An Institutional View on Categorical Logic

Florian Rabe[1], Till Mossakowski[2], Valeria De Paiva[3], Lutz Schröder[2], and Joseph Goguen[4]

[1] School of Engineering and Science, International University Bremen
[2] DFKI-Lab Bremen and Department of Computer Science, University of Bremen
[3] Intelligent Systems Laboratory, Palo Alto Research Center, California
[4] Dept. of Computer Science & Engineering, University of California, San Diego

Abstract. We introduce a generic notion of propositional categorical logic and provide a construction of a proof-theoretic institution out of such a logic, following the Curry-Howard-Tait paradigm. The logics are specified as theories of a meta-logic within the logical framework LF. We prove logic-independent soundness and completeness theorems and instantiate our framework with a number of examples: classical, intuitionistic, linear and modal logic.

We wish to dedicate this work to the memory of our dear friend and colleague Joseph Goguen who passed away during its preparation.

1 Introduction

The well-known Curry-Howard-Tait isomorphism establishes a correspondence between propositions and types, proofs and terms, and proof reductions and term reductions. A number of deep correspondences between categories and logical theories have been established, see Fig. 1.

conjunctive logic	cartesian categories	[19]
positive logic	cartesian closed categories	[19]
intuitionistic propositional logic	bicartesian closed categories	[19]
classical propositional logic	bicartesian closed categories with $\neg\neg$-elimination	[19]
modal logics, e.g., intuitionistic S4	monoidal comonad with strong monad	[4]
linear logic	$*$-autonomous categories	[33]
first-order logic	hyperdoctrines	[31]
Martin-Löf type theory	locally cartesian closed categories	[32]

Fig. 1. Curry-Howard correspondences

Here, we present work aimed at casting the propositional part of these correspondences in a common framework based on the theory of *institutions*. The notion of institution arose within computer science as a response to the population explosion among logics in use [15, 17]. Its key idea is to focus on abstractly axiomatizing the *satisfaction* relation between sentences and models. A wide range of logics, e.g., higher-order [5], polymorphic [30], temporal [14], process [14], behavioral [3], and object-oriented [16] logics, have been formalized as institutions. A surprisingly large amount of meta-logical reasoning can be carried out in this abstract framework; e.g., institutions have been used to give general foundations for modularization of theories and programs [29, 11], and substantial portions of classical model theory can be lifted to the level of institutions [34, 7, 8, 10]. The objective is to be able to apply each meta-mathematical result to the widest possible range of abstract institutions; see e.g. [6–10]

In the sequel, we give a general notion of propositional categorical logic, and a construction of a proof-theoretic institution out of such a logic that formalizes the abstract Curry-Howard-Tait correspondence. Then we use the institutional meta-theory to establish logic-independent results including general soundness and completeness theorems, and we discuss how the Curry-Howard-Tait isomorphism can be cast as a morphism of institutions.

2 Institutions and Logics

We assume that the reader is familiar with basic notions from category theory (cf. e.g. [1, 20]). We denote the class of objects of a category $\mathbb{C}$ by $|\mathbb{C}|$, the set of morphisms form A to B by $\mathbb{C}(A, B)$, and composition by $\circ$. Let $\mathbb{CAT}$ be the the quasi-category of all categories (quasi-categories are categories that live in a higher set-theoretic universe [1]). Institutions are defined as follows.

Definition 1. An *institution* $\mathcal{I} = (\mathbb{S}ign^{\mathcal{I}}, \mathsf{Sen}^{\mathcal{I}}, \mathsf{Mod}^{\mathcal{I}}, \models^{\mathcal{I}})$ consists of

- a category $\mathbb{S}ign^{\mathcal{I}}$ of *signatures*;
- a functor $\mathsf{Sen}^{\mathcal{I}} : \mathbb{S}ign^{\mathcal{I}} \to \mathbb{S}et$ giving, for each signature Σ, the set $\mathsf{Sen}^{\mathcal{I}}(\Sigma)$ of *sentences*, and for each signature morphism $\sigma \colon \Sigma \to \Sigma'$, the *sentence translation map* $\mathsf{Sen}^{\mathcal{I}}(\sigma) \colon \mathsf{Sen}^{\mathcal{I}}(\Sigma) \to \mathsf{Sen}^{\mathcal{I}}(\Sigma')$, with $\sigma\varphi$ denoting $\mathsf{Sen}^{\mathcal{I}}(\sigma)(\varphi)$;
- a functor $\mathsf{Mod}^{\mathcal{I}} \colon (\mathbb{S}ign^{\mathcal{I}})^{op} \to \mathbb{CAT}$ giving, for each signature Σ, the category $\mathsf{Mod}^{\mathcal{I}}(\Sigma)$ of *models*, and for each signature morphism $\sigma \colon \Sigma \to \Sigma'$, the *reduct functor* $\mathsf{Mod}^{\mathcal{I}}(\sigma) \colon \mathsf{Mod}^{\mathcal{I}}(\Sigma') \to \mathsf{Mod}^{\mathcal{I}}(\Sigma)$, with $M'|_\sigma$ denoting $\mathsf{Mod}^{\mathcal{I}}(\sigma)(M')$, the *$\sigma$-reduct* of M'; and
- a satisfaction relation $\models^{\mathcal{I}}_{\Sigma} \subseteq |\mathsf{Mod}^{\mathcal{I}}(\Sigma)| \times \mathsf{Sen}^{\mathcal{I}}(\Sigma)$ for each $\Sigma \in |\mathbb{S}ign^{\mathcal{I}}|$,

such that for each $\sigma \colon \Sigma \to \Sigma'$ in $\mathbb{S}ign^{\mathcal{I}}$, the *satisfaction condition*

$$M' \models^{\mathcal{I}}_{\Sigma'} \sigma\varphi \iff M'|_\sigma \models^{\mathcal{I}}_{\Sigma} \varphi$$

holds for all $M' \in |\mathsf{Mod}^{\mathcal{I}}(\Sigma')|$ and all $\varphi \in \mathsf{Sen}^{\mathcal{I}}(\Sigma)$.

Example 2 (Classical propositional logic). As a running example, we give the institution **CPL** of classical propositional logic. Its signatures are just sets Σ of propositional variables, i.e., $\mathbb{S}ign = \mathbb{S}et$. $\mathsf{Sen}(\Sigma)$ is the set of propositional formulas with propositional variables from Σ defined in the usual way. A signature morphism σ is a mapping between the propositional variables, and sentence translation $\mathsf{Sen}(\sigma)$ is the extension of σ to all formulas. Models of Σ are truth valuations, i.e., mappings from Σ into the boolean algebra $Bool = \{0, 1\}$. For simplicity, we omit model morphisms here. Finally, $M \models_\Sigma \varphi$ holds iff φ evaluates to 1 under the usual extension of M to all formulas.

Proof-theoretic institutions use richer structures than sets of sentences to express the syntax of an institution, they were first explored in [23]. Here, we introduce entailment institutions that do not use proof between sets of sentences but proof categories with sentences as objects and proofs as morphisms. Moreover, reductions between proof terms are modeled by a preorder on morphisms. We write $f \rightsquigarrow g$ to express that f reduces to g for two proofs f, g with the same domain and codomain. In other words, proof categories are small preorder-enriched categories, and we write $\mathbb{O}rdCat$ for the category of preorder-enriched small categories. If $U : \mathbb{C}at \to \mathbb{S}et$ is the functor forgetting morphisms, this is formally defined as follows.

Definition 3. An *entailment institution* is a tuple $(\mathbb{S}ign, \mathsf{Pr}, \mathsf{Mod}, \models)$ where $\mathsf{Pr} : \mathbb{S}ign \to \mathbb{O}rdCat$ is a functor such that $(\mathbb{S}ign, U \circ \mathsf{Pr}, \mathsf{Mod}, \models)$ is an institution. In that case, we will use Sen to abbreviate $U \circ \mathsf{Pr}$.

Example 4. CPL can be turned into an entailment institution in various ways. Let $\mathbf{CPL}^{ND}$ denote propositional logic with natural deduction. A morphism $p \in \mathsf{Pr}(\Sigma)(\varphi, \psi)$ is a natural deduction proof of $\varphi \vdash \psi$ (we use proof terms as in [4] to be definite); the axiom $\varphi \vdash \varphi$ is the identity morphism; and the composition of morphisms p and q proving $\varphi \vdash \psi$ and $\psi \vdash \chi$, respectively, is a special case of the cut rule. Note that the category laws, identity and associativity, impose some equalities on these natural deduction proofs. We do not use any rewrites, i.e., all orderings of morphisms are discrete.

We define a functor $\overline{\cdot} : \mathbb{O}rdCat \to \mathbb{CAT}$ by quotienting out the preorder; i.e., given a preorder-enriched category A, $\overline{A}$ is the quotient of A by the equivalence generated by the preorder on hom-sets. Similarly, $thin(\cdot) : \mathbb{O}rdCat \to \mathbb{CAT}$ is a functor that quotients all non-empty hom-sets to singletons.

In any institution, we have the usual notion of *semantic consequence*: For a set Φ of Σ-sentences, let $M \models_\Sigma \Phi$ denote $M \models_\Sigma \varphi$ for every $\varphi \in \Phi$. Then we say that that a Σ-sentence ψ is a consequence of Φ, and write $\Phi \models_\Sigma \psi$, iff $M \models_\Sigma \Phi$ implies $M \models_\Sigma \psi$ for every Σ-model M.

In an entailment institution, we can also define an *entailment* relation $\vdash_\Sigma$ between Σ-sentences as follows: $\varphi \vdash_\Sigma \psi$ iff there exists a morphism $\varphi \to \psi$ in $\mathsf{Pr}(\Sigma)$. In the presence of a binary connective $\times$ and a nullary connective $\top$, which semantically behave like conjunction and truth, we extend this relation to sets of hypotheses by putting $\Phi \vdash_\Sigma \psi$ if there exist $\varphi_1, \ldots, \varphi_n \in \Phi$ such that

$$\varphi_1 \times \cdots \times \varphi_n \vdash_\Sigma \psi$$

where we put $\varphi_1, \ldots, \varphi_n = \top$ if $n = 0$.

