Readability and Refactoring of Regular Expressions

Carl Chapman
Department of Computer Science
Iowa State University
carl1976@iastate.edu

Kathryn T. Stolee
Department of Computer Science
North Carolina State University
ktstolee@ncsu.edu

ABSTRACT

Regexes are hard to understand. Let me tell you how.

1. INTRODUCTION

Our contributions are as follows:

- Identified two strong regex refactoring categories
- Identified 3 or so other regex refactorings categories and specific instances that are worthy of further investigations
- Identified a few regex refactorings that can be eliminated because both options are equally readable

The rest of this paper is organized as follows:
Related work, study, results, discussion, conclusion.

2. RELATED WORK

TODO.MID: We are building on the survey that indicates regexes are hard to read, and the apparent lack of any regex readability refactoring attempts. Many papers have talked about refactoring, basically it is changing the form but not the behavior.

3. STUDY

First we define a 'Functional Regex'(FR) as some regex that performs in a specific way. For many FRs, there are several concrete ways to express a single FR. We define a concrete regex(CR) as a regex expressed with a particular pattern String. Here is one illustration of these definitions:

TODO.NOW: create some examples for these terms

We identified 10 loose groups of FRs, described in this table:

TODO.NOW: create a table explaining the 10 groups

For each of these groups we created either two concrete versions of three FRs or three concrete versions of two FRs.

Each of the 10 categories had 6 concrete versions of some FR and so there are 60 CRs. For each CR, we selected 5 *example strings* designed to test the understanding of the CR. The idea is that different CRs may have different levels of readability, even when they are representing the same FR. We define readability as the ability to look at the CR and determine if an *example string* can be matched by it or not.

TODO.NOW: create some illustration of one matching subtask

In mechanical Turk, we designed a 180 tasks composed of 10 matching subtasks, so that each of the 60 CRs had 30 separate observations (each an average of 5 *example string* problems). These 1800 observations are what the analysis will focus on.

4. RESULTS

(for rough draft...)

Looks like M6 and M8 are the best meta-refactorings according to ANOVA. If you peek at MTResults Processing.csv on google docs, M6 has the best refactoring...every OCTAL type should be converted to an OR or preferably a CCC.

M8 has a weak P value, but still ok in one case (0.12) and consistently says that 'aa*' should be written as 'a+'.

Looks like M0, M1, M2, M3 and M9 are very dependent on the regex chosen, so regex-specific refactorings like: 0.1401 `&d([aeiou][aeiou])z'` `&d([aeiou]{2})z'` 0.075 `[\t\r\f\n ]'` `[\s]'` 0.1024 `[a-f]([0-9]+)[a-f]'` `[a-f](\d+)[a-f]'` 0.1271 `[\{][\$](\d+[.]\d)[}]'` `\\\{\\\$(\d+\.\d)\}'` (from M0,M1,M2,M9 respectively)

have okay P-values and may indicate regex-specific refactorings, but do not indicate an overall trend for that type of refactoring. Notice that M3 does not even have a strong p-value candidate, but this may be thrown off because of the very confusing regex chosen for CCC: 0.78 0.79 `xyz[_\[\]'\^\\]'` `xyz[\x5b-\x5f]'` which has a lot of escape characters, so that the hex group was easier to understand than the CCC.

Meanwhile M4,M5 and M7 have both ambiguous p-values and anova results. But this is still a finding: that no refactoring is needed between things like: `(q4fab|ab)'` `((q4f){0,1}ab)'` `tri[abcdef]3'` `tri(a|b|c|d|e|f)3'` `&(\w+);'` `&([A-Za-z0-9_]+);'` (from M4,M5,M7 respectively)

Although one refactoring from M5 might be of slight interest: 0.1196 FALSE `tri[a-f]3'` `tri(a|b|c|d|e|f)3'`

