Bulletin of the **S**ection of **L**ogic

Alexander V. Gheorghiu
David J. Pym

DEFINITE FORMULAE, NEGATION-AS-FAILURE, AND THE BASE-EXTENSION SEMANTICS OF INTUITIONISTIC PROPOSITIONAL LOGIC

Abstract

Proof-theoretic semantics (P-tS) is the paradigm of semantics in which meaning in logic is based on proof (as opposed to truth). A particular instance of P-tS for intuitionistic propositional logic (IPL) is its *base-extension semantics* (B-eS). This semantics is given by a relation called support, explaining the meaning of the logical constants, which is parameterized by systems of rules called *bases* that provide the semantics of atomic propositions. In this paper, we interpret bases as collections of definite formulae and use the operational view of them as provided by uniform proof-search — the proof-theoretic foundation of logic programming (LP) — to establish the completeness of IPL for the B-eS. This perspective allows negation, a subtle issue in P-tS, to be understood in terms of the *negation-as-failure* protocol in LP. Specifically, while the denial of a proposition is traditionally understood as the assertion of its negation, in B-eS we may understand the denial of a proposition as the failure to find a proof of it. In this way, assertion and denial are both prime concepts in P-tS.

Keywords: Logic programming, proof-theoretic semantics, bilateralism, negation-as-failure.

1. Introduction

The definition of a system of logic may be given *proof-theoretically* as a collection of rules of inference that, when composed, determine proofs;

that is, formal constructions of arguments that establish that a conclusion is a consequence of some assumptions:

$$\frac{\text{Established Premiss}_1 \quad \ldots \quad \text{Established Premiss}_k}{\text{Conclusion}} \Downarrow$$

The systematic use of symbolic and mathematical techniques to determine the forms of valid deductive argument defines *deductive logic*: conclusions are inferred from assumptions.

This is all very well as a way of defining what proofs are, but it relatively rarely reflects either how logic is used in practical reasoning problems or the method by which proofs are found. Rather, proofs are more often constructed by starting with a desired, or putative, conclusion and applying the rules of inference 'backwards'. In this usage, the rules are sometimes called *reduction operators*, read from conclusion to premisses, and denoted

$$\frac{\text{Sufficient Premiss}_1 \quad \ldots \quad \text{Sufficient Premiss}_k}{\text{Putative Conclusion}} \Uparrow$$

Constructions in a system of reduction operators are called *reductions*. This paradigm is known as *reductive logic*. The space of reductions of a putative conclusion is larger than its space of proofs, including also failed searches — Pym and Ritter [22] have studied the reductive logic for intuitionistic and classical logic in which such objects are meaningful entities.

As one fixes more and more control structure relative to a set of reduction operators, which determining what reductions are made at what time, one increasingly delegates work to a machine. The extreme case is *logic programming* (LP) in which such controls are fully specified. This view is, perhaps, somewhat obscured by the usual presentation of Horn-clause LP with SLD-resolution — see, for example, Kowalski [14] and Lloyd [17] — but it is explicit in work by Miller et al. [19, 20]. What makes this work is that one restricts to the *hereditary Harrop fragment* of a logic in which contexts contain only *definite formulae* — essentially, formulae in which disjunction only appears negatively. In LP, one typically thinks of the formulae in the context of a sequent as *definional*, which underpins its use in symbolic artificial intelligence.

While deductive logic is suitable for considering the validity of propositions relative to sets of axioms, reductive logic is suitable for considering the meaning of propositions relative to *systems of inference*. That the semantics of a statement is determined by its inferential behaviour is known as *inferentialism* (see Brandom [2]), which has a mathematical realization as *proof-theoretic semantics* (P-tS).

In P-tS, the meaning of the logical connectives is usually derived from the rules of a natural deduction system for the logic — for example, typically, one uses Gentzen's [32] NJ for intuitionistic logic. Meanwhile, the meanings of atomic propositions is supplied by an *atomic system* — a set of rules over atomic propositions. For example, taken from Sandqvist [26], the meaning of the proposition 'Tammy is a vixen' can be understood as arising from the following rule:

$$\frac{\text{Tammy is a fox} \quad \text{Tammy is female}}{\text{Tammy is a vixen}}$$

Sandqvist [29] gave a P-tS for intuitionistic propositional logic (IPL) called *base-extension semantics* (B-eS). It proceeds by a judgement called *support*, parameterized by atomic systems, that defines the logical constants whose base case, the meaning of atoms, is given by derivability in an atomic system.

There is an intuitive relationship between P-tS and LP: the way in bases are *definitional* in P-tS is precisely how sets of definite formulae are *definitional* in LP. Schroeder-Heister and Hallnäs [9, 10] have used this relationship to address questions of *harmony* and *inversion* in P-tS.

In this paper, we show that the completeness of IPL for the B-eS can be understood in terms of the operational view of definite formulae. Miller [19] gave this operational view of the hereditary Harrop fragment of IPL a proof-theoretic denotational semantics which proceeds by a least fixed point construction over the Herbrand base. A set of definite formulae parameterizes the construction. By thinking of this set as a base, we prove the completeness of IPL for the aforementioned B-eS by passing through the denotational semantics.

This work exposes an interpretation of negation in P-tS as a manifestation of the *negation-as-failure* (NAF) protocol. The P-tS of negation is a subtle issue — see, for example, Kürbis [16]. Meanwhile, in LP, the relationship between provability and refutation is made through NAF: a statement $\neg\varphi$ is established precisely when the system fails to find a proof for φ. The completeness argument for IPL in this paper shows that negation in B-eS can be understood in terms of the failure to find a proof. Hence, from the perspective of B-eS, it is not the case, as advanced by Frege [6] and endorsed by Dummett [4], that denying a statement φ is equal to asserting the negation of φ. Instead, denial in P-tS is conceptually prior to negation. In this way, through the lens of reductive logic, P-tS may be regarded as practising a form of *bilateralism* — the philosophical practice of giving equal consideration to dual concepts such as assertion and denial, truth and falsity, and so on. Of course, bilateralism with respect to negation in logic is a subject that received serious attention in the literature — see, for example, Smiley [31], Rumfitt [25], Francez [5], Wansing [35], and Kürbis [16].

The paper brings together the following fields: proof-theoretic semantics, reductive logic, and logic programming. Some such connexions have already been witnessed in the literature — see, for example, Hallnäs and Schroeder-Heister [9, 10]. The value is that we can mutually use one to explicate phenomena in the other, such as understanding the meaning of negation in terms of NAF. That is not to argue in favour of NAF as an explanation of negation, but only that it manifests in the operational account of B-eS provided by the LP perspective.

The paper has three parts. In the first part, Section 2, we give the relevant background on IPL: Section 2.1 contains the syntax and terminology that we adopt for IPL; Section 2.2 defines the hereditary Harrop fragment (i.e., definite formulae) and gives their operational reading. In the second part, Section 3, we summarize the B-eS for IPL as given by Sandqvist [29]: in Section 3.1 we define the support relation giving the semantics, and in Section 3.2 we summarize the existing proof of completeness. In the third part, Section 4, we study B-eS from the perspective of the operational reading of definite formulae: Section 4.1 relates atomic systems and sets of definite formulae; Section 4.2 proves completeness argument for IPL for the B-eS through the operational reading of definite formulae; and, Section 4.3 discusses how this perspective manifests negation-as-failure as an explanation of the proof-theoretic meaning of negation. The paper concludes in Section 5 with a summary of our results and a discussion of future work.

2. Intuitionistic Propositional Logic

2.1. Syntax and Consequence

There are various presentation of intuitionistic propositional logic (IPL) in the literature. We begin by fixing the relevant concepts and terminology used in this paper.

Definition 2.1 (Formulae). Fix a (denumerable) set of atomic propositions $\mathbb{A}$. The set of formulae $\mathbb{F}$ (over $\mathbb{A}$) is constructed by the following grammar:

$$\varphi ::= \mathrm{p} \in \mathbb{A} \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \varphi \to \varphi \mid \bot$$

Definition 2.2 (Sequent). A sequent is a pair $\Gamma \triangleright \varphi$ in which Γ is a (countable) set of formulae and φ is a formula.

We use $\vdash$ as the consequence judgement relation defining IPL — that is, $\Gamma \vdash \varphi$ denotes that the sequent $\Gamma \triangleright \varphi$ is a consequence of IPL. We may write $\vdash \varphi$ to abbreviate $\varnothing \vdash \varphi$.

