Decorating proofs

Helmut Schwichtenberg

joint work with Luca Chiarabini and Diana Ratiu
Mathematisches Institut, LMU, München

Leeds Symposium on Proof Theory and Constructivism,
3 - 16 July 2009

Logic

- Typed language, with free algebras as base types.
- Intended domains: partial continuous functionals.
- Terms are those of T^+, a common extension of Gödel's T and Plotkin's PCF.
- Natural deduction rules for $\to$ and $\forall$ ("minimal logic").
- All predicates are defined inductively. Examples: (Leibniz) equality Eq, totality, $\exists$, $\wedge$, $\vee$.

Computational content

- Proofs have two aspects:
  (i) They guarantee correctness.
  (ii) They may have computational content.
- Computational content only enters a proof via inductively (or coinductively) defined predicates.
- To fine tune the computational content of a proof, distinguish $\to^{\mathrm{c}}$, $\forall^{\mathrm{c}}$ (computational) and $\to$, $\forall$ (non-computational).

Natural deduction: assumption variables u^A. Rules for $\to^{\mathrm{c}}$:

derivation	proof term
$\dfrac{\begin{array}{c}[u\colon A]\\ \mid M\\ B\end{array}}{A \to^{\mathrm{c}} B}\,(\to^{\mathrm{c}})^+\,u$	$(\lambda_{u^A} M^B)^{A\to^{\mathrm{c}}B}$
$\dfrac{\begin{array}{c}\mid M\\ A \to^{\mathrm{c}} B\end{array}\qquad\begin{array}{c}\mid N\\ A\end{array}}{B}\,(\to^{\mathrm{c}})^-$	$(M^{A\to^{\mathrm{c}}B} N^A)^B$

Natural deduction: rules for $\forall^{\mathrm{c}}$

derivation	proof term
$\dfrac{\begin{array}{c}\mid M \\ A\end{array}}{\forall^{\mathrm{c}}_x A}\,(\forall^{\mathrm{c}})^+\,x \quad \text{(var. cond.)}$	$(\lambda_x M^A)^{\forall^{\mathrm{c}}_x A}$ (var. cond.)
$\dfrac{\begin{array}{c}\mid M \\ \forall^{\mathrm{c}}_x A(x)\end{array} \qquad r}{A(r)}\,(\forall^{\mathrm{c}})^-$	$(M^{\forall^{\mathrm{c}}_x A(x)}\, r)^{A(r)}$

Restrictions to $\to^+$ and $\forall^+$ (non-computational)

$\mathrm{CV}(M) :=$ the set of "computational variables" of a derivation M, relative to a fixed assigment $u^A \mapsto x_u^{\tau(A)}$. Consider

$$\dfrac{\begin{array}{c}[u\colon A]\\ \vert\ M\\ B\end{array}}{A \to B}\to^+ u \qquad \text{or as proof term} \quad (\lambda_{u^A} M^B)^{A\to B}.$$

$(\lambda_{u^A} M^B)^{A\to B}$ is correct if M^B is and $x_u \notin \mathrm{CV}(M^B)$. Consider

$$\dfrac{\begin{array}{c}\vert\ M\\ A\end{array}}{\forall_x A}\forall^+ x \qquad \text{or as proof term} \quad (\lambda_x M^A)^{\forall_x A} \qquad \text{(with var. condition)}.$$

$(\lambda_x M^A)^{\forall_x A}$ is correct if M^A is and $x \notin \mathrm{CV}(M^A)$.

Formulas as computational problems

- Kolmogorov (1925) proposed to view a formula A as a computational problem, of type $\tau(A)$, the type of a potential solution or "realizer" of A.
- Example: $\forall^{\mathrm{c}}_n \exists_{m>n} \mathrm{Prime}(m)$ has type $\mathbf{N} \to \mathbf{N}$.
- $A \mapsto \tau(A)$, a type or the "nulltype" symbol o.
- In case $\tau(A) = o$ proofs of A have no computational content; such formulas A are called computationally irrelevant (c.i.) or Harrop formulas; the others computationally relevant (c.r.).

