形式语言与自动机理论笔记

Leo Lu

2021 年 10 月 17 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 Suffix” 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:

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

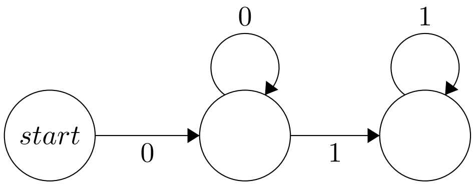
Kleene closure of a language:

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

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
    * $\forall x\in S,x0\in S$
    * $S\to 0$
    * $S\to S0$
  - $\{0^n1^m|n,m\geq 1\}$
    * $0\in S$
      $\forall x\in S,0x,x1\in S$
    * $S\to 01$
      $S\to 0S|S1$
    * $S\to S_1S_2$
      $S_1\to 0$
      $S_1\to S_10$
      $S_2\to 0$
      $S_2\to S_20$
  - $\{0,1\}^*$
    * $\epsilon\in S$
      $\forall x\in S,0x,1x\in S$

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

- 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:

$\xRightarrow[G]{n} = \left(\underset{G}{\Rightarrow}\right)^n$

$\xRightarrow[G]{*} = \left(\underset{G}{\Rightarrow}\right)^*$

$\xRightarrow[G]{=} = \left(\underset{G}{\Rightarrow}\right)^+$

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

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 $V_1 \cup V_2 = \emptyset$，$S \notin V_1 \cup V_2$
   $S_1 \Rightarrow a_1A_1$
   $\Rightarrow a_1a_2 \ldots a_{n-1}A_n$ 所以，我们需要改造 P_1。
   $\Rightarrow a_1a_2 \ldots a_nS_2$
   $G = (V_1 \cup V_2 \cup S, T, P_1' \cup P_2 \cup P_3, S_1)$
   $P_1' = \{A \to aB | A \to aB \in P_1\}$
   $P_3 = \{A \to aS_2 | A \to a \in P_1\}$
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

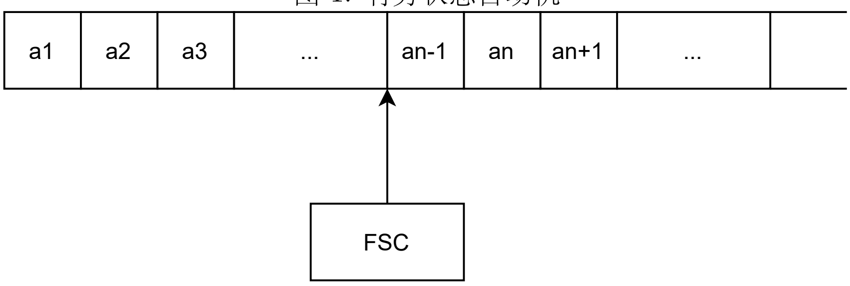
习题：p67 3，4，8.2，8.6，10.3，11.3

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：$L(M) = \{x | \delta(q_0, x) \in F\}$

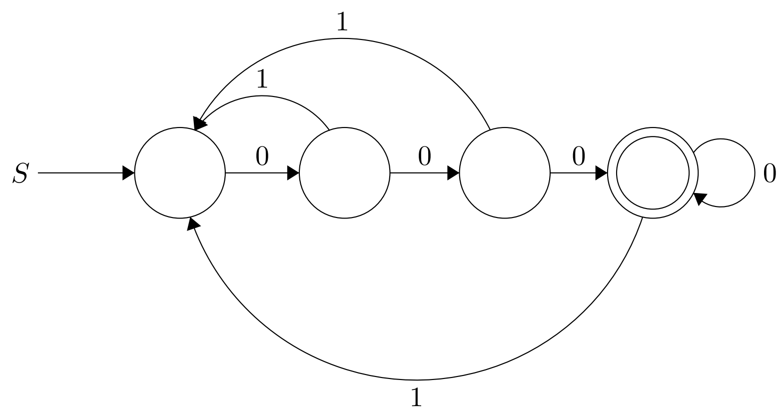
定义 3.4. 对于有穷状态自动机 M_1, M_2：若有 $L(M_1) = L(M_2)$，则称二者等价。

