

Module enriched realizability

Helmut Schwichtenberg

Mathematisches Institut, LMU, München

University of Genua, 24. March 2014

Constructive mathematics needs sufficiently rich data. Examples:

- ▶ Real number: Cauchy sequence **and** Cauchy modulus.
- ▶ Uniformly continuous real function on a compact interval: approximating rational function h **and** a modulus ω of uniform continuity.

Overview

- ▶ Spector's bar recursion
- ▶ Module enriched types
- ▶ Module enriched realizability
- ▶ Module enriched bar induction and bar recursion

Notation

$\langle x_0, \dots, x_{n-1} \rangle :=$ the finite list with elements $x_0, \dots, x_{n-1}$

$|s| :=$ the length of s , i.e., $|\langle x_0, \dots, x_{n-1} \rangle| = n$

$s_i :=$ the i -th element of s if $i < |s|$

$s@f :=$ overwriting f by s , i.e., $s@f = \lambda_i \begin{cases} (s)_i & \text{if } i < |s| \\ f(i) & \text{otherwise} \end{cases}$

$s * x :=$ the finite list s extended by one element x

Spector's bar recursion

$$\Phi_s YGH =_{\tau} \begin{cases} Gs & \text{if } Y(s @ \lambda_x 0) < |s|, \\ H(s, \lambda_x \Phi(s * x, Y, G, H)) & \text{otherwise.} \end{cases}$$

This makes implicit use of the assumption that Y is continuous, i.e., only needs a finite initial part of its function argument to produce a numerical value.

- ▶ Here: make this assumption explicit.
- ▶ Benefit: bar recursion turns into ordinary transfinite recursion on the well-founded tree of “unsecured” sequences of Y .

- ▶ Spector's bar recursion
- ▶ **Module enriched types**
- ▶ Module enriched realizability
- ▶ Module enriched bar induction and bar recursion

Work in the Scott-Ershov model of partial continuous functionals.

$$\begin{aligned} |Ca_1^* \dots a_n^*| &:= 1 + \max\{|a_i^*| \mid i = 1, \dots, n\}, & |*| &:= 0, \\ |(U, b)| &:= \max\{1 + |U|, 1 + |b|\}, \\ |\{a_i \mid i \in I\}| &:= \max\{1 + |a_i| \mid i \in I\}. \end{aligned}$$

Let $f: \vec{\rho} \rightarrow \iota$ and $m: \mathbf{N}$. The **restriction** $f \upharpoonright m$ is the result of removing from f all tokens $(\vec{U}, a)$ with height $|\vec{U}| \geq m$:

$$f \upharpoonright m := \{(\vec{U}, a) \in f \mid |\vec{U}| < m\}.$$

Call m a **bar** for f at $\vec{x}$ if for all tokens $a: \iota$

$$a \in f\vec{x} \text{ implies } a \in f(\vec{x} \upharpoonright m).$$

For any type ρ define its **module enrichment** ρ^+ by recursion on ρ . If $\text{lev}(\rho) \leq 1$ then $\rho^+ := \rho$. Let $\text{lev}(\rho) \geq 2$, say $\rho = \vec{\sigma} \rightarrow \iota$. Then

$$\rho^+ := \vec{\sigma}^+ \rightarrow \mathbf{N} \times \iota.$$

- ▶ A functional f of type $\rho^+ = \vec{\sigma}^+ \rightarrow \mathbf{N} \times \iota$ can be viewed as a pair of two functionals $\hat{f}: \vec{\sigma}^+ \rightarrow \mathbf{N}$ and $\mathring{f}: \vec{\sigma}^+ \rightarrow \iota$.
- ▶ Call $\hat{f}$ the **module part** and $\mathring{f}$ the **object part** of $f: \rho^+$.

Examples

- ▶ The module enrichment of $\rho := (\iota \rightarrow \iota) \rightarrow \iota \rightarrow \iota$ with ι a base type is $\rho^+ := (\iota \rightarrow \iota) \rightarrow \iota \rightarrow \mathbf{N} \times \iota$.
- ▶ If we go one type level up, then

$$\tau := ((\iota \rightarrow \iota) \rightarrow \iota \rightarrow \iota) \rightarrow (\iota \rightarrow \iota) \rightarrow \iota \rightarrow \iota$$

has the module enrichment

$$\tau^+ := ((\iota \rightarrow \iota) \rightarrow \iota \rightarrow \mathbf{N} \times \iota) \rightarrow (\iota \rightarrow \iota) \rightarrow \iota \rightarrow \mathbf{N} \times \iota.$$

In the following we only consider module enriched types.

