From Applicative Programming to Verification-based Knowledge: A Curry-Howard-Lambek Reading

Cosimo Perini Brogi

Scuola IMT Alti Studi Lucca

Abstract

We discuss a modular natural deduction framework for intuitionistic modal logics dealing with applicative functors and verificationist knowledge, extending a calculus for belief/applicative programming (IEL^{-}) to one (IEL) for knowledge in mathematics and other contexts as well. Through the proofs-as-programs paradigm, we map these systems into modal type theories and prove strong normalisation via CPS translations, uniformly handling detours and permutations. Finally, we give a categorical interpretation of the modal operator as a monoidal functor, reflecting the rewrite structure of the calculi.

Keywords

Verificationist epistemic logic, Proofs-as-programs, Intuitionistic modalities, Normalisation theorems, Categorical semantics

1. Introduction

Functional programming with applicative functors and verificationist epistemology share, perhaps unexpectedly, a common trait. At an abstract level, they are both grounded in a logical kernel based on the minimal intuitionistic modal logic $\mathbb{IK}$, extended with (at least) the *coreflection principle*: $A \to \Box A$. Here, the modality $\Box$ plays a dual role: it denotes either the parameterised type constructor of the applicative functor or the epistemic operator in a verificationist framework, depending on the context.

Applicative programming. Applicative functors were introduced by [1] and first implemented in Haskell [2], to appear afterwards in other functional languages, including OCaml [3] and F# [4], as well as in proof assistants such as Idris [5] and Agda [6]. They extend the computational capabilities of ordinary functors by allowing the application of functions within a computational context to arguments that are also embedded in such a context. In doing so, they enable the composition of effects without imposing the sequential dependencies characteristic of monads. This capability makes them particularly well-suited to scenarios where effects must be combined systematically while preserving independence.

Let us consider a concrete example to make the properties of applicative programming more evident. We implement a validation system using a parameterised sum type `('a, 'e) validation_result` that represents either success (containing validated data) or failure (containing accumulated errors), as shown in Figure 1a. The system validates user registration data (Figure 1b) through an applicative functor instance (Figure 1c) implementing `pure` and `apply` operations. The `apply` operation accumulates errors via list concatenation when both operands fail: unlike monadic validation, which short-circuits on the first error, applicative validation continues to collect all failures. Individual validators examine specific fields using pure, referentially transparent predicates (Figure 1d). Each encapsulates domain-specific

OVERLAY 2025, 7th International Workshop on Artificial Intelligence and Formal Verification, Logic, Automata, and Synthesis, October 26, 2025, Bologna, Italy

cosimo.perinibrogi@imtlucca.it (C. Perini Brogi)

https://sysma.imtlucca.it/people/cosimo-perini-brogi (C. Perini Brogi)

0000-0001-7883-5727 (C. Perini Brogi)

rules whilst maintaining consistent error reporting. These compose applicatively to validate entire records (Figure 1e), guaranteeing all validation functions execute regardless of individual failures.

A key advantage of this approach is that it enables computational optimisations which are unavailable in a purely monadic context. Because the arguments in an applicative computation are independent of one another, validation functions can be executed concurrently without violating referential transparency. Furthermore, under parallel evaluation, time complexity transforms from sequential $\mathcal{O}(t_1 + t_2 + \cdots + t_k)$ – where $k \leq n$ validations execute before first failure – to parallel $\mathcal{O}(\max(t_1, t_2, \ldots, t_n))$ – where all n validators execute concurrently.

```
type ('a, 'e) validation_result =
  | Success of 'a
  | Failure of 'e list

type validation_error =
  | UsernameTooShort
  | InvalidEmail
  | AgeTooYoung
```

(a) Validation result and error types

```
type user = {
  username: string;
  email: string;
  age: int;
}
```

(b) User record definition

```
module ValidationApplicative = struct
  type ('a, 'e) t = ('a, 'e) validation_result
  let pure x = Success x
  let apply f_result x_result =
    match f_result, x_result with
    | Success f, Success x -> Success (f x)
    | Success _, Failure errs -> Failure errs
    | Failure errs, Success _ -> Failure errs
    | Failure errs1, Failure errs2 ->
        Failure (errs1 @ errs2)
  let (<*>) = apply
end
```

(c) Applicative instance

```
let validate_username username =
  if String.length username >= 3
  then Success username
  else Failure [UsernameTooShort]

let validate_email email =
  try
    let at_pos = String.index email '@' in
    let len = String.length email in
    if at_pos > 0 && at_pos < len - 1 &&
       not (String.contains_from email (at_pos + 1)
       ↪ '@')
    then Success email
    else Failure [InvalidEmail]
  with Not_found -> Failure [InvalidEmail]

let validate_age age =
  if age >= 13
  then Success age
  else Failure [AgeTooYoung]
```

(d) Individual validators

```
open ValidationApplicative

let create_user username email age =
  { username; email; age }

let validate_user username email age =
  pure create_user
  <*> validate_username username
  <*> validate_email email
  <*> validate_age age
```

(e) Composition of validators using applicative style

Figure 1: Validation types and logic in applicative style. The type of `pure` corresponds to the coreflection principle $A \to \Box A$, and the type of `apply` to Kripke's principle $\Box(A \to B) \to \Box A \to \Box B$.

