



2023年暑期研修班 感谢聊城大学矩阵半张量积理论与应用研究中心的邀请

有限状态机的代数建模与分析

张志鹏

天津工业大学

2023年 8月15 日

提 纲

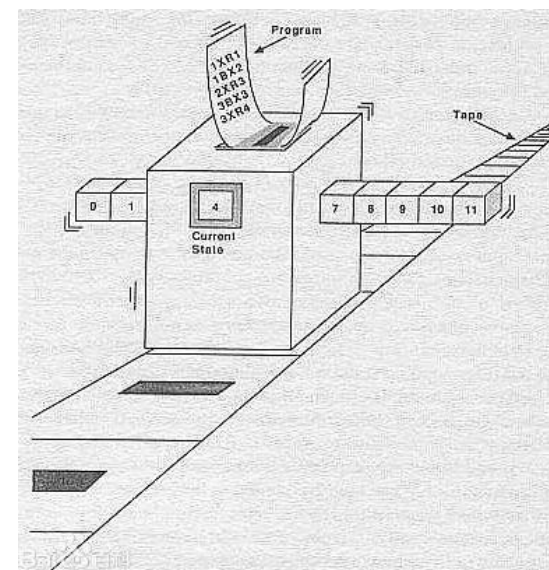
- 一、有限自动机的发展历史
- 二、语言与有限自动机
- 三、有限状态机的代数建模与分析
 - A. 带输出不确定有限状态机
 - B. 受控有限状态机
 - C. 概率有限状态机
- 四、进阶导读
- 五、习题与思考

一、有限自动机的发展历史

1936年英国数学家**艾伦·麦席森·图灵**提出一种抽象自动机，称为图灵机。为了模拟人的这种运算过程，图灵构造出一台假想的机器，该机器由以下几个部分组成：

它有一条**无限长的纸带**，纸带分成了一个一个小方格，每个方格有不同的颜色。

有一个**读写头**在纸带上移来移去。在每个时刻，读写头都要从当前纸带上读入一个方格信息，然后**控制器**根据程序制定的状态转移规则，决定读写头的具体操作，并转换内部状态，然后进行移动。如果进入拒绝和接收状态立即停机；



注意这个机器的每一部分都是有限的，但它有一个潜在的无限长的纸带，因此这种机器只是一个理想的设备。图灵认为这样的一台机器就能模拟人类所能进行的任何计算过程。

一、有限自动机的发展历史

图灵机具备了今天计算机的所有能力，至少在计算能力上是这样的。在给出通用图灵机的同时，图灵就指出，**通用图灵机在计算时，其“机械性的复杂性”是有临界限度的**，超过这一限度，就要靠增加程序的长度和存贮量来解决。进而有了后来计算机科学中计算复杂性理论中两个重要问题：

1、**计算机到底能做什么？**这种研究称为“可判定性”，计算机能解决的问题称为“可判定的”。图灵机精确地描述一条界线，这条界线区分计算机能做什么和不能做什么；图灵的结论不仅适用于抽象的图灵机，也适用于今天的真实计算机。

2、**计算机能有效地做什么？**一类问题是计算机能有效解决的，另一类问题是计算机原则上能解决，但实际上要花费太长时间，以致于除了非常小的问题实例以外，计算机是毫无用处的。后一类问题称为“难解的”或“NP-难的”。计算机硬件一直遵循着计算速度呈指数增长的规律(摩尔定律)，但这也不太可能显著地影响解决难解问题大实例的能力。

一、有限自动机的发展历史

艾伦·麦席森·图灵（Alan Mathison Turing，1912年6月23日~1954年6月7日），英国数学家、逻辑学家，被称为计算机科学之父，人工智能之父。

1950年他发表论文《计算机器与智能》（Computing Machinery and Intelligence），为后来的人工智能科学提供了开创性的构思。提出著名的“图灵测试”，指出如果第三者无法辨别人类与人工智能机器反应的差别，则可以论断该机器具备人工智能。

图灵对于人工智能的发展贡献：一是提出了一种用于判定机器是否具有智能的试验方法，即图灵试验，每年都有试验的比赛。二是图灵提出的著名的图灵机模型为现代计算机的逻辑工作方式奠定了基础。



一、有限自动机的发展历史

1935 ~ 1938年苏联数学家B. И. **舍斯塔科夫**和美国应用数学家C.E.**香农**分别独立地应用布尔代数来分析和综合继电器接点电路，建立逻辑自动机理论。

1943年美国神经生理学家W.S.**麦卡洛克**和数学家W.**皮茨**在《关于神经活动中思维的逻辑演算》一文中，提出有限状态系统的形式化研究。

1945年美国数学家**冯·诺伊曼**设计出存储程序式通用电子数字计算机,并研制成功，世称冯·诺伊曼机。冯·诺伊曼机成为现代通用电子数字计算机的主要模式^[注]。

注：近几年，在计算机体系结构研究方面也已经有了重大进展，越来越多的非冯计算机相继出现，如光子计算机、量子计算机、神经生物计算机等等

一、有限自动机的发展历史

1948年美国**数学家N.维纳**在《控制论》一书中对现代自动机作了详细的分析，指出“自动机的功能是变换和处理信息”。这就为自动机理论奠定了基础。

在20世纪40和50年代，许多研究者研究过**更简单类型的机器**，今天称为“**有限自动机**”。起初建议用这些自动机来为人脑功能建立模型，后来发现这些自动机对于各种工程建模等其他目的也极为有用。

一、有限自动机的发展历史

50 ~ 70年代自动机理论得到迅速的发展。

先后提出**有限自动机**(S.C.克林,1951; E.F.穆尔,1956)

非确定型有限自动机(M.拉宾,1959年, M.明斯基,1967) **图灵奖获得者**

概率自动机(C.E.香农,1956; T.L.布思,1965)

模糊自动机(E.S.桑托斯,1968; 傅京孙, 田中, 1969)

细胞自动机 (E.F.古德,1968) 等多种抽象自动机;

建立了**形式语言与自动机**的对应关系 (N.乔姆斯基,1959,1964) ;

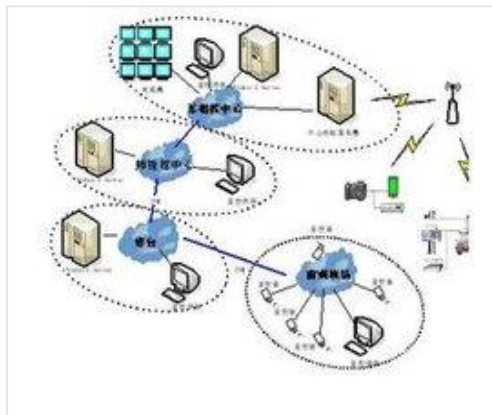
创立了自动机的综合理论 (B.M.格罗什科夫, 1961)、半群理论 (K.克劳恩和J.L.罗兹, 1965) 和可靠性理论 (J.von诺伊曼,1952) 。

一、有限自动机的发展历史

80年代，以制造系统、通信系统、能源系统等人造复杂工程系统为背景，美国哈佛大学何毓琦院士提出离散事件动态系统(DEDS)，研究状态演化基于事件驱动、服从人为逻辑规律的动态系统建模、分析、控制、优化等问题的控制理论分支。**有限自动机成为研究离散事件动态系统逻辑层次的重要模型，用于定性分析。**



柔性制造系统



军事指挥的C3I系统



大规模计算机和通信网络

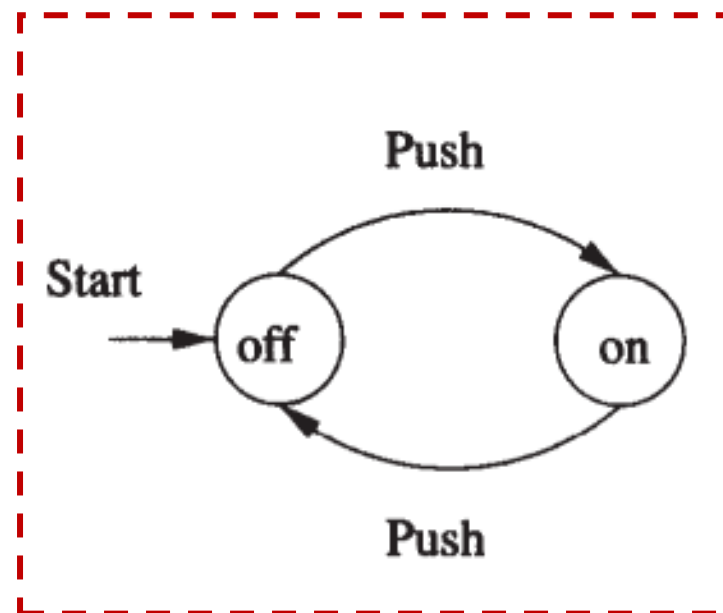


排队论系统

二、语言与有限自动机

例2.1 最简单的非平凡有限自动机是两相开关。这个装置记住处在“开”状态或“关”状态，允许用户按一个按钮，这个按钮根据开关状态起不同作用。

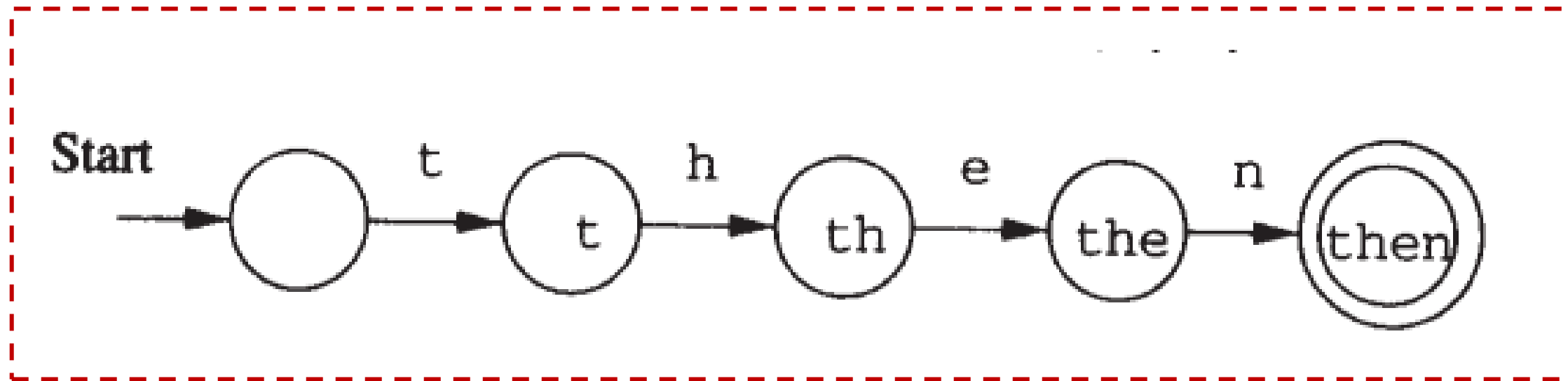
也就是说，如果开关处在关状态，按这个按钮就变成开状态；如果开关处在开状态，按这个按钮就变成关状态。



通常需要说明一个或多个状态是“**终止**”或“**接受**”状态。在经历一个输入序列后进入这些状态之一，说明这个输入序列在某种程度上是好的。例如，可以认为上图中的on状态是接受状态，因为在这个状态下，这个由开关控制的装置将会运行。习惯上用双圆圈来说明接受状态。

二、语言与有限自动机

例2.2 一个状态记忆的内容比一个开关选择要复杂得多。给出另一个有限自动机例子，它可能是**词法分析器**的一部分。这个自动机的任务是**识别关键字then**。因此这个自动机需要5个状态，每个状态表示在单词then中目前已经到达的不同位置。这些位置对应于这个单词的前缀，范围从空串(即一点也没有看见这个单词)到整个单词。



二、语言与有限自动机

语言的基本要素

字符
字母表
字符串

字符串

字母表 Σ : 非空的有穷集合

字符串: Σ 中符号的有穷序列

如 $\Sigma = \{a, b\}$

$a, b, aab, babb$

字符串 ω 的**长度** $|\omega|$: ω 中的字符个数

如 $|a|=1, |aab|=3$

空字符串 ε : 长度为0, 即不含任何符号的字符串

a^n : n 个 a 组成的字符串

Σ^* : Σ 上字符串的全体

子字符串(子串): 字符串中若干连续符号组成的字符串

前缀: 最左端的子串

后缀: 最右端的子串

例如 $\omega = abbaab$

a, ab, abb 是 ω 的前缀

aab, ab, b 是 ω 的后缀

ba 是 ω 的子串, 但既不是前缀, 也不是后缀

ω 本身也是 ω 的子串, 且既是前缀, 也是后缀

ε 也是 ω 的子串, 且既是前缀, 也是后缀

二、语言与有限自动机

字符串的连接运算

设 $\alpha = a_1 a_2 \dots a_n$, $\beta = b_1 b_2 \dots b_m$,

$\alpha\beta = a_1 a_2 \dots a_n b_1 b_2 \dots b_m$ 称作 $\alpha\beta$ 与 β 作的**连接**

