有限状态自动机

乔明达

2014年2月8日

开场白

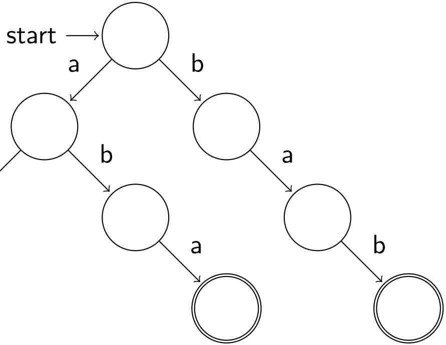
Michael Sipser《计算理论导引》

可计算性(computability)与计算复杂性(computational complexity)

例：Trie

判断一个串t是否在给定的串的集合S中

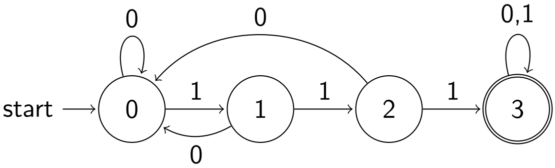
例：$S = \{aa, aba, bab\}$



例：KMP

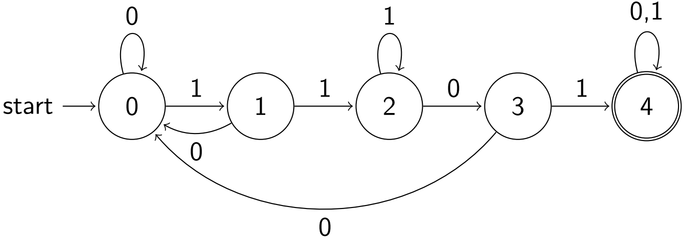
判断一个串t是否包含子串s

例：$s = 111$



例：KMP

例：$s = 1101$



有限状态自动机

Finite Automaton/ˈfaɪˌnaɪt ɔːˈtɒmətən/

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

状态集合Q

字母表Σ

转移函数$\delta : Q \times \Sigma \to Q$

开始状态$q_0 \in Q$

接受状态集合$F \subseteq Q$

有限状态自动机

$s = w_1 w_2 \dots w_n, w_i \in \Sigma$

$r_0 = q_0, r_i = \delta(r_{i-1}, w_i)$

若$r_n \in F$，则有限状态自动机接受串s，否则拒绝串s

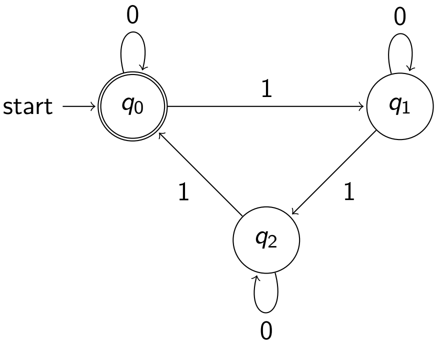
有限状态自动机

字母表$\Sigma = \{0,1\}$，$p \in N^+$

接受包含1的个数是p的倍数的串

例如对于$p = 3$，应该接受串$111, 001011, 000$，拒绝串$10, 1001$

有限状态自动机



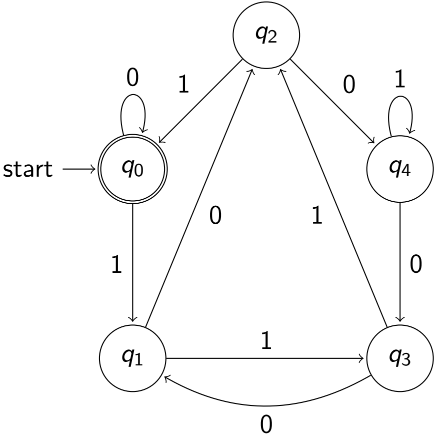
有限状态自动机

字母表$\Sigma = \{0, 1\}$，$p \in N^+$

接受p的倍数的二进制表示（允许前导零）

例如对于$p = 5$，应该接受串$101, 01010, 000$，拒绝串$11, 0001$

有限状态自动机



串与语言

Σ^*为字母表Σ上串的集合

例：当$\Sigma = \{0, 1\}$时，$\Sigma^* = \{\varepsilon, 0, 1, 00, 01, 10, 11, \ldots\}$

