Completeness Theorems for First-Order Logic Analysed in Constructive Type Theory

Yannick Forster, Dominik Kirst, Dominik Wehr

Saarland University, Saarland Informatics Campus, Saarbrücken, Germany
{forster,kirst}@ps.uni-saarland.de dwehr@dortselb.st

Abstract. We study various formulations of the completeness of first-order logic phrased in constructive type theory and mechanised in the Coq proof assistant. Specifically, we examine the completeness of variants of classical and intuitionistic natural deduction and sequent calculi with respect to model-theoretic, algebraic, and game semantics. As completeness with respect to standard model-theoretic semantics is not readily constructive, we analyse the assumptions necessary for particular syntax fragments and discuss non-standard semantics admitting assumption-free completeness. We contribute a reusable Coq library for first-order logic containing all results covered in this paper.

1 Introduction

Completeness theorems are central to the field of mathematical logic. Once completeness of a sound deduction system with respect to a semantic account of the syntax is established, the typically infinitary notion of semantic validity is reduced to the finitary, and hence algorithmically more tractable, notion of syntactic deduction. In the case of first-order logic, being the formalism underlying traditional mathematics based on a set-theoretic foundation, completeness enables the use of semantic techniques to study the deductive consequence of axiomatic systems.

The seminal completeness theorem for first-order logic proven by Gödel [18] and later refined by Henkin [21,20] yields a syntactic deduction of every formula valid in the canonical Tarski semantics based on interpreting the non-logical function and relation symbols in models providing the corresponding structure. However, this result may not be understood as an effective procedure in the sense that a formal deduction for a formula satisfied by all models can be computed by an algorithm, since even for finite signatures the proof relies on non-constructive assumptions. Specifically, when admitting all logical connectives, completeness is equivalent to a weak form of König's lemma [33]. Even restricted to the classically sufficient $\rightarrow, \forall, \bot$-fragment, the classically vacuous but constructively contested[1] assumption of Markov's principle, asserting that every non-diverging computation terminates, is necessary [29]. We defer a more detailed overview of known dependencies to the discussion of related work in Section 7.1.

[1] Accepted in Russian constructivism while in conflict with Brouwer's intuitionism.

The aim of this paper is to coherently analyse the computational content of completeness theorems concerning various semantics and deduction systems. Naturally, such matters of *constructive reverse mathematics* [26] need to be addressed in an intuitionistic meta-logic such as constructive type theory. In fact, the results in this paper are formalised in the Coq proof assistant [51] that implements the *predicative calculus of cumulative inductive constructions* (pCuIC) [52], yielding executable programs for all constructively given completeness proofs. For ease of language, we reserve the term "constructive" for statements provable in this specific system, hence excluding Markov's principle [7,42]. In fact, coming with an internal notion of computation, constructive type theory allows us to state Markov's principle both internally (MP) as well as for any concrete model of computation (MP_L), whereby the former implies the latter and both can be related to completeness statements. The two main questions in focus are which specific assumptions are necessary for particular formulations of completeness and how the statements can be modified such that they hold constructively.

Applying this strategy to Tarski semantics, a first observation is that the model existence theorem, central to Henkin's completeness proof, holds constructively [23]. Model existence directly implies that valid formulas cannot be unprovable. Thus, a single application of MP, rendering enumerable predicates such as deduction stable under double negation, yields completeness. Similarly, MP_L yields the stability of deduction from finite contexts and hence the corresponding form of completeness. Because MP is admissible in pCuIC [42], so are MP_L and the two completeness statements. Finally, we illustrate that completeness for the minimal $\rightarrow, \forall$-fragment does not depend on additional assumptions and, consequently, how the interpretation of $\bot$ can be relaxed to *exploding models* [54,31] admitting a constructive completeness proof for the $\rightarrow, \forall, \bot$-fragment.

Turning to intuitionistic logic, we discuss analogous relationships for Kripke semantics and a cut-free intuitionistic sequent calculus [24]. Again, completeness for the $\rightarrow, \forall, \bot$-fragment is equivalent to Markov's principle while being constructive if restricted to the minimal $\rightarrow, \forall$-fragment or employing a relaxed treatment of $\bot$. The intuitionistically undefinable connectives $\vee$ and $\exists$ add further complexity [25] and need to remain untreated in this paper. As a side note, we explain how the constructivised completeness theorem for intuitionistic logic can be used to implement a semantic cut-elimination procedure.

After considering such model-theoretic semantics, mainly based on embedding the object-logic into the meta-logic, we exemplify two rather different approaches to assigning meaning to formulas, namely algebraic semantics and game semantics. Differing fundamentally from model-theoretic semantics, both share a constructive rendering of completeness for the full syntax of first-order logic, agnostic to the intuitionistic or classical flavour of the deduction system.

In algebraic semantics, the embedding of formulas into the meta-logic is generalised to an evaluation in algebras providing the structure of the logical connectives. In this setting, completeness follows from the observation that provability induces such an algebra on formulas. We discuss intuitionistic and classical logic evaluated in complete Heyting and complete Boolean algebras (cf. [47]).

Dialogue game semantics as introduced by Lorenzen [35,36], on the other hand, completely disposes of interpreting logical connectives as operations and instead understand logic as a dialectic game of assertion and argument. An assertion is considered valid if every sceptic can be convinced through substantive reasoning, i.e. if there is a strategy such that every argument about the assertion can be won. Hence, game semantics are inherently closer to deduction systems than the previous semantic accounts and in fact a very general isomorphism of winning strategies and formal deductions has been established [48]. We instantiate this isomorphism to a first-order intuitionistic sequent calculus.

Contributions. We present a comprehensive analysis of the computational content of completeness theorems for first-order logic considering various semantics and deduction systems. Concerning model-theoretic semantics, we refine the well-known relation of completeness for $\rightarrow, \forall, \bot$-formulas to Markov's principle to constructive completeness up to double negation, hence entailing the admissibility of completeness in pCuIC. Our elaboration of game semantics introduces a streamlined representation of dialogues as state transition systems suitable for mechanisation and translates the generic completeness result for classical logic from [48] to the case of intuitionistic first-order logic. Finally, we provide a reusable Coq library[2] for first-order logic including all results covered in this paper. Notably, the development is based on a de Bruijn encoding of binders [8,50] and is parametric in the signature of non-logical symbols and thus adjustable to any particular first-order theory (see Appendix B for more formalisation details).

Outline. In Section 2, we begin with some preliminary definitions concerning the syntax of first-order logic, deduction systems, and synthetic computability. We then analyse completeness for model-theoretic semantics (Section 3) and its connection to Markov's principle (Section 4). Subsequently, we give constructive completeness proofs for algebraic semantics (Section 5) and game semantics (Section 6). We end with a discussion of related and future work in Section 7.

2 Syntax, Deduction, Computability

We work in a constructive type theory with a predicative hierarchy of type universes above a single impredicative universe $\mathbb{P}$ of propositions. Assumed type formers are function spaces $X \rightarrow Y$, products $X \times Y$, sums $X + Y$, dependent products $\forall x : X.\, F\, x$, and dependent sums $\Sigma\, x : X.\, F\, x$. The propositional versions of these connectives are denoted by the usual logical symbols ($\rightarrow$, $\wedge$, $\vee$, $\forall$, and $\exists$) in addition to $\top : \mathbb{P}$ and $\bot : \mathbb{P}$ denoting truth and falsity.

Basic inductive types are the Booleans $\mathbb{B} ::= \mathsf{tt} \mid \mathsf{ff}$ and the natural numbers $\mathbb{N} ::= 0 \mid \mathsf{S}\, n$ for $n : \mathbb{N}$. Given a type X, we further define options $\mathcal{O}(X) ::= \emptyset \mid \ulcorner x \urcorner$ and lists $\mathcal{L}(X) ::= [] \mid x :: A$ for $x : X$ and $A : \mathcal{L}(X)$. On lists we employ the standard notation for membership $x \in A$, inclusion $A \subseteq B$, concatenation $A +\!\!+ B$, and map $f @ A$. These notations are shared with vectors $\mathbf{x} : X^n$ of fixed length $n : \mathbb{N}$. Possibly infinite collections are expressed by sets $p : X \rightarrow \mathbb{P}$ with set-theoretic notations like $x \in p$, $p \subseteq q$, and $p \cap q$.

[2] On `www.ps.uni-saarland.de/extras/fol-completeness` and hyperlinked with this document.

2.1 Syntax of First-Order Logic

We represent the terms and formulas of first-order logic as inductive types over a fixed signature $\Sigma = (\mathcal{F}_\Sigma, \mathcal{P}_\Sigma)$ specialising function symbols $f : \mathcal{F}_\Sigma$ and predicate symbols $P : \mathcal{P}_\Sigma$ together with their arities $|f| : \mathbb{N}$ and $|P| : \mathbb{N}$. Variable binding is implemented using de Bruijn indices [8] well-suited for formalisation [50].

Definition 1. We define the terms and formulas of first-order logic by

$$t : \mathbb{T} ::= x \mid f\,\mathbf{t} \quad \varphi, \psi : \mathbb{F} ::= \dot\bot \mid P\,\mathbf{t} \mid \varphi \dot\to \psi \mid \varphi \dot\wedge \psi \mid \varphi \dot\vee \psi \mid \dot\forall \varphi \mid \dot\exists \varphi \quad x : \mathbb{N}, f : \mathcal{F}_\Sigma, P : \mathcal{P}_\Sigma$$

where the vectors $\mathbf{t}$ are of the expected lengths $|f|$ and $|P|$, respectively. We set $\dot\neg \varphi := \varphi \dot\to \dot\bot$ and isolate the type $\mathbb{F}^*$ of formulas in the $\to, \forall, \bot$-fragment.

A bound variable is encoded as the number of quantifiers shadowing its relevant binder, e.g. $P\,x\,y \to \forall x.\, \exists y.\, P\,x\,y$ may be represented by $P\,7\,4 \dot\to \dot\forall\, \dot\exists P\,1\,0$. The variables 7 and 4 in this example are called *free* and variables that do not occur freely are called *fresh*. A formula with no free variables is called *closed*.

Definition 2. Instantiating with a substitution $\sigma : \mathbb{N} \to \mathbb{T}$ is defined by

$$\begin{array}{rclrclrcl} x[\sigma] & := & \sigma\,x & \dot\bot[\sigma] & := & \dot\bot & (\varphi \boxdot \psi)[\sigma] & := & \varphi[\sigma] \boxdot \psi[\sigma] \\ (f\,\mathbf{t})[\sigma] & := & f\,(\mathbf{t}\,[\sigma]) & (P\,\mathbf{t})[\sigma] & := & P\,(\mathbf{t}\,[\sigma]) & (\boxdot\,\varphi)[\sigma] & := & \boxdot\,\varphi[\uparrow\sigma] \end{array}$$

where $\mathbf{t}\,[\sigma]$ is short for $(\lambda t.\, t[\sigma])\,@\,\mathbf{t}$, $\uparrow\sigma$ denotes the substitution $\lambda n.\, \sigma\,(\mathsf{S}\,n)$, and $\boxdot$ is used as placeholder for the logical connectives and quantifiers, respectively.

Useful shorthands are $\varphi[t;\sigma]$ for instantiating 0 with t and $\mathsf{S}\,x$ with $\sigma\,x$, $\varphi[t]$ for $\varphi[t; \lambda x.\, x]$, and $\uparrow\varphi$ for the shift $\varphi[\lambda x.\, \mathsf{S}\,x]$. All terminology and notation concerning formulas carries over to *contexts* $\Gamma : \mathcal{L}(\mathbb{F})$ and *theories* $\mathcal{T} : \mathbb{F} \to \mathbb{P}$. For ease of notation we freely identify contexts Γ with their theory $\lambda\varphi.\, \varphi \in \Gamma$.

2.2 Deduction Systems

We represent deduction systems as inductive predicates of type $\mathcal{L}(\mathbb{F}) \to \mathbb{F} \to \mathbb{P}$ or similar. The archetypal system is natural deduction (ND), exemplified by an intuitionistic version $\Gamma \vdash \varphi$ as defined in Definition 55 of Appendix A. Since most rules are standard, we only discuss the quantifier rules in more detail as they rely on the de Bruijn representation of formulas:

$$\frac{\uparrow\Gamma \vdash \varphi}{\Gamma \vdash \dot\forall\varphi}\ \text{AI} \qquad \frac{\Gamma \vdash \dot\forall\varphi}{\Gamma \vdash \varphi[t]}\ \text{AE} \qquad \frac{\Gamma \vdash \varphi[t]}{\Gamma \vdash \dot\exists\varphi}\ \text{EI} \qquad \frac{\Gamma \vdash \dot\exists\varphi \quad \uparrow\Gamma, \varphi \vdash \uparrow\psi}{\Gamma \vdash \psi}\ \text{EE}$$

Note that $\uparrow\Gamma, \varphi$ is notation for $\varphi :: \uparrow\Gamma$. In a shifted context $\uparrow\Gamma$ there is no reference to the variable 0 which hence plays the role of an arbitrary but fixed individual. So if $\uparrow\Gamma \vdash \varphi$ then we can conclude $\Gamma \vdash \dot\forall\varphi$ as expressed by the rule (AI) for $\forall$-introduction. Similarly, the shifts in the rule (EE) for $\exists$-elimination simulate that Γ together with φ instantiated to the witness provided by $\Gamma \vdash \dot\exists\varphi$ proves ψ and hence admits the conclusion that already $\Gamma \vdash \psi$. For many proofs it will be helpful to employ fresh variables explicitly as justified by Lemma 4, which we state after observing *weakening* and *substitutivity*:

Lemma 3. *If $\Gamma \vdash \varphi$, then $\Delta \vdash \varphi$ for all $\Delta \supseteq \Gamma$ and $\Gamma[\sigma] \vdash \varphi[\sigma]$ for all σ.*

Lemma 4. *Given Γ, φ, and ψ one can compute a fresh variable x such that*
1. $\uparrow\Gamma \vdash \varphi$ iff $\Gamma \vdash \varphi[x]$ and *2. $\uparrow\Gamma, \varphi \vdash \uparrow\psi$ iff $\Gamma, \varphi[x] \vdash \psi$.*

A classical variant $\Gamma \vdash_c \varphi$ of the ND system can be obtained without referring to $\dot{\bot}$ by adding the axiom $\Gamma \vdash_c ((\varphi \dot{\to} \psi) \dot{\to} \varphi) \dot{\to} \varphi$ expressing Peirce's law (Definition 56). Then the structural properties stated in the two lemmas above are maintained while the typical classical proof rules become available.

