正则语言

吴 铤

2017 年 10 月 16 日

正则语言

正则语言

1 有穷自动机

2 非确定性

正则语言

1. 有穷自动机
2. 非确定性
3. 正则运算

正则语言

正则语言

例1（参考书目）

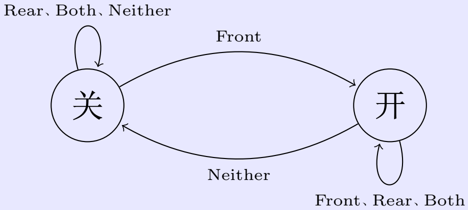
- 计算理论导引，*Michael Siper*著，唐常杰等译，机械工业出版社;
- 计算理论基础，*Lewis*著，清华大学出版社;
- 自动机理论、语言和计算导论，*J.E.Hopcroft*著，孙家驌等译，机械工业出版社
- 形式语言与自动机，陈有祺著，机械工业出版社

自动门控制器：

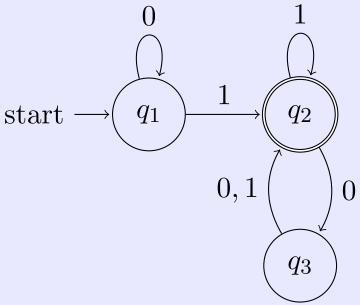
自动门控制器:

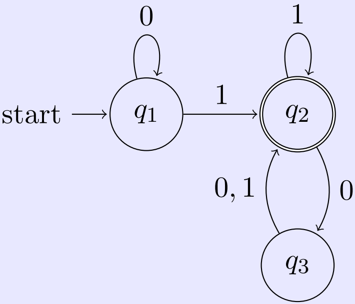
表: 自动门控制器状态转移表

	Neither	Front	Rear	Both
Closed	Closed	Open	Closed	Closed
Open	Closed	Open	Open	Open

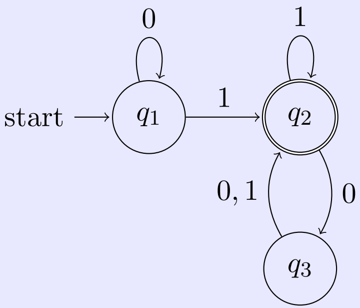

Rear、Both、Neither
Front
关
开
Neither
Front、Rear、Both

有穷自动机的形式化定义

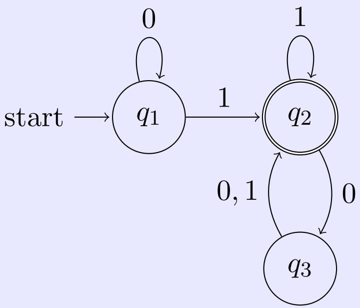




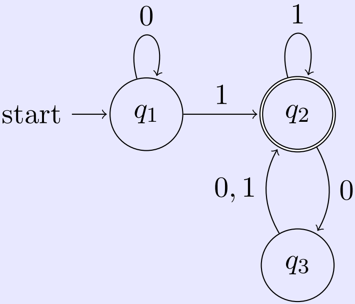
- 考虑该有限自动机是否接受字符串$1101, 1010$;



- 考虑该有限自动机是否接受字符串$1101, 1010$;
- 若A是机器M接受的全体字符串集合，则称A是M的语言，记为$L(M) = A$。也称为M识别/接受A;

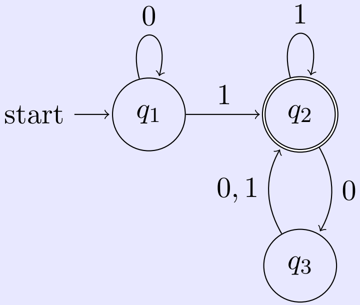


- 考虑该有限自动机是否接受字符串1101, 1010;
- 若A是机器M接受的全体字符串集合，则称A是M的语言，记为$L(M) = A$。也称为M识别/接受A;
- 如果M不接受任何字符串，则其仍然识别一个语言，即空语言$\emptyset$;



- $L(M) =$

- 考虑该有限自动机是否接受字符串1101, 1010;
- 若A是机器M接受的全体字符串集合，则称A是M的语言，记为$L(M) = A$。也称为M识别/接受A;
- 如果M不接受任何字符串，则其仍然识别一个语言，即空语言$\emptyset$;



- $L(M) = \{\omega | \omega$至少有一个1且最后的1后面有偶数个0$\}$

- 考虑该有限自动机是否接受字符串1101, 1010;
- 若A是机器M接受的全体字符串集合，则称A是M的语言，记为$L(M) = A$。也称为M识别/接受A;
- 如果M不接受任何字符串，则其仍然识别一个语言，即空语言$\emptyset$;

定义1 (有穷自动机)

有穷自动机是一个五元组$(Q, \sum, \delta, q_0, F)$，其中

定义1（有穷自动机）

有穷自动机是一个五元组$(Q, \sum, \delta, q_0, F)$，其中

- 状态集：Q是一个有穷集合；

定义1 (有穷自动机)

有穷自动机是一个五元组$(Q, \sum, \delta, q_0, F)$，其中

- 状态集：Q是一个有穷集合；
- 字母表：$\sum$是一个有穷集合；

定义1 (有穷自动机)

有穷自动机是一个五元组$(Q, \sum, \delta, q_0, F)$，其中

- 状态集：Q是一个有穷集合；
- 字母表：$\sum$是一个有穷集合；
- 转移函数：$\delta : Q \times \sum \rightarrow Q, \delta(x, a) = y$；

定义1 (有穷自动机)

有穷自动机是一个五元组$(Q, \sum, \delta, q_0, F)$，其中

- 状态集：Q是一个有穷集合；
- 字母表：$\sum$是一个有穷集合；
- 转移函数：$\delta : Q \times \sum \rightarrow Q, \delta(x, a) = y$;
- 起始状态$q_0 \in Q$;

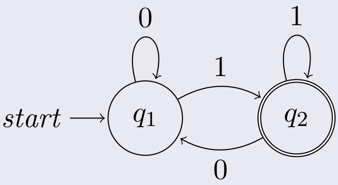
定义1 (有穷自动机)

有穷自动机是一个五元组$(Q, \sum, \delta, q_0, F)$，其中

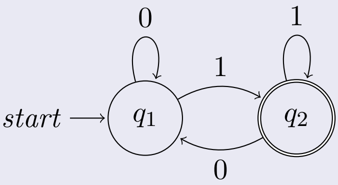
- 状态集：Q是一个有穷集合；
- 字母表：$\sum$是一个有穷集合；
- 转移函数：$\delta : Q \times \sum \to Q, \delta(x, a) = y$；
- 起始状态$q_0 \in Q$；
- 接受状态集$F \subseteq Q$；

例2 (考虑 $M = (\{q_1, q_2\}, \{0, 1\}, \delta, q_1, \{q_2\})$)

例2 (考虑 $M = (\{q_1, q_2\}, \{0,1\}, \delta, q_1, \{q_2\})$)

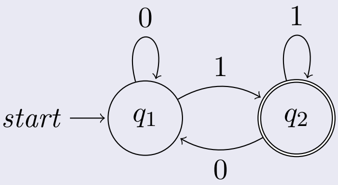


例2 (考虑$M=(\{q_1,q_2\},\{0,1\},\delta,q_1,\{q_2\})$)



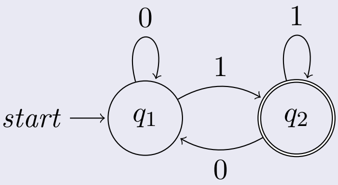
- M是否是识别$011011, 100110$

例2 (考虑$M = (\{q_1, q_2\}, \{0, 1\}, \delta, q_1, \{q_2\})$)



- M是否是识别011011, 100110
- $L(M) =$

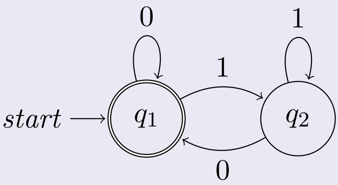
例2 (考虑$M = (\{q_1, q_2\}, \{0,1\}, \delta, q_1, \{q_2\})$)



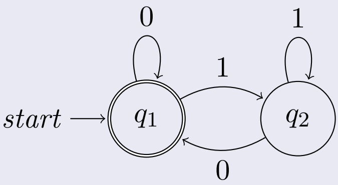
- M是否是识别011011, 100110
- $L(M) = \{\omega | \omega$以1结束$\}$

例3 (考虑$M=(\{q_1,q_2\},\{0,1\},\delta,q_1,\{q_1\}))$

例3 (考虑 $M = (\{q_1, q_2\}, \{0,1\}, \delta, q_1, \{q_1\}))$

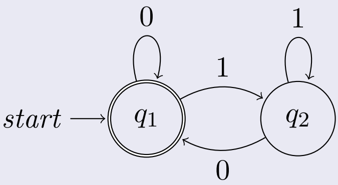


例3 (考虑$M = (\{q_1, q_2\}, \{0, 1\}, \delta, q_1, \{q_1\})$)



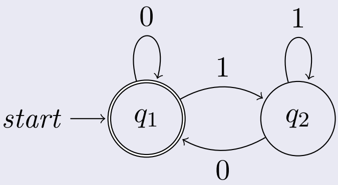
- M是否是识别$011011, 100110$

例3 (考虑$M = (\{q_1, q_2\}, \{0,1\}, \delta, q_1, \{q_1\}))$



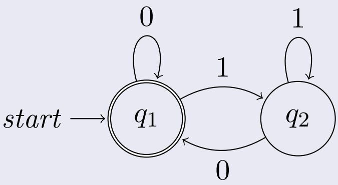
- M是否是识别$011011, 100110$
- 由于起始状态也是接受状态，因此空串$\varepsilon \in L(M)$

例3 (考虑$M=(\{q_1,q_2\},\{0,1\},\delta,q_1,\{q_1\})$)



- M是否是识别011011, 100110
- 由于起始状态也是接受状态，因此空串$\varepsilon\in L(M)$
- $L(M)=$

例3 (考虑$M = (\{q_1, q_2\}, \{0, 1\}, \delta, q_1, \{q_1\})$)



- M是否是识别011011, 100110
- 由于起始状态也是接受状态，因此空串$\varepsilon \in L(M)$
- $L(M) = \{\omega | \omega = \varepsilon$或以0结束$\}$

定义2

设$M=(Q,\sum,\delta,q_0,F)$是有限自动机，$w=w_1w_2\cdots w_n$是一个字符串且$w_i\in\sum$。如果存在Q中的状态序列

$$r_0,r_1,\cdots,r_n$$

使得

定义2

设$M=(Q,\sum,\delta,q_0,F)$是有限自动机，$w=w_1w_2\cdots w_n$是一个字符串且$w_i\in\sum$。如果存在Q中的状态序列

$$r_0,r_1,\cdots,r_n$$

使得

- $r_0=q_0$

定义2

设$M=(Q,\sum,\delta,q_0,F)$是有限自动机，$w=w_1w_2\cdots w_n$是一个字符串且$w_i\in\sum$。如果存在Q中的状态序列

$$r_0,r_1,\cdots,r_n$$

使得

- $r_0=q_0$
- $\delta(r_i,w_{i+1})=r_{i+1},i=0,1,\cdots,n-1$

定义2

设$M=(Q,\sum,\delta,q_0,F)$是有限自动机，$w=w_1w_2\cdots w_n$是一个字符串且$w_i\in\sum$。如果存在Q中的状态序列

$$r_0,r_1,\cdots,r_n$$

使得

- $r_0=q_0$
- $\delta(r_i,w_{i+1})=r_{i+1},i=0,1,\cdots,n-1$
- $r_n\in F$

定义2

设$M=(Q,\sum,\delta,q_0,F)$是有限自动机，$w=w_1w_2\cdots w_n$是一个字符串且$w_i\in\sum$。如果存在Q中的状态序列

$$r_0,r_1,\cdots,r_n$$

使得

- $r_0=q_0$
- $\delta(r_i,w_{i+1})=r_{i+1}, i=0,1,\cdots,n-1$
- $r_n\in F$

则称M**接受**w。

定义3

如果$A = \{w | M 接受 w\}$，则称M**识别**语言A。

定义3

如果$A = \{w | M\text{接受}w\}$，则称M**识别**语言A。

定义4 (正则语言)

如果一个语言被一台有限自动机识别，则称其是**正则语言**。

设计的基本思想：

设计的基本思想：

- 每接收到一个符号，必须确定目前为止的字符串是否属于某个语言；

设计的基本思想：

- 每接收到一个符号，必须确定目前为止的字符串是否属于某个语言；
- 确定所需记录的关键消息，以此确定各个状态；

设计的基本思想：

- 每接收到一个符号，必须确定目前为止的字符串是否属于某个语言；
- 确定所需记录的关键消息，以此确定各个状态；

例4

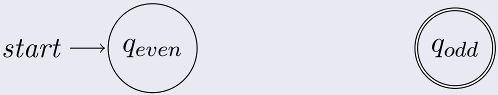
假设字母表是$\{0, 1\}$，要求构造能够识别汉明重量是奇数的字符串的自动机。

设计的基本思想：

- 每接收到一个符号，必须确定目前为止的字符串是否属于某个语言；
- 确定所需记录的关键消息，以此确定各个状态；

例4

假设字母表是$\{0,1\}$，要求构造能够识别汉明重量是奇数的字符串的自动机。

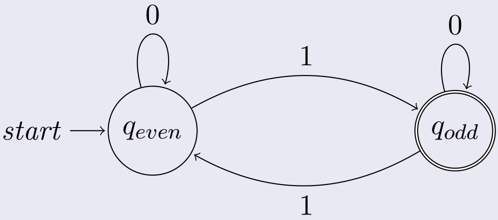


设计的基本思想：

- 每接收到一个符号，必须确定目前为止的字符串是否属于某个语言；
- 确定所需记录的关键消息，以此确定各个状态；

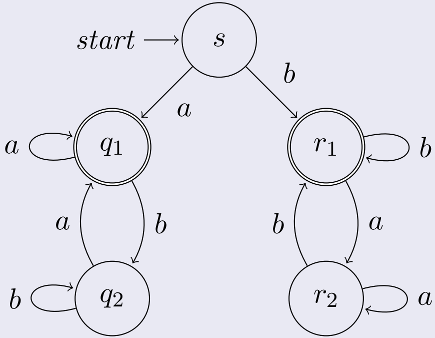
例4

假设字母表是$\{0,1\}$，要求构造能够识别汉明重量是奇数的字符串的自动机。



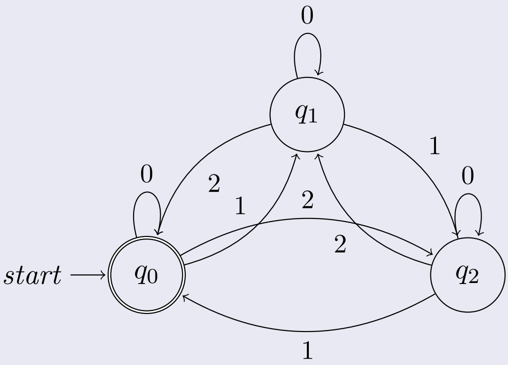
例5 (假设字母表是$\{a, b\}$，要求考虑构造$L(M) = \{\omega | \omega = a, b$或$\omega$的起始与末位字符相同$\}$的自动机)

例5 (假设字母表是$\{a,b\}$，要求考虑构造$L(M)=\{\omega|\omega=a,b$或ω的起始与末位字符相同$\}$的自动机)



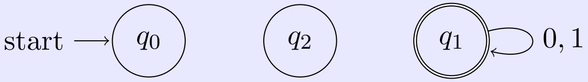
例6 (设字母表为$\{0,1,2\}$,构造$L(M)=\{\omega|\omega$结尾或数字之和是3的倍数$\}$的自动机)

例6 (设字母表为$\{0,1,2\}$,构造$L(M)=\{\omega|\omega$结尾或数字之和是3的倍数$\}$的自动机)

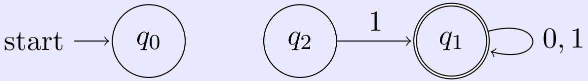


例7 (构造识别$\{w|x01y, x, y \in \{0,1\}^*\}$的自动机)

