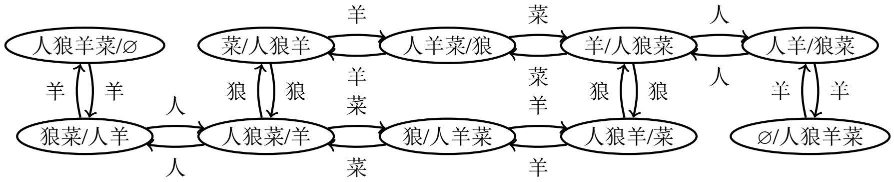
Chapter 2

有穷自动机

2.1 有穷状态系统

有穷状态系统是具有离散输入和输出系统的一种模型. 系统内可以处于任一有穷个内部的格局或称“状态”. 系统的状态概括了关于过去输入的某些信息, 并为确定系统以后的行为所必须. 有穷自动机, 也称为有限状态机, 是有穷状态系统的抽象模型. 有穷自动机是关于存储量极其有限的计算机的很好的模型, 一台计算机用如此小的存储能做什么呢? 回答是: 能做很多有用的事情! 在实际应用中使用最多的两种有穷自动机的变形是: Moore 机和 Mealy 机, 它们的应用, 在我们日常生活中可以说到处都是, 电灯开关, 电梯控制, 自动售货机, 自动取款机等等. 有限状态机的应用领域非常广泛, 比如数字电路的设计, 电脑游戏的 AI 设计, 几乎所有的通讯协议, 比如 TCP, HTTP, 蓝牙, Wifi, 甚至整个电信行业的通讯协议等等. 在计算机应用中也非常多, 比如文本搜索, 词法分析等等. 而我们学习的有穷自动机, 是作为语言的一种识别装置.

例. 狼, 羊, 菜, 人的过河问题. 一个人带着狼、山羊、白菜在一条河的左岸, 有一条船, 大小正好能装下这个人和其它三者之一. 每次只能带一件东西过河, 剩下的两件, 如果没人照顾, 狼会吃羊, 羊会吃菜. 问是否有可能安全过河, 使得羊和白菜都不会被吃掉?

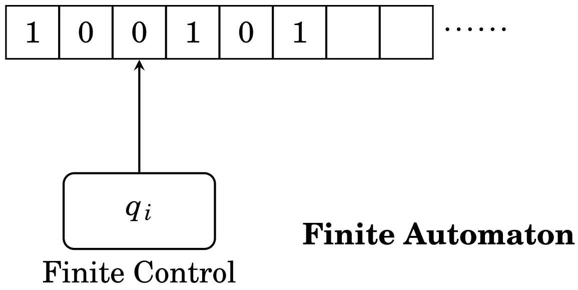


2.2 确定的有穷自动机

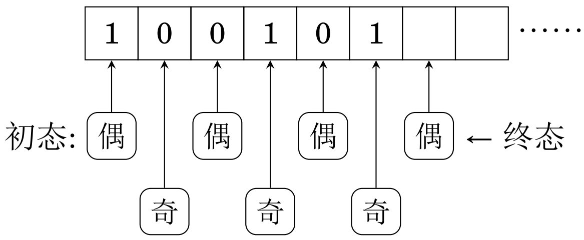
2.2.1 形式定义

确定的有穷自动机

- 一条输入带
- 一个读头
- 一个有穷控制器



例 1. 用有穷自动机识别 $\{ w \in \{0,1\}^* \mid w \text{ 的长度 } |w| \text{ 是偶数.} \}$



有穷自动机可以看做是这样的一个抽象装置, 它具有一条输入带, 一个读头和一个有穷控制器 (*Finite Control*). 输入带上划分为单元格, 每个格子可以放置一个字符, 那么整条带子就可以用来放置一个被扫描的字符串; 因为字符串的长度总是有限的, 在字符串之外的格子上, 总是空白的; 读头可以读入带子上单元格中的字符, 并可以从左向右移动, 每次移动一个单元格; 有穷控制器可以存储有限个状态, 并且可以根据读头读入的字符和当前的状态进行状态改变.

有穷自动机, 在扫描输入的字符串之前, 读头在输入串的第一个字符下面, 然后从左向右一次一个单元格的读入字符, 移动读头, 并修改状态, 然后自动的循环这个过程, 直到扫描完整个字符串之后, 通过有穷控制器中的当前状态, 对这个字符串进行判断, 回答有两种: 接受或拒绝.

我们在需要表示这样几个方面信息: 有穷控制器中的状态, 输入带上的符号, 状态改变的规则, 第一个状态, 哪些状态可以被接受. 将这样的抽象装置再抽象为数学语言, 也就是它的形式定义.

确定的有穷自动机的形式定义

定义. *确定的有穷自动机 (DFA, Deterministic Finite Automaton) A 为五元组*

$$A = (Q, \Sigma, \delta, q_0, F)$$

1. Q : 有穷状态集;
2. Σ : 有穷输入符号集或字母表;
3. $\delta : Q \times \Sigma \to Q$, 状态转移函数;
4. $q_0 \in Q$: 初始状态;
5. $F \subseteq Q$: 终结状态集或接受状态集.

开始时, 输入串在输入带上, 读头在第一个字符, 有穷控制器初始处于 q_0. 自动机的读头每次读入一个字符, 根据转移函数修改当前状态, 并向后移动一个单元格. 若输入串全部读入后, 处于接受状态, 那么自动机接受这个输入串, 否则拒绝该串.

例 2. 请设计 DFA, 在任何由 0 和 1 构成的串中, 接受含有 01 子串的全部串.

