Constructive completeness proofs and delimited control
PhD thesis defence

Danko Ilik

École Polytechnique – INRIA – Université Paris Diderot

thesis directed by Hugo Herbelin

Paris, October 22, 2010
Constructive Mathematics and Computer Science

- The Curry-Howard correspondence:
  - proofs are programs;
  - theorems are specifications.
- Constructive type theory and the Coq proof assistant
- Coq is a tool for developing formal proofs:
  - of theorems in Constructive Mathematics;
  - of correctness of programs with respect to a specification.
Completeness theorems as programs

- A formalised Completeness theorem – a tool to switch between model theoretic and proof theory arguments inside Coq
- Connections between Completeness and Normalisation-by-Evaluation
Outline

Boolean completeness

Classical NBE

Intuitionistic NBE

Delimited control in Logic
Outline

Boolean completeness

Classical NBE

Intuitionistic NBE

Delimited control in Logic
Completeness for standard semantics

Theorem (Gödel 1930)

*A is valid if and only if A is derivable*

*A - derivable* there is a derivation tree for A in classical 1st-order logic

*A - valid* Tarski’s truth definition:

\[\begin{align*}
\mathcal{M} \models A \land B & := \mathcal{M} \models A \text{ and } \mathcal{M} \models B \\
\mathcal{M} \models A \lor B & := \mathcal{M} \models A \text{ or } \mathcal{M} \models B \\
\mathcal{M} \models A \rightarrow B & := \mathcal{M} \models A \text{ implies } \mathcal{M} \models B \\
\mathcal{M} \models \exists x A(x) & := \text{exists } t \text{ with } \mathcal{M} \models A(t) \\
\mathcal{M} \models \forall x A(x) & := \text{for any } t, \mathcal{M} \models A(t) \\
\mathcal{M} \models \bot & := \text{false}
\end{align*}\]
Is it constructive?

**Theorem (McCarty 1996)**

*No? – Completeness implies Markov's Principle (MP)*
Is it constructive?

**Theorem (McCarty 1996)**

*No?* – *Completeness implies Markov's Principle (MP)*

**Theorem (Krivine 1996)**

*Yes?* – *Gödel's proof is constructive, if we allow one more model – the model that validates ⊥*
What is the algorithm behind Krivine’s proof?

- Krivine’s proof carried out in *classical* 2nd-order arithmetic
- From the form of the statement, he concludes there is a proof in intuitionistic 2nd-order arithmetic
- Formalisation in Phox (PA₂) by Raffalli; algorithm extracted but “unreadable”
- Proof unwound in (Berardi-Valentini 2004): main ingredient a constructive ultra-filter theorem
Constructive Ultra-filter Theorem

\( \mathcal{B} \) countable Boolean algebra

Filter subset of \( \mathcal{B} \) which is inhabited, \( \leq \)-closed and \( \land \)-closed

\( b \in \uparrow X = \exists a_1, \ldots, a_n \in X. \ a_1 \land \cdots \land a_n \leq b \)

\( X \)-complete \( (\neg c \in X \rightarrow \bot \in X) \rightarrow c \in X, \text{ for all } c \in \mathcal{B} \)
Constructive Ultra-filter Theorem

\( \mathcal{B} \) countable Boolean algebra

Filter subset of \( \mathcal{B} \) which is inhabited, \( \leq \)-closed and \( \land \)-closed

\( b \in \uparrow X = \exists a_1, \ldots, a_n \in X. a_1 \land \cdots \land a_n \leq b \)

\( X \)-complete \( (\neg c \in X \rightarrow \bot \in X) \rightarrow c \in X \), for all \( c \in \mathcal{B} \)

Theorem (Berardi-Valentini 2004)

Every filter \( F \) can be extended to a complete filter \( Z(F) \), so that

\( F \sim Z(F) \) (\( \bot \in F \iff \bot \in Z(F) \))

Proof.

\[
\begin{align*}
F_0 & := F \\
F_{n+1} & := \uparrow (F_n \cup \{ b \mid \neg b \downarrow = n, F_n \sim \uparrow (F_n \cup \{b\}) \}) \\
Z & := \bigcup_{n \in \mathbb{N}} F_n
\end{align*}
\]
From Ultra-filter theorem to Completeness

