



HAL
open science

A Simple Loopcheck for Intuitionistic K

Marianna Girlando, Roman Kuznets, Sonia Marin, Marianela Morales, Lutz
Straßburger

► **To cite this version:**

Marianna Girlando, Roman Kuznets, Sonia Marin, Marianela Morales, Lutz Straßburger. A Simple Loopcheck for Intuitionistic K. 2024. hal-04569308

HAL Id: hal-04569308

<https://inria.hal.science/hal-04569308>

Preprint submitted on 6 May 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution - NonCommercial 4.0 International License

A Simple Loopcheck for Intuitionistic K

Marianna Girlando^{1*}, Roman Kuznets^{2**}, Sonia Marin³,
Mariana Morales⁴, and Lutz Straßburger⁵

¹ University of Amsterdam

² TU Wien

³ University of Birmingham

⁴ IMDEA Software Institute

⁵ INRIA Saclay

Abstract In this paper, we present an algorithm for establishing decidability and finite model property of intuitionistic modal logic IK. These two results have been previously established independently by proof theoretic and model theoretic techniques respectively. Our algorithm, by contrast, enables us to establish both properties at the same time and simplifies previous approaches. It implements root-first proof search in a labelled sequent calculus that employs two binary relations: one corresponding to the modal accessibility relation and the other to the preorder relation of intuitionistic models. As a result, all the rules become invertible, hence semantic completeness could be established directly by extracting a (possibly infinite) countermodel from a failed proof attempt. To obtain the finite model property, we rather introduce a simple loopcheck ensuring that root-first proof search always terminates. The resulting finite countermodel displays a layered structure akin to that of intuitionistic first-order models.

Keywords: Intuitionistic modal logic · Labelled sequent calculus · Decidability · Root-first proof search · Finite model property

1 Introduction

The search for intuitionistic variants of classical modal logics originated from several different research traditions. In particular, computer scientists defined intuitionistic modal systems which could model reasoning on partial information in a concurrent setting [14], or which extended the Curry–Howard correspondence to the modal language [1], leading to varied ways of combining intuitionistic and modal behaviors [6]. For pure logicians, the preferred intuitionistic variant of modal logic is typically what came to be known as *intuitionistic modal logic* IK, originally axiomatized by Fischer Servi as some adequately chosen subset of K-validities [3]. She calls it a *reasonable intuitionistic analog* of classical modal K as it can be understood as a bimodal logic by directly extending the Gödel translation of intuitionistic propositional logic IPL into modal logic S4 to a product

* Funded by Horizon 2021, Marie Skłodowska-Curie grant CYDER (101064105).

** Funded by the FWF ByzDEL project (P33600).

with K . In terms of semantics [2], IK also corresponds to combining the relational models of classical modal logic K , which use branching to represent modal operators, with those for IPL, which use branching to strengthen the implication.

Simpson [13] also observed the *naturalness* of this way of defining intuitionistic modal logic for its ability to translate into intuitionistic first-order logic via the *standard translation*, thus facilitating the design of proof systems in labelled natural deduction and sequent calculus formalisms. In his Ph.D. thesis, Simpson studied modal logic IK in detail and provided two distinct methods to prove its decidability. The first, non-constructive method adapts a standard model-theoretic argument by filtration. The finite model property (FMP) for IK is established by constructing a finite quotient of the canonical model. Together with Fisher Servi’s finite axiomatization, the FMP then yields decidability. The second, constructive method is based on Simpson’s labelled sequent calculus. The validity of a formula is determined by using a proof-search strategy supplied with a termination measure that guarantees the formula is invalid whenever the proof search is terminated with failure. Both methods are quite involved. The first one relies on heavy semantic machinery to provide finite countermodels for invalid formulas without computing proofs for valid ones. Conversely, when the proof search in the second method does not yield a proof, it terminates on a partial derivation which does not provide a method of constructing a countermodel, finite or infinite. Since Simpson’s calculus employs non invertible rules, a countermodel cannot be immediately “read off” from a failed branch.

In this work we introduce a novel algorithm which implements terminating proof search for IK , constructively producing either a proof or a countermodel for the formula at the root. By building *one* proof search tree we check for validity of a formula and, in case of failure, the same object allows to construct a suitable finite countermodel. To this end, we extend Simpson’s labelled sequents, which use a single relational symbol for the modalities, to *fully labelled sequents* that also add a symbol for the intuitionistic preorder relation [9]. This strategy, which represents an alternative to Gentzen’s *one formula on the right* methodology, allows us to impose the intuitionistic discipline to the calculus, similarly to [11] for IPL and [8] for intuitionistic epistemic logics. It has the consequence of rendering all rules invertible, meaning that backtracking in root-first proof search is not needed anymore. Yet, as in Simpson’s proof, one still needs to know when to terminate proof search. In [11], the proof-search algorithm for IPL applies most rules eagerly except for the right implication, for which a loopcheck needs to be employed. If a loop is detected in a proof, proof search stops and a finite countermodel can be generated by reproducing the loop in the model.

By moving from (singly) labelled to fully labelled sequents, we similarly recover invertibility of all rules and a backtracking-free proof search. On the other hand, devising a criterion to terminate proof search is not as straightforward since we also need to account for the modal part of the relational structure. In previous work [4,10], we relied on several intricate loopchecks to ensure termination in the case where both the intuitionistic and modal relations are transitive and require separate loopchecks. By contrast, in the case of IK , the modal search

terminates with no need for loops, making it possible to separate the modal search from the intuitionistic one in a modular way. In this work, we introduce the concept of *shrinking* which turns out to be the missing piece to recover the termination of proof search for IK as a simple extension of the one for IPL in [11].

In summary, via this simple loopcheck, we obtain a proof-search algorithm that, starting from a given formula, either gives a correct proof or a suitable finite countermodel. This provides an alternative simpler constructive proof of decidability and the finite model property of IK.

The paper is organized as follows. In Section 2, we lay out preliminaries for the syntax and semantics of intuitionistic modal logic. In Section 3, we present the proof system labIK^s , a fully labelled sequent calculus based on sets rather than multisets to streamline proof search. In Section 4, we describe the strong connection between fully labelled sequents and birelational models that we then exploit in Section 5 to deduce properties of the proof-search algorithm. We close with conclusions and future directions in Section 6. For the sake of brevity, all the proofs can be found in the Appendix, together with some additional examples.

2 Preliminaries

The formulas of our language, denoted by A, B, C, \dots are constructed from a countable set \mathcal{A} of atomic propositions, denoted by a, b, c, \dots through the grammar $A ::= \perp \mid a \mid (A \wedge A) \mid (A \vee A) \mid (A \supset A) \mid \Box A \mid \Diamond A$. An axiomatization of intuitionistic modal logic IK is obtained by adding to an axiomatization of intuitionistic propositional logic IPL the following axioms:

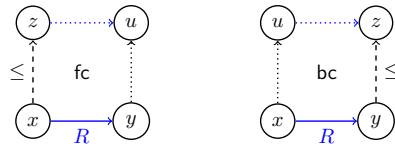
$$\begin{array}{lll} k_1: \Box(A \supset B) \supset (\Box A \supset \Box B) & k_3: \Diamond(A \vee B) \supset (\Diamond A \vee \Diamond B) & k_5: \Diamond \perp \supset \perp \\ k_2: \Box(A \supset B) \supset (\Diamond A \supset \Diamond B) & k_4: (\Diamond A \supset \Box B) \supset \Box(A \supset B) & \end{array}$$

A formula is a theorem of IK iff it is derivable from the axioms via the rules of *modus ponens* (infer B from A and $A \supset B$) and *necessitation* (infer $\Box A$ from A).

Let us now recall the *birelational models* [2,12] for intuitionistic modal logics, which combine the Kripke semantics for IPL and for classical modal logics.

