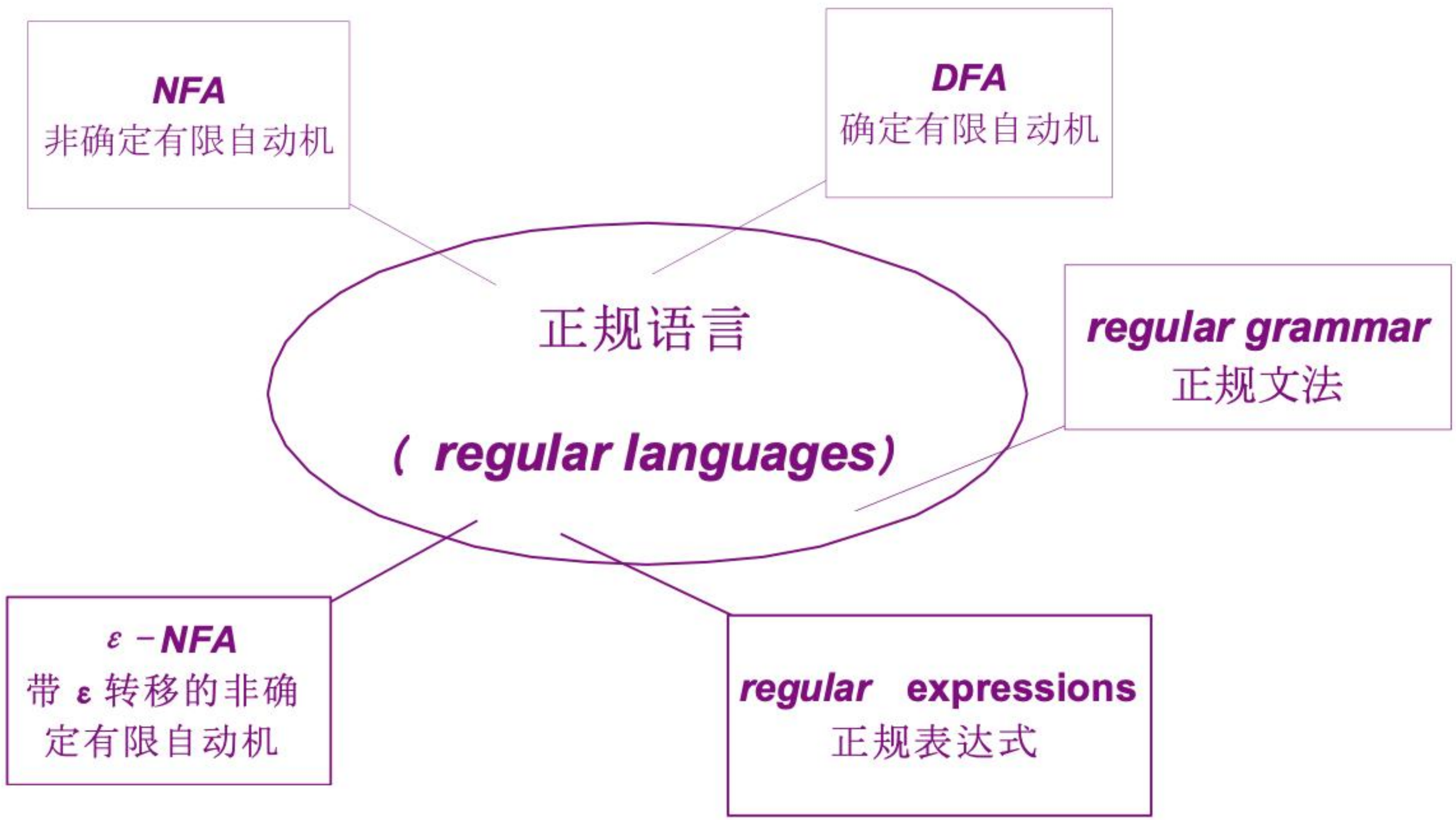
✧ 正规表达式与正规语言

正规语言的不同表达形式



正规表达式与正规语言

- ✧ 正规表达式
- ✧ 正规语言
- ✧ 正规表达式的代数性质

正规表达式

- 用代数的方法表示正规语言
- 语义 正规语言（Regular Languages，RL）

  作用于正规语言上的三种代数运算:

  – 联合（union） $L \cup M = \{w \mid w \in L \vee w \in M\}$
  – 连接（concatenation） $L \cdot M = \{w_1 w_2 \mid w_1 \in L \wedge w_2 \in M\}$
  – （星）闭包（closure） $L^* = \cup_{i \geq 0} L^i$
- 语法
  – 基本正规表达式 3 个运算符 vs.上述3 个运算
  – 对应不同应用形式会扩展一些助记运算符

    如 LEX 中的正规表达式

例1：存在两个正规语言

$$L = \{001, 10, 111\} \qquad M = \{\epsilon, 001\}$$

计算 $L \cup M$ 、LM 和 M^3 。

解： $L \cup M = \{\epsilon, 001, 10, 111\}$

$$LM = \{001, 001001, 10, 10001, 111, 111001\}$$

$$M^2 = MM = \{\epsilon, 001, 001001\}$$

$$M^3 = MM^2 = \{\epsilon, 001, 001001, 001001001\}$$

正规表达式

✧ 语法

– 设 Σ 为字母表。 Σ 上的正规表达式集合 R 递归定义如下:

基础. 1. ε , ϕ ∈ R.

2. If a ∈Σ , then a ∈ R.

3. 任一变量 L ∈ R.

归纳.

1. If E ∈ R and F ∈ R , then E + F ∈ R .
2. If E ∈ R and F ∈ R , then EF ∈ R.
3. If E ∈ R , then E* ∈ R.
4. If E ∈ R , then (E) ∈ R.

正规表达式

✧ 语义

– 设 R 为 Σ 上的正规表达式集合。对每个不含变量的 E∈R， E的语言 L(E) 递归定义如下：

基础.

1. $L(\varepsilon)=\{\varepsilon\}$ and $L(\phi)=\phi$.
2. If $a\in\Sigma$ ， then $L(a)=\{a\}$.

归纳.

1. If $E\in R$ and $F\in R$， then $L(E+F) = L(E)\cup L(F)$.
2. If $E\in R$ and $F\in R$， then $L(EF) = L(E)L(F)$.
3. If $E\in R$， then $L(E^*) = (L(E))^*$.
4. If $E\in R$， then $L((E)) = L(E)$.

✧ 正规表达式算符优先级

算符优先级（*precedence*）依次为

- *
- • 连接
- +

$$\mathbf{01}^* + \mathbf{1} \text{ 等价于 } (\mathbf{0}(\mathbf{1}^*)) + \mathbf{1}$$

✧ 正规表达式的几个派生运算符

- $L^+ = LL^* = L^*L$
- $L? = \varepsilon + L$
- $L^n = LL^{n-1} \quad (n>0)$
  $L^0 = \varepsilon$

✧ 正规语言（*regular language*）

– 归纳定义

字母表 Σ 上的正规语言归纳定义如下：

基础 1 $\{\varepsilon\}$ 和 ϕ 是正规语言

2 若 $a \in \Sigma$，则 $\{a\}$ 是正规语言

归纳 1 若 L 和 R 是正规语言，则 $L \cup R$ 是正规语言

2 若 L 和 R 是正规语言，则 LR 是正规语言

3 若 L 是正规语言，则 L^* 是正规语言

正规语言

- 正规语言（*regular language*）
  - 利用正规表达式定义

    对于字母表 ∑ 上的语言 ***R***，若存在 ∑ 上的正规表达式 ***E***，满足 ***L(E) = R***，则 ***R*** 是正规语言

正规表达式的代数定律

- 正规表达式的代数定律
  - 交换律和结合律
  - 零元和幺元
  - 分配律
  - 等幂律
  - 与闭包相关的定律
- 代数定律的具体化
  - 用于发现和测试定律

正视表达式的代数定律

- 交换律（*commutativity*）和结合律 （*associativeity*）
  - $L+M = M+L$
  - $(L+M)+N = L+(M+N)$
  - $(LM)N = L(MN)$
- 幺元（*identities*）和零元（*annihilators*）
  - $\phi + L = L + \phi = L$
  - $\varepsilon L = L\varepsilon = L$
  - $\phi L = L\phi = \phi$

正规表达式的代数定律

- 分配律（*distributive law*）
  - $L(M+N) = LM+LN$
  - $(M+N)L = ML+NL$
- 等幂律（*idempotent law*）
  - $L + L = L$

