

Concepts of Programming Languages, CSCI 305, Fall 2021
Writing an LL(1) Grammar, Oct. 20

Section 2.3.2 Writing an LL(1) Grammar, pages 79-82

A grammar is not LL(1) if there are multiple production entries in the recursive descent parse table. (Equivalently, PREDICT sets for a variable overlap.)

Most common obstacles to LL(1) are:

- Left recursion - there is a variable A such that $A \Rightarrow^+ A \alpha$ for α any combination of terminals and variables.
- Common prefixes - two productions with the same left-hand side begin with the same symbol or symbols, either directly or indirectly

Example:

1. $\text{id_list} \rightarrow \text{list ;}$
2. $\text{list} \rightarrow \text{list , id}$
3. $\text{list} \rightarrow \text{id}$

Solution to left recursion: Replace the left-recursive definition by a right-recursive variant, moving terminals directly after the arrow and adding a new variable (tail).

A grammar has common prefixes when two different productions with the same left-hand side begin with the same symbol or symbols, either directly or indirectly.

Solution to common prefixes: Factor out the common symbol