MOIN:
A Nested Sequent Theorem Prover for Intuitionistic Modal Logics
(System Description)

Marianna Girlando and Lutz Straßburger

Inria, Equipe Partout, Centre Inria Saclay & École Polytechnique, LIX, France

Abstract. We present a simple Prolog prover for intuitionistic modal logics based on nested sequent proof systems. We have implemented single-conclusion systems (Gentzen-style) and multi-conclusion systems (Maehara-style) for all logics in the intuitionistic modal IS5-cube. While the single-conclusion system are better investigated and have an internal cut-elimination, the multi-conclusion systems can provide a countermodel in case the proof search fails. To our knowledge this is the first automated theorem prover for intuitionistic modal logics. For wider usability, we also implemented classical normal modal logics in the S5-cube.

Keywords: Intuitionistic modal logic, nested sequents, Prolog.

1 Introduction

In the last decade, nested sequent calculi have been successfully used to define cut-free deductive systems for various normal modal logics (classical [3], intuitionistic [23,17], constructive [1] and tense logics [10]) and non-normal modal logics [14]. For some variants there also exist focused calculi [5,6].

Even though many of these calculi yield terminating decision procedures, most of them have not (yet) been implemented in automated theorem provers. In fact, the only implementations of nested sequent systems that we are aware of are for non-normal logics [14,13] and normal conditional logics [18]. But for the more widely used classical and intuitionistic normal modal logics in the S5-cube no implementations using nested sequents exist. In fact, for intuitionistic modal logics we are not aware of any automated theorem provers.

For this reason we present here a modular Prolog implementation for nested sequent calculi for all intuitionistic normal modal logics in the IS5-cube (see Figure 1). For the systems whose decidability is known, our prover terminates. For IS4, IK4, and ID4, for which decidability is still an open problem, we cannot ensure termination. For the sake of higher usability, we included in the implementation also all classical normal modal logics in the S5-cube, but they are not discussed here, as their implementation is rather straightforward [3].

The prover, called MOIN[1] (for *MOdal Intuitionistic Nested sequents*), is inspired by the *lean deduction* methodology introduced in [2]: the program is con-

---

[1] Available at www.lix.polytechnique.fr/Labo/Lutz.Strassburger/Software/Moin.

stituted by a set of Prolog clauses, one for each inference rule of the calculus, and the proof search is provided by the Prolog depth-first mechanism[2]. The advantages of this approach, as stated in [2], are high degree of adaptability and safety (i.e. the Prolog clauses are easy to verify), and high efficiency.

This article is organized as follows. Section 2 introduces intuitionistic modal logics; Section 3 and 4 present nested sequents (single- and multi-conclusion) and the proof systems we use. Section 5 treats termination and countermodel construction, and Section 6 describes the main features of MOIN.

2   Intuitionistic modal logics

We consider here the variant of intuitionistic modal logic that has first been studied in [7] and [20], and then been investigated in detail by Simpson in his PhD-thesis [22]. It is obtained from intuitionistic propositional logic, by adding the two modalities $\square$ and $\lozenge$, together with the necessitation rule (if A is provable, then so is $\square A$), and the following five axioms:

$\mathsf{k_1} : \square(A \supset B) \supset (\square A \supset \square B)$ $\mathsf{k_3} : \lozenge(A \vee B) \supset (\lozenge A \vee \lozenge B)$ $\mathsf{k_5} : \lozenge\bot \supset \bot$

$\mathsf{k_2} : \square(A \supset B) \supset (\lozenge A \supset \lozenge B)$ $\mathsf{k_4} : (\lozenge A \supset \square B) \supset \square(A \supset B)$

In classical modal logic, only $\mathsf{k_1}$ is added, as the others follow via the De Morgan duality between $\square$ and $\lozenge$, which is not present in the intuitionistic case.

But as in the classical case, we can extend the logic by additional axioms. We restrict ourselves here to the five axioms d, t, b, $\mathsf{4}$ and $\mathsf{5}$, which in the classical as well as in the intuitionistic case generate 15 different logics, which can be arranged in the so-called $\mathsf{S5}$-cube (classical) or $\mathsf{IS5}$-cube (intuitionistic). Figure 1 shows the intuitionistic variant of that cube, together with the five aforementioned axioms. The reason for the restriction to these logics in this presentation is that the nested sequent proof systems for them are particularly well-studied [3,8,23,17,5,6]. The Prolog code is written in a way that it can easily be extended to other logics. The semantics of the logics in the $\mathsf{IS5}$-cube is defined in terms of *bi-relational models*, introduced in [7,20,22]. No complexity results are known for these logics; however, their complexity has to be at least PSPACE-hard, as intuitionistic propositional logic is PSPACE-complete.