正规表达式的代数定律

例2：对于任意3个正规表达式E_1、E_2和E_3，存在

$$E_1(E_2+E_3)=E_1E_2+E_1E_3$$

证明：问题可以转化为证明以下命题

$$w\in L(E_1(E_2+E_3))\text{当且仅当}w\in L(E_1E_2+E_1E_3)$$

（仅当）如果$w\in L(E_1(E_2+E_3))$，则可将w表示为$w=xy$的形式，其中$x\in L(E_1)$且$y\in L(E_2+E_3)$。分两种情况：

第一种情况是$y\in L(E_2)$，则$xy\in L(E_1E_2)$。因此可得到$w=xy\in L(E_1E_2)\cup L(E_1E_3)=L(E_1E_2+E_1E_3)$。

第二种情况是$y\in L(E_3)$，则$xy\in L(E_1E_3)$。因此可得到$w=xy\in L(E_1E_2)\cup L(E_1E_3)=L(E_1E_2+E_1E_3)$。

正规表达式的代数定律

例2：对于任意3个正规表达式E_1、E_2和E_3，存在

$$E_1(E_2+E_3)=E_1E_2+E_1E_3$$

证明：问题可以转化为证明以下命题

$$w\in L(E_1(E_2+E_3))\text{当且仅当}w\in L(E_1E_2+E_1E_3)$$

（当）如果$w\in L(E_1E_2+E_1E_3)$，则有$w\in L(E_1E_2)$或者$w\in L(E_1E_3)$。可以分为两种情况。

如果$w\in L(E_1E_2)$，则w可以写成$w=xy$的形式，其中$x\in L(E_1)$且$y\in L(E_2)$。由于$y\in L(E_2)$，则可以得到$y\in L(E_2+E_3)$。因此，$w=xy\in L(E_1(E_2+E_3))$。

如果$w\in L(E_1E_3)$，可以类似地处理。

正视表达式的代数定律

- 与闭包相关的定律
  - $(L^*)^* = L^*$
  - $\phi^* = \varepsilon$
  - $\varepsilon^* = \varepsilon$
  - $L^+ = LL^* = L^*L$　（L^+的定义）
  - $L^* = L^+ + \varepsilon$
- 与任选运算相关的定律
  - $L? = \varepsilon + L$　（$L?$的定义）