Instantiate $\mathcal{B}$ with the Lindenbaum Boolean algebra:

$$a \leq b := a \vdash b$$

$$a \land b := \vdash \neg(a \Rightarrow \neg b)$$

If $X$ is a set of axioms, then

$$a \in Z(\uparrow X)$$

means

$$\exists n. \exists \Gamma \subseteq F_n(\uparrow X). \Gamma \vdash a,$$

which implies,

$$\exists \Gamma \subseteq X. \Gamma \vdash a.$$
Computational content

Reflection:

\[(a \Rightarrow b) \in Z \rightarrow a \in Z \rightarrow b \in Z\]

\[m \mapsto n \mapsto \max(m, n)\]

Reification:

\[(a \in Z \rightarrow b \in Z) \rightarrow (a \Rightarrow b) \in Z\]

let \(c \coloneqq (a \Rightarrow b)\) in \(Z\)-complete

\(Z\)-complete is a kind of meta-level \(\neg\neg_E\):

\[((c \in Z \rightarrow \bot \in Z) \rightarrow \bot \in Z) \rightarrow c \in Z\]
Conclusion

Contribution:

- detailed Henkin-style argument formalised in Type Theory;
- generalisation to setoids of the Ultra-filter Theorem.

Future work:

- develop a proof/algorithm not parametrised by an enumeration (using delimited control);
- finish the Coq formalisation.
Outline

Boolean completeness

Classical NBE

Intuitionistic NBE

Delimited control in Logic
Get a completeness theorem for computational classical calculi – reduction relation should be preserved.

Follow Normalization-by-Evaluation (NBE) methodology (Berger-Schwichtenberg 1991):

**Theorem (Soundness/Evaluation)**
\[ \Gamma \vdash A \rightarrow \forall w, w \models \Gamma \rightarrow w \models A \]

**Theorem (Completeness/Reification)**
\[ (\forall w, w \models \Gamma \rightarrow w \models A) \rightarrow \Gamma \vdash_{nf} A \]

**Corollary (NBE)**

*The composition (Completeness \circ Soundness) normalizes proof terms into η-long β-normal form.*
Standard Kripke models

Start with a structure \( (K, \leq, D, \models, \models_{\bot}) \), and extend \( \models \) to non-atomic formulas:

- \( w \models A \land B \iff w \models A \) and \( w \models B \)
- \( w \models A \lor B \iff w \models A \) or \( w \models B \)
- \( w \models A \rightarrow B \) for any \( w' \geq w \), if \( w' \models A \) then \( w' \models B \)
- \( w \models \forall x P(x) \) for any \( w' \geq w \) and any \( a \in D(w') \), \( w' \models P(a) \)
- \( w \models \exists x P(x) \) there is \( a \in D(w) \) such that \( w \models P(a) \)
Kripke-style models (Call-by-value variant)

Like with Kripke models, start with a structure \((K, \leq, D, \models_s, \models_\bot)\), and extend strong forcing \((\models_s)\) to non-atomic formulas:

- **\(A \land B\)** \(w \models A\) and \(w \models B\)
- **\(A \lor B\)** \(w \models A\) or \(w \models B\)
- **\(A \rightarrow B\)** for any \(w' \geq w\), if \(w' \models A\) then \(w' \models B\)
- **\(\forall x P(x)\)** for any \(w' \geq w\) and any \(a \in D(w')\), \(w' \models P(a)\)
- **\(\exists x P(x)\)** there is \(a \in D(w)\) such that \(w \models P(a)\)

where the non-s-annotated \(\models\) is (non-strong) forcing:

\[
\forall w_1 \geq w. (\forall w_2 \geq w_1. w_2 \models_s A \rightarrow w_2 \models_\bot) \rightarrow w_1 \models_\bot
\]

"refutation" \(w_1 : A \models\)
Completeness for Kripke-style models and LK\(\mu\tilde{\mu}\)