3   Nested sequents

Nested sequents have been introduced independently in [11], [3], and [21]. Whereas an ordinary sequent is a list (or multiset or set) of formulas, a nested sequent is a tree of lists (or multisets or sets) of formulas. In the classical setting a nested sequent Γ is written as

$$\Gamma ::= A_1, \ldots, A_n, [\Gamma_1], \ldots, [\Gamma_m] \quad ,$$

---

[2] An introduction on Prolog theorem provers design is Jens Otten's tutorial at TABLEAUX 2019 [19].

$$\text{d} : \Box A \supset \Diamond A$$
$$\text{t} : (A \supset \Diamond A) \wedge (\Box A \supset A)$$
$$\text{b} : (A \supset \Box \Diamond A) \wedge (\Diamond \Box A \supset A)$$
$$\text{4} : (\Diamond \Diamond A \supset \Diamond A) \wedge (\Box A \supset \Box \Box A)$$
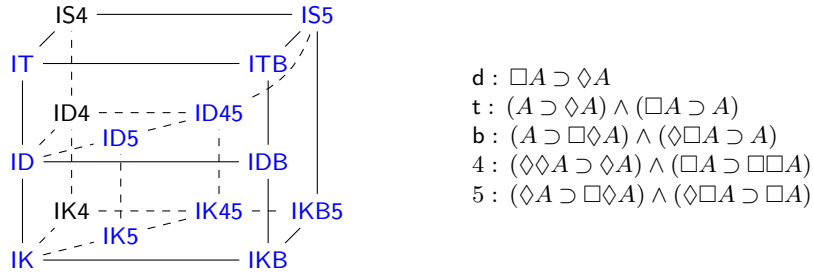$$\text{5} : (\Diamond A \supset \Box \Diamond A) \wedge (\Diamond \Box A \supset \Box A)$$

Fig. 1. The intuitionistic modal IS5-cube

where $A_1, \ldots, A_n$ are formulas and $\Gamma_1, \ldots, \Gamma_m$ are nested sequents. This corresponds to a one-sided setting, and the formula interpretation $fm(\Gamma)$ of the sequent above is $fm(\Gamma) = A_1 \vee \cdots \vee A_n \vee \Box fm(\Gamma_1) \vee \cdots \vee \Box fm(\Gamma_m)$.

This setting is used by Brünnler in [3], to give cut-free systems for all logics in the classical S5-cube, including a decision procedure. He has also shown how a finite countermodel can be extracted from a failed proof search. We implemented his method in Moin, but we will not go into details here because there are already many different provers for classical modal logic in the literature[3], and the details of our implementation for classical modal logics follow straightforwardly from our implementation for intuitionistic modal logics, that we discuss below.

The first observation is that for intuitionistic logic we need two-sided sequents, which are formally generated by

$$\Gamma ::= A_1^{\bullet}, \ldots, A_n^{\bullet}, B_1^{\circ}, \ldots, B_k^{\circ}, [\Gamma_1], \ldots, [\Gamma_m] \quad ,$$

where $A_1, \ldots, A_n$ are the formulas that would occur on the left of the turnstile if there was a turnstile, $B_1, \ldots, B_k$ are the formulas that would occur on the right of the turnstile if there was a turnstile, and $\Gamma_1, \ldots, \Gamma_m$ are nested sequents. We use the $\bullet$- and $\circ$-superscripts as polarity indication for formulas.

In this (multi-conclusion) intuitionistic two-sided setting, it is not possible to give a formula interpretation as in the one-side classical case. However, this is possible in the single-conclusion setting, where there is only one formula B° in the whole sequent. Single-conclusion sequents are formally generated by

$$\Gamma ::= \Lambda, B^{\circ} \mid \Lambda, [\Gamma] \qquad \Lambda ::= \emptyset \mid A^{\bullet}, \Lambda \mid [\Lambda_1], \Lambda_2$$

where Γ stands for a (non-empty) sequent that contains exactly one formula with $\circ$-polarity, and Λ for a (possibly empty) sequent in which all formulas have $\bullet$-polarity. Then the formula interpretation is:

$$fm(\emptyset) = \bot \qquad fm(A^{\bullet}, \Lambda) = A \wedge fm(\Lambda) \qquad fm([\Lambda_1], \Lambda_2) = \Diamond fm(\Lambda_1) \wedge fm(\Lambda_2)$$
$$fm(\Lambda, B^{\circ}) = fm(\Lambda) \supset B \qquad fm(\Lambda, [\Gamma]) = fm(\Lambda) \supset \Box fm(\Gamma)$$

In order define inference rules on nested sequents, we need the notion of context, denoted as $\Gamma\{\cdot\}$, which is a nested sequent that contains exactly one occurrence of the hole $\{\cdot\}$ in the place of a formula (of either polarity). Then we write

---

[3] For an extensive list of provers for classical modal logic, see http://www.cs.man.ac.uk/~schmidt/tools/, maintained by Renate Schmidt.

