A Formal Development of Modal Logic IK in Coq: Labeled Sequent Calculus and Meta-theoretical Properties

Samar Rahmouni
samar.rahmouni@outlook.com
Institut Polytechnique de Paris, France
Advised by Lutz Strassburger
INRIA Saclay, France

Abstract

This project presents a complete formalization of the labelled sequent calculus for intuitionistic modal logic (LabIK) in the Coq proof assistant. We go over the system proposed in [23] and formalize it. It is then used to prove key meta-theoretical properties including admissibility of identity, weakening, invertibility and completeness with regards to intuitionistic modal logic (IK). We discuss different approaches and design choices made, precisely, when it comes to representing the different contexts and the labels necessary for the system. We also discuss the challenges faced during the formalization and the limitations of our implementation. The development in Coq is around 1200 lines and is available on Github.[1]
Keywords: intuitionistic modal logic, labeled sequent calculus, Coq

1 Introduction

Logics have long been used to model reasoning: classical logic for truth values [30], constructive logic for proofs [25], linear logic for resources management [7], and many more. Modal logic is a natural extension, as we strive to represent non-truth functional operators [3]. Precisely, the world "modal" refers to different ways to express new modes of truth.

It is in C.I. Lewis's seminal work [2] that the modal logic operators $\Box$ (necessity) and $\diamond$ (possibility) were introduced, allowing us to reason about what is necessarily or possibly true in a given context. An example of this is the modal logic S4. As Gödel investigated [36] the provability of mathematics, he found a way to reduce intuitionistic logic to classical logic augmented by a provability operator; precisely, Lewis' S4 logic [20, 22].

For our work, we turn our focus to intuitionistic modal logic (IK) [28], which foundations allow us to later formalize IS4 [11]. The problem of its decidability has been a long-standing open question since [32] and recently solved in [15]. Those results open up new avenues for mechanized formalizations and automated reasoning that would best be investigated in a proof assistant such as Coq.

As such, our motivation stands in laying the foundation for a complete formalization of the decidability proof of intuitionistic logic IS4. To this effect, we start by formalizing the labelled sequent calculus for intuitionistic logic (IK) [23], and proceed to prove key meta-theoretical properties such as weakening, admissibility of identity and the inversion of multiple rules. The labelled sequent calculus system is necessary to proceed with the proof, and has yet to be formalized in Coq.

2 Formal System

Intuitionistic modal logic (IK) will be the focus of our formalization, as it is the basis on which extensions of modal logics will be built upon. It is defined as follows. Let $\mathcal{A}$ be the set of atoms denoted by lowercase letters. A modal formula denoted by uppercase letters is constructed with the following grammar:

$$A, B ::= a \mid A \wedge B \mid A \vee B \mid A \supset B \mid \Box A \mid \diamond A \mid \bot$$

As the intuitionistic version of K does not allow for the DeMorgan duality $\neg\Box A \iff \diamond\neg A$, its axiomatization [28] that extends intuitionistic propositional logic (IPL) with the axioms of necessity (nec) but also

[1] https://github.com/natvern/labelled-Intuitionistic-logic-coq

axioms k1 to k5 which defines variants of the distributivity of k [20].

$$k: \quad \Box(p \supset q) \supset (\Box p \supset \Box q)$$

And such the axioms of the system IK are as follows:

$$\begin{array}{ll} nec: & A \supset \Box A \\ k2: & \Box(A \supset B) \supset (\diamond A \supset \diamond B) \\ k4: & (\diamond A \supset \Box B) \supset \Box(A \supset B) \end{array} \qquad \begin{array}{ll} k1: & \Box(A \supset B) \supset (\Box A \supset \Box B) \\ k3: & \diamond (A \vee B) \supset (\diamond A \vee \diamond B) \\ k5: & \diamond \bot \supset \bot \end{array}$$

As a note, the IS4 logic (intuitionistic S4) is an extension of IK with the addition of the following axioms:

$$4: \quad (\diamond \diamond A \supset \diamond A) \wedge (\Box A \supset \Box\Box A) \qquad\qquad t: \quad (A \supset \diamond A) \wedge (\Box A \supset A)$$

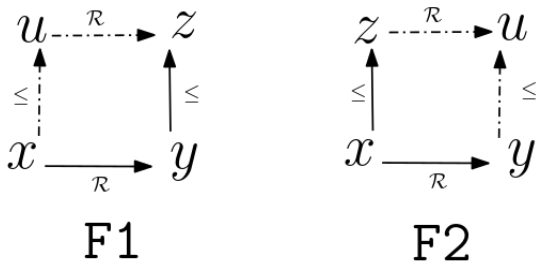
2.1 Kripke Semantics for IK

The Kripke semantics for IK as first defined in [11] is based on birelation frames. Precisely, it makes use of two relations, one based on the semantics of IPL and the other for classical modal logic (K). A Kripke frame is a pair $\mathcal{F} = (W, R)$ where W is a set of worlds and R is a binary relation on W. A bi-relational frame instead is a triple $\mathcal{F} = (W, R, \leq)$ where R defines the accessibility relations for modal logic (K) and $\leq$ is the preorder relation introduced by intuitionistic logic. These two relations are transcribed in the labelled sequent calculus as explained in the next section.