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。

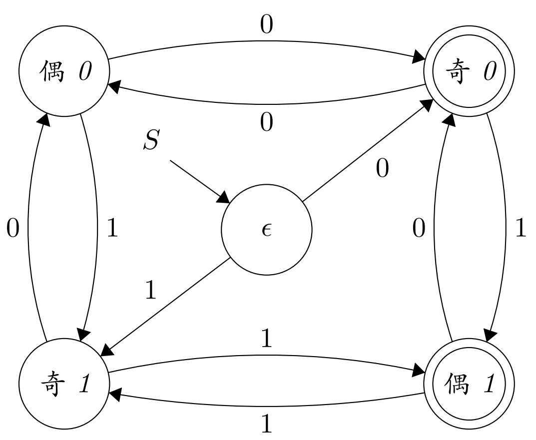
定义 3.6.

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

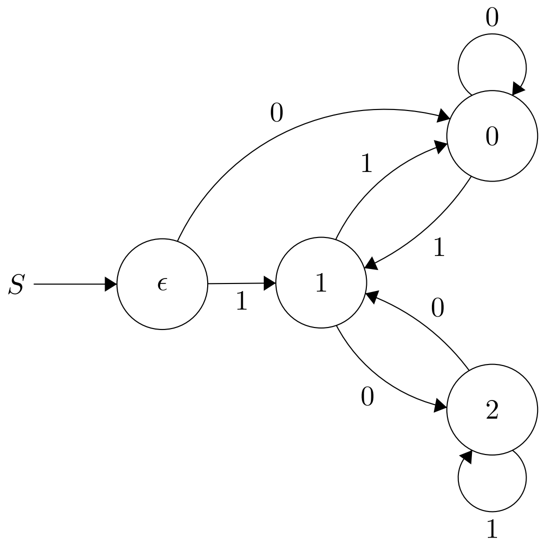
例 3.1. 构造 M，使得 $L(M) = \left\{x000 | x \in \{0,1\}^*\right\}$。



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，$x \mod 3=1\}$。



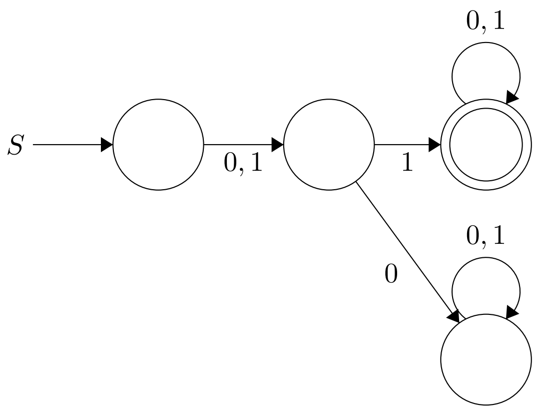
例 3.4. 对于 *FSA* $M=(Q,\Sigma,\delta,q_0,F)$，构造 *FA* M' 使得 $L(M')=\Sigma^*-L(M)$

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

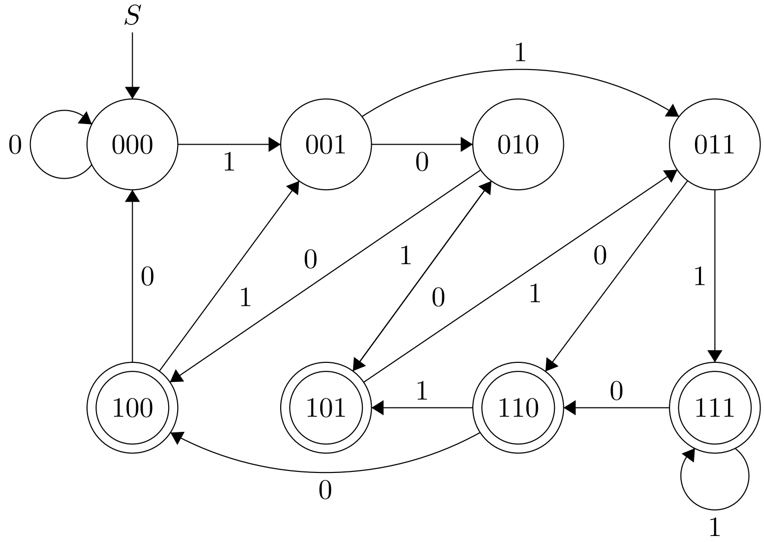
例 3.5. 给定 $\begin{cases} M_1=(Q_1,\Sigma,\delta_1,q_{01},F_1) \\ M_2=(Q_2,\Sigma,\delta_2,q_{02},F_2) \end{cases}$ 构造：