Deduction systems such as intuitionistic ND introduced above naturally extend to theories by writing $\mathcal{T} \vdash \varphi$ if there is a finite context $\Gamma \subseteq \mathcal{T}$ with $\Gamma \vdash \varphi$. Then $\mathcal{T} \vdash \varphi$ satisfies proof rules analogous to $\Gamma \vdash \varphi$.

2.3 Synthetic Computability

Since every function definable in constructive type theory is computable, the standard notions of computability theory can be synthesised by type-level operations [1,14], eliminating references to a concrete model of computation such as Turing machines, μ-recursive functions, or the untyped lambda calculus.

Definition 5. Let X be a type and $p : X \to \mathbb{P}$ be a predicate.

- p is *decidable* if there is $f : X \to \mathbb{B}$ with $\forall x.\, p\, x \leftrightarrow f\, x = \mathsf{tt}$.
- p is *enumerable* if there is $f : \mathbb{N} \to \mathcal{O}(X)$ with $\forall x.\, p\, x \leftrightarrow \exists n.\, f\, n = \ulcorner x \urcorner$.

These two notions generalise to predicates of higher arity as expected.

- X is *enumerable* if there is $f : \mathbb{N} \to \mathcal{O}(X)$ with $\forall x. \exists n.\, f \ = \ulcorner x \urcorner$.
- X is *discrete* if equality $\lambda xy.x = y$ on X is decidable.
- X is a *data type* if it is both enumerable and discrete.

We assume that the components $\mathcal{F}_\Sigma$ and $\mathcal{P}_\Sigma$ of our fixed signature Σ are data types. Then applying the terminology to the syntax and deductions systems introduced in the previous sections leads to the following observations.

Fact 6. *$\mathbb{T}$ and $\mathbb{F}$ are data types and $\Gamma \vdash \varphi$ and $\Gamma \vdash_c \varphi$ are enumerable.*

Proof. By the techniques discussed in [14], e.g. Fact 3.19. □

The standard model-theoretic completeness proofs analysed in Section 3 require the assumption of Markov's principle. A proposition $P : \mathbb{P}$ is called *stable* if $\neg\neg P \to P$ and, analogously, a predicate $p : X \to \mathbb{P}$ is called stable if $p\, x$ is stable for all x. A synthetic version of Markov's principle states that satisfiability of Boolean sequences is stable (cf. [38]):

$$\mathsf{MP} := \forall f : \mathbb{N} \to \mathbb{B}.\, \neg\neg(\exists n.\, f\, n = \mathsf{tt}) \to \exists n.\, f\, n = \mathsf{tt}$$

Note that MP is trivially implied by excluded middle $\mathsf{EM} := \forall P : \mathbb{P}.\, P \lor \neg P$. Moreover, MP regulates the behaviour of computationally tractable predicates:

Fact 7. MP *implies that enumerable predicates on data types are stable.*

Proof. This is Fact 2.18 in [14]. □

As a consequence of Fact 6 and Fact 7, MP implies that the deduction systems $\Gamma \vdash \varphi$ and $\Gamma \vdash_c \varphi$ are stable. In fact, only these stabilities are required for the standard model-theoretic completeness proofs discussed in the next section and they are equivalent to $\mathsf{MP_L}$, a version of Markov's principle stated for the call-by-value λ-calculus L [43,17] and its halting problem $\mathcal{E}$:

$$\mathsf{MP_L} := \forall s.\, \neg\neg\mathcal{E}s \to \mathcal{E}s$$

We will prove the following in Section 4:

Lemma 8. $\mathsf{MP_L}$*, stability of* $\Gamma \vdash \varphi$*and stability of* $\Gamma \vdash_c \varphi$ *are all equivalent.*

3 Model-Theoretic Semantics

The first variant of semantics we consider is based on the idea of interpreting terms as objects in a model and embedding the logical connectives into the meta-logic. A formula is considered valid if it is satisfied by all models. The simplest case is Tarski semantics, coinciding with classical deduction via Henkin's completeness proof factoring through a (constructive) model-existence theorem [22]. Kripke semantics, coinciding with intuitionistic deduction, add more structure by connecting several models through an accessibility relation and admit a simpler completeness proof using a universal model. In this section, we only consider formulas $\varphi : \mathbb{F}^*$ in the $\to, \forall, \bot$-fragment.

3.1 Tarski Semantics

Definition 9. A *(Tarski) model* $\mathcal{M}$ over a domain D is a pair of functions

$$\_^{\mathcal{M}} \;:\; \forall f : \mathcal{F}_\Sigma.\, D^{|f|} \to D \qquad\qquad \_^{\mathcal{M}} \;:\; \forall P : \mathcal{P}_\Sigma.\, D^{|P|} \to \mathbb{P}.$$

Assignments $\rho : \mathbb{N} \to D$ are extended to *term evaluations* $\hat\rho : \mathbb{T} \to D$ by $\hat\rho\, x := \rho\, x$ and $\hat\rho\,(f\,\mathbf{t}\,) := f^{\mathcal{M}}\,(\hat\rho\,@\,\mathbf{t}\,)$ and to formulas via the relation $\mathcal{M} \vDash_\rho \varphi$ defined by

$$\mathcal{M} \vDash_\rho \dot\bot \;:=\; \bot \qquad\qquad \mathcal{M} \vDash_\rho \varphi \dot\to \psi \;:=\; \mathcal{M} \vDash_\rho \varphi \to \mathcal{M} \vDash_\rho \psi$$
$$\mathcal{M} \vDash_\rho P\,\mathbf{t} \;:=\; P^{\mathcal{M}}\,(\hat\rho\,@\,\mathbf{t}\,) \qquad\qquad \mathcal{M} \vDash_\rho \dot\forall\varphi \;:=\; \forall a : D.\, \mathcal{M} \vDash_{a;\rho} \varphi$$

where the assignment $a;\rho$ maps 0 to a and $\mathsf{S}\,x$ to $\rho\,x$. We write $\mathcal{M} \vDash \varphi$ if $\mathcal{M} \vDash_\rho \varphi$ for all ρ. $\mathcal{M}$ is called *classical* if it validates all instances of Peirce's law, i.e. $\mathcal{M} \vDash ((\varphi\dot\to\psi)\dot\to\varphi)\dot\to\varphi$ for all $\varphi, \psi : \mathbb{F}^*$. We write $\mathcal{M} \vDash_\rho \mathcal{T}$ if $\mathcal{M}_\rho \vDash \varphi$ for all $\varphi \in \mathcal{T}$ and $\mathcal{T} \vDash \varphi$ if $\mathcal{M} \vDash_\rho \varphi$ for every classical $\mathcal{M}$ and ρ with $\mathcal{M} \vDash_\rho \mathcal{T}$.

We first show that the classical deduction system $\Gamma \vdash_c \varphi$ (restricted to the considered $\to, \forall, \bot$-fragment) is *sound* for Tarski semantics.

Fact 10. *$\Gamma \vdash_c \varphi$ implies $\Gamma \vDash \varphi$.*

Proof. By induction on $\Gamma \vdash_c \varphi$ similar to the soundness proof in [14, Fact 3.14]. The classical Peirce axioms $\Gamma \vdash_c ((\varphi \dot{\to} \psi) \dot{\to} \varphi) \dot{\to} \varphi$ are sound given that we only consider classical models. □

Formally, *completeness* denotes the converse property, i.e. that $\Gamma \vDash \varphi$ implies $\Gamma \vdash_c \varphi$. We now outline a Henkin-style completeness proof for $\Gamma \vdash_c \varphi$ based on the presentation by Herbelin and Ilik [23]. The main idea is to factor through a model existence theorem, stating that every consistent context is satisfied by a syntactic model. The model existence theorem in turn is based on a theory extension lemma generalising the role of $\dot{\bot}$ to an arbitrary substitute $\varphi_\bot$:

Lemma 11. *For every closed formula $\varphi_\bot$ and closed $\mathcal{T}$ there is $\mathcal{T}' \supseteq \mathcal{T}$ with:*

1. *$\mathcal{T}'$ maintains $\varphi_\bot$-consistency, i.e. $\mathcal{T} \vdash_c \varphi_\bot$ whenever $\mathcal{T}' \vdash_c \varphi_\bot$.*
2. *$\mathcal{T}'$ is deductively closed, i.e. $\varphi \in \mathcal{T}'$ whenever $\mathcal{T}' \vdash_c \varphi$.*
3. *$\mathcal{T}'$ respects implication, i.e. $\varphi \dot{\to} \psi \in \mathcal{T}'$ iff $\varphi \in \mathcal{T}' \to \psi \in \mathcal{T}'$.*
4. *$\mathcal{T}'$ respects universal quantification, i.e. $\dot{\forall}\varphi \in \mathcal{T}'$ iff $\forall t.\, \varphi[t] \in \mathcal{T}'$.*

Proof. We fix an enumeration φ_n of $\mathbb{F}^*$ such that x is fresh for φ_n if $x \geq n$. The extension can be separated into three steps, all maintaining $\varphi_\bot$-consistency:

a. $\mathcal{E} \supseteq \mathcal{T}$ which is *exploding*, i.e. $(\varphi_\bot \dot{\to} \varphi) \in \mathcal{E}$ for all closed φ.
b. $\mathcal{H} \supseteq \mathcal{E}$ which is *Henkin*, i.e. $(\varphi_n[n] \dot{\to} \dot{\forall} \varphi_n) \in \mathcal{H}$ for all n.
c. $\Omega \supseteq \mathcal{H}$ which is *maximal*, i.e. $\varphi \in \Omega$ whenever $\Omega, \varphi \vdash_c \varphi_\bot$ implies $\Omega \vdash_c \varphi_\bot$.

Note that being exploding allows to use $\varphi_\bot$ analogously to $\dot{\bot}$ and that being Henkin ensures that there is no mismatch between the provability of a universal formula and all its instances. We first argue why Ω satisfies the claims (1)-(4) of the extension lemma.

1. Ω is a $\varphi_\bot$-consistent extension of $\mathcal{T}$ since all steps maintain $\varphi_\bot$-consistency.
2. Let $\Omega \vdash_c \varphi$ and assume $\Omega, \varphi \vdash_c \varphi_\bot$, so $\Omega \vdash_c \varphi_\bot$. Thus $\varphi \in \Omega$ per maximality.
3. The first direction is immediate as Ω is deductively closed. We prove the converse using maximality, so assume $\Omega, \varphi \dot{\to} \psi \vdash_c \varphi_\bot$. It suffices to show that $\Omega \vdash_c \varphi$ since then $\varphi \in \Omega$, $\psi \in \Omega$, and ultimately $\Omega \vdash_c \varphi_\bot$ follow. $\Omega \vdash_c \varphi$ can be derived by proof rules for $\varphi_\bot$ analogous to the ones for $\dot{\bot}$.
4. The first direction is again immediate by Ω being deductively closed and the converse exploits that Ω is Henkin as follows. Suppose $\forall t.\, \varphi[t] \in \Omega$ and let φ be φ_n in the given enumeration. Then in particular $\varphi_n[n] \in \Omega$ and since Ω is Henkin also $\varphi_n[n] \dot{\to} \dot{\forall} \varphi_n \in \Omega$ which is enough to derive $\dot{\forall}\varphi \in \Omega$.

