A computational analysis of Gödel's completeness theorem

Hugo Herbelin, Danko Ilik

(work in progress)

TYPES '10

Warsaw

The completeness of classical first-order logic wrt boolean models

First proof by Gödel [1929]

- reasoning on the prenex form + induction on the number of alternation of quantifiers + contradiction by reasoning + weak König's lemma/fan theorem

Standard proof by Henkin [1949]

- reasoning by contradiction + construction of a counter-model by enumeration of the formulas over a language extended with Henkin constants coming from the skolemization of the drinker's paradox ($\exists x(P(x) \rightarrow \forall y\, P(y))$).

Alternative proofs (Kleene, ...)

- build a tableau + reasoning by contraction to show it has no infinite branch + weak König lemma/fan theorem

The need for Markov's principle

- Kreisel [1962], after Gödel [1957]: completeness implies Markov's principle

The completeness of classical first-order logic (continued)

Krivine's constructive proof [1996]

- restrict to minimal classical logic (no $\bot \to A$) so that negation disappears in the definition of the model and Friedman's A-translation [1978] is applicable to get rid of Markov's principle

- analyzed by Berardi and Valentini [2001]: Krivine adds one extra (degenerated) model, the always-true model (similar to Friedman's fallible models and Veldman's exploding nodes in intuitionistic logic semantics)

- Krivine's statement still classically equivalent to the original statement

- formalized by Christophe Raffalli in PhoX

- formalized by Danko Iliḱ in Coq

We actually do not need the degenerated model nor A-translation

- Markov's principle = an exception mechanism [LICS '2010]

The statement of completeness
(restricted to the negative fragment)

$$
\begin{aligned}
t \in \mathcal{T} &::= x \mid f t_1 \ldots t_{a_f} \\
A, B \in \mathcal{F} &::= P t_1 \ldots t_{a_P} \mid \dot{\bot} \mid A \dot{\rightarrow} B \mid \dot{\forall} x\, A
\end{aligned}
$$

A model is a triple $(\mathcal{M}_D,\ \mathcal{M}(f) \in \mathcal{M}_D^{a_f} \rightarrow \mathcal{M}_D,\ \mathcal{M}(P) \in \mathcal{P}(\mathcal{M}_D^{a_P}))$ such that $\neg\neg((t_1, \ldots, t_{a_P}) \in \mathcal{M}(P)) \rightarrow (t_1, \ldots, t_{a_P}) \in \mathcal{M}(P)$. Truth in $\mathcal{M}$ is defined recursively:

$$
\begin{array}{lcl}
[\![x]\!]^{\sigma}_{\mathcal{M}} & \triangleq & \sigma(x) \\
[\![f t_1 \ldots t_{a_f}]\!]^{\sigma}_{\mathcal{M}} & \triangleq & \mathcal{M}(f)[\![t_1]\!]^{\sigma}_{\mathcal{M}} \ldots [\![t_{a_f}]\!]^{\sigma}_{\mathcal{M}} \\
[\![P t_1 \ldots t_{a_f}]\!]^{\sigma}_{\mathcal{M}} & \triangleq & \mathcal{M}(P)[\![t_1]\!]^{\sigma}_{\mathcal{M}} \ldots [\![t_{a_P}]\!]^{\sigma}_{\mathcal{M}} \\
[\![\dot{\bot}]\!]^{\sigma}_{\mathcal{M}} & \triangleq & \bot \\
[\![A \dot{\rightarrow} B]\!]^{\sigma}_{\mathcal{M}} & \triangleq & [\![A]\!]^{\sigma}_{\mathcal{M}} \rightarrow [\![B]\!]^{\sigma}_{\mathcal{M}} \\
[\![\dot{\forall} x\, A]\!]^{\sigma}_{\mathcal{M}} & \triangleq & \forall t \in \mathcal{M}_D\, [\![A]\!]^{\sigma[x \leftarrow t]}_{\mathcal{M}}
\end{array}
$$

The completeness statement : $\forall A\, (\forall \mathcal{M} \forall \sigma [\![A]\!]^{\sigma}_{\mathcal{M}}) \rightarrow \vdash A$

Remarks on the formulation

We placed ourself in intuitionistic second-order arithmetic.

