

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

Marianna Girlando, Lutz Straßburger

► To cite this version:

Marianna Girlando, Lutz Straßburger. MOIN: A Nested Sequent Theorem Prover for Intuitionistic Modal Logics (System Description). 2020. hal-02457240

HAL Id: hal-02457240

<https://hal.inria.fr/hal-02457240>

Submitted on 27 Jan 2020

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.

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

Marianna Girlando and Lutz Straßburger

Inria, Equipe Partout, 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 counter model in case the proof search fails. To our knowledge this is the first automated theorem prover for intuitionistic modal logics. For wider usability of our system, we also implemented all 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 [20,14], and constructive [1]) and non-normal modal logics [12]. For some variants there also exist focused calculi [4,5].

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 [12,11] and normal conditional logics [15]. 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 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 MOIN¹ (for *MODal Intuitionistic Nested sequents*), follows the *lean deduction* methodology introduced in [2]: the program is constituted 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 mechanism². 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.

The biggest difficulty in using the “lean deduction” for nested sequents lies in 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 two approaches to overcome this difficulty: the implementation of conditional modal logics in [15] directly implements the nested sequent formalism as nested lists, and the implementations of non-normal modal logics in [12,11] uses annotations to represent nested sequents. We follow this second approach, as it provides a cleaner implementation, easier readability of the code, and higher 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 [6] and [17], and then been investigated in detail by Simpson in his PhD-thesis [19]. It is obtained from intuitionistic propositional logic, by adding the two modalities \Box and \Diamond , together with the necessitation rule (if A is provable, then so is $\Box A$), and the following five axioms:

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

In classical modal logic, only k_1 is added, as the others follow via the De Morgan duality between \Box and \Diamond , 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**, **4** and **5**, which in the classical as well as in the intuitionistic case generate 15 different logics, which can be arranged in the so-called **S5**-cube (classical) or **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 [3,7,20,14,4,5]. The Prolog code is written in a way that it can easily be extended to other logics.

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

² An introduction on Prolog theorem provers design is Jens Otten’s tutorial at TABLEAUX 2019 [16].

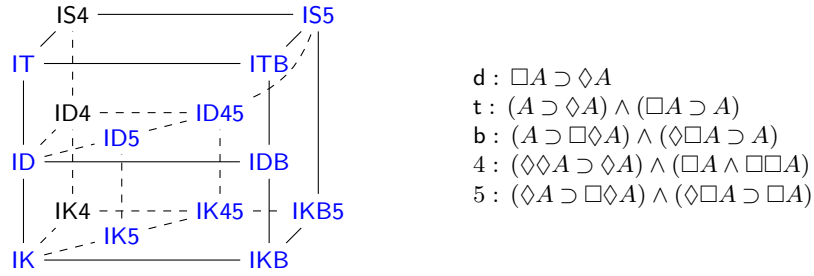


Fig. 1. The intuitionistic modal IS5-cube

3 Nested sequents

Nested sequents have been introduced independently in [9], [3], and [18]. 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

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

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

$$fm(\Gamma) = A_1 \vee \dots \vee A_n \vee \Box fm(\Gamma_1) \vee \dots \vee \Box fm(\Gamma_m) \quad .$$

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 literature³, 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

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

where A_1, \dots, A_n are the formulas that would occur on the left of the turnstile if there was a turnstile, and B_1, \dots, B_k are the formulas that would occur on the right of the turnstile if there was a turnstile, and $\Gamma_1, \dots, \Gamma_m$ 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

$$\Gamma ::= A, B^\circ \mid A, [\Gamma] \quad \quad \Lambda ::= \emptyset \mid A^\bullet, A \mid [A_1], A_2$$

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

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

$$\begin{aligned} fm(\emptyset) &= \perp & fm(A^\bullet, \Lambda) &= A \wedge fm(\Lambda) & fm([A_1], \Lambda_2) &= \Diamond fm(\Lambda_1) \wedge fm(\Lambda_2) \\ fm(\Lambda, B^\circ) &= fm(\Lambda) \supset B & fm(\Lambda, [\Gamma]) &= fm(\Lambda) \supset \Box fm(\Gamma) \end{aligned}$$

In order to 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 $\Gamma\{A\}$ (respectively $\Gamma\{\Delta\}$) for the sequent obtained from the context $\Gamma\{\cdot\}$ by replacing the hole $\{\cdot\}$ by the formula A (respectively the sequent Δ). 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 $(A \wedge B \wedge \Diamond(D \wedge E \wedge \Diamond F) \supset A)$.

