Revisiting the duality of computation: an algebraic analysis of classical realizability models

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

Abstract

In an impressive series of papers, Krivine showed at the edge of the last decade how classical realizability furnishes 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...

1 Introduction

It is well-known since Griffin's seminal work [12] 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 [19], 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* [19], 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 [20], and gives rise to surprisingly new models. In particular, its generalizes Cohen's forcing [20, 27] and allows for the direct definition of a model in which neither the continuum hypothesis nor the axiom of choice holds [22].

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* [35], followed by Ferrer, Frey, Guillermo, Malherbe and Miquel who introduced other structures peculiar to classical realizability [7, 8, 5, 9, 10, 37]. 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 [35] how classical realizability could be analyzed in terms of *triposes* [34], the categorical framework emerging from intuitionistic realizability models, while the later work of Ferrer *et al.* [7, 8] connected it to Hofstra and Van Oosten's notion of *ordered combinatory algebras* [15]. More recently, Alexandre Miquel introduced the concept of *implicative algebra* [28], 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 [18, 17, 26, 13, 16]. 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 [18], using memoisation [14, 31] or with a bar recursor [24]. Yet, such choices may also have an impact on the structures of the corresponding realizability models: the non-deterministic operator ⫛ is known to make the model collapse on a forcing situation [21], while the bar recursor requires some continuity properties [24].

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 [32], and which 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.

Due to the lack of space, proofs are given in appendices. Most of them 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 computation 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].

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

[1] Available at `https://gitlab.com/emiquey/ImplicativeAlgebras/`
[2] For a detailed introduction on this topic, we refer the reader to [19] or [29].
[3] One specificity of Krivine classical realizability is that the set of terms contains the control operator $\boldsymbol{cc}$ 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 [19, 29].

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 [15, 36, 35, 6], also induces an implicative structure [30]. 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]: $a \to b \triangleq a^{\perp\!\!\perp} \cdot b = \{t \cdot \pi : t \in a^{\perp\!\!\perp}, \pi \in b\}$ ([28, 30].

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 $\mathbf{k}$ and $\mathbf{s}$, 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 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)

[4] This is actually nothing more than the definition of the falsity value $\|A \Rightarrow B\|$.
[5] The latter indeed implies the closure of separators under application.

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* $(\mathcal{A}, \preccurlyeq, \to)$ *where* $\mathcal{A} = \mathcal{P}(\Pi)$*,* $a \preccurlyeq b \Leftrightarrow a \supseteq b$ *and* $a \to b = a^{\perp\!\!\perp} \cdot b$*. The set of realized formulas, namely* $\mathcal{S} = \{a \in \mathcal{A} : a^{\perp\!\!\perp} \cap \mathbf{PL} \neq \emptyset\}$*, defines a valid separator [28].*

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 $a \to b \in \mathcal{S}$. 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: $a \vdash_{\mathcal{S}} b \to c \quad$ if and only if $\quad a \times b \vdash_{\mathcal{S}} c$.

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: $[a] \preccurlyeq_{\mathcal{H}} [b] \quad \triangleq \quad a \vdash_{\mathcal{S}} b$ (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 $(\mathcal{H}, \preccurlyeq_{\mathcal{H}}, \wedge_{\mathcal{H}}, \vee_{\mathcal{H}}, \to_{\mathcal{H}})$.

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 $\mathcal{T} : \mathbf{Set}^{op} \to \mathbf{HA}$ which admits a generic predicate[7]. 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:

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

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

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

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 [28]).

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 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 [11], Girard refines the structure of the sequent calculus LK, introducing in particular negative and positive

[6] That is to say that we define $a \times b \triangleq \curlywedge_{c\in\mathcal{A}}((a \to b \to c) \to c)$ and $a + b \triangleq \curlywedge_{c\in\mathcal{A}}((a \to c) \to (b \to c) \to c)$.
[7] See Appendix A for more details on the tripos construction.

connectives for disjunctions and conjunctions[8]. With this finer set of connectives, Girard shows that 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 [32]. 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 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 $L^{\parr}$, that is Munch-Maccagnoni's system L restricted to the fragment corresponding to negative formulas: $A, B := X \mid A \parr B \mid \neg A \mid \forall X.A$ [32]. To leave room for more details on disjunctive algebras, we elude here the introduction of $L^{\parr}$ and its relation to the call-by-name λ-calculus, we refer the interested reader to Appendix B.1.

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 $L^{\parr}$ 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 (recall 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 $a \preccurlyeq a'$ then $\neg a' \preccurlyeq \neg a$*
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. $\neg \curlywedge_{a\in A} a = \curlyvee_{a\in A} \neg a$

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.*

---

8 We insist on the fact that even though we use linear notations afterwards, nothing will be linear here.

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 (see App. B.1).

▶ **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:*

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

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 (see Proposition 51).*

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 (see Appendix B.3), 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 [7, 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 $a \xrightarrow{\parr} b \triangleq \neg a \parr b$:

▶ **Proposition 9** . *If $(\mathcal{A}, \preccurlyeq, \parr, \neg)$ is a disjunctive structure, then $(\mathcal{A}, \preccurlyeq, \xrightarrow{\parr})$ 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 $\mathcal{A}^{\rightarrow} = (\mathcal{A}, \preccurlyeq, \xrightarrow{\parr})$ 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 $\iota(t^{\mathcal{A}^{\rightarrow}}) = [\![t]\!]^{\mathcal{A}^{\parr}}$.*

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[10] for the disjunction and the negation instead of Hilbert's combinators $\mathbf{s}$ and $\mathbf{k}$. We thus consider the following combinators:

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

[10] These axioms can be found for instance in Whitehead and Russell's presentation of logic [39]. 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.

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 $(\bigcurlywedge_{i\in I} a_i \to b_i) \in \mathcal{S}$ and $\vdash_I a$ for $(\bigcurlywedge_{i\in I} a_i) \in \mathcal{S}$. 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). *Recall 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*[11] *defines a valid separator (see Proposition 74 for further details).*

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:

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

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:*

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

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 ($\neg \bigcurlyvee_{a\in A} a = \bigcurlywedge_{a\in A} \neg a$) 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 [28]).

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 [21, 23, 24, 26], 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 [12]. 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 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, 32], this fragment is dual to the $\mathrm{L}^{\parr}$ 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 $\mathrm{L}^{\parr}$: 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.

[12] 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 [32, 25].

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 Appendix C 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 [32, 25]. 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:* $a \otimes b \triangleq \bot$ $\qquad$ $\neg a \triangleq \top$.

▶ **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 $\mathrm{L}^{\otimes}$ to define:*

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

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 (see Proposition 90).

It is worth noting that even though we can define an arrow by $a \overset{\otimes}{\to} b \triangleq \neg(a \otimes \neg b)$, it does not induce an implicative structure: indeed, the distributivity law is not true in general[13]. In turns, we have another distributivity law which is usually wrong in implicative structure:

$$(\curlyvee_{a\in A} a) \overset{\otimes}{\to} b = \curlywedge_{a \in A} (a \overset{\otimes}{\to} b) \qquad\qquad \curlywedge_{b\in B} (a \overset{\otimes}{\to} b) \not\preccurlyeq a \overset{\otimes}{\to} (\curlywedge_{b\in B} b)$$

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

[13] For instance, it is false in $\mathrm{L}^{\otimes}$ realizability models.

Here again, we can embed $L^{\otimes}$ commands, contexts and terms into any conjunctive structure. The embedding, given in Appendix C.4, is very similar to the one for $L^{\parr}$in disjunctive structure, and is sound with respect to typing and reductions.

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[14], 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[15]:

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

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 $a \overset{\otimes}{\to} b\in\mathcal{S}$ 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 $\boldsymbol{s}_1^{\parr}, \boldsymbol{s}_3^{\parr}, \boldsymbol{s}_3^{\parr}, \boldsymbol{s}_4^{\parr}$ and $\boldsymbol{s}_5^{\parr}$ are equal to the maximal element $\top$. Therefore, the singleton $\{\top\}$ is a valid separator.*

▶ **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.*

▶ **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 $(a_i)_{i\in I},(b_i)_{i\in I}$:*

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

[14] See Appendix C.6 for more details.

[15] Observe that are directly dual to the combinators for disjunctive separators and that they can be alternatively given the shape $\neg\curlyvee_{\_\in\mathcal{A}}\dots$.

where we write $\vdash_I a$ for $(\bigcurlywedge_{i\in I} a_i) \in \mathcal{S}$ 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[16] $b \in \mathcal{S}$. 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:

▶ **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[17], 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:*

---

[16] 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 $a \vdash_{\mathcal{S}} b$.

[17] 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 $\neg\neg t^{\mathcal{A}} \in \mathcal{S}$.

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*[18] *conjunctive algebra. The following functor (where $f : J \to I$) defines a tripos:*

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

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 $\mathrm{L}^{\parr}$. 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 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:*

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

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

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

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

then $(\mathcal{A}^{\otimes}, \lhd, \otimes, \neg)$ 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.

[18] 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.

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

▶ **Theorem 40** . *Let $(\mathcal{A}^{⅋}, \mathcal{S}^{⅋})$ be a disjunctive algebra. The set $\mathcal{S}^{\otimes} \triangleq \{a \in \mathcal{A} : \neg a \in \mathcal{S}^{⅋}\}$ 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 $\mathcal{S}$: by definition, $\neg^2(\mathcal{S}) = \{a : \neg\neg a \in \mathcal{S}\}$, and it is easy to see that $a \in \mathcal{S}$ implies $\neg\neg a \in \mathcal{S}$, hence $\mathcal{S} \subseteq \neg^2(\mathcal{S})$.

Actually, the duality between disjunctive and (classical) conjunctive algebras is even stronger, in the sense that through the translation, the induced triposes are isomorphic. Recall 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 $I \in \mathbf{Set}$) 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 family of maps:*

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

defines a tripos isomorphism.

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 $\neg a ⅋ b$ 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 [32], 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 [28]; 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.

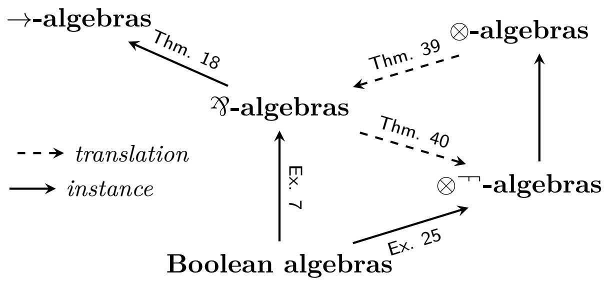

Figure 1 Final picture

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 [33, 27]. 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 $\to$-algebras via $\parr$-algebras.

6.3 Future work

While Theorem 41 implies that call-by-value and call-by-name models based on the $L^{\otimes}$ and $L^{\parr}$ 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 is obviously 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 [14]) or model constructions (*e.g.*, forcing [20]), for the time being we have no idea on how to interpret them in the realm of implicative algebras.

Acknowledgment

The author would like to thank Alexandre Miquel to which several ideas in this paper, especially the definition of conjunctive separators, should be credited; as well as the Uruguayan national research agency ANII for financing a visit in Montevideo which somehow led to the present paper.

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. URL: `http://dx.doi.org/10.1007/3-540-54415-1_73`, `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. W. Ferrer and O. Malherbe. The category of implicative algebras and realizability. *ArXiv e-prints*, December 2017. URL: `https://arxiv.org/abs/1712.06043`, `arXiv:1712.06043`.
6. W. Ferrer Santos, M. Guillermo, and O. Malherbe. A Report on Realizability. *ArXiv e-prints*, 2013. `arXiv:1309.0706`.
7. W. Ferrer Santos, M. Guillermo, and O. Malherbe. Realizability in OCAs and AKSs. *ArXiv e-prints*, 2015. URL: `https://arxiv.org/abs/1512.07879`, `arXiv:1512.07879`.
8. 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`.
9. 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`.
10. 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`.
11. Jean-Yves Girard. Linear logic. *Theoretical Computer Science*, 50(1):1 – 101, 1987. `doi:10.1016/0304-3975(87)90045-4`.
12. 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`.
13. Hugo Herbelin. An intuitionistic logic that proves markov's principle. In *LICS 2010, Proceedings*, 2010. `doi:10.1109/LICS.2010.49`.
14. 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. URL: `http://dx.doi.org/10.1109/LICS.2012.47`, `doi:10.1109/LICS.2012.47`.
15. 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`.
16. 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.

17 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.

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

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

20 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.

21 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.

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

23 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.

24 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.

25 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.

26 A. Miquel. Forcing as a program transformation. In *LICS*, pages 197–206. IEEE Computer Society, 2011.

27 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.

28 Alexandre Miquel. Implicative algebras: a new foundation for realizability and forcing. *ArXiv e-prints*, 2018. URL: https://arxiv.org/abs/1802.00528, arXiv:1802.00528.

29 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.

30 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.

31 Étienne Miquey. A sequent calculus with dependent types for classical arithmetic. In *LICS 2018*, pages 720–729. ACM, 2018. URL: http://doi.acm.org/10.1145/3209108.3209199, doi:10.1145/3209108.3209199.

32 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.

33 P. Oliva and T. Streicher. On Krivine's realizability interpretation of classical second-order arithmetic. *Fundam. Inform.*, 84(2):207–220, 2008.

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

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

36 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.

37 Jaap van Oosten and Zou Tingxiang. Classical and relative realizability. *Theory and Applications of Categories*, 31:571–593, 03 2016.