Some authors reason instead in the arithmetic of types of rank 2 and define $\mathcal{M}(P)$ as a boolean function in $\mathcal{M}_D^{a_f} \to \texttt{bool}$. The completeness proof then needs the axiom of unique choice (AC!) which means the metatheory is actually equivalent to second-order arithmetic. To avoid having to computationally interpret AC!, we prefer to directly reason in second-order arithmetic.

It is also common to replace $\mathcal{M}_P$ by a set $\mathcal{M}_F$ of formulas enriched over $\mathcal{D}$ such that:

$$
\begin{array}{lcl}
\dot{\bot} \in \mathcal{M}_F & \leftrightarrow & \bot \\
A \dot{\to} B \in \mathcal{M}_F & \leftrightarrow & A \in \mathcal{M}_F \to B \in \mathcal{M}_F \\
\dot{\forall} x\, A \in \mathcal{M}_F & \leftrightarrow & \forall t, A[t/x] \in \mathcal{M}_F \\
A \in \mathcal{M}_F & \leftrightarrow & \neg\neg A \in \mathcal{M}_F
\end{array}
$$

Our approach has both the advantage of avoiding to consider formulas enriched over $\mathcal{D}$ and to make the connection with intuitionistic models (e.g. Kripke) closer.

We are now going to present a simplified form of Henkin's proof

- using a `reify`/`reflect` approach, as done in the normalization-by-evaluation and reducibility-based normalization proofs

- using exceptions for dealing with the semantics of $\bot$

The proof

Let $\lceil A \rceil$ and ϕ form a Gödel's numbering of formulas such that $\lceil \phi(n) \rceil = n$. Let x_n be a variable fresh in $\phi(1), \ldots, \phi(n)$. Let A_0 the formula we want a proof of.

Let F_n be (informally) the countermodel built at step n. We write $A \in F_\omega$ for $\exists n \exists \Gamma \subset F_n \, \Gamma \vdash A$ ("A gets provable at some step of the construction of a model equiconsistent to $\neg A_0$") where $\Gamma \subset F_n$ is formally defined inductively:

$$\frac{}{(a_0 : \neg A_0) \subset F_0} \, I_0 \qquad\qquad \frac{\Gamma \subset F_n}{\Gamma \subset F_{n+1}} \, I_S$$

$$\frac{\Gamma \subset F_n \qquad \forall \Gamma' \subset F_n, \;\; \Gamma'(a : \{A\}_n) \vdash \dot{\bot} \;\rightarrow\; \Gamma' \vdash \dot{\bot}}{\Gamma(a : \{A\}_n) \subset F_{n+1}} \, I_n$$

where $\{A\}_n$ is $A[x_n/x] \wedge (A[x_n/x] \rightarrow \forall x \, A)$ if $\phi(n+1) = \dot{\forall} x \, A$ and A otherwise.

The (syntactic) model $\mathcal{M}_0$ is defined by $\mathcal{D} \triangleq \mathcal{T}$, $\mathcal{M}(f)(t_1, \ldots, t_{a_f}) \triangleq f(t_1, \ldots, t_{a_f})$, $\mathcal{M}(P)(t_1, \ldots, t_{a_P}) \triangleq \neg\neg P(t_1, \ldots, t_{a_P}) \in F_\omega$ and the side-condition on $\mathcal{M}_P$ is proved by $m \mapsto m' \mapsto m(k \mapsto km')$.

(hereafter, $\dot{\mathtt{ax}}$, $\dot{\mathtt{cut}}$, $\dot{\mathtt{efq}}$, $\dot{\mathtt{cc}}$, $\pi_1^{\dot{\rightarrow}}$, $\pi_2^{\dot{\rightarrow}}$, $mk^{\dot{\forall}}$, $\dot{\pi}_1$, $\dot{\mathtt{app}}^{\rightarrow}$ and $\dot{\mathtt{app}}^{\forall}$ are constructions of the object language)