Intuitionistic epistemic states. On the purely modal front, verificationist epistemic logic has been developed axiomatically and semantically by [7] based on the coreflection principle. The minimal system known as $\mathbb{IEL}^-$ provides a formal framework for Kripke and coreflection principles. A modular extension, $\mathbb{IEL}$, introduces a further axiom schema for intuitionistic factivity of knowledge: $\Box A \to \neg\neg A$.

This subtle reinforcement reflects the verificationist stance that whatever is known through verification cannot be false. The logics IEL$^-$ and IEL bridge Brouwer-Heyting-Kolmogorov's proof-centric semantics (BHK) [8, 9] with Kripke's epistemic structures for intuitionistic reasoning [10], arguing that proof construction provides sufficient reason for belief. In the BHK interpretation, we have an implicit notion of proof whose epistemic counterparts are modelled by Kripke structures, and the construction of a proof for a specific proposition that we carried out as a cumulative mental process gives us sufficient reason for (at least) believing in that proposition. It is then possible to also cover traditional epistemic states of belief and knowledge within such a framework, once we recognise the correct operational clauses for corresponding modal operators: Proving A assures that A is true, proving $\Box A$ is weaker, for we may at least believe A even though we do not have a direct proof of A itself. Nevertheless, knowledge of A via a verification forces us to reject any claims for $\neg A$.

The two-part scenario in Figure 2 would make these points more concrete. The story's first part aligns with a verificationist account of *belief*. The second part explains *factivity of knowledge* in intuitionistic terms and qualifies it as a formal BHK-based version of *knowledge as belief tracking the truth of the matter under discussion* proposed by [11]. That reading suggests the application of this approach to epistemic reasoning in broader contexts than mathematical practice, including empirical knowledge and experimental practice in natural sciences [12, 13], evidence-based knowledge and classified/secret information [14, 15], and zero-knowledge interactions [16, 17], which would reveal more and more relevant as the technical integration of symbolic- and neural-based artificial intelligence progresses [18].

Part I: Your favourite top-rated mathematician has posted a proof sketch of a conjecture A in a very specialised mathematical field on her blog. Her informal argument is convincing, and her expertise justifies your trust in the conjecture. You are not an expert in that field, but she is; you do not have direct access to a detailed proof of A, but it is reasonable for you to believe that A does hold.

Part II: Some time has passed since the mathematician's post on conjecture A. You are again surfing the Net looking for new results in the field A belongs, and come across another post about that conjecture. This time, a famous top-rated computer scientist's blog announces that A has been refuted. And there's more: Her refutation has been computerised, and the formal proof is freely available on the Net. If you know the programming language at the base of her computer program to refute A – a proof assistant, or a highly specialised theorem prover – you can read the proof and see that $\neg A$ holds, so that your original trust in A is misplaced.

Figure 2: From verification-based belief to factive knowledge

Our contribution. In this paper, we report on our modular approach to the **design of natural deduction systems** for these logics – IEL$^-$ and IEL – with direct translations into modal type theories, that is, functional calculi for deductions. This extends the **proofs-as-programs** paradigm, familiar from intuitionistic logic and simple type theory, to the modalities involved in applicatives and epistemic states. This correspondence naturally lifts to the **categorical semantics**: the belief/applicative modality is interpreted as a monoidal functor with a monoidal point; the knowledge modality, as one that is also dense. These structures preserve conjunctions and reflect the initial object $\bot$, aligning categorical interpretation with modal reasoning. **Modularity** in the design of deduction rules underpins our central results. The original **strong normalisation proof** for the modal λ-calculus of IEL$^-$, first given in [19], is here extended – *modularly* – to IEL, via the addition of a modal elimination rule. Both proofs rely on a **CPS-translation** into simple type theory and instantiate the Friedman-Dragalin A-translation [20] in a modal setting. Further **logical properties** follow **uniformly** as corollaries of normalisation. We also prove that the **categorical structure** supporting belief in [19] carries over to our extended calculus IEL for knowledge, preserving the intended interpretation of deductions. We thus refine the **proof-theoretic and categorical analysis** of intuitionistic epistemic modalities within the Curry-Howard-Lambek correspondence to encompass knowledge and applicative functors in a more unified manner.

2. Curry-Howard-Lambek, Modularly

Background on proofs-as-programs. We start by a short presentation of the natural deduction paradigm and the calculus NJp for intuitionistic propositional logic – which is further discussed in proof-theoretic literature, such as [21].

Introduced first in [22], natural deduction systems for classical and intuitionistic logic, usually denoted by NK and NJ, respectively, provide a formal model for *deduction under hypotheses*. The key feature of these systems is the *absence of logical axioms* from the calculus. The entire deductive apparatus relies only on inference rules that encode the atomic deductive steps involved in any logical argument and reflect the operational meaning of the logical connectives of a given base language. Starting from *assumptions* – any formula A is per se a proof whose premise and conclusion are A itself – intuitionistic deductions are then developed according to the rules in Figure 3, defining the propositional fragment of NJ, that we denote by NJp.