Σ^*的任意子集称为字母表Σ上的语言

有限状态自动机M接受的串的集合称为M的语言，记作$L(M)$

可以被确定有限状态自动机识别的语言称为正则语言(regular language)

串的运算

$$xy$$

$$x^k = \underbrace{xx\cdots x}_{k}$$

$$x^0 = \varepsilon$$

语言的运算

交与并

补集：$\overline{A} = \{x|x \in \Sigma^*, x \notin A\}$

连接：$A \circ B = \{xy|x \in A, y \in B\}$

幂：$A^k = \{x_1x_2 \dots x_k|x_1, x_2, \dots, x_k \in A\}$

闭包：$A^* = A^0 \cup A^1 \cup A^2 \cup \cdots$

语言的运算

设有语言$A = \{\varepsilon\}, B = \emptyset$

求：$A \circ A, A \circ B, B \circ B, A^*, B^*$

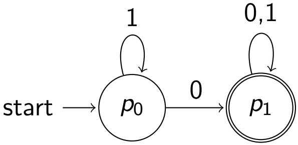
正则语言的封闭性

并集、交集：若A和B是正则语言，则$A \cup B$和$A \cap B$是正则语言

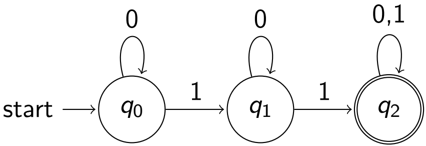
正则语言的封闭性

$\Sigma = \{0, 1\}$

$A = \{x|x$包含至少1个0$\}$

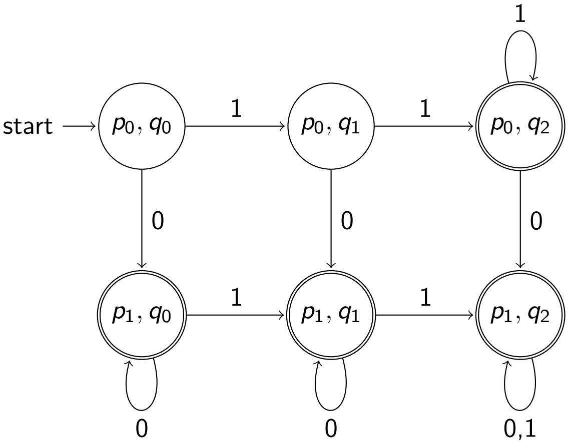


$B = \{x|x$包含至少2个1$\}$



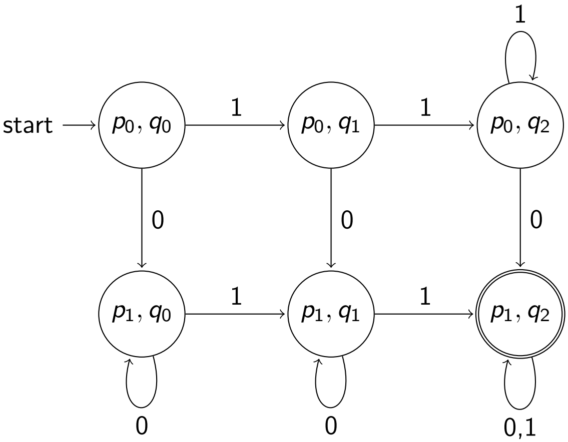
正则语言的封闭性

$A \cup B$



正则语言的封闭性

$A \cap B$



正则语言的封闭性

$M_A = (Q_A, \Sigma, \delta_A, q_A, F_A)$识别语言$A$

$M_B = (Q_B, \Sigma, \delta_B, q_B, F_B)$识别语言$B$

令$M = (Q, \Sigma, \delta, q_0, F)$

其中$Q = Q_A \times Q_B, q_0 = (q_A, q_B)$

$\delta((q_1, q_2), c) = (\delta_A(q_1, c), \delta_B(q_2, c))$

对于$A \cup B$，令$F = (F_A \times Q_B) \cup (Q_A \times F_B)$

对于$A \cap B$，令$F = F_A \times F_B$

正则语言的封闭性

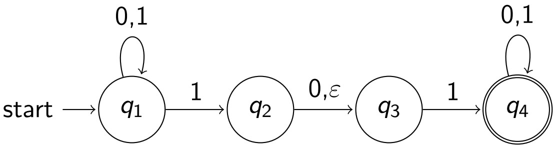
补集：若A是正则语言，则$\overline{A}$是正则语言

正则语言的封闭性

连接：若A和B是正则语言，则$A \circ B$是正则语言

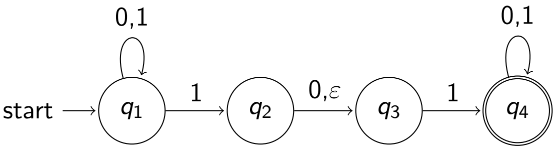
非确定有限状态自动机

DFA(Deterministic Finite Automaton)与NFA(Nondeterministic Finite Automaton)



