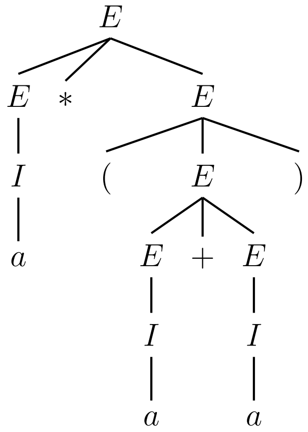
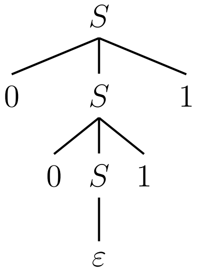
上下文无关文法

- 上下文无关文法
- 语法分析树
  - 形式定义
  - 语法树和派生的等价性
- 文法和语言的歧义性
- 文法的化简与范式

派生或归约的过程可以表示成树形结构.

- 例 2 文法 G_{exp} 中推导算数表达式 $a * (a + a)$ 的过程
- 例 6 中语言 L_{eq} 的文法中推导 0011 的过程





语法分析树的形式定义

定义

$CFG\ G = (V, T, P, S)$ 的语法分析树(语法树或派生树)为:

❶ 每个内节点标记为 V 中的变元符号;

❷ 每个叶节点标记为 $V \cup T \cup \{\varepsilon\}$ 中的符号;

❸ 如果某内节点标记是 A, 其子节点从左至右分别为

$$X_1,\ X_2,\ \cdots,\ X_n$$

那么

$$A \to X_1 X_2 \cdots X_n \in P,$$

若有 $X_i = \varepsilon$, 则 ε 是 A 唯一子节点, 且 $A \to \varepsilon \in P$.

定义

语法树的全部叶节点从左到右连接起来，称为该树的**产物**或结果. 如果树根节点是初始符号 S，叶节点是终结符或 ε，那么该树的产物属于 $\mathbf{L}(G)$.

定义

语法树中标记为 A 的内节点及其全部子孙节点构成的子树，称为 **A 子树**.

语法分析树和派生的等价性

定理 17

$CFG\ G = (V, T, P, S)$ 且 $A \in V$, 那么文法 G 中

$$A \xRightarrow{*} \alpha$$

当且仅当 G 中存在以 A 为根节点产物为 α 的语法树.

$$\Longleftrightarrow A \xRightarrow{*} \alpha$$

证明: [充分性] 对 $A \overset{j}{\Rightarrow} \alpha$ 的步骤数 j 归纳证明.

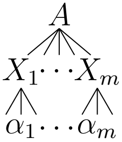
归纳基础: $j=1$ 时, $A \Rightarrow \alpha$, 有 $A \to \alpha \in P$, 可构造 $\begin{array}{c}A\\ \bigwedge\\ \alpha\end{array}$.

归纳递推: 假设 $j \le n$ 时命题成立. 当 $j = n+1$ 时, $A \overset{n+1}{\Longrightarrow} \alpha$ 的派生过程为

$j = 1 + n$

$$A \Rightarrow X_1 \cdots X_m \overset{n}{\Rightarrow} \alpha_1 \cdots \alpha_m = \alpha,$$

其中 $A \to X_1 \cdots X_m \in P$. 而 X_i 若非终结符, 一定有 $X_i \overset{*}{\Rightarrow} \alpha_i$ 且不超过 n 步, 由归纳假设存在 $\begin{array}{c}X_i\\ \bigwedge\\ \alpha_i\end{array}$, 因此可以构造以 A 为根, 以 X_i 为子树 (或叶子) 的语法树, 其产物刚好为 α.



[必要性] 对语法分析树的内节点数 j 归纳证明.

归纳基础: $j=1$ 时, 由 $\begin{array}{c}A\\ \wedge\\ \alpha\end{array}$, 有 $A \to \alpha \in P$, 那么 $A \overset{*}{\Rightarrow} \alpha$.

