Canonicity for Indexed Inductive-Recursive Types

András Kovács

University of Gothenburg & Chalmers University of Technology

1st Oct 2025, Workshop in Honour of Peter Dybjer, Gothenburg

Inductive-Recursive Types

Mutual definition of an inductive type and a function acting on it.

$$
\begin{array}{l}
\textbf{mutual} \\
\quad \textbf{data}\ \mathsf{Code} : \mathsf{Set}_0\ \textbf{where} \\
\quad\quad \mathsf{Nat}' : \mathsf{Code} \\
\quad\quad \Pi' \quad : (A : \mathsf{Code}) \to (\mathsf{El}\, A \to \mathsf{Code}) \to \mathsf{Code} \\
\\
\quad \mathsf{El} : \mathsf{Code} \to \mathsf{Set}_0 \\
\quad \mathsf{El}\, \mathsf{Nat}' \quad\quad = \mathsf{Nat} \\
\quad \mathsf{El}\, (\Pi'\, A\, B) = (a : \mathsf{El}\, A) \to \mathsf{El}\, (B\, a)
\end{array}
$$

Early and informal use by Per Martin-Löf [ML75, ML84].

Formal syntax & semantics developed by Peter and Anton [Dyb00, DS99, DS03, DS06].

Inductive-Recursive Types

Use-cases:

- Metatheory for TTs with various universe hierarchies:
  - Normalization for TTs with countable universes [ML75, AÖV18, PT23, ADE23].
  - Consistency [Kov22] and canonicity [CW25] for notions of first-class universe levels.
- Others: partial functions [BC01], generic programming [BDJ03, Die17], large countable ordinals [Kov23, Kyu25].

Inductive-Recursive Types

Use-cases:

- Metatheory for TTs with various universe hierarchies:
  - Normalization for TTs with countable universes [ML75, AÖV18, PT23, ADE23].
  - Consistency [Kov22] and canonicity [CW25] for notions of first-class universe levels.
- Others: partial functions [BC01], generic programming [BDJ03, Die17], large countable ordinals [Kov23, Kyu25].

Canonicity: can we evaluate every closed term in MLTT+IR to a value?

IR has been supported for a long time in Agda with no canonicity issues, but without formal proof.

Inductive-Recursive Types

Use-cases:

- Metatheory for TTs with various universe hierarchies:
  - Normalization for TTs with countable universes [ML75, AÖV18, PT23, ADE23].
  - Consistency [Kov22] and canonicity [CW25] for notions of first-class universe levels.
- Others: partial functions [BC01], generic programming [BDJ03, Die17], large countable ordinals [Kov23, Kyu25].

Canonicity: can we evaluate every closed term in MLTT+IR to a value?

IR has been supported for a long time in Agda with no canonicity issues, but without formal proof.

In this talk & upcoming paper: formal proof of canonicity.

How to specify an IR type?

The type of **IR signatures** is an inductive type.

$$
\begin{aligned}
&\textbf{data}\,\mathsf{Sig}\;i\;\{j\}\;(O : \mathsf{Set}_j) : \mathsf{Set}_{(i+1\,\sqcup\,j)}\;\textbf{where}\\
&\quad \iota\;\, : O \to \mathsf{Sig}\;i\;O\\
&\quad \sigma : (A : \mathsf{Set}_i) \to (A \to \mathsf{Sig}\;i\;O) \to \mathsf{Sig}\;i\;O\\
&\quad \delta\; : (A : \mathsf{Set}_i) \to ((A \to O) \to \mathsf{Sig}\;i\;O) \to \mathsf{Sig}\;i\;O
\end{aligned}
$$

The signature of the previous Code example:

$$
\begin{aligned}
&\mathsf{SigCode} : \mathsf{Sig}\;0\;\mathsf{Set}_0\\
&\mathsf{SigCode} :\equiv \sigma\;\mathsf{Bool}\;\lambda\,t.\,\textbf{case}\;t\;\textbf{of}\\
&\quad \mathsf{true}\;\to \iota\,\mathsf{Nat}\\
&\quad \mathsf{false} \to \delta\,\top\,\lambda\,ElA.\,\delta\,(ElA\,\mathsf{tt})\,\lambda\,ElB.\,\iota\,((x : ElA\,\mathsf{tt}) \to ElB\,x)
\end{aligned}
$$

For each signature, we postulate type formation, term formation, elimination and computation rules, for the described IR type.