$\Gamma\{A\}$ (respectively $\Gamma\{\Delta\}$, respectively $\Gamma\{\emptyset\}$) for the sequent obtained from the context $\Gamma\{\cdot\}$ by replacing the hole $\{\cdot\}$ by the formula A (respectively the sequent Δ, respectively deleting the hole). Finally, we define $\Gamma^*\{\cdot\}$ to be the context that is obtained from $\Gamma\{\cdot\}$ by removing all $\circ$-formulas.

Example 3.1 The nested sequent $\Gamma\{[C^\bullet]\} = A^\bullet, B^\bullet, [C^\bullet], [D^\bullet, E^\bullet, [F^\bullet, G^\circ]]$ is the result of filling context $\Gamma\{\cdot\} = A^\bullet, B^\bullet, \{\cdot\}, [D^\bullet, E^\bullet, [F^\bullet, G^\circ]]$ with $[C^\bullet]$. As $\Gamma\{[C^\bullet]\}$ is a single-conclusion sequent, we can give its formula interpretation: $(A{\wedge}B{\wedge}\Diamond C){\supset}\Box((D{\wedge}E){\supset}\Box(F{\supset}G))$. Then, for $\Gamma^*\{\cdot\} = A^\bullet, B^\bullet, \{\cdot\}, [D^\bullet, E^\bullet, [F^\bullet]]$, we have $\Gamma^*\{A^\circ\} = A^\bullet, B^\bullet, A^\circ, [D^\bullet, E^\bullet, [F^\bullet]]$, whose formula interpretation is $(A \wedge B \wedge \Diamond(D \wedge E \wedge \Diamond F) \supset A)$. Finally, $\Gamma\{\emptyset\} = A^\bullet, B^\bullet, [D^\bullet, E^\bullet, [F^\bullet, G^\circ]]$.

4   The Proof Systems

There are two kinds of nested sequent systems for intuitionistic logic: single-conclusion [23,17,6] in the style of Gentzen [9] and multi-conclusion [12], in the style of Maehara [15]. In MOIN, we implemented both. More precisely, we implemented the single-conclusion systems of [23] and (a minor variation of) the multi-conclusion systems presented in [12].

Figure 2 shows the basic single-conclusion system NIKs for the modal logic IK as presented in [23]. Formulas marked in gray are formally not part of the rules, but are part of the implementation in order to ease termination checks, and to reduce the search space. Gray formulas can never be principal in a rule application; they are kept in the premises for book-keeping. The use of Γ^* in the $\supset_s^\bullet$-rule is due to the fact that every sequent occurring in a proof has to be a single-conclusion sequent.

Then, Figure 3 shows the extension rules corresponding to the axioms in Figure 1. They are taken from [23], and we refer the reader to [23] or [17] for a discussion on which set of rules gives which logic in the cube in Figure 1. For a set $X \subseteq \{d, t, b, 4, 5\}$, we write NIKs $+ X$ for the system obtained from NIKs by adding the corresponding rules in Figure 3. If $d \in X$ we need in some cases also the $d_s^{[]}$-rule shown in Figure 4 (see [16] for more details).[4]

Finally, Figure 4 shows the rules needed for the multi-conclusion system. For each $X \subseteq \{d, t, b, 4, 5\}$, we write NIKm$+X$ for the system obtained from NIKs$+X$ by removing the rules with an s in the subscript and replacing them with the corresponding rules in Figure 4 with an m in the subscript. The $d_s^{[]}$-rule is never needed in the multi-conclusion system. For more details see [12].

5   Termination and Countermodel Construction

All the classical modal logics in the S5-cube are decidable, and a simple loop check in the rules that create a new nesting is enough to ensure termination

---

[4] If $d_s^{[]}$ is in the system, the rules d_s° and $d^\bullet$ can be omitted, as they become admissible.