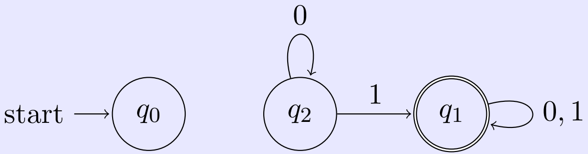
例7 (构造识别$\{w|x01y, x, y \in \{0,1\}^*\}$的自动机)



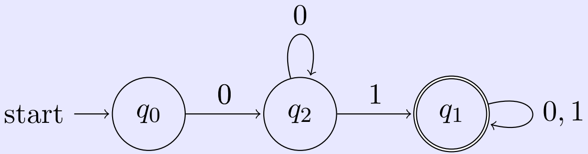
例7 (构造识别$\{w|x01y, x, y \in \{0,1\}^*\}$的自动机)



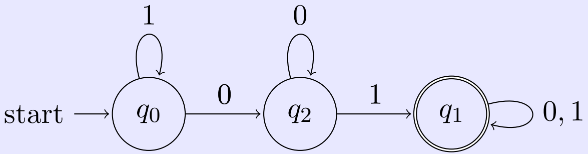
例7 (构造识别$\{w|x01y, x, y \in \{0,1\}^*\}$的自动机)



例7 (构造识别$\{w|x01y, x, y \in \{0,1\}^*\}$的自动机)



例7 (构造识别$\{w|x01y, x, y \in \{0,1\}^*\}$的自动机)



例8

假设字母表是$\{0,1\}$，要求构造能够识别包含字符串001的所有字符串的自动机。

例8

假设字母表是$\{0,1\}$，要求构造能够识别包含字符串001的所有字符串的自动机。

- q:没有得到任何可能出现在001中的符号;

例8

假设字母表是$\{0, 1\}$，要求构造能够识别包含字符串001的所有字符串的自动机。

- q:没有得到任何可能出现在001中的符号;
- q_0:得到第一个0;

例8

假设字母表是$\{0,1\}$，要求构造能够识别包含字符串001的所有字符串的自动机。

- q:没有得到任何可能出现在001中的符号;
- q_0:得到第一个0;
- q_{00}:得到00;

例8

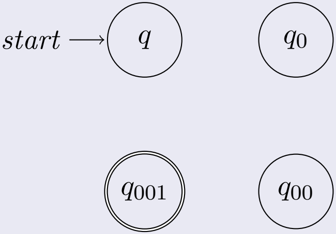
假设字母表是$\{0,1\}$，要求构造能够识别包含字符串001的所有字符串的自动机。

- q:没有得到任何可能出现在001中的符号;
- q_0:得到第一个0;
- q_{00}:得到00;
- q_{001}:得到子串001;

例8

假设字母表是$\{0,1\}$，要求构造能够识别包含字符串001的所有字符串的自动机。

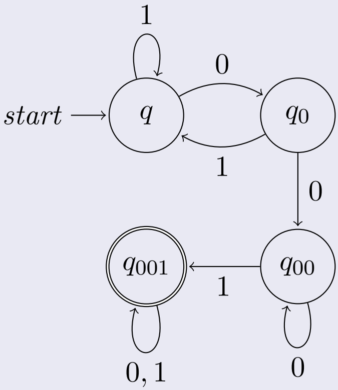
- q:没有得到任何可能出现在001中的符号;
- q_0:得到第一个0;
- q_{00}:得到00;
- q_{001}:得到子串001;



例8

假设字母表是$\{0,1\}$，要求构造能够识别包含字符串001的所有字符串的自动机。

- q:没有得到任何可能出现在001中的符号;
- q_0:得到第一个0;
- q_{00}:得到00;
- q_{001}:得到子串001;



练习1

- 构造$L(M)=\{\omega|\omega$以1开始且以0结束$\}$的DFA

练习1

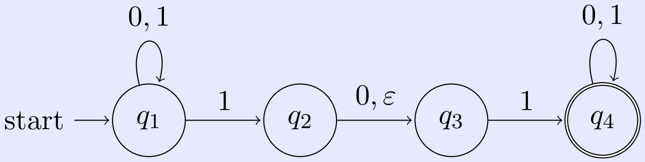
- 构造$L(M)=\{\omega|\omega$以1开始且以0结束$\}$的DFA
- 构造$L(M)=\{\omega|\omega$长度不小于3，且第三个符号是$0\}$

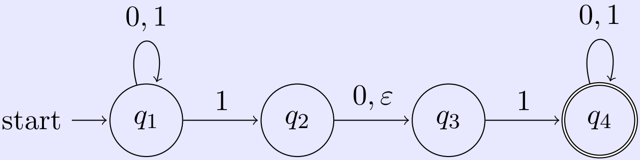
练习1

- 构造$L(M) = \{\omega|\omega 以1开始且以0结束\}$的$DFA$
- 构造$L(M) = \{\omega|\omega 长度不小于3，且第三个符号是0\}$
- 构造$L(M) = \{\omega|\omega 是除11和111以外的任意串\}$

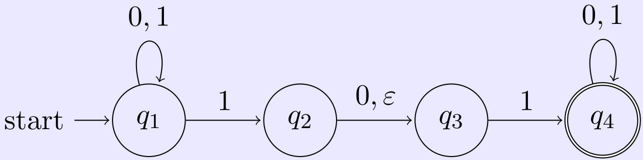
练习1

- 构造$L(M)=\{\omega|\omega\text{以}1\text{开始且以}0\text{结束}\}$的$DFA$
- 构造$L(M)=\{\omega|\omega\text{长度不小于}3\text{，且第三个符号是}0\}$
- 构造$L(M)=\{\omega|\omega\text{是除}11\text{和}111\text{以外的任意串}\}$
- 构造$L(M)=\{\varepsilon,0\}$


0, 1
0, 1
start
q_1
1
q_2
0, ε
q_3
1
q_4

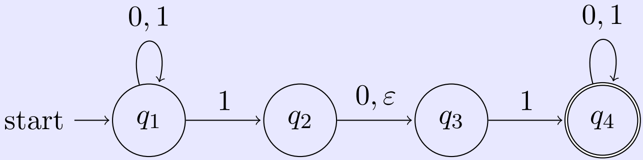


确定型有穷自动机(DFA)与非确定性有穷自动机(NFA)的区别:



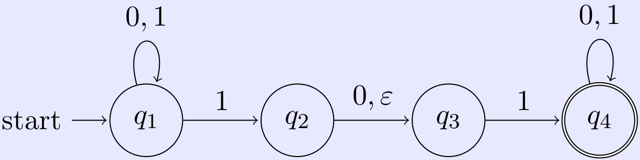
确定型有穷自动机(DFA)与非确定性有穷自动机(NFA)的区别:

- DFA中每个状态对于字母表中的每个符号恰有一个转移箭头射出;



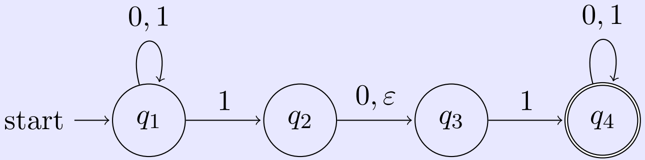
确定型有穷自动机(DFA)与非确定性有穷自动机(NFA)的区别：

- DFA中每个状态对于字母表中的每个符号恰有一个转移箭头射出；
- NFA中每个状态对于字母表中的每个符号可能有0、1或多个转移箭头射出；



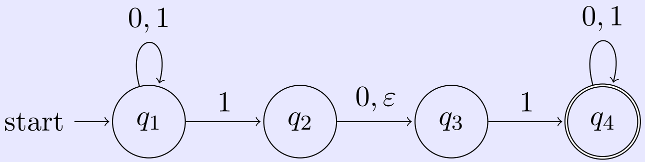
确定型有穷自动机(DFA)与非确定性有穷自动机(NFA)的区别：

- DFA中每个状态对于字母表中的每个符号恰有一个转移箭头射出；
- NFA中每个状态对于字母表中的每个符号可能有0、1或多个转移箭头射出；
- DFA中转移箭头的标号是字母表中的符号；

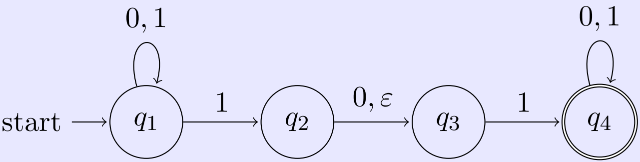


确定型有穷自动机(DFA)与非确定性有穷自动机(NFA)的区别：

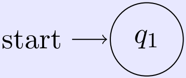
- DFA中每个状态对于字母表中的每个符号恰有一个转移箭头射出；
- NFA中每个状态对于字母表中的每个符号可能有0、1或多个转移箭头射出；
- DFA中转移箭头的标号是字母表中的符号；
- NFA中转移箭头的标号可以是字母表中的符号或ε，且从一个状态可能有多个标有ε的转移箭头；

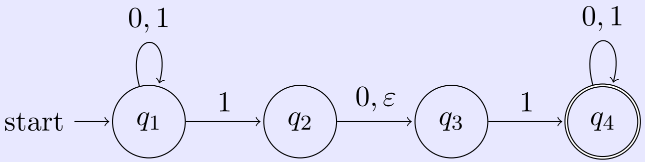


现考虑其对于输入串010110的运行情况:

