IOSR Journal of Engineering (IOSR JEN) **www.iosrjen.org**
ISSN (e): 2250-3021, ISSN (p): 2278-8719
PP 10-13

A Study of Pattern Recognition with Wildcard Algorithmic Rule

D.Ananthi
Assistant Professor in Computer Science, Annai Women's College, Karur.
sivananthi.sanju@gmail.com.

Abstract: *This paper expresses the fundamentals of Wildcard algorithmic rule and the way it's helpful for recognizing pattern. Pattern recognition could be a root of computer science and branch of machine learning that emphasizes the knowledge information patterns or data regularities during a given state of events. The single wildcard defines to match any one character in a sequence. The Multiple wildcards is defines to match more than one character in a sequence. The Multiple Wildcards form a gap. The length of a flexible gap is arbitrary. The objective of this paper is to summarize and compare some pattern recognition techniques included in Wildcard algorithms.*

Keywords: *Wildcard Characters, Wildcard Gap, Constraints, Matching.*

I. Introduction

Pattern recognition is the study of how machines can observe the environment and learn distinguishes patterns from their background to make reasonable decisions about the patterns. In Computer Science, Pattern matching refers to the method of checking and locating the occurrences of the pattern in a sequence. Output of pattern matching is the total number of occurrences of pattern P in a sequence S and all possible locations of a pattern P within a sequence S. In this work, we consider one main string and another wildcard patterns. Now we are going to check whether the wildcard pattern is matching with the main text or not.

II. Methodology

Pattern Matching Techniques

Pattern matching algorithms can be broadly classified into two main categories:

1. Single Pattern Matching

To search a single pattern is presence in a sequence.

2. Multi Pattern Matching

More than one pattern is searched simultaneously for presence in a sequence. It provides high performance security and usability than single pattern matching.

PATTERN MATCHING WITH WILDCARD CHARACTERS

Wildcard refers to the special character that can be replaced by zero or more characters in a string. Wildcards are mostly used in regular expressions, SQL queries, Dictionary navigation etc. It should cover the entire text not partial text in a sequence. It includes the characters '?' and '*'.

1. **'?'** – Is used to match a single character in a sequence.
2. '*' – is used to match the sequence of characters including the empty sequence.

Case 1:

(i) When the character is '*': We can ignore the '*' character and move to check next characters in the pattern.

(ii) '*' character matches with one or more characters in Text. Here we will move to next character in the string.

Case 2:

When the next character is '?', then we can ignore only the current character in the text, and check for the next character in pattern and text.

Case 3:

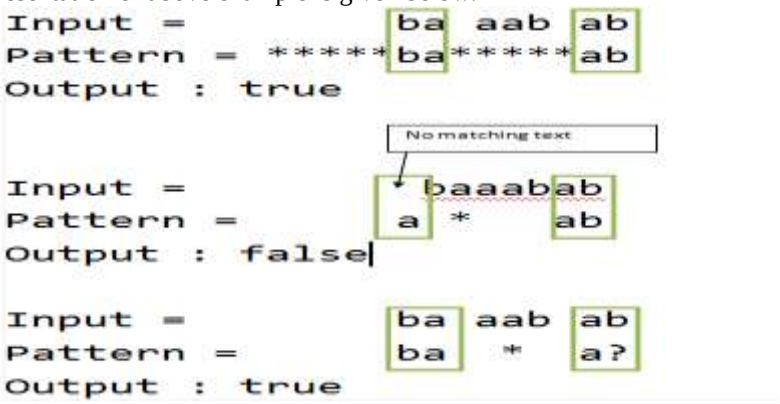
The character is not a wildcard character, if current character in Text matches with current character in Pattern, we move to next character in the Pattern and Text. If they do not match, wildcard pattern and Text do not match.

Example:

Text = "baaabab",

1. Pattern = "*****ba*****ab", Output: true
2. Pattern = "baaa?ab", Output : true
3. Pattern = "ba*a?", Output : true
4. Pattern = "a*ab", Output: false

Diagrammatical representation of above example is given below:



Constraints

Various constraints that can be related to the pattern matching with wildcard gaps are as follows:

Fixed wildcard gap: Fixed wildcard gap mean that the number of wildcard characters that can occur in pattern are fixed. While matching with a string, these wildcard characters can be replaced by any character from the alphabet.

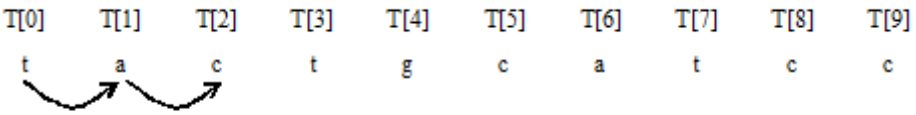
Variable wildcard gap: Variable wildcard gap means that the number of wildcard characters between two consecutive characters can be a range rather than a fixed number.

Local length constraints: It is the constraint in the form of the range of length of wildcard characters between each two consecutive letters of the pattern. This gives flexibility to control queries.

Global length constraints: Global length constraint is the constraint on the overall length of each matching substring of sequence with the given pattern.

