Canonicity and Decidability of Equality for Setoid Type Theory

István Donkó[1], Ambrus Kaposi

Faculty of Informatics
Eötvös Loránd University, Budapest

28th International Conference on Types for Proofs and Programs
(2022. June)

[1]Supported by the ÚNKP-21-3 New National Excellence Program of the Ministry for Innovation and Technology from the source of the National Research, Development and Innovation Fund.

Contents

Motivation

Motivation

Why?

Motivation

Why?

To express algebraic theories with equalities.

Motivation

Why?

To express algebraic theories with equalities.

What?

Motivation

Why?

To express algebraic theories with equalities.

What?

Quotient Inductive Types

Motivation

Why?

To express algebraic theories with equalities.

What?

Quotient Inductive Types

How?

Motivation

Why?

To express algebraic theories with equalities.

What?

Quotient Inductive Types

How?

Using Setoid Type Theory

Examples

Functional extensionality

- $C = A \to B$ $(A, B \in Set)$
- $f \sim g = \forall a \in A : f(a) = g(a)$

Propositional extensionality

- $C = Prop$
- $P \sim Q = (P \to Q) \times (Q \to P)$

Semantical program equivalence

- C : Syntax of programming language
- $\_ \sim \_$: Reflexive, symmetric and transitive closure of semantic transitions

Also: Bisimilarity of coinductive datatype instances

Example - Teaching

Type System course

- Algebraic models of programming languages
- Well typed syntax with equations

Example - Teaching

Type System course

- Algebraic models of programming languages
- Well typed syntax with equations

Problem

Agda inefficiency

- Manual transports
- Rewrite rules

Example - Teaching

Type System course

- Algebraic models of programming languages
- Well typed syntax with equations

Problem

Agda inefficiency

- Manual transports
- Rewrite rules

Aim

Better computability

- No additional axioms necessary
- Avoiding transport hell

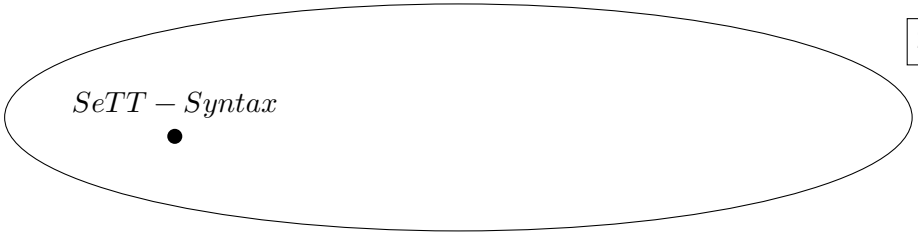
Setoid Type Theory

We need canonicity.

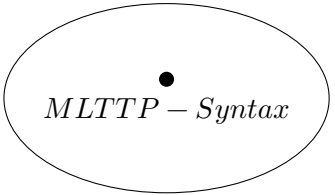
Previously:

- Observational Equality, Now! - Altenkirch et al. [2007]
- Cubical Syntax for Reflection-Free Extensional Equality - Sterling et al. [2019]
- Shallow Embedding of Type Theory is Morally Correct - Kaposi et al. [2019]
- Setoid type theory - a syntactic translation - Altenkirch et al. [2019]
- Constructing a universe for the setoid model - Altenkirch et al. [2021]
- Observational Equality: Now For Good - Pujet et al. [2021]

