



Nested Sequents for Intuitionistic Modal Logics via Structural Refinement

Tim Lyon

EasyChair preprints are intended for rapid dissemination of research results and are integrated with the rest of EasyChair.

November 22, 2021

Nested Sequents for Intuitionistic Modal Logics via Structural Refinement^{*}

Tim S. Lyon^[0000–0003–3214–0828]

Computational Logic Group, Institute of Artificial Intelligence, Technische
Universität Dresden, Germany
`timothy.stephen.lyon@tu-dresden.de`

Abstract. We employ a recently developed methodology—called *structural refinement*—to extract nested sequent systems for a sizable class of intuitionistic modal logics from their respective labelled sequent systems. This method can be seen as a means by which labelled sequent systems can be transformed into nested sequent systems through the introduction of propagation rules and the elimination of structural rules, followed by a notational translation. The nested systems we obtain incorporate propagation rules that are parameterized with formal grammars, and which encode certain frame conditions expressible as first-order Horn formulae that correspond to a subclass of the Scott-Lemmon axioms. We show that our nested systems are sound, cut-free complete, and admit hp-admissibility of typical structural rules.

Keywords: Bi-relational model · Intuitionistic modal logic · Labelled sequent · Nested sequent · Proof theory · Propagation rule · Refinement

1 Introduction

Intuitionistic modal logics enable intuitionistic reasoning with the intensional operators \diamond and \Box . While a variety of different intuitionistic modal logics have been proposed [1, 9, 28, 30, 31], we focus on those defined in [28], which extend the intuitionistic modal logic \mathbf{IK} with Scott-Lemmon axioms [20]. These logics were placed on a firm philosophical footing in [31] due to their satisfaction of certain requirements that one might reasonably impose upon an intuitionistic version of modal logic. Although such logics are interesting in their own right, intuitionistic modal logics have proven useful in practical applications: having been applied in the verification of computer hardware [8], to facilitate reasoning about functional programs [27], and in defining programming languages [7].

The development of intuitionistic modal logics naturally gave rise to an accompanying proof theory. Labelled natural deduction and sequent systems were provided for \mathbf{IK} extended with geometric axioms in [31]. In [13] and [14], label-free natural deduction systems and tree-sequent calculi were respectively provided for

^{*} Work supported by the European Research Council (ERC) Consolidator Grant 771779 (DeciGUT).

extensions of IK with combinations of the reflexivity axiom (T), symmetry axiom (B), transitivity axiom (4), and Euclidean axiom (5). In [32], nested sequent systems were proposed for all logics within the intuitionistic modal cube (i.e. logics axiomatized by extending IK with a subset of the axioms T, B, 4, 5, and the seriality axiom D). Such systems provide a suitable basis for developing automated reasoning and proof-search methods, having been used—in particular—to establish the decidability of logics within the intuitionistic modal cube [14, 31].

With the exception of the systems introduced in [31], the drawback of the aforementioned proof systems is that they are rather limited, only being defined for a handful of logics. Indeed, in a recent paper on nested systems for intuitionistic modal logics [25], the authors leave open the problem of defining rules within the nested sequent formalism that allow for the capture of logics *outside* the intuitionistic modal cube. Accomplishing such a task would prove beneficial, since systems built within the nested formalism tend to be more economical (viz. they utilize simpler data structures) than those built within the labelled formalism, and have proven well-suited for the construction of analytic calculi [2, 3, 19], for writing decision algorithms [14, 33], and for verifying interpolation [12, 24].

In this paper, we answer the open problem of [25] to a large extent, and provide cut-free nested sequent systems for extensions of IK with what we call *Horn-Scott-Lemmon axioms (HSLs)*, namely, axioms of the form $(\Diamond^n \Box A \supset \Box^k A) \wedge (\Diamond^k A \supset \Box^n \Diamond A)$. We obtain such systems through the recently developed *structural refinement* methodology [21], which consists of transforming a labelled sequent system into a nested system through the introduction of *propagation rules* (cf. [4, 10]) and the elimination of structural rules, followed by a notational translation. The propagation rules operate by viewing labelled sequents (which encode binary labelled graphs) as automata, allowing for formulae to be propagated along a path in the underlying graph of a labelled sequent, so long as the path is encoded by a string derivable in a certain formal grammar. The refinement methodology grew out of works relating labelled systems to ‘more refined’ or nested systems [5, 18, 23, 26]. Also, the propagation rules we use are largely based upon the work of [17, 33], where such rules were used in the setting of display and nested calculi. These rules were then transported to the labelled setting to prove the decidability of agency logics [23], to establish translations between calculi within various proof-theoretic formalisms [6], and to provide a basis for the structural refinement methodology [21].

This paper accomplishes the following: First, we show that structural refinement can be used to extract nested sequent systems from Simpson’s labelled sequent systems [31] with proofs in the latter formalism algorithmically translatable into proofs of the nested formalism. Second, we provide sound and cut-free complete nested sequent systems for a considerable class of intuitionistic modal logics, and show that such systems admit the height-preserving admissibility (which we refer to as *hp-admissibility*) of certain structural rules (e.g. forms of weakening and contraction). Third, we provide an answer to the open problem of [25] to a large degree, giving a straightforward procedure for transforming axioms (viz. HSLs) into propagation/logical rules.

We have organized this paper accordingly: In Sect. 2, we define the intuitionistic modal logics considered, along with their axiomatizations and semantics. In Sect. 3, we introduce fundamental concepts in grammar theory that are needed for the definition of our propagation rules. We then introduce Simpson’s labelled sequent calculi for intuitionistic modal logics in Sect. 4, and show how to structurally refine them in Sect. 5. Last, in Sect. 6, we translate the refined labelled systems of the previous section into sound and cut-free complete nested sequent systems admitting the hp-admissibility of certain structural rules.

2 Logical Preliminaries

In this section, we introduce the language, semantics, and axiomatization for the intuitionistic modal logic \mathbf{IK} [28].¹ Moreover, we also introduce extensions of \mathbf{IK} (referred to as *intuitionistic modal logics* more generally) with the seriality axiom \mathbf{D} and axioms that we refer to as *Horn-Scott-Lemmon Axioms (HSLs)*.

We define our intuitionistic modal language \mathcal{L} to be the set of formulae generated via the following BNF grammar:

$$A ::= p \mid \perp \mid A \vee A \mid A \wedge A \mid A \supset A \mid \diamond A \mid \Box A$$

where p ranges over the set of propositional atoms $\Phi := \{p, q, r, \dots\}$. We use A, B, C, \dots (occasionally annotated) to range over formulae in \mathcal{L} , and define $\sim A := A \supset \perp$ and $A \equiv B := (A \supset B) \wedge (B \supset A)$. For $n \in \mathbb{N}$, we use $\diamond^n A$ and $\Box^n A$ to represent the formula A prefixed with a sequence of n diamonds or boxes, respectively. We interpret such formulae on *bi-relational models* [28, 31]:

Definition 1 (Bi-relational Model [28]). *We define a bi-relational model to be a tuple $M := (W, \leq, R, V)$ such that:*

- W is a non-empty set of worlds w, u, v, \dots (potentially annotated);
- The intuitionistic relation $\leq \subseteq W \times W$ is reflexive and transitive;
- The accessibility relation $R \subseteq W \times W$ satisfies:
 - (F1) For all $w, v, v' \in W$, if wRv and $v \leq v'$, then there exists a $w' \in W$ such that $w \leq w'$ and $w'Rv'$;
 - (F2) For all $w, w', v \in W$, if $w \leq w'$ and wRv , then there exists a $v' \in W$ such that $w'Rv'$ and $v \leq v'$;
- $V : W \rightarrow 2^\Phi$ is a valuation function satisfying the monotonicity condition: For each $w, u \in W$, if $w \leq u$, then $V(w) \subseteq V(u)$.

Formulae from \mathcal{L} may then be interpreted over bi-relational models as specified by the semantic clauses below.

Definition 2 (Semantic Clauses [28]). *Let M be a bi-relational model with $w \in W$ of M . The satisfaction relation $M, w \Vdash A$ is defined recursively:*

¹ See Simpson’s 1994 PhD Thesis [31] for a detailed introduction and discussion of \mathbf{IK} .

- $M, w \Vdash p$ iff $p \in V(w)$, for $p \in \Phi$;
- $M, w \not\Vdash \perp$;
- $M, w \Vdash A \vee B$ iff $M, w \Vdash A$ or $M, w \Vdash B$;
- $M, w \Vdash A \wedge B$ iff $M, w \Vdash A$ and $M, w \Vdash B$;
- $M, w \Vdash A \supset B$ iff for all $w' \in W$, if $w \leq w'$ and $M, w' \Vdash A$, then $M, w' \Vdash B$;
- $M, w \Vdash \diamond A$ iff there exists a $v \in W$ such that wRv and $M, v \Vdash A$;
- $M, w \Vdash \Box A$ iff for all $w', v' \in W$, if $w \leq w'$ and $w'Rv'$, then $M, v' \Vdash A$.

