IT5440- NGUYÊN LÝ VÀ KỸ THUẬT PHÂN TÍCH CHƯƠNG TRÌNH

(Principle and Technique of Program Analysis)

AY 2025-2026

Giảng viên: PGS. TS.Huỳnh Quyết Thắng
Khoa Khoa học máy tính
Trường Công nghệ thông tin và Truyền thông
www.soict.hust.edu.vn/~thanghq

Chapter I. Introduction

- **Program analysis** is essential to enforcing security, achieving correctness, improving performance, and reducing costs when developing and maintaining software.
- **Program analysis** can be divided into *static analysis* and *dynamic analysis*.
  - **A static program analysis** is best described as a method of debugging that is done by automatically examining the source code without having to execute the program. This provides developers with an understanding of their code base and helps ensure that it is compliant, safe, and secure.
  - **A dynamic program analysis** identifies defects after you run a program (e.g., during unit testing). However, some coding errors might not surface during unit testing. So, there are defects that dynamic testing might miss that static code analysis can find.

Chapter I. Introduction

- **Static Program analysis**
  - Static analysis works with a representation of the source code alone, e.g., the program text, the abstract syntax tree, a graph representing the control flow.
- *Abstract interpretation* Instead of attempting to analyze the full behaviour of a program (which is in general undecidable), the idea is to create a more abstract representation, typically a finite one, which can be fully analyzed.
- *Type systems*, where tasks include type checking, type inference, and application of types to non-standard problems, such as aliasing;
- *Verification*, which gives a proof about all possible executions of a program and is needed in mission-critical situations
- *Dataflow Analysis*
- *Control Flow Analysis: Path Sensitivity*
- *CFG (Interprocedural Analysis)*
- *Pointer Analysis*

Chapter I. Introduction

A dynamic program analysis

- *Tracing*
- *Profiling*
- *Checkpointing and replay*
- *Dynamic slicing*
- *Execution indexing*
- *Delta debugging*
- *Memory Detection*
- *Fault localization*

Chapter I. Introduction

Abstract interpretation is a theory of sound approximation of the semantics of computer programs, based on monotonic functions over ordered sets, especially lattices.

It can be viewed as a partial execution of a computer program which gains information about its semantics (e.g., control-flow, data-flow) without performing all the calculations.

Its main concrete application is formal static analysis, the automatic extraction of information about the possible executions of computer programs; such analyses have two main usages:

- inside compilers, to analyse programs to decide whether certain optimizations or transformations are applicable;
- for debugging or even the certification of programs against classes of bugs.

Chapter I. Introduction

- Verification: define and prove automatically a property of the possible behaviors of a complex computer program (example: program semantics);
- Abstraction: the reasoning/calculus can be done on an abstraction of these behaviors dealing only with those elements of the behaviors related to the considered property;
- Theory: abstract interpretation.
- Semantics: The concrete semantics of a program formalizes (is a mathematical model of) the set of all its possible executions in all possible execution environments.

Chapter I. Introduction

Undecidability

- The concrete mathematical semantics of a program is an infinite mathematical object, not computable;
- All non-trivial questions on the concrete program semantics are undecidable.

Example: termination

- Assume termination (P) would always terminates and returns true iff P always terminates on all input data;
- The following program yields a contradiction P iff while termination(P) do skip od.

Homework 1

1) Khái niệm Abstract interpretation có ỹ nghĩa như thế nào trong quá trình phát triển Static Program Analysis
2) Tại sao các công cụ dựa trên ý tưởng Abstract Interpretation chỉ mang tính ứng dụng lý thuyết nhiều hơn
3) Hãy cho biết vào thời điểm 2024 có công cụ nào dựa trên Abstract Interpretation phát triển ứng dụng được trong công nghiệp hay không