

语言的识别

定义

1. 正则表达式：递归定义为
 - 符号 \emptyset 是一个正则表达式
 - 符号 λ 是一个正则表达式
 - 若 $x \in I$ ，则符号 x 是一个正则表达式
 - 当 \mathbf{A}, \mathbf{B} 是正则表达式时，符号 (\mathbf{AB}) , $(\mathbf{A} \cup \mathbf{B})$, \mathbf{A}^* 都是正则表达式。
2. 每一个正则表达式都表示一个正则集合：
 - \emptyset 表示空集；
 - λ 表示集合 $\{\lambda\}$ 即空串组成的集合；
 - \mathbf{x} 表示集合 $\{x\}$ ，即只包含单个符号 x 的字符串；
 - (\mathbf{AB}) 表示 \mathbf{A} 和 \mathbf{B} 表示的集合的连接。
 - $(\mathbf{A} \cup \mathbf{B})$ 表示 \mathbf{A} 和 \mathbf{B} 表示的集合的并。
 - \mathbf{A}^* 表示 \mathbf{A} 代表的集合的克林闭包。

定理

- 一个集合是正则集合当且仅当该集合能被一个有限状态自动机识别。
- 一个集合是正则集合当且仅当该集合能由一个正则文法生成。

方法

- 如何构造一个非确定性的有限状态自动机使其能识别一个正则文法产生的集合？
 - 对于正则文法 $A \rightarrow aB$ ，在转移函数中包含 $f(s_A, a) = s_B$ ，其中 s_A 和 s_B 分别是 A , B 对应的在自动机中的状态。对于正则文法 $A \rightarrow a$ ，在转移函数中包含 $f(s_A, a) = s_F$ 其中 s_F 是终止状态。对于正则语法 $A \rightarrow \lambda$ ，在自动机的终止状态中包含 s_A 即可。自动机的起始状态即为 s_{s_0} ，其中 s_0 是正则文法的起始字符串。
- 如何通过一个非确定性的有限状态自动机构造一个相应的正则文法。
 - 与通过正则文法生成自动机的过程刚好相反，可以归纳为如果在转移函数中存在 $f(s_A, a) = s_B$ ，则在正则文法中要包含 $A \rightarrow aB$ 。