An Unfinished Story: Decidability of Intuitionistic S4

Marianna Girlando, Roman Kuznets, Sonia Marin, Lutz Straßburger

First introduced by C. I. Lewis as the fourth modal system, classical S4 is a well-known modal logic, whose most commonly used axiomatization is due to Gödel. The possible-world semantics for S4 is the class of models built on reflexive and transitive frames. The logic has been extensively studied, both from a model-theoretic and a proof-theoretic viewpoint, and enjoys several desirable properties — most notably decidability, which was proved in [5].

Intuitionistic modal logics result from adding modalities to intuitionistic, instead of classical, propositional logic. Different choices can be made when defining such systems and, as a result, different versions of intuitionistic modal logics have been defined in the literature. We here follow the approach from Fischer Servi [3] and Plotkin and Stirling [7]. For a comprehensive overview, refer to Simpson [8].

While decidability of most intuitionistic modal logics has been established [8], logic IS4, i.e., intuitionistic S4, represents a notable exception: its decidability has been an open question since its formulation in 1994, in Simpson's PhD thesis [8]. We thought to have settled the issue: our LICS'23 paper [4] claimed that IS4 was decidable, and provided a proof-theoretic algorithm to check decidability of formulas. However, some six months after publication, Agi Kurucz designed a clever yet simple IS4 formula that could not be decided by our algorithm. In this talk, we set the record straight by discussing the main ideas behind our LICS'23 paper, exploring Kurucz's counterexample, and discussing ongoing work on solutions to the problem of showing decidability of IS4.

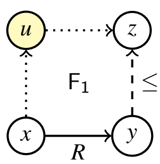
Given a set $\mathscr{A}$ of propositional variables, ***logic*** IS4 is formulated in the language $A ::= \bot \mid a \mid (A \wedge A) \mid (A \vee A) \mid (A \supset A) \mid \Box A \mid \Diamond A$, for $a \in \mathscr{A}$. Differently from what happens in the classical case, the modalities $\Box$ and $\Diamond$ are independent. An axiom system for IS4 is obtained by extending any axiom system for IPL with the standard necessitation rule and the axioms:

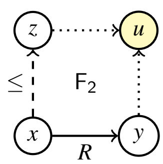
k_1:	$\Box(A \supset B) \supset (\Box A \supset \Box B)$	k_2:	$\Box(A \supset B) \supset (\Diamond A \supset \Diamond B)$		
k_3:	$\Diamond(A \vee B) \supset (\Diamond A \vee \Diamond B)$	k_4:	$(\Diamond A \supset \Box B) \supset \Box(A \supset B)$	k_5:	$\Diamond\bot \supset \bot$
4:	$(\Diamond\Diamond A \supset \Diamond A) \wedge (\Box A \supset \Box\Box A)$	t:	$(A \supset \Diamond A) \wedge (\Box A \supset A)$		

The semantics for IS4 is defined in terms of birelational models. Note that, as in its classical counterpart, the accessibility relation of IS4 is reflexive and transitive.

An IS4-***birelational model*** $\mathfrak{M}$ is a quadruple $\langle W, R, \leq, V \rangle$ of a set $W \neq \varnothing$ of ***worlds*** equipped with two preorders (i.e., reflexive and transitive relations) — an ***accessibility relation*** R and future relation $\leq$ — and a ***valuation function*** $V : W \to 2^{\mathscr{A}}$ satisfying:

(F_1) For all $x, y, z \in W$, if xRy and $y \leq z$, there exists $u \in W$ s.t. $x \leq u$ and uRz.
(F_2) For all $x, y, z \in W$, if $x \leq z$ and xRy, there exists $u \in W$ s.t. zRu and $y \leq u$.
(Mon) If $w \leq w'$, then $V(w) \subseteq V(w')$.





Forcing for atomic formulas is determined as $\mathfrak{M}, w \Vdash a$ iff $a \in V(w)$, with $\mathfrak{M}, w \nVdash \bot$. The relation is then recursively extended to all formulas (clauses for $\wedge$ and $\vee$ are standard):

$\mathfrak{M}, w \Vdash A \supset B$	iff	for all w' with $w \leq w'$, if $\mathfrak{M}, w' \Vdash A$, then $\mathfrak{M}, w' \Vdash B$;
$\mathfrak{M}, w \Vdash \Box A$	iff	for all w' and u with $w \leq w'$ and $w'Ru$, we have $\mathfrak{M}, u \Vdash A$;
$\mathfrak{M}, w \Vdash \Diamond A$	iff	there exists u such that wRu and $\mathfrak{M}, u \Vdash A$.