1. 未发现 01, 即使 0 都还没出现过;
2. 未发现 01, 但刚刚读入字符是 0;
3. 已经发现了 01.

因此 DFA A 的可定义为:

$$A = (\{q_1, q_2, q_3\}, \{0, 1\}, \delta, q_1, \{q_3\})$$

其中 δ 为:

$$\begin{array}{lll} \delta(q_1, 1) = q_1 & \delta(q_2, 1) = q_3 & \delta(q_3, 1) = q_3 \\ \delta(q_1, 0) = q_2 & \delta(q_2, 0) = q_2 & \delta(q_3, 0) = q_3 \end{array}$$

2.2.2 DFA 的表示

DFA 除了使用其形式定义的五元组表示, 也可以有两种简化的表示方法, 分别为状态转移图 (*transition diagram*) 和状态转移表 (*transition table*).

状态转移图

定义. 状态转移图

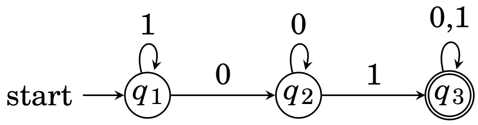
1. 每个状态 q 对应一个节点, 用圆圈表示;

2. 状态转移 $\delta(q,a)=p$ 为一条从 q 到 p 且标记为字符 a 的有向边;

3. 开始状态 q_0 用一个标有 *start* 的箭头表示;

4. 接受状态的节点, 用双圆圈表示.

续例 2. 含有 01 子串的全部串的状态转移图



状态转移表

定义. 状态转移表

1. 每个状态 q 对应一行, 每个字符 a 对应一列;

2. 若有 $\delta(q,a)=p$, 用第 q 行第 a 列中填入的 p 表示;

3. 开始状态 q_0 前, 标记箭头 $\rightarrow$ 表示;

4. 接受状态 $q \in F$ 前, 标记星号 $*$ 表示.

续例 2. 含有 01 子串的全部串的状态转移表

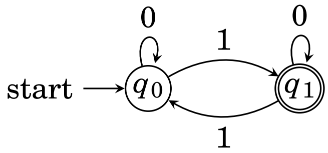
	0	1
$\rightarrow q_1$	q_2	q_1
q_2	q_2	q_3
$*q_3$	q_3	q_3

2.2.3 DFA 的设计举例

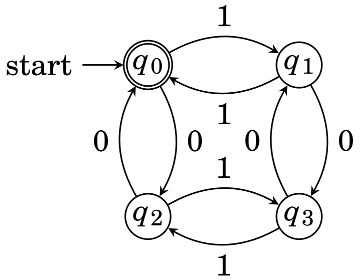
典型问题

设计 DFA 使其接受且仅接受给定的语言 L.

例 3. 若 $\Sigma=\{0,1\}$, 给出接受全部含有奇数个 1 的串 DFA.



例 4. 若 $\Sigma = \{0,1\}$, 给出接受全部含有偶数个 0 和偶数个 1 的串 DFA.



思考题

若 $\Sigma = \{0,1\}$, 那么

1. 如何设计接受 $\varnothing$ 的 DFA?
2. 如何设计接受 Σ^* 的 DFA?
3. 如何设计接受 $\{\varepsilon\}$ 的 DFA?

2.2.4 扩展转移函数

转移函数 δ 是 $Q\times\Sigma$ 上的函数, 所以只能处理 Σ 中的字符, 为了使用方便, 定义字符串上的转移函数 $\hat{\delta}$.

定义. *扩展 δ 到字符串, 定义扩展转移函数 $\hat{\delta}: Q\times\Sigma^*\to Q$ 为*

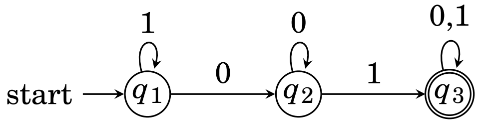
$$\hat{\delta}(q,w)=\begin{cases} q & w=\varepsilon \\ \delta\big(\hat{\delta}(q,x),a\big) & w=xa \end{cases}$$

其中 $a\in\Sigma$, $w,x\in\Sigma^*$.

那么, 当 $w=a_0a_1\cdots a_n$, 则有

$$\begin{aligned}\hat{\delta}(q,w)&=\delta\big(\hat{\delta}(q,a_0a_1\cdots a_{n-1}),a_n\big)\\&=\delta\Big(\delta\big(\hat{\delta}(q,a_0a_1\cdots a_{n-2}),a_{n-1}\big),a_n\Big)=\cdots\\&=\delta\Big(\delta\big(\cdots\delta\big(\hat{\delta}(q,\varepsilon),a_0\big)\cdots,a_{n-1}\big),a_n\Big)\end{aligned}$$

续例 2. 接受全部含有 01 子串的 DFA, $\hat{\delta}$ 处理串 0101 的过程.



$$
\begin{aligned}
\hat{\delta}(q_1,0101) &= \delta(\hat{\delta}(q_1,010),1)\\
&= \delta(\delta(\hat{\delta}(q_1,01),0),1)\\
&= \delta(\delta(\delta(\hat{\delta}(q_1,0),1),0),1)\\
&= \delta(\delta(\delta(\delta(\hat{\delta}(q_1,\varepsilon),0),1),0),1)\\
&= \delta(\delta(\delta(\delta(q_1,0),1),0),1)\\
&= \delta(\delta(\delta(q_2,1),0),1)\\
&= \delta(\delta(q_3,0),1) = \delta(q_3,1) = q_3
\end{aligned}
$$

思考题

从任意状态 q, 对任意的串 w, $\hat{\delta}(q,w)$ 一定会到某个状态吗?