**Theorem (Soundness)**

\[c : (\Gamma \vdash \Delta) \implies \text{for any } w, w \vDash \Gamma \text{ and } w : \Delta \vdash \text{ implies } w \vDash \bot\]
\[\Gamma \vdash t : A|\Delta \implies \text{for any } w, w \vDash \Gamma \text{ and } w : \Delta \vdash \text{ implies } w \vDash A\]
\[\Gamma | e : A \vdash \Delta \implies \text{for any } w, w \vDash \Gamma \text{ and } w : \Delta \vdash \text{ implies } w : A \vDash\]

**Theorem (Completeness)**

\[(\Gamma, \Delta) \vdash A \implies \text{there is a term } t \text{ such that } \Gamma \vdash_{cf} t : A|\Delta\]
\[(\Gamma, \Delta) : A \vDash \implies \text{there is an ev. context } e \text{ such that } \Gamma | e : A \vdash_{cf} \Delta\]

**Proof.**

Make a Universal model \(\mathcal{U}\) from the derivation system:

- worlds are pairs \((\Gamma, \Delta)\)
- strong forcing is cut-free derivability of atoms:
  \[(\Gamma, \Delta) \vdash_s X := \exists t. \Gamma \vdash_{cf} t : X|\Delta\]
- exploding nodes are cuts: \((\Gamma, \Delta) \vdash_{\perp} := \exists c. c : (\Gamma \vdash_{cf} \Delta)\)
Conclusion

- New notion of model for classical logic
- Not as simple as Boolean models
- But, reduction is preserved
- Dual notion of model that gives call-by-name normalization strategy
- Proofs formalised in Coq
Introduction

Boolean completeness

Classical NBE

Intuitionistic NBE

Delimited control in Logic

Completeness of Intuitionistic Logic for Kripke models

- Kripke models are a standard semantics for intuitionistic logic
- But, there is no (simple) constructive proof with $\lor, \exists$:
  - classical Henkin-style proofs (Kripke 1965)
  - using Fan Theorem (Veldman 1976)
  - a constructive proof would imply MP (Kreisel 1962)
- On the other hand, a well-typed functional program for NBE of $\lambda \to \lor$ (Danvy 1996)
  - using delimited-control operators $\textit{shift}$ and $\textit{reset}$ (Danvy-Filinski 1989)
Completeness/NBE for $\lambda \rightarrow^\vee$

What the problem is

Theorem (NBE)

$\downarrow^A_\Gamma ("reify") : \Gamma \vdash A \rightarrow \Gamma \vdash_{nf} A$

$\uparrow^A_\Gamma ("reflect") : \Gamma \vdash_{ne} A \rightarrow \Gamma \vdash A$

Proof of case $\uparrow^{A \lor B}$.

Given a derivation $\Gamma \vdash_{ne} A \lor B$, decide: $\Gamma \vdash A$ or $\Gamma \vdash B$?
**Shift (\(\mathcal{S}\)) and reset (\(#\)) delimited control operators**

**Examples**

\[ #V \rightarrow V \]

\[ #F[\mathcal{S}k.p] \rightarrow #p\{k := \lambda x.#F[x]\} \]
Shift (\(\mathcal{S}\)) and reset (\(#\)) delimited control operators

Examples

\[\begin{align*}
#V & \rightarrow V \\
#F[\mathcal{S} k. p] & \rightarrow #p\{k := \lambda x. #F[x]\}
\end{align*}\]

\[\begin{align*}
1 + # (2 + \mathcal{S} k. k(k4)) & \\
\rightarrow 1 + # ((\lambda a. #(2 + a)) ((\lambda a. #(2 + a))4)) & \\
\rightarrow^+ 1 + #(#(#8)) & \\
\rightarrow^+ 9 &
\end{align*}\]
Completeness/NBE for $\lambda \to ^{\vee}$
Solution of Danvy: use delimited control operators shift ($\mathcal{S}$) and reset (#)

**Theorem (NBE)**

$\downarrow^{A}_{\Gamma} ("reify") : \quad \Gamma \vdash A \longrightarrow \Gamma \vdash^{nf} A$

$\uparrow^{A}_{\Gamma} ("reflect") : \quad \Gamma \vdash^{ne} A \longrightarrow \Gamma \vdash A$