We now discuss the three extension steps separately:

a. Since the requirement is unconditional, we just add all needed formulas:

$$\mathcal{E} := \mathcal{T} \cup \{\varphi_\bot \dot{\to} \varphi \mid \varphi \text{ closed}\}$$

We only have to argue that $\mathcal{E}$ maintains $\varphi_\bot$-consistency over $\mathcal{T}$. So suppose $\mathcal{E} \vdash_c \varphi_\bot$, meaning that $\Gamma \vdash_c \varphi_\bot$ for some $\Gamma \subseteq \mathcal{E}$. We show that all added instances of explosion for $\varphi_\bot$ in Γ can be eliminated. Indeed, for $\Gamma = \Delta, \varphi_\bot \dot{\to} \varphi$ we have $\Delta \vdash_c (\varphi_\bot \dot{\to} \varphi) \dot{\to} \varphi_\bot$ and hence $\Delta \vdash_c \varphi_\bot$ by the Peirce rule. Thus by iteration there is $\Gamma' \subseteq \mathcal{T}$ with $\Gamma' \vdash_c \varphi_\bot$, justifying $\mathcal{T} \vdash_c \varphi_\bot$.

b. As above, to make $\mathcal{E}$ Henkin we just add all necessary Henkin-axioms

$$\mathcal{H} := \mathcal{E} \cup \{\varphi_n[n] \dot{\to} \dot{\forall} \varphi_n \mid n : \mathbb{N}\}$$

and justify that the extension maintains $\varphi_\bot$-consistency. So let $\Gamma \vdash_c \varphi_\bot$ for some $\Gamma \subseteq \mathcal{H}$, we again show that all added instances can be eliminated. Hence suppose $\Gamma = \Delta, \varphi_n[n] \dot{\to} \dot{\forall} \varphi_n$. Once can show that in a context Δ' extending Δ by suitable instances of $\varphi_\bot$-explosion one can derive $\Delta' \vdash_c \varphi_\bot$. In this derivation one exploits that n is fresh for φ_n and that the input theory $\mathcal{E}$ is closed. Thus ultimately $\mathcal{E} \vdash_c \varphi_\bot$.

c. The last step maximises $\mathcal{H}$ by adding all formulas maintaining $\varphi_\bot$-consistency:

$$\Omega_0 := \mathcal{H} \quad \Omega_{n+1} := \Omega_n \cup \{\varphi_n \mid \Omega_n, \varphi_n \vdash_c \varphi_\bot \text{ implies } \Omega_n \vdash_c \varphi_\bot\} \quad \Omega := \bigcup_{n:\mathbb{N}} \Omega_n$$

Note that Ω maintains $\varphi_\bot$-consistency over all Ω_n and hence $\mathcal{H}$ by construction so it remains to justify that Ω is maximal. So suppose $\Omega, \varphi_n \vdash_c \varphi_\bot$ implies $\Omega \vdash_c \varphi_\bot$, we have to show that $\varphi_n \in \Omega$. This is the case if the condition in the definition of Ω_{n+1} is satisfied, so let $\Omega_n, \varphi_n \vdash_c \varphi_\bot$. Then by the assumed implication $\Omega \vdash_c \varphi_\bot$ and since Ω maintains $\varphi_\bot$-consistency over Ω_n also $\Omega_n \vdash_c \varphi_\bot$ as required. □

Since the proof of this lemma relies on the input theory $\mathcal{T}$ to be closed, we only consider completeness for closed formulas. This is in fact enough for usual applications but we refer to the Coq development and [55] for a technically more involved generalisation incorporating formulas with free variables.

The generalisation via the falsity substitute $\varphi_\bot$ will become important later, for now the instance $\varphi_\bot := \dot{\bot}$ suffices. Also note that in usual jargon the extension $\mathcal{T}'$ of a consistent theory $\mathcal{T}$ is called *maximal consistent*, as no further formulas can be added to $\mathcal{T}'$ without breaking consistency.

Maximal consistent theories $\mathcal{T}$ give rise to equivalent *syntactic models* $\mathcal{M}_\mathcal{T}$ over the domain $\mathbb{T}$ of terms by setting $f^\mathcal{T}\,\mathbf{t} := f\,\mathbf{t}$ and $P^\mathcal{T}\,\mathbf{t} := (P\,\mathbf{t} \in \mathcal{T})$. We then observe that $\mathcal{M}_\mathcal{T} \vDash_\sigma \varphi$ iff $\varphi[\sigma] \in \mathcal{T}$ for all substitutions σ by a straighforward induction on φ using the properties stated in Lemma 11. Hence in particular $\mathcal{M}_\mathcal{T} \vDash_{\mathsf{id}} \varphi$ iff $\varphi \in \mathcal{T}$ for the identity substitution $\mathsf{id}\,x := x$. From this observation we directly conclude the model existence theorem:

Theorem 12. *Every closed and consistent theory is satisfied in a classical model.*

Proof. Let $\mathcal{T}$ be closed and consistent and let $\mathcal{T}'$ be its extension per Lemma 11 for $\varphi_\bot := \dot{\bot}$. To show $\mathcal{M}_{\mathcal{T}'} \vDash_{\mathsf{id}} \mathcal{T}$, let $\varphi \in \mathcal{T}$, hence $\varphi \in \mathcal{T}'$. Then since $\mathcal{M}_{\mathcal{T}'}$ is equivalent to $\mathcal{T}'$ we conclude $\mathcal{M}_{\mathcal{T}'} \vDash_{\mathsf{id}} \varphi$ as desired. Finally, $\mathcal{M}_{\mathcal{T}'}$ is classical due to (2) of Lemma 11. □

The model existence theorem yields completeness up to double negation:

Fact 13. *$\mathcal{T} \vDash \varphi$ implies $\neg\neg(\mathcal{T} \vdash_c \varphi)$ for closed $\mathcal{T}$ and φ.*

Proof. Suppose that $\mathcal{T} \vDash \varphi$ for closed $\mathcal{T}$ and φ and assume $\mathcal{T} \nvdash_c \varphi$ which is equivalent to $\mathcal{T}, \dot\neg\varphi$ being consistent. But then there must be a model of $\mathcal{T}, \dot\neg\varphi$ in conflict to the assumption $\mathcal{T} \vDash \varphi$. □

In fact, the remaining double negation elimination turns out to be necessary:

Fact 14. *Completeness of $\Gamma \vdash_c \varphi$ is equivalent to stability of $\Gamma \vdash_c \varphi$.*

Proof. Assuming stability, Fact 13 directly yields the completeness of $\Gamma \vdash_c \varphi$. Conversely, assume completeness and let $\neg\neg(\Gamma \vdash_c \varphi)$. Employing completeness, to get $\Gamma \vdash_c \varphi$ it suffices to show $\Gamma, \dot\neg\varphi \vDash \dot\bot$, so suppose $\mathcal{M} \vDash_\rho \Gamma, \dot\neg\varphi$ for some $\mathcal{M}$ and ρ. As we now aim at a contradiction, we can turn $\neg\neg(\Gamma \vdash_c \varphi)$ into $\Gamma \vdash_c \varphi$ and therefore obtain $\Gamma \vDash_c \varphi$ by soundness, a conflict to $\mathcal{M} \vDash_\rho \Gamma, \dot\neg\varphi$. □

Hence, we can characterise completeness of classical ND as follows.

Theorem 15. *1. Completeness of $\Gamma \vdash_c \varphi$ is equivalent to $\mathsf{MP_L}$.*
2. Completeness of $\mathcal{T} \vdash_c \varphi$ for enumerable $\mathcal{T}$ is equivalent to MP.
3. Completeness of $\mathcal{T} \vdash_c \varphi$ for arbitrary $\mathcal{T}$ is equivalent to EM.

Proof. 1. By Fact 14 completeness is equivalent to the stability of $\Gamma \vdash_c \varphi$ which is shown equivalent to $\mathsf{MP_L}$ in Section 4.
2. $\mathcal{T} \vdash_c \varphi$ for enumerable $\mathcal{T}$ is enumerable, hence stable under MP and thus complete per Fact 13. For the converse, assume a function $f : \mathbb{N} \to \mathbb{B}$ and consider $\mathcal{T} := (\lambda\varphi.\, \varphi = \dot\bot \wedge \exists n.\, f\,n = \mathsf{tt})$. Since $\mathcal{T}$ is enumerable, completeness yields that $\mathcal{T} \vDash \dot\bot$ is equivalent to $\mathcal{T} \vdash_c \dot\bot$ which in turn is equivalent to $\exists n.\, f\,n = \mathsf{tt}$. Then since $\mathcal{T} \vDash \dot\bot$ is stable so must be $\exists n.\, f\,n = \mathsf{tt}$.
3. EM particularly implies that $\mathcal{T} \vdash_c \varphi$ is stable and hence complete. Conversely given a proposition $P : \mathbb{P}$, completeness for $\mathcal{T} := (\lambda\varphi.\, \varphi = \dot\bot \wedge P)$ yields the stability of P with an argument as in (2). □

Having analysed the usual Henkin-style completeness proof, we now turn to its constructivisation. The central observation is that completeness already holds constructively for the minimal $\to, \forall$-fragment, by an elaboration of the classical proof for the minimal fragment given in [46]. To this end, we further restrict the deduction system and semantics to the minimal fragment and prove completeness via a suitable form of model existence.

Lemma 16. *In the $\to, \forall$-fragment, for closed $\mathcal{T}$ and φ there is a classical model $\mathcal{M}$ and an assignment ρ such that (1) $\mathcal{M} \vDash_\rho \mathcal{T}$ and (2) $\mathcal{M} \vDash_\rho \varphi$ implies $\mathcal{T} \vdash_c \varphi$.*

Proof. Let $\mathcal{T}'$ be the extension of $\mathcal{T}$ for $\varphi_\bot := \varphi$. As before, we have $\mathcal{M}_{\mathcal{T}'} \vDash_{\mathsf{id}} \mathcal{T}'$. So now let $\mathcal{M}_{\mathcal{T}'} \vDash_{\mathsf{id}} \varphi$, then $\varphi \in \mathcal{T}'$ and $\mathcal{T} \vdash_c \varphi$ by (1) of Lemma 11. □

Corollary 17. *In the $\to, \forall$-fragment, $\Gamma \vDash \varphi$ implies $\Gamma \vdash_c \varphi$ for closed Γ and φ.*

As opposed to completeness for fomulas incorporating $\dot\bot$, completeness in the minimal fragment does not rely on consistency requirements. Consequently, if these requirements are eliminated by allowing models treating inconsistency more liberal, completeness for formulas with $\dot\bot$ can be established constructively (cf. [54,31]).

So we now turn back to the $\rightarrow, \forall, \bot$-fragment and define a satisfaction relation $\mathcal{M} \vDash_\rho^A \varphi$ for arbitrary propositions A with the relaxed rule $(\mathcal{M} \vDash_\rho^A \dot\bot) := A$. A model $\mathcal{M}$ is *A-exploding* if $\mathcal{M} \vDash^A \dot\bot \rightarrow \varphi$ for all φ and *exploding* if it is A-exploding for some choice of A. Note that $A := \top$ and $P^{\mathcal{M}}\,\mathbf{t} := \top$ in particular yields an exploding model satisfying all formulas, hence accommodating inconsistent theories. This leads to the following formulation of model existence.

Lemma 18. *For every closed theory $\mathcal{T}$ there is an exploding classical model $\mathcal{M}$ and an assignment ρ such that (1) $\mathcal{M} \vDash_\rho^A \mathcal{T}$ and (2) $\mathcal{M} \vDash_\rho^A \dot\bot$ implies $\mathcal{T} \vdash_c \dot\bot$.*

Proof. Let $\mathcal{T}$ be closed and let $\mathcal{T}'$ be its extension for $\varphi_\bot := \dot\bot$. We set $A := \dot\bot \in \mathcal{T}'$ and observe that the syntactic model $\mathcal{M}_{\mathcal{T}'}$ still coincides with $\mathcal{T}'$, i.e. $\mathcal{M}_{\mathcal{T}'} \vDash_\sigma^A \varphi$ iff $\varphi[\sigma] \in \mathcal{T}'$. Hence we have (1) $\mathcal{M}_{\mathcal{T}'} \vDash_{\mathsf{id}}^A \mathcal{T}$. Moreover, $\mathcal{M}_{\mathcal{T}'}$ is A-exploding since proving $\mathcal{M}_{\mathcal{T}'} \vDash_\sigma^A \dot\bot \rightarrow \varphi$ in this case means to prove that $\dot\bot \dot\rightarrow \varphi[\sigma] \in \mathcal{T}'$, a straightforward consequence of $\mathcal{T}'$ being deductively closed. Finally, (2) follows from (1) of Lemma 11 as seen before. □

We write $\Gamma \vDash_e \varphi$ if $\mathcal{M} \vDash_\rho^A \varphi$ for all $A : \mathbb{P}$ and A-exploding $\mathcal{M}$ and ρ with $\mathcal{M} \vDash_\rho^A \Gamma$ and finally establish completeness with respect to exploding models:

Fact 19. *$\Gamma \vDash_e \varphi$ implies $\Gamma \vdash_c \varphi$ for closed Γ and φ.*

Proof. Let $\Gamma \vDash_e \varphi$, then $\Gamma, \dot\neg\varphi \vdash_c \dot\bot$ follows by Lemma 18 for $\mathcal{T} := \Gamma, \dot\neg\varphi$. □

3.2 Kripke Semantics

Turning to intuitionistic logic, we present Kripke semantics immediately generalised to arbitrary interpretations of falsity.

Definition 20. A *Kripke model* $\mathcal{K}$ over a domain D is a preorder $(\mathcal{W}, \preceq)$ with

$$\_^{\mathcal{K}} : \forall f : \mathcal{F}_\Sigma.\, D^{|f|} \rightarrow D \qquad \_^{\mathcal{K}} : \forall P : \mathcal{P}_\Sigma.\, \mathcal{W} \rightarrow D^{|P|} \rightarrow \mathbb{P} \qquad \bot^{\mathcal{K}} : \mathcal{W} \rightarrow \mathbb{P}.$$

The interpretations of predicates and falsity are required to be monotone, i.e. $P_v^{\mathcal{K}}\,\mathbf{a} \rightarrow P_w^{\mathcal{K}}\,\mathbf{a}$ and $\bot_v^{\mathcal{K}} \rightarrow \bot_w^{\mathcal{K}}$ whenever $v \preceq w$. Assignments ρ and their term evaluations $\hat\rho$ are extended to formulas via the relation $w \Vdash_\rho \varphi$ defined by

$$w \Vdash_\rho \dot\bot \;:=\; \bot_w^{\mathcal{K}} \qquad w \Vdash_\rho \varphi \dot\rightarrow \psi \;:=\; \forall v \succeq w.\, v \Vdash_\rho \varphi \rightarrow v \Vdash_\rho \psi$$
$$w \Vdash_\rho P\,\mathbf{t} \;:=\; P_w^{\mathcal{K}}\,(\hat\rho \mathbin{@} \mathbf{t}) \qquad w \Vdash_\rho \dot\forall \varphi \;:=\; \forall a : D.\, w \Vdash_{a;\rho} \varphi$$

We write $\mathcal{K} \Vdash \varphi$ if $w \Vdash_\rho \varphi$ for all ρ and w. $\mathcal{K}$ is *standard* if $\bot_w^{\mathcal{K}}$ implies $\bot$ for all w and *exploding* if $\mathcal{K} \Vdash \dot\bot \dot\rightarrow \varphi$ for all φ. We write $\mathcal{T} \Vdash \varphi$ if $\mathcal{K} \Vdash_\rho \varphi$ for all standard $\mathcal{K}$ and ρ with $\mathcal{K} \Vdash_\rho \mathcal{T}$, and $\mathcal{T} \Vdash_e \varphi$ when relaxing to exploding models.