We say that a formula A is globally true on M , written $M \Vdash A$, iff $M, u \Vdash A$ for all worlds $u \in W$ of M , and we say that a formula A is valid, written $\Vdash A$, iff A is globally true on all bi-relational models.

As shown by Plotkin and Stirling in [28], the validities of IK are axiomatizable:

Definition 3 (Axiomatization [28]). We define the axiomatization HIK as:

| | |
|--|--|
| $A0$ All theorems of propositional intuitionistic logic | $A4 \ \diamond(A \vee B) \supset (\diamond A \vee \diamond B)$ |
| $A1 \ \Box(A \supset B) \supset (\Box A \supset \Box B)$ | $A5 \ (\diamond A \supset \Box B) \supset \Box(A \supset B)$ |
| $A2 \ \Box(A \supset B) \supset (\diamond A \supset \diamond B)$ | $R0 \ \frac{A \quad A \supset B}{B} \text{ (mp)}$ |
| $A3 \ \sim \diamond \perp$ | $R1 \ \frac{A}{\Box A} \text{ (nec)}$ |

We define IK to be the smallest set of formulae closed under substitutions of the above axioms and applications of the inference rules, and define A to be a theorem of IK iff $A \in \text{IK}$.

We also consider extensions of HIK with sets \mathcal{A} of the following axioms:

$$D : \Box A \supset \diamond A \quad \text{HSL} : (\diamond^n \Box A \supset \Box^k A) \wedge (\diamond^k A \supset \Box^n \diamond A)$$

The above left axiom is referred to as the *seriality axiom D* and axioms of the form above right are referred to as *Horn-Scott-Lemmon axioms (HSLs)*, which we use $\phi(n, k)$ to denote.² For the remainder of the paper, we use \mathcal{A} to denote an arbitrary set of the above axioms, that is:

$$\mathcal{A} \subseteq \{D\} \cup \{(\diamond^n \Box A \supset \Box^k A) \wedge (\diamond^k A \supset \Box^n \diamond A) \mid n, k \in \mathbb{N}\}$$

The set of HSLs includes well-known axioms such as:

$$T : (A \supset \diamond A) \wedge (\Box A \supset A) \quad 4 : (\diamond \diamond A \supset \diamond A) \wedge (\Box A \supset \Box \Box A)$$

$$B : (\diamond \Box A \supset A) \wedge (A \supset \Box \diamond A) \quad 5 : (\diamond \Box A \supset \Box A) \wedge (\diamond A \supset \Box \diamond A)$$

The work of Simpson [31] establishes that any extension of HIK with a set \mathcal{A} of axioms is sound and complete relative to a subclass of the bi-relational models.

² We note that the term *Horn-Scott-Lemmon axiom* arises from the fact that such axioms form a proper subclass of the well-known *Scott-Lemmon Axioms* [20] and are associated with frame conditions that are expressible as Horn formulae [31, Sect. 7.2].

| Axiom | Frame Condition |
|--|--|
| $\Box A \supset \Diamond A$ | $\forall w \exists u (wRu)$ |
| $(\Diamond^n \Box A \supset \Box^k A) \wedge (\Diamond^k A \supset \Box^n \Diamond A)$ | $\forall w, u, v (wR^n u \wedge wR^k v \supset uRv)$ |

Fig. 1. Axioms and their related frame conditions. We note that when $n = 0$, the related frame condition is $\forall w, v (wR^k v \supset wRv)$, when $k = 0$, the related frame condition is $\forall w, u (wR^n u \supset uRw)$, and when $n = k = 0$, the related frame condition is $\forall w (wRw)$.

In particular, the extension of HIK with a set \mathcal{A} of axioms is sound and complete relative to the set of bi-relational models satisfying the frame conditions related to the axioms of \mathcal{A} , as specified in Fig. 1.³ We define axiomatic extensions of HIK along with their corresponding models below:

Definition 4 (Extensions, Bi-relational \mathcal{A} -model, \mathcal{A} -valid). *The axiomatization $\text{HIK}(\mathcal{A})$ is defined to be HIK extended with the axioms from \mathcal{A} , and we define the logic $\text{IK}(\mathcal{A})$ to be the smallest set of formulae closed under substitutions of the axioms of $\text{HIK}(\mathcal{A})$ and applications of the inference rules. Also, a theorem of $\text{IK}(\mathcal{A})$ is a formula A such that $A \in \text{IK}(\mathcal{A})$. Moreover, we define a bi-relational \mathcal{A} -model to be a bi-relational model satisfying each frame condition related to an axiom $A \in \mathcal{A}$ (as specified in Fig. 1). Last, a formula A is \mathcal{A} -valid iff it is globally true on all \mathcal{A} -models.*

Remark 1. We note that $\text{HIK} = \text{HIK}(\emptyset)$ and that a bi-relational \emptyset -model is a bi-relational model.

Theorem 1 (Soundness and Completeness [31]). *A formula is a theorem of $\text{HIK}(\mathcal{A})$ iff it is valid in all \mathcal{A} -frames.*

Proof. Follows from Thm. 6.2.1 and Thm. 8.1.4 of [31]. □

3 Grammar Theoretic Preliminaries

As will be seen later on (viz. in Sect. 5 and 6), a central component to our refinement methodology—i.e. the extraction of nested calculi from labelled—is the use of inference rules whose applicability is determined on the basis of strings generated by a formal grammar. We therefore introduce grammar-theoretic notions that are essential to the functionality of such rules.

We let Σ be our *alphabet* consisting of the *characters* \diamond and \blacklozenge , that is, $\Sigma := \{\diamond, \blacklozenge\}$. The symbols \diamond and \blacklozenge will be used to encode information about the accessibility relation R of a bi-relational model in certain inference rules of our calculi. In particular, \diamond will be used to encode information about what is

³ We note that the axioms we consider do not *characterize* the set of frames satisfying the frame properties related to the axioms as they do in the classical setting. For more details concerning this point, see [31, p. 56], and for details concerning the proper characterization results of the above axioms, see [28].

happening in the *future* of the accessibility relation, and \blacklozenge will be used to encode information about what is happening in the *past* of the accessibility relation. We note that such symbols have been chosen due to their analogous meaning in the context of tense logics [17, 19]. Also, following [17], we let $\langle ? \rangle \in \Sigma$ and $\langle ? \rangle^{-1} \in \Sigma \setminus \{\langle ? \rangle\}$, i.e. $\diamond^{-1} := \blacklozenge$ and $\blacklozenge^{-1} := \diamond$; we refer to \diamond and \blacklozenge as *converses* of one another. We may define *strings* over our alphabet Σ accordingly:

Definition 5 (Σ^*). *We let \cdot be the concatenation operation with ε the empty string. We define the set Σ^* of strings over Σ to be the smallest set such that:*

- $\Sigma \cup \{\varepsilon\} \subseteq \Sigma^*$
- If $s \in \Sigma^*$ and $\langle ? \rangle \in \Sigma$, then $s \cdot \langle ? \rangle \in \Sigma^*$

For a set Σ^* of strings, we use s, t, r, \dots (potentially annotated) to represent strings in Σ^* . Also, the empty string ε is taken to be the identity element for the concatenation operation, i.e. $s \cdot \varepsilon = \varepsilon \cdot s = s$ for $s \in \Sigma^*$. Furthermore, we will not explicitly mention the concatenation operation in practice and let $st := s \cdot t$, that is, we denote concatenation by simply gluing two strings together. Beyond concatenation, another useful operation to define on strings is the *converse operation*, adapted from [33].

Definition 6 (String Converse). *We extend the converse operation to strings as follows:*

- $\varepsilon^{-1} := \varepsilon$;
- If $s = \langle ? \rangle_1 \dots \langle ? \rangle_n$, then $s^{-1} := \langle ? \rangle_n^{-1} \dots \langle ? \rangle_1^{-1}$.

We let $\langle ? \rangle^n$ denote a string consisting of n copies of $\langle ? \rangle$, which is ε when $n = 0$. Making use of such notation, we can compactly define the notion of an \mathcal{A} -grammar, which encodes information contained in a set \mathcal{A} of axioms, and which will be employed in the definition of certain inference rules (see Sect. 5).