Definition 2.1. *A bi-relational frame $\mathcal{F} = (W, R, \leq)$ is a set of worlds W equipped with two relations R and $\leq$ (reflexive and transitive) such that:*

(F1) For all $x, y, z \in W$, if xRy and $y \leq z$, then there exists a u such that $x \leq u$ and uRz.

(F2) For all $x, y, z \in W$, if $x \leq y$ and xRz, then there exists a u such that yRu and $z \leq u$.



We can use this definition to build a *bi-relation model* $\mathcal{M}$.

Definition 2.2. *A bi-relational model $\mathcal{M}$ is a quadruple $(W, R, \leq, V)$ where $(W, R, \leq)$ is a bi-relational frame and $V : W \to 2^{\mathcal{A}}$ is a valuation function satisfying the monotonicity condition: for all $w, w' \in W$, if $w \leq w'$, then $V(w) \subseteq V(w')$ where V maps each w to the subset of the propositional atoms that are true at w.*

Applied to the rules of IK, the satisfaction relation $\models$ is defined as follows:

$$\begin{array}{ll} w \models a & \text{iff } a \in V(w) \\ w \models A \wedge B & \text{iff } w \models A \text{ and } w \models B \\ w \models A \vee B & \text{iff } w \models A \text{ or } w \models B \\ w \models A \supset B & \text{iff for all } w' \text{ such that } w \leq w', w' \models A \text{ implies } w' \models B \\ w \models \Box A & \text{iff for all } w' \text{ and } u \text{ such that } w \leq w' \text{ and } w'Ru, w' \models A \\ w \models \diamond A & \text{iff there exists } u \text{ such that } wRu \text{ and } u \models A \end{array}$$

2.2 Labelled Sequent Calculus System

The labelled proof calculi [14,35] allows to reason over Kripke semantics in the proof system itself. Precisely, every formula of the system is equipped with a label. Those labels allow us to build relational atoms. This power allows us to reason over a multitude of logics [17,27] which enjoy a close relation to Kripke semantics.

Consider two labels x and y. The relational atom xRy encoded the accessibility relation mentioned above. In the case of IK, since the Kripke semantics is based on birelation frames, there is two possible relational atoms, the xRy to define accessibility, but also $x \leq y$ to define preorder relation introduced by intuitionistic logic. This will be the building block for labelled IK. An inference rules then follows the same structure as the sequent calculus, but with the addition of labels and a relational context we denote by $\mathcal{R}$.

$$\frac{\mathcal{R}', \Gamma' \longrightarrow \Delta'}{\mathcal{R}, \Gamma \longrightarrow \Delta}\ r$$

The labelled sequent is then a triple $(\mathcal{R}, \Gamma, \Delta)$ where $\mathcal{R}$ is the relational atoms context, Γ and Δ are the labelled formulas contexts. Each context is a multiset; there was multiple ways to represent this in Coq, and more will be shown once we present the implementation.

3 Coq Formalization

$$\mathsf{id}\ \frac{}{\mathcal{R}, x \leq y, \Gamma, x{:}a \Longrightarrow \Delta, y{:}a} \qquad \bot_{\mathsf{L}}\ \frac{}{\mathcal{R}, \Gamma, x{:}\bot \Longrightarrow \Delta}$$

$$\wedge_{\mathsf{L}}\ \frac{\mathcal{R}, \Gamma, x{:}A, x{:}B \Longrightarrow \Delta}{\mathcal{R}, \Gamma, x{:}A \wedge B \Longrightarrow \Delta} \qquad \wedge_{\mathsf{R}}\ \frac{\mathcal{R}, \Gamma \Longrightarrow \Delta, x{:}A \quad \mathcal{R}, \Gamma \Longrightarrow \Delta, x{:}B}{\mathcal{R}, \Gamma \Longrightarrow \Delta, x{:}A \wedge B}$$

$$\vee_{\mathsf{L}}\ \frac{\mathcal{R}, \Gamma, x{:}A \Longrightarrow \Delta \quad \mathcal{R}, \Gamma, x{:}B \Longrightarrow \Delta}{\mathcal{R}, \Gamma, x{:}A \vee B \Longrightarrow \Delta} \qquad \vee_{\mathsf{R}}\ \frac{\mathcal{R}, \Gamma \Longrightarrow \Delta, x{:}A, x{:}B}{\mathcal{R}, \Gamma \Longrightarrow \Delta, x{:}A \vee B}$$

$$\supset_{\mathsf{R}}\ \frac{\mathcal{R}, x \leq y, \Gamma, y{:}A \Longrightarrow \Delta, y{:}B}{\mathcal{R}, \Gamma \Longrightarrow \Delta, x{:}A \supset B}\ y \text{ fresh}$$

$$\supset_{\mathsf{L}}\ \frac{\mathcal{R}, x \leq y, x{:}A \supset B, \Gamma \Longrightarrow \Delta, y{:}A \quad \mathcal{R}, x \leq y, \Gamma, y{:}B \Longrightarrow \Delta}{\mathcal{R}, x \leq y, \Gamma, x{:}A \supset B \Longrightarrow \Delta}$$