接受串010110

非确定有限状态自动机



开始：q_1

读入0：q_1

读入1：q_1, q_2, q_3

读入0：q_1, q_3

读入1：q_1, q_2, q_3, q_4

读入1：q_1, q_2, q_3, q_4

读入0：q_1, q_3, q_4

非确定有限状态自动机

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

令$\Sigma_\varepsilon = \Sigma \cup \{\varepsilon\}$

状态集合Q

字母表Σ

转移函数$\delta : Q \times \Sigma_\varepsilon \to \mathcal{P}(Q)$

开始状态$q_0 \in Q$

接受状态集合$F \subseteq Q$

非确定有限状态自动机

若存在$w_1, w_2, \ldots, w_n, r_0, r_1, \ldots, r_n$使得:

$s = w_1 w_2 \ldots w_n, w_i \in \Sigma \cup \{\varepsilon\}$

$r_0 = q_0, r_i \in \delta(r_{i-1}, w_i), r_n \in F$

则NFA接受串s，否则NFA拒绝串s

例：连接

连接：若A和B是正则语言，则$A \circ B$是正则语言

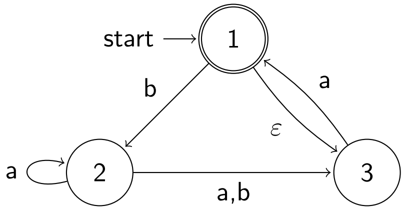
设DFA M_A和M_B分别识别语言A和语言B

对于M_A中的每个接受状态，增加到M_B开始状态的ε转移

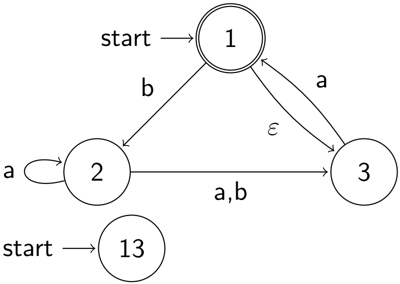
NFA与DFA的等价性

证明：对于每个NFA，存在一个DFA与它识别相同的语言

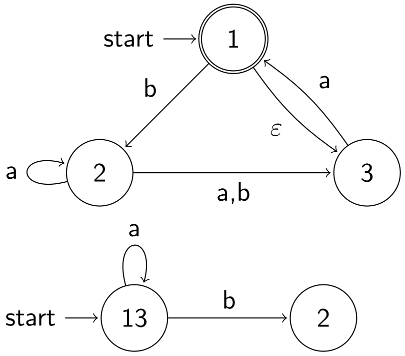
NFA与DFA的等价性



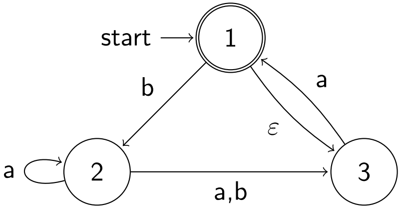
NFA与DFA的等价性

