IC637 Program Analysis

Lecture 0: Course Introduction

Minseok Jeon

2025 Fall

Outline

1. **Basic Information**

2. **Course Overview**

Basic Information

Basic Information

Instructor : Minseok Jeon

- **Position :** Assistant Professor, Department of EECS, DGIST
- **Expertise :** Programming Language, Software Engineering
- **Office :** 211, E7 Building
- **Email :** minseok_jeon@dgist.ac.kr
- **Office Hours :** Thursday 2:00 PM - 4:00 PM

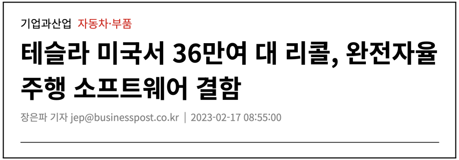
Course slides will be uploaded to:

- LMS
- https://dgistpl.github.io/courses/ic637/2025/

Course Overview

Why?

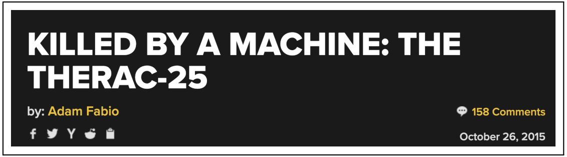
- We should **find** bugs before they are deployed!
- SW bugs are everywhere



기업과산업 자동차·부품

테슬라 미국서 36만여 대 리콜, 완전자율 주행 소프트웨어 결함

장은파 기자 jep@businesspost.co.kr | 2023-02-17 08:55:00



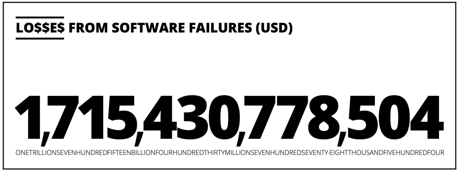
KILLED BY A MACHINE: THE THERAC-25

by: Adam Fabio

158 Comments

October 26, 2015

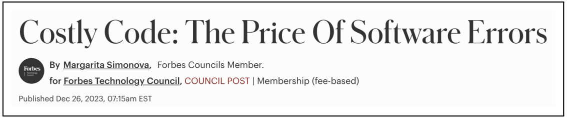
- Enormous cost due to SW bugs



LO$$E$ FROM SOFTWARE FAILURES (USD)

1,715,430,778,504

ONETRILLIONSEVENHUNDREDFIFTEENBILLIONFOURHUNDREDTHIRTYMILLIONSEVENHUNDREDSEVENTY-EIGHTTHOUSANDFIVEHUNDREDFOUR



Costly Code: The Price Of Software Errors

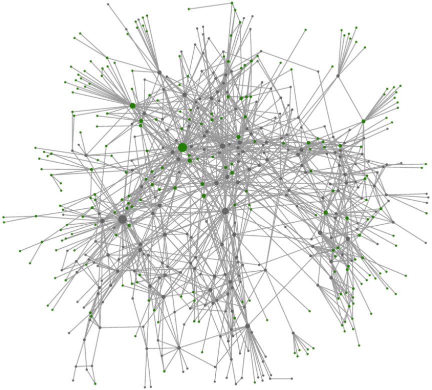
By Margarita Simonova, Forbes Councils Member.

for Forbes Technology Council, COUNCIL POST | Membership (fee-based)

Published Dec 26, 2023, 07:15am EST

How?

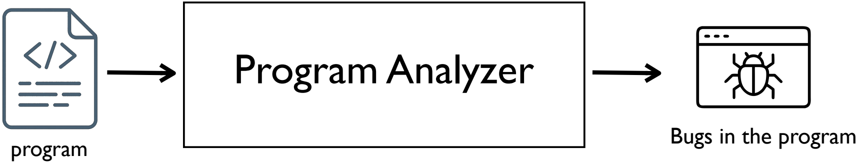
- Manually investigate the code and finding bugs is extremely difficult[1].



[1] Klein, Gerwin & Andronick, June & Elphinstone, Kevin & Murray, Toby & Sewell, Thomas & Kolanski, Rafal & Heiser, Gernot. (2014). Comprehensive Formal Verification of an OS Microkernel. ACM Transactions on Computer Systems (TOCS). 32. 10.1145/2560537.

How?

- We need to develop a tool that analyzes the code automatically.



What?

Program analysis aims to reason about program behavior (e.g., bugs) automatically.

- Static analysis: analyzing code without execution.
- Dynamic analysis: analyzing program during execution.

Applications: bug finding, security, optimization, verification.

Topics Covered

In this course, we will focus on foundational topics on program analysis

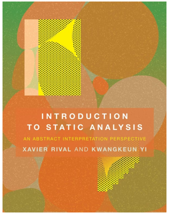
Weeks	Topics	Weeks	Topics
Week 1	Course Overview	Week 9	Axiomatic Semantics
Week 2	Introduction to Program Analysis	Week 10	Abstract Interpretation
Week 3	Static Analysis Examples 1	Week 11	Abstract Interpretation Example 1
Week 4	Static Analysis Examples 2	Week 12	Abstract Interpretation Example 2
Week 5	Operational Semantics	Week 13	Pointer Analysis
Week 6	Denotational Semantics	Week 14	Advanced Techniques 1
Week 7	—	Week 15	Advanced Techniques 2
Week 8	—	Week 16	Final Exam

Assessment

- Programming Assignments: 40%
- Final Exam: 50%
- Class Participation: 10%

Course Materials

- Lecture slides
- Xavier Rival and Kwangkeun Yi. Introduction to Static Analysis: An Abstract Interpretation Perspective. MIT Press


INTRODUCTION
TO STATIC ANALYSIS
AN ABSTRACT INTERPRETATION PERSPECTIVE
XAVIER RIVAL AND KWANGKEUN YI

Questions?

Thank you!

Any questions about the course?