Logic of inductive definitions with formal neighbourhoods

Helmut Schwichtenberg

Mathematisches Institut, LMU, München

Proof and Computational Complexity, Oslo, 8-9. August 2008

Why extract computational content from proofs?

- Proofs are machine checkable $\Rightarrow$ no logical errors.
- Program on the proof level $\Rightarrow$ maintenance becomes easier.
- Discover unexpected content, in proofs of $\tilde{\exists}_x A := \neg\forall_x \neg A$, via proof interpretations: (refined) A-translation or Gödel's Dialectica interpretation (Ratiu, Trifonov).

Here:

- Content of proofs in analysis.
- Allow abstract treatment (Cruz-Filipe 2004, O'Connor 2008, Zumkeller 2008). Concrete data types for realizers only:
  real $\sim$ stream of signed digits,
  continuous function $\sim$ stream transformer.

(Cf. U. Berger, From coinductive proofs to exact real arithmetic. Draft, 2008).

Computable functionals of finite types

- Gödel 1958: "Über eine bisher noch nicht benützte Erweiterung des finiten Standpunkts", namely computable finite type functions.
- Need partial continuous functionals as their intendend domain (Scott 1969). The total ones then appear as a dense subset (Kreisel 1959, Ershov 1972).
- Type theory of Martin-Löf 1983 deals with total (structural recursive) functionals only. Fresh start, based on (a simplified form of) information systems (Scott 1982).

Atomic coherent information systems (acis's)

- Acis: $(A, \smile, \geq)$ such that $\smile$ (consistent) is reflexive and symmetric, $\geq$ (entails) is reflexive and transitive and $a \smile b \to b \geq c \to a \smile c$.
- Formal neighborhood: $U \subseteq A$ finite and consistent. We write $U \geq a$ for $\exists_{b \in U} b \geq a$, and $U \geq V$ for $\forall_{a \in V} U \geq a$.
- Function space: Let $\mathbf{A} = (A, \smile_A, \geq_A)$ and $\mathbf{B} = (B, \smile_B, \geq_B)$ be acis's. Define $\mathbf{A} \to \mathbf{B} = (C, \smile, \geq)$ by

$$
\begin{aligned}
C &:= \mathrm{Con}_A \times B, \\
(U, b) \smile (V, c) &:= U \smile_A V \to b \smile_B c, \\
(U, b) \geq (V, c) &:= V \geq_A U \wedge b \geq_B c.
\end{aligned}
$$

$\mathbf{A} \to \mathbf{B}$ is an acis again.

Ideals, Scott topology

- Ideal: $x \subseteq A$ consistent and deductively closed. $|\mathbf{A}|$ is the set of ideals (points, objects) of $\mathbf{A}$.
- $|\mathbf{A}|$ carries a natural topology, with cones $\tilde{U} := \{ z \mid z \supseteq U \}$ generated by the formal neighborhoods U as basis.

Theorem (Scott 1982)

The continuous maps $f\colon |\mathbf{A}| \to |\mathbf{B}|$ and the ideals $r \in |\mathbf{A} \to \mathbf{B}|$ are in a bijective correspondence.

Free algebras

are given by their constructors. Examples

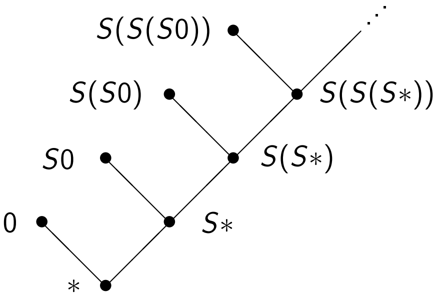
- Natural numbers **N**: 0, S.
- Binary trees **T**: nil, C.
- Unit **U**: **u**.
- Booleans **B**: $\mathtt{tt}$, $\mathtt{ff}$.
- Signed digits **SD**: -1, 0, $+1$.
- Lists of signed digits **L**(**SD**): nil, $d :: l$.

We always require a nullary constructor.

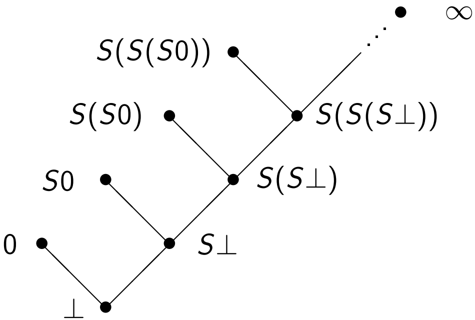
Turning free algebras into information systems

- Commonly done by adding $\bot$: "flat cpo". Problems arise:
- Problem 1: Constructors are not injective: $\mathrm{C}(\bot, b) = \bot = \mathrm{C}(a, \bot)$.
- Problem 2: Constructors do not have disjoint ranges: $\mathrm{C}_1(\bot) = \bot = \mathrm{C}_2(\bot)$.
- Solution: Use as atoms constructor expressions involving a symbol $*$, meaning "no information".