如 $\alpha = ab$, $\beta = baa$, $\alpha\beta = abbaa$, $\beta\alpha = baaab$

对任意的字符串 $\alpha\beta$, β , γ

$$(1) (\alpha\beta)\gamma = \alpha(\beta\gamma)$$

即, 连接运算满足结合律

$$(2) \varepsilon\alpha = \alpha\varepsilon = \alpha$$

即, 空串 ε 是连接运算的单位元

n 个 α 的连接记作 α^n

如 $(ab)^3 = ababab$, $\alpha^0 = \varepsilon$

形式语言

定义: Σ^* 的子集称作字母表 Σ 上的**形式语言**, 简称 **语言**

例如 $\Sigma = \{a, b\}$

$$A = \{a, b, aa, bb\}$$

$$B = \{a^n \mid n \in N\}$$

$$C = \{a^n b^m \mid n, m \geq 1\}$$

$$D = \{\varepsilon\}$$

空语言 $\emptyset$

二、语言与有限自动机

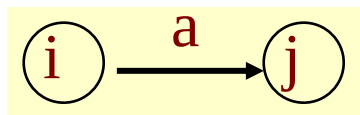
• 确定型有限自动机

定义2.1 确定型有限自动机(DFA)是一个有序5元组 $A = \langle X, \Sigma, \delta, x_0, X_m \rangle$, 其中

- (1) **状态集合** X : 非空有限集合
- (2) **输入字母表** Σ : 非空有限集合
- (3) **初始状态** $x_0 \in X$
- (4) **接受状态集** $X_m \subseteq X$
- (5) **状态转移函数** $\delta: X \times \Sigma \rightarrow X$



$\delta(i, a) = j$ 或



$i, j \in X, a \in \Sigma;$

即:

状态 i 遇符号 a 时变换为状态 j

二、语言与有限自动机

- 确定型有限自动机

- 表现形式

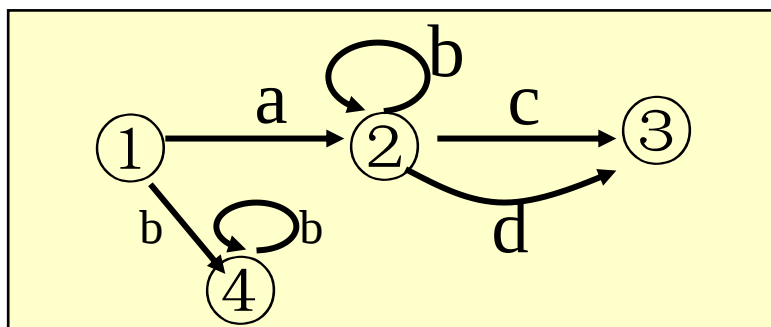
例2.3 确定型有限自动机： $A=(X, \Sigma, x_0, X_m, \delta)$

其中： $Q=\{1,2,3,4\}, \Sigma=\{a,b,c,d\}, S=\{1\},$

δ : $\delta(1,a)=2; \delta(1,b)=4; \delta(2,b)=2;$
 $\delta(2,c)=3; \delta(2,d)=3; \delta(4,b)=4;$

有限自动机 的两种表现形式:

(1) 状态转换图:



(2) 转换表:

	a	b	c	d
1	2	4		
2		2	3	3
3				
4		4		

※ 变换表结构: 行(状态), 列(符号), 表项(变换后状态)

二、语言与有限自动机

- 确定型有限自动机

- 表现形式

- 接受语言

把 δ 扩张到 $X \times \Sigma^*$ 上 $\delta^*: X \times \Sigma^* \rightarrow X$, 递归定义如下 $\forall x \in X$,
 $a \in \Sigma$ 和 $w \in \Sigma^*$

$$\delta^*(x, \varepsilon) = x$$

$$\delta^*(x, wa) = \delta(\delta^*(x, w), a)$$

定义2.2 $\forall w \in \Sigma^*$, 如果 $\delta^*(x_0, w) \in F$, 则称 **A 接受** w .

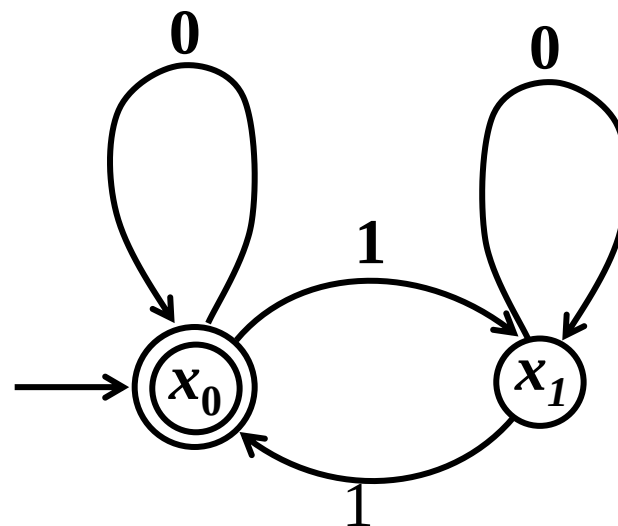
M 接受的字符串的全体称作**A 接受的语言**, 记作
 $L(M)$, 即

$$L(M) = \{ w \in \Sigma^* \mid \delta^*(x_0, w) \in X_m \}$$

二、语言与有限自动机

例2.4 奇偶校验码

$A = \langle \{x_0, x_1\}, \{0, 1\}, \delta, x_0, \{x_0\} \rangle$, $L(M) = \{\omega \in \{0, 1\}^* \mid \omega \text{ 含有偶数个 } 1\}$



- 确定型有限自动机

- 表现形式

- 接受语言

二、语言与有限自动机

- 确定型有限自动机
- 非确定型有限自动机

- 确定有限自动机是有限自动机理论中最基础的模型，它的初始状态只有一个，而且状态转移函数将每个状态至多转移到一个状态，其结构与状态演化相对简单。
- 实际生产过程中，系统状态有时会出现**状态合并**，此时，合并后状态在某事件驱动下可能转移到多个状态。抑或是由于环境、成本等限制，传感器无法测量到系统内部发生的事件，即**状态发生了转移但观测不到有事件发生**，就会导致状态转移的不确定。
- 非确定有限自动机模型更加贴合实际系统，因此其控制问题也一直是研究的重点。

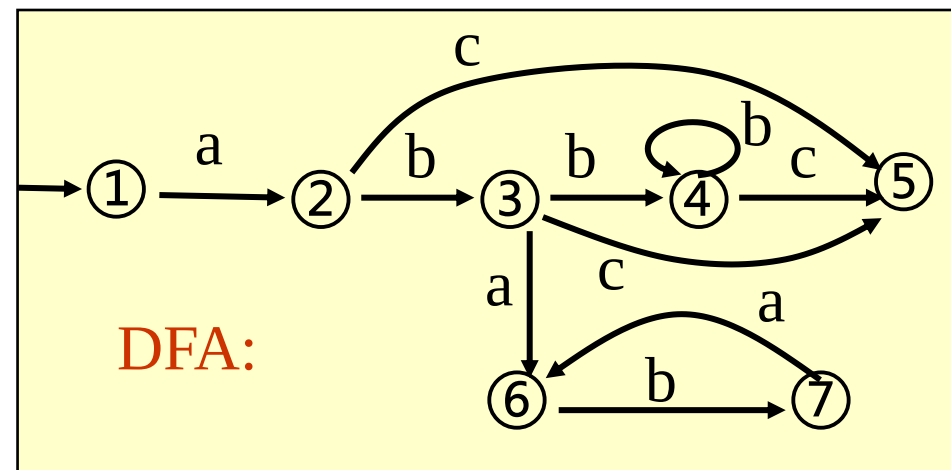
二、语言与有限自动机

- 确定型有限自动机
- 非确定型有限自动机

$L = \{ ab^n c, (ab)^n | n \geq 0 \}$

※ 确定型有限自动机如右图所示:

【有限自动机分类】



1. 确定型有限自动机(DFA)

特征: ①开始状态唯一; ②转移函数单值; ③不带 ϵ 边。

2. 非确定型有限自动机(NFA)

-- 不能全部满足上述特征者!

- (1) 不带有 ϵ 边的非确定有限自动机
- (2) 带有 ϵ 边的非确定有限自动机

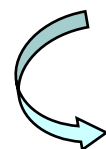
二、语言与有限自动机

- 确定型有限自动机
 - 非确定型有限自动机
- 不带有 ε 边

定义2.2 非确定型有限自动机(NFA)是一个有序5元组

$A = \langle X, \Sigma, \delta, X_0, X_m \rangle$, 其中

- (1) **状态集合** X : 非空有限集合
- (2) **输入字母表** Σ : 非空有限集合
- (3) **初始状态集** $X_0 \in X$
- (4) **接受状态集** $X_m \subseteq X$
- (5) **状态转移函数** $\delta: X \times \Sigma \rightarrow 2^X$ 或 $\delta: X \times \{\Sigma + \{\varepsilon\}\} \rightarrow 2^X$



2^X 表示状态集合 X 的幂集

这里的 X, Σ, X_m 的定义与 DFA 的相同

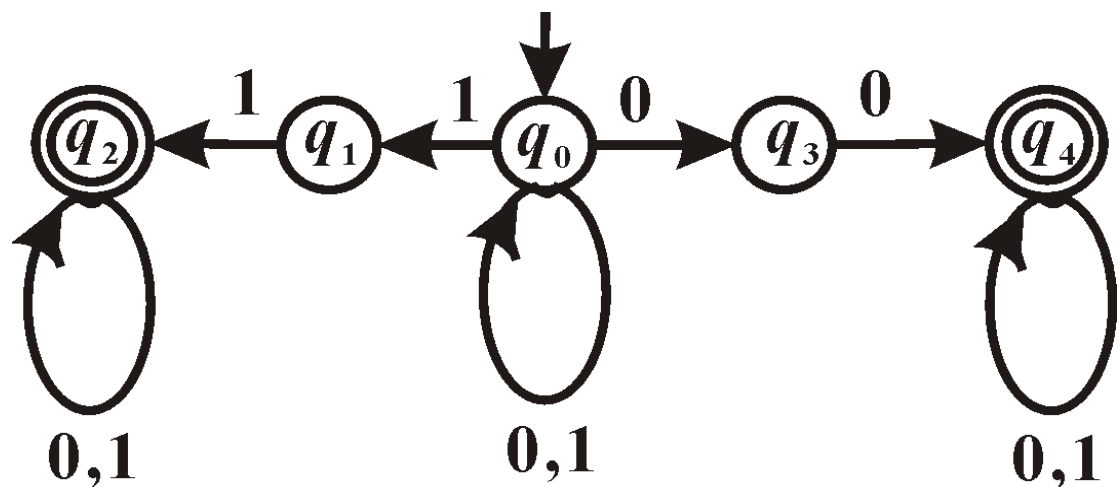
二、语言与有限自动机

- 确定型有限自动机
- 非确定型有限自动机

➤ 不带有 ε 边

例2.5 一台NFA

δ	$\rightarrow q_0$	q_1	$*q_2$	q_3	$*q_4$
0	$\{q_0, q_3\}$	$\emptyset$	$\{q_2\}$	$\{q_4\}$	$\{q_4\}$
1	$\{q_0, q_1\}$	$\{q_2\}$	$\{q_2\}$	$\emptyset$	$\{q_4\}$



二、语言与有限自动机

- 确定型有限自动机
- 非确定型有限自动机

➤ 不带有 ε 边

➤ 带有 ε 边

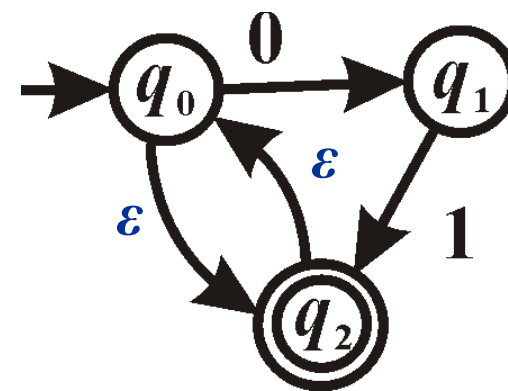
带 ε 转移的非确定型有限自动机

ε 转移: 不读任何符号, 自动转移状态.

