단계제거 변환을 통한 다단계 프로그램의 정적분석

Wontae Choi[1], Baris Aktemur[2], Kwangkeun Yi[1], and Makoto Tastuta[3]

[1] Seoul National University [2] Ozyegin University [3] National Institute of Informatics

1. 다단계프로그램

n단계 프로그램 실행 결과 = n+1단계 프로그램

```
#define SQUARE(x) ((x)*(x))
int main(){
    return SQUARE(2);
}
```

예) C + macro : 2단계 프로그램

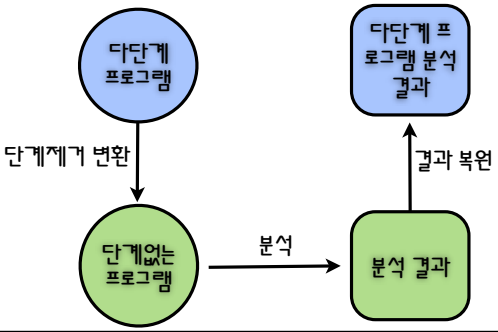
다단계 프로그램은 어디에나 있다

LISP, C, C#, C++, Haskell, Python, PHP ...

2. 문제 : 다단계프로그램 분석

실행전에 프로그램 코드를 알 수 없어 분석이 어렵다.

3. 아이디어 : 단계없는 프로그램으로 변환해서 분석



4. 변환 : 다단계에서 단계없는 프로그램으로

다단계언어

e := λx.e | ee | x | `e | ,e | run e

코드 생성 코드 조립 코드 실행

단계없는 언어

e := λx.e | ee | x | {} | e{x=e} | e.x

빈 레코드 갱신 접근

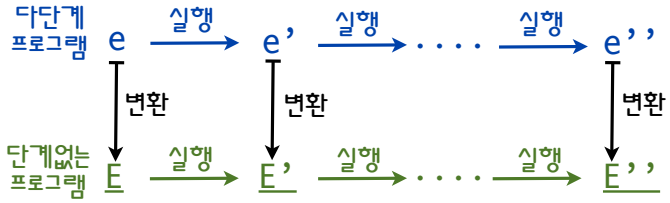
변환 로제타석

다단계 프로그램	실행 결과		변환 결과	
`(1+1)	`(1+1)	코드값 "1+1"을 정의한다.	λp.1+1	코드는 함수로 변환
run `(1+1)	2	코드값을 실행한다.	(λp.1+1) {}	실행은 함수호출로 변환
`(1+,((λx.x) `1))	`(1+1)	빨간부분을 실행하고 결과 코드를 바깥쪽에 끼워넣는다.	(λh.λp.1+(hp)) ((λx.x) (λp.1))	빨간부분을 밖에서 실행시키고 함수 인자로 넘겨줌
`(λx.x y)	`(λx.x y)	자유변수가 있는 코드.	λp.λx.x p.y	자유변수는 레코드 접근으로 변환
`(λx.,(`x))	`(λx.x)	자유변수는 다른 코드에 끼워넣을때 묶인다.	(λh.λp.λx.h p{x=x}) (λp.p.x)	레코드를 통해 바인딩을 처리.

5. 변환의 성질

변환은 실행 의미를 보존

실행후 변환한 결과와 변환후 실행한 결과는 같다 (매스텝)



6. 결론

* 다단계 프로그램은 단계없는 프로그램으로 변환 가능

* 변환결과는 원본 프로그램의 의미를 보존한다

* 기존의 단계 없는 프로그램 (람다프로그램)을 위한 분석을 이용해 다간계 프로그램을 분석할수 있다.

이 작업은 POPL'11 에서 발표될 예정입니다.

ROSAEC center
Research On Software Analysis for Error-free Computing
소프트웨어 무결점 연구센터 NRF ERC

ROPAS Programming Research Laboratory