Journal of Management Science and Statistical Decision
Vol.7 No1.(2010) p66-69

A Multi-pattern Matching Algorithm Based on WM Algorithm

Genzhen Yu, Qinquan Gao, Fanlin Meng, Changhong Fu, Shunxiang Wu*
Department of Automation
Xiamen University
Xiamen, China
wsx1009@163.com

***Abstract*-The research on the algorithms of pattern-matching is an important subject in the field of computer study. The algorithms can range from single-pattern matching and multi-pattern matching algorithms to extended characters matching and regular expression. Among the many multi-pattern matching algorithms, AC algorithm and WM algorithm would be the two most classical algorithms, but these two algorithms have their obvious shortcomings. The multi-pattern matching algorithm proposed in this paper filtrates the texts which do not match correctly with the idea of jumping ahead of the WM algorithm firstly, and then matches the text with the idea of rapidly matching of the AC algorithm which can improve the efficiency of the algorithm.**

Keywords-Multi-pattern Matching; AC; WM

I. INTRODUCTION

The pattern-matching algorithm is one of the key technologies which can be applied in the string analysis, intrusion detection, detection of DNA sequence and other disciplinary field. The Research of the pattern-matching algorithm in the world also achieved remarkable achievements, such as the KMP algorithm [1], the AC algorithm [2], the BM algorithm [3] as well as the WM algorithm [4].The matching-pattern also developed from the single-pattern and the multi-pattern to extended characters matching, regular expression, approximate matching and so on. The book which is named the "Flexible pattern matching in strings"[6] wrote by Gonzalo Navarro and Mathieu Raffinot described the pattern-matching algorithms which are the most popular in the world in detail.

The WM algorithm is one of the pattern-matching algorithms which have the highest efficiency. At present, many improved algorithms based on the WM algorithm have been proposed, as Hui Jiang and Yu-hong Zhang [7] said that there are many methods which can accelerate the match, such as separating patterns according to their length, promoting the PREFIX table, improving the HASH table and so on. The obvious inadequacy of W-M algorithm is that it is not optimized for the match after finds out the string list which may be successfully matched. While the automata model of the A-C algorithm can solve the problem better.

The AC-WM algorithm proposed in this paper is a multi-pattern matching algorithm based on the filtering method. The algorithm skips the texts which do not match correctly with the idea of jumping ahead of the WM algorithm, and then match the text with the idea of rapidly matching of the AC algorithm which can greatly improve the efficiency of the algorithm.

II. DESCRIPTION OF AC ALGORITHM

The AC algorithm is the expansion of the KMP algorithm in the case of multi-pattern matching. Its core part is that establish the model of AC automaton (DFA), failure link and output function in the pre-processing stage, alternately apply the DFA, failure link and output function in the search stage to reach the effect which is the match of not backtracking.

It is supposed that we apply the following pattern strings to match the texts: "he", "she", "his", "hers", the DFA is established in the pre-processing stage, and then calculate the failure links and output functions, the Fig. 1 has shown the above process. Where: the solid line arrows show the state transition of the automaton, while the dashed line arrows show the failure link in which the end state of the substring points at the end state of the suffix of the substring. In addition, the state of two circles shows that the string has been successfully matched. As shown in Figure. 1, "h" is the "sh" the longest suffix, so the failure links of the fourth state have pointed at the first state.

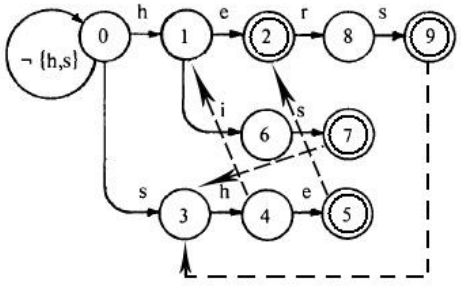

Figure 1. AC automation.

III. DESCRIPTION OF W-M ALGORITHM

The WM algorithm is the expansion of the BM algorithm based on the suffix in the case of multi-pattern matching, the core idea is that the following tables have been established in the pre-processing stage: the table of SHIFT, the table of HASH and the table of PREFIX. The BM algorithm is applied in the process of matching to find out the position where the single character in the pattern string, while the WM algorithm is applied to match the characters based on the matching block which is the substring of fixed length (the "B" represents the fixed length in the paper, its value is equal to two or three). Compared to the BM algorithm, the latter has the greater jump after unsuccessfully matched each time.

The table of SHIFT has been established firstly. The blocks which length is possible two or three have been mapped to the index i of the table of SHIFT with the function of hash in the patterns P1P2 ... Pn. If the block does not appear in any pattern string of the set of the patterns, the value of the SHIFT [i] is equal to m-B +1, where "m" is the shortest length of the patterns. If the block appears in the some patterns, we should calculate the rightest position q of the block in the patterns, and then the SHIFT [i] obtains the minimum between m-q and m-B +1.

The table of HASH has been established secondly. The value of hash of the block whose length is the B has been used as the index, and the HASH [j] is a pointer named the "p" which point at initial address of pattern-list which has the same suffix.

