A type theory with internal parametricity

Ambrus Kaposi

Department of Programming Languages and Compilers,
Eötvös Loránd University,
Budapest

j.w.w. Thorsten Altenkirch, Yorgo Chamoun and Michael Shulman

6th World Logic Day
Pécs
12 January 2024

Structuralist language for mathematics

- Paul Benacerraf. What numbers could not be (1965)
  - Zermelo: $\{\emptyset, \{\emptyset\}, \{\{\emptyset\}\}, \dots\}$
  - von Neumann: $\{\emptyset, \{\emptyset\}, \{\emptyset, \{\emptyset\}\}, \dots\}$

Structuralist language for mathematics

- Paul Benacerraf. What numbers could not be (1965)
  - Zermelo: $\{\emptyset, \{\emptyset\}, \{\{\emptyset\}\}, \dots\}$
  - von Neumann: $\{\emptyset, \{\emptyset\}, \{\emptyset, \{\emptyset\}\}, \dots\}$
- Different names for the same idea:
  - structuralism
  - abstraction
  - representation independence
  - information hiding
  - uniformity
  - naturality
  - parametricity

John C. Reynolds (1935-2013)

Some contributions:

- polymorphic lambda calculus (System F by Girard)
- definitional interpreters
- defunctionalisation
- separation logic
- parametricity
  - Types, abstraction and parametric polymorphism (1983)

Reynolds' fable 1/3

Once upon a time, there was a university with a peculiar tenure policy. All faculty were tenured, and could only be dismissed for moral turpitude. What was peculiar was the definition of moral turpitude: making a false statement in class. Needless to say, the university did not teach computer science. However, it had a renowned department of mathematics.

One semester, there was such a large enrollment in complex variables that two sections were scheduled. In one section, Professor Descartes announced that a complex number was an ordered pair of reals, and that two complex numbers were equal when their corresponding components were equal. He went on to explain how to convert reals into complex numbers, what "i" was, how to add, multiply, and conjugate complex numbers, and how to find their magnitude.

Reynolds' fable 2/3

In the other section, Professor Bessel announced that a complex number was an ordered pair of reals the first of which was nonnegative, and that two complex numbers were equal if their first components were equal and either the first components were zero or the second components differed by a multiple of 2π. He then told an entirely different story about converting reals, "i", addition, multiplication, conjugation, and magnitude.

Then, after their first classes, an unfortunate mistake in the registrar's office caused the two sections to be interchanged. Despite this, neither Descartes nor Bessel ever committed moral turpitude, even though each was judged by the other's definitions. The reason was that they both had an intuitive understanding of type. Having defined complex numbers and the primitive operations upon them, thereafter they spoke at a level of abstraction that encompassed both of their definitions.

Reynolds' fable 3/3

The moral of this fable is that:

> Type structure is a syntactic discipline for enforcing levels of abstraction.

For instance, when Descartes introduced the complex plane, this discipline prevented him from saying Complex = Real × Real, which would have contradicted Bessel's definition. Instead, he defined the mapping f: Real × Real → Complex such that f(x, y) = x + i × y, and proved that this mapping is a bijection.

More subtly, although both lecturers introduced the set Int* of sequences of integers, and spoke of sets such as Int* + Complex, Int* × Complex, and Int* → Complex, they never mentioned Int* ∪ Complex or Int* ∩ Complex. Intuitively, they thought of sequences of integers and complex numbers as entities so immiscible that the union and intersection of Int* and Complex are undefined.

Reynolds' parametricity

- Everything preserves relations
- In the context of the polymorphic lambda calculus

Example of parametricity 1

$$\begin{array}{ll} f & : (A : \mathsf{Type}) \to A \to A \\ \\ R & : A \to B \to \mathsf{Type} \\ r & : R\, a\, b \\ \hline f^{\mathsf{P}}\, R\, r & : R\, (f\, A\, a)\, (f\, B\, b) \end{array}$$

Example of parametricity 1

$$\frac{\begin{array}{ll} f & : (A : \mathsf{Type}) \to A \to A \\ \\ g & : A \to B \\ a & : A \end{array}}{f^{\mathsf{P}}\, R\, r : g\,(f\,A\,a) = f\,B\,(g\,a)}$$

