Decorating proofs

Helmut Schwichtenberg (with Diana Ratiu)

Mathematisches Institut, LMU, München

Classical Logic and Computation, Reykjavik, 13. July 2008

Why extract computational content from proofs?

- Proofs are machine checkable $\Rightarrow$ no logical errors.
- Program on the proof level $\Rightarrow$ maintenance becomes easier. Possibility of program development by proof transformation (Goad 1980).
- Discover unexpected content:
  - U. Berger 1993: Tait's proof of the existence of normal forms for the typed λ-calculus $\Rightarrow$ "normalization by evaluation".
  - Content in weak (or "classical") existence proofs, of

$$\tilde{\exists}_x A := \neg\forall_x\neg A,$$

via proof interpretations: (refined) A-translation or Gödel's Dialectica interpretation.

Falsity as a predicate variable $\bot$

In some proofs no knowledge about **F** is required. Then a predicate variable $\bot$ instead of **F** will do, and we can define

$$\tilde{\exists}_y G := \forall_y (G \to \bot) \to \bot.$$

Why is this of interest? We can substitute an arbitrary formula for $\bot$, for instance, $\exists_y G$. Then a proof of $\tilde{\exists}_y G$ is turned into a proof of

$$\forall_y (G \to \exists_y G) \to \exists_y G.$$

As the premise is provable, we have a proof of $\exists_y G$. – (A-translation; H. Friedman 1978, Dragalin 1979).

Problems

Unfortunately, this simple argument is not quite correct.

- G may contain $\bot$, and hence is changed under the substitution $\bot \mapsto \exists_y G$.
- We may have used axioms or lemmata involving $\bot$ (e.g., $\bot \to P$), which need not be derivable after the substitution.

But in spite of this, the simple idea can be turned into something useful. Assume that

- the lemmata $\vec{D}$ and the goal formula G are such that we can derive $\vec{D} \to D_i[\bot := \exists_y G]$, $G[\bot := \exists_y G] \to \exists_y G$.
- the substitution $\bot \mapsto \exists_y G$ turns the axioms into instances of the same scheme with different formulas, or else into derivable formulas.

Problems (continued)

Then from our given derivation (in minimal logic) of $\vec{D} \to \forall_y(G \to \bot) \to \bot$ we obtain

$$\vec{D}[\bot := \exists_y G] \to \forall_y(G[\bot := \exists_y G] \to \exists_y G) \to \exists_y G.$$

Now $\vec{D} \to D_i[\bot := \exists_y G]$ allows to drop the substitution in $\vec{D}$, and by $G[\bot := \exists_y G] \to \exists_y G$ the second premise is derivable. Hence we obtain as desired

$$\vec{D} \to \exists_y G.$$

Definite and goal formulas

A formula is relevant if it "ends" with $\bot$. More precisely:

- $\bot$ is relevant,
- if C is relevant and B is arbitrary, then $B \to C$ is relevant, and
- if C is relevant, then $\forall_x C$ is relevant.

We define goal formulas G and definite formulas D inductively.
P ranges over prime formulas (including $\bot$).

$$G ::= P \mid D \to G \quad \text{if } G \text{ relevant \& } D \text{ irrelevant} \Rightarrow D \text{ quantifier-free}$$
$$\mid \forall_x G \quad \text{if } G \text{ irrelevant,}$$

$$D ::= P \mid G \to D \quad \text{if } D \text{ irrelevant} \Rightarrow G \text{ irrelevant}$$
$$\mid \forall_x D.$$

Let $A^{\mathbf{F}}$ denote $A[\bot := \mathbf{F}]$.

