Revisiting the model of Hereditary Recursive Operations

Félix Castro

September 26, 2024

Context

This talk was presented in the workshop "Meta-programming, quoting, and modalities in type theory".

This is part of my work around intensional and extensional equality in $\mathbf{HA}^\omega$.

We will see how functionals with behaviors closed to the one of `quote` naturally exist in a model of $\mathbf{HA}^\omega$.

Intuition

The instruction `quote` takes a program as input and returns its source code.

Motivation

Understanding how `quote` should behave in a typed λ-calculus: how to extend System T with such an instruction? Links with Church Thesis?

Table of Contents

What is System T?

System T is

1. a programming language that can describe (some) "functionals" (i.e functions over natural numbers, functions over functions over natural numbers...)
2. expressed as a typed lambda calculus
3. a theoretical tool invented by Gödel (in 1958) to give a computational interpretation of arithmetic with his so-called Dialectica interpretation

Syntax of System T

System T is obtained by extending simply typed λ-calculus with a base type $\mathbf{N}$ and native constructors to use it:

$$
\begin{array}{llcll}
\text{Sorts} & \sigma, \tau & ::= & \mathbf{N} \mid \sigma \to \tau & (\to \text{ right associative}) \\
\text{Terms} & t, u & ::= & x^\sigma \mid \lambda x^\sigma . t \mid (tu) & (tu \text{ left associative}) \\
 & & & \mid 0 \mid \mathrm{S}\, t \mid \mathrm{Rec}^\sigma\, t\, u\, v & \\
\text{Signatures} & \Delta & ::= & \emptyset \mid \Delta, x^\sigma & (\text{technicality})
\end{array}
$$

Examples

$$
\begin{array}{lclcl}
\mathrm{add} & \equiv & \lambda x^{\mathbf{N}} \lambda y^{\mathbf{N}} . \mathrm{Rec}^{\mathbf{N}}\, x\, (\lambda h^{\mathbf{N}} \lambda m^{\mathbf{N}} . \mathrm{S}\, h)\, y & : & \mathbf{N} \to (\mathbf{N} \to \mathbf{N}) \\
\mathrm{app}_{\sigma,\tau} & \equiv & \lambda f^{\sigma \to \tau} \lambda x^\sigma . (f\, x) & : & (\sigma \to \tau) \to \sigma \to \tau
\end{array}
$$

I may omit sort annotations when writing terms.

Derivation in System T

$$\dfrac{}{\Delta_1, x^\sigma, \Delta_2 \vdash_{\mathrm{T}} x^\sigma : \sigma}\ (\mathsf{ax})$$

$$\dfrac{\Delta, x^\sigma \vdash_{\mathrm{T}} t : \tau}{\Delta \vdash_{\mathrm{T}} \lambda x^\sigma . t : \sigma \to \tau}\ (\lambda\text{-intro}) \qquad \dfrac{\Delta \vdash_{\mathrm{T}} t : \sigma \to \tau \quad \Delta \vdash_{\mathrm{T}} u : \sigma}{\Delta \vdash_{\mathrm{T}} tu : \tau}\ (\mathsf{app})$$

$$\dfrac{}{\Delta \vdash_{\mathrm{T}} 0 : \mathbf{N}}\ (0) \qquad \dfrac{\Delta \vdash_{\mathrm{T}} t : \mathbf{N}}{\Delta \vdash_{\mathrm{T}} \mathrm{S}\, t : \mathbf{N}}\ (\mathrm{S})$$

$$\dfrac{\Delta \vdash_{\mathrm{T}} t : \sigma \quad \Delta \vdash_{\mathrm{T}} u : \sigma \to \mathbf{N} \to \sigma \quad \Delta \vdash_{\mathrm{T}} v : \mathbf{N}}{\Delta \vdash_{\mathrm{T}} \mathrm{Rec}^\sigma\ t\, u\, v : \sigma}\ (\mathrm{Rec})$$

Computation in System T

We consider the following rules on terms

$$\begin{array}{rcl} (\lambda x.t)\,u & \succ & t[x ::= u] \\ \mathrm{Rec}\,t\ \,u\,0 & \succ & t \\ \mathrm{Rec}\,t\ \,u\,(\mathrm{S}\,v) & \succ & u\,(\mathrm{Rec}\,t\,u\,v)\,v \end{array}$$

from which we generate reduction[1]

$$t \quad \twoheadrightarrow \quad u$$

and congruence[2]

$$t \quad \cong \quad u$$

[1] least compatible, reflexive and transitive relation containing $\succ$

[2] least equivalence relation containing $\twoheadrightarrow$.

Metatheoretical results

1. Canonicity : closed normal terms of type **N** are of the form $\mathrm{S}^n\,0$, closed normal terms of type $\sigma \to \tau$ are of the form $\lambda x^\sigma . t$.
2. Strong normalisation : terms of System T are strongly normalisable.
3. Representable functions : a function $f : \mathbf{N} \to \mathbf{N}$ can be (extensionally) expressed as a term of System T if and only if it is a recursive function provably total in **PA**(Peano Arithmetic).

What is $\mathbf{HA}^\omega$?

$\mathbf{HA}^\omega$ is

1. a first order many sorted (intuitionistic) theory
2. a formal theory to speak about functionals
3. a conservative extension of Heyting Arithmetic (a.k.a the intuitionistic fragment of Peano Arithmetic)

Formulas of $\mathbf{HA}^\omega$

Formulas of $\mathbf{HA}^\omega$ are generated as follow:

$$\begin{array}{rcl} \Phi, \Psi & ::= & t =_{\mathbf{N}} u \mid \bot \\ & & \mid \Phi \Rightarrow \Psi \mid \Phi \wedge \Psi \\ & & \mid \forall x^\sigma . \Phi \mid \exists x^\sigma . \Phi \end{array}$$

where equality is restricted to the sort $\mathbf{N}$.

