Revisiting the Duality of Computation: An Algebraic Analysis of Classical Realizability Models

Étienne Miquey
Équipe Gallinette, INRIA
LS2N, Université de Nantes, France
etienne.miquey@inria.fr

Abstract

In an impressive series of papers, Krivine showed at the edge of the last decade how classical realizability provides a surprising technique to build models for classical theories. In particular, he proved that classical realizability subsumes Cohen's forcing, and even more, gives rise to unexpected models of set theories. Pursuing the algebraic analysis of these models that was first undertaken by Streicher, Miquel recently proposed to lay the algebraic foundation of classical realizability and forcing within new structures which he called *implicative algebras*. These structures are a generalization of Boolean algebras based on an internal law representing the implication. Notably, implicative algebras allow for the adequate interpretation of both programs (i.e. proofs) and their types (i.e. formulas) in the same structure.

The very definition of implicative algebras takes position on a presentation of logic through universal quantification and the implication and, computationally, relies on the call-by-name λ-calculus. In this paper, we investigate the relevance of this choice, by introducing two similar structures. On the one hand, we define *disjunctive algebras*, which rely on internal laws for the negation and the disjunction and which we show to be particular cases of implicative algebras. On the other hand, we introduce *conjunctive algebras*, which rather put the focus on conjunctions and on the call-by-value evaluation strategy. We finally show how disjunctive and conjunctive algebras algebraically reflect the well-known duality of computation between call-by-name and call-by-value.

2012 ACM Subject Classification Theory of computation → Logic; Theory of computation → Proof theory; Theory of computation → Type theory

Keywords and phrases realizability, model theory, forcing, proofs-as-programs, λ-calculus, classical logic, duality, call-by-value, call-by-name, lattices, tripos

Digital Object Identifier 10.4230/LIPIcs.CSL.2020.30

Related Version An extended version of this paper including proofs and further details is available at: `https://hal.archives-ouvertes.fr/hal-02305560`.

Funding This research was partially funded by the ANII research project FCE_1_2014_1_104800.

Acknowledgements The author would like to thank Alexandre Miquel to which several ideas in this paper, especially the definition of conjunctive separators, should be credited.

1 Introduction

It is well-known since Griffin's seminal work [13] that a classical Curry-Howard correspondence can be obtained by adding control operators to the λ-calculus. Several calculi were born from this idea, amongst which Krivine λ_c-calculus [20], defined as the λ-calculus extended with Scheme's `call/cc` operator (for *call-with-current-continuation*). Elaborating on this calculus, Krivine's developed in the late 90s the theory of *classical realizability* [20], which is a complete reformulation of its intuitionistic twin. Originally introduced to analyze the computational content of classical programs, it turned out that classical realizability also provides interesting semantics for classical theories. While it was first tailored to Peano second-order arithmetic (i.e. second-order type systems), classical realizability actually scales to more complex classical theories like ZF [21], and gives rise to surprisingly new models. In particular, its generalizes Cohen's forcing [21, 30] and allows for the direct definition of a model in which neither the continuum hypothesis nor the axiom of choice holds [23].

Algebraization of classical realizability. During the last decade, the algebraic structure of the models that classical realizability induces has been actively studied. This line of work was first initiated by Streicher, who proposed the concept of *abstract Krivine structure* [38], followed among others by Ferrer, Frey, Guillermo, Malherbe and Miquel who introduced other structures peculiar to classical realizability [8, 9, 6, 10, 11, 40]. In addition to the algebraic study of classical realizability models, these works had the interest of building the bridge with the algebraic structures arising from intuitionistic realizability. In particular, Streicher showed in [38] how classical realizability could be analyzed in terms of *triposes* [37], the categorical framework emerging from intuitionistic realizability models, while the later work of Ferrer et al. [8, 9] connected it to Hofstra and Van Oosten's notion of *ordered combinatory algebras* [16]. More recently, Alexandre Miquel introduced the concept of *implicative algebra* [31], which appear to encompass the previous approaches and which we present in this paper.

Implicative algebras. In addition to providing an algebraic framework conducive to the analysis of classical realizability, an important feature of implicative structures is that they allow us to identify *realizers* (i.e. λ-terms) and *truth values* (i.e. formulas). Concretely, implicative structures are complete lattices equipped with a binary operation $a \to b$ satisfying properties coming from the logical implication. As we will see, they indeed allow us to interpret both the formulas and the terms in the same structure. For instance, the ordering relation $a \preccurlyeq b$ will encompass different intuitions depending on whether we regard a and b as formulas or as terms. Namely, $a \preccurlyeq b$ will be given the following meanings:

- the formula a is a *subtype* of the formula b;
- the term a is a *realizer* of the formula b;
- the realizer a is *more defined* than the realizer b.

In terms of the Curry-Howard correspondence, this means that we not only identify types with formulas and proofs with programs, but *we also identify types and programs.*

Side effects. Following Griffin's discovery on control operators and classical logic, several works have renewed the observation that within the proofs-as-programs correspondence, with side effects come new reasoning principles [19, 18, 29, 14, 17]. More generally, it is now clear that computational features of a calculus may have consequences on the models it induces. For instance, computational proofs of the axiom of dependent choice can be obtained by adding a `quote` instruction [19], using memoisation [15, 33] or with a bar recursor [25]. Yet, such choices may also have an impact on the structures of the corresponding realizability models: the non-deterministic operator $\pitchfork$ is known to make the model collapse on a forcing situation [22], while the bar recursor requires some continuity properties [25].

If we start to have a deep understanding of the algebraic structure of classical realizability models, the algebraic counterpart of side effects on these structures is still unclear. As a first step towards this problem, it is natural to wonder: does the choice of an evaluation strategy have algebraic consequences on realizability models? This paper aims at bringing new tools for addressing this question.

Outline of the paper. We start by recalling the definition of Miquel's implicative algebras and their main properties in Section 2. We then introduce the notion of *disjunctive algebras* in Section 3, which naturally arises from the negative decomposition of the implication $A \to B = \neg A \parr B$. We explain how this decomposition induces realizability models based on a call-by-name fragment of Munch-Maccagnoni's system L [35], and we show that disjunctive algebras are in fact particular cases of implicative algebras. In Section 4, we explore the positive dual decomposition $A \to B = \neg(A \otimes \neg B)$, which naturally corresponds to a call-by-value fragment of system L. We show the corresponding realizability models naturally induce a notion of *conjunctive algebras*. Finally, in Section 5 we revisit the well-known duality of computation through this algebraic structures. In particular, we show how to pass from conjunctive to disjunctive algebras and vice-versa, while inducing isomorphic triposes.

Most of the proofs have been formalized in the Coq proof assistant, in which case their statements include hyperlinks to their formalizations[1].

2 Implicative algebras

2.1 Krivine classical realizability in a glimpse

We give here an overview of the main characteristics of Krivine realizability and of the models it induces[2]. Krivine realizability models are usually built above the λ_c-calculus, a language of abstract machines including a set of terms Λ and a set of stacks Π (i.e. evaluation contexts). Processes $t \star \pi$ in the abstract machine are given as pairs of a term t and a stack π.

Krivine realizability interprets a formula A as a set of closed terms $|A| \subseteq \Lambda$, called the *truth value* of A, and whose elements are called the *realizers* of A. Unlike in intuitionistic realizability models, this set is actually defined by orthogonality to a *falsity value* $\|A\|$ made of stacks, which intuitively represents a set of opponents to the formula A. Realizability models are parameterized by a pole $\perp\!\!\!\perp$, a set of processes in the underlying abstract machine which somehow plays the role of a referee betweens terms and stacks. The pole allows us to define the orthogonal set $X^{\perp\!\!\!\perp}$ of any falsity value $X \subseteq \Pi$ by: $X^{\perp\!\!\!\perp} \triangleq \{t \in \Lambda : \forall \pi \in X, t \star \pi \in \perp\!\!\!\perp\}$. Valid formulas A are then defined as the ones admitting a proof-like *realizer*[3] $t \in |A|$.

Before defining implicative algebras, we would like to draw the reader's attention on an important observation about realizability: there is an omnipresent lattice structure, which is reminiscent of the concept of subtyping [3]. Given a realizability model it is indeed always possible to define a semantic notion of subtyping: $A \preccurlyeq B \triangleq \|B\| \subseteq \|A\|$. This informally reads as *"A is more precise than B"*, in that A admits more opponents than B. In this case, the relation $\preccurlyeq$ being induced from (reversed) set inclusions comes with a richer structure of complete lattice, where the meet $\wedge$ is defined as a union and the join $\vee$ as an intersection. In particular, the interpretation of a universal quantifier $\|\forall x.A\|$ is given by an union $\bigcup_{n\in\mathbb{N}} \|A[n/x]\| = \curlywedge_{n\in\mathbb{N}} \|A[n/x]\|$, while the logical connective $\wedge$ is interpreted as the type of pairs $\times$ i.e. with a computational content. As such, *realizability* corresponds to the following picture: $\boxed{\forall = \curlywedge \qquad \wedge = \times}$. This is to compare with *forcing*, that can be expressed in terms of Boolean algebras where both the universal quantifier and the conjunction are interpreted by meets without any computational content: $\boxed{\forall = \wedge = \curlywedge}$ [1].

