

Syntax Analysis

Sudakshina Dutta

IIT Goa

9th February, 2022

Ambiguity

There are two ways to resolve ambiguity

- ▶ Developer's intervention
- ▶ Other ways

Problem faced in designing parser

- ▶ One problem is that we are unable to choose production rules
- ▶ We apply left factoring for this

In general..

► $A \rightarrow \alpha\beta_1 | \alpha\beta_2$

A transformation is applied to immediately say which production is to be applied

► $A \rightarrow \alpha A'$

► $A' \rightarrow \beta_1 | \beta_2$

This grammar transformation is called left factoring

Ambiguity

Consider the following grammar

$$\blacktriangleright E \rightarrow E + E$$

$$\blacktriangleright E \rightarrow E * E$$

$$\blacktriangleright E \rightarrow -E$$

$$\blacktriangleright E \rightarrow (E)$$

$$\blacktriangleright E \rightarrow id$$

Lets take the word : $id + id * id$

With this grammar, Using LEFT derivation at every step, We could construct more than 1 parse Trees. So this grammar having ambiguity. Note that we are using different production rules at different positions to construct the parse trees that give the required string.

The same thing holds even for RIGHT DERIVATION at each step.

After left factoring, it is the following

$$\blacktriangleright E \rightarrow EE'$$

$$\blacktriangleright E' \rightarrow +E \mid * E$$

$$\blacktriangleright E \rightarrow -E$$

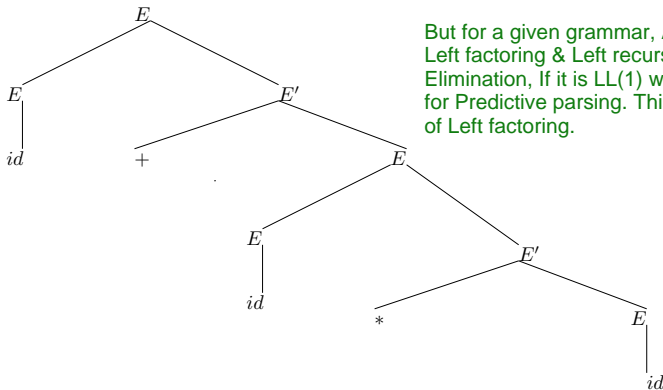
$$\blacktriangleright E \rightarrow (E)$$

$$\blacktriangleright E \rightarrow id$$

Disambiguation by left factoring

Consider the string

► $id + id * id$



Here also it is still disambiguous since we are able to draw 2 parse trees.

It seems Left factoring & Disambiguation are not having much relation.

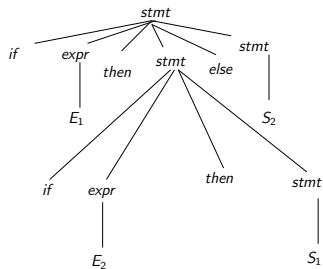
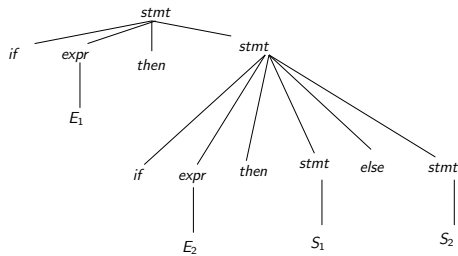
But for a given grammar, After doing Left factoring & Left recursion Elimination, If it is LL(1) we can use it for Predictive parsing. This is the use of Left factoring.

Ambiguity

Consider the following grammar

- ▶ $stmt \rightarrow \text{if } expr \text{ then } stmt$
- ▶ $stmt \rightarrow \text{if } expr \text{ then } stmt \text{ else } stmt$
- ▶ $stmt \rightarrow other$

Ambiguity



Manual intervention : Use some rule to disambiguate

Disambiguation by Left Factoring

The given grammar after left factoring

- ▶ $stmt \rightarrow if\ expr\ then\ stmt\ stmt'$
- ▶ $stmt' \rightarrow \epsilon \mid else\ stmt$
- ▶ $stmt \rightarrow other$

Disambiguation by Left Factoring

To check, We need to see if it is possible to construct more than 1 parse trees for the given word using only LEFT/RIGHT derivations from the grammar.

Check whether left factoring could disambiguate using the following string

► *if E_1 then if E_2 then S_1 else S_2*

It is still disambiguous since we constructed 2 parse trees.