例3：设计正规表达式 E 使得

(1) $L(E) = \{1\}$

(2) $L(E) = \{0\}$

(3) $L(E) = \{0, 1\}$

(4) $L(E) = \{0, 1\}^*$

解：(1) $E = \mathbf{1}$

(2) $E = \mathbf{0}$

(3) $E = \mathbf{0} + \mathbf{1}$

(4) $E = (\mathbf{0} + \mathbf{1})^*$

正规表达式

例4：设计正规表达式 E 使得

$$L(E) = \{w | w\text{长度为2且第2个字符是1}\}$$

该语言的字母表包含0和1。

解：可以分成两种情况。第一种情况是首字符是0：

$$\mathbf{01}$$

第二种情况是首字符是1：

$$\mathbf{11}$$

合并两种情况可以得到：

$$\mathbf{01 + 11}$$

根据分配律：

$$(\mathbf{0 + 1})\mathbf{1}$$

例5：设计正规表达式 E 使得

$$L(E) = \{w | w\text{至少包含3个连续的0}\}$$

该语言的字母表包含0和1。

解：这道题要求至少包含3个连续的0，所以可以先写出来

$$\mathbf{000}$$

由于左右可以是任意字符串，即任意长度、任意字符，因此可以利用星闭包来表示左右两侧。最终结果是：

$$(\mathbf{0} + \mathbf{1})^*\mathbf{000}(\mathbf{0} + \mathbf{1})^*$$

例6：设计正规表达式 E 使得

$$L(E) = \{w|w\text{以01结尾}\}$$

该语言的字母表包含0和1。

解： 根据例5的技巧，先给出结尾的01，再补上左侧的任意字符串，因此结果为：

$$E = (\mathbf{0} + \mathbf{1})^*\mathbf{01}$$

例7：设计正规表达式 E 使得

$$L(E)=\{w|w\text{以1开头且以0结尾}\}$$

该语言的字母表包含0和1。

解：根据例5和例6的经验，这道题就比较简单了，除了开头和结尾以外的内部为任意字符串。

$$E=\mathbf{1}(\mathbf{0}+\mathbf{1})^{*}\mathbf{0}$$

例8：设计正规表达式 E 使得

$$L(E)=\{w|w\text{的开头字符和结尾字符相同}\}$$

该语言的字母表包含0和1。

解：这道题是例7的扩展。区分三种情况，第一种情况是开头字符和结尾字符都是0：

$$\mathbf{0}(\mathbf{0}+\mathbf{1})^*\mathbf{0}$$

第二种情况是开头字符和结尾字符都是1：

$$\mathbf{1}(\mathbf{0}+\mathbf{1})^*\mathbf{1}$$

第三种情况是字符串的长度为1。因此结果为：

$$\mathbf{0}(\mathbf{0}+\mathbf{1})^*\mathbf{0}+\mathbf{1}(\mathbf{0}+\mathbf{1})^*\mathbf{1}+\mathbf{0}+\mathbf{1}$$

例9：设计正规表达式 E 使得

$$L(E) = \{w|w\text{由交替的0和1组成}\}$$

该语言的字母表包含0和1。

解：这道题一个直观的想法是 $E = (\mathbf{01})^*$，但是这样仅包含了以0开头、以1结尾的情况。需要考虑四种情况：

(1) 以0开头、以1结尾：$E_1 = (\mathbf{01})^*$

(2) 以1开头、以0结尾：$E_2 = (\mathbf{10})^*$

(3) 以0开头、以0结尾：$E_3 = \mathbf{0}(\mathbf{10})^*$

(4) 以1开头、以1结尾：$E_4 = \mathbf{1}(\mathbf{01})^*$

正规表达式

例9：设计正规表达式 E 使得

$$L(E) = \{w | w\text{由交替的0和1组成}\}$$

该语言的字母表包含0和1。

解：合并这四种情况可以得到

$$E = (\mathbf{01})^* + (\mathbf{10})^* + \mathbf{0}(\mathbf{10})^* + \mathbf{1}(\mathbf{01})^*$$

上面的式子是否可以写成更紧凑的形式呢？可以利用分配律来实现。

例9：设计正规表达式 E 使得

$$L(E) = \{w|w\text{由交替的0和1组成}\}$$

该语言的字母表包含0和1。

解： 利用分配律来处理

$$
\begin{aligned}
& (\mathbf{01})^* + (\mathbf{10})^* + \mathbf{0}(\mathbf{10})^* + \mathbf{1}(\mathbf{01})^* \\
&= (\epsilon + \mathbf{1})(\mathbf{01})^* + (\mathbf{10})^* + \mathbf{0}(\mathbf{10})^* \\
&= (\epsilon + \mathbf{1})(\mathbf{01})^* + (\mathbf{10})^* + (\mathbf{01})^*\mathbf{0} \\
&= (\epsilon + \mathbf{1})(\mathbf{01})^*(\epsilon + \mathbf{0}) + (\mathbf{10})^* \\
&= (\epsilon + \mathbf{1})(\mathbf{01})^*(\epsilon + \mathbf{0}) + \epsilon + \mathbf{1}(\mathbf{01})^*\mathbf{0} \\
&= (\epsilon + \mathbf{1})(\mathbf{01})^*(\epsilon + \mathbf{0})
\end{aligned}
$$

例9：设计正规表达式 E 使得

$$L(E) = \{w | w\text{由交替的0和1组成}\}$$

该语言的字母表包含0和1。

解：还有另一种处理方式

$$\begin{aligned}
&\mathbf{(01)^* + (10)^* + 0(10)^* + 1(01)^*} \\
= &(\epsilon + \mathbf{0})(\mathbf{10})^* + (\mathbf{01})^* + \mathbf{1}(\mathbf{01})^* \\
= &(\epsilon + \mathbf{0})(\mathbf{10})^* + (\mathbf{01})^* + (\mathbf{10})^*\mathbf{1} \\
= &(\epsilon + \mathbf{0})(\mathbf{10})^*(\epsilon + \mathbf{1}) + (\mathbf{01})^* \\
= &(\epsilon + \mathbf{0})(\mathbf{10})^*(\epsilon + \mathbf{1}) + \epsilon + \mathbf{0}(\mathbf{10})^*\mathbf{1} \\
= &(\epsilon + \mathbf{0})(\mathbf{10})^*(\epsilon + \mathbf{1})
\end{aligned}$$

