

问题

构造一个算法，判断正则语言 L 是否包含无穷多个长度为偶数的符号串。

解答

假设字母表 $\Sigma = \{a_1, a_2, \dots, a_m\}$ ，并且设语言 $L_1 = \{\text{长度为偶数的符号串}\}$ ，有

$$L_1 = L(((a_1 + a_2 + \dots + a_m)(a_1 + a_2 + \dots + a_m))^*).$$

找到所有长度为偶数的符号串所构成的语言

由此可得，语言 L_1 是正则语言。又语言 L 是正则的，则 $L_1 \cap L$ 是正则语言。

容易得出：语言 L 包含无穷多个长度为偶数的符号串当且仅当语言 $L_1 \cap L$ 包含无穷多个符号串。语言 $L_1 \cap L$ 是正则语言，存在算法判定语言 $L_1 \cap L$ 是否包含无穷多个符号串。因此，存在算法，判断正则语言 L 是否包含无穷多个长度为偶数的符号串。

具体算法：首先，构造一个接受正则语言 $L_1 \cap L$ 的确定型有穷接受器，并给出它的状态转移图；然后，找出状态转移图中可以构成某个回路的所有顶点；最后，判断这些顶点中是否有一个顶点位于从初始状态到终止状态的某条路径中。

如果有，则正则语言 L 包含无穷多个长度为偶数的符号串。

如果没有，则正则语言 L 仅包含有限多个长度为偶数的符号串。