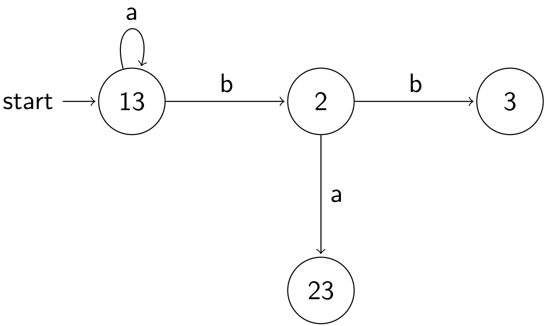


NFA与DFA的等价性

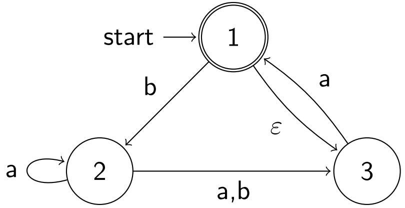


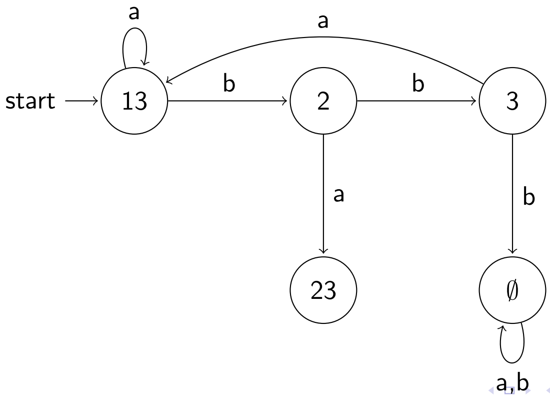
NFA与DFA的等价性



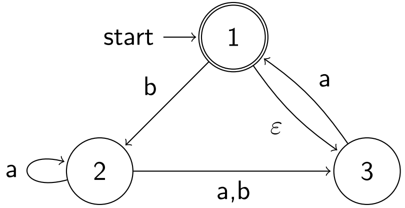


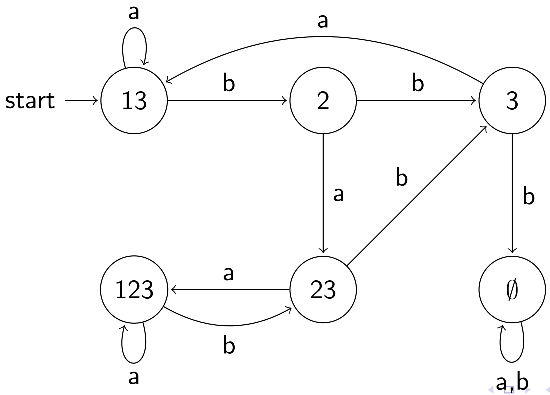
NFA与DFA的等价性



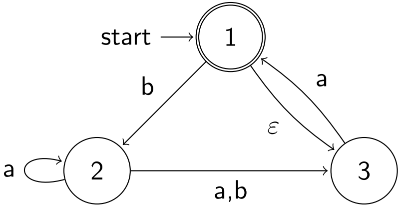


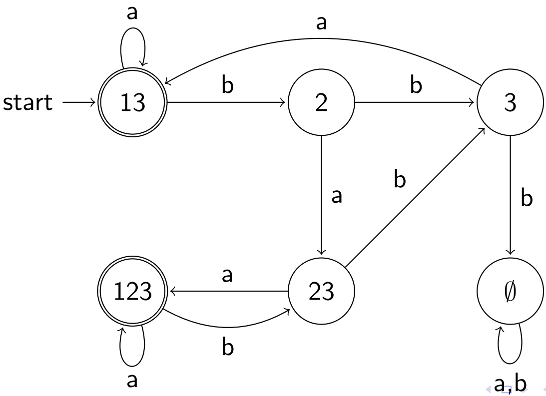
NFA与DFA的等价性





NFA与DFA的等价性





NFA与DFA的等价性

设$N = (Q, \Sigma, \delta, q_0, F)$

定义$E(q)$为：从状态q出发，只走ε转移能够到达的状态集合

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

状态集合$Q' = \mathcal{P}(Q)$

转移函数$\delta' : Q' \times \Sigma \to Q'$

其中$\delta'(S, c) = \bigcup_{q \in S, q' \in \delta(q,c)} E(q')$

$F' = \{S \subseteq Q | S \cap F \neq \emptyset\}$

NFA与DFA的等价性

回忆定义：可以被确定有限状态自动机识别的语言称为正则语言

推论：一个语言是正则的，当且仅当存在一个NFA识别它

正则语言的封闭性

并集：若A和B是正则语言，则$A \cup B$是正则语言

设DFAM_A和M_B分别识别语言A和语言B，开始状态分别为q_A和q_B

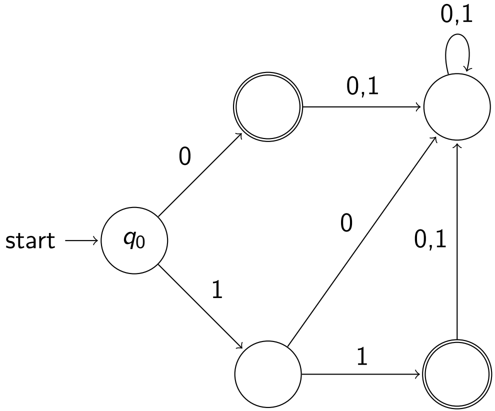
新增开始状态q_0，令$\delta(q_0, \varepsilon) = \{q_A, q_B\}$

