形式语言与自动机理论笔记

Leo Lu

2021 年 12 月 8 日

蒋宗礼 信息楼 214 jiangzl@bjut.edu.cn

第一章　第一章绪论

1.1　引入：过河问题

人 -> m 狼 -> w 羊 -> g 白菜 -> c 初状态：

```
mwgc -
wc   - mg
mwc  - g
c    - mwg
mgc  - w
g    - mcw
mg   - cw
     - mgcw
```

1.2　重要性

GRE 中 80 道题，其中有 8～15 道形式语言。

1.3　Basic Concepts

1.3.1　Alphabet

An alphabet is a collection of characters.

Product of two alphabets:

$$\{0,1\} \times \{a,b\} = \{0a, 0b, 1a, 1b\}$$

Power of an alphabet:

$$\Sigma^0 = \epsilon, \Sigma^n = \Sigma^{n-1} \times \Sigma$$

Positive closure of an alphabet:

$$\Sigma^+ = \bigcup_{i=1}^{\infty} \Sigma^i$$

Kleene closure of an alphabet:

$$\Sigma^* = \bigcup_{i=0}^{\infty} \Sigma^i = \{\epsilon\} \cup \Sigma^+$$

1.3.2 Senctence

X is a “Sentence” if:$\forall X \in \Sigma^*$

1.3.3 Empty sentence

An empty sentence, denoted by ϵ or λ, is a string with no characters at all.

1.3.4 “Length” of a “Sentence”

$\forall X \in \Sigma^*$, the count of characters appeared in x is called the length of x, denote by $|x|$
For example: $|ababab| = 6; |\epsilon| = 0$
Note that $\{\epsilon\} \neq \emptyset$.

1.3.5 Concatenation of sentences

$\forall x, y \in \Sigma^*$, the concatenation of sentences x, y, denoted by $|xy|$, is the direct join of two strings.

$$|xy| = |x| + |y|$$

1.3.6 N-power of sentences

$\forall x \in \Sigma^*$, the n-power of sentence x:

$$x^n = \begin{cases} \epsilon & \text{n=0} \\ x^{n-1}x & \text{Other} \end{cases}$$

1.3.7 Prefix and Suffix

$\forall x, y, z, w, v \in \Sigma^*$, given that $x = yz, w = yv$, then:

1. y is the Prefix of x
2. z is the Suffix of x
3. if $z \neq \epsilon$, y is ”Proper Prefix” of x
4. if $y \neq \epsilon$, z is ”Proper Suffix” of x
5. y is the ”Common prefix” of x and w

For example, if $x = 0110$:

- Prefix of x is $\epsilon, 0, 01, 011, 0110$
- Proper prefix of x is $\epsilon, 0, 01, 011$
- Suffix of x is $\epsilon, 0, 10, 110, 0110$
- Proper suffix of x is $\epsilon, 0, 10, 110$

1.3.8 Reverse of a sentence

The reverse of sentence x is denoted by x^R or x^T.

1.3.9 Language on Alphabet Σ

$\forall L \subseteq \Sigma^*$, L is called a Language on alphabet Σ
$\forall x \in L$, x is called *a sentence of* L
For example: let $\Sigma = \{0, 1\}$, we have

1. $L_1 = \{0, 1\}$
2. $L_2 = \{00, 01, 10, 11\}$
3. $L_3 = \{0, 1, 00, 01, 10, 11, \ldots\} = \Sigma^+$
4. $L_4 = \{\epsilon, 0, 1, 00, 01, 10, 11, \ldots\} = \Sigma^*$
5. $L_5 = \{0^n | n \geq 1\}$
6. $L_6 = \{0^n 1^n | n \geq 1\}$
7. $L_7 = \{1^n | n \geq 1\}$
8. $L_8 = \{0^n 1^m | n, m \geq 1\}$
9. $L_9 = \{0^n 1^n 0^n | n \geq 1\}$
10. $L_{10} = \left\{0^n 1^m 0^k | n, m, l \geq 1\right\}$
11. $L_{11} = \{x | x \in \Sigma^+ \text{and the number of 0 and 1 of x are same}\}$

1.3.10 Operation of Language

All operatio on Sets also works on Language.
Note that $\cup, \cap, -, \bar{}$ *are closure*（封闭的）.
Product of Languages:
Given $L_1 \subseteq \Sigma_1^*, L_2 \subseteq \Sigma_2^*$, the product of L_1 and L_2 is a Language on alphabet $\Sigma_1 \cup \Sigma_2$.

$$L_1 L_2 = \{xy | x \in L_1, y \in L_2\}$$

Power of Languages:
Given a language L, we have:

$$L^n = \begin{cases} \epsilon & n = 0 \\ L^{n-1} L & Other \end{cases}$$

Positive closure of a language:

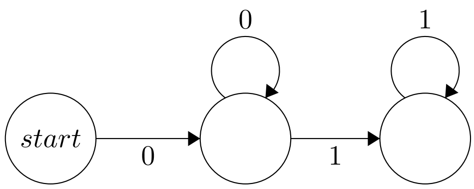
$$L^+ = \bigcup_{i=1}^{\infty} L^i$$

Kleene closure of a language:

$$L^* = \bigcup_{i=0}^{\infty} L^i = \{\epsilon\} \cup L^+$$

Note that: $L^+ = L^* \iff \epsilon \in L$
Examples

- 给定 Σ，讨论 Σ 上典型语言的结构特征
  - $\{0^n1^m|n,m\geq 1\}$:



  - $\{0^n1^n0^n|n\geq 1\}$
- 给定 Σ，讨论语言的结构与表示
  - $\{xx|x\in\Sigma^+\}=\{a_1a_2\dots a_na_1a_2\dots a_n|a_1,a_2,\dots,a_n\in\Sigma,n\geq 1\}$
  - $\left\{xx^T|x\in\Sigma^+\right\}$
  - $\left\{xx^Tw|x,w\in\Sigma^+\right\}$
    $=\{a_1a_2\dots a_n\dots a_1b_1\dots b_m|a_1,\dots,a_n,b_1,\dots,b_m\in\Sigma,n,m\geq 1\}$
  - $\left\{xwx^T|x,w\in\Sigma^+\right\}$
    $=\{a_1\dots a_nb_1b_2\dots b_na_n\dots a_1|a_1,\dots,a_n,b_1,\dots,b_m\in\Sigma,n,m\geq 1\}$
    $=\{aa_1\dots a_na|a,a_1,a_2,\dots,a_n\in\Sigma,n\geq 1\}$

第二章　第二章 文法

2.1　启示

- 对无穷对象的描述
  - $\{0^n|n\geq 1\}$
    - 0 是 S 的元素
    - $\forall x\in S,x0\in S$
    - $S\rightarrow 0$
    - $S\rightarrow S0$
  - $\{0^n1^m|n,m\geq 1\}$
    - $0\in S$
      $\forall x\in S,0x,x1\in S$
    - $S\rightarrow 01$
      $S\rightarrow 0S|S1$
    - $S\rightarrow S_1S_2$
      $S_1\rightarrow 0$
      $S_1\rightarrow S_10$
      $S_2\rightarrow 0$
      $S_2\rightarrow S_20$
  - $\{0,1\}^*$
    - $\epsilon\in S$
      $\forall x\in S,0x,1x\in S$