How to prove canonicity?

Every closed term with IR type should be convertible to an IR constructor. E.g. a closed term t : Code should be either Nat' or Π'.

Type-theoretic Artin gluing [Coq19, KHS19]:

- A closed type is interpreted as a proof-relevant predicate over its closed terms. The predicate should imply canonicity.
- A closed term is interpreted as a predicate witness.
- Open types & terms are generalized over closed substitutions.

How to prove canonicity?

Notation: we write **Ty** for the set of closed types, **Tm *A*** for sets of closed terms, and **U : Ty** for object-theoretic universes (omitting levels).

Example: canonicity predicate for natural numbers.

$$\begin{aligned}
&\textbf{data}\ \mathrm{Nat}^\circ : \mathrm{Tm}\,\mathrm{Nat} \to \mathrm{Set}\ \textbf{where} \\
&\quad \mathrm{zero}^\circ : \mathrm{Nat}^\circ\,\mathrm{zero} \\
&\quad \mathrm{suc}^\circ\ \ : (n : \mathrm{Tm}\,\mathrm{Nat}) \to \mathrm{Nat}^\circ\,n \to \mathrm{Nat}^\circ\,(\mathrm{suc}\,n)
\end{aligned}$$

Generally: the canonicity predicate for an inductive type is a metatheoretic indexed inductive type.

How to prove canonicity?

Example: the canonicity predicate for Code is the following metatheoretic indexed IR type:

$$
\begin{aligned}
&\textbf{data}\,\mathsf{Code}^\circ : \mathsf{Tm}\,\mathsf{Code} \to \mathsf{Set}\,\textbf{where}\\
&\quad \mathsf{Nat}'^\circ : \mathsf{Code}^\circ\,\mathsf{Nat}'\\
&\quad \Pi'^\circ \quad : \{A : \mathsf{Tm}\,\mathsf{Code}\}(A^\circ : \mathsf{Code}^\circ\,A)\\
&\qquad\qquad \{B : \mathsf{Tm}\,(\mathsf{El}\,A \to \mathsf{Code})\}(B^\circ : \{a : \mathsf{Tm}\,(\mathsf{El}\,a)\} \to \mathsf{El}^\circ\,A^\circ\,a \to \mathsf{Code}^\circ\,(B\,a))\\
&\qquad\qquad \to \mathsf{Code}^\circ\,(\Pi'\,A\,B)
\end{aligned}
$$

$$
\begin{aligned}
&\mathsf{El}^\circ : \{t : \mathsf{Tm}\,\mathsf{Code}\} \to \mathsf{Code}^\circ\,t \to (\mathsf{Tm}\,(\mathsf{El}\,t) \to \mathsf{Set})\\
&\mathsf{El}^\circ\,\mathsf{Nat}'^\circ \qquad\quad = \mathsf{Nat}^\circ\\
&\mathsf{El}^\circ\,(\Pi'^\circ\,A^\circ\,B^\circ) = \lambda\,f.\,\{a : \mathsf{Tm}\,A\} \to \mathsf{El}^\circ\,A^\circ\,a \to \mathsf{El}^\circ\,B^\circ\,(f\,a)
\end{aligned}
$$

If I have $t : \mathsf{Tm}\,\mathsf{Code}$, the gluing interpretation hands me an element of $\mathsf{Code}^\circ\,t$.

How to prove canonicity?

In the object theory, let's assume general IR types specified by an internal Sig type.

- Previous slide: the object theory supports Code, the canonicity proof involves Code°.

