An analysis of the constructive content of Henkin's proof of Gödel's completeness theorem
DRAFT

Hugo Herbelin and Danko Ilik

December 6, 2016

Abstract

Gödel's completeness theorem for first-order logic is one of the best-known theorems of logic. Central to any foundational course in logic, it connects the notion of valid formula to the notion of provable formula.

There are various views on the completeness theorem, various presentations, various formalisations, various proofs of it.

We survey the most standard different approaches and eventually focus on a formalization of a slight modification of Henkin's proof in intuitionistic second order arithmetic.

In many cases, proofs *compute*: this is the Curry-Howard correspondence between proofs and programs, and its further extensions. In particular, it is standard that proofs of completeness of intuitionistic logic with respect to various semantics such as Kripke or Beth semantics can be rephrased as programs which turn proofs of validity for these semantics into proofs of derivability.

We apply this approach to Henkin's proof to rephrase it as a program which transforms proofs of validity with respect to Tarski semantics into proofs of derivability.

By doing so, we hope to shed an "effective" light on the relation between Tarski semantics and syntax: semantics proofs are syntactic objects of a meta-language that we can manipulate and compute with like ordinary syntax.

Contents

1 Preliminaries

1.1 The completeness theorem

The completeness theorem for first-order logic is one of the most basic and standard theorems of logic. Proved by Gödel in 1929 [22] as an answer to a question raised by Hilbert and Ackermann in 1928 [32], it states that any of the standard formal systems for defining provability in first-order logic is complete enough to contain a derivation for every valid formula. A formula A is *valid* when it is true under all interpretations of its primitive symbols over any domain of quantification.

Let $\mathcal{L}$ be a language of first-order logic, i.e. the data of a set[1] $\mathcal{F}un$ of function symbols, each of them coming with an arity $a_f \geq 0$, as well as of a set $\mathcal{P}red$ of predicate symbols, each of them also coming with an arity $a_P \geq 0$. We call constants the function symbols of arity 0. When studying the computational content of Gödel's completeness in Section 2, we shall restrict the language to a countable one but the rest of this section does not require restrictions on the cardinal of the language.

We let f range over $\mathcal{F}un$ and P range over $\mathcal{P}red$. Let x range over a countable set $\mathcal{X}$ of variables and let t range over the set $\mathcal{T}erm$ of terms over $\mathcal{L}$ as described by the following grammar:

$$t \quad ::= \quad x \mid f(t_1, ..., t_{a_f})$$

Let A range over the set $\mathcal{F}orm$ of formulae over $\mathcal{L}$ as described by the following grammar:

$$A \quad ::= \quad P(t_1, ..., t_{P_f}) \mid \dot\bot \mid A \dot\Rightarrow A \mid \dot\forall x\, A$$

Note that since we are in classical logic, we can restrict ourselves to a language of *negative* connectives and quantifiers, such as $\dot\Rightarrow$, $\dot\forall$ and $\dot\bot$ (handling conjunction is straightforward, while the case of disjunction and existential quantification will be discussed in Section 2.5). We write a dot over each notation of a connective or quantifier to clarify that we are defining an object logic, i.e. a logic we are talking about. In particular, we will use notations without a dot for the connectives and quantifiers of the meta-logic in which the completeness theorem is stated and proved. We take $\dot\bot$ as a primitive connective which allows to express consistency in a direct way as the non-provability of $\dot\bot$. Negation can then be defined as $\dot\neg A \triangleq A \dot\Rightarrow \dot\bot$. Also, in $\dot\forall x\, A$, we say that x is a binding variable which binds all occurrences of x in A (if any). If the occurrence of a variable is not in the scope of a $\dot\forall$ with same name, it is called free.

Let us write Γ for finite contexts of hypotheses, as defined by the following grammar:

$$\Gamma \quad ::= \quad \epsilon \mid \Gamma, A$$

In particular, ϵ denotes the empty context, which we might also not write at all, as e.g. in $\vdash A$ standing for $\epsilon \vdash A$.

We assume having chosen a formal system for provability in classical first-order logic, e.g. one of the axiomatic systems given in Frege [17] or in Hilbert and Ackermann [32], or one of the systems such as Gentzen-Jaśkowski's natural deduction [38, 21] or Gentzen's sequent calculus [21], etc., and we write $\Gamma \vdash A$ for the statement that A is provable under the finite context of hypotheses Γ. If $\mathcal{M}$ is a model for classical logic and σ an interpretation of the variables from $\mathcal{X}$ in the model, we write $\mathcal{M} \vDash_\sigma A$ for the statement expressing that A is true in the model $\mathcal{M}$. Validity of A under assumptions Γ, written $\Gamma \vDash A$ is defined to be $\forall \mathcal{M}\, \forall \sigma\, (\mathcal{M} \vDash_\sigma \Gamma \Rightarrow \mathcal{M} \vDash_\sigma A)$ where $\mathcal{M} \vDash_\sigma \Gamma$ is the conjunction of all $\mathcal{M} \vDash_\sigma B$ for every B in Γ, i.e. $\bigwedge_{B \in \Gamma} \mathcal{M} \vDash_\sigma B$. Note that $\Rightarrow$, $\forall$, $\wedge$ and later on, below, $\vee$, $\exists$, $\bot$, as well as derived $\neg$, represent the connectives and quantifiers of the meta-language.

We say that Γ is inconsistent if $\Gamma \vdash \dot\bot$ and consistent if $\Gamma \nvdash \dot\bot$, i.e. if $(\Gamma \vdash \dot\bot) \Rightarrow \bot$, i.e., if a contradiction in the object language is reflected as a contradiction in the meta-logic. We say that Γ has a model if there exists $\mathcal{M}$ and σ such that $\mathcal{M} \vDash_\sigma \Gamma$. The completeness theorem, actually a weak form of the completeness theorem as discussed in the next section, is commonly stated under one of the following classically but not intuitionistically equivalent forms:

$$\begin{array}{ll} C1. & \vDash A \;\Rightarrow\; \vdash A \\ C2. & \Gamma \text{ is consistent } \Rightarrow \Gamma \text{ has a model} \\ C3. & \Gamma, \dot\neg A \text{ has a model } \vee\; \Gamma \vdash A \end{array}$$

[1] We use here "set" in an informal way, not necessarily assuming the meta-language to be specifically set theory.

1.2 Weak and strong completeness

In a strong form, referred to as strong completeness[2], completeness states that any formula valid under some possibly infinite theory is provable under a finite subset of this theory. This is the most standard formulation of completeness in textbooks, and, as such, it is a key component of the compactness theorem. Also proved by Gödel [23], the compactness theorem states that any formula valid with respect to any finite subset of a theory is valid with respect to the whole theory. In contrast, completeness with respect to finite theories as stated above is referred to as weak completeness. Let $\mathcal{T}$ be a set of formulae and let $\mathcal{T} \vdash A$ mean the existence of a finite sequence Γ of formulae in $\mathcal{T}$ such that $\Gamma \vdash A$. Let $\mathcal{M} \models_\sigma \mathcal{T}$ be $\forall B \in \mathcal{T}\ \mathcal{M} \models_\sigma B$ and let the definitions of $\mathcal{T}$ is consistent and of $\mathcal{T}$ has model be extended accordingly. The strong formulations of the three perceptions at weak completeness above are now the following:

$$\begin{array}{ll} S1. & \mathcal{T} \models A \ \Rightarrow\ \mathcal{T} \vdash A \\ S2. & \mathcal{T} \text{ is consistent } \Rightarrow\ \mathcal{T} \text{ has a model} \\ S3. & \mathcal{T} \cup \{\neg A\} \text{ has a model } \vee\ \mathcal{T} \vdash A \end{array}$$

We shall consider the formalisation and computational content of strong completeness. Weak completeness will then arrive as a special case.

1.3 The standard existing proofs of completeness

Let us list the traditional proofs from the classic literature[3].

- Gödel's original proof [22] considers formulae in prenex form and works by induction on the number of quantifiers for reducing the completeness of first-order predicate logic completeness to the completeness of propositional logic.
- Henkin's proof [26] is related to statement 2: from the assumption that $\mathcal{T}$ is consistent, a syntactic model over the terms is built as a maximal consistent extension of $\mathcal{T}$ obtained by ordering the set of formulae and extending $\mathcal{T}$ with those formulae that preserve consistency, following the ordering.
- In the 1950's, a new kind of proof credited to Beth [9], Hintikka [34, 35], Kanger [39] and Schütte [57], independently, was given. The underlying idea is to build an infinite normal derivation, typically in sequent calculus. Rules are applied in a fair way, such that all possible combinations of rules are considered. If the derivation happens to be finite, a proof is obtained. Otherwise, by weak König's lemma, there is an infinite branch and this infinite branch gives rise to a countermodel. The intuition underlying this proof is then best represented by statement 3.
- In the 1950's also, Rasiowa and Sikorski [55] gave a variant of Henkin's proof by using the existence of an ultrafilter for the Lindenbaum algebra of classes of logically equivalent formulae, identifying validity with having value 1 in all interpretations of a formula within the two-value Boolean algebra $\{0, 1\}$. This is connected to Henkin's proof in the sense that Henkin's proof de facto implicitly builds an ultrafilter of the Lindenbaum algebra of formulae.

Our main contribution is the analysis in Section 2 of the computational content of Henkin's proof.

[2] We follow here a terminology dubbed by Henkin in his 1947 dissertation, according to [27].

However, in the context of intuitionistic logic, some authors use the weak and strong qualifying with different meanings. For instance, in Kreisel [41, 42], the statement $(\models A) \Rightarrow (\vdash A)$ is called strong completeness while weak completeness is the statement $(\models A) \Rightarrow \neg\neg(\vdash A)$.

In the context of semantic cut-elimination, e.g. in Okada [52], $(\models A) \Rightarrow (\vdash A)$ is only a weak form of completeness whose strong form is the statement $(\models A) \Rightarrow (\vdash_{\text{cut-free}} A)$, for a notion of cut-freeness similar to the notion of cut-freeness in Gentzen's sequent calculus or to Prawitz' notion of normal proofs [54] in natural deduction.

[3] We cite the most common proofs in the classic pre-1960 literature. Recent developments include e.g. Joyal's categorical presentation of a completeness theorem. We can also cite Berger's [56, Sec. 1.4.3] or Krivtsov [48] construction of a classical model from a Beth model for classical provability.

1.4 Models and truth

The interpretation of terms in a model $\mathcal{M}$ is given by a domain $\mathcal{D}$ and by an interpretation $\mathcal{F}$ of the symbols in $\mathcal{F}un$ such that $\mathcal{F}(f) \in \mathcal{D}^{a_f} \to \mathcal{D}$, where $\mathcal{D}^{a_f} \to \mathcal{D}$ denotes the set of functions of arity a_f over $\mathcal{D}$. Then, given an assignment $\sigma \in \mathcal{X} \to \mathcal{D}$ of the variables to arbitrary values of the domain, the interpretation of terms in $\mathcal{D}$ is given by:

$$\begin{array}{lcl} [\![x]\!]^{\sigma}_{\mathcal{M}} & \triangleq & \sigma(x) \\ [\![f(t_1, \ldots, t_{a_f})]\!]^{\sigma}_{\mathcal{M}} & \triangleq & \mathcal{F}(f)([\![t_1]\!]^{\sigma}_{\mathcal{M}}, \ldots, [\![t_{a_f}]\!]^{\sigma}_{\mathcal{M}}) \end{array}$$

Let us look at the different possible concrete definitions of a model. In particular, to define a two-valued interpretation of formulae, two approaches are generally considered.

- *Predicates as predicates.* The approach followed e.g. in the Handbook of Mathematical Logic [5] or the Handbook of Proof Theory [10] is to interpret formulae of the object language propositionally, i.e. as formulae of the meta-language. In this case, the interpretation is not explicitly two-valued and the interpretation depends on whether the meta-logic is classical or not. For instance, in a classical meta-language, the theory

  $$Classic \triangleq \{\dot{\neg}\dot{\neg}A \dot{\Rightarrow} A \mid A \in \mathcal{F}orm\}$$

  would be true in all models. On the other side, in an intuitionistic meta-language, a formula such as, say, $\dot{\neg}\dot{\neg}X \dot{\Rightarrow} X$ could not be proved true in all models, and, in a strongly anti-classical intuitionistic meta-language refuting double-negation elimination, it could even be proved that there are models which refute $\dot{\neg}\dot{\neg}X \dot{\Rightarrow} X$.

  As we shall see, this is not a problem for proving completeness, whose proof works by producing a particular syntactic model and looking at the truth of a sequent in this model, independently of whether the sequent is valid or not.

  However, the difference of interpretation of validity whether the meta-language is classical or intuitionistic is a problem for the soundness property, namely the standard statement that the provability of A implies the validity of A.

  As a consequence, for the definition of validity to be meaningful in this approach, whether the setting is intuitionistic or classical, we shall define explicitly classical validity as

  $$\mathcal{T} \vDash A \triangleq \forall \mathcal{M}\, \forall \sigma\, (\mathcal{M} \vDash_{\sigma} Classic \Rightarrow \mathcal{M} \vDash_{\sigma} \mathcal{T} \Rightarrow \mathcal{M} \vDash_{\sigma} A)$$

- *Predicates as binary functions.* Another common approach is to interpret formulae explicitly within a two-valued set $\{0, 1\}$, in such a way that $(\mathcal{M} \vDash_{\sigma} A)$ is $truth_{\mathcal{M}}(A, \sigma) = 1$ for some function $truth$ such that $truth_{\mathcal{M}}(A, \sigma) = 0 \vee truth_{\mathcal{M}}(A, \sigma) = 1$ holds. This is the approach followed e.g. in Rasiowa-Sikorski's proof, or also e.g. in [11, 58], among others.