Throughout, we assume familiarity with the standard natural deduction system NJ for IPL as introduced by Gentzen [32] — see, for example, van Dalen [34] and Troelstra and Schwichtenberg [33]. Nonetheless we provide the relevant definitions in quick succession to keep the paper self-contained

Definition 2.3 (Natural Deduction Argument). A natural deduction argument is a rooted tree of formulas in which some (possibly no) leaves are marked as discharged. An argument is open if it has undischarged assumptions; otherwise, it is closed.

The leaves of an argument are its *assumptions*, the root is its *conclusion*. That $\mathcal{A}$ has open assumptions Γ, closed assumptions Δ, and conclusion φ may be denoted as follows:

$$\begin{array}{c} \mathcal{A} \\ \varphi \end{array} \qquad \begin{array}{c} \Gamma, [\Delta] \\ \mathcal{A} \end{array} \qquad \begin{array}{c} \Gamma, [\Delta] \\ \mathcal{A} \\ \varphi \end{array}$$

Definition 2.4 (Natural Deduction System NJ). The natural deduction system NJ is composed of the rules in Figure 1.

Definition 2.5 (NJ-Derivation). The set of NJ-derivations is defined inductively as follows:

- Base Case. If φ is a formula, then the one element tree φ is an NJ-derivation.
- Inductive Step. Let r be a rule in NJ and $\mathcal{D}_1, ..., \mathcal{D}_n$ be a (possibly empty) list of NJ-derivations. If $\mathcal{D}$ is an argument arising from applying r to $\mathcal{D}_1, ..., \mathcal{D}_n$, then $\mathcal{D}$ is an NJ-derivation.

$$\frac{\varphi \quad \psi}{\varphi \wedge \psi}\,\wedge_{\mathsf{I}} \qquad \frac{\varphi \wedge \psi}{\varphi}\,\wedge^1_{\mathsf{E}} \quad \frac{\varphi \wedge \psi}{\psi}\,\wedge^2_{\mathsf{E}}$$

$$\frac{\varphi}{\varphi \vee \psi}\,\vee^1_{\mathsf{I}} \quad \frac{\psi}{\varphi \vee \psi}\,\vee^2_{\mathsf{I}} \qquad \frac{\varphi \vee \psi \quad \begin{matrix}[\varphi] \\ \chi\end{matrix} \quad \begin{matrix}[\psi] \\ \chi\end{matrix}}{\chi}\,\vee_{\mathsf{E}}$$

$$\frac{\begin{matrix}[\varphi] \\ \psi\end{matrix}}{\varphi \to \psi}\,\to_{\mathsf{I}} \qquad \frac{\varphi \quad \varphi \to \psi}{\psi}\,\to_{\mathsf{E}} \qquad \frac{\bot}{\varphi}\,\bot_{\mathsf{E}}$$

Figure 1. Calculus NJ

If $\mathcal{D}$ is an NJ-derivation with undischarged leaves composing the set Γ and root φ, then it is an argument for the sequent $\Gamma \triangleright \varphi$. In this paper, we characterize IPL by NJ:

$$\Gamma \vdash \varphi \qquad \text{iff} \qquad \text{there is an } \mathsf{NJ}\text{-derivation for } \Gamma \triangleright \varphi$$

2.2. The Hereditary Harrop Fragment

The hereditary Harrop fragment of IPL admits an operational reading that we use to deliver the completeness of a proof-theoretic semantics for IPL. This section closely follows work by Miller [19] (see also Harland [11]).

The propositional hereditary Harrop formulae are generated by the following grammar in which $A \in \mathbb{A}$ is an atomic proposition, D is a *definite formula*, and G is a *goal* formula:

$$\begin{array}{lcl} D & ::= & A \mid G \to A \mid D \wedge D \\ G & ::= & A \mid D \to G \mid G \wedge G \mid G \vee G \end{array}$$

A set of definite formulae $\mathscr{P}$ is a *program* — typically, it is a finite set, but we shall have cause to consider infinite sets. The set of all programs is $\mathbb{P}$. We call a sequent $\mathscr{P} \triangleright G$, in which $\mathscr{P}$ is a program and G is a goal, a *query*.

The hereditary Harrop fragment of IPL admits an operational reading which renders it a logic programming language, here called hHLP. The operational semantics of hHLP is given by *uniform* proof-search for $\mathscr{P} \triangleright G$ in a sequent calculus for IPL — see Miller et al. [20].

$\mathscr{P} \vdash A$	iff	$A \in \mathsf{cl}(\mathscr{P})$	(IN)
$\mathscr{P} \vdash A$	iff	$G \to A \in \mathsf{cl}(\mathscr{P})$ and $\mathscr{P} \vdash G$	(CLAUSE)
$\mathscr{P} \vdash G_1 \vee G_2$	iff	$\mathscr{P} \vdash G_1$ or $\mathscr{P} \vdash G_2$	(OR)
$\mathscr{P} \vdash G_1 \wedge G_2$	iff	$\mathscr{P} \vdash G_1$ and $\mathscr{P} \vdash G_2$	(AND)
$\mathscr{P} \vdash D \to G$	iff	$\mathscr{P} \cup \{D\} \vdash G$	(LOAD)

Figure 2. Operational Semantics for hHLP

For purely technical reasons, we require a decomposition function $\mathsf{cl}(-) : \mathbb{P} \to \mathbb{P}$ that will unpack conjunctions. Let $[\mathscr{P}]$ be the least set satisfying the following:

- $\mathscr{P} \subseteq \mathsf{cl}(\mathscr{P})$
- If $D_1 \wedge D_2 \in \mathsf{cl}(\mathscr{P})$, then $D_1 \in \mathsf{cl}(\mathscr{P})$ and $D_2 \in \mathsf{cl}(\mathscr{P})$.

Definition 2.6 (Operational Semantics for hHLP). The operational semantics for hHLP is given by the clauses in Figure 2.

Importantly, hHLP language is complete for the hereditary Harrop fragment of IPL; that is, $\mathscr{P} \triangleright G$ has a successful execution iff it is a consequence of IPL — see Miller [20].

The standard frame semantics for IPL by Kripke [15] forms a model-theoretic semantics for hHLP. However, the hereditary Harrop fragment is sufficiently restrictive that we may simplify the semantics in a useful way.

Definition 2.7 (Interpretation). An interpretation is a mapping $I : \mathbb{P} \to \mathcal{P}(\mathbb{A})$ such that $\mathscr{P} \subseteq \mathscr{Q}$ implies $I(\mathscr{P}) \subseteq I(\mathscr{Q})$.

Definition 2.8 (Satisfaction). The satisfaction judgement is given by the clauses of Figure 3.

We desire a particular interpretation J such that the following holds:

$$J, \mathscr{P} \vDash G \qquad \text{iff} \qquad \mathscr{P} \vdash G$$

To this end, we consider a function T from interpretations to interpretations that corresponds to unfolding derivability in a base:

$$\begin{aligned} T(I)(\mathscr{P}) \quad := \quad & \{A \mid A \in \mathsf{cl}(\mathscr{P})\} \cup \\ & \{A \mid (G \to A) \in \mathsf{cl}(\mathscr{P}) \text{ and } I, \mathscr{P} \vDash G\} \end{aligned}$$

$$
\begin{array}{lll}
I, \mathscr{P} \vDash A & \text{iff} & A \in I(\mathscr{P}) \\
I, \mathscr{P} \vDash G_1 \vee G_2 & \text{iff} & I, \mathscr{P} \vDash G_1 \text{ or } I, \mathscr{P} \vDash G_2 \\
I, \mathscr{P} \vDash G_1 \wedge G_2 & \text{iff} & I, \mathscr{P} \vDash G_1 \text{ and } I, \mathscr{P} \vDash G_2 \\
I, \mathscr{P} \vDash D \to G & \text{iff} & I, \mathscr{P} \cup \{D\} \vDash G
\end{array}
$$

Figure 3. Denotational Semantics for hHLP

Interpretations form a lattice under point-wise union ($\sqcup$), point-wise intersection ($\sqcap$), and point-wise subset ($\sqsubseteq$); the bottom of the lattice is given by $I_\bot : \mathscr{P} \mapsto \varnothing$. It is easy to see that T is monotonic and continuous on this lattice, and, by the Knaster-Tarski Theorem [1], its least fixed-point is given as follows:

$$T^\omega I_\bot := I_\bot \sqcup T(I_\bot) \sqcup T^2(I_\bot) \sqcup \ldots$$

Intuitively, each application of T concerns the application of a clause so that $T^\omega I_\bot$ corresponds to arbitrarily many applications.