The core of the proof

$$\downarrow_A \;:\; A \in \mathcal{M} \;\longrightarrow\; A \in F_\omega$$

$$\downarrow_{P(\vec{t})}\; m \;\triangleq\; (n, (a_0 : \neg A_0)(a : P(\vec{t})),$$
$$I_n(\mathtt{inj}_n, (\Gamma, f, p) \mapsto \mathtt{catch}_\alpha(m\,((n', \Gamma', f', p') \mapsto \mathtt{throw}_\alpha \mathtt{flush}^{\Gamma\cup\Gamma'}_{max(n,n')}(\mathtt{join}^{\Gamma\Gamma'}_{nn'}(f, f'), \dot{\mathtt{cut}}(p$$
$$\dot{\mathtt{ax}}(a)) \qquad \text{where } n = \lceil P(\vec{t}) \rceil$$

$$\downarrow_\bot\; m \;\triangleq\; \dot{\mathtt{efq}}\, m$$

$$\downarrow_{A\dot{\to}B}\; m \;\triangleq\; (n, (a_0 : \neg A_0)(a : A\dot{\to}B),$$
$$I_n(\mathtt{inj}_n, (\Gamma, f, p) \mapsto \begin{array}{l}\mathtt{dest}\ \downarrow_B (m(\uparrow_A (n, \Gamma, f, \pi_1^{\dot{\to}}\, p)))\ \mathtt{as}\ (n', \Gamma', f', p') \\ \mathtt{in}\ (max(n, n'), \Gamma \cup \Gamma', \mathtt{join}^{\Gamma\Gamma'}_{nn'}(f, f'), (\pi_2^{\dot{\to}}\, p) p')\end{array}),$$
$$\dot{\mathtt{ax}}(a)) \qquad \text{where } n = \lceil A\dot{\to}B \rceil$$

$$\downarrow_{\dot{\forall} x A}\; m \;\triangleq\; (n, (a_0 : \neg A_0)(a : A[x_n/x] \wedge (A[x_n/x] \to \dot{\forall} x\, A)),$$
$$I_n(\mathtt{inj}_n, (\Gamma, f, p) \mapsto \begin{array}{l}\mathtt{dest}\ \downarrow_{A[x_n/x]} (m\, x_n)\ \mathtt{as}\ (n', \Gamma', f', p') \\ \mathtt{in}\ (max(n, n'), \Gamma \cup \Gamma', \mathtt{join}^{\Gamma\Gamma'}_{nn'}(f, f'), mk^{\dot{\forall}}(p, p'))\end{array}),$$
$$\dot{\pi}_1(\dot{\mathtt{ax}}(a)) \qquad \text{where } n = \lceil \dot{\forall} x\, A \rceil$$

$$\uparrow_A \;:\; A \in F_\omega \;\longrightarrow\; A \in \mathcal{M}$$

$$\uparrow_{P(\vec{t})}\, (n, \Gamma, f, p) \;\triangleq\; k \mapsto k\,(n, \Gamma, f, p)$$

$$\uparrow_\bot\, (n, \Gamma, f, p) \;\triangleq\; \mathtt{throw}_{\alpha_0}\, \mathtt{flush}^{\Gamma}_{n}(f, p)$$

$$\uparrow_{A\dot{\to}B}\, (n, \Gamma, f, p) \;\triangleq\; m \mapsto \mathtt{dest}\ \downarrow_A m\ \mathtt{as}\ (n', \Gamma', f', p')\ \mathtt{in}\ \uparrow_B (max(n, n'), \Gamma \cup \Gamma', \mathtt{join}^{\Gamma\Gamma'}_{nn'}(f, f'), \dot{\mathtt{app}}^{\to}(p, p'))$$

$$\uparrow_{\dot{\forall} x A}\, (n, \Gamma, f, p) \;\triangleq\; t \mapsto \uparrow_{A[t/x]} (n, \Gamma, f, \dot{\mathtt{app}}^{\forall}(p, t))$$