The connectives $\top, \neg$ and $\vee$ are not included in the syntax of $\mathbf{HA}^\omega$ but they can be defined as

$$\begin{array}{rcl} \top & \equiv & \bot \Rightarrow \bot \\ \neg\Phi & \equiv & \Phi \Rightarrow \bot \\ \Phi \vee \Psi & \equiv & \exists x^{\mathbf{N}} (x =_{\mathbf{N}} 0 \Rightarrow \Phi \wedge x \neq_{\mathbf{N}} 0 \Rightarrow \Psi) \end{array}$$

Axioms and rules of $\mathbf{HA}^\omega$

The axioms and rules of $\mathbf{HA}^\omega$ are the universal closures of:

1. the axioms and rules for many-sorted intuitionistic predicate logic
2. the axioms of equality (roughly an axiomatization of $\cong$) [1]
3. for all formulas Φ:
$$x =_{\mathbf{N}} y \Rightarrow \Phi[z := x] \Rightarrow \Phi[z := y]$$
(equal terms should satisfy the same properties)
4. $\neg(\mathrm{S}\,x) =_{\mathbf{N}} 0$ (successors are different from 0)
5. for all formulas Φ:
$$\Phi[z := 0] \Rightarrow (\forall x^{\mathbf{N}}\, \Phi[z := x] \Rightarrow \Phi[z := \mathrm{S}\,x]) \Rightarrow \forall x^{\mathbf{N}}\, \Phi[z := x]$$
(a scheme of induction over natural numbers)

[1] because equality is restricted to the sort $\mathbf{N}$, one needs to fully apply terms to axiomatize $\cong$.

Axioms and rules of $\mathbf{HA}^\omega$

The axioms and rules of $\mathbf{HA}^\omega$ are the universal closures of:

1. the axioms and rules for many-sorted intuitionistic predicate logic
2. the axioms of equality (roughly an axiomatization of $\cong$) [1]
3. for all formulas Φ:
$$x =_{\mathbf{N}} y \Rightarrow \Phi[z := x] \Rightarrow \Phi[z := y]$$
(equal terms should satisfy the same properties)
4. $\neg(\mathrm{S}\, x) =_{\mathbf{N}} 0$ (successors are different from 0)
5. for all formulas Φ:
$$\Phi[z := 0] \Rightarrow (\forall x^{\mathbf{N}}\, \Phi[z := x] \Rightarrow \Phi[z := \mathrm{S}\, x]) \Rightarrow \forall x^{\mathbf{N}}\, \Phi[z := x]$$
(a scheme of induction over natural numbers)

This theory can be formulated as a proof system.

[1] because equality is restricted to the sort $\mathbf{N}$, one needs to fully apply terms to axiomatize $\cong$.

The proof system $\lambda\mathbf{HA}^\omega$ (1/3): syntax

Formulas, proof terms and contexts of $\lambda\mathbf{HA}^\omega$ are generated by the following grammar:

$$
\begin{array}{llcl}
\text{Formulas} & \Phi, \Psi & ::= & t =_{\mathbf{N}} u \mid \bot \mid \mathrm{null}(t) \\
 & & & \mid \Phi \Rightarrow \Psi \mid \Phi \wedge \Psi \\
 & & & \mid \forall x^\sigma \Phi \mid \exists x^\sigma \Phi \\
\text{Proof terms} & M, N & ::= & \xi \mid \mathrm{refl}_{\mathbf{N}}\, t \mid \mathrm{peel}_{\mathbf{N}}(t, u, M, \hat{x}.\Phi, N) \\
 & & & \mid \mathrm{efq}(M, \Phi) \\
 & & & \mid \lambda\xi.M \mid M\, N \\
 & & & \mid (M, N) \mid M.1 \mid M.2 \\
 & & & \mid \lambda x^\sigma.M \mid M\, t \\
 & & & \mid [t, M] \mid \mathrm{let}\ [x, \xi] := M \ \mathrm{in}\ N \\
 & & & \mid \mathrm{Ind}(\hat{x}.\Phi, M, N, t) \\
\text{Contexts} & \Gamma & ::= & \emptyset \mid \Gamma, \xi : \Phi
\end{array}
$$

Sequents of $\lambda\mathbf{HA}^\omega$ have the shape

$$\Delta; \Gamma \vdash M : \Phi$$

The proof system $\lambda\mathbf{HA}^\omega$ (2/3): derivation

$$\frac{(\Delta;\Gamma)\ \mathbf{wf}}{\Delta;\Gamma \vdash \xi : \Phi}\ (\xi : \phi \in \Gamma) \qquad \frac{\Delta;\Gamma \vdash M : \bot}{\Delta;\Gamma \vdash \mathrm{efq}(M,\Phi) : \Phi}\ (\mathbf{FV}(\Phi) \subseteq \Delta) \qquad \frac{\Delta;\Gamma \vdash M : \Phi}{\Delta;\Gamma \vdash M : \Psi}\ (\Phi \simeq \Psi)$$

$$\frac{\Delta;\Gamma,\xi : \Phi \vdash M : \Psi}{\Delta;\Gamma \vdash \lambda\xi.M : \Phi \Rightarrow \Psi} \qquad \frac{\Delta;\Gamma \vdash M : \Phi \Rightarrow \Psi \quad \Delta;\Gamma \vdash N : \Phi}{\Delta;\Gamma \vdash M\,N : \Psi}$$

$$\frac{\Delta;\Gamma \vdash M_1 : \Phi_1 \quad \Delta;\Gamma \vdash M_2 : \Phi_2}{\Delta;\Gamma \vdash (M_1, M_2) : \Phi_1 \wedge \Phi_2} \qquad \frac{\Delta;\Gamma \vdash M : \Phi_1 \wedge \Phi_2}{\Delta;\Gamma \vdash M.i : \Phi_i}\ (i = 1,2)$$

