ENS Paris-Saclay
Master Parisien de Recherche en Informatique
Academic Year 2019-2020

Master 1 Internship Report

~

A CATEGORY OF IMPLICATIVE ALGEBRAS

Author :
Simon Mirwasser

Supervised by :
Valentin Blot
Alexandre Miquel
Étienne Miquey

April-July 2020

Contents

Introduction

We define and study various notions of morphisms of implicative algebras, in order to find a categorical structure for these algebras.

The notion of implicative algebra is an algebraic structure which underlies two different methods for finding new logical models, forcing and realizability, in both intuitionistic and classical logic. From this structure, it is possible to define a tripos which factors every triposes constructions that exist so far.

The first three sections give the definitions and explanations that we will need to define morphisms of implicative algebras, this is the state of the art of this report. In Section 1, we introduce the notions of forcing and realizability, with the algebraic structures related. Section 2 is about the notion of implicative algebra. We define it and give some important results about it. In Section 3, we give intuitions behind the notion of tripos and we make the construction of the implicative tripos. The last three sections are about morphisms in particular, this is where new results are presented. In Section 4 we define the most natural and naive kind of morphism and give the main results on this notion. Section 5 is about applicative morphisms which generalize naive morphisms. Finally, in Section 6, we give some ideas on other possible definitions, which are directions for future reseach on this topic.

1 Logic, Forcing and Realizability

The mathematics consist in finding results called theorems and proving them in a precise framework. An essential aspect of this study is that mathematic are formal : a theorem or a proof can not be approximate. This formalization goes through logic.

1.1. **Formalisation for Logic.** A logic is defined with a language corresponding to the formulas, the axioms which is a set of formulas and a deductive system which is a set of syntaxical rules allowing to obtain the valid theorems of a theory.

In logic, an important distinction is made between intuitionistic and classical logic. This distinction allows to know which proofs are constructive and which are not, because deduction rules in an intuitionistic system are made in order to have constructive proofs. For example, the excluded middle is not a principle allowed in intuitionistic logic because it is not constructive.

The point of view on logic presented so far is only syntactical. But an important point in logic is the notion of models, which is a semantic interpretation of a logic. A model can be seen as a world where the axioms and the rules are satisfied. Then, an important part of research in logic is to find new logical models.

1.2. **Forcing.** Historically, forcing was introduced by Cohen in 1963 [Coh63] to prove the independence of the continuum hypothesis with regard to the ZFC theory. Forcing can be seen as a technique to transform a model into an other. To formalize these models, two definitions are important.

Definition 1.1. A *lattice* $(\mathcal{L}, \leq)$ is a partially ordered set such that each pair $a, b \in \mathcal{L}$ has a meet $a \wedge b$ and a join $a \vee b$. A lattice is *bounded* when there is two elements $\top, \bot \in \mathcal{L}$ such that for all $a \in \mathcal{L}$, $\bot \leq a \leq \top$. When each subset $A \subseteq \mathcal{L}$ has a meet and a join, $\mathcal{L}$ is said *complete.*
An *Heyting algebra* $(\mathcal{H}, \leq)$ is a bounded lattice such that for all $a, b \in \mathcal{H}$ there is a greatest element $x \in \mathcal{H}$ such that $a \wedge x \leq b$. We denote $x = a \to b$. If $\mathcal{H}$ is a complete lattice, it is a *complete Heyting algebra.*

Definition 1.2. A *boolean algebra* is a quadruple $(\mathcal{B}, \leq, \top, \bot)$ such that

1. $(\mathcal{B}, \leq)$ is a bounded lattice where $\top$ is the upper bound and $\perp$ is the lower bound.
2. For all $a, b, c \in \mathcal{B}$,
$$a \vee (b \wedge c) = (a \vee b) \wedge (a \vee c) \qquad a \wedge (b \vee c) = (a \wedge b) \vee (a \wedge c).$$
3. For all $a \in \mathcal{B}$ there is $\neg a \in \mathcal{B}$ such that
$$a \vee \neg a = \top \qquad a \wedge \neg a = \perp$$
.

When $(\mathcal{B}, \leq)$ is a complete lattice, $\mathcal{B}$ is a *complete boolean algebra*.

When we use the forcing with an intuitionistic theory, this technique gives a complete Heyting algebra. Each formula is interpreted by an element of this algebra. For models of a classical theory, this is a complete boolean algebra instead of a complete Heyting algebra.

1.3. **Realizability.** Introducted by Kleene in 1945 [Kle45], realizability consists of an interpretation of each formula by the set of programs realizing this formula. This corresponds, using the Curry-Howard correspondence, to interpret a formula with the set of proofs of this formula.

In intuitionistic realizability, we get an algebra of programs which induces a model, for example a partial combinatory algebra (the definition is not given here).

The Curry-Howard correspondence establishes a link between logic and computation. An essential tool in the study of computation is the so-called λ-calculus. It is a minimalistic programming language, with a syntax given by
$$t, u := x \mid \lambda x.t \mid tu$$
where x belongs to a set of variables. A λ-term corresponds to a program and there is a syntactic rules that we can apply to a program, the β-reduction. The reduction of a term corresponds to the execution of the program and it is defined with the rewrinting rule $(\lambda x.u)v \rightarrow_\beta u[x := v]$, the usual substitution in λ-calculus. For example, $(\lambda x.x + 1)4 \rightarrow_\beta 4 + 1$. The strong fact about λ-calculus is that, even if this language is very simple, it is as expressive as any other programming language : it is Turing-complete.

Classical realizability is a great example of the link between logic and computing and the importance of λ-calculus. The beginning of this classical realizability comes from the mid-90's, fifty years after the intuitionistic one. This method created by Krivine [Kri09] is based on a particular λ-calculus, but the intuitions are the same in both intuitionistic and classical realizability. The structure of the induced model is however different, because it is based on this specific λ-calculus. This structure is called an abstract Krivine structure and has been defined by Streicher [Str13]. Thanks to this new method, Krivine has discovered new models of ZF with some other axioms like DC the axiom of dependent choice.

2 Implicative Algebras

Implicative algebras are the main object of this report. Defined by Miquel [Miq20], it is an algebraic structure underlying forcing and realizability. As explained by Van Oosten [VO02], Scott was the first to make this link, but the central point here is that we get a unique and simple structure where we used to have four different structures (complete Heying algebra, complete boolean algebra, partial combinatory algebra and abstract Krivine structure), so one unique way to represent truth values.

2.1. **Intuitions.** An implicative structure defined in the next subsection is an algebraic structure equipped with a partial order. The main idea of this definition is that both formulas and realizers are elements of the same set. Moreover, the realizers can also be seen as truth values, which is exactly what gives to implicative algebras the power to underlie both forcing and realizability.

Hence, the partial order of the structure can be understood in different ways depending on how the elements are considered. For two elements a and b, if they are both considered as truth values, if $a \preccurlyeq b$ then the order corresponds to the usual order of subtyping. But if they both are seen as realizers, it is a relation on the power of the realizer. This point of view still has a meaning in terms of forcing, and can be understood as the strength of a condition. Finally, if a is seen as a program and b as a type, this order means that a has type b.

This structure is therefore able to gather the intuitions from both forcing and realizability by using the same set for each kind of object studied, thanks to the Curry-Howard correspondance. The notion of truth will then be given by a subset of the structure containing the true formulas.

2.2. **Implicative Structures.**

Definition 2.1. An *implicative structure* $\mathscr{A}$ is a complete meet-semilattice $(\mathscr{A}, \preccurlyeq)$[1] equipped with a binary operation $(a, b) \mapsto a \to b$ such that

1. For $a, a', b, b' \in \mathscr{A}$,
$$\text{if } a' \preccurlyeq a \text{ and } b \preccurlyeq b' \text{ then } (a \to b) \preccurlyeq (a' \to b')$$
2. For $a \in \mathscr{A}$ and $B \subseteq \mathscr{A}$,
$$a \to \bigcurlywedge_{b \in B} b = \bigcurlywedge_{b \in B} (a \to b)$$

Complete Boolean algebras, complete Heyting algebras, total combinatory algebras and abstract Krivine structures can be translated as implicative structures [Miq20]. This result illustrate how implicative structures underlie forcing and realizability and their algebraization.

2.3. **Interpreting λ-terms.** In applicative structures, we want to be able to interpret both proofs and programs. Then, we shall now define λ-terms and their interpretation.

Firstly, we need to intrepret application and abstraction in an implicative structure.

Definition 2.2. The *application* and the *abstraction* are defined such that for $a, b \in \mathscr{A}$ and $f : \mathscr{A} \to \mathscr{A}$,
$$ab = \bigcurlywedge \{c \in \mathscr{A} : a \preccurlyeq (b \to c)\} \qquad\qquad \boldsymbol{\lambda} f = \bigcurlywedge_{a \in \mathscr{A}} (a \to f(a))$$

It is now possible to define λ-terms and how to interpret them.

Definition 2.3. In an applicative structure, λ-terms are defined by :
$$t, u := x \mid a \mid \lambda x.t \mid tu$$
with x a variable and $a \in \mathscr{A}$. Then, each closed λ-term t is interpreted by an element $t^{\mathscr{A}} \in \mathscr{A}$ defined by induction with
$$\begin{aligned} a^{\mathscr{A}} &= a && \text{(for } a \in \mathscr{A}) \\ (tu)^{\mathscr{A}} &= t^{\mathscr{A}} u^{\mathscr{A}} \\ (\lambda x.t)^{\mathscr{A}} &= \boldsymbol{\lambda}(a \mapsto (t[a/x]^{\mathscr{A}}) \end{aligned}$$
where $t[a/x]$ is the usual substitution in λ-calculus.