The forcing relation is easily shown to be ***monotone***: if $w \leq z$ and $\mathfrak{M}, w \Vdash A$, then $\mathfrak{M}, z \Vdash A$.

Theorem ([3, 7])**.** Formula A is a theorem of IS4 iff A is valid in every IS4-birelational model.

The decision algorithm for IS4 we presented in [4] performs proof search in the fully labelled sequent calculus for the logic introduced in [6]. Every semantic component is explicitly represented in the calculus: in particular, worlds of a birelational models are represented by ***labels*** $(x, y, z, \ldots)$, with the forcing relation represented by a labelled formula $x{:}A$ and the structure of the model represented by relational atoms xRy and $x \leq y$. Here follow some labelled rules, where 'y fresh' means that y does not occur in the conclusion of the rule:

$$\supset_{\mathsf{R}} \frac{x \leq y, y{:}A, \Gamma \Rightarrow \Delta, y{:}B}{\Gamma \Rightarrow \Delta, x{:}A \supset B}\ y \text{ fresh} \qquad \Diamond_{\mathsf{L}} \frac{xRy, y{:}A, \Gamma \Rightarrow \Delta}{x{:}\Diamond A, \Gamma \Rightarrow \Delta}\ y \text{ fresh} \qquad \mathsf{mon} \frac{x \leq z, z{:}A, x{:}A, \Gamma \Rightarrow \Delta}{x \leq z, x{:}A, \Gamma \Rightarrow \Delta}$$

Usually, to obtain a decision procedure for a logic using a proof system one implements a ***terminating*** strategy to perform root-first proof search. If the procedure finds a proof for a given formula, then obviously the formula is valid in the logic. Otherwise, a failed proof-search procedure provides enough information to construct a countermodel for the formula. To ensure termination of proof search, external loop-checks often need to be added to detect repetitions. Termination for several calculi for both IPL and S4 can be ensured in this way [2, 1].

The unique challenges of IS4 are due to the fact that there are two sources of repetition, i.e., transitivity of the $\leq$-relation and the R-relation. Their interaction creates a possibly infinite proof search, where repetition of full sequents is never encountered. To overcome this problem, in [4] we incorporated several loop-checks within the rules of the labelled calculus. That is, during proof search we are allowed to apply rules (called ***loop-rules***) which identify components of the sequents which are "the same," provided that some specific side conditions are met. Then, if the algorithm terminates without finding a proof, it is immediate how to construct a countermodel for the formula at the root. However, if a proof for the formula is found, we instead need to do some more work, to "remove" all the loop-rules from the derivation. This process is called ***unfolding***, and it consist of repeating specific parts of the proof until a "copy" of the component of the sequent identified by the loop-rule is created.

However, in [4] we did not consider all the sources of non-termination that could occur in root-first proof search. In particular, we did not consider all possible interactions of the two rules $\supset_{\mathsf{R}}$ and $\Diamond_{\mathsf{L}}$ depicted above. In combination with the other rules of the calculus, these rules can make a sequent "grow" (bottom-up) in two different ways. Rule $\supset_{\mathsf{R}}$ introduces a new $\leq$-successor y of a label x. Only the formulas occurring in the antecedent and labelled with x can transfer to y, via the mon rule — we therefore lose information when moving from x to y. Rule $\Diamond_{\mathsf{L}}$ instead introduces a new R-successor of a label. The interaction of $\supset_{\mathsf{R}}$ and $\Diamond_{\mathsf{L}}$ is at the heart of Kurucz's formula:

$$\Big(\Box\big(a \supset (\Diamond b \wedge \Diamond c)\big) \wedge \Box\big((d \supset \bot) \supset \bot\big) \wedge \Box\big((e \supset \bot) \supset \bot\big)\Big) \supset \bot$$

On this formula, proof search as implemented by the algorithm from [4] runs forever (refer to the Appendix for more details).

Despite this setback, we still think that logic IS4 is decidable, and in our talk we will present a more general algorithm that is also based on fully labelled calculus and that can safely cover the looping behavior exemplified by Kurucz's formula.