1 Available at `https://gitlab.com/emiquey/ImplicativeAlgebras/`
2 For a detailed introduction on this topic, we refer the reader to [20] or [32].
3 One specificity of Krivine classical realizability is that the set of terms contains the control operator `cc` and continuation constants $\boldsymbol{k}_\pi$. Therefore, to preserve the consistency of the induced models, one has to consider only proof-like terms, i.e. terms that do not contain any continuations constants see [20, 32].

2.2 Implicative algebras

Implicative structures are tailored to represent both the formulas of second-order logic and realizers arising from Krivine's λ_c-calculus. For their logical facet, they are defined as meet-complete lattices (for the universal quantification) with an internal binary operation satisfying the properties of the implication:

▶ **Definition 1.** *An* implicative structure *is a complete lattice* $(\mathcal{A}, \preccurlyeq)$ *equipped with an operation* $(a, b) \mapsto (a \to b)$*, such that for all* $a, a_0, b, b_0 \in \mathcal{A}$ *and any subset* $B \subseteq \mathcal{A}$*:*
1. *If* $a_0 \preccurlyeq a$ *and* $b \preccurlyeq b_0$ *then* $(a \to b) \preccurlyeq (a_0 \to b_0)$.
2. $\bigcurlywedge_{b\in B}(a \to b) = a \to \bigcurlywedge_{b\in B} b$

It is then immediate to embed any closed formula of second-order logic within any implicative structure. Obviously, any complete Heyting algebra or any complete Boolean algebra defines an implicative structure with the canonical arrow. More interestingly, any ordered combinatory algebras, a structure arising naturally from realizability [16, 39, 38, 7], also induces an implicative structure [34]. Last but not least, any classical realizability model induces as expected an implicative structure on the lattice $(\mathcal{P}(\Pi), \supseteq)$ by considering the arrow defined by[4]: $a \to b \triangleq a^{\perp\!\!\perp} \cdot b = \{t \cdot \pi : t \in a^{\perp\!\!\perp}, \pi \in b\}$ ([31, 34].

Interestingly, if any implicative structure $\mathcal{A}$ trivially provides us with an embedding of second-order formulas, we can also encode λ-terms with the following definitions:

$$ab \triangleq \bigcurlywedge\{c : a \preccurlyeq b \to c\} \qquad\qquad \lambda f \triangleq \bigcurlywedge_{a\in\mathcal{A}} (a \to f(a))$$

In both cases, one can understand the meet as a conjunction of all the possible approximations of the desired term. From now on, we will denote by $t^{\mathcal{A}}$ (resp. $A^{\mathcal{A}}$) the interpretation of the closed λ-term t (resp. formula A). Notably, these embeddings are at the same time:
1. Sound with respect to the β-reduction, in the sense that $(\lambda f)a \preccurlyeq f(a)$ (and more generally, one can show that if $t \to_\beta u$ implies $t^{\mathcal{A}} \preccurlyeq u^{\mathcal{A}}$);
2. Adequate with respect to typing, in the sense that if t is of type A, then we have $t^{\mathcal{A}} \preccurlyeq A^{\mathcal{A}}$ (which can reads as *"t realizes A"*).

In the case of certain combinators, including Hilbert's combinator **k** and **s**, their interpretations as λ-term is even equal to the interpretation of their principal types, that is to say that we have $\mathbf{k}^{\mathcal{A}} = \bigcurlywedge_{a,b\in\mathcal{A}}(a \to b \to a)$ and $\mathbf{s}^{\mathcal{A}} = \bigcurlywedge_{a,b,c\in\mathcal{A}}((a \to b \to c) \to (a \to b) \to a \to c)$. This justifies the definition $\mathbf{cc}^{\mathcal{A}} \triangleq \bigcurlywedge_{a,b}(((a \to b) \to a) \to a)$.

Implicative structure are thus suited to interpret both terms and their types. To give an account for realizability models, one then has to define a notion of validity:

▶ **Definition 2** (Separator)**.** *Let* $(\mathcal{A}, \preccurlyeq, \to)$ *be an implicative structure. We call a* separator *over* $\mathcal{A}$ *any set* $\mathcal{S} \subseteq \mathcal{A}$ *such that for all* $a, b \in A$*, the following conditions hold:*
1. *If* $a \in \mathcal{S}$ *and* $a \preccurlyeq b$*, then* $b \in \mathcal{S}$.
2. $\mathbf{k}^{\mathcal{A}} \in \mathcal{S}$*, and* $\mathbf{s}^{\mathcal{A}} \in \mathcal{S}$.
3. *If* $(a \to b) \in \mathcal{S}$ *and* $a \in \mathcal{S}$*, then* $b \in \mathcal{S}$.

A separator $\mathcal{S}$ *is said to be* classical *if* $\mathbf{cc}^{\mathcal{A}} \in \mathcal{S}$ *and* consistent *if* $\bot \notin \mathcal{S}$*. We call* implicative algebra *any implicative structure* $(\mathcal{A}, \preccurlyeq, \to, \mathcal{S})$ *equipped with a separator* $\mathcal{S}$ *over* $\mathcal{A}$*.*

Intuitively, thinking of elements of an implicative structure as truth values, a separator should be understood as the set which distinguishes the valid formulas (think of a filter in a

---
[4] This is actually nothing more than the definition of the falsity value $\|A \Rightarrow B\|$.

Boolean algebra). Considering the elements as terms, it should rather be viewed as the set of valid realizers. Indeed, conditions (2) and (3) ensure that all closed λ-terms are in any separator[5]. Reading $a \preccurlyeq b$ as "*the formula* a *is a subtype of the formula* b", condition (2) ensures the validity of semantic subtyping. Thinking of the ordering as "a *is a realizer of the formula* b", condition (2) states that if a formula is realized, then it is in the separator.

▶ **Example 3.** *Any* Krivine realizability model *induces an implicative structure* $(\mathcal{A}, \preccurlyeq, \rightarrow)$ *where* $\mathcal{A} = \mathcal{P}(\Pi)$, $a \preccurlyeq b \Leftrightarrow a \supseteq b$ *and* $a \rightarrow b = a^{\perp\!\!\perp} \cdot b$. *The set of realized formulas, namely* $\mathcal{S} = \{a \in \mathcal{A} : \exists t \in a^{\perp\!\!\perp},\ t \text{ proof-like}\}$, *defines a valid separator [31].*

2.3 Internal logic & implicative tripos

In order to study the internal logic of implicative algebras, we define an *entailment* relation: we say that a entails b and we write $a \vdash_{\mathcal{S}} b$ if $a \rightarrow b \in \mathcal{S}$. This relation induces a preorder on $\mathcal{A}$. Then, by defining products $a \times b$ and sums $a + b$ through their usual impredicative encodings in System F[6], we recover a structure of pre-Heyting algebra with respect to the entailment relation: $a \vdash_{\mathcal{S}} b \rightarrow c$ if and only if $a \times b \vdash_{\mathcal{S}} c$.

In order to recover a Heyting algebra, it suffices to consider the quotient $\mathcal{H} = \mathcal{A}/_{\cong_{\mathcal{S}}}$ by the equivalence relation $\cong_{\mathcal{S}}$ induced by $\vdash_{\mathcal{S}}$, which is naturally equipped with an order relation: $[a] \preccurlyeq_{\mathcal{H}} [b] \triangleq a \vdash_{\mathcal{S}} b$ (where we write $[a]$ for the equivalence class of $a \in \mathcal{A}$). Likewise, we can extend the product, the sum and the arrow to equivalences classes to obtain a Heyting algebra $(\mathcal{H}, \preccurlyeq_{\mathcal{H}}, \wedge_{\mathcal{H}}, \vee_{\mathcal{H}}, \rightarrow_{\mathcal{H}})$.