38 Philip Wadler. Call-by-value is dual to call-by-name. In Colin Runciman and Olin Shivers, editors, *Proceedings of the Eighth ACM SIGPLAN International Conference on Functional Programming, ICFP 2003, Uppsala, Sweden, August 25-29, 2003*, pages 189–201. ACM, 2003. URL: http://doi.acm.org/10.1145/944705.944723, doi:10.1145/944705.944723.

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

A Implicative tripos

▶ **Definition 42** (Hyperdoctrine). *Let $\mathcal{C}$ be a Cartesian closed category. A* first-order hyperdoctrine *over $\mathcal{C}$ is a contravariant functor $\mathcal{T} : \mathcal{C}^{op} \to \mathbf{HA}$ with the following properties:*

1. *For each diagonal morphism $\delta_X : X \to X \times X$ in $\mathcal{C}$, the left adjoint to $\mathcal{T}(\delta_X)$ at the top element $\top \in \mathcal{T}(X)$ exists. In other words, there exists an element $=_X \in \mathcal{T}(X \times X)$ such that for all $\varphi \in \mathcal{T}(X \times X)$:*

$$\top \preccurlyeq \mathcal{T}(\delta_X)(\varphi) \qquad \Leftrightarrow \qquad =_X \preccurlyeq \varphi$$

2. *For each projection $\pi^1_{\Gamma,X} : \Gamma \times X \to \Gamma$ in $\mathcal{C}$, the monotonic function $\mathcal{T}(\pi^1_{\Gamma,X}) : \mathcal{T}(\Gamma) \to \mathcal{T}(\Gamma \times X)$ has both a left adjoint $(\exists X)_\Gamma$ and a right adjoint $(\forall X)_\Gamma$:*

$$\begin{array}{ccc} \varphi \preccurlyeq \mathcal{T}(\pi^1_{\Gamma,X})(\psi) & \Leftrightarrow & (\exists X)_\Gamma(\varphi) \preccurlyeq \psi \\ \mathcal{T}(\pi^1_{\Gamma,X})(\varphi) \preccurlyeq \psi & \Leftrightarrow & \varphi \preccurlyeq (\forall X)_\Gamma(\psi) \end{array}$$

3. *These adjoints are natural in Γ, i.e. given $s : \Gamma \to \Gamma'$ in $\mathcal{C}$, the following diagrams commute:*

$$\begin{tikzcd} \mathcal{T}(\Gamma' \times X) \arrow[r, "\mathcal{T}(s \times id_X)"] \arrow[d, "(\exists X)_{\Gamma'}"'] & \mathcal{T}(\Gamma \times X) \arrow[d, "(\exists X)_\Gamma"] \\ \mathcal{T}(\Gamma') \arrow[r, "\mathcal{T}(s)"'] & \mathcal{T}(\Gamma) \end{tikzcd}$$

$$\begin{tikzcd} \mathcal{T}(\Gamma' \times X) \arrow[r, "\mathcal{T}(s \times id_X)"] \arrow[d, "(\forall X)_{\Gamma'}"'] & \mathcal{T}(\Gamma \times X) \arrow[d, "(\forall X)_\Gamma"] \\ \mathcal{T}(\Gamma') \arrow[r, "\mathcal{T}(s)"'] & \mathcal{T}(\Gamma) \end{tikzcd}$$

This condition is also called the Beck-Chevaley conditions.

The elements of $\mathcal{T}(X)$, as X ranges over the objects of $\mathcal{C}$, are called the $\mathcal{T}$-predicates.

▶ **Definition 43** (Tripos). *A* tripos *over a Cartesian closed category $\mathcal{C}$ is a first-order hyperdoctrine $\mathcal{T} : \mathcal{C}^{op} \to \mathbf{HA}$ which has a* generic predicate, i.e. *there exists an object* $\mathrm{Prop} \in \mathcal{C}$ *and a predicate* $\mathrm{tr} \in \mathcal{T}(\mathrm{Prop})$ *such that for any object $\Gamma \in \mathcal{C}$ and any predicate $\varphi \in \mathcal{T}(\Gamma)$, there exists a (not necessarily unique) morphism $\chi_\varphi \in \mathcal{C}(\Gamma, \mathrm{Prop})$ such that:*

$$\varphi = \mathcal{T}(\chi_\varphi)(\mathrm{tr})$$

Implicative tripos

Let us fix an implicative algebra $(\mathcal{A}, \preccurlyeq, \to, \mathcal{S})$ for the rest of this section. In order to recover a Heyting algebra, it suffices to consider the quotient $\mathcal{A}/_{\cong_\mathcal{S}}$ by the relation $\cong_\mathcal{S}$ defined as $a \cong_\mathcal{S} b \triangleq (a \vdash_\mathcal{S} b \wedge b \vdash_\mathcal{S} a)$. We equip this quotient with the canonical order relation:

$$[a] \preccurlyeq_\mathcal{H} [b] \quad \triangleq \quad a \vdash_\mathcal{S} b \qquad\qquad \text{(for all } a, b \in \mathcal{A})$$

where we write $[a]$ for the equivalence class of $a \in \mathcal{A}$. We define:

$$\begin{array}{rl} [a] \to_{\mathcal{H}} [b] \triangleq [a \to b] & \\ [a] \wedge_{\mathcal{H}} [b] \triangleq [a \times b] & \top_{\mathcal{H}} \triangleq [\top] = \mathcal{S} \\ [a] \vee_{\mathcal{H}} [b] \triangleq [a + b] & \bot_{\mathcal{H}} \triangleq [\bot] = \{a \in \mathcal{A} : \neg a \in \mathcal{S}\} \end{array}$$

The quintuple $(\mathcal{H}, \preccurlyeq_{\mathcal{H}}, \wedge_{\mathcal{H}}, \vee_{\mathcal{H}}, \to_{\mathcal{H}})$ is a Heyting algebra.

We define:

$$\begin{array}{rl} [a] \to_{\mathcal{H}} [b] \triangleq [a \to b] & \\ [a] \wedge_{\mathcal{H}} [b] \triangleq [a \times b] & \top_{\mathcal{H}} \triangleq [\top] = \mathcal{S} \\ [a] \vee_{\mathcal{H}} [b] \triangleq [a + b] & \bot_{\mathcal{H}} \triangleq [\bot] = \{a \in \mathcal{A} : \neg a \in \mathcal{S}\} \end{array}$$

The quintuple $(\mathcal{H}, \preccurlyeq_{\mathcal{H}}, \wedge_{\mathcal{H}}, \vee_{\mathcal{H}}, \to_{\mathcal{H}})$ is a Heyting algebra.

▶ **Theorem 44** (Implicative tripos [28]). *Let $(\mathcal{A}, \preccurlyeq, \to, \mathcal{S})$ be an implicative algebra. The following functor:*

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

defines[19] *a tripos.*

Proof. We verify that $\mathcal{T}$ satisfies all the necessary conditions to be a tripos.

- The functoriality of $\mathcal{T}$ is clear.
- For each $I \in \mathbf{Set}$, the image of the corresponding diagonal morphism $\mathcal{T}(\delta_I)$ associates to any element $[(a_{ij})_{(i,j)\in I\times I}] \in \mathcal{T}(I \times I)$ the element $[(a_{ii})_{i\in I}] \in \mathcal{T}(I)$. We define[20]:

$$(=_I) \;:\; i, j \mapsto \begin{cases} \curlywedge_{a\in\mathcal{A}}(a \to a) & \text{if } i = j \\ \bot \to \top & \text{if } i \neq j \end{cases}$$

  and we need to prove that for all $[a] \in \mathcal{T}(I \times I)$:

$$[\top]_I \preccurlyeq_{\mathcal{S}[I]} \mathcal{T}(\delta_I)(a) \qquad \Leftrightarrow \qquad [=_I] \preccurlyeq_{\mathcal{S}[I\times I]} [a]$$

  Let then $[(a_{ij})_{i,j\in I}]$ be an element of $\mathcal{T}(I \times I)$. From left to right, assume that $[\top]_I \preccurlyeq_{\mathcal{S}[I]} \mathcal{T}(\delta_I)(a)$, that is to say that there exists $s \in \mathcal{S}$ such that for any $i \in I$, $s \preccurlyeq \top \to a_{ii}$. Then it is easy to check that for all $i, j \in I$, $\lambda z.z(s(\lambda x.x)) \preccurlyeq i =_I j \to a_{ij}$. Indeed, using the adjunction and the β-reduction it suffices to show that for all $i, j \in I$, $(i =_I j) \preccurlyeq (s(\lambda x.x)) \to a_{ij}$. If $i = j$, this follows from the fact that $(s(\lambda x.x)) \preccurlyeq a_{ii}$. If $i \neq j$, this is clear by subtyping.

  From right to left, if there exists $s \in \mathcal{S}$ such that for any $i, j \in I$, $s \preccurlyeq i =_I j \to a_{ij}$, then in particular for all $i \in I$ we have $s \preccurlyeq (\lambda x.x) \to a_{ii}$, and then $\lambda\_.s(\lambda x.x) \preccurlyeq \top \to a_{ii}$ which concludes the case.
- For each projection $\pi^1_{I\times J} : I \times J \to I$ in $\mathcal{C}$, the monotone function $\mathcal{T}(\pi^1_{I,J}) : \mathcal{T}(I) \to \mathcal{T}(I \times J)$ has both a left adjoint $(\exists J)_I$ and a right adjoint $(\forall J)_I$ which are defined by:

$$(\forall J)_I\big([(a_{ij})_{i,j\in I\times J}]\big) \triangleq \big[(\underset{j\in J}{\forall} a_{ij})_{i\in I}\big] \qquad\qquad (\exists J)_I\big([(a_{ij})_{i,j\in I\times J}]\big) \triangleq \big[(\underset{j\in J}{\exists} a_{ij})_{i\in I}\big]$$

[19] Note that the definition of the functor on functions $f : J \to I$ assumes implicitly the possibility of picking a representative in any equivalent class $[a] \in \mathcal{A}/\mathcal{S}[I]$, *i.e.* the full axiom of choice.

[20] The reader familiar with classical realizability might recognize the usual interpretation of Leibniz's equality.

The proofs of the adjointness of this definition are again easy manipulation of λ-calculus. We only give the case of $\exists$, the case for $\forall$ is easier. We need to show that for any $[(a_{ij})_{(i,j)\in I\times J}] \in \mathcal{T}(I\times J)$ and for any $[(b_i)_{i\in I}]$, we have:

$$[(a_{ij})_{(i,j)\in I\times J}] \preccurlyeq_{\mathcal{S}[I\times J]} [(b_i)_{(i,j)\in I}] \qquad \Leftrightarrow \qquad [(\underset{j\in J}{\exists} a_{ij})_{i\in I}] \preccurlyeq_{\mathcal{S}[I]} [(b_i)_{i\in I}]$$

Let us fix some $[a]$ and $[b]$ as above. From left to right, assume that there exists $s\in\mathcal{S}$ such that for all $i\in I$, $j\in J$, $s\preccurlyeq a_{ij}\to b_i$, and thus $sa_{ij}\preccurlyeq b_i$. Using the semantic elimination rule of the existential quantifier, we deduce that for all $i\in I$, if $t\preccurlyeq \exists_{j\in J} a_{ij}$, then $t(\lambda x.sx)\preccurlyeq b_i$. Therefore, for all $i\in I$ we have $\lambda y.y(\lambda x.sx)\preccurlyeq \exists_{j\in J} a_{ij}\to b_i$. From right to left, assume that there exists $s\in\mathcal{S}$ such that for all $i\in I$, $s\preccurlyeq \exists_{j\in J} a_{ij}\to b_i$. For any $j\in J$, using the semantic introduction rule of the existential quantifier, we deduce that for all $i\in I$, $\lambda x.xa_{ij}\preccurlyeq \exists_{j\in J} a_{ij}$. Therefore, for all $i\in I$ we have $\lambda x.s(\lambda z.zx)\preccurlyeq a_{ij}\to b_i$.