    - $S \to \epsilon$
      $S \to 0S$
      $S \to 1S$
  - $\{0,1\}^*\{11\}\{0,1\}^*$
    - $11 \in S$
      $\forall x \in S, 0x, 1x, x0, x1 \in S$
    - $S \to 11$
      $S \to 0S|1S|S0|S1$

- 如何定义中缀表达式：递归
  - 描述：
    - $ident$ 是表达式
    - 表达式加表达式是表达式
    - 表达式减表达式是表达式
    - 表达式乘表达式是表达式
    - 表达式除表达式是表达式
    - 表达式加括号是表达式
  - 定义：
    - 表达式定义为标识符
    - 表达式定义为表达式 + 表达式
    - 表达式定义为表达式 − 表达式
    - 表达式定义为表达式 × 表达式
    - 表达式定义为表达式 ÷ 表达式
    - 表达式定义为（表达式）
  - 符号化
    $E \to ident$
    $E \to E + E$
    $E \to E - E$
    $E \to E \times E$
    $E \to E \div E$
    $E \to (E)$
  - 表示优先级
    - 因子是标识符
    - 因子是括号的表达式
    - 项是因子
    - 项是因子 */ 因子
    - 表达式是项
    - 表达式是表达式 +− 表达式
  - 符号化
    - Variables: E, T, F

* Terminals: $+ - \times \div ident()$
* Products:
  $E \to T + T$
  $E \to T - T$
  $E \to T$
  $T \to F \times F$
  $T \to F \div F$
  $T \to F$
  $F \to ident$
  $F \to (E)$
* Start Symbol：E

2.2 形式定义

定义 2.1. 文法（*Grammar*）G 是一个四元组

$$G = (V, T, P, S)$$

其中，

V— 变量（Variable）的非空有穷集。$\forall A \in V$，A 叫做语法变量（*syntactic variable*），也叫非终极符号（*nonterminal*）。

T— 终极符（Terminal）的非空有穷集。$\forall a \in T$，a 叫做终极符。$V \cup T = \emptyset$。

P— 产生式（Production）的非空有穷集。对于 $a \to b$，a 是左部，b 是右部。

S— $S \in V$，文法 G 的开始符号（Start symbol）。

约定：

- 只写产生式，第一个产生式的左部为开始符号
- 对一组有相同左部的产生式
  $\alpha \to \beta_1, \alpha \to \beta_2, \alpha \to \beta_3, \ldots$ 可以记为 $\alpha \to \beta_1|\beta_2|\beta_3 \ldots$ $\beta_1, \beta_2, \beta_3$ 称为候选式（Candidate）
- 形如 $\alpha \to \epsilon$ 的产生式叫做空产生式，也可叫做 ϵ 产生式
- 符号
  - 英文大写字母为语法变量
  - 英文小写字母为终结符号
  - 英文较后的大写字母为语法变量或者终极符号
  - 英文较后的大写字母为终极符号行
  - 希腊字母表示语法变量和终极符号组成的行

定义 2.2. 设 $G = (V, T, P, S)$ 是一个文法，如果 $\alpha \to \beta \in P, \gamma, \delta \in (V \cup T)$，则称 $\gamma\alpha\delta$ 在 G 中直接推导（*Derivation*）出 $\gamma\beta\delta$，记作 $\gamma\alpha\delta \underset{G}{\Rightarrow} \gamma\beta\delta$。

于此相对应，$\gamma\beta\delta$ 归约到 $\gamma\alpha\delta$，简称 β 归约为 α。

$\underset{G}{\Rightarrow}$ 是 $(V \cup T)^*$ 上的二元关系。

定义 2.3. 对于文法 G:

$\underset{G}{\overset{n}{\Longrightarrow}} = \left(\underset{G}{\Rightarrow}\right)^n$

$\underset{G}{\overset{*}{\Longrightarrow}} = \left(\underset{G}{\Rightarrow}\right)^*$

$\underset{G}{\overset{=}{\Longrightarrow}} = \left(\underset{G}{\Rightarrow}\right)^+$

当只有唯一的文法 G 时，可以省略 G: $\overset{n}{\Longrightarrow}, \overset{*}{\Longrightarrow}, \overset{+}{\Longrightarrow}$

定义 2.4. 对于语言 $G = (V, T, P, S)$:

语法范畴 A $L(A) = \left\{w | w \in T^* \text{且} A \overset{*}{\Rightarrow} w\right\}$

语言（*Language*）$L(G) = \left\{w | w \in T^* \text{且} S \overset{*}{\Rightarrow} w\right\}$

句子（*Sentence*）$\forall w \in L(G)$

句型（*Sentential Form*）$\forall \alpha \in (V \cup T)^*$，如果 $S \overset{*}{\Rightarrow} \alpha$，则称 α 是 G 产生的一个句型。

定义 2.5. 对于文法 G_1, G_2，如果 $L(G_1) = L(G_2)$，则称 G_1 与 G_2 等价。

2.3 文法的构造

- $L(G) = \{0, 1, 00, 11\}$
  - $G_1 : S \to 0|1|00|11$
  - $G_2 : S \to A|B|AA|BB, A \to 0, B \to 1$
  - $G_3 : S \to 0|1|0A|1B, A \to 0, B \to 1$
  - $G_4 : S \to A|B|AA|BB, A \to 0, B \to 1, C \to 1$
- $\{x | x$是至少 3 个 1 的 0，1 串$\}$
  - $G : S \to A1A1A1A, A \to \epsilon|0A|1A$
  - $G : S \to A1A1A1B, A \to \epsilon|0A, B \to \epsilon|0B|1B$
  - $G : S \to AAAB, A \to 1|0A, B \to \epsilon|0B|1B$

2.4 文法的乔姆斯体系

定义 2.6. 对于文法 $G = (V, T, P, S)$:

G 叫做 0 型文法，Type 0 Grammar，也叫短语结构文法（PSG，Phrase Structure Grammar）

$L(G)$ 是 0 型语言，也叫短结构语言，可递归枚举集。

定义 2.7. 对于 *0* 型文法文法 $G = (V, T, P, S)$:

如果对于 $\forall \alpha \to \beta \in P$，均有 $|\beta| \geq |\alpha|$，则 G 是 *1* 型文法，或上下文有关文法。

定义 2.8. 对于 *1* 型文法文法 $G = (V, T, P, S)$:

如果对于 $\forall \alpha \to \beta \in P$，均有 $|\beta| \geq |\alpha|$，并且 $\alpha \in V$ 则 G 是 *2* 型文法，或上下文无关文法。

定义 2.9. 对于 *2* 型文法文法 $G = (V, T, P, S)$:

如果对于 $\forall \alpha \to \beta \in P$:

