Poster: A Qualitative Reasoning Approach to Spectrum-based Fault Localization

Alexandre Perez
University of Porto and HASLab, INESC-TEC
Portugal
alexandre.perez@fe.up.pt

Rui Abreu
IST, University of Lisbon and INESC-ID
Portugal
rui@computer.org

ABSTRACT

As spectrum-based fault localization (SFL) reasons about coverage rather than source code, it allows for a lightweight, language agnostic way of pinpointing faults in software. However, SFL misses certain faults, such as errors of omission, and may fail to provide enough contextual information about its diagnoses. We propose Q-SFL, that leverages the concept of qualitative reasoning to augment the information made available to SFL techniques, by qualitatively partitioning the values of data units from the system, and treating each qualitative state as a new SFL component to be used when diagnosing.

CCS CONCEPTS

• **Software and its engineering** → **Software testing and debugging**; • **Computing methodologies** → *Causal reasoning and diagnostics*;

KEYWORDS

Spectrum-based Fault Localization, Qualitative Reasoning

ACM Reference Format:

Alexandre Perez and Rui Abreu. 2018. Poster: A Qualitative Reasoning Approach to Spectrum-based Fault Localization. In *Proceedings of 40th International Conference on Software Engineering Companion, Gothenburg, Sweden, May 27-June 3, 2018 (ICSE '18 Companion),* 2 pages. https://doi.org/10.1145/3183440.3195015

1 INTRODUCTION

SFL [4] was shown to be a lightweight, yet effective, technique for locating faults in a software system [1]. The spectrum abstraction level, while enabling a language-independent, lightweight analysis, imposes a tradeoff both in accuracy and comprehension, since it necessarily lacks fine-grained contextual information, essential for understanding *why* a given component is considered suspicious [6]. Aside from *comprehension*, the abstract nature of the spectra may also lead to the formation of *ambiguity groups* [7] and facilitate the occurrence of *coincidental correctness* [8].

To mitigate the issues above, we propose the inspection of the runtime value for relevant system data, with the intent of augmenting reports generated by SFL techniques. We leverage Qualitative Reasoning (QR), which provides a way of describing continuous values by their discrete, behavioral qualities, to enable the ability of reasoning about a system's behavior without exact quantitative information [3, 9]. QR is applied to the SFL analysis, in an approach named Q-SFL, enabling the introduction — both manually and automatically — of quantitative landmarks that partition the domains of system components into a set of qualitative descriptions. Since these descriptions are themselves considered as SFL components, we are then able to record their coverage and to diagnose them as per the SFL methodology, thus not only suggesting the likely location of the bug, but also pinpointing which behavioral properties induce a failure, *enriching* the SFL report as a result.

ICSE '18 Companion, May 27-June 3, 2018, Gothenburg, Sweden

ACM ISBN 978-1-4503-5663-3/18/05.
https://doi.org/10.1145/3183440.3195015

2 QUALITATIVE REASONING

QR studies the creation of discrete representations of the continuous world [2, 3, 9], enabling the reasoning of space, time, and quantity with merely a small amount of information. To reason about *qualitative states*, one needs to establish *landmarks*. Landmarks are constant quantitative values that establish a point of comparison to reason about the qualitative states [5]. QR also supports the representation of *derivatives*, enabling the use of sign algebra to reason about *direct influence* and *proportionality* between two qualitative values. Derivatives also enable *envisionments*. An *envisionment* establishes a set of transitions between qualitative states [2], essentially modeling the abstracted world.

3 Q-SFL

Our Q-SFL approach consists of partitioning several SFL components into multiple, meaningful, qualitatively distinct *subcomponents*, to be used in the fault localization. We leverage the QR concept of domain partitioning to inspect existing components during each test execution and assign them a qualitative state. Each qualitative state is to be considered as a separate SFL component whose involvement per transaction is recorded in the spectrum.

We envision distinct types of *landmarking strategies* that can be employed to define qualitative state boundaries: (i) *manual landmarking*, where developers manually define the possible qualitative states for a given component; (ii) *static landmarking*, where landmarks depend on the *type* of a component; and (iii) *dynamic landmarking*, where a component's value is inspected at runtime, and partitioned into a set of categories. An example *static landmarking*: if a component c represents a method that contains a numeric parameter i, we can create three qualitative components c', c'' and c''' that represent invocations of c with positive, negative and zero values of i, respectively. Note that the original component c is not removed from the QR-augmented spectrum, as partitions may not

Table 1: Evaluation results and statistical tests.

	Original	QR-enhanced
Mean C_d	60.28	37.56
Median C_d	6.00	2.50
C_d Variance	2.10×10^4	1.56×10^4
Shapiro-Wilk	$W = 0.46$ p-value = 2.20×10^{-22}	$W = 0.32$ p-value = 1.10×10^{-24}
Wilcoxon Signed-rank	$Z = 5.45$ p-value = 5.10×10^{-10}	

provide further fault isolation. Examples of dynamic strategies will be presented in Section 4.

By partitioning components and gathering their (qualitative) runtime value, we are providing more context to the diagnostic process, and in some cases, consequently reducing the diagnostic effort. Partitioning can also be of crucial importance towards minimizing occurrence *ambiguity grouping* and *coincidental correctness*, since new, distinct components are added are fed into the SFL framework.