Properties of definite and goal formulas

Lemma

For definite formulas D and goal formulas G we have derivations from $\mathbf{F} \to \bot$ of

$$((D^{\mathbf{F}} \to \mathbf{F}) \to \bot) \to D \quad \text{for } D \text{ relevant},$$
$$D^{\mathbf{F}} \to D,$$
$$G \to G^{\mathbf{F}} \quad \text{for } G \text{ irrelevant},$$
$$G \to (G^{\mathbf{F}} \to \bot) \to \bot.$$

Lemma

For goal formulas $\vec{G} = G_1, \ldots, G_n$ we have a derivation from $\mathbf{F} \to \bot$ of

$$(\vec{G}^{\mathbf{F}} \to \bot) \to \vec{G} \to \bot.$$

Elimination of $\bot$ from weak existence proofs

Assume that for arbitrary formulas $\vec{A}$, definite formulas $\vec{D}$ and goal formulas $\vec{G}$ we have a derivation of

$$\vec{A} \to \vec{D} \to \forall_{\vec{y}}(\vec{G} \to \bot) \to \bot.$$

Then we can also derive

$$(\mathbf{F} \to \bot) \to \vec{A} \to \vec{D}^{\mathbf{F}} \to \forall_{\vec{y}}(\vec{G}^{\mathbf{F}} \to \bot) \to \bot.$$

In particular, substitution of the formula

$$\exists_{\vec{y}} \vec{G}^{\mathbf{F}} := \exists_{\vec{y}}(G_1^{\mathbf{F}} \land \cdots \land G_n^{\mathbf{F}})$$

for $\bot$ yields

$$\vec{A}[\bot := \exists_{\vec{y}} \vec{G}^{\mathbf{F}}] \to \vec{D}^{\mathbf{F}} \to \exists_{\vec{y}} \vec{G}^{\mathbf{F}}.$$

The type of a formula

- Every formula A can be seen as a computational problem (Kolmogorov). We define $\tau(A)$ as the type of a potential realizer of A, i.e., the type of the term to be extracted from a proof of A.
- Assign $A \mapsto \tau(A)$ (a type or the "nulltype" symbol ε). In case $\tau(A) = \varepsilon$ proofs of A have no computational content.

$$\tau(T(x)) := \tau(\mathrm{Eq}(x,y)) := \varepsilon, \quad \tau(\exists_{x^\rho} A) := \begin{cases} \rho & \text{if } \tau(A) = \varepsilon \\ \rho \times \tau(A) & \text{otherwise,} \end{cases}$$

$$\tau(A \to B) := (\tau(A) \to \tau(B)), \quad \tau(\forall_{x^\rho} A) := (\rho \to \tau(A)),$$

with the convention

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

Realizability

Let A be a formula and z either a variable of type $\tau(A)$ if it is a type, or the nullterm symbol ε if $\tau(A) = \varepsilon$. We define the formula $z\ \mathbf{r}\ A$, to be read z realizes A:

$$
\begin{aligned}
z\ \mathbf{r}\ \mathrm{Eq}(r,s) &:= \mathrm{Eq}(r,s),\\
z\ \mathbf{r}\ T(r) &:= T(r),\\
z\ \mathbf{r}\ \exists_x A(x) &:= \begin{cases} A(z) & \text{if } \tau(A) = \varepsilon \\ z_0\ \mathbf{r}\ A(z_1) & \text{otherwise,} \end{cases}\\
z\ \mathbf{r}\ (A \to B) &:= \forall_x (x\ \mathbf{r}\ A\ \to\ zx\ \mathbf{r}\ B),\\
z\ \mathbf{r}\ \forall_x A &:= \forall_x\ zx\ \mathbf{r}\ A,
\end{aligned}
$$

with the convention $\varepsilon x := \varepsilon$, $z\varepsilon := z$, $\varepsilon\varepsilon := \varepsilon$.

Extracted terms

For derivations M^A with $\tau(A) = \varepsilon$ let $[\![M]\!] := \varepsilon$ (nullterm symbol).
Now assume that M derives a formula A with $\tau(A) \neq \varepsilon$.

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

Extracted terms for axioms

The extracted term of an induction axiom is defined to be a recursion operator. For example, in case of an induction scheme

$$\mathrm{Ind}_{n,A}\colon \forall_m\big(A(0) \to \forall_n(A(n) \to A(\mathrm{S}n)) \to A(m^{\mathbf{N}})\big)$$

we have

$$[\![\mathrm{Ind}_{n,A}]\!] := \mathcal{R}^{\tau}_{\mathbf{N}}\colon \mathbf{N} \to \tau \to (\mathbf{N} \to \tau \to \tau) \to \tau \quad (\tau := \tau(A) \neq \varepsilon).$$

Soundness

Theorem

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 $\bar{u}_i : x_{u_i}\ \mathbf{r}\ C_i$.

