A computational look at soundness, completeness and reducibility

Hugo Herbelin

Journée de rentrée PPS

22 September 2014
Context

- The Curry-Howard correspondence suggests to think of proofs as programs
- With proof assistants, we tend to see models or semantics (functions, predicates, ...) as syntax of the meta-language
- With this respect, what does a proof of normalisation or a proof of soundness, or a proof of completeness say?
Outline

- Semantic normalisability via soundness and completeness
- Reflecting semantic normalisability: Normalisation-by-Evaluation
- A comparison with normalisation by reducibility
- The case of Tarskian semantics (Gödel’s completeness)
Simply-typed $\lambda$-calculus for minimal negative propositional logic

\[
A ::= X \mid A \rightarrow A \\
\Gamma ::= \emptyset \mid \Gamma, A
\]

\[
\frac{\Gamma, A, A_1, \ldots, A_n \vdash A}{\Gamma, A, A_1, \ldots, A_n \vdash A} \quad \text{Var}_n
\]

\[\Gamma, A \vdash B \quad \frac{\Gamma, A \vdash B}{\Gamma \vdash A \rightarrow B} \quad \text{Lam} \quad \frac{\Gamma \vdash A \rightarrow B}{\Gamma \vdash B} \quad \text{App} \]

Weakening is admissible:

\[
\frac{\Gamma \subseteq \Gamma'}{\Gamma \vdash A \quad \frac{\Gamma \vdash A}{\Gamma' \vdash A}} \quad \text{Weak}
\]

We write $\Gamma \vdash_{nf} A$ for a normal derivation and $\Gamma \vdash_{neut} A$ for a neutral derivation, i.e. a normal derivation made of an iteration of App over Var.
Semantic normalisability

Assume we have a sound and **strongly** complete notion of validity in minimal propositional logic w.r.t some kinds of model (Kripke model, Tarskian model, phase/pretopology semantics, ...)

\[ \text{sound} : \mathcal{T} \vdash A \Rightarrow \mathcal{T} \models A \]

\[ \text{compl} : \mathcal{T} \models A \Rightarrow \mathcal{T} \vdash_{nf} A \]

Then, composing soundness and strong completeness gives normalisability (C. Coquand 1993, Okada 2002, ...)

Applicable to intuitionistic logic, classical logic by taking for \( \mathcal{T} \) all instances of ex falso quodlibet and all instances of double-negation elimination respectively.
An approach popular from the 90’s: Normalisation-by-Evaluation

Let $\pi : \mathcal{T} \vdash A$, then $\text{compl}(\text{sound}(\pi)) : \mathcal{T} \vdash_{nf} A$

If the meta-theory has cut-elimination, then, for any $\pi$, $\text{compl}(\text{sound}(\pi))$ can be turned into some normal proof $\pi'$, which is determined by how cut-elimination is implemented.

E.g., if the meta-theory is based on $\lambda$-calculus, the computational content is easy to observe:

- Soundness maps $\pi$ to its interpretation in the model
- Strong completeness depends on the model:
  - E.g., for Tarskian semantics, the situation is delicate.
  - However, for Kripke/Beth (C. Coquand 1993) or phase/pretopology semantics (Okada 2002), completeness can be done quite syntactically so that $\pi'$ is the $\beta$-normal $\eta$-long normal form of $\pi$. 
An approach popular from the 90’s: Normalisation-by-Evaluation

Normalisation-by-Evaluation relies on that the latter can itself be proved:

- If the meta-theory is type-theoretic, statements about proofs can be expressed and the fact that $\pi'$ is related to $\pi$ can be proved explicitly.

- Even the meta-theory is not type-theoretic but has enough higher-order functions, a proof of validity can be lifted at the level of a functional object which we can talk about.
Soundness in the case of Kripke semantics

Let \((\mathcal{K}, \geq, \models_{\mathcal{K}}, \text{mon}_{\mathcal{X}})\) be a Kripke model.

We extend \(\models_{\mathcal{K}}\) to all formulae:

\[
\begin{align*}
\forall w \models_{\mathcal{K}} X & \triangleq \forall w \models_{X} \\
\forall w \models_{\mathcal{K}} A \rightarrow B & \triangleq \forall w' \geq w (w' \models_{\mathcal{K}} A \Rightarrow w' \models_{\mathcal{K}} B)
\end{align*}
\]

We show that \(\text{mon}_{\mathcal{X}}\) extends to to all formulae:

\[
\text{mon}_{A}(h) : w \models_{\mathcal{K}} A \Rightarrow w' \models_{\mathcal{K}} A
\]

whenever \(h : w \leq w'\).

