Introduzione alla Realizzabilità classica di Krivine

Davide Barbarossa

Université Paris13 / Università di Roma Tre

barbarossa@lipn.univ-paris13.fr
https://lipn.univ-paris13.fr/ barbarossa/

XXI Congresso Unione Matematica Italiana

Corrispondenza di Curry-Howard

Dimostrazioni: Logica intuizionista minimale
Trasformazioni: $\pi \rightsquigarrow \pi'$ secondo la cut-elimination

Programmi: $\Lambda ::= \ x \mid \lambda x M \mid MN$
Esecuzione: $(\lambda x M)N \to_\beta M[N/x]$

Tipaggio

$$\frac{}{\Gamma, x : A \vdash x : A}$$

$$\frac{\Gamma \vdash M : A \to B \qquad \Gamma \vdash N : A}{\Gamma \vdash MN : B} \qquad\qquad \frac{\Gamma, x : A \vdash M : B}{\Gamma \vdash \lambda x M : A \to B}$$

Cut-elimination = Esecuzione programma

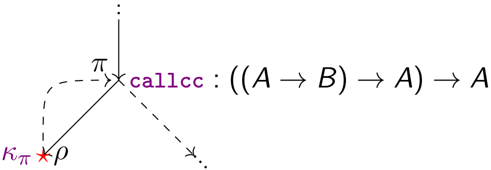
Se π dimostra $\Gamma \vdash M : A$ e π' dimostra $\Gamma \vdash M' : A$, allora:

$$\pi \rightsquigarrow \pi' \Leftrightarrow M \to_\beta M'.$$

Computazione più realistica: *interazione con l'ambiente*

$$\textit{Processo} := (\text{programma},\ \text{pila})$$

(Esempio di) Esecuzione nella *Krivine-Abstract-Machine*: *backtracking*



Esempio in logica *classica*: la "drunk man formula"

In ogni bar non vuoto, $\exists x_0$ t.c. se x_0 è ubriaco allora sono tutti ubriachi.

Dimostrazione.

Sia x nel bar. Wlog x è ubriaco e $\exists x'$ s.t. x' non è ubriaco (altrimenti prendi $x_0 := x$).
Prendi $x_0 := x'$. □

"Corrispondenza" dimostrazioni classiche e programmi

λ_c-calcolo

(Programmi) $\Lambda_c ::= \ x \mid \texttt{callcc} \mid \kappa_\pi \mid \lambda x M \mid MN$ (Pile) $\Pi_c ::= M.\pi$

(Processi) $\Lambda \star \Pi := \Lambda_c \times \Pi_c$

Tipaggio

L'usuale (+ quantificatori) + la regola:

$$\frac{}{\Gamma \vdash \texttt{callcc} : ((A \to B) \to A) \to A}\,(\text{Griffin})$$

KAM per il λ_c-calcolo

(push) $MN \star \pi \succ M \star N.\pi$

(pop) $\lambda x M \star N.\pi \succ M[N/x] \star \pi$

(save) $\texttt{callcc} \star M.\pi \succ M \star \kappa_\pi.\pi$

(restore) $\kappa_\pi \star M.\rho \succ M \star \pi.$

Interpretazioni di realizzabilità

"Realizzare" una formula A = trovare un programma M che giustifica A

Semantica di realizzabilità

Formule a parametri:

- l'interpretazione standard $a \in \mathbb{N}$ per ogni espressione al I ordine
- un $\mathcal{H} \in \mathscr{P}(\Pi_c)^{\mathbb{N}^k}$ per ogni variabile X al II ordine

Si fissi un segmento iniziale $⫫$ di $(\Lambda_c \star \Pi_c, \succ)$.
Si definiscono $\|.\| : \mathcal{F}_{par} \to \mathscr{P}(\Pi_c)$ e $|.| : \mathcal{F}_{par} \to \mathscr{P}(\Lambda_c)$ come segue:

$$\|\mathcal{H}[a]\| := \mathcal{H}(a) \qquad \|\forall x A\| := \bigcup_{a\in\mathbb{N}} \|A[a/x]\|$$

$$\|A \to B\| := |A|.\|B\| \qquad \|\forall X A\| := \bigcup_{\mathcal{H}\in\mathscr{P}(\Pi_c)^{\mathbb{N}^k}} \|A[\mathcal{H}/X]\|$$

$$|A| := \{M \in \Lambda_c \text{ t.c. } \forall\pi \in \|A\|,\ M \star \pi \in ⫫\}$$

Si pone $M \Vdash A$ come abbreviazione di $M \in |A|$.

