Computational content of proofs involving coinduction

Helmut Schwichtenberg
(j.w.w. Kenji Miyamoto and Fredrik Nordvall Forsberg)

Mathematisches Institut, LMU, München

Kyoto University, 19. March 2014
Proof: 2 aspects

- provides insight (uniformity)
- may have computational content

Mathematics = logic + data + inductive definitions

- Logic: minimal, intro and elim for →, ∀
- Proof ~ lambda-term (Curry-Howard correspondence)
- Can embed classical and intuitionistic logic
Computable functionals

Arguments of any finite type, not only numbers and functions.

- **Principle of finite support.** If $H(\Phi)$ is defined with value $n$, then there is a finite approximation $\Phi_0$ of $\Phi$ such that $H(\Phi_0)$ is defined with value $n$.

- **Monotonicity principle.** If $H(\Phi)$ is defined with value $n$ and $\Phi'$ extends $\Phi$, then also $H(\Phi')$ is defined with value $n$.

- **Effectivity principle.** An object is computable iff its set of finite approximations is (primitive) recursively enumerable (or equivalently, $\Sigma^0_1$-definable).
Tokens, consistency and entailment at base types

Types

- Base types $\iota$: free algebras, given by constructors (e.g. 0, $S$).
- Function types: $\rho \rightarrow \sigma$.

Example: $\iota := D$ (derivations, or binary trees), by constructors $\circ$ (leaf, or nil) and $C : D \rightarrow D \rightarrow D$ (branch, or cons).

- Token $a^D$: $\circ, C \ast \circ, C \circ *, C(C \ast \circ) \circ$.
- $U^D := \{a_1, \ldots, a_n\}$ consistent if
  - all $a_i$ start with the same constructor,
  - (proper) tokens at $j$-th argument positions are consistent
    (example: $\{C \ast \circ, C \circ *\}$).
- $U^D \vdash a$ (entails) if
  - all $a_i \in U$ and $a$ start with the same constructor,
  - (proper) tokens at $j$-th argument positions of $a_i$ entail $j$-th argument of $a$ (example: $\{C \ast \circ, C \circ *\} \vdash C \circ \circ$).

An ideal $x^\rho$ is a (possibly infinite) set of tokens which is

- consistent and
- closed under entailment.
Tokens and entailment for \( \mathbb{N} \)

\[
\{ a \} \vdash b \text{ iff there is a path from } a \text{ (up) to } b \text{ (down).}
\]
Total and cototal ideals of base type

An ideal \( x' \) is **cototal** if every constructor tree \( P(\ast) \in x \) has a "\( \succsim_1 \)-predecessor" \( P(C_{\vec{\ast}}) \in x \); it is **total** if it is cototal and the relation \( \succsim_1 \) on \( x \) is well-founded.

Examples. **N:**

- Every total ideal is the deductive closure of a token \( S(S\ldots(S0)\ldots) \). The set of all tokens \( S(S\ldots(S\ast)\ldots) \) is a cototal ideal.

**D** (derivations):

- Total ideal \( \sim \) finite derivation.
- Cototal ideal \( \sim \) finite or infinite "locally correct" derivation [Mints 78].
- Arbitrary ideal \( \sim \) incomplete derivation, with "holes".
Tokens, consistency and entailment at function types

Ideals: partial continuous functionals $f^{\rho \to \sigma}$ (Scott, Ershov).

- Tokens of type $\rho \to \sigma$ are pairs $(U, a)$ with $U \in \text{Con}_{\rho}$.

- $\{ (U_i, a_i) \mid i \in I \} \in \text{Con}_{\rho \to \sigma}$ means

\[
\forall J \subseteq I (\bigcup_{j \in J} U_j \in \text{Con}_{\rho} \to \{ a_j \mid j \in J \} \in \text{Con}_{\sigma}).
\]

“Formal neighborhood”.

- $W \vdash_{\rho \to \sigma} (U, a)$ means $W U \vdash_\sigma a$, where application $W U$ of $W = \{ (U_i, a_i) \mid i \in I \}$ to $U$ is $\{ a_i \mid U \vdash_\rho U_i \}$.

Application of $f^{\rho \to \sigma}$ to $x^\rho$ is

\[
f(x) := \{ a^\sigma \mid \exists U \subseteq x (U, a) \in f \}.
\]

Principles of finite support and monotonicity hold.
Computable functionals

A partial continuous functional $f^\rho$ is **computable** if it is a (primitive) recursively enumerable set of tokens.