Note that standard models are exploding, hence $\mathcal{T} \Vdash_e \varphi$ implies $\mathcal{T} \Vdash \varphi$. Moreover, the monotonicity required for the predicate and falsity interpretations lifts to all formulas, i.e. $w \Vdash_\rho \varphi$ implies $v \Vdash_\rho \varphi$ whenever $w \preceq v$. This property together with the usual facts about the interaction of assignments and substitutions yields soundness:

Fact 21. *$\Gamma \vdash \varphi$ implies $\Gamma \Vdash_e \varphi$.*

Proof. By induction on $\Gamma \vdash \varphi$ and analogous to [14, Fact 3.34]. □

Turning to completeness, instead of showing that $\Gamma \Vdash_e \varphi$ implies $\Gamma \vdash \varphi$ directly, we follow Herbelin and Lee [24] and reconstruct a formal derivation in the normal sequent calculus LJT, hence implementing a cut-elimination procedure. LJT is defined by judgements $\Gamma \Rightarrow \varphi$ and $\Gamma\,;\psi \Rightarrow \varphi$ for a focused formula ψ:

$$\frac{}{\Gamma\,;\varphi \Rightarrow \varphi}\ \text{A} \qquad \frac{\Gamma\,;\varphi \Rightarrow \psi \quad \varphi \in \Gamma}{\Gamma \Rightarrow \psi}\ \text{C} \qquad \frac{\Gamma \Rightarrow \varphi \quad \Gamma\,;\psi \Rightarrow \theta}{\Gamma\,;\varphi \dot{\to} \psi \Rightarrow \theta}\ \text{IL}$$

$$\frac{\Gamma, \varphi \Rightarrow \psi}{\Gamma \Rightarrow \varphi \dot{\to} \psi}\ \text{IR} \qquad \frac{\Gamma\,;\varphi[t] \Rightarrow \psi}{\Gamma\,;\dot{\forall}\varphi \Rightarrow \psi}\ \text{AL} \qquad \frac{\uparrow\Gamma \Rightarrow \varphi}{\Gamma \Rightarrow \dot{\forall}\varphi}\ \text{AR} \qquad \frac{\Gamma \Rightarrow \dot{\bot}}{\Gamma \Rightarrow \varphi}\ \text{E}$$

Fact 22. *Every sequent $\Gamma \Rightarrow \varphi$ can be translated into a normal derivation $\Gamma \vdash \varphi$.*

Proof. By simultaneous induction on both forms of judgements, where every sequent $\Gamma\,;\psi \Rightarrow \varphi$ is translated to an implication from $\Gamma \vdash \psi$ to $\Gamma \vdash \varphi$. □

By the previous fact, completeness for LJT implies completeness for intuitionistic ND. The technique to establish completeness for Kripke semantics is based on universal models coinciding with intuitionistic provability. We in fact construct two syntactic Kripke models over the domain $\mathbb{T}$.

- An exploding model $\mathcal{U}$ on contexts s.t. $\Gamma \Vdash^{\mathcal{U}}_\sigma \varphi$ iff $\Gamma \Rightarrow \varphi[\sigma]$.
- A standard model $\mathcal{C}$ on consistent contexts s.t. $\Gamma \Vdash^{\mathcal{C}}_\sigma \varphi$ iff $\neg\neg(\Gamma \Rightarrow \varphi[\sigma])$.

These constructions are adaptions of those in [55], which in turn are based on the proof and comments in [24]. We begin with the exploding model $\mathcal{U}$.

Definition 23. The model $\mathcal{U}$ over the domain $\mathbb{T}$ of terms is defined on the contexts Γ preordered by inclusion $\subseteq$. Further, we set:

$$f^{\mathcal{U}}\,\mathbf{d} := f\,\mathbf{d} \qquad P^{\mathcal{U}}_\Gamma\,\mathbf{d} := \Gamma \Rightarrow P\,\mathbf{d} \qquad \bot^{\mathcal{U}}_\Gamma := \Gamma \Rightarrow \dot{\bot}$$

The desired properties of $\mathcal{U}$ can be derived from the next lemma, which takes the shape of a normalisation-by-evaluation procedure [3,10].

Lemma 24. *In the universal Kripke model $\mathcal{U}$ the following hold.*

1. $\Gamma \Vdash_\sigma \varphi \to \Gamma \Rightarrow \varphi[\sigma]$
2. $(\forall\, \Gamma' \psi.\ \Gamma \subseteq \Gamma' \to \Gamma'\,;\varphi[\sigma] \Rightarrow \psi \to \Gamma' \Rightarrow \psi) \to \Gamma \Vdash_\sigma \varphi$

Proof. We prove (1) and (2) at once by induction on φ generalising Γ and σ. We only discuss the case of implications $\varphi\dot{\to}\psi$ in full detail.

1. Assuming $\forall \Gamma'.\ \Gamma \subseteq \Gamma' \to \Gamma' \Vdash_\sigma \varphi \to \Gamma' \Vdash_\sigma \psi$, one has to derive that $\Gamma \Rightarrow (\varphi\dot{\to}\psi)[\sigma]$. Per (IR) and inductive hypothesis (2) for ψ it suffices to show $\Gamma, \varphi[\sigma] \Vdash_\sigma \psi$. Applying the inductive hypothesis (2) for φ and the assumption, it suffices to show that $\Gamma' ; \varphi[\sigma] \Rightarrow \theta[\sigma]$ implies $\Gamma' \Rightarrow \theta[\sigma]$ for any $\Gamma, \varphi[\sigma] \subseteq \Gamma'$ and θ, which holds per (C).
2. Assuming $\forall\, \Gamma'\, \theta.\ \Gamma \subseteq \Gamma' \to \Gamma' ; (\varphi\dot{\to}\psi)[\sigma] \Rightarrow \theta \to \Gamma' \Rightarrow \theta$ one has to deduce $\Gamma' \Vdash_\sigma \varphi$ entailing $\Gamma' \Vdash_\sigma \psi$ for any $\Gamma \subseteq \Gamma'$. Because of the inductive hypothesis (2) for ψ it suffices to show $\Delta ; \psi[\sigma] \Rightarrow \theta$ implying $\Delta \Rightarrow \theta$ for any $\Gamma' \subseteq \Delta$. By using the assumption, $\Delta \Rightarrow \theta$ reduces to $\Delta ; (\varphi\dot{\to}\psi)[\sigma] \Rightarrow \theta$. This follows by (IL), as the assumption $\Gamma' \Vdash_\sigma \varphi$ implies $\Delta \Rightarrow \varphi[\sigma]$ per inductive hypothesis (2). ◻

Corollary 25. *$\mathcal{U}$ is exploding and satisfies $\Gamma \Vdash_\sigma \varphi$ iff $\Gamma \Rightarrow \varphi[\sigma]$.*

Proof. Suppose that $\Gamma \Rightarrow \dot{\bot}$, then (2) of Lemma 24 yields that $\Gamma \Vdash_\sigma \varphi$ for arbitrary φ. Thus $\mathcal{U}$ is exploding. The claimed equivalence then follows by (1) of Lemma 24 and soundness of LJT. ◻

Being universal, $\mathcal{U}$ witnesses completeness for exploding Kripke models:

Fact 26. *1. $\Gamma \Vdash_e \varphi$ implies $\Gamma \Rightarrow \varphi$.*
2. In the $\to, \forall$-fragment, $\Gamma \Vdash \varphi$ implies $\Gamma \Rightarrow \varphi$.

Proof. 1. Since $\Gamma \Vdash^{\mathcal{U}}_{\mathsf{id}} \Gamma$ we have that $\Gamma \Vdash_e \varphi$ implies $\Gamma \Vdash^{\mathcal{U}}_{\mathsf{id}} \varphi$ and hence $\Gamma \Rightarrow \varphi$.
2. In the minimal fragment, $\dot{\bot}$ remains uninterpreted and hence imposes no condition on the models. Hence $\mathcal{U}$ yields the completeness in this case.

Before we move on to completeness for standard models, we illustrate how the previous fact already establishes the cut rule for LJT.

Lemma 27. *If $\Gamma \Rightarrow \varphi$ and $\Gamma ; \varphi \Rightarrow \psi$, then $\Gamma \Rightarrow \psi$.*

Proof. By the translation given in Fact 22, we obtain a derivation $\Gamma \vdash \psi$ from the two assumptions. This can be turned into $\Gamma \Rightarrow \psi$ using soundness (Fact 21) and completeness (Fact 26).

We now construct the universal standard model $\mathcal{C}$ as a refinement of $\mathcal{U}$. As standard models require that $\bot^{\mathcal{K}}_v$ implies $\bot$ for any v, the model $\mathcal{U}$ has to be restricted to the consistent contexts, those which do not prove $\dot{\bot}$.

Definition 28. The model $\mathcal{C}$ over the domain $\mathbb{T}$ of terms is defined on the consistent contexts $\Gamma \not\Rightarrow \dot{\bot}$ preordered by inclusion $\subseteq$. Further, we set:

$$f^{\mathcal{C}}\, \mathbf{d} := f\, \mathbf{d} \qquad P^{\mathcal{C}}_{\Gamma}\, \mathbf{d} := \neg\neg(\Gamma \Rightarrow P\, \mathbf{d}) \qquad \bot^{\mathcal{C}}_{\Gamma} := \bot$$

Note that $\mathcal{C}$ is obviously standard and that we weakened the interpretation of atoms to doubly negated provability. This admits the following normalisation-by-evaluation procedure for doubly negated sequents:

Lemma 29. *In the universal Kripke model $\mathcal{C}$ the following hold.*

1. $\Gamma \Vdash_\sigma \varphi \to \neg\neg(\Gamma \Rightarrow \varphi[\sigma])$
2. $(\forall\, \Gamma' \psi.\ \Gamma \subseteq \Gamma' \to \Gamma' \,;\varphi[\sigma] \Rightarrow \psi \to \neg\neg(\Gamma' \Rightarrow \psi)) \to \Gamma \Vdash_\sigma \varphi$

Proof. We prove (1) and (2) at once by induction on φ generalising Γ and σ. Most cases are completely analogous to those in Lemma 24. Therefore we only discuss the crucial case (1) for implications $\varphi \dot\to \psi$.

1. Assuming $\Gamma \Vdash_\sigma \varphi \dot\to \psi$ we need to derive $\neg\neg(\Gamma \Rightarrow \varphi[\sigma] \dot\to \psi[\sigma])$. So we assume $\neg(\Gamma \Rightarrow \varphi[\sigma] \dot\to \psi[\sigma])$ and derive a contradiction. Because of the negative goal, we may assume that either $\Gamma, \varphi[\sigma]$ is consistent or not. In the positive case, we proceed as in Lemma 24 since the extended context is a node in $\mathcal{C}$. On the other hand, if $\Gamma, \varphi[\sigma] \Rightarrow \dot\bot$, then $\Gamma, \varphi[\sigma] \Rightarrow \psi[\sigma]$ by (E) and hence $\Gamma \Rightarrow \varphi[\sigma] \dot\to \psi[\sigma]$ by (IR), contradicting the assumption. □

Corollary 30. *$\mathcal{C}$ satisfies $\Gamma \Vdash_\sigma \varphi$ iff $\neg\neg(\Gamma \Rightarrow \varphi[\sigma])$.*

Proof. The first direction is (1) of Lemma 29 and the converse follows with (2) since $\neg\neg(\Gamma \Rightarrow \varphi[\sigma])$ and $\Gamma' \,; \varphi[\sigma] \Rightarrow \psi$ for $\Gamma' \supseteq \Gamma$ together imply $\neg\neg(\Gamma' \Rightarrow \psi)$ via the cut rule established in Lemma 27. □

The advantage of the additional double negations is that, in contrast to the proof in [24], we only need a single application of stability to derive completeness. Thus we can prove the completeness of $\Gamma \vdash \varphi$ admissible in Section 4.

Fact 31. *1. $\Gamma \Vdash \varphi$ implies $\Gamma \Rightarrow \varphi$, provided that $\Gamma \Rightarrow \varphi$ is stable.*
2. $\Gamma \Vdash \varphi$ implies $\Gamma \vdash \varphi$, provided that $\Gamma \vdash \varphi$ is stable.

Proof. 1. Since $\Gamma \Vdash \varphi$ implies $\neg\neg(\Gamma \Rightarrow \varphi)$, we can conclude $\Gamma \Rightarrow \varphi$ per stability.
2. Since $\Gamma \Rightarrow \varphi$ iff $\Gamma \vdash \varphi$ per soundness and completeness (Facts 21 and 26). □

Conversely, unrestricted completeness requires the stability of classical ND.

