Program Extraction (Part 2)

Kenji Miyamoto

3.10 – 8.10 Aurachhof, Fischbachau

Review

- Formal theory TCF for program extraction
  - first-order minimal logic + inductive definitions
  - an extension of Gödel's T as term calculus
  - predicates and formulas
    - introduction and elimination axioms I_i^+, I^-.
  - natural deduction represented in proof term
  - decoration "nc" for fine-tuning extracted programs

Today's topics

- meaning of "nc", notion of proofs involving $\to^{nc}, \forall^{nc}$.
- computational solution given in our term language.
  - once the formula (i.e. problem) is given, we can compute the type of the computational solution
  - formalizing the relation
    "--- is a computational solution of ---"
    by means of Kreisel's realizability interpretation.
  - defining program extraction
  - Soundness Theorem.

Informal meaning of $\rightarrow^{nc}$

In order to prove $A \rightarrow^{nc} B$, we can assume A and derive B using the assumption A, but it is **not** allowed to use A computationally.

example $A \rightarrow^{nc} A$

— Same idea works for $\forall^{nc}$, too.

— $(\rightarrow^{nc})^-$ and $(\forall^{nc})^-$ are same as $\rightarrow^-$ and $\forall^-$

Computational assumption variables

For a derivation M^A,

$CA(M^A) := \emptyset$ if A is n.c. (non-computational)

$CA(u^A) := \{u\}$, $CA(c^A) := \emptyset$ (c^A is an axiom)

$$CA\left(\left(\lambda u^A M^B\right)^{A \to^{(nc)} B}\right) := CA(M) \setminus \{u\}$$

$$CA\left(\left(M^{A\to B} N^A\right)^B\right) := CA(M) \cup CA(N)$$

$$CA\left(\left(M^{A \to^{nc} B} N^A\right)^B\right) := CA(M)$$

$$CA\left(\left(\lambda x M^{A(x)}\right)^{\forall_x^{(nc)} A(x)}\right) := CA(M), \quad CA\left(\left(M^{\forall_x^{(nc)} A(x)} t\right)^{A(t)}\right) := CA(M)$$

– Additional condition to $(\to^{nc})^+$

$$\frac{\begin{array}{c}[u:A]\\ \vdots M \\ B\end{array}}{A \to^{nc} B} \to^{nc}, u \qquad u \notin CA(M^A)$$

Computational object variables

$CV(M^A) := \emptyset$ if A is n.c.

$CV(u^A) := CV(C^A) := \emptyset$.

$CV((\lambda u^A M^B)^{A \to^{(nc)} B}) := CV(M)$

$CV((M^{A \to B} N^A)^B) := CV(M) \cup CV(N)$

$CV((M^{A \to^{nc} B} N^A)^B) := CV(M)$

$CV((\lambda x M^{A(x)})^{\forall_x^{(nc)} A(x)}) := CV(M) \setminus \{x\}$

$CV((M^{\forall_x A(x)} t)^{A(t)}) := CV(M) \cup FV(t)$

$CV((M^{\forall_x^{nc} A(x)} t)^{A(t)}) := CV(M)$

- Additional cond. to $(\forall^{nc})^+$.

$$\dfrac{\begin{array}{c}|\,M\\ A\end{array}}{\forall_x^{nc} A}(\forall^{nc})^+, x \qquad x \notin CV(M^A)$$

Examples of idpcs involving nc

- $\mathrm{Ex}D_0^+ : \forall_x^\rho(Y(x) \to \mathrm{Ex}D_Y)$ where the arity of $\mathrm{rs}(\rho)$.

  $\mathrm{Ex}D^- : \mathrm{Ex}D_Y \to \forall_x(Yx \to P) \to P$

  $\exists_x Y(x) := \mathrm{Ex}D_Y$ (D stands for "double")

  using $\forall^{nc}$, we obtain $\mathrm{Ex}R$, (R stands for "right")

  $\exists_x^r Y(x) := \mathrm{Ex}R_Y$.

  using $\to^{nc}$, we obtain $\mathrm{Ex}L$ (L for "left")

  $\exists_x^l Y(x) := \mathrm{Ex}L_Y$

  using $\forall^{nc}$ and $\to^{nc}$, and also μ^{nc}, we obtain $\mathrm{Ex}Nc$ (Nc for n.c.)