Overview



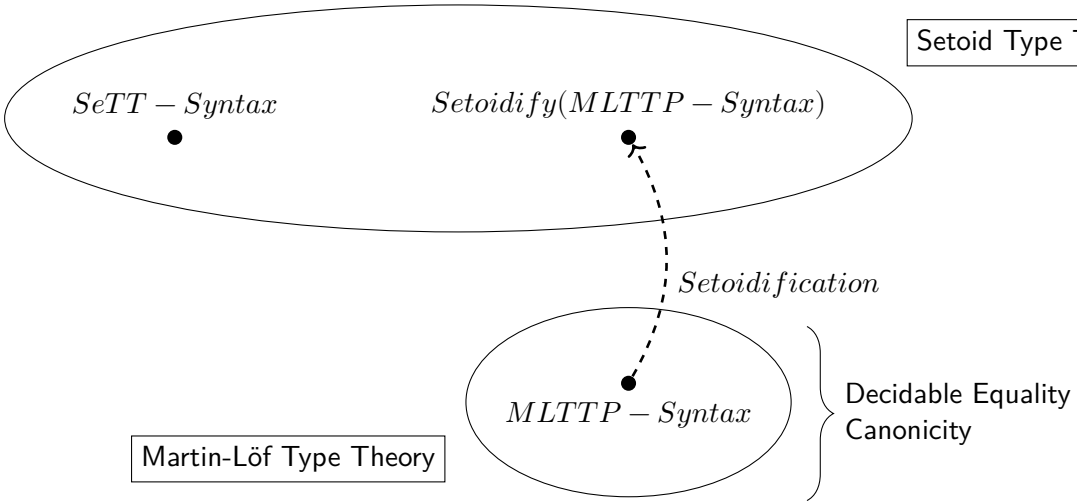
Setoid Type Theory



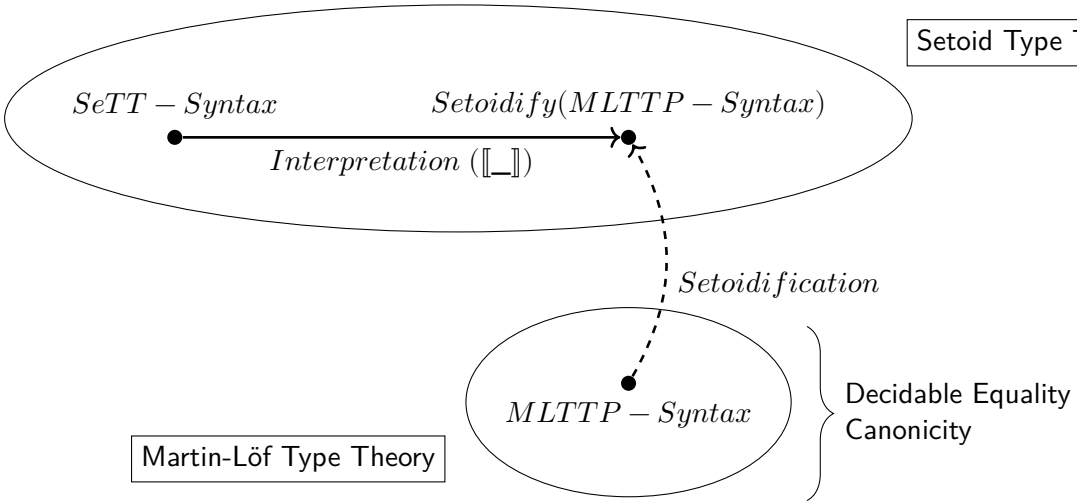
Martin-Löf Type Theory

Decidable Equality
Canonicity

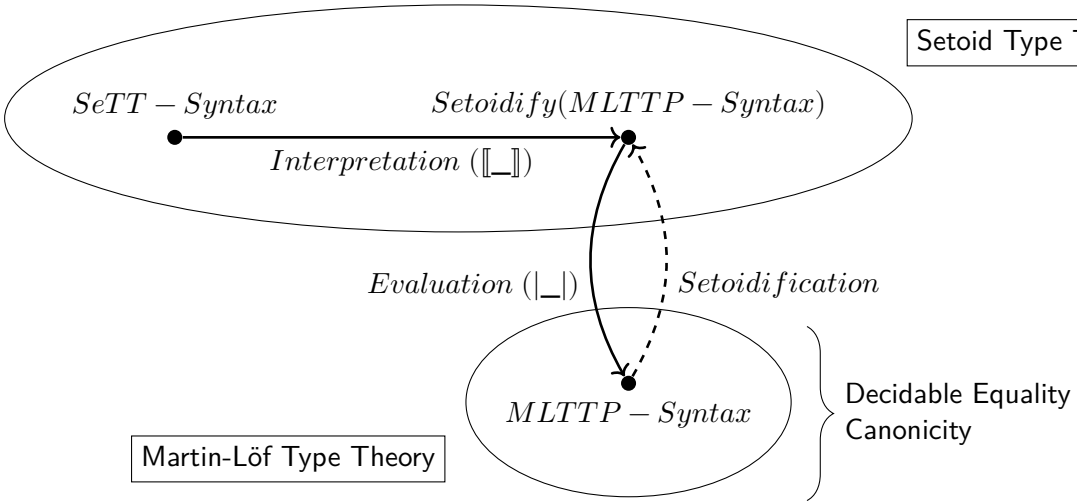
Overview


Setoid Type Theory
$SeTT - Syntax$
$Setoidify(MLTTP - Syntax)$
$Setoidification$
$MLTTP - Syntax$
Decidable Equality
Canonicity
Martin-Löf Type Theory

