Point-free equations for strict algebraic structures

Ambrus Kaposi[1]

[1]Eötvös Loránd University

26 May 2022, Budapest

National Research, Development and Innovation Office
Hungary

PROGRAM
FINANCED FROM
THE NRDI FUND

Contents

- Strict algebraic structures
- Point-free equations
- Summary of results of the last three years

Strict equations

There are two kinds of equalities, weak and strict.

Lotto: strict or weak?

$$
\begin{aligned}
(a) &\quad 1 + 1 = 2 \\
(b) &\quad x + 0 = x \\
(c) &\quad 0 + x = x \\
(d) &\quad x + y = y + x
\end{aligned}
$$

Strict equations

There are two kinds of equalities, weak and strict.

Lotto: strict or weak?

$$
\begin{aligned}
&(a) \quad 1 + 1 = 2 \\
&(b) \quad x + 0 = x \\
&(c) \quad 0 + x = x \\
&(d) \quad x + y = y + x
\end{aligned}
$$

Definition of addition on natural numbers:

$$
\begin{aligned}
0 \quad\quad\; &+ y := y \\
\mathrm{suc}(x) &+ y := \mathrm{suc}(x + y)
\end{aligned}
$$

Strict equations

There are two kinds of equalities, weak $=$ and strict $\equiv$.

Lotto: strict or weak?

$$
\begin{aligned}
&(a) \quad 1 + 1 \equiv 2 \\
&(b) \quad x + 0 = x \\
&(c) \quad 0 + x \equiv x \\
&(d) \quad x + y = y + x
\end{aligned}
$$

Definition of addition on natural numbers:

$$
\begin{aligned}
0 \qquad &+ y :\equiv y \\
\mathrm{suc}(x) &+ y :\equiv \mathrm{suc}(x + y)
\end{aligned}
$$

Strict equations in type theory

- We cannot talk about strict equalities.
  - $a \equiv b$ is a *judgement*, not a type
  - $a = b$ is a type

Strict equations in type theory

- We cannot talk about strict equalities.
  - $a \equiv b$ is a *judgement*, not a type
  - $a = b$ is a type
- However in intensional type theory without function extensionality
  - $(\forall x.f(x) = g(x)) \to f = g$

  strict and weak equality coincide in the empty context.

Strict equations in type theory

- We cannot talk about strict equalities.
  - $a \equiv b$ is a *judgement*, not a type
  - $a = b$ is a type
- However in intensional type theory without function extensionality
  - $(\forall x.f(x) = g(x)) \to f = g$

  strict and weak equality coincide in the empty context.
- Idea: we only state equations in the empty context. E.g.
  - $x + 0 = x$ is in a context including $x : \mathbb{N}$
  - $(\lambda x.x + y) = (\lambda x.x)$ is in the empty context

Example

A graph is a 4-tuple: $(V : Set) \times (E : Set) \times (dom : E \to V) \times (cod : E \to V)$.

A graph homomorphism from (V, E, dom, cod) to (V', E', dom', cod') is:

Example

A graph is a 4-tuple: $(V : Set) \times (E : Set) \times (dom : E \to V) \times (cod : E \to V)$.

A graph homomorphism from (V, E, dom, cod) to (V', E', dom', cod') is:

(a) $(f : V \to V')$

Example

A graph is a 4-tuple: $(V : Set) \times (E : Set) \times (dom : E \to V) \times (cod : E \to V)$.

A graph homomorphism from (V, E, dom, cod) to (V', E', dom', cod') is:

(a) $(f : V \to V') \times (g : E \to E')$

Example

A graph is a 4-tuple: $(V : Set) \times (E : Set) \times (dom : E \to V) \times (cod : E \to V)$.

A graph homomorphism from (V, E, dom, cod) to (V', E', dom', cod') is:

(a) $(f : V \to V') \times (g : E \to E') \times (\forall e. f(dom(e)) = dom'(g(e)))$

Example

A graph is a 4-tuple: $(V : Set) \times (E : Set) \times (dom : E \to V) \times (cod : E \to V)$.

A graph homomorphism from (V, E, dom, cod) to (V', E', dom', cod') is:

(a) $(f : V \to V') \times (g : E \to E') \times (\forall e. f(dom(e)) = dom'(g(e))) \times$ same for cod

Example

A graph is a 4-tuple: $(V : Set) \times (E : Set) \times (dom : E \to V) \times (cod : E \to V)$.

A graph homomorphism from (V, E, dom, cod) to (V', E', dom', cod') is:

(a) $(f : V \to V') \times (g : E \to E') \times (\forall e. f(dom(e)) = dom'(g(e))) \times$ same for cod

(b) $(f : V \to V') \times (g : E \to E') \times (\lambda e. f(dom(e))) = (\lambda e. dom'(g(e))) \times$ same for cod

Example

A graph is a 4-tuple: $(V : Set) \times (E : Set) \times (dom : E \to V) \times (cod : E \to V)$.

A graph homomorphism from (V, E, dom, cod) to (V', E', dom', cod') is:

(a) $(f : V \to V') \times (g : E \to E') \times (\forall e. f(dom(e)) = dom'(g(e))) \times$ same for cod

(b) $(f : V \to V') \times (g : E \to E') \times (\lambda e. f(dom(e))) = (\lambda e. dom'(g(e))) \times$ same for cod

We cannot define composition of homomorphisms in case (a). In case (b) it works.

Example

A graph is a 4-tuple: $(V : Set) \times (E : Set) \times (dom : E \to V) \times (cod : E \to V)$.

A graph homomorphism from (V, E, dom, cod) to (V', E', dom', cod') is:

(a) $(f : V \to V') \times (g : E \to E') \times (\forall e.f(dom(e)) = dom'(g(e))) \times$ same for cod
(b) $(f : V \to V') \times (g : E \to E') \times (\lambda e.f(dom(e))) = (\lambda e.dom'(g(e))) \times$ same for cod

We cannot define composition of homomorphisms in case (a). In case (b) it works.

Paper: using point-free equations to define the setoid model in intensional type theory

- Donkó, Kaposi: Internal strict propositions using point-free equations
- Conditionally accepted in the TYPES 2021 post-proceedings (LiPiCS).

TKP project in the past 3 years in numbers

Numbers for: Ambrus Kaposi, András Kovács, Rafaël Bocquet, István Donkó

- Published: 8 conference papers, 3 journal papers
- Submitted papers: 4 in the review process, 1 rejected
- 5 invited talks at workshops or online seminars
- Conference abstracts: at least 12
- 3 presentations at Researchers's Night, ELTE Eötvös College
- Rejected grant submissions: 1 ERC, 1 Lendület, 1 SNN
- Pending grant submission: 1 Lendület, 1 OTKA
- In the core group of the EuroProofNet COST Action
- 1 pending PhD defence (András Kovács)

TKP project in the past 3 years by subtopics

- Metatheory of type theory
  - Shallow embedding project: 1 published
  - Universes: 1 published
  - Point-free project: 1 submitted, in progress
  - Münchhausen: in progress
  - Coherent type theory: in progress
  - HOAS: 1 to be resubmitted
- New type theories
  - Setoid type theory: 4 published, 1 to write up, in progress
  - Higher observational type theory: in progress
- Applied type theory
  - Elaboration: 1 published
  - Staged compilation: 1 accepted
  - Implementation of setoid type theory: in progress

Collaborators