Definition 7 (\mathcal{A} -grammar). *We define an \mathcal{A} -grammar to be a set $g(\mathcal{A})$ such that:*

$$(\diamond \longrightarrow \blacklozenge^n \diamond^k), (\blacklozenge \longrightarrow \blacklozenge^k \diamond^n) \in g(\mathcal{A}) \text{ iff } (\diamond^n \square A \supset \square^k A) \wedge (\diamond^k A \supset \square^n \diamond A) \in \mathcal{A}.$$

We call rules of the form $\langle ? \rangle \longrightarrow s$ production rules, where $\langle ? \rangle \in \Sigma$ and $s \in \Sigma^$.*

An \mathcal{A} -grammar $g(\mathcal{A})$ is a type of *Semi-Thue system* (cf. [29]), i.e. it is a string re-writing system. For example, assuming that $\langle ? \rangle \longrightarrow s \in g(\mathcal{A})$, we may derive the string tsr from $t\langle ? \rangle r$ in one-step by applying the mentioned production rule. As usual, through successive applications of production rules to a string $s \in \Sigma^*$, one obtains derivations of new strings, the collection of which, determines a language. We make such notions precise by means of the following definition:

Definition 8 (Derivation, Language). *Let $g(\mathcal{A})$ be an \mathcal{A} -grammar. The one-step derivation relation $\longrightarrow_{g(\mathcal{A})}$ holds between two strings s and t in Σ^* , written $s \longrightarrow_{g(\mathcal{A})} t$, iff there exist $s', t' \in \Sigma^*$ and $\langle ? \rangle \longrightarrow r \in S$ such that $s = s'\langle ? \rangle t'$*

and $t = s'rt'$. The derivation relation $\longrightarrow_{g(\mathcal{A})}^*$ is defined to be the reflexive and transitive closure of $\longrightarrow_{g(\mathcal{A})}$. For two strings $s, t \in \Sigma^*$, we refer to $s \longrightarrow_{g(\mathcal{A})}^* t$ as a derivation of t from s , and define its length to be equal to the minimal number of one-step derivations needed to derive t from s in $g(\mathcal{A})$. Last, for a string $s \in \Sigma^*$, the language of s relative to $g(\mathcal{A})$ is defined to be the set $L_{g(\mathcal{A})}(s) := \{t \mid s \longrightarrow_{g(\mathcal{A})}^* t\}$.

4 Labelled Sequent Systems

We introduce equivalent variants of Simpson's labelled sequent systems for intuitionistic modal logics [31], which are uniformly presented in Fig. 2. We use the name $\mathbf{L}_{\square\Diamond}(\mathcal{A})$ to denote a labelled system as opposed to Simpson's name $\mathbf{L}_{\square\Diamond}(\mathcal{T})$ since we define each system relative to a set \mathcal{A} of axioms (cf. [31]). The sole difference between Simpson's original systems and the systems presented here is that we copy principal formulae into the premises of some rules. This minor change will facilitate our work in the subsequent section.

Simpson's systems make use of a denumerable set $Lab := \{w, u, v, \dots\}$ of labels (which we sometimes annotate), as well as two distinct types of formulae: *labelled formulae*, which are of the form $w : A$ with $w \in Lab$ and $A \in \mathcal{L}$, and *relational atoms*, which are of the form wRu for $w, u \in Lab$. We define a *labelled sequent* to be a formula of the form $\mathcal{R}, \Gamma \vdash w : A$, where \mathcal{R} is a (potentially empty) multiset of relational atoms, and Γ is a (potentially empty) multiset of labelled formulae. Also, we define a sequence of relational atoms $wR^n u := wRw_1, w_1Rw_2, \dots, w_{n-1}Ru$, for $n \in \mathbb{N}$, and note that $wR^0 u := (w = u)$.

We refer to the (*id*) and (\perp_l) rules as *initial rules*, to the (*d*) and ($S_{n,k}$) rules as *structural rules*, and to the remaining rules in Fig. 2 as *logical rules*. Our use of the term *structural rules* in reference to (*d*) and ($S_{n,k}$) is consistent with the use of the term in the literature on proof systems for modal and related logics [2, 6, 16, 17] and is based on the fact that such rules manipulate the underlying data structure of sequents as opposed to introducing more complex logical formulae. Also, we point out that the ($S_{n,k}$) rules form a proper subclass of Simpson's (S_χ) *geometric structural rules* (see [31, p. 126]) used to generate labelled sequent systems for IK extended with any number of *geometric axioms*. When $n = 0$ or $k = 0$ in an HSL, i.e. when $\phi(0, k) \in \mathcal{A}$, $\phi(n, 0) \in \mathcal{A}$, or $\phi(0, 0) \in \mathcal{A}$, the structural rules ($S_{0,k}$), ($S_{n,0}$), and ($S_{0,0}$) are defined accordingly:

$$\frac{\mathcal{R}, wR^k v, wRv, \Gamma \vdash z : A}{\mathcal{R}, wR^k v, \Gamma \vdash z : A} (S_{0,k}) \quad \frac{\mathcal{R}, wR^n u, uRw, \Gamma \vdash z : A}{\mathcal{R}, wR^n u, \Gamma \vdash z : A} (S_{n,0})$$

$$\frac{\mathcal{R}, wRw, \Gamma \vdash z : A}{\mathcal{R}, \Gamma \vdash z : A} (S_{0,0})$$

Let us now define the semantics for our labelled sequents, and then we state the soundness and completeness theorem for $\mathbf{L}_{\square\Diamond}(\mathcal{A})$.

$$\begin{array}{c}
\frac{}{\mathcal{R}, w : p, \Gamma \vdash w : p} (id) \quad \frac{}{\mathcal{R}, w : \perp, \Gamma \vdash u : A} (\perp_l) \\
\\
\frac{\mathcal{R}, \Gamma, w : A \vdash u : C \quad \mathcal{R}, \Gamma, w : B \vdash u : C}{\mathcal{R}, \Gamma, w : A \vee B \vdash u : C} (\vee_l) \quad \frac{\mathcal{R}, \Gamma \vdash w : A_i}{\mathcal{R}, \Gamma \vdash w : A_1 \vee A_2} (\vee_r) \quad i \in \{1, 2\} \\
\\
\frac{\mathcal{R}, \Gamma, w : A, w : B \vdash u : C}{\mathcal{R}, \Gamma, w : A \wedge B \vdash u : C} (\wedge_l) \quad \frac{\mathcal{R}, \Gamma \vdash w : A \quad \mathcal{R}, \Gamma \vdash w : B}{\mathcal{R}, \Gamma \vdash w : A \wedge B} (\wedge_r) \\
\\
\frac{\mathcal{R}, \Gamma, w : A \supset B \vdash w : A \quad \mathcal{R}, \Gamma, w : B \vdash u : C}{\mathcal{R}, \Gamma, w : A \supset B \vdash u : C} (\supset_l) \quad \frac{\mathcal{R}, \Gamma, w : A \vdash w : B}{\mathcal{R}, \Gamma \vdash w : A \supset B} (\supset_r) \\
\\
\frac{\mathcal{R}, wRu, \Gamma, u : A \vdash v : B}{\mathcal{R}, \Gamma, w : \diamond A \vdash v : B} (\diamond_l)^\dagger \quad \frac{\mathcal{R}, wRu, \Gamma \vdash u : A}{\mathcal{R}, wRu, \Gamma \vdash w : \diamond A} (\diamond_r) \\
\\
\frac{\mathcal{R}, wRu, \Gamma \vdash u : A}{\mathcal{R}, \Gamma \vdash w : \Box A} (\Box_r)^\dagger \quad \frac{\mathcal{R}, wRu, \Gamma, w : \Box A, u : A \vdash v : C}{\mathcal{R}, wRu, \Gamma, w : \Box A \vdash v : C} (\Box_l) \\
\\
\frac{\mathcal{R}, wRu, \Gamma \vdash v : A}{\mathcal{R}, \Gamma \vdash v : A} (d)^\dagger \quad \frac{\mathcal{R}, wR^n u, wR^k v, uRv, \Gamma \vdash z : A}{\mathcal{R}, wR^n u, wR^k v, \Gamma \vdash z : A} (S_{n,k})
\end{array}$$

Fig. 2. The labelled calculi $\mathbf{L}_{\Box \diamond}(\mathcal{A})$. We have (d) as a rule in the calculus, if $D \in \mathcal{A}$, and $(S_{n,k})$ as a rule in the calculus, for each $\phi(n,k) \in \mathcal{A}$. The side condition \dagger states that u must be an *eigenvariable*, i.e. u may not occur in the conclusion.

Definition 9 (Labelled Sequent Semantics). Let $M := (W, \leq, R, V)$ be a bi-relational \mathcal{A} -model with $I : \text{Lab} \mapsto W$ an interpretation function mapping labels to worlds. We define the satisfaction of relational atoms and labelled formulae:

- $M, I \models wRu$ iff $I(w)RI(u)$;
- $M, I \models w : A$ iff $M, I(w) \Vdash A$.

A labelled sequent $\Lambda := \mathcal{R}, \Gamma \vdash v : B$ is satisfied in M with I , written $M, I \models \Lambda$, iff if $M, I \models wRu$ for all $wRu \in \mathcal{R}$ and $M, I \models w : A$ for all $w : A \in \Gamma$, then $M, I \models v : B$. A labelled sequent Λ is falsified in M with I iff $M, I \not\models \Lambda$, that is, Λ is not satisfied by M with I .

