Concepts of Programming Languages, CSCI 305, Fall 2021
Precedence & Associatively in Context-Free Grammars
Introduction to Parsing, Oct. 4

Precedence & Associatively in Context-Free Grammars

A grammar is ambiguous if there is more than one parse tree for a statement in the language. Avoid ambiguous grammars since they allow a statement to have multiple meanings (one for each parse tree).

To give an operator higher precedence, make it appear lower in the parse tree by defining non-terminal symbols in the grammar that force the lower precedence operator to occur before the higher one.

To make an operator left-associative, place the non-terminal that appears recursively to the left of the operator. Similarly, for right-associative, make the recursion on the right.

Parsing

Syntactic analysis (also called parsing)
Goals:

- detect syntax errors
- produce diagnostic messages and recover
- create parse tree or trace

It can be shown that given any context-free grammar, a parser can be created for the language which runs in $0(n^3)$ time, where n is the size of the program. With small restrictions to grammars, can get parsers that run in linear time