形式语言与自动机理论

有穷自动机

王春宇
chunyu@hit.edu.cn

计算学部
哈尔滨工业大学

2022 年 2 月

有穷自动机

- 有穷状态系统
- 确定的有穷自动机
- 非确定有穷自动机
- 带有空转移的非确定有穷自动机

有穷状态系统

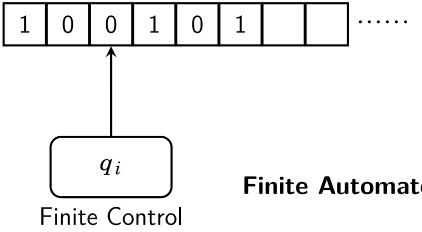
- 有限状态机: Moore Machine, Mealy Machine
- 数字电路设计
- 电脑游戏的 AI 设计
- 各种通讯协议: TCP, HTTP, Bluetooth, Wifi
- 文本搜索, 词法分析

有穷自动机

- 有穷状态系统
- 确定的有穷自动机
  - 形式定义
  - DFA 的设计举例
  - 扩展转移函数
  - DFA 的语言与正则语言
- 非确定有穷自动机
- 带有空转移的非确定有穷自动机

确定的有穷自动机

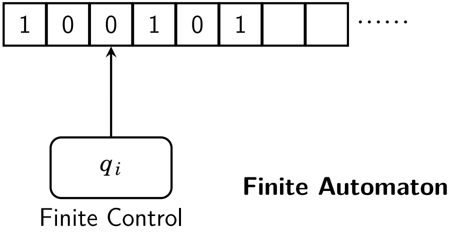
- 一条输入带
- 一个读头
- 一个有穷控制器



Finite Automaton

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 $\{\, w \in \{0,1\}^* \mid w$ 的长度 $|w|$ 是偶数. $\}$

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.

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.

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.

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:

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

状态转移图

定义

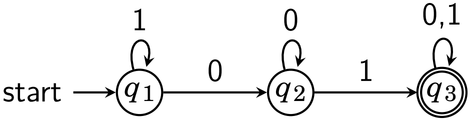
1. 每个状态 q 对应一个节点, 用圆圈表示;
2. 状态转移 $\delta(q,a) = p$ 为一条从 q 到 p 且标记为字符 a 的有向边;
3. 开始状态 q_0 用一个标有 *start* 的箭头表示;
4. 接受状态的节点, 用双圆圈表示.

状态转移图

定义

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

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.

状态转移表

定义

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

典型问题

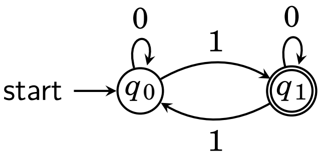
设计 DFA 使其接受且仅接受给定的语言 L.

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

典型问题

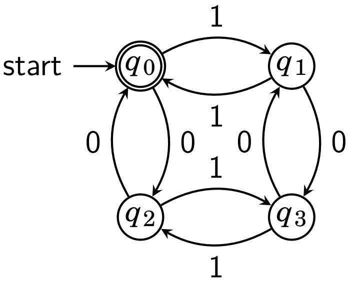
设计 DFA 使其接受且仅接受给定的语言 L.

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.

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?

扩展转移函数

定义

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

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

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

扩展转移函数

定义

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

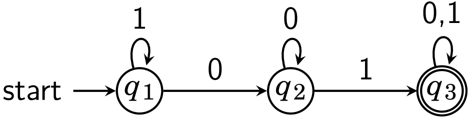
$$\hat{\delta}(q, w) = \begin{cases} q & w = \varepsilon \\ \delta\big(\hat{\delta}(q, x), a\big) & w = xa \end{cases}$$

