Automated Repair of Programs from Large Language Models

- Zhiyu Fan, Xiang Gao, Martin Mirchev, Abhik Roychoudhury, Shin Hwei Tan

Discussion Lead

- Wasique Islam Shafin (40304330)

Question 1

In the paper, authors evaluates APR tools on code generated on easy to medium tasks.

What would you change to enable repairing larger and complex code? How would you make it efficient? What are the drawbacks/considerations?

Question 2

Codex, The LLM used in the paper has been deprecated by OpenAI.

Are the findings in the paper still relevant? What changes can happen when using newer LLMs?

Question 3

The APR tools used in the paper need a set of test cases to fix the generated code.

How would you go about fixing code when you don't have any tests?

Question 4

The authors only used 2 APR tools: TBar & Recorder

Do you think these 2 can represent for all of APR tools?

What are the drawbacks of each tool? What APR tools would you include to complement these?

Question 5

In the paper the authors focused on fixing Java Programs only.

Why did they only fix Java programs? How much effort & changes will it take to repair non-Java Programs?

Question 5:

What changes or ideas could be studied to improve upon this paper?

Thank You