1. Program Analysis

20190065 김기환 (kimkihwan@kaist.ac.kr)

* semantic = runtime behaviors of program
* semantic property = property about semantics
* program analysis = checking whether the program satisfies semantic properties
* concepts of program analysis =

\* Target Program

1. Domain Specific Analysis: Focus on purpose of the program

2. Non-Domain Specific Analysis: Analyze regardless of sake of program

3. Program Level Analysis: Analyze on real code

4. Model Level Analysis: Analyze on model of semantics

\* Target Property

1. Safety Property: Check existence of bad behavior of program

2. Liveness Property: Check whether program will not terminate

3. Information Flow Property: Check dependency between set of programs

* static vs dynamic – why static analysis is good?

1. Does not affect to cost of program in run-time

2. Some behavior like termination cannot be checked by dynamic analysis

3. Developer can fix potential error or wrong behavior immediately after analysis

* General analyzer? No, because of ‘Halting Problem’ and ‘Rice Theorem’
* Approximation: Soundness & Completeness

1. Sound: Reject all program which not satisfy the property (e.g. Typing)

2. Complete: Pass all program which satisfy the property (e.g. assertion)

* Approaches of program analysis

1. Testing: Execute program for some given inputs

Pros: Easy to automate and lots of useful test case generator exist.

Cons: Hard to achieve high coverage and program must be deterministic.

2. Assisted Proof: User gives additional information that helps analyzing

Pros: Both sound and complete analyzing is possible.

Cons: Requires expertise and user’s efforts.

3. Model Checking: Check (finite) model of program

Pros: Automated, sound and complete.

Cons: It depends on given finite state model.

*(if user gives model, it is model-level. Otherwise, if analyzer automatically generates model from given program, it is program-level)*

4. Conservative Static Analysis: Check source code with some logics

Pros: Automated and sound. It doesn’t need finite model.

Cons: Most case, it is incomplete.

5. Bug Finding: Low-cost analysis

Pros: Simple and not requires many costs for running.

Cons: Neither sound nor complete.