Some Proof Translations for Intuitionistic Modal Logic

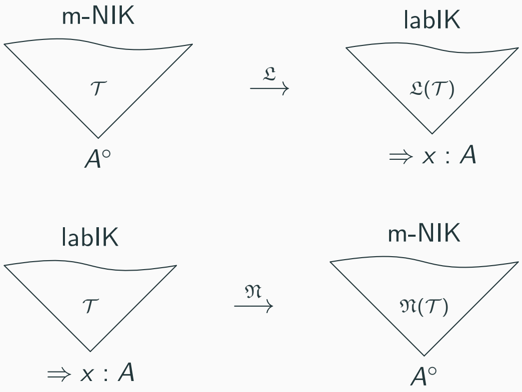
Justus Becker and Marianna Girlando

6 September 2024

Institute for Logic Language and Computation, University of Amsterdam

The Idea...

... translate derivations from one calculus into another one.



The Intuitionistic Modal Logic IK

Language of IK

We define the language $\mathcal{L}^{\Box\Diamond}$ by a countably infinite set of propositional atoms $\Phi = \{p, q, r, ...\}$ and some independent logical connectives and modalities.

$$A, B ::= \bot \mid p \mid (A \wedge B) \mid (A \vee B) \mid (A \supset B) \mid \Box A \mid \Diamond A$$

We further include the following abbreviations: $\neg A := (A \supset \bot)$ and $\top := (\bot \supset \bot)$.

Axiomatisation of IK

Introduced as a "good" intuitionistic modal logic by Fischer Servi 1984 and Simpson 1994.

Axiom schemas:

Intuitionistic propositional tautologies

$k_1 : \Box(A \supset B) \supset (\Box A \supset \Box B)$

$k_2 : \Box(A \supset B) \supset (\Diamond A \supset \Diamond B)$

$k_3 : \Diamond(A \vee B) \supset (\Diamond A \vee \Diamond B)$

$k_4 : (\Diamond A \supset \Box B) \supset \Box(A \supset B)$

$k_5 : \neg\Diamond\bot$

Rules:

(mp) from $A \supset B$ and A infer B

(nec) from A infer $\Box A$

Semantics for IK

A birelational model is a tuple $\langle W, \leq, R, V\rangle$ with a non-empty set W, a reflexive and transitive relation $\leq \subseteq W \times W$, a relation $R \subseteq W \times W$ and a valuation function $V : \Phi \to \mathcal{P}(W)$. It further satisfies the condition of monotonicity and forward/backward confluence.

$$
\begin{tikzcd}
z' \arrow[r, dotted, "R"] & z \\
x \arrow[u, dotted, "\leq"] \arrow[r, "R"'] & y \arrow[u, "\leq"']
\end{tikzcd}
\qquad
\begin{tikzcd}
z \arrow[r, dotted, "R"] & z' \\
x \arrow[u, "\leq"] \arrow[r, "R"'] & y \arrow[u, dotted, "\leq"']
\end{tikzcd}
$$

Truth is defined in the "usual" ways for modal and intuitionistic logic, except for:

$\mathcal{M}, w \Vdash \Box A$ iff for all $v, u \in W$: if $w \leq v$ and vRu then $\mathcal{M}, u \Vdash A$

Proof Theory of IK

Extending Sequents

We use derivations of sequents $\Gamma \Rightarrow \Delta$, as it is common practice in structural proof theory.

There are mainly two approaches when it comes to defining sequent calculi for modal logics; extending the *structure* and extending the *language* in sequents.

Extending Sequents

We use derivations of sequents $\Gamma \Rightarrow \Delta$, as it is common practice in structural proof theory.

There are mainly two approaches when it comes to defining sequent calculi for modal logics; extending the *structure* and extending the *language* in sequents.

Nested sequents: $\Gamma, [\Delta_1], ..., [\Delta_n]$
with a standard sequent Γ and potentially nested sequents $\Delta_1, ..., \Delta_n$.