$$\Box_{\mathsf{L}}\ \frac{\mathcal{R}, x \leq y, yRz, \Gamma, x{:}\Box A, z{:}A \Longrightarrow \Delta}{\mathcal{R}, x \leq y, yRz, \Gamma, x{:}\Box A \Longrightarrow \Delta} \qquad \Box_{\mathsf{R}}\ \frac{\mathcal{R}, x \leq y, yRz, \Gamma \Longrightarrow \Delta, z{:}A}{\mathcal{R}, \Gamma \Longrightarrow \Delta, x{:}\Box A}\ y, z \text{ fresh}$$

$$\Diamond_{\mathsf{L}}\ \frac{\mathcal{R}, xRy, \Gamma, y{:}A \Longrightarrow \Delta}{\mathcal{R}, \Gamma, x{:}\Diamond A \Longrightarrow \Delta}\ y \text{ fresh} \qquad \Diamond_{\mathsf{R}}\ \frac{\mathcal{R}, xRy, \Gamma \Longrightarrow \Delta, x{:}\Diamond A, y{:}A}{\mathcal{R}, xRy, \Gamma \Longrightarrow \Delta, x{:}\Diamond A}$$

$$\mathsf{refl}\ \frac{\mathcal{R}, x \leq x, \Gamma \Longrightarrow \Delta}{\mathcal{R}, \Gamma \Longrightarrow \Delta} \qquad \mathsf{trans}\ \frac{\mathcal{R}, x \leq y, y \leq z, x \leq z, \Gamma \Longrightarrow \Delta}{\mathcal{R}, x \leq y, y \leq z, \Gamma \Longrightarrow \Delta}$$

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

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

Figure 1: Inference rules of Lab-IK$_{\leq}$

In this section, we present the Coq formalization of the labelled sequent calculus for IK. We will focus on the implementation used throughout the project for the proofs provided, and discuss the design choices and alternatives considered.

The rules in the figure are taken directly from [23] ; the symbols might differ from the ones used in this report. The colors in the figure show the labels in blue and the formulas in green. The implication symbol is also different. However, I would like to emphasize that is the same system.

3.1 The system labIK$_{\leq}$

The labelled IK system is defined in [23].

The syntax of the labelled sequent calculus is an inductive type that makes use of a label and a formula type as defined below. We represent formulas as an inductive type in Coq:

```
Inductive form : Type :=
| atom (A : string)
| bot
| and (A B : form)
| or (A B : form)
| impl (A B : form)
| box (A : form)
| diamond (A : form).
```

Listing 1: Formula representation in Coq

We represent relational atoms as a separate entity from formulas. This will make our $\mathcal{R}$ context easier to manage, and enforce.

```
Inductive relational_atom : Type :=
| Le (x y : label)  % x <= y
| Re (x y : label)  % x R y
```

Listing 2: Relational atoms

Putting these two together, a labelled formula is a pair of a label and a formula, defined as `Inductive labeled_form : Type := L (x : label) (A : form)` where its noted as `x : A`

The main decision that we made was the representation of the contexts. My first attempt was to define a sequent as a triple of a list of relational atoms, a list of labeled formulas, and a list of labeled formulas. Doing this, there are a couple of ways we can go about pattern matching for the rules. (1) as shown below is to only case on the head of the list.

```
Inductive labik : list relational_atom → list labeled_form → list labeled_form → Type :=
| labik_id :
  ∀ R Γ Δ s x y, In (y : atom s) Δ ∧
   In (x : atom s) Γ ∧
   In (x ≤ y) R → labik R Γ Δ
| labik_botl {R Γ Δ} (x : label) :
   In (x : ⊥) Γ → labik R Γ Δ
| labik_andl {R Γ Δ} (x : label) (A B : form) :
   labik R ((x : A)::(x : B):: Γ) Δ
   → labik R ((x : and A B):: Γ) (Δ)
| labik_andr {R Γ Δ} (x : label) (A B : form) :
   labik R Γ ((x : A):: Δ) → labik R Γ ((x : B)::Δ)
   → labik R Γ ((x : and A B):: Δ)
| (* other inference rules *)
```

Listing 3: Derivation in LabIK$_{\leq}$

Note that the cases that call for fresh variables are instantiated using the forall quantifier in the rule definition. Precisely, let us take the example of $\diamond_L$.

The rule :

$$\frac{xRy, \mathcal{R}, y : A \longrightarrow \Delta}{\mathcal{R}, x : \diamond A, \Gamma \longrightarrow \Delta} \diamond_L$$

This rule will look like this in Coq:

```
labik_dial {R Γ Δ} (A : form) (x : label):
    (∀ y : label, labik (x R y:: R) ((y : A):: Γ) (Δ))
```

This allows us to make use of the Coq's type system to generate those fresh variables for us.

Regarding the design choice of our formalization of contexts as lists that pattern match over the head, the important thing to note is that we have to prove the admissibility of the exchange rule. A way to circumvent this is to define the exchange rule as part of the system as we assume its admissibility, which is what is done in the formalization above. Precisely, for Δ we can define:

```
| labik_delta {R Γ Δ} (A : labeled_form):
  labik R Γ (Δ ++[A])
  → labik R Γ ( A :: Δ )
```

Listing 4: Exchange rule

This same definition is extended for Γ and $\mathcal{R}$. Clearly, this makes some of our proofs more complex, because of a successive application of the exchange rule depending on where the formula we want to focus on is located. To circumvent this, we can define helper rules that are only concerned with the two elements at the head of the list, and then switch between them. For most our proofs, this type of exchange is sufficient, since the rules themselves are defined in a way that the head of the list is the formula of interest. A question we might ask regarding those observations is whether a *focused sequent calculus* would be suitable for $labIK_{\leq}$, one both motivated by the decidability results of IS4 and automated proof search. We will not pursue this question in this work, but it is an interesting avenue for future work.