Example of parametricity 1

$$\frac{\begin{array}{l} f \quad : (A : \mathrm{Type}) \to A \to A \\ (\lambda x.b) : A \to B \end{array}}{f^{\mathrm{P}}\, R\, r \;: b = f\; B\; b}$$

Example of parametricity 2

$$
\begin{array}{ll}
f & : (A : \mathsf{Type}) \to A^* \to A^* \\
\\
R & : A \to B \to \mathsf{Type} \\
as & : A^* \\
bs & : B^* \\
rs & : R^*\ as\ bs \\
\hline
f^{\mathsf{P}}\ R\ rs & : R\,(f\ A\ as)\,(f\ B\ bs)
\end{array}
$$

Example of parametricity 2

$$
\begin{array}{ll}
f & : (A : \mathsf{Type}) \to A^* \to A^* \\
\\
g & : A \to B \\
\multicolumn{2}{l}{\mathsf{R\,a\,b} := (\mathsf{g\,a} = \mathsf{b})} \\
\multicolumn{2}{l}{R^*\, as\, bs = (g^*\, as = bs)} \\
as & : A^* \\
\hline
\multicolumn{2}{l}{f^{\mathsf{P}}\, R\, rs : g^*(f\, A\, as) = f\, B\, (g^*\, as)}
\end{array}
$$

Example of parametricity 2

$$\begin{aligned}
&f && : (A : \mathsf{Type}) \to A^* \to A^* \\
\\
&g && : A \to B \\
&\mathsf{R\,a\,b} && := (\mathsf{g\,a} = \mathsf{b}) \\
&R^*\, as\, bs && = (g^*\, as = bs) \\
&as && : A^*
\end{aligned}$$

$$\overline{\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad}$$

$$f^{\mathsf{P}}\; R\; rs : g^*(f\; A\; as) = f\; B\; (g^*\; as)$$

Examples:

$$\begin{aligned}
f &= \mathsf{reverse}, & g &= \mathsf{code} : \mathsf{Char} \to \mathbb{N} \\
f &= \mathsf{tail}, & g &= \mathsf{inc} : \mathbb{N} \to \mathbb{N}
\end{aligned}$$

Example of parametricity 2

$$
\begin{array}{ll}
f & : (A : \mathsf{Type}) \to A^* \to A^* \\
\\
g & : A \to B \\
\multicolumn{2}{l}{\mathrm{R\,a\,b} := (\mathrm{g\,a} = \mathrm{b})} \\
\multicolumn{2}{l}{R^*\ as\ bs = (g^*\ as = bs)} \\
as & : A^* \\
\hline
\multicolumn{2}{l}{f^{\mathsf{P}}\ R\ rs : g^*(f\ A\ as) = f\ B\ (g^*\ as)}
\end{array}
$$

Examples:

$$
\begin{array}{ll}
f = \mathsf{reverse}, & g = \mathsf{code} : \mathsf{Char} \to \mathbb{N} \\
f = \mathsf{tail}, & g = \mathsf{inc} : \mathbb{N} \to \mathbb{N}
\end{array}
$$

Not example:

$$
f = \mathsf{odds} : \mathbb{N} \to \mathbb{N}, \qquad g = \mathsf{inc} : \mathbb{N} \to \mathbb{N}
$$

$$
\mathsf{inc}^*\,(\mathsf{odds}\,[1,2,3]) = \mathsf{inc}^*\,[1,3] = [2,4] \neq [3] = \mathsf{odds}\,(\mathsf{inc}^*\,[1,2,3])
$$

Questions

	how many such fs?
$f : (A : \text{Type}) \to A \to A$	1
$f : (A : \text{Type}) \to A \to A \to A$	$?a$
$f : (A : \text{Type}) \to A$	$?b$
$f : (A : \text{Type}) \to A \to (A \to A) \to A$	$?c$

Questions

	how many such fs?
$f : (A : \text{Type}) \to A^* \to A^*$	1
$f : (A : \text{Type}) \to A \to A \to A$	2
$f : (A : \text{Type}) \to A$	0
$f : (A : \text{Type}) \to A \to (A \to A) \to A$	ω

Theories of representation-independence

Preservation of ...

- homomorphisms
  - natural transformation (category theory)
  - does not work for higher order (work towards this: directed type theory)
  - the above two examples are covered

Theories of representation-independence

Preservation of ...