归纳递推: 假设 $j \le n$ 时命题成立. 当 $j = n+1$ 时, 根节点 A 的儿子为 $X_1, X_2, \dots, X_m$, 则

$$A \to X_1 \cdots X_m \in P, \text{ 且 } A \Rightarrow X_1 \cdots X_m.$$

而 X_i 子树 (或叶子) 内节点数都不超过 n, 由归纳假设有

$$X_i \overset{*}{\Rightarrow} \alpha_i,$$

从左至右连接 α_i, 刚好为树的产物 α, 所以有

$$X_1 X_2 \cdots X_m \overset{*}{\Rightarrow} \alpha_1 X_2 \cdots X_m \overset{*}{\Rightarrow} \cdots \overset{*}{\Rightarrow} \alpha_1 \alpha_2 \cdots \alpha_m = \alpha.$$

因此 $A \overset{*}{\Rightarrow} \alpha$ 命题成立. $\square$

语法树唯一确定最左 (右) 派生

- 每棵语法分析树都有唯一的最左 (右) 派生
- 给定 CFG $G = (V, T, P, S)$, $A \in V$, 以下命题等价:

❶ 通过递归推理, 确定串 w 在变元 A 的语言中.

$A \overset{*}{\Rightarrow} w$

❷ 存在以 A 为根节点, 产物为 w 的语法分析树.

❸ $A \overset{*}{\Rightarrow} w$.

❹ $A \underset{\text{lm}}{\overset{*}{\Rightarrow}} w$.

❺ $A \underset{\text{rm}}{\overset{*}{\Rightarrow}} w$.

上下文无关文法

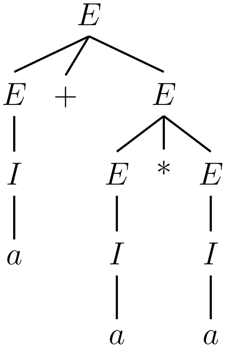
- 上下文无关文法
- 语法分析树
- 文法和语言的歧义性
  - 文法歧义性的消除
  - 语言的固有歧义性
- 文法的化简与范式

文法的歧义性

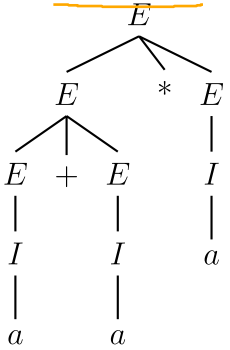
定义

如果 CFG G 使某些符号串有两棵不同的语法分析树，则称文法 G 是**歧义的**.

续例 2. 算数表达式的文法 G_{exp} 中，对句型 $a+a*a$ 有下面两棵语法分析树:



语义



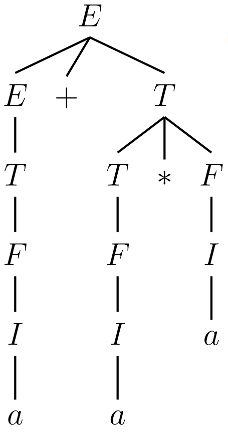
$$
\begin{aligned}
(1)\ E &\Rightarrow E+E \\
&\Rightarrow E+E*E \\
&\stackrel{*}{\Rightarrow} a+a*a
\end{aligned}
\qquad\qquad
\begin{aligned}
(2)\ E &\Rightarrow E*E \\
&\Rightarrow E+E*E \\
&\stackrel{*}{\Rightarrow} a+a*a
\end{aligned}
$$

文法歧义性的消除

有些文法的歧义性，可以通过重新设计文法来消除.

续例 2. 文法 G_{exp} 重新设计为文法 $G_{\text{exp}'}$ 可消除歧义.

$$
\begin{aligned}
E &\to E + T \mid T \\
T &\to T * F \mid F \\
F &\to (E) \mid I \\
I &\to a \\
I &\to b \\
I &\to Ia \\
I &\to Ib \\
I &\to I0 \\
I &\to I1
\end{aligned}
$$



a + a*a

语言的固有歧义性

定义

定义同样的语言可以有多个文法，如果 CFL L 的所有文法都是歧义的，那么称语言 L 是固有歧义的.

- 固有歧义的语言确实存在，如语言

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

  中任何形为 $a^n b^n c^n$ 的串，总会有两棵语法树.

- “判定任何给定 CFG G 是否歧义”是一个不可判定问题.