$$\frac{\Delta, x^\sigma;\Gamma \vdash M : \Phi}{\Delta;\Gamma \vdash \lambda x^\sigma.M : \forall x^\sigma\Phi}\ (x^\sigma \notin \mathbf{FV}(\Gamma)) \qquad \frac{\Delta;\Gamma \vdash M : \forall x^\sigma\Phi \quad \Delta \vdash_{\mathrm{T}} t : \sigma}{\Delta;\Gamma \vdash M\,t : \Phi[x^\sigma := t]}$$

$$\frac{\Delta;\Gamma \vdash M : \Phi[x^\sigma := t] \quad \Delta \vdash_{\mathrm{T}} t : \sigma}{\Delta;\Gamma \vdash [t, M] : \exists x^\sigma\Phi} \qquad \frac{\Delta;\Gamma \vdash M : \exists x^\sigma\Phi \quad \Delta, x^\sigma;\Gamma,\xi : \Phi \vdash N : \Psi}{\Delta;\Gamma \vdash \mathrm{let}\ [x,\xi] := M\ \mathrm{in}\ N : \Psi}\ (x^\sigma \notin \mathbf{FV}(\Gamma,\Psi))$$

$$\frac{(\Delta;\Gamma)\ \mathbf{wf} \quad \Delta \vdash_{\mathrm{T}} t : \mathbf{N}}{\Delta;\Gamma \vdash \mathrm{refl}_{\mathbf{N}}\ t : t = t} \qquad \frac{\Delta;\Gamma \vdash M : t =_{\mathbf{N}} u \quad \Delta;\Gamma \vdash N : \Phi[x^{\mathbf{N}} := t]}{\Delta;\Gamma \vdash \mathrm{peel}_{\mathbf{N}}(t, u, M, \hat{x}.\Phi, N) : \Phi[x^{\mathbf{N}} := u]}$$

$$\frac{\Delta;\Gamma \vdash M : \Phi[x^{\mathbf{N}} := 0] \quad \Delta;\Gamma \vdash N : \forall x^{\mathbf{N}}.(\Phi \Rightarrow \Phi[x^{\mathbf{N}} := \mathrm{S}\,x^{\mathbf{N}}]) \quad \Delta \vdash_{\mathrm{T}} t : \mathbf{N}}{\Delta;\Gamma \vdash \mathrm{Ind}(\hat{x}.\Phi, M, N, t) : \Phi[x := t]}$$

The proof system $\lambda\mathbf{HA}^\omega$ (2/3): derivation

$$\frac{(\Delta;\Gamma)\ \mathbf{wf}}{\Delta;\Gamma\vdash \xi : \Phi}\ (\xi:\phi\in\Gamma)\qquad \frac{\Delta;\Gamma\vdash M:\bot}{\Delta;\Gamma\vdash \mathrm{efq}(M,\Phi):\Phi}\ (\mathbf{FV}(\Phi)\subseteq\Delta)\qquad \frac{\Delta;\Gamma\vdash M:\Phi}{\Delta;\Gamma\vdash M:\Psi}\ (\Phi\simeq\Psi)$$

$$\frac{\Delta;\Gamma,\xi:\Phi\vdash M:\Psi}{\Delta;\Gamma\vdash \lambda\xi.M:\Phi\Rightarrow\Psi}\qquad \frac{\Delta;\Gamma\vdash M:\Phi\Rightarrow\Psi\quad \Delta;\Gamma\vdash N:\Phi}{\Delta;\Gamma\vdash M\,N:\Psi}$$

$$\frac{\Delta;\Gamma\vdash M_1:\Phi_1\quad \Delta;\Gamma\vdash M_2:\Phi_2}{\Delta;\Gamma\vdash (M_1,M_2):\Phi_1\wedge\Phi_2}\qquad \frac{\Delta;\Gamma\vdash M:\Phi_1\wedge\Phi_2}{\Delta;\Gamma\vdash M.i:\Phi_i}\ (i=1,2)$$

$$\frac{\Delta,x^\sigma;\Gamma\vdash M:\Phi}{\Delta;\Gamma\vdash \lambda x^\sigma.M:\forall x^\sigma\Phi}\ (x^\sigma\notin\mathbf{FV}(\Gamma))\qquad \frac{\Delta;\Gamma\vdash M:\forall x^\sigma\Phi\quad \Delta\vdash_{\mathrm{T}} t:\sigma}{\Delta;\Gamma\vdash M\,t:\Phi[x^\sigma:=t]}$$

$$\frac{\Delta;\Gamma\vdash M:\Phi[x^\sigma:=t]\quad \Delta\vdash_{\mathrm{T}} t:\sigma}{\Delta;\Gamma\vdash [t,M]:\exists x^\sigma\Phi}\qquad \frac{\Delta;\Gamma\vdash M:\exists x^\sigma\Phi\quad \Delta,x^\sigma;\Gamma,\xi:\Phi\vdash N:\Psi}{\Delta;\Gamma\vdash \mathrm{let}\ [x,\xi]:=M\ \mathrm{in}\ N:\Psi}\ (x^\sigma\notin\mathbf{FV}(\Gamma,\Psi))$$

$$\frac{(\Delta;\Gamma)\ \mathbf{wf}\quad \Delta\vdash_{\mathrm{T}} t:\mathbf{N}}{\Delta;\Gamma\vdash \mathrm{refl}_{\mathbf{N}}\,t:t=t}\qquad \frac{\Delta;\Gamma\vdash M:t=_{\mathbf{N}}u\quad \Delta;\Gamma\vdash N:\Phi[x^{\mathbf{N}}:=t]}{\Delta;\Gamma\vdash \mathrm{peel}_{\mathbf{N}}(t,u,M,\hat{x}.\Phi,N):\Phi[x^{\mathbf{N}}:=u]}$$

$$\frac{\Delta;\Gamma\vdash M:\Phi[x^{\mathbf{N}}:=0]\quad \Delta;\Gamma\vdash N:\forall x^{\mathbf{N}}.(\Phi\Rightarrow\Phi[x^{\mathbf{N}}:=\mathrm{S}\,x^{\mathbf{N}}])\quad \Delta\vdash_{\mathrm{T}} t:\mathbf{N}}{\Delta;\Gamma\vdash \mathrm{Ind}(\hat{x}.\Phi,M,N,t):\Phi[x:=t]}$$