Last, a labelled sequent Λ is \mathcal{A} -valid, written $\models_{\mathcal{A}} \Lambda$, iff it is satisfiable in every bi-relational \mathcal{A} -model M with every interpretation function I . We say that a labelled sequent Λ is \mathcal{A} -invalid iff $\not\models_{\mathcal{A}} \Lambda$, i.e. Λ is not \mathcal{A} -valid.

Theorem 2 ($\mathbf{L}_{\Box \diamond}(\mathcal{A})$ Soundness and Completeness). $\mathcal{R}, \Gamma \vdash w : A$ is derivable in $\mathbf{L}_{\Box \diamond}(\mathcal{A})$ iff $\mathcal{R}, \Gamma \vdash w : A$ is \mathcal{A} -valid.

Proof. Follows from Thm. 7.2.1 and Thm. 8.1.4 of [31]. \square

5 Structural Refinement

We show how to *structurally refine* the labelled systems introduced in the previous section, that is, we implement a methodology introduced and applied in [6,

21–23] (referred to as *structural refinement*, or *refinement* more simply) for simplifying labelled systems and/or permitting the extraction of nested systems. The methodology consists of eliminating structural rules (viz. the $(S_{n,k})$ rules in our setting) through the addition of *propagation rules* (cf. [4, 10, 31]) to the labelled calculi, begetting systems that are translatable into nested systems.

The propagation rules we introduce are based on those of [6, 17, 22, 23, 33], and operate by viewing a labelled sequent as an automaton, allowing for the propagation of a formula (when applied bottom-up) from a label w to a label u given that a certain path of relational atoms exists between w and u (corresponding to a string generated by an \mathcal{A} -grammar). We note that Simpson likewise introduced a variation of these rules, named $(\diamond R)_{\mathcal{T}_H}$ and $(\square L)_{\mathcal{T}_H}$ (see [31, p. 126]), by closing the relational atoms of a sequent under the frame conditions related to each HSL $\phi(n, k) \in \mathcal{A}$. We opt to use propagation rules based on formal grammars however because such rules permit the formulation of nested systems outside the class of HSL extensions of IK, thus setting the stage for the construction of nested systems for even broader classes of logics in future work.⁴

The definition of our propagation rules is built atop the notions introduced in the following two definitions:

Definition 10 (Propagation Graph). *The propagation graph $PG(\mathcal{R})$ of a multiset of relational atoms \mathcal{R} is defined recursively on the structure of \mathcal{R} :*

- $PG(\emptyset) := (\emptyset, \emptyset)$;
- $PG(wRu) := (\{w, u\}, \{(w, \diamond, u), (u, \blacklozenge, w)\})$;
- $PG(\mathcal{R}_1, \mathcal{R}_2) := (V_1 \cup V_2, E_1 \cup E_2)$ where $PG_x(\mathcal{R}_i) = (V_i, E_i)$.

We will often write $w \in PG(\mathcal{R})$ to mean $w \in V$, and $(w, \langle ? \rangle, u) \in PG(\mathcal{R})$ to mean $(w, \langle ? \rangle, u) \in E$.

Definition 11 (Propagation Path). *We define a propagation path from w_1 to w_n in $PG(\mathcal{R}) := (V, E)$ to be a sequence of the following form:*

$$\pi(w_1, w_n) := w_1, \langle ? \rangle_1, w_2, \langle ? \rangle_2, \dots, \langle ? \rangle_{n-1}, w_n$$

such that $(w_1, \langle ? \rangle_1, w_2), (w_2, \langle ? \rangle_2, w_3), \dots, (w_{n-1}, \langle ? \rangle_{n-1}, w_n) \in E$. Given a propagation path of the above form, we define its converse as shown below top and its string as shown below bottom:

$$\pi^{-1}(w_n, w_1) := w_n, \langle ? \rangle_{n-1}^{-1}, w_{n-1}, \langle ? \rangle_{n-2}^{-1}, \dots, \langle ? \rangle_1^{-1}, w_1$$

$$s_\pi(w_1, w_n) := \langle ? \rangle_1 \langle ? \rangle_2 \cdots \langle ? \rangle_{n-1}$$

Last, we let $\lambda(w, w) := w$ represent an empty path with the string of the empty path defined as $s_\lambda(w, w) := \varepsilon$.

⁴ For instance, we could define our propagation rules relative to the formal grammar $\{\diamond \rightarrow \diamond\blacklozenge\}$, which would give a calculus for a logic outside the class of HSL extensions of IK.

$$\frac{\mathcal{R}, \Gamma \vdash u : A}{\mathcal{R}, \Gamma \vdash w : \diamond A} (p_\diamond) \text{ only if } \exists \pi(w, u) \in PG(\mathcal{R})(s_\pi(w, u) \in L_{g(\mathcal{A})}(\diamond))$$

$$\frac{\mathcal{R}, \Gamma, w : \square A, u : A \vdash v : B}{\mathcal{R}, \Gamma, w : \square A \vdash v : B} (p_\square) \text{ only if } \exists \pi(w, u) \in PG(\mathcal{R})(s_\pi(w, u) \in L_{g(\mathcal{A})}(\diamond))$$

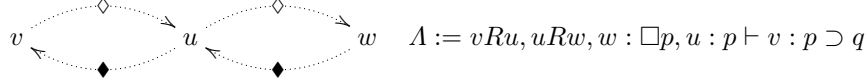
Fig. 3. Propagation rules.

We are now in a position to define the operation of our propagation rules (p_\diamond) and (p_\square) , which are displayed in Fig. 3. Each propagation rule (p_\diamond) and (p_\square) is applicable only if *there exists a propagation path* $\pi(w, u)$ *from* w *to* u *in the propagation graph* $PG(\mathcal{R})$ *such that the string* $s_\pi(w, u)$ *is in the language* $L_{g(\mathcal{A})}(\diamond)$. We express this statement compactly by making use of its equivalent first-order representation:

$$\exists \pi(w, u) \in PG(\mathcal{R})(s_\pi(w, u) \in L_{g(\mathcal{A})}(\diamond))$$

We provide further intuition regarding such rules by means of an example:

Example 1. Let $\mathcal{R} := vRu, uRw$. We give a graphical depiction of $PG(\mathcal{R})$:



Let $\mathcal{A} := \{(\diamond^2 \square A \supset \square^1 A) \wedge (\diamond^1 A \supset \square^2 \diamond A)\}$, so that the corresponding \mathcal{A} -grammar is $g(\mathcal{A}) = \{\diamond \rightarrow \blacklozenge \blacklozenge \diamond, \blacklozenge \rightarrow \blacklozenge \diamond \diamond\}$. Then, the path $\pi(w, u) := w, \blacklozenge, u, \blacklozenge, v, \diamond, u$ exists between w and u . The first production rule of $g(\mathcal{A})$ implies that $s_\pi(w, u) = \blacklozenge \blacklozenge \diamond \in L_{g(\mathcal{A})}(\diamond)$. Therefore, we are permitted to (top-down) apply the propagation rule (p_\square) to A to delete the labelled formula $u : p$, letting us derive $vRu, uRw, w : \square p \vdash v : p \supset q$.

Remark 2. The (\diamond_r) and (\square_l) rules are instances of (p_\diamond) and (p_\square) , respectively.

Definition 12 (Refined Labelled Calculus). *We define the refined labelled calculus* $\text{IK}(\mathcal{A})\text{L} := \text{L}_{\square \diamond}(\mathcal{A}) + \{(p_\diamond), (p_\square)\} - \{(S_{n,k}) \mid \phi(n, k) \in \mathcal{A}\}$.

We show that each calculus $\text{IK}(\mathcal{A})\text{L}$ is complete by means of a proof transformation procedure. That is, we show that through the elimination of structural rules we can transform a proof in $\text{L}_{\square \diamond}(\mathcal{A})$ into a proof in $\text{IK}(\mathcal{A})\text{L}$. We note that Simpson proved a similar result, showing that labelled derivations with structural rules are transformable into derivations with his propagation rules $(\diamond R)_{\mathcal{T}_H}$ and $(\square L)_{\mathcal{T}_H}$ (see [31, Sect. 7.2]). In our context, the proof of structural rule eliminability requires more complex methods however due to the use of our new propagation rules that are parameterized with formal grammars. We first prove two crucial lemmata, and then show the elimination result.

Lemma 1. *Let $\mathcal{R}_1 := \mathcal{R}, wR^n u, wR^k v, uRv$ and $\mathcal{R}_2 := \mathcal{R}, wR^n u, wR^k v$. Suppose we are given a derivation in $\mathbf{L}_{\square\Diamond}(\mathcal{A}) + \{(p_\Diamond), (p_\square)\}$ ending with:*

$$\frac{\frac{\mathcal{R}, wR^n u, wR^k v, uRv, \Gamma \vdash z : A}{\mathcal{R}, wR^n u, wR^k v, uRv, \Gamma \vdash x : \Diamond A} (p_\Diamond)}{\mathcal{R}, wR^n u, wR^k v, \Gamma \vdash x : \Diamond A} (S_{n,k})$$

where the side condition $\exists \pi(x, z) \in PG(\mathcal{R}_1)(s_\pi(x, z) \in L_{g(\mathcal{A})}(\Diamond))$ holds due to (p_\Diamond) . Then, $\exists \pi'(x, z) \in PG(\mathcal{R}_2)(s_{\pi'}(x, z) \in L_{g(\mathcal{A})}(\Diamond))$, that is to say, the $(S_{n,k})$ rule is permutable with the (p_\Diamond) rule.