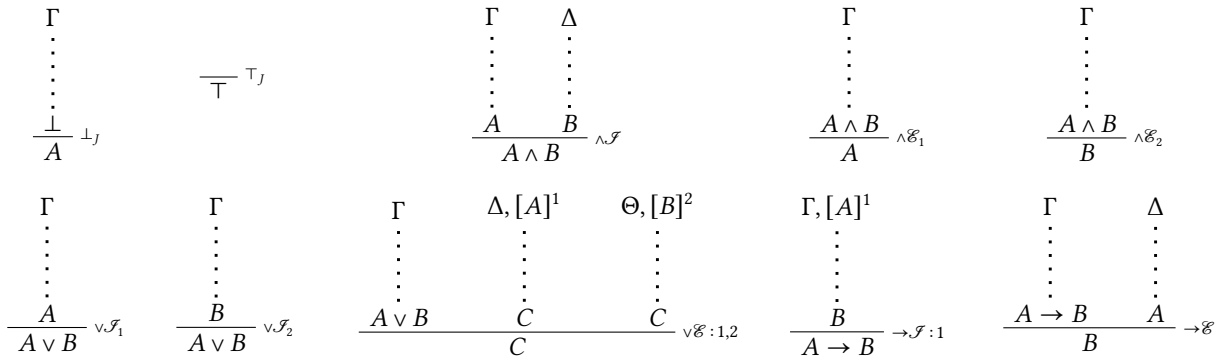

Figure 3: Rules for NJp

By identifying the types of a simple type theory with the same grammar generating the propositional formulas of NJp, we can then define a further correspondence between labelled deductions in NJp and the type assignments for typed terms, via the mapping shown in Figure 4. Thus, we can manipulate deductions in the proof system by performing computations in simple type theory.

Definition 1. *Let Ξ denote the reflexive and transitive closure of the relation $>_\Xi$ defined as follows:*[1]

Detours:
- $(\lambda x.f)g >_\Xi f[x := g]$
- $\pi_i(f_1, f_2) >_\Xi f_i$
- $\mathsf{C}(\mathsf{in}_i f, x.f_1, y.f_2) >_\Xi f_i[x_i := f]$

Permutations:
- $\mathsf{C}(\mathsf{C}(f, x.f_1, y.f_2), u.g_1, v.g_2) >_\Xi \mathsf{C}(f, x.\mathsf{C}(f_1, u.g_1, v.g_2), y.\mathsf{C}(f_2, u.g_1, v.g_2))$
- $\mathsf{C}(f, x.f_1, y.f_2)g >_\Xi \mathsf{C}(f, x.f_1 g, y.f_2 g)$
- $\pi_i \mathsf{C}(f, x.f_1, y.f_2) >_\Xi \mathsf{C}(f, x.\pi_i f_1, y.\pi_i f_2)$
- $\bot_J(f)g >_\Xi \bot_J(f)$
- $\pi_i \bot_J(f) >_\Xi \bot_J(f)$
- $\mathsf{C}(\bot_J(f), x.f_1, y.f_2) >_\Xi \bot_J(f)$
- $\bot_J(\bot_J(f)) >_\Xi \bot_J(f)$

where $g[x_i := f]$ denotes the substitution of f for all free occurrences of x_i in g.

A Ξ-normal form is just a typed term that cannot be rewritten with respect to Ξ. For this system of rewritings, it is possible to prove a confluence property and a **strong normalisation theorem**,[2] whose correspondent for NJp is proven in [24, 25].

Calculus design. We refer to [7] for the axiomatic calculi and relational semantics for IEL$^-$ and IEL ($\mathbb{IEL}^-$ and $\mathbb{IEL}$, respectively). We thus assume a monomodal grammar is given for the propositional fragment of intuitionistic logic extended by the $\Box$-modality: $A, B ::= p \mid \top \mid \bot \mid A \wedge B \mid A \vee B \mid A \to B \mid \Box A$, where p ranges among a infinite denumerable set of atomic propositions, and negation is defined as $\neg A := A \to \bot$.

[1] Type annotations are omitted for the sake of readability.

[2] Refer to [23] for the proofs.