- These adjoints clearly satisfy the Beck-Chevaley condition. For instance, for the existential quantifier, we have for all I, I', J, for any $[(a_{i'j})_{(i',j)\in I'\times J}]\in\mathcal{T}(I'\times J)$ and any $s: I\to I'$,

$$\begin{aligned}(\mathcal{T}(s)\circ(\exists J)_{I'})([(a_{i'j})_{(i',j)\in I'\times J}]) &= \mathcal{T}(s)\big([(\exists_{j\in J} a_{i'j})_{i'\in I'}\big]) \\ &= \big[(\exists_{j\in J} a_{s(i)j})_{i\in I}\big] \\ &= ((\exists J)_I)([(a_{s(i)j})_{ij\in I\times J}]) \\ &= ((\exists J)_I\circ\mathcal{T}(s\times \mathrm{id}_J)([(a_{ij})_{i,j\in I\times J}])\end{aligned}$$

- Finally, we define $\mathrm{Prop}\triangleq\mathcal{A}$ and verify that $\mathrm{tr}\triangleq[\mathrm{id}_{\mathcal{A}}]\in\mathcal{T}(\mathrm{Prop})$ is a generic predicate. Let then I be a set, and $a=[(a_i)_{i\in I}]\in\mathcal{T}(I)$. We let $\chi_a: i\mapsto a_i$ be the characteristic function of a (it is in $I\to\mathrm{Prop}$), which obviously satisfies that for all $i\in I$:

$$\mathcal{T}(\chi_a)(\mathrm{tr}) = [(\chi_a(i))_{i\in I}] = [(a_i)_{i\in I}]$$

◀

B Disjunctive algebras

B.1 The $L^{⅋}$ calculus

The $L^{⅋}$-calculus is the restriction of Munch-Maccagnoni's system L [32], to the negative fragment corresponding to the connectives ⅋, $\neg^-$ (which we simply write $\neg$ since there is no ambiguity here) and $\forall$. To simplify things (and ease the connection with the $\lambda\mu\tilde{\mu}$-calculus [4]), we slightly change the notations of the original paper. As Krivine's λ_c-calculus, this language describes *commands* of abstract machines c that are made of a *term* t taken within its *evaluation context* e. The syntax is given by[21]:

$$\begin{array}{llcl|llcl}
\textbf{Terms} & t & ::= & x \mid \mu(\alpha_1,\alpha_2).c \mid \mu[x].c \mid \mu\alpha.c & \textbf{Contexts} & e & ::= & \alpha \mid (e_1,e_2) \mid [t] \mid \mu x.c \\
\textbf{Values} & V & ::= & \alpha \mid (V_1,V_2) \mid [t] & \textbf{Commands} & c & ::= & \langle t \,\|\, e\rangle
\end{array}$$

We write $\mathcal{T}_0$, $\mathcal{V}_0$, $\mathcal{E}_0$, $\mathcal{C}_0$ for the sets of closed terms, values, contexts and commands. the corresponding set of closed values. We shall say a few words about it:

- (e_1,e_2) are pairs of contexts, which we will relate to usual stacks;
- $\mu(\alpha_1,\alpha_2).c$, which binds the co-variables α_1,α_2, is the dual destructor for pairs;
- $[t]$ is a constructor for the negation, which allows us to embed a term into a context;
- $\mu[x].c$, which binds the variable x, is the dual destructor;
- $\mu\alpha.c$ binds a covariable and allows to capture a context: as such, it implements classical control.

▶ **Remark 45** (Notations). *We shall explain that in (full) L, the same syntax allows us to define terms t and contexts e (thanks to the duality between them). In particular, no distinction is made between t and e, which are both written t, and commands are indifferently of the shape $\langle t^+ \,\|\, t^-\rangle$ or $\langle t^- \,\|\, t^+\rangle$. For this reason, in [32] is considered a syntax where a notation $\bar{x}$ is used to distinguish between the positive variable x (that can appear in the left-member $\langle x|$ of a command) and the positive co-variable $\bar{x}$ (resp. in the right member $|x\rangle$ of a command). In particular, the $\mu\alpha$ binder of the $\lambda\mu\tilde{\mu}$-calculus would have been written $\mu\bar{x}$ and the $\tilde{\mu}x$ binder would have been denoted by $\mu\alpha$ (see [32, Appendix A.2]). We thus switched the x and α of L (and removed the bar), in order to stay coherent with the notations in the rest of this manuscript.*

The reduction rules correspond to what could be expected from the syntax of the calculus: destructors reduce in front of the corresponding constructors, both μ binders catch values in front of them and pairs of contexts are expanded if they are not values[22].

$$\begin{array}{rcl|rcl}
\langle \mu[x].c \,\|\, [t]\rangle & \to & c[t/x] & \langle \mu(\alpha_1,\alpha_2).c \,\|\, (V_1,V_2)\rangle & \to & c[V_1/\alpha_1, V_2/\alpha_2] \\
\langle t \,\|\, \mu x.c\rangle & \to & c[t/x] & \langle t \,\|\, (e,e')\rangle & \to & \Big\langle \mu\alpha.\langle \mu\alpha'.\langle t \,\|\, (\alpha,\alpha')\rangle \,\|\, e'\rangle \,\Big\|\, e\Big\rangle \\
\langle \mu\alpha.c \,\|\, V\rangle & \to & c[V/\alpha] & & &
\end{array}$$

where in the last rule, $(e,e') \notin V$.

Finally, we shall present the type system of $L^{⅋}$.

---

[21] The reader may observe that in this setting, values are defined as contexts, so that we may have called them *covalues* rather than values. We stick to this denomination to stay coherent with Munch-Maccagnoni's paper [32].

[22] The reader might recognize the rule (ζ) of Wadler's sequent calculus [38].

$$\frac{\Gamma \vdash t : A \mid \Delta \quad \Gamma \mid e : A \vdash \Delta}{\langle t \,\|\, e \rangle : \Gamma \vdash \Delta}\ (\textsc{Cut}) \qquad \frac{(\alpha : A) \in \Delta}{\Gamma \mid \alpha : A \vdash \Delta}\ (ax\vdash) \qquad \frac{(x : A) \in \Gamma}{\Gamma \vdash x : A \mid \Delta}\ (\vdash ax)$$

$$\frac{c : \Gamma, x : A \vdash \Delta}{\Gamma \mid \mu x.c : A \vdash \Delta}\ (\mu\vdash) \qquad \frac{\Gamma \mid e_1 : A \vdash \Delta \quad \Gamma \mid e_2 : B \vdash \Delta}{\Gamma \mid (e_1, e_2) : A \parr B \vdash \Delta}\ (\parr\vdash) \qquad \frac{\Gamma \vdash t : A \mid \Delta}{\Gamma \mid [t] : \neg A \vdash \Delta}\ (\neg\vdash)$$

$$\frac{c : \Gamma \vdash \Delta, \alpha : A}{\Gamma \vdash \mu\alpha.c : A \mid \Delta}\ (\vdash\mu) \qquad \frac{c : \Gamma \vdash \Delta, \alpha_1 : A, \alpha_2 : B}{\Gamma \vdash \mu(\alpha_1, \alpha_2).c : A \parr B \mid \Delta}\ (\vdash\parr) \qquad \frac{c : \Gamma, x : A \vdash \Delta}{\Gamma \vdash \mu[x].c : \neg A \mid \Delta}\ (\vdash\neg)$$

$$\frac{\Gamma \mid e : A[B/X] \vdash \Delta}{\Gamma \mid e : \forall X.A \vdash \Delta}\ (\forall\vdash) \qquad \frac{\Gamma \vdash t : A \mid \Delta \quad X \notin FV(\Gamma, \Delta)}{\Gamma \vdash t : \forall X.A}\ (\vdash\forall)$$

Figure 2 Typing rules for the $\mathrm{L}_{\parr,\neg}$-calculus

In the continuity of the presentation of implicative algebras, we are interested in a second-order settings. Formulas are then defined by the following grammar:

Formulas $\qquad A, B := X \mid A \parr B \mid \neg A \mid \forall X.A$

The type system is presented in a sequent calculus fashion. We work with two-sided sequents, where typing contexts are defined as usual as finite lists of bindings between variable and formulas:

$$\Gamma ::= \varepsilon \mid \Gamma, x : A \qquad\qquad \Delta ::= \varepsilon \mid \Delta, \alpha : A$$

Sequents are of three kinds: $\Gamma \vdash t : A \mid \Delta$ for typing terms, $\Gamma \mid e : A \vdash \Delta$ for typing contexts, $c : \Gamma \vdash \Delta$ for typing commands. In the type system, left rules corresponds to constructors while right rules type destructors. The type system is given in Figure 2.

Embedding of the λ-calculus

Following Munch-Maccagnoni's paper [32, Appendix E], we can embed the λ-calculus into the $\mathrm{L}^{\parr}$-calculus. To this end, we are guided by the expected definition of the arrow $A \to B \triangleq \neg A \parr B$. It is easy to see that with this definition, a stack $u \cdot e$ in $A \to B$ (that is with u a term of type A and e a context of type B) is naturally defined as a shorthand for the pair $([u], e)$, which indeed inhabits the type $\neg A \parr B$. Starting from there, the rest of the definitions are straightforward:

$$\begin{array}{rcl|rcl} u \cdot e & \triangleq & ([u], e) & \lambda x.t & \triangleq & \tilde{\mu}([x], \beta).\langle t \,\|\, \beta \rangle \\ \mu([x], \beta).c & \triangleq & \mu(\alpha, \beta).\langle \mu[x].c \,\|\, \alpha \rangle & t\,u & \triangleq & \mu\alpha.\langle t \,\|\, u \cdot \alpha \rangle \end{array}$$

These definitions are sound with respect to the typing rules expected from the $\lambda\mu\tilde{\mu}$-calculus [4]. In addition, they induce the usual rules of β-reduction for the call-by-name evaluation strategy in the Krivine abstract machine[23]:

$$\langle t\,u \,\|\, \pi \rangle \longrightarrow_\beta \langle t \,\|\, u \cdot \pi \rangle \qquad \langle \lambda x.t \,\|\, u \cdot \pi \rangle \longrightarrow_\beta \langle t[u/x] \,\|\, \pi \rangle \qquad (\pi \in V)$$

[23] Note that in the KAM, all stacks are values.

Realizability models

We briefly go through the definition of the realizability interpretation *à la* Krivine for $L^{\parr}$. As is usual, we begin with the definition of a pole:

▶ **Definition 46** (Pole). *A* pole *is defined as any subset* $\perp\!\!\!\perp \subseteq \mathcal{C}$ *s.t. for all* $c, c' \in \mathcal{C}$*, if* $c \longrightarrow_\beta c'$ *and* $c' \in \perp\!\!\!\perp$ *then* $c \in \perp\!\!\!\perp$*.*

As it is common in Krivine's call-by-name realizability, falsity values are defined primitively as sets of contexts. Truth values are then defined by orthogonality to the corresponding falsity values. We say that a term t is *orthogonal* (with respect to the pole $\perp\!\!\!\perp$) to a context e and we write $t \perp\!\!\!\perp e$ when $\langle t \,\|\, e \rangle \in \perp\!\!\!\perp$. A term t (resp. a context e) is said to be orthogonal to a set $S \subseteq \mathcal{E}_0$ (resp. $S \subseteq \mathcal{T}_0$), which we write $t \perp\!\!\!\perp S$, when for all $e \in S$, t is orthogonal to e. Due to the call-by-name[24] (which is induced here by the choice of connectives), a formula A is primitively interpreted by its *ground falsity value*, which we write $\|A\|_V$ and which is a set in $\mathcal{P}(\mathcal{V}_0)$. Its *truth value* $|A|$ is then defined by orthogonality to $\|A\|_V$ (and is a set in $\mathcal{P}(\mathcal{T}_0)$), while its *falsity value* $\|A\| \in \mathcal{P}(\mathcal{E}_0)$ is again obtained by orthogonality to $|A|$. To ease the definitions we assume that for each subset S of $\mathcal{P}(\mathcal{V}_0)$, there is a constant symbol $\dot{S}$ in the syntax of formula. Given a fixed pole $\perp\!\!\!\perp$, the interpretation is given by:

$$
\begin{array}{rcl}
\|\dot{S}\|_V & \triangleq & S \\
\|\forall X.A\|_V & \triangleq & \bigcup_{S \in \mathcal{P}(\mathcal{V}_0)} \|A\{X := \dot{S}\}\|_V \\
\|A \parr B\|_V & \triangleq & \{(V_1, V_2) : V_1 \in \|A\|_V \wedge V_2 \in \|B\|_V\}
\end{array}
\quad\Bigg|\quad
\begin{array}{rcl}
\|\neg A\|_V & \triangleq & \{[t] : t \in |A|\} \\
|A| & \triangleq & \{t : \forall V \in \|A\|_V, t \perp\!\!\!\perp V\} \\
\|A\| & \triangleq & \{e : \forall t \in |A|, t \perp\!\!\!\perp e\}
\end{array}
$$

We shall now verify that the type system of $L^{\parr}$ is indeed adequate with this interpretation. We first prove the following simple lemma:

▶ **Lemma 47** (Substitution). *Let A be a formula whose only free variable is X. For any closed formula B, if $S = \|B\|_V$, then $\|A[B/X]\|_V = \|A[\dot{S}/X]\|_V$.*

Proof. Easy induction on the structure of formulas, with the observation that the statement for primitive falsity values implies the same statement for truth values ($|A[B/X]| = |A[\dot{S}/X]|$) and falsity values ($\|A[B/X]\| = \|A[\dot{S}/X]\|$). The key case is for the atomic formula $A \equiv X$, where we easily check that:

$$\|X[B/X]\|_V = \|B\|_V = S = \|\dot{S}\|_V = \|X[\dot{S}/X]\|_V$$

◀

We define $\Gamma \cup \Delta$ as the union of both contexts where we annotate the type of hypothesis $(\kappa : A) \in \Delta$ with $\kappa : A^{\perp}$:

$$
\begin{array}{rcl}
\Gamma \cup (\Delta, \kappa : A) & \triangleq & (\Gamma \cup \Delta), \kappa : A^{\perp} \\
\Gamma \cup \varepsilon & \triangleq & \Gamma
\end{array}
$$

The last step before proving adequacy consists in defining substitutions and valuations. We say that a *valuation*, which we write ρ, is a function mapping each second-order variable to a primitive falsity value $\rho(X) \in \mathcal{P}(\mathcal{V}_0)$. A *substitution*, which we write σ, is a function mapping each variable x to a closed term c and each variable α to a closed value $V \in \mathcal{V}_0$:

$$\sigma ::= \varepsilon \mid \sigma, x \mapsto t \mid \sigma, \alpha \mapsto V^{+}$$

[24] See [29, Chapter 3] for a more detailed explanation on this point.

