

## 邻间关系匹配算法研究

周大庆<sup>1,2</sup>, 蔺娟茹<sup>2</sup>

(1. 西南交通大学 计算机与通信工程学院, 四川 成都 610031;

2. 山西师范大学 数学与计算机科学学院, 山西 临汾 041004)

**摘要:**对于26个字母的全排, 它们的邻间关系是唯一的。文中根据这个特性, 针对子串长度较长的(大于26)字符串匹配问题, 提出了一种基于邻间关系的匹配算法。该算法把字符串的邻间关系转化为十进制的数值, 并利用这一数值实现字符串的快速匹配。该算法时间复杂度为  $O(m-n)$ , 且算法简便, 容易实现。

**关键词:**字符串匹配; KMP; Hash; BM; 邻间关系

**中图分类号:** TP301.6

**文献标识码:** A

**文章编号:** 1673-629X(2006)11-0117-02

## Neighbor Relationship - Based String Matching Algorithm

ZHOU Da-qing<sup>1,2</sup>, LIN Juan-ru<sup>2</sup>

(1. School of Computer Sci. and Communication Eng., Southwest Jiaotong Univ., Chengdu 610031, China;

2. School of Maths and Computer Science, Shanxi Teachers' University, Linfen 041004, China)

**Abstract:** Giving the arrangement of twenty-six letters, the sequence is exclusive, and the neighbor relationship of the letters in the sequence is a fixed value. Taking advantage of the feature given above, presents a string matching algorithm based on neighbor relationship, to solve the question of long string matching (more than twenty-six). The algorithm transforms the neighbor relationship to a value, and achieves the fast string matching by using this value. The time complexity of the algorithm is  $O(m-n)$ . By the way, this algorithm has the feature of simplicity and convenience, and it is easy to realize.

**Key words:** string matching; KMP; Hash; BM; neighbor relationship

## 1 Hash 算法, KMP 算法和 BM 算法

字符串匹配算法由来已久, 最著名且使用最广泛的算法是 Hash 算法、KMP 算法和 BM 算法。

Hash 算法的思想是样本总空间思想, 字符的 ASCII 码值为: 1~255, 对于长度为  $n$  的子串, 样本总空间是  $255n$ , 利用样本总空间是绝对没有冲突的可能, 但是  $n$  大于一定的数值, 就没办法实现, 于是采用 Hash 函数法:  $\text{Hash}(X_i) = X_i * B_i / R_i$ ,  $B_i$  可以是素数序列, 也可以固定为某个值,  $R_i$  可以是自然数序列也可以固定为某个值, 这些系数的选择要根据不同的问题而定, 虽然如此, 子串的长度还是有一定的限制, 实现算法需要  $(m-n)$  步<sup>[1]</sup>。

KMP 算法是利用已经匹配的结果来简化算法的复杂度<sup>[2]</sup>, 当子串中的第  $i$  个字符出现不匹配时, 子串回退到  $\text{next}(i)$  个位置, 主串的比较窗口和比较位都做相应的变化, 如下例:

子串  $S[i]$ : asasaadtas  
 $\text{next}[i]$ : 1112342112

主串  $M[i]$ : asasasetasaaadtastasasasaa...

假定: 匹配的窗口标志为  $p$ , 即匹配的内容是  $M[p] \cdots M[n]$ ,  $n$  为子串长度,  $M[]$  为主串数组。

当匹配到主串 6 时  $M[6]$  为  $s$ ,  $S[6]$  为  $a$ , 由于不匹配, 下一次匹配时, 主串的窗口标志从  $p=1$  跳到  $p=3=p+i-\text{next}[6]$ , 即: 主串中的第 2 个  $a$ ; 子串回退到第  $\text{next}[6]$  个位置, 即: 子串中的第 2 个  $s$ 。主串中的比较位是  $6=p+\text{next}[6]-1$ , 也就是用  $M[6]$  跟  $S[4]$  比较。该算法避免了  $(m*n)$  的运算复杂度, 但是匹配步数远大于  $(m-n)$ <sup>[3]</sup>。

BM 算法的思想与 KMP 思想类似, 但是方向相反, 它利用子串还没有匹配的内容来简化算法的复杂度, 如果一个没有匹配成功的字符 ( $M[i]$ , 对应于子串的字符  $S[j]$ ) 没在子串还没有匹配过的内容中, 则主串的窗口后跳到该字符的后面, 如果在子串剩余的内容中, 取离  $j$  最近的位置, 则主串的窗口后退到主串中该字符与子串中该位置对齐。该算法的运算复杂度同样大于  $(m-n)$ <sup>[4,5]</sup>。

BM 算法跳的优势要体现的条件是子串不能太长, 当子串过长的情况下, 主串中的字符在子串中的概率将比较高, 跳的可能性就会大大减少, BM 算法的优势也就不存在了。

收稿日期: 2006-02-25

作者简介: 周大庆 (1974-), 男, 陕西渭南人, 硕士研究生, 讲师, 研究方向为数据挖掘; 导师: 戴齐, 副教授, 研究方向为数据挖掘、人工智能。

## 2 邻间关系匹配算法(Neighbor Relationship Based String Matching Algorithm)

对于一个子串,如果能从子串的某一个总体特征出发,每次用主串窗口的特征与子串的特征相比较,如果总体特征一致,则进行详细匹配;这种思想需要解决 2 个问题。

(1) 冲突的概率。

(2) 总体特征的计算是否简便。

简单的总体特征是子串的 ASCII 码值总和或者平均数,显然冲突太多,不可实现。Hash 算法的样本总空间思想也是子串的总体特征之一,但是算法复杂度较高,子串的长度有限制,参数的设计会影响冲突的概率。还有没有别的总体特征呢?