Realizability

Let t be either a term of type $\tau(A)$ if this is a type, or the "nullterm" symbol ε if $\tau(A) = o$. Extend term application to ε:

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

We define the formula $t\ \mathbf{r}\ A$, read t realizes A.

$$\begin{aligned}
&\varepsilon\ \mathbf{r}\ I\vec{r} := I\vec{r} \quad \text{for } I \text{ not requiring witnesses (e.g., } \mathrm{Eq}),\\
&t\ \mathbf{r}\ (A \to^{\mathrm{c}} B) := \forall_x(x\ \mathbf{r}\ A \to tx\ \mathbf{r}\ B),\\
&t\ \mathbf{r}\ (A \to B) := \forall_x(x\ \mathbf{r}\ A \to t\ \mathbf{r}\ B),\\
&t\ \mathbf{r}\ \forall^{\mathrm{c}}_x A := \forall_x(tx\ \mathbf{r}\ A), \quad t\ \mathbf{r}\ \forall_x A := \forall_x(t\ \mathbf{r}\ A)
\end{aligned}$$

and similarly for $\exists$, $\wedge$, $\vee$ and other inductively defined I's.

Derivations and extracted terms

For M^A with A c.i. let $[\![M]\!] := \varepsilon$. Assume A is c.r. Then

$$
\begin{aligned}
[\![u^A]\!] &:= x_u^{\tau(A)} \quad (x_u^{\tau(A)} \text{ uniquely associated with } u^A),\\
[\![(\lambda_{u^A} M^B)^{A\to^c B}]\!] &:= \lambda_{x_u^{\tau(A)}} [\![M]\!],\\
[\![(M^{A\to^c B} N^A)^B]\!] &:= [\![M]\!][\![N]\!],\\
[\![(\lambda_{x^\rho} M^A)^{\forall^c_x A}]\!] &:= \lambda_{x^\rho} [\![M]\!],\\
[\![(M^{\forall^c_x A(x)} r)^{A(r)}]\!] &:= [\![M]\!] r,\\
[\![(\lambda_{u^A} M^B)^{A\to B}]\!] &:= [\![(M^{A\to B} N^A)^B]\!] := [\![(\lambda_{x^\rho} M^A)^{\forall_x A}]\!]\\
&:= [\![(M^{\forall_x A(x)} r)^{A(r)}]\!] := [\![M]\!].
\end{aligned}
$$

Define $\mathrm{CV}(M) := \mathrm{FV}([\![M]\!])$.

Soundness

Let x_{u^A} denote the nullterm symbol ε in case A is c.i.

Theorem (Soundness)

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

Proof.

Case $u : A$. Then $[\![u]\!] = x_u$.
Case $(\lambda_{u^A} M^B)^{A \to B}$. We must find a derivation of

$$[\![M]\!]\ \mathbf{r}\ (A \to B), \quad \text{which is} \quad \forall_x (x\ \mathbf{r}\ A \to [\![M]\!]\ \mathbf{r}\ B),$$

Use the IH.
Case $M^{A \to B} N^A$. We must find a derivation $[\![M]\!]\ \mathbf{r}\ B$. By IH we have $\forall_x (x\ \mathbf{r}\ A \to [\![M]\!]\ \mathbf{r}\ B)$ and $[\![N]\!]\ \mathbf{r}\ A$. Hence the claim. □

Decoration can simplify extracts

- Suppose that a proof M uses a lemma $L^{\mathrm{d}}\colon A \vee^{\mathrm{d}} B$.
- Then the extract $[\![M]\!]$ will contain the extract $[\![L^{\mathrm{d}}]\!]$.
- Suppose that the only computationally relevant use of L^{d} in M was which one of the two alternatives holds true, A or B.
- Express this by using a weakened lemma $L\colon A \vee B$.
- Since $[\![L]\!]$ is a boolean, the extract of the modified proof is "purified": the (possibly large) extract $[\![L^{\mathrm{d}}]\!]$ has disappeared.

Decoration algorithm

Goal: Insert as few as possible decorations into a proof.