How to define computable functionals? By computation rules

$$D\vec{P}_i(\vec{y}_i) = M_i \quad (i = 1, \ldots, n)$$

with free variables of $\vec{P}_i(\vec{y}_i)$ and $M_i$ among $\vec{y}_i$, where $\vec{P}_i(\vec{y}_i)$ are “constructor patterns”.

**Terms** (a common extension of Gödel’s T and Plotkin’s PCF)

$$M, N ::= x^\rho \mid C^\rho \mid D^\rho \mid (\lambda x^\rho M^\sigma)^{\rho \rightarrow \sigma} \mid (M^{\rho \rightarrow \sigma} N^\rho)^\sigma.$$
Examples

\[ + : \mathbb{N} \rightarrow \mathbb{N} \rightarrow \mathbb{N} \] defined by

\[ n + 0 = n, \]
\[ n + Sm = S(n + m). \]

\[ Y : (\tau \rightarrow \tau) \rightarrow \tau \] defined by

\[ Yf = f(Yf). \]

\[ R_{\mathbb{N}}^\tau : \mathbb{N} \rightarrow \tau \rightarrow (\mathbb{N} \rightarrow \tau \rightarrow \tau) \rightarrow \tau \] defined by

\[ R_{\mathbb{N}}^\tau 0xf = x, \]
\[ R_{\mathbb{N}}^\tau (S\,n)xf = fx(R_{\mathbb{N}}^\tau nxf). \]

Reduction (including \( \beta \), \( \eta \)) is non-terminating, but confluent.
Denotational semantics

How to use computation rules to define a computable functional?
Inductively define \((\vec{U}, a) \in [\lambda \vec{x} M] \ (\text{FV}(M) \subseteq \{\vec{x}\})\).

Case \(\lambda_{\vec{x},y,z} M\) with \(\vec{x}\) free in \(M\), but not \(y\).

\[
\frac{\text{Case } \lambda_{\vec{x},y,z} M \text{ with } \vec{x} \text{ free in } M, \text{ but not } y.}{(\vec{U}, \vec{W}, a) \in [\lambda_{\vec{x},z} M]} \quad (K).
\]

Case \(\lambda_{\vec{x}} M\) with \(\vec{x}\) the free variables in \(M\).

\[
\frac{\text{Case } \lambda_{\vec{x}} M \text{ with } \vec{x} \text{ the free variables in } M.}{U \vdash a \quad (U, a) \in [\lambda_{\vec{x}} x]} \quad (V), \quad (\vec{U}, V, a) \in [\lambda_{\vec{x}} M] \quad (\vec{U}, V) \subseteq [\lambda_{\vec{x}} N] \quad (A).
\]

For every constructor \(C\) and defined constant \(D\):

\[
\frac{\vec{U} \vdash a^* \quad (\vec{U}, Ca^*) \in [C]} \quad (C), \quad (\vec{V}, a) \in [\lambda_{\vec{x}} M] \quad \vec{U} \vdash \vec{P}(\vec{V}) \quad (D),
\]

with one rule \((D)\) for every defining equation \(D \vec{P}(\vec{x}) = M\).
Properties of the denotational semantics

- The value is preserved under standard $\beta, \eta$-conversion and the computation rules.

- An adequacy theorem holds: whenever a closed term $M^\iota$ has a token $a \in P(\vec{V})$ in its denotation $\llbracket M \rrbracket$, then $M$ head reduces to a constructor term entailing $a$. 
A theory of computable functionals (TCF)

A variant of HA$^\omega$.

Formulas $A$ and predicates $P$ are defined simultaneously

$$A, B ::= P \vec{r} \mid A \to B \mid \forall X A$$

$$P ::= X \mid \{ \vec{x} \mid A \} \mid I \quad (I \text{ inductively defined}).$$

$\forall X A$ not allowed, since this would be impredicative: in the predicate existence axiom $P ::= \{ \vec{x} \mid A \}$ the formula $A$ could contain quantifiers with the newly created $P$ in its range.

$\forall x^p A$ is unproblematic: no such existence axioms.
Brouwer - Heyting - Kolmogorov

Have $\rightarrow^\pm$, $\forall^\pm$, $I^\pm$. **BHK-interpretation:**

- $p$ proves $A \rightarrow B$ iff $p$ is a construction transforming any proof $q$ of $A$ into a proof $p(q)$ of $B$.
- $p$ proves $\forall x \rho A(x)$ iff $p$ is a construction such that for all $a^\rho$, $p(a)$ proves $A(a)$.

Leaves open:

- What is a “construction”?
- What is a proof of a prime formula?

**Proposal:**

- Construction: computable functional.
- Proof of a prime formula $I\bar{r}$: generation tree.