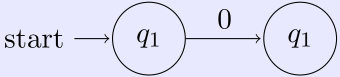


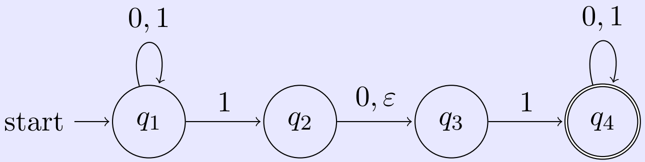
现考虑其对于输入串010110的运行情况：



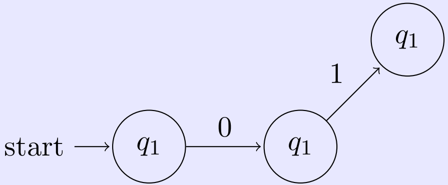


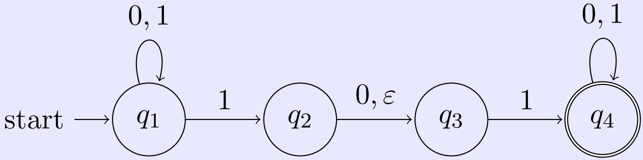
现考虑其对于输入串010110的运行情况：



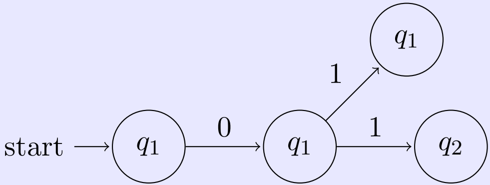


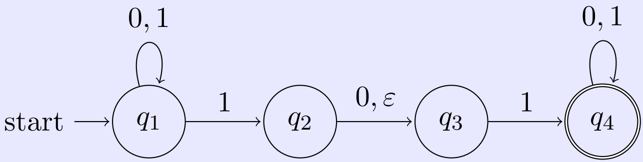
现考虑其对于输入串010110的运行情况：



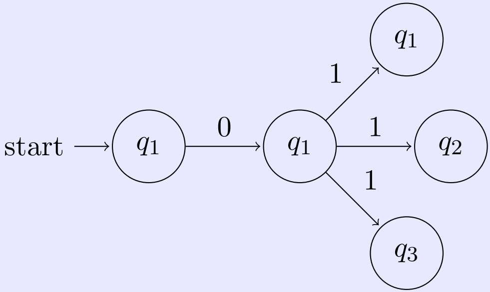


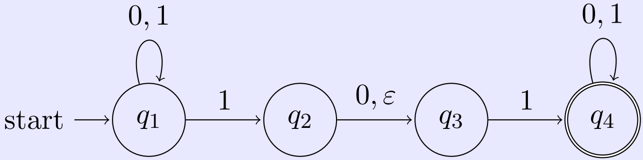
现考虑其对于输入串010110的运行情况：



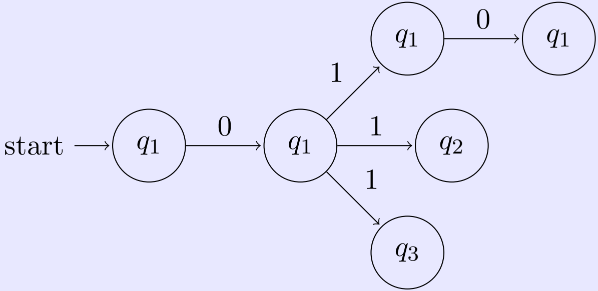


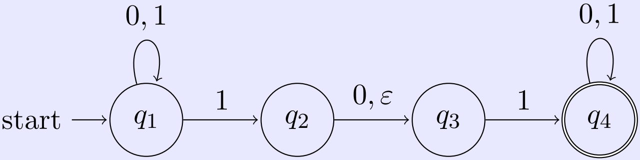
现考虑其对于输入串010110的运行情况：



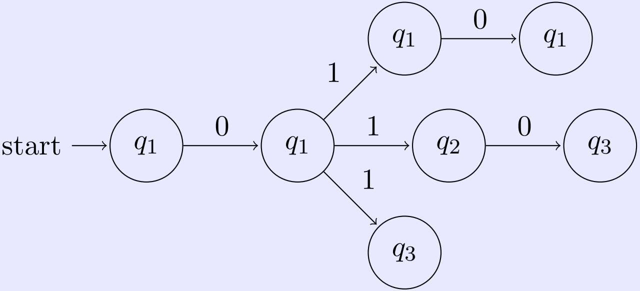


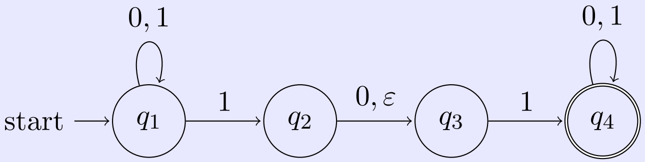
现考虑其对于输入串010110的运行情况：

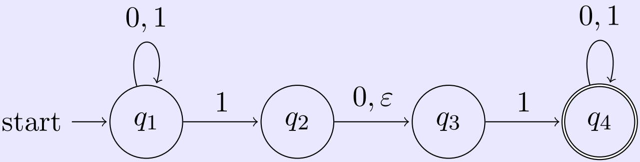




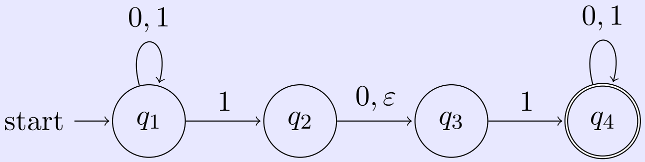
现考虑其对于输入串010110的运行情况：

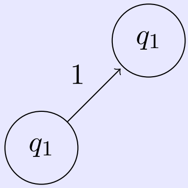


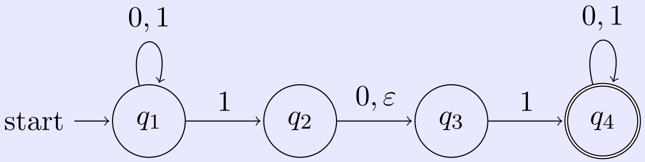

0, 1
start
q_1
1
q_2
0, ε
q_3
1
q_4
0, 1

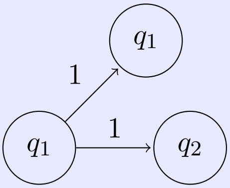

0, 1
start
q_1
1
q_2
0, ε
q_3
1
q_4
0, 1

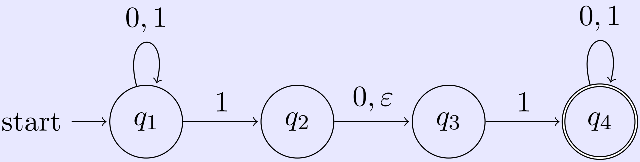

q_1

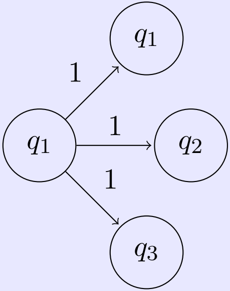

0, 1
0, 1
start
q_1
1
q_2
0, ε
q_3
1
q_4

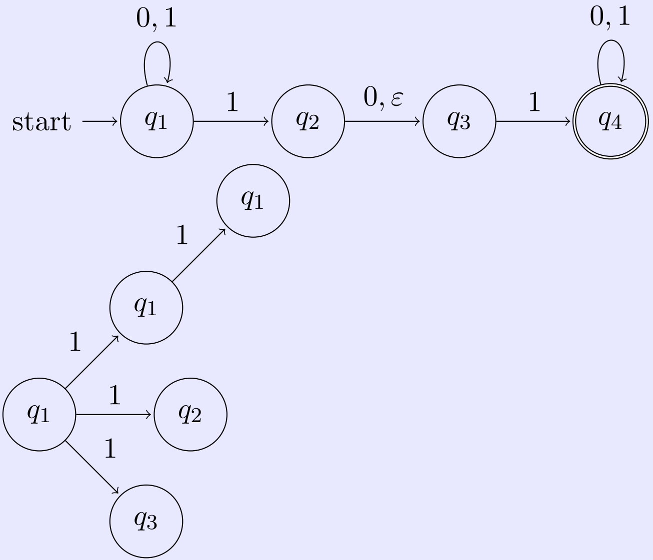

q_1
1
q_1

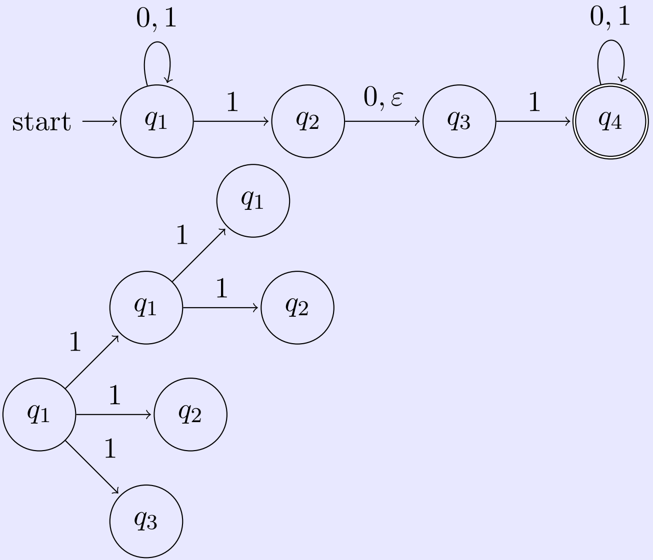

0, 1
start
q_1
1
q_2
0, ε
q_3
1
q_4
0, 1

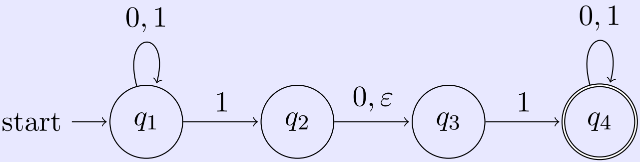

q_1
1
q_1
1
q_2


0, 1
start
q_1
1
q_2
0, ε
q_3
1
q_4
0, 1

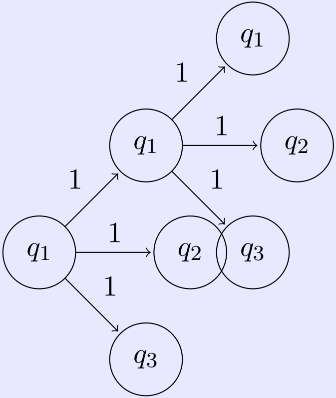

q_1
1
q_1
1
q_2
1
q_3


0, 1
start
q_1
1
q_2
0, ε
q_3
1
q_4
0, 1
q_1
1
q_1
1
q_1
1
q_2
1
q_3

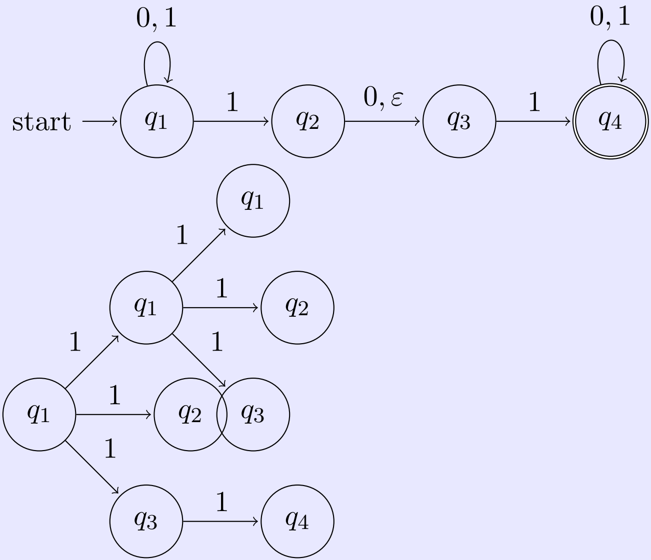

0, 1
0, 1
start
q_1
1
q_2
0, ε
q_3
1
q_4
q_1
1
q_1
1
q_2
1
q_1
1
q_2
1
q_3

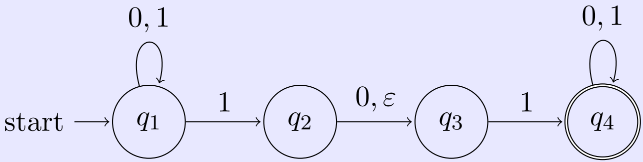


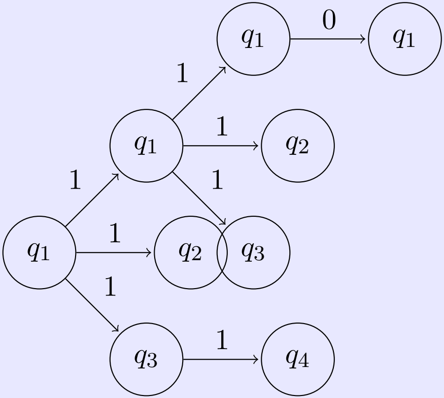
0, 1
0, 1
start
q_1
1
q_2
$0, \varepsilon$
q_3
1
q_4

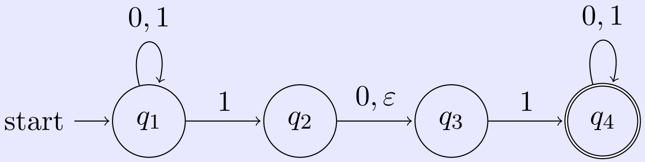


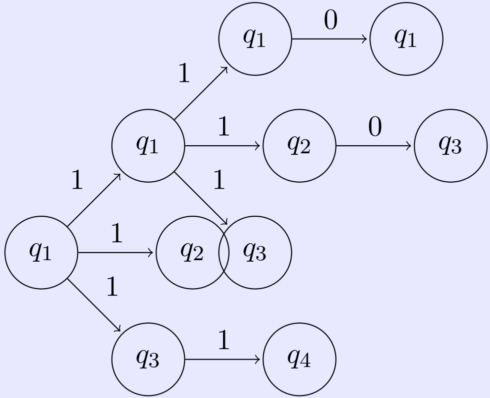
q_1
1
q_1
1
q_2
1
q_1
1
1
q_2
q_3
1
q_3

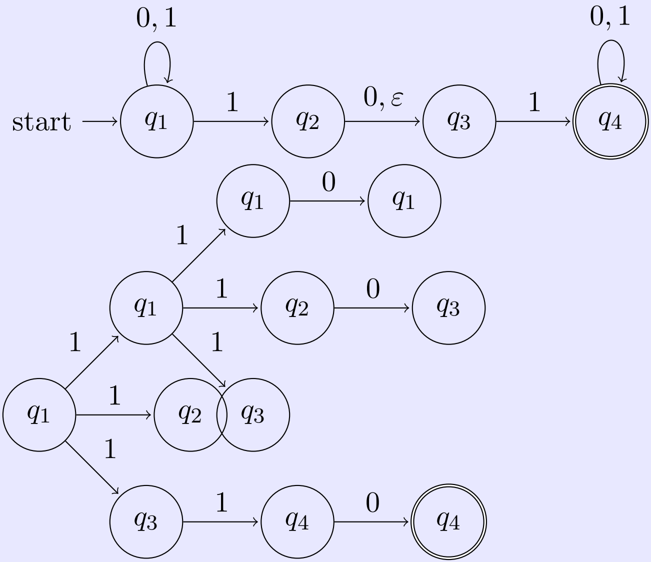

0, 1
start
q_1
1
q_2
0, ε
q_3
1
q_4
0, 1
q_1
1
q_1
1
q_2
1
1
q_1
1
q_2
q_3
1
q_3
1
q_4

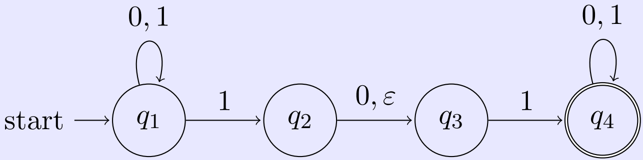

0, 1
start
q_1
1
q_2
0, ε
q_3
1
0, 1
q_4


q_1
0
q_1
1
q_1
1
q_2
1
1
q_1
1
q_2
q_3
1
q_3
1
q_4

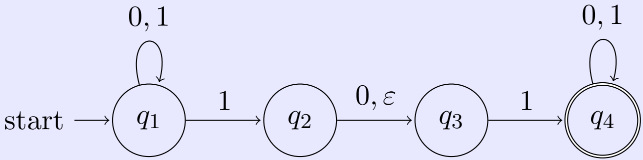

0, 1
start
q_1
1
q_2
0, ε
q_3
1
q_4
0, 1


q_1
1
q_1
1
q_1
0
q_1
1
q_2
0
q_3
1
q_3
1
q_2
q_3
1
q_3
1
q_4

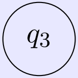

0, 1
0, 1
start
q1
1
q2
0, ε
q3
1
q4
q1
0
q1
1
q1
1
q2
0
q3
1
1
q1
1
q2
q3
1
q3
1
q4
0
q4

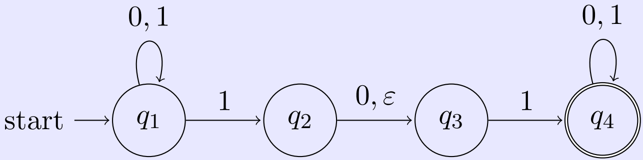


接下的输入是110：

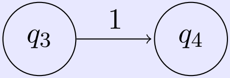


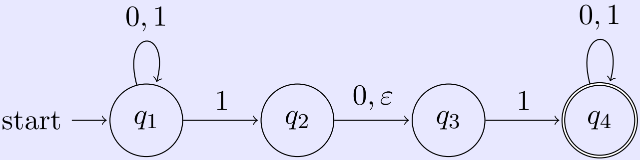
接下的输入是110：



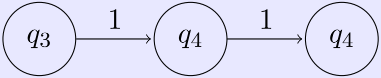


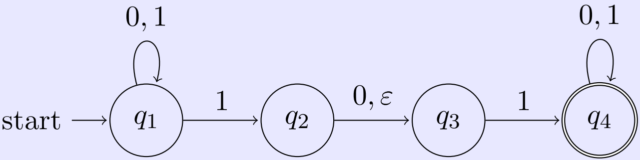
接下的输入是110:



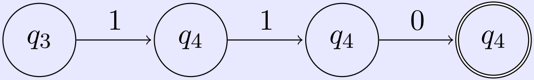


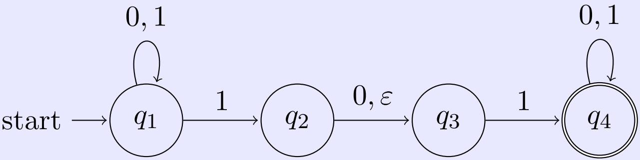
接下的输入是110：



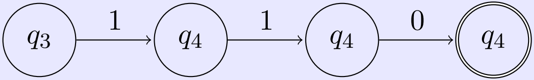


接下的输入是110:





接下的输入是110：



由此可知该NFA接受所有包含101或11为子串的字符串。

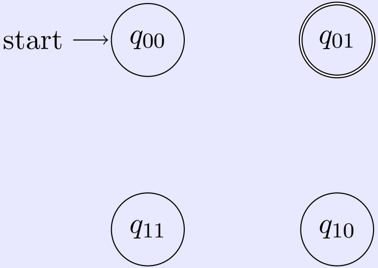
例9 (设计识别以01结尾的0, 1串的DFA、NFA)

例9 (设计识别以01结尾的0, 1串的DFA、NFA)

- 需要记录哪些状态？

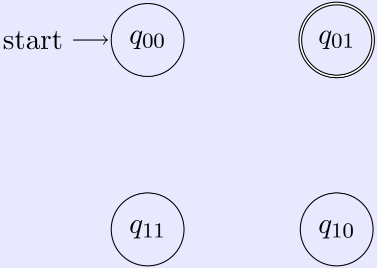
例9 (设计识别以01结尾的0, 1串的DFA、NFA)

- 需要记录哪些状态？



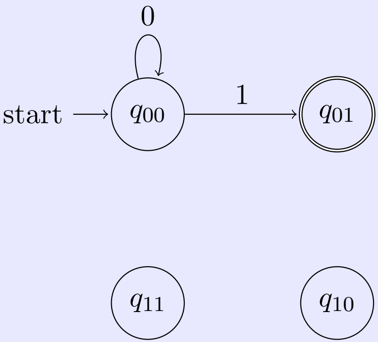
例9 (设计识别以01结尾的0, 1串的DFA、NFA)

- 需要记录哪些状态?
- 状态之间如何转换?



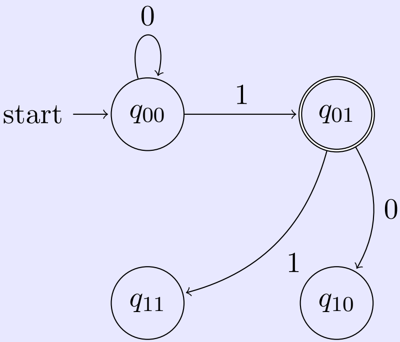
例9 (设计识别以01结尾的0, 1串的DFA、NFA)

- 需要记录哪些状态?
- 状态之间如何转换?



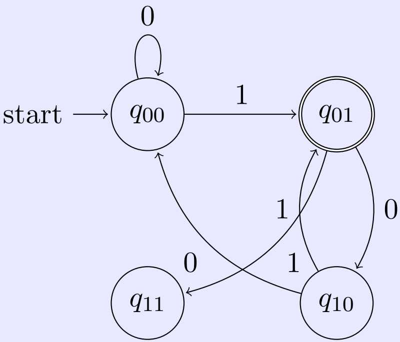
例9 (设计识别以01结尾的0, 1串的DFA、NFA)

- 需要记录哪些状态？
- 状态之间如何转换？



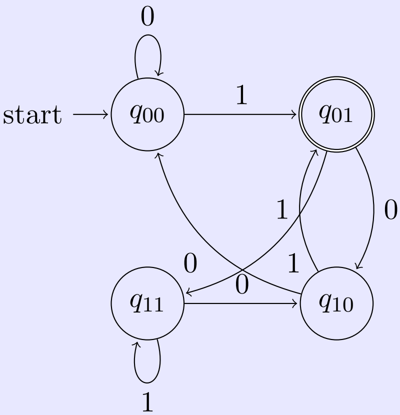
例9 (设计识别以01结尾的0, 1串的DFA、NFA)

- 需要记录哪些状态?
- 状态之间如何转换?