3.2 Further definitions

As most of the proofs are done over the height of the derivation tree, we define it as follows.

```
Fixpoint height {R Γ Δ} (p : labik R Γ Δ) : nat :=
match p with
| labik_id _ _ _ _ _ _ _ ⇒ 1
| labik_andl _ _ _ _ _ _ p' ⇒ 1 + (height p')
| labik_andr _ _ _ _ _ p1 p2 ⇒ 1 + max (height p1) (height p2)
| (* other cases *)
| labik_delta _ p' ⇒ height p'
end.
```

Listing 5: Proof height definition

Note that we assume the exchange rule does not change the height of the derivation tree; this is a reasonable assumption, as the initial definition of the contexts are that of multisets.

3.3 Proof Engineering Techniques

Most proofs proceed by structural induction on the derivation tree:

```
Theorem structural_property : ∀ R Γ Δ,
  labik R Γ Δ → (* property holds *).
Proof.
  intros R Γ Δ H.
  induction H.
  – (* identity case *)
  – (* conjunction left case *)
  – (* other cases *)
Qed.
```

Listing 6: Typical proof structure

For the proofs of stronger properties that are height-preserving, we use induction on the proof height.

```
Theorem height_preserving_property : ∀ R Γ Δ,
  ∀ H : labik R Γ Δ → sig (fun H1 : labik R (A::Γ) Δ ⇒ height H1 <= height H).
Proof.
  intros R Γ Δ H.
  – assert (conditions).
  – destruct IH.
    ∃ (derivation).
    simpl. assumption.
Qed.
```

Listing 7: Height induction pattern

For the base cases, we can easily prove that the conditions hold. For example, in the case of $\bot_L$ it must be that $\bot$ exists in Γ. For the inductive cases, we destruct the hypothesis to obtain the necessary information about the derivation tree, from that we can build the new derivation tree H1. Given H1 and the height of H, the proof of the height-preserving property is straightforward.

3.4 Proofs of Key Properties

The key properties we focus on in our formalization include: (1) Admissibility of general identity, (2) Weakening is height-preserving admissible, (3) Completeness with regards to IK and (4) Invertibility for single premises rules.

3.5 Admissibility of General Identity

The admissibility of general identity is a key property of the labelled sequent calculus, and will be used in the remainder of the proofs. It is our first structural proof on the formula A and states the following for all formulas A:

Proposition 3.1. *The general identity axiom* $\dfrac{}{\mathcal{R}, x \leq y, \Gamma, x : A \longrightarrow \Delta, y : A}\ id_g$ *is admissible in labIK*$_{\leq}$.

Proof. The proof proceeds by structural induction on the formula A. □

In Coq, the proof is straightforward with a single application of each rule, and the exchange rule is used minimally.

3.6 Weakening

Other properties, we are however concerned about are weakening and invertibility. For both proofs, the properties are made even stronger by an induction on the height; meaning that for every invertible rule, and weakening, the rule is height-preserving admissible. We start by proving the admissibiliy of weakening, which we define as follows:

$$\frac{\mathcal{R}, \Gamma \longrightarrow \Delta}{\mathcal{R}, \mathcal{R}', \Gamma, \Gamma' \longrightarrow \Delta, \Delta'}\ wk$$

Definition 3.2. *A rule is height-preserving admissible if every derivation of its premises, there exists a derivation of its conclusion of less or equal height.*

A rule is height-preserving invertible if for every derivation of the conclusion, there exists a derivation of its premises of less or equal height.

With this definition in mind, we can prove the following lemmas.

Lemma 3.3. *The weakening rule is height-preserving admissible for labIK$_\leq$.*

Proof. The formalization of the proof is surprisingly involved, and requires three auxiliary lemmas. Precisely, each lemma is concerned with one of the three contexts, i.e. $\mathcal{R}, \Gamma$, and Δ. We pose the following:

Lemma 3.4. $\forall\ x\ A$, *if* $\mathcal{R}, \Gamma \longrightarrow \Delta$ *then* $\mathcal{R}, x : A, \Gamma \longrightarrow \Delta$.