We say that a substitution σ realizes a context Γ and note $\sigma \Vdash \Gamma$ when for each binding $(x : A) \in \Gamma$, $\sigma(x) \in |A|$. Similarly, we say that σ realizes a context Δ if for each binding $(\alpha : A) \in \Delta$, $\sigma(\alpha) \in \|A\|_V$.

We can now state the property of adequacy of the realizability interpretation:

▶ **Proposition 48** (Adequacy). *Let Γ, Δ be typing contexts, ρ be a valuation and σ be a substitution such that $\sigma \Vdash \Gamma[\rho]$ and $\sigma \Vdash \Delta[\rho]$. We have:*

1. *If V^+ is a positive value such that $\Gamma \mid V^+ : A \vdash \Delta$, then $V^+[\sigma] \in \|A[\rho]\|_V$.*
2. *If t is a term such that $\Gamma \vdash t : A \mid \Delta$, then $t[\sigma] \in |A[\rho]|$.*
3. *If e is a context such that $\Gamma \mid e : A \vdash \Delta$, then $e[\sigma] \in \|A[\rho]\|$.*
4. *If c is a command such that $c : (\Gamma \vdash \Delta)$, then $c[\sigma] \in \perp\!\!\!\perp$.*

Proof. We only give some key cases, the full proof can be found in [32]. We proceed by induction over the typing derivations. Let σ be a substitution realizing $\Gamma[\rho]$ and $\Delta[\rho]$.

Case $(\vdash \neg)$.

Assume that we have:

$$\frac{c : \Gamma, x : A \vdash \Delta}{\Gamma \vdash \tilde{\mu}[x].c : \neg A}\ (\vdash\neg)$$

and let $[t]$ be a term in $\|A[\rho]\|_V$, that is to say that $t \in |A[\rho]|$. We know by induction hypothesis that for any valuation $\sigma' \Vdash (\Gamma, x : A)[\rho]$, $c[\sigma'] \in \perp\!\!\!\perp$ and we want to show that $\mu[x].c[\sigma]\perp\!\!\!\perp[t]$. We have that:

$$\mu[x].c\perp\!\!\!\perp[t] \quad \longrightarrow_\beta \quad c[\sigma][t/x] = c[\sigma, x \mapsto t]$$

hence it is enough by saturation to show that $c[\sigma][u/x] \in \perp\!\!\!\perp$. Since $t \in |A[\rho]|$, $\sigma[x \mapsto t] \Vdash (\Gamma, x : A)[\rho]$ and we can conclude by induction hypothesis. The cases for $(\mu \vdash)$, $(\vdash \mu)$ and $(\vdash \parr)$ proceed similarly.

Cases $(\neg \vdash)$.

Trivial by induction hypotheses.

Case $(\parr \vdash)$.

Assume that we have:

$$\frac{\Gamma \mid e_1 : A \vdash \Delta \quad \Gamma \mid u : B \vdash \Delta}{\Gamma \mid (e_1, e_2) : A \parr B \vdash \Delta}\ (\parr\vdash)$$

Let then t be a term in $|(A \parr B)[\rho]|$, to show that $\langle\, t \parallel (e_1, e_2)\,\rangle \in \perp\!\!\!\perp$, we proceed by anti-reduction:

$$\langle t \,\|\, (e, e')\rangle \longrightarrow_\beta \Big\langle \mu\alpha.\big\langle \mu\alpha'.\langle t \,\|\, (\alpha, \alpha')\rangle \,\big\|\, e'\big\rangle \,\Big\|\, e \Big\rangle$$

It now easy to show, using the induction hypotheses for e and e' that this command is in the pole: it suffices to show that the term $\mu\alpha.\big\langle\mu\alpha'.\langle t \,\|\, (\alpha, \alpha')\rangle \,\big\|\, e'\big\rangle \in |A|$, which amounts to showing that for any value $V_1 \in \|A\|_V$:

$$\Big\langle \mu\alpha.\big\langle\mu\alpha'.\langle t \,\|\, (\alpha, \alpha')\rangle \,\big\|\, V\big\rangle \,\Big\|\, \longrightarrow_\beta \Big\rangle \big\langle \mu\alpha'.\langle t \,\|\, (V, \alpha')\rangle \,\big\|\, e'\big\rangle \in \perp\!\!\!\perp$$

Again this holds by showing that for any $V' \in |B|$,

$$\big\langle \mu\alpha'.\langle t \,\|\, (V, \alpha')\rangle \,\big\|\, V'\big\rangle \longrightarrow_\beta \langle t \,\|\, (V, V')\rangle \in \perp\!\!\!\perp$$

Case ($\vdash \forall$).

Trivial.

Case ($\forall \vdash$).

Assume that we have:
$$\frac{\Gamma \mid e : A[B/X] \vdash \Delta}{\Gamma \mid e : \forall X.A \vdash \Delta}\ (\forall \vdash)$$

By induction hypothesis, we obtain that $e[\sigma] \in \|A[B/X][\rho]\|$; so that if we denote $\|B[\rho]\|_V \in \mathcal{P}(\mathcal{V}_0)$
by S, we have:

$$e[\sigma] \in \|A[\dot{S}/X]\| \subseteq S \in \mathcal{P}(\mathcal{V}_0) \sqcup \|A[\dot{S}/X][\rho]\|_V^{\perp\!\!\perp\perp\!\!\perp} \subseteq (S \in \mathcal{P}(\mathcal{V}_0) \sqcup \|A[\dot{S}/X][\rho]\|_V)^{\perp\!\!\perp\perp\!\!\perp} = \|\forall X.A[\rho]\|$$

where we make implicit use of Lemma 47. ◀

B.2 Disjunctive structures

We should now define the notion of *disjunctive structure*. Regarding the expected commutations, as we choose negative connectives and in particular a universal quantifier, we should define commutations with respect to arbitrary meets. The following properties of the realizability interpretation for $L^{\parr}$ provides us with a safeguard for the definition to come:

▶ **Proposition 49** (Commutations). *In any $L^{\parr}$ realizability model (that is to say for any pole $\perp\!\!\perp$), the following equalities hold:*

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

Proof. **1.** Assume the $X \notin FV(B)$, then we have:

$$\begin{aligned}
\|\forall X.(A \parr B)\|_V &= S \in \mathcal{P}(\mathcal{V}_0) \sqcup \|A\{X := \dot{S}\} \parr B\|_V \\
&= S \in \mathcal{P}(\mathcal{V}_0) \sqcup \{(V_1, V_2) : V_1 \in \|A\{X := \dot{S}\}\|_V \wedge V_2 \in \|B\|_V\} \\
&= \{(V_1, V_2) : V_1 \in S \in \mathcal{P}(\mathcal{V}_0) \sqcup \|A\{X := \dot{S}\}\|_V \wedge V_2 \in \|B\|_V\} \\
&= \{(V_1, V_2) : V_1 \in \|\forall X.A\|_V \wedge V_2 \in \|B\|\} \quad = \quad \|(\forall X.A) \parr B\|_V
\end{aligned}$$

2. Identical.
3. The proof is again a simple unfolding of the definitions:

$$\begin{aligned}
\|\neg(\forall X.A)\|_V = \{[t] : t \in |\forall X.A|\} \quad &= \quad \{[t] : t \in \bigcap_{S \in \mathcal{P}(\mathcal{V}_0)} |A\{X := \dot{S}\}|\} \\
&= \bigcap_{S \in \mathcal{P}(\mathcal{V}_0)} \{[t] : t \in |A\{X := \dot{S}\}|\} = \bigcap_{S \in \mathcal{P}(\mathcal{V}_0)} \|\neg A\{X := \dot{S}\}\|_V
\end{aligned}$$

◀

▶ **Proposition 50**. *If $(\mathcal{A}, \preccurlyeq, \parr, \neg)$ is a disjunctive structure, then the following hold for all $a \in \mathcal{A}$:*

1. $\top \parr a = \top$ *2.* $a \parr \top = \top$ *3.* $\neg\top = \bot$

Proof. Using the axioms of disjunctive structures, we prove:

1. for all $a \in \mathcal{A}$, $\top \parr a = (\curlywedge \emptyset) \parr a = \curlywedge_{x,a\in\mathcal{A}}\{x \parr a : x \in \emptyset\} = \curlywedge \emptyset = \top$
2. for all $a \in \mathcal{A}$, $a \parr \top = a \parr (\curlywedge \emptyset) = \curlywedge_{x,a\in\mathcal{A}}\{a \parr x : x \in \emptyset\} = \curlywedge \emptyset = \top$
3. $\neg\top = \neg(\curlywedge \emptyset) = \curlyvee_{x\in\mathcal{A}}\{\neg x : x \in \emptyset\} = \curlyvee \emptyset = \bot$

◀

Disjunctive structures from $\mathrm{L}^{\parr}$ realizability models

If we abstract the structure of the realizability interpretation of $\mathrm{L}^{\parr}$, it is a structure of the form $(\mathcal{T}_0, \mathcal{E}_0, \mathcal{V}_0, (\cdot,\cdot), [\cdot], \perp\!\!\!\perp)$, where $(\cdot,\cdot)$ is a binary map from $\mathcal{E}_0^2$ to $\mathcal{E}_0$ (whose restriction to $\mathcal{V}_0$ has values in $\mathcal{V}_0$), $[\cdot]$ is an operation from $\mathcal{T}_0$ to $\mathcal{V}_0$, and $\perp\!\!\!\perp \subseteq \mathcal{T}_0 \times \mathcal{E}_0$ is a relation. From this sextuple, we can define:

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

▶ **Proposition 51.** *The quadruple $(\mathcal{A}, \preccurlyeq, \parr, \neg)$ is a disjunctive structure.*

Proof. We show that the axioms of Definition 5 are satisfied.

1. (Contravariance) Let $a, a' \in \mathcal{A}$, such that $a \preccurlyeq a'$ ie $a' \subseteq a$. Then $a^{\perp\!\!\!\perp} \subseteq a'^{\perp\!\!\!\perp}$ and thus
$$\neg a = \{[t] : t \in a^{\perp\!\!\!\perp}\} \subseteq \{[t] : t \in a'^{\perp\!\!\!\perp}\} = \neg a'$$
i.e. $\neg a' \preccurlyeq \neg a$.
2. (Covariance) Let $a, a', b, b' \in \mathcal{A}$ such that $a' \subseteq a$ and $b' \subseteq b$. Then we have
$$a \parr b = \{(V_1, V_2) : V_1 \in a \wedge V_2 \in b\} \subseteq \{(V_1, V_2) : V_1 \in a' \wedge V_2 \in b'\} = a' \parr b'$$
i.e. $a \parr b \preccurlyeq a' \parr b'$.
3. (Distributivity) Let $a \in \mathcal{A}$ and $B \subseteq \mathcal{A}$, we have:
$$\curlywedge_{b\in B}(a \parr b) = \curlywedge_{b\in B}\{(V_1, V_2) : V_1 \in a \wedge e_2 \in b\} = \{(V_1, V_2) : V_1 \in a \wedge V_2 \in \curlywedge_{b\in B} b\} = a \parr (\curlywedge_{b\in B} b)$$
4. (Commutation) Let $B \subseteq \mathcal{A}$, we have (recall that $\curlyvee_{b\in B} b = \bigcap_{b\in B} b$):
$$\curlyvee_{b\in B}(\neg b) = \curlyvee_{b\in B}\{[t] : t \in b^{\perp\!\!\!\perp}\} = \{[t] : t \in \curlyvee_{b\in B} b^{\perp\!\!\!\perp}\} = \{[t] : t \in (\curlywedge_{b\in B} b)^{\perp\!\!\!\perp}\} = \neg(\curlywedge_{b\in B} b)$$

◀

B.3 Interpreting $\mathrm{L}^{\parr}$

Following the interpretation of the λ-calculus in implicative structures, we shall now see how $\mathrm{L}^{\parr}$ commands can be recovered from disjunctive structures. From now on, we assume given a disjunctive structure $(\mathcal{A}, \preccurlyeq, \parr, \neg)$.