ε -NFA: $\delta: X \times (\Sigma \cup \{\varepsilon\}) \rightarrow 2^X$

例2.6 ε -NFA

δ	0	1	ε
$\rightarrow q_0$	$\{q_1\}$	$\emptyset$	$\{q_2\}$
q_1	$\emptyset$	$\{q_2\}$	$\emptyset$
$*q_2$	$\emptyset$	$\emptyset$	$\{q_0\}$



$$L(M) = \{ (01)^n \mid n \geq 0 \}$$

二、语言与有限自动机

- 确定型有限自动机
- 非确定型有限自动机
- 有限自动机等价性

※ 有限自动机的等价转换,主要包含两个内容:

(1) 有限自动机的确定化($NFA \Rightarrow DFA$);

非确定机(NFA) 较易构造, 但不好用!

确定机 (DFA) 较难构造, 但好用!

任何一非确定机NFA, 皆可通过有效算法将其转化为等价的确定机DFA (二者描述的语言相同)。

(2) 有限自动机的最小化($DFA \Rightarrow$ 最小的DFA);

有限自动机的最小化,又称有限自动机的化简; 是指:
对给定的确定机DFA1,构造另一个确定机DFA2,
使得 $L(DFA1)=L(DFA2)$,且DFA2的状态最少。

二、语言与有限自动机

- 确定型有限自动机
- 非确定型有限自动机
- 有限自动机等价性

DFA与NFA的等价性

定理2.2 对每一个NFA A 都存在DFA A' 使得 $L(A)=L(A')$

证明：用 $A'=\langle X', \Sigma, \delta', x_0', X_m' \rangle$ 模拟 $A=\langle X, \Sigma, \delta, x_0, X_m \rangle$

$$X'=2^X, x_0'=\{x_0\}$$

$$X_m'=\{X_1 \subseteq X \mid X_1 \cap X_m \neq \emptyset\}$$

$\forall X_1 \subseteq X$ 和 $a \in \Sigma$,

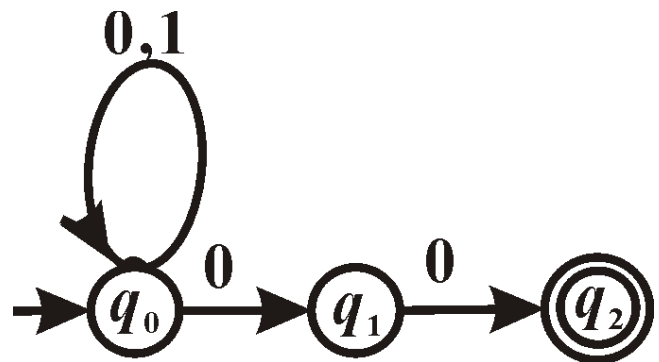
$\delta'(X_1, a)$ 是所有 $p \in X_1, \delta(p, a)$ 并起来

二、语言与有限自动机

- 确定型有限自动机
- 非确定型有限自动机
- 有限自动机等价性

例2.6 NFA A

δ	0	1
$\rightarrow q_0$	$\{q_0, q_1\}$	$\{q_0\}$
q_1	$\{q_2\}$	$\emptyset$
$*q_2$	$\emptyset$	$\emptyset$



DFA A'

δ'	0	1
$\rightarrow\{q_0\}$	$\{q_0, q_1\}$	$\{q_0\}$
$\{q_1\}$	$\{q_2\}$	$\emptyset$
$*\{q_2\}$	$\emptyset$	$\emptyset$
$\{q_0, q_1\}$	$\{q_0, q_1, q_2\}$	$\{q_0\}$
$*\{q_0, q_2\}$	$\{q_0, q_1\}$	$\{q_0\}$
$*\{q_1, q_2\}$	$\{q_2\}$	$\emptyset$
$*\{q_0, q_1, q_2\}$	$\{q_0, q_1, q_2\}$	$\{q_0\}$
$\emptyset$	$\emptyset$	$\emptyset$

二、语言与有限自动机

- 确定型有限自动机
- 非确定型有限自动机
- 有限自动机等价性

用DFA模拟 ε -NFA

定理2.3 对每一个 ε -NFA A 都存在DFA A' , $L(A)=L(A')$

即: DFA, NFA 和 ε -NFA 接受同一个语言类

证明: 设 ε -NFA $A=\langle X, \Sigma, \delta, x_0, X_m \rangle$, $x \in X$

x 的 ε 闭包 $E(x)$: 从 x 出发, 经过 ε 转移能够到达的所有状态, 递归定义如下

- (1) $E(x)$ 包含 x ;
- (2) 如果 $x \in E(x)$, 则 $\delta(x, \varepsilon) \subseteq E(x)$.

模拟的方法与用DFA模拟不带 ε 的NFA的方法基本相, 只是要用 $E(x)$ 代替 x .

二、语言与有限自动机

- 确定型有限自动机
- 非确定型有限自动机
- 有限自动机等价性

用DFA模拟 ε -NFA

定理2.3 对每一个 ε -NFA A 都存在DFA A' , $L(A)=L(A')$

即: DFA, NFA 和 ε -NFA 接受同一个语言类

证明: (续) 用确定型有限自动机 $A' = \langle X', \Sigma', \delta', x_0', X_m' \rangle$
模拟 ε -NFA $A = \langle X, \Sigma, \delta, x_0, X_m \rangle$ $X' = 2^X$, $x_0' = E(x_0)$

$$X_m' = \{X_1 \in Q \mid X_1 \cap X_m \neq \emptyset\}$$

$\forall X_1 \in X$ 和 $a \in \Sigma$,

$$\delta(A, a) = \bigcup_{q \in A} \{r \in E(t) \mid \exists p, t \text{ 使 } p \in E(q), t \in \delta(p, a)\}$$

构造DFA A' 时不需要对不可达状态进行计算,做法如下:
从 $x_0' = E(x_0)$ 开始, 对每一个 $a \in \Sigma$ 计算 δ' 的值, 然后对每一个新出现的子集计算 δ' 的值, 重复进行, 直至没有新的子集出现为止.

二、语言与有限自动机

- 确定型有限自动机
- 非确定型有限自动机
- 有限自动机等价性

例2.7

ε -NFA A

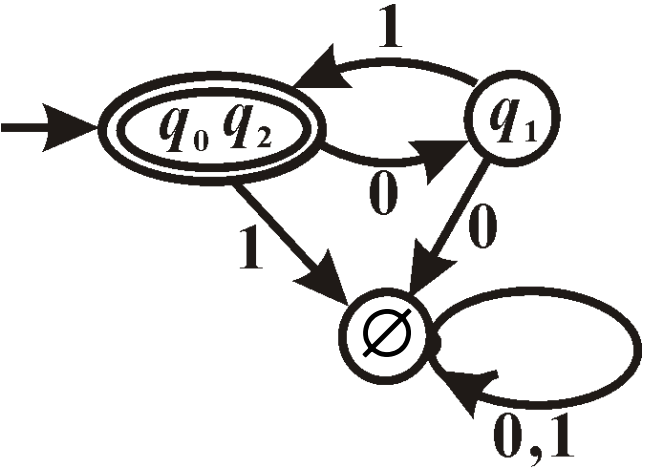
δ	0	1	ε
$\rightarrow q_0$	$\{q_1\}$	$\emptyset$	$\{q_2\}$
q_1	$\emptyset$	$\{q_2\}$	$\emptyset$
$*q_2$	$\emptyset$	$\emptyset$	$\{q_0\}$

用DFA模拟 ε -NFA

q	$E(q)$
q_0	$\{q_0, q_2\}$
q_1	$\{q_1\}$
q_2	$\{q_0, q_2\}$

DFA A'

δ'	0	1
$\rightarrow^* \{q_0, q_2\}$	$\{q_1\}$	$\emptyset$
$\{q_1\}$	$\emptyset$	$\{q_0, q_2\}$
$\emptyset$	$\emptyset$	$\emptyset$



三、有限状态机的代数建模与分析

• 带输出的有限状态机

定义3.1: 带输出的有限自动机表示一个成七元组

$\mathcal{A} = (X, Y, \Sigma, f, h, x_0, X_m)$, 其中:

- (1) **状态集合** X : 非空有限集合
- (2) **输入字母表** Σ : 非空有限集合
- (3) **输出字母表** Y : 非空有限集合
- (3) **初始状态**: $x_0 \in X$
- (4) **接受状态集** $X_m \subseteq X$
- (5) **状态转移函数** $f: X \times \Sigma \rightarrow 2^X$ 或 $\delta: X \times \{\Sigma + \{\varepsilon\}\} \rightarrow 2^X$
- (6) **输出函数** h

三、有限状态机的代数建模与分析

- 带输出的有限状态机

- Moore机

根据输出与输入、系统状态的关系，有限状态机又可分为**Moore型有限状态机**和**Mealy型有限状态机**。

Moore型有限状态机是指输出仅与系统状态有关，与输入信号无关的状态机。由摩尔在 1956 年提出，其优点是将输入和输出分隔开。

其输出函数定义为 $h: X \rightarrow Y$

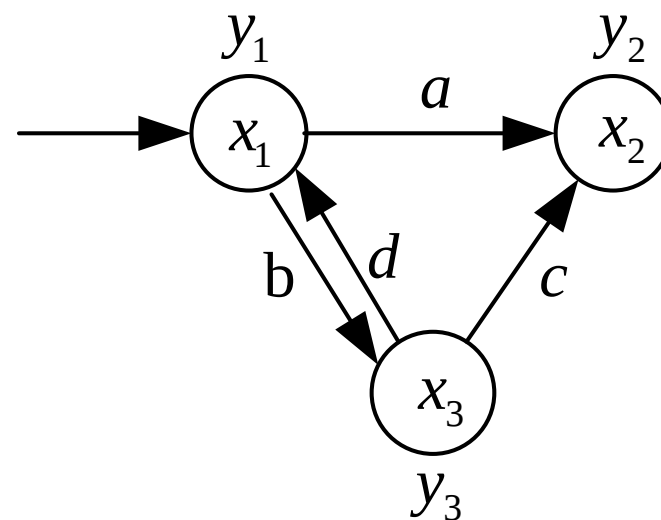
三、有限状态机的代数建模与分析

- 带输出的有限状态机

- Moore机

• **例3.1** 考虑带输出的有限自动机, 状态集 $X = \{x_1, x_2, x_3\}$, 输出集 $Y = \{y_1, y_2, y_3\}$, 事件集 $\Sigma = \{a, b, c, d\}$ 。其中状态转移, 输出以及状态转移图表示如下:

- $f(x_1, a) = x_2$
- $f(x_1, b) = x_3$
- $f(x_3, c) = x_2$
- $f(x_2, d) = x_1$
- $h(x_1) = y_1$
- $h(x_2) = y_2$
- $h(x_3) = y_3$

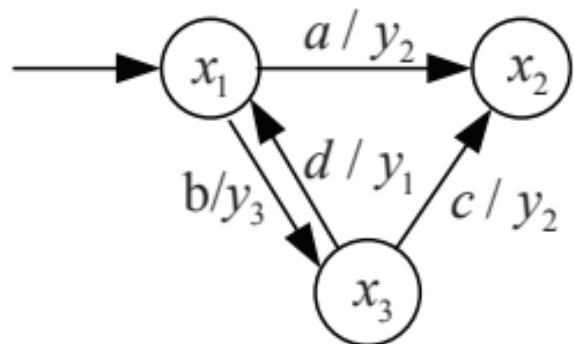


三、有限状态机的代数建模与分析

- 带输出的有限状态机

- Moore机

- Mealy机



Mealy型有限状态机是指输出与系统状态和输入均有关系的有限状态机。由米兰在1955年提出。**其输出函数定义为 $h: X \times \Sigma \rightarrow Y$ 。**

对于Mealy型有限状态机在有向曲线段上的字符表示系统的输入和输出，用“/”分隔。

Mealy型有限状态机和Moore型有限状态机之间是可以相互转换的。

三、有限状态机的代数建模与分析

- 带输出的有限状态机

- Moore机

- Mealy机

- 代数建模 (以Moore机为例)

带输出的Moore有限自动机

$$\mathcal{A} = (X, Y, \Sigma, f, h, x_0, X_m),$$

在半张量积的框架下, 其状态集, 输入符号集和输出符号集可以重新定义为 $x_i \sim \delta_n^i, y_j \sim \delta_p^j, e_k \sim \delta_m^k$, 有限自动机的状态转移结构矩阵定义为:

$$F := [F_1, F_2, \dots, F_m] \in \mathcal{B}_{n \times mn}$$

其中 $F_k \in \mathcal{B}_{n \times n}$ 可定义为:

$$F_{k(l,i)} = \begin{cases} 1, & \delta_n^l \in f(\delta_n^i, \delta_m^k) \\ 0, & otherwise \end{cases}$$

