GMeta : 데이터타입 제너릭 프로그래밍 기법을 Coq 증명에 활용한 라이브러리

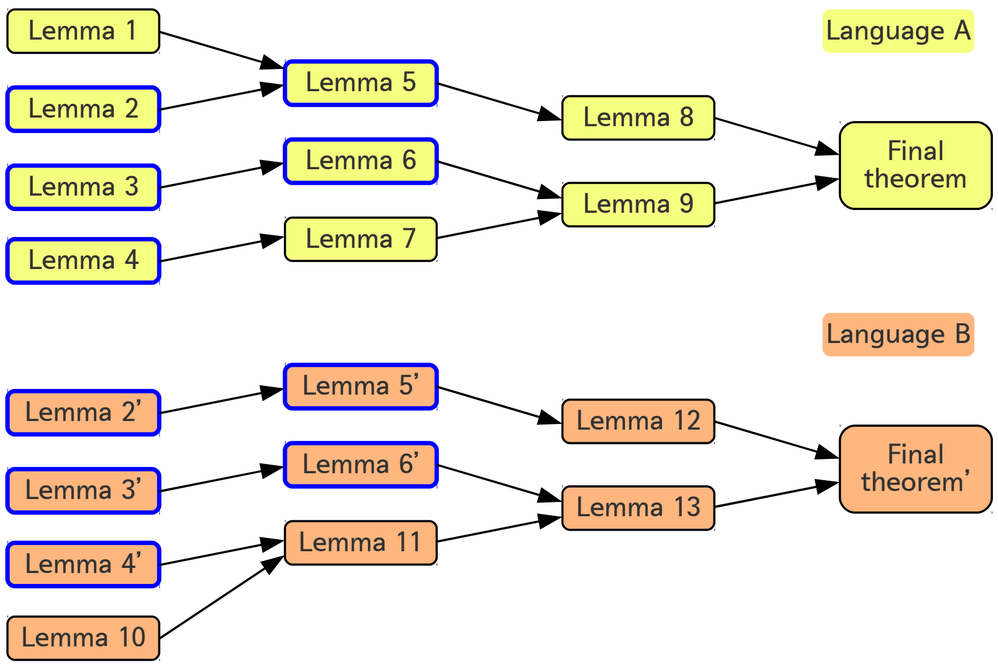
이계식, Bruno C. d. S. Oliveira, 조성근
서울대학교 프로그래밍 연구실

배경

- Coq이란 타입이론을 기반으로 하는 증명 보조기(proof assistant)입니다. 어떠한 프로그래밍 언어가 있을 때 관련된 어떤 성질을 Coq에서 증명할 수 있어요.
- 대부분의 프로그래밍 언어는 변수 정의하기(variable binding)를 포함하고 있어요.

문제

- 대상이 되는 프로그래밍 언어가 바뀔 때마다 변수 정의하기(binding)에 관련된 비슷한 성질의 증명들이 필요해요. 언어에 따라서 증명과정이 비슷하기는 하지만 같지는 않아요.



- 반복되는 비슷한 정리들을 한 번만 제너릭하게 증명하고, 증명된 정리들을 다시 이용하고자 합니다. 특히 변수 정의하기(binding)에 관련된 정리들을 말이에요.

사용된 기술

- 데이터타입 제너릭 프로그래밍(DGP) 기술
  - 유니버스(universe) - 데이터타입들을 표현하는 문법을 정하고,
  - 제너릭 함수/정리 - 그 데이터타입들의 원소를 다루는 제너릭 함수를 정의하거나 제너릭 정리를 증명합니다.
  - 예) Data Rep = 1 | Rep + Rep | Rep x Rep | K Rep | R
    RNat : Rep RNat = 1 + R
    RList : Rep RList = 1 + K RNat x R