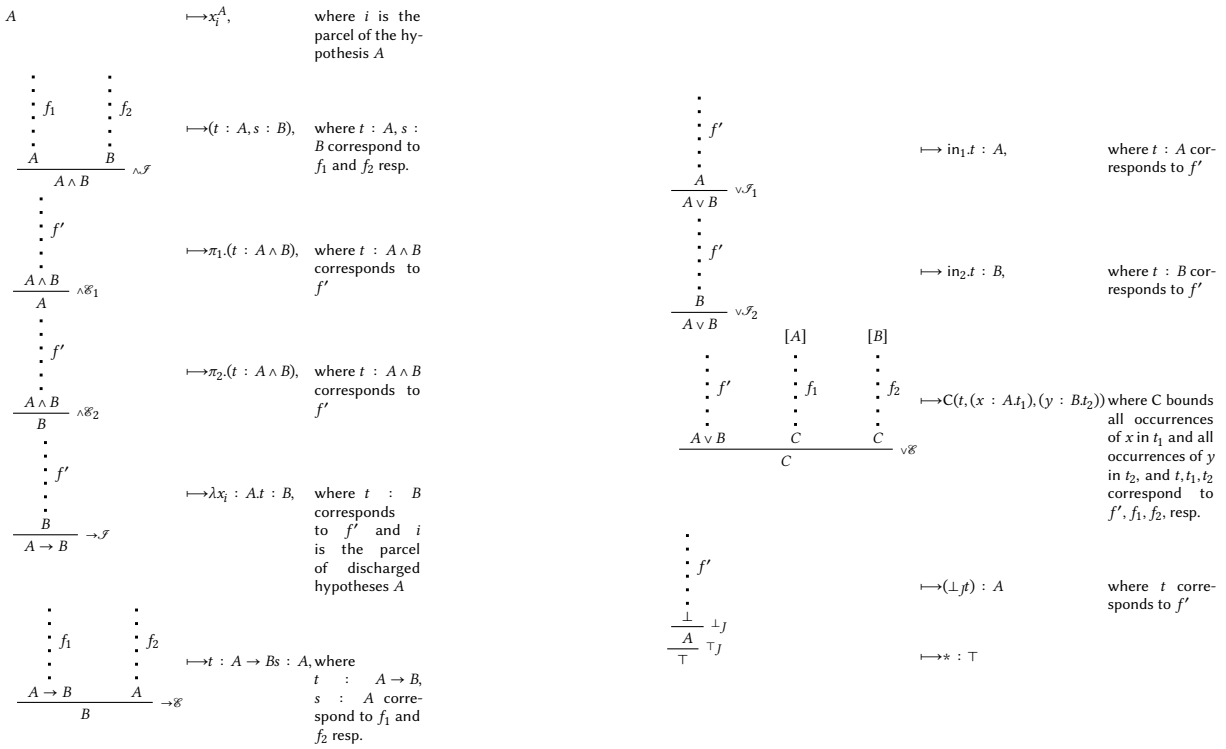

Figure 4: Mapping from deductions to typed-terms

Definition 2. *Let* IEL *be the calculus extending* NJp *by the following rules:*

$$\frac{\Gamma_1 \quad \cdots \quad \Gamma_n \quad [A_1, \cdots, A_n], \Delta}{\Box A_1 \quad \cdots \quad \Box A_n \quad B}\ \Box.\mathscr{I} \qquad \frac{\Gamma \quad [A], \Delta}{\Box A \quad \bot}\ \Box\mathscr{E}$$

where: Γ *and* Δ *are* finite multisets of formulas*;* $A_1, \cdots, A_n$ *are* all *discharged in* $\Box.\mathscr{I}$*; and* A *is discharged by* $\Box\mathscr{E}$*.*

Notice that the rule $\Box.\mathscr{I}$ differs from the one introduced by [26] by allowing the set Δ of additional hypotheses in the subdeduction of B. The $\Box\mathscr{E}$-free fragment of IEL is denoted by IEL^- and corresponds to the natural deduction calculus for intuitionistic belief of [19].

We write $\Gamma \Rightarrow_{\mathsf{L}} A$ when A is derivable in L for $\mathsf{L} \in \{\mathsf{IEL}^-, \mathsf{IEL}\}$ assuming the set of hypotheses Γ, and we write $\mathsf{L} \vdash A$ when $\emptyset \Rightarrow_{\mathsf{L}} A$. Denoting provability in the axiomatic calculi of [7] by the analogous $\Gamma \vdash_{\mathbb{L}} A$, we have – by straightforward induction on the length of the formal proofs – that $\Gamma \Rightarrow_{\mathsf{L}} A$ iff $\Gamma \vdash_{\mathbb{L}} A$.

Normalisation. Building on common notations for the standard Curry-Howard correspondence we recall in the opening of this section, we obtain modal λ-calculi by decorating L-deductions with proof names: we only need to extend the mapping of Figure 4 with type constructors for $\Box.\mathscr{I}$ ($\mathsf{box}[x_1, \cdots, x_n].\, g\, \mathsf{with}\, f_1, \cdots, f_n$) and $\Box\mathscr{E}$ ($\mathsf{unbox}\, f\, \mathsf{with}\, x.g$).

The rewriting system Ξ of Def. 1 (based on the conversions from [24]) is then extended by the conversions in Figure 5, which we state in natural deduction fashion for the sake of readability.