Theorem

$\vdash_{\mathbf{HA}^\omega}\Phi$ *if and only if there exists some proof term* M *such that* $\vdash M:\Phi$.

The proof system $\lambda\mathbf{HA}^\omega$ (3/3): final technical details

The congruence relation

$$\Phi \simeq \Psi$$

between formulas is generated from the reduction rules of System T and the two extra rules:

$$\begin{array}{rcl} \mathrm{null}(0) & \succ & \top \\ \mathrm{null}(\mathrm{S}\,x) & \succ & \bot \end{array}$$

The predicate $\mathrm{null}(x)$ is used to prove that successors are different from 0.

A pair of a signature and a context $(\Delta; \Gamma)$ is well formed if the free first-order variables of Γ are contained in Δ, i.e.

$$(\Delta; \Gamma)\ \mathbf{wf} \quad \equiv \quad \mathbf{FV}(\Gamma) \quad \subseteq \quad \Delta$$

As a consequence:

- if $\Delta; \Gamma \vdash M : \Phi$ then $\mathbf{FV}(\Phi) \subseteq \Delta$

Extensions of $\mathbf{HA}^\omega$

$$\begin{tikzcd}
& \mathbf{HA}^\omega \arrow[d, "\text{Extending equality}"] & \\
& \mathbf{N\text{-}HA}^\omega \arrow[dl, "\text{ext}"'] \arrow[dr, "\text{dec}"] & \\
\mathbf{E\text{-}HA}^\omega & & \mathbf{I\text{-}HA}^\omega
\end{tikzcd}$$

- $\mathbf{N\text{-}HA}^\omega$: adding equality at all level of sorts
- $\mathbf{E\text{-}HA}^\omega$: adding extensional equality at all level of sorts
- $\mathbf{I\text{-}HA}^\omega$: adding decidable equality at all level of sorts
  - $\mathtt{eqdec}_\sigma : \sigma \to \sigma \to \mathbf{N}$ such that $\forall x^\sigma \forall y^\sigma (x =_\sigma y \Leftrightarrow \mathtt{eqdec}_\sigma\, x\, y =_{\mathbf{N}} 0)$

[Anne S. Troelstra - Metamathematical Investigation of Intuitionistic Arithmetic and Analysis]

Models of $\mathbf{HA}^\omega$

A model of $\mathbf{HA}^\omega$ is given by the data of:

1. a family of sets $\{M_\sigma\}_\sigma$ indexed by the sorts of System T
2. an interpretation of all the constructions of System T inside these sets
3. such that this structure satisfies the axioms of $\mathbf{HA}^\omega$.

Models of $\mathbf{HA}^\omega$

A model of $\mathbf{HA}^\omega$ is given by the data of:

1. a family of sets $\{M_\sigma\}_\sigma$ indexed by the sorts of System T
2. an interpretation of all the constructions of System T inside these sets
3. such that this structure satisfies the axioms of $\mathbf{HA}^\omega$.

Spoiler

In this talk, we will try to see a model as a syntactic translation.

A model of $\mathbf{E\text{-}HA}^\omega$: the set-theoretic model

The set-theoretic model $\mathbf{M}$ is defined in $\mathbf{ZF}$ by

$$\begin{array}{rcl} \mathbf{M}_{\mathbf{N}} & \equiv & \mathbb{N} \\ \mathbf{M}_{\sigma\to\tau} & \equiv & \mathbf{M}_{\tau}^{\mathbf{M}_\sigma} \end{array}$$

where terms of System T are interpreted as functions.

A model of $\mathbf{I\text{-}HA}^\omega$: the model **HRO**

The model of Hereditary Recursive Operations **HRO** is defined by

$$\begin{array}{rcl} \mathbf{HRO_N} & \equiv & \mathbb{N} \\ \mathbf{HRO}_{\sigma\to\tau} & \equiv & \{e \in \mathbb{N} \mid \forall n \in \mathbf{HRO}_\sigma \; \{e\}(n) \downarrow\in \mathbf{HRO}_\tau\} \end{array}$$

where $\{e\}(n) \downarrow\in E$ means that the computation of the function of index e terminates on the input n and that the result of this computation is in E.

In this model, terms of System T are interpreted as indexes of the recursive functions they denote.

A model of $\mathbf{I}\text{-}\mathbf{HA}^\omega$: the model HRO

The model of Hereditary Recursive Operations **HRO** is defined by

$$\begin{array}{rcl} \mathbf{HRO_N} & \equiv & \mathbb{N} \\ \mathbf{HRO}_{\sigma\to\tau} & \equiv & \{e \in \mathbb{N} \mid \forall n \in \mathbf{HRO}_\sigma \ \{e\}(n) \downarrow\in \mathbf{HRO}_\tau\} \end{array}$$

where $\{e\}(n) \downarrow\in E$ means that the computation of the function of index e terminates on the input n and that the result of this computation is in E.

In this model, terms of System T are interpreted as indexes of the recursive functions they denote.

