Formalising the initiality conjecture in Coq

jww Anders Mörtberg

Peter LeFanu Lumsdaine

Stockholms universitet

Göteborg–Stockholm Joint Type Theory Seminar
Chalmers / Göteborg University, 12 December 2018

Formalising the initiality conjecture in Coq and Agda

jww Menno de Boer, Guillaume Brunerie, Anders Mörtberg

Peter LeFanu Lumsdaine   Guillaume Brunerie

Stockholms universitet

Göteborg–Stockholm Joint Type Theory Seminar
Chalmers / Göteborg University, 12 December 2018

Initiality for dependent type theories

By initiality for a type theory, mean a statement like:

Template

The syntactic category of dependent type theory with Σ, Π, and Id-types forms the initial contextual category with Σ, Π, and Id-structure.

Initiality for dependent type theories

By initiality for a type theory, mean a statement like:

Template

The syntactic category of dependent type theory with Σ, Π, and Id-types forms the initial contextual category with Σ, Π, and Id-structure.

- Justifies categorical-algebraic definition of "models of DTT with XYZ-types" as "contextual cats with XYZ-structure"
- Packages the bureaucracy of interpreting syntax into such structures
- Should hold uniformly for all dependent type theories
- Variations: could state with CwA's, CwF's, C-systems, etc.; with various different presentations of the type theory; with 2-categorical initiality; …

Status

- **Thesis: Initiality is established**
  - Proven by Streicher (1991 book) for Calculus of Constructions; by Hofmann (1995 thesis) for DTT with Π, Id, $\mathbb{N}$.
  - Proof extends straightforwardly + robustly to other type theories.
  - (NB: Other presentations in literature (that I'm aware of) use techniques specific to their particular type theories, don't extend so robustly; or handwave many details.)

Status

- **Thesis: Initiality is established**
  - Proven by Streicher (1991 book) for Calculus of Constructions; by Hofmann (1995 thesis) for DTT with Π, Id, $\mathbb{N}$.
  - Proof extends straightforwardly + robustly to other type theories.
  - (NB: Other presentations in literature (that I'm aware of) use techniques specific to their particular type theories, don't extend so robustly; or handwave many details.)
- **Antithesis: Initiality is an open problem**
  - Extension not really straightforward at all!
  - What type theories is it even supposed to hold for? It fails for some!

Status

- **Thesis: Initiality is established**
  - Proven by Streicher (1991 book) for Calculus of Constructions; by Hofmann (1995 thesis) for DTT with Π, Id, $\mathbb{N}$.
  - Proof extends straightforwardly + robustly to other type theories.
  - (NB: Other presentations in literature (that I'm aware of) use techniques specific to their particular type theories, don't extend so robustly; or handwave many details.)
- **Antithesis: Initiality is an open problem**
  - Extension not really straightforward at all!
  - What type theories is it even supposed to hold for? It fails for some!
- **Synthesis: Initiality is heuristically well-understood**
  - "Experts" do understand what kinds of type theories it holds for, and how to extend Streicher's proof.
  - But: this understanding not clearly articulated anywhere, rigorously or even heuristically.
  - Extension of proof mostly straightforward — minor tweaks needed, no substantial new ideas — but carefully making sure of this involves checking a lot of details.

Solution proposals

Long-term solution

Define general class of dependent type theories; state and prove initiality for these.

Solution proposals

Long-term solution

Define general class of dependent type theories; state and prove initiality for these.

- Precisely: define rigorously a general class of dependent type theories…
- …yielding, as instances, as many of the specific theories of interest as possible…
- …modulo minor differences in presentation, as minor as possible.
- Define corresponding categorical-algebraic structures for these…
- …yielding the established definitions, as closely as possible…
- …and prove initiality with respect to these.

Solution proposals

Long-term solution

Define general class of dependent type theories; state and prove initiality for these.

- Precisely: define rigorously a general class of dependent type theories…
- …yielding, as instances, as many of the specific theories of interest as possible…
- …modulo minor differences in presentation, as minor as possible.
- Define corresponding categorical-algebraic structures for these…
- …yielding the established definitions, as closely as possible…
- …and prove initiality with respect to these.

Proposals for this (in progress): Bauer–Lumsdaine–Haselwarter; Brunerie. Related proposals: Isaev, Capriotti.

Solution proposals

Long-term solution

Define general class of dependent type theories; state and prove initiality for these.

- Precisely: define rigorously a general class of dependent type theories…
- …yielding, as instances, as many of the specific theories of interest as possible…
- …modulo minor differences in presentation, as minor as possible.
- Define corresponding categorical-algebraic structures for these…
- …yielding the established definitions, as closely as possible…
- …and prove initiality with respect to these.

Proposals for this (in progress): Bauer–Lumsdaine–Haselwarter; Brunerie. Related proposals: Isaev, Capriotti.

Short-term solution

Just damn well prove it for some more of the specific type theories of interest!

Proofs of initiality are like buses

Goal

Prove initiality carefully, for some specific type theory, ideally approaching “book HoTT”.

Proofs of initiality are like buses

Goal

Prove initiality carefully, for some specific type theory, ideally approaching "book HoTT".

- Major recent assault on this: Initiality Project.

  Collaborative many-author project; led by Shulman; aiming for book HoTT; written, not formalised.

Proofs of initiality are like buses

Goal

Prove initiality carefully, for some specific type theory, ideally approaching "book HoTT".

- Major recent assault on this: Initiality Project.