Fact 32. *Completeness of $\Gamma \Rightarrow \varphi$ implies stability of $\Gamma \vdash_c \varphi$.*

Proof. Assume completeness of $\Gamma \Rightarrow \varphi$ and suppose $\neg\neg(\Gamma \vdash_c \varphi)$. We prove $\Gamma \vdash_c \varphi$, so it suffices to show $\Gamma, \dot\neg\varphi \vdash_c \dot\bot$. Employing a standard double negation translation φ^N on formulas φ, it is equivalent to establish $(\Gamma, \dot\neg\varphi)^N \Rightarrow \dot\bot$. Applying completeness, however, we may assume a standard model $\mathcal{K}$ with $\mathcal{K} \Vdash_\rho (\Gamma, \dot\neg\varphi)^N$ and derive a contradiction. Hence we conclude $\Gamma \vdash_c \varphi$ and so $\Gamma^N \Vdash \varphi^N$ from $\neg\neg(\Gamma \vdash_c \varphi)$ and soundness, in conflict to $\mathcal{K} \Vdash_\rho (\Gamma, \dot\neg\varphi)^N$. □

Thus, the completeness of intuitionistic ND is similar to the classical case.

Theorem 33. *1. Completeness of $\Gamma \vdash \varphi$ is equivalent to* $\mathsf{MP_L}$.
2. Completeness of $\mathcal{T} \vdash \varphi$ for enumerable $\mathcal{T}$ implies MP.
3. Completeness of $\mathcal{T} \vdash \varphi$ for arbitrary $\mathcal{T}$ implies EM.

4 On Markov's Principle

We show that the stability of $\Gamma \vdash_c \varphi$ and $\Gamma \vdash \varphi$ is equivalent to an object-level version of Markov's principle referencing procedures in a concrete model of computation. For formalisation purposes, we will use the call-by-value λ-calculus L [43,17] as model of computation. Since on paper the same proofs can be carried out for any model of computation we will not go into details of L. We only need two notions: first, L-enumerability [15, Definition 6], which is defined like synthetic enumerability, but where the enumerator is an L-computable function. Secondly, the halting problem for L, defined as $\mathcal{E}s :=$ *"the term s terminates"*.

We define the object-level Markov's principle $\mathsf{MP_L}$ as stability of $\mathcal{E}$:

$$\mathsf{MP_L} := \forall s.\ \neg\neg\mathcal{E}s \to \mathcal{E}s$$

$\mathsf{MP_L}$ can also be phrased similarly to MP with a condition on the sequence:

Lemma 34. ([17, Theorem 45]) $\mathsf{MP_L}$ *is equivalent to*

$$\forall f : \mathbb{N} \to \mathbb{B}.\ \mathsf{L}\text{-}\mathit{computable}\, f \to \neg\neg(\exists n.\, f\, n = \mathsf{tt}) \to \exists n.\, f\, n = \mathsf{tt}.$$

Corollary 35. MP *implies* $\mathsf{MP_L}$.

We show Lemma 8, i.e. that $\mathsf{MP_L}$ is equivalent to both the stability of $\vdash_c$ and $\vdash$ for finite contexts, thereby establishing that completeness of provability for standard Tarski and Kripke semantics for finite theories is equivalent to $\mathsf{MP_L}$.

Lemma 36. ([14, Fact 2.16]) *Let p and q be predicates. If p many-one reduces to q (i.e. $\exists f.\forall x.\, px \leftrightarrow q(fx)$, written $p \preceq q$) and q is stable, then p is stable.*

Thus, in order to prove the equivalence of the stability of $\mathcal{E}$, $\Gamma \vdash \varphi$, and $\Gamma \vdash_c \varphi$, it suffices to give many-one reductions between them. We start with the two simpler reductions:

Lemma 37. $\vdash_c \preceq \vdash$*, and thus stability of $\Gamma \vdash \varphi$ implies the stability of $\Gamma \vdash_c \varphi$.*

Proof. Using a standard double-negation translation proof. □

Lemma 38. $\mathcal{E} \preceq \vdash_c$*, and thus stability of $\Gamma \vdash_c \varphi$ implies* $\mathsf{MP_L}$.

Proof. $\mathcal{E}$ reduces to the halting problem of Turing machines [56], which reduces to the Post correspondence problem [13], which in turn reduces to $\vdash_c$ by adapting [14, Corollary 3.49]. □

Since $p \preceq \mathcal{E}$ for all L-enumerable predicates p [15, Theorem 7], it suffices to give an L-computable enumeration of type $\mathbb{N} \to \mathcal{L}(\mathbb{F})$ of provable formulas $\vdash \varphi$. Note that we continue to assume signatures to be (synthetically) enumerable and do *not* have to restrict to L-enumerability, which is enabled by the following signature extension lemma:

Lemma 39. *Let ι be an invertible embedding from Σ to Σ'. Then $\vdash \varphi$ over Σ if and only if $\vdash \iota\varphi$ over Σ', where $\iota\varphi$ is the recursive application of ι to formulas.*

Proof. $\Gamma \vdash \varphi \to \iota\Gamma \vdash \iota\varphi$ follows trivially by induction. For the inverse direction, we show that Kripke models M over Σ can be extended to Kripke models ιM over Σ s.t. $\rho, u \Vdash_M \varphi \leftrightarrow \rho, u \Vdash_{\iota M} \iota\varphi$. Then $\iota\Gamma \vdash \iota\varphi \to \Gamma \vdash \varphi$ follows from soundness and completness w.r.t. exploding models. □

Lemma 40. *$\Gamma \vdash \varphi$ is* L*-enumerable for any enumerable signature Σ.*

Proof. Since Σ is enumerable, it can be injectively embedded via ι into the maximal signature $\Sigma_{\max} := (\mathbb{N}^2, \mathbb{N}^2)$ where the arity functions are just the second projections. Since $\mathbb{N}^2$ is L-enumerable, terms and formulas over $\Sigma_{\max}$ are also L-enumerable, and thus provability over $\Sigma_{\max}$ is L-enumerable. By Lemma 39 we obtain that provability over Σ is L-enumerable. □

Corollary 41. *$\vdash \preceq \mathcal{E}$, and thus* $\mathsf{MP_L}$ *implies the stability of $\Gamma \vdash \varphi$.*

We conclude the section with observations on independence and admissible of several statements in Coq's type theory pCuIC. By *independence* of a statement P, we mean that neither P nor $\neg P$ is provable in pCuIC without assumptions. By *admissibility* of a statement $\forall x.\ P(x) \to Q(x)$ we mean that whenever $P(t)$ is provable in pCuIC for a concrete term t without assumptions, $Q(t)$ is as well. Pédrot and Tabareau [42] show MP independent (Corollary 41) and admissible (Theorem 33). This transports to $\mathsf{MP_L}$ as well as stability of deduction systems and completeness with respect to model-theoretic semantics.

Theorem 42. *The following are all independent and admissible in pCuIC:*

1. $\mathsf{MP_L}$
2. *Stability of both $\Gamma \vdash_c \varphi$ and $\Gamma \vdash \varphi$.*
3. *Completeness of $\mathcal{T} \vdash_c \varphi$ for enumerable $\mathcal{T}$ w.r.t. standard Tarski semantics.*
4. *Completeness of $\Gamma \vdash_c \varphi$ w.r.t. standard Tarski semantics.*
5. *Completeness of $\Gamma \vdash_c \varphi$ w.r.t. standard Tarski semantics.*

Proof. We exemplarily show (1) and (4), the other proofs are similar.

For (1), $\mathsf{MP_L}$ is consistent since it is a consequence of EM. Lemma 40 in [42] shows that no theory conservative over the calculus of inductive constructions (CIC) can prove both the independence of premise rule IP and MP, by turning these assumptions into a decider for the halting problem of the untyped term language of CIC. One can adapt the proof to show that pCuIC cannot prove both IP and $\mathsf{MP_L}$, by constructing a decider for the L-halting problem instead, which yields a contradiction as well. The admissibility of $\mathsf{MP_L}$ follows from the admissibility of MP since a single application of MP suffices to derive $\mathsf{MP_L}$.

For (4), independence follows directly from (1) and Theorem 15. For admissibility, assume that $\Gamma \vDash \varphi$ is provable in pCuIC. By Fact 13, $\neg\neg(\Gamma \vdash_c \varphi)$ is provable in pCuIC. Thus by (2), $\Gamma \vdash_c \varphi$ is provable in pCuIC. □

5 Algebraic Semantics

In contrast to the model-theoretic semantics discussed in Section 3, algebraic semantics are not based on models interpreting the non-logical symbols but on algebras suitable for interpreting the logical connectives of the syntax. A formula is valid if it is satisfied by all algebras and completeness follows from the observation that deduction systems have the corresponding algebraic structure. Following [47], we discuss complete Heyting and Boolean algebras coinciding with intuitionistic and classical ND, respectively. We consider all formulas $\varphi : \mathbb{F}$.

Definition 43. A *Heyting algebra* consists of a preorder $(\mathcal{H}, \leq)$ and operations

$$0 : \mathcal{H}, \qquad \sqcap : \mathcal{H} \to \mathcal{H} \to \mathcal{H}, \qquad \sqcup : \mathcal{H} \to \mathcal{H} \to \mathcal{H}, \qquad \Rightarrow : \mathcal{H} \to \mathcal{H} \to \mathcal{H}$$

for bottom, meet, join, and implication satisfying the following properties:

1. $0 \leq x$
2. $z \sqcap x \leq y \leftrightarrow z \leq x \Rightarrow y$
3. $z \leq x \land z \leq y \leftrightarrow z \leq x \sqcap y$
4. $x \leq z \land y \leq z \leftrightarrow x \sqcup y \leq z$

Moreover, $\mathcal{H}$ is *complete* if there is a constant $\bigsqcap : (\mathcal{H} \to \mathbb{P}) \to \mathcal{H}$ for arbitrary meets satisfying $(\forall y \in P.\, x \leq y) \leftrightarrow x \leq \bigsqcap P$. Then $\mathcal{H}$ also has arbitrary joins $\bigsqcup P := \bigsqcap(\lambda x.\, \forall y \in P.\, y \leq x)$ satisfying $(\forall y \in P.\, y \leq x) \leftrightarrow \bigsqcup P \leq x$.

Arbitrary meets and joins indexed by a function $F : I \to \mathcal{H}$ on a type I are defined by $\bigsqcap_i F\,i := \bigsqcap(\lambda x.\, \exists i.\, x = F\,i)$ and $\bigsqcup_i F\,i := \bigsqcup(\lambda x.\, \exists i.\, x = F\,i)$, respectively. As we do not require $\leq$ to be antisymmetric in order to avoid quotient constructions, we establish equational facts about Heyting algebras only up to equivalence $x \equiv y := x \leq y \land y \leq x$ rather than actual equality.

Note that every Heyting algebra embeds into its down set algebra consisting of the sets $x\Downarrow := \lambda y.\, y \leq x$. The *MacNeille completion* [37] adding arbitrary meets and joins is a refinement of this embedding.

Fact 44. *Every Heyting algebra $\mathcal{H}$ embeds into a complete Heyting algebra $\mathcal{H}_c$, i.e. there is a function $f : \mathcal{H} \to \mathcal{H}_c$ with $x \leq y \leftrightarrow f\,x \leq_c f\,y$ and:*

1. $f\,0 \equiv 0_c$
2. $f\,(x \Rightarrow y) \equiv f\,x \Rightarrow_c f\,y$
3. $f\,(x \sqcap y) \equiv f\,x \sqcap_c f\,y$
4. $f\,(x \sqcup y) \equiv f\,x \sqcup_c f\,y$

Proof. Given a set $X : \mathcal{H} \to \mathbb{P}$, we define the sets $\mathfrak{L}\,X := \lambda x.\, \forall y \in X.\, x \leq y$ of lower bounds and $\mathfrak{U}\,X := \lambda x.\, \forall y \in X.\, y \leq x$ of upper bounds of X. We say that a set X is down-complete if $\mathfrak{L}\,(\mathfrak{U}\,X) \subseteq X$. Note that in particular down sets $x\Downarrow$ are down-complete and that down-complete sets are downwards closed, i.e. satisfy $x \in X$ whenever $x \leq y$ for some $y \in X$.

Now consider the type $\mathcal{H}_c := \Sigma X.\, \mathfrak{L}\,(\mathfrak{U}\,X) \subseteq X$ of down-complete sets preordered by set inclusion $X \subseteq Y$. It is immediate by construction that the operation $\bigsqcap_c P := \bigcap P$ defines arbitrary meets in $\mathcal{H}_c$. Moreover, it is easily verified that further setting

$$0_c := 0\Downarrow \quad X \sqcap_c Y := X \cap Y \quad X \sqcup_c Y := \mathfrak{L}\,(\mathfrak{U}\,(X \cup Y)) \quad X \Rightarrow_c Y := \lambda x.\, \forall y \in X.\, x \sqcap y \in Y$$

turns $\mathcal{H}_c$ into a (hence complete) Heyting algebra. The only non-trivial case is implication, where $X \Rightarrow_c Y \equiv \bigsqcap_c(\lambda Z. \exists x \in X. Z \equiv (\lambda y. y \sqcap x \in Y))$ is a helpful characterisation to show that $X \Rightarrow_c Y$ is down-complete whenever Y is.

Finally, $x\Downarrow$ clearly is a structure preserving embedding as specified. □

We now define how formulas can be evaluated in a complete Heyting algebra.