$$\bot^\bullet \; \frac{}{\Gamma\{\bot^\bullet\}} \qquad\qquad\qquad \mathsf{id} \; \frac{}{\Gamma\{a^\bullet, a^\circ\}}$$

$$\wedge^\bullet \; \frac{\Gamma\{A \wedge B^\bullet, A^\bullet, B^\bullet\}}{\Gamma\{A \wedge B^\bullet\}} \qquad\qquad \wedge^\circ \; \frac{\Gamma\{A \wedge B^\circ, A^\circ\} \quad \Gamma\{A \wedge B^\circ, B^\circ\}}{\Gamma\{A \wedge B^\circ\}}$$

$$\vee^\bullet \; \frac{\Gamma\{A \vee B^\bullet, A^\bullet\} \quad \Gamma\{A \vee B^\bullet, B^\bullet\}}{\Gamma\{A \vee B^\bullet\}} \qquad \vee^\circ_{s1} \; \frac{\Gamma\{A^\circ\}}{\Gamma\{A \vee B^\circ\}} \quad \vee^\circ_{s2} \; \frac{\Gamma\{B^\circ\}}{\Gamma\{A \vee B^\circ\}}$$

$$\supset^\bullet_s \; \frac{\Gamma^*\{A \supset B^\bullet, A^\circ\} \quad \Gamma\{A \supset B^\bullet, B^\bullet\}}{\Gamma\{A \supset B^\bullet\}} \qquad\qquad \supset^\circ_s \; \frac{\Gamma\{A^\bullet, B^\circ\}}{\Gamma\{A \supset B^\circ\}}$$

$$\Box^\bullet \; \frac{\Gamma\{\Box A^\bullet, [A^\bullet, \Delta]\}}{\Gamma\{\Box A^\bullet, [\Delta]\}} \quad \Diamond^\bullet \; \frac{\Gamma\{\Diamond A^\bullet, [A^\bullet]\}}{\Gamma\{\Diamond A^\bullet\}} \quad \Box^\circ_s \; \frac{\Gamma\{[A^\circ]\}}{\Gamma\{\Box A^\circ\}} \quad \Diamond^\circ \; \frac{\Gamma\{\Diamond A^\circ, [A^\circ, \Delta]\}}{\Gamma\{\Diamond A^\circ, [\Delta]\}}$$

Fig. 2. System NIKs

$$\mathsf{d}^\circ_s \; \frac{\Gamma\{\Diamond A^\circ, [A^\circ]\}}{\Gamma\{\Diamond A^\circ\}} \quad \mathsf{t}^\circ_s \; \frac{\Gamma\{\Diamond A^\circ, A^\circ\}}{\Gamma\{\Diamond A^\circ\}} \quad \mathsf{b}^\circ_s \; \frac{\Gamma\{[\Delta, \Diamond A^\circ], A^\circ\}}{\Gamma\{[\Delta, \Diamond A^\circ]\}} \quad \mathsf{4}^\circ_s \; \frac{\Gamma\{\Diamond A^\circ, [\Diamond A^\circ, \Delta]\}}{\Gamma\{\Diamond A^\circ, [\Delta]\}}$$

$$\mathsf{d}^\bullet \; \frac{\Gamma\{\Box A^\bullet, [A^\bullet]\}}{\Gamma\{\Box A^\bullet\}} \quad \mathsf{t}^\bullet \; \frac{\Gamma\{\Box A^\bullet, A^\bullet\}}{\Gamma\{\Box A^\bullet\}} \quad \mathsf{b}^\bullet \; \frac{\Gamma\{[\Delta, \Box A^\bullet], A^\bullet\}}{\Gamma\{[\Delta, \Box A^\bullet]\}} \quad \mathsf{4}^\bullet \; \frac{\Gamma\{\Box A^\bullet, [\Box A^\bullet, \Delta]\}}{\Gamma\{\Box A^\bullet, [\Delta]\}}$$

$$\mathsf{5}^\circ_{1s} \; \frac{\Gamma\{[\Delta, \Diamond A^\circ], \Diamond A^\circ\}}{\Gamma\{[\Delta, \Diamond A^\circ]\}} \qquad \mathsf{5}^\circ_{2s} \; \frac{\Gamma\{[\Delta, \Diamond A^\circ], [\Diamond A^\circ, \Sigma]\}}{\Gamma\{[\Delta, \Diamond A^\circ], [\Sigma]\}} \qquad \mathsf{5}^\circ_{3s} \; \frac{\Gamma\{[\Delta, \Diamond A^\circ, [\Diamond A^\circ, \Sigma]]\}}{\Gamma\{[\Delta, \Diamond A^\circ, [\Sigma]]\}}$$

$$\mathsf{5}^\bullet_1 \; \frac{\Gamma\{[\Delta, \Box A^\bullet], \Box A^\bullet\}}{\Gamma\{[\Delta, \Box A^\bullet]\}} \qquad \mathsf{5}^\bullet_2 \; \frac{\Gamma\{[\Delta, \Box A^\bullet], [\Box A^\bullet, \Sigma]\}}{\Gamma\{[\Delta, \Box A^\bullet], [\Sigma]\}} \qquad \mathsf{5}^\bullet_3 \; \frac{\Gamma\{[\Delta, \Box A^\bullet, [\Box A^\bullet, \Sigma]]\}}{\Gamma\{[\Delta, \Box A^\bullet, [\Sigma]]\}}$$