Example: generation tree for $\text{Even}(6)$ should consist of a single branch with nodes $\text{Even}(0)$, $\text{Even}(2)$, $\text{Even}(4)$ and $\text{Even}(6)$.
The type \( \tau(A) \) of a formula \( A \)

Distinguish non-computational (n.c.) (or Harrop) and computationally relevant (c.r.) formulas. Example:

- \( r = s \) is n.c.
- \( \text{Even}(n) \) is c.r.

Extend the use of \( \rho \rightarrow \sigma \) to the “nulltype symbol” \( \circ \):

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

Define the type \( \tau(A) \) of a formula \( A \) by

\[
\tau(I \vec{r}) = \begin{cases} 
\iota_I & \text{if } I \text{ is c.r.}, \\
\circ & \text{if } I \text{ is n.c.},
\end{cases}
\]

\[
\tau(A \rightarrow B) := \tau(A) \rightarrow \tau(B),
\]

\[
\tau(\forall x \rho A) := \rho \rightarrow \tau(A)
\]

with \( \iota_I \) associated naturally with \( I \).
Realizability

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

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

Definition ($t \text{ r } A$, $t$ realizes $A$)

Let $A$ be a formula and $t$ either a term of type $\tau(A)$ if the latter is a type, or the nullterm symbol $\varepsilon$ for n.c. $A$.

$$t \text{ r } l\breve{s} := \begin{cases} l^r t\breve{s} & \text{if } l \text{ is c.r. (} l^r \text{ inductively defined)}, \\ l\breve{s} & \text{if } l \text{ is n.c.}, \end{cases}$$

$$t \text{ r } (A \rightarrow B) := \forall x (x \text{ r } A \rightarrow tx \text{ r } B),$$

$$t \text{ r } \forall x A := \forall x (tx \text{ r } A).$$
Extracted terms, soundness theorem

For a derivation $M$ of a formula $A$ 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

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

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

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

$$
\text{et}((\lambda x^\rho M^A)^{\forall x^A}) := \lambda x^\rho \text{et}(M),
$$

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

Extracted terms for the axioms: let $I$ be c.r.

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

where both the constructor $C_i$ and the recursion operator $\mathcal{R}$ refer to the algebra $\iota_I$ associated with $I$.

**Soundness.** Let $M$ be a derivation of $A$ from assumptions $u_i : C_i$. Then we can derive $\text{et}(M) \ r A$ from assumptions $x_{u_i} \ r C_i$. 
Relation of TCF to type theory

- Main difference: partial functionals are first class citizens.
- “Logic enriched”: Formulas and types kept separate.
- Minimal logic: $\to, \forall$ only. $x = y$ (Leibniz equality), $\exists$, $\lor$, $\land$ inductively defined (Martin-Löf).
- $\bot := (\text{False} = \text{True})$. Ex-falso-quodlibet: $\bot \rightarrow A$ provable.
- “Decorations” $\rightarrow^{\text{nc}}$, $\forall^{\text{nc}}$ (i) allow abstract theory (ii) remove unused data.
Case study: uniformly continuous functions (U. Berger)

- Formalization of an abstract theory of (uniformly) continuous real functions $f : I \rightarrow I$ $(I := [-1, 1])$.
- Let $Cf$ express that $f$ is a continuous real function. Assume the abstract theory proves

$$Cf \rightarrow \forall \, n \exists \, m \forall \, a \exists \, b (f[I_{a,m}] \subseteq I_{b,n}) \quad \text{with } I_{b,n} := [b - \frac{1}{2n}, b + \frac{1}{2n}]$$

Then

- $n \mapsto m$ modulus of (uniform) continuity ($\omega$)
- $n, a \mapsto b$ approximating rational function ($h$)
Read_X and its witnesses

Inductively define a predicate Read_X of arity (φ) by the clauses

\[ \forall^{nc}_f \forall_d (f[I] \subseteq I_d \rightarrow X(Out_d \circ f) \rightarrow \text{Read}_X f), \quad (\text{Read}_X)_0^+ \]

\[ \forall^n_c (\text{Read}_X (f \circ \text{In}_{-1}) \rightarrow \text{Read}_X (f \circ \text{In}_0) \rightarrow \text{Read}_X (f \circ \text{In}_1) \rightarrow \text{Read}_X f). \quad (\text{Read}_X)_1^+ \]

where \( l_d = \left[ \frac{d-1}{2}, \frac{d+1}{2} \right] \) (\( d \in \{-1, 0, 1\} \)) and

\[ (\text{Out}_d \circ f)(x) := 2f(x) - d, \quad (f \circ \text{In}_d)(x) := f\left(\frac{x + \frac{d}{2}}{2}\right). \]

Witnesses for Read_X f: total ideals in

\[ R_\alpha := \mu_\xi(\text{Put}^{SD \rightarrow \alpha \rightarrow \xi}, \text{Get}^{\xi \rightarrow \xi \rightarrow \xi \rightarrow \xi}) \]

where SD := \{-1, 0, 1\}. 