Many questions:

1. How a program is interpreted in **HRO**?
2. What is the meta-theory used to define this model? How to "formalize" this model?
3. Is there "a unique model" of **HRO**, i.e. does defining **HRO** in two different meta-theories give the same result?
4. Which other logical principles does the model **HRO** satisfy?

A deeper study of the model **HRO**

An historical motivation: the theory **HRO**

2.4.10. The formal theories HRO, HRO$^-$.

HRO is an extension of $\mathbf{I}$-$\mathbf{HA}^\omega$ in which it is asserted that the objects of type σ coincide with the hereditarily recursive operations of type σ.

The language of HRO is obtained by adding constants $\Phi_\sigma \in (\sigma)0$, $\Phi'_{\sigma,\tau} \in ((\sigma)\tau)(\sigma)0$ for all $\sigma, \tau \in \mathbf{T}$ to the language of $\mathbf{I}$-$\mathbf{HA}^\omega$. HRO is axiomatized by addition of the following axioms to $\mathbf{I}$-$\mathbf{HA}^\omega$:

G1. $\Phi_0 x^0 = x^0$

G2. $\Phi_\sigma x^\sigma = \Phi_\sigma y^\sigma \leftrightarrow x^\sigma = y^\sigma$

G3. $T(\Phi_{(\sigma)\tau} x^{(\sigma)\tau}, \Phi_\sigma y^\sigma, \Phi'_{\sigma,\tau} x^{(\sigma)\tau} y^\sigma)$

G4. $\Phi_\tau x^{(\sigma)\tau} y^\sigma = U(\Phi'_{\sigma,\tau} x^{(\sigma)\tau} y^\sigma)$ (T, U as in 1.3.9 A)

G5. $\forall x \in V_\sigma \exists y^\sigma (\Phi_\sigma y = x)$.

G1 - 4 express that all objects of finite type are hereditarily recursive operations; G5 expresses that every hereditarily recursive operation is an object of finite type. HRO$^-$ is obtained by deleting G5 from HRO.

HRO is a model for HRO. To see this, we only have to interpret Φ_σ by $(\Lambda x.x, (\sigma)0)$ and $\Phi'_{\sigma,\tau}$ by $(\Lambda x \Lambda y . \min_z T(x,y,z), ((\sigma)\tau)(\sigma)0)$.

[Anne S. Troelstra - Metamathematical Investigation of Intuitionistic Arithmetic and Analysis]

Rejuvenating $\underset{\sim}{\mathbf{HRO}}$

The theory $\underset{\sim}{\mathbf{HRO}}$ is obtained from $\mathbf{N\text{-}HA}^\omega$ by adding

- the following constants to the language of terms:
  1. $\mathtt{quote}_\sigma : \sigma \to \mathbf{N}$
  2. $\mathtt{ev}_{\sigma,\tau} : (\sigma \to \tau) \to \sigma \to \mathbf{N}$
- the closure of the following formulas as axioms:
  1. $\mathtt{quote}_\mathbf{N}\, x^\mathbf{N} =_\mathbf{N} x$ ($\mathtt{quote}_\mathbf{N} : \mathbf{N} \to \mathbf{N}$ is the identity)
  2. $(\mathtt{quote}_\sigma\, x =_\mathbf{N} \mathtt{quote}_\sigma\, y \Leftrightarrow x =_\sigma y)$ ($\mathtt{quote}_\sigma$ is injective)
  3. $\mathbf{T}(\mathtt{quote}_{\sigma\to\tau}\, x^{\sigma\to\tau}, \mathtt{quote}_\sigma\, y^\sigma, \mathtt{ev}_{\sigma,\tau}\, x^{\sigma\to\tau}\, y^\sigma)$ ($\mathtt{ev}_{\sigma,\tau}\, x^{\sigma\to\tau}\, y^\sigma$ encodes a computation)
  4. $\mathtt{quote}_\tau\, (x^{\sigma\to\tau} y^\sigma) =_\mathbf{N} \mathbf{U}(\mathtt{ev}_{\sigma,\tau}\, x^{\sigma\to\tau}\, y^\sigma)$ (codes of normal forms)
  5. $\forall x^\mathbf{N} \in \mathbf{HRO}_\sigma \exists y^\sigma\, (\mathtt{quote}_\sigma\, y =_\mathbf{N} x)$ (an "unskolemized" unquote)

Glossary

- $\mathbf{T}(e, n, k)$ is the Kleene's predicate stating that the integer k codes the computation of the algorithm e applied to the entry n
- $\mathbf{U} : \mathbf{N} \to \mathbf{N}$ is a program extracting the result of a code of computations (given as an entry)
- $\forall x^\mathbf{N} \in \mathbf{HRO}_\sigma \Phi \equiv \forall x^\mathbf{N}(\mathbf{HRO}_\sigma(x) \Rightarrow \Phi)$ where $\mathbf{HRO}_\sigma(x)$ is a formula that will be defined later

HRO is a model for... $\underset{\sim}{\textbf{HRO}}$

> HRO is a model for $\underset{\sim}{\text{HRO}}$. To see this, we only have to interpret Φ_σ by $(\Lambda x.x, (\sigma)0)$ and $\Phi'_{\sigma,\tau}$ by $(\Lambda x \Lambda y . \min_z T(x,y,z), ((\sigma)\tau)(\sigma)0)$.

[Anne S. Troelstra - Metamathematical Investigation of Intuitionistic Arithmetic and Analysis]

We only have to

1. interpret $\mathtt{quote}_\sigma$ as a code for the identity function, viewed as an element of $\mathbf{HRO}_{\sigma \to \mathbf{N}}$
2. interpret $\mathtt{ev}_{\sigma,\tau}$ using unbounded search.

