Automatically Finding Theory Morphisms for Knowledge Management

Dennis Müller[1] Florian Rabe[1,2] Michael Kohlhase[1]

Computer Science, FAU Erlangen-Nürnberg

LRI, Université Paris Sud

August 13, 2018

Introduction

Motivation

Formal methods in mathematics are succeeding!

⇒ Reached new problems at larger scales

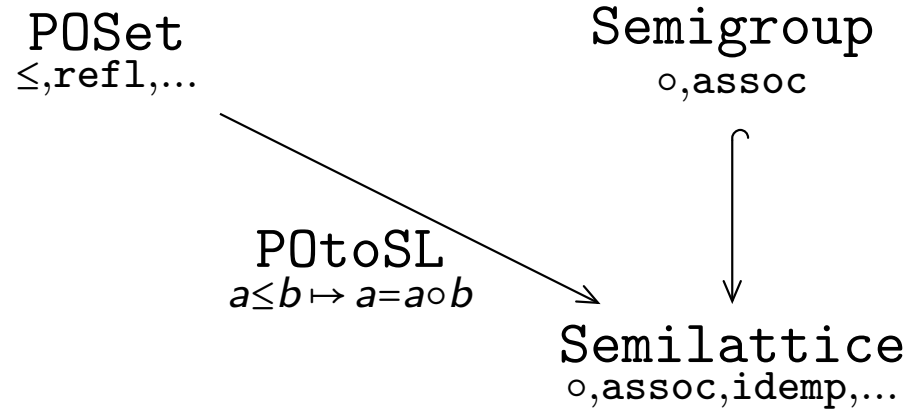
- Interoperability between systems
- Huge libraries Difficult to get an overview of all their contents
- Knowledge Discovery / Search

⇒ Non-local problems

Need automated methods!

Theories and Views

Modularity helps with managing large libraries



Theories are sets of constants with types (can include other theories) Simplified

Views map constants in one theory to expressions over another theory Truth-preserving (If $t : T$, then $v(t) : v(T)$)

Views

Views are great concept for representing non-local relations between concepts

A **total** view $V : A \to B$ means:

- B is a model of A
- B is an example for A
- A is a generalization of B

  B could be refactored as an extension of A
- Theorems/Definitions over A are valid over B

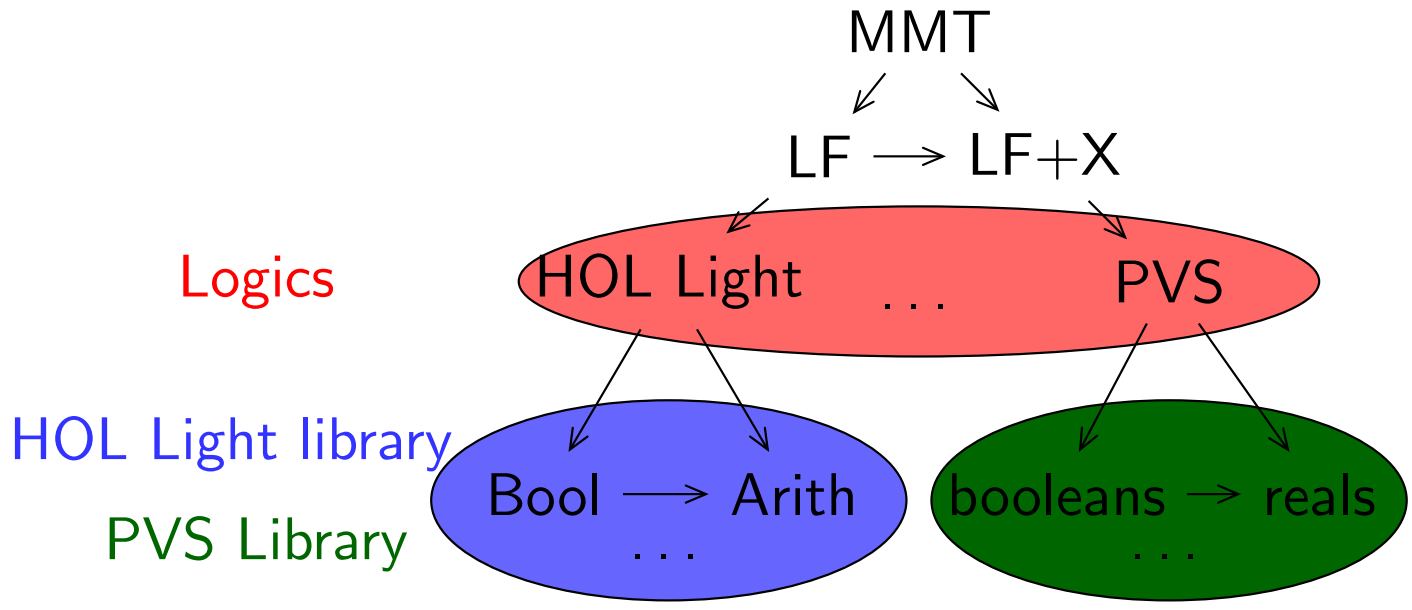
A **partial** view $V : A \to B$ means:

- B is potentially an interesting counterexample for A
- A and B have a common subtheory

  A and B could be refactored as extensions of $A \cap B$

$\Rightarrow$ Automated viewfinding helps with non-local knowledge management problems

MMT: A General Framework for Formal Libraries



- Foundation-independent
  ⇒ Foundations, logics, logical frameworks all formalized as theories
- Importers for various formal libraries (OAF)
  HOLLight, Mizar, PVS, TPTP, Imps...

⇒ We can now study inter-library knowledge management problems generically in a unified framework!

Finding Views Efficiently

Finding Views is Difficult!

Viewfinding between two collections of theories is computationally expensive:

- Finding complex views subsumes theorem proving
  Equality of expressions, typing judgments - "math complete"
- Number of candidate theories quadratic over number of total theories
- Number of candidate views between two theories infinite in general Even canonical candidates exponential (n^m)

⇒ No efficient, accurate viewfinding methods feasible

PVS: ≈800 theories

But: Efficiency often more relevant than accuracy

⇒ Special case first: reduce viewfinding to **simple** views and syntactical heuristics only

Only map constants to constant symbols directly

Our Algorithm

Step 1: Normalize theories
Logic normalizations, definition expansions, droping implicit arguments,. .

Step 2: Compute hashed representation of constants (types) commutative with viewfinding
Here: Abstract syntax trees(t, ℓ), where ℓ is a list of symbol occurences

Step 3: Two constants can be matched in a (partial) view, if their abstract syntax trees t_1, t_2 are equal and (recursively) the symbols in ℓ_1, ℓ_2 are pairwise matched.

Yields dependency-closed partial views

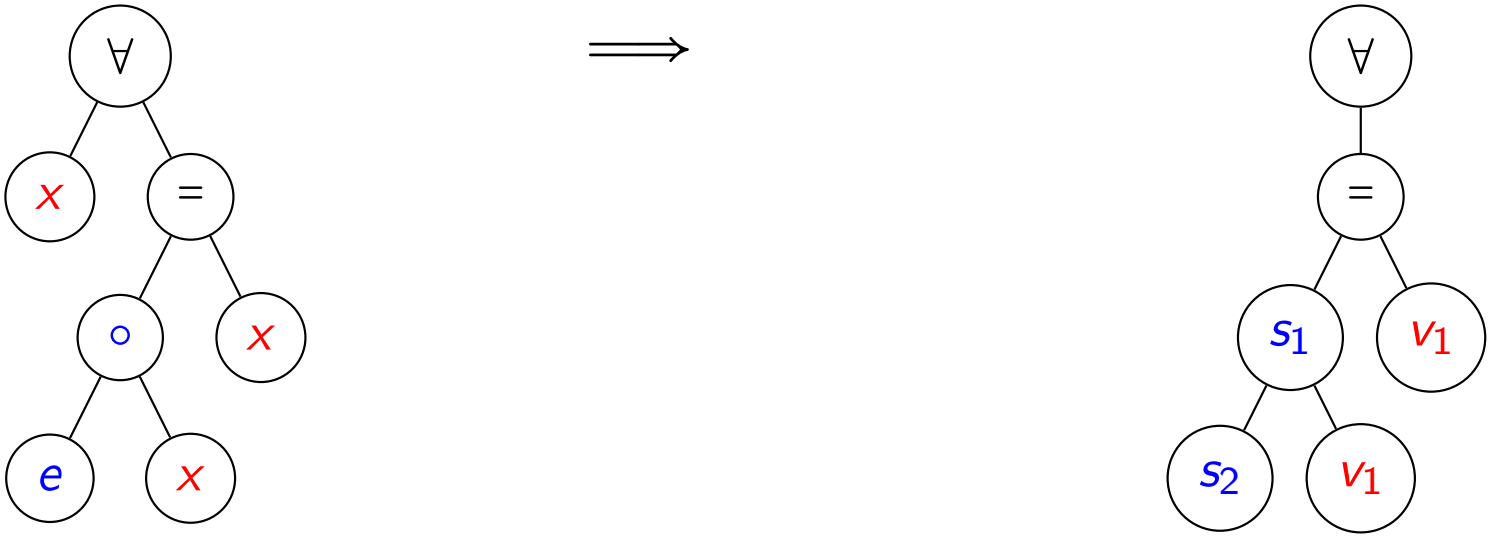
Step 4: Two partial views (obtained from previous step) can be merged, if they do not disagree on any matches.

