编译原理作业 4

23.09.27

练习 2.2.1

(1) 练习 4.2.1 的 (1) 或 (2) 的推导方式都可以生成该串.

(2) 同练习 4.2.1 (3).

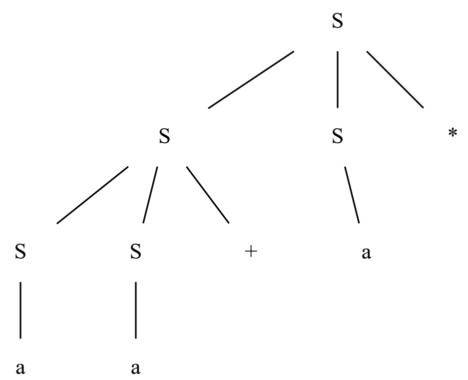
(3) 所有运算符为 $+$ 或 $*$, 操作数为 a 的后缀表达式. 所有这样的表达式都能由该文法生成, 且该文法生成的所有串都是这样的表达式, 两者的证明都是显然的.

练习 4.2.1

(1) $S \underset{lm}{\Rightarrow} SS* \underset{lm}{\Rightarrow} SS+S* \underset{lm}{\Rightarrow} aS+S* \underset{lm}{\Rightarrow} aa+S* \underset{lm}{\Rightarrow} aa+a*$.

(2) $S \underset{rm}{\Rightarrow} SS* \underset{rm}{\Rightarrow} Sa* \underset{rm}{\Rightarrow} SS+a* \underset{rm}{\Rightarrow} Sa+a* \underset{rm}{\Rightarrow} aa+a*$.

(3)



(4) 该文法无二义性. 我们可以通过查看末尾字符, 确定所使用的替换规则, 因此显然语言中的每个串都对应着唯一的 AST.

(5) 所有运算符为 $+$ 或 $*$, 操作数为 a 的后缀表达式.

练习 4.2.3

(1)

$$\begin{aligned} S &\to AB \\ A &\to \epsilon | 1 | AA \\ B &\to \epsilon | C | BB \\ C &\to 01A \end{aligned}$$

A 代表若干个 1. "若干"可以是零个, 后面同.

C 代表前缀为 01, 后面跟着若干个 1 的串.

B 代表若干个 C.

(2)

$$S \to \epsilon|1|0|1S1|0S0$$

(3)

$$S \to \epsilon|SS|0S1|1S0$$

S 代表 0 和 1 个数相等的串. 注意到如果 S 的始末字符相同, 例如 S 形如 $0x0$, 由于 0 中 0 比 1 多一个, $0x$ 中 0 比 1 少一个, 根据介值原理, 必存在 $x = yz$, $0y$ 和 $z0$ 中都有等量 0 和 1, 因此 S 必形如 SS. 而如果始末字符不同, 中间的部分就也是 S 了.

(4)

$$\begin{aligned} S &\to S_1|S_2 \\ S_1 &\to A_1|A_1S_1 \\ A_1 &\to B0B \\ S_2 &\to A_2|A_2S_2 \\ A_2 &\to B1B \\ B &\to \epsilon|BB|0B1|1B0 \end{aligned}$$

S_1 代表 0 比 1 多的串.

A_1 代表 0 比 1 多一个的串. 根据介值原理, 容易证明 S_1 一定可以拆成一个或多个 A_1.

B 代表 0 和 1 个数相等的串. 如果 A_1 的首字符是 0, 那么它形如 $0B$; 如果 A_1 的首字符是 1, 容易用介值原理证明它必然形如 $B0B$.

同理, S_2 代表 0 比 1 少的串, A_2 代表 0 比 1 少一个的串.

(5)

$$\begin{aligned} S &\to AB \\ A &\to \epsilon|1|AA \\ B &\to \epsilon|C|BB \\ C &\to 0|01 \end{aligned}$$

A 代表若干个 1. C 代表 0 或 01. B 代表若干个 C.

(6)

$$\begin{aligned} S &\to AB|BA \\ A &\to TAT|1 \\ B &\to TBT|0 \\ T &\to 0|1 \end{aligned}$$

注意到, 对于一个合法的串, 例如形如 $a0bc1d$, 其中 $|a| = |c| = n_1$, $|b| = |d| = n_2$, 可以将 bc 视为 xy, 其中 $|x| = n_1$, $|y| = n_2$. 这样 a 与 x 等长, y 与 d 等长, 就容易构造文法生成 $a0xy1d$ 了.