

# Compiler 2

孔静 2014K8009929022

September 24, 2017

## 1 2.2.1

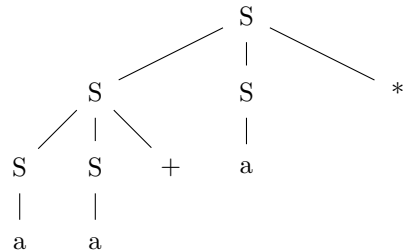
考虑下面的上下文无关文法：

$$S \rightarrow SS + | SS * | a$$

1.1 试说明如何使用文法生成串  $aa+ a^*$

$$S \rightarrow SS* \rightarrow SS + S* \rightarrow aS + S* \rightarrow aa + S* \rightarrow aa + a*$$

1.2 试为这个串构造一颗语法分析树



1.3 该文法生成的语言是什么？

视  $a$  为运算数， $L =$  支持加法和乘法的表达式的后缀表示形式

1.4 该文法具有二义性吗？为什么？

不存在，因为存在唯一倒推，从最右侧终止符开始。

## 2 2.2.5

2.1 证明：用下面文法生成的所有二进制串都能被 3 整除（提示：数学归纳法）

$$num \rightarrow 11 | 1101 | num\ 0 | num\ num$$

显然，因为出现的终止符号，11,1001,0 都能被 3 整除，而 num0 和 num num 能否被 3 整除，即看 num，递推下去即终止符能否被整除。

$$\begin{aligned} &= \sum_n (2^1 + 2^0) * 2^n + \sum_m (2^3 + 2^0) * 2^m \\ &= \sum_n 3 * 2^n + \sum_m 9 * 2^m \end{aligned}$$

显然能被 3 整除。

2.2 上面的文法是否能够生成所有能被 3 整除的二进制串？

显然不行，比如 100001,10000001.....

## 3 2.3.1

构建一个语法制导翻译方案，该方案把算术表达式从中缀表示方式翻译为运算符在运算分量之前的前缀表示方式。例如，-xy 是表达式 x-y 的前缀表示。给出输入 9-5+2 和 9-5\*2 的注释分析树。

产生式：

$$\begin{aligned} expr &\rightarrow expr + term \\ &\quad | expr - term \\ &\quad | term \\ term &\rightarrow term * factor \\ &\quad | term / factor \\ &\quad | factor \\ factor &\rightarrow digit | (expr) \end{aligned}$$

翻译方案：

$$\begin{aligned}
expr &\rightarrow \{print(' +')\} expr + term \\
&\quad | \{print(' -')\} expr - term \\
&\quad | term \\
term &\rightarrow \{print(' *')\} term * factor \\
&\quad | \{print(' /')\} term / factor \\
&\quad | factor \\
factor &\rightarrow digit \{print(digit)\} \\
&\quad | (expr)
\end{aligned}$$

注释分析树：

