Efficient Regular Pattern Matching avoiding Denial of Service

Daniel Afonso de Resende
Mestrado em Segurança Informática
Departamento de Ciência de Computadores
2025

Orientador
Nelma Resende Araújo Moreira, Categoria
Faculdade de Ciências da Universidade do Porto

Coorientador
Nome do Orientador, Categoria
Faculdade de Ciências da Universidade do Porto

U.PORTO

FACULDADE DE CIÊNCIAS
UNIVERSIDADE DO PORTO

Todas as correções determinadas pelo júri, e só essas, foram efetuadas.

O Presidente do Júri,

Porto, ______/______/_________

MSc

Abstract

Hey, this is the abstract of my thesis. It should be a brief summary of the work, highlighting the main objectives, methods, results, and conclusions. The abstract should be concise and informative, allowing readers to quickly understand the essence of the research. **Keywords:** key, word.

Resumo

O teu resumo COOL, its me TEST WOWIEESSS

Palavras-chave: palavra, chave..

Acknowledgements

First of all, I would like to thank my family, etc, etc

Dedico à minha mãe . . .

Contents

List of Figures

Listings

Acronyms

Chapter 1

Introduction

In this chapter, the problem is overviewed, the study's importance is explained along with goals for the proposed solution.

1.1 Background

Despite recent advances in [1], ...

Chapter 2

Preliminaries

Theory builds upon theory, therefore it is essential to establish a solid foundation by understanding the basic concepts and terminology that compose the core topics of formal languages and automata theory. In this chapter we begin by formally defining what a language is and then move on to describe the class of languages known as regular languages. Along the way, we will also introduce various concepts such as finite/non-finite automata and regular expressions.

2.1 Alphabets, Strings and Languages

Alphabets

An *alphabet* is a finite, non-empty set of symbols, typically denoted by the Greek letter Σ. That is,

$$\Sigma = \{a_1, a_2, \ldots, a_n\}$$

where each a_i is a symbol in the alphabet.

For example, one can represent the binary alphabet as $\Sigma = \{0, 1\}$, or the English alphabet as $\Sigma = \{a, b, c, \ldots, z\}$.

Strings

A *string* over an alphabet Σ is a finite sequence of symbols from Σ. Strings are typically denoted by w, and the *length* of a string w is denoted by $|w|$.

The set of all strings over the alphabet Σ is denoted by Σ^* and defined as:

$$\Sigma^* = \{w \mid w \text{ is a finite sequence of symbols from } \Sigma\}$$

The unique string of length zero is called the *empty string*, denoted by ε. It is important to note that $\varepsilon \in \Sigma^*$.

For example, if $\Sigma = \{0, 1\}$, then we have that:

$$\Sigma^* = \{\varepsilon, 0, 1, 00, 01, 10, 11, 000, 001, 010, 011, 100, 101, 110, 111, \ldots\}$$

Where the empty string is, as mentioned above, denoted by ε and also belongs to Σ^*.

Languages

A *language* over an alphabet Σ is a set of strings over Σ.

$$L \subseteq \Sigma^*$$

That is, a language is any subset of Σ^*, possibly infinite, finite, or even empty.
Since a language is a set of strings, the following standard set operations can be applied:

- *Intersection*: $A \cap B = \{x \mid x \in A \text{ and } x \in B\}$
- *Union*: $A \cup B = \{x \mid x \in A \text{ or } x \in B\}$
- *Difference*: $A - B = \{x \mid x \in A \text{ and } x \notin B\}$

Furthermore, we can also operate specifically over languages with the following operations:

- *Concatenation*: $L_1 \cdot L_2 = \{xy \mid x \in L_1 \text{ and } y \in L_2\}$
- *Kleene Star*: $L^* = \bigcup_{n=0}^{\infty} L^n$, where $L^0 = \{\varepsilon\}$ and $L^n = L \cdot L^{n-1}$ for $n > 0$.

This operation combines every string

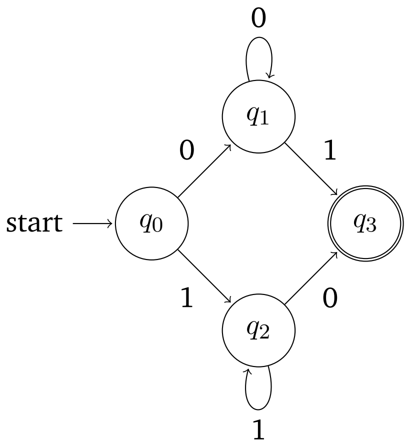
The *complement* of a language L over an alphabet Σ is denoted by $\overline{L}$ and is defined as: This

2.2 Regular Languages

2.3 Regular Expressions

2.3.1 Derivatives

2.4 Automata



2.4.1 Finite Automata

2.4.2 Non-finite Automata

2.4.3 Position Automata

Chapter 3

State of the Art

3.1 Overview of XYZ

Computers are devices that

Chapter 4

Matching

The implementation chapter gives insights into

4.1 Normal Matching

4.1.1 Standard Matching

4.1.2 Greedy Matching

4.2 Multi Matching

Chapter 5

Results and Discussion

This is a test

5.1 Algorithm Analysis

5.2 Accuracy

The methods of evaluating

5.3 Comparison with Other Methods

5.4 Examples

Chapter 6

Conclusion

6.1 Findings Summary

This research and development project served the objective of

6.2 Contributions

Bibliography

[1] Chen-Lin Lee. ‘Exploring the Introduction of Cloud Computing into Medical Information Systems’. In: *Journal of Computers* (2018) (cit. on p. 1).