Definition 2.1. A *birelational frame* \mathcal{F} is a triple $\langle W, R, \leq \rangle$ of a nonempty set W of *worlds* equipped with an *accessibility relation* R and a preorder \leq (i.e., a reflexive and transitive relation) satisfying the conditions of *forward confluence* fc and *backward confluence* bc:

- fc For all $x, y, z \in W$, if $x \leq z$ and xRy , there exists $u \in W$ with zRu and $y \leq u$.
- bc For all $x, y, z \in W$, if xRy and $y \leq z$, there exists $u \in W$ with $x \leq u$ and uRz .



Definition 2.2. A *birelational model* \mathcal{M} is a quadruple $\langle W, R, \leq, V \rangle$ where $\langle W, R, \leq \rangle$ is a birelational frame and $V: W \rightarrow 2^A$ a *valuation function*, i.e., a function mapping each world w to the subset of propositional atoms that are true at w , additionally subject to *monotonicity*: if $w \leq w'$, then $V(w) \subseteq V(w')$. We write $\mathcal{M}, w \Vdash a$ iff $a \in V(w)$ and we recursively extend relation \Vdash to compound formulas as follows:

$$\begin{aligned} \mathcal{M}, w &\not\Vdash \perp \\ \mathcal{M}, w \Vdash A \wedge B &\text{ iff } \mathcal{M}, w \Vdash A \text{ and } \mathcal{M}, w \Vdash B; \\ \mathcal{M}, w \Vdash A \vee B &\text{ iff } \mathcal{M}, w \Vdash A \text{ or } \mathcal{M}, w \Vdash B; \\ \mathcal{M}, w \Vdash A \supset B &\text{ iff for all } w' \text{ with } w \leq w', \text{ if } \mathcal{M}, w' \Vdash A, \text{ then } \mathcal{M}, w' \Vdash B; \\ \mathcal{M}, w \Vdash \Box A &\text{ iff for all } w' \text{ and } u \text{ with } w \leq w' \text{ and } w' R u, \text{ we have } \mathcal{M}, u \Vdash A; \\ \mathcal{M}, w \Vdash \Diamond A &\text{ iff there exists } u \text{ such that } w R u \text{ and } \mathcal{M}, u \Vdash A. \end{aligned}$$

Monotonicity of the valuation V extends to the relation \Vdash by induction on the complexity of formulas.

Proposition 2.3 (Monotonicity). *For any formula A and for any $w, w' \in W$, if $w \leq w'$ and $\mathcal{M}, w \Vdash A$, then $\mathcal{M}, w' \Vdash A$.*

Definition 2.4 (Validity). Formula A is *valid in a model* $\mathcal{M} = \langle W, R, \leq, V \rangle$ iff $\mathcal{M}, w \Vdash A$ for all $w \in W$. Formula A is *valid in a frame* $\mathcal{F} = \langle W, R, \leq \rangle$ iff it is valid in $\langle W, R, \leq, V \rangle$ for all valuations V .

The correspondence between syntax and semantics for IK is as follows:

Theorem 2.5 ([3,12]). *A formula A is a theorem of IK if and only if A is valid in every birelational frame $\langle W, R, \leq \rangle$.*

From now on, we shall say that a formula is *IK-valid*, or just *valid*, if it is valid in every birelational frame $\langle W, R, \leq \rangle$.

3 Labelled Sequent Calculus labIK^s

We now present the fully labelled sequent sequent calculus labIK^s for IK that is used in our decision algorithm. It is an equivalent formulation (together with some notational variations) of the sequent calculus for IK presented in [9].

To define a labelled proof system, we first enrich the language of IK by a countable set of *labels*, denoted by x, y, z , etc. Then, we define *relational atoms* as expression xRy or $x \leq y$ where x and y are labels, and *labelled formulas* as pairs $x:A$ of a label x and a formula A . The labelled calculus introduced by Simpson in [13] employs only one kind of relational atoms, xRy . We here follow the fully labelled approach from [9], which instead employs relational atoms in correspondence to both relations of birelational frames.

In the literature, a *labelled sequent* is usually defined as a triple $\mathcal{R}, \Gamma \Longrightarrow \Delta$ where \mathcal{R} is a set of relational atoms and Γ and Δ are multisets (or sets) of labelled formulas respectively, all written as comma-separated lists. To simplify subsequent definitions and proofs, we employ a different notation for our sequents.

$\frac{\text{id} \frac{}{\mathcal{G}, x:a^\bullet, x:a^\circ}}{\wedge^\bullet \frac{\mathcal{G}, x:A \wedge B^\bullet, x:A^\bullet, x:B^\bullet}{\mathcal{G}, x:A \wedge B^\bullet}}$	$\frac{\perp^\bullet \frac{}{\mathcal{G}, x:\perp^\bullet}}{\wedge^\circ \frac{\mathcal{G}, x:A \wedge B^\circ, x:A^\circ \quad \mathcal{G}, x:A \wedge B^\circ, x:B^\circ}{\mathcal{G}, x:A \wedge B^\circ}}$
$\vee^\bullet \frac{\mathcal{G}, x:A \vee B^\bullet, x:A^\bullet \quad \mathcal{G}, x:A \vee B^\bullet, x:B^\bullet}{\mathcal{G}, x:A \vee B^\bullet}$	$\vee^\circ \frac{\mathcal{G}, x:A \vee B^\circ, x:A^\circ, x:B^\circ}{\mathcal{G}, x:A \vee B^\circ}$
$\supset^\bullet \frac{\mathcal{G}, x:A \supset B^\bullet, x:A^\circ \quad \mathcal{G}, x:A \supset B^\bullet, x:B^\bullet}{\mathcal{G}, x:A \supset B^\bullet}$	$\supset^\circ \frac{\mathcal{G}, x \leq z, z:A^\bullet, x:A \supset B^\circ, z:B^\circ}{\mathcal{G}, x:A \supset B^\circ} \quad z \text{ fresh}$
$\square^\bullet \frac{\mathcal{G}, xRy, x:\Box A^\bullet, y:A^\bullet}{\mathcal{G}, xRy, x:\Box A^\bullet}$	$\square^\circ \frac{\mathcal{G}, x \leq u, uRz, x:\Box A^\circ, z:A^\circ}{\mathcal{G}, x:\Box A^\circ} \quad u, z \text{ fresh}$
$\diamond^\bullet \frac{\mathcal{G}, xRy, x:\Diamond A^\bullet, y:A^\bullet}{\mathcal{G}, x:\Diamond A^\bullet} \quad y \text{ fresh}$	$\diamond^\circ \frac{\mathcal{G}, xRy, x:\Diamond A^\circ, y:A^\circ}{\mathcal{G}, xRy, x:\Diamond A^\circ}$
<hr style="border-top: 1px dotted black;"/>	
$\text{mon}^\bullet \frac{\mathcal{G}, x \leq y, x:A^\bullet, y:A^\bullet}{\mathcal{G}, x \leq y, x:A^\bullet}$	$\text{weak} \frac{\mathcal{G}}{\mathcal{G}, \mathcal{G}'}$
$\text{fc} \frac{\mathcal{G}, xRy, x \leq z, y \leq u, zRu}{\mathcal{G}, xRy, x \leq z} \quad u \text{ fresh}$	$\leq_{\text{rf}} \frac{\mathcal{G}, x \leq x}{\mathcal{G}} \quad \leq_{\text{tr}} \frac{\mathcal{G}, x \leq y, y \leq z, x \leq z}{\mathcal{G}, x \leq y, y \leq z}$
$\text{bc} \frac{\mathcal{G}, xRy, y \leq z, x \leq u, uRz}{\mathcal{G}, xRy, y \leq z} \quad u \text{ fresh}$	

 Figure 1. System labK^s

First, we assign a polarity to each labelled formula: \bullet (input) or \circ (output). Then, we define a *polarized labelled sequent* \mathcal{G} as a *set* of relational atoms and labelled formulas with polarities, with $\mathcal{G}, \mathcal{G}'$ meaning $\mathcal{G} \cup \mathcal{G}'$ and \mathcal{G}, F standing for $\mathcal{G} \cup \{F\}$, for F being a relational atom or a labelled formula. In the following, we simply write *sequent* to mean *polarized labelled sequent*.

