Formal Grammar

- Definition:
  - A formal grammar is a 4-tuple
-
- Notation:
  - $G = (N, T, P, S)$
- Where:
  - G : formal grammar
  - N : set of nonterminal symbols (capital letters)
  - T : set of terminal symbols (small letters)
  - S : start symbol (goal / sentence symbol)
-
- Note: $N \cap T = \phi$ $S \notin (N \cup T)$
-
  - P : set of productions
    - A production is an ordered pair (α, β) which is written in the form
  - $\alpha \dashrightarrow \beta$.
    - where: $\alpha = \varphi A \psi$
    - $\beta = \varphi \omega \psi$
    - such that: $\omega, \varphi, \psi \in (N \cup T)^*$
    - $A \in N \cup \{ \Sigma \}$

Exercise

- Construct a Formal Grammar for the set of numerical strings. That is, {0,1,2,3,4,5,6,7,8,9,10,11, …100, 101, …, 00, 01, 001, 005}

Exercise

- Construct a Formal Grammar for the set of numerical strings. That is, {0,1,2,3,4,5,6,7,8,9,10,11, …100, 101, …}

Formal Grammar, G=(N,T,P,Σ}

Σ=Start Symbol

N={A}

T={0,1,2,3,4,5,6,7,8,9}

P={Σ→A, A→0, A→1, A→2, A→3, A→4, A→5, A→6, A→7, A→8, A→9, A→0A, A→1A, A→2A, A→3A, A→4A, A→5A, A→6A, A→7A, A→8A, A→9A}

Exercise

- Construct a Formal Grammar for the set of numerical strings. That is, {0,1,2,3,4,5,6,7,8,9,10,11, …100, 101, …}

Formal Grammar, G=(N,T,P,Σ}

Σ=Start Symbol

N={A}

T={0,1,2,3,4,5,6,7,8,9}

P={$\Sigma \rightarrow$A, A$\rightarrow$0, A$\rightarrow$1, A$\rightarrow$2, A$\rightarrow$3, A$\rightarrow$4, A$\rightarrow$5,

A$\rightarrow$6, A$\rightarrow$7, A$\rightarrow$8, A$\rightarrow$9, A$\rightarrow$0A, A$\rightarrow$1A, A$\rightarrow$2A, A$\rightarrow$3A, A$\rightarrow$4A, A$\rightarrow$5A, A$\rightarrow$6A, A$\rightarrow$7A, A$\rightarrow$8A, A$\rightarrow$9A}

The above grammar generates 01, 001, 0001, etc.. Leading 0's to any integer is allowed.

Exercise: Construct a Formal Grammar such that leading zeroes are disallowed.

Formal Grammar, $G=(N,T,P,\Sigma\}$

Σ=Start Symbol

N={A,B}

T={0,1,2,3,4,5,6,7,8,9}

$P=\{\Sigma \rightarrow AB,\ \Sigma \rightarrow A,\ \Sigma \rightarrow 0,\ A \rightarrow 1,\ A \rightarrow 2,\ A \rightarrow 3,\ A \rightarrow 4,$
$A \rightarrow 5,\ A \rightarrow 6,\ A \rightarrow 7,\ A \rightarrow 8,\ A \rightarrow 9,\ A \rightarrow 1B,\ A \rightarrow 2B,$
$A \rightarrow 3B,\ A \rightarrow 4B,\ A \rightarrow 5B,\ A \rightarrow 6B,\ A \rightarrow 7B,\ A \rightarrow 8B,$
$A \rightarrow 9B,\ B \rightarrow 0,\ B \rightarrow 1,\ B \rightarrow 2,\ B \rightarrow 3,\ B \rightarrow 4,$
$B \rightarrow 5, B \rightarrow 6,\ B \rightarrow 7,\ B \rightarrow 8,\ B \rightarrow 9,\ B \rightarrow 0B,\ B \rightarrow 1B,$
$B \rightarrow 2B,\ B \rightarrow 3B,\ B \rightarrow 4B,\ B \rightarrow 5B,\ B \rightarrow 6B,\ B \rightarrow 7B,$
$B \rightarrow 8B,\ B \rightarrow 9B\}$

Types of Formal Grammar (Chomsky's Hierarchy of Formal Grammars)

- 1) Type 0 : Unrestricted Grammar (G_0)
- : Contracting Grammar
- Production Rule: $\varphi A \psi \rightarrow \varphi \omega \psi$

  where $\omega, \varphi, \psi \in (N \cup T)^*$ and $A \in (N \cup \{\Sigma\})$

- 2) Type 1 : Context-Sensitive Grammar (G_1)
- Production Rules: $\varphi A \psi \rightarrow \varphi \omega \psi, \quad \omega \neq \lambda$
- $\Sigma \rightarrow \lambda$ (when $\lambda \in L(G)$)
-
- 3) Type 2 : Context-Free Grammar (G_2)
- Production Rules: $A \rightarrow \omega$
- $\Sigma \rightarrow \lambda$ (when $\lambda \in L(G)$
- where: $A \in N \cup \{\Sigma\}$
- $\omega \in (N \cup T)^* - \{\lambda\}$
- 4) Type 3 : Regular Grammar (G_3)
- a. Right-Linear
- b. Left-Linear