Acknowledgments. The authors are deeply grateful to Agi Kurucz for finding the counterexample.

References

[1] Kai Brünnler. Deep sequent systems for modal logic. *Archive for Mathematical Logic*, 48(6):551–577, July 2009. doi:10.1007/s00153-009-0137-3.

[2] Roy Dyckhoff. Intuitionistic decision procedures since Gentzen. In Reinhard Kahle, Thomas Strahm, and Thomas Studer, editors, *Advances in Proof Theory*, volume 28 of *Progress in Computer Science and Applied Logic*, pages 245–267. Birkhäuser, 2016. doi:10.1007/978-3-319-29198-7_6.

[3] Gisèle Fischer Servi. Axiomatizations for some intuitionistic modal logics. *Rendiconti del Seminario Matematico, Università e Politecnico di Torino*, 42(3):179–194, 1984.

[4] Marianna Girlando, Roman Kuznets, Sonia Marin, Marianela Morales, and Lutz Straßburger. 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. doi:10.1109/LICS56636.2023.10175684.

[5] Richard E. Ladner. The computational complexity of provability in systems of modal propositional logic. *SIAM Journal on Computing*, 6(3):467–480, September 1977. doi:10.1137/0206033.

[6] Sonia Marin, Marianela Morales, and Lutz Straßburger. A fully labelled proof system for intuitionistic modal logics. *Journal of Logic and Computation*, 31(3):998–1022, April 2021. doi:10.1093/logcom/exab020.

[7] Gordon Plotkin and Colin Stirling. A framework for intuitionistic modal logics. In Joseph Y. Halpern, editor, *Theoretical Aspects of Reasoning About Knowledge: Proceedings of the 1986 Conference*, pages 399–406. Morgan Kaufmann, 1986. URL: http://tark.org/proceedings/tark_mar19_86/p399-plotkin.pdf.

[8] Alex K. Simpson. *The Proof Theory and Semantics of Intuitionistic Modal Logic*. PhD thesis, University of Edinburgh, Edinburgh, Scotland, UK, 1994. URL: http://hdl.handle.net/1842/407.

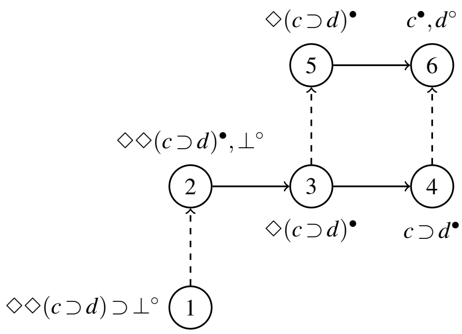
Appendix

Here we describe the infinite proof search branch generated by the algorithm in [4] on Kurucz's formula. For full details about the proof system and the algorithm, we refer the reader to [4].

A ***labelled sequent*** is a pair $\Gamma \Longrightarrow \Delta$, where Γ is a set of relational atoms and labelled formulas and Δ is a set of labelled formulas. As in [4], we represent labelled sequents by means of a directed graph whose nodes are the labels of the sequent (for convenience we shall use natural numbers), dashed edges are the $\leq$-relations, and solid edges are the R-relations. We write a formula $A^{\bullet}$ (resp. A°) next to a label w to indicate that the labelled formula $w{:}A$ occurs on the left-hand side (resp. the right-hand side) of the sequent arrow $\Longrightarrow$.

A labelled sequent and its graphical representation are reported below, where we take $\mathscr{R} = 1{\leq}2,\ 2R2,\ 2R3,\ 3{\leq}3,\ 3{\leq}5,\ 3R4,\ 4{\leq}6,\ 5R6$ (reflexive R- and $\leq$-loops are omitted).

$$\mathscr{R}, 2{:}\Diamond\Diamond(c \supset d), 3{:}\Diamond(c \supset d), 4{:}c \supset d, 5{:}\Diamond(c \supset d), 6{:}c \Longrightarrow 1{:}\Diamond\Diamond(c \supset d) \supset \bot, 2{:}\bot, 6{:}d$$



