Closure of System T under the Bar Recursion Rule

Paulo Oliva

(joint work with S. Steila)

Queen Mary University of London

University Leeds

Wednesday, 1 November 2017

Outline

1. Spector's bar recursion
2. Schwichtenberg's proof
3. A new (more direct) proof

Outline

Spector's Bar Recursion

(1958) Gödel's Dialectica interpretation of *arithmetic* (system T)

(1962) Spector extends interpretation to *analysis* (T + BR)

(1968) Howard interpretation of bar induction (T + BR)

(1971) Scarpellini shows $\mathcal{C}$ is a model of BR

(1979) Schwichtenberg closure theorem (*low types*)

(1981) Howard's ordinal analysis of BR (*low types*)

(1985) Bezem shows $\mathcal{M}$ is a model of BR

Spector's Bar Recursion (Rule)

Given $s\colon \tau^*$ let $\hat{s}\colon \tau^{\mathbb{N}}$ be the extension of s with 0's

For each pair of types τ, σ, and given G, H and Y

$$\mathsf{BR}^{\tau,\sigma}(s) \stackrel{\sigma}{=} \begin{cases} G(s) & \text{if } Y(\hat{s}) < |s| \\ H(s)(\lambda x^\tau.\mathsf{BR}(s * x)) & \text{otherwise} \end{cases}$$

where

$$\begin{aligned} G &: \tau^* \to \sigma \\ Y &: \tau^{\mathbb{N}} \to \mathbb{N} \\ H &: \tau^* \to (\tau \to \sigma) \to \sigma \end{aligned}$$

Schwichtenberg's Closure Theorem

Theorem

System T is closed under the bar recursion rule when τ's type level is either 0 or 1

That is, given G, H and Y terms in T, the functional

$$\mathsf{BR}^{\tau,\sigma}(s) \stackrel{\sigma}{=} \begin{cases} G(s) & \text{if } Y(\hat{s}) < |s| \\ H(s)(\lambda x^{\tau}.\mathsf{BR}(s * x)) & \text{otherwise} \end{cases}$$

is also T definable

Counter-example for $\tau > 1$

Howard (1968) showed that bar recursion of type ρ can be defined using the bar recursion rule of type $(\mathbb{N} \to \rho) \to \rho$

Counter-example for $\tau > 1$

Howard (1968) showed that bar recursion of type ρ can be defined using the bar recursion rule of type $(\mathbb{N} \to \rho) \to \rho$

Since bar recursion, even of type $\rho = \mathbb{N}$, is not T definable

Counter-example for $\tau > 1$

Howard (1968) showed that bar recursion of type ρ can be defined using the bar recursion rule of type $(\mathbb{N} \to \rho) \to \rho$

Since bar recursion, even of type $\rho = \mathbb{N}$, is not T definable

it follows that T is not closed under the bar recursion rule for $\tau = (\mathbb{N} \to \mathbb{N}) \to \mathbb{N}$

Outline

Schwichtenberg's Proof

Published in The Journal of Symbolic Logic (1971)

"On bar recursion of type 0 and 1"

5 pages long (actual proof only two pages long)

Schwichtenberg's Proof

1. Translate terms G, H, Y into infinitary terms (get rid of recursor)

Schwichtenberg's Proof

1. Translate terms G, H, Y into infinitary terms (get rid of recursor)
2. Define a bar $S_Y(s) =$ "sequence s is secure for term Y"

Schwichtenberg's Proof

1. Translate terms G, H, Y into infinitary terms (get rid of recursor)
2. Define a bar $S_Y(s)$ = "sequence s is secure for term Y"
3. Complement of $S_Y(s)$ is a tree

Schwichtenberg's Proof

1. Translate terms G, H, Y into infinitary terms (get rid of recursor)
2. Define a bar $S_Y(s) =$ "sequence s is secure for term Y"
3. Complement of $S_Y(s)$ is a tree
4. See BR as a recursion on this tree

