Morphism Equality in Theory Graphs

Florian Rabe and Franziska Weber

September 2023

Motivation

Theory Graphs

Overview

- Diagrams in the category of theories and theory morphisms
- Generate category of nodes and paths
- Applicable to virtually any formal language
- Great for high-level structure of large libraries
- MMT: language-independent system for theory graphs

Example: an isomorphism of FOL-theories

$$\begin{tikzcd}
 & \mathtt{FOL} \arrow[dl, hook] \arrow[dr, hook] & \\
\mathtt{Group}\ (U, \circ, e, ^{-1}) \arrow[rr, bend left, "\mathtt{GtoDG}"] & & \mathtt{DivGroup}\ (U, /, e) \arrow[ll, bend left, "\mathtt{DGtoG}"]
\end{tikzcd}$$

Example: Isomorphism of $\mathtt{Group}$ and $\mathtt{DivGroup}$ in MMT

theory $\mathtt{Carrier} =$ **include** $\mathtt{FOL}, U : \mathtt{tp}$

theory $\mathtt{Group} =$
include $\mathtt{Carrier}$

$$\begin{array}{lll} \circ & : & \mathtt{tm}\,U \to \mathtt{tm}\,U \to \mathtt{tm}\,U \\ e & : & \mathtt{tm}\,U \\ ^{-1} & : & \mathtt{tm}\,U \to \mathtt{tm}\,U \end{array}$$

theory $\mathtt{DivGroup} =$
include $\mathtt{Carrier}$

$$\begin{array}{lll} / & : & \mathtt{tm}\,U \to \mathtt{tm}\,U \to \mathtt{tm}\,U \\ e & : & \mathtt{tm}\,U \end{array}$$

(omitted: axioms)

morph $\mathtt{DGtoG} : \mathtt{DivGroup} \longrightarrow \mathtt{Group} =$
include $id_{\mathtt{Carrier}}$

$$\begin{array}{lll} / & = & \lambda x, y.\, x \circ y^{-1} \\ e & = & e \end{array}$$

morph $\mathtt{GtoDG} : \mathtt{Group} \longrightarrow \mathtt{DivGroup} =$
include $id_{\mathtt{Carrier}}$

$$\begin{array}{lll} \circ & = & \lambda x, y.\, x/(e/y) \\ e & = & e \\ ^{-1} & = & \lambda x.\, e/x \end{array}$$

(omitted: axioms mapped to proofs)

An Orthogonal Motivation: Realms

Realms — Carette, Farmer, Kohlhase; CICM 2014

- Merge equivalent axiomatizations of the same theory into one interface
  e.g., `Group`, `DivGroup`
  - use *any* when *creating* a model
  - use *all* when *using* a model
- Challenge for system development
  - critical in practical mathematics
  - not supported by current formal systems

Our attempt to implement — CICM WiP 2022

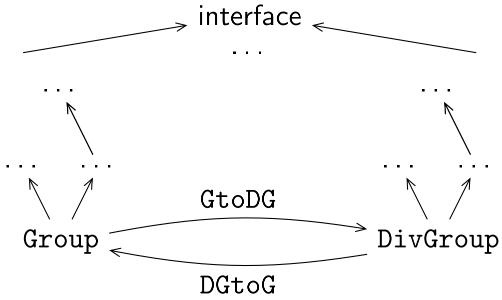
- 2014 definition very heavyweight — probably the reason why no one implemented it
- Our idea: find small set of language features that
  - are needed anyway
  - allow emulating realm-like structures

Simple Realms

Realms consists of

- set of equivalent formalizations
- for each a hierarchy of conservative extensions
- a theory merging all of the above into one interface

2014 definition vague about "equivalent", "conservative"



our observation:

- those theories are usually all isomorphic
- morphism equality critical

Diagrams in Category Theory

Diagram = Category Generator

- set of objects (= nodes)
- set of named morphisms (= edges)
- identity, composition induce morphisms (= paths)
- set of equalities between morphisms — often omitted

