Defining logical systems via algebraic constraints on proofs

ALEXANDER V. GHEORGHIU, *University College London, London WC1E 6BT, UK.*
E-mail: alexander.gheorghiu.19@ucl.ac.uk

DAVID J. PYM, *University College London, London WC1E 6BT, UK and Institute of Philosophy, University of London, London WC1H 0AR, UK.*
E-mail: d.pym@ucl.ac.uk

Abstract

We present a comprehensive programme analysing the decomposition of proof systems for non-classical logics into proof systems for other logics, especially classical logic, using an algebra of constraints. That is, one recovers a proof system for a target logic by enriching a proof system for another, typically simpler, logic with an algebra of constraints that act as correctness conditions on the latter to capture the former; e.g. one may use Boolean algebra to give constraints in a sequent calculus for classical propositional logic to produce a sequent calculus for intuitionistic propositional logic. The idea behind such forms of decomposition is to obtain a tool for uniform and modular treatment of proof theory and to provide a bridge between semantics logics and their proof theory. The paper discusses the theoretical background of the project and provides several illustrations of its work in the field of intuitionistic and modal logics: including, a uniform treatment of modular and cut-free proof systems for a large class of propositional logics; a general criterion for a novel approach to soundness and completeness of a logic with respect to a model-theoretic semantics; and a case study deriving a model-theoretic semantics from a proof-theoretic specification of a logic.

Keywords: Logic, proof theory, model theory, semantics, modal logic, intuitionistic logic

1 Introduction

The general goal of this paper is to provide a unifying meta-level framework for studying logics. To this end, we introduce a framework in which one can represent the *reasoning* in a logic, as captured by a concept of proof for that logic, in terms of the reasoning within another logic through an algebra of constraints—as a slogan,

$$\text{Proof in } \mathsf{L} = \text{Proof in } \mathsf{L}' + \text{Algebra of Constraints } \mathcal{A}$$

We shall refer back to this slogan often and will use the following abbreviated form:

$$\mathsf{L} = \mathsf{L}' \oplus \mathcal{A}$$

The $\oplus$ is not formal—i.e. $\mathsf{L}' \oplus \mathcal{A}$ may denote any one of several ways of applying constraints $\mathcal{A}$ to L'.

Such decompositions of L into L' and $\mathcal{A}$ allow us to study the meta-theory of the former by analysing the latter. This is advantageous when the latter is typically simpler in some desirable way—e.g. it may relax the side conditions on the use of specific rules—which facilitates the study of the original logic of interest. There are already some examples of such relationships within the literature—they are discussed below. The framework herein provides a general view of the phenomena and provides an umbrella for these seemingly disparate cases. Importantly, there is no

Vol. 35, No. 1, © The Author(s) 2023. Published by Oxford University Press.

Advance Access Publication on 24 November 2023 https://doi.org/10.1093/logcom/exad065

guarantee that the constraints will be solvable algorithmically as it depends on how complex the algebra is, but even relatively simple algebras can have a dramatic impact; e.g. we illustrate how constraint systems using Boolean algebra, which admits solvers, are helpful for the study of proof-search and meta-theory in substructural and intuitionistic logics.

The decompositions expressed by the slogan above may be iterated in valuable ways; i.e. it is possible to decompose L' in the slogan above. Each time we do such a decomposition, the combinatorics of the proof system become more tractable as more and more is delegated to the algebra. Eventually, the combinatorics becomes as simple as possible, and one recovers something with all the flexibility of the proof theory for classical logic. Therefore, we advance the view that, in general, classical logic forms a combinatorial core of syntactic reasoning since its proof theory is comparatively relaxed—i.e. possibly after iterating decompositions of the kind above, one eventually witnesses the following:

$$\text{Proof in } \mathsf{L} \;=\; \text{Classical Proof} + \text{Algebra of Constraints } \mathcal{A}.$$

The view of classical logic as the core of logic has, of course, been advanced before—see, e.g. Gabbay [21].

Using techniques from universal algebra, we define the algebraic constraints by a theory of first-order classical logic; e.g. we may define Boolean algebra by its axiomatization—see Section 2. We then enrich rules of a system L with expressions from $\mathcal{A}$ to express rules of another system $;\mathsf{L}'$—e.g. by using Boolean algebra for the constraints, we may express the single-conclusioned $\vee_{\mathsf{R}}$-rule from Gentzen's LJ [27] with the combinatorics of the multiple-conclusioned $\vee_{\mathsf{R}}$-rule from Gentzen's LK [27],

$$\frac{\Gamma \vdash \phi \cdot x, \psi \cdot \bar{x}}{\Gamma \vdash \phi \vee \psi} \;,$$

One recovers the single-conclusion condition by assigning the variable x to 0 or 1 in the Boolean algebra and evaluating the formula accordingly: one keeps formulae that carry a 1 and deletes formulae that carry 0. A system of rules enriched in this way is called a *constraint system*. Detailed examples are below.

We consider two kinds of relationships the constraint systems may have with the logic of interest. A constraint system is *sound and complete* when the evaluation of construction from the constraints system concludes a sequent iff that sequent is valid in the logic. A stronger relationship is *faithfulness and adequacy*:

- (*Faithful*) The evaluation of a construction from the constraint system is a proof in the logical system of interest.
- (*Adequate*) Every proof in the logical system of interest is the evaluation of some construction from the constraint system.

Both relationships are important, as illustrated by the examples below.

The point of constraint systems is that they allow us to study the meta-theory of the logic of interest. There are two principal such activities presented in this paper: first, one may use constraint systems to study questions of proof-search (i.e. how one constructs proofs via reduction in a formal system) in the logic of interest; second, they may be used to bridge the gap between the proof theory and model theory of a logic. On the latter use, constraint systems allow a novel approach to soundness and completeness proofs, which bypasses truth-in-a-model and term-model constructors; furthermore, they give a principled way of generating a correct-by-design model-theoretic semantics for the logic of interest by analysing a proof system for it, making essential use of algebraic constraints and the aforementioned decomposition to classical logic.

Of course, the idea that one may use labels to internalize the semantics of logics within proof systems has taken several forms and goes back as far as Kanger [38]. It underpins a systematic development of analytic tableaux (see e.g. Fitting [19, 20], Catach [10], Massacci [49], Baldoni [1], Docherty and Pym [15–17] and Galmiche and Méry [23–26]), natural deduction systems (see, e.g. Simpson [67] and Basin *et al.* [2]) and sequent calculi (see, e.g. Mints [53, 54], Viganò [70] and Kushida and Okada [44]). Particularly significant within this stream are the relational calculi studied by Negri [55–57].

The notion of constraint system presented in this paper is closely related to Gabbay's *Labelled Deductive Systems* (LDSs) [22]—see also Russo [64]. However, the paper deviates from the established theory of LDSs in two fundamental ways: first, one may choose any syntactic structure in the grammar of the object-logic (e.g. data composed of formulae, such as sets, multisets, bunches), not just formulae, to annotate; second, the labels do not only express additional information but have an action on the structure. Note, there are other proof systems in the literature in which one labels data composed of formulae—see e.g. Marx *et al.* [48]. Consequently, more subtle examples are available, not otherwise captured by LDSs.

In summary, two main ideas are developed in this paper: a meta-logic for studying object-logics and algebraic constraints for studying the meta-theory of a logic. Both ideas are present elsewhere in the literature and have been studied for different logics but have yet to be formalized and uniformed. This paper aims to provide a general and uniform framework for analysing and understanding the proof theory of logics represented in this way.

The paper begins in Section 2 with an example of a constraint system already in the literature. It continues in Section 3 with the background and notation required for the general treatment throughout the rest of the paper. Constraint systems are defined in general in Section 4, where the correctness properties of soundness, completeness, faithfulness and adequacy are also discussed. In Section 5, we study relational calculi as a general class of constraint systems and study an approach to proving soundness and completeness, which works with validity directly. We give a concrete illustration of the approach applied to intuitionistic propositional logic (IPL) in Section 6—specifically, by using constraint systems, we derive the model-theoretic semantics given by Kripke [42] from LJ, which is sound and complete by construction. In Section 7, we consider the treatment of first-order logics with constraints, the rest of the paper being restricted to propositional logics. The paper concludes in Section 8 with a summary and a discussion of future research.

2 Example: Resource-distribution via Boolean Constraints

In this section, we summarize the resource-distribution via Boolean constraints (RDvBC) mechanism, which was introduced by Harland and Pym [32, 60] as a tool for reasoning about the context-management problem during proof-search in logics with multiplicative connectives, such as Linear Logic (LL) and the logic of Bunched Implications (BI). It is the original example of a decomposition of a proof system in the sense of this paper, as explained at the end of the section. We present RDvBC to motivate the abstract technical work in Section 4 for the general approach. We concentrate on the case of BI to indicate the scope of the approach and the challenges involved in setting it up because the logic operates over quite a subtle data structure—bunches.

2.1 The Logic of Bunched Implications

One may regard BI as the free combination (i.e. the fibration—see Gabbay [21]) of IPL, with connectives $\wedge, \vee, \rightarrow, \top, \bot$, and intuitionistic multiplicative linear logic (IMLL), with connectives

$*, -\!\!*, \top^*$. Let $\mathbb{A}$ be a set of atomic propositions. The following grammar generates the formulae of BI:

$$\phi ::= p \in \mathbb{A} \mid \top \mid \bot \mid \top^* \mid \phi \wedge \phi \mid \phi \vee \phi \mid \phi \to \phi \mid \phi * \phi \mid \phi -\!\!* \phi.$$

The set of all formulae is $\mathbb{FORM}$.

A distinguishing feature of BI is that it has two primitive implications, $\to$ and $-\!\!*$, each corresponding to a different context-former, $;$ and $,$, representing the two conjunctions $\wedge$ and $*$, respectively. As these context-formers do not commute with each other, though individually they behave as usual, contexts in BI are not one of the familiar structures of lists, multisets or sets. Instead, its contexts are *bunches*—a term that derives from the relevance logic literature (see e.g. Read [63]). The set of bunches $\mathbb{BUNCH}$ is defined by the following grammar:

$$\Gamma ::= \phi \in \mathbb{FORM} \mid \varnothing_+ \mid \varnothing_\times \mid \Gamma ; \Gamma \mid \Gamma , \Gamma.$$

A bunch Δ is a sub-bunch of a bunch Γ iff Δ is a sub-tree of Γ. One may write $\Gamma(\Delta)$ to mean that Δ is a sub-bunch of Γ. The operation $\Gamma[\Delta \mapsto \Delta']$—abbreviated to $\Gamma(\Delta')$ where no confusion arises—is the result of replacing the occurrence of Δ by Δ'.

Bunches have similar structural behaviour to the more familiar data-structures used for contexts in logic (e.g. lists, multisets, sets, etc.). We define this behaviour explicitly by means of an equivalence relation called *coherent equivalence*. Two bunches $\Gamma, \Gamma' \in \mathbb{BUNCH}$ are coherently equivalent when $\Gamma \equiv \Gamma'$, where $\equiv$ is the smallest relation satisfying:

- commutative monoid equations for $;$ with unit $\varnothing_+$
- commutative monoid equations for $,$ with unit $\varnothing_\times$
- coherence; i.e. if $\Delta \equiv \Delta'$, then $\Gamma(\Delta) \equiv \Gamma(\Delta')$.

A sequent in BI is a pair $\Gamma \rhd \phi$ in which Γ is a bunch, and ϕ is a formula. We use $\rhd$ as a pairing symbol defining sequents to distinguish it from the judgment $\vdash$ that asserts that a sequent is a consequence of BI. We may characterize the consequence judgment $\vdash$ for BI by provability in the sequent calculus LBI in Figure 1 (see Pym [33]). That is, $\Gamma \vdash \phi$ iff there is an LBI-proof of $\Gamma \rhd \phi$.

As bunches are intended to be the syntactic trees of $\mathbb{BUNCH}$ modulo $\equiv$, we may somewhat relax the formal reading of the rules of LBI. The effect of coherent equivalence is, essentially, to render bunches into two-sorted nested multisets—see Gheorghiu and Marin [28]. Therefore, we may suppress brackets for sections of the bunch with the same context-former and apply rules sensitive to context-formers (e.g. $*_{\mathsf{R}}$) accordingly. For example, any context-former may be used in $*_{\mathsf{R}}$ applied to $p_1 , p_1 , p_3 \rhd q_1 * q_2$; the possibilities are as follows:

$$\frac{p_1 \rhd q_1 \quad p_2 , p_3 \rhd q_2}{p_1 , p_2 , p_3 \rhd q_1 * q_2}\ *_{\mathsf{R}} \qquad \frac{p_1 , p_2 \rhd q_1 \quad p_3 \rhd q_2}{p_1 , p_2 , p_3 \rhd q_1 * q_2}\ *_{\mathsf{R}}.$$

This concludes the introduction of BI. In the next section, we apply the RDvBC mechanism to analyse proof-search in LBI.

2.2 Resource-distribution via Boolean Constraints

Proof-search in LBI is complex because the presence of multiplicative connectives (i.e. $*$ and $-\!\!*$) requires deciding how to distribute the formulae (or, rather, sub-bunches) when making reductions.

$$\frac{}{\phi \vdash \phi}\ \mathsf{taut} \qquad \frac{}{\Gamma(\bot) \vdash \phi}\ \bot_{\mathsf{L}} \qquad \frac{}{\varnothing_{\times} \vdash \top^{*}}\ \top^{*}_{\mathsf{R}} \qquad \frac{}{\varnothing_{+} \vdash \top}\ \top_{\mathsf{R}}$$

$$\frac{\Delta' \vdash \phi \quad \Gamma(\Delta \mathbin{,} \psi) \vdash \chi}{\Gamma(\Delta \mathbin{,} \Delta' \mathbin{,} \phi \mathrel{-\!\!*} \psi) \vdash \chi}\ \mathrel{-\!\!*}_{\mathsf{L}} \qquad \frac{\Delta \mathbin{,} \phi \vdash \psi}{\Delta \vdash \phi \mathrel{-\!\!*} \psi}\ \mathrel{-\!\!*}_{\mathsf{R}}$$

$$\frac{\Delta(\phi \mathbin{,} \psi) \vdash \chi}{\Delta(\phi * \psi) \vdash \chi}\ *_{\mathsf{L}} \qquad \frac{\Delta \vdash \phi \quad \Delta' \vdash \psi}{\Delta \mathbin{,} \Delta' \vdash \phi * \psi}\ *_{\mathsf{R}} \qquad \frac{\Delta(\varnothing_{\times}) \vdash \chi}{\Delta(\top^{*}) \vdash \chi}\ \top^{*}_{\mathsf{L}}$$

$$\frac{\Delta(\phi \mathbin{;} \psi) \vdash \chi}{\Delta(\phi \wedge \psi) \vdash \chi}\ \wedge_{\mathsf{L}} \qquad \frac{\Delta \vdash \phi \quad \Delta \vdash \psi}{\Delta \vdash \phi \wedge \psi}\ \wedge_{\mathsf{R}} \qquad \frac{\Delta(\varnothing_{+}) \vdash \chi}{\Delta(\top) \vdash \chi}\ \top_{\mathsf{L}}$$

$$\frac{\Delta(\phi) \vdash \chi \quad \Delta(\psi) \vdash \chi}{\Delta(\phi \vee \psi) \vdash \chi}\ \vee_{\mathsf{L}} \qquad \frac{\Delta \vdash \phi}{\Delta \vdash \phi \vee \psi}\ \vee_{\mathsf{R1}} \qquad \frac{\Delta \vdash \psi}{\Delta \vdash \phi \vee \psi}\ \vee_{\mathsf{R2}}$$

$$\frac{\Delta \vdash \phi \quad \Gamma(\Delta \mathbin{;} \psi) \vdash \chi}{\Gamma(\Delta \mathbin{;} \phi \to \psi) \vdash \chi}\ \to_{\mathsf{L}} \qquad \frac{\Delta \mathbin{;} \phi \vdash \psi}{\Delta \vdash \phi \to \psi}\ \to_{\mathsf{R}}$$

$$\frac{\Delta(\Delta') \vdash \chi}{\Delta(\Delta' \mathbin{;} \Delta'') \vdash \chi}\ \mathsf{w} \qquad \frac{\Delta \vdash \chi}{\Delta' \vdash \chi}\ \mathsf{e}_{(\Delta \equiv \Delta')} \qquad \frac{\Delta(\Delta' \mathbin{;} \Delta') \vdash \chi}{\Delta(\Delta') \vdash \chi}\ \mathsf{c}$$

FIGURE 1. Sequent Calculus **LBI**.

EXAMPLE 2.1
The following proof-search attempts differ only in the choice of distribution, but one successfully produces a proof, and the other fails:

$$\underbrace{\dfrac{\dfrac{}{\mathrm{p} \vdash \mathrm{p}}\ \mathsf{taut} \quad \dfrac{\dfrac{}{\mathrm{q} \vdash \mathrm{q}}\ \mathsf{taut} \quad \dfrac{}{\mathrm{r} \vdash \mathrm{r}}\ \mathsf{taut}}{\mathrm{q} \mathbin{,} \mathrm{r} \vdash \mathrm{q} * \mathrm{r}}\ *_{\mathsf{R}}}{\mathrm{p} \mathbin{,} \mathrm{q} \mathbin{,} \mathrm{r} \vdash \mathrm{p} * (\mathrm{q} * \mathrm{r})}\ *_{\mathsf{R}}}_{\text{succeeds}} \qquad \underbrace{\dfrac{\dfrac{?}{\mathrm{p} \mathbin{,} \mathrm{q} \vdash \mathrm{p}} \quad \dfrac{?}{\mathrm{r} \vdash \mathrm{q} * \mathrm{r}}}{\mathrm{p} \mathbin{,} \mathrm{q} \mathbin{,} \mathrm{r} \vdash \mathrm{p} * (\mathrm{q} * \mathrm{r})}\ *_{\mathsf{R}}}_{\text{fails}}.$$

How can we analyse the various distribution strategies? This is the question RDvBC addresses.

There is substantial literature on intricate rules of inference in multiplicative logics that are used to keep track of the relevant information to enable proof-search. However, they are generally tailored for one particular distribution method—see e.g. Hodas and Miller [34, 35], Winikoff and Harland [71], Cervasto [11] and Lopez [46]. It is in this context that Harland and Pym [32, 60] introduced the RDvBC mechanism. The idea is that rather than commit to a particular strategy for managing the distribution, one uses Boolean expressions to express that a resource distribution needs to be made and the conditions it needs to satisfy.

Before presenting the technical details of RDvBC, we give a heuristic account. Essentially, one assigns a Boolean expression to each formula requiring distribution. Constraints on the possible values of this expression are then generated during the proof-search and propagated up the search tree, resulting in a set of Boolean equations. A successful proof-search in the enriched system will generate a soluble set of equations corresponding to a distribution of formulae

across the branches of the structure, and instantiating that distribution results in an actual proof. It remains to give the formal detail. We begin by defining the constraint algebra that delivers RDvBC.

A *Boolean algebra* is a tuple $\mathcal{B} := \langle \mathbb{B}, \{+, \times, \bar{\cdot}\}, \{0, 1\}\rangle$ in which $\mathbb{B}$ is a set, $+ : \mathbb{B}^2 \to \mathbb{B}$, $\times : \mathbb{B}^2 \to \mathbb{B}$, $\bar{\cdot} : \mathbb{B} \to \mathbb{B}$ be operators on $\mathbb{B}$, and $0, 1 \in \mathbb{B}$, satisfying the following conditions for any $a, b, c \in \mathbb{B}$:

$$\begin{array}{cc} a + (b + c) = (a + b) + c \quad a \times (b \times c) = (a \times b) \times c & a + b = b + c \quad a \times b = b \times a \\ a + (a \times b) = a \quad a \times (a + b) = a & a + 0 = a \quad a \times 1 = a \\ a + (b \times c) = (a + b) \times (a + c) & a \times (b + c) = (a \times b) + (a \times c) \\ a + \bar{a} = 1 & a \times \bar{a} = 0. \end{array}$$

A *presentation* of the Boolean algebra is a first-order classical logic with equality for which the Boolean algebra is a model. We use the following, in which X is a set of *variables*, e are *Boolean expressions* and ϕ are *Boolean formulae*:

$$e ::= x \in X \mid e + e \mid e \times e \mid \bar{e} \mid 0 \mid 1 \quad \phi ::= (e = e) \mid \phi \,\&\, \phi \mid \phi \parr \phi \mid \neg\phi \mid \forall x\phi \mid \exists x\phi.$$

The symbols $\&$ and $\parr$ are used as classical conjunction and disjunction, respectively.

We are overloading $+$ and $\times$ to be both function-symbols in the term language and their corresponding operators in the Boolean algebra; similarly, we are overloading 0 and 1 to be both constants in the term language and the bottom and top element of the Boolean algebra. This is to economize on notation. We may suppress the $\times$ when no confusion arises—i.e. $t_1 \times t_2$ may be expressed $t_1 t_2$. For a list of Boolean expressions $V = [e_1, \ldots e_n]$, let $\bar{V} := [\bar{e}_1, \ldots \bar{e}_n]$; we may write $V = e$ to denote that V is a list containing only e.

Let some presentation of Boolean algebra be fixed. An *annotated BI-formula* is a pair $\phi \cdot e$ in which ϕ is a BI-formula and is a Boolean expression e. The annotation of a bunch Γ by a list of Boolean expressions V is defined as follows:

- if $\Gamma = \gamma$, where $\gamma \in \mathbb{FORM} \cup \{\varnothing_+, \varnothing_\times\}$ and $V = [e]$, then $\Gamma \cdot V := \gamma \cdot e$;
- if $\Gamma = (\Delta_1 \parr \Delta_2)$, and $V = [e]$, then $\Gamma \cdot V := (\Delta_1 \parr \Delta_2) \cdot e$;
- if $\Gamma = (\Delta_1 \,,\, \Delta_2)$, and V is the concatenation of V_1 and V_2, then $\Gamma \cdot V := (\Delta_1 \cdot V_1 \parr \Delta_2 \cdot V_2)$.

For example, $p \,,\, (q \parr r) \cdot [x, y] := p \cdot x \,,\, (q \parr r) \cdot y$. Notably, the annotation only acts on the top-level of multiplicative connectives and treats everything below (e.g. additive sub-bunches) as formulae. This makes sense as all of the distributions in LBI take place at this level of the bunch.

This concludes the technical overhead required to define the RDvBC mechanism for BI. Roughly, Boolean constraints are used to mark the multiplicative distribution of formulae. The mechanism is captured by proof-search in the sequent calculus $\mathsf{LBI}_{\mathcal{B}}$ comprised of the rules in Figure 2, in which V is a list of Boolean variables that do not appear in any sequents present in the tree and labels that do not change are suppressed. The same names are used for rules in $\mathsf{LBI}_{\mathcal{B}}$ and LBI to economize on notation.

An $\mathsf{LBI}_{\mathcal{B}}$-reduction is a tree constructed by applying the rules of $\mathsf{LBI}_{\mathcal{B}}$ reductively, beginning with a sequent in which each formula is annotated by 1.

$$\frac{y = 1 \wedge V = 0}{\phi \cdot y \,,\, \Delta \cdot V \rhd \phi \cdot 1}\ \mathsf{taut} \qquad \frac{y = 1 \wedge V = 0}{\bot \cdot y \,,\, \Delta \cdot V \rhd \phi}\ \bot_{\mathsf{L}}$$

$$\frac{y = 1 \,\&\, V = 0}{\varnothing_{\times} \cdot y \,,\, \Delta \cdot V \rhd \top^{*}}\ \top^{*}_{\mathsf{R}} \qquad \frac{y = 1 \,\&\, V = 0}{\varnothing_{+} \cdot y \,,\, \Delta \cdot V \rhd \top}\ \top_{\mathsf{R}}$$

$$\frac{\Delta \cdot V \rhd \phi \cdot e \quad \Gamma(\Delta \cdot \bar{V} \,,\, \psi \cdot e) \rhd \chi \quad e = 1}{\Gamma(\Delta \,,\, \phi \mathrel{-\!\!*} \psi \cdot e) \rhd \chi}\ \mathrel{-\!\!*}_{\mathsf{L}} \qquad \frac{\Delta \,,\, \phi \cdot e \rhd \psi \cdot e \quad e = 1}{\Delta \rhd (\phi \mathrel{-\!\!*} \psi) \cdot e}\ \mathrel{-\!\!*}_{\mathsf{R}}$$

$$\frac{\Delta(\phi \cdot e \,,\, \psi \cdot e) \rhd \chi \quad e = 1}{\Delta((\phi * \psi) \cdot e) \rhd \chi}\ *_{\mathsf{L}} \qquad \frac{\Delta \cdot V \rhd \phi \quad \Delta' \cdot \bar{V} \rhd \psi}{\Delta \,,\, \Delta' \rhd \phi * \psi}\ *_{\mathsf{R}} \qquad \frac{\Delta(\varnothing_{\times} \cdot e) \rhd \chi \quad e = 1}{\Delta(\top^{*} \cdot e) \rhd \chi}\ \top^{*}_{\mathsf{L}}$$

$$\frac{\Delta(\phi \cdot e \,;\, \psi \cdot e) \rhd \chi \quad e = 1}{\Delta((\phi \wedge \psi) \cdot e) \rhd \chi}\ \wedge_{\mathsf{L}} \qquad \frac{\Delta \rhd \phi \quad \Delta \rhd \psi}{\Delta \rhd \phi \wedge \psi}\ \wedge_{\mathsf{R}} \qquad \frac{\Delta(\varnothing_{+} \cdot e) \rhd \chi \quad e = 1}{\Delta(\top \cdot e) \rhd \chi}\ \top_{\mathsf{L}}$$

$$\frac{\Delta(\phi \cdot e) \rhd \chi \quad \Delta(\psi \cdot e) \rhd \chi \quad e = 1}{\Delta((\phi \vee \psi) \cdot e) \rhd \chi}\ \vee_{\mathsf{L}} \qquad \frac{\Delta \rhd \phi}{\Delta \rhd \phi \vee \psi}\ \vee_{\mathsf{R1}} \qquad \frac{\Delta \rhd \psi}{\Delta \rhd \phi \vee \psi}\ \vee_{\mathsf{R2}}$$

$$\frac{\Delta \rhd \phi \quad \Gamma(\Delta \,;\, \psi \cdot e) \rhd \chi \quad e = 1}{\Gamma(\Delta \,;\, (\phi \to \psi) \cdot e) \rhd \chi}\ \to_{\mathsf{L}} \qquad \frac{\Delta \,;\, \phi \cdot e \rhd \psi \quad e = 1}{\Delta \rhd \phi \to \psi}\ \to_{\mathsf{R}}$$

$$\frac{\Gamma(\Delta \cdot e) \rhd \chi \quad e = 1}{\Gamma(\Delta \cdot e \,;\, \Delta \cdot e) \rhd \chi}\ \mathsf{w} \qquad \frac{\Delta \rhd \chi}{\Delta' \rhd \chi}\ \mathsf{e}_{(\Delta \equiv \Delta')} \qquad \frac{\Gamma(\Delta \cdot e \,;\, \Delta \cdot e) \rhd \chi \quad e = 1}{\Gamma(\Delta \cdot e) \rhd \chi}\ \mathsf{c}$$

FIGURE 2. Sequent Calculus $\mathsf{LBI}_{\mathcal{B}}$.