HRO is a model for... **HRO**

> HRO is a model for **HRO**. To see this, we only have to interpret Φ_σ by $(\Lambda x.x, (\sigma)0)$ and $\Phi'_{\sigma,\tau}$ by $(\Lambda x \Lambda y . \min_z T(x,y,z), ((\sigma)\tau)(\sigma)0)$.

[Anne S. Troelstra - Metamathematical Investigation of Intuitionistic Arithmetic and Analysis]

We only have to

1. interpret $\mathtt{quote}_\sigma$ as a code for the identity function, viewed as an element of $\mathbf{HRO}_{\sigma\to\mathbf{N}}$
2. interpret $\mathtt{ev}_{\sigma,\tau}$ using unbounded search.

...Still a lot of question! But we answered one: the theory **HRO** is worth studying!

Formalizing the model **HRO**

In fact, the model **HRO** can be formalized in **HA**.
It can be described as a translation from a proof system $\vdash_{\underset{\sim}{\mathbf{HRO}}}$ for $\underset{\sim}{\mathbf{HRO}}$ into a proof system $\vdash_{\mathbf{HA}}$ for **HA** such that:

$$\vdash_{\underset{\sim}{\mathbf{HRO}}} M : \phi \text{ implies } \vdash_{\mathbf{HA}} [\![M]\!] : [\![\phi]\!]$$

This translation shows a result of relative consistency between $\underset{\sim}{\mathbf{HRO}}$ and **HA**.
But, it shows even more!

Theorem

$\underset{\sim}{\mathbf{HRO}}$ is a conservative extension of **HA**.

Next step: study this translation (on-going work)!

A glimpse of the translation (1/3): proof systems

Assume a proof system (à la Church) for $\underset{\sim}{\mathbf{HRO}}$ where the sequents have the form

$$\Delta; \Gamma \vdash_{\underset{\sim}{\mathbf{HRO}}} M : \Phi$$

where

- Δ is a signature of System T
- Γ is a context
- M is a proof term
- Φ is a formula of the language of $\underset{\sim}{\mathbf{HRO}}$

and a proof system (à la Curry) for **HA** where the sequents have the form

$$\Gamma \vdash_{\mathbf{HA}} M : \Phi$$

We show how to translate the former in the latter.

Remark

Designing such proof systems is not a challenge: they are not included only to save time.

A glimpse of the translation (2/3): preliminaries in **HA**

1. Define an encoding of the λ-calculus with an extra instruction $\mathtt{q}$
   - if x and y are codes, the code of the application of x to y is denoted $x@y$
   - The instruction $\mathtt{q}$ has the following operational semantic: it returns the code of its entry as a Church numeral if its entry is a closed normal form and it does not reduce otherwise
2. Define in the language of **HA**
   - a ternary formula $c_1 \overset{k}{\rightsquigarrow} c_2$ in the language of **HA** stating that k is the code of a reduction from the code c_1 to the code c_2
   - a formula $\mathbf{SN_c}(x)$ stating that x is the code of a closed strongly normalizable term
   - a formula $\mathbf{Num_{Ch}}(x)$ stating that x is the code of a Church numeral
3. Define formulas $\mathbf{HRO}_\sigma(x)$ in the language of **HA**
   - $\mathbf{HRO_N}(x) \equiv \mathbf{SN_c}(x) \wedge \exists n \exists k(\mathbf{Num_{Ch}}(n) \wedge x \overset{k}{\rightsquigarrow} n)$
   - $\mathbf{HRO}_{\sigma\to\tau}(x) \equiv \mathbf{SN_c}(x) \wedge \forall y(\mathbf{HRO}_\sigma(y) \Rightarrow \mathbf{HRO}_\tau(x@y))$

Remark

- Codes are not normalized. I chose to normalize in **HA** and not in the meta-theory.
- As a consequence, codes should be reduced before being compared.

A glimpse of the translation (3/3): the translation

1. Translate terms of $\underset{\sim}{\mathbf{HRO}}$ into codes: $t \mapsto [\![t]\!]$
   - Terms are translated syntactically, $t_1 \cong t_2$ does not imply $[\![t_1]\!] = [\![t_2]\!]$
   - A variable x^σ of System T is translated into a first-order variable n_x of **HA**
   - $\texttt{quote}_\sigma$ is translated as $\texttt{q}$
2. Translate terms of $\underset{\sim}{\mathbf{HRO}}$ into proofs of **HA**: $t \mapsto \hat{t}$
   - A variable x^σ of System T is translated into a proof variable $\hat{x} : \mathbf{HRO}_\sigma(n_x)$
   - If $\Delta \vdash_{\mathrm{T}} t : \sigma$, then $[\![\Delta]\!] \vdash_{\mathbf{HA}} \hat{t} : \mathbf{HRO}_\sigma([\![t]\!])$
3. Translate formulas of $\underset{\sim}{\mathbf{HRO}}$ into formulas of **HA**: $\Phi \mapsto [\![\Phi]\!]$
   - $[\![t_1 =_\sigma t_2]\!] \equiv \exists c \exists k_1 \exists k_2 ([\![t_1]\!] \overset{k_1}{\rightsquigarrow} c \wedge [\![t_2]\!] \overset{k_2}{\rightsquigarrow} c)$
   - $[\![\forall x^\sigma \Phi]\!] \equiv \forall n_x\, (\mathbf{HRO}_\sigma(n_x) \Rightarrow [\![\Phi]\!])$ (etc...)
4. Translate proofs of $\underset{\sim}{\mathbf{HRO}}$ into proofs of **HA**: $M \mapsto [\![M]\!]$
   - If $\Delta; \Gamma \vdash_{\underset{\sim}{\mathbf{HRO}}} M : \Phi$ then $[\![\Delta]\!][\![\Gamma]\!] \vdash_{\mathbf{HA}} [\![M]\!] : [\![\Phi]\!]$

Disclaimer: Free variables should be treated carefully!

