ISSN (Print) : 2320 – 9798
ISSN (Online): 2320 – 9801

International Journal of Innovative Research in Computer and Communication Engineering
Vol. 1, Issue 3, May 2013

Design and Implementation of a Pattern Recognition Algorithm

S. Manikandan, B.Annapoorani, C. Arul Murugan

Assistant Professor, Department of ECE, Karpagam College of Engineering, Coimbatore, India

.

Abstract: Pattern matching problem has attracted a lot of interest throughout the history of image processing i.e. finger print recognition, face recognition, computer science, image processing ,character recognition, signature recognition, speech recognition, text processing, information retrieval and computational biology. Pattern matching also plays an important role in the field of information security, for example, in signature-based IDs and virus pattern matching. Here we demonstrate a method can be devised from theoretical analysis by extensive experimentation and modification of existing algorithms. It works consistently well for both nucleotides and amino acids sequences. The result shows the robustness of the proposed algorithm and thus can be incorporated in exact pattern-matching application involving biological sequences.

I. INTRODUCTION

Pattern matching can be defined as finding the occurrence of particular pattern of characters in a large junk of text. An exact pattern matching involves identification of all the occurrences of a given pattern of m characters (x=x1, x2, x3……xm), in a text of n characters (y=y1, y2,y3…….yn), built over finite alphabet set ∑ of size σ[1].

Pattern matching problem has attracted a lot of interest throughout the history of image processing i.e. finger print recognition, face recognition, computer science, image processing ,character recognition, signature recognition, speech recognition, text processing, information retrieval and computational biology. Pattern matching also plays an important role in the field of information security, for example, in signature-based IDs and virus pattern matching. To solve this problem, a simplified representation of the problem and powerful pattern matching techniques are the two ways used in improving efficiency and performance. On the other hand, threats like virus and worms which can replicate and unleash automatically are always quite serious and out of control by time they have are discovered. Real time virus detection is a proper way to defeat them. Also pattern matching is prerequisite for information retrieval. Modern technology has made it possible to produce, process, store and transmit document images efficiently. In an attempt to move towards the paperless office, large quantities of printed documents are digitized and stored as images in databases. The popularity and importance of the document images as an information source are evident. As a matter of fact, many organizations currently use and are dependent on document image databases. However, such databases are often not equipped with adequate index information. This makes it much harder to retrieve user-relevant information from image data than from user data. Thus, the study of information retrieval in document image databases is an important subject in knowledge and engineering work.

Thus pattern detection plays an important role in document / image retrieval.

For pattern detection, many algorithms have been developed. These algorithms are applied in most of search engines on the internet, retrieval of information (from text, image or sound) and searching nucleotides/ amino acids sequence pattern in genome and protein sequence databases. Theoretical studies of different algorithms suggest various possible means by which these are likely to perform.

Pattern matching algorithms have been evolved drastically. However, the most naïve algorithm to solve this problem is to take the skip value as one. This simple approach is known is brute force method. After this basic approach, various algorithms were developed, which improved the efficiency and each had its own advantages and limitations.

Pattern-matching algorithms can be addressed in different ways. In particular, they are amenable to approach that range from the extremely theoretical to practical. On other hand, practical implementations of the algorithm, though hard to assimilate, have proved to be more beneficial and useful. Hence, it is extremely difficult to identify a suitable and preferable fast algorithm for a given database (because of the variation in the alphabet size; for proteins the alphabet size is 20, where as it is 4 for nucleotide sequence).

In addition, only a real-time practitioner can locate and use the most appropriate algorithm to perform a particular task. Given a situation with a Pattern-matching problem, amateur software engineers, computational biologists, researcher or students tend to dig out information through search engines(for example , Google).This approach will solve the problem, but does not guarantee that the chosen algorithm is an efficient one.

