Bulletin of the **S**ection of **L**ogic
Volume 54/1 (2025), pp. 59–151
https://doi.org/10.18778/0138-0680.2025.02

Tim S. Lyon
Agata Ciabattoni
Didier Galmiche
Marianna Girlando
Dominique Larchey-Wendling
Daniel Méry
Nicola Olivetti
Revantha Ramanayake

INTERNAL AND EXTERNAL CALCULI: ORDERING THE JUNGLE WITHOUT BEING LOST IN TRANSLATIONS

Abstract

This paper gives a broad account of the various sequent-based proof formalisms in the proof-theoretic literature. We consider formalisms for various modal and tense logics, intuitionistic logic, conditional logics, and bunched logics. After

Presented by: Andrzej Indrzejczak
Received: November 29, 2024, **Received in revised form**: April 29, 2025,
Accepted: May 12, 2025, **Published online**: June 6, 2025

providing an overview of the logics and proof formalisms under consideration, we show how these sequent-based formalisms can be placed in a hierarchy in terms of the underlying data structure of the sequents. We then discuss how this hierarchy can be traversed using translations. Translating proofs up this hierarchy is found to be relatively straightforward while translating proofs down the hierarchy is substantially more difficult. Finally, we inspect the prevalent distinction in structural proof theory between 'internal calculi' and 'external calculi.' We discuss the ambiguities involved in the informal definitions of these categories, and we critically assess the properties that (calculi from) these classes are purported to possess.

Keywords: bunched implication, conditional logic, display calculus, external calculus, hypersequent, internal calculus, intuitionistic logic, labeled calculus, modal logic, nested calculus, proof theory, sequent, tense logic.

Introduction

The widespread application of logical methods in computer science, epistemology, and artificial intelligence has resulted in an explosion of new logics. These logics are more expressive than classical logic, allowing for finer distinctions and a direct representation of notions that cannot be well-stated in classical logic. For instance, they are used to express different modes of truth (e.g., modal logics [6]) and to study different types of reasoning, e.g., hypothetical or plausible reasoning (e.g., conditional logics [69]) or reasoning about the separation and sharing of resources (e.g., bunched implication logics [103]). In addition to formalizing reasoning, these logics are also used to model systems and prove properties about them, leading, for example, to applications in software verification (e.g., [89]).

These applications require the existence of analytic calculi. Analytic calculi consist of rules that compose (decompose, in the case of tableau calculi) the formulae to be proved in a stepwise manner, and in particular, the key rule of cut—used to simulate *modus ponens*—is not needed. As a result, the proofs from an analytic calculus possess the subformula property: every formula that appears (anywhere) in the proof is a subformula of the formulae proved. This is a powerful restriction on the form of proofs, which can be exploited to develop automated reasoning methods [107] and establish important properties of logics such as consistency [37, 38], decidability [28], and interpolation [85].

Since its introduction by Gentzen in the 1930s and his seminal proof of the *Hauptsatz* (see [37, 38]), the sequent calculus formalism has become one of the preferred frameworks for constructing analytic calculi. This is because such systems are relatively simple and do not require much technical machinery (that is, 'bureaucracy') to enable analyticity. The downside of this simplicity is that Gentzen sequent calculi are often not expressive enough to capture many logics of interest in an analytic manner. In response, many proof-theoretic formalisms extending Gentzen's formalism have been proposed over the last 50 years to recapture analyticity for more expressive logics. Such formalisms include hypersequent calculi [2, 91], nested sequent calculi [13, 55], bunched sequent calculi [103], display calculi [5, 113], and labeled sequent calculi [106, 112]. Each of these proof-theoretic formalisms (or, *formalisms* for short) is characterized in terms of the standard notation it uses, the data structures employed in sequents, the types of inference rules that normally appear, and the types of properties ordinarily shared by the proof calculi thereof (which serve as *instances* of a formalism). As the notion is central to this paper, we further remark that a proof-theoretic formalism is a *paradigm* in which calculi are built or defined, i.e., a formalism constitutes the *way* in which calculi are constructed, giving rise to a family resemblance shared by systems within the same formalism.

In the literature, proof-theoretic formalisms and calculi have often been classified into *internal* or *external* (e.g., [20, 25, 104]). There is no formal definition of these properties, and the proof-theoretic community lacks consensus on how each term should be precisely defined. Nevertheless, the literature abounds with informal definitions of internal and external calculi. Typically, internal calculi are described along multiple (sometimes intersecting, sometimes conflicting) lines; e.g., internal calculi have been qualified as proof systems omitting semantic elements from their syntax, or proof systems where every sequent is translatable into an equivalent logical formula. Often times, external calculi are defined as the opposite of internal calculi and, therefore, have also been qualified in multiple ways. It has been claimed that internal calculi are better suited than external calculi for establishing properties such as termination, interpolation, and optimal complexity, while it is purported that external calculi are better suited for counter-model generation and permit easier proofs of cut-admissibility and completeness. We will challenge this divide in this paper.

Due to the diverse number of proof-theoretic formalisms, a large body of work has been dedicated to investigating the relationships between calculi within distinct formalisms by means of *translations*. A translation is a function from one proof formalism into another which extends in a natural way to yield structure-preserving maps between derivations in the concrete calculi instantiating the formalisms. Translations represent a useful tool to formally compare and classify different kinds of proof systems.

The goal of this paper is threefold: (1) we discuss the various sequent-style formalisms that have come to prominence in structural proof theory, (2) we map out the relationships between various proof-theoretic formalisms by means of translations, and (3) we investigate the internal and external distinction in light of these relationships.[1] What we find is that proof-theoretic formalisms sit within a hierarchy that increases in complexity from Gentzen sequents up to labeled sequents, and is based upon the underlying data structure of the sequents used in the system. We will argue that it is 'easier' to translate proofs up this hierarchy than down this hierarchy. Furthermore, we will explain the ambiguities involved in the terms 'internal' and 'external,' and dispel myths about the properties such calculi are purported to possess. To provide a broad account of sequent-based systems we consider a large number of formalisms and systems for a wide array of logics, including modal and tense logics, intuitionistic logic, conditional logics, and bunched implication logic.

This paper is organized as follows: In Section 1, we introduce the various families of logics we consider and their semantics, including modal and tense logics, intuitionistic logic, conditional logics, and bunched logics.

---

[1] We restrict our study to sequent-based formalisms in this paper. Nevertheless, our study retains generality as other types of proof systems, e.g., tableau systems and natural deduction systems, can be transformed into sequent calculi.

In Section 2, we explain the various sequent-based formalisms and specific systems that have been introduced for these logics, giving a broad account of the types of sequent systems that appear in the literature. In the subsequent section (Section 3), we organize these proof-theoretic formalisms and systems into a hierarchy and explain how to traverse this hierarchy by means of translations. Lastly, in Section 4, we discuss the internal and external distinction, and clarify what properties 'internal' and 'external' calculi can be expected to satisfy.

1. Logical preliminaries

To keep the paper self contained and make for a more general approach, we introduce a variety of logics: modal, tense, intuitionistic, conditional, and bunched logics. We will discuss various sequent-style systems for these logics in the sequel.

All logics we consider as propositional, and thus, rely on a set $\mathsf{Prop} := \{p, q, r, ...\}$ of *propositional atoms* (which are occasionally annotated). For convenience, we will make use of the following two (equivalent) languages:

$$A \quad ::= \quad p \mid \bot \mid A \vee A \mid A \wedge A \mid A \to A \tag{1.1}$$

$$A \quad ::= \quad p \mid \overline{p} \mid A \vee A \mid A \wedge A \tag{1.2}$$

When adopting (1.1), we define $\neg A$ as $A \to \bot$ and $\top$ as $\neg\bot$. In (1.2), implication is not a primitive operator and the dual $\bar{\cdot}$ is allowed to occur only on propositional atoms. However, by taking $\overline{A \vee B} := \overline{A} \wedge \overline{B}$ and $\overline{A \wedge B} := \overline{A} \vee \overline{B}$, we can define $A \to B$ as $\overline{A} \vee B$. The propositional language (1.1) is traditionally used to define *two-sided* sequents, while (1.2) is convenient when working with *one-sided* sequents. This distinction will become clear in Section 2; for the moment, observe that the two formulations are equivalent, as the connectives are interdefinable. For the logics based on classical propositional language, we shall sometimes use (1.1) and sometimes (1.2), depending on the corresponding proof system we consider. When introducing intuitionistic logic, we instead need to use (1.1). To avoid any confusion, we shall use $\supset$ to denote intuitionistic implication.

1.1. Modal logics

Even though a study of modalities dates back to Aristotle, modal logic as we know it originates within the work of C.I. Lewis [68], who formulated a notion of strict implication in an attempt to resolve certain paradoxes of material implication. Since then, various modal logics have been defined by expanding a base logic (e.g., classical or intuitionistic logic) with *modalities*, that is, logical operators that qualify the truth of a proposition. *Normal modal logics* extend classical propositional logic through the incorporation of *alethic* modalities, namely, "it is possible that" (denoted by $\Diamond$) and "it is necessary that" (denoted by $\Box$). For an in-depth treatment and presentation of such logics, see Blackburn et al. [6].

For p ranging over Prop, we define the language $\mathcal{L}_M$ of modal logics by adding the $\Box$ modality to (1.1) above:[2]

$$A ::= p \mid \bot \mid A \vee A \mid A \vee A \mid A \rightarrow A \mid \Box A$$

We then set $\Diamond A := \neg\Box\neg A$. Modal formulae are interpreted over *modal Kripke models*. We define such models below, and afterward, define how formulae are interpreted over them.

DEFINITION 1.1 (Modal Kripke Model). A *Kripke frame* is defined to be an ordered pair $F := (W, R)$ such that W is a non-empty set of points, called *worlds*, and $R \subseteq W \times W$ is the *accessibility relation*. A *modal Kripke model* is defined to be a tuple $M = (F, V)$ such that F is a Kripke frame and $V : \mathsf{Prop} \rightarrow 2^W$ is a *valuation function* mapping propositions to sets of worlds.

DEFINITION 1.2 (Semantic Clauses). Let $M = (W, R, V)$ be a modal Kripke model. We define a forcing relation $\models$ such that

- $M, w \models p$ *iff* $w \in V(p)$;
- $M, w \not\models \bot$;
- $M, w \models A \vee B$ *iff* $M, w \models A$ or $M, w \models B$;

[2]This choice is functional to the choice of the proof systems we will introduce in Section 2; the language could have been defined by adding $\Box$ and $\Diamond$ to (1.2) instead.

Name	Frame Property	Modal Axiom
Reflexivity	$\forall w wRw$	$\Box A \to A$
Symmetry	$\forall w, u(wRu \to uRw)$	$A \to \Box\neg\Box\neg A$
Transitivity	$\forall w, v, u(wRv \wedge vRu \to wRu)$	$\Box A \to \Box\Box A$
Euclideanity	$\forall w, v, u(wRv \wedge wRu \to vRu)$	$\neg\Box A \to \Box\neg\Box A$

Figure 1: Frame properties and corresponding axioms.

- $M, w \vDash A \wedge B$ *iff* $M, w \vDash A$ and $M, w \vDash B$;
- $M, w \vDash A \to B$ *iff* if $M, w \vDash A$, then $M, w \vDash B$;
- $M, w \vDash \Box A$ *iff* for every $u \in W$, if wRu, then $M, u \vDash A$;
- $M \vDash A$ *iff* for every $w \in W$, $M, w \vDash A$.

We define a formula $A \in \mathcal{L}_M$ to be $\mathcal{L}_M$*-valid iff* for all modal Kripke models M, $M \vDash A$. We define the minimal normal modal logic K to be the set of $\mathcal{L}_M$-valid formulae.

The truth condition for $\Diamond$ formulae, which is not included in the definition above, is the following: $M, w \vDash \Diamond A$ *iff* there exists $u \in W$ such that $M, u \vDash A$. As is well-known in the domain of modal logics, certain formulae are valid on a class of modal Kripke frames if and only if the accessibility relation of those frames satisfies a certain property. This discovery led to the formulation of *correspondence theory* [6], which investigates relationships between modal axioms and the properties possessed by modal Kripke frames. In Figure 1, we display some popular and well-studied correspondences, and define the two prominent modal logics $\mathsf{S4}$ and $\mathsf{S5}$ accordingly:

Definition 1.3 ($\mathsf{S4}$ and $\mathsf{S5}$). The modal logic $\mathsf{S4}$ is defined to be the set of $\mathcal{L}_M$-valid formula over modal Kripke frames whose relation is reflexive and transitive. The modal logic $\mathsf{S5}$ is defined to be the set of $\mathcal{L}_M$-valid formula over modal Kripke frames whose relation is reflexive and Euclidean.

1.2. Tense logics

Tense logics were invented by Prior in the 1950s [102], and are types of normal modal logics that not only include the $\Diamond$ and $\Box$ modalities, but the *converse modalities* $\blacklozenge$ and $\blacksquare$. These modalities are interpreted in a temporal manner, that is, $\Diamond$ is read as "in some future moment," $\Box$ is read as "in every future moment," $\blacklozenge$ is read as "in some past moment," and $\blacksquare$ is read as "in every past moment." In this paper, we consider the *minimal tense logic* Kt [6], whose language $\mathcal{L}_T$ is defined by adding tense modalities to (1.2):

$$A ::= p \mid \overline{p} \mid A \vee A \mid A \wedge A \mid \langle ? \rangle A \mid [?]A$$

where p ranges over Prop, $\langle ? \rangle \in \{\Diamond, \blacklozenge\}$, and $[?] \in \{\Box, \blacksquare\}$. Formulae from $\mathcal{L}_T$ are in negation normal form as this will simplify the sequent systems we consider later on. Note that negation $\neg$ and implication $\rightarrow$ can be defined as usual; see e.g. [20]. Like formulae in $\mathcal{L}_M$, we interpret formulae from $\mathcal{L}_T$ over modal Kripke models (Definition 1.1).

Definition 1.4 (Semantic Clauses). Let $M = (W, R, V)$ be a modal Kripke model. We define the forcing relation $\Vdash$ as follows, where the clauses for $\wedge$ and $\vee$ are as in Definition 1.2:

- $M, w \Vdash p$ *iff* $w \in V(p)$;
- $M, w \Vdash \overline{p}$ *iff* $w \notin V(p)$;
- $M, w \Vdash \Diamond A$ *iff* there exists a $u \in W$ such that wRu and $M, u \Vdash A$;
- $M, w \Vdash \blacklozenge A$ *iff* there exists a $u \in W$ such that uRw and $M, u \Vdash A$;
- $M, w \Vdash \Box A$ *iff* for every $u \in W$, if wRu, then $M, u \Vdash A$;
- $M, w \Vdash \blacksquare A$ *iff* for every $u \in W$, if uRw, then $M, u \Vdash A$;
- $M \Vdash A$ *iff* for every $w \in W$, $M, w \Vdash A$.

We define a formula $A \in \mathcal{L}_T$ to be $\mathcal{L}_T$*-valid iff* for all modal Kripke models M, $M \Vdash A$. We define the minimal tense logic Kt to be the set of $\mathcal{L}_T$-valid formulae.

1.3. Conditional Logics

Conditional logics formalize hypothetical statements that cannot be faithfully represented using material implication and/or the modal operator $\Box$. Examples of such sentences are *counterfactual conditionals*, e.g., “if A were the case, then B would be the case,” and non-monotonic statements, such as “Normally, if A then B.” To represent counterfactuals and non-monotonic sentences, conditional logics introduce in a classical propositional language a binary modal operator, the *conditional*, which we denote by $A > B$. Although the family of conditional logics contains over 50 systems, we concentrate on the conditional logic V, which is the basic logic of counterfactual reasoning as introduced by D. Lewis [69]. We choose to focus our attention on this conditional logic as its proof-theoretical treatment, while being simpler than for other systems, illustrates the methods needed to capture conditionals.

In Lewis’s account, the conditional operator is defined in terms of another operator, referred to as *comparative plausibility* and denoted $\preccurlyeq$. The formula $A \preccurlyeq B$ states that “A is at least as plausible as B.” The conditional operator $A > B$ may then be defined as $(\bot \preccurlyeq A) \lor \neg((A \land \neg B) \preccurlyeq (A \land B))$, meaning that “either A is impossible or $A \land \neg B$ is less plausible than $A \land B$.” This definition can be simplified by replacing $A \land B$ by A in the second disjunct, yielding: $A > B := (\bot \preccurlyeq A) \lor \neg((A \land \neg B) \preccurlyeq A)$.

Conversely, the comparative plausibility $\preccurlyeq$ can be defined in terms of the conditional operator $>$. Our full language is defined by adding $\preccurlyeq$ to the language (1.1) above, for p ranging over Prop:

$$A ::= p \mid \bot \mid A \lor A \mid A \land A \mid A \to A \mid A \preccurlyeq A$$

From a semantic point of view, logic V is characterized by special kinds of neighborhood models, introduced by Lewis and called *sphere models*. In this semantics, each world is assigned a *system of spheres*, i.e., a set of nested neighborhoods. The intuition is that spheres represent degrees of plausibility, so that worlds in smaller/innermost spheres are considered more plausible than worlds contained solely in larger/outermost spheres.

Definition 1.5 (Sphere Model). A sphere model $M = (W, S, V)$ is a triple such that W is a non-empty set of worlds, $S : W \to 2^{2^W}$, $V : \mathsf{Prop} \to 2^W$ is a valuation function, and the following conditions are satisfied, for every $w \in W$:

- *Non-emptiness*: For every $\alpha \in S(w)$, $\alpha \neq \emptyset$;
- *Nesting*: For every $\alpha, \beta \in S(w)$, either $\alpha \subseteq \beta$ or $\beta \subseteq \alpha$.

The elements of a system of spheres $S(w)$ are called *spheres* and we use $\alpha, \beta, \ldots$ to denote them.

Definition 1.6 (Semantic Clauses). Given a sphere model $M = (W, S, V)$, the forcing relation $\Vdash$ is defined by adding to the forcing relation defined in Definition 1.2 the following clause for $\preccurlyeq$:

- $M, w \Vdash A \preccurlyeq B$ *iff* for all $\alpha \in S(w)$, if there exists $u \in \alpha$ such that $M, u \Vdash B$, then there exists $v \in \alpha$ such that $M, v \Vdash A$.

We define a formula A to be *valid iff* for any sphere model M, $M \Vdash A$, and we define the conditional logic V to be the set of all valid formulae over the class of sphere models.

Given a sphere model $M = (W, S, V)$, we find it useful to introduce the following notation for a sphere $\alpha \in S(w)$:

- $\alpha \Vdash^{\exists} A$ *iff* there is $w \in \alpha$ such that $M, w \Vdash A$;
- $\alpha \Vdash^{\forall} A$ *iff* for all $w \in \alpha$, it holds that $M, w \Vdash A$.

With this notation, the semantic clause for the comparative plausibility operator becomes the following:

- $M, w \Vdash A \preccurlyeq B$ *iff* for all $\alpha \in S(w)$, if $\alpha \Vdash^{\exists} B$, then $\alpha \Vdash^{\exists} A$.

For completeness, we report the truth condition of the conditional operator, which is defined in our language:

- $M, w \Vdash A > B$ *iff* either for all $\alpha \in S(w)$, $\alpha \nVdash^{\exists} A$, or there is an $\alpha \in S(w)$ such that $\alpha \Vdash^{\exists} A$ and $\alpha \Vdash^{\forall} A \to B$.

By imposing additional properties on sphere models, we obtain Lewis's family of conditional logics. Later on, we will define calculi for conditional logics that incorporate inference rules for the comparative plausibility operator.

1.4. Intuitionistic logic

Intuitionistic logic aims to capture the notion of constructive proof, something which classical logic fails to do [49]. For this reason intuitionistic logic does not contain familiar classical axioms such as the law of the excluded middle ($p \vee \neg p$) and double negation elimination ($\neg\neg p \supset p$). The language of intuitionistic logic $\mathcal{L}_I$ is just the language of classical logic (1.1), where classical implication is replaced by intuitionistic implication $\supset$:

$$A ::= p \mid \bot \mid A \vee A \mid A \wedge A \mid A \supset A$$

where p ranges over Prop. As usual, we define $\neg A := A \supset \bot$. Contrary to the classical case, the connectives $\wedge$ and $\vee$ are not inter-definable. Intuitionistic formulae are interpreted over *intuitionistic Kripke models*.

Definition 1.7 (Intuitionistic Kripke Model). An *intuitionistic Kripke frame* is defined to be an ordered pair $F := (W, \leq)$ such that W is a non-empty set of points, called *worlds*, and the *accessibility relation* $\leq \subseteq W \times W$ is reflexive and transitive. A *intuitionistic Kripke model* is defined to be a tuple $M = (F, V)$ such that F is an intuitionistic Kripke frame and $V : \mathsf{Prop} \to 2^W$ is a *valuation function* satisfying the *persistence* condition, that is, if $w \in V(p)$ and $w \leq u$, then $u \in V(p)$.

Definition 1.8 (Semantic Clauses). Given an intuitionistic Kripke model $M = (W, \leq, V)$, we define a forcing relation $\Vdash$ for propositional atoms, $\bot$, $\vee$, and $\wedge$ as in Definition 1.2, but replace the clause for $\to$ with the following:

- $M, w \Vdash A \supset B$ *iff* for all $u \in W$, if $w \leq u$ and $M, u \Vdash A$, then $M, u \Vdash B$;

We define a formula $A \in \mathcal{L}_I$ to be *intuitionistically-valid iff* for all intuitionistic Kripke models M, $M \Vdash A$. We define *intuitionistic logic* IL to be the set of intuitionistically-valid formulae.

A basic fact of intuitionistic logic is that it forms a proper subset of classical logic. Conversely, via the double negation translation, classical logic can be embedded into IL [14]. Moreover, there is also a natural embedding of (axiomatic extensions of) intuitionistic logic (called *intermediate logics*) into (axiomatic extensions of) the modal logic S4 [57].

1.5. Bunched logics

Bunched logics are substructural logics[3] arising from mixing different kinds of connectives associated with a resource aware interpretation. In this paper we focus on the logic of bunched implications (BI) [95, 103], which combines propositional intuitionistic logic with intuitionistic multiplicative linear logic. More formally, the set of formulae of BI, denoted Fm, is given by the following grammar in BNF:

$$A ::= p \mid \underbrace{\top_{\mathfrak{m}} \mid A * A \mid A \mathrel{-\!\!*} A}_{\text{multiplicatives}} \mid \underbrace{\top_{\mathfrak{a}} \mid \bot \mid A \land A \mid A \lor A \mid A \supset A}_{\text{additives}}$$

where p ranges over Prop.

BI admits various forcing semantics [103], called *resource semantics*, which use more elaborate models than those used for intuitionistic logic or modal logics. The most intuitive and widespread resource semantics for BI is the monoid based Kripke semantics that arises from the definition of a (multiplicative) resource composition $\otimes$ on worlds, viewed as resources. The monoid based Kripke semantics can be generalized to a relational semantics [35] replacing both the accessibility $\sqsubseteq$ and the monoidal composition $\otimes$ with a ternary relation R on worlds à la Routley-Meyer (thus reading $w \otimes w' \sqsubseteq u$ as a particular case of $Rww'u$).

The standard monoid based Kripke semantics [35] requires only one resource composition reflecting the properties of the multiplicative connectives. The specifics of the additive connectives are implicitly reflected in their forcing clauses using the properties of the accessibility relation. However, in this paper, we follow [34] and use a monoid based Kripke semantics

[3]Logics that include connectives for which at least one of the usual structural rules (weakening, contraction, exchange, associativity) does not hold.

in which we add a second (additive) resource composition $\oplus$ that explicitly reflects the syntactic behaviour of $\wedge$ into the semantics.

DEFINITION 1.9 (Resource Monoid). A *resource monoid (RM)* is a structure $M = (M, \otimes, 1, \oplus, 0, \infty, \sqsubseteq)$ where $(M, \otimes, 1)$, $(M, \oplus, 0)$ are commutative monoids and $\sqsubseteq$ is a preordering relation on M such that:

- for all $w \in M$, $w \sqsubseteq \infty$ and $\infty \sqsubseteq \infty \otimes w$;
- for all $w, u \in M$, $w \sqsubseteq w \oplus u$ and $w \oplus w \sqsubseteq w$;
- if $w \sqsubseteq u$ and $w' \sqsubseteq u'$, then $w \otimes w' \sqsubseteq u \otimes u'$ and $w \oplus w' \sqsubseteq u \oplus u'$.

Let us remark that the conditions of Definition 1.9 imply that ∞ and 0 respectively are greatest and least elements and that $\oplus$ is idempotent.

DEFINITION 1.10 (Resource Interpretation). Given a resource monoid M, a *resource interpretation (RI) for* M, is a function $[\,-\,] : \mathsf{Fm} \longrightarrow 2^M$ satisfying $\forall\, p \in \mathsf{Prop}$, $\infty \in [\,p\,]$ and $\forall\, w, u \in M$, if $w \in [\,p\,]$ and $w \sqsubseteq u$, then $u \in [\,p\,]$.

DEFINITION 1.11 (Kripke Resource Model). A *Kripke resource model (KRM)* is a structure $\mathcal{K} = (M, \models, [\,-\,])$ where M is a resource monoid, $[\,-\,]$ is a resource interpretation and $\models$ is a forcing relation such that:

- $M, w \models p$ *iff* $w \in [\,p\,]$;
- $M, w \models \bot$ *iff* $\infty \sqsubseteq w$; $M, w \models \top_{\mathsf{a}}$ *iff* $0 \sqsubseteq w$; $M, w \models \top_{\mathsf{m}}$ *iff* $1 \sqsubseteq w$;
- $M, w \models A * B$ *iff* for some u, u' in M, $u \otimes u' \sqsubseteq w$, $M, u \models A$ and $M, u' \models B$;
- $M, w \models A \wedge B$ *iff* for some u, u' in M, $u \oplus u' \sqsubseteq w$, $M, u \models A$ and $M, u' \models B$;
- $M, w \models A \mathrel{-\!\!*} B$ *iff* for all u, u' in M such that $M, u \models A$ and $w \otimes u \sqsubseteq u'$, $M, u' \models B$;
- $M, w \models A \supset B$ *iff* for all u, u' in M such that $M, u \models A$ and $w \oplus u \sqsubseteq u'$, $M, u' \models B$;
- $M, w \models A \vee B$ *iff* $M, w \models A$ or $M, w \models B$.

A formula A is valid in the Kripke resource semantics *iff* $M, 1 \vDash A$ in all Kripke resource models.

Let us call dmKRS the Kripke resource semantics based on double monoids as defined in this section (and first introduced in [34]) and call smKRS the standard one based on single monoids (as defined in [35]). The smKRS is recovered from the dmKRS by erasing all references to $\oplus$ in Definition 1.9 and replacing the forcing clauses for the additive connectives in Definition 1.11 with the following ones:

- $M, w \vDash \top_{\mathfrak{a}}$ *iff* always;
- $M, w \vDash A \wedge B$ *iff* $M, w \vDash A$ and $M, w \vDash B$;
- $M, w \vDash A \supset B$ *iff* for all u in M such that $w \sqsubseteq u$, if $M, u \vDash A$ then $M, u \vDash B$.

The forcing clauses for the additive connectives might seem more natural in the smKRS as they convey their intuitive interpretation in terms of resource sharing (see [103] for details). Indeed, it is immediately seen that $A \wedge B$ is about A and B sharing the same resource (namely, the resource w in $M, w \vDash A \wedge B$). Let us remark that the interpretation of the multiplicative connectives in terms of resource separation ($A * B$ holds for resource w if it can be split into two resources u and u', one satisfying A and the other satisfying B) remains the same in both semantics. Let us also mention that the interpretation of BI formulae in terms of resource sharing and separation is one of the key differences with Linear Logic [39] and its interpretation of formulae in terms of resource accounting (consumption and production).

Although (arguably) less intuitive, the dmKRS clearly makes the presentation of the semantics more uniform as the differences between the additive and multiplicative connectives are captured at the level of the algebraic properties that the corresponding monoidal operators should satisfy (e.g., idempotence for $\oplus$ but not for $\otimes$) and not at the level of the forcing clauses which can therefore be formulated in a similar way. As we shall see later in Section 2.6 it also makes the dmKRS more in tune with the bunched sequent calculus of BI in which the differences between the additive and

System Type	**Data Structure of Sequent**
Labeled Sequents	Graphs of Gentzen Sequents
Display Sequents	(Pairs of)(Poly-)Tree(s) of Gentzen Sequents
Nested, Tree-hypersequents, & Bunched Sequents	Trees of Gentzen Sequents
2-Sequents & Linear Nested Sequents	Lines of Gentzen Sequents
Hypersequents	(Multi-)Set of Gentzen Sequents
Gentzen Sequents	(Pairs of)(Multi-)Set(s)

Figure 2: Common sequent formalisms and their data structure.

multiplicative connectives are handled at the level of the structural rules that the connectives should satisfy and not at the level of the logical rules (which share a similar form).

2. An overview of the proof-theoretic jungle

In this section, we give a broad overview of the various sequent-based formalisms that have come to prominence as generalizations of Gentzen's sequent formalism [37, 38]. Each formalism enriches the data structure employed in Gentzen sequents. Figure 2 summarizes the formalisms we will consider, and the data structure used in the sequents of the formalism. These formalisms form a hierarchy, starting from Gentzen sequents at the bottom and increasing in complexity up to labeled sequents at the top.

2.1. Gentzen system: Classical logic

Gentzen [37, 38] introduced the sequent formalism to proof theory by defining sequent calculi for classical and intuitionistic logic. We begin by recalling the sequent calculus for classical logic. A sequent is an object of the form $\Gamma \Rightarrow \Delta$ where Γ and Δ are (possibly empty) *multisets* of formulae

$$\frac{}{\Gamma, p \Rightarrow p, \Delta}\ (id) \qquad \frac{}{\bot, \Gamma \Rightarrow \Delta}\ (\bot_l)$$

$$\frac{\Gamma, A, B \Rightarrow \Delta}{\Gamma, A \wedge B \Rightarrow \Delta}\ (\wedge_l) \qquad \frac{\Gamma \Rightarrow A, \Delta \qquad \Gamma \Rightarrow B, \Delta}{\Gamma \Rightarrow A \wedge B, \Delta}\ (\wedge_r)$$

$$\frac{\Gamma, A \Rightarrow \Delta \qquad \Gamma, B \Rightarrow \Delta}{\Gamma, A \vee B \Rightarrow \Delta}\ (\vee_l) \qquad \frac{\Gamma \Rightarrow A, B, \Delta}{\Gamma \Rightarrow A \vee B, \Delta}\ (\vee_r)$$

$$\frac{\Gamma \Rightarrow A, \Delta \qquad \Gamma, B \Rightarrow \Delta}{\Gamma, A \rightarrow B \Rightarrow \Delta}\ (\rightarrow_l) \qquad \frac{\Gamma, A \Rightarrow B, \Delta}{\Gamma \Rightarrow A \rightarrow B, \Delta}\ (\rightarrow_r)$$

Figure 3: Initial sequents and logical rules of $\mathsf{S}(\mathsf{CP})$.

from language (1.1). We call Γ the *antecedent* and Δ the *consequent* of the sequent. Each sequent $\Gamma \Rightarrow \Delta$ with $\Gamma = A_1, \ldots, A_m$ and $\Delta = B_1, \ldots, B_n$ can be interpreted as a formula of the following form:

$$\tau(\Gamma \Rightarrow \Delta) := (A_1 \wedge \cdots \wedge A_m) \rightarrow (B_1 \vee \cdots \vee B_n)$$

When $m = 0$, the empty conjunction is interpreted as $\top$, and when $n = 0$, the empty disjunction is interpreted as $\bot$.

A sequent calculus contains axioms, also called *initial sequents*, and rules that let one derive sequents from sequents. The latter are divided into *logical* rules that introduce complex formulae in either the antecedent or consequent of a sequent, and *structural* rules which modify the structure of the antecedent/consequent, without changing the formulae themselves. Figure 3 contains the axioms (id) and $(\bot_l)$, and the logical rules for the sequent calculus $\mathsf{S}(\mathsf{CP})$ for classical propositional logic.

We define a *derivation* $\mathcal{D}$ of a sequent S to be a (potentially infinite) tree whose nodes are sequents satisfying the following conditions: (1) the root of $\mathcal{D}$ is the sequent S, and (2) every parent node is the instance of the conclusion of a rule with its children the corresponding premises. We say that a derivation $\mathcal{D}$ of S is a *proof* of S if all the leaves of $\mathcal{D}$ are axioms. We say that a sequent S is *provable iff* it has a proof. The *height* of a derivation is equal to the number of sequents along a maximal path from the root to a leaf. In a rule, we define the *principal formulae* to be

$$\frac{\Gamma \Rightarrow \Delta}{\Gamma, A \Rightarrow \Delta}\ (wk_l) \qquad \frac{\Gamma \Rightarrow \Delta}{\Gamma \Rightarrow \Delta, A}\ (wk_r) \qquad \frac{\Gamma, A, A \Rightarrow \Delta}{\Gamma, A \Rightarrow \Delta}\ (cr_l)$$

$$\frac{\Gamma \Rightarrow A, A, \Delta}{\Gamma \Rightarrow A, \Delta}\ (cr_r) \qquad \frac{\Gamma \Rightarrow A, \Delta \qquad \Gamma, A \Rightarrow \Delta}{\Gamma \Rightarrow \Delta}\ (cut)$$

Figure 4: Structural rules for the sequent calculus $\mathsf{S}(\mathsf{CP})$.

those explicitly introduced in the conclusion, and the *auxiliary formulae* to be those explicitly used in the premise(s) to derive the conclusion. For example, $A \to B$ is the principal formula in $(\to_l)$ and A and B are the auxiliary formulae. By *proof-search* we mean an algorithm that builds a derivation by applying inference rules bottom-up.

The structural rules for $\mathsf{S}(\mathsf{CP})$ are displayed in Figure 4. The *weakening* rules (wk_l) and (wk_r) introduce formulae into the antecedent and consequent of a sequent, while the *contraction* rules (cr_l) and (cr_r) remove additional copies of formulae. The (cut) rule can be seen as a generalization of modus ponens and has a special status, namely, it encodes the transitivity of deduction. Observe that the (cut) rule is not analytic, as the premises contain an arbitrary formula that disappears in the conclusion. It is important to notice that all structural rules, and in particular the (cut) rule, are *admissible* in the calculus $\mathsf{S}(\mathsf{CP})$, meaning that if instances of the premises are provable, then so is the corresponding conclusion. By this fact, structural rules are recognized to be unnecessary for completeness.

The logical rules for negation, which we have chosen not to include as primitive rules, are also admissible in $\mathsf{S}(\mathsf{CP})$, and we will sometimes use them in derivations:

$$\frac{\Gamma \Rightarrow A, \Delta}{\Gamma, \neg A \Rightarrow \Delta}\ (\neg_l) \qquad \frac{\Gamma, A \Rightarrow \Delta}{\Gamma \Rightarrow \neg A, \Delta}\ (\neg_r)$$

The Gentzen calculus $\mathsf{S}(\mathsf{CP})$ is a 'two-sided' proof system, meaning that sequents are composed of an antecedent and consequent, and consequently the proof system is constituted by left and right logical rules. By taking the language of classical propositional logic to be (1.2), it is possible to define a more compact ‘one-sided’ version of $\mathsf{S(CP)}$. In this case, a sequent is just a multiset of formulae $\Delta = B_1 \vee \cdots \vee B_n$, and it is interpreted as the formula $\tau(\Delta) := B_1 \vee \cdots \vee B_n$. The rules of the one-sided calculus are displayed in Figure 5. It is easy to see that the one-sided and the two-sided versions of $\mathsf{S(CP)}$ are equivalent.

The Gentzen calculus $\mathsf{S(CP)}$ has some important properties, discussed below, that set it as an ‘ideal’ proof system. The following terminology will also be applied to the other kinds of sequent-style systems we consider later on.

Analyticity: the premises of each rule only contain subformulae of the conclusion. Thus, if we do not consider multiple occurrences of the same formulae (i.e., we consider a sequent as a pair of sets), given a proof of a sequent S, there are only finitely many different sequents that can occur in $\mathcal{P}$. This follow from the admissibility of the cut rule (i.e., *cut-elimination*), which means that every proof containing applications of (*cut*) can be transformed into a cut-free proof of the same conclusion [37, 38].

Termination: the premises of each rule are less complex than the conclusion. This property holds in $\mathsf{S(CP)}$ (without structural rules) since the auxiliary formulae are always less complex than the principal formulae. This property together with analyticity ensures that the process of building a derivation (bottom-up) always terminates, that is, every branch of a derivation $\mathcal{D}$ terminates at an axiom or an unprovable sequent (usually containing only atoms).

Invertibility: if any instance of the conclusion of a rule is provable, then its corresponding premises are provable. By this property, the order of bottom-up applications of rules during proof-search does not matter: either we obtain a proof of the root sequent, or we get a (finite) derivation containing an unprovable sequent as a leaf. When this property is present, backtracking (i.e., searching for alternative proofs) is unnecessary during proof-search.

Counter-model generation: if proof-search yields a derivation $\mathcal{D}$ that is not a proof of the conclusion, then there exists an unprovable sequent as a leaf which can be used to define a counter-model of the conclusion. In the case

$$\frac{}{\Delta, p, \overline{p}}\ (id) \qquad \frac{A, \Delta \qquad B, \Delta}{A \wedge B, \Delta}\ (\wedge_r) \qquad \frac{A, B, \Delta}{A \vee B, \Delta}\ (\vee_r)$$

Figure 5: One-sided rules of $\mathsf{S(CP)}$.

of $\mathsf{S(CP)}$, if $\Gamma \Rightarrow \Delta$ is such a leaf in a derivation $\mathcal{D}$, then $\Gamma \cap \Delta = \emptyset$, and one can define a propositional evaluation '$V(p) = \mathbf{t}$ *iff* $p \in \Gamma$' that falsifies the conclusion of $\mathcal{D}$.

Complexity-optimal: the proof system admits a (relatively straightforward) proof-search algorithm that decides the (in)validity of formulae in the complexity of the logic.

The calculus $\mathsf{S(CP)}$ satisfies the above four properties; as a consequence, the calculus provides a decision procedure for classical propositional logic. Proof-search is carried out by building *just one* derivation that will either be a proof, or from which a counter-model of the conclusion can be extracted. Furthermore, the decision procedure based on the calculus has an optimal complexity (CoNP).

2.2. Gentzen system: Intuitionistic logic

Gentzen's sequent systems are flexible enough to capture other logics. For example, intuitionistic logic can be provided a sequent calculus by making simple modifications to $\mathsf{S(CP)}$. The calculus $\mathsf{S(IL)}$ for intuitionistic logic is obtained by replacing the $(\rightarrow_l)$ and $(\rightarrow_r)$ rules in $\mathsf{S(CP)}$ with the $(\supset_l)$ and $(\supset_r)$ rules shown in Figure 6. Originally, Gentzen obtained a sequent calculus for intuitionistic logic by imposing a restriction on the sequent calculus $\mathsf{S(CP)}$ for classical logic, namely, only sequents with at most one formula in the consequent (i.e., sequents $\Gamma \Rightarrow \Delta$ such that $|\Delta| \leq 1$) could be used in derivations. However, Gentzen's restriction invalidates certain admissibility and invertibility properties, which can be regained allowing multiple formulae to occur in the conclusion. The calculus $\mathsf{S(IL)}$, due to Maehara [84], is a variant of Gentzen's sequent calculus for intuitionistic logic that has all structural rules, including (cut), admissible.

$$\frac{\Gamma, A \supset B, B \Rightarrow \Delta \qquad \Gamma, A \supset B \Rightarrow A, \Delta}{\Gamma, A \supset B \Rightarrow \Delta}\,(\supset_l) \qquad \frac{\Gamma, A \Rightarrow B}{\Gamma \Rightarrow A \supset B, \Delta}\,(\supset_r)$$

Figure 6: Intuitionistic implication rules for $\mathsf{S}(\mathsf{IL})$.

The calculus $\mathsf{S}(\mathsf{IL})$ is analytic, though not terminating as the premises of a rule may be as complex as the conclusion as witnessed by the premises of $(\supset_l)$. Additionally, the $(\supset_r)$ rule is not invertible, which is an impediment to proof-search. In particular, if proof-search constructs a derivation that is not a proof, then a proof may still exist and the constructed derivation may not provide a counter-model of the conclusion. Due to analyticity a decision procedure can still be obtained; however, the procedure will also require loop checking which diminishes its efficiency. For an overview of various proof systems for intuitionistic logic and associated decision procedures, see [29].

Although Gentzen systems have been provided for many logics, the formalism is still not general enough to yield cut-free systems for many logics of interest (e.g., $\mathsf{S5}$ and bi-intuitionistic logics [66, 12]). This motivates the search for more expressive formalisms that enrich Gentzen sequents to recapture analyticity and other properties.

2.3. Beyond Gentzen's formalism

In the previous section we highlighted some desirable properties of proof systems, which Gentzen sequent systems often times satisfy. However, we are here interested in the definition of formalisms satisfying desirable properties for large families of logics. Thus, we identify five desiderata for proof-theoretic formalisms:[4]

(1) *Generality*: the formalism covers a sizable class of logics with proof systems sharing desirable properties;

[4]For discussions of other desiderata for proof systems and formalisms, see [113, 2].

(2) *Uniformity*: the formalism need not be enriched to obtain a system for a logic within a given class;

(3) *Modularity*: a system for one logic within the considered class can be transformed into a system for another, with properties preserved, by adding/deleting rules or modifying the functionality of rules;

(4) *Constructibility*: a method is known for constructing a calculus for a given logic in the considered class;

(5) *Syntactic Parsimony*: the data structures employed are as simple as required by the logic or purpose of the proof systems.

When the desiderata (1)–(4) are satisfied, a proof formalism is expected to generate large classes of proof calculi for logics without requiring substantial work on the side of the logician. According to requirement (5), a formalism should employ sequents that are as simple as possible, in order to maintain their interpretation as formulae of the language and simplify derivations.

It is not to be taken for granted that a single proof formalism can fulfill all of the above requirements, which justifies the study of alternative proof systems and formalisms with different properties and applications. For instance, although Gentzen's sequent formalism satisfies syntactic parsimony to a high degree, the formalism lacks uniformity and modularity, since simple modifications to a calculus can nullify key properties such as analyticity. Similarly, although nested sequents employ trees of Gentzen sequents, they are better suited for counter-model extraction than Gentzen sequent calculi, and so, if we aim to use our systems to extract counter-models of formulae, then it is sensible to trade the simple structure of Gentzen sequents for nested sequents.

In the next subsections we will present a number of formalisms that are less parsimonious that Gentzen sequents, but are more satisfactory than Gentzen sequents regarding requirements (1)–(4).

$$\frac{}{G \mid \Gamma, p \Rightarrow p, \Delta}\ (id) \qquad \frac{}{G \mid \Gamma, \bot \Rightarrow \Delta}\ (\bot_l) \qquad \frac{G \mid \Gamma \Rightarrow \Delta \mid \Rightarrow A}{G \mid \Gamma \Rightarrow \Box A, \Delta}\ (\Box_r)$$

$$\frac{G \mid \Gamma, \Box A, A \Rightarrow \Delta}{G \mid \Gamma, \Box A \Rightarrow \Delta}\ (\Box_{l1}) \qquad \frac{G \mid \Gamma, \Box A \Rightarrow \Delta \mid \Sigma, A \Rightarrow \Pi}{G \mid \Gamma, \Box A \Rightarrow \Delta \mid \Sigma \Rightarrow \Pi}\ (\Box_{l2})$$

Figure 7: Rules for a hypersequent calculus $\mathsf{H(S5)}$ for $\mathsf{S5}$.

2.4. Hypersequents

Introduced independently by Mints [91], Pottinger [101], and Avron [1], the hypersequent formalism is a simple generalization of Gentzen's sequent formalism. A *hypersequent* is an expression of the form $\Gamma_1 \Rightarrow \Delta_1 \mid \cdots \mid \Gamma_n \Rightarrow \Delta_n$ such that each *component* $\Gamma_i \Rightarrow \Delta_i$ is a Gentzen sequent. That is, a hypersequent is a (multi)set of Gentzen sequents, where each element of the (multi)set is separated by the '$|$' operator. Usually, we interpret the '$|$' operator disjunctively, meaning, hypersequents are interpreted as disjunctions of Gentzen sequents. We use G, H, ... to denote hypersequents.

To demonstrate the hypersequent formalism, we provide an example of a hypersequent calculus $\mathsf{H(S5)}$ for the modal logic $\mathsf{S5}$, which is due to Poggiolesi [97] though adapted to the language we are using for $\mathsf{S5}$.[5] The hypersequent calculus $\mathsf{H(S5)}$ contains the rules shown in Figure 7 together with analogs for the rules $(\vee_l)$, $(\vee_r)$, $(\wedge_l)$, $(\wedge_r)$, $(\rightarrow_l)$, and $(\rightarrow_r)$ from the Gentzen calculus $\mathsf{S(CL)}$. These latter rules perform the same operation as their Gentzen calculus counterparts and are applied to components of hypersequents; for example, the $(\rightarrow_l)$ and $(\rightarrow_r)$ rules are defined as follows:

$$\frac{G \mid \Gamma \Rightarrow A, \Delta \qquad G \mid \Gamma, B \Rightarrow \Delta}{G \mid \Gamma, A \rightarrow B \Rightarrow \Delta}\ (\rightarrow_l) \qquad \frac{G \mid \Gamma, A \Rightarrow B, \Delta}{G \mid \Gamma \Rightarrow A \rightarrow B, \Delta}\ (\rightarrow_r)$$

As with Gentzen calculi, hypersequent calculi may contain axioms, logical rules, and structural rules. For instance, the hypersequent calculus $\mathsf{H(S5)}$ contains the axioms (id) and $(\bot_l)$ and all remaining rules are logical

[5]See [4, 58, 105] for alternative hypersequent systems for the modal logic $\mathsf{S5}$.

rules. Moreover, similar to Gentzen systems, one can often find a selection of structural rules that are admissible in a hypersequent system. Due to the additional structure present in hypersequent calculi, structural rules can be classified into a wider variety of types. That is to say, the hypersequent structure makes it possible to define new external structural rules that allow for the exchange of information between different components of a hypersequent. This increases the expressive power of hypersequent calculi compared to ordinary Gentzen systems.

As an example, for the hypersequent calculus $\mathsf{H}(\mathsf{S5})$, one can define both *internal* and *external* structural rules. Internal structural rules strictly affect components; for instance, the following internal weakening (iw) and internal contraction (ic) rules apply weakenings and contractions only within components of hypersequents:

$$\frac{G \mid \Gamma \Rightarrow \Delta}{G \mid \Gamma, \Sigma \Rightarrow \Pi, \Delta}\ (iw) \qquad \frac{G \mid \Gamma, \Sigma, \Sigma \Rightarrow \Pi, \Pi, \Delta}{G \mid \Gamma, \Sigma \Rightarrow \Pi, \Delta}\ (ic)$$

On the other hand, external structural rules are more general and affect the overall structure of a hypersequent; for instance, the following external weakening (ew) and external contraction (ec) rules weaken in new components and contract components, respectively:

$$\frac{G}{G \mid \Gamma \Rightarrow \Delta}\ (ew) \qquad \frac{G \mid \Gamma \Rightarrow \Delta \mid \Gamma \Rightarrow \Delta}{G \mid \Gamma \Rightarrow \Delta}\ (ec)$$

We remark that all of the above rules are admissible in $\mathsf{H}(\mathsf{S5})$ as is a hypersequent version of the cut rule [97].

It is well-known that the hypersequent formalism allows for the formulation of cut-free sequent-style systems for logics failing to possess a cut-free Gentzen system. The formalism also supports the algorithmic transformation of large classes of Hilbert axioms and frame properties into cut-free hypersequent calculi for wide classes of logics, including substructural logics [18], intermediate logics [24], and modal logics [62, 65]. Therefore, the hypersequent formalism can be seen to satisfy our five desiderata to a large degree: with only a basic increase in syntactic complexity from that of Gentzen sequents, hypersequent systems with favorable properties (e.g., analyticity) can be algorithmically generated for wide classes of logics. This demonstrates the generality, uniformity, modularity, and constructibility of such systems. Nevertheless, there are logics for which the hypersequent formalism is ill-suited for providing analytic systems (e.g., the tense logic Kt and some modal logics characterized by geometric frame conditions [106]), showing that the generality of the formalism is still limited in scope.

2.5. 2-Sequents and linear nested sequents

The 2-sequent formalism was introduced by Masini [87, 88] as a generalization of Gentzen's sequent formalism whereby an *infinite list* of multisets of formulae implies another infinite list. For instance, an example of a 2-sequent is shown below left and an another example is shown below right:

$$
\begin{array}{ccc}
A, B & & D \\
C & \Rightarrow & E, F \\
 & & G
\end{array}
\quad \rightsquigarrow \quad
\begin{array}{ccc}
A, B & & D \\
C & \Rightarrow & E, F, \Box G
\end{array}
$$

In the 2-sequent above left, the antecedent consists of the list whose first element is the multiset A, B, second element is the singleton C, and where every other element is the empty multiset. By contrast, the consequent consists of a list beginning with the three multisets (1) D, (2) E, F, and (3) G, and where every other element is the empty multiset. The 2-sequent shown above right is derivable from the 2-sequent shown above left by 'shifting' the G formula up one level and introducing a $\Box$ modality, thus demonstrating how modal formulae may be derived in the formalism. Systems built with such sequents have been provided for various logics—e.g., modal logics [87], intuitionistic logic [88] and tense logics [3]—and tend to exhibit desirable proof-theoretic properties such as generalized forms of cut-elimination and the subformula property.

More recently, a refined but equivalent re-formulation of 2-sequents was provided by Lellmann [64]. Rather than employing sequents with infinite lists for antecedents and consequents, the formalism employs *linear nested* *sequents*, which are finite lists of Gentzen sequents. For example, the 2-sequents shown above left and right may be re-written as the linear nested sequents shown below left and right respectively with the '$\mathbin{/\!\!/}$' constructor separating the components (i.e., each Gentzen sequent) in each list:

$$A, B \Rightarrow D \mathbin{/\!\!/} C \Rightarrow E, F \mathbin{/\!\!/} \emptyset \Rightarrow G \qquad A, B \Rightarrow D \mathbin{/\!\!/} C \Rightarrow E, F, \Box G$$

Linear nested sequent systems have been provided for a diverse selection of logics; e.g., modal logics [64], Gödel-Löb provability logic [76], propositional and first-order Gödel-Dummett logic [61, 71], the tense logic Kt [42] and the tense logics with linear time [51, 52]. Moreover, such calculi have been used to write constructive interpolation proofs [61] and decision procedures [42].

As discussed by Lellmann [64], there is a close connection between Gentzen sequent calculi, nested sequent calculi (discussed in Section 2.7 below), and linear nested sequent calculi. In particular, certain linear nested sequent systems have been found to encode *branches* within sequent calculus proofs as well as branches within nested sequents. For example, the standard ($\Box$) rule that occurs in the sequent calculus for the modal logic K (shown below left) corresponds to $|\Gamma|$ many applications of the ($\Box_l$) rule followed by an application of the ($\Box_r$) rule in the linear nested sequent calculus for K (cf. [64]).

$$\frac{\Gamma \Rightarrow A}{\Sigma, \Box\Gamma \Rightarrow \Box A, \Delta}\ (\Box) \quad \rightsquigarrow \quad \cfrac{\cfrac{\Sigma, \Box\Gamma \Rightarrow \Box A, \Delta \mathbin{/\!\!/} \Gamma \Rightarrow A}{\Sigma, \Box\Gamma \Rightarrow \Box A, \Delta \mathbin{/\!\!/} \emptyset \Rightarrow A}\ (\Box_l) \times |\Gamma|}{\Sigma, \Box\Gamma \Rightarrow \Box A, \Delta}\ (\Box_r)$$

Observe that the top linear nested sequent in the inferences shown above right stores the conclusion and premise of the ($\Box$) rule, thus demonstrating how linear nested sequents can encode branches (i.e., sequences of inferences) in sequent calculus proofs.

Due to the fact that linear nested sequents employ a relatively simple data structure, the formalism typically allows for complexity-optimal proof-search algorithms, similar to (depth-first) algorithms written within sequent and nested sequent systems. As such, the linear nested sequent formalism strikes a balance between complexity-optimality on the one hand,

and expressivity on the other, since the formalism allows for many logics to be captured in a cut-free manner while exhibiting desirable invertibility and admissibility properties. Note that if we let the '$/\!\!/$' constructor be commutative, then it can be seen as the hypersequent '$|$' constructor, showing that every hypersequent calculus is technically a linear nested sequent calculus, i.e., the latter formalism generalizes the former [64]. It can be seen that the linear nested sequent formalism satisfies the same desiderata as the hypersequent formalism, though improves upon generality as the formalism is known to capture logics lacking a cut-free hypersequent calculus, e.g., the tense logic Kt [42].

2.6. Bunched sequents

As recalled in Section 2.1, a standard Gentzen sequent is an object of the form $\Gamma \Rightarrow \Delta$ where the contexts Γ and Δ usually are sets or multisets, sometimes (but less often[6]) lists. Those data structures are one dimensional and built from a single context forming operator usually written as a comma or a semi-colon. An interesting extension of Gentzen sequents are *bunched sequents*, which arise when the contexts are built from more than one context forming operators. For example, in [17] two context forming operators, ";" and ',' are used to split the contexts into several zones the formulae of which are handled differently by a focused sequent calculus. Nevertheless, inside a zone, the formulae are arranged as a one dimensional structure (usually a multiset) and the inference rules can be applied to any formula in that shallow structure (thus making the inference rules shallow). Let us remark that bunched structures can also be used to extend the hypersequent framework to bunched hypersequents (forests of sequents) as illustrated in a recent work [23]. However, in the remaining of the section, we shall focus on the most representative witness of a bunched sequent calculus, which is undoubtedly the one given for Bunched Implications logic (BI) in [103]. Let us note that such kind of structured calculi were initially proposed in the field of relevant logics [27, 92].

[6] This is the case for logics lacking *all* structural rules.

$$\frac{}{A \Rightarrow A}\ (\mathrm{id}) \qquad \frac{}{\emptyset_{\mathfrak{m}} \Rightarrow \top_{\mathfrak{m}}}\ (\top_{\mathfrak{m}\,r}) \qquad \frac{}{\emptyset_{\mathfrak{a}} \Rightarrow \top_{\mathfrak{a}}}\ (\top_{\mathfrak{a}\,r}) \qquad \frac{}{\Gamma(\bot) \Rightarrow A}\ (\bot_l)$$

$$\frac{\Gamma(\emptyset_{\mathfrak{m}}) \Rightarrow A}{\Gamma(\top_{\mathfrak{m}}) \Rightarrow A}\ (\top_{\mathfrak{m}\,l}) \qquad \frac{\Gamma(\emptyset_{\mathfrak{a}}) \Rightarrow A}{\Gamma(\top_{\mathfrak{a}}) \Rightarrow A}\ (\top_{\mathfrak{a}\,l}) \qquad \frac{\Gamma(B) \Rightarrow A \quad \Gamma(C) \Rightarrow A}{\Gamma(B \vee C) \Rightarrow A}\ (\vee_l)$$

$$\frac{\Gamma \Rightarrow A_{\mathrm{i} \in \{1,2\}}}{\Gamma \Rightarrow A_1 \vee A_2}\ (\vee_r^{\mathrm{i}}) \qquad \frac{\Delta \Rightarrow B \quad \Gamma(C) \Rightarrow A}{\Gamma(B \mathrel{-\!\!*} C, \Delta) \Rightarrow A}\ (\mathrel{-\!\!*}_l) \qquad \frac{\Gamma, A \Rightarrow B}{\Gamma \Rightarrow A \mathrel{-\!\!*} B}\ (\mathrel{-\!\!*}_r)$$

$$\frac{\Gamma(B, C) \Rightarrow A}{\Gamma(B * C) \Rightarrow A}\ (*_l) \qquad \frac{\Gamma \Rightarrow A \quad \Delta \Rightarrow B}{\Gamma, \Delta \Rightarrow A * B}\ (*_r) \qquad \frac{\Delta \Rightarrow B \quad \Gamma(C) \Rightarrow A}{\Gamma(B \supset C; \Delta) \Rightarrow A}\ (\supset_l)$$

$$\frac{\Gamma ; A \Rightarrow B}{\Gamma \Rightarrow A \supset B}\ (\supset_r) \qquad \frac{\Gamma(B; C) \Rightarrow A}{\Gamma(B \wedge C) \Rightarrow A}\ (\wedge_l) \qquad \frac{\Gamma \Rightarrow A \quad \Delta \Rightarrow B}{\Gamma ; \Delta \Rightarrow A \wedge B}\ (\wedge_r)$$

$$\frac{\Gamma(\Delta_1) \Rightarrow A}{\Gamma(\Delta_1 ; \Delta_2) \Rightarrow A}\ (\mathrm{WK}) \qquad \frac{\Gamma(\Delta ; \Delta) \Rightarrow A}{\Gamma(\Delta) \Rightarrow A}\ (\mathrm{CR}) \qquad \frac{\Gamma \Rightarrow A}{\Delta \Rightarrow A}\ (\equiv)$$

$$\frac{\Delta \Rightarrow B \quad \Gamma(B) \Rightarrow A}{\Gamma(\Delta) \Rightarrow A}\ (\mathrm{CUT})$$

Figure 8: The sequent calculus LBI.

In BI, formulae are arranged as “bunches” which can be viewed as trees whose leaves are labeled with formulae and whose internal nodes are labeled with either “;” or “,”. More formally, bunches are trees given by the following grammar:

$$\Gamma ::= A \mid \emptyset_{\mathsf{a}} \mid \Gamma\,;\Gamma \mid \emptyset_{\mathsf{m}} \mid \Gamma\,,\Gamma$$

notation $\Gamma(\Delta)$ denotes a bunch Γ that contains the bunch Δ as a subtree.

Bunches are considered up to a structural equivalence $\equiv$ given by commutative monoid equations for “;” and “,” with units $\emptyset_{\mathsf{a}}$ and $\emptyset_{\mathsf{m}}$ respectively, together with the substitution congruence for subbunches. From a logical point of view, bunches relate to formulae as follows: let Γ be a bunch, the corresponding formula is $_{\Gamma}$ which is obtained from Γ by replacing each $\emptyset_{\mathsf{m}}$ with $\top_{\mathsf{m}}$, each $\emptyset_{\mathsf{a}}$ with $\top_{\mathsf{a}}$, each “,” with $*$ and each “;” with $\wedge$.

The standard internal calculus for BI is a single conclusion bunched sequent calculus called LBI. In LBI, bunches arise (on the left-hand side only) from the two kinds of implications $\supset$ and $-\!\!*$, that respectively give rise to two distinct context forming operators “;” and “,” as follows:

$$\frac{\Gamma\,; A \Rightarrow B}{\Gamma \Rightarrow A \supset B}(\supset_r) \qquad\qquad \frac{\Gamma\,, A \Rightarrow B}{\Gamma \Rightarrow A -\!\!* B}(-\!\!*_r)$$

From a syntactic point of view, the main distinction between “;” (associated with $\wedge$) and “,” (associated with $*$) is that “;” admits both weakening and contraction while “,” does not.

The LBI sequent calculus, depicted in Figure 8, derives sequents of the form $\Gamma \Rightarrow C$, where Γ is a bunch and C is a formula. A formula C is a *theorem* of LBI iff $\emptyset_{\mathsf{m}} \Rightarrow C$ is provable in LBI. Let us remark that the inference rules of LBI are deep in that they can be applied to formulae anywhere in the tree structure of a bunch and not only at the root. Let us also mention that the CUT rule is admissible in LBI [103] and that the contraction rule CR may duplicate whole bunches and not just formulae. Indeed, as shown in Example 2.1, restricting contraction to single formulae would not allow to prove the end sequent.

Example 2.1. LBI-proof of $((p -\!\!* (q \supset r) \wedge p -\!\!* q) * p) -\!\!* r$.

$$
\begin{prooftree}
\AxiomC{}
\RightLabel{(id)}
\UnaryInfC{$p \Rightarrow p$}
\AxiomC{}
\RightLabel{(id)}
\UnaryInfC{$p \Rightarrow p$}
\AxiomC{}
\RightLabel{(id)}
\UnaryInfC{$q \Rightarrow q$}
\AxiomC{}
\RightLabel{(id)}
\UnaryInfC{$r \,; q \Rightarrow r$}
\RightLabel{$(\supset_l)$}
\BinaryInfC{$q \supset r \,; q \Rightarrow r$}
\RightLabel{$(-\!\!*_l)$}
\BinaryInfC{$q \supset r \,; (p -\!\!* q \,, p) \Rightarrow r$}
\RightLabel{$(-\!\!*_l)$}
\BinaryInfC{$(p -\!\!* (q \supset r) \,, p) \,; (p -\!\!* q \,, p) \Rightarrow r$}
\RightLabel{(WK)}
\UnaryInfC{$(p -\!\!* (q \supset r) \,, p) \,; ((p -\!\!* (q \supset r) \,; p -\!\!* q) \,, p) \Rightarrow r$}
\RightLabel{(WK)}
\UnaryInfC{$((p -\!\!* (q \supset r) \,; p -\!\!* q) \,, p) \,; ((p -\!\!* (q \supset r) \,; p -\!\!* q) \,, p) \Rightarrow r$}
\RightLabel{(CR)}
\UnaryInfC{$(p -\!\!* (q \supset r) \,; p -\!\!* q) \,, p \Rightarrow r$}
\RightLabel{$(\wedge_r)$}
\UnaryInfC{$((p -\!\!* (q \supset r) \wedge p -\!\!* q) \,, p) \Rightarrow r$}
\RightLabel{$(\equiv)$}
\UnaryInfC{$\emptyset_{\mathrm{m}} \,, ((p -\!\!* (q \supset r) \wedge p -\!\!* q) \,, p) \Rightarrow r$}
\RightLabel{$(*_l)$}
\UnaryInfC{$\emptyset_{\mathrm{m}} \,, ((p -\!\!* (q \supset r) \wedge p -\!\!* q) * p) \Rightarrow r$}
\RightLabel{$(-\!\!*_r)$}
\UnaryInfC{$\emptyset_{\mathrm{m}} \Rightarrow ((p -\!\!* (q \supset r) \wedge p -\!\!* q) * p) -\!\!* r$}
\end{prooftree}
$$

2.7. Nested sequents

Nested sequent calculi were originally defined by Kashima for tense logics [55] and Bull for the fragment of PDL without the Kleene star [13].[7] The characteristic feature of such calculi is the use of *trees* of Gentzen sequents in proofs. This additional structure has led to the development of cut-free calculi for various logics not known to possess a cut-free Gentzen sequent calculus. This formalism is general in the sense that sizable classes of logics can be uniformly captured with such systems. For example, cut-free nested sequent calculi have been given for classical modal logics [11, 98], for intuitionistic modal logics [108, 72], for classical tense logics [55, 44], and for first-order non-classical logics [74, 78]. Moreover, the rules of nested sequent calculi are usually invertible, which—as mentioned above—are useful in extracting counter-models from failed proof-search (cf. [109, 77]). We remark that nested sequents have also been referred to as *tree-hypersequents* [98, 99]; however, we will stick to the term *nested*

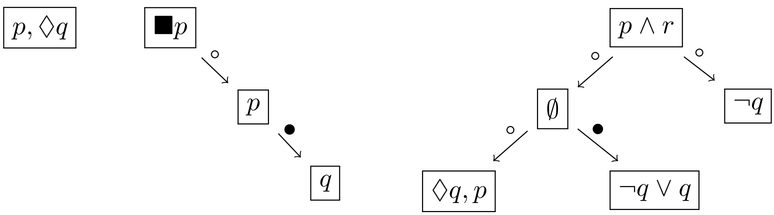
[7] We note that nested sequent calculi can be seen as 'upside-down' versions of prefixed tableaux [31, 32]. Furthermore, Leivant's 1981 paper [63] introduces a calculus for PDL that is structurally equivalent to a nested sequent calculus. Both of these works predate the work of Kashima and Bull.

sequent as it is far more prevalent in the literature and is the original term given by Kashima [55] and Bull [13].

Nested sequents generalize the syntax of one-sided Gentzen sequents via the incorporation of a nesting constructor. For instance, for the tense logic Kt, nested sequents are generated via the following grammar in BNF:

$$\Sigma ::= A \mid \emptyset \mid \Sigma, \Sigma \mid \circ[\Sigma] \mid \bullet[\Sigma]$$

where $A \in \mathcal{L}_M$ and $\emptyset$ is the empty nested sequent. Examples of nested sequents generated in the above syntax include (1) $\Sigma_1 = p, \Diamond q$, (2) $\Sigma_2 = \blacksquare p, \circ[p, \bullet[q]]$, and (3) $\Sigma_3 = p \wedge r, \circ[\emptyset, \circ[\Diamond q, p], \bullet[\neg q \vee q]], \circ[\neg q]$, which are graphically displayed below as trees with labeled edges in order from left to right.



Nested sequent calculi typically exhibit a mode of inference referred to as *deep-inference*, whereby inference rules may be applied to any node within the tree encoded by the sequent [47]. This contrasts with *shallow-inference*, where inference rules are only applicable to the root of the tree encoded by the sequent. We remark that shallow-inference is an essential feature of *display calculi*, which will be discussed in Section 2.8 below. Although nested calculi are typically formulated with deep-inference, shallow-inference versions have been introduced [43, 55].[8] Nevertheless, as has become standard in the literature, we understand the term *nested sequent calculus* to mean *deep-inference nested sequent calculus* as the shallow-inference variants are known to be subsumed by the display calculus formalism [20] and will be considered as such.

[8] We note that in shallow-inference nested calculi certain rules called *display* or *resid-*

$$\frac{}{\Sigma\{\Gamma, p, \overline{p}\}}\ (id) \qquad \frac{\Sigma\{\Gamma, A, B\}}{\Sigma\{\Gamma, A \vee B\}}\ (\vee) \qquad \frac{\Sigma\{\Gamma, A\} \qquad \Sigma\{\Gamma, B\}}{\Sigma\{\Gamma, A \wedge B\}}\ (\wedge)$$

$$\frac{\Sigma\{\Gamma, \circ[A]\}}{\Sigma\{\Gamma, \Box A\}}\ (\Box) \qquad \frac{\Sigma\{\Diamond A, \circ[\Gamma, A]\}}{\Sigma\{\Diamond A, \circ[\Gamma]\}}\ (\Diamond_1) \qquad \frac{\Sigma\{\Gamma, A, \bullet[\Delta, \Diamond A]\}}{\Sigma\{\Gamma, \bullet[\Delta, \Diamond A]\}}\ (\Diamond_2)$$

$$\frac{\Sigma\{\Gamma, \bullet[A]\}}{\Sigma\{\Gamma, \blacksquare A\}}\ (\blacksquare) \qquad \frac{\Sigma\{\blacklozenge A, \bullet[\Gamma, A]\}}{\Sigma\{\blacklozenge A, \bullet[\Gamma]\}}\ (\blacklozenge_1) \qquad \frac{\Sigma\{\Gamma, A, \circ[\Delta, \blacklozenge A]\}}{\Sigma\{\Gamma, \circ[\Delta, \blacklozenge A]\}}\ (\blacklozenge_2)$$

Figure 9: The nested sequent system $\mathsf{N}(\mathsf{Kt})$ for the modal logic Kt [55].

An example of a nested sequent calculus for the tense logic Kt, referred to as $\mathsf{N}(\mathsf{Kt})$, is provided in Figure 11 and is due to Kashima [55]. The notation $\Sigma\{\Gamma\}$ is commonly employed in the formulation of nested inference rules and exhibits deep-inference. We read $\Sigma\{\Gamma\}$ as stating that the nested sequent Γ occurs at some node in the tree encoded by the nested sequent Σ. For example, we can write the nested sequent Σ_2 above as $\Sigma_2\{p, \bullet[q]\}$, or the nested sequent Σ_3 above as $\Sigma_3\{\neg q\}$ in this notation, thus letting us refer to the displayed nodes and the data confined within. Similarly, we may refer to multiple nodes in a nested sequent Σ simultaneously by means of the notation $\Sigma\{\Gamma_1\}\{\Gamma_2\} \cdots \{\Gamma_n\}$. For instance, we could write Σ_2 as $\Sigma_2\{p\}\{q\}$ or Σ_3 as $\Sigma_3\{p \wedge r\}\{\circ[\Diamond q, p]\}$.

The (id) rule in $\mathsf{N}(\mathsf{Kt})$ states that any nested sequent containing both p and $\neg p$ at a node is an axiom. The remaining rules tell us how complex logical formulae may be constructed within any given node of a derivable nested sequent. For example, $(\vee)$ states that A, B can be replaced by $A \vee B$, and $(\Box)$ states that $\circ[A]$ can be replaced by $\Box A$. As an example of how derivations are constructed in nested sequent calculi, we show how the modal axiom K (in negation normal form) can be derived in $\mathsf{N}(\mathsf{Kt})$ below.

---

uation rules are required for completeness. This will be discussed in the next section.

$$\frac{}{\Sigma\{\Gamma_1, p \Rightarrow \Delta_1\}_w\{\Gamma_2 \Rightarrow p, \Delta_2\}_u} (id)^{\dagger} \qquad \frac{}{\Sigma\{\Gamma, \bot \Rightarrow \Delta\}_w} (\bot_l)$$

$$\frac{\Sigma\{\Gamma, A \Rightarrow \Delta\}_w \qquad \Sigma\{\Gamma, B \Rightarrow \Delta\}_w}{\Sigma\{\Gamma, A \vee B \Rightarrow \Delta\}_w} (\vee_l) \qquad \frac{\Sigma\{\Gamma \Rightarrow A, B, \Delta\}_w}{\Sigma\{\Gamma \Rightarrow A \vee B, \Delta\}_w} (\vee_r)$$

$$\frac{\Sigma\{\Gamma, A, B \Rightarrow \Delta\}_w}{\Sigma\{\Gamma, A \wedge B \Rightarrow \Delta\}_w} (\wedge_l) \qquad \frac{\Sigma\{\Gamma \Rightarrow A, \Delta\}_w \qquad \Sigma\{\Gamma \Rightarrow B, \Delta\}_w}{\Sigma\{\Gamma \Rightarrow A \wedge B, \Delta\}_w} (\wedge_r)$$

$$\frac{\Sigma\{\Gamma_1, A \supset B \Rightarrow \Delta_1\}_w\{\Gamma_2, B \Rightarrow \Delta_2\}_u \quad \Sigma\{\Gamma_1, A \supset B \Rightarrow \Delta_1\}_w\{\Gamma_2 \Rightarrow A, \Delta_2\}_u}{\Sigma\{\Gamma_1, A \supset B \Rightarrow \Delta_1\}_w\{\Gamma_2 \Rightarrow \Delta_2\}_u} (\supset_l)^{\dagger}$$

$$\frac{\Sigma\{\Gamma \Rightarrow \Delta, [A \Rightarrow B]_u\}_w}{\Sigma\{\Gamma \Rightarrow \Delta, A \supset B\}_w} (\supset_r)$$

Side condition: $\dagger = u$ must be reachable from w.

Figure 10: The nested sequent system $\mathsf{N}(\mathsf{IL})$ for intuitionistic logic.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\Diamond(p \wedge \neg q), \Diamond\neg p, \circ[p, \neg p, q]} (id) \qquad \dfrac{}{\Diamond(p \wedge \neg q), \Diamond\neg p, \circ[\neg q, \neg p, q]} (id)}{\Diamond(p \wedge \neg q), \Diamond\neg p, \circ[p \wedge \neg q, \neg p, q]} (\wedge)}{\Diamond(p \wedge \neg q), \Diamond\neg p, \circ[q]} (\Diamond) \times 2}{\Diamond(p \wedge \neg q), \Diamond\neg p, \Box q} (\Box)}{\Diamond(p \wedge \neg q) \vee \Diamond\neg p \vee \Box q} (\vee) \times 2$$

Nested sequent calculi admit a couple methods of construction, which have proven to be rather general. One method is due to Goré et al. [43, 44] and consists of extracting nested sequent calculi from display calculi. The second method, referred to as *structural refinement*, is due to Lyon [72, 73] and consists of extracting nested sequent calculi from labeled sequent calculi or semantic presentations of non-classical logics.[9] In fact, a general algorithm was recently defined for extracting (cut-free) nested sequent calculi from (Horn) labeled sequent calculi [79]. Since methods of construction

[9]Labeled sequent calculi are discussed in Section 2.9 below.

for display and labeled calculi are well-understood and general, these approaches have led to the formulation of broad classes of cut-free nested sequent calculi for a variety of logics, including bi-intuitionistic logics [43, 80], intuitionistic modal logics [72], (deontic) agency logics [82, 73], and standpoint logic [77] (used in knowledge integration).

Both methods rely on the elimination of structural rules in a display or labeled calculus, replacing them with *propagation rules* [16, 31, 106], or the more general class of *reachability rules* [73, 74, 80]. Propagation rules operate by (bottom-up) propagating data along paths within a sequent, whereas reachability rules have the added functionality that data can be searched for within a sequent, and potentially propagated elsewhere (see [73, Chapter 5] for a discussion of these types of rules). Since propagation and reachability rules play a crucial role in the formulation of nested sequent calculi, we will demonstrate their functionality by means of an example. More specifically, we will introduce the nested sequent calculus $\mathsf{N}(\mathsf{IL})$, shown in Figure 10, which employs the $(\supset_l)$ propagation rule and (id) reachability rule.[10]

Nested sequents in $\mathsf{N}(\mathsf{IL})$ are generated via the following grammars:

$$\Sigma ::= \Gamma \Rightarrow \Gamma \mid \Sigma, [\Sigma]_w \qquad \Gamma ::= A \mid \emptyset \mid \Gamma, \Gamma$$

where $A \in \mathcal{L}_I$, w is among a countable set of labels w, u, v ..., and $\emptyset$ is the empty multiset. The notation used in the rules of $\mathsf{N}(\mathsf{IL})$ marks nestings with labels, e.g., in (id) and $(\supset_l)$ the labels w and u are used. It is assumed that each label is used once in a nested sequent and we note that such labels are merely a naming device used to simplify the formulation of certain inference rules. As stated in Figure 10, the (id) and $(\supset_l)$ rules have a side condition stating that each respective rule is applicable only if the node u is reachable from the node w. This means that in the tree encoded by the nested sequent Σ, the rule is applicable only if there is a path (which could be of length 0) from w to u. For example, in the $\mathsf{N}(\mathsf{IL})$ proof below the $(\supset_l)$ rule recognizes the $p \supset q$ in the w nesting and propagates q into

[10] The calculus $\mathsf{N}(\mathsf{IL})$ is the propositional fragment of the nested calculi given for first-order intuitionistic logics in [73], and is a variation of the nested calculus given by Fitting [32].

$[p \Rightarrow q]_u$ in the left premise and p into $[p \Rightarrow q]_u$ in the right premise when read bottom-up. Each premise of $(\supset_l)$ can be read as an instance of (id) since u is reachable from u with a path of length 0.

$$
\cfrac{\cfrac{}{\Rightarrow (p \supset q) \supset (p \supset q)}}{}
$$

$$
\dfrac{\dfrac{\dfrac{\dfrac{}{\Rightarrow [p \supset q \Rightarrow [p, q \Rightarrow q]_u]_w}(id) \quad \dfrac{}{\Rightarrow [p \supset q \Rightarrow [p \Rightarrow p, q]_u]_w}(id)}{\Rightarrow [p \supset q \Rightarrow [p \Rightarrow q]_u]_w}(\supset_l)}{\Rightarrow [p \supset q \Rightarrow p \supset q]_w}(\supset_r)}{\Rightarrow (p \supset q) \supset (p \supset q)}(\supset_r)
$$

In Section 3, we show how $\mathsf{N}(\mathsf{IL})$ can be extracted from a labeled sequent calculus, thus exemplifying the structural refinement method [73, 72].

We end this subsection with a brief discussion concerning the relationship between propagation/reachability rules and the property of *modularity*, that is, the ease with which a calculus for one logic may be transformed into a calculus for another logic within a given class. An interesting feature of propagation and reachability rules concerns the means by which they introduce modularity into a proof calculus. It has been argued—most notably by Avron [2, Section 1] and Wansing [113, Section 3.3]—that modularity ought to be obtained via *Došen's Principle*, which is stated accordingly:

> [T]he rules for the logical operations are never changed: all changes are made in the structural rules [26, p. 352]

Although we agree that modularity is an important feature of a proof formalism, we argue that Došen's principle is *too strict*. This perspective is supported by the formulation of propagation/reachability rules within nested systems, which attain modularity by a different means. Since these types of rules generalize the functionality of logical rules by permitting data to be shifted or consumed along paths within a nested sequent, systems which include such rules possess a high degree of modularity, obtained by simply changing the paths considered, irrespective of structural rules.

2.8. Display sequents

Introduced by Belnap [5] (and originally called *Display Logic*), the *Display Calculus* formalism generalizes Gentzen's sequent calculus by supplementing the structural connective (,) and the turnstile ($\Rightarrow$) with a host of new structural connectives—corresponding to pairs of dual connectives—and rules manipulating them. Incorporating structural connectives for pairs of dual connectives has proven fruitful for the construction of cut-free proof systems for large classes of logics, including modal and intuitionistic logics [5], tense logics [56, 113], bunched implication logics [9], resource sensitive logics [45], and bi-intuitionistic logic [115]. Display calculi also admit algorithmic constructibility starting from Hilbert axioms [22, 46]. To provide the reader with intuition concerning display systems and related concepts we accompany our general descriptions of such systems with concrete examples in the context of tense logics [102] and which comes from the work in [55, 56, 113].

First, to demonstrate the concept of a structural connective, let us define *structures*, which serve as the entire antecedent or consequent of a display sequent and fuse together formulae by means of structural connectives. When defining display sequents for tense logics, we let a structure X be a formula obtained via the following grammar in BNF:

$$X ::= A \mid I \mid *X \mid \bullet X \mid (X \circ X)$$

where A is a formula in the language of tense logic, i.e., within the language $\mathcal{L}_T$. Using X, Y, Z, ... to represent structures, we define a *display sequent* to be a formula of the form $X \Rightarrow Y$. We provide the reader with an example of a display sequent as well as show the pair of graphs representing the structures present in the antecedent and consequent of the display sequent.

Example 2.2. As can be seen in the example below the antecedent (shown bottom left) and consequent (shown bottom right) of a (tense) display sequent encodes a polytree; cf. [75].

$$\underbrace{\bullet\, (A_1 \circ *A_2) \circ \bullet A_3 \circ A_4}_{Antecedent} \Rightarrow \underbrace{A_5 \circ \bullet(*A_6 \circ *A_7) \circ \bullet A_8}_{Consequent}$$

$$\begin{tikzcd}
\boxed{A_1 \circ * A_2} \arrow[dr] & & & & \boxed{* A_6 \circ * A_7} \\
& \boxed{A_4} & & \boxed{A_5} \arrow[ur] \arrow[dr] & \\
\boxed{A_3} \arrow[ur] & & & & \boxed{A_8}
\end{tikzcd}$$

A characteristic feature of the display calculus is the display property, which states that every occurrence of a substructure in a sequent can be written (displayed) as the entire antecedent or succedent (but not both). Rules enabling the display property are called *display rules* or *residuation rules*, and display sequents derivable from one another via such rules are called *display equivalent*. These rules are invertible and hence a sequent can be identified with the class of its display equivalent sequents. For example, the bullet $\bullet$ represents a $\blacklozenge$ in the antecedent of a display sequent and a $\Box$ in the consequent, and since $\blacklozenge A \rightarrow B$ and $A \rightarrow \Box B$ are equivalent in the setting of tense logics, the display sequents $\bullet A \Rightarrow B$ and $A \Rightarrow \bullet B$ are defined to be mutually derivable from one another. This gives rise to the following *display rule* introduced by Wansing [113].

$$\frac{\bullet X \Rightarrow Y}{\overline{X \Rightarrow \bullet Y}}\ (\bullet)$$

As mentioned in the previous section, nested sequent calculi employing shallow-inference are also types of display calculi. As an example, if we take the nested calculus $\mathsf{N}(\mathsf{Kt})$ and add the display/residuation rules (rf) and (rp) rules shown below left as well as replace the $(\Diamond_1)$, $(\Diamond_2)$, $(\blacklozenge_1)$, and $(\blacklozenge_2)$ rules with the $(\Diamond)$ and $(\blacklozenge)$ rules shown below right, then we obtain Kashima's shallow-inference (i.e., display) calculus $\mathsf{D}(\mathsf{Kt})$ for the logic Kt [55]. The calculus $\mathsf{D}(\mathsf{Kt})$ can be seen as a 'one-sided' display calculus that equates nested sequents with structures.

$$\frac{\Gamma, \circ[\Delta]}{\bullet[\Gamma], \Delta}\ (rf) \qquad \frac{\Gamma, \bullet[\Delta]}{\circ[\Gamma], \Delta}\ (rp) \qquad \frac{\Gamma, \Diamond A, \circ[\Delta, A]}{\Gamma, \Diamond A, \circ[\Delta]}\ (\Diamond) \qquad \frac{\Gamma, \blacklozenge A, \bullet[\Delta, A]}{\Gamma, \blacklozenge A, \bullet[\Delta]}\ (\blacklozenge)$$

The rules (rf) and (rp) are similar to Wansing's display rule $(\bullet)$, however, they rely on the equivalence between $\blacksquare\neg A \vee B$ and $\neg A \vee \Box B$.

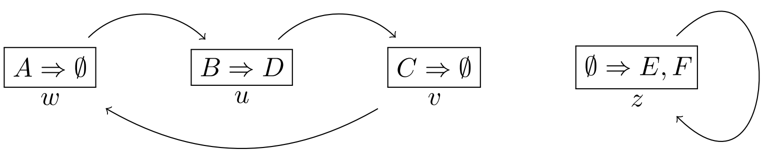
Quite significantly, Belnap's seminal paper [5] also proves a *general cut-elimination theorem*, that is, if a display calculus satisfies a set of eight conditions, then cut-elimination follows as a corollary. The display formalism has been used to supply truly sizable classes of logics with cut-free proof systems, proving the formalism general. Furthermore, the formalism is highly uniform and modular as structural rules are used to capture distinct logics, and due to the algorithms permitting their construction [22, 46], they enjoy constructibility. These nice features come at the cost of syntactic parsimony however as display sequents utilize complex structures to facilitate reasoning.

2.9. Labeled sequents

Labeled sequents generalize Gentzen sequents by annotating formulae with labels and introducing semantic elements into the syntax of sequents. For example, the labeled sequents used by Simpson [106] and Viganò [112] have the form shown below top in the following example and encode a binary graph of Gentzen sequents. Thus, labeled sequents properly generalize all sequents considered in previous sections.

Example 2.3. A labeled sequent is shown below top and its corresponding graph is shown below bottom.

$$\underbrace{wRu, uRv, vRw, zRz}_{\text{Relational Atoms}}, \underbrace{w : A, u : B, v : C}_{\text{Labeled Formulae}} \Rightarrow \underbrace{u : D, z : E, z : F}_{\text{Labeled Formulae}}$$



The idea of labeling formulae in sequents comes from Kanger [54], who made use of *spotted formulae* to construct sequent systems for normal modal logics.[11] The labeled sequent formalism is quite general and covers many logics; e.g., intuitionistic (modal) logics [30, 106], normal modal logics [48, 93, 112], predicate modal logics [15, 59], relevance logics [112], and (deontic) agency logics [110, 111, 82].

The labeled sequent formalism offers a high degree of uniformity and modularity. This is typically obtained by taking a *base calculus* for a particular logic and showing that through the addition of structural rules the calculus can be extended into a calculus for another logic within a specified class. A favorable feature of the labeled sequent formalism is the existence of general theorems confirming properties such as cut-admissibility, invertibility of rules, and admissibility of standard structural rules [30, 106, 112]. Another desirable characteristic concerns the ease with which labeled systems are constructed; e.g., for logics with a Kripkean semantics one straightforwardly obtains labeled calculi by transforming the semantic clauses and frame/model conditions into inference rules. Thus, the labeled formalism is highly general, uniform, modular, and constructible, but as with the display formalism, this comes at a cost of syntactic complexity.

To provide the reader with intuition about the generality, uniformity, and modularity of labeled sequent systems we consider a specific system for the tense logic Kt (cf. [20, 7]). The labeled sequents used are defined to be expressions of the form $\mathcal{R} \Rightarrow \Gamma$, where $\mathcal{R}$ is a set of *relational atoms* wRu and Γ is a multiset of *labeled formulae* $w : A$ with $A \in \mathcal{L}_T$. The labeled sequent system $\mathsf{L}(\mathsf{Kt})$ is presented in Figure 11 and contains the axiom (id) as well as six logical rules; note that the $(\Box)$ and $(\blacksquare)$ rules is subject to a side condition, namely, the label u must be *fresh* and not occur in the conclusion of a rule application.

This calculus may be extended with structural rules to obtain labeled sequent systems for *extensions* of Kt (e.g., tense $\mathsf{S4}$ and $\mathsf{S5}$). Such rules encode frame properties corresponding to axioms. For example, to obtain a labeled sequent system for Kt with serial frames one can extend $\mathsf{L}(\mathsf{Kt})$ with the (ser) rule shown below, and to obtain a labeled sequent calculus for tense $\mathsf{S4}$ one can extend $\mathsf{L}(\mathsf{Kt})$ with the structural rules (ref) and (tra).

---

[11] We note that labeling has also been used in tableaux for modal logics, e.g., the prefixed tableaux of Fitting [31].

$$\frac{}{\mathcal{R} \Rightarrow w : p, w : \overline{p}, \Gamma}\ (id) \qquad \frac{\mathcal{R} \Rightarrow w : A, w : B, \Gamma}{\mathcal{R} \Rightarrow w : A \vee B, \Gamma}\ (\vee)$$

$$\frac{\mathcal{R} \Rightarrow w : A, \Gamma \qquad \mathcal{R} \Rightarrow w : B, \Gamma}{\mathcal{R} \Rightarrow w : A \wedge B, \Gamma}\ (\wedge)$$

$$\frac{\mathcal{R}, wRu \Rightarrow w : \Diamond A, u : A, \Gamma}{\mathcal{R}, wRu \Rightarrow w : \Diamond A, \Gamma}\ (\Diamond) \qquad \frac{\mathcal{R}, uRw \Rightarrow w : \blacklozenge A, u : A, \Gamma}{\mathcal{R}, uRw \Rightarrow w : \blacklozenge A, \Gamma}\ (\blacklozenge)$$

$$\frac{\mathcal{R}, wRu \Rightarrow u : A, \Gamma}{\mathcal{R} \Rightarrow w : \Box A, \Gamma}\ (\Box) \qquad \frac{\mathcal{R}, uRw \Rightarrow u : A, \Gamma}{\mathcal{R} \Rightarrow w : \blacksquare A, \Gamma}\ (\blacksquare)$$

Side condition: u must be fresh in $(\Box)$ and $(\blacksquare)$.

Figure 11: Labeled sequent system $\mathsf{L}(\mathsf{Kt})$ for the tense logic Kt [7, 20].

$$\frac{\mathcal{R}, wRu \Rightarrow \Gamma}{\mathcal{R} \Rightarrow \Gamma}\ (ser)\ \textit{with } u \textit{ fresh} \qquad \frac{\mathcal{R}, wRw \Rightarrow \Gamma}{\mathcal{R} \Rightarrow \Gamma}\ (ref)$$

$$\frac{\mathcal{R}, wRu, uRv, wRv \Rightarrow \Gamma}{\mathcal{R}, wRu, uRv \Rightarrow \Gamma}\ (tra)$$

In fact, most commonly studied frame properties for logics with Kripkean semantics (such as seriality, reflexivity, and transitivity) can be transformed into equivalent structural rules. Simpson showed that a large class of properties, referred to as *geometric axioms*, could be transformed into equivalent structural rules, referred to as *geometric rules*, in labeled sequent calculi [106]. A geometric axiom is a formula of the form shown below, where each A_i and $B_{j,k}$ is a first-order atom.

$$\forall x_1, \dots, x_t \Big(A_1 \wedge \dots \wedge A_n \rightarrow \exists y_1, \dots, y_s (\bigvee_{j=1}^{m} \bigwedge_{k=1}^{l_j} B_{1,k})\Big)$$

Each geometric formula is equivalent to the geometric rule of the form shown below, where $\vec{A} := A_1, \dots, A_n$, $\vec{B}_j := B_{j,1}, \dots, B_{j,l_j}$, and each variable

$$\frac{\mathcal{R} \Rightarrow \Delta}{\mathcal{R}, wRu \Rightarrow \Delta}\ (wk_l) \qquad \frac{\mathcal{R} \Rightarrow \Delta}{\mathcal{R} \Rightarrow w : A, \Delta}\ (wk_r) \qquad \frac{\mathcal{R} \Rightarrow \Delta}{\mathcal{R}[w/u] \Rightarrow \Delta[w/u]}\ (ls)$$

$$\frac{\mathcal{R} \Rightarrow w : A, w : A, \Delta}{\mathcal{R} \Rightarrow w : A, \Delta}\ (ctr) \qquad \frac{\mathcal{R} \Rightarrow w : A, \Delta \qquad \mathcal{R} \Rightarrow w : \overline{A}, \Delta}{\mathcal{R} \Rightarrow \Delta}\ (cut)$$

Figure 12: Labeled structural rules.

$y_1, \dots, y_s$ is fresh, i.e., the rule can be applied only if none of the variables $y_1, \dots, y_s$ occur in the conclusion.

$$\frac{\mathcal{R}, \vec{A}, \vec{B}_1 \Rightarrow \Delta \qquad \cdots \qquad \mathcal{R}, \vec{A}, \vec{B}_m \Rightarrow \Delta}{\mathcal{R}, \vec{A} \Rightarrow \Delta}$$

Figure 12 gives a selection of structural rules that are typically admissible in labeled sequent systems with geometric rules. The weakening rule (wk) adds additional labeled formulae to the consequent of a labeled sequent, the contraction rule (ctr) removes additional copies of labeled formulae, the substitution rule (ls) replaces a label u by a label w in a labeled sequent, and the (cut) rule encodes the transitivity of implication. The admissibility of these rules tends to hold generally for labeled sequent systems, along with all logical rules being invertible [30, 48, 82, 106]. Beyond admissibility and invertibility properties, labeled systems allow for easy counter-model extraction due to the incorporation of semantic notions into the syntax of sequents, though termination of proof-search is not easily achieved as labeled sequents contain a large amount of structure.

Last, we note that although (cut) is usually admissible in labeled sequent systems, it is often the case that a *strict* form of the subformula property fails to hold. This phenomenon arises due to the incorporation of geometric rules which may delete relational atoms from the premise when inferring the conclusion. Nevertheless, it is usually the case that labeled sequent systems possess a *weak* version of the subformula property, i.e., it can be shown that every *labeled formula* occurring in a derivation is a subformula of some labeled formula in the conclusion [112].

3. Navigating the proof-theoretic jungle

As discussed in Section 2 and shown in Figure 2, the data structure underlying sequents naturally imposes a hierarchy on sequent-style formalisms. At the base of this hierarchy sits Gentzen sequents, and each level of the hierarchy gets incrementally more general until labeled sequents are reached at the top. As we are interested in exploring this hierarchy, we present translations of proofs between systems in different proof-theoretic formalisms, thus letting us 'shift' derivations up and down the hierarchy. The lesson we learn is that translating proofs down the hierarchy (usually) requires significantly more work than translating proofs up the hierarchy.

3.1. Translations for S5: Labeled and hypersequent calculi

We begin our demonstration of how to translate proofs between distinct formalisms by considering translations between hypersequent and labeled calculi for the modal logic S5. In particular, we will explain how proofs are translated between the hypersequent calculus H(S5) (see Figure 7) and the labeled sequent calculus L(S5) shown in Figure 13.

In this section, we define a labeled sequent to be an expression of the form $\Gamma \Rightarrow \Delta$ such that Γ and Δ are finite multisets of labeled formulae $w : A$ with w among a denumerable set $\mathsf{Lab} := \{w, u, v, ...\}$ of labels and $A \in \mathcal{L}_M$. For a multiset Γ of labeled formulae, we define $\mathsf{Lab}(\Gamma)$ to be the set of all labels occurring in Γ, for a multiset $\{A_1, \dots, A_n\}$ of formulae, we define $w : \{A_1, \dots, A_n\} = \{w : A_1, \dots, w : A_n\}$, and we let $\Gamma(w)$ be the multiset $\{A \mid w : A \in \Gamma\}$.

A labeled sequent calculus L(S5) for the modal logic S5 is shown in Figure 13. We remark that the labeled sequents used in L(S5) have a simpler structure than those discussed in Section 2.9, namely, they do not use relational atoms. This is a special case and a byproduct of the fact that L(S5) is a calculus for the modal logic S5; in general, more complex modal logics require the use of relational atoms (cf. [106, 112]).

It is straightforward to define translations that map labeled sequents to hypersequents and vice-versa. To translate labeled sequents into hypersequents, we make use of the h translation, defined as follows:

$$\frac{}{\Gamma, w : p \Rightarrow w : p, \Delta}\ (id) \qquad \frac{}{\Gamma, w : \bot \Rightarrow \Delta}\ (\bot_l)$$

$$\frac{\Gamma \Rightarrow w : A, w : B, \Delta}{\Gamma \Rightarrow w : A \lor B, \Delta}\ (\lor_r) \qquad \frac{\Gamma, w : A \Rightarrow \Delta \qquad \Gamma, w : B \Rightarrow \Delta}{\Gamma, w : A \lor B \Rightarrow \Delta}\ (\lor_l)$$

$$\frac{\Gamma, w : A, w : B \Rightarrow \Delta}{\Gamma, w : A \land B \Rightarrow \Delta}\ (\land_l) \qquad \frac{\Gamma \Rightarrow w : A, \Delta \qquad \Gamma \Rightarrow w : B, \Delta}{\Gamma \Rightarrow w : A \land B, \Delta}\ (\land_r)$$

$$\frac{\Gamma \Rightarrow w : A, \Delta \qquad \Gamma, w : B \Rightarrow \Delta}{\Gamma, w : A \rightarrow B \Rightarrow \Delta}\ (\rightarrow_l) \qquad \frac{\Gamma, w : A \Rightarrow w : B, \Delta}{\Gamma \Rightarrow w : A \rightarrow B, \Delta}\ (\rightarrow_r)$$

$$\frac{\Gamma, w : \Box A, u : A \Rightarrow \Delta}{\Gamma, w : \Box A \Rightarrow \Delta}\ (\Box_l)^{\dagger_1} \qquad \frac{\Gamma \Rightarrow u : A, \Delta}{\Gamma \Rightarrow w : \Box A, \Delta}\ (\Box_r)^{\dagger_2}$$

Side conditions: $\dagger_1$ stipulates that $u \in \mathsf{Lab}(\Gamma, \Delta)$ in $(\Box_l)$ and $\dagger_2$ stipulates that u must be fresh in $(\Box_r)$.

Figure 13: The labeled calculus $\mathsf{L}(\mathsf{S5})$ for the modal logic $\mathsf{S5}$.

$$h(\Gamma \Rightarrow \Delta) := \Gamma(w_1) \Rightarrow \Delta(w_1) \mid \cdots \mid \Gamma(w_n) \Rightarrow \Delta(w_n)$$

where $\mathsf{Lab}(\Gamma, \Delta) := \{w_1, \ldots, w_n\}$. To translate hypersequents into labeled sequents, we make use of the ℓ translation, defined as follows:

$$\ell(\Gamma_1 \Rightarrow \Delta_1 \mid \cdots \mid \Gamma_n \Rightarrow \Delta_n) := \bigcup_{1 \leq i \leq n} w_i : \Gamma_i \Rightarrow \bigcup_{1 \leq i \leq n} w_i : \Delta_i$$

Using the above translations, we can confirm that all derivations (which, properly includes all proofs; see Section 2.1) in $\mathsf{L}(\mathsf{S5})$ and $\mathsf{H}(\mathsf{S5})$ are isomorphic to each other.

Proposition 3.1. Every derivation in $\mathsf{L}(\mathsf{S5})$ is isomorphic to a derivation in $\mathsf{H}(\mathsf{S5})$ under the h translation, and every derivation in $\mathsf{H}(\mathsf{S5})$ is isomorphic to a derivation in $\mathsf{L}(\mathsf{S5})$ under the ℓ translation.

Proof: We prove the case for the h translation by induction on the height of the given derivation $\mathcal{D}$, and remark that the case of translating proofs with the reverse translation ℓ is similar.

Base case. If $\Gamma \Rightarrow \Delta$ is an axiom, i.e., an instance of (id) or $(\bot_l)$, then

$h(\Gamma \Rightarrow \Delta)$ will be an axiom in $\mathsf{H}(\mathsf{S5})$ as well. If $\Gamma \Rightarrow \Delta$ is a leaf in the derivation $\mathcal{D}$, but not an axiom, then $h(\Gamma \Rightarrow \Delta)$ trivially translates to a leaf in the hypersequent derivation.

Inductive step. We show how to translate the $(\Box_l)$ case. There are two cases to consider in the $(\Box_l)$ case: in the first case, the label of the auxiliary formula A is identical to the label of the principal formula $\Box A$. This is resolved as shown below where $G = h(\Gamma \setminus w : \Gamma(w) \Rightarrow \Delta \setminus w : \Delta(w))$.

$$\dfrac{\dfrac{\dfrac{h(\Gamma, w : \Box A, w : A \Rightarrow \Delta)}{G \mid \Box A, A, \Gamma(w) \Rightarrow \Delta(w)} =}{G \mid \Box A, \Gamma(w) \Rightarrow \Delta(w)} (\Box_{l1})}{h(\Gamma, w : \Box A \Rightarrow \Delta)} =$$

In the second case, the label u of the auxiliary formula is distinct from the label of the principal formula. This case is resolved as shown below where $G = h(\Gamma \setminus \{w : \Gamma(w), u : \Gamma(u)\} \Rightarrow \Delta \setminus \{w : \Delta(w), u : \Delta(u)\})$

$$\dfrac{\dfrac{\dfrac{h(\Gamma, w : \Box A, u : A \Rightarrow \Delta)}{G \mid \Box A, \Gamma(w) \Rightarrow \Delta(w) \mid A, \Gamma(u) \Rightarrow \Delta(u)} =}{G \mid \Box A, \Gamma(w) \Rightarrow \Delta(w) \mid \Gamma(u) \Rightarrow \Delta(u)} (\Box_{l2})}{h(\Gamma, w : \Box A \Rightarrow \Delta)} =$$

The remaining cases are easily resolved by applying IH and then the corresponding rule in $\mathsf{H}(\mathsf{S5})$. $\Box$

3.2. Translations for Kt: Labeled and display calculi

We show how to translate proofs from $\mathsf{L}(\mathsf{Kt})$ into $\mathsf{D}(\mathsf{Kt})$. The method of translation we present was first defined in [20] and is strong enough to not only translate labeled proofs into display proofs for Kt, but also for any extension of Kt with *path axioms* of the form $\langle ? \rangle_1 \cdots \langle ? \rangle_n p \rightarrow \langle ? \rangle_{n+1} p$ with $\langle ? \rangle_i \in \{\blacklozenge, \Diamond\}$ for $1 \leq i \leq n+1$. A generalization of this technique is presented in [75] and shows how to translate cut-free display proofs into cut-free labeled sequent proofs for the even wider class of *primitive tense logics* [56]. We note that the converse translation from $\mathsf{D}(\mathsf{Kt})$ to $\mathsf{L}(\mathsf{Kt})$

is simpler so we omit it, though the details can be found in [20] for the interested reader.

The key to translating labeled proofs into display proofs is to recognize that 'non-treelike' data (e.g., loops and cycles) cannot occur in proofs of theorems. We refer to these labeled sequents as *labeled polytree sequents* [20] and define them below. This insight is useful as labeled polytree sequents and display sequents are notational variants of one another, which facilitates our translation from $\mathsf{L}(\mathsf{Kt})$ to $\mathsf{D}(\mathsf{Kt})$.

DEFINITION 3.2 (Labeled Polytree). Let $\Lambda := \mathcal{R}, \Gamma \Rightarrow \Delta$ be a labeled sequent, and define the graph $G(\mathcal{R}) = (V, E)$ such that V is the set of labels ocurring in $\mathcal{R}$ and $E = \{(w, u) \mid wRu \in \mathcal{R}\}$. We define Λ to be a *labeled polytree sequent iff* $\mathcal{R}$ forms a polytree, i.e., the graph $G(\mathcal{R})$ is connected and cycle-free, and all labels in Γ, Δ occur in $\mathcal{R}$ (unless $\mathcal{R}$ is empty, in which case every labeled formula in Γ, Δ must share the same label). We define a *labeled polytree derivation* to be a derivation containing only labeled polytree sequents.

LEMMA 3.3. *Every derivation of a formula A in $\mathsf{L}(\mathsf{Kt})$ is a labeled polytree derivation.*

PROOF: Suppose we are given a derivation of the labeled polytree sequent $\Rightarrow w : A$ in $\mathsf{L}(\mathsf{Kt})$. Observe that every rule of $\mathsf{L}(\mathsf{Kt})$, if applied bottom-up to a labeled polytree sequent, yields a labeled polytree sequent since rules either preserve the set $\mathcal{R}$ of relational atoms when applied bottom-up (e.g., $(\vee)$ and $(\Diamond)$), or via $(\Box)$ or $(\blacksquare)$, add a new relational atom from a label occurring in the labeled sequent to a fresh label (which has the effect of adding a new forward or backward edge in the polytree encoded by the labeled sequent). Hence, the derivation of $\Rightarrow w : A$ in $\mathsf{L}(\mathsf{Kt})$ must be a labeled polytree derivation. $\Box$

We now define the d function that maps labeled polytree sequents to display sequents, which can be stepwise applied to translate entire labeled polytree proofs into display proofs. As it will be useful here, and later on, we define the *sequent composition* $\Lambda \odot \Lambda'$ between two labeled sequents $\Lambda = \mathcal{R} \Rightarrow \Gamma$ and $\Lambda' = \mathcal{R}' \Rightarrow \Gamma'$ to be $\Lambda \odot \Lambda' := \mathcal{R}, \mathcal{R}' \Rightarrow \Gamma, \Gamma'$.

Definition 3.4 (Translation d). Let $\Lambda := \mathcal{R} \Rightarrow \Gamma$ be a labeled polytree sequent containing the label u. We define $\Lambda' \subseteq \Lambda$ *iff* there exists a labeled polytree sequent Λ'' such that $\Lambda = \Lambda' \odot \Lambda''$. Let us define $\Lambda_u := \mathcal{R}' \Rightarrow \Gamma'$ to be the unique labeled polytree sequent rooted at u such that $\Lambda_u \subseteq \Lambda$ and $\Gamma' \restriction u = \Gamma \restriction u$. We recursively define $d_u(\Lambda)$:

(1) if $\mathcal{R} = \emptyset$, then $d_v(\Lambda) := (\Rightarrow \Gamma \restriction v)$, and

(2) if $vRx_1, \ldots vRx_n$ and $y_1Rv, \ldots y_nRv$ are all relational atoms of the form vRx and yRx, respectively, then

$$d_v(\Lambda) := \Gamma \restriction v, \circ[d_{x_1}(\Lambda_{x_1})], \ldots, \circ[d_{x_n}(\Lambda_{x_n})], \bullet[d_{y_1}(\Lambda_{y_1})], \ldots, \bullet[d_{y_k}(\Lambda_{y_k})].$$

Example 3.5. We let $\Lambda = \; wRv, vRu \Rightarrow w : \Diamond q, w : r \vee q, v : p, v : q, u : \blacksquare p$ and show the output display sequent for w, u, and v.

$$\begin{aligned} d_w(\Lambda) \quad &= \Diamond q, r \vee q, \circ[p, q, , \circ[\blacksquare p]] \\ d_v(\Lambda) \quad &= \bullet[\Diamond q, r \vee q], p, q, \circ[\blacksquare p] \\ d_u(\Lambda) \quad &= \bullet[\bullet[\Diamond q, r \vee q], p, q], \blacksquare p \end{aligned} \tag{3.1}$$

We find something interesting if we observe the display sequents $d_w(\Lambda)$, $d_v(\Lambda)$, and $d_u(\Lambda)$ above, namely, each display sequent is derivable from the other by means of the display rules (*rf*) and (*rp*). In fact, as stated in the following lemma, this relationship holds generally; its proof can be found in [20].

Lemma 3.6. *If $\Lambda = \mathcal{R} \Rightarrow \Gamma$ is a labeled polytree sequent with labels w and u, then $d_w(\Lambda)$ and $d_u(\Lambda)$ are display equivalent, i.e., both are mutually derivable with the (rf) and (rp) rules.*

Relying on Lemma 3.3 and 3.6, we can define a proof translation from $\mathsf{L}(\mathsf{Kt})$ to $\mathsf{D}(\mathsf{Kt})$ as specified in the proof of the following theorem.

Theorem 3.7. *Every proof of a formula A in $\mathsf{L}(\mathsf{Kt})$ can be step-wise translated into a proof of A in $\mathsf{D}(\mathsf{Kt})$.*

Proof: Suppose we are given a proof of a formula A in $\mathsf{L}(\mathsf{Kt})$, we know by Lemma 3.15 that the proof is a labeled polytree proof, and thus, d is defined for every labeled sequent in the proof. We show that the proof can be translated into a proof in $\mathsf{D(Kt)}$ by induction on the height of the proof. We only consider the $(\Box)$ and $(\blacklozenge)$ cases of the inductive step as the remaining cases are trivial or similar.

$$\dfrac{\mathcal{R}, wRu \Rightarrow u : A, \Gamma}{\mathcal{R} \Rightarrow w : \Box A, \Gamma}(\Box) \quad \rightsquigarrow \quad \dfrac{\dfrac{\dfrac{d_w(\mathcal{R}, wRu \Rightarrow u : A, \Gamma)}{d_w(\mathcal{R} \Rightarrow \Gamma), \circ[A]} =}{d_w(\mathcal{R} \Rightarrow \Gamma), \Box A}(\Box)}{d_w(\mathcal{R} \Rightarrow w : \Box A, \Gamma)} =$$

$$\dfrac{\mathcal{R}, wRu \Rightarrow u : A, \Gamma}{\mathcal{R} \Rightarrow w : \blacklozenge A, \Gamma}(\blacklozenge) \quad \rightsquigarrow \quad \dfrac{\dfrac{\dfrac{d_u(\mathcal{R}, wRu \Rightarrow w : A, u : \blacklozenge A, \Gamma)}{X, \blacklozenge A, \bullet[Y, A]} =}{X, \blacklozenge A, \bullet[Y]}(\blacklozenge)}{d_u(\mathcal{R}, wRu \Rightarrow u : \blacklozenge A, \Gamma)} =$$

The remaining cases of the translation can be found in [20]. □

3.3. Translations for IL: Labeled, nested, and sequent calculi

We now consider translating proofs between the sequent calculus $\mathsf{S(IL)}$ and a labeled calculus $\mathsf{L(IL)}$ for intuitionistic logic shown in Figure 14. The translation from the sequent calculus to the 'richer' labeled sequent calculus is relatively straightforward and demonstrates the ease with which proofs may be translated up the proof-theoretic hierarchy (Figure 2). As traditional sequents are simpler than labeled sequents, the converse translation requires special techniques to remove extraneous structure from labeled proofs. To accomplish this task we utilize structural rule elimination (cf. [73, 70]) to first transform labeled proofs into nested proofs in $\mathsf{N(IL)}$ (see Figure 10), and then extract sequent proofs from these.

3.3.1. From sequents to labeled sequents

The labeled sequent calculus $\mathsf{L(IL)}$ (Figure 14) makes use of labeled sequents of the form $\mathcal{R}, \Gamma \Rightarrow \Delta$, where $\mathcal{R}$ is a (potentially empty) multiset of relational atoms of the form $w \leq u$ and Γ and Δ are (potentially empty) multisets of labeled formulae of the form $w : A$ with $A \in \mathcal{L}_I$. The theorem below gives a translation of proofs in $\mathsf{S}(\mathsf{IL})$ into proofs in $\mathsf{L}(\mathsf{IL})$.

THEOREM 3.8. *Every proof of a sequent* $\Gamma \Rightarrow \Delta$ *in* $\mathsf{S}(\mathsf{IL})$ *can be step-wise translated into a proof of* $w : \Gamma \Rightarrow w : \Delta$ *in* $\mathsf{L}(\mathsf{IL})$.

PROOF: By induction on the height of the given proof in $\mathsf{S}(\mathsf{IL})$.

Base case. The (id) rule is translated as shown below; translating the $(\bot_l)$ rule is similar.

$$\frac{}{\Gamma, p \Rightarrow p, \Delta}(id) \quad \rightsquigarrow \quad \dfrac{\dfrac{}{w \leq w, w : \Gamma, w : p \Rightarrow w : p, w : \Delta}(id)}{w : \Gamma, w : p \Rightarrow w : p, w : \Delta}(ref)$$

Inductive step. As the $(\vee_l)$, $(\vee_r)$, $(\wedge_l)$, and $(\wedge_r)$ cases are simple, we only show the more interesting cases of translating the $(\supset_l)$ and $(\supset_r)$ rules.

$(\supset_l)$. For the $(\supset_l)$ case, we assume we are given a derivation in $\mathsf{S}(\mathsf{IL})$ ending with an application of the $(\supset_l)$ rule, as shown below:

$$\frac{\Gamma, A \supset B, B \Rightarrow \Delta \qquad \Gamma, A \supset B \Rightarrow A, \Delta}{\Gamma, A \supset B \Rightarrow \Delta}(\supset_l)$$

To translate the proof and inference into the desired proof in $\mathsf{L}(\mathsf{IL})$, we invoke IH, apply the admissible (wk_l) rule, apply $(\supset_l)$, and finally, apply (ref) as shown below:

$$\mathcal{D} = \quad \dfrac{\dfrac{}{w : \Gamma, w : A \supset B, w : B \Rightarrow w : \Delta}\text{IH}}{w \leq w, w : \Gamma, w : A \supset B, w : B \Rightarrow w : \Delta}(wk_l)$$

$$\dfrac{\dfrac{\mathcal{D} \qquad \dfrac{\dfrac{}{w : \Gamma, w : A \supset B \Rightarrow w : A, w : \Delta}\text{IH}}{w \leq w, w : \Gamma, w : A \supset B \Rightarrow w : A, w : \Delta}(wk_l)}{w \leq w, w : \Gamma, w : A \supset B \Rightarrow w : \Delta}(\supset_l)}{w : \Gamma, w : A \supset B \Rightarrow w : \Delta}(ref)$$

$(\supset_r)$. Translating the $(\supset_r)$ rule requires more effort. We must make use of the admissible weakening and label substitution rules (wk_l) and (ls) along with the following admissible lift rule:

$$\frac{\mathcal{R}, w \leq u, \Gamma, w : A, u : A \Rightarrow \Delta}{\mathcal{R}, w \leq u, \Gamma, w : A \Rightarrow \Delta}(lift)$$

The translation is defined as shown below:

$$
\dfrac{\Gamma, A \Rightarrow B}{\Gamma \Rightarrow A \supset B, \Delta}\ (\supset_r) \quad \rightsquigarrow \quad \dfrac{\dfrac{\dfrac{\dfrac{\overline{w : \Gamma, w : A \Rightarrow w : B}\ \mathrm{IH}}{u : \Gamma, u : A \Rightarrow u : B}\ (ls)}{w \leq u, w : \Gamma, u : \Gamma, u : A \Rightarrow u : B}\ (wk_l)}{w \leq u, w : \Gamma, u : A \Rightarrow u : B}\ (lift)}{w : \Gamma \Rightarrow w : A \supset B}\ (\supset_r)
$$

□

3.3.2. From labeled sequents to sequents

We now consider the converse translation from $\mathsf{L(IL)}$ to $\mathsf{S(IL)}$, which demonstrates the non-triviality of translating from the richer labeled sequent formalism to the sequent formalism. In this section, our main aim is to establish the following theorem:

THEOREM 3.9. *Every proof of a formula A in $\mathsf{L(IL)}$ can be step-wise translated into a proof of A in $\mathsf{S(IL)}$.*

We prove the above theorem by establishing two lemmata: (1) we translate labeled proofs from $\mathsf{L(IL)}$ into nested proofs in $\mathsf{N(IL)}$, and (2) we translate nested proofs into sequent proofs in $\mathsf{S(IL)}$. We then obtain the desired translation from $\mathsf{L(IL)}$ to $\mathsf{S(IL)}$ by composing the two aforementioned ones. We first focus on proving the labeled to nested translation, and then argue the nested to sequent translation.

DEFINITION 3.10 (Labeled Tree). We define a *labeled tree sequent* to be a labeled sequent $\Lambda := \mathcal{R}, \Gamma \Rightarrow \Delta$ such that $\mathcal{R}$ forms a tree and all labels in Γ, Δ occur in $\mathcal{R}$ (unless $\mathcal{R}$ is empty, in which case every labeled formula in Γ, Δ must share the same label). We define a *labeled tree derivation* to be a proof containing only labeled tree sequents. We say that a labeled tree derivation has the *fixed root property iff* every labeled sequent in the derivation has the same root.

DEFINITION 3.11 (Translation n). Let $\Lambda := \mathcal{R}, \Gamma \Rightarrow \Delta$ be a labeled tree sequent with root u. We define $\Lambda' \subseteq \Lambda$ *iff* there exists a labeled tree sequent Λ'' such that $\Lambda = \Lambda' \odot \Lambda''$. Let us define $\Lambda_u := \mathcal{R}', \Gamma' \Rightarrow \Delta'$ to be the

$$\frac{}{\mathcal{R}, w \leq u, \Gamma, w : p \Rightarrow u : p, \Delta}\ (id) \qquad \frac{}{\mathcal{R}, \Gamma, w : \bot \Rightarrow \Delta}\ (\bot_l)$$

$$\frac{\mathcal{R}, w \leq u, \Gamma, u : A \Rightarrow u : B, \Delta}{\mathcal{R}, \Gamma \Rightarrow w : A \supset B, \Delta}\ (\supset_r)$$

$$\frac{\mathcal{R}, \Gamma, w : A \Rightarrow \Delta \qquad \mathcal{R}, \Gamma, w : B \Rightarrow \Delta}{\mathcal{R}, \Gamma, w : A \lor B \Rightarrow w : \Delta}\ (\lor_l) \qquad \frac{\mathcal{R}, \Gamma \Rightarrow w : A, w : B, \Delta}{\mathcal{R}, \Gamma \Rightarrow w : A \lor B, \Delta}\ (\lor_r)$$

$$\frac{\mathcal{R}, \Gamma, w : A, w : B \Rightarrow \Delta}{\mathcal{R}, \Gamma, w : A \land B \Rightarrow \Delta}\ (\land_l) \qquad \frac{\mathcal{R}, \Gamma \Rightarrow w : A, \Delta \qquad \mathcal{R}, \Gamma \Rightarrow w : B, \Delta}{\mathcal{R}, \Gamma \Rightarrow w : A \land B, \Delta}\ (\land_r)$$

$$\frac{\mathcal{R}, w \leq u, \Gamma, w : A \supset B, u : B \Rightarrow \Delta \qquad \mathcal{R}, w \leq u, \Gamma, w : A \supset B \Rightarrow u : A, \Delta}{\mathcal{R}, w \leq u, \Gamma, w : A \supset B \Rightarrow \Delta}\ (\supset_l)$$

$$\frac{\mathcal{R}, w \leq w, \Gamma \Rightarrow \Delta}{\mathcal{R}, \Gamma \Rightarrow \Delta}\ (ref) \qquad \frac{\mathcal{R}, w \leq u, u \leq v, w \leq v, \Gamma \Rightarrow \Delta}{\mathcal{R}, w \leq u, u \leq v, \Gamma \Rightarrow \Delta}\ (tra)$$

Side conditions: u is fresh in $(\supset_r)$.

Figure 14: The labeled sequent calculus $\mathsf{L(IL)}$ for intuitionistic logic [30].

unique labeled tree sequent rooted at u such that $\Lambda_u \subseteq \Lambda$, $\Gamma' \upharpoonright u = \Gamma \upharpoonright u$, and $\Delta' \upharpoonright u = \Delta \upharpoonright u$. We recursively define $n(\Lambda) := n_u(\Lambda)$:

$$n_v(\Lambda) := \begin{cases} \Gamma \upharpoonright v \Rightarrow \Delta \upharpoonright v & \text{if } \mathcal{R} = \emptyset; \\ \Gamma \upharpoonright v \Rightarrow \Delta \upharpoonright v, [n_{z_1}(\Lambda_{z_1})], \ldots, [n_{z_n}(\Lambda_{z_n})] & \text{otherwise.} \end{cases}$$

In the second case above, we assume that $v \leq z_1, \ldots v \leq z_n$ are all of the relational atoms occurring in the input sequent which have the form $v \leq x$.

Example 3.12. We let $\Lambda := w \leq v, v \leq u, v : p, u : p \Rightarrow w : p \supset q, v : r, u : q$ and show the output nested sequent under the translation n.

$$n(\Lambda) = n_w(\Lambda) = \emptyset \Rightarrow p \supset q, [p \Rightarrow r, [p \Rightarrow q]]$$

As discussed in the section on the nested sequent formalism (Section 2.7), propagation and reachability rules play a crucial role in the formulation

$$\frac{}{\mathcal{R}, \Gamma, w : p \Rightarrow u : p, \Delta} (r_{id})$$

$$\frac{\mathcal{R}, \Gamma, w : A \supset B \Rightarrow u : A, \Delta \qquad \mathcal{R}, \Gamma, w : A \supset B, u : B \Rightarrow \Delta}{\mathcal{R}, \Gamma, w : A \supset B \Rightarrow \Delta} (p_{\supset_l})$$

Side conditions: Both rules are applicable only if $w \rightsquigarrow_{\mathcal{R}} u$.

Figure 15: Reachability rules for $\mathsf{L}(\mathsf{IL})$.

of nested sequent calculi. As we aim to transform labeled proofs in $\mathsf{L}(\mathsf{IL})$ into nested sequent proofs in $\mathsf{N}(\mathsf{IL})$, we must define reachability rules in the context of labeled sequents. Toward this end, we define *directed paths* in labeled sequents accordingly.

Definition 3.13 (Directed Path [70]). Let $\Lambda = \mathcal{R}, \Gamma \Rightarrow \Delta$ be a labeled sequent. We say that there exists a *directed path* from w to u in $\mathcal{R}$ (written $w \rightsquigarrow_{\mathcal{R}} u$) *iff* $w = u$, or there exist labels v_i (with $i \in \{1, \ldots, n\}$) such that $w \leq v_1, \ldots, v_n \leq u \in \mathcal{R}$ (we stipulate that $w \leq u \in \mathcal{R}$ when $n = 0$).

Directed paths are employed in the formulation of the labeled reachability rule (r_{id}) and the labeled propagation rule $(p_{\supset_l})$, shown in Figure 15 and based on the work of [70, 73]. As the lemma below demonstrates, by adding these rules to $\mathsf{L}(\mathsf{IL})$, the structural rules (ref) and (tra) become eliminable. Since analogs of these structural rules do not exist in $\mathsf{N}(\mathsf{IL})$, showing their eliminability is a crucial step in translating proofs from the labeled setting to the (nested) sequent setting, as discussed later on.

Lemma 3.14. *(ref) and (tra) are eliminable in $\mathsf{L}(\mathsf{IL}) + \{(r_{id}), (p_{\supset_l})\}$.*

Proof: We argue the eliminability of both rules by induction on the height of the given derivation.

Base case. We first argue the (ref) case. Note that (ref) is freely permutable above (id), except when the principal relational atom is auxiliary in (ref). This case is resolved by making use of the (r_{id}) propagation rule as shown below, where the side condition is satisfied since $w \rightsquigarrow_{\mathcal{R}} u$ holds (taking w and u to be equal).

$$\dfrac{\overline{\mathcal{R}, w \leq w, \Gamma, w : p \Rightarrow w : p, \Delta}\ (id)}{\mathcal{R}, \Gamma, w : p \Rightarrow w : p, \Delta}\ (ref) \quad \rightsquigarrow \quad \overline{\mathcal{R}, \Gamma, w : p \Rightarrow w : p, \Delta}\ (r_{id})$$

Similar to the (ref) case, the only non-trivial case of permuting (tra) above (id) is when the principal relational atom of (id) is auxiliary in (tra). Observe that the conclusion of the proof shown below left is an instance of (r_{id}) because $w \rightsquigarrow_{\mathcal{R}} v$ holds. Thus, the proof can be replaced by the instance of (r_{id}) as shown below right.

$$\dfrac{\overline{\mathcal{R}, w \leq u, u \leq v, w \leq v, \Gamma, w : p \Rightarrow v : p, \Delta}\ (id)}{\mathcal{R}, w \leq u, u \leq v, \Gamma, w : p \Rightarrow v : p, \Delta}\ (tra) \quad \rightsquigarrow$$

$$\overline{\mathcal{R}, w \leq u, u \leq v, \Gamma, w : p \Rightarrow v : p, \Delta}\ (r_{id})$$

Inductive step. With the exception of the $(\supset_l)$ rule, (ref) and (tra) freely permute above every rule of $\mathsf{L(IL)}$. Below, we show how to resolve the non-trivial cases where the relational atom principal in $(\supset_l)$ is auxiliary in (ref) or (tra). In the (ref) case below, observe that $(p_{\supset_l})$ can be applied after (ref) since $w \rightsquigarrow_{\mathcal{R}} w$ holds.

$$\dfrac{\dfrac{\mathcal{R}, w \leq w, \Gamma, w : A \supset B \Rightarrow w : A, \Delta \qquad \mathcal{R}, w \leq w, \Gamma, w : A \supset B, w : B \Rightarrow \Delta}{\mathcal{R}, w \leq w, \Gamma, w : A \supset B \Rightarrow \Delta}\ (\supset_l)}{\mathcal{R}, \Gamma, w : A \supset B \Rightarrow \Delta}\ (ref)$$

The above inference may be simulated with $(p_{\supset_l})$ as shown below:

$$\mathcal{D} \;=\; \dfrac{\mathcal{R}, w \leq w, \Gamma, w : A \supset B \Rightarrow w : A, \Delta}{\mathcal{R}, \Gamma, w : A \supset B \Rightarrow w : A, \Delta}\ (ref)$$

$$\dfrac{\mathcal{D} \qquad \dfrac{\mathcal{R}, w \leq w, \Gamma, w : A \supset B, w : B \Rightarrow \Delta}{\mathcal{R}, \Gamma, w : A \supset B \Rightarrow w : A, \Delta}\ (ref)}{\mathcal{R}, \Gamma, w : A \supset B \Rightarrow \Delta}\ (p_{\supset_l})$$

Let us consider the non-trivial (tra) case below:

$$\Lambda = \mathcal{R}, w \leq u, u \leq v, w \leq v, \Gamma, w : A \supset B \Rightarrow v : A, \Delta$$

$$\cfrac{\cfrac{\Lambda \qquad \mathcal{R}, w \leq u, u \leq v, w \leq v, \Gamma, w : A \supset B, v : B \Rightarrow \Delta}{\mathcal{R}, w \leq u, u \leq v, w \leq v, \Gamma, w : A \supset B \Rightarrow \Delta}\ (\supset_l)}{\mathcal{R}, w \leq u, u \leq v, \Gamma, w : A \supset B \Rightarrow \Delta}\ (tra)$$

Observe that $(p_{\supset_l})$ can be applied after applying (tra) since $w \rightsquigarrow_{\mathcal{R}} v$ holds.

$$\mathcal{D} \ = \ \cfrac{\mathcal{R}, w \leq u, u \leq v, w \leq v, \Gamma, w : A \supset B \Rightarrow v : A, \Delta}{\mathcal{R}, w \leq u, u \leq v, \Gamma, w : A \supset B \Rightarrow w : A, \Delta}\ (tra)$$

$$\cfrac{\mathcal{D} \qquad \cfrac{\mathcal{R}, w \leq u, u \leq v, w \leq v, \Gamma, v : A \supset B, w : B \Rightarrow \Delta}{\mathcal{R}, w \leq u, u \leq v, \Gamma, w : A \supset B \Rightarrow w : A, \Delta}\ (tra)}{\mathcal{R}, w \leq u, u \leq v, \Gamma, w : A \supset B \Rightarrow \Delta}\ (p_{\supset_l})$$

Thus, the structural rules (ref) and (tra) are eliminable from every proof in $\mathsf{L}(\mathsf{IL}) + \{(r_{id}), (p_{\supset_l})\}$. $\square$

A consequence of the above elimination result is that $\mathsf{L}'(\mathsf{IL}) := \mathsf{L}(\mathsf{IL}) + \{(r_{id}), (p_{\supset_l})\} - \{(ref), (tra)\}$ serves as a complete calculus for intuitionistic logic. Moreover, the calculus has the interesting property that every proof of a theorem is a labeled tree derivation; the proof of the following lemma is similar to the proof of Lemma 3.3.

LEMMA 3.15. *Every derivation of a formula A in $\mathsf{L}'(\mathsf{IL})$ is a labeled tree derivation.*

By means of the above lemmata, we may translate every proof of a theorem A in $\mathsf{L}(\mathsf{IL})$ into a proof of the theorem in $\mathsf{N}(\mathsf{IL})$. The translation is explained in the lemma given below and completes part (1) of the translation from $\mathsf{L}(\mathsf{IL})$ to $\mathsf{S}(\mathsf{IL})$.

LEMMA 3.16. *Every proof of a formula A in $\mathsf{L}(\mathsf{IL})$ can be step-wise translated into a proof of A in $\mathsf{N}(\mathsf{IL})$.*

PROOF: By Lemma 3.14 above, we know that every proof of a formula A in $\mathsf{L}(\mathsf{IL})$ can be transformed into a proof that is free of (ref) and (tra) inferences. By Lemma 3.15, we know that this proof is a labeled tree derivation, and therefore, every labeled sequent in the proof can be translated via n into a nested sequent. One can show by induction on the height of the given proof that every rule directly translates to the corresponding rule in $\mathsf{N}(\mathsf{IL})$, though with (r_{id}) translating to (id) and $(p_{\supset_l})$ translating to $(\supset_l)$. □

There are two methods by which nested sequent proofs in $\mathsf{N}(\mathsf{IL})$ can be transformed into sequent proofs in $\mathsf{S}(\mathsf{IL})$. The first method, discussed in [96], shows how proofs within nested calculi of a suitable shape can be directly transformed into sequent calculus proofs. Alternatively, the second method [76] explains a *linearization technique*, which first transforms nested sequent proofs into linear nested sequent proofs, which are then transformable into sequent calculus proofs. Both methods rely on restructuring nested sequent proofs by means of rule permutations and shedding the extraneous treelike structure inherent in nested sequents to obtain a proof in a sequent calculus. As the details of these procedures are tedious and involved, we omit them from the presentation and refer the interested reader to the papers [96] and [76], noting that these methods imply the following lemma.

LEMMA 3.17. *Every proof of a formula A in* $\mathsf{N}(\mathsf{IL})$ *can be step-wise translated into a proof of A in* $\mathsf{S}(\mathsf{IL})$.

3.4. Translating proofs for conditional logic

In this section, we discuss translations between two sequent-style calculi for the conditional logic V (introduced in Section 1.3). The first calculus is a labeled sequent calculus, dubbed $\mathsf{G3V}$, and consists of the rules shown in Figure 16 along with the initial sequents and propositional rules for $\vee$, $\wedge$, and $\rightarrow$ from the labeled sequent calculus $\mathsf{G3K}$ for the modal logic K (see [94] for these latter rules). We remark that this labeled sequent calculus was introduced in [41].

$$\frac{x \in a, x : A, \Gamma \Rightarrow \Delta}{a \Vdash^{\exists} A, \Gamma \Rightarrow \Delta} \;(\Vdash^{\exists}_{l}) \qquad \frac{x \in a, \Gamma \Rightarrow \Delta, x : A, a \Vdash^{\exists} A}{x \in a, \Gamma \Rightarrow \Delta, a \Vdash^{\exists} A} \;(\Vdash^{\exists}_{r})$$

$$\frac{a \in S(x), a \Vdash^{\exists} B, \Gamma \Rightarrow \Delta, a \Vdash^{\exists} A}{\Gamma \Rightarrow \Delta, x : A \preccurlyeq B} \;(\preccurlyeq_{r})$$

$$\frac{a \in S(x), x : A \preccurlyeq B, \Gamma \Rightarrow \Delta, a \Vdash^{\exists} B \qquad a \Vdash^{\exists} A, a \in S(x), x : A \preccurlyeq B, \Gamma \Rightarrow \Delta}{a \in S(x), x : A \preccurlyeq B, \Gamma \Rightarrow \Delta} \;(\preccurlyeq_{l})$$

$$\frac{x \in a, a \subseteq b, x \in b, \Gamma \Rightarrow \Delta}{x \in a, a \subseteq b, \Gamma \Rightarrow \Delta} \;(\subseteq_{l})$$

$$\frac{a \subseteq b, a \in S(x), b \in S(x), \Gamma \Rightarrow \Delta \qquad b \subseteq a, a \in S(x), b \in S(x), \Gamma \Rightarrow \Delta}{a \in S(x), b \in S(x), \Gamma \Rightarrow \Delta} \;(\mathsf{nes})$$

Side conditions: Label x must be fresh in $(\Vdash^{\exists}_{l})$, and label a must be fresh in $(\preccurlyeq_{r})$.

Figure 16: Some labeled calculus rules for the conditional logic V.

The labeled calculus $\mathsf{G3V}$ uses two sorts of labels: $\mathsf{WLab} := \{x, y, z, \dots\}$ for worlds and $\mathsf{SLab} := \{a, b, c, \dots\}$ for spheres. We define a *labeled formula* to be an expression of the form $x : A$ or $a \Vdash^{\exists} A$ with $x \in \mathsf{WLab}$ and $a \in \mathsf{SLab}$. Given a sphere model, labeled formulae of the form $x : A$ and $a \Vdash^{\exists} A$ are interpreted as $x \vDash A$ and $a \vDash^{\exists} A$ in the model, respectively. We define a *relational atom* to be an expression of the form $x \in a$, $a \in S(x)$, or $a \subseteq b$ with $x \in \mathsf{WLab}$ and $a, b \in \mathsf{SLab}$. A *labeled sequent* is an expression of the form $\Gamma \Rightarrow \Delta$ such that Γ and Δ are finite multisets of relational atoms and labeled formulae.

The second sequent-style calculus we consider in this section is the *structured calculus* $\mathcal{I}^{\mathsf{i}}_{\mathsf{V}}$ [40], which is a kind of nested sequent calculus[12]. Sequents in this calculus make use of a special structure called a *block*, which is an expression of the form $[\Sigma \lhd B]$ such that Σ, B is a multiset of conditional formulae. In this setting, a *sequent* is an expression $\Gamma \Rightarrow \Delta$, where Γ is a multiset of conditional formulae, and Δ is a multiset of conditional formulae and blocks. The *formula interpretation* $\iota(\Gamma \Rightarrow \Delta, [\Sigma_1 \lhd B_1], \dots, [\Sigma_n \lhd B_n])$ of a sequent is taken to be equal to the following formula:

$$\bigwedge \Gamma \to \bigvee \Delta' \vee \bigvee_{1 \leq i \leq n} \bigvee_{A \in \Sigma_i} (A \preccurlyeq B_i)$$

Some interesting rules from the structured sequent calculus $\mathcal{I}^{\mathsf{i}}_{\mathsf{V}}$ are presented in Figure 17; see [40] for the full list of rules.

We remark that the translation between the labeled and structured calculi is not straightforward. The non-triviality of translating proofs between the two systems not only arises from the fact that both systems use a different language, but also from the fact that there is no direct correspondence between the relevant rules of the two calculi. In the following, we first discuss the translation from the structured calculus to the labeled one, and afterward, we discuss the reverse translation. See [41] for a formal and complete description of both translations.

[12]The i in $\mathcal{I}^{\mathsf{i}}_{\mathsf{V}}$ stands for 'invertible', as in [40] a version of the same proof system with less invertible rules is also introduced. In $\mathcal{I}^{\mathsf{i}}_{\mathsf{V}}$, the only non-invertible rule is jump (see Figure 17).

$$\frac{\Gamma \Rightarrow \Delta, [A \lhd B]}{\Gamma \Rightarrow \Delta, A \preccurlyeq B}\ (\preccurlyeq^{\mathsf{i}}_{r})$$

$$\frac{\Gamma, A \preccurlyeq B \Rightarrow \Delta, [B, \Sigma \lhd C] \qquad \Gamma, A \preccurlyeq B \Rightarrow \Delta, [\Sigma \lhd A], [\Sigma \lhd C]}{\Gamma, A \preccurlyeq B \Rightarrow \Delta, [\Sigma \lhd C]}\ (\preccurlyeq^{\mathsf{i}}_{l})$$

$$\frac{\Gamma \Rightarrow \Delta, [\Sigma_1, \Sigma_2 \lhd A], [\Sigma_2 \lhd B] \qquad \Gamma \Rightarrow \Delta, [\Sigma_1 \lhd A], [\Sigma_1, \Sigma_2 \lhd B]}{\Gamma \Rightarrow \Delta, [\Sigma_1 \lhd A], [\Sigma_2 \lhd B]}\ (\mathsf{com}^{\mathsf{i}})$$

$$\frac{A \Rightarrow \Sigma}{\Gamma \Rightarrow \Delta, [\Sigma \lhd A]}\ (\mathsf{jump})$$

Figure 17: Some structured calculus rules for the conditional logic V.

3.4.1. From structured sequents to labeled sequents

We illustrate the translation from the structured calculus to the labeled calculus. We adopt the following notational convention: given multisets of formulae $\Gamma = \{A_1, \dots, A_m\}$ and $\Sigma = \{D_1, \dots, D_k\}$, we shall write Γ^x and $a \Vdash^{\exists} \Sigma$ as abbreviations for $x : A_1, \dots, x : A_m$ and $a \Vdash^{\exists} D_1, \dots, a \Vdash^{\exists} D_k$ respectively. To illustrate the translation, consider a sequent of the following shape, with Γ, Δ multisets of formulae:

$$S = \Gamma \Rightarrow \Delta, [\Sigma_1 \lhd B_1], \dots, [\Sigma_n \lhd B_n]$$

Then, fix as parameters a world label x and a set of sphere labels $\bar{a} = a_1, \dots, a_n$. The translation $t(S)^{x,\bar{a}}$ of S is the following labeled sequent:

$$\begin{array}{rcl} t(S)^{x,\bar{a}} & := & a_1 \in S(x), .., a_n \in S(x),\ a_1 \Vdash^{\exists} B_1, .., a_n \Vdash^{\exists} B_n, \\ & & \Gamma^x \Rightarrow \Delta^x, a_1 \Vdash^{\exists} \Sigma_1, .., a_n \Vdash^{\exists} \Sigma_n \end{array}$$

The idea is that for each block $[\Sigma_i \lhd B_i]$ we introduce a new sphere label a_i such that $a_i \in S(x)$, and formulae $a_i \Vdash^{\exists} B_i$ in the antecedent and $a_i \Vdash^{\exists} \Sigma_i$ in the consequent. These formulae correspond to the semantic condition for a block i.e., a disjunction of $\preccurlyeq$ formulae in sphere models.

We can then define a formal translation of any derivation $\mathcal{D}$ of a sequent S in the structured calculus $\mathcal{I}^{\mathsf{i}}_{V}$ to a derivation $\{\mathcal{D}\}^{x,\bar{a}}$ in the labeled calculus G3V of the translated sequent $t(S)^{x,\bar{a}}$. Some cases of the translation are reported in Figure 18.

The most interesting case is the translation of the rule $(\mathsf{com}^{\mathsf{i}})$. Since this rule encodes sphere nesting, it is worth noticing that its translation requires the (nes)-rule applied to $t(\Gamma \Rightarrow \Delta, [\Sigma_1 \lhd A], [\Sigma_2 \lhd B])^{x,\bar{a},b,c}$, which is derived by (nes) from the two sequents:

$$\begin{array}{l} b \subseteq c, b \Vdash^{\exists} A, c \Vdash^{\exists} B, t(\Gamma)^{x,\bar{a}} \Rightarrow t(\Delta)^{x,\bar{a}}, b \Vdash^{\exists} \Sigma_1, c \Vdash^{\exists} \Sigma_2 \\ c \subseteq b, b \Vdash^{\exists} A, c \Vdash^{\exists} B, t(\Gamma)^{x,\bar{a}} \Rightarrow t(\Delta)^{x,\bar{a}}, b \Vdash^{\exists} \Sigma_1, c \Vdash^{\exists} \Sigma_2 \end{array}$$

Thus, the $(\mathsf{com}^{\mathsf{i}})$ rule can be 'mimicked' using the (nes) rule. Moreover, the translation uses the following rule $(\mathsf{mon}\exists)$, admissible in G3V:

$$\frac{b \subseteq a, \Gamma \Rightarrow \Delta, a \Vdash^{\exists} A, b \Vdash^{\exists} A}{b \subseteq a, \Gamma \Rightarrow \Delta, a \Vdash^{\exists} A}\ (\mathsf{mon}\exists)$$

This rule propagates (in a backward semantic reading) a false $\Vdash^{\exists}$-statement from a larger to a smaller neighborhood. The translation is correct:

THEOREM 3.18. *Let $\mathcal{D}$ be a derivation of a sequent S in $\mathcal{I}^{\mathsf{i}}_{\mathsf{V}}$, then $\{\mathcal{D}\}^{x,\bar{a}}$ is a derivation of $t(S)^{x,\bar{a}}$ in* G3V.

Example 3.19. As an example let us consider a derivation of $(A \preccurlyeq B) \vee (B \preccurlyeq A)$, one of the axioms of V, in $\mathcal{I}^{\mathsf{i}}_{\mathsf{V}}$:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{B \Rightarrow A, B}{\Rightarrow [A, B \lhd B], [B \lhd A]}\ (\mathsf{jump}) \qquad \dfrac{A \Rightarrow A, B}{\Rightarrow [A \lhd B], [A, B \lhd A]}\ (\mathsf{jump})}{\Rightarrow [A \lhd B], [B \lhd A]}\ (\mathsf{com}^{\mathsf{i}})}{\Rightarrow [A \lhd B], B \preccurlyeq A}\ (\preccurlyeq^{\mathsf{i}}_{r})}{\Rightarrow A \preccurlyeq B, B \preccurlyeq A}\ (\preccurlyeq^{\mathsf{i}}_{r})}{\Rightarrow (A \preccurlyeq B) \vee (B \preccurlyeq A)}\ (\vee_{r})$$

$$\left\{ \dfrac{\begin{matrix}\mathcal{D}_1 \\ \Gamma \Rightarrow \Delta, [A \lhd B]\end{matrix}}{\Gamma \Rightarrow \Delta, A \preccurlyeq B}\ (\preccurlyeq^{\mathsf{i}}_r) \right\}^{x,\bar{a}} \rightsquigarrow \dfrac{\begin{matrix}\{\mathcal{D}_1\}^{x,\bar{a}\,b} \\ t(\Gamma \Rightarrow \Delta, [A \lhd B])^{x,\bar{a}\,b}\end{matrix}}{t(\Gamma \Rightarrow \Delta, A \preccurlyeq B)^{x,\bar{a}}}\ (\preccurlyeq_r)$$

$$\left\{ \dfrac{\begin{matrix}\mathcal{D}_1 \\ \Gamma \Rightarrow \Delta, [\Sigma_1, \Sigma_2 \lhd A], [\Sigma_2 \lhd B]\end{matrix} \quad \begin{matrix}\mathcal{D}_2 \\ \Gamma \Rightarrow \Delta, [\Sigma \lhd A], [\Sigma, \Pi \lhd B]\end{matrix}}{\Gamma \Rightarrow \Delta, [\Sigma_1 \lhd A], [\Sigma_1 \lhd B]}\ (\mathsf{com}^{\mathsf{i}}) \right\}^{x,\bar{a}\,b\,c} \rightsquigarrow$$

$$\dfrac{\dfrac{\dfrac{\begin{matrix}\{\mathcal{D}_1\}^{x,\bar{a}\,b\,c} \\ t(\Gamma \Rightarrow \Delta, [\Sigma_1, \Sigma_2 \lhd A], [\Sigma_2 \lhd B])^{x,\bar{a}\,b\,c}\end{matrix}}{b \subseteq c, b \Vdash^{\exists} A, c \Vdash^{\exists} B, t(\Gamma)^{x,\bar{a}} \Rightarrow t(\Delta)^{x,\bar{a}}, b \Vdash^{\exists} \Sigma_1, b \Vdash^{\exists} \Sigma_2, c \Vdash^{\exists} \Sigma_2}\ (wk)}{b \subseteq c, b \Vdash^{\exists} A, c \Vdash^{\exists} B, t(\Gamma)^{x,\bar{a}} \Rightarrow t(\Delta)^{x,\bar{a}}, b \Vdash^{\exists} \Sigma_1, c \Vdash^{\exists} \Sigma_2}\ (\mathsf{mon}\exists) \quad \mathcal{E}}{t(\Gamma \Rightarrow \Delta, [\Sigma_1 \lhd A], [\Sigma_1 \lhd B])^{x,\bar{a}\,b\,c}}\ (\mathsf{nes})$$

In the above, $\mathcal{E}$ is a derivation of sequent $c \subseteq b, b \Vdash^{\exists} A, c \Vdash^{\exists} B, t(\Gamma)^{x,\bar{a}} \Rightarrow t(\Delta)^{x,\bar{a}}, b \Vdash^{\exists} \Sigma_1, c \Vdash^{\exists} \Sigma_2$ from the translation of the rightmost premiss of $(\mathsf{com}^{\mathsf{i}})$. $\mathcal{E}$ is constructed similarly to the displayed derivation of the premiss of (nes).

$$\left\{ \dfrac{\begin{matrix}\mathcal{D}_1 \\ x : \Sigma \Rightarrow x : A\end{matrix}}{\Gamma \Rightarrow \Delta, [\Sigma \lhd A]}\ (\mathsf{jump}) \right\}^{x,\bar{a}\,b} \rightsquigarrow$$

$$\dfrac{\dfrac{\dfrac{\begin{matrix}t\{\mathcal{D}_1\}^{x\,[x/y]} \\ t(x : \Sigma \Rightarrow x : A)^{x\,[x/y]}\end{matrix}}{y \in b, b \in S(x), y : A, t(\Gamma)^{x,\bar{a}} \Rightarrow t(\Delta)^{x,\bar{a}}, y : \Sigma, b \Vdash^{\exists} \Sigma}\ (wk)}{y \in b, b \in S(x), y : A, t(\Gamma)^{x,\bar{a}} \Rightarrow t(\Delta)^{x,\bar{a}}, b \Vdash^{\exists} \Sigma}\ (\Vdash^{\exists}_r) \times n}{t(\Gamma \Rightarrow \Delta, [\Sigma \lhd A])^{x,\bar{a}\,b}}\ (\Vdash^{\exists}_l)$$

Figure 18: Some cases of the translation from $\mathcal{I}^{\mathsf{i}}_{\mathsf{V}}$ to G3V.

The derivation above can be translated into a derivation in $\mathsf{G3V}$ as shown below. We only show the derivation of the left premise of (nes) as the other is symmetric.

$$
\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{y : B \Rightarrow y : A, y : B}{a \in S(x), b \in S(x), y \in a, y \in b, y : B, b \Vdash^{\exists} A \Rightarrow a \Vdash^{\exists} A, a \Vdash^{\exists} B, b \Vdash^{\exists} B, y : A, y : B}\,(Wk)}{a \in S(x), b \in S(x), y \in a, y \in b, y : B, b \Vdash^{\exists} A \Rightarrow a \Vdash^{\exists} A, a \Vdash^{\exists} B, b \Vdash^{\exists} B}\,(\Vdash^{\exists}_{r} \times 2)}{a \in S(x), b \in S(x), a \Vdash^{\exists} B, b \Vdash^{\exists} A \Rightarrow a \Vdash^{\exists} A, a \Vdash^{\exists} B, b \Vdash^{\exists} B}\,(\Vdash^{\exists}_{l})}{a \subseteq b, a \in S(x), b \in S(x), a \Vdash^{\exists} B, b \Vdash^{\exists} A \Rightarrow a \Vdash^{\exists} A, a \Vdash^{\exists} B, b \Vdash^{\exists} B}\,(Wk)}{a \subseteq b, a \in S(x), b \in S(x), a \Vdash^{\exists} B, b \Vdash^{\exists} A \Rightarrow a \Vdash^{\exists} A, b \Vdash^{\exists} B}\,(\mathsf{mon}\exists)}{a \in S(x), b \in S(x), a \Vdash^{\exists} B, b \Vdash^{\exists} A \Rightarrow a \Vdash^{\exists} A, b \Vdash^{\exists} B}\,(\mathsf{nes})}{a \in S(x), a \Vdash^{\exists} B \Rightarrow x : B \preccurlyeq A, a \Vdash^{\exists} A}\,(\preccurlyeq_{r})}{\Rightarrow x : A \preccurlyeq B, x : B \preccurlyeq A}\,(\preccurlyeq_{r})}{\Rightarrow x : A \preccurlyeq B \vee B \preccurlyeq A}\,(\vee_{r})
$$

3.4.2. From labeled sequents to structured sequents.

The translation from the labeled calculus $\mathsf{G3V}$ to the structured calculus $\mathcal{I}^{\mathsf{i}}_{\mathsf{V}}$ is more difficult, as not every sequent of the labeled calculus can be translated into a sequent of the structured calculus. Consequently, a derivation in $\mathsf{G3V}$ might contain steps that cannot be simulated in the calculus $\mathcal{I}^{\mathsf{i}}_{\mathsf{V}}$. In this section we only describe the general strategy behind the translation; for a formal treatment we refer the reader to [41].

More specifically, the translation only applies to labeled sequents of the form $t(\Gamma \Rightarrow \Delta)^x$ which are the image of the translation of a sequent $\Gamma \Rightarrow \Delta$ of the structured calculus $\mathcal{I}^{\mathsf{i}}_{\mathsf{V}}$. Then, since a proof of $t(\Gamma \Rightarrow \Delta)^x$ in $\mathsf{G3V}$ may involve sequents that are not translatable, the first step is to rearrange the proof in a specific *normal form*, in which rules are applied in a certain order. Then, one shows that derivations in normal form can be 'partitioned' into subderivations $\mathcal{S}$ such that, for each $\mathcal{S}$, the premisses of $\mathcal{S}$ are translatable into premisses of a rule r of $\mathcal{I}^{\mathsf{i}}_{\mathsf{V}}$, and the conclusion of $\mathcal{S}$ can be translated into the conclusion of r. Thus, the rules of the structured calculus $\mathcal{I}^{\mathsf{i}}_{\mathsf{V}}$ act as 'macros' over the rules of the labeled calculus, 'skipping' the untranslatable sequents.

We illustrate the translation of labeled sequents into the sequents with blocks of $\mathcal{I}^{\mathsf{i}}_{\mathsf{V}}$ with an example. Let S be the following sequent, where Γ, Δ

only contain formulae of the language:

$$a_1 \subseteq a_2, a_2 \subseteq a_3, a_1 \subseteq a_3, a_1 \in S(x), a_2 \in S(x), a_3 \in S(x),$$
$$a_1 \Vdash^{\exists} A_1, a_2 \Vdash^{\exists} A_2, a_3 \Vdash^{\exists} A_3, x : \Gamma \Rightarrow x : \Delta, a_1 \Vdash^{\exists} \Sigma_1, a_2 \Vdash^{\exists} \Sigma_2, a_3 \Vdash^{\exists} \Sigma_3$$

The translation of S is the following sequent with blocks:

$$\Gamma \Rightarrow \Delta, [\Sigma_1, \Sigma_2, \Sigma_3 \lhd A_1], [\Sigma_2, \Sigma_3 \lhd A_2], [\Sigma_3 \lhd A_3]$$

Intuitively, the translation re-assembles the blocks from formulae labeled with the same sphere label. Furthermore, for each inclusion $a_i \subseteq a_j$ we add to the corresponding block also formulae Σ_j such that $a_j \Vdash^{\exists} \Sigma_j$ occurs in the consequent of the labeled sequent. Thus, each block in the internal calculus consists of $\preccurlyeq$-formulae relative to some sphere i.e., labeled with the same sphere label in $\mathsf{G3V}$.

Moreover, a labeled sequent is translatable only if it has a *tree-like structure*. This tree-like structure is generated by the two spheres/worlds relations $x \to a$ iff $a \in S(x)$, the relation $a \to y$ iff $y \in a$, and their composition: $x \to y$ iff $x \to a \to y$ for some a. Intuitively, a labeled sequent $\Gamma \Rightarrow \Delta$ is tree-like if, for every label x occurring in Γ, the set of labels y reachable from x by the transitive closure of the relation $x \to y$ forms a tree[13].

Concerning the translation of a normal form derivation $\mathcal{D}$, the idea is that one first translates, starting from the root, the rules whose sequents have a translation in $\mathcal{I}^{\mathsf{i}}_{\mathsf{V}}$, until labeled sequents that cannot be translated are reached. Next, we need to deal with untranslatable (but derivable!) sequents. For this, we show that a derivable untranslatable sequent $\Gamma \Rightarrow \Delta$ can be *replaced* by a derivable translatable sequent $\Gamma_1 \Rightarrow \Delta_1$ obtained by a decomposition of $\Gamma \Rightarrow \Delta$ determined by the tree-like structure associated to every label x occurring in Γ: either $\Gamma_1 \Rightarrow \Delta_1$ is the subsequent containing only the labels in the tree rooted in x and the formulae/relation involving these labels, or it is the subsequent obtained by *removing* from $\Gamma \Rightarrow \Delta$ the labels and formulae of the tree of x, or it is obtained from the latter by

[13] To be precise, only a subset of sequents with a tree-like structure can be translated in sequents of the language of $\mathcal{I}^{\mathsf{i}}_{\mathsf{V}}$, but we avoid giving full details here.

iterating the process (on another label). If $\Gamma \Rightarrow \Delta$ is derivable, there exists a translatable subsequent $\Gamma_1 \Rightarrow \Delta_1$ of $\Gamma \Rightarrow \Delta$ which is derivable too (with the same height).

Thus, in order to define the translation of the whole derivation $\mathcal{D}$, when an untranslatable sequent $\Gamma \Rightarrow \Delta$ is reached, we consider then the translation of the sub-derivation $\mathcal{D}'$ of a subsequent $\Gamma_1 \Rightarrow \Delta_1$ obtained by decomposition of $\Gamma \Rightarrow \Delta$. Since the sequent $\Gamma_1 \Rightarrow \Delta_1$ is not determined in advance and it is not necessarily unique, the translation of $\mathcal{D}$ is not entirely *deterministic*.

3.5. A more difficult case: Translating bunched logics

Using the Kripke resource semantics of BI it is not difficult to build a labeled sequent or labeled tableau proof system. As usual, the first step is to devise a labeling algebra that reflects the properties of the semantics. The units 1, 0 and ∞ are reflected into the labels units m, a and ϖ. The semantic properties of the binary operators $\otimes$, $\oplus$ and the preodering relation $\sqsubseteq$ are reflected into the binary functors $\mathfrak{m}$, $\mathfrak{a}$ and the binary relation $\leqslant$.

Definition 3.20. A countable set L of symbols is a set of *label letters* if it is disjoint from the set $U = \{\, \mathrm{m}, \mathrm{a}, \varpi \,\}$ of *label units*. $\mathcal{L}_L^0 = L \cup U$ is the set of *atomic labels over* L. The set $\mathcal{L}_L$ of *labels over* L is defined as $\bigcup_{n\in\mathbb{N}} \mathcal{L}_L^n$ where

$$\mathcal{L}_L^{n+1} := \mathcal{L}_L^n \cup \{\, \mathfrak{r}(\ell, \ell') \mid \ell, \ell' \in \mathcal{L}_L^n \text{ and } \mathfrak{r} \in \{\, \mathfrak{m}, \mathfrak{a} \,\}\,\}.$$

A *label constraint* is an expression $\ell \leqslant \ell'$, where ℓ and ℓ' are labels. A *labeled formula* is an expression $\ell : A$, where A is a formula and ℓ is a label.

The second step is to define labeled sequents (as in GBI) of the form $\Gamma \Rightarrow \Delta$, where Γ is a multiset mixing both labeled formulae and label constraints and Δ is a multiset of labeled formulae.

The third and final step is to devise logical rules capturing the meaning of the connectives and structural rules reflecting the properties of the underlying frame. The logical rules of GBI are given in Figure 19 and are direct translations of their semantic clauses. The structural rules of GBI are given in Figure 20 where $\mathfrak{r}$ (resp. r) denotes either $\mathfrak{m}$ or $\mathfrak{a}$ (resp. m and

$$\frac{}{\Gamma, \varpi \leqslant \ell \Rightarrow A : \ell, \Delta}\ (\bot_r) \qquad \frac{}{\Gamma, A : \ell \Rightarrow A : \ell, \Delta}\ (\mathrm{id}) \qquad \frac{}{\Gamma, \mathrm{m} \leqslant \ell \Rightarrow \top_{\mathfrak{m}} : \ell, \Delta}\ (\top_{\mathfrak{m}r})$$

$$\frac{\Gamma, \varpi \leqslant \ell \Rightarrow \Delta}{\Gamma, \bot : \ell \Rightarrow \Delta}\ (\bot_l) \qquad \frac{\Gamma, \mathrm{m} \leqslant \ell \Rightarrow \Delta}{\Gamma, \top_{\mathfrak{m}} : \ell \Rightarrow \Delta}\ (\top_{\mathfrak{m}l})$$

$$\frac{\Gamma, \mathrm{a} \leqslant \ell \Rightarrow \Delta}{\Gamma, \top_{\mathfrak{a}} : \ell \Rightarrow \Delta}\ (\top_{\mathfrak{a}l}) \qquad \frac{}{\Gamma, \mathrm{a} \leqslant \ell \Rightarrow \top_{\mathfrak{a}} : \ell, \Delta}\ (\top_{\mathfrak{a}r})$$

$$\frac{\mathfrak{a}(\ell, \ell_1) \leqslant \ell_2, \Gamma, A \supset B : \ell \Rightarrow A : \ell_1, \Delta \quad \mathfrak{a}(\ell, \ell_1) \leqslant \ell_2, \Gamma, A \supset B : \ell, B : \ell_2 \Rightarrow \Delta}{\mathfrak{a}(\ell, \ell_1) \leqslant \ell_2, \Gamma, A \supset B : \ell \Rightarrow \Delta}\ (\supset_l)$$

$$\frac{\mathfrak{m}(\ell, \ell_1) \leqslant \ell_2, \Gamma, A \mathrel{-\!\!*} B : \ell \Rightarrow A : \ell_1, \Delta \quad \mathfrak{m}(\ell, \ell_1) \leqslant \ell_2, \Gamma, A \mathrel{-\!\!*} B : \ell, B : \ell_2 \Rightarrow \Delta}{\mathfrak{m}(\ell, \ell_1) \leqslant \ell_2, \Gamma, A \mathrel{-\!\!*} B : \ell \Rightarrow \Delta}\ (\mathrel{-\!\!*}_l)$$

$$\frac{\mathfrak{a}(\ell, \ell_1) \leqslant \ell_2, \Gamma, A : \ell_1 \Rightarrow B : \ell_2, \Delta}{\Gamma \Rightarrow A \supset B : \ell, \Delta}\ (\supset_r) \qquad \frac{\mathfrak{m}(\ell, \ell_1) \leqslant \ell_2, \Gamma, A : \ell_1 \Rightarrow B : \ell_2, \Delta}{\Gamma \Rightarrow A \mathrel{-\!\!*} B : \ell, \Delta}\ (\mathrel{-\!\!*}_r)$$

$$\frac{\mathfrak{a}(\ell_1, \ell_2) \leqslant \ell, \Gamma, A : \ell_1, B : \ell_2 \Rightarrow \Delta}{\Gamma, A \wedge B : \ell \Rightarrow \Delta}\ (\wedge_l) \qquad \frac{\mathfrak{m}(\ell_1, \ell_2) \leqslant \ell, \Gamma, A : \ell_1, B : \ell_2 \Rightarrow \Delta}{\Gamma, A * B : \ell \Rightarrow \Delta}\ (*_l)$$

$$\frac{\mathfrak{a}(\ell_1, \ell_2) \leqslant \ell, \Gamma \Rightarrow A : \ell_1, \Delta \quad \mathfrak{a}(\ell_1, \ell_2) \leqslant \ell, \Gamma \Rightarrow B : \ell_2, \Delta}{\mathfrak{a}(\ell_1, \ell_2) \leqslant \ell, \Gamma \Rightarrow A \wedge B : \ell, \Delta}\ (\wedge_r)$$

$$\frac{\mathfrak{m}(\ell_1, \ell_2) \leqslant \ell, \Gamma \Rightarrow A * B : \ell, A : \ell_1, \Delta \quad \mathfrak{m}(\ell_1, \ell_2) \leqslant \ell, \Gamma \Rightarrow A * B : \ell, B : \ell_2, \Delta}{\mathfrak{m}(\ell_1, \ell_2) \leqslant \ell, \Gamma \Rightarrow A * B : \ell, \Delta}\ (*_r)$$

$$\frac{\Gamma, A : \ell \Rightarrow \Delta \quad \Gamma, B : \ell \Rightarrow \Delta}{\Gamma, A \vee B : \ell \Rightarrow \Delta}\ (\vee_l) \qquad \frac{\Gamma \Rightarrow A_{\mathrm{i} \in \{1,2\}} : \ell, \Delta}{\Gamma \Rightarrow A_1 \vee A_2 : \ell, \Delta}\ (\vee_r^{\mathrm{i}})$$

Side conditions: ℓ_1 and ℓ_2 must be fresh label letters in $*_{\mathrm{L}}$, $\wedge_{\mathrm{L}}$, $\mathrel{-\!\!*}_{\mathrm{R}}$, and $\supset_{\mathrm{R}}$.

Figure 19: Logical rules of GBI.

$$\frac{\ell \leqslant \ell, \Gamma \Rightarrow \Delta}{\Gamma \Rightarrow \Delta}\ (\mathrm{R}) \qquad \frac{\ell_0 \leqslant \ell, \ell_0 \leqslant \ell_1, \ell_1 \leqslant \ell, \Gamma \Rightarrow \Delta}{\ell_0 \leqslant \ell_1, \ell_1 \leqslant \ell, \Gamma \Rightarrow \Delta}\ (\mathrm{T})$$

$$\frac{\mathfrak{a}(\ell, \ell) \leqslant \ell, \Gamma \Rightarrow \Delta}{\Gamma \Rightarrow \Delta}\ (\mathrm{I}_{\mathfrak{a}})$$

$$\frac{\mathfrak{r}(\ell, \mathrm{r}) \leqslant \ell, \Gamma \Rightarrow \Delta}{\Gamma \Rightarrow \Delta}\ (\mathrm{U}^1_{\mathfrak{r}}) \qquad \frac{\mathfrak{r}(\mathrm{r}, \ell) \leqslant \ell, \Gamma \Rightarrow \Delta}{\Gamma \Rightarrow \Delta}\ (\mathrm{U}^2_{\mathfrak{r}}) \qquad \frac{\mathfrak{r}(\ell_2, \ell_1) \leqslant \ell, \Gamma \Rightarrow \Delta}{\mathfrak{r}(\ell_1, \ell_2) \leqslant \ell, \Gamma \Rightarrow \Delta}\ (\mathrm{E}_{\mathfrak{r}})$$

$$\frac{\mathfrak{r}(\ell_3, \ell_2) \leqslant \ell_0, \mathfrak{r}(\ell_4, \ell_0) \leqslant \ell, \Gamma \Rightarrow \Delta}{\mathfrak{r}(\ell_4, \ell_3) \leqslant \ell_1, \mathfrak{r}(\ell_1, \ell_2) \leqslant \ell, \Gamma \Rightarrow \Delta}\ (\mathrm{A}^1_{\mathfrak{r}})$$

$$\frac{\mathfrak{r}(\ell_1, \ell_4) \leqslant \ell_0, \mathfrak{r}(\ell_0, \ell_3) \leqslant \ell, \Gamma \Rightarrow \Delta}{\mathfrak{r}(\ell_4, \ell_3) \leqslant \ell_2, \mathfrak{r}(\ell_1, \ell_2) \leqslant \ell, \Gamma \Rightarrow \Delta}\ (\mathrm{A}^2_{\mathfrak{r}})$$

$$\frac{\ell_i \leqslant \ell, \mathfrak{a}(\ell_1, \ell_2) \leqslant \ell, \Gamma \Rightarrow \Delta}{\mathfrak{a}(\ell_1, \ell_2) \leqslant \ell, \Gamma \Rightarrow \Delta}\ (\mathrm{P}^{\mathrm{i}}_{\mathfrak{a}}) \qquad \frac{\ell_i \leqslant \ell, \mathfrak{m}(\ell_1, \ell_2) \leqslant \ell, \Gamma \Rightarrow \Delta}{\mathfrak{m}(\ell_1, \ell_2) \leqslant \ell, \Gamma \Rightarrow \Delta}\ (\mathrm{P}^{\mathrm{i}}_{\mathfrak{m}})$$

$$\frac{\mathfrak{r}(\ell_0, \ell_2) \leqslant \ell, \ell_0 \leqslant \ell_1, \mathfrak{r}(\ell_1, \ell_2) \leqslant \ell, \Gamma \Rightarrow \Delta}{\ell_0 \leqslant \ell_1, \mathfrak{r}(\ell_1, \ell_2) \leqslant \ell, \Gamma \Rightarrow \Delta}\ (\mathrm{C}^1_{\mathfrak{r}}) \qquad \frac{\ell \leqslant \ell_1, \Gamma, A : \ell_1 \Rightarrow \Delta}{\ell \leqslant \ell_1, \Gamma, A : \ell \Rightarrow \Delta}\ (\mathrm{K}_l)$$

$$\frac{\mathfrak{r}(\ell_1, \ell_0) \leqslant \ell, \ell_0 \leqslant \ell_2, \mathfrak{r}(\ell_1, \ell_2) \leqslant \ell, \Gamma \Rightarrow \Delta}{\ell_0 \leqslant \ell_2, \mathfrak{r}(\ell_1, \ell_2) \leqslant \ell, \Gamma \Rightarrow \Delta}\ (\mathrm{C}^2_{\mathfrak{r}}) \qquad \frac{\ell_1 \leqslant \ell, \Gamma \Rightarrow A : \ell_1, \Delta}{\ell_1 \leqslant \ell, \Gamma \Rightarrow A : \ell, \Delta}\ (\mathrm{K}_r)$$

$$\frac{\Gamma_0 \Rightarrow \Delta}{\Gamma_0, \Gamma_1 \Rightarrow \Delta}\ (\mathrm{W}_l) \qquad \frac{\Gamma \Rightarrow \Delta_0}{\Gamma \Rightarrow \Delta_0, \Delta_1}\ (\mathrm{W}_r) \qquad \frac{\Gamma_0, \Gamma_1, \Gamma_1 \Rightarrow \Delta}{\Gamma_0, \Gamma_1 \Rightarrow \Delta}\ (\mathrm{C}_l)$$

$$\frac{\Gamma \Rightarrow \Delta_0, \Delta_1, \Delta_1}{\Gamma \Rightarrow \Delta_0, \Delta_1}\ (\mathrm{C}_r)$$

Side conditions:
$i \in \{1, 2\}$ and $\mathfrak{r} \in \{\mathfrak{m}, \mathfrak{a}\}$.
ℓ_0 is a fresh label letter in $\mathrm{A}^{\mathrm{i}}_{\mathfrak{r}}$. ℓ_{3-i} in $\mathrm{P}^{\mathrm{i}}_{\mathfrak{m}}$ must be in $\{\mathrm{m}, \varpi\}$.
ℓ in R and $\mathrm{I}_{\mathfrak{a}}$, ℓ_1, ℓ_2 in $\mathrm{P}^{\mathrm{i}}_{\mathfrak{a}}$ and ℓ_i in $\mathrm{P}^{\mathrm{i}}_{\mathfrak{m}}$ must occur in Γ, Δ or $\{\mathrm{m}, \mathrm{a}, \varpi\}$.

Figure 20: Structural Rules of GBI.

a) in contexts where the multiplicative or additive nature of the functor (resp. unit) is not important (*e.g.*, for properties that hold in both cases).

The structural rules R and T capture the reflexivity and transitivity of the accessibility relation. Rules $\mathrm{U}^{\mathrm{i}}_{\mathfrak{r}}$ capture the identity of the functors $\mathfrak{m}$ and $\mathfrak{a}$ w.r.t. m and a. The superscript $i \in \{1, 2\}$ in a rule name denotes which argument of an $\mathfrak{r}$-functor is treated by the rule and can be dropped if we consider the $\mathfrak{r}$-functor as implicitly commutative instead of having the explicit exchange rules $\mathrm{E}_{\mathfrak{r}}$ for commutativity. The rules $\mathrm{A}^{\mathrm{i}}_{\mathfrak{r}}$ reflect the associativity of the $\mathfrak{r}$-functors and $\mathrm{I}_{\mathfrak{a}}$ reflects the idempotency of $\oplus$ into the $\mathfrak{a}$-functor. The projection rules $\mathrm{P}^{\mathrm{i}}_{\mathfrak{a}}$ reflect into the $\mathfrak{a}$-functor the fact that $\oplus$ is increasing, *i.e.*, $w \sqsubseteq w \oplus u$. The projection rules $\mathrm{P}^{\mathrm{i}}_{\mathfrak{m}}$ capture the fact that $w \sqsubseteq w \otimes u$ only holds if u is ∞ or 1. The compatibility rules $\mathrm{C}^{\mathrm{i}}_{\mathfrak{r}}$ reflect that $\oplus$ and $\otimes$ are both order preserving.

Definition 3.21. A formula A is a theorem of GBI *iff* $\mathrm{m} \leqslant \ell \Rightarrow \ell : A$ is provable in GBI for some label letter ℓ.

Figure 21 gives an example of a proof in GBI, where the notation "..." subsumes all the elements we omit to keep the proof more concise. Let us also remark that in order to keep the proof shorter we do not explicitly represent the weakening steps before occurring before applying the axiom rule id.

3.5.1. From bunched to labeled proofs

In order to highlight the relationships between the labels and the tree structure of bunches more easily let us use label letters of the form xs where x is a non-greek letter and $s \in \{0, 1\}^*$ is a binary string that encodes the path of the node xs in a tree structure the root of which is x. Let us call x the root of a label letter xs and let us use greek letters to range over label letters with the convention that distinct greek letters denote label letters with distinct roots.

Definition 3.22. Given a bunch Γ and a label letter δ, $\mathfrak{L}(\Gamma, \delta)$, the *translation of* Γ according to δ, is defined by induction on the structure of Γ as follows:

$$\dfrac{\dfrac{}{\ell_1 : q \quad \Rightarrow \ell_1 : q}\,(\mathrm{id}) \qquad \dfrac{\dfrac{}{\ell_1 : r \quad \Rightarrow \ell_1 : r}\,(\mathrm{id})}{\ell_1 \leqslant \ell_2, \ldots, \quad \ell_1 : r, \ldots \quad \Rightarrow \ell_2 : r}\,(\mathrm{K}_r)}{\begin{array}{ll} \mathfrak{a}(\ell_1,\ell_1) \leqslant \ell_1, & \\ \ell_1 \leqslant \ell_2, \mathfrak{m}(\mathrm{m},\ell_1) \leqslant \ell_2 & \\ \mathfrak{m}(\ell_5,\ell_4) \leqslant \ell_1, \mathfrak{m}(\ell_6,\ell_4) \leqslant \ell_1, & \ell_1 : q, \\ \ell_5 \leqslant \ell_3, \ell_6 \leqslant \ell_3, & \ell_1 : q \supset r, \\ \mathfrak{a}(\ell_5,\ell_6) \leqslant \ell_3, \mathfrak{m}(\ell_3,\ell_4) \leqslant \ell_1, & \ell_5 : p \mathrel{-\!\!*} (q \supset r), \\ \mathfrak{m}(\ell_0,\ell_1) \leqslant \ell_2, \mathrm{m} \leqslant \ell_0, & \ell_6 : p \mathrel{-\!\!*} q, \ell_4 : p \quad \Rightarrow \ell_2 : r \end{array}}\,(\supset_l)$$

$$\dfrac{\begin{array}{ll} \mathfrak{a}(\ell_1,\ell_1) \leqslant \ell_1, & \\ \ell_1 \leqslant \ell_2, \mathfrak{m}(\mathrm{m},\ell_1) \leqslant \ell_2 & \\ \mathfrak{m}(\ell_5,\ell_4) \leqslant \ell_1, \mathfrak{m}(\ell_6,\ell_4) \leqslant \ell_1, & \ell_1 : q, \\ \ell_5 \leqslant \ell_3, \ell_6 \leqslant \ell_3, & \ell_1 : q \supset r, \\ \mathfrak{a}(\ell_5,\ell_6) \leqslant \ell_3, \mathfrak{m}(\ell_3,\ell_4) \leqslant \ell_1, & \ell_5 : p \mathrel{-\!\!*} (q \supset r), \\ \mathfrak{m}(\ell_0,\ell_1) \leqslant \ell_2, \mathrm{m} \leqslant \ell_0, & \ell_6 : p \mathrel{-\!\!*} q, \ell_4 : p \quad \Rightarrow \ell_2 : r \end{array}}{\begin{array}{ll} \ell_1 \leqslant \ell_2, \mathfrak{m}(\mathrm{m},\ell_1) \leqslant \ell_2 & \\ \mathfrak{m}(\ell_5,\ell_4) \leqslant \ell_1, \mathfrak{m}(\ell_6,\ell_4) \leqslant \ell_1, & \ell_1 : q, \\ \ell_5 \leqslant \ell_3, \ell_6 \leqslant \ell_3, & \ell_1 : q \supset r, \\ \mathfrak{a}(\ell_5,\ell_6) \leqslant \ell_3, \mathfrak{m}(\ell_3,\ell_4) \leqslant \ell_1, & \ell_5 : p \mathrel{-\!\!*} (q \supset r), \\ \mathfrak{m}(\ell_0,\ell_1) \leqslant \ell_2, \mathrm{m} \leqslant \ell_0, & \ell_6 : p \mathrel{-\!\!*} q, \ell_4 : p \quad \Rightarrow \ell_2 : r \end{array}}\,(\mathrm{I}_{\mathfrak{a}})$$

$$\dfrac{\cdots}{\begin{array}{ll} \mathfrak{m}(\mathrm{m},\ell_1) \leqslant \ell_2 & \\ \mathfrak{m}(\ell_5,\ell_4) \leqslant \ell_1, \mathfrak{m}(\ell_6,\ell_4) \leqslant \ell_1, & \ell_1 : q, \\ \ell_5 \leqslant \ell_3, \ell_6 \leqslant \ell_3, & \ell_1 : q \supset r, \\ \mathfrak{a}(\ell_5,\ell_6) \leqslant \ell_3, \mathfrak{m}(\ell_3,\ell_4) \leqslant \ell_1, & \ell_5 : p \mathrel{-\!\!*} (q \supset r), \\ \mathfrak{m}(\ell_0,\ell_1) \leqslant \ell_2, \mathrm{m} \leqslant \ell_0, & \ell_6 : p \mathrel{-\!\!*} q, \ell_4 : p \quad \Rightarrow \ell_2 : r \end{array}}\,(\mathrm{P}_{\mathfrak{m}})$$

$$\dfrac{\cdots}{\begin{array}{ll} \mathfrak{m}(\ell_5,\ell_4) \leqslant \ell_1, \mathfrak{m}(\ell_6,\ell_4) \leqslant \ell_1, & \ell_1 : q, \\ \ell_5 \leqslant \ell_3, \ell_6 \leqslant \ell_3, & \ell_1 : q \supset r, \\ \mathfrak{a}(\ell_5,\ell_6) \leqslant \ell_3, \mathfrak{m}(\ell_3,\ell_4) \leqslant \ell_1, & \ell_5 : p \mathrel{-\!\!*} (q \supset r), \\ \mathfrak{m}(\ell_0,\ell_1) \leqslant \ell_2, \mathrm{m} \leqslant \ell_0, & \ell_6 : p \mathrel{-\!\!*} q, \ell_4 : p \quad \Rightarrow \ell_2 : r \end{array}}\,(\mathrm{C}_{\mathfrak{m}})$$

$$\dfrac{\dfrac{}{\ell_4 : p \quad \Rightarrow \ell_4 : p}\,(\mathrm{id}) \qquad \cdots}{\begin{array}{ll} \mathfrak{m}(\ell_5,\ell_4) \leqslant \ell_1, \mathfrak{m}(\ell_6,\ell_4) \leqslant \ell_1, & \\ \ell_5 \leqslant \ell_3, \ell_6 \leqslant \ell_3, & \ell_1 : q \supset r, \\ \mathfrak{a}(\ell_5,\ell_6) \leqslant \ell_3, \mathfrak{m}(\ell_3,\ell_4) \leqslant \ell_1, & \ell_5 : p \mathrel{-\!\!*} (q \supset r), \\ \mathfrak{m}(\ell_0,\ell_1) \leqslant \ell_2, \mathrm{m} \leqslant \ell_0, & \ell_6 : p \mathrel{-\!\!*} q, \ell_4 : p \quad \Rightarrow \ell_2 : r \end{array}}\,(\mathrel{-\!\!*}_l)$$

$$\dfrac{\dfrac{}{\ell_4 : p \quad \Rightarrow \ell_4 : p}\,(\mathrm{id}) \qquad \cdots}{\begin{array}{ll} \mathfrak{m}(\ell_5,\ell_4) \leqslant \ell_1, \mathfrak{m}(\ell_6,\ell_4) \leqslant \ell_1, & \\ \ell_5 \leqslant \ell_3, \ell_6 \leqslant \ell_3, & \\ \mathfrak{a}(\ell_5,\ell_6) \leqslant \ell_3, \mathfrak{m}(\ell_3,\ell_4) \leqslant \ell_1, & \ell_5 : p \mathrel{-\!\!*} (q \supset r), \\ \mathfrak{m}(\ell_0,\ell_1) \leqslant \ell_2, \mathrm{m} \leqslant \ell_0, & \ell_6 : p \mathrel{-\!\!*} q, \ell_4 : p \quad \Rightarrow \ell_2 : r \end{array}}\,(\mathrel{-\!\!*}_l)$$

$$\dfrac{\cdots}{\begin{array}{ll} \ell_5 \leqslant \ell_3, \ell_6 \leqslant \ell_3, & \\ \mathfrak{a}(\ell_5,\ell_6) \leqslant \ell_3, \mathfrak{m}(\ell_3,\ell_4) \leqslant \ell_1, & \ell_5 : p \mathrel{-\!\!*} (q \supset r), \\ \mathfrak{m}(\ell_0,\ell_1) \leqslant \ell_2, \mathrm{m} \leqslant \ell_0, & \ell_6 : p \mathrel{-\!\!*} q, \ell_4 : p \quad \Rightarrow \ell_2 : r \end{array}}\,(\mathrm{C}_{\mathfrak{m}})$$

$$\dfrac{\cdots}{\begin{array}{ll} \mathfrak{a}(\ell_5,\ell_6) \leqslant \ell_3, \mathfrak{m}(\ell_3,\ell_4) \leqslant \ell_1, & \ell_5 : p \mathrel{-\!\!*} (q \supset r), \\ \mathfrak{m}(\ell_0,\ell_1) \leqslant \ell_2, \mathrm{m} \leqslant \ell_0, & \ell_6 : p \mathrel{-\!\!*} q, \ell_4 : p \quad \Rightarrow \ell_2 : r \end{array}}\,(\mathrm{P}_{\mathfrak{a}})$$

$$\dfrac{\cdots}{\begin{array}{lll} \mathfrak{m}(\ell_3,\ell_4) \leqslant \ell_1, & \ell_3 : p \mathrel{-\!\!*} (q \supset r) \wedge p \mathrel{-\!\!*} q, & \\ \mathfrak{m}(\ell_0,\ell_1) \leqslant \ell_2, \mathrm{m} \leqslant \ell_0, & \ell_4 : p & \Rightarrow \ell_2 : r \end{array}}\,(\wedge_l)$$

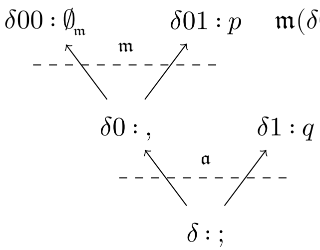
$$\dfrac{\dfrac{\cdots}{\mathfrak{m}(\ell_0,\ell_1) \leqslant \ell_2, \mathrm{m} \leqslant \ell_0, \quad \ell_1 : (p \mathrel{-\!\!*} (q \supset r) \wedge p \mathrel{-\!\!*} q) * p \quad \Rightarrow \ell_2 : r}\,(*_l)}{\mathrm{m} \leqslant \ell_0 \quad \Rightarrow \ell_0 : ((p \mathrel{-\!\!*} (q \supset r) \wedge p \mathrel{-\!\!*} q) * p) \mathrel{-\!\!*} r}\,(\mathrel{-\!\!*}_r)$$

Figure 21: GBI-proof of $((p \mathrel{-\!\!*} (q \supset r) \wedge p \mathrel{-\!\!*} q) * p) \mathrel{-\!\!*} r$.

- $\mathfrak{L}(A,\delta) = \{\, \delta : A \,\}$, $\mathfrak{L}(\emptyset_{\mathfrak{a}},\delta) = \{\, \mathrm{a} \leqslant \delta \,\}$, $\mathfrak{L}(\emptyset_{\mathfrak{m}},\delta) = \{\, \mathrm{m} \leqslant \delta \,\}$,
- $\mathfrak{L}((\Delta_0 \,,\Delta_1),\delta) = \mathfrak{L}(\Delta_0,\delta 0) \cup \mathfrak{L}(\Delta_1,\delta 1) \cup \{\, \mathfrak{m}(\delta 0,\delta 1) \leqslant \delta \,\}$,
- $\mathfrak{L}((\Delta_0 \,;\Delta_1),\delta) = \mathfrak{L}(\Delta_0,\delta 0) \cup \mathfrak{L}(\Delta_1,\delta 1) \cup \{\, \mathfrak{a}(\delta 0,\delta 1) \leqslant \delta \,\}$.

Given a sequent $\Gamma \Rightarrow A$, $\mathfrak{L}(\Gamma \Rightarrow A,\delta)$ is defined as $\mathfrak{L}(\Gamma,\delta) \Rightarrow \delta : A$.

We write $\delta : \Gamma$ as a shorthand for $\mathfrak{L}(\Gamma,\delta)$ so that $\mathfrak{L}(\Gamma \Rightarrow A,\delta) = \delta : \Gamma \Rightarrow \delta : A$. The following is an illustration of Definition 3.22:



$$\begin{array}{c}
\mathfrak{m}(\delta 00,\delta 01) \leqslant \delta 0, \mathfrak{a}(\delta 0,\delta 1) \leqslant \delta, \delta 01 : p, \delta 1 : q \Rightarrow \delta : r \\
\uparrow \\
\mathfrak{a}(\delta 0,\delta 1) \leqslant \delta, \delta 0 : (\emptyset_{\mathfrak{m}} \,, p), \delta 1 : q \Rightarrow \delta : r \\
\uparrow \\
\delta : ((\emptyset_{\mathfrak{m}} \,, p) \,; q) \Rightarrow \delta : r
\end{array}$$

As a second example, the translation of the sequent $(p \mathrel{-\!\!*} (q \supset r) \,; p \mathrel{-\!\!*} q) \,, p \Rightarrow r$, which is the premiss of the $\wedge_l$ rule in the LBI-proof presented in Example 2.1, results in the following labeled sequent:

$$\mathfrak{a}(\delta 00,\delta 01) \leqslant \delta 0, \mathfrak{m}(\delta 0,\delta 1) \leqslant \delta, \delta 01 : p \mathrel{-\!\!*} (q \supset r), \delta 00 : p \mathrel{-\!\!*} q, \delta 1 : p \;\Rightarrow \delta : r$$

Using Definition 3.22 it is not particularly difficult to translate LBI-proofs into GBI-proofs and we have the following result:

THEOREM 3.23. *If a sequent $\Gamma \Rightarrow A$ is provable in* LBI*, then for any label letter δ, the labeled sequent $\delta : \Gamma \Rightarrow \delta : A$ is provable in* GBI*.*

PROOF: The proof is by induction on the structure of an LBI-proof using an appropriate definition of label substitutions. See [34] for details. □

3.5.2. From labeled to bunched proofs

Trying to translate labeled GBI-proofs into bunched LBI-proofs is much harder than the opposite way and is currently only known for a subclass of GBI-proofs satisfying a conjunction of specific conditions called *the tree* *property*. Since describing the tree property in full technical details is out of the scope of this paper (see [36] for details), we should focus here on giving an intuitive account of its content and discuss the related issues. Let us also mention that the tree property arises from a careful inspection of the proof of Theorem 3.23 which shows that in all sequents of a translated LBI-proof, the label constraints of GBI-sequent describe a tree structure which allows the reconstruction of a bunch from the label of the formula on its right-hand side.

Let us write $\mathfrak{B}(s, \ell)$ the function that translates a label sequent s to a bunch using the label ℓ (required to occur in s) as its reference point. The result $s@l$ of $\mathfrak{B}(s, \ell)$ is called "the bunch translation of s at ℓ". For conciseness, we shall omit s when clear from the context. Let us finally define $\mathfrak{B}(s)$ as $\mathfrak{B}(s, \ell)$ where ℓ is the label of the formula on the right-hand side of s. In the light of Definition 3.22, let us make an intuitive attempt at algorithmically defining $\mathfrak{B}(s, \ell)$.

Definition 3.24. $\mathfrak{B}(s, \ell)$, the translation of a GBI labeled sequent s at label ℓ, recursively constructs a bunch from the label-constraints in s as follows:

- if $\mathfrak{m}(\ell_i, \ell_j) \leqslant \ell \in s$, then $\mathfrak{B}(\ell) = (\mathfrak{B}(\ell_i)\,, \mathfrak{B}(\ell_j))$
- if $\mathfrak{a}(\ell_i, \ell_j) \leqslant \ell \in s$, then $\mathfrak{B}(\ell) = (\mathfrak{B}(\ell_i)\,; \mathfrak{B}(\ell_j))$
- if $\ell' \leqslant \ell \in s$ and $\ell' \in \mathcal{L}^0$, then $\mathfrak{B}(\ell) = \mathfrak{B}(\ell')$
- otherwise $\mathfrak{B}(\mathrm{m}) = \emptyset_{\mathfrak{m}}$, $\mathfrak{B}(\mathrm{a}) = \emptyset_{\mathfrak{a}}$, $\mathfrak{B}(\ell) = \wedge\{\, A \mid \ell : A \,\}$ with $\wedge\emptyset = \top_{\mathfrak{a}}$

The translation described in Definition 3.24 is illustrated in Example 3.25. Remark that label constraints of the form $\ell' \leqslant \ell$, where ℓ' is an atomic label, act as "jumps" that move the reference point from one label to another (such "jumps" are in fact reduction orderings described more precisely in [34]).

Example 3.25. Let s be the labeled sequent which is the premiss of the $\wedge_l$ rule in the GBI-proof presented in Figure 21. The starting point of $\mathfrak{B}(s)$ is ℓ_2, the label occurring of the right-hand side of s. The translation proceeds as described in the following table, where the comment on the right-hand side at Step n is the justification of the result obtained on the left-hand side of Step $n+1$.

1.	$\mathfrak{B}(\ell_2) \Rightarrow r$	$\mathfrak{m}(\ell_0, \ell_1) \leqslant \ell_2,$
2.	$(\mathfrak{B}(\ell_0), \mathfrak{B}(\ell_1)) \Rightarrow r$	$\mathrm{m} \leqslant \ell_0$
3.	$(\mathfrak{B}(\mathrm{m}), \mathfrak{B}(\ell_1)) \Rightarrow r$	$\mathfrak{B}(\mathrm{m}) = \emptyset_{\mathrm{m}}$
4.	$(\emptyset_{\mathrm{m}}, \mathfrak{B}(\ell_1)) \Rightarrow r$	$\mathfrak{m}(\ell_3, \ell_4) \leqslant \ell_1$
5.	$(\emptyset_{\mathrm{m}}, (\mathfrak{B}(\ell_3), \mathfrak{B}(\ell_4)) \Rightarrow r$	$\ell_3 : p \mathrel{-\!\!*} (q \supset r) \wedge p \mathrel{-\!\!*} q$
6.	$(\emptyset_{\mathrm{m}}, (p \mathrel{-\!\!*} (q \supset r) \wedge p \mathrel{-\!\!*} q, \mathfrak{B}(\ell_4)) \Rightarrow r$	$\ell_4 : p$
7.	$(\emptyset_{\mathrm{m}}, (p \mathrel{-\!\!*} (q \supset r) \wedge p \mathrel{-\!\!*} q, p)) \Rightarrow r$	stop

Unfortunately, the translation in Definition 3.24 only works when all of the sequents in a GBI-proof satisfy the tree property. As shown in [34], all LBI-translated GBI-proofs satisfy the tree property, but at the cost of the flexibility of the labeled proof system in full generality. Indeed, translating contraction and weakening steps requires contrived labeled versions of the contraction and weakening rules that preserve the tree structure. For instance, the tree-preserving contraction rule looks like this (the subtree root at $\delta s : \Theta$ is duplicated into two new subtrees $\delta s0 : \Theta$ and $\delta s1 : \Theta$ and linked as children of the old subtree):

$$\frac{\delta : \Gamma(\mathfrak{a}(\delta s0, \delta s1) \leqslant \delta s, \delta s0 : \Theta, \delta s1 : \Theta) \Rightarrow \delta : A}{\delta : \Gamma(\delta s : \Theta) \Rightarrow \delta : A}\,(\mathrm{C_T})$$

Figure 21 and Example 2.1 respectively are GBI- and LBI-proofs of the same formula. Comparing both proofs, we notice that they share the same logical proof plan, more precisely, they decompose the same logical connectives in the same order. However, the GBI-proof does not use any of the tree-preserving rules of GBI and thus does not correspond to an LBI-translated GBI-proof. Translating the LBI-proof would require the tree-preserving contraction rule discussed previously to perform the contraction step above the $\wedge_l$ rule. Such tree-preserving rules are very restrictive, do not mimic the semantics and would not be naturally devised in a conventional labeled system. Although conventional GBI structural rules such as weakening, contraction and idempotency can easily break the tree property, they also allow more flexibility in the labeled proofs. For example, let s be the sequent that is the premiss of the $\mathrm{I}_{\mathfrak{a}}$ rule in the GBI-proof depicted in Figure 21. Trying to compute $\mathfrak{B}(s, \ell_2)$ would fail for the following reasons:

(1) We have several distinct label-constraints with the same root (*i.e.*, with the same label on the right-hand side). For instance, we have $\ell_1 \leqslant \ell_2$ and $\mathfrak{m}(\ell_0, \ell_1) \leqslant \ell_2$. Should we "jump" from ℓ_2 to ℓ_1 or should we recursively translate $\mathfrak{B}(\ell_0)$ and $\mathfrak{B}(\ell_1)$? We could define a strategy for deterministically choosing between distinct label-constraints with the same root. A reasonable one would be to choose the label-constraint that has been introduced the more recently (as being closer to the translated sequent might be more pertinent), but it emphasizes the fact that a suitable translation should take the global structure of the labeled proof into account and not just labeled sequents locally.

(2) Anyway, whatever strategy we might come up with in the previously discussed point, the label-constraint $\mathfrak{a}(\ell_1, \ell_1) \leqslant \ell_1$ clearly does not describe a tree structure, but a cycle forcing $\mathfrak{B}(s, \ell_1)$ into an infinite loop. We could place a bound on the number of loops allowed, but then which one? It is clear that the idempotency rule $\mathrm{I}_{\mathfrak{a}}$ in GBI is related with contraction in LBI, but it is not currently clear to us how to predict the correct number of copies a bunch might need in a LBI-proof using a general GBI-proof that, on one hand, does not correspond to an LBI-translated proof and, on the other hand, does not itself need any copy.

It is currently an open problem whether a general GBI-proof can always be turned into a GBI-proof satisfying the tree property.

3.5.3. Lost in translation: Why it fails when it fails

Bunched (and resource) logics exhibit a first notable difference with intuitionistic logic and modal logics like K in that the corresponding semantics do not rely only on properties of an accessibility relation in a Kripke model, but also on world (resource) composition. In particular, since BI admits both an additive and a multiplicative composition, the relational atoms uRw are generalized into relations of the form $\mathfrak{r}(\ell_1, \ell_2) \leqslant \ell$ where $\mathfrak{r}$ is one of the binary functors $\mathfrak{a}$ or $\mathfrak{m}$. Moreover, in intuitionistic logic or modal logics like K, $\mathsf{S4}$, $\mathsf{S5}$, the semantic and the syntactic readings of a relational atom uRw coincide when interpreted in terms of ordering relations "successor" and "expanded after." More precisely, consider the rule for right implication in intuitionistic logic depicted in Figure 14. The semantic reading is that, when interpreted in a Kripke structure, u should be the successor of w w.r.t. the accessibility relation, which can be written as $w \leqslant_{\text{succ}} u$. The syntactic reading of uRw is that since A and B are labeled with w and $A \supset B$ is labeled with u, A and B are subformulae of $A \supset B$ and should therefore necessarily appear (and be expanded) after $A \supset B$ in a (shallow) proof system. In other words, the subformula interpretation induces a rule application order in a syntactic proof system, which could be written as $f(w) \leqslant_{\text{after}} f(u)$ (the formulae labeled with u must be expanded after the ones labeled with w). Notice that $\leqslant_{\text{succ}}$ and $\leqslant_{\text{after}}$ are covariant (w and u occur on the same side in both orders).

However, a key problem in BI (and in resource logics more generally) is that the syntactic and the semantic readings are contravariant and sometimes even fully lost. Indeed, if we consider the rule for the left multiplicative conjunction $*$ given in Figure 19, it is clearly seen that since A and B are subformulae of $A * B$, we syntactically have $f(\ell) \leqslant_{\text{after}} f(\ell_i)$ (for $i \in \{1, 2\}$), but we semantically have (reading $\leqslant$ as $\leqslant_{\text{succ}}$) $\mathfrak{m}(\ell_1, \ell_2) \leqslant_{\text{succ}} \ell$ (with ℓ_1 and ℓ_2 occurring on the opposite side compared with $\leqslant_{\text{after}}$). Moreover, we do not even get any relation of the form $\ell \leqslant_{\text{succ}} \ell_i$ or $\ell_i \leqslant_{\text{succ}} \ell$ at all.

The immediate consequence of losing the general connection between the syntactic subformula ordering and the semantic successor ordering is that finding an extension of (the translation in) Definition 3.24 that could work for unrestricted labeled proofs is not at all trivial and might even be impossible to achieve.

3.6. Some remarks on translations

The above translations substantiate our claim that translating up the proof-theoretic hierarchy tends to be 'easier' than translating down. In particular, we found that structural rule elimination was needed to translate labeled proofs into nested proofs for intuitionistic logic (Section 3.3). Moreover, translating labeled proofs to structured sequent proofs for conditional logics introduced non-determinism (Section 3.4) and translating labeled proofs into bunched proofs (Section 3.5) was only possible given that the labeled proof was of a 'treelike' shape. Converse translations were far simpler to obtain, e.g., translating sequent proofs into labeled proofs for intuitionistic logic (Section 3.3) and translating display proofs into labeled proofs for the tense logic Kt (Section 3.2). The sophistication required in translating proofs down the hierarchy supports the claim that formalisms higher up in the hierarchy are more expressive than those below them.

4. The internal and external distinction

In the literature, proof formalisms and calculi have been classified into *internal* or *external*.[14] Typically, a formalism or calculus is placed into one of these two classes based on the syntactic elements present within the sequents used and/or the interpretability of sequents as logical formulae. Various informal definitions have been given for 'internal' and 'external,' and are often expressed in one of two ways:

(1) Internal calculi omit semantic elements from the syntax of their sequents, whereas external calculi explicitly include semantic elements.

(2) Internal calculi are those where every sequent is interpretable as a formula in the language of the logic, whereas external calculi are those without a formula translation.

For example, hypersequent and nested calculi are often considered internal since their sequents are (usually) interpretable as logical formulae [86]. On the other hand, labeled calculi are often classified as external as they incorporate semantic information in their syntax and labeled sequents exist

[14] As discussed below, the distinction between internal and external systems is rather vague. Some interpretations of this distinction are essentially the same as the distinction between semantically polluted and syntactically pure proof systems; cf. [104, 100].

which resist interpretation as logical formulae [19]. We remark that sometimes extra machinery is inserted into a proof calculus for 'bureaucratic' reasons (e.g., to correctly formulate proof-search algorithms); such machinery should be ignored when considering a calculus internal or external.

A core motivation for separating formalisms/calculi into these two categories, is that internal and external formalisms/calculi are *claimed* to possess distinct advantages over one another. It has been argued that internal calculi are better suited for establishing properties such as termination, interpolation, and optimal complexity, while external calculi are more easily constructed and permit simpler proofs of completeness, cut-admissibility, and counter-model generation (from terminating proof-search). However, we will argue that a large number of such claims are false.

In this section, we delve into the internal and external distinction, and discuss two main themes. First, we attempt to formally define the notions of internal and external, arguing that each candidate definition comes with certain drawbacks, or fails to satisfy our intuitions concerning internal and external systems in some way. Second, we aim to dispel myths about the claimed properties of internal and external systems, while identifying which attributes are genuinely useful for certain applications.

4.1. Analyzing definitions of internal and external

We begin by investigating definition (1) above, where external calculi are those which incorporate 'semantic elements' into the syntax of their sequents while internal calculi are those which do not. An immediate issue that arises with this definition is that it relies on an inherently vague notion: what do we take to be a 'semantic element'? Admittedly, it seems clear that the labels and relational atoms used in labeled sequents should qualify as 'semantic elements' as such syntactic objects encode features of relational models. Yet, via the translation from labeled to nested sequents (see Definition 3.11 in Section 3), one can see that the tree structure encoded in a nested sequent also encodes features of relational models (with points in the tree corresponding to worlds and edges in the tree corresponding to the accessibility relation). Similarly, the components of linear nested sequents and hypersequents directly correspond to worlds in relational models with the linear nested structure '$\sslash$' and the hypersequent bar '$|$' encoding features of the accessibility relation (cf. [67, 62]). It seems that (linear) nested sequents and hypersequents should qualify as external systems then, contrary to the fact that such systems are almost always counted as internal. As another example, 'semantic elements' are encoded in the language of the sequent calculi used for hybrid modal logics [8]. Yet, many would qualify such proofs systems as internal since their sequents are straightforwardly interpretable as formulae in the language of the logic; in fact, it is the incorporation of 'semantic elements' that allows this.

The issue with the first proposed definition is that it is too vague to properly distinguish between internal and external systems as the concept of a 'semantic element' is too vague. Thus, we find that definition (1) is unsuitable for distinguishing internality and externality.

Let us now investigate definition (2) above, where internal calculi are qualified as those with sequents interpretable as formulae in the language of the logic, and external calculi are those for which this property does not hold. A couple of questions come to the fore when we consider this definition. First, what does it mean for a sequent to be *interpretable* as a formula? For instance, in the context of display calculi for modal and tense logics [56], display sequents are naturally translatable to tense formulae, yet, some of these tense formula can actually be reinterpreted as modal formulae. This shows that it is not always *prima facie* clear that a sequent in fact translates to a formula in the language of the logic. A second question is: what *properties* should such an interpretation possess?

We begin investigating these questions by considering a few examples of 'internal' systems from the literature. Our aim is to extract general underlying patterns from the examples with the goal of supplying a formal definition of 'internality' along the lines of definition (2) above. What we will find is that regardless of how we attempt to rigorously specify this definition, calculi (intuitively) recognized as 'internal' and 'external' exist which fail to satisfy the definition, thus witnessing its inadequacy.

Gentzen calculi, nested sequent calculi, and hypersequent calculi are normally characterized as internal systems. Typically, what is meant by an 'interpretation of a sequent as a formula' is a *translation* τ that maps every sequent to a (i) 'structurally similar' and (ii) 'logically equivalent' formula in the language of the logic. For instance, Gentzen sequents in $\mathsf{S}(\mathsf{CP})$, nested sequents in $\mathsf{N}(\mathsf{IL})$, and hypersequents for $\mathsf{S5}$ admit the following translations:

$$\tau(\Gamma \Rightarrow \Delta) := \bigwedge \Gamma \rightarrow \bigvee \Delta$$
$$\tau(\Gamma \Rightarrow \Delta, [\Sigma_1]_{w_1}, \ldots, [\Sigma_n]_{w_n}) := \bigwedge \Gamma \supset (\bigvee \Delta \vee \tau(\Sigma_1) \vee \cdots \vee \tau(\Sigma_n))$$
$$\tau(\Gamma_1 \Rightarrow \Delta_1 \mid \cdots \mid \Gamma_n \Rightarrow \Delta_n) := \bigvee_{1 \leq i \leq n} \Box(\bigwedge \Gamma_i \rightarrow \bigvee \Delta_i)$$

We can see that the output of every translation produces a formula that is 'structurally similar' to the input in the sense that it serves as a *homomorphism* mapping every sequent into a formula of the same shape (by replacing all structural connectives in the sequent with logical connectives). Moreover, the input and output are 'logically equivalent' by definition, i.e., a sequent is satisfied on a model of the underlying logic *iff* its output is. This indeed seems a promising candidate for formalizing the notion of 'internality,' however, let us consider the labeled sequents from $\mathsf{L}'(\mathsf{IL})$ (defined on p. 110).

As witnessed by Definition 3.11, every labeled sequent of a *treelike* shape can be interpreted as a nested sequent, and thus, by the second translation above, can be interpreted as a 'structurally similar' and 'logically equivalent' formula in the language of the logic IL. Yet, $\mathsf{L}'(\mathsf{IL})$ is permitted to use labeled sequents of a *non-treelike* shape (e.g., $w \leq u, u \leq w \Rightarrow w : A$), despite that fact that such sequents play no role in deriving theorems of IL as shown in Lemma 3.15. Since it appears that labeled sequents of a non-treelike shape do not admit a 'structurally similar' translation in the language of IL, we are forced to conclude by the above notion of internality that $\mathsf{L}'(\mathsf{IL})$ is external. Nevertheless, if we re-define $\mathsf{L}'(\mathsf{IL})$ slightly so that *only* labeled tree sequents are permitted in proofs, then $\mathsf{L}'(\mathsf{IL})$ ceases to be external and becomes internal by what was said above. Hence, the above notion of 'internality' implies that being internal or external is not a property of a formalism, but of the *language* of a calculus, that is, the set of sequents that the calculus draws from to construct proofs. We should therefore speak of internal and external *sequent languages* (i.e., the set of sequents used by a proof system) rather than internal or external calculi.

Based on the discussion above, we could identify calculi as internal or external if the sequent language of the calculus is internal or external, respectively. Nevertheless, two practical issues arise: first, confirming that a language is external is subject to the difficulty that one must confirm the non-existence of any translation mapping sequents to 'structurally similar' and 'logical equivalent' formulae. Although confirming the non-existence of such a translation is perhaps not impossible,[15] it appears to be a relatively difficult feature. Second, if we define internal or external systems relative to an internal or external sequent language, then proof systems may 'switch' from being internal or external simply based on expanding or contracting the sequent language associated with the calculus.

However, it must be conceded that the above notion of 'internal' is aligned with our intuition concerning what an internal system ought to be, and can be taken as a sufficient (but not necessary) criterion for applying the term 'internal' to a proof system. What we find to be important however, is less about whether a proof system satisfies our intuitions concerning 'internality,' and more about the existence of translations from sequents to 'structurally similar' and 'logically equivalent' formulae—a fact that will be discussed in more detail below.

4.2. Purported properties of internal and external systems

Here we consider various properties attributed to internal and external calculi, and clarify how such claims are (in)correct. For ease of presentation, we first present each claim in italics, and after, provide our perspective of the claim. Although this section is intended to dispel myths about internal and external systems, we do put forth positive applications of 'internal' calculi at the end of the section (which satisfy the sufficient criterion discussed at the end of the previous section). In particular, we explain how

[15] This has been confirmed, for instance, for the hypersequents in [90] for Łukasiewicz logic—one of the main fuzzy logics.

the existence of a translation from sequents to 'structurally similar' and 'logically equivalent' formulae can be practically leveraged in a few ways.

Internal calculi are better suited than external calculi for decidability. There are two standard methods in which decidability is obtained via proof-search in a sequent calculus, which we call (1) the brute-force method, and (2) the counter-model extraction method. In the former method, one establishes that every theorem has a proof of a certain form, and shows that only a finite number of such proofs exist. Decidability is then obtained by searching this finite space, and if a proof is found, the input is known to be valid; otherwise, the input is known to be invalid. In the latter method, one attempts to construct a proof of the input, and shows that if a proof-search fails, then a counter-model of the input can be extracted. The brute-force method is more easily applied to (analytic) Gentzen systems, which are typically characterized as internal systems. This is due to the simplicity of Gentzen sequents for which it is straightforward to establish an upper finite bound on the space of analytic derivations for a given formula. Nevertheless, external systems, e.g., those of Simpson [106], also admit decidability via the brute-force method.

When it comes to applying the counter-model extraction method, there appears to be a trade-off between using internal and external calculi. Note that this method consists of two components: (1) one must establish the termination of the proof-search procedure, and (2) one must extract a counter-model if proof-search fails. We point out that 'internal' calculi seem better for securing termination while 'external' calculi appear better suited for extracting a counter-model. First, since the sequents in internal systems tend to utilize simpler data structures, establishing the termination of proof-search tends to be more easily obtained than for external systems (which utilize more complex and difficult to control data structures). Second, extracting a counter-model from failed proof-search tends to be easier in external systems than internal systems as the former tend to encode model-theoretic information.

Nevertheless, this observation merely points out that there are trade-offs in using one type of system as opposed to another, and does not outright prove that one type of system is more advantageous than another in establishing decidability. Indeed, there are many examples of decision/proof-search algorithms for wide classes of logics based on internal systems [77, 109, 107] and external systems [50, 106, 83], so we find that this claim is not warranted.

Internal calculi are better suited than external calculi for interpolation. The method of establishing interpolation via sequent-style systems is due to Maehara [85], and was originally introduced in the context of Gentzen systems. This method has been adapted to linear nested and hypersequent systems [61], nested systems [33, 81], display systems [10], and labeled systems [60]. If one compares such works on proof-theoretic interpolation, they will find that both internal and external systems alike are used in securing interpolation properties for a logics; e.g., truly sizable classes of logics have been shown to exhibit Craig and Lyndon interpolation with both internal (viz., nested) systems [73] and external (viz., labeled) systems [60]. Therefore, the claim that internal calculi are better suited for establishing interpolation does not appear warranted.

Internal calculi are harder to find/construct then external calculi. We somewhat agree with the claim that internal calculi are more difficult to find/construct in contrast to external calculi. First, we note that it is rather straightforward to generate labeled calculi for diverse classes of logics [21, 106]. Nevertheless, techniques do exist for generating internal calculi as well. For example, numerous logics have been provided (internal) display calculi [5, 114], algorithms exist for producing sequent and hypersequent calculi from suitable Hilbert systems [18], and it is now understood how to transform certain semantic properties into nested sequent systems [72, 79] or hypersequent systems [62]. Even though such methods yield sizable classes of internal calculi, they are more involved than the method of generating labeled systems.

Cut-admissibility is more difficult to establish for internal calculi. The claim that cut-admissibility is more difficult to shown with internal calculi does not appear to be warranted. Both the labeled and display formalisms yield uniform and modular calculi for extensive classes of logics, yet, general cut-admissibility results exist for labeled calculi [106] and a general cut-elimination theorem holds for display calculi [5].

In spite of the various properties attributed to 'internal' and 'external' systems, we have identified three ways in which 'internal' systems (i.e., sequent-style systems with a 'structurally similar' and 'logically equivalent' formula translation) are useful. The first use concerns a relationship between *formulaic completeness*, which is when every valid formula in a logic is provable in the proof system, and *sequential completeness*, which is when every valid sequent is provable in the proof system. If the rules of an 'internal' system are invertible and the system has formuliac completeness, then one can (typically) establish sequential completeness. It is straightforward to establish this property: if we assume a sequent is valid, then its formula translation is valid, meaning the formula translation is provable as the system has formulaic completeness. One can then apply the invertibility of the inference rules to the formula translation to prove the original sequent, which establishes sequential completeness. Although we do not claim that this property holds of any system that might be reasonably deemed 'internal' we do note that this method of lifting formulaic completeness to sequential completeness works in a variety of cases; e.g., (linear) nested sequents [61, 72].

A second favorable property of 'internal' calculi concerns the lack of a 'meta-semantics.' Since sequents are interpreted via their 'structurally similar' and 'logically equivalent' formula translations, there is no need to define a more general semantics as is done with labeled systems, for example. Third, it has been shown that 'internal' (viz., nested) systems can be used to derive Hilbert systems, i.e., axiomatizations, for logics [53]. This is obviously beneficial for anyone interested in characterizing a logic purely in terms of its formulae with simple inference rules.

References

[1] A. Avron, *A Constructive Analysis of RM*, **The Journal of Symbolic Logic**, vol. 52(4) (1987), pp. 939–951, DOI: https://doi.org/10.2307/2273828.

[2] A. Avron, *The Method of Hypersequents in the Proof Theory of Propositional Non-Classical Logics*, [in:] W. Hodges, M. Hyland, C. Steinhorn, J. Truss (eds.), **Logic: From Foundations to Applications: European Logic Colloquium**, Clarendon Press, USA (1996), p. 1–32.

[3] S. Baratella, A. Masini, *An approach to infinitary temporal proof theory*, **Archive for Mathematical Logic**, vol. 43(8) (2004), pp. 965–990, DOI: https://doi.org/10.1007/s00153-004-0237-z.

[4] K. Bednarska, A. Indrzejczak, *Hypersequent Calculi for S5: The Methods of Cut Elimination*, **Logic and Logical Philosophy**, vol. 24(3) (2015), pp. 277–311, DOI: https://doi.org/10.12775/LLP.2015.018.

[5] N. D. Belnap, *Display logic*, **Journal of Philosophical Logic**, vol. 11(4) (1982), pp. 375–417, DOI: https://doi.org/10.1007/BF00284976.

[6] P. Blackburn, M. de Rijke, Y. Venema, **Modal Logic**, vol. 53 of Cambridge Tracts in Theoretical Computer Science, Cambridge University Press (2001), DOI: https://doi.org/10.1017/CBO9781107050884.

[7] B. Boretti, **Proof Analysis in Temporal Logic**, Ph.D. thesis, University of Milan (2008).

[8] T. Braüner, **Hybrid Logic and its Proof-Theory**, vol. 37 of Applied Logic Series, Springer Dordrecht (2011), DOI: https://doi.org/10.1007/978-94-007-0002-4.

[9] J. Brotherston, *Bunched Logics Displayed*, **Studia Logica**, vol. 100(6) (2012), p. 1223–1254, DOI: https://doi.org/10.1007/s11225-012-9449-0.

[10] J. Brotherston, R. Goré, *Craig Interpolation in Displayable Logics*, [in:] K. Brünnler, G. Metcalfe (eds.), **Automated Reasoning with Analytic Tableaux and Related Methods**, Springer, Berlin-Heidelberg (2011), pp. 88–103, DOI: https://doi.org/10.1007/978-3-642-22119-4_9.

[11] K. Brünnler, *Deep sequent systems for modal logic*, **Archive of Mathematical Logic**, vol. 48(6) (2009), pp. 551–577, DOI: https://doi.org/10.1007/s00153-009-0137-3.

[12] L. Buisman, R. Goré, *A Cut-Free Sequent Calculus for Bi-intuitionistic Logic*, [in:] N. Olivetti (ed.), **Automated Reasoning with Analytic

Tableaux and Related Methods, Springer, Berlin-Heidelberg (2007), pp. 90–106, DOI: https://doi.org/10.1007/978-3-540-73099-6_9.

[13] R. A. Bull, *Cut elimination for propositional dynamic logic without **, **Zeitschrift für Mathematische Logik und Grundlagen der Mathematik**, vol. 38(2) (1992), pp. 85–100, DOI: https://doi.org/10.1002/malq.19920380107.

[14] S. R. Buss, *An introduction to proof theory*, **Handbook of Proof Theory**, vol. 137 (1998), pp. 1–78.

[15] C. Castellini, A. Smaill, *A systematic presentation of quantified modal logics*, **Logic Journal of the IGPL**, vol. 10(6) (2002), pp. 571–599, DOI: https://doi.org/10.1093/jigpal/10.6.571.

[16] M. A. Castilho, L. F. del Cerro, O. Gasquet, A. Herzig, *Modal tableaux with propagation rules and structural rules*, **Fundamenta Informaticae**, vol. 32(3, 4) (1997), pp. 281–297, DOI: https://doi.org/10.3233/FI-1997-323404.

[17] K. Chaudhuri, N. Guenot, L. Strassburger, *The Focused Calculus of Structures*, [in:] M. Bezem (ed.), **20th EACSL Annual Conference on Computer Science Logic**, vol. 12 of Leibniz International Proceedings in Informatics (LIPIcs), Bergen, Norway (2011), pp. 159–173, DOI: https://doi.org/10.4230/LIPIcs.CSL.2011.159.

[18] A. Ciabattoni, N. Galatos, K. Terui, *From Axioms to Analytic Rules in Nonclassical Logics*, [in:] **2008 23rd Annual IEEE Symposium on Logic in Computer Science** (2008), pp. 229–240, DOI: https://doi.org/10.1109/LICS.2008.39.

[19] A. Ciabattoni, T. S. Lyon, R. Ramanayake, *From Display to Labelled Proofs for Tense Logics*, [in:] S. N. Artëmov, A. Nerode (eds.), **Logical Foundations of Computer Science – International Symposium, LFCS 2018**, vol. 10703 of Lecture Notes in Computer Science, Springer (2018), pp. 120–139, DOI: https://doi.org/10.1007/978-3-319-72056-2_8.

[20] A. Ciabattoni, T. S. Lyon, R. Ramanayake, A. Tiu, *Display to Labeled Proofs and Back Again for Tense Logics*, **ACM Transactions**

in Computational Logic, vol. 22(3) (2021), DOI: https://doi.org/10.1145/3460492.

[21] A. Ciabattoni, P. Maffezioli, L. Spendier, *Hypersequent and Labelled Calculi for Intermediate Logics*, [in:] D. Galmiche, D. Larchey-Wendling (eds.), **Automated Reasoning with Analytic Tableaux and Related Methods**, vol. 8123 of Lecture Notes in Computer Science, Springer, Berlin-Heidelberg (2013), pp. 81–96, DOI: https://doi.org/10.1007/978-3-642-40537-2_9.

[22] A. Ciabattoni, R. Ramanayake, *Power and Limits of Structural Display Rules*, **ACM Transactions on Computational Logic**, vol. 17(3) (2016), DOI: https://doi.org/10.1145/2874775.

[23] A. Ciabattoni, R. Ramanayake, *Bunched Hypersequent Calculi for Distributive Substructural Logics*, [in:] T. Eiter, D. Sands (eds.), **LPAR-21, 21st International Conference on Logic for Programming, Artificial Intelligence and Reasoning, Maun, Botswana, May 7–12, 2017**, vol. 46 of EPiC Series in Computing, EasyChair (2017), pp. 417–434, DOI: https://doi.org/10.29007/ngp3.

[24] A. Ciabattoni, L. Straßburger, K. Terui, *Expanding the Realm of Systematic Proof Theory*, [in:] E. Grädel, R. Kahle (eds.), **Computer Science Logic, 23rd international Workshop, CSL 2009, 18th Annual Conference of the EACSL, Coimbra, Portugal, September 7–11, 2009**, vol. 5771 of Lecture Notes in Computer Science, Springer (2009), pp. 163–178, DOI: https://doi.org/10.1007/978-3-642-04027-6_14.

[25] T. Dalmonte, B. Lellmann, N. Olivetti, E. Pimentel, *Hypersequent calculi for non-normal modal and deontic logics: countermodels and optimal complexity*, **Journal of Logic and Computation**, vol. 31(1) (2020), pp. 67–111, DOI: https://doi.org/10.1093/logcom/exaa072.

[26] K. Došen, *Sequent-systems and groupoid models. I*, **Studia Logica**, vol. 47(4) (1988), pp. 353–385, DOI: https://doi.org/10.1007/BF00671566.

[27] M. Dunn, *A 'Gentzen' system for positive relevant implication*, **The Journal of Symbolic Logic**, vol. 38 (1974), pp. 356–357.

[28] R. Dyckhoff, *Contraction-free sequent calculi for intuitionistic logic*, **The Journal of Symbolic Logic**, vol. 57(3) (1992), pp. 795–807, DOI: https://doi.org/10.2307/2275431.

[29] R. Dyckhoff, *Intuitionistic decision procedures since Gentzen*, **Advances in Proof Theory**, (2016), pp. 245–267.

[30] R. Dyckhoff, S. Negri, *Proof analysis in intermediate logics*, **Archive for Mathematical Logic**, vol. 51(1–2) (2012), pp. 71–92, DOI: https://doi.org/10.1007/s00153-011-0254-7.

[31] M. Fitting, *Tableau methods of proof for modal logics*, **Notre Dame Journal of Formal Logic**, vol. 13(2) (1972), pp. 237–247.

[32] M. Fitting, *Nested Sequents for Intuitionistic Logics*, **Notre Dame Journal of Formal Logic**, vol. 55(1) (2014), pp. 41–61, DOI: https://doi.org/10.1215/00294527-2377869.

[33] M. Fitting, R. Kuznets, *Modal interpolation via nested sequents*, **Annals of Pure and Applied Logic**, vol. 166(3) (2015), pp. 274–305, DOI: https://doi.org/10.1016/j.apal.2014.11.002.

[34] D. Galmiche, M. Marti, D. Méry, *Relating Labelled and Label-Free Bunched Calculi in BI Logic*, [in:] **28th International Conference on Automated Reasoning with Analytic Tableaux and Related Methods, TABLEAUX 2019**, vol. 11714, Springer, Londres, United Kingdom (2019), pp. 130–146, DOI: https://doi.org/10.1007/978-3-030-29026-9_8.

[35] D. Galmiche, D. Méry, D. Pym, *The Semantics of BI and Resource Tableaux*, **Mathematical Structures in Computer Science**, vol. 15(6) (2005), pp. 1033–1088, DOI: https://doi.org/10.1017/S0960129505004858.

[36] D. Galmiche, Y. Salhi, *Tree-sequent calculi and decision procedures for intuitionistic modal logics*, **Journal of Logic and Computation**, vol. 28(5) (2018), pp. 967–989, DOI: https://doi.org/https://doi.org/10.1093/logcom/exv039.

[37] G. Gentzen, *Untersuchungen über das logische Schließen. I*, **Mathematische zeitschrift**, vol. 39(1) (1935), pp. 176–210.

[38] G. Gentzen, *Untersuchungen über das logische Schließen. II*, **Mathematische Zeitschrift**, vol. 39(1) (1935), pp. 405–431.

[39] J.-Y. Girard, *Linear logic*, **Theoretical Computer Science**, vol. 50(1) (1987), pp. 1–101, DOI: https://doi.org/10.1016/0304-3975(87)90045-4.

[40] M. Girlando, B. Lellmann, N. Olivetti, G. L. Pozzato, *Standard Sequent Calculi for Lewis' Logics of Counterfactuals*, [in:] L. Michael, A. C. Kakas (eds.), **Logics in Artificial Intelligence – 15th European Conference, JELIA**, vol. 10021 of Lecture Notes in Computer Science (2016), pp. 272–287, DOI: https://doi.org/10.1007/978-3-319-48758-8_18.

[41] M. Girlando, N. Olivetti, S. Negri, *Counterfactual Logic: Labelled and Internal Calculi, Two Sides of the Same Coin?*, [in:] G. Bezhanishvili, G. D'Agostino, G. Metcalfe, T. Studer (eds.), **Advances in Modal Logic 12**, College Publications (2018), pp. 291–310, URL: http://www.aiml.net/volumes/volume12/Girlando-Olivetti-Negri.pdf.

[42] R. Goré, B. Lellmann, *Syntactic Cut-Elimination and Backward Proof-Search for Tense Logic via Linear Nested Sequents*, [in:] S. Cerrito, A. Popescu (eds.), **Automated Reasoning with Analytic Tableaux and Related Methods**, Springer International Publishing, Cham (2019), pp. 185–202, DOI: https://doi.org/10.1007/978-3-030-29026-9_11.

[43] R. Goré, L. Postniece, A. Tiu, *Cut-elimination and proof-search for bi-intuitionistic logic using nested sequents*, [in:] C. Areces, R. Goldblatt (eds.), **Advances in Modal Logic 7**, College Publications (2008), pp. 43–66, URL: http://www.aiml.net/volumes/volume7/Gore-Postniece-Tiu.pdf.

[44] R. Goré, L. Postniece, A. Tiu, *On the Correspondence between Display Postulates and Deep Inference in Nested Sequent Calculi for Tense Logics*, **Logical Methods in Computer Science**, vol. 7(2) (2011), DOI: https://doi.org/10.2168/LMCS-7(2:8)2011.

[45] R. Goré, *Substructural logics on display*, **Logic Journal of the IGPL**, vol. 6(3) (1998), pp. 451–504, DOI: https://doi.org/10.1093/jigpal/6.3.451.

[46] G. Greco, M. Ma, A. Palmigiano, A. Tzimoulis, Z. Zhao, *Unified correspondence as a proof-theoretic tool*, **Journal of Logic and Computation**, vol. 28(7) (2018), pp. 1367–1442, DOI: https://doi.org/10.1093/logcom/exw022.

[47] A. Guglielmi, *A System of Interaction and Structure*, **ACM Transactions on Computational Logic**, vol. 8(1) (2007), p. 1–es, DOI: https://doi.org/10.1145/1182613.1182614.

[48] R. Hein, **Geometric theories and proof theory of modal logic**, Master's thesis, Technische Universität Dresden (2005).

[49] A. Heyting, *Die formalen Regeln der intuitionistischen Logik*, **Sitzungsbericht PreuBische Akademie der Wissenschaften Berlin, physikalisch-mathematische Klasse II**, (1930), pp. 42–56.

[50] I. Horrocks, U. Sattler, *Decidability of SHIQ with complex role inclusion axioms*, **Artificial Intelligence**, vol. 160(1–2) (2004), pp. 79–104, DOI: https://doi.org/10.1016/j.artint.2004.06.002.

[51] A. Indrzejczak, *Linear Time in Hypersequent Framework*, **The Bulletin of Symbolic Logic**, vol. 22(1) (2016), pp. 121–144, DOI: https://doi.org/10.1017/bsl.2016.2.

[52] A. Indrzejczak, *Cut elimination in Hypersequent Calculus for some Logics of Linear Time*, **The Review of Symbolic Logic**, vol. 12(4) (2019), pp. 806–822, DOI: https://doi.org/10.1017/S1755020319000352.

[53] R. Ishigaki, K. Kikuchi, *Tree-Sequent Methods for Subintuitionistic Predicate Logics*, [in:] N. Olivetti (ed.), **Automated Reasoning with Analytic Tableaux and Related Methods**, vol. 4548 of Lecture Notes in Computer Science, Springer, Berlin-Heidelberg (2007), pp. 149–164.

[54] S. Kanger, **Provability in logic**, Almqvist & Wiksell (1957).

[55] R. Kashima, *Cut-free sequent calculi for some tense logics*, **Studia Logica**, vol. 53(1) (1994), pp. 119–135, DOI: https://doi.org/10.1007/978-3-319-10061-6_4.

[56] M. Kracht, *Power and Weakness of the Modal Display Calculus*, [in:] H. Wansing (ed.), **Proof Theory of Modal Logic**, Springer Nether-

lands, Dordrecht (1996), pp. 93–121, DOI: https://doi.org/10.1007/978-94-017-2798-3_7.

[57] S. A. Kripke, *Semantical Analysis of Intuitionistic Logic I*, [in:] J. Crossley, M. Dummett (eds.), **Formal Systems and Recursive Functions**, vol. 40 of Studies in Logic and the Foundations of Mathematics, Elsevier (1965), pp. 92–130, DOI: https://doi.org/10.1016/S0049-237X(08)71685-9.

[58] H. Kurokawa, *Hypersequent Calculi for Modal Logics Extending S4*, **New Frontiers in Artificial Intelligence**, (2013), pp. 51–68.

[59] H. Kushida, M. Okada, *A proof-theoretic study of the correspondence of classical logic and modal logic*, **Journal of Symbolic Logic**, vol. 68(4) (2003), pp. 1403–1414, DOI: https://doi.org/10.2178/jsl/1067620195.

[60] R. Kuznets, *Proving Craig and Lyndon Interpolation Using Labelled Sequent Calculi*, [in:] L. Michael, A. Kakas (eds.), **Logics in Artificial Intelligence**, vol. 10021 of Lecture Notes in Computer Science, Springer International Publishing, Cham (2016), pp. 320–335, DOI: https://doi.org/10.1007/978-3-319-48758-8_21.

[61] R. Kuznets, B. Lellmann, *Interpolation for Intermediate Logics via Hyper- and Linear Nested Sequents*, [in:] G. Bezhanishvili, G. D'Agostino, G. Metcalfe, T. Studer (eds.), **Advances in Modal Logic 12**, College Publications (2018), pp. 473–492, URL: http://www.aiml.net/volumes/volume12/Kuznets-Lellmann.pdf.

[62] O. Lahav, *From Frame Properties to Hypersequent Rules in Modal Logics*, [in:] **28th Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2013, New Orleans, LA, USA, June 25–28, 2013** (2013), pp. 408–417, DOI: https://doi.org/10.1109/LICS.2013.47.

[63] D. Leivant, *Proof theoretic methodology for propositional dynamic logic*, [in:] J. Díaz, I. Ramos (eds.), **Formalization of Programming Concepts**, Springer, Berlin-Heidelberg (1981), pp. 356–373, DOI: https://doi.org/10.1007/3-540-10699-5_111.

[64] B. Lellmann, *Linear Nested Sequents, 2-Sequents and Hypersequents*, [in:] H. De Nivelle (ed.), **Automated Reasoning with Analytic Tableaux**

and Related Methods, vol. 9323 of Lecture Notes in Computer Science, Springer International Publishing, Cham (2015), pp. 135–150, DOI: https://doi.org/10.1007/978-3-319-24312-2_10.

[65] B. Lellmann, *Hypersequent rules with restricted contexts for propositional modal logics*, **Theoretical Computer Science**, vol. 656 (2016), pp. 76–105, DOI: https://doi.org/https://doi.org/10.1016/j.tcs.2016.10.004.

[66] B. Lellmann, D. Pattinson, *Correspondence between modal Hilbert axioms and sequent rules with an application to S5*, [in:] **Automated Reasoning with Analytic Tableaux and Related Methods: 22nd International Conference, TABLEAUX 2013, Nancy, France, September 16–19, 2013**, Springer (2013), pp. 219–233, DOI: https://doi.org/10.1007/978-3-642-40537-2_19.

[67] B. Lellmann, E. Pimentel, *Proof Search in Nested Sequent Calculi*, [in:] M. Davis, A. Fehnker, A. McIver, A. Voronkov (eds.), **Logic for Programming, Artificial Intelligence, and Reasoning – 20th International Conference, LPAR-20**, vol. 9450 of Lecture Notes in Computer Science, Springer-Verlag, Berlin-Heidelberg (2015), p. 558–574, DOI: https://doi.org/10.1007/978-3-662-48899-7_39.

[68] C. I. Lewis, **A survey of symbolic logic**, University of California Press (1918).

[69] D. Lewis, **Counterfactuals**, Blackwell, Hobokin (1973).

[70] T. S. Lyon, *On the Correspondence between Nested Calculi and Semantic Systems for Intuitionistic Logics*, **Journal of Logic and Computation**, vol. 31(1) (2020), pp. 213–265, DOI: https://doi.org/10.1093/logcom/exaa078.

[71] T. S. Lyon, *Syntactic Cut-Elimination for Intuitionistic Fuzzy Logic via Linear Nested Sequents*, [in:] S. N. Artëmov, A. Nerode (eds.), **Logical Foundations of Computer Science – International Symposium, LFCS 2020, Deerfield Beach, FL, USA, January 4–7, 2020**, vol. 11972 of Lecture Notes in Computer Science, Springer (2020), pp. 156–176, DOI: https://doi.org/10.1007/978-3-030-36755-8_11.

[72] T. S. Lyon, *Nested Sequents for Intuitionistic Modal Logics via Structural Refinement*, [in:] A. Das, S. Negri (eds.), **Automated Reasoning with**

Analytic Tableaux and Related Methods, Springer International Publishing, Cham (2021), pp. 409–427, DOI: https://doi.org/https://doi.org/10.1007/978-3-030-86059-2_24.

[73] T. S. Lyon, **Refining Labelled Systems for Modal and Constructive Logics with Applications**, Ph.D. thesis, Technische Universität Wien (2021).

[74] T. S. Lyon, *Nested Sequents for Intermediate Logics: The Case of Gödel-Dummett Logics*, **Journal of Applied Non-Classical Logics**, vol. 33(2) (2023), pp. 121–164, DOI: https://doi.org/10.1080/11663081.2023.2233346.

[75] T. S. Lyon, *Realizing the Maximal Analytic Display Fragment of Labeled Sequent Calculi for Tense Logics*, **Found on arXiv**, (2024), URL: https://arxiv.org/abs/2406.19882.

[76] T. S. Lyon, *Unifying Sequent Systems for Gödel-Löb Provability Logic via Syntactic Transformations*, [in:] J. Endrullis, S. Schmitz (eds.), **33rd EACSL Annual Conference on Computer Science Logic (CSL 2025)**, vol. 326 of Leibniz International Proceedings in Informatics (LIPIcs), Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl, Germany (2025), pp. 42:1–42:23, DOI: https://doi.org/10.4230/LIPIcs.CSL.2025.42.

[77] T. S. Lyon, L. Gómez Álvarez, *Automating Reasoning with Standpoint Logic via Nested Sequents*, [in:] **Proceedings of the 19th International Conference on Principles of Knowledge Representation and Reasoning** (2022), pp. 257–266, DOI: https://doi.org/10.24963/kr.2022/26.

[78] T. S. Lyon, E. Orlandelli, *Nested Sequents for Quantified Modal Logics*, [in:] R. Ramanayake, J. Urban (eds.), **Automated Reasoning with Analytic Tableaux and Related Methods**, Springer Nature Switzerland, Cham (2023), pp. 449–467, DOI: https://doi.org/10.1007/978-3-031-43513-3_24.

[79] T. S. Lyon, P. Ostropolski-Nalewaja, *Foundations for an Abstract Proof Theory in the Context of Horn Rules*, **Found on arXiv**, (2024), URL: https://arxiv.org/abs/2304.05697.

[80] T. S. Lyon, I. Shillito, A. Tiu, *Taking Bi-Intuitionistic Logic First-Order: A Proof-Theoretic Investigation via Polytree Sequents*, [in:] J. Endrullis, S. Schmitz (eds.), **33rd EACSL Annual Conference on Computer Science Logic (CSL 2025)**, vol. 326 of Leibniz International Proceedings in Informatics (LIPIcs), Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl, Germany (2025), pp. 41:1–41:23, DOI: https://doi.org/10.4230/LIPIcs.CSL.2025.41.

[81] T. S. Lyon, A. Tiu, R. Goré, R. Clouston, *Syntactic Interpolation for Tense Logics and Bi-Intuitionistic Logic via Nested Sequents*, [in:] M. Fernández, A. Muscholl (eds.), **28th EACSL Annual Conference on Computer Science Logic, CSL 2020**, vol. 152 of LIPIcs, Schloss Dagstuhl - Leibniz-Zentrum für Informatik (2020), pp. 28:1–28:16, DOI: https://doi.org/10.4230/LIPIcs.CSL.2020.28.

[82] T. S. Lyon, K. van Berkel, *Automating Agential Reasoning: Proof-Calculi and Syntactic Decidability for STIT Logics*, [in:] M. Baldoni, M. Dastani, B. Liao, Y. Sakurai, R. Zalila Wenkstern (eds.), **PRIMA 2019: Principles and Practice of Multi-Agent Systems - 22nd International Conference**, vol. 11873 of Lecture Notes in Computer Science, Springer International Publishing, Cham (2019), pp. 202–218.

[83] T. S. Lyon, K. van Berkel, *Proof Theory and Decision Procedures for Deontic STIT Logics*, **Journal of Artificial Intelligence Research**, vol. 81 (2024), pp. 837–876, DOI: https://doi.org/https://doi.org/10.48550/arXiv.2402.03148.

[84] S. Maehara, *Eine Darstellung der Intuitionistischen Logik in der Klassischen*, **Nagoya Mathematical Journal**, vol. 7 (1954), p. 45–64, DOI: https://doi.org/10.1017/S0027763000018055.

[85] S. Maehara, *On the interpolation theorem of Craig*, **Sûgaku**, vol. 12(4) (1960), pp. 235–237.

[86] S. Marin, M. Morales, L. Straßburger, *A fully labelled proof system for intuitionistic modal logics*, **Journal of Logic and Computation**, vol. 31(3) (2021), pp. 998–1022, DOI: https://doi.org/10.1093/logcom/exab020.

[87] A. Masini, *2-sequent calculus: A proof theory of modalities*, **Annals of Pure and Applied Logic**, vol. 58(3) (1992), pp. 229–246, DOI: https://doi.org/10.1016/0168-0072(92)90029-Y.

[88] A. Masini, *2-sequent calculus: Intuitionism and natural deduction*, **Journal of Logic and Computation**, vol. 3(5) (1993), pp. 533–562, DOI: https://doi.org/10.1093/logcom/3.5.533.

[89] K. L. McMillan, *Interpolation and Model Checking*, [in:] E. M. Clarke, T. A. Henzinger, H. Veith, R. Bloem (eds.), **Handbook of Model Checking**, Springer International Publishing, Cham (2018), pp. 421–446, DOI: https://doi.org/10.1007/978-3-319-10575-8_14.

[90] G. Metcalfe, N. Olivetti, D. M. Gabbay, *Sequent and hypersequent calculi for abelian and Łukasiewicz logics*, **ACM Transactions on Computational Logic**, vol. 6(3) (2005), pp. 578–613, DOI: https://doi.org/10.1145/1071596.1071600.

[91] G. E. Mints, *Some calculi of modal logic*, **Trudy Matematicheskogo Instituta imeni VA Steklova**, vol. 98 (1968), pp. 88–111.

[92] G. E. Mints, **Studies in constructive mathematics and mathematical logic. Part V**, Nauka, Leningrad. Otdel, chap. Cut-elimination theorem for relevant logics (in Russian) (1972), pp. 90–97.

[93] G. E. Mints, *Indexed systems of sequents and cut-elimination*, **Journal of Philosophical Logic**, vol. 26(6) (1997), pp. 671–696, DOI: https://doi.org/10.1023/A:1017948105274.

[94] S. Negri, *Proof analysis in modal logic*, **Journal of Philosophical Logic**, vol. 34(5–6) (2005), p. 507, DOI: https://doi.org/10.1007/s10992-005-2267-3.

[95] P. W. O'Hearn, D. Pym, *The Logic of Bunched Implications*, **Bulletin of Symbolic Logic**, vol. 5(2) (1999), pp. 215–244, DOI: https://doi.org/https://doi.org/10.2307/421090.

[96] E. Pimentel, R. Ramanayake, B. Lellmann, *Sequentialising Nested Systems*, [in:] S. Cerrito, A. Popescu (eds.), **Automated Reasoning with Analytic Tableaux and Related Methods**, vol. 11714 of

Lecture Notes in Computer Science, Springer International Publishing, Cham (2019), pp. 147–165, DOI: https://doi.org/10.1007/978-3-030-29026-9_9.

[97] F. Poggiolesi, *A Cut-free Simple Sequent calculus for Modal Logic S5*, **The Review of Symbolic Logic**, vol. 1(1) (2008), p. 3–15, DOI: https://doi.org/10.1017/S1755020308080040.

[98] F. Poggiolesi, *The Method of Tree-Hypersequents for Modal Propositional Logic*, [in:] D. Makinson, J. Malinowski, H. Wansing (eds.), **Towards Mathematical Philosophy**, vol. 28 of Trends in logic, Springer (2009), pp. 31–51, DOI: https://doi.org/10.1007/978-1-4020-9084-4_3.

[99] F. Poggiolesi, **A Tree-Hypersequent Calculus for the Modal Logic of Provability**, Springer Netherlands, Dordrecht (2011), pp. 187–201, DOI: https://doi.org/10.1007/978-90-481-9670-8_10.

[100] F. Poggiolesi, G. Restall, **Interpreting and Applying Proof Theories for Modal Logic**, Palgrave Macmillan UK, London (2012), pp. 39–62, DOI: https://doi.org/10.1057/9781137003720_4.

[101] G. Pottinger, *Uniform, cut-free formulations of T, S4 and S5*, **Journal of Symbolic Logic**, vol. 48(3) (1983), p. 900.

[102] A. N. Prior, **Time and Modality**, Oxford University Press (1957).

[103] D. Pym, **The Semantics and Proof Theory of the Logic of Bunched Implications**, vol. 26 of Applied Logic Series, Kluwer Academic Publishers (2002), DOI: https://doi.org/10.1007/978-94-017-0091-7.

[104] S. Read, *Semantic Pollution and Syntactic Purity*, **The Review of Symbolic Logic**, vol. 8(4) (2015), DOI: https://doi.org/10.1017/S1755020315000210.

[105] G. Restall, *Proofnets for S5: sequents and circuits for modal logic*, [in:] **Logic Colloquium 2005, Lecture Notes in Logic**, vol. 28, Cambridge University Press (2007), pp. 151–172.

[106] A. K. Simpson, **The proof theory and semantics of intuitionistic modal logic**, Ph.D. thesis, University of Edinburgh. College of Science and Engineering. School of Informatics (1994).

[107] J. K. Slaney, *Minlog: A Minimal Logic Theorem Prover*, [in:] W. McCune (ed.), **Automated Deduction – CADE-14, 14th International Conference on Automated Deduction, Townsville, North Queensland, Australia**, vol. 1249 of Lecture Notes in Computer Science, Springer (1997), pp. 268–271, DOI: https://doi.org/10.1007/3-540-63104-6_27.

[108] L. Straßburger, *Cut Elimination in Nested Sequents for Intuitionistic Modal Logics*, [in:] F. Pfenning (ed.), **Foundations of Software Science and Computation Structures**, vol. 7794 of Lecture Notes in Computer Science, Springer, Berlin-Heidelberg (2013), pp. 209–224, DOI: https://doi.org/10.1007/978-3-642-37075-5_14.

[109] A. Tiu, E. Ianovski, R. Goré, *Grammar Logics in Nested Sequent Calculus: Proof Theory and Decision Procedures*, [in:] T. Bolander, T. Braüner, S. Ghilardi, L. S. Moss (eds.), **Advances in Modal Logic 9**, College Publications (2012), pp. 516–537, URL: http://www.aiml.net/volumes/volume9/Tiu-Ianovski-Gore.pdf.

[110] K. van Berkel, T. S. Lyon, *Cut-Free Calculi and Relational Semantics for Temporal STIT Logics*, [in:] F. Calimeri, N. Leone, M. Manna (eds.), **Logics in Artificial Intelligence**, Springer International Publishing, Cham (2019), pp. 803–819, DOI: https://doi.org/10.1007/978-3-030-19570-0_52.

[111] K. van Berkel, T. S. Lyon, *The Varieties of Ought-implies-Can and Deontic STIT Logic*, [in:] F. Liu, A. Marra, P. Portner, F. V. D. Putte (eds.), **Deontic Logic and Normative Systems: 15th International Conference (DEON2020/2021)**, College Publications (2021), URL: https://www.collegepublications.co.uk/DEON/Van%20Berkel%20&%20Lyon_DEON2020.pdf.

[112] L. Viganò, **Labelled Non-Classical Logics**, Springer Science & Business Media (2000), DOI: https://doi.org/10.1007/978-1-4757-3208-5.

[113] H. Wansing, *Sequent calculi for normal modal propositional logics*, **Journal of Logic and Computation**, vol. 4(2) (1994), pp. 125–142, DOI: https://doi.org/10.1093/logcom/4.2.125.

[114] H. Wansing, *Sequent Systems for Modal Logics*, [in:] D. M. Gabbay, F. Guenthner (eds.), **Handbook of Philosophical Logic: Volume 8**, Springer Netherlands, Dordrecht (2002), pp. 61–145, DOI: https://doi.org/10.1007/978-94-010-0387-2_2.

[115] F. Wolter, *On Logics with Coimplication*, **Journal of Philosophical Logic**, vol. 27(4) (1998), pp. 353–387, DOI: https://doi.org/10.1023/A:1004218110879.

Tim S. Lyon
Technische Universität Dresden
Germany
e-mail: timothy_stephen.lyon@tu-dresden.de

Agata Ciabattoni
Vienna University of Technology
Austria
e-mail: agata@logic.at

Didier Galmiche
Université de Lorraine, CNRS, LORIA
France
e-mail: didier.galmiche@loria.fr

Marianna Girlando
University of Amsterdam
Netherlands
e-mail: m.girlando@uva.nl

Dominique Larchey-Wendling
Université de Lorraine, CNRS, LORIA
France
e-mail: dominique.larchey-wendling@loria.fr

Daniel Méry
Université de Lorraine, CNRS, LORIA
France
e-mail: daniel.mery@loria.fr

Nicola Olivetti
Aix-Marseille University
France
e-mail: nicola.olivetti@univ-amu.fr

Revantha Ramanayake
University of Groningen
Netherlands
e-mail: d.r.s.ramanayake@rug.nl

Funding information: Tim S. Lyon was supported by the European Research Council, Consolidator Grant DeciGUT (771779).

Conflict of interests: None.

Ethical considerations: The Authors assure of no violations of publication ethics and take full responsibility for the content of the publication.

The percentage share of the author in the preparation of the work: Tim S. Lyon 30%, Agata Ciabattoni 10%, Didier Galmiche 10%, Marianna Girlando 10%, Dominique Larchey-Wendling 10%, Daniel Méry 10%, Nicola Olivetti 10%, Revantha Ramanayake 10%

Declaration regarding the use of GAI tools: Not used.