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

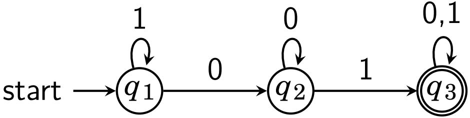
那么，当 $w = a_0 a_1 \cdots a_n$，则有

$$\begin{aligned} \hat{\delta}(q, w) &= \delta\big(\hat{\delta}(q, a_0 a_1 \cdots a_{n-1}), a_n\big) \\ &= \delta\Big(\delta\big(\hat{\delta}(q, a_0 a_1 \cdots a_{n-2}), a_{n-1}\big), a_n\Big) = \cdots \\ &= \delta\Big(\delta\big(\cdots \delta\big(\hat{\delta}(q, \varepsilon), a_0\big) \cdots, a_{n-1}\big), a_n\Big) \end{aligned}$$

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



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 $\hat{\delta}\big(q,xy\big)=\hat{\delta}\big(\hat{\delta}(q,x),y\big)$.

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

证明: 对 y 使用归纳法.

1. 当 $y = \varepsilon$ 时

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

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

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

$\square$

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

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

$$\mathbf{L}(D) = \{\, w \in \Sigma^* \mid \hat{\delta}(q_0, w) \in F \,\}.$$

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

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

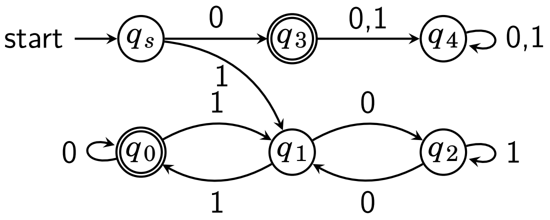
定义

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.

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.

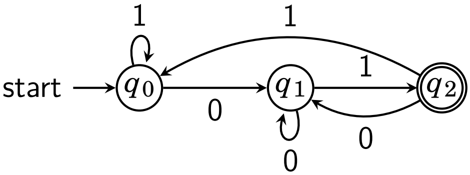


有穷自动机

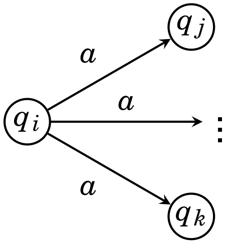
- 有穷状态系统
- 确定的有穷自动机
- 非确定有穷自动机
  - 形式定义
  - 扩展转移函数
  - NFA 的语言
  - DFA 与 NFA 的等价性
- 带有空转移的非确定有穷自动机

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?

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?

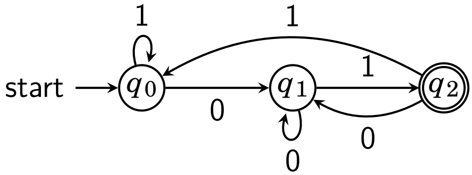


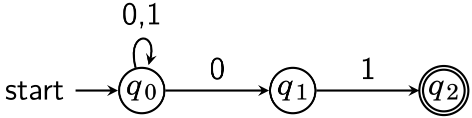
状态的非确定转移



- 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

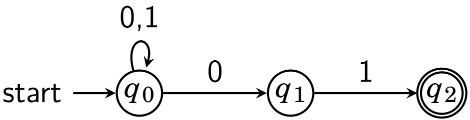
定义

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.

续例 7. 接受全部以 01 结尾的串的 NFA.

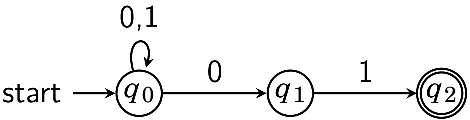


五元组为 $A = \big(\{q_0, q_1, q_2\}, \{0,1\}, \delta, q_0, \{q_2\}\big)$, 转移函数 δ:

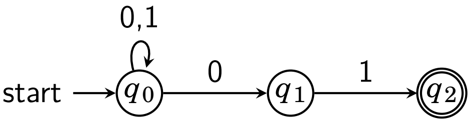
$$\delta(q_0, 0) = \{q_0, q_1\} \qquad \delta(q_1, 0) = \varnothing \qquad \delta(q_2, 0) = \varnothing$$

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

续例 7. 接受全部以 01 结尾的串的 NFA. 识别字符串 00101 的过程.



续例 7. 接受全部以 01 结尾的串的 NFA.