如果形如 $A \to wB$ 和 $A \to w$，其中 $A, B \in V, w \in T^+$: G 是右线性文法。

如果形如 $A \to Bw$ 和 $A \to w$，其中 $A, B \in V, w \in T^+$: G 是左线性文法。

则 G 是 *3* 型文法，或正则文法。

2.5 空产生式

允许在 CSG，CFG，RG 文法中存在空产生式。
允许在 CSL，CFL，RL 语言中存在空语句。
特点：

- 对于 $\forall$ 右线性文法 G_1，$\exists$ 左线性文法 G_2 使得 $L(G_1)=L(G_2)$
- 对于 $\forall$ 左线性文法 G_1，$\exists$ 右线性文法 G_2 使得 $L(G_1)=L(G_2)$
  所以，某种意义上二者等价。其中，左线性文法的表述好。
  左递归？ 线性文法不是正则文法！
- $\forall G, \exists G', L(G')=L(G)$，但是 G' 中的开好似符号不出现在任何产生式的右部，且在 $\epsilon \in L(G')$ 时，G' 中只有 $S' \to \epsilon$ 这样一个 ϵ 产生式。

2.5.1 语言运算

给定上下文无关文法 G_1, G_2，构造 G 使得：

1. $L(G)=L(G_1)L(G_2)$
   其中 $V_1 \cup V_2=\emptyset$，$S \notin V_1 \cup V_2$
   $G=(V_1 \cup V_2 \cup S, T, P_1 \cup P_2 \cup S \to S_1S_2, S)$
2. $L(G)=L(G_1) \cup L(G_2)$
   $G_\cup=(V_1 \cup V_2 \cup S, T, P_1 \cup P_2 \cup P_3, S)$
   $P_3=\{S \to S_1|S_2\}$
3. $L(G)=L(G_1)^*$
   $G_*=(V_1 \cup \{S\}, T, P_1 \cup P_2, S)$
   $P_2=\{S \to \epsilon|SS_1\}$

给定 RG G_1, G_2，构造 RG G 使得：

1. $L(G)=L(G_1)L(G_2)$
   其中 $V_1 \cup V_2=\emptyset$，$S \notin V_1 \cup V_2$
   $S_1 \Rightarrow a_1A_1$
   $\Rightarrow a_1a_2 \dots a_{n-1}A_n$ 所以，我们需要改造 P_1。
   $\Rightarrow a_1a_2 \dots a_nS_2$
   $G=(V_1 \cup V_2 \cup S, T, P_1' \cup P_2 \cup P_3, S_1)$
   $P_1'=\{A \to aB|A \to aB \in P_1\}$
   $P_3=\{A \to aS_2|A \to a \in P_1\}$
2. $L(G)=L(G_1) \cup L(G_2)$
   $G_\cup=(V_1 \cup V_2 \cup S, T, P_1 \cup P_2 \cup P, S)$
   $P=\{S \to \alpha|S_1 \to \alpha \in P_1\}$
   $\cup \{S \to \alpha|S_2 \to \alpha \in P_2\}$
3. $L(G)=L(G_1)^*$
   自己想！

例子：

1. 设 $L = \{x|101 \text{ in } x\}$，构造 $G, L(G) = L$。

   朴素构造：$\begin{cases} S \to A101A \\ A \to \epsilon|0A|1A \end{cases}$

   正则构造：$\begin{cases} S \to 0S|1A \\ A \to 0B|1A \\ B \to 1C|0S \\ C \to 0C|1C|\epsilon \end{cases}$

2. 构造 G 使 $L(G) = \{x|101 \text{ not in } x\}$

   正则构造：$\begin{cases} S \to 0S|1A \\ A \to 0B|1A \\ B \to 0S|\epsilon \end{cases}$

2.6　小结

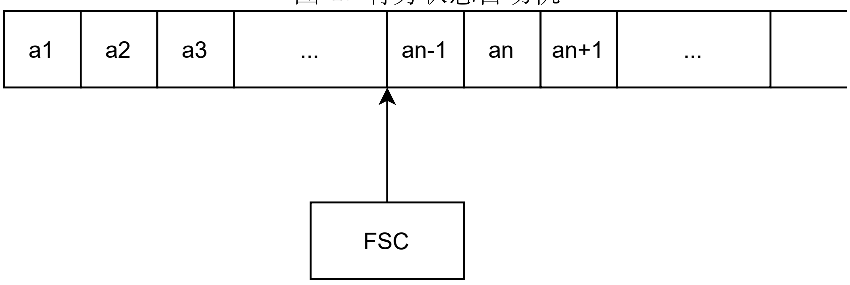
习题：p67 3，4，8.2，8.6，10.3，11.3

第三章　有穷状态自动机 Finite Automata

习题：p103 2.4, 2.11; p104 7, 10.6

3.1　FA 的基本定义

图 1: 有穷状态自动机



定义 3.1. $M = (Q, \Sigma, \delta, q_0, F)$，其中：

Q ：状态的有穷集合

Σ ：输入字母表

δ ：状态转义函数
　$Q \times \Sigma \to Q$。$\forall (q, a) \in Q \times \Sigma$，$\delta(q, a) = p$ 表示 M 在状态 q 读入一个字符 a，状态改为 p 并指向下一个字符。

q_0 ：开始状态

F ：终止状态

定义 3.2. 有穷状态自动机 $M = (Q, \Sigma, \delta, q_0, F)$:

状态转移图是满足如下条件的有向图。

1. 对于 $q \in Q$，q 是一个顶点
2. $\forall(q, a) \in \delta$, q 到 p 有一条标记为 a 的弧。
3. 标有 S 的箭头所指的状态为开始状态。
4. 用双圈标记结束状态。

扩展 δ 为 $\hat{\delta} : Q \times \Sigma^* \to Q$

1. $\hat{\delta}(q, \epsilon) = q$
2. $\hat{\delta}(q, wa) = \delta(\hat{\delta}(q, w), a)$

注意到 $Q \subset Q \times \Sigma^*$，且对 $\forall(q, a) \in Q \times \Sigma$，$\hat{\delta}(q, a) = \delta(q, a)$ 所以，不用区分 δ 与 $\hat{\delta}$。δ 是 $Q \times \Sigma^* \to Q$ 上的映射。

定义 3.3. 有穷状态自动机 $M = (Q, \Sigma, \delta, q_0, F)$ 识别的语言：$L(M) = \{x|\delta(q_0, x) \in F\}$

定义 3.4. 对于有穷状态自动机 M_1, M_2：若有 $L(M_1) = L(M_2)$，则称二者等价。

定义 3.5. 对于有穷自动状态机 $M = (Q, \Sigma, \delta, q_0, F)$，$set(q) = \{x|\delta(q_0, x) = q\}$