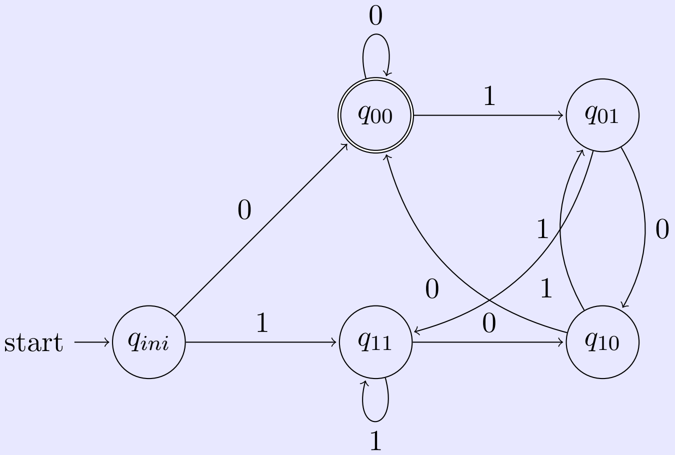


例9 (设计识别以01结尾的0, 1串的DFA、NFA)

- 需要记录哪些状态?
- 状态之间如何转换?

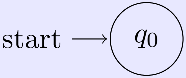


例10 (设计识别以01结尾的0,1串的DFA、NFA)

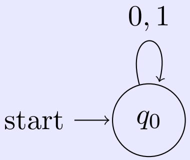


例11 (设计识别以01结尾的0, 1串的DFA、NFA)

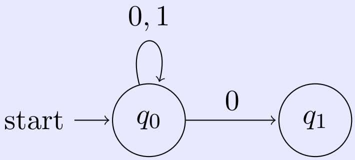
例11 (设计识别以01结尾的0, 1串的DFA、NFA)



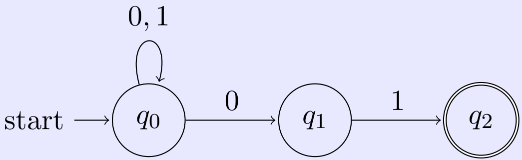
例11 (设计识别以01结尾的0, 1串的DFA、NFA)



例11 (设计识别以01结尾的0, 1串的DFA、NFA)



例11 (设计识别以01结尾的0, 1串的DFA、NFA)



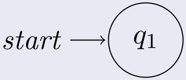
例12 (设计识别倒数第三个符号为1的NFA)

例12 (设计识别倒数第三个符号为1的NFA)

解:

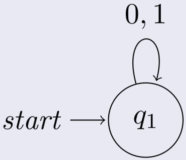
例12 (设计识别倒数第三个符号为1的NFA)

解:



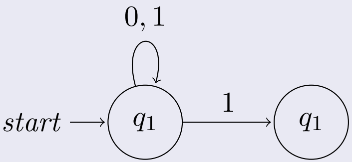
例12 (设计识别倒数第三个符号为1的NFA)

解:



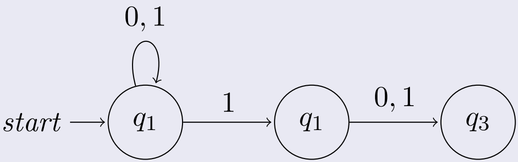
例12 (设计识别倒数第三个符号为1的NFA)

解:



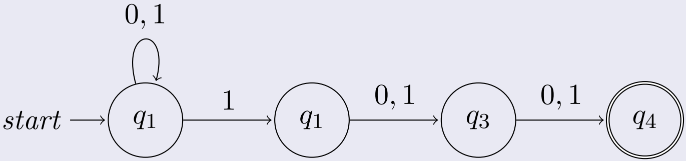
例12 (设计识别倒数第三个符号为1的NFA)

解:

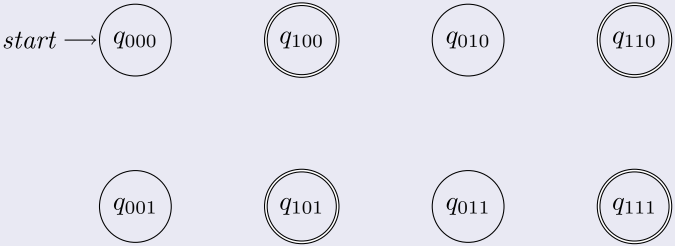


例12 (设计识别倒数第三个符号为1的NFA)

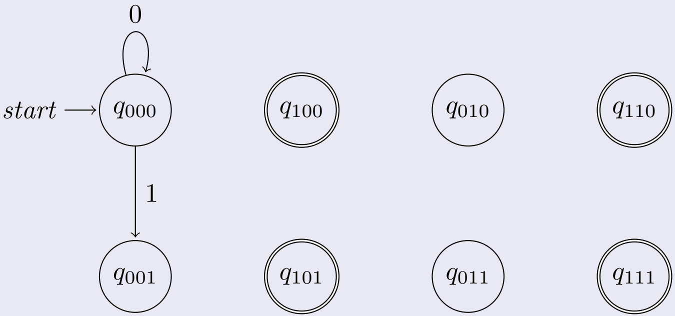
解:



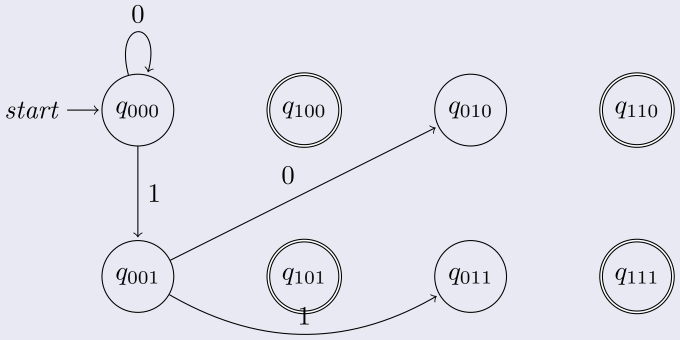
例12 (设计识别倒数第三个符号为1的NFA)



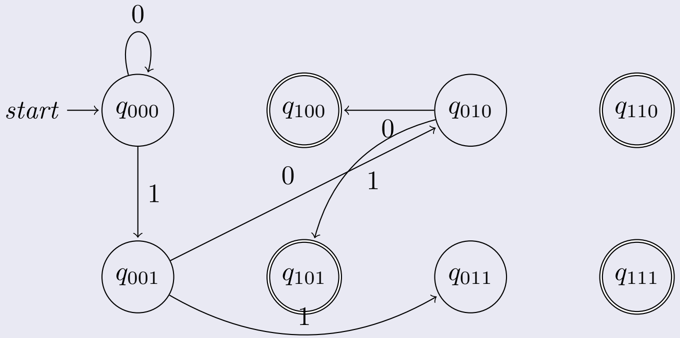
例12 (设计识别倒数第三个符号为1的NFA)



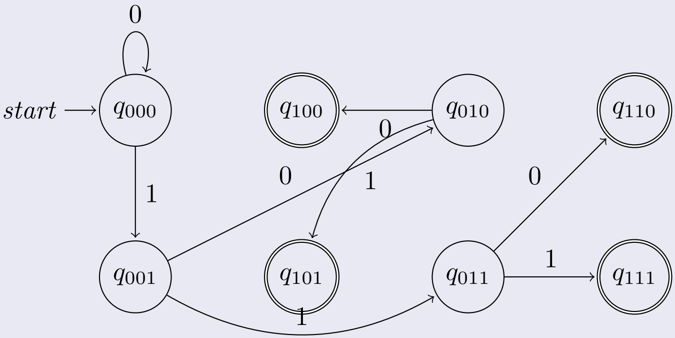
例12 (设计识别倒数第三个符号为1的NFA)



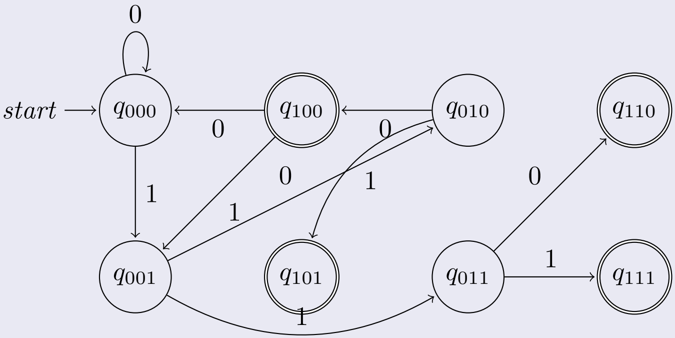
例12 (设计识别倒数第三个符号为1的NFA)



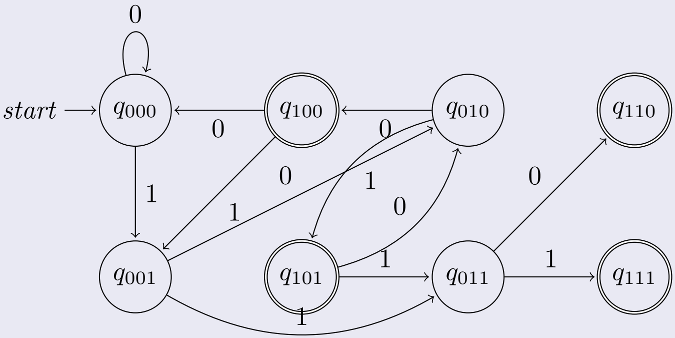
例12 (设计识别倒数第三个符号为1的NFA)



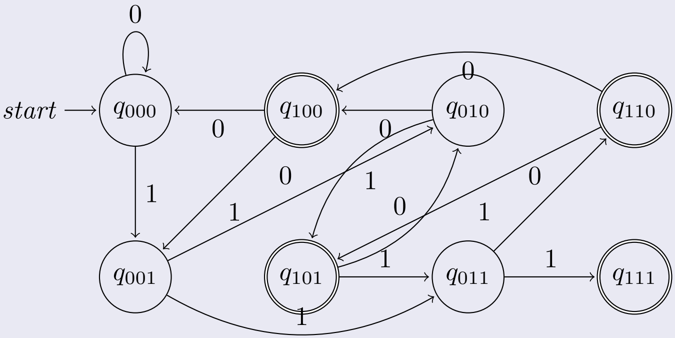
例12 (设计识别倒数第三个符号为1的NFA)



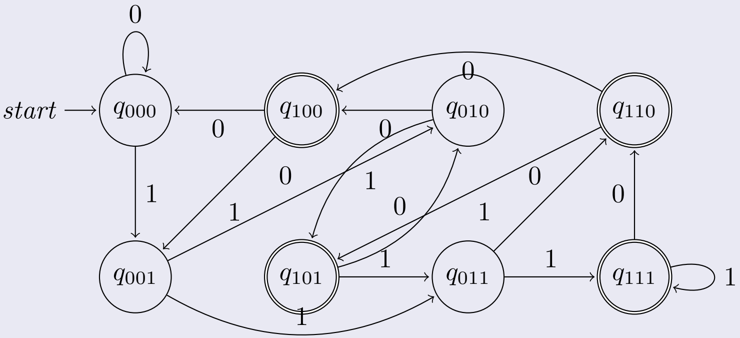
例12 (设计识别倒数第三个符号为1的NFA)



例12 (设计识别倒数第三个符号为1的NFA)



例12 (设计识别倒数第三个符号为1的NFA)



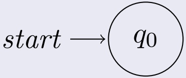
例13（设$\sum=\{0\}$，设计识别0^k，其中k是2或3倍数的NFA）

例13 (设$\sum = \{0\}$，设计识别0^k，其中k是2或3倍数的NFA)

解:

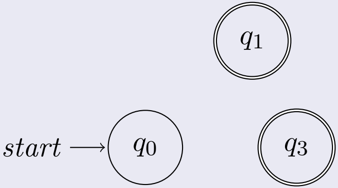
例13 (设$\sum = \{0\}$，设计识别0^k，其中k是2或3倍数的NFA)

解：



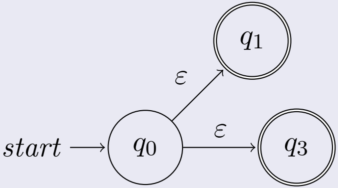
例13 (设$\sum = \{0\}$，设计识别0^k，其中k是2或3倍数的NFA)

解:



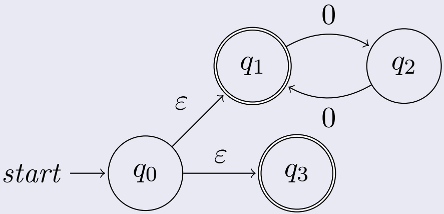
例13（设$\sum = \{0\}$，设计识别0^k，其中k是2或3倍数的NFA）

解：



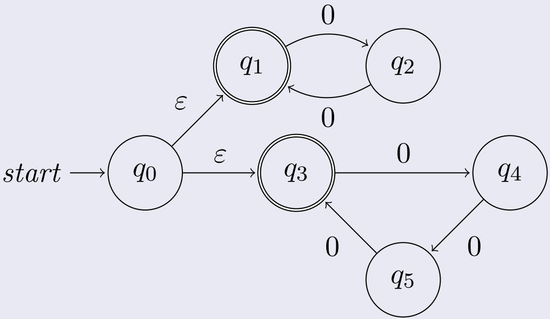
例13（设$\sum=\{0\}$，设计识别0^k，其中k是2或3倍数的NFA）

解：



例13（设$\sum = \{0\}$，设计识别0^k，其中k是2或3倍数的NFA）

解：



定义5 (非确定型有穷自动机(NFA))

定义5 (非确定型有穷自动机(NFA))

NFA是一个5元组$(Q, \sum, \delta, q_0, F)$，其中

定义5 (非确定型有穷自动机(NFA))

NFA是一个5元组$(Q, \sum, \delta, q_0, F)$，其中

- Q是有穷的状态集;

定义5 (非确定型有穷自动机(NFA))

NFA是一个5元组$(Q, \sum, \delta, q_0, F)$，其中

- Q是有穷的状态集;
- $\sum$是有穷的字母表;

定义5 (非确定型有穷自动机(NFA))

NFA是一个5元组$(Q, \sum, \delta, q_0, F)$，其中

- Q是有穷的状态集；
- $\sum$是有穷的字母表；
- $\delta : Q \times \sum_\varepsilon \rightarrow \mathcal{P}(Q)$是转移函数，其中

$$\sum_\varepsilon = \sum \cup \{\varepsilon\}$$

而$\mathcal{P}(Q)$是Q的幂集；

定义5 (非确定型有穷自动机(NFA))

NFA是一个5元组$(Q, \sum, \delta, q_0, F)$，其中

- Q是有穷的状态集;
- $\sum$是有穷的字母表;
- $\delta : Q \times \sum_\varepsilon \to \mathcal{P}(Q)$是转移函数，其中

$$\sum_\varepsilon = \sum \cup \{\varepsilon\}$$

而$\mathcal{P}(Q)$是Q的幂集;

- $q_0 \in Q$是起始状态;

定义5 (非确定型有穷自动机(NFA))

NFA是一个5元组$(Q,\sum,\delta,q_0,F)$，其中

- Q是有穷的状态集；
- $\sum$是有穷的字母表；
- $\delta : Q \times \sum_\varepsilon \to \mathcal{P}(Q)$是转移函数，其中

$$\sum_\varepsilon = \sum \cup \{\varepsilon\}$$

而$\mathcal{P}(Q)$是Q的幂集；

- $q_0 \in Q$是起始状态；
- $F \subseteq Q$是接受状态集；

定义6

设$N=(Q,\sum,\delta,q_0,F)$是一台NFA，$w=y_1y_2\cdots y_m$，其中$y_i\in\sum_\varepsilon$。若存在Q中的状态序列$r_0,r_1,\cdots,r_m$满足

定义6

设$N=(Q,\sum,\delta,q_0,F)$是一台NFA，$w=y_1y_2\cdots y_m$，其中$y_i\in\sum_\varepsilon$。若存在Q中的状态序列$r_0,r_1,\cdots,r_m$满足

1. $r_0=q_0$;

定义6

设$N=(Q,\sum,\delta,q_0,F)$是一台NFA，$w=y_1y_2\cdots y_m$，其中$y_i\in\sum_\varepsilon$。若存在Q中的状态序列$r_0,r_1,\cdots,r_m$满足

1. $r_0=q_0$;
2. $r_{i+1}\in\delta(r_0,y_{i+1}),i=0,1,\cdots,m-1$;

定义6

设$N=(Q,\sum,\delta,q_0,F)$是一台NFA，$w=y_1y_2\cdots y_m$，其中$y_i\in\sum_\varepsilon$。若存在Q中的状态序列$r_0,r_1,\cdots,r_m$满足

1. $r_0=q_0$;
2. $r_{i+1}\in\delta(r_0,y_{i+1}),i=0,1,\cdots,m-1$;
3. $r_m\in F$;

定义6

设$N=(Q,\sum,\delta,q_0,F)$是一台NFA，$w=y_1y_2\cdots y_m$，其中$y_i\in\sum_\varepsilon$。若存在Q中的状态序列$r_0,r_1,\cdots,r_m$满足

1. $r_0=q_0$;
2. $r_{i+1}\in\delta(r_0,y_{i+1}),i=0,1,\cdots,m-1$;
3. $r_m\in F$;

则称N**接受**w。

定义7 (等价)

定义7 (等价)

若两台机器识别相同的语言，则称它们是等价的。

定义7 (等价)

若两台机器识别相同的语言，则称它们是等价的。

定理1 (NFA、DFA的等价性)

每一台非确定性有穷自动机(NFA)都等价于某台确定型有穷自动机(DFA)。

定义7 (等价)

若两台机器识别相同的语言，则称它们是等价的。

定理1 (NFA、DFA的等价性)

每一台非确定性有穷自动机(NFA)都等价于某台确定型有穷自动机(DFA)。

- NFA的输入字母表包括ε；

定义7 (等价)

若两台机器识别相同的语言，则称它们是**等价**的。

定理1 (NFA、DFA的等价性)

每一台非确定性有穷自动机(NFA)都等价于某台确定型有穷自动机(DFA)。

- NFA的输入字母表包括ε；
- NFA的状态转移函数的值域可能包含多个状态，即其是状态集合的某个子集；

证明.

