.

Theorem

If $A = (Q, \Sigma, \tau, \delta, q_0, Z_0, F)$ accepts L by final state, we can find a pda B accepting L by empty store: i.e. $L = T(A) = N(B)$.

Language accepted by PDA

Question: Construct a NPDA for accepting the language
L=$\{ww^R$: w $\in \{a, b\}^+ \}$

Language accepted by PDA

Question: Construct a NPDA for accepting the language
L=$\{ww^R$: w $\in \{a, b\}^+$ }
Solution: M=(Q,Σ, τ, δ, q_0, z, F)
where Q= $\{q_0, q_1, q_2\}$, Σ={a,b}
τ= {a,b,z}, F=$\{q_2\}$

Language accepted by PDA

Question: Construct a NPDA for accepting the language
L=$\{ww^R$: w $\in \{a, b\}^+$ }
Solution: M=(Q,Σ, τ, δ, q_0, z, F)
where Q= $\{q_0, q_1, q_2\}$, Σ={a,b}
τ= {a,b,z}, F=$\{q_2\}$
$\delta(q_0, a, a) = (q_0, aa)$,

Language accepted by PDA

Question: Construct a NPDA for accepting the language
L=$\{ww^R$: w $\in \{a, b\}^+ \}$
Solution: M=(Q,Σ, τ, δ, q_0, z, F)
where Q= $\{q_0, q_1, q_2\}$, Σ={a,b}
τ= {a,b,z}, F=$\{q_2\}$
$\delta(q_0, a, a) = (q_0, aa)$, $\delta(q_0, b, a) = (q_0, ba)$

Language accepted by PDA

Question: Construct a NPDA for accepting the language

L=$\{ww^R$: w $\in \{a, b\}^+$ }

Solution: M=(Q,Σ, τ, δ, q_0, z, F)

where Q= $\{q_0, q_1, q_2\}$, Σ={a,b}

τ= {a,b,z}, F=$\{q_2\}$

$\delta(q_0, a, a) = (q_0, aa)$, $\delta(q_0, b, a) = (q_0, ba)$

$\delta(q_0, a, b) = (q_0, ab)$,

Language accepted by PDA

Question: Construct a NPDA for accepting the language

L=$\{ww^R$: w $\in \{a, b\}^+ \}$

Solution: M=(Q,Σ, τ, δ, q_0, z, F)

where Q= $\{q_0, q_1, q_2\}$, Σ={a,b}

τ= {a,b,z}, F=$\{q_2\}$

$\delta(q_0, a, a) = (q_0, aa)$, $\delta(q_0, b, a) = (q_0, ba)$

$\delta(q_0, a, b) = (q_0, ab)$, $\delta(q_0, b, b) = (q_0, bb)$

Language accepted by PDA

Question: Construct a NPDA for accepting the language
L=$\{ww^R$: w $\in \{a, b\}^+$ }

Solution: M=(Q,Σ, τ, δ, q_0, z, F)

where Q= $\{q_0, q_1, q_2\}$, Σ={a,b}

τ= {a,b,z}, F=$\{q_2\}$

$\delta(q_0, a, a) = (q_0, aa)$, $\delta(q_0, b, a) = (q_0, ba)$

$\delta(q_0, a, b) = (q_0, ab)$, $\delta(q_0, b, b) = (q_0, bb)$

$\delta(q_0, a, z) = (q_0, az)$,

Language accepted by PDA

Question: Construct a NPDA for accepting the language
L=$\{ww^R$: w $\in \{a, b\}^+ \}$

Solution: M=(Q,Σ, τ, δ, q_0, z, F)
where Q= $\{q_0, q_1, q_2\}$, Σ={a,b}
τ= {a,b,z}, F=$\{q_2\}$
$\delta(q_0, a, a) = (q_0, aa)$, $\delta(q_0, b, a) = (q_0, ba)$
$\delta(q_0, a, b) = (q_0, ab)$, $\delta(q_0, b, b) = (q_0, bb)$
$\delta(q_0, a, z) = (q_0, az)$, $\delta(q_0, b, z) = (q_0, bz)$

Language accepted by PDA

Question: Construct a NPDA for accepting the language
L=$\{ww^R$: w $\in \{a, b\}^+$ $\}$

Solution: M=(Q,Σ, τ, δ, q_0, z, F)
where Q= $\{q_0, q_1, q_2\}$, Σ={a,b}
τ= {a,b,z}, F=$\{q_2\}$

$\delta(q_0, a, a) = (q_0, aa)$, $\delta(q_0, b, a) = (q_0, ba)$
$\delta(q_0, a, b) = (q_0, ab)$, $\delta(q_0, b, b) = (q_0, bb)$
$\delta(q_0, a, z) = (q_0, az)$, $\delta(q_0, b, z) = (q_0, bz)$

For middle:

Language accepted by PDA

Question: Construct a NPDA for accepting the language
L=$\{ww^R$: w $\in \{a, b\}^+$ }
Solution: M=(Q,Σ, τ, δ, q_0, z, F)
where Q= $\{q_0, q_1, q_2\}$, Σ={a,b}
τ= {a,b,z}, F=$\{q_2\}$
$\delta(q_0, a, a) = (q_0, aa)$, $\delta(q_0, b, a) = (q_0, ba)$
$\delta(q_0, a, b) = (q_0, ab)$, $\delta(q_0, b, b) = (q_0, bb)$
$\delta(q_0, a, z) = (q_0, az)$, $\delta(q_0, b, z) = (q_0, bz)$
For middle:
$\delta(q_0, \Lambda, a) = (q_1, a)$,

Language accepted by PDA

Question: Construct a NPDA for accepting the language
L=$\{ww^R$: w $\in \{a, b\}^+$ $\}$
Solution: M=(Q,Σ, τ, δ, q_0, z, F)
where Q= $\{q_0, q_1, q_2\}$, Σ={a,b}
τ= {a,b,z}, F=$\{q_2\}$
$\delta(q_0, a, a) = (q_0, aa)$, $\delta(q_0, b, a) = (q_0, ba)$
$\delta(q_0, a, b) = (q_0, ab)$, $\delta(q_0, b, b) = (q_0, bb)$
$\delta(q_0, a, z) = (q_0, az)$, $\delta(q_0, b, z) = (q_0, bz)$
For middle:
$\delta(q_0, \Lambda, a) = (q_1, a)$, $\delta(q_0, \Lambda, b) = (q_1, b)$

Language accepted by PDA

Question: Construct a NPDA for accepting the language

L=$\{ww^R$: w $\in \{a, b\}^+ \}$

Solution: M=(Q,Σ, τ, δ, q_0, z, F)

where Q= $\{q_0, q_1, q_2\}$, Σ={a,b}

τ= {a,b,z}, F=$\{q_2\}$

$\delta(q_0, a, a) = (q_0, aa)$, $\delta(q_0, b, a) = (q_0, ba)$

$\delta(q_0, a, b) = (q_0, ab)$, $\delta(q_0, b, b) = (q_0, bb)$

$\delta(q_0, a, z) = (q_0, az)$, $\delta(q_0, b, z) = (q_0, bz)$

For middle:

$\delta(q_0, \Lambda, a) = (q_1, a)$, $\delta(q_0, \Lambda, b) = (q_1, b)$

For matching w^R against contents of stack

Language accepted by PDA

Question: Construct a NPDA for accepting the language

L=$\{ww^R$: w $\in \{a, b\}^+ \}$

Solution: M=(Q,Σ, τ, δ, q_0, z, F)

where Q= $\{q_0, q_1, q_2\}$, Σ={a,b}

τ= {a,b,z}, F=$\{q_2\}$

$\delta(q_0, a, a) = (q_0, aa)$, $\delta(q_0, b, a) = (q_0, ba)$

$\delta(q_0, a, b) = (q_0, ab)$, $\delta(q_0, b, b) = (q_0, bb)$

$\delta(q_0, a, z) = (q_0, az)$, $\delta(q_0, b, z) = (q_0, bz)$

For middle:

$\delta(q_0, \Lambda, a) = (q_1, a)$, $\delta(q_0, \Lambda, b) = (q_1, b)$

For matching w^R against contents of stack

$\delta(q_1, a, a) = (q_1, \Lambda)$,

Language accepted by PDA

Question: Construct a NPDA for accepting the language
L=$\{ww^R$: w $\in \{a, b\}^+ \}$
Solution: M=(Q,Σ, τ, δ, q_0, z, F)
where Q= $\{q_0, q_1, q_2\}$, Σ={a,b}
τ= {a,b,z}, F=$\{q_2\}$
$\delta(q_0, a, a) = (q_0, aa)$, $\delta(q_0, b, a) = (q_0, ba)$
$\delta(q_0, a, b) = (q_0, ab)$, $\delta(q_0, b, b) = (q_0, bb)$
$\delta(q_0, a, z) = (q_0, az)$, $\delta(q_0, b, z) = (q_0, bz)$
For middle:
$\delta(q_0, \Lambda, a) = (q_1, a)$, $\delta(q_0, \Lambda, b) = (q_1, b)$
For matching w^R against contents of stack
$\delta(q_1, a, a) = (q_1, \Lambda)$, $\delta(q_1, b, b) = (q_1, \Lambda)$

Language accepted by PDA

Question: Construct a NPDA for accepting the language

L={ww^R: w $\in \{a, b\}^+$ }

Solution: M=(Q,Σ, τ, δ, q_0, z, F)

where Q= $\{q_0, q_1, q_2\}$, Σ={a,b}

τ= {a,b,z}, F={q_2}

$\delta(q_0, a, a) = (q_0, aa)$, $\delta(q_0, b, a) = (q_0, ba)$

$\delta(q_0, a, b) = (q_0, ab)$, $\delta(q_0, b, b) = (q_0, bb)$

$\delta(q_0, a, z) = (q_0, az)$, $\delta(q_0, b, z) = (q_0, bz)$

For middle:

$\delta(q_0, \Lambda, a) = (q_1, a)$, $\delta(q_0, \Lambda, b) = (q_1, b)$

For matching w^R against contents of stack

$\delta(q_1, a, a) = (q_1, \Lambda)$, $\delta(q_1, b, b) = (q_1, \Lambda)$

Finally, $\delta(q_1, \Lambda, z) = (q_2, z)$

Language accepted by PDA

Question: Construct a NPDA for accepting the language
L=$\{ww^R$: w $\in \{a, b\}^+ \}$
Solution: M=(Q,Σ, τ, δ, q_0, z, F)
where Q= $\{q_0, q_1, q_2\}$, Σ={a,b}
τ= {a,b,z}, F=$\{q_2\}$
$\delta(q_0, a, a) = (q_0, aa)$, $\delta(q_0, b, a) = (q_0, ba)$
$\delta(q_0, a, b) = (q_0, ab)$, $\delta(q_0, b, b) = (q_0, bb)$
$\delta(q_0, a, z) = (q_0, az)$, $\delta(q_0, b, z) = (q_0, bz)$
For middle:
$\delta(q_0, \Lambda, a) = (q_1, a)$, $\delta(q_0, \Lambda, b) = (q_1, b)$
For matching w^R against contents of stack
$\delta(q_1, a, a) = (q_1, \Lambda)$, $\delta(q_1, b, b) = (q_1, \Lambda)$
Finally, $\delta(q_1, \Lambda, z) = (q_2, z)$
Let the String assumed be 'abba':
$(q_0, abba, z)$

Language accepted by PDA

Question: Construct a NPDA for accepting the language

L=$\{ww^R$: w $\in \{a,b\}^+ \}$

Solution: M=(Q,Σ, τ, δ, q_0, z, F)

where Q= $\{q_0, q_1, q_2\}$, Σ={a,b}

τ= {a,b,z}, F=$\{q_2\}$

$\delta(q_0, a, a) = (q_0, aa)$, $\delta(q_0, b, a) = (q_0, ba)$

$\delta(q_0, a, b) = (q_0, ab)$, $\delta(q_0, b, b) = (q_0, bb)$

$\delta(q_0, a, z) = (q_0, az)$, $\delta(q_0, b, z) = (q_0, bz)$

For middle:

$\delta(q_0, \Lambda, a) = (q_1, a)$, $\delta(q_0, \Lambda, b) = (q_1, b)$

For matching w^R against contents of stack

$\delta(q_1, a, a) = (q_1, \Lambda)$, $\delta(q_1, b, b) = (q_1, \Lambda)$

Finally, $\delta(q_1, \Lambda, z) = (q_2, z)$

Let the String assumed be 'abba':

$(q_0, abba, z) \vdash (q_0, bba, az)$

Language accepted by PDA

Question: Construct a NPDA for accepting the language
L=$\{ww^R$: w $\in \{a, b\}^+ \}$
Solution: M=(Q,Σ, τ, δ, q_0, z, F)
where Q= $\{q_0, q_1, q_2\}$, Σ=$\{$a,b$\}$
τ= $\{$a,b,z$\}$, F=$\{q_2\}$
$\delta(q_0, a, a) = (q_0, aa)$, $\delta(q_0, b, a) = (q_0, ba)$
$\delta(q_0, a, b) = (q_0, ab)$, $\delta(q_0, b, b) = (q_0, bb)$
$\delta(q_0, a, z) = (q_0, az)$, $\delta(q_0, b, z) = (q_0, bz)$
For middle:
$\delta(q_0, \Lambda, a) = (q_1, a)$, $\delta(q_0, \Lambda, b) = (q_1, b)$
For matching w^R against contents of stack
$\delta(q_1, a, a) = (q_1, \Lambda)$, $\delta(q_1, b, b) = (q_1, \Lambda)$
Finally, $\delta(q_1, \Lambda, z) = (q_2, z)$
Let the String assumed be 'abba':
$(q_0, abba, z) \vdash (q_0, bba, az) \vdash (q_0, ba, baz)$

Language accepted by PDA

Question: Construct a NPDA for accepting the language

L={ww^R: w $\in \{a, b\}^+$ }

Solution: M=(Q,Σ, τ, δ, q_0, z, F)

where Q= $\{q_0, q_1, q_2\}$, Σ={a,b}

τ= {a,b,z}, F={q_2}

$\delta(q_0, a, a) = (q_0, aa)$, $\delta(q_0, b, a) = (q_0, ba)$

$\delta(q_0, a, b) = (q_0, ab)$, $\delta(q_0, b, b) = (q_0, bb)$

$\delta(q_0, a, z) = (q_0, az)$, $\delta(q_0, b, z) = (q_0, bz)$

For middle:

$\delta(q_0, \Lambda, a) = (q_1, a)$, $\delta(q_0, \Lambda, b) = (q_1, b)$

For matching w^R against contents of stack

$\delta(q_1, a, a) = (q_1, \Lambda)$, $\delta(q_1, b, b) = (q_1, \Lambda)$

Finally, $\delta(q_1, \Lambda, z) = (q_2, z)$

Let the String assumed be 'abba':

$(q_0, abba, z) \vdash (q_0, bba, az) \vdash (q_0, ba, baz) \vdash (q_1, ba, baz)$

Language accepted by PDA

Question: Construct a NPDA for accepting the language
L=$\{ww^R$: w $\in \{a, b\}^+$ }
Solution: M=(Q,Σ, τ, δ, q_0, z, F)
where Q= $\{q_0, q_1, q_2\}$, Σ={a,b}
τ= {a,b,z}, F=$\{q_2\}$
$\delta(q_0, a, a) = (q_0, aa)$, $\delta(q_0, b, a) = (q_0, ba)$
$\delta(q_0, a, b) = (q_0, ab)$, $\delta(q_0, b, b) = (q_0, bb)$
$\delta(q_0, a, z) = (q_0, az)$, $\delta(q_0, b, z) = (q_0, bz)$
For middle:
$\delta(q_0, \Lambda, a) = (q_1, a)$, $\delta(q_0, \Lambda, b) = (q_1, b)$
For matching w^R against contents of stack
$\delta(q_1, a, a) = (q_1, \Lambda)$, $\delta(q_1, b, b) = (q_1, \Lambda)$
Finally, $\delta(q_1, \Lambda, z) = (q_2, z)$
Let the String assumed be 'abba':
$(q_0, abba, z) \vdash (q_0, bba, az) \vdash (q_0, ba, baz) \vdash (q_1, ba, baz)$
$\vdash (q_1, a, az)$

Language accepted by PDA

Question: Construct a NPDA for accepting the language
L=$\{ww^R$: w $\in \{a, b\}^+$ $\}$

Solution: M=(Q,Σ, τ, δ, q_0, z, F)
where Q= $\{q_0, q_1, q_2\}$, Σ={a,b}
τ= {a,b,z}, F=$\{q_2\}$
$\delta(q_0, a, a) = (q_0, aa)$, $\delta(q_0, b, a) = (q_0, ba)$
$\delta(q_0, a, b) = (q_0, ab)$, $\delta(q_0, b, b) = (q_0, bb)$
$\delta(q_0, a, z) = (q_0, az)$, $\delta(q_0, b, z) = (q_0, bz)$
For middle:
$\delta(q_0, \Lambda, a) = (q_1, a)$, $\delta(q_0, \Lambda, b) = (q_1, b)$
For matching w^R against contents of stack
$\delta(q_1, a, a) = (q_1, \Lambda)$, $\delta(q_1, b, b) = (q_1, \Lambda)$
Finally, $\delta(q_1, \Lambda, z) = (q_2, z)$
Let the String assumed be 'abba':
$(q_0, abba, z) \vdash (q_0, bba, az) \vdash (q_0, ba, baz) \vdash (q_1, ba, baz)$
$\vdash (q_1, a, az) \vdash (q_1, \Lambda, z)$

Language accepted by PDA

Question: Construct a NPDA for accepting the language

L=$\{ww^R$: w $\in \{a, b\}^+ \}$

Solution: M=(Q,Σ, τ, δ, q_0, z, F)

where Q= $\{q_0, q_1, q_2\}$, Σ={a,b}

τ= {a,b,z}, F=$\{q_2\}$

$\delta(q_0, a, a) = (q_0, aa)$, $\delta(q_0, b, a) = (q_0, ba)$

$\delta(q_0, a, b) = (q_0, ab)$, $\delta(q_0, b, b) = (q_0, bb)$

$\delta(q_0, a, z) = (q_0, az)$, $\delta(q_0, b, z) = (q_0, bz)$

For middle:

$\delta(q_0, \Lambda, a) = (q_1, a)$, $\delta(q_0, \Lambda, b) = (q_1, b)$

For matching w^R against contents of stack

$\delta(q_1, a, a) = (q_1, \Lambda)$, $\delta(q_1, b, b) = (q_1, \Lambda)$

Finally, $\delta(q_1, \Lambda, z) = (q_2, z)$

Let the String assumed be 'abba':

$(q_0, abba, z) \vdash (q_0, bba, az) \vdash (q_0, ba, baz) \vdash (q_1, ba, baz)$

$\vdash (q_1, a, az) \vdash (q_1, \Lambda, z) \vdash (q_2, z)$

Language accepted by a PDA

1. Construct a NPDA that accepts the language
   L=$\{a^n b^m$: n$\geq$0, n$\neq$m$\}$
2. Find NPDA on Σ=$\{$a,b,c$\}$ that accepts the language
   L=$\{w_1 c w_2$: $w_1, w_2 \in \{a, b\}^*$, $w_1 \neq {w_2}^R\}$

CFG to PDA

Theorem

If L is a context-free language, then we can construct a pda A accepting L by empty store, i.e. $L = N(A)$.