A functional f of (module enriched) type ρ^+ is **modulated** if

- ▶ $\text{lev}(\rho^+) \leq 1$ or else
- ▶ for all modulated $\vec{x}$ the number $\hat{f}\vec{x}$ is a bar for $\mathring{f}$ at $\vec{x}$.

Example. Let

$$\tau^+ := ((\iota \rightarrow \iota) \rightarrow \iota \rightarrow \mathbf{N} \times \iota) \rightarrow (\iota \rightarrow \iota) \rightarrow \iota \rightarrow \mathbf{N} \times \iota.$$

$f: \tau^+$ splits into the module part $\hat{f}$ and the object part $\mathring{f}$ of types

$$\hat{f}: ((\iota \rightarrow \iota) \rightarrow \iota \rightarrow \mathbf{N} \times \iota) \rightarrow (\iota \rightarrow \iota) \rightarrow \iota \rightarrow \mathbf{N},$$

$$\mathring{f}: ((\iota \rightarrow \iota) \rightarrow \iota \rightarrow \mathbf{N} \times \iota) \rightarrow (\iota \rightarrow \iota) \rightarrow \iota \rightarrow \iota.$$

f is modulated if for all modulated arguments

$$x^+: (\iota \rightarrow \iota) \rightarrow \iota \rightarrow \mathbf{N} \times \iota,$$

$$y: \iota \rightarrow \iota,$$

$$z: \iota$$

the number $\hat{f}x^+yz$ is a bar for $\mathring{f}$ at x^+, y, z .

- ▶ Spector's bar recursion
- ▶ Module enriched types
- ▶ **Module enriched realizability**
- ▶ Module enriched bar induction and bar recursion

A theory of modulated computable functionals

- ▶ Variables, constants and all terms are of module enriched type.
- ▶ Formulas are built from predicates (predicate variables, comprehension terms or inductive predicates) applied to terms by (decorated) implication $A \rightarrow^c B$ or $A \rightarrow^{nc} B$ and universal quantification $\forall_x^c A$ or $\forall_x^{nc} A$.
- ▶ Following Kolmogorov (1932), for every formula or predicate C we define $\tau(C)$ (a type or the “nulltype” symbol $\circ$). In case $\tau(C) = \circ$ proofs of C have no computational content; such C are called **non-computational** (n.c.) (**Harrop**); the other ones are called **computationally relevant** (c.r.).

The type $\tau(C)$ of a formula or predicate C , and ι_I

$\tau(C)$ type or the “nulltype symbol” $\circ$. Extend use of $\rho \rightarrow \sigma$ to $\circ$:

$$(\rho \rightarrow \circ) := \circ, \quad (\circ \rightarrow \sigma) := \sigma, \quad (\circ \rightarrow \circ) := \circ.$$

Assume a global assignment of a type variable ξ to every X .

$$\tau(P\vec{r}) := \tau(P),$$

$$\tau(A \rightarrow^c B) := (\tau(A) \rightarrow \tau(B)), \quad \tau(A \rightarrow^{\text{nc}} B) := \tau(B),$$

$$\tau(\forall_{x\rho}^c A) := (\rho \rightarrow \tau(A)), \quad \tau(\forall_{x\rho}^{\text{nc}} A) := \tau(A),$$

$$\tau(X) := \xi,$$

$$\tau(\{\vec{x} \mid A\}) := \tau(A),$$

$$\tau(\mu_X^{\text{nc}}(K_0, \dots, K_{k-1})) := \circ,$$

$$\tau(\underbrace{\mu_X(\forall_{\vec{x}_i}^{\text{nc}} \forall_{\vec{y}_i}^c (\vec{A}_i \rightarrow^{\text{nc}} \vec{B}_i \rightarrow^c X \vec{r}_i))}_{I})_{i < k} := \underbrace{\mu_\xi(\tau(\vec{y}_i) \rightarrow \tau(\vec{B}_i) \rightarrow \xi)}_{\iota_I})_{i < k}.$$

Call ι_I the **algebra associated with I** .