Auxiliary lemmas

$$\texttt{flush}^{\Gamma}_{n} \;:\; \Gamma \subset F_n \wedge \Gamma \vdash \bot \;\longrightarrow\; \neg A_0 \vdash \bot$$

$$\begin{array}{lll}
\texttt{flush}^{\Gamma}_{0} & (\mathtt{I}_{\emptyset}, p) & \triangleq\; p \\
\texttt{flush}^{\Gamma}_{n+1} & (\mathtt{I}_{\mathtt{S}} f, p) & \triangleq\; \texttt{flush}^{\Gamma}_{n}(f, p) \\
\texttt{flush}^{\Gamma A}_{n+1} & (\mathtt{I}_{\mathtt{n}}(f, H), p) & \triangleq\; \texttt{flush}^{\Gamma}_{n}(f, H\,\Gamma\, f\; p)
\end{array}$$

$$\texttt{join}^{\Gamma_1\Gamma_2}_{n_1 n_2} \;:\; \Gamma_1 \subset F_{n_1} \;\wedge\; \Gamma_2 \subset F_{n_2} \;\longrightarrow\; \Gamma_1 \cup \Gamma_2 \subset F_{max(n_1,n_2)}$$

$$\begin{array}{lllll}
\texttt{join}^{\neg A_0 \neg A_0}_{00} & \mathtt{I}_{\emptyset} & \mathtt{I}_{\emptyset} & \triangleq & \mathtt{I}_{\emptyset} \\
\texttt{join}^{(\Gamma_1 A)(\Gamma_2 A)}_{(n+1)(n+1)} & \mathtt{I}_{\mathtt{n}}(f_1, H_1) & \mathtt{I}_{\mathtt{n}}(f_2, H_2) & \triangleq & \mathtt{I}_{\mathtt{n}}(\texttt{join}^{\Gamma_1'\Gamma_2'}_{nn} f_1 f_2, H_1) \\
\texttt{join}^{(\Gamma_1 A)\Gamma_2}_{(n+1)(n+1)} & \mathtt{I}_{\mathtt{n}}(f_1, H_1) & \mathtt{I}_{\mathtt{S}} f_2 & \triangleq & \mathtt{I}_{\mathtt{n}}(\texttt{join}^{\Gamma_1'\Gamma_2}_{nn} f_1 f_2, H_1) \\
\texttt{join}^{\Gamma_1(\Gamma_2 A)}_{(n+1)(n+1)} & \mathtt{I}_{\mathtt{S}} f_1 & \mathtt{I}_{\mathtt{n}}(f_2, H_2) & \triangleq & \mathtt{I}_{\mathtt{n}}(\texttt{join}^{\Gamma_1\Gamma_2'}_{nn} f_1 f_2, H_2) \\
\texttt{join}^{\Gamma_1\Gamma_2}_{(n+1)(n+1)} & \mathtt{I}_{\mathtt{S}} f_1 & \mathtt{I}_{\mathtt{S}} f_2 & \triangleq & \mathtt{I}_{\mathtt{S}}(\texttt{join}^{\Gamma_1\Gamma_2}_{nn} f_1 f_2) \\
\texttt{join}^{\Gamma_1\Gamma_2}_{n_1 n_2} & \mathtt{I}_{\mathtt{S}} f_1 & f_2 & \triangleq & \mathtt{I}_{\mathtt{S}}(\texttt{join}^{\Gamma_1\Gamma_2}_{n_1' n_2} f_1 f_2) \quad \text{if } n_1 = n_1' + 1 > n_2 \\
\texttt{join}^{(\Gamma_1 A_1)\Gamma_2}_{n_1 n_2} & \mathtt{I}_{\mathtt{n}_1'}(f_1, H_1) & f_2 & \triangleq & \mathtt{I}_{\mathtt{n}_1'}(\texttt{join}^{\Gamma_1\Gamma_2}_{n_1' n_2} f_1 f_2, H_1) \quad \text{if } n_1 = n_1' + 1 > n_2 \\
\texttt{join}^{\Gamma_1\Gamma_2}_{n_1 n_2} & f_1 & \mathtt{I}_{\mathtt{S}} f_2 & \triangleq & \mathtt{I}_{\mathtt{S}}(\texttt{join}^{\Gamma_1\Gamma_2}_{n_1 n_2'} f_1 f_2) \quad \text{if } n_1 < n_2' + 1 = n_2 \\
\texttt{join}^{\Gamma_1(\Gamma_2 A_2)}_{n_1 n_2} & f_1 & \mathtt{I}_{\mathtt{n}_2'}(f_2, H_2) & \triangleq & \mathtt{I}_{\mathtt{n}_2'}(\texttt{join}^{\Gamma_1\Gamma_2}_{n_1 n_2'} f_1 f_2, H_2) \quad \text{if } n_1 < n_2' + 1 = n_2
\end{array}$$