The pointer of HASH [j] points at the list of pattern strings, and the table of PREFIX at the same time. The current value of hash of the string which length is the B has been stored in the table of PREFIX.

The average complexity of the WM algorithm is O (BN / m), where the "B" is the length of the character block, while the "N" is the length of the text, and the "m" is the shortest length of the patterns.

IV. THE NEW METHOD OF AC-WM ALGORITHM

The AC-WM algorithm is base on the WM algorithm, and combined with the advantages which belong to the idea of jumping ahead of the WM algorithm as well as the idea of rapidly matching of the AC algorithm, it greatly improve the efficiency of the algorithm.

The AC algorithm is based on the prefix algorithms, while the WM algorithm is based on the suffix algorithm, the matching process of the both of them is from left to right. To combine with the advantages of both of them, it should transform one of them. In this paper, we reserve the AC machine model and output function, and transform the WM algorithm, making the WM algorithm into matching algorithm based on the prefix, and the way of text matching has changed which is from right to left. The tables of SHIFT and PREFIX of the WM algorithm have been altered, and the table of HASH has been given up using.

There are 65536 different kinds of blocks, and the sizes of the SHIFT table and the PREFIX table are both 256*256. let all the SHIFT value be m-B+1 firstly, where "m" is the shortest length of the patterns and "B" is the length of the block. For each block, if it occurs in some patterns, we mark the leftest position of its head. We set the SHIFT value as the position, if the value of the position is less than the corresponding SHIFT value of the block. The PREFIX value, if the corresponding SHIFT value is zero, is the pointer of the corresponding state in the AC machine.

It is supposed that we use pattern strings which are "action", "section" and "sector" text to match the following string:"... disk sector buffer ..."

The table of SHIFT is as follows:

TABLE 1. THE TABLE OF SHIFT

	SHIFT		SHIFT		SHIFT
ac	0	ct	mini(1,2)	ti	mini(2,3)
io	mini(3,4)	on	mini(4,5)	se	0
ec	1	to	3	or	4
ELSE	5				

The figure of the AC automaton model is as follows:

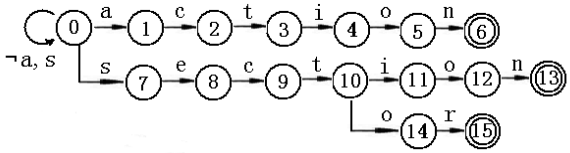

Figure 2. The AC automaton model.

The flowchart of the algorithm is portrayed in Figure. 3. Figure. 4 shows the process of the algorithm. The patterns are "action", "section" and "sector", and the matched string is "… disk sector buffer… ". Firstly, align the matching window to the rightest 6 characters. Secondly we can find that the SHIFT value of the head block "bu" is 5, so move the matching window 5 characters to the left. Recursive the above process until the SHIFT value of "se" is 0, as depicted in Figure. 4(d). Then align the head of the AC automation to the left of the matching window, match the pattern like the AC algorithm, and we can find pattern "sector" matched, which is shown in Figure. 4(e). After this, move the window to the left by one character and recursive the process until the text is exhausted.

The average complexity of the AC-WM algorithm is also the O (BN / m), where the "B" is the length of the character block, while the N is the length of the text, and the "m" is the shortest length of the patterns.

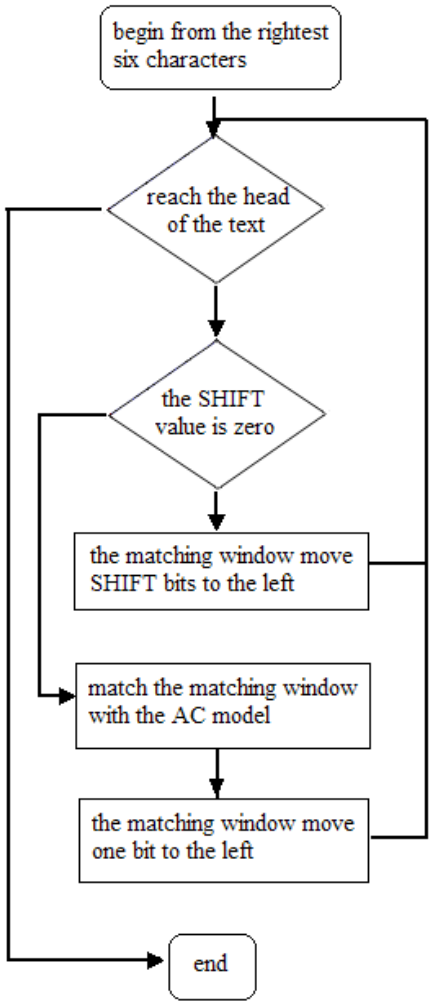

Figure 3. The flowchart of the matching.