Example 3.1. $\mathcal{G} = x \leq y, yRz, yRu, x:a \wedge b^\bullet, y:c^\circ, z:\Box a^\circ$ is such a sequent.

Intuitively, the polarities play the role of the sequent arrow \implies , allowing us to identify the left- and right-hand side of a sequent. The input formulas, or \bullet -formulas, are those that would occur in the antecedent of a labelled sequent, and the output formulas, or \circ -formulas, are those occurring in the succedent.⁶ Thus, every sequent can be translated into a labelled sequent, and vice versa.

Example 3.2. Sequent \mathcal{G} from Example 3.1 corresponds to the labelled sequent $x \leq y, yRz, yRu, x:a \wedge b \implies y:c, z:\Box a$.

The rules of labK^s are displayed in Figure 1.⁷ The rules in the upper part of the Figure are *logical rules*, and those in the lower part of the figure are

⁶ We do not assign polarities to relational atoms because, as in the case of standard labelled sequents, they only occur in the antecedent.

⁷ The superscript *s* stands for *sets*, as our calculus works on sets, unlike the original labK [9] that works on multisets.

structural rules. Observe that, since sequents are defined as sets, contraction is embedded into the system. We have chosen a cumulative version of the rules, with the principal formula repeated in the premise(s), as this will become useful in the definition of the proof-search algorithm. Finally, thanks to the presence of mon^\bullet , an explicit structural rule for monotonicity, rules id , \supset^\bullet , and \square^\bullet are formulated as in the classical case.

Definition 3.3. A *derivation tree* (or *derivation* for short) over a set \mathcal{S} of inference rules is a tree whose nodes are labeled with sequents and such that a node labeled with \mathcal{G} is the parent of nodes labeled with $\mathcal{G}_1, \dots, \mathcal{G}_k$, where $k \geq 0$, iff $r \frac{\mathcal{G}_1 \ \dots \ \mathcal{G}_k}{\mathcal{G}}$ is an instance of inference rule $r \in \mathcal{S}$. A **proof** is a derivation where each leaf is the conclusion of id or \perp^\bullet .

An example of a proof in labIK^s can be found in the Appendix. A rule r is **admissible** iff, whenever there are proofs of its premise(s), there is a proof of its conclusion. Rule r is **derivable** whenever there is a derivation of its conclusion whose leaves consist of its premise(s). Rule r is **invertible** iff, whenever there is a proof of its conclusion, then there are proofs of all its premises. All the rules of labIK^s are invertible, and the proof closely follows the proof in [9].

To prove soundness and completeness of labIK^s , we show how to translate every proof in labIK^s into a proof in labIK and back. System labIK is the fully labelled sequent calculus for IK introduced in [9], which employs labelled sequents $\mathcal{R}, \Gamma \Longrightarrow \Delta$ as discussed above in their multiset formulation. The rules of labIK are the same as the rules of labIK^s , except that they are not cumulative, rules weak and mon^\bullet are admissible, and rules id , \supset^\bullet , and \square^\bullet are formulated as follows (employing our notation):

$$\begin{array}{c} \text{id} \frac{}{\mathcal{G}, x \leq y, x : a^\bullet, y : a^\circ} \\ \supset^\bullet \frac{\mathcal{G}, x \leq y, x : A \supset B^\bullet, y : A^\circ \quad \mathcal{G}, y : B^\bullet}{\mathcal{G}, x \leq y, x : A \supset B^\bullet} \quad \square^\bullet \frac{\mathcal{G}, x \leq y, y R z, x : \square A^\bullet, z : A^\bullet}{\mathcal{G}, x \leq y, y R z, x : \square A^\bullet} \end{array}$$

Using admissibility of the structural rules (including contraction) in labIK and through trivial translations of labIK^s sequents into labIK labelled sequents and vice versa, it is easy to show that the two proof systems are equivalent. From this, we immediately obtain soundness and completeness of labIK^s .

Theorem 3.4. *A formula A is a theorem of IK iff for every x , the sequent $x : A^\circ$ has a proof in labIK^s .*

4 Models from Sequents

A fully labelled sequent contains sufficient information to extract a birelational model. This will be useful when proving completeness, as we will be able to immediately construct a (counter)model from a leaf of a failed proof-search tree.

Notation 4.1. Let \mathcal{G} be a sequent. We write $\ell(\mathcal{G})$ for the set of labels occurring in \mathcal{G} . On this set we define two binary relations $R_{\mathcal{G}}$ and $\leq_{\mathcal{G}}$ as follows: $xR_{\mathcal{G}}y$ iff $xRy \in \mathcal{G}$ and $x \leq_{\mathcal{G}}y$ iff $x \leq y \in \mathcal{G}$.

Definition 4.2 (Model of a sequent). Let \mathcal{G} be a sequent. We define the *model* $\mathcal{M}_{\mathcal{G}}$ of \mathcal{G} to be the quadruple $\mathcal{M}_{\mathcal{G}} = \langle \ell(\mathcal{G}), R_{\mathcal{G}}, \leq_{\mathcal{G}}, V \rangle$ where $V: \ell(\mathcal{G}) \rightarrow 2^{\mathcal{A}}$ is such that for all atoms $a \in \mathcal{A}$ we have $a \in V(w)$ iff $w:a^{\bullet} \in \mathcal{G}$.

This model will be a proper birelational model, provided that the sequent satisfies a number of properties. Intuitively, we want all the rules of labLK^s to have been exhaustively but non-redundantly applied to the sequent. We shall express these requirements with the notion of a *happy sequent* below.

Definition 4.3 (Happy labelled formula). Let \mathcal{G} be a sequent. We say that a formula $x:A^{\bullet} \in \mathcal{G}$ (resp. $x:A^{\circ} \in \mathcal{G}$) is *happy* iff the following conditions hold:

- $x:a^{\bullet} \in \mathcal{G}$ is always happy;
- $x:a^{\circ} \in \mathcal{G}$ is happy iff $x:a^{\bullet} \notin \mathcal{G}$;
- $x:\perp^{\bullet} \in \mathcal{G}$ is never happy;
- $x:\perp^{\circ} \in \mathcal{G}$ is always happy;
- $x:A \wedge B^{\bullet} \in \mathcal{G}$ is happy iff $x:A^{\bullet} \in \mathcal{G}$ and $x:B^{\bullet} \in \mathcal{G}$;
- $x:A \wedge B^{\circ} \in \mathcal{G}$ is happy iff $x:A^{\circ} \in \mathcal{G}$ or $x:B^{\circ} \in \mathcal{G}$;
- $x:A \vee B^{\bullet} \in \mathcal{G}$ is happy iff $x:A^{\bullet} \in \mathcal{G}$ or $x:B^{\bullet} \in \mathcal{G}$;
- $x:A \vee B^{\circ} \in \mathcal{G}$ is happy iff $x:A^{\circ} \in \mathcal{G}$ and $x:B^{\circ} \in \mathcal{G}$;
- $x:A \supset B^{\bullet} \in \mathcal{G}$ is happy iff $x:A^{\circ} \in \mathcal{G}$ or $x:B^{\bullet} \in \mathcal{G}$;
- $x:A \supset B^{\circ} \in \mathcal{G}$ is happy iff $y:A^{\bullet} \in \mathcal{G}$ and $y:B^{\circ} \in \mathcal{G}$ for some y with $x \leq_{\mathcal{G}}y$;
- $x:\Box A^{\bullet} \in \mathcal{G}$ is happy iff $z:A^{\bullet} \in \mathcal{G}$ for all z with $xR_{\mathcal{G}}z$;
- $x:\Box A^{\circ} \in \mathcal{G}$ is happy iff $z:A^{\circ} \in \mathcal{G}$ for some y, z s.t. $x \leq_{\mathcal{G}}y$ and $yR_{\mathcal{G}}z$;
- $x:\Diamond A^{\bullet} \in \mathcal{G}$ is happy iff $y:A^{\bullet} \in \mathcal{G}$ for some y with $xR_{\mathcal{G}}y$;
- $x:\Diamond A^{\circ} \in \mathcal{G}$ is happy iff $y:A^{\circ} \in \mathcal{G}$ for all y s.t. $xR_{\mathcal{G}}y$.