B.3.1 Commands

We define the *commands* of the disjunctive structure $\mathcal{A}$ as the pair (a, b) (which we continue to write $\langle a \parallel b \rangle$) with $a, b \in \mathcal{A}$, and we define the pole $\perp\!\!\!\perp$ as the ordering relation $\preccurlyeq$. We write $\mathcal{C}_\mathcal{A} = \mathcal{A} \times \mathcal{A}$ for the set of commands in $\mathcal{A}$ and $(a, b) \in \perp\!\!\!\perp$ for $a \preccurlyeq b$. Besides, we define an ordering on commands which extends the intuition that the order reflect the "definedness" of objects: given two commands c, c' in $\mathcal{C}_\mathcal{A}$, we say that c is lower than c' and we write $c \trianglelefteq c'$ if $c \in \perp\!\!\!\perp$ implies that $c' \in \perp\!\!\!\perp$. It is straightforward to check that:

▶ **Proposition 52** . *The relation $\trianglelefteq$ is a preorder.*

Besides, the relation $\trianglelefteq$ verifies the following property of variance with respect to the order $\preccurlyeq$:

▶ **Proposition 53** (Commands ordering). *For all $t, t', \pi, \pi' \in \mathcal{A}$, if $t \preccurlyeq t'$ and $\pi' \preccurlyeq \pi$, then $\langle t \parallel \pi \rangle \trianglelefteq \langle t' \parallel \pi' \rangle$.*

Proof. Trivial by transitivity of $\preccurlyeq$. ◀

Finally, it is worth noting that meets are covariant with respect to $\trianglelefteq$ and $\preccurlyeq$, while joins are contravariant:

▶ **Lemma 54** . *If c and c' are two functions associating to each $a \in \mathcal{A}$ the commands $c(a)$ and $c'(a)$ such that $c(a) \trianglelefteq c'(a)$, then we have:*

$$\bigcurlywedge_{a \in \mathcal{A}} \{a : c(a) \in \perp\!\!\!\perp\} \preccurlyeq \bigcurlywedge_{a \in \mathcal{A}} \{a : c'(a) \in \perp\!\!\!\perp\} \qquad \bigcurlyvee_{a \in \mathcal{A}} \{a : c'(a) \in \perp\!\!\!\perp\} \preccurlyeq \bigcurlyvee_{a \in \mathcal{A}} \{a : c(a) \in \perp\!\!\!\perp\}$$

Proof. Assume c, c' are such that for all $a \in \mathcal{A}$, $ca \trianglelefteq c'a$. Then it is clear that by definition we have the inclusion $\{a \in \mathcal{A} : c(a) \in \perp\!\!\!\perp\} \subseteq \{a \in \mathcal{A} : c'(a) \in \perp\!\!\!\perp\}$, whence the expected results. ◀

B.3.2 Contexts

We are now ready to define the interpretation of $\mathrm{L}^{\parr}$ contexts in the disjunctive structure $\mathcal{A}$. The interpretation for the contexts corresponding to the connectives is very natural:

▶ **Definition 55** (Pairing). *For all $a, b \in \mathcal{A}$, we let $(a, b) \triangleq a \parr b$.*

▶ **Definition 56** (Boxing). *For all $a \in \mathcal{A}$, we let $[a] \triangleq \neg a$.*

Note that with these definitions, the encodings of pairs and boxes directly inherit of the properties of the internal law $\parr$ and $\neg$ in disjunctive structures. As for the binder $\mu x.c$, which we write $\tilde{\mu}^+ c$, it should be defined in such a way that if c is a function mapping each $a \in \mathcal{A}$ to a command $c(a) \in \mathcal{C}_\mathcal{A}$, then $\mu^+.c$ should be "compatible" with any a such that $c(a)$ is well-formed (*i.e.* $c(a) \in \perp\!\!\!\perp$). As it belongs to the side of opponents, the "compatibility" means that it should be greater than any such a, and we thus define it as a join.

▶ **Definition 57** (μ^+). *For all $c : \mathcal{A} \to \mathcal{C}_\mathcal{A}$, we define:*

$$\mu^+.c := \bigcurlyvee_{a \in \mathcal{A}} \{a : c(a) \in \perp\!\!\!\perp\}$$

These definitions enjoy the following properties with respect to the β-reduction and the η-expansion:

▶ **Proposition 58** (Properties of μ^+). *For all functions $c, c' : \mathcal{A} \to \mathcal{C}_\mathcal{A}$, the following hold:*

1. *If for all $a \in \mathcal{A}$, $c(a) \trianglelefteq c'(a)$, then $\mu^+.c' \preccurlyeq \mu^+.c$* (Variance)
2. *For all $t \in \mathcal{A}$, then $\langle t \,\|\, \mu^+.c \rangle \trianglelefteq c(t)$* ($\beta$-reduction)
3. *For all $e \in \mathcal{A}$, then $t = \mu^+.(a \mapsto \langle a \,\|\, e \rangle)$* ($\eta$-expansion)

Proof. **1.** Direct consequence of Proposition 54.
2,3. Trivial by definition of μ^+.

◀

▶ **Remark 59** (Subject reduction). *The β-reduction $c \longrightarrow_\beta c'$ is reflected by the ordering relation $c \trianglelefteq c'$, which reads* "if c is well-formed, then so is c'"*. In other words, this corresponds to the usual property of subject reduction. In the sequel, we will see that β-reduction rules of $L^{⅋}$ will always been reflected in this way through the embedding in disjunctive structures.*

B.3.3 Terms

Dually to the definitions of (positive) contexts μ^+ as a join, we define the embedding of (negative) terms, which are all binders, by arbitrary meets:

▶ **Definition 60** (μ^-). *For all $c : \mathcal{A} \to \mathcal{C}_\mathcal{A}$, we define:*

$$\mu^-.c := \bigwedge_{a \in \mathcal{A}} \{a : c(a) \in \perp\!\!\!\perp\}$$

▶ **Definition 61** ($\mu^{()}c$). *For all $c : \mathcal{A}^2 \to \mathcal{C}_\mathcal{A}$, we define:*

$$\mu^{()}.c := \bigwedge_{a,b \in \mathcal{A}} \{a ⅋ b : c(a,b) \in \perp\!\!\!\perp\}$$

▶ **Definition 62** ($\mu^{[]}$). *For all $c : \mathcal{A} \to \mathcal{C}_\mathcal{A}$, we define:*

$$\mu^{[]}.c := \bigwedge_{a \in \mathcal{A}} \{\neg a : c(a) \in \perp\!\!\!\perp\}$$

These definitions also satisfy some variance properties with respect to the preorder $\trianglelefteq$ and the order relation $\preccurlyeq$, namely, negative binders for variable ranging over positive contexts are covariant, while negative binders intended to catch negative terms are contravariant.

▶ **Proposition 63** (Variance). *For any functions c, c' with the corresponding arities, the following hold:*

1. *If $c(a) \trianglelefteq c'(a)$ for all $a \in \mathcal{A}$, then $\mu^-.c \preccurlyeq \mu^-.c'$*
2. *If $c(a,b) \trianglelefteq c'(a,b)$ for all $a, b \in \mathcal{A}$, then $\mu^{()}.c \preccurlyeq \mu^{()}.c'$*
3. *If $c(a) \trianglelefteq c'(a)$ for all $a \in \mathcal{A}$, then $\mu^{[]}.c' \preccurlyeq \mu^{[]}.c$*

Proof. Direct consequences of Proposition 54. ◀

The η-expansion is also reflected as usual by the ordering relation $\preccurlyeq$:

▶ **Proposition 64** (η-expansion). *For all $t \in \mathcal{A}$, the following holds:*

1. $t = \mu^-.(a \mapsto \langle t \,\|\, a \rangle)$
2. $t \preccurlyeq \mu^{()}.(a, b \mapsto \langle t \,\|\, (a,b) \rangle)$
3. $t \preccurlyeq \mu^{[]}.(a \mapsto \langle t \,\|\, [a] \rangle)$

Proof. Trivial from the definitions. ◀

The β-reduction is reflected by the preorder $\trianglelefteq$:

▶ **Proposition 65** (β-reduction). *For all $e, e_1, e_2, t \in \mathcal{A}$, the following holds:*

1. $\langle \mu^{-}.c \,\|\, e\rangle \trianglelefteq c(e)$
2. $\langle \mu^{()}.c \,\|\, (e_1, e_2)\rangle \trianglelefteq c(e_1, e_2)$
3. $\langle \mu^{[]}.c \,\|\, [t]\rangle \trianglelefteq c(t)$

Proof. Trivial from the definitions. ◀

Finally, we call a *$L^{⅋}$ term with parameters in $\mathcal{A}$* (resp. context, command) any $\mathrm{L}^{⅋}$ term (possibly) enriched with constants taken in the set $\mathcal{A}$. Commands with parameters are equipped with the same rules of reduction as in $\mathrm{L}^{⅋}$, considering parameters as inert constants. To every closed $\mathrm{L}^{⅋}$ term t (resp. context e,command c) we associate an element $t^{\mathcal{A}}$ (resp. $e^{\mathcal{A}}$, $c^{\mathcal{A}}$) of $\mathcal{A}$, defined by induction on the structure of t as follows:

Contexts :

$$\begin{array}{rcl} a^{\mathcal{A}} & \triangleq & a \\ (e_1, e_2)^{\mathcal{A}} & \triangleq & (e_1^{\mathcal{A}}, e_2^{\mathcal{A}}) \\ [t]^{\mathcal{A}} & \triangleq & [t^{\mathcal{A}}] \\ (\mu x.c)^{\mathcal{A}} & \triangleq & \mu^{-}(a \mapsto (c[x := a])^{\mathcal{A}}) \end{array}$$

Terms :

$$\begin{array}{rcl} a^{\mathcal{A}} & \triangleq & a \\ (\mu\alpha.c)^{\mathcal{A}} & \triangleq & \mu^{-}(a \mapsto (c[\alpha := a])^{\mathcal{A}}) \\ (\mu(\alpha_1, \alpha_2).c)^{\mathcal{A}} & \triangleq & \mu^{()}(a, b \mapsto (c[\alpha_1 := a, \alpha_2 := b])^{\mathcal{A}}) \\ (\mu[x].c)^{\mathcal{A}} & \triangleq & \mu^{[]}(a \mapsto (c[x := a])^{\mathcal{A}}) \end{array}$$

Commands: $\qquad \langle t \,\|\, e\rangle^{\mathcal{A}} \triangleq \langle t^{\mathcal{A}} \,\|\, e^{\mathcal{A}}\rangle)$

In particular, this definition has the nice property of making the pole $\perp\!\!\!\perp$ (*i.e.* the order relation $\preccurlyeq$) closed under anti-reduction, as reflected by the following property of $\trianglelefteq$:

▶ **Proposition 66** (Subject reduction). *For any closed commands c_1, c_2 of $L^{⅋}$, if $c_1 \longrightarrow_\beta c_2$ then $c_1^{\mathcal{A}} \trianglelefteq c_2^{\mathcal{A}}$,* i.e. *if $c_1^{\mathcal{A}}$ belongs to $\perp\!\!\!\perp$ then so does $c_2^{\mathcal{A}}$.*

Proof. Direct consequence of Propositions 58 and 100. ◀

B.4 Adequacy

We shall now prove that the interpretation of $\mathrm{L}^{⅋}$ is adequate with respect to its type system. Again, we extend the syntax of formulas to define second-order formulas with parameters by:

$$A, B ::= a \mid X \mid \neg A \mid A ⅋ B \mid \forall X.A \qquad (a \in \mathcal{A})$$

This allows us to embed closed formulas with parameters into the disjunctive structure $\mathcal{A}$. The embedding is trivially defined by:

$$\begin{array}{rcll} a^{\mathcal{A}} & \triangleq & a & \text{(if } a \in \mathcal{A}) \\ (\neg A)^{\mathcal{A}} & \triangleq & \neg A^{\mathcal{A}} & \\ (A ⅋ B)^{\mathcal{A}} & \triangleq & A^{\mathcal{A}} ⅋ B^{\mathcal{A}} & \\ (\forall X.A)^{\mathcal{A}} & \triangleq & \curlywedge_{a \in \mathcal{A}} (A\{X := a\})^{\mathcal{A}} & \end{array}$$

As for the adequacy of the interpretation for the second-order λ_c-calculus, we define substitutions, which we write σ, as functions mapping variables (of terms, contexts and types) to element of $\mathcal{A}$:

$$\sigma ::= \varepsilon \mid \sigma[x \mapsto a] \mid \sigma[\alpha \mapsto a] \mid \sigma[X \mapsto a] \qquad (a \in \mathcal{A},\ x, X \text{ variables})$$

In the spirit of the proof of adequacy in classical realizability, we say that a substitution σ realizes a typing context Γ, which write $\sigma \Vdash \Gamma$, if for all bindings $(x : A) \in \Gamma$ we have $\sigma(x) \preccurlyeq (A[\sigma])^{\mathcal{A}}$. Dually, we say that σ realizes Δ if for all bindings $(\alpha : A) \in \Delta$, we have $\sigma(\alpha) \succcurlyeq (A[\sigma])^{\mathcal{A}}$. We can now prove

▶ **Theorem 67** (Adequacy). *The typing rules of $L^{\parr}$ (Figure 2) are adequate with respect to the interpretation of terms, contexts, commands and formulas. Indeed, for all contexts Γ, Δ, for all formulas with parameters A then for all substitutions σ such that $\sigma \Vdash \Gamma$ and $\sigma \Vdash \Delta$, we have:*

1. *for any term t, if $\Gamma \vdash t : A \mid \Delta$, then $(t[\sigma])^{\mathcal{A}} \preccurlyeq A[\sigma]^{\mathcal{A}}$;*
2. *for any context e, if $\Gamma \mid e : A \vdash \Delta$, then $(e[\sigma])^{\mathcal{A}} \succcurlyeq A[\sigma]^{\mathcal{A}}$;*
3. *for any command c, if $c : (\Gamma \vdash \Delta)$, then $(c[\sigma])^{\mathcal{A}} \in \perp\!\!\!\perp$.*

Proof. By induction over the typing derivations.

Case ($\textsc{Cut}$).

Assume that we have:

$$\frac{\Gamma \vdash t : A \mid \Delta \quad \Gamma \mid e : A \vdash \Delta}{\langle t \| e \rangle : \Gamma \vdash \Delta} \;(\textsc{Cut})$$

By induction hypotheses, we have $(t[\sigma])^{\mathcal{A}} \preccurlyeq A[\sigma]^{\mathcal{A}}$ and $(e[\sigma])^{\mathcal{A}} \succcurlyeq A[\sigma]^{\mathcal{A}}$. By transitivity of the relation $\preccurlyeq$, we deduce that $(t[\sigma])^{\mathcal{A}} \preccurlyeq (e[\sigma])^{\mathcal{A}}$, so that $(\langle t \| e \rangle[\sigma])^{\mathcal{A}} \in \perp\!\!\!\perp$.

