Do Automated Program Repair techniques repair *difficult* Bugs?

Research Idea Proposal

Manish Motwani

Sandhya Sankaranarayanan

INTRODUCTION

- Automated Program Repair Techniques
  - *Testing-Based Approaches*: use faulty programs + positive, negative test cases
  - For e.g. GenProg, PAR etc.
- Evaluation of these techniques
  - Specific Datasets
  - Parameters such as #Bugs fixed, Types of bugs fixed, understandability of patches
  - **Notion:** These techniques DONOT repair bugs that are “Difficult” to repair by Humans (experienced software developer).
  - Evaluation parameters DONOT explicitly address the “Difficulty” aspect of bugs.

Research Question

Notion: These techniques DONOT repair bugs that are “Difficult” to repair by Humans (experienced software developer).

RQ: Do Automated Program Repair techniques (testing-based) repair the bugs that seem “Difficult” to humans (experienced software developers)?

- How to quantify “Difficulty” of a Bug repair?

Key Idea

RQ: Do Automated Program Repair techniques (testing-based) repair the bugs that seem "Difficult" to humans (experienced software developers)?

- How to quantify "Difficulty" of a Bug repair?

Idea:

- ➢ Create a bug repository from various sources along with additional meta-information.
- ➢ Identify parameters associated with the "Difficulty" of a bug repair using meta-information.
  - ➢ Time taken to repair the bug
  - ➢ #components impacted for repairing the bug
  - ➢ #lines modified
  - ➢ Difficulty to replicate the bug (may be based on its type)
  - ➢ What else ??
- ➢ Formulate "Difficulty-Score" in terms of these parameters such that the implication of quantified "Difficulty" is consistent with the understanding of "Difficulty" for humans.
- ➢ Test existing approach(es) to identify if they can actually fix "Difficult" bugs.

Evaluation Plan

- Analyze existing datasets and compare it with our repository to evaluate the comprehensiveness of the repository
- Empirical validation of the “Difficulty-Score” obtained for bugs by software developers.
- For a given technique, validate its results against the results obtained after running it on our repository. These should be consistent.