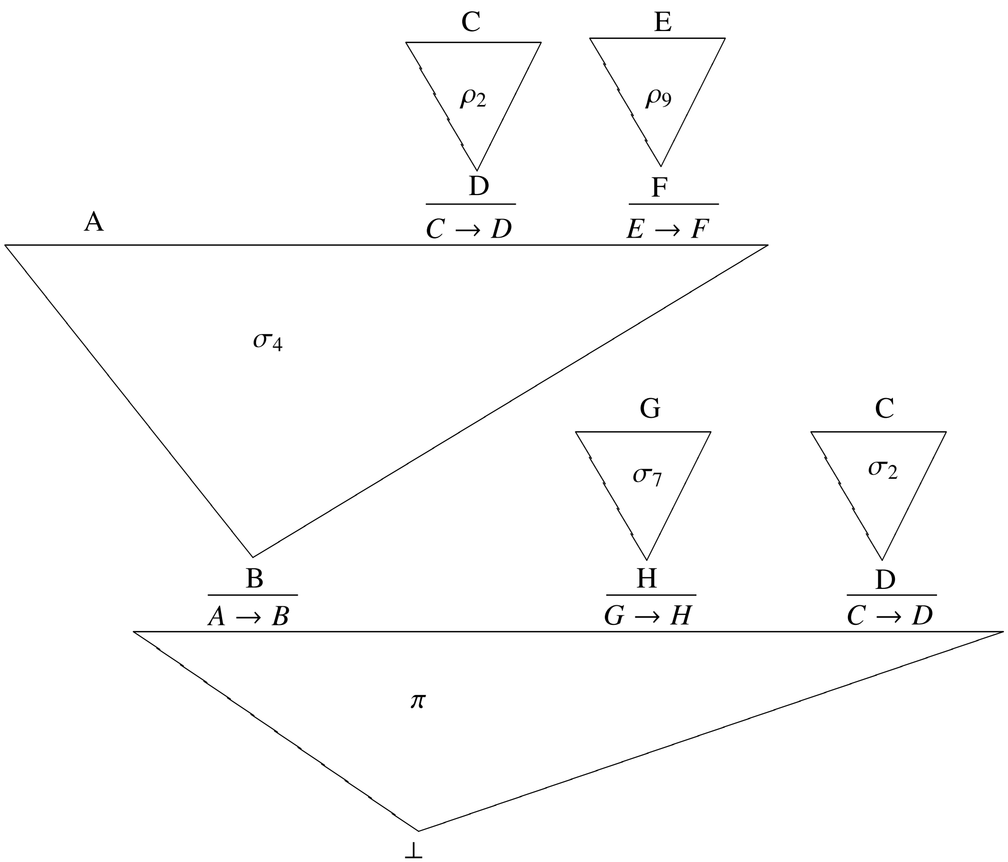
$$\texttt{inj}_n \;:\; (a_0 : \neg A_0) \subset F_n$$

$$\begin{array}{ll}
\texttt{inj}_0 & \triangleq\; \mathtt{I}_{\emptyset} \\
\texttt{inj}_{n+1} & \triangleq\; \mathtt{I}_{\mathtt{S}}(\texttt{inj}_n)
\end{array}$$

Final result

$$
\begin{array}{llll}
\texttt{compl}_{\texttt{A}_0} & : (\forall \mathcal{M} \forall \sigma [\![A_0]\!]^{\sigma}_{\mathcal{M}}) & \longrightarrow & \vdash A_0 \\
\texttt{compl}_{\texttt{A}_0} & \psi & \triangleq & \texttt{catch}_{\alpha_0}\,(\texttt{dest}\ \downarrow_A (\psi\, \mathcal{M}_0\, \texttt{id})\, \texttt{as}\, (n, \Gamma, f, p)\, \texttt{in}\, \dot{\texttt{c}}\texttt{c}_{a_0}.\texttt{flush}^{\Gamma}_{n}(f, p))
\end{array}
$$