TODO.MID: more data from the composition problems

5. DISCUSSION: REFACTORING OPPORTUNITIES

ACM ISBN 978-1-4503-2138-9.
DOI: 10.1145/1235

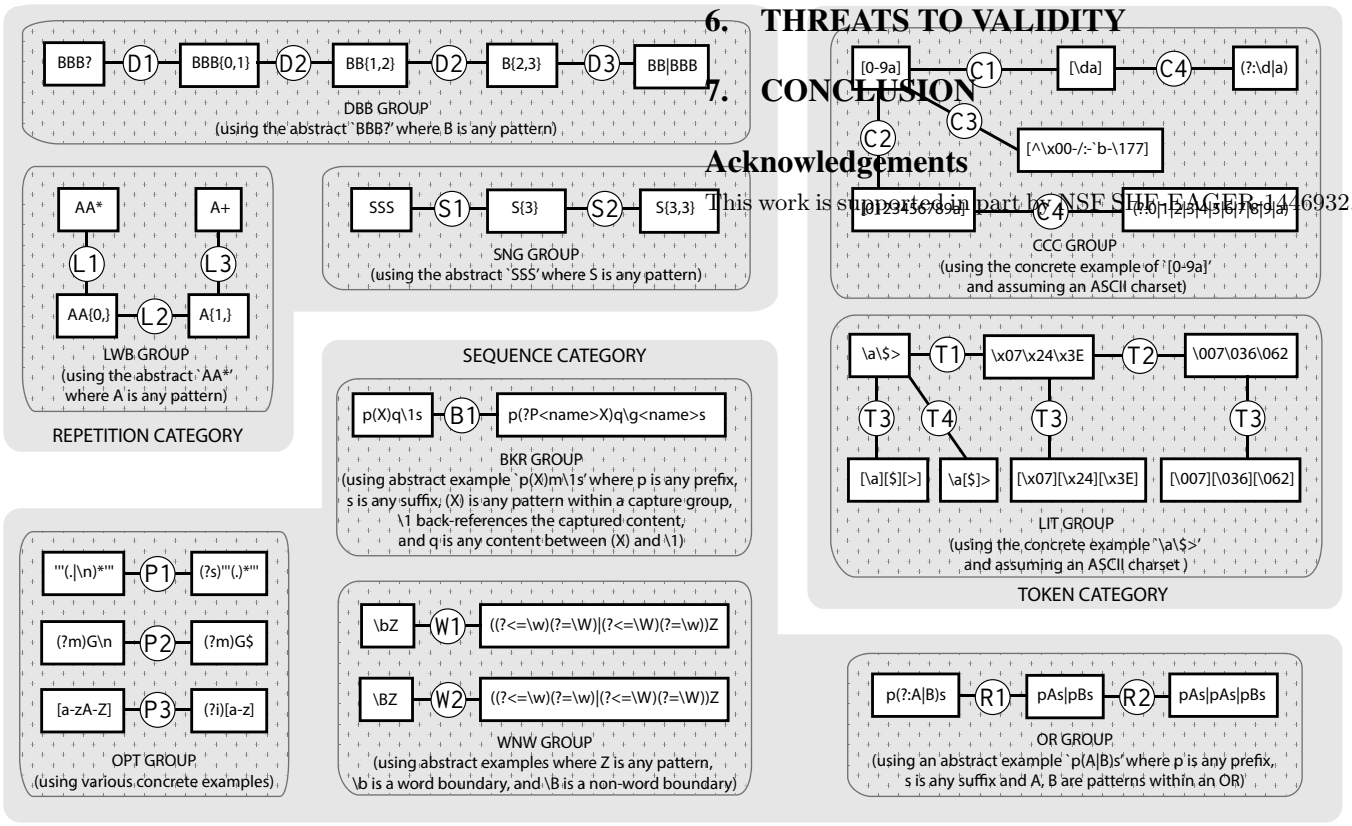

Figure 1: Some possible refactorings

6. THREATS TO VALIDITY

7. CONCLUSION

Acknowledgements

This work is supported in part by NSF SHF EAGER 1446932.