三、有限状态机的代数建模与分析

- 带输出的有限状态机

- Moore机

- Mealy机

- 代数建模 (以Moore机为例)

同样，输出结构矩阵定义为：

$$H \in \mathcal{B}_{p \times n}$$

其中 $H \in \mathcal{B}_{n \times n}$ 可定义为：

$$H_{(j,i)} = \begin{cases} 1, & \delta_p^j \in h(\delta_n^i) \\ 0, & otherwise \end{cases}$$

则系统的代数状态空间表示为：

$$\begin{cases} x(t+1) = Fu(t)x(t) \\ y(t) = Hx(t) \end{cases} \quad (3.1)$$

其中， $x(t)$ 是从初始状态 $x(0)$ 出发在 t 步到达的状态， $y(t)$ 是相对应的状态输出， $u(t)$ 是事件的向量表示。 F 和 H 分别表示系统的状态转移结构矩阵和输出结构矩阵。

三、有限状态机的代数建模与分析

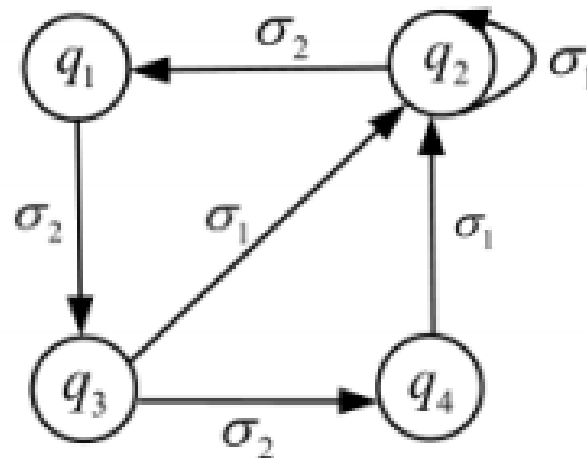
例3.2

- 带输出的有限状态机

- Moore机

- Mealy机

- 代数建模 (以Moore机为例)



$$q_1 = f(q_2, \sigma_2)$$



$$\begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} = f\left(\begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \end{pmatrix}\right)$$



$$F = \left[\begin{array}{cccc|cccc} 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{array} \right]$$

三、有限状态机的代数建模与分析

- 带输出的有限状态机

在验证系统状态能否到达平衡状态，并不需要知道有多少路径能够到达，因此，我们引入布尔-半张量积运算，对模型进一步改进，得到如下新的Moore有限自动机的代数方程：

$$\begin{cases} x(t+1) = \hat{F} \ltimes_B u(t) \ltimes_B x(t) \\ y(t) = \hat{H} \ltimes_B x(t) \end{cases} \quad (3.2)$$

三、有限状态机的代数建模与分析

- 带输出的有限状态机
- 受控有限状态机

- 相比于一般的有限自动机，受控有限状态机增加了**事件的控制模式**。假如模型中某些事件是不可控的，那么该事件在整个系统运行过程中不能被禁止，即恒发生的。
- 受控有限状态机的建模与输出分析在计算机科学与工程领域有着重要的应用，利用它们可以讨论系统的弹性和错误恢复等性质。在控制领域的研究，如状态重构问题、可观测性等。

三、有限状态机的代数建模与分析

- 带输出的有限状态机
- 受控有限状态机

定义3.2: 受控的带输出有限状态机表示一个成七元组

$\mathcal{A} = (X, Y, \Sigma, f, h, x_0, X_m)$, 其中:

- (1) **状态集合** X : 非空有限集合
- (2) **输入字母表** $\Sigma = \Sigma_c \cup \Sigma_{\{uc\}}$
- (3) **输出字母表** Y : 非空有限集合
- (3) **初始状态**: $x_0 \in X$
- (4) **接受状态集** $X_m \subseteq X$
- (5) **状态转移函数** $f: X \times \Sigma \rightarrow 2^X$ 或 $\delta: X \times \{\Sigma + \{\varepsilon\}\} \rightarrow 2^X$
- (6) **输出函数** h

三、有限状态机的代数建模与分析

- 带输出的有限状态机
- 受控有限状态机

集合 Σ 可以划分成两部分 $\Sigma = \Sigma_c \cup \Sigma_{\{uc\}}$ ，其中 Σ_c 表示可控事件集，即该集合内事件存在两种状态：使能状态和禁止状态； $\Sigma_{\{uc\}}$ 表示**不可控事件集**，即该集合内事件不可被禁止；

为了方便控制，记函数

$$\gamma: \Sigma \rightarrow \{0,1\}$$

为控制模式，该控制模式可以描述可控事件与不可控事件，即如果该事件为可控事件 $e \in \Sigma_c$ ，则用 $\gamma(e) = 1$ 和 $\gamma(e) = 0$ 分别表示该事件处于使能状态和禁止状态；如果该事件为不可控事件 $e \in \Sigma_{\{uc\}}$ ，则用 $\gamma(e) = 1$ 表示该事件一直处于使能状态。

三、有限状态机的代数建模与分析

- 带输出的有限状态机
- 受控有限状态机

在此，我们定义一个新的状态转移函数

$$f_c: X \times \gamma \times \Sigma \rightarrow 2^{\{X\}},$$

称作受控状态转移函数，定义如下：

$$f_c(x, \gamma, e) = \begin{cases} f(x, e), & \text{如果 } f(x, e) \neq \emptyset \text{ 且 } \gamma(e) = 1, \\ \emptyset, & \text{否则.} \end{cases}$$

则受控有限状态机可写成如下形式：

$$A_c = (X, Y, \gamma \times \Sigma, f_c, h, x_0, X_m)。$$

三、有限状态机的代数建模与分析

- 带输出的有限状态机
- 受控有限状态机

考虑受控有限状态机

$$A_c = (X, Y, \gamma \times \Sigma, f_c, h, x_0, X_m),$$

其中 $X = \{x_1, x_2, \dots, x_n\}$, $Y = \{y_1, y_2, \dots, y_p\}$, $\Sigma = \{e_1, e_2, \dots, e_{\{m\}}\}$ 。在矩阵半张量积的框架下, 设定 $x_i \sim \delta_n^i$ ($i \in [1, n]$), $y_j \sim \delta_p^j$ ($j \in [1, p]$), $e_{\{k\}} \sim \delta_m^{\{k\}}$ ($k \in [1, m]$), 重新定义受控有限状态机的状态转移结构矩阵为:

$$F^c := [F_1^c, F_2^c, \dots, F_s^c]$$

其中 $F_k^c \in \mathcal{M}_{n \times n}$ 可定义如下:

$$F_{k(l,i)}^c = \begin{cases} 1, & \delta_n^l \in f_c(\delta_n^i, \gamma, \delta_m^k) \\ 0, & \text{否则.} \end{cases}$$

三、有限状态机的代数建模与分析

- 带输出的有限状态机
- 受控有限状态机

受控有限状态机动态的代数表示如下：

$$\begin{cases} x(t+1) = F^c \bowtie_B u(t) \bowtie_B x(t) \\ y(t) = H \bowtie_B x(t) \end{cases} \quad (3.3)$$

其中， $x(t)$ 代表从初始状态 $x(0)$ 出发 t 步到达的状态， $y(t)$ 是相对应的状态输出。 $u(t)$ 是输入向量。 H 是输出结构矩阵。

三、有限状态机的代数建模与分析

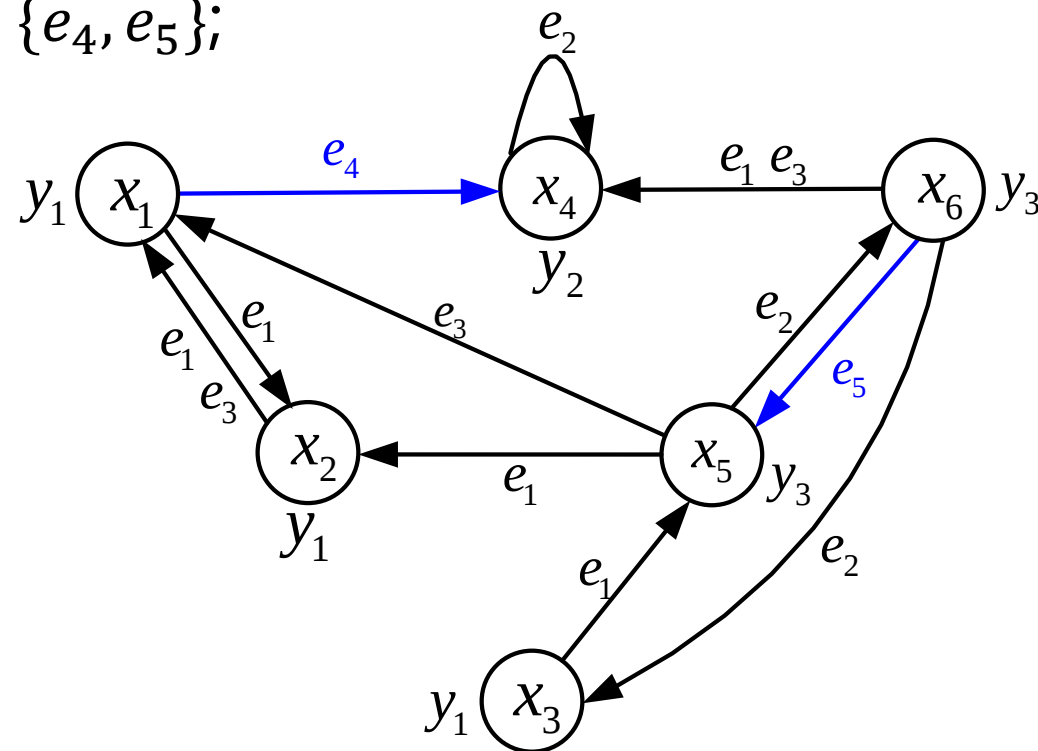
- 带输出的有限状态机
- 受控有限状态机

例3.3

状态集: $X = \{x_1, x_2, x_3, x_4, x_5, x_6\}$;

可控事件集: $\Sigma_c = \{e_1, e_2, e_3\}$;

不可控事件集: $\Sigma_{uc} = \{e_4, e_5\}$;



三、有限状态机的代数建模与分析

- 带输出的有限状态机
- 受控有限状态机
- 概率有限状态机

- 在确定有限自动机中，初始状态是一个单独状态，转移函数是确定的，即接收状态能由发出状态唯一确定，此时，确定有限自动机演化的是确定性系统行为。
- 然而，实际中，许多系统呈现出随机性系统行为。比如，在序列电路中，电路基本元器件的随机性失效会导致整个序列电路呈现随机性演化行为。为了刻画该随机行为，文献[1]最早系统完整提出了概率有限自动机的概念。

[1] Michael O Rabin, 'Probabilistic Automata', *Information and control*, 6 (1963), 230-45.

