Національний технічний університет України
“Київський Політехнічний Інститут імені Ігора Сікорського”

Концептуальна модель системи доведення теорем

Студент PhD другого року навчання — Максим Сохацький
Науковий керівник — Павло Маслянко

Спеціальності:
113 — Прикладна математика
124 — Системний аналіз

Кафедра прикладної математики 2018

Об'єкт дослідження

Мови програмування, середовища виконання, верифікатори моделей, системи доведення теорем, SMT-солвери

1) мови для сертифікації та специфікації (Z, UML);
2) системи верифікації ПЗ (TLA+, Twelf, Dedukti, Z3);
3) мови програмування (Haskell, OCaml, Erlang, Scala, LISP);
4) системи доведення теорем (Agda, Coq, HOL, ACL2);
5) уніфіковані середовища виконання (HaLVM, LING, Mirage);
6) їх поєднання, формальна система виконання, верифікації та валідації програмного забезпечення як концептуальна система доведення теорем

Актуальність дослідження

Математична верифікація алгоритмів для унеможливлення широкого класу помилок для критичних галузей як основна мотивація роботи

1) Mars Climate Orbiter (1998), перетворення брит/метр — 80 млн;
2) Ariane Rocket (1996), кастинг з 64 до 16 біт — 500 млн;
3) Помилка в FPU в перших Pentium (1994) — 300 млн;
4) Помилка у логіці бізнес-контрактів EVM — 50 млн;
5) Помилка в SSL (heartbleed) — 400 млн.

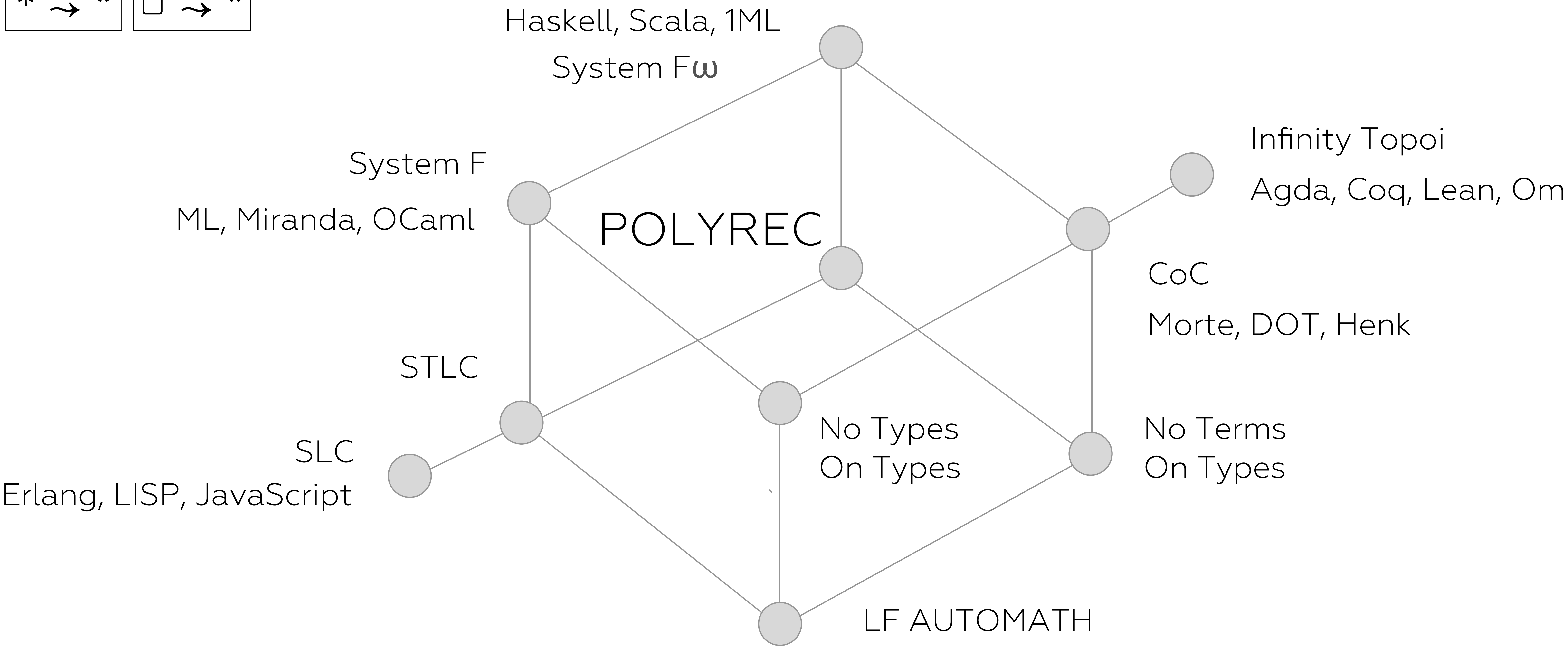
1) IEEE Std 1012-2016 — V&V Software verification and validation;
2) ESA PSS-05-10 1-1 1995 — Guide to software verification and validation;
3) ISO/IEC 13568:2002 — Z formal specification notation.