**Proof of case $\uparrow^{A \vee B}$**.

Given a derivation $e$ of $\Gamma \vdash^{ne} A \vee B$, decide: $\Gamma \vdash A$ or $\Gamma \vdash B$, by

$\mathcal{S} k. \ \vee_{E} e (x \mapsto \#k(\text{left } \uparrow^{A}_{x:A,\Gamma} x)) (y \mapsto \#k(\text{right } \uparrow^{B}_{y:B,\Gamma} y))$
Completeness/NBE for $\lambda \rightarrow^\vee$

Solution of Danvy: Issues

- We are convinced the **program** computes correctly
- There should be a corresponding completeness **proof** for Kripke model
- Type-and-effect system: types $A \rightarrow B$ become $A/\alpha \rightarrow B/\beta$, what is the logical meaning?
- Typing via classical logic
Completeness for Intuitionistic Predicate Logic (IQC)

Extracting a notion of model from Danvy's solution

Like with Kripke models, start with a structure \((K, \leq, D, \models_s, \models^{(\cdot)}_s, \bot)\), and extend \textbf{strong forcing} \((\models_s)\) to non-atomic formulas:

- **\(w \models_s A \)** and \(w \models_s B\)
- **\(A \land B\)**
- **\(A \lor B\)**
- **\(A \rightarrow B\)** for any \(w' \geq w\), if \(w' \models A\) then \(w' \models B\)
- **\(\forall x P(x)\)** for any \(w' \geq w\) and any \(a \in D(w')\), \(w' \models P(a)\)
- **\(\exists x P(x)\)** there is \(a \in D(w)\) such that \(w \models P(a)\)

where the non-s-annotated \(\models\) is \textbf{(non-strong) forcing}:

\[
\models A := \forall C. \forall w_1 \geq w. (\forall w_2 \geq w_1. w_2 \models_s A \rightarrow w_2 \models^C \bot) \rightarrow w_1 \models^C \bot
\]
Completeness for IQC via Kripke-style models

Theorem (NBE)
\[ \downarrow^A \Gamma \text{ ("reify") : } \Gamma \vdash A \longrightarrow \Gamma \vdash^nf A \]
\[ \uparrow^A \Gamma \text{ ("reflect") : } \Gamma \vdash ne A \longrightarrow \Gamma \vdash A \]

Proof of case \( \uparrow^{A \lor B} \).

Given a derivation \( e \) of \( \Gamma \vdash ne A \lor B \), prove \( \Gamma \vdash A \lor B \) i.e.

\[ \forall C. \forall \Gamma_1 \geq \Gamma. (\forall \Gamma_2 \geq \Gamma_1. \Gamma_2 \vdash s A \text{ or } \Gamma_2 \vdash s B \rightarrow \Gamma_2 \vdash \bot) \rightarrow \Gamma_1 \vdash \bot \]

by
\[ k \mapsto \lor_E e (x \mapsto k(\text{left } \uparrow^A x : A, \Gamma x)) \ (y \mapsto k(\text{right } \uparrow^B y : B, \Gamma y)) \]
Conclusion

- Contribution:
  - New notion of model for Intuitionistic logic
  - $\beta$-Normalises $\lambda$-calculus with sum
  - But, not as simple as Kripke models
  - Formalised in Coq

- Future work:
  - Find a good logical system for delimited control that can prove completeness for standard Kripke models
# Outline

<table>
<thead>
<tr>
<th>Introduction</th>
<th>Boolean completeness</th>
<th>Classical NBE</th>
<th>Intuitionistic NBE</th>
<th>Delimited control in Logic</th>
</tr>
</thead>
</table>

- **Boolean completeness**
- **Classical NBE**
- **Intuitionistic NBE**
- **Delimited control in Logic**
Delimited control operators in Logic

- Should allow us to give a constructive proof of completeness for Kripke semantics (Danvy’s NBE functional program)
- Herbelin: delimited control allows to derive Markov’s Principle (Herbelin 2010) and the Double Negation Shift
- Allow to simulate any monadic computational effect (Filinski 1994)
Proof term $\lambda$-calculus with $\mathcal{S}$ and $\#$