Proof. We have two cases: either (i) the relational atom uRv is not active in the (p_\Diamond) inference, or (ii) it is. Since (i) is easily resolved, we show (ii).

Let us suppose that the relational atom uRv is active in (p_\Diamond) , i.e. uRv occurs along the propagation path $\pi(x, z)$. To prove the claim, we need to show that $\exists \pi'(x, z) \in PG(\mathcal{R}_2)(s_{\pi'}(x, z) \in L_{g(\mathcal{A})}(\Diamond))$. Therefore, we construct such a propagation path by performing the following operations on $\pi(x, z)$:

- replace each occurrence of u, \Diamond, v in $PG(\mathcal{R}_1)$ with

$$u, \blacklozenge, u_1, \dots, u_{n-1}, \blacklozenge, w, \Diamond, w_1, \dots, w_{k-1}, \Diamond, v;$$

- replace each occurrence of v, \blacklozenge, u in $PG(\mathcal{R}_1)$ with

$$v, \blacklozenge, w_{k-1}, \dots, w_1, \blacklozenge, w, \Diamond, u_{n-1}, \dots, u_1, \Diamond, u.$$

We let $\pi'(x, z)$ denote the path obtained by performing the above operations on $\pi(x, z)$, and note that first half of the first propagation path and the second half of the second propagation path correspond to the edges $(u, \blacklozenge, u_1), \dots, (u_{n-1}, \blacklozenge, w) \in PG(\mathcal{R}_1)$ and $(w, \Diamond, u_{n-1}), \dots, (u_1, \Diamond, u) \in PG(\mathcal{R}_1)$, respectively, obtained from the relational atoms $wR^n u \in \mathcal{R}_1$, whereas the second half of the first propagation path and the first half of the second propagation path correspond to the edges $(w, \Diamond, w_1), \dots, (w_{k-1}, \Diamond, v) \in PG(\mathcal{R}_1)$ and $(v, \blacklozenge, w_{k-1}), \dots, (w_1, \blacklozenge, w) \in PG(\mathcal{R}_1)$, respectively, obtained from the edges $wR^k v \in \mathcal{R}_1$ (by Def. 13). Since the sole difference between $PG(\mathcal{R}_1)$ and $PG(\mathcal{R}_2)$ is that the former is guaranteed to contain the edges (u, \Diamond, v) and (v, \blacklozenge, u) obtained from uRv , while the latter is not, and since $\pi'(x, z)$ omits the use of such edges (i.e. u, \Diamond, v and v, \blacklozenge, u do not occur in $\pi'(x, z)$), we have that $\pi'(x, z)$ is a propagation path in $PG(\mathcal{R}_2)$.

To complete the proof, we need to additionally show that $s_{\pi'}(x, z) \in L_{g(\mathcal{A})}(\Diamond)$. By assumption, $s_\pi(x, z) \in L_{g(\mathcal{A})}(\Diamond)$, which implies that $\Diamond \xrightarrow{*}_{g(\mathcal{A})} s_\pi(x, z)$ by Def. 8. Since $(S_{n,k})$ is a rule in $\mathbf{L}_{\square\Diamond}(\mathcal{A})$, it follows that $\Diamond \xrightarrow{\blacklozenge^n \Diamond^k}$ and $\blacklozenge \xrightarrow{\blacklozenge^k \Diamond^n} \in g(\mathcal{A})$ by Def. 7. If we apply $\Diamond \xrightarrow{\blacklozenge^n \Diamond^k}$ to each occurrence of \Diamond in $s_\pi(x, z)$ corresponding to the edge (u, \Diamond, v) (and relational atom uRv), and apply $\blacklozenge \xrightarrow{\blacklozenge^k \Diamond^n}$ to each occurrence of \blacklozenge in $s_\pi(x, z)$ corresponding to the edge (v, \blacklozenge, u) (and relational atom uRv), we obtain the string $s_{\pi'}(x, z)$ and show that $\Diamond \xrightarrow{*}_{g(\mathcal{A})} s_{\pi'}(x, z)$, i.e. $s_{\pi'}(x, z) \in L_{g(\mathcal{A})}(\Diamond)$. \square

Lemma 2. *Let $\mathcal{R}_1 := \mathcal{R}, wR^n u, wR^k v, uRv$ and $\mathcal{R}_2 := \mathcal{R}, wR^n u, wR^k v$. Suppose we are given a derivation in $\mathbf{L}_{\square\Diamond}(\mathcal{A}) + \{(p_\Diamond), (p_\square)\}$ ending with:*

$$\frac{\mathcal{R}, wR^n u, wR^k v, uRv, x : \Box A, y : A, \Gamma \vdash z : C}{\mathcal{R}, wR^n u, wR^k v, uRv, x : \Box A, \Gamma \vdash z : C} (p_{\Box})$$

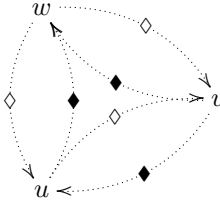
$$\frac{\mathcal{R}, wR^n u, wR^k v, uRv, x : \Box A, \Gamma \vdash z : C}{\mathcal{R}, wR^n u, wR^k v, x : \Box A, \Gamma \vdash z : C} (S_{n,k})$$

where the side condition $\exists \pi(x, y) \in PG(\mathcal{R}_1)(s_{\pi}(x, y) \in L_{g(\mathcal{A})}(\diamond))$ holds due to (p_{\Box}) . Then, $\exists \pi'(x, y) \in PG(\mathcal{R}_2)(s_{\pi'}(x, y) \in L_{g(\mathcal{A})}(\diamond))$, that is to say, the $(S_{n,k})$ rule is permutable with the (p_{\Box}) rule.

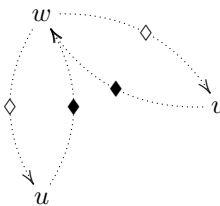
Proof. Similar to the proof of Lem. 1 above. \square

To improve the comprehensibility of the above lemmata, we provide an example of permuting an instance of the structural rule $(S_{n,k})$ above an instance of a propagation rule.

Example 2. Let $\mathcal{A} := \{(\diamond \Box A \supset \Box A) \wedge (\diamond A \supset \Box \diamond A)\}$ so that the \mathcal{A} -grammar $g(\mathcal{A}) = \{\diamond \rightarrow \blacklozenge \diamond, \blacklozenge \rightarrow \blacklozenge \diamond\}$. In the top derivation below, we assume that (p_{\diamond}) is applied due to the existence of the propagation path $\pi(u, v) = u, \diamond, v$ in $PG(wRu, wRv, uRv)$, where $s_{\pi}(u, v) = \diamond \in L_{g(\mathcal{A})}(\diamond)$ by Def. 8. The propagation graph $PG(wRu, wRv, uRv)$ corresponding to the top sequent of the derivation shown below left is shown below right:

$$\frac{\frac{\frac{}{wRu, wRv, uRv, u : p \vdash u : p} (id)}{wRu, wRv, uRv, u : p \vdash v : \diamond p} (p_{\diamond})}{wRu, wRv, u : p \vdash v : \diamond p} (S_{1,1})$$


If we apply $\diamond \rightarrow \blacklozenge \diamond \in g(\mathcal{A})$ to $s_{\pi}(u, v) = \diamond$, then we obtain the string $\blacklozenge \diamond$. Hence, $\diamond \rightarrow_{g(\mathcal{A})}^* \blacklozenge \diamond$, i.e. $\blacklozenge \diamond \in L_{g(\mathcal{A})}(\diamond)$, meaning that a propagation path $\pi'(u, v) (= u, \blacklozenge, w, \diamond, v)$ exists in $PG(wRu, wRv)$ such that $s_{\pi'}(u, v) = \blacklozenge \diamond \in L_{g(\mathcal{A})}(\diamond)$. We may therefore apply $(a_{1,1})$ and then (p_{\diamond}) as shown below left; the propagation graph $PG(wRu, wRv)$ is shown below right:

$$\frac{\frac{\frac{}{wRu, wRv, uRv, u : p \vdash u : p} (id)}{wRu, wRv, u : p \vdash u : p} (S_{1,1})}{wRu, wRv, u : p \vdash v : \diamond p} (p_{\diamond})$$


Theorem 3. *Every derivation in $\mathbf{L}_{\Box \diamond}(\mathcal{A})$ can be algorithmically transformed into a derivation in $\mathbf{IK}(\mathcal{A})\mathbf{L}$.*

Proof. We consider a derivation in $\mathbf{L}_{\Box \diamond}(\mathcal{A})$, which is a derivation in $\mathbf{L}_{\Box \diamond}(\mathcal{A}) + \{(p_{\diamond}), (p_{\Box})\}$. By Remark 2, each instance of (\diamond_r) and (\Box_l) can be replaced by

a (p_\diamond) or (p_\square) instance, respectively, meaning we may assume our derivation in $\mathbf{L}_{\square\diamond}(\mathcal{A}) + \{(p_\diamond), (p_\square)\}$ is free of (\diamond_r) and (\square_l) instances. We show that the derivation can be transformed into a derivation in $\mathbf{IK}(\mathcal{A})\mathbf{L}$ by induction on its height, that is, we consider a topmost occurrence of a structural rule $(S_{n,k})$ and show that it can be eliminated. We obtain a derivation in $\mathbf{IK}(\mathcal{A})\mathbf{L}$ by successively eliminating topmost instances of $(S_{n,k})$ rules.

Base case. Observe that any application of $(S_{n,k})$ to (id) or (\perp_l) yields another instance of the rule.

Inductive step. It is straightforward to verify that any instance of $(S_{n,k})$ freely permutes above instances of all rules in $\mathbf{L}_{\square\diamond}(\mathcal{A}) + \{(p_\diamond), (p_\square)\}$ with the exception of $(S_{n,k})$, (p_\diamond) , and (p_\square) (this follows from the fact that all other rules do not have active relational atoms in their conclusion). Since we are considering a topmost application of $(S_{n,k})$, we need not consider the permutation of $(S_{n,k})$ above another instance of $(S_{n,k})$. The last two cases of permuting $(S_{n,k})$ above (p_\diamond) and (p_\square) follow from Lem. 1 and 2, respectively. \square

Theorem 4 (IK(A)L Soundness and Completeness). $\mathcal{R}, \Gamma \vdash w : A$ is derivable in $\mathbf{IK}(\mathcal{A})\mathbf{L}$ iff $\mathcal{R}, \Gamma \vdash w : A$ is \mathcal{A} -valid.

Proof. The forward direction (soundness) is shown by induction on the height of the given derivation, and the backward direction (completeness) follows from Thm. 2 and 3. \square

6 Nested Sequent Systems

In our setting, nested sequents are taken to be trees of multisets of formulae containing a unique formula that occupies a special status. We utilize the nested sequents of [32], but note that the data structure underlying such sequents was originally used in [13], and is similar to the nested sequents for classical modal logics employed in [2]. Following [32], we mark the special, unique formula with a white circle \circ indicating that the formula is of *output polarity*, and mark the other formulae with a black circle \bullet indicating that the formulae are of *input polarity*. A nested sequent Σ is defined via the following BNF grammars:

$$\Sigma ::= \Delta, \Pi \quad \Delta ::= A_1^\bullet, \dots, A_n^\bullet, [\Delta_1], \dots, [\Delta_k] \quad \Pi ::= A^\circ \mid [\Sigma]$$

We assume that the comma operator associates and commutes, implying that such sequents are truly trees of multisets of formulae, and we let the *empty sequent* be the empty multiset \emptyset . We refer to a sequent in the shape of Δ (which contains only input formulae) as an *LHS-sequent*, a sequent in the shape of Π as an *RHS-sequent*, and a sequent Σ as a *full sequent*. We use both Σ and Δ to denote LHS- and full sequents with the context differentiating the usage.

As for classical modal logics (e.g. [2, 17]), we define a *context* $\Sigma\{ \} \cdots \{ \}$ to be a nested sequent with some number of holes $\{ \}$ in the place of formulae. This gives rise to two types of contexts: *input contexts*, which require holes to be filled with LHS-sequents to obtain a full sequent, and *output contexts*, which

require a single hole to be filled with an RHS-sequent and the remaining holes to be filled with LHS-sequents to obtain a full sequent. We also define the *output pruning* of an input context $\Sigma\{\}\cdots\{\}$ or full sequent Σ , denoted $\Sigma^\downarrow\{\}\cdots\{\}$ and Σ^\downarrow respectively, to be the same context or sequent with the unique output formula deleted. We note that all of the above terminology is due to [32].

Example 3. Let $\Sigma_1\{\} := p^\bullet, [\diamond q^\bullet, \{\}]$, $\Sigma_2\{\} := p^\bullet, [\diamond q^\circ, \{\}]$, $\Delta_1 := \perp^\bullet, [q \supset r^\circ]$, and $\Delta_2 := \perp^\bullet, [q \supset r^\bullet]$. Observe that neither $\Sigma_1\{\Delta_2\}$ nor $\Sigma_2\{\Delta_1\}$ are full sequents since the former has no output formula and the latter has two output formulae. Conversely, both $\Sigma_1\{\Delta_1\}$ and $\Sigma_2\{\Delta_2\}$ are full sequents.

Our nested sequent systems are presented in Fig. 4 and are generalizations of those for the logics of the intuitionistic modal cube given in [32]. For example, a nested sequent system for the intuitionistic modal logic $\mathbf{IK} + \{(\diamond^0 \Box A \supset \Box^3 A) \wedge (\diamond^3 A \supset \Box^0 \diamond A)\}$ incorporating the 3-to-1 transitivity axiom, which falls outside the intuitionistic modal cube, is obtained by employing the \mathcal{A} -grammar $g(\mathcal{A}) = \{\diamond \rightarrow \diamond \diamond \diamond, \blacklozenge \rightarrow \blacklozenge \blacklozenge \blacklozenge\}$ in the propagation rules (p_\diamond) and (p_\Box) . As in the previous section, our propagation rules (p_\diamond) and (p_\Box) rely on auxiliary notions (e.g. propagation graphs and paths), which we define for nested sequents.

Definition 13 (Propagation Graph/Path). *Let w be the label assigned to the root of the nested sequent Σ . We define the propagation graph $PG(\Sigma) := PG_w(\Sigma)$ of a nested sequent Σ recursively on the structure of the nested sequent.*

- $PG_u(\emptyset) := (\emptyset, \emptyset, \emptyset)$;
- $PG_u(A) := (\emptyset, \emptyset, \{(u, A)\})$ with $A \in \{A^\bullet, A^\circ\}$;
- $PG_u(\Delta_1, \Delta_2) := (V_1 \cup V_2, E_1 \cup E_2, L_1 \cup L_2)$ where $PG_u(\Delta_i) = (V_i, E_i, L_i)$;
- $PG_u([\Sigma]) := (V \cup \{u\}, E \cup \{(u, \diamond, v), (v, \blacklozenge, u)\}, L)$ where $PG_v(\Sigma) = (V, E, L)$ and v is fresh.

We will often write $u \in PG(\Sigma)$ to mean $u \in V$, and $(u, \langle ? \rangle, v) \in PG(\Sigma)$ to mean $(u, \langle ? \rangle, v) \in E$. Also, we define propagation paths, converses of propagation paths, and the string of a propagation path as in Def. 11.

For input or output formulae A and B , we use the notation $\Sigma\{A\}_w\{B\}_u$ to mean that $(w, A), (u, B) \in L$ in $PG(\Sigma)$. For example, if $\Sigma := p \supset q^\circ, [p^\bullet, [\Box p^\bullet]]$ with $PG(\Sigma) := (V, E, L)$ and $(v, p \supset q^\circ), (u, p^\bullet), (w, \Box p^\bullet) \in L$, then both $\Sigma\{p \supset q^\circ\}_v\{\Box p^\bullet\}_w$ and $\Sigma\{p^\bullet\}_u\{p \supset q^\circ\}_v$ are valid representations of Σ in our notation.

We now prove that proofs can be translated between our refined labelled and nested systems. In order to prove this fact, we make use of the following definitions, which are based on the work of [18, 21].

Definition 14 (Labelled Tree Sequent/Derivation). *We define a labelled tree sequent to be a labelled sequent $\Lambda := \mathcal{R}, \Gamma \vdash w : A$ such that \mathcal{R} forms a tree and all labels in $\Gamma, w : A$ occur in \mathcal{R} . We define a labelled tree derivation to be a proof containing only labelled tree sequents. We say that a labelled tree derivation has the fixed root property iff every labelled sequent in the derivation has the same root.*

$$\begin{array}{c}
 \frac{}{\Sigma\{\perp^\bullet\}} (\perp^\bullet) \quad \frac{}{\Sigma\{p^\bullet, p^\circ\}} (id) \quad \frac{\Sigma\{A^\bullet, B^\bullet\}}{\Sigma\{A \wedge B^\bullet\}} (\wedge^\bullet) \quad \frac{\Sigma\{A^\circ\} \quad \Sigma\{B^\circ\}}{\Sigma\{A \wedge B^\circ\}} (\wedge^\circ) \\
 \\
 \frac{\Sigma\{A^\bullet\} \quad \Sigma\{B^\bullet\}}{\Sigma\{A \vee B^\bullet\}} (\vee^\bullet) \quad \frac{\Sigma\{A^\bullet, B^\circ\}}{\Sigma\{A \supset B^\circ\}} (\supset^\circ) \quad \frac{\Sigma\{A_i^\circ\}}{\Sigma\{A_1 \vee A_2^\circ\}} (\vee^\circ) \quad i \in \{1, 2\} \\
 \\
 \frac{\Sigma^\perp\{A \supset B^\bullet, A^\circ\} \quad \Sigma\{B^\bullet\}}{\Sigma\{A \supset B^\bullet\}} (\supset^\bullet) \quad \frac{\Sigma\{[A^\circ]\}}{\Sigma\{\Box A^\circ\}} (\Box^\circ) \quad \frac{\Sigma\{[A^\bullet]\}}{\Sigma\{\Diamond A^\bullet\}} (\Diamond^\bullet) \quad \frac{\Sigma\{[\emptyset]\}}{\Sigma\{\emptyset\}} (d) \\
 \\
 \frac{\Sigma\{\Delta_1\}_w\{A^\circ, \Delta_2\}_u}{\Sigma\{\Diamond A^\circ, \Delta_1\}_w\{\Delta_2\}_u} (p_\Diamond) \quad \text{only if } \exists \pi(w, u) \in PG(\Sigma)(s_\pi(w, u) \in L_{g(A)}(\Diamond)) \\
 \\
 \frac{\Sigma\{\Box A^\bullet, \Delta_1\}_w\{A^\bullet, \Delta_2\}_u}{\Sigma\{\Box A^\bullet, \Delta_1\}_w\{\Delta_2\}_u} (p_\Box) \quad \text{only if } \exists \pi(w, u) \in PG(\Sigma)(s_\pi(w, u) \in L_{g(A)}(\Diamond))
 \end{array}$$

Fig. 4. The nested sequent calculi $\mathbf{NIK}(\mathcal{A})$. The (d) rule occurs in a calculus iff $D \in \mathcal{A}$.

We now define our translation functions which transform a *full* nested sequent into a labelled tree sequent, and vice-versa. Our translations additionally depend on *sequent compositions* and *labelled restrictions*. If $A_1 := \mathcal{R}_1, \Gamma_1 \vdash \Gamma'_1$ and $A_2 := \mathcal{R}_2, \Gamma_2 \vdash \Gamma'_2$, then we define its sequent composition $A_1 \otimes A_2 := \mathcal{R}_1, \mathcal{R}_2, \Gamma_1, \Gamma_2 \vdash \Gamma'_1, \Gamma'_2$. Given that Γ is a multiset of labelled formulae, we define the labelled restriction $\Gamma \upharpoonright w := \{A \mid w : A \in \Gamma\}$, and we note that if w is not a label in Γ , then $\Gamma \upharpoonright w := \emptyset$. Moreover, for a multiset A_1, \dots, A_n of formulae, we define $(A_1, \dots, A_n)^* := A_1^*, \dots, A_n^*$ and $(\emptyset)^* := \emptyset$, where $*$ $\in \{\bullet, \circ\}$.

Definition 15 (Translation \mathfrak{L}). We define $\mathfrak{L}_w(\Sigma) := \mathcal{R}, \Gamma \vdash u : A$ as follows:

- $\mathfrak{L}_v(\emptyset) := \emptyset \vdash \emptyset$
- $\mathfrak{L}_v(A^\bullet) := v : A \vdash \emptyset$
- $\mathfrak{L}_v(A^\circ) := \emptyset \vdash v : A$
- $\mathfrak{L}_v(\Delta_1, \Delta_2) := \mathfrak{L}_v(\Delta_1) \otimes \mathfrak{L}_v(\Delta_2)$
- $\mathfrak{L}_v([\Sigma]) := (vRu \vdash \emptyset) \otimes \mathfrak{L}_u(\Sigma)$ with u fresh

We note that since Σ is a full sequent, the obtained labelled sequent will contain a single labelled formula in its consequent.

Example 4. We let $\Sigma := p \supset q^\circ, [p^\bullet], [\Box p^\bullet]$ and show the output labelled sequent under the translation \mathfrak{L} .

$$\mathfrak{L}_w(\Sigma) = wRv, vRu, v : p, u : \Box p \vdash w : p \supset q$$

Definition 16 (Translation \mathfrak{N}). Let $\Lambda := \mathcal{R}, \Gamma \vdash w : A$ be a labelled tree sequent with root u . We define $\Lambda_1 \subseteq \Lambda$ iff there exists a labelled tree sequent Λ_2 such that $\Lambda = \Lambda_1 \otimes \Lambda_2$. Let us further define Λ_u to be the unique labelled tree sequent rooted at the label u such that $\Lambda_u \subseteq \Lambda$. We define $\mathfrak{N}(\Lambda) := \mathfrak{N}_u(\Lambda)$ recursively on the tree structure of Λ :

$$\mathfrak{N}_v(\Lambda) := \begin{cases} (\Gamma \upharpoonright v)^\bullet, (w : A \upharpoonright v)^\circ & \text{if } \mathcal{R} = \emptyset; \\ (\Gamma \upharpoonright v)^\bullet, (w : A \upharpoonright v)^\circ, [\mathfrak{N}_{z_1}(\Lambda_{z_1})], \dots, [\mathfrak{N}_{z_n}(\Lambda_{z_n})] & \text{otherwise.} \end{cases}$$

$$\frac{\Sigma}{[\Sigma]} (n) \quad \frac{\Sigma\{\emptyset\}}{\Sigma\{\Delta\}} (w) \quad \frac{\Sigma\{A^\bullet, A^\bullet\}}{\Sigma\{A^\bullet\}} (c) \quad \frac{\Sigma\{[\Delta_1], [\Delta_2]\}}{\Sigma\{[\Delta_1, \Delta_2]\}} (m)$$

Fig. 5. Height-preserving (hp-)admissible structural rules.

In the second case above, we assume that vRz_1, \dots, vRz_n are all of the relational atoms occurring in the input sequent which have the form vRx .

Example 5. We let $\Lambda := wRv, vRu, v : p, u : \Box p \vdash w : p \supset q$ and show the output nested sequent under the translation \mathfrak{N} .

$$\mathfrak{N}(\Lambda) = \mathfrak{N}_w(\Lambda) = p \supset q^\circ, [p^\bullet, [\Box p^\bullet]]$$

Lemma 3. *Every proof in $\text{IK}(\mathcal{A})\text{L}$ of a labelled tree sequent is a labelled tree proof with the fixed root property.*

Proof. The lemma follows from the observation that if any rule of $\text{IK}(\mathcal{A})\text{L}$ is applied bottom-up to a labelled tree sequent, then each premise is a labelled tree sequent with the same root. \square

Theorem 5. *Every proof of a labelled tree sequent in $\text{IK}(\mathcal{A})\text{L}$ is transformable into a proof in $\text{NIK}(\mathcal{A})$, and vice-versa.*

Proof. Follows from Lem. 3, and the fact that the rules of $\text{IK}(\mathcal{A})\text{L}$ and $\text{NIK}(\mathcal{A})$ are translations of one another under the \mathfrak{N} and \mathfrak{L} functions. \square

Theorem 6 (NIK(\mathcal{A}) Soundness and Completeness). *A formula A is derivable in $\text{NIK}(\mathcal{A})$ iff A is \mathcal{A} -valid.*

Proof. Follows from Thm. 4 and 5. \square

Theorem 7. *The rules (n), (w), (c), and (m) are hp-admissible in $\text{NIK}(\mathcal{A})$.*

Proof. By induction on the height of the given derivation; the proofs are similar to those of [2, Lem. 1] and [32, Lem. 6.4]. For the (m) rule, we note that propagation paths are preserved from premise to conclusion (cf. [17, Fig. 12]), showing that the rule can be permuted above (p_\diamond) and (p_\Box). \square

7 Conclusion

In this paper, we employed the structural refinement methodology to extract nested sequent systems for a broad class of intuitionistic modal logics. The attainment of such systems answers the open problem of [25] to a large extent by showing how to transform axioms (namely, HSLs) into propagation/logical rules as well as how to obtain nested sequent systems for logics outside the intuitionistic modal cube. We aim to write proof-search algorithms in future work based on our nested systems which utilize saturation conditions and loop-checking (cf. [11, 22, 33]) to provide decision procedures for logics within the class considered. Our primary concern will be to establish the decidability of transitive extensions of IK , which has remained a longstanding open problem [15, 31].