LEMMA 2.9. *For any program $\mathscr{P}$ and goal G,*

$$T^\omega I_\bot, \mathscr{P} \vDash G \qquad \textit{iff} \qquad \mathscr{P} \vdash G$$

PROOF: The result was proved by Miller [19] — see also Harland [11]. □

3. Base-extension Semantics

In this section, we give a brief, but complete, synopsis of the base-extension semantics (B-eS) for IPL as introduced by Sandqvist [29]. The semantics proceeds through a *support* relation parametrized by certain atomic systems, called *bases*. There are related base-extension semantics for classical logic — see Sandqvist [27, 28] and Makinson [18].

We differ slightly in presentation from Sandqvist [29]. First, we refer to more the possibility of more general definitions (e.g., considering nth level atomic systems for $n > 2$). Second, we make use of derivations as mathematical objects. Third, we parameterize support over a notion of base called a *basis*, a class of atomic systems. These differences help bridge the gap between the earlier work and the connexions to logic programming in this paper. It also sets the B-eS for IPL within the wider literature of P-tS from which we draw the generalizations.

3.1. Support in a Base

A common idea in proof-theoretic semantics — the paradigm of meaning in which B-eS operates — is that the meaning of atomic propositions is given by sets of atomic rules governing their inferential behaviour. Piecha and Schroeder-Heister [30, 21] have given a useful inductive hierarchy of them.

Definition 3.1 (Atomic Rule). An nth-level atomic rule is defined as follows:

- A zeroth-level atomic rule is a rule of the following form in which $\mathrm{c} \in \mathbb{A}$:

$$\overline{\mathrm{c}}$$

- A first-level atomic rule is a rule of the following form in which $\mathrm{p}_1, ..., \mathrm{p}_n, \mathrm{c} \in \mathbb{A}$,

$$\frac{\mathrm{p}_1 \quad \dots \quad \mathrm{p}_n}{\mathrm{c}}$$

- An $(n+1)$th-level atomic rule is a rule of the following form in which $\mathrm{p}_1, ..., \mathrm{p}_n, \mathrm{c} \in \mathbb{A}$ and $\Sigma_1, ..., \Sigma_n$ are (possibly empty) sets of nth-level atomic rules:

$$\frac{\begin{matrix}[\Sigma_1] \\ \mathrm{p}_1\end{matrix} \quad \dots \quad \begin{matrix}[\Sigma_n] \\ \mathrm{p}_n\end{matrix}}{\mathrm{c}}$$

We take that premisses may be empty such that an mth-level atomic rule is an nth-level atomic rule for any $n > m$. Having sets of atomic rule as hypotheses is more general than have sets of atomic propositions as hypotheses; the latter is captured by the former by taking zeroth-order atomic rules. Nonetheless, the generalization is, perhaps, unexpected. We discuss it further in Section 4.2.

Definition 3.2 (Atomic System). An atomic system is a set of atomic rules.

Atomic systems may have infinitely many rules but they are at most countably infinite. They are used to base validity in P-tS on proof. The definition of a derivation is a generalization of natural deduction *à la* Gentzen [32], which was given by Piecha and Schroeder-Heister [30, 21].

Definition 3.3 (Derivation in an Atomic System). Let $\mathscr{A}$ be an atomic system. The set of $\mathscr{A}$-derivations is defined inductive as follows:

- Base Case. If $\mathscr{A}$ contains a zeroth-level rule concluding c, then the natural deduction argument consisting of just the node c is a $\mathscr{A}$-derivation.

- Induction Step. Suppose $\mathscr{A}$ contains an $(n+1)$th-level rule r of the following form:

$$\frac{\begin{matrix}[\Sigma_1] & & [\Sigma_n] \\ \mathrm{p}_1 & \dots & \mathrm{p}_n\end{matrix}}{\mathrm{c}}$$

  And suppose that for each $1 \leq i \leq n$ there is a $\mathscr{A}$-derivation $\mathcal{D}_i$ of the following form:

$$\begin{matrix}\Gamma_i, \Sigma_i \\ \mathcal{D}_i \\ \mathrm{p}_i\end{matrix}$$

  Then the natural deduction argument with root c and immediate sub-trees $\mathcal{D}_1$,...,$\mathcal{D}_n$ is a $\mathscr{A}$-argument from $\Gamma_1 \cup ... \cup \Gamma_n$ to c.

An atom c is derivable from Γ in $\mathscr{A}$ — denoted $\Gamma \vdash_{\mathscr{A}}$ c — iff there is a $\mathscr{A}$-derivation from Γ to c.

Typically, we do not consider all atomic systems, but restrict attention to some particular class.

Definition 3.4 (Basis). A basis is a set of atomic systems.

Having fixed a basis $\mathfrak{B}$, an atomic system $\mathscr{B} \in \mathfrak{B}$ is called a *base*. A base-extension semantics is formulated relative to a basis via a support relation.

Definition 3.5 (Support in a Base). Fix a basis $\mathfrak{B}$. Support over $\mathscr{B}$ is the least relation $\Vdash_{\_}$ on sequents and bases in $\mathfrak{B}$ defined by the clause of Figure 4. The validity judgement over $\mathfrak{B}$ is the following relation $\Vdash$ one sequent:

$$\Gamma \Vdash \varphi \qquad \text{iff} \qquad \Gamma \Vdash_{\mathscr{B}} \varphi \text{ for any } \mathscr{B} \in \mathfrak{B}$$

$\Gamma \Vdash_{\mathscr{B}} \varphi$	iff	for any $\mathscr{C} \in \mathfrak{B}$ such that $\mathscr{B} \subseteq \mathscr{C}$, if $\Vdash_{\mathscr{C}} \psi$ for all $\psi \in \Gamma$, then $\Vdash_{\mathscr{C}} \varphi$	$(\Rightarrow)$
$\Vdash_{\mathscr{B}} \mathrm{p}$	iff	$\vdash_{\mathscr{B}} \mathrm{p}$	$(\mathbb{A})$
$\Vdash_{\mathscr{B}} \varphi \to \psi$	iff	$\varphi \Vdash_{\mathscr{B}} \psi$	$(\to)$
$\Vdash_{\mathscr{B}} \varphi \wedge \psi$	iff	$\Vdash_{\mathscr{B}} \varphi$ and $\Vdash_{\mathscr{B}} \psi$	$(\wedge)$
$\Vdash_{\mathscr{B}} \varphi \vee \psi$	iff	for any $\mathscr{C} \in \mathfrak{B}$ such that $\mathscr{B} \subseteq \mathscr{C}$ and any $\mathrm{p} \in \mathbb{A}$, if $\varphi \Vdash_{\mathscr{C}} \mathrm{p}$ and $\psi \Vdash_{\mathscr{C}} \mathrm{p}$, then $\Vdash_{\mathscr{C}} \mathrm{p}$	$(\vee)$
$\Vdash_{\mathscr{B}} \bot$	iff	$\Vdash_{\mathscr{B}} \mathrm{p}$ for any $\mathrm{p} \in \mathbb{A}$	$(\bot)$

Figure 4. Support in a Base

Sandqvist [27] gave this semantics with a basis $\mathfrak{S}$ consisting of atomic rules that are *properly* second-level; that is, rules of the form

$$\frac{\begin{matrix}[\Sigma_1] \\ \mathrm{p}_1\end{matrix} \quad \ldots \quad \begin{matrix}[\Sigma_n] \\ \mathrm{p}_n\end{matrix}}{\mathrm{c}}$$

in which Σ_1,...,Σ_n are sets of atoms.

THEOREM 3.6 (Soundness & Completeness). *$\Gamma \vdash \varphi$ iff $\Gamma \Vdash \varphi$ over $\mathfrak{S}$.*

PROOF: Proved by Sandqvist [29] — see Section 3.2. □

The support relation satisfies some important expected properties, such as the following:

LEMMA 3.7. *If $\Gamma \Vdash_{\mathscr{B}} \varphi$ and $\mathscr{C} \supseteq \mathscr{B}$, then $\Gamma \Vdash_{\mathscr{C}} \varphi$.*

PROOF: Proved by Sandqvist [29] by induction on support in a base. □

This summarizes the B-eS for IPL Sandqvist [29] proved the soundness of IPL for the B-eS by showing that validity admits all the rules of NJ. His proof of completeness is more complex. In essence, Sandqvist [29] proved completeness of IPL for the B-eS by constructing a bespoke atomic system $\mathscr{N}$ to a given validity judgement that allows us to *simulate* an NJ-derivation for the sequent in question. We present the main ideas here as we refer to them in Section 4.2.

3.2. Completeness of IPL via a Natural Base

We want to show that $\Gamma \Vdash \gamma$ implies $\Gamma \vdash \gamma$. We understand the latter in terms of provability in NJ. Therefore, we associate to each formula ρ in the sequent $\Gamma \triangleright \gamma$ a unique atom r and construct a base $\mathscr{N}$ emulating NJ such that r behaves in $\mathscr{N}$ as ρ behaves in NJ.

