

文章编号:1001-9081(2005)02-0365-02

基于改进双链树的多模式匹配算法

唐 皓, 卢显良

(电子科技大学 计算机科学与工程学院, 四川 成都 610054)

(tanghao@uestc.edu.cn)

摘 要:在基于键树的多模式匹配算法中,键树的物理存储方式为双链树。通过借鉴 KMP 算法的思想,在键树的基础上增加了将辅助跳转结点变成改进的双链树。改进后的存储方式和匹配算法加快了匹配过程,并且做到了在搜索匹配的过程中不用回溯。

关键词:双链树;多模式匹配;无回溯

中图分类号:TP301.6 **文献标识码:**A

Arithmetic for matching multiple patterns based on improved doubly-chained tree

TANG Hao, LU Xian-liang

(College of Computer Science and Engineering, University of Electronic Science and Technology of China,
Chengdu Sichuan 610054, China)

Abstract: In the arithmetic for matching multiple patterns based on digital search tree, the physical storage mode of digital search tree is doubly-chained tree. Using the idea of KMP arithmetic, digital search tree has been turned into improved doubly-chained tree through added assistant jump-node. The improved storage mode and arithmetic have quickened the speed of matching, and have implemented non-backtracking in process of matching.

Key words: doubly-chained tree; multiple pattern matching; non-backtracking

0 引言

模式匹配已经广泛应用于入侵检测、全文检索、邮件过滤等技术领域。模式匹配可分为单模式匹配和多模式匹配。单模式匹配是指一次操作只能判断待匹配序列是否和某一个模式匹配;多模式匹配是指一次操作能判断待匹配序列是否和多个模式匹配。单模式匹配在小型正常数据库和实时性低的实验环境下是可行的;但是在要求实时性强的系统中,尤其是对拥有较大正常模式数据库的庞大程序,这样的搜索方式效率低下。所以我们的目标是进一步的缩短检测的时间。在各个序列模式有较多的共同元素情况下,采用多模式匹配的方法可以大大加快检索速度。传统的多模式匹配采用的是键树的存储方式,在本文的系统中,键树的物理存储方式为双链树。

1 传统多模式匹配算法分析

键树多用于字符串的匹配。以树的孩子兄弟链表来表示键树,每个分支结点包括三个域:symbol 域存储关键字的一个字符;next 域存储指向右兄弟的指针;first 域存储指向第一棵子树根结点的指针;同时,叶子结点的 infoptr 域存储指向该关键字记录的指针。此时的键树又称为双链树。

在匹配过程中,从双链树的根指针出发,顺着 first 指针对一棵子树进行匹配,若不匹配,则沿 next 域进入右兄弟子树进行匹配,当匹配失败的时候,回到树的根部继续匹配下一个待匹配序列,如此以来在匹配失败的时候已经部分匹配的信息就浪费了。

如图 1 所示:模式库中的关键字为“UFO”、“UESTC”、“STOP”、“SAD”、“OEM”。现在有形如“UESTOEF”的待匹配序列。当匹配操作执行到 O 时,序列和模式库中的“UESTC”失配,需要回溯再次进行匹配。但是从已经部分匹配的序列“UEST”可以看出,待匹配序列中的 ST 和模式库中的关键字“STOP”的前两个元素匹配。如果从“UES...”开始匹配,则浪费了第一次匹配过程中已经得到的部分匹配信息。

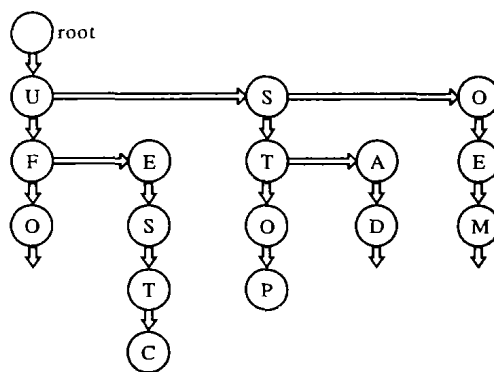


图 1 双链树的存储结构

2 实现机制

2.1 基本思想

我们把双链树的存储结构进行改进,在普通结点的基础上增加了辅助跳转结点,即元素值为 0 的结点(设正常元素中没有 0)。这样在双链树上进行匹配过程时遇到空指针或

收稿日期:2004-07-16;修订日期:2004-10-15

作者简介:唐皓(1977-),男,四川乐山人,硕士研究生,主要研究方向:操作系统、计算机网络;卢显良(1943-),男,教授,博士生导师,主要研究方向:计算机系统软件、计算机网络、操作系统。

跳转到了辅助跳转结点都表示匹配失败,此时根据跳转信息可以跳过肯定不匹配的若干结点,直接转换到下一个可能匹配的结点。

根据正常行为模式特征库中的最大模式:“UFO”、“UESTC”、“STOP”、“SAD”、“OEM”。把库中各个元素组织成普通结点,并根据部分匹配信息建立 3 个辅助跳转结点,如图 2。

