ISSN:0975-9646

Importance of Aho-Corasick String Matching Algorithm in Real World Applications

Saima Hasib, Mahak Motwani, Amit Saxena

Truba Institute of Engineering and Information Technology
Bhopal (M.P), India

Abstract: String matching problem is to find all the occurrences of a given string pattern in a large string text. This problem is fundamental in computer Science and is the basic need of many applications, one of the most popular Multipattern String Matching Algorithm is "Aho-Corasick"; based on constructing DFA (Deterministic Finite Automata) between pattern characters, It is an exact matching algorithm. In this paper we will discuss the working of "Aho-Corasick" algorithm with its advantages, disadvantages and various application areas like intrusion detection, detecting plagiarism, bioinformatics, digital forensic, and text mining etc.

Keywords: Aho-Corasick, intrusion detection, plagiarism, bioinformatics, digital forensic, text mining.

I. INTRODUCTION

In String Matching Algorithms we try to find the position where patterns are found within a larger string or text. String matching can be performed in the Text String through Single Pattern and Multiple Pattern occurrences. Multiple pattern matching provides solution concept in many applications. The Aho–Corasick is one of the string matching algorithms, invented by Alfred V. Aho and Margaret J. Corasick. For finding Multipattern occurrences in the text string this algorithm is more appropriate because it performs exact matching of the patterns in the text. It seems to be like dictionary-matching algorithm which starts finding pattern on the basis of sub-string matching, each time character of pattern string is read and it tries to find the transition of that character in the already constructed automata, after reading the whole pattern string if the automata found to be entered in the final state so the pattern occurrence will be reported. Similarly it matches all patterns simultaneously. This algorithm can be applied to solve various problems like intrusion detection, detecting plagiarism, bioinformatics, digital forensic and text mining etc. Intrusion Detection is a technique in which intrusions are detected by Intrusion Detection System (IDS).Plagiarism Detection is process of finding plagiarism within a work or document. Bioinformatics is the application of computer technology to the management of biological information. Digital Forensic is a method for retrieving information from digital devices after being processed and generates some result. Text mining or Text Data Mining is the process that attempts to discover patterns in large data sets.[1,2,3,4,5,6,7,8,9,10]

II. AHO CORASICK ALGORITHM

Aho Corasick is the Multipattern matching algorithm which locates all the occurrence of set of patterns in a text of string. It first creates deterministic finite automata for all the predefined patterns and then by using automaton, it processes a text in a single pass. It Consists of constructing a finite state pattern matching automata from the patterns and then using the pattern matching automata to process the text string in a single pass.[9,10]

1. AHO-CORASICK EXAMPLE

a. Aho-Corasick Preprocessing Phase:

Example: Suppose we have a finite set of patterns {WOMAN, MAN, MEAT, and ANIMAL}. Aho corasick algorithm first creates finite automata for set of patterns.

Automata: for Patterns Set= {WOMAN, MAN, MEAT, ANIMAL}

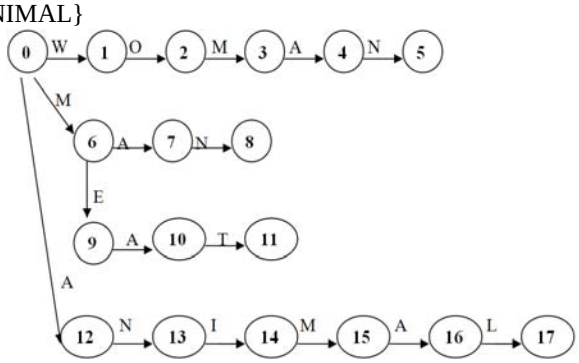

Figure1: Automata

STATE	INPUT								
	W	0	M	A	N	E	T	I	L
0	1	–	–	–	–	–	–	–	–
1	–	2	–	–	–	–	–	–	–
2	–	–	3	–	–	–	–	–	–
3	–	–	–	4	–	–	–	–	–
4	–	–	–	–	5	–	–	–	–
5	–	–	–	–	–	–	–	–	–
6	–	–	–	–	–	9	–	–	–
7	–	–	–	–	8	–	–	–	–
8	–	–	–	–	–	–	–	–	–
9	–	–	–	10	–	–	–	–	–
10	–	–	–	–	–	11	–	–	–
11	–	–	–	–	–	–	–	–	–
12	–	–	–	–	13	–	–	–	–
13	–	–	–	–	–	–	–	14	–
14	–	–	15	–	–	–	–	–	–
15	–	–	–	16	–	–	–	–	–
16	–	–	–	–	–	–	–	–	17
17	–	–	–	–	–	–	–	–	–

Figure2: Transition Table of Automata

FAILURE FUNCTION:
Failure function can be defined as the longest suffix of the string that is also the prefix of some node. The goal of the failure function is to allow the algorithm not to scan any character more than once.