状态转移表:

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

扩展转移函数

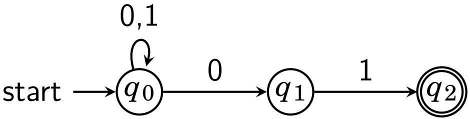
定义

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

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

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.

NFA 的语言

回顾

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

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

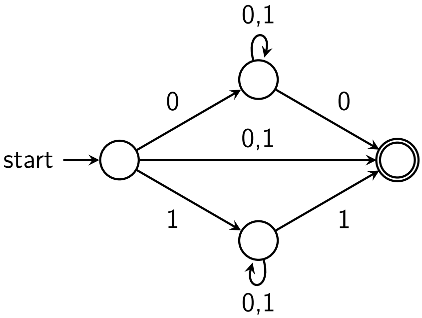
定义

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

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

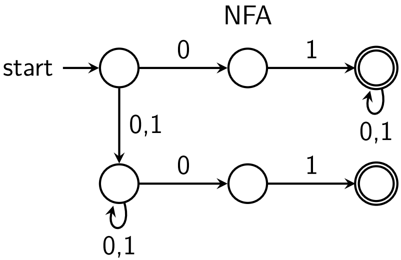
例 8. 设计 $L = \left\{ w \in \{0,1\}^* \,\middle|\, w \text{ 的首尾字符相同} \right\}$ 的 NFA.

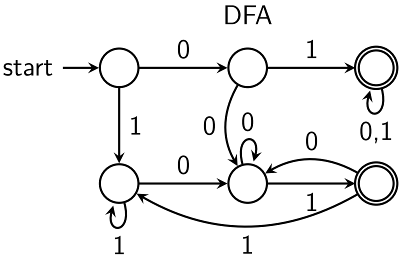
例 8. 设计 $L = \left\{ w \in \{0,1\}^* \,\middle|\, w \text{的首尾字符相同} \right\}$ 的 NFA.



例 9. $L = \{\, w \in \{0,1\}^* \mid w$ either begins or ends with $01.\,\}$.

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





DFA 与 NFA 的等价性

定理 1

如果语言 L 被 *NFA* 接受, 当且仅当 L 被 *DFA* 接受.

DFA 与 NFA 的等价性

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. $F_D = \{ S \mid S \subseteq Q_N, S \cap F_N \neq \emptyset \}$;
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

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

因此上式成立.

因为

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

所以, 对 $\forall w \in \Sigma^*$ 有

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

所以

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

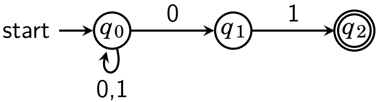
□

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

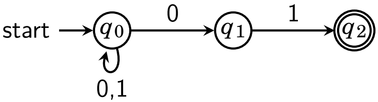
续例 7. 将接受全部以 01 结尾的串的 NFA 转换为 DFA.

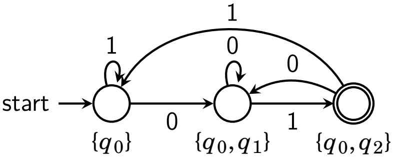


	0	1
$\rightarrow q_0$	$\{q_0, q_1\}$	$\{q_0\}$
q_1	$\varnothing$	$\{q_2\}$
$*q_2$	$\varnothing$	$\varnothing$

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.

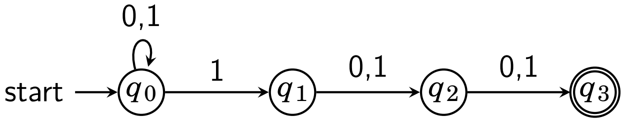




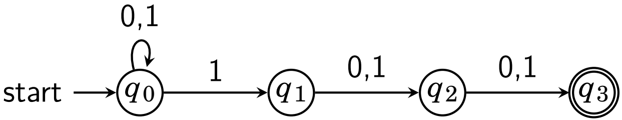
	0	1
$\rightarrow q_0$	$\{q_0, q_1\}$	$\{q_0\}$
q_1	$\varnothing$	$\{q_2\}$
$*q_2$	$\varnothing$	$\varnothing$
$\rightarrow \{q_0\}$	$\{q_0, q_1\}$	$\{q_0\}$
$\cancel{\{q_1\}}$	$\varnothing$	$\{q_2\}$
$\cancel{*\{q_2\}}$	$\varnothing$	$\varnothing$
$\{q_0, q_1\}$	$\{q_0, q_1\}$	$\{q_0, q_2\}$
$*\{q_0,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

