

Using ambiguous grammars

No ambiguous grammar is LR. Nonetheless, ambiguous grammars are often used in LR parsing.

Why? Because ambiguous grammars are often simpler.

For instance, compare

$$E \rightarrow E + E \mid E * E \mid (E) \mid a$$

to the equivalent unambiguous grammar

$$\begin{aligned} E &\rightarrow E + T \mid T \\ T &\rightarrow T * F \mid F \\ F &\rightarrow (E) \mid a \end{aligned}$$

The second grammar has important virtues though:

1. It is unambiguous.
2. It reflects the fact that operators $+$ and $*$ are left-associative.
3. It reflects the fact that $*$ has higher precedence than $+$.

$$E \rightarrow E + E \mid E * E \mid (E) \mid a$$

$$E \rightarrow E + T \mid T$$

$$T \rightarrow T * F \mid F$$

$$F \rightarrow (E) \mid a$$

In practice, we can obtain the same advantages for the first grammar.

In particular, we can stipulate — as a guide to using the first grammar — that $*$ and $+$ are left-associative and that $*$ has precedence over $+$.

And if we consider only parse trees that reflect these stipulations, the grammar is rendered unambiguous “in practice.” (Moreover, the language generated by the grammar is unchanged!)

That is, for every sentence generated by the grammar, there is exactly one parse tree that respects the stipulations.

Consider alternative parse trees for $a + a + a$ and $a + a * a$, for example.

There are two parse trees for each, only one of which satisfies the stipulations.

“Dangling else” ambiguity

When parsing if-then-else constructs, it is customary to associate each **else** with the closest preceding unmatched **then**.

$$\begin{aligned} stmt &\rightarrow \text{if expr then } stmt \\ &\quad | \text{ if expr then } stmt \text{ else } stmt \\ &\quad | \text{ other} \end{aligned}$$

Consider the two parse trees for

if expr then if expr then other else other

The preference for matching the closest unmatched **then** is a bit awkward to express directly in a grammar:

$$\begin{aligned} stmt &\rightarrow matched-stmt \mid unmatched-stmt \\ matched-stmt &\rightarrow \text{if expr then } matched-stmt \text{ else } matched-stmt \\ &\quad | \text{ other} \\ unmatched-stmt &\rightarrow \text{if expr then } stmt \\ &\quad | \text{ if expr then } matched-stmt \text{ else } unmatched-stmt \end{aligned}$$

This grammar is SLR, I believe, but again we can instead can work directly with the ambiguous grammar...

$$\begin{aligned} stmt &\rightarrow \text{if expr then } stmt \\ &\quad | \text{ if expr then } stmt \text{ else } stmt \\ &\quad | \text{ other} \end{aligned}$$

First simplify the representation:

$$S \rightarrow iS \mid iSeS \mid o$$

An example of the problem with ambiguity in the simplified grammar:

$$S \Rightarrow iSeS \Rightarrow iSeo \Rightarrow iiSeo \Rightarrow iioeo$$

$$S \Rightarrow iS \Rightarrow iiSeS \Rightarrow iiSeo \Rightarrow iioeo$$

Which of these rightmost derivations do we prefer?