For example, let $\Gamma \triangleright \gamma$ contain $\rho := \mathrm{p} \land \mathrm{q}$. The rules governing ρ are the conjunction introduction and elimination rules of NJ, so we require $\mathscr{N}$ to contain the following rules in which r is alien to $\Gamma \triangleright \gamma$:

$$\frac{\mathrm{p} \quad \mathrm{q}}{\mathrm{r}} \qquad \frac{\mathrm{r}}{\mathrm{p}} \quad \frac{\mathrm{r}}{\mathrm{q}}$$

These rules are designed such that r behaves in $\mathscr{N}$ precisely as ρ does in NJ. That is, they emulate the conjunction rules. The shorthand for r is $(\mathrm{p} \land \mathrm{q})^\flat$ — that is $r = \rho^\flat$ — so that the above rules may be expressed more clearly as follows:

$$\frac{\mathrm{p} \quad \mathrm{q}}{(\mathrm{p} \land \mathrm{q})^\flat} \qquad \frac{(\mathrm{p} \land \mathrm{q})^\flat}{\mathrm{p}} \quad \frac{(\mathrm{p} \land \mathrm{q})^\flat}{\mathrm{q}}$$

For clarity, we give another example. Suppose $\Gamma \triangleright \gamma$ also contains $\sigma := \mathrm{p} \to \mathrm{q}$, then $\mathscr{N}$ contains rules that emulate the implication introduction and elimination rules of NJ for σ using an atom $s := \sigma^\flat := (\mathrm{p} \to \mathrm{q})^\flat$ alien to Γ and γ. That is, $\mathscr{N}$ contains the following rules:

$$\frac{\begin{matrix}[\mathrm{p}] \\ \mathrm{q}\end{matrix}}{(\mathrm{p} \to \mathrm{q})^\flat} \qquad \frac{\mathrm{p} \quad (\mathrm{p} \to \mathrm{q})^\flat}{\mathrm{q}}$$

The details of how $\mathscr{N}$ is constructed and how it delivers completeness are below.

Fix a sequent $\Gamma \triangleright \gamma$. To every sub-formula φ of $\Gamma \triangleright \gamma$ associate a unique atomic proposition $\varphi^\flat$ as follows:

- if $\varphi \notin \mathbb{A}$, then $\varphi^\flat$ is an atom that does not occur in $\Gamma \triangleright \gamma$;
- if $\varphi \in \mathbb{A}$, then $\varphi^\flat = \varphi$.

The right-inverse of $-^\flat$ is $-^\natural$ and both functions act on sets point-wise,

$$\Sigma^\flat := \{\varphi^\flat \mid \varphi \in \Sigma\} \qquad P^\natural := \{\mathrm{p}^\natural \mid \mathrm{p} \in P\}$$

$$\frac{\varphi^\flat \quad \psi^\flat}{(\varphi \wedge \psi)^\flat}\ \wedge_{\mathsf{I}}{}^\flat \qquad \frac{(\varphi \wedge \psi)^\flat}{\varphi^\flat}\ \wedge_{\mathsf{E}}{}^\flat \qquad \frac{(\varphi \wedge \psi)^\flat}{\psi^\flat}\ \wedge_{\mathsf{E}}{}^\flat$$

$$\frac{\varphi^\flat}{(\varphi \vee \psi)^\flat}\ \vee_{\mathsf{I}}{}^\flat \qquad \frac{\psi^\flat}{(\varphi \vee \psi)^\flat}\ \vee_{\mathsf{I}}{}^\flat \qquad \frac{(\varphi \vee \psi)^\flat \quad \begin{array}{c}[\varphi^\flat]\\ \mathrm{p}\end{array} \quad \begin{array}{c}[\psi^\flat]\\ \mathrm{p}\end{array}}{\mathrm{p}}\ \vee_{\mathsf{E}}{}^\flat$$

$$\frac{\begin{array}{c}[\varphi^\flat]\\ \psi^\flat\end{array}}{(\varphi \to \psi)^\flat}\ \to_{\mathsf{I}}{}^\flat \qquad \frac{\varphi^\flat \quad (\varphi \to \psi)^\flat}{\psi^\flat}\ \to_{\mathsf{E}}{}^\flat \qquad \frac{\bot^\flat}{\mathrm{p}}\ \bot_{\mathsf{E}}{}^\flat$$

Figure 5. Atomic System $\mathscr{N}$

Let $\mathscr{N}$ be the atomic system containing precisely the rules of Figure 5 for any φ, ψ occurring in $\Gamma \triangleright \gamma$ and any $\mathrm{p} \in \mathbb{A}$. These rules are precisely such that $\varphi^\flat$ behaves in $\mathscr{N}$ as φ does in NJ. Note that, for any validity judgement, the atomic system $\mathscr{N}$ thus generated is indeed a Sandqvist base.

In this set-up, Sandqvist [29] establishes three properties that collectively deliver completeness.

Lemma 3.8. *Let* $\mathrm{P} \subseteq \mathbb{A}$ *and* $\mathrm{p} \in \mathbb{A}$ *and let* $\mathscr{B} \in \mathfrak{S}$*,*

$$\mathrm{P} \Vdash_{\mathscr{B}} \mathrm{p} \qquad \textit{iff} \qquad \mathrm{P} \vdash_{\mathscr{B}} \mathrm{p}$$

This claim is a basic completeness result in which the context Σ is restricted to a set of atomic propositions and the extract p is an atomic proposition.

Lemma 3.9. *For every* φ *occurring in* $\Gamma \triangleright \gamma$ *and any* $\mathscr{N}' \supseteq \mathscr{N}$*,*

$$\Vdash_{\mathscr{N}'} \varphi^\flat \qquad \textit{iff} \qquad \Vdash_{\mathscr{N}'} \varphi$$

In other words, $\varphi^\flat$ and φ are equivalent in $\mathscr{N}$ — that is, $\varphi^\flat \Vdash_{\mathscr{N}} \varphi$ and $\varphi \Vdash_{\mathscr{N}} \varphi^\flat$. The property allows us to move between the basic case (i.e., the set-up of Lemma 3.8) and the general case (i.e., completeness — Theorem 3.6). This is the crucial step in the proof of completeness. In Section 4.2, we study it in terms of the operational account of definite formulae given in Section 2.2.

Lemma 3.10. *Let* $\mathrm{P} \subseteq \mathbb{A}$ *and* $\mathrm{p} \in \mathbb{A}$,

$$\mathrm{P} \Vdash_{\mathscr{N}} p \qquad \textit{implies} \qquad \mathrm{P}^{\natural} \vdash p^{\natural}$$

This property is the simulation statement. It allows us to make the final move from derivability in $\mathscr{N}$ to derivability in NJ.

These lemmas collectively suffice for completeness:

Proof: Theorem 3.6 — Completeness. Let $\mathscr{N}$ be the bespoke base for $\Gamma \rhd \varphi$. By 3.9, for any $\mathscr{N}' \supseteq \mathscr{N}$ we have $\Gamma^{\flat} \Vdash_{\mathscr{N}'} \varphi^{\flat}$. Since $\mathscr{N} \supseteq \mathscr{N}$, we infer $\Gamma^{\flat} \Vdash_{\mathscr{N}} \varphi^{\flat}$. Therefore, by 3.8, we have $\Gamma^{\flat} \vdash \mathscr{N} \varphi^{\flat}$. Finally, by 3.10, $\Gamma \vdash \varphi$, as required. □

In the next section, we show that the completeness follows intuitively from regarding $\mathscr{N}$ as a program capturing the inferential content of NJ. In general, a base may be regarded as a program, so that the application of a rule in the base corresponds to the use of a clause in the program. We demonstrate that the validity of a formula φ in the base $\mathscr{N}$ emulates the execution of a goal $\varphi^{\flat}$ relative to the program $\mathscr{N}$. By construction of $\mathscr{N}$, such executions simulate the construction of an NJ proof of φ. Hence, IPL is complete with respect to the B-eS.

4. Definite Formulae, Proof-search, and Completeness

There is an intuitive encoding of atomic rules as formulae. More precisely, as *definite* formulae. Under this encoding, the bases which deliver B-eS live within the hereditary Harrop fragment of IPL. The latter has a simple operational reading via proof-search for uniform proofs (see Section 2.2) that enables a proof-theoretic denotational semantics — the least fixed point construction. We use this well-understood phenomenon to deliver the completeness of IPL with respect to Sandqvist's B-eS [29] — see Section 3.

