A theory of computable functionals

Helmut Schwichtenberg

Mathematisches Institut, LMU, München

University of Canterbury, Christchurch, 12 Feb 2016
Formulas and predicates
A theory of computable functionals
Brouwer - Heyting - Kolmogorov and decorations
The type of a formula or predicate
Realizability
Extracted terms
Simultaneously define formulas and predicates

\[ A, B ::= P\vec{r} \mid A \to B \mid \forall_x A, \]
\[ P, Q ::= X \mid \{ \vec{x} \mid A \} \mid \mu X (\forall_{\vec{x}}((A_{i\nu})_{\nu<n_i} \to X\vec{r}_i))_{i<k} \]

Need restriction: \( X \) at most strictly positive in \( A_{i\nu} \).
$$T_N := \mu X (X_0, \forall n (X_n \rightarrow X(Sn)))$$,

$$\text{Even} := \mu X (X_0, \forall n (X_n \rightarrow X(S(Sn))))$$,

$$\text{Eq} := \mu X (\forall_x X_{xx})$$,

$$\text{Ex}_Y := \mu X (\forall_x (Y_x \rightarrow X))$$,

$$\text{Cap}_{Y,Z} := \mu X (\forall_{\vec{x}} (Y_{\vec{x}} \rightarrow Z_{\vec{x}} \rightarrow X_{\vec{x}}))$$,

$$\text{Cup}_{Y,Z} := \mu X (\forall_{\vec{x}} (Y_{\vec{x}} \rightarrow X_{\vec{x}}), \forall_{\vec{x}} (Z_{\vec{x}} \rightarrow X_{\vec{x}}))$$.

**Abbreviations**

$$\exists_x A := \text{Ex}_{\{x|A\}}$$,

$$P \cap Q := \text{Cap}_{P,Q}$$,

$$P \cup Q := \text{Cup}_{P,Q}$$.
- Formulas and predicates
- A theory of computable functionals
- Brouwer - Heyting - Kolmogorov and decorations
- The type of a formula or predicate
- Realizability
- Extracted terms
Relation to type theory

- Main difference: partial functionals are first class citizens.
- “Logic enriched”: Formulas and types kept separate.
- Minimal logic: →, ∀ only. \(\text{Eq}(x, y)\) (Leibniz equality), \(\exists\), \(\lor\), \(\land\) inductively defined (Russell, Martin-Löf).
- \(\mathbf{F} := \text{Eq}(\text{ff}, \text{tt})\). Ex-falso-quodlibet: \(\mathbf{F} \rightarrow A\) provable.
- “Decorations” \(\rightarrow^{nc}\), \(\forall^{nc}\) (i) allow abstract theory (ii) remove unused data.
Theory of computable functionals TCF

Typed variables, ranging over the partial continuous functionals. Minimal logic, with intro and elim for $\rightarrow$ and $\forall$. Axioms:

- $I^+_i : \forall \bar{x}((A_\nu(I))_{\nu<n} \rightarrow I\bar{r})$
- $I^- : \forall \bar{x}(I\bar{x} \rightarrow (\forall \bar{x}_i((A_{i\nu}(I \cap X))_{\nu<n_i} \rightarrow X\bar{r}_i))_{i<k} \rightarrow X\bar{x})$

Induction $= \text{elimination for totality over } \mathbb{N}$.

$T^-_N : \forall_n(T_N n \rightarrow X0 \rightarrow \forall_n(T_N n \rightarrow Xn \rightarrow X(Sn)) \rightarrow Xn)$.

Remarks

- Every “competitor” $X$ satisfying the clauses contains $T_N$.
- Induction for $\mathbb{N}$, which only holds for total numbers.
- Fits the logical elimination rules (main part comes first).
- “Strengthened” step formula $\forall_n(T_N n \rightarrow Xn \rightarrow X(Sn))$. 
For nullary predicates $P = \{ A \}$ and $Q = \{ B \}$ we write $A \land B$ for $P \cap Q$ and $A \lor B$ for $P \cup Q$. Introduction axioms:

\[
\forall_x (A \rightarrow \exists_x A),
\]

\[
A \rightarrow B \rightarrow A \land B,
\]

\[
A \rightarrow A \lor B, \quad B \rightarrow A \lor B.
\]

Elimination axioms:

\[
\exists_x A \rightarrow \forall_x (A \rightarrow B) \rightarrow B \quad (x \notin \text{FV}(B)),
\]

\[
A \land B \rightarrow (A \rightarrow B \rightarrow C) \rightarrow C,
\]

\[
A \lor B \rightarrow (A \rightarrow C) \rightarrow (B \rightarrow C) \rightarrow C.
\]
Equalities

(i) Defined function constants $D$ introduced by computation rules, written $\ell = r$, but intended as left-to-right rewrites.