Below are the logical and main structural rules of the labelled calculus needed to construct the proof search tree for Kurucz's formula:

$$\supset_{\mathsf{R}}\ \frac{x{\leq}y, y{:}A, \Gamma \Longrightarrow \Delta, y{:}B}{\Gamma \Longrightarrow \Delta, x{:}A \supset B}\ y \text{ fresh} \qquad \Diamond_{\mathsf{L}}\ \frac{xRy, y{:}A, \Gamma \Longrightarrow \Delta}{x{:}\Diamond A, \Gamma \Longrightarrow \Delta}\ y \text{ fresh} \qquad \mathsf{mon}\ \frac{x{\leq}z, z{:}A, x{:}A, \Gamma \Longrightarrow \Delta}{x{\leq}z, x{:}A, \Gamma \Longrightarrow \Delta}$$

$$\supset_{\mathsf{L}}\ \frac{x{\leq}y, x{:}A \supset B, \Gamma \Longrightarrow \Delta, y{:}A \qquad x{\leq}y, \Gamma, y{:}B \Longrightarrow \Delta}{x{\leq}y, \Gamma, x{:}A \supset B \Longrightarrow \Delta} \qquad \Box_{\mathsf{L}}\ \frac{x{\leq}y, yRz, \Gamma, z{:}A, x{:}\Box A \Longrightarrow \Delta}{x{\leq}y, yRz, \Gamma, x{:}\Box A \Longrightarrow \Delta} \qquad \wedge_{\mathsf{L}}\ \frac{\Gamma, x{:}A, x{:}B \Longrightarrow \Delta}{\Gamma, x{:}A \wedge B \Longrightarrow \Delta}$$

$$\mathsf{F}_1\ \frac{xRy, y{\leq}z, x{\leq}u, uRz, \Gamma \Longrightarrow \Delta}{xRy, y{\leq}z, \Gamma \Longrightarrow \Delta}\ u \text{ fresh} \qquad \mathsf{F}_2\ \frac{xRy, x{\leq}z, y{\leq}u, zRu, \Gamma \Longrightarrow \Delta}{xRy, x{\leq}z, \Gamma \Longrightarrow \Delta}\ u \text{ fresh}$$

We now take F to be Kurucz's formula and G to be the antecedent of the formula, that is:

$$F = \overbrace{\Big(\Box\big(a \supset (\Diamond b \wedge \Diamond c)\big) \wedge \Box\big((d \supset \bot) \supset \bot\big) \wedge \Box\big((e \supset \bot) \supset \bot\big)\Big)}^{G} \supset \bot$$

Figure 1 depicts the evolution of a sequent, which grows as we move (bottom-up) along a branch of a proof search tree having sequent $0{\leq}0 \Longrightarrow 0{:}F$ at the root. So, the first rule applied to the sequent is $\supset_{\mathsf{R}}$, which generates a new label 1, and the sequent $0{\leq}0, 0{\leq}1, 1{:}G \Longrightarrow 1{:}\bot$. Then, more formulas get labelled with 1, thanks to applications of $\wedge_{\mathsf{L}}$, and so on.

Colors identify labels / nodes which are "the same," meaning that they label the same formulas both in the antecedent and in the consequent of the sequent. At a first glance, our sequent contains only finitely many colors, that are nevertheless combined in a peculiar way, making it difficult to find exact repetitions between sets of nodes.