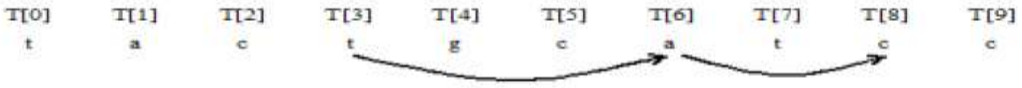
Example:

The following figure (a) shows there are no wildcard characters in the pattern. In this case jump from one character to the next character is consecutive while searching for the match.



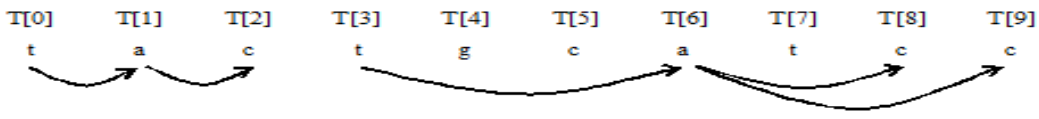
a) When pattern P = tac

The following figure (b) shows there is fixed wildcard gap in the pattern i.e. number of wildcard characters are fixed. In this case there is constant jump from one character to the next character while searching for the match.



b) When pattern P = tΦΦaΦc

The following figure (c) shows the case when there is variable wildcard gap in the pattern i.e. number of wildcard characters between two consecutive characters of the pattern is a range. In this case there is flexible jump from one character to the next character depending on the range while searching for the match.



c) When pattern P = t[0,3]a[0,2]c

One-Off Condition

In addition to the global constraint, the concept of one-off condition was taken into consideration in some algorithms. One-off condition means every positional index of a character in a sequence can be used at most once while matching with the given pattern. Applying One-off condition makes the solution to removes useless information.

Example:

(i) SAIL algorithm: It consumes a lot of time for the large pattern length. Three main steps involved in this algorithm are:

1. Location: It searches for the position of the last alphabet of Pattern in Sequence by considering the global constraint.

2. Forward: This phase eliminates all those solutions that do not satisfy local constraints and gives the underlying matching positions.

3. Backward: This phase selects one optimal solution out of all possible solutions.

(ii) RSAIL algorithm: A SAIL has a problem of left-optimization as it chooses the left-most letters. To eliminate this problem with SAIL algorithm, RSAIL was proposed. The idea behind RSAIL is as follows:

1. If pattern is not having recurring tail characters, SAIL is called.
2. If pattern is having recurring tail characters, convert it into a pattern having no recurring tail characters and call SAIL.

III. Comparison Of Various Algorithms Involving One-Off Condition

Various algorithms to solve the problem of maximal pattern matching with length constraints and one-off condition have been compared in Table 1.1 on the basis of data structure, time and space complexities.

Meaning of various symbols used in the comparison table is as follows:

n – Length of the sequence
m – Length of the pattern
f - Frequency of occurrence of pattern's last character in the sequence
W - Maximum gap between consecutive letters of the pattern
l – Maximum allowed length of the occurrence (Global Length Constraint)
c - Number of parts into which sequence is divided
num – total number of occurrences of the pattern in a sequence
α – Total number of occurrences of sub patterns in a sequence
A – Sum of lower limits of gap range
B - Sum of upper limits of gap range
s – Number of sub patterns in a pattern
L – Number of characters in the last sub pattern of the pattern
B/w – Number of machine words to store each bit mask

Table.1: Comparison of Algorithms

Algorithm	Data Structure	Time Complexity	Space Complexity
SAIL	Search Table	O(n+flmW)	O(lm)
RSAIL	Search Table	O(n + flmW)	O(lm)
PST	Suffix Tree	O(n+m+num+n/c)	O(2n/c)
BPBM	Non-deterministic Finite automata	O((Bm+n+f(l+s − 1))(B/w))	O((m+L+2s + 4) (⌊B/w⌋))
PMW	Aho-Corasick Automation	O(m+n+ f(l+α))	O(m+A)
HSO	Nettree	O(Wn(n+m2))	O(Wmn)
WOW	WON-Net	O(Wmn+mn2) by LMO/RMO O(Wmn+mn3) by CMP	O(mn)

IV. Conclusion

In this work, we discuss different existing algorithms for solving the problem of maximal pattern matching with flexible wildcard gaps, length constraints under the one-off condition and various pattern matching techniques have been studied along with their merits and de-merits. These algorithms are then compared on the basis of data structure used by them, technique incorporated in the algorithm, time and space complexities.

References

[1]. Moreau, P.E., Ringeissen, C., Vittek, M.: A Pattern Matching Compiler for Multiple Target Languages. In: In Proc. of Compiler Construction (CC), volume 2622 of LNCS. (2003) 61–76.
[2]. Oderksy, M., Wadler, P.: Pizza into Java: Translating theory into practice. In: Proc. of Principles of Programming Languages (POPL). (1997).
[3]. Zenger, M., Odersky, M.: Extensible Algebraic Datatypes with Defaults. In: Proc. of Int. Conference on Functional Programming (ICFP). (2001).
[4]. G. Navarro, "Flexible pattern matching," in Journal of Applied Statistics. Citeseer, 2002.
[5]. S. Neuburger, "Pattern matching algorithms: An overview," 2009.
[6]. R. Bhukya and D. Somayajulu, "Exact multiple pattern matching algorithm using dna sequence and pattern pair." International Journal of Computer Applications, vol. 17, 2011.
[7]. S. Wu and U. Manber, "Fast text searching: allowing errors," Communications of the ACM, vol. 35, no. 10, pp. 83–91, 1992.