- $\mathrm{Seq}(M)$ of a proof M consists of its context and end formula.
- The uniform proof pattern $U(M)$ of a proof M is the result of changing in c.r. formulas of M (i.e., not above a c.i. formula) all $\to^{\mathrm{c}}$, $\forall^{\mathrm{c}}$ into $\to$, $\forall$ (some restrictions on axioms, theorems).
- A formula D extends C if D is obtained from C by changing some $\to$, $\forall$ into $\to^{\mathrm{c}}$, $\forall^{\mathrm{c}}$.
- A proof N extends M if (i) N and M are the same up to variants of $\to$, $\forall$ in their formulas, and (ii) every c.r. formula of M is extended by the corresponding one in N.

Decoration algorithm

Assumption: For every axiom or theorem A and every decoration variant C of A we have another axiom or theorem whose formula D extends C, and D is the least among those extensions.

Theorem (Ratiu, S.)

Under the assumption above, for every uniform proof pattern U and every extension of its sequent $\mathrm{Seq}(U)$ we can find a decoration M_∞ of U such that

(a) $\mathrm{Seq}(M_\infty)$ *extends the given extension of* $\mathrm{Seq}(U)$, *and*

(b) M_∞ *is optimal in the sense that any other decoration M of U whose sequent $\mathrm{Seq}(M)$ extends the given extension of $\mathrm{Seq}(U)$ has the property that M also extends M_∞.*

Case $\to^-$. Consider a uniform proof pattern

$$\dfrac{\begin{array}{cc}\Phi,\Gamma & \Gamma,\Psi\\ \mid U & \mid V\\ A\to B & A\end{array}}{B}\ \to^-$$

Given: extension $\Pi,\Delta,\Sigma \Rightarrow D$ of $\Phi,\Gamma,\Psi \Rightarrow B$. Alternating steps:

- $\mathrm{IH}_a(U)$ for extension $\Pi,\Delta \Rightarrow A\to D \mapsto$ decoration M_1 of U whose sequent $\Pi_1,\Delta_1 \Rightarrow C_1 \overset{\smile}{\to} D_1$ extends $\Pi,\Delta \Rightarrow A\to D$. Suffices if A is c.i.: extension $\Delta_1,\Sigma \Rightarrow C_1$ of V is a proof (in c.i. parts of a proof $\to$, $\forall$ and $\to^{\mathrm{c}}$, $\forall^{\mathrm{c}}$ are identified). For A c.r:
- $\mathrm{IH}_a(V)$ for the extension $\Delta_1,\Sigma \Rightarrow C_1 \mapsto$ decoration N_2 of V whose sequent $\Delta_2,\Sigma_2 \Rightarrow C_2$ extends $\Delta_1,\Sigma \Rightarrow C_1$.
- $\mathrm{IH}_a(U)$ for $\Pi_1,\Delta_2 \Rightarrow C_2 \overset{\smile}{\to} D_1 \mapsto$ decoration M_3 of U whose sequent $\Pi_3,\Delta_3 \Rightarrow C_3 \overset{\smile}{\to} D_3$ extends $\Pi_1,\Delta_2 \Rightarrow C_2 \overset{\smile}{\to} D_1$.
- $\mathrm{IH}_a(V)$ for the extension $\Delta_3,\Sigma_2 \Rightarrow C_3 \mapsto$ decoration N_4 of V whose sequent $\Delta_4,\Sigma_4 \Rightarrow C_4$ extends $\Delta_3,\Sigma_2 \Rightarrow C_3$. . . .

Decorating axioms and theorems

- The "uninstantiated" formula of the axiom or theorem may contain the same predicate variable Q many times. The decoration algorithm needs to pick the "least upper bound" (w.r.t. extension) of the formula substituted for Q.
- The data base of theorems is checked whether there is one that fits as well, has its assumptions in the present context, and is minimal (w.r.t. extension) among all those. This alternative then is preferred.