Construction of pda A Let $L = L(G)$, where $G = (V, \Sigma, P, S)$ is a context-free grammar. We construct a pda A as $A = (\{q\}, \Sigma, V \cup \Sigma, \delta, q, S, \phi)$, where δ is defined by the following rules:

R_1 For every $A \to \alpha$ in P, $\delta(q, \wedge, A) = \{(q, \alpha)\}$

R_2 For every a in Σ, $\delta(q, a, a) = \{(q, \wedge)\}$

CFG to PDA

- Construct a PDA that accepts the language generated by the grammar with productions
  $S \rightarrow aSA|a,\ A \rightarrow bB,\ B \rightarrow b$
  Solution: Step-1 The given productions are:
  $S \rightarrow aSA|a$
  $A \rightarrow bB$
  $B \rightarrow b$
  δ is defined by the following rules:
  S-productions
  $\delta(q, \wedge, S) = \{(q, aSA), (q, a)\}$
  A-productions
  $\delta(q, \wedge, A) = \{(q, bB)\}$
  B-productions
  $\delta(q, \wedge, B) = \{(q, b)\}$
  Productions for Σ

CFG to PDA

$\delta(q, a, a) = \{(q, \wedge)\}$
$\delta(q, b, b) = \{(q, \wedge)\}$
Appearance of Λ on top of stack implies completion of derivation and PDA is put to final state by
$\delta(q, \Lambda, Z) = (q_f, \Lambda)$

CFG to PDA

Question: Consider the grammar

S $\rightarrow$ aA, A $\rightarrow$aABC |bB |a, B $\rightarrow$b, C $\rightarrow$c

CFG to PDA

Question: Consider the grammar

S $\rightarrow$ aA, A $\rightarrow$aABC |bB |a, B $\rightarrow$b, C $\rightarrow$c

Solution: Putting Start Symbol on stack

$\delta(q_0, \Lambda, Z) = (q_1, SZ)$

CFG to PDA

Question: Consider the grammar

S $\rightarrow$ aA, A $\rightarrow$aABC |bB |a, B $\rightarrow$b, C $\rightarrow$c

Solution: Putting Start Symbol on stack

$\delta(q_0, \Lambda, Z) = (q_1, SZ)$

Final Production: $\delta(q_1, \Lambda, Z) = (q_f, Z)$

CFG to PDA

Question: Consider the grammar

S $\rightarrow$ aA, A $\rightarrow$aABC |bB |a, B $\rightarrow$b, C $\rightarrow$c

Solution: Putting Start Symbol on stack

$\delta(q_0, \Lambda, Z) = (q_1, SZ)$

Final Production: $\delta(q_1, \Lambda, Z) = (q_f, Z)$

Now, according to the productions

CFG to PDA

Question: Consider the grammar

S $\rightarrow$ aA, A $\rightarrow$aABC |bB |a, B $\rightarrow$b, C $\rightarrow$c

Solution: Putting Start Symbol on stack

$\delta(q_0, \Lambda, Z) = (q_1, SZ)$

Final Production: $\delta(q_1, \Lambda, Z) = (q_f, Z)$

Now, according to the productions

$\delta(q_1, a, S) = (q_1, A)$,

CFG to PDA

Question: Consider the grammar

S $\rightarrow$ aA, A $\rightarrow$aABC |bB |a, B $\rightarrow$b, C $\rightarrow$c

Solution: Putting Start Symbol on stack

$\delta(q_0, \Lambda, Z) = (q_1, SZ)$

Final Production: $\delta(q_1, \Lambda, Z) = (q_f, Z)$

Now, according to the productions

$\delta(q_1, a, S) = (q_1, A)$, $\delta(q_1, a, A) = (q_1, ABC)$,

CFG to PDA

Question: Consider the grammar

S $\rightarrow$ aA, A $\rightarrow$aABC |bB |a, B $\rightarrow$b, C $\rightarrow$c

Solution: Putting Start Symbol on stack

$\delta(q_0, \Lambda, Z) = (q_1, SZ)$

Final Production: $\delta(q_1, \Lambda, Z) = (q_f, Z)$

Now, according to the productions

$\delta(q_1, a, S) = (q_1, A)$, $\delta(q_1, a, A) = (q_1, ABC)$,

$\delta(q_1, b, A) = (q_1, B)$,

CFG to PDA

Question: Consider the grammar

S $\rightarrow$ aA, A $\rightarrow$aABC |bB |a, B $\rightarrow$b, C $\rightarrow$c

Solution: Putting Start Symbol on stack

$\delta(q_0, \Lambda, Z) = (q_1, SZ)$

Final Production: $\delta(q_1, \Lambda, Z) = (q_f, Z)$

Now, according to the productions

$\delta(q_1, a, S) = (q_1, A)$, $\delta(q_1, a, A) = (q_1, ABC)$,

$\delta(q_1, b, A) = (q_1, B)$, $\delta(q_1, a, A) = (q_1, \Lambda)$,

CFG to PDA

Question: Consider the grammar

S $\rightarrow$ aA, A $\rightarrow$aABC |bB |a, B $\rightarrow$b, C $\rightarrow$c

Solution: Putting Start Symbol on stack

$\delta(q_0, \Lambda, Z) = (q_1, SZ)$

Final Production: $\delta(q_1, \Lambda, Z) = (q_f, Z)$

Now, according to the productions

$\delta(q_1, a, S) = (q_1, A)$, $\delta(q_1, a, A) = (q_1, ABC)$,

$\delta(q_1, b, A) = (q_1, B)$, $\delta(q_1, a, A) = (q_1, \Lambda)$, $\delta(q_1, b, B) = (q_1, \Lambda)$,

CFG to PDA

Question: Consider the grammar

S $\rightarrow$ aA, A $\rightarrow$aABC |bB |a, B $\rightarrow$b, C $\rightarrow$c

Solution: Putting Start Symbol on stack

$\delta(q_0, \Lambda, Z) = (q_1, SZ)$

Final Production: $\delta(q_1, \Lambda, Z) = (q_f, Z)$

Now, according to the productions

$\delta(q_1, a, S) = (q_1, A)$, $\delta(q_1, a, A) = (q_1, ABC)$,

$\delta(q_1, b, A) = (q_1, B)$, $\delta(q_1, a, A) = (q_1, \Lambda)$, $\delta(q_1, b, B) = (q_1, \Lambda)$,

$\delta(q_1, c, C) = (q_1, \Lambda)$

CFG to PDA

Question: Consider the grammar

S → aA, A →aABC |bB |a, B →b, C →c

Solution: Putting Start Symbol on stack

$\delta(q_0, \Lambda, Z) = (q_1, SZ)$

Final Production: $\delta(q_1, \Lambda, Z) = (q_f, Z)$

Now, according to the productions

$\delta(q_1, a, S) = (q_1, A)$, $\delta(q_1, a, A) = (q_1, ABC)$,

$\delta(q_1, b, A) = (q_1, B)$, $\delta(q_1, a, A) = (q_1, \Lambda)$, $\delta(q_1, b, B) = (q_1, \Lambda)$,

$\delta(q_1, c, C) = (q_1, \Lambda)$

Let the derivation be

S → aA → a**aAB**C → aa**a**BC → aaa**b**C → aaab**c**

CFG to PDA

Question: Consider the grammar

S $\rightarrow$ aA, A $\rightarrow$aABC |bB |a, B $\rightarrow$b, C $\rightarrow$c

Solution: Putting Start Symbol on stack

$\delta(q_0, \Lambda, Z) = (q_1, SZ)$

Final Production: $\delta(q_1, \Lambda, Z) = (q_f, Z)$

Now, according to the productions

$\delta(q_1, a, S) = (q_1, A)$, $\delta(q_1, a, A) = (q_1, ABC)$,
$\delta(q_1, b, A) = (q_1, B)$, $\delta(q_1, a, A) = (q_1, \Lambda)$, $\delta(q_1, b, B) = (q_1, \Lambda)$,
$\delta(q_1, c, C) = (q_1, \Lambda)$

Let the derivation be

S $\rightarrow$ aA $\rightarrow$ a**aAB**C $\rightarrow$ aa**a**BC $\rightarrow$ aaa**b**C $\rightarrow$ aaab**c**

Therefore, the sequence of moves by M for the processing of "aaabc" is:

CFG to PDA

Question: Consider the grammar

S → aA, A →aABC |bB |a, B →b, C →c

Solution: Putting Start Symbol on stack

$\delta(q_0, \Lambda, Z) = (q_1, SZ)$

Final Production: $\delta(q_1, \Lambda, Z) = (q_f, Z)$

Now, according to the productions

$\delta(q_1, a, S) = (q_1, A)$, $\delta(q_1, a, A) = (q_1, ABC)$,

$\delta(q_1, b, A) = (q_1, B)$, $\delta(q_1, a, A) = (q_1, \Lambda)$, $\delta(q_1, b, B) = (q_1, \Lambda)$,

$\delta(q_1, c, C) = (q_1, \Lambda)$

Let the derivation be

S → aA → a**aAB**C → aa**a**BC → aaa**b**C → aaab**c**

Therefore, the sequence of moves by M for the processing of "aaabc" is:

$(q_0, aaabc, Z)$

CFG to PDA

Question: Consider the grammar

S → aA, A →aABC |bB |a, B →b, C →c

Solution: Putting Start Symbol on stack

$\delta(q_0, \Lambda, Z) = (q_1, SZ)$

Final Production: $\delta(q_1, \Lambda, Z) = (q_f, Z)$

Now, according to the productions

$\delta(q_1, a, S) = (q_1, A)$, $\delta(q_1, a, A) = (q_1, ABC)$,
$\delta(q_1, b, A) = (q_1, B)$, $\delta(q_1, a, A) = (q_1, \Lambda)$, $\delta(q_1, b, B) = (q_1, \Lambda)$,
$\delta(q_1, c, C) = (q_1, \Lambda)$

Let the derivation be

S → aA → a**aAB**C → aa**a**BC → aaa**b**C → aaab**c**

Therefore, the sequence of moves by M for the processing of "aaabc" is:

$(q_0, aaabc, Z) \vdash (q_1, aaabc, SZ)$

CFG to PDA

Question: Consider the grammar

S → aA, A →aABC |bB |a, B →b, C →c

Solution: Putting Start Symbol on stack

$\delta(q_0, \Lambda, Z) = (q_1, SZ)$

Final Production: $\delta(q_1, \Lambda, Z) = (q_f, Z)$

Now, according to the productions

$\delta(q_1, a, S) = (q_1, A)$, $\delta(q_1, a, A) = (q_1, ABC)$,

$\delta(q_1, b, A) = (q_1, B)$, $\delta(q_1, a, A) = (q_1, \Lambda)$, $\delta(q_1, b, B) = (q_1, \Lambda)$,

$\delta(q_1, c, C) = (q_1, \Lambda)$

Let the derivation be

S → aA → a**aAB**C → aa**a**BC → aaa**b**C → aaab**c**

Therefore, the sequence of moves by M for the processing of "aaabc" is:

$(q_0, aaabc, Z) \vdash (q_1, aaabc, SZ) \vdash (q_1, aabc, AZ)$

CFG to PDA

Question: Consider the grammar

S $\rightarrow$ aA, A $\rightarrow$aABC |bB |a, B $\rightarrow$b, C $\rightarrow$c