We also write \(\text{refl}_{w} : w \geq w\).
Soundness in the case of Kripke semantics

For simplicity, we restrict ourselves to finite theories, written $\Gamma$.

We write $\Gamma \models_\mathcal{K} A \triangleq \forall w \ (w \vdash_\mathcal{K} \Gamma \Rightarrow w \vdash_\mathcal{K} A)$

We write $\Gamma \models A \triangleq \forall (\mathcal{K}, \geq, \models_X, \text{mon}_X) \Gamma \models_\mathcal{K} A$

We map syntax to semantics by induction on the derivation:

\[
\begin{align*}
\llbracket \top \rrbracket_\mathcal{K} & : \quad \Gamma \vdash A \ \Rightarrow \ \Gamma \models_\mathcal{K} A \\
\llbracket \text{Var}_n \rrbracket_\mathcal{K} & \triangleq \lambda w. \lambda \vec{\alpha}. \alpha_n \\
\llbracket \text{Lam}(\pi) \rrbracket_\mathcal{K} & \triangleq \lambda w. \lambda \vec{\alpha}. \lambda w'. \lambda h. \lambda \alpha. [\llbracket \pi \rrbracket_\mathcal{K} w'(\text{mon}_\Gamma(h)(\vec{\alpha}), \alpha')] \\
\llbracket \text{App}(\pi, \pi') \rrbracket_\mathcal{K} & \triangleq \lambda w. \lambda \vec{\alpha}. [\llbracket \pi \rrbracket_\mathcal{K} w \text{refl}_w (\llbracket \pi' \rrbracket_\mathcal{K} w \vec{\alpha})]
\end{align*}
\]

\textit{sound} : \quad \Gamma \vdash A \ \Rightarrow \ \Gamma \models A

\textit{sound} \quad \pi \quad \triangleq \lambda (\mathcal{K}, \geq, \models_X, \text{mon}_X). [\llbracket \pi \rrbracket_\mathcal{K}]

9
Completeness in the case of Kripke semantics

We use the universal model of contexts \( \mathcal{U} \):

\[
\begin{align*}
\mathcal{K} & \triangleq \text{Contexts} \\
\Gamma \leq \Gamma' & \triangleq \Gamma \subset \Gamma' \\
\Gamma \vdash_X & \triangleq \Gamma \vdash_{nf} X \\
\text{mon}_X & \triangleq \text{Weak}
\end{align*}
\]

We write \( \text{shift}_{\Gamma,A} : \Gamma \subset (\Gamma, A) \).

We map semantics to syntax back and forth by induction on the type:

\[
\begin{align*}
\uparrow^\Gamma_A & \quad \Gamma \vdash_{neut} A \Rightarrow \Gamma \vdash_{\mathcal{U}} A \\
\uparrow^\Gamma_X & \quad \pi \triangleq \pi \\
\uparrow^\Gamma_{A \to B} & \quad \pi \triangleq \lambda \Gamma'. \lambda h. \lambda \alpha. \uparrow^\Gamma_B (\text{App}(\text{Weak}(h, \pi), \downarrow^\Gamma \alpha)) \\
\downarrow^\Gamma_A & \quad \Gamma \vdash_{\mathcal{U}} A \Rightarrow \Gamma \vdash_{nf} A \\
\downarrow^\Gamma_X & \quad \alpha \triangleq \alpha \\
\downarrow^\Gamma_{A \to B} & \quad \alpha \triangleq \text{Lam}(\downarrow^\Gamma_B (\alpha (\Gamma, A) \text{shift}_{\Gamma,A} (\uparrow^\Gamma_A \text{Var}_0)))
\end{align*}
\]

Hence: \( \text{compl} \ v \ \triangleq \downarrow^\Gamma_A (v \ \mathcal{U} \ \Gamma (\uparrow^\Gamma \text{Var}_1)) \)
Normalisation-by-Evaluation with Kripke models (C. Coquand)

We shortcut the generalisation over all models and take instead

\[ \text{sound} : \Gamma \vdash A \Rightarrow \Gamma \vDash_{\mathcal{U}} A \]
\[ \text{compl} : \Gamma \vDash_{\mathcal{U}} A \Rightarrow \Gamma \vdash_{nf} A \]

We define \( \alpha \sim_{A} \pi \) by induction on \( A \) with iteration on \( \Gamma \).