Schwichtenberg's Proof

1. Translate terms G, H, Y into infinitary terms (get rid of recursor)
2. Define a bar $S_Y(s) =$ "sequence s is secure for term Y"
3. Complement of $S_Y(s)$ is a tree
4. See BR as a recursion on this tree
5. Define order-preserving embedding of tree into ε_0-ordinals

Schwichtenberg's Proof

1. Translate terms G, H, Y into infinitary terms (get rid of recursor)
2. Define a bar $S_Y(s) =$ "sequence s is secure for term Y"
3. Complement of $S_Y(s)$ is a tree
4. See BR as a recursion on this tree
5. Define order-preserving embedding of tree into ε_0-ordinals
6. Hence, BR can be mimicked by ε_0-ordinal recursion

Schwichtenberg's Proof

1. Translate terms G, H, Y into infinitary terms (get rid of recursor)
2. Define a bar $S_Y(s) =$ "sequence s is secure for term Y"
3. Complement of $S_Y(s)$ is a tree
4. See BR as a recursion on this tree
5. Define order-preserving embedding of tree into ε_0-ordinals
6. Hence, BR can be mimicked by ε_0-ordinal recursion
7. By Tait, we can find equivalent T definition of BR(s)

Outline

Base case: $Y(\alpha)$ is constant

When $Y(\alpha)$ is constant n, BR becomes

$$\mathsf{BR}^{\tau,\sigma}(s) \stackrel{\sigma}{=} \begin{cases} G(s) & \text{if } |s| > n \\ H(s)(\lambda x^{\tau}.\mathsf{BR}(s * x)) & \text{if } |s| \leq n \end{cases}$$

Base case: $Y(\alpha)$ is constant

When $Y(\alpha)$ is constant n, BR becomes

$$\mathsf{BR}^{\tau,\sigma}(s) \stackrel{\sigma}{=} \begin{cases} G(s) & \text{if } |s| > n \\ H(s)(\lambda x^{\tau}.\mathsf{BR}(s * x)) & \text{if } |s| \leq n \end{cases}$$

It is easy to write down a T term (uniformly in G and H) computing the same function

Base case: $Y(\alpha)$ is constant

When $Y(\alpha)$ is constant n, BR becomes

$$\mathsf{BR}^{\tau,\sigma}(s) \stackrel{\sigma}{=} \begin{cases} G(s) & \text{if } |s| > n \\ H(s)(\lambda x^{\tau}.\mathsf{BR}(s * x)) & \text{if } |s| \leq n \end{cases}$$

It is easy to write down a T term (uniformly in G and H) computing the same function

Needs primitive recursion of type $\tau^* \to \sigma$

Base case: $Y(\alpha)$ is constant

When $Y(\alpha)$ is constant n, BR becomes

$$\mathsf{BR}^{\tau,\sigma}(s) \stackrel{\sigma}{=} \begin{cases} G(s) & \text{if } |s| > n \\ H(s)(\lambda x^\tau.\mathsf{BR}(s * x)) & \text{if } |s| \leq n \end{cases}$$

It is easy to write down a T term (uniformly in G and H) computing the same function

Needs primitive recursion of type $\tau^* \to \sigma$

Let us refer to this T term as cBR

Proof Idea

Part 1: Show that BR is definable in "general BR"

Part 2: Show that T is closed under "general BR"

(*first part works for any type, second part requires the type restriction*)

General BR

For any *bar* S consider the defining equation

$$\mathsf{gBR}^S(s) \stackrel{\sigma}{=} \begin{cases} G(s) & \text{if } S(s) \\ H(s)(\lambda x^\tau.\mathsf{gBR}^S(s * x)) & \text{if } \neg S(s) \end{cases}$$