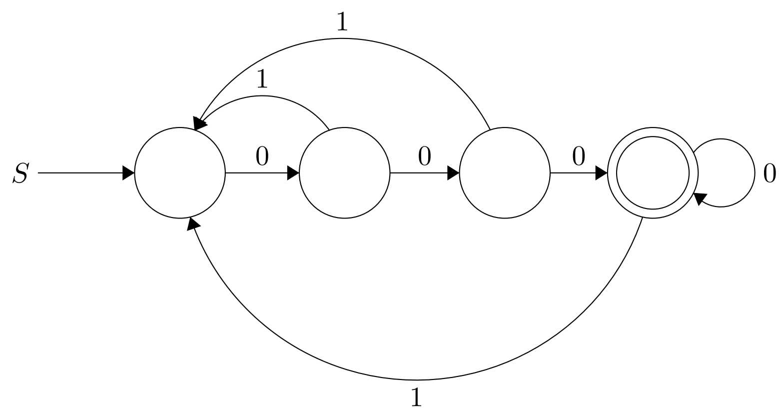
所以有：$\Sigma^* = \bigcup_{q \in Q} set(q)$；$\forall q, p \in Q, q \neq p, set(q) \cap set(p) = \emptyset$

同时，如果 Q 中不存在不可达状态，则 $set(q_0), set(q_1) \ldots$ 是 Σ^* 的一个划分。

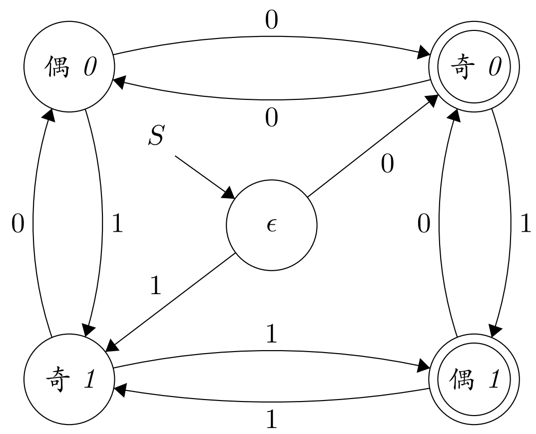
定义 3.6.

$$\begin{aligned} xR_My &\iff \delta(q_0, x) = \delta(q_0, y) \\ &\iff \exists q \in Q, x, y \in set(q) \end{aligned}$$

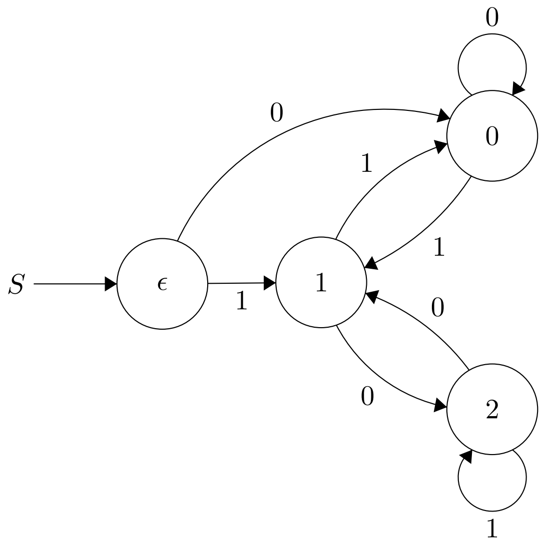
例 3.1. 构造 M，使得 $L(M) = \left\{x000|x \in \{0,1\}^*\right\}$。



例 3.2. 构造 M，使得 $L(M) = \left\{0x0|x \in \{0,1\}^*\right\}$。



例 3.3. 构造 M，使得 $L(M) = \{x|x$ 看作二进制时，$x \mod 3 = 1\}$。



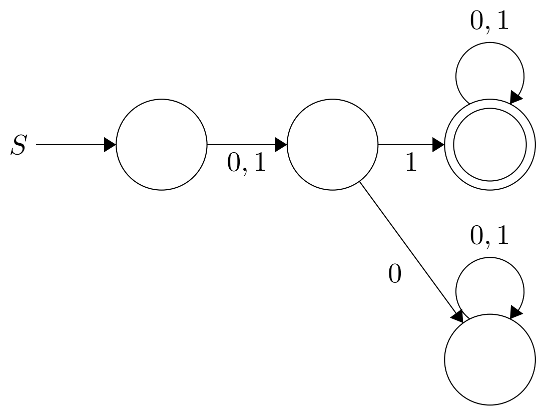
例 3.4. 对于 *FSA* $M = (Q, \Sigma, \delta, q_0, F)$，构造 *FA* M' 使得 $L(M') = \Sigma^* - L(M)$

$M' = (Q, \Sigma, \delta, q_0, \complement_Q F)$

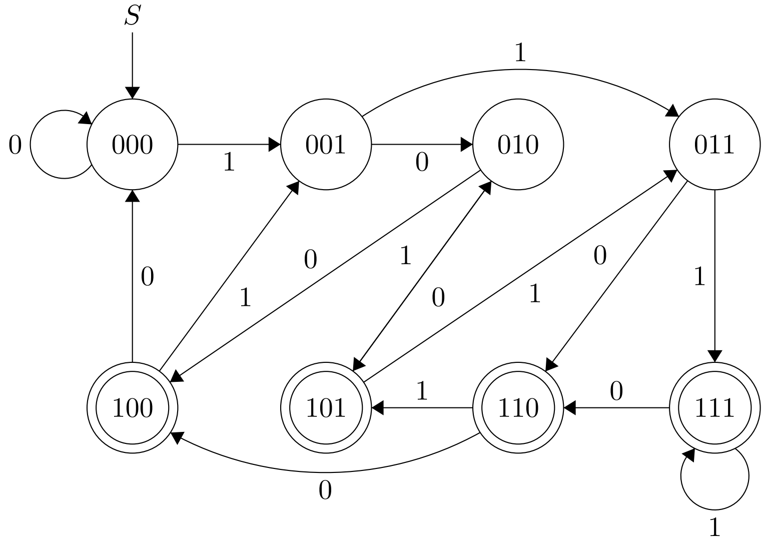
例 3.5. 给定 $\begin{cases} M_1 = (Q_1, \Sigma, \delta_1, q_{01}, F_1) \\ M_2 = (Q_2, \Sigma, \delta_2, q_{02}, F_2) \end{cases}$ 构造：

- M_3 使得 $L(M_3) = L(M_1) \cap L(M_2)$
- M_3 使得 $L(M_3) = L(M_1) \cup L(M_2)$

例 3.6. 构造 M，使得 $L(M) = \left\{x|x \in \{0,1\}^+ \text{and } x \text{ 的第三个字符是 } 1\right\}$。



例 3.7. 构造 M，使得 $L(M)=\Big\{x|x\in\{0,1\}^{+}\text{and } x \text{ 的倒数第三个字符是 } 1\Big\}$。



3.2 NFA 不确定的有穷状态自动机

定义 3.7. *NFA* $M=(Q,\Sigma,\delta,q_0,F)$，其中：

Q,Σ,q_0,F 同 *FA*。

δ: $Q\times\Sigma\to 2^Q$ (幂集),

$\delta(q,a)=\{p_1,p_2,\ldots,p_n\}$ 表示 M 在状态 q 读入 a 可以选择进入 p_i

扩展 δ 为 $\hat{\delta}:Q\times\Sigma^*\to 2^Q$

$$\hat{\delta}(q,\epsilon)=\{q\}$$
$$\hat{\delta}(q,wa)=\bigcup_{q_0\in\hat{\delta}(q,w)}\delta(q_0,a)$$