例 10. $L = \left\{ w \in \{0,1\}^* \,\middle|\, w \text{ 倒数第 3 个字符是 1} \right\}$

例 10. $L = \left\{ w \in \{0,1\}^* \,\middle|\, w \text{ 倒数第 3 个字符是 } 1 \right\}$



例 10. $L = \left\{ w \in \{0,1\}^* \,\middle|\, w \text{ 倒数第 3 个字符是 1} \right\}$



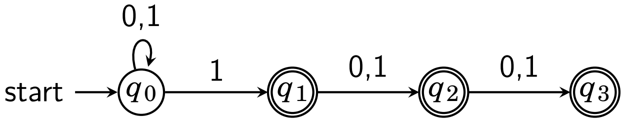
课堂练习. 用子集构造法将其转换为等价的 DFA.

有穷自动机

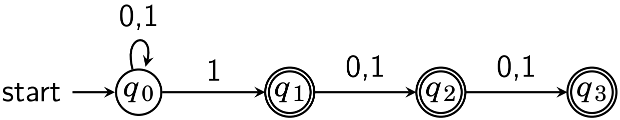
- 有穷状态系统
- 确定的有穷自动机
- 非确定有穷自动机
- 带有空转移的非确定有穷自动机
  - 形式定义
  - ε-闭包
  - 扩展转移函数
  - ε-NFA 的语言
  - ε-NFA 与 DFA 等价性

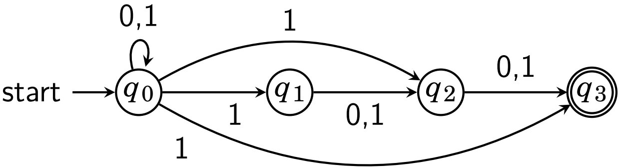
例 11. $L = \left\{ w \in \{0,1\}^* \,\middle|\, w \text{ 倒数 3 个字符至少有一个是 1} \right\}$

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

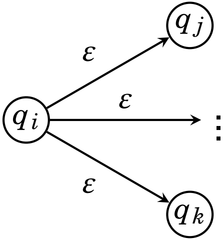


例 11. $L = \left\{ w \in \{0,1\}^* \,\middle|\, w \text{ 倒数 3 个字符至少有一个是 1} \right\}$



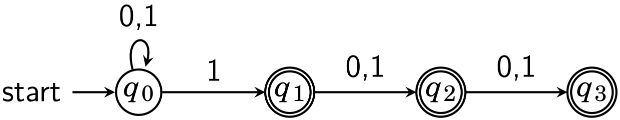


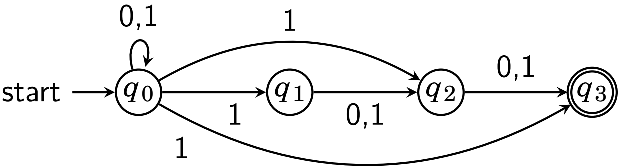
状态的 ε 转移

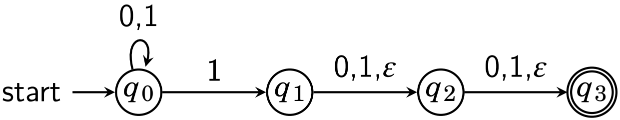


- 允许状态因空串 ε 而转移, 即不消耗输入字符就发生状态的改变
- 使自动机的设计更容易

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







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;

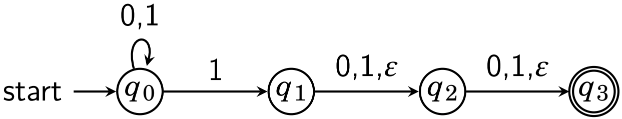
3. 自动机在某状态, 可能不读入字符, 就进行转移.

