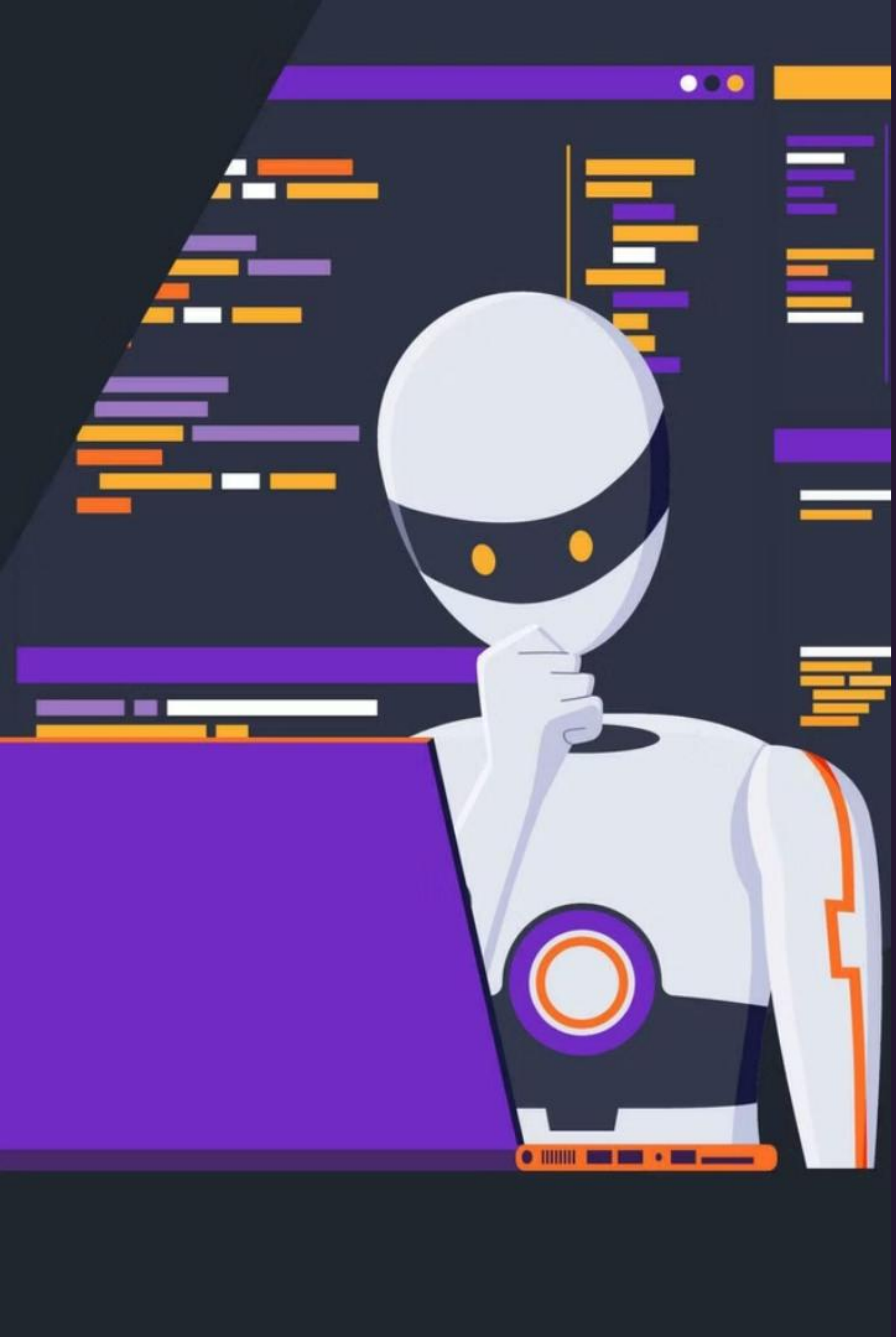


Refactory: Re-Factoring Based Program Repair Applied to Programming Assignments

Tool Demo Presentation

Prepared by: Priyadarshine Kumar

Date: March 18, 2025

Motivation

1 Automated Program Repair (APR) helps correct buggy student programs.

2 Existing tools require manual effort or large datasets of correct solutions.

3 Student submissions often contain severe errors, making repair challenging.

4 Need an approach that works with fewer correct solutions and scales efficiently.

Limitations of Existing Tools

Clara

Requires many correct solutions, suffers from scalability issues.

Refazer

Relies on learned transformations but lacks accuracy.

Neural-based approaches

Suffer from low precision and unpredictable repairs.

Autograder

Manual effort required for error models in tools like Autograder.

Introduction to Refactory

Fully automated program repair tool.

Works with only one correct solution.

Uses refactoring to generate new correct solutions.

Matches buggy programs with refactored correct programs.

Synthesizes small and efficient patches.

Technical Overview: Three-Phase Approach

1. Refactoring Correct Solutions

Generates alternative versions of correct programs.

2. Structure Alignment

Matches buggy programs with refactored correct solutions.

3. Block Repair

Uses variable mapping and synthesis to correct buggy code.

Phase 1: Refactoring Correct Solutions

1. **Apply Refactoring Rules**

   Applies 10 refactoring rules to generate semantically equivalent correct programs.