  However, the proof of completeness requires to build a model in which truth is not a recursive function (Gödel's completeness is logically equivalent over recursive arithmetic to the existence of an infinite branch in any infinite binary tree [58], which is not a recursive process). Hence, $truth_{\mathcal{M}}(A, \sigma) = 0 \vee truth_{\mathcal{M}}(A, \sigma) = 1$ could hold only by requiring classical reasoning.

  It is known how to compute with classical logic in second-order arithmetic [25, 53, 46] and we could study the computational content of a formalisation of the completeness proof which uses this definition of truth. The extra need for classical reasoning in this approach looks however like a useless complication, so we shall favour the first approach.

So, to summarise, we shall adopt from now on the following definitions[4] of validity and existence of a model:

$$\begin{array}{lcl}\mathcal{T} \vDash A & \triangleq & \forall \mathcal{M} \forall \sigma\, (\mathcal{M} \vDash_\sigma \mathit{Classic} \Rightarrow \mathcal{M} \vDash_\sigma \mathcal{T} \Rightarrow \mathcal{M} \vDash_\sigma A) \\ \mathcal{T} \text{ has a model} & \triangleq & \exists \mathcal{M} \exists \sigma\, (\mathcal{M} \vDash_\sigma \mathit{Classic} \wedge \mathcal{M} \vDash_\sigma \mathcal{T})\end{array}$$

Two auxiliary choices of formalisation can be made[5].

- *Recursively-defined truth.* The approach followed e.g. in the Handbook of Mathematical Logic [5] or the Handbook of Proof Theory [10] is to have the model interpret only the predicate symbols and to have the truth of formulae defined recursively. This is obtained by giving an interpretation $\mathcal{M}_\mathcal{P}$ mapping any symbol $P \in \mathcal{P}red$ to a set $\mathcal{P}(P) \subset \mathcal{D}^{a_P}$. Then, the truth of a formula with respect to some assignment σ of the free variables is given recursively by:

$$\begin{array}{lcl}\mathcal{M} \vDash_\sigma P(t_1, \dots, t_{a_f}) & \triangleq & ([\![t_1]\!]^\sigma_\mathcal{M}, \dots, [\![t_{a_P}]\!]^\sigma_\mathcal{M}) \in \mathcal{P}(P) \\ \mathcal{M} \vDash_\sigma \dot{\bot} & \triangleq & \bot \\ \mathcal{M} \vDash_\sigma A \dot{\wedge} B & \triangleq & \mathcal{M} \vDash_\sigma A \wedge \mathcal{M} \vDash_\sigma B \\ \mathcal{M} \vDash_\sigma A \dot{\Rightarrow} B & \triangleq & \mathcal{M} \vDash_\sigma A \Rightarrow \mathcal{M} \vDash_\sigma B \\ \mathcal{M} \vDash_\sigma \dot{\forall} x\, A & \triangleq & \forall v \in \mathcal{D}\, \mathcal{M} \vDash_{\sigma \cup [x \leftarrow v]} A\end{array}$$

- *Axiomatically-defined truth.* A common alternative approach is to define truth as a subset $\mathcal{S}$ of closed formulae in the language of terms extended with the constants of $\mathcal{D}$, such that: $\bot$ is not in $\mathcal{S}$; $A \dot{\wedge} B$ is in $\mathcal{S}$ iff both A and B are; $A \dot{\Rightarrow} B$ is in $\mathcal{S}$ iff B is whenever A is; $\dot{\forall} x\, A$ is in $\mathcal{S}$ iff $A[x \leftarrow v]$ is for all values $v \in \mathcal{D}$; $A[x \leftarrow f(v_1, ..., v_n)]$ is in $\mathcal{S}$ iff $A[x \leftarrow v]$ is in $\mathcal{S}$ whenever $\mathcal{F}(f)(v_1, ..., v_n) = v$ for some value $v \in \mathcal{S}$.

We shall retain the first approach as it reflects more closely the view that a model is literally about replicating the uninterpreted (hence generic) symbols of the object language as symbols of the meta-language (the domain of quantification, the functional symbols, the predicate symbols) and in particular that $\forall \mathcal{M} \forall \sigma\, \mathcal{M} \vDash_\sigma A$ is the reflection of the universal closure of the object language formula A as a meta-language formula. Additionally, the first approach exempts us from defining the set of formulae enriched with constants from $\mathcal{D}$, which is convenient.

1.5 Regarding the meta-language as a formal system

Let M be the meta-language in which completeness is stated and O be the object language used to represent provability in first-order logic. In M, a proof of the validity of a formula A is essentially a proof of the universal second-order closure of A, seen as a formula of M, along the free predicate atoms of A and over the domain of individuals over which the predicate atoms range. Otherwise said, the weak completeness theorem in form C1 states that from a generic proof of A given in M one can extract a generic proof of A in the proof object language O (and conversely, the soundness theorem can be seen as stating an embedding of O into M). Similarly, a proof of the validity of a formula A with respect to an infinite theory $\mathcal{T}$ is a proof in M of the universal closure of $\ldots \Rightarrow B_i \Rightarrow \ldots \Rightarrow A$ for B_i ranging over $\mathcal{T}$, and, computationally speaking, statement S1 is a process to turn such a proof in M (which is a finite object using only a finite subset of $\mathcal{T}$ in the meta-language seen as a formal system) into a proof in O.

The key point is however that this transformation of a proof in M into a proof in O is done in M itself, and, within M itself, there is no way to observe a proof of the validity of A in M. The only way to extract information out of a proof of validity is by instantiating the free symbols of the interpretation of A in M by actual function and predicate symbols of M, i.e. by producing what at the end is a model, i.e. an effective domain and effective function and predicate expressions definable in M.

[4] For the record, note that, in the presence of only negative connectives, an equivalent way to define $\vDash A$ so that it means the same in an intuitionistic and classical setting is to replace the definition of $\mathcal{M} \vDash_\sigma P(t_1, ..., t_{a_P})$ by

$$\mathcal{M} \vDash_\sigma P(t_1, ..., t_{a_P}) \triangleq \neg\neg([\![t_1]\!]^\sigma_\mathcal{M}, \dots, [\![t_{a_P}]\!]^\sigma_\mathcal{M}) \in \mathcal{M}(P)$$

or even, saving a negation as in Krivine [44], by

$$\mathcal{M} \vDash_\sigma P(t_1, ..., t_{a_P}) \triangleq \neg([\![t_1]\!]^\sigma_\mathcal{M}, \dots, [\![t_{a_P}]\!]^\sigma_\mathcal{M}) \in \mathcal{M}(P)$$

Indeed, in these cases, the definition of truth becomes a purely negative formula for which intuitionistic and classical provability coincide.

[5] These auxiliary choices would have been relevant as well if we had chosen to represent truth as a map to $\{0, 1\}$.

1.6 Former results about the computational content of completeness proofs for intuitionistic logic

It is known that composing soundness and completeness for propositional or predicate logic gives a cut-elimination theorem, as soon as completeness is formulated in such a way that it produces a normal proof[6]. Now, if soundness and completeness are formulated in a meta-logic equipped with a normalisation procedure, e.g. by formulating completeness and soundness in a proofs-as-programs presentation of second-order arithmetic [43, 45, Ch. 9], one gets an effective cut-elimination theorem, namely an effective procedure which turns any non-necessary normal proof of $\Gamma \vdash A$ into a normal proof of $\Gamma \vdash A$. In the context of intuitionistic provability, i.e. of $\Gamma \vdash_I A$, and more generally in the context of typed λ-calculus, this has been explored several times under the name of semantic normalisation, also known as normalisation by evaluation, based on ideas by Berger and Schwichtenberg [8], and further studied under various angles and contexts in e.g. [14, 2], C. Coquand [13], Okada [52], Hermant [30], Lipton [31], ...

Let us recall how it works e.g. for implicative propositional logic using soundness and completeness with respect to Kripke models. Let $\mathcal{K}$ range over Kripke models $(\mathcal{W}, \leq, \Vdash_X)$ where $\leq$ is a preorder on $\mathcal{W}$ and $\Vdash_X$ a monotonic predicate over $\mathcal{W}$ for each propositional atom X. Let w range over $\mathcal{W}$, i.e. worlds in the corresponding Kripke models. Let us write $w \Vdash_{\mathcal{K}} A$ (resp. $w \Vdash_{\mathcal{K}} \Gamma$) for truth of A (resp. for the conjunction of the truth of all formulae in Γ) at world w in the Kripke model $\mathcal{K}$. In particular, $w \Vdash_{\mathcal{K}} A$ is extended from atoms to all formulae by defining $w \Vdash_{\mathcal{K}} A \dot{\Rightarrow} B \triangleq \forall w' (w' \geq w \Rightarrow w' \Vdash_{\mathcal{K}} A \Rightarrow w' \Vdash_{\mathcal{K}} B)$. Let us write $\Gamma \vDash_I A$ for validity of A relative to Γ at all worlds of all Kripke models, i.e. for the formula $\forall \mathcal{K} \forall w\, (w \Vdash_{\mathcal{K}} \Gamma \Rightarrow w \Vdash_{\mathcal{K}} A)$.

We shall write proofs of the meta-language as functions, defining a proof f of an implication with a notation of the form $f(x) \triangleq t$. We might also write $x \mapsto t$ for the proof of an implication and proofs of universal quantification, possibly also writing $(x : A) \mapsto t$ to make explicit that x is the name of a proof of A. We shall use application of functions, written $t\,u$, for modus ponens and instantiation of universal quantification. We shall use the notation $()$ for the canonical proof of an empty conjunction and the notation (t, u) for the proof of a conjunction, seen as a product type and obtained by taking the pair of the proofs of the components of the conjunction.

For instance, the proof that Kripke forcing is monotone, i.e. that $\forall w w'\, (w' \geq w \wedge w \Vdash A \Rightarrow w' \Vdash A)$, can be written as the following function $\Uparrow_A$, recursive in the structure of A, taking as arguments two worlds w and w':

$$
\begin{array}{llllllll}
\Uparrow_A^{w,w'} & : & w' \geq w & \wedge & w \Vdash A & \Rightarrow & w' \Vdash A \\
\Uparrow_X^{w,w'} & & (\,h & , & m &) & \triangleq & p_X^{w,w'}(h, m) \\
\Uparrow_{A \dot{\Rightarrow} B}^{w,w'} & & (\,h & , & m &) & \triangleq & w'' \mapsto (h' : w'' \geq w') \mapsto m\, w''\, (\mathit{trans}(h, h'))
\end{array}
$$

where p_X is the proof of monotonicity of $\Vdash_X$ and *trans* is the proof of transitivity of $\geq$, both coming with the definition of Kripke models, while, in the definition, h is a proof of $w' \geq w$ and m a proof of $w \Vdash A$.

Similarly, the extension of $\Uparrow$ to a proof that forcing of contexts is monotone can be written as follows, where we reuse the notation $\Uparrow$, with now a context as index, to denote a proof of $\forall w w'\, (w' \geq w \wedge w \Vdash \Gamma \Rightarrow w' \Vdash \Gamma)$:

$$
\begin{array}{llllllll}
\Uparrow_\Gamma^{w,w'} & : & w' \geq w & \wedge & w \Vdash \Gamma & \Rightarrow & w' \Vdash \Gamma \\
\Uparrow_\epsilon^{w,w'} & & (\,h & , & () &) & \triangleq & () \\
\Uparrow_{\Gamma,A}^{w,w'} & & (\,h & , & (\sigma, m) &) & \triangleq & (\Uparrow_\Gamma^{w,w'}\ (h, \sigma), \Uparrow_A^{w,w'}\ (h, m))
\end{array}
$$

Let us now consider the canonical proof $\mathit{soundness}_A^\Gamma$ of $(\Gamma \vdash_I A) \Rightarrow (\Gamma \Vdash_I A)$ proved by reasoning by induction on the derivation of $\Gamma \vdash_I A$. We write the proof as a recursive function:

$$
\begin{array}{llll}
\mathit{soundness}_A^\Gamma & :\ \Gamma \vdash_I A & \Rightarrow & \Gamma \vDash_I A \\
\mathit{soundness}_A^\Gamma & \dot{ax}_i & \triangleq & \mathcal{K} \mapsto w \mapsto \sigma \mapsto \sigma(i) \\
\mathit{soundness}_{A \Rightarrow B}^\Gamma & \dot{abs}(p) & \triangleq & \mathcal{K} \mapsto w \mapsto \sigma \mapsto w' \mapsto (h : w \leq w') \mapsto m \mapsto \mathit{soundness}_B^{\Gamma,A}\ p\, \mathcal{K}\, w'\, (\Uparrow_\Gamma^{w,w'}\ (h, \sigma), m) \\
\mathit{soundness}_B^\Gamma & \dot{app}(p, q) & \triangleq & \mathcal{K} \mapsto w \mapsto \sigma \mapsto (\mathit{soundness}_{A \Rightarrow B}^\Gamma\ p\, \mathcal{K}\, w\, \sigma)\ w\ \mathit{refl}\ (\mathit{soundness}_A^\Gamma\ q\, \mathcal{K}\, w\, \sigma)
\end{array}
$$

where u is a proof of $\Gamma \vdash_I A$ in the last line, $\dot{app}$, $\dot{abs}$, $\dot{ax}_i$ are the name of inference rules defining object-level implicational propositional logic as a natural deduction (see Figure 1), $\sigma(i)$ is the $(i+1)^{\text{th}}$ component of σ starting from the right, and *refl* is the proof of reflexivity of $\geq$ coming with the definition of Kripke models.