We prove: \( \forall \Gamma A \forall \pi : \Gamma \vdash A \ (\text{sound}(\pi) \sim_{A} \pi) \)

We prove: \( \forall \Gamma A \forall \pi : \Gamma \vdash A \ \forall \alpha : \Gamma \vDash A \ (\alpha \sim_{A} \pi \Rightarrow \text{compl}(\alpha) =_{\beta \eta} \pi) \)

Hence \( \text{compl}(\text{sound}(\pi)) =_{\beta \eta} \pi \)
Let us first annotate derivations with proof-terms.

\[ p, q ::= a \mid p \ q \mid \lambda a. p \]

\[ \Gamma ::= \emptyset \mid \Gamma, a : A \]

\[ \begin{array}{c}
\frac{}{\Gamma, a : A, a_1 : A_1, \ldots, a_n : A_n \vdash a : A} \quad \text{Var}_n^a \\
\end{array} \]

\[ \begin{array}{c}
\frac{\Gamma, a : A \vdash p : B}{\Gamma \vdash \lambda a.p : A \rightarrow B} \quad \text{Lam}^{\lambda a.p} \\
\end{array} \]

\[ \begin{array}{c}
\frac{\Gamma \vdash p : A \rightarrow B \quad \Gamma \vdash q : A}{\Gamma \vdash p \ q : B} \quad \text{App}^{pq} \\
\end{array} \]
Reduction is taken to be $\beta$-reduction and $\eta$-expansion

\[
\begin{align*}
  t & \rightarrow_{\beta\eta} t' & u & \rightarrow_{\beta\eta} u' & t & \rightarrow_{\beta\eta} t' \\
  tu & \rightarrow_{\beta\eta} t'u & tu & \rightarrow_{\beta\eta} tu' & \lambda x.t & \rightarrow_{\beta\eta} \lambda x.t' \\
  (\lambda x.t)u & \rightarrow_{\beta\eta} t[x := u] & x \text{ fresh in } t & \rightarrow_{\beta\eta} \lambda x.(t \, x)
\end{align*}
\]
Reducibility/realisability semantics

A set $X$ is closed by anti-reduction if $p[x := q]r \in X \Rightarrow (\lambda x. p)q r \in X$.

Let an untyped reducibility/realisability model be an assignment $\rho$ of atoms to sets closed by anti-reduction.

We define:

\[ p \triangleright X \quad \triangleq \quad p \in \rho(X) \]
\[ p \triangleright A \rightarrow B \quad \triangleq \quad \forall q (q \triangleright A \Rightarrow p q r B) \]

\[ \overrightarrow{a : B} \vdash p \triangleright A \quad \triangleq \quad \forall \overrightarrow{q} (q \triangleright \overrightarrow{B} \Rightarrow p[\overrightarrow{a := \overrightarrow{q}}] r A) \]

Anti-reduction scales to arbitrary type:

\[ \text{anti} : p[x := q]r r A \Rightarrow (\lambda x. p)q r r A \]
Normalisation by reducibility: adequacy

Soundness/Adequacy: \( \forall \Gamma A \forall p (\Gamma \vdash p : A \Rightarrow \Gamma \vdash p \text{ r } A) \)

Proof by induction on the derivation:

\[
\begin{align*}
\llbracket p \rrbracket^p &: \quad \Gamma \vdash p : A \Rightarrow \Gamma \vdash p \text{ r } A \\
\llbracket \Var_n \rrbracket^a &: \quad \lambda \vec{q}.\lambda \vec{\alpha}.\alpha_n \\
\llbracket \Lam(\pi) \rrbracket^{\lambda a.p} &: \quad \lambda \vec{q}.\lambda \vec{\alpha}.\lambda q'.\lambda \alpha'.\anti(\llbracket \pi \rrbracket^p (\vec{q} q')(\vec{\alpha} \alpha')) \\
\llbracket \App(\pi, \pi') \rrbracket^{pp'} &: \quad \lambda \vec{q}.\lambda \vec{\alpha}. \llbracket \pi \rrbracket^p \vec{q} \vec{\alpha} (\llbracket \pi' \rrbracket^{p'} \vec{q} \vec{\alpha})
\end{align*}
\]
Normalisation by reducibility: escape lemma

We define $p \equiv \exists p' (p' \text{ nf} \land p \rightarrow_{\beta\eta} p')$ and take the normalisation model defined by $p \equiv p_w n$ and which satisfies stability by anti-reduction.