Definition 45. Given a complete Heyting algebra $\mathcal{H}$ we extend interpretations $[\![\_]\!] : \forall P : \mathcal{P}_\Sigma. \mathbb{T}^{|P|} \to \mathcal{H}$ of atoms to formulas using size recursion by

$$[\![\dot\bot]\!] := 0 \qquad [\![\varphi \dot\wedge \psi]\!] := [\![\varphi]\!] \sqcap [\![\psi]\!] \qquad [\![\dot\forall\varphi]\!] := \textstyle\bigsqcap_t [\![\varphi[t]]\!]$$
$$[\![\varphi \dot\to \psi]\!] := [\![\varphi]\!] \Rightarrow [\![\psi]\!] \qquad [\![\varphi \dot\vee \psi]\!] := [\![\varphi]\!] \sqcup [\![\psi]\!] \qquad [\![\dot\exists\varphi]\!] := \textstyle\bigsqcup_t [\![\varphi[t]]\!]$$

and to contexts by $[\![\Gamma]\!] := \bigsqcap \lambda x. \exists \varphi \in \Gamma. x = [\![\varphi]\!]$. A formula φ is *valid in* $\mathcal{H}$ whenever $x \leq [\![\varphi]\!]$ for all $x : \mathcal{H}$.

We first show that intuitionistic ND is sound for this semantics.

Fact 46. *$\Gamma \vdash \varphi$ implies $\forall \sigma. [\![\Gamma[\sigma]]\!] \leq [\![\varphi[\sigma]]\!]$ in every complete Heyting algebra.*

Proof. By induction on $\Gamma \vdash \varphi$, all cases but (DE) and (EE) are trivial. □

Corollary 47. *$\Gamma \vdash \varphi$ implies $[\![\Gamma]\!] \leq [\![\varphi]\!]$ in every complete Heyting algebra.*

Next turning to completeness, a strategy reminiscent to the case of Kripke semantics can be employed by exhibiting a universal structure, the so-called *Lindenbaum algebra*, that exactly coincides with provability.

Fact 48. *The type $\mathbb{F}$ of formulas together with the preorder $\varphi \vdash \psi$ and the logical connectives as corresponding algebraic operations forms a Heyting algebra.*

We write $\mathcal{L}$ for the Lindenbaum algebra (Fact 48) and $\mathcal{L}_c$ for its MacNeille completion (Fact 44). Formulas are evaluated in $\mathcal{L}_c$ according to Definition 45 using the syntactic atom interpretation $[\![P\,\mathbf{t}]\!] := (P\,\mathbf{t}\,)\Downarrow$.

Lemma 49. *Evaluating φ in $\mathcal{L}_c$ yields the set of all ψ with $\psi \vdash \varphi$, i.e. $[\![\varphi]\!] \equiv \varphi\Downarrow$.*

Proof. By size induction on φ. The case for atoms is by construction and the cases for all connectives but the quantifiers are immediate since $\Downarrow$ preserves the structure of $\mathcal{L}$ as specified in Fact 44. □

Theorem 50. *If φ is valid in every complete Heyting algebra, then $\vdash \varphi$.*

Proof. If φ is valid, then Lemma 49 implies that $\psi \vdash \varphi$ forall ψ. By for instance choosing $\psi := \dot\bot \dot\to \dot\bot$ we can derive $\vdash \varphi$ since $\vdash \dot\bot \dot\to \dot\bot$. □

A Heyting algebra is *Boolean* if it satisfies $(x \Rightarrow y) \Rightarrow x \leq x$ for all x and y.

Theorem 51. *If φ is valid in every complete Boolean algebra, then $\vdash_c \varphi$.*

Proof. Analogous to the intuitionistic case, using that the Lindenbaum algebra over $\varphi \vdash_c \psi$ and hence its MacNeille completion are Boolean. □

6 Game Semantics

Dialogues are games modeling a proponent defending the validity of a formula against an opponent. In the terminology of Felscher [11], the dialogues we consider in this section are the intuitionistic E-dialogues, generalised over their local rules $(\mathbb{F}, \mathbb{F}^a, \mathcal{A}, \rhd, \mathcal{D}_{-})$. Given abstract types for formulas $\mathbb{F}$ and attacks $\mathcal{A}$, the relation $a \,|\, \psi \rhd \varphi$ states that a player may attack $\varphi : \mathbb{F}$ with $a : \mathcal{A}$ by possibly admitting a unique $\psi : \mathcal{O}(\mathbb{F})$. If $\psi = \emptyset$, no admission is made. Each $a : \mathcal{A}$ has an associated set $\mathcal{D}_a$ of formulas that may be admitted to fend off a. Special rules restrict when the proponent may admit atomic formulas, members of the set $\mathbb{F}^a$. We write $a \rhd \varphi$ for $a \,|\, \emptyset \rhd \varphi$. The local rules of first-order logic are given below with atomic formulas $\mathbb{F}^a := \{P\,\mathbf{t} \mid P : \mathcal{P}_\Sigma\}$.

$$a_{\dot\vee} \rhd \varphi \dot\vee \psi \quad \mathcal{D}_{a_{\dot\vee}} = \{\varphi, \psi\} \qquad a_{\dot\rightarrow} \,|\, \ulcorner\varphi\urcorner \rhd \varphi \dot\rightarrow \psi \quad \mathcal{D}_{a_{\dot\rightarrow}} = \{\psi\} \qquad a_L \rhd \varphi \dot\wedge \psi \quad \mathcal{D}_{a_L} = \{\varphi\}$$

$$a_t \rhd \dot\forall \varphi \quad \mathcal{D}_{a_t} = \{\varphi[t]\} \qquad a_{\dot\bot} \rhd \dot\bot \quad \mathcal{D}_{a_{\dot\bot}} = \{\} \qquad a_R \rhd \varphi \dot\wedge \psi \quad \mathcal{D}_{a_R} = \{\psi\}$$

$$a_{\dot\exists} \rhd \dot\exists \varphi \quad \mathcal{D}_{a_{\dot\exists}} = \{\varphi[t] \mid t : \mathbb{T}\}$$

In contrast to their usual presentation as sequences of alternating moves, we define dialogues as state transition systems over elements (A_o, c) of the type $\mathcal{L}(\mathbb{F}) \times \mathcal{A}$ containing the opponent's admissions (A_o) and last attack (c). The proponent opens each round by picking a move. She can defend against the opponent's attack c by admitting a justified defense formula $\varphi \in \mathcal{D}_c$, meaning $\varphi \in \mathbb{F}^a$ implies $\varphi \in A_o$. Alternatively, she can launch an attack a against any of the opponent's admissions if the admission resulting from a is justified.

$$\frac{\varphi \in \mathcal{D}_c \quad \text{justified } A_o\, \varphi}{(A_o, c) \rightsquigarrow_p \varphi}\ \text{PD} \qquad \frac{\varphi \in A_o \quad a \,|\, \psi \rhd \varphi \quad \text{justified } A_o\, \psi}{(A_o, c) \rightsquigarrow_p (a, \varphi)}\ \text{PA}$$

Given such a move m, the opponent reacts to it by transforming the state s into s' (written as $s\,; m \rightsquigarrow_o s'$). The opponent may attack the proponent's defense formula (OA), defend against her attack (OD) or counter her attack by attacking her admission (OC). We define $\ulcorner\varphi\urcorner :: A := \varphi :: A$ and $\emptyset :: A := A$.

$$\frac{c' \,|\, \psi \rhd \varphi}{(A_o, c)\,; \varphi \rightsquigarrow_o (\psi :: A_o, c')}\ \text{OA} \qquad \frac{\psi \in \mathcal{D}_a}{(A_o, c)\,; (a, \varphi) \rightsquigarrow_o (\psi :: A_o, c)}\ \text{OD}$$

$$\frac{a \,|\, \ulcorner\psi\urcorner \rhd \varphi \quad c' \,|\, \theta \rhd \psi}{(A_o, c)\,; (a, \varphi) \rightsquigarrow_o (\theta :: A_o, c')}\ \text{OC}$$

A formula φ is then considered E-valid if it is non-atomic and for all $c \,|\, \psi \rhd \varphi$, there is a winning strategy $\text{Win}\,([\psi], c)$ as defined below.

$$\frac{s \rightsquigarrow_p m \quad \forall s'.\ s\,; m \rightsquigarrow_o s' \rightarrow \text{Win}\, s'}{\text{Win}\, s}$$

Following the strategy of [48], we first prove the soundness and completeness of the sequent calculus LJD which is defined in terms of the same notions as the dialogues. Indeed, as witnessed in the proofs of soundness and completeness, derivations of LJD are isomorphic to winning strategies, the R- and L-rule corresponding to a proponent defense and attack, their premises matching the possible opponent responses to each move. The statement $\Gamma \Rightarrow_D \mathcal{S}$ means that the context Γ entails the disjunction of the formulas contained in the set $\mathcal{S}$.

$$\frac{\varphi \in \mathcal{S} \quad \text{justified } \Gamma\, \varphi \quad \forall a \,|\, \psi \rhd \varphi.\ \Gamma, \psi \Rightarrow_D \mathcal{D}_a}{\Gamma \Rightarrow_D \mathcal{S}}\ \text{R}$$

$$\frac{\varphi \in \Gamma \quad \text{justified } \Gamma\, \psi \quad a \,|\, \psi \rhd \varphi \quad \forall\, \theta \in \mathcal{D}_a.\ \Gamma, \theta \Rightarrow_D \mathcal{S} \quad \forall a' \,|\, \tau \rhd \psi.\ \Gamma, \tau \Rightarrow_D \mathcal{D}_{a'}}{\Gamma \Rightarrow_D \mathcal{S}}\ \text{L}$$

Theorem 52. *Any formula φ is E-valid if and only if one can derive $[] \Rightarrow_D \{\varphi\}$.*

Proof. $\text{Win}\,(A_o, c) \to A_o \Rightarrow_D \mathcal{D}_c$ holds per induction on $\text{Win}\,(A_o, c)$. From this, completeness follows with an application of the R-rule, transforming $\text{Win}\,([\psi], c)$ for any $c \,|\, \psi \rhd \varphi$ into $[\psi] \Rightarrow_D \mathcal{D}_c$. Soundness can be proven symmetrically. □

To arrive at a more traditional soundness and completeness result, we show that one can translate between derivations in LJD and the intuitionistic sequent calculus LJ deriving sequents $\Gamma \Rightarrow_J \varphi$ as defined in Definition 58 of Appendix A.

Lemma 53. *One can derive $\Gamma \Rightarrow_D \{\varphi\}$ if and only if one can derive $\Gamma \Rightarrow_J \varphi$.*

Proof. Completeness is generalised as below and shown per induction on $\Gamma \Rightarrow_D \mathcal{S}$:

$$\Gamma \Rightarrow_D \mathcal{S} \ \to\ \forall \varphi.\ (\forall \psi, \Gamma \subseteq \Gamma'.\ \Gamma' \Rightarrow_J \psi \to \Gamma' \Rightarrow_J \varphi) \to \Gamma \Rightarrow_J \varphi$$

Soundness follows analogously from $\Gamma \Rightarrow_J \varphi \to \forall \sigma.\ \Gamma[\sigma] \Rightarrow_D \{\varphi[\sigma]\}$. □

Corollary 54. *Any formula φ is E-valid if and only if one can derive $[] \Rightarrow_J \varphi$.*

7 Discussion

We have analysed the completeness of common deduction systems for first-order logic with regards to various explanations of logical validity. Model-theoretic semantics are the most direct implementation of the idea that terms represent objects of a domain of discourse. Particularly in a formal meta-theory such as constructive type theory, model-theoretic completeness justifies the common practice to verify consequences of a first-order axiomatisation by studying models satisfying corresponding meta-level axioms. However, model-theoretic semantics typically do not admit constructive completeness and, if not generalised to exploding models, require Markov's principle as soon as falsity is involved. Contrarily, evidence for the validity of a first-order formula in algebraic semantics and game semantics can be algorithmically transformed into syntactic derivations.

Of course, there are more semantics than the selection studied in this paper. For instance, there are hybrid variants such as interpreting both terms in a model and logical operations in an algebra, or dialogues with atomic formulas represented as underlying games. More generally, there are entirely different approaches like realisability semantics or proof-theoretic semantics, all coming with interesting completeness problems worth analysing in constructive type theory. Ideas for future work are outlined after a brief summary of related work.

7.1 Related Work

Our analysis of completeness in constructive type theory was motivated by previous work [14], carried out in Wehr's bachelor's thesis [55], and is directly influenced by multiple prior works. In their analysis of Henkin's proof, Herbelin and Ilik [23] give a constructive model existence proof and the constructivisation of completeness via exploding models. Herbelin and Lee [24] demonstrate the constructive Kripke completeness proof for minimal models and mention how to extend the approach to standard and exploding models. Scott [47] establishes completeness of free logic interpreted in a hybrid semantics comprising model-theoretic and algebraic components. Urzyczyn and Sørensen [48] give a proof of dialogue completeness via generalised dialogues for classical propositional logic.

The first proof that the completeness of intuitionistic first-order logic entails Markov's principle was given by Kreisel [29], although he attributes the proof idea to Gödel. The proof has since inspired a range of works deriving related non-constructivity results for different kinds of completeness [30,34,39,41,40]. By almost exclusively focusing our analysis on the $\forall, \rightarrow, \bot$-fragment, we did not concern ourselves with the contributions of $\exists$ and $\vee$ to the non-constructivity of completeness. Krivtsov's [32,33] work has the exact opposite focus: His analysis reveals that completeness with regards to exploding Tarski and Beth models, for full classical and intuitionistic first-order logic, respectively, are equivalent to the weak fan theorem. Another noteworthy work is that of Berardi [2], who analyses which abstract notions of models admit constructive completeness.

The completeness of first-order logic has been formalised in many interactive theorem provers such as Isabelle/HOL [4,44,45], NuPRL [6,53], Mizar [5], Lean [19], and Coq [24,25]. Among them, [6] and [25] share our focus on the constructivity of completeness. Constable and Bickford [6] give a constructive proof of completeness for the BHK-realisers of full intuitionistic first-order logic in NuPRL. Their proof is fully constructive when realisers are restricted to be normal terms, requiring Brouwer's fan theorem when lifting that restriction. In his PhD thesis [25], Ilik formalises multiple constructive proofs of first-order completeness in Coq. Especially noteworthy are the highly non-standard, constructivised Kripke models for full classical and intuitionistic first-order logic he presents in Chapters 2 and 3.

