# Computer-supported Exploration of a Categorical Axiomatization of Modeloids

Lucca Tiemens[1,3] Dana Scott[2] Christoph Benzmüller[3,4]
Miroslav Benda[5]

[1]Technische Universität Berlin, Berlin, Germany
[2]University of California, Berkeley, CA, USA
[3]Freie Universität Berlin, Berlin, Germany
[4]University of Luxembourg, Esch-sur-Alzette, Luxembourg
[5]Formerly at University of Washington, Seattle, USA

October 27, 2020

# Outline

1. From Modeloids to Categorical Modeloids
2. Categorical Derivative and Ehrenfeucht-Fraïssé games
3. Automated Theorem Proving in Context

# Motivation

- Paper “Modeloids. I” by Benda from 1979
- Dana Scott’s suggestion to formulate a modeloid in the language of category theory after a conversation in early 2019

Questions:

1. Can the properties of a modeloid be stated in categorical language?
2. Can computer-supported theorem proving help in the research process?

# Notions

- Vocabulary

  Let **S** be a relational structure. That is

  $$\mathbf{S} = (S, R_1, R_2, .., R_n)$$

  for some set $S$ and $n \in \mathbb{N}$, where $R_1, .., R_n$ are relational symbols. We call the non-logical symbols $R_1, ..., R_n$ the vocabulary of **G**.

- Partial automorphism

  A partial automorphism of a structure $M$ is an isomorphism between two substructures of $M$.

  Example: Let $G$ be a group having two isomorphic subgroups. This isomorphism is then a partial automorphism of $G$.

# A Modeloid

Idea: Study the abstract properties of the set of partial automorphisms of a relational structure $S$ instead of $S$ itself.

**Definition (Modeloid)**

A modeloid is a set $M$ of partial bijections on a finite set $\Sigma$ that satisfies

1. Closure of composition: $f, g \in M \Rightarrow f \circ g \in M$
2. Closure of taking inverses: $f \in M \Rightarrow f^{-1} \in M$
3. Inclusion property: $f \in M$ and $A \subset dom(f)$ implies $f|_A \in M$
4. Identity: $id_\Sigma \in M$

# Roadmap to Category Theory

Modeloid is an inverse monoid

$\Downarrow$

Wagner-Preston representation theorem
enables a faithful translation to semigroup language
(Semimodeloid)

$\Downarrow$

A semimodeloid is an one-object inverse category

$\Downarrow$

Categorical modeloid

# Wagner-Preston representation theorem

## Definition (Inverse semigroup)

$(\Sigma, ^{-1}, *)$ is called an inverse semigroup if for all $x, y, z \in \Sigma$

1. $(x * y) * z = x * (y * z)$
2. $x * x^{-1} * x = x$
3. $(x^{-1})^{-1} = x$
4. $x * x^{-1} * y * y^{-1} = y * y^{-1} * x * x^{-1}$

## Theorem (Wagner-Preston)

*Let* $\Sigma = (\Sigma, ^{-1}, *)$ *be an inverse semigroup. Then there is an injective homomorphism* $\Omega : \Sigma \rightarrow \{f : \Sigma \rightarrow \Sigma \mid f \text{ is a partial bijection}\}$, *such that for* $a, b \in \Sigma$ *we have* $a \leq b \iff \Omega(a) \subseteq \Omega(b)$.

# Abstraction to Inverse Semigroups

Idea: What is a function in a *modeloid*, is now to become an element in an *inverse monoid*. We call this construction a semimodeloid.

**Definition (Semimodeloid)**

Let $S^1 = (\Sigma, {}^{-1}, *, e, 0)$ be an inverse monoid. Then a semimodeloid $M \subseteq \Sigma$ has to satisfy

1. $\forall x, y \in M : (x * y) \in M$
2. $\forall x \in M : x^{-1} \in M$
3. $\forall x \in M \, \forall y \in S^1 : y \leq x \Rightarrow y \in M$
4. $e \in M$

Note: $\leq$ is the natural partial order resembling function restriction.

# Towards Category Theory

Reminder: Inverse monoid is one-object inverse category.

## Definition (Categorical modeloid)

Let **C** be a small inverse category **C** with all zero elements. Then a *categorical modeloid* $M$ on **C** is such that $M \subseteq C$ satisfies the following axioms

1. $a, b \in M \Rightarrow a \cdot b \in M$
2. $a \in M \Rightarrow a^{-1} \in M$
3. $\forall\, a \in C\, \forall b \in M : a \leq b \Rightarrow a \in M$
4. $\forall$ objects $X \in C : X \in M$

# Outline

# The Derivative on a Modeloid

Idea: An operation that checks if a partial bijection is arbitrarily extendable by one element.

**Definition (Derivative)**

Let $M$ be a modeloid on $\Sigma$. Then the derivative $D(M) \subseteq F(\Sigma)$ is defined by

$$\{(x_1, y_1), .., (x_n, y_n)\} \in D(M) :\Leftrightarrow$$
$$\forall a \in \Sigma \, \exists b \in \Sigma : \{(x_1, y_1), .., (x_n, y_n), (a, b)\} \in M \wedge$$
$$\forall a \in \Sigma \, \exists b \in \Sigma : \{(x_1, y_1), .., (x_n, y_n), (b, a)\} \in M$$

# Categorical Derivative

Difference: Checks for general extendability, not just for one element.