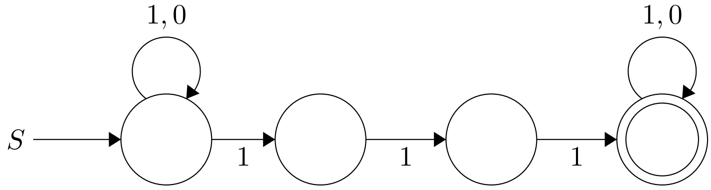
进一步拓展：$\hat{\delta}:2^Q\times\Sigma^*\to 2^Q$

$$\hat{\delta}(P,x)=\bigcup_{p\in P}\hat{\delta}(p,x)$$

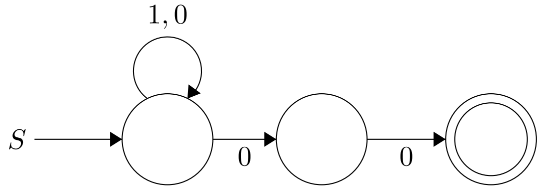
注意到 $Q\times\Sigma\subset 2^Q\times\Sigma^*$，且对 $\forall(q,a)\in Q\times\Sigma^*$，$\hat{\delta}(q,a)=\delta(q,a)$ 所以不用区分 δ 与 $\hat{\delta}$。

δ 是 $Q\times\Sigma^*\to Q$ 上的映射。

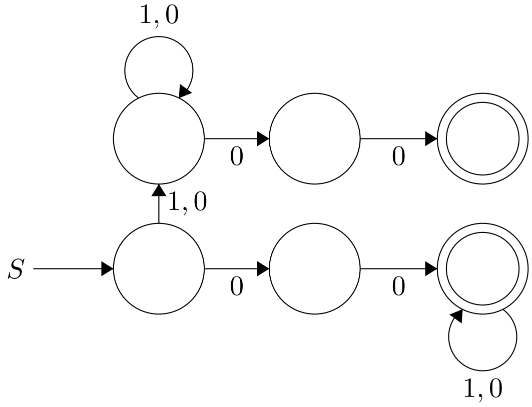
例 3.8. 构造 *NFA* M，使得 $L(M)=\{x|x$ 是含 *110* 的 *01* 串$\}$



例 3.9. 构造 *NFA* M，使得 $L(M)=\{x|x$ 是 *00* 结尾的 *01* 串$\}$



例 3.10. 构造 *NFA* M，使得 $L(M)=\{x|x$ 是 *00* 开头或结尾的 *01* 串$\}$



定理 3.1. *NFA* 与 *DFA* 等价。

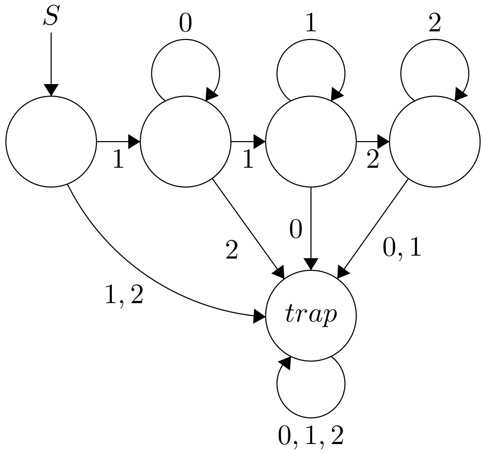
证：对于 *NFA* $M=(Q,\Sigma,\delta,q_0,F)$，构造 *DFA* $M'=(2^Q,\Sigma,\delta',[q_0],F')$

$F'=\{[q_1,q_2,\ldots,q_n]|\{q_1,q_2,\ldots,q_n\}\cap F\neq\emptyset\}$

$\delta'([q_1,q_2,\ldots,q_n],a)=[\beta_1,\beta_2,\ldots,\beta_n]\iff\delta(\{q_1,q_2,\ldots,q_n\},a)=\{\beta_1,\beta_2,\ldots,\beta_n\}$

所以，对于 $\forall M$ 是 *NFA*，可以构造 *DFA* M'

例 3.11. 构造 *DFA* M，$L(M)=\left\{0^n1^m2^k|n,m,k\geq 1\right\}$



3.3　带空移动的有穷状态自动机

定义 3.8. $\epsilon-\mathrm{NFA}\ M=(Q,\Sigma,\delta,q_0,F)$：

δ：状态转义函数。$\delta:Q\times(\Sigma\cup\{\epsilon\})\to 2^Q$

其中，对于 $\forall q\in Q,\delta(q,\epsilon)=\{p_1,p_2,\ldots,p_n\}$：表示在 q 状态不读入任何字符，可以将状态变为 $p_1,p_2,\ldots,p_n$，称为 M 在 q 状态做了一次空移动（ϵ 移动）

1. $\epsilon-CLOSURE(q)=\{p|$从 q 到 p 有一条标记为ϵ的路$\}$

2. $\epsilon-CLOSURE(P)=\bigcup\limits_{p\in P}\epsilon-CLOSURE(p)$

3. $\hat{\delta}(q,\epsilon)=\epsilon-CLOSURE(q)$

4. $\hat{\delta}(q,wa)=\epsilon-CLOSURE(P)$

   $P=\bigcup\limits_{r\in\hat{\delta}(q,w)}\delta(r,a)$

5. 进一步拓展 $\delta : 2^Q \times \Sigma \to w^Q$:

$$\delta(P,a) = \bigcup_{q \in P} \delta(q,a)$$

6. 进一步拓展 $\hat{\delta} : 2^Q \times \Sigma^* \to 2^Q$:

$$\delta(P,w) = \bigcup_{q \in P} \hat{\delta}(q,w)$$

注意：$\hat{\delta} \neq \delta$

定义 3.9. $\epsilon - NFA\ M$ 识别的语言：$L(M) = \{x | \hat{\delta}(q_0, x) \cap F \neq \emptyset\}$

定理 3.2. $\epsilon - NFA$ 等价与 NFA。

证明：设 $\epsilon - NFA\ M = (Q, \Sigma, \delta, q_0, F)$:

取 $M' = \left(Q, \Sigma, \hat{\delta}, q_0, \begin{cases} F \cup \{q_0\} & F \cap \epsilon - CLOSURE(q_0) \neq \emptyset \\ F & F \cap \epsilon - CLOSURE(q_0) = \emptyset \end{cases} \right)$

由此可见，DFA，NFA，$\epsilon - NFA$ 等价。以后统称为 FA。

3.4 FA 与 RG 等价

定理 3.3. 对 $\forall\ DFA\ M$, $\exists\ RG\ G$ 使得 $L(G) = L(M)$。

构造：对于 $DFA\ M = (Q, \Sigma, \delta, q_0, F)$，取 $RG\ G = (Q, \Sigma, P, q_0)$

$$P = \{q \to ap | \delta(q,a) = p\} \cup \{q \to a | \forall \delta(q,a) \in F\} \cup \{q_0 \to \epsilon | q_0 \in F\}$$
$$= \{q \to ap | \delta(q,a) = p\} \cup \{q \to \epsilon | q \in F\}$$