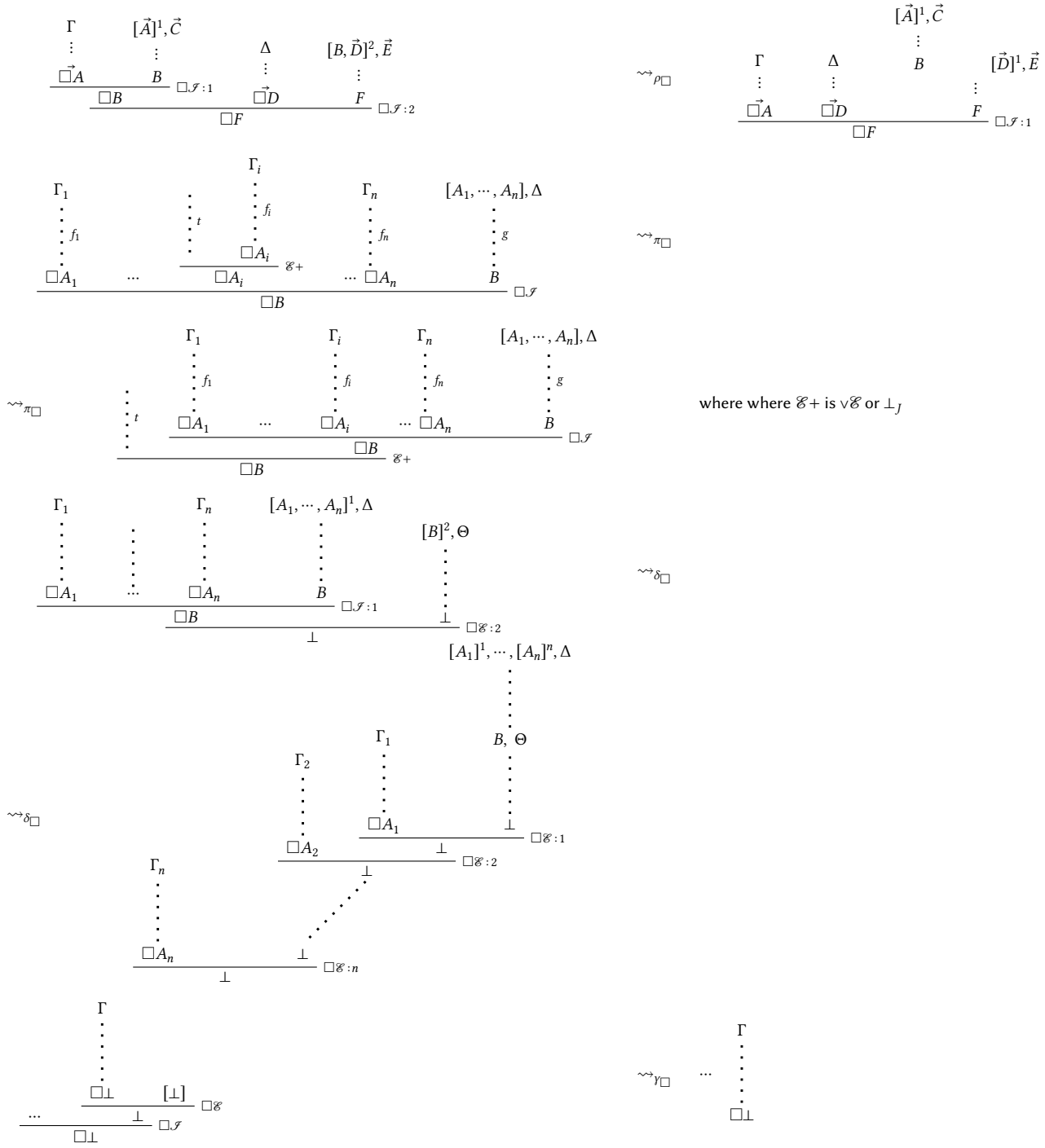

Figure 5: Modal conversions for IEL$^-$ ($\rho_\Box, \pi_\Box$) and IEL ($\rho_\Box, \pi_\Box, \delta_\Box, \gamma_\Box$), respectively.

The main theoretical result can be stated as

Theorem 1. *The following hold:*

1. *Deductions in* IEL$^-$ *strongly normalise w.r.t. the rewriting system* $\Xi + \rho_\Box + \pi_\Box$.
2. *Deductions in* IEL *strongly normalise w.r.t. the rewriting system* $\Xi + \rho_\Box + \pi_\Box + \delta_\Box + \gamma_\Box$.

Proof. A simple CPS translation as in [19] cannot handle modal permutations: we need a variant $\langle - \rangle$ of it, from our modal λ-calculus to simple type theory, defined as follows:

$$
\begin{array}{llll}
 & \langle \bot \rangle & := & \bot \\
 & \langle \top \rangle & := & \top \\
 & \langle p \rangle & := & p \\
\textit{To 1.} & \langle A \to B \rangle & := & \langle A \rangle \to \langle B \rangle \\
 & \langle A \wedge B \rangle & := & \langle A \rangle \wedge \langle B \rangle \\
 & \langle A \vee B \rangle & := & \langle A \rangle \vee \langle B \rangle \\
 & \langle \Box A \rangle & := & \langle A \rangle \vee q
\end{array}
$$

$$
\begin{array}{l}
\langle x \rangle := x \\
\langle * \rangle := * \\
\langle \bot_J(f) \rangle := \bot_J(\langle f \rangle) \\
\langle \lambda x\, f \rangle := \lambda x.\langle f \rangle \\
\langle f\, g \rangle := \langle f \rangle \langle g \rangle \\
\langle (f, g) \rangle := (\langle f \rangle, \langle g \rangle) \\
\langle \pi_i(f) \rangle := \pi_i(\langle f \rangle) \\
\langle \mathsf{C}(f, x.f_1, y.f_2) \rangle := \mathsf{C}(\langle f \rangle, x.\langle f_1 \rangle, y.\langle f_2 \rangle) \\
\langle \mathsf{in}_i(f) \rangle := \mathsf{in}_i(\langle f \rangle) \\
\langle \mathsf{box}[x_1, \cdots, x_n].\, g \,\mathsf{with}\, f_1, \cdots, f_n \rangle := \\
\quad \mathsf{C}(\langle f_n \rangle, x_n. \cdots \mathsf{C}(\langle f_2 \rangle, x_2. \\
\qquad \mathsf{C}(\langle f_1 \rangle, x_1.\mathsf{in}_1(\langle g \rangle)), y_1.\mathsf{in}_2(y_1)), y_2.\mathsf{in}_2(y_2)) \cdots, y_n.\mathsf{in}_2(y_n))
\end{array}
$$