  $\exists_x^{nc} Y(x) := \mathrm{Ex}Nc\,Y$

- $\mathrm{And}D_0^+ : A \to B \to \mathrm{And}D(A,B)$

  $\mathrm{And}D^- : \mathrm{And}D(A,B) \to (A \to B \to P) \to P$.

  we have L, R, Nc variants as well, and write $\wedge, \wedge^r, \wedge^l, \wedge^{nc}$

- $\mathrm{OrD}_0^+ : A \to \mathrm{OrD}(A,B)$

  $\mathrm{OrD}_1^+ : B \to \mathrm{OrD}(A,B)$

  $\mathrm{OrD}^- : \mathrm{OrD}(A,B) \to (A \to P) \to (B \to P) \to P$

  OrL and OrR in the same idea. Write $\vee, \vee^l, \vee^r$

- OrNC can be defined, but OrNC^- is restricted to dense non-computational formulas. Write $\vee^{nc}$

- $\mathrm{OrU}_0^+ : A \to^{nc} \mathrm{OrU}(A,B)$

  $\mathrm{OrU}_1^+ : B \to^{nc} \mathrm{OrU}(A,B)$

  $\mathrm{OrU}^- : \mathrm{OrU}(A,B) \to (A \to^{nc} P) \to (B \to^{nc} P) \to P$

  U stands for "uniform" write $\vee^u$.

- in any I^-, free variables in the axiom formula should be quantified by $\forall^{nc}$.

— Leibniz Equality "eqd". we define it by μ^{nc}

$$eqd^+_0 : \forall^{nc}_{x^\rho}(x\ eqd\ x)$$

$$eqd^- : x\ eqd\ y \to \forall^{nc}_{x^\rho}(Q\,x\,x) \to Q\,x\,y$$

- We can prove $\forall_{x,y}(x\ eqd\ y \to Px \to Py)$
- We can define the arithmetic falsity $\mathbb{F} := tt^{\mathbb{B}}\ eqd\ ff^{\mathbb{B}}$
- We can prove ex-falso-quodlibet for A.

Thm $Efq\,A := \forall_{\vec{I}}(\mathbb{F} \to A)$, assuming $Efq\,r$ (r prar in A) and Efq_I (I tdpl without a nullary I^+_i)

just sketch we can prove $x\ eqd\ y$ for any x^ρ, y^ρ,
Then by induction on the construction of A.

Type of computational solutions

For a given formula A we define the type of A.

We use the symbol " o " (null type) such as

$$o \to \rho := \rho \quad , \quad \rho \to o := o \quad , \quad o \to o := o \; .$$

We define $\tau(A)$ for any formula A.

$\tau(A) := o$ if A is n.c.

$\tau(P) := \xi_P \quad , \quad \tau(\{\vec{x} \mid A\}) = \tau(A)$

$$\tau\left(\mu_X\left(\forall^{nc}_{\vec{x_i}} \forall^{\vec{\rho_i}}_{\vec{y_i}} \left(\vec{A_i} \to^{nc} \vec{B_i} \to X\vec{t_i}\right)\right)_{i<k}\right) := \mu^{\xi_X}\left(\vec{\rho_i} \to \tau(\vec{B_i}) \to \xi_X\right)_{i<k}$$

$\tau(P\vec{t}) := \tau(P) \quad , \quad \tau(I\vec{t}) := \tau(I)$

$\tau(A \to B) := \tau(A) \to \tau(B) \quad , \quad \tau(A \to^{nc} B) := \tau(B)$

$\tau(\forall_x A) := \rho \to \tau(A) \quad , \quad \tau(\forall^{nc}_x A) := \tau(A)$

Notation $\iota_I := \tau(I)$

Kreisel's modified realizability

For a term (we say "realizer") and a computational formula A, we define the binary relation $\Vdash$ (bold $\vdash$).

We assume for all computational (c.r.)* predicate variables X of arity $(\vec{\rho})$ a global assignment giving a non-computational (n.c.) predicate variable $X^{\Vdash}$ of arity $(\tau(X), \vec{\rho})$.

* ... c.r. stands for computationally relevant.
formula C is n.c. if $fp(C) = X^{nc}$ or I^{nc}.
and c.r. otherwise.

Kreisel's modified realizability (Cont.)

$$z \Vdash P\vec{t} := P^{\Vdash} z\vec{t}$$

$$z \Vdash A \to B := \forall_w (w \Vdash A \to zw \Vdash B) \quad \text{if } A \text{ is c.r.}$$

$$z \Vdash A \to^{nc} B := \forall_w (w \Vdash A \to z \Vdash B) \quad \text{if } A \text{ is c.r.}$$

$$z \Vdash A \to^{(nc)} B := A \to z \Vdash B \quad \text{if } A \text{ is n-c.}$$

$$z \Vdash \forall_x A := \forall_x (zx \Vdash A)$$

$$z \Vdash \forall_x^{nc} A := \forall_x (z \Vdash A)$$

$$z \Vdash I\vec{t} := I^{\Vdash} z\vec{t}$$

Invariance axiom

We have the following invariance axiom under realizability.

Axiom

$$A \leftrightarrow \exists_z (z \Vdash A) \quad \text{for c.r. formula } A,$$

– we take the following 1 and 2 equal.

