Proofs and computations

Helmut Schwichtenberg
(j.w.w. Kenji Miyamoto)

Mathematisches Institut, LMU, München

Humboldt-Kolleg “Proof”, Universität Bern,
9.-13. September 2013

What more do we know if we have proved a theorem by restricted means rather than knowing that it is true? (G. Kreisel)

- Proofs can have (hidden) computational content.
- Extract such computational content from proofs.

Tools

- Computationally relevant and irrelevant logical connectives.
- Inductively/coinductively defined predicates.

Classical and constructive proofs

- View classical logic as a fragment of constructive logic.
- Use both $\exists_x A$ and its "weak" variant $\tilde{\exists}_x A := \neg \forall_x \neg A$.

Proof transformations:

- From $\vdash \forall_x \tilde{\exists}_y A$ obtain $\vdash \forall_x \exists_y A$.

Dickson's lemma

$$\forall_{f.g} \tilde{\exists}_{i,j} (i < j \wedge f(i) \leq f(j) \wedge g(i) \leq g(j)).$$

Proof by a simplification of Nash–Williams' (1963) "minimal bad sequence" argument for Higman's lemma. It is not constructive since it requires determining the minimum of an infinite set.

Guarded general recursion with measure μ:

$$\mathcal{F}_\mu^+ xG\mathrm{tt} = Gx(\lambda_y \mathcal{F}_\mu^+ yG(\mu y < \mu x)),$$
$$\mathcal{F}_\mu^+ xG\mathrm{ff} = \varepsilon.$$

Let $\mathcal{F}_\mu xG := \mathcal{F}_\mu^+ xG\mathrm{tt}$ (general recursion). Term extracted from the (transformed) proof of Dickson's lemma:

$$\lambda_{f,g}\mathcal{F}_f 0(\lambda_n \mathcal{F}_g n(\lambda_{i,\xi,h}\mathcal{F}_f(i+1)(\lambda_{j,h}\begin{cases} hj & \text{if } fj < fi \\ \xi jh & \text{if } gj < gi\,))) \\ \langle i,j\rangle & \text{else}\end{cases}$$

- The 3 $\mathcal{F}$'s correspond to 3 uses of the minimum principle.
- Higher-order term: $\mathcal{F}$ has a type-3 argument (ξ is type-2).
- It is unlikely that a human would write such a program.
- Experiments: better than brute-force search.

Ishihara's trick. Let f be a linear map from a Banach space X into a normed linear space Y, and (u_m) a sequence in X converging to 0. Then for $0 < a < b$

$$\exists_m (a \leq \|fu_m\|) \quad \text{or} \quad \forall_m (\|fu_m\| \leq b).$$

Proof. Let M be a modulus of convergence of (u_m) to 0. Call m a hit on n if $M_n \leq m < M_{n+1}$ and $a \leq \|fu_m\|$. Define $h \colon \mathbb{N} \to \mathbb{N}$:

- $h_n = 0$ if for all $n' \leq n$ there is no hit;
- $h_n = m + 2$ if at n for the first time we have a hit, with m;
- $h_n = 1$ if there is an $n' < n$ with a hit.

To define h use $g \colon \mathbb{N} \to \mathbb{B}$:

$$\begin{cases} a \leq \|fu_m\| & \text{if } gm \\ \|fu_m\| \leq b & \text{otherwise.} \end{cases}$$

From h define a Cauchy sequence (v_n) in X:

- $v_n = 0$ if $h_n = 0$;
- $v_n = (n+1)u_m$ if $h_n = m+2$;
- $v_n = v_{n-1}$ if $h_n = 1$.

By completeness of X: limit v of (v_n). Pick n_0 s.t. $\|fv\| \leq n_0 a$. Assume first hit at $n > n_0$, with value m. Then $v{=}v_n{=}(n{+}1)u_m$,

$$(n{+}1)a \leq (n{+}1)\|fu_m\| = \|(n{+}1)(fu_m)\| = \|f((n{+}1)u_m)\| = \|fv\| \leq na,$$

a contradiction. Hence beyond n_0 there is no first hit.