[6] Using e.g. Beth-Hintikka-Kanger-Schütte's proof.

Primitive rules

$$\frac{|\Gamma'| = i}{\Gamma, A, \Gamma' \vdash A}\ \dot{ax}_i \qquad \frac{\Gamma \vdash A \dot{\Rightarrow} B \qquad \Gamma \vdash A}{\Gamma \vdash B}\ \dot{app} \qquad \frac{\Gamma, A \vdash B}{\Gamma \vdash A \dot{\Rightarrow} B}\ \dot{abs}$$

Admissible rule

$$\frac{\Gamma \subset \Gamma' \qquad \Gamma \vdash A}{\Gamma' \vdash A}\ \dot{weak}$$

Figure 1: Inference rules characterising minimal implicational logic

Let us also consider the following canonical proof of cut-free completeness *completeness* : $(\Gamma \vDash_I A) \Rightarrow (\Gamma \vdash_I^{cf} A)$ based on the universal model of context $\mathcal{K}_0$ defined by taking for $\mathcal{W}$ the set of contexts Γ ordered by inclusion and by taking $\Gamma \vdash_I^{cf} X$ for the forcing $\Vdash_X$ of atom X at world Γ. It comes from mutually proving the two directions of $\Gamma \Vdash_{\mathcal{K}_0} A \Leftrightarrow \Gamma \vdash_I^{cf} A$ by induction on A. It is common to write $\downarrow$ for the left-to-right direction (called reify, or quote) and $\uparrow$ for the right-to-left direction (called reflect, or eval):

$$\begin{array}{llll}
\downarrow_A^\Gamma \ : & \Gamma \Vdash_{\mathcal{K}_0} A & \Rightarrow & \Gamma \vdash A \\
\downarrow_P^\Gamma & m & \triangleq & m \\
\downarrow_{A \Rightarrow B}^\Gamma & m & \triangleq & \dot{abs}\,(\downarrow_B^{\Gamma,A}\ (m\ (\Gamma, A)\ inj_\Gamma^A\ (\uparrow_A^{\Gamma,A}\ \dot{ax}_0))) \\
 & & & \\
\uparrow_A^\Gamma \ : & \Gamma \vdash A & \Rightarrow & \Gamma \Vdash_{\mathcal{K}_0} A \\
\uparrow_P^\Gamma & p & \triangleq & p \\
\uparrow_{A \Rightarrow B}^\Gamma & p & \triangleq & \Gamma' \mapsto f \mapsto m \mapsto \uparrow_B^{\Gamma'}\ (\dot{app}(\dot{weak}(f, p), \downarrow_A^{\Gamma'}\ m)) \\
 & & & \\
init_{\Gamma'}^\Gamma & & : & \Gamma \Vdash_{\mathcal{K}_0} \Gamma' \\
init_\epsilon^\Gamma & & \triangleq & () \\
init_{\Gamma',A}^\Gamma & & \triangleq & (init_{\Gamma'}^\Gamma, \uparrow_A^\Gamma\ (\dot{ax}_{|\Gamma| - |\Gamma', A|}) \\
 & & & \\
completeness_A^\Gamma \ : & \Gamma \vDash_I A & \Rightarrow & \Gamma \vdash_I A \\
completeness_A^\Gamma & m & \triangleq & \downarrow_A^\Gamma\ (m\, \mathcal{K}_0\, \Gamma\ init_\Gamma^\Gamma)
\end{array}$$

where $|\Gamma|$ is the length of Γ, $\dot{weak}$ is the admissible rule of weakening in object implicational propositional logic and inj_Γ^A is a proof of $\Gamma \subset \Gamma, A$.

In particular, by placing ourselves in a meta-meta-logic, such that the meta-logic is seen as a proofs-as-programs-style natural deduction object language, one would be able to show that

- for every given proof of $\Gamma \vdash_I$, soundness produces, by normalisation in the meta-logic, a proof of $\Gamma \vDash_I A$ whose structure follows the one of the proof of $\Gamma \vdash_I$;
- for every proof of validity taken in canonical form (i.e. as a closed β-normal η-long λ-term of type $\Gamma \vDash A$ in the meta-logic), the resulting proof of $\Gamma \vdash_I^{cf} A$ obtained by completeness is, by normalisation in the meta-logic which we took to be a natural deduction, i.e. a λ-calculus, the same λ-term with the abstractions and applications over $\mathcal{K}$, w and proofs of $w \leq w'$ removed.

That the composition of completeness and soundness performs normalisation can be shown from the meta-logic itself. This is what C. Coquand did by showing that the above proofs of soundness and completeness, seeing the statements as types and the proofs as functions, satisfy the following properties:

$$\forall p : (\Gamma \vdash_I A)\ p \sim soundness_A^\Gamma\ p\, \mathcal{K}_0\, \Gamma\ init_\Gamma^\Gamma$$
$$\forall p : (\Gamma \vdash_I A)\, \forall m : (\Gamma \vDash_I A)\ (p \sim m\, \mathcal{K}_0\, \Gamma\ init_\Gamma^\Gamma \ \Rightarrow\ p =_{\beta\eta} completeness_A^\Gamma\, m)$$

where $\sim$ is an appropriate "Tait computability" relation between object proofs and semantic proofs expressing that *soundness* p reflects p.

Then, since *completeness* returns normal forms, we get that *completeness*(*soundness* p) is a normal form q such that $q =_\beta p$.

Let us conclude this section by saying that the extension of this proof to universal quantification and falsity, using so called exploding nodes, or to classical logic has been studied e.g. in [29]. The case of disjunction and existential quantification is however more difficult ([36, 37] or [1] give a partial answer to it).

One of the purposes of this paper is precisely to start comparatively exploring the computational content of proofs of Gödel's completeness theorem and their ability to provide normalisation. In the case of Henkin's proof, the answer is negative: even if the resulting object proof is related to the proof of validity in the meta-logic, it is neither cut-free nor isomorphic to it. In particular, it drops information from the meta-logic proof by sharing subparts that prove the same subformula as will be emphasised in Section 2.2.

1.7 The intuitionistic provability of the different statements of completeness

Statements C1, C2 and C3, as well as statements S1, S2, S3 are classically equivalent but not intuitionistically equivalent.

Since our object language is only composed of negative connectives, the formula $\mathcal{M} \vDash_\sigma \mathcal{T}$ is itself composed of only negative connectives in the meta-language. Hence, the only positive connective in statements C2 and S2 is the existential quantifier asserting the existence of a model.

This existential quantifier is intuitionistically provable as Henkin's proof of S2 given in the next section shows: given a proof of consistency of a theory, we can define a predicate which happens to be a model of the theory.

It shall however be noted that this predicate is not itself recursive in general, since constructing this model is equivalent in general to producing an infinite path in any arbitrary infinite binary tree (such an infinite path is a priori not recursive, see Kleene [40], Simpson [58]).

From an intuitionistic point of view, statements C1 and S1 are the most interesting ones, as they promise to produce (object) proofs in the object language out of proofs of validity in the meta-logic. However, Kreisel [41] showed, using a result by Gödel [24], that C1 is equivalent to Markov's principle over intuitionistic second order arithmetic. This has been studied in depth by McCarty [51] as a conclusion of which it turns out that S1 is also equivalent to Markov's principle if the theory is recursively enumerable.

However, for theories with arbitrary logical complexity, reasoning by contradiction on formulae of arbitrarily large logical complexity is correspondingly needed as the following adaptation of McCarty's proof shows: Let A be an arbitrary formula and consider e.g. the theory defined by $B \in \mathcal{T} \triangleq (B = \dot{\bot}) \wedge A \vee (B = \dot{\neg}X) \wedge \neg A$. We intuitionistically have that $\mathcal{T} \vDash \neg X$ because this is a negative formulation of a classically provable statement[7]. By completeness, we gets $\mathcal{T} \vdash \neg X$, and, by case analysis on the normal form of the so-obtained proof, one infers that either A or $\neg A$.

The need for Markov's principle is connected to how $\bot$ is interpreted in the model. Krivine [47] showed that for a language without $\bot$[8], C1 is provable intuitionistically. As analysed by Berardi [6] and Berardi and Valentini [7], Markov's principle is not needed anymore if we accept the extra degenerate model where all formulae including $\bot$ are interpreted as true[9]. Let us formalise this precisely.

We define a *possibly-exploding* model $\mathcal{M}$ to be a model $(\mathcal{D}, \mathcal{F}, \mathcal{P}, X)$ such that $(\mathcal{D}, \mathcal{F}, \mathcal{P})$ is a model in the sense previously defined together with the following modified definition of truth of $\bot$:

$$\mathcal{M} \vDash^e_\sigma \dot{\bot} \quad \triangleq \quad X$$

and the rest of clauses unchanged. In [7], a classical possibly-exploding model is called a minimal model.

[7]for $\mathcal{M}$ and σ being given, prove $\mathcal{M} \vDash_\sigma \dot{\neg}X$ by taking $B = \dot{\neg}X$ where the proof that $B \in \mathcal{T}$ needs a proof of $\neg A$. The latter is obtained by assuming A, and using again the hypothesis $\mathcal{M} \vDash_\sigma \mathcal{T}$, but this time with $B = \dot{\bot}$, thanks to A.

[8]so-called minimal classical logic in [3], which is not functionally complete since no formula can then be given the falsified.

[9]This is similar to the approach followed by Friedman [19] and Veldman [61] to intuitionistically prove the completeness of intuitionistic logic with respect to a relaxing of Beth models with so-called fallible models and to a relaxing of Kripke models with so-called exploding nodes, respectively.

Note that because $\bot \dot{\Rightarrow} A$ is a consequence of $\dot{\neg}\dot{\neg}A \dot{\Rightarrow} A$, the following holds for all A and all σ in any classical possibly-exploding model:

$$X \Rightarrow \mathcal{M} \vDash^e_\sigma A$$

So we do not need to enforce it further[10]. Let us rephrase C1 and S1 using classical possibly-exploding models:

$$C1'.\quad \vDash^e A \Rightarrow \vdash A$$
$$S1'.\quad \mathcal{T} \vDash^e A \Rightarrow \mathcal{T} \vdash A$$

where

$$\mathcal{T} \vDash^e A \triangleq \forall \mathcal{M} \forall \sigma (\mathcal{M} \vDash^e_\sigma \mathit{Classic} \Rightarrow \mathcal{M} \vDash^e_\sigma \mathcal{T} \Rightarrow \mathcal{M} \vDash^e_\sigma A)$$

In particular, it is worthwhile to notice that $\mathcal{T} \vDash^e A$ and $\mathcal{T} \vDash A$ are classically equivalent since $\vDash^e$ only differs from $\vDash$ by an extra quantification over the degenerate always-true model. Hence C1 and C1', as well as S1 and S1', are classically equivalent too. But C1' as well as S1' for recursively enumerable theories are provable intuitionistically, while C1 and S1, even for recursively enumerable theories, would require Markov's principle[11].

Let us conclude this section by considering statements C3 and S3. These statements are strongly not intuitionistically provable, as, if they were, provability could be decided. This does not mean however that we cannot compute with C3 and S3. Classical logic is computational (see e.g. [53]), but for an evaluation to be possible, an interaction with a proof of a statement which uses C3 or S3 is needed. We will not explore this further here.

1.8 Related works

We discovered after writing of this paper that the intuitionistic provability of Gödel's completeness theorem with respect to models with exploding nodes was studied independently of Berardi and Valentini by Krivtsov [48]. Krivtsov calls these models intuitionistic structures and he shows that the weak Fan theorem is enough to get an intuitionistic proof for an object language with all connectives [49].

2 The computational content of Henkin's proof of Gödel's completeness

We shall now recall Henkin's proof of completeness and analyse its computational content.

2.1 Henkin's proof of statement S2, slightly simplified

We shall give a simplified form of Henkin's proof of the strong form of Gödel's completeness theorem [26], formulated as statement S2. The simplification is on the use of free variables instead of constants in Henkin axioms and in the use of only implicative formulae in the process of completion of a consistent set of formulae into a maximally consistent one.

Let $\mathcal{T}$ be a consistent set of formulae mentioning an at most countable[12] number of function symbols and predicate symbols. Let $\mathcal{X}_1$ and $\mathcal{X}_2$ be countable sets of variables forming a partition of $\mathcal{X}$. We can assume without loss of generality that the free variables of the formulae in $\mathcal{T}$ are in $\mathcal{X}_1$ leaving $\mathcal{X}_2$ as a pool of variables fresh in $\mathcal{T}$.

[10] As a matter of purity, since it is standard that the classical scheme $\dot{\neg}\dot{\neg}A \dot{\Rightarrow} A$ is equivalent to the conjunction of a purely classical part, namely Peirce's law representing the scheme $((A \dot{\Rightarrow} B) \dot{\Rightarrow} A) \dot{\Rightarrow} B$ and of a purely intuitionistic part, namely ex falso quodlibet representing the scheme $\bot \dot{\Rightarrow} A$, we could have decomposed *Classic* into the disjoint sum of $\mathcal{P}\mathit{eirce} \triangleq \{((A \dot{\Rightarrow} B) \dot{\Rightarrow} A) \dot{\Rightarrow} A \mid A \in \mathcal{F}\mathit{orm}\}$ and of $\mathcal{E}\mathit{xfalso} \triangleq \{\bot \dot{\Rightarrow} A \mid A \in \mathcal{F}\mathit{orm}\}$.