Given any implicative algebra, we can define construction of the implicative tripos is quite similar. Recall that a (set-based) *tripos* is a first-order hyperdoctrine $\mathcal{T} : \mathbf{Set}^{op} \rightarrow \mathbf{HA}$ which admits a generic predicate. To define a tripos, we roughly consider the functor of the form $I \in \mathbf{Set}^{op} \mapsto \mathcal{A}^I$. Again, to recover a Heyting algebra we quotient the product $\mathcal{A}^I$ (which defines an implicative structure) by the *uniform separator* $\mathcal{S}[I]$ defined by:

$$\mathcal{S}[I] \triangleq \{a \in \mathcal{A}^I : \exists s \in \mathcal{S}. \forall i \in I. s \preccurlyeq a_i\}$$

▶ **Theorem 4** (Implicative tripos [31]). *Let* $(\mathcal{A}, \preccurlyeq, \rightarrow, \mathcal{S})$ *be an implicative algebra. The following functor (where* $f : J \rightarrow I$*) defines a tripos:*

$$\mathcal{T} : I \mapsto \mathcal{A}^I/\mathcal{S}[I] \qquad\qquad \mathcal{T}(f) : \begin{cases} \mathcal{A}^I/\mathcal{S}[I] & \rightarrow \quad \mathcal{A}^J/\mathcal{S}[J] \\ [(a_i)_{i\in I}] & \mapsto \quad [(a_{f(j)})_{j\in J}] \end{cases}$$

Observe that we could also quotient the product $\mathcal{A}^I$ by the separator product $\mathcal{S}^I$. Actually, the quotient $\mathcal{A}^I/\mathcal{S}^I$ is in bijection with $(\mathcal{A}/\mathcal{S})^{\mathcal{I}}$, and in the case where $\mathcal{S}$ is a classical separator, $\mathcal{A}/\mathcal{S}$ is actually a Boolean algebra, so that the product $(\mathcal{A}/\mathcal{S})^{\mathcal{I}}$ is nothing more than a Boolean-valued model (as in the case of forcing). Since $\mathcal{S}[I] \subseteq \mathcal{S}^I$, the realizability models that can not be obtained by forcing are exactly those for which $\mathcal{S}[I] \neq \mathcal{S}^I$ (see [31]).

3 Decomposing the arrow: disjunctive algebras

We shall now introduce the notion of disjunctive algebra, which is a structure primarily based on disjunctions, negations (for the connectives) and meets (for the universal quantifier). Our main purpose is to draw the comparison with implicative algebras, as an attempt to

[5] The latter indeed implies the closure of separators under application.
[6] That is to say that we define $a \times b \triangleq \bigcurlywedge_{c\in\mathcal{A}}((a \rightarrow b \rightarrow c) \rightarrow c)$ and $a + b \triangleq \bigcurlywedge_{c\in\mathcal{A}}((a \rightarrow c) \rightarrow (b \rightarrow c) \rightarrow c)$.

justify eventually that the latter are more general than the former, and to lay the bases for a dualizable definition. In the seminal paper introducing linear logic [12], Girard refines the structure of the sequent calculus LK, introducing in particular negative and positive connectives for disjunctions and conjunctions[7]. With this finer set of connectives, the usual implication can be retrieved using either the negative disjunction: $A \to B \triangleq \neg A \parr B$ or the positive conjunction: $A \to B \triangleq \neg(A \otimes \neg B)$.

In 2009, Munch-Maccagnoni gave a computational account of Girard's presentation for classical logic [35]. In his calculus, named L, each connective corresponds to the type of a particular constructor (or destructor). While L is in essence close to Curien and Herbelin's $\lambda\mu\tilde{\mu}$-calculus [4] (in particular it is presented with the same paradigm of duality between proofs and contexts), the syntax of terms does not include λ-abstraction (and neither does the syntax of formulas includes an implication). The two decompositions of the arrow evoked above are precisely reflected in decompositions of λ-abstractions (and dually, of stacks) in terms of L constructors. Notably, the choice of a decomposition corresponds to a particular choice of an evaluation strategy[8] for the encoded λ-calculus: picking the negative $\parr$ connective corresponds to call-by-name, while the decomposition using the $\otimes$ connective reduces in a call-by-value fashion.

We shall begin by considering the call-by-name case, which is closer to the situation of implicative algebras. The definition of disjunctive structures and algebras are guided by an analysis of the realizability model induced by $\mathrm{L}^{\parr}$, that is Munch-Maccagnoni's system L restricted to the fragment corresponding to negative formulas: $A, B := X \mid A \parr B \mid \neg A \mid \forall X.A$ [35]. To leave room for more details on disjunctive algebras, we elude here the introduction of $\mathrm{L}^{\parr}$ and its relation to the call-by-name λ-calculus, we refer the interested reader to the extended version.

3.1 Disjunctive structures

We are now going to define the notion of *disjunctive structure*. Since we choose negative connectives and in particular a universal quantifier, we should define commutations with respect to arbitrary meets. The realizability interpretation for $\mathrm{L}^{\parr}$ provides us with a safeguard in this regard, since in the corresponding models, if $X \notin FV(B)$ the following equalities[9] hold:

1. $\|\forall X.(A \parr B)\|_V = \|(\forall X.A) \parr B\|$.
2. $\|\forall X.(B \parr A)\|_V = \|B \parr (\forall X.A)\|$.
3. $\|\neg(\forall X.A)\|_V = \bigcap_{S\in\mathcal{P}(\mathcal{V}_0)} \|\neg A\{X := \dot{S}\}\|_V$

Algebraically, the previous proposition advocates for the following definition (remember that the order is defined as the reversed inclusion of primitive falsity values (whence $\cap$ is $\curlyvee$) and that the $\forall$ quantifier is interpreted by $\curlywedge$):

▶ **Definition 5** (Disjunctive structure). *A* disjunctive structure *is a complete lattice $(\mathcal{A}, \preccurlyeq)$ equipped with a binary operation $(a, b) \mapsto a \parr b$, together with a unary operation $a \mapsto \neg a$, such that for all $a, a', b, b' \in \mathcal{A}$ and for any $B \subseteq \mathcal{A}$:*

1. *if $a \preccurlyeq a'$ then $\neg a' \preccurlyeq \neg a$*
2. *if $a \preccurlyeq a'$ and $b \preccurlyeq b'$ then $a \parr b \preccurlyeq a' \parr b'$*
3. $\curlywedge_{b\in B}(b \parr a) = (\curlywedge_{b\in B} b) \parr a$
4. $\curlywedge_{b\in B}(a \parr b) = a \parr (\curlywedge_{b\in B} b)$
5. $\neg \curlywedge_{a\in A} a = \curlyvee_{a\in A} \neg a$

[7] We insist on the fact that even though we use linear notations afterwards, nothing will be linear here.
[8] Phrased differently, this observation can be traced back to different works, for instance by Blain-Levy [28, Fig. 5.10], Laurent [26] or Danos, Joinet and Schellinx [5].
[9] Technically, $\mathcal{V}_0$ is the set of closed values which, in this setting, are evaluation contexts (think of Π in usual Krivine models), and $\|A\|_V \in \mathcal{P}(\mathcal{V}_0)$ is the (ground) falsity value of a formula A.

Observe that the commutation laws imply the value of the internal laws when applied to the maximal element $\top$: **1.** $\top \parr a = \top$ **2.** $a \parr \top = \top$ **3.** $\neg\top = \bot$

We give here some examples of disjunctive structures.

▶ **Example 6** (Dummy disjunctive structure). *Given any complete lattice $(\mathcal{L}, \preccurlyeq)$, defining $a \parr b \triangleq \top$ and $\neg a \triangleq \bot$ gives rise to a dummy structure that fulfills the required properties.*

▶ **Example 7** (Complete Boolean algebras). *Let $\mathcal{B}$ be a complete Boolean algebra. It encompasses a disjunctive structure defined by:*

- $\mathcal{A} \triangleq \mathcal{B}$
- $a \preccurlyeq b \triangleq a \preccurlyeq b$
- $a \parr b \triangleq a \vee b$
- $\neg a \triangleq \neg a$

▶ **Example 8** ($\mathrm{L}^{\parr}$ realizability models). *Given a realizability interpretation of $L^{\parr}$, we define:*

- $\mathcal{A} \triangleq \mathcal{P}(\mathcal{V}_0)$
- $a \parr b \triangleq \{(V_1, V_2) : V_1 \in a \wedge V_2 \in b\}$
- $a \preccurlyeq b \triangleq a \supseteq b$
- $\neg a \triangleq [a^{\perp\!\!\!\perp}] = \{[t] : t \in a^{\perp\!\!\!\perp}\}$

where $\perp\!\!\!\perp$ is the pole, $\mathcal{V}_0$ is the set of closed values[9]*, and $(\cdot,\cdot)$ and $[\cdot]$ are the maps corresponding to $\parr$ and $\neg$. The resulting quadruple $(\mathcal{A}, \preccurlyeq, \parr, \neg)$ is a disjunctive structure.*

Following the interpretation of the λ-terms in implicative structures, we can embed $\mathrm{L}^{\parr}$ terms within disjunctive structures. We do not have the necessary space here to fully introduce here[10], but it is worth mentioning that the orthogonality relation $t \perp\!\!\!\perp e$ is interpreted via the ordering $t^{\mathcal{A}} \preccurlyeq e^{\mathcal{A}}$ (as suggested in [8, Theorem 5.13] by the definition of an abstract Krivine structure and its pole from an ordered combinatory algebra).

3.2 The induced implicative structure

As expected, any disjunctive structure directly induces an implicative structure through the definition $a \overset{\parr}{\rightarrow} b \triangleq \neg a \parr b$:

▶ **Proposition 9.** *If $(\mathcal{A}, \preccurlyeq, \parr, \neg)$ is a disjunctive structure, then $(\mathcal{A}, \preccurlyeq, \overset{\parr}{\rightarrow})$ is an implicative structure.*

Therefore, we can again define for all a, b of $\mathcal{A}$ the application ab as well as the abstraction λf for any function f from $\mathcal{A}$ to $\mathcal{A}$; and we get for free the properties of these encodings in implicative structures.