1. A holds.
2. there is a realizer (computational solution) of A.

Witnessing predicate $I^\sharp$

For given $I = \mu_X \left(\forall^{(nc)}_{\vec{x_i}} \left(\left(A_{iv}(X) \right)_{v<n_i} \to^{(nc)} X\, \vec{t_i} \right) \right)_{i<k}$

of arity $(\vec{\rho})$, we define $I^\sharp$ of arity $(\tau(I), \vec{\rho})$ to be

$$\mu^{nc}_{X^\sharp} \left(\forall_{\vec{x_i}} \forall_{\vec{z}_i} \left(\left(A'_{iv}(z_{iv}) \right)_{v<n_i} \to X^\sharp(C_i\, \vec{x_i}\, \vec{z}_i, \vec{t_i}) \right) \right)$$

where

1. C_i in the conclusion takes x_{ij} quantified by $\forall^c$ in I.
2. (i) For c.r. $A_{iv}(X)$ followed by $\to$, $A'_{iv}(z_{iv}) := z_{iv} \sharp A_{iv}(X)$ and C_i takes z_{iv}.
   (ii) For c.r. $A_{iv}(X)$ followed by $\to^{nc}$, $A'_{iv}(z_{iv}) := z_{iv} \sharp A_{iv}(X)$ and C_i takes z_{iv}
   (ii) For n.c. $A_{iv}(X)$, we have no z_{iv}.

Examples

— $\mathrm{Ex}\,D^+$. $\quad \tau(\mathrm{Ex}\,D) = \mu_\xi(\alpha \to \beta \to \xi)$ where $\mathrm{Ex}\,D := \mu_X(\forall_{x^\rho}(Yx \to X))$

$$\mathrm{Ex}\,P^+ := \mu^{nc}_{X^+}\left(\forall_x \forall_z (Y^+ zx \to X^+(\langle x,z\rangle))\right)$$

$$(\mathrm{Ex}\,D^+)^+_0 \;:\; \forall_{x,z}\left(Y^+_{zx} \to \mathrm{Ex}\,D^+_{Y^+}(\langle x,z\rangle)\right)$$

$$(\mathrm{Ex}\,D^+)^- \;:\; \mathrm{Ex}\,D^+_{Y^+}(\langle x,z\rangle) \to \forall_{x,z}\left(Y^+_{zx} \to P(\langle x,z\rangle) \to P(\langle x,z\rangle)\right)$$

— List^+ . $\quad \mathrm{List} := \mu_X\left(X(\mathrm{nil}),\ \forall^{nc}_{x^\rho, xs^{\mathrm{lin}\rho}}\left(Yx \to Xxs \to X(x:xs)\right)\right)$

then $\quad L_{\mathrm{List}} = \tau(\mathrm{List}) = \mu_\xi(\xi,\ \alpha \to \xi \to \xi) = \mathbb{L}_\alpha$

$$(\mathrm{List}^+)^+_0 \;:\; \mathrm{List}^+_{Y^+}(\mathrm{nil},\mathrm{nil})$$

$$(\mathrm{List}^+)^+_1 \;:\; \forall_{x,xs,u,us}\left(Y^+ ux \to \mathrm{List}^+_{Y^+}(us,xs) \to \mathrm{List}^+_{Y^+}(u:us, x:xs)\right)$$

$$(\mathrm{List}^+)^- \;:\; \mathrm{List}^+(us,xs) \to Q(\mathrm{nil},\mathrm{nil}) \to \forall_{x,xs,u,us}\left(Y^+ us \to \mathrm{List}^+(us,xs) \to Q(us,xs) \to Q(u:us, x:xs)\right) \to Q(us,xs)$$

Program Extraction

A proof (derivation) M is translated into a term t by et.

$et(M^A) := \varepsilon$ (nullterm) if A is n.c.

$et(u^A) := x_u^{\tau(A)}$

$$et((\lambda u^A M^B)^{A\to B}) := \begin{cases} \lambda x_u^{\tau(A)}\, et(M) & \text{if } A \text{ is c.r.} \\ et(M) & \text{if } A \text{ is n.c.} \end{cases}$$

$$et((M^{A\to B} N^A)^B) := \begin{cases} et(M)\, et(N) & \text{if } A \text{ is c.r.} \\ et(M) & \text{if } A \text{ is n.c.} \end{cases}$$

$$et((\lambda u^A M^B)^{A\to^{nc} B}) := et(M)$$

$$et((M^{A\to^{nc} B} N^A)^B) := et(M)$$

Program Extraction (cont.)

$$et\left(\left(\lambda x^{\rho} M^{A}\right)^{\forall_x \rho A}\right) := \lambda x^{\rho}\, et(M)$$

$$et\left(\left(M^{\forall_x \rho A(x)}\, t^{\rho}\right)^{A(t)}\right) := et(M)\, t$$

$$et\left(\left(\lambda x\, M^{A}\right)^{\forall_x^{nc} A}\right) := et(M)$$

$$et\left(\left(M^{\forall_x^{nc} A(x)}\, t\right)^{A(t)}\right) := et(M)$$

$et(I_i^+) := C_i$ where C_i is the i-th constr. of C_I

$et(I^-) := R^{\tau}_{C_I}$ where τ is the type of competitor pred.