Def: Given a CFG $G = (V, \Sigma, R, S)$, a parse tree for a string $x \in L(G)$ has:
- Root labeled $S$
- Leaves labeled with terminals or $E$
- Interior nodes have labels $A \in V$

If its children are $U_1, \ldots, U_m$, then $A \rightarrow U_1 \ldots U_m$ is a rule in $R$.

Example:

$\Sigma = \{+, -, *, /, (, )^3, \text{digits and alphanumeric}\}$

$E \rightarrow \langle \text{const} \rangle | \langle \text{var} \rangle | E + E | E - E | E \times E | E / E | (E)$

Example: $a - b \times c$

$E \rightarrow E \times E \Rightarrow E - E \times E$
$\Rightarrow a - E \times E$
$\Rightarrow a - b \times E$
$\Rightarrow a - b \times c$.
*Nier*

**Parse:**

Follow rules of precedence.

**Fact:** Parse trees are in 1-1 corresp with leftmost derivations.

- Any derivation $\hat{d}$ gives a unique parse tree $T$.
- Any $T$ gives a unique leftmost derivation $\hat{d}$.

**Defn:** A string $x \in L(G)$ is unambiguous in $G$ if it has a unique parse tree, equivalently if it has a unique leftmost derivation in $G$.

**Defn:** $G$ is unambiguous if every $x \in L(G)$ is unambiguous.

$G' = \{E \rightarrow (E + E) \mid (E - E) \mid (E \times E) \mid (E \div E) \mid E\}$ is unambiguous.
Different trees, but also different final strings.
Fact: $G'$ is unambiguous.

Can we design a CFG $G''$ such that $L(G'')$ includes formulas such as $a - b * c$ in a way we naturally write them, but is unambiguous?

A: Yes, but we need more variables, i.e., more "Syntactic Categories."

Expression E, Term T, Factor F

$$E \rightarrow T \mid E + T \mid E - T$$
$$T \rightarrow F \mid T * F \mid T / F$$
$$F \rightarrow \text{const} \mid \text{variable} \mid (E)$$

The yields "a - b * c" and "(a - b) * c" are different strings. Hence this "ETF Grammar" avoids the ambiguity.

Fact (not proved in text either)
This $G''$ is unambiguous.
Q: What is \(30/2\times3\) ?

\[
\begin{align*}
E \rightarrow T & \rightarrow T/F \rightarrow F/F \rightarrow 30/F \\
& \rightarrow 30/(E) \rightarrow 30/(T) \rightarrow 30/(T\times F) \\
& \rightarrow 30/(F\times F) \rightarrow 30/(2\times F) \rightarrow 30/(2\times 3).
\end{align*}
\]

This yielded a different string = how we should have grouped it, if we want \(5\).

\[
\begin{align*}
E \rightarrow T & \rightarrow T\times F \rightarrow T/F\times F \\
& \rightarrow F/F\times F \rightarrow 30/F\times F \rightarrow 30/2\times F \rightarrow 30/2\times 3
\end{align*}
\]

What about \(a\times b\times c\) ? = \(a^{b\times c}\) Group as \((a\times b)^c\) or \(a^{(b\times c)}\) ?

Because \((a\times b)^c = a^{(b\times c)}\)

However, this means "**" operator must have right precedence, hence any CEG giving it must use a syntactic category "between" between \(T\) and \(F\) for it.

So complicated that many langs skip giving a ** operator.

Note also: Function composition associates to the right: \(f\circ g\circ h = f\circ (g\circ h)\).

An often- tolerated ambiguity: the "Dangling If-Else". If \(b > 0\) then \(x = d\in L(3)\), else \(x = \star\text{stmt1}\).

\(G\) is ambiguous but often used in the above form.