# 보다 친숙한 Coq 증명을 위한 방안 연구

이계식

서울대 , ROPAS
ROSAEC Center 4th Workshop

# Outline

1. More about GMeta

2. Easier Access to Coq

# What You Usually Start with

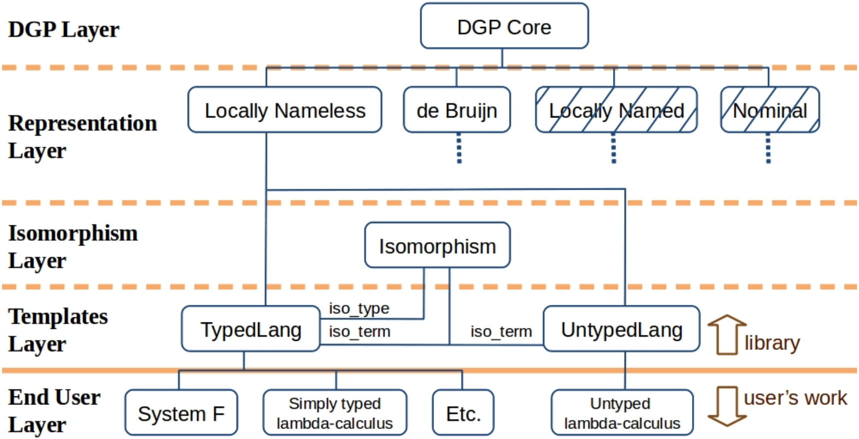
- Choice of a theorem prover
  - Coq, Isabelle\HOL, Agda, ACL2, Nuprl, PVS, Mizar, ...
- Choice of a representation style
  - de Bruijn indices
  - Locally nameless approach
  - Locally-named approach
  - Nominal approach
  - Higher-Order Abstract Syntax
  - ...
- Specification of the target language
- There are many other choices to be made.

# Four 1st-order Representation Styles

## Representation of $\lambda x\,.\,x\,y$

| | | |
|---|---|---|
| Nominal | $\lambda x\,.\,x\,y$ | where $x, y \in V$ |
| de Bruijn | $\lambda\,.\,0\,1$ | |
| Locally nameless | $\lambda\,.\,0\,y$ | where $x, y \in FV$ |
| Locally-named | $\lambda x\,.\,x\,a$ | where $x \in BV$ and $a \in FV$ |

# GMeta Library



What does GMeta say besides saving some boilerplate?

# Immediate Goal

- A slight extension of DGP core
  - to make the expressions more familiar,
  - to deal with languages without variables in a general way
  - to include systems from mathematics
- Extension of the meta-level library
  - Quantification style in locally nameless approach

# Locally nameless vs. locally-named

## Term: Locally nameless

```
Inductive trm : Set :=
| trm_bvar : nat -> trm
| trm_fvar : var -> trm
| trm_abs : typ -> trm -> trm
| trm_app : trm -> trm -> trm.
```

$(\lambda\ T.\ 0\ \ y)$

## Term: Locally-named

```
Inductive trm : Set :=
| trm_bvar : nat -> trm
| trm_fvar : var -> trm
| trm_abs : nat -> typ -> trm -> trm
| trm_app : trm -> trm -> trm.
```

$(\lambda\ x : T\ .\ x\ \ y)$

# DGP Core

### Universe of Representations

```
Inductive Rep : Type :=
| UNIT   : Rep
| CONST : Rep -> Rep
| REPR   : Rep -> Rep
| PLUS   : Rep -> Rep -> Rep
| PROD   : Rep -> Rep -> Rep
| BIND   : Rep -> Rep -> Rep
| REC    : Rep.
```

### iso_term

```
PLUS (PROD Rtyp (BIND REC REC)) (PROD REC REC)
```

# Better is ...

- `BIND : Rep -> Rep -> Rep -> Rep`

- `PLUS (PROD Rtyp (BIND REC REC)) (PROD REC REC)`

# Mid-Term Goal

- Development of the library for locally-named approach
- Extension of DGP core with multi-binders and mutually inductive definitions

# Long-Term Goal

- Development of the library for nominal approach
- Other well-known representation styles should be included.

# About GMeta library

- Are all these goals related the issue of easier access to Coq?
- The answer is implicitly there in the structure of GMeta library.
- Using GMeta: It is not just about saving boilerplate.
- The structure and contents of GMeta library shows you what to do when you want to use Coq for a formalization of something.
- And more:
  - quantification style,
  - presentation of environment,
  - look-up function, etc.

# Access to Coq

- How much should I pay to learn Coq?
- I learned Coq and wish to do some formalization. Now what to do?
- Which choice should I do? Any criteria?

# GMeta and Some Criteria for Formalization

- Cost of entry
  - how much does a user need to know in order to successfully develop a formalization
- Difficulty
  - in defining syntax and proving properties
  - POPLmark
- Efficiency
  - in handling of definitions and proofs
  - Appel and Leroy's CIVmark
- Transparency
  - how intuitive a formalization technique is.

GMeta could be helpful in these respects.

# GMeta and Some Criteria for Formalization

- Cost of entry
  - how much does a user need to know in order to successfully develop a formalization
- Difficulty
  - in defining syntax and proving properties
  - POPLmark
- Efficiency
  - in handling of definitions and proofs
  - Appel and Leroy's CIVmark
- Transparency
  - how intuitive a formalization technique is.

GMeta could be helpful in these respects.

# Current Work

Focus: easier and conventional approach

- Formalization of AI framework (jointwork with Sungkeun Cho, Kwangkeun Yi, and others)
- Formalization of context-free type systems (jointwork with Sungwoo Park)
- Thinking of nominal approach in Coq (extension of GMeta library)

# Current Work

Focus: easier and conventional approach

- Formalization of AI framework
(jointwork with Sungkeun Cho, Kwangkeun Yi, and others)
- Formalization of context-free type systems
(jointwork with Sungwoo Park)
- Thinking of nominal approach in Coq (extension of GMeta library)

Thank you for listening!

Questions and Comments?