设$N = (Q, \Sigma, \delta, q_0, F)$是识别语言$A$的NFA，构造DFA M如下：

证明.

设$N=(Q,\Sigma,\delta,q_0,F)$是识别语言$A$的NFA，构造DFA M如下：

- 状态集合：$Q'=\mathcal{P}(Q)$；

证明.

设$N = (Q, \Sigma, \delta, q_0, F)$是识别语言$A$的NFA，构造DFA M如下：

- 状态集合：$Q' = \mathcal{P}(Q)$；
- 状态转移函数：

$$\delta'(R, a) = \{q \in Q \mid \exists r \in R, q \in \quad \delta(r, a))\}$$

□

证明.

设$N=(Q,\Sigma,\delta,q_0,F)$是识别语言$A$的NFA，构造DFA M如下：

- 状态集合：$Q'=\mathcal{P}(Q)$；
- 状态转移函数：

$$\delta'(R,a)=\{q\in Q\mid \exists r\in R, q\in E(\delta(r,a))\}$$

其中$E(R)=\{q|$从R出发沿0个或多个ε箭头可到达$q\}$

$\square$

证明.

设$N=(Q,\Sigma,\delta,q_0,F)$是识别语言$A$的NFA，构造DFA M如下：

- 状态集合：$Q'=\mathcal{P}(Q)$；
- 状态转移函数：

$$\delta'(R,a)=\{q\in Q\mid \exists r\in R, q\in E(\delta(r,a))\}$$

其中$E(R)=\{q|$从R出发沿0个或多个ε箭头可到达$q\}$

- 初始状态：$q_0'=E\{q_0\}$；

$\square$

证明.

设$N=(Q,\Sigma,\delta,q_0,F)$是识别语言$A$的NFA，构造DFA M如下：

- 状态集合：$Q'=\mathcal{P}(Q)$；
- 状态转移函数：

$$\delta'(R,a)=\{q\in Q \mid \exists r\in R, q\in E(\delta(r,a))\}$$

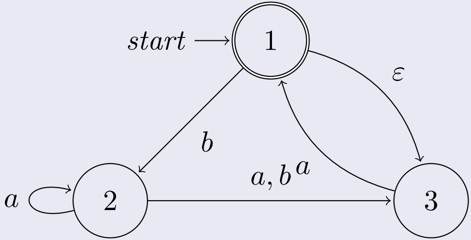
其中$E(R)=\{q|$从R出发沿0个或多个ε箭头可到达$q\}$

- 初始状态：$q_0'=E\{q_0\}$；
- 接收状态：$F'=\{R\in Q'|R$中包含N的一个接受状态$\}$；

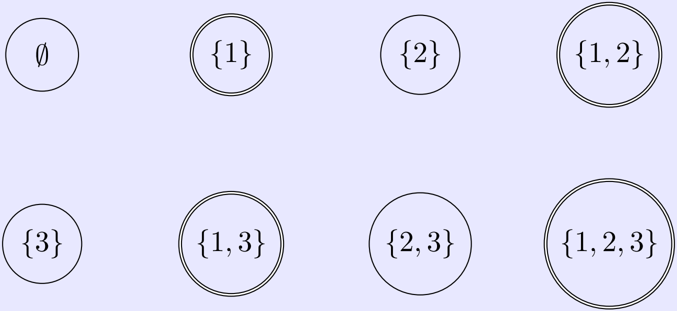
$\square$

例14 (NFA、DFA的转换)

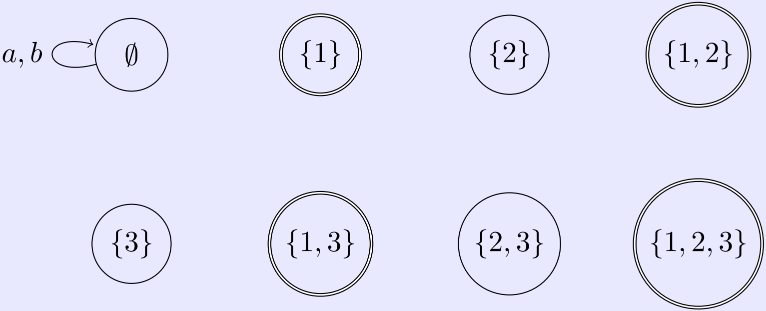
设N为如下所示的NFA



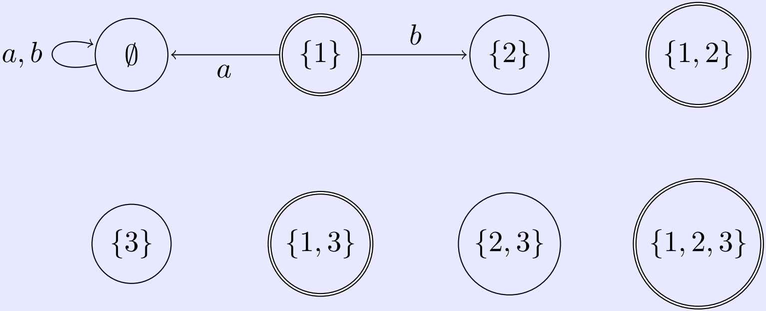
求与之等价的DFA。


∅
{1}
{2}
{1, 2}
{3}
{1, 3}
{2, 3}
{1, 2, 3}

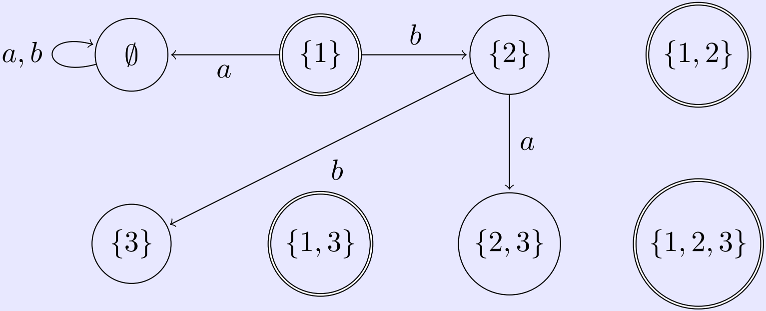
NFA、DFA的等价性

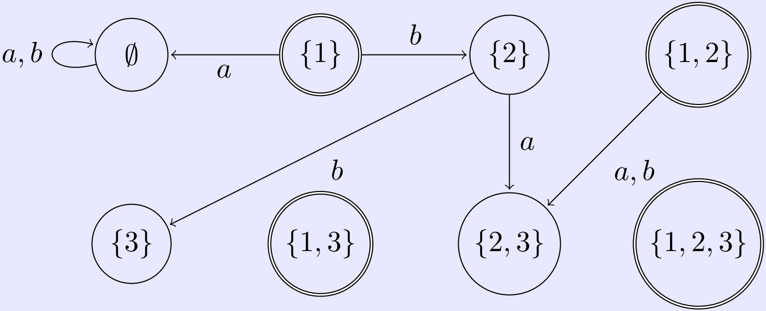


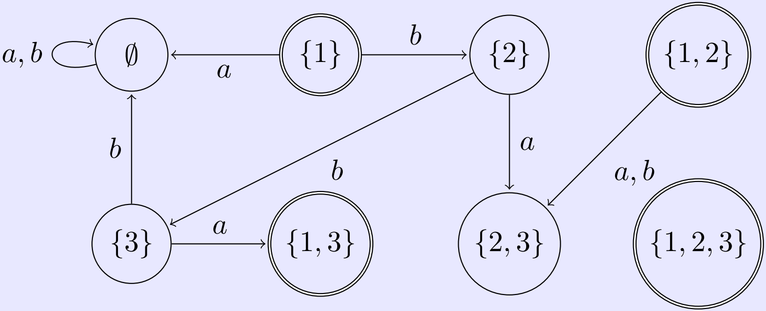
NFA、DFA的等价性

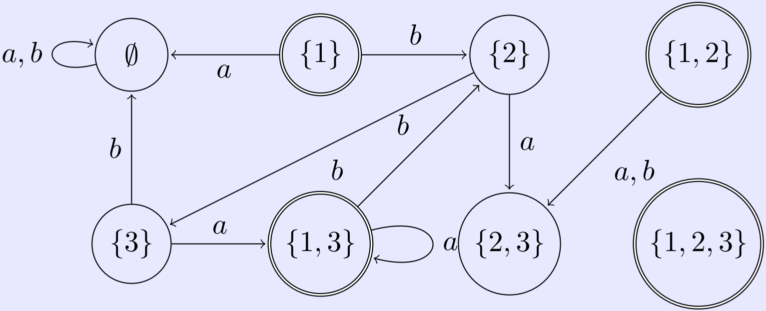


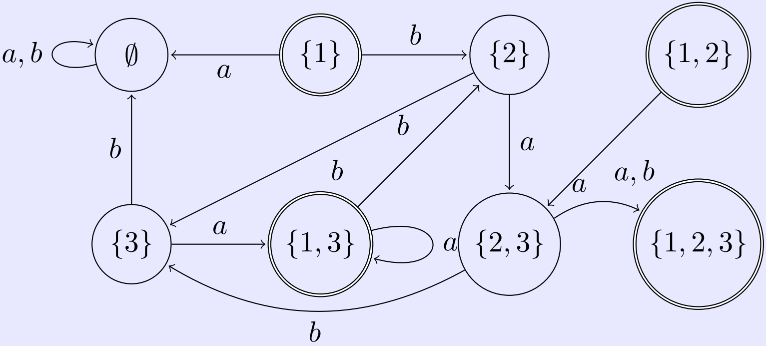
NFA、DFA的等价性

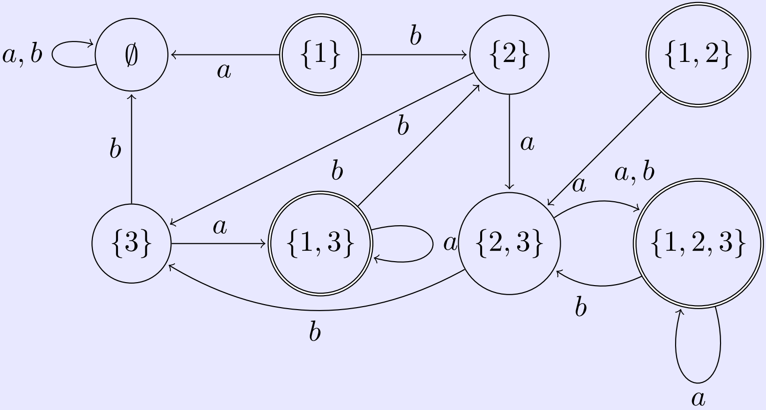


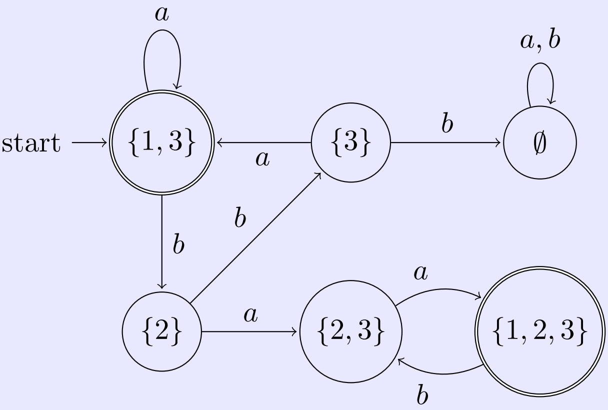

a, b
$\emptyset$
a
$\{1\}$
b
$\{2\}$
$\{1,2\}$
a
b
a, b
$\{3\}$
$\{1,3\}$
$\{2,3\}$
$\{1,2,3\}$


a, b
∅
a
{1}
b
{2}
{1, 2}
b
b
a
a, b
{3}
a
{1, 3}
{2, 3}
{1, 2, 3}


a, b
$\emptyset$
a
$\{1\}$
b
$\{2\}$
$\{1,2\}$
b
b
a
b
a, b
$\{3\}$
a
$\{1,3\}$
a
$\{2,3\}$
$\{1,2,3\}$


a, b
$\emptyset$
a
$\{1\}$
b
$\{2\}$
$\{1, 2\}$
b
b
a
b
a
a, b
$\{3\}$
a
$\{1, 3\}$
a
$\{2, 3\}$
$\{1, 2, 3\}$
b


a, b
$\emptyset$
a
$\{1\}$
b
$\{2\}$
$\{1, 2\}$
b
b
a
b
a
a, b
$\{3\}$
a
$\{1, 3\}$
a
$\{2, 3\}$
$\{1, 2, 3\}$
b
b
a


start
a
{1, 3}
{3}
b
∅
a, b
a
b
b
{2}
a
{2, 3}
a
{1, 2, 3}
b

定义8 (正则运算)

设A, B是两个语言，定义正则运算:

定义8 (正则运算)

设A, B是两个语言，定义正则运算:

- 并: $A \cup B = \{x | x \in A \text{ 或 } x \in B\}$

定义8 (正则运算)

设A, B是两个语言，定义正则运算:

- 并: $A \cup B = \{x | x \in A \text{ 或 } x \in B\}$
- 连结: $A \circ B = \{xy | x \in A \text{ 且 } y \in B\}$

定义8 (正则运算)

设A, B是两个语言，定义正则运算:

- 并: $A \cup B = \{x | x \in A \text{ 或 } x \in B\}$
- 连结: $A \circ B = \{xy | x \in A \text{ 且 } y \in B\}$
- 星号: $A^* = \{x_1 x_2 \cdots x_k | k \geq 0 \text{且每个} x_i \in A\}$

定义8 (正则运算)

设A, B是两个语言，定义正则运算:

- 并: $A \cup B = \{x | x \in A \text{ 或 } x \in B\}$
- 连结: $A \circ B = \{xy | x \in A \text{ 且 } y \in B\}$
- 星号: $A^* = \{x_1 x_2 \cdots x_k | k \geq 0 \text{且每个} x_i \in A\}$

注意：无论A是什么，空串$\varepsilon \in A^*$。

定义8 (正则运算)

设A, B是两个语言，定义正则运算:

- 并: $A \cup B = \{x | x \in A \text{ 或 } x \in B\}$
- 连结: $A \circ B = \{xy | x \in A \text{ 且 } y \in B\}$
- 星号: $A^* = \{x_1 x_2 \cdots x_k | k \geq 0 \text{且每个} x_i \in A\}$

注意：无论A是什么，空串$\varepsilon \in A^*$。

特别地：$\emptyset^* =$

定义8（正则运算）

设A, B是两个语言，定义正则运算：

- 并：$A \cup B = \{x | x \in A \text{ 或 } x \in B\}$
- 连结：$A \circ B = \{xy | x \in A \text{ 且 } y \in B\}$
- 星号：$A^* = \{x_1 x_2 \cdots x_k | k \geq 0 \text{且每个} x_i \in A\}$