It is then straightforward to check that both $\rho_\Box$ *and* $\pi_\Box$ are preserved by this translation, and recovered by permutation conversions of Ξ. Since deductions in NJp are strongly normalising w.r.t. Ξ – as proven in e.g. [24] – we have the desired strong normalisation for IEL^--deductions.

To 2. As for the previous point, by imposing $\langle \Box A \rangle := A \vee \bot$ and $\langle \mathsf{unbox}\, f \,\mathsf{with}\, x.g \rangle := \mathsf{C}(\langle f \rangle, x.\langle g \rangle, y.y)$. This translation tweaks preserve $\delta_\Box$ and $\gamma_\Box$ too, using applications of rewritings in Ξ concerning $\vee$-detours and permutations. □

Combined with associated proofs of the subformula property of normal L-deductions, normalisation has interesting corollaries for these logics, namely: consistency, decidability, disjunction property, $\Box$-primality, and modal disjunction property [19, 27].

Categorical semantics. The correspondence between type theory and natural deduction is further extended to category theory via the Curry-Howard-Lambek correspondence summarised by the partial glossary of Figure 6.

Logic	Type Theory	Category Theory
proposition	type	object
proof	term	arrow
theorem	inhabitant	element-arrow
conjunction	product type	product
true	unit type	terminal object
implication	function type	exponential
disjunction	sum type	(weak) coproduct
false	empty type	(weak) initial object

Figure 6: A Curry-Howard-Lambek correspondence glossary

The defining properties of these categorical constructions correspond to rewritings of deductions in NJp, as documented in [28, Ch. I.8]. Note, however, that the system of rewritings imposed by the universal properties of the initial object and coproducts *is a proper extension* of Ξ from Def. 1. We thus refer to the categorical counterparts of $\bot$ and $\vee$ w.r.t. system Ξ as the **weak initial object** and **weak coproduct**, respectively.

For space reasons, we omit the standard definition of the notions for categories, functors, and natural transformations – we refer the reader to classic textbooks such as [28] and [29] for more details – to recall that any category having products and exponentials for any of its objects is called **cartesian closed** (CCCat); moreover, if it has also (weak) coproducts, it is called **bi-cartesian closed** (bi-CCCat).

For epistemic operators, we need to add to the basics of the categorical semantics for NJp the following definitions.

Definition 3. *Given any CCCat $\mathscr{C}$,*

- *an endofunctor $\mathfrak{F} : \mathscr{C} \to \mathscr{C}$ is **pointed** iff there exists a natural transformation $\pi : Id_{\mathscr{C}} \Rightarrow \mathfrak{F}$ such that $\mathfrak{F}f \circ \pi_A = \pi_B \circ f$, where π is called the **point** of $\mathfrak{F}$.*

- *a* ***monoidal endofunctor*** *consists of a functor* $\mathfrak{F} : \mathscr{C} \to \mathscr{C}$ *together with*
  - *a natural transformation* $m_{A,B} : \mathfrak{F}A \times \mathfrak{F}B \to \mathfrak{F}(A \times B)$*;*
  - *a morphism* $m_1 : 1 \to \mathfrak{F}1$*,*

  preserving the monoidal structure of $\mathscr{C}$*.*[3] *These are called* ***structure morphisms*** *of* $\mathfrak{F}$*.*
- *given monoidal endofunctors* $\mathfrak{F}, \mathfrak{G} : \mathscr{C} \to \mathscr{C}$*, a natural transformation* $\kappa : \mathfrak{F} \Rightarrow \mathfrak{G}$ *is* ***monoidal*** *when* $\kappa_{A\times B} \circ m^{\mathfrak{F}}_{A,B} = m^{\mathfrak{G}}_{A,B} \circ \kappa_A \times \kappa_B$ *and* $\kappa_1 \circ m^{\mathfrak{F}}_1 = m^{\mathfrak{G}}_1 \circ id_1$*.*

Definition 4. *An* IEL^-***-category*** *is given by a bi-CCCat* $\mathscr{C}$ *together with a monoidal pointed endofunctor* $\mathfrak{K}$ *whose point* κ *is monoidal. When* $\mathfrak{K}$ *preserves the initial object* 0 *of* $\mathscr{C}$ *up-to-isomorphism, we say that* $\mathfrak{K}$ *is* ***dense****, and* $\mathscr{C}$ *is an* IEL***-category****.*

