Normalization by Yoneda embedding

Djordje Čubrić, Cambridge
Peter Dybjer, Göteborg
Phil Scott, Ottawa

August 16, 1996

What is a normalization proof?

Traditional approach: prove about *reduction*

- **(weak) normalization - existence**
- **Church-Rosser - uniqueness**

Reduction-free approach: prove about *conversion* ($\sim$): there is an algorithm nf such that

- $t \sim nf\ t$
- $t \sim t' \supset nf\ t = nf\ t'$

Corollary - solution of the word problem:

- $t \sim t' \leftrightarrow nf\ t = nf\ t'$

Normalization by intuitionistic representation theorem

Syntax is free model $(T, \sim)$ (classically, $T/\sim$).

Find "strict" model M with (left) inverse of unique interpretation map:

$$(T, \sim) \underset{[\![-]\!]^{-1}}{\overset{[\![-]\!]}{\rightleftarrows}} (M, =)$$

that is

$$nf\ t = [\![[\![t]\!]]\!]^{-1} \sim t$$

$$t \sim t' \supset [\![t]\!] = [\![t']\!] \supset nf\ t = nf\ t'$$

Intuitionistic framework (Martin-Löf Type Theory, etc): function = algorithm!

Normalization by Yoneda embedding?

The functor $\mathcal{Y} : \mathcal{C} \to \mathcal{S}et^{\mathcal{C}^{op}}$

$$\begin{aligned} \mathcal{Y}B &= \mathcal{C}(-, B) \\ \mathcal{Y}g &= g \circ - \end{aligned}$$

induces a bijection of hom-sets:

$$\mathcal{C}(A, B) \underset{-_A\ id_A}{\overset{\mathcal{Y}}{\rightleftarrows}} \mathcal{S}et^{\mathcal{C}^{op}}(\mathcal{Y}A, \mathcal{Y}B)$$

Monoids - one object categories:

$$M \underset{-\ id}{\overset{\mathcal{Y}}{\rightleftarrows}} \{\phi \in M^M | \phi \textbf{ natural}\}$$

Groups - Cayley's representation theorem:

$$G \underset{-\ id}{\overset{\mathcal{Y}}{\rightleftarrows}} \{\phi \in G^G | \phi \textbf{ natural iso}\}$$

Constructive Yoneda: functor = algorithm. But

$$\mathcal{C}(A,B) \underset{-_A\; id_A}{\overset{\mathcal{Y}}{\rightleftarrows}} \mathcal{S}et^{\mathcal{C}^{op}}(\mathcal{Y}A,\mathcal{Y}B)$$

only maps $g \in \mathcal{C}(A,B)$ **to** $g \circ id_A$**!**

Syntax = free category (monoid) $\mathcal{T}/\sim$**. Unique interpretation functor:**

$$\mathcal{T}/\sim \xrightarrow{[\![-]\!]} \mathcal{C}$$

For presheaves, if $[\![-]\!] = \mathcal{Y}$ **for atoms, then**

$$(\mathcal{T}/\sim)(A,B) \underset{-_A\; id_A}{\overset{[\![-]\!]=\mathcal{Y}}{\rightleftarrows}} \mathcal{S}et^{(\mathcal{T}/\sim)^{op}}(\mathcal{Y}A,\mathcal{Y}B)$$

and now $nf\ g = [\![g]\!]\ id_A$**!!**

(For cccs, unique means up to isomorphism!)

Plan

1. **Earlier work**

2. **Constructive algebra: P-sets and P-monoids**

3. **The word problem for monoids**

4. **Constructive category theory: P-categories, P-Yoneda for P-cccs**

5. **The word problem for cccs**

6. **Conclusion**

Earlier work

- **Martin-Löf (1973, 1974): normalization by intuitionistic model construction (combinators and weak type theory).**

- **Berger and Schwichtenberg (1991): normalization for $\lambda\beta\eta$ by inverting set-theoretic interpretation; Friedman's theorem.**

- **T. Coquand and Dybjer (1993): footnote to Martin-Löf (1973, 1974) (algebraic aspects).**

- **C. Coquand (1993): normalization for $\lambda\beta\eta$ by inverting Kripke interpretation.**

- **Altenkirch, Hofmann, and Streicher (1995): normalization for $\lambda\beta\eta$ by Yoneda and glueing.**

Sets with equality

Bishop's distinction between *preset* and *set*.

An *E-set* (*setoid*) is a set A with an equivalence relation $\sim$.

An *E-map* from $(A, \sim)$ to $(A', \sim')$ is a function from A to A' which preserves equivalence.

Want better separation of "algorithmic" and "logical" properties - P-sets encode both "subsets" and "quotients":

A *P-set* is a set A with a per $\sim$.

A *P-map* from $(A, \sim)$ to $(A', \sim')$ is a function from A to A' which preserves pers.