An entailment institution is *sound* if $\Phi \vdash_\Sigma \psi$ implies $\Phi \models_\Sigma \psi$. It is *strongly complete* if the converse implication holds, and *weakly complete* if the converse implication holds for the case $\Phi = \emptyset$. Clearly, soundness and strong completeness hold for $\mathbf{CPL}^{ND}$ if we choose the propositional conjunction and truth for $\times$ and $\top$.

In an institution $\mathcal{I}$, a *theory* is a pair $T = \langle\Sigma, \Gamma\rangle$, where $\Sigma \in |\mathbb{S}ign|$ and $\Gamma \subseteq \mathsf{Sen}(\Sigma)$. A *theory morphism* $\sigma\colon \langle\Sigma, \Gamma\rangle \to \langle\Sigma', \Gamma'\rangle$ is a signature morphism $\sigma\colon \Sigma \to \Sigma'$ for which $\Gamma' \models_{\Sigma'} \sigma(\Gamma)$, that is, σ maps axioms to consequences.

This defines a category $\mathbf{Th}$ of theories, and it is easy to extend Sen (or Pr) and Mod to $\mathbf{Th}$ by putting $\mathsf{Sen}(\langle\Sigma, \Gamma\rangle) = \mathsf{Sen}(\Sigma)$ and letting $\mathsf{Mod}(\langle\Sigma, \Gamma\rangle)$ be the full subcategory of $\mathsf{Mod}(\Sigma)$ induced by the class of those models M satisfying Γ.

Relationships between institutions are captured by several variants of institution morphisms and comorphisms. Here we use the following definition.

Definition 5. Given entailment institutions $\mathcal{I}$ and $\mathcal{J}$, a *comorphism* between them is a tuple (Φ, α, β) consisting of

- a functor $\Phi\colon \mathbb{S}ign^{\mathcal{I}} \to \mathbb{S}ign^{\mathcal{J}}$,
- a natural transformation $\alpha\colon \mathsf{Pr}^{\mathcal{I}} \to \mathsf{Pr}^{\mathcal{J}} \circ \Phi$,
- a natural transformation $\beta\colon \mathsf{Mod}^{\mathcal{J}} \circ \Phi^{op} \to \mathsf{Mod}^{\mathcal{I}}$

such that the following satisfaction condition holds for all $\Sigma \in |\mathbb{S}ign^{\mathcal{I}}|$, $M' \in |\mathsf{Mod}^{\mathcal{J}}(\Phi(\Sigma))|$ and $\varphi \in |\mathsf{Sen}^{\mathcal{I}}(\Sigma)|$:

$$M' \models^{\mathcal{J}}_{\Phi(\Sigma)} \alpha_\Sigma(\varphi) \ \text{ iff } \ \beta_\Sigma(M') \models^{\mathcal{I}}_{\Sigma} \varphi.$$

Example 6. To understand the intuition behind this definition, assume an entailment institution $\mathbf{CFOL}$ of first-order logic that has more complex signatures and models, and more formulas and proofs than $\mathbf{CPL}^{ND}$. Then the natural inclusion of propositional into first-order logic is formalized as a comorphism from $\mathbf{CPL}^{ND}$ to $\mathbf{CFOL}$. Every $\mathbf{CPL}^{ND}$ signature Σ is mapped to the first-order signature without function symbols and containing only nullary predicate symbols, namely those in Σ. The sentence and proof translation α_Σ is an inclusion because every propositional formula and every propositional proof over Σ is also a first-order formula or proof over $\Phi(\Sigma)$ (assuming that the proofs in $\mathbf{CFOL}$ are defined by adding rules to the rules used for $\mathbf{CPL}^{ND}$ and not by some entirely different proof system). The model translation β goes in the opposite direction: Every $\Phi(\Sigma)$-model in $\mathbf{CFOL}$ is reduced to Σ-model in $\mathbf{CPL}^{ND}$ by forgetting the universe. The satisfaction condition, intuitively, requires that truth and consequence are preserved under a comorphism.

Clearly, comorphisms are more interesting and powerful if they express less trivial relationships between two institutions.

Together with the obvious composition and identities, this defines a category[5] $\mathbb{C}oIns$ of entailment institutions and comorphisms.

3 Categorical Logic

Lambek and Scott [19] study categorical logic by introducing *deductive systems* as directed graphs with a composition structure, and later impose the usual

[5] Of course, this category lives in a higher set-theoretic universe. We omit the foundational issues here.

axioms for (cartesian, cartesian closed, bicartesian closed) categories on these. Objects in categories serve as 'types' for morphisms, hence the 'propositions as types' paradigm becomes 'propositions as objects'.

We will a use formal meta-language to formalize categorical logic, namely an institution $\mathcal{DFOL}$ that extends many-sorted first-order logic with dependent sorts. Since we only need one dependent sort constructor, namely for morphisms, we refer to [26] for a rigorous introduction and resort to an example.

The main $\mathcal{DFOL}$-signature we will use is (For simplicity, we omit the $\mathcal{DFOL}$-symbol $Univ$.)

$Ob : sort.\ Mor : Ob \times Ob \to sort.$
$id : \Pi A : Ob.\ Mor(A, A).$
$; : \Pi A, B, C : Ob.\ Mor(A, B) \to Mor(B, C) \to Mor(A, C).$
$\rightsquigarrow : \Pi A, B : Ob.\ Mor(A, B) \to Mor(A, B) \to Form.$
$Des : Ob.$

Here, Ob is a base sort, and Mor is a dependent sort, i.e., for all terms A, B of sort Ob, $Mor(A, B)$ is a sort. id is a function symbol that takes an argument of type Ob and returns a morphism of a sort that depends on the argument. Similarly ; takes three arguments of sort Ob, say A, B and C, and then two arguments of the sorts $Mor(A, B)$ and $Mor(B, C)$ and returns a term of sort $Mor(A, C)$. $\rightsquigarrow$ is a predicate symbol: It takes two arguments of sort Ob, say A and B, and two arguments of sort $Mor(A, B)$ and returns a formula. Des is an object constant.

The intended semantics is that Ob is the set of objects of a small category (via the Curry-Howard interpretation: formulas of a categorical logic), $Mor(A, B)$ is the set of morphisms (proofs) from A to B, $id(A)$ is the identity (self-proof) of A, $;(A, B, C, f, g)$ is the composition of $f : Mor(A, B)$ and $g : Mor(B, C)$ (the proof of C from A obtained from applying cut with B), and $\rightsquigarrow (A, B, f, g)$ expresses that $f \rightsquigarrow g$ in $Mor(A, B)$ (the reducibility of f to g). Des singles out a certain object (the minimal designated truth value). For simplicity, we will write $f; g$ instead of $;(A, B, C, f, g)$ and $f \rightsquigarrow g$ instead of $\rightsquigarrow (A, B, f, g)$.

The sentences of $\mathcal{DFOL}$ use the symbols $\wedge$, $\Longrightarrow$, $==$ and $\forall$ with the usual semantics and notation, where $==$ is overloaded for all sorts, and $\forall$ binds a sorted variable. (This is a slight simplification of the formal definitions in [26].) Formulas and terms do not contain free variables unless mentioned otherwise. We define the theory $CatLog$ by adding to the above signature the axioms

$$\begin{array}{l} \forall A, B, C, D : Ob\ \ \forall f, f', f'' : Mor(A, B)\ \ \forall g, g' : Mor(B, C)\ \ \forall h : Mor(C, D) \\ \quad id(A); f == f \\ \quad h; id(D) == h \\ \quad (f; g); h == f; (g; h) \\ \quad f \rightsquigarrow f \\ \quad f \rightsquigarrow f' \wedge f' \rightsquigarrow f'' \Longrightarrow f \rightsquigarrow f'' \\ \quad f \rightsquigarrow f' \wedge g \rightsquigarrow g' \Longrightarrow f; g \rightsquigarrow f'; g' \end{array}$$

The model category $Mod^{\mathcal{DFOL}}(CatLog)$ is essentially $\mathbb{O}rdCat$. Precisely, a model is an element of $\mathbb{O}rdCat$ with one distinguished object interpreting Des.

Definition 7. A *categorical logic* $\mathcal{L}$ is an extension of *CatLog* with function symbols and axioms such that

- axioms are in Horn form
- function symbols that return objects do not take morphisms as arguments
- there are congruence axioms for $\rightsquigarrow$, i.e., for every new function symbol of the form

  $$c : \Pi \boldsymbol{A} : Ob.\ Mor(S_1, T_1) \to \ldots \to Mor(S_n, T_n) \to Mor(S, T)$$

  where $\boldsymbol{A}$ abbreviates $A_1, \ldots, A_m$ and S_i, T_i, S, T are terms of the sort Ob with free variables $\boldsymbol{A} : Ob$, there is an axiom

  $$\forall \boldsymbol{A} : Ob\ \ \forall f_1, f_1' : Mor(S_1, T_1) \ldots \forall f_n, f_n' : Mor(S_n, T_n)$$
  $$(f_1 \rightsquigarrow f_1' \wedge \ldots \wedge f_n \rightsquigarrow f_n') \implies c(\boldsymbol{A}, f_1, \ldots, f_n) \rightsquigarrow c(\boldsymbol{A}, f_1', \ldots, f_n')$$

The category of categorical logics, denoted by $\mathbb{C}atLog$, has such theories $\mathcal{L}$ as objects and theory morphisms that are the identity for the symbols of *CatLog* as morphisms. (More generally, we could use the slice category of $\mathcal{DFOL}$-theories with domain *CatLog*.)

Then for any categorical logic $\mathcal{L}$, an element of $\mathsf{Mod}^{\mathcal{DFOL}}(\mathcal{L})$ consists of an element of $\mathbb{O}rdCat$ along with interpretations for the symbol *Des* and the added function symbols. $\mathsf{Mod}^{\mathcal{DFOL}}(\mathcal{L})$-morphisms are functors in $\mathbb{O}rdCat$ that preserve the interpretation of *Des* and that commute with the added function symbols. We call this category $\mathbb{C}(\mathcal{L})$, its elements $\mathcal{L}$-categories, and its morphisms $\mathcal{L}$-functors. As usual, we use the same symbol for a model M and its underlying category; the preorders on all hom-sets are denoted by $\rightsquigarrow^M$; and the interpretation of a function symbol c is denoted by c^M. Then the usual notion of (small) enriched categories becomes a special case of this construction.