A first-order variable $x : \sigma$ (in $\underset{\sim}{\mathbf{HRO}}$) needs to be translated into a first-order variable n_x of **HA** but also into a proof variable $\xi_x : \mathbf{HRO}_\sigma([\![x]\!])$ (of **HA**).

Remarks on the translation

1. An equality in $\underset{\sim}{\mathbf{HRO}}$ is translated into a Σ_1^0-formula of **HA**: two codes are reduced and then compared, i.e. the "meta-theoretic" quotient over System T in $\underset{\sim}{\mathbf{HRO}}$ (axiomatized or defined as a conversion rule) is internalized in **HA**.
   - The predicate $c_1 \overset{k}{\rightsquigarrow} c_2$ can be modified to enforce c_2 to be a normal form. In this case, the normal form acts as a canonical represent for classes of β-equivalent terms.
2. However, the equality is (internally) decidable in $\underset{\sim}{\mathbf{HRO}}$: it is a consequence of the injectivity of $\mathtt{quote}_\sigma$.
   - Note that the formula $[\![t =_\sigma u]\!]$ (of **HA**) is not decidable in **HA**! One cannot prove the normalization of all terms of System T of sorts σ in **HA**.
3. The translation does not speak explicitly about reduction rules of `quote` but it cannot distinguish two β-equivalent terms.
4. Is the relativization to the predicate $\mathbf{HRO}_\sigma(x)$ necessary to interpret `quote` via the translation ?
   - I think that possibly not. But, in this case, codes can't be normalized and the reduction of `quote` should be changed.

Answering the questions (1/3)

Question 1

How a program is interpreted?

I interpreted a program as its "own" code (statically). The normal forms are computed in **HA**. However, one could imagine to first normalize a term (in the meta-theory) and then to interpret it. I think that this is the process used by Troelstra:

We now define our HRO - analogue λ - HRO as follows.

$$t \in V_0^\lambda \equiv_{def} \exists \underline{n}(t \equiv \underline{n})$$

$$t \in V_{(\sigma)\tau}^\lambda \equiv_{def} t \text{ normal, closed, } \forall t' \in V_\sigma^\lambda \exists t'' \in V_\tau^\lambda (tt' = t'') .$$

[Anne S. Troelstra - Metamathematical Investigation of Intuitionistic Arithmetic and Analysis]

Answering the questions (1/3)

Question 1

How a program is interpreted?

I interpreted a program as its "own" code (statically). The normal forms are computed in **HA**. However, one could imagine to first normalize a term (in the meta-theory) and then to interpret it. I think that this is the process used by Troelstra:

We now define our HRO - analogue λ - HRO as follows.

$$t \in V_0^{\lambda} \equiv_{def} \exists \underline{n}(t \equiv \underline{n})$$

$$t \in V_{(\sigma)\tau}^{\lambda} \equiv_{def} t \text{ normal, closed, } \forall t' \in V_{\sigma}^{\lambda}\, \exists t'' \in V_{\tau}^{\lambda}(tt' = t'') .$$

[Anne S. Troelstra - Metamathematical Investigation of Intuitionistic Arithmetic and Analysis]

Slogan

(My version of) **HRO** is "just" an internalization of the conversion rule!

Answering the questions (2/3)

Question

What is the meta-theory used to define this model? How to "formalize" this model?

It seems that one can use **HA** as a theory to define the model **HRO** while **PRA** seems to be enough to design the translation from $\vdash_{\utilde{\mathbf{HRO}}}$ into $\vdash_{\mathbf{HA}}$.

Answering the questions (3/3)

Question

Is there “a unique model” of **HRO**, i.e. does defining **HRO** in two different “meta-theories” give the same result?

The set $\mathbf{HRO}_{\mathbf{N}\to\mathbf{N}}$ contains all the (codes of) recursive functions provably total in the meta-theory. Therefore it contains more functions if the “meta-theory” is stronger.

Disclaimer

HRO is not a term model (initial model) for $\mathbf{HA}^{\omega}$!

Future work (1/3): revisiting $\underset{\sim}{\textbf{HRO}}$

1. System T with a family of new types of the form Code_σ with a syntax for normal forms
2. Reduction rules for $\texttt{quote}_\sigma : \sigma \to \mathrm{Code}_\sigma$
3. Add instructions $\texttt{unquote}_\sigma : \mathrm{Code}_\sigma \to \sigma$
4. All this can again be interpreted in **HA** (this is a proof of relative consistency, not of normalization)

Take inspirations from the works of Jason Hu and Brigitte Pientka about Layered Modal Type Theories?

Future work (2/3): Kreisel realizability in $\underset{\sim}{\mathbf{HRO}}$

1. Kreisel realizability formalized in a proof system for $\underset{\sim}{\mathbf{HRO}}$
2. Can realize AC and CT_0 (mix of Kleene and Kreisel realizability)
   - $\mathrm{AC} \equiv \forall x^{\sigma} \exists y^{\tau} \Phi(x, y) \Rightarrow \exists f^{\sigma \to \tau} \forall x^{\sigma} \Phi(x, f(x))$
   - $\mathrm{CT}_0 \equiv \forall x^{\mathbf{N}} \exists y^{\mathbf{N}} \Phi(x, y) \Rightarrow \exists f^{\mathbf{N}} \forall x^{\mathbf{N}} \exists k^{\mathbf{N}} (\mathbf{T}(f, x, k) \wedge \Phi(x, \mathbf{U}(k)))$
3. Proof: AC is validated by Kreisel realizability while CT is provable in $\underset{\sim}{\mathbf{HRO}}$
   - $\mathrm{CT} \equiv \forall f^{\mathbf{N} \to \mathbf{N}} \exists c^{\mathbf{N}} \forall x^{\mathbf{N}} \exists k^{\mathbf{N}} (\mathbf{T}(c, x, k) \wedge f(x) = \mathbf{U}(k))$

