GMeta의 평가와 보다 쉬운 정형증명에 대한 연구

이계식

한경대학교

ROSAEC Workshop, 2012년 1월 16 - 19일

내용 소개

GMeta 평가

GMeta 평가

- 공식
  - 나름 괜찮은 평에도 불구하고 POPL '11, ICFP '11, CPP '11로부터 거절됨
  - 절대적 호평과 함께 ESOP '12에 간택됨

    GMeta: A Generic Formal Metatheory Framework for First-Order Representations
    (공동연구: Bruno Oliveira, Sungkeun Cho, Kwangkeun Yi)

- 비공식
  - ITP '11에서의 관련 질문
  - 11년 12월 INRIA Coq팀 방문시 발표에서의 관심 (P. Martin-Löf 포함)

GMeta 평가

- 공식
  - 나름 괜찮은 평에도 불구하고 POPL '11, ICFP '11, CPP '11로부터 거절됨
  - 절대적 호평과 함께 ESOP '12에 간택됨

    GMeta: A Generic Formal Metatheory Framework for First-Order Representations
    (공동연구: Bruno Oliveira, Sungkeun Cho, Kwangkeun Yi)

- 비공식
  - ITP '11에서의 관련 질문
  - 11년 12월 INRIA Coq팀 방문시 발표에서의 관심
    (P. Martin-Löf 포함)

평가 의미

- 연구 성과 인정 받음
- 새롭게 시작할 수 있는 계기 마련
- 확장 연구에 대한 본격적 고민 시작

GMeta 간략 소개

– 보다 쉬운 정형증명을 위해 –

정형증명을 위한 준비과정

Scott Lee 교수님 발표에서 구체적인 예를 이용하여 자세히 설명됨:

- 대상 언어의 명세에 대한 고민
- 정리증명기/증명보조기 선택
  - Coq, Isabelle\HOL, Agda, PVS, etc.
- Variable binding (변수묶기) 문제가 발생하는 경우 원하는 표현법 선택
  - de Bruijn indices, locally nameless, locally-named, nominal, HOAS 등등
- 이외에도 많은 크고 작은 선택 과정을 거쳐야 함
  - 개발 과정에서 다양한 시도들이 요구됨

정형증명을 위한 준비과정

Scott Lee 교수님 발표에서 구체적인 예를 이용하여 자세히 설명됨:

- 대상 언어의 명세에 대한 고민
- 정리증명기/증명보조기 선택
  - Coq, Isabelle\HOL, Agda, PVS, etc.
- Variable binding (변수묶기) 문제가 발생하는 경우 원하는 표현법 선택
  - de Bruijn indices, locally nameless, locally-named, nominal, HOAS 등등
- 이외에도 많은 크고 작은 선택 과정을 거쳐야 함
  - 개발 과정에서 다양한 시도들이 요구됨

판단 기준

- 기본 지식 및 경험 등 시작을 위해 필요한 조건
- 구현 과정의 어려운 정도 및 효율성
- 사용된 기술의 직관성

GMeta의 출발점을 이룸

판단 기준

- 기본 지식 및 경험 등 시작을 위해 필요한 조건
- 구현 과정의 어려운 정도 및 효율성
- 사용된 기술의 직관성

GMeta의 출발점을 이룸

GMeta 출발점

앞서의 판단 기준을 만족시키는 방법으로

- 다양한 언어와
- 다양한 표현방식을

일괄적으로 다룰 수 있도록 datatype-generic programming(DGP)과 modular-programming 기술을 활용함

- DGP 이용: 많은 인프라를 DGP universe 상에서 한 번에 해결
- 모듈 이용: DGP universe 상에서 정의된 것들과 또는 증명된 사실들을 특정한 언어의 세계로 끌어 내림

GMeta 출발점

앞서의 판단 기준을 만족시키는 방법으로

- 다양한 언어와
- 다양한 표현방식을

일괄적으로 다룰 수 있도록 datatype-generic programming(DGP)과 modular-programming 기술을 활용함

- DGP 이용: 많은 인프라를 DGP universe 상에서 한 번에 해결
- 모듈 이용: DGP universe 상에서 정의된 것들과 또는 증명된 사실들을 특정한 언어의 세계로 끌어 내림

대표적 표현 방식

Representations	$\lambda x.(y\,x)$
Nominal	$\lambda x.(y\,x)$
de Bruijn indices	$\lambda.(1\,0)$
Locally nameless	$\lambda.(a\,0)$
Locally named	$\lambda x.(a\,x)$

현재 지원 방식: de Bruijn indices와 locally nameless 방식

대표적 표현 방식

Representations	$\lambda x.(y\,x)$
Nominal	$\lambda x.(y\,x)$
de Bruijn indices	$\lambda.(1\,0)$
Locally nameless	$\lambda.(a\,0)$
Locally named	$\lambda x.(a\,x)$

현재 지원 방식: de Bruijn indices와 locally nameless 방식

GMeta 영향

- 기본적으로 필요한 인프라
  - free 또는 bound variables, substitutions, α-conversion, 등등
  - 위 함수들과 연관된 여러 중요 성질 증명
- Aydemir et al. (POPL '08 논문)과 비교:

	locally nameless	기본 인프라	전체
STLC	GMeta / Aydemir et al.	87.5 %	45%
$F_{<:}$	GMeta / Aydemir et al.	82 %	56%

GMeta 영향

- 기본적으로 필요한 인프라
  - free 또는 bound variables, substitutions, α-conversion, 등등
  - 위 함수들과 연관된 여러 중요 성질 증명
- Aydemir et al. (POPL '08 논문)과 비교:

	locally nameless	기본 인프라	전체
STLC	GMeta / Aydemir et al.	87.5 %	45%
$F_{<:}$	GMeta / Aydemir et al.	82 %	56%