- homomorphisms
  - natural transformation (category theory)
  - does not work for higher order (work towards this: directed type theory)
  - the above two examples are covered
- relations
  - parametricity
  - inconsistent with LEM

Theories of representation-independence

Preservation of ...

- homomorphisms
  - natural transformation (category theory)
  - does not work for higher order (work towards this: directed type theory)
  - the above two examples are covered
- relations
  - parametricity
  - inconsistent with LEM
- isomorphisms
  - Homotopy Type Theory, Voevodsky's univalence
  - consistent with LEM

Example which cannot be derived from naturality

$$
\begin{array}{ll}
f & : (A : \mathsf{Type}) \to A \to (A \to A) \to A \\
 & \\
R & : A_0 \to A_1 \to \mathsf{Type} \\
z_R & : R\, z_0\, z_1 \\
s_R & : \forall a_0, a_1 \,.\, R\, a_0\, a_1 \to R\, (s_0\, a_0)\, (s_1\, a_1) \\
\hline
f^{\mathsf{P}}\, R\, z_R\, s_R & : R\, (f\, A_0\, z_0\, s_0)\, (f\, A_1\, z_1\, s_1)
\end{array}
$$

Example which cannot be derived from naturality

$$f : (A : \mathsf{Type}) \to A \to (A \to A) \to A$$

$\mathbb{N}$ is the initial "pointed set with an endofunction" (PSE).

Example which cannot be derived from naturality

$$f : (A : \mathsf{Type}) \to A \to (A \to A) \to A$$

$\mathbb{N}$ is the initial "pointed set with an endofunction" (PSE).

$$
\begin{aligned}
\mathsf{Nat} \quad & := (A : \mathsf{Type}) \to A \to (A \to A) \to A \\
\mathsf{zero} \quad & := \lambda A\, z\, s.z \\
\mathsf{suc}\, n \quad & := \lambda A\, z\, s.s\, (n\, A\, z\, s) \\
\mathsf{ite}\, A\, z\, s\, n \quad & := n\, A\, z\, s \\
\mathsf{ite}\, A\, z\, s\, \mathsf{zero} \quad & = z \\
\mathsf{ite}\, A\, z\, s\, (\mathsf{suc}\, n) & = s\, (\mathsf{ite}\, A\, z\, s\, n)
\end{aligned}
$$

Example which cannot be derived from naturality

$$f : (A : \mathrm{Type}) \to A \to (A \to A) \to A$$

$\mathbb{N}$ is the initial "pointed set with an endofunction" (PSE).

$$\begin{aligned}
\mathrm{Nat} &:= (A : \mathrm{Type}) \to A \to (A \to A) \to A \\
\mathrm{zero} &:= \lambda A\, z\, s. z \\
\mathrm{suc}\, n &:= \lambda A\, z\, s. s\, (n\, A\, z\, s) \\
\mathrm{ite}\, A\, z\, s\, n &:= n\, A\, z\, s \\
\mathrm{ite}\, A\, z\, s\, \mathrm{zero} &= z \\
\mathrm{ite}\, A\, z\, s\, (\mathrm{suc}\, n) &= s\, (\mathrm{ite}\, A\, z\, s\, n)
\end{aligned}$$

This is already weakly initial.

Example which cannot be derived from naturality

$$f : (A : \text{Type}) \to A \to (A \to A) \to A$$

$\mathbb{N}$ is the initial "pointed set with an endofunction" (PSE).

$$\begin{aligned}
\text{Nat} &:= (A : \text{Type}) \to A \to (A \to A) \to A \\
\text{zero} &:= \lambda A\, z\, s. z \\
\text{suc}\, n &:= \lambda A\, z\, s. s\, (n\, A\, z\, s) \\
\text{ite}\, A\, z\, s\, n &:= n\, A\, z\, s \\
\text{ite}\, A\, z\, s\, \text{zero} &= z \\
\text{ite}\, A\, z\, s\, (\text{suc}\, n) &= s\, (\text{ite}\, A\, z\, s\, n)
\end{aligned}$$

This is already weakly initial.
We need that for any other PSE-homomorphism g from $(\text{Nat}, \text{zero}, \text{suc})$ to (A, z, s), we have that $g = \text{ite}\, A\, z\, s$.