Let $\text{App}_{wn}$ proves $a \vec{q}_w n \land q' \equiv q' \equiv q_w n \Rightarrow a \vec{q} q' \equiv q_w n$, $\text{Var}_{wn} a$ proves $a \equiv q_w n$ and $\text{Lam}_{wn}^a$ proves $t \equiv \lambda a.t \equiv t_w n$.

Escape lemma: $\forall A \left\{ \forall \vec{q} (a \vec{q} \equiv q_w n \Rightarrow a \vec{q} r A) \land \forall p (p \equiv p_w n \Rightarrow p \equiv p_w n) \right\}$

Proof mutually by induction on the type:

\[
\begin{align*}
\uparrow_A & \quad a \vec{q} \equiv q_w n \Rightarrow a \vec{q} r A \\
\uparrow_X & \quad \pi \equiv \pi \\
\uparrow_{A \Rightarrow B} & \quad \pi \equiv \lambda q'. \lambda \alpha. \uparrow_B (\text{App}_{wn}(\pi, \downarrow_A \alpha)) \\
\downarrow_A & \quad p \equiv p_w n \Rightarrow p_w n \\
\downarrow_X & \quad \alpha \equiv \alpha \\
\downarrow_{A \Rightarrow B} & \quad \alpha \equiv \text{Lam}_{wn}^a(\downarrow_B (\alpha \equiv \alpha (\uparrow_A (\text{Var}_{wn} a)))))
\end{align*}
\]
Normalisation by typed reducibility

It becomes obvious that we can generalise realisability and Kripke semantics into a typed notion of reducibility.

Notes:

- To emphasise the similarity of computational content, we produced $\eta$-long normal form. We could also reason avoiding $\eta$.

- The proof scales to first-order universal quantification and to positive connectives when interpreted negatively.

- Interpreting positive connectives positively raises problems (see Ilik 2010).

- This kind of normalisation proof is also related to Type-Directed Partial Evaluation.
Tarskian semantics: soundness

Let $\mathcal{M}$ be a Tarskian model, i.e. an interpretation $\rho_\mathcal{M}$ of object-language atoms $X$ as meta-language atoms. Truth and validity are defined by:

$$\models_\mathcal{M} X \triangleq \rho_\mathcal{M}(X)$$

$$\models_\mathcal{M} A \rightarrow B \triangleq \models_\mathcal{M} A \Rightarrow \models_\mathcal{M} B$$

$$\Gamma \models_\mathcal{M} A \triangleq \models_\mathcal{M} \Gamma \Rightarrow \models_\mathcal{M} A$$

$$\models_\mathcal{M} A \triangleq \forall \mathcal{M} \models_\mathcal{M} A$$

Soundness (for minimal logic) works as in the Kripke and reducibility cases:

$$[[\Gamma]]_\mathcal{M} : \Gamma \vdash A \Rightarrow \Gamma \models_\mathcal{M} A$$

$$[[\text{Var}_n]]_\mathcal{M} \triangleq \lambda \vec{\alpha}.\alpha_n$$

$$[[\text{Lam}(\pi)]]_\mathcal{M} \triangleq \lambda \vec{\alpha}.\lambda \alpha.[[\pi]]_\mathcal{M}(\vec{\alpha}, \alpha')$$

$$[[\text{App}(\pi, \pi')]_\mathcal{M} \triangleq \lambda \vec{\alpha}.[[\pi]]_\mathcal{M}\vec{\alpha} ([[\pi']]_\mathcal{M}\vec{\alpha})$$

$$\text{sound} : \Gamma \vdash A \Rightarrow \Gamma \not\models A$$

$$\text{sound} \quad \pi \triangleq \lambda \mathcal{M}.[[\pi]]_\mathcal{M}$$
Tarskian semantics: completeness

There are several proofs of completeness (for classical logic):

- Henkin’s proof
- Beth-Hintikka-Kanger-Schütte’s proofs of *strong* completeness
- Rasiowa-Sikorski’s variant of Henkin’s proof
- ...