Solution: Putting Start Symbol on stack

$\delta(q_0, \Lambda, Z) = (q_1, SZ)$

Final Production: $\delta(q_1, \Lambda, Z) = (q_f, Z)$

Now, according to the productions

$\delta(q_1, a, S) = (q_1, A)$, $\delta(q_1, a, A) = (q_1, ABC)$,
$\delta(q_1, b, A) = (q_1, B)$, $\delta(q_1, a, A) = (q_1, \Lambda)$, $\delta(q_1, b, B) = (q_1, \Lambda)$,
$\delta(q_1, c, C) = (q_1, \Lambda)$

Let the derivation be

S $\rightarrow$ aA $\rightarrow$ a**aAB**C $\rightarrow$ aa**a**BC $\rightarrow$ aaa**b**C $\rightarrow$ aaab**c**

Therefore, the sequence of moves by M for the processing of "aaabc" is:

$(q_0, aaabc, Z) \vdash (q_1, aaabc, SZ) \vdash (q_1, aabc, AZ)$
$\vdash (q_1, abc, ABCZ)$

CFG to PDA

Question: Consider the grammar

S $\rightarrow$ aA, A $\rightarrow$aABC |bB |a, B $\rightarrow$b, C $\rightarrow$c

Solution: Putting Start Symbol on stack

$\delta(q_0, \Lambda, Z) = (q_1, SZ)$

Final Production: $\delta(q_1, \Lambda, Z) = (q_f, Z)$

Now, according to the productions

$\delta(q_1, a, S) = (q_1, A)$, $\delta(q_1, a, A) = (q_1, ABC)$,

$\delta(q_1, b, A) = (q_1, B)$, $\delta(q_1, a, A) = (q_1, \Lambda)$, $\delta(q_1, b, B) = (q_1, \Lambda)$,

$\delta(q_1, c, C) = (q_1, \Lambda)$

Let the derivation be

S $\rightarrow$ aA $\rightarrow$ a**aAB**C $\rightarrow$ aa**a**BC $\rightarrow$ aaa**b**C $\rightarrow$ aaab**c**

Therefore, the sequence of moves by M for the processing of "aaabc" is:

$(q_0, aaabc, Z) \vdash (q_1, aaabc, SZ) \vdash (q_1, aabc, AZ)$

$\vdash (q_1, abc, ABCZ) \vdash (q_1, bc, BCZ)$

CFG to PDA

Question: Consider the grammar

S → aA, A →aABC |bB |a, B →b, C →c

Solution: Putting Start Symbol on stack

$\delta(q_0, \Lambda, Z) = (q_1, SZ)$

Final Production: $\delta(q_1, \Lambda, Z) = (q_f, Z)$

Now, according to the productions

$\delta(q_1, a, S) = (q_1, A)$, $\delta(q_1, a, A) = (q_1, ABC)$,
$\delta(q_1, b, A) = (q_1, B)$, $\delta(q_1, a, A) = (q_1, \Lambda)$, $\delta(q_1, b, B) = (q_1, \Lambda)$,
$\delta(q_1, c, C) = (q_1, \Lambda)$

Let the derivation be

S → aA → a**aAB**C → aa**a**BC → aaa**b**C → aaab**c**

Therefore, the sequence of moves by M for the processing of "aaabc" is:

$(q_0, aaabc, Z) \vdash (q_1, aaabc, SZ) \vdash (q_1, aabc, AZ)$
$\vdash (q_1, abc, ABCZ) \vdash (q_1, bc, BCZ) \vdash (q_1, c, CZ)$

CFG to PDA

Question: Consider the grammar

S $\rightarrow$ aA, A $\rightarrow$aABC |bB |a, B $\rightarrow$b, C $\rightarrow$c

Solution: Putting Start Symbol on stack

$\delta(q_0, \Lambda, Z) = (q_1, SZ)$

Final Production: $\delta(q_1, \Lambda, Z) = (q_f, Z)$

Now, according to the productions

$\delta(q_1, a, S) = (q_1, A)$, $\delta(q_1, a, A) = (q_1, ABC)$,

$\delta(q_1, b, A) = (q_1, B)$, $\delta(q_1, a, A) = (q_1, \Lambda)$, $\delta(q_1, b, B) = (q_1, \Lambda)$,

$\delta(q_1, c, C) = (q_1, \Lambda)$

Let the derivation be

S $\rightarrow$ aA $\rightarrow$ a**aAB**C $\rightarrow$ aa**a**BC $\rightarrow$ aaa**b**C $\rightarrow$ aaab**c**

Therefore, the sequence of moves by M for the processing of "aaabc" is:

$(q_0, aaabc, Z) \vdash (q_1, aaabc, SZ) \vdash (q_1, aabc, AZ)$

$\vdash (q_1, abc, ABCZ) \vdash (q_1, bc, BCZ) \vdash (q_1, c, CZ) \vdash (q_1, \Lambda, Z)$

CFG to PDA

Question: Consider the grammar

S $\rightarrow$ aA, A $\rightarrow$aABC |bB |a, B $\rightarrow$b, C $\rightarrow$c

Solution: Putting Start Symbol on stack

$\delta(q_0, \Lambda, Z) = (q_1, SZ)$

Final Production: $\delta(q_1, \Lambda, Z) = (q_f, Z)$

Now, according to the productions

$\delta(q_1, a, S) = (q_1, A)$, $\delta(q_1, a, A) = (q_1, ABC)$,

$\delta(q_1, b, A) = (q_1, B)$, $\delta(q_1, a, A) = (q_1, \Lambda)$, $\delta(q_1, b, B) = (q_1, \Lambda)$,

$\delta(q_1, c, C) = (q_1, \Lambda)$

Let the derivation be

S $\rightarrow$ aA $\rightarrow$ a**aAB**C $\rightarrow$ aa**a**BC $\rightarrow$ aaa**b**C $\rightarrow$ aaab**c**

Therefore, the sequence of moves by M for the processing of "aaabc" is:

$(q_0, aaabc, Z) \vdash (q_1, aaabc, SZ) \vdash (q_1, aabc, AZ)$

$\vdash (q_1, abc, ABCZ) \vdash (q_1, bc, BCZ) \vdash (q_1, c, CZ) \vdash (q_1, \Lambda, Z)$

$\vdash (q_f, \Lambda, Z)$

CFG to PDA

Question Construct a PDA 'A' equivalent to the following Context free grammar:
$S \rightarrow 0BB$, $B \rightarrow 0S \mid 1S \mid 0$.
Test whether 010000 is in N(A).

CFG to PDA

Question Construct a PDA 'A' equivalent to the following Context free grammar:

S $\rightarrow$ 0BB, B $\rightarrow$ 0S $|$1S $|$0.

Test whether 010000 is in N(A).

Solution: Let A $=$({q},{0,1},{S,B,0,1},δ,q,S,ϕ)

CFG to PDA

Question Construct a PDA 'A' equivalent to the following Context free grammar:

S $\rightarrow$ 0BB, B $\rightarrow$ 0S |1S |0.

Test whether 010000 is in N(A).

Solution: Let A =({q},{0,1},{S,B,0,1},δ,q,S,ϕ)

δ is defined by following rules:

$\delta(q, \Lambda, Z) = (q, SZ)$

CFG to PDA

Question Construct a PDA 'A' equivalent to the following Context free grammar:

S $\rightarrow$ 0BB, B $\rightarrow$ 0S |1S |0.

Test whether 010000 is in N(A).

Solution: Let A =({q},{0,1},{S,B,0,1},δ,q,S,ϕ)

δ is defined by following rules:

$\delta(q, \Lambda, Z) = (q, SZ)$

$\delta(q, 0, S) = (q, BB)$

CFG to PDA

Question Construct a PDA 'A' equivalent to the following Context free grammar:

S $\rightarrow$ 0BB, B $\rightarrow$ 0S $|$1S $|$0.

Test whether 010000 is in N(A).

Solution: Let A $=(\{q\},\{0,1\},\{S,B,0,1\},\delta,q,S,\phi)$

δ is defined by following rules:

$\delta(q, \Lambda, Z) = (q, SZ)$

$\delta(q, 0, S) = (q, BB)$

$\delta(q, 0, B) = (q, S)$

CFG to PDA

Question Construct a PDA 'A' equivalent to the following Context free grammar:

S $\rightarrow$ 0BB, B $\rightarrow$ 0S |1S |0.

Test whether 010000 is in N(A).

Solution: Let A =({q},{0,1},{S,B,0,1},δ,q,S,ϕ)

δ is defined by following rules:

$\delta(q, \Lambda, Z) = (q, SZ)$

$\delta(q, 0, S) = (q, BB)$

$\delta(q, 0, B) = (q, S)$

$\delta(q, 1, B) = (q, S)$

CFG to PDA

Question Construct a PDA 'A' equivalent to the following Context free grammar:

S $\rightarrow$ 0BB, B $\rightarrow$ 0S |1S |0.

Test whether 010000 is in N(A).

Solution: Let A =({q},{0,1},{S,B,0,1},δ,q,S,ϕ)

δ is defined by following rules:

$\delta(q, \Lambda, Z) = (q, SZ)$

$\delta(q, 0, S) = (q, BB)$

$\delta(q, 0, B) = (q, S)$

$\delta(q, 1, B) = (q, S)$

$\delta(q, 0, B) = (q, \Lambda)$

CFG to PDA

Question Construct a PDA 'A' equivalent to the following Context free grammar:

S $\rightarrow$ 0BB, B $\rightarrow$ 0S $|$1S $|$0.

Test whether 010000 is in N(A).

Solution: Let A $=(\{q\},\{0,1\},\{S,B,0,1\},\delta,q,S,\phi)$

δ is defined by following rules:

$\delta(q, \Lambda, Z) = (q, SZ)$

$\delta(q, 0, S) = (q, BB)$

$\delta(q, 0, B) = (q, S)$

$\delta(q, 1, B) = (q, S)$

$\delta(q, 0, B) = (q, \Lambda)$

$\delta(q, \Lambda, Z) = (q_f, \Lambda)$

CFG to PDA

Question Construct a PDA 'A' equivalent to the following Context free grammar:

S $\rightarrow$ 0BB, B $\rightarrow$ 0S |1S |0.

Test whether 010000 is in N(A).