Realizability

Introduce a special **nullterm** symbol ε to be used as a “realizer” for n.c. formulas. Extend term application to ε by

$$\varepsilon t := \varepsilon, \quad t\varepsilon := t, \quad \varepsilon\varepsilon := \varepsilon.$$

Assume a global assignment giving for every predicate variable X of arity $\vec{\rho}$ a predicate variable X^r of arity $(\tau(X), \vec{\rho})$.

$$t \mathbf{r} X\vec{r} := X^r t\vec{r},$$

$$t \mathbf{r} (A \rightarrow^c B) := \forall_x (x \mathbf{r} A \rightarrow tx \mathbf{r} B) \quad \text{if } \text{lev}(\tau(A)) \neq 1,$$

$$t \mathbf{r} (A \rightarrow^c B) := \begin{cases} \forall_x (x \mathbf{r} A \rightarrow \dot{t}(x \upharpoonright m) \mathbf{r} B) \\ \text{and } m := \hat{t}x \text{ is a bar for } \dot{t} \text{ at } x \end{cases} \quad \text{if } \text{lev}(\tau(A)) = 1,$$

$$t \mathbf{r} (A \rightarrow^{\text{nc}} B) := \forall_x (x \mathbf{r} A \rightarrow t \mathbf{r} B),$$

$$t \mathbf{r} \forall_x^c A := \forall_x (tx \mathbf{r} A), \quad \text{if } \text{lev}(x) \neq 1,$$

$$t \mathbf{r} \forall_x^c A := \begin{cases} \forall_x (\dot{t}(x \upharpoonright m) \mathbf{r} A) \\ \text{and } m := \hat{t}x \text{ is a bar for } \dot{t} \text{ at } x \end{cases} \quad \text{if } \text{lev}(x) = 1,$$

$$t \mathbf{r} \forall_x^{\text{nc}} A := \forall_x (t \mathbf{r} A),$$

Realizability (continued)

$$t \mathbf{r} \underbrace{(\mu_X(K_0, \dots, K_{k-1}))}_I \vec{s} := I^{\mathbf{r}} t \vec{s}.$$

In case A is n.c., $\forall_x (x \mathbf{r} A \rightarrow B(x))$ means $\varepsilon \mathbf{r} A \rightarrow B(\varepsilon)$. For

$$I := \mu_X(\forall_{\vec{x}_i}^{\text{nc}} \forall_{\vec{y}_i}^{\text{c}} ((A_{i\nu})_{\nu < n_i} \rightarrow^{\text{nc}} (B_{i\nu})_{\nu < m_i} \rightarrow^{\text{c}} X \vec{r}_i))_{i < k}$$

define

$$I^{\mathbf{r}} := \{ w, \vec{x} \mid (\mu_X^{\text{nc}}(\forall_{\vec{x}_i, \vec{y}_i, \vec{v}_i}^{\text{nc}} ((\exists_{u_{i\nu}} u_{i\nu} \mathbf{r} A_{i\nu})_{\nu < n_i} \rightarrow^{\text{nc}} (v_{i\nu} \mathbf{r} B_{i\nu})_{\nu < m_i} \rightarrow X(C_i \vec{y}_i \vec{v}_i) \vec{r}_i))_{i < k}) w \vec{x} \}.$$

For a derivation M of a formula A we define its **extracted term** $\text{et}(M)$, of type $\tau(A)$. For M^A with A n.c. let $\text{et}(M^A) := \varepsilon$. Else

$$\text{et}(u^A) \quad := x_u^{\tau(A)} \quad (x_u^{\tau(A)} \text{ uniquely associated to } u^A),$$

$$\text{et}((\lambda_{u^A} M^B)^{A \rightarrow^c B}) := \lambda_{x_u}^{\tau(A)} \text{et}(M),$$

$$\text{et}((M^{A \rightarrow^c B} N^A)^B) := \text{et}(M) \text{et}(N),$$

$$\text{et}((\lambda_{x^\rho} M^A)^{\forall_x^c A}) := \lambda_x^\rho \text{et}(M),$$