定理 3.4. 对 $\forall\ RG\,G$, $\exists\ FA\ M$ 使得 $L(G) = L(M)$。

构造：对于 $RG\ G = (V, T, P, S)$，RG 为右线性文法。

取 $FA\ M = (V \cup \{f\}, T, \delta, S, \{f\})$，其中

$$\delta(A,a) = \{B | \forall A \to aB \in P\} \cup \{f | \forall A \to a \in P\}$$

第四章　正则表达式

4.1 正则表达式的基本概念

定义 4.1. σ 上的 RE 是满足如下条件的式子：

- ϵ 是 RE，表示的语言 $L(\epsilon) = \{\epsilon\}$
- $\emptyset$ 是 RE，表示的语言 $L(\emptyset) = \emptyset$
- $\forall a \in \sigma$，a 是 RE，表示的语言 $L(a) = \{a\}$
- $\forall r, s$ 是 RE 且分别表示 R, S，则有：
  - $(r+s)$ 是 RE，表达的语言 $L(r+s) = R \cup S$
  - (rs) 是 RE，表达的语言 $L(r+s) = RS$
  - (r^*) 是 RE，表达的语言 $L(r+s) = R^*$

约定 1：运算符优先级 $* > \times > +$

约定 2：引入正闭包 $r^+ = r^* r$

定义 4.2. 如果 $L(r)=L(s)$，则称 $r=s$

不难看出：

- $(r+s)+t=r+(s+t)$
- $r+s=s+r$
- $(rs)t=r(st)$
- $r(s+t)=rs+rt$
- $(r+s)t=rt+st$
- $r+r=r$
- 如果 $L(s)\subseteq L(r)$，则 $s+r=r$
- $r\epsilon=\epsilon r=r$
- $\emptyset r=r\emptyset=\emptyset$
  $\emptyset+r=r$

约定：当意思明确时，用 r 表示 $L(r)$

例 4.1. $\Sigma=\{0,1\}$

1. *至少含 3 个 1 的 01 串构成的语言：*
   $0^*10^*10^*1(0+1)^*$
2. *最多含 3 个 1 的 01 串构成的语言：*
   $0^*(1+\epsilon)0^*(1+\epsilon)0^*(1+\epsilon)0^*$
3. *不含 1 的 01 串构成的语言：*
   0^*
4. *偶数个 0 的 01 串：*
   $(1^*01^*0)^*1^*$

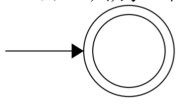
定义 4.3.
- $r^0=\epsilon$
- $r^n=r^{n-1}r$

显然：

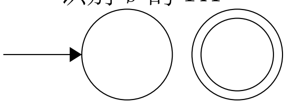
- $r^nr^m=r^{n+m}$
- $r^n+s^n\neq(r+s)^n$
- $r^ns^n\neq(rs)^n$

4.2 RE →FA

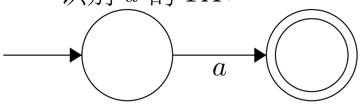
证明. 识别 ϵ 的 FA：



识别 $\emptyset$ 的 FA：



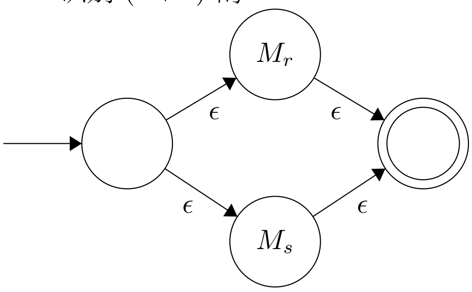
识别 a 的 FA：



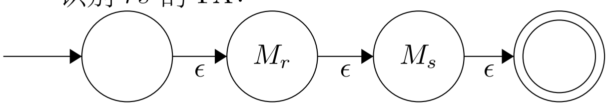
观察得到：上述 3 个 FA 具有如下性质：

1. 有且只有一个终止状态
2. 没有从终止状态出发的狐

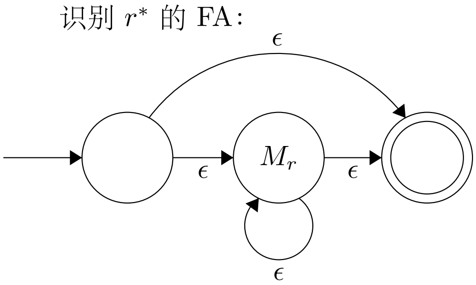
设 M_r, M_S 是 r, s 的具有上述 2 个性质的 FA，则：
识别 $(r+s)$ 的 FA：



识别 rs 的 FA：



识别 r^* 的 FA：



□

定理 4.1. *对于 $\forall$ RE r,$\exists$ FA M，使得 $L(M) = L(r)$*

第五章　正则语言的性质

5.1　正则语言的泵引理

例 5.1. *如何证明 $L = \{0^n 1^n \ n \geq 1\}$ 不是 RL?*

假设 L 是 RL。$DFA\ M = (Q, \Sigma, \delta, q_0, F)$ 使得 $L(M) = L$。

记 $|Q| = N$，$N \geq 1$，取 $0^n 1^n \in L, n \geq N$。

设 q_i 为 M 读取 $0^n 1^n$ 的前 i 位时的状态。注意到 $n \geq N$，所以在 $q_0, q_1, \ldots, q_n$ 中至少有两个重复状态。不妨设 $q_i = q_j$ 是最早重复的状态。则 $0 \leq i, j \leq N$。

所以：

- $\delta(q_0, 0^i) = q_i$
- $\delta(q_i, 0^{j-i}) = q_j$

- $\delta(q_j, 0^{n-j}1^n) = q_2n \in F$

  所以：$\delta(q_i, (0^{j-i})^k) = q_i = q_j$。也就是说，对于 $\forall k \geq 0$，$\delta(q_0, 0^i(0^{j-i})^k 0^{n-j}1^n) \in F$

引理 5.1. 正则语言的的泵引理。对于 $\forall$ RL L，存在一个仅依赖与 L 的正整数 N，对于 $\forall z \in L, |z| \geq N$，则存在 u, v, w 满足以下条件：

- $uvw = z$
- $|uv| \leq N$
- $|v| \geq 1$
- 对于 $\forall k \geq 0$，$uv^k w \in L$

引理 5.2. 拓展的泵引理。对于 $\forall$ RL L，存在一个仅依赖与 L 的正整数 N，对于 $\forall z = z_1z_2z_3 \in L, |z_2| \geq N$，则存在 u, v, w 满足以下条件：

- $uvw = z_2$
- $|uv| \leq N$
- $|v| \geq 1$
- 对于 $\forall k \geq 0$，$uv^k w \in L$

例 5.2. 证明 $L = \{0^p | p\text{是质数}\}$ 不是 RL。

设 L 是 RL。N 是泵引理所说正整数。设 x 为大于 N 的最小质数。
取 $z = 0^x$。取 $v = 0^l$, $l \geq 1$。$uv^k w = 0^{x+(k-1)l}$。
当 $k = x + 1$ 时，有 $x + (x + 1 - 1)l = (l + 1)x$ 是合数，所以 $uv^k w \notin L$，与泵引理矛盾。
所以 L 不是 RL。