Example: Maximal Scoring Segment (MSS)

- Let X be linearly ordered by $\leq$. Given $\mathrm{seg}\colon \mathbf{N} \to \mathbf{N} \to X$.
  Want: maximal segment

$$\forall_n^{\mathrm{c}} \exists_{i \leq k \leq n} \forall_{i' \leq k' \leq n} (\mathrm{seg}(i', k') \leq \mathrm{seg}(i, k)).$$

- Example: Regions with high G, C content in DNA.

$$X := \{G, C, A, T\},$$
$$g\colon \mathbf{N} \to X \quad \text{(gene)},$$
$$f\colon \mathbf{N} \to \mathbf{Z}, \quad f(i) := \begin{cases} 1 & \text{if } g(i) \in \{G, C\}, \\ -1 & \text{if } g(i) \in \{A, T\}, \end{cases}$$
$$\mathrm{seg}(i, k) = f(i) + \cdots + f(k).$$

- Special case: maximal end segment

$$\forall_n^{\mathrm{c}} \exists_{j \leq n} \forall_{j' \leq n} (\mathrm{seg}(j', n) \leq \mathrm{seg}(j, n)).$$

Example: MSS (ctd.)

Two proofs of the existence of a maximal end segment for $n+1$:
$\forall^{\mathrm{c}}_n \exists_{j \le n+1} \forall_{j' \le n+1} (\mathrm{seg}(j', n+1) \le \mathrm{seg}(j, n+1))$.

- Introduce an auxiliary parameter m; prove by induction on m

$$\forall^{\mathrm{c}}_n \forall^{\mathrm{c}}_{m \le n+1} \exists_{j \le n+1} \forall_{j' \le m} (\mathrm{seg}(j', n+1) \le \mathrm{seg}(j, n+1)).$$

- Use $\mathtt{ES}_n$: $\exists_{j \le n} \forall_{j' \le n} (\mathrm{seg}(j', n) \le \mathrm{seg}(j, n))$ and the additional assumption of monotonicity

$$\forall_{i,j,n} (\mathrm{seg}(i, n) \le \mathrm{seg}(j, n) \to \mathrm{seg}(i, n+1) \le \mathrm{seg}(j, n+1)).$$

Proceed by cases on $\mathrm{seg}(j, n+1) \le \mathrm{seg}(n+1, n+1)$.
If $\le$, take $n+1$, else the previous j.

Example: MSS (ctd.)

Prove the existence of a maximal segment by induction on n, simultaneously with the existence of a maximal end segment.

$$\forall^{\mathrm{c}}_n(\exists_{i\le k\le n}\forall_{i'\le k'\le n}(\mathrm{seg}(i',k')\le \mathrm{seg}(i,k)) \wedge \exists_{j\le n}\forall_{j'\le n}(\mathrm{seg}(j',n)\le \mathrm{seg}(j,n)))$$

In the step:

- Compare the maximal segment i,k for n with the maximal end segment $j,n+1$ proved separately.
- If $\le$, take the new i,k to be $j,n+1$. Else take the old i,k.

Depending on how the existence of a maximal end segment was proved, we obtain a quadratic or a linear algorithm.

Example: MSS (ctd.)

Could the better proof be found automatically? Have L1 and L2:

$$\forall_n^c \forall_{m\le n+1}^c \exists_{j\le n+1} \forall_{j'\le m}(\mathrm{seg}(j', n{+}1) \le \mathrm{seg}(j, n{+}1)),$$
$$\mathtt{Mon} \to \forall_n^c(\mathtt{ES}_n \to^c \forall_{m\le n+1} \exists_{j\le n+1} \forall_{j'\le m}(\mathrm{seg}(j', n{+}1) \le \mathrm{seg}(j, n{+}1))).$$

- The decoration algorithm arrives at L1 with

$$\forall_{m\le n+1} \exists_{j\le n+1} \forall_{j'\le m}(\mathrm{seg}(j', n{+}1) \le \mathrm{seg}(j, n{+}1)).$$

- L2 fits as well, its assumptions Mon and $\mathtt{ES}_n$ are in the context, and it is the less extended ($\forall_{m\le n+1}$ rather than $\forall_{m\le n+1}^c$), hence is preferred.