EXAMPLE 2.2
The following is an $\mathsf{LBI}_{\mathcal{B}}$-reduction $\mathcal{D}$:

$$\frac{\dfrac{(x_1 = 1) \wedge (x_2 = 0) \wedge (x_3 = 0)}{(\mathrm{p} \cdot x_1) \,,\, (\mathrm{q} \cdot x_2) \,,\, (\mathrm{r} \cdot x_3) \rhd \mathrm{p} \cdot 1}\ \mathsf{taut} \quad \mathcal{D}'}{(\mathrm{p} \cdot 1) \,,\, (\mathrm{q} \cdot 1) \,,\, (\mathrm{r} \cdot 1) \rhd \mathrm{p} * (\mathrm{q} * \mathrm{r}) \cdot 1}\ *_{\mathsf{R}},$$

the sub-tree $\mathcal{D}'$ is the following:

$$\frac{\dfrac{(\bar{x}_1 y_1 = 0) \wedge (\bar{x}_2 y_2 = 1) \wedge (\bar{x}_3 y_3 = 0)}{(\mathrm{p} \cdot \bar{x}_1 y_1) \,,\, (\mathrm{q} \cdot \bar{x}_2 y_2) \,,\, (\mathrm{r} \cdot \bar{x}_3 y_3) \rhd \mathrm{q}}\ \mathsf{taut} \quad \dfrac{(\bar{x}_1 \bar{y}_1 = 0) \wedge (\bar{x}_2 \bar{y}_2 = 0) \wedge (\bar{x}_3 \bar{y}_3 = 1)}{\mathrm{p} \cdot (\bar{x}_1 \bar{y}_1) \,,\, \mathrm{q} \cdot (\bar{x}_2 \bar{y}_2) \,,\, \mathrm{r} \cdot (\bar{x}_3 \bar{y}_3) \rhd \mathrm{r}}\ \mathsf{taut}}{(\mathrm{p} \cdot \bar{x}_1) \,,\, (\mathrm{q} \cdot \bar{x}_2) \,,\, (\mathrm{r} \cdot \bar{x}_3) \rhd \mathrm{q} * \mathrm{r} \cdot 1}\ *_{\mathsf{R}}.$$

Having produced an $\mathsf{LBI}_{\mathcal{B}}$-reduction, if the constraints are consistent, they determine the variables' interpretations to satisfy the constraints. Such interpretations I induce a valuation ν_I that acts on formulae by keeping formulae whose label evaluates to 1 and deleting (i.e. producing the empty-string ε) for formulae whose label evaluates to 0; i.e. let ϕ be a BI-formula and e a Boolean expression,

$$\nu_I(\phi \cdot e) := \begin{cases} \phi & \text{if } I(e) = 1 \\ \varepsilon & \text{if } I(e) = 0. \end{cases}$$

A valuation extends to sequents by acting on each formulae occurring in it, and it extends to $\mathsf{LBI}_{\mathcal{B}}$-reductions by acting on each sequent occurring in it and removing the constraints.

EXAMPLE 2.3 (Example 2.2 cont'd).
The constraints on $\mathcal{D}$ are satisfied by any interpretation $I(z) = 1$ for $z \in \{x_1, y_2\}$ and $I(z) = 0$ for $z \in \{x_2, x_3, y_1, y_3\}$. For any such I, the tree $\nu_I(D)$ is as follows:

$$\dfrac{\dfrac{}{\mathsf{p} \rhd \mathsf{p}}\ \mathsf{taut} \quad \dfrac{\dfrac{}{\mathsf{q} \rhd \mathsf{q}}\ \mathsf{taut} \quad \dfrac{}{\mathsf{r} \rhd \mathsf{r}}\ \mathsf{taut}}{\mathsf{q}\ ,\ \mathsf{r} \rhd \mathsf{q} * \mathsf{r}}\ *_{\mathsf{R}}}{\mathsf{p}\ ,\ \mathsf{q}\ ,\ \mathsf{r} \rhd \mathsf{p} * (\mathsf{q} * \mathsf{r})}\ *_{\mathsf{R}} \quad .$$

This is the successful derivation in LBI in Example 2.1. According to the constraints, a distribution strategy results in a successful proof-search just in case it sends only the first formula to the left branch.

Harland and Pym [32, 60] proved that $\mathsf{LBI}_{\mathcal{B}}$ is *faithful* and *adequate* for LBI in the following sense:

- *Faithfulness*. If $\mathcal{R}$ is an $\mathsf{LBI}_{\mathcal{B}}$-reduction and I is an interpretation satisfying those constraints, then there is a LBI-proof $\mathcal{D}$ such that $\nu_I(\mathcal{R}) = \mathcal{D}$.
- *Adequacy*. If $\mathcal{D}$ is an LBI-proof, then there is a $\mathsf{LBI}_{\mathcal{B}}$-reduction $\mathcal{R}$ and an interpretation I satisfying its constraints such that $\nu_I(R) = \mathcal{D}$.

Recall that we may think of BI as the combination of IPL and IMLL. We may express LBI as the combination of sequent calculi for these two logics (i.e. IMLL and LJ, respectively)—i.e.

$$\mathsf{LBI} = \mathsf{IMLL} \cup \mathsf{LJ}$$

The RDvBC outsources the substructurality in IMLL to Boolean constraints. Hence, in the form of the slogan of this paper,

$$\mathsf{IMLL} = \mathsf{LJ} \oplus \mathcal{B}$$

In this section, we have chosen to study BI (as opposed to just IMLL) to illustrate the modularity of constraint systems. That is, we only have constraints participating actively in part of the sequent calculus for BI, but with the same overall effect since the other part conserves them. Abusing the slogan somewhat, we may express the work of this section as follows:

$$\mathsf{LBI} = (\mathsf{LJ} \oplus \mathcal{B}) \cup \mathsf{LJ}$$

This paper aims to study the use of constraints in proof systems in general. It provides conditions under which they exist and illustrates the use of constraints to study meta-theory.

3 Background

We have two things to set up to give a general presentation of algebraic constraint systems: algebra and propositional logic. The former is captured by first-order classical logic (FOL) (e.g. as Boolean algebra is captured by its axiomatization in Section 2), and the latter is given by a general account of propositional logic as a propositional language together with a consequence relation. There are many presentations of these subjects within the literature; therefore, to avoid confusion, in this section, we define them as they are used in this paper. Importantly, this section introduces much notation used

in the rest of the paper. As we wish to reserve traditional symbols such as $\vdash$ and $\rightarrow$ for the object-logics, we will use $\blacktriangleright$ and $\Rightarrow$ for the meta-logic. In both cases, we use the symbol $\triangleright$ as the sequents symbol, regarding $\vdash$ and $\blacktriangleright$ as *consequence* relations.

3.1 First-order Classical Logic

This section presents first-order classical logic (FOL), which we use to define what we mean by algebra in *algebraic* constraints. We assume familiarity with FOL, so give a terse (but complete) summary to keep the paper self-contained. In particular, we assume familiarity with proof theory for FOL—as covered in e.g. Troelstra and Schwichtenberg [68] and Negri and von Plato [58].

DEFINITION 3.1 (First-order Language).
An alphabet is a tuple $\mathscr{A} := \langle \mathbb{R}, \mathbb{F}, \mathbb{K}, \mathbb{V} \rangle$ in which $\mathbb{R}$, $\mathbb{F}$, $\mathbb{K}$ and $\mathbb{V}$ are pairwise disjoint countable sets of symbols, and each element of $\mathbb{R}$, $\mathbb{F}$ and $\mathbb{K}$ has a fixed arity. The terms, atoms and well-formed formulae (wffs) of an alphabet are as follows:

- The set $\mathbb{TERM}(\mathscr{A})$ of *terms* from $\mathscr{A}$ is the smallest set containing $\mathbb{K}$ and $\mathbb{V}$ such that, for any $F \in \mathbb{F}$, if F has arity n and $T_1, ..., T_n \in \mathbb{TERM}(\mathscr{A})$, then $F(T_1, ..., T_n) \in \mathbb{TERM}(\mathscr{A})$
- The set $\mathbb{ATOMS}(\mathbb{A})$ is set of strings $R(T_1, ..., T_n)$ such that $R \in \mathbb{R}$ has arity n and $T_1, ..., T_n \in \mathbb{TERM}(\mathscr{A})$
- The set $\mathbb{WFF}(\mathscr{A})$ of *formulae* from $\mathscr{A}$ is defined by the following grammar, in which $X \in \mathbb{V}$:

$$\Phi := A \in \mathbb{ATOMS}(\mathbb{A}) \mid \Phi \Rightarrow \Phi \mid \Phi \,\&\, \Phi \mid \Phi \parr \Phi \mid \bot \mid \forall X \Phi \mid \exists X \Phi.$$

The symbols $\Rightarrow$, $\&$, $\parr$ and $\bot$ are *implication*, *conjunction*, *disjunction* and *absurdity*, respectively, in FOL. The more traditional symbols, such as $\rightarrow$, $\wedge$ and $\vee$, are reserved for other logics in the paper. We use the usual convention for suppressing brackets; i.e. conjunction ($\&$) and disjunction ($\parr$) bind more strongly than implication ($\Rightarrow$). Moreover, we may use the usual auxiliary terminology for first-order languages (e.g. *sub-formula*, *closed-formula*, *sentence*, etc.) without further explanation. Let be X a variable, T be a term and Φ a wff; we write $\Phi[X \mapsto T]$ to denote the result of replacing every free occurrence of X by the term T so that no variable in T becomes bound in Φ after the substitution.

DEFINITION 3.2 (First-order Sequent).
A first-order sequent is a pair $\Pi \triangleright \Sigma$ in which Π and Σ are multisets of first-order formulae.

We think of sequents as unjudged statements, hence we use a sequent constructor $\triangleright$ instead of the consequence relation ($\blacktriangleright$). For example, though $\varnothing \triangleright \varnothing$ is a well-formed sequent in FOL, it is not a consequence of the logic.

3.1.1 Proof Theory One way to characterize FOL—i.e. the consequence relation $\blacktriangleright$—is by provability in a sequent calculus.

DEFINITION 3.3 (Sequent Calculus G3c).
The sequent calculus G3c is composed of the rules in Figure 3 in which T is a term free for X in Φ and Y is an eigenvariable.

$$\frac{}{\Phi, \Pi \triangleright \Sigma, \Phi}\ \mathsf{ax} \qquad \frac{}{\bot, \Pi \triangleright \Sigma}\ \bot$$

$$\frac{\Phi, \Psi, \Pi \triangleright \Sigma}{\Phi \,\&\, \Psi, \Pi \triangleright \Sigma}\ \&_{\mathsf{L}} \qquad \frac{\Pi \triangleright \Sigma, \Phi \quad \Pi \triangleright \Sigma, \Psi}{\Pi \triangleright \Sigma, \Phi \,\&\, \Psi}\ \&_{\mathsf{R}}$$

$$\frac{\Phi, \Pi \triangleright \Sigma \quad \Psi, \Pi \triangleright \Sigma}{\Phi \parr \Psi, \Pi \triangleright \Sigma}\ \parr_{\mathsf{L}} \qquad \frac{\Pi \triangleright \Sigma, \Phi, \Psi}{\Pi \triangleright \Sigma, \Phi \parr \Psi}\ \parr_{\mathsf{R}}$$

$$\frac{\Pi \triangleright \Sigma, \Phi \quad \Psi, \Pi \triangleright \Sigma}{\Phi \Rightarrow \Psi, \Pi \triangleright \Sigma}\ \Rightarrow_{\mathsf{L}} \qquad \frac{\Phi, \Pi \triangleright \Sigma, \Psi}{\Pi \triangleright \Sigma, \Phi \Rightarrow \Psi}\ \Rightarrow_{\mathsf{R}}$$

$$\frac{\forall X \Phi, \Phi[X \mapsto T], \Pi \triangleright \Sigma}{\forall X \Phi, \Pi \triangleright \Sigma}\ \forall_{\mathsf{L}} \qquad \frac{\Pi \triangleright \Sigma, \Phi[X \mapsto Y]}{\Pi \triangleright \Sigma, \forall X \Phi}\ \forall_{\mathsf{R}}$$

$$\frac{\Phi[X \mapsto Y], \Pi \triangleright \Sigma}{\exists X \Phi, \Pi \triangleright \Sigma}\ \exists_{\mathsf{L}} \qquad \frac{\Pi \triangleright \Sigma, \exists X \Phi, \Phi[X \mapsto T]}{\Pi \triangleright \Sigma, \exists X \Phi}\ \exists_{\mathsf{R}}$$

FIGURE 3. Sequent Calculus G3c.

We write $\Pi \vdash_{G3c} \Sigma$ to denote that there is a G3c-proof of $\Pi \triangleright \Sigma$. Troelstra and Schwichtenberg [68] proved that G3c-provability characterizes classical consequence:

PROPOSITION 3.4
Let Π and Σ be multisets of formulae,

$$\Pi \blacktriangleright \Sigma \qquad \text{iff} \qquad \Pi \vdash_{G3c} \Sigma.$$

We have chosen to use G3c to characterize FOL, as opposed to other proof systems, because of its desirable proof-theoretic properties—e.g. Troelstra and Schwichtenberg [68] have shown that the rules of the calculus are (height-preserving) invertible, and that the following rules are admissible:

$$\frac{\Pi \triangleright \Sigma}{\Phi, \Pi \triangleright \Sigma}\ \mathsf{w}_{\mathsf{L}} \quad \frac{\Pi \triangleright \Sigma}{\Pi \triangleright \Sigma, \Phi}\ \mathsf{w}_{\mathsf{R}} \quad \frac{\Phi, \Phi, \Pi \triangleright \Sigma}{\Phi, \Pi \triangleright \Sigma}\ \mathsf{c}_{\mathsf{L}} \quad \frac{\Pi \triangleright \Sigma, \Phi, \Phi}{\Pi \triangleright \Sigma, \Phi}\ \mathsf{c}_{\mathsf{R}}.$$

3.1.2 Model-theoretic Semantics. Another way to characterize FOL is by validity in its model-theoretic semantics. As mentioned above, we assume familiarity with the subject and therefore give a terse but complete account of definitions to keep the paper self-contained.

DEFINITION 3.5 (First-order Structure).
A first-order structure is a tuple $\mathcal{S} = \langle \mathbb{U}, \mathbb{R}, \mathbb{F}, \mathbb{K} \rangle$ in which $\mathbb{U}$ is a countable set of elements, $\mathbb{K} \subseteq \mathbb{U}$, $\mathbb{F}$ is a countable set of operators on $\mathbb{U}$ (i.e. endomorphisms $f : \mathbb{U}^n \to \mathbb{U}$, for finite n) and $\mathbb{R}$ is a countable set of relations on $\mathbb{U}$.

$\mathfrak{M} \Vdash R(T_1, ..., T_n)$	iff	$\langle [\![T_1]\!], ..., [\![T_n]\!] \rangle \in [\![P]\!]$
$\mathfrak{M} \Vdash \Phi \Rightarrow \Psi$	iff	not $\mathfrak{M} \Vdash \Phi$ or $\mathfrak{M} \Vdash \Psi$
$\mathfrak{M} \Vdash \Phi \,\&\, \Psi$	iff	$\mathfrak{M} \Vdash \Phi$ and $\mathfrak{M} \Vdash \Psi$
$\mathfrak{M} \Vdash \Phi \parr \Psi$	iff	$\mathfrak{M} \Vdash \Phi$ or $\mathfrak{M} \Vdash \Psi$
$\mathfrak{M} \Vdash \bot$	iff	never
$\mathfrak{M} \Vdash \forall X \Phi$	iff	$\mathfrak{M} \Vdash \Phi[X \mapsto T]$ for any $T \in \mathbb{TERM}(\mathscr{A})$
$\mathfrak{M} \Vdash \exists X \Phi$	iff	$\mathfrak{M} \Vdash \Phi[X \mapsto T]$ for some $T \in \mathbb{TERM}(\mathscr{A})$

FIGURE 4. Truth in an Abstraction.

DEFINITION 3.6 (Interpretation).
Let $\mathcal{S} := \langle \mathbb{U}, \mathbb{R}, \mathbb{F}, \mathbb{K} \rangle$ be a first-order structure, and let $\mathscr{A} := \langle \mathbb{R}', \mathbb{F}', \mathbb{K}', \mathbb{V} \rangle$ be an alphabet. An interpretation of $\mathscr{A}$ in $\mathcal{S}$ is a function $[\![-]\!]$ satisfying the following:

- if $X \in \mathbb{V}$, then $[\![X]\!] \in \mathbb{U}$;
- if $C \in \mathbb{K}'$, then $[\![C]\!] \in \mathbb{K}$;
- if $F \in \mathbb{F}'$, then $[\![F]\!] \in \mathbb{F}$, and the arity of $[\![F]\!]$ is the arity of F;
- if $R \in \mathbb{R}'$, then $[\![R]\!] \in \mathbb{R}$, and the arity of $[\![R]\!]$ is the arity of R.

We may write $[\![-]\!] : \mathscr{A} \to \mathcal{S}$ to denote that $[\![-]\!]$ is an interpretation of $\mathscr{A}$ in $\mathcal{S}$. Interpretations extend to terms as follows:

$$[\![F(T_1, ..., T_n)]\!] := [\![F]\!]([\![T_1]\!], ..., [\![T_n]\!]).$$

In this paper, we use the term *abstraction* for what is traditionally referred to as a *model*. This is to avoid confusion as we consider the semantics of various propositional logics in subsequent sections, where the term *model* will be significant.

DEFINITION 3.7 (Abstraction).
An abstraction of an alphabet $\mathscr{A}$ is a pair $\mathfrak{A} := \langle \mathcal{S}, [\![-]\!] \rangle$ in which $\mathcal{S}$ is a structure and $[\![-]\!] : \mathscr{A} \to \mathcal{S}$ is an interpretation.

DEFINITION 3.8 (Truth in an Abstraction).
Let $\mathscr{A}$ be an alphabet, let ϕ be a formula over $\mathscr{A}$ and let $\mathfrak{A} = \langle \mathcal{S}, [\![-]\!] \rangle$ be an abstraction of $\mathscr{A}$. The formula ϕ is true in $\mathfrak{A}$ iff $\mathfrak{A} \Vdash\!\!\Vdash \phi$, which is defined inductively by the clauses in Figure 4.

We may extend the truth of formulae in an abstraction to the truth of (multi-)sets of formulae by requiring that all the elements in the set are true in the abstraction—i.e. if $\mathfrak{A}$ is a model and Π is a multiset of formulae,

$$\mathfrak{A} \Vdash\!\!\Vdash \Pi \qquad \text{iff} \qquad \mathfrak{A} \Vdash\!\!\Vdash \Phi \text{ for every } \Phi \in \Pi.$$

Gödel [31]—see also van Dalen [69]—proved that abstractions characterize FOL:

PROPOSITION 3.9
Let Π and Σ be multisets of formulae,

$$\Pi \blacktriangleright \Sigma \quad \text{iff} \quad \begin{array}{l}\text{for any abstraction } \mathfrak{A}, \text{ if } \mathfrak{A} \Vdash \phi \text{ for any } \phi \in \Pi, \\ \text{then there is } \psi \in \Sigma \text{ such that } \mathfrak{A} \Vdash \psi.\end{array}$$

This concludes the summary of FOL.

3.2 Propositional Logic

There is no consensus in the literature on what propositional logic means. This paper uses a relatively broad definition that captures the most common propositional logics (e.g. classical propositional logic, IPL, modal logics, linear logics, bunched logics, etc.). We include context-formers explicitly as a part of the language of propositional logics. More precisely, we include *data*-formers as they may appear on either the left or right of sequents for the propositional logics, and we use 'context' to refer to the left of sequents. This is useful for two reasons: first, it enables us to move between the propositional logics without ambiguity; second, it enables us to handle propositional logics that are expressed in terms of more complex data structures of formulae than lists, multisets or sets, such as the family of relevance logics (see e.g. Read [63]) and the family of bunched logics (see e.g. the work by Docherty, O'Hearn and Pym [15, 33, 59]). Throughout, we give a running example of normal modal logics, which relates the work of this paper to that of Negri [56].

DEFINITION 3.10 (Propositional Alphabet).
A propositional alphabet is a tuple of three elements $\mathscr{P} := \langle \mathbb{A}, \mathbb{O}, \mathbb{C} \rangle$ such that $\mathbb{A}$, $\mathbb{O}$ and $\mathbb{C}$ are pairwise disjoint sets of symbols such that $\mathbb{A}$ is countable and $\mathbb{O}$ and $\mathbb{C}$ are finite. The symbols in $\mathbb{O}$ and $\mathbb{C}$ have a fixed arity.

The elements of $\mathbb{A}$ are *atomic propositions*, the elements of $\mathbb{O}$ are *operators* and the elements of $\mathbb{C}$ are *data-constructors*. We use the term *operators* to subsume 'connectives' and 'modalities' in the traditional terminology. Similarly, we use the term 'data-constructor' as a neutral term for what is sometimes called a 'context-former' as we shall have data both on the left and right of sequents with possibly different constructors and reserve the term 'context' for the left-hand side.

DEFINITION 3.11 (Formula, data, sequent).
Let $\mathscr{P} := \langle \mathbb{A}, \mathbb{O}, \mathbb{C} \rangle$ be a propositional alphabet. The set of formulae, data and sequents from $\mathscr{P}$ are as follows:

- The set of propositional formulae $\mathbb{FORM}(\mathscr{P})$ is the smallest set containing $\mathbb{A}$ such that, for any $\phi_1, ..., \phi_k \in \mathbb{FORM}(\mathscr{P})$ and $\circ \in \mathbb{O}$, if $\circ$ has arity n, then $\circ(\phi_1, ..., \phi_n) \in \mathbb{FORM}(\mathscr{P})$
- The set $\mathbb{DATA}(\mathscr{P})$ is the smallest set containing $\mathbb{FORM}(\mathscr{P})$ such that, for any $\delta_1, ..., \delta_n \in \mathbb{DATA}(\mathscr{P})$ and $\bullet \in \mathbb{C}$, if $\bullet$ has arity n, then $\bullet(\delta_1, ..., \delta_n) \in \mathbb{DATA}(\mathscr{P})$
- A $\mathscr{P}$-sequent is a pair $\Gamma \triangleright \Delta$ in which $\Gamma, \Delta \in \mathbb{DATA}(\mathscr{P})$.

EXAMPLE 3.12
The basic modal alphabet is $\mathscr{B} = \langle \mathbb{A}, \{\wedge, \vee, \neg, \Box\}, \{\varnothing, ,, ;\} \rangle$. The arities of $\wedge$, $\vee$, , and ; are 2; the arities of $\neg$ and $\Box$ are 1; and the arity of $\varnothing$ is 0. We may write $\phi \supset \psi$ to abbreviate $\neg\phi \vee \psi$.

Let $p_1, p_2, p_3 \in \mathbb{A}$. Using infix notation, the following are examples of elements from $\mathbb{FORM}(\mathscr{B})$:

$$p_3 \qquad (p_1 \wedge p_2) \qquad (p_3 \supset (p_1 \wedge p_2))$$

As well as being elements of $\mathbb{FORM}(\mathscr{B})$, they are also elements in $\mathbb{DATA}(\mathscr{B})$.

Another example of an element from $\mathbb{DATA}(\mathscr{B})$ is the following:

$$p_3 \circ (p_3 \supset (p_1 \wedge p_2))$$

The following is an example of a $\mathscr{B}$-sequent:

$$p_3 \circ p_3 \supset (p_1 \wedge p_2) \rhd p_1 \wedge p_2$$

This completes the definition of the language of a propositional logic generated by an alphabet. What makes language into a logic is a notion of consequence.