Fig. 3. Intuitionistic $\Diamond^\circ$- and $\Box^\bullet$-rules for the axioms d, t, b, 4, and 5.

$$\mathsf{d}^{[]}_s \; \frac{\Gamma\{[\,]\}}{\Gamma\{\emptyset\}} \qquad\qquad \supset^\bullet_m \; \frac{\Gamma\{A \supset B^\bullet, A^\circ\} \quad \Gamma\{A \supset B^\bullet, B^\bullet\}}{\Gamma\{A \supset B^\bullet\}}$$

$$\supset^\circ_m \; \frac{\Gamma^*\{A^\bullet, B^\circ\}}{\Gamma\{A \supset B^\circ\}} \qquad \Box^\circ_m \; \frac{\Gamma^*\{[A^\circ]\}}{\Gamma\{\Box A^\circ\}} \qquad \vee^\circ_m \; \frac{\Gamma\{A \vee B^\circ, A^\circ, B^\circ\}}{\Gamma\{A \vee B^\circ\}}$$

$$\mathsf{d}^\circ_m \; \frac{\Gamma\{\Diamond A^\circ, [A^\circ]\}}{\Gamma\{\Diamond A^\circ\}} \quad \mathsf{t}^\circ_m \; \frac{\Gamma\{\Diamond A^\circ, A^\circ\}}{\Gamma\{\Diamond A^\circ\}} \quad \mathsf{b}^\circ_m \; \frac{\Gamma\{[\Delta, \Diamond A^\circ], A^\circ\}}{\Gamma\{[\Delta, \Diamond A^\circ]\}} \quad \mathsf{4}^\circ_m \; \frac{\Gamma\{\Diamond A^\circ, [\Diamond A^\circ, \Delta]\}}{\Gamma\{\Diamond A^\circ, [\Delta]\}}$$

$$\mathsf{5}^\circ_{1m} \; \frac{\Gamma\{[\Delta, \Diamond A^\circ], \Diamond A^\circ\}}{\Gamma\{[\Delta, \Diamond A^\circ]\}} \qquad \mathsf{5}^\circ_{2m} \; \frac{\Gamma\{[\Delta, \Diamond A^\circ], [\Diamond A^\circ, \Sigma]\}}{\Gamma\{[\Delta, \Diamond A^\circ], [\Sigma]\}} \qquad \mathsf{5}^\circ_{3m} \; \frac{\Gamma\{[\Delta, \Diamond A^\circ, [\Diamond A^\circ, \Sigma]]\}}{\Gamma\{[\Delta, \Diamond A^\circ, [\Sigma]]\}}$$

Fig. 4. The structural $\mathsf{d}^{[]}_s$-rule, and the rules for the multi-conclusion system $\mathsf{NIKm + X}$

(see [3] for details). It is also explained in [3] how to obtain a finite countermodel from a failed proof search, and we implemented that method in Moin.

In the intuitionistic case, not all logics in the IS5-cube are known to be decidable. For the logics ID4, IK4 and IS4, this is still an open problem. But, as already observed in [22], all other logics are decidable. Termination for logics that do not need the the 4- and 5-axioms can be proved similarly as in the

case of propositional intuitionistic logic: the only loop-check to be performed concerns application of the $\supset_s^\bullet$-rule, where it is needed to check whether the current sequent already occurs in the derivation branch.

For the logics containing the 5-axiom, it is enough to restrict the depth of a sequent to 1: when the $\Diamond^\bullet$-, $\Box^\circ$-, or one of the d-rules creates a new nesting at a depth > 1, the nesting is introduced at the root level of the sequent. Completeness of the resulting proof system has been proved in [8]. Thus, the restriction on the $\supset_s^\bullet$ rule mentioned above is the only loop-check needed for these systems.

For the remaining logics (ID4, IK4, IS4), decidability is not known, and we implement only two naive loop-checks: one for $\supset_s^\bullet$ mentioned above, and one that is similar to the loop-check for classical modal logics. Refer to [22] for an explanation on why this strategy does not suffice to ensure termination.

When using the single-conclusion systems, termination and completeness can easily be shown via cut-elimination [23]. However, it is not clear how to obtain a finite countermodel from a failed proof search. The reason is that there is not such a close correspondence between models and sequents as for classical modal logic. For this reason we also implemented the multi-conclusion systems of [12]. They are related to the intuitionistic (bi-relational) Kripke models [7,20], as classical nested sequent systems to classical Kripke models. The tree-structure of the sequent tree corresponds to the R-relation (the accessibility relation for the modalities), and the tree-structure of the proof tree corresponds to the intuitionistic $\leq$-relation (often interpreted as future-relation). We follow the construction of [12] for obtaining a countermodel from a failed proof search, with certain simplifications, as in our case the model is always finite.