正则语言的封闭性

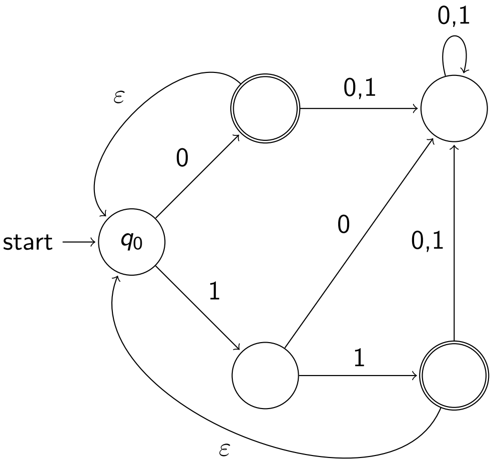
闭包：若A是正则语言，则A^*是正则语言

正则语言的封闭性

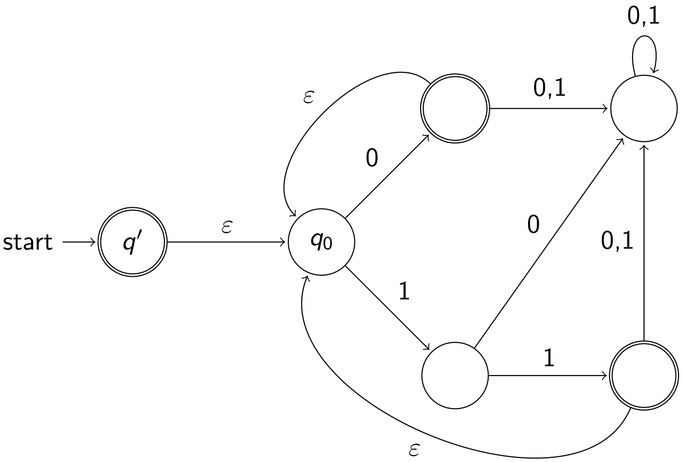
$A = \{0, 11\}$



正则语言的封闭性



正则语言的封闭性



正则语言的封闭性

存在DFA识别语言A，设其开始状态为q_0，接受状态集合为F

新增开始状态q'，增加q'到q_0的ε转移，并规定q'为接受状态

对于原来的每个接受状态$q \in F$，增加q到q_0的ε转移

正则表达式

正则表达式	语言
a	$\{a\}$
ε	$\{\varepsilon\}$
$\emptyset$	$\emptyset$
$(R_1 \cup R_2)$	$L(R_1) \cup L(R_2)$
$(R_1 \circ R_2)$	$L(R_1) \circ L(R_2)$
(R_1^*)	$L(R_1)^*$

其中R_1, R_2为正则表达式，$L(R)$表示正则表达式R对应的语言。

正则表达式

设$\Sigma = \{0,1\}$

0^*10^*：恰好包含一个1的串

$\Sigma^*1\Sigma^*$：至少包含一个1的串

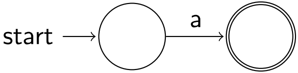
$(\Sigma\Sigma\Sigma)^*$：长度为3的倍数的串

正则表达式与DFA的等价性

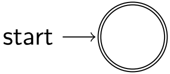
证明：一个语言是正则的当且仅当它能被某个正则表达式描述。

正则表达式与DFA的等价性

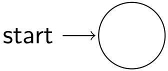
对于正则表达式a：



对于正则表达式ε：



对于正则表达式$\emptyset$：



另三种情况：参考正则语言封闭性的证明

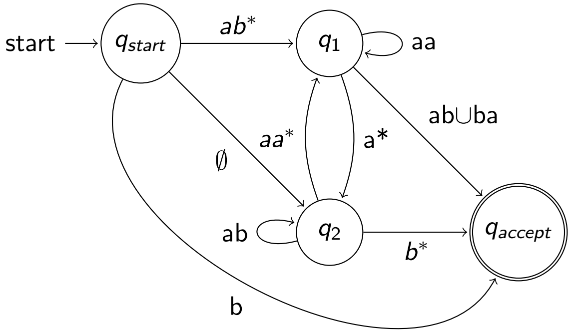
正则表达式与DFA的等价性

广义非确定有限状态自动机(Generalized Nondeterministic Finite Automaton)