三、有限状态机的代数建模与分析

- 带输出的有限状态机
- 受控有限状态机
- 概率有限状态机

定义 3.3 概率有限自动机可以用一个四元组 $A^p = (X, \Sigma, P, x^0)$ 表示, 其中

$X = \{x_1, x_2, \dots, x_n\}$ 表示有限状态集,

x^0 **表示初始状态的概率分布向量**;

$\Sigma = \{\sigma_1, \sigma_2, \dots, \sigma_m\}$ 表示有限字母表(事件集);

$P: X \times \Sigma \times X \rightarrow [0, 1]$ 为**状态转移概率函数**, 即对于所有 $x_i, x_j \in X, \sigma_k \in \Sigma$, 从状态 x_i 出发在事件 σ_k 的驱动下, 到达状态 x_j 的概率; 该状态转移概率函数需满足规范性条件:

$$\sum_{\{\sigma_k \in \Sigma\}} \sum_{\{x_j \in X\}} P(x_i, \sigma_k, x_j) = 1$$

或者

$$\sum_{\{\sigma_k \in \Sigma\}} \sum_{\{x_j \in X\}} P(x_i, \sigma_k, x_j) = 0。$$

三、有限状态机的代数建模与分析

- 带输出的有限状态机
- 受控有限状态机
- 概率有限状态机

在矩阵半张量积的框架下，我们对概率有限自动机 $\mathcal{A}^p$ 设计了一个新的结构矩阵，记为 $\mathcal{F}$ ，该矩阵称作转移概率结构矩阵，其定义如下：

$$\mathcal{F}^p = [\mathcal{F}_1, \mathcal{F}_2, \dots, \mathcal{F}_m] \in \tilde{\mathcal{L}}_{n \times mn}^s,$$

其中， $\mathcal{F}_k \in \tilde{\mathcal{L}}_{n \times n}^s$ 是与事件 σ_k 相关的转移概率结构矩阵，可定义如下：

$$\mathcal{F}_{k(j,i)} = \begin{cases} P_{ij}^k, x_j \in f(x_i, \sigma_k) \\ 0, otherwise \end{cases}$$

其中， P_{ij}^k 是状态转移概率； f 是概率有限自动机对应的不确定有限自动机 $\mathcal{A}^{non}$ 的状态转移函数。同时，状态 x_i 和事件 σ_k 的向量表示为 δ_n^i 和 δ_m^k 。

三、有限状态机的代数建模与分析

- 带输出的有限状态机
- 受控有限状态机
- 概率有限状态机

定理3.3: 考虑概率有限自动机 $\mathcal{A}^p = (X, \Sigma, P, x^0)$, 其动态代数状态空间表示如下:

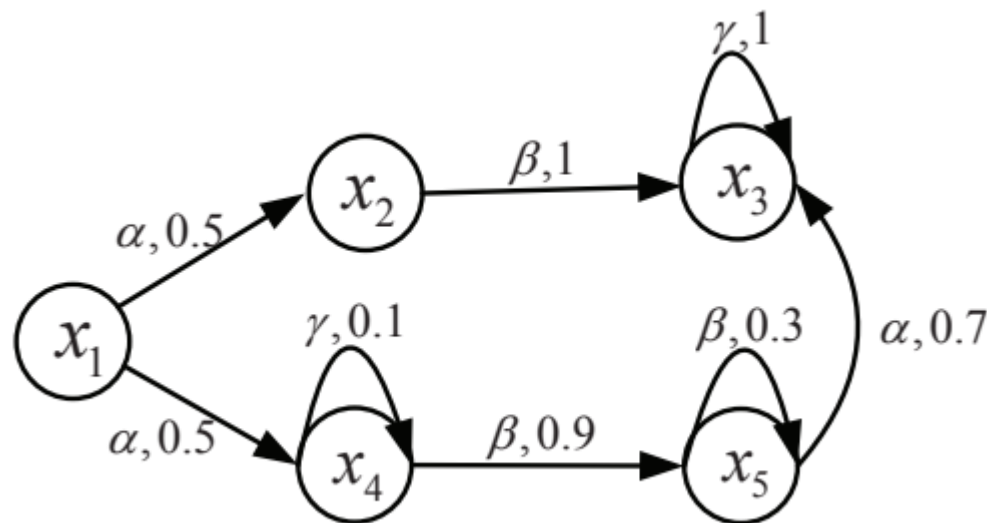
$$E[x(t + 1)] = \mathcal{F}u(t)E[x(t)], \quad (3.4)$$

其中, $x(t)$ 代表从初始状态 $x(0)$ 出发在 t 步到达的状态, $E[x(t)] \in \tilde{\mathcal{L}}_n^s$ 表示到达状态 $x(t)$ 的概率期望值; 初始状态 $x(0)$ 的概率期望值定义为 $E[x(0)] = x^0$; $u(t)$ 表示控制输入; $\mathcal{F}$ 表示转移概率结构矩阵。

三、有限状态机的代数建模与分析

- 带输出的有限状态机
- 受控有限状态机
- 概率有限状态机

例3.4 考虑概率有限自动机 $\mathcal{A}^p = (X, \Sigma, P, x^0)$, 其中, $\sigma_k (k \in [1,3])$ 代表事件 α, β, γ , 设定 $x_i \sim \delta_5^i (i \in [1,5])$ 且 $\sigma_k \sim \delta_3^k (k \in [1,3])$ 。

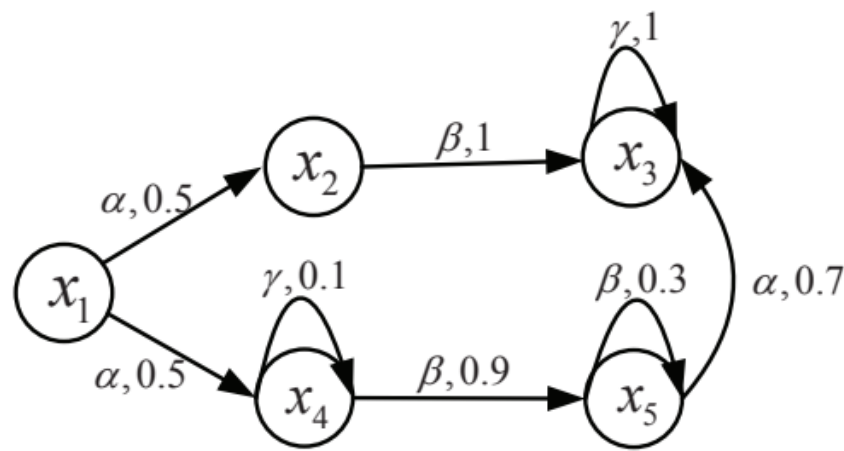


三、有限状态机的代数建模与分析

- 带输出的有限状态机
- 受控有限状态机
- 概率有限状态机

例3.4 考虑概率有限自动机 $\mathcal{A}^p = (X, \Sigma, P, x^0)$, 其中, $\sigma_k (k \in [1,3])$ 代表事件 α, β, γ , 设定 $x_i \sim \delta_5^i (i \in [1,5])$ 且 $\sigma_k \sim \delta_3^k (k \in [1,3])$ 。

$$\mathcal{F}_1 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0.5 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.7 \\ 0.5 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

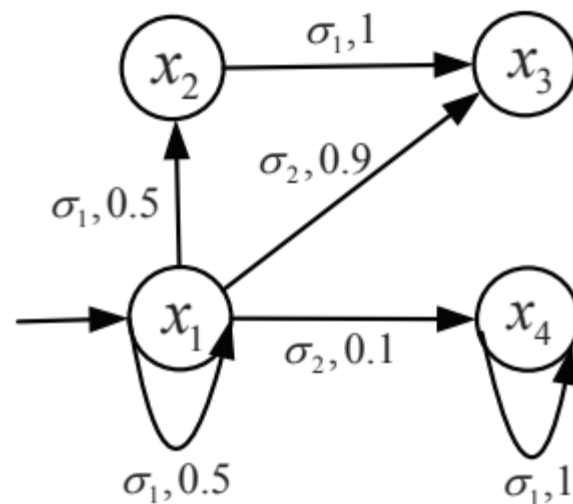


$$\mathcal{F} = [\mathcal{F}_1, \mathcal{F}_2, \mathcal{F}_3] \quad \mathcal{F}_2 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.9 & 0.3 \end{bmatrix} \quad \mathcal{F}_3 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0.1 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

三、有限状态机的代数建模与分析

- 带输出的有限状态机
- 受控有限状态机
- 概率有限状态机

作业：考虑不确定有限自动机 $\mathcal{A}^p = (X, \Sigma, P, x^0)$ ，其状态转移图如下。其中状态集 $X = \{x_1, x_2, x_3, x_4\}$ ，输入符号集 $\Sigma = \{\sigma_1, \sigma_2\}$ ，初始状态集 $x_0 = \{x_1\}$ 。



三、有限状态机的代数建模与分析

- 带输出的有限状态机
- 受控有限状态机
- 概率有限状态机

注意：对 $k \in \{1, 2, \dots, m\}$,

- (1) 列向量 $Col_i(\mathcal{F}_k)$ 元素之和是小于等于一的，且列向量 $Col_i(\sum_{k=1}^m \mathcal{F}_k)$ 的元素之和是等于一的。
- (2) 矩阵 $\sum_{k=1}^m \mathcal{F}_k$ 是一个随机矩阵。矩阵 $\mathcal{F}_k$ 一般情况下是一个子随机矩阵，
- (3) 矩阵 $\mathcal{F}$ 也可以看作是概率有限自动机 $\mathcal{A}^p$ 基于事件 σ 子图的加权邻接矩阵

四、进阶导读

• 有限状态机平衡点

有限自动机的平衡点与闭环状态转移矩阵，这两个概念将在有限自动机代数分析中起着重要作用。首先，我们给出不确定有限自动机平衡点的确切定义如下：

定义4.1： 给定Moore有限自动机，状态 $x \in X$ 称作平衡点，如果存在事件 $e \in \Sigma$ 使得状态转移函数 $f(x, e)$ 表示从状态 x 出发经过一步唯一到达 x 。

注： 在不确定有限自动机中，同一个状态在同一个事件驱动下会产生多个转移状态。根据上述定义，如果状态 x_i 在一个事件驱动下称作平衡状态，当且仅当

$$|f(x_i, e_k)| = 1 \text{ 且 } f(x_i, e_k) = x_i。$$

四、进阶导读

记 $\Gamma: X \rightarrow 2^{\Sigma}$ 表示有限自动机的可行事件函数。在矩阵半张量积的框架下，可行事件函数可以表示成：

$$\hat{u}(t) = Ex(t),$$

其中 $\hat{u}(t) = (\hat{u}_1(t), \hat{u}_2(t), \dots, \hat{u}_m(t))^T$ 表示可行事件对应的向量表示。 $E \in B_{\{m \times n\}}$ 称作有限自动机的可行事件矩阵，可定义如下：

$$E_{(k,i)} = \begin{cases} 1, & |f(x_i, e_k)| > 0 \\ 0, & \text{否则.} \end{cases}$$

• 有限状态机平衡点

四、进阶导读

• 有限状态机平衡点

给定一个有限自动机, 其中 $X \sim \Delta_n$, $\Sigma \sim \Delta_m$ 。则可行事件矩阵与可行事件集可等价表示为:

$$\Gamma(x_i) = \Gamma(\delta_n^i) = \Theta(\text{Col}_i(E)), i \in [1, n],$$

其中, 函数 Θ 的定义为: 对于非零布尔向量 $x \in B_{\{m \times 1\}}$, 则

$$\Theta(x) := \{y \in \Delta_m | y \wedge x = y\}.$$

对于一个给定有限自动机, 如果存在一个事件 $e \in \Sigma$ 使得 $f(x, e) = x$, 则状态 $x \in X$ 称作一个平衡点。对应到确定有限自动机的图中, 平衡点可认为是该状态存在自环。

四、进阶导读

- 有限状态机平衡点
- 闭环状态转移矩阵

给定一个Moore有限状态机的代数方程：

$$\begin{cases} x(t+1) = \hat{F} \ltimes_{\mathcal{B}} u(t) \ltimes_{\mathcal{B}} x(t) \\ y(t) = \hat{H} \ltimes_{\mathcal{B}} x(t) \end{cases} \quad (4.1)$$

针对每个状态设定一个控制输入事件子集

$$S_i = e_{1_i}, e_{2_i}, \dots, e_{\epsilon_i} \subseteq \Gamma(x_i), i \in [1, n],$$

用该子集的事件驱动所对应的状态得到的自动机，称作**闭环有限自动机**。闭环有限自动机所对应的状态转移矩阵称作**闭环转移矩阵**，记作 $A^S \in \mathcal{B}_{n \times n}$ 。

四、进阶导读

- 有限状态机平衡点
- 闭环状态转移矩阵

实际上，给定子集下的闭环状态转移矩阵决定了整个闭环自动机系统的状态演化。闭环状态转移矩阵定义如下：

$$A_{(l,i)}^s = \begin{cases} 1, & x_l \in f(x_i, e_k) \text{ and } e_{k_i} \in S_i \\ 0, & otherwise \end{cases} \quad (4.2)$$

四、进阶导读

- 有限状态机平衡点
- 闭环状态转移矩阵

定理4.1：考虑有限状态机，其代数表示式为(4.1)。给定一个控制输入事件子集 $S_i = e_{1_i}, e_{2_i}, \dots, e_{\epsilon_i} \subseteq \Gamma(x_i), i \in [1, n]$ ，则

$$D^S = \hat{F} \ltimes_{\mathcal{B}} E^S \ltimes_{\mathcal{B}} \Phi_n \quad (4.3)$$

其中， E^S 表示该控制输入子集对应的可行事件矩阵，即可行事件子矩阵，定义如下：

$$E_{(k,i)}^S = \begin{cases} 1, & e_{k_i} \in S_i \\ 0, & otherwise \end{cases}$$