注意：无论A是什么，空串$\varepsilon \in A^*$。

特别地：$\emptyset^* = \{\varepsilon\}$。

定理2（正则语言的封闭性）

- "正则语言类在并运算下封闭";

定理2 (正则语言的封闭性)

- “正则语言类在并运算下封闭”;
- “正则语言类在连结运算下封闭”;

定理2（正则语言的封闭性）

- “正则语言类在并运算下封闭”;
- “正则语言类在连结运算下封闭”;
- “正则语言类在星号运算下封闭”;

定理2 (正则语言的封闭性)

- "正则语言类在并运算下封闭";
- "正则语言类在连结运算下封闭";
- "正则语言类在星号运算下封闭";
- 正则语言类在补下封闭;

定理2 (正则语言的封闭性)

- “正则语言类在并运算下封闭”;
- “正则语言类在连结运算下封闭”;
- “正则语言类在星号运算下封闭”;
- 正则语言类在补下封闭;
- 正则语言类在交下封闭;

定理3 (正则语言类在并 运算下封闭)

定理3 (正则语言类在并 运算下封闭)

设DFA M_1, M_2分别识别正则语言A_1, A_2，其中

$$M_i = (Q_i, \Sigma, \delta_i, q_i, F_i), i \in \{0, 1\}$$

则构造识别$A_1 \cup A_2$的DFA M如下：

定理3 (正则语言类在并 运算下封闭)

设DFA M_1, M_2分别识别正则语言A_1, A_2，其中

$$M_i = (Q_i, \Sigma, \delta_i, q_i, F_i), i \in \{0, 1\}$$

则构造识别$A_1 \cup A_2$的DFA M如下：

- $Q = Q_1 \times Q_2$;

定理3 (正则语言类在并 运算下封闭)

设DFA M_1, M_2分别识别正则语言A_1, A_2，其中

$$M_i = (Q_i, \Sigma, \delta_i, q_i, F_i), i \in \{0, 1\}$$

则构造识别$A_1 \cup A_2$的DFA M如下：

- $Q = Q_1 \times Q_2$;
- $\Sigma' = \Sigma$;

定理3 (正则语言类在并 运算下封闭)

设DFA M_1, M_2分别识别正则语言A_1, A_2，其中

$$M_i = (Q_i, \Sigma, \delta_i, q_i, F_i), i \in \{0, 1\}$$

则构造识别$A_1 \cup A_2$的DFA M如下：

- $Q = Q_1 \times Q_2$;
- $\Sigma' = \Sigma$;
- $\delta((r_1, r_2), a) = \{\delta_1(r_1, a), \delta_2(r_2, a)\}$;

定理3 (正则语言类在并 运算下封闭)

设DFA M_1, M_2分别识别正则语言A_1, A_2，其中

$$M_i = (Q_i, \Sigma, \delta_i, q_i, F_i), i \in \{0, 1\}$$

则构造识别$A_1 \cup A_2$的DFA M如下：

- $Q = Q_1 \times Q_2$;
- $\Sigma' = \Sigma$;
- $\delta((r_1, r_2), a) = \{\delta_1(r_1, a), \delta_2(r_2, a)\}$;
- $q_0 = (q_1, q_2)$

定理3 (正则语言类在并运算下封闭)

设DFA M_1, M_2分别识别正则语言A_1, A_2，其中

$$M_i = (Q_i, \Sigma, \delta_i, q_i, F_i), i \in \{0, 1\}$$

则构造识别$A_1 \cup A_2$的DFA M如下：

- $Q = Q_1 \times Q_2$;
- $\Sigma' = \Sigma$;
- $\delta((r_1, r_2), a) = \{\delta_1(r_1, a), \delta_2(r_2, a)\}$;
- $q_0 = (q_1, q_2)$
- $F = \{(r_1, r_2) \mid (r_1 \in F_1) \vee (r_2 \in F_2)\}$

定理3 (正则语言类在并"交"运算下封闭)

设DFA M_1, M_2分别识别正则语言A_1, A_2，其中

$$M_i = (Q_i, \Sigma, \delta_i, q_i, F_i), i \in \{0, 1\}$$

则构造识别$A_1 \cup A_2$的DFA M如下：

- $Q = Q_1 \times Q_2$;
- $\Sigma' = \Sigma$;
- $\delta((r_1, r_2), a) = \{\delta_1(r_1, a), \delta_2(r_2, a)\}$;
- $q_0 = (q_1, q_2)$
- $F = \{(r_1, r_2) \mid (r_1 \in F_1) \vee (r_2 \in F_2)\}$

定理3 (正则语言类在并”交”运算下封闭)

设DFA M_1, M_2分别识别正则语言A_1, A_2，其中

$$M_i = (Q_i, \Sigma, \delta_i, q_i, F_i), i \in \{0, 1\}$$

则构造识别$A_1 \cup A_2$的DFA M如下：

- $Q = Q_1 \times Q_2$;
- $\Sigma' = \Sigma$;
- $\delta((r_1, r_2), a) = \{\delta_1(r_1, a), \delta_2(r_2, a)\}$;
- $q_0 = (q_1, q_2)$
- $F = \{(r_1, r_2) \mid (r_1 \in F_1) \vee (r_2 \in F_2)\}\{(r_1, r_2) \mid (r_1 \in F_1) \wedge (r_2 \in F_2)\}$

定理4 (“正则语言类在并运算下封闭”)

考虑NFA, DFA的等价性

- $Q = \{q_0\} \cup Q_1 \cup Q_2$

定理4 (“正则语言类在并运算下封闭”)

考虑NFA, DFA的等价性

- $Q = \{q_0\} \cup Q_1 \cup Q_2$
- $\delta(q,a) = \begin{cases} \delta_1(q,a) & q \in Q_1 \\ \delta_2(q,a) & q \in Q_2 \\ \{q_1, q_2\} & q = q_0, a = \varepsilon \\ \emptyset & q = q_0, a \neq \varepsilon \end{cases}$

定理4（“正则语言类在并运算下封闭”）

考虑NFA, DFA的等价性

- $Q = \{q_0\} \cup Q_1 \cup Q_2$
- $\delta(q,a) = \begin{cases} \delta_1(q,a) & q \in Q_1 \\ \delta_2(q,a) & q \in Q_2 \\ \{q_1, q_2\} & q = q_0, a = \varepsilon \\ \emptyset & q = q_0, a \neq \varepsilon \end{cases}$
- $F = F_1 \cup F_2$

定理5（“正则语言类在连接运算下封闭”）

- $Q = Q_1 \cup Q_2$

定理5（“正则语言类在连接运算下封闭”）

- $Q = Q_1 \cup Q_2$
- q_1

定理5（“正则语言类在连接运算下封闭”）

- $Q = Q_1 \cup Q_2$
- q_1
- $\delta(q,a) = \begin{cases} \delta_1(q,a) & q \in Q_1, q \notin F_1 \\ \delta_1(q,a) & q \in F_1, a \neq \varepsilon \\ \delta_1(q,a) \cup \{q_2\} & q \in F_1, a = \varepsilon \\ \delta_2(q,a) & q \in Q_2 \end{cases}$

定理5（“正则语言类在连接运算下封闭”）

- $Q = Q_1 \cup Q_2$
- q_1
- $\delta(q,a) = \begin{cases} \delta_1(q,a) & q \in Q_1, q \notin F_1 \\ \delta_1(q,a) & q \in F_1, a \neq \varepsilon \\ \delta_1(q,a) \cup \{q_2\} & q \in F_1, a = \varepsilon \\ \delta_2(q,a) & q \in Q_2 \end{cases}$
- $F = F_2$

定理6 (“正则语言类在星号运算下封闭”)

- $Q = \{q_0\} \cup Q_1$

定理6 (“正则语言类在星号运算下封闭”)

- $Q = \{q_0\} \cup Q_1$
- q_0

定理6 (“正则语言类在星号运算下封闭”)

- $Q = \{q_0\} \cup Q_1$
- q_0
- $\delta(q,a) = \begin{cases} \delta_1(q,a) & q \in Q_1, q \notin F_1 \\ \delta_1(q,a) & q \in F_1, a \neq \varepsilon \\ \delta_1(q,a) \cup \{q_1\} & q \in F_1, a = \varepsilon \\ \{q_1\} & q = q_0, a = \varepsilon \\ \emptyset & q = q_0, a \neq \varepsilon \end{cases}$

定理6（“正则语言类在星号运算下封闭”）

- $Q = \{q_0\} \cup Q_1$
- q_0
- $\delta(q,a) = \begin{cases} \delta_1(q,a) & q \in Q_1, q \notin F_1 \\ \delta_1(q,a) & q \in F_1, a \neq \varepsilon \\ \delta_1(q,a) \cup \{q_1\} & q \in F_1, a = \varepsilon \\ \{q_1\} & q = q_0, a = \varepsilon \\ \emptyset & q = q_0, a \neq \varepsilon \end{cases}$
- $F = \{q_0\} \cup F_2$

定义9 (正则表达式)

定义9 (正则表达式)

能够用正则运算符构造描述的语言的表达式

定义9 (正则表达式)

能够用正则运算符构造描述的语言的表达式

- $(0 \cup 1)0^*$:

定义9 (正则表达式)

能够用正则运算符构造描述的语言的表达式

- $(0\cup 1)0^*$：由0或1后面跟任意个0的所有字符串组成的语言

定义9 (正则表达式)

能够用正则运算符构造描述的语言的表达式

- $(0 \cup 1)0^*$：由0或1后面跟任意个0的所有字符串组成的语言
- $(0 \cup 1)^*$:

定义9 (正则表达式)

能够用正则运算符构造描述的语言的表达式

- $(0\cup 1)0^*$：由0或1后面跟任意个0的所有字符串组成的语言
- $(0\cup 1)^*$：由0,1的所有字符串组成的语言

定义9 (正则表达式)

能够用正则运算符构造描述的语言的表达式

- $(0 \cup 1)0^*$：由0或1后面跟任意个0的所有字符串组成的语言
- $(0 \cup 1)^*$：由0, 1的所有字符串组成的语言
- 设$\sum$是任意的字母表，则

定义9 (正则表达式)

能够用正则运算符构造描述的语言的表达式

- $(0\cup 1)0^*$：由0或1后面跟任意个0的所有字符串组成的语言
- $(0\cup 1)^*$：由0,1的所有字符串组成的语言
- 设$\sum$是任意的字母表，则
  - 正则表达式$\sum$表示该字母表上所有长度为*1*的字符串所组成的语言*;*

定义9 (正则表达式)

能够用正则运算符构造描述的语言的表达式

- $(0\cup 1)0^*$：由0或1后面跟任意个0的所有字符串组成的语言
- $(0\cup 1)^*$：由0, 1的所有字符串组成的语言
- 设$\sum$是任意的字母表，则
  - 正则表达式$\sum$表示该字母表上所有长度为1的字符串所组成的语言;
  - $\sum^*$表示该字母表上所有字符串组成的语言;

定义9 (正则表达式)

能够用正则运算符构造描述的语言的表达式

- $(0\cup 1)0^*$：由0或1后面跟任意个0的所有字符串组成的语言
- $(0\cup 1)^*$：由0, 1的所有字符串组成的语言
- 设$\sum$是任意的字母表，则
  - 正则表达式$\sum$表示该字母表上所有长度为*1*的字符串所组成的语言;
  - $\sum^*$表示该字母表上所有字符串组成的语言;
- 在正则表达式中，优先次序是：星号、连结、并

定义10 (正则表达式)

称R是一个正则表达式，如果R是

定义10 (正则表达式)

称R是一个正则表达式，如果R是

1. $\{a\}$(简记为a)，其中$a \in \sum$，即语言$\{a\}$;
2. ε，即语言$\{\varepsilon\}$;
3. $\emptyset$，即不包含任何字符串的语言-空语言;
4. $(R_1 \cup R_2)$;
5. $(R_1 \circ R_2)$;
6. (R_1^*);

其中R_1, R_2是正则表达式。

定义10 (正则表达式)

称R是一个正则表达式，如果R是

1. $\{a\}$(简记为a)，其中$a \in \sum$，即语言$\{a\}$;
2. ε，即语言$\{\varepsilon\}$;
3. $\emptyset$，即不包含任何字符串的语言-空语言;
4. $(R_1 \cup R_2)$;
5. $(R_1 \circ R_2)$;
6. (R_1^*);

其中R_1, R_2是正则表达式。

特别地，记$R^+ = RR^*$，即有一个或多个R中串连结构成的串。

例15 (设字母表$\sum = \{0,1\}$)

- $0^*10^* =$
- $\sum^* 1 \sum^* =$
- $(01^+)^* =$
- $(\sum\sum)^* =$
- $0\sum^* 0 \cup 1 \sum^* 1 \cup 0 \cup 1 =$
- $(0 \cup \varepsilon)(1 \cup \varepsilon) =$
- $1^*\emptyset =$
- $\emptyset^* =$

例15 (设字母表$\sum = \{0,1\}$)

- $0^*10^* = \{\omega|\omega$中恰有一个$1\}$
- $\sum^* 1 \sum^* =$
- $(01^+)^* =$
- $(\sum\sum)^* =$
- $0\sum^* 0 \cup 1 \sum^* 1 \cup 0 \cup 1 =$
- $(0\cup\varepsilon)(1\cup\varepsilon) =$
- $1^*\emptyset =$
- $\emptyset^* =$

例15 (设字母表$\sum = \{0,1\}$)

- $0^*10^* = \{\omega|\omega$中恰有一个1$\}$
- $\sum^* 1 \sum^* = \{\omega|\omega$中至少有一个1$\}$
- $(01^+)^* =$
- $(\sum\sum)^* =$
- $0\sum^* 0 \cup 1 \sum^* 1 \cup 0 \cup 1 =$
- $(0 \cup \varepsilon)(1 \cup \varepsilon) =$
- $1^*\emptyset =$
- $\emptyset^* =$

例15 (设字母表$\sum = \{0,1\}$)

- 0^*10^* $=\{\omega|\omega$中恰有一个1$\}$
- $\sum^* 1 \sum^*$ $=\{\omega|\omega$中至少有一个1$\}$
- $(01^+)^*$ $=\{\omega|\omega$中每个0后面至少跟一个1$\}$
- $(\sum\sum)^* =$
- $0\sum^* 0 \cup 1 \sum^* 1 \cup 0 \cup 1 =$
- $(0 \cup \varepsilon)(1 \cup \varepsilon) =$
- $1^*\emptyset =$
- $\emptyset^* =$

例15 (设字母表$\sum = \{0, 1\}$)

- 0^*10^* =$\{\omega|\omega$中恰有一个1$\}$
- $\sum^* 1 \sum^*$ =$\{\omega|\omega$中至少有一个1$\}$
- $(01^+)^*$ =$\{\omega|\omega$中每个0后面至少跟一个1$\}$
- $(\sum\sum)^*$ =$\{\omega|\omega$是长度为偶数的字符串$\}$
- $0\sum^* 0 \cup 1 \sum^* 1 \cup 0 \cup 1 =$
- $(0 \cup \varepsilon)(1 \cup \varepsilon) =$
- $1^*\emptyset =$
- $\emptyset^* =$

例15 (设字母表$\sum = \{0,1\}$)

- 0^*10^* =$\{\omega|\omega$中恰有一个1$\}$
- $\sum^* 1 \sum^*$ =$\{\omega|\omega$中至少有一个1$\}$
- $(01^+)^*$ =$\{\omega|\omega$中每个0后面至少跟一个1$\}$
- $(\sum\sum)^*$ =$\{\omega|\omega$是长度为偶数的字符串$\}$
- $0\sum^* 0 \cup 1 \sum^* 1 \cup 0 \cup 1$ =$\{\omega|\omega$中头尾符号相同$\}$
- $(0 \cup \varepsilon)(1 \cup \varepsilon) =$
- $1^*\emptyset =$
- $\emptyset^* =$