[1]This means that each subset $B \subseteq \mathscr{A}$ has a greatest lower bound noted $\bigcurlywedge B$

2.3.1. *Adequacy.* In an implicative algebra, it is possible to define conjuction, disjunction and usual quantifiers following Girard's work [Gir72] [GTL89] :

$$a \times b = \bigcurlywedge_{c \in \mathscr{A}} ((a \to b \to c) \to c) \qquad \forall_{i \in I} a_i = \bigcurlywedge_{i \in I} a_i$$

$$a + b = \bigcurlywedge_{c \in \mathscr{A}} ((a \to c) \to (b \to c) \to c) \qquad \exists_{i \in I} a_i = \bigcurlywedge_{c \in \mathscr{A}} \left(\bigcurlywedge_{i \in I} (a_i \to c) \to c \right).$$

We have an adequacy result in [Miq20] for the typing rules coming from these operations : they are valid in any implicative structure.

Another important result on typing is about the following combinators

$$\mathbf{I} = \lambda x.x \qquad \mathbf{K} = \lambda xy.x$$
$$\mathbf{B} = \lambda xyz.x(yz) \qquad \mathbf{W} = \lambda xy.xyy$$
$$\mathbf{C} = \lambda xyz.xzy \qquad \mathbf{S} = \lambda xyz.xz(yz)$$

They have well-known types and these types corresponds to the value of there interpretation in any implicative structure [Miq20] :

$$\mathbf{I}^{\mathscr{A}} = \bigcurlywedge_{a \in \mathscr{A}} (a \to a) \qquad \mathbf{K}^{\mathscr{A}} = \bigcurlywedge_{a,b \in \mathscr{A}} (a \to b \to a)$$

$$\mathbf{B}^{\mathscr{A}} = \bigcurlywedge_{a,b,c \in \mathscr{A}} ((a \to b) \to (c \to a) \to c \to b) \qquad \mathbf{W}^{\mathscr{A}} = \bigcurlywedge_{a,b \in \mathscr{A}} ((a \to a \to b) \to a \to b)$$

$$\mathbf{C}^{\mathscr{A}} = \bigcurlywedge_{a,b,c \in \mathscr{A}} ((a \to b \to c) \to b \to a \to c) \qquad \mathbf{S}^{\mathscr{A}} = \bigcurlywedge_{a,b,c \in \mathscr{A}} ((a \to b \to c) \to (a \to b) \to a \to c)$$

In order to compare intuitonistic and classical logic, we define

$$\text{œ}^{\mathscr{A}} = \bigcurlywedge_{a,b \in \mathscr{A}} (((a \to b) \to a) \to a)$$

following Pierce's law.

2.4. **Implicative Algebras.** Implicative structures are now equipped with λ-calculus but an essential element is still missing : the translation of the notion on truth.

Definition 2.4. A *separator* of an implicative structure $\mathscr{A}$ is a subset $S \subseteq \mathscr{A}$ such that for all $a, b \in \mathscr{A}$

1. If $a \in S$ and $a \preccurlyeq b$ then $b \in S$
2. $\mathbf{K}^{\mathscr{A}} = (\lambda xy.x)^{\mathscr{A}} \in S$ and $\mathbf{S}^{\mathscr{A}} = (\lambda xyz.xz(yz))^{\mathscr{A}} \in S$
3. If $(a \to b) \in S$ and $a \in S$ then $b \in S$

An implicative structure equipped with a separator is an *implicative algebra* and if $\perp \notin S$ the implicative algebra is called *consistent*. When $\text{œ}^{\mathscr{A}} \in S$, the algebra is said *classical*.

This definition matches with the λ-calculus defined previously :

Proposition 2.5. *For each closed term t, $t^{\mathscr{A}} \in S$.*

Some particular separators will be useful in what follows.

Definition 2.6. Let $(\mathscr{A}, \preccurlyeq, \to)$ be an implicative structure. We define $S^0_J(\mathscr{A})$, the *intuitionistic core* of $\mathscr{A}$ as the smallest separator of $\mathscr{A}$, and $S^0_K(\mathscr{A})$ the *classical core* of $\mathscr{A}$ as the smallest separator containing $\text{æ}^{\mathscr{A}}$.

As for implicative strutures, complete Boolean algebras, complete Heyting algebras, total combinatory algebras and abstract Krivine structures can be traduced as implicative algebras. In addition, these algebras are consistent (in the sense of Definition 2.4).

3 The Implicative Tripos

3.1. **Triposes.** Triposes were introduced by Hyland, Johnstone and Pitts[2] [HJP80] in 1980 in order to create categorical models for higher-order logic. Here we will see that it is possible to create a tripos from an implicative algebra but before that we have to introduce the notion of category.

3.1.1. *Category Theory.* We give here the definitions that will be used later.

Definition 3.1. A *category* $\mathbf{C}$ is given by a class of objects and a class of morphisms $\mathbf{C}(a,b)$ for all $a, b \in \mathbf{C}$ such that :

- Morphisms can be composed associatively with an operator noted $\circ$: for each $f \in \mathbf{C}(a,b)$, $g \in \mathbf{C}(b,c)$, we have $g \circ f \in \mathbf{C}(a,c)$.
- For each $a \in \mathbf{C}$ there is a morphism $id_a \in \mathbf{C}(a,a)$ such that :

$$\forall f \in \mathbf{C}(a,b),\ f \circ id_a = id_b \circ f = f.$$

Some categories will be useful for what follows :

Set The category where the objects are the sets and the morphisms are the functions between sets

Pos The category of partially ordered sets where morphisms are monotonic functions

HA The category of Heyting algebras with Heyting algebras morphisms

Definition 3.2. An *Heying algebras morphism* is a map $f : \mathcal{H} \to \mathcal{H}'$ such that $\mathcal{H}$ and $\mathcal{H}'$ are Heyting algebras and

1. for all $a, b \in \mathcal{H}$,

$$f(a \wedge b) = f(a) \wedge f(b) \qquad f(a \vee b) = f(a) \vee f(b) \qquad f(a \to b) = f(a) \to f(b)$$

2. $f(\top) = \top$ and $f(\bot) = \bot$.

Definition 3.3. Let $\mathbf{C}$ be a category. The *dual category* $\mathbf{C}^{op}$ is the category with the same objects where morphisms are reversed : $\mathbf{C}^{op}(a,b) = \mathbf{C}(b,a)$ for each $a, b \in \mathbf{C}$. The composition is also reversed : $f \circ_{\mathbf{C}^{op}} g = g \circ_{\mathbf{C}} f$.

It is possible to define maps between categories.

[2] To be more precise, they introduced **Set**-based triposes, which are one kind of triposes, but they are the only one that we will study here.

Definition 3.4. A *functor* F between two categories $\mathbf{C}$ and $\mathbf{D}$ is a correspondence mapping each object a of $\mathbf{C}$ to an object $F(a)$ of $\mathbf{D}$ and each morphism $f \in \mathbf{C}(a,b)$ to a morphism $F(f) \in \mathbf{D}(F(a), F(b))$ such that :

- for each object a of $\mathbf{C}$, $F(id_a) = id_{F(a)}$
- for each morphisms $f \in \mathbf{C}(a,b)$ and $g \in \mathbf{C}(b,c)$, $F(g \circ f) = F(g) \circ F(f)$.

The Cartesian product between sets can be described and generalized in category theory :

Definition 3.5. For a category $\mathbf{C}$ and two objects $a, b \in \mathbf{C}$, a *product* of a and b is a triple $(a \times b, \pi_1, \pi_2)$ where $a \times b \in \mathbf{C}, \pi_1 \in \mathbf{C}(a \times b, a)$ and $\pi_2 \in \mathbf{C}(a \times b, b)$ such that for each $c \in \mathbf{C}, f \in \mathbf{C}(c,a), g \in \mathbf{C}(c,b)$ there is a unique morphism $h \in \mathbf{C}(c, a \times b)$ such that

$$\begin{tikzcd} & c \arrow[dl, "f"'] \arrow[d, "h"] \arrow[dr, "g"] & \\ a & a \times b \arrow[l, "\pi_1"] \arrow[r, "\pi_2"'] & b \end{tikzcd}$$

commutes (i.e each pair of path in this diagram with the same points at the beginning and at the end is equal, using the composition of morphisms). A category $\mathbf{C}$ is *Cartesian* when each pair of object has a product and $\mathbf{C}$ contains a terminal object (an object t is said *terminal* if for each object a of $\mathbf{C}$ there is a unique morphism in $\mathbf{C}(a,t)$).
If $f, g \in \mathbf{C}(a,b)$, $eq \in \mathbf{C}(e,a)$ is an *equalizer* if for each $m \in \mathbf{C}(o,e)$ there is a unique $u \in \mathbf{o}, \mathbf{e}$ such that

$$\begin{tikzcd} e \arrow[r, "eq"] & a \arrow[r, shift left, "f"] \arrow[r, shift right, "g"'] & b \\ o \arrow[u, "u"] \arrow[ur, "m"'] & & \end{tikzcd}$$

commutes. When a Cartesian category has all the equalizers, it is said to be *finitely complete*[3].
If for each index set I and each family $(x_i)_{i \in I}$ of objects of $\mathbf{C}$ there is x an object of $\mathbf{C}$ and family of morphisms $(\pi_i)_{i \in I}$ such that for each $i \in I$, $\pi_i \in \mathbf{C}(x, x_i)$ such that for each y object and $(f_i)_{i \in I}$ family of morphisms from y to x_i there is a unique $f \in \mathbf{C}(y,x)$ such that for each $i \in I$

$$\begin{tikzcd} & x \arrow[d, "\pi_i"] \\ y \arrow[ur, "f"] \arrow[r, "f_i"'] & x_i \end{tikzcd}$$

commutes, the category $\mathbf{C}$ has all small products.