例 5. 对任何状态 q 及字符串 x 和 y, 证明 $\hat{\delta}\big(q,xy\big) = \hat{\delta}\big(\hat{\delta}(q,x),y\big)$.

证明: 对 y 使用归纳法.

1. 当 $y=\varepsilon$ 时

$$
\begin{aligned}
\hat{\delta}\big(\hat{\delta}(q,x),\varepsilon\big) &= \hat{\delta}\big(q,x\big) && \hat{\delta}\text{的定义}\\
&= \hat{\delta}\big(q,x\varepsilon\big)
\end{aligned}
$$

2. 假设 $y=w$ $\big(w\in\Sigma^*\big)$ 时命题成立, 当 $y=wa$ $\big(a\in\Sigma\big)$ 时

$$
\begin{aligned}
\hat{\delta}\big(q,xwa\big) &= \delta\big(\hat{\delta}(q,xw),a\big) && \hat{\delta}\text{和连接的定义}\\
&= \delta\Big(\hat{\delta}\big(\hat{\delta}(q,x),w\big),a\Big) && \text{归纳假设}\\
&= \hat{\delta}\big(\hat{\delta}(q,x),wa\big) && \hat{\delta}\text{的定义}
\end{aligned}
$$

□

课堂练习. Design DFA over $\Sigma=\{0,1\}$ for the language with only one string 000.

2.2.5 DFA 的语言与正则语言

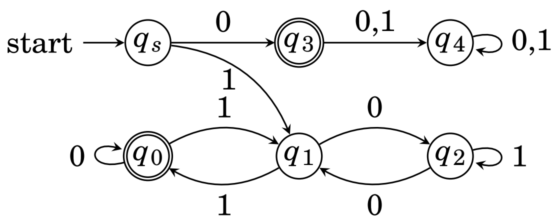
定义. *若 $D=(Q,\Sigma,\delta,q_0,F)$ 是一个 DFA, 则 D 接受的语言为*

$$\mathbf{L}(D)=\left\{ w\in\Sigma^* \;\middle|\; \hat{\delta}(q_0,w)\in F \right\}.$$

定义. *如果语言 L 是某个 DFA D 的语言, 即 $L=\mathbf{L}(D)$, 则称 L 是正则语言.*

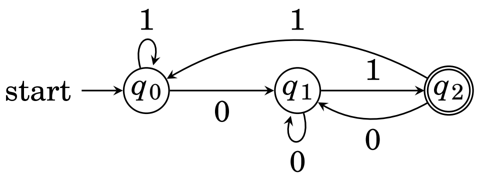
- $\varnothing$, $\{\varepsilon\}$ 都是正则语言
- 若 Σ 是字母表, Σ^*, Σ^n 都是 Σ 上的正则语言

例 6. 设计 DFA 接受 $\{0,1\}$ 上的字符串 w, 且 w 是 3 的倍数的二进制表示.

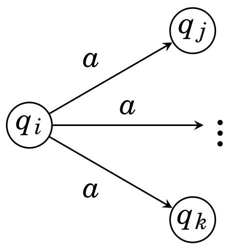


2.3 非确定有穷自动机

例 7. 由 0 和 1 构成的串中, 接受全部以 01 结尾的串, 如何设计 DFA?



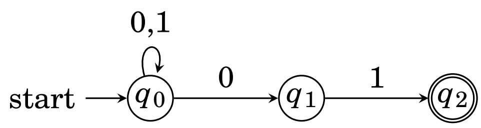
状态的非确定转移



- 同一个状态在相同的输入下, 可以有多个转移状态
- 自动机可以处在多个当前状态

- 使自动机的设计更容易

续例 7. 由 0 和 1 构成的串中, 接受全部以 01 结尾的串.



思考题

有穷自动机有了非确定性, 能否增加它识别语言的能力?

非确定性概念无论在计算理论中起着重要的作用. 在有穷自动机的简单情况下, 透彻的理解这个概念是非常有益的. 对 FA 的模型稍加修改, 使之对同一输入符号, 从一个状态可以有零个、一个或多个的转移. 这种新模型, 称为非确定有穷自动机. 非确定的有穷自动机具有同时处在几个状态的能力, 在处理输入串时, 几个当前状态能"并行的"跳转到下一个状态.

2.3.1 形式定义

非确定有穷自动机的形式定义

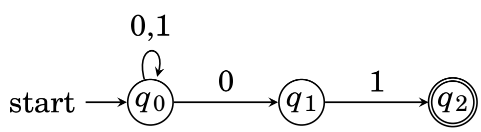
定义. *非确定有穷自动机 (NFA, Nondeterministic Finite Automaton) A 为五元组*

$$A = (Q, \Sigma, \delta, q_0, F)$$

1. Q : 有穷状态集;
2. Σ : 有穷输入符号集或字母表;
3. $\delta : Q \times \Sigma \to 2^Q$ 状态转移函数;
4. $q_0 \in Q$: 为初始状态;
5. $F \subseteq Q$: 为终结状态集或接受状态集.

在形式定义上, NFA 与 DFA 的区别是转移函数和接受方式: NFA 转移函数一般为 $\delta(q,a) = \{p_1, p_2, \cdots, p_n\}$; 当输入串全部读入时, NFA 所处的状态中, 只要包括 F 中的状态, 就称为接受该串.

续例 7. 接受全部以 01 结尾的串的 NFA.