Write, $\text{coWrite}$ and its witnesses

Nested inductive definition of a predicate $\text{Write}$ of arity ($\varphi$):

$$\text{Write}(\text{Id}), \quad \forall^n_{nc}(\text{Read}_\text{Write} f \rightarrow \text{Write} f) \quad (\text{Id identity function}).$$

Witnesses for $\text{Write} f$: total ideals in

$$\mathcal{W} := \mu_\xi (\text{Stop}_\xi, \text{Cont}^{R_\xi \rightarrow \xi}).$$

Define $\text{coWrite}$, a companion predicate of $\text{Write}$, by

$$\forall^n_{nc}(\text{coWrite} f \rightarrow f = \text{Id} \lor \text{Read}_{\text{coWrite}} f). \quad (\text{coWrite})^-$$

Witnesses for $\text{coWrite} f$: $\mathcal{W}$-cototal $R_{\mathcal{W}}$-total ideals $t$. 
**W-cototal** $R_W$-total ideals are possibly non well-founded trees $t$:

- Get-Put-part: well-founded,
- **Stop-Cont**-part: not necessarily well-founded.
W-cototal $\mathbb{R}_W$-total ideals as stream transformers

View them as read-write machines.

- Start at the root of the tree.
- At node $\text{Put}_d t$, output the digit $d$, carry on with the tree $t$.
- At node $\text{Get } t_{-1} t_0 t_1$, read a digit $d$ from the input stream and continue with the tree $t_d$.
- At node Stop, return the rest of the input unprocessed as output.
- At node Cont $t$, continue with the tree $t$.

Output might be infinite, but $\mathbb{R}_W$-totality ensures that the machine can only read finitely many input digits before producing another output digit.

The machine represents a continuous function.
Cf implies $\text{coWrite } f$: informal proof

The greatest-fixed-point axiom ($\text{coWrite}^+$ (coinduction)) is

$$\forall_{nc}^f (Q f \rightarrow \forall_{nc}^f (Q f \rightarrow f = \text{Id} \vee \text{Read}_{\text{coWrite}} \vee Q f) \rightarrow \text{coWrite } f).$$

**Theorem** [Type-1 u.c.f. into type-0 u.c.f.]. $\forall_{nc}^f (C f \rightarrow \text{coWrite } f)$.

**Proof.** Assume $C f$. Use ($\text{coWrite}^+$ with competitor $C$. Suffices $\forall_{nc}^f (C f \rightarrow f = \text{Id} \vee \text{Read}_{\text{coWrite}} \vee C f)$. Assume $C f$, in particular $B_{m,2} f := \forall_a \exists_b (f[I_a,m] \subseteq I_{b,2})$ for some $m$. Get rhs by Lemma 1.

**Lemma 1.** $\forall_m \forall_{nc}^f (B_{m,2} f \rightarrow C f \rightarrow \text{Read}_{\text{coWrite}} \vee C f)$.

**Proof.** Induction on $m$, using Lemma 2 in the base case.

**Lemma 2** [FindSD]. $\forall_{nc}^f (B_{0,2} f \rightarrow \exists_d (f[I] \subseteq l_d))$.

**Proof.** Assume $B_{0,2} f$. Then $f[I_{0,0}] \subseteq I_{b,2}$ for some $b$, by definition of $B_{n,m}$. Have $b \leq -\frac{1}{4}, -\frac{1}{4} \leq b \leq \frac{1}{4}$ or $\frac{1}{4} \leq b$. Can determine either of $I_{b,2} \subseteq l_{-1}, I_{b,2} \subseteq l_0$ or $I_{b,2} \subseteq l_1$, hence $\exists_d (f[I] \subseteq l_d)$. 