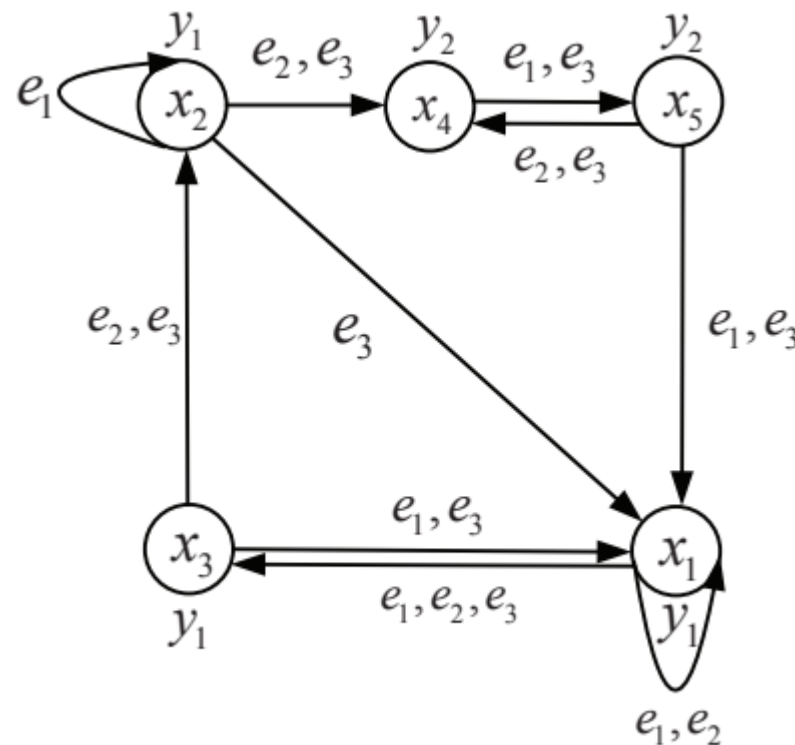
特别地，当 $S_i = \Gamma(x_i)$ ，即 $E^S = E$ 时，则

$$\mathbf{D} = \hat{F} \ltimes_{\mathcal{B}} E \ltimes_{\mathcal{B}} \Phi_n = (\mathcal{B}) \Sigma_{k=1}^m \hat{F}_k.$$

四、进阶导读

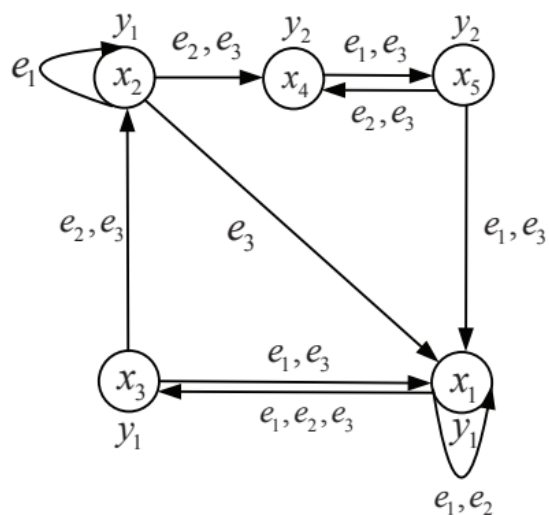
- 有限状态机平衡点
- 闭环状态转移矩阵

例4.1 考虑有限状态机 $\mathcal{A} = (X, Y, \Sigma, f, h, x_0, X_m)$ ，其状态转移如下图， $x_i \sim \delta_5^i (i \in [1,5])$ ， $y_j \sim \delta_2^j (j \in [1,2])$ 和 $e_k \sim \delta_3^k (k \in [1,3])$ 。设定初始状态是任意状态，每个状态的控制输入子集为： $S_1 = \{e_1, e_2\}$ ， $S_2 = \{e_2, e_3\}$ ， $S_3 = \{e_1, e_2\}$ ， $S_4 = \emptyset$ ， $S_5 = \{e_2\}$ 。



四、进阶导读

- 有限状态机平衡点
- 闭环状态转移矩阵



例4.1

$$\hat{F} = \begin{bmatrix} 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}$$

$$E^s = \begin{bmatrix} 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 \end{bmatrix}$$

$$A^s = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\hat{F} \times_B E^s \times_B \Phi_n = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

四、进阶导读

- 有限状态机平衡点
- 闭环状态转移矩阵

注：从上例，我们发现集合矩阵 $\hat{F}$ 的列向量并不全部是逻辑向量，此时，矩阵半张量积公式

$$x \ltimes_B x = \Phi_n \ltimes_B x$$

不成立。因此，式（4.3）的获取并不是把可行事件子矩阵代入状态方程，而是经过构造得到的。

四、进阶导读

- 有限状态机平衡点
- 闭环状态转移矩阵
- 输出反馈镇定

- 研究受控有限状态机**输出反馈稳定性判定问题**。
- 研究受控有限状态机的**综合问题**，即如何设计一个输出反馈监控器使得闭环系统是稳定的。
- 研究如何寻找**逻辑型输出反馈监控器** (Logically Output Feedback Supervisor) 的一个有效算法, 这里的**逻辑型**指的是每个状态只由一个可控事件进行驱动, 并满足系统期望的行为。
- 研究如何寻找**最大输出反馈监控器** (Maximally Output Feedback Supervisor) 的一个有效算法, 这里的**最大**指的是尽可能少的失效、禁止可控事件。该类控制器可以在保证系统镇定前提下最大程度地维持系统的行为。

四、进阶导读

- 有限状态机平衡点
- 闭环状态转移矩阵
- 输出反馈镇定

定理4.2: 考虑受控有限自动机

$$\begin{cases} x(t+1) = \hat{F} \ltimes_{\mathcal{B}} u(t) \ltimes_{\mathcal{B}} x(t) \\ y(t) = \hat{H} \ltimes_{\mathcal{B}} x(t) \end{cases}$$

和该自动机的平衡点 $x_1 = \delta_n^1$, 系统是基于平衡点输出反馈可镇定的, 当且仅当存在**输出反馈控制矩阵** $K \in \mathcal{B}_{m \times p}$ 使

$$(D^s)^{(n)} = \delta_n^1 1_n^T \quad (4.4)$$

其中, $D^s = (\hat{F} \ltimes_{\mathcal{B}} E^s \ltimes_{\mathcal{B}} \Phi_n)$, $E^s = K \ltimes_{\mathcal{B}} H$, Φ_n 是降幂矩阵。

四、进阶导读

- 有限状态机平衡点
- 闭环状态转移矩阵
- 输出反馈镇定

我们定义受控有限状态机的闭环转移矩阵记作 $\tilde{D}$ ，可定义如下：

$$\tilde{D}_{(i,l)} = \begin{cases} 1, & \text{if } D_{(i,l)} = 1 \text{ and } \exists e \in \Sigma_{uc}, s.t. f(x_l, e) = x_i; \\ & \text{or } \mathcal{P}_j^{\nu_k} = 1, x_l \in O(y_j), e_{\nu_k} \in \Sigma_c \cap \pi(y_j), f(x_l, e_{\nu_k}) = x_i. \\ 0, & \text{otherwise} \end{cases}$$

则受控有限状态机的闭环代数表示为：

$$x(t+1) = \tilde{D} \bowtie_{\mathcal{B}} x(t)$$

根据以上定义与结果，我们给出基于集合的输出反馈可镇定的**充分必要条件**。

四、进阶导读

- 有限状态机平衡点
- 闭环状态转移矩阵
- 输出反馈镇定
- 逻辑型输出反馈监控的充要条件

定理4.3: 考虑受控有限自动机

$$\begin{cases} x(t+1) = \hat{F} \bowtie_{\mathcal{B}} u(t) \bowtie_{\mathcal{B}} x(t) \\ y(t) = \hat{H} \bowtie_{\mathcal{B}} x(t) \end{cases}$$

以及不变子集 $\delta_n[N_1, N_2, \dots, N_s]$, 其相对应的事件为 $\delta_m[k_1, k_1, k_2, \dots, k_{s-1}]$ 。系统是基于集合输出反馈可镇定的, 当且仅当存在**输出反馈控制模式** $\mathcal{P}$ 使得

$$\text{Col}(\tilde{\mathcal{D}}^{(n-s+1)}) \subseteq \Delta_Z. \quad (4.5)$$

其中, $\Delta_Z = \delta_n[N_1, N_2, \dots, N_s]$.

四、进阶导读

- 有限状态机平衡点
- 闭环状态转移矩阵
- 输出反馈镇定
- 逻辑型输出反馈监控的充要条件
- 对应控制器算法

步骤 1: 计算集合 $\pi(y_j) = \{\delta_m^{v_1^j}, \delta_m^{v_2^j}, \dots, \delta_m^{v_{M_j}^j}\}$, 检验是否是相容的, 如果是, 则更新集合 $\pi(y_j) := \{\delta_m^{k_i}\}, \forall j \in [1, s-1]$, 进行下一步。

步骤 2: 检验行列序号在 $[N_j, n_j]$ 内同时去掉行列号为 N_j ($j \in [1, s-1]$) 后的子矩阵是否为幂零矩阵, 如果是, 设置 $j = s, \tau_j := 1$, 进行步骤 3。

步骤 3: 情况 1: 如果 $j \leq p$ 且 $\tau_j \leq M_j$, 则检验行列序号在 $[1, \sum_{c=1}^j n_c]$ 内同时去掉行列号为 N_j ($j \in [1, s]$) 后的子矩阵是否为幂零矩阵。如果是, 则设置 $j := j + 1, \tau_j := 1$; 否则, 设置 $\tau_j := \tau_j + 1$, 重复步骤 3。

情况 2: 如果 $j \leq p$ 且 $\tau_j > M_j$, 设置 $j := j - 1$ 且 $\tau_j := \tau_j + 1$, 重复步骤 3。

情况 3: 如果 $j > p$, 则找到输出反馈控制模式 $\mathcal{P}$ 。

四、进阶导读

- 有限状态机平衡点
- 闭环状态转移矩阵
- 输出反馈镇定
- 最大输出反馈监控的充要条件

我们定义新的受控有限状态机闭环转移矩阵记作 $\hat{D}$, 可定义如下:

$$\hat{D}_{(i,l)} = \begin{cases} 1 & , if D_{(i,l)} = 1 \text{ and } \exists e \in \Sigma_{uc}, s.t. f(x_l, e) = x_i. \\ (B)\Sigma_{k=1}^{M_j} \mathcal{P}_j^{v_k}, x_l \in O(y_j), e_{v_k} \in \Sigma_c \cap \pi(y_j), f(x_l, e_{v_k}) = x_i & \\ 0 & , otherwise \end{cases}$$

结合**定理4.3**条件 $Col(\tilde{D}^{(n-s+1)}) \subseteq \Delta_Z$ 和STP 与布尔运算性质, 可得方程:

$$L_1 \alpha_1 = \beta_1$$

根据以上定义与结果, 我们给出基于集合的最大输出反馈监控的**充分必要条件**。

四、进阶导读

- 有限状态机平衡点
- 闭环状态转移矩阵
- 输出反馈镇定
- 最大输出反馈监控的充要条件

定理4.4: 考虑受控有限自动机

$$\begin{cases} x(t+1) = \hat{F} \bowtie_{\mathcal{B}} u(t) \bowtie_{\mathcal{B}} x(t) \\ y(t) = \hat{H} \bowtie_{\mathcal{B}} x(t) \end{cases}$$

以及不变子集 $\delta_n[N_1, N_2, \dots, N_s]$, 其相对的事件为 $\delta_m[k_1, k_1, k_2, \dots, k_{s-1}]$ 。系统是基于集合输出反馈可镇定的, 当且仅当它的输出反馈镇定方程

$$L_1 \alpha_1 = \beta_1 \quad (4.6)$$

有解。

四、进阶导读

- 有限状态机平衡点
- 闭环状态转移矩阵
- 输出反馈镇定
- 最大输出反馈监控的充要条件
- 对应控制器算法

步骤 1: 计算集合 $O(y_j)$ 和集合 $\pi(y_j)$ 。

步骤 2: 构建矩阵 $\hat{D}$, 并且计算矩阵 $\hat{D}^{(n-s+1)}$ 。

步骤 3: 利用矩阵半张量积, 构建矩阵方程 $L_1 \alpha_1 = \beta_1$, 并计算方程组的解。

步骤 4: 如果方程可解, 则输出反馈控制模式为 $\mathcal{P}_i^{k_i}, i \in [1, s-1]$ 和 $(\mathcal{P}_s^{\nu_s}, \mathcal{P}_{s+1}^{\nu_{s+1}}, \dots, \mathcal{P}_p^{\nu_p})$ 。

Zhang Z, Chen Z, Han X, et al. On the static output feedback stabilisation of discrete event dynamic systems based upon the approach of semi-tensor product of matrices[J]. **International Journal of Systems Science**, 2019, 50(8): 1595-1608.