Abstract Syntax Trees

Preselect potential pairs of constants by computing an **abstract syntax tree** (t, ℓ) using DeBruijn-Indices and enumerating symbol references:

For a constant of type $\forall x\ e \circ x = x$:

Assume $\forall$ and $=$ are provided by a meta-theory



$\Rightarrow t = \forall(=(s_1(s_2, v_1), v_1)) \quad \ell = (\circ, e)$

Example

$C_1 \ : \ \forall x : \text{set}\ \forall y : \text{set}\ P(x) \wedge y \subseteq_1 x \Rightarrow P(y)$
$C_1 \ : \ \forall x : \text{powerset}\ \forall y : \text{powerset}\ Q(x) \wedge y \subseteq_2 x \Rightarrow Q(y)$

$t_1 = t_2 = \forall\{s_1\}(\forall\{s_2\}(\Rightarrow(\wedge(s_3(v_2), s_4(v_1, v_2)), s_5(v_1))))$
$\ell_1 = (\text{set}, \text{set}, P, \subseteq_1, P)$
$\ell_2 = (\text{powerset}, \text{powerset}, Q, \subseteq_2, Q)$

since $t_1 = t_2$ we recursively (try to) match
$\text{set} \mapsto \text{powerset},\ P \mapsto Q\ \subseteq_1 \mapsto \subseteq_2$, yielding the partial view

$$C_1 \mapsto C_2,\ \text{set} \mapsto \text{powerset},\ P \mapsto Q\ \subseteq_1 \mapsto \subseteq_2$$

Given a second partial view that agrees on all assignments
$D_1 \mapsto D_2,\ \text{set} \mapsto \text{powerset},\ R \mapsto S$, we can form the union

$$C_1 \mapsto C_2,\ D_1 \mapsto D_2,\ \text{set} \mapsto \text{powerset},\ P \mapsto Q\ \subseteq_1 \mapsto \subseteq_2,\ R \mapsto S$$

Optimizations

Still inefficient: Lots of spurious matches - interesting results buried under noise (any two types, binary connectives,. . .)

- *Biasing:* Start matching only with e.g. axioms (i.e. other symbols covered only during recursion)
  Assures matched symbols share at least one property
- Set of symbols to be fixed (e.g. equality, quantifiers and logical connectives above) can be extended
  Currently: Symbols from meta-theory
- Using maximal theories only
  Included theories are covered by elaborating includes
- Fix *aligned* symbols
  two symbols informally deemed "the same"

Demonstration

Future Work

This is only the first step!

- Are there better hashed representations?
  Substitution Tree Indices?
- Sufficiently general normalization techniques
  Elimination of language features
- Combination of various approaches
  Kaliszyk et al: Machine learning for finding Alignments

⇒ Use automated theorem proving?
  at least in special cases? For specific applications?

- Specialized user interfaces for different applications