General BR

For any *bar* S consider the defining equation

$$\mathsf{gBR}^S(s) \overset{\sigma}{=} \begin{cases} G(s) & \text{if } S(s) \\ H(s)(\lambda x^\tau.\mathsf{gBR}^S(s * x)) & \text{if } \neg S(s) \end{cases}$$

Definition

We say that a bar S *secures* $Y : \tau^{\mathbb{N}} \to \mathbb{N}$ if for all s^{τ^*}

$$S(s) \quad \Rightarrow \quad \lambda\beta.Y(s * \beta) \text{ is constant}$$

Part 1: BR definable in general BR

Theorem

Fix $Y : \tau^{\mathbb{N}} \to \mathbb{N}$. *The functional*

$$\lambda G, H, s.\mathsf{BR}^{\tau,\sigma}(G, H, Y)(s)$$

is T-definable in gBR^{S}, *for any bar* S *securing* Y

Part 1: BR definable in general BR

Theorem

Fix $Y : \tau^{\mathbb{N}} \to \mathbb{N}$*. The functional*

$$\lambda G, H, s.\mathsf{BR}^{\tau,\sigma}(G, H, Y)(s)$$

is T-definable in gBR^S*, for any bar* S *securing* Y

Proof.

Use the bar S to spot when Y becomes constant, then apply the T construction for the case when Y is constant. □

Part 2: Closure of T under gBR rule

Theorem

Fix a T-term $Y : \tau^{\mathbb{N}} \to \mathbb{N}$*. For some* S *securing* Y *the functional* gBR^S *is* T *definable.*

Part 2: Closure of T under gBR rule

Theorem

Fix a T-term $Y : \tau^{\mathbb{N}} \to \mathbb{N}$. For some S securing Y the functional gBR^S is T definable.

Proof.

(*Construction*) By induction on Y.

(*Correctness proof*) Use a logical relation to show that the constructed term is indeed equivalent to gBR^S. □

The Construction (case $\tau = \mathbb{N}$)

Let $\mathbb{N}^\circ \equiv$ the type of gBR. We will map $\mathbb{N}$ to $\mathbb{N}^\circ$.

Let α be a special variable of type $\mathbb{N} \to \mathbb{N}$ (generic)

$$0^\circ \quad = \quad \lambda G.G$$

The Construction (case $\tau = \mathbb{N}$)

Let $\mathbb{N}^\circ \equiv$ the type of gBR. We will map $\mathbb{N}$ to $\mathbb{N}^\circ$.

Let α be a special variable of type $\mathbb{N} \to \mathbb{N}$ (generic)

$$
\begin{aligned}
0^\circ &= \lambda G.G \\
\mathsf{Succ}^\circ &= \lambda \Phi^{\mathbb{N}^\circ}.\Phi
\end{aligned}
$$

The Construction (case $\tau = \mathbb{N}$)

Let $\mathbb{N}^\circ \equiv$ the type of gBR. We will map $\mathbb{N}$ to $\mathbb{N}^\circ$.

Let α be a special variable of type $\mathbb{N} \to \mathbb{N}$ (generic)

$$
\begin{aligned}
0^\circ &= \lambda G.G \\
\mathsf{Succ}^\circ &= \lambda \Phi^{\mathbb{N}^\circ}.\Phi \\
\alpha^\circ &= \lambda \Phi^{\mathbb{N}^\circ} \lambda G.\Phi(\lambda s'.\mathsf{cBR}(G, Y(\hat{s'}))(s'))
\end{aligned}
$$

The Construction (case $\tau = \mathbb{N}$)

Let $\mathbb{N}^\circ \equiv$ the type of gBR. We will map $\mathbb{N}$ to $\mathbb{N}^\circ$.