They are constructive as soon as:

- We interpret \( \bot \) by an arbitrary formula (stronger than all other formulae)
- We interpret positive formulae negatively
- We strictly consider Tarskian semantics rather than two-valued semantics which would require a functional reification axiom: \( \forall x \exists b (b = true \iff P(x)) \implies \exists f \forall x (f(x) = true \iff P(x)) \), which itself would require an instance of unique choice and classical logic.
Henkin’s proof: Assumptions on the object language

We assume to have a distinguished atom $\bot$ and we reason in the theory $\text{Class} \triangleq \{ \neg \neg A \rightarrow A \mid A \text{ formula} \}$.

The following rules are then admissible:

$$\frac{\text{Class}, \Gamma, A \rightarrow B \vdash \bot}{\text{Class}, \Gamma \vdash A} \quad \text{Proj}_1 \quad \frac{\text{Class}, \Gamma, A \rightarrow B \vdash \bot}{\text{Class}, \Gamma \vdash \bot B} \quad \text{Proj}_2$$

$$\frac{\text{Class}, \Gamma \vdash \bot \neg \neg A}{\text{Class}, \Gamma \vdash A} \quad \text{Dn}$$
Tarskian semantics: A computational presentation of Henkin’s proof

Let $[A]$ and $\phi$ form a Gödel’s numbering of implicative formulae such that $[\phi(n)] = n$.

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

\[
\frac{\neg A_0 \subset F_0}{I_0} \quad \frac{\Gamma \subset F_n}{\Gamma \subset F_{n+1}} \quad \frac{\Gamma \subset F_n \quad \forall \Gamma' \subset F_n (\text{Class}, \Gamma', \phi(n) \vdash \bot \Rightarrow \text{Class}, \neg A_0 \vdash \bot)}{I_n}
\]

The syntactic model $\mathcal{M}_0$ is defined by $\rho_\mathcal{M}(X) \triangleq F_\omega \vdash X$. 

21
The core of the proof

\[ \uparrow_A : F_\omega \vdash A \rightarrow \models_M A \]
\[ \uparrow_{\rightarrow} (n, \Gamma, f, \pi) \triangleq (n, \Gamma, f, \pi) \]
\[ \uparrow_{\rightarrow B} (n, \Gamma, f, \pi) \triangleq m \mapsto \text{dest } \downarrow_A m \text{ as } (n', \Gamma', f', \pi') \]
\[ \text{in } \uparrow_B (\max(n, n'), \Gamma \cup \Gamma', \text{join}_nn'(f, f'), \text{App}(\pi, \pi')) \]

\[ \downarrow_A : \models_M A \rightarrow F_\omega \vdash A \]
\[ \downarrow_X m \triangleq m \]
\[ \downarrow_{\rightarrow B} m \triangleq (n, (\neg A_0, A \rightarrow B), \]
\[ I_n(\text{inj}_n, (\Gamma, f, \pi) \mapsto \text{dest } \downarrow_B (m(\uparrow_A (n, \Gamma, f, \text{Proj}_1 \pi))) \text{ as } (n', \Gamma', f', \pi') \]
\[ \text{in flush}_{\max(n, n')} (\text{join}_{nn'}(f, f'), \text{App} (\text{Proj}_2 \pi, \pi')) \), \]
\[ \text{Var}_1 \)