Proof terms:

\[
p, q, r ::= a | i_1 p | i_2 p | \text{case } p \text{ of } (a.q||b.r) | (p, q) | \pi_1 p | \pi_2 p | \lambda a.p |
\]
\[
| pq | \lambda x.p | pt | (t, p) | \text{dest } p \text{ as } (x.a) \text{ in } q | \#p | \mathcal{S} k.p
\]
Proof term $\lambda$-calculus with $\mathcal{S}$ and $\#$

Proof terms:

\[ p, q, r ::= a \mid \iota_1 p \mid \iota_2 p \mid \text{case } p \text{ of } \left( a.q \parallel b.r \right) \mid (p, q) \mid \pi_1 p \mid \pi_2 p \mid \lambda a.p \mid \]
\[ \mid pq \mid \lambda x.p \mid pt \mid (t, p) \mid \text{dest } p \text{ as } (x.a) \text{ in } q \mid \#p \mid \mathcal{S} k.p \]

Values:

\[ V ::= a \mid \iota_1 V \mid \iota_2 V \mid (V, V) \mid (t, V) \mid \lambda a.p \mid \lambda x.p \]
Proof term $\lambda$-calculus with $\mathcal{S}$ and $\#$

Proof terms:

$$p, q, r ::= a | \iota_1 p | \iota_2 p | \text{case } p \text{ of } (a.p||b.r) | (p, q) | \pi_1 p | \pi_2 p | \lambda a.p | p.q | \lambda x.p | pt | (t, p) | \text{dest } p \text{ as } (x.a) \text{ in } q | \#p | \mathcal{S} k.p$$

Values:

$$V ::= a | \iota_1 V | \iota_2 V | (V, V) | (t, V) | \lambda a.p | \lambda x.p$$

Pure evaluation contexts:

$$P ::= [] | \text{case } P \text{ of } (a_1.p_1||a_2.p_2) | \pi_1 P | \pi_2 P | \text{dest } P \text{ as } (x.a) \text{ in } p | Pq | (\lambda a.q)P | Pt | \iota_1 P | \iota_2 P | (P, p) | (V, P) | (t, P)$$
Proof term $\lambda$-calculus with $S$ and $#$

Proof terms:

$$p, q, r ::= a \mid \iota_1 p \mid \iota_2 p \mid \text{case } p \text{ of } (a.q \parallel b.r) \mid (p, q) \mid \pi_1 p \mid \pi_2 p \mid \lambda a.p \mid\allowbreak pq \mid \lambda x.p \mid pt \mid (t, p) \mid \text{dest } p \text{ as } (x.a) \text{ in } q \mid #p \mid S k.p$$

Values:

$$V ::= a \mid \iota_1 V \mid \iota_2 V \mid (V, V) \mid (t, V) \mid \lambda a.p \mid \lambda x.p$$

Pure evaluation contexts:

$$P ::= [ ] \mid \text{case } P \text{ of } (a_1.p_1 \parallel a_2.p_2) \mid \pi_1 P \mid \pi_2 P \mid \text{dest } P \text{ as } (x.a) \text{ in } p \mid Pq \mid (\lambda a.q)P \mid Pt \mid \iota_1 P \mid \iota_2 P \mid (P, p) \mid (V, P) \mid (t, P)$$

Reduction: (Call-by-value strategy)

$$(\lambda a.p) V \rightarrow p\{V/a\} \quad \text{case } \iota_i V \text{ of } (a_1.p_1 \parallel a_2.p_2) \rightarrow p_i\{V/a_i\}$$

$$(\lambda x.p) t \rightarrow p\{t/x\} \quad \text{dest } (t, V) \text{ as } (x.a) \text{ in } p \rightarrow p\{t/x\}\{V/a\}$$

$$\pi_i(V_1, V_2) \rightarrow V_i$$

$$\#P[S k.p] \rightarrow \#p\{\lambda a.\#P[a] \}/k$$

$$\#V \rightarrow V$$

$$E[p] \rightarrow E[p'] \text{ when } p \rightarrow p'$$
Typing/Logical system MQC$^+$

