An empirical study of regular expression use in practice, sampling from Python projects on Github, leading to new concepts for refactoring regular expressions for readability.

by

Carl Allen Chapman

A thesis submitted to the graduate faculty
in partial fulfillment of the requirements for the degree of
MASTER OF SCIENCE

Major: Computer Science

Program of Study Committee:
Kathryn Stolee, Major Professor
Samik Basu
Tien Nguyen

Iowa State University
Ames, Iowa
2016

DEDICATION

I would like to dedicate this thesis to my mother, who believed in me and supported me through many years on a long winding road leading to a satisfying career. I'd also like to thank my wife Chien Wen Hung and our cat Siva for practical and moral support.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ACKNOWLEDGEMENTS

I would like to take this opportunity to express my thanks to those who helped me with various aspects of conducting research and the writing of this thesis. First and foremost, Dr. Kathryn Stolee for her guidance, patience and support throughout this research and the writing of this thesis. I would also like to thank my committee members for their efforts and contributions to this work: Dr. Samik Basu and Dr. Tien Nguyen.

ABSTRACT

Abstract

Though regular expressions (regex) provide a powerful search technique that is baked into every major language, is incorporated into a myriad of essential tools, and has been a fundamental aspect of Computer Science since Kleene in 1956, no one has ever formally studied how they are used in practice, or what can be done to make them easier to understand. This thesis presents the original work of studying a sample of regex taken from Python projects pulled from Github, determining what features are used most often, defining some categories that illuminate common use cases, and identifying areas of significance for tool builders. Furthermore, this thesis defines an equivalence class model used to explore comprehension of regex, identifying the most common and most understandable representations of semantically identical regex, suggesting several refactorings and preferred representations. Opportunities for future work include the novel and rich field of regex refactoring, semantic search of regexes, and further fundamental research into regex usage and understandability.

CHAPTER 1. OVERVIEW

1.1 Background on regex

1.2 Terminology used in this thesis

1.3 Structure of this thesis

CHAPTER 2. RELATED WORK

2.1 Programming languages that support regex

2.2 User tools that depend on regex

2.3 Analyzing and testing regex

2.4 Applications of regex

2.5 Formalisms addressing regex

CHAPTER 3. RESEARCH QUESTIONS

3.1 Gap in fundamental research into regex use in practice

3.2 Questions explored in this thesis and their motivations

3.2.1 RQ1: How are regex used in practice, especially what features are most commonly used?

3.2.2 RQ2: What preferences, behaviors and opinions do professional developers have about using regex?

3.2.3 RQ3: What behavioral categories can be observed in regex?

3.2.4 RQ4: Within five equivalence classes, what representations are most frequently observed?

3.2.5 RQ5: What representations are more comprehensible?

3.2.6 RQ6: For each equivalence class, which representation is preferred according to frequency and comprehensibility?

CHAPTER 4. Feature Analysis

4.1 Extracting utilizations from github repositories containing python

4.1.1 Experiment design

4.1.2 Implementation details

4.2 Utilizations of the re module

4.3 Building the corpus of patterns

4.3.1 Experiment design

4.3.2 Implementation details

4.4 Features of patterns in the corpus

4.5 Discussion of utilization and feature analysis results

4.5.1 Implications

4.5.2 Opportunities for future work

4.5.3 Threats to validity

CHAPTER 5. Developer Survey

5.1 Survey design based on feature analysis

5.2 Summary of results

5.3 Discussion of survey results

5.3.1 Implications

5.3.2 Opportunities for future work

5.3.3 Threats to validity

CHAPTER 6. Behavioral clustering

6.1 Experimental design

6.1.1 Conceptual basis

6.1.2 Overview of process

6.2 Similarity matrix creation

6.2.1 Implementation details

6.2.2 Results

6.3 Markov clustering

6.3.1 Background

6.3.2 Tuning parameters

6.3.3 Results

6.4 Categorization of clusters

6.4.1 Implementation details

6.4.2 Results

6.5 Discussion of cluster categories

6.5.1 Implications

6.5.2 Opportunities for future work

6.5.3 Threats to validity

CHAPTER 7. Equivalent representations of regex and their frequencies in the corpus

7.1 Experiment design

7.1.1 Defining five equivalence classes

7.1.2 Implementation details

7.2 Frequency analysis results

7.3 Discussion of representation frequency analysis

7.3.1 Context and common sense about these representations

7.3.2 Community support indicates a preference

7.3.3 Implications

7.3.4 Opportunities for future work

7.3.5 Threats to validity

CHAPTER 8. Comprehension of regex representations

8.1 Experiment design

8.1.1 Conceptual basis

8.1.2 Implementation details

8.2 Matching comprehension results

8.3 Composing comprehension results

8.4 Unsure answers as a result

8.5 Discussion of comprehension results

8.5.1 Implications

8.5.2 Opportunities for future work

8.5.3 Threats to validity

CHAPTER 9. Topological sort of representations by frequency and comprehensibility

9.1 Design of topological sort

9.1.1 Conceptual Basis

9.1.2 Implementation details

9.2 Total ordering of representations

9.3 Discussion of ordering results

9.3.1 Implications

9.3.2 Opportunities for future work

9.3.3 Threats to validity

CHAPTER 10. DISCUSSION

10.1 Implications of the thesis as a whole

10.2 Opportunities for future work studying regular expressions

10.2.1 Semantic search

10.2.2 Ephemeral regex

10.2.3 Comparing regex usage across communities

10.2.4 Evolution of patterns

CHAPTER 11. CONCLUSION

11.1 Summary of contributions

APPENDIX A. Patterns in Python projects from Github

This is now the same as any other chapter except that all sectioning levels below the chapter level must begin with the *-form of a sectioning command.

top 100 clusters used

top 10 by feature group

13,579 patterns: the corpus(1510 per page=9 pages)

Supplemental material.

APPENDIX B. Developer Survey

This is now the same as any other chapter except that all sectioning levels below the chapter level must begin with the *-form of a sectioning command.

Survey Questions

Survey Responses

Survey Statistics

More stuff.

APPENDIX C. Mechanical Turk Study

This is now the same as any other chapter except that all sectioning levels below the chapter level must begin with the *-form of a sectioning command.

Qualifying Test

Template

MT Data input

MT All Results

MT Summary Statistics

More stuff.

APPENDIX D. Community Analysis

This is now the same as any other chapter except that all sectioning levels below the chapter level must begin with the *-form of a sectioning command.

Filter Criteria

Summary Statistics

More stuff.