Doing this reveals a subtle interpretation of the meaning of negation in terms of the negation-as-failure protocol. A reductive logic view of the denial of a formula is the failure to find a proof of it. Thus, according to the view of B-eS arising from the account passing through the operational reading of definite formulae, in B-eS denial is conceptionally prior to negation and both require equal consideration.

4.1. Atomic Systems vs. Programs

Intuitively, atomic systems in B-eS are definitional in precisely the same way as programs in hHLP are definitional. To illustrate this, we must systematically move between them, which we do by encoding atomic systems as programs.

Let $\lfloor - \rfloor$ be as follows:

- The encoding of zeroth-level rule is as follows:

$$\left\lfloor \frac{}{\mathrm{c}} \right\rfloor := \mathrm{c}$$

- The encoding of a first-level rule is as follows:

$$\left\lfloor \frac{\mathrm{p}_1 \quad \ldots \quad \mathrm{p}_n}{\mathrm{c}} \right\rfloor := (\mathrm{p}_1 \wedge \ldots \wedge \mathrm{p}_n) \to \mathrm{c}$$

- The encoding of an nth-level rule is as follows:

$$\left\lfloor \frac{\begin{matrix}[\Sigma_1] & & [\Sigma_n] \\ \mathrm{p}_1 & \ldots & \mathrm{p}_n\end{matrix}}{\mathrm{c}} \right\rfloor := \big((\lfloor \Sigma_1 \rfloor \to \mathrm{p}_1) \wedge \ldots \wedge (\lfloor \Sigma_n \rfloor \to \mathrm{p}_n)\big) \to \mathrm{c}$$

For example, $\to_{\mathsf{I}}{}^{\flat}$ in Figure 5 yields the following schematically:

$$(\varphi^\flat \to \psi^\flat) \to (\varphi \to \psi)^\flat$$

The hierarchy of atomic system provided by Piecha and Schroeder-Heister [30, 21] (Definition 3.1) precisely corresponds to the inductive depth of the grammar for hereditary Harrop formulae — that is, if $\mathscr{A}$ is an n-th level atomic system, then

$$\vdash_{\mathscr{A}} \mathrm{p} \qquad \text{iff} \qquad \lfloor \mathscr{A} \rfloor \vdash \mathrm{p}$$

Therefore, we may suppress the encoding function, and henceforth use atomic systems and programs interchangeably — that is, we may write $\mathscr{A} \vdash \mathrm{p}$ to denote $\lfloor \mathscr{A} \rfloor \vdash \mathrm{p}$.

Of course, in the Sanqvist basis, we are limited to properly second-level atomic systems, but the grammar of definite clauses can handle considerably more. Indeed, the work below suggests that completeness holds for nth-level atomic systems for $n \geq 2$.

Formally, to say that bases are definitional in the sense of programs, we mean the following:

$$\Vdash_{\mathscr{B}} \varphi \qquad \text{iff} \qquad \mathscr{N} \cup \mathscr{B} \vdash \varphi^{\flat} \tag{$*$}$$

Here $\mathscr{N}$ contains rules governing φ when the formula is complex — that is, φ is a sub-formula of a sequent $\Gamma \triangleright \psi$ which generates $\mathscr{N}$ — and arbitrary otherwise.

It is important that we use $\varphi^{\flat}$ rather than φ in ($*$). It is certainly *not* the case that bases behave exactly as contexts; that is, we do *not* have the following equivalence:

$$\Vdash_{\mathscr{B}} \varphi \qquad \text{iff} \qquad \mathscr{B} \vdash \varphi \tag{$**$}$$

That this generalization fails is shown by the following counter-example:

Example 4.1. Consider the following formula:

$$\varphi := (\mathrm{a} \to \mathrm{b} \vee \mathrm{c}) \to \big((\mathrm{a} \to \mathrm{b}) \vee (\mathrm{a} \to \mathrm{c})\big)$$

The formula φ is not a consequence of IPL; hence, by completeness of IPL with respect to the B-eS we have $\Vdash_{\mathscr{B}} (\mathrm{a} \to \mathrm{b} \vee \mathrm{c})$ and $\nVdash_{\mathscr{B}} (\mathrm{a} \to \mathrm{b}) \vee (\mathrm{a} \to \mathrm{c})$, for some $\mathscr{B}$. This shows that ($**$) does not hold as it means the second judgment follows from the first — that is, $\Vdash_{\mathscr{B}} (\mathrm{a} \to \mathrm{b} \vee \mathrm{c})$ implies $\Vdash_{\mathscr{B}} (\mathrm{a} \to \mathrm{b}) \vee (\mathrm{a} \to \mathrm{c})$, for any $\mathscr{B}$ — as witnessed by the following computation in hHLP:

$$\begin{array}{lllr}
\Vdash_{\mathscr{B}} \mathrm{a} \to \mathrm{b} \vee \mathrm{c} & \text{implies} & \mathscr{B} \vdash \mathrm{a} \to \mathrm{b} \vee \mathrm{c} & (**) \\
& \text{implies} & \mathscr{B} \cup \{\mathrm{a}\} \vdash \mathrm{b} \vee \mathrm{c} & (\mathsf{LOAD}) \\
& \text{implies} & \mathscr{B} \cup \{\mathrm{a}\} \vdash \mathrm{b} \text{ or } \mathscr{B} \cup \{\mathrm{a}\} \vdash \mathrm{c} & (\mathsf{OR}) \\
& \text{implies} & \mathscr{B} \vdash \mathrm{a} \to \mathrm{b} \text{ or } \mathscr{B} \vdash \mathrm{a} \to \mathrm{c} & (\mathsf{LOAD}) \\
& \text{implies} & \mathscr{B} \vdash (\mathrm{a} \to \mathrm{b}) \vee (\mathrm{a} \to \mathrm{c}) & (\mathsf{OR}) \\
& \text{implies} & \Vdash_{\mathscr{B}} (\mathrm{a} \to \mathrm{b}) \vee (\mathrm{a} \to \mathrm{c}) & (**)
\end{array}$$

To see how ($*$) works it is instructive to consider an example that explicitly uses the proof-search for the definite formulae as a meta-calculus for derivability in a base.

Example 4.2. By Theorem 3.6, we have $\Vdash_{\varnothing} \mathrm{a} \vee \mathrm{b} \to \mathrm{b} \vee \mathrm{a}$. That $\mathscr{N} \vdash (\mathrm{a} \vee \mathrm{b} \to \mathrm{b} \vee \mathrm{a})^{\flat}$ indeed obtains is witnessed by the computation,

$$
\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\mathscr{N}, (\mathrm{a} \lor \mathrm{b})^\flat \vdash (\mathrm{a} \lor \mathrm{b})^\flat}\ \Uparrow \mathsf{IN} \qquad \mathcal{R}_a \quad \mathcal{R}_b}{\mathscr{N}, (ata \lor \mathrm{b})^\flat \vdash (\mathrm{a} \lor \mathrm{b})^\flat \land (\mathrm{a} \to (\mathrm{b} \lor \mathrm{a})^\flat) \land (\mathrm{b} \to (\mathrm{b} \lor \mathrm{a})^\flat)}\ \Uparrow \mathsf{AND}}{\mathscr{N}, (\mathrm{a} \lor \mathrm{b})^\flat \vdash (\mathrm{b} \lor \mathrm{a})^\flat}\ \Uparrow \mathsf{CLAUSE}\,(\lor_{\mathsf{E}})^\flat}{\mathscr{N}, (\mathrm{a} \lor \mathrm{b})^\flat \vdash (\mathrm{b} \lor \mathrm{a})^\flat}\ \Uparrow \mathsf{LOAD}}{\mathscr{N} \vdash (\mathrm{a} \lor \mathrm{b} \to \mathrm{b} \lor \mathrm{a})^\flat}\ \Uparrow \mathsf{CLAUSE}\,(\to_{\mathsf{I}})^\flat
$$

where $\mathcal{R}_x$ for $x \in \{a, b\}$ is

$$
\dfrac{\dfrac{\dfrac{}{\mathscr{N}, (\mathrm{b} \lor \mathrm{a})^\flat, \mathrm{x} \vdash \mathrm{x}}\ \Uparrow \mathsf{IN}}{\mathscr{N}, (\mathrm{b} \lor \mathrm{a})^\flat, \mathrm{x} \vdash (\mathrm{b} \lor \mathrm{a})^\flat}\ \Uparrow \mathsf{CLAUSE}\,(\lor_{\mathsf{I}})^\flat}{\mathscr{N}, (\mathrm{b} \lor \mathrm{a})^\flat \vdash \mathrm{x} \to (\mathrm{b} \lor \mathrm{a})^\flat}\ \Uparrow \mathsf{LOAD}
$$

In the next section, we use the relationship between atomic systems and programs to prove completeness of IPL with respect to the B-eS.