Case ($\vdash ax$).

Straightforward, since if $(x : A) \in \Gamma$, then $(x[\sigma])^{\mathcal{A}} \preccurlyeq (A[\sigma])^{\mathcal{A}}$. The case $(ax \vdash)$ is identical.

Case ($\vdash \mu$).

Assume that we have:

$$\frac{c : \Gamma \vdash \Delta, \alpha : A}{\Gamma \vdash \mu\alpha.c : A \mid \Delta} \;(\vdash\mu)$$

By induction hypothesis, we have that $(c[\sigma, \alpha \mapsto (A[\sigma])^{\mathcal{A}}])^{\mathcal{A}} \in \perp\!\!\!\perp$. Then, by definition we have:

$$((\mu\alpha.c)[\sigma])^{\mathcal{A}} = (\mu\alpha.(c[\sigma]))^{\mathcal{A}} = \bigcurlywedge_{b \in \mathcal{A}} \{b : (c[\sigma, \alpha \mapsto b])^{\mathcal{A}} \in \perp\!\!\!\perp\} \preccurlyeq (A[\sigma])^{\mathcal{A}}$$

Case ($\mu \vdash$).

Similarly, assume that we have:

$$\frac{c : \Gamma, x : A \vdash \Delta}{\Gamma \mid \mu x.c : A \vdash \Delta} \;(\mu\vdash)$$

By induction hypothesis, we have that $(c[\sigma, x \mapsto (A[\sigma])^{\mathcal{A}}])^{\mathcal{A}} \in \perp\!\!\!\perp$. Therefore, we have:

$$((\mu x.c)[\sigma])^{\mathcal{A}} = (\mu x.(c[\sigma]))^{\mathcal{A}} = \bigcurlyvee_{b \in \mathcal{A}} \{b : (c[\sigma, x \mapsto b])^{\mathcal{A}} \in \perp\!\!\!\perp\} \succcurlyeq (A[\sigma])^{\mathcal{A}} .$$

Case $(\parr \vdash)$.

Assume that we have:

$$\frac{\Gamma \mid e_1 : A_1 \vdash \Delta \quad \Gamma \mid e_2 : A_2 \vdash \Delta}{\Gamma \mid (e_1, e_2) : A_1 \parr A_2 \vdash \Delta} \ (\parr \vdash)$$

By induction hypotheses, we have that $(e_1[\sigma])^{\mathcal{A}} \succcurlyeq (A_1[\sigma])^{\mathcal{A}}$ and $(e_2[\sigma])^{\mathcal{A}} \succcurlyeq (A_2[\sigma])^{\mathcal{A}}$. Therefore, by monotonicity of the $\parr$ operator, we have:

$$((e_1, e_2)[\sigma])^{\mathcal{A}} = (e_1[\sigma], e_2[\sigma])^{\mathcal{A}} = (e_1[\sigma])^{\mathcal{A}} \parr (e_2[\sigma])^{\mathcal{A}} \succcurlyeq (A_1[\sigma])^{\mathcal{A}} \parr (A_2[\sigma])^{\mathcal{A}}.$$

Case $(\vdash \parr)$.

Assume that we have:

$$\frac{c : \Gamma \vdash \Delta, \alpha_1 : A_1, \alpha_2 : A_2}{\Gamma \vdash \mu(\alpha_1, \alpha_2).c : A_1 \parr A_2 \mid \Delta} \ (\vdash \parr)$$

By induction hypothesis, we get that $(c[\sigma, \alpha_1 \mapsto (A_1[\sigma])^{\mathcal{A}}, \alpha_2 \mapsto (A_2[\sigma])^{\mathcal{A}}])^{\mathcal{A}} \in \perp\!\!\!\perp$. Then by definition we have

$$((\mu(\alpha_1, \alpha_2).c)[\sigma])^{\mathcal{A}} = \bigcurlywedge_{a,b \in \mathcal{A}} \{a \parr b : (c[\sigma, \alpha_1 \mapsto a, \alpha_2 \mapsto b])^{\mathcal{A}} \in \perp\!\!\!\perp\} \preccurlyeq (A_1[\sigma])^{\mathcal{A}} \parr (A_2[\sigma])^{\mathcal{A}}.$$

Case $(\neg \vdash)$.

Assume that we have:

$$\frac{\Gamma \vdash t : A \mid \Delta}{\Gamma \mid [t] : \neg A \vdash \Delta} \ (\neg \vdash)$$

By induction hypothesis, we have that $(t[\sigma])^{\mathcal{A}} \preccurlyeq (A[\sigma])^{\mathcal{A}}$. Then by definition of $[\,]^{\mathcal{A}}$ and covariance of the $\neg$ operator, we have:

$$([t[\sigma]])^{\mathcal{A}} = \neg(t[\sigma])^{\mathcal{A}} \succcurlyeq \neg(A[\sigma])^{\mathcal{A}}.$$

Case $(\vdash \neg)$.

Assume that we have:

$$\frac{c : \Gamma, x : A \vdash \Delta}{\Gamma \vdash \mu[x].c : \neg A \mid \Delta} \ (\vdash \neg)$$

By induction hypothesis, we have that $(c[\sigma, x \mapsto (A[\sigma])^{\mathcal{A}}])^{\mathcal{A}} \in \perp\!\!\!\perp$. Therefore, we have:

$$((\mu[x].c)[\sigma])^{\mathcal{A}} = (\mu[x].(c[\sigma]))^{\mathcal{A}} = \bigcurlywedge_{b \in \mathcal{A}} \{\neg b : (c[\sigma, x \mapsto b])^{\mathcal{A}} \in \perp\!\!\!\perp\} \preccurlyeq \neg(A[\sigma])^{\mathcal{A}}.$$

Case $(\forall \vdash)$.

Assume that we have:

$$\frac{\Gamma \vdash e : A\{X := B\} \mid \Delta}{\Gamma \mid e : \forall X.A \vdash \Delta} \ (\forall \vdash)$$

By induction hypothesis, we have that $(e[\sigma])^{\mathcal{A}} \succcurlyeq ((A\{X := B\})[\sigma])^{\mathcal{A}} = (A[\sigma, X \mapsto (B[\sigma])^{\mathcal{A}}])^{\mathcal{A}}$. Therefore, we have that $(e[\sigma])^{\mathcal{A}} \succcurlyeq (A[\sigma, X \mapsto (B[\sigma])^{\mathcal{A}}])^{\mathcal{A}} \succcurlyeq \bigcurlywedge_{b \in \mathcal{A}} \{A\{X := b\}[\sigma]^{\mathcal{A}}\}$.

Case $(\vdash\forall)$.

Similarly, assume that we have:

$$\frac{\Gamma \vdash t : A \mid \Delta \quad X \notin FV(\Gamma,\Delta)}{\Gamma \vdash t : \forall X.A}\ (\vdash\forall)$$

By induction hypothesis, we have that $(t[\sigma])^{\mathcal{A}} \preccurlyeq (A[\sigma, X \mapsto b])^{\mathcal{A}}$ for any $b \in A$. Therefore, we have that $(t[\sigma])^{\mathcal{A}} \preccurlyeq \curlywedge_{b\in\mathcal{A}}(A\{X := b\}[\sigma]^{\mathcal{A}})$. ◀

B.5 The induced implicative structure

As expected, any disjunctive structures directly induces an implicative structure:

▶ **Proposition 68** . *If $(\mathcal{A}, \preccurlyeq, ⅋, \neg)$ is a disjunctive structure, then $(\mathcal{A}, \preccurlyeq, \rightarrow)$ is an implicative structure.*

Proof. We need to show that the definition of the arrow fulfills the expected axioms:

1. (Variance) Let $a, b, a', b' \in \mathcal{A}$ be such that $a' \preccurlyeq a$ and $b \preccurlyeq b'$, then we have:
$$a \rightarrow b = \neg a ⅋ b \preccurlyeq \neg a' ⅋ b' = a' \rightarrow b'$$
since $\neg a \preccurlyeq \neg a'$ by contra-variance of the negation and $b \preccurlyeq b'$.
2. (Distributivity) Let $a \in \mathcal{A}$ and $B \subseteq \mathcal{A}$, then we have:
$$\curlywedge_{b\in B}(a \rightarrow b) = \curlywedge_{b\in B}(\neg a ⅋ b) = \neg a ⅋ (\curlywedge_{b\in B} b) = a \rightarrow (\curlywedge_{b\in B} b)$$
by distributivity of the infimum over the disjunction.

◀

▶ **Lemma 69** . *The shorthand $\mu([x], \alpha).c$ is interpreted in $\mathcal{A}$ by:*

$$(\mu([x], \alpha).c)^{\mathcal{A}} = \curlywedge_{a,b\in\mathcal{A}} \{(\neg a) ⅋ b : c[x := a, \alpha := b] \in\preccurlyeq\}$$

Proof.

$$\begin{aligned}
\mu([x], \alpha).c)^{\mathcal{A}} &= (\mu(x_0, \alpha).\langle \mu[x].c \,\|\, x_0\rangle)^{\mathcal{A}} \\
&= \curlywedge_{a',b\in\mathcal{A}} \{a' ⅋ b : (\langle \mu[x].c[\alpha := b] \,\|\, a'\rangle)^{\mathcal{A}} \in\preccurlyeq\} \\
&= \curlywedge_{a',b\in\mathcal{A}} \{a' ⅋ b : (\curlywedge_{a\in\mathcal{A}} \{\neg a : c^{\mathcal{A}}[x := a, \alpha := b] \in\preccurlyeq\} \preccurlyeq a'\} \\
&= \curlywedge_{a,b\in\mathcal{A}} \{(\neg a) ⅋ b : c^{\mathcal{A}}[x := a, \alpha := b] \in\preccurlyeq\}
\end{aligned}$$

◀

▶ **Proposition 70** (λ-calculus). *Let $\mathcal{A}^{⅋} = (\mathcal{A}, \preccurlyeq, ⅋, \neg)$ be a disjunctive structure, and $\mathcal{A}^{\rightarrow} = (\mathcal{A}, \preccurlyeq, \rightarrow)$ 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 $L^{⅋}$. Then we have*
$$\iota(t^{\mathcal{A}^{\rightarrow}}) = [\![t]\!]^{\mathcal{A}^{⅋}}$$
where $t^{\mathcal{A}^{\rightarrow}}$ (resp. $t^{\mathcal{A}^{⅋}}$) is the interpretation of t within $\mathcal{A}^{\rightarrow}$ (resp. $\mathcal{A}^{⅋}$).

In other words, this proposition expresses the fact that the following diagram commutes:

$$\begin{tikzcd} \lambda\text{-calculus} \arrow[r, "\llbracket\,\rrbracket"] \arrow[d, "{[\,]^{\mathcal{A}^{\to}}}"'] & \mathrm{L}^{\parr} \arrow[d, "{[\,]^{\mathcal{A}^{\parr}}}"] \\ (\mathcal{A}^{\to}, \preccurlyeq, \to) \arrow[r, "\iota"] & (\mathcal{A}^{\parr}, \preccurlyeq, \parr, \neg) \end{tikzcd}$$

Proof. By induction over the structure of terms.

Case a for some $a \in \mathcal{A}^{\parr}$.

This case is trivial as both terms are equal to a.

Case $\lambda x.u$.

We have $\llbracket \lambda x.u \rrbracket = \mu([x], \alpha).\langle \llbracket t \rrbracket \parallel \alpha \rangle$ and

$$\begin{aligned}
(\mu([x], \alpha).\langle \llbracket t \rrbracket \parallel \alpha \rangle)^{\mathcal{A}^{\parr}} &= \bigcurlywedge_{a,b \in \mathcal{A}} \{\neg a \parr b : (\llbracket t[x := a] \rrbracket^{\mathcal{A}^{\parr}}, b) \in \perp\!\!\!\perp\} \\
&= \bigcurlywedge_{a,b \in \mathcal{A}} \{\neg a \parr b : \llbracket t[x := a] \rrbracket^{\mathcal{A}^{\parr}} \preccurlyeq b\} \\
&= \bigcurlywedge_{a \in \mathcal{A}} (\neg a \parr \llbracket t[x := a] \rrbracket^{\mathcal{A}^{\parr}})
\end{aligned}$$

On the other hand,

$$\iota([\lambda x.t]^{\mathcal{A}^{\to}}) = \iota(\bigcurlywedge_{a \in \mathcal{A}} (a \to (t[x := a])^{\mathcal{A}^{\to}})) = \bigcurlywedge_{a \in \mathcal{A}} (\neg a \parr \iota(t[x := a]^{\mathcal{A}^{\to}}))$$

Both terms are equal since $\llbracket t[x := a] \rrbracket^{\mathcal{A}^{\parr}} = \iota(t[x := a])^{\mathcal{A}^{\to}})$ by induction hypothesis.

Case $u\,v$.