4 The Proof Systems

There are two kinds of nested sequent systems for intuitionistic logic: single-conclusion [20,14,5] in the style of Gentzen [8] and multi-conclusion [10], in the style of Maehara [13]. In MOIN, we implemented both. More precisely, we implemented the single-conclusion systems of [20] and (a minor variation of) the multi-conclusion systems presented in [10].

Figure 2 shows the basic single-conclusion system NIKs for the modal logic IK as presented in [20]. In that figure, the formulas marked in blue 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. Blue formulas can never be principal in a rule application; they are kept in the premisses for book-keeping. Note the use of Γ^* in the \supset_s -rule, which 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 shown in Figure 1. They are taken from [20], and we refer the reader to [20] or [14] for a discussion on which set of rules gives which logic in the cube in Figure 1. For a set $X \subseteq \{\mathbf{d}, \mathbf{t}, \mathbf{5}, \mathbf{4}, \mathbf{5}\}$, we write NIKs + X for the system obtained from NIKs by adding the corresponding rules in Figure 3.

Finally, Figure 4 shows the rules needed for the multi-conclusion system. For each $X \subseteq \{\mathbf{d}, \mathbf{t}, \mathbf{5}, \mathbf{4}, \mathbf{5}\}$, 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. For more details the reader is referred to [10].

$$\begin{array}{c}
\frac{\perp^\bullet}{\Gamma\{\perp^\bullet\}} \quad \frac{\text{id}}{\Gamma\{a^\bullet, a^\circ\}} \\
\frac{\wedge^\bullet \frac{\Gamma\{A \wedge B^\bullet, A^\bullet, B^\bullet\}}{\Gamma\{A \wedge B^\bullet\}}}{\Gamma\{A \wedge B^\bullet\}} \quad \frac{\wedge^\circ \frac{\Gamma\{A \wedge B^\circ, A^\circ\} \quad \Gamma\{A \wedge B^\circ, B^\circ\}}{\Gamma\{A \wedge B^\circ\}}}{\Gamma\{A \wedge B^\circ\}} \\
\frac{\vee^\bullet \frac{\Gamma\{A \vee B^\bullet, A^\bullet\} \quad \Gamma\{A \vee B^\bullet, B^\bullet\}}{\Gamma\{A \vee B^\bullet\}}}{\Gamma\{A \vee B^\bullet\}} \quad \frac{\vee_{s1}^\circ \frac{\Gamma\{A^\circ\}}{\Gamma\{A \vee B^\circ\}} \quad \vee_{s2}^\circ \frac{\Gamma\{B^\circ\}}{\Gamma\{A \vee B^\circ\}}}{\Gamma\{A \vee B^\circ\}} \\
\frac{\supset_s^\bullet \frac{\Gamma\{A \supset B^\bullet, A^\circ\} \quad \Gamma\{A \supset B^\bullet, B^\bullet\}}{\Gamma\{A \supset B^\bullet\}}}{\Gamma\{A \supset B^\bullet\}} \quad \frac{\supset_s^\circ \frac{\Gamma\{A^\bullet, B^\circ\}}{\Gamma\{A \supset B^\circ\}}}{\Gamma\{A \supset B^\circ\}} \\
\frac{\Box^\bullet \frac{\Gamma\{\Box A^\bullet, [A^\bullet, \Delta]\}}{\Gamma\{\Box A^\bullet, [\Delta]\}} \quad \Diamond^\bullet \frac{\Gamma\{\Diamond A^\bullet, [A^\bullet]\}}{\Gamma\{\Diamond A^\bullet\}}}{\Gamma\{\Box A^\bullet, [\Delta]\}} \quad \frac{\Box_s^\circ \frac{\Gamma\{[A^\circ]\}}{\Gamma\{\Box A^\circ\}} \quad \Diamond^\circ \frac{\Gamma\{\Diamond A^\circ, [A^\circ, \Delta]\}}{\Gamma\{\Diamond A^\circ, [\Delta]\}}}{\Gamma\{\Box A^\circ, [\Delta]\}}
\end{array}$$