Up to this point, we have two ways of interpreting a λ-term into a disjunctive structure: either through the implicative structure which is induced by the disjunctive one, or by embedding into the $\mathrm{L}^{\parr}$-calculus which is then interpreted within the disjunctive structure. As a sanity check, we verify that both coincide:

▶ **Proposition 10** (λ-calculus). *Let $\mathcal{A}^{\parr} = (\mathcal{A}, \preccurlyeq, \parr, \neg)$ be a disjunctive structure, and $\mathcal{A}^{\rightarrow} = (\mathcal{A}, \preccurlyeq, \overset{\parr}{\rightarrow})$ the implicative structure it canonically defines, we write ι for the corresponding inclusion. Let t be a closed λ-term (with parameter in $\mathcal{A}$), and $[\![t]\!]$ his embedding in $L^{\parr}$. Then we have $\iota(t^{\mathcal{A}^{\rightarrow}}) = [\![t]\!]^{\mathcal{A}^{\parr}}$.*

[10] See the extended version for more details.

3.3 Disjunctive algebras

We shall now introduce the notion of disjunctive separator. To this purpose, we adapt the definition of implicative separators, using standard axioms[11] for the disjunction and the negation instead of Hilbert's combinators **s** and **k**. We thus consider the following combinators:

$$\begin{array}{lcl} \boldsymbol{s}_1^{\parr} & \triangleq & \curlywedge_{a\in\mathcal{A}}[(a \parr a) \to a] \\ \boldsymbol{s}_2^{\parr} & \triangleq & \curlywedge_{a,b\in\mathcal{A}}[a \to (a \parr b)] \\ \boldsymbol{s}_3^{\parr} & \triangleq & \curlywedge_{a,b\in\mathcal{A}}[(a \parr b) \to b \parr a] \end{array} \quad\Bigg|\quad \begin{array}{lcl} \boldsymbol{s}_4^{\parr} & \triangleq & \curlywedge_{a,b,c\in\mathcal{A}}[(a \to b) \to (c \parr a) \to (c \parr b)] \\ \boldsymbol{s}_5^{\parr} & \triangleq & \curlywedge_{a,b,c\in\mathcal{A}}[(a \parr (b \parr c)) \to ((a \parr b) \parr c)] \end{array}$$

Separators for $\mathcal{A}$ are defined similarly to the separators for implicative structures, replacing the combinators $\mathbf{k}, \mathbf{s}$ and $\mathbf{cc}$ by the previous ones.

▶ **Definition 11** (Separator). *We call* separator *for the disjunctive structure $\mathcal{A}$ any subset $\mathcal{S} \subseteq \mathcal{A}$ that fulfills the following conditions for all $a, b \in \mathcal{A}$:*

1. *If $a \in \mathcal{S}$ and $a \preccurlyeq b$ then $b \in \mathcal{S}$.*
2. *$\boldsymbol{s}_1^{\parr}, \boldsymbol{s}_2^{\parr}, \boldsymbol{s}_3^{\parr}, \boldsymbol{s}_4^{\parr}$ and $\boldsymbol{s}_5^{\parr}$ are in $\mathcal{S}$.*
3. *If $a \to b \in \mathcal{S}$ and $a \in \mathcal{S}$ then $b \in \mathcal{S}$.*

A separator $\mathcal{S}$ is said to be consistent *if $\bot \notin \mathcal{S}$. We call* disjunctive algebra *the given of a disjunctive structure together with a separator $\mathcal{S} \subseteq \mathcal{A}$.*

▶ **Remark 12**. *The reader may notice that in this section, we do not distinguish between classical and intuitionistic separators. Indeed, $L^{\parr}$ and the corresponding fragment of the sequent calculus are intrinsically classical. As we shall see thereafter, so are the disjunctive algebras: the negation is always involutive modulo the equivalence $\cong_{\mathcal{S}}$ (Proposition 16).*

▶ **Remark 13** (Generalized modus ponens). *The modus ponens, that is the unique deduction rule we have, is actually compatible with meets. Consider a set I and two families $(a_i)_{i\in I}, (b_i)_{i\in I} \in \mathcal{A}^I$, we have:*

$$\frac{a \vdash_I b \quad \vdash_I a}{\vdash_I b}$$

where we write $a \vdash_I b$ for $(\curlywedge_{i\in I} a_i \to b_i) \in \mathcal{S}$ and $\vdash_I a$ for $(\curlywedge_{i\in I} a_i) \in \mathcal{S}$. As our axioms are themselves expressed as meets, the results that we will obtain internally (that is by deduction from the separator's axioms) can all be generalized to meets.

▶ **Example 14** (Complete Boolean algebras). *Once again, if $\mathcal{B}$ is a complete Boolean algebra, $\mathcal{B}$ induces a disjunctive structure in which it is easy to verify that the combinators $\boldsymbol{s}_1^{\parr}, \boldsymbol{s}_3^{\parr}, \boldsymbol{s}_3^{\parr}, \boldsymbol{s}_4^{\parr}$ and $\boldsymbol{s}_5^{\parr}$ are equal to the maximal element $\top$. Therefore, the singleton $\{\top\}$ is a valid separator for the induced disjunctive structure. In fact, the filters for $\mathcal{B}$ are exactly its separators.*

▶ **Example 15** ($L^{\parr}$ realizability model). *Remember from Example 8 that any model of classical realizability based on the $L^{\parr}$-calculus induces a disjunctive structure. As in the implicative case, the set of formulas realized by a closed term[12] defines a valid separator.*

---

[11] These axioms can be found for instance in Whitehead and Russell's presentation of logic [41]. In fact, the fifth axiom is deducible from the first four as was later shown by Bernays [2]. For simplicity reasons, we preferred to keep it as an axiom.

[12] Proof-like terms in $L^{\parr}$ simply correspond to closed terms.

3.4 Internal logic

As in the case of implicative algebras, we say that a *entails* b and write $a \vdash_{\mathcal{S}} b$ if $a \to b \in \mathcal{S}$. Through this relation, which is again a preorder relation, we can relate the primitive negation and disjunction to the negation and sum type induced by the underlying implicative structure:

$$a + b \triangleq \bigwedge_{c \in \mathcal{A}} ((a \to c) \to (b \to c) \to c) \qquad (\forall a, b \in \mathcal{A})$$

In particular, we show that from the point of view of the separator the principle of double negation elimination is valid and the disjunction and this sum type are equivalent:

▶ **Proposition 16** (Implicative connectives). *For all $a, b \in \mathcal{A}$, the following holds:*

1. $\neg a \vdash_{\mathcal{S}} a \to \bot$
2. $a \to \bot \vdash_{\mathcal{S}} \neg a$
3. $a \vdash_{\mathcal{S}} \neg\neg a$
4. $\neg\neg a \vdash_{\mathcal{S}} a$
5. $a \parr b \vdash_{\mathcal{S}} a + b$
6. $a + b \vdash_{\mathcal{S}} a \parr b$

3.5 Induced implicative algebras

In order to show that any disjunctive algebra is a particular case of implicative algebra, we first verify that Hilbert's combinators belong to any disjunctive separator:

▶ **Proposition 17** (Combinators). *We have:* **1.** $\mathbf{k}^{\mathcal{A}} \in \mathcal{S}$ **2.** $\mathbf{s}^{\mathcal{A}} \in \mathcal{S}$ **3.** $\mathbf{cc}^{\mathcal{A}} \in \mathcal{S}$

As a consequence, we get the expected theorem:

▶ **Theorem 18.** *Any disjunctive algebra is a classical implicative algebra.*

Since any disjunctive algebra is actually a particular case of implicative algebra, the construction leading to the implicative tripos can be rephrased entirely in this framework. In particular, the same criteria allows us to determine whether the implicative tripos is isomorphic to a forcing tripos. Notably, a disjunctive algebra admitting an extra-commutation rule the negation $\neg$ with arbitrary joins ($\neg \bigvee_{a \in A} a = \bigwedge_{a \in A} \neg a$) will induce an implicative algebra where the arrow commutes with arbitrary joins. In that case, the induced tripos would collapse to a forcing situation (see [31]).

4 A positive decomposition: conjunctive algebras

4.1 Call-by-value realizability models

While there exists now several models build of classical theories constructed via Krivine realizability [22, 24, 25, 29], they all have in common that they rely on a presentation of logic based on negative connectives/quantifiers. If this might not seem shocking from a mathematical perspective, it has the computational counterpart that these models all build on a call-by-name calculus, namely the λ_c-calculus[13]. In light of the logical consequences that computational choices have on the induced theory, it is natural to wonder whether the choice of a call-by-name evaluation strategy is anecdotal or fundamental.