例 5.3. 设 $L = \{0^{2n} | n \geq 1\}$，证明 L 不是 RL。

假设 N 是仅依赖于 L 的正整数。取 $z = 0^{2N}$，取 $u = 0^l$, $v = 0$, $w = 0^{2N-l-1}$。
$uv^k w = 0^{2N+(k-1)}$ 当 $k = 0$ 时，$uv^k w = 0^{2N-1} \neq L$。
与泵引理矛盾，所以 L 不是 RL。
证明错误！不能取 $v = 0$，要让 $\forall v$ 矛盾！

5.2 RL 的封闭性

定理 5.1. RL 对于并、乘、闭包封闭。

证明：根据正则表达式的定义，定理显然成立。 □

定理 5.2. RL 对补运算封闭。

证明：把对应的 DFA 的 F 取反，可构造新的 DFA，可构造新的 RL。 □

定理 5.3. RL 对交运算封闭。

证明：$L1 \cup L2 = \overline{\overline{L1} \cap \overline{L2}}$ □

5.3 Myhill 定理

定义 5.1. $xR_My \iff \delta(q_0,x)=\delta(q_0,y)$
$\iff \exists q\in Q, x,y\in set(q)$

定义 5.2. $xR_Ly \iff \forall z\in\Sigma^*, xz\in L \iff yz\in L$

定义 5.3. *R 是右不变的是指如果 xRy 则有 $\forall z\in\Sigma^*, xzRyz$*

定理 5.4. *对于 DFA M，如果 xR_My，则 $xR_{L(M)}y$。*

证明：由 xR_My 知 $\delta(q_0,x)=\delta(q_0,y)$。不妨设其为 q_1。

对于 $\forall z\in\Sigma^*$，$\delta(q_0,xz)=\delta(\delta(q_0,x),z)=\delta(q_1,z)=\delta(q_0,yz)$。不妨设其为 q_2。

如果 $xz\in L$，则有 $q_2\in F$，所以 $yz\in L$。反之亦然。

所以 $xz\in L \iff yz\in L$。所以 $xR_My \iff xR_{L(M)}y$ □

定理 5.5. *Myhill 定理。以下 3 个命题等价：*

1. *L 是 RL*
2. *L 是 Σ^* 上的某个具有有穷指数的右不变等价关系的某些等价类的并。*
3. *R_L 具有有穷指数。*

证明. $1\Rightarrow 2$。

设 L 是 RL，DFA M 使得 $L(M)=L$。

R_M 是 Σ^* 上的右不变等价关系，且 $|\Sigma^*/R_M|\le|Q|$。

$L=\bigcup\limits_{q\in F} set(q)$ □

证明. $2\Rightarrow 3$。

设 L 是 Σ^* 上的具有有穷指数的右不变等价关系 R 的某些等价类的并。

下面证明如果 xRy，那么 xR_Ly。

设 $x,y\in\Sigma^*, xRy$。由 R 的右不变性，对于 $\forall z\in\Sigma^*$，$xzRyz$。

再注意到 L 是 R 的某些等价类的并，且 xz,yz 在同一个等价类中，所以 $xz\in L \iff yz\in L$。

所以 xR_Ly。 □

证明. $3\Rightarrow 1$。

设 R_L 具有有穷指数。取 $M'=(\Sigma^*/R_L,\Sigma,\delta',[\epsilon],\{[x]|x\in L\})$

$\delta'([a],x)=[ax]$。显然，$L(M')=L$。 □

意义：L 是 RL 的充要条件

定理 5.6. *在同构意义下，M' 是状态最少的唯一识别 L 的 DFA。*

5.4 DFA 的最小化

用极小化算法得到的 DFA 去掉不可达状态后得到的 DFA 为状态最少的 DFA。

- 寻找可以（应该）合并的类/状态
- 算法实现：
  找出所有不可合并的类

5.4.1　证明 L 是/不是 RL

1. 证明 L 是 RL。

   证明 R_L 的指数有穷。

2. 证明 L 不是 RL:

   证明 R_L 的指数无穷。

例 5.4. 证明 $\{0^n1^m|n,m\geq 1\}$ 是 RL。
R_L 的指数应该有穷。首先分析 L 的特征:

1. 0 和 1 的个数非 0。
2. 0 不能出现在 1 后面。

故，分成如下几类:

1. 0 在 1 后面（$\{x|x\text{中有子串}10\}$）
2. $\{0^n1^m|n,m\geq 1\}$
3. $\{\epsilon\}$
4. 1^n
5. 0^n

观察发现，1 和 5 可以合并，因此得到:
$\Sigma^*/R_L=\{[\epsilon],[0],[01],[10]\}$。$R_L$ 的指数为 4，所以 L 是 RL。

例 5.5. 证明 $\{0^n1^n|n\geq 0\}$ 不是 RL。
易得，对于 $\forall i\neq j\in N$，$0^i\not R_L 0^j$ 因此 R_L 的指数无穷。

第六章　CFL 上下文无关语言

6.1　CFG

定义 6.1. *CFG* $G=(V,T,P,S)$ 的语法树为满足如下条件的树:

1. 每个节点的标记 $x\in V\cup T\cup\{\epsilon\}$
2. 如果节点 V 的标记为 A, V 从左到右的子节点 $V_1,V_2,\ldots,V_k$ 的标记依次为 $y_1,y_2,\ldots,y_k$，则 $A\to y_1y_2\ldots y_k\in P$
3. 根节点标记为 S
4. 中间节点的标记为变量 $x\in V$
5. 从左到右的叶子节点 $v_1,\ldots,v_n$ 的标记 $x_1,\ldots,x_n$ 组成的串 $x_1x_2\ldots x_n$ 为该树的结果。
6. 如果 v 的标记为 ϵ，则它没有兄弟。

例 6.1. $X\to(S)|S|\epsilon$，请画出句型 $((S))$ 的语法树。

定义 6.2. 满足语法树定义中除第三条外条件的树，称作 A-子树。

定理 6.1. 有一颗结果为 α 的语法树 $\iff S \overset{*}{\Rightarrow} \alpha$

定义 6.3. 每一步派生均实施在当前句型最右变量上的派生叫最右派生。

每一步派生均实施在当前句型最左变量上的派生叫最左派生。

定理 6.2. 最左派生与最右派生的语法树是一一对应的。

定义 6.4. 如果 *CFG G* 有句子有棵颗不同的语法书，则 G 是二义性的。

定义 6.5. 如果 *CFL L* 没有非二义性文法，则称之为固有二义性的。

6.2 去无用符号

6.2.1 去除无用符号

定义 6.6. X 是有用符号，即 $\exists X \in L(G)$，$S \overset{*}{\Rightarrow} \alpha X \beta \overset{*}{\Rightarrow} x$。

X 是有用的，必须同时满足如下两条:

1. $S \overset{*}{\Rightarrow} \alpha X \beta$

   如何判断?