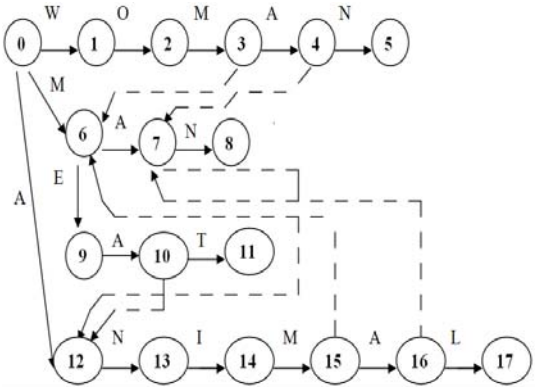

Figure3: Failure function transitions

NODE	FAILURE
0	0
1	0
2	0
3	6
4	7
5	0
6	0
7	12
8	0
9	0
10	12
11	0
12	0
13	0
14	0
15	6
16	7
17	0

Figure4: Failure function table

OUTPUT FUNCTION:
Output function gives the set of patterns recognized when entering final state.

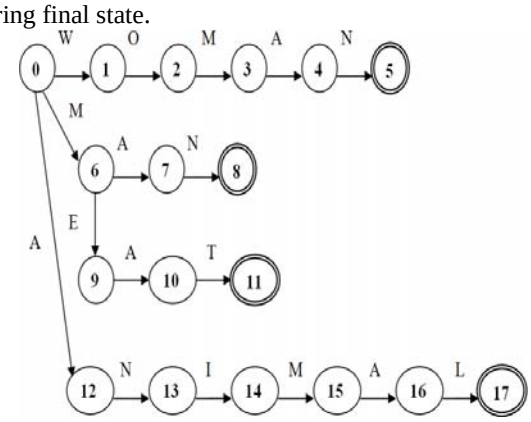

Figure5: Output Function transitions

FINAL STATE	OUTPUT
NODE 5	WOMAN, MAN
NODE 8	MAN
NODE 11	MEAT
NODE 17	ANIMAL

Figure6: Output Function table

b. Aho-Corasick Searching Phase: the searching phase of aho corasick is straightforward while scanning the text it walk through automata if any transition found, it get transition, otherwise check the failure function.
Text: WOMANETIMEAT

STATE	CHARACTER	TRANSITION	FAILURE	COMMENT
0	W	0→1	-	TRANSITION FOUND
1	O	1→2	-	TRANSITION FOUND
2	M	2→3	-	TRANSITION FOUND
3	A	3→4	-	TRANSITION FOUND
4	N	4→5	-	TRANSITION FOUND
5	E	-	0	NO TRANSITION FOUND
0	T	-	0	NO TRANSITION FOUND
0	I	-	0	NO TRANSITION FOUND
0	M	0→6	-	TRANSITION FOUND
6	E	6→9	-	TRANSITION FOUND
9	A	9→10	-	TRANSITION FOUND
10	T	10→11	-	TRANSITION FOUND

Figure7: Searching transition table

2. ALGORITHM

Preprocessing Phase

Step 1: Construct finite state automata for the set of predefined patterns (or pattern tree) which are supposed to be found in the text string. The states will be numbered by their names and transitions between the defined states would be represented by the characters existing in the particular pattern.

Step 2: After constructing automata, failure function of each node is calculated and its corresponding transitions are also required to be mentioned, so the constructed automata would be called as "Automata with failure links".

Step 3: Lastly in the automata output function for final states has to be calculated for recognizing the pattern string which may be found in the text string. And the resulting automata would be called as “Automata with Output Functions”

Searching Phase

By using the Aho Corasick Searching Algorithm search the text using the pre constructed Finite State Automata for the set of predefined patterns.

III. APPLICATIONS OF AHO-CORASICK

1. Intrusion Detection: Security has become an issue while dealing with computer networks. Hackers and intruders try to bring down networks and web services. For securing the communication over the Internet some methods have been already proposed like use of firewalls, encryption and virtual private networks etc. By intrusion detection methods we are able to find the kind of attack which is being done on the network or host on the basis of collecting information regarding known attacks. The methods detect suspicious activity on network and host level. Intrusion detection systems are categorized like signature-based intrusion detection systems and anomaly detection systems. Intruders have signatures, like computer viruses, that can be detected using software. Snort is an open source IDS available to the general public. NIDS are intrusion detection systems that capture data packets travelling on the network media (cables, wireless) and match them to a database of signatures. The intrusion detection systems make use of the Aho-Corasick Algorithm which is an automaton based multiple string matching algorithms which find all the occurrences of patterns in a text string. It first builds a finite state machine of all the keywords in a string and then uses the machine to process the text in a single scan. [2, 3, 8]