Practical systems

- implemented in various systems — MMT, Specware, Hets, Isabelle, PVS, ...
- usually without formalizing morphism equality

$\Rightarrow$ no direct formalization of
  - commutative diagrams
  - diagram chase
  - isomorphism

disclaimer: various *indirect* representations of morphisms do support equality

Adding Morphism Equality to MMT

MMT-Grammar

$$
\begin{array}{rll}
G & ::= \emptyset \mid G,\ \textbf{theory}\ s = \{D^*\} & \text{nodes} \\
  & \mid G,\ \textbf{morph}\ m : S \longrightarrow S = \{d^*\} & \text{edges} \\
D & ::= c[: A][= E] \mid \textbf{include}\ S & \text{theory bodies} \\
d & ::= c[: A] = E \mid \textbf{include}\ M & \text{morphism bodies} \\
S & ::= s \mid \ \ldots & \text{theory expressions} \\
M & ::= m \mid id_S \mid M; M \mid \ \ldots & \text{morphism expressions (paths)}
\end{array}
$$

MMT-Grammar

$$
\begin{array}{rll}
G ::= & \emptyset \mid G,\ \textbf{theory}\ s = \{D^*\} & \text{nodes} \\
& \mid G,\ \textbf{morph}\ m : S \longrightarrow S = \{d^*\} & \text{edges} \\
& \mid G,\ \textbf{morpheq}\ k : M \overset{mor}{=} M : S \longrightarrow S = \{\delta^*\} & \text{morph. equalities} \\
D ::= & c[: A][= E] \mid \textbf{include}\ S & \text{theory bodies} \\
d ::= & c[: A] = E \mid \textbf{include}\ M & \text{morphism bodies} \\
\delta ::= & c[: A] = E \mid \textbf{include}\ K & \text{morph. eq. bodies} \\
S ::= & s \mid \ldots & \text{theory expressions} \\
M ::= & m \mid id_S \mid M; M \mid \ldots & \text{morphism expressions (paths)} \\
K ::= & k \mid \textbf{refl}\ M \mid \ldots & \text{morph. eq. proof terms}
\end{array}
$$

Morphism Equality for Group and DivGroup

$$\begin{array}{l} \textbf{morph}\ \mathtt{DGtoG} : \mathtt{DivGroup} \longrightarrow \mathtt{Group} = \\ \quad \textbf{include}\ \mathit{id}_{\mathtt{Carrier}} \\ \quad / \quad = \quad \lambda x, y.\, x \circ y^{-1} \\ \quad e \quad = \quad e \end{array}$$

$$\begin{array}{l} \textbf{morph}\ \mathtt{GtoDG} : \mathtt{Group} \longrightarrow \mathtt{DivGroup} = \\ \quad \textbf{include}\ \mathit{id}_{\mathtt{Carrier}} \\ \quad \circ \quad = \quad \lambda x, y.\, x/(e/y) \\ \quad e \quad = \quad e \\ \quad ^{-1} \quad = \quad \lambda x.\, e/x \end{array}$$

$$\begin{array}{l} \textbf{morpheq}\ k : \mathtt{DGtoG}; \mathtt{GtoDG} \overset{mor}{=} \mathit{id}_{\mathtt{DivGroup}} : \mathtt{DivGroup} \longrightarrow \mathtt{Group} = \\ \quad \textbf{include}\ \mathtt{refl}\ \mathit{id}_{\mathtt{Carrier}} \\ \quad / \quad = \quad \text{proof of } (\mathtt{DGtoG}; \mathtt{GtoDG})(/) = / \\ \quad e \quad = \quad \text{proof of } e = e \end{array}$$

(omitted: axioms mapped to proof equality, trivial if proof-irrelevant)