Labelled sequents: $\mathcal{R}, \Gamma \Rightarrow \Delta$
uses labelled formulas $x : A$ and xRy, where $\mathcal{R}$ contains only formulas of the form xRy and Γ, Δ only $x : A$.

A Fully Labelled Calculus

Marin, Morales, and Straßburger 2021 introduced an extension of classical sequent calculus that internalises the full semantics of IK. Some of the rules of labIK are as follows.

$$\text{Ax}\ \frac{}{\mathcal{R}, x \leq y, x : p, \Gamma \Rightarrow \Delta, y : p}$$

$$\mathit{Trans}\ \frac{\mathcal{R}, x \leq y, y \leq z, x \leq z, \Gamma \Rightarrow \Delta}{\mathcal{R}, x \leq y, y \leq z, \Gamma \Rightarrow \Delta}$$

$$\Diamond\text{R}\ \frac{\mathcal{R}, xRy, \Gamma \Rightarrow \Delta, y : A}{\mathcal{R}, xRy, \Gamma \Rightarrow \Delta, x : \Diamond A}$$

$$\Box\text{R}\ \frac{\mathcal{R}, x \leq y, yRz, \Gamma \Rightarrow \Delta, z : A}{\mathcal{R}, \Gamma \Rightarrow \Delta, x : \Box A}\ (y, z \text{ fresh})$$

A Fully Labelled Calculus

The system labIK admits the the structural rules of weakening (wk), label substitution (sub_V), monotonicity (mon_L, mon_R), contraction (c^R, $c^{\leq}$) as well as the cut rule.

It is special as a calculus for IK as it is fully invertible (no information gets lost in proof search), but it also carries a lot of information.

A Maehara-Style Nested Calculus

Kuznets and Straßburger 2019 introduced a nested extension of the intuitionistic Maehara calculus. We work here with a "contraction variant" of this calculus. Some of the rules of m-NIK are as follows.

$$\text{Ax}\ \frac{}{\Gamma\{p^\bullet, p^\circ\}} \qquad \Diamond^\circ\ \frac{\Gamma\{\Diamond A^\circ, [A^\circ, \Delta]\}}{\Gamma\{\Diamond A^\circ, [\Delta]\}} \qquad \Box^\circ\ \frac{\Gamma^\downarrow\{[A^\circ]\}}{\Gamma\{\Box A^\circ\}}$$

The system m-NIK admits the usual rules of weakening, contraction, as well as cut.

Unlike labIK, it is not fully invertible as potentially necessary information can get lost ($\supset^\circ$, $\Box^\circ$). At the same time, proofs in m-NIK carry much less information.

Proof Translations

Previous Results

Goré and Ramanayake 2014 introduced translations between (simple) tree-labelled and (simple) nested sequents. These formalisms are thus notational variants of each other.

$\Box\neg p^\bullet, [[\Diamond p^\circ], p \vee q^\circ]$ $\qquad$ $xRy, yRz, x : \Box\neg p \Rightarrow y : p \vee q, z : \Diamond p$

$$\begin{array}{c} \Box\neg p^\bullet \\ \downarrow \\ p \vee q^\circ \\ \downarrow \\ \Diamond p^\circ \end{array} \qquad\qquad \begin{array}{c} x : \Box\neg p \Rightarrow \\ \downarrow R \\ \Rightarrow y : p \vee q \\ \downarrow R \\ \Rightarrow z : \Diamond p \end{array}$$

This also allowed them to find effective translation between derivations and also compare systems of different formulations.

Translating from m-NIK to labIK

The main idea is to translate each rule of m-NIK separately into a derivation in labIK (including potentially some admissible rules).