例10：设计正规表达式 E 使得

$$L(E)=\{w|w\text{从右端数第5个位置是1}\}$$

该语言的字母表包含0和1。

解：倒数第5个位置是1，那么倒数前4个位置必然或者为0，或者为1。前面的字串则可以是任意字符串。因此：

$$E=(\mathbf{0}+\mathbf{1})^*\mathbf{1}(\mathbf{0}+\mathbf{1})^4$$

正规表达式

例11：设计正规表达式 E 使得

$$L(E) = \{w|w\text{的前5位至少包含一个}1 \wedge |w| \geq 1\}$$

该语言的字母表包含0和1。

解：这道题的难点在于如何是实现前5位至少包含一个字符。

注意：字符串的长度并不一定超过5。此处的技巧是巧妙地使用空串：

$$E = (\mathbf{0} + \mathbf{1} + \epsilon)^4\mathbf{1}(\mathbf{0} + \mathbf{1})^*$$

$(0+\varepsilon)^4 \, 1 \, (0+1)^*$

例12：设计正规表达式 E 使得

$$L(E) = \{w|w\text{的后5位至少包含一个}1 \wedge |w| \geq 1\}$$

该语言的字母表包含0和1。

解：根据例10的经验，这道题就比较简单了

$$E = (\mathbf{0} + \mathbf{1})^*\mathbf{1}(\mathbf{0} + \mathbf{1} + \epsilon)^4$$

例13：设计正规表达式 E 使得

$$L(E)=\{w|w\text{中至少包含一个}a\text{和一个}b\}$$

该语言的字母表包含 a、b和c。

解：由于w中至少包含一个a和b，不妨就考虑从左往右数的第一个a和第一个b。那么可以分为两种情况。

第一种情况是第一个a在第一个b之前。那么，在第一个a之前必然只可能出现c，在第一个a和第一个b之间必然只可能出现a和c，在第一个b之后可以出现任意字符串。因此，可以写成

$$\mathbf{c}^*\mathbf{a}(\mathbf{a}+\mathbf{c})^*\mathbf{b}(\mathbf{a}+\mathbf{b}+\mathbf{c})^*$$

例13：设计正规表达式 E 使得

$$L(E)=\{w|w\text{中至少包含一个}a\text{和一个}b\}$$

该语言的字母表包含 a、b和c。

解：第二种情况是第一个b在第一个a之前，可以类似处理

$$\mathbf{c^*b(b+c)^*a(a+b+c)^*}$$

因此，最终的结果是

$$E=\mathbf{c^*a(a+c)^*b(a+b+c)^*}+$$
$$\mathbf{c^*b(b+c)^*a(a+b+c)^*}$$

例14：设计正规表达式 E 使得

$$L(E)=\{w|w\text{中每对相邻的0都出现在每对相邻的1之前}\}$$

该语言的字母表包含0和1。

解：这道题看似无从下手。技巧是把字符串分成两部分，左部允许0相邻但不允许1相邻，右部允许1相邻但不允许0相邻，这样自然就实现了。

首先看允许0相邻但不允许1相邻，直接的写法是

$$(\mathbf{10}+\mathbf{0})^*$$

但是没有考虑以1结尾的情况，因此可以写成

$$(\mathbf{10}+\mathbf{0})^*(\epsilon+\mathbf{1})$$

例14：设计正规表达式 E 使得

$$L(E) = \{w|w\text{中每对相邻的0都出现在每对相邻的1之前}\}$$

该语言的字母表包含0和1。

解：再来看右部，允许1相邻但不允许0相邻

$$(\mathbf{01}+\mathbf{1})^*(\epsilon+\mathbf{0})$$

把两部分拼接起来可得到：

$$E = (\mathbf{10}+\mathbf{0})^*\cancel{(\epsilon+\mathbf{1})}(\mathbf{01}+\mathbf{1})^*(\epsilon+\mathbf{0})$$

