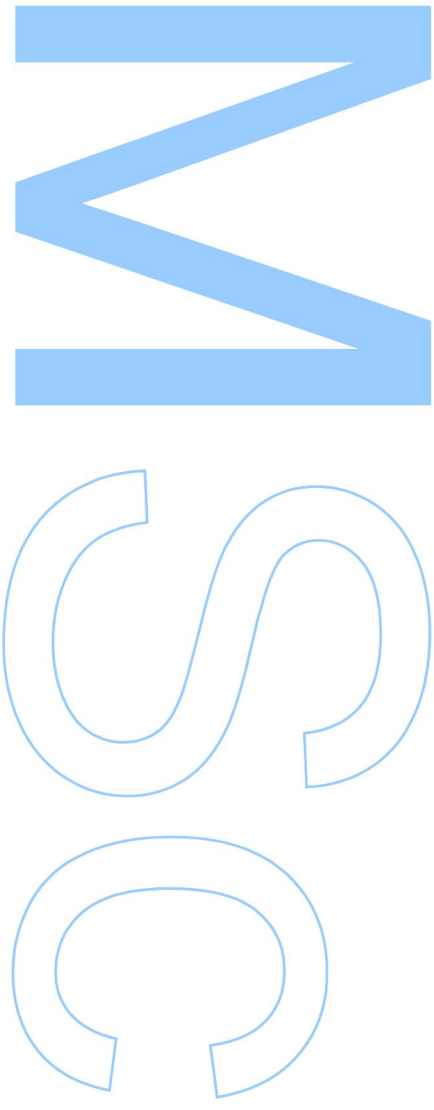
# Efficient Regular Pattern Matching avoiding Denial of Service



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


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

O Presidente do Júri,

Porto, _____/_____/_____

# 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

**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. It is usually denoted by the Greek letter $\Sigma$.

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

where each $a_i$ is a symbol.

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 $\Sigma$ is a finite sequence of symbols from $\Sigma$.

- The empty string (the string of length 0) is denoted by $\varepsilon$. - If $w$ is a string, then $|w|$ denotes the *length* of $w$. - The set of all strings (including the empty string) that can be formed from $\Sigma$ is denoted by $\Sigma^*$.

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

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 $\varepsilon$ and also belongs to $\Sigma^*$.

### Languages

A *language* over an alphabet $\Sigma$ is a set of strings over $\Sigma$.

$$L \subseteq \Sigma^*$$

That is, a language is any subset of $\Sigma^*$, 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

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

# Chapter 3

# State of the Art

## 3.1 Overview of XYZ

Computers are devices that

# Chapter 4

# Implementation

The implementation chapter gives insights into

## 4.1 Client-Server Architecture

This section describes the client-server architecture, which is important in the development of the application. It focuses on coordination of the mobile/web clients developed using Flutter and the Firebase server to support instant data flow, secure sign-in, and retrieval/storing of the data.

# Chapter 5

# Results and Discussion

This is a test

## 5.1 Evaluation

The methods of evaluating

# 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).