GMeta 확장 연구

우선 목표

- 버젼 업그레이드(v8.2 $\Rightarrow$ v8.4)
  - Dependent type programming 관련 많은 변화 이루어짐
- Locally named 방식 지원
  - Locally nameless 방식과 비슷
  - α-conversion 때문에 다소 시간을 요할 수 있음
  - 최근에 소개된 Sato-Pollack의 canonically locally-named 방식 활용 고민중

우선 목표

- 버젼 업그레이드(v8.2 $\Rightarrow$ v8.4)
  - Dependent type programming 관련 많은 변화 이루어짐
- Locally named 방식 지원
  - Locally nameless 방식과 비슷
  - α-conversion 때문에 다소 시간을 요할 수 있음
  - 최근에 소개된 Sato-Pollack의 canonically locally-named 방식 활용 고민중

다음/병행 목표

- Nominal 방식 지원
  - Coq과 관련되어 어떤 연구도 이루어지지 않았음
  - 이유 추정: 할 일이 너무 많음
  - 하지만 GMeta의 존재 이유중 하나임
- A. Stoughton의 1988년 논문 “Substitution Revisited”라는 논문 아이디어 활용 가능성 타진
  - meta 차원에서 행해지는 내용을 syntax 차원에서 다루는 방안의 가능성을 제시한 논문
  - 매우 간단하지만 아무도 주목하지 않은 아이디어임
  - 자체적으로 의미를 가지는 연구이면서 GMeta의 확장에 기본으로 활용될 것임
- 저널용 논문 쓰기

다음/병행 목표

- Nominal 방식 지원
  - Coq과 관련되어 어떤 연구도 이루어지지 않았음
  - 이유 추정: 할 일이 너무 많음
  - 하지만 GMeta의 존재 이유중 하나임
- A. Stoughton의 1988년 논문 “Substitution Revisited”라는 논문 아이디어 활용 가능성 타진
  - meta 차원에서 행해지는 내용을 syntax 차원에서 다루는 방안의 가능성을 제시한 논문
  - 매우 간단하지만 아무도 주목하지 않은 아이디어임
  - 자체적으로 의미를 가지는 연구이면서 GMeta의 확장에 기본으로 활용될 것임
- 저널용 논문 쓰기

다음/병행 목표

- Nominal 방식 지원
  - Coq과 관련되어 어떤 연구도 이루어지지 않았음
  - 이유 추정: 할 일이 너무 많음
  - 하지만 GMeta의 존재 이유중 하나임
- A. Stoughton의 1988년 논문 “Substitution Revisited”라는 논문 아이디어 활용 가능성 타진
  - meta 차원에서 행해지는 내용을 syntax 차원에서 다루는 방안의 가능성을 제시한 논문
  - 매우 간단하지만 아무도 주목하지 않은 아이디어임
  - 자체적으로 의미를 가지는 연구이면서 GMeta의 확장에 기본으로 활용될 것임
- 저널용 논문 쓰기

이후 목표

- 보다 다양한 언어 지원
  - DGP universe 확장
  - Mutual induction, record types 등등 지원 가능
  - 꽤 어려울 수 있음: Coq이 지원하는 dependent type programming 기술을 최대한 활용해야 할 것으로 기대
  - v8.3 이후 Coq의 지원이 보다 강화됨 (버젼 업의 본질적 이유)
- GMeta의 변화시도
  - PL에서 기존에 연구된 방식을 theorem proving 방식에 활용
  - 예) Gonthier et al.의 “How to Make Ad Hoc Proof Automation Less Ad Hoc” (ICFP ’11)에서 활용된 type classe를 활용한 증명 기술 접목

이후 목표

- 보다 다양한 언어 지원
  - DGP universe 확장
  - Mutual induction, record types 등등 지원 가능
  - 꽤 어려울 수 있음: Coq이 지원하는 dependent type programming 기술을 최대한 활용해야 할 것으로 기대
  - v8.3 이후 Coq의 지원이 보다 강화됨 (버젼 업의 본질적 이유)
- GMeta의 변화시도
  - PL에서 기존에 연구된 방식을 theorem proving 방식에 활용
  - 예) Gonthier et al.의 “How to Make Ad Hoc Proof Automation Less Ad Hoc” (ICFP ’11)에서 활용된 type classe를 활용한 증명 기술 접목

그이후 목표

- Nominal Coq
  - C. Urban의 Nominal Isabelle에 상응하는 Coq plugin 개발
  - 아직 아무도 시도 안함
  - 많은 관심을 불러 일으킬 것으로 기대
  - 한계: 개발 경험 부족

기타 Coq 관련 프로젝트

Reverse Math in Coq

- 한-프 국가간협력기반조성사업 과제 신청
- 한국측: 박성우 교수팀 참여, 프랑스측: 인리아 Coq 개발팀장 참여
- 국문과제명: 수학자들이 사용하기 쉬운 증명보조기 소프트웨어 개발

Reverse Math in Coq (cont.)

- 연구목표:
  - 프로그래밍을 잘 모르는 수학자/논리학자들이 보다 쉽게 접근할 수 있는 증명보조기 개발
- 배경 및 기대성과
  - Reverse mathematics는 수학자/논리학자들이 진행 (Coq의 바탕이론인 CIC보다 훨씬 약한 논리 활용)
  - Coq과 같은 자동증명기는 주로 전산학자들이 사용
  - 두 분야의 깊은 연관성을 유기적으로 구현하는 증명보조기 개발
  - 수리논리와 PL의 새로운 형태의 학문융합 기대

감사합니다!