- 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 $L(M)=\Big\{x|x\in\{0,1\}^+\text{and } x$ 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 $\hat{\delta}:Q\times\Sigma^*\to 2^Q$

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

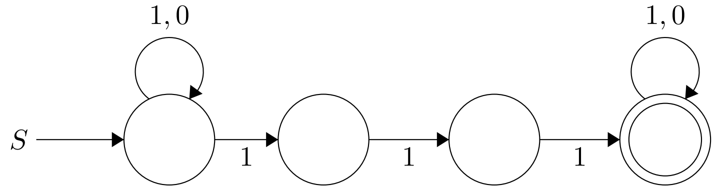
进一步拓展：$\hat{\delta}:2^Q\times\Sigma^*\to 2^Q$

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

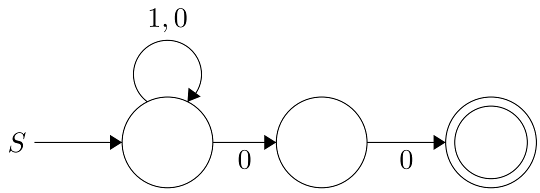
注意到 $Q\times\Sigma\subset 2^Q\times\Sigma^*$，且对 $\forall(q,a)\in Q\times\Sigma^*$，$\hat{\delta}(q,a)=\delta(q,a)$ 所以不用区分 δ 与 $\hat{\delta}$。

δ 是 $Q\times\Sigma^*\to Q$ 上的映射。

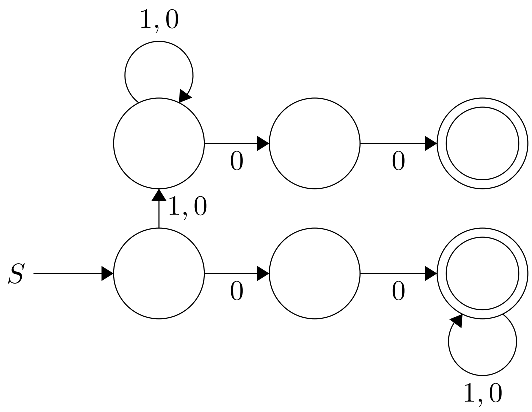
例 3.8. 构造 *NFA M*，使得 $L(M)=\{x|x$ 是含 *110* 的 *01* 串$\}$



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。

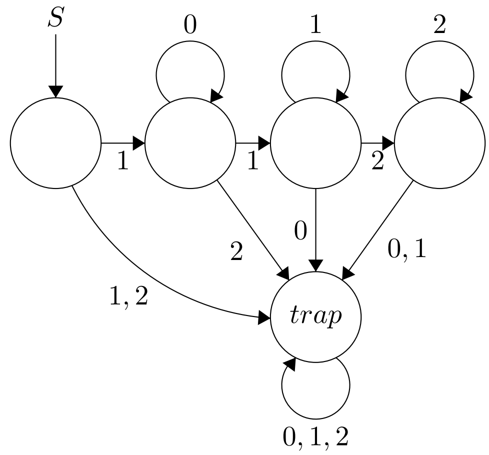
证：对于 *NFA* $M=(Q,\Sigma,\delta,q_0,F)$，构造 *DFA* $M'=(2^Q,\Sigma,\delta',[q_0],F')$

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

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

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

例 3.11. 构造 *DFA M*，$L(M)=\{0^n1^m2^k|n,m,k\geq1\}$



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。$\delta:Q\times(\Sigma\cup\{\epsilon\})\to2^Q$

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 $RG\ G = (Q, \Sigma, P, q_0)$

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

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

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

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