$(Q, \Sigma, \delta, q_{start}, q_{accept})$

转移函数$\delta : (Q - \{q_{accept}\}) \times (Q - \{q_{start}\}) \to \mathcal{R}$

正则表达式与DFA的等价性



接受串abbabb

$q_{start} \xrightarrow{\text{abb}} q_1 \xrightarrow{\varepsilon} q_2 \xrightarrow{\text{ab}} q_2 \xrightarrow{\text{b}} q_{accept}$

正则表达式与DFA的等价性

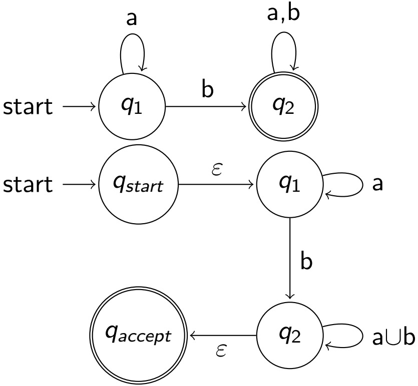
若存在$w_1, w_2, \ldots, w_n, r_0, r_1, \ldots, r_n$使得:

$s = w_1 w_2 \ldots w_n, w_i \in \Sigma^*$

$r_0 = q_{start}, r_n = q_{accept}, w_i \in L(\delta(r_{i-1}, r_i))$

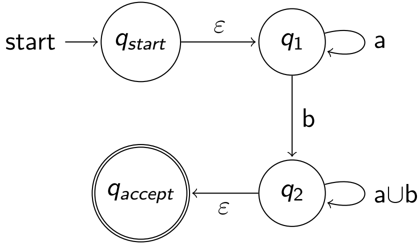
则GNFA接受串s，否则GNFA拒绝串s

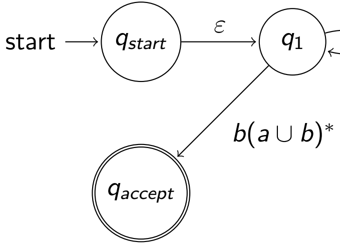
DFA→GNFA


a
a,b
start
q_1
b
q_2
start
q_{start}
ε
q_1
a
b
q_{accept}
q_2
a∪b
ε

正则表达式与DFA的等价性

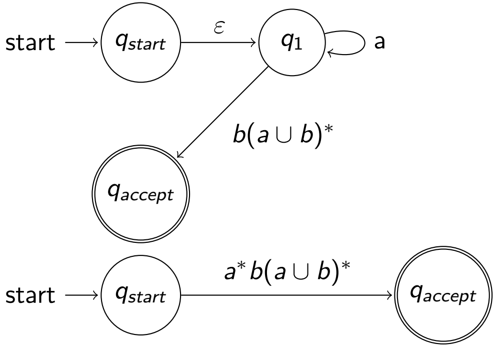
4状态GNFA→3状态GNFA





正则表达式与DFA的等价性

3状态GNFA→2状态GNFA→正则表达式



有限状态自动机的局限性

$\Sigma = \{0, 1\}$，是否存在DFA识别：0与1的个数相等的串？

有限状态自动机的局限性

$\Sigma = \{0, 1\}$，是否存在DFA识别：子串01与10出现次数相等的串？

严格证明：设存在一个包含k个状态的DFA识别该语言，考虑$0^1, 0^2, \ldots, 0^k, 0^{k+1}$

泵引理

正则语言的泵引理(pumping lemma for regular languages)

对于任意正则语言A，存在一个正整数p使得：语言A中任意长度大于等于p的串s，都可以分拆成$s = xyz$，并且满足：

1. $xy^iz \in A$，对于$i = 0, 1, 2, \ldots$
2. $y \neq \varepsilon$
3. $|xy| \leq p$

