

1 两种范式

范式有两种范式, 一种是乔姆斯基范式 (CNF), 另一种是格雷巴赫范式 (GNF).

定义 1.1 (CNF). 每一个不包含 ϵ 的 G 的产生式都能够写为下面的形式:

$$A \rightarrow BC \quad A \rightarrow a$$

其中 A, B, C 是变量, 然后 a 是终结符.

证明. 能够知道这个定理实际上是显然的. 对于 G 的一个产生式, 其写为

$$A \rightarrow X_1 X_2 X_3 \dots X_m$$

其中 $X_i \in V \cup \Sigma$. 这个时候, 我们将其中的终结符进行这样的处理, X_i 换为 V_i , V_i 是新引入的变量. 并且引入产生式 $V_i \rightarrow X_i$. 这就是将一个表达式 $A \rightarrow X_1 \dots X_m$ 之中的终结符处理掉了.

我们将表达式 $A \rightarrow X_i \dots$ 写为 $A \rightarrow V_1 \dots V_m$, V_i 是变量. 我们再次引入变量, 将 $A \rightarrow V_1 \dots V_m$, 拆解为多个产生式, 就是 $A \rightarrow V_1 B_1$, $B_1 \rightarrow V_2 B_2 \dots$ 这样我们就能够将这个产生式拆分为多个满足乔姆斯基范式的形式产生式 □

Example 1.2. 给定 $CFG^1 G$, P 为

$$S \rightarrow bA \mid aB, \quad A \rightarrow bAA \mid aS \mid a, \quad B \rightarrow aBB \mid bS \mid b$$

流程为

1. 对于产生式之中有终结符的, 进行变量的替换, 比如说 $A \rightarrow bA$ 我们引入新的变量 C_b , 将这个式子换为

$$A \rightarrow C_b A, \quad C_b \rightarrow b$$

¹CFG stands for context free grammar. 只是有点忘了

2. 对于形如 $A \rightarrow B_1 B_2 \dots B_m$ 的式子, 引入中间变量.

Example 1.3. 将 G 的产生式进行处理, 使得其为乔姆斯基范式.

$$S \rightarrow AS \mid BABC \quad A \rightarrow A1 \mid 0A1 \mid 01 \quad B \rightarrow 0B \mid 0 \quad C \rightarrow 1C \mid 1$$

定义 1.4 (格雷巴赫范式). 每个产生式的形式为 $A \rightarrow a\alpha$ 其中 a 是一个终结符, 然后 α 是变元的串, 也就是 $\alpha \in V^*$. 简称为 GNF.

Example 1.5. 将 $S \rightarrow AB, A \rightarrow AaA \mid bB \mid b, B \rightarrow b$ 转化为 GNF.

虽然说, CNF 的转化是不要求掌握的.

就尼玛讲完了!

2 下推自动机

Contents:

1. 接受的语言
2. 其和文法的等价性
3. 确定型下推自动机.

2.1 下推自动机的定义

定义 2.1 (下推自动机). 下推自动机 (PDA)². 一个 PDA G 是一个七元组. PDA 相当于一个 NFA 加上一个栈. 进行每一次状态转移的时候, 会将栈顶元素弹出, 然后压入某些字符串. 初始阶段, 栈里面有一个初始元素 Z_0 . 比如说我们接受了 0, 而后弹出 Z_0 , 压入 $0Z_0$. 对于一个字符串, 压入顺序是从后往前的.

Example 2.2. 设计识别 $L_{01} = \{0^n 1^n \mid n \geq 1\}$ 的 PDA

Example 2.3. 设计识别 $L_{ww^R} = \{ww^R \mid w \in (0+1)^*\}$

²PDA for push down automata