- Case $\forall_{n<n_0}(h_n = 0)$. Then: no hit, hence $\|fu_n\| \leq b$ for all n.
- Else: hit before n_0, hence $a \leq \|fu_n\|$ for some n.

```
[f,us,M,a,a0,k]
 [let g
   ([n]negb
       (cAC([n0]cApproxSplitRat a a0 lnorm(f(us n0))k)n))
   [case (H g M
          (cRealPosRatBound
           lnorm(f((cXCompl xi)((V xi)g M us)
                   ([k0]abs(IntS(2*k0)max 0))))
           a))
    (Zero -> False)
    (Succ n0 -> True)]]
```

XCompl:

$$\forall_{us,M}(\forall_{k,m>n\geq M_k}\|u_n - u_m\| \leq 1/2^k \to \exists_v \forall_{k,n\geq M_k}\|v - u_n\| \leq 1/2^k)$$

RealPosRatBound: $\forall_{x,a>0}\exists_n x \leq na$
ApproxSplitRat: $\forall_{a,b,x,k}(1/2^k \leq b - a \to x \leq b \vee a \leq x)$
AC: $\forall_m \exists_p R(m,p) \to \exists_g \forall_m R(m, g(m))$.

Computing with infinite data

Real number

- Type-1: Cauchy sequence of rationals (with modulus).
- Type-0: "Stream" of signed digits $\{-1, 0, 1\}$.

Real function

- Type-2: type-1 reals $\mapsto$ reals.
- Type-1: type-0 reals $\mapsto$ reals.

Example: average of two reals

Ulrich Berger and Monika Seisenberger (2009, 2010).

- Extraction from a proof dealing with abstract reals.
- Proof involving coinduction of the proposition that any two reals in $[-1, 1]$ have their average in the same interval.
- B & S informally extract a Haskell program from this proof, which works with stream representations of reals.

Here: formalization of the proof, and machine extraction of its computational content.

Average of two reals in $[-1,1]$:

$$\forall_{x.y}(x, y \in [-1,1] \to \frac{x+y}{2} \in [-1,1]).$$

Has type-2 content, since "x is Cauchy sequence" is of type 1.
Want: type-0 representation.

$$\texttt{Average}\colon \forall^{\mathrm{nc}}_{x.y}(Rx \to Ry \to R\frac{x+y}{2}).$$

How to define predicates with type-0 content? Inductive predicates.

Semantics

- Base types: "ideals" (possibly infinite) in free algebras.
- Function types: Scott-Ershov domains of partial continuous functionals.

Free algebra **J** of intervals: constructors

$$\mathbb{I}\colon \mathbf{J} \quad \text{(for } [-1,1]\text{)},$$
$$\mathrm{C}\colon \mathbf{SD} \to \mathbf{J} \to \mathbf{J} \quad \text{(for left, middle, right half)}.$$

Ideals in **J**:

- infinite path (stream, cototal ideal),
- finite interval (total ideal).

Define inductively a unary predicate I by the clauses

$$I0, \qquad \forall_x^{\mathrm{nc}} \forall_d (Ix \to I\frac{x+d}{2})$$

and the least-fixed-point axiom (induction).

I's generation trees: ideals in $\mathbf{J}$. Reason: clauses $\sim$ constructors

$$\mathbb{I} \colon \mathbf{J}, \qquad \mathrm{C} \colon \mathbf{SD} \to \mathbf{J} \to \mathbf{J}.$$

Dual:

$$\forall_x^{\mathrm{nc}} ({}^{\mathrm{co}}Ix \to x = 0 \lor \exists_y^{\mathrm{r}} \exists_d ({}^{\mathrm{co}}Iy \land x = \frac{y+d}{2}))$$

and the greatest-fixed-point axiom (coinduction).

Content of least- and greatest-fixed-point axioms: $\mathcal{R}$ and ${}^{\mathrm{co}}\mathcal{R}$.

- $\mathcal{R}^{\tau}_{\mathbf{J}}\colon \mathbf{J} \to \tau \to (\mathbf{SD} \to \mathbf{J} \to \tau \to \tau) \to \tau$.
- The conversion rules for $\mathcal{R}$ with total ideals as recursion arguments work from the leaves towards the root, and terminate because total ideals are well-founded.
- For cototal ideals (streams) a similar operator is available to define functions with cototal ideals as values: corecursion.
- ${}^{\mathrm{co}}\mathcal{R}^{\tau}_{\mathbf{J}}\colon \tau \to (\tau \to \mathbf{U} + \mathbf{SD} \times (\mathbf{J} + \tau)) \to \mathbf{J} \quad$ ($\mathbf{U}$ unit type).
- Conversion rule

$$
\begin{aligned}
{}^{\mathrm{co}}\mathcal{R}^{\tau}_{\mathbf{J}} N M \mapsto [&\mathbf{case}\ (MN)^{\mathbf{U}+\mathbf{SD}\times(\mathbf{J}+\tau)}\ \mathbf{of} \\
&\mathrm{inl}\ \_ \mapsto \square \mid \\
&\mathrm{inr}\langle d, z\rangle \mapsto \mathrm{C}_d[\mathbf{case}\ z^{\mathbf{J}+\tau}\ \mathbf{of} \\
&\qquad\qquad\qquad \mathrm{inl}\ \_ \mapsto \square \mid \\
&\qquad\qquad\qquad \mathrm{inr}\ u^{\tau} \mapsto {}^{\mathrm{co}}\mathcal{R}^{\tau}_{\mathbf{J}} u M]].
\end{aligned}
$$