Many of pattern matching algorithms are implemented in two phases, viz pre-processing phase and searching phase. During pre-processing phase, the algorithm pre-processes the search pattern and generates the skip value that can be used in the search phase. The pre-processed skip value helps to reduce the total number of character comparison during the search phase, thereby reducing the overall execution time. Hence, the pre-processing phase aims at the operations in the search phase. However, the efficiency of an algorithm is mainly dependent on the methodology adopted in the search phase. The search phase can be improved by altering the order in which the characters are compared at each attempt testing the most appropriate skip value that maximizes the shift of the window on the text [6].

II. SURVEY ON THE EXISTING ALGORITHMS

The algorithms reported in the literature are ranked based on their average-case and worst-case time complexities. The Boyer-Moore algorithm and its variants are widely used in the software industry. The algorithm, Quick-search, performs better, when the pattern length is small and the alphabet size is large (which is true in most of the practical situations). In the Raita algorithm, the order of the comparison is modified to attain maximum efficiency. The Horspoolalgorithm performs the comparison in a simple way, which works for most of the practical cases[1].

THE BOYER–MOORE ALGORITHM is widely used in the software industry. This algorithm uses Boyer–Moore bad character (bmBc) and Boyer–Moore good suffix (bmGs) tables to determine the number of shifts required to slide the window on the text, so that no match is left unconsidered. The maximum shift value from both the tables is considered for the shift after each attempt in the searching phase[3].

HORSPOOL ALGORITHM uses a shift value by finding the bad character shift for the rightmost character of the window. The shift value is computed in the pre-processing stage for all the characters in the alphabet set. Thus, the algorithm effective in practical situations where the alphabet size is large and the length of the pattern is small[4].

IN RAITA ALGORITHM**,** the order of comparison is modified to attain maximum efficiency. Here, the rightmost characters of the pattern and the window are compared, and on an exact match, the leftmost character of the pattern and that of the window are compared. If they match, the algorithm compares the middle characters of both the pattern and the window, and then the characters from the second to the last but one position of the pattern and the window are compared[5].

IN QUICK SEARCH ALGORITHM, the Quick Search bad character (qsBc) shift table is used to store the shift value of each character in the pattern. The shift value is given by the corresponding position of that character in the pattern from right to left. A shift value of $(m + 1)$ is assigned to the characters which are not present in the pattern. In the searching phase, character comparisons between the text and the pattern can be done in any order.
[1]

III. PROPOSED ALGORITHM

The idea behind the proposed algorithm is to find sum of absolute difference value.
The algorithm has two phases

A) PREPROCESSING PHASE: In this, periodic of pattern is obtained so that pattern length i.e. window size is equal to text size.

B) SEARCHING PHASE: In this phase, absolute difference of binary value of each alphabet of text and pattern is obtained for the length of Pattern sequence, then the summing absolute difference and comparing with zero. If matched then pattern is detected. The sum is obtained and stored in memory, wherever sum is zero, noting down the position at which pattern is present in the text. Then the window is shifted by one and procedure is repeated. The number of shifts required is equal to the length of pattern sequence.

Thus proposed algorithm drastically reduces number of shift required. All mentioned algorithm applied only to uniprocessor but the proposed algorithm can be applied to both uniprocessor as well as multiprocessor .The proposed algorithm is more efficient on multiprocessor which reduces the total time required to detect the total number of patterns present in the text unlike other algorithm.

```
METHODOLOGY:
Step1: Read the text[DNA sequence]
Step2: Read pattern list[motifs list] to be detected
         in text.
  {Total number of patterns    P1……………..Pn
      Where   P1=p11, p12………………….
                 P2= p21, p22,
                     •
                     •
                     •

            Pn=pn1, pn2……………..
   }
Step3: For n=1 to total_num_pattern sequence
          {
           For m=1 to total_num_elements for $n_{th}$ pattern      sequence.
              {
                  $\text{Number of cycles} = \frac{[\text{Length of text}]}{[\text{Length of element of nth pattern sequence}]}$

a)Generate periodic of element of $n_{th}$ pattern  sequence

    Pattern_length=(Number of cycles) *( length
                         of element   of $n_{th}$ pattern  sequence)

b) Find [store in an array] the absolute value  of difference  beween each alphabet  of    text sequence
 and periodic  pattern of  an element of $n_{th}$ pattern sequence )

c) Calculate the sum of the elements in the array
  sequentially for every N element ,where N=length          of element in the pattern sequence .If (sum==0) then note the
position of first element of  $p_{th}$ cycle corresponding to the zero sum.

d) Repeat the above step till the end of an array.

e) Repeat the steps from (c) with a relative position
    between periodic pattern and the text                    increased by  one till the relative position value
     equal to the length of element.
            }
}
```