4.2. Completeness of IPL via Logic Programming

We may prove completeness of IPL with respect to the B-eS by passing through hHLP as follows:

$$
\begin{array}{ccc}
T^\omega I_\bot, \mathscr{N} \vDash \varphi^\flat & \longleftrightarrow & \mathscr{N} \vdash \varphi^\flat \\
\uparrow & & \downarrow \\
\Vdash_{\mathscr{N}} \varphi & & \vdash \varphi
\end{array}
$$

The diagram requires three claims, the middle one of which is Lemma 2.9. The other two are Lemma 4.3 and Lemma 4.4, respectively, reading in the direction of the arrows.

The intuition of the completeness argument is two-fold: firstly, that $\mathscr{N}$ is to $\varphi^\flat$ as NJ is to φ; secondly, the use of a rule in a base corresponds to the use of a clause in the corresponding program; thirdly, execution in $\mathscr{N}$ corresponds to proof(-search) in NJ. In this set-up, the T^ω construction captures the construction of a proof: the application of a rule corresponds to a use of T, the iterative application of rules corresponds to the iterative application of T — that is, to T^ω.

It remains to prove the claims and completeness. Fix a sequent $\Gamma \rhd \varphi$

and let $-^\flat$ and $\mathscr{N}$ be constructed as in Section 3.2 for this sequent. Let Δ be an arbitrary set of sub-formulae of the sequent and ψ an arbitrary subformula of the sequent.

LEMMA 4.3 (Emulation). *If* $\Vdash_{\mathscr{N}} \psi$, *then* $T^\omega I_\bot, \mathscr{N} \vDash \psi^\flat$.

PROOF: We prove a stronger proposition: for any $\mathscr{N}' \supseteq \mathscr{N}$, if $\Vdash_{\mathscr{N}'} \psi$, then $T^\omega I_\bot, \mathscr{N}' \vDash \psi^\flat$. We proceed by induction on support in a base according to the various cases of Figure 4. As above, for the sake of economy, we combine the clauses $\Rightarrow$ and $\to$.

- $\psi \in \mathbb{A}$. Note $\psi^\flat = \psi$, by definition. Therefore, if $\Vdash_{\mathscr{N}'} \psi$, then $\vdash_{\mathscr{N}'} \psi$, but this is precisely emulated by application of T. Hence, $T^\omega I_\bot, \mathscr{N}' \vDash \psi$.

- $\psi = \bot$. If $\Vdash_{\mathscr{N}'} \bot$, then $\Vdash_{\mathscr{N}'} \mathrm{p}$, for every $\mathrm{p} \in \mathbb{A}$. By the induction hypothesis (IH), $T^\omega I_\bot, \mathscr{N}' \vDash \mathrm{p}$ for every $\mathrm{p} \in \mathbb{A}$. It follows that $T^\omega I_\bot, \mathscr{N}' \vDash \bot^\flat$.

- $\psi := \psi_1 \wedge \psi_2$. By the $\wedge$-clause for support, $\Vdash_{\mathscr{N}'} \psi_1$ and $\Vdash_{\mathscr{N}'} \psi_2$. Hence, by the IH, $T^\omega I_\bot, \mathscr{N}' \vDash \psi_1^\flat$ and $T^\omega I_\bot, \mathscr{N}' \vDash \psi_2^\flat$. By $\wedge$-clause for satisfaction, $T^\omega I_\bot, \mathscr{N}' \vDash \psi_1^\flat \wedge \psi_2^\flat$. The result follows by $\wedge_{\mathsf{I}}{}^\flat$-schema.

- $\psi := \psi_1 \vee \psi_2$. By Lemma 3.9, $\psi_1 \Vdash_{\mathscr{N}'} \psi_1^\flat$ and $\psi_2 \Vdash_{\mathscr{N}'} \psi_2^\flat$. By the $\vee_{\mathsf{I}}$-scheme in $\mathscr{N}'$, both $\psi_1^\flat \Vdash (\psi_1 \vee \psi_2)^\flat$ and $\psi_2^\flat \Vdash (\psi_1 \vee \psi_2)^\flat$. Therefore, by $\Rightarrow$-clause for support, we have $\psi_1 \Vdash_{\mathscr{N}'} (\psi_1 \vee \psi_2)^\flat$ and $\psi_2 \Vdash_{\mathscr{N}'} (\psi_1 \vee \psi_2)^\flat$. Using the $\vee$-clause for support on the assumption $\Vdash_{\mathscr{N}'} \psi_1 \vee \psi_2$ with these results means that $\Vdash_{\mathscr{N}'} (\psi_1 \vee \psi_2)^\flat$. That is, $T^\omega, \mathscr{N}' \vDash (\psi_1 \vee \psi_2)^\flat$, as required.

- $\psi := \psi_1 \to \psi_2$. By the $\to$-clause for satisfaction, $\psi_1 \Vdash_{\mathscr{N}'} \psi_2$. So, by the $\Rightarrow$-clause for satisfaction, $\Vdash_{\mathscr{N}''} \psi_1$ implies $\Vdash_{\mathscr{N}''} \psi_2$ for any $\mathscr{N}'' \supseteq \mathscr{N}'$. Let $\mathscr{N}'' := \mathscr{N}' \cup \{\psi_1^\flat\}$. Since $\Vdash_{\mathscr{N}', \psi^\flat} \psi^\flat$, by Lemma 3.9, we have $\Vdash_{\mathscr{N}', \psi^\flat} \psi$, hence we infer $\Vdash_{\mathscr{N}', \psi^\flat} \psi_2$. By the IH, $T^\omega I_\bot, \mathscr{N}' \cup \{\psi_1^\flat\} \vDash \psi_2^\flat$. Hence, $T^\omega I_\bot, \mathscr{N}' \vDash \psi_1^\flat \to \psi_2^\flat$. By the $\to_{\mathsf{I}}{}^\flat$-scheme, $T^\omega I_\bot \mathscr{N}' \vDash (\psi_1 \to \psi_2)^\flat$, as required.

This completes the induction. □

LEMMA 4.4 (Simulation). *If* $\mathscr{N} \cup \Delta^\flat \vdash \psi^\flat$*, then* $\Delta \vdash \psi$.

PROOF: We proceed by induction on the length of execution. Intuitively, the execution of $\mathscr{N} \cup \Delta^\flat \vdash \psi^\flat$ simulates the reductive construction of a proof of ψ from Δ in NJ — that is, a proof-search. We proceed by induction on the length of the execution.

BASE CASE: It must be that $\psi \in \Delta$, so $\Delta \vdash \psi$ is immediate.

INDUCTIVE STEP: By construction of $\mathscr{N}$, the execution concludes by CLAUSE applied to a definite clause ρ simulating a rule r $\in$ NJ; that is, $\mathscr{N} \cup \Delta^\flat \vdash \psi_i^\flat$ for ψ_i such that $\psi_1^\flat \wedge .... \wedge \psi_n^\flat \to \psi^\flat$. By the induction hypothesis (IH), $\Delta \vdash \psi_i$ for $1 \leq i \leq n$. It follows that $\Delta \vdash \psi$ by applying $r \in$ NJ.

For example, if the execution concludes by CLAUSE applied to the clause for $\wedge$-introduction (i.e., $\psi^\flat \wedge \psi^\flat \to (\psi \wedge \psi)^\flat$), then the trace is as follows:

$$\dfrac{\dfrac{\begin{matrix}\vdots \\ \mathscr{N} \cup \Delta^\flat \vdash \psi^\flat\end{matrix} \quad \begin{matrix}\vdots \\ \mathscr{N} \cup \Delta^\flat \vdash \psi^\flat\end{matrix}}{\mathscr{N} \cup \Delta^\flat \vdash \psi^\flat \wedge \psi^\flat}}{\mathscr{N} \cup \Delta^\flat \vdash (\psi \wedge \psi)^\flat}$$

By the induction hypothesis, we have proofs witnessing $\Delta \vdash \psi$ and $\Delta \vdash \psi$, and by $\wedge$-introduction:

$$\dfrac{\begin{matrix}\vdots \\ \psi\end{matrix} \quad \begin{matrix}\vdots \\ \psi\end{matrix}}{\psi \wedge \psi}$$

This completes the induction. □

Following the diagram, we have the completeness of IPL with respect to the B-eS:

PROOF: Theorem 3.6 — Completeness. We require to show that $\Vdash \varphi$ implies $\Vdash_{\mathscr{N}} \varphi$ for arbitrary φ. To this end, assume $\Vdash \varphi$. Let $\mathscr{N}$ be the natural base generated by φ. By definition, from the assumption, we have $\Vdash_{\mathscr{N}} \varphi$. Hence, by Lemma 4.3, it follows that $T^\omega I_\bot, \mathscr{N} \vDash \varphi^\flat$. Whence, by Lemma 2.9, we obtain $\mathscr{N} \vdash \varphi^\flat$. Thus, by Lemma 4.4, $\vdash \varphi$, as required. □

In the following section, we discuss how reductive logic delivers the completeness proof above and the essential role played by both proofs and refutations.