On the one hand, we have $\llbracket u\,v \rrbracket = \mu(\alpha).\langle \llbracket u \rrbracket \parallel ([\llbracket v \rrbracket], \alpha) \rangle$ and

$$\begin{aligned}
(\mu(\alpha).\langle \llbracket u \rrbracket \parallel ([\llbracket v \rrbracket], \alpha) \rangle)^{\mathcal{A}^{\parr}} &= \bigcurlywedge_{a \in \mathcal{A}} \{a : (\llbracket u \rrbracket^{\mathcal{A}^{\parr}}, (\neg \llbracket v \rrbracket^{\mathcal{A}^{\parr}} \parr a)) \in \perp\!\!\!\perp\} \\
&= \bigcurlywedge_{a \in \mathcal{A}} \{a : \llbracket u \rrbracket^{\mathcal{A}^{\parr}} \preccurlyeq (\neg \llbracket v \rrbracket^{\mathcal{A}^{\parr}} \parr a)\}
\end{aligned}$$

On the other hand,

$$\iota([u\,v]^{\mathcal{A}^{\to}}) = \iota(\bigcurlywedge_{a \in \mathcal{A}} \{a : (u^{\mathcal{A}^{\to}}) \preccurlyeq (v^{\mathcal{A}^{\to}}) \to a\}) = \bigcurlywedge_{a \in \mathcal{A}} \{a : \iota(u^{\mathcal{A}^{\to}}) \preccurlyeq \neg(\iota(v^{\mathcal{A}^{\to}}) \parr a\}))$$

Both terms are equal since $\llbracket u \rrbracket^{\mathcal{A}^{\parr}} = \iota(u^{\mathcal{A}^{\to}})$ and $\llbracket v \rrbracket^{\mathcal{A}^{\parr}} = \iota(v^{\mathcal{A}^{\to}})$ by induction hypotheses. ◀

B.6 Disjunctive algebras

Separation in disjunctive structures

We recall the definition of separators for disjunctive structures:

▶ **Definition 71.** *11[Separator][ParAlgebra] 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 $b \in \mathcal{S}$.* *(upward closure)*
2. *$\boldsymbol{s}_1, \boldsymbol{s}_2, \boldsymbol{s}_3, \boldsymbol{s}_4$ and $\boldsymbol{s}_5$ are in $\mathcal{S}$.* *(combinators)*
3. *If $a \to b \in \mathcal{S}$ and $a \in \mathcal{S}$ then $b \in \mathcal{S}$.* *(modus ponens)*

A separator $\mathcal{S}$ is said to be consistent *if $\bot \notin \mathcal{S}$.*

▶ **Remark 72** (Generalized Modus Ponens). *The modus ponens, that is the unique rule of deduction 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 $(\curlywedge_{i\in I} a_i) \in \mathcal{S}$. The proof is straightforward using that the separator is closed upwards and by application, and that:

$$\begin{array}{rll} & (\curlywedge_{i\in I} a_i \to b_i)(\curlywedge_{i\in I} a_i) \preccurlyeq (\curlywedge_{i\in I} b_i) & \\ \Leftarrow & (\curlywedge_{i\in I} a_i \to b_i) \preccurlyeq (\curlywedge_{i\in I} a_i) \to (\curlywedge_{i\in I} b_i) & \textit{(by adj.)} \end{array}$$

which is clearly true.

▶ **Example 73** (Realizability model). *Recall 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[25]:*

$$\mathcal{S}_{\perp\!\!\!\perp} \triangleq \{a \in \mathcal{P}(\mathcal{V}_0) : a^{\perp\!\!\!\perp} \cap \mathcal{T}_0 \neq \emptyset\}$$

defines a valid separator. The conditions (1) and (3) are clearly verified (for the same reasons as in the implicative case), but we should verify that the formulas corresponding to the combinators are indeed realized.

Let us then consider the following closed terms:

$$\begin{array}{l} PS_1 \triangleq \mu([x],\alpha).\langle x \,\|\, (\alpha,\alpha)\rangle \\ PS_2 \triangleq \mu([x],\alpha).\langle \mu(\alpha_1,\alpha_2).\langle x \,\|\, \alpha_1\rangle \,\|\, \alpha\rangle \\ PS_3 \triangleq \mu([x],\alpha).\langle \mu(\alpha_1,\alpha_2).\langle x \,\|\, (\alpha_2,\alpha_1)\rangle \,\|\, \alpha\rangle \\ PS_4 \triangleq \mu([x],\alpha).\Big\langle \mu([y],\beta).\Big\langle \mu(\gamma,\delta).\langle y \,\|\, (\gamma, \mu z.\langle x \,\|\, ([z],\delta)\rangle)\rangle \,\Big\|\, \beta\Big\rangle \,\Big\|\, \alpha \Big\rangle \\ PS_5 \triangleq \mu([x],\alpha).\Big\langle \mu(\beta,\alpha_3).\langle \mu(\alpha_1,\alpha_2).\langle x \,\|\, (\alpha_1,(\alpha_2,\alpha_3))\rangle \,\|\, \beta\rangle \,\Big\|\, \alpha\Big\rangle \end{array}$$

▶ **Proposition 74.** *The previous terms have the following types in $L^{\parr}$:*

1. $\vdash PS_1 : \forall A.(A \parr A) \to A \mid$
2. $\vdash PS_2 : \forall AB.A \to A \parr B \mid$
3. $\vdash PS_3 : \forall AB.A \parr B \to B \parr A \mid$

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

4. $\vdash PS_4 : \forall ABC.(A \to B) \to (C \parr A \to C \parr B) \mid$
5. $\vdash PS_5 : \forall ABC.(A \parr (B \parr C)) \to ((A \parr B) \parr C) \mid$

Proof. Straightforward typing derivations in $\mathrm{L}^{\parr}$. ◀

We deduce that $\mathcal{S}_{\perp\!\!\perp}$ is a valid separator:

▶ **Proposition 75.** *The quintuple $(\mathcal{P}(\mathcal{V}_0), \preccurlyeq, \parr, \neg, \mathcal{S}_{\perp\!\!\perp})$ as defined above is a disjunctive algebra.*

Proof. Conditions (1) and (3) are trivial. Condition (2) follows from the previous proposition and the adequacy lemma for the realizability interpretation of $\mathrm{L}^{\parr}$ (Proposition 48). ◀

B.7 Internal logic

From the combinators, we directly get that:

▶ **Proposition 76** (Combinators). *For all $a, b, c \in \mathcal{A}$, the following holds:*

1. $(a \parr a) \vdash_{\mathcal{S}} a$
2. $a \vdash_{\mathcal{S}} (a \parr b)$
3. $(a \parr b) \vdash_{\mathcal{S}} (b \parr a)$
4. $(a \to b) \vdash_{\mathcal{S}} (c \parr a) \to (c \parr b)$
5. $a \parr (b \parr c) \vdash_{\mathcal{S}} (a \parr b) \parr c$

▶ **Proposition 77** (Preorder). *For any $a, b, c \in \mathcal{A}$, we have:*

1. $a \vdash_{\mathcal{S}} a$ *(Reflexivity)*
2. *if $a \vdash_{\mathcal{S}} b$ and $b \vdash_{\mathcal{S}} c$ then $a \vdash_{\mathcal{S}} c$* *(Transitivity)*

Proof. We first that (2) holds by applying twice the closure by modus ponens, then we use it with the relation $a \vdash_{\mathcal{S}} a \parr a$ and $a \parr a \vdash_{\mathcal{S}}$ that can be deduced from the combinators $\boldsymbol{s}_1^{\parr}, \boldsymbol{s}_2^{\parr}$ to get 1. ◀

Negation

We can relate the primitive negation to the one induced by the underlying implicative structure:

▶ **Proposition 78** (Implicative negation). *For all $a \in \mathcal{A}$, the following holds:*

1. $\neg a \vdash_{\mathcal{S}} a \to \bot$
2. $a \to \bot \vdash_{\mathcal{S}} \neg a$

Proof. The first item follows directly from $\boldsymbol{s}_2^{\parr}$ belongs to the separator, since $a \to \bot = \neg a \parr \bot$ and that $\neg a \vdash_{\mathcal{S}} \neg a \parr \bot$.

For the second item, we use the transitivity with the following hypotheses:

$$(a \to \bot) \vdash_{\mathcal{S}} a \to \neg a \qquad\qquad (a \to \neg a) \vdash_{\mathcal{S}} \neg a$$

The statement on the left hand-side is proved by subtyping from the identity ($\bigcurlywedge_{a\in\mathcal{A}}(a \to a)$, which is in $\mathcal{S}$ as the generalized version of $a \vdash_{\mathcal{S}} a$ above). On the right hand-side, we use twice the modus ponens to prove that

$$(a \to a) \vdash_{\mathcal{S}} (\neg a \to \neg a) \to (a \to \neg a) \to \neg a$$

The two extra hypotheses are trivially subtypes of the identity again. This statement follows from this more general property (recall that $a \to a = \neg a \parr a$):

$$\bigcurlywedge_{a,b\in\mathcal{A}} ((a \parr b) \to a + b) \in \mathcal{S}$$

that we shall prove thereafter (see Proposition 80). ◀

Additionally, we can show that the principle of double negation elimination is valid with respect to any separator:

▶ **Proposition 79** (Double negation). *For all $a \in \mathcal{A}$, the following holds:*

1. $a \vdash_{\mathcal{S}} \neg\neg a$
2. $\neg\neg a \vdash_{\mathcal{S}} a$

Proof. The first item is easy since for all $a \in \mathcal{A}$, we have $a \to \neg\neg a = (\neg a) \parr \neg\neg a \cong_{\mathcal{S}} \neg\neg a \parr \neg a = \neg a \to \neg a$. As for the second item, we use Lemma 102 and Proposition 78 to it reduce to the statement:

$$\bigcurlywedge_{a\in\mathcal{A}} ((\neg a) \to \bot) \to a \quad \in \mathcal{S}$$

We use again Lemma 102 to prove it, by showing that:

$$\bigcurlywedge_{a\in\mathcal{A}} ((\neg a) \to \bot) \to (\neg a) \to a \quad \in \mathcal{S} \qquad\qquad \bigcurlywedge_{a\in\mathcal{A}} ((\neg a) \to a) \to (\neg a) \to a \quad \in \mathcal{S}$$

where the statement on the left hand-side from by subtyping from the identity. For the one on the right hand-side, we use the same trick as in the last proof in order to reduce it to:

$$\bigcurlywedge_{a\in\mathcal{A}} (a \to \neg a) \to (a \to a) \to (\neg a \to a) \to a) \ \in \mathcal{S}$$

◀

Sum type

As in implicative structures, we can define the sum type by:

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

We can prove that the disjunction and this sum type are equivalent from the point of view of the separator:

▶ **Proposition 80** (Implicative sum type). *For all $a, b \in \mathcal{A}$, the following holds:*

1. $a \parr b \vdash_{\mathcal{S}} a + b$
2. $a + b \vdash_{\mathcal{S}} a \parr b$

Proof. We prove in both cases a slightly more general statement, namely that the meet over all a, b or the corresponding implication belongs to the separator. For the first item, we have:

$$\bigcurlywedge_{a,b\in\mathcal{A}} (a \parr b) \to a + b = \bigcurlywedge_{a,b,c\in\mathcal{A}} (a \parr b) \to (a \to c) \to (b \to c) \to c$$

Swapping the order of the arguments, we prove that $\bigcurlywedge_{a,b,c\in\mathcal{A}}(b \to c) \to (a \parr b) \to (a \to c) \to c \in \mathcal{S}$. For this, we use Lemma 102 and the fact that:

$$\bigcurlywedge_{a,b,c\in\mathcal{A}} (b \to c) \to (a \parr b) \to (a \parr c) \ \in \mathcal{S} \qquad\qquad \bigcurlywedge_{a,c\in\mathcal{A}} (a \parr c) \to (a \to c) \to c \ \in \mathcal{S}$$

The left hand-side statement is proved using $\boldsymbol{s}_4^{\parr}$, while on the right hand-side we prove it from the fact that:

$$\bigcurlywedge_{a,c\in\mathcal{A}} (a \to c) \to (a \parr c) \to c \parr c \ \in \mathcal{S}$$

which is a subtype of $\boldsymbol{s}_4^{\parr}$, by using Lemma 102 again with $\boldsymbol{s}_1^{\parr}$ and by manipulation on the order of the argument.

The second item is easier to prove, using Lemma 102 again and the fact that:

$$\bigcurlywedge_{a,b\in\mathcal{A}} a+b \to (a \to (a \parr b)) \to (b \to (a \parr b)) \to (a \parr b) \in \mathcal{S}$$

which is a subtype of $\mathbf{I}^{\mathcal{A}}$ (which belongs to $\mathcal{S}$). The other part, which is to prove that:

$$\bigcurlywedge_{a,b\in\mathcal{A}} ((a \to (a \parr b)) \to (b \to (a \parr b)) \to (a \parr b)) \to (a \parr b) \quad \in \mathcal{S}$$

follows from Lemma 72 and the fact that $\bigcurlywedge_{a,b\in\mathcal{A}}(a \to (a \parr b))$ and $\bigcurlywedge_{a,b\in\mathcal{A}}(b \to (a \parr b))$ are both in the separator.

◀

B.8 Induced implicative algebras

▶ **Proposition 81** (Combinator $\mathbf{k}^{\mathcal{A}}$). *We have* $\mathbf{k}^{\mathcal{A}} \in \mathcal{S}$.

Proof. This directly follows by upwards closure from the fact that $\bigcurlywedge_{a,b\in\mathcal{A}}(a \to (b \parr a)) \in \mathcal{S}$. ◀

▶ **Proposition 82** (Combinator $\mathbf{s}^{\mathcal{A}}$). *For any disjunctive algebra* $(\mathcal{A}, \preccurlyeq, \parr, \neg, \mathcal{S})$*, we have* $\mathbf{s}^{\mathcal{A}} \in \mathcal{S}$.