例15（设字母表$\sum = \{0,1\}$）

- 0^*10^* ={$\omega|\omega$中恰有一个1}
- $\sum^* 1 \sum^*$ ={$\omega|\omega$中至少有一个1}
- $(01^+)^*$ ={$\omega|\omega$中每个*0*后面至少跟一个1}
- $(\sum\sum)^*$ ={$\omega|\omega$是长度为偶数的字符串}
- $0\sum^* 0 \cup 1 \sum^* 1 \cup 0 \cup 1$ ={$\omega|\omega$中头尾符号相同}
- $(0\cup\varepsilon)(1\cup\varepsilon)$ =$\{\varepsilon, 0, 1, 01\}$
- $1^*\emptyset =$
- $\emptyset^* =$

例15 (设字母表$\sum = \{0, 1\}$)

- 0^*10^* =$\{\omega|\omega$中恰有一个1$\}$
- $\sum^* 1 \sum^*$ =$\{\omega|\omega$中至少有一个1$\}$
- $(01^+)^*$ =$\{\omega|\omega$中每个0后面至少跟一个1$\}$
- $(\sum\sum)^*$ =$\{\omega|\omega$是长度为偶数的字符串$\}$
- $0\sum^* 0 \cup 1 \sum^* 1 \cup 0 \cup 1$ =$\{\omega|\omega$中头尾符号相同$\}$
- $(0 \cup \varepsilon)(1 \cup \varepsilon)$ =$\{\varepsilon, 0, 1, 01\}$
- $1^*\emptyset$ =$\emptyset$
- $\emptyset^*$ =

例15 (设字母表$\sum = \{0,1\}$)

- 0^*10^* ={$\omega|\omega$中恰有一个1}
- $\sum^* 1 \sum^*$ ={$\omega|\omega$中至少有一个1}
- $(01^+)^*$ ={$\omega|\omega$中每个0后面至少跟一个1}
- $(\sum\sum)^*$ ={$\omega|\omega$是长度为偶数的字符串}
- $0\sum^* 0 \cup 1 \sum^* 1 \cup 0 \cup 1$ ={$\omega|\omega$中头尾符号相同}
- $(0 \cup \varepsilon)(1 \cup \varepsilon) = \{\varepsilon, 0, 1, 01\}$
- $1^*\emptyset = \emptyset$
- $\emptyset^* = \{\varepsilon\}$

定理7 (一个语言是正则的充要条件是其可以用正则表达式加以描述。)

定理7 (一个语言是正则的充要条件是其可以用正则表达式加以描述。)

例16 (将正则表达式$(a\cup b)^*aba$转化为NFA)

定理7 (一个语言是正则的充要条件是其可以用正则表达式加以描述。)

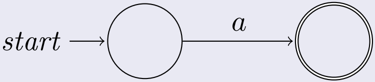
例16 (将正则表达式$(a \cup b)^* aba$转化为NFA)

- a

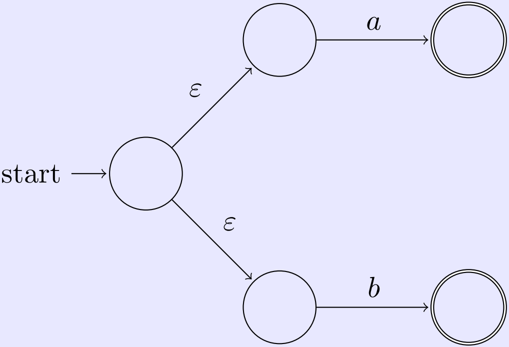
定理7 (一个语言是正则的充要条件是其可以用正则表达式加以描述。)

例16 (将正则表达式$(a \cup b)^*aba$转化为NFA)

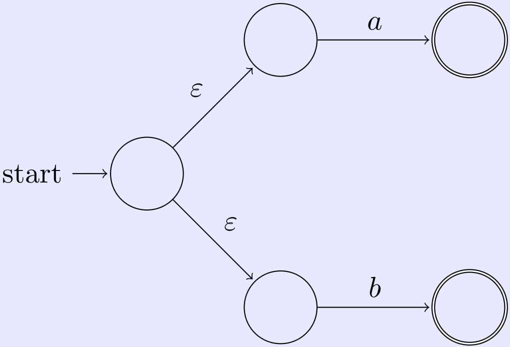
- a



- $a \cup b$

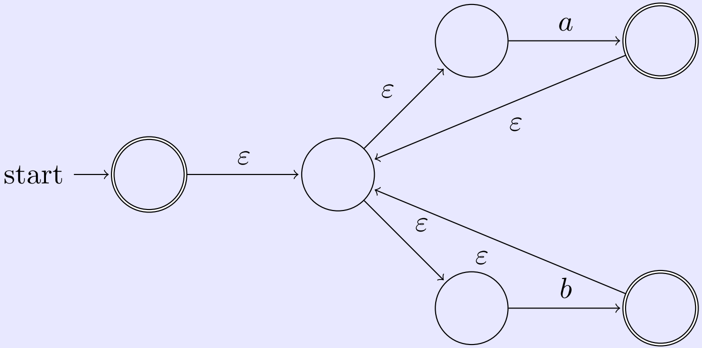


- $a \cup b$



- $(a \cup b)^*$

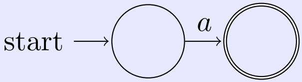
- $(a \cup b)^*$



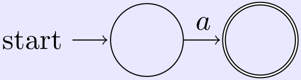
- a :

与有穷自动机的等价性

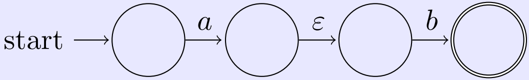
- a ：



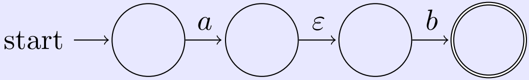
- ab :



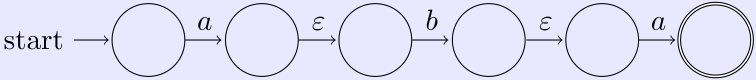
- ab ：



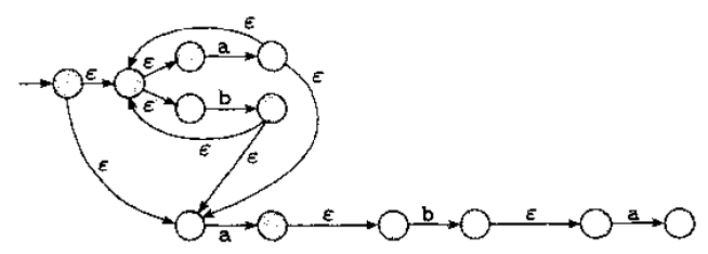
- aba:



- aba:



- $(a \cup b)^* aba$:



定义11 (广义非确定型有穷自动机(GNFA))

以任何正则表达式为标号的NFA被称为**广义非确定型有穷自动机(GNFA)**，即其标号不只是字母表中元素或ε。

定义11 (广义非确定型有穷自动机(GNFA))

以任何正则表达式为标号的NFA被称为**广义非确定型有穷自动机(GNFA)**，即其标号不只是字母表中元素或ε。

特殊形式的GNFA:

定义11 (广义非确定型有穷自动机(GNFA))

以任何正则表达式为标号的NFA被称为*广义非确定型有穷自动机(GNFA)*，即其标号不只是字母表中元素或ε。

特殊形式的GNFA:

- 起始状态有射到其他每个状态的箭头，而没有从其他状态射入的箭头；

定义11 (广义非确定型有穷自动机(GNFA))

以任何正则表达式为标号的NFA被称为**广义非确定型有穷自动机(GNFA)**，即其标号不只是字母表中元素或ε。

特殊形式的GNFA:

- 起始状态有射到其他每个状态的箭头，而没有从其他状态射入的箭头;
- 有唯一的接受状态，其有从每个状态射入的箭头，而没有射出的箭头，且接受状态与起始状态不同;

定义11 (广义非确定型有穷自动机(GNFA))

以任何正则表达式为标号的NFA被称为**广义非确定型有穷自动机(GNFA)**，即其标号不只是字母表中元素或ε。

特殊形式的GNFA:

- 起始状态有射到其他每个状态的箭头，而没有从其他状态射入的箭头；
- 有唯一的接受状态，其有从每个状态射入的箭头，而没有射出的箭头，且接受状态与起始状态不同；
- 除起始、接受状态外，每个状态到自身、到其他每个状态都有一个箭头；

DFA到(特殊形式)GNFA的转化:

DFA到(特殊形式)GNFA的转化：

- 添加一个新起始状态和接受状态，并使得

DFA到(特殊形式)GNFA的转化：

- 添加一个新起始状态和接受状态，并使得
  - 从新起始状态到每个原起始状态有一个ε箭头；

DFA到(特殊形式)GNFA的转化：

- 添加一个新起始状态和接受状态，并使得
  - 从新起始状态到每个原起始状态有一个ε箭头；
  - 从每个原接受状态到新接受状态有一个ε箭头；

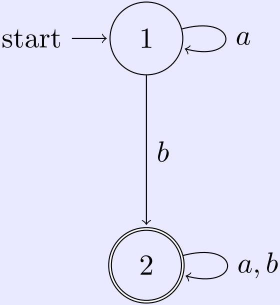
DFA到(特殊形式)GNFA的转化：

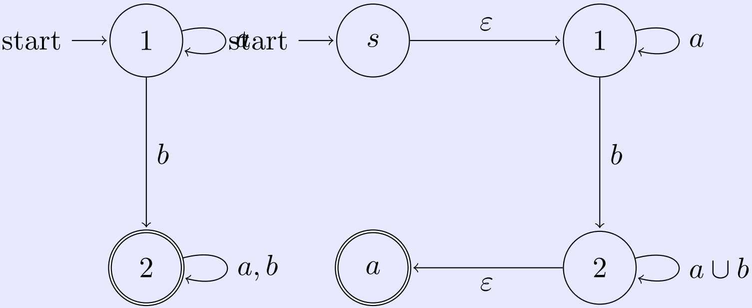
- 添加一个新起始状态和接受状态，并使得
  - 从新起始状态到每个原起始状态有一个ε箭头；
  - 从每个原接受状态到新接受状态有一个ε箭头；
- 如果两个状态间有多个方向相同的箭头，则将其替换为以原标记的并集为标记的箭头；

DFA到(特殊形式)GNFA的转化：

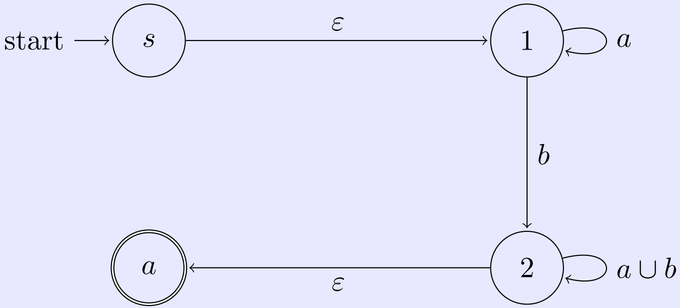
- 添加一个新起始状态和接受状态，并使得
  - 从新起始状态到每个原起始状态有一个ε箭头；
  - 从每个原接受状态到新接受状态有一个ε箭头；
- 如果两个状态间有多个方向相同的箭头，则将其替换为以原标记的并集为标记的箭头；
- 在没有箭头的状态之间添加标记为$\emptyset$的箭头；

与有穷自动机的等价性

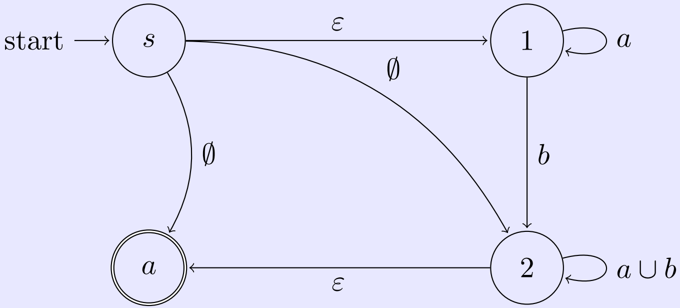


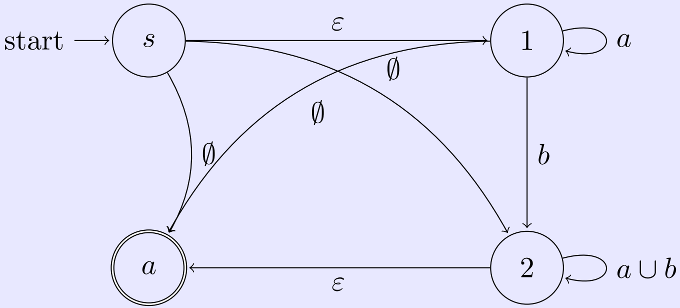

start
1
a
b
2
a, b
start
s
ε
1
a
b
2
a ∪ b
ε
a

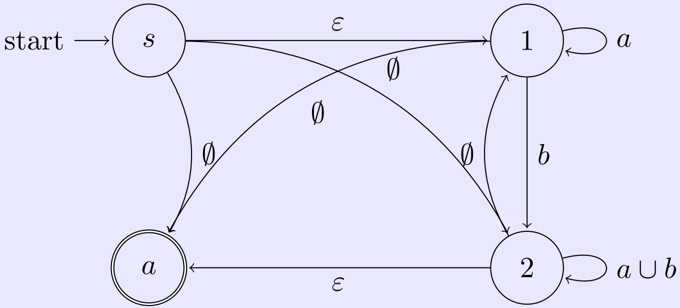
与有穷自动机的等价性



与有穷自动机的等价性


start
s
ε
$\emptyset$
1
a
$\emptyset$
b
a
2
ε
$a \cup b$


start
s
ε
1
a
$\emptyset$
$\emptyset$
$\emptyset$
b
a
2
$a \cup b$
ε


start
s
1
a
2
ε
∅
∅
∅
∅
a
b
a ∪ b
ε

非起始或接受的状态的删除：设被删除状态为q_{del}，则如果

非起始或接受的状态的删除：设被删除状态为q_{del}，则如果

- q_i到q_{del}有标记为R_1的箭头；

非起始或接受的状态的删除：设被删除状态为q_{del}，则如果

- q_i到q_{del}有标记为R_1的箭头；
- q_{del}有到自身且标记为R_2的箭头;

非起始或接受的状态的删除：设被删除状态为q_{del}，则如果

- q_i到q_{del}有标记为R_1的箭头；
- q_{del}有到自身且标记为R_2的箭头;
- q_{del}到q_j有标记为R_3的箭头;

非起始或接受的状态的删除：设被删除状态为q_{del}，则如果

- q_i到q_{del}有标记为R_1的箭头；
- q_{del}有到自身且标记为R_2的箭头;
- q_{del}到q_j有标记为R_3的箭头;
- q_i到q_j有标记为R_4的箭头；

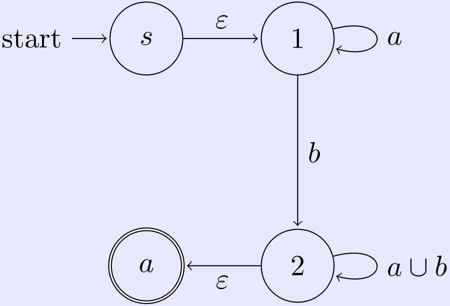
非起始或接受的状态的删除：设被删除状态为q_{del}，则如果

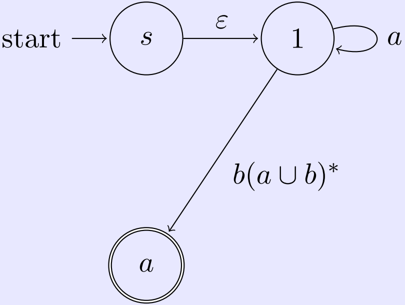
- q_i到q_{del}有标记为R_1的箭头；
- q_{del}有到自身且标记为R_2的箭头;
- q_{del}到q_j有标记为R_3的箭头;
- q_i到q_j有标记为R_4的箭头；

