

SNU 4541.664A 프로그램 분석, 2019 가을

kwangkeunyi.snu.ac.kr/4190.664A/19

1 시간 장소

- 강의: 화/목 15:30-16:45 @ 302동 309-1호
- 과목 홈페이지: kwangkeunyi.snu.ac.kr/4541.664A/19
- 강의동영상: kwangkeunyi.snu.ac.kr/4541.0644A/mooc

2 교수: 이광근

연구실: 302동 428호, 컴퓨터공학부

교수 홈페이지: kwangkeunyi.snu.ac.kr

이메일: kwang@ropas.snu.ac.kr

조교: 이동권 dklee@ropas.snu.ac.kr

3 목표

작성한 소프트웨어가 생각대로 실행될 지를 실행시키기 전에 미리 엄밀하게 확인해주는 기술들은 있는가? 이에 대한 답이 정적 프로그램 분석(static program analysis)이라는 이름으로 모여지는 기술들이다. 그동안 다양한 이름으로 다양한 수준에서 다양한 필요에 맞추어 불리워지는 기술들을 모두 포섭한다: “static analysis”, “abstract interpretation”, “type system”, “software model checking”, “data-flow analysis”, “program logics and proof system” 등.

이 강좌에서는 정적 프로그램 분석 기술들의 이론과 실재를 익힌다:

- 강의
 - 위의 모든 기술들을 포섭하는 일반 정적분석 디자인 틀인 요약해석(abstract interpretation)과 그 배경이론을 익힌다.
 - 요약해석만큼 일반적이지는 않지만 제한된 경우에 간편하게 사용할 수 있는 특화된 정적분석 디자인 틀 세 가지(analysis by equation, by monotonic closure, and by proof construction)를 익힌다.
 - 정적분석기의 실행 비용을 줄이는 기술들을 익힌다.
- 숙제
 - 정적분석기 디자인에 필요한 개념 훈련하기.
 - 요약해설 틀(이론)을 이용해서 정적분석기를 디자인하고 디자인한 분석기가 안전함을 증명하기.
 - 디자인을 실제 프로그램으로 구현하고 실행해 보기.

4 내용

- Preliminaries: abstract syntax, compositional semantics, logics and inference, inductions, fixpoints
- Semantic formalisms: natural semantics, structural operational semantics, abstract machine semantics
- **General framework:** abstract interpretation, transitional approach, compositional approach, abstraction, concretization, correctness, fixpoint iterations, algorithms
- **Specialized frameworks:** analysis by equations, analysis by monotonic closure, analysis by proof construction, data flow analysis, constraint-based analysis, monomorphic type system, polymorphic type system, effect system
- Implementations: algorithms, worklist, sparse analysis
- Exercises: design and development of static analyzers

5 성적

절대평가. 숙제 90%. 기타 10%.

- 숙제는 프로그램숙제와 문제풀기로 구성된다.
- 모든 숙제는 제출기한을 지나 48시간 이내로 늦어지면 10% 감점.
- 프로그래밍은 OCaml로 한다.