Future work (2/3): Kreisel realizability in $\underset{\sim}{\mathbf{HRO}}$

1. Kreisel realizability formalized in a proof system for $\underset{\sim}{\mathbf{HRO}}$
2. Can realize AC and CT_0 (mix of Kleene and Kreisel realizability)
   - $\mathrm{AC} \equiv \forall x^\sigma \exists y^\tau \Phi(x, y) \Rightarrow \exists f^{\sigma \to \tau} \forall x^\sigma \Phi(x, f(x))$
   - $\mathrm{CT}_0 \equiv \forall x^{\mathbf{N}} \exists y^{\mathbf{N}} \Phi(x, y) \Rightarrow \exists f^{\mathbf{N}} \forall x^{\mathbf{N}} \exists k^{\mathbf{N}} (\mathbf{T}(f, x, k) \wedge \Phi(x, \mathbf{U}(k)))$
3. Proof: AC is validated by Kreisel realizability while CT is provable in $\underset{\sim}{\mathbf{HRO}}$
   - $\mathrm{CT} \equiv \forall f^{\mathbf{N} \to \mathbf{N}} \exists c^{\mathbf{N}} \forall x^{\mathbf{N}} \exists k^{\mathbf{N}} (\mathbf{T}(c, x, k) \wedge f(x) = \mathbf{U}(k))$

... but already done by Troelstra!

(ii) CT is $\underset{\sim}{\mathrm{HRO}}^-$ - $\underset{\approx}{\mathrm{mr}}$ - realizable, but not $\underset{\sim}{\mathrm{HEO}}$ - $\underset{\approx}{\mathrm{mr}}$ - realizable. CT_0 (w.r.t. $\underset{\sim}{\mathrm{HRO}}^-$) is $\underset{\sim}{\mathrm{HRO}}^-$ - $\underset{\approx}{\mathrm{mr}}$ - realizable, hence also $\underset{\sim}{\mathrm{HRO}}$ - $\underset{\approx}{\mathrm{mr}}$ - realizable.

[Anne S. Troelstra - Metamathematical Investigation of Intuitionistic Arithmetic and Analysis]

Personal opinion

I think it would be interesting to take a closer look at this realizability interpretation.

Future work (3/3): BBC functional in (some variant of) HRO

1. Berardi, Bezem and Coqand gave a realizability interpretation of the negative translation of the axiom of choice $\mathrm{G\ AC}(\mathbf{N}, \tau)$
   - $\mathrm{G\ AC}(\mathbf{N}, \tau) \equiv \forall x^{\mathbf{N}} \neg\neg\exists y^{\tau} \neg\Phi(x, y) \Rightarrow \neg\neg\exists f^{\mathbf{N}\to\tau} \forall x^{\sigma} \neg\Phi(x, f(x))$
2. Their interpretation uses the decidable equality over the type **N**

   It should be noted also that our interpretation works for $\mathsf{AC}(\tau, \tau')$ where $\tau = \mathsf{N}$ but τ' is arbitrary. As it is given, it uses in an essential way the restriction $\tau = \mathsf{N}$, but we shall present below heuristic arguments that suggest a possible extension to higher types.

   [Stefano Berardi, Marc Bezem and Thierry Coquand
   On the computational content of the axiom of choice]

3. It should be possible to extend their interpretation to the full axiom of choice in HRO (or even in **I-HA**$^{\omega}$)

Future work (3/3): BBC functional in (some variant of) HRO

1. Berardi, Bezem and Coqand gave a realizability interpretation of the negative translation of the axiom of choice $\mathrm{G\ AC}(\mathbf{N}, \tau)$
   - $\mathrm{G\ AC}(\mathbf{N}, \tau) \equiv \forall x^{\mathbf{N}} \neg\neg\exists y^{\tau} \neg\Phi(x, y) \Rightarrow \neg\neg\exists f^{\mathbf{N}\to\tau} \forall x^{\sigma} \neg\Phi(x, f(x))$
2. Their interpretation uses the decidable equality over the type **N**

> It should be noted also that our interpretation works for $\mathsf{AC}(\tau, \tau')$ where $\tau = \mathsf{N}$ but τ' is arbitrary. As it is given, it uses in an essential way the restriction $\tau = \mathsf{N}$, but we shall present below heuristic arguments that suggest a possible extension to higher types.

[Stefano Berardi, Marc Bezem and Thierry Coquand
On the computational content of the axiom of choice]

3. It should be possible to extend their interpretation to the full axiom of choice in HRO (or even in **I-HA**$^{\omega}$)

Disclaimer

It is already known, as remarked by É.Miquey: “At first sight, our approach seems to extend to any type A with decidable equality...” (*A constructive proof of dependent choice in classical arithmetic via memoization*).

quote in other frameworks

1. Links with the Constructible Universe **L** in Set Theory
   - A set is associated to an ordinal and a code
   - If the hierarchy is truncated to a "recursive level", `quote` could be fully implemented (in a classical setting)?
2. Links with Tarski universes in type theory

   5.2 Ad-hoc polymorphism

   In this section, we push the concept of type intensionality into its utmost consequencies, namely by showing that there exists a syntactic model of type theory interpreting a quoting operator on types. Such an operator allows to do

   [Simon Boulier, Pierre-Marie Pédrot and Nicolas Tabareau
   The Next 700 Syntactical Models of Type Theory]

3. Links with the MetaCoq project
   - Works of Jason Gross about proving Löb's theorem in MetaCoq
4. `quote` in dependent type theory
   - Works of Pierre-Marie Pédrot about the consistency of Church's Thesis with respect to MLTT