注意

此后, 不再明确区分 ε-NFA 和 NFA, 而认为它们都是 NFA.

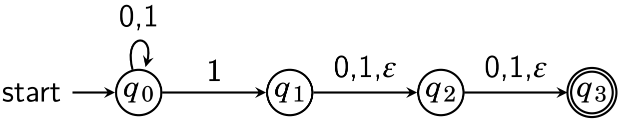
续例 11. $L = \left\{ w \in \{0,1\}^* \,\middle|\, w \text{ 倒数 3 个字符至少有一个是 1} \right\}$ 的 ε-NFA.

利用 ε 转移设计的有穷自动机:



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:

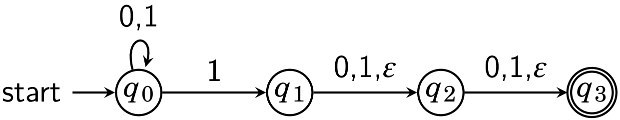


状态转移表:

	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 $\}$

利用 ε 转移设计的有穷自动机:



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?

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:

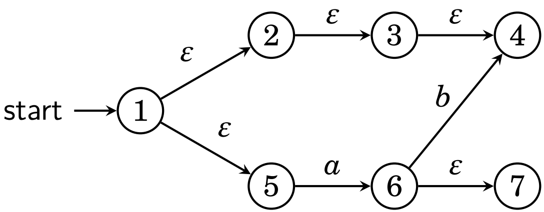
1. $q \in \textsc{Eclose}(q)$;
2. $\forall p \in \textsc{Eclose}(q)$, 若 $r \in \delta(p, \varepsilon)$, 则 $r \in \textsc{Eclose}(q)$.

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 $r \in \textsc{Eclose}(q)$.



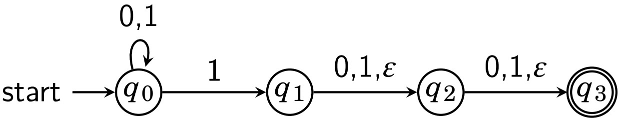
状态集合的 ε-闭包

定义

状态集 S 的 ε-闭包为

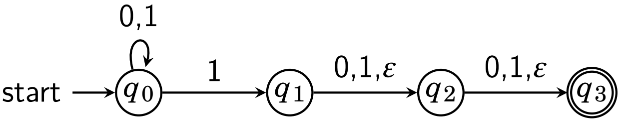
$$\textsc{Eclose}(S) = \bigcup_{q \in S} \textsc{Eclose}(q).$$

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



状态转移表及每个状态的闭包:

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	$\text{ECLOSE}(\sqcup)$
$\rightarrow q_0$	$\{q_0\}$	$\{q_0, q_1\}$	$\varnothing$	$\{q_0\}$
q_1	$\{q_2\}$	$\{q_2\}$	$\{q_2\}$	$\{q_1, q_2, q_3\}$
q_2	$\{q_3\}$	$\{q_3\}$	$\{q_3\}$	$\{q_2, q_3\}$
$*q_3$	$\varnothing$	$\varnothing$	$\varnothing$	$\{q_3\}$

扩展转移函数

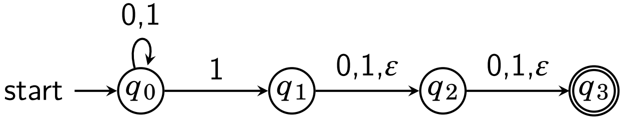
定义

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

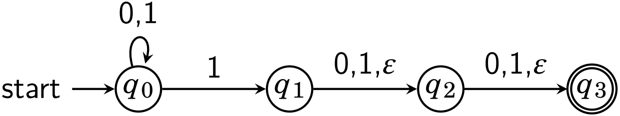
$$\hat{\delta}(q, w) = \begin{cases} \textsc{Eclose}(q) & w = \varepsilon \\ \textsc{Eclose}\left(\bigcup_{p \in \hat{\delta}(q,x)} \delta(p, a)\right) & w = xa \end{cases}$$

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

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 $\hat{\delta}(q_0, 10)$.