$\texttt{CauchySds}\colon \forall^{\mathrm{nc}}_{x\in[-1,1]}(\forall_n\exists_a(a-1/2^n \le x \le a+1/2^n) \to {}^{\mathrm{co}}Ix).$

Proof via coinduction. Content:

```
[as]
 (CoRec (nat=>rat)=>iv)as
 ([as0]
   Inr[let d
        [case (as0(Succ(Succ Zero)))
         (k#p ->
         [case k
           (p0 -> [if (SZero(SZero p0)<p) Mid Rht])
           (0 -> Mid)
           (IntN p0 ->
             [if (SZero(SZero p0)<=p) Mid Lft])])]
        (d@(InR nat=>rat iv)
           ([n]2*as0(Succ n)-SDToInt d))])
```

Haskell translation

$$\texttt{Average}\colon \forall^{\mathrm{nc}}_{x,y}({}^{\mathrm{co}}Ix \to {}^{\mathrm{co}}Iy \to {}^{\mathrm{co}}I\,\frac{x+y}{2}).$$

Proof via coinduction.

Define $(1/2)\sqrt{2}$ and $3/4$ as terms, by their Cauchy sequences.

```
(terms-to-haskell-program
 "~/temp/average.hs"
 (list (list neterm-average "neterm_average")
       (list neterm-cauchysds "neterm_cauchysds")
       (list halfsqrttwo "halfsqrttwo")
       (list threebyfour "threebyfour")))
```

Experiment

```
*Main> takeIv 9 (neterm_average
                    (neterm_cauchysds halfsqrttwo)
                    (neterm_cauchysds threebyfour))

["Rht","Rht","Mid","Mid","Lft","Rht","Lft","Mid","Rht"]
```

$$
\begin{aligned}
0.728515625 &= \frac{1}{2} + \frac{1}{4} - \frac{1}{32} + \frac{1}{64} - \frac{1}{128} + \frac{1}{512} \\
0.728553\ldots &= \frac{\sqrt{2}/2 + 3/4}{2}
\end{aligned}
$$

A theory of computable functionals, TCF

- A variant of HA^ω.
- Intended model:
  - Base types: "ideals" (possibly infinite) in (non-flat information systems for) free algebras.
  - Function types: Scott-Ershov domains of partial continuous functionals.
- Constants for (partial) computable functionals, defined by equations.
- Inductively and coinductively defined predicates. (Co)totality for base types (co)inductively defined.

Relation to type theory

- Main difference: partial functionals are first class citizens.
- "Logic enriched": Formulas and types kept separate.
- Minimal logic: $\to, \forall$ only. $\mathrm{Eq}(x, y)$ (Leibniz equality), $\exists$, $\vee$, $\wedge$ inductively defined (Martin-Löf).
- $\bot := \mathrm{Eq}(\mathrm{False}, \mathrm{True})$. Ex-falso-quodlibet: $\bot \to A$ provable.

Decorations

$\to, \forall$ and $\to^{\mathrm{nc}}, \forall^{\mathrm{nc}}$ for removal of abstract data, and fine-tuning. Introduction rules for $\to^{\mathrm{nc}}, \forall^{\mathrm{nc}}$ restricted to "non-computational" (assumption or object) variables.

Example: decorating disjunction.

- $A \vee B$ is inductively defined by the clauses

$$A \to A \vee B, \qquad B \to A \vee B$$

  with least-fixed-point axiom

$$A \vee B \to (A \to P) \to (B \to P) \to P.$$

- Decoration leads to variants $\vee^{\mathrm{d}}, \vee^{\mathrm{l}}, \vee^{\mathrm{r}}, \vee^{\mathrm{u}}$ (d for "double", l for "left", r for "right" and u for "uniform").