As a first step in this direction, we analyze here the algebraic structure of realizability models based on the $\mathrm{L}^{\otimes}$ calculus, the positive fragment of Munch-Maccagnoni's system L

[13] Actually, there is two occurrences of realizability interpretations for call-by-value calculus, including Munch-Maccagnoni's system L, but both are focused on the analysis of the computational behavior of programs rather than constructing models of a given logic [35, 27].

corresponding to the formulas defined by: $A, B ::= X \mid \neg A \mid A \otimes B \mid \exists X.A$. Through the well-known duality between terms and evaluation contexts [4, 35], this fragment is dual to the $\mathrm{L}^{\invamp}$ calculus and it naturally allows to embed the λ-terms evaluated in a call-by-value fashion. We shall now reproduce the approach we had for $\mathrm{L}^{\invamp}$: guided by the analysis of the realizability models induced by the $\mathrm{L}^{\otimes}$ calculus, we first define *conjunctive structures*. We then show how these structures can be equipped with a separator and how the resulting *conjunctive algebras* lead to the construction of a *conjunctive tripos*. We will finally show in the next section how conjunctive and disjunctive algebras are related by an algebraic duality.

4.2 Conjunctive structures

As in the previous section, we will not introduce here the $\mathrm{L}^{\otimes}$ calculus and the corresponding realizability models (see the extended version for details). Their main characteristic is that, being build on top of a call-by-value calculus, a formula A is primitively interpreted by its *ground truth value* $|A|_v \in \mathcal{P}(\mathcal{V}_O)$ which is a set of values. Its falsity and truth values are then defined by orthogonality [35, 27]. Once again, we can observe the existing commutations in these realizability models. Insofar as we are in a structure centered on positive connectives, we especially pay attention to the commutations with joins. As a matter of fact, in any $\mathrm{L}^{\otimes}$ realizability model, we have that if $X \notin FV(B)$:

1. $|\exists X.(A \otimes B)|_V = |(\exists X.A) \otimes B|_V$.
2. $|\exists X.(B \otimes A)|_V = |B \otimes (\exists X.A)|_V$.
3. $|\neg(\exists X.A)|_V = \bigcap_{S \in \mathcal{P}(\mathcal{V}_0)} |\neg A\{X := \dot{S}\}|_V$

Since we are now interested in primitive truth values, which are logically ordered by inclusion (in particular, the existential quantifier is interpreted by unions, thus joins), the previous proposition advocates for the following definition:

▶ **Definition 19** (Conjunctive structure). *A* conjunctive structure *is a complete join-semilattice $(\mathcal{A}, \preccurlyeq)$ equipped with a binary operation $(a, b) \mapsto a \otimes b$, and a unary operation $a \mapsto \neg a$, such that for all $a, a', b, b' \in \mathcal{A}$ and for all subset $B \subseteq \mathcal{A}$ we have:*

1. *if $a \preccurlyeq a'$ then $\neg a' \preccurlyeq \neg a$*
2. *if $a \preccurlyeq a'$ and $b \preccurlyeq b'$ then $a \otimes b \preccurlyeq a' \otimes b'$*
3. $\curlyvee_{b \in B}(a \otimes b) = a \otimes (\curlyvee_{b \in B} b)$
4. $\curlyvee_{b \in B}(b \otimes a) = (\curlyvee_{b \in B} b) \otimes a$
5. $\neg \curlyvee_{a \in A} a = \curlywedge_{a \in A} \neg a$

As in the cases of implicative and disjunctive structures, the commutation rules imply that: **1.** $\bot \otimes a = \bot$ **2.** $a \otimes \bot = \bot$ **3.** $\neg \bot = \top$

▶ **Example 20** (Dummy conjunctive structure). *Given a complete lattice L, the following definitions give rise to a dummy conjunctive structure:* $a \otimes b \triangleq \bot$ $\neg a \triangleq \top$.

▶ **Example 21** (Complete Boolean algebras). *Let $\mathcal{B}$ be a complete Boolean algebra. It embodies a conjunctive structure, that is defined by:*

- $\mathcal{A} \triangleq \mathcal{B}$
- $a \preccurlyeq b \triangleq a \preccurlyeq b$
- $a \otimes b \triangleq a \wedge b$
- $\neg a \triangleq \neg a$

▶ **Example 22** ($\mathrm{L}^{\otimes}$ realizability models). *As for the disjunctive case, we can abstract the structure of the realizability interpretation of $L^{\otimes}$ to define:*

- $\mathcal{A} \triangleq \mathcal{P}(\mathcal{V}_0)$
- $a \preccurlyeq b \triangleq a \subseteq b$
- $a \otimes b \triangleq \{(V_1, V_2) : V_1 \in a \wedge V_2 \in b\}$
- $\neg a \triangleq [a^{\perp\!\!\perp}] = \{[e] : e \in a^{\perp\!\!\perp}\}$

where $\perp\!\!\perp$ is the pole, $\mathcal{V}_0$ is the set of closed values and $(\cdot,\cdot)$ and $[\cdot]$ are the maps corresponding to $\otimes$ and $\neg$. The resulting quadruple $(\mathcal{A}, \preccurlyeq, \otimes, \neg)$ is a conjunctive structure.

It is worth noting that even though we can define an arrow by $a \overset{\otimes}{\rightarrow} b \triangleq \neg(a \otimes \neg b)$, it does not induce an implicative structure: indeed, the distributivity law is not true in general[14]. In turns, we have another distributivity law which is usually wrong in implicative structure:

$$(\bigcurlyvee_{a\in A} a) \overset{\otimes}{\rightarrow} b = \bigcurlywedge_{a\in A} (a \overset{\otimes}{\rightarrow} b) \qquad\qquad \bigcurlywedge_{b\in B} (a \overset{\otimes}{\rightarrow} b) \not\preccurlyeq a \overset{\otimes}{\rightarrow} (\bigcurlywedge_{b\in B} b)$$

Actually, implicative structures where both are true corresponds precisely to a degenerated forcing situation.

Here again, we can define an embedding of $L^{\otimes}$ into any conjunctive structure which is sound with respect to typing and reductions[15].

4.3 Conjunctive algebras

The definition of conjunctive separators turns out to be more subtle than in the disjunctive case. Among others things, conjunctive structures mainly axiomatize joins, while the combinators or usual mathematical axioms that we could wish to have in a separator are more naturally expressed via universal quantifications, hence meets. Yet, an analysis of the sequent calculus underlying $L^{\otimes}$ type system[15], shows that we could consider a tensorial calculus where deduction systematically involves a conclusion of the shape $\neg A$. This justifies to consider the following combinators[16]:

$$\begin{array}{l|l}
\boldsymbol{s}_1^{\otimes} \triangleq \curlywedge_{a\in\mathcal{A}} \neg[\neg(a\otimes a)\otimes a] & \\
\boldsymbol{s}_2^{\otimes} \triangleq \curlywedge_{a,b\in\mathcal{A}} \neg[\neg a\otimes(a\otimes b)] & \boldsymbol{s}_4^{\otimes} \triangleq \curlywedge_{a,b,c\in\mathcal{A}} \neg[\neg(\neg a\otimes b)\otimes(\neg(c\otimes a)\otimes(c\otimes b))] \\
\boldsymbol{s}_3^{\otimes} \triangleq \curlywedge_{a,b\in\mathcal{A}} \neg[\neg(a\otimes b)\otimes(b\otimes a)] & \boldsymbol{s}_5^{\otimes} \triangleq \curlywedge_{a,b,c\in\mathcal{A}} \neg[\neg(a\otimes(b\otimes c))\otimes((a\otimes b)\otimes c)]
\end{array}$$

and to define conjunctive separators as follows:

▶ **Definition 23** (Separator). *We call* separator *for the disjunctive structure $\mathcal{A}$ any subset $\mathcal{S} \subseteq \mathcal{A}$ that fulfills the following conditions for all $a, b \in \mathcal{A}$:*

1. *If $a \in \mathcal{S}$ and $a \preccurlyeq b$ then $b \in \mathcal{S}$.*
2. *$\boldsymbol{s}_1^{\otimes}, \boldsymbol{s}_2^{\otimes}, \boldsymbol{s}_3^{\otimes}, \boldsymbol{s}_4^{\otimes}$ and $\boldsymbol{s}_5^{\otimes}$ are in $\mathcal{S}$.*
3. *If $\neg(a \otimes b) \in \mathcal{S}$ and $a \in \mathcal{S}$ then $\neg b \in \mathcal{S}$.*
4. *If $a \in \mathcal{S}$ and $b \in \mathcal{S}$ then $a \otimes b \in \mathcal{S}$.*

A separator $\mathcal{S}$ is said to be classical *if besides $\neg\neg a \in \mathcal{S}$ implies $a \in \mathcal{S}$.*

▶ **Remark 24** (Modus Ponens). *If the separator is classical, it is easy to see that the modus ponens is valid: if $a \overset{\otimes}{\rightarrow} b \in \mathcal{S}$ and $a \in \mathcal{S}$, then $\neg\neg b \in \mathcal{S}$ by (3) and thus $b \in \mathcal{S}$.*

▶ **Example 25** (Complete Boolean algebras). *Once again, if $\mathcal{B}$ is a complete Boolean algebra, $\mathcal{B}$ induces a conjunctive structure in which it is easy to verify that the combinators $\boldsymbol{s}_1^{⅋}, \boldsymbol{s}_3^{⅋}, \boldsymbol{s}_3^{⅋}, \boldsymbol{s}_4^{⅋}$ and $\boldsymbol{s}_5^{⅋}$ are equal to the maximal element $\top$. Therefore, the singleton $\{\top\}$ is a valid separator.*

▶ **Example 26** ($L^{\otimes}$realizability model). *As expected, the set of realized formulas by a proof-like term: defines a valid separator for the conjunctive structures induced by $L^{\otimes}$ realizability models.*

[14] For instance, it is false in $L^{\otimes}$ realizability models.
[15] See the extended version for more details.
[16] Observe that are directly dual to the combinators for disjunctive separators and that they can be alternatively given the shape $\neg \bigcurlyvee_{\_\in\mathcal{A}} \cdots$.