Otherwise, the formula is *unhappy*.

Definition 4.4 (Happy label). A label x occurring in a sequent \mathcal{G} is *happy* iff all formulas occurring at x in \mathcal{G} are happy.

Definition 4.5 (Structurally happy sequent). A sequent \mathcal{G} is *structurally happy* iff the following holds:

- (mon $^{\bullet}$) if $x \leq_{\mathcal{G}}y$ and $x:C^{\bullet} \in \mathcal{G}$, then $y:C^{\bullet} \in \mathcal{G}$;
- (fc) if $xR_{\mathcal{G}}y$ and $x \leq_{\mathcal{G}}z$, then there is u such that $y \leq_{\mathcal{G}}u$ and $zR_{\mathcal{G}}u$;
- (bc) if $xR_{\mathcal{G}}y$ and $y \leq_{\mathcal{G}}z$, then there is u such that $x \leq_{\mathcal{G}}u$ and $uR_{\mathcal{G}}z$;
- (\leq tr) if $x \leq_{\mathcal{G}}y$ and $y \leq_{\mathcal{G}}z$, then $x \leq_{\mathcal{G}}z$;
- (\leq rf) $x \leq_{\mathcal{G}}x$ for all x occurring in \mathcal{G} ;

Definition 4.6 (Happy sequent). A sequent \mathcal{G} is *happy* iff it is structurally happy and all labels in the sequent are happy.

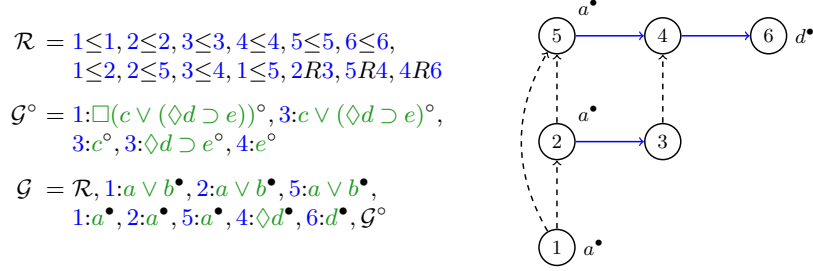


Figure 2. Model $\mathcal{M}_{\mathcal{G}}$ of sequent \mathcal{G} . The nodes represent the worlds of the model, the dashed arrows represent the \leq -relation, and the solid arrow represent the R -relation. For simplicity, we do not represent the reflexive \leq -edges of $\mathcal{M}_{\mathcal{G}}$.

We can now show that the model of a happy sequent is a birelational model. Moreover, all input formulas are satisfied in the model, and all output formulas are falsified in the model.

Theorem 4.7. *For a happy sequent \mathcal{G} , its model $\mathcal{M}_{\mathcal{G}} = \langle \ell(\mathcal{G}), R_{\mathcal{G}}, \leq_{\mathcal{G}}, V \rangle$ is a birelational model satisfying the following: i) if $x:A^\bullet \in \mathcal{G}$, then $\mathcal{M}_{\mathcal{G}}, x \Vdash A$; and ii) if $x:A^\circ \in \mathcal{G}$, then $\mathcal{M}_{\mathcal{G}}, x \not\Vdash A$.*

Example 4.8. Take the model $\mathcal{M}_{\mathcal{G}}$ of \mathcal{G} represented in Figure 2. It holds that, e.g., $\mathcal{M}_{\mathcal{G}}, 4 \Vdash \Diamond d$ and $\mathcal{M}_{\mathcal{G}}, 1 \not\Vdash \Box(c \vee (\Diamond d \supset e))$.

In what follows, we will often make use of the notion of axiomatic sequent.

Definition 4.9 (Axiomatic sequent). A sequent \mathcal{G} is *axiomatic* iff there is a label x such that either $x:a^\bullet \in \mathcal{G}$ and $x:a^\circ \in \mathcal{G}$ for some a , or $x:\perp^\bullet \in \mathcal{G}$. Otherwise, \mathcal{G} is called *non-axiomatic*.

Remark 4.10. An axiomatic sequent \mathcal{G} is never happy, because either $x:a^\circ \in \mathcal{G}$ or $x:\perp^\bullet \in \mathcal{G}$ is unhappy.

5 The Proof-search Algorithm

We shall here define an algorithmic procedure that performs proof search in lablK^5 , with the aim of obtaining either a set of axiomatic sequents, from which a lablK^5 derivation can be easily constructed, or a happy sequent, from which a (counter)model can be extracted (also a sequent that can be easily transformed into a happy sequent will suffice for this purpose). Our algorithm relies on the fact that the sequents created by the algorithm are all *layered* (provided the endsequent contains only one label), where each layer can be thought of as a classical K-model.

Definition 5.1 (Layer). For a sequent \mathcal{G} , we define the relation $R_{\mathcal{G}}^{\leftrightarrow}$ to be the transitive and reflexive closure of $R_{\mathcal{G}} \cup R_{\mathcal{G}}^{-1}$. Since this is an equivalence relation, we can define a *layer* L in \mathcal{G} to be an equivalence class of $R_{\mathcal{G}}^{\leftrightarrow}$. A layer L is *happy* if all labels $x \in L$ are happy.

Definition 5.2 (Layered sequent). We say that a sequent \mathcal{G} is *layered* iff for any labels x, x', y , and y' occurring in \mathcal{G} :

1. if $xR_{\mathcal{G}}^{\leftrightarrow}y$ for $x \neq y$, then $x \not\leq_{\mathcal{G}}y$ and $y \not\leq_{\mathcal{G}}x$; and
2. if $xR_{\mathcal{G}}^{\leftrightarrow}y, x'R_{\mathcal{G}}^{\leftrightarrow}y'$, and $x \leq_{\mathcal{G}}x'$ for $x \neq x'$, then $y' \not\leq_{\mathcal{G}}y$.

For layers L_1 and L_2 , we define $L_1 \leq L_2$ whenever there are labels $x \in L_1$ and $y \in L_2$ such that $x \leq_{\mathcal{G}}y$. We write $L_1 < L_2$ iff $L_1 \leq L_2$ and $L_1 \neq L_2$.

Proposition 5.3. *For a layered structurally saturated sequent \mathcal{G} , relation \leq is an order relation on its layers.*

The main observation to make is that all logical rules of lablK^s , except for \supset° and \square° , only work locally on a layer, whereas \supset° and \square° create a new layer. As in classical K, where proof search terminates without any particular loopcheck, we can exhaustively apply the rules $\wedge^{\bullet}, \wedge^{\circ}, \vee^{\bullet}, \vee^{\circ}, \supset^{\bullet}, \square^{\bullet}, \diamond^{\bullet}$, and \diamond° . This process is called *saturation* and motivates the following definition:

Definition 5.4 (Almost happy). A label x in a sequent \mathcal{G} is *almost happy* iff all formulas occurring at x are happy except, possibly, those of the shapes $\perp^{\bullet}, a^{\circ}, A \supset B^{\circ}$, and $\square A^{\circ}$. A layer L (resp. sequent \mathcal{G}) is *almost happy* iff all labels in L (resp. \mathcal{G}) are almost happy.

Definition 5.5 (Saturation). A *saturation tree* is a derivation tree \mathcal{T} over the rule set $\{\wedge^{\bullet}, \wedge^{\circ}, \vee^{\bullet}, \vee^{\circ}, \supset^{\bullet}, \square^{\bullet}, \diamond^{\bullet}, \diamond^{\circ}\}$, whose leaves are almost happy. If \mathcal{G} is the root of \mathcal{T} , then the set of its leaves form a *saturation* of \mathcal{G} .