Proof. The proof proceeds by induction on the height of the derivation, and specializes for labelled formula $x : A$. The proof requires three auxiliary lemmas. We start with the case of Γ, as the other cases are similar.

```
Lemma labik_weakening :
  ∀ {R Γ Δ} (A : labeled_form) (H : labik R Γ Δ),
  sig (fun H1 : labik R (A:: Γ) Δ ⇒ labik_height H1 <= labik_height H).
```

Listing 8: Height-preserving weakening

The proof proceeds by induction on the height of the derivation. Precisely, let us look at the base case of identity.

```
  (* Base case : identity *)
  (* We want to assert the conditions for which identity holds *)
  - assert (In (y : atom s) Δ ∧ In (x : atom s) (A:: Γ) ∧ In (x ≤ y) R).
  (* what we have is a : that the identity holds for labik R Γ Δ *)
    + destruct a. destruct H0.
      split. (* we now have the conditions*)
      * simpl. assumption. (* Clearly the first condition holds trivially *)
      * split.
        -- apply list_weakening. assumption. (* The second condition holds by weakening on lists *)
        -- simpl. assumption. (* Trivial *)
    (* We are able to construct the identity proof for Γ = A, Γ*)
    + ∃ (labik_id R (A:: Γ) Δ s x y H). simpl. lia.
```

The inductive cases follow trivially from the inductive hypothesis and the application of the exchange rule. Interesting cases are ones that introduce fresh variables like implication right for instance.

```
  (* Implication right case *)
  (* The construction of the rule requires us to
     construct from the application of the generate variable
     to the proof *)
  - ∃ (labik_impr A0 B x (fun y ⇒
    let (H1, H1_height) := H y in
    labik_switch_gamma (y : A0) A H1)).
    simpl. edestruct H. simpl. lia.
```

This is part of the reason why the proof uses signatures, and not a logical exists, as we need to be able to give the generated y to the proof of the inductive hypothesis.

□

From the previous lemma, we can deduce a more general lemma for the context Γ that is if $\mathcal{R}, \Gamma \longrightarrow \Delta$ then $\mathcal{R}, \Gamma, \Gamma' \longrightarrow \Delta$ forall Γ'. The proof proceeds by induction on Γ. From a successive application of the three lemmas, we can deduce the more general weakening as defined above. The cases for Δ and $\mathcal{R}$ are similar. □

3.7 Invertibility

As mentioned, another property we are interested in is invertibility. We pose the following proposition, however we do not yet have the full formalization of the proof in Coq.

Proposition 3.5. *Single premises rules are height-preserving invertible in labIK$_\leq$. The $\wedge_L$ and $\vee_R$ are height-preserving invertible on both premises. The $\supset_L$ is height-preserving invertible on the right premise.*

The proof as shown in [23] proceeds by induction on the height of the derivation. In our formalized fragment of the proof, we do a standard induction on the derivation, meaning that we do not prove the height-preserving property. An example of this induction for $\wedge_L$ is as follows:

```
intros.
remember (z: A ∧ B::Γ) as Γ'.
induction (H).
– subst ; apply (labik_id _ _ _ s x y).
  destruct a. destruct H1. destruct H1.
  + discriminate.
  + split.
    * assumption.
    * split.
      –– simpl. right. right. assumption.
      –– assumption.
```

Precisely, for this case, we are concerned with the first rule application. We apply the identity rules with `x` and `y` as the labels, and `s` as the atom. So now we need to prove:

```
In (y : s) Δ ∧ In (x : s) (z : (A ∧ B) :: Γ) ∧ In (x ≤ y) R
```

We are able to prove each condition either through the assumptions or because of a contradiction in the hypothesis, precisely in the case where `z : A ∧ B = x : s`, which is impossible.

The most problematic cases are the ones that introduce free variables. Let us look at the inductive case for $\diamond_L$.

```
– inversion HeqΓ'. inversion HeqΓ''. subst. apply labik_impr.
  intro y.
  specialize l with (y:=y).
  specialize H0 with (y:=y).
  (* The issue is here *)
```

The issue precisely is that in most cases, we could discriminate over equalities that cannot occur. Precisely, saying that `z : A ∧ B = x : s` is impossible because a conjunction cannot be an atom. The same thing happens here. Even though we could discriminate directly in the case that do not introduce fresh variables, in the cases that do, they are part of an implication that I could not destroy.

```
H0 : y : A0 :: z : (A ∧ B) :: Γ = z : (A ∧ B) :: Γ →
     labik ((x ≤ y) :: R) (y : A0 :: z : (A ∧ B):: Γ) (y : B0 :: Δ) →
     z : A :: z : B :: Γ = z : A :: z : B :: Γ →
     labik ((x ≤ y) :: R) (z : A :: z : B :: Γ) (y : B0 :: Δ)
```

Precisely, the hypothesis `H0` should be reduced to the left hand side but I could not further the proof.

Another reason why I proceeded with proving the invertibility without the height-preserving property is because of the scope of the `forall`, which we use to introduce a fresh available. Let us consider the case of $\diamond_L$. A fresh label y is introduced in Γ, however, is also used in $\mathcal{R}$. If I am to build the proof H, the `forall` passed as a function can only be applied to build Γ.

As mentioned above, this is a work in progress, but below is the general idea, precisely for the case of $\wedge_L$. The remaining single premises cases will follow the same steps. We proceed by induction on the derivation.

```
Lemma invertible_andl:
  ∀ R Γ Δ A B z
  (H : labik R (z: A ∨ B::Γ) Δ),
  labik R (z: A:: z: B:: Γ) Δ.
  (* base case for identity *)
  (* we apply the identity rules for x y and atom s *)
  - apply (labik_id _ _ _ s x y).
    destruct a. destruct H1. destruct H1.
    (* case where z : A ∨ B = x : atom s => Impossible *)
    + discriminate.
    (* case where x : atom in Γ*)
    + split.
      * assumption.
      * split.
        -- simpl. right. right. assumption.
        -- assumption.
    (* case where the rule applied is ∧L *)
    - inversion HeqΓ'. subst. (* where can the rule be applied? *)
      + inversion HeqΓ''. (* what are the enforced equalities *)
        subst. assumption.
```

3.7.1 Cut-elimination

The ultimate goal of this section is to prove cut-elimination for labIK$_{\leq}$.