## Definition (Derivative on Homset)

Let $M$ be a categorical modeloid. We define the derivative on $Hom_M(X, Y)$ for $X, Y \in M$ as

$$D(Hom_M(X,Y)) := \{f \in Hom_M(X,Y) \mid \forall \text{ idempotent atomic } a \in End_M(X)\, \exists h \in Hom_M(X,Y) : (f \leq h \wedge a \leq h^{-1}h) \wedge \forall \text{ idempotent atomic } b \in End_M(Y)\, \exists g \in Hom_M(X,Y) : (f \leq g \wedge b \leq gg^{-1})\}$$

# Derivative and Ehrenfeucht-Fraïssé games

Background: Two structures $X$ and $Y$ are called m-equivalent, denoted by $X \equiv_m Y$, if they satisfy the same sentences of *quantifier rank* $\leq m$ for some $m \in \mathbb{N}$.

Example: $\forall x \exists y.\, R(x, y)$ is a sentence of *quantifier rank* 2.

**Theorem**

*Let $X$ and $Y$ be finite relational structures with the same vocabulary and let $M$ be the categorical modeloid on $X$ and $Y$. Then*

$$\exists f : X \to Y \in D^m(M) \iff X \equiv_m Y, \quad m \in \mathbb{N}$$

# Outline

1. From Modeloids to Categorical Modeloids
2. Categorical Derivative and Ehrenfeucht-Fraïssé games
3. Automated Theorem Proving in Context

# Proofs in Inverse Semigroup Theory

```
lemma idem_comp: "x ∈ carrier S ⟶ Idemp (x ⊗ invs x)"
  by (metis inv_closed mult_assoc mult_closed regular)

lemma idem_comp_rev: "x ∈ carrier S ⟶ Idemp (invs x ⊗ x)"
  using inv_closed mult_assoc mult_closed regular by auto

lemma regular_inv: "x ∈ carrier S ⟶ invs x ⊗ x ⊗ invs x = invs x"
  using inv_closed invinv regular by fastforce

lemma idem_inv: "x ∈ carrier S ∧ Idemp x ⟶ invs x = x"
  by (smt idem_comm inv_closed invinv mult_assoc mult_closed regular)

lemma idem_comp_comm: "x ∈ carrier S ∧ y ∈ carrier S ⟶ Idemp x ∧ Idemp y ⟶ x ⊗ y = y ⊗ x"
  by (smt idem_comm inv_closed invinv mult_assoc mult_closed regular)

lemma idem_comp_idem: "e ∈ carrier S ∧ f ∈ carrier S ∧ Idemp e ∧ Idemp f ⟶ Idemp (e ⊗ f)"
  by (smt idem_comm inv_closed invinv mult_assoc mult_closed regular)

lemma inverse_unique: "x ∈ carrier S ∧ y ∈ carrier S ∧ z ∈ carrier S ∧ y inverseOf x ∧ z inverseOf x ⟶ y = z"
  by (smt idem_comm inv_closed invinv mult_assoc mult_closed regular)

theorem inverse_unique_quant: "∀x∈carrier S. ∃!y∈carrier S. y inverseOf x"
  by (smt idem_comm inv_closed invinv mult_assoc mult_closed regular)

proposition inv_hom: "x ∈ carrier S ∧ y ∈ carrier S ⟶ invs (x⊗y) = invs y ⊗ invs x"
  sledgehammer(mult_closed mult_assoc inv_closed regular invinv idem_comm)
  by (smt idem_comm inv_closed invinv mult_assoc mult_closed regular)
```

☑ Proof state ☑ Auto update Update Search:

```
Sledgehammering...
Proof found...
"z3": Try this: by (smt idem_comm inv_closed invinv mult_assoc mult_closed regular) (> 1.0 s, timed out)
```

# Inverse Category in Isabelle/HOL

```
theory InverseCategory imports Main FreeLogic

begin

typedecl α —‹This type can be thought of to represent the morphisms of a category.›

locale category =
        —‹We need three functions to define a category.›
  fixes domain:: "α ⇒ α" ("dom _" [108] 109) and
        codomain:: "α ⇒ α" ("cod _" [110] 111) and
        composition:: "α ⇒ α ⇒ α" (infix "·" 110) and
        star::α ("*") —‹Symbol for non-existing elements›

assumes
—‹Here we define the axioms that the morphisms in interaction with the functions have to obey.›
        S1: "E(dom x) → E x" and
        S2: "E(cod y) → E y" and
        S3: "E(x·y) ↔ dom x ≃ cod y" and
        S4: "x·(y·z) ≅ (x·y)·z" and
        S5: "x·(dom x) ≅ x" and
        S6: "(cod y)·y ≅ y" and

—‹We add one more axiom that is not present in the original formulation. We want all non-existing morphisms to be equal.›
        L1: "¬(E m) → (m = *)"

locale inverseCategoryQantFree = category +
  fixes inverse::"α ⇒ α" ("inv _")
  assumes C1: "x·((inv x)·x) ≅ x" and
          C2: "((inv (inv x)) ≅ x)" and
          C3: "((x·inv x) · (y·(inv y))) ≅ ((y·(inv y)) · (x·(inv x)))"
begin
lemma "True" nitpick[satisfy] oops

—‹We show the strictness of @{text "E"} for the inverse›
lemma "(E x) ↔ (E (inv x))"
  by (metis C1 C2 S1 S2 S3)
```

# Current and future work

- Implementation of this project in Isabelle/HOL.
- Further comparison between categorical modeloids and modeloids

Thank you very much for your attention!