3.1.2. *Definition.* The formal definition of triposes will not be presented here, because it is a very technical definition with a lot of categorical notions and it is not useful for our work. However, we shall give some intuitive ideas about triposes.

The triposes considered here are functors from the category $\mathbf{Set}^{op}$ to the category $\mathbf{HA}$ which fullfil three conditions. The objects of $\mathbf{Set}^{op}$ represent the types, and the functor associates each type I to the set of predicates over I (which is represented by an element of $\mathbf{HA}$). The order between two elements of an Heyting algebra represents the logical implication and the equality represents the equivalence. The tripos has to be a functor because with the commutativity

[3] This is not the usual definition, but the usual one needs a lot more categorical definitions that we would only need to define a complete category.

properties of functors it is possible to make substitutions in the formulas. The first two conditions in the definition of a tripos allow us to use existential and universal quantifications and ensure that substitutions commute with these quantifiers. The last condition expresses that we have a particular type (which is here an element of $\mathbf{Set}^{op}$) representing the type of propositions and a particular predicate over this type asserting the truth of a proposition.

3.2. **The Implicative Tripos.** Triposes can be constructed from complete Heyting or Boolean algebras, from partial combinatory algebras and from abstract Krivine structures. It means that from both (intuitionistic or classical) forcing and realizability there is a construction of a tripos. But it is possible to construct a tripos from implicative algebras, which factors each construction from forcing and realizability.

3.2.1. *Construction.* The first step of this construction is to define the product between implicative structures. For an index set I, let $(\mathscr{A}_i)_{i\in I} = (\mathscr{A}_i, \preccurlyeq_i, \to_i)_{i\in I}$ be a familly of implicative structure indexed by I. With $\mathscr{A} = \prod_{i\in I} \mathscr{A}_i$, we equip $\mathscr{A}$ with an order and an implication defined by

$$(a_i)_i \preccurlyeq (b_i)_i \Leftrightarrow \forall i \in I,\ a_i \preccurlyeq_i b_i$$
$$(a_i)_i \to (b_i)_i = (a_i \to_i b_i)_i.$$

The triple $(\mathscr{A}, \preccurlyeq, \to)$ is an applicative structure.

For each implicative structure $(\mathscr{A}, \preccurlyeq, \to)$ any separator induces a preorder relation called relation of entailment defined by

$$a \vdash_S b \Leftrightarrow (a \to b) \in S$$

Now, let $(\mathscr{A}, \preccurlyeq, \to, S)$ be an implicative algebra. For each index set I, the implicative structure $\mathscr{A}^I = \prod_{i\in I} \mathscr{A}$ is defined following the product construction. It is possible to define the *uniform power separator* for each index set I by

$$S[I] = \{(a_i) \in \mathscr{A}^I : \exists s \in S,\ \forall i \in I,\ s \preccurlyeq a_i\}.$$

For each I, this set is a separator of $\mathscr{A}^I$ (so $(\mathscr{A}^I, S[I])$ is an implicative algebra).

The last definition that we need for the construction of the tripos is the Heyting algebra induced by the relation of entailement. Let $(\mathscr{A}, \preccurlyeq, \to, S)$ be an implicative algebra. This relation induces an equivalence relation :

$$a \dashv\vdash_S b \Leftrightarrow (a \to b), (b \to a) \in S$$

Hence, it is possible to quotient $\mathscr{A}$ by this relation $\dashv\vdash_S$, we obtain a set $\mathscr{A}/S$. Writing $[a] \in \mathscr{A}/S$ the equivalence class of $a \in \mathscr{A}$, we define the order

$$[a] \leq_S [b] \Leftrightarrow a \vdash_S b.$$

Let $\mathcal{H} = (\mathscr{A}/S, \leq_S)$, defining for all $a, b \in \mathscr{A}$ the operations

$$[a] \to_{\mathcal{H}} [b] = [a \to b] \qquad [a] \wedge_{\mathcal{H}} [b] = [a \times b] \qquad [a] \vee_{\mathcal{H}} [b] = [a + b]$$

and the elements

$$\top_{\mathcal{H}} = [\top] = S \qquad \bot_{\mathcal{H}} = [\bot],$$

$\mathcal{H}$ is a Heyting algebra.

Finally, let $\mathscr{A}$ be an implicative algebra. For each set I we write $\mathsf{P}I = \mathscr{A}^I/S[I]$. For $f \in \mathbf{Set}^{op}(I,J)$ ($f : J \to I$ is a function), $\mathsf{P}f$ is defined such that for each $(a_i)_{i\in I}$, $\mathsf{P}f([(a_i)_{i\in I}]) = [(a_{f(j)})_{j\in J}]$ and $\mathsf{P}f \in \mathbf{HA}(\mathsf{P}I, \mathsf{P}J)$. The map $\mathsf{P} : \mathbf{Set}^{op} \to \mathbf{HA}$ is a functor and this functor is a tripos [Miq20].

A strong result proved by Miquel is that each tripos is isomorphic to an implicative tripos. This result illustrates that this construction factors every existing tripos constructions.

4 Naive Morphisms

After having recalled the basics of the theory of implicative algebras, we can now address the central topic of this internship, namely: the definition of a category of implicative algebras. In order to define such a category where objects are implicative algebras, the main point is to find a good definition for morphisms between implicative algebras.

By good definition we mean two things. First, the most important, the definition needs to make sense from a logical point of view. That is : a morphism between two implicative algebras $\mathscr{A}$ and $\mathscr{B}$ should preserve some logical invariants, typically by commuting with conjunction, and possibly with other connectives. The second feature that we want for a good definition is that the corresponding category has properties which seem important. For exemple, the product of implicative algebras is an implicative algebra, hence the category should be Cartesian.

For these reasons, the first definition to consider is the most naive one, because it is the most natural : the definition where the considered map commutes with the two fundamental operations of implicative algebras: the implication, and arbitrary meets.

4.1. Definition.

Definition 4.1. For $(\mathscr{A}, \to_{\mathscr{A}}, S_{\mathscr{A}})$ and $(\mathscr{B}, \to_{\mathscr{B}}, S_{\mathscr{B}})$ two implicative algebras, a map $\varphi : \mathscr{A} \to \mathscr{B}$ is called a *naive morphism* if it fulfills

1. For each $P \subseteq \mathscr{A}$, $\varphi(\bigwedge P) = \bigwedge \varphi(P)$.
2. For each $a, b \in \mathscr{A}$, $\varphi(a \to_{\mathscr{A}} b) = \varphi(a) \to_{\mathscr{B}} \varphi(b)$.
3. $\varphi(S_{\mathscr{A}}) \subseteq S_{\mathscr{B}}$.

Note that the first condition 1. when $P = \emptyset$ corresponds to $\varphi(\top) = \top$.

The category where objects are implicative algebras and the morphisms are naive morphisms is denoted by **IAn**

Remark 4.2. From this definition it is possible to define naive morphisms for implicative structures, saying that a map is such a morphism if and only if it fullfils axioms 1. and 2. of Definition 4.1. Hence, it would be possible to study the category **ISn** of implicative structures with naive morphisms.

4.2. Logical and Categorical Results.

Here, we only give intuitions and some results without proofs on naive morphisms, because this notion is not the center of this work. It is the most natural idea but it seems that the notion of applicative morphisms given in Section 5 is better.

First, the following proposition gives which logical results can be given from naive morphisms.

Proposition 4.3. *Let $\mathscr{A}$ and $\mathscr{B}$ be two implicative structures and $\varphi : \mathscr{A} \to \mathscr{B}$ be a naive morphism* [4]*. We get the following results :*

[4] Here we use Remark 4.2 : φ only fullfils axioms 1. and 2. from Definition 4.1.

1. *for each* $a, b \in \mathscr{A}$, $\varphi(ab) \succcurlyeq \varphi(a)\varphi(b)$
2. *for each closed term* t, $\varphi(t^{\mathscr{A}}) \succcurlyeq t^{\mathscr{B}}$
3. *if* $S_{\mathscr{A}}$ *and* $S_{\mathscr{B}}$ *are the intuitionistic (resp. classical) cores of* $\mathscr{A}$ *and* $\mathscr{B}$, $\varphi(S_{\mathscr{A}}) \subseteq S_{\mathscr{B}}$.

In terms of realizability, naive morphisms are convincing but in order to consider this definition as a good one, we would like to have some categorical results about it.

Proposition 4.4. *The category* **IAn** *is Cartesian and the category* **ISn** *is complete.*

This proposition is, as explained before, essential on an intuitive point of view. To prove it[5], we need the following lemma.

Lemma 4.5. *For an index set* I *and a family of implicative algebras* $(\mathscr{A}_i)_{i\in I}$ *where for each* $i \in I$ *the separator of* $\mathscr{A}_i$ *is* S_i, $S = \prod_{i\in I} S_i$ *is a separator for the implicative structure* $\mathscr{A} = \prod_{i\in I} \mathscr{A}_i$

Proof.