Theorem 3.6. *Any sequent provable in labIK$_{\leq}$ can be proved without using the cut rule.*

To prove this, a lemma is introduced if the premises of a cut rule are both cut-free (do not use any cut), then the conclusion is also cut-free. The proof is very involved, and has not been proven in our formalization yet. The complete proof is shown in [23]. Part of the issue is that the proof requires the height-preserving invertibility (Proposition 3.8), and such, we cannot proceed with the proof until we have it.

3.8 Completeness

The proofs that the axioms of propositional intuitionistic logic and IK are trivial, however, they showcase clearly how the rules are applied for an interactive proof in LabIK$_{\leq}$.

```
(* k3 : ◇(A ∨ B)  > (◇ A ∨ ◇B) *)
Lemma k3 : ∀ (x : label) (A B : form),
     ( nil , nil ) ⇒ (x : ◇(A ∨ B)  > ◇ A ∨ ◇ B) :: nil.
Proof.
  intros.
  apply labik_impr.                (* implication right *)
  intro y.                         (* fresh variable y *)
  apply labik_dial.                (* diamond left *)
  intro z.                         (* fresh variable z *)
  apply labik_orl.                 (* or left *)
  (* left premise *)
  – apply labik_orr.               (* or right *)
    apply labik_diar.              (* diamond right *)
    apply (labik_refl z).          (* reflexivity on z *)
    (* exchange rule *)
    apply id_g.
  (* right premise *)
  – apply labik_orr.               (* or right *)
    (* exchange rule *)
    apply labik_diar.              (* diamond right *)
    apply (labik_refl z).          (* reflexivity on z *)
    (* exchange rule *)
    apply id_g.
Qed.
```

Listing 9: Proof of axiom k3

Both premises are closed by general identity.

Theorem 3.7. LabIK$_{\leq}$ *is complete with respect to IK. For any theorem A of IK, A is also provable in* LabIK$_{\leq}$ + *cut*.

To be able to proceed with the proof, we define **cut** as follows:

$$\frac{\mathcal{R}, \Gamma \vdash \Delta, x : A \quad \mathcal{R}, \Gamma, x : A \vdash \Delta}{\mathcal{R}, \Gamma \vdash \Delta} \; cut$$

Proof. The proof proceeds by showing that all axioms k1 k5 and necessity are admissible in labIK$_{\leq}$. We extend our proof by showing that the axioms of propositional intuitionistic logic are also admissible. Here is an example of the proof of k5 : $x : \Diamond\bot \supset \bot$.

```
Lemma k5 : ∀ (x : label),
     (nil, nil) ⇒ (x : (◇⊥ ⟶ ⊥)) :: nil.
Proof.
  intro.
  apply labik_impr.
  intro y.
  apply labik_dial.
  intro z.
  apply (labik_botl z).
  simpl. left. reflexivity.
Qed.
```

This is equivalent to :

$$\dfrac{\dfrac{\dfrac{}{x \leq y, yRz, z : \bot \longrightarrow y : \bot}\,\bot_L}{x \leq y, y : \Diamond\bot \longrightarrow y : \bot}\,\Diamond_L}{\longrightarrow : x : \Diamond\bot \supset \bot}\,\supset_R$$

Note that we introduce y and z as fresh labels.

Cut is only needed to prove modus ponens, i.e. if P and P $\longrightarrow$ Q then Q, the proof is a direct application of cut.

The last remaining piece is the admissibility of the necessitation rule, which we formulate as follows:

Lemma 3.8. *$\forall x A$, if $x : A$ then $\forall y, y : \Box A$.*

The proof uses the admissibility of weakening, as well as a lemma to generalize over labels for empty contexts. Precisely,

Lemma 3.9. *if $x : A$ then $\forall y, y : A$.*

The proof of the above lemma is trivial since we are only concerned with the empty context, meaning that the label x is not used in the proof and every A is an axiom. □

Proof. To prove necessitation, suppose that $\Box A$ holds.

This is done by a direct application of $\Box_R$. $\dfrac{x \leq y, yRz \longrightarrow z : A}{\longrightarrow x : \Box A}\ \Box_R$. By lemma 3.4, we deduce $z : A$. Then, we apply weakening to $z : A$ with $x \leq y$ and yRz such that the hypothesis holds and $z : \Box A$ holds. Another application of the lemma yields us the result for $x : \Box A$.

```
    Goal ∀ (A : form),
  ∀ x : label, labik nil nil (x : A :: nil) →
  ∀ y : label,
  (nil, nil) ⇒ (y : □ A :: nil).
Proof.
  intros.
  assert (H1 : (nil, nil) ⇒ (x : □ A :: nil)).
  – apply labik_boxr. intros.
    assert (∀ z : label, (nil, nil) ⇒ (z : A :: nil)).
    + apply (helper_necc A x). assumption.
    + specialize H0 with (z := z).
      apply (labik_weakening_r (y0 R z)) in H0.
      apply (labik_weakening_r (x ≤ y0)) in H0.
      assumption.
  – apply (helper_necc (□ A) x). assumption.
Qed.
```

Note that `helper_necc` is the associated lemma 3.4.