4 EVALUATION

Experimental subjects are sourced from the Defects4J[1] (D4J) database. We run the fault-revealing test suite of each buggy D4J subject, gathering method-level coverage and test outcomes, to construct its spectrum. We also record the value of all *primitive-type* arguments and return values for every method call, in order to create multiple (automated) partitioning models resulting in several Q-SFL variants. A *static partitioning* variant using automated sign partitioning based on the variable's type was considered. For *dynamic partitioning*, several clustering and classification algorithms[2] were considered. Test outcomes are used as the class labels in the case of supervised models. Because we use automated, domain independent partitioning, only primitive types are considered in the evaluation. To evaluate a QR-enhanced spectrum against its respective original spectrum, we use the C_d metric [1], also known as wasted effort, that measures the position of the faulty component in the ranked list of diagnostic candidates produced by the SFL framework.

We were able to automatically partition the faulty method in 167 D4J subjects. In order to assess whether the addition of qualitative components to the spectrum improves SFL accuracy, we consider, for each D4J subject, the landmarking strategy able to create the largest set of distinct, non-ambiguous qualitative components out of the faulty method. Results gathered and respective statistical tests are shown in Table 1. QR-enhanced spectra exhibits an overall lower effort to diagnose when compared to the original spectra, with less variance. To assess significance, we first performed the Shapiro-Wilk test for normality of effort data in both the original spectra case and QR case. Given that C_d is not normally distributed and that each observation is paired, we use the non-parametrical statistical hypothesis test Wilcoxon signed-rank. Our null-hypothesis is that the median difference between the two observations (*i.e.*, ΔC_d) is zero. The fifth row from Table 1 shows the resulting Z statistic and p-value of Wilcoxon's test. With 99% confidence, we refute the null-hypothesis.

The results shown above indicate that augmenting faulty spectra with new components resulting from qualitative landmarking of method parameter (and method return) values yields a *statistically significant* improved diagnostic report. Note, however, that while we show the *existence* of QR-augmented spectra with improved diagnosability, we have not observed evidence that a single automated strategy — out of the ones considered in the evaluation — can consistently outperform the original spectra.

5 CONCLUSION

This paper proposes a new approach to spectrum-based fault localization that leverages qualitative reasoning (QR). QR is a research field of Artificial Intelligence focused on the study of ways to describe continuous variables by a set of finite qualitative states, allowing for the modeling, simulation and reasoning of complex systems. Q-SFL splits runtime data values into qualitative states through the creation of qualitative landmarks that partition the data domain. These qualitative states are then considered as SFL components to be ranked using traditional fault-localization methodologies. Since we treat qualitative descriptions of variable domain partitions as components, our diagnostic reports not only recommend likely fault locations, but also what qualitative *behaviors* suspicious components assume when failures are detected, facilitating the comprehension of the fault.

We evaluate the approach on subjects from the Defects4J catalog of real faults from medium and large-sized open source projects. Results show that spectra which were *augmented* using qualitative partitioning of method parameters shows a (statistically significant) improvement in the diagnostic accuracy. However, we also found no single (automated) landmarking strategy that was consistently better than the original spectra, meaning that the use of more intricate, *white-box* strategies will likely be necessary for practical applications of the approach.

ACKNOWLEDGMENTS

This material is based upon work supported by the scholarship number SFRH/BD/95339/2013 from Fundação para a Ciência e Tecnologia (FCT).

REFERENCES

[1] Rui Abreu, Peter Zoeteweij, Rob Golsteijn, and Arjan J. C. van Gemund. 2009. A practical evaluation of spectrum-based fault localization. *Journal of Systems and Software* 82, 11 (2009), 1780–1792.
[2] Johan de Kleer. 1977. Multiple Representations of Knowledge in a Mechanics Problem-Solver. In *Proceedings of IJCAI'77*. 299–304.
[3] Kenneth D. Forbus. 1997. Qualitative Reasoning. In *The Computer Science and Engineering Handbook*. 715–733.
[4] James A. Jones, Mary Jean Harrold, and John T. Stasko. 2002. Visualization of test information to assist fault localization. In *Proceedings of ICSE'02*. 467–477.
[5] Benjamin Kuipers. 1986. Qualitative Simulation. *Artificial Intelligence* 29, 3 (1986), 289–338.
[6] Chris Parnin and Alessandro Orso. 2011. Are automated debugging techniques actually helping programmers?. In *Proceedings of ISSTA'11*. 199–209.
[7] G. N. Stenbakken, T. M. Souders, and G. W. Stewart. 1989. Ambiguity groups and testability. *IEEE Transactions on Instrumentation and Measurement* 38, 5 (1989), 941–947.
[8] Xinming Wang, Shing-Chi Cheung, Wing Kwong Chan, and Zhenyu Zhang. 2009. Taming coincidental correctness: Coverage refinement with context patterns to improve fault localization. In *Proceedings of ICSE'09*. 45–55.
[9] Brian C. Williams and Johan de Kleer. 1991. Qualitative Reasoning about Physical Systems: A Return to Roots. *Artificial Intelligence* 51, 1-3 (1991), 1–9.

[1] Defects4J 1.1.0 is available at https://github.com/rjust/defects4j.
[2] Namely, k-NN, linear classification, logistic regression, decision trees, and random forest, as implemented in the `Scikit-learn` package. X-means, as implemented in the `pyclustering` package, was also selected.