P-monoid

- **a P-set $(M, \sim)$;**

- **a binary P-map $\circ$ on $(M, \sim)$;**

- **an element id in M;**

- **such that**

$$\begin{aligned} (\theta \circ \delta) \circ \gamma &\sim \theta \circ (\delta \circ \gamma) \\ id \circ \gamma &\sim \gamma \\ \gamma \circ id &\sim \gamma \end{aligned}$$

for θ in the domain of $\sim$, that is, $\theta \sim \theta$, etc.

The word problem for monoids

T is the set of binary trees generated by a set X of atoms x.

$$t ::= t \circ t \mid id \mid x$$

$\sim$ is the congruence relation between elements of T generated by identity and associativity laws.

$(T, \sim)$ is a P-free P-monoid generated by X (and an E-free E-monoid!).

Decide $t \sim t'$!

Use the constructive P-iso

$$(T, \sim) \underset{-\ id}{\overset{[\![-]\!]}{\rightleftarrows}} (T^T, \sim)$$

analogue of

$$T/\sim \underset{-\ id}{\overset{[\![-]\!]}{\rightleftarrows}} \{\phi \in (T/\sim)^{T/\sim} | \phi \text{ natural}\}$$

Hence $nf\ t = [\![t]\!]\ id \sim t$

But we also have

$$(T, \sim) \xrightarrow{[\![-]\!]} (T^T, \equiv)$$

where $\phi \equiv \phi'$ **iff the *underlying* functions are extensionally equal and natural! Hence, if** $t \sim t'$**, then** $[\![t]\!] \equiv [\![t']\!]$ **and** $nf\ t = nf\ t'$**.**

Strict notions

$(T^T, \equiv)$ is a *strict* monoid: if $\phi \equiv \phi'$ then $\phi = \phi'$.

$(T, \sim)$ and $(T^T, \sim)$ are *non-strict*.

Suggestive terminology?

$\sim$	$\equiv$
non-strict	**strict**
abstract	**concrete**
syntactic	**semantic**
formal	**real**
static	**dynamic**

Compare category theory: $\cong$ vs $=$!

The word problem for groups?

G is the set of "group-expressions":

$$t ::= t \circ t \mid id \mid t^{-1} \mid x$$

$\sim$ is the congruence relation so that $(G, \sim)$ is a P-free P-group.

Try $nf\ t = [\![t]\!]\ id \sim t$! Still get $nf\ t \sim t$.

But we do not have $t \sim t'$ implies $nf\ t = nf\ t'$. Because

$$[\![x]\!] = \mathcal{Y}\ x = x \circ -$$

is only a "formal" $\sim$-iso but not a "real" $=$-iso!

P-categories

- **A *set* of objects;**
- **hom-*P-sets*;**
- **composition is a *P-map*;**
- **the category axioms refer to $\sim$.**

***Object equality*? Not part of the definition of P-category, but objects form a P-set (and E-set) under P-isomorphism.**

The P-category analogue $\mathcal{PS}et$ of the ordinary category of sets

- **P-sets as objects**
- $\mathcal{PS}et((A, \sim_A), (B, \sim_B)) = (B^A, \sim_{B^A})$, **where** $\phi \sim_{B^A} \phi'$ **iff** $a \sim_A a'$ **implies** $\phi\ a \sim_B \phi'\ a'$.

P-functor, P-natural transformations, P-functor P-category, P-presheaf, P-ccc, P-free, ...

Yoneda and cccs

- **The Yoneda functor $\mathcal{Y} : \mathcal{C} \to \mathcal{S}et^{\mathcal{C}^{op}}$ preserves ccc-structure.**

  If $\mathcal{T}$ is a free ccc then there is a natural isomorphism

$$\mathcal{T} \;\underset{\mathcal{Y}}{\overset{[\![-]\!]}{\rightrightarrows}}\; \mathcal{S}et^{\mathcal{T}^{op}} \qquad q \Downarrow\Uparrow q^{-1}$$

  where $[\![-]\!] = \mathcal{Y}$ on atoms.

  P-version gives normal form algorithm!

- **$\mathcal{S}et^{\mathcal{C}^{op}}$ is a ccc for any category $\mathcal{C}$.**

  P-version helps proving uniqueness of normal forms!