Solution: Let A =({q},{0,1},{S,B,0,1},δ,q,S,ϕ)

δ is defined by following rules:

$\delta(q, \Lambda, Z) = (q, SZ)$

$\delta(q, 0, S) = (q, BB)$

$\delta(q, 0, B) = (q, S)$

$\delta(q, 1, B) = (q, S)$

$\delta(q, 0, B) = (q, \Lambda)$

$\delta(q, \Lambda, Z) = (q_f, \Lambda)$

Let the derivation be:

S $\rightarrow$ 0BB $\rightarrow$ 01SB $\rightarrow$ 010BBB $\rightarrow$ 010000

CFG to PDA

Question Construct a PDA 'A' equivalent to the following Context free grammar:

S $\rightarrow$ 0BB, B $\rightarrow$ 0S |1S |0.

Test whether 010000 is in N(A).

Solution: Let A =({q},{0,1},{S,B,0,1},δ,q,S,ϕ)

δ is defined by following rules:

$\delta(q, \Lambda, Z) = (q, SZ)$

$\delta(q, 0, S) = (q, BB)$

$\delta(q, 0, B) = (q, S)$

$\delta(q, 1, B) = (q, S)$

$\delta(q, 0, B) = (q, \Lambda)$

$\delta(q, \Lambda, Z) = (q_f, \Lambda)$

Let the derivation be:

S $\rightarrow$ 0BB $\rightarrow$ 01SB $\rightarrow$ 010BBB $\rightarrow$ 010000

Acceptability of 010000:

CFG to PDA

Question Construct a PDA 'A' equivalent to the following Context free grammar:

S $\rightarrow$ 0BB, B $\rightarrow$ 0S |1S |0.

Test whether 010000 is in N(A).

Solution: Let A =({q},{0,1},{S,B,0,1},δ,q,S,ϕ)

δ is defined by following rules:

$\delta(q, \Lambda, Z) = (q, SZ)$

$\delta(q, 0, S) = (q, BB)$

$\delta(q, 0, B) = (q, S)$

$\delta(q, 1, B) = (q, S)$

$\delta(q, 0, B) = (q, \Lambda)$

$\delta(q, \Lambda, Z) = (q_f, \Lambda)$

Let the derivation be:

S $\rightarrow$ 0BB $\rightarrow$ 01SB $\rightarrow$ 010BBB $\rightarrow$ 010000

Acceptability of 010000:

(q,010000,Z)

CFG to PDA

Question Construct a PDA 'A' equivalent to the following Context free grammar:

S $\rightarrow$ 0BB, B $\rightarrow$ 0S |1S |0.

Test whether 010000 is in N(A).

Solution: Let A =({q},{0,1},{S,B,0,1},δ,q,S,ϕ)

δ is defined by following rules:

$\delta(q, \Lambda, Z) = (q, SZ)$

$\delta(q, 0, S) = (q, BB)$

$\delta(q, 0, B) = (q, S)$

$\delta(q, 1, B) = (q, S)$

$\delta(q, 0, B) = (q, \Lambda)$

$\delta(q, \Lambda, Z) = (q_f, \Lambda)$

Let the derivation be:

S $\rightarrow$ 0BB $\rightarrow$ 01SB $\rightarrow$ 010BBB $\rightarrow$ 010000

Acceptability of 010000:

(q,010000,Z) $\vdash$ (q,010000,SZ)

CFG to PDA

Question Construct a PDA 'A' equivalent to the following Context free grammar:

S $\rightarrow$ 0BB, B $\rightarrow$ 0S |1S |0.

Test whether 010000 is in N(A).

Solution: Let A =({q},{0,1},{S,B,0,1},δ,q,S,ϕ)

δ is defined by following rules:

$\delta(q, \Lambda, Z) = (q, SZ)$

$\delta(q, 0, S) = (q, BB)$

$\delta(q, 0, B) = (q, S)$

$\delta(q, 1, B) = (q, S)$

$\delta(q, 0, B) = (q, \Lambda)$

$\delta(q, \Lambda, Z) = (q_f, \Lambda)$

Let the derivation be:

S $\rightarrow$ 0BB $\rightarrow$ 01SB $\rightarrow$ 010BBB $\rightarrow$ 010000

Acceptability of 010000:

(q,010000,Z) $\vdash$ (q,010000,SZ) $\vdash$ (q,10000,BB)

CFG to PDA

Question Construct a PDA 'A' equivalent to the following Context free grammar:

S $\rightarrow$ 0BB, B $\rightarrow$ 0S |1S |0.

Test whether 010000 is in N(A).

Solution: Let A =({q},{0,1},{S,B,0,1},δ,q,S,ϕ)

δ is defined by following rules:

$\delta(q, \Lambda, Z) = (q, SZ)$

$\delta(q, 0, S) = (q, BB)$

$\delta(q, 0, B) = (q, S)$

$\delta(q, 1, B) = (q, S)$

$\delta(q, 0, B) = (q, \Lambda)$

$\delta(q, \Lambda, Z) = (q_f, \Lambda)$

Let the derivation be:

S $\rightarrow$ 0BB $\rightarrow$ 01SB $\rightarrow$ 010BBB $\rightarrow$ 010000

Acceptability of 010000:

(q,010000,Z) $\vdash$ (q,010000,SZ) $\vdash$ (q,10000,BB) $\vdash$ (q,0000,SB)

CFG to PDA

Question Construct a PDA 'A' equivalent to the following Context free grammar:

S $\rightarrow$ 0BB, B $\rightarrow$ 0S |1S |0.

Test whether 010000 is in N(A).

Solution: Let A =({q},{0,1},{S,B,0,1},δ,q,S,ϕ)

δ is defined by following rules:

$\delta(q, \Lambda, Z) = (q, SZ)$

$\delta(q, 0, S) = (q, BB)$

$\delta(q, 0, B) = (q, S)$

$\delta(q, 1, B) = (q, S)$

$\delta(q, 0, B) = (q, \Lambda)$

$\delta(q, \Lambda, Z) = (q_f, \Lambda)$

Let the derivation be:

S $\rightarrow$ 0BB $\rightarrow$ 01SB $\rightarrow$ 010BBB $\rightarrow$ 010000

Acceptability of 010000:

(q,010000,Z) $\vdash$ (q,010000,SZ) $\vdash$ (q,10000,BB) $\vdash$ (q,0000,SB) $\vdash$ (q,000,BBB)

CFG to PDA

Question Construct a PDA 'A' equivalent to the following Context free grammar:

S $\rightarrow$ 0BB, B $\rightarrow$ 0S |1S |0.

Test whether 010000 is in N(A).

Solution: Let A =({q},{0,1},{S,B,0,1},δ,q,S,ϕ)

δ is defined by following rules:

$\delta(q, \Lambda, Z) = (q, SZ)$

$\delta(q, 0, S) = (q, BB)$

$\delta(q, 0, B) = (q, S)$

$\delta(q, 1, B) = (q, S)$

$\delta(q, 0, B) = (q, \Lambda)$

$\delta(q, \Lambda, Z) = (q_f, \Lambda)$

Let the derivation be:

S $\rightarrow$ 0BB $\rightarrow$ 01SB $\rightarrow$ 010BBB $\rightarrow$ 010000

Acceptability of 010000:

(q,010000,Z) $\vdash$ (q,010000,SZ) $\vdash$ (q,10000,BB) $\vdash$ (q,0000,SB) $\vdash$ (q,000,BBB) $\vdash$ (q, 00,BB)

CFG to PDA

Question Construct a PDA 'A' equivalent to the following Context free grammar:

S $\rightarrow$ 0BB, B $\rightarrow$ 0S |1S |0.

Test whether 010000 is in N(A).

Solution: Let A =({q},{0,1},{S,B,0,1},δ,q,S,ϕ)

δ is defined by following rules:

$\delta(q, \Lambda, Z) = (q, SZ)$

$\delta(q, 0, S) = (q, BB)$

$\delta(q, 0, B) = (q, S)$

$\delta(q, 1, B) = (q, S)$

$\delta(q, 0, B) = (q, \Lambda)$

$\delta(q, \Lambda, Z) = (q_f, \Lambda)$

Let the derivation be:

S $\rightarrow$ 0BB $\rightarrow$ 01SB $\rightarrow$ 010BBB $\rightarrow$ 010000

Acceptability of 010000:

(q,010000,Z) $\vdash$ (q,010000,SZ) $\vdash$ (q,10000,BB) $\vdash$ (q,0000,SB) $\vdash$ (q,000,BBB) $\vdash$ (q, 00,BB) $\vdash$ (q,0,B)

CFG to PDA

Question Construct a PDA 'A' equivalent to the following Context free grammar:

S $\rightarrow$ 0BB, B $\rightarrow$ 0S |1S |0.

Test whether 010000 is in N(A).

Solution: Let A =({q},{0,1},{S,B,0,1},δ,q,S,ϕ)

δ is defined by following rules:

$\delta(q, \Lambda, Z) = (q, SZ)$

$\delta(q, 0, S) = (q, BB)$

$\delta(q, 0, B) = (q, S)$

$\delta(q, 1, B) = (q, S)$

$\delta(q, 0, B) = (q, \Lambda)$

$\delta(q, \Lambda, Z) = (q_f, \Lambda)$

Let the derivation be:

S $\rightarrow$ 0BB $\rightarrow$ 01SB $\rightarrow$ 010BBB $\rightarrow$ 010000

Acceptability of 010000:

(q,010000,Z) $\vdash$ (q,010000,SZ) $\vdash$ (q,10000,BB) $\vdash$ (q,0000,SB) $\vdash$ (q,000,BBB) $\vdash$ (q, 00,BB) $\vdash$ (q,0,B) $\vdash$ (q,Λ,Z)

CFG to PDA

Question Construct a PDA 'A' equivalent to the following Context free grammar:

S $\rightarrow$ 0BB, B $\rightarrow$ 0S |1S |0.

Test whether 010000 is in N(A).