▶ **Example 27** (Kleene realizability). *We do not want to enter into too much details here, but it is worth mentioning that realizability interpretations* à la *Kleene of intuitionistic calculi equipped with primitive pairs (e.g. (partial) combinatory algebras, the λ-calculus) induce conjunctive algebras. Insofar as many Kleene realizability models takes position against classical reasoning (for $\forall X.X \vee \neg X$ is not realized and hence its negation is), these algebras have the interesting properties of not being classical (and are even incompatible with a classical completion).*

▶ **Remark 28** (Generalized axioms). *Once again, the axioms (3) and (4) generalize to meet of families $(a_i)_{i\in I}, (b_i)_{i\in I}$:*

$$\frac{\vdash_I \neg(a \otimes b) \quad \vdash_I a}{\vdash_I \neg b} \qquad\qquad \frac{\vdash_I a \quad \vdash_I b}{\vdash_I a \otimes b}$$

where we write $\vdash_I a$ for $(\curlywedge_{i\in I} a_i) \in \mathcal{S}$ and where the negation and conjunction of families are taken pointwise. Once again, the axioms being themselves expressed as meets, this means that any result obtained from the separator's axioms (but the classical one) can be generalized to meets.

4.4 Internal logic

As before, we consider the entailment relation defined by $a \vdash_{\mathcal{S}} b \triangleq (a \overset{\otimes}{\to} b) \in \mathcal{S}$. Observe that if the separator is not classical, we do not have that $a \vdash_{\mathcal{S}} b$ and $a \in \mathcal{S}$ entails[17] $b \in \mathcal{S}$. Nonetheless, this relation still defines a preorder in the sense that:

▶ **Proposition 29** (Preorder). *For any $a, b, c \in \mathcal{A}$, we have:*

1. $a \vdash_{\mathcal{S}} a$
2. *If $a \vdash_{\mathcal{S}} b$ and $b \vdash_{\mathcal{S}} c$ then $a \vdash_{\mathcal{S}} c$*

Intuitively, this reflects the fact that despite we may not be able to extract the value of a computation, we can always chain it with another computation expecting a value.

Here again, we can relate the negation $\neg a$ to the one induced by the arrow $a \overset{\otimes}{\to} \bot$:

▶ **Proposition 30** (Implicative negation). *For all $a \in \mathcal{A}$, the following holds:*

1. $\neg a \vdash_{\mathcal{S}} a \overset{\otimes}{\to} \bot$
2. $a \overset{\otimes}{\to} \bot \vdash_{\mathcal{S}} \neg a$
3. $a \vdash_{\mathcal{S}} \neg\neg a$
4. $\neg\neg a \vdash_{\mathcal{S}} a$

As in implicative structures, we can define the abstraction and application of the λ-calculus:

$$\lambda f \triangleq \bigcurlywedge_{a\in\mathcal{A}} (a \overset{\otimes}{\to} f(a)) \qquad\qquad ab \triangleq \bigcurlywedge\{\neg\neg c : a \preccurlyeq b \overset{\otimes}{\to} c\}$$

Observe that here we need to add a double negation, since intuitively ab is a *computation* of type $\neg\neg c$ rather than a value of type c. In other words, values are not stable by applications, and extracting a value from a computation requires a form of classical control. Nevertheless, for any separator we have:

▶ **Proposition 31**. *If $a \in \mathcal{S}$ and $b \in \mathcal{S}$ then $ab \in \mathcal{S}$.*

Similarly, the beta reduction rule now involves a double-negation on the reduced term:

[17] Actually we can consider a different relation $a \vdash^{\neg} b \triangleq \neg(a \otimes b)$ for which $a \vdash^{\neg} b$ and $a \in \mathcal{S}$ entails $\neg b$. This one turns out to be useful to ease proofs, but from a logical perspective, the significant entailment is the one given by $a \vdash_{\mathcal{S}} b$.

▶ **Proposition 32.** $(\lambda f)a \preccurlyeq \neg\neg f(a)$

We show that Hilbert's combinators **k** and **s** belong to any conjunctive separator:

▶ **Proposition 33** (**k** and **s**)**.** *We have:*

1. $(\lambda xy.x)^{\mathcal{A}} \in \mathcal{S}$
2. $(\lambda xyz.x\,z\,(y\,z))^{\mathcal{A}} \in \mathcal{S}$

By combinatorial completeness, for any closed λ-term t we thus have the a combinatorial term t_0 (i.e. a composition of **k** and **s**) such that $t_0 \to^* t$. Since $\mathcal{S}$ is closed under application, $t_0^{\mathcal{A}}$ also belong to $\mathcal{S}$. Besides, since for each reduction step $t_n \to t_{n+1}$, we have $t_n^{\mathcal{A}} \preccurlyeq \neg\neg t_{n+1}^{\mathcal{A}}$, if the separator is classical[18], we can thus deduce that it contains the interpretation of t :

▶ **Theorem 34** (λ-calculus)**.** *If $\mathcal{S}$ is classical and t is a closed λ-term, then $t^{\mathcal{A}} \in \mathcal{S}$.*

Once more, the entailment relation induces a structure of (pre)-Heyting algebra, whose conjunction and disjunction are naturally given by $a \times b \triangleq a \otimes b$ and $a + b \triangleq \neg(\neg a \otimes \neg b)$:

▶ **Proposition 35** (Heyting Algebra)**.** *For any $a, b, c \in \mathcal{A}$ For any $a, b, c \in \mathcal{A}$, we have:*

1. $a \times b \vdash_{\mathcal{S}} a$
2. $a \times b \vdash_{\mathcal{S}} b$
3. $a \vdash_{\mathcal{S}} a + b$
4. $b \vdash_{\mathcal{S}} a + b$
5. $a \vdash_{\mathcal{S}} b \overset{\otimes}{\to} c$ *iff* $a \times b \vdash_{\mathcal{S}} c$

We can thus quotient the algebra by the equivalence relation $\cong_{\mathcal{S}}$ and extend the previous operation to equivalence classes in order to obtain a Heyting algebra $\mathcal{A}/\cong_{\mathcal{S}}$. In particular, this allows us to obtain a tripos out of a conjunctive algebra by reproducing the construction of the implicative tripos in our setting:

▶ **Theorem 36** (Conjunctive tripos)**.** *Let $(\mathcal{A}, \preccurlyeq, \to, \mathcal{S})$ be a classical*[19] *conjunctive algebra. The following functor (where $f : J \to I$) defines a tripos:*

$$\mathcal{T} : I \mapsto \mathcal{A}^I/\mathcal{S}[I] \qquad\qquad \mathcal{T}(f) : \begin{cases} \mathcal{A}^I/\mathcal{S}[I] & \to & \mathcal{A}^J/\mathcal{S}[J] \\ [(a_i)_{i\in I}] & \mapsto & [(a_{f(j)})_{j\in J}] \end{cases}$$

5 The duality of computation, algebraically

In [4], Curien and Herbelin introduce the $\lambda\mu\tilde{\mu}$ in order to emphasize the so-called duality of computation between terms and evaluation contexts. They define a simple translation inverting the role of terms and stacks within the calculus, which has the notable consequence of translating a call-by-value calculus into a call-by-name calculus and vice-versa. The very same translation can be expressed within L, in particular it corresponds to the trivial translation from mapping every constructor on terms (resp. destructors) in $\mathrm{L}^{\otimes}$ to the corresponding constructor on stacks (resp. destructors) in $\mathrm{L}^{⅋}$. We shall now see how this fundamental duality of computation can be retrieved algebraically between disjunctive and conjunctive algebras.