DEFINITION 3.13 (Propositional Logic).
Let $\mathscr{A}$ be a propositional alphabet. A propositional logic over $\mathscr{A}$ is a relation $\vdash$ over $\mathscr{A}$-sequents.

The relation $\vdash$ is called the *consequence judgment* of the logic; its elements are *consequences*. We write $\Gamma \vdash \Delta$ to denote that the sequent $\Gamma \rhd \Delta$ is a consequence. This definition of propositional logic needs more sophistication in many regards but the point is not to satisfy the doxastic interpretation of what constitutes a logic. Interesting though that may be, it amounts to refining the current definition. What is given here suffices for present purposes and encompasses the vast array of propositional logics in the literature.

3.2.1 Proof Theory In this paper, we are concerned about the proof-theoretic characterization of a logic and what it tells us about that logic. Fix a propositional alphabet $\mathscr{P}$.

DEFINITION 3.14 (Sequent calculus).
A rule r over $\mathscr{P}$-sequents is a (non-empty) relation on $\mathscr{P}$-sequents; a rule with arity one is an axiom. A sequent calculus is a set of rules at smallest one of which is an axiom.

Note that $\mathsf{LBI}_{\mathcal{B}}$ in Section 2 does not contain axioms and is, therefore, not a sequent calculus according to this definition. This is because we regard it as a *constraint system* in which axioms are not necessary—see Section 4.

We have not defined rules by rule-figures and do not assume they are necessarily closed under substitution. This allows us to speak of rules with side-conditions—see e.g. $\mathsf{e} \in \mathsf{LBI}$ in Section 2. Of course, we will otherwise follow standard conventions—see e.g. Troelstra and Schwichtenberg [68].

Let r be a rule, the situation $\mathsf{r}(C, P_1, ..., P_n)$ may be denoted as follows:

$$\frac{P_1 \quad \dots \quad P_n}{C}\ \mathsf{r}.$$

In such instances, the string C is called the conclusion, and the strings $P_1, ..., P_n$ are called the premisses.

EXAMPLE 3.15 (Example 3.12 cont'd).
The rule $\wedge_{\mathsf{R}}$ over basic modal sequents is defined by the following figure without any side-conditions:

$$\frac{\Gamma \rhd \Delta \circ \phi \quad \Gamma \rhd \Delta \circ \psi}{\Gamma \rhd \Delta \circ \phi \wedge \psi}\ \wedge_{\mathsf{R}}.$$

This rule is admissible for the normal modal logic K—see Blackburn *et al.* [5]. That is, let $\vdash_K$ be the consequence relation for K (over the basic modal alphabet $\mathscr{B}$): if $\Gamma \vdash_K \Delta, \phi$ and $\Gamma \vdash_K \Delta, \psi$, then $\Gamma \vdash_K \Delta, \phi \wedge \psi$.

DEFINITION 3.16 (Derivation).
Let L be a sequent calculus. The set of L-derivations is defined inductively as follows:

- BASE CASE. If C is a $\mathscr{P}$-sequent, then the tree consisting of just the node C is an L-derivation.
- INDUCTIVE STEP. Let $\mathcal{D}_1, ..., \mathcal{D}_n$ are L-derivations, with roots $P_1, ..., P_n$, respectively, and let $\mathsf{r} \in \mathsf{L}$ be a rule such that $\mathsf{r}(C, P_1, ..., P_n)$ obtains. The tree with root C and immediate sub-trees $\mathcal{D}_1, ..., \mathcal{D}_n$ is a L-derivation.

DEFINITION 3.17 (Proof).
Let L be a sequent calculus. An L-derivation $\mathcal{D}$ is a proof iff the leaves of $\mathcal{D}$ are instances of axioms of L.

We write $\Gamma \vdash_L \Delta$ to denote that there is a L-proof of the sequent $\Gamma \triangleright \Delta$. A sequent calculus may have the following relationships to a propositional logic ($\vdash$):

- Soundness: If $\Gamma \vdash_L \Delta$, then $\Gamma \vdash \Delta$.
- Completeness: If $\Gamma \vdash \Delta$, then $\Gamma \vdash_L \Delta$.

In saying that L is a sequent calculus for a propositional logic (i.e. that it characterizes that logic), we assert that L is sound and complete for that logic.

EXAMPLE 3.18 (Example 3.15 cont'd).
We may characterize modal logics, including K, by axiom systems—see e.g. Blackburn *et al.* [5]. Each such system A is a sequent calculus in the general sense of this paper.

This concludes the proof-theoretic account of propositional logics in this paper.

3.2.2 Model-theoretic Semantics We now give a generic account of model-theoretic semantics (M-tS) that can define a logic over a propositional language. By M-tS we mean a frame semantics *à la* Kripke [42, 43]—see also Beth [3]. We follow Blackburn *et al.* [5] in the approach for a general account of M-tS.

DEFINITION 3.19 (Type).
A type τ is a list of non-negative integers.

DEFINITION 3.20 (Frame, Assignment, Pre-model).
Let $\tau := \langle t_1, ..., t_n \rangle$ be a type. A τ-frame is a tuple $\langle \mathbb{U}, k_1, ..., k_n \rangle$ in which $\mathbb{U}$ is a set and k_i is a relation on $\mathbb{U}$ of arity t_i. Let $\mathscr{P} = \langle \mathbb{A}, \mathbb{O}, \mathbb{C} \rangle$ be a propositional alphabet. An assignment of $\mathscr{P}$ to $\mathcal{F}$ is a mapping from propositional atoms to sets of worlds, $I : \mathbb{A} \to \wp(\mathbb{U})$. A τ-pre-model over $\mathscr{P}$ is a pair $\mathfrak{M} := \langle \mathcal{F}, I \rangle$, in which $\mathcal{F}$ is a τ-frame and I is an assignment of $\mathscr{P}$ to $\mathcal{F}$.

The elements of $\mathbb{U}$ are called possible worlds. One possible objection to the definition of a frame is the absence of operators (i.e. endomorphisms $f : \mathbb{U}^n \to \mathbb{U}$). This is to simplify the setup and

is without loss of generality as operators may be regarded as particular types of relations; i.e. the operator $f : \mathbb{U}^n \to \mathbb{U}$ corresponds to the $(n+1)$-ary relation R satisfying $R(w, u_1, ..., u_n)$ iff $w = f(u_1, ..., u_n)$.

Intuitively, a formula ϕ is true in a model $\mathfrak{M}$ at a world w if the world w satisfies the formulas.

DEFINITION 3.21 (Satisfaction for a Type).
Let τ be a type and $\mathscr{P}$ a propositional alphabet. A τ-satisfaction relation for $\mathscr{P}$ is a relation $\Vdash$ parameterized by τ-pre-models between worlds w in the pre-models $\mathfrak{M} = \langle \mathcal{F}, I \rangle$ and $\mathscr{P}$-data such that the following holds:

$$\mathfrak{M}, w \Vdash \mathrm{p} \qquad \text{iff} \qquad w \in I(\mathrm{p}).$$

DEFINITION 3.22 (Semantics).
Let τ be a type and $\mathscr{P}$ a propositional alphabet. A semantics is a pair $\mathfrak{S} := \langle \mathbb{M}, \Vdash \rangle$ in which $\mathbb{M}$ is a set of τ-pre-models and $\Vdash$ is a τ-satisfaction relation for $\mathscr{P}$.

DEFINITION 3.23
Let $\mathfrak{S}$ be a semantics. A sequent $\Gamma \triangleright \Delta$ is valid in $\mathfrak{S}$—denoted $\Gamma \vDash_{\mathfrak{S}} \Delta$—iff, for any $\mathfrak{M} \in \mathbb{M}$ and any $w \in \mathfrak{M}$, if $\mathfrak{M}, w \Vdash \Gamma$, then $\mathfrak{M}, w \Vdash \Delta$.

EXAMPLE 3.24 (Example 3.18 cont'd).
Fix the type $\tau := \langle 2 \rangle$. An example of a τ-frame is a pair $\langle \{x, y\}, R \rangle$ in which R is a binary relation on $\{x, y\}$. Partition the atoms $\mathbb{A}$ into two classes $\mathbb{A}_1$ and $\mathbb{A}_2$; an example of an assignment $I : \mathbb{A} \to \wp(\{x, y\})$ is given as follows:

$$I(\mathrm{p}) := \begin{cases} x & \text{if } \mathrm{p} \in \mathbb{A}_1 \\ y & \text{if } \mathrm{p} \in \mathbb{A}_2. \end{cases}$$

The pair $\mathfrak{M} := \langle \mathcal{F}, I \rangle$ is an example of a model over the basic modal alphabet $\mathscr{B}$. The basic semantics $\mathfrak{K}$ is the pair $\langle \mathbb{K}, \Vdash \rangle$ in which $\mathbb{K}$ is the set of all τ-pre-models and $\Vdash$ is the smallest relation satisfying the clauses in Figure 5 together with the following:

$$\begin{array}{lll} \mathfrak{M}, w \Vdash \Delta \, , \, \Delta' & \text{iff} & \mathfrak{M}, w \Vdash \Delta \text{ and } \mathfrak{M}, w \Vdash \Delta' \\ \mathfrak{M}, w \Vdash \Delta \, ; \, \Delta' & \text{iff} & \mathfrak{M}, w \Vdash \Delta \text{ or } \mathfrak{M}, w \Vdash \Delta'. \end{array}$$

The validity judgment $\vDash_{\mathfrak{K}}$ defines the modal logic K—see e.g. Kripke [42], Blackburn *et al.* [5] and Fitting and Mendelsohn [20].

The significance of $\mathbb{M}$ in the definition of a semantics is that one may not want to consider all pre-models but instead require them to satisfy a specific condition—see e.g. the persistence condition for the semantics of IPL in Section 6.

The notion of semantics in this paper is generous, including many relations that one would not typically accept as semantics. This is to keep the presentation simple and intuitive. In the next section, we restrict attention to satisfaction relations that admit particular presentations that enable us to analyse them, but doing so presently would obscure the setup.

Historically, the *a priori* definition of a consequence relation has been by validity in a semantics. In this paper, we only work with logics for which we assume there is a sequent calculus. Therefore,

$\mathfrak{M}, w \Vdash \mathrm{p}$	iff	$w \in I(\mathrm{p})$
$\mathfrak{M}, w \Vdash \phi \wedge \psi$	iff	$\mathfrak{M}, w \Vdash \phi$ and $\mathfrak{M}, w \Vdash \psi$
$\mathfrak{M}, w \Vdash \phi \vee \psi$	iff	$\mathfrak{M}, w \Vdash \phi$ or $\mathfrak{M}, w \Vdash \psi$
$\mathfrak{M}, w \Vdash \neg\phi$	iff	not $\mathfrak{M}, w \Vdash \phi$
$\mathfrak{M}, w \Vdash \Box\phi$	iff	for any u, if wRu, then $\mathfrak{M}, u \Vdash \phi$

FIGURE 5. Satisfaction for K.

we may use the nomenclature of Section 3.2 to relate entailment to consequence via provability. A sequent calculus L may have the following relationship to a semantics $\mathfrak{S}$:

- Soundness: If $\Gamma \vdash_L \Delta$, then $\Gamma \models_{\mathfrak{S}} \Delta$.
- Completeness: If $\Gamma \models_{\mathfrak{S}} \Delta$, then $\Gamma \vdash_L \Delta$.

This completes the summary of propositional logics. Moreover, it completes the technical background to this paper. There are various judgments present, whose relationship are important for the rest of the paper. In the beginning of the next section, we provide a brief summary of how all this background is used before proceeding with the technical work.

4 Constraint Systems

This section provides a formal definition of constraint systems. Briefly, a *constraint system* is a sequent calculus in which the data may carry labels representing expressions over some algebra. Rules may manipulate those expressions or demand constraints on them. At the end of a construction in a constraint system, one checks to see that the constraints are coherent and admit an interpretation in the intended algebra. This generalizes the setup of RDvBC in Section 2 to an arbitrary algebra and an arbitrary propositional logic. In the next section (Section 5), we provide a method for producing constraint systems in a modular way, and in the one after (Section 6), we illustrate their use in studying model theory.

The work in this section is technical and abstract; therefore, we give a brief overview summarizing the main ideas. We begin with a propositional logic $\vdash$ over an alphabet $\mathscr{P}$ and a sequent calculus L, which has some desirable features but that is not immediately related to the logic—e.g. in Section 2, we had BI as the propositional logic and L a version of LJ in which contexts are maintained during reduction. We then introduce an algebra, which we understand in terms of first-order structures $\mathcal{A}$, and present in terms of a first-order alphabet $\mathscr{A}$—e.g. see the presentation of Boolean algebra in Section 2. We fuse $\mathscr{P}$ and $\mathscr{A}$ creating a language $\mathscr{P} \oplus \mathscr{A}$ in which the algebra is used to label $\mathscr{P}$-data. We introduce a valuation map ν_I, parameterized by interpretations $I : \mathscr{A} \to \mathcal{A}$, which maps $\mathscr{P} \oplus \mathscr{A}$-data to $\mathscr{P}$-data. This defines the action the algebra has on the data. A constraint system is a generalized notion of a sequent calculus of $\mathscr{P} \oplus \mathscr{A}$-sequents, which may have $\mathscr{A}$ expressions as local or global constraints on the correctness of inferences—see e.g. $\mathsf{LBI}_{\mathcal{B}}$ in Section 2. Finally, we give correctness conditions for constraint systems: first, a constraint system C is sound and complete, relative to ν_I, for the logic $\vdash$ iff its constructions witness all and only the consequence of the logic; second, a constraint system C is faithful and adequate, relative to ν_I, for a sequent calculus L' iff

its constructions witness all and only L'-proofs. Throughout the rest of the paper, we illustrate how constraint systems with these correctness conditions aid in studying logic.

The section is composed of three parts. Section 4.1 explains the paradigmatic shift necessary for constraint systems: one constructs proofs upward rather than downward. In Section 4.2, we define constraint systems formally as the enrichment of a sequent calculus by an algebra of constraints. Finally, in Section 4.3, we define correctness conditions relating constraint systems to logics and their proof-theoretic formulations with sequent calculi.

4.1 Reductive logic

The traditional paradigm of logic proceeds by inferring a conclusion from established premisses using an *inference rule*. This is the paradigm known as *deductive logic*:

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

In contrast, the experience of the use of logic is often dual to deductive logic in the sense that it proceeds from a putative conclusion to a collection of premisses that suffice for the conclusion. This is the paradigm known as *reductive logic*:

$$\frac{\text{Sufficient Premiss}_1 \quad \ldots \quad \text{Sufficent Premiss}_n}{\text{Putative Conclusion}} \Uparrow.$$

Rules used backward in this way are called *reduction operators*. The objects created using reduction operators are called *reductions*. We believe that this idea of reduction was first explained in these terms by Kleene [39]. There are many ways of studying reduction, and a number of models have been considered, especially in the case of classical and intuitionistic logic—see e.g. Pym and Ritter [61].

Historically, the deductive paradigm has dominated since it exactly captures the meaning of truth relative to some set of axioms and inference rules, and therefore is the natural point of view when considering *foundations of mathematics*. However, it is the reductive paradigm from which much of computational logic derives, including various instances of automated reasoning—see e.g. Kowalski [41], Bundy [8] and Milner [52].

Constraint systems (e.g. $\mathsf{LBI}_{\mathcal{B}}$) sit more naturally within the reductive perspective, with the intuition that one generates constraints as one applies rules backwards. Therefore, in constraint systems, when we *use* a rule we mean it in the reductive of sense.

Having given the overall paradigm on logic in which constraint systems are situated, we are now able to define them formally and uniformly.

4.2 Expressions, Constraints and Reductions

In Section 3.2, we defined what we mean by propositional logic. Recall that we may say *algebra* to mean a first-order structure (see Section 3.1). In this context, what we mean by expressions and constraints are terms and formulae, respectively, from an alphabet in which that algebra is interpreted.

Let $\mathscr{A}$ be a (first-order) alphabet.

DEFINITION 4.1 (Expression).
An $\mathscr{A}$-expression is a term over $\mathscr{A}$— an element of $\mathbb{TERM}(\mathscr{A})$.

DEFINITION 4.2 (Constraint).
An $\mathscr{A}$-constraint is a formula over $\mathscr{A}$—i.e. an element of $\mathbb{WFF}(\mathscr{A})$.

When it is clear that an alphabet has been fixed, we may elide it alphabet when discussing labelled formulae, labelled data and enriched sequents. We use the terms 'expression' and 'constraint' to draw attention to the fact that we have a certain algebra in mind and a certain way that the constants and functions of the alphabet are meant to be interpreted. For example, in Section 2, we always take symbol $+$ to always be interpreted as Boolean addition. What may change is the interpretation of variables. In short, we have some set of intended interpretations that are *coherent*.

DEFINITION 4.3 (Coherent Interpretations).
Let $\mathbb{I}$ be a set of interpretations of an algebra $\mathcal{A}$ in $\mathscr{A}$. The set $\mathbb{I}$ is coherent iff, for any $I_1, I_2 \in \mathbb{I}$, they behave the same except possibly for atoms.

Typically, the set of intended interpretations is maximal in the sense that any interpretation of the algebra in the alphabet is either in the set or is not a variant of an interpretation in the set.

We use expressions to enrich the language of the propositional logic and thereby express meta-theoretic conditions on formulae and sequents. Let $\mathscr{P}$ be a propositional alphabet.

DEFINITION 4.4 (Labelled Data).
The set of labelled $\mathscr{P}$-data is defined inductively as follows:

- BASE CASE. If ϕ is a formula and e is an $\mathscr{A}$-expression, then $\phi \cdot e$ is a $\mathscr{A}$-labelled $\mathscr{P}$-datum.
- INDUCTIVE STEP. If δ_1,..., δ_n are labelled $\mathscr{P}$-data, $\bullet$ is a data-constructor in $\mathscr{P}$ with arity n, and e is an $\mathscr{A}$-expression, then $\bullet(\delta_1, ..., \delta_n) \cdot e$ is a $\mathscr{A}$-labelled $\mathscr{P}$-datum.

DEFINITION 4.5 (Enriched sequent).
An $\mathscr{A}$-enriched $\mathscr{P}$-sequent is a pair $\Pi \triangleright \Sigma$, in which Π and Σ are multisets of $\mathscr{A}$-labelled $\mathscr{P}$-data and constraints.

We may suppress $\mathscr{A}$ and $\mathscr{P}$ when it is clear what alphabet for what algebra is labeling what propositional language. Observe that we have shifted from the object-logic to the meta-logic; i.e. enriched sequents are a restricted form of meta-logic sequents that encapsulate object-logic sequents with conditions expressed by expressions from the algebra. This setup differs slightly from the presentation of RDvBC in Section 2 to simplify presentation of the general case. Recall that data is the general name for contexts in the propositional logic, which may be bunches. Consequently, the presentation of RDvBC in terms of enriched sequents would consist of pairs of multisets each of which contain only one element, the labelled bunch.

EXAMPLE 4.6
The are various enriched sequents in RDvBC—see Section 2. An additional example is as follows:

$$\mathrm{p} \cdot x \, , \, (q \, ; \, r) \cdot y \triangleright (\mathrm{p} \wedge \mathrm{q}) \cdot x.$$

A *constraint system* is a generalization of sequent calculus that uses enriched sequents and constraints. The constructions of a constraint system are generated co-recursively on enriched sequents, producing constraints along the way along.

DEFINITION 4.7 (Constraint System).
A constraint rule is a relation between an enriched sequents and a list of enriched sequents and constraints. A constraint system is a set of constraint rules.

We use the same notation as in Section 3.2 for constraint rules; i.e. the situation $\mathsf{r}(C, P_1, ..., P_n)$ may be expressed as follows:

$$\frac{P_1 \quad ... \quad P_n}{C}\ \mathsf{r}.$$

In this case, C is an enriched sequent and $P_1, ..., P_n$ are either enriched sequents or constraints. The terms *premiss* and *conclusion* are analogous to those employed for sequent calculus rules in Section 3.2. We assume the convention of putting constraints after enriched sequents in the list of premisses.

EXAMPLE 4.8 (Example 4.6 cont'd).
System $\mathsf{LBI}_{\mathcal{B}}$ in Section 2 is a constraint system. Therefore, any rule in it is an example of a constraint rule. We shall consider two examples.

The following is a constraint rule:

$$\frac{\Delta \cdot V \triangleright \phi \quad \Delta' \cdot \bar{V} \triangleright \psi}{\Delta \,;\, \Delta' \triangleright \phi * \psi}\ *_{\mathsf{R}}.$$

If e is a label on Δ, then $\Delta \cdot V$ denotes the result of replacing e by a product of e and V. The following inference is an instance of the rule:

$$\frac{\mathrm{p} \cdot (1x) \,;\, (q \,⨟\, r) \cdot (1y) \triangleright \mathrm{p} \cdot 1 \quad \mathrm{p} \cdot (1\bar{x}) \,;\, (q \,⨟\, r) \cdot (1\bar{y}) \triangleright \mathrm{q} \cdot 1}{\mathrm{p} \cdot 1 \,;\, (q \,⨟\, r) \cdot 1 \triangleright (\mathrm{p} * \mathrm{q}) \cdot 1}.$$

Another example of a constraint rule is as follows:

$$\frac{y = 1 \wedge V = 0}{\phi \cdot y \,;\, \Delta \cdot V \triangleright \phi \cdot 1}\ \mathsf{taut}.$$

Here V are all the labels on Δ and $V = 0$ denotes $x_1 = 0 \& .... \& x_n = 0$ if V is the list $x_1, ..., x_n$. An instance of the rule is the following:

$$\frac{1x = 1 \quad 1y = 0}{p \cdot (1x) \,;\, (q \,⨟\, r) \cdot 1y \triangleright p \cdot 1.}$$

Intuitively, this corresponds to an *axiom* of a sequent calculus as it reduces to constraints, which, when solved, state weather or not the sequent in the conclusion is a consequence of BI.

Unlike sequent calculi, a constraint system does not necessarily contain axioms. This is possible because the set of things generated by a constraint system is defined co-inductively, so the restriction is unnecessary. We define reductions co-inductively because constraint systems sit within the paradigm of reductive logic. The reductions in this paper are all finite.

DEFINITION 4.9 (Reduction in a Constraint System).
Let C be a constraint system and let S be an enriched sequent. A tree of enriched sequents $\mathcal{R}$ is a C-reduction of S iff there is a rule $\mathsf{r} \in \mathsf{C}$ such that $\mathsf{r}(S, P_1, .., P_n)$ obtains and the immediate

sub-trees R_i, with root P_i, are as follows: if P_i is an enriched sequent, it is a C-reduction of P_i; the single node P_i, otherwise (i.e. if P_i is a constraint).

EXAMPLE 4.10 (Example 4.8 cont'd).
A reduction in a constraint system (the system $\mathsf{LBI}_{\mathcal{B}}$) is given in Example 2.2.

This concludes the definition of a constraint system and reduction from it. It remains to give conditions defining how it relates to logics.

4.3 Correctness Conditions

The distinguishing feature of reductions in a constraint system is the constraints. These constraints are understood as correctness conditions in two ways. They are *global* correctness conditions when they determine that a completed reduction is a valid certificate witnessing that some sequent is the consequence of a logic; this is, global correctness means *soundness* and *completeness*. They are *local* correctness conditions when they determine that each reduction step corresponds to a valid inference (i.e. an admissible inference) in the logic; this is, local correctness means *faithfulness* and *adequacy* with respect to some sequent calculus for the logic. Of course, local correctness implies global correctness. In either reading, we regard constraints in using rules as side-conditions on the reduction.

DEFINITION 4.11 (Side-condition).
Let C be a constraint system and let $\mathcal{R}$ be a C-reduction. A side-condition of $\mathcal{R}$ is a constraint that is a leaf of $\mathcal{R}$.

The side-conditions are global constraints on the reduction, determining the conditions for which the structure is meaningful.

DEFINITION 4.12 (Coherent Reduction).
Let C be a constraint system, let $\mathcal{R}$ be a C-reduction and let $\mathbb{S}$ be the set of side-conditions of $\mathcal{R}$. The set $\mathbb{S}$ is coherent iff there is an interpretation in which all of the side-conditions in $\mathbb{S}$ are valid; the reduction $\mathcal{R}$ is coherent iff $\mathbb{S}$ is coherent.

We may regard coherent reductions as proofs of certain sequents, but this requires a method of reading what sequent of the propositional logic the reduction asserts.

DEFINITION 4.13 (Ergo).
An ergo is a map ν_I, parameterized by intended interpretations, from enriched sequents to sequents.

Let C be a constraint system and ν an ergo. We write $\Gamma \vdash^{\nu}_{C} \Delta$ to denote that there is a coherent C-reduction $\mathcal{R}$ of an enriched sequent S such that $\nu_I(S) = \Gamma \triangleright \Delta$, where I is an interpretation satisfying all the side-conditions of $\mathcal{R}$. An example of this is the valuation given for RDvBC in Section 2 that deletes formulae and bunches labelled by an expression that evaluates to 0 and keeps those that evaluate to 1.

DEFINITION 4.14 (Soundness and completeness of constraint systems).
A constraint system C may have the following relationships to a propositional logic:

- Soundness: If $\Gamma \vdash_C^{\nu} \Delta$, then $\Gamma \vdash \Delta$.
- Completeness: If $\Gamma \vdash \Delta$, then $\Gamma \vdash_C^{\nu} \Delta$.

This defines constraint systems and their relationship to logics. Observe that soundness and completeness is a *global* correctness condition on reductions in the sense that only once the reduction has been completed and one has generated all of the constraints and solved them to find an interpretation does one know whether or not the reduction witnesses the validity of some sequent in the logic. In other words, a partial reduction (i.e. a reduction to which one may still apply rules) does not necessarily contain any proof-theoretic information about the logic.

In contrast, one may consider a *local* correctness conditions in which applying a reduction operator from a constraint system corresponds to using some rules of inference in a sequent calculus for the logic. This is stronger than the global correctness condition as when the reduction is completed, and the constraints are solved, the resulting interpretations that allow one to read a reduction in a constraint system as a proof in a sequent calculus for the logic, and thus as a certificate for the validity of some sequent.

Fix a propositional alphabet $\mathscr{P}$, an algebra $\mathcal{A}$, an alphabet $\mathscr{A}$ for that algebra and a set $\mathbb{I}$ of intended interpretations of $\mathscr{A}$ in $\mathcal{A}$. Fix a constraint system C and an ergo ν. The ergo extends to C-reductions by pointwise application to the enriched sequents in the tree and by deleting all the constraints.

Using this extension, constraint systems are computational devices capturing sequent calculi. For this reason, we do not use the terms *soundness* and *completeness*, but rather use the more computational terms of *faithfulness* and *adequacy*.

DEFINITION 4.15 (Faithful and Adequate).
Let C be a constraint system, let L be a sequent calculus and let ν be a valuation.

- System C is *faithful* to L if, for any C-reduction $\mathcal{R}$ and interpretation I satisfying the constraints of $\mathcal{R}$, the application $\nu_I(\mathcal{R})$ is an L-proof.
- System C is *adequate* for L if, for any L-proof $\mathcal{D}$, there is a C-reduction $\mathcal{R}$ and an interpretation I satisfying the constraints of R such that $\nu_I(\mathcal{R}) = \mathcal{D}$.

Intuitively, constraint systems for a logic (more precisely, constraint systems that are faithful and adequate with respect to a sequent calculus for a logic) separate combinatorial and idiosyncratic aspects of that logic. The former refers to how rules manipulate the data in sequents, while the latter refers to the constraints generated by the rules. Note that this gives a *local* correctness condition of reductions from a constraint system as each reductive inference in the constraint system corresponds to some reductive inference in a sequent calculus for the logic.

In the next section (Section 5), we provide sufficient conditions for a propositional logic to have a constraint system that evaluates to a sequent calculus for that logic. The conditions are quite encompassing and automatically give soundness and completeness for the sequent calculus for a semantics for the logic. Attempting a total characterization (i.e. precisely defining the properties a logic must satisfy in order for it to have a constraint system and a valuation to a sequent calculus for the logic) is unrealistic. The reason is that propositional logics, constraints systems, algebras and valuations have so many degrees of freedom that one could not plainly present their dependencies at once. Instead, one should consider such a classification relative to a fixed structure of propositional logics (e.g. substructural logics with data comprising multisets of formulae), for a fixed algebra (e.g. Boolean algebra), and fixed valuations (i.e. keeping formulae whose label evaluate to 1 and deleting formulae whose label evaluate to 0).

Significantly, having constructed a reduction in a constraint system, one must solve the constraints before one knows what 'proof' the reduction represents for the target logic. This may be challenging depending on the algebra over which the constraint take place. Nonetheless, with even relatively simple algebras, one may have relatively useful constraint systems; e.g. RDvBC uses Boolean algebra, which admits algorithmic solvers, that means one can outsource the solving of constraints for $\mathsf{LBI}_{\mathcal{B}}$-reductions. However, the point of constraint systems is not to *do* proof-search but to *study* proof-search. In this way, how the constraints are solved is less important than how they can be interpreted in terms of control during proof-search in the object-logic.

5 Example: Relational Calculi

The relational calculi introduced by Negri [56] can be viewed as constraint systems; i.e. the *constraint algebra* is provided by a first-order theory capturing an M-tS for a logic, and the labelling action captures satisfaction in that semantics. Traditionally, $x : \phi$ is used in place of $\phi \cdot x$ for relational calculi, and we shall adopted this notation for this section to be consistent with the existing work. The change in notation is a *aide-mémoire* that we are working with a particular form of constraint systems.

This section gives sufficient conditions for a sequent calculus to admit a relational calculus. We further give conditions under which these relational calculi (regarded as constraint systems) are faithful and adequate for a sequent calculus for the logic. We continue the study of the modal logic *K* in Section 3.2 as a running example.

First, we define what it means for a semantics of a propositional logic to be first-order definable; this is a pre-condition for producing relational calculi that express the semantics. We call the propositional logic we are studying the *object-logic*; and, we call FOL the *meta-logic*. For clarity, we use the convention prefixing *meta-* for structures at the level of the meta-logic where the terminology might otherwise overlap; e.g. *formulae* are syntactic construction at the object level, and *meta-formulae* are syntactic construction in the meta-logic.

Second, we give a sufficient condition, called *tractability*, for us to take a first-order definition Ω of a semantics and produce a relational calculus from it. Essentially, the condition amounts to unfolding Ω within $\mathsf{G3c}$ so that we can suppress all the logical structures from the meta-logic, leaving only a labelled calculus for the propositional logic—namely, the relational calculus.

Third, we give a method for transforming tractable definitions into sequent calculi and prove that the result is sound and complete for the semantics.

5.1 Tractable Propositional Logics

Relational calculi for a logic work by internalizing a semantics of that logic. In the work by Negri [56] on relational calculi for normal modal logics, the basic atomic formulae over which the relational calculi operate come in two forms: they are either of the form $(x : \phi)$, in which x is a variable denoting an arbitrary world, ϕ is a formula and : is a pairing symbol intuitively saying that ϕ is satisfied at x; or, they are of the form xRy, in which x and y are variables denoting worlds and R is a relation denoting the accessibility relation of the frame semantics. Therefore, we begin by fusing the language $\mathscr{P}$ of the object-logic with a first-order language $\mathscr{F}$, able to express frames for the semantics, into the first-order language we use for the relational calculi.

DEFINITION 5.1 (Fusion).
Let $\mathscr{F} := \langle \mathbb{R}, \varnothing, \mathbb{K}, \mathbb{V} \rangle$ be a first-order alphabet and let $\mathscr{P} := \langle \mathbb{A}, \mathbb{O}, \mathbb{C} \rangle$. The fusion $\mathscr{F} \otimes \mathscr{P}$ is the first-order alphabet $\langle \mathbb{R} \cup \{:\}, \mathbb{O} \cup \mathbb{C}, \mathbb{K} \cup \mathbb{A}, \mathbb{V} \rangle$

$$
\begin{array}{lcl}
(w : \hat{\phi} \wedge \hat{\psi}) & \Longleftrightarrow & (w : \hat{\phi}) \,\&\, (w : \hat{\psi}) \\
(w : \hat{\phi} \vee \hat{\psi}) & \Longleftrightarrow & (w : \hat{\phi}) \parr (w : \hat{\psi}) \\
(w : \neg\hat{\phi}) & \Longleftrightarrow & ((w : \hat{\phi}) \Rightarrow \bot) \\
(w : \Box\hat{\phi}) & \Longleftrightarrow & \forall u(wRu \Rightarrow (u : \hat{\phi}))
\end{array}
$$

FIGURE 6. Satisfaction for Modal Logic K (symbolic).

Observe that $\mathscr{P}$-formulae and $\mathscr{F}$-terms both becomes terms in $\mathscr{F} \oplus \mathscr{P}$, and : is a relation. In particular, the object-logic operators (i.e. connectives, modalities) are function-symbols in the fusion. Further note that $(x : \phi)$ and $(\phi : x)$ are well-formed formulae in the fusion; the former is desirable, and the latter is not. We require a *model theory* Ω over the fused language such that : is interpreted as satisfaction in the semantics. Relative to such a theory, while well-formed, the meta-formulae $(\phi : x)$ are nonsense. To aid readability, we shall use the convention of writing $\hat{\phi}$ for meta-variables that we intend to be interpreted as object-formulae and $\hat{\Gamma}$ or $\hat{\Delta}$ for meta-variables that we intend to be interpreted as object-data.

DEFINITION 5.2 (Definition of a Semantics).
Let Ω be a set of sentences from a fusion $\mathscr{F} \otimes \mathscr{P}$ and let $\mathfrak{S}$ be a semantics over $\mathscr{P}$. The set Ω defines the semantics $\mathfrak{S}$ iff the following holds: $\Omega, (x : \Gamma) \vdash (x : \Delta)$ iff $\Gamma \vDash \Delta$.

Such theories Ω may at first appear obscure, but in practice they can be fairly systematically constructed. Intuitively, the abstractions of Ω are composed of models from the semantics together with an interpretation of the satisfaction relation. Thus, Ω is typically composed of two theories Ω_1 and Ω_2. The theory Ω_1 captures frames; e.g. in modal logic, if the accessibility relation is transitive, then Ω_1 contains $\forall x, y, z(xRy \& yRz \Rightarrow xRz)$. The theory Ω_2 captures the conditions of the satisfaction relation; e.g. if the object-logic contains an additive conjunction $\wedge$, then Ω may contain $\forall x, \hat{\phi}, \hat{\psi}((x : \hat{\phi} \wedge \hat{\psi}) \Rightarrow (x : \hat{\phi}) \& (x : \hat{\psi}))$ and $\forall x, \hat{\phi}, \hat{\psi}((x : \hat{\phi}) \& (x : \hat{\psi}) \Rightarrow (x : \hat{\phi} \wedge \hat{\psi}))$. For an illustration of how Ω can be constructed according to this intuition in even relatively complex settings, see the work on the logic of Bunched Implications by Gheorghiu and Pym [30].

EXAMPLE 5.3
By the universal closure of $(\Phi \Longleftrightarrow \Psi)$, we mean the meta-formulae Θ and Θ' in which Θ is the universal closure of $\Phi \Rightarrow \Psi$ and Θ' is the universal closure of $\Psi \Rightarrow \Phi$. Consider the semantics $\mathfrak{K} = \langle \mathbb{M}, \Vdash \rangle$ in Example 3.24. It is defined by the universal closures of the formulae in Figure 6, which merits comparison with Figure 5, together with the universal closure of the following:

$$
\begin{array}{lcl}
(x : \hat{\Gamma} \,,\, \hat{\Delta}) & \Longleftrightarrow & (x : \hat{\Gamma}) \,\&\, (x : \hat{\Delta}) \\
(x : \hat{\Gamma} \,;\, \hat{\Delta}) & \Longleftrightarrow & (x : \hat{\Gamma}) \parr (x : \hat{\Delta}).
\end{array}
$$

Every model in $\mathbb{M}$ intuitively gives an abstraction of these formulas since they simply give a formal expression of the clauses defining satisfaction.

Notably, there is no meta-formula corresponding to atomic satisfaction—i.e. $(w : \mathrm{p})$—because it is handled by the structure of meta-sequents. That is, it follows from working with validity directly (i.e. without passing though truth-in-a-model): atomic satisfaction is captured by an atomic tautology, $\Omega, (x : \mathrm{p}) \triangleright (x : \mathrm{p})$.

We may use the meta-logic to characterize those propositional logics whose semantics is particularly amenable to analysis; first-order definability is, perhaps, the most general condition we may demand. What are some other properties of Ω that may be useful? Since we are interested in a *computational* analysis of the semantics, we require that it is finite, among other things. In particular, we restrict the structure of the theory to something amenable to proof-theoretic analysis according to G3c.

There is literature on generating proof systems for propositional logics defined axiomatically; see e.g. work by Ciabattoni *et al.* [12–14]. Within this tradition, Marin *et al.* [47] have used *focusing* in intuitionistic and classical logic, conceived of as a meta-logic, as a general tool to uniformly express an algorithm for turning axioms into rules applicable across different domains. We use a similar method and, therefore, polarize the syntax for the meta-logic.

Let $\mathbb{MA}$ be the set of meta-atoms. The *positive* meta-formulae P and *negative* meta-formulae N are defined as follows:

$$\begin{array}{lll} P & ::= & A \in \mathbb{MA} \mid \bot \mid P \,\&\, P \mid N \Rightarrow P \mid P \parr P \mid \exists X P \\ N & ::= & A \in \mathbb{MA} \mid N \,\&\, N \mid P \Rightarrow N \mid \forall X N. \end{array}$$

This taxonomy arises from behaviour; specifically,using this taxonomy we can define a class of formulae that we can systematically transform them into *synthetic rules* using focusing in G3c. While closely related to the taxonomy used by Marin *et al.* [47], it is not the same as they work over a syntax that has positive and negative connectives.

DEFINITION 5.4 (Polarity Alternation).
The number of *polarity alternations* in a polarized formula Φ is, which is $\pi(\Phi)$defined as follows:

$$\pi(\Phi) := \begin{cases} 0 & \text{if } \Phi \in \mathbb{MA} \\ \max\{\pi(\Phi_1), \pi(\Phi_2)\} & \text{if } \Phi = \Phi_1 \circ \Phi_2 \text{ and } \circ \in \{\&, \parr\} \\ \pi(\Phi_1) & \text{if } \Phi = QX\Psi \text{ and } Q \in \{\forall, \exists\} \\ 1 + \max\{\pi(\Phi_1), \pi(\Phi_2)\} & \text{if } \Phi = \Phi_1 \Rightarrow \Phi_2. \end{cases}$$

DEFINITION 5.5 (Tractable Meta-formula).
A meta-formula Φ is tractable iff Φ is negative and $\pi(\Phi) \leq 2$, or Φ is positive and $\pi(\Phi) \leq 1$.

The class of *geometric implications* studied by Negri [55] for the systematic generation of sequent calculus rules from axioms defining propositional logics is a subset of the tractable formulae. A meta-formula Θ is a geometric implication iff Θ is the universal closure of a meta-formula of the form $(\Phi_1 \& ... \& \Phi_m) \Rightarrow (\exists Y_1 \Psi_1 \parr ... \parr \exists Y_n \Psi_n)$ such that $\Psi_i := \Psi_1^i \& ... \& \Psi_{m_i}^i$, with the Ψ_j^i meta-atoms for $1 \leq j \leq m_i$ and $1 \leq i \leq n$, and Φ_i meta-atoms for $1 \leq i \leq m$. To see this, observe that geometric implications are of the form $N \Rightarrow P$ in which N is the conjunction of atoms and P is the disjunction of (positive) formulae of existentially quantified conjunctions (i.e. P is of the form $\exists P'$, where P' is a conjunction of atoms). Docherty and Pym [15, 16] have used this notion of meta-formulae to give

a uniform account of proof systems internalizing semantics for the family of bunched logics, with application to separation logics.

The motivation for tractability is to make a certain step in the generation of relational calculi possible, as seen in the proof of Proposition 5.9.

DEFINITION 5.6 (Tractable Theory, Semantics, Logic).
A set of meta-formulae Ω is a tractable theory iff Ω is finite and any $\Phi \in \Omega$ is a negative tractable meta-sentence. A semantics $\mathfrak{S}$ is tractable iff it is defined by a tractable theory Ω. A propositional logic is tractable iff it admits a tractable semantics $\mathfrak{S}$.

EXAMPLE 5.7 The semantics for modal logic in Example 3.24 is tractable, as witnessed by the tractable definition in Example 5.3.

It remains to give an algorithm that generates a relational calculus given a tractable definition and to prove correctness of that algorithm. Fix a semantics $\mathfrak{S} := \langle \mathbb{M}, \Vdash \rangle$ with a tractable definition Ω. Recall that $\Gamma \vDash \Delta$ obtains iff $\Omega, (x : \Gamma) \vdash (x : \Delta)$ obtains. The relational calculus we generate is a meta-sequent calculus R for the meta-logic expressive enough to capture all instances $\Omega, (x : \Gamma) \vdash (x : \Delta)$ but sufficiently restricted such that all the meta-connectives and quantifiers may be suppressed.

5.2 Generating Relational Calculi

By *generic hereditary reduction* on a meta-formula Φ, we mean the indefinite use of reduction operators from $\mathsf{G3c}$ on Φ and the generated sub-formulae, until they are meta-atoms, beginning with a meta-sequent $\Phi, \Pi \triangleright \Sigma$, with generic Π and Σ. For example, the following is a generic hereditary reduction for $(A \& B) \parr (C \& D)$ with A, B, C and D as meta-atoms:

$$\dfrac{\dfrac{A, B, \Pi \triangleright \Sigma}{(A \,\&\, B), \Pi \triangleright \Sigma} \Uparrow \&_{\mathsf{R}} \quad \dfrac{C, D, \Pi \triangleright \Sigma}{(C \,\&\, D), \Pi \triangleright \Sigma} \Uparrow \&_{\mathsf{R}}}{(A \,\&\, B) \parr (C \,\&\, D), \Pi \triangleright \Sigma} \Uparrow \parr_{\mathsf{R}}.$$

Such reductions are collapsed into *synthetic* rules, which is the rule-relation taking the putative conclusion to the premisses. The above instance collapses to the following:

$$\dfrac{A, B, \Pi \triangleright \Sigma \quad C, D, \Pi \triangleright \Sigma}{(A \,\&\, B) \parr (C \,\&\, D), \Pi \triangleright \Sigma.}$$

The quantifier rules have side-conditions in order to be applicable, and we assert these conditions in the synthetic rule. For example, when using $\forall_{\mathsf{L}}$ when doing generic hereditary reduction on $\forall X \Phi$, we require that the term T for which the variable X is substituted in Φ is already present in the meta-sequent; e.g. let $\Phi := (A(X) \& B(X)) \parr (C(X) \& D(X))$, we have the following synthetic rule for $\forall X \Phi$ with the side condition that T occurs in either Π or Σ:

$$\dfrac{A(T), B(T), \Pi \triangleright \Sigma \quad C(T), D(T), \Pi \triangleright \Sigma}{\forall X (A(X) \,\&\, B(X) \parr (C(X) \,\&\, D(X))), \Pi \triangleright \Sigma.}$$

DEFINITION 5.8 (Sequent Calculus for a Tractable Theory).
Let Ω be a tractable theory. The sequent calculus $\mathsf{G3c}(\Omega)$ is composed of $\mathsf{ax}, \bot, \mathsf{c}_L, \mathsf{c}_R$ and the synthetic rules for the meta-formulae in Ω.

The tractability condition is designed such that the following holds:

PROPOSITION 5.9
Let Ω be a tractable definition and let Π and Σ be multisets of meta-atoms,

$$\Omega, \Pi \vdash_{G3c} \Sigma \qquad \text{iff} \qquad \Omega, \Pi \vdash_{\mathsf{G3c}(\Omega)} \Sigma.$$

PROOF. Assume $\Omega, \Pi \vdash_{G3c} \Sigma$. Without loss of generality (see e.g. Liang and Miller [45] and Marin *et al.* [47]), there is a focused $\mathsf{G3c} + \mathsf{c}_R + \mathsf{c}_L$-proof $\mathcal{D}$ of $\Omega, \Pi \triangleright \Sigma$. We can assume that $\mathcal{D}$ is *focused* up to possibly using instance of c_{L} or c_{R}. That is, $\mathcal{D}$ is structured by sections of alternating phases of the following kind:

- an instance of c_{L} or c_{R}
- hereditary reduction on positive meta-formulae on the right and negative meta-formulae on the left
- eager reduction on negative meta-formulae on the right and positive meta-formulae on the left.

Since Π and Σ are composed of meta-atoms and Ω is composed of negative meta-formulae, $\mathcal{D}$ begins by a contraction and then hereditary reducing on some $\Phi \in \Omega$. Since Φ is tractable, this section in $\mathcal{D}$ may be replaced by the synthetic rule for Φ. Doing this to all the phases in $\mathcal{D}$ yields a tree of sequents $\mathcal{D}'$, which is a $\Omega, \Pi \vdash_{\mathsf{G3c}(\Omega)} \Sigma$.

Assume $\Omega, \Pi \vdash_{\mathsf{G3c}(\Omega)} \Sigma$. Since all the rules in $\mathsf{G3c}(\Omega)$ are admissible in $\mathsf{G3c}$, we immediately have $\Omega, \Pi \vdash_{\mathsf{G3c}} \Sigma$. □

EXAMPLE 5.10
Consider the tractable theory $\Omega_{\mathfrak{K}}$ in Example 5.3. The sequent calculus $\mathsf{G3c}(\Omega_{\mathfrak{K}})$ contains, among other things, the following rules corresponding to the clause for $\wedge$ in Figure 6 in which w, ϕ and ψ already occur in Ω, Π or Σ:

$$\frac{\Omega, \Pi \triangleright \Sigma, (w : \phi \wedge \psi) \quad \Omega, (w : \phi), (w : \psi), \Pi \triangleright \Sigma}{\Omega, \Pi \triangleright \Sigma}$$

$$\frac{\Omega, \Pi, (w : \phi \wedge \psi) \triangleright \Sigma \quad \Omega, \Pi \triangleright \Sigma, (w : \phi) \quad \Omega, \Pi \triangleright \Sigma, (w : \psi)}{\Omega, \Pi \triangleright \Sigma}.$$

In practice, one does not use the rules in this format. Rather, one would only apply the rules if one already knew that the left-branch would terminate; i.e. one uses the following:

$$\frac{\Omega, (w : \phi), (w : \psi), (w : \phi \wedge \psi), \Pi \triangleright \Sigma}{\Omega, (w : \phi \wedge \psi), \Pi \triangleright \Sigma}$$

$$\frac{\Omega, \Pi \triangleright \Sigma, (w : \phi \wedge \psi), (w : \phi) \quad \Omega, \Pi \triangleright \Sigma, (w : \phi \wedge \psi), (w : \psi)}{\Omega, \Pi \triangleright \Sigma, (w : \phi \wedge \psi)}.$$

This simplification can be made systematically according to the shape of the meta-formula generating the rules; it corresponds to *forward-chaining* and *back-chaining* in the proof-theoretic analysis of the meta-formula—see e.g. Marin *et al.* [47].

One desires a systematic account of the transformation of rules of arbitrary shape into rules of other (more desirable) shape. This remains to be considered in the context of relational calculi and demands further analysis on the structure of Ω. Some results of such transformations for arbitrary sequent calculi have been provided by Indrzejczak [37].

The calculus $\mathsf{G3c}(\Omega)$ is a restriction of $\mathsf{G3c}$ precisely encapsulating the proof-theoretic behaviours of the meta-formulae in Ω. It remains to suppress the logical constants of the meta-logic entirely, and thereby yield a relational calculus expressed as a labelled sequent calculus for the propositional logic.

DEFINITION 5.11 (Relational Calculus for a Tractable Theory).
Let Ω be a tractable theory. The relational calculus for Ω is the sequent calculus $\mathsf{R}(\Omega)$ that results from $\mathsf{G3c}(\Omega)$ by suppressing Ω.

THEOREM 5.12 (Soundness and Completeness).
Let $\mathfrak{S}$ be a tractable semantics and let Ω be a tractable definition for, $\mathfrak{S}$

$$\Gamma \vDash_{\mathfrak{S}} \Delta \qquad \text{iff} \qquad (x : \Gamma) \vdash_{R(\Omega)} (x : \Delta)$$

PROOF. We have the following:

$$\begin{array}{lll} \Gamma \vDash_{\mathfrak{S}} \Delta & \text{iff} \quad \Omega, (x : \Gamma) \vdash (x : \Delta) & \text{(Definition 5.6)} \\ & \text{iff} \quad \Omega, (x : \Gamma) \vdash_{G3c} (x : \Delta) & \text{(Proposition 3.4)} \\ & \text{iff} \quad \Omega, (x : \Gamma) \vdash_{G3c(\Omega)} (x : \Delta) & \text{(Proposition 5.9).} \end{array}$$

It remains to show that $\Omega, (x : \Gamma) \vdash_{G3c(\Omega)} (x : \Delta)$ iff $(x : \Gamma) \vdash_{R(\Omega)} (x : \Delta)$.

Let $\mathcal{D}$ be a $\mathsf{G3c}(\Omega)$-proof of $\Omega, (x : \Gamma) \triangleright (x : \Delta)$, and let $\mathcal{D}'$ be the result of removing Ω from every meta-sequent in $\mathcal{D}$. By Definition 5.11, we have that $\mathcal{D}'$ is a $\mathsf{R}(\Omega)$-proof of $(x : \Gamma) \triangleright (x : \Delta)$. Thus, $\Omega, (x : \Gamma) \vdash_{G3c(\Omega)} (x : \Delta)$ implies $(x : \Gamma) \vdash_{R(\Omega)} (x : \Delta)$.

Let $\mathcal{D}$ be a $\mathsf{R}(\Omega)$-proof of $(x : \Gamma) \triangleright (x : \Delta)$, and let $\mathcal{D}'$ be the result of putting Ω in every meta-sequent in $\mathcal{D}$. By Definition 5.11, we have that $\mathcal{D}'$ is a $\mathsf{G3c}(\Omega)$-proof of $\Omega, (x : \Gamma) \triangleright (x : \Delta)$. Thus, $(x : \Gamma) \vdash_{R(\Omega)} (x : \Delta)$ implies $\Omega, (x : \Gamma) \vdash_{G3c(\Omega)} (x : \Delta)$.

EXAMPLE 5.13
The sequent calculus in Example 5.10 becomes a relational calculus $\mathsf{R}(\Omega_{\mathfrak{K}})$ by suppressing Ω in the rules; e.g.

$$\frac{\Omega, \Pi \triangleright \Sigma, (w : \phi \wedge \psi) \quad \Omega, (w : \phi), (w : \psi), \Pi \triangleright \Sigma}{\Omega, \Pi \triangleright \Sigma}$$

becomes

$$\frac{\Pi \triangleright \Sigma, (w : \phi \wedge \psi) \quad (w : \phi), (w : \psi), \Pi \triangleright \Sigma}{\Pi \triangleright \Sigma}.$$

Abbreviating $\neg\Box\neg\phi$ by $\Diamond\phi$ and doing some proof-theoretic analysis on $\mathsf{R}(\Omega_{\mathfrak{K}})$, we have the simplified system RK in Figure 7—Φ denotes a meta-formula, Π and Σ denote multiset of meta-formulae, x and y denote world-variables, Δ denotes object-logic data ϕ and ψ denote object-logic formulae. This is, essentially, the relational calculus for K introduced by Negri [56].

While we have effectively transformed (tractable) semantics into relational calculi, giving a general, uniform and systematic proof theory to an ample space of logics, significant analysis remains to be done. In Example 5.13, we showed that under relatively mild conditions, one expects the relational calculus to have a particularly good shape. This begs for further characterization of the definitions of semantics and what properties one may expect the resulting relational calculus to

$$\frac{}{\Phi, \Pi \triangleright \Sigma, \Phi}\ \mathsf{ax} \qquad \frac{}{\bot, \Pi \triangleright \Sigma}\ \bot_{\mathsf{R}}$$

$$\frac{(x:\phi), (x:\psi), \Pi \triangleright \Sigma}{(x:\phi \wedge \psi), \Pi \triangleright \Sigma}\ \wedge_{\mathsf{L}} \qquad \frac{\Pi \triangleright \Sigma, (x:\phi) \quad \Pi \triangleright \Sigma, (x:\psi)}{\Pi \triangleright \Sigma, (x:\phi \wedge \psi)}\ \wedge_{\mathsf{R}}$$

$$\frac{(x:\phi), \Pi \triangleright \Sigma \quad (x:\psi), \Pi \triangleright \Sigma}{(x:\phi \vee \psi), \Pi \triangleright \Sigma}\ \vee_{\mathsf{L}} \qquad \frac{\Pi \triangleright \Sigma, (x:\phi), (x:\psi)}{\Pi \triangleright \Sigma, (x:\phi \vee \psi)}\ \vee_{\mathsf{R}}$$

$$\frac{(y:\phi), (x:\Box\phi), xRy, \Pi \triangleright \Sigma}{(x:\Box\phi), xRy, \Pi \triangleright \Sigma}\ \Box_{\mathsf{L}} \qquad \frac{xRy, \Pi \triangleright \Sigma, (y:\phi)}{\Pi \triangleright \Sigma, (x:\Box\phi)}\ \Box_{\mathsf{R}}$$

$$\frac{xRy, (y:\phi), \Pi \triangleright \Sigma}{(x:\Diamond\phi), \Pi \triangleright \Sigma}\ \Diamond_{\mathsf{L}} \qquad \frac{xRy, \Pi \triangleright \Sigma, (x:\Diamond\phi), (y:\phi)}{xRy, \Pi \triangleright \Sigma, (x:\Diamond\phi)}\ \Diamond_{\mathsf{R}}$$

$$\frac{\bot, \Pi \triangleright \Sigma}{(x:\bot), \Pi \triangleright \Sigma}\ \bot_{\mathsf{L}} \qquad \frac{\bot, \Pi \triangleright \Sigma, \bot}{\Pi \triangleright \Sigma, (x:\bot)}\ \bot_{\mathsf{R}}$$

$$\frac{(x:\Gamma), (x:\Gamma'), \Pi \triangleright \Sigma}{(x:\Gamma \,⅋\, \Gamma'), \Pi \triangleright \Sigma}\ ⅋_{\mathsf{L}} \qquad \frac{\Pi \triangleright \Sigma, (x:\Delta), (x:\Delta')}{\Pi \triangleright \Sigma, (x:\Delta \,⅋\, \Delta')}\ ⅋_{\mathsf{R}}$$

FIGURE 7. Relational Calculus RK.

have; the beginnings of such an analysis are given below Definition 5.2 in which we require Ω to contain a first-order definition of frames together with an inductive definition of the semantics.

We have presented a general account of relational calculi, but certain specific families of logics ought to be studied in particular. For example, Negri [56] demonstrated that relational calculi for modal logics are particularly simple. An adjacent class is the family of hybrid logics—see e.g. Blackburn *et al.* [5, 6], Areces and ten Cate [9], Bräuner [7] and Indrzejczak [36]. Indeed, one may regard the meta-logics for propositional logics (i.e. the fused language) as hybrid logics—see e.g. Blackburn [4].

5.3 Faithfulness and Adequacy

In this section, we give sufficient conditions for faithfulness and adequacy of a relational calculus with respect to a sequent calculus. More precisely, we give conditions under which one may transform a relational calculus into a sequent calculus for the object-logic. The result is immediate proof of soundness and completeness for the sequent calculus concerning the semantics; significantly, it bypasses term- or counter-model construction. This idea has already been implemented for the logic of Bunched Implications by Gheorghiu and Pym [30].

While the work of the preceding section generates a relational calculus, one may require some proof theory to yield a relational calculus that meets the conditions in this section faithfulness and adequacy. Likewise, one may require proof theory on the generated sequent calculus to yield a sequent calculus one recognizes as sound and complete concerning a logic of interest. We do not consider these problems here, but they are addressed explicitly for BI in the previous work by the authors.

Our objective is to systematically transform (co-)inferences in the relational calculus into (co-) inferences of the propositional logic. Regarded as constraint systems, relational calculi do not have any side-conditions on inferences; instead, all of the constraints are carried within sequents. Thus we do not need to worry about assignments and aim only to develop a valuation ν. We shall define ν by its action on sequents and extend it to reductions like in Section 4.3.

Fix a propositional logic $\vdash$ and relational calculus R. We assume the propositional logic has data-constructors $\circ$ and $\bullet$ such that

$$\Gamma \circ \Gamma' \vdash \Delta \quad \text{iff} \quad (w : \Gamma)\&(w : \Gamma') \vdash_R (w : \Delta)$$

and

$$\Gamma \vdash \Delta \bullet \Delta' \quad \text{iff} \quad (w : \Gamma) \vdash_R (w : \Delta)⅋(w : \Delta').$$

This means that the weakening, contraction and exchange structural rules are admissible for $\circ$ and $\bullet$ on the left and right, respectively. In particular, these data-constructors behave like classical conjunction and disjunction, respectively.

EXAMPLE 5.14
The logic with relational calculus RK satisfies the data-constructor condition—specifically, ⅋ is conjunctive and & is disjunctive.

A list of meta-formulae is *monomundic* iff it only contains one world-variable (but possibly many) occurrences of that world-variable; we write Π^w or Σ^w to denote monomundic lists contain the world-variable w. A monomundic list is *basic* iff it only contains meta-atoms of the form $(w : \Gamma)$, which is denoted $\bar{\Pi}^w$ or $\bar{\Sigma}^w$.

DEFINITION 5.15 (Basic Validity Sequent).
A basic validity sequent (BVS) is a pair of basic monomundic lists, $\bar{\Pi}^w \triangleright \bar{\Sigma}^w$.

DEFINITION 5.16 (Basic Rule).
A rule in a relational calculus is basic iff it is a rule over BVSs—i.e. it has the following form:

$$\frac{\bar{\Pi}_1^{w_1} \triangleright \Sigma_1^{w_1} \quad \ldots \quad \bar{\Pi}_n^{w_n} \triangleright \Sigma_n^{w_n}}{\bar{\Pi}^w \triangleright \Sigma^w}.$$

DEFINITION 5.17 (Basic Relational Calculus).
A relational calculus r is basic iff it is composed of basic rules.

Using the data-structures $\bullet$ and $\circ$, a BVS intuitively corresponds to a sequent in the propositional logic. Define $\lfloor - \rfloor_\circ$ and $\lfloor - \rfloor_\bullet$ on basic monomundic lists as follows:

$$\lfloor (w : \Gamma_1), \ldots, (w : \Gamma_m) \rfloor_\circ := \Gamma_1 \circ \ldots \circ \Gamma_m \qquad \lfloor (w : \Delta_1), \ldots, (w : \Delta_n) \rfloor_\bullet := \Delta_1 \bullet \ldots \bullet \Delta_n.$$

We can define ν on BVSs by this encoding,

$$\nu(\bar{\Pi}^w \triangleright \bar{\Sigma}^w) := \lfloor \bar{\Pi}^w \rfloor \triangleright \lfloor \bar{\Sigma}^w \rfloor$$

The significance is that whatever inference is made in the semantics using BVSs immediately yields an inference it terms of propositional sequents.

Let r be a basic rule, its propositional encoding $\nu(\mathsf{r})$ is the following:

$$\frac{\nu(\bar{\Pi}_1^{w_1} \triangleright \Sigma_1^{w_1}) \quad \ldots \quad \nu(\bar{\Pi}_n^{w_n} \triangleright \Sigma_n^{w_n})}{\nu(\bar{\Pi}^{w} \triangleright \Sigma^{w})}\,.$$

This extends to basic relational calculi pointwise,

$$\nu(\mathsf{R}) := \{\nu(\mathsf{r}) \mid \mathsf{r} \in \mathsf{R}\}$$

Despite their restrictive shape, basic rules are quite typical. For example, if the body of a clause is composed of only conjunctions and disjunctions of assertions, the rules generated by the algorithm presented above will be basic. Sets of basic rules can sometimes replace more complex rules in relational calculi to yield a basic relational calculus from a non-basic relational calculus—see Section 6 for an example.

We are thus in a situation where the rules of a reduction system intuitively correspond to the rules of a sequent calculus. The formal statement of this is below.

THEOREM 5.18
A basic relational calculus R is faithful and adequate with respect to its propositional encoding $\nu(\mathsf{R})$.

PROOF. The result follows by Definition 4.15 because a valuation of an instance of a rule in R corresponds to an instance of a rule of R on the states of the sequents involved.

Faithfulness follows by application of ν on R-proofs. That is, for any R-reduction $\mathcal{D}$, one produces a corresponding $\nu(\mathcal{R})$-proof by apply ν to each sequent in $\mathcal{D}$.

Adequacy follows by introducing arbitrary world-variables into a $\nu(\mathsf{R})$-proof. Let $\mathcal{D}$ be a $\nu(\mathsf{R})$-proof, it concludes by an inference of the following form:

$$\frac{\begin{array}{cc}\mathcal{D}_1 & \mathcal{D}_n \\ \nu(\bar{\Pi}_1^{w_1} \triangleright \Sigma_1^{w_1}) \quad \ldots & \nu(\bar{\Pi}_n^{w_n} \triangleright \Sigma_n^{w_n})\end{array}}{\nu(\bar{\Pi}^{w} \triangleright \Sigma^{w})}\,.$$

We can co-inductively define a corresponding R-with the following co-recursive step in which $\mathcal{R}_i$ is the reduction corresponding to $\mathcal{D}_i$:

$$\frac{\begin{array}{cc}\mathcal{R}_1 & \mathcal{R}_n \\ \bar{\Pi}_1^{w_1} \triangleright \Sigma_1^{w_1} \quad \ldots & \bar{\Pi}_n^{w_n} \triangleright \Sigma_n^{w_n}\end{array}}{\bar{\Pi}^{w} \triangleright \Sigma^{w}}\,.$$

Hence, for any $\nu(\mathsf{R})$-proof, there is a R-reduction $\mathcal{R}$ such that $\nu(\mathcal{R}) = \mathcal{D}$, as required. □

Of course, despite basic rules being relatively typical, many relational calculi are not comprised of *only* basic rules. Nonetheless, the phenomenon does occur for even quite complex logic. It can be used for the semantical analysis of that logic in those instances—see e.g. Gheorghiu and Pym [30] for an example of this in the case of the logic of Bunched Implications. Significantly, this approach to soundness and completeness differs from the standard term-model approach and has the advantage of bypassing truth-in-a-model (i.e. satisfaction).

6 Example: Intuitionistic Propositional Logic

In Section 5, we gave a general, uniform and systematic procedure for generating proof systems for logics that have model-theoretic semantics satisfying certain conditions. What about the reverse

problem? That is, given a proof-theoretic characterization of a propositional logic, can we *derive* a model-theoretic semantics for it (in the sense of Section 3.2)? This chapter provides an example of this for IPL.

We begin from a naive position on IPL. Our *definition* of IPL is by LJ—see Gentzen [27]. We choose this over other systems (e.g. G3i—see Troelstra and Schwichtenberg [68]) because we assume that we do not even known much about its proof theory so that we may explain through the analysis what we require. In the end, we recover the model-theoretic semantics by Kripke [42] using constraint systems as the enabling technology. Of course, for the purposes of this chapter, we shall imagine that we do not know about the semantics.

Reflecting on Section 5, we expect that the semantics we synthesize for IPL will be tractable. Therefore, we intend to build a relational calculus to bridge the proof theory and semantics of IPL as in Section 5, but this time we build it from the proof theory side. Recall that relational calculi are fragments of proof systems for FOL (i.e. the meta-logic); therefore, we begin in Section 6.1 by building a constraint system for IPL that is *classical* in shape. The system is derived in a principled way from this desire, but it is only sound and complete for IPL. We require it to be *faithful* and *adequate* for LJ because we hope to generate clause governing each connective from its rules. Hence, in Section 6.2, we analyse the constraint system to recover a faithful and adequate constraint system for IPL. In Section 6.3, we study the reductive behaviour of connectives of IPL in this constraint systems and write tractable FOL-formulae that capture the same behaviour in G3c. The resulting theory Ω determines a model-theoretic semantics for IPL, as shown in Section 6.4.

6.1 Multiple-conclusions via Boolean Constraints

We begin by defining IPL naively—i.e. from LJ (see Gentzen [27]). We do it in the style of Section 3.2 to keep it consistent with the treatment of propositional logics in the rest of this paper.

DEFINITION 6.1 (Alphabet $\mathscr{J}$).
The alphabet is $\mathscr{J} := \langle \mathbb{P}, \{\wedge, \vee, \to, \neg\}, \{,, ;, \varnothing\}\rangle$, in which symbols $\wedge, \vee, \to, ,, ;$ have arity 2, the symbol $\neg$ has arity 1 and $\varnothing$ has arity 0.

Let $\equiv$ be the smallest relation satisfying commutative monoid equations for , and ; with unit $\varnothing$.

DEFINITION 6.2 (System LJ).
Sequent calculus LJ is comprised of the rules in Figure 8, in which Δ is either a $\mathscr{J}$-formula ϕ or $\varnothing$, and $\Gamma \equiv \Gamma'$ and $\Delta \equiv \Delta'$ in e.

In this section, LJ-provability $\vdash_{LJ}$ *defines* the judgment relation for IPL. Our task is to derive a model-theoretic characterization of IPL. As we saw in Section 5, the logic in which semantics is defined is classical. Therefore, our strategy is to use the constraint to present IPL in a sequent calculus with a classical shape; i.e. the constraint informs precisely where the semantics of IPL diverge from those of FOL. This tells us how the semantic clauses of FOL need to be augmented to define the connectives of IPL. The calculus in question is Gentzen's LK [27].

The essential point of distinction between LJ and LK is in c_{R}, $\to_{\mathsf{L}}$ and $\neg_{\mathsf{L}}$ as it is these rules that enable multiple-conclusioned sequents to appear in the latter but not the former. To bring these behaviours closer, we introduce a constraints constraint system for IPL whose combinatorial behaviour is like LK, but for which we have constraints to recover LJ.

The algebra of the constraint system is Boolean algebra—see Section 2.

$$\frac{\Gamma \rhd \Delta}{\phi \,,\, \Gamma \rhd \Delta}\ \mathsf{w_L} \qquad \frac{\Gamma \rhd \varnothing}{\Gamma \rhd \phi}\ \mathsf{w_R} \qquad \frac{\phi \,,\, \phi \,,\, \Gamma \rhd \Delta}{\phi \,,\, \Gamma \rhd \Delta}\ \mathsf{c_L} \qquad \frac{\Gamma' \rhd \Delta'}{\Gamma \rhd \Delta}\ \mathsf{e}$$

$$\frac{\Gamma \rhd \phi \quad \Gamma \rhd \psi}{\Gamma \rhd \phi \wedge \psi}\ \wedge_{\mathsf{R}} \qquad \frac{\phi \,,\, \Gamma \rhd \Delta}{\phi \wedge \psi \,,\, \Gamma \rhd \Delta}\ \wedge^1_{\mathsf{L}} \qquad \frac{\psi \,,\, \Gamma \rhd \Delta}{\phi \wedge \psi \,,\, \Gamma \rhd \Delta}\ \wedge^2_{\mathsf{L}} \qquad \frac{\phi \,,\, \Gamma \rhd \varnothing}{\Gamma \rhd \neg\phi}\ \neg_{\mathsf{R}}$$

$$\frac{\phi \,,\, \Gamma \rhd \Delta \quad \psi \,,\, \Gamma \rhd \Delta}{\phi \vee \psi \,,\, \Gamma \rhd \Delta}\ \vee_{\mathsf{L}} \qquad \frac{\Gamma \rhd \phi}{\Gamma \rhd \phi \vee \psi}\ \vee^1_{\mathsf{R}} \qquad \frac{\Gamma \rhd \psi}{\Gamma \rhd \phi \vee \psi}\ \vee^2_{\mathsf{R}} \qquad \frac{\Gamma \rhd \phi}{\neg\phi \,,\, \Gamma \rhd \varnothing}\ \neg_{\mathsf{L}}$$

$$\frac{\phi \,,\, \Gamma \rhd \psi}{\Gamma \rhd \phi \to \psi}\ \to_{\mathsf{R}} \qquad \frac{\Gamma_1 \rhd \phi \quad \psi \,,\, \Gamma_2 \rhd \Delta}{\phi \to \psi \,,\, \Gamma_1 \,,\, \Gamma_2 \rhd \Delta}\ \to_{\mathsf{L}} \qquad \frac{}{\phi \rhd \phi}\ \mathsf{ax}$$

FIGURE 8. Sequent Calculus LJ.

EXAMPLE 6.3
The following is an enriched $\mathscr{J}$-sequent.

$$(\Gamma \cdot 1) \,,\, (\phi \cdot x) \rhd (\Delta \cdot \bar{x}) \mathbin{⨟} (\psi \cdot x).$$

For the rules that are the same across both systems, the expressions are inherited from the justifying sub-formulae of the conclusion; e.g. in the case of conjunction, one has the following:

$$\frac{\Gamma \rhd \phi \quad \Gamma \rhd \psi}{\Gamma \rhd \phi \wedge \psi}\ \wedge_{\mathsf{R}} \quad \text{becomes} \quad \frac{\Gamma \rhd \Delta \mathbin{⨟} (\phi \cdot x) \quad \Gamma \rhd \Delta \mathbin{⨟} (\psi \cdot x)}{\Gamma \rhd \Delta \mathbin{⨟} (\phi \wedge \psi \cdot x)}\ \wedge^{\mathcal{B}}_{\mathsf{R}}.$$

For readability, we may suppress the Boolean expressions on these rules.

DEFINITION 6.4 (System $\mathsf{LK} \oplus \mathcal{B}$).
Sequent calculus $\mathsf{LK} \oplus \mathcal{B}$ is comprised of the rules in Figure 9, in which Γ and Δ are enriched $\mathscr{J}$-datum, and $\Gamma \equiv \Gamma'$ and $\Delta \equiv \Delta'$ in $\mathsf{e}^{\mathcal{B}}$.

The ergo rendering $\mathsf{LK} \oplus \mathcal{B}$ sound and complete for IPL is precisely the demand for one to choose which of the formulae in the succedent of a sequent to assert as a consequence of the context. This reading is closely related to the semantics of intuitionistic proof-search provided by Pym and Ritter [62].

DEFINITION 6.5 (Choice ergo).
Let $I : X \to \mathcal{B}$ be an interpretation of the language of the Boolean algebra. The choice ergo is the function σ_I, which acts on $\mathscr{J}$-formulae as follows:

$$\sigma_I(\phi) \mapsto \begin{cases} \phi & \text{if } \phi \text{ unlabelled} \\ \sigma_I(\psi) & \text{if } I(x) = 1 \text{ and } \phi = \psi \cdot x \\ \varnothing & \text{if } I(x) = 0 \text{ and } \phi = \psi \cdot x. \end{cases}$$

$$\frac{\Gamma \rhd \Delta}{\phi \,,\, \Gamma \rhd \Delta}\ \mathsf{w}^{\mathcal{B}}_{\mathsf{L}} \qquad \frac{\Gamma \rhd \Delta}{\Gamma \rhd \Delta \,,\, \phi}\ \mathsf{w}^{\mathcal{B}}_{\mathsf{R}} \qquad \frac{\phi \,,\, \phi \,,\, \Gamma \rhd \Delta}{\phi \,,\, \Gamma \rhd \Delta}\ \mathsf{c}^{\mathcal{B}}_{\mathsf{L}} \qquad \frac{\Gamma \rhd \Delta \fatsemi \phi \cdot x \fatsemi \phi \cdot \bar{x}}{\Gamma \rhd \Delta \fatsemi \phi}\ \mathsf{c}^{\mathcal{B}}_{\mathsf{R}}$$

$$\frac{\Gamma' \rhd \Delta'}{\Gamma \rhd \Delta}\ \mathsf{e}^{\mathcal{B}} \qquad \frac{x = y = 1}{\phi \cdot x \rhd \phi \cdot y}\ \mathsf{ax}^{\mathcal{B}} \qquad \frac{\Gamma \rhd \Delta \cdot \bar{x} \,,\, \phi \cdot x}{\neg\phi \,,\, \Gamma \rhd \Delta}\ \neg^{\mathcal{B}}_{\mathsf{L}} \qquad \frac{\phi \,,\, \Gamma \rhd \Delta}{\Gamma \rhd \Delta \fatsemi \neg\phi}\ \neg^{\mathcal{B}}_{\mathsf{R}}$$

$$\frac{\Gamma \rhd \phi \fatsemi \Delta \quad \Gamma \rhd \psi \fatsemi \Delta}{\Gamma \rhd \Delta \fatsemi \phi \wedge \psi}\ \wedge^{\mathcal{B}}_{\mathsf{R}} \qquad \frac{\phi \,,\, \Gamma \rhd \Delta}{\phi \wedge \psi \,,\, \Gamma \rhd \Delta}\ \wedge^{\mathcal{B}}_{\mathsf{L}_1} \qquad \frac{\psi \,,\, \Gamma \rhd \Delta}{\phi \wedge \psi \,,\, \Gamma \rhd \Delta}\ \wedge^{\mathcal{B}}_{\mathsf{L}_2}$$

$$\frac{\phi \,,\, \Gamma \rhd \Delta \quad \psi \,,\, \Gamma \rhd \Delta}{\phi \vee \psi \,,\, \Gamma \rhd \Delta}\ \vee^{\mathcal{B}}_{\mathsf{L}} \qquad \frac{\Gamma \rhd \Delta \fatsemi \phi}{\Gamma \rhd \Delta \fatsemi \phi \vee \psi}\ \vee^{\mathcal{B}}_{\mathsf{R}_1} \qquad \frac{\Gamma \rhd \Delta \fatsemi \psi}{\Gamma \rhd \Delta \fatsemi \phi \vee \psi}\ \vee^{\mathcal{B}}_{\mathsf{R}_2}$$

$$\frac{\phi \,,\, \Gamma \rhd \Delta \fatsemi \psi}{\Gamma \rhd \Delta \fatsemi \phi \to \psi}\ \to^{\mathcal{B}}_{\mathsf{R}} \qquad \frac{\Gamma_1 \rhd \Delta_1 \cdot \bar{x} \,,\, \phi \cdot x \quad \psi \,,\, \Gamma_2 \rhd \Delta_2}{\phi \to \psi \,,\, \Gamma_1 \,,\, \Gamma_2 \rhd \Delta_1 \fatsemi \Delta_2}\ \to^{\mathcal{B}}_{\mathsf{L}}$$

FIGURE 9. Constraint System $\mathsf{LK} \oplus \mathcal{B}$.

The choice ergo acts on enriched $\mathscr{J}$-data by acting point-wise on the formulae; and it acts on enriched sequents by acting on each component independently—i.e. $\sigma_I(\Gamma \rhd \Delta) = \sigma_I(\Gamma) \rhd \sigma_I(\Delta)$.

EXAMPLE 6.6
Let S be the sequent in Example 6.3. If $I(x) = 1$, then $\sigma_I(S) = \Gamma \,,\, \phi \rhd \psi$.

PROPOSITION 6.7
System $\mathsf{LK} \oplus \mathcal{B}$, with the choice ergo σ, is sound and complete for IPL,

$$\Gamma \vdash^{\sigma}_{\mathsf{LK}\oplus\mathcal{B}} \Delta \qquad \text{iff} \qquad \Gamma \vdash_{LJ} \Delta.$$

PROOF OF SOUNDNESS. Suppose $\Gamma \vdash^{\sigma}_{\mathsf{LK}\oplus\mathcal{B}} \Delta$, then there is a coherent $\mathsf{LK} \oplus \mathcal{B}$-reduction $\mathcal{R}$ of an enriched sequent S such that $\sigma_I(S) := \Gamma \rhd \Delta$, where I is any assignment satisfying $\mathcal{R}$. It follows that S is equivalent (up to exchange) to the sequent $\Gamma' \,,\, \Pi \rhd \Sigma \fatsemi \Delta'$ in which Γ' and Δ' are like Γ and Δ but with labelled data and I applied to the expressions in Π and Σ evaluates to 0 but applied to expressions in Γ' and Δ' evaluates to 1. We proceed by induction n on the height of $\mathcal{R}$—i.e. the maximal number of reductive inferences in a branch of the tree.

BASE CASE. If $n = 1$, then $\Gamma', \Pi \rhd \Sigma, \Delta'$ is an instances of $\mathsf{ax}^{\mathcal{B}}$. But then $S = \phi \cdot x \rhd \phi \cdot y$, for some formula ϕ. We have $\phi \vdash_{LJ} \phi$ by ax.

INDUCTIVE STEP. The induction hypothesis (IH) is as follows: if $\Gamma' \vdash^{\sigma}_{\mathsf{LK}\oplus\mathcal{B}} \Delta'$ is witnessed by $\mathsf{LK} \oplus \mathcal{B}$-reductions of $k \leq n$, then $\Gamma \vdash_{LJ} \Delta$.

Suppose that the shortest reduction witnessing $\Gamma' \vdash^{\sigma}_{\mathsf{LK}\oplus\mathcal{B}} \Delta'$ is of height $n + 1$. Let $\mathcal{R}$ be such a reduction. Without loss of generality, we assume the root of $\mathcal{R}$ is of the form $\Gamma' \,,\, \Pi \rhd \Sigma \fatsemi \Delta'$, as above. It follows by case analysis on the final inferences of $\mathcal{R}$ (i.e. reductive inferences applied to the root) that $\Gamma \vdash_{LJ} \Delta$. We show two cases, the rest being similar.

- Suppose the last inference of $\mathcal{R}$ was by $c^{\mathcal{B}}_{\mathsf{R}}$. In this case, $\mathcal{R}$ has an immediate sub-tree $\mathcal{R}'$ that is a coherent $\mathsf{LK} \oplus \mathcal{B}$-reduction of either $\Gamma' \,,\, \Sigma \rhd \Sigma' \fatsemi \Delta'$ or $\Gamma' \,,\, \Sigma \rhd \Sigma' \fatsemi \Delta''$, in which Σ' and Δ'' are like Σ and Δ', respectively, but with some formula repeated such that one occurrence

carries an additional expression x and the other occurrence with an $\bar{x}$. The coherent assignment of $\mathcal{R}$ are the same as those $\mathcal{R}'$ since the two reductions have the same constraints. We observe that under these coherent assignment $\mathcal{R}'$ witnesses $\Gamma' \vdash^{\sigma}_{\mathsf{LK}\oplus\mathcal{B}} \Delta'$. By the IH, since $\mathcal{R}'$ is of height n, it follows that $\Gamma \vdash_{LJ} \Delta$.

- Suppose the last inference of $\mathcal{R}$ was by $\rightarrow^{\mathcal{B}}_{\mathsf{R}}$. In this case, $\mathcal{R}$ has an immediate sub-tree $\mathcal{R}'$. If the principal formula of the inference is not in Δ', then $\mathcal{R}'$ witnesses $\Gamma' \vdash^{\sigma}_{\mathsf{LK}\oplus\mathcal{B}} \Delta'$. Hence, by the IH, we conclude $\Gamma \vdash_{LJ} \Delta$. If the principal formula of the inference is in Δ', then $\mathcal{R}'$ is a proof of $\phi \, , \Gamma' \, , \Pi \rhd \Sigma \fatsemi \Delta''; \psi$, where $\Delta' := \Delta'' \fatsemi \phi \rightarrow \psi$. It follows, by the IH, that $\phi, \Gamma' \vdash_{LJ} \Delta'' \fatsemi \psi$. By the $\rightarrow_{\mathsf{R}}$-rule in LJ, we have $\Gamma \vdash_{LJ} \phi \rightarrow \psi$—i.e. $\Gamma \vdash_{LJ} \Delta$, as required.

Of course, since we are working with LJ, we know that Δ contains only one formula. This distinction was not important for the proof, so we have left with the more general notation. □

PROOF OF COMPLETENESS. This follows immediately from the fact that all the rules of LJ may be simulated in $\mathsf{LK} \oplus \mathcal{B}$. □

The point of this work is that $\mathsf{LK}\oplus\mathcal{B}$ characterizes IPL in a way that is combinatorially comparable to FOL. This is significant as the semantics of IPL is given classically, hence $\mathsf{LK} \oplus \mathcal{B}$ bridges the proof-theoretic and model-theoretic characterizations of IPL.

6.2 Faithfulness and Adequacy

Though we may use $\mathsf{LK} \oplus \mathcal{B}$ to reason about IPL with classical combinatorics, the system does not immediately reveal the meaning of the connectives of IPL in terms of their counterparts in FOL. The problem is that $\mathsf{LK} \oplus \mathcal{B}$-proofs are only *globally* valid for IPL, with respect to the choice ergo σ. Therefore, to conduct a semantical analysis of IPL in terms of FOL, we require a constraint system based on FOL whose proofs are *locally* valid—i.e. a system that is not only sound and complete for IPL, but *faithful* and *adequate*. In this section, we analyse the relationship between $\mathsf{LK} \oplus \mathcal{B}$ and LJ to produce such a system.

A significant difference between LK and LJ is the use of richer data-structures for the succedent in the former than in the latter (i.e. list or multisets verses formulae). Intuitively, the data-constructor in the succedent acts as a meta-level disjunction, thus we may investigate how $\mathsf{LK} \oplus \mathcal{B}$ captures IPL by considering how $\mathsf{c}^{\mathcal{B}}_{R}$ interacts with $\vee^{\mathcal{B}}_{\mathsf{R}}$. We may restrict attention to interactions of the following form:

$$\dfrac{\dfrac{\dfrac{\Gamma \rhd \Delta \fatsemi (\phi \cdot x) \fatsemi (\psi \cdot \bar{x})}{\Gamma \rhd \Delta \fatsemi (\phi \cdot x) \fatsemi (\phi \vee \psi \cdot \bar{x})} \vee^{\mathcal{B}_2}_{\mathsf{R}}}{\Gamma \rhd \Delta \fatsemi (\phi \vee \psi \cdot x) \fatsemi (\phi \vee \psi \cdot \bar{x})} \vee^{\mathcal{B}_1}_{\mathsf{R}}}{\Gamma \rhd \Delta \fatsemi \phi \vee \psi} \mathsf{c}^{\mathcal{B}}_{\mathsf{R}} \, .$$

These may be collapsed into single inference rules,

$$\dfrac{\Gamma \rhd \phi \cdot xy \fatsemi \psi \cdot x\bar{y} \fatsemi \Delta}{\Gamma \rhd \phi \vee \psi \cdot x \, , \Delta} \, .$$

The other connectives either make use of constraints, and therefore have no significant interaction with disjunction, or simply can be permuted without loss of generality—e.g. a typical interaction

$$\frac{}{\phi \,,\, \Gamma \rhd \Delta \fatsemi \phi}\ \mathsf{ax} \qquad \frac{\Gamma \rhd \Delta}{\phi \,,\, \Gamma \rhd \Delta}\ \mathsf{w}^{\mathcal{B}}_{\mathsf{L}} \qquad \frac{\Gamma \rhd \Delta}{\Gamma \rhd \Delta \fatsemi \phi}\ \mathsf{w}^{\mathcal{B}}_{\mathsf{R}} \qquad \frac{\phi \,,\, \phi \,,\, \Gamma \rhd \Delta}{\phi \,,\, \Gamma \rhd \Delta}\ \mathsf{c}^{\mathcal{B}}_{\mathsf{L}} \qquad \frac{\Gamma' \rhd \Delta'}{\Gamma \rhd \Delta}\ \mathsf{e}^{\mathcal{B}}$$

$$\frac{\Gamma \rhd \Delta \fatsemi \phi}{\neg\phi \fatsemi \Gamma \rhd \Delta}\ \neg^{\mathcal{B}}_{\mathsf{L}} \qquad \frac{\phi \cdot xy \,,\, \Gamma \rhd \Delta \quad xy = 1}{\Gamma \rhd \Delta \fatsemi \neg\phi \cdot x}\ \neg^{\mathcal{B}}_{\mathsf{R}}$$

$$\frac{\phi \,,\, \psi \,,\, \Gamma \vdash \Delta}{\phi \wedge \psi \,,\, \Gamma \rhd \Delta}\ \wedge^{\mathcal{B}}_{\mathsf{L}} \qquad \frac{\Gamma \rhd \Delta \fatsemi \phi \quad \Gamma \rhd \Delta \fatsemi \psi}{\Gamma \rhd \Delta \fatsemi \phi \wedge \psi}\ \wedge^{\mathcal{B}}_{\mathsf{R}}$$

$$\frac{\phi \,,\, \Gamma \rhd \Delta \quad \psi \,,\, \Gamma \rhd \Delta}{\phi \vee \psi \,,\, \Gamma \rhd \Delta}\ \vee^{\mathcal{B}}_{\mathsf{L}} \qquad \frac{\Gamma \rhd \phi \cdot xy \fatsemi \psi \cdot x\bar{y} \fatsemi \Delta}{\Gamma \rhd \phi \vee \psi \cdot x \fatsemi \Delta}\ \vee^{\mathcal{B}}_{\mathsf{R}}$$

$$\frac{\Gamma \rhd \Delta \fatsemi \phi \quad \psi \,,\, \Gamma \rhd \Delta}{\phi \to \psi \,,\, \Gamma \rhd \Delta}\ \to^{\mathcal{B}}_{\mathsf{L}} \qquad \frac{\Gamma \,,\, \phi \cdot xy \rhd \psi \cdot xy \fatsemi \Delta \quad xy = 1}{\Gamma \rhd \phi \to \psi \cdot x \fatsemi \Delta}\ \to^{\mathcal{B}}_{\mathsf{R}}$$

FIGURE 10. Constraint System $\mathsf{LK}^{+} \oplus \mathcal{B}$.

between $\mathsf{c}^{\mathcal{B}}_{R}$ and such as $\wedge^{\mathcal{B}}_{\mathsf{R}}$

$$\dfrac{\dfrac{\dfrac{\Gamma \rhd \Delta_1 \fatsemi \phi \fatsemi \phi \quad \Gamma \rhd \Delta_2 \fatsemi \phi \fatsemi \psi}{\Gamma \rhd \Delta_1 \fatsemi \Delta_2 \fatsemi \phi \fatsemi \phi \wedge \psi}\ \wedge^{\mathcal{B}}_{\mathsf{R}}}{\Gamma \rhd \Delta_1 \fatsemi \Delta_2 \fatsemi \phi \wedge \psi \fatsemi \phi}\ \mathsf{e}^{\mathcal{B}}_{\mathsf{R}} \qquad \Gamma \rhd \Delta_3 \fatsemi \psi}{\Gamma \rhd \Delta_1 \fatsemi \Delta_2 \fatsemi \Delta_3 \fatsemi \phi \wedge \psi \fatsemi \phi \wedge \psi}\ \wedge^{\mathcal{B}}_{\mathsf{R}}}{\Gamma \rhd \Delta_1 \fatsemi \Delta_2 \fatsemi \Delta_3 \fatsemi \phi \wedge \psi}\ \mathsf{c}^{\mathcal{B}}_{\mathsf{R}}$$

may be replaced by the derivation,

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\Gamma \rhd \Delta_1 \fatsemi \phi \fatsemi \phi}{\Gamma \rhd \Delta_1 \fatsemi \phi \fatsemi \phi \fatsemi \Delta_2}\ \mathsf{w}^{\mathcal{B}}_{\mathsf{R}}}{\Gamma \rhd \Delta_1 \fatsemi \phi \fatsemi \Delta_2 \fatsemi \phi}\ \mathsf{e}^{\mathcal{B}}_{\mathsf{R}}}{\Gamma \rhd \Delta_1 \fatsemi \Delta_2 \fatsemi \phi \fatsemi \phi}\ \mathsf{e}^{\mathcal{B}}_{\mathsf{R}}}{\Gamma \rhd \Delta_1 \fatsemi \Delta_2 \fatsemi \phi}\ \mathsf{c}^{\mathcal{B}}_{\mathsf{R}} \qquad \Gamma \rhd \Delta_3 \fatsemi \psi}{\Gamma \rhd \Delta_1 \fatsemi \Delta_2 \fatsemi \Delta_3 \fatsemi \phi \wedge \psi}\ \wedge^{\mathcal{B}}_{\mathsf{R}}.$$

This analysis allows us to eliminate $\mathsf{c}^{\mathcal{B}}_{R}$ as it is captured wherever it is needed by the augmented rule for disjunction; similarly, we may eliminate $\mathsf{c}^{\mathcal{B}}_{L}$ by incorporating it in the other rules. In total, this yields a new constraint system, $\mathsf{LK}^{+} \oplus \mathcal{B}$.

DEFINITION 6.8 (System $\mathsf{LK}^{+} \oplus \mathcal{B}$).
System $\mathsf{LK}^{+} \oplus \mathcal{B}$ is given in Figure 10, in which Γ and Δ are enriched $\mathscr{J}$-datum, and $\Gamma \equiv \Gamma$ and $\Delta \equiv \Delta'$ in e.

System $\mathsf{LK}^{+} \oplus \mathcal{B}$ characterizes IPL locally—i.e. it is faithful and adequate with respect to *some* sequent calculus for IPL. That sequent calculus, however, is not LJ, but rather a multiple-

$$\frac{}{\phi \,,\, \Gamma \triangleright \Delta \fatsemi \phi}\ \mathsf{ax} \qquad \frac{\Gamma \triangleright \Delta}{\phi \,,\, \Gamma \triangleright \Delta}\ \mathsf{w_L} \qquad \frac{\Gamma \triangleright \Delta}{\Gamma \triangleright \Delta \fatsemi \phi}\ \mathsf{w_R} \qquad \frac{\phi \,,\, \phi \,,\, \Gamma \triangleright \Delta}{\phi \,,\, \Gamma \triangleright \Delta}\ \mathsf{c_L} \qquad \frac{\Gamma' \triangleright \Delta'}{\Gamma \triangleright \Delta}\ \mathsf{e}$$

$$\frac{\Gamma \triangleright \Delta \fatsemi \phi}{\neg\phi \,,\, \Gamma \triangleright \Delta}\ \neg_{\mathsf{L}} \qquad \frac{\phi \,,\, \Gamma \triangleright \varnothing}{\Gamma \triangleright \Delta \fatsemi \neg\phi}\ \neg_{\mathsf{R}}$$

$$\frac{\phi \,,\, \psi \,,\, \Gamma \triangleright \Delta}{\phi \wedge \psi \,,\, \Gamma \triangleright \Delta}\ \wedge_{\mathsf{L}} \qquad \frac{\Gamma \triangleright \Delta \fatsemi \phi \quad \Gamma \triangleright \Delta \fatsemi \psi}{\Gamma \triangleright \Delta \fatsemi \phi \wedge \psi}\ \wedge_{\mathsf{R}}$$

$$\frac{\phi \,,\, \Gamma \triangleright \Delta \quad \psi \,,\, \Gamma \triangleright \Delta}{\phi \vee \psi \,,\, \Gamma \triangleright \Delta}\ \vee_{\mathsf{L}} \qquad \frac{\Gamma \triangleright \phi \fatsemi \psi \fatsemi \Delta}{\Gamma \triangleright \phi \vee \psi \fatsemi \Delta}\ \vee_{\mathsf{R}}$$

$$\frac{\Gamma \triangleright \Delta \fatsemi \phi \quad \psi \,,\, \Gamma \triangleright \Delta}{\phi \to \psi \,,\, \Gamma \triangleright \Delta}\ \to_{\mathsf{L}} \qquad \frac{\Gamma \fatsemi \phi \triangleright \psi}{\Gamma \triangleright \phi \to \psi \fatsemi \Delta}\ \to_{\mathsf{R}}$$

FIGURE 11. Sequent Calculus LJ^+.

conclusioned system LJ^+. Essentially, LJ^+ is the multiple-conclusioned sequent calculus introduced by Dummett [18] with certain instances of the structural rules incorporated into the operational rules.

DEFINITION 6.9 (Sequent calculus LJ^+).
Sequent calculus LJ^+ is given by the rules in Figure 11, in which Γ and Δ are $\mathscr{J}$-datum, and $\Gamma \equiv \Gamma'$ and $\Delta \equiv \Delta'$ in e.

PROPOSITION 6.10
Sequent calculus LJ^+ is sound and complete for IPL,

$$\Gamma \vdash_{LJ} \phi \qquad \text{iff} \qquad \Gamma \vdash_{LJ^+} \phi.$$

PROOF. Follows from Dummett [18]. □

The choice ergo σ extends to a valuation from $\mathsf{LK}^+ \oplus \mathcal{B}$ to LJ^+ by pointwise application to every sequent within the reduction.

PROPOSITION 6.11
System $\mathsf{LK}^+ \oplus \mathcal{B}$, with valuation σ, is faithful and adequate with respect to LJ^+.

PROOF. Faithfulness follows from the observation that each rule in $\mathsf{LK}^+ \oplus \mathcal{B}$ produces the corresponding rule in LJ^+ when its constraints are observed. Adequacy follows from the observation that every instance of every rule in LJ^+ is an evaluation of an instance of a rules of $\mathsf{LK}^+ \oplus \mathcal{B}$ respecting its constraints. □

The force of this result is that we may use $\mathsf{LK}^+ \oplus \mathcal{B}$ to study intuitionistic connectives in terms of their classical counterparts. This analysis allows us to synthesize a model-theoretic for IPL from the semantics of FOL.

6.3 Semantical Analysis of IPL

Our approach to deriving a semantics for IPL is to construct a set of meta-formulae Ω that forms a tractable definition of a semantics for IPL. The idea is that we use $\mathsf{LK}^{+} \oplus \mathcal{B}$ to determine meta-formulae for each connective that simulate the proof theory of IPL within LK^{+}.

Recall that in Section 5.3 we require a data-constructor to represent classical conjunction ($\&$) and one for classical disjunction ($\parr$). These are $,$ and $;$, respectively. We may now proceed to analyse the connectives of IPL.

We observe in $\mathsf{LK}^{+} \oplus \mathcal{B}$ that intuitionistic conjunction has the same inferential behaviour as classical conjunction,

$$\frac{\Gamma \vartriangleright \Delta ; \phi \quad \Gamma \vartriangleright \Delta ; \psi}{\Gamma \vartriangleright \Delta ; \phi \wedge \psi}\ \wedge^{\mathcal{B}}_{\mathsf{R}} \qquad \text{vs.} \qquad \frac{\Pi \vartriangleright \Sigma, \Phi \quad \Pi \vartriangleright \Sigma, \Psi}{\Pi \vartriangleright \Sigma, \Phi \,\&\, \Psi}\ \&_{\mathsf{R}}.$$

Therefore, it seems that $\wedge$ in IPL should be defined as $\&$ in FOL. A candidate meta-formula governing the connective is the universal closure of the following (we use the convention in Section 5.1 in which $\hat{\phi}$ and $\hat{\psi}$ are used as meta-variables for formulae of the object-logic):

$$(w : \hat{\phi} \wedge \hat{\psi}) \iff (w : \hat{\phi}) \& (w : \hat{\psi}).$$

This is the appropriate clause for the connective as it enables the following behaviour in the meta-logic in which the double-line suppresses the use of the clause:

$$\dfrac{\dfrac{\Omega, \Pi, (w : \Gamma) \vartriangleright (w : \phi), \Sigma \quad \Omega, \Pi, (w : \Gamma) \vartriangleright (w : \psi), \Sigma}{\Omega, \Pi, (w : \Gamma) \vartriangleright (w : \phi) \,\&\, (w : \psi), \Sigma}\ \&_{\mathsf{R}}}{\Omega, \Pi, (w : \Gamma) \vartriangleright (w : \phi \wedge \psi), \Sigma}.$$

Recall, such derivations correspond to the *use* of the clause—see Section 5.2—which may be collapsed into rules themselves,

$$\frac{\Omega, \Pi, (w : \Gamma) \vartriangleright (w : \phi), \Sigma \quad \Omega, \Pi, (w : \Gamma) \vartriangleright (w : \psi), \Sigma}{\Omega, \Pi, (w : \Gamma) \vartriangleright (w : \phi \wedge \psi), \Sigma}\ \wedge\text{-clause}.$$

Intuitively, this rule precisely recovers $\wedge_{\mathsf{R}} \in \mathsf{LJ}^{+}$,

$$\frac{\Gamma \vartriangleright \Delta ; \phi \quad \Gamma \vartriangleright \Delta ; \psi}{\Gamma \vartriangleright \Delta ; \phi \wedge \psi}\ \wedge_{\mathsf{R}}.$$

Of course, it is important to check that the clause also has the correct behaviour in the left-hand side of sequents; we discuss this at the end of the present section.

We obtain the universal closure of the following for the clauses governing disjunction ($\vee$) and ($\bot$) analogously:

$$(x : \hat{\phi} \vee \hat{\psi}) \iff ((x : \hat{\phi}) \parr (x : \hat{\psi})) \qquad (x : \bot) \iff \bot.$$

It remains to analyse implication ($\rightarrow$). The above reasoning does not follow *mutatis mutandis* because the constraints in $\mathsf{LK}^{+} \oplus \mathcal{B}$ becomes germane, so we require something additional to get the appropriate simulation.

How may we express $\to$ in terms of the classical connectives? We begin by considering $\to^{\mathcal{B}}_{\mathsf{R}} \in \mathsf{LK}^{+} \oplus \mathcal{B}$,

$$\frac{\Gamma \,⨾\, (\phi \cdot xy) \triangleright (\psi \cdot xy) \,⨾\, \Delta \quad xy = 1}{\Gamma \triangleright (\phi \to \psi \cdot x) \,⨾\, \Delta} \to^{\mathcal{B}}_{\mathsf{R}}.$$

Since $\mathsf{LK}^{+} \oplus \mathcal{B}$ is not only sound and complete for IPL but faithful and adequate, we know that this rule characterizes the connective. The rule admits two assignment classes: $x \mapsto 0$ or $x \mapsto 1$. Super-imposing these valuations can capture the behaviour we desire in the meta-logic on each other by using *possible worlds* to distinguish the possible cases,

$$\frac{\Omega, \Pi[w \mapsto u], \Pi[w \mapsto v], (u : \phi) \triangleright (u : \psi), \Sigma[w \mapsto v], \Sigma[w \mapsto u]}{\Omega, \Pi \triangleright (w : \phi \to \psi), \Sigma}.$$

We assume that since u and v are distinct, they do not interact so that the rule captures the following possibilities:

$$\frac{\Omega, \Pi[w \mapsto u], (u : \phi) \triangleright (u : \psi), \Sigma[w \mapsto u]}{\Omega, \Pi \triangleright (w : \phi \to \psi), \Sigma} \qquad \frac{\Omega, \Pi[w \mapsto v] \triangleright \Sigma[w \mapsto v]}{\Omega, \Pi \triangleright (w : \phi \to \psi), \Sigma}.$$

The assumption is proved valid below—see Proposition 6.18. Applying the state function to these rules does indeed recover the possible cases of $\to^{\mathcal{B}}_{\mathsf{R}}$, which justifies that this super-imposing behaviour is what we desire of the clause governing implication. It remains only to find that clause.

One of these possibilities amounts to a weakening, a behaviour already present through interpreting the data-structures as classical conjunction and disjunction. The other possibility we recognize as having the combinatorial behaviour of classical implication concerns creating a meta-formula in the antecedent of the premiss by taking part in a meta-formula in the succedent of the conclusion. Naively, we may consider the following as the clause:

$$(w : \hat{\phi} \to \hat{\psi}) \iff \big((w : \hat{\phi}) \Rightarrow (w : \hat{\psi})\big)$$

However, this fails to account for the change in world. Thus, we require the clause to have a universal quantifier over worlds and a precondition that enables the $\Pi[w \mapsto u]$ substitution. Analysing the possible use cases, we observe that R must satisfy reflexivity so that the substitution for u may be trivial (e.g. when validating $(w : \phi \wedge (\phi \to \psi)) \triangleright (w : \psi)$). In total, we have the universal closure of the following meta-formulae:

$$\begin{array}{c} (x : \hat{\phi} \to \hat{\psi}) \iff \forall y\big((xRy)\&(y : \hat{\phi}) \Rightarrow (y : \hat{\psi})\big) \\ xRx \qquad xRy \Rightarrow \forall \hat{\Gamma}\big((x : \hat{\Gamma}) \Rightarrow (y : \hat{\Gamma})\big) \end{array}$$

Observe that we have introduced an ancillary relation R precisely to recover the behaviour determined by the algebraic constraints; curiously, we do not need transitivity, which would render R a pre-order and recover Kripke's semantics for IPL [42] —we discuss this further at the end of Section 6.4. Moreover, since the data-constructors behave exactly as conjunction ($\wedge$) and disjunction ($\vee$), we may replace $\hat{\Gamma}$ with $\hat{\phi}$ without loss of generality.

This concludes the analysis. Altogether, the meta-formulae thus generated comprise a tractable definition for a model-theoretic semantics for IPL, called Ω_{IPL}. Any abstraction of this theory gives the semantics.

$w \Vdash \mathrm{p}$	iff	$w \in [\![\mathrm{p}]\!]$
$w \Vdash \phi \wedge \psi$	iff	$w \Vdash \phi$ and $w \Vdash \psi$
$w \Vdash \phi \vee \psi$	iff	$w \Vdash \phi$ or $w \Vdash \psi$
$w \Vdash \phi \to \psi$	iff	for any u, if wRu and $u \Vdash \phi$, then $u \Vdash \psi$
$w \Vdash \bot$		never

FIGURE 12. Satisfaction for IPL.

DEFINITION 6.12 (Intuitionistic Frame, Satisfaction and Model).
An intuitionistic frame is a pair $\mathcal{F} := \langle \mathbb{V}, R \rangle$ in which R is a reflexive relation on $\mathbb{V}$.

Let $[\![-]\!]$ be an interpretation mapping $\mathscr{J}$-atoms to $\mathbb{U}$. Intuitionistic satisfaction is the relation between elements $w \in \mathbb{V}$ and $\phi \in \mathbb{F}$ defined by the clauses of Figure 12.

A pair $\langle \mathcal{F}, [\![-]\!] \rangle$ is an intuitionistic model iff it is persistent—i.e. for any $\mathscr{J}$-formula ϕ and worlds w and v,

$$\text{if } wRv \text{ and } w \Vdash \phi, \text{ then } v \Vdash \phi.$$

The class of all intuitionistic models is $\mathbb{K}$.

This semantics generates the following validity judgment:

$$\Gamma \vDash_{\mathrm{IPL}} \Delta \qquad \text{iff} \qquad \text{for any } \mathfrak{M} \in \mathbb{K} \text{ and any } w \in \mathfrak{M}, \text{ if } w \Vdash \Gamma, \text{ then } w \Vdash \Delta.$$

It remains to prove soundness and completeness for the semantics, which we do in Section 6.4. Of course, we have *designed* the semantics so that it corresponds to LJ^+, rendering the proof a formality. Nonetheless, it is instructive to see how it unfolds.

As a remark, *tertium non datur* is known not to apply in IPL. How does encoding of IPL within classical logic avoid it? It is instructive to study this question as it explicates the clause for implication, which defines the intuitionistic connective in terms of a (meta-level) classical one.

EXAMPLE 6.13
The following reduction is a canonical instances of *using* the clause (see Section 5.2):

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\Omega_{\mathrm{IPL}}, (wRu), (u : \phi) \triangleright (w : \phi), \bot}{\Omega_{\mathrm{IPL}}, (wRu \,\&\, u : \phi) \triangleright (w : \phi), \bot}\,\&_{\mathsf{L}}}{\Omega_{\mathrm{IPL}} \triangleright (w : \phi), (wRu \,\&\, u : \phi \Rightarrow \bot)}\,\Rightarrow_{\mathsf{R}}}{\Omega_{\mathrm{IPL}} \triangleright (w : \phi), \forall x (wRx \,\&\, x : \phi \Rightarrow \bot)}\,\forall_{\mathsf{R}}}{\Omega_{\mathrm{IPL}} \triangleright (w : \phi), (w : \phi \to \bot)}\,\to\text{-clause}}{\Omega_{\mathrm{IPL}} \triangleright (w : \phi) \parr (w : \phi \to \bot)}\,\parr_{\mathsf{R}}}{\Omega_{\mathrm{IPL}} \triangleright (w : \phi \vee \phi \to \bot)}\,\vee\text{-clause}$$

Since $(u : \phi)$ in the antecedent and $(w : \phi)$ in the succedent are different atoms, since u and w are different world-variables, one has not reached an axiom. In short, despite working in a classical

system, the above calculation witnesses that $\phi \vee \neg\phi$ is valid in IPL if and only if one already knows that ϕ is valid in IPL or one already knows that $\neg\phi$ is valid in IPL.

Curiously, in all instances above, we established the clause for a connective by analysing the behaviour of the connective in the succedent (i.e. on the right-hand side of sequents), paying no attention to its behaviour in the antecedent (i.e. in the left-hand side of sequents). Nevertheless, despite this bias in all the above cases, the clauses generated behave correctly on both sides of the sequents. This observation is significant because it reaffirms an impactful statement by Gentzen [27]:

> The introductions represent, as it were, the 'definitions' of the symbols concerned, and the eliminations are no more, in thefinal analysis, than the consequences of these definitions. This fact may be expressed as follows: in eliminating a symbol, we may use the formula with whose terminal symbol we are dealing only 'in the sense afforded it by the introduction of that symbol.'

This statement is one of the warrants for proof-theoretic semantics [65], a mathematical instantiation of *inferentialism*—the semantic paradigm according to which inferences and the rules of inference determine the meaning of expressions. The method for semantics presented here supports the inferentialist view as it precisely determines the meaning of the connectives according to their inferential behaviour. Notably, that the left- and right- behaviours cohere is intuitively a consequence of *harmony*, which is a pre-condition for proof-theoretic semantics—see e.g. Schroeder-Heister [66].

6.4 Soundness and Completeness

There are two relationships the proposed semantics may have with IPL: soundness and completeness. Since the semantics generated is the same as the one given by Kripke [42], both of these properties are known to hold. Nonetheless, the method by which the semantics was determined gives a different method for establishing these relationships—namely, those of Section 5. In summary, we need only show that relational calculus generated by the semantics has the same behaviour (i.e. is faithful and adequate) as a sequent calculus for IPL. This is a formalized reading of the traditional approach to soundness in which one demonstrates that every rule of the system can be simulated by the clauses of the semantics—in the parlance of this paper, this is just showing that the relational calculus generated by the semantics is faithful to the sequent calculus.

THEOREM 6.14 (Soundness).
If $\Gamma \vdash \phi$, then $\Gamma \models_{\mathrm{IPL}} \phi$.

PROOF. Apply the traditional inductive proof—see e.g. Van Dalen [69]. □

In this paper, we shall prove completeness symmetrically. We show that the relational calculus generated by the semantics is adequate for a sequent calculus characterizing IPL. This suffices because the relational calculus is sound and complete for the semantics, as per Theorem 5.12. To simplify the presentation, we shall have contraction explicit in the relational calculus for Ω_{IPL} rather than implicit.

DEFINITION 6.15 (Relational calculus RJ).
The relational calculus RJ is comprised of the rules in Figure 13—Φ denotes a meta-formula, Π and Σ denote multiset of meta-formulae, x and y denote world-variables, Δ denotes object-logic data and ϕ and ψ denote object-logic formulae. The rules $_{,\mathsf{L}}$ and $^{;}_{\mathsf{R}}$ are invertible, and the world-variable y does not appear elsewhere in the sequents in $\rightarrow_{\mathsf{R}}$.

$$\frac{\Phi, \Phi, \Pi \triangleright \Sigma}{\Phi, \Pi \triangleright \Sigma}\ \mathsf{c} \qquad \frac{}{\Phi, \Pi \triangleright \Sigma, \Phi}\ \mathsf{taut} \qquad \frac{}{\bot, \Pi \triangleright \Sigma}\ \bot \qquad \frac{}{\Pi \triangleright \Sigma, xRx}\ \mathsf{ref}$$

$$\frac{(x:\phi), (x:\psi), \Pi \triangleright \Sigma}{(x:\phi \wedge \psi), \Pi \triangleright \Sigma}\ \wedge_{\mathsf{L}} \qquad \frac{\Pi \triangleright \Sigma, (x:\phi) \quad \Pi \triangleright \Sigma, (x:\psi)}{\Pi \triangleright \Sigma, (x:\phi \wedge \psi)}\ \wedge_{\mathsf{R}}$$

$$\frac{(x:\phi), \Pi \triangleright \Sigma \quad (x:\psi), \Pi \triangleright \Sigma}{(x:\phi \vee \psi), \Pi \triangleright \Sigma}\ \vee_{\mathsf{L}} \qquad \frac{\Pi \triangleright \Sigma, (x:\phi), (x:\psi)}{\Pi \triangleright \Sigma, (x:\phi \vee \psi)}\ \vee_{\mathsf{R}}$$

$$\frac{\Pi \triangleright \Sigma, (xRy) \quad \Pi \triangleright \Sigma, (y:\phi) \quad (y:\psi), \Pi \triangleright \Sigma}{(x:\phi \to \psi), \Pi \triangleright \Sigma}\ \to_{\mathsf{L}}$$

$$\frac{(xRy), (y:\phi), \Pi \triangleright \Sigma, (y:\psi)}{\Pi \triangleright \Sigma, (x:\phi \to \psi)}\ \to_{\mathsf{R}} \qquad \frac{(xRy), (x:\Gamma), (y:\Gamma), \Pi \triangleright \Sigma}{(xRy), (x:\Gamma), \Pi \triangleright \Sigma}\ \mathsf{pers}$$

$$\frac{\bot, \Pi \triangleright \Sigma}{(x:\bot), \Pi \triangleright \Sigma}\ \bot_{\mathsf{L}} \qquad \frac{\Pi \triangleright \Sigma, \bot}{\Pi \triangleright \Sigma, (x:\bot)}\ \bot_{\mathsf{L}}$$

$$\frac{(x:\Gamma), (x:\Gamma'), \Pi \triangleright \Sigma}{(x:\Gamma\ ;\ \Gamma'), \Pi \triangleright \Sigma}\ ;_{\mathsf{L}} \qquad \frac{\Pi \triangleright \Sigma, (x:\Delta), (x:\Delta')}{\Pi \triangleright \Sigma, (x:\Delta \fatsemi \Delta')}\ \fatsemi_{\mathsf{R}}$$

FIGURE 13. Relational Calculus RJ.

COROLLARY 6.16
$\Gamma \vDash_{\mathsf{IPL}} \Delta$ iff $\Gamma \vdash^{\sigma}_{\mathsf{IPL}} \Delta$

PROOF. Instance of Theorem 5.12. □

We desire to transform RJ into a sequent calculus for which it is adequate, which we may then show is a characterization of IPL. Such transformations are discussed in Section 5.3, but the rules of IPL are slightly too complex for the procedure of that section to apply immediately. Therefore, we require some additional meta-theory.

The complexity comes from the $\to$-clause as it may result in non-BVSs. However, we immediately use persistence to create a composite behaviour that a basic rule can capture. This is because the combined effect yields BVSs whose contents may be partitioned; by design, persistence uses world-variables that do not, and cannot, interact throughout the rest of the proof.

DEFINITION 6.17 (World-independence).
Let Π and Σ be lists of meta-formulae. The lists Π and Σ are world-independent iff the set of world-variable in Π is disjoint from the set of world-variables in Σ.

Let $\mathfrak{S}$ be a tractable semantics and let Ω be a tractable definition of it. Let Π_1, Σ_1 and Π_2, Σ_2 be world-independent lists of meta-formulae. The semantics $\mathfrak{S}$ has world-independence iff, if $\Omega, \Pi_1, \Pi_2 \blacktriangleright \Sigma_1, \Sigma_2$, then either $\Omega, \Pi_1 \blacktriangleright \Sigma_1$ or $\Omega, \Pi_2 \blacktriangleright \Sigma_2$.

Intuitively, world-independence says that whatever is true at a world in the semantics does not depend on truth at a world not related to it.

Let $\Pi_i^1, \Pi_i^2, \Sigma_i^1$ and Σ_i^2 be lists of meta-formulae, for $1 \leq i \leq n$, and suppose that Π_i^1, Σ_i^1 is world-independent from Π_i^2, Σ_i^2. Consider a rule of the following form:

$$\frac{\Pi_1^1, \Pi_1^2 \triangleright \Sigma_1^1, \Sigma_1^2 \quad \ldots \quad \Pi_n^1, \Pi_n^2 \triangleright \Sigma_n^1, \Sigma_n^2}{\Pi \triangleright \Sigma}$$

Assuming world-independence of the semantics, this rule can be replaced by the following two rules:

$$\frac{\Pi_1^1 \triangleright \Sigma_1^1 \quad \ldots \quad \Pi_n^1 \triangleright \Sigma_n^1}{\Pi \triangleright \Sigma} \qquad \frac{\Pi_2^1 \triangleright \Sigma_2^1 \quad \ldots \quad \Pi_n^2 \triangleright \Sigma_n^2}{\Pi \triangleright \Sigma}.$$

If all the lists were basic, iterating these replacements may eventually yield a set of basic rules with the same expressive power as the original rule.

PROPOSITION 6.18
The semantics of IPL—i.e. the semantics $\langle \mathbb{K}, \Vdash \rangle$ defined by Ω_{IPL}—has world-independence.

PROOF. If $\Omega_{\mathrm{IPL}}, \Pi_1, \Pi_2 \vdash \Sigma_1, \Sigma_2$, then there is a G3-proof $\mathcal{D}$ of it. We proceed by induction on the number of resolutions in such a proof.

BASE CASE. Recall, without loss of generality, an instantiation of any clause from Ω_{IPL} is a resolution. Therefore, if $\mathcal{D}$ contains no resolutions, then $\Omega_{\mathrm{IPL}}, \Pi_1, \Pi_2 \vdash \Sigma_1, \Sigma_2$ is an instance of taut. In this case, either $\Omega_{\mathrm{IPL}}, \Pi_1 \vdash \Sigma_1$ or $\Omega_{\mathrm{IPL}}, \Pi_2 \vdash \Sigma_2$ is also an instance of taut, by world-independence.

INDUCTION STEP. After a resolution of a sequent of the form $\Omega_{\mathrm{IPL}}, \Pi_1, \Pi_2 \vdash \Sigma_1, \Sigma_2$, one returns a meta-sequent of the same form—i.e. a meta-sequent in which we may partition the meta-formulae in the antecedent and succedent into world-independent multisets. This being the case, the result follows immediately from the induction hypothesis.

The only non-obvious case is in the case of a closed resolution using the $\rightarrow$-clause in the antecedent because they have universal quantifiers that would allow one to produce a meta-atom that contains both a world from Σ_1, Π_1 and Σ_2, Π_2 simultaneously, thereby breaking world-independence.

Let $\Pi_1 = \Pi_1', (w \Vdash \phi \rightarrow \psi)$ and suppose u is a world-variable appearing in Σ_2, Π_2. Consider the following computation—for readability, we suppress Ω_{IPL}:

$$\dfrac{\dfrac{\dfrac{\dfrac{\Pi_1', \Pi_2 \triangleright \Sigma_1, \Sigma_2, wRu \qquad \Pi_1', \Pi_2 \triangleright \Sigma_1, \Sigma_2, (u : \phi)}{\Pi_1', \Pi_2 \triangleright \Sigma_1, \Sigma_2, (wRu)\ \&\ (u : \phi)}\ \&_{\mathsf{L}} \qquad \Pi_1', \Pi_2, (u : \psi) \triangleright \Sigma_1, \Sigma_2}{\Pi_1', (wRu\ \&\ (u : \phi) \Rightarrow u : \psi), \Pi_2 \triangleright \Sigma_1, \Sigma_2}\ \Rightarrow_{\mathsf{L}}}{\Pi_1', \forall x(wRx\ \&\ x : \phi \Rightarrow x : \psi), \Pi_2 \triangleright \Sigma_1, \Sigma_2}\ \forall_{\mathsf{L}}}{\Pi_1', (w : \phi \rightarrow \psi), \Pi_2 \triangleright \Sigma_1, \Sigma_2}\ \rightarrow\text{-clause}$$

The wRu may be deleted (by w_L) from the leftmost premiss because the only way for the meta-atom to be used in the remainder of the proof is if wRu appears in the context, but this is impossible (by world-independence). Hence, without loss of generality, this branch reduces to $\Pi_1', \Pi_2 \vdash \Sigma_1, \Sigma_2$. Each premiss now has the desired form. □

Using world-independence, we may give a relational calculus RJ^+ characterizing the semantics comprised of basic rules. It arises from analysing the rôle of the atom xRy in RJ in an effort to get rid of it. Essentially, we incorporate it in $\rightarrow_{\mathsf{R}}$, which was always its purpose—see Section 6.3.

$$\frac{\Phi, \Phi, \Pi \triangleright \Sigma}{\Phi, \Pi \triangleright \Sigma}\ \mathsf{c} \qquad \frac{}{\Phi, \Pi \triangleright \Sigma, \Phi}\ \mathsf{taut} \qquad \frac{}{\bot, \Pi \triangleright \Sigma}\ \bot \qquad \frac{}{\Pi \triangleright \Sigma, xRx}\ \mathsf{ref}$$

$$\frac{(x:\phi), (x:\psi), \Pi \triangleright \Sigma}{(x:\phi \wedge \psi), \Pi \triangleright \Sigma}\ \wedge_{\mathsf{L}} \qquad \frac{\Pi \triangleright \Sigma, (x:\phi) \quad \Pi \triangleright \Sigma, (x:\psi)}{\Pi \triangleright \Sigma, (x:\phi \wedge \psi)}\ \wedge_{\mathsf{R}}$$

$$\frac{(x:\phi), \Pi \triangleright \Sigma \quad (x:\psi), \Pi \triangleright \Sigma}{(x:\phi \vee \psi), \Pi \triangleright \Sigma}\ \vee_{\mathsf{L}} \qquad \frac{\Pi \triangleright \Sigma, (x:\phi), (x:\psi)}{\Pi \triangleright \Sigma, (x:\phi \vee \psi)}\ \vee_{\mathsf{R}}$$

$$\frac{\Pi \triangleright \Sigma, (x:\phi) \quad (x:\psi), \Pi \triangleright \Sigma}{(x:\phi \to \psi), \Pi \triangleright \Sigma}\ \to_{\mathsf{L}} \qquad \frac{(y:\phi), \Pi[x \mapsto y] \triangleright (x:\psi)}{\Pi \triangleright \Sigma, (x:\phi \to \psi)}\ \to_{\mathsf{R}}$$

$$\frac{\bot, \Pi \triangleright \Sigma}{(x:\bot), \Pi \triangleright \Sigma}\ \bot_{\mathsf{L}} \qquad \frac{\Pi \triangleright \Sigma, \bot}{\Pi \triangleright \Sigma, (x:\bot)}\ \bot_{\mathsf{R}}$$

$$\frac{(x:\Gamma), (x:\Gamma'), \Pi \triangleright \Sigma}{(x:\Gamma \,;\, \Gamma'), \Pi \triangleright \Sigma}\ ;_{\mathsf{L}} \qquad \frac{\Pi \triangleright \Sigma, (x:\Delta), (x:\Delta')}{\Pi \triangleright \Sigma, (x:\Delta \,\mathring{;}\, \Delta')}\ \mathring{;}_{\mathsf{R}}$$

FIGURE 14. Relational Calculus RJ^+.

DEFINITION 6.19 (System RJ^+).
System RJ^+ is comprised of the rules in Figure 14, in which $;_{\mathsf{L}}$ and $\mathring{;}_{\mathsf{R}}$ are invertible.

PROPOSITION 6.20
$\Gamma \vdash_{RJ} \Delta$ iff $\Gamma \vdash_{RJ^+} \Delta$

PROOF. Every RJ-proof can be simulated in RJ^+ by using (i.e. reduce with) pers eagerly after using $\to_{\mathsf{L}}$. Thus, $\Gamma \vdash_{RJ} \Delta$ implies $\Gamma \vdash_{RJ^+} \Delta$. It remains to show that $\Gamma \vdash_{RJ^+} \Delta$ implies $\Gamma \vdash_{RJ} \Delta$.

Without loss of generality, in RJ^+ one may always use pers immediately after $\to_{\mathsf{R}}$, as otherwise the use of $\to_{\mathsf{L}}$ could be postponed. Similarly, without loss of generality, $\to_{\mathsf{L}}$ always instantiates with $y = x$—this follows as we require the leftmost branch of the following to close, which it does by ref:

$$\frac{\Pi \triangleright \Sigma, (xRx) \qquad \Pi \triangleright \Sigma, (x:\phi) \qquad \Pi, (x:\psi) \triangleright \Sigma}{\Pi, (x:\phi \to \psi) \triangleright \Sigma}$$

An RJ^+-proof following these principles maps to an RJ-proof simply by collapsing the instances of $\to_{\mathsf{L}}$ and $\to_{\mathsf{R}}$ in the former to capture $\to_{\mathsf{L}}$ and $\to_{\mathsf{R}}$ in the latter. □

Observe that the propositional encoding of RJ^+ is precisely LJ^+. The connexion to IPL follows immediately:

COROLLARY 6.21
System RJ^+ is faithful and adequate with respect to LJ^+.

PROOF. Instance of Theorem 5.18. □

THEOREM 6.22 (Completeness).
If $\Gamma \vDash_{\mathrm{IPL}} \Delta$, then $\Gamma \vdash \Delta$.

PROOF. We have the following:

$$\begin{array}{lll} \Gamma \vDash_{\mathrm{IPL}} \Delta & \text{implies} & (w:\Gamma) \vdash_{RJ} (w:\Delta) \qquad \text{(Corollary 6.16)} \\ & \text{implies} & (w:\Gamma) \vdash_{RJ^+} (w:\Delta) \qquad \text{(Proposition 6.20)} \\ & \text{implies} & \Gamma \vdash_{LJ^+} \Delta \qquad \text{(Corollary 6.21)} \\ & \text{implies} & \Gamma \vdash_{LJ} \Delta \qquad \text{(Proposition 6.10).} \end{array}$$

Since LJ characterizes IPL, this completeness the proof. □

Thus, we have derived a semantics of IPL for LJ and proved its soundness and completeness using the constraint systems. This semantics is not quite Kripke's one [42], which insists that R be transitive, thus rendering it a pre-order. This requirement is naturally seen from the connection to Heyting algebra and the modal logic S4. In the analysis of Section 6.3, from which the semantics in this paper comes, there was no need for transitivity; the proofs of soundness and completeness go through without it.

One may add transitivity—i.e. the meta-formula $\forall x, y, z(xRy \& yRz \implies xRz)$—to Ω_{IPL} and proceed as above, adding the following rule to RJ:

$$\frac{xRy, yRz, xRz, \Pi \triangleright \Sigma}{xRy, yRz, \Pi \triangleright \Sigma}.$$

The proof of completeness passes again through RJ^+ by observing that eagerly using persistence does all the work required of transitivity; i.e. according to the eager use of pers, in a sequent $xRy, yRz, \Pi \triangleright \Sigma$, the set Π is of the form $\Pi'[x \mapsto y] \cup \Pi'[y \mapsto z]$, so that whatever information was essential about (the world denoted by) x is already known about (the world denoted by) z by passing through (the world denoted by) y.

This concludes the case-study of IPL: we have used constraint systems to decompose it into classical logic from which we derived a semantics and proved soundness and completeness. Adding other axioms to Ω_{IPL}, ones that are not redundant in the above sense recovers various intermediate logics. In this way, constraint systems offer a uniform and modular approach to studying them, which we leave as future work.

7 Extension: First-order Logic

So far, we have concentrated on propositional logics to enable a uniform account of constraint systems across a large class of logics. Nevertheless, there is nothing within the paradigm that is inherently propositional. In this section, we extend the phenomenon of decomposing a logical system according to its combinatorial aspect and algebraic aspects to the setting of first-order logic, as in the case of the original example of a constraint system (i.e. RDvBC in Section 2), constraint systems have computational advantages. Therefore, we illustrate the extension to first-order logic by application to *logic programming*.

Logic programming (LP) is the programming language paradigm whose operational semantics is based on proof-search (see e.g. Miller *et al.* [51]). It is a core discipline in (symbolic) artificial intelligence as proof-search is used to characterize reasoning. The central part of LP is a step known as *resolution*, and the most challenging aspect of resolution is a process called *unification*; the output of the execution of a configuration in LP is a *unifier*. A resolution in LP is a reductive application of a quantifier left-rule combined with an implication left-rule in a sequent calculus; unification is the choice of substitution in the quantifier rule.

In this section, we show how algebraic constraints may handle unification. The authors have discussed this idea in earlier work [29], but in a limited way, as the underlying framework of algebraic constraints had yet to be developed.

7.1 A basic logic programming language

The Basic Logic Programming language (BLP) is based on uniform proof-search in the hereditary Harrop fragment of intuitionistic logic (see Miller *et al.* [50, 51]), which we think of as the *basic logic* (B).

Fix a first-order alphabet $\mathscr{A} = \langle \mathbb{R}, \mathbb{F}, \mathbb{K}, \mathbb{V} \rangle$. Denote the set of atoms by $\mathbb{P}$.

DEFINITION 7.1 (Goal Formula, Definite Clauses, Programs, Query).
The set of goal formulae $\mathbb{G}$ and definite clauses $\mathbb{D}$ are the sets of formula G and D defined as follows, respectively:

$$\begin{array}{lll} G & ::= & A \in \mathbb{P} \mid G \mid D \to G \mid G \wedge G \mid G \vee G \\ D & ::= & A \in \mathbb{P} \mid D \mid G \to A \mid D \wedge D. \end{array}$$

A set P of definite formula is a *program*. A query is a pair $P \triangleright G$ in which P is a program and G is goal-formula.

There is an apparent lack of quantifiers in the language. However, restricting attention to goal formulae and definite clauses means the quantifiers can be suppressed without loss of information: definite clauses containing a variable as universally quantified, and goal formulae containing a variable as existentially quantified.

DEFINITION 7.2 (Substitution).
A substitution is a mapping $\theta : \mathbb{V} \to \mathbb{T}$. If ϕ is a formula (i.e. a definite clause or a goal), then $\phi\theta$ is the formula that results from replacing every occurrence of a variable in ϕ by its image under θ.