Induction vs. cases

- Recall the induction axiom

$$\forall^{\mathrm{c}}_n(Q0 \to^{\mathrm{c}} \forall^{\mathrm{c}}_n(Qn \to^{\mathrm{c}} Q(\mathrm{S}n)) \to^{\mathrm{c}} Qn).$$

- The cases axiom can only non-computationally use the step hypothesis (but it is available for c.i. parts of the proof)

$$\forall^{\mathrm{c}}_n(Q0 \to^{\mathrm{c}} \forall^{\mathrm{c}}_n(Qn \to Q(\mathrm{S}n)) \to^{\mathrm{c}} Qn).$$

- Extracts:

$$\mathcal{R}^{\tau}_{\mathbf{N}} \colon \mathbf{N} \to \tau \to (\mathbf{N} \to \tau \to \tau) \to \tau,$$
$$\mathcal{C}^{\tau}_{\mathbf{N}} \colon \mathbf{N} \to \tau \to (\mathbf{N} \to \tau) \to \tau.$$

Eager vs. lazy evaluation

- When normalizing terms, one may replace recursion operators by cases operators, where possible.
- Transfer this to the proof level: decoration replaces induction by cases axioms, where possible.
- Extraction at cases axioms returns "if-terms" (evaluated lazily). The soundness theorem continues to hold.
- Why transfer program optimization to the proof level? Correctness of proofs is machine checkable.

Code Carrying Proofs

Passing continuations

Double induction

$$\forall^{\mathrm{c}}_n(Qn \to^{\mathrm{c}} Q(\mathrm{S}n) \to^{\mathrm{c}} Q(\mathrm{S}(\mathrm{S}n))) \to^{\mathrm{c}} \forall^{\mathrm{c}}_n(Q0 \to^{\mathrm{c}} Q1 \to^{\mathrm{c}} Qn)$$

is proved in continuation passing style, i.e., not directly, but using as an intermediate assertion (proved by induction)

$$\forall^{\mathrm{c}}_{n,m}((Qn \to^{\mathrm{c}} Q(\mathrm{S}n) \to^{\mathrm{c}} Q(n+m)) \to^{\mathrm{c}} Q0 \to^{\mathrm{c}} Q1 \to^{\mathrm{c}} Q(n+m))$$

After decoration, the formula proved by induction becomes

$$\forall^{\mathrm{c}}_n\forall_m((Qn \to^{\mathrm{c}} Q(\mathrm{S}n) \to^{\mathrm{c}} Q(n+m)) \to^{\mathrm{c}} Q0 \to^{\mathrm{c}} Q1 \to^{\mathrm{c}} Q(n+m)).$$

Example: Fibonacci numbers

Goal: continuation based tail recursive definition of the Fibonacci function, from a proof of its totality.

- Let G be the graph of the Fibonacci function:

$$G(0,0), \quad G(1,1),$$
$$\forall_{n,v,w}(G(n,v) \to G(\mathrm{S}n,w) \to G(\mathrm{S}(\mathrm{S}n),v+w)).$$

- From these assumptions one can derive

$$\forall_n^{\mathrm{c}}\exists_v G(n,v),$$

using double induction (proved in continuation passing style).

Result of demo

- Extracted term

```
[n0]
 (Rec nat=>nat=>(nat=>nat=>nat)=>nat=>nat=>nat)
 n0([n1,k2]k2)
 ([n1,p2,n3,k4]p2(Succ n3)([n7,n8]k4 n8(n7+n8)))
```

  applied to `0`, `([n1,n2]n1)`, `0` and `1`.

- Unclean aspect of this term: recursion operator has value type `nat=>(nat=>nat=>nat)=>nat=>nat=>nat` rather than `(nat=>nat=>nat)=>nat=>nat=>nat`, which would correspond to an iteration.
- We repair this by decoration.

Result of demo (continued)

- Extracted term, after decoration

```
[n0]
 (Rec nat=>(nat=>nat=>nat)=>nat=>nat=>nat)
 n0([k1]k1)
 ([n1,p2,k3]p2([n6,n7]k3 n7(n6+n7)))
```

  applied to `([n1,n2]n1)`, `0` and `1`.

- This is iteration in continuation passing style.

References

- U. Berger, Uniform Heyting arithmetic. APAL 133 (2005).
- D. Ratiu and H.S., Decorating proofs. To appear, Mints volume (S. Feferman and W. Sieg, eds.), 2009.