- If $(a_i) \in S$ and $(b_i) \in \mathscr{A}$ such that for each $i \in I$, $a_i \preccurlyeq b_i$, using the axiom 1. of the separator, each $b_i \in S_i$: $(b_i) \in S$.
- Using the Proposition 4.3 from [Miq20], $\mathbf{K}^{\mathscr{A}} = (\mathbf{K}^{\mathscr{A}_i})_i$ and $\mathbf{S}^{\mathscr{A}} = (\mathbf{S}^{\mathscr{A}_i})_i$. Hence we have (with axiom 2. of the separator) $\mathbf{K}^{\mathscr{A}}$, $\mathbf{S}^{\mathscr{A}} \in S$.
- If $(a_i) \in S$ and $(b_i) \in \mathscr{A}$ such that for each $i \in I$, $a_i \to b_i \in S_i$, for each $i \in I$, $b_i \in S_i$ so $(b_i) \in S$.

∎

However, the definition of the following section seems better than this one : we give a characterization in terms of tripos that may not be possible with naive morphisms. Moreover, in a more intuitive point of view, this definition may be too restrictive and some important maps, like transformations between logic or lambda calculus, could not be naive morphisms.

5 Applicative Morphisms

Naive morphisms defined in Section 4 fullfil important logical properties. The results proved in this section will give a justification to this point : here, we will define the notion of applicative morphism which is a generalization of the notion of naive morphisms.

The Definition 5.1 of applicative morphisms seems, at least at first sight, very technical and it is hard to understand the ideas behind. That is why it is important to explain where this definition comes from and what is hidden behind.

5.1. **Origins.** The notion of applicative morphism comes from [vOZ16]. In this paper, van Oosten and Zou link different kind of realizability using some strutures like ordered partial combinatory algebras (OPCA) introduced by Hofstra [Hof06]. The Definition 1.12 in [vOZ16] is a definition of an applicative morphism between two filtered opcas. The important point here is that filtered opcas and implicative algebras are very similar. Hence, it is easy to transpose applicative morphisms to implicative algebras. This work has been made by Ferrer and Malherbe [FM17] in order to find an adjunction between AKS and implicative algebras.

[5] Here we prove this lemma because it will be useful later, but as explained before the proof of the proposition will not be given.

However, Miquel found an issue in the Proposition 1.13 of [vOZ16]. Then, the definition of applicative morphisms has to be slightly modified in order to get the property of commutation with finite meets in the induced Heyting algebra (here we only consider the definition for implicative algebras, but it is easy to give one for filtered opcas).

5.2. **Définition and Fundamental Results.**

Definition 5.1. A map $\varphi : \mathscr{A} \to \mathscr{B}$ is an *applicative morphism* when it fulfills

1. $\varphi(S_{\mathscr{A}}) \subseteq S_{\mathscr{B}}$
2. There is $r \in S_{\mathscr{B}}$ such that for all $a, a' \in \mathscr{A}$, $r\varphi(a')\varphi(a) \preccurlyeq \varphi(a'a)$.
3. There is $u \in S_{\mathscr{B}}$ such that for all $a, a' \in \mathscr{A}$, if $a \preccurlyeq a'$ then $u\varphi(a) \preccurlyeq \varphi(a')$.

Remark 5.2. This definition is a correction of Definition 3.1 of [FM17] using the correction of Proposition 1.13 of [vOZ16].

These conditions 1. and 2. are equivalent to

1. There is $r \in S_{\mathscr{B}}$ such that for all $a, a' \in \mathscr{A}$, $r \preccurlyeq \varphi(a') \to \varphi(a) \to \varphi(a'a)$.
2. There is $u \in S_{\mathscr{B}}$ such that for all $a, a' \in \mathscr{A}$, if $a \preccurlyeq a'$ then $u \preccurlyeq \varphi(a) \to \varphi(a')$.

In order to use this definition, there is a useful analogy with modal logic : the map φ can be seen as the $\Box$ ("necessity") of modal logic, and the two rules (N) the necessitation rule and (K) the distribution axiom

$$\frac{\vdash A}{\vdash \Box A}\ (N) \qquad\qquad \vdash \Box(A \Rightarrow B) \Rightarrow (\Box A \Rightarrow \Box B)\ (K)$$

correspond respectively to the axiom 1. and the axiom 2. of the Definition 5.1 because, using the entailement preorder, conditions 1. and 2. can be understood by

1. for all $a \in \mathscr{A}$, $\vdash_{S_{\mathscr{A}}} a$ implies $\vdash_{S_{\mathscr{B}}} \varphi(a)$
2. for all $a, a' \in \mathscr{A}$, $\varphi(a \to a') \vdash_{S_{\mathscr{B}}} \varphi(a) \to \varphi(a')$.

The proof of the next proposition uses this analogy.

An important result in modal logic is that rules (N) and (K) imply some commutation properties with usual operators. Hence, following the previous analogy, we get a similar result on applicative morphisms.

Proposition 5.3. *For each applicative morphism $\varphi : \mathscr{A} \to \mathscr{B}$, there are :*

$$\mathtt{fold}_{\times}, \mathtt{unfold}_{\times}, \mathtt{fold}_{+}, \mathtt{unfold}_{\forall}, \mathtt{fold}_{\top}, \mathtt{unfold}_{\top}, \mathtt{fold}_{\bot} \in S_{\mathscr{B}}$$

such that for all $a_1, a_2 \in \mathscr{A}$, for all index set I and for all $(a_i)_{i\in I} \in \mathscr{A}^I$

$$\begin{aligned} \mathtt{fold}_{\times} &\preccurlyeq \varphi(a_1) \times \varphi(a_2) \to \varphi(a_1 \times a_2) \\ \mathtt{unfold}_{\times} &\preccurlyeq \varphi(a_1 \times a_2) \to \varphi(a_1) \times \varphi(a_2) \\ \mathtt{fold}_{+} &\preccurlyeq \varphi(a_1) + \varphi(a_2) \to \varphi(a_1 + a_2) \\ \mathtt{unfold}_{\forall} &\preccurlyeq \varphi(\forall_{i\in I} a_i) \to \forall_{i\in I}\varphi(a_i) \end{aligned} \qquad \begin{aligned} \mathtt{fold}_{\top} &\preccurlyeq \top \to \varphi(\top) \\ \mathtt{unfold}_{\top} &\preccurlyeq \varphi(\top) \to \top \\ \mathtt{fold}_{\bot} &\preccurlyeq \bot \to \varphi(\bot) \end{aligned}$$

Before giving the proof of this result, we introduce a lemma where the terms used in the proof are characterised.

Lemma 5.4. *There are terms* $\mathtt{inl}, \mathtt{inr}, \mathtt{split}, \mathtt{projl}, \mathtt{projr}, \mathtt{pair}$ *such that for each implicative algebra* $\mathscr{A}$

$$\mathtt{inl}^{\mathscr{A}}, \mathtt{inr}^{\mathscr{A}}, \mathtt{split}^{\mathscr{A}}, \mathtt{projl}^{\mathscr{A}}, \mathtt{projr}^{\mathscr{A}}, \mathtt{pair}^{\mathscr{A}} \in S$$

and for each $a_1, a_2, a \in \mathscr{A}$ *:*

$$\begin{aligned} \mathtt{inl}^{\mathscr{A}} &\preccurlyeq a_1 \to a_1 + a_2 & \mathtt{projl}^{\mathscr{A}} &\preccurlyeq a_1 \times a_2 \to a_1 \\ \mathtt{inr}^{\mathscr{A}} &\preccurlyeq a_2 \to a_1 + a_2 & \mathtt{projr}^{\mathscr{A}} &\preccurlyeq a_1 \times a_2 \to a_2 \\ \mathtt{case}^{\mathscr{A}} &\preccurlyeq a_1 + a_2 \to (a_1 \to a) \to (a_2 \to a) \to a & \mathtt{pair}^{\mathscr{A}} &\preccurlyeq a_1 \to a_2 \to a_1 \times a_2 \end{aligned}$$

Proof. Let

$$\begin{aligned} \mathtt{inl}^{\mathscr{A}} &= \lambda a \lambda x \lambda y.xa & \mathtt{projl}^{\mathscr{A}} &= \lambda x.x(\lambda yz.y) \\ \mathtt{inr}^{\mathscr{A}} &= \lambda a \lambda x \lambda y.ya & \mathtt{projr}^{\mathscr{A}} &= \lambda x.x(\lambda yz.z) \\ \mathtt{case}^{\mathscr{A}} &= \lambda xab.xab & \mathtt{pair}^{\mathscr{A}} &= \lambda abx.xab \end{aligned}$$

Then, all these elements are in S and by adequacy of typing rules, we get the wanted inequalities. ∎

Proof of Proposition 5.3. Let $\varphi : \mathscr{A} \to \mathscr{B}$ be an applicative morphism.

