Internal Proofs of Strong Normalization

Cody Roux

June 10, 2025

Mantra

Type theory should eat itself.

— James Chapman[Chapman, 2009]

Mantra

A type theory $\mathcal{T}$ should eat itself.

— Me, just now

Strong Normalization

Traditionally, whenever we come up with a type theory $\mathcal{T}$

- We have an operational semantics in mind
- We prove progress and preservation
- We prove SN
- We get consistency, decidability, etc.

Strong Normalization

This is all quite tedious. We'd like to have a big meta-theorem to avoid this.

These days, most people prefer to do NbE.

But SN does have the advantage of concreteness:

1. no "secret" reductions.
2. any reduction strategy works.

Strong Normalization

Still one or two open problems. The one I care about:

Conjecture

Barendregt Geuvers Klop (...):

For every Pure Type System $\mathcal{T}$, $\mathcal{T}$ is SN iff $\mathcal{T}$ is WN.

We'll come back to this.

What We Want

I want to be able to prove SN for a type theory $\mathcal{T}$ *in* $\mathcal{T}$.

Roughly following the "usual" candidates proof.

$$[\![\Pi x : A.B]\!]\ t = \Pi u : \mathrm{Term}.\Pi x : [\![A]\!]\ u.[\![B]\!]\ x \ulcorner t\ u \urcorner$$

$$[\![\Pi X : s.B]\!]\ t = \Pi \mathcal{X} : \mathrm{Term} \to s.\text{"}\mathcal{X} \text{ is a candidate"} \to [\![B]\!]\ t$$

Obstructions

1. Gödel
2. $\mathcal{T}$ may not even be able to talk about syntax!
3. $\mathcal{T}$ may not have induction

Obstructions

1. Gödel
2. $\mathcal{T}$ may not even be able to talk about syntax!
3. $\mathcal{T}$ may not have induction
4. ... and $\mathcal{T}$ may not be normalizing

Obstruction 1: Gödel

Here is a folklore theorem (Heyting?):

Theorem

For any proposition P, if

$$\mathit{HA} \vdash P$$

then

$$\mathit{HA} \vdash \ulcorner t \Vdash P \urcorner$$

for some term t.

Note that this is not $\mathit{HA} \vdash \forall P, \ulcorner \mathit{HA} \vdash P \urcorner \rightarrow \exists t, \ulcorner t \Vdash P \urcorner$!
Uses *meta induction*.

Obstruction 1: Gödel

For a type theory $\mathcal{T}$, for each term $\Gamma \vdash_{\mathcal{T}} t : T$, build a proof

$$\cdot \vdash_{\mathcal{T}} t' : SN \ulcorner t \urcorner$$

but what of Γ?

Obstruction 2: Syntax

Actually work in a theory $\hat{\mathcal{T}}$ that contains a *new sort* Syn and

$$\mathrm{Term} : \mathrm{Syn}, \mathrm{var}\ i : \mathrm{Term}, \lambda.\mathrm{var}\ i\langle \mathrm{var}\ j\rangle, \ldots$$

a new Π:

$$\Pi : \mathrm{Syn} \rightsquigarrow s$$

idea: add explicit substitution rules as new conversion rules!

We also add a predicate

$$\mathsf{SN} : \mathrm{Term} \to s$$

for our target sorts s. (think $s = \mathrm{Type}$ or $s = \mathrm{Prop}$)

Obstruction 2: Syntax

Theorem (Main theorem)

$\hat{\mathcal{T}}$ is (weakly) normalizing iff $\mathcal{T}$ is.

Obstruction 3: Induction

We are *not allowed* to use induction over terms.

The Girard candidates proof actually uses an inner induction in the object theory!

We use (a variant of) the saturated sets proof. Types become predicates, terms become proofs.
Complication: we need

$$\mathsf{ne}_{\mathcal{X}} : \forall t : Term, \mathsf{NE}\ t \to \mathcal{X}\ t$$

$$\mathsf{sn}_{\mathcal{X}} : \forall t : Term, \mathcal{X}\ t \to \mathsf{SN}\ t$$

$$\mathsf{wh}_{\mathcal{X}} : \forall t\ t' : Term, (t \longrightarrow_{\mathsf{wh}} t') \to \mathcal{X}\ t' \to \mathcal{X}\ t$$

for all new "candidate variables" $\mathcal{X} : \mathrm{Term} \to s$, and some computation rules.
This preserves the main theorem.

Obstruction 4: Non-normalizing systems

This translation still works for non-normalizing systems!

However you cannot trust the resulting proofs of SN.

Observation 1

Proofs of SN t in normal form are *real proofs* of strong normalization!

Observation 2

This implies the BGK conjecture for $\mathcal{T}$:

- If $\mathcal{T}$ is WN, build $\hat{\mathcal{T}}$ (also WN)
- For each $\Gamma \vdash_{\mathcal{T}} t : T$ there is some $\vdash_{\hat{\mathcal{T}}} t' : \mathsf{SN} \ulcorner t \urcorner$
- t' has a normal form: t is actually SN!

Disclaimers/Confessions

For which $\mathcal{T}$ can we carry this translation out?

So far:

- STLC
- System F
- The "trivial non-terminating" theory. $(D = D \to D)$

Should generalize easily to

- Left hand side of Barendregt cube
- "Non dependent logical PTSes" [Coquand and Herbelin, 1994]

Sadly, these are already known cases for BGK! [Barthe et al., 2001]

But we know how to internalize logical relations for all PTS![Bernardy and Lasson, 2011]
(and many other versions of internal logical relations)

Question

Are we ready to put the question to rest? What of other type theories?

Related Topics

Flavor is very reminiscent of *synthetic Tait computability*[Sterling, 2022].

What is the relationship?

References

Barthe, G., Hatcliff, J., and Sørensen, M. H. (2001).
Weak normalization implies strong normalization in a class of non-dependent pure type systems.
Theoretical Computer Science, 269(1):317–361.

Bernardy, J.-P. and Lasson, M. (2011).
Realizability and parametricity in pure type systems.
In Hofmann, M., editor, *Foundations of Software Science and Computational Structures*, pages 108–122, Berlin, Heidelberg. Springer Berlin Heidelberg.

Chapman, J. (2009).
Type theory should eat itself.
Electron. Notes Theor. Comput. Sci., 228:21–36.

Coquand, T. and Herbelin, H. (1994).
A-translation and looping combinators in pure type systems.
Journal of Functional Programming, 4(1):77–88.