Augmenting Automated Black Box Testing with Abductive Reasoning

Doha Ara, Ross Rannells and Dr. James Hill
Indiana University School of CSCI
Indiana University-Purdue University Indianapolis

IUPUI
CENTER FOR RESEARCH AND LEARNING
Undergraduate Research Opportunities Program
Division of Undergraduate Education

Introduction

- Modern software systems, e.g., large scale web sites, autonomous vehicles, are large collections of software must be thoroughly tested.
- Testing is a painstaking process that results in limited application of valid techniques, e.g., black box testing
- Limited application produces gaps and/or bias in software validation & verification
  - e.g., how do we know we wrote the correct code we are testing?

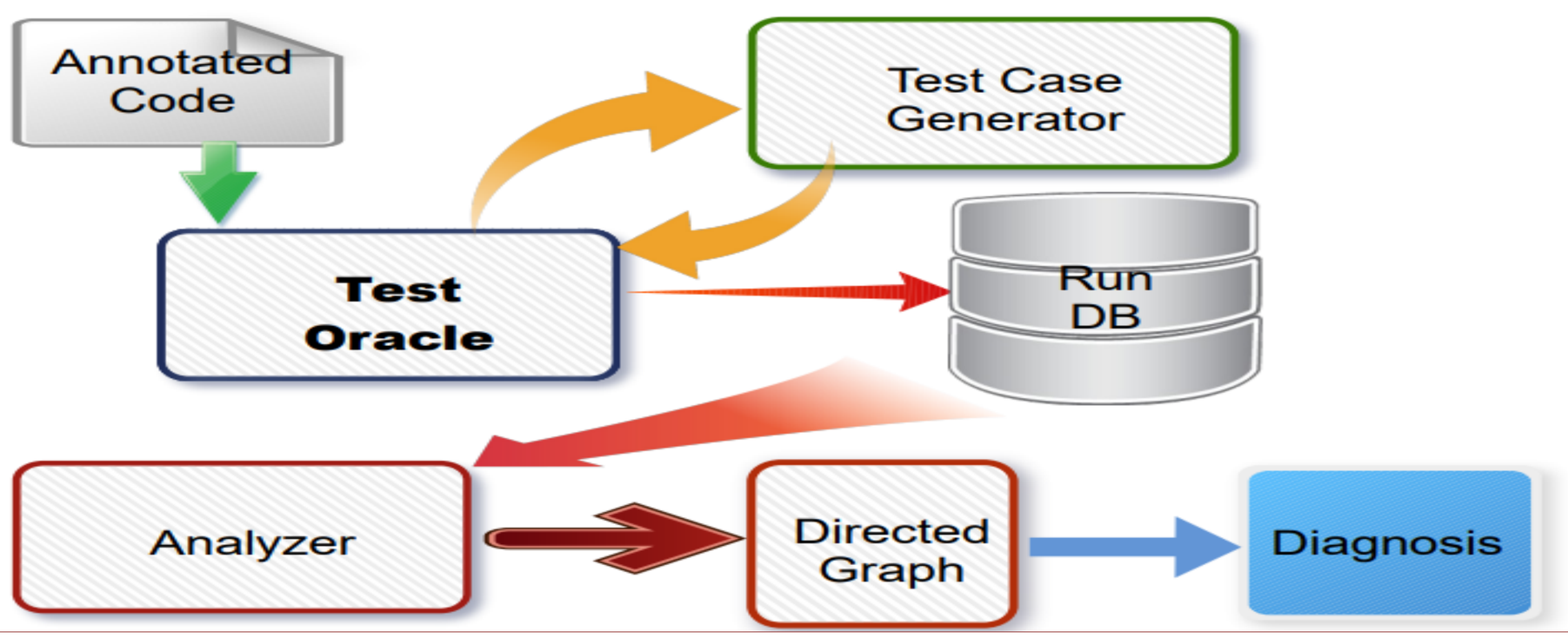
Objectives

Automate black-box testing by integrating an abductive reasoning artificial intelligent (AI) engine into the testing process to identify and locate errors during the testing process.

Apply our approach to software and models used for design and implement autonomous vehicles.

Data and Research Methodology

1. Load Annotated Code to Oracle
2. Get Test Cases from Test Case Generator
3. Store Test Results to Database
4. Load Test Data Into Analyzer
5. Generate Directed Graph
6. Abduce Diagnosis



Results

- Created test case generator for numerical values; extending to other built-in types and abstract data types
- Created engine that can auto-generate black box test oracles for Java code without any human intervention
- Implemented Parsimonious Covering Reasoning engine that can find primary cause of error(s) in test applications

Conclusions

- Our approach will produce unbiased test results that validate systems requirements.
- Our approach can easily integrate into a CI/CD pipeline allowing for earlier error detection.
- Next steps, apply our approach on real-world software applications