IV. WORKING EXAMPLE

The plant genome (Arabidopsis thaliana) consists of 27,242 gene sequences distributed over five chromosomes (CHR-I to CHR-V) (NCBI site, fttp://ftp.ncbi,nih.gov/genomes/ Arabidopsis thaliana/CHR-I). part of a nucleotide sequence of a gene (only 15 nucleotides) from chromosome I(CHR-I)has been used (see below for details) to test the proposed algorithm.
Sequence in FASTA format, Part of the sequence considered for the test run.
Y=ATCTAACATCATAACCCTAATTGG
This sequence has been taken from the gene index 32854 to 32868
Gi\22330780/ref/NC_003070.3/ Arabidopsis thaliana chromosome 1, complete sequence
Y= ATCTAACATCATAACCCTAATTG

ISSN (Print) : 2320 – 9798
ISSN (Online): 2320 – 9801

International Journal of Innovative Research in Computer and Communication Engineering
Vol. 1, Issue 3, May 2013

X=CATC
N=15
M=4

PREPROCESSING PHASE

No-Periodic-cycle=15/4=3

Periodic of pattern generated are=
CATC CATC CATC

Now,Total number of alphabets in pattern =15

Total number of shifts required in this example=M=4

SEARCHING PHASE:

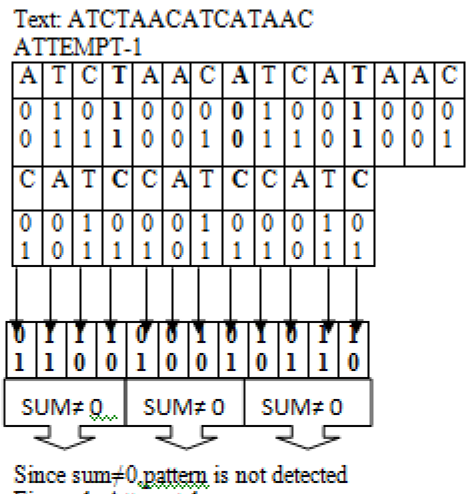


Figure 1: Attempt-1

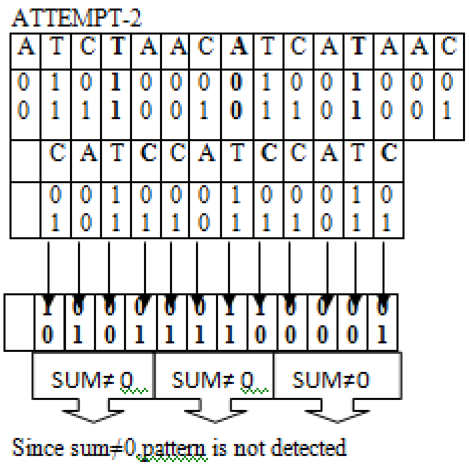
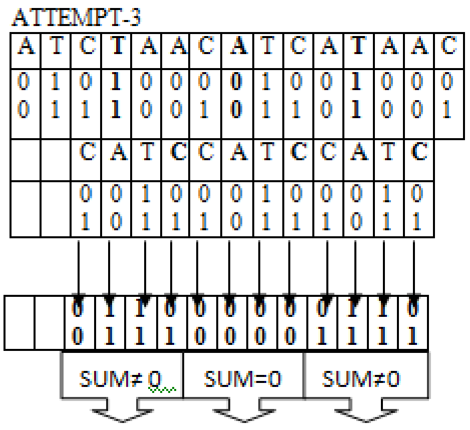
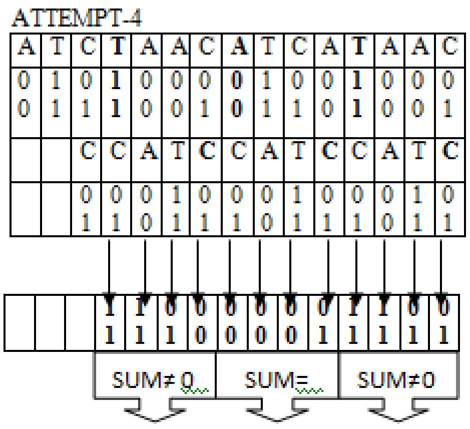


