



프로그램 따로분석의 이론적 기틀

재귀적 모듈 지원하기

이준협, 이광근

2024년 1월 31일

2024년 SIGPL 겨울학교

무엇을 하고 싶은가?

- 따로분석이란,
- 프로그램에서 쓰이는 이름이 어디에서 정의되는지 몰라도,
- 그 프로그램이 할 수 있는 일을 포섭하는 분석이다.

왜 필요한가?

- 우리가 분석하는 프로그램은 코드 전체가 알려져있는 일이 드물다.
ex. 외부 라이브러리 함수를 부르는 경우
- 외부 환경과 “따로” 분석해야한다

어떻게 할 것인가?

- “Algebraic Effects”: 환경을 받는 함수를 자료구조로 나타냄.
- “환경”: 이름을 물어보면, 답을 돌려줌
- 환경과의 상호작용을 요약하고, 나중에 환경이 알려지면 풀자!

왜 어려운가?

- 언어마다 이름을 알아내는 방법이 다 다르다.
- 이름을 알아내는 방법이 복잡할 수도 있다.
ex. First class modules, Functors, Recursive modules
- 심지어, OCaml에서 링킹은 끝나지 않을 수도 있다!
False를 증명할 수 있다 (Girard's paradox)
- 나중에 풀 때, 어떻게 “안전하게” 풀 것인가?

잘 요약하기 위해서는

- 요약해석이 알려주는 사실은,
- 기초가 되는 의미구조만 잘 정의해주면,
- 안전한 요약은 “자연스럽게” 가능하다는 것이다.

- 이름 알아내기 어려운 언어에서
- 외부와의 상호작용을 발생시키고
- 이후 푸는 연산을 최대한 간단하게 표현하는 의미구조?

특히, 재귀적 모듈을 위한 의미구조?

의미구조 정의와 이후 로드맵
포스터에 있습니다.

들어주셔서 감사합니다.