例15：设计正规表达式 E 使得

$$L(E)=\{w|w=xy\wedge x,y\in\{0,1\}^*\wedge|y|=3\wedge y=y^R\}$$

解：一般而言正规表达式是没法处理回文的，但是这道题中的回文长度受限，因此是可以全枚举的：

$$E=(\mathbf{0}+\mathbf{1})^*(\mathbf{000}+\mathbf{010}+\mathbf{111}+\mathbf{101})$$

例16：设计正规表达式 E 使得

$$L(E)=\{w|w\text{中既不包含子串00，也不包含子串11}\}$$

该语言的字母表包含0和1。

解：这道题实际上和例9是等价的。要注意同一问题的不同表述。

$$E=(\epsilon+\mathbf{1})(\mathbf{01})^*(\epsilon+\mathbf{0})$$

正规表达式

例17：设计正规表达式 E 使得

$$L(E) = \{0^m 1^n | m \geq 0 \wedge n \geq 0 \wedge m + n \text{是偶数}\}$$

该语言的字母表包含0和1。

解： $m+n$是偶数，只有两种情况。第一种情况是m和n都是偶数：

$$(\mathbf{00})^*(\mathbf{11})^*$$

第二种情况是m和n都是奇数：

$$(\mathbf{00})^*\mathbf{01}(\mathbf{11})^*$$

合并两种情况可以得到：

$$(\mathbf{00})^*(\mathbf{01}+\epsilon)(\mathbf{11})^*$$

例18：设计正规表达式 E 使得

$$L(E) = \{w|\text{当}w\text{以0结尾时，其长度是奇数} \wedge |w| \geq 1\}$$

该语言的字母表包含0和1。

解：需要考虑两种情况。第一种情况是以0结尾：

$$((\mathbf{0}+\mathbf{1})^2)^*\mathbf{0}$$

第二种情况是以1结尾：

$$(\mathbf{0}+\mathbf{1})^*\mathbf{1}$$

因此，最终结果是

$$((\mathbf{0}+\mathbf{1})^2)^*\mathbf{0}+(\mathbf{0}+\mathbf{1})^*\mathbf{1}$$

例19：设计正规表达式 E 使得

$$L(E) = \{w|w\text{的前5位至少有一个子串00} \wedge |w| \geq 2\}$$

该语言的字母表包含0和1。

解：这是例11的一个扩展，技巧类似：

$$(\mathbf{0}+\mathbf{1}+\epsilon)^3\mathbf{00}(\mathbf{0}+\mathbf{1})^*$$

例20：设计正规表达式 E 使得

$$L(E) = \{w | w\text{的第2位至第5位至少有一个1} \wedge |w| \geq 2\}$$

该语言的字母表包含0和1。

解：对于并不是顺数或者倒数第几位的情况，需要加上位置的偏移：

$$E = (\mathbf{0} + \mathbf{1})(\mathbf{0} \cancel{+ \mathbf{1}} + \epsilon)^3 \mathbf{1}(\mathbf{0} + \mathbf{1})^*$$

正规表达式

例21：设计正规表达式 E 使得

$$L(E) = \{w|w\text{至少包含3个1，且倒数第3位为1}\}$$

该语言的字母表包含0和1。

解：像这样的题目，要分情况讨论。这里需要考虑三种情况。

第一种情况是剩下两个1都在倒数第3位左侧：

$$(\mathbf{0}+\mathbf{1})^*\mathbf{1}(\mathbf{0}+\mathbf{1})^*\mathbf{1}(\mathbf{0}+\mathbf{1})^*\mathbf{100}$$

第二种情况是剩下两个1分别在倒数第3位左侧和右侧：

$$(\mathbf{0}+\mathbf{1})^*\mathbf{1}(\mathbf{0}+\mathbf{1})^*\mathbf{1}(\mathbf{01}+\mathbf{10})$$

第三种情况是剩下两个1都在倒数第3位右侧：

$$(\mathbf{0}+\mathbf{1})^*\mathbf{111}$$

正规表达式

例21：设计正规表达式 E 使得

$$L(E) = \{w|w\text{至少包含3个1，且倒数第3位为1}\}$$

该语言的字母表包含0和1。

解：因此，最终的结果是

$$E = (\mathbf{0}+\mathbf{1})^*\mathbf{1}(\mathbf{0}+\mathbf{1})^*\mathbf{1}(\mathbf{0}+\mathbf{1})^*\mathbf{100}+$$
$$(\mathbf{0}+\mathbf{1})^*\mathbf{1}(\mathbf{0}+\mathbf{1})^*\mathbf{1}(\mathbf{01}+\mathbf{10})\ +$$
$$(\mathbf{0}+\mathbf{1})^*\mathbf{111}$$