The usual rules of MQC (minimal predicate logic), potentially annotated,

\[ \cdots \vdash_T \cdots \]

plus rules for reset and shift:

\[ \Gamma \vdash_T p : T \quad \Rightarrow \quad \Gamma \vdash_{\#} #p : T \]

\[ \Gamma, k : A \Rightarrow T \vdash_T p : T \quad \Rightarrow \quad \Gamma \vdash_T \sharp k.p : A \]

\( T \) denotes a \{\Rightarrow, \forall\}-free formula ("\( \Sigma \)-formula")
Deriving MP and DNS

**Markov’s Principle** (predicate logic version):

\[ \neg\neg S \Rightarrow S, \quad \text{for } S \text{ a } \Sigma\text{-formula} \]

\[\lambda a.\#\bot_E (a (\lambda b. \mathcal{S} k.b))\]
Deriving MP and DNS

**Markov’s Principle** (predicate logic version):

\[ \neg\neg S \Rightarrow S, \quad \text{for } S \text{ a } \Sigma\text{-formula} \]

\[ \lambda a.\# \bot_E (a(\lambda b. \not S k.b)) \]

**Double Negation Shift** (predicate logic version):

\[ \forall x(\neg\neg A(x)) \Rightarrow \neg\neg(\forall xA(x)) \]

\[ \lambda a.\lambda b.\# b(\lambda x. \not S k.axk) \]
Equiconsistency of MQC$^+$ with MQC

By the call-by-value continuation-passing-style translation (related to Glivenko's double-negation translation)

$$A^T := (A_T \Rightarrow T) \Rightarrow T$$

$$A_T := A$$

$$(A \Box B)_T := A_T \Box B_T$$

$$(A \Rightarrow B)_T := A_T \Rightarrow B^T$$

$$(\exists A)_T := \exists A_T$$

$$(\forall A)_T := \forall A^T$$

if $A$ is a atomic

for $\Box = \lor, \land$
Theorem (Equiconsistency)

Given a derivation of $\Gamma \vdash^+ A$, which uses $\mathcal{S}$ and $\#$ for the $\Sigma$-formula $T$, we can build a derivation of $\Gamma_T \vdash^m A_T$.

Theorem (Glivenko’s Theorem extended to quantifiers)

\[ \vdash^+ \neg\neg A \leftrightarrow DNS\vdash^i A^\perp \leftrightarrow \vdash^c A \]
Properties of MQC$^+$

**Theorem (Subject Reduction)**

If $\Gamma \vdash^+ p : A$ and $p \rightarrow q$, then $\Gamma \vdash^+ q : A$.

**Theorem (Progress)**

If $\vdash^+ p : A$, $p$ is not a value, and $p$ is not of form $P[S\ k.p']$, then $p$ reduces in one step to some proof term $r$.

**Theorem (Normalisation)**

For every closed proof term $p_0$, such that $\vdash^+ p_0 : A$, there is a finite reduction path $p_0 \rightarrow p_1 \rightarrow \ldots \rightarrow p_n$ ending with a value $p_n$.

**Corollary (Disjunction and Existence Properties)**

If $\vdash^+ A \lor B$, then $\vdash^+ A$ or $\vdash^+ B$.

If $\vdash^+ \exists x A(x)$, then there exists a closed term $t$ such that $\vdash^+ A(t)$. 
Conclusion

- Contribution:
  - A typing system for delimited control which remains intuitionistic (DP and EP) while deriving MP, DNS
  - But, only one use of MP is allowed

- Future work:
  - Annotating a derivation by a context $\Delta$, like in (Herbelin 2010):
    \[
    \Gamma \vdash_{\alpha:T,\Delta}^+ p:T \\
    \Gamma \vdash_{\Delta}^+ \#_\alpha p:T
    \]
    \[
    \Gamma, k:A \Rightarrow T \vdash_{\alpha:T,\Delta}^+ p:T \\
    \Gamma \vdash_{\alpha:T,\Delta}^+ \mathcal{J}_\alpha k.p:A
    \]
  - Connection to Fan Theorem, Open Induction, and other principles of Intuitionistic Reverse Mathematics
  - A logical study of computational effects