We have the following result:

Proposition 8. *For every categorical logic $\mathcal{L}$ and every set X of sorted variables of the sort Ob, there is a free model $T_{\mathcal{L}}(X) \in |\mathbb{C}(\mathcal{L})|$, i.e., for every $A \in |\mathbb{C}(\mathcal{L})|$ and every mapping $m\colon X \to |A|$ there is a unique extension $\overline{m}$ of m to an $\mathcal{L}$-functor $T_{\mathcal{L}}(X) \to A$.*

Proof. This a special case of the result in [26]. In particular, the restrictions in Def. 7 are used. □

In fact, the set X may contain arbitrary sorted variables, but we will only use this more general result in the proof of Theorem 13. The objects and morphisms of $T_{\mathcal{L}}(X)$ are equivalence classes of terms. For simplicity, we use every term as an abbreviation for its equivalence class in $T_{\mathcal{L}}(X)$.

Example 9 (Cartesian Categories). The categorical logic *Cartesian* arises by extending *CatLog* with declarations

$\top : Ob$
$\times : Ob \to Ob \to Ob$
$! : \Pi A : Ob.\ Mor(A, \top).$

$\times : \mathit{\Pi} A, B, C : Ob.\ Mor(A, B) \to Mor(A, C) \to Mor(A, B \times C)$
$\pi_1 : \mathit{\Pi} A, B, C : Ob.\ Mor(A, B \times C) \to Mor(A, B)$
$\pi_2 : \mathit{\Pi} A, B, C : Ob.\ Mor(A, B \times C) \to Mor(A, C)$
and axioms
$f; (g \times h) \rightsquigarrow (f; g) \times (f; h)$

as well as congruence axioms and axioms that make $\top$ a terminal object and $\times$ a binary product with projections π_1 and π_2. Note that we overload $\times$ in accordance with the usual notation for functors. Logically, this corresponds to adding "true" ($\top$) and conjunction ($\times$). The rewrite axiom is a proof normalization step to eliminate cut (i.e., ;).

Note that we have some freedom in specifying the axioms. For example, we can either add the axiom $(f \times g); \pi_1 == f$ (omitting the universal quantifiers), which identifies the two proof terms and hence, regards their difference as merely bureaucratic and caused by the needs of syntactic representation. Or we can specify the rewrite rule $(f \times g); \pi_1 \rightsquigarrow f$, which keeps $(f \times g); \pi_1$ and f distinct. In the sequel, we always assume that universal properties are specified using rewrite rules.

Extensions $\mathcal{L}$ of *Cartesian* are called cartesian. If $\mathcal{L}$ has the axiom $Des == \top$, $\mathcal{L}$ is called $\top$-cartesian. Note that in every $\top$-cartesian categorical logics, we can define the entailment relation $\mathit{\Phi} \vdash_{\mathit{\Sigma}} \psi$ for sets of sentences $\mathit{\Phi}$.

4 The Institutional Curry-Howard-Tait Construction

Now we construct an entailment institution out of a categorical logic, thereby following the Curry-Howard-Tait isomorphism paradigm. First, given a categorical logic $\mathcal{L}$, we define two theory extensions: Let $\overline{\mathcal{L}}$ be the extension of $\mathcal{L}$ with the axiom

$$\forall A, B : Ob\ \ \forall f, g : Mor(A, B)\ (f \rightsquigarrow g \Longrightarrow f == g),$$

which, semantically, quotients out the preorder in the $\mathcal{L}$-categories, and let $\mathcal{L}_{thin}$ be the extension of $\overline{\mathcal{L}}$ with the axiom

$$\forall A, B : Ob\ \ \forall f, g : Mor(A, B)\ f == g,$$

which, semantically, quotients the $\mathcal{L}$-categories to preorders. Let $\overline{\mathbb{C}(\mathcal{L})} = \mathbb{C}(\overline{\mathcal{L}})$ and $\mathbb{C}_{thin}(\mathcal{L}) = \mathbb{C}(\mathcal{L}_{thin})$.

Definition 10. Given a categorical logic $\mathcal{L}$, the entailment institution $\mathcal{I}(\mathcal{L}) = (\mathbb{S}ign, \mathsf{Pr}, \mathsf{Mod}, \models)$ is defined by:

- $\mathbb{S}ign$ is the category of sets (seen as sets of propositional variables).
- Pr is the universal functor of Prop. 8. Explicitly, $\mathsf{Pr}(\mathit{\Sigma}) = T_{\mathcal{L}}(\mathit{\Sigma})$ for a signature $\mathit{\Sigma}$, and for a signature morphism (that is, a function) $\sigma \colon \mathit{\Sigma} \to \mathit{\Sigma}'$, $\mathsf{Pr}(\sigma)$ is the unique extension of σ to an $\mathcal{L}$-functor from $T_{\mathcal{L}}(\mathit{\Sigma})$ to $T_{\mathcal{L}}(\mathit{\Sigma}')$.
- Mod is the lax comma category functor $(\mathsf{Pr}(\cdot) \downarrow \overline{\mathbb{C}(\mathcal{L})})$. Explicitly, Mod assigns to a signature $\mathit{\Sigma}$ a category $\mathsf{Mod}(\mathit{\Sigma})$ such that

  - objects are pairs (A, m) where $A \in |\overline{\mathbb{C}(\mathcal{L})}|$ and m is a valuation $m\colon \Sigma \to |A|$ of the propositional variables to A-objects,[6]
  - model morphisms from (A, m) to (A', m') are pairs (F, μ) where $F\colon A \to A'$ is an $\mathcal{L}$-functor and μ is a family of A'-morphisms over $a \in \Sigma$ such that $\mu(a)\colon F(m(a)) \to m'(a)$,

  and for a signature morphism $\sigma\colon \Sigma \to \Sigma'$, the model reduct functor $\mathsf{Mod}(\sigma)$ is given by composition: $\mathsf{Mod}(\sigma)(A, m) = (A, m \circ \sigma)$ for models and $\mathsf{Mod}(\sigma)(F, \mu) = (F, \mu \circ \sigma)$ for model morphisms.
- Satisfaction is defined by: $(A, m) \models_\Sigma \varphi$ iff $A(Des^A, \overline{m}(\varphi))$ is inhabited.

For the proof of the satisfaction condition, see Prop. 11 below.

For a signature Σ, the Curry-Howard-Tait correspondence then takes the following shape:

Propositions as types/objects Sentences are the objects of $\mathsf{Pr}(\Sigma)$, i.e., sentences are $\mathcal{L}$-terms of sort Ob with propositional variables from Σ.

Proofs as terms Proofs are the morphisms of $\mathsf{Pr}(\Sigma)$. That is, a Σ-proof between sentences φ and ψ is simply an equivalence class of $\mathcal{L}$-terms of sort $Mor(\varphi, \psi)$. If the only equations in $\mathcal{L}$ are those of $CatLog$, this means that Σ-proofs are strings of composable composition-free proof terms.

Proof reduction as morphism ordering The reducibility of proofs is given by the $\rightsquigarrow$ predicate in $T_{\mathcal{L}}(\Sigma)$, which is a preorder on morphisms (which is preserved under composition).

Categorical models A Σ-model is determined by a category A in $\overline{\mathbb{C}(\mathcal{L})}$ and a valuation of the propositional variables into A. Proof reduction is modeled by the interpretation of $\rightsquigarrow$.

Satisfaction via designated truth values For a Σ-model $M = (A, m)$, the preorder $thin(A)$ gives the truth values of M. Then the set of designated truth values is the right segment of $thin(A)$ generated by Des^M. (Defining more complex sets of designated truth values is possible by making Des a predicate on objects instead of a constant, as in [2].)

In the remainder of this section, we derive some crucial properties of this construction.

Proposition 11 (Soundness). *For any categorical logic $\mathcal{L}$, $\mathcal{I}(\mathcal{L})$ is an entailment institution. If $\mathcal{L}$ has operations $\times_n : Mor(Des, A_1) \to \ldots \to Mor(Des, A_n) \to Mor(Des, A_1 \times \ldots \times A_n)$, $\mathcal{I}(\mathcal{L})$ is sound with respect to $\times$ and Des.*

Proof. The satisfaction condition is shown as follows. Let $\sigma\colon \Sigma \to \Sigma'$ be a morphism, let φ be a Σ-sentence, and let $M' = (A', m')$ be a Σ'-model. Then $(A', m')|_\sigma \models_\Sigma \varphi$ iff $A'(Des^M, \overline{m' \circ \sigma}(\varphi)) \neq \emptyset$ iff $A'(Des^{M'}, \overline{m'}(\sigma(\varphi))) \neq \emptyset$ iff $(A', m') \models_{\Sigma'} \sigma(\varphi)$.

To prove soundness, assume $\Phi \vdash_\Sigma \psi$, i.e., $\varphi_1 \times \ldots \times \varphi_n \vdash_\Sigma \psi$. That means there is a morphism $p : Mor(\varphi_1 \times \ldots \times \varphi_n, \psi)$ in $\mathsf{Pr}(\Sigma)$. Then there is a morphism

[6] Remember that such a valuation uniquely determines an $\mathcal{L}$-functor $\overline{m}$ from $Pr(\Sigma)$ to A.

$\overline{m}(p) \in A(\overline{m}(\varphi_1) \times^M \ldots \times^M \overline{m}(\varphi_n), \overline{m}(\psi))$ in every Σ-model $M = (A, m)$. Thus if $A(Des^M, \overline{m}(\varphi_i))$ is inhabited by some q_i for every i, then $A(Des^M, \overline{m}(\psi))$ is inhabited by $\overline{m}(p) \circ \times_n^M(q_1, \ldots, q_n)$ (putting $\times_0 = id(Des)$). Therefore $\Phi \models \psi$. □

In particular, $\mathcal{I}(\mathcal{L})$ is sound for every $\top$-cartesian $\mathcal{L}$.