For example:

$$\vee^\circ \ \frac{\Box p^\circ, [p^\circ, q^\circ]}{\Box p^\circ, [p \vee q^\circ]} \quad \rightsquigarrow \quad \frac{xRy \Rightarrow x : \Box p, y : p, y : q}{xRy \Rightarrow x : \Box p, y : p \vee q} \ \vee\mathrm{R}$$

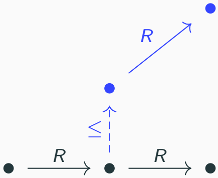
$$\begin{tikzcd}
\Box p^\circ \arrow[rr] & {} \arrow[d, dotted] & p^\circ, q^\circ \\
\Box p^\circ \arrow[rr] & {} & p \vee q^\circ
\end{tikzcd}$$

Translating from m-NIK to labIK

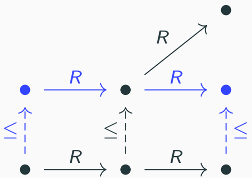
But what about rules that introduce $\leq$-formulas?

Answer: **Lift layers!**

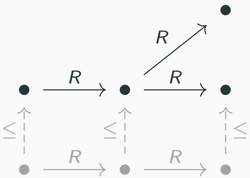
$$\Box\mathrm{R}\ \frac{\mathcal{R}, x \leq y, yRz, \Gamma \Rightarrow \Delta, z : A}{\mathcal{R}, \Gamma \Rightarrow \Delta, x : \Box A}\ (y, z \text{ fresh})$$



$\mathrm{F}_1\ \mathrm{F}_2\ \mathrm{mon}_L$ $\rightsquigarrow$



wk $\rightsquigarrow$



Translating from m-NIK to labIK

Theorem 4.12

For any sequent Γ, if m-NIK $\vdash \Gamma$ then labIK $\vdash \mathfrak{L}^x(\Gamma)$. Furthermore, the translation is effective in that we can constructively translate the full derivation into an admissible derivation for labIK using only rules of labIK$^+$.

Translating from labIK to m-NIK

Not all prooftrees from labIK are translatable into m-NIK because labIK is fully invertible and m-NIK is not.

Before translating we have to ensure that the labIK derivations have the correct form: All sequents in the prooftree must be *linearly layered* (no branching in $\leq$).

This allows one to always find a maximum layer (wrt. $\leq$), which we then can translate into a simple nested sequent.

There are generally two ways to change the prooftrees of labIK such that they become "linear": Edit them *directly* or *rebuild* a new derivation under a certain procedure. We do the latter as it is easier.

Searching for Linear Prooftrees

Primitive Proof Search

0. Start with a derivable sequent $\Rightarrow x : A$.
1. *Relationally saturate* the leaves of $\mathcal{T}_i$.
2. If all leaves of $\mathcal{T}_i$ are initial sequents, terminate.
   $\rightarrow$ A proof of $\Rightarrow x : A$ is obtained.
3. *Apply any non-redundant rule* ($\supset$R or $\Box$R) to some open $\mathfrak{S}'$, and go back to step 1 ($i \mapsto i + 1$).

Note: We assume an already derivable formula for our proof search, for actually defining a proper *decision procedure* one also has to incorporate loop checks.

Lemma 4.24 (single succession)

Let $\mathcal{R}, \Gamma \Rightarrow \Delta$ be a relationally saturated sequent appearing in the proof search described in figure 4.3.2, then $\mathsf{labIK} \vdash \mathcal{R}, \Gamma \Rightarrow \Delta$ iff $\mathsf{labIK} \vdash \mathcal{R}, \Gamma \Rightarrow x : C$ for some $x : C \in \Delta$. We call $x : C$ the single succedent of the sequent.

Searching for Linear Prooftrees

Linear Search Algorithm

0. Start with a derivable sequent $\Rightarrow x : A$.
1. *Saturate* the leaves of $\mathcal{T}_i$.
2. If all leaves of $\mathcal{T}_i$ are initial sequents, terminate.
   $\rightarrow$ A linear proof of $\Rightarrow x : A$ is obtained.
3. Otherwise, pick a non-axiomatic leaf sequent $\mathfrak{S}'$ in $\mathcal{T}_i$.
   (a) Compute the *lifting* $\mathfrak{S} \otimes \mathfrak{S}{\uparrow}^{x:F}$ (if possible) and go back to Step 1 ($i \mapsto i+1$).
   (b) Otherwise, *backtrack*.