Example: atoms and entailment for **N**



Example: ideals for **N**



Total and cototal ideals

For a base type ι, the total ideals are defined inductively:

- 0 is total (0 being the nullary constructor), and
- If $\vec{z}$ are total, then so is $\mathrm{C}\vec{z}$.

The cototal ideals x are those of the form $\mathrm{C}\vec{z}$ with C a constructor of ι and $\vec{z}$ cototal. – For example, in **L**(**SD**),

- the total ideals are the finite and
- the cototal ideals are the finite or infinite

lists of signed digits ($\sim$ an interval with rational end points or a stream real, both in $[-1, 1]$).

Totality in higher types, density

- An ideal r of type $\rho \to \sigma$ is total iff for all total z of type ρ, the result $|r|(z)$ of applying r to z is total.
- Density theorem (Kreisel 1959, Ershov 1972, U. Berger 1993): Assume that all base types are finitary. Then for every $U \in \mathrm{Con}_\rho$ we can find a total x such that $U \subseteq x$.

A common extension T^+ of Gödel's T and Plotkin's PCF

- Terms $M, N ::= x^\rho \mid C \mid D \mid (\lambda_{x^\rho} M^\sigma)^{\rho\to\sigma} \mid (M^{\rho\to\sigma} N^\rho)^\sigma$.
- Constants D defined by computation rules. Examples:
  Recursion $\mathcal{R}_{\mathbf{N}}^\tau \colon \mathbf{N} \to (\mathbf{U} \times \tau \times \mathbf{N} \to \tau) \to \tau$.

  $$\mathcal{R}0xy = x, \quad \mathcal{R}(\mathrm{S}n)xy = yn(\mathcal{R}nxy).$$

  Corecursion $\mathcal{C}_{\mathbf{N}}^\tau \colon \tau \to (\tau \to \mathbf{U} + \tau + \mathbf{N}) \to \mathbf{N}$.

  $$\mathcal{C}xy = [\textbf{case } yx \textbf{ of } 0 \mid \lambda_z(\mathrm{S}[\textbf{case } z^{\tau+\mathbf{N}} \textbf{ of } \lambda_u(\mathcal{C}uy) \mid \lambda_n n])].$$

  Case of type $\rho + \sigma \to (\rho \to \tau) \to (\sigma \to \tau) \to \tau$:

  $$[\textbf{case } (\mathrm{inl}(M))^{\rho+\sigma} \textbf{ of } \lambda_x N(x) \mid \lambda_y K(y)] = N(M),$$
  $$[\textbf{case } (\mathrm{inr}(M))^{\rho+\sigma} \textbf{ of } \lambda_x N(x) \mid \lambda_y K(y)] = K(M).$$

Destructors

Every algebra ι with k constructors each of arity n_i $(i < k)$ has a destructor D_ι of type

$$\iota \to \sum_{i<k} \prod_{j<n_i} \iota.$$

Computation rules:

$$D_\iota(\mathrm{C}_i(\vec{x})) = \mathrm{in}_i(\vec{x}).$$

Example: $D_{\mathbf{N}} \colon \mathbf{N} \to \mathbf{U} + \mathbf{N}$ is defined by the computation rules

$$D_{\mathbf{N}}(\mathrm{S}n) = \mathrm{inr}(n),$$
$$D_{\mathbf{N}}(0) = \mathrm{inl}(\mathbf{u}).$$

Operational and denotational semantics

- Denotational: inductive definition of $(\vec{U}, b) \in [\![\lambda_{\vec{x}} M]\!]$.
- Operational: define $M \in [a]$, by induction on the type of a.
- Plotkin (1977) proved: Whenever an atom b belongs to the value of a closed term M, then M head-reduces to an atom entailing b. Here we have more generally:

Theorem (Adequacy)

$$(\vec{U}, b) \in [\![\lambda_{\vec{x}} M]\!] \to \lambda_{\vec{x}} M \in [(\vec{U}, b)].$$

Logic of inductive definitions LID

- is based on T^+. Terms with the same reduct are identified.
- It contains inductively and coinductively defined predicates, given by their clauses and (least and greatest) fixed point axioms. Examples: $T, T^\infty, \mathrm{Eq}, \exists$.
- Uses minimal logic only: introduction and elimination rules for $\to$ and $\forall$.
- Ex falso quodlibet is provable, when one defines falsity by $\mathbf{F} := \mathrm{Eq}_{\mathbf{B}}(\mathrm{ff}, \mathrm{tt})$.