正规表达式的代数定律

✧ 代数定律的具体化

- 具体化：将正规表达式中的每个变量用单个符号替换.
- 一般化：将具体表达式中的单个符号用变量表示.
- 结论：正规表达式的一般形式所代表的任何语言与其对应的具体表达式的语言之间可以建立特定的对应关系.
- 应用

  用于发现和测试关于正规表达式的定律

正规表达式的代数定律

✧ 代数定律的具体化

– 定理：正规表达式的一般形式所代表的任何语言与其对应的具体表达式的语言之间存在如下对应关系:

设 E 为正规表达式，$L_1, L_2, \ldots, L_m$为其中的变量.（这里，假设E中不含非变量符号，否则需推广）将每一L_i替换为符号a_i，得到对应E的一个具体表达式 C. 则对这些变量的任何实例语言 $S_1, S_2, \ldots, S_m$，$L(E)$中的任何串 w 可写成$w = w_1w_2\ldots w_k$ 的形式，其中w_i是某一语言 S_{ji} $(1 \le j_i \le m)$ 中的串，并且串 $a_{j1}a_{j2}\ldots a_{jk}$ 属于语言 $L(C)$; 另一方面, 若串 $a_{j1}a_{j2}\ldots a_{jk}$ 属于语言$L(C)$, w_i 是某一语言S_{ji} $(1 \le j_i \le m)$ 中的任意串，则$w=w_1w_2\ldots w_k$ 属于语言$L(E)$

正规表达式的代数定律

✧ 代数定律的具体化

– 举例: 正规表达式 S^*M 对应的一个具体表达式为 a^*b . 任取S和M的一个实例，比如设

$S=\{01,10\}$, $M=L(2^*)$. 则有:

任一 $w \in L(S^*M)=\{01,10\}^* L(2^*)$, 可以写成 $w_1w_2\ldots w_k$ 的形式，w_i 是S或M中的串，且有 $c_1c_2\ldots c_k \in L(a^*b)$ (另一方面类似).

其中，若 w_i 是 S中的串，则有$c_i = a$ ，否则 $c_i = b$.

（注：默认的字母表包含了所涉及到的所有非变量符号。前述定理和后续证明皆视如此。）

正规表达式的代数定律

✧ 代数定律的具体化

– （上述定理的）**证明思路:** （选讲）
归纳于正规表达式 *E* 的结构. (仅证一方面)

基础：若 *E* 为ε, ϕ, *a*，显然有 *E* = *C*, 定理成立;

若 *E* 为*L*，将唯一的变量 *L* 替换为符号 *c*，则其具体表达式为 c. *L*的任何一个实例语言中的串*w*，对应表达式 c 的语言*L*(c)中的串*c*.

（接下页）

正视表达式的代数定律

✧ 代数定律的具体化 （接上页证明）

归纳：若 $E=E_1E_2$，E_1中的变量为 $L_1, L_2, \ldots, L_m$，E_2中的变量为 $L_1', L_2', \ldots, L_n'$，可能有交叉. 分别用$a_1, a_2, \ldots, a_m, a_1', a_2', \ldots, a_n'$替换它们（也可能有交叉），则$E$具体化为 C，E_1和E_2分别具体化为C_1和C_2，并且$C = C_1C_2$.

任意取定上述各变量的实例语言. 设任何$w \in L(E)$，则存在$w_1 \in L(E_1)$ 和$w_2 \in L(E_2)$，且满足$w = w_1w_2$. 由归纳假设，w_1可写成 $s_1s_2\ldots s_k$ 的形式，其中s_i是某一语言S_{j_i} $(1 \le j_i \le m)$中的串，并且$a_{j1}a_{j2}\ldots a_{jk}$属于语言$L(C_1)$；同样，$w_2$可写成$t_1t_2\ldots t_h$ 的形式，其中t_i是某一语言S_{l_i}' $(1 \le l_i \le n)$中的串，并且$a_{l1}' a_{l2}' \ldots a_{lh}'$属于语言$L(C_2)$. 这样，$w$可写成 $w = s_1s_2\ldots s_k t_1t_2\ldots t_h$ 的形式，并且有$a_{j1}a_{j2}\ldots a_{jk}\, a_{l1}' a_{l2}' \ldots a_{lh}'$属于语言$L(C)$.