Proof.

Induction on M. □

Uniform universal quantifier $\forall^{\mathsf{U}}$ and implication $\to^{\mathsf{U}}$

- We want to select relevant parts of the computational content of a proof.
- This will be possible if some "uniformities" hold. Use a uniform variant $\forall^{\mathsf{U}}$ of $\forall$ (U. Berger 2005) and $\to^{\mathsf{U}}$ of $\to$.
- Both are governed by the same rules as the non-uniform ones. However, we will put some uniformity conditions on a proof to ensure that the extracted computational content is correct.

Extending the definitions of $\tau(A)$ and $z\ \mathbf{r}\ A$

- The definition of the type $\tau(A)$ of a formula A is extended by the two clauses

$$\tau(A \to^{\mathsf{U}} B) := \tau(B), \quad \tau(\forall^{\mathsf{U}}_{x^\rho} A) := \tau(A).$$

- The definition of realizability is extended by

$$z\ \mathbf{r}\ (A \to^{\mathsf{U}} B) := (A \to z\ \mathbf{r}\ B), \quad z\ \mathbf{r}\ (\forall^{\mathsf{U}}_{x} A) := \forall_x\, z\ \mathbf{r}\ A.$$

Extracted terms and uniform proofs

We define the extracted term of a proof, and (using this concept) the notion of a uniform proof, which gives a special treatment to the uniform universal quantifier $\forall^{\mathsf{U}}$ and uniform implication $\to^{\mathsf{U}}$.

More precisely, for a proof M we simultaneously define

- its extracted term $[\![M]\!]$, of type $\tau(A)$, and
- when M is uniform.

Extracted terms and uniform proofs (continued)

For derivations M^A where $\tau(A) = \varepsilon$ let $[\![M]\!] := \varepsilon$ (the nullterm symbol); every such M is uniform. Now assume that M derives a formula A with $\tau(A) \neq \varepsilon$. Then

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

In all these cases uniformity is preserved, except possibly in those involving λ:

Extracted terms and uniform proofs (continued)

Consider

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

$(\lambda_{u^A} M)^{A\to^{\mathsf{U}} B}$ is uniform if M is and $x_u \notin \mathrm{FV}([\![M]\!])$. Similarly: Consider

$$\dfrac{\begin{array}{c}\mid M\\ A\end{array}}{\forall^{\mathsf{U}}_x A}\,(\forall^{\mathsf{U}})^{+}\,x \qquad \text{or as term} \quad (\lambda_x M)^{\forall^{\mathsf{U}}_x A} \qquad \text{(VarC)}.$$

$(\lambda_x M)^{\forall^{\mathsf{U}}_x A}$ is uniform if M is and $x \notin \mathrm{FV}([\![M]\!])$.

Why uniformity?

- Suppose that in a proof M we have made use of a case distinction based on a lemma stating a disjunction: $L: A \vee B$.
- Then the extract $[\![M]\!]$ will contain the extract $[\![L]\!]$ of the proof of the auxiliary lemma, which may be large.
- Suppose further that in the proof M, the only computationally relevant use of the lemma was which one of the two alternatives holds true, A or B.
- We can express this fact by using a weakened form of the lemma instead: $L': A \vee^{\mathsf{U}} B$.
- Since the extract $[\![L']\!]$ is a boolean, the extract of the modified proof has been "purified" in the sense that the (possibly large) extract $[\![L]\!]$ has disappeared.

Decorating proofs

Goal: "optimal" insertion of uniformity marks into a proof.

- The sequent $\mathrm{Seq}(M)$ of a proof M consists of its context and its end formula.
- The uniform proof pattern $\mathrm{UP}(M)$ of a proof M is the result of changing in M all occurrences of $\to, \forall, \exists, \wedge$ in its formulas into their uniform counterparts $\to^{\mathsf{U}}, \forall^{\mathsf{U}}, \exists^{\mathsf{U}}, \wedge^{\mathsf{U}}$, except the uninstantiated formulas of axioms and theorems.
- A formula D extends C if D is obtained from C by changing some connectives into one of their more informative versions, according to the following ordering: $\to^{\mathsf{U}} \leq \to$, $\forall^{\mathsf{U}} \leq \forall$, $\exists^{\mathsf{U}} \leq \exists^{\mathsf{L}}, \exists^{\mathsf{R}} \leq \exists$ and $\wedge^{\mathsf{U}} \leq \wedge^{\mathsf{L}}, \wedge^{\mathsf{R}} \leq \wedge$.