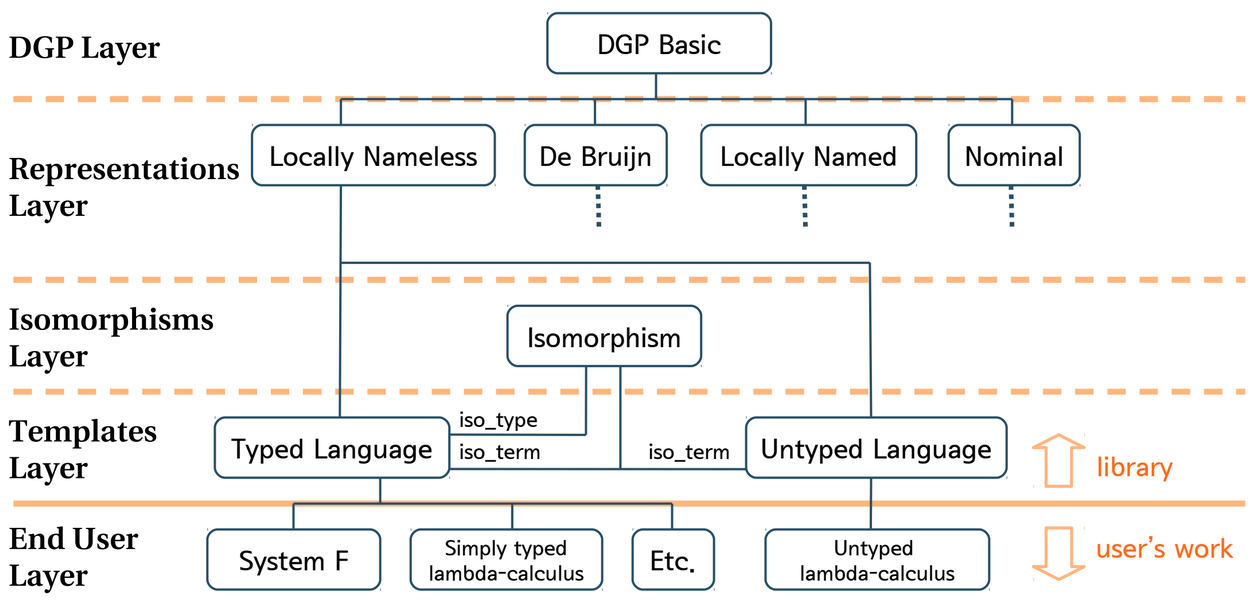
- 모듈 프로그래밍(modular programming) 기술
  - 변수 정의하기(binding)의 대표적인 네 가지 방법을 모듈을 이용해서 표현합니다.
  - 예) Nominal $\lambda x\ .\ x\ y$
    Locally named $\lambda x\ .\ x\ a$
    Locally nameless $\lambda\ .\ 0\ a$
    De Bruijn $\lambda\ .\ 0\ 1$

GMeta 라이브러리

- 사용자들에게 변수 정의하기(binding)에 관련된 Coq 정의와 Coq 증명을 제공하는 라이브러리

- 사용 과정
  1. 대상 언어의 문법 정의
  2. GMeta에서 정의된 언어와의 동일성(isomorphism) 정의
     이 과정은 라이브러리에서 제공하는 자동화 도구를 사용 가능
  3. 라이브러리로부터 보조 정의/증명들을 불러옴
  4. 불러온 보조 정의/증명들을 이용하여 최종 정리를 증명

- 라이브러리의 구조



얼마나 편해졌을까요?

- Locally Nameless

공통된 연산과 그에 관련된 정리들

		boilerplate	total	ratio
STLC	Aydemir et al.	17	31	55%
	GMeta basic	7	21	33%
	GMeta adv.	1	15	7%
System F<:	Aydemir et al.	60	93	65%
	GMeta basic	25	58	43%
	GMeta adv.	11	45	24%

- De Bruijn

STLC	Vouillon	7	26	27%
	GMeta basic	0	19	0%
System F<:	Vouillon	37	101	37%
	GMeta basic	2	65	3%

- GMeta를 이용하면 전체 증명 중 반복적인, 귀찮은 증명의 비율이 줄어듭니다. 이제 중요한 증명에 집중하면 되겠죠?

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

http://ropas.snu.ac.kr/gmeta/

ROPAS 서울대학교 프로그래밍 연구실