Termination for NIKm + X is ensured by the same arguments as for NIKs + X, with the difference that the loop-check (which is performed for $\supset_s^\bullet$ in NIKs) is performed for $\supset_m^\circ$ and $\Box_m^\circ$ in NIKm + X.

6   The Prolog Implementation

MOIN implements the nested sequents for classical modal logics from [3], and the single- and multi-conclusion calculi for intuitionistic modal logics shown in Figures 2,3, 4). The prover is composed of a set clauses, each implementing a rule of the sequent calculus. The structure of MOIN reflects the modularity of the calculi: implementations of stronger systems are obtained by adding the clauses corresponding to the rules to the set of clauses for weaker systems.

The biggest difficulty in implementing nested sequents lies in the choice of the data structure. For ordinary sequents, it is straightforward to use the Prolog-lists. However, nested sequents are not lists of formulas, but trees of lists of formulas, and Prolog does not come with an efficient representation of trees, which would allow formulas occurring in lists inside a node of the tree to be easily accessed and replaced. There exist several approaches to overcome this difficulty. The implementation of conditional modal logics in [18] and of non-normal modal logics in [14] implement nested sequents as nested lists, while the implementation of non-normal modal logics in [13] uses a tree structure, in which nodes are

annotated with sequents. We follow a different approach, representing nested sequents as Prolog lists with *annotations*. This can be compared to labelled calculi [22]: an annotation is an index labelling formulas of a nested sequent, with formulas occurring at the same node sharing the same annotation. This data structure allows for an easier countermodel extraction from a failed branch.

Propositional variables are represented in Moin as Prolog atoms `a`,`b`,...; $\perp$ and $\top$ are Prolog `false` and `true`, and the connectives $\neg$, $\wedge$, $\vee$, $\supset$, $\square$ and $\lozenge$ are respectively represented by `~`, `v` `^`, `->`, `!` and `?`. Nested sequents for classical modal logic are represented by means of two Prolog lists `Rel,Seq`. `Seq` is a list of triples `(X,F,Sign)`, where `F` is a formula in Moin syntax, `X` is the annotation of `F`, i.e., an integer keeping track of the component of the nested sequent at which the formula occurs, and `Sign` is either `+` or `-`. Rules can only be applied to formulas with a positive sign, while formulas with a negative sign are used for book-keeping (they are the gray formulas from Figures 2,3 and 4). `Rel` is a list of pairs `(X,Y)`, representing the parent-child relation between nodes of a nested sequent. Single-conclusion nested sequents are represented as `Rel,Lambda,Out`, where `Lambda` is a list of elements `(X,F,Sign)` storing the •-formulas, and `Out` is a pair `(X,F)` storing the ∘-formula. Finally, multi-conclusion nested sequents are represented by means of three lists `Rel,Gamma,Delta`, where `Gamma` and `Delta` are lists of elements `(X,F,Sign)`, respectively representing •- and ∘-formulas.

Proof search is invoked by the predicate `derive(PS,Axioms,F)`, where `F` is the formula to be checked, `PS` selects the proof system (`k` for classical, `i` for intuitionistic single-conclusion, `m` for intuitionistic multi-conclusion nested sequents), and `Axioms` is a (possibly empty) list specifying the additional axioms to be used (d, t, b, 4 and 5). For instance, `derive(i,[b,5], (?a -> !?a) ^ (?!a -> !a))` triggers the derivation of axiom 5 in NIKs $+ \{$b, 5$\}$. The predicate `derive` queries the corresponding predicate `prove_k\4`, `prove_i\3` or `prove_m\4`.These predicates are recursively invoked and generate the proof-search tree (and the countermodel). To ensure termination, application of some rules needs to be restricted (see Section 5). The loop-checks are implemented by auxiliary predicates. The application of `prove` to a branch stops when an axiom clause is reached (success), or when no clause succeeds, producing a failed branch. In case of success of the proof-search, Moin produces a LaTeX file containing the derivation. Otherwise, for classical and intuitionistic multi-conclusion sequents, Moin prints out a countermodel in a LaTeX file. While in the classical system the information contained in the leaf of a failed branch is enough to extract a countermodel, this is not the case for NIKm $+$ X, where all the failed branches need to be considered.