4.3. Negation-as-Failure

A reduction in a proof system is constructed co-recursively by applying the rules of inference backwards. Even though each step corresponds to the application of a rule, the reduction can fail to be a proof as the computation arrives at an irreducible sequent that is not an instance of an axiom in the logic. For example, in hHLP, one may compute the following:

$$\dfrac{\dfrac{\mathrm{p} \triangleright \mathrm{q}}{\mathrm{p} \triangleright \mathrm{p} \vee \mathrm{q}} \Uparrow \mathsf{LOAD}}{\varnothing \triangleright \mathrm{p} \to (\mathrm{p} \vee \mathrm{q})} \Uparrow \mathsf{OR}$$

This reduction fails to be a proof, despite every step being a valid inference, since the initial sequent is not an instance of IN. In reductive logic, such failed attempts at constructing proofs are not meaningless: Pym and Ritter [22] have provided a semantics of the reductive logic of IPL in which such reductions are given meaning by using hypothetical rules — that is, the construction would succeed in the presence of the following rule:

$$\frac{\mathrm{p}}{\mathrm{q}}$$

The categorical treatment of this semantics has them as *indeterminates* in a polynomial category — this adumbrates current work by Pym et al. [23], who have shown that the B-eS is entirely natural from the perspective of categorical logic. The use of such additional rules to give semantics to constructions that are not proofs directly corresponds to the use of atomic systems in the B-eS for IPL; for example, let $\mathscr{A}$ be the atomic system containing the rule above, then the judgement $p \Vdash_{\mathscr{A}} q$ obtains. Altogether, this suggests a close relationship between B-eS and reductive logic, which manifests with the operational reading of definite clauses and their relationship to atomic rules in Section 4.

Within P-tS, negation is a subtle issue — see Kürbis [16]. We may use the perspective of LP developed herein to review the meaning of absurdity ($\bot$).

There is no introduction rule for $\bot$ in NJ. One may not construct a proof of absurdity without it already being, *in some sense*, assumed; for example, $\varphi, \varphi \to \bot \vdash \bot$ obtains because the context $\{\varphi, \varphi \to \bot\}$ is already, in some sense, absurd. We may use LP to understand what that sense is. To simplify matters, observe that the judgement $\Gamma \vdash \bot$ is equivalent to $\vdash \varphi \to \bot$ for some formula φ. Therefore, we may restrict attention to negations of this kind to understand the meaning of absurdity.

By Theorem 3.6 (Soundness) and Lemma 4.4 (Simulation), we see that the converse of Theorem 4.3 holds. Therefore,

$$\Vdash \neg\varphi \qquad \text{iff} \qquad T^{\omega} I_{\bot}, \mathscr{N} \vdash (\neg\varphi)^{\flat}$$

Unfolding the semantics, this is equivalent to $T^{\omega} I_{\bot}, \mathscr{N} \cup \{\varphi^{\flat}\} \vdash \bot^{\flat}$. Thus, the sense in which φ is absurd is that its interpretation under $T^{\omega} I_{\bot}$ contains absurdity; that is, φ is absurd iff $\bot^{\flat} \in T^{\omega} I_{\bot}(\varphi)$. What does this tell us about the meaning of $\neg\varphi$? Since there s no proof of $\bot^{\flat}$, we have that the meaning of $\neg\varphi$ is that there is no proof of $(\varphi)^{\flat}$ in $\mathscr{N}$. This is the *negation-as-failure* principle. How does it yield the clause for $\bot$ in Figure 4?

Passing through $(*)$ in Section 4.1,

$$\Vdash_{\mathscr{B}} \bot \qquad \text{iff} \qquad \mathscr{N} \cup \mathscr{B} \vdash \bot^{\flat}$$

Since there is no introduction rule for $\bot^{\flat}$ in $\mathscr{N}$, it must be that $\mathscr{B}$ derives it. Thus, there is rule in $\mathscr{B}$ of the following form:

$$\dfrac{\begin{matrix}[\Sigma_1] \\ \mathrm{p}_1\end{matrix} \quad \ldots \quad \begin{matrix}[\Sigma_n] \\ \mathrm{p}_n\end{matrix}}{\bot^{\flat}}$$

To simplify matters, we introduce alien q and $\bar{\mathrm{q}}$ as 'conjunctions' of some subset $q_1, ..., q_k$ and $q_{k+1}, ..., q_n$ of $\mathrm{p}_1, ..., \mathrm{p}_n$ in the inferentialist sense. That is, we introduce the following, where $\Pi_i = \Sigma_j$ iff $\mathrm{q}_i = \mathrm{p}_i$ for $i, j \in \{1, ..., n\}$:

$$\dfrac{\begin{matrix}[\Pi_1] \\ \mathrm{q}_1\end{matrix} \quad \ldots \quad \begin{matrix}[\Pi_n \\ \mathrm{q}_n]\end{matrix}}{\mathrm{q}} \qquad \dfrac{\begin{matrix}[\Pi_{k+1}] \\ \mathrm{q}_{k+1}\end{matrix} \quad \ldots \quad \begin{matrix}[\Pi_n \\ \mathrm{q}_n]\end{matrix}}{\bar{\mathrm{q}}}$$

Doing this allows us to replace the above rule with the following:

$$\dfrac{\mathrm{q} \quad \bar{\mathrm{q}}}{\bot^{\flat}}$$

In this case, the inferential behaviour of q and $\bar{q}$ is that they are contradictory propositions: together, they infer absurdity.

In this way, negation is implicit in atoms. What is significant from this analysis is that the semantics of $\bot$ requires us to observe that there is no proof of it and thus extend the space with proofs of contradictory q and $\bar{q}$. If they are proved in $\mathscr{B}$, then one has proved absurdity; if $\mathscr{B}$ has proved absurdity, then one has proofs for each of these. The subtlety is that since we do not have negation *explicit* in our atoms, we only admit the principle that some atoms are contradictory. If we prove all atoms, then we prove these contradictory atoms; and, if we prove these contradictory atoms, then we have proved absurdity. This justifies the clause for $\bot$,

$$\Vdash_{\mathscr{B}} \bot \qquad \text{iff} \qquad \Vdash_{\mathscr{B}} \mathrm{p} \text{ for any } \mathrm{p} \in \mathbb{A}$$

Piecha and Schroeder-Heister [30, 21] have argued that there are two perspectives on atomic systems: the knowledge view and the definitional view. This becomes clear according to various ways in which a program may be regarded in LP. The negation-as-failure protocol makes use of the definitional perspective; its analogue in terms of knowledge is the *closed-world assumption*. In this case, a knowledge base treats everything that is not known to be valid as invalid. There is significant literature about the closed-world assumption that may be useful for understanding P-tS and what it tells us about reasoning — see, for example, Clark [3], Reiter [24], and Kowalski [14, 13], and Harland [11, 12].

5. Conclusion

Proof-theoretic semantics is the paradigm of meaning based on proof (as opposed to truth). Essential to this approach is the use of atomic systems, which give meaning to atomic propositions. Base-extension semantics is a particular instance of proof-theoretic semantics that proceeds by an inductively defined judgement whose base case is given by provability in an atomic system. It may be regarded as capturing the declarative content of proof-theoretic semantics in the Dummett-Prawitz tradition — see Gheorghiu and Pym [8]. Sandqvist [27] has given a base-extension semantics for intuitionistic propositional logic. Completeness follows by constructing a special bespoke base in which the validity of a complex proposition simulates a natural deduction proof of that formula.

In the base-extension semantics, the meaning of the logical constants is derived from the rules of NJ, while the atomic systems give the meaning of atomic propositions. These atomic systems, which include Sandqvist's special bases that delivers completeness, all sit within the hereditary Harrop fragment of IPL. The significance of this is that an effective operational reading of definite formulae renders them meaning-conferring in a sense analogous to the use of atomic systems. Moreover, this operational account coheres with the independently conceived notion of derivability in an atomic system. Of course, that atomic systems and programs are intimately related has been studied before — see Schroeder-Heister and Hallnäs [9, 10].