Formal Grammar

- 4) Type 3 : Regular Grammar (G_3)
-
- a. Right-Linear b. Left-Linear
-
- A --> aB A --> Ba
- A --> a A --> a
-
- where: $A \in N \cup \{\Sigma\}$ where: $A \in N \cup \{\Sigma\}$
- $B \in N$ $B \in N$
- $a \in T$ $a \in T$

Formal Grammar

- **TYPES OF GRAMMAR:**
-
- 1) Type 0 : Unrestricted Grammar (G_0)
- : Contracting Grammar
-
- Production Rule: $\varphi A \psi$ --> $\varphi \omega \psi$
-
- 2) Type 1 : Context-Sensitive Grammar (G_1)
-
- Production Rules: $\varphi A \psi \rightarrow \varphi \omega \psi$, $\omega \neq \lambda$
- Σ --> λ (when $\lambda \in L(G)$)
-
- 3) Type 2 : Context-Free Grammar (G_2)
-
- Production Rules: A --> ω
- Σ --> λ
-
- where: $A \in N \cup \{ \Sigma \}$
- $\omega \in (N \cup T)^* - \{\lambda\}$
-
- 4) Type 3 : Regular Grammar (G_3)
-
- a. Right-Linear
- b. Left-Linear

Types of Formal Grammar

- 1) Type 0 : Unrestricted Grammar (G_0)
- : Contracting Grammar
-
- Production Rule: $\varphi A \psi \rightarrow \varphi \omega \psi$
-
- 2) Type 1 : Context-Sensitive Grammar (G_1)
-
- Production Rules: $\varphi A \psi \rightarrow \varphi \omega \psi$, $\omega \neq \lambda$
- $\Sigma \rightarrow \lambda$ (when $\lambda \in L(G)$)
-
- 3) Type 2 : Context-Free Grammar (G_2)
-
- Production Rules: A --> ω
- Σ --> λ
-
- where: $A \in N \cup \{\Sigma\}$
- $\omega \in (N \cup T)^* - \{\lambda\}$
-
- 4) Type 3 : Regular Grammar (G_3)
-
- a. Right-Linear
- b. Left-Linear

Formal Grammar

- 4) Type 3 : Regular Grammar (G_3)
-
- a. Right-Linear b. Left-Linear
-
- $A \text{ --> } aB$ $A \text{ --> } Ba$
- $A \text{ --> } a$ $A \text{ --> } a$
-
- where: $A \in N \cup \{\Sigma\}$ where: $A \in N \cup \{\Sigma\}$
- $B \in N$ $B \in N$
- $a \in T$ $a \in T$

Formal Grammar (Chomsky's Hierarchy of Grammars)

G_0 = Unrestricted Grammar

$L(G_0)$ = Language of the Unrestricted Grammar

G_1 = Context-Sensitive Grammar

$L(G_1)$ = Language of the Context-Sensitive Grammar

G_2 = Context-Free Grammar

$L(G_2)$ = Language of the Context-Free Grammar

G_3 = Regular Grammar

$L(G_1)$ = Language of the Regular Grammar

$$L(G_0) \supseteq L(G_1) \supseteq L(G_2) \supseteq L(G_3)$$

Is the following formal grammar a Regular Grammar?

Formal Grammar, $G=(N,T,P,\Sigma\}$

Σ=Start Symbol

N={A}

T={0,1,2,3,4,5,6,7,8,9}

P={$\Sigma\to$1A, $\Sigma\to$2A, $\Sigma\to$3A, $\Sigma\to$4A, $\Sigma\to$5A, $\Sigma\to$6A, $\Sigma\to$7A, $\Sigma\to$8A, $\Sigma\to$9A, $\Sigma\to$0, $\Sigma\to$1, $\Sigma\to$2, $\Sigma\to$3, $\Sigma\to$4, $\Sigma\to$5,$\Sigma\to$6, $\Sigma\to$7, $\Sigma\to$8, $\Sigma\to$9, A$\to$0, A$\to$1, A$\to$2, A$\to$3, A$\to$4, A$\to$5,A$\to$6, A$\to$7, A$\to$8, A$\to$9, A$\to$0A, A$\to$1A, A$\to$2A, A$\to$3A, A$\to$4A, A$\to$5A, A$\to$6A, A$\to$7A, A$\to$8A, A$\to$9A}

Formal Grammar

- Exercise:
-
- Given the grammar G = (Σ, N, T, P} :
- Σ = start symbol
- N = { A, B, C }
- T = { a, b, c }
- P = {Σ --> A, A --> aABC, A --> abC, CB --> BC,
- bB--> bb, bC --> bc, cC --> cc }
-
- Derive $a^3b^3c^3$
-

