On sum types and control operators

Gallium seminar, Rocquencourt, March 9, 2015

Danko ILIK (team Parsifal, ERC AdG ProofCert)

Contents

1

Sum types – Isomorphism

Types as exponential polynomials

The language of types

$$\mathcal{E} \ni f, g ::= 1 \mid x_i \mid f + g \mid f \times g \mid f \to g.$$

is the language of *exponential* polynomials

$$\mathcal{E} \ni f, g ::= 1 \mid x_i \mid f + g \mid fg \mid g^f.$$

Example

The type

$$f + g \to (f + g \to h) \to h$$

is written as

$$\left(h^{h^{f+g}}\right)^{f+g}$$

Type isomorphism generalizes arithmetic equality

Arithmetic equality $\mathbb{N}^+ \vDash f = g$ means:

> *For all substitutions of variables by positive natural numbers, f and g evaluate to the same number.*

Type isomorphism generalizes arithmetic equality

Arithmetic equality $\mathbb{N}^+ \models f = g$ means:

> *For all substitutions of variables by positive natural numbers, f and g evaluate to the same number.*

Type isomorphism $f \cong g$ means:

> *There are lambda terms $M : f \to g$ and $N : g \to f$ such that $\lambda x.N(Mx) =_{\beta\eta} \lambda x.x$ and $\lambda y.M(Ny) =_{\beta\eta} \lambda y.y$.*

Type isomorphism generalizes arithmetic equality

Arithmetic equality $\mathbb{N}^+ \models f = g$ means:

> *For all substitutions of variables by positive natural numbers, f and g evaluate to the same number.*

Type isomorphism $f \cong g$ means:

> *There are lambda terms $M : f \to g$ and $N : g \to f$ such that $\lambda x.N(Mx) =_{\beta\eta} \lambda x.x$ and $\lambda y.M(Ny) =_{\beta\eta} \lambda y.y$.*

Theorem

Type isomorphism generalizes arithmetic equality:

$$f \cong g \implies \mathbb{N}^+ \models f = g.$$

A proof system for type isomorphism?

Isomorphism/equality is a wild semantic notion. Can we axiomatize it?

A proof system for type isomorphism?

Isomorphism/equality is a wild semantic notion. Can we axiomatize it?

Definition (High School Identities (HSI))

$$
\begin{array}{ccc}
f \doteq f & f + g \doteq g + f & (f + g) + h \doteq f + (g + h) \\
fg \doteq gf & (fg)h \doteq f(gh) & f(g + h) \doteq fg + fh \\
1f \doteq f & f^1 \doteq f & 1^f \doteq 1 \\
f^{g+h} \doteq f^g f^h & (fg)^h \doteq f^h g^h & (f^g)^h \doteq f^{gh}
\end{array}
$$

HSI are sound as type isos:

$$\mathsf{HSI} \vdash f \doteq g \implies f \cong g$$

A proof system for type isomorphism?

Isomorphism/equality is a wild semantic notion. Can we axiomatize it?

Definition (High School Identities (HSI))

$$
\begin{array}{ccc}
f \doteq f & f + g \doteq g + f & (f + g) + h \doteq f + (g + h) \\
fg \doteq gf & (fg)h \doteq f(gh) & f(g + h) \doteq fg + fh \\
1f \doteq f & f^1 \doteq f & 1^f \doteq 1 \\
f^{g+h} \doteq f^g f^h & (fg)^h \doteq f^h g^h & (f^g)^h \doteq f^{gh}
\end{array}
$$

HSI are sound as type isos:

$$\mathsf{HSI} \vdash f \doteq g \implies f \cong g$$

Are they complete?

High School Algebra Problem (1960s)

Tarski, Skolem

*Are all **true** arithmetic equalities on $\mathbb{N}^+$ **provable** in HSI?*

That is, do we have

$$\mathbb{N}^+ \vDash f = g \implies \mathsf{HSI} \vdash f \doteq g?$$

High School Algebra Problem (1960s)

Tarski, Skolem

*Are all **true** arithmetic equalities on $\mathbb{N}^+$ **provable** in HSI?*

That is, do we have

$$\mathbb{N}^+ \vDash f = g \implies \mathsf{HSI} \vdash f \doteq g?$$

(If so, we could close the circle:

$$\mathbb{N}^+ \vDash f = g \Longrightarrow \mathsf{HSI} \vdash f \doteq g \Longrightarrow f \cong g \Longrightarrow \mathbb{N}^+ \vDash f = g)$$

Negative solution of HSA Problem

Martin 1973, Wilkie 1980, Gurevič 1990

Example (Wilkie)

Take

$$(A^x + B^x)^y (C^y + D^y)^x = (A^y + B^y)^x (C^x + D^x)^y,$$

where $A = 1 + x, B = 1 + x + x^2, C = 1 + x^3, D = 1 + x^2 + x^4$.
The equation holds in $\mathbb{N}^+$, but it is **not derivable** from the HSI axioms.

Theorem (Gurevič)

No finite extension *of HSI axiomatizes arithmetic equality in* $\mathbb{N}^+$.

Theorem (Fiore-DiCosmo-Balat 2004)

The counterexample equations also hold as type isomorphism.

Decidability of type isomorphism?

Richardson 1969, Macintyre 1981

What about decidability?

Theorem (Richardson 1969, Macintyre 1981)

One can effectively decide $\mathbb{N}^+ \vDash f = g$ *for any* $f, g \in \mathcal{E}$.

Decidability of type isomorphism?

Richardson 1969, Macintyre 1981

What about decidability?

Theorem (Richardson 1969, Macintyre 1981)

One can effectively decide $\mathbb{N}^+ \vDash f = g$ *for any* $f, g \in \mathcal{E}$.

Unfortunately, although

$$\mathsf{HSI} \vdash f \doteq g \;\Rightarrow\; f \cong g \;\Rightarrow\; \mathbb{N}^+ \vDash f = g,$$

a proof of

$$f \cong g \;\Leftarrow\; \mathbb{N}^+ \vDash f = g$$

is not known, and HSI is not complete:

$$\mathsf{HSI} \vdash f \doteq g \;\not\Leftarrow\; \mathbb{N}^+ \vDash f = g.$$

The $\mathcal{L}$ class of types

Levitz 1975, Henson-Rubel 1984, Gurevič 1993

Definition (The subclass $\mathcal{L}$)

$$\mathcal{L} \ni f, g ::= 1 \mid x_i \mid f + g \mid fg \mid l^f,$$

where $l \in \Lambda$ is defined by

$$\Lambda \ni f, g ::= 1 \mid x_i \mid f + g \mid fg \mid l_0^f,$$

and $l_0 \in \Lambda$ has no variables.

Theorem (Henson-Rubel 1984)

For all $f, g \in \mathcal{L}$,

$$\mathbb{N}^+ \vDash f \equiv g \;\Rightarrow\; HSI \vdash f \doteq g.$$

Corollary

Type isomorphisms for $\mathcal{L}$ is decidable and finitely axiomatizable.

Example

Consider again Wilkie's identity

$$(A^x + B^x)^y (C^y + D^y)^x \equiv (A^y + B^y)^x (C^x + D^x)^y,$$

where $A = 1 + x, B = 1 + x + x^2, C = 1 + x^3, D = 1 + x^2 + x^4$.

We have $(A^x + B^x)^y, (C^x + D^x)^y \notin \mathcal{L}$, because bases of exponentiation are not allowed to contain bases of exponentiation that contain variables

Example

The simply typed versions of the induction axiom for a decidable predicate,

$$(y+z)^{x(y+z)(y+z)^{x(y+z)}} \in \mathcal{L},$$

but its curried form,

$$\Big(\big((y+z)^x\big)^{((y+z)^{y+z})^x}\Big)^{y+z} \notin \mathcal{L}$$

although the two terms are inter-derivable using the HSI axioms.

Example

The simply typed versions of the induction axiom for a decidable predicate,

$$(y+z)^{x(y+z)(y+z)^{x(y+z)}} \in \mathcal{L},$$

but its curried form,

$$\Big(\big((y+z)^x\big)^{((y+z)^{y+z})^x}\Big)^{y+z} \notin \mathcal{L}$$

although the two terms are inter-derivable using the HSI axioms.

This means that one could in principle further extend $\mathcal{L}$.

Positive solution of HSA Problem

Wilkie 1980

For the whole of $\mathcal{E}$, the axioms of HSI are *almost* complete.

Positive solution of HSA Problem

Wilkie 1980

For the whole of $\mathcal{E}$, the axioms of HSI are *almost* complete.
Define

$$\mathcal{E}^* \ni f, g ::= t_z \mid 1 \mid x_i \mid g^f \mid fg \mid f + g,$$

where z is a *(positive) polynomial with integer monomial coefficients* and t_z are new constant symbols indexed by such polynomials.

Positive solution of HSA Problem

Wilkie 1980

For the whole of $\mathcal{E}$, the axioms of HSI are *almost* complete.
Define

$$\mathcal{E}^* \ni f, g ::= t_z \mid 1 \mid x_i \mid g^f \mid fg \mid f + g,$$

where z is a *(positive) polynomial with integer monomial coefficients* and t_z are new constant symbols indexed by such polynomials.
Define HSI* by extending HSI with

$$\begin{aligned} t_1 &\doteq 1 \\ t_{x_i} &\doteq x_i \\ t_{zu} &\doteq t_z t_u \\ t_{z+u} &\doteq t_z + t_u \\ t_z &\doteq t_u \qquad \text{(when } \mathbb{N}^+ \vDash z \equiv u\text{)} \end{aligned}$$

Positive solution of HSA Problem

Wilkie 1980

Theorem (Wilkie 1981)

*For all $f, g \in \mathcal{E}$ (i.e. all f, g of $\mathcal{E}^*$ that do **not** contain t_z-symbols), we have that $\mathbb{N}^+ \vDash f \equiv g$ implies HSI* $\vdash f \doteq g$.*

Corollary

Type isomorphism for $\mathcal{E}$ is axiomatizable by the primitively recursive set HSI.*

Corollary (LICS 2014)

Type isomorphism for $\mathcal{E}$ is decidable for base types finite sets.

Open questions

- Practical decision procedures?
  - In absence of g^f, complexity $O(n \log n)$
  - In absence of $f + g$, complexity $O(n \log n)$ (Considine 2000, Gil-Zibin 2005, 2007)
- Practical completion procedures?
  - Could be used in inductive theorem proving

2

Sum types – Canonical forms

Canonical representatives for the class of $=_{\beta\eta}$-terms of λ-calculus with sums

Example 1

Quiz

Which of the following η-long β-normal terms of type

$$f + g \to (f + g \to h) \to h$$

is *the* canonical one?

$$\lambda x.\lambda y.\delta(x, z.y(\mathsf{inl}\, z), z.y(\mathsf{inr}\, z)) \tag{1}$$

$$\lambda x.\delta(x, z.\lambda y.y(\mathsf{inl}\, z), z.\lambda y.y(\mathsf{inr}\, z)) \tag{2}$$

$$\lambda x.\lambda y.y\delta(x, z.\mathsf{inl}\, z, z.\mathsf{inr}\, z). \tag{3}$$

Canonical representatives and Identity of Proofs

Empirical evidence suggests problem is not easy:

- Proof Theory: Identity of Proofs problem (Kreisel, Prawitz 1960s)
  - Focusing Sequent Calculi solve the problem when no sums present
- Lambda Calculus
  - Dougherty-Subrahmanyam 1995
  - Ghani 1995
  - Altenkirch-Dybjer-Hofmann-Scott 2001
  - Balat-DiCosmo-Fiore 2004, Balat 2009
  - Lindley 2007
  - Ahmad-Licata-Harper 2010 (unpublished)

Identity of Proofs through Focusing

Focusing Sequent Calculi:

- Minimize backtracking during proof search
- Accept only η-long proof terms
- Unifies 2 of the 3 terms of Example 1
- Deterministic (“asynchronous”) phases essentially **apply type isomorphisms**

Identity of Proofs through Focusing

Focusing Sequent Calculi:

- Minimize backtracking during proof search
- Accept only η-long proof terms
- Unifies 2 of the 3 terms of Example 1
- Deterministic (“asynchronous”) phases essentially **apply type isomorphisms**

Focusing Next-Generation

Apply all applicable type isomorphisms!

Apply all applicable type isomorphisms

Recall the arithmetic notation for $f + g \to (f + g \to h) \to h$:

$$\left(h^{h^{f+g}}\right)^{f+g}.$$

Apply all applicable type isomorphisms

Recall the arithmetic notation for $f + g \to (f + g \to h) \to h$:

$$\left(h^{h^{f+g}}\right)^{f+g}.$$

Normalize to

$$h^{fh^f h^g}\, h^{gh^f h^g}$$

i.e. the type

$$(f \times (f \to h) \times (g \to h) \to h) \times (g \times (f \to h) \times (g \to h) \to h).$$

Apply all applicable type isomorphisms

Example 1

Recall the arithmetic notation for $f + g \to (f + g \to h) \to h$:

$$\left(h^{h^{f+g}}\right)^{f+g}.$$

Normalize to

$$h^{fh^f h^g}\, h^{gh^f h^g}$$

i.e. the type

$$(f \times (f \to h) \times (g \to h) \to h) \times (g \times (f \to h) \times (g \to h) \to h).$$

Since there are no sum types, one gets a **unique** canonical representative for terms of this type.

Generalizing the canonicity problem

The original problem is too specialized – generalize it!

Instead of picking a unique representative for the class of $=_{\beta\eta}$-equal terms of type τ, do so for the **enlarged** class of terms of **types isomorphic to** τ.

Problem (in the light of results from Section 1)

Is there a normal form of **types** in presence of sums?

Exp-log normal form of types (ENF)

Du Bois-Reymond, Hardy 1910

Generalization of the Disjunctive Normal Form (DNF) handling $f \to g$ by the equation

$$g^f = e^{f \log g}$$

Exp-log normal form of types (ENF)

Du Bois-Reymond, Hardy 1910

Generalization of the Disjunctive Normal Form (DNF) handling $f \to g$ by the equation

$$g^f = e^{f \log g}$$

Rewriting presentation (avoiding e and log):

$$\begin{aligned}
(f + g) + h &\mapsto f + (g + h) \\
(f \times g) \times h &\mapsto f \times (g \times h) \\
f \times (g + h) &\mapsto f \times g + f \times h \\
(g + h) \to f &\mapsto (g \to f) \times (h \to f) \\
h \to f \times g &\mapsto (h \to f) \times (h \to g) \\
h \to (g \to f) &\mapsto h \times g \to f
\end{aligned}$$

Exp-log normal form of types (ENF)

Du Bois-Reymond, Hardy 1910

Generalization of the Disjunctive Normal Form (DNF) handling $f \to g$ by the equation

$$g^f = e^{f \log g}$$

Rewriting presentation (avoiding e and log):

$$\begin{aligned}
(f + g) + h &\mapsto f + (g + h) \\
(f \times g) \times h &\mapsto f \times (g \times h) \\
f \times (g + h) &\mapsto f \times g + f \times h \\
(g + h) \to f &\mapsto (g \to f) \times (h \to f) \\
h \to f \times g &\mapsto (h \to f) \times (h \to g) \\
h \to (g \to f) &\mapsto h \times g \to f
\end{aligned}$$

Conjecture: ENF provides a decision procedure for *provably* isomorphic types.

Exp-log normal form of types (ENF)

Theorem

If a type is in ENF then it satisfies the inductive definition of D-type:

$$\begin{aligned} \textit{D-type} \ni d &::= c \mid c + d \\ \textit{C-type} \ni c, c' &::= a \mid a \times c \\ \textit{Atom} \ni a &::= p \mid c \to p \mid c' \to c + d, \end{aligned}$$

where p is a type variable (atomic logical proposition).

Computing canonical representatives

1. Full β-reduction to ENF-restricted NF
2. η-Expansion along ENF-restricted evaluation contexts

1. Full β-reduction to ENF-restricted NF representatives

Use a Normalization-by-evaluation (NBE) program (written in CPS) to obtain

$$R ::= E \mid \lambda x^{c}.R^{p} \mid \lambda x^{c'}.R^{c+d} \mid \mathsf{pair}\, R_1\, R_2\, {}^{a\times c} \mid (\mathsf{inl}\, R)^{c+d} \mid (\mathsf{inr}\, R)^{c+d}$$
$$\mid \delta(E, x_1.R_1, x_2.R_2)$$
$$E ::= x^{p} \mid x^{a\times c} \mid E^{c\to p} R^{c} \mid E^{c'\to c+d} R^{c'} \mid \mathsf{fst}\,(E^{a\times c}) \mid \mathsf{snd}\,(E^{a\times c}).$$

2. η-Expansion along ENF-restricted evaluation contexts

Computing canonical representatives

Eta-expansion $|-|^{\tau}$ for the axiom

$$N[M] =_{\eta} \delta(M, x.N[\mathsf{inl}\, x], y.N[\mathsf{inr}\, y]) \qquad (\eta^{+})$$

is done using the clauses:

$$|ER|^{c+d} \mapsto \delta(|E|^{c' \to c+d}\, |R|^{c'}, x_1.\mathsf{inl}\, x_1, x_2.\mathsf{inr}\, x_2)$$

$$\left|\delta(E^{c+d}, x_1.R_1[x_1], x_2.R_2[x_2])\right|^{d'} \mapsto \delta(|E|^{c+d}, x_1.|R_1[x_1]|^{d'}, x_2.|R_2[x_2]|^{d'})$$

$$\left|N[\mathsf{inr}\,[E^{c+d}]]\right|^{d'} \mapsto \delta(|E|^{c+d}, x_1.N\,[\mathsf{inr}\,\mathsf{inl}\, x_1], x_2.N\,[\mathsf{inr}\,\mathsf{inr}\, x_2])$$

$$\left|N\left[\delta([E^{c+d}], x_1.R[x_1], x_2.R[x_2])\right]\right|^{d'} \mapsto \delta(|E|^{c+d}, x_1.N\left[|R_1[x_1]|^{d'}\right], x_2.N\left[|R_2[x_2]|^{d'}\right])$$

2. η-Expansion along ENF-restricted evaluation contexts

Computing canonical representatives

Example

$$N\left[\delta([E^{c+d}], x_1.R[x_1], x_2.R[x_2])\right] =_\eta$$
$$\delta(E, y_1.N\,[\delta(\mathsf{inl}\, y_1, x_1.R_1[x_1], x_2.R_2[x_2])], y_2.N\,[\delta(\mathsf{inr}\, y_2, x_1.R_1[x_1], x_2.R_2[x_2])]) =_\beta$$
$$\delta(E, y_1.N\,[R_1[y_1]], y_2.N\,[R_2[y_2]]).$$

Quiz

Which of the following η-long β-normal terms of type

$$f + g \to (f + g \to h) \to h$$

is *the* canonical one?

$$\lambda x.\lambda y.\delta(x, z.y(\mathsf{inl}\, z), z.y(\mathsf{inr}\, z))$$
$$\lambda x.\delta(x, z.\lambda y.y(\mathsf{inl}\, z), z.\lambda y.y(\mathsf{inr}\, z))$$
$$\lambda x.\lambda y.y\delta(x, z.\mathsf{inl}\, z, z.\mathsf{inr}\, z).$$

Quiz

Which of the following η-long β-normal terms of type

$$f + g \to (f + g \to h) \to h$$

is *the* canonical one?

$$\lambda x.\lambda y.\delta(x, z.y(\mathsf{inl}\, z), z.y(\mathsf{inr}\, z))$$
$$\lambda x.\delta(x, z.\lambda y.y(\mathsf{inl}\, z), z.\lambda y.y(\mathsf{inr}\, z))$$
$$\lambda x.\lambda y.y\delta(x, z.\mathsf{inl}\, z, z.\mathsf{inr}\, z).$$

$$\langle \lambda x.(\mathsf{fst}\,(\mathsf{snd}\, x))(\mathsf{fst}\, x), \lambda x.(\mathsf{snd}\,(\mathsf{snd}\, x))(\mathsf{fst}\, x)\rangle$$

The $\beta\eta$-equal terms

$$\lambda xyzu.x(yz) \tag{4}$$

$$\lambda xyzu.\delta(\delta(u, x_1.\mathsf{inl}\, z, x_2.\mathsf{inr}\,(yz)), y_1.x(yy_1), y_2.xy_2). \tag{5}$$

of type

$$(f \to g) \to (h \to f) \to h \to i + j \to g$$

are normalized the two to the unique form

$$\langle \lambda x.(\mathsf{fst}\, x)((\mathsf{fst}\, \mathsf{snd}\, x)(\mathsf{fst}\, \mathsf{snd}\, \mathsf{snd}\, x)),$$
$$\lambda x.(\mathsf{fst}\, x)((\mathsf{fst}\, \mathsf{snd}\, x)(\mathsf{fst}\, \mathsf{snd}\, \mathsf{snd}\, x))\rangle.$$

This example shows that the normalizer can in addition handle vacuous η-expansions.

Example

The following terms of type $(f \to g) \to (h \to g) \to i \to (i \to f + h) \to g$,

$$\lambda xyzu.\delta(uz, w.xw, w.yw)$$
$$\lambda xyzu.\delta(uz, w.\delta(uz, w'.xw', w'.yw'), w.yw),$$

are normalized at their ENF type to themselves (modulo uncurrying) i.e. to two *different* canonical forms. This example shows the importance of the restriction of (η^+) to evaluation contexts, namely our normalizer does not target duplicated subterms.

Example

The following terms of type $k \to l \to (f \to g + h) \to (f \to i + j) \to f \to k + l$,

$$\lambda xyzuv.\delta(zv, x_1.\mathrm{inl}\, x, x_2.\delta(uv, y_1.\mathrm{inr}\, y, y_2.\mathrm{inl}\, x))$$
$$\lambda xyzuv.\delta(uv, y_1.\delta(zv, x_1.\mathrm{inl}\, x, x_2.\mathrm{inr}\, y), y_2.\mathrm{inl}\, x),$$

like in the previous example, are normalized at their ENF type to the uncurried versions of themselves. This example shows that our normalizer does not handle permuting conversions, although such conversions are a consequence of the *categorical* η^+-axiom. A way to obtain canonical terms for permuting conversions is to fix an ordering on terms deciding whether $zv \preceq uv$, as suggested by Altenkirch & al.

Example

The following terms of type $(f + g) \to h \to h \to h$,

$$\lambda xyz.y \qquad \lambda xyz.z \qquad \lambda xyz.\delta(x, x_1.y, x_2.z),$$

are normalized to

$$\langle \lambda x.\, \mathsf{fst}\, \mathsf{snd}\, x, \lambda x.\, \mathsf{fst}\, \mathsf{snd}\, x \rangle$$
$$\langle \lambda x.\, \mathsf{snd}\, \mathsf{snd}\, x, \lambda x.\, \mathsf{snd}\, \mathsf{snd}\, x \rangle$$
$$\langle \lambda x.\, \mathsf{fst}\, \mathsf{snd}\, x, \lambda x.\, \mathsf{snd}\, \mathsf{snd}\, x \rangle.$$

Although the similarity between the output terms is as good as it gets, the three output terms are distinct canonical forms.

Open problems

- Efficient implementation through sharing: Types and terms as DAGs rather than trees
- What is Eta-equality vs Observational equality?

3

Control operators as syntactic sugar

Control operators in Proof Theory

Extend the Curry-Howard correspondence to “classical logic”.

But, there are classically true formulas that are not recursively realizable, ex.

$$\forall x \exists y \forall z \exists u \big((u = 0 \to T(x, x, y)) \wedge (u \neq 0 \to T(x, x, z)) \big)$$

For realizable formulas of Arithmetic, the programming language System T suffices.

Do we need more than System T for classical Analysis?

The choice in Proof Theory:

1. System T + Bar Recursion
2. System T + “computational side-effects”

Do we need more than System T for classical Analysis?

The choice in Proof Theory:

1. System T + Bar Recursion
2. System T + “computational side-effects”

This talk:

Control operators extend System T conservatively (in absence of real PL side-effects).

System T^+

$$\text{hyp}\ \frac{}{(\sigma;\gamma)\vdash\sigma} \qquad \text{wkn}\ \frac{\gamma\vdash\sigma}{(\tau;\gamma)\vdash\sigma} \qquad \text{lam}\ \frac{(\sigma;\gamma)\vdash\tau}{\gamma\vdash\sigma\to\tau}$$

$$\text{app}\ \frac{\gamma\vdash\sigma\to\tau \quad \gamma\vdash\sigma}{\gamma\vdash\tau} \qquad \text{pair}\ \frac{\gamma\vdash\sigma \quad \gamma\vdash\tau}{\gamma\vdash\sigma*\tau} \qquad \text{fst}\ \frac{\gamma\vdash\sigma*\tau}{\gamma\vdash\sigma}$$

$$\text{snd}\ \frac{\gamma\vdash\sigma*\tau}{\gamma\vdash\tau} \qquad \text{zero}\ \frac{}{\gamma\vdash\mathbb{N}} \qquad \text{succ}\ \frac{\gamma\vdash\mathbb{N}}{\gamma\vdash\mathbb{N}}$$

$$\text{rec}\ \frac{\gamma\vdash\mathbb{N} \quad \gamma\vdash\sigma \quad \gamma\vdash\mathbb{N}\to\sigma\to\sigma}{\gamma\vdash\sigma} \qquad \text{shift}\ \frac{(\sigma\to\mathbb{N};\gamma)\vdash\mathbb{N}}{\gamma\vdash\sigma}$$

$$\mathrm{A} := \lambda m.\, \mathrm{R}\, m(\lambda n.n+1)(\lambda m'.\lambda u.\lambda n.\, \mathrm{R}\, n(u1)(\lambda n'.\lambda w.uw)),$$

is represented by

```
lam
  (rec hyp(lam(succ hyp))
     (lam
        (lam
           (lam
              (rec hyp(app(wkn hyp)(succ zero))
                 (lam(lam(app(wkn(wkn(wkn hyp))) hyp)))))))))
```

i.e. a 1st-order representation with de Bruijn indices 0 := hyp, 1 := wkn hyp, ...

Theorem (Normalization)

*There is a normalization function $\downarrow [\![-]\!]$ s.t. for every term p of **System T^+** of type $\gamma \vdash \tau$, the term $\downarrow [\![p]\!]$ is a normal form of **System T** of the same type ($\gamma \vdash_r \tau$).*

Theorem (Normalization)

*There is a normalization function $\downarrow [\![-]\!]$ s.t. for every term p of **System T$^+$** of type $\gamma \vdash \tau$, the term $\downarrow [\![p]\!]$ is a normal form of **System T** of the same type ($\gamma \vdash_T \tau$).*

Theorem (Equations)

$$\downarrow [\![\mathit{wkn}\, p]\!]_{\alpha,\rho} = \downarrow [\![p]\!]_\rho \qquad \downarrow [\![\mathit{hyp}]\!]_{\alpha,\rho} = \downarrow \alpha$$

$$\downarrow [\![\mathit{fst}\, \mathit{pair}(p,q)]\!]_\rho = \downarrow [\![p]\!]_\rho \qquad \downarrow [\![\mathit{snd}\, \mathit{pair}(p,q)]\!]_\rho = \downarrow [\![q]\!]_\rho$$

$$\downarrow [\![\mathit{app}(\mathit{lam}\, p, q)]\!]_\rho = \downarrow [\![p]\!]_{[\![q]\!]_\rho,\rho} \qquad \downarrow [\![\mathit{rec}(\mathit{zero}, p, q)]\!]_\rho = \downarrow [\![p]\!]_\rho$$

$$\downarrow [\![\mathit{rec}(\mathit{succ}\, r, p, q)]\!]_\rho = \cdots$$

$$\downarrow^{\mathbb{N}} [\![\mathit{shift}\, p]\!]_\rho = \downarrow^{\mathbb{N}} [\![p]\!]_{\phi,\rho} \qquad \downarrow^{\mathbb{N}} [\![\mathit{app}(\mathit{app}(\mathit{hyp}, x), y)]\!]_{\phi,\rho} = \downarrow^{\mathbb{N}} [\![y]\!]_{\phi,\rho}$$

$$\phi := \eta(\geq_3 \alpha \mapsto \eta(\mu\alpha))$$

System T (*not* T^+)

Normal terms ($\vdash_{\mathsf{r}}$),

$$\mathsf{e}\ \frac{\gamma \vdash_{\mathsf{e}} \sigma}{\gamma \vdash_{\mathsf{r}} \sigma} \qquad \mathsf{lam}\ \frac{(\sigma;\gamma) \vdash_{\mathsf{r}} \tau}{\gamma \vdash_{\mathsf{r}} \sigma \to \tau} \qquad \mathsf{pair}\ \frac{\gamma \vdash_{\mathsf{r}} \sigma \quad \gamma \vdash_{\mathsf{r}} \tau}{\gamma \vdash_{\mathsf{r}} \sigma * \tau}$$

$$\mathsf{zero}\ \frac{}{\gamma \vdash_{\mathsf{r}} \mathbb{N}} \qquad \mathsf{succ}\ \frac{\gamma \vdash_{\mathsf{r}} \mathbb{N}}{\gamma \vdash_{\mathsf{r}} \mathbb{N}}$$

and **neutral** terms ($\vdash_{\mathsf{e}}$),

$$\mathsf{hyp}\ \frac{}{(\sigma;\gamma) \vdash_{\mathsf{e}} \sigma} \qquad \mathsf{wkn}\ \frac{\gamma \vdash_{\mathsf{r}} \sigma}{(\tau;\gamma) \vdash_{\mathsf{e}} \sigma} \qquad \mathsf{app}\ \frac{\gamma \vdash_{\mathsf{e}} \sigma \to \tau \quad \gamma \vdash_{\mathsf{r}} \sigma}{\gamma \vdash_{\mathsf{e}} \tau}$$

$$\mathsf{fst}\ \frac{\gamma \vdash_{\mathsf{e}} \sigma * \tau}{\gamma \vdash_{\mathsf{e}} \sigma} \qquad \mathsf{snd}\ \frac{\gamma \vdash_{\mathsf{e}} \sigma * \tau}{\gamma \vdash_{\mathsf{e}} \tau} \qquad \mathsf{rec}\ \frac{\gamma \vdash_{\mathsf{e}} \mathbb{N} \quad \gamma \vdash_{\mathsf{r}} \sigma \quad \gamma \vdash_{\mathsf{r}} \mathbb{N} \to \sigma \to \sigma}{\gamma \vdash_{\mathsf{e}} \sigma}.$$

Normalization-by-evaluation for T^+

Normalization is proven using a constructive normalization-by-evaluation proof in continuation-passing style (CPS). System T^+ is evaluated into the following continuation monad:

$$\gamma \Vdash \sigma = \forall \gamma_1 \geq \gamma \, (\forall \gamma_2 \geq \gamma_1 \, (\gamma_2 \Vdash_{\mathsf{s}} \sigma \Rightarrow \gamma_2 \vdash_{\mathsf{r}} \mathbb{N}) \Rightarrow \gamma_1 \vdash_{\mathsf{r}} \mathbb{N})$$

$$\begin{aligned} \gamma \Vdash_{\mathsf{s}} \mathbb{N} &= \gamma \vdash_{\mathsf{r}} \mathbb{N} \\ \gamma \Vdash_{\mathsf{s}} (\sigma \to \tau) &= \forall \gamma' \geq \gamma (\gamma' \Vdash \sigma \Rightarrow \gamma' \Vdash \tau) \\ \gamma \Vdash_{\mathsf{s}} (\sigma * \tau) &= \gamma \Vdash \sigma \times \gamma \Vdash \tau \end{aligned}$$

where

$$\geq_{\text{refl}} \frac{}{\gamma \geq \gamma} \qquad \geq_{\text{cons}} \frac{\gamma_2 \geq \gamma_1}{(\sigma; \gamma_2) \geq \gamma_1}$$

The 'return' and 'run' operations:

$$\eta(-) : \gamma \Vdash_{\mathsf{s}} \sigma \Rightarrow \gamma \Vdash \sigma \qquad \mu(-) : \gamma \Vdash \mathbb{N} \Rightarrow \gamma \Vdash_{\mathsf{s}} \mathbb{N}$$
$$\eta H = \geq_1 \kappa \mapsto \kappa \geq_{\mathsf{refl}} \lfloor H \rfloor_{\geq_1} \qquad \mu H = H \geq_{\mathsf{refl}} (\geq_1 \alpha \mapsto \alpha)$$

Monotonicity properties:

$$\ulcorner - \urcorner(-) : \gamma_2 \geq \gamma_1 \Rightarrow \gamma_1 \vdash_{\mathsf{r}} \sigma \Rightarrow \gamma_2 \vdash_{\mathsf{r}} \sigma$$
$$\llcorner - \lrcorner(-) : \gamma_2 \geq \gamma_1 \Rightarrow \gamma_1 \vdash_{\mathsf{e}} \sigma \Rightarrow \gamma_2 \vdash_{\mathsf{e}} \sigma$$
$$\lceil - \rceil^{(-)} : \gamma_2 \geq \gamma_1 \Rightarrow \gamma_1 \Vdash \sigma \Rightarrow \gamma_2 \Vdash \sigma$$
$$\lfloor - \rfloor_{(-)} : \gamma_2 \geq \gamma_1 \Rightarrow \gamma_1 \Vdash_{\mathsf{s}} \sigma \Rightarrow \gamma_2 \Vdash_{\mathsf{s}} \sigma$$
$$[\![-]\!]^{(-)} : \gamma_2 \geq \gamma_1 \Rightarrow \gamma_1 \Vdash\Vdash \gamma \Rightarrow \gamma_2 \Vdash\Vdash \gamma$$

Normalization-by-evaluation for T^+

The *reify* function

$$
\begin{aligned}
&{}^{\gamma}\!\downarrow^{\sigma} (-) : \gamma \Vdash \sigma \Rightarrow \gamma \vdash_{\mathsf{r}} \sigma \\
&\quad {}^{\gamma}\!\downarrow^{\mathbb{N}} H = \mu H \\
&{}^{\gamma}\!\downarrow^{\sigma\to\tau} H = \mathsf{lam}({}^{\gamma}\!\downarrow^{\tau} \\
&\qquad\qquad (\geq_1 \kappa \mapsto \\
&\qquad\qquad\quad H(\geq_1 \cdot \geq_{\mathsf{cons}} \geq_{\mathsf{refl}}) \\
&\qquad\qquad\quad (\geq_2 \phi \mapsto \\
&\qquad\qquad\qquad \phi \geq_{\mathsf{refl}} (\lceil^{\sigma;\gamma}\!\uparrow^{\sigma} \mathit{hyp}\rceil^{\geq_2\cdot\geq_1}) \geq_{\mathsf{refl}} \\
&\qquad\qquad\qquad (\geq_3 \mapsto \kappa(\geq_3 \cdot \geq_2)))))) \\
&{}^{\gamma}\!\downarrow^{\sigma * \tau} H = \mathsf{pair} \\
&\qquad\qquad {}^{\alpha}\!\downarrow^{\sigma} (\geq_1 \kappa \mapsto \\
&\qquad\qquad\qquad H \geq_1 (\geq_2 \alpha \mapsto \mathsf{proj}_1\, \alpha \geq_{\mathsf{refl}} (\geq_3 \mapsto \kappa(\geq_3 \cdot \geq_2)))) \\
&\qquad\qquad {}^{\alpha}\!\downarrow^{\tau} (\geq_1 \kappa \mapsto \\
&\qquad\qquad\qquad H \geq_1 (\geq_2 \alpha \mapsto \mathsf{proj}_2\, \alpha \geq_{\mathsf{refl}} (\geq_3 \mapsto \kappa(\geq_3 \cdot \geq_2))))
\end{aligned}
$$

Normalization-by-evaluation for T^{+}

The *reify* function

$$
\begin{aligned}
{}^{\gamma}\!\downarrow^{\sigma} (-) &: \gamma \Vdash \sigma \Rightarrow \gamma \vdash_{\mathsf{r}} \sigma \\
{}^{\gamma}\!\downarrow^{\mathbb{N}} H &= \mu H \\
{}^{\gamma}\!\downarrow^{\sigma\to\tau} H &= \mathsf{lam}({}^{\gamma}\!\downarrow^{\tau} \\
&\qquad (\geq_1 \kappa \mapsto \\
&\qquad\quad H(\geq_1 \cdot \geq_{\mathsf{cons}} \geq_{\mathsf{refl}}) \\
&\qquad\quad (\geq_2 \phi \mapsto \\
&\qquad\qquad \phi \geq_{\mathsf{refl}} (\lceil {}^{\sigma;\gamma}\!\uparrow^{\sigma} \mathit{hyp} \rceil^{\geq_2 \cdot \geq_1}) \geq_{\mathsf{refl}} \\
&\qquad\qquad (\geq_3 \mapsto \kappa(\geq_3 \cdot \geq_2)))))) \\
{}^{\gamma}\!\downarrow^{\sigma * \tau} H &= \mathsf{pair} \\
&\qquad {}^{\alpha}\!\downarrow^{\sigma} (\geq_1 \kappa \mapsto \\
&\qquad\quad H \geq_1 (\geq_2 \alpha \mapsto \mathsf{proj}_1\, \alpha \geq_{\mathsf{refl}} (\geq_3 \mapsto \kappa(\geq_3 \cdot \geq_2)))) \\
&\qquad {}^{\alpha}\!\downarrow^{\tau} (\geq_1 \kappa \mapsto \\
&\qquad\quad H \geq_1 (\geq_2 \alpha \mapsto \mathsf{proj}_2\, \alpha \geq_{\mathsf{refl}} (\geq_3 \mapsto \kappa(\geq_3 \cdot \geq_2))))
\end{aligned}
$$

This function shows that we can actually run the monad at *any* type

Normalization-by-evaluation for T^+

$$
\begin{aligned}
{}^{\gamma}\!\uparrow^{\sigma}(-) &: \gamma \vDash_{\mathsf{e}} \sigma \Rightarrow \gamma \Vdash \sigma \\
{}^{\gamma}\!\uparrow^{\mathbb{N}} p &= \eta(\mathsf{e}\, p) \\
{}^{\gamma}\!\uparrow^{\sigma\to\tau} p &= \eta(\geq_2 \alpha \mapsto {}^{\gamma}\!\uparrow^{\tau} \mathsf{app}(\llcorner p \lrcorner_{\geq_2}, {}^{\gamma}\!\downarrow^{\sigma} \alpha)) \\
{}^{\gamma}\!\uparrow^{\sigma * \tau} p &= \eta({}^{\gamma}\!\uparrow^{\sigma} \mathsf{fst}\, p, {}^{\gamma}\!\uparrow^{\tau} \mathsf{snd}\, p)
\end{aligned}
$$

This function is needed *only* by the $\sigma \to \tau$-case of reify. Morally, it only performs η-expansion.

Normalization-by-evaluation for T^{+}

The *evaluation* function

$$
\begin{aligned}
{}^{\gamma}[\![-]\!]^{\sigma}_{(-)} : \gamma \vdash \sigma &\Rightarrow \forall \gamma' \Vdash \gamma (\gamma' \Vdash \sigma) \\
[\![\mathsf{hyp}]\!]_{\rho} &= \mathsf{proj}_1\ \rho \\
[\![\mathsf{wkn}\, p]\!]_{\rho} &= [\![p]\!]_{\mathsf{proj}_2\ \rho} \\
[\![\mathsf{lam}\, p]\!]_{\rho} &= \eta(\geq_1 \alpha \mapsto [\![p]\!]_{(\alpha, \lceil \rho \rceil^{\geq_1})}) \\
&\vdots \\
[\![\mathsf{pair}(p,q)]\!]_{\rho} &= \eta([\![p]\!]_{\rho}, [\![q]\!]_{\rho}) \\
[\![\mathsf{fst}\, p]\!]_{\rho} &= \geq_1 \kappa \mapsto [\![p]\!]_{\rho} \geq_1 (\geq_2 \alpha \mapsto \mathsf{proj}_1\ \alpha \geq_{\mathsf{refl}} (\geq_3 \mapsto \kappa(\geq_3 \cdot \geq_2))) \\
&\vdots \\
[\![\mathsf{shift}\, p]\!]_{\rho} &= \geq_1 \kappa \mapsto \mu [\![p]\!]_{\eta(\geq_3 \alpha \mapsto \eta(\alpha \geq_{\mathsf{refl}} (\geq_4 \mapsto \kappa(\geq_4 \cdot \geq_3)))), \lceil \rho \rceil^{\geq_1}} \\
&\vdots
\end{aligned}
$$

Normalization-by-evaluation for T^+

Theorem (Normalization)

*There is a normalization function $\downarrow [\![-]\!]$ s.t. for every term p of **System T^+** of type $\gamma \vdash \tau$, the term $\downarrow [\![p]\!]$ is a normal form of **System T** of the same type ($\gamma \vdash_{\mathsf{r}} \tau$).*

Proof.

Compose the defined functions:

$$
\begin{aligned}
&{}^{\gamma}[\![-]\!]^{\sigma}_{(-)} : \gamma \vdash \sigma \Rightarrow \forall \gamma' \Vdash \gamma (\gamma' \Vdash \sigma) \\
&{}^{\gamma}\uparrow^{\sigma} (-) : \gamma \vdash_{\mathsf{e}} \sigma \Rightarrow \gamma \Vdash \sigma \\
&{}^{\gamma}\downarrow^{\sigma} (-) : \gamma \Vdash \sigma \Rightarrow \gamma \vdash_{\mathsf{r}} \sigma
\end{aligned}
$$

□

Correctness of NBE

Theorem

The following definitional equalities hold,

$$\downarrow [\![\mathit{wkn}\, p]\!]_{\alpha,\rho} = \downarrow [\![p]\!]_{\rho} \tag{6}$$

$$\downarrow [\![\mathit{hyp}]\!]_{\alpha,\rho} = \downarrow \alpha \tag{7}$$

$$\downarrow [\![\mathit{fst}\, \mathit{pair}(p, q)]\!]_{\rho} = \downarrow [\![p]\!]_{\rho} \tag{8}$$

$$\downarrow [\![\mathit{snd}\, \mathit{pair}(p, q)]\!]_{\rho} = \downarrow [\![q]\!]_{\rho} \tag{9}$$

$$\downarrow [\![\mathit{app}(\mathit{lam}\, p, q)]\!]_{\rho} = \downarrow [\![p]\!]_{[\![q]\!]_{\rho},\rho} \tag{10}$$

$$\downarrow [\![\mathit{rec}(\mathit{zero}, p, q)]\!]_{\rho} = \downarrow [\![p]\!]_{\rho} \tag{11}$$

$$\downarrow [\![\mathit{rec}(\mathit{succ}\, r, p, q)]\!]_{\rho} = \downarrow [\![\mathit{app}(\mathit{app}(q, r), \mathit{rec}(r, p, q))]\!]_{\rho} \tag{12}$$

$$\downarrow^{\mathbb{N}} [\![\mathit{shift}\, p]\!]_{\rho} = \downarrow^{\mathbb{N}} [\![p]\!]_{\phi,\rho} \tag{13}$$

$$\downarrow^{\mathbb{N}} [\![\mathit{app}(\mathit{app}(\mathit{hyp}, x), y)]\!]_{\phi,\rho} = \downarrow^{\mathbb{N}} [\![y]\!]_{\phi,\rho} \tag{14}$$

where for the last two equations,

$$\phi := \eta(\geq_3 \alpha \mapsto \eta(\mu\alpha)).$$

Correctness of NBE

Proof of the theorem.

Equations (6)–(12) follow from the ones that hold already of the $[\![-]\!]_{(-)}$ function. Equations (13)–(14) also follow by definition, this time reification being applied for only one concrete type, $\mathbb{N}$. □

Correctness of NBE

Proof of the theorem.

Equations (6)–(12) follow from the ones that hold already of the $[\![-]\!]_{(-)}$ function. Equations (13)–(14) also follow by definition, this time reification being applied for only one concrete type, $\mathbb{N}$. □

This is easy to say but difficult to prove: one needs to find the right formulation of defining equations for $[\![-]\!]$.

Links

- *Axioms and Decidability for Type Isomorphism in Presence of Sums*, in Proceedings of CSL-LICS 2014
- *The Exp-Log Normal Form of Types and Canonical Terms for Lambda Calculus with Sums* (with Zakaria Chihani), submitted
  `https://github.com/dankoi/metamath/tree/master/NBE-at-ENF`
- (*An interpretation of the Sigma-2 fragment of classical Analysis in System T*, submitted)
  `https://github.com/dankoi/metamath/tree/master/shift-analysis`
- *Type Directed Partial Evaluation for Level-1 Shift and Reset*, in Electronic Proceedings in Theoretical Computer Science 127, 2013
  `https://github.com/dankoi/metamath/tree/master/coq/nbe_shift_1`