Example which cannot be derived from naturality

$$
\begin{tikzcd}
(A_0, z_0, s_0) \arrow[rr, "g"] & & (A_1, z_1, s_1) \\
& (\mathsf{Nat}, \mathsf{zero}, \mathsf{suc}) \arrow[ul, "\mathsf{ite}\ A_0\ z_0\ s_0"] \arrow[ur, "\mathsf{ite}\ A_1\ z_1\ s_1"'] &
\end{tikzcd}
$$

Example which cannot be derived from naturality

$$
\begin{tikzcd}
(A_0, z_0, s_0) \arrow[rr, "g"] & & (A_1, z_1, s_1) \\
& (\mathrm{Nat}, \mathrm{zero}, \mathrm{suc}) \arrow[ul, "\mathrm{ite}\ A_0\ z_0\ s_0"] \arrow[ur, "\mathrm{ite}\ A_1\ z_1\ s_1"'] &
\end{tikzcd}
$$

From parametricity for an $n : \mathrm{Nat}$ taking R be the graph of g.

Example which cannot be derived from naturality

$$
\begin{tikzcd}
(A_0, z_0, s_0) \arrow[rr, "g"] & & (A_1, z_1, s_1) \\
& (\mathsf{Nat}, \mathsf{zero}, \mathsf{suc}) \arrow[ul, "\mathsf{ite}\ A_0\ z_0\ s_0"] \arrow[ur, "\mathsf{ite}\ A_1\ z_1\ s_1"'] &
\end{tikzcd}
$$

From parametricity for an $n : \mathsf{Nat}$ taking R be the graph of g.
We use that g is a PSE-homomorphism.

Example which cannot be derived from naturality

In particular:

$$\begin{tikzcd}
(\mathsf{Nat}, \mathsf{zero}, \mathsf{suc}) \arrow[rr, "\mathsf{ite}\, A\, z\, s"] & & (A, z, s) \\
& (\mathsf{Nat}, \mathsf{zero}, \mathsf{suc}) \arrow[ul, "\mathsf{ite}\, \mathsf{Nat}\, \mathsf{zero}\, \mathsf{suc}"] \arrow[ur, "\mathsf{ite}\, A\, z\, s"'] &
\end{tikzcd}$$

That is:

$$\mathsf{ite}\, A\, z\, s\, (\mathsf{ite}\, \mathsf{Nat}\, \mathsf{zero}\, \mathsf{suc}\, n) = \mathsf{ite}\, A\, z\, s\, n$$

Example which cannot be derived from naturality

In particular:

$$
\begin{tikzcd}
(\mathsf{Nat}, \mathsf{zero}, \mathsf{suc}) \arrow[rr, "\mathsf{ite}\, A\, z\, s"] & & (A, z, s) \\
& (\mathsf{Nat}, \mathsf{zero}, \mathsf{suc}) \arrow[ul, "\mathsf{ite}\,\mathsf{Nat}\,\mathsf{zero}\,\mathsf{suc}"] \arrow[ur, "\mathsf{ite}\, A\, z\, s"'] &
\end{tikzcd}
$$

That is:

$$\mathsf{ite}\, A\, z\, s\, (n\, \mathsf{Nat}\, \mathsf{zero}\, \mathsf{suc}) = n\, A\, z\, s$$

Example which cannot be derived from naturality

In particular:

$$\begin{tikzcd}
(\mathsf{Nat}, \mathsf{zero}, \mathsf{suc}) \arrow[rr, "\mathsf{ite}\, A\, z\, s"] & & (A, z, s) \\
& (\mathsf{Nat}, \mathsf{zero}, \mathsf{suc}) \arrow[ul, "\mathsf{ite}\, \mathsf{Nat}\, \mathsf{zero}\, \mathsf{suc}"] \arrow[ur, "\mathsf{ite}\, A\, z\, s"'] &
\end{tikzcd}$$

That is:

$$n\, \mathsf{Nat}\, \mathsf{zero}\, \mathsf{suc}\, A\, z\, s = n\, A\, z\, s$$

Example which cannot be derived from naturality

In particular:

$$
\begin{tikzcd}
(\mathrm{Nat}, \mathrm{zero}, \mathrm{suc}) \arrow[rr, "\mathrm{ite}\,A\,z\,s"] & & (A, z, s) \\
& (\mathrm{Nat}, \mathrm{zero}, \mathrm{suc}) \arrow[ul, "\mathrm{ite}\,\mathrm{Nat}\,\mathrm{zero}\,\mathrm{suc}"] \arrow[ur, "\mathrm{ite}\,A\,z\,s"'] &
\end{tikzcd}
$$