Formal Grammar

- Given the grammar G1 = (Σ, N, T, P} :
- Σ = start symbol
- N = { A, B, C }
- T = { a, b, c }
- P = { S -> A, A --> aABC, A --> abC, CB --> BC, bB --> bb, bC --> bc, cC --> cc }
-
- Derive $a^3b^3c^3$
-
- Solution:
-
- Σ --> A --> aABC --> aaABCBC --> aaabCBCBC --> aaabBCCBC --> aaabBCBCC --> aaabbCBCC --> aaabbBCCC --> aaabbbCCC --> aaabbbcCC --> aaabbbccC --> aaabbbccc
-
- It can be shown that the Language of Grammar G1 = { $a^kb^kc^k \mid k \geq 1$ }
-
- OR L(G1) = { $a^kb^kc^k \mid k \geq 1$ }

Formal Grammar

Given the grammar G2={Σ, N, T, P} :

N = { A, B, C }

T = { a, b }

P = {Σ --> A, A --> aABC, A --> abc, CB --> BC, bB --> bb, bC --> b}

-

  Derive a^2b^2

-

Formal Grammar

Given the grammar G2={Σ, N, T, P} :

N = { A, B, C }

T = { a, b }

P = {Σ --> A, A --> aABC, A --> abC, CB --> BC, bB --> bb, bC --> b}

Derive a^2b^2

Solution:

- Σ --> A --> aABC --> aabCBC --> aabBC --> aabbC --> aabb

It can be verified that L (G2) = { a^kb^k | $k \geq 1$ }

For $k \geq 0$, include λ in the production, that is, Σ --> λ

Formal Grammar

- RIGHT-LINEAR GRAMMAR
- ex: N = { A,B }
- T = { 0,1 }
- P : Σ --> 1B
- Σ --> 1
- A --> 1B
- B --> 0A
- A --> 1
- Solution:
- Σ --> 1
- Σ --> 1B --> 10A --> 101
- Σ --> 1B --> 10A --> 101B --> 1010A --> 10101
- L(G) = 1 (01)*

Formal Grammar

- LEFT-LINEAR GRAMMAR
- ex: N = { A,B }
- T = { 0,1 }
- P : Σ --> B1
- Σ --> 1
- A --> B1
- B --> A0
- A --> 1
- Solution:
- Σ --> 1
- Σ --> B1 --> A01 --> 101
- Σ --> B1 --> A01 --> B101 --> A0101 --> 10101
- L(G) = 1 (01)*

Formal Grammar

- **AMBIGUITY OF A GRAMMAR**

Simplified Definition (Regular Grammar):

- A grammar is ambiguous if there exist two or more distinct leftmost derivations (derivation trees) for a certain string.

Formal Grammar

Determine if the following grammar is ambiguous or not.

N = {A,B}

T = {0,1}

P:

Σ --> A

A --> B0

A --> A0

B --> B0

A --> 1

B --> 1

Formal Grammar

Determine if the following grammar is ambiguous or not.

N = {A}

T ={0,1}

P:

Σ --> A

A --> A0A

A --> 1

Formal Grammar

Exercise:

Determine which types of grammar hold for the grammars described in the previous two slides.

Show that a=b*a+b/c-a is a member/element of the language of the following grammar

- N = {L,R,O}
- T ={+,*,/,-,=,a,b,c}
- P:

$\Sigma \rightarrow L = R$	$R \rightarrow a$	
$L \rightarrow a$	$R \rightarrow b$	$O \rightarrow *$
$L \rightarrow b$	$R \rightarrow c$	$O \rightarrow /$
$L \rightarrow c$	$O \rightarrow +$	
$R \rightarrow ROR$	$O \rightarrow -$	

Backus-Naur Form and Context-free Grammar

- The Backus-Naur Form (BNF), named after John Backus, who invented it, and Peter Naur, who refined it for the specification of the programming language ALGOL, is essentially a context-free grammar(CFG). Instead of adapting the conventions on the CFG as presented in Chomsky's Heirarchy of Grammars, the following modifications are used for the BNF.
- 1. Instead of using the symbol $\rightarrow$ in a production, the symbol ::= is used.
- 2. Nonterminal symbols are enclosed by angle brackets, < >.
- 3. Instead of listing separately the production rules with similar left hand sides, they are combined into one statement.The right-hand sides of the productions that are combined into one statement are separated by bars. For example, the productions $A \rightarrow Aa$, $A \rightarrow a$ and $A \rightarrow AB$ is represented by the single production

<A> ::= <A>a | a | <A><B> in BNF.

Exercise
Backus-Naur Form

Required: Construct a Backus-Naur form of a grammar for the production of signed integers. (A signed integer is a nonnegative integer preceded by a plus sign or minus sign). Let N={R,S,I,D} where R represents signed integers, S represents a sign(i.e. +, -), I represents an integer and D represents a numeral digit(i.e. 0,1,...,9)