As already said in Section 1.4, the condition $\mathcal{M} \vDash_\sigma \mathit{Classic}$, and in particular the conditions $\mathcal{M} \vDash_\sigma \mathcal{P}\mathit{eirce}$ and $\mathcal{M} \vDash_\sigma \mathcal{E}\mathit{xfalso}$ are needed to show soundness with respect to classical models in a minimal setting. In an intuitionistic setting, $\mathcal{M} \vDash_\sigma \mathcal{E}\mathit{xfalso}$ holds by default and does not have to be explicitly enforced. In a classical setting, $\mathcal{M} \vDash_\sigma \mathcal{P}\mathit{eirce}$ does not have to be explicitly enforced. So, requiring these conditions is to ensure that the definition of validity is the one we want independently of the specific properties of the meta-language.

In contrast, for the purpose of completeness, possible explosion is needed for an intuitionistic proof of C1 to be possible, but none of $\mathcal{P}\mathit{eirce}(\mathcal{M})$ and $\mathcal{E}\mathit{xfalso}(\mathcal{M})$ are required.

[11] Markov's principle can actually be "intuitionistically" implemented e.g. by using an exception mechanism [28], so a computational content to weak completeness and strong completeness for recursively enumerable theories can be obtained without any change in the interpretation of $\bot$.

[12] in the presence of uncountably many symbols, one would need the ultrafilter lemma to well-order formulae and we do not know how to make the proof constructive in this case.

We want to show that $\mathcal{T}$ has a model, and for that purpose, we shall complete it into a consistent set $\mathcal{S}_\omega$ of formulae which is maximal in the sense that if $A \notin \mathcal{S}_\omega$ then $\dot{\neg} A \in \mathcal{S}_\omega$. We shall also ensure that for every universally quantified formula $\dot{\forall} x\, A(x)$, there is a corresponding so-called Henkin axiom $A(y) \dot{\Rightarrow} \dot{\forall} x\, A(x)$ in $\mathcal{S}_\omega$ with y fresh in $\dot{\forall} x\, A(x)$. For the purpose of this construction, we fix an injective enumeration ϕ of formulae of the form $\dot{\forall} x\, A(x)$ or $A \dot{\Rightarrow} B$ and write $\lceil A \rceil$ for the index of a formula A in the enumeration. We also take ϕ so that formulae of even index are of the form $\dot{\forall} x\, A(x)$ and formulae of odd index are of the form $A \dot{\Rightarrow} B$.

Let $\mathcal{S}_0$ be $\mathcal{T}$ and assume that we have already built $\mathcal{S}_n$. If n is even, $\phi(n)$ has the form $\dot{\forall} x\, A(x)$. We then consider a variable $x_{n/2} \in \mathcal{X}_2$ which is fresh in all $\phi(i)$ for $i \leq n$ and we set $\mathcal{S}_{n+1} \triangleq \mathcal{S}_n \cup (A(x_{n/2}) \dot{\Rightarrow} \dot{\forall} x\, A(x))$. Otherwise, $\phi(n)$ is an implicative formula and we consider two cases. If $\mathcal{S}_n \cup \phi(n)$ is consistent, i.e. if $(\mathcal{S}_n \cup \phi(n) \vdash \bot) \Rightarrow \bot$, we set $\mathcal{S}_{n+1} \triangleq \mathcal{S}_n \cup \phi(n)$. Otherwise, we set $\mathcal{S}_{n+1} \triangleq \mathcal{S}_n$. We finally define the predicate $A \in \mathcal{S}_\omega \triangleq \exists n\, (\mathcal{S}_n \vdash A)$, i.e. $\exists n \exists \Gamma \subset \mathcal{S}_n\ (\Gamma \vdash A)$, and this is the base of a syntactic model $\mathcal{M}_0$ defined by taking

$$
\begin{array}{lcl}
\mathcal{D} & \triangleq & \mathcal{T}erm \\
\mathcal{F}(f)(t_1, ..., t_n) & \triangleq & f(t_1, ..., t_n) \\
\mathcal{P}(P)(t_1, ..., t_n) & \triangleq & P(t_1, ..., t_n) \in \mathcal{S}_\omega
\end{array}
$$

By induction, each $\mathcal{S}_n$ is consistent. Indeed, if $\phi(n)$ is implicative and $\mathcal{S}_{n+1} \equiv \mathcal{S}_n \cup \phi(n)$, it is precisely because $\mathcal{S}_{n+1}$ is consistent. Otherwise, the consistency of $\mathcal{S}_{n+1}$ comes from the consistency of $\mathcal{S}_n$. If $\phi(n)$ is some $\dot{\forall} x\, A(x)$, then $\mathcal{S}_{n+1} \equiv \mathcal{S}_n \cup (A(x_{n/2}) \dot{\Rightarrow} \dot{\forall} x\, A(x))$. This is consistent by freshness of $x_{n/2}$ in both $\mathcal{T}$ and in the $\phi(i)$ for $i \leq n$ otherwise. Indeed, because $x_{n/2}$ is fresh, any proof of $\mathcal{S}_n \cup (A(x_{n/2}) \dot{\Rightarrow} \dot{\forall} x\, A(x)) \vdash \bot$ can be turned into a proof of $\mathcal{S}_n \cup \dot{\neg} \dot{\forall} y\, \dot{\neg} (A(y) \dot{\Rightarrow} \dot{\forall} x\, A(x)) \vdash \bot$, which itself can be turned into a proof of $\mathcal{S}_n \vdash \bot$ since $\dot{\neg} \dot{\forall} y\, \dot{\neg}\, (A(y) \dot{\Rightarrow} \dot{\forall} x\, A(x))$ is a classical tautology.

Let id be the identity substitution. We can show by induction on A that $\mathcal{M}_0 \vDash_{id} A \Leftrightarrow A \in \mathcal{S}_\omega$. This is sometimes considered easy because mainly combinatoric but we shall detail the proof because it is here that the computational content of the proof is non-trivial. Moreover, we do not closely follow Henkin's proof who is making strong use of classical reasoning. We shall instead reason intuitionistically, which does not raise any practical difficulty here.

- Let us focus first on the case when A is $B \dot{\Rightarrow} C$. One way to show $B \dot{\Rightarrow} C \in \mathcal{S}_\omega$ from $\mathcal{M}_0 \vDash_{id} B \dot{\Rightarrow} C$ is to show that for n being $\lceil B \dot{\Rightarrow} C \rceil$, the set $\mathcal{S}_n \cup (B \dot{\Rightarrow} C)$ is consistent, i.e. that a contradiction arises from $\mathcal{S}_n \cup (B \dot{\Rightarrow} C) \vdash \bot$. Indeed, from the latter, we get both $\mathcal{S}_n \vdash B$ and $\mathcal{S}_n \vdash \dot{\neg} C$. From $\mathcal{S}_n \vdash B$ we get $\mathcal{M}_0 \vDash_{id} B$ by induction hypothesis, hence $\mathcal{M}_0 \vDash_{id} C$ by assumption on the truth of $B \dot{\Rightarrow} C$. Then $C \in \mathcal{S}_\omega$ again by induction hypothesis, hence $\mathcal{S}_{n'} \vdash C$ for some n'. But also $\mathcal{S}_n \vdash \dot{\neg} C$, hence $\mathcal{S}_{max(n,n')} \vdash \bot$ which contradicts the consistency of $\mathcal{S}_{max(n,n')}$.

- Conversely, if $B \dot{\Rightarrow} C \in \mathcal{S}_\omega$, this means $\mathcal{S}_n \vdash B \dot{\Rightarrow} C$ for some n. To prove $\mathcal{M}_0 \vDash_{id} B \dot{\Rightarrow} C$, let us assume $\mathcal{M}_0 \vDash_{id} B$. By induction hypothesis we get $\mathcal{S}_{n'} \vdash B$ for some n' and hence $\mathcal{S}_{max(n,n')} \vdash C$, i.e. $C \in \mathcal{S}_\omega$. We conclude by induction hypothesis to get $\mathcal{M}_0 \vDash_{id} C$.

- Let us then focus on the case when A is $\dot{\forall} x\, B(x)$. For n even being $\lceil \dot{\forall} x\, B(x) \rceil$, we have the formula $(B(x_{n/2}) \dot{\Rightarrow} \dot{\forall} x\, B(x)) \in \mathcal{S}_{n+1}$. By the induction hypothesis applied on $\mathcal{M}_0 \vDash_{id} B(x_{n/2})$, we also get the existence of some n' such that $\mathcal{S}_{n'} \vdash B(x_{n/2})$. Hence, $\mathcal{S}_{max(n+1,n')} \vdash \dot{\forall} x\, B(x)$, which means $\dot{\forall} x\, B(x) \in \mathcal{S}_\omega$.

- Conversely, assume $\mathcal{S}_n \vdash \dot{\forall} x\, B(x)$ for some n and prove $\mathcal{M}_0 \vDash_{id} \forall x\, B(x)$. Let t be a term. From $\mathcal{S}_n \vdash \dot{\forall} x\, B(x)$ we get $\mathcal{S}_n \vdash B(t)$ and hence $\mathcal{M}_0 \vDash_{id} B(t)$ by induction hypothesis, i.e. $\mathcal{M}_0 \vDash_{(x \mapsto t)} B(x)$.

- Let us then consider the case A is $\bot$. By ex falso quodlibet in the meta-logic, it is direct that $\bot \Rightarrow (\bot \in \mathcal{S}_\omega)$.

- Conversely, let us prove $(\bot \in \mathcal{S}_\omega) \Rightarrow \bot$. From $\bot \in \mathcal{S}_\omega$ we know $\mathcal{S}_n \vdash \bot$ for some n which, again, contradicts the consistency of $\mathcal{S}_n$.

- The case when A is $P(t_1, \ldots, t_n)$ is trivial.

Before completing the proof, it remains to prove that the model is classical. Using the equivalence between $\mathcal{M}_0 \vDash_{id} A$ and $A \in \mathcal{S}_\omega$ it is enough to prove that $\dot{\neg}\dot{\neg} A \in \mathcal{S}_\omega$ implies $A \in \mathcal{S}_\omega$. But the former means $\mathcal{S}_n \vdash \dot{\neg}\dot{\neg} A$ for some n, hence $\mathcal{S}_n \vdash A$ by classical reasoning in the object language, hence $A \in \mathcal{S}_\omega$.

We are now ready to complete the proof. For every $B \in \mathcal{T}$, since $\mathcal{T} \vdash B$, we get $B \in \mathcal{S}_\omega$ and hence $\mathcal{M}_0 \vDash_{id} B$.

In Henkin's proof, the language is extended with constants and Henkin axioms are of the form $\dot{\exists} x\, A(x) \dot{\Rightarrow} A(c)$ with c not occurring in $\dot{\exists} x\, A(x)$. In practice fresh variables are as good as constants, and, since our language does not have the $\dot{\exists}$ quantifier, we also have to use dual axioms $A(x_n) \dot{\Rightarrow} \dot{\forall} x\, A(x)$. In standard presentations of Henkin's proof, including the original proof by Henkin, the completion is made by stages, completing $\mathcal{S}_o$ into $\mathcal{S}_{0\omega}$ so that either $A \notin \mathcal{S}_{0\omega}$ or $\dot{\neg} A \in \mathcal{S}_{0\omega}$ for A defined on the the initial language, then considering a countable set of constant c_{0i} and completing $\mathcal{S}_{0\omega}$ with Henkin axioms to get $\mathcal{S}_{1\omega}$, repeating the process countable many times on a language extended with the constants to get $\mathcal{S}_{\omega}$. One step is however enough as shown e.g. in [58, Th. IV.3.3], and using variables instead of constants is as well enough.

2.2 From Henkin's proof of statement S2 to a proof of statement S1'

Let us fix a formula A_0 and a recursively enumerable theory $\mathcal{T}_0$, i.e. a theory defined by a Σ^0_1-statement. To get a proof of statement S1 for $\mathcal{T}_0$ and A_0 is easy by using Markov's principle: to prove $\mathcal{T}_0 \vdash A_0$ from $\mathcal{T}_0 \vDash^e A_0$, let us assume the contrary, namely that $\mathcal{T} \triangleq \mathcal{T}_0 \cup \dot{\neg} A_0$ is inconsistent. Then, we can complete $\mathcal{T}$ into $\mathcal{S}_\omega$ and build out of it a classical model $\mathcal{M}_0$ such that $\forall B \in \mathcal{T}_0\; \mathcal{M}_0 \vDash^e_{id} B$ as well as $\mathcal{M}_0 \vDash^e_{id} \dot{\neg} A_0$, i.e. $\neg(\mathcal{M}_0 \vDash^e_{id} A_0)$. But this contradicts $\mathcal{T}_0 \vDash^e A_0$ and, because $\mathcal{T}_0$ is Σ^0_1, hence $\mathcal{T}_0 \vdash A$ as well, Markov's principle applies.

As discussed in Section 1.7, S1 cannot be proved without Markov's principle, so we shall instead prove S1'. To turn the proof of C1 into a proof of S1' which does not require reasoning by contradiction, we shall slightly change the construction of $\mathcal{S}_\omega$ from $\mathcal{T}$ so that it is not consistent in an absolute sense, but instead consistent relative to $\mathcal{T}$. In particular, we change the condition for extending $\mathcal{S}_{2n+2}$ with $\phi(2n+1)$ to be that $\mathcal{S}_{2n+1} \cup \phi(2n+1)$ is *consistent relative* to $\mathcal{T}$.

Then, we show by induction not that $\mathcal{S}_n$ is consistent but that its inconsistency reduces to the inconsistency of $\mathcal{T}$.

For the construction of the now possibly-exploding model, we take as interpretation of $\bot$ the formula $\mathcal{T} \vdash \bot$. Proving $\bot \in \mathcal{S}_\omega \Rightarrow \mathcal{M}_0 \vDash^e_{id} \bot$ now reduces to proving $\mathcal{S}_n \vdash \bot \Rightarrow \mathcal{T} \vdash \bot$ which is the statement of relative consistency[13].