How to prove canonicity?

In the object theory, let's assume general IR types specified by an internal Sig type.

- Previous slide: the object theory supports Code, the canonicity proof involves Code°.
- Now: the object theory has all IR types, the canonicity proof involves all canonicity predicates.

How to prove canonicity?

In the object theory, let's assume general IR types specified by an internal Sig type.

- Previous slide: the object theory supports Code, the canonicity proof involves Code°.
- Now: the object theory has all IR types, the canonicity proof involves all canonicity predicates.

Interpreting IR types in the glued model:

- We write $\mathsf{IR}\,S : \mathsf{Tm}\,\mathsf{U}$ for the IR *type formation* rule, for $S : \mathsf{Tm}\,(\mathsf{Sig}\,O)$.

How to prove canonicity?

In the object theory, let's assume general IR types specified by an internal Sig type.

- Previous slide: the object theory supports Code, the canonicity proof involves Code°.
- Now: the object theory has all IR types, the canonicity proof involves all canonicity predicates.

Interpreting IR types in the glued model:

- We write $\mathsf{IR}\,S : \mathsf{Tm}\,\mathsf{U}$ for the IR *type formation* rule, for $S : \mathsf{Tm}\,(\mathsf{Sig}\,O)$.
- To interpret $\mathsf{IR}\,S$:
  - By induction hypothesis, we get $S^\circ : \mathsf{Sig}^\circ\,S$ witnessing the canonicity of the signature S itself.
  - We compute a metatheoretic indexed IR signature by induction on S°. This yields the canonicity predicate for $\mathsf{IR}\,S$.

How to prove canonicity?

In the object theory, let's assume general IR types specified by an internal Sig type.

- Previous slide: the object theory supports Code, the canonicity proof involves Code°.
- Now: the object theory has all IR types, the canonicity proof involves all canonicity predicates.

Interpreting IR types in the glued model:

- We write $\mathsf{IR}\,S : \mathsf{Tm}\,\mathsf{U}$ for the IR *type formation* rule, for $S : \mathsf{Tm}\,(\mathsf{Sig}\,O)$.
- To interpret $\mathsf{IR}\,S$:
  - By induction hypothesis, we get $S^{\circ} : \mathsf{Sig}^{\circ}\,S$ witnessing the canonicity of the signature S itself.
  - We compute a metatheoretic indexed IR signature by induction on S°. This yields the canonicity predicate for $\mathsf{IR}\,S$.
- We also need to interpret *term formation*, *elimination* and *computation* rules.
- For these, we have to show that universal properties of IR types are preserved through the indexed IR encoding.

How to prove canonicity?

The construction is a moderately technical "generic programming" exercise, where we have to do some tricky induction over signatures.

We use metatheoretic IR to show canonicity of object-theoretic IR. The metatheory has to have more universes; in our case the metatheory has extra levels ω and $\omega + 1$[1].

The canonicity interpretation of IR is formalized in Agda using a shallow embedding of Ty and Tm.

[1] But $\omega + 1$ is only used for convenience and could be omitted.

Indexed Induction-Recursion

What about canonicity for **indexed** IR types?

We only show canonicity for plain IR types, but also show that indexed IR types are constructible in MLTT+IR.

The construction supports strict computation rules for all IIR types that are definable in Agda. For *neutral* signatures we only get propositional computation.

We use the well-known construction that converts indices to parameters and propositional identities.[2]

Here, since we work in plain MLTT without function extensionality and UIP, we have to do a modest amount of HoTT reasoning.

[2] Popularized as "fording" by Conor McBride.

Closing notes

- Future work:
  - Normalization for IR types. For this, we first have to show that presheaf models have IR types (which is already quite technical!).
  - Generalize the canonicity proof to a model construction (i.e. extend general type-theoretic gluing with IR types). This seems to be incompatible with first-class signatures.
- Paper: under review, I'll publish a preprint in a few weeks.