(ii) Leibniz equality $\text{Eq}$ (inductively defined).

(iii) Pointwise equality between partial continuous functionals, defined inductively as well.

(iv) If $\ell$ and $r$ have a finitary algebra as their type, $\ell = r$ by (i) is a boolean term. Take $\text{Eq}((\ell = r)^B, \text{tt})$.

In TCF formulas $A(r)$ and $A(s)$ are identified if $r, s \in T^+$ have a common reduct.
\[\text{Eq}^+: \forall_x \text{Eq}(x^\rho, x^\rho)\]
\[\text{Eq}^-: \forall_{x, y} (\text{Eq}(x, y) \rightarrow \forall_x X_{x} \rightarrow X_{y}).\]

**Compatibility of Eq:** \(\forall_{x, y} (\text{Eq}(x, y) \rightarrow A(x) \rightarrow A(y)).\)

(Use \(\text{Eq}^-\) with \(\{ x, y \mid A(x) \rightarrow A(y) \}\) for \(X\).)
Define falsity by \( F := \text{Eq}(\text{ff}, \text{tt}) \).

**Ex-falso-quodlibet:** TCF \( \vdash F \rightarrow A \) where \( A \) has no strictly positive occurrences of (i) predicate variables (ii) inductive predicates without nullary clauses.

**Proof.**

1. Show \( F \rightarrow \text{Eq}(x^\rho, y^\rho) \).

   \[
   \begin{align*}
   \text{Eq}(\mathcal{R}_B^\rho \text{ff}xy, \mathcal{R}_B^\rho \text{ff}xy) & \quad \text{by } \text{Eq}^+ \\
   \text{Eq}(\mathcal{R}_B^\rho \text{tt}xy, \mathcal{R}_B^\rho \text{ff}xy) & \quad \text{by compatibility from } \text{Eq}(\text{ff}, \text{tt}) \\
   \text{Eq}(x^\rho, y^\rho) & \quad \text{by conversion.}
   \end{align*}
   \]

2. Show \( F \rightarrow A \), by induction on \( A \). **Case \( \vec{s} \).**

   Let \( K_0 \) be the nullary clause, with final conclusion \( I\vec{t} \).

   By IH from \( F \) we can derive all parameter premises, hence \( I\vec{t} \).

   From \( F \) we also have \( \text{Eq}(s_i, t_i) \) by 1.

   Hence \( I\vec{s} \) by compatibility.

   The cases \( A \rightarrow B \) and \( \forall x A \) are obvious. \( \square \)
Formulas and predicates
A theory of computable functionals
Brouwer - Heyting - Kolmogorov and decorations
The type of a formula or predicate
Realizability
Extracted terms
Have $\rightarrow^\pm$, $\forall^\pm$, $I^\pm$. BHK-interpretation:

- $p$ proves $A \rightarrow B$ if and only if $p$ is a construction transforming any proof $q$ of $A$ into a proof $p(q)$ of $B$.
- $p$ proves $\forall_x \rho A(x)$ if and only if $p$ 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 $I\vec{r}$: 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 $\text{Even}(6)$. 
Decoration

Which of the variables $\vec{x}$ and assumptions $\vec{A}$ are actually used in the “solution” provided by a proof of

$$\forall\vec{x}(\vec{A} \rightarrow I\vec{r})?$$

To express this we split each of $\rightarrow, \forall$ into two variants:

- a “computational” one $\rightarrow^c, \forall^c$ and
- a “non-computational” one $\rightarrow^{nc}, \forall^{nc}$ (with restricted rules)

and consider

$$\forall^{nc}_x \forall^c_y (\vec{A} \rightarrow^{nc} \vec{B} \rightarrow^c X\vec{r}).$$

This will lead to a different (simplified) algebra $\nu_I$ associated with the inductive predicate $I$. 
Each inductive predicate is marked as computationally relevant (c.r.) or non-computational (n.c., or Harrop): $\mu^\text{nc}_X(K_0, \ldots, K_{k-1})$. Then the elimination scheme must be restricted to n.c. formulas.