  Collaborative many-author project; led by Shulman; aiming for book HoTT; written, not formalised.

- Second (and third) assault: from Stockholm, formalised. Two parallel developments, started together: Brunerie–de Boer in Agda, Lumsdaine–Mörtberg in Coq.

  Small type theory at first: Π-types, a dependent family of base types.

  Key design criterion: robust extensibility. Avoid doing anything that wouldn't extend to arbitrary constructors/rules.

  "We can have this done within a week." — PLL, 19 October 2018

Proofs of initiality are like buses

Goal

Prove initiality carefully, for some specific type theory, ideally approaching "book HoTT".

- Major recent assault on this: Initiality Project.

  Collaborative many-author project; led by Shulman; aiming for book HoTT; written, not formalised.

- Second (and third) assault: from Stockholm, formalised. Two parallel developments, started together: Brunerie–de Boer in Agda, Lumsdaine–Mörtberg in Coq.

  Small type theory at first: Π-types, a dependent family of base types.

  Key design criterion: robust extensibility. Avoid doing anything that wouldn't extend to arbitrary constructors/rules.

  "We can have this done within a week." — PLL, 19 October 2018

  Developments begun 22 October; Brunerie-de Boer initiality attained 19 November!

Over to Guillaume!

[Here PLL hands over to Guillaume, who presents the Brunerie–de Boer Agda formalisation before handing back to PLL with the rest of these slides.]

Lumsdaine–Mörtberg formalisation details

Background

Approach based in part on previous attempt by PLL from 2014–15 (in joint development Gylterud–Lumsdaine–Palmgren).

Attempt foundered due to combination of several design choices making life hard:

- use of named variables
- use of setoids for the target models
- started with slightly overambitious type theory
- …

Very useful experience to build on — both the good and the bad aspects…

Design choices

This time round:

- Proof assistant: Coq; specifically, over UniMath. (Mainly: for a well-developed category theory library that both authors were familiar with.)
- Models: Categories with attributes, not assuming objects form a set (so, CwA a 2-category); for 1-categorical initiality, contextual categories as CwA's plus contextuality axiom (implying objects a set).
- Variables in raw syntax: using de Bruijn indices. Raw syntax: well-scoped. These enable:
- All inductions purely structural, over either raw syntax or derivations. No size measures, auxiliary well-founded relations, etc.
- Context and context-equality judgements subsidiary, not primitive, don't appear in derivations. Substitution admissible, not a primitive rule. These enable:
- Interpretation fuction (partial + totality): into arbitrary CwA's. No use of equality on objects/contexts needed.

Experience

Good news

1. Interpretation function (partial + total): went very smoothly.

Experience

Good news

1. Interpretation function (partial + total): went very smoothly.
2. Admissibility of substitution, etc: went surprisingly smoothly.

Experience

Good news

1. Interpretation function (partial + total): went very smoothly.
2. Admissibility of substitution, etc: went surprisingly smoothly.
3. Categorical operations on syntax: clean to define, derive properties as judgemental equalities, etc.

Experience

Good news

1. Interpretation function (partial + total): went very smoothly.
2. Admissibility of substitution, etc: went surprisingly smoothly.
3. Categorical operations on syntax: clean to define, derive properties as judgemental equalities, etc.

Bad news

1. Quotients

Experience

Good news

1. Interpretation function (partial + total): went very smoothly.
2. Admissibility of substitution, etc: went surprisingly smoothly.
3. Categorical operations on syntax: clean to define, derive properties as judgemental equalities, etc.

Bad news

1. Quotients
2. Quotients

Experience

Good news

1. Interpretation function (partial + total): went very smoothly.
2. Admissibility of substitution, etc: went surprisingly smoothly.
3. Categorical operations on syntax: clean to define, derive properties as judgemental equalities, etc.

Bad news

1. Quotients
2. Quotients
3. Quotients

Experience

Good news

1. Interpretation function (partial + total): went very smoothly.
2. Admissibility of substitution, etc: went surprisingly smoothly.
3. Categorical operations on syntax: clean to define, derive properties as judgemental equalities, etc.

Bad news

Specifically, interaction of 2 issues:

1. Syntactic CwA a dependently-typed structure (maps depend on objects), where the objects are quotiented (contexts, up to judgemental equality)
2. In UniMath's quotients, the dependent eliminator doesn't compute judgementally.

Together: ends up about as painful as using setoids.

Status

1. "Pre-quotient" parts: almost all done, remaining part expected quick. Contains all the mathematically interesting bits.
    - Done: definition of type theory, and suitable structured CwA's; admissibility of substitution, presuppositions; structural operations of the syntactic CwA, and interpretation function into CwA's/contextual categories, i.e. pre-quotient aspects of existence part of initiality.
    - Remaining to do: functoriality of interpretation under CwA maps, amounting to uniqueness part of initiality.

    Around 4,000 lines of code (not including libraries).
2. "Post-quotient" parts, i.e. assembling the pre-quotient parts into the syntactic CwA and functions thereon: some parts done, much remaining. Mathematically fairly uninteresting, but diffcult and slow.
    - Done: syntactic category; part of CwA structure thereon; most of underlying functor of the interpretation map.
    - Remaining to do: rest of CwA structure, and logical structure thereon; interpretation as a map of CwA's with structure; uniqueness of the interpretation map.

    Arround 1,000 lines of code, so far!

Dinner: Restaurant Zozaki, Stora Nygatan 3, 18:00!