It is not hard to see that $\rho_\Box$ is needed to prove that the construction induced by the modal operator preserves arrow composition. However, that construction cannot be a functor unless one considers the following extensionality principle $\eta_\Box$ for $\Box.\mathscr{I}$: $\dfrac{\begin{array}{c}\Gamma\\ \vdots\\ \Box A\end{array} \qquad [A]}{\Box A}\,\Box.\mathscr{I} \quad \rightsquigarrow \quad \begin{array}{c}\Gamma\\ \vdots\\ \Box A\end{array}$

IEL^--deductions strongly normalise w.r.t. the rewriting system $\Xi + \rho_\Box + \delta_\Box + \gamma_\Box + \eta_\Box$: The translation $\langle - \rangle$ is capable of mimicking $\eta_\Box$ in a simple type theory extended by a sum extensionality principle $\eta_\vee$, which in type-theoretic syntax can be defined as $\mathsf{C}(f, x.\mathsf{in}_1(x), y.\mathsf{in}_2(y)) \;>_{\eta_\vee}\; f$.
Normalisation of L-deductions w.r.t. $\Xi + \rho_\Box + \delta_\Box + \gamma_\Box + \eta_\Box$ is then derived from the fact that $\eta_\vee$-reductions can be postponed in any sequence of $\Xi + \eta_\vee$-rewritings, so that simple type theory is normalising w.r.t. $\Xi + \eta_\vee$ too [23, Ch. 4]. The adequacy of this categorical interpretation is now natural to state and prove:

Theorem 2. *The following hold:*

(i) Let $\mathscr{C}$ *be an* IEL^-*-category. Then there is a canonical interpretation* $[\![-]\!]$ *of* IEL^- *in* $\mathscr{C}$ *such that*

- *a formula* A *is mapped to a* $\mathscr{C}$*-object* $[\![A]\!]$*;*
- *a deduction* f *of* $A_1, \cdots, A_n \Rightarrow_{\mathsf{IEL}^-} B$ *is mapped to an arrow* $[\![f]\!] : [\![A_1]\!] \times \cdots \times [\![A_n]\!] \to [\![B]\!]$*;*
- *for any two deductions* f *and* g *which are equal modulo* $\Xi + \rho_\Box + \eta_\Box$*-rewritings, we have* $[\![f]\!] = [\![g]\!]$*.*

The analogous soundness result holds for any IEL*-category, by considering the system of rewritings* $\Xi + \rho_\Box + \delta_\Box + \gamma_\Box$ *extended by* $\eta_\Box$*.*

(ii) If the interpretation of two L*-deductions is equal in all* L*-categories, then the two deductions are equal modulo* $\Xi + \rho_\Box + \delta_\Box + \gamma_\Box + \eta_\Box$*.*

Acknowledgments

I am grateful to the two anonymous reviewers for their constructive feedback and suggestions.

This work was partially supported by: the project SERICS – Security and Rights in CyberSpace PE0000014, financed within PNRR, M4C2 I.1.3, funded by the European Union - NextGenerationEU (MUR Code: 2022CY2J5S, CUP: D67G22000340001); the International Research Network "Logic and Interaction"; the EC-COST Action (CA20111) "EuroProofNet".

Declaration on Generative AI

The author has not employed any Generative AI tools.

[3] See [29, Sect. VII.1] for the corresponding commuting diagrams and the definition of monoidal category.

References

[1] C. McBride, R. Paterson, Applicative programming with effects, Journal of functional programming 18 (2008) 1–13.

[2] HaskellWiki, Applicative functor — haskellwiki, 2022. URL: https://wiki.haskell.org/index.php?title=Applicative_functor&oldid=65058, [Online; accessed 24-June-2025].

[3] Jane Street Developers, Applicative (base.Base.Applicative), https://ocaml.janestreet.com/ocaml-core/v0.12/doc/base/Base/Applicative/index.html, 2025. [Accessed 24-06-2025].

[4] F#+ Developers, Functors and Applicatives, https://fsprojects.github.io/FSharpPlus/applicative-functors.html, 2025. [Accessed 24-06-2025].

[5] Idris Doc Developers, IdrisDoc: Prelude.Applicative, https://www.idris-lang.org/docs/current/prelude_doc/docs/Prelude.Applicative.html, 2025. [Accessed 24-06-2025].

[6] A. Abel, Equivalence of applicative functors and multifunctors, arXiv preprint arXiv:2401.14286 (2024).

[7] S. Artemov, T. Protopopescu, Intuitionistic epistemic logic, The Review of Symbolic Logic 9.2 (2016) 266–298.

[8] A. Troelstra, Principles of Intuitionism, Springer, 1969.

[9] D. van Dalen, A. Troelstra, Constructivism in Mathematics. An Introduction I, volume 121 of *Studies in Logic and the Foundations of Mathematics*, Elsevier, 1988.

[10] S. A. Kripke, Semantical analysis of intuitionistic logic I, in: Studies in Logic and the Foundations of Mathematics, volume 40, Elsevier, 1965, pp. 92–130.

[11] R. Nozick, Philosophical explanations, Ethics 94 (1981).