We usually write $\rightarrow, \forall, \mu$ for $\rightarrow^c, \forall^c, \mu^c$. Notice that in the clauses of an n.c. inductive predicate $\mu^\text{nc}_X\overrightarrow{K}$ decorations play no role.

For the even numbers we now have two variants:

Even := $\mu_X(X_0, \forall^\text{nc}_n(X_n \rightarrow X(S(Sn))))$,

Even$^\text{nc}$ := $\mu^\text{nc}_X(X_0, \forall_n(X_n \rightarrow X(S(Sn))))$.

Generally for every c.r. inductive predicate $I$ (i.e., defined as $\mu_X\overrightarrow{K}$) we have an n.c. variant $I^\text{nc}$ defined as $\mu^\text{nc}_X\overrightarrow{K}$.
\[
\text{ExD}_Y := \mu_X (\forall_x (Yx \rightarrow X)), \\
\text{ExL}_Y := \mu_X (\forall_x (Yx \rightarrow^{\text{nc}} X)). \\
\text{ExR}_Y := \mu_X (\forall_x^{\text{nc}} (Yx \rightarrow X)), \\
\text{ExU}_Y := \mu_X^{\text{nc}} (\forall_x (Yx \rightarrow X)).
\]

D for “double”, L for “left”, R for “right”, U for “uniform”. Abbreviations

\[
\exists^d_x A := \text{ExD}_{\{x|A\}}, \\
\exists^l_x A := \text{ExL}_{\{x|A\}}, \\
\exists^r_x A := \text{ExR}_{\{x|A\}}, \\
\exists^u_x A := \text{ExU}_{\{x|A\}}.
\]
CupD_{Y,Z} := \mu_X (Y \to X, \ Z \to X),
CupL_{Y,Z} := \mu_X (Y \to X, \ Z \to \text{nc} \ X),
CupR_{Y,Z} := \mu_X (Y \to \text{nc} \ X, \ Z \to X),
CupU_{Y,Z} := \mu_X (Y \to \text{nc} \ X, \ Z \to \text{nc} \ X),
CupNC_{Y,Z} := \mu_X (Y \to X, \ Z \to X).

The final nc-variant suppresses even the information which clause has been used. Abbreviations

A \lor^d B := \text{CupD}_{\{|A\},\{|B\}},
A \lor^l B := \text{CupL}_{\{|A\},\{|B\}},
A \lor^r B := \text{CupR}_{\{|A\},\{|B\}},
A \lor^u B := \text{CupU}_{\{|A\},\{|B\}},
A \lor^{nc} B := \text{CupNC}_{\{|A\},\{|B\}}.
Formulas and predicates
A theory of computable functionals
Brouwer - Heyting - Kolmogorov and decorations
The type of a formula or predicate
Realizability
Extracted terms
Examples

Let \( a, b \in \mathbb{Q}, \ x \in \mathbb{R}, \ k \in \mathbb{Z}, \ f \in \mathbb{R} \rightarrow \mathbb{R}. \)

- \( \forall_{a,b,x}(a < b \rightarrow x \leq b \lor^{u} a \leq x) \) has type \( \mathbb{Q} \rightarrow \mathbb{Q} \rightarrow \mathbb{R} \rightarrow \mathbb{B}. \)

- \( \forall_{a,b,x}(a < b \rightarrow x < b \lor^{d} a < x) \) has type \( \mathbb{Q} \rightarrow \mathbb{Q} \rightarrow \mathbb{R} \rightarrow \mathbb{Z} + \mathbb{Z}. \)

- The formula

\[
\forall_{f,k}(f(0) \leq 0 \leq f(1) \rightarrow \forall_{a,b}\left(\frac{1}{2k}|b - a| \leq |f(b) - f(a)|\right) \rightarrow \exists_{x}f(x)=0)
\]

has type \( (\mathbb{R} \rightarrow \mathbb{R}) \rightarrow \mathbb{Z} \rightarrow \mathbb{R}. \)
The type $\tau(C)$ of a formula or predicate $C$