It is important to recognize that, unlike the categories A in Σ-models (A, m), the proof categories $\mathsf{Pr}(\Sigma)$ are generally *not* elements of $\mathbb{C}(\mathcal{L})$. This avoids the problem stated in [19] that for classical propositional logic, the categorical semantics of proofs collapses: All classical bicartesian closed categories are partial orders (and thus boolean algebras). If one does not wish to distinguish between rewritable proofs, one can collapse the proof categories by using $\overline{\mathcal{I}(\mathcal{L})} := \mathcal{I}(\overline{\mathcal{L}})$. Proof irrelevance is obtained by using $\mathcal{I}_{thin}(\mathcal{L}) := \mathcal{I}(\mathcal{L}_{thin})$.

Definition 12 (Deduction Theorem). For a $\top$-cartesian categorical logic $\mathcal{L}$, we say that the *deduction theorem* holds in $\mathcal{L}$ if for every Σ-proof term $p : Mor(\varphi, \psi)$ in free variables X, $x{:}Mor(\top, \chi)$, there is a Σ-proof term $\kappa x.p$ of type $Mor(\varphi \times \chi, \psi)$ in free variables X.

Theorem 13 (Completeness). *For every categorical logic $\mathcal{L}$, $\mathcal{I}(\mathcal{L})$ is weakly complete. And if the deduction theorem holds for $\mathcal{L}$, then $\mathcal{I}(\mathcal{L})$ is strongly complete.*

Proof. To prove the first result, assume $\emptyset \models_\Sigma \psi$, then in particular $(\overline{\mathsf{Pr}(\Sigma)}, \pi) \models_\Sigma \psi$ where π is the canonical projection. This implies $\mathsf{Pr}(\Sigma)(Des, \psi) \neq \emptyset$. And that is the definiton of $\emptyset \vdash_\Sigma \psi$.

To prove the second result, assume $\Phi \models_\Sigma \psi$. Let $F \in |\mathsf{Mod}^{\mathcal{DFOL}}(\mathcal{L})|$ be the free $\mathcal{L}$-category existing by Prop. 8 over the following sorted variables: variables $v : Ob$ for every $v \in \Sigma$ and variables $x_\varphi : Mor(\top, \varphi)$ for all $\varphi \in \Phi$. Then $F \models_\Sigma \varphi$ for all $\varphi \in \Phi$. Then by the assumption of the theorem, we have $F \models_\Sigma \psi$. Then, since F is a free term model differing from $\mathsf{Pr}(\Sigma)$ only by having more variables, there must be a $\mathsf{Pr}(\Sigma)$-term $p(x_\varphi, \ldots) : Mor(\top, \psi)$ in free variables x_φ. Clearly, p can only refer to finitely many x_φ. Thus repeated application of the deduction theorem yields a closed $\mathsf{Pr}(\Sigma)$-term of type $Mor(\top \times \varphi_1 \times \ldots \times \varphi_n, \psi)$, and applying a projection yields the definition of $\Phi \vdash \psi$. □

Proposition 14. *The deduction theorem holds in any $\top$-cartesian categorical logic provided that newly introduced function symbols that return morphisms do not take morphisms as arguments (that is, there are* no new logical rules*).*

Proof. See Proposition I.2.1 of [19]. The addition of operations adhering to the above restriction does not destroy the induction proof given there. □

Example 15. Consider the minimal $\top$-cartesian categorical logic *Cartesian′* The institution $\mathcal{I}(Cartesian')$ is described as follows. Signatures are sets of propositional variables. Sentences are the corresponding fragment of propositional logic. A model consists of a category together with an interpretation of the propositional variables as objects in this category. The conjunction is interpreted by a product, and truth by a terminal element. Evaluation of sentences is just term evaluation in the category. The designated truth values are $\top$ and everything provable from it.

If $p\colon C \to A$ and $q\colon C \to B$ are proofs in a model $M = (A, m)$, they can be combined to $p \times^M q\colon C \to A \times^M B$, and $(p \times^M q);^M \pi_1^M\colon C \to A$ is another proof. The rewriting structure gives us $(p \times^M q);^M \pi_1^M \rightsquigarrow p$. In the institution $\overline{\mathcal{I}(IProp)}$, these proofs are identified.

We arrive at cartesian categories ([19]) if we quotient out rewrites, i.e., use $\mathcal{I}_{thin}(Cartesian')$.

In the context of the Curry-Howard correspondence, the cut rule corresponds to the composition of morphisms. Note that the more common, more complex forms of the cut rule can be derived from this version and other elementary rules (see for example the extended version of [13]). Then *cut elimination* means that the composition operation can be eliminated from proof terms.

Definition 16. A categorical logic $\mathcal{L}$ admits cut if for all signatures Σ and all objects $A, B \in |T_{\mathcal{L}}(\Sigma)|$, for every Σ-proof term $p : Mor(A, B)$ there is a Σ-proof term $p' : Mor(A, B)$ such that p' does not contain the function symbol ;. $\mathcal{L}$ has cut elimination if, in addition, $p \rightsquigarrow p'$ holds in $T_{\mathcal{L}}(\Sigma)$.

Both cut admissibility and cut elimination need to be established independently for every categorical logic: Minor changes in the specification can destroy these properties, or require redoing large portions of their proofs. While some of the above examples have cut admissibility, additional rewrites are necessary to establish cut elimination. These additional rewrites correspond to the various cases and subcases of the induction step of a constructive cut elimination proof.

Proposition 17. *Cartesian has cut elimination.*

Proof. The proof proceeds by nested term inductions on f and g in $f; g$. This result can also be found in [12].

Exactness of institutions is a property important for modular specifications and proofs [29]:

Definition 18. An institution is said to be *semi-exact*, if Mod takes push-outs in $\mathbb{S}ign$ to pullbacks in $\mathbb{C}at$.

Since the model reduct functor $\mathsf{Mod}(\sigma)$ is defined by composition, $\mathcal{I}(\mathcal{L})$ is semi-exact:

Proposition 19. *For any categorical logic $\mathcal{L}$, $\mathcal{I}(\mathcal{L})$ is semi-exact.*

Definition 20. An institution is said to be *(weakly) liberal*, if the reduct functor of each *theory morphism* $\sigma\colon (\Sigma_1, \Psi_1) \to (\Sigma_2, \Psi_2)$ has a (weak) left adjoint. A functor F is a weak left adjoint to U via unit $\eta\colon Id \to UF$, if any morphism $f\colon X \to U\,A$ factors (not necessarily uniquely) as $U\,g \circ \eta_X$ for some $g\colon F\,X \to A$.

$\mathcal{I}(IProp)$ and $\mathcal{I}(Prop)$ are not liberal: the theory $(\{A, B\}, \{A + B\})$ (viewed as extension of the empty theory) has no free model. However, we have

Proposition 21. *For any categorical logic $\mathcal{L}$ with truth, $\mathcal{I}(\mathcal{L})$ is weakly liberal.*

Proposition 22. *The construction $\mathcal{I}(-)$ is functorial, i.e., it can be extended to a functor $\mathcal{I}(-)\colon \mathbb{C}atLog \to \mathbb{C}oIns$.*

Proof. Given two categorical logics $\mathcal{L}_1$ and $\mathcal{L}_2$ and a $\mathcal{DFOL}$ theory morphism $\sigma\colon \mathcal{L}_1 \to \mathcal{L}_2$, we can construct an institution comorphism $(\Phi, \alpha, \beta)\colon \mathcal{I}(\mathcal{L}_1) \to \mathcal{I}(\mathcal{L}_2)$ as follows:

- Φ is the identity functor in the category $\mathbb{S}et$.
- To define $\alpha_\Sigma :: \mathsf{Pr}^{\mathcal{L}_1}(\Sigma) \to \mathsf{Pr}^{\mathcal{L}_2}(\Phi(\Sigma))$, first note that $\Phi(\Sigma) = \Sigma$, and remember that the objects and morphisms of $\mathsf{Pr}^{\mathcal{L}_i}$ are equivalence classes of terms over $\mathcal{L}_i$. But since the $\mathsf{Sen}^{\mathcal{DFOL}}(\sigma)$ images of $\mathcal{L}_1$-axioms are consequences of the $\mathcal{L}_2$-axioms, all elements of such an equivalence class of $\mathcal{L}_1$ are mapped to the same equivalence class in $\mathcal{L}_2$. Therefore, α_Σ is well-defined as the translation of $\mathcal{L}_1$-terms over Σ to $\mathcal{L}_2$-terms over Σ, in the institution $\mathcal{DFOL}$, induced by σ.
- β is induced by the model reduction functor in $\mathcal{DFOL}$.
- The proof of the satisfaction condition is straightforward.

□

Proposition 23. *Let $\mathcal{L}$ be a categorical logic. There are entailment institution comorphisms from $\mathcal{I}(\mathcal{L})$ to $\overline{\mathcal{I}(\mathcal{L})}$ and from $\overline{\mathcal{I}(\mathcal{L})}$ to $\mathcal{I}_{thin}(\mathcal{L})$, and institution comorphisms in the opposite directions. In particular, semantic consequence is the same in the three institutions.*

Proof. The entailment institution comorphisms are given by Prop. 22. For the other direction, the translation of signatures and sentences is the identity, a model $(A, m) \in |\mathsf{Mod}^{\mathcal{I}(\mathcal{L})}(\Sigma)|$ is translated to $(\overline{A}, m)$, $(B, m) \in |\mathsf{Mod}^{\overline{\mathcal{I}(\mathcal{L})}}(\Sigma)|$ is translated to $(thin(B), m)$. Since satisfaction is defined by the existence of certain morphisms, the satisfaction condition for these comorphisms is straightforward. Coincidence of semantic consequence then follows easily. □

5 Examples

Our framework permits the use of $\mathcal{DFOL}$-theory morphisms to obtain a precise semantics for parametric and modular specifications. Fig. 2 gives a hierarchy of several examples of modular categorical logic specifications. Nodes are categorical logics, the solid arrows are $\mathcal{DFOL}$ theory inclusions, and all rectangles are push-outs. For example, we obtain classical *S4* as a push-out of classical logic *Prop* and intuitionistic modal logic *IS4* over intuitionistic logic *IProp*. Furthermore, we can reuse a specification for monoidal comonads in the specification of both modal and linear logic: The dotted arrows are non-trivial theory morphisms. The precise specifications can be found at [27]. Since $\mathcal{DFOL}$ is realized as a signature in the logical framework LF ([18]), all specifications are given as Twelf ([25]) files that can be type-checked automatically.