五元组为 $A=(\{q_0,q_1,q_2\},\{0,1\},\delta,q_0,\{q_2\})$, 转移函数 δ:

$$\delta(q_0,0)=\{q_0,q_1\} \qquad \delta(q_1,0)=\varnothing \qquad \delta(q_2,0)=\varnothing$$
$$\delta(q_0,1)=\{q_0\} \qquad \delta(q_1,1)=\{q_2\} \qquad \delta(q_2,1)=\varnothing$$

状态转移表:

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

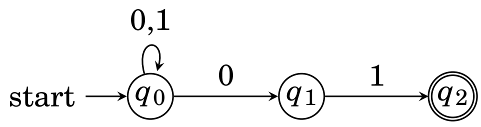
2.3.2 扩展转移函数

定义. *扩展 δ 到字符串, 定义扩展转移函数 $\hat{\delta}:Q\times\Sigma^*\rightarrow 2^Q$ 为*

$$\hat{\delta}(q,w)=\begin{cases}\{q\} & w=\varepsilon\\ \bigcup_{p\in\hat{\delta}(q,x)}\delta(p,a) & w=xa\end{cases}$$

其中 $a\in\Sigma$, $w,x\in\Sigma^*$.

续例 7. 接受 01 结尾的串的 NFA, $\hat{\delta}$ 处理 00101 时每步的状态转移.



1. $\hat{\delta}(q_0,\varepsilon)$=$\{q_0\}$
2. $\hat{\delta}(q_0,0)=\delta(q_0,0)=\{q_0,q_1\}$
3. $\hat{\delta}(q_0,00)=\delta(q_0,0)\cup\delta(q_1,0)=\{q_0,q_1\}\cup\varnothing=\{q_0,q_1\}$
4. $\hat{\delta}(q_0,001)=\delta(q_0,1)\cup\delta(q_1,1)=\{q_0\}\cup\{q_2\}=\{q_0,q_2\}$
5. $\hat{\delta}(q_0,0010)=\delta(q_0,0)\cup\delta(q_2,0)=\{q_0,q_1\}\cup\varnothing=\{q_0,q_1\}$
6. $\hat{\delta}(q_0,00101)=\delta(q_0,1)\cup\delta(q_1,1)=\{q_0\}\cup\{q_2\}=\{q_0,q_2\}$

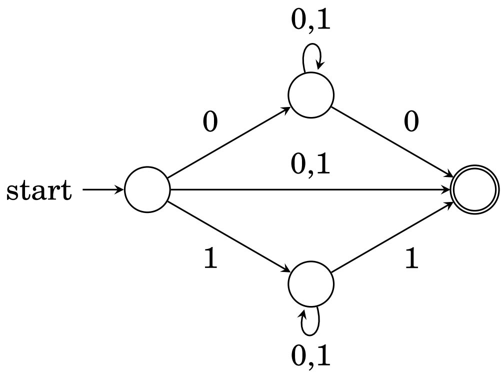
因为 q_2 是接受状态, 所以 NFA 接受 00101.

2.3.3 NFA 的语言

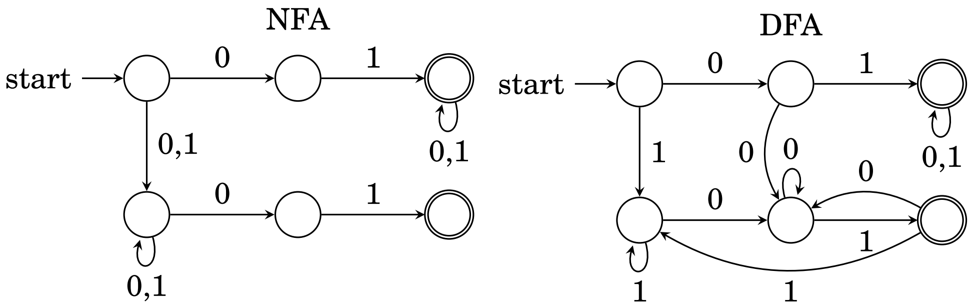
定义. *若 $N = (Q, \Sigma, \delta, q_0, F)$ 是一个 NFA, 则 N 接受的语言为*

$$\mathbf{L}(N) = \left\{ w \in \Sigma^* \,\middle|\, \hat{\delta}(q_0, w) \cap F \neq \varnothing \right\}.$$

例 8. 设计 $L = \left\{ w \in \{0,1\}^* \,\middle|\, w \text{的首尾字符相同} \right\}$ 的 NFA.



例 9. $L = \left\{ w \in \{0,1\}^* \,\middle|\, w \text{ either begins or ends with 01.} \right\}$.



2.3.4 DFA 与 NFA 的等价性

每个 DFA 都是一个 NFA, 显然, NFA 接受的语言包含正则语言. 下面的定理给出, NFA 也仅接受正则语言. 证明的关键表明 DFA 能够模拟 NFA, 即, 对每个 NFA, 能够构造一个等价的 DFA. 使用一个 DFA 模拟一个 NFA 的方法是让 DFA 的状态对应于 NFA 的状态集合.

定理 1. *如果语言 L 被 NFA 接受, 当且仅当 L 被 DFA 接受.*

子集构造法

如果 NFA $N = (Q_N, \Sigma, \delta_N, q_0, F_N)$ 构造 DFA

$$D = (Q_D, \Sigma, \delta_D, \{q_0\}, F_D)$$

1. $Q_D = 2^{Q_N}$;
2. $F_D = \{ S \mid S \subseteq Q_N, S \cap F_N \neq \varnothing \}$;
3. $\forall S \subseteq Q_N$, $\forall a \in \Sigma$:

$$\delta_D(S,a) = \bigcup_{p \in S} \delta_N(p,a).$$

那么有 $\mathbf{L}(D) = \mathbf{L}(N)$.

课堂练习. The set of all strings over $\Sigma = \{0,1\}$ that contain either 00 or 11 as a substring.