则q_i到q_j的箭头标记为

$$(R_1)(R_2)^*(R_3) \cup (R_4)$$

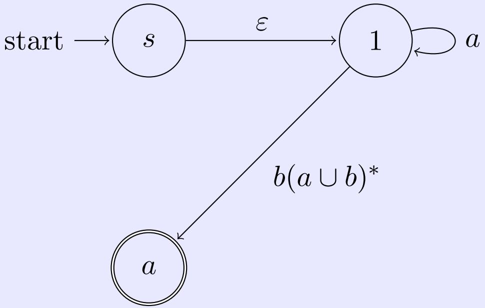
与有穷自动机的等价性



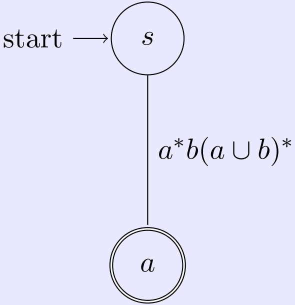

start
s
ε
1
a
b(a ∪ b)*
a

与有穷自动机的等价性

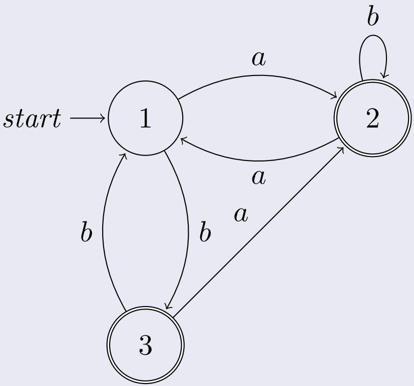
与有穷自动机的等价性



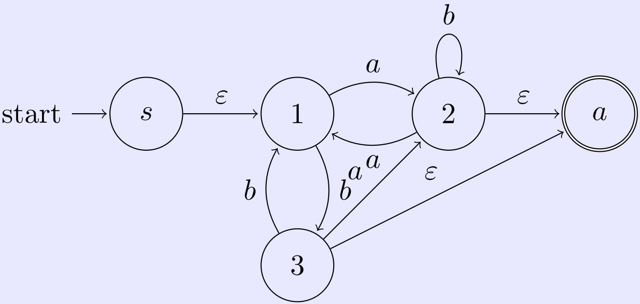
与有穷自动机的等价性

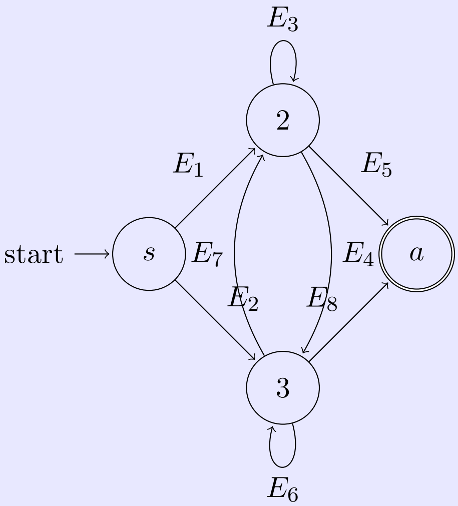


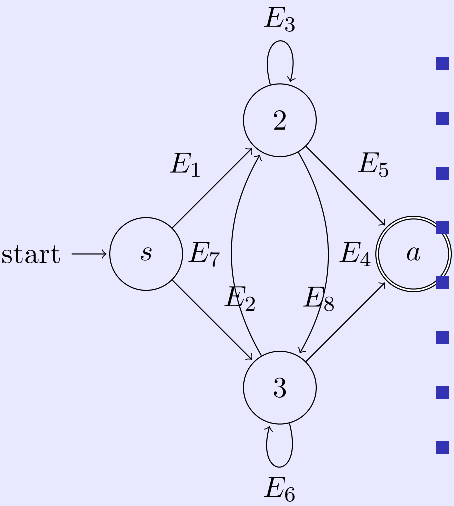
例17 (确定以下DFA对应的正则表达式)



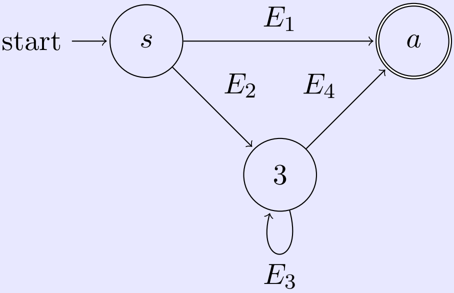
与有穷自动机的等价性

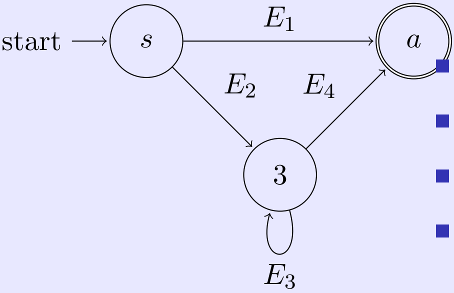



E_3
2
E_1
E_5
start
s
E_7
E_4
a
E_2
E_8
3
E_6



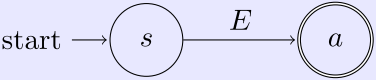
- $E_1 : a$
- $E_2 : b$
- $E_3 : aa \cup b$
- $E_4 : ab$
- $E_5 : \varepsilon$
- $E_6 : bb$
- $E_7 : ba \cup a$
- $E_8 : \varepsilon$


start
s
a
3
E_1
E_2
E_4
E_3

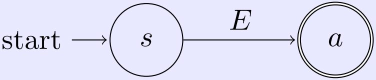


- $E_1 : a(aa \cup b)^*$
- $E_2 : a(aa \cup b)^* ab \cup b$
- $E_3 : (ba \cup a)(aa \cup b)^* ab \cup bb$
- $E_4 : (ba \cup a)(aa \cup b)^* \cup \varepsilon$

与有穷自动机的等价性



与有穷自动机的等价性



$$E:(a(aa\cup b)^*ab\cup b)((ba\cup a)(aa\cup b)^*ab\cup bb)^*((ba\cup a)(aa\cup b)^*\cup\epsilon)\cup(a(aa\cup b)^*)$$

定理8（泵引理）

定理8（泵引理）

若A是一个正则语言，则存在数p使得：如果s是A中任一长度不小于p的字符串，则s可以被分为3段，即$s = xyz$，满足

定理8 (泵引理)

若A是一个正则语言，则存在数p使得：如果s是A中任一长度不小于p的字符串，则s可以被分为3段，即$s = xyz$，满足

1. 对每个$i \geq 0$，有$xy^iz \in A$;

定理8 (泵引理)

若A是一个正则语言，则存在数p使得：如果s是A中任一长度不小于p的字符串，则s可以被分为3段，即$s = xyz$，满足

1. 对每个$i \geq 0$，有$xy^iz \in A$；
2. $|y| > 0$；

定理8 (泵引理)

若A是一个正则语言，则存在数p使得：如果s是A中任一长度不小于p的字符串，则s可以被分为3段，即$s = xyz$，满足

1. 对每个$i \geq 0$，有$xy^iz \in A$;
2. $|y| > 0$;
3. $|xy| \leq p$。

定理8（泵引理）

若A是一个正则语言，则存在数p使得：如果s是A中任一长度不小于p的字符串，则s可以被分为3段，即$s = xyz$，满足

1. 对每个$i \geq 0$，有$xy^iz \in A$；
2. $|y| > 0$；
3. $|xy| \leq p$。

称p为泵长度。

例18 (证明$B=\{0^n1^n|n\geq 0\}$不是正则语言)

证:

例18 (证明$B=\{0^n1^n|n\geq 0\}$不是正则语言)

证:　假设B是正则的，而p是其相应的泵长度。

例18 (证明$B=\{0^n1^n|n\geq 0\}$不是正则语言)

证: 假设B是正则的，而p是其相应的泵长度。考虑字符串$s=0^p1^p\in B$，而$s=xyz$为在泵引理下的分段，则

例18 (证明$B=\{0^n1^n|n\geq 0\}$不是正则语言)

证： 假设B是正则的，而p是其相应的泵长度。考虑字符串$s=0^p1^p\in B$，而$s=xyz$为在泵引理下的分段，则

- 若$y=0^t, 0<t<p$，则

例18 (证明$B=\{0^n1^n|n\geq 0\}$不是正则语言)

证：　假设B是正则的，而p是其相应的泵长度。考虑字符串$s=0^p1^p\in B$，而$s=xyz$为在泵引理下的分段，则

- 若$y=0^t,0<t<p$，则$xyyz\notin B$

例18 (证明$B=\{0^n1^n|n\geq 0\}$不是正则语言)

证： 假设B是正则的，而p是其相应的泵长度。考虑字符串$s=0^p1^p\in B$，而$s=xyz$为在泵引理下的分段，则

- 若$y=0^t,0<t<p$，则$xyyz\notin B$
- 若$y=1^t,0<t<p$，则

例18 (证明$B=\{0^n1^n|n\geq 0\}$不是正则语言)

证： 假设B是正则的，而p是其相应的泵长度。考虑字符串$s=0^p1^p\in B$，而$s=xyz$为在泵引理下的分段，则

- 若$y=0^t, 0<t<p$，则$xyyz\notin B$
- 若$y=1^t, 0<t<p$，则$xyyz\notin B$

例18 (证明$B=\{0^n1^n|n\geq 0\}$不是正则语言)

证： 假设B是正则的，而p是其相应的泵长度。考虑字符串$s=0^p1^p\in B$，而$s=xyz$为在泵引理下的分段，则

- 若$y=0^t,0<t<p$，则$xyyz\notin B$
- 若$y=1^t,0<t<p$，则$xyyz\notin B$
- 若$y=0^{t_1}1^{t_2},t_1,t_2>0$，则

例18 (证明$B=\{0^n1^n|n\geq 0\}$不是正则语言)

证：　假设B是正则的，而p是其相应的泵长度。考虑字符串$s=0^p1^p\in B$，而$s=xyz$为在泵引理下的分段，则

- 若$y=0^t, 0<t<p$，则$xyyz\notin B$
- 若$y=1^t, 0<t<p$，则$xyyz\notin B$
- 若$y=0^{t_1}1^{t_2}, t_1, t_2>0$，则$xyyz\notin B$

例19 (证明$B=\{w|w$中$0,1$的个数相同$\}$不是正则语言)

证:

例19 (证明$B=\{w|w$中$0,1$的个数相同$\}$不是正则语言)

证: 假设B是正则的，而p是其相应的泵长度。

例19 (证明$B=\{w|w$中$0,1$的个数相同$\}$不是正则语言)

证： 假设B是正则的，而p是其相应的泵长度。考虑字符串$s=0^p1^p\in B$，而$s=xyz$为在泵引理下的分段，则

例19 (证明$B=\{w|w$中$0,1$的个数相同$\}$不是正则语言)

证：　假设B是正则的，而p是其相应的泵长度。考虑字符串$s=0^p1^p\in B$，而$s=xyz$为在泵引理下的分段，则

- 由于$|xy|\leq p$，因此$y=0^t,0<t\leq p$。

例19 (证明$B = \{w|w$中0,1的个数相同$\}$不是正则语言)

证： 假设B是正则的，而p是其相应的泵长度。考虑字符串$s = 0^p1^p \in B$，而$s = xyz$为在泵引理下的分段，则

- 由于$|xy| \leq p$，因此$y = 0^t, 0 < t \leq p$。
- 而$xyyz \notin B$。

例20 (证明$C = \{ww|w \in \{0,1\}^*\}$不是正则语言)

证:

例21 (证明$D = \{0^i1^j|i > j\}$不是正则语言)

证:

例20 (证明$C = \{ww|w \in \{0,1\}^*\}$不是正则语言)

证： 假设C是正则的，而p是其相应的泵长度。

例21 (证明$D = \{0^i1^j|i > j\}$不是正则语言)

证：

例20 (证明$C = \{ww|w \in \{0,1\}^*\}$不是正则语言)

证： 假设C是正则的，而p是其相应的泵长度。考虑字符串$s = 0^p10^p1 \in C$，而$s = xyz$为在泵引理下的分段，则

例21 (证明$D = \{0^i1^j|i > j\}$不是正则语言)

证：

例20 (证明$C=\{ww|w\in\{0,1\}^*\}$不是正则语言)

证： 假设C是正则的，而p是其相应的泵长度。考虑字符串$s=0^p10^p1\in C$，而$s=xyz$为在泵引理下的分段，则

- 由于$|xy|\leq p, |y|>0$，则

例21 (证明$D=\{0^i1^j|i>j\}$不是正则语言)

证：

例20 (证明$C=\{ww|w\in\{0,1\}^*\}$不是正则语言)

证： 假设C是正则的，而p是其相应的泵长度。考虑字符串$s=0^p10^p1\in C$，而$s=xyz$为在泵引理下的分段，则

- 由于$|xy|\le p, |y|>0$，则$y=0^t, t\le p$

例21 (证明$D=\{0^i1^j|i>j\}$不是正则语言)

证：

例20 (证明$C = \{ww|w \in \{0,1\}^*\}$不是正则语言)

证：　假设C是正则的，而p是其相应的泵长度。考虑字符串$s = 0^p10^p1 \in C$，而$s = xyz$为在泵引理下的分段，则

- 由于$|xy| \le p, |y| > 0$，则$y = 0^t, t \le p$
- 由此可知$xyyz \notin C$

例21 (证明$D = \{0^i1^j|i > j\}$不是正则语言)

证：

例20 (证明$C = \{ww|w \in \{0,1\}^*\}$不是正则语言)

证：　假设C是正则的，而p是其相应的泵长度。考虑字符串$s = 0^p10^p1 \in C$，而$s = xyz$为在泵引理下的分段，则

- 由于$|xy| \le p, |y| > 0$，则$y = 0^t, t \le p$
- 由此可知$xyyz \notin C$

例21 (证明$D = \{0^i1^j|i > j\}$不是正则语言)

证：　假设D是正则的，而p是其相应的泵长度。

例20 (证明$C=\{ww|w\in\{0,1\}^*\}$不是正则语言)

证： 假设C是正则的，而p是其相应的泵长度。考虑字符串$s=0^p10^p1\in C$，而$s=xyz$为在泵引理下的分段，则

- 由于$|xy|\le p,|y|>0$，则$y=0^t,t\le p$
- 由此可知$xyyz\notin C$

例21 (证明$D=\{0^i1^j|i>j\}$不是正则语言)

证： 假设D是正则的，而p是其相应的泵长度。考虑字符串$s=0^{p+1}1^p\in D$，而$s=xyz$为在泵引理下的分段，则

例20 (证明$C = \{ww|w \in \{0,1\}^*\}$不是正则语言)

证： 假设C是正则的，而p是其相应的泵长度。考虑字符串$s = 0^p10^p1 \in C$，而$s = xyz$为在泵引理下的分段，则

- 由于$|xy| \leq p, |y| > 0$，则$y = 0^t, t \leq p$
- 由此可知$xyyz \notin C$

例21 (证明$D = \{0^i1^j|i > j\}$不是正则语言)

证： 假设D是正则的，而p是其相应的泵长度。考虑字符串$s = 0^{p+1}1^p \in D$，而$s = xyz$为在泵引理下的分段，则

- 由于$|xy| \leq p, |y| > 0$，则$xyyx \overset{?}{\notin} D$

例20 (证明$C = \{ww|w \in \{0,1\}^*\}$不是正则语言)

证： 假设C是正则的，而p是其相应的泵长度。考虑字符串$s = 0^p10^p1 \in C$，而$s = xyz$为在泵引理下的分段，则

- 由于$|xy| \le p, |y| > 0$，则$y = 0^t, t \le p$
- 由此可知$xyyz \notin C$

例21 (证明$D = \{0^i1^j|i > j\}$不是正则语言)

证： 假设D是正则的，而p是其相应的泵长度。考虑字符串$s = 0^{p+1}1^p \in D$，而$s = xyz$为在泵引理下的分段，则

- 由于$|xy| \le p, |y| > 0$，则$y = 0^t, 0 < t \le p$

例20 (证明$C = \{ww | w \in \{0,1\}^*\}$不是正则语言)

证： 假设C是正则的，而p是其相应的泵长度。考虑字符串$s = 0^p10^p1 \in C$，而$s = xyz$为在泵引理下的分段，则

- 由于$|xy| \leq p, |y| > 0$，则$y = 0^t, t \leq p$
- 由此可知$xyyz \notin C$

例21 (证明$D = \{0^i1^j | i > j\}$不是正则语言)

证： 假设D是正则的，而p是其相应的泵长度。考虑字符串$s = 0^{p+1}1^p \in D$，而$s = xyz$为在泵引理下的分段，则

- 由于$|xy| \leq p, |y| > 0$，则$y = 0^t, 0 < t \leq p$
- 由此可知$xy^0z = xz \notin D$