Normal form algorithm for cccs

$$
\begin{array}{ccc}
\mathcal{T}(A,B) & \xrightarrow{[\![-]\!]} & \mathcal{PS}et^{\mathcal{T}^{op}}([\![A]\!],[\![B]\!]) \\
 & \overset{\mathcal{Y}}{\searrow}\ \underset{-_A\ id_A}{\nwarrow} & \downarrow q_B \circ - \circ q_A^{-1} \\
 & & \mathcal{PS}et^{\mathcal{T}^{op}}(\mathcal{Y}\ A, \mathcal{Y}\ B)
\end{array}
$$

is a commuting diagram of P-sets and P-maps. Hence

$$nf\ t = q_{BA}\ ([\![t]\!]\ (q_{AA}^{-1}\ id_A)) \sim t$$

$$
\begin{aligned}
[\![g \circ f]\!]_C\ a &= [\![g]\!]_C([\![f]\!]_C\ a)\\
[\![\mathtt{id}]\!]_C\ a &= a\\
[\![\,!\,]\!]_C\ a &= ()\\
[\![<f, g>]\!]_C\ a &= ([\![f]\!]_C\ a, [\![g]\!]_C\ a)\\
[\![\pi]\!]_C\ (a, a') &= a\\
[\![\pi']\!]_C\ (a, a') &= a'\\
([\![f^*]\!]_C\ a)_D\ (g, a') &= [\![f]\!]_D\ ([\![A]\!]\ g\ a, a')\\
[\![\varepsilon]\!]_C\ (\theta, a) &= \theta_C\ (\mathtt{id}, a)
\end{aligned}
$$

$$
\begin{array}{rcl}
q_{X,C}\ f & = & f \\
q_{1,C}\ () & = & ! \\
q_{A\times B,C}\ (a,b) & = & <q_{A,C}\ a, q_{B,C}\ b> \\
q_{A\Rightarrow B,C}\ \theta & = & (q_{B,C\times A}\ (\theta_{C\times A}\ (\pi_{C,A}, q^{-1}_{A,C\times}
\end{array}
$$

$$
\begin{array}{rcl}
q^{-1}_{X,C}\ f & = & f \\
q^{-1}_{1,C}\ f & = & () \\
q^{-1}_{A\times B,C}\ f & = & (q^{-1}_{A,C}\ (\pi_{A,B}\circ f), q^{-1}_{B,C}\ (\pi'_{A,B}\circ \\
(q^{-1}_{A\Rightarrow B,C}\ f)_D\ g\ x & = & q^{-1}_{B,D}\ (\varepsilon_{A,B}\circ <f\circ g, q_{A,D}\ x>)
\end{array}
$$

Uniqueness of normal forms?

What about

$$t \sim t' \supset nf\ t = nf\ t'?$$

It depends on *what* P-free P-ccc we choose!

If $\mathcal{T}$ is built up by ccc-expressions (categorical combinators) under the congruence generated by the ccc-laws, then No!

If $\mathcal{T}$ is built up from the typed $\lambda\beta\eta$-calculus, then Yes - nf will return the η-long normal form of a term.

$\mathcal{T}_{\beta\eta}$ - a P-free P-ccc from the typed $\lambda\beta\eta$-calculus

objects: sequences of types $\Gamma = (A_1, \ldots, A_m)$

arrows: sequences of terms

$$(A_1, \ldots, A_m) \xrightarrow{(t_1, \ldots, t_n)} (B_1, \ldots, B_n)$$

equivalence of arrows: pointwise $\beta\eta$-convertibility

P-ccc structure:

$$\begin{aligned} 1 &= () \\ \Gamma \times \Delta &= \Gamma, \Delta \\ (B_1, \ldots, B_m)^\Gamma &= (B_1^\Gamma, \ldots, B_m^\Gamma) \end{aligned}$$

where

$$B^{(A_1, \ldots, A_m)} = A_1 \to \cdots \to A_m \to B$$

$\mathcal{T}_\alpha$ - the α-congruence P-category

$\mathcal{T}_\alpha$ is the P-category of sequences of λ-terms under α-congruence $\equiv$.

$\mathcal{T}_{\beta\eta}$ and $\mathcal{T}_\alpha$ have the same data part!

$\mathcal{PS}et^{\mathcal{T}_\alpha^{op}}$ is a P-ccc. Hence $t \sim t'$ implies $[\![t]\!] \equiv [\![t']\!]$.

To prove that this entails $nf\ t \equiv nf\ t'$ it remains to prove that $q_{B,A}$ and $q_{B,A}^{-1}$ preserve $\equiv$. This uses that $\mathcal{T}_\alpha$ is a P-cartesian P-category which also satisfies the ccc-law which corresponds to substitution under λ:

$$t^* \circ u \equiv (t \circ \langle u \circ \pi, \pi' \rangle)^*$$

More related work

- **Categorical coherence proofs: Lafont (1988), Power (1987), Beylin and Dybjer (1995).**

- **Computational category theory:**

  Burstall and Rydeheard (1988): category theory in ML.

  Aczel (1993), Huet and Saibi (1995): E-category theory in Lego and Coq.

- **Extracting programs from intuitionistic proofs. Various methods including realizability models, Berger (1993).**

- **Other calculi: system F; linear λ-calculus; dependent types; ...?**