With this, we can define a *macro rule*

$$\text{sat} \frac{\mathcal{G}_1 \quad \dots \quad \mathcal{G}_n}{\mathcal{G}} \text{ where } \{\mathcal{G}_1, \dots, \mathcal{G}_n\} \text{ is a saturation of } \mathcal{G} \quad (1)$$

and immediately obtain the following:

Proposition 5.6. *The sat-rule is derivable in lablK^s .*

Remark 5.7. The saturation of a sequent is in general not uniquely defined since sequents can differ up to a renaming of labels. However, we can fix a strategy and naming scheme for fresh labels when applying the inference rules $\wedge^{\bullet}, \wedge^{\circ}, \vee^{\bullet}, \vee^{\circ}, \supset^{\bullet}, \square^{\bullet}, \diamond^{\bullet}, \diamond^{\circ}$, so that without loss of generality, we can for the context of this paper speak of *the* saturation of a sequent \mathcal{G} , and denote it by $[\mathcal{G}]$.

The rules \supset° and \square° create a new layer, and our algorithm immediately completes the layer, making the sequent structurally happy, by applying the structural rules of lablK^s . To make this formal, we introduce the notion of *lifting*.

Construction 5.8 (Lifting). Let \mathcal{G} be an almost happy sequent, and let L be a layer in \mathcal{G} . Then $L = \{y_1, \dots, y_l\}$ for some $l \geq 1$. Let us assume $x \in L$. Now let \hat{L} be a set of fresh labels $\{\hat{y}_1, \dots, \hat{y}_l\}$. We define $\mathcal{G}^{\uparrow x}$ to be the sequent containing for every $i = 1 \dots l$:

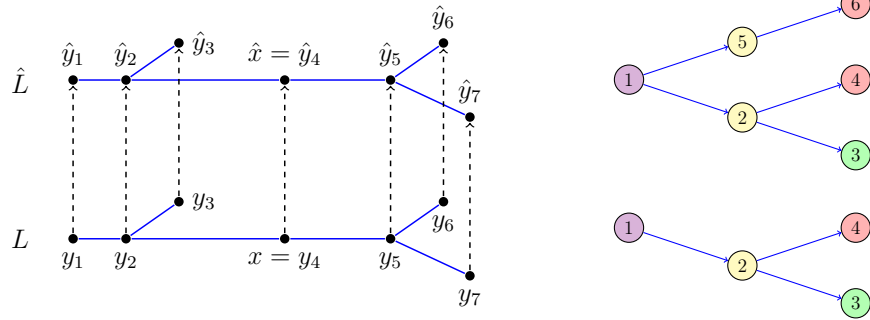


Figure 3. *Left:* Lifting of the layer L . *Right:* An example of shrinking, for $2 \sim 5, 4 \sim 6$.

1. relational atom $\hat{y}_i \leq \hat{y}_i$;
2. for every label w occurring in \mathcal{G} : the relational atom $w \leq \hat{y}_i$ whenever $w \leq_{\mathcal{G}} y_i$;
3. for every $i' = 1 \dots l$: the relational atom $\hat{y}_i R \hat{y}_{i'}$ whenever $y_i R_{\mathcal{G}} y_{i'}$,
4. for every formula C : the labelled formula $\hat{y}_i : C^\bullet$ whenever $y_i : C^\bullet \in \mathcal{G}$.

In other words, $\mathcal{G} \uparrow^x$ contains a copy of L , without the \circ -formulas, that would be put on top of L if $\mathcal{G} \uparrow^x$ was added to \mathcal{G} , so that $\mathcal{G} \cup \mathcal{G} \uparrow^x$ would be structurally happy. Now assume there is some unhappy formula $x:F^\circ \in \mathcal{G}$, for $F = A \supset B$ or $F = \Box B$, and $x \in L$. Then $x = y_j$ for some $j = 1 \dots l$. We let $\hat{x} = \hat{y}_j$, and we define the **lifting** $\mathcal{G} \uparrow^{x:F^\circ}$ as follows:

- If $F = A \supset B$, then $\mathcal{G} \uparrow^{x:A \supset B^\circ} = \mathcal{G} \uparrow^x \cup \{\hat{x}:A^\bullet, \hat{x}:B^\circ\}$.
- If $F = \Box B$, then $\mathcal{G} \uparrow^{x:\Box B^\circ} = \mathcal{G} \uparrow^x \cup \{\hat{x}Rz, z \leq z, z:B^\circ\}$ for some fresh label z .

We can now define a second macro rule, denoted **lift**:

$$\text{lift} \frac{\mathcal{G} \cup \mathcal{G} \uparrow^{x:F^\circ}}{\mathcal{G}} \text{ where } F \in \{A \supset B, \Box B\} \text{ and } x:F^\circ \text{ is unhappy in } \mathcal{G}. \quad (2)$$

The idea behind lifting a layer is depicted on the left in Figure 3. Any application of the lift rule to a formula $x:F^\circ \in \mathcal{G}$ can be simulated by the rules of **lablK^s**: it suffices to apply to \mathcal{G} one instance of rule \supset° or \Box° (depending on the shape of F), followed by possibly multiple applications of rules **fc**, **bc**, $\leq\text{rf}$, $\leq\text{tr}$ and **mon[•]**. Thus, the following holds:

Proposition 5.9. *The lift-rule is derivable in lablK^s.*

One might assume that it is enough to simply repeat saturation and lifting until we see a repetition of a layer. This is indeed the basic idea of what the algorithm is doing. And when \supset° is the only layer creating rule, this is enough, as it would correspond to the standard loopcheck for proof search in IPL [11]. However, the \Box° rule makes layers grow, and a possible repetition is not visible at the moment of the layer creation, but possibly only several steps later. A simple example is shown in Figure 5. For this reason we need another operation that we call **shrinking**.

Definition 5.10 (Equivalent labels and simulation). Let \mathcal{G} be a sequent and let $x, y \in \ell(\mathcal{G})$. We say that x and y are *equivalent*, denoted as $x \sim y$, iff for all formulas A , we have $x:A^\bullet \in \mathcal{G}$ iff $y:A^\bullet \in \mathcal{G}$ and also $x:A^\circ \in \mathcal{G}$ iff $y:A^\circ \in \mathcal{G}$. A *simulation* xSy of y by x is a relation $S \subseteq \ell(\mathcal{G}) \times \ell(\mathcal{G})$ such that whenever xSy , we have $x \sim y$ and, for any y' such that $yR_{\mathcal{G}}y'$, there is x' such that $xR_{\mathcal{G}}x'$ and $x'Sy'$.

In the following, we write $\mathcal{G}[y/x]$ for the sequent obtained by uniformly substituting x for y in \mathcal{G} .

Definition 5.11 (Shrinking). Let \mathcal{G} be a sequent. Two labels $x, y \in \ell(\mathcal{G})$ are *siblings*, denoted as $x|y$, if there is a label $z \in \ell(\mathcal{G})$ with $zR_{\mathcal{G}}x$ and $zR_{\mathcal{G}}y$. A label $y \in \ell(\mathcal{G})$ is *redundant* if there is a sibling x of y with xSy for some simulation S . A sequent \mathcal{G} is *shrunk* iff there are no redundant labels in $\ell(\mathcal{G})$. A *shrinking tree* is a derivation tree \mathcal{T} over the inference rule

$$\text{shr} \frac{\mathcal{G}[y/x]}{\mathcal{G}} \text{ where } x|y \text{ and } xSy \text{ for some simulation } S \quad (3)$$

whose leaf⁸ is a shrunk sequent. If \mathcal{G} is the conclusion of \mathcal{T} , then the leaf of \mathcal{T} is called a *shrinking* of \mathcal{G} .

Example 5.12. To illustrate the above definition, on the top right of Figure 3 is displayed a sequent with only one layer. Colours indicate equivalence of labels, so we assume that $2 \sim 5$ and $4 \sim 6$. Moreover, it holds that $2|5$. Thus, the sequent can be shrunk by uniformly substituting first label 2 for label 5 and then 4 for 6 . The result is displayed in the lower part of the figure.

Remark 5.13. As with saturation, also the shrinking of a sequent \mathcal{G} is not uniquely defined because of possible label renamings. But as before, we can impose a strategy on the order of the applications of the shrk-rule when computing the shrinking, so that we can speak of *the* shrinking of \mathcal{G} and denote it by $\llbracket \mathcal{G} \rrbracket$.