Мови програмування

в розширеному Лямбда-кубі Барендрехта

CoC:	$* \to *$	$\square \to *$	$* \to \square$	$\square \to \square$
Fω:	$* \to *$	$\square \to *$		$\square \to \square$
F:	$* \to *$	$\square \to *$		

4



Мови середовища виконання

Через призму інженерії

JIT	Interpreters	LLVM	Non-LLVM
LuaJIT	K	Rust	OCaml
V8	LING/Erlang	Julia	GHC
SpiderMonkey	O	C/C++	
EDGE			
JVM/HotSpot			
CLR			

Вищі мови програмування

Для доведення теорем та верифікації моделей

Target	Class	Higher Language	Type Theory
CPU	Non-LLVM	Spiral	System F
JVM	JIT	Scala	System F-omega
GHC	Non-LLVM	Morte	CoC
Erlang	Interpreter	Om	PTS-infinity
O	Interpreter	Om	PTS-infinity
Haskell	Extract	Coq/Agda	CiC

Предмет дослідження

Концептуальна модель системи доведення теорем на основі Теорії типі Пера Мартіна-Льофа

Основною частиною предмета дослідження такої системи мов є теорія типів, яка вивчає обчислювальні властивості мов. Теорія типів виділилася в окрему науку Пером Мартіном-Льофом як запит на вакантне місце у трикутнику теорій, які відповідають ізоморфізму Каррі-Говарда-Ламбека (Логіки, Мови, Категорії). Інші дві це: теорія категорій та логіка вищих порядків. Друга частина предмета дослідження є вираження концептуальної системи доведення теорем використовуючи теорію типів як основний інструмент.

Всесвіти теорії типів

Інфініті Топос

$U_0 : U_1 : U_2 : U_3 : \ldots \infty$

U_0 — propositions
U_1 — sets
U_2 — types
U_3 — sorts

$S\ (n : nat) = U\ n$
$A_1\ (n\ m : nat) = U\ n : U\ m$ where $m > n$ — cumulative
$R_1\ (m\ n : nat) = U\ m \longrightarrow U\ n : U\ (\max\ m\ n)$ — predicative

$A_2\ (n : nat) = U\ n : U\ (n + 1)$ — non-cumulative
$R_2\ (m\ n : nat) = U\ m \longrightarrow U\ n : U\ n$ — impredicative

$Prop = Large\ \Omega_0 = U_0$

$\Sigma = Large\ \Omega_2 = U_2$

П Тип

Inductive Type, AST, Logical Framework

```
data O₁ := U : nat → O₁
         | Var: Ident → O₁
         | App: O₁ → O₁ → O₁
         | Lambda: Binder → O₁ → O₁ → O₁
         | Arrow: O₁ → O₁ → O₁
         | Pi: name → O₁ → O₁ → O₁.

record Pi (A: Type) :=
    intro: (A → Type) → Type :=
    fun: (B: A → Type) → ∀ (a: A) → B a → intro B
    app: (B: A → Type) → intro B → ∀ (a: A) → B a
    app-fun (B: A → Type) (f: ∀ (a: A) → B a): ∀ (a: A) → app (fun f) a = f a
    fun-app (B: A → Type) (p: intro B): fun (λ (a: A) → app p a) = p
```

Σ Тип

Syntax and Model

```
data O₂ := O₁
          | Sigma: name → O₂ → O₂ → O₂
          | Pair: O₂ → O₂ → O₂
          | Fst: O₂ → O₂
          | Snd: O₂ → O₂.
```

$\Sigma\ x : A, B\ x : U$ — formation rule

$\mathrm{pair}\ (x : A)\ (y : B\ x)$ — introduction

$\mathrm{pr}_1\ s : A$ — elimination

$\mathrm{pr}_2\ s : B\ x$ — elimination

```
data   Sigma (A: Type) (P: A -> Type) (x: A): Type =
        intro: P x -> Sigma A P
```