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

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

Assume a global injective assignment of a type variable $\xi$ to every c.r. predicate variable $X$. Let $\tau(C) := \circ$ if $C$ is non-computational. In case $C$ is c.r. let

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

$\tau(A \to B) := (\tau(A) \to \tau(B))$, \quad $\tau(A \to^{nc} B) := \tau(B)$,

$\tau(\forall_{\chi \rho} A) := (\rho \to \tau(A))$, \quad $\tau(\forall^{nc}_{\chi \rho} A) := \tau(A)$,

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

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

$\tau(\mu_{\chi}(\forall_{\vec{x_i}}^{nc} \forall_{\vec{y_i}}(\vec{A_i} \to^{nc} \vec{B_i} \to X\vec{r_i}))_{i<k}) := \mu_{\xi}(\tau(\vec{y_i}) \to \tau(\vec{B_i}) \to \xi)_{i<k}.$

$\nu_I$ is the algebra associated with $I$. 
Formulas and predicates
A theory of computable functionals
Brouwer - Heyting - Kolmogorov and decorations
The type of a formula or predicate
Realizability
Extracted terms
Realizability

For every predicate or formula \( C \) we define an n.c. predicate \( C^r \).

For n.c. \( C \) let

\[
C^r := C.
\]

In case \( C \) is c.r. the arity of \( C^r \) is \( (\tau(C), \vec{\sigma}) \) with \( \vec{\sigma} \) the arity of \( C \).

For c.r. formulas define

\[
(P^r) := \{ u \mid P^ru^r \}
\]

\[
(A \rightarrow B)^r := \begin{cases} 
\{ u \mid \forall v(A^rv \rightarrow B^ruv) \} & \text{if } A \text{ is c.r.} \\
\{ u \mid A \rightarrow B^ru \} & \text{if } A \text{ is n.c.}
\end{cases}
\]

\[
(A \rightarrow^\text{nc} B)^r := \{ u \mid A \rightarrow B^ru \}
\]

\[
(\forall_x A)^r := \{ u \mid \forall_x A^r(ux) \}
\]

\[
(\forall^\text{nc}_x A)^r := \{ u \mid \forall_x A^ru \}.
\]

For c.r. predicates: given n.c. \( X^r \) for all predicate variables \( X \).

\[
\{ \vec{x} \mid A \}^r := \{ u, \vec{x} \mid A^ru \}.
\]
Consider a c.r. inductive predicate

\[ I := \mu X (\forall_{\vec{x}}^{c/nc} ((A_{i\nu})_{\nu<n_i} \rightarrow^{c/nc} X \vec{r}_i))_{i<k}. \]

\( \vec{Y} \): all predicate variables strictly positive in some \( A_{i\nu} \) except \( X \).
Define the witnessing predicate with free predicate variables \( \vec{Y}^r \) by

\[ I^r := \mu_{X^r}^{nc}(\forall_{\vec{x},\vec{u}} ((A^r_{i\nu} u_{i\nu})_{\nu<n_i} \rightarrow X^r(C_i \vec{x}_i \vec{u}_i) \vec{r}_i))_{i<k} \]

with the understanding that

(i) \( u_{i\nu} \) occurs only when \( A_{i\nu} \) is c.r., and it occurs as an argument in \( C_i \vec{x}_i \vec{u}_i \) only if \( A_{i\nu} \) is c.r. and followed by \( \rightarrow \), and

(ii) only those \( x_{ij} \) with \( \forall_{x_{ij}}^c \) occur as arguments in \( C_i \vec{x}_i \vec{u}_i \).