Let α be a special variable of type $\mathbb{N} \to \mathbb{N}$ (generic)

$$
\begin{array}{lcl}
0^\circ & = & \lambda G.G \\
\mathsf{Succ}^\circ & = & \lambda \Phi^{\mathbb{N}^\circ}.\Phi \\
\alpha^\circ & = & \lambda \Phi^{\mathbb{N}^\circ} \lambda G.\Phi(\lambda s'.\mathsf{cBR}(G, Y(\hat{s'}))(s')) \\
(\lambda x^\eta.t)^\circ & = & \lambda x^\circ.t^\circ
\end{array}
$$

The Construction (case $\tau = \mathbb{N}$)

Let $\mathbb{N}^\circ \equiv$ the type of gBR. We will map $\mathbb{N}$ to $\mathbb{N}^\circ$.

Let α be a special variable of type $\mathbb{N} \to \mathbb{N}$ (generic)

$$
\begin{aligned}
0^\circ &= \lambda G.G \\
\mathsf{Succ}^\circ &= \lambda \Phi^{\mathbb{N}^\circ}.\Phi \\
\alpha^\circ &= \lambda \Phi^{\mathbb{N}^\circ} \lambda G.\Phi(\lambda s'.\mathsf{cBR}(G, Y(\hat{s'}))(s')) \\
(\lambda x^\eta.t)^\circ &= \lambda x^\circ.t^\circ \\
(uv)^\circ &= u^\circ v^\circ
\end{aligned}
$$

The Construction (case $\tau = \mathbb{N}$)

Let $\mathbb{N}^\circ \equiv$ the type of gBR. We will map $\mathbb{N}$ to $\mathbb{N}^\circ$.

Let α be a special variable of type $\mathbb{N} \to \mathbb{N}$ (generic)

$$
\begin{array}{lcl}
0^\circ & = & \lambda G.G \\
\mathsf{Succ}^\circ & = & \lambda \Phi^{\mathbb{N}^\circ}.\Phi \\
\alpha^\circ & = & \lambda \Phi^{\mathbb{N}^\circ} \lambda G.\Phi(\lambda s'.\mathsf{cBR}(G, Y(\hat{s'}))(s')) \\
(\lambda x^\eta.t)^\circ & = & \lambda x^\circ.t^\circ \\
(uv)^\circ & = & u^\circ v^\circ \\
(\mathsf{Rec}^\eta)^\circ & = & \ldots
\end{array}
$$

The Construction (case $\tau = \mathbb{N}$)

Let $\mathbb{N}^\circ \equiv$ the type of gBR. We will map $\mathbb{N}$ to $\mathbb{N}^\circ$.

Let α be a special variable of type $\mathbb{N} \to \mathbb{N}$ (generic)

$$
\begin{array}{lcl}
0^\circ & = & \lambda G.G \\
\mathsf{Succ}^\circ & = & \lambda \Phi^{\mathbb{N}^\circ}.\Phi \\
\alpha^\circ & = & \lambda \Phi^{\mathbb{N}^\circ} \lambda G.\Phi(\lambda s'.\mathsf{cBR}(G, Y(\hat{s'}))(s')) \\
(\lambda x^\eta.t)^\circ & = & \lambda x^\circ.t^\circ \\
(uv)^\circ & = & u^\circ v^\circ \\
(\mathsf{Rec}^\eta)^\circ & = & \ldots
\end{array}
$$

(*H can be fixed at outset, but extra work to remember Y*)

The Construction: Recursor

Suppose $Y(\alpha) = \mathsf{Rec}(n_\alpha, x_\alpha, f_\alpha)$

The Construction: Recursor

Suppose $Y(\alpha) = \mathsf{Rec}(n_\alpha, x_\alpha, f_\alpha)$

first ensure term n_α is secure (i.e. constant n)

The Construction: Recursor

Suppose $Y(\alpha) = \mathsf{Rec}(n_\alpha, x_\alpha, f_\alpha)$

first ensure term n_α is secure (i.e. constant n)

then ensure x_α is secure

The Construction: Recursor

Suppose $Y(\alpha) = \mathsf{Rec}(n_\alpha, x_\alpha, f_\alpha)$

first ensure term n_α is secure (i.e. constant n)

then ensure x_α is secure

and $f_\alpha(x_\alpha)$ is secure

The Construction: Recursor

Suppose $Y(\alpha) = \mathsf{Rec}(n_\alpha, x_\alpha, f_\alpha)$

first ensure term n_α is secure (i.e. constant n)

then ensure x_α is secure

and $f_\alpha(x_\alpha)$ is secure

...