- $\mathtt{fold}_\times$: First, for a_1 and a_2 in $\mathscr{A}$ we have

$$\begin{aligned} \mathtt{pair}^{\mathscr{A}} &\preccurlyeq a_1 \to a_2 \to a_1 \times a_2 & &\text{(by adequacy of typing)} \\ u\varphi(\mathtt{pair}^{\mathscr{A}}) &\preccurlyeq \varphi(a_1 \to a_2 \to a_1 \times a_2) \\ r(u\varphi(\mathtt{pair}^{\mathscr{A}}))\varphi(a_1) &\preccurlyeq \varphi(a_2 \to a_1 \times a_2) \\ r(r(u\varphi(\mathtt{pair}^{\mathscr{A}}))\varphi(a_1)) &\preccurlyeq \varphi(a_2) \to \varphi(a_1 \times a_2) \end{aligned}$$

Now, for $x \preccurlyeq \varphi(a_1)$, by monotonicity of the application,

$$\begin{aligned} r(u\varphi(\mathtt{pair}^{\mathscr{A}})x &\preccurlyeq r(u\varphi(\mathtt{pair}^{\mathscr{A}})\varphi(a_1) \\ r(r(u\varphi(\mathtt{pair}^{\mathscr{A}})x) &\preccurlyeq r(r(u\varphi(\mathtt{pair}^{\mathscr{A}})\varphi(a_1)) \end{aligned}$$

so using the previous inequality

$$r(r(u\varphi(\mathtt{pair}^{\mathscr{A}})x) \preccurlyeq \varphi(a_2) \to \varphi(a_1 \times a_2)$$

and by adequacy of the rule of introduction of the $\to$

$$(\lambda x.r(r(u\varphi(\mathtt{pair}^{\mathscr{A}})x))^{\mathscr{A}} \preccurlyeq \varphi(a_1) \to \varphi(a_2) \to \varphi(a_1 \times a_2).$$

Let $t = (\lambda x.r(r(u\varphi(\mathtt{pair}^{\mathscr{A}})x))^{\mathscr{A}}$ for the sake of clarity. Then, using $\mathtt{projl}$ and $\mathtt{projr}$ we get, by adequacy :

$$(\lambda x.t(\mathtt{projl}x)(\mathtt{projr}x))^{\mathscr{A}} \preccurlyeq \varphi(a_1) \times \varphi(a_2) \to \varphi(a_1 \times a_2)$$

The following points are very similar to the first one, they are proved in Appendix A. ∎

5.3. **Action on the Induced Structures.**

5.3.1. *Heyting Algebras.* For two implicative algebras $\mathscr{A}$ and $\mathscr{B}$, let $\mathcal{H}_{\mathscr{A}} = \mathscr{A}/S_{\mathscr{A}}$ and $\mathcal{H}_{\mathscr{B}} = \mathscr{B}/S_{\mathscr{B}}$ be the Heyting algebra induced by $(\mathscr{A}, \vdash_{\mathscr{A}})$ and $(\mathscr{B}, \vdash_{\mathscr{B}})$. The two canonical surjections associated with these Heyting algebras are denoted by $\pi_{\mathscr{A}} : \mathscr{A} \to \mathcal{H}_{\mathscr{A}}$ and $\pi_{\mathscr{B}} : \mathscr{B} \to \mathcal{H}_{\mathscr{B}}$. Hence, each map $\varphi : \mathscr{A} \to \mathscr{B}$ factors into a map $\tilde{\varphi}$ such that

$$\begin{array}{ccc} \mathscr{A} & \xrightarrow{\varphi} & \mathscr{B} \\ {\scriptstyle \pi_{\mathscr{A}}}\downarrow & & \downarrow{\scriptstyle \pi_{\mathscr{B}}} \\ \mathcal{H}_{\mathscr{A}} & \xrightarrow[\tilde{\varphi}]{} & \mathcal{H}_{\mathscr{B}} \end{array}$$

commutes because axiom 1. of Definition 5.1 ensure that if two elements of $\mathscr{A}$ are equivalent (for $\dashv\vdash_{S_{\mathscr{A}}}$), their images are equivalent (for $\dashv\vdash_{S_{\mathscr{B}}}$). The map $\tilde{\varphi}$ is not a morphism of Heyting algebra in general but we can give a characterization of this map using the notion of morphism of bounded meet-semilattices

Definition 5.5. A map $\varphi : \mathcal{L} \to \mathcal{L}'$ where $\mathcal{L}$ and $\mathcal{L}'$ are bounded meet-semilattices is a *morphism of bounded meet-semilattices* if

- $\varphi(\top) = \top$
- for all $x, y \in \mathcal{L},\ \varphi(x \wedge y) = \varphi(x) \wedge \varphi(y)$.

Proposition 5.6. *A map $\varphi : \mathscr{A} \to \mathscr{B}$ factors into a morphism of bounded meet-semilattices $\tilde{\varphi}$ if and only if the following conditions are satisfied*

(i) $\varphi(S_{\mathscr{A}}) \subseteq \varphi(S_{\mathscr{B}})$

(ii) *for all* $a, a' \in \mathscr{A},\ (\varphi(a \to a') \to \varphi(a) \to \varphi(a')) \in S_{\mathscr{B}}$.

Remark 5.7. Here, the condition (ii) is weaker than the axiom 2. from Definition 5.1, because this condition is not uniform. Condition (ii) implies also a non-uniform version of axiom 3. of Definition 5.1. For $a, a' \in \mathscr{A}$ such that $a \preccurlyeq a'$, by (ii) there is $r'_{a,a'} \in S_{\mathscr{B}}$ such that $r'_{a,a'} \preccurlyeq \varphi(a \to a') \to \varphi(a) \to \varphi(a')$, so with $u_{a,a'} = r'_{a,a'}\varphi(a \to a') \in S_{\mathscr{B}}$, we get $u_{a,a'} \preccurlyeq \varphi(a) \to \varphi(a')$.

To prove this result we first prove a general lemma on Heyting algebras.

Lemma 5.8. *Let $\mathcal{H}$ and $\mathcal{H}'$ be two Heyting algebras and $f : \mathcal{H} \to \mathcal{H}'$ a map such that $f(\top_{\mathcal{H}}) = \top_{\mathcal{H}'}$. Then the two conditions*

1. *For all* $x, y \in \mathcal{H},\ f(x \to y) \leq f(x) \to f(y)$
2. *For all* $x, y \in \mathcal{H},\ f(x \wedge y) = f(x) \wedge f(y)$

are equivalent, and if f fullfils one of these conditions, f is monotonic.

Proof. For two Heyting algebras $\mathcal{H}$ and $\mathcal{H}'$ and $f : \mathcal{H} \to \mathcal{H}'$ such that $f(\top_{\mathcal{H}}) = \top_{\mathcal{H}'}$,

- $(1) \Rightarrow (2)$: Let $x, y \in \mathcal{H}$. First, let us show that f is monotonic : for $x, y \in \mathcal{H}$ such that $x \leq y$, we have $x \to y = \top_{\mathcal{H}'}$ so
$$\top_{\mathcal{H}'} = f(\top_{\mathcal{H}}) = f(x \to y) \leq f(x) \to f(y) \leq \top_{\mathcal{H}'}$$
Then, $f(x) \to f(y) = \top_{\mathcal{H}'}$ which implies that $f(x) \leq f(y)$: f is monotonic.
By monotonicity, $f(x \wedge y) \leq f(x)$ and $f(x \wedge y) \leq f(y)$ so $f(x \wedge y) \leq f(x) \wedge f(y)$. The reverse inequality uses the fact $x \to (x \wedge y) \geq y$ because it gives
$$f(x) \wedge f(y) = f(x) \wedge f(x \to x \wedge y) \leq f(x) \wedge (f(x) \to f(x \wedge y)) \leq f(x \wedge y)$$

- (2) $\Rightarrow$ (1) : Let $x, y \in \mathcal{H}$. We notice the equality (**) $x \wedge (x \to y) = x \wedge y$ because $x \wedge (x \to y) \leq x$, $x \wedge (x \to y) \leq y$ and if $z \leq x \wedge y$ then $x \wedge z \leq y$ so $z \leq x \to y$ by definition of $x \to y$ so $z \leq x \wedge (x \to y)$. We deduce the inequality:

$$f(x) \wedge f(x \to y) \underset{\text{by (2)}}{=} f(x \wedge (x \to y)) \underset{\text{by (**)}}{=} f(x \wedge y) \underset{\text{by (2)}}{=} f(x) \wedge f(y) \leq f(y)$$

  which corresponds exactly with (1).

■

We can now prove the wanted result.

Proof of Proposition 5.6. This technical proof is given in Appendix A. ■

Then, we can deduce an obvious corollary :

Corollary 5.9. *If $\varphi : \mathscr{A} \to \mathscr{B}$ is an applicative morphism, then φ factors into a morphism of bounded meet-semilattices $\tilde{\varphi}$.*

5.3.2. *Triposes.* Here, recalling the notations from Subsection 3.2, for two implicative algebras $\mathscr{A}$ and $\mathscr{B}$, for each index set I, $\mathsf{P}I = \mathscr{A}^I/S_{\mathscr{A}}[I]$ and $\mathsf{Q}I = \mathscr{B}^I/S_{\mathscr{B}}[I]$. The maps $\pi_I : \mathscr{A}^I \to \mathsf{P}I$ and $\omega_I : \mathscr{B}^I \to \mathsf{Q}I$ are the canonical surjections. Here, most of the maps that we will use are monotonic but in general they are not morphisms of Heyting algebras. Then, P and Q are here considered as functors whose destination is in **Pos** rather that in **HA**.

For a map $\varphi : \mathscr{A} \to \mathscr{B}$ and an index set I, the map $\varphi^I : \mathscr{A}^I \to \mathscr{B}^I$ is defined such that for all $(a_i)_{i\in I} \in \mathscr{A}^I$, $\varphi^I((a_i)_{i\in I}) = (\varphi(a_i))_{i\in I}$.