Combining Remarks 5.7 and 5.13, we can define the *shrink-saturation* of a sequent \mathcal{G} as follows

$$\llbracket \mathcal{G} \rrbracket = \{ \llbracket \mathcal{H} \rrbracket \mid \mathcal{H} \in [\mathcal{G}] \} \quad (4)$$

i.e., for computing it, we first compute the saturation of \mathcal{G} , and then compute the shrinking of each sequent in the saturation.

We are now ready to formally define our loopcheck, which is essentially checking whether we see a repetition of a layer. In that case we stop applying \supset° and \square° and do not create new layers.

Definition 5.14 (Equivalent layers). Let \mathcal{G} be a sequent and let L_1 and L_2 be layers in \mathcal{G} . We say that L_1 and L_2 are *equivalent*, denoted as $L_1 \sim L_2$, if there is a bijection $f: L_1 \rightarrow L_2$ such that for all $x, y \in L_1$ we have $x \sim f(x)$ and $xR_{\mathcal{G}}y$ iff $f(x)R_{\mathcal{G}}f(y)$.

⁸ There is exactly one leaf since the shrk-rule has one premise.

0. Given a formula F , let $\mathcal{G}_0 = r \leq r, r:F^\circ$ and let $\mathfrak{S}_0 = \llbracket \mathcal{G}_0 \rrbracket$.
1. If all sequents in \mathfrak{S}_i are axiomatic, then terminate.
 - The formula F is provable and we can give a proof of $r:F^\circ$ in \mathbf{labIK}^s .
2. Otherwise, pick a non-axiomatic sequent $\mathcal{G}_i \in \mathfrak{S}_i$.
 - (a) If there is an allowed formula $x:F^\circ \in \mathcal{G}_i$, then compute the lifting $\mathcal{G}_i \uparrow^{x:F^\circ}$, let $\mathfrak{S}_{i+1} = (\mathfrak{S}_i \setminus \{\mathcal{G}_i\}) \cup \llbracket \mathcal{G}_i + \mathcal{G}_i \uparrow^{x:F^\circ} \rrbracket$, and go to Step 1.
 - (b) Otherwise, if \mathcal{G}_i does not contain any allowed formulas, then terminate.
 - The formula F is not provable, and \mathcal{G}_i defines a countermodel.

Figure 4. Proof-search algorithm

Definition 5.15 (Allowed formula). Let \mathcal{G} be a saturated sequent with an unhappy formula $x:A \supset B^\circ$ (or $x:\Box A^\circ$), and let L be the layer of x . We say that the formula $x:A \supset B^\circ$ (resp. $x:\Box A^\circ$) is *allowed* iff: there is no layer $L' < L$ with $L' \sim L$, and all layers $L' < L$ are happy.

We can now put all the pieces together in our proof search algorithm, which is given in Figure 4, and which has the following properties:

Theorem 5.16. *The algorithm given in Figure 4 always terminates.*

Theorem 5.17. *If the algorithm shown in Figure 4 terminates in Step 1, then the formula F is a theorem of \mathbf{IK} .*

Theorem 5.18. *If the algorithm shown in Figure 4 terminates in Step 2b, then there is a finite model in which the formula F is not valid.*

Corollary 5.19. *The logic \mathbf{IK} is decidable and has the FMP.*

6 Conclusions and Future Work

We have presented a proof-search algorithm to decide validity of \mathbf{IK} formulas, essentially implementing proof search within \mathbf{labIK}^s , a fully labelled calculus for \mathbf{IK} . When compared to Simpson's model-theoretic and proof-theoretic techniques to establish decidability of \mathbf{IK} , our algorithm has the advantage of producing either a proof or a countermodel for a formula through the same construction. Moreover, since a finite countermodel is extracted from a finite proof search branch, we are able to establish in one go both decidability and the FMP for \mathbf{IK} . Thanks to invertibility of all the rules of \mathbf{labIK}^s , backtracking is not necessary and the countermodel construction is immediate, in contrast to [7].

In future work, we plan to study the complexity of our decision procedure. Specifically, we are interested in bounding the size of the countermodels generated by the algorithm, which at the moment we estimate as non-elementary. We conjecture that smaller countermodels could be found by refining the steps of our algorithm. Moreover, we plan to further extend our algorithm to other logics in the intuitionistic modal family. Finally, we intend to create a theorem prover implementing our algorithm, by analogy with [5].

References

1. Bellin, G., de Paiva, V., Ritter, E.: Extended Curry–Howard correspondence for a basic constructive modal logic. In: Areces, C., de Rijke, M. (eds.) *Workshop Proceedings of Methods for Modalities 2* (2001)
2. Fischer Servi, G.: Semantics for a class of intuitionistic modal calculi. In: Dalla Chiara, M.L. (ed.) *Italian Studies in the Philosophy of Science, Boston Studies in the Philosophy of Science*, vol. 47, pp. 59–72. D. Reidel Publishing Company (1980). https://doi.org/10.1007/978-94-009-8937-5_5
3. Fischer Servi, G.: Axiomatizations for some intuitionistic modal logics. *Rendiconti del Seminario Matematico, Università e Politecnico di Torino* **42**(3), 179–194 (1984)
4. Girlando, M., Kuznets, R., Marin, S., Morales, M., Straßburger, L.: Intuitionistic S4 is decidable. In: *2023 38th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)*, 26–29 June 2023, Boston, USA. IEEE (2023). <https://doi.org/10.1109/LICS56636.2023.10175684>
5. 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: 10th International Joint Conference, IJCAR 2020, Paris, France, July 1–4, 2020, Proceedings, Part II, Lecture Notes in Artificial Intelligence*, vol. 12167, pp. 398–407. Springer (2020). https://doi.org/10.1007/978-3-030-51054-1_25
6. Kavvos, G.A.: The many worlds of modal λ -calculi: I. Curry–Howard for necessity, possibility and time. Eprint 1605.08106, arXiv (2016). <https://doi.org/10.48550/arXiv.1605.08106>
7. Kuznets, R., Straßburger, L.: Maehara-style modal nested calculi. *Archive for Mathematical Logic* **58**(3–4), 359–385 (May 2019). <https://doi.org/10.1007/s00153-018-0636-1>
8. Maffezoli, P., Naibo, A., Negri, S.: The Church–Fitch knowability paradox in the light of structural proof theory. *Synthese* **190**(14), 2677–2716 (September 2013). <https://doi.org/10.1007/s11229-012-0061-7>
9. Marin, S., Morales, M., Straßburger, L.: A fully labelled proof system for intuitionistic modal logics. *Journal of Logic and Computation* **31**(3), 998–1022 (April 2021). <https://doi.org/10.1093/logcom/exab020>
10. Morales, M.: Unusual proof systems for modal logics with applications to decision problems. Ph.D. thesis, Polytechnic Institute of Paris, Palaiseau, France (2023), <https://theses.hal.science/tel-04546959>, prepared at École polytechnique
11. Negri, S.: Proofs and countermodels in non-classical logics. *Logica Universalis* **8**(1), 25–60 (March 2014). <https://doi.org/10.1007/s11787-014-0097-1>
12. Plotkin, G., Stirling, C.: A framework for intuitionistic modal logics. In: Halpern, J.Y. (ed.) *Theoretical Aspects of Reasoning About Knowledge: Proceedings of the 1986 Conference*, pp. 399–406. Morgan Kaufmann (1986), <https://dl.acm.org/doi/10.5555/1029786.1029823>
13. Simpson, A.K.: The Proof Theory and Semantics of Intuitionistic Modal Logic. Ph.D. thesis, University of Edinburgh, Edinburgh, Scotland, UK (1994), <http://hdl.handle.net/1842/407>
14. Wijesekera, D.: Constructive modal logics I. *Annals of Pure and Applied Logic* **50**(3), 271–301 (December 1990). [https://doi.org/10.1016/0168-0072\(90\)90059-B](https://doi.org/10.1016/0168-0072(90)90059-B)

