CFGs continued.

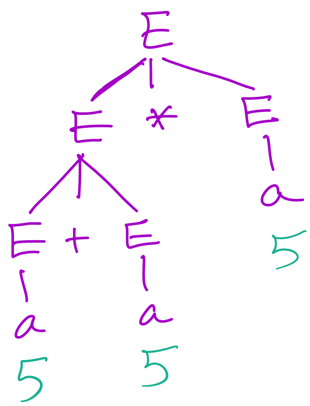
We often give grammars just as a set of rules — variables appear on LHS, terminals appear only on RHS. Start variable must be identified.

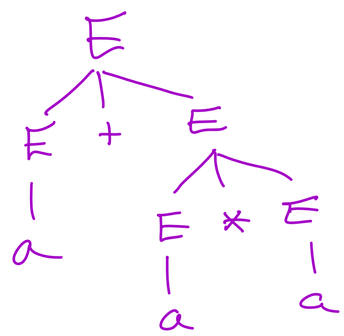
$$\underline{E} \to E+E \mid E*E \mid a$$

$a+a*a$

$a*a*a+a$

$a+a*a$





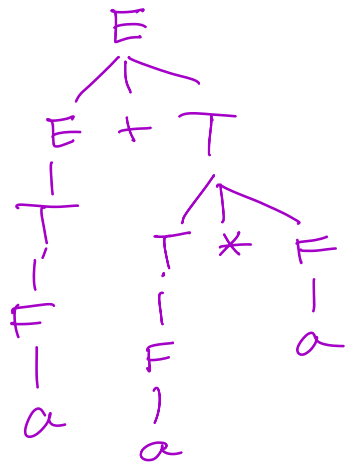
$a+a*a$ is ambiguous.

Def 2.7 A string w is derived ambiguously in CFG G if it has 2 or more different leftmost derivations (derivation tree).
G is ambiguous if it generates some string ambiguously.

Sometimes we can remove the ambiguity — find a new grammar that generates the same language but is not ambiguous.

$$E \to E + T \mid T$$
$$T \to T * F \mid F$$
$$F \to a$$

$a + a * a$



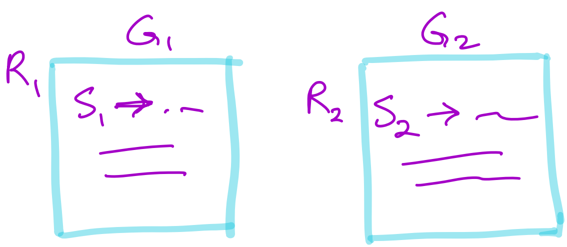
Can every ambiguous CFG be disambiguated?

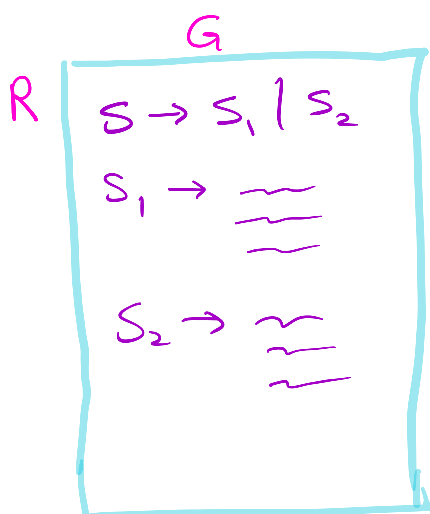
Def^{n}: A language L is $\boxed{\text{Context-Free}}$ if $\exists$ a CFG for L.

Designing CFGs.

Theorem: The class CFL is closed under $\cup$.

Proof: by construction. For any grammars G_1 and G_2.





construct a grammar G s.t.

$$L(G) = L(G_1) \cup L(G_2)$$

$\blacksquare$

Eg. G_1: $\boxed{S_1 \to aS_1b \mid \varepsilon}$ G_2: $\boxed{S_2 \to bS_2a \mid \varepsilon}$

$V_1 \cap V_2' = \emptyset$

G:

$$
\boxed{
\begin{aligned}
S &\to S_1 \\
S &\to S_2 \\
S_1 &\to aS_1b \\
S_1 &\to \varepsilon \\
S_2 &\to bS_2b \\
S_2 &\to \varepsilon
\end{aligned}
}
$$

Note: The two grammars must not have variable-names in common - leads to trouble.

Rename variables if necessary.

Theorem: The Class CFL is closed under concat.

Proof: Exercise

Some instructive examples.

$L = \{a^i b^i c^k : i, j, k \geq 0\}$

Note: $L = A^n B^n \cdot L(c^*)$

$S \rightarrow$

$L = \{a^i b^k c^k : i, j, k \geq 0\}$

Note: $L = L(a^*) \cdot \underbrace{B^n C^n}_{\text{new language}}$

$S \rightarrow$

The union of these 2 languages is

$\{ a^i b^j c^k \mid \text{either } i = j \text{ or } j = k \}$

Defn Chomsky Normal Form (CNF)

A CFG is CNF if every rule is of the form

$A \to BC$ or

$A \to a$ where a is a terminal;

and where A, B, C are any variables, but B, C cannot be the start variable. Also we permit $S \to \varepsilon$, where S is the start variable.

Eg. $S \to \varepsilon \mid E$

$E \to aEb$

$E \to ab$

$\Downarrow$

$S \to \varepsilon \mid E$

Theorem: $\forall$ CFL has a CNF grammar that generates it.

How can we run into trouble if we take the union (or concat) but we don't have $V_1 \cap V_2 = \emptyset$?

$L_1 = \{a^i b^i \mid i \geq 0\}$

$L_2 = \{b^i a^i \mid i \geq 0\}$.

$S \to aSb \mid \varepsilon$

$S \to bSa \mid \varepsilon$.

$L_1 \cup L_2$ should be strings of form $a^i b^i$ and $b^i a^i$

But if we just combine the grammars, we get: