Extracting computational content from proofs

Helmut Schwichtenberg
(j.w.w. Diana Ratiu)

Mathematisches Institut, LMU, München

National Institute of Informatics, Tokyo, Japan, 13. May 2009

Logic for inductive definitions LID

- Typed language, with the partial continuous functionals as intended domains (cf. Peano arithmetic and $\mathbb{N}$).
- Base types: "lazy" free algebras. Reason: then constructors are injective and have disjoint ranges.
- Terms are those of T^+, a common extension of Gödel's T and Plotkin's PCF.
- Equivalence of terms generated by conversion. Identify equivalent terms.
- All predicates are defined inductively. Examples: totality, Leibniz equality, $\exists$, $\wedge$, $\vee$.
- Natural deduction rules for $\to$ and $\forall$ ("minimal logic").

derivation	term
$u\colon A$	u^A
$\dfrac{\begin{array}{c}[u\colon A]\\ \vert\ M\\ B\end{array}}{A \to B} \to^+ u$	$(\lambda_{u^A} M^B)^{A\to B}$
$\dfrac{\begin{array}{cc}\vert\ M & \vert\ N\\ A \to B & A\end{array}}{B} \to^-$	$(M^{A\to B} N^A)^B$

Natural deduction: $\forall$-rules

derivation	term
$\dfrac{\begin{array}{c}\mid M \\ A\end{array}}{\forall_x A}\,\forall^+ x$ (Variable Cond.)	$(\lambda_x M^A)^{\forall_x A}$ (Variable Cond.)
$\dfrac{\begin{array}{c}\mid M \\ \forall_x A(x)\end{array} \qquad r}{A(r)}\,\forall^-$	$(M^{\forall_x A(x)}\, r)^{A(r)}$

Predicates and formulas

Define $\mathrm{F}(\vec{Y})$, $\mathrm{Preds}(\vec{Y})$, $\mathrm{Cl}_X(\vec{Y})$ (formulas, predicates, clauses, all strictly positive in $\vec{Y}$, with $X, \vec{Y}$ predicate variables).

$$Y_l\vec{r} \in \mathrm{F}(\vec{Y}), \qquad \frac{A \in \mathrm{F} \quad B \in \mathrm{F}(\vec{Y})}{A \to B \in \mathrm{F}(\vec{Y})}, \qquad \frac{A \in \mathrm{F}(\vec{Y})}{\forall_x A \in \mathrm{F}(\vec{Y})},$$

$$\frac{C \in \mathrm{F}(\vec{Y})}{\{\,\vec{x} \mid C\,\} \in \mathrm{Preds}(\vec{Y})}, \qquad \frac{P \in \mathrm{Preds}(\vec{Y})}{P\vec{r} \in \mathrm{F}(\vec{Y})},$$

$$\frac{K_0, \ldots, K_{k-1} \in \mathrm{Cl}_X(\vec{Y})}{\mu_X(K_0, \ldots, K_{k-1}) \in \mathrm{Preds}(\vec{Y})} \quad (k \geq 1),$$

$$\frac{\vec{A} \in \mathrm{F}(\vec{Y}) \qquad \vec{B}_0, \ldots, \vec{B}_{n-1} \in \mathrm{F}}{\forall_{\vec{x}}\big(\vec{A} \to \big(\forall_{\vec{y}_\nu}(\vec{B}_\nu \to X\vec{s}_\nu)\big)_{\nu<n} \to X\vec{t}\,\big) \in \mathrm{Cl}_X(\vec{Y})} \quad (n \geq 0).$$

K_0 must be "nullary" (i.e., no "recursive" premises).

Logic for inductive definitions

LID is the system in minimal logic for $\to$ and $\forall$. Formulas: in F.
Axioms: Consider $I := \mu_X(K_0, \ldots K_{k-1})$. Let

$$K_i(X) := \forall_{\vec{x}}\big(\vec{A} \to \big(\forall_{\vec{y}_\nu}(\vec{B}_\nu \to X\vec{s}_\nu)\big)_{\nu<n} \to X\vec{t}\,\big).$$

Then the corresponding introduction axiom I_i^+ is $K_i(I)$, i.e.,

$$\forall_{\vec{x}}\big(\vec{A} \to \big(\forall_{\vec{y}_\nu}(\vec{B}_\nu \to I\vec{s}_\nu)\big)_{\nu<n} \to I\vec{t}\,\big).$$

The elimination axiom I^- is

$$\forall_{\vec{x}}\big(I\vec{x} \to \big(K_i(I, \{\,\vec{x} \mid C(\vec{x})\,\})\big)_{i<k} \to C(\vec{x})\big),$$

where

$$K(I, \{\,\vec{x} \mid C(\vec{x})\,\}) := \forall_{\vec{x}}\big(\vec{A} \to \big(\forall_{\vec{y}_\nu}(\vec{B}_\nu \to I\vec{s}_\nu)\big)_{\nu<n} \to \big(\forall_{\vec{y}_\nu}(\vec{B}_\nu \to C(\vec{s}_\nu))\big)_{\nu<n} \to C(\vec{t}\,)\big).$$

Example: totality

Totality predicates T_ρ are defined by induction on ρ.

- For base types, e.g. for **N**. Inductive definition, by the clauses

$$T0, \qquad \forall_n(Tn \to T(\mathrm{S}n)).$$

Elimination axiom (writing $\forall_{n\in T}A$ for $\forall_n(Tn \to A)$):

$$\forall_{n\in T}(A(0) \to \forall_{n\in T}(A(n) \to A(\mathrm{S}n)) \to A(n)).$$

This is the induction scheme.

- For $\rho \to \sigma$. Explicit definition (formally: inductive), by

$$\forall_{x^\rho\in T}\, T_\sigma(fx) \to T_{\rho\to\sigma} f,$$

writing $\forall_{x^\rho\in T}A$ for $\forall_{x^\rho}(T_\rho x \to A)$.

Example: Leibniz equality Eq

- Inductively defined by the introduction axiom

$$\forall_x \mathrm{Eq}(x^\rho, x^\rho).$$

- Elimination axiom:

$$\forall_{x,y}\big(\mathrm{Eq}(x,y) \to \forall_x C(x,x) \to C(x,y)\big).$$

- With $C(x,y) := A(x) \to A(y)$ this implies

$$\forall_{x,y}(\mathrm{Eq}(x,y) \to A(x) \to A(y)) \quad \text{(compatibility of } \mathrm{Eq}).$$

- Compatibility gives symmetry and transitivity of Eq.

Ex-Falso-Quodlibet

need not be assumed, but can be proved.

$$\mathbf{F} \to A, \text{ with } \mathbf{F} := \mathrm{Eq}(\mathrm{ff}, \mathrm{tt}) \text{ ("falsity").}$$

The proof is in 2 steps. (1) $\mathbf{F} \to \mathrm{Eq}(x^\rho, y^\rho)$, since from $\mathrm{Eq}(\mathrm{ff}, \mathrm{tt})$ by compatibility

$$\mathrm{Eq}\, \underbrace{[\textbf{if } \mathrm{tt} \textbf{ then } x \textbf{ else } y]}_{x}\, \underbrace{[\textbf{if } \mathrm{ff} \textbf{ then } x \textbf{ else } y]}_{y}.$$

(2) Induction on (the sim. definition of) predicates and formulas.

- Case $I\vec{s}$. Let K_0 be the nullary clause $A_1 \to \cdots \to A_n \to I\vec{t}$. By IH: $\mathbf{F} \to A_i$. Hence $I\vec{t}$. From $\mathbf{F}$ we also obtain $\mathrm{Eq}(s_i, t_i)$, by (1). Hence $I\vec{s}$ by compatibility.
- The cases $A \to B$ and $\forall_x A$ are clear.

Embedding classical arithmetic

- Let $\neg A := (A \to \mathbf{F})$, and

$$\tilde{\exists}_x A := \neg\forall_x \neg A, \qquad A \,\tilde{\vee}\, B := (\neg A \to \neg B \to \mathbf{F}).$$

- Consider a total boolean term $r^{\mathbf{B}}$ as representing a decidable predicate. Let

$$\mathrm{atom}(r) := \mathrm{Eq}(r, \mathrm{tt}).$$

- Prove $\forall_{p\in T}(\neg\neg\mathrm{atom}(p) \to \mathrm{atom}(p))$ by boolean induction.
- Lift this via $\to, \forall$ using

$$\vdash (\neg\neg B \to B) \to \neg\neg(A \to B) \to A \to B,$$
$$\vdash (\neg\neg A \to A) \to \neg\neg\forall_x A \to \forall_x A.$$

- For formulas A built from $\mathrm{atom}(\cdot)$ by $\to, \forall_{x\in T}$ prove stability

$$T(\vec{x}) \to \neg\neg A \to A \qquad (\mathrm{FV}(A) \text{ among } \vec{x}).$$

Examples: ∃, ∧, ∨

are defined inductively by the introduction and elimination axioms

$$\forall_x(A \to \exists_x A),$$
$$\exists_x A \to \forall_x(A \to B) \to B \quad (x \notin \mathrm{FV}(B)),$$

$$A \to B \to A \wedge B,$$
$$A \wedge B \to (A \to B \to C) \to C,$$

$$A \to A \vee B, \qquad B \to A \vee B,$$
$$A \vee B \to (A \to C) \to (B \to C) \to C.$$

Computational content of proofs

- Traditionally arises when the formula contains a strictly positive occurrence of $\exists$, as in $\forall_x \exists_y A(x, y)$.
- For us $\exists$ is inductively defined, and inductive definitions are the only way computational content can arise.
- The computational content of a proof of $I\vec{r}$ is a "generation tree", witnessing how the arguments $\vec{r}$ were put into I.
- For example, consider the clauses

$$\mathrm{Even}(0), \qquad \forall_n(\mathrm{Even}(n) \to \mathrm{Even}(\mathrm{S}(\mathrm{S}n))).$$

A generation tree for $\mathrm{Even}(6)$ should consist of a single branch with nodes $\mathrm{Even}(0)$, $\mathrm{Even}(2)$, $\mathrm{Even}(4)$ and $\mathrm{Even}(6)$.

Computational and non-computational variants of $\to$, $\forall$

- Idea: switch on and off the computational effect of $\to$, $\forall$.
- For instance, in $\forall_n(\mathrm{Even}(n) \to \mathrm{Even}(\mathrm{S}(\mathrm{S}n)))$ only the premise $\mathrm{Even}(n)$ should be computationally relevant, not the $\forall_n$.
- Following Ulrich Berger (1993) we distinguish between a computational $\forall^{\mathrm{c}}$ and non-computational ("uniform") $\forall$.
- Also: allow a computational $\to^{\mathrm{c}}$ and non-computational $\to$.

Example: $\exists$

- For $\exists_x A$ one may decorate in its single clause $\forall_x(A \to \exists_x A)$ independently both, $\forall$ and $\to$.
- This gives four (only) computationally different variants $\exists^{\mathrm{d}}, \exists^{\mathrm{l}}, \exists^{\mathrm{r}}, \exists$ of the existential quantifier, with axioms

$$\begin{array}{ll}
\forall_x^{\mathrm{c}}(A \to^{\mathrm{c}} \exists_x^{\mathrm{d}} A), & \exists_x^{\mathrm{d}} A \to^{\mathrm{c}} \forall_x^{\mathrm{c}}(A \to^{\mathrm{c}} B) \to^{\mathrm{c}} B, \\
\forall_x^{\mathrm{c}}(A \to \exists_x^{\mathrm{l}} A), & \exists_x^{\mathrm{l}} A \to^{\mathrm{c}} \forall_x^{\mathrm{c}}(A \to B) \to^{\mathrm{c}} B, \\
\forall_x(A \to^{\mathrm{c}} \exists_x^{\mathrm{r}} A), & \exists_x^{\mathrm{r}} A \to^{\mathrm{c}} \forall_x(A \to^{\mathrm{c}} B) \to^{\mathrm{c}} B, \\
\forall_x(A \to \exists_x A), & \exists_x A \to \forall_x(A \to B) \to^{\mathrm{c}} B.
\end{array}$$

Similarly for $\wedge$, $\vee$.

Formulas as computational problems (Kolmogorov)

- 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.
- $\tau(A)$ should be the type of the term (or "program") to be extracted from a proof of A.
- Formally, we assign to every formula A an object $\tau(A)$ (a type or the nulltype symbol ε).
- In case $\tau(A) = \varepsilon$ proofs of A have no computational content; such formulas A are called computationally irrelevant (c.i.); the other ones computationally relevant (c.r.).

The type of a formula

Extend the use of $\rho \to \sigma$ and $\rho \times \sigma$ to the nulltype symbol ε:

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

Define

$\tau(I\vec{r}) := \varepsilon$ for I not requiring witnesses (e.g., Eq),
$\tau(A \to^{\mathrm{c}} B) := \tau(A) \to \tau(B), \quad \tau(A \to B) := \tau(B),$
$\tau(\forall^{\mathrm{c}}_{x^\rho} A) := \rho \to \tau(A), \quad \tau(\forall_{x^\rho} A) := \tau(A),$
$\tau(\exists^{\mathrm{d}}_{x^\rho} A) := \rho \times \tau(A),\ \tau(\exists^{\mathrm{l}}_{x^\rho} A) := \rho,\ \tau(\exists^{\mathrm{r}}_{x^\rho} A) := \tau(A),\ \tau(\exists_{x^\rho} A) := \varepsilon$

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

Computational variables of a derivation

For M^A with A c.i. let $\mathrm{CV}(M) := \emptyset$. Assume A is c.r. Then

$$
\begin{aligned}
\mathrm{CV}(u^A) &:= \{x_u^{\tau(A)}\} \quad (x_u^{\tau(A)} \text{ uniquely associated with } u^A),\\
\mathrm{CV}(\lambda_{u^A} M^B)^{A\to^{\mathrm{c}} B} &:= \mathrm{CV}(M) \setminus \{x_u^{\tau(A)}\},\\
\mathrm{CV}(M^{A\to^{\mathrm{c}} B} N^A)^B &:= \mathrm{CV}(M) \cup \mathrm{CV}(N),\\
\mathrm{CV}(\lambda_{x^\rho} M^A)^{\forall^{\mathrm{c}}_x A} &:= \mathrm{CV}(M) \setminus \{x^\rho\},\\
\mathrm{CV}(M^{\forall^{\mathrm{c}}_x A(x)} r)^{A(r)} &:= \mathrm{CV}(M) \cup \mathrm{FV}(r),\\
\mathrm{CV}(\lambda_{u^A} M^B)^{A\to B} &:= \mathrm{CV}(M^{A\to B} N^A)^B := \mathrm{CV}(\lambda_{x^\rho} M^A)^{\forall_x A}\\
&:= \mathrm{CV}(M^{\forall_x A(x)} r)^{A(r)} := \mathrm{CV}(M).
\end{aligned}
$$

Correct derivations

Restrictions to $\to^+$ and $\forall^+$: consider

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

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

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

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

Computational strengthening

Define a relation $A_1 \sqsupseteq A_2$ (A_1 is a computational strengthening of A_2) between c.r. formulas A_1, A_2 inductively. It is reflexive, transitive and satisfies

$$
\begin{array}{l}
(A \to B) \sqsupseteq (A \to^{\mathrm{c}} B), \\
(A \to^{\mathrm{c}} B) \sqsupseteq (A \to B) \quad \text{if } A \text{ is c.i.}, \\
(A \overset{\breve{}}{\to} B_1) \sqsupseteq (A \overset{\breve{}}{\to} B_2) \quad \text{if } B_1 \sqsupseteq B_2, \text{ with } \overset{\breve{}}{\to} \in \{\to^{\mathrm{c}}, \to\}, \\
(A_2 \overset{\breve{}}{\to} B) \sqsupseteq (A_1 \overset{\breve{}}{\to} B) \quad \text{if } A_1 \sqsupseteq A_2, \text{ with } \overset{\breve{}}{\to} \in \{\to^{\mathrm{c}}, \to\}, \\
\forall_x A \sqsupseteq \forall^{\mathrm{c}}_x A, \\
\breve{\forall}_x A_1 \sqsupseteq \breve{\forall}_x A_2 \quad \text{if } A_1 \sqsupseteq A_2, \text{ with } \breve{\forall} \in \{\forall^{\mathrm{c}}, \forall\}.
\end{array}
$$

and similarly for $\exists$, $\wedge$, $\vee$.

$$\text{If } A_1 \sqsupseteq A_2, \text{ then } \vdash A_1 \to^{\mathrm{c}} A_2.$$

Realizability

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

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

We define the formula $t\ \mathbf{r}\ A$, to be read t realizes A. This formula is "invariant" in the sense that $\varepsilon\ \mathbf{r}\ (t\ \mathbf{r}\ A)$ and $t\ \mathbf{r}\ A$ are identical.

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

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

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

Notice that $\mathrm{CV}(M) = \mathrm{FV}([\![M]\!])$.

Extracted terms for axioms

- Consider

$$\forall^{\mathrm{c}}_v(A(\mathrm{nil}) \to^{\mathrm{c}} \forall^{\mathrm{c}}_{x,v}(A(v) \to^{\mathrm{c}} A(xv)) \to^{\mathrm{c}} A(v)),$$

with x, v variables of type $\rho, \mathbf{L}(\rho)$ and xv denoting $\mathrm{cons}(x, v)$. We write $\forall^{\mathrm{c}}_v A$ for $\forall_v(Tv \to^{\mathrm{c}} A)$ etc.

- The extracted term is the corresponding recursion operator in the sense of Gödel (1958), of type

$$\mathcal{R}^{\tau}_{\mathbf{L}(\rho)} \colon \mathbf{L}(\rho) \to \tau \to (\rho \to \mathbf{L}(\rho) \to \tau \to \tau) \to \tau$$

where $\tau := \tau(A)$.

Soundness

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

$$\begin{cases} x_{u_i}\ \mathbf{r}\ C_i & \text{for } \tau(C_i) \neq \varepsilon \text{ and } x_{u_i} \in \mathrm{CV}(M) \\ \exists_x(x\ \mathbf{r}\ C_i) & \text{for } \tau(C_i) \neq \varepsilon \text{ and } x_{u_i} \notin \mathrm{CV}(M) \\ \varepsilon\ \mathbf{r}\ C_i & \text{for } \tau(C_i) = \varepsilon. \end{cases}$$

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 $L\colon A \vee B$.
- Since $[\![L]\!]$ is a boolean, the extract of the modified proof is "purified": the (possibly large) extract $[\![L^{\mathrm{d}}]\!]$ has disappeared.

Decorating proofs

Goal: Insertion of as few as possible decorations into a proof.
Write $\forall^{\mathrm{c}}_n A$ for $\forall_n(T_{\mathbf{N}} n \to^{\mathrm{c}} A)$.

- $\mathrm{Seq}(M)$ 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 $\to$, $\forall$, except "uninstantiated" formulas of axioms, e.g., $\forall^{\mathrm{c}}_n(P0 \to^{\mathrm{c}} \forall^{\mathrm{c}}_n(Pn \to^{\mathrm{c}} P(\mathrm{S}n)) \to^{\mathrm{c}} Pn)$.
- 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 (1) N and M are the same up to variants of $\to$, $\forall$ in their formulas, and (2) every c.r. formula of M is extended by the corresponding one in N.

Decoration algorithm

Assumption: We have an algorithm assigning to every axiom A and every decoration variant C of A another axiom whose formula D extends C, and D is the least among those extensions.

Theorem (Ratiu, H.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 $\Delta_3,\Sigma_2 \Rightarrow C_3$. . . .

Example: list reversal (Ulrich Berger)

Define the graph Rev of the list reversal function inductively, by

$$\mathrm{Rev}(\mathrm{nil},\mathrm{nil}), \tag{1}$$

$$\mathrm{Rev}(v,w) \to \mathrm{Rev}(vx,xw). \tag{2}$$

We prove weak existence of the reverted list:

$$\forall^{\mathrm{c}}_v \tilde{\exists}^{\mathrm{l}}_w \mathrm{Rev}(v,w) \qquad (:= \forall^{\mathrm{c}}_v(\forall^{\mathrm{c}}_w(\mathrm{Rev}(v,w)\to\bot)\to^{\mathrm{c}}\bot)).$$

Fix v and assume $u\colon \forall^{\mathrm{c}}_w \neg\mathrm{Rev}(v,w)$. To show $\bot$. To this end we prove that all initial segments v_1 of v are non-revertible, which contradicts (1). More precisely, from u and (2) we prove

$$\forall^{\mathrm{c}}_{v_2} A(v_2), \quad A(v_2) := \forall^{\mathrm{c}}_{v_1}(v_1 v_2 = v \to \forall^{\mathrm{c}}_w \neg\mathrm{Rev}(v_1,w))$$

by induction on v_2. Base $v_2 = \mathrm{nil}$: Use u. Step. Assume $v_1(xv_2) = v$, fix w and assume further $\mathrm{Rev}(v_1,w)$. Properties of the append function imply that $(v_1x)v_2 = v$. IH for v_1x gives $\forall^{\mathrm{c}}_w \neg\mathrm{Rev}(v_1x,w)$. Now (2) yields $\bot$.

Results of demo

- Weak existence proof formalized.
- Translated into an existence proof. Extracted algorithm: $f(v_1) := h(v_1, \mathrm{nil}, \mathrm{nil})$ with

$$h(\mathrm{nil}, v_2, v_3) := v_3, \quad h(xv_1, v_2, v_3) := h(v_1, v_2x, xv_3).$$

  The second argument of h is not needed, but makes the algorithm quadratic. (In each recursion step v_2x is computed, and the list append function is defined by recursion over its first argument.)
- Optimal decoration of existence proof computed. Extracted algorithm: $f(v_1) := g(v_1, \mathrm{nil})$ with

$$g(\mathrm{nil}, v_2) := v_2, \quad g(xv_1, v_2) := g(v_1, xv_2).$$

  This is the well-known linear algorithm, with an accumulator.

Example: avoiding factorization

Let Pn mean "n is prime". Consider

$$\texttt{Fact}: \forall_n^{\mathrm{c}}(Pn \vee^{\mathrm{r}} \exists^{\mathrm{l}}_{m,k>1}(n = mk)) \quad \text{factorization,}$$

$$\texttt{PTest}: \forall_n^{\mathrm{c}}(Pn \vee \exists^{\mathrm{l}}_{m,k>1}(n = mk)) \quad \text{prime number test.}$$

($\exists^{\mathrm{d}}_n A := \exists_n(Tn \wedge^{\mathrm{d}} A)$ and $\exists^{\mathrm{l}}_{m,k>1} A := \exists_{m,k>1}(Tm \wedge^{\mathrm{d}} (Tk \wedge^{\mathrm{l}} A))$).
Euler's φ has the properties

$$\begin{cases} \varphi(n) = n-1 & \text{if } Pn, \\ \varphi(n) < n-1 & \text{if } n \text{ is composed.} \end{cases}$$

Using factorization and these properties we obtain a proof of

$$\forall_n^{\mathrm{c}}(\varphi(n) = n-1 \vee \varphi(n) < n-1).$$

Goal: get rid of the expensive factorization algorithm in the computational content, via decoration.

Example: avoiding factorization (ctd.)

How could the better proof be found? We have

$$\texttt{Fact}: \forall^{\mathrm{c}}_n(Pn \vee^{\mathrm{r}} \exists^{\mathrm{l}}_{m,k>1}(n = mk)),$$
$$\texttt{PTest}: \forall^{\mathrm{c}}_n(Pn \vee \exists^{\mathrm{l}}_{m,k>1}(n = mk)).$$

- The decoration algorithm arrives at `Fact` with

$$Pn \vee \exists^{\mathrm{l}}_{m,k>1}(n = mk).$$

- `PTest` fits as well, and it has $\vee$ rather than $\vee^{\mathrm{r}}$, hence is preferred.

Example: maximal segment problem

Due to Bates and Constable (1985).

- Let X be linearly ordered by $\leq$. Given

$$\mathrm{seg}\colon \mathbf{N} \to \mathbf{N} \to X.$$

(Example: $X = \mathbb{Z}$ and $\mathrm{seg}(i,k) = f(i) + \cdots + f(k)$ for some $f\colon \mathbf{N} \to \mathbb{Z}$.)

- Want: maximal segment

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

- Special case: maximal end segment

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

Example: maximal segment problem (ctd.)

2 proofs of the existence of a maximal end segment for $n+1$

$$\forall_n^{\mathrm{c}} \exists_{j \le n+1}^{\mathrm{l}} \forall_{j' \le n+1} (\mathrm{seg}(j', n+1) \le \mathrm{seg}(j, n+1)).$$

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

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

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

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

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: maximal segment problem (ctd.)

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

$$\forall_n^{\mathrm{c}}(\exists^{\mathrm{l}}_{i\le k\le n}\forall_{i'\le k'\le n}(\mathrm{seg}(i',k')\le \mathrm{seg}(i,k))\wedge^{\mathrm{d}}$$
$$\exists^{\mathrm{l}}_{j\le n}\forall_{j'\le n}(\mathrm{seg}(j',n)\le \mathrm{seg}(j,n)))$$

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: maximal segment problem (ctd.)

How could the better proof be found? We have

$$\mathtt{L1}: \forall_n^{\mathrm{c}} \exists_{j \le n+1}^{\mathrm{l}} \forall_{j' \le n+1} (\mathrm{seg}(j', n+1) \le \mathrm{seg}(j, n+1)),$$
$$\mathtt{L2}: \forall_n (\mathtt{ES}_n \to^{\mathrm{c}} \mathtt{Mon} \to \exists_{j \le n+1}^{\mathrm{l}} \forall_{j' \le n+1} (\mathrm{seg}(j', n+1) \le \mathrm{seg}(j, n+1))).$$

- The decoration algorithm arrives at L1 with

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

- L2 fits as well, its assumptions $\mathtt{ES}_n$ and Mon are in the context, and it has $\forall_n$ rather than $\forall_n^{\mathrm{c}}$, hence is preferred.

References

- U. Berger, Program extraction from normalization proofs. In: Proc. TLCA 1993 (Springer LNCS 664).
- U. Berger, Uniform Heyting arithmetic. APAL 133 (2005) 125–148.
- D. Ratiu and H.S., Decorating proofs. To appear, Mints volume (ed. S. Feferman and W. Sieg), 2009.