A Proofs

Proof (of Theorem 4.7). The worlds W of $\mathcal{M}_{\mathcal{G}}$ are the labels of the sequent. Conditions fc and bc, as well as transitivity and reflexivity of $\leq_{\mathcal{G}}$, and monotonicity of V all follow by construction due to structural happiness. Thus, $\mathcal{M}_{\mathcal{G}}$ is a birelational model. It only remains to show the two properties about forcing, which we prove by mutual induction on the size of A , proceeding by case analysis on the main connective of A :

- $x:\perp^{\bullet} \in \mathcal{G}$: it is not possible for a happy sequent.
- $x:\perp^{\circ} \in \mathcal{G}$: we have $\mathcal{M}_{\mathcal{G}}, x \not\Vdash \perp$ by definition.
- $x:a^{\bullet} \in \mathcal{G}$: by Definition 4.2, $\mathcal{M}_{\mathcal{G}}, x \Vdash a$.
- $x:a^{\circ} \in \mathcal{G}$: it is not the case that $x:a^{\bullet} \in \mathcal{G}$ by happiness of x , hence, $\mathcal{M}_{\mathcal{G}}, x \not\Vdash a$ by Definition 4.2.
- $x:B \wedge C^{\bullet} \in \mathcal{G}$: by happiness of x , both $x:B^{\bullet} \in \mathcal{G}$ and $x:C^{\bullet} \in \mathcal{G}$. Then $\mathcal{M}_{\mathcal{G}}, x \Vdash B$ and $\mathcal{M}_{\mathcal{G}}, x \Vdash C$ by IH. Therefore, $\mathcal{M}_{\mathcal{G}}, x \Vdash B \wedge C$.
- $x:B \wedge C^{\circ} \in \mathcal{G}$: by happiness of x , either $x:B^{\circ} \in \mathcal{G}$ or $x:C^{\circ} \in \mathcal{G}$. Then either $\mathcal{M}_{\mathcal{G}}, x \not\Vdash B$ or $\mathcal{M}_{\mathcal{G}}, x \not\Vdash C$ by IH. Therefore, $\mathcal{M}_{\mathcal{G}}, x \not\Vdash B \wedge C$.
- Cases for $x:B \vee C^{\bullet} \in \mathcal{G}$ and $x:B \vee C^{\circ} \in \mathcal{G}$ are analogous.
- $x:B \supset C^{\bullet} \in \mathcal{G}$: consider any y with $x \leq_{\mathcal{G}} y$. By (mon^{\bullet}) -structural saturation, $y:B \supset C^{\bullet} \in \mathcal{G}$. By happiness of y , either $y:B^{\circ} \in \mathcal{G}$ or $y:C^{\bullet} \in \mathcal{G}$. By IH, either $\mathcal{M}_{\mathcal{G}}, y \not\Vdash B$ or $\mathcal{M}_{\mathcal{G}}, y \Vdash C$. Thus, $\mathcal{M}_{\mathcal{G}}, y \Vdash B$ implies $\mathcal{M}_{\mathcal{G}}, y \Vdash C$ for all y with $x \leq_{\mathcal{G}} y$. Therefore, $\mathcal{M}_{\mathcal{G}}, x \Vdash B \supset C$.
- $x:B \supset C^{\circ} \in \mathcal{G}$: by happiness of x , there is a world y such that $x \leq_{\mathcal{G}} y$, $y:B^{\bullet} \in \mathcal{G}$, and $y:C^{\circ} \in \mathcal{G}$. By IH, $\mathcal{M}_{\mathcal{G}}, y \Vdash B$ and $\mathcal{M}_{\mathcal{G}}, y \not\Vdash C$. Therefore, $\mathcal{M}_{\mathcal{G}}, x \not\Vdash B \supset C$.
- $x:\diamond B^{\bullet} \in \mathcal{G}$: by happiness of x , there is a world y such that $xR_{\mathcal{G}}y$ and $y:B^{\bullet} \in \mathcal{G}$. By IH, we have $\mathcal{M}_{\mathcal{G}}, y \Vdash B$ and, therefore, $\mathcal{M}_{\mathcal{G}}, x \Vdash \diamond B$.
- $x:\diamond B^{\circ} \in \mathcal{G}$: by happiness of x , we have $y:B^{\circ} \in \mathcal{G}$ for all worlds y such that $xR_{\mathcal{G}}y$. Thus, by IH, $\mathcal{M}_{\mathcal{G}}, y \not\Vdash B$ whenever $xR_{\mathcal{G}}y$. Therefore, $\mathcal{M}_{\mathcal{G}}, x \not\Vdash \diamond B$.
- $x:\square B^{\bullet} \in \mathcal{G}$: consider arbitrary y and z with $x \leq_{\mathcal{G}} y$ and $yR_{\mathcal{G}}z$. By (mon^{\bullet}) -structural saturation, $y:\square B^{\bullet} \in \mathcal{G}$. By happiness of y , we have $z:B^{\bullet} \in \mathcal{G}$. Thus, by IH, $\mathcal{M}_{\mathcal{G}}, z \Vdash B$ whenever $x \leq_{\mathcal{G}} y$ and $yR_{\mathcal{G}}z$. Therefore, $\mathcal{M}_{\mathcal{G}}, x \Vdash \square B$.
- $x:\square B^{\circ} \in \mathcal{G}$: by happiness of x , there exist y and z such that $x \leq_{\mathcal{G}} y$, $yR_{\mathcal{G}}z$, and $z:B^{\circ} \in \mathcal{G}$. By IH, $\mathcal{M}_{\mathcal{G}}, z \not\Vdash B$. Therefore, $\mathcal{M}_{\mathcal{G}}, x \not\Vdash \square B$. \square

Proof (of Theorem 5.16). First, all formulas that occur in a sequent must be subformulas of F . This means that the number of equivalence classes of the label equivalence relation \sim is finite. Next, observe that the algorithm only produces layered sequents, and the labels of each layer in a sequent \mathcal{G} form a tree defined by the $R_{\mathcal{G}}$ -relation. These trees have the following properties:

- (i) Their depth is bounded by the maximal modal depth of F . Whenever a new node is created or a formula moves from a parent to a child, one modality is removed (this happens in the rules \diamond^{\bullet} , \diamond° , \square^{\bullet} , and \square°).
- (ii) The number of children that a node can have is also bounded, because for every given depth, the number of different trees with that depth that do

not simulate each other is also finite. Note that the only rules that create new branches are \diamond^\bullet and \square° , where \diamond^\bullet is only applied when there is no child with the corresponding subformula, and \square° is only applied after the shrinking operation.

These two properties ensure that the number of layer equivalence classes (Definition 5.14) is finite. Since the layers in the sequent form a tree with respect to the layer ordering \leq , we must eventually see a repetition along each branch of that tree. This triggers our loopcheck, i.e., there are no allowed formulas left. \square

Proof (of Theorem 5.17). We are going to show that whenever the algorithm terminates in Step 1, then we can produce a proof of the formula F in \mathbf{labIK}^s . By inspecting the algorithm in Figure 4, we can observe that the set \mathfrak{S}_i that is computed at each round is in fact the set of leaves of a derivation tree with root \mathcal{G}_0 , over the rules \mathbf{sat} , \mathbf{shrk} , and \mathbf{lift} .

We have already observed (Propositions 5.6 and 5.9) that the rules \mathbf{sat} and \mathbf{lift} are derivable in \mathbf{labIK}^s . Furthermore, every instance of the \mathbf{shrk} -rule is an instance of the \mathbf{weak} with an augmentation of the premise by additional relational atoms. We are now going to show that these additional relational atoms are never used in the derivation by other rule instances. Therefore, all instances of \mathbf{shrk} can be replaced by \mathbf{weak} , and we obtain a derivation in \mathbf{labIK}^s of the sequent \mathcal{G}_0 , from which we can get $r:F^\circ$, as required by Theorem 3.4.