Example 24 (Intuitionistic logic). *Cocartesian* categories specify disjunction + and falsity ⊥. Extending *Cartesian* with implication $\longrightarrow$ yields cartesian closed categories *CartClosed*. And intuitionistic logic *IProp* arises as the push-out of those two thus arriving at bicartesian closed categories ([19]). Models interpret implication as an exponential, disjunction as a coproduct, and falsity as an initial object. We can add negation to *Prop* as a defined operation by putting $-A := A \longrightarrow \bot$. Furthermore, we add the axiom $Des == \top$, which expresses that $\top$ is the minimal designated truth value. Since $\top$ is also a terminal element, the models have (up to isomorphism) one greatest truth value.

Example 25 (Classical logic). Several possibilities exist to extend *IProp* to classical logic (see e.g., [13]). We define *Prop* be adding an operation $tnd : \ \Pi A : Ob.\ Mor(\top, A + -A)$. Again there is flexibility as to whether proofs from A to A via $- - A$ are reducible or even equal to $id(A)$.

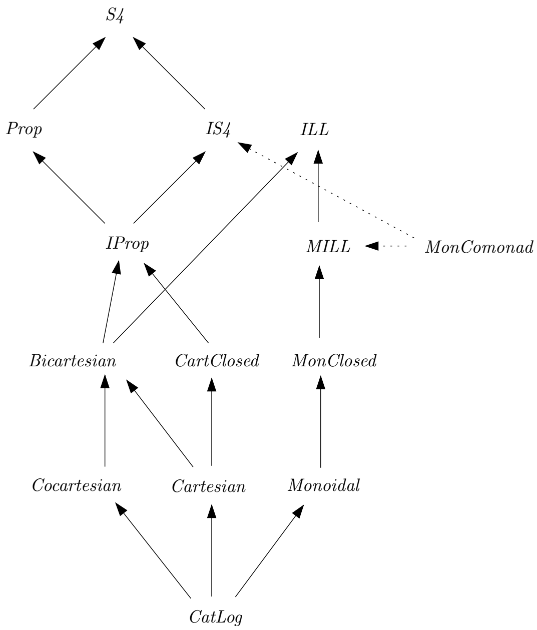

Fig. 2. Graph of Logics

Example 26 (Intuitionistic modal logic). We extend *IProp* by adding operations $\Box, \Diamond : Ob \rightarrow Ob$ and other operations on morphisms modeling the necessity and possibility operators from constructive modal S4, as defined by [4]. The necessity modality $\Box$ is interpreted as a monoidal comonad, while the possibility modality $\Diamond$ is interpreted as a monad, which is strong relative to the necessity comonad.

Example 27 (Linear logic). For a different kind of categorical logic, which does not build up from *CartClosed*, but simply from *CatLog*, consider multiplicative intuitionistic linear logic *MILL* as in [24]. Multiplicative conjunction and linear implication are interpreted as a symmetric monoidal closed category.

The of-course operator ! is interpreted as a comonad, and each object $!A$ is equipped with a comonoid structure such that the comonoid maps are also coalgebra maps. We add the axiom $Des == I$, i.e., all truth values greater than the multiplicative truth I are designated. If we take the push-out of *MILL* and *Bicartesian* over *CatLog*, we obtain intuitionistic linear logic *ILL* where the bicartesian structure corresponds to the additive connectives. Note that in *ILL*, we have a morphism from I to $\top$, i.e., $\top$ is also a designated truth value, and in non-trivial models $\top$ is strictly greater than I.

The first three examples are $\top$-cartesian. And for linear logic, we can give the operations $\times_n$ of Prop. 11 explicitly. Thus all entailment institutions induced by these examples are sound. Prop. 14 permits the introduction of falsity and disjunction, which yields the deduction theorem for *IProp* and *Prop*. Thus *IProp* and *Prop* are strongly complete. The deduction theorem does not hold for modal logic: For every proof term $x : Mor(\top, A)$ there is a proof term $\Box x : Mor(\Box\top, \Box A)$, which, together with the canonical morphism $m\_\top : Mor(\top, \Box\top)$, yields a morphism $m\_\top; \Box x : Mor(\top, \Box A)$, but there is in general no closed proof term of the sort $Mor(A, \Box A)$. A similar argument disproves the deduction theorem for linear logic. These entailment institutions are only weakly complete.

For modal and linear logic, it is not trivial to define the rewriting structure in a way that permits normalization. Therefore, we give most properties for these as equalities instead of as rewrites.

6 The Curry-Howard-Tait isomorphism

$\mathbf{CPL}^{ND}$ models are valuations into the boolean algebra $\{0, 1\}$ whereas $\mathcal{I}(Prop)$-models are valuations into arbitrary boolean algebras. This difference is not trivial. In a boolean algebra, regarded as a *Prop*-category, an object has a global element iff it is terminal (i.e., equal to the 1 of the boolean algebra). There is no mapping from valuations into arbitrary boolean algebras to valuations into *Bool* that preserves and reflects truth (i.e. terminalhood) of formulas: Let $\nu \colon \{p, q\} \to Bool^2$ map p to $(\top, \bot)$ and q to $(\bot, \top)$. Assume that $\nu' \colon \{p, q\} \to Bool$ makes true the same formulas as ν. Then $\nu'(p)$ and $\nu'(q)$ must both be $\bot$, because neither $\nu(p)$ nor $\nu(q)$ is terminal. However, $\nu(p \vee q)$ *is* terminal, while $\nu'(p \vee q)$ is not, a contradiction. Hence, the model translation of the comorphism from $\mathcal{I}(\mathcal{L})$ to $\overline{\mathcal{I}(\mathcal{L})}$ introduced in Prop. 23 cannot be reversed.

A related observation is that $\mathbf{CPL}^{ND}$ and $\mathcal{I}(Prop)$ differ in their behavior of disjunction. While in $\mathbf{CPL}^{ND}$, a model M satisfies a disjunction iff it satisfies either of the disjuncts (which is called "model-theoretic disjunction" in [23]), this is not the case for $\mathcal{I}(Prop)$: Consider the weakly free model over the theory $(\{A, B\}, \{A + B\})$, existing by Prop. 21.

However, from the point of view of semantic consequence, we can restrict ourselves to valuations into *Bool*: Semantic consequence in $\mathbf{CPL}^{ND}$ and $\mathcal{I}(Prop)$ coincide. We turn this observation into a general notion:

Definition 28. Let $\mathcal{L}$ be a categorical logic, and let $A \in |\overline{\mathbb{C}(\mathcal{L})}|$. Let $\mathcal{I}^A(\mathcal{L})$ denote the institution obtained from $\mathcal{I}(\mathcal{L})$ by allowing only those Σ-models (M, m) for which $M = A$. We denote satisfaction and semantic consequence in this institution with the superscript A. The category A is called a *weak truth value object* for $\mathcal{L}$ if $\varphi \models^A \psi$ implies $\varphi \models \psi$ for all sentences φ, ψ, and a *strong truth value object* if $\Phi \models^A \psi$ implies $\Phi \models \psi$ for all sets Φ of sentences and sentences ψ.

Proposition 29. *Let $\mathcal{L}$ be a categorical logic extending the theory of cartesian closed categories (i.e., minimal intuitionistic logic with $\top$, $\times$, $\to$), and let $A \in |\overline{\mathbb{C}(\mathcal{L})}|$ be thin and skeletal, hence essentially being a partial ordering $\leq$. Then A is a strong truth value object for $\mathcal{L}$ iff the following condition holds.*

($*$) *Let $B \in |\overline{\mathbb{C}(\mathcal{L})}|$, and let $a, b \in |B|$. If $f(a) \leq f(b)$ for each $\mathcal{L}$-morphism $f : B \to A$, then* $\hom_B(a, b) \neq \emptyset$.

(If B is a thin category, i.e. a preorder, then condition ($*$) states that the source of all morphisms from B into A is jointly order-reflecting.)

Proof. Assume that ($*$) holds, let $\Phi \models^A \psi$, and let $(B, m) \models \Phi$. Then for every $f : B \to A$, $(A, f \circ m) \models \Phi$ and hence $(A, f \circ m) \models \psi$, i.e., $\top \leq f(m(\psi))$. By ($*$), it follows that $B(\top, m(\psi)) \neq \emptyset$, i.e., $(B, m) \models \psi$.

Conversely, let A be a strong truth value object, let $B \in |\overline{\mathbb{C}(\mathcal{L})}|$, and let $a, b \in |B|$ such that $A(f(a), f(b)) \neq \emptyset$ for all $f : B \to A$. Let $\Sigma = |B|$, and let Φ be the theory (i.e., the set of valid formulas) of the Σ-model (B, η) where $\eta \colon \Sigma \to |B|$ is the inclusion. Since morphisms $f : B \to A$ are then just the Σ-valuations in A that validate Φ (because by the assumption on $\mathcal{L}$, formulas χ_1 and χ_2 denote isomorphic objects of B iff $(B, \eta) \models \chi_1 \leftrightarrow \chi_2$), the premise says that $\Phi \models^A a \to b$ (note that the assumption on $\mathcal{L}$ implies that an implication holds iff there exists a morphism between the corresponding objects). Thus, $\Phi \models a \to b$, so that $a \to b$ holds in (B, η); i.e., $B(a, b) \neq \emptyset$ as claimed. □

Example 30. *Bool*, seen as a *Prop*-category, is a strong truth value object for *Prop*. For any dense-in-itself metric space X, the Heyting algebra $\mathfrak{O}(X)$ of open sets in X (qua bicartesian closed category) is a weak truth value object for *IProp* [28].