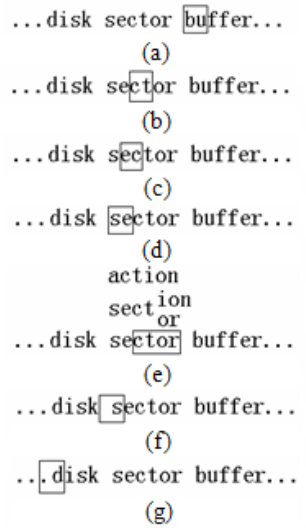

Figure 4. The process of the matching.

V. Performance Testing And Analysis

In order to test the performance of the improved algorithm, the WM algorithm and AC-WM algorithm which are programmed and tested in the environment of Microsoft Visual Studio 2005. The concrete experimental environment is as follows: the operation system is the windows XP, the dominant frequency of the system is 1.73GHz, the processor is the Intel Pentium Dual T2370, and the memory is 1GB.

The character set of the experimental data is the ASCII set, which size is 256, and the length of text is 1MB.In this paper, the common pattern strings are extracted from the text, and some pattern strings which have the same suffix are also randomly generated. In the experiment, the length of character block is 2.

The experimental results are shown in Figure. 5.

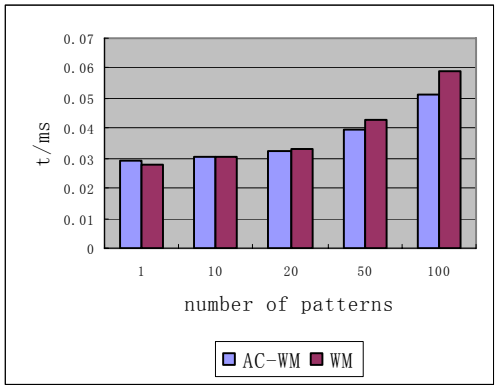

Figure 5. Performance testing.

As shown in Figure 4, the AC-WM algorithm has more advantages compared to the W-M algorithm in the case of more pattern strings.

VI. Conclusion

The AC-WM algorithm is aimed at reducing the time of matching; it improved the efficiency of pattern matching. But it still doesn't overcome the shortcomings of the BM algorithms which are highly dependent on the shortest length of patterns. If it combines with the other advantages of WM algorithms, the matching efficiency can be further improved, but it also adds the complexity of the algorithm in some degree.

On the one hand, we have higher requirements to the efficiency of pattern matching algorithms with the rapid increase of information. Therefore, we devoted to researching the better algorithm in order to meet our needs. On the other hand, we will further expand the application domain of pattern matching algorithm.

ACKNOWLEDGMENT

This Project is supported by the Planning Project of the National Eleventh-Five Science and Technology (2007BAK34B04) and the Chinese National Natural Science Fund (60704042) and Aeronautical Science Foundation (20080768004) and the Program of 211 Innovation Engineering on Information in Xiamen University (2009-2011).

REFERENCES

[1] D.E. Knuth, J.H. Morris, and V.R. Pratt. “Fast pattern matching in strings,” TR CS-74-440, Stanford University, Stanford, California, 1974

[2] A.V. Aho, and M.J. Corasick. “Efficient string matching: an aid to bibliographic search,” Communications of ACM, vol.18, no.6, pp.333-340, 1975

[3] R.S. Boyer, J. S. Moore. “A fastest ring searching algorithm,” Communications of the ACM, vol.20, no.10, pp.762-772, 1977

[4] S. Wu, U. Manber. “A fast algorithm for multi-pattern searching,” Report TR-94-17, Department of Computer Science, University of Arizona, Tucson, AZ, 1994

[5] R.N. Horspool. “Practical fast searching in strings,” Software-Practice and Experience, vol.10, pp.501-506, 1980

[6] G. Navarro, M. Raffinot. Flexible Pattern Matching in Strings. United Kingdom : Cambridge University Press, 2002.

[7] H. Jiang, Y. Zhang. “An improved W-M algorithm for multi-pattern match,” Mechanical & Electrical Engineering Magazine, vol.25, no.9, pp.25-27, 2008

[8] Y. Chen, G. Chen. “The performance analysis of wu-manber algorithm and its improvement,” Computer Science, vol.33, no.6, pp.203-209, 2006

[9] D. Yang, K. Xu, Y. Cui. “Improved Wu-Manber multiple patterns matching alg orithm,” Journal of Tsinghua University (Science and Technology), vol.46, no.4, pp.555-558, 2006

[10] X. Sun, Q. Wang, Y. Guan, X. Wang. “An improved Wu-Manber multiple-pattern matching algorithm and its application,” Journal of Chinese Information Processing, vol.20, no.2, pp.47-52, 2006

[11] W. Ma, Y. Liu, F. Ye, X. Yang. “An improved Wu-Manber multiple patterns matching algorithm,” Applied Science and Technology, vol.34, no.10, pp.32-34, 2007

[12] Q. Huang. “Application of identifying data packets basing on improved AC_BM algorithm,” Software Guide, 8(1): pp.54-56, 2009

[13] Y. Qian, Y. Hou. “A fast string matching algorithm,” Mini-micro Systems, vol.25, no.3, pp.410-413, 2004