证明: 为证明 $\mathbf{L}(D) = \mathbf{L}(N)$, 对 $|w|$ 用归纳法, 往证

$$\hat{\delta}_D(\{q_0\}, w) = \hat{\delta}_N(q_0, w).$$

1. 归纳基础: 当 $w = \varepsilon$ 时, $\hat{\delta}_D(\{q_0\}, \varepsilon) = \{q_0\} = \hat{\delta}_N(q_0, \varepsilon)$;
2. 归纳递推: 假设 $w = x$ $(x \in \Sigma^*)$ 时成立, 当 $w = xa$ $(a \in \Sigma)$ 时

$$\begin{aligned}
\hat{\delta}_N(q_0, xa) &= \bigcup_{p \in \hat{\delta}_N(q_0,x)} \delta_N(p,a) && \text{NFA 的 } \hat{\delta} \text{ 定义} \\
&= \bigcup_{p \in \hat{\delta}_D(\{q_0\},x)} \delta_N(p,a) && \text{归纳假设} \\
&= \delta_D\Big(\hat{\delta}_D(\{q_0\},x), a\Big) && D \text{ 的构造} \\
&= \hat{\delta}_D(\{q_0\}, xa). && \text{DFA 的 } \hat{\delta} \text{ 定义}
\end{aligned}$$

因此上式成立.

因为

$$\hat{\delta}_D(\{q_0\}, w) = \hat{\delta}_N(q_0, w)$$

所以, 对 $\forall w \in \Sigma^*$ 有

$$\begin{aligned}
w \in \mathbf{L}(N) &\iff \hat{\delta}_N(q_0, w) \cap F_N \neq \varnothing && \text{NFA 的语言} \\
&\iff \hat{\delta}_D(\{q_0\}, w) \cap F_N \neq \varnothing && \text{刚证明的} \\
&\iff \hat{\delta}_D(\{q_0\}, w) \in F_D && D \text{ 的构造} \\
&\iff w \in \mathbf{L}(D) && \text{DFA 的语言}
\end{aligned}$$

所以

$$\mathbf{L}(D) = \mathbf{L}(N).$$

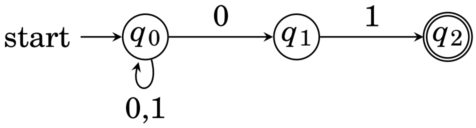
□

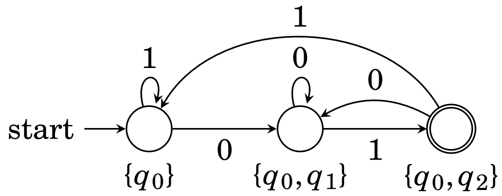
思考题

非确定性没能增加有穷自动机识别语言的能力, 原因是什么呢?

子集构造法: 构造与 NFA 等价的 DFA

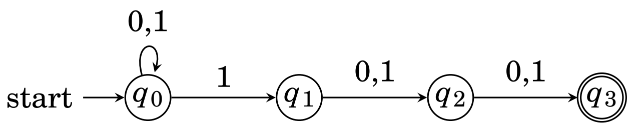
续例 7. 将接受全部以 01 结尾的串的 NFA 转换为 DFA.





	0	1
$\rightarrow q_0$	$\{q_0,q_1\}$	$\{q_0\}$
q_1	$\varnothing$	$\{q_2\}$
$*q_2$	$\varnothing$	$\varnothing$
$\rightarrow \{q_0\}$	$\{q_0,q_1\}$	$\{q_0\}$
$\cancel{\{q_1\}}$	$\varnothing$	$\{q_2\}$
$\cancel{*\{q_2\}}$	$\varnothing$	$\varnothing$
$\{q_0,q_1\}$	$\{q_0,q_1\}$	$\{q_0,q_2\}$
$*\{q_0,q_2\}$	$\{q_0,q_1\}$	$\{q_0\}$
$\cancel{\varnothing}$	$\varnothing$	$\varnothing$
$\cancel{*\{q_1,q_2\}}$	$\varnothing$	$\{q_2\}$
$\cancel{*\{q_0,q_1,q_2\}}$	$\{q_0,q_1\}$	$\{q_0,q_2\}$

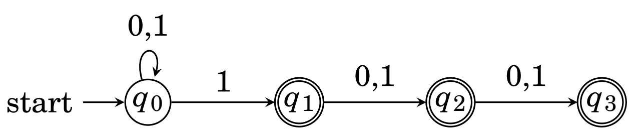
例 10. $L = \{\, w \in \{0,1\}^* \mid w$ 倒数第 3 个字符是 1 $\}$

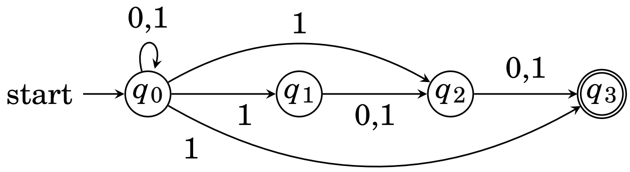


课堂练习. 用子集构造法将其转换为等价的 DFA.

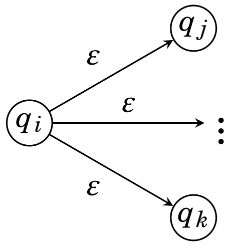
2.4 带有空转移的非确定有穷自动机

例 11. $L = \{\, w \in \{0,1\}^* \mid w$ 倒数 3 个字符至少有一个是 1 $\}$



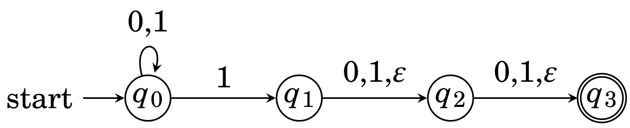


状态的 ε 转移



- 允许状态因空串 ε 而转移, 即不消耗输入字符就发生状态的改变
- 使自动机的设计更容易

续例 11.