Decorating disjunction

Clauses:

$$\begin{aligned} &A \to^{\mathrm{c}} A \vee^{\mathrm{d}} B, && B \to^{\mathrm{c}} A \vee^{\mathrm{d}} B, \\ &A \to^{\mathrm{c}} A \vee^{\mathrm{l}} B, && B \to^{\mathrm{nc}} A \vee^{\mathrm{l}} B, \\ &A \to^{\mathrm{nc}} A \vee^{\mathrm{r}} B, && B \to^{\mathrm{c}} A \vee^{\mathrm{r}} B, \\ &A \to^{\mathrm{nc}} A \vee^{\mathrm{u}} B, && B \to^{\mathrm{nc}} A \vee^{\mathrm{u}} B. \end{aligned}$$

Least-fixed-point axioms:

$$\begin{aligned} &A \vee^{\mathrm{d}} B \to^{\mathrm{c}} (A \to^{\mathrm{c}} P) \to^{\mathrm{c}} (B \to^{\mathrm{c}} P) \to^{\mathrm{c}} P, \\ &A \vee^{\mathrm{l}} B \to^{\mathrm{c}} (A \to^{\mathrm{c}} P) \to^{\mathrm{c}} (B \to^{\mathrm{nc}} P) \to^{\mathrm{c}} P, \\ &A \vee^{\mathrm{r}} B \to^{\mathrm{c}} (A \to^{\mathrm{nc}} P) \to^{\mathrm{c}} (B \to^{\mathrm{c}} P) \to^{\mathrm{c}} P, \\ &A \vee^{\mathrm{u}} B \to^{\mathrm{c}} (A \to^{\mathrm{nc}} P) \to^{\mathrm{c}} (B \to^{\mathrm{nc}} P) \to^{\mathrm{c}} P. \end{aligned}$$

Decorating the existential quantifier

- $\exists_x A$ is inductively defined by the clause

$$\forall_x (A \to \exists_x A)$$

with least-fixed-point axiom

$$\exists_x A \to \forall_x (A \to P) \to P.$$

- Decoration leads to variants $\exists^{\mathrm{d}}, \exists^{\mathrm{l}}, \exists^{\mathrm{r}}, \exists^{\mathrm{u}}$.

$$\begin{array}{ll}
\forall_x (A \to \exists^{\mathrm{d}}_x A), & \exists^{\mathrm{d}}_x A \to \forall_x (A \to P) \to P, \\
\forall_x (A \to^{\mathrm{nc}} \exists^{\mathrm{l}}_x A), & \exists^{\mathrm{l}}_x A \to \forall_x (A \to^{\mathrm{nc}} P) \to P, \\
\forall^{\mathrm{nc}}_x (A \to \exists^{\mathrm{r}}_x A), & \exists^{\mathrm{r}}_x A \to \forall^{\mathrm{nc}}_x (A \to P) \to P, \\
\forall^{\mathrm{nc}}_x (A \to^{\mathrm{nc}} \exists^{\mathrm{u}}_x A), & \exists^{\mathrm{u}}_x A \to^{\mathrm{nc}} \forall^{\mathrm{nc}}_x (A \to^{\mathrm{nc}} P) \to P.
\end{array}$$

Realizability interpretation

- Define a formula $t \mathbf{r} A$, for A a formula and t a term

$$
\begin{aligned}
t \mathbf{r} (A \to B) &:= \forall_x (x \mathbf{r} A \to tx \mathbf{r} B), \\
t \mathbf{r} (A \to^{\mathrm{nc}} B) &:= \forall_x (x \mathbf{r} A \to t \mathbf{r} B), \\
t \mathbf{r} (\forall_x A) &:= \forall_x (tx \mathbf{r} A), \\
t \mathbf{r} (\forall_x^{\mathrm{nc}} A) &:= \forall_x (t \mathbf{r} A).
\end{aligned}
$$

- From a proof M we can extract its computational content, a term $\mathrm{et}(M)$.
- Soundness theorem:
  If M proves A, then $\mathrm{et}(M) \mathbf{r} A$ can be proved.

Further work

Type-0 representation of (uniformly) continuous real functions: "Read-Write tree". Output signed digit after reading finitely many (possibly 0) input signed digits, and carry on.

- Based on work of Ulrich Berger.
- Requires "nested" algebras and simultaneous inductively/coinductively defined predicates.
- Details in forthcoming thesis of Kenji Miyamoto.

References

- U. Berger, From coinductive proofs to exact real arithmetic. CSL 2009.
- H. Ishihara, A constructive closed graph theorem. 1990
- K. Miyamoto and H.S., Program extraction in exact real arithmetic. MSCS 2012.
- K. Miyamoto, F. Nordvall Forsberg and H.S., Program extraction from nested definitions. ITP 2013
- H.S. and S.S. Wainer, Proofs and Computations. Perspectives in Logic, ASL & Cambridge UP, 2012.