References

1. Bierman, G.M., de Paiva, V.C.V.: On an intuitionistic modal logic. *Studia Logica: An International Journal for Symbolic Logic* **65**(3), 383–416 (2000), <http://www.jstor.org/stable/20016199>
2. Brünnler, K.: Deep sequent systems for modal logic. *Arch. Math. Log.* **48**(6), 551–577 (2009). <https://doi.org/10.1007/s00153-009-0137-3>, <https://doi.org/10.1007/s00153-009-0137-3>
3. Bull, R.A.: Cut elimination for propositional dynamic logic without *. *Z. Math. Logik Grundlag. Math.* **38**(2), 85–100 (1992)
4. Castilho, M.A., del Cerro, L.F., Gasquet, O., Herzig, A.: Modal tableaux with propagation rules and structural rules. *Fundamenta Informaticae* **32**(3, 4), 281–297 (1997)
5. Ciabattoni, A., Lyon, T., Ramanayake, R.: From display to labelled proofs for tense logics. In: Artëmov, S.N., Nerode, A. (eds.) *Logical Foundations of Computer Science - International Symposium, LFCS 2018, Deerfield Beach, FL, USA, January 8-11, 2018, Proceedings. Lecture Notes in Computer Science*, vol. 10703, pp. 120–139. Springer (2018). https://doi.org/10.1007/978-3-319-72056-2_8, https://doi.org/10.1007/978-3-319-72056-2_8
6. Ciabattoni, A., Lyon, T., Ramanayake, R., Tiu, A.: Display to labelled proofs and back again for tense logics. *ACM Transactions on Computational Logic* **22**(3), 1–31 (2021). <https://doi.org/https://doi.org/10.1145/3460492>
7. Davies, R., Pfenning, F.: A modal analysis of staged computation. *J. ACM* **48**(3), 555–604 (May 2001). <https://doi.org/10.1145/382780.382785>
8. Fairtlough, M., Mendler, M.: An intuitionistic modal logic with applications to the formal verification of hardware. In: Pacholski, L., Tiuryn, J. (eds.) *Computer Science Logic*. pp. 354–368. Springer Berlin Heidelberg, Berlin, Heidelberg (1995)
9. Fitch, F.B.: Intuitionistic modal logic with quantifiers. *Portugaliae mathematica* **7**(2), 113–118 (1948), <http://eudml.org/doc/114664>
10. Fitting, M.: Tableau methods of proof for modal logics. *Notre Dame Journal of Formal Logic* **13**(2), 237–247 (1972)
11. Fitting, M.: *Proof methods for modal and intuitionistic logics*, vol. 169. Springer Science & Business Media (1983)
12. Fitting, M., Kuznets, R.: Modal interpolation via nested sequents. *Annals of pure and applied logic* **166**(3), 274–305 (2015). <https://doi.org/10.1016/j.apal.2014.11.002>
13. Galmiche, D., Salhi, Y.: Label-free natural deduction systems for intuitionistic and classical modal logics. *Journal of Applied Non-Classical Logics* **20**(4), 373–421 (2010). <https://doi.org/10.3166/jancl.20.373-421>
14. Galmiche, D., Salhi, Y.: Tree-sequent calculi and decision procedures for intuitionistic modal logics. *Journal of Logic and Computation* **28**(5), 967–989 (06 2015). <https://doi.org/10.1093/logcom/exv039>
15. Girlando, M., Straßburger, L.: Moin: A nested sequent theorem prover for intuitionistic modal logics (system description). In: Peltier, N., Sofronie-Stokkermans, V. (eds.) *Automated Reasoning*. pp. 398–407. Springer International Publishing, Cham (2020)
16. Goré, R., Postniece, L., Tiu, A.: Cut-elimination and proof-search for bi-intuitionistic logic using nested sequents. In: Areces, C., Goldblatt, R. (eds.) *Advances in Modal Logic 7, papers from the seventh conference on "Advances in Modal Logic," held in Nancy, France, 9-12 September 2008*. pp. 43–66. College Publications (2008), <http://www.aiml.net/volumes/volume7/Gore-Postniece-Tiu.pdf>

17. Goré, R., Postniece, L., Tiu, A.: On the correspondence between display postulates and deep inference in nested sequent calculi for tense logics. *Log. Methods Comput. Sci.* **7**(2) (2011). [https://doi.org/10.2168/LMCS-7\(2:8\)2011](https://doi.org/10.2168/LMCS-7(2:8)2011), [https://doi.org/10.2168/LMCS-7\(2:8\)2011](https://doi.org/10.2168/LMCS-7(2:8)2011)
18. Goré, R., Ramanayake, R.: Labelled tree sequents, tree hypersequents and nested (deep) sequents. In: Bolander, T., Braüner, T., Ghilardi, S., Moss, L.S. (eds.) *Advances in Modal Logic 9, papers from the ninth conference on "Advances in Modal Logic,"* held in Copenhagen, Denmark, 22-25 August 2012. pp. 279–299. College Publications (2012), <http://www.aiml.net/volumes/volume9/Gore-Ramanayake.pdf>
19. Kashima, R.: Cut-free sequent calculi for some tense logics. *Studia Logica* **53**(1), 119–135 (1994)
20. Lemmon, E.J., Scott, D.S.: *An Introduction to Modal Logic: the Lemmon Notes*. Blackwell (1977)
21. Lyon, T.: On the correspondence between nested calculi and semantic systems for intuitionistic logics. *Journal of Logic and Computation* **31**(1), 213–265 (12 2020). <https://doi.org/10.1093/logcom/exaa078>
22. Lyon, T.: *Refining Labelled Systems for Modal and Constructive Logics with Applications*. Ph.D. thesis, Technische Universität Wien (2021)
23. Lyon, T., van Berkel, K.: Automating agential reasoning: Proof-calculi and syntactic decidability for stit logics. In: Baldoni, M., Dastani, M., Liao, B., Sakurai, Y., Zalila Wenkstern, R. (eds.) *PRIMA 2019: Principles and Practice of Multi-Agent Systems - 22nd International Conference, Turin, Italy, October 28-31, 2019, Proceedings*. Lecture Notes in Computer Science, vol. 11873, pp. 202–218. Springer International Publishing, Cham (2019)
24. Lyon, T., Tiu, A., Goré, R., Clouston, R.: Syntactic interpolation for tense logics and bi-intuitionistic logic via nested sequents. In: Fernández, M., Muscholl, A. (eds.) *28th EACSL Annual Conference on Computer Science Logic, CSL 2020, January 13-16, 2020, Barcelona, Spain. LIPIcs*, vol. 152, pp. 28:1–28:16. Schloss Dagstuhl - Leibniz-Zentrum für Informatik (2020). <https://doi.org/10.4230/LIPIcs.CSL.2020.28>
25. Marin, S., Straßburger, L.: Label-free modular systems for classical and intuitionistic modal logics. In: *Advances in Modal Logic 10, invited and contributed papers from the tenth conference on "Advances in Modal Logic,"* held in Groningen, The Netherlands, August 5-8, 2014. pp. 387–406 (2014), <http://www.aiml.net/volumes/volume10/Marin-Strassburger.pdf>
26. Pimentel, E.: A semantical view of proof systems. In: Moss, L.S., de Queiroz, R.J.G.B., Martínez, M. (eds.) *Logic, Language, Information, and Computation - 25th International Workshop, WoLLIC 2018, Bogota, Colombia, July 24-27, 2018, Proceedings*. Lecture Notes in Computer Science, vol. 10944, pp. 61–76. Springer (2018). https://doi.org/10.1007/978-3-662-57669-4_3
27. Pitts, A.M.: Evaluation logic. In: *IV Higher Order Workshop, Banff 1990*. pp. 162–189. Springer (1991)
28. Plotkin, G., Stirling, C.: A framework for intuitionistic modal logics: Extended abstract. In: *Proceedings of the 1986 Conference on Theoretical Aspects of Reasoning about Knowledge*. p. 399–406. TARK '86, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (1986)
29. Post, E.L.: Recursive unsolvability of a problem of Thue. *The Journal of Symbolic Logic* **12**(1), 1–11 (1947)
30. Servi, G.F.: Axiomatizations for some intuitionistic modal logics. *Rend. Sem. Mat. Univers. Politecn. Torino* **42**(3), 179–194 (1984)

31. Simpson, A.K.: The proof theory and semantics of intuitionistic modal logic. Ph.D. thesis, University of Edinburgh. College of Science and Engineering. School of Informatics (1994)
32. Straßburger, L.: Cut elimination in nested sequents for intuitionistic modal logics. In: Pfenning, F. (ed.) Foundations of Software Science and Computation Structures. Lecture Notes in Computer Science, vol. 7794, pp. 209–224. Springer Berlin Heidelberg, Berlin, Heidelberg (2013)
33. Tiu, A., Ianovski, E., Goré, R.: Grammar logics in nested sequent calculus: Proof theory and decision procedures. In: Bolander, T., Braüner, T., Ghilardi, S., Moss, L.S. (eds.) Advances in Modal Logic 9, papers from the ninth conference on "Advances in Modal Logic," held in Copenhagen, Denmark, 22-25 August 2012. pp. 516–537. College Publications (2012), <http://www.aiml.net/volumes/volume9/Tiu-Ianovski-Gore.pdf>