Significantly, the operational reading of the definite formulae allows from a simple proof-theoretic model-theoretic semantics that captures the idea of *unfolding* the inferential content of a set of definite clauses or an atomic system. In this paper, we have used the operational account of definite formulae to prove the completeness of intuitionistic propositional logic with respect to its base-extension semantics. The aforementioned special base is interpreted as a program so that completeness follows immediately from the existing completeness result of the model-theoretic semantics of the logic programming language. Doing this reveals the subtle meaning of negation in proof-theoretic semantics.

Historically, the negation of a formula is understood as the denial of the formula itself. This is indeed the case in the model-theoretic semantics of IPL — see Kripke [15]. Using the connection to logic programming in this paper, we see that in base-extension semantics, negation is defined by the failure for there to be a proof. Thus, denial is conceptionally prior to negation. In short, base-extension semantics consider the space of reductions, which is larger than the space of proofs, including failed searches. As illustrated above, the connection between logic programming and base-extension semantics is quite intuitive and useful. More specifically, the T operator delivering the semantics of logic programming corresponds to the application of a rule in a proof system; hence, the T^{ω} construction is fundamental to proof-theoretic semantics. Since logic programming has been studied for various logics (see, for example, the treatment of BI in Gheorghiu et al. [7]), this suggests the possibility for uniform approaches to setting up base-extension semantics for logics by studying their proof-search behaviours. In particular, work by Harland [11, 12] on handling negation in logic programming may be used to address the difficulties posed by the connective — see Kürbis [16].

It remains to investigate further the connection between proof-theoretic semantics and reductive logic, in general, and base-extension semantics and logic programming, in particular.

Acknowledgements. We are grateful to Edmund Robinson, for suggesting the formula in Example 4.1, and to Elaine Pimentel, Yll Buzoku, and the reviewers of an earlier version of the paper for their helpful comments and feedback.

References

[1] K. R. Apt, M. H. Van Emden, *Contributions to the theory of logic programming*, **Journal of the ACM (JACM)**, vol. 29(3) (1982), pp. 841–862.

[2] R. Brandom, **Articulating Reasons: An Introduction to Inferentialism**, Harvard University Press (2000).

[3] K. L. Clark, *Negation as Failure*, [in:] **Logic and Data Bases**, Springer (1978), pp. 293–322.

[4] M. Dummett, **The Logical Basis of Metaphysics**, Harvard University Press (1993).

[5] N. Francez, *Bilateralism in Proof-theoretic Semantics*, **Journal of Philosophical Logic**, vol. 43 (2014), pp. 239–259.

[6] G. Frege, *Die Verneinung. Eine Logische Untersuchung*, **Beiträge Zur Philosophie des Deutschen Idealismus**, vol. 1(3/4) (1919), pp. 143–157.

[7] A. V. Gheorghiu, S. Docherty, D. J. Pym, *Reductive Logic, Coalgebra, and Proof-search: A Perspective from Resource Semantics*, [in:] A. Palmigiano, M. Sadrzadeh (eds.), **Samson Abramsky on Logic and Structure in Computer Science and Beyond**, Springer Outstanding Contributions to Logic Series, Springer (2021), to appear.

[8] A. V. Gheorghiu, D. J. Pym, *From Proof-theoretic Validity to Base-extension Semantics for Intuitionistic Propositional Logic* (Accessed 08 February 2023), URL: https://arxiv.org/abs/2210.05344, submitted.

[9] L. Hallnäs, P. Schroeder-Heister, *A Proof-theoretic Approach to Logic Programming: I. Clauses as Rules*, **Journal of Logic and Computation**, vol. 1(2) (1990), pp. 261–283.

[10] L. Hallnäs, P. Schroeder-Heister, *A Proof-theoretic Approach to Logic Programming: II. Programs as Definitions*, **Journal of Logic and Computation**, vol. 1(5) (1991), pp. 635–660.

[11] J. Harland, **On Hereditary Harrop Formulae as a Basis for Logic Programming**, Ph.D. thesis, The University of Edinburgh (1991).

[12] J. Harland, *Success and Failure for hereditary Harrop Formulae*, **The Journal of Logic Programming**, vol. 17(1) (1993), pp. 1–29.

[13] R. Kowalski, *Logic for Problem Solving*, https://www.doc.ic.ac.uk/~rak/papers/LFPScommentary.pdf (Accessed 15 August 2022), commentary on the book 'Logic for Problem Solving' by R. Kowalski.

[14] R. Kowalski, **Logic for Problem-Solving**, North-Holland Publishing Co. (1986).

[15] S. A. Kripke, *Semantical Analysis of Intuitionistic Logic I*, [in:] **Studies in Logic and the Foundations of Mathematics**, vol. 40, Elsevier (1965), pp. 92–130.

[16] N. Kürbis, **Proof and Falsity: A Logical Investigation**, Cambridge University Press (2019).

[17] J. W. Lloyd, **Foundations of Logic Programming**, Symbolic Computation, Springer-Verlag (1984).

[18] D. Makinson, *On an Inferential Semantics for Classical Logic*, **Logic Journal of IGPL**, vol. 22(1) (2014), pp. 147–154.

[19] D. Miller, *A Logical Analysis of Modules in Logic Programming*, **Journal of Logic Programming**, vol. 6(1-2) (1989), pp. 79–108.

[20] D. Miller, G. Nadathur, F. Pfenning, A. Scedrov, *Uniform Proofs as a Foundation for Logic Programming*, **Annals of Pure and Applied Logic**, vol. 51(1) (1991), pp. 125 – 157.

[21] T. Piecha, P. Schroeder-Heister, *The Definitional View of Atomic Systems in Proof-theoretic Semantics*, [in:] **The Logica Yearbook 2016**, College Publications London (2017), pp. 185–200.

[22] D. J. Pym, E. Ritter, **Reductive logic and Proof-search: Proof Theory, Semantics, and Control**, vol. 45 of Oxford Logic Guides, Oxford University Press (2004).

[23] D. J. Pym, E. Ritter, E. Robinson, *Proof-theoretic Semantics in Sheaves (Extended Abstract)*, [in:] **Proceedings of the Eleventh Scandinavian Logic Symposium — SLSS 11** (2022), pp. 36–38.

[24] R. Reiter, *On closed world data bases*, [in:] **Readings in artificial intelligence**, Elsevier (1981), pp. 119–140.

[25] I. Rumfitt, *'Yes and No'*, **Mind**, vol. 109(436) (2000), pp. 781–823.

[26] T. Sandqvist, *Atomic Bases and the Validity of Peirce's Law*, https://drive.google.com/file/d/1fX8PWh8w2cpOkYS39zR2OGfNEhQESKkl/view (Accessed 15 August 2022), presentation at the World Logic Day event at UCL: The Meaning of Proofs.

[27] T. Sandqvist, **An Inferentialist Interpretation of Classical Logic**, Ph.D. thesis, Uppsala University (2005).

[28] T. Sandqvist, *Classical Logic without Bivalence*, **Analysis**, vol. 69(2) (2009), pp. 211–218.

[29] T. Sandqvist, *Base-extension Semantics for Intuitionistic Sentential Logic*, **Logic Journal of the IGPL**, vol. 23(5) (2015), pp. 719–731.

[30] P. Schroeder-Heister, T. Piecha, *Atomic Systems in Proof-Theoretic Semantics: Two Approaches*, [in:] Ángel Nepomuceno Fernández, O. P. Martins, J. Redmond (eds.), **Epistemology, Knowledge and the Impact of Interaction**, Springer Verlag (2016), pp. 47–62.

[31] T. Smiley, *Rejection*, **Analysis**, vol. 56(1) (1996), pp. 1—9.

[32] M. E. Szabo (ed.), **The Collected Papers of Gerhard Gentzen**, North-Holland Publishing Company (1969).

[33] A. S. Troelstra, H. Schwichtenberg, **Basic Proof Theory**, vol. 43 of Cambridge Tracts in Theoretical Computer Science, Cambridge University Press (2000).

[34] D. van Dalen, **Logic and Structure**, Universitext, Springer (2012).

[35] H. Wansing, *Falsification, Natural Deduction and Bi-intuitionistic Logic*, **Journal of Logic and Computation**, vol. 26(1) (2016), pp. 425–450.

Alexander V. Gheorghiu

University College London
Department of Computer Science
Gower St, London WC1E 6BT
London, United Kingdom

e-mail: alexander.gheorghiu.19@ucl.ac.uk

David J. Pym

University College London
Department of Computer Science
Gower St, London WC1E 6BT
London, United Kingdom

University College London
Department of Philosophy
Gower St, London WC1E 6BT
London, United Kingdom

University of London
Institute of Philosophy
Senate House, Malet St, London WC1E 7HU
London, United Kingdom

e-mail: d.pym@ucl.ac.uk