Morphism Equality for Group and DivGroup

$$\begin{array}{l} \textbf{morph}\ \mathtt{DGtoG} : \mathtt{DivGroup} \longrightarrow \mathtt{Group} = \\ \quad \textbf{include}\ id_{\mathtt{Carrier}} \\ \quad / \quad = \quad \lambda x, y.\, x \circ y^{-1} \\ \quad e \quad = \quad e \end{array}$$

$$\begin{array}{l} \textbf{morph}\ \mathtt{GtoDG} : \mathtt{Group} \longrightarrow \mathtt{DivGroup} = \\ \quad \textbf{include}\ id_{\mathtt{Carrier}} \\ \quad \circ \quad = \quad \lambda x, y.\, x/(e/y) \\ \quad e \quad = \quad e \\ \quad ^{-1} \quad = \quad \lambda x.\, e/x \end{array}$$

$$\begin{array}{l} \textbf{morpheq}\ k : \mathtt{DGtoG}; \mathtt{GtoDG} \stackrel{mor}{=} id_{\mathtt{DivGroup}} : \mathtt{DivGroup} \longrightarrow \mathtt{Group} = \\ \quad \textbf{include}\ \mathtt{refl}\ id_{\mathtt{Carrier}} \\ \quad / \quad = \quad \text{proof of } (\mathtt{DGtoG}; \mathtt{GtoDG})(/) = / \\ \quad e \quad = \quad \text{proof of } e = e \end{array}$$

(omitted: axioms mapped to proof equality, trivial if proof-irrelevant)

$(\mathtt{DGtoG}; \mathtt{GtoDG})(/) = \lambda x, y.\, x/(y^{-1})^{-1}$ ... is that even equal to $/$?

Morphism Equality Depends on Object Equality

Works for every *individual* logic

- $\lambda x, y.\, x/(y^{-1})^{-1} \;=\; /$ *is* a theorem of FOL
- so we're good if our formalization hard-codes FOL-equality

Problem for logical *frameworks*

- FOL defined in logical framework LF
- LF-equality is just $\alpha\beta\eta$
- FOL-equality is just some connective with rules defined in LF
- $\lambda x, y.\, x/(y^{-1})^{-1} \;=\; /$ *not* true in LF

Challenge

- no uniform LF-level definition of morphism equality

  contrary to everything else about theories/morphism
- even worse: different applications may prefer different object equality variants

  morphism equality depends on the context

Modular Choice of Equality

Bad solution

- when proving morph. eq., state which equality to use
- Problem: now multiple different categories with different morph. eq. very confusing to use

Our solution

- only LF-equality matters, only one category of theories
- object logics may register additional LF-equalities
- morphism equality determined by codomain

Modular Choice of Equality: Example

A mixin for FOL that quotients out provable equality:

$$\begin{array}{l} \textbf{theory}\ \mathtt{FOLQ} = \\ \quad \textbf{include}\ \mathtt{FOL} \\ \quad \mathtt{folTermEq} \quad : \quad \Pi a, x, y : \mathtt{tm}\, a.\ x \overset{FOL}{=} y \ \to\ x \overset{LF}{=} y \\ \quad \mathtt{folFormEq} \quad : \quad \Pi x, y : \mathtt{prop}.\ x \Leftrightarrow y \ \to\ x \overset{LF}{=} y \end{array}$$

not needed for FOL theories or morphisms

Mix in to the codomain to choose equality:

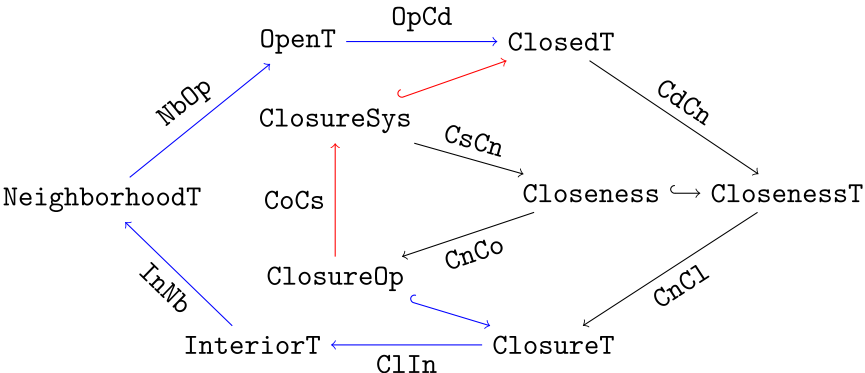
$$\begin{array}{l} \textbf{morpheq}\ k : \mathtt{DGtoG}; \mathtt{GtoDG} \overset{mor}{=} id_{\mathtt{DivGroup}} : \mathtt{DivGroup} \longrightarrow \mathtt{Group} \cup \mathtt{FOLQ} \\ \quad \textbf{include}\ \mathtt{refl}\ id_{\mathtt{Carrier}} \\ \quad / \quad = \quad \mathtt{folTermEq}\ \big(\text{FOL-proof of } (\mathtt{DGtoG}; \mathtt{GtoDG})(/) = /\big) \\ \quad e \quad = \quad \mathtt{refl}\, e \end{array}$$