We first show that we can simply pass from one structure to another by reversing the order relation. We know that reversing the order in a complete lattice yields a complete

[18] Actually, since we always have that if $\neg\neg\neg\neg a \in \mathcal{S}$ then $\neg\neg a \in \mathcal{S}$, the same proof shows that in the intuitionistic case we have at $\neg\neg t^{\mathcal{A}} \in \mathcal{S}$.

[19] For technical reasons, we only give the proof in case where the separator is classical (recall that it allows to directly use λ-terms), but as explained, by adding double negation everywhere the same reasoning should work for the general case as well. Yet, this is enough to express our main result in the next section which only deals with the classical case.

lattice in which meets and joins are exchanged. Therefore, it only remains to verify that the axioms of disjunctive and conjunctive structures can be deduced through this duality one from each other, which is the case.

▶ **Proposition 37.** *Let* $(\mathcal{A}, \preccurlyeq, \parr, \neg)$ *be a disjunctive structure. Let us define:*

- $\mathcal{A}^{\otimes} \triangleq \mathcal{A}^{\parr}$
- $a \triangleleft b \triangleq b \preccurlyeq a$
- $a \otimes b \triangleq a \parr b$
- $\neg a \triangleq \neg a$

then $(\mathcal{A}^{\otimes}, \triangleleft, \otimes, \neg)$ *is a conjunctive structure.*

▶ **Proposition 38.** *Let* $(\mathcal{A}, \preccurlyeq, \otimes, \neg)$ *be a conjunctive structure. Let us define:*

- $\mathcal{A}^{\parr} \triangleq \mathcal{A}^{\otimes}$
- $a \triangleleft b \triangleq b \preccurlyeq a$
- $a \parr b \triangleq a \otimes b$
- $\neg a \triangleq \neg a$

then $(\mathcal{A}^{\otimes}, \triangleleft, \otimes, \neg)$ *is a disjunctive structure.*

Intuitively, by considering stacks as realizers, we somehow reverse the algebraic structure, and we consider as valid formulas the ones whose orthogonals were valid. In terms of separator, it means that when reversing a structure we should consider the separator defined as the preimage through the negation of the original separator.

▶ **Theorem 39.** *Let* $(\mathcal{A}^{\otimes}, \mathcal{S}^{\otimes})$ *be a conjunctive algebra, the set* $\mathcal{S}^{\parr} \triangleq \{a \in \mathcal{A} : \neg a \in \mathcal{S}^{\otimes}\}$ *defines a valid separator for the dual disjunctive structure* $\mathcal{A}^{\parr}$*.*

▶ **Theorem 40.** *Let* $(\mathcal{A}^{\parr}, \mathcal{S}^{\parr})$ *be a disjunctive algebra. The set* $\mathcal{S}^{\otimes} \triangleq \{a \in \mathcal{A} : \neg a \in \mathcal{S}^{\parr}\}$ *defines a classical separator for the dual conjunctive structure* $\mathcal{A}^{\otimes}$*.*

It is worth noting that reversing in both cases, the dual separator is classical. This is to connect with the fact that classical reasoning principles are true on negated formulas. Moreover, starting from a non-classical conjunctive algebra, one can reverse it twice to get a classical algebra. This corresponds to a classical completion of the original separator $\mathcal{S}$: it is easy to see that $a \in \mathcal{S}$ implies $\neg\neg a \in \mathcal{S}$, hence $\mathcal{S} \subseteq \{a : \neg\neg a \in \mathcal{S}\}$.

Actually, the duality between disjunctive and (classical) conjunctive algebras is even stronger, in the sense that through the translation, the induced triposes are isomorphic. Remember that an isomorphism φ between two (**Set**-based) triposes $\mathcal{T}, \mathcal{T}'$ is defined as a natural isomorphism $\mathcal{T} \Rightarrow \mathcal{T}'$ in the category **HA**, that is as a family of isomorphisms $\varphi_I : \mathcal{T}(I) \xrightarrow{\sim} \mathcal{T}'(I)$ (indexed by all $I \in \textbf{Set}$) that is natural in $\mathcal{I}$.

▶ **Theorem 41** (Main result)**.** *Let* $(\mathcal{A}, \mathcal{S})$ *be a disjunctive algebra and* $(\bar{\mathcal{A}}, \bar{\mathcal{S}})$ *its dual conjunctive algebra. The following family of maps defines a tripos isomorphism:*

$$\varphi_I : \left\{ \begin{array}{ccc} \bar{\mathcal{A}}/\bar{\mathcal{S}}[I] & \to & \mathcal{A}/\mathcal{S}[I] \\ [a_i] & \mapsto & [\neg a_i] \end{array} \right.$$

6 Conclusion

6.1 An algebraic view on the duality of computation

To sum up, in this paper we saw how the two decompositions of the arrow $a \to b$ as $\neg a \parr b$ and $\neg(a \otimes \neg b)$, which respectively induce decompositions of a call-by-name and call–by-value λ-calculi within Munch-Maccagnoni's system L [35], yield two different algebraic structures reflecting the corresponding realizability models. Namely, call-by-name models give rise to disjunctive algebras, which are particular cases of Miquel's implicative algebras [31]; while conjunctive algebras correspond to call-by-value realizability models.

The well-known duality of computation between terms and contexts is reflected here by simple translations from conjunctive to disjunctive algebras and vice-versa, where the underlying lattices are simply reversed. Besides, we showed that (classical) conjunctive algebras induce triposes that are isomorphic to disjunctive triposes. The situation is summarized in Figure 1, where $\otimes^{\neg\neg}$ denotes classical conjunctive algebras.

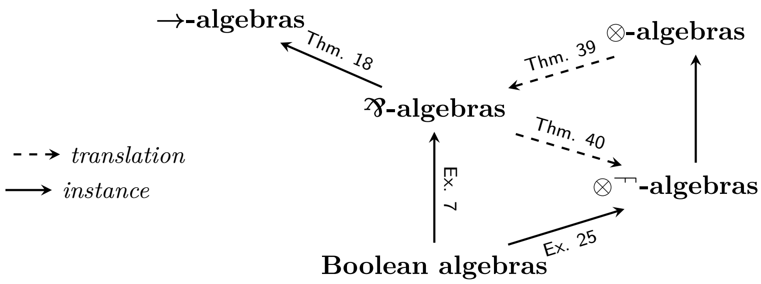

Figure 1 Final picture.

6.2 From Kleene to Krivine via negative translation

We could now re-read within our algebraic landscape the result of Oliva and Streicher stating that Krivine realizability models for PA2 can be obtained as a composition of Kleene realizability for HA2 and Friedman's negative translation [36, 30]. Interestingly, in this setting the fragment of formulas that is interpreted in HA2 correspond exactly to the positive formulas of $L^{\otimes}$, so that it gives rise to an (intuitionistic) conjunctive algebra. Friedman's translation is then used to encode the type of stacks within this fragment via a negation. In the end, realized formulas are precisely the ones that are realized through Friedman's translation: the whole construction exactly matches the passage from a intuitionistic conjunctive structure defined by Kleene realizability to a classical implicative algebras through the arrow from $\otimes$-algebras to $\rightarrow$-algebras via $⅋$-algebras.

6.3 Future work

While Theorem 41 implies that call-by-value and call-by-name models based on the $L^{\otimes}$ and $L^{⅋}$ calculi are equivalents, it does not provide us with a definitive answer to our original question. Indeed, just as (by-name) implicative algebras are more general than disjunctive algebras, it could be the case that there exists a notion of (by-value) implicative algebras that is strictly more general than conjunctive algebras and which is not isomorphic to a by-name situation.

Also, if we managed to obtain various results about conjunctive algebras, there is still a lot to understand about them. Notably, the interpretation we have of the λ-calculus is a bit disappointing in that it does not provide us with an adequacy result as nice as in implicative algebras. In particular, the fact that each application implicitly gives rise to a double negation breaks the compositionality. This is of course to connect with the definition of *truth values* in by-value models which requires three layers and a double orthogonal. We thus feel that many things remain to understand about the underlying structure of by-value realizability models.

Finally, on a long-term perspective, the next step would be to understand the algebraic impact of more sophisticated evaluation strategy (e.g., call-by-need) or side effects (e.g., a monotonic memory). While both have been used in concrete cases to give a computational content to certain axioms (e.g., the axiom of dependent choice [15]) or model constructions (e.g., forcing [21]), for the time being we have no idea on how to interpret them in the realm of implicative algebras.

References

1 John L. Bell. *Set Theory: Boolean-Valued Models and Independence Proofs*. Oxford: Clarendon Press, 2005.

2 P. Bernays. Axiomatische Untersuchung des Aussagen-Kalküls der "Principia Mathematica". *Mathematische Zeitschrift*, 25:305–320, 1926.

3 Luca Cardelli, Simone Martini, John C. Mitchell, and Andre Scedrov. *An extension of system F with subtyping*, pages 750–770. Springer Berlin Heidelberg, Berlin, Heidelberg, 1991. doi:10.1007/3-540-54415-1_73.