2.4.1　形式定义

带空转移非确定有穷自动机的形式定义

定义. *带空转移非确定有穷自动机 (ε-NFA) A 为五元组*

$$A = (Q, \Sigma, \delta, q_0, F)$$

1. Q *: 有穷状态集;*
2. Σ *: 有穷输入符号集或字母表;*
3. $\delta : Q \times (\Sigma \cup \{\varepsilon\}) \to 2^Q$*, 转移函数;*
4. $q_0 \in Q$ *: 初始状态;*
5. $F \subseteq Q$ *: 终结状态集或接受状态集.*

ε-NFA, NFA, DFA 之间的主要区别

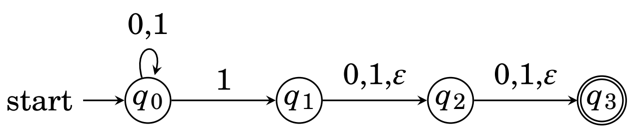
1. 自动机在某状态, 读入某个字符时, 可能有多个转移;
2. 自动机在某状态, 读入某个字符时, 可能没有转移;
3. 自动机在某状态, 可能不读入字符, 就进行转移.

注意

此后, 不再明确区分 ε-NFA 和 NFA, 而认为它们都是 NFA.

续例 11. $L = \{\, w \in \{0,1\}^* \mid w$ 倒数 3 个字符至少有一个是 $1 \,\}$ 的 ε-NFA.

利用 ε 转移设计的有穷自动机:



状态转移表:

	0	1	ε
$\rightarrow q_0$	$\{q_0\}$	$\{q_0, q_1\}$	$\varnothing$
q_1	$\{q_2\}$	$\{q_2\}$	$\{q_2\}$
q_2	$\{q_3\}$	$\{q_3\}$	$\{q_3\}$
$*q_3$	$\varnothing$	$\varnothing$	$\varnothing$

续例 11. $L = \{\, w \in \{0,1\}^* \mid w$ 倒数 3 个字符至少有一个是 $1 \,\}$

当输入字符串是 011 时, ε-NFA 的状态变化.

思考题

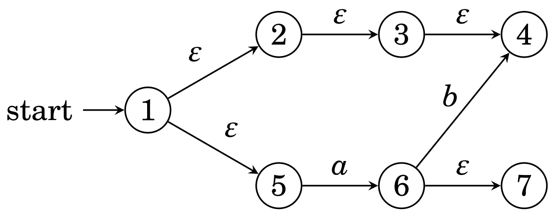
1. 如果初始状态有 ε 转移, 第 1 个字符该如何处理?
2. 如果最后的字符所到的状态有 ε 转移呢?

2.4.2 ε-闭包

状态的 ε-闭包

定义. 状态 q 的 ε-闭包 (*ε-Closure*), 记为 $\textsc{Eclose}(q)$, 表示从 q 经过 ε 序列可达的全部状态集合, 递归定义为:

1. $q \in \textsc{Eclose}(q)$;
2. $\forall p \in \textsc{Eclose}(q)$, 若 $r \in \delta(p,\varepsilon)$, 则 $r \in \textsc{Eclose}(q)$.

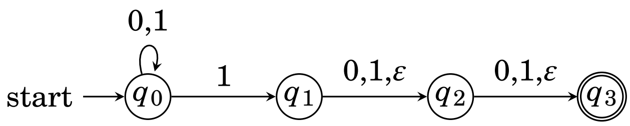


状态集合的 ε-闭包

定义. 状态集 S 的 ε-闭包为

$$\textsc{Eclose}(S) = \bigcup_{q \in S} \textsc{Eclose}(q).$$

续例 11. $L = \{\, w \in \{0,1\}^* \mid w \text{ 倒数 3 个字符至少有一个 } 1 \,\}$



状态转移表及每个状态的闭包:

	0	1	ε	$\textsc{Eclose}(\llcorner\!\lrcorner)$
$\to q_0$	$\{q_0\}$	$\{q_0,q_1\}$	$\varnothing$	$\{q_0\}$
q_1	$\{q_2\}$	$\{q_2\}$	$\{q_2\}$	$\{q_1,q_2,q_3\}$
q_2	$\{q_3\}$	$\{q_3\}$	$\{q_3\}$	$\{q_2,q_3\}$
$*q_3$	$\varnothing$	$\varnothing$	$\varnothing$	$\{q_3\}$

2.4.3 扩展转移函数

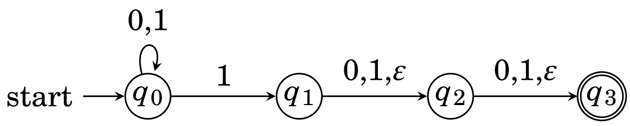
定义. 扩展 δ 到字符串, 定义扩展转移函数 $\hat{\delta}: Q \times \Sigma^* \to 2^Q$ 为

$$\hat{\delta}(q,w) = \begin{cases} \textsc{Eclose}(q) & w = \varepsilon \\ \textsc{Eclose}\Big(\bigcup_{p \in \hat{\delta}(q,x)} \delta(p,a)\Big) & w = xa \end{cases}$$

其中 $a \in \Sigma$, $w, x \in \Sigma^*$.