Case Studies

Realm of Topological Spaces

Formalization in MMT

- logic: first-order logic + power types (to talk about sets of points)
- 3 isomorphic theories for closure systems
- 6 isomorphic theories for topological space
- morphism equality proofs for every point in the two loops



Open Problem: Diagram Chase

Equality of morphism *expressions*

- given: some equalities $M_i \overset{mor}{=} N_i : S_i \to T_i$
- determine whether they imply $M \overset{mor}{=} N$
- if true, critical to check it instantaneously
- Example: red $\overset{mor}{=}$ blue in the topological space example

undecidable in theory graphs with loops, but heuristics usually work

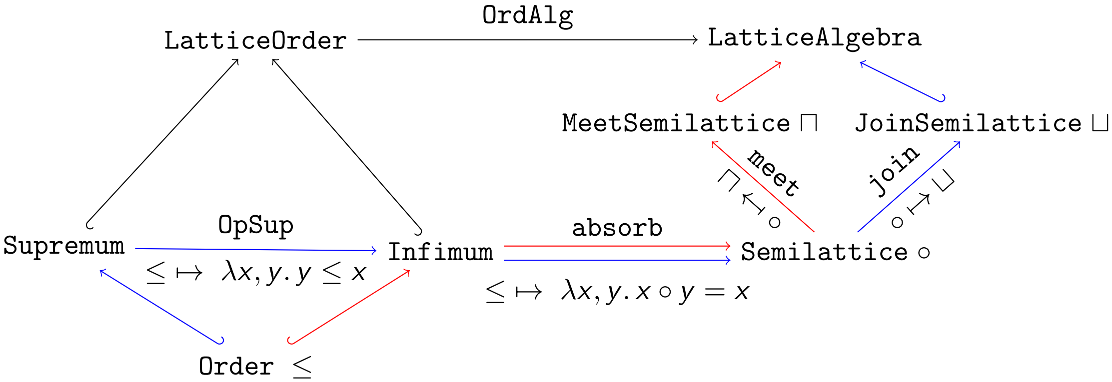
Our Plan

- MMT implements some heuristic for morphism equality inference
- users can supply explicit morph. eq. proofs where the heuristic is incomplete

The Realm of Lattices

Formalization in MMT

- 2 isomorphic first-order logic theories: based on algebra vs. based on order
- 15-year-old issue: `OrdAlg` : `LatticeOrder` $\to$ `LatticeAlgebra` defines $x \leq y$ twice
  $x \sqcap y = x$ vs. $x \sqcup y = y$ type-checks only after proving *red* $\overset{mor}{=}$ *blue*



Conclusion

Morphism Equality in Theory Graphs

Summary

- MMT toplevel = theories + morphisms + morphism equalities
- two kinds of morph. eq. proofs
  - atomic case: one equality proof for every domain constant
    essentially functional extensionality
  - complex case: proof terms for implied morphism equalities
    much more efficient — no inspection of theory/morphism bodies
    but undecidable
- lightweight patterns for realms (= sets of isomorphic theories)

Future Work

- robust implementation in MMT system
- better reasoning for implied equalities
- investigate scalability