The change in the definition of $\mathcal{S}_\omega$ as well as the use of possibly-exploding models is directly connected to using Friedman's A-translation [20] to absorb the need for Markov's principle. Here, A is the Σ^0_1-formula $\mathcal{T}_0 \vdash A_0$ and by replacing $\bot$ by A in the definition of model, hence of validity, as well as in the definition of $\mathcal{S}_\omega$, we are able to prove $(A \Rightarrow A) \Rightarrow A$ whereas only $(A \Rightarrow \bot) \Rightarrow \bot$ was provable. Then, A comes trivially from $(A \Rightarrow A) \Rightarrow A$.

This was the idea followed by Krivine [47] in his constructive proof of Gödel's theorem for a language restricted to $\dot{\Rightarrow}$ and $\dot{\forall}$, as analysed and clarified in Berardi and Valentini [7].

As a final remark, one could wonder whether the construction of $\mathcal{S}_{2n+2}$ by case on an undecidable statement is compatible with intuitionistic reasoning. Indeed, constructing the sequence of formulae added to $\mathcal{T}$ in order to get $\mathcal{S}_n$ seems to require a use of excluded-middle. However, in the proof of completeness, only the property $A \in \mathcal{S}_n$ matters, and this property is directly definable by induction as

$$
\begin{array}{lcl}
A \in \mathcal{S}_0 & \triangleq & A \in \mathcal{T} \\
A \in \mathcal{S}_{n+1} & \triangleq & A \in \mathcal{S}_n \\
 & & \vee\ (\exists p\,(n = 2p+1)\ \wedge\ \phi(n) = \dot{\forall} x\, B(x)\ \wedge\ A = (B(x_{p+1}) \dot{\Rightarrow} \dot{\forall} x\, B(x)) \\
 & & \vee\ (\exists p\,(n = 2p)\ \wedge\ (\mathcal{S}_n, A \vdash \bot \Rightarrow \mathcal{T} \vdash \bot)\ \wedge\ A = \phi(n))
\end{array}
$$

Note however that $\mathcal{S}_n$ is used in negative position of an implication in the definition of $\mathcal{S}_{n+1}$. Hence, the complexity of the formula $A \in \mathcal{S}_n$ seen as a type of functions is a type of higher-order functions of depth n.

We will represent proofs in this language using a λ-calculus with pairs. In particular, we use the same notation for the construction of proofs of universally quantified formulae and for the construction of proofs of implicative formulae, as it is done in type theory where both are a particular case of building an object in a dependent product: our notation is $x \mapsto p$. We also identify the construction of existentially quantified formulae and the construction of proofs of conjunctive formulae as both are a particular case of dependent sums (also known as Σ-types). For the proofs in a n-ary combined existential and conjunction connective, our notation is $(p_1, ..., p_n)$.

[13] Interestingly enough, since $\mathcal{T}_0 \vdash A$ indeed holds a proof of validity of A being given, the model we built is then the degenerate one in which all formulae are true.

Modus Ponens and universal quantification elimination are written as a concatenation while elimination from n-ary combined existential and conjunction is written *dest p as* $(x_1, ..., x_n)$ *in q*.

2.3 The computational content of the proof of completeness

We are now ready to formulate the proof as a program. We shall place ourselves in an axiom-free second-order intuitionistic arithmetic equipped with a proof-as-program interpretation[14], as already considered in Section 1.6. Additionally, we shall identify the construction of existentially quantified formulae and the construction of proofs of conjunctive formulae. For instance, we shall use the $(p_1, ..., p_n)$ for the proof of a n-ary combined existential and conjunction connective. We shall also write *dest p as* $(x_1, ..., x_n)$ *in q* for a proof obtained by decomposition of the proof p of n-ary combined existential and conjunction. We shall write *efq p* for a proof of A from a proof of p of $\bot$ (ex falso quodlibet).

We shall use the letters $n, A, \Gamma, m, p, h, g, f, k$ and their variants to refer to natural numbers, formulae, contexts of formulae, proofs of truth, proofs of derivability in the object language, proofs of inclusion in $\mathcal{T}_0$, proofs of belonging to $\mathcal{T}_0$, proofs of inclusion in extensions of $\mathcal{T}_0$, proofs of relative consistency, respectively.

The key property is $A \in \mathcal{S}_\omega$ which unfolds as $\exists n\, \exists \Gamma\, (\Gamma \subset \mathcal{S}_n \wedge \Gamma \vdash A)$. Rather than defining $\Gamma \subset \mathcal{S}_n$ from $A \in \mathcal{S}_n$ and the latter by cases, we directly define $\Gamma \subset \mathcal{S}_n$ by cases as our primitive concept. Now, for the formula $\phi(n)$ to be in $\mathcal{S}_{n+1}$, a proof that the possible inconsistency of $\mathcal{S}_{n+1}$ reduces to the inconsistency already of $\mathcal{T}_0 \cup \dot{\neg} A_0$ is required. Otherwise said, the property $\Gamma \subset \mathcal{S}_n$ is a collection of proofs expressing, for each $B \in \Gamma$, that either $B \in \mathcal{T}_0$ (clause J_{cons}), or B is $\dot{\neg} A_0$, or B is an Henkin axiom (clause $I_{\dot\forall}$), or B is an implication together with a proof of relative consistency of Γ with respect to $\mathcal{T}_0 \cup \dot{\neg} A_0$ (clause $I_{\dot\Rightarrow}$).

We first define by cases the predicate $\Gamma \subset \mathcal{T}_0$:

$$\frac{}{\epsilon \subset \mathcal{T}_0}\, J_{base} \qquad\qquad \frac{\Gamma \subset \mathcal{T}_0 \qquad A \in \mathcal{T}_0}{\Gamma, A \subset \mathcal{T}_0}\, J_{cons}$$

Defining $\dot{\neg} A_0, \mathcal{T}_0 \vdash \bot$ to be $\exists \Gamma\, (\Gamma \subset \mathcal{T}_0 \;\wedge\; \neg A_0, \Gamma' \vdash \bot)$, we shall now define by cases $\Gamma \subset \mathcal{S}_n$:

$$\frac{\Gamma \subset \mathcal{T}_0}{\dot{\neg} A_0, \Gamma \subset \mathcal{S}_0}\, I_0 \qquad \frac{\Gamma \subset \mathcal{S}_n}{\Gamma \subset \mathcal{S}_{n+1}}\, I_S \qquad \frac{\Gamma \subset \mathcal{S}_{2n}}{\Gamma,\; A(x_n) \dot\Rightarrow \dot\forall x\, A(x) \;\subset\; \mathcal{S}_{2n+1}}\, I_{\dot\forall}$$

$$\frac{\Gamma \subset \mathcal{S}_{2n+1} \qquad \exists \Gamma'\, (\Gamma' \subset \mathcal{S}_{2n+1} \;\wedge\; \Gamma', A \dot\Rightarrow B \vdash \bot) \Rightarrow \dot{\neg} A_0, \mathcal{T}_0 \vdash \bot}{\Gamma,\; A \dot\Rightarrow B \;\subset\; \mathcal{S}_{2n+2}}\, I_{\dot\Rightarrow}$$

where $\phi(2n)$ is $\dot\forall x\, A(x)$ in $I_{\dot\forall}$ and $\phi(2n+1)$ is $A \dot\Rightarrow B$ in $I_{\dot\Rightarrow}$.

We now write as a program the proof that $\mathcal{S}_\omega$ is consistent relative to $\mathcal{T}_0 \cup \dot{\neg} A_0$, i.e. that for all n and Γ such that $\Gamma \subset S_n$ and $\Gamma \vdash \bot$, then already $\dot{\neg} A_0, \mathcal{T}_0 \vdash \bot$.

Computationally, it happens to work, for n odd, by calling the *continuation* justifying that adding the formula $\phi(2p+1)$ preserves consistency, and, for n even, to compose the resulting proof of inconsistency with a proof of the Drinker's paradox ($\dot{drinker}_y$ is the proof which builds a proof of $\Gamma \vdash \bot$ from a proof of $\Gamma, A(y) \dot\Rightarrow \dot\forall x\, A(x) \vdash \bot$, knowing that y does not occur in $\Gamma, \dot\forall x\, A(x)$, see Figure 2).

$$\begin{array}{llllllll}
\mathit{flush}_n^{\Gamma} & : & \Gamma \subset \mathcal{S}_n & \wedge & \Gamma \vdash \bot & & \Rightarrow & \dot{\neg} A_0, \mathcal{T}_0 \vdash \bot \\
\mathit{flush}_0^{\dot{\neg} A_0, \Gamma} & & (\; I_0\, g & , & p &) & \triangleq & (\Gamma, g, p) \\
\mathit{flush}_{n+1}^{\Gamma} & & (\; I_S\, f & , & p &) & \triangleq & \mathit{flush}_n^{\Gamma}(f, p) \\
\mathit{flush}_{2n+1}^{\Gamma, A} & & (\; I_{\dot\forall}\, f & , & p &) & \triangleq & \mathit{flush}_{2n}^{\Gamma}(f, \dot{drinker}_{x_n}\, p) \\
\mathit{flush}_{2n+2}^{\Gamma, A} & & (\; I_{\dot\Rightarrow}\, (f, k) & , & p &) & \triangleq & k\, (\Gamma, f, p)
\end{array}$$

[14] A typical effective framework for that purpose would be a fragment of the Calculus of Inductive Constructions such as it is implemented in the Coq proof assistant [12], or Matita [4]. The Calculus of Inductive Constructions is an impredicative extension of Martin-Löf's type theory [50].

A boring lemma which is implicit in the proof of completeness in natural language is that $\Gamma \subset \mathcal{S}_n$ and $\Gamma' \subset \mathcal{S}_{n'}$ imply $\Gamma \cup \Gamma' \subset \mathcal{S}_{max(n,n')}$. It looks obvious because one tends to think of $\Gamma \subset \mathcal{S}_n$ as denoting the inclusion of Γ within a uniquely defined relatively consistent set $\mathcal{S}_n$. However, the computational approach to the proof shows that $\mathcal{S}_n$ has no computational content: only proofs of $\Gamma \subset \mathcal{S}_n$ have, and such proofs are collections of proofs of relative consistency for only those implicative formulae which are in Γ. These formulae are those inspected by the lemma $A \in \mathcal{S}_\omega \Leftrightarrow \mathcal{M}_0 \vDash^e_\sigma A$, which in practice are subformulae of the formulae in $\mathcal{T}_0$.

Contexts being defined as sequences of formulae, we first need to define $\Gamma \cup \Gamma'$ so that it yields a context of formulae ordered in a way compatible with the enumeration:

$$
\begin{array}{lllll}
\epsilon & \cup\ \epsilon & ::= & \epsilon & \\
\epsilon & \cup\ (\Gamma', A) & ::= & \Gamma', A & \\
(\Gamma, A) & \cup\ \epsilon & ::= & \Gamma, A & \\
(\Gamma, A) & \cup\ (\Gamma', A) & ::= & (\Gamma \cup \Gamma'), A & \\
(\Gamma, A) & \cup\ (\Gamma', B) & ::= & (\Gamma, A \cup \Gamma'), B & \text{if } \lceil B \rceil > \lceil A \rceil \\
(\Gamma, A) & \cup\ (\Gamma', B) & ::= & (\Gamma \cup \Gamma', B), A & \text{if } \lceil A \rceil > \lceil B \rceil
\end{array}
$$

We can then define straightforwardly the merge of two proofs of $\Gamma \subset \mathcal{S}_n$:

$$
\begin{array}{llllllll}
join_\subset^{\Gamma_1\Gamma_2} & : & \Gamma_1 \subset \mathcal{T}_0 & \wedge & \Gamma_2 \subset \mathcal{T}_0 & \Rightarrow & \Gamma_1 \cup \Gamma_2 \subset \mathcal{T}_0 & \\
join_\subset^{\epsilon\epsilon} & & (\ J_{base} & , & J_{base} \ ) & \triangleq & J_{base} & \\
join_\subset^{(\Gamma_1,A)(\epsilon)} & & (\ J_{cons}(g_1,h_1) & , & J_{base} \ ) & \triangleq & J_{cons}(g_1,h_1) & \\
join_\subset^{(\epsilon)(\Gamma_2,A)} & & (\ J_{base} & , & J_{cons}(g_2,h_2) \ ) & \triangleq & J_{cons}(g_2,h_2) & \\
join_\subset^{(\Gamma_1,A)(\Gamma_2,A)} & & (\ J_{cons}(g_1,h_1) & , & J_{cons}(g_2,h_2) \ ) & \triangleq & J_{cons}(join_\subset^{\Gamma_1\Gamma_2}(g_1,g_2),h_1) & \\
join_\subset^{(\Gamma_1,A_1)(\Gamma_2,A_2)} & & (\ f_1 & , & J_{cons}(f_2,h_2) \ ) & \triangleq & J_{cons}(join_\subset^{(\Gamma_1,A_1)(\Gamma_2)}(f_1,f_2),h_2) & \text{if } \lceil A_2 \rceil > \lceil A_1 \rceil \\
join_\subset^{(\Gamma_1,A_1)(\Gamma_2,A_2)} & & (\ J_{cons}(f_1,h_1) & , & f_2 \ ) & \triangleq & J_{cons}(join_\subset^{(\Gamma_1)(\Gamma_2,A_2)}(f_1,f_2),h_1) & \text{if } \lceil A_2 \rceil < \lceil A_1 \rceil
\end{array}
$$