若设 $\hat{\delta}(q,x)=\{p_1,p_2,\dots,p_k\}$, 则从每个 p_i 经过 a 边到达的所有状态为 $\bigcup_{i=1}^{k}\delta(p_i,a)$; 再设 $\bigcup_{i=1}^{k}\delta(p_i,a)=\{r_1,r_2,\dots,r_m\}$, 则每个 r_j 再求 ε-闭包, 所得到的状态集, 定义为 $\hat{\delta}(q,w)$; 即 $\hat{\delta}(q,w)=\textsc{Eclose}(\{r_1,r_2,\dots,r_m\})$

续例 11. 若 $L=\{\,w\in\{0,1\}^*\mid w$ 倒数 3 个字符至少有一个 1 $\}$ 的 ε-NFA 如下, 求 $\hat{\delta}(q_0,10)$.



$$
\begin{aligned}
\hat{\delta}(q_0,\varepsilon)&=\textsc{Eclose}(q_0)=\{q_0\}\\
\hat{\delta}(q_0,1)&=\textsc{Eclose}\left(\cup_{p\in\hat{\delta}(q_0,\varepsilon)}\delta(p,1)\right)=\textsc{Eclose}\left(\cup_{p\in\{q_0\}}\delta(p,1)\right)\\
&=\textsc{Eclose}(\delta(q_0,1))=\textsc{Eclose}(\{q_0,q_1\})=\{q_0,q_1,q_2,q_3\}\\
\hat{\delta}(q_0,10)&=\textsc{Eclose}\left(\cup_{p\in\hat{\delta}(q_0,1)}\delta(p,0)\right)=\textsc{Eclose}\left(\cup_{p\in\{q_0,q_1,q_2,q_3\}}\delta(p,0)\right)\\
&=\textsc{Eclose}(\delta(q_0,0)\cup\delta(q_1,0)\cup\delta(q_2,0)\cup\delta(q_3,0))\\
&=\textsc{Eclose}(\{q_0,q_2,q_3\})=\{q_0,q_2,q_3\}
\end{aligned}
$$

2.4.4 ε-NFA 的语言

定义. *若 $E=(Q,\Sigma,\delta,q_0,F)$ 是一个 ε-NFA, 则 E 接受的语言为*

$$\mathbf{L}(E)=\left\{\,w\in\Sigma^*\;\middle|\;\hat{\delta}(q_0,w)\cap F\neq\varnothing\,\right\}.$$

2.4.5 ε-NFA 与 DFA 等价性

若有 ε-NFA E, 构造 DFA D, 使 $\mathbf{L}(D)=\mathbf{L}(E)$, 方法与子集构造法类似, 但使用 ε 闭包代替状态转移后的集合, 也称为消除空转移的子集构造法.

消除空转移的子集构造法

构造方法

如果 ε-NFA $E=(Q_E,\Sigma,\delta_E,q_E,F_E)$, 构造 DFA

$$D=(Q_D,\Sigma,\delta_D,q_D,F_D)$$

1. $Q_D=2^{Q_E}$, 或 $Q_D=\{\,S\subseteq Q_E\mid S=\textsc{Eclose}(S)\,\}$;

2. $q_D = \text{Eclose}(q_E)$;

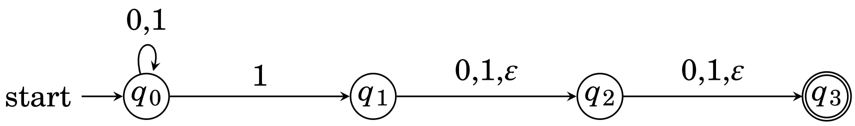
3. $F_D = \{ S \mid S \in Q_D, S \cap F_E \neq \varnothing \}$;

4. $\forall S \in Q_D$, $\forall a \in \Sigma$,

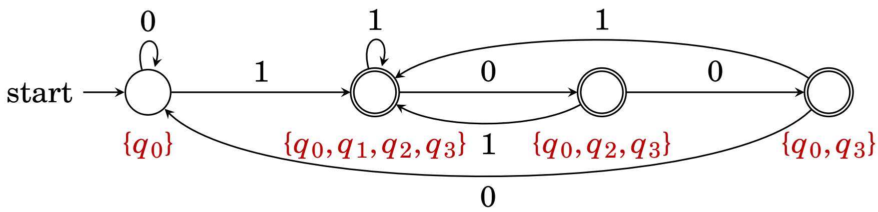
$$\delta_D(S,a) = \text{Eclose}\left(\bigcup_{p\in S} \delta_E(p,a)\right).$$

那么有 $\mathbf{L}(D) = \mathbf{L}(E)$.

续例 11. 将下图 L 的 ε-NFA, 转为等价的 DFA.



	0	1	ε	$\text{Eclose}(\textvisiblespace)$
$\rightarrow q_0$	$\{q_0\}$	$\{q_0,q_1\}$	$\varnothing$	$\{q_0\}$
q_1	$\{q_2\}$	$\{q_2\}$	$\{q_2\}$	$\{q_1,q_2,q_3\}$
q_2	$\{q_3\}$	$\{q_3\}$	$\{q_3\}$	$\{q_2,q_3\}$
$*q_3$	$\varnothing$	$\varnothing$	$\varnothing$	$\{q_3\}$



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

定理 2. *如果语言 L 被 ε-NFA 接受, 当且仅当 L 被 DFA 接受.*

证明: 必要性显然成立, 因为任何 DFA 都是 ε-NFA. 为证明充分性, 对 w 归纳, 往证 $\hat{\delta}_E(q_E,w) = \hat{\delta}_D(q_D,w)$.