泵引理

例：$\Sigma = \{0, 1\}, A = \{x|x$含有恰好2个1$\}$

取$p = 3$，将前三位中的0作为$s = xyz$中的y

例如对于$100100 \in A$，令$x = 1, y = 0, z = 0100$

泵引理

使用泵引理证明以下语言不是正则语言：

$A = \{0^n 1^n | n = 0, 1, 2, \ldots\}$

$B = \{x | x$中0与1的个数相等$\}$

$C = \{ww | w \in \{0, 1\}^*\}$

泵引理

设正则语言A可以被一个k状态DFA识别，取$p = k$

设DFA的开始状态为q_0，转移函数为δ

对于任意$s \in A$且$|s| = n \geq p$

令$r_0 = q_0, r_i = \delta(r_{i-1}, s_i)(i = 1, 2, \ldots, p)$

由抽屉原理，必然存在$0 \leq a < b \leq p$使得$r_a = r_b$

令$x = s_1 s_2 \cdots s_a, y = s_{a+1} s_{a+2} \cdots s_b, z = s_{b+1} s_{b+2} \cdots s_n$

从状态$r_0 = q_0$出发，输入串x会转移到r_a

从状态r_a出发，输入串y会转移回到r_a

从状态r_a出发，输入串z会转移到某个接受状态

思考题

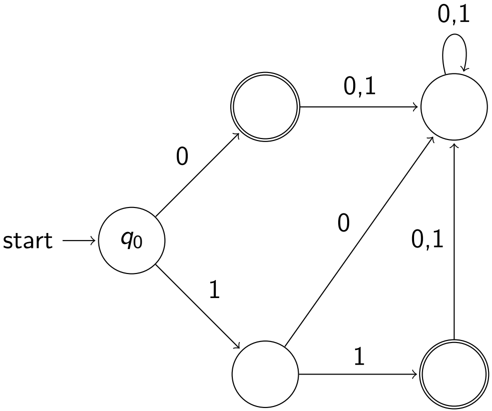
对于字母表Σ上的语言A，定义语言$S(A) = \{xy | x \in \Sigma^*, y \in A\}$
证明：若A是正则语言，则$S(A)$是正则语言

思考题

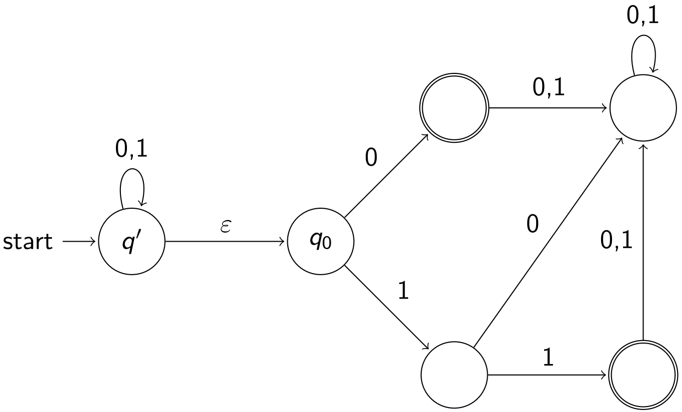
$R \to \Sigma^* R$

思考题

$A = \{0, 11\}$



思考题



例：数字搜索

来源：CTSC2004

$\Sigma = \{0, 1, \ldots, 9\}$，给出一个正则表达式$R$和一段文本$s$。求：对于哪些$x$，存在$y$使得子串$s[y..x]$被$R$匹配。

思考题

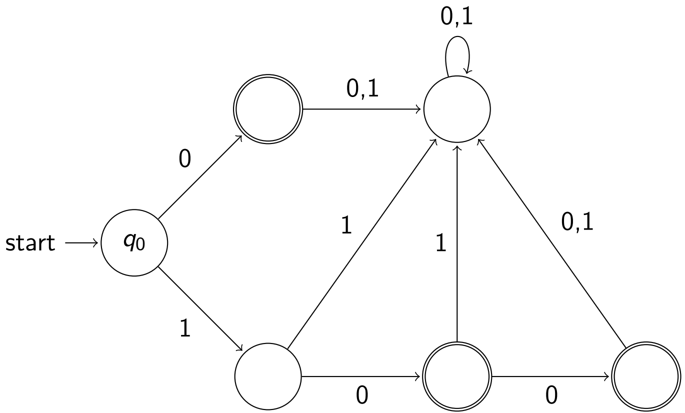
对于串$s = s_1 s_2 \ldots s_n$，定义$s^{\mathcal{R}} = s_n s_{n-1} \ldots s_1$