7.2 Future Work

We plan to further extend our constructive analysis and Coq library to all logical connectives and to uncountable signatures, both relying on additional logical assumptions. Subsequently, it would be interesting to study other aspects of model theory in the setting of constructive type theory, for instance the Löwenheim-Skolem theorems or first-order axiomatisations of arithmetic and set theory. Another idea is to analyse the completeness of second-order logic interpreted in Henkin semantics, as this formalism suffices to express the higher-order axiomatisation of set theory studied in [28]. Lastly, we conjecture that $\mathsf{MP_L}$ is strictly weaker than MP, but are not aware of a proof.

Acknowledgments We thank Kathrin Stark for adapting Autosubst according to our needs, Fabian Kunze for helping with technicalities during the formalisation of Corollary 41, and Hugo Herbelin for fruitful discussion and pointers to relevant work.

References

1. A. Bauer. First steps in synthetic computability theory. *Electronic Notes in Theoretical Computer Science*, 155:5 – 31, 2006. Proceedings of the 21st Annual Conference on Mathematical Foundations of Programming Semantics (MFPS XXI).
2. S. Berardi. Intuitionistic completeness for first order classical logic. *The Journal of Symbolic Logic*, 64(1):304–312, 1999.
3. U. Berger and H. Schwichtenberg. An inverse of the evaluation functional for typed lambda-calculus. In *[1991] Proceedings Sixth Annual IEEE Symposium on Logic in Computer Science*, pages 203–211. IEEE, 1991.
4. J. C. Blanchette, A. Popescu, and D. Traytel. Unified classical logic completeness. In *International Joint Conference on Automated Reasoning*, pages 46–60. Springer, 2014.
5. P. Braselmann and P. Koepke. Gödel's completeness theorem. *Formalized Mathematics*, 13(1):49–53, 2005.
6. R. Constable and M. Bickford. Intuitionistic completeness of first-order logic. *Annals of Pure and Applied Logic*, 165(1):164–198, 2014.
7. T. Coquand and B. Mannaa. The Independence of Markov's Principle in Type Theory. *Logical Methods in Computer Science ; Volume 13*, page Issue 3 ; 18605974, 2017. arXiv: 1602.04530.
8. N. G. de Bruijn. Lambda calculus notation with nameless dummies, a tool for automatic formula manipulation, with application to the Church-Rosser theorem. *Indagationes Mathematicae (Proceedings)*, 75(5):381–392, Jan. 1972.
9. B. Delaware, B. C. d S Oliveira, and T. Schrijvers. Meta-theory à la carte. In *ACM SIGPLAN Notices*, volume 48, pages 207–218. ACM, 2013.
10. P. Dybjer and A. Filinski. Normalization and partial evaluation. In *International Summer School on Applied Semantics*, pages 137–192. Springer, 2000.
11. W. Felscher. Dialogues, strategies, and intuitionistic provability. *Annals of pure and applied logic*, 28(3):217–254, 1985.
12. Y. Forster, E. Heiter, D. Kirst, D. Larchey-Wendling, and G. Smolka. A library of formalised undecidable problems in Coq.
13. Y. Forster, E. Heiter, and G. Smolka. Verification of PCP-related computational reductions in Coq. In *International Conference on Interactive Theorem Proving*, pages 253–269. Springer, 2018.
14. Y. Forster, D. Kirst, and G. Smolka. On synthetic undecidability in Coq, with an application to the Entscheidungsproblem. In *International Conference on Certified Programs and Proofs*, pages 38–51. ACM, 2019.
15. Y. Forster and F. Kunze. Verified extraction from Coq to a lambda-calculus. In *Coq Workshop*, volume 2016, 2016.
16. Y. Forster and F. Kunze. A Certifying Extraction with Time Bounds from Coq to Call-By-Value Lambda Calculus. In J. Harrison, J. O'Leary, and A. Tolmach, editors, *10th International Conference on Interactive Theorem Proving*, volume 141 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 17:1–17:19, Dagstuhl, Germany, 2019. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

17. Y. Forster and G. Smolka. Weak call-by-value lambda calculus as a model of computation in Coq. In *International Conference on Interactive Theorem Proving*, pages 189–206. Springer, 2017.
18. K. Gödel. Die Vollständigkeit der Axiome des logischen Funktionenkalküls. *Monatshefte für Mathematik und Physik*, 37:349–360, 1930.
19. J. Han and F. van Doorn. A formalization of forcing and the consistency of the failure of the continuum hypothesis. In *International Conference on Interactive Theorem Proving*. Springer, 2019.
20. G. Hasenjaeger. Eine Bemerkung zu Henkin's Beweis für die Vollständigkeit des Prädikatenkalküls der Ersten Stufe. *The Journal of Symbolic Logic*, 18(1):42–48, 1953.
21. L. Henkin. The Completeness of the First-Order Functional Calculus. *The Journal of Symbolic Logic*, 14(3):159–166, 1949.
22. L. Henkin. The completeness of the first-order functional calculus. *The Journal of Symbolic Logic*, 14(3):159–166, 1949.
23. H. Herbelin and D. Ilik. An analysis of the constructive content of Henkin's proof of Gödel's completeness theorem. Draft, 2016.
24. H. Herbelin and G. Lee. Forcing-based cut-elimination for Gentzen-style intuitionistic sequent calculus. In *International Workshop on Logic, Language, Information, and Computation*, pages 209–217. Springer, 2009.
25. D. Ilik. *Constructive completeness proofs and delimited control*. PhD thesis, Ecole Polytechnique X, 2010.
26. H. Ishihara. Reverse Mathematics in Bishop's Constructive Mathematics. *Philosophia Scientae*, pages 43–59, 2006.
27. S. Keuchel and T. Schrijvers. Generic datatypes à la carte. In *ACM SIGPLAN Workshop on Generic Programming*, pages 13–24. ACM, 2013.
28. D. Kirst and G. Smolka. Categoricity results and large model constructions for second-order zf in dependent type theory. *Journal of Automated Reasoning*, 2018.
29. G. Kreisel. On weak completeness of intuitionistic predicate logic. *The Journal of Symbolic Logic*, 27(2):139–158, 1962.
30. G. Kreisel and A. S. Troelstra. Formal systems for some branches of intuitionistic analysis. *Annals of mathematical logic*, 1(3):229–387, 1970.
31. J.-L. Krivine. Une preuve formelle et intuitionniste du théorème de complétude de la logique classique. *Bulletin of Symbolic Logic*, 2(4):405–421, 1996.
32. V. N. Krivtsov. An intuitionistic completeness theorem for classical predicate logic. *Studia Logica*, 96(1):109–115, 2010.
33. V. N. Krivtsov. Semantical completeness of first-order predicate logic and the weak fan theorem. *Studia Logica*, 103(3):623–638, 2015.
34. D. Leivant. Failure of completeness properties of intuitionistic predicate logic for constructive models. *Annales scientifiques de l'Université de Clermont. Mathématiques*, 60(13):93–107, 1976.
35. P. Lorenzen. Logik und Agon. In *Atti del XII Congresso Internazionale di Filosofia*, volume 4, pages 187–194, 1960.
36. P. Lorenzen. Ein dialogisches Konstruktivitätskriterium. In *Proceedings of the Symposium on Foundations of Mathematics (Warsaw, 2 – 9 September 1959)*, pages 193–200, 1961.
37. H. M. MacNeille. Partially ordered sets. *Transactions of the American Mathematical Society*, 42(3):416–460, 1937.
38. B. Mannaa and T. Coquand. The independence of Markov's principle in type theory. *Logical Methods in Computer Science*, 13, 2017.

39. C. McCarty. Constructive validity is nonarithmetic. *The Journal of Symbolic Logic*, 53:1036–1041, 1988.

40. C. McCarty. Completeness and incompleteness for intuitionistic logic. *The Journal of Symbolic Logic*, 73(4):1315–1327, 2008.

41. D. C. McCarty et al. Incompleteness in intuitionistic metamathematics. *Notre Dame journal of formal logic*, 32(3):323–358, 1991.

42. P.-M. Pédrot and N. Tabareau. Failure is not an option. In *European Symposium on Programming*, pages 245–271. Springer, 2018.

43. G. D. Plotkin. Call-by-Name, Call-by-Value and the lambda-Calculus. *Theor. Comput. Sci.*, 1(2):125–159, 1975.

44. T. Ridge and J. Margetson. A mechanically verified, sound and complete theorem prover for first order logic. In *International Conference on Theorem Proving in Higher Order Logics*, pages 294–309. Springer, 2005.

45. A. Schlichtkrull. Formalization of the resolution calculus for first-order logic. *Journal of Automated Reasoning*, 61(1-4):455–484, 2018.

46. G. F. Schumm. A Henkin-style completeness proof for the pure implicational calculus. *Notre Dame J. Formal Logic*, 16(3):402–404, July 1975.

47. D. Scott. The algebraic interpretation of quantifiers: Intuitionistic and classical. In V. M. A. Ehrenfeucht and M. Srebrny, editors, *Andrzej Mostowski and Foundational Studies*. IOS Press, 2008.

48. M. H. Sørensen and P. Urzyczyn. Sequent calculus, dialogues, and cut elimination. *Reflections on Type Theory, λ-Calculus, and the Mind*, pages 253–261, 2007.

49. M. Sozeau and C. Mangin. Equations reloaded: high-level dependently-typed functional programming and proving in coq. *Proceedings of the ACM on Programming Languages*, 3(ICFP):86, 2019.

50. K. Stark, S. Schäfer, and J. Kaiser. Autosubst 2: reasoning with multi-sorted de Bruijn terms and vector substitutions. In *International Conference on Certified Programs and Proofs*, pages 166–180. ACM, 2019.

51. The Coq Proof Assistant. `http://coq.inria.fr`, 2019.

52. A. Timany and M. Sozeau. Cumulative Inductive Types In Coq. In H. Kirchner, editor, *International Conference on Formal Structures for Computation and Deduction*, volume 108 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 29:1–29:16, Dagstuhl, Germany, 2018. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

53. J. Underwood. Aspects of the computational content of proofs. Technical report, Cornell University, 1994.

54. W. Veldman. An intuitiomstic completeness theorem for intuitionistic predicate logic 1. *The Journal of Symbolic Logic*, 41(1):159–166, 1976.

55. D. Wehr. A Constructive Analysis of First-Order Completeness Theorems in Coq, 2019. Bachelor's thesis, Saarland University.

56. M. Wuttke. Verified Programming Of Turing Machines In Coq, 2018. Bachelor's thesis, Saarland University.

A Overview of Deduction Systems

Definition 55. Intuitionistic natural deduction is defined by the following rules:

$$\frac{\varphi \in \Gamma}{\Gamma \vdash \varphi}\,\text{C} \qquad \frac{\Gamma \vdash \dot\bot}{\Gamma \vdash \varphi}\,\text{E} \qquad \frac{\Gamma, \varphi \vdash \psi}{\Gamma \vdash \varphi \dot\to \psi}\,\text{II} \qquad \frac{\Gamma \vdash \varphi \dot\to \psi \quad \Gamma \vdash \varphi}{\Gamma \vdash \varphi}\,\text{IE}$$

$$\frac{\Gamma \vdash \varphi \quad \Gamma \vdash \psi}{\Gamma \vdash \varphi \dot\wedge \psi}\,\text{CI} \qquad \frac{\Gamma \vdash \varphi \dot\wedge \psi}{\Gamma \vdash \varphi}\,\text{CE}_1 \qquad \frac{\Gamma \vdash \varphi \dot\wedge \psi}{\Gamma \vdash \psi}\,\text{CE}_2$$

$$\frac{\Gamma \vdash \varphi}{\Gamma \vdash \varphi \dot\vee \psi}\,\text{DI}_1 \qquad \frac{\Gamma \vdash \psi}{\Gamma \vdash \varphi \dot\vee \psi}\,\text{DI}_2 \qquad \frac{\Gamma \vdash \varphi \dot\vee \psi \quad \Gamma, \varphi \vdash \theta \quad \Gamma, \psi \vdash \theta}{\Gamma \vdash \theta}\,\text{DE}$$

$$\frac{\uparrow\!\Gamma \vdash \varphi}{\Gamma \vdash \dot\forall \varphi}\,\text{AI} \qquad \frac{\Gamma \vdash \dot\forall \varphi}{\Gamma \vdash \varphi[t]}\,\text{AE} \qquad \frac{\Gamma \vdash \varphi[t]}{\Gamma \vdash \dot\exists \varphi}\,\text{EI} \qquad \frac{\Gamma \vdash \dot\exists \varphi \quad \uparrow\!\Gamma, \varphi \vdash \uparrow\!\psi}{\Gamma \vdash \psi}\,\text{EE}$$

We write $\vdash \varphi$ whenever φ is intuitionistically provable from the empty context.