That is:

$$n\,\mathrm{Nat}\,\mathrm{zero}\,\mathrm{suc} = n$$

Example which cannot be derived from naturality

And we reach our goal by:

$$\begin{tikzcd}
(\mathsf{Nat}, \mathsf{zero}, \mathsf{suc}) \arrow[rr, "g"] & & (A, z, s) \\
& (\mathsf{Nat}, \mathsf{zero}, \mathsf{suc}) \arrow[ul, "\mathsf{ite}\,\mathsf{Nat}\,\mathsf{zero}\,\mathsf{suc}"] \arrow[ur, "\mathsf{ite}\,A\,z\,s"'] &
\end{tikzcd}$$

That is:

$$g\,(\mathsf{ite}\,\mathsf{Nat}\,\mathsf{zero}\,\mathsf{suc}\,n) = \mathsf{ite}\,A\,z\,s\,n$$

Example which cannot be derived from naturality

And we reach our goal by:

$$
\begin{tikzcd}
(\mathsf{Nat}, \mathsf{zero}, \mathsf{suc}) \arrow[rr, "g"] & & (A, z, s) \\
\\
& (\mathsf{Nat}, \mathsf{zero}, \mathsf{suc}) \arrow[uul, "\mathsf{ite}\,\mathsf{Nat}\,\mathsf{zero}\,\mathsf{suc}"] \arrow[uur, "\mathsf{ite}\,A\,z\,s"'] &
\end{tikzcd}
$$

That is:

$$g\,(n\,\mathsf{Nat}\,\mathsf{zero}\,\mathsf{suc}) = \mathsf{ite}\,A\,z\,s\,n$$

Example which cannot be derived from naturality

And we reach our goal by:

$$\begin{tikzcd}
(\mathsf{Nat}, \mathsf{zero}, \mathsf{suc}) \arrow[rr, "g"] & & (A, z, s) \\
& (\mathsf{Nat}, \mathsf{zero}, \mathsf{suc}) \arrow[ul, "\mathsf{ite}\,\mathsf{Nat}\,\mathsf{zero}\,\mathsf{suc}"] \arrow[ur, "\mathsf{ite}\,A\,z\,s"'] &
\end{tikzcd}$$

That is:

$$g\ n = \mathsf{ite}\,A\,z\,s\,n$$

Example which cannot be derived from naturality

And we reach our goal by:

$$
\begin{tikzcd}
(\mathsf{Nat}, \mathsf{zero}, \mathsf{suc}) \arrow[rr, "g"] & & (A, z, s) \\
& (\mathsf{Nat}, \mathsf{zero}, \mathsf{suc}) \arrow[ul, "\mathsf{ite}\,\mathsf{Nat}\,\mathsf{zero}\,\mathsf{suc}"] \arrow[ur, "\mathsf{ite}\,A\,z\,s"'] &
\end{tikzcd}
$$

That is:

$$g = \mathsf{ite}\,A\,z\,s$$

Parametricity for type theory

- Bernardy–Jansson–Paterson (2012) extended parametricity to Martin-Löf's type theory
  - A language for the structuralist formalisation of mathematics, e.g. $\mathbb{N}$ is defined as the initial PSE.
  - Proof assistants: Lean, Coq, Agda

Parametricity for type theory

- Bernardy–Jansson–Paterson (2012) extended parametricity to Martin-Löf's type theory
  - A language for the structuralist formalisation of mathematics, e.g. $\mathbb{N}$ is defined as the initial PSE.
  - Proof assistants: Lean, Coq, Agda
- The parametricity theorem can be expressed in the same language.

Parametricity for type theory

- Bernardy–Jansson–Paterson (2012) extended parametricity to Martin-Löf's type theory
  - A language for the structuralist formalisation of mathematics, e.g. $\mathbb{N}$ is defined as the initial PSE.
  - Proof assistants: Lean, Coq, Agda
- The parametricity theorem can be expressed in the same language.
- But is still a metatheorem.