Having fixed a program P, one gives the system a goal G with the desire of finding a substitution θ such that $P \vdash G\theta$ obtains in IL. The substitution θ is the unifer. It is the object that the language BLP computes upon an input query. The operational semantics for BLP is given by proof-search in LB.

DEFINITION 7.3 (Sequent calculus LB).
Sequent calculus LB comprises the rules of Figure 15.

The operational semantics of BLP is as follows: beginning with the query $P \triangleright G$, one gives a candidate θ and checks by reducing in LB (in the sense of Section 4.1) whether or not $P \vdash G\theta$ obtains or not. The first step (i.e. introducing θ) is also captured by a reduction operator in LB—namely, the $\exists_{\mathsf{R}}$-rule—hence, the operational semantics is entirely based on reduction.

The following example, also appearing in Gheorghiu *et al.* [29], illustrates how BLP works:

EXAMPLE 7.4
To complete the informatics course at Unseen University, students must pick one module for each of the three terms of the year, which are called R(ed), G(reen) and B(lue), respectively. The available

$$\frac{P \mathbin{,} D \rhd G}{P \rhd D \to G}\ {\to_{\mathsf{R}}} \qquad \frac{P \rhd G_0 \quad P \rhd G_1}{P \rhd G_0 \wedge G_1}\ {\wedge_{\mathsf{R}}} \qquad \frac{P \rhd G_i}{P \rhd G_0 \vee G_1}\ {\vee_{\mathsf{R}}} \qquad \frac{}{P \mathbin{,} A \rhd A}\ \mathsf{ax}$$

$$\frac{P \mathbin{,} G \to A \rhd G}{P \mathbin{,} G \to A \rhd A}\ {\to_{\mathsf{L}}} \qquad \frac{P \mathbin{,} D_0 \mathbin{,} D_1 \rhd A}{P \mathbin{,} D_0 \wedge D_1 \rhd A}\ {\wedge_{\mathsf{L}}} \qquad \frac{P \mathbin{,} D \mathbin{,} D\theta \rhd A}{P \mathbin{,} D \rhd A}\ {\forall_{\mathsf{L}}} \qquad \frac{P \rhd G\theta}{P \rhd G}\ {\exists_{\mathsf{R}}}$$

FIGURE 15. Sequent Calculus LB.

Red	Green	Blue
Al(gebra)	Lo(gic)	Da(tabases)
Pr(obability)	Ca(tegories)	Co(mpilers)
Gr(aphs)	Au(tomata)	AI

FIGURE 16. The Extensional Database.

choices are shown in Figure 16. More formally, we have a relation S that obtains for valid selections of courses; e.g.

$S(Al, Ca, Co)$ obtains, but $S(Pr, Gr, Ca)$ does not.

We may use BLP to capture this situation. The setup is captured by a program P composed of two parts: the *extensional* database ED and the *intensional* database ID. The extensional database contains the information about the modules:

$$ED := R(Al), R(Pr), R(Gr), G(Lo), G(Ca), G(Au), B(Da), B(Co), B(AI)$$

Meanwhile, the intensional database comprises the selection logic:

$$ID := R(x) \wedge G(y) \wedge B(x) \to S(x, y, z)$$

To find the possible combinations of modules, one queries the system for different choices of M_1 and M_2 and M_3; i.e. one considers the validity of the following sequent:

$$P \rhd S(M_1, M_2, M_3).$$

One possible execution is the following, in which $\phi := S(Al, Lo, AI) \leftarrow (R(Al) \wedge G(Lo)) \wedge B(AI)$:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{P \mathbin{,} \phi \rhd R(Al)}\ \mathsf{ax} \quad \dfrac{}{P \mathbin{,} \phi \rhd G(Lo)}\ \mathsf{ax} \quad \dfrac{}{P \mathbin{,} \phi \rhd B(AI)}\ \mathsf{ax}}{P \mathbin{,} \phi \rhd (R(Al) \wedge G(Lo)) \wedge B(AI)}\ {\wedge_{\mathsf{R}}}}{P \mathbin{,} \phi \rhd S(Al, Lo, AI)}\ {\to_{\mathsf{L}}}}{P \rhd S(Al, Lo, AI)}\ {\forall_{\mathsf{L}}}}{P \rhd S(x, y, z)}\ {\exists_{\mathsf{R}}}$$

There are $9^3 = 729$ possible ways of selecting three courses out of the nine, and there are only 27 acceptable choices out of these possibilities. In terms of proof-search spaces, there are 729 branches, out of which only 27 may lead to successful reductions. Random selection is, therefore, intractable. Moreover, to the extent that logic programming concerns problem solving (see e.g. Kowalski [40]), this process does not reflect how one hopes a student would reason.

$$\frac{P \,,\, D \rhd G}{P \rhd D \to G}\ {\to_{\mathsf{R}}} \qquad \frac{P \rhd G_0 \quad P \rhd G_1}{P \rhd G_0 \wedge G_1}\ {\wedge_{\mathsf{R}}} \qquad \frac{P \rhd G_i}{P \rhd G_0 \vee G_1}\ {\vee_{\mathsf{R}}} \qquad \frac{\parr_{A \in P \text{ st.} A \sim B}(A \equiv B)}{P \rhd B}\ \mathsf{ax}$$

$$\frac{P \,,\, G \to A \rhd G \quad A \equiv B}{P \,,\, G \to A \rhd B}\ {\to_{\mathsf{L}}} \qquad \frac{P \,,\, D_0 \,,\, D_1 \rhd A}{P \,,\, D_0 \wedge D_1 \rhd A}\ {\wedge_{\mathsf{L}}} \qquad \frac{P \,,\, D \,,\, D\theta \rhd A}{P \,,\, \forall x D \rhd A}\ {\forall_{\mathsf{L}}} \qquad \frac{P \rhd G\theta}{P \rhd G}\ {\exists_{\mathsf{R}}}$$

FIGURE 17. Constraint System $\mathsf{PLB} \oplus \mathcal{U}$.

Example 7.4 is naive in that, rather than simply guessing a possibility, a student would likely observe that one must choose three modules one from each R, G and B, respectively. How may we capture such reasoning? One possibility is to keep the major logical steps in Example 7.4, but without committing to a particular substitution. Instead, the proof structures thus constructed inform us what kinds of substitutions make sense, precisely those that render the proof structure a LB-proof. We may capture this approach with algebraic constraints.

7.2 Unification via Nominal Constraints

A unification algebra allows the substitution in LB to be managed intelligently. It does this by managing unification, as opposed to merely managing substitution, to be expressed logically and enforced at the end of the computation. We only need a way to track what substitution took place and what needs to be unified, which labels governed by equality can handle.

DEFINITION 7.5 (Unification Algebra).
The unification algebra $\mathcal{U}$ consists of the universe $\mathbb{TERM}(\mathscr{A})$ and nothing else.

The notion of *enriched sequents* in this setting extends that for propositional logics in that terms within formulae may also carry labels. For example, $P \rhd R(x \cdot n_1, t_1 \cdot n_2)$, in which $R \in \mathbb{R}$ is a relation-symbol, $x \in \mathbb{V}$ and $t_1 \in \mathbb{TERM}(\mathscr{A})$ are terms and n_1 and n_2 are expressions for the algebra (i.e. variables), is an enriched sequent.

We have no operators in the algebra, but we do have an equality predicate for the constraints. For example, we may write $n = k$, in which n is a variable and k is (interpreted) as a term, to denote that the value of n must be whatever is denoted by k. Let A and B be enriched atoms. We write $A \equiv B$ to denote the meta-disjunction of all equations $n = m$ such that n is the label of a term in A occurring in P and m is a label of a term in an atom B such that A and B have the same relation-symbol and n and m occur on corresponding terms. For example, $R(x_1 \cdot n_1, y_1 \cdot n_2) \equiv R(x_2 \cdot m_1, y_2 \cdot m_2)$ denotes $(n_1 = m_1) \parr (n_2 = m_2)$. The notation denotes falsum when the disjunction is empty (i.e. when there are no correspondences). Let P be a program. We write $\parr_{A \in P \text{ st.} A \sim B}(A \equiv B)$ to denote the disjunction $A \equiv B$ for all A in P. We may also write $n \in \{k_1, ..., k_n\}$ to denote the disjunction $(n = k_1) \parr ... \parr (n = k_n)$.

DEFINITION 7.6 (Constraint System $\mathsf{PLB} \oplus \mathcal{U}$).
Constraint system $\mathsf{LB} \oplus \mathcal{U}$ comprises the rules in Figure 17, in which θ denotes a substitution that always introduce fresh labels.

PROPOSITION 7.7
Constraint system $\mathsf{PLB} \oplus \mathcal{U}$, with valuation σ, is faithful and adequate with respect to LB.

We use $\mathsf{LB} \oplus \mathcal{U}$ to simulate the actual reasoning one intends BLP to represent. To see this, we return to Example 7.4 and show that the constraint system captures the reasoning process one hopes a student would use more realistically.

EXAMPLE 7.8 (Example 7.4 cont'd).
To simplify notation, let $ID \cdot [n_1, n_2, n_3]$ denote $(R(x.n_1) \wedge G(y.n_2)) \wedge B(x.n_3) \rightarrow CS(x.n_1, y.n_2, z.n_3)$. The computation-trace in BLP using the algebraic constraint system $\mathsf{LB} \oplus \mathcal{U}$ is as follows:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{n_1 \in \{Al, Pr, Gr\}}{P \triangleright R(x \cdot n_1)} \quad \dfrac{n_2 \in \{Lo, Ca, Au\}}{P \triangleright G(y \cdot n_2)}}{P \triangleright R(x \cdot n_1) \wedge G(y \cdot n_2)} \quad \dfrac{n_3 \in \{Da, Co, AI\}}{P \triangleright B(z \cdot n_3)}}{P \triangleright (R(x \cdot n_1) \wedge G(y \cdot n_2)) \wedge B(z \cdot n_3)} \quad \begin{array}{l} n_1 = m_1 \\ n_2 = m_2 \\ n_3 = m_3 \end{array}}{P, ID \cdot [n_1, n_2, n_3] \triangleright S(x.m_1, y.m_2, z.m_3)}}{P \triangleright CS(x \cdot m_1, y.m_2, z.m_3)}}{P \triangleright S(x, y, z)}$$

Every valid execution in LB of the initial query is a coherent instantiation of this proof; e.g. to recover the one presented in Example 7.4, we use the following interpretation:

$$I(n_1) := I(m_1) := Al \quad I(n_2) := I(m_2) := Lo \quad I(n_3) := I(m_3) := AI$$

This study of logic programming illustrates that algebraic constraint systems extend quite naturally to the predicate logic setting and have practical applications, as showcased by how it addresses unification in logic programming. Of course, this section is cursory compared to the extensive study of propositional logic in the rest of the paper, with the open question of a general theory.

8 Conclusion

This paper has introduced the concept of a constraint system, which serves as a uniform tool for studying the meta-theory of one logic in terms of the meta-theory of another. The advantage is that the latter may be simpler or more well-understood in some practical sense. In short, a constraint system is a labelled sequent calculus in which the labels carry an algebraic structure to determine correctness conditions on proof structures. A motivating example of a class of constraint system already present in the literature are those captured by the *resource-distribution via Boolean constraints* (RDvBC) mechanism by Harland and Pym [32, 60], which serve as a meta-theoretic tool to analyse the possible context-management strategies during proof-search in logics with multiplicative connectives—see Section 2.

Constraint systems have two possible relationships with a logic of interest: soundness and completeness, and faithfulness and adequacy. The former is a *global* correctness condition that says that completed reductions in the constraint system (i.e. constructions to which one cannot apply further reduction operators) whose constraints are coherent (i.e. admit a solution) characterize the consequence of a logic. Meanwhile, the latter is a *local* correctness condition in that each reduction step in the constraint system corresponds to a valid inference for the logic when its constraints are satisfied; consequently, a completed reduction corresponds to a proof in a sequent calculus for the logic. Both correctness criteria are valuable in applications of constraint systems for studying meta-theory.

We illustrated the framework's usefulness in studying meta-theory through various examples beyond RDvBC. First, in Section 5, we show that they yield a general, uniform, and systematic process for generating relational calculi after Negri [55, 56, 58]; moreover, that relative to this theory of relational calculi, one has an approach to proving soundness and completeness for model-theoretic semantics (M-tS) in the sense of Kripke [42, 43] (see also Beth [3]) that entirely bypasses term- and counter-model constructions (see e.g. van Dalen [69]). An example of this method in practice has already been given in the case of the logic of Bunched Implications by Gheorghiu and Pym [30]. Second, in Section 6, we show by a case-study for IPL that constraint systems enable one to synthesize an M-tS for a logic from an analysis of its proof theory. Overall, these examples witness that constraint systems help bridge the gap between semantics and proof theory for logics. It needs to be clarified how this paper's approach to soundness and completeness relates to the more traditional approaches. However, such investigation may aid in understanding the underlying principles and is left for future work.

While this paper is only concerned with propositional logics, we showed in Section 7 how it could intuitively be extended to the first-order setting. Moreover, there are good computational reasons for doing such extensions in the context of logic as a reasoning technology.

Of course, this paper concerns only the initial framework for algebraic constraint systems. Future work includes giving more examples of constraint systems and developing the applications presented in this paper. For example, the case analysis of deriving semantics for IPL should be repeated for other adjacent logics, especially intermediate, hybrid and substructural logics. Moreover, in this paper, we have only considered three different notions of algebra—Boolean algebra, world algebra (i.e. the algebra corresponding to a frame) and unification algebra—what are some other valuable algebras and constraint systems, and what can they tell us about the logics being studied?

Overall, constraint systems provide a general framework for defining and studying logics and have the potential to bridge the gap between model theory and proof theory, as well as give insights about proof-search.

Acknowledgements

We are grateful to Tao Gu and the referees for their thorough and thoughtful comments on this work.

References

[1] M. Baldoni. *Normal Multimodal Logics: Automatic Deduction and Logic Programming Extension*. PhD Thesis, Università degli Studi di Torino, 1998.

[2] D. Basin, S. Matthews and L. Viganò. Natural deduction for non-classical logics. *Studia Logica*, **60**, 119–160, 1998.

[3] E. W. Beth. Semantic construction of intuitionistic logic. *Indagationes Mathematicae*, **17**, 327–338, 1955.

[4] P. Blackburn. Internalizing labelled deduction. *Journal of Logic and Computation*, **10**, 137–168, 2000.

[5] P. Blackburn, M. de Rijke and Y. Venema. *Modal Logic*. Cambridge Tracts in Theoretical Computer Science. Cambridge University Press, 2014.

[6] P. Blackburn and J. Seligman. Hybrid languages. *Journal of Logic, Language and Information*, **4**, 251–272, 1995.

[7] T. Braüner. *Hybrid Logic and Its Proof-Theory*. Springer, 2011.

[8] A. Bundy. *The Computer Modelling of Mathematical Reasoning*. Academic Press Professional, Inc., 1985.
[9] B. ten Cate and C. Areces. Hybrid logic. In *Handbook of Modal Logic*, P. Blackburn and M. Marx., eds, pp. 893–939. Elsevier, 2007.
[10] L. Catach. TABLEAUX: a general theorem prover for modal logics. *Journal of Automated Reasoning*, **7**, 489–510, 1991.
[11] I. Cervesato, J. S. Hodas and F. Pfenning. Efficient resource management for linear logic proof search. *Theoretical Computer Science*, **232**, 133–163, 2000.
[12] A. Ciabattoni, N. Galatos and K. Terui. From axioms to analytic rules in nonclassical logics. In *Logic in Computer Science—LICS*. IEEE Computer Society, 2008.
[13] A. Ciabattoni, N. Galatos and K. Terui. Algebraic proof theory for substructural logics: cut-elimination and completions. *Annals of Pure and Applied Logic*, **163**, 266–290, 2012.
[14] A. Ciabattoni, L. Straßburger and K. Terui. Expanding the realm of systematic proof theory. In *Computer Science Logic—CSL*, E. Grädel and R. Kahle., eds, pp. 163–178. Springer, 2009.
[15] S. Docherty. *Bunched Logics: A Uniform Approach*. PhD Thesis, University College London, 2019.
[16] S. Docherty and D. J. Pym. Modular tableaux calculi for separation theories. In *Foundations of Software Science and Computation Structures—FoSSaCS*, C. Baier and U. D. Lago., eds, pp. 441–458, 2018.
[17] S. Docherty and R. N. S. Rowe. A non-wellfounded, labelled proof system for propositional dynamic logic. In *Automated Reasoning With Analytic Tableaux and Related Methods—TABLEAUX*, pp. 335–352. Springer, 2019.
[18] M. Dummett. *Elements of Intuitionism*. Oxford Logic Guides, vol. 39. Oxford University Press, 2000.
[19] M. Fitting. *Proof Methods for Modal and Intuitionistic Logics*. Springer, 1983.
[20] M. Fitting and R. L. Mendelsohn. *First-Order Modal Logic*. Springer, Dordrecht, 1998.
[21] D. M. Gabbay. *Classical vs Non-Classical Logics: The Universality of Classical Logic (MPI-I-93-230)*.
[22] D. M. Gabbay. *Labelled Deductive Systems*. Oxford University Press, 1996.
[23] D. Galmiche and D. Méry. Semantic labelled tableaux for propositional $\mathrm{BI}^{\perp}$. *Journal of Logic and Computation*, **13**, 707–753, 2003.
[24] D. Galmiche and D. Méry. Tableaux and resource graphs for separation logic. *Journal of Logic and Computation*, **20**, 189–231, 2010.
[25] D. Galmiche, D. Méry and D. Pym. Resource Tableaux. In *Computer Science Logic—CSL*, pp. 183–199. Springer, 2002.
[26] D. Galmiche, D. Méry and D. Pym. The semantics of BI and resource tableaux. *Mathematical Structures in Computer Science*, **15**, 1033–1088, 2005.
[27] G. Gentzen. Investigations into logical deduction. In *The Collected Papers of Gerhard Gentzen*, M. E. Szabo., ed. North-Holland Publishing Company, 1969.
[28] A. Gheorghiu and S. Marin. Focused proof-search in the logic of bunched implications. In *Foundations of Software Science and Computation Structures—FoSSaCS*, S. Kiefer and C. Tasson., eds, pp. 247–267. Springer, 2021.
[29] A. V. Gheorghiu, S. Docherty and D. J. Pym. Reductive logic, coalgebra, and proof-search: a perspective from resource semantics. In *Samson Abramsky on Logic and Structure in Computer Science and Beyond*. Springer Outstanding Contributions to Logic Series, A. Palmigiano and M. Sadrzadeh., eds. Springer, 2021, in press.

[30] A. V. Gheorghiu and D. J. Pym. Semantical analysis of the logic of bunched implications. In *Studia Logica*, vol. 11, pp. 525–571, 2023.
[31] K. Gödel. On the completeness of the calculus of logic. In *Kurt Gödel: Collected Works: Publications 1929–1936*, S. Feferman, J. W. Dawson, S. C. Kleene, G. H. Moore, R. M. Solovay and J. van Heijenoort., eds, vol. 1. Oxford University Press, 1986.
[32] J. Harland and D. J. Pym. Resource-distribution via Boolean constraints. In *Automated Deduction—CADE*, W. McCune., ed., pp. 222–236. Springer, 1997.
[33] J. Harland and D. J. Pym. Resource-distribution via Boolean constraints. *ACM Transactions on Computational Logic*, **4**, 56–90, 2003.
[34] J. S. Hodas. *Logic programming in intutionistic linear logic*. In *Theory, Design and Implementation*. PhD Thesis,, vol. 110, pp. 327–365. University of Pennsylvania, 1994.
[35] J. S. Hodas and D. Miller. Logic programming in a fragment of intuitionistic linear logic. *Information and Computation*, **110**, 327–365, 1994.
[36] A. Indrzejczak. *Natural Deduction, Hybrid Systems and Modal Logics*. Springer, 2010.
[37] A. Indrzejczak. *Sequents and Trees*. Studies in Universal Logic. Springer, 2021.
[38] S. Kanger. *Provability in Logic*. Almqvist & Wiksell, 1957.
[39] S. C. Kleene. *Mathematical Logic*. Dover Publications, 2002.
[40] R. Kowalski. *Logic for Problem-Solving*. North-Holland Publishing Co., 1986.
[41] R. Kowalski and D. Kuehner. Linear resolution with selection function. *Artificial Intelligence*, **2**, 227–260, 12, 1971.
[42] S. A. Kripke. Semantical analysis of modal logic I: normal modal propositional calculi. *FoSSaCS—Logic Quarterly*, **9**, 67–96, 1963.
[43] S. A. Kripke. Semantical analysis of intuitionistic logic I. In *Studies in Logic and the Foundations of Mathematics*, vol. 40, pp. 92–130. Elsevier, 1965.
[44] H. Kushida and M. Okada. A proof-theoretic study of the correspondence of classical logic and modal logic. *The Journal of Symbolic Logic*, **68**, 1403–1414, 2003.
[45] C. Liang and D. Miller. Focusing and polarization in linear, intuitionistic, and classical logic. *Theoretical Computer Science*, **410**, 4747–4768, 2009.
[46] P. López and J. Polakow. Implementing efficient resource management for linear logic programming. In *International Conference on Logic for Programming, Artificial Intelligence, and Reasoning—LPAR*, pp. 528–543. Springer, 2005.
[47] S. Marin, D. Miller, E. Pimentel and M. Volpe. From axioms to synthetic inference rules via focusing. *Annals of Pure and Applied Logic*, **173**, 103091, 2022.
[48] M. Marx, S. Mikulás and M. Reynolds. The mosaic method for temporal logics. In *Automated Reasoning With Analytic Tableaux and Related Methods—TABLEAUX*, vol. 9, pp. 324–340. Springer, 2000.
[49] F. Massacci. Strongly analytic tableaux for normal modal logics. In *Conference on Automated Deduction—CADE*, **814**, 723–737, 1994.
[50] D. Miller. A logical analysis of modules in logic programming. *The Journal of Logic Programming*, **6**, 79–108, 1989.
[51] D. Miller, G. Nadathur, F. Pfenning and A. Scedrov. Uniform proofs as a foundation for logic programming. *Annals of Pure and Applied Logic*, **51**, 125–157, 1991.
[52] R. Milner. The use of machines to assist in rigorous proof. *Philosophical Transactions of the Royal Society of London. Series A, Mathematical and Physical Sciences*, **312**, 411–422, 1984.
[53] G. Mints. Cut-free calculi of the S5-type. *Zapiski Nauchnykh Seminarov—LOMI*, **8**, 166–174, 1968.

[54] G. Mints. Indexed systems of sequents and cut-elimination. *Journal of Philosophical Logic*, **26**, 671–696, 1997.
[55] S. Negri. Contraction-free sequent calculi for geometric theories with an application to Barr's theorem. *Archive for Mathematical Logic*, **42**, 389–401, 2003.
[56] S. Negri. Proof analysis in modal logic. *Journal of Philosophical Logic*, **34**, 507–544, 2005.
[57] S. Negri and J. Von Plato. Cut elimination in the presence of axioms. *Bulletin of Symbolic Logic*, **4**, 418–435, 1998.
[58] S. Negri, J. von Plato and A. Ranta. *Structural Proof Theory*. Cambridge University Press, 2001.
[59] P. W. O'Hearn and D. J. Pym. The logic of bunched implications. *Bulletin of Symbolic Logic*, **5**, 215–244, 1999.
[60] D. J. Pym and E. Ritter. On the semantics of classical disjunction. *Journal of Pure and Applied Algebra*, **159**, 315–338, 2001.
[61] D. J. Pym and E. Ritter. *Reductive Logic and Proof-Search: Proof Theory, Semantics, and Control*. Clarendon Press, Oxford Logic Guides, 2004.
[62] D. J. Pym and E. Ritter. A games semantics for reductive logic and proof-search. In *Proceedings of the 1st Workshop on Games for Logic and Programming Languages—GALOP@ETAPS*, D. R. Ghica and G. McCusker., eds, pp. 107–123, 2005.
[63] S. Read. *Relevant Logic*. Basil Blackwell, 1988.
[64] A. M. Russo. *Modal Logics as Labelled Deductive Systems*. PhD Thesis, Department of Computing, Imperial College London, 1996.
[65] P. Schroeder-Heister. Proof-theoretic semantics. In *The Stanford Encyclopedia of Philosophy*, E. N. Zalta., ed., Spring 2018 edn. Metaphysics Research Lab, Stanford University, 2018.
[66] P. Schroeder-Heister. Proof-theoretic semantics. In *The Stanford Encyclopedia of Philosophy*, E. N. Zalta., ed., Spring 2018 edn. Metaphysics Research Lab, Stanford University, 2018.
[67] A. K. Simpson. *The Proof Theory and Semantics of Intuitionistic Modal Logic*. PhD Thesis, University of Edinburgh, 1994.
[68] A. S. Troelstra and H. Schwichtenberg. *Basic Proof Theory*. Cambridge Tracts in Theoretical Computer Science, vol. 43. Cambridge University Press, 2000.
[69] D. van Dalen. *Logic and Structure*. Universitext. Springer, 2012.
[70] L. Viganò. *Labelled Non-Classical Logics*. Springer, 2013.
[71] M. Winikoff and J. Harland. Implementing the linear logic programming language Lygon. In *Proceedings of the International Logic Programming Symposium—ILPS*, pp. 66–80. MIT Press, 1995.

Received 22 May 2022