Solution: Let A =({q},{0,1},{S,B,0,1},δ,q,S,ϕ)

δ is defined by following rules:

$\delta(q, \Lambda, Z) = (q, SZ)$

$\delta(q, 0, S) = (q, BB)$

$\delta(q, 0, B) = (q, S)$

$\delta(q, 1, B) = (q, S)$

$\delta(q, 0, B) = (q, \Lambda)$

$\delta(q, \Lambda, Z) = (q_f, \Lambda)$

Let the derivation be:

S $\rightarrow$ 0BB $\rightarrow$ 01SB $\rightarrow$ 010BBB $\rightarrow$ 010000

Acceptability of 010000:

(q,010000,Z) $\vdash$ (q,010000,SZ) $\vdash$ (q,10000,BB) $\vdash$ (q,0000,SB) $\vdash$ (q,000,BBB) $\vdash$ (q, 00,BB) $\vdash$ (q,0,B) $\vdash$ (q,Λ,Z) $\vdash$ (q_f, Λ)

PDA to CFG

Theorem

If $A = (Q, \Sigma, \tau, \delta, q_0, Z_0, F)$ is a pda then there exists a context-free grammar G such that $L(G) = N(A)$.

Construction of G for CFG

We define $G = (V, \Sigma, P, S)$, where
$V = \{S\} \cup \{[q, Z, q'] | q, q' \in Q, Z \in \tau\}$ i.e. any element of V is either the new symbol S acting as the start symbol for G or an ordered triple whose first and third elements are states and the second element is a pushdown symbol. The productions in P are induced by moves of pda as follows:

- R_1 S-productions are given by $S \rightarrow [q_0, Z_0, q]$ for every $q \in Q$.
- R_2 Each transition erasing a pushdown symbol given by $\delta(q, a, Z) = (q', \Lambda)$ induces the production $[q, Z, q'] \rightarrow a$

PDA to CFG

R_3 Each transition not erasing a pushdown symbol giving by $\delta(q, a, Z) = (q_1, Z_1 Z_2 \ldots Z_m)$ induces the production $[q, Z, q'] \rightarrow a[q_1, Z_1, q_2][q_2, Z_2, q_3] \ldots [q_m, Z_m, q']$, where each of the states $q', q_2, \ldots, q_m$ can be any state in Q

PDA to CFG

1. $S \rightarrow [q_0, Z_0, q_i]$

PDA to CFG

1. $S \rightarrow [q_0, Z_0, q_i]$
2. $\delta(q, a, Z) = (q', \Lambda)$

PDA to CFG

1. $S \rightarrow [q_0, Z_0, q_i]$
2. $\delta(q, a, Z) = (q', \Lambda)$
   $[q, Z, q'] \rightarrow a$

PDA to CFG

1. $S \rightarrow [q_0, Z_0, q_i]$
2. $\delta(q, a, Z) = (q', \Lambda)$
   $[q, Z, q'] \rightarrow a$
3. $\delta(q, \Lambda, Z) = (q', \Lambda)$

PDA to CFG

1. $S \to [q_0, Z_0, q_i]$
2. $\delta(q, a, Z) = (q', \Lambda)$
   $[q, Z, q'] \to a$
3. $\delta(q, \Lambda, Z) = (q', \Lambda)$
   $[q, Z, q'] \to \Lambda$

PDA to CFG

1. $S \rightarrow [q_0, Z_0, q_i]$
2. $\delta(q, a, Z) = (q', \Lambda)$
   $[q, Z, q'] \rightarrow a$
3. $\delta(q, \Lambda, Z) = (q', \Lambda)$
   $[q, Z, q'] \rightarrow \Lambda$
4. $\delta(q, a, Z) = (q', b)$

PDA to CFG

1. $S \rightarrow [q_0, Z_0, q_i]$
2. $\delta(q, a, Z) = (q', \Lambda)$
   $[q, Z, q'] \rightarrow a$
3. $\delta(q, \Lambda, Z) = (q', \Lambda)$
   $[q, Z, q'] \rightarrow \Lambda$
4. $\delta(q, a, Z) = (q', b)$
   $[q, Z, q_i] \rightarrow a[q', b, q_i]$

PDA to CFG

1. $S \rightarrow [q_0, Z_0, q_i]$
2. $\delta(q, a, Z) = (q', \Lambda)$
   $[q, Z, q'] \rightarrow a$
3. $\delta(q, \Lambda, Z) = (q', \Lambda)$
   $[q, Z, q'] \rightarrow \Lambda$
4. $\delta(q, a, Z) = (q', b)$
   $[q, Z, q_i] \rightarrow a[q', b, q_i]$
5. $\delta(q, a, Z) = (q', bX)$

PDA to CFG

1. $S \rightarrow [q_0, Z_0, q_i]$
2. $\delta(q, a, Z) = (q', \Lambda)$
   $[q, Z, q'] \rightarrow a$
3. $\delta(q, \Lambda, Z) = (q', \Lambda)$
   $[q, Z, q'] \rightarrow \Lambda$
4. $\delta(q, a, Z) = (q', b)$
   $[q, Z, q_i] \rightarrow a[q', b, q_i]$
5. $\delta(q, a, Z) = (q', bX)$
   $[q, Z, q_i] \rightarrow a[q', b, \ldots][\ldots, X, q_i]$

PDA to CFG

1. $S \rightarrow [q_0, Z_0, q_i]$
2. $\delta(q, a, Z) = (q', \Lambda)$
   $[q, Z, q'] \rightarrow a$
3. $\delta(q, \Lambda, Z) = (q', \Lambda)$
   $[q, Z, q'] \rightarrow \Lambda$
4. $\delta(q, a, Z) = (q', b)$
   $[q, Z, q_i] \rightarrow a[q', b, q_i]$
5. $\delta(q, a, Z) = (q', bX)$
   $[q, Z, q_i] \rightarrow a[q', b, \ldots][\ldots, X, q_i]$
6. $\delta(q, a, Z) = (q', bXY)$

PDA to CFG

1. $S \rightarrow [q_0, Z_0, q_i]$
2. $\delta(q, a, Z) = (q', \Lambda)$
   $[q, Z, q'] \rightarrow a$
3. $\delta(q, \Lambda, Z) = (q', \Lambda)$
   $[q, Z, q'] \rightarrow \Lambda$
4. $\delta(q, a, Z) = (q', b)$
   $[q, Z, q_i] \rightarrow a[q', b, q_i]$
5. $\delta(q, a, Z) = (q', bX)$
   $[q, Z, q_i] \rightarrow a[q', b, \ldots][\ldots, X, q_i]$
6. $\delta(q, a, Z) = (q', bXY)$
   $[q, Z, q_i] = a[q', b, \ldots][\ldots, X, \ldots][\ldots, Y, q_i]$

PDA to CFG

Question: Construct a Context free grammar G which accepts N(A), where A=($\{q_0, q_1\}$,{a,b},$\{Z_0$,Z$\}$, δ, q_0, Z_0, ϕ) and δ is given by:

$\delta(q_0, b, Z_0) = (q_0, ZZ_0)$, $\delta(q_0, \Lambda, Z_0) = (q_0, \Lambda)$
$\delta(q_0, b, Z) = (q_0, ZZ)$, $\delta(q_0, a, Z) = (q_1, Z)$
$\delta(q_1, b, Z) = (q_1, \Lambda)$, $\delta(q_1, a, Z_0) = (q_0, Z_0)$

PDA to CFG

Question: Construct a Context free grammar G which accepts N(A), where A=($\{q_0, q_1\}$,{a,b},$\{Z_0$,Z$\}$, δ, q_0, Z_0, ϕ) and δ is given by:

$\delta(q_0, b, Z_0) = (q_0, ZZ_0)$, $\delta(q_0, \Lambda, Z_0) = (q_0, \Lambda)$
$\delta(q_0, b, Z) = (q_0, ZZ)$, $\delta(q_0, a, Z) = (q_1, Z)$
$\delta(q_1, b, Z) = (q_1, \Lambda)$, $\delta(q_1, a, Z_0) = (q_0, Z_0)$

Solution: Let G=$(V_N, \{a, b\}, P, S)$

PDA to CFG

Question: Construct a Context free grammar G which accepts N(A), where A=($\{q_0, q_1\}$,{a,b},$\{Z_0$,Z$\}$, δ, q_0, Z_0, ϕ) and δ is given by:

$\delta(q_0, b, Z_0) = (q_0, ZZ_0)$, $\delta(q_0, \Lambda, Z_0) = (q_0, \Lambda)$

$\delta(q_0, b, Z) = (q_0, ZZ)$, $\delta(q_0, a, Z) = (q_1, Z)$

$\delta(q_1, b, Z) = (q_1, \Lambda)$, $\delta(q_1, a, Z_0) = (q_0, Z_0)$

Solution: Let G=$(V_N, \{a, b\}, P, S)$

$V_N = \{S, [q_0, Z_0, q_0], [q_0, Z_0, q_1], [q_1, Z_0, q_0], [q_1, Z_0, q_1],$
$[q_0, Z, q_0], [q_0, Z, q_1], [q_1, Z, q_0], [q_1, Z, q_1]\}$

PDA to CFG

Question: Construct a Context free grammar G which accepts N(A), where A=($\{q_0, q_1\}$,{a,b},{Z_0,Z}, δ, q_0, Z_0, ϕ) and δ is given by:

$\delta(q_0, b, Z_0) = (q_0, ZZ_0)$, $\delta(q_0, \Lambda, Z_0) = (q_0, \Lambda)$
$\delta(q_0, b, Z) = (q_0, ZZ)$, $\delta(q_0, a, Z) = (q_1, Z)$
$\delta(q_1, b, Z) = (q_1, \Lambda)$, $\delta(q_1, a, Z_0) = (q_0, Z_0)$

Solution: Let G=$(V_N, \{a, b\}, P, S)$

$V_N = \{S, [q_0, Z_0, q_0], [q_0, Z_0, q_1], [q_1, Z_0, q_0], [q_1, Z_0, q_1],$
$[q_0, Z, q_0], [q_0, Z, q_1], [q_1, Z, q_0], [q_1, Z, q_1]\}$

The Productions are:

Initial: S $\rightarrow [q_0, Z_0, q_0], [q_0, Z_0, q_1]$

PDA to CFG

Question: Construct a Context free grammar G which accepts N(A), where A=($\{q_0, q_1\}$,{a,b},$\{Z_0$,Z$\}$, δ, q_0, Z_0, ϕ) and δ is given by:

$\delta(q_0, b, Z_0) = (q_0, ZZ_0)$, $\delta(q_0, \Lambda, Z_0) = (q_0, \Lambda)$