We now have that all the theorems of IK are provable in labIK$_\leq$. As a result, labIK$_\leq$ is complete with respect to IK, i.e. Theorem 3.2. □

Although [23] goes into depth on the soundness proof of labIK$_\leq$, we will not cover it here. The proof requires a further formalization of the Kripke semantics and interpretations which go beyond the scope of this project.

3.9 Another Design Choice

In this section, we will discuss different approaches we tried for the representation of the contexts. The initial design and the one we stuck with is the one used in the report, however, one we have switched to (and have yet to finalize) also makes use of lists but do not enforce a pattern matching over the head of the list. Precisely, let us look at how the derivations differ.

```
Inductive Provable :
list relational_atom → list labeled_form → list labeled_form → Type :=
  | PId {G G' D D' C C'} x y a :
    Provable (G ++Le x y :: G') (D ++x : a :: D') (C ++y : a :: C')
  | BotL {G D D' C x} : Provable G (D ++(x : ⊥) :: D') C
  | AndL {G D D' C x A B} :
  (* other cases *)
```

The main difference is that we do not have a single context, but multiple depending on where the formula is located. This allows us to focus anywhere in the list, but also not enforce the exchange rule in the rules themselves. The exchange rule is still admissible, but not required. The main issues with this approach that made it difficult to work with is in the proofs themselves: Most of our proofs case on the head of the list, which while possible in this approach, still requires to give both two contexts. The pattern matching in Coq is not satisfactory to make the proofs clear.

Consider the cases where we must prove that the axioms of IK are admissible in our system, while it is enough to apply a rule directly and let Coq infer the context (as it is not important), in the case of this approach, we have to give not only the name of the rule but also the contexts that are used in the rule. This makes the proofs harder to read and more complex.

Another approach we looked at, and might have been the most promising, is to use multisets instead of lists. The setup of this approach is non trivial and tricky as shown in [31]. It would have taken us longer to implement, but in retrospect, would have made some of the proofs easier, and clearer.

4 Related Work

Modal logic was studied by several authors [3, 18, 26, 34], and a cut-free formulazation for S5, for instance, was presented in [4]. A completeness theorem for the logic been established in [19] and its complexity has been further studied in [21,33]. In [24,29], a higher-order approach is introduced and embedded. The theory of modal logic has been extensively studied, and a generalization over it as a 'hybrid logic' has even been proposed in [5]. However, intuitionistic modal logic has not been as widely explored.

Part of the reason is the difficulty to define the concept of what it means for a modal logic to be intuitionistic [11]. The proof theory and semantics behind intuitionistic modal logic is presented in [32]. Some methods [12] to reason about the logic use a contraction-free sequent calculi [8], hypersequent [1], the tableau method [16], and now the labelled sequent calculus [23].

The reason there is continued interest in modal logic is its wide range of applications, mostly in the temporal [6, 9] and epistemic reasoning [10].

So far, when it comes to intuitionistic modal logic, only a framework [28] has been proposed.

As such when it comes to formalization, limited work is found, but a lot of our implementation is based on Ian Shillito's work [31] for his formalization of intuitionistic logic, which we extend with the grammar and the rules of modal logic. The proofs we present ended up being a lot trickier due to the introduction of free variables in some of the rules.

Moreover, although the labelled sequent calculus [13, 14, 35] has been used in the proof theory of boolean bunched implications [17], and recently intuitionistic modal logic [15, 23] there has not been any Coq formalization of it, which had been an additional challenge, as it introduces a third context to represent the relations between labels.

5 Conclusion

Our project offers the basis of a formalization for the labelled sequent calculus for intuitionistic modal logic. We have successfully formalized the syntax and semantics of the system, and considered multiple design choices to make both the proofs clearer and its usage more intuitive. This formalization allowed us to prove some key theoretical properties of the system: weakening, admissibility of general identity, completeness with regards to IK, and invertibility for single premises rules. The project is available in GitHub[2] and documentation should be added for easier extension in the future.

The next step that is yet to be done is using our formalization and building blocks to ultimately prove cut elimination for labIK$_{\leq}$. Our hope is that this formalization allows us to formally prove in Coq the decidability of IS4.

[2] https://github.com/natvern/labelled-Intuitionistic-logic-coq

References

[1] Arnon Avron. The method of hypersequents in the proof theory of propositional non-classical logics. *Logic: from foundations to applications*, pages 1–32, 1996.

[2] Roberta Ballarin. Modern Origins of Modal Logic. In Edward N. Zalta and Uri Nodelman, editors, *The Stanford Encyclopedia of Philosophy*. Metaphysics Research Lab, Stanford University, Fall 2023 edition, 2023.

[3] Patrick Blackburn, Maarten De Rijke, and Yde Venema. *Modal logic*. Cambridge University Press, 2001.

[4] Torben Braüner. A cut-free gentzen formulation of the modal logic s5. *Logic Journal of IGPL*, 8(5):629–643, 2000.