2. Detecting Plagiarism: Plagiarism Detection is process of finding plagiarism within a work or document. Plagiarism is the act of copying the idea of someone else and representing that as our thought and it's really a severe problem which is being faced in the world now a day, so it is need to be always noticed through the process of detection. There are so many algorithms have been proposed for plagiarism detection. Many types of plagiarism is studied like "Copy & Paste Plagiarism", "Word Switch Plagiarism", "Style Plagiarism", "Metaphor Plagiarism", " Idea Plagiarism".[10]

3. Bioinformatics: Bioinformatics is the study of biological science which deals with the methods of storing, retrieving and analyzing biological data, such as nucleic acid (DNA/RNA) and protein sequence, structure, function and genetic interactions. Bioinformatics is related domain of information technology and computer science engineering for dealing with biological problems, usually to resolve the issues introducing in genetic sequences. One of the applications of String matching algorithms is dealt for finding biological sequence information. Approximate matching of a search pattern in the text string is a basic need in molecular biology. Pattern is called as "query" and text is called "sequence database", but we will use "pattern" and "text" consistent with usage in computer science. While exact string matching is more commonly used in computer science, it is often useful in biology. One reason for this is that biological sequences are experimentally determined, and may include errors.[4,5]

4. Digital Forensics: A **digital signature** or **digital signature scheme** is a mathematical scheme for demonstrating the authenticity of a digital message or document. A valid digital signature gives a recipient reason to believe that the message was created by a known sender, and that it was not altered in transit. Digital signatures are commonly used for software distribution, financial transactions, and in other cases where it is important to detect forgery or tampering. Digital forensic1 text string searches are designed to search every byte of the digital evidence, at the physical level, to locate specific text strings of interest to the investigation. Given the nature of the data sets typically encountered.[6,7]

5. Text mining: Text Data Mining similar to Text Analysis refers to the process of deriving high-quality information from text. Deriving patterns within the structured data, and finally evaluation and interpretation of the output. [7]

IV.CONCLUSION

As we have observed that Aho-Corasick algorithm is best suited for multiple pattern matching and it can be used in many application areas, but it has been observed that as the size of automata increases drastically the performance of algorithm degrades in terms of time and space both. The complexity of the algorithm is linear in the length of the patterns plus the length of the searched text plus the number of output matches. It is found to be attractive in large numbers of keywords, since all keywords can be simultaneously matched in one pass. Aho-Corasick provides solution to many real world problems like Intrusion detection, Plagiarism detection, bioinformatics, digital forensic, text mining and many more. Aho-Corsick is one of the most fruitful algorithm in computer science.

REFERENCES

[1] Thomas H Corman, Charles E. Leiserson, Ronald L.Rivest & Clifford Stein "Introduction to AlgorithmsString matching", IEEE Edition, 2nd Edition, Page no .906-907.

[2] Ali Peiravi, "Application of string matching in Internet Security and Reliability", Marsland Press Journal of American Science 2010, 6(1): 25-33.

[3] Peifeng Wang , Yue Hu, Li Li, "An Efficient Automaton Based String Matching Algorithm and its application in Intrusion Detection", International Journal of Advancements in Computing Techology(IJACT), Vol 3, Number 9 , October 2011.

[4] Pekka Kilpelainen, "Set Matching and Aho-Corasick Algorithm", Biosequence Algorithms, Spring 2005, BSA Lecture 4.

[5] Robert M. Horton, Ph.D. "Bioinformatics Algorithm Demonstrations in Microsoft Excel" , 2004 - cybertory.org

[6] Nicole Lang Beebe, Jan Guynes Clark, "Digital forensic text string searching: Improving information retrieval effectiveness by thematically clustering search results", d i g i t a l i n v e s t i g a t ion 4 S (2 0 0 7).

[7] Beebe NL, Dietrich G. "A new process model for text string searching". In: Shenoi S, Craiger P, editors. Research advances in digital forensics III. Norwell: Springer; 2007. p. 73–85.

[8] Rafeeq Ur Rehman , "Intrusion Detection Systems with Snort Advanced IDS Techniques Using Snort Apache, MySQL, PHP, and ACID" page 348-351.

[9] Xinyan Zha and Sartaj Sahni "Multipattern String Matching On A GPU",IEEE,2011,pp. 277-282

[10]Ramazan S. Aygün "structural-to-syntactic matching similar documents", Journal Knowledge and Information Systems archive, Volume 16 Issue 3, August 2008.