$\delta(q_0, b, Z) = (q_0, ZZ)$, $\delta(q_0, a, Z) = (q_1, Z)$

$\delta(q_1, b, Z) = (q_1, \Lambda)$, $\delta(q_1, a, Z_0) = (q_0, Z_0)$

Solution: Let G=$(V_N, \{a, b\}, P, S)$

$V_N = \{S, [q_0, Z_0, q_0], [q_0, Z_0, q_1], [q_1, Z_0, q_0], [q_1, Z_0, q_1],$
$[q_0, Z, q_0], [q_0, Z, q_1], [q_1, Z, q_0], [q_1, Z, q_1]\}$

The Productions are:

Initial: S $\rightarrow [q_0, Z_0, q_0], [q_0, Z_0, q_1]$

For $\delta(q_0, b, Z_0) = (q_0, ZZ_0)$

PDA to CFG

Question: Construct a Context free grammar G which accepts N(A), where A=($\{q_0, q_1\}$,{a,b},$\{Z_0$,Z$\}$, δ, q_0, Z_0, ϕ) and δ is given by:

$\delta(q_0, b, Z_0) = (q_0, ZZ_0)$, $\delta(q_0, \Lambda, Z_0) = (q_0, \Lambda)$
$\delta(q_0, b, Z) = (q_0, ZZ)$, $\delta(q_0, a, Z) = (q_1, Z)$
$\delta(q_1, b, Z) = (q_1, \Lambda)$, $\delta(q_1, a, Z_0) = (q_0, Z_0)$

Solution: Let G=($V_N, \{a, b\}, P, S$)

$V_N = \{S, [q_0, Z_0, q_0], [q_0, Z_0, q_1], [q_1, Z_0, q_0], [q_1, Z_0, q_1],$
$[q_0, Z, q_0], [q_0, Z, q_1], [q_1, Z, q_0], [q_1, Z, q_1]\}$

The Productions are:

Initial: S $\rightarrow [q_0, Z_0, q_0], [q_0, Z_0, q_1]$

For $\delta(q_0, b, Z_0) = (q_0, ZZ_0)$

$[q_0 Z_0 \ldots] \rightarrow b[q_0 Z \ldots][\ldots Z_0 \ldots]$
$[q_0 Z_0 \ldots] \rightarrow b[q_0 Z \ldots][\ldots Z_0 \ldots]$
$[q_0 Z_0 \ldots] \rightarrow b[q_0 Z \ldots][\ldots Z_0 \ldots]$
$[q_0 Z_0 \ldots] \rightarrow b[q_0 Z \ldots][\ldots Z_0 \ldots]$

PDA to CFG

Question: Construct a Context free grammar G which accepts N(A), where A=($\{q_0, q_1\}$,{a,b},{Z_0,Z}, δ, q_0, Z_0, ϕ) and δ is given by:

$\delta(q_0, b, Z_0) = (q_0, ZZ_0)$, $\delta(q_0, \Lambda, Z_0) = (q_0, \Lambda)$
$\delta(q_0, b, Z) = (q_0, ZZ)$, $\delta(q_0, a, Z) = (q_1, Z)$
$\delta(q_1, b, Z) = (q_1, \Lambda)$, $\delta(q_1, a, Z_0) = (q_0, Z_0)$

Solution: Let G=$(V_N, \{a, b\}, P, S)$

$V_N = \{S, [q_0, Z_0, q_0], [q_0, Z_0, q_1], [q_1, Z_0, q_0], [q_1, Z_0, q_1],$
$[q_0, Z, q_0], [q_0, Z, q_1], [q_1, Z, q_0], [q_1, Z, q_1]\}$

The Productions are:

Initial: S $\rightarrow [q_0, Z_0, q_0], [q_0, Z_0, q_1]$

For $\delta(q_0, b, Z_0) = (q_0, ZZ_0)$

$[q_0 Z_0 q_0] \rightarrow b[q_0 Z q_0][q_0 Z_0 q_0]$
$[q_0 Z_0 q_0] \rightarrow b[q_0 Z q_1][q_1 Z_0 q_0]$
$[q_0 Z_0 q_1] \rightarrow b[q_0 Z q_0][q_0 Z_0 q_1]$
$[q_0 Z_0 q_1] \rightarrow b[q_0 Z q_1][q_1 Z_0 q_1]$

PDA to CFG

For $\delta(q_0, \wedge, Z_0) = (q_0, \wedge)$
$[q_0 Z_0 q_0] \rightarrow \wedge$