Totality

Totality $T_{\mathbf{N}}$ is inductively defined by the clauses

$$\exists_{m \in T_{\mathbf{N}}}(m=0),$$
$$\forall_{n \in T_{\mathbf{N}}} \exists_{m \in T_{\mathbf{N}}}(m=\mathrm{S}n).$$

and the least fixed point axiom (or induction)

$$\forall_{n \in T_{\mathbf{N}}}\big(A(0) \to \forall_{n \in T_{\mathbf{N}}}(A(n) \to A(\mathrm{S}n)) \to A(n^{\mathbf{N}})\big).$$

Cototality

Cototality $T_{\mathbf{N}}^{\infty}$ is coinductively defined by the clause

$$\forall^{\mathrm{U}}_{n \in T_{\mathbf{N}}^{\infty}} (n{=}0 \vee \exists^{\mathrm{U}}_{m \in T_{\mathbf{N}}^{\infty}} (n{=}\mathrm{S}m))$$

and the greatest fixed point axiom (or coinduction)

$$\forall^{\mathrm{U}}_{n} (A(n) \to \\ \quad \forall^{\mathrm{U}}_{n} (A(n) \to n{=}0 \vee \exists^{\mathrm{U}}_{m} [n{=}\mathrm{S}m \wedge (A(m) \vee T_{\mathbf{N}}^{\infty}(m))]) \to \\ \quad T_{\mathbf{N}}^{\infty}(n)).$$

Soundness

For every proof M in LID we can define its extracted term $[\![M]\!]$ (modified realizability interpretation: Kreisel 1959, Seisenberger 2003). In particular this needs to be done for the axioms.

Theorem

Let M be a derivation of A from assumptions $u_i \colon C_i$ $(i < n)$. Then we can find a derivation of $[\![M]\!]$ $\mathbf{r}$ A from assumptions $\bar{u}_i \colon x_{u_i}$ $\mathbf{r}$ C_i.

Proof.

Induction on M. □

Realizing the fixed point axiom of T^∞

- Recall the (greatest) fixed point axiom $(T^\infty_{\mathbf{N}})^{\mathrm{fp}}$ for cototality

$$\forall^{\mathsf{U}}_n(A(n) \to \forall^{\mathsf{U}}_n(A(n) \to n{=}0 \lor \exists^{\mathsf{U}}_m[n{=}\mathrm{S}m \land (A(m) \lor T^\infty_{\mathbf{N}}(m))]) \to T^\infty_{\mathbf{N}}(n)).$$

- Its type is

$$\tau \to (\tau \to \mathbf{U} + \tau + \mathbf{N}) \to \mathbf{N},$$

since $\tau(T^\infty_{\mathbf{N}}(n)) := \mathbf{N}$ and $\tau(\forall^{\mathsf{U}}_x B) := \tau(\exists^{\mathsf{U}}_x B) := \tau(B)$.

- Its extracted term is the corecursion operator $\mathcal{C}^\tau_{\mathbf{N}}$.

Realizing the clause of T^∞

- Recall the clause for cototality

$$\forall^{\mathbf{U}}_{n \in T^\infty_{\mathbf{N}}} (n{=}0 \vee \exists^{\mathbf{U}}_{m \in T^\infty_{\mathbf{N}}} (n{=}\mathrm{S}m)).$$

- Its type is

$$\mathbf{N} \to \mathbf{U} + \mathbf{N}$$

since $\tau(T^\infty_{\mathbf{N}}(n)) := \mathbf{N}$ and $\tau(\forall^{\mathbf{U}}_x B) := \tau(\exists^{\mathbf{U}}_x B) := \tau(B)$.

- Its extracted term is the destructor $D_{\mathbf{N}}$.

Continuous functions $f\colon \mathbb{I} \to \mathbb{I}$ where $\mathbb{I} := [-1, 1]$

Wf coinductively defined: f continuous function in write mode.
Rf inductively defined: f continuous function in read mode.
(Simultaneous) clauses:

$$\forall_f (Wf \to \mathrm{Id} f \lor Rf),$$
$$\forall_f (f[\mathbb{I}] \subseteq \mathbb{I}_d \to W(\mathrm{out}_d \circ f) \to Rf) \quad (d \in \mathbf{SD}),$$
$$\forall_f (\forall_d R(f \circ \mathrm{in}_d) \to Rf).$$

The corresponding (greatest and least) fixed point axioms are

$$\forall_f (A(f) \to \forall_f (A(f) \to \mathrm{Id} f \lor Rf) \to Wf),$$
$$\forall_f (Rf \to (\forall_f (f[\mathbb{I}] \subseteq \mathbb{I}_d \to W(\mathrm{out}_d \circ f) \to A(f)))_{d \in \mathbf{SD}} \to$$
$$\forall_f (\forall_d A(f \circ \mathrm{in}_d) \to \forall_d R(f \circ \mathrm{in}_d) \to A(f)) \to$$
$$A(f)).$$

Conclusion

- Partial continuous functionals: Acis's, ideals, free algebras, totality and cototality.
- T^+, a common extension of Gödel's T and Plotkin's PCF: Constants defined by computation rules, denotational and operational semantics, adequacy theorem.
- Logic of inductive definitions LID: based on T^+.
- Computational content: Soundness theorem. May treat continuous functions abstractly. Concrete data types for realizers only: real $\sim$ stream of signed digits, continuous function $\sim$ stream transformer.