4 Pierre-Louis Curien and Hugo Herbelin. The duality of computation. In *Proceedings of ICFP 2000*, SIGPLAN Notices 35(9), pages 233–243. ACM, 2000. doi:10.1145/351240.351262.

5 Vincent Danos, Jean-Baptiste Joinet, and Harold Schellinx. LKQ and LKT : Sequent calculi for second order logic based upon dual linear decompositions of classical implication. In *Advances in Linear Logic*, 1995.

6 W. Ferrer and O. Malherbe. The category of implicative algebras and realizability. *ArXiv e-prints*, December 2017. arXiv:1712.06043.

7 W. Ferrer Santos, M. Guillermo, and O. Malherbe. A Report on Realizability. *ArXiv e-prints*, 2013. arXiv:1309.0706.

8 W. Ferrer Santos, M. Guillermo, and O. Malherbe. Realizability in OCAs and AKSs. *ArXiv e-prints*, 2015. arXiv:1512.07879.

9 Walter Ferrer Santos, Jonas Frey, Mauricio Guillermo, Octavio Malherbe, and Alexandre Miquel. Ordered combinatory algebras and realizability. *Mathematical Structures in Computer Science*, 27(3):428–458, 2017. doi:10.1017/S0960129515000432.

10 Jonas Frey. Realizability Toposes from Specifications. In Thorsten Altenkirch, editor, *13th International Conference on Typed Lambda Calculi and Applications (TLCA 2015)*, volume 38 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 196–210, Dagstuhl, Germany, 2015. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik. doi:10.4230/LIPIcs.TLCA.2015.196.

11 Jonas Frey. Classical Realizability in the CPS Target Language. *Electronic Notes in Theoretical Computer Science*, 325(Supplement C):111–126, 2016. The Thirty-second Conference on the Mathematical Foundations of Programming Semantics (MFPS XXXII). doi:10.1016/j.entcs.2016.09.034.

12 Jean-Yves Girard. Linear logic. *Theoretical Computer Science*, 50(1):1–101, 1987. doi:10.1016/0304-3975(87)90045-4.

13 Timothy G. Griffin. A Formulae-as-type Notion of Control. In *Proceedings of the 17th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*, POPL '90, pages 47–58, New York, NY, USA, 1990. ACM. doi:10.1145/96709.96714.

14 Hugo Herbelin. An Intuitionistic Logic that Proves Markov's Principle. In *LICS 2010, Proceedings*, 2010. doi:10.1109/LICS.2010.49.

15 Hugo Herbelin. A Constructive Proof of Dependent Choice, Compatible with Classical Logic. In *Proceedings of the 27th Annual IEEE Symposium on Logic in Computer Science, LICS 2012, Dubrovnik, Croatia, June 25-28, 2012*, pages 365–374. IEEE Computer Society, 2012. doi:10.1109/LICS.2012.47.

16 Pieter Hofstra and Jaap Van Oosten. Ordered partial combinatory algebras. *Mathematical Proceedings of the Cambridge Philosophical Society*, 134(3):445–463, 2003. doi:10.1017/S0305004102006424.

17 Guilhem Jaber, Gabriel Lewertowski, Pierre-Marie Pédrot, Matthieu Sozeau, and Nicolas Tabareau. The Definitional Side of the Forcing. In *Proceedings of the 31st Annual ACM/IEEE Symposium on Logic in Computer Science*, LICS '16, pages 367–376, New York, NY, USA, 2016. ACM. doi:10.1145/2933575.2935320.

18 Guilhem Jaber, Nicolas Tabareau, and Matthieu Sozeau. Extending Type Theory with Forcing. In *Proceedings of the 2012 27th Annual IEEE/ACM Symposium on Logic in Computer*

Science, LICS '12, pages 395–404, Washington, DC, USA, 2012. IEEE Computer Society. doi:10.1109/LICS.2012.49.

19 J.-L. Krivine. Dependent choice, 'quote' and the clock. *Th. Comp. Sc.*, 308:259–276, 2003.

20 J.-L. Krivine. Realizability in classical logic. In Interactive models of computation and program behaviour. *Panoramas et synthèses*, 27, 2009.

21 J.-L. Krivine. Realizability algebras: a program to well order R. *Logical Methods in Computer Science*, 7(3), 2011. doi:10.2168/LMCS-7(3:2)2011.

22 J.-L. Krivine. Realizability algebras II : new models of ZF + DC. *Logical Methods in Computer Science*, 8(1):10, February 2012. 28 p. doi:10.2168/LMCS-8(1:10)2012.

23 J.-L. Krivine. Quelques propriétés des modèles de réalisabilité de ZF, February 2014. URL: http://hal.archives-ouvertes.fr/hal-00940254.

24 Jean-Louis Krivine. On the Structure of Classical Realizability Models of ZF. In Hugo Herbelin, Pierre Letouzey, and Matthieu Sozeau, editors, *20th International Conference on Types for Proofs and Programs (TYPES 2014)*, volume 39 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 146–161, Dagstuhl, Germany, 2015. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik. doi:10.4230/LIPIcs.TYPES.2014.146.

25 Jean-Louis Krivine. Bar Recursion in Classical Realisability: Dependent Choice and Continuum Hypothesis. In Jean-Marc Talbot and Laurent Regnier, editors, *25th EACSL Annual Conference on Computer Science Logic (CSL 2016)*, volume 62 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 25:1–25:11, Dagstuhl, Germany, 2016. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik. doi:10.4230/LIPIcs.CSL.2016.25.

26 Olivier Laurent. *A study of polarization in logic.* PhD thesis, Université de la Méditerranée - Aix-Marseille II, March 2002. URL: https://tel.archives-ouvertes.fr/tel-00007884.

27 Rodolphe Lepigre. A Classical Realizability Model for a Semantical Value Restriction. In Peter Thiemann, editor, *Programming Languages and Systems - 25th European Symposium on Programming, ESOP 2016, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2016, Eindhoven, The Netherlands, April 2-8, 2016, Proceedings*, volume 9632 of *Lecture Notes in Computer Science*, pages 476–502. Springer, 2016. doi:10.1007/978-3-662-49498-1_19.

28 Paul Blain Levy. *Call-By-Push-Value: A Functional/Imperative Synthesis*, volume 2 of *Semantics Structures in Computation*. Springer, 2004. doi:10.1007/978-94-007-0954-6.

29 A. Miquel. Forcing as a Program Transformation. In *LICS*, pages 197–206. IEEE Computer Society, 2011. doi:10.1109/LICS.2011.47.

30 Alexandre Miquel. Existential witness extraction in classical realizability and via a negative translation. *Logical Methods in Computer Science*, 7(2):188–202, 2011. doi:10.2168/LMCS-7(2:2)2011.

31 Alexandre Miquel. Implicative algebras: a new foundation for realizability and forcing. *ArXiv e-prints*, 2018. arXiv:1802.00528.

32 Étienne Miquey. *Classical realizability and side-effects.* Ph.D. thesis, Université Paris Diderot ; Universidad de la República, Uruguay, November 2017. URL: https://hal.inria.fr/tel-01653733.

33 Étienne Miquey. A Sequent Calculus with Dependent Types for Classical Arithmetic. In *LICS 2018*, pages 720–729. ACM, 2018. doi:10.1145/3209108.3209199.

34 Étienne Miquey. Formalizing Implicative Algebras in Coq. In Jeremy Avigad and Assia Mahboubi, editors, *Interactive Theorem Proving*, pages 459–476. Springer International Publishing, 2018. doi:10.1007/978-3-319-94821-8_27.

35 Guillaume Munch-Maccagnoni. Focalisation and Classical Realisability. In Erich Grädel and Reinhard Kahle, editors, *Computer Science Logic '09*, volume 5771 of *Lecture Notes in Computer Science*, pages 409–423. Springer, Heidelberg, 2009. doi:10.1007/978-3-642-04027-6_30.

36 P. Oliva and T. Streicher. On Krivine's Realizability Interpretation of Classical Second-Order Arithmetic. *Fundam. Inform.*, 84(2):207–220, 2008. URL: http://iospress.metapress.com/content/f51774wm73404583/.

37 Andrew M. Pitts. Tripos theory in retrospect. *Mathematical Structures in Computer Science*, 12(3):265–279, 2002. doi:10.1017/S096012950200364X.

38 Thomas Streicher. Krivine's classical realisability from a categorical perspective. *Mathematical Structures in Computer Science*, 23(6):1234–1256, 2013. doi:10.1017/S0960129512000989.

39 Jaap van Oosten. Studies in Logic and the Foundations of Mathematics. In *Realizability: An Introduction to its Categorical Side*, volume 152 of *Studies in Logic and the Foundations of Mathematics*, pages ii–. Elsevier, 2008. doi:10.1016/S0049-237X(13)72046-9.

40 Jaap van Oosten and Zou Tingxiang. Classical and Relative Realizability. *Theory and Applications of Categories*, 31:571–593, March 2016.

41 Alfred North Whitehead and Bertrand Russell. *Principia Mathematica*. Cambridge University Press, 1925–1927.