where \( n = [A \rightarrow B] \)
### Auxiliary lemmas

\[
\begin{align*}
\text{flush}^\Gamma_n & : \Gamma \subset F_n \land \Gamma \vdash \_ \quad \rightarrow \, \vdash A_0 \vdash \_ \\
\text{flush}_0^\Gamma & (I_\emptyset, \pi) \quad \triangleq \quad \pi \\
\text{flush}_{n+1}^\Gamma & (I_S^f, \pi) \quad \triangleq \quad \text{flush}^\Gamma_n(f, \pi) \\
\text{flush}_{n+1}^{\Gamma A} & (I_n(f, H), \pi) \quad \triangleq \quad H \Gamma f \pi \\
\text{join}^{\Gamma_1\Gamma_2}_{n_1n_2} & : \Gamma_1 \subset F_{n_1} \land \Gamma_2 \subset F_{n_2} \quad \rightarrow \quad \Gamma_1 \cup \Gamma_2 \subset F_{\text{max}(n_1,n_2)} \\
\text{join}^{\_A_0\_A_0}_{00} & I_\emptyset \quad \triangleq \quad I_\emptyset \\
\text{join}^{(\Gamma_1A)(\Gamma_2A)}_{(n+1)(n+1)} & I_n(f_1, H_1) \quad I_n(f_2, H_2) \quad \triangleq \quad I_n(\text{join}^{\Gamma_1\Gamma_2}_{n_1n_2} f_1 f_2, H_1) \\
\text{join}^{(\Gamma_1A)\Gamma_2}_{(n+1)(n+1)} & I_n(f_1, H_1) \quad I_S f_2 \quad \triangleq \quad I_n(\text{join}^{\Gamma_1\Gamma_2}_{n_1n_2} f_1 f_2, H_1) \\
\text{join}^{\Gamma_1(\Gamma_2A)}_{(n+1)(n+1)} & I_S f_1 \quad I_n(f_2, H_2) \quad \triangleq \quad I_n(\text{join}^{\Gamma_1\Gamma_2}_{n_1n_2} f_1 f_2, H_2) \\
\text{join}^{\Gamma_1\Gamma_2}_{n_1n_2} & I_S f_1 \quad I_S f_2 \quad \triangleq \quad I_S(\text{join}^{\Gamma_1\Gamma_2}_{n_1n_2} f_1 f_2) \\
\text{join}^{(\Gamma_1A)\Gamma_2}_{n_1n_2} & I_{n_1}'(f_1, H_1) \quad f_2 \quad \triangleq \quad I_{n_1}'(\text{join}^{\Gamma_1\Gamma_2}_{n_1n_2} f_1 f_2, H_1) \quad \text{if } n_1 = n_1' + 1 > n_2 \\
\text{join}^{\Gamma_1\Gamma_2}_{n_1n_2} & f_1 \quad I_S f_2 \quad \triangleq \quad I_S(\text{join}^{\Gamma_1\Gamma_2}_{n_1n_2} f_1 f_2) \quad \text{if } n_1 < n_2' + 1 = n_2 \\
\text{join}^{\Gamma_1(\Gamma_2A)}_{n_1n_2} & f_1 \quad I_{n_2}'(f_2, H_2) \quad \triangleq \quad I_{n_2}'(\text{join}^{\Gamma_1\Gamma_2}_{n_1n_2} f_1 f_2, H_2) \quad \text{if } n_1 < n_2' + 1 = n_2 \\
\text{inj}_n & : \vdash A_0 \subset F_n \\
\text{inj}_0 & \triangleq \ I_\emptyset \\
\text{inj}_{n+1} & \triangleq \ I_S(\text{inj}_n)
\end{align*}
\]
Final completeness result

class_0 : \forall A (\text{Class} \models_{\mathcal{M}_0} \downarrow \sim A \rightarrow A)
class_0 \triangleq \lambda A. \lambda m. \uparrow_A (\text{dest} \downarrow \sim_A m \text{ as } (n, \Gamma, f, \pi) \text{ in } (n, \Gamma, f, \text{Dn\pi}))

\text{compl}_{A_0} : \forall M \text{Class} \models_{\mathcal{M}} A_0 \rightarrow \text{Class} \vdash A_0
\text{compl}_{A_0} \psi \triangleq \text{Dn(Lam}(\text{dest} \downarrow_{A_0} (\psi \mathcal{M}_0 \text{ class}_0) \text{ as } (n, \Gamma, f, \pi) \text{ in } \text{flush}_n(f, \text{App(Var}_{\Gamma}, \pi)) \text{)))}
Henkin’s proof: comments

In Kripke semantics, knowledge can be extended whenever a new assumption is known.

In Tarskian semantics instead, knowledge cannot be extended. However, we can consider that an assumption holds whenever we know how to get rid of it.

*The model construction is computationally a type of continuation: any formula can be added to the model as soon as its addition comes with a continuation showing that it preserves consistency*

No concrete “model” is built, even though the ordering matters on what the resulting proof is.
Henkin’s proof: further comments

When composing Henkin’s completeness with soundness:

- the resulting proof is not necessary normal
- even if the structure of the initial proof is used, it is “damaged” in the resulting proof
- can we twist Henkin’s proof so as to return normal proofs?
- use effects to mimic the semantic and get a normal form?
A few references

NbE for $\lambda$-calculus: Berger-Schwichtenberg 1991
Computational analysis of reducibility proofs: Berger 1993
NbE by Kripke semantics: C. Coquand 1993, 2002; NbE by phase semantics: Okada 2002
Semantic normalisation for classical NbE: Herbelin-Gyesik-Lee 2010

...