Figure 2: Attempt-2



Since sum=0 pattern is detected
Figure 3: Attempt-3



Since sum≠0 pattern is not detected

Figure 4: Attempt-4

TABLE I
COMPARISON TABLE

Algorithm	Time complexity for pre-phase	Time complexity for Searching phase
Proposed algorithm	O(n/m)	O(mn)
Brute-force	-	O(mn
Boyer-Moore	O(m+σ)	O(mn)
Quick search	O(m+σ)	O(mn)
Horspool	O(m+σ)	O(mn)
Raita	O(m+σ)	O(mn)

Table 1: Comparison of different algorithm

ISSN (Print) : 2320 – 9798
ISSN (Online): 2320 – 9801

International Journal of Innovative Research in Computer and Communication Engineering
Vol. 1, Issue 3, May 2013

V. CONCLUSION

In this paper, we proposed a new algorithm for exact pattern matching by defining a new order of character-character comparisons between the window and text at each attempt and shifting the window by one.The new algorithm has been tested on large database.Hence it can be implemented in all applications related to exact pattern matching.

REFERENCES

(1) S. S. Sheik, Sumit K. Aggarwal, Anindya Poddar, N. Balakrishnan and K. Sekar, 'A FAST Pattern Matching Algorithm',Bioinformatics Centre and Supercomputer Education and Research Centre, Indian Institute of Science,Bangalore 560 012, India Received June 18, 2003 J. Chem. Inf. Comput. Sci. 2004, 44, 1251-1256 1251 10.1021/ci030463z CCC: $27.50 © 2004 American Chemical Society Published on Web 05/13/2004
(2)Ahmad Fadal Klaib and Hugh Osborne-'Searching Protein Sequence Database Using BRBMH Matching algorithm'.(IJCSNS International Journal of computer Science and Network security,Vol. 8 No-12, December 2008)
(3) Boyer, R. S.; Moore, J. S. 'A fast string searching algorithm'. Commun.ACM 1977, 20, 762-772.
(4) Horspool, R. N., Software – Practice Experience, 1980, 10, 501–506.
(5) Raita, T. Tuning the Boyer-Moore-Horspool' string-searching algorithm'.Software - Practice Experience 1992, 22(10), 879-884.
10.1021/ci030463z CCC: $27.50 © 2004 American Chemical Society Published on Web 05/13/2004
(5)Ahmad Fadal Klaib and Hugh Osborne-'Searching Protein Sequence Database Using BRBMH Matching algorithm'.(IJCSNS International Journal of computer Science and Network security,Vol. 8 No-12, December 2008)
(6) S. S. Sheik, Sumit K. Aggarwal, A.Poddar,B.sathiyabhama N. Balakrishnan and K. Sekar "analysis of string searching algorithms on biological sequence databases"-Bioinformatics center and supercomputer Education center,Indian Institute of science, Bangalore.