2. **Modify Control Flow**

   Modifies control flow by adding/removing branches, loops, and conditions.

3. **Improve Alignment**

   Helps align correct solutions with buggy student submissions.

Phase 2: Structure Alignment

1

Find Closest Match

Finds the closest refactored correct program based on control flow graphs.

2

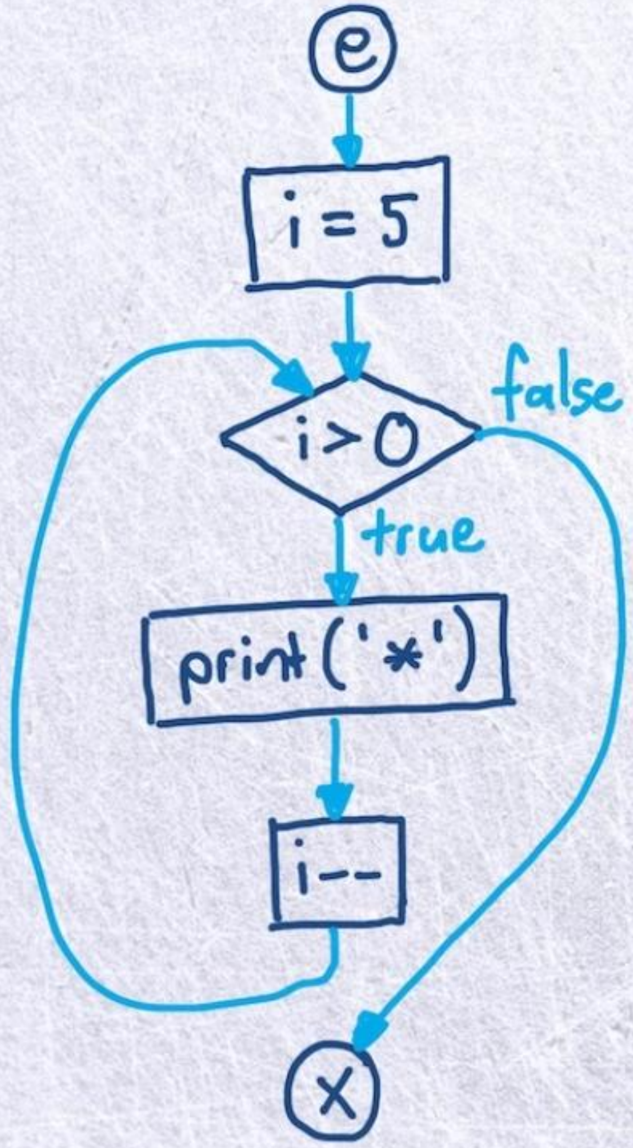
Modify If Needed

If no match is found, modifies the buggy program's structure to align it.

3

Ensure Accuracy

Ensures accurate patch synthesis by improving similarity.



Phase 3: Block Repair

Analyze Equivalence

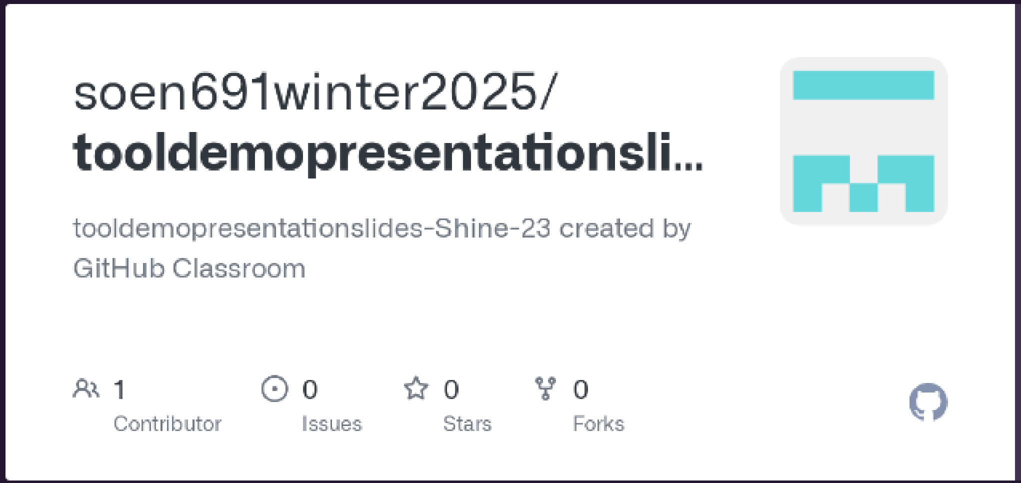
Uses dynamic equivalence analysis for variable mapping.

Map Basic Blocks

Maps basic blocks between buggy and correct programs.

Synthesize Expressions

Synthesizes expressions to create minimal and accurate patches.



soen691winter2025/
tooldemopresentationsli...

tooldemopresentationslides-Shine-23 created by GitHub Classroom

1 Contributor
0 Issues
0 Stars
0 Forks

GitHub

Refactory

Installation Steps

Tool Installation

Conclusion & Future Work

1 Refactory is an effective automated repair tool for student programming assignments.

2 Outperforms Clara in repair accuracy, efficiency, and scalability.

3 Can be extended to professional software repair beyond educational use.

4 Future work includes supporting object-oriented programming and additional Python features.