23 / 29
([oh](CoRec (nat=>nat@(rat=>rat))=>algwrite)oh
([oh0]Inr((Rec nat=>..[type]..)
  left(oh0(Succ(Succ Zero))))
([g,oh1] [let sd (cFindSd(g 0))
  (Put sd
   (InR([n]left(oh1(Succ n))@
    ([a]2*right(oh1(Succ n))a-SDToInt sd))))]
([n,st,g,oh1]
  Get
  (st([a]g((a+IntN 1)/2))
   ([n0]left(oh1 n0)@
    ([a]right(oh1 n0)((a+IntN 1)/2)))))
  (st([a]g(a/2))([n0]left(oh1 n0)@
    ([a]right(oh1 n0)(a/2)))))
  (st([a]g((a+1)/2))([n0]left(oh1 n0)@
    ([a]right(oh1 n0)((a+1)/2)))))
right(oh0(Succ(Succ Zero))))
oh0))
Corecursion

The rules for $\mathcal{R}$ work from the leaves towards the root, and terminate because total ideals are well-founded.

For cototal ideals a similar operator defines functions with cototal ideals as values: **corecursion**. Consider $\iota = \mu_\xi(\kappa_0, \ldots, \kappa_{k-1})$.

**constructor type:**
\[
\sum_{i<k} \prod_{\nu<n_i} \rho_{i\nu}(\iota) \rightarrow \iota
\]

**decorator type:**
\[
\iota \rightarrow \sum_{i<k} \prod_{\nu<n_i} \rho_{i\nu}(\iota)
\]

**type of recursion operator:**
\[
\iota \rightarrow (\sum_{i<k} \prod_{\nu<n_i} \rho_{i\nu}(\iota \times \tau) \rightarrow \tau) \rightarrow \tau
\]

**type of corecursion operator:**
\[
\tau \rightarrow (\tau \rightarrow \sum_{i<k} \prod_{\nu<n_i} \rho_{i\nu}(\iota + \tau)) \rightarrow \iota
\]
Examples

\[\text{co} \mathcal{R}^\tau_N : \tau \to (\tau \to U + (N + \tau)) \to N,\]
\[\text{co} \mathcal{R}^\tau_{L(\rho)} : \tau \to (\tau \to U + \rho \times (L(\rho) + \tau)) \to L(\rho).\]

For \(f : \rho \to \tau, \ g : \sigma \to \tau\) define \([f, g]^{\rho + \sigma \to \tau} := \lambda_x (\mathcal{R}^\tau_{\rho + \sigma} x f g)\). Let \(x_1, x_2\) denote the two projections of \(x\) of type \(\rho \times \sigma\).

\[\text{co} \mathcal{R}^\tau_N NM = [\lambda_0, \lambda_x (S([\text{id}^N \to N, \lambda_y (\text{co} \mathcal{R}^\tau_N y M) x])]) (MN),\]
\[\text{co} \mathcal{R}^\tau_{L(\rho)} NM = [\lambda_\text{Nil}, \lambda_x (x_1 :: [\text{id}, \lambda_y (\text{co} \mathcal{R}^\tau_{L(\rho)} y M)] x_2]) (MN).\]
Corecursion for \( W \)

The corecursion operator \( \text{co}R^\tau_W \) has type

\[
\tau \to (\tau \to U + R_{W+\tau}) \to W.
\]

Conversion rule

\[
\text{co}R^\tau_W NM \mapsto [\text{case } (MN)^{U+R(W+\tau)} \text{ of } \\
\quad \text{DummyL } \mapsto \text{Stop } | \\
\quad \text{Inr } x \mapsto \text{Cont}(\mathcal{M}_W^{R(W+\tau)}(\lambda p [\text{case } p^{W+\tau} \text{ of } \\
\quad \quad \text{InL } y^W \mapsto y | \\
\quad \quad \text{InR } z^\tau \mapsto \text{co}R^\tau_W zM]) \\
\quad x^{R(W+\tau)} ]
\]

with \( \mathcal{M} \) a "map"-operator.

- Here \( \tau \) is \( N \to N \times (Q \to Q) \), for pairs of \( \omega: N \to N \) and \( h: N \to Q \to Q \) (variable name \( oh \)).
- No termination; translate into Haskell for evaluation.
Conclusion

TCF (theory of computable functionals) as a possible foundation for exact real arithmetic.

- Simply typed theory, with “lazy” free algebras as base types (⇒ constructors are injective and have disjoint ranges).
- Variables range over partial continuous functionals.
- Constants denote computable functionals (:= r.e. ideals).
- Minimal logic (→, ∀), plus inductive & coinductive definitions.
- Computational content in abstract theories.
- Decorations (→, ∀ and →^{nc}, ∀^{nc}) for fine-tuning.
References

- U. Berger, From coinductive proofs to exact real arithmetic. CSL 2009.
- K. Miyamoto and H.S., Program extraction in exact real arithmetic. To appear, MSCS.
- K. Miyamoto, F. Nordvall Forsberg and H.S., Program extraction from nested definitions. ITP 2013.