对于语言A，定义$A^{\mathcal{R}} = \{x^{\mathcal{R}} | x \in A\}$

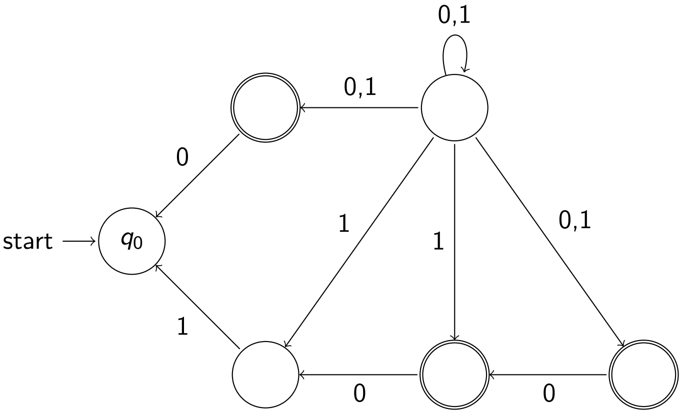
证明：若A是正则语言，则$A^{\mathcal{R}}$是正则语言

思考题

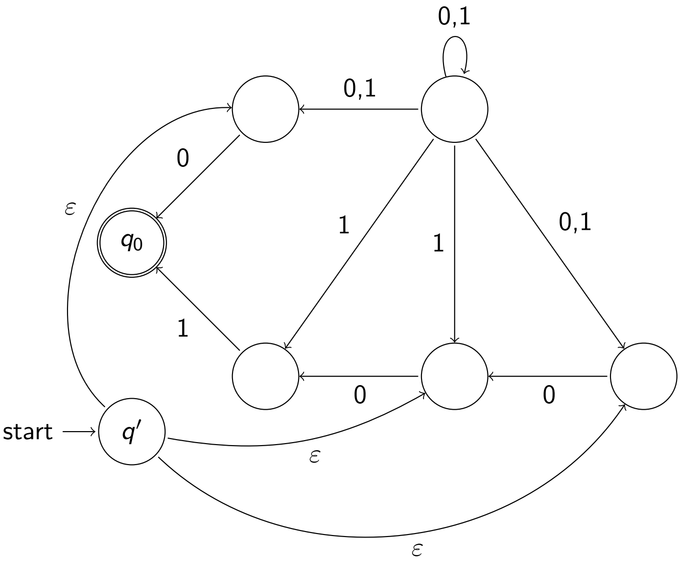
$A = \{0, 10, 100\}$



思考题



思考题



思考题

对于语言$A = \{a^i b^j c^k | i, j, k \geq 0$且若$i = 1$则$j = k\}$

1. A在泵引理中表现得“和正则语言一样”

2. A不是正则语言

例：341在以2为底的费马小定理素性测试中表现得“和素数一样”，因为$2^{341-1} \equiv 1 \pmod{341}$

思考题

令$p = 3$

$c^n = \varepsilon + c + c^{n-1}$

$b^n c^m = \varepsilon + b + b^{n-1} c^m (n > 1)$

$ab^n c^n = \varepsilon + a + b^n c^n$

$aab^n c^m = \varepsilon + aa + b^n c^m$

$aaaa^n b^m c^l = aa + a + a^n b^m c^l$

思考题

$A = \{a^i b^j c^k | i, j, k \geq 0$且若$i = 1$则$j = k\}$

$B = \{ab^i c^j | i, j \geq 0\}$

$C = \{ab^i c^i | i \geq 0\} = A \cap B$

反设A为正则语言

因为$B = L(ab^* c^*)$是正则语言，所以$C = A \cap B$为正则语言

使用泵引理可以证明C不是正则语言（考虑串$ab^p c^p$）

思考题

构造一个正则表达式描述$\Sigma = \{0, 1, 2\}$上的语言：
$\{x|x$在十进制中被3整除$\}$

思考题

构造一个正则表达式描述$\Sigma = \{0, 1\}$上的语言：
$\{x|x$不包含子串$110\}$

总结

DFA、正则语言、正则表达式的定义

正则语言的运算封闭性

DFA、NFA、正则表达式的等价性

DFA的局限性

总结

Manufactoria

regex.alf.nu