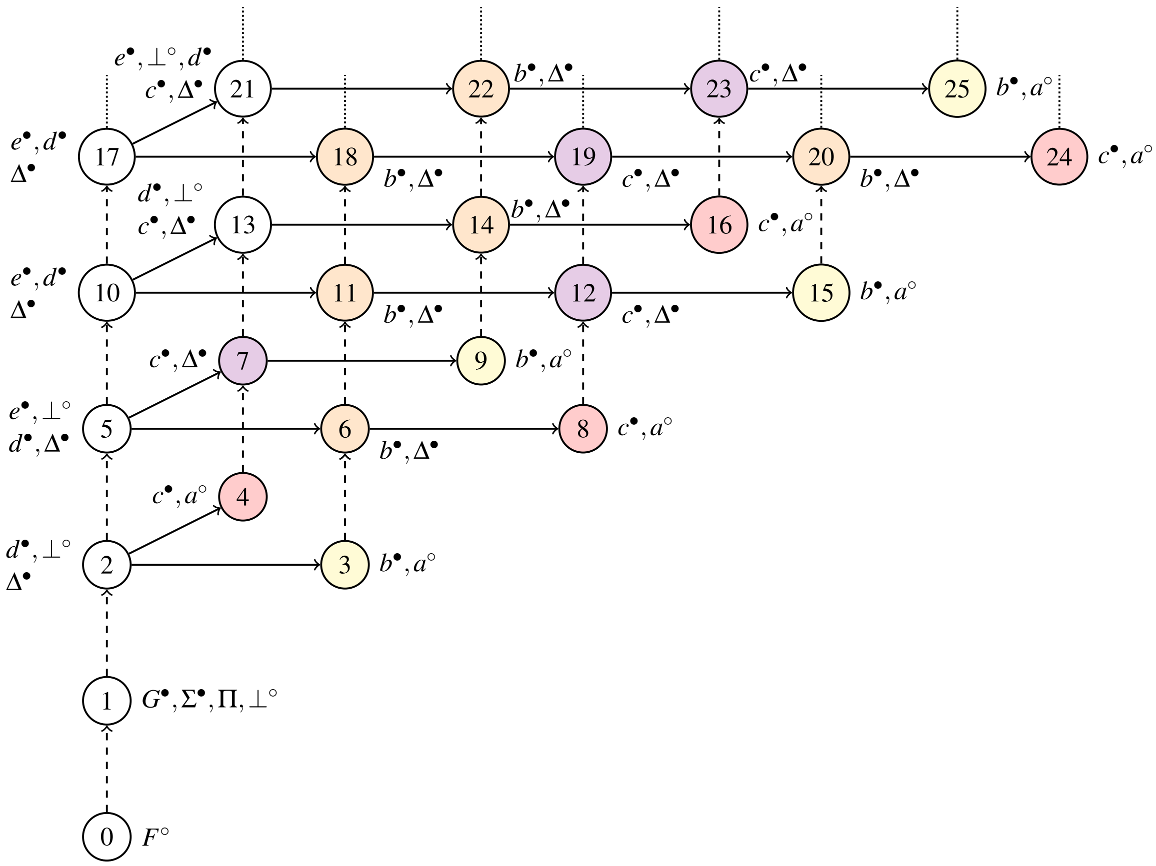


Figure 1: Proof search for Kurucz's formula. Same color means same formulas.

In Figure 1 we use the following shorthands:

$$
\begin{aligned}
\Delta^\bullet &= \{\Diamond b \wedge \Diamond c^\bullet,\ \Diamond b^\bullet,\ \Diamond c^\bullet\} \\
\Sigma^\bullet &= \left\{\Box\big(a \supset (\Diamond b \wedge \Diamond c)\big)^\bullet, \Box\big((d \supset \bot) \supset \bot\big)^\bullet, \Box\big((e \supset \bot) \supset \bot\big)^\bullet\right\} \\
\Pi &= \{a \supset (\Diamond b \wedge \Diamond c)^\bullet, (d \supset \bot) \supset \bot^\bullet, (e \supset \bot) \supset \bot^\bullet, d \supset \bot^\circ, e \supset \bot^\circ\}
\end{aligned}
$$

Moreover, we take the formulas in Π to be in all the nodes (i.e., labelled with all the labels) greater than or equal to 1. The formulas in $\Delta^\bullet$ are in all nodes greater than or equal to 2 that are not R-leaves. In particular, observe that formulas $d \supset \bot^\circ$ and $e \supset \bot^\circ$ are in all the nodes greater than or equal to 1. However, in the graph we do not show the result of applying $\supset_{\mathsf{R}}$ to each of these formulas, but only to some of them (namely, those occurring in nodes 1, 2, 7, and 13). The graph thus represent only *a part* of the sequent as it evolves along a branch.

The part of the sequent showcased in Figure 1 is already enough to demonstrate non-termination of the algorithm from [4]. Whenever the sequent grows "vertically" (by application of $\supset_{\mathsf{R}}$) we lose all the formulas a°, and in the newly introduced nodes we instead choose $\Delta^\bullet$, whence formulas $\Diamond b^\bullet$ and $\Diamond c^\bullet$ "reappear" in the sequent. These formulas make the sequent grow "horizontally," as rule $\Diamond_{\mathsf{L}}$ can further be applied to them.

However, Figure 1 already exhibits repetition in the proof search, and the algorithm from [4] is simply missing the correct *looping condition* for termination. In our talk we will discuss these conditions.