Decorating proofs (continued)

- A proof N extends M if (1) $\mathrm{UP}(M) = \mathrm{UP}(N)$, and (2) each formula in N extends the corresponding one in M. In this case $\mathrm{FV}([\![N]\!])$ is essentially (i.e., up to extensions of assumption formulas) a superset of $\mathrm{FV}([\![M]\!])$.
- Every proof M whose uniform proof pattern $\mathrm{UP}(M)$ is U is called a decoration of U.

Decoration algorithm

We define a decoration algorithm, assigning to every uniform proof pattern U and every extension of its sequent an "optimal" decoration M_∞ of U, which further extends the given extension. Need such an algorithm for every axiom. Examle: induction.

$$\mathrm{Ind}_{n,A} : \forall_m\big(A(0) \to \forall_n(A(n) \to A(\mathrm{S}n)) \to A(m^{\mathbf{N}})\big).$$

- The given extension of the four A's might be different. One needs to pick their "least upper bound" as further extension.
- If $\tau(A) \neq \varepsilon$, the $\to, \forall$ must be made proper.

Decoration algorithm

Theorem (Ratiu, S)

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_∞.*

Proof, by induction on U

Case $(\to^{\mathsf{U}})^-$. Consider a uniform proof pattern

$$\dfrac{\begin{array}{c}\Phi,\Gamma \\ \mid U \\ A \to^{\mathsf{U}} B\end{array} \qquad \begin{array}{c}\Gamma,\Psi \\ \mid V \\ A\end{array}}{B}\,(\to^{\mathsf{U}})^-$$

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

- $\mathrm{IH}_a(U)$ for extension $\Pi,\Delta \Rightarrow A \to^{\mathsf{U}} D \mapsto$ decoration M_1 of U whose sequent $\Pi_1,\Delta_1 \Rightarrow C_1 \overset{\smile}{\to} D_1$ extends $\Pi,\Delta \Rightarrow A \to^{\mathsf{U}} D$.
- $\mathrm{IH}_a(V)$ 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 (U. Berger)

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

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

$$\mathrm{Rev}(v, w) \to \mathrm{Rev}(v :+: x:, x :: w). \tag{2}$$

We prove weak existence of the reverted list:

$$\forall_{v \in T} \tilde{\exists}_{w \in T} \mathrm{Rev}(v, w) \qquad (:= \forall_{v \in T}(\forall_{w \in T}(\mathrm{Rev}(v, w) \to \bot) \to \bot)).$$

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

$$\forall_{v_2 \in T} A(v_2), \quad A(v_2) := \forall_{v_1 \in T}(v_1 :+: v_2 = v \to \forall_{w \in T} \neg \mathrm{Rev}(v_1, w))$$

by induction on v_2. Base $v_2 = \mathrm{nil}$: Use u. Step. Assume $v_1 :+: (x :: v_2) = v$, fix w and assume further $\mathrm{Rev}(v_1, w)$. Properties of the append function imply that $(v_1 :+: x:) :+: v_2 = v$. IH for $v_1 :+: x:$ gives $\forall_{w \in T} \neg \mathrm{Rev}(v_1 :+: x:, w)$. 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

  $$h(\mathrm{nil}, v_2, v_3) := v_3, \quad h(x :: v_1, v_2, v_3) := h(v_1, v_2{:+:}x{:}, x :: v_3).$$

  The second argument of h is not needed, but makes the algorithm quadratic. (In each recursion step $v_2 :+: x:$ 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

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

  This is the usual linear algorithm, with an accumulator.

Future work

- Explore applications of refined A-translation and automated decoration: Combinatorics, Gröbner bases (Diana Ratiu).
- Logic of inductive definitions: Include formal neighborhoods into the language (Basil Karadais).
- Compare refined A-translation and Gödel's Dialectica interpretation (Trifon Trifonov).