Fatti importanti

Teorema (Lemma di adequazione)

$$\vdash M : A \;\Rightarrow\; M \Vdash A$$

Programmi che "barano"

Per ogni $M \star \pi \in \perp\!\!\!\perp$, si ha: $\kappa_\pi M \Vdash \perp$. (con $\perp := \forall X X$)

La realizzabilità induce modelli à la Tarski

La teoria $\{A \text{ formula t.c. } \exists M \in \mathrm{QP} \subseteq \Lambda \text{ t.c. } M \Vdash A\}$ è coerente.
I suoi modelli sono chiamati *modelli di realizzabilità*.

Il modello standard come modello di realizzabilità degenere

Se $\perp\!\!\!\perp = \varnothing$ allora $\exists!$ modello di realizzabilità, ed è $\mathbb{N}$.

Questioni generali

Realizzare assiomi

Data una formula A, esiste un programma $M \in \mathrm{QP}$ t.c. $M \Vdash A$?

Esempio se $\vdash M : A$; allora $M \in \mathrm{QP}$ e $M \Vdash A$.

Esempio se $A = \mathrm{l'assioma\ di\ scelta}$; allora...??!

Problema della specificazione

Dato $M \Vdash A$, *in che modo* M giustifica A ?

Esempio (specificazione per l'identità polimorfa)

se $\vdash M : \forall X(X \to X)$ allora $M \approx \lambda xx$
(nel senso che $M \star N.\pi \succ N \star \pi \prec \lambda xx \star N.\pi$)

Esempio (Axiom of Countable Choice)

Esiste $M \in \mathrm{QP}$ t.c. $M \Vdash \mathrm{ACC}$
"the main function of M is to update files".

Generalizzare!

Algebre di realizzabilità

- tre insiemi Λ (programmi), Π (pile), $\Lambda \star \Pi$ (processi)
- tre applicazioni $\mathrm{cons} : \Lambda \times \Pi \to \Pi$, $\mathrm{cont} : \Pi \to \Lambda$, $\mathrm{proc} : \Lambda \times \Pi \to \Lambda \star \Pi$
- "un modo per vedere le quasi-prove come programmi: $\mathrm{QP} \subseteq \Lambda_c \to \Lambda$"
- un *polo* $\perp\!\!\!\perp \subseteq \Lambda \star \Pi$ saturo rispetto ad una KAM

Stesse identiche costruzioni, risultati, osservazioni,...

Estrazione di contenuto computazionale dalla matematica

- Realizzare assiomi di ZF
- Costruzione di nuovi modelli *non* ottenibili per forcing
- Forcing = trasformazione programmi + esecuzione "protected mode"
- Ecc...

Grazie!

Jean-Louis Krivine:
- Typed lambda-calculus in classical Zermelo-Fraenkel set theory. Archives of Mathematical Logic (2001)
- Realizability in classical logic. Lessons for PhD's at Université d'Aix-Marseille (2004)
- Realizability algebras: a program to well order $\mathbb{R}$. Logical Methods in Computer Science (2011)
- À propos de la théorie des démonstrations. Colloque en l'honneur de René Cori (2014).
- À propos de l'intuition en mathématiques. (2018).

Lionel Rieg:
- On Forcing and Classical Realizability. PhD thesis in Computer Science at École Normale Supérieure de Lyon (2014)

Alexandre Miquel:
- Forcing as a program transformation. LICS (2011) (short version)/MSCS (2012) (long version).