第 三 讲

✧ 正规表达式与正规语言

正规语言的不同表达形式

NFA
非确定有限自动机

DFA
确定有限自动机

正规语言

（regular languages）

regular grammar
正规文法

ε－NFA
带 ε 转移的非确定有限自动机

regular expressions
正规表达式

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

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 $E\in R$，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

✧ 正规表达式算符优先级

算符优先级（*precedence*）依次为

– *
– • 连接
– +

正规表达式

✧ 正规表达式的几个派生运算符

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

✧ 正规表达式举例

设计表示如下语言的正规表达式:
该语言中的每个字符串由交替的 0 和 1 构成

- $(01)^* + (10)^* + 0(10)^* + 1(01)^*$
- $(\varepsilon + 1)\,(01)^*\,(\varepsilon + 0)$
- $(\varepsilon + 0)\,(10)^*\,(\varepsilon + 1)$

正规表达式

✧ 正规表达式举例

课堂练习 设计如下语言的正规表达式:

✧ 正规语言（*regular language*）

– 归纳定义

字母表 ∑ 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 Σ 上的语言 ***R***，若存在 Σ 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

正规表达式的代数定律

- 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

- ✧ 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）

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

✧ 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$S_{li}'\,(1 \le l_i \le n)$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

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

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)
- 思考题:
  - !!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∈{a, b}* ∣ 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 ∣ w∈{a, b }*，∣w∣≥ 1，且 w 的后20位至少有一个 a }

6) { w ∣ w∈{a, b }*，且 w 中既没有相邻的 a，也没有相邻的 b }

7) { w ∣ w∈{a, b }*，∣w∣≥ 1，且当 w 以a结尾时，它的长度为奇数 }

8) { w ∣ w∈{a, b }*，∣w∣≥ 2，且 w 的前5位至少有一个子串 aa }

9) { w ∣ w∈{a, b }*，∣w∣≥ 2，且 w 的第2位至第5位至少有一个 a }

That's all for today.

Thank You