$$\text{et}((M^{\forall_x^c A(x)} r)^{A(r)}) := \text{et}(M) r,$$

$$\text{et}((\lambda_{u^A} M^B)^{A \rightarrow^{\text{n.c.}} B}) := \text{et}(M),$$

$$\text{et}((M^{A \rightarrow^{\text{n.c.}} B} N^A)^B) := \text{et}(M),$$

$$\text{et}((\lambda_{x^\rho} M^A)^{\forall_x^{\text{n.c.}} A}) := \text{et}(M),$$

$$\text{et}((M^{\forall_x^{\text{n.c.}} A(x)} r)^{A(r)}) := \text{et}(M).$$

Here $\lambda_{x_u}^{\tau(A)} \text{et}(M)$ means $\text{et}(M)$ if A is n.c.

Extracted terms for the axioms.

- ▶ Let I be c.r.

$$\text{et}(I_i^+) := C_i, \quad \text{et}(I^-) := \mathcal{R},$$

where both C_i and $\mathcal{R}$ refer to the algebra ι_I associated with I .

- ▶ For the bar induction axiom the extracted term is the bar recursion operator.

Theorem (Soundness)

Let M be a derivation of A from assumptions $u_i: C_i$. Then we can derive $\text{et}(M) \text{ r } A$ from assumptions $x_{u_i} \text{ r } C_i$.

Proof. By induction on M . Usual proof works for the logical rules and for I_i^+ and I^- . It remains to consider bar induction.

- ▶ Spector's bar recursion
- ▶ Module enriched types
- ▶ Module enriched realizability
- ▶ Module enriched bar induction and bar recursion

- ▶ Every object s of list type $\rho^* := \mathbf{L}(\rho)$ can be viewed as a (partial) object of function type $\rho^\omega := \mathbf{N} \rightarrow \rho$.
- ▶ For a total $\dot{Y}$ of type $\rho^\omega \rightarrow \tau$ and a total sequence s of type ρ^* we can form the total object $\dot{Y}(s@_{\lambda_n}0)$ of type ρ .

Let Y of type $\text{level} \geq 2$ be a functional of module enriched type $\vec{\sigma}^+ \rightarrow \mathbf{N} \times \iota$. For example, consider the standard (i.e., not yet module enriched) type

$$\rho^* \rightarrow (\rho^\omega \rightarrow \iota) \rightarrow (\rho^* \rightarrow (\rho \rightarrow \iota) \rightarrow \iota) \rightarrow \iota. \quad (1)$$

with ρ a base type. Its module enrichment has type

$$\rho^* \rightarrow (\rho^\omega \rightarrow \mathbf{N} \times \iota) \rightarrow (\rho^* \rightarrow (\rho \rightarrow \iota) \rightarrow \mathbf{N} \times \iota) \rightarrow \mathbf{N} \times \iota. \quad (2)$$

The **bar recursion** operator Φ with standard type (1) now has module enriched type (2). Its object part satisfies

$$\Phi s Y H = \begin{cases} \dot{Y}(s@_{\lambda_n}0) & \text{if } |s| = \hat{Y}(s@_{\lambda_n}0) \\ \dot{H}(s, \lambda_x \dot{\Phi}(s * x, Y, H)) & \text{if } |s| < \hat{Y}(s@_{\lambda_n}0). \end{cases}$$

Let P be a predicate of arity (ρ^*) and $\tau(P) = \iota$ with ρ a base type.
 The axiom of **bar induction** is

$$\forall_s(\forall_f \exists_n P(\bar{f}n) \rightarrow \forall_s(\forall_x P(s * x) \rightarrow Ps) \rightarrow Ps).$$

The object part of its type is

$$\rho^* \rightarrow (\rho^\omega \rightarrow \mathbf{N} \times \iota) \rightarrow (\rho^* \rightarrow (\rho \rightarrow \iota) \rightarrow \mathbf{N} \times \iota) \rightarrow \iota.$$