Now, we can give a full characterization in terms of triposes of the notion of applicative morphism.

Theorem 1. *A map $\varphi : \mathscr{A} \to \mathscr{B}$ is an applicative morphism if and only if for each index set I, φ^I factors through π_I and ω_I into a morphism of bounded meet-semilattices $\tilde{\varphi}$.*

Proof. Let $\varphi : \mathscr{A} \to \mathscr{B}$ be a map.

- If φ is an applicative morphism, and I is an index set, $\varphi^I : \mathscr{A}^I \to \mathscr{B}^I$ is also an applicative morphism because :

  1. For $(a_i)_{i\in I} \in S_{\mathscr{A}}[I]$ let $s \in S_{\mathscr{A}}$ such that $\forall i \in I,\ s \preccurlyeq a_i$, which implies that $u\varphi(s) \preccurlyeq \varphi(a_i)$ for all $i \in I$, and $u\varphi(s) \in S_{\mathscr{B}}$ because $u, \varphi(s) \in S_{\mathscr{B}}$ so $\varphi^I((a_i)) \in S_{\mathscr{B}}[I]$.
  2. With $r^I = (r)_{i\in I} \in S_{\mathscr{B}}[I]$, for $(a_i), (a'_i) \in \mathscr{A}^I$ and for each $i \in I$ we have :

$$r \preccurlyeq \varphi(a_i \to a'_i) \to \varphi(a_i) \to \varphi(a'_i)$$

     and

$$(\varphi(a_i \to a'_i) \to \varphi(a_i) \to \varphi(a'_i))_{i\in I} = \varphi^I((a_i)_{i\in I} \to (a'_i)_{i\in I}) \to \varphi^I((a_i)_{i\in I}) \to \varphi^I((a'_i)_{i\in I}).$$

     We can then conclude :

$$r^I \preccurlyeq \varphi^I((a_i)_{i\in I} \to (a'_i)_{i\in I}) \to \varphi^I((a_i)_{i\in I}) \to \varphi^I((a'_i)_{i\in I}).$$

  3. With $u^I = (u)_{i\in I} \in S_{\mathscr{B}}[I]$, using the same methods as in 2. we get that, for $(a_i), (a'_i) \in \mathscr{A}^I$ such that $(a_i) \preccurlyeq (a'_i)$

$$u^I \preccurlyeq \varphi^I((a_i)) \to \varphi^I((a'_i)).$$

By Corollary 5.9, $\tilde{\varphi}^I$ is a morphism of bounded meet-semilattices.

- Let us now suppose that for each index set I, $\tilde{\varphi}^I$ is a morphism of bounded meet-semilattices. It is easy to deduce that $\varphi(S_{\mathscr{A}}) \subseteq S_{\mathscr{B}}$ using a set with only one element and the Proposition 5.6 and the fact that if I has only one element then $S_{\mathscr{A}}[I] = S_{\mathscr{A}}$, $S_{\mathscr{B}}[I] = S_{\mathscr{B}}$ and $\varphi^I = \varphi$. Now let $I = \mathscr{A} \times \mathscr{A}$. We define now $(a_i)_{i\in I}$ by $a_{(a_1,a_2)} = a_1 \in \mathscr{A}$ and $(a'_i)_{i\in I}$ by $a'_{(a_1,a_2)} = a_2 \in \mathscr{A}$. Moreover,

$$\varphi^I((a_i)_{i\in I} \to (a'_i)_{i\in I}) \to \varphi^I((a_i)_{i\in I}) \to \varphi^I((a'_i)_{i\in I}) = (\varphi(a_i \to a'_i) \to \varphi(a_i) \to \varphi(a'_i))_{i\in I}$$

and $\tilde{\varphi}^I$ is a morphism of bouded meet-semilattices so, by Proposition 5.6 :

$$(\varphi(a_i \to a'_i) \to \varphi(a_i) \to \varphi(a'_i))_{i\in I} \in S_{\mathscr{B}}[I].$$

Then, by definition of $S_{\mathscr{B}}[I]$, there is $r \in S_{\mathscr{B}}$ such that

$$\forall i \in I,\ r \preccurlyeq \varphi(a_i \to a'_i) \to \varphi(a_i) \to \varphi(a'_i)$$

which is equivalent, with $i = (a, a')$ to

$$\forall a, a' \in \mathscr{A},\ r \preccurlyeq \varphi(a \to a') \to \varphi(a) \to \varphi(a').$$

Finally, let $J = \{(a, a') \in \mathscr{A} \times \mathscr{A} : a \preccurlyeq a'\}$ and $(a_i)_{i\in J}$, $(a'_i)_{i\in J}$ such that $a_{(a_1,a_2)} = a_1$ and $a'_{(a_1,a_2)} = a_2$. Using Remark 5.7 and the fact that $\tilde{\varphi}^J$ is a morphism of bounded meet-semilattices we have :

$$\varphi^J((a_i)_{i\in J}) \to \varphi^J((a'_i)_{i\in J}) = (\varphi(a_i) \to \varphi(a'_i))_{i\in J} \in S_{\mathscr{B}}[J].$$

Hence there is $u \in S_{\mathscr{B}}$ such that :

$$\forall a, a' \in (\preccurlyeq_{\mathscr{A}}),\ u \preccurlyeq \varphi(a) \to \varphi(a').$$

■

5.4. **Categorical Result.** As for naive morphsisms, if **IA** is the category of implicative algebras where the morphisms are applicative morphisms, this category should be Cartesian. We prove a stronger result.

Theorem 2. *The category* **IA** *has all small products.*

Proof. First, for an index set I and implicative algebras $(\mathscr{A}_i, \to_i, S_i)$ for all $i \in I$ we define the product $(\prod_{i\in I} \mathscr{A}_i, \to, \prod_{i\in I} S_i)$ where the product $\prod_{i\in I} \mathscr{A}_i$ and $\prod_{i\in I} S_i$ are usual products in **Set**, the arrow is defined such that $(a_i)_{i\in I} \to (a'_i)_{i\in I} = (a_i \to_i a'_i)_{i\in I}$ and $(a_i)_{i\in I} \preccurlyeq (a'_i)_{i\in I}$ if and only if for all $i \in I$, $a_i \preccurlyeq a'_i$. This is an implicative algebra because

(1) for $(a'_i)_{i\in I} \preccurlyeq (a_i)_{i\in I}$ and $(b_i)_{i\in I} \preccurlyeq (b'_i)_{i\in I}$, for $i \in I$, $(a_i \to b_i) \preccurlyeq (a'_i \to b'_i)$ so $((a_i)_{i\in I} \to (b_i)_{i\in I}) \preccurlyeq ((a'_i)_{i\in I} \to (b'_i)_{i\in I})$

(2) for $(a_i)_{i\in I}$ and $B_i \subseteq \mathscr{A}_i$ for all i,

$$(a_i)_{i\in I} \to \bigcurlywedge_{(b_i)_{i\in I}\in\prod_{i\in I} B_i} (b_i)_{i\in I} = (\bigcurlywedge_{b_i\in B_i} (a_i \to_i b_i)) = \bigcurlywedge_{(b_i)_i\in\prod_{i\in I} B_i} ((a_i)_{i\in I} \to (b_i)_{i\in I})$$

so $\prod_{i\in I} \mathscr{A}_i$ is an applicative structure. For the sake of clarity, let $\mathscr{A} = \prod_{i\in I} \mathscr{A}_i$.

By Lemma 4.5, $\prod_{i \in I} S_i$ is a separator of $\mathscr{A}$.

Now, let us define, for all $j \in I$ a map $\pi_j : \mathscr{A} \to \mathscr{A}_j$ such that for all $(a_i)_{i\in I} \in \mathscr{A}$ we have $\pi_j((a_i)_{i\in I}) = a_j$. We need to show that these maps are applicative morphisms : first $\pi_j(\prod_{i\in I} S_i) = S_j$, then for $(a_i)_{i\in I}, (a'_i)_{i\in I} \in \mathscr{A}$ we have $\mathbf{I}^{\mathscr{A}_j} \in S_j$ so

$$\mathbf{I}^{\mathscr{A}_j} \preccurlyeq (a_j \to a'_j) \to a_j \to a'_j = \pi_j((a_i)_{i\in I} \to (a'_i)_{i\in I}) \to \pi_j((a_i)_{i\in I}) \to \pi_j((a'_i)_{i\in I})$$

and if $(a_i)_{i\in I} \preccurlyeq (a'_i)_{i\in I}$ we have

$$\mathbf{I}^{\mathscr{A}_j} \preccurlyeq a_j \to a'_j = \pi_j((a_i)_{i\in I}) \to \pi_j((a'_i)_{i\in I})$$

Finally, we need to prove that we have the universal property : let $\mathscr{B}$ be an implicative algebra and for all $j \in I$, $\varphi_j : \mathscr{B} \to \mathscr{A}_j$ is an applicative morphism. Then, let $\varphi : \mathscr{B} \to \mathscr{A}$ be a map such that $\varphi(b) = (\varphi_i(b))_{i\in I}$. Then φ is an applicative morphism because every φ_j is also one and for each $j \in I$

$$\begin{tikzcd}
& \mathscr{B} \arrow[dl, "\varphi_j"'] \arrow[dr, "\varphi"] & \\
\mathscr{A}_j & & \mathscr{A} \arrow[ll, "\pi_j"]
\end{tikzcd}$$

commutes and if an other morphism ϕ has the same properties, for all $b \in \mathscr{B}$ we have $\phi(b) = (\varphi_i(b))_{i\in I} = \varphi(b)$. Hence the algebra defined is a product. ■

6 Other Possible Definitions

Applicative morphisms is the best definition for morphisms of implicative algebras so far, but it is still possible to imagine a better definition. As we saw in the Theorem 1, applicative morphisms are exactly the maps inducing morphisms of bounded meet-semilattices. Hence, it is natural to search which maps induce well known types of morphisms between Heyting algebras.