四、进阶导读

- 有限状态机平衡点
- 闭环状态转移矩阵
- 输出反馈镇定
- 概率有限状态机状态反馈控制

- 基于概率有限自动机的代数状态空间表示，研究了概率有限自动机状态**概率为一可达性**问题。
- 通过讨论，研究了状态**概率为一镇定性**的代数判定问题。
- 通过设计基于矩阵的算法，求出使得系统概率为一镇定的**状态反馈控制器**。

四、进阶导读

- 有限状态机平衡点
- 闭环状态转移矩阵
- 输出反馈镇定
- 概率有限状态机状态反馈控制

➤ 可达性

定义4.2: 考虑概率有限自动机 $\mathcal{A}^p = (X, \Sigma, P, x^0)$ 如

$$E[x(t + 1)] = \mathcal{F}u(t)E[x(t)],$$

- (1) 如果存在一个事件串 $s = \sigma_{l_1} \sigma_{l_2} \dots, \sigma_{l_k} \in \Sigma^*$ 使得 $P_{pq}^s = 1$, 则称状态 $x^d = \delta_n^q \in X$ 是从状态 $x^0 = \delta_n^p$ 出发**第 k 步概率为一可达**的。
- (2) 如果存在一个有限事件串集 $\{s_1, s_2, \dots, s_r\} \in \Sigma^*$ 使得 $P_{pq}^{s_c} = 1$, 其中, $c \in \{1, 2, \dots, r\}$, 则称状态 $x^d = \delta_n^q \in X$ 是从状态 $x^0 = \delta_n^p$ 出发**有限步概率为一可达**的。

四、进阶导读

- 有限状态机平衡点
- 闭环状态转移矩阵
- 输出反馈镇定
- 概率有限状态机状态反馈控制

➤ 可达性

定理4.5: 考虑概率有限自动机 $\mathcal{A}^p = (X, \Sigma, P, x^0)$ 如式

$$E[x(t+1)] = \mathcal{F}u(t)E[x(t)],$$

(1) 状态 $x^d = \delta_n^q \in X$ 是从状态 $x^0 = \delta_n^p$ 出发第 k 步概率为一可达的, 当且仅当,

$$\delta_n^q \in \text{Col}(D_p^k). \quad (4.7)$$

其中: $D^k = (\mathcal{F} \times W_{[m,n]})^k = [D_1^k, D_2^k, \dots, D_n^k]$

(2) 状态 $x^d = \delta_n^q \in X$ 是从状态 $x^0 = \delta_n^p$ 出发有限步概率为一可达的, 当且仅当, 存在一个正整数 $r < n$ 使

$$\delta_n^q \in \text{Col}(\mathbb{D}_p^r) \quad (4.8)$$

其中: $\mathbb{D}_i^k$ 表示矩阵 $[D_i^1, D_i^2, \dots, D_i^k]$

四、进阶导读

- 有限状态机平衡点
 - 闭环状态转移矩阵
 - 输出反馈镇定
 - 概率有限状态机状态反馈控制
- 可达性
- 可控性

定义4.3: 考虑概率有限自动机 $\mathcal{A}^p = (X, \Sigma, P, x^0)$ 如式

$$E[x(t + 1)] = \mathcal{F}u(t)E[x(t)],$$

以及给定初始状态 $x^0 = \delta_n^p$ 和标记状态 $x^m = \delta_n^m$ 。

(1) 如果存在一个事件序列 $s = \sigma_{l_1} \sigma_{l_2} \dots, \sigma_{l_k} \in \Sigma^*$ 使得 $P_{pm}^s = 1$ 成立, 则称标记状态 $x^m = \delta_n^m$ 是初始状态 $x^0 = \delta_n^p$ 出发第 k 步概率为一可控的。

(2) 如果对于任意状态 $x_j \in X$ 总存在一个事件序列 $s = \sigma_{l_1} \sigma_{l_2} \dots, \sigma_{l_k} \in \Sigma^*$ 使得 $P_{pj}^s = 1$ 成立, 则称系统是从初始状态 $x^0 = \delta_n^p$ 出发第 k 步概率为一可控的。

四、进阶导读

- 有限状态机平衡点
 - 闭环状态转移矩阵
 - 输出反馈镇定
 - 概率有限状态机状态反馈控制
- 可达性
- 可控性

定理4.6: 考虑概率有限自动机 $\mathcal{A}^p = (X, \Sigma, P, x^0)$ 如式

$$E[x(t+1)] = \mathcal{F}u(t)E[x(t)],$$

以及给定初始状态 $x^0 = \delta_n^p$ 和标记状态 $x^m = \delta_n^m$ 。标记状态 $x^m = \delta_n^m$ 是从初始状态 $x^0 = \delta_n^p$ 出发第 k 步概率为一可控的, 当且仅当,

$$\delta_n^m \in \Theta([\langle D_p^k \rangle]), \quad (4.9)$$

其中, $D^k = (\mathcal{F} \ltimes W_{[m,n]})^k = [D_1^k, D_2^k, \dots, D_n^k]$ 。

四、进阶导读

- 有限状态机平衡点
- 闭环状态转移矩阵
- 输出反馈镇定
- 概率有限状态机状态反馈控制
 - 可达性
 - 可控性
 - 镇定性

定义4.4: 考虑概率有限自动机 $\mathcal{A}^p = (X, \Sigma, P, x^0)$ 如式

$$E[x(t + 1)] = \mathcal{F}u(t)E[x(t)],$$

给定标记状态 $x^m = \delta_n^m$ 。如果对任意状态 $x_i \in X$ ，都能找到一个对应的事件序列 $s_i = \sigma_{l_1} \sigma_{l_2} \dots, \sigma_{l_k} \in \Sigma^*$ 使得 $P_{im}^{s_i} = 1$ ，当 $t \geq k_i$ 时，概率为一地保持在标记状态，则称系统是基于标记状态概率为一可镇定的。

引入一个状态集合，记 $\tilde{\Omega}_k(m)$ ，其定义如下：

$$\tilde{\Omega}_k(m) := \{x^p = \delta_n^p \in \Delta_n \mid \exists s_p \sigma_{l_1} \sigma_{l_2} \dots, \sigma_{l_{k_p}} \in \Sigma^* \text{ s.t. } P_{pm}^{s_p} = 1\} \quad (4.10)$$

四、进阶导读

- 有限状态机平衡点
- 闭环状态转移矩阵
- 输出反馈镇定
- 概率有限状态机状态反馈控制
 - 可达性
 - 可控性
 - 镇定性

定理4.7: 考虑概率有限自动机 $\mathcal{A}^p = (X, \Sigma, P, x^0)$ 如式

$$E[x(t+1)] = \mathcal{F}u(t)E[x(t)],$$

给定标记状态 $x^m = \delta_n^m$, 则

$$\delta_n^m \in \tilde{\Omega}_1(m)$$

$$\tilde{\Omega}_k(m) = \Theta \left(\left(\text{Row}_m(R_k) \right)^T \right), R_k \in \mathcal{M}_{n \times n}.$$

其中, 涉及矩阵定义如下:

$$\text{Col}_i(R_k) = \begin{cases} \delta_n^m, \delta_n^m \in \Theta([\langle D_i^k \rangle]) \\ \delta_n^0, \delta_n^m \notin \Theta([\langle D_i^k \rangle]), \end{cases} \quad (4.11)$$

$$\text{且 } D^k = (\mathcal{F} \times W_{[m,n]})^k = [D_1^k, D_2^k, \dots, D_n^k]$$

四、进阶导读

- 有限状态机平衡点
- 闭环状态转移矩阵
- 输出反馈镇定
- 概率有限状态机状态反馈控制
 - 可达性
 - 可控性
 - 镇定性

定理4.8: 考虑概率有限自动机 $\mathcal{A}^p = (X, \Sigma, P, x^0)$ 如式

$$E[x(t + 1)] = \mathcal{F}u(t)E[x(t)],$$

给定标记状态 $x^m = \delta_n^m$ 。假定存在正整数 $\tau \leq n - 1$ 使得

$$\tilde{\Omega}_\tau(m) = \Delta_n$$

则存在最优反馈逻辑矩阵 E^s 使得闭环系统概率为一镇定到标记状态 $x^m = \delta_n^m$ 。

四、进阶导读

- 有限状态机平衡点
- 闭环状态转移矩阵
- 输出反馈镇定
- 概率有限状态机状态反馈控制
 - 可达性
 - 可控性
 - 镇定性

算法8.1: 给定概率有限自动机 $\mathcal{A}^p = (X, \Sigma, P, x^0)$, 给定标记状态 $x^m = \delta_n^m$ 。首先, 计算集合 $\tilde{\Omega}_k(m)$, 并验证是否存在正整数 $\tau \leq n - 1$ 使得 $\tilde{\Omega}_\tau(m) = \Delta_n$ 成立, 如果成立, 则进行以下步骤:

步骤 1: 将 Δ_n 表示成形式 $\Delta_n = \tilde{\Omega}_1(m) \cup \tilde{\Omega}_2(m) \setminus \tilde{\Omega}_1(m) \cup \dots \cup \tilde{\Omega}_\tau(m) \setminus \tilde{\Omega}_{\tau-1}(m)$ 。

步骤 2: 对于 $i \in \{1, 2, \dots, n\}$, 通过 $\delta_n^i \in \tilde{\Omega}_{k_i}(m) \setminus \tilde{\Omega}_{k_i-1}(m)$ 计算 k_i 。如果 $k_i = 1$, $Col_{\epsilon_i^1}(D_i^1) = \delta_n^m$, 则记 $\lambda_\tau \in \{\epsilon_i^{k_i}\}$; 如果 $k_i \in \{2, 3, \dots, \tau\}$, $Col_{\lambda_{\tau-k_i+1}}(D_i^{k_i}) = \delta_n^m$, 其中, $\lambda_{\tau-k_i+1} = (\epsilon_i^{k_i} - 1)q^{k_i-1} + \lambda_{\tau-k_i+2}$ 。则记 $S_i = \sigma_{\epsilon_i}^{k_i}$ 。

步骤3: 由下式构建最优反馈逻辑矩阵 E^s

$$E_{(k,i)}^s = \begin{cases} 1, & e_{k_i} \in S_i \\ 0, & otherwise \end{cases}$$

四、进阶导读

- 有限状态机平衡点
- 闭环状态转移矩阵
- 输出反馈镇定
- 概率有限状态机状态反馈控制
- 可达性
- 可控性
- 镇定性

算法的复杂度：利用矩阵半张量积，计算集合 $\tilde{\Omega}_k(m)$ 需要计算矩阵 $D^{(k)}$ 。其中，计算矩阵需要

$$n \times nq \times nq^2 + n \times nq \times nq^3 + \cdots n \times nq \times nq^{\tau+1} \\ = n^3 q^3 (q^\tau - 1) / (q - 1)$$

次计算。因此，验证系统概率为一镇定的计算复杂度为

$$O(n^3 q^3 (q^\tau - 1) / (q - 1)),$$

其中， n ， q 和 τ 分别表示 $|X|$ ， $|\Sigma|$ 和定理中存在的最小的正整数。

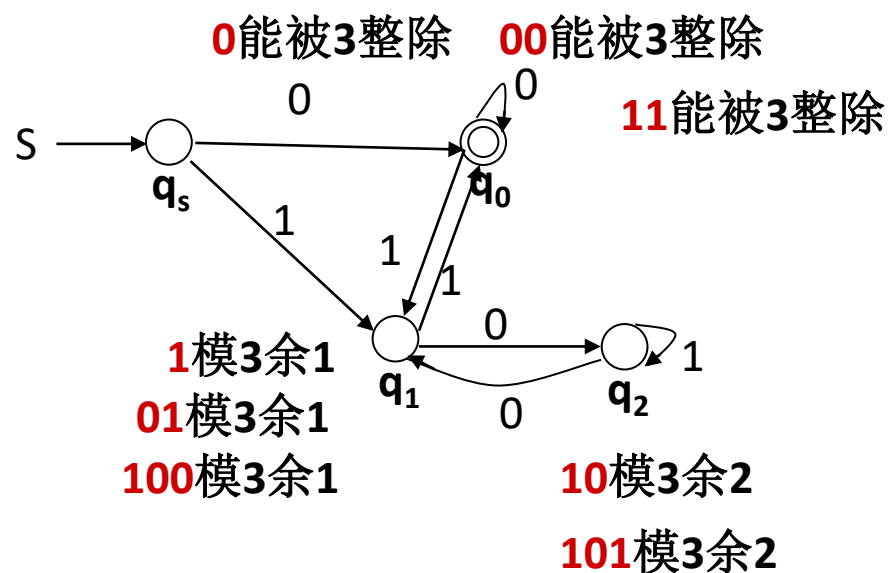
五、习题与思考：习题

1、构造一个DFA，它接受的语言为 $\{x|x \in \{0,1\}^*, \text{且当把}x\text{看成二进制数时, }x\text{模}3\text{与}0\text{同余}\}$ 。