PDA to CFG

For $\delta(q_0, \wedge, Z_0) = (q_0, \wedge)$

$[q_0 Z_0 q_0] \rightarrow \wedge$

$(q_0, b, Z) = (q_0, ZZ)$

$[q_0 Z \ldots] \rightarrow b[q_0 Z \ldots][\ldots Z \ldots]$

$[q_0 Z \ldots] \rightarrow b[q_0 Z \ldots][\ldots Z \ldots]$

$[q_0 Z \ldots] \rightarrow b[q_0 Z \ldots][\ldots Z \ldots]$

$[q_0 Z \ldots] \rightarrow b[q_0 Z \ldots][\ldots Z \ldots]$

PDA to CFG

For $\delta(q_0, \wedge, Z_0) = (q_0, \wedge)$

$[q_0 Z_0 q_0] \rightarrow \wedge$

$(q_0, b, Z) = (q_0, ZZ)$

$[q_0 Z q_0] \rightarrow b[q_0 Z q_0][q_0 Z q_0]$

$[q_0 Z q_0] \rightarrow b[q_0 Z q_1][q_1 Z q_0]$

$[q_0 Z q_1] \rightarrow b[q_0 Z q_0][q_0 Z q_1]$

$[q_0 Z q_1] \rightarrow b[q_0 Z q_1][q_1 Z q_1]$

PDA to CFG

For $\delta(q_0, \wedge, Z_0) = (q_0, \wedge)$

$[q_0 Z_0 q_0] \to \wedge$

$(q_0, b, Z) = (q_0, ZZ)$

$[q_0 Z q_0] \to b[q_0 Z q_0][q_0 Z q_0]$

$[q_0 Z q_0] \to b[q_0 Z q_1][q_1 Z q_0]$

$[q_0 Z q_1] \to b[q_0 Z q_0][q_0 Z q_1]$

$[q_0 Z q_1] \to b[q_0 Z q_1][q_1 Z q_1]$

For $(q_0, a, Z) = (q_1, Z)$

PDA to CFG

For $\delta(q_0, \wedge, Z_0) = (q_0, \wedge)$

$[q_0 Z_0 q_0] \rightarrow \wedge$

$(q_0, b, Z) = (q_0, ZZ)$

$[q_0 Z q_0] \rightarrow b[q_0 Z q_0][q_0 Z q_0]$

$[q_0 Z q_0] \rightarrow b[q_0 Z q_1][q_1 Z q_0]$

$[q_0 Z q_1] \rightarrow b[q_0 Z q_0][q_0 Z q_1]$

$[q_0 Z q_1] \rightarrow b[q_0 Z q_1][q_1 Z q_1]$

For $(q_0, a, Z) = (q_1, Z)$

$[q_0 Z \ldots] \rightarrow a[q_1 Z \ldots]$

$[q_0 Z \ldots] \rightarrow a[q_1 Z \ldots]$

PDA to CFG

For $\delta(q_0, \wedge, Z_0) = (q_0, \wedge)$

$[q_0 Z_0 q_0] \rightarrow \wedge$

$(q_0, b, Z) = (q_0, ZZ)$

$[q_0 Z q_0] \rightarrow b[q_0 Z q_0][q_0 Z q_0]$

$[q_0 Z q_0] \rightarrow b[q_0 Z q_1][q_1 Z q_0]$

$[q_0 Z q_1] \rightarrow b[q_0 Z q_0][q_0 Z q_1]$

$[q_0 Z q_1] \rightarrow b[q_0 Z q_1][q_1 Z q_1]$

For $(q_0, a, Z) = (q_1, Z)$

$[q_0 Z q_0] \rightarrow a[q_1 Z q_0]$

$[q_0 Z q_1] \rightarrow a[q_1 Z q_1]$

PDA to CFG

For $\delta(q_0, \wedge, Z_0) = (q_0, \wedge)$

$[q_0 Z_0 q_0] \rightarrow \wedge$

$(q_0, b, Z) = (q_0, ZZ)$

$[q_0 Z q_0] \rightarrow b[q_0 Z q_0][q_0 Z q_0]$

$[q_0 Z q_0] \rightarrow b[q_0 Z q_1][q_1 Z q_0]$

$[q_0 Z q_1] \rightarrow b[q_0 Z q_0][q_0 Z q_1]$

$[q_0 Z q_1] \rightarrow b[q_0 Z q_1][q_1 Z q_1]$

For $(q_0, a, Z) = (q_1, Z)$

$[q_0 Z q_0] \rightarrow a[q_1 Z q_0]$

$[q_0 Z q_1] \rightarrow a[q_1 Z q_1]$

For $\delta(q_1, b, Z) = (q_1, \wedge)$

$[q_1, Z, q_1] \rightarrow b$

PDA to CFG

For $\delta(q_0, \wedge, Z_0) = (q_0, \wedge)$

$[q_0 Z_0 q_0] \rightarrow \wedge$

$(q_0, b, Z) = (q_0, ZZ)$

$[q_0 Z q_0] \rightarrow b[q_0 Z q_0][q_0 Z q_0]$

$[q_0 Z q_0] \rightarrow b[q_0 Z q_1][q_1 Z q_0]$

$[q_0 Z q_1] \rightarrow b[q_0 Z q_0][q_0 Z q_1]$

$[q_0 Z q_1] \rightarrow b[q_0 Z q_1][q_1 Z q_1]$

For $(q_0, a, Z) = (q_1, Z)$

$[q_0 Z q_0] \rightarrow a[q_1 Z q_0]$

$[q_0 Z q_1] \rightarrow a[q_1 Z q_1]$

For $\delta(q_1, b, Z) = (q_1, \wedge)$

$[q_1, Z, q_1] \rightarrow b$

For $\delta(q_1, a, Z_0) \rightarrow (q_0, Z_0)$

$[q_1, Z_0, \ldots] \rightarrow a[q_0 Z_0 \ldots]$

$[q_1, Z_0, \ldots] \rightarrow a[q_0 Z_0 \ldots]$

PDA to CFG

For $\delta(q_0, \wedge, Z_0) = (q_0, \wedge)$

$[q_0 Z_0 q_0] \rightarrow \wedge$

$(q_0, b, Z) = (q_0, ZZ)$

$[q_0 Z q_0] \rightarrow b[q_0 Z q_0][q_0 Z q_0]$

$[q_0 Z q_0] \rightarrow b[q_0 Z q_1][q_1 Z q_0]$

$[q_0 Z q_1] \rightarrow b[q_0 Z q_0][q_0 Z q_1]$

$[q_0 Z q_1] \rightarrow b[q_0 Z q_1][q_1 Z q_1]$

For $(q_0, a, Z) = (q_1, Z)$

$[q_0 Z q_0] \rightarrow a[q_1 Z q_0]$

$[q_0 Z q_1] \rightarrow a[q_1 Z q_1]$

For $\delta(q_1, b, Z) = (q_1, \wedge)$

$[q_1, Z, q_1] \rightarrow b$

For $\delta(q_1, a, Z_0) \rightarrow (q_0, Z_0)$

$[q_1, Z_0, q_0] \rightarrow a[q_0 Z_0 q_0]$

$[q_1, Z_0, q_1] \rightarrow a[q_0 Z_0 q_1]$

PDA to CFG

1. Let M=($\{q_0, q_1\}$,{a,b},{a,Z_0}, δ, q_0, Z_0, ϕ)
   where productions are
   $\delta(q_0, a, Z_0) = (q_0, aZ_0)$
   $\delta(q_0, a, a) = (q_0, aa)$
   $\delta(q_0, b, a) = (q_1, \wedge)$
   $\delta(q_1, b, a) = (q_1, \wedge)$
   $\delta(q_1, \wedge, Z_0) = (q_1, \wedge)$
   Find grammar G.
2. Find a context free grammar that generates the language accepted by NPDA
   M=($\{q_0, q_1\}$,{a,b},{A,Z},δ,q_0,Z,$\{q_1\}$) with transitions
   $\delta(q_0, a, Z) = (q_0, AZ)$
   $\delta(q_0, b, A) = (q_0, AA)$
   $\delta(q_0, a, A) = (q_1, \Lambda)$

PDA to CFG

Example: Construct a PDA accepting $\{a^n b^m a^n | m, n \geq 1\}$ by null store. Construct the corresponding CFG accepting the same set.

PDA to CFG

Example: Construct a PDA accepting $\{a^n b^m a^n | m, n \geq 1\}$ by null store. Construct the corresponding CFG accepting the same set.

Solution: The PDA 'A' accepting $\{a^n b^m a^n | m, n \geq 1\}$ is defined as

PDA to CFG

Example: Construct a PDA accepting $\{a^n b^m a^n | m, n \geq 1\}$ by null store. Construct the corresponding CFG accepting the same set.
Solution: The PDA 'A' accepting $\{a^n b^m a^n | m, n \geq 1\}$ is defined as A=$(\{q_0, q_1\}$,{a,b},{a,Z_0}, $\delta, q_0, Z_0, \phi)$

PDA to CFG

Example: Construct a PDA accepting $\{a^n b^m a^n | m, n \geq 1\}$ by null store. Construct the corresponding CFG accepting the same set.

Solution: The PDA 'A' accepting $\{a^n b^m a^n | m, n \geq 1\}$ is defined as

A=($\{q_0, q_1\}$,{a,b},{a,Z_0}, δ, q_0, Z_0, ϕ)

where δ is defined by:

$\delta(q_0, a, Z_0) = (q_0, aZ_0)$

PDA to CFG

Example: Construct a PDA accepting $\{a^n b^m a^n | m, n \geq 1\}$ by null store. Construct the corresponding CFG accepting the same set.

Solution: The PDA 'A' accepting $\{a^n b^m a^n | m, n \geq 1\}$ is defined as

A=($\{q_0, q_1\}$,{a,b},{a,Z_0}, δ, q_0, Z_0, ϕ)

where δ is defined by:

$\delta(q_0, a, Z_0) = (q_0, aZ_0)$

$\delta(q_0, a, a) = (q_0, aa)$

PDA to CFG

Example: Construct a PDA accepting $\{a^n b^m a^n | m, n \geq 1\}$ by null store. Construct the corresponding CFG accepting the same set.

Solution: The PDA 'A' accepting $\{a^n b^m a^n | m, n \geq 1\}$ is defined as A=($\{q_0, q_1\}$,{a,b},{a,Z_0}, δ, q_0, Z_0, ϕ)

where δ is defined by:

$\delta(q_0, a, Z_0) = (q_0, aZ_0)$

$\delta(q_0, a, a) = (q_0, aa)$

$\delta(q_0, b, a) = (q_1, a)$

PDA to CFG

Example: Construct a PDA accepting $\{a^n b^m a^n | m, n \geq 1\}$ by null store. Construct the corresponding CFG accepting the same set.

Solution: The PDA 'A' accepting $\{a^n b^m a^n | m, n \geq 1\}$ is defined as

A=($\{q_0, q_1\}$,{a,b},{a,Z_0}, δ, q_0, Z_0, ϕ)

where δ is defined by:

$\delta(q_0, a, Z_0) = (q_0, aZ_0)$

$\delta(q_0, a, a) = (q_0, aa)$

$\delta(q_0, b, a) = (q_1, a)$

$\delta(q_1, b, a) = (q_1, a)$

PDA to CFG

Example: Construct a PDA accepting $\{a^n b^m a^n | m, n \geq 1\}$ by null store. Construct the corresponding CFG accepting the same set.

Solution: The PDA 'A' accepting $\{a^n b^m a^n | m, n \geq 1\}$ is defined as

A=($\{q_0, q_1\}$,{a,b},{a,Z_0}, δ, q_0, Z_0, ϕ)

where δ is defined by:

$\delta(q_0, a, Z_0) = (q_0, aZ_0)$

$\delta(q_0, a, a) = (q_0, aa)$

$\delta(q_0, b, a) = (q_1, a)$

$\delta(q_1, b, a) = (q_1, a)$

$\delta(q_1, a, a) = (q_1, \wedge)$

PDA to CFG

Example: Construct a PDA accepting $\{a^n b^m a^n | m, n \geq 1\}$ by null store. Construct the corresponding CFG accepting the same set.
Solution: The PDA 'A' accepting $\{a^n b^m a^n | m, n \geq 1\}$ is defined as
A=($\{q_0, q_1\}$,{a,b},{a,Z_0}, δ, q_0, Z_0, ϕ)
where δ is defined by:

$\delta(q_0, a, Z_0) = (q_0, aZ_0)$
$\delta(q_0, a, a) = (q_0, aa)$
$\delta(q_0, b, a) = (q_1, a)$
$\delta(q_1, b, a) = (q_1, a)$
$\delta(q_1, a, a) = (q_1, \wedge)$
$\delta(q_1, \wedge, Z_0) = (q_1, \wedge)$

PDA to CFG

Example: Construct a PDA accepting $\{a^n b^m a^n | m, n \geq 1\}$ by null store. Construct the corresponding CFG accepting the same set.

Solution: The PDA 'A' accepting $\{a^n b^m a^n | m, n \geq 1\}$ is defined as

A=($\{q_0, q_1\}$,{a,b},{a,Z_0}, δ, q_0, Z_0, ϕ)

where δ is defined by:

$\delta(q_0, a, Z_0) = (q_0, aZ_0)$

$\delta(q_0, a, a) = (q_0, aa)$

$\delta(q_0, b, a) = (q_1, a)$

$\delta(q_1, b, a) = (q_1, a)$

$\delta(q_1, a, a) = (q_1, \wedge)$

$\delta(q_1, \wedge, Z_0) = (q_1, \wedge)$

Let the required grammar G=($V_N, \{a, b\}, P, S$)

PDA to CFG

Example: Construct a PDA accepting $\{a^n b^m a^n | m, n \geq 1\}$ by null store. Construct the corresponding CFG accepting the same set.

Solution: The PDA 'A' accepting $\{a^n b^m a^n | m, n \geq 1\}$ is defined as

A=($\{q_0, q_1\}$,{a,b},{a,Z_0}, δ, q_0, Z_0, ϕ)

where δ is defined by:

$\delta(q_0, a, Z_0) = (q_0, aZ_0)$

$\delta(q_0, a, a) = (q_0, aa)$

$\delta(q_0, b, a) = (q_1, a)$

$\delta(q_1, b, a) = (q_1, a)$

$\delta(q_1, a, a) = (q_1, \wedge)$

$\delta(q_1, \wedge, Z_0) = (q_1, \wedge)$

Let the required grammar G=$(V_N, \{a, b\}, P, S)$

$V_N =$

$(S, [q_0 Z_0 q_0], [q_0 Z_0 q_1], [q_1 Z_0 q_0], [q_1 Z_0 q_1], q_0 a q_0], [q_0 a q_1], [q_1 a q_0], [q_1 a q_1])$

PDA to CFG

Example: Construct a PDA accepting $\{a^n b^m a^n | m, n \geq 1\}$ by null store. Construct the corresponding CFG accepting the same set.

Solution: The PDA 'A' accepting $\{a^n b^m a^n | m, n \geq 1\}$ is defined as

A=($\{q_0, q_1\}$,{a,b},{a,Z_0}, δ, q_0, Z_0, ϕ)

where δ is defined by:

$\delta(q_0, a, Z_0) = (q_0, aZ_0)$

$\delta(q_0, a, a) = (q_0, aa)$

$\delta(q_0, b, a) = (q_1, a)$

$\delta(q_1, b, a) = (q_1, a)$

$\delta(q_1, a, a) = (q_1, \wedge)$

$\delta(q_1, \wedge, Z_0) = (q_1, \wedge)$

Let the required grammar G=$(V_N, \{a, b\}, P, S)$

$V_N =$

$(S, [q_0 Z_0 q_0], [q_0 Z_0 q_1], [q_1 Z_0 q_0], [q_1 Z_0 q_1], q_0 a q_0], [q_0 a q_1], [q_1 a q_0], [q_1 a q_1])$

...

Exercise

1. What do you understand by LL(k) grammar? Explain with a suitable example.
2. What do you understand by Parsing? How Top-Down Parsing is different from Bottom-Up Parsing? Explain with suitable example.
3. What is left factoring? How is it different from Left recursion?
4. Construct a PDA accepting the set of ll even-length palindromes over {a,b} by the empty store.

Questions?

+91 9911375598, akyadav@akyadav.in

Thank you