对于$E=E_1+E_2$和 $E=E_1^*$ 的情形，可以类似证明.

正规表达式的代数定律

✧ 代数定律的具体化

– 推论: 设 E, F 为正规表达式，它们具有相同的变量集；采用同样的替换方式，得到对应于 E, F 的具体表达式分别为 C, D. 则对 E, F 中的变量对应的所有语言，满足

$$L(E) = L(F) \quad \text{iff} \quad L(C) = L(D)$$

证明思路: 设 E, F 的变量集为 $L_1, L_2, \ldots, L_m$.

$\Rightarrow$ 设 $c = c_1 c_2 \ldots c_k \in L(C)$, 其中每个 c_i 均为单个符号. 任取 $w \in L(E)$，满足 $w = w_1 w_2 \ldots w_k$ ，且有 if $w_i \in L_j$，then E 具体化为 C 时使用 c_i 替换 L_j .
$\because L(E) = L(F)$ ，$\therefore w \in L(F)$. 因而，有 $c \in L(D)$.
$\therefore L(C) \subseteq L(D)$. 同理可证 $L(D) \subseteq L(C)$. $\therefore L(C) = L(D)$.

$\Leftarrow$ 假设 $L(C) = L(D)$ ，证明 $L(E) = L(F)$. （留作思考）

正规表达式的代数定律

✧ 代数定律的具体化（应用举例）

- 用于发现和测试关于正规表达式的定律.
- 举例: 对于具体符号*a*, 容易证明 $a\,a^* = a^*\,a$, 由此可以发现定律 $L\,L^* = L^*\,L$, 其中 L 为变量，可以实例化为任何语言.
- 举例: 若要验证定律 $L(M+N) = LM+LN$，只要验证，对于具体符号*a*、*b*、*c*，$a(b+c) = ab+ac$ 成立.
- 举例: 若要验证 $L+ML = (L+M)\,L$ 是否成立，可以验证对于具体符号*a*、*b*，$a+ba = (a+b)a$ 是否成立. 但后者不成立，*aa* 属于 $(a+b)a$ 代表的语言，而不属于 $a+ba$ 代表的语言.

课后练习

- 必做题:
  - Ex.3.1.1 (b), (c)
  - ! Ex.3.1.2 (b)
  - *! Ex.3.1.5
  - Ex.3.4.1 (c), (g)
  - Ex.3.4.2 (b), (d)
  - !!Ex.3.1.3(a), (b)

课后练习

✧ 自测题:

– 试给出下列每个正规语言的一个正规表达式:

1) $\{ xwx^R \mid x, w \in (a+b)^+ \}$,
其中 $(a+b)^+ = (a+b)(a+b)^*$，x^R为x的反向(即反转)

2）$\{ w \mid w \in \{a, b\}^* \wedge \exists x, y(x, y \in \{a, b\}^* \wedge w = xy \wedge |y| = 3 \wedge y = y^R) \}$

3）$\{ w \in \{a, b\}^* \mid$ w 中既不包含子串 aa， 也不包含子串 bb $\}$

4）$\{ a^n b^m \mid n, m \geq 0$ 且 $n + m$ 为偶数 $\}$

5) $\{ w \mid w \in \{a, b\}^*$，$|w| \geq 1$，且 w 的后20位至少有一个 a $\}$

6) $\{ w \mid w \in \{a, b\}^*$，$|w| \geq 1$，且当 w 以a结尾时，它的长度为奇数 $\}$

7) $\{ w \mid w \in \{a, b\}^*$，$|w| \geq 2$，且 w 的前5位至少有一个子串 aa $\}$

8) $\{ w \mid w \in \{a, b\}^*$，$|w| \geq 2$，且 w 的第2位至第5位至少有一个 a $\}$

9) $\{ w \mid w \in \{0,1\}^*$，w至少含有3个1，且倒数第3位为1$\}$

That's all for today.

Thank You