- 分析：换句话说， x 要能被3整除。
- 当二进制数 x 的位数向右不断增加时，它的值（换算成十进制）的增加很有规律： $x0$ 的值等于 $2x$ ， $x1$ 的值等于 $2x+1$ 。
 - 例如 $x=100$ ，它的十进制值是4，右边添上0后为1000，它的十进制值是8；右边添上1后为1001，它的十进制值是9。
 -
 - 如 x 模3余1，则 $2x$ 模3余2，而 $2x+1$ 模3余3，即能被3整除了。
 - 又如有某个 x 模3余2，则 $2x$ 模3余4，而余数4大于3，被3除余1，所以 $2x$ 模3余1；而 $2x+1$ 模3余5，相当于模3余2。
- 经这样分析以后，除初始状态外，只需要设3个状态：模3余0状态（即终结状态），模3余1状态，模3余2状态。

五、习题与思考：习题

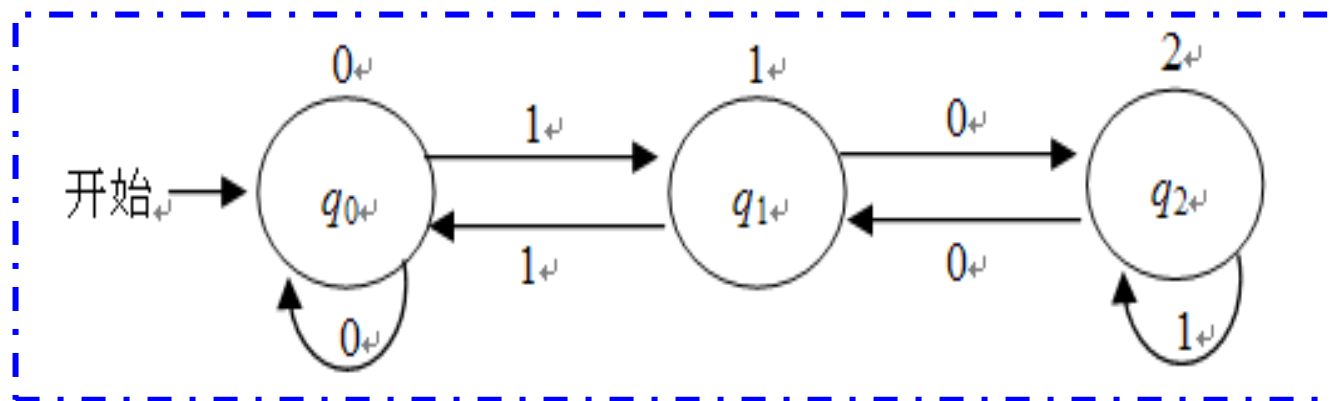
1、构造一个DFA，它接受的语言为 $\{x|x \in \{0,1\}^*, \text{且当把}x\text{看成二进制数时, } x \bmod 3 \text{与} 0 \text{同余}\}$ 。



五、习题与思考：习题

分析：因为模3余数只能有0, 1, 2三个值，因此取 $\Delta=\{0, 1, 2\}$ ，并且只设三个状态 q_0, q_1, q_2 ，分别对应这三种余数。

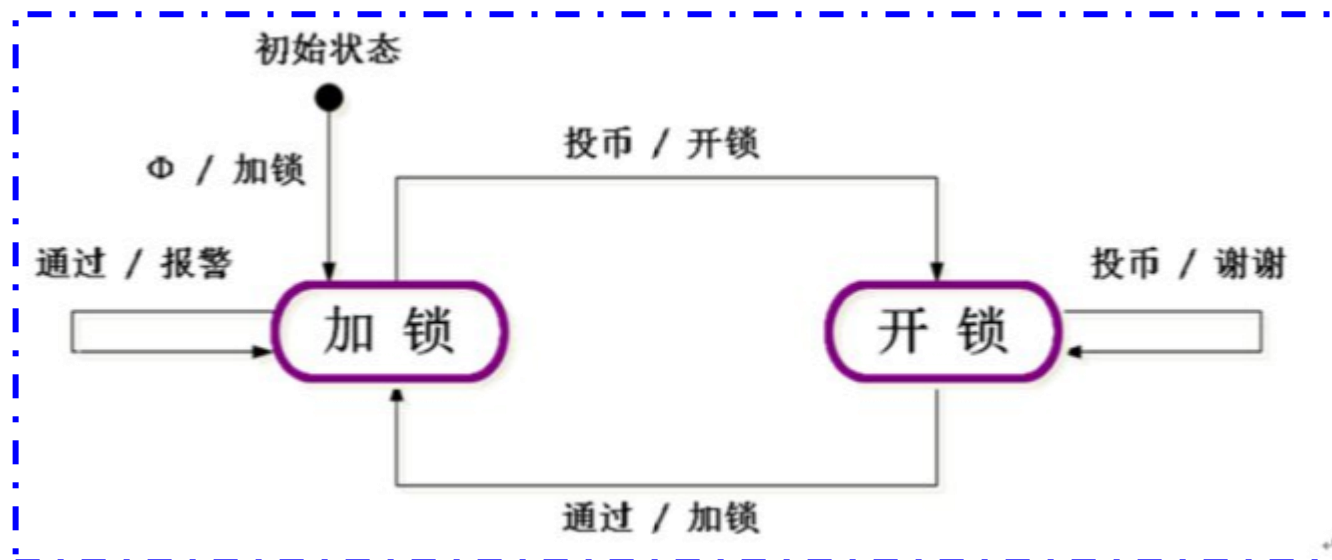
2、设计一个Moore机， $\Sigma=\{0, 1\}$ ，若将输入串看成一个二进制数，要求在读入过程中，能输出它已读过子串的模3余数。



五、习题与思考：习题

要求如下：有一自动门，它可以被锁上，也可以开锁。当门锁上时，某人可以在它的槽中塞进一枚硬币。这样，门就会自动开锁，转变到开锁的状态；人通过后，门就会自动锁上。

3、有限状态机在现实生活中其实随处可见，用实现一个简单的有限状态机-自动门。



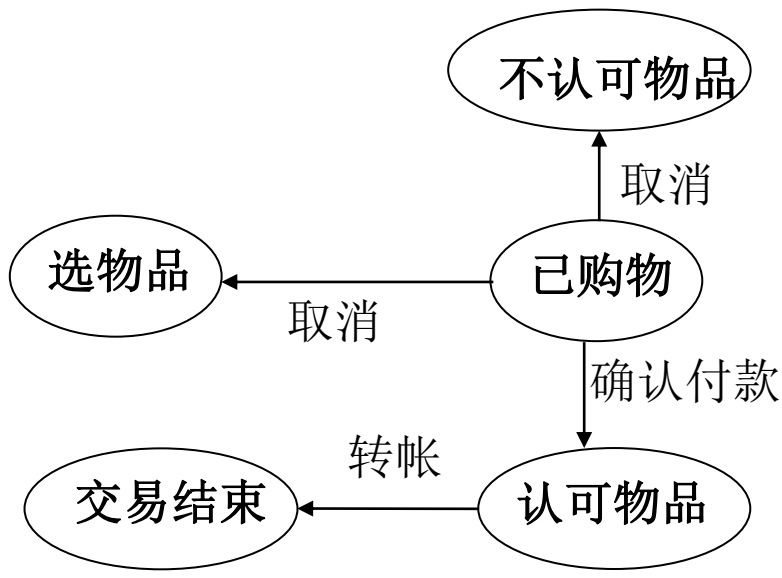
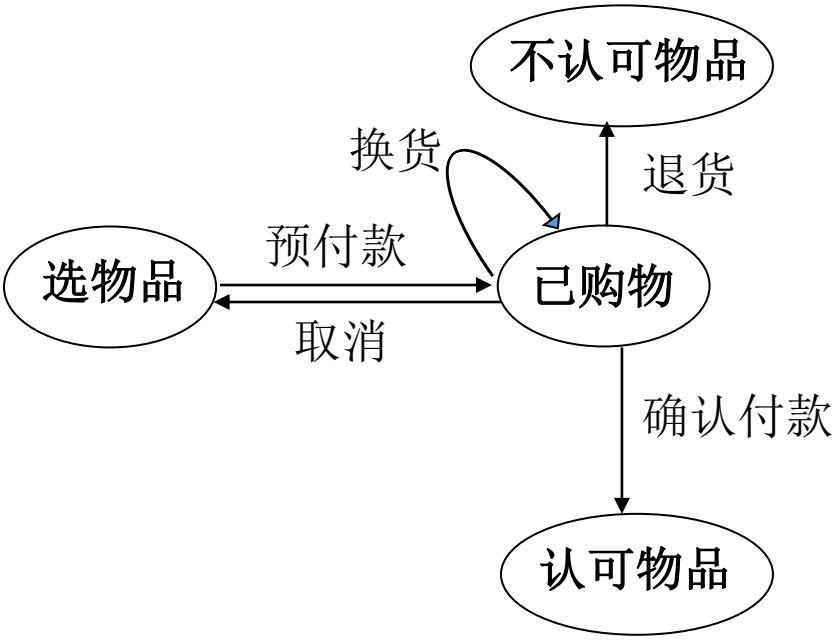
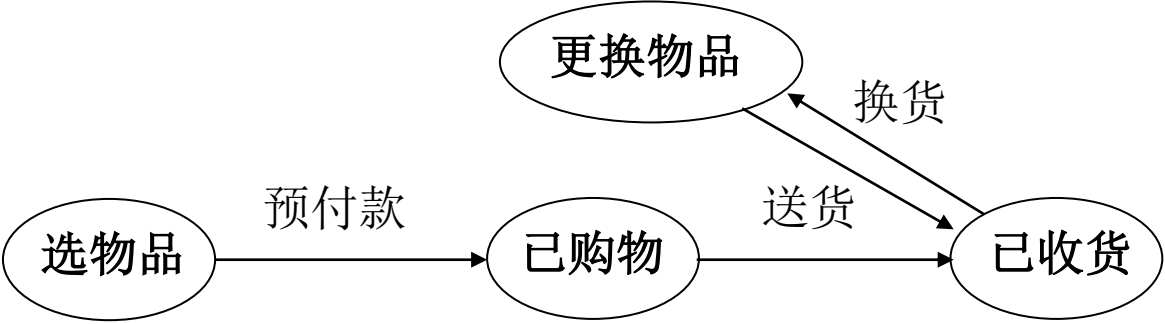
五、习题与思考：习题

4、构造自动机：顾客、店家和支付宝网三方之间的交互限于以下几种事件：

- 1、顾客告诉店家购买某种物品，决定**预付款**购物。并将钱款转入支付宝。
- 2、顾客决定**取消**预付款。顾客通知支付宝，把购物这笔钱保留在自己的银行帐号上。
- 3、店家**送货**给顾客。
- 4、顾客收到货后
 - (1) **确认付款**。通知支付宝将钱款划拨到店家帐号，转到 (5)
 - (2) **退货**。把购物这笔钱保留在自己的银行帐号上，结束。
 - (3) **换货**。寄回店家，店家重**送货**给顾客。
- 5、支付宝将这笔钱**转帐**。把顾客购物这笔钱划拨给店家的帐号。

以上的事件以及事件间在一定条件下转化的情况，可以表示成有限状态系统，每个状态表示某一方所处的局面。

五、习题与思考：习题



五、习题与思考：思考

- ◆ 思考有限值逻辑网络与有限自动机的状态空间表达式的异同，并讨论这些不同因素给有限自动机的分析带来哪些问题。
- ◆ 在半张量积的框架下，如何利用有限自动机对有限值逻辑网络进行建模、分析与控制。
- ◆ 随着自动化生产规模的扩大和自动化程度的提高，以及物联网、大数据、云计算和智能计算等领域的需求不断增加，这些自动化系统在建模、设计、分析和控制过程中面临着复杂性增加、状态指数爆炸问题等巨大挑战。**如何降低计算复杂性、提升计算效率**仍然是亟待解决的问题，目前有大量的研究人员投身于这一领域中。



2023年暑期研修班 感谢聊城大学矩阵半张量积理论与应用研究中心的邀请

衷心感谢!

张志鹏

天津工业大学

2023年 8月15 日