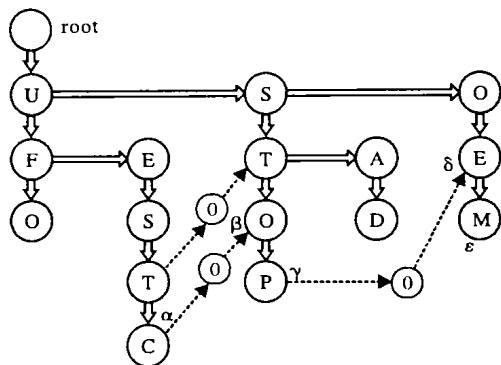


图2 改进的双链树

假设有一个待匹配序列“UESTOEF”，在利用传统的双链树检测时，在 α 结点上失配后，就回到 root 结点，从待匹配序列的下一个字符开始继续搜索。这样前一次搜索过程中获得的信息就浪费了，因为在匹配过程中有一部分的匹配结果仍然可以应用。例如在 α 结点失配时，模式中的“UE”两个元素肯定失配了，但是“ST”这两个元素仍然存在匹配到后面某个模式的可能性，这样的信息可以利用来跳转到合理的检测状态。

2.2 改进的双链树搜索算法

我们根据各个模式的内容构造了若干辅助跳转结点，在对序列进行扫描的时候我们记录其扫描窗口的起始位置和匹配游标的位置，如图 3。

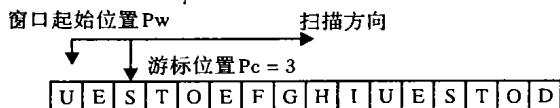


图3 对待匹配序列的扫描

当匹配程序进行的时候，扫描窗口 P_w 定在最左边，游标 P_c 为 1，失配元素个数 M_c 为 0，从双链树的 root 结点开始搜

索，比较游标指向的元素和结点元素：

1) 若匹配，则跳转到当前结点左指针指向的孩子结点，同时 P_c 加 1，若孩子指针为空则模式匹配成功， M_c 清 0，序列窗口向后滑动游标个元素 ($P_w = P_w + P_c$)，并跳到 root 结点，同时 P_c 恢复到 1。

2) 若失配，则跳转到当前结点右指针指向的兄弟结点，若右指针为空则模式匹配失败， $M_c = M_c + P_c$ ，序列窗口向后滑动游标个元素 ($P_w = P_w + P_c$)，并跳到 root 结点，同时 P_c 恢复到 1。

3) 若失配时当前结点右指针指向的兄弟结点为辅助跳转结点，则结点左指针存储的是部分失配数目 PM_c ，序列窗口向后滑动游标个元素 ($P_w = P_w + PM_c$)，并跳到辅助跳转结点右指针指向的结点，同时让 $P_c = P_c - PM_c$ 。

4) 当失配数 M_c 大于某一阈值 (通常定为 7) 时，认为有异常行为，产生告警。

例如：序列‘UESTOEF’在双链树上的匹配路径为：‘root $\rightarrow \alpha \rightarrow \beta \rightarrow \gamma \rightarrow \delta \rightarrow \epsilon$ ’，在每一结点的失配数为：

$$M_c(\alpha) = 2 \quad M_c(\beta) = 2 \quad M_c(\gamma) = 4 \quad M_c(\delta) = 4 \quad M_c(\epsilon) = 7$$

可见当检测进行到 η 结点时，系统告警。

3 性能分析及对比

本文对 Linux 下的各个程序的系统调用序列进行匹配运算次数对比实验，并且把实验中取得模式长度根据变长模式平均长度分为四类：

超短模式：模式平均长度 3 以内；短模式：模式平均长度 3~5；中长模式：模式平均长度 6~9；长模式：模式平均长度 10 以上。

针对每一种模式的长度进行了三种算法的比较：串匹配算法、双链树算法、改进的双链树算法。其中的运算次数是指所有类型的运算：数值赋值、指针移动、元素比较等运算的次数。

实验数据表明采用双链树的匹配算法比一般串匹配算法性能明显高出许多。而改进后的双链树比传统的双链树又有较大的性能改进，平均可以提高性能 48%。模式的平均长度越长，双链树越比字符串匹配算法有优势，但是改进的双链树算法对比双链树的优势却逐渐下降，如表 1。

表1 三种模式库搜索方法性能比较

模式长度	性能对比				
	串匹配法运算次数	双链树法运算次数	改进双链树法运算次数	改进双链树对串匹配改进率	改进双链树对双链树改进率
超短模式	3621	2554	1441	2.51	1.77
短模式	4398	1879	1165	4.56	1.61
中长模式	5187	2125	1593	5.81	1.33
长模式	6369	927	825	7.72	1.12
平均	4894	1871	1031	4.75	1.48

根据对实验数据的分析，以下两个因素导致了图中的变化趋势：

1) 有共同前缀的模式较多，共同前缀也较长，当模式变长的时候节省的比较运算时间显著增多。

2) 跳转结点大多建立在模式靠后的位置，这样在模式短的情况下跳转比较有效率，而在长的模式下则体现不出来。

参考文献：

- [1] IDURY RM, SCHÄFFER AA. Multiple matching of rectangular patterns[J]. Information and Computation, 1995, 117(1): 78-90.
- [2] RICHARD C, HARIHARAN R. Tree Pattern Matching and Subset

Matching in randomized $O(n \log^3 m)$ time[A]. Conference Proceedings of the Annual ACM Symposium on Theory of Computing[C], 1997.

- [3] RICHARD C. Tree pattern matching to subset matching in linear time[J]. SIAM Journal on Computing, 2003, 32(4): 1056-1066.
- [4] RAMANA MI. Multiple matching of parameterized patterns [J]. Theoretical Computer Science, 1996, 154(2): 203-224.
- [5] FAN J-J. Design of efficient algorithms for two-dimensional pattern matching[J]. IEEE Transactions on Knowledge and Data Engineering, 1995, 7(2): 318-327.
- [6] 严蔚敏, 吴伟民. 数据结构(第二版)[M]. 北京: 清华大学出版社, 1997.