The Construction: Recursor

Suppose $Y(\alpha) = \mathsf{Rec}(n_\alpha, x_\alpha, f_\alpha)$

first ensure term n_α is secure (i.e. constant n)

then ensure x_α is secure

and $f_\alpha(x_\alpha)$ is secure

...

until $f_\alpha^n(x_\alpha)$ is secure

The Construction: Recursor

Suppose $Y(\alpha) = \mathsf{Rec}(n_\alpha, x_\alpha, f_\alpha)$

first ensure term n_α is secure (i.e. constant n)

then ensure x_α is secure

and $f_\alpha(x_\alpha)$ is secure

...

until $f_\alpha^n(x_\alpha)$ is secure

can be done by induction hypothesis + primitive recursion

The Correctness Proof

Recall $\mathbb{N}^\circ \equiv$ the type of gBR

Fix H. Define logical relation between T terms

Base case:

$$f^{\mathbb{N}^\circ} \sim_{\mathbb{N}} g^{\mathbb{N}^{\mathbb{N}} \to \mathbb{N}} \equiv \exists S \text{ securing } g \text{ such that } f = \mathrm{gBR}^S$$

The Correctness Proof

Recall $\mathbb{N}^\circ \equiv$ the type of gBR

Fix H. Define logical relation between T terms

Base case:

$$f^{\mathbb{N}^\circ} \sim_{\mathbb{N}} g^{\mathbb{N}^{\mathbb{N}} \to \mathbb{N}} \equiv \exists S \text{ securing } g \text{ such that } f = \text{gBR}^S$$

and, as usual:

$$f^{\rho_0^\circ \to \rho_1^\circ} \sim_{\rho_0 \to \rho_1} g^{\mathbb{N}^{\mathbb{N}} \to (\rho_0 \to \rho_1)}$$
$$\equiv \forall x^{\rho_0^\circ} \forall y^{\mathbb{N}^{\mathbb{N}} \to \rho_0} (x \sim_{\rho_0} y \to f(x) \sim_{\rho_1} \lambda\alpha.g(\alpha)(y\alpha))$$

Main Result

Theorem

Given a closed T term $Y : \mathbb{N}^{\mathbb{N}} \to \mathbb{N}$*, then* $(Y\alpha)^{\circ} \sim Y$

Main Result

Theorem

Given a closed T term $Y : \mathbb{N}^{\mathbb{N}} \to \mathbb{N}$, then $(Y\alpha)^{\circ} \sim Y$

Proof.

By structural induction on Y □

Main Result

Theorem

Given a closed T term $Y : \mathbb{N}^{\mathbb{N}} \to \mathbb{N}$*, then* $(Y\alpha)^{\circ} \sim Y$

Proof.

By structural induction on Y □

Corollary

Fix $Y : \mathbb{N}^{\mathbb{N}} \to \mathbb{N}$ *in T. Then* $\lambda G, H, s.\mathsf{BR}(G, H, Y)(s)$ *is T definable*

Conclusion

Stronger result:

- Only Y needs to be T definable

More explicit construction:

- Given concrete Y, reasonably easy to find T definition of $\lambda G, H, s.\mathsf{BR}(G, H, Y)(s)$

Easy to calibrate T fragments:

- If Y is T_i then $\lambda G, H, s.\mathsf{BR}(G, H, Y)(s)$ is in T_j, where $j = 1 + \max\{1, \mathrm{level}(\sigma)\} + i$.