LECTURE 2

Types of Grammars

Introduction

Different constraints on Production rules define different classes of Grammars and Languages.

1. Chomsky Classification for Grammars:

The concept of grammar classification was introduced by Chomsky in the 1950s. Grammars can be classified according to the types of productions that are allowed into the following types:

- **Type 0 grammar (Unrestricted Grammar):**
  it's the grammar that has no restrictions on its productions.
- **Type 1 grammar (Context-Sensitive Grammar):**
  A grammar is context-sensitive if all its production rules are of the form $\alpha A\beta \rightarrow \alpha\gamma\beta$
  - A is a single nonterminal.
  - γ is a nonempty string of nonterminals and terminals.
  - α and β may be empty or string of nonterminals and terminals.
  - $S \rightarrow \varepsilon$ is allowed if S is NOT on the right side of any rule.
  - The restriction here is that |left side| ≤ |right side|.
- **Type 2 grammar (Context-Free Grammar):**
  A grammar is context-free if all its production rules are of the form $A \rightarrow \beta$
  - A is a single nonterminal on the left-hand side.
  - β is a string of terminals and nonterminals, or ε on the right-hand side
- **Type 3 grammar (Regular Grammar):**
  A grammar is regular if all its production rules are of the form: $A \rightarrow aB$, or $A \rightarrow a$
  - A is a single nonterminal on the left-hand side.
  - on the right-hand side a is a set or single terminal, possibly followed (or preceded, but not both) by B which is a single nonterminal.
  - $S \rightarrow \varepsilon$ is allowed if S is NOT on the right side of any rule.

  Regular Grammar could be
  Right-regular Grammar: $A \rightarrow aB$, or $A \rightarrow a$
  Left- regular grammar: $A \rightarrow Ba$, or $A \rightarrow a$

Example 1:

Determine the type of the following grammars:

1. G= ({S,A,B, 0, 1, 2}, {0, 1, 2}, S, P)
   P = {
   S → 0AB
   S → ε
   BA → AB
   0A → 01
   1A → 11
   1B → 12
   2B → 22
   }
2. G = ({S, a, b}, {a, b}, S, P) where
   P = {
   S → aSb
   S → ε
   }
3. G = ({S, 0, 1}, {0, 1}, S, P) where
   P = {
   S → S10
   S → 0
   }
4. G=({S, A, C, a, b, c}, {a, b, c}, S, P) where
   P={
   S → aAbc | abc
   A → aAbC | abC
   Cb → bC
   Cc → cc
   }
5. G=({S, A, B, a, b}, {a, b}, S, P) where
   P={
   S → AB
   B → aAb
   aAb → b
   }
6. G = ({S, X, Y, x, y}, {x, y}, S, P)
   P = {
   S → xX|yY
   X → x
   Y → y
   S → ε
   }

Solution:

1. Context-sensitive grammar
   There are many symbols on the left-hand side
   |left side| ≤ |right side|,
   S → ε and S does not appear on the right side of any rule.

2. Context-free grammar
   There is a single symbol on the left-hand side and string of terminal and nonterminal on the right-hand side

3. Regular grammar.
   There is a single symbol on the left-hand side and single nonterminal followed by set terminals set on the right-hand side.

4. Context-sensitive grammar
   There are many symbols on the left-hand side
   |left side| ≤ |right side|

5. unrestricted grammar
   There are many symbols on the left but there is a rule in which
   |left side| > |right side|

6. Regular grammar.
   There is a single symbol on the left-hand side and single nonterminal preceded by set of terminals on the right -hand side.

2. Parsing:

Given a formal grammar and a string produced by that grammar, parsing is figuring out the production process for that string. There are two basic types of parsing.

1. **Top-down parsing**: it begins with the starting symbol and proceeds by applying production rule.
2. **Bottom-up Parsing:** it begins from the string backward to the starting symbol.

Example 2:

Determine whether the word cbab belongs to the language generated by the grammar G =(V,T,S,P), where V ={a, b, c, A, B, C, S}, T ={a, b, c}, and the production rules are:

S → AB

A → Ca

B → Ba

B → Cb

B → b

C → cb

C → b.

Solution:

Top-down parsing: S→ AB →CaB →cbaB →cbab.

Bottom-up parsing: cbab→cbaB→CaB→AB→S.

So, the string cbab belongs to L(G).

3. Derivation Tree (Parse Tree):

In the study of grammars, trees are often used to show the derivation of sentences from production rules visually.

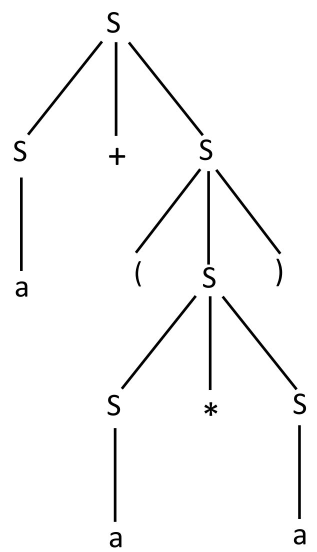
The trees that represent a derivation of a string within a language are called derivation trees or parse trees. In the derivation tree:

- The root of the derivation tree represents the starting symbol.
- The internal vertices of the tree represent the nonterminal symbols.
- The leaves of the tree represent the terminal symbols.
- Each yields-step (production rule) represents an edge.
- The final result can be seen by reading the leaves left-to-right.

Example 3:

Let G= ({S, a, +, *, (,) }, { a, +, *, (,) }, S, P) where P={ S →a | S + S | S ∗ S | (S) }. Construct derivation trees for string a + (a * a)

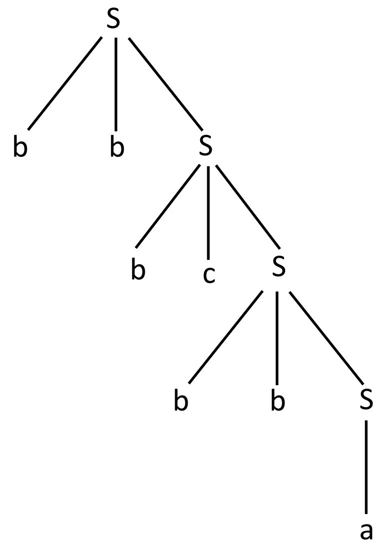
Solution:



Example 4:

Let G= ({a, b, c, S}, {a,b, c}, S, P) where P={S → abS | bcS | bbS |a | cb }.
Construct derivation trees for string bbbcbba.

Solution:



4. Homework:

HW 1:

Determine whether each of the following strings belongs to the language generated by the grammar in example 4.

1. abab / by using top-down parsing.
2. cbaba / by using bottom-up parsing

HW 2:

Let G is the grammar in example 4. Construct derivation trees for string bcabbbbbcb.

HW 3:

Let G=({S,A,B,a,b}, {a,b}, S, P). Determine the type of the following grammar if P, the set of productions, is:

1. S → ABa, AB → a.
2. S → bA, A → b, S → λ.