1. 当 $w = \varepsilon$ 时

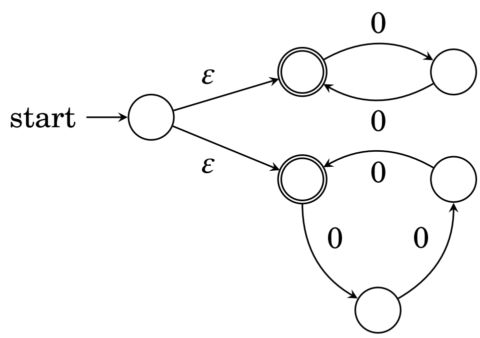
$$\hat{\delta}_E(q_E,\varepsilon) = \text{Eclose}(q_E) = q_D = \hat{\delta}_D(q_D,\varepsilon).$$

2. 当 $w = xa$ 时

$$\hat{\delta}_E(q_E, xa) = \text{ECLOSE}\left(\bigcup_{p \in \hat{\delta}_E(q_E, x)} \delta_E(p, a)\right) = \text{ECLOSE}\left(\bigcup_{p \in \hat{\delta}_D(q_D, x)} \delta_E(p, a)\right)$$
$$= \delta_D\left(\hat{\delta}_D(q_D, x), a\right) = \hat{\delta}_D(q_D, xa)$$

□

例 12. Design ε-NFA for $L = \{ 0^k \mid k \text{ is a multiple of 2 or 3 } \}$.



2.5 练习题

1. Describe deterministic finite-state automata that accept each of the following languages over the alphabet $\Sigma = \{0, 1\}$.

   (a) All strings containing the substring 000.

   (b) All strings not containing the substring 000.

   (c) All strings in which every run of 0s has length at least 3.

   (d) All strings in which every substring 000 appears after every 1.

   (e) The language with only one string 000.

   (f) Every string except 000.

   (g) All strings w such that in *every prefix* of w, the number of 0s and 1s differ by at most 1.

2. [Execise 2.2.2] We defined $\hat{\delta}$ by breaking the input string into any string followed by a single symbol (in the inductive part, Equation 2.1). However, we informally think of $\hat{\delta}$ as describing what happens along a path with a certain string of labels, and if so, then it should not matter how we break the input string in the definition of $\hat{\delta}$. Show that in fact, $\hat{\delta}(q, xy) = \hat{\delta}(\hat{\delta}(q, x), y)$ for any state q and strings x and y. Hint: Perform an induction on $|y|$.

3. [Execise 2.2.3] Show that for any state q, string x, and input symbol a, $\hat{\delta}(q,ax) = \hat{\delta}(\delta(q,a),x)$. Hint: Use Exercise 2.2.2.

4. [Execise 2.2.4] Give DFA's accepting the following languages over the alphabet $\{0,1\}$.

   a) The set of all strings ending in 00. (所有以 00 结尾的串.)

   b) The set of all strings with three consecutive 0's (not necessarily at the end).

   c) The set of strings with 011 as a substring.

5. [Execise 2.2.5] Design DFA.

   a) The set of all strings such that each block of five consecutive symbols contains at least two 0's.

   b) The set of all strings whose tenth symbol from the right end is a 1.

   c) The set of strings that either begin or end (or both) with 01.

   d) The set of strings such that the number of 0's is divisible by 5, and the number of 1's is divisible by 3.

6. [Execise 2.2.6]

   a) The set of all strings beginning with a 1 that, when interpreted as binary integer, is a multiple of 5. for example, strings 101(5), 1010(10), and 1111(15) are in the language; 0, 100(4) and 111(7) are not.

   b) The set of all strings that, when interpreted *in reverse* as a binary integer, is divisible by 5. Examples of string in the language are 0, 10011(25), 1001100(25), and 0101(10).

7. [Execise 2.2.7] Let A be a DFA and q a particular state of A, such that $\delta(q,a) = q$ for all input symbols a. Show by induction on the length of the input that for all input strings w, $\hat{\delta}(q,w) = q$.

8. [Execise 2.2.8] Let A be a DFA and a a particular input symbol of A, such that for all states q of A we have $\delta(q,a) = q$.

   a) Show by induction on n that for all $n \geq 0$, $\hat{\delta}(q,a^n) = q$, where a^n is the string consisting of n a's.

   b) Show that either $\{a\}^* \subseteq L(A)$ or $\{a\}^* \cap L(A) = \emptyset$.

9. [Execise 2.2.9] Let $A = (Q,\Sigma,\delta,q_0,\{q_f\})$ be a DFA, and suppose that for all a in Σ we have $\delta(q_0,a) = \delta(q_f,a)$

   (a) Show that for all $w \neq \varepsilon$ we have $\hat{\delta}(q_0,w) = \hat{\delta}(q_f,w)$.

(b) Show that if x is a nonempty string in $L(A)$, then for all $k > 0$, x^k (i.e. x written k times) is also in $L(A)$.

10. [Execise 2.2.10] Consider the DFA with the following transition table:

    Informally describe the language accepted by this DFA, and prove by induction on the length of an input string that your description is correct. *Hint*: When setting up the inductive hypothesis, it is wise to make a statement about what inputs get you to each state, not just what inputs get you to the accepting state.

	0	1
→ A	A	B
*B	B	A

11. [Execise 2.3.4] Give NFA, try to take advantage of nondeterminism as much as possible.

    (a) The set of strings over alphabet $\{0,1,\cdots,9\}$ such that the final digit has appear before.

    (b) The set of strings over alphabet $\{0,1,\ldots,9\}$ such that the final digit has *not* appeared before.

    (c) The set of strings of 0's and 1's such that there are two 0's separated by a number of positions that is a mutiple of 4. (Note that 0 is an allowable multiple of 4.)

chunyu@hit.edu.cn

http://nclab.net/~chunyu