$$
\begin{array}{lllllll}
join_n^{\Gamma_1\Gamma_2} & : & \Gamma_1 \subset \mathcal{S}_n & \wedge & \Gamma_2 \subset \mathcal{S}_n & \Rightarrow & \Gamma_1 \cup \Gamma_2 \subset \mathcal{S}_n \\
join_0^{(\dot\neg A_0,\Gamma_1)(\dot\neg A_0,\Gamma_2)} & & (\ I_0(f_1) & , & I_0(f_2) \ ) & \triangleq & I_0(join_\subset^{\Gamma_1\Gamma_2}(f_1,f_2)) \\
join_{2n+1}^{(\Gamma_1 A)(\Gamma_2 A)} & & (\ \mathtt{I}_\forall(f_1) & , & \mathtt{I}_\forall(f_2) \ ) & \triangleq & \mathtt{I}_\forall(join_{2n}^{\Gamma_1\Gamma_2}(f_1,f_2)) \\
join_{2n+2}^{(\Gamma_1 A)(\Gamma_2 A)} & & (\ \mathtt{I}_\Rightarrow(f_1,k_1) & , & \mathtt{I}_\Rightarrow(f_2,k_2) \ ) & \triangleq & \mathtt{I}_\Rightarrow(join_{2n+1}^{\Gamma_1\Gamma_2}(f_1,f_2),k_1) \\
join_{2n+1}^{(\Gamma_1 A)\Gamma_2} & & (\ \mathtt{I}_\forall(f_1) & , & I_S(f_2) \ ) & \triangleq & \mathtt{I}_\forall(join_{2n}^{\Gamma_1\Gamma_2}(f_1,f_2)) \\
join_{2n+2}^{(\Gamma_1 A)\Gamma_2} & & (\ \mathtt{I}_\Rightarrow(f_1,k_1) & , & I_S(f_2) \ ) & \triangleq & \mathtt{I}_\Rightarrow(join_{2n+1}^{\Gamma_1\Gamma_2}(f_1,f_2),k_1) \\
join_{2n+1}^{\Gamma_1(\Gamma_2 A)} & & (\ I_S(f_1) & , & \mathtt{I}_\forall(f_2) \ ) & \triangleq & \mathtt{I}_\forall(join_{2n}^{\Gamma_1\Gamma_2}(f_1,f_2)) \\
join_{2n+2}^{\Gamma_1(\Gamma_2 A)} & & (\ I_S(f_1) & , & \mathtt{I}_\Rightarrow(f_2,k_2) \ ) & \triangleq & \mathtt{I}_\Rightarrow(join_{2n+1}^{\Gamma_1\Gamma_2}(f_1,f_2),k_2) \\
join_{n+1}^{\Gamma_1\Gamma_2} & & (\ I_S(f_1) & , & I_S(f_2) \ ) & \triangleq & I_S(join_n^{\Gamma_1\Gamma_2}(f_1,f_2))
\end{array}
$$

$$
\begin{array}{llllllll}
merge_{n_1n_2}^{\Gamma_1\Gamma_2} & : & \Gamma_1 \subset \mathcal{S}_{n_1} & \wedge & \Gamma_2 \subset \mathcal{S}_{n_2} & \Rightarrow & \Gamma_1 \cup \Gamma_2 \subset \mathcal{S}_{max(n_1,n_2)} & \\
merge_{n_1n_2}^{\Gamma_1\Gamma_2} & & (\ f_1 & , & f_2 \ ) & \triangleq & join_n^{\Gamma_1\Gamma_2}(f_1,f_2) & \text{if } n = n_1 = n_2 \\
merge_{n_1n_2}^{\Gamma_1\Gamma_2} & & (\ I_S(f_1) & , & f_2 \ ) & \triangleq & I_S(merge_{n'_1n_2}^{\Gamma_1\Gamma_2}(f_1,f_2)) & \text{if } n_1 = n'_1 + 1 > n_2 \\
merge_{n_1n_2}^{(\Gamma_1A_1)\Gamma_2} & & (\ I_\forall(f_1,k_1) & , & f_2 \ ) & \triangleq & I_\forall(merge_{n'_1n_2}^{\Gamma_1\Gamma_2}(f_1,f_2),k_1) & \text{if } n_1 = n'_1 + 1 > n_2 \\
merge_{n_1n_2}^{(\Gamma_1A_1)\Gamma_2} & & (\ I_\Rightarrow(f_1,k_1) & , & f_2 \ ) & \triangleq & I_\Rightarrow(merge_{n'_1n_2}^{\Gamma_1\Gamma_2}(f_1,f_2),k_1) & \text{if } n_1 = n'_1 + 1 > n_2 \\
merge_{n_1n_2}^{\Gamma_1\Gamma_2} & & (\ f_1 & , & I_S(f_2) \ ) & \triangleq & I_S(merge_{n_1n'_2}^{\Gamma_1\Gamma_2}(f_1,f_2)) & \text{if } n_1 < n'_2 + 1 = n_2 \\
merge_{n_1n_2}^{\Gamma_1(\Gamma_2A_2)} & & (\ f_1 & , & \mathtt{I}_\forall(f_2) \ ) & \triangleq & \mathtt{I}_\forall(merge_{n_1n'_2}^{\Gamma_1\Gamma_2}(f_1,f_2)) & \text{if } n_1 < n'_2 + 1 = n_2 \\
merge_{n_1n_2}^{\Gamma_1(\Gamma_2A_2)} & & (\ f_1 & , & \mathtt{I}_\Rightarrow(f_2,k_2) \ ) & \triangleq & \mathtt{I}_\Rightarrow(merge_{n_1n'_2}^{\Gamma_1\Gamma_2}(f_1,f_2),k_2) & \text{if } n_1 < n'_2 + 1 = n_2
\end{array}
$$

In particular, it has to be noticed that the merge possibly does arbitrary choices: when the same formula occurs in both contexts, i.e. when the same formula occurs in both contexts with a constructor of object proofs telling how to reduce $\Gamma, A \vdash \bot$ to $\exists\Gamma(\Gamma \subset \mathcal{T}_0 \wedge \dot\neg A_0, \Gamma \vdash \bot)$ (third clause of $join_n$), only one of these two constructors of object proofs is (arbitrarily) kept.

Primitive rules

$$\frac{|\Gamma'| = i}{\Gamma, A, \Gamma' \vdash A}\ \dot{ax}_i \qquad \frac{\Gamma \vdash \dot{\neg}\dot{\neg} A}{\Gamma \vdash A}\ \dot{dn}$$

$$\frac{\Gamma \vdash A \dot{\Rightarrow} B \qquad \Gamma' \vdash A}{\Gamma \cup \Gamma' \vdash B}\ \dot{app}^{\dot{\Rightarrow}} \qquad \frac{\Gamma \vdash \dot{\forall} x\, A(x)}{\Gamma \vdash A(t)}\ \dot{app}^{\dot{\forall}}{}_t$$

$$\frac{\Gamma, A \vdash B}{\Gamma \vdash A \dot{\Rightarrow} B}\ \dot{abs}^{\dot{\Rightarrow}} \qquad \frac{\Gamma \vdash A(y) \qquad y \text{ not in } \dot{\forall} x\, A(x), \Gamma}{\Gamma \vdash \dot{\forall} x\, A(x)}\ \dot{abs}^{\dot{\forall}}$$

Admissible rules

$$\frac{\Gamma, A(y) \dot{\Rightarrow} \dot{\forall} x\, A(x) \vdash \dot{\bot} \qquad y \text{ not in } \dot{\forall} x\, A(x), \Gamma}{\Gamma \vdash \dot{\bot}}\ \dot{drinker}_y$$

$$\frac{\Gamma, A \dot{\Rightarrow} B \vdash \dot{\bot}}{\Gamma \vdash A}\ \pi_1^{\dot{\Rightarrow}} \qquad \frac{\Gamma, A \dot{\Rightarrow} B \vdash \dot{\bot}}{\Gamma \vdash \dot{\neg} B}\ \pi_2^{\dot{\Rightarrow}} \qquad \frac{\Gamma \vdash \dot{\bot}}{\Gamma \vdash A}\ efq$$

Figure 2: Inference rules characterising classical first-order predicate calculus

Another lemma is also implicit in the proof of completeness in natural language: it simply says that $\dot{\neg} A_0 \subset \mathcal{S}_n$ and it is proved by induction on n:

$$\begin{array}{lcl} inj_n & : & \dot{\neg} A_0 \subset \mathcal{S}_n \\ inj_0 & \triangleq & I_0(J_{base}) \\ inj_{n+1} & \triangleq & I_S(inj_n) \end{array}$$

We are now ready to present the main computational piece of the completeness proof and we shall use for that notations reminiscent from semantic normalisation [13], or type-directed partial evaluation [15], as considered when proving completeness of intuitionistic logic with respect to models such a Kripke or Beth models.

We have to prove $\mathcal{M}_0 \vDash^e_{id} A \Leftrightarrow A \in \mathcal{S}_\omega$, which means proving $\mathcal{M}_0 \vDash^e_{id} A \Rightarrow A \in \mathcal{S}_\omega$ and $A \in \mathcal{S}_\omega \Rightarrow \mathcal{M}_0 \vDash^e_{id} A$. As in semantic normalisation (see Section 1.6), we shall call reification and write $\downarrow_A$ the proof mapping a semantic formula (i.e. $\mathcal{M}_0 \vDash^e_{id} A$) to a syntactic formula, i.e. $A \in \mathcal{S}_\omega$. We shall call reflection and write $\uparrow_A$ for the way up going from the syntactic view to the semantic view.

Our object logic is defined by the rules on Figure 2. Note that we shall use non standard derived rules. For instance, we shall not use the rule $\dot{abs}^{\dot{\Rightarrow}}$ and $\dot{abs}^{\dot{\forall}}$ but instead the derived rules $\pi_1^{\dot{\Rightarrow}}$, $\pi_2^{\dot{\Rightarrow}}$ and $\dot{drinker}_y$.

$$
\begin{array}{llcl}
\downarrow_A \; : & \mathcal{M}_0 \models^e_{id} A & \rightarrow & A \in \mathcal{S}_\omega \\
\downarrow_{P(\vec{t})} & m & \triangleq & m \\
\downarrow_\bot & m & \triangleq & \mathit{efq}\, m \\
\downarrow_{A \dot\Rightarrow B} & m & \triangleq & (\,2n+1, (\dot\neg A_0, A \dot\Rightarrow B), \\
 & & & I_\Rightarrow(\mathit{inj}_{2n+1}, (\Gamma, f, p) \mapsto \begin{array}{l} \mathit{dest}\ \downarrow_B (m(\uparrow_A (2n+1, \Gamma, f, \pi_1^{\dot\Rightarrow}\, p)))\, \mathit{as}\, (n', \Gamma', f', p') \\ \mathit{in}\ \mathit{flush}^{\Gamma\cup\Gamma'}_{\mathit{max}(2n+1,n')}(\mathit{merge}^{\Gamma\Gamma'}_{(2n+1)n'}(f, f'), \dot{\mathit{app}}^{\Rightarrow}(\pi_2^{\dot\Rightarrow}\, p, p')) \end{array}), \\
 & & & \dot{a}x_0) \qquad \mathit{where}\ 2n+1 = \ulcorner A \dot\Rightarrow B \urcorner \\
\downarrow_{\dot\forall x\, A(x)} & m & \triangleq & \mathit{dest}\ \downarrow_{A(x_n)} (m\, x_n)\, \mathit{as}\, (n', \Gamma', f', p') \\
 & & & \mathit{in}\ (\mathit{max}(2n+1, n'), \Gamma', \mathit{merge}^{(\dot\neg A_0)\Gamma'}_{(2n+1)n'}(\mathtt{I}_\forall(\mathit{inj}_{2n}), f'), \dot{\mathit{app}}^{\Rightarrow}(\dot{a}x_0, p')) \\
 & & & \mathit{where}\ 2n = \ulcorner \dot\forall x\, A \urcorner
\end{array}
$$

$$
\begin{array}{llcl}
\uparrow_A \; : & A \in \mathcal{S}_\omega & \rightarrow & \mathcal{M}_0 \models^e_{id} A \\
\uparrow_{P(\vec{t})} & (n, \Gamma, f, p) & \triangleq & (n, \Gamma, f, p) \\
\uparrow_\bot & (n, \Gamma, f, p) & \triangleq & \mathit{flush}^\Gamma_n(f, p) \\
\uparrow_{A \dot\Rightarrow B} & (n, \Gamma, f, p) & \triangleq & m \mapsto \begin{array}{l} \mathit{dest}\ \downarrow_A m\, \mathit{as}\, (n', \Gamma', f', p') \\ \mathit{in}\ \uparrow_B (\mathit{max}(n, n'), \Gamma \cup \Gamma', \mathit{merge}^{\Gamma\Gamma'}_{nn'}(f, f'), \dot{\mathit{app}}^{\Rightarrow}(p, p')) \end{array} \\
\uparrow_{\dot\forall x\, A(x)} & (n, \Gamma, f, p) & \triangleq & t \mapsto \uparrow_{A(t)} (n, \Gamma, f, \dot{\mathit{app}}^{\forall}(p, t))
\end{array}
$$

We still have to prove that the model is classical:

$$
\begin{array}{llcl}
\mathit{classic}^A_0 & : \ \mathcal{M}_0 \models^e_{id} (\dot\neg\dot\neg A) & \Rightarrow & \mathcal{M}_0 \models^e_{id} A \\
\mathit{classic}^A_0 & \quad m & \triangleq & \uparrow_A (\mathit{dest}\ \downarrow_{\dot\neg\dot\neg A} m\, \mathit{as}\, (n, \Gamma, f, p)\, \mathit{in}\, (n, \Gamma, f, \dot{d}n p))
\end{array}
$$

and we write $\mathit{classic}_0 \triangleq A \mapsto \mathit{classic}^A_0$.