7   Performances

Due to the absence of benchmarks for intuitionistic modal logics, we have measured the performance Moin with some adhoc (valid and non valid) formulas shown in Figure 5. The construction of a set of benchmark formulas has to be postponed to future work, as the 15 different logics in the IS5-cube make this not an easy task.

$$A = \Box_1, \ldots, \Box_8(a \supset b) \supset (\Box_1, \ldots, \Box_8\, a \supset \Box_1, \ldots, \Box_8\, b)$$
$$B = \ (\Diamond_1, \ldots, \Diamond_5 a \supset \Box_1, \ldots, \Box_5 b) \supset \Box_1, \ldots, \Box_5(a \supset b)$$
$$C = \qquad (\Box a \wedge \neg\Box\Box a) \supset \Diamond(\Box\Box a \wedge \neg\Box\Box\Box a)$$
$$D = \qquad\quad \Box(\Box(a \supset \Box a) \supset a) \supset (\Diamond\Box a \supset a)$$
$$E = \ (\Diamond(a \vee b) \wedge \Box(\Diamond(a \vee b))) \supset (\Diamond\Diamond\Diamond\Diamond a \vee \Diamond\Diamond\Diamond\Diamond b)$$

Fig. 5. Formulas A and B are generated by adding modal operators to k_1 and k_4, and are both derivable in all systems. Formulas C and D are from [4]; C is valid in all systems with 4 and D in all systems with b. Formula E is not derivable in any system.

	IK	ID	IT	IKB	IK5	IK45	IKB5	IDB	ID5	ID45	ITB	IS5
A	$0,1\cdot0,1$	$0,1\cdot0,7$	$0,2\cdot0,2$	$0,1\cdot0,1$	$19,6\cdot0,2$	$113,9\cdot0,3$	$16,1\cdot0,3$	$0,2\cdot38,7$	$*\cdot*$	$*\cdot*$	$0,2\cdot0,2$	$32,8\cdot0,5$
B	$0,1\cdot0,1$	$0,1\cdot0,2$	$0,1\cdot0,1$	$0,1\cdot0,1$	$0,1\cdot0,1$	$0,1\cdot0,2$	$0,1\cdot0,2$	$0,1\cdot0,5$	$0,2\cdot40,2$	$0,2\cdot46,2$	$0,1\cdot0,1$	$0,1\cdot0,2$
C	$0,1\cdot0,1$	$0,3\cdot0,1$	$0,1\cdot0,1$	$0,6\cdot0,1$	$0,2\cdot0,1$	$0,1\cdot0,1$	$0,2\cdot0,1$	$3,0\cdot0,1$	$8,1\cdot0,1$	$0,1\cdot0,1$	$1,2\cdot0,1$	$0,1\cdot0,1$
D	$0,1\cdot0,2$	$0,1\cdot0,1$	$0,1\cdot0,1$	$0,1\cdot0,1$	$0,1\cdot0,1$	$0,1\cdot0,1$	$0,1\cdot0,1$	$0,1\cdot0,1$	$0,1\cdot0,1$	$0,1\cdot0,1$	$0,1\cdot0,1$	$0,1\cdot0,1$
E	$0,1\cdot0,1$	$0,1\cdot0,1$	$0,1\cdot0,1$	$0,1\cdot0,1$	$0,1\cdot0,1$	$0,1\cdot0,1$	$0,1\cdot0,1$	$0,1\cdot0,1$	$0,1\cdot0,3$	$0,1\cdot0,8$	$0,1\cdot0,1$	$0,1\cdot0,1$

Fig. 6. Results in seconds of tests in implementations for NIKs + X (red / left entry) and NIKm + X (blue / right entry). Symbol $*$ means execution time > 150 seconds.

We used SWI Prolog 8.0.3 on a Dell XPS 13 9370 laptop running with a 1.80GHz Quad Core, Intel Core i7-8550U, 8 GB RAM, under Linux Mint 19.1. The numbers reported in the Figure 6 are the results given in seconds of the user `time` command from bash script, approximated at 0,05 seconds. Numbers in red (left side) are the results of the single-conclusion calculus, numbers in blue (right side) of the multi-conclusion one. We removed the LaTeX-output part from MOIN before proceeding to the tests.

The results vary depending on the formula under scope. In some cases (formula A tested in IK45 and IS5; formula C in IDB, ID5) the implementation of NIKm + X seems to have better performances than the one of NIKs + X, since the multi-conclusion prover has a smaller number of backtrack points (only the rules $\supset_m^\circ$ and $\Box_m^\circ$ are non-invertible). In some other cases (formula A tested in IDB; formula B tested in ID5 and ID45), the NIKm + X prover is slower than the NIKs + X one: this is due to the fact that in the multi-conclusion implementation the order in which the rules are applied is different from the single-conclusion one, and a larger number of sequents can be introduced in the proof search.

8   Conclusion and Future Work

The main purpose of MOIN is to be a tool for experimentation. The proof theory of intuitionistic modal logics has seen many advances in the last decade, and we felt the need for a tool that makes these advances accessible to a wider audience. We also hope to get through the use of MOIN some new insights in the decision problem for ID4, IK4 and IS4.

