Lecture 5

Ambiguity – 1/5

Ambiguity implies the possibility of **different interpretation of a source string**

In natural languages, ambiguity may concern the meaning or syntax category of a word, or syntactic structure of a construct

Eg. A word can have multiple meanings or can be both noun and verb, and a sentence can have multiple syntactic structures

Ambiguity – 2/5

Formal language grammars avoid ambiguity at the level of a lexical unit or a syntax category

This is achieved by the **simple rule that identical strings cannot appear on the RHS of more than one production in the grammar**

Existence of ambiguity at the level of the syntactic structure of a string would mean that more parse tree can be built for the string

Ambiguity – 3/5

Example

E → E + E | E * E | id

String is **id + id * id**

Ambiguous grammar since we will have two parse trees

Ambiguity – 4/5

First parse tree

E → E + E
→ E + E * E
→ id + E * E
→ id + id * E
→ id + id * id

Ambiguity – 5/5

Second parse tree

E → E * E
→ E * E + E
→ id * E + E
→ id * id + E
→ id * id + id

Eliminating ambiguity – 1/2

An ambiguous grammar should be rewritten to eliminate ambiguity

E → E + T | T
T → T * F | F
F → id

Enforce precedence of * over +

Eliminating ambiguity – 2/2

E → E + T
→ T + T
→ F + T
→ id + T
→ id + T * F
→ id + F * F
→ id + id * F
→ id + id * id

Examples

Find out if these grammars are ambiguous or not. If ambiguous then convert them into unambiguous grammar

1. E → E + E | id
2. E → E or E | E and E | not E | True | False
3. E → E + E | E * E | E ^ E | id