Closing notes

- Future work:
  - Normalization for IR types. For this, we first have to show that presheaf models have IR types (which is already quite technical!).
  - Generalize the canonicity proof to a model construction (i.e. extend general type-theoretic gluing with IR types). This seems to be incompatible with first-class signatures.
- Paper: under review, I'll publish a preprint in a few weeks.

Thank you!

Andreas Abel, Nils Anders Danielsson, and Oskar Eriksson.
A graded modal dependent type theory with a universe and erasure, formalized.
Proc. ACM Program. Lang., 7(ICFP):920–954, 2023.

Andreas Abel, Joakim Öhman, and Andrea Vezzosi.
Decidability of conversion for type theory in type theory.
Proc. ACM Program. Lang., 2(POPL):23:1–23:29, 2018.

Ana Bove and Venanzio Capretta.
Nested general recursion and partiality in type theory.
In Richard J. Boulton and Paul B. Jackson, editors, *Theorem Proving in Higher Order Logics, 14th International Conference, TPHOLs 2001, Edinburgh, Scotland, UK, September 3-6, 2001, Proceedings*, volume 2152 of *Lecture Notes in Computer Science*, pages 121–135. Springer, 2001.

Marcin Benke, Peter Dybjer, and Patrik Jansson.
Universes for generic programs and proofs in dependent type theory.
Nord. J. Comput., 10(4):265–289, 2003.

Thierry Coquand.
Canonicity and normalization for dependent type theory.
Theor. Comput. Sci., 777:184–191, 2019.

Jonathan Chan and Stephanie Weirich.
Bounded first-class universe levels in dependent type theory.

CoRR, abs/2502.20485, 2025.

Larry Diehl.
Fully Generic Programming over Closed Universes of Inductive-Recursive Types.
PhD thesis, Portland State University, 2017.

Peter Dybjer and Anton Setzer.
A finite axiomatization of inductive-recursive definitions.
In Jean-Yves Girard, editor, *Typed Lambda Calculi and Applications, 4th International Conference, TLCA'99, L'Aquila, Italy, April 7-9, 1999, Proceedings*, volume 1581 of *Lecture Notes in Computer Science*, pages 129–146. Springer, 1999.

Peter Dybjer and Anton Setzer.
Induction-recursion and initial algebras.
Ann. Pure Appl. Log., 124(1-3):1–47, 2003.

Peter Dybjer and Anton Setzer.
Indexed induction-recursion.
J. Log. Algebraic Methods Program., 66(1):1–49, 2006.

Peter Dybjer.
A general formulation of simultaneous inductive-recursive definitions in type theory.
Journal of Symbolic Logic, 65:525–549, 2000.

Ambrus Kaposi, Simon Huber, and Christian Sattler.
Gluing for type theory.
In Herman Geuvers, editor, *4th International Conference on Formal Structures for Computation and Deduction, FSCD 2019, June 24-30, 2019, Dortmund, Germany*, volume 131 of *LIPIcs*, pages 25:1–25:19. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2019.

András Kovács.
Generalized universe hierarchies and first-class universe levels.
In Florin Manea and Alex Simpson, editors, *30th EACSL Annual Conference on Computer Science Logic, CSL 2022, February 14-19, 2022, Göttingen, Germany (Virtual Conference)*, volume 216 of *LIPIcs*, pages 28:1–28:17. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2022.

András Kovács.
Large Countable Ordinals in Agda, 2023.

Chou Kyuhei.
Brouwer tree barrier ordinal, 2025.

Per Martin-Löf.
An intuitionistic theory of types: Predicative part.
In *Studies in Logic and the Foundations of Mathematics*, volume 80, pages 73–118. Elsevier, 1975.

Per Martin-Löf.
Intuitionistic type theory, volume 1 of *Studies in Proof Theory*.

Bibliopolis, 1984.

Loïc Pujet and Nicolas Tabareau.
Impredicative observational equality.
Proc. ACM Program. Lang., 7(POPL):2171–2196, 2023.