Definition 56. Classical natural deduction is defined by the following rules:

$$\frac{\varphi \in \Gamma}{\Gamma \vdash_c \varphi}\,\text{C} \qquad \frac{\Gamma \vdash_c \dot\bot}{\Gamma \vdash_c \varphi}\,\text{E} \qquad \frac{\Gamma, \varphi \vdash_c \psi}{\Gamma \vdash_c \varphi \dot\to \psi}\,\text{II} \qquad \frac{\Gamma \vdash_c \varphi \dot\to \psi \quad \Gamma \vdash_c \varphi}{\Gamma \vdash_c \varphi}\,\text{IE}$$

$$\frac{\Gamma \vdash_c \varphi \quad \Gamma \vdash_c \psi}{\Gamma \vdash_c \varphi \dot\wedge \psi}\,\text{CI} \qquad \frac{\Gamma \vdash_c \varphi \dot\wedge \psi}{\Gamma \vdash_c \varphi}\,\text{CE}_1 \qquad \frac{\Gamma \vdash_c \varphi \dot\wedge \psi}{\Gamma \vdash_c \psi}\,\text{CE}_2$$

$$\frac{\Gamma \vdash_c \varphi}{\Gamma \vdash_c \varphi \dot\vee \psi}\,\text{DI}_1 \qquad \frac{\Gamma \vdash_c \psi}{\Gamma \vdash_c \varphi \dot\vee \psi}\,\text{DI}_2 \qquad \frac{\Gamma \vdash_c \varphi \dot\vee \psi \quad \Gamma, \varphi \vdash_c \theta \quad \Gamma, \psi \vdash_c \theta}{\Gamma \vdash_c \theta}\,\text{DE}$$

$$\frac{\uparrow\!\Gamma \vdash_c \varphi}{\Gamma \vdash_c \dot\forall \varphi}\,\text{AI} \qquad \frac{\Gamma \vdash_c \dot\forall \varphi}{\Gamma \vdash_c \varphi[t]}\,\text{AE} \qquad \frac{\Gamma \vdash_c \varphi[t]}{\Gamma \vdash_c \dot\exists \varphi}\,\text{EI} \qquad \frac{\Gamma \vdash_c \dot\exists \varphi \quad \uparrow\!\Gamma, \varphi \vdash_c \uparrow\!\psi}{\Gamma \vdash_c \psi}\,\text{EE}$$

$$\frac{}{\Gamma \vdash_c ((\varphi \dot\to \psi) \dot\to \varphi) \dot\to \varphi}\,\text{P}$$

We write $\vdash_c \varphi$ whenever φ is classically provable from the empty context.

Definition 57. The intuitionistic sequent calculus LJT is defined as follows:

$$\frac{}{\Gamma\,;\varphi \Rightarrow \varphi}\,\text{A} \qquad \frac{\Gamma\,;\varphi \Rightarrow \psi \quad \varphi \in \Gamma}{\Gamma \Rightarrow \psi}\,\text{C} \qquad \frac{\Gamma \Rightarrow \varphi \quad \Gamma\,;\psi \Rightarrow \theta}{\Gamma\,;\varphi \dot\to \psi \Rightarrow \theta}\,\text{IL}$$

$$\frac{\Gamma, \varphi \Rightarrow \psi}{\Gamma \Rightarrow \varphi \dot\to \psi}\,\text{IR} \qquad \frac{\Gamma\,;\varphi[t] \Rightarrow \psi}{\Gamma\,;\dot\forall \varphi \Rightarrow \psi}\,\text{AL} \qquad \frac{\uparrow\!\Gamma \Rightarrow \varphi}{\Gamma \Rightarrow \dot\forall \varphi}\,\text{AR} \qquad \frac{\Gamma \Rightarrow \dot\bot}{\Gamma \Rightarrow \varphi}\,\text{E}$$

Definition 58. The intuitionistic sequent calculus LJ is defined as follows:

$$\frac{}{\Gamma, \varphi \Rightarrow_J \varphi}\ \text{A} \qquad \frac{\Gamma, \varphi, \varphi \Rightarrow_J \psi}{\Gamma, \varphi \Rightarrow_J \psi}\ \text{C} \qquad \frac{\Gamma \Rightarrow_J \psi}{\Gamma, \varphi \Rightarrow_J \psi}\ \text{W}$$

$$\frac{\Gamma, \psi, \varphi, \Gamma' \Rightarrow_J \theta}{\Gamma, \varphi, \psi, \Gamma' \Rightarrow_J \theta}\ \text{P} \qquad \frac{\Gamma \Rightarrow_J \dot{\bot}}{\Gamma \Rightarrow_J \varphi}\ \text{E} \qquad \frac{\Gamma \Rightarrow_J \varphi \quad \Gamma, \psi \Rightarrow_J \theta}{\Gamma, \varphi \dot{\to} \psi \Rightarrow_J \theta}\ \text{IL}$$

$$\frac{\Gamma, \varphi \Rightarrow_J \psi}{\Gamma \Rightarrow_J \varphi \dot{\to} \psi}\ \text{IR} \qquad \frac{\Gamma, \varphi, \psi \Rightarrow_J \theta}{\Gamma, \varphi \dot{\wedge} \psi \Rightarrow_J \theta}\ \text{CL} \qquad \frac{\Gamma \Rightarrow_J \varphi \quad \Gamma \Rightarrow_J \psi}{\Gamma \Rightarrow_J \varphi \dot{\wedge} \psi}\ \text{CR}$$

$$\frac{\Gamma, \varphi \Rightarrow_J \theta \quad \Gamma, \psi \Rightarrow_J \theta}{\Gamma, \varphi \dot{\vee} \psi \Rightarrow_J \theta}\ \text{DL} \qquad \frac{\Gamma \Rightarrow_J \varphi}{\Gamma \Rightarrow_J \varphi \dot{\vee} \psi}\ \text{DR}_1 \qquad \frac{\Gamma \Rightarrow_J \psi}{\Gamma \Rightarrow_J \varphi \dot{\vee} \psi}\ \text{DR}_2$$

$$\frac{\Gamma, \varphi[t] \Rightarrow_J \psi}{\Gamma, \dot{\forall} \varphi \Rightarrow_J \psi}\ \text{AL} \qquad \frac{\uparrow\Gamma \Rightarrow_J \varphi}{\Gamma \Rightarrow_J \dot{\forall} \varphi}\ \text{AR} \qquad \frac{\uparrow\Gamma, \varphi \Rightarrow_J \uparrow\psi}{\Gamma, \dot{\exists} \varphi \Rightarrow_J \psi}\ \text{EL} \qquad \frac{\Gamma \Rightarrow_J \varphi[t]}{\Gamma \Rightarrow_J \dot{\exists} \varphi}\ \text{ER}$$

B Notes on the Coq Formalisation

Our formalisation consists of about 7500 lines of code, with an even split between specification and proofs. The code is structured as follows.

Section	**Specification**	**Proofs**
Preliminaries Autosubst	169	53
Preliminaries for $\mathbb{F}^*$	680	599
Tarski Semantics	655	682
Kripke Semantics	342	255
On Markov's Principle	593	978
Preliminaries for $\mathbb{F}$	523	430
Heyting Semantics	297	456
Dialogue Semantics	312	488
Total	3571	3941

In general, we find that Coq provides the ideal grounds for formalising projects like ours. It has external libraries supporting the formalisation of syntax, enough automation to support the limited amounts we need and allows constructive reverse mathematics due to its axiomatic minimality.

In the remainder of the section, we elaborate on noteworthy design choices of the formalisation.

Formalisation of binders There are various competing techniques to formalise binders in proof assistants. In first-order logic, binders occur in quantification. The chosen technique especially affects the definition of deduction systems and can considerably ease or impede proofs of standard properties like weakening.

We opted for a de Bruijn representation of variables and binders with parallel substitutions. The Autosubst 2 tool [50] provides convenient automation for the definition of and proofs about this representation of syntax.

Notably, our representation then results in very straightforward proofs for weakening with only 5 lines. In contrast, using other representations for binders results in considerably more complicated weakening proofs, e.g. 150 lines in an approach using names [14] and 95 lines in an approach using traced syntax [24].

Also note that first-order logic has the simplest structure of binders possible: Since quantifiers range over terms, but terms do not contain binders, we do not need a prior notion of renaming, as usually standard in de Bruijn presentations of syntax. This observation results in more compact code (because usually, every statement on substitutions has to be proved for renamings first, with oftentimes the same proof) and was incorporated into Autosubst 2, which now does not generate renamings if they are not needed. Furthermore, we remark that the HOAS encoding of such simple binding structures results in a strictly positive inductive type and would thus be in principle definable in Coq.

Formalisoation of signatures Our whole development is parametrised against a signature, defined as a typeclass in Coq:

```
Class Signature := B_S { Funcs : Type; fun_ar  : Funcs -> nat ;
                         Preds : Type; pred_ar : Preds -> nat }.
```

We implement term and predicate application using the dependent vector type. While the vector type is known to cause issues in dependent programming, in this instance it was the best choice. Recursion on terms is accepted by Coq's guardness checker, and while the generated induction principle (as is always the case for nested inductives) is too weak, a sufficient version can easily be implemented by hand:

```
Inductive vec_in (A : Type) (a : A) : forall n, vector A n -> Type :=
| vec_inB n (v : vector A n) : vec_in a (cons a v)
| vec_inS a’ n (v :vector A n) : vec_in a v -> vec_in a (cons a’ v).

Lemma strong_term_ind (p : term -> Type) :
  (forall x, p (var_term x)) ->
  (forall F v, (forall t, vec_in t v -> p t) -> p (Func F v)) ->
   forall (t : term), p t.
```

Syntactic fragments There are essentially four ways to formalise the syntactic fragment $\mathbb{F}^*$. First, we could parametrise the type of formulas with tags, as done in [14] and second, we could use well-explored techniques for modular syntax [27,9]. However, both of these approaches would not be compatible with the Autosubst tool. Additionally, modular syntax would force users of our developed library for first-order logic to work on the peculiar representation of syntax using containers or functors instead of regular inductive types.

The third option is to only define the type $\mathbb{F}$, and then define a predicate on this formulas characterising the fragment $\mathbb{F}^*$. This approach introduces many additional assumptions in almost all statements, decreasing their readability and yielding many simple but repetitive proof obligations. Furthermore, we would have to parameterise natural deduction over predicates as well, in order for the (IE) rule to not introduce terms e.g. containing $\dot{\exists}$ when only deductions over $\mathbb{F}^*$ should be considered.

To make the formalisation as clear and reusable as possible, we chose the fourth and most simple possible approach: We essentially duplicate the contents of Section 2 for both $\mathbb{F}^*$ and $\mathbb{F}$, resulting in two independent developments on top of the two preliminary parts.

Parametrised deduction systems When defining the minimal, intuitionistic, and classical versions of natural deduction, a similar issue arises. Here, we chose to use one single predicate definition, where the rules for explosion and Peirce can be enabled or disabled using tags, which are parameters of the predicate.

```
Inductive peirce := class | intu.
Inductive bottom := expl  | lconst.
Inductive prv : forall (p : peirce) (b : bottom),
   list (form) -> form -> Prop := (* ... *).
```

We can then define all considered variants of ND by fixing those parameters:

```
Notation "A ⊢CE phi" := (@prv class expl A phi) (at level 30).
Notation "A ⊢CL phi" := (@prv class lconst A phi) (at level 30).
Notation "A ⊢IE phi" := (@prv intu expl A phi) (at level 30).
```

This definition allows us to give for instance a general weakening proof, which can then be instantiated to the different versions. Similarly, we can give a parametrised soundness proof, and depending on the parameters fix required properties on the models used in the definition of validity.

Object tactics At several parts of our developments we have to build concrete ND derivations. This can always be done by explicitly applying the constructors of the ND predicate, which however becomes tedious quickly. We thus developed object tactics reminiscent of the tactics available in Coq. The tactic `ointros` for instance applies the (II) rule, whereas the tactic `oapply` can apply hypotheses, i.e. combine the rules (IE) and (C). All object tactics are in the file `FullND.v`.

Extraction to λ-calculus The proof that completeness of provability w.r.t. standard Tarski and Kripke semantics is equivalent to $\mathsf{MP_L}$ crucially relies on an L-enumeration of provable formulas. While giving a Coq enumeration is easy using techniques described in [14], the translation of any function to a model of computation is considered notoriously hard. We use the framework by Forster and Kunze [16] which allows the automated translation of Coq functions to L.

Using the framework was mostly easy and spared us considerable formalisation effort. However, the framework covers only simple types, whereas our representation of both terms and formulas contains the dependent vector type. We circumvent this problem by defining a non-dependent term type `term’` and a predicate `wf` characterising exactly the terms in correspondence with our original type of terms.

```
Inductive term’ := var_term’ : nat -> term’ | Func’ (name : nat)
                 | App’ : term’ -> term’ -> term’.

Inductive varornot := isvar | novar.
Inductive wf : varornot -> term’ -> Prop :=
| wf_var n : wf isvar (var_term’ n)
| wf_fun f : wf novar (Func’ f)
| wf_app v s t : wf v s -> wf novar t -> wf novar (App’ s t).
```

We then define a formula type `form’` based on `term’` and a suitable deduction system. One can give a bijection between well-formed non-dependent terms `term’` and dependent terms `term` and prove the equivalence of the corresponding deduction systems under this bijection.

Functions working on `term’` and `form’` were easily extracted to L using the framework, yielding an L-enumerability proof for ND essentially with no manual formalisation effort.

Library of formalised undecidable problems in Coq We take the formalisation of synthetic undecidability from [14], which is part of the Coq library of formalised undecidable problems [12]. The reduction from L-halting to provability is factored via Turing machines, Minsky machines, binary stack machines and the Post correspondence problem (PCP), all part of the library as well.

Equations package Defining non-structurally recursive functions is sometimes considered hard in Coq and other proof assistants based on dependent type theory. One such example is the function $[\![\_]\!]$ used to embed formulas into Heyting algebras (Definition 45). We use the Equations package [49] to define this function by recursion on the size of the formula, ignoring terms. The definition then becomes entirely straightforward and the provided `simp` tactic, while sometimes a bit premature, enables compact proofs.