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 $\hat{\delta}(q_0, 10)$.



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

ε-NFA 的语言

回顾

DFA $D = (Q,\Sigma,\delta,q_0,F)$ 和 NFA $N = (Q,\Sigma,\delta,q_0,F)$ 的语言分别为

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

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

定义

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

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

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

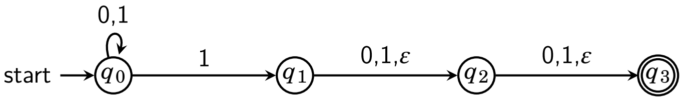
$$D = (Q_D, \Sigma, \delta_D, q_D, F_D)$$

1. $Q_D = 2^{Q_E}$, 或 $Q_D = \{ S \subseteq Q_E \mid S = \textsc{Eclose}(S) \}$;
2. $q_D = \textsc{Eclose}(q_E)$;
3. $F_D = \{ S \mid S \in Q_D, S \cap F_E \neq \emptyset \}$;
4. $\forall S \in Q_D$, $\forall a \in \Sigma$,

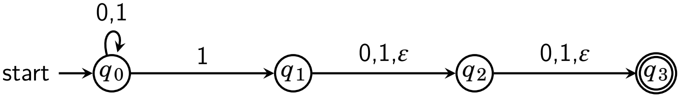
$$\delta_D(S, a) = \textsc{Eclose}\left(\bigcup_{p \in S} \delta_E(p, a) \right).$$

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.

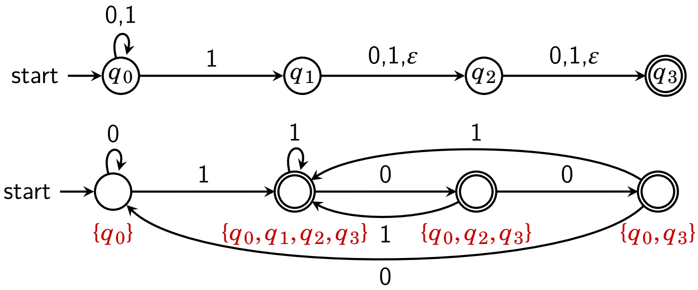


续例 11. 将下图 L 的 ε-NFA, 转为等价的 DFA.



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

续例 11. 将下图 L 的 ε-NFA, 转为等价的 DFA.



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

ε-NFA 与 DFA 等价性

定理 2

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

ε-NFA 与 DFA 等价性

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

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

2. 当 $w = xa$ 时

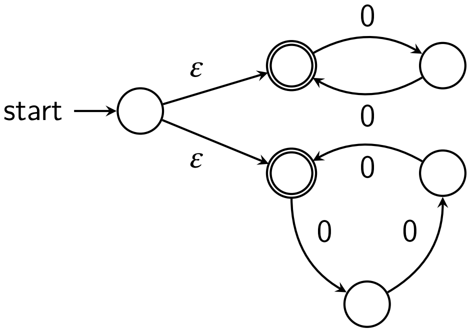
$$\hat{\delta}_E(q_E, xa) = \textsc{Eclose}\left(\bigcup_{p \in \hat{\delta}_E(q_E, x)} \delta_E(p, a)\right) = \textsc{Eclose}\left(\bigcup_{p \in \hat{\delta}_D(q_D, x)} \delta_E(p, a)\right)$$

$$= \delta_D\left(\hat{\delta}_D(q_D, x), a\right) = \hat{\delta}_D(q_D, xa)$$

□

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

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



哈爾濱工業大學
HARBIN INSTITUTE OF TECHNOLOGY

chunyu@hit.edu.cn
http://iilab.net/chunyu