Fig. 2. System NIKs

$$\begin{array}{c}
d_s^\circ \frac{\Gamma\{\Diamond A^\circ, [A^\circ]\}}{\Gamma\{\Diamond A^\circ\}} \quad t_s^\circ \frac{\Gamma\{\Diamond A^\circ, A^\circ\}}{\Gamma\{\Diamond A^\circ\}} \quad b_s^\circ \frac{\Gamma\{[\Delta, \Diamond A^\circ], A^\circ\}}{\Gamma\{[\Delta, \Diamond A^\circ]\}} \quad 4_s^\circ \frac{\Gamma\{\Diamond A^\circ, [\Diamond A^\circ, \Delta]\}}{\Gamma\{\Diamond A^\circ, [\Delta]\}} \\
d^\bullet \frac{\Gamma\{\Box A^\bullet, [A^\bullet]\}}{\Gamma\{\Box A^\bullet\}} \quad t^\bullet \frac{\Gamma\{\Box A^\bullet, A^\bullet\}}{\Gamma\{\Box A^\bullet\}} \quad b^\bullet \frac{\Gamma\{[\Delta, \Box A^\bullet], A^\bullet\}}{\Gamma\{[\Delta, \Box A^\bullet]\}} \quad 4^\bullet \frac{\Gamma\{\Box A^\bullet, [\Box A^\bullet, \Delta]\}}{\Gamma\{\Box A^\bullet, [\Delta]\}} \\
5_{1s}^\circ \frac{\Gamma\{[\Delta, \Diamond A^\circ], \Diamond A^\circ\}}{\Gamma\{[\Delta, \Diamond A^\circ]\}} \quad 5_{2s}^\circ \frac{\Gamma\{[\Delta, \Diamond A^\circ], [\Diamond A^\circ, \Sigma]\}}{\Gamma\{[\Delta, \Diamond A^\circ], [\Sigma]\}} \quad 5_{3s}^\circ \frac{\Gamma\{[\Delta, \Diamond A^\circ, [\Diamond A^\circ, \Sigma]]\}}{\Gamma\{[\Delta, \Diamond A^\circ, [\Sigma]]\}} \\
5_1^\bullet \frac{\Gamma\{[\Delta, \Box A^\bullet], \Box A^\bullet\}}{\Gamma\{[\Delta, \Box A^\bullet]\}} \quad 5_2^\bullet \frac{\Gamma\{[\Delta, \Box A^\bullet], [\Box A^\bullet, \Sigma]\}}{\Gamma\{[\Delta, \Box A^\bullet], [\Sigma]\}} \quad 5_3^\bullet \frac{\Gamma\{[\Delta, \Box A^\bullet, [\Box A^\bullet, \Sigma]]\}}{\Gamma\{[\Delta, \Box A^\bullet, [\Sigma]]\}}
\end{array}$$

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

$$\begin{array}{c}
\supset_m^\bullet \frac{\Gamma\{A \supset B^\bullet, A^\circ\} \quad \Gamma\{A \supset B^\bullet, B^\bullet\}}{\Gamma\{A \supset B^\bullet\}} \quad \supset_m^\circ \frac{\Gamma^*\{A^\bullet, B^\circ\}}{\Gamma\{A \supset B^\circ\}} \quad \Box_m^\circ \frac{\Gamma^*\{[A^\circ]\}}{\Gamma\{\Box A^\circ\}} \quad \vee_m^\circ \frac{\Gamma\{A \vee B^\circ, A^\circ, B^\circ\}}{\Gamma\{A \vee B^\circ\}} \\
d_m^\circ \frac{\Gamma\{\Diamond A^\circ, [A^\circ]\}}{\Gamma\{\Diamond A^\circ\}} \quad t_m^\circ \frac{\Gamma\{\Diamond A^\circ, A^\circ\}}{\Gamma\{\Diamond A^\circ\}} \quad b_m^\circ \frac{\Gamma\{[\Delta, \Diamond A^\circ], A^\circ\}}{\Gamma\{[\Delta, \Diamond A^\circ]\}} \quad 4_m^\circ \frac{\Gamma\{\Diamond A^\circ, [\Diamond A^\circ, \Delta]\}}{\Gamma\{\Diamond A^\circ, [\Delta]\}} \\
5_{1m}^\circ \frac{\Gamma\{[\Delta, \Diamond A^\circ], \Diamond A^\circ\}}{\Gamma\{[\Delta, \Diamond A^\circ]\}} \quad 5_{2m}^\circ \frac{\Gamma\{[\Delta, \Diamond A^\circ], [\Diamond A^\circ, \Sigma]\}}{\Gamma\{[\Delta, \Diamond A^\circ], [\Sigma]\}} \quad 5_{3m}^\circ \frac{\Gamma\{[\Delta, \Diamond A^\circ, [\Diamond A^\circ, \Sigma]]\}}{\Gamma\{[\Delta, \Diamond A^\circ, [\Sigma]]\}}
\end{array}$$

