Multiple Pattern String Matching

Raphael Ribeiro

Supervisor: Carlos E. ferreira

1 Introduction

An **alphabet** Σ is a finite set. A **symbol** is an element $s \in \Sigma$. A **string** is a finite sequence of symbols, i.e, elements of an alphabet. The set of all strings over Σ is denoted by Σ^*. We say a string $p \in \Sigma^*$ is a **pattern** over a fixed alphabet Σ if p consists of symbols from Σ. Let s be a string and denote $s := s_1 \cdots s_n$. We say that a subsequence $s_i \cdots s_j$ of s is a **substring** of s. A **ocurrence** of u in s is a pair (i, j) such that $u := s_i \cdots s_j$ is a substring of s.

Let P be a set of patterns over a fixed alphabet Σ and let T be a fixed string. The Multiple Pattern String Matching (MPSM) is the problem of finding all occurrences of all patterns of P in T. The Single Pattern String Matching (SPSM) is a special case of (MPSM) by adding the constraint $|P| = 1$.

The MPSM is one of the basic string algorithms problems and several algorithms have been proposed to solve it. There are practical solutions to real-world problems that can be developed using these algorithms, including, but not limited to, intrusion detection systems, evolutionary biology, computational linguistics, and data retrieval.

2 Objectives

- Provide theoretical results of several algorithms proposed to solve the MPSM problem such as the Finite States Machines and Pattern Matching Machine introduced by [**1**] to develop the Aho-Corasick Algorithm, the Shifting Technique, and Boyer-Moore algorithm [2], the Commentz-Walter algorithm which combines the shifting technique and a trie variant (Pattern Tree) [3], and the Wu-Manber algorithm which uses a variant of the shifting technique called Bad Caracter Shifting [6]
- Study of variants of these algorithms such as the Set Backward Oracle Matching Algorithm which uses a trie data structure [5], and the Set Horspool which is a simple variant derived from the Commentz-Walter algorithm [5].
- Study of pattern preprocessing with Q-grams and its applications to the algorithms [4]
- Implementation of all the algorithms and data structures, as described in the Work Plan.

- Provide a comparative analysis of the performance of all algorithms and their different implementations.

3 Work Plan

1. Aho-Corasick Algorithm, as described in [1]
2. Aho-Corasick variant: Set Backward Oracle Matching Algorithm, as described in [5]
3. Boyer-Moore Algorithm, as described in [2]
4. Commentz-Walter Algorithm, as described in [3]
5. Commentz-Walter variant: Set Horspool, as described in [5]
6. Wu-Manber Algorithm, as described in [6]
7. Pattern Preprocessing with Q-Grams, as described in [4]
8. Comparative Analysis of Performance

Activity	Months							
	May	Jun	Jul	Aug	Sep	Out	Nov	Dec
1.	x	x						
2.			x					
3.			x					
4.				x	x			
5.					x	x		
6.						x	x	
7.							x	
8.							x	x

References

[1] Margaret J. Aho Alfred V. Corasick. "Efficient string matching: an aid to bibliographic search." In: *Communications of the ACM* 2018 (1975).

[2] J. S. Boyer R. S. Moore. "A fast string searching algorithm." In: *Communications of the ACM* 20 (1977), pp. 762–772.

[3] Beate. Commentz-Walter. "A string matching algorithm fast on the average". In: *Automata Languages and Programming* 6 (1979).

[4] Jari Kytojoki Leena Salmela Jorma Tarhio. "A fast algorithm for multi-pattern searching." In: *ACM Journal of Experimental Algorithmics* 11 (2006), pp. 1–19.

[5] Raffinot M. Navarro G. *Flexible pattern matching in strings.* Cambridge University Press, Cambrige, UK., 2002.

[6] Udi. Wu Sun Manber. “A fast algorithm for multi-pattern searching.” In: *Technical Report* 94-17 (1994).