Parametricity for type theory

- Bernardy–Jansson–Paterson (2012) extended parametricity to Martin-Löf's type theory
  - A language for the structuralist formalisation of mathematics, e.g. $\mathbb{N}$ is defined as the initial PSE.
  - Proof assistants: Lean, Coq, Agda
- The parametricity theorem can be expressed in the same language.
- But is still a metatheorem.
- Internalisation?

Parametricity for type theory

- Bernardy–Jansson–Paterson (2012) extended parametricity to Martin-Löf's type theory
  - A language for the structuralist formalisation of mathematics, e.g. $\mathbb{N}$ is defined as the initial PSE.
  - Proof assistants: Lean, Coq, Agda
- The parametricity theorem can be expressed in the same language.
- But is still a metatheorem.
- Internalisation?
  - A structuralist language which knows that it is structuralist.

Parametricity for type theory

- Bernardy–Jansson–Paterson (2012) extended parametricity to Martin-Löf's type theory
  - A language for the structuralist formalisation of mathematics, e.g. $\mathbb{N}$ is defined as the initial PSE.
  - Proof assistants: Lean, Coq, Agda
- The parametricity theorem can be expressed in the same language.
- But is still a metatheorem.
- Internalisation?
  - A structuralist language which knows that it is structuralist.
  - Difficulty: the witness of parametricity has to be parametric itself.

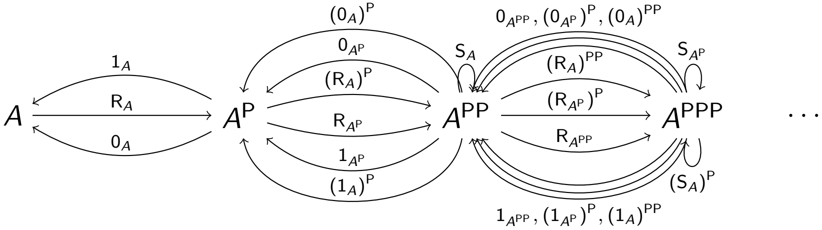
Iterated external parametricity

$$\begin{tikzcd}[column sep=huge]
A & A^{\mathsf{P}} \arrow[l, bend right, "1_A"'] \arrow[l, bend left, "0_A"] & A^{\mathsf{PP}} \arrow[l, bend right=50, "(0_A)^{\mathsf{P}}"'] \arrow[l, bend right=25, "0_{A^{\mathsf{P}}}"] \arrow[l, bend left=25, "1_{A^{\mathsf{P}}}"'] \arrow[l, bend left=50, "(1_A)^{\mathsf{P}}"] & A^{\mathsf{PPP}} \arrow[l, bend right=50, "{0_{A^{\mathsf{PP}}},\, (0_{A^{\mathsf{P}}})^{\mathsf{P}},\, (0_A)^{\mathsf{PP}}}"'] \arrow[l, bend right=45] \arrow[l, bend right=40] \arrow[l, bend left=40] \arrow[l, bend left=45] \arrow[l, bend left=50, "{1_{A^{\mathsf{PP}}},\, (1_{A^{\mathsf{P}}})^{\mathsf{P}},\, (1_A)^{\mathsf{PP}}}"] & \cdots
\end{tikzcd}$$

Iterated internal parametricity

$$\begin{tikzcd}[column sep=huge]
A \arrow[r, "\mathsf{R}_A"] & A^{\mathsf{P}} \arrow[l, bend right, "1_A"'] \arrow[l, bend left, "0_A"] \arrow[r, bend left=15, "(\mathsf{R}_A)^{\mathsf{P}}"] \arrow[r, bend right=15, "\mathsf{R}_{A^{\mathsf{P}}}"'] & A^{\mathsf{PP}} \arrow[l, bend right=50, "(0_A)^{\mathsf{P}}"'] \arrow[l, bend right=25, "0_{A^{\mathsf{P}}}"'] \arrow[l, bend left=25, "1_{A^{\mathsf{P}}}"] \arrow[l, bend left=50, "(1_A)^{\mathsf{P}}"] \arrow[loop above, "\mathsf{S}_A"] \arrow[r, bend left=20, "(\mathsf{R}_A)^{\mathsf{PP}}"] \arrow[r, "(\mathsf{R}_{A^{\mathsf{P}}})^{\mathsf{P}}"] \arrow[r, bend right=20, "\mathsf{R}_{A^{\mathsf{PP}}}"'] & A^{\mathsf{PPP}} \arrow[l, bend right=45, "{0_{A^{\mathsf{PP}}},\, (0_{A^{\mathsf{P}}})^{\mathsf{P}},\, (0_A)^{\mathsf{PP}}}"'] \arrow[l, bend right=50] \arrow[l, bend right=55] \arrow[l, bend left=45, "{1_{A^{\mathsf{PP}}},\, (1_{A^{\mathsf{P}}})^{\mathsf{P}},\, (1_A)^{\mathsf{PP}}}"] \arrow[l, bend left=50] \arrow[l, bend left=55] \arrow[loop above, "\mathsf{S}_{A^{\mathsf{P}}}"] \arrow[loop below, "(\mathsf{S}_A)^{\mathsf{P}}"] & \cdots
\end{tikzcd}$$