One can use the above result to show that no strong truth value object for *IProp* exists. To see this, assume that A is a Heyting algebra satisfying condition ($*$) of Proposition 29. Let $\alpha > |A|$ be a cardinal, and let B be the ordinal $\alpha + 2$, considered as a Heyting algebra. The two largest elements of $\alpha + 2$ are α and $\alpha + 1$. We show that for every morphism $f : B \to A$, $f(\alpha) = f(\alpha + 1)$; then by ($*$), $\alpha + 1 \leq \alpha$, contradiction. For cardinality reasons, we have $a < b$ in B such that $f(a) = f(b)$. Then

$$f(a) = f(b \to a) = (f(b) \to f(a)) = \top = f(\alpha + 1),$$

and since $a \leq \alpha$, we obtain $f(\alpha) = f(\alpha + 1)$ as claimed.

A consequence of this observation is that, for every dense-in-itself metric space X, the consequence relation on $\mathfrak{O}(X)$ is non-compact, since otherwise, the weak truth value object $\mathfrak{O}(X)$ would also be a strong truth value object.

We are now ready to formalize the Curry-Howard-Tait correspondence in terms of entailment institutions. We cannot expect this to be constructed in an institution-independent manner. Rather, for some given propositional categorical logic $\mathcal{L}$, we can relate $\mathcal{I}(\mathcal{L})$ to a well-known institution.

Definition 31. For two entailment institutions $\mathcal{I}$ and $\mathcal{I}'$, a *semi-correspondence* from $\mathcal{I}$ to $\mathcal{I}'$ is a pair of two entailment institution comorphisms $\mu^1 = (\Phi^1, \alpha^1, \beta^1) : \mathcal{I} \to \mathcal{I}'$ and $\mu^2 = (\Phi^2, \alpha^2, \beta^2) : \mathcal{I}' \to \mathcal{I}$ that are mutually inverse isomorphisms, except for the proof structure: The α^i need not be inverses of each other on morphisms and need not preserve the reduction ordering. We only require that $\alpha^2_{\Phi^1(\Sigma)}(\alpha^1_\Sigma(p)) \rightsquigarrow p$, and impose no condition on $\alpha^1_{\Phi^2(\Sigma)}(\alpha^2_\Sigma(q))$.

The above definition is motivated by the observation that it is generally easy to translate proofs, but often very hard to obtain an isomorphic, or even reduction-preserving translation. However, one can expect to obtain a "lax retraction" of proofs, that is, when starting at the side of the categorical logic, the back-and-forth translation of a proof should be reducible to the original proof.

Theorem 32. *There is a semi-correspondence from $\mathcal{I}^{Bool}(Prop)$ to* $\mathbf{CPL}^{ND}$.

Proof. The signature translations Φ^1 and Φ^2 are the identity functors; the model translations β^i_Σ are the obvious bijections; and the sentence translation, i.e., the object part of α^i_Σ, is the obvious bijection, too. We omit the straightforward proof translations, i.e., the morphism parts of α^i_Σ. □

Similarly, there is a semi-correspondence between $\mathcal{I}_{thin}(IProp)$ and intuitionistic logic with Heyting algebra semantics. For modal and linear logic, the Curry-Howard-Tait correspondence is not so much a proved result, but rather a design paradigm that is satisfied a priori. In these cases, it makes more sense to see the institution constructed within our framework as the incarnation of the correspondence.

7 Conclusion and Future Work

We have presented a canonical way of obtaining proof-theoretic institutions for propositional categorical logics, following the spirit of the Curry-Howard-Tait isomorphism. We have proved generic deduction, soundness and completeness theorems, and given examples of categorical logics, for which categorical treatment had already been established in a non-institutional framework. Our definitions have the crucial advantage that they offer the possibility of parametric and modular specifications.

For classical logic, the institutional structure sheds light on the usual collapsing of proofs problem in classical logic (classical bicartesian closed categories are boolean algebras), which we avoided by using preorder-enriched categories, as in [13]. For Linear Logic, the specification of the modality ! is considerably involved; to simplify this specification, we might use a different categorical model (see [22]). But this would be too much of a departure from the point of view we have taken in this paper of using established notions of categorical models, not involving fibrations or indexed categories, for the time being. Another interesting direction left to future work is to consider linear logic with both a classical and a linear function space as in [21].

While trying to recover the Curry-Howard-Tait isomorphism as an explicit isomorphism between institutions, only a limited correspondence could be set up. One obstacle is the difference between, e.g., boolean algebra-valued and *Bool*-valued models. We have provided some general results about the relations between such models. Another obstacle is the different notions of proofs and proof reductions that make it hard to obtain isomorphic ore reduction-preserving translations. However, we have been able to obtain a "lax retraction" of proofs for classical logic where the back-and-forth translation of a proof can be reduced to the original proof.

The study of further properties, such as Craig interpolation and Beth definability, is the subject of future work, as is the extension of our framework to first-order logic. The latter will require a more powerful meta-language, namely one that permits to declare function symbols that take functions as arguments. Another interesting question is whether the institutions $\mathcal{I}(\mathcal{L})$ have elementary diagrams in the sense of [8]. Our current notion of model is obviously too weak to ensure this; for ensuring elementary diagrams one would need "intensional models" over signatures containing proof variables, to be valuated with proofs — such models would not only determine which propositions are true, but also why. With such models, it is also easy to show liberality of $\mathcal{I}(\mathcal{L})$.

References

1. J. Adámek, H. Herrlich, and G. E. Strecker. *Abstract and Concrete Categories.* Wiley Interscience, 1990.
2. A. Avron. The Semantics and Proof Theory of Linear Logic. Technical Report ECS–LFCS–87–27, University Edinburgh, 1987.
3. M. Bidoit and R. Hennicker. On the integration of observability and reachability concepts. In *Fossacs 2002*, volume 2303 of *LNCS*, pages 21–36. Springer, 2002.
4. G. Bierman and V. de Paiva. On an intuitionistic modal logic. *Studia Logica*, 65:383–416, 2000.
5. T. Borzyszkowski. Moving specification structures between logical systems. In *WADT 98*, volume 1589 of *LNCS*, pages 16–30. Springer, 1999.
6. R. Diaconescu. *Institution-independent model theory.* Manuscript, University of Bucharest.
7. R. Diaconescu. Institution-independent ultraproducts. *Fundamenta Informaticae*, 55:321–348, 2003.
8. R. Diaconescu. Elementary diagrams in institutions. *J. Logic Computation*, 14(5):651–674, 2004.
9. R. Diaconescu. Herbrand theorems in arbitrary institutions. *Information Processing Letters*, 90:29–37, 2004.
10. R. Diaconescu. An institution-independent proof of Craig Interpolation Theorem. *Studia Logica*, 77(1):59–79, 2004.
11. R. Diaconescu, J. Goguen, and P. Stefaneas. Logical support for modularisation. In *Logical Environments*, pages 83–130. Cambridge, 1993.
12. K. Došen. *Cut Elimination in Categories.* Kluwer Academic Publishers, Dordrecht, 1999.
13. C. Führmann and D. Pym. On the geometry of interaction for classical logic. In *LICS 2004*, pages 211–220. IEEE Computer Society Press, 2004. Extended version at `http://www.cs.bath.ac.uk/~cf/dj.pdf`.
14. J. L. Fiadeiro and J. F. Costa. Mirror, mirror in my hand: A duality between specifications and models of process behaviour. *Math. Struct. Comput. Sci.*, 6(4):353–373, 1996.

15. J. A. Goguen and R. M. Burstall. Institutions: Abstract model theory for specification and programming. *J. ACM*, 39:95–146, 1992.

16. J. A. Goguen and R. Diaconescu. Towards an algebraic semantics for the object paradigm. In *WADT 94*, volume 785 of *LNCS*. Springer, 1994.

17. J. A. Goguen and G. Rosu. Institution morphisms. *Formal Aspects of Computing*, 13:274–307, 2002.

18. R. Harper, F. Honsell, and G. Plotkin. A framework for defining logics. *Journal of the Association for Computing Machinery*, 40(1):143–184, 1993.

19. J. Lambek and P. J. Scott. *Introduction to Higher Order Categorical Logic*. Cambridge, 1986.

20. S. Mac Lane. *Categories for the Working Mathematician*. Springer, 1997.

21. M. Maietti, V. de Paiva, and E. Ritter. Categorical Models for Intuitionistic and Linear Type Theory. In J. Tiuryn, editor, *Foundations of Software Science and Computation Structures, Third International Conference*, pages 223–237. Springer, 2000.

22. M. E. Maietti, P. Maneggia, V. de Paiva, and E. Ritter. Relating categorical semantics for intuitionistic linear logic. *Appl. Categ. Structures*, 13(1):1–36, 2005.

23. T. Mossakowski, J. Goguen, R. Diaconescu, and A. Tarlecki. What is a logic? In J.-Y. Beziau, editor, *Logica Universalis*, pages 113–133. Birkhäuser, 2005.

24. V. d. P. M. H. N. Benton, G. Bierman. Term Assignment for Intuitionistic Linear Logic. Technical Report 262, University of Cambridge Computer Laboratory, 1992.

25. F. Pfenning and C. Schürmann. System description: Twelf - a meta-logical framework for deductive systems. *Lecture Notes in Computer Science*, 1632:202–206, 1999.

26. F. Rabe. First-Order Logic with Dependent Types. In N. Shankar and U. Furbach, editors, *Proceedings of the 3rd International Joint Conference on Automated Reasoning*, volume 4130 of *Lecture Notes in Computer Science*, pages 377–391. Springer, 2006.

27. F. Rabe, V. de Paiva, and T. Mossakowski. Categorical Logic Specifications, 2007. See `http://kwarc.eecs.iu-bremen.de/frabe/Research/catlog`.

28. H. Rasiowa and R. Sikorski. *The Mathematics of Metamathematics*. PWN, Warsaw, 1963.

29. D. Sannella and A. Tarlecki. Specifications in an arbitrary institution. *Inform. and Comput.*, 76:165–210, 1988.

30. L. Schröder, T. Mossakowski, and C. Lüth. Type class polymorphism in an institutional framework. In J. Fiadeiro, editor, *WADT 04*, volume 3423 of *LNCS*, pages 234–248. Springer, 2004.