It still remains to show that every formula of $\mathcal{T}_0$ is true in $\mathcal{M}_0$:

$$
\begin{array}{llcl}
\mathit{init}_B & : \ B \in \mathcal{T}_0 & \Rightarrow & \mathcal{M}_0 \models^e_{id} B \\
\mathit{init}_B & \quad h & \triangleq & \uparrow_B (I_0(J_{cons}(J_{base}, h))))
\end{array}
$$

and we then write $\mathit{init} \triangleq B \mapsto \mathit{init}_B$. Finally, we get the completeness result stated as S2 by:

$$
\begin{array}{lcl}
\mathit{completeness} : \forall \mathcal{M} \forall \sigma\, (\mathcal{M} \models^e_\sigma \mathit{Classic} \Rightarrow \mathcal{M} \models^e_\sigma \mathcal{T}_0 \Rightarrow \mathcal{M} \models^e_\sigma A_0) & \Rightarrow & \mathcal{T}_0 \vdash A_0 \\
\mathit{completeness} \quad \psi \quad \triangleq & & \begin{array}{l} \mathit{dest}\ \downarrow_{A_0} (\psi\, \mathcal{M}_0\, \mathtt{id}\, \mathit{classic}_0\, \mathit{init})\, \mathit{as}\, (n, \Gamma, f, p) \\ \mathit{in}\ \mathit{dest}\, \mathit{flush}^\Gamma_n(f, \dot{\mathit{app}}^{\Rightarrow}(\dot{a}x_{|\Gamma|-1}, p))\, \mathit{as}\, (\Gamma', g, p')\, \mathit{in}\, (\Gamma', g, \dot{d}n(\dot{\mathit{abs}}^{\Rightarrow}(p'))) \end{array}
\end{array}
$$

where $|\Gamma|$ is the length of Γ which necessarily contains $\dot\neg A_0$ in first position. Notice that the result is a triple (Γ', g, q) such that q is a proof of $\Gamma' \vdash A$ and g is a proof of $\Gamma \subset \mathcal{T}_0$.

2.4 The computational content on examples

To illustrate the behaviour of the completeness proofs, we look at its behaviour on two examples. We use notations of λ-calculus to represent proofs in the meta-logic and constructors from Figure 2 for proofs in the object logic.

Let us consider $A_0 \triangleq X \Rightarrow Y \Rightarrow X$ with X and Y propositional atoms. There is a canonical proof of $\models^e A_0$, which, as a λ-term, is the K combinator.

The expansion of $\models^e A_0$ is $\forall \mathcal{M}\, \forall \sigma\, (\sigma \models^e_{\mathcal{M}} \mathit{Classic} \Rightarrow \mathcal{M} \models^e X \Rightarrow \mathcal{M} \models^e Y \Rightarrow \mathcal{M} \models^e X)$ and its canonical proof is:

$$m \triangleq (\mathcal{D}, \mathcal{F}, \mathcal{P}, B) \mapsto \sigma \mapsto c \mapsto (x : \mathcal{P}(X)) \mapsto (y : \mathcal{P}(Y)) \mapsto x$$

Applying completeness means instantiating the model by the syntactic model and the empty substitution so as to obtain from m the proof $m_0 \triangleq (x : X \in \mathcal{S}_\omega) \mapsto (y : Y \in \mathcal{S}_\omega) \mapsto x$.

Our object proof is then the result of evaluating

$$\dot{d}n(\dot{\mathit{abs}}^{\Rightarrow}(\mathit{dest}\ \downarrow_{A_0} m_0\, \mathit{as}\, (n, \Gamma, f, p)\, \mathit{in}\, \mathit{flush}^\Gamma_n(f, \dot{\mathit{app}}^{\Rightarrow}(\dot{a}x_{|\Gamma|-1}, p))))$$

Evaluating $\downarrow_{A_0}\ m_0$ gives a tuple $v_0 \triangleq (2n_0+1, (\dot\neg A_0, A_0), I_{\Rightarrow}(inj_{2n_0+1}, k_0), \dot{ax}_0)$ where $2n_0+1$ is $\lceil A_0 \rceil$ and k_0 the following continuation mapping a proof of $\Gamma', \dot\neg A_0 \vdash \bot$ for some $\Gamma' \subset S_{2n_0+1}$ to a proof of $\exists \Gamma \subset (\dot\neg A_0)\,(\Gamma \vdash \bot)$, as given in clause $\Rightarrow$ of $\downarrow$:

$$k_0(\Gamma, f, p) \triangleq \phi(2n_0+1, \Gamma, f, \pi_2^{\dot\Rightarrow}\ p, \downarrow_{Y\dot\Rightarrow X}\ (m_0(\uparrow_X\ (2n_0+1, \Gamma, f, \pi_1^{\dot\Rightarrow}\ p))))$$

where

$$\phi(n, \Gamma, f, p, m) \triangleq dest\, m\, as\, (n', \Gamma', f', p')\, in\, flush^{\Gamma\cup\Gamma'}_{max(n,n')}(merge^{\Gamma\Gamma'}_{nn'}(f, f'), \dot{app}^{\Rightarrow}(p, p'))$$

Our object proof is then now the result of evaluating

$$\dot{dn}(\dot{abs}^{\Rightarrow}(flush^{\dot\neg A_0, A_0}_{2n_0+1}(I_{\Rightarrow}(inj_{2n_0+1}, k_0), p_0)))$$

where $p_0 \triangleq \dot{app}^{\Rightarrow}(\dot{ax}_1, \dot{ax}_0)$ is a proof of $\dot\neg A_0, A_0 \vdash \bot$ obtained by application of the two axiom rules proving $\dot\neg A_0, A_0 \vdash \dot\neg A_0$ and $\dot\neg A_0, A_0 \vdash A_0$.

Evaluating *flush* forces the continuation k_0 to be applied to p_0 resulting in:

$$\dot{dn}(\dot{abs}^{\Rightarrow}(k_0(I_{\Rightarrow}(inj_{2n_0+1}, k_0), p_0)))$$

The proof $\uparrow_X\ (2n_0+1, (\dot\neg A_0), inj_{2n_0+1}, p_0)$ reflects the object proof $\pi_1^{\dot\Rightarrow}(p_0)$ of $\dot\neg A_0 \vdash X$ into a proof of $X \in \mathcal{S}_\omega$ which is then given as argument to m_0, giving a proof $m_1 \triangleq (y : Y \in \mathcal{S}_\omega) \mapsto (2n_0+1, (\dot\neg A_0), inj_{2n_0+1}, \pi_1^{\dot\Rightarrow}(p_0))$. Now, m_1 is turned using $\downarrow_{Y\dot\Rightarrow X}$ into a proof of $Y\dot\Rightarrow X \in \mathcal{S}_\omega$. The proof m_1 drops its argument, so that $\downarrow_{Y\dot\Rightarrow X}$ returns the tuple $(2n_1+1, (\dot\neg A_0, Y\dot\Rightarrow X), I_{\Rightarrow}(inj_{2n_1+1}, k_1), \dot{ax}_0)$ where $2n_1+1$ is $\lceil Y\dot\Rightarrow X \rceil$ and k_1 is the continuation

$$k_1(\Gamma, f, p) \triangleq \phi(2n_1+1, \Gamma, f, \pi_2^{\dot\Rightarrow}\ p, (2n_0+1, (\dot\neg A_0), inj_{2n_0+1}, \pi_1^{\dot\Rightarrow}\ p_0))$$

We are then evaluating the following:

$$\dot{dn}(\dot{abs}^{\Rightarrow}(\phi(2n_0+1, (\dot\neg A_0), inj_{2n_0+1}, \pi_2^{\dot\Rightarrow}\ p_0, (2n_1+1, (\dot\neg A_0, Y\dot\Rightarrow X), I_{\Rightarrow}(inj_{2n_1+1}, k_1), \dot{ax}_0))))$$

which is the same as

$$\dot{dn}(\dot{abs}^{\Rightarrow}(flush^{(\dot\neg A_0)\cup(\dot\neg A_0, Y\dot\Rightarrow X)}_{max(2n_0+1, 2n_1+1)}(merge^{(\dot\neg A_0)(\dot\neg A_0, Y\dot\Rightarrow X)}_{(2n_0+1)(2n_1+1)}(inj_{2n_0+1}, I_{\Rightarrow}(inj_{2n_1+1}, k_1)), p_1))$$

where $p_1 \triangleq \dot{app}^{\Rightarrow}(\pi_2^{\dot\Rightarrow}\ p_0, \dot{ax}_0)$ is a proof of $\dot\neg A_0, Y\dot\Rightarrow X \vdash \bot$ obtained as the application of the proof $\pi_2^{\dot\Rightarrow}(p_0)$ of $\dot\neg A_0 \vdash \dot\neg(Y\dot\Rightarrow X)$ to the proof $\dot{ax}_0$ of $\dot\neg A_0, Y\dot\Rightarrow X \vdash Y\dot\Rightarrow X$ coming from calling $\downarrow_{Y\dot\Rightarrow X}$.

The *merge* is direct even if its result depends on whether $n_1 \geq n_0$ or $n_1 < n_0$. In both cases, it results in making *flush* applying k_1 so as to remove the hypothesis $Y\dot\Rightarrow X$ from p_1. We are now at evaluating the following:

$$\dot{dn}(\dot{abs}^{\Rightarrow}(k_1((\dot\neg A_0), inj_{2n_1+1}, p_1)))$$

which gives:

$$\dot{dn}(\dot{abs}^{\Rightarrow}(flush^{(\dot\neg A_0)\cup(\dot\neg A_0)}_{max(2n_1+1, 2n_0+1)}(merge^{(\dot\neg A_0)(\dot\neg A_0)}_{(2n_1+1)(2n_0+1)}(inj_{2n_1+1}, inj_{2n_0+1}), \dot{app}^{\Rightarrow}(\pi_2^{\dot\Rightarrow}(p_1), \pi_1^{\dot\Rightarrow}(p_0)))))$$

The *merge* and *flush* are now trivial and we obtain

$$\dot{dn}(\dot{abs}^{\Rightarrow}(p_2))$$

where $p_2 \triangleq \dot{app}^{\Rightarrow}(\pi_2^{\dot\Rightarrow}(p_1), \pi_1^{\dot\Rightarrow}(p_0))$ combines a proof of $\dot\neg A_0 \vdash \dot\neg X$ with a proof of $\dot\neg A_0 \vdash X$ to get a proof of $\dot\neg A_0 \vdash \bot$.

To summarise, the object proof produced is:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\dot\neg A_0 \vdash \dot\neg A_0}\dot{ax} \quad \dfrac{}{A_0 \vdash A_0}\dot{ax}}{\dot\neg A_0, A_0 \vdash \bot}\dot{app}^{\Rightarrow}}{\dot\neg A_0 \vdash \dot\neg(Y\dot\Rightarrow X)}\pi_2^{\dot\Rightarrow} \quad \dfrac{}{\dot\neg A_0, Y\dot\Rightarrow X \vdash Y\dot\Rightarrow X}\dot{ax}}{\dot\neg A_0, Y\dot\Rightarrow X \vdash \bot}\dot{app}^{\Rightarrow}}{\dot\neg A_0 \vdash \dot\neg X}\pi_2^{\dot\Rightarrow} \quad \dfrac{\dfrac{\dfrac{}{\dot\neg A_0 \vdash \dot\neg A_0}\dot{ax} \quad \dfrac{}{A_0 \vdash A_0}\dot{ax}}{\dot\neg A_0, A_0 \vdash \bot}\dot{app}^{\Rightarrow}}{\dot\neg A_0 \vdash X}\pi_1^{\dot\Rightarrow}}{\dot\neg A_0 \vdash \bot}\dot{app}^{\Rightarrow}}{\vdash \dot\neg\dot\neg A_0}\dot{abs}^{\Rightarrow}}{\vdash A_0}\dot{dn}$$

As a matter of comparison, for the canonical proof of the validity of $A_1 \triangleq X \dot{\Rightarrow} Y \dot{\Rightarrow} Y$, one would have obtained instead:

$$
\dfrac{
\dfrac{
\dfrac{
\dfrac{
\dfrac{
\dfrac{}{\dot\neg A_1 \vdash \dot\neg A_1}\,\dot{a}x
\quad
\dfrac{}{A_1 \vdash A_1}\,\dot{a}x
}{\dot\neg A_1, A_1 \vdash \dot\bot}\,\dot{app}^{\dot\Rightarrow}
}{\dot\neg A_1 \vdash \dot\neg(Y \dot\Rightarrow Y)}\,\pi_2^{\dot\Rightarrow}
\quad
\dfrac{}{\dot\neg A_1, Y \dot\Rightarrow Y \vdash Y \dot\Rightarrow Y}\,\dot{a}x
}{\dot\neg A_1, Y \dot\Rightarrow Y \vdash \dot\bot}\,\dot{app}^{\dot\Rightarrow}
}{\dot\neg A_1 \vdash \dot\neg Y}\,\pi_2^{\dot\Rightarrow}
\quad
\dfrac{
\dfrac{
\dfrac{
\dfrac{
\dfrac{}{\dot\neg A_1 \vdash \dot\neg A_1}\,\dot{a}x
\quad
\dfrac{}{A_1 \vdash A_1}\,\dot{a}x
}{\dot\neg A_1, A_1 \vdash \dot\bot}\,\dot{app}^{\dot\Rightarrow}
}{\dot\neg A_1 \vdash \dot\neg(Y \dot\Rightarrow Y)}\,\pi_2^{\dot\Rightarrow}
\quad
\dfrac{}{\dot\neg A_1, Y \dot\Rightarrow Y \vdash Y \dot\Rightarrow Y}\,\dot{a}x
}{\dot\neg A_1, Y \dot\Rightarrow Y \vdash \dot\bot}
}{\dot\neg A_1 \vdash Y}\,\pi_1^{\dot\Rightarrow}
}{
\dfrac{
\dfrac{\dot\neg A_1 \vdash \dot\bot}{\vdash \dot\neg\dot\neg A_1}\,\dot{abs}^{\dot\Rightarrow}
}{\vdash A_1}\,\dot{d}n
}\,\dot{app}^{\dot\Rightarrow}
$$

In particular, this means that the two canonical proofs of validity of $X \dot\Rightarrow X \dot\Rightarrow X$ would not produce the same object language proofs.