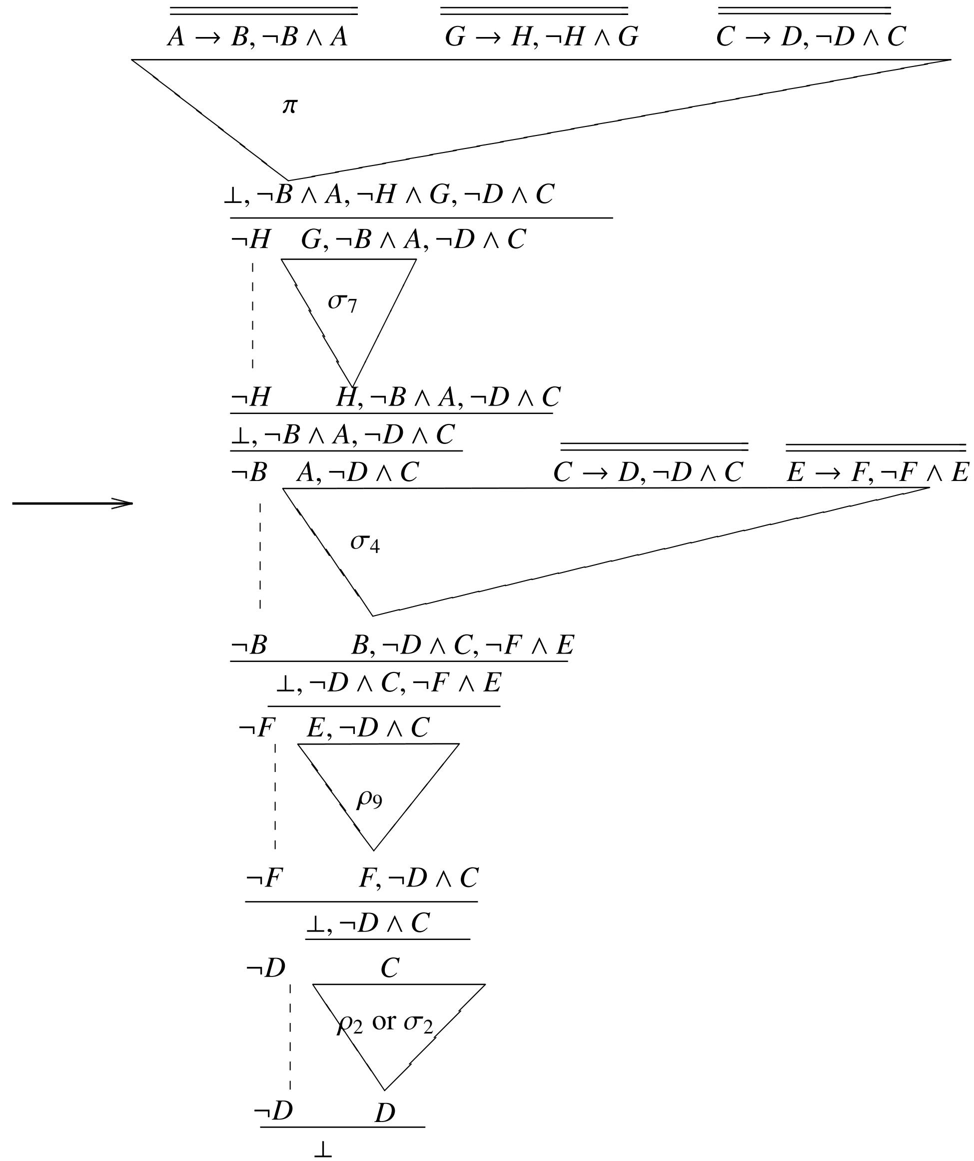
Intuition about the computation content



$\lceil A \to B \rceil = 4$ $\qquad$ $\lceil E \to F \rceil = 9$

$\lceil C \to D \rceil = 2$ $\qquad$ $\lceil G \to H \rceil = 7$

β-normal η-expanded meta-proof



resulting object language classical prof (not cut-free!)

Open questions

Can we modify the proof such that it produces normal forms?

Can we get rid of the total ordering of formulas?

$\hookrightarrow$ uses a memory to share information between branches of the meta-language proof and delimits its effect by a `reset`!