31. R. A. G. Seely. Hyperdoctrines, natural deduction and the Beck condition. *Zeitschrift für mathematische Logik und Grundlagen der Mathematik*, 29(6):505–542, 1983.

32. R. A. G. Seely. Locally cartesian closed categories and type theory. *Mathematical Proceedings of the Cambridge Philosophical Society*, 95(1):33–48, 1984.

33. R. A. G. Seely. Linear logic, *-autonomous categories, and cofree coalgebras. In *Categories in Computer Science and Logic*, Contemp. Math., pages 371–382. AMS, 1989.

34. A. Tarlecki. On the existence of free models in abstract algebraic institutions. *Theoret. Comput. Sci.*, 37(3):269–304, 1985.

(Not for inclusion in the final version.)

A Specifications of the logics mentioned in the paper

The specifications are given in Twelf syntax. %% starts comments, declarations are of the form `symbol : type.` or `symbol : type = value.`. $\Pi x : A.B$ is denoted by `{x:A}B` and $A \to B$ by `A->B`. Axioms are given as `name : |- axiom`. %prefix and %infix define fixity and precedence of operators. The operators of $\mathcal{DFOL}$ are written as `forall[x:A]F`, `and`, and `impl`, $Mor(A, B)$ is written as `A=>B` and $\rightsquigarrow$ as `~>`.

Listing 1.1. *Cartesian*

```
%% Categorical Logic signature for cartesian structure, i.e.,
    conjuction and truth
%% requires catlog.elf
%% Florian Rabe, Till Mossakowski

%% truth
top : Ob.
bang : A => top.

%% conjunction
* : Ob -> Ob -> Ob.
                                                  %infix none 20 *.
*' : A => B  ->  A => C  ->  A => B * C.
                                                  %infix none 20 *'.
pi1  : A => B * C  ->  A => B.
                                                  %prefix 22 pi1.
pi2  : A => B * C  ->  A => C.
                                                  %prefix 22 pi2.
Pi1 = [A: Ob] [B: Ob] pi1 id (A * B).
Pi2 = [A: Ob] [B: Ob] pi2 id (A * B).
**' : {f : A => C} {g : B => D} A * B => C * D
    = [f] [g] (Pi1 A B ; f) *' (Pi2 A B ; g).
                                                        %infix left 20 **'.
%% Pi1 and Pi2 are the projections. pi1 and pi2 take their
    argument and post-compose the projection.
%% **' is the morphism component taking te product taking the
    product of the codomains.

%% rewrites for truth
rew_top : |- forall [f: A => top] f ~> bang.

%% rewrites for conjunction
rew_conj_1 : |- forall [f: A => B * C] pi1 f *' pi2 f ~> f.
rew_conj_2 : |- forall [f: A => B] forall [g: A => C] pi1 (f
    *' g) ~> f.
rew_conj_3 : |- forall [f: A => B] forall [g: A => C] pi2 (f
    *' g) ~> g.
rew_conj_4 : |- forall [f: A => B] forall [g: B => C] forall
    [h: B => D]
```

```
                  (f ; g) *' (f ; h) ~> f ; (g *' h).

%% derived morphism: associativity of product
assoc1 : (A * B) * C => A * (B * C) = (Pi1 (A * B) C ; Pi1 A
    B) *' ((Pi1 (A * B) C ; Pi2 A B) *' Pi2 (A * B) C).
assoc2 : A * (B * C) => (A * B) * C = (Pi1 A (B * C) *' (Pi2
    A (B * C) ; Pi1 B C)) *' (Pi2 A (B * C) ; Pi2 B C).
%% derived morphism: commutativity of product
comm : A * B => B * A = Pi2 A B *' Pi1 A B.
```

Listing 1.2. *Cocartesian*

```
%% Categorical Logic signature for disjunction and falsity
%% requires catlog.elf
%% Florian Rabe, Till Mossakowski

%% falsity
bot : Ob.
bangbang : bot => A.

%% disjunction
+ : Ob -> Ob -> Ob.
    %infix left 18 +.
inl : {A: Ob} {B: Ob} A => A + B.
inr : {A: Ob} {B: Ob} B => A + B.
+' : A => C  ->  B => C  ->  A + B => C.
    %infix left 18 +'.

%% rewrites for false
rew_bot : |- forall [f: bot => A] f ~> bangbang.

%% rewrites for disjunction
rew_disj_1 : |- forall [f: A + B => C] (inl A B ; f) +' (inr
    A B ; f) ~> f.
rew_disj_2 : |- forall [f: A => C] forall [g: B => C] inl A B
    ; f +' g ~> f.
rew_disj_3 : |- forall [f: A => C] forall [g: B => C] inr A B
    ; f +' g ~> g.

%% derived morphisms: distributive laws
%% dist2 cannot be derived since the category is not
    cartesian closed
dist1 : (A * C) + (B * C) => (A + B) * C
      = ((Pi1 A C ; inl A B) +' (Pi1 B C ; inr A B)) *' (Pi2
          A C +' Pi2 B C).
```

Listing 1.3. *CartClosed*

```
%% Categorical Logic signature for positive propositional
    logic, i.e., cartesian closed categories
%% requires cartesian.elf
%% Florian Rabe, Till Mossakowski
%% implication
--> : Ob -> Ob -> Ob.
                                        %infix right 17 -->.
 eval : {B: Ob} {C: Ob} (B --> C) * B => C.
 curry : A * B => C  ->  A => (B --> C).

%% rewrites for implication
 rew_imp_1 : |- forall [f: A * B => C] ( curry f **' id B ) ;
     eval B C ~> f.
 rew_imp_2 : |- forall [f: A => B --> C] curry ( ( f **' id B)
     ; eval B C ) ~> f.

%% the name of an arrow
name : A => B -> top => (A --> B)
      = [f] curry (Pi2 top A ; f).
unname : top => (A --> B) -> A => B
        = [g] bang *' id A ; g **' id A ; eval A B.

%% axiom for designated truth value
 law_des : |- Des == top.
```

Listing 1.4. Additional declarations in *CartClosed*

```
%% Categorical Logic signature for positive propositional
     logic, i.e., cartesian closed categories
%% requires cartesian.elf
%% Florian Rabe, Till Mossakowski

%% implication
--> : Ob -> Ob -> Ob.
                                        %infix right 17 -->.
 eval : {B: Ob} {C: Ob} (B --> C) * B => C.
 curry : A * B => C  ->  A => (B --> C).

%% rewrites for implication
 rew_imp_1 : |- forall [f: A * B => C] ( curry f **' id B ) ;
     eval B C ~> f.
 rew_imp_2 : |- forall [f: A => B --> C] curry ( ( f **' id B)
     ; eval B C ) ~> f.

%% the name of an arrow
name : A => B -> top => (A --> B)
      = [f] curry (Pi2 top A ; f).
unname : top => (A --> B) -> A => B
        = [g] bang *' id A ; g **' id A ; eval A B.

%% axiom for designated truth value
 law_des : |- Des == top.
```

Listing 1.5. Additional declarations to define intuitionistic negation

```
%% Categorical Logic signature for intuitionistic negation
     (only derived operations)
```

```
%% requires cartesian_closed.elf
%% Florian Rabe, Till Mossakowski

%% negation
- : Ob -> Ob
  = [A] A --> bot.
      %prefix 24 -.
neg : A * B => bot  ->  A => - B
    = curry.                    %prefix 22 neg.
contr : A * - A => bot
      = comm ; eval A bot.
negneg_i : A => - - A
         = neg contr.
```

Listing 1.6. Additional declarations for *Prop*

```
%% Categorical Logic signature to make an intuitionistic
    logic classical
%% requires int_negation.elf, cocartesian.elf
%% Florian Rabe, Till Mossakowski

%% morphism for tertium non datur
tnd : top => A + - A.

%% derived morphism for not-not elimination
negneg_e : - - A => A =
         (bang ; tnd) *' id (- - A) ; dist2 ; (Pi1 A (- - A)
             +' (contr ; bangbang)).

%% rewrites for negation
rew_negneg_1 : |- negneg_i ; negneg_e == id A.
rew_negneg_2 : |- negneg_e ; negneg_i == id - - A.
```

