第4卷 第5期
1993年10月

软 件 学 报
JOURNAL OF SOFTWARE

Vol. 4, No. 5
Oct. 1993

关于一个串为正则语言中某串的子串的判定算法

庄 雷

(郑州大学计算机系,郑州 450052)

ALGORITHM FOR DECIDING WHETHER A STRING IS A SUBSTRING OF THE STRING WHICH BELONGS TO A REGULAR LANGUAGE

Zhuang Lei

(*Department of Computer Science, Zhengzhou University, Zhengzhou* 450052)

Abstract The algorithm is given for deciding whether a string s is a substring of the string which is in a regular language L. That also means assuming s is a string over Σ, L is a regular language over Σ. s is a substring of the string in L, if and only if s is a substring of the string which belongs to the set $V_n=\{w\in\Sigma^*\,|\,w\in L, |w|\leqslant 2n+k-2\}$ where $k=|s|$, n is a nature number.

摘要 本文给出一个判定Σ上的任意串s是否为一正则语言L中某个串的子串的算法.即设s为Σ上的任一串,L是Σ上的任一正则语言,则s为L中某个串的子串,当且仅当s为集合$V_n=\{w\in\Sigma^*\,|\,w\in L, |w|\leqslant 2n+k-2\}$中某串的子串,其中$k=|s|$,n是某个自然数.

本文给出一个有穷字母表Σ上的任意串是否为Σ上的一正则语言中某个串的子串的判定算法.设L是Σ上的一个正则语言,s是Σ上长为$k(k\geqslant 2)$的任意一个串,则存在一个判定s是否是L中某个串的子串的算法,即s是L中某个串的子串,当且仅当s是集合

$$V_n=\{w\in\Sigma^*\,|\,w\in L, |w|\leqslant 2n+k-2\}$$

中某个串的子串,其中n为某个不依赖于k的自然数.

这个算法的意义在于,只需在有穷多个字符串上逐个进行检验,即可判定s是否为L中某个串的子串.这对研究字符串之间的关系,特别对研究由子串关系定义的Σ^*上的半序关系所引进的几种广义凸语言都是很有意义的.

定理(子串判定算法):

本文1991年7月收到.本文系国家自然科学基金资助项目.作者庄雷,讲师,主要研究领域为形式语言与自动机.

设 L 是有穷字母表 Σ 上的一个正则语言，则存在一自然数 n，使对于 Σ 上的长为 k 的任意串 s，$k\geqslant 2$，s 为 L 中某个串的子串，当且仅当 s 为集合

$$V_n=\{w\in\Sigma^*\,|\,w\in L,\ |w|\leqslant 2n+k-2\}$$

中某个串的子串.

证明：由于 L 是有穷字母表 Σ 上的正则语言，则存在一接受 L 的有穷状态自动机

$$M=(K,\Sigma,\delta,q_0,F),$$

令 $|K|=n$，即自动机的状态个数为 n（若 $|K|=1$，则令 $n=2$）. 显然，只需证明定理的"仅当"部分. 为此，只需证明，若 $s=a_1a_2\cdots a_k$ 是 $w(\in T(M)-V_n)$ 的子串，则 s 必是 V_n 中某个串的子串.

设 $w=b_1b_2\cdots b_{n-1}b_n\cdots b_{n+k-1}b_{n+k}\cdots b_{2n+k-2}\cdots b_m$，$m>2n+k-2$，w 是 $T(M)-V_n$ 中的任意字，而 s 是 w 的子串. 现将 w 分成三部分. 第一部分由前 $n+k-1$ 个符号组成，记作〈Ⅰ〉；第二部分从 b_{n+k} 到 b_{2n-k-2} 由 $n-1$ 个符号组成，记作〈Ⅱ〉；从 b_{2n+k-1} 到串结束为第三部分，记作〈Ⅲ〉. 根据 s 在 w 中的位置，采取相应的步骤证明.

1. 若串 s 完全位于〈Ⅰ〉中，则在有穷状态自动机 M 输入字 w 的过程中，在〈Ⅱ〉中已输入 $n-1$ 个字符，经历了 n 个状态，当输入完字段〈Ⅲ〉时，M 在〈Ⅱ〉、〈Ⅲ〉部分所经历的状态中至少有一个要重复出现两次，设这个状态为 q. 于是可将 w 写成 $w=x_1x_2x_3$，使得 $q(q_0,x_1)=q$，$\delta(q,x_2)=q$，$\delta(q,x_3)=p$，而 $p\in F$. 显然，$x_1x_3\in T(M)$，而 $|x_2|>0$，所以 $|x_1x_3|<|x_1x_2x_3|=|w|$. 若 $|x_1x_3|\leqslant 2n+k-2$，则定理得证，否则重复上述过程，经过有限次以后，得到一个字 y，使得 $y\in V_n$，且 s 仍为其子串.

2. 若 s 不是完全位于〈Ⅰ〉中，即有元素 $a_i\in s$ 使其位于〈Ⅰ〉以后的位置，则在 M 输入字 w 的过程中，在输入了前 $n-1$ 个字符后，当输入完第 n 个字符时，M 中至少有一个状态重复出现二次. 同样，可将 w 分成三段，即 $w=x_1x_2x_3$，而 $|x_2|>0$，使得 $x_1x_3\in T(M)$ 且 s 在 x_1x_3 中. 若 $|x_1x_3|>2n+k-2$ 且 s 仍不完全位于〈Ⅰ〉中，则重复上述过程；若 $|x_1x_3|>2n+k-2$，但 s 完全位于〈Ⅰ〉中，则按步骤 1 进行；若 $|x_1x_3|\leqslant 2n+k-2$，，则定理得证.

总之，经过有限次上述相应的步骤，即可证明：s 为 V_n 中某串的子串. 于是定理得证.

参考文献

1　J. E. Hopcroft, J. D. Ullman, Introduction to Automata Theory, Languages and Computation, Addison－Vesley Publishing Company, 1979.