The first observation to make is that in every sequent \mathcal{G} that the algorithm visits, the relation $R_{\mathcal{G}}$ defines a tree-structure on each layer. Assume now we have an instance of the \mathbf{shrk} -rule with conclusion \mathcal{G} and premise $\mathcal{G}' = \mathcal{G}[y/x]$, eliminating all occurrences of the label y . Then all labels u with $yR_{\mathcal{G}}u$ are redundant in \mathcal{G}' , by using the same simulation S as for y in \mathcal{G} . Thus, whenever the shrinking removes a label y , it also removes the whole R -subtree rooted at that label. Therefore, a shrinking never introduces additional R -relational atoms.

The next observation to make is that the \mathbf{shrk} -rule only removes labels that are in a layer that is maximal with respect to the layer ordering \leq . The reason is that the only way to create new layers is via the \mathbf{lift} -rule, and that only happens if the layers below that new layer are already shrunk. Therefore, a shrinking only introduces additional relational atoms of the form $v \leq w$, where w is in a layer that is maximal and almost happy.

The only inference rule in \mathbf{labIK}^s that uses \leq -relational atoms to move formulas into another label is the \mathbf{mon}^\bullet -rule that is employed in the lifting. But there it only uses the freshly created \leq -relational atoms for the new layer. Hence, the additional relational atoms are never used, and we can replace all \mathbf{shrk} -instances by \mathbf{weak} . \square

Proof (of Theorem 5.18). Assume the algorithm in Figure 4 terminates in Step 2b, i.e., there is an almost happy sequent \mathcal{G}_i with no allowed formulas. That means that whenever there is an unhappy formula, it is of shape $x:A \supset B^\circ$ or $x:\square A^\circ$, and x is in a layer L , such that (i) L is maximal with respect to \leq , and (ii) there is a layer L' with $L' \sim L$ and $L' < L$.

We have (i) because if a formula is not allowed, then there is no other allowed formula in the same layer, and we have (ii) because otherwise the unhappy $x:A \supset B^\circ$ or $x:\square A^\circ$ would be allowed.

We can now take the bijection $f: L \rightarrow L'$ that makes L and L' equivalent and define \mathcal{G}'_i to be the sequent obtained from \mathcal{G}_i by adding the relational atom $v \leq f(v)$ for every $v \in L$. This makes every unhappy $x:A \supset B^\circ$ or $x:\Box A^\circ$ with $x \in L$ happy, and does not affect happiness of other formulas in the sequent.

We can repeat this for all layers that contain unhappy formulas, yielding eventually a happy sequent, to which we can apply Theorem 4.7. \square

B Examples

Example B.1. We report a proof of sequent $1:(\Diamond p \supset \Box q) \supset \Box(p \supset q)^\circ$. For reasons of space, we avoid repeating the principal formulas in the premise(s) of the rules, and we abbreviate with \mathcal{R} the sequent $1 \leq 2, 2R3, 3 \leq 4, 2 \leq 5, 5R4, 1 \leq 5$.

$$\begin{array}{c}
\frac{\text{id} \frac{\mathcal{R}, 4:p^\bullet, 4:p^\circ, 4:q^\circ}{\Diamond^\circ} \quad \text{id} \frac{\mathcal{R}, 4:p^\bullet, 4:q^\bullet, 4:q^\circ}{\Box^\bullet}}{\mathcal{R}, 4:p^\bullet, 5:\Diamond p^\circ, 4:q^\circ} \quad \frac{\mathcal{R}, 4:p^\bullet, 5:\Box q^\bullet, 4:q^\circ}{\Box^\bullet}}{\frac{1 \leq 2, 2R3, 3 \leq 4, 2 \leq 5, 5R4, 1 \leq 5, 4:p^\bullet, 5:\Diamond p \supset \Box q^\bullet, 4:q^\circ}{1 \leq 2, 2R3, 3 \leq 4, 2 \leq 5, 5R4, 1 \leq 5, 4:p^\bullet, 1:\Diamond p \supset \Box q^\bullet, 4:q^\circ}}{\text{mon}^\bullet} \\
\frac{\frac{1 \leq 2, 2R3, 3 \leq 4, 2 \leq 5, 5R4, 4:p^\bullet, 1:\Diamond \supset \Box q^\bullet, 4:q^\circ}{1 \leq 2, 2R3, 3 \leq 4, 4:p^\bullet, 1:\Diamond p \supset \Box q^\bullet, 4:q^\circ}}{\text{bc}}}{\frac{1 \leq 2, 2R3, 1:\Diamond p \supset \Box q^\bullet, 3:p \supset q^\circ}{1:\Diamond p \supset \Box q^\bullet, 1:\Box(p \supset q)^\circ}}{\text{bc}} \\
\frac{1:\Diamond p \supset \Box q^\bullet, 1:\Box(p \supset q)^\circ}{1:(\Diamond p \supset \Box q) \supset \Box(p \supset q)^\circ}
\end{array}$$

Example B.2. Let us consider the formula $A = ((a \supset \Box b) \supset \perp) \supset \perp$. To check whether the formula is valid in IK , we apply the procedure described in Section 5, by taking $\mathcal{G}_0 = r \leq r, r:A^\circ$. On the right of Figure 5 is displayed one of the sequents \mathcal{G}_i produced by the algorithm, where $\mathcal{H} = (a \supset \Box b) \supset \perp^\bullet, a \supset \Box b^\circ$. Observe that \mathcal{G}_i is layered, and that all the layers $L_0 - L_5$ are happy, while layer L_6 is almost happy: formula $10:\Box b^\circ$ is not happy. Moreover, the sequent can be shrunk: again, we use colours to indicate label equivalence, and it holds that $11 \sim 12$, and furthermore that $11|12$. In the middle of the Figure is represented the result applying shrinking to \mathcal{G}_i . After the shrinking, it holds that layer L_6 and layer L_4 are equivalent. Thus, the unhappy formula $10:\Box b^\circ$ is not allowed and, since there are no allowed formulas, the algorithm terminates. At this point, using the construction described above, we make formula $10:\Box b^\circ$ happy by adding the relational atoms $10 \leq 5$ and $11 \leq 6$ to the sequent. The result is represented on the left-hand side of Figure 5. It is easy to check that this sequent is happy, and thus defines a countermodel for A .

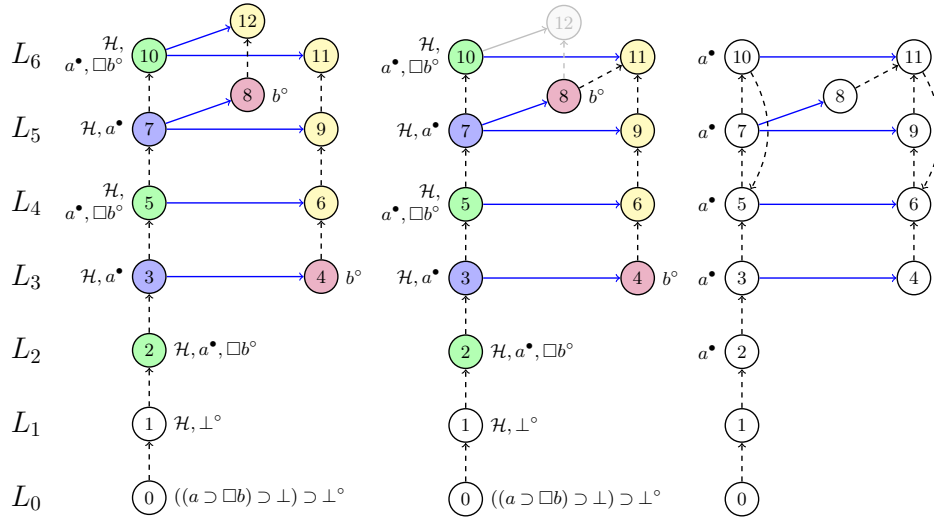


Figure 5. *Left:* Graphical representation of a sequent \mathcal{G} produced by the algorithm, where $\mathcal{H} = (a \supset \Box b) \supset \perp^\bullet, a \supset \Box b^\circ$. *Middle:* The result of shrinking \mathcal{G} . *Right:* The countermodel produced according to the procedure described in the proof of Theorem 5.18.