[12] A. Aldini, G. Curzi, P. Graziani, M. Tagliaferri, Trust evidence logic, in: J. Vejnarová, N. Wilson (Eds.), Symbolic and Quantitative Approaches to Reasoning with Uncertainty - 16th European Conference, ECSQARU 2021, Prague, Czech Republic, September 21-24, 2021, Proceedings, volume 12897 of *Lecture Notes in Computer Science*, Springer, 2021, pp. 575–589. doi:10.1007/978-3-030-86772-0_41.

[13] A. Aldini, P. Graziani, M. Tagliaferri, Reasoning about ignorance and beliefs, in: L. Cleophas, M. Massink (Eds.), Software Engineering and Formal Methods. SEFM 2020 Collocated Workshops - ASYDE, CIFMA, and CoSim-CPS, Amsterdam, The Netherlands, September 14-15, 2020, Revised Selected Papers, volume 12524 of *Lecture Notes in Computer Science*, Springer, 2020, pp. 214–230. doi:10.1007/978-3-030-67220-1_17.

[14] A. Aldini, G. Curzi, P. Graziani, M. Tagliaferri, A probabilistic modal logic for context-aware trust based on evidence, Int. J. Approx. Reason. 169 (2024) 109167. doi:10.1016/J.IJAR.2024.109167.

[15] A. Aldini, D. Fazio, P. Graziani, R. Mascella, M. Tagliaferri, A logical perspective on intending to keep a true secret, Journal of Logic and Computation 35 (2025) exaf028. doi:10.1093/logcom/exaf028.

[16] M. Backes, F. Bendun, M. Maffei, E. Mohammadi, K. Pecina, Symbolic malleable zero-knowledge proofs, in: C. Fournet, M. W. Hicks, L. Viganò (Eds.), IEEE 28th Computer Security Foundations Symposium, CSF 2015, Verona, Italy, 13-17 July, 2015, IEEE Computer Society, 2015, pp. 412–426. URL: https://doi.org/10.1109/CSF.2015.35. doi:10.1109/CSF.2015.35.

[17] G. Costa, C. Perini Brogi, Toward dynamic epistemic verification of zero-knowledge protocols, in: G. D'Angelo, F. L. Luccio, F. Palmieri (Eds.), Proceedings of the 8th Italian Conference on Cyber Security (ITASEC 2024), Salerno, Italy, April 8-12, 2024, volume 3731 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2024. URL: https://ceur-ws.org/Vol-3731/paper25.pdf.

[18] M.-C. Dinu, C. Leoveanu-Condrei, M. Holzleitner, W. Zellinger, S. Hochreiter, SymbolicAI: A framework for logic-based approaches combining generative models and solvers, in: V. Lomonaco, S. Melacci, T. Tuytelaars, S. Chandar, R. Pascanu (Eds.), Proceedings of The 3rd Conference on Lifelong Learning Agents, volume 274 of *Proceedings of Machine Learning Research*, PMLR, 2025, pp. 869–914. URL: https://proceedings.mlr.press/v274/dinu25a.html.

[19] C. Perini Brogi, Curry-Howard-Lambek Correspondence for Intuitionistic Belief, Studia Logica 109 (2021) 1441–1461. doi:10.1007/S11225-021-09952-3.

[20] D. Leivant, Syntactic translations and provably recursive functions, The Journal of Symbolic Logic 50 (1985) 682–688.
[21] D. van Dalen, Logic and Structure, 4th ed., Springer, 2008.
[22] G. Gentzen, Untersuchungen über das logische Schließen I, Mathematische zeitschrift 39 (1935) 176–210.
[23] M. H. Sørensen, P. Urzyczyn, Lectures on the Curry-Howard isomorphism, volume 149 of *Studies in Logic and the Foundations of Mathematics*, Elsevier, 2006.
[24] D. Prawitz, Ideas and results in proof theory, in: Studies in Logic and the Foundations of Mathematics, volume 63, Elsevier, 1971, pp. 235–307.
[25] P. Mancosu, S. Galvan, R. Zach, An Introduction to Proof Theory: Normalization, Cut-Elimination, and Consistency Proofs, Oxford University Press, 2021.
[26] V. de Paiva, E. Ritter, Basic constructive modality, Logic without Frontiers: Festschrift for Walter Alexandre Carnielli on the occasion of his 60th Birthday (2011) 411–428.
[27] C. Perini Brogi, Investigations of proof theory and automated reasoning for non-classical logics, Ph.D. thesis, University of Genoa, Italy, 2022.
[28] J. Lambek, P. J. Scott, Introduction to higher-order categorical logic, volume 7, Cambridge University Press, 1988.
[29] S. Mac Lane, Categories for the working mathematician, volume 5, Springer Science & Business Media, 2013.
[30] I. van der Giessen, Admissible rules for six intuitionistic modal logics, Ann. Pure Appl. Log. 174 (2023) 103233. doi:10.1016/J.APAL.2022.103233.
[31] S. Guerrini, A. Masini, M. Zorzi, Natural deduction calculi for classical and intuitionistic S5, J. Appl. Non Class. Logics 33 (2023) 165–205. doi:10.1080/11663081.2023.2233750.
[32] C. Hagemeier, D. Kirst, Constructive and mechanised meta-theory of IEL and similar modal logics, J. Log. Comput. 32 (2022) 1585–1610. doi:10.1093/LOGCOM/EXAC068.