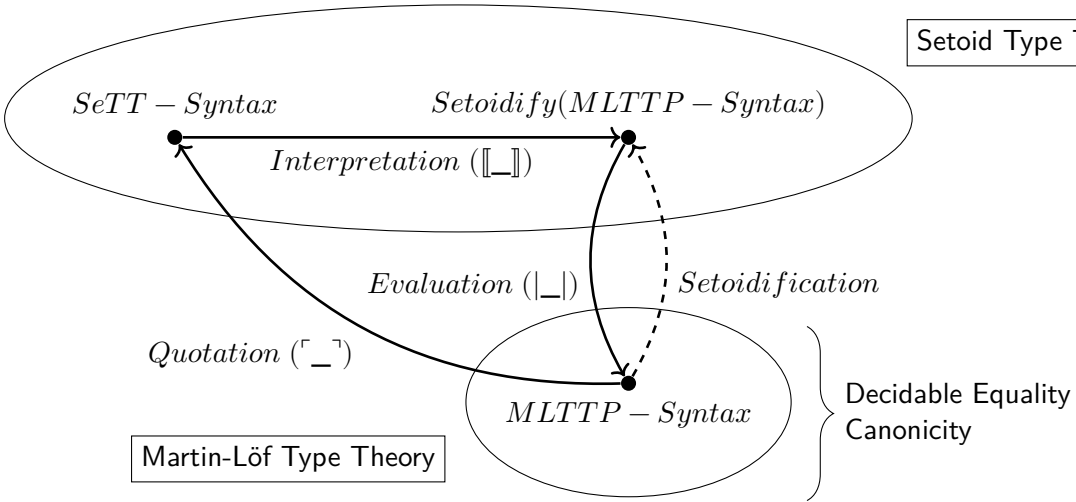
Overview


Setoid Type Theory
$SeTT - Syntax$
$Setoidify(MLTTP - Syntax)$
$Interpretation\ (\llbracket \_ \rrbracket)$
$Setoidification$
$MLTTP - Syntax$
Martin-Löf Type Theory
Decidable Equality
Canonicity

Overview


Setoid Type Theory
$SeTT - Syntax$
$Setoidify(MLTTP - Syntax)$
$Interpretation\ (\llbracket \_ \rrbracket)$
$Evaluation\ (|\_|)$
$Setoidification$
$MLTTP - Syntax$
Martin-Löf Type Theory
Decidable Equality
Canonicity

Overview


Setoid Type Theory
$SeTT - Syntax$
$Setoidify(MLTTP - Syntax)$
$Interpretation\ (\llbracket \_ \rrbracket)$
$Evaluation\ (|\_|)$
$Setoidification$
$Quotation\ (\ulcorner \_ \urcorner)$
$MLTTP - Syntax$
Martin-Löf Type Theory
Decidable Equality
Canonicity

Constructing a setoid model from an MLTTP model.

Types $(A : Setoidify(MLTTP\ model).Ty)$

- $|A|$: type from the input MLTTP model
- $A \sim$: heterogeneous equivalence relation over terms of the type
- RA, SA, TA : the relation is reflexive, symmetric and transitive
- $coeA$: coercion over context equivalences
- $cohA$: the result of coercion is in relation with the original

Properties

Let the combined $|[\![\_]\!]|$ operation be the interpretation and evaluation from the syntax of SeTT to the setoidified MLTTP syntax.
The injectivity of this composite function: $\underline{|[\![t]\!]| = |[\![t']\!]| \to t = t'}$ where

- t, t' are terms in the syntax of SeTT
- $|[\![t]\!]|, |[\![t']\!]|$ are terms in the syntax of MLTTP

Decidability of equality

- $|[\![t]\!]| = |[\![t']\!]|$ implies $t = t'$ by injectivity
- $|[\![t]\!]| \neq |[\![t']\!]|$ implies $t \neq t'$ by contradiction

Canonicity

For example, on type Bool:

- $|[\![t]\!]| = M.false$ implies $t = S.false$ by injectivity
- $|[\![t]\!]| = M.true$ implies $t = S.true$ by injectivity

Injectivity

Dependent model, where:

- $Con^\bullet\ \Gamma :=$ comparison map from Γ to $\ulcorner|[\![\Gamma]\!]|\urcorner$
- $Sub^\bullet\ \Delta^\bullet\ \Gamma^\bullet\ \gamma := (\Gamma^\bullet \circ \gamma = \ulcorner|[\![\gamma]\!]|\urcorner \circ \Delta^\bullet)$
- $Ty^\bullet\ \Gamma^\bullet\ A :=$ isomorphism between terms of type A and $\ulcorner|[\![A]\!]|\urcorner$
- $Tm^\bullet\ \Gamma^\bullet\ A^\bullet\ t := (A^\bullet[id, t] = \ulcorner|[\![t]\!]|\urcorner[\Gamma^\bullet])$

Injectivity from the dependent model:

- Given: $[\![t]\!] = [\![t']\!]$
- We have: $A^\bullet[id,\ t\ ] = \ulcorner|[\![t]\!]|\urcorner\ [\Gamma^\bullet] = \ulcorner|[\![t']\!]|\urcorner\ [\Gamma^\bullet] = A^\bullet[id,\ t'\ ]$

Issues

Eliminator terms for types without η rules applied on terms with context-dependent types.
Example: Bool + Sigma

$$\Sigma^{\bullet}[id, ite\ u\ v\ t] = ... = (\_, \_) \neq ite\ \_\ \_\ t = ... = \ulcorner[\![|ite\ u\ v\ t|]\!]\urcorner[\Gamma^{\bullet}]$$

Possible solutions:

- Add η rules?
- Prove the equalities type by type?

Thank you for your attention!

Questions?

Supported by the ÚNKP-21-3 New National Excellence Program of the Ministry for Innovation and Technology from the source of the National Research, Development and Innovation Fund.