[5] Torben Braüner. Hybrid logic and its proof-theory. volume 37. Springer, 2006.

[6] Edmund M Clarke, E Allen Emerson, and A Prasad Sistla. Automatic verification of finite-state concurrent systems using temporal logic specifications. *ACM Transactions on Programming Languages and Systems*, 8(2):244–263, 1986.

[7] Roberto Di Cosmo and Dale Miller. Linear Logic. In Edward N. Zalta and Uri Nodelman, editors, *The Stanford Encyclopedia of Philosophy*. Metaphysics Research Lab, Stanford University, Fall 2023 edition, 2023.

[8] Roy Dyckhoff. Contraction-free sequent calculi for intuitionistic logic. *The Journal of Symbolic Logic*, 57(3):795–807, 1992.

[9] E Allen Emerson. *Temporal and modal logic*, volume 2. Elsevier, 1990.

[10] Ronald Fagin, Joseph Y Halpern, Yoram Moses, and Moshe Y Vardi. *Reasoning about knowledge*. MIT press, 1995.

[11] Gisèle Fischer Servi. On modal logic with an intuitionistic base. *Studia Logica*, 36(3):141–149, 1977.

[12] Melvin Fitting. *Proof methods for modal and intuitionistic logics*, volume 169. Springer, 1983.

[13] Dov M Gabbay. *Labelled deductive systems*. Oxford University Press, 1996.

[14] Dov M. Gabbay. *Introduction to Labelled Deductive Systems*, pages 179–266. Springer Netherlands, Dordrecht, 2014.

[15] 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)*, 2023 38th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS), pages 1–13, Boston, United States, June 2023. IEEE.

[16] Rajeev Gore. Tableau methods for modal and temporal logics. *Handbook of tableau methods*, pages 297–396, 1999.

[17] Zhé Hóu, Alwen Tiu, and Rajeev Goré. A labelled sequent calculus for bbi: Proof theory and proof search. In Didier Galmiche and Dominique Larchey-Wendling, editors, *Automated Reasoning with Analytic Tableaux and Related Methods*, pages 172–187, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.

[18] George Edward Hughes and Max J Cresswell. *A new introduction to modal logic*. Routledge, 1996.

[19] Saul A Kripke. A completeness theorem in modal logic. *The Journal of Symbolic Logic*, 24(1):1–14, 1959.

[20] Saul A. Kripke. Semantical analysis of modal logic i normal modal propositional calculi. *Mathematical Logic Quarterly*, 9(5-6):67–96, 1963.

[21] Richard E Ladner. The computational complexity of provability in systems of modal propositional logic. *SIAM journal on computing*, 6(3):467–480, 1977.

[22] Clarence Irving Lewis. *Symbolic Logic*. The Century co., New York,, 1932.

[23] Sonia Marin, Marianela Morales, and Lutz Straßburger. A fully labelled proof system for intuitionistic modal logics. *Journal of Logic and Computation*, 31, 05 2021.

[24] Sean McLaughlin and Frank Pfenning. A higher-order approach to modal logic. In *International Conference on Theorem Proving in Higher Order Logics*, pages 246–261. Springer, 2008.

[25] Joan Moschovakis. Intuitionistic Logic. In Edward N. Zalta and Uri Nodelman, editors, *The Stanford Encyclopedia of Philosophy*. Metaphysics Research Lab, Stanford University, Summer 2024 edition, 2024.

[26] Sara Negri. Proof analysis in modal logic. *Journal of Philosophical Logic*, 34(5-6):507–544, 2005.

[27] Shoshin Nomura, Katsuhiko Sano, and Satoshi Tojo. *Revising a Labelled Sequent Calculus for Public Announcement Logic*, pages 131–157. Springer Berlin Heidelberg, Berlin, Heidelberg, 2016.

[28] Gordon Plotkin and Colin Stirling. A framework for intuitionistic modal logics: extended abstract. In *Proceedings of the 1986 Conference on Theoretical Aspects of Reasoning about Knowledge*, TARK '86, page 399–406, San Francisco, CA, USA, 1986. Morgan Kaufmann Publishers Inc.

[29] Thomas Raths, Jens Otten, and Christoph Kreitz. An embedding of modal logic in higher-order logic. In *International Conference on Automated Reasoning*, pages 35–49. Springer, 2007.

[30] Stewart Shapiro and Teresa Kouri Kissel. Classical Logic. In Edward N. Zalta and Uri Nodelman, editors, *The Stanford Encyclopedia of Philosophy*. Metaphysics Research Lab, Stanford University, Spring 2024 edition, 2024.

[31] Ian Shillito. Formalisation of proof systems (hilbert and sequent calculi) for intuitionistic logic. `https://github.com/ianshil/BIRS2025`, 2024.

[32] Alex K Simpson. *The proof theory and semantics of intuitionistic modal logic*. PhD thesis, University of Edinburgh, 1994.

[33] Edith Spaan. Complexity of modal logics. *PhD thesis, University of Amsterdam*, 1993.

[34] Johan van Benthem. *Modal Logic for Open Minds*. CSLI Publications, 2010.

[35] Luca Viganò. Labelled non-classical logics. 2000.

[36] Jan von Plato. Gödel's modal interpretation of intuitionistic logic and its proof theory. *Monatshefte für Mathematik*, 2025.