6.1. **Natural Ideas.** The first natural idea is to study the monotonic maps. The order is an essential property in the stuctures that we use, because it is a caracterisation of the level of truth. Hence, it is important to characterize what are the maps inducing monotonic maps.

Proposition 6.1. *A map $\varphi : \mathscr{A} \to \mathscr{B}$ induces a family of monotonic maps if and only if*

$$\forall s \in S_{\mathscr{A}}, \exists s' \in S_{\mathscr{B}}, \forall a, a' \in \mathscr{A},\ s \preccurlyeq a \to a' \ \Rightarrow\ s' \preccurlyeq \varphi(a) \to \varphi(a')$$

Proof. Let $\varphi : \mathscr{A} \to \mathscr{B}$ be a map and $(\tilde{\varphi}_I)_I$ the family induced by φ. Describing this property in the implicative algebra $\mathscr{A}^I$ we get :

$$\begin{aligned}
&\forall I, \forall (a_i)_{i\in I}, (a'_i)_{i\in I} \in \mathscr{A}^I,\ [(a_i)_{i\in I}] \preccurlyeq \left[(a'_i)_{i\in I}\right] \ \Rightarrow\ [(\varphi(a_i))_{i\in I}] \preccurlyeq \left[(\varphi(a'_i))_{i\in I}\right] \\
\Leftrightarrow &\forall I, \forall (a_i)_{i\in I}, (a'_i)_{i\in I} \in \mathscr{A}^I, \forall s \in S_{\mathscr{A}}, (\forall i \in I,\ s \preccurlyeq a_i \to a'_i) \Rightarrow \left[(\varphi(a_i) \to \varphi(a'_i))_{i\in I}\right] \in S_{\mathscr{B}}[I] \\
\Leftrightarrow &\forall I, \forall s \in S_{\mathscr{A}} \exists s' \in S_{\mathscr{B}}, \forall (a_i)_{i\in I}, (a'_i)_{i\in I} \in \mathscr{A}^I,\ (\forall i \in I,\ s \preccurlyeq a_i \to a'_i) \Rightarrow (\forall i \in I,\ s' \preccurlyeq \varphi(a_i) \to \varphi(a'_i)) \\
\Leftrightarrow &\forall s \in S_{\mathscr{A}}, \exists s' \in S_{\mathscr{B}}, \forall a, a' \in \mathscr{A},\ s \preccurlyeq a \to a' \ \Rightarrow s' \preccurlyeq \varphi(a) \to \varphi(a')
\end{aligned}$$

which prove this result. ■

This proposition is a first step of a study that is not done here : the next step is to use this characterization to explore the category of implicative algebras where the morphisms are the maps inducing monotonic maps.

An other natural idea is to study the maps inducing Heyting algebra morphisms.

Proposition 6.2. *A map $\varphi : \mathscr{A} \to \mathscr{B}$ induces a family of morphisms of Heyting algebras if and only if φ is an applicative morphism and there are elements*

$$\mathtt{fold}_{\to},\ \mathtt{unfold}_{+} \in S_{\mathscr{B}}$$

such that

$$\mathtt{fold}_{\to} \preccurlyeq (\varphi(a_1) \to \varphi(a_2)) \to \varphi(a_1 \to a_2)$$

$$\mathtt{unfold}_{+} \preccurlyeq \varphi(a_1 + a_2) \to \varphi(a_1) + \varphi(a_2)$$

for all $a_1, a_2 \in \mathscr{A}$.

Proof. Let $\varphi : \mathscr{A} \to \mathscr{B}$ be a map and $(\tilde{\varphi}^I)_I$ be the induced family. For each $(a_i)_{i\in I}, (a'_i)_{i\in I} \in \mathscr{A}^I$ we have :

$$\tilde{\varphi}^I([(a_i) \to (a'_i)]) = [(\varphi(a_i \to a'_i))] \quad \text{and} \quad \tilde{\varphi}^I([(a_i)]) \to \tilde{\varphi}^I([(a'_i)]) = [(\varphi(a_i) \to \varphi(a'_i))]$$

We can then deduce :

$$\begin{array}{lll}
 & \forall (a_i), (a'_i) \in \mathscr{A}^I, & \tilde{\varphi}^I([(a_i) \to (a'_i)]) = \tilde{\varphi}^I([(a_i)]) \to \tilde{\varphi}^I([(a'_i)]) \\
\Leftrightarrow & \forall (a_i), (a'_i) \in \mathscr{A}^I,\ \forall i \in I, & \varphi(a_i \to a'_i) \to \varphi(a_i) \to \varphi(a'_i) \in S_{\mathscr{B}}[I] \\
 & \text{and} & (\varphi(a_i) \to \varphi(a'_i)) \to \varphi(a_i \to a'_i) \in S_{\mathscr{B}}[I] \\
\Leftrightarrow & \exists s_1, s_2 \in S_{\mathscr{B}},\ \forall a, a' \in \mathscr{A} & (1)\ s_1 \preccurlyeq \varphi(a \to a') \to \varphi(a) \to \varphi(a') \\
 & & (2)\ s_2 \preccurlyeq (\varphi(a) \to \varphi(a')) \to \varphi(a \to a')
\end{array}$$

by the same methods we have

$$\begin{array}{lll}
 & \forall (a_i), (a'_i) \in \mathscr{A}^I, & \tilde{\varphi}^I([(a_i) \wedge (a'_i)]) = \tilde{\varphi}^I([(a_i)]) \wedge \tilde{\varphi}^I([(a'_i)]) \\
\Leftrightarrow & \forall (a_i), (a'_i) \in \mathscr{A}^I,\ \forall i \in I, & \varphi(a_i \times a'_i) \to \varphi(a_i) \times \varphi(a'_i) \in S_{\mathscr{B}}[I] \\
 & \text{and} & (\varphi(a_i) \times \varphi(a'_i)) \to \varphi(a_i \times a'_i) \in S_{\mathscr{B}}[I] \\
\Leftrightarrow & \exists s_3, s_4 \in S_{\mathscr{B}},\ \forall a, a' \in \mathscr{A} & (1)\ s_3 \preccurlyeq \varphi(a \times a') \to \varphi(a) \times \varphi(a') \\
 & & (2)\ s_4 \preccurlyeq (\varphi(a) \times \varphi(a')) \to \varphi(a \times a')
\end{array}$$

and

$$\begin{array}{lll}
 & \forall (a_i), (a'_i) \in \mathscr{A}^I, & \tilde{\varphi}^I([(a_i) \vee (a'_i)]) = \tilde{\varphi}^I([(a_i)]) \vee \tilde{\varphi}^I([(a'_i)]) \\
\Leftrightarrow & \forall (a_i), (a'_i) \in \mathscr{A}^I,\ \forall i \in I, & \varphi(a_i + a'_i) \to \varphi(a_i) + \varphi(a'_i) \in S_{\mathscr{B}}[I] \\
 & \text{and} & (\varphi(a_i) + \varphi(a'_i)) \to \varphi(a_i + a'_i) \in S_{\mathscr{B}}[I] \\
\Leftrightarrow & \exists s_5, s_6 \in S_{\mathscr{B}},\ \forall a, a' \in \mathscr{A} & (1)\ s_5 \preccurlyeq \varphi(a + a') \to \varphi(a) + \varphi(a') \\
 & & (2)\ s_6 \preccurlyeq (\varphi(a) + \varphi(a')) \to \varphi(a + a')
\end{array}$$

Moreover, if for each index set I, $\tilde{\varphi}^I$ is a morphism of Heyting algebras then φ is an applicative morphism by Theorem 1. Then the elements s_1, s_3, s_4 and s_6 exist by Proposition 2.5. Hence $(\tilde{\varphi}^I)$ is a family of morphisms of Heyting algebras if and only if φ is an applicative morphism such that there are $\mathtt{fold}_{\to}, \mathtt{unfold}_{+} \in S_{\mathscr{B}}$ such that

$$\mathtt{fold}_{\to} \preccurlyeq (\varphi(a_1) \to \varphi(a_2)) \to \varphi(a_1 \to a_2)$$

$$\mathtt{unfold}_{+} \preccurlyeq \varphi(a_1 + a_2) \to \varphi(a_1) + \varphi(a_2)$$

for all $a_1, a_2 \in \mathscr{A}$. ■

Here we have a characterization as well. The maps inducing Heyting algebra morphisms are obviously included in the applicative morphisms, so it could be useful to define the morphisms of implicative algebras like this if applicative morphisms do not fullfil some important logical properties because a stronger definition like this one could solve this potential issue.

Conclusion

Various definitions of morphisms of implicative algebras are possible, so the category of implicative algebras has also a lot a definitions. Applicative morphisms are fully characterized in terms of triposes, and have a lot of commutation properties. In the future, it may be interesting to study more precisely the category with this definition of morphisms, for exemple we proved that this category is Cartesian but the question to know if it is complete or not is still open.