Формалізація завдання

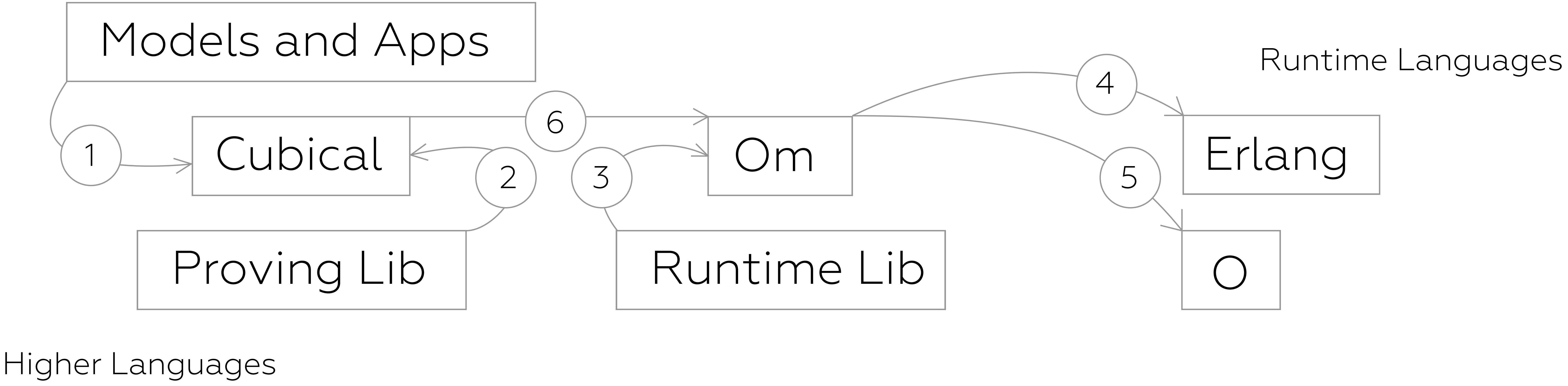
Побудова системи доведення теорем

Задачою цього дослідження є розробка концептуальної моделі системи доведення теорем та її реалізації для побудови ефективного циклу верифікації програмного забезпечення та доведення теорем.

1) формалізація середовища виконання;
2) ієрархія мов як протоколів системи доведення теорем;
3) уніфікована базова бібліотека;
4) інтегрування мов та концептуальна модель системи доведення теорем.

Процес верифікації моделей

Динамічне представлення концептуальної моделі системи доведення теорем



[3,4] cover the presented work, [1,2,5,6] cover the future works.

Публікації дослідження

From proving to Extraction, Linking and Running

1. Стаття про мову середнього рівня Om (Кембрідж)
2. Стаття про Рівність (Equality Type and Its derivability)
3. Стаття про мову високого рівня Infinity
4. Сттатя про мову низького рівня O (інтерпретатор)
5. Стаття про F-Алгебри та рекурсивні схеми
6. Стаття про модель індуктивних типів через W-Types
7. Стаття про індуктивно-рекурсивне моделювання IR-Types
8. Стаття про ізоморфізми шляхів як приклад потужності мови Infinity
9. Стаття про базову бібліотеку до мови Infinity
10. Загальний опис мов та концептуальна модель роботи

Структура моделі

Компоненти, протоколи та мови у структурному представленні концептуальної системи доведення теорем

1) Models

- IR/II
- Bohm
- HoTT

2) Core — Infinity Language

- Model Verification
- Normalization
- Bidirectional Checking

- Pure Type System (Om)
- Identity
- Induction
- Homotopy Interval [0,1]

3) Extraction

- LLVM
- Interpreters
- Detyping
- Linking
- Optimization

4) Runtimes

- O
- Erlang
- V8
- JVM

Структура диссертації

From proving to Extraction, Linking and Running

Відповідність статей до структури дисертації:

0 Глава. Вступ. Глава 10
1 Глава. Om Стаття 1
2 Глава. Infinity Стаття 3
Моделі індуктивних типів. Статті 5, 6, 7, 8
3 Глава. Базова бібліотека Infinity. Стаття 9
4 Глава. O Стаття 4

Бібліографія

Groupoid Infinity

1. Barendregt. The Lambda Calculus with Types http://5ht.co/pts.pdf
2. Martin-Löf. Intuitionistic Type Theory http://5ht.co/mltt.pdf
3. Awodey. Category Theory http://5ht.co/cat.pdf
4. Jacobs. Categorical Logic http://5ht.co/fibrations.pdf
5. Streicher. The groupoid interpretation of TT http://5ht.co/groupoid.pdf
6. Voevodsky et all. Homotopy Type Theory http://5ht.co/hott.pdf
7. Huber, Coquand. Cubical Type Theory http://5ht.co/cubicaltt.pdf