We write \( u \models A \) for \( A^r u \) (\( u \) realizes \( A \)).
For the even numbers we obtain

\[
\text{Even} := \mu_X (X_0, \forall_{n}^\text{nc} (X_n \rightarrow X(S(Sn)))) \\
\text{Even}^r := \mu_{X^r} (X^r_0, \forall_{n,m} (X^r_{mn} \rightarrow X^r(Sm)(S(Sn)))).
\]

**Axiom** (Invariance under realizability)

\[
\text{Inv}_A : A \leftrightarrow \exists^1_u (u \ r A) \quad \text{for c.r. formulas } A.
\]

**Lemma**

*For c.r. formulas* \(A\) *we have*

\[
(\lambda_u u) \ r (A \rightarrow \exists^1_u (u \ r A)), \\
(\lambda_u u) \ r (\exists^1_u (u \ r A) \rightarrow A).
\]
From the invariance axioms we can derive

**Theorem (Choice)**

\[ \forall_x \exists_y^1 A(y) \rightarrow \exists_f^1 \forall_x A(fx) \quad \text{for } A \text{ n.c.} \]

\[ \forall_x \exists_y^d A(y) \rightarrow \exists_f^d \forall_x A(fx) \quad \text{for } A \text{ c.r.} \]

**Theorem (Independence of premise).** Assume \( x \notin \text{FV}(A) \).

\( (A \rightarrow \exists_x^1 B) \rightarrow \exists_x^1 (A \rightarrow B) \quad \text{for } A, B \text{ n.c.} \)

\( (A \rightarrow_{\text{nc}}^1 \exists_x^1 B) \rightarrow \exists_x^1 (A \rightarrow B) \quad \text{for } B \text{ n.c.} \)

\( (A \rightarrow \exists_x^d B) \rightarrow \exists_x^d (A \rightarrow B) \quad \text{for } A \text{ n.c., } B \text{ c.r.} \)

\( (A \rightarrow_{\text{nc}}^d \exists_x^d B) \rightarrow \exists_x^d (A \rightarrow B) \quad \text{for } B \text{ c.r.} \)
Formulas and predicates

A theory of computable functionals

Brouwer - Heyting - Kolmogorov and decorations

The type of a formula or predicate

Realizability

Extracted terms
For derivations $M^A$ with $A$ n.c. let $\text{et}(M^A) := \varepsilon$. Otherwise

\[
\text{et}(u^A) := v_u^{\tau(A)} \quad (v_u^{\tau(A)} \text{ uniquely associated to } u^A),
\]

\[
\text{et}(\left(\lambda_{u^A} M^B\right)^{A \rightarrow B}) := \begin{cases} 
\lambda_{v_u^{\tau(A)}} \text{et}(M) & \text{if } A \text{ is c.r.} \\
\text{et}(M) & \text{if } A \text{ is n.c.}
\end{cases}
\]

\[
\text{et}(\left(M^A \rightarrow B N^A\right)^B) := \begin{cases} 
\text{et}(M) \text{et}(N) & \text{if } A \text{ is c.r.} \\
\text{et}(M) & \text{if } A \text{ is n.c.}
\end{cases}
\]

\[
\text{et}(\left(\lambda_{x^\rho} M^A\right)^{\forall x^A}) := \lambda_{x}^\rho \text{et}(M),
\]

\[
\text{et}(\left(M^{\forall x A(x)} r\right)^A(r)) := \text{et}(M)r,
\]

\[
\text{et}(\left(\lambda_{u^A} M^B\right)^{A \rightarrow^{\text{nc}} B}) := \text{et}(M),
\]

\[
\text{et}(\left(M^{A \rightarrow^{\text{nc}} B} N^A\right)^B) := \text{et}(M),
\]

\[
\text{et}(\left(\lambda_{x^\rho} M^A\right)^{\forall^{\text{nc}} x^A}) := \text{et}(M),
\]

\[
\text{et}(\left(M^{\forall^{\text{nc}} x A(x)} r\right)^A(r)) := \text{et}(M).
\]
Extracted terms for the axioms.

- Let $I$ be c.r.

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

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

- For the invariance axioms we take identities.

**Theorem (Soundness)**

Let $M$ be a derivation of a c.r. formula $A$ from assumptions $u_i : C_i$ $(i < n)$. Then we can derive $\text{et}(M) \circ A$ from assumptions $v_{u_i} \circ C_i$ in case $C_i$ is c.r. and $C_i$ otherwise.

**Proof.**

By induction on $M$. □
Conclusion

- Assume $M$ proves $A$. The derivation in TCF of $\text{et}(M) \rightarrow A$ is automatically generated and can be machine checked.
- Minlog can translate $\text{et}(M)$ into Scheme and Haskell code.
- Coq’s extraction returns Ocaml, Scheme or Haskell code, not terms in a “logical” language like $T^+$.
- Agda views (complete) proofs as programs.