Further future work is to find more efficient implementations of the termination checks and to implement the focused systems of [5,6].

References

1. Arisaka, R., Das, A., Straßburger, L.: On nested sequents for constructive modal logics. Logical Methods in Computer Science **11**(3) (2015). https://doi.org/10.2168/LMCS-11(3:7)2015
2. Beckert, B., Posegga, J.: leantap: Lean tableau-based deduction. Journal of Automated Reasoning **15**(3), 339–358 (1995)
3. Brünnler, K.: Deep sequent systems for modal logic. Archive for Mathematical Logic **48**(6), 551–577 (2009)
4. Catach, L.: Tableaux: A general theorem prover for modal logics. Journal of automated reasoning **7**(4), 489–510 (1991)
5. Chaudhuri, K., Marin, S., Straßburger, L.: Focused and synthetic nested sequents. In: Jacobs, B., Löding, C. (eds.) Proceedings of the 19th International Conference on Foundations of Software Science and Computation Structures (FoSSaCS) (2016)
6. Chaudhuri, K., Marin, S., Straßburger, L.: Modular focused proof systems for intuitionistic modal logics. In: Kesner, D., Pientka, B. (eds.) 1st International Conference on Formal Structures for Computation and Deduction, FSCD 2016, June 22-26, 2016, Porto, Portugal. LIPIcs, vol. 52, pp. 16:1–16:18. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik (2016)
7. Fischer-Servi, G.: Axiomatizations for some intuitionistic modal logics. Rend. Sem. Mat. Univers. Politecn. Torino **42**(3), 179–194 (1984)
8. Galmiche, D., Salhi, Y.: Label-free proof systems for intuitionistic modal logic IS5. In: International Conference on Logic for Programming Artificial Intelligence and Reasoning. pp. 255–271. Springer (2010)
9. Gentzen, G.: Untersuchungen über das logische Schließen. I. Mathematische Zeitschrift **39**, 176–210 (1935)
10. Goré, R., Postniece, L., Tiu, A.: Cut-elimination and proof search for bi-intuitionistic tense logic. In: Advances in Modal Logic 2010. pp. 156–177 (2010)
11. Kashima, R.: Cut-free sequent calculi for some tense logics. Studia Logica **53**(1), 119–136 (1994)
12. Kuznets, R., Straßburger, L.: Maehara-style modal nested calculi. Archive for Mathematical Logic **58**(3-4), 359–385 (2019)
13. Lellmann, B.: Combining monotone and normal modal logic in nested sequents– with countermodels. In: International Conference on Automated Reasoning with Analytic Tableaux and Related Methods. pp. 203–220. Springer (2019)
14. Lellmann, B., Pimentel, E.: Modularisation of sequent calculi for normal and non-normal modalities. ACM Transactions on Computational Logic (TOCL) **20**(2), 7 (2019)
15. Maehara, S.: Eine darstellung der intuitionistischen logik in der klassischen. Nagoya mathematical journal **7**, 45–64 (1954)
16. Marin, S.: Modal proof theory through a focused telescope. Ph.D. thesis, Université Paris Saclay (2018)
17. Marin, S., Straßburger, L.: Label-free Modular Systems for Classical and Intuitionistic Modal Logics. In: Advances in Modal Logic 10. Groningen, Netherlands (Aug 2014)
18. Olivetti, N., Pozzato, G.L.: Nescond: an implementation of nested sequent calculi for conditional logics. In: S. Demri, D. Kapur, C.W. (ed.) International Joint Conference on Automated Reasoning. pp. 511–518. Springer (2014)
19. Otten, J.: How to build an automated theorem prover (2019), invited tutorial at TABLEAUX 2019 in London, available at http://www.jens-otten.de/tutorial_tableaux19/

20. Plotkin, G.D., Stirling, C.P.: A framework for intuitionistic modal logic. In: Halpern, J.Y. (ed.) Theoretical Aspects of Reasoning About Knowledge (1986)
21. Poggiolesi, F.: The method of tree-hypersequents for modal propositional logic. In: Makinson, D., Malinowski, J., Wansing, H. (eds.) Towards Mathematical Philosophy. Trends in Logic, vol. 28, pp. 31–51. Springer (2009)
22. Simpson, A.: The Proof Theory and Semantics of Intuitionistic Modal Logic. Ph.D. thesis, University of Edinburgh (1994)
23. Straßburger, L.: Cut elimination in nested sequents for intuitionistic modal logics. In: Pfenning, F. (ed.) FoSSaCS'13. LNCS, vol. 7794, pp. 209–224. Springer (2013)