Theorem

$$\mathring{\Phi} \mathbf{r} \forall_s(\forall_f \exists_n P(\bar{f}n) \rightarrow \forall_s(\forall_x P(s * x) \rightarrow Ps) \rightarrow Ps).$$

$$\mathring{\Phi} \mathbf{r} \forall_s (\forall_f \exists_n P(\bar{f}n) \rightarrow \forall_s (\forall_x P(s * x) \rightarrow Ps) \rightarrow Ps).$$

Proof. Unfolding the definition of $\mathbf{r}$ gives

$$\forall_s (\mathring{\Phi} s \mathbf{r} (\forall_f \exists_n P(\bar{f}n) \rightarrow \forall_s (\forall_x P(s * x) \rightarrow Ps) \rightarrow Ps))$$

$$\forall_{s,Y} (Y \mathbf{r} \forall_f \exists_n P(\bar{f}n) \rightarrow \mathring{\Phi} s Y \mathbf{r} (\forall_s (\forall_x P(s * x) \rightarrow Ps) \rightarrow Ps))$$

$$\forall_{s,Y,H} (Y \mathbf{r} \forall_f \exists_n P(\bar{f}n) \rightarrow H \mathbf{r} \forall_s (\forall_x P(s * x) \rightarrow Ps) \rightarrow \mathring{\Phi} s Y H \mathbf{r} Ps).$$

Fix Y, H with $Y \mathbf{r} \forall_f \exists_n P(\bar{f}n)$ and $H \mathbf{r} \forall_s (\forall_x P(s * x) \rightarrow Ps)$. Show

$$\forall_s (\mathring{\Phi} s Y H \mathbf{r} Ps)$$

by transfinite induction on the unsecured sequences of Y .

Base case. Recall that Y splits into its object part $\mathring{Y}$ and module part $\hat{Y}$. Assume $Y \mathbf{r} \forall_f \exists_n P(\bar{f}n)$, i.e.,

$$\forall_f (\mathring{Y}f \mathbf{r} P(\bar{f}(\hat{Y}f))) \quad (3)$$

with $\hat{Y}f =: m$ a bar for $\mathring{Y}$ at f , i.e., $\mathring{Y}f = \mathring{Y}((f \upharpoonright m)@h)$ for all h .

Assume s is a secured sequence for Y , i.e., $|s| = \hat{Y}(s@ \lambda_n 0)$. Then by definition $\mathring{\Phi}sYH = \mathring{Y}(s@ \lambda_n 0)$, and by (3)

$$\mathring{Y}(s@ \lambda_n 0) \mathbf{r} P(\overline{(s@ \lambda_n 0)}(\hat{Y}(s@ \lambda_n 0))) \quad \text{i.e.,} \quad \mathring{\Phi}sYH \mathbf{r} Ps.$$

Step case. Note that the second premise unfolds into

$$\begin{aligned} & \forall_s \dot{H}s \text{ r } (\forall_x P(s * x) \rightarrow Ps) \\ & \forall_{s,g} (g \text{ r } \forall_x P(s * x) \rightarrow \dot{H}sg \text{ r } Ps) \\ & \forall_{s,g} (\forall_x (gx \text{ r } P(s * x)) \rightarrow \dot{H}sg \text{ r } Ps). \end{aligned}$$

Assume s is an unsecured sequence for Y , i.e., $|s| < \hat{Y}(s@ \lambda_n 0)$.
Then by definition

$$\dot{\Phi}sYH = \dot{H}(s, \lambda_x \dot{\Phi}(s * x, Y, H)).$$

By induction hypothesis $\dot{\Phi}(s * x, Y, H) \text{ r } P(s * x)$. Thus $\lambda_x \dot{\Phi}(s * x, Y, H)$ is a g for which the unfolded second premise can be applied. Hence

$$\dot{H}(s, \lambda_x \dot{\Phi}(s * x, Y, H)) \text{ r } Ps \quad \text{i.e.,} \quad \dot{\Phi}sYH \text{ r } Ps.$$

References

- ▶ U. Berger and P. Oliva, Modified bar recursion. MSCS 2006.
- ▶ H.S. and S.S. Wainer, Proofs and Computations. Perspectives in Logic, ASL & Cambridge UP, 2012.
- ▶ M. Seisenberger, On the constructive content of proofs. Doctoral dissertation, LMU 2003.