Translating from labIK to m-NIK

Labelled sequents occurring in the algorithm might bare some structure like this

$$
\begin{tikzcd}
\bullet \arrow[r, "R"] & \bullet \arrow[r, "R"] & \bullet \\
\bullet \arrow[u, dashed, "\leq"'] \arrow[r, "R"'] & \bullet \arrow[u, dashed, "\leq"'] \arrow[r, "R"'] & \bullet \arrow[u, dashed, "\leq"'] \\
\bullet \arrow[u, dashed, "\leq"'] & &
\end{tikzcd}
$$

and will be translated into a nested sequent by only considering the top layer.

Translating from m-NIK to labIK

Theorem 4.40
For any formula $A \in \mathcal{L}^{\Box\Diamond}$, if labIK $\vdash \Rightarrow x : A$ then a derivation tree for m-NIK $\vdash A^\circ$ can be effectively obtained.

Corollary 4.41
For any formula $A \in \mathcal{L}^{\Box\Diamond}$: m-NIK $\vdash A$ iff labIK $\vdash A$. Also, for any nested sequent Γ: m-NIK $\vdash \Gamma$ iff labIK $\vdash \mathfrak{L}^x(\Gamma)$.

Corollary 4.42
For any formula $A \in \mathcal{L}$: If labG3I $\vdash A$ then a derivation tree of m-G3i $\vdash A$ can be effectively obtained. If m-G3i $\vdash A$ then a derivation tree of labG3I $\vdash A$ can be effectively obtained.

Conclusion

Conclusion

Summary

First we looked at the logic IK and some of its proof theory.

An effective translation from m-NIK to labIK was obtain by reformulating the tree-structure of sequents and implementing the *lift* rule.
The reverse translation relies on a separate search algorithm that emulates the behaviour of m-NIK such that it can properly reduce the information of labIK sequents.
This result establishes a direct completeness between these systems, as well as their modal-free counterparts m-G3i and G3I.

Conclusion

Future Works

Main point of interest will be translating systems for more IMLs, such as (modal and intermediate) extensions of IK or the logic FK.
Another direction would be to connect these systems to other calculi for IK.
As a more applied investigation, one might implement the translation and some form of the search algorithm.

Q&A

Thank you!

References

Fischer Servi, Gisèle (1984). **"Axiomatisations for some intuitionistic modal logics".** In: *Rendiconti del Seminario Matematico - PoliTO*. Vol. 42. 3, pp. 179–194.

Goré, Rajeev and Revantha Ramanayake (Jan. 2014). **"Labelled tree sequents, tree hypersequents and nested (Deep) sequents".** In: *Advances in Modal Logic* 9.

Kuznets, Roman and Lutz Straßburger (May 2019). **"Maehara-style modal nested calculi".** In: *Archive for Mathematical Logic* 58, pp. 359–385.

Marin, Sonia, Marianela Morales, and Lutz Straßburger (2021). **"A fully labelled proof system for intuitionistic modal logics".** In: *Journal of Logic and Computation* 31.3, pp. 998–1022.

Simpson, Alex K. (1994). **"The Proof Theory and Semantics of Intuitionistic Modal Logic".** PhD thesis. University of Edinburgh.

Example of two Frames

$$
\begin{tikzcd}
z \arrow[loop above, dashed, "\leq"] \arrow[dr, "R"] & \\
x \arrow[u, dashed, "\leq"'] \arrow[r, "R"'] \arrow[loop below, dashed, "\leq"] & y \arrow[loop below, dashed, "\leq"]
\end{tikzcd}
\qquad\qquad
\begin{tikzcd}
 & w \arrow[loop above, dashed, "\leq"] \\
u' \arrow[loop left, dashed, "\leq"] \arrow[ur, "R"] \arrow[r, "R"'] & v' \arrow[loop right, dashed, "\leq"] \\
u \arrow[u, dashed, "\leq"'] \arrow[r, "R"'] \arrow[loop below, dashed, "\leq"] & v \arrow[u, dashed, "\leq"'] \arrow[loop below, dashed, "\leq"]
\end{tikzcd}
$$