Iterated internal parametricity



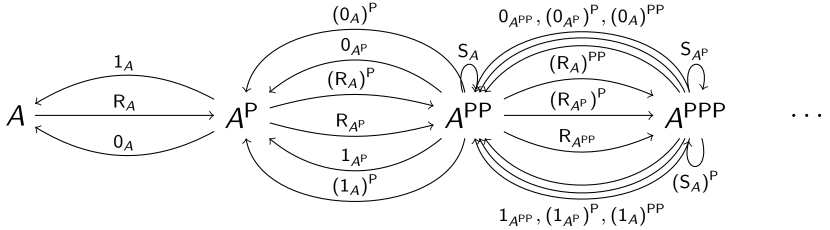
- We need a syntax for this. Becomes very complicated.

Iterated internal parametricity

$$
\begin{tikzcd}[column sep=huge]
A \arrow[r, "\mathsf{R}_A"] & A^{\mathsf{P}} \arrow[l, bend right, "1_A"'] \arrow[l, bend left, "0_A"] \arrow[r, bend left=15, "(\mathsf{R}_A)^{\mathsf{P}}"] \arrow[r, bend right=15, "\mathsf{R}_{A^{\mathsf{P}}}"'] & A^{\mathsf{PP}} \arrow[l, bend right=30, "0_{A^{\mathsf{P}}}"'] \arrow[l, bend right=60, "(0_A)^{\mathsf{P}}"'] \arrow[l, bend left=30, "1_{A^{\mathsf{P}}}"] \arrow[l, bend left=60, "(1_A)^{\mathsf{P}}"] \arrow[loop above, "\mathsf{S}_A"] \arrow[r, bend left=20, "(\mathsf{R}_A)^{\mathsf{PP}}"] \arrow[r, "(\mathsf{R}_{A^{\mathsf{P}}})^{\mathsf{P}}"] \arrow[r, bend right=20, "\mathsf{R}_{A^{\mathsf{PP}}}"'] & A^{\mathsf{PPP}} \arrow[l, bend right=50, "{0_{A^{\mathsf{PP}}},\, (0_{A^{\mathsf{P}}})^{\mathsf{P}},\, (0_A)^{\mathsf{PP}}}"'] \arrow[l, bend left=50, "{1_{A^{\mathsf{PP}}},\, (1_{A^{\mathsf{P}}})^{\mathsf{P}},\, (1_A)^{\mathsf{PP}}}"] \arrow[loop above, "\mathsf{S}_{A^{\mathsf{P}}}"] \arrow[loop below, "(\mathsf{S}_A)^{\mathsf{P}}"] & \cdots
\end{tikzcd}
$$

- We need a syntax for this. Becomes very complicated.
- Thierry Coquand's idea: $A^{\mathsf{P}} := \mathbb{I} \to A$.
  - Issue: substructural.

Iterated internal parametricity



- We need a syntax for this. Becomes very complicated.
- Thierry Coquand's idea: $A^{\mathsf{P}} := \mathbb{I} \to A$.
  - Issue: substructural.
- Our contribution:
  - $-^{\mathsf{P}}$, 0_{-}, 1_{-}, R_{-}, S_{-} and 5 equations generate everything.
  - Simple, structural syntax. Emergent geometry.
  - It computes!
  - Details: our paper "Internal parametricity, without an interval", POPL 2024.