Listing 1.7. Additional declarations for *IS4*

```
%% Categorical Logic signature for modalities of
    intuitionistic S4
%% requires cartesian.elf
%% Florian Rabe, Till Mossakowski
%% following Sect. 7 of Biermann, de Paiva, "On an
    Intuitionistic Modal Logic" [1]

%% box is a functor
box : Ob -> Ob.
    %prefix 24 box.
box' : A => B -> box A => box B.
    %prefix 24 box'.

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% (box, delta, eps) is comonad

%% eps and delta are natural transformations
```

```
 delta : {A : Ob} box A => box (box A).
    %prefix 22 delta.
 eps : {A : Ob} box A => A.
    %prefix 22 eps.
 law_nat_eps : |- forall [f: A => B] eps A ; f == box' f ; eps
    B.
 law_nat_delta : |- forall [f: A => B] delta A ; box' (box' f)
    == box' f ; delta B.
%% comonad laws
law_comonad_1: |- delta A ; eps box A == id box A.
law_comonad_2: |- delta A ; box' eps A == id box A.
law_comonad_3: |- delta A ; delta box A == delta A ; box'
    delta A.

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% (box, m, mtop) is a monoidal functor

m :  {A: Ob} {B: Ob} box A * box B => box (A * B).
 mtop : top => box top.
 law_nat_m: |- forall [f: A => B] forall [g: C => D] box' f
     **' box' g ; (m B D) == (m A C) ; box' (f **' g).
 law_m_1 : |- id (box A) **' mtop ; m A top ; box' (Pi1 A top)
    == Pi1 (box A) top.
 law_m_2 : |- mtop **' id (box A) ; m top A ; box' (Pi2 top A)
    == Pi2 top (box A).
 law_m_3 : |- m A B **' id box C ; m (A * B) C ; box' assoc1 ==
               assoc1 ; id box A **' m B C ; m A (B * C).
 law_m_4 : |- comm ; m B A == m A B ; box' comm.

%% Note that assoc2 is used in [1]

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% (box, eps, delta, m, mtop) is a monoidal comonad

 law_eps_1: |- (m A B) ; eps (A * B) == eps A **' eps B.
 law_eps_2: |- mtop ; eps top == id top.
 law_delta_1: |- m A B ; delta (A * B) == delta A **' delta B
    ; m (box A) (box B) ; box' (m A B).
 law_delta_2: |- mtop ; box' mtop == mtop ; delta top.

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% (dia, eta, mu) is a monad

%% diamond is a functor
 dia : Ob -> Ob.
    %prefix 24 dia.
 dia' : A => B -> dia A => dia B.
    %prefix 24 dia'.
%% eta and mu are natural transformations
mu : {A : Ob} dia (dia A) => dia A.
    %prefix 22 mu.
```

```
eta : {A : Ob} A => dia A.
    %prefix 22 eta.
law_nat_eta : |- forall [f: A => B] eta A ; dia' f == f ; eta
    B.
law_nat_mu : |- forall [f: A => B] mu A ; dia' f == dia'
    (dia' f) ; mu B.
%% monad laws
law_monad_1 : |- eta dia A ; mu A == id dia A.
law_monad_2 : |- dia' eta A ; mu A == id dia A.
law_monad_3 : |- mu dia A ; mu A == dia' mu A ; mu A.

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% (dia, eta, mu) is a (box, eps, delta, m, mtop)-strong monad

st : {A : Ob} {B : Ob} box A * dia B => dia (box A * B).

law_st_1 : |- eps top **' id (dia A) ; Pi2 top (dia A)
           == st top A ; dia'(eps top **' id A) ; dia' (Pi2
              top A).
law_st_2 : |- assoc2 ; m A B **' id (dia C) ; st (A * B) C
           == id (box A) **' st B C ; st A (box B * C) ; dia'
              assoc2 ; dia' (m A B **' id C).
law_st_3 : |- id (box A) **' eta B ; st A B == eta (box A *
    B).
law_st_4 : |- st A (dia B) ; dia' (st A B) ; mu (box A * B)
           == id box A **' mu B ; st A B.
```

Listing 1.8. *Monoidal*

```
%% Categorical Logic signature for monoidal categories
%% requires catlog.elf
%% Florian Rabe, Valeria dePaiva, Till Mossakowski

%% multiplicative true
I : Ob.

%% multiplicative conjuction (tensor)
** : Ob -> Ob -> Ob.
    %infix none 20 **.
**' : A => C  ->  B => D  -> A ** B => C ** D.
    %infix none 20 **'.

%% morphisms
leftI : {A: Ob} I ** A => A.
rightI : {A: Ob} A ** I => A.
assoc : {A: Ob} {B: Ob} {C: Ob} (A ** B) ** C => A ** (B **
    C).
comm : {A: Ob} {B: Ob} A ** B => B ** A.

%% inverse morphisms
leftI-inv : {A: Ob} A => I ** A.
rightI-inv : {A: Ob} A => A ** I.
```

```
assoc-inv : {A: Ob} {B: Ob} {C: Ob} A ** (B ** C) => (A ** B)
    ** C.
comm-inv : {A: Ob} {B: Ob} B ** A => A ** B.

%% laws
law_leftI_iso_1 : |- leftI A ; leftI-inv A == id (I ** A).
law_leftI_iso_2 : |- leftI-inv A ; leftI A == id A.
law_rightI_iso_1 : |- rightI A ; rightI-inv A == id (A ** I).
law_rightI_iso_2 : |- rightI-inv A ; rightI A == id A.
law_assoc_iso_1 : |- (assoc A B C) ; (assoc-inv A B C) == id
    ((A ** B) ** C).
law_assoc_iso_2 : |- (assoc-inv A B C) ; (assoc A B C) == id
    (A ** (B ** C)).
law_comm_iso_1 : |- (comm A B) ; (comm-inv A B) == id (A **
    B).
law_comm_iso_2 : |- (comm-inv A B) ; (comm A B) == id (B **
    A).

law_unit_1 : |- leftI I == rightI I.
law_unit_2 : |- (assoc A I B) ; id A **' leftI B == (rightI A
    **' id B).
law_maclane : |- (assoc A B C) **' id D ; (assoc A (B ** C)
    D) ; id A **' (assoc B C D)
               == (assoc (A ** B) C D) ; (assoc A B (C ** D)).
law_comm_1 : |- (comm A B) ; (comm B A) == id A **' id B.
law_comm_2 : |- (comm A I) ; leftI A == rightI A.
law_comm_3 : |- (assoc A B C) ; (comm A (B ** C)) ; (assoc B
    C A)
             == (comm A B) **' id C ; (assoc B A C) ; id B
                **' (comm A C).
```

Listing 1.9. Additional declarations for *MonClosed*

```
%% Categorical Logic signature for monoidal closed categories
%% requires monoidal.elf
%% Florian Rabe, Valeria dePaiva, Till Mossakowski

%% linear implication
-o : Ob -> Ob -> Ob.                                  %infix left
    18 -o.

%% morphisms
lin_eval : {B: Ob} {C: Ob} (B -o C) ** B => C.
lin_curry : A ** B => C  ->  A => (B -o C).

%% rewrites
rew_adj_1 : |- forall [f: A ** B => C] (lin_curry f) **' id B
    ; (lin_eval B C) ~> f.
rew_adj_2 : |- forall [f: A => B -o C] lin_curry (f **' id B
    ; lin_eval B C) ~> f.

%% Law for designated truth values in linear logics
```

```
law_des_lin : |- Des == I.
```

Listing 1.10. Additional declarations for *MILL*

```
%% Categorical Logic signature for multiplicative exponential
    intuitionistic linear logic
%% requires monoidal_closed.elf
%% Florian Rabe, Valeria dePaiva, Till Mossakowski
%% Following Sect. 8 of Benton, Bierman, de Paiva, Hyland,
    "Term Assignment for Intuitionistic Linear Logic"

%% ! is a functor
! : Ob -> Ob.
    %prefix 24 !.
!' : A => B -> ! A => ! B.
    %prefix 24 !'.

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% (!, delta, eps) is comonad

%% eps and delta are natural transformations
delta : {A : Ob} ! A => ! (! A).
    %prefix 22 delta.
eps : {A : Ob} ! A => A.
    %prefix 22 eps.
law_nat_eps : |- forall [f: A => B] eps A ; f == !' f ; eps B.
law_nat_delta : |- forall [f: A => B] delta A ; !' (!' f) ==
    !' f ; delta B.
%% comonad laws
law_comonad_1: |- delta A ; eps ! A == id ! A.
law_comonad_2: |- delta A ; !' eps A == id ! A.
law_comonad_3: |- delta A ; delta ! A == delta A ; !' delta A.

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% (!, m, mI) is a monoidal functor

m :  {A: Ob} {B: Ob} ! A ** ! B => ! (A ** B).
mI : I => ! I.
law_nat_m: |- forall [f: A => B] forall [g: C => D] !' f **'
    !' g ; (m B D) == (m A C) ; !'(f **' g).
law_m_1 : |- id ! A **' mI ; (m A I) ; !' (rightI A) ==
    (rightI ! A).
law_m_2 : |- mI **' id ! A ; (m I A) ; !' (leftI A) == leftI
    (! A).
law_m_3 : |- m A B **' id ! C ; m (A ** B) C ; !' (assoc A B
    C) ==
             (assoc (! A) (! B) (! C)) ; id ! A **' m B C ; m
                 A (B ** C).
law_m_4 : |- m A B ; !' comm A B == comm (! A) (! B) ; m B A.

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% (!, eps, delta, m, mI) is a monoidal comonad
```

```
law_eps_1: |- (m A B) ; eps (A ** B) == eps A **' eps B.
law_eps_2: |- mI ; eps I == id I.
law_delta_1: |- m A B ; delta (A ** B) == delta A **' delta B
    ; m (! A) (! B) ; !' (m A B).
law_delta_2: |- mI ; !' mI == mI ; delta I.

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% each object ! A is endowed with a comonoid structure (! A,
    erase A, copy A)

%% erase and copy are natural transformations
erase : {A: Ob} ! A => I.
copy  : {A: Ob} ! A => ! A ** ! A.
law_nat_erase : |- forall [f: A => B] erase A  ==  !' f ;
    erase B.
law_nat_copy  : |- forall [f: A => B] copy A ; (!' f **' !'
    f) == !' f ; copy B.
%% comonoid laws
law_comonoid_1 : |- copy A ; id ! A **' erase A == rightI-inv
    ! A.
law_comonoid_2 : |- copy A ; erase A **' id ! A == leftI-inv
    ! A.
law_comonoid_3 : |- copy A ;  id ! A **' copy A ; assoc-inv
    (! A) (! A) (! A) == copy A ; copy A **' id ! A.

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% interaction between comonad and comonoid structure

%% (! A, delta) is a coalgebra
prom : ! A => B  ->  ! A => ! B
     = [f] delta A ; !' f.
law_prom_1 : |- forall [f: ! A => B] copy A ; prom f **' prom
    f == prom f ; copy B.
law_prom_2 : |- forall [f: ! A => B] erase A == prom f ;
    erase B.
%% erase, copy are also coalgebra maps
law_coalg_1 : |- erase A ; mI  == delta A ; !' erase A.
law_coalg_2 : |- delta A ; !' (copy A) == copy A ; delta A
    **' delta A ; m (! A) (! A).

%% every free coalgebra map f : (! A, delta) -> (! B, delta)
    is a map of comonoids f : (! A, erase A, copy A) -> (! B,
    erase B, copy B)
law_coalg_comonoid_1 : |- forall [f : ! A => ! B] (delta A ;
    !' f == f ; delta B  impl  erase A == f ; erase B).
law_coalg_comonoid_2 : |- forall [f : ! A => ! B] (delta A ;
    !' f == f ; delta B  impl  f ; copy B == copy A ; f **'
    f).
```