Contents:

1 无关文法

2 语法分析树

3 歧义

4 文法的化简和范式

1 上下文无关文法

考虑前面学过的, $\{0^n1^n \mid n \in \mathbb{N}\}$ 不是正则表达式的语言.

Example 1.1 (回文)**.** *if* $w \in \Sigma^*, w = w^R$, *then* w *为回文*

Example 1.2 (回文语言)**.** *if* $L = \{w \in \Sigma^* \mid w = w^R\}$ *then* L *是回文语言.*

运用前面的知识, 可以使用 Pump lemma 证明 L 不是正则的. 可是那么我们该如何表示该语言呢?

可以使用递归方法定义:

1. 首先 $\epsilon, 0, 1$ 都是回文.

2. if w 是回文, 那么 $0w0, 1w1$ 都是回文.

数理逻辑之中, 命题的递归定义也是类似的, 这种生成语句的规则就是上下文无关文法.

$$\left.\begin{aligned}&1.A \to \epsilon\\&2.A \to 0\\&3.A \to 1\\&4.A \to 0A0\\&5.A \to 1A1\end{aligned}\right\}\tag{1}$$

定义 1.3 (文法)**.** 文法 G 是一个 $4p$ 结构, $G = (V, T, P, S)$. 其中 V for variable, T for terminator, P for Production, S for start.

1 V 是变量的集合.

2 T 称为 terminators.

3 P 是生产规则, 形式为 $A \to \alpha \mid \beta$, 其中 α 是 ϵ 或者是 $\alpha \in T$, 或者是 $\alpha \in E$, 或者是 $\alpha \in (V \cup T)^*$, β 也是如此.

4 $S \in V$, 表示的是开始的变量, 所有句型, 语句都是从 S 开始派生的.

Example 1.4 (0,1 组成的回文)**.** $G = (V, T, P, S)$, 其中 $V = \{A\}$, $T = \{0, 1\}$, $P = A \to \epsilon \mid 0 \mid 1 \mid 0A0 \mid 1A1$, $S = A$

Example 1.5 (0^n1^n)**.** $L = \{0^n1^n \mid n \in \mathbb{N}\}$ 和上面一个完全类似, 这里就懒得说了

1.1 归约和派生

定义 1.6 (文法派生的句型)**.** $\alpha \in (V \cup T)^*$, α 可能是派生出的语句. 因为考虑到 $A \to t$, $t \in T$, 或者是 $A \to At$. 也就是说, 这个语句里面只会有 variable or terminator.

定义 1.7 (归约)**.**

定义 1.8 (派生)**.** $\alpha, \beta, \gamma \in (V \cup T)^*$, if $A \to \gamma$, then $\alpha A \beta \underset{G}{\Rightarrow} \alpha\gamma\beta$.

定义 1.9 (多步派生)**.**

$$\alpha_1 \overset{i}{\underset{G}{\Rightarrow}} \alpha_{1+i} \tag{2}$$

对于一般的多步派生, 记为 $\overset{*}{\underset{G}{\Rightarrow}}$

定义 1.10 (最左派生, 最右派生)**.** 对于一个句型 α, 我们考虑只对最左 (右) 的变量进行派生, 这就是最左 (右) 派生. 记为 $\underset{lm}{\Rightarrow}$ $(\underset{rm}{\Rightarrow})$

定理 1.11 (派生的等价性)**.** 对于任意一个派生, 都存在一个最左 (右) 派生与其对应

1.2 文法的语言

定义 1.12 (语言)**.**

$$L(G) = \{w \mid w \in T^*, S \xRightarrow[G]{*} w\} \tag{3}$$

Remark 1.13 (为什么称为是上下文无关文法). 我们参考 $\alpha A\beta \underset{G}{\Rightarrow} \alpha\gamma\beta$, 这个派生是否成立, 是跟 α, β 无关的, 于是称为上下文无关文法.

定义 1.14 (等价性)**.** G_1, G_2 满足 $L(G_1) = L(G_2)$, 则 G_1 等价于 G_2

定义 1.15 (句型, 左句型, 右句型)**.** G 生成的句型定义如下:

$$\{w \mid w \in (V \cup T)^*, S \xRightarrow[G]{*} w\} \tag{4}$$

也定义了左句型, 右句型.

Example 1.16. $L = \{a^n b^n \mid n \geq 1\}$

Example 1.17. $G = \{\}$

Example 1.18. $L = \{w \mid 0,1 \text{ 数量相等}\}$, $G = (V, T, P, S)$, *其中* $V = \{S\}, T = \{0,1\}$, P *是下面这个:*

$$S \to 0S_1 1S_2 \mid 1S_1 0S_2 \mid \epsilon$$

Example 1.19 (算术表达式)**.**

2 语法分析树