Fig. 4. Rules for the multi-conclusion system NIKm + X

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 (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 [19], 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° -, 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 [7]. 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 [19] 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 [20]. 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 [10]. They are related to the intuitionistic (birelational) Kripke models [6,17], 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 [10] for obtaining a counter model from a failed proof search, with certain simplifications, as in our case the model is always finite.

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

6 The Prolog Implementation

The MOIN theorem prover⁴ is implemented following the well-known “lean deduction” methodology, introduced in [2]. The program is composed of a set clauses, each implementing a rule of the sequent calculus. The clauses are recursively applied to a given formula, constructing a proof-search tree. MOIN implements the nested sequents for classical modal logics from [3], and the single- and multi-conclusions calculi for intuitionistic modal logics shown in Figures 2,3, 4).

While ordinary sequent systems can be straightforwardly implemented using Prolog lists, some more complex data structure is needed to represent nested sequents, which are trees of lists of formulas. The NesCond theorem prover in [15] uses a tree of lists and allows rules to be applied at any level in the tree, and the prover of Lellmann [11] represents nested sequents as Prolog lists with *annotations*. This can be compared to labeled calculi [19]: an annotation is an index labelling formulas of a nested sequent, with the proviso that formulas

⁴ Available at www.lix.polytechnique.fr/Labo/Lutz.Strassburger/Software/Moin.

occurring at the same node of the nested sequent share the same annotation. The advantage of this approach is a more direct countermodel extraction in case of a failed proof search.

Propositional variables are represented in MOIN as Prolog atoms a, b, \dots ; \perp and \top are Prolog **false** and **true**, and the connectives \neg , \wedge , \vee , \supset , \Box and \Diamond are respectively represented by \sim , \wedge , \vee , \rightarrow , $!$ 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 **blue** 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 \bullet -formulas, and **Out** is a pair (X, F) storing the \circ -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 \bullet - and \circ -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], $(?a \rightarrow !?a) \wedge (!a \rightarrow !a)$) triggers the derivation of axiom 5 in $\text{NIKs} + \{b, 5\}$. 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 $\text{NIKm} + X$, where all the failed branches need to be considered.

7 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 [4,5].

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](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. 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)
5. 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)
6. Fischer-Servi, G.: Axiomatizations for some intuitionistic modal logics. *Rend. Sem. Mat. Univ. Politecn. Torino* **42**(3), 179–194 (1984)
7. 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)
8. Gentzen, G.: Untersuchungen über das logische Schließen. I. *Mathematische Zeitschrift* **39**, 176–210 (1935)
9. Kashima, R.: Cut-free sequent calculi for some tense logics. *Studia Logica* **53**(1), 119–136 (1994)
10. Kuznets, R., Straßburger, L.: Maehara-style modal nested calculi. *Archive for Mathematical Logic* **58**(3-4), 359–385 (2019)
11. 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)
12. Lellmann, B., Pimentel, E.: Modularisation of sequent calculi for normal and non-normal modalities. *ACM Transactions on Computational Logic (TOCL)* **20**(2), 7 (2019)
13. Maehara, S.: Eine darstellung der intuitionistischen logik in der klassischen. *Nagoya mathematical journal* **7**, 45–64 (1954)
14. Marin, S., Straßburger, L.: Label-free Modular Systems for Classical and Intuitionistic Modal Logics. In: *Advances in Modal Logic 10*. Groningen, Netherlands (Aug 2014)
15. 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)
16. 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/>
17. Plotkin, G.D., Stirling, C.P.: A framework for intuitionistic modal logic. In: Halpern, J.Y. (ed.) *Theoretical Aspects of Reasoning About Knowledge* (1986)
18. 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)

19. Simpson, A.: The Proof Theory and Semantics of Intuitionistic Modal Logic. Ph.D. thesis, University of Edinburgh (1994)
20. 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)