Other possible definitions may be considered. The maps inducing monotonic maps or Heyting algebra morphisms are characterized here but it may be important to study commutation properties fullfiled by these definitions.

Finally, an other possible idea that is not presented before is the notion of applicative dense morphisms. Coming from [vOZ16] like applicative morphisms, it is possible to adapt the definition to implicative algebras and it may be interesting to compare the results from this definition and those from applicative morphisms.

References

[Coh63] Paul J. Cohen. The independence of the continuum hypothesis. *Proceedings of the National Academy of Sciences*, 50(6):1143–1148, 1963.

[FM17] Walter Ferrer and Octavio Malherbe. The category of implicative algebras and realizability. 12 2017.

[Gir72] J.Y. Girard. *Interprétation fonctionnelle et élimination des coupures de l'arithmétique d'ordre supérieur.* Éditeur inconnu, 1972.

[GTL89] Jean-Yves Girard, Paul Taylor, and Yves Lafont. *Proofs and Types.* Cambridge University Press, USA, 1989.

[HJP80] J. M. E. Hyland, P. T. Johnstone, and A. M. Pitts. Tripos theory. *Mathematical Proceedings of the Cambridge Philosophical Society*, 88(2):205–232, 1980.

[Hof06] Pieter J. W. Hofstra. All realizability is relative. *Mathematical Proceedings of the Cambridge Philosophical Society*, 141(2):239–264, 2006.

[Kle45] S. C. Kleene. On the interpretation of intuitionistic number theory. *J. Symbolic Logic*, 10(4):109–124, 12 1945.

[Kri09] Jean-Louis Krivine. Realizability in classical logic. *Panoramas et synthèses*, 27:197–229, 2009.

[Miq20] Alexandre Miquel. Implicative algebras: a new foundation for realizability and forcing. *Mathematical Structures in Computer Science*, 30(5):458–510, 2020.

[Str13] Thomas Streicher. Krivine's classical realizability from a categorical perspective. *Mathematical Structures in Computer Science*, 23, 12 2013.

[VO02] Jaap Van Ooster. Realizability: a historical essay. *Mathematical Structures in Computer Science*, 12(3):239–263, 2002.

[vOZ16] Jaap van Oosten and Tingxiang Zou. Classical and relative realizability, 2016.

A Technical Proofs

We already proved one point of Proposition 5.3. The rest of the proof is given here

Proof of Proposition 5.3.

- $\texttt{unfold}_\times$: Now, for a_1 and a_2 in $\mathscr{A}$,

$$\begin{aligned} \texttt{projl}^{\mathscr{A}} &\preccurlyeq a_1 \times a_2 \to a_1 \\ u\varphi(\texttt{projl}^{\mathscr{A}}) &\preccurlyeq \varphi(a_1 \times a_2 \to a_1) \\ r'u\varphi(\texttt{projl}^{\mathscr{A}}) &\preccurlyeq \varphi(a_1 \times a_2) \to \varphi(a_1) \end{aligned}$$

  and the same calculus with $\texttt{projr}$ gives

$$r'u\varphi(\texttt{projr}^{\mathscr{A}}) \preccurlyeq \varphi(a_1 \times a_2) \to \varphi(a_2)$$

  Then, by adequacy, we get

$$(\lambda x.\texttt{pair}(r'u\varphi(\texttt{projl}^{\mathscr{A}})x)(r'u\varphi(\texttt{projr}^{\mathscr{A}})x))^{\mathscr{A}} \preccurlyeq \varphi(a_1 \times a_2) \to \varphi(a_1) \times \varphi(a_2)$$

  .

- $\texttt{fold}_+$: Here, for a_1 and a_2 in $\mathscr{A}$,

$$\begin{aligned} \texttt{inl}^{\mathscr{A}} &\preccurlyeq a_1 \to a_1 + a_2 \\ u\varphi(\texttt{inl}^{\mathscr{A}}) &\preccurlyeq \varphi(a_1 \to a_1 + a_2) \\ r'u\varphi(\texttt{inl}^{\mathscr{A}}) &\preccurlyeq \varphi(a_1) \to \varphi(a_1 + a_2) \end{aligned}$$

  and similarly

$$r'u\varphi(\texttt{inr}^{\mathscr{A}}) \preccurlyeq \varphi(a_2) \to \varphi(a_1 + a_2)$$

  Hence,

$$(\lambda x.\texttt{case}x(r'u\varphi(\texttt{inl}^{\mathscr{A}}))(r'u\varphi(\texttt{inr}^{\mathscr{A}})))^{\mathscr{A}} \preccurlyeq \varphi(a_1) + \varphi(a_2) \to \varphi(a_1 + a_2)$$

- $\texttt{unfold}_\forall$: Let $\texttt{unfold}_\forall = u$ and $(a_i)_{i\in I} \in \mathscr{A}^I$. Then,

$$\forall i \in I,\ \texttt{unfold}_\forall \preccurlyeq \varphi(\bigcurlywedge_{i\in I} a_i) \to \varphi(a_i) \qquad \text{(because } \textstyle\bigcurlywedge_{i\in I} a_i \preccurlyeq a_i\text{)}$$

  which implies that

$$\texttt{unfold}_\forall \preccurlyeq \bigcurlywedge_{i\in I}(\varphi(\bigcurlywedge_{i'\in I} a'_i) \to \varphi(a_i)) = \varphi(\bigcurlywedge_{i\in I} a_i) \to \bigcurlywedge_{i\in I} \varphi(a_i)$$

- $\texttt{fold}_\top$: By axiom 1. of Definition 5.1, $\varphi(\top) \in S_{\mathscr{B}}$, and $\varphi(\top) \to \top \to \varphi(\top) \succcurlyeq \mathbf{K}^{\mathscr{B}} \in S_{\mathscr{B}}$, then by adjunction of the application, $\top \to \varphi(\top) \succcurlyeq \mathbf{K}^{\mathscr{B}}\varphi(\top) \in S_{\mathscr{B}}$.

- $\texttt{unfold}_\top$: Let $\texttt{unfold}_\top = \top$. Then $\texttt{unfold}_\top = \top = \varphi(\top) \to \top$ and $\texttt{unfold}_\top \in S_{\mathscr{B}}$.

- $\texttt{fold}_\bot$: We have the inequality $u \preccurlyeq \varphi(\bot) \to \varphi(\bot) \preccurlyeq \bot \to \varphi(\bot)$. Then, with $\texttt{fold}_\bot = u$ we have the wanted result.

∎

Proof of Proposition 5.6. Let $\varphi : \mathscr{A} \to \mathscr{B}$ be a map.

- When $\tilde{\varphi}$ is a morphism of bounded semi-lattices, if $a \in S_{\mathscr{A}}$ we have :

$$\pi_{\mathscr{B}}(\varphi(a)) = \tilde{\varphi}(\pi_{\mathscr{A}}(a)) \underset{a \in S_{\mathscr{A}}}{=} \tilde{\varphi}(\pi_{\mathscr{A}}(\top_{\mathscr{A}})) = \tilde{\varphi}(\top_{H_{\mathscr{A}}}) = \top_{H_{\mathscr{B}}} = S_{\mathscr{B}}$$

  Then, $\varphi(a) \in S_{\mathscr{B}}$ so we can deduce (i).
  By Lemma 5.8,

$$\forall a, a' \in \mathscr{A},\ \tilde{\varphi}(\pi_{\mathscr{A}}(a \to a')) \leq_{S_{\mathscr{B}}} \tilde{\varphi}(\pi_{\mathscr{A}}(a)) \to \tilde{\varphi}(\pi_{\mathscr{A}}(a'))$$

  which implies that :

$$\forall a, a' \in \mathscr{A},\ \pi_{\mathscr{B}}(\varphi(a \to a')) \leq_{S_{\mathscr{B}}} \pi_{\mathscr{B}}(\varphi(a)) \to \pi_{\mathscr{B}}(\varphi(a'))$$

  Then, by definition of the order $\leq_{S_{\mathscr{B}}}$, we have :

$$\forall a, a' \in \mathscr{A},\ \varphi(a \to a') \to \varphi(a) \to \varphi(a') \in S_{\mathscr{B}}$$

- Let us now assume that φ satisfies conditions (i) and (ii).
  By (i), $\varphi(\top_{\mathscr{A}}) \in S_{\mathscr{B}}$, then

$$\tilde{\varphi}(\top_{H_{\mathscr{A}}}) = \tilde{\varphi}(\pi_{\mathscr{A}}(\top_{\mathscr{A}})) = \pi_{\mathscr{B}}(\varphi(\top_A)) = S_{\mathscr{B}} = \top_{H_{\mathscr{B}}}$$

  For $\alpha_1, \alpha_2 \in H_{\mathscr{A}}$, let $a, a' \in \mathscr{A}$ such that $\pi_{\mathscr{A}}(a) = \alpha_1$ and $\pi_{\mathscr{A}}(a') = \alpha_1$. The condition (ii) implies that :

$$\pi_{\mathscr{B}}(\varphi(a \to a')) \leq_{S_{\mathscr{B}}} \pi_B(\varphi(a) \to \varphi(a'))$$

  then,

$$\tilde{\varphi}(\alpha_1 \to \alpha_2) \leq_{S_{\mathscr{B}}} \tilde{\varphi}(\alpha_1) \to \tilde{\varphi}(\alpha_2)$$

  We can now conclude by Lemma 5.8 that $\tilde{\varphi}$ is a morphism of bounded meet-semilattices.

■