子串字符之间的邻间关系也是一个总体特征,比如:“abcd”的邻间关系是“+++”,而“zyxw”的邻间关系是“---”,如果用 2 表示“+”,1 表示“=”,0 表示“-”,则“icomefromchina”的邻间关系是个 3 进制数值:0220222202220,转换成十进制是:49 1910,这个数值就是该子串的邻间关系值,该字符串的长度是 14,这个长度的邻间关系值的样本空间是  $3^{13} - 1 = 159\ 4322$ 。对于长整形的变量,可以存储的最大数值是  $2^{64} - 1$ ,也可以存储  $3^{40}$ ,也就是说,在不对该数值做附加处理的情况下,可以用该方法来实现 41 位长的字符串匹配算法。由此可见,利用邻间关系可以实现较长字符串的快速匹配。如果把 26 个字母拿来排列的话,它们的邻间关系值是唯一的,应该说子串的长度  $n$  大于 26 的情况下,就适合用邻间关系匹配算法,当然, $n$  越大,冲突的可能就越小。

## 3 算法的实现

假定:子串长度为  $n(n \leq 41)$ ,主串长度为  $m$ 。

(1) 邻间关系如下:

$$0 \quad X_i < X_{i-1}$$

$$R(X_i) = 1 \quad X_i = X_{i-1} \quad i > 1, i \leq m$$

$$2 \quad X_i > X_{i-1}$$

(2) 主串前  $n$  位邻间关系匹配值为:

$$V(X_n) = R(X_2) * 3^{(n-2)} + R(X_3) * 3^{(n-3)} + \dots + R(X_n) * 3^{(n-n)}$$

(3) 其他邻间关系匹配值为:

$$V(X_{k+1}) = 3 * (V(X_k) - R(X_{k-n+2}) * 3^{(m-2)}) + R(X_{k+1}) \quad k \geq n, K < m$$

说明:

该算法的时间复杂度为  $(m - n)$ 。

对于  $n > 41$  的情况下,可以进行分段,即第一段长度

为 41,后面的每段长度为 40,通过多段比较可以实现匹配。

## 4 冲突的解决

假设存在一个排列  $P$ ,是 26 个字母的排列,那么,它的邻间关系是唯一的,现在在这个排列的某个位置插入一个字母(比如:在序列  $\dots az \dots$  的  $az$  之间插入一个字母),产生一个新的字符串,并产生一个新的邻间关系值,对于这个新的邻间值,有 24 个字母都合适,因为,从  $b \dots y$  都满足大于  $a$  且小于  $z$ 。

所以,在这里可以给冲突下一个定义:在一个排列中,有一定数量的字母在一定的范围内浮动,但是不影响它的邻间关系值。

要减少这个冲突,可以在匹配的过程中引入一个参数  $\epsilon$ ,  $\epsilon$  为序列的数学期望,  $\epsilon = \sum (X_i - \mu)^2$ ,  $\mu$  为子串序列的平均值,  $X_i$  为主串窗口的元素。

那么,对一个序列,通过计算它的邻间关系值和数学期望,就可以大大减少它冲突的几率,从而实现快速的字符串匹配。

## 5 总结

对于 26 个字母的全排,它们的邻间关系是唯一的。文中根据这个特性,针对子串长度较长的(大于 26)字符串匹配问题,提出了一种基于邻间关系的匹配算法。该算法把字符串的邻间关系转化为十进制的数值,并利用这一数值实现字符串的快速匹配。该算法时间复杂度为  $O(m - n)$ ,且算法简便,容易实现。对于文中提出的冲突解决的方案,没有任何理由认为这个方法是唯一的或者是最佳的,还有没有更好的方案期待大家进一步的研究。

## 参考文献:

- [1] Park J S, Chen M S, Yu P S. An effective hash-based algorithm for mining association rules[C]//Proceedings of ACM SIGMOD International Conference on Management of Data. [s.l.]:[s.n.],1995:175-186.
- [2] Knuth D E, Jr. Morris J H, Pratt V R. Fast pattern matching in strings[J]. SIAM Journal on Computing, 1977,6(1):323-350.
- [3] 李 静. 字符串的模式匹配算法——基于 KMP 算法的讨论[J]. 青岛化工学院学报:自然科学版,2002(2):78-80.
- [4] Boyer R S, Moore J S. A fast string searching algorithm[J]. Communications of the ACM,1977,20(10):762-772.
- [5] 庞善臣,王淑栋,蒋昌俊. BM 串匹配的一个改进算法[J]. 计算机应用,2004(12):13-15.

2006 年起《微机发展》更名为《计算机技术与发展》

欢迎投稿,欢迎订阅!