   (a) $V' = \{S\} \cup \{A | S \to \alpha A \beta \in P\}$
       $T' = \{a | S \to \alpha a \beta \in P\}$

   (b) 重复
       $V' = V' \cup \{B | A \to \alpha B \beta \in P \& A in V'\}$
       $T' = T' \cup \{a | A \to \alpha a \beta \in P \& A in V'\}$

2. $X \overset{*}{\Rightarrow} w, w \in T^*$

   $G = (V, T, P, S)$，对于:

   $\forall a \in T$，a 满足 2。

   $\forall A \in V$，如何判断 A 是否满足 2?

   (a) $V' = \{A | A \to w \in P\}$

   (b) 重复 $V' = \{A | A \to \alpha, \alpha \in (V' \cup T)^*\} \cup V'$

定理 6.3. 对于 $\forall$ *CFG G*，$\exists$ *CFG* G'，

1. $L(G') = L(G)$

2. G' 中无无用符号。

6.2.2 去除 $\epsilon-$ 产生式

定义 6.7. 如果 $A \overset{*}{\Rightarrow} \epsilon$，则称 A 为可空变量。

定理 6.4. 对于 $\forall$ *CFG* G, G':

1. G' 中无空产生式

2. $L(G') = L(G) - \{\epsilon\}$

对于 $A \to x_1 x_2 \ldots x_n$，替换为 $A \to y_1 y_2 \ldots y_n$，其中当 x_i 不是可空变量时，$y_i = x_i$，否则 $y_i = x_i$ 或 ϵ。

注意 $y_1 y_2 \ldots y_n$ 不能都为 ϵ。

6.2.3 去单一产生式

定义 6.8. 形如 $A \to B$ 的产生式是单一产生式。

定义 6.9. 对 $\forall$ *CFG* G，$\exists$ *CFG* $G', L(G') = L(G)$，G' 中无单一产生式。

推论：对于 $\forall$ *CFG* G, 存在 *CFG* G' 使得 $L(G') = L(G)$，且 G' 没有无用符，$\epsilon-$ 产生式和单一产生式。

6.3 CNF 乔姆斯基范式

定义 6.10. 如果 G 的产生式均具有以下形式：

$$\begin{cases} A \to BC \\ A \to a \end{cases}$$

则称之为 *CNF*。

6.4 GNF 格雷巴赫范式

定义 6.11. 如果 G 的产生式均具有以下形式：

$$A \to a\alpha, \alpha \in V^*$$

则称之为 *GNF*。

例 6.2. $A \to Aa|b = \begin{cases} A \to b|bB \\ B \to a|aB \end{cases}$

例 6.3. $A \to Aa|Ab|c|d = \begin{cases} A \to c|d|cB|dB \\ B \to a|b|aB|bB \end{cases}$

例 6.4. $A \to A\alpha_1|A\alpha_2|A\alpha_3|\beta_1|\beta_2 = \begin{cases} A \to \beta_1|\beta_2|\beta_1 B|\beta_2 B \\ B \to \alpha_1|\alpha_2|\alpha_3|\alpha_1 B|\alpha_2 B|\alpha_3 B \end{cases}$

例 6.5. $\begin{cases} S \to ABS|BAA \\ A \to BB|BAA \\ B \to aB|b \end{cases} = \begin{cases} S \to aBBBS|bBBS|aBAABS|bAABSaBAA|bAA \\ A \to aBB|bB|aBAA|bAA \\ B \to aB|b \end{cases}$

步骤：

1. 给变量排序
2. 从 A_1 到 A_n 逐一使产生式满足如下要求：

   $A_i \to A_h\alpha, j \geq i$
3. 从 A_{n-1} 开始通过回代，逐一使 $A_{n-1}, A_{n-2} \dots$ 的产生式满足要求
4. 通过代入，使第二步中引入的新变量的产生式满足要求。

   关键：去左递归

   如：

$$\begin{cases} A \to A\alpha_1|A\alpha_2|\dots|A\alpha_n \\ A \to \beta_1|\beta_2|\dots|\beta_m \end{cases}$$

为所有 A 产生式，且 $\beta_1, \beta_2, \dots, \beta_m$ 的首字母不是 A，可以用如下的产生式组替代：

$$\begin{cases} A \to \beta_1|\beta_2|\dots|\beta_m|\beta_1 A'|\beta_2 A'|\dots|\beta_, A' \\ A' \to \alpha_1 A'|\alpha_2 A'|\dots|\alpha_n A'|\alpha_1|\alpha_2|\dots|\alpha_n \end{cases}$$

定理 6.5. 对于 $\forall$ 化简了的 *CFG*，$\exists$ *GNF* 与之等价。

第七章 PDA 下推自动机

7.1 PDA 的定义

定义 7.1. *PDA* $M = (Q, \Sigma, \Gamma, \delta, q_0, z_0, F)$ 其中：

Q ： 状态的有穷集合

Σ ： 输入字母表

Γ ： 栈符号的非空有穷集

δ ： 状态转义函数

$Q \times (\Sigma \cup \{\epsilon\}) \times \Gamma \to 2^{Q \times \Gamma^*}$。

$\forall (q, a, A) \in Q \times (\Sigma \cup \{\epsilon\}) \times \Gamma$，$\delta(q, a, A) = \{(p_1, \gamma_1), \dots, (p_k, r_k)\}$ 表示 M 在状态 q，栈顶为 A 时读到 a，将栈顶符号 A 弹出，将 γ_i 依次压入栈，并将状态改为 p_i。

$\forall (q, A) \in Q \times \Gamma$，$\delta(q, \epsilon, A) = \{(p_1, \gamma_1), \dots, (p_k, r_k)\}$ 表示 M 在状态 q，栈顶为 A 时做空移动，将栈顶符号 A 弹出，将 γ_i 依次压入栈，并将状态改为 p_i。

q_0 ： 开始状态

$z_0 \in \Gamma$ ： 栈底符号

F ： 终止状态

定义 7.2. *PDA* M 的 *ID* $(q, x, \alpha) = Q \times \Sigma^* \times \Gamma^*$

其中 q 是 M 的当前状态，x 是 M 的输入带上剩余串，α 是 M 的栈中当前的内容。

设 M 当前的 *ID* 是 $(q, ax, A\alpha)$，如果 $(p, \gamma) \in \delta(q, a, A)$，则 M 的 *ID* 变为 $(p, x, \gamma\alpha)$，记作 $(q, ax, A\alpha) \vdash_M (p, x, \gamma\alpha)$。

如果 $(p, \gamma) \in \delta(q, \epsilon, A)$，则 M 的 *ID* 变为 $(p, ax, \gamma\alpha)$，记作 $(q, ax, A\alpha) \vdash_M (p, ax, \gamma\alpha)$。

$\vdash_M$ 是 $Q \times \Sigma^* \times \Gamma^*$ 上的二元关系。

定义 7.3. $M = (Q, \Sigma, \Gamma, \delta, q_0, z_0, F)$

用终态识别的语言 $L(M) = x|(q_0, x, z_0) \vdash^* (q, \epsilon, \alpha)$ *and* $q \in F$。

用空栈识别的语言 $N(M) = x|(q_0, x, z_0) \vdash^* (q, \epsilon, \epsilon)$ *and* $q \in F$。