2.5 Discussion on positive connectives

We suspect that our presentation of Henkin's proof can be extended into a computational proof of completeness (with respect to possibly-exploding models) in the presence of disjunction by adapting Veldman's intuitionistic proof of completeness for intuitionistic logic with disjunction [61] to the case of Gödel's completeness. One would then need the Fan theorem.

To support existential quantification, it is enough to consider an enumeration of formulae which take existential formulae into account and then to add a clause to the definition of $\Gamma \subset \mathcal{S}_n$ similar to the one for universal quantification, but using instead Henkin's axiom $\dot\exists y\, A(y) \dot\Rightarrow A(x)$ for x taken fresh in the finite set of formulas coming before $\dot\exists y\, A(y)$ in the enumeration[15].

References

[1] Thorsten Altenkirch, Peter Dybjer, Martin Hofmann, and Philip J. Scott. Normalization by evaluation for typed lambda calculus with coproducts. In *LICS '01: Proceedings of the 16th Annual IEEE Symposium on Logic in Computer Science*, pages 303–310, Washington, DC, USA, 2001. IEEE Computer Society.

[2] Thorsten Altenkirch, Martin Hofmann, and Thomas Streicher. Reduction-free normalisation for a polymorphic system. In *Proceedings, 11th Annual IEEE Symposium on Logic in Computer Science, New Brunswick, New Jersey, USA, July 27-30, 1996*, pages 98–106, 1996.

[3] Zena M. Ariola and Hugo Herbelin. Minimal classical logic and control operators. In *Thirtieth International Colloquium on Automata, Languages and Programming , ICALP'03, Eindhoven, The Netherlands, June 30 - July 4, 2003*, volume 2719 of *Lecture Notes in Computer Science*, pages 871–885. Springer-Verlag, 2003.

[4] Andrea Asperti, Wilmer Ricciotti, Claudio Sacerdoti Coen, and Enrico Tassi. Formal metatheory of programming languages in the matita interactive theorem prover. *J. Autom. Reasoning*, 49(3):427–451, 2012.

[5] Jon Barwise. An introduction to first-order logic. In Jon Barwise, editor, *Handbook of Mathematical Logic*, volume 90 of *Studies in Logic and The Foundations of Mathematics*, pages 5–46. Elsevier, 1977.

[6] Stefano Berardi. Intuitionistic completeness for first order classical logic. *J. Symb. Log.*, 64(1):304–312, 1999.

[7] Stefano Berardi and Silvio Valentini. Krivine's intuitionistic proof of classical completeness (for countable languages). *Ann. Pure Appl. Logic*, 129(1-3):93–106, 2004.

[15] We can also reuse Henkin's axiom $\dot\neg A(x) \dot\Rightarrow \dot\forall y\, \dot\neg A(y)$ up to some extra classical reasoning in the object language.

[8] Ulrich Berger and Helmut Schwichtenberg. An inverse of the evaluation functional for typed lambda-calculus. In *Proceedings, Sixth Annual IEEE Symposium on Logic in Computer Science, 15-18 July, 1991, Amsterdam, The Netherlands*, pages 203–211. IEEE Computer Society, 1991.

[9] Evert Willem Beth. Semantic entailment and formal derivability. *Meded. Kon. Ned. Akad. Wetenschappen, Afd. Letterhunde, N.R.*, 18:309–342, 1955.

[10] Samuel R. Buss. An introduction to proof theory. In Samuel R. Buss, editor, *Handbook of Proof Theory*, pages 1–78. North-Holland, North-Holland, 1998.

[11] Alonzo Church. *Introduction to Mathematical Logic*, volume 1. The Princeton University Press, 1956.

[12] The Coq Development Team. *The Coq Reference Manual, version 8.5*, January 2016. Distributed electronically at `http://coq.inria.fr/doc`.

[13] Catarina Coquand. A formalised proof of the soundness and completeness of a simply typed lambda-calculus with explicit substitutions. *Higher Order Symbol. Comput.*, 15(1):57–90, 2002.

[14] Thierry Coquand and Peter Dybjer. Intuitionistic model constructions and normalization proofs. *Mathematical Structures in Computer Science*, 7(1):75–94, 1997.

[15] Olivier Danvy. Type-directed partial evaluation. In *POPL*, pages 242–257, 1996.

[16] Solomon Feferman, John W. Dawson Jr, Stephen C. Kleene, Gregory H. Moore, Robert M. Solovay, and Jean van Heijenoort, editors. *Kurt Gödel: Collected Works. Vol. 1: Publications 1929-1936*. Oxford University Press, New York, 1986.

[17] Gottlob Frege. *Begriffschrift, eine der arithmetischen nachgebildete Formelsprache des reinen Denkens*. Halle, 1879. English translation e.g. in [18] or [60].

[18] Gottlob Frege. *Begriffschrift, eine der arithmetischen nachgebildete Formelsprache des reinen Denkens*. Halle, 1879. English translation e.g. in [60].

[19] Harvey Friedman. Intuitionistic completeness of heyting's predicate calculus. *Notices of the American Mathematical Society*, 22:A–648, 1975.

[20] Harvey Friedman. Classically and intuitionistically provably recursive functions. In D. S. Scott and G. H. Muller, editors, *Higher Set Theory*, volume 669 of *Lecture Notes in Mathematics*, pages 21–27. Springer, Berlin/Heidelberg, 1978.

[21] Gerhard Gentzen. Untersuchungen über das logische Schließen. *Mathematische Zeitschrift*, 39:176–210,405–431, 1935. English Translation in [59], "Investigations into logical deduction", pages 68-131.

[22] Kurt Gödel. *Über die Vollständigkeit des Logikkalküls*. Doctoral thesis, University of Vienna, 1929. English translation in [60] or [16].

[23] Kurt Gödel. Die vollständigkeit des axiome des logischen functionenkalküls. *Monatshefte für Mathematik und Physik*, 37:349–360, 1930. Reprinted in [16].

[24] Kurt Gödel. Über formal unentscheidbare sätze der principia mathematica und verwandter systeme. *Monatshefte für Mathematik und Physik*, 38:173–198, 1931.

[25] Timothy G. Griffin. The formulae-as-types notion of control. In *Conf. Record 17th Annual ACM Symp. on Principles of Programming Languages, POPL '90, San Francisco, CA, USA, 17-19 Jan 1990*, pages 47–57. ACM Press, New York, 1990.

[26] Leon Henkin. The completeness of the first-order functional calculus. *J. Symb. Log.*, 14(3):159–166, 1949.

[27] Leon Henkin. The discovery of my completeness proofs. *The Bulletin of Symbolic Logic*, 2(2):127–158, June 1996.

[28] Hugo Herbelin. An intuitionistic logic that proves Markov's principle. In *Proceedings of the 25th Annual IEEE Symposium on Logic in Computer Science, LICS 2010, 11-14 July 2010, Edinburgh, United Kingdom*, pages 50–56. IEEE Computer Society, 2010.

[29] Hugo Herbelin and Gyesik Lee. Forcing-based cut-elimination for gentzen-style intuitionistic sequent calculus. In Hiroakira Ono, Makoto Kanazama, and Ruy de Queiroz, editors, *Logic, Language, Information and Computation, 16th International Workshop, WoLLIC 2009, Tokyo, Japan, June 21-25, 2009, Proceedings*, volume 5514 of *Lecture Notes in Computer Science*, pages 209–217. Springer, 2009.

[30] Olivier Hermant. Semantic cut elimination in the intuitionistic sequent calculus. In Pawel Urzyczyn, editor, *Typed Lambda Calculi and Applications, 7th International Conference, TLCA 2005, Nara, Japan, April 21-23, 2005, Proceedings*, volume 3461 of *Lecture Notes in Computer Science*, pages 221–233. Springer, 2005.

[31] Olivier Hermant and James Lipton. Completeness and cut-elimination in the intuitionistic theory of types - part 2. *J. Log. Comput.*, 20(2):597–602, 2010.

[32] David Hilbert and Wilhem Ackermann. *Grundzüge der theoretischen Logik*. Springer Verlag, Berlin, 1928. English translation of the 1938 edition in [33].

[33] David Hilbert and Wilhem Ackermann. *Principles of Mathematical Logic*. Oxford University Press, New York, 1928.

[34] Jaakko Hintikka. Form and content in quantification theory. *Acta Philosophica Fennica*, 8:11–55, 1955.

[35] Jaakko Hintikka. Notes on the quantification theory. *Societas Scientiarum Fennica, Commentationes physico-mathematicae*, 17(1), 1955.

[36] Danko Ilik. *Constructive Completeness Proofs and Delimited Control*. Ph.D. thesis, École Polytechnique, October 2010.

[37] Danko Ilik. Continuation-passing style models complete for intuitionistic logic. *Annals of Pure and Applied Logic*, 164(6), 2016.

[38] Stanislas Jaśkowski. On the rules of suppositions in formal logic. *Studia Logica*, 1, 1934. Reprinted in S. McCall (1967) Polish Logic 1920-1939, Oxford: Oxford Univ. Press, pp. 232–258.

[39] Stig Kanger. *Provability in Logic*. Acta Universitatis Stockholmiensis. Stockholm, Almqvist & Wiksell, 1957.

[40] Stephen C. Kleene. Recursive functions and intuitionistic mathematics. In L.M. Graves, E. Hille, P.A. Smith, and O. Zariski, editors, *Proceedings of the International Congress of Mathemaiticans*, page 679–685. Cambridge, Mass., August 1950.

[41] Georg Kreisel. Elementary completeness properties of intuitionistic logic with a note on negations of prenex formulae. *Journal of Symbolic Logic*, 23(3):317–330, 1958.

[42] Georg Kreisel. On weak completeness of intuitionistic predicate logic. *J. Symb. Log.*, 27(2):139–158, 1962.

[43] Jean-Louis Krivine. *Lambda-calcul, types et modèles*. Masson, Paris, 1990. English translation by René Cori at https://www.irif.fr/ krivine/articles/Lambda.pdf.

[44] Jean-Louis Krivine. Opérateurs de mise en mémoire et traduction de gödel. *Arch. Math. Logic*, 30(4):241–267, 1990.

[45] Jean-Louis Krivine. *Lambda-calculus, types and models*. Ellis Horwood, 1993.

[46] Jean-Louis Krivine. Classical logic, storage operators and second-order lambda-calculus. *Ann. Pure Appl. Logic*, 68(1):53–78, 1994.

[47] Jean-Louis Krivine. Une preuve formelle et intuitionniste du théorème de complétude de la logique classique. *Bulletin of Symbolic Logic*, 2(4):405–421, 1996.

[48] Victor N. Krivtsov. An intuitionistic completeness theorem for classical predicate logic. *Studia Logica: An International Journal for Symbolic Logic*, 96(1):109–115, October 2010.

[49] Victor N. Krivtsov. Semantical completeness of first-order predicate logic and the weak fan theorem. *Studia Logica*, 103(3):623–638, 2015.

[50] Per Martin-Löf. An intuitionistic theory of types, predicative part. In *Logic Colloquium*, pages 73–118. North Holland, 1975.

[51] Charles McCarty. Completeness and incompleteness for intuitionistic logic. *The Journal of Symbolic Logic*, 73(4), December 2008.

[52] Mitsuhiro Okada. A uniform semantic proof for cut-elimination and completeness of various first and higher order logics. *Theor. Comput. Sci.*, 281(1-2):471–498, 2002.

[53] Michel Parigot. Lambda-mu-calculus: An algorithmic interpretation of classical natural deduction. In *Logic Programming and Automated Reasoning: International Conference LPAR '92 Proceedings, St. Petersburg, Russia*, pages 190–201. Springer-Verlag, 1992.

[54] Dag Prawitz. *Natural Deduction, a Proof-Theoretical Study*. Almqvist and Wiksell, Stockholm, 1965.

[55] Helena Rasiowa and Roman Sikorski. A proof of the completeness theorem of grödel. *Fundamenta Mathematicae*, 37(1):193–200, 1950.

[56] Helmut Schwichtenberg and Stanley S. Wainer. *Proofs and Computations*. Cambridge University Press, 2011.

[57] Kurt Schütte. Ein system des verknüpfenden schliessens. *Archiv für mathematische Logik und Grundlagenforschung*, 2(2-4):55–67, 1956.

[58] Stephen G. Simpson. *Subsystems of Second Order Arithmetic*. Springer seminars in immunopathology. Springer, 1999.

[59] Manfred E. Szabo, editor. *The Collected Works of Gerhard Gentzen*. North Holland, Amsterdam, 1969.

[60] Jean van Heijenoort, editor. *From Frege to Gödel, A Source Book in Mathematical Logic, 1879-1931*. Harvard University Press, New York, 1967.

[61] Wim Veldman. An intuitionistic completeness theorem for intuitionistic predicate logic. *J. Symb. Log.*, 41(1):159–166, 1976.