Proof. See Appendix B. We make several applications of Lemmas 102 and 72 consecutively. We prove that:

$$\bigcurlywedge_{a,b,c\in\mathcal{A}} ((a \to b \to c) \to (a \to b) \to a \to c) \quad \in \mathcal{S}$$

is implied by Lemma 102 and:

$$\bigcurlywedge_{a,b,c\in\mathcal{A}} ((a \to b \to c) \to (b \to a \to c)) \ \in \mathcal{S} \qquad \text{and} \qquad \bigcurlywedge_{a,b,c\in\mathcal{A}} ((b \to a \to c) \to (a \to b) \to a \to c) \ \in \mathcal{S}$$

The statement on the left hand-side is an ad-hoc lemma, while the other is proved by generalized transitivity (Lemma 72), using a subtype of $\boldsymbol{s}_4^{\parr}$ as hypothesis, from:

$$\bigcurlywedge_{a,b,c\in\mathcal{A}} ((a \to b) \to (a \to a \to c)) \to (a \to b) \to a \to c \in \mathcal{S}$$

The latter is proved, using again generalized transitivity with a subtype of $\boldsymbol{s}_4^{\parr}$ as premise, from:

$$\bigcurlywedge_{a,b,c\in\mathcal{A}} (a \to a \to c) \to (a \to c) \in \mathcal{S}$$

This is proved using again Lemmas 102 and 72 with $\boldsymbol{s}_5^{\parr}$ and a variant of $\boldsymbol{s}_4^{\parr}$. ◀

▶ **Proposition 83** (Combinator $\mathbf{cc}^{\mathcal{A}}$). *We have* $\mathbf{cc}^{\mathcal{A}} \in \mathcal{S}$.

Proof. We make several applications of Lemmas 102 and 72 consecutively. We prove that:

$$\bigcurlywedge_{a,b\in\mathcal{A}} ((a \to b) \to a) \to a \ \in \mathcal{S}$$

is implied by generalized modus ponens (Lemma 102) and:

$$\bigcurlywedge_{a,b\in\mathcal{A}}((a \to b) \to a) \to (\neg a \to a \to b) \to \neg a \to a \ \in \mathcal{S}$$

and

$$\bigcurlywedge_{a,b\in\mathcal{A}}((\neg a \to a \to b) \to \neg a \to a) \to a \ \in \mathcal{S}$$

The statement above is a subtype of $\boldsymbol{s}_4^{\parr}$, while the other is proved, by Lemma 102, from:

$$\bigcurlywedge_{a,b\in\mathcal{A}}((\neg a \to a \to b) \to \neg a \to a) \to \neg a \to a \ \in \mathcal{S}$$

and

$$\bigcurlywedge_{a\in\mathcal{A}}((\neg a) \to a) \to a \ \in \mathcal{S}$$

The statement below is proved as in Proposition 16, while the statement above is proved by a variant of the modus ponens and:

$$\bigcurlywedge_{a,b\in\mathcal{A}} (\neg a \to a \to b) \ \in \mathcal{S}$$

We conclude by proving this statement using the connections between $\neg a$ and $a \to \bot$, reducing the latter to:

$$\bigcurlywedge_{a,b\in\mathcal{A}} (a \to \bot) \to a \to b \ \in \mathcal{S}$$

which is a subtype of the identity. ◀

As a consequence, we get the expected theorem:

▶ **Theorem 84** . *Any disjunctive algebra is a classical implicative algebra.*

Proof. The conditions of upward closure and closure under modus ponens coincide for implicative and disjunctive separators, and the previous propositions show that $\mathbf{k}, \mathbf{s}$ and $\mathbf{cc}$ belong to the separator of any disjunctive algebra. ◀

▶ **Corollary 85.** *If t is a closed λ-term and $(\mathcal{A}, \preccurlyeq, \parr, \neg, \mathcal{S})$ a disjunctive algebra, then $t^{\mathcal{A}} \in \mathcal{S}$.*

C Conjunctive algebras

The $\mathrm{L}^{\otimes}$calculus corresponds exactly to the restriction of L to the positive fragment induced by the connectives $\otimes, \neg$ and the existential quantifier $\exists$ [32]. Its syntax is given by:

$$\begin{array}{llll|llll} \textbf{Terms} & t & ::= & x \mid (t,t) \mid [e] \mid \mu\alpha.c & \textbf{Contexts} & e & ::= & \alpha \mid \mu(x,y).c \mid \mu[\alpha].c \mid \mu x.c \\ \textbf{Values} & V & ::= & x \mid (V,V) \mid [e] & \textbf{Commands} & c & ::= & \langle t \,\|\, e\rangle \end{array}$$

We denote by $\mathcal{V}_0$, $\mathcal{T}_0$, $\mathcal{E}_0$, $\mathcal{C}_0$ for the sets of closed values, terms, contexts and commands. The syntax is really close to the one of $\mathrm{L}^{⅋}$: it has the same constructors but on terms, while destructors are now evaluation contexts. We recall the meanings of the different constructions:

- (t,t) are pairs of positive terms;
- $\mu(x_1,x_2).c$, which binds the variables x_1,x_2, is the dual destructor;
- $[e]$ is a constructor for the negation, which allows us to embed a negative context intro a positive term;
- $\mu[x].c$, which binds the variable x, is the dual destructor;
- $\mu\alpha.c$ and $\mu x.c$ are unchanged.

The reduction rules correspond again to the intuition one could have from the syntax of the calculus: all destructors and binders reduce in front of the corresponding values, while pairs of terms are expanded if needed:

$$\begin{array}{rcl|rcl} \langle \mu\alpha.c \,\|\, e\rangle & \to & c[e/\alpha] & \langle (V,V') \,\|\, \mu(x,x').c\rangle & \to & c[V/x,V'/x'] \\ \langle [e] \,\|\, \mu[\alpha].c\rangle & \to & c[e/\alpha] & \langle (t,u) \,\|\, e\rangle & \to & \Big\langle t \,\Big\|\, \mu x.\langle u \,\|\, \mu y.\langle (x,y) \,\|\, e\rangle\rangle \Big\rangle \\ \langle V \,\|\, \mu x.c\rangle & \to & c[V/x] & & & \end{array}$$

where $(t,u) \notin V$ in the last β-reduction rule.

Finally, we shall present the type system of $\mathrm{L}^{\otimes}$. Once more, we are interested in the second-order formulas defined from the positive connectives:

$$\textbf{Formulas} \qquad A,B := X \mid A \otimes B \mid \neg A \mid \exists X.A$$

We still work with threetwo-sided sequents, where typing contexts are defined as finite lists of bindings between variable and formulas:

$$\Gamma ::= \varepsilon \mid \Gamma, x : A \qquad\qquad \Delta ::= \varepsilon \mid \Delta, \alpha : A$$

Sequents are again of three kinds, as in the $\lambda\mu\tilde{\mu}$-calculus and $\mathrm{L}^{⅋}$:

- $\Gamma \vdash t : A \mid \Delta$ for typing terms,
- $\Gamma \mid e : A \vdash \Delta$ for typing contexts,
- $c : \Gamma \vdash \Delta$ for typing commands.

The type system, which uses the same three kinds of sequents for terms, contexts, commands than in $\mathrm{L}^{⅋}$ type system, is given in Figure 3.

C.1 Embedding of the λ-calculus

Guided by the expected definition of the arrow $A \to B \triangleq \neg(A \otimes \neg B)$, we can follow Munch-Maccagnoni's paper [32, Appendix E], to embed the λ-calculus into $\mathrm{L}^{\otimes}$.

With such a definition, a stack $u \cdot e$ in $A \to B$ (that is with u a term of type A and e a context of type B) is naturally embedded as a term $(u,[e])$, which is turn into the context

$$
\dfrac{\Gamma \vdash t : A \mid \Delta \quad \Gamma \mid e : A \vdash \Delta}{\langle t \,\|\, e\rangle : \Gamma \vdash \Delta}\;{\scriptstyle(\textsc{Cut})}
\qquad
\dfrac{(\alpha : A) \in \Delta}{\Gamma \mid \alpha : A \vdash \Delta}\;{\scriptstyle(ax\vdash)}
\qquad
\dfrac{(x : A) \in \Gamma}{\Gamma \vdash x : A \mid \Delta}\;{\scriptstyle(\vdash ax)}
$$

$$
\dfrac{c : \Gamma \vdash \Delta, x : A}{\Gamma \mid \mu x.c : A \vdash \Delta}\;{\scriptstyle(\mu\vdash)}
\qquad
\dfrac{c : (\Gamma, x : A, x' : B \vdash \Delta)}{\Gamma \mid \mu(x,x').c : A \otimes B \vdash \Delta}\;{\scriptstyle(\otimes\vdash)}
\qquad
\dfrac{c : \Gamma, x : A \vdash \Delta}{\Gamma \mid \mu[\alpha].c : \neg A}\;{\scriptstyle(\neg\vdash)}
$$

$$
\dfrac{c : \Gamma, \alpha : A \vdash \Delta}{\Gamma \vdash \mu\alpha.c : A \mid \Delta}\;{\scriptstyle(\vdash\mu)}
\qquad
\dfrac{\Gamma \vdash t : A \mid \Delta \quad \Gamma \vdash u : B \mid \Delta}{\Gamma \vdash (t,u) : A \otimes B \mid \Delta}\;{\scriptstyle(\vdash\otimes)}
\qquad
\dfrac{\Gamma \mid e : A \vdash \Delta}{\Gamma \vdash [e] : \neg A \vdash \Delta}\;{\scriptstyle(\vdash\neg)}
$$

$$
\dfrac{\Gamma \vdash e : A \mid \Delta \quad X \notin FV(\Gamma,\Delta)}{\Gamma \mid e : \exists X.A \vdash \Delta}\;{\scriptstyle(\exists_l)}
\qquad
\dfrac{\Gamma \vdash V : A[B/X] \mid \Delta}{\Gamma \vdash V : \exists X.A}\;{\scriptstyle(\exists_r)}
$$

Figure 3 Typing rules for the $\mathrm{L}^{\otimes}$-calculus

$\mu[\alpha].\langle\, (u,[e]) \,\|\, \alpha \,\rangle$ which indeed inhabits the "arrow" type $\neg(A \otimes \neg B)$. The rest of the definitions are then direct:

$$
\begin{aligned}
\mu(x,[\alpha]).c &\triangleq \mu(x,x').\langle x' \,\|\, \mu[\alpha].c\rangle \\
\lambda x.t &\triangleq [\mu(x,[\alpha]).\langle t \,\|\, \alpha\rangle]
\end{aligned}
$$

These shorthands allow for the expected typing rules:

▶ **Proposition 86.** *The following typing rules are admissible:*

$$
\dfrac{\Gamma, x : A \vdash t : B}{\Gamma \vdash \lambda x.t : A \to B}
\qquad
\dfrac{\Gamma \vdash u : A \mid \Delta \quad \Gamma \mid e : B \vdash \Delta}{\Gamma \mid u \cdot e : A \to B \vdash \Delta}
\qquad
\dfrac{\Gamma \vdash t : A \to B \mid \Delta \quad \Gamma \vdash u : A \mid \Delta}{\Gamma \vdash t\,u : B \mid \Delta}
$$

Proof. Each case is directly derivable from $\mathrm{L}^{\otimes}$ type system. We abuse the notations to denote by (*def*) a rule which simply consists in unfolding the shorthands defining the λ-terms.

- **Case** $\mu(x,[\alpha]).c$:

$$
\dfrac{\dfrac{\dfrac{\dfrac{c : (\Gamma, x : A \vdash \Delta, \alpha : B)}{\Gamma \vdash \mu[x].c : \neg A \mid \Delta, \beta : B}\;{\scriptstyle(\mu\vdash)} \qquad \dfrac{}{\Gamma, x : A, x' : \neg B \vdash x' : \neg B \mid \Delta}\;{\scriptstyle(\vdash ax)}}{\langle x' \,\|\, \mu[\alpha].c\rangle : (\Gamma, x : A, x' : \neg B \vdash \Delta)}\;{\scriptstyle(\textsc{Cut})}}{\Gamma \mid \mu(x,x').\langle x' \,\|\, \mu[\alpha].c\rangle : A \otimes \neg B \vdash \Delta}\;{\scriptstyle(\otimes\vdash)}}{\Gamma \mid \mu(x,[\alpha]).c : A \otimes \neg B \vdash \Delta}\;{\scriptstyle(def)}
$$

- **Case** $\lambda x.t$:

$$
\dfrac{\dfrac{\dfrac{\dfrac{\Gamma, x : A \vdash t : B \mid \Delta \qquad \dfrac{}{\Gamma \mid \beta : B \vdash \Delta, \beta : B}\;{\scriptstyle(ax\vdash)}}{\langle t \,\|\, \beta\rangle : (\Gamma, x : A \vdash \beta : B, \Delta)}\;{\scriptstyle(\textsc{Cut})}}{\Gamma \mid \mu(x,[\beta]).\langle t \,\|\, \beta\rangle : A \otimes \neg B \vdash \Delta}}{\Gamma \vdash [\mu(x,[\beta]).\langle t \,\|\, \beta\rangle] : \neg(A \otimes \neg B) \mid \Delta}\;{\scriptstyle(\vdash\neg)}}{\Gamma \vdash \lambda x.t : A \to B \mid \Delta}\;{\scriptstyle(def)}
$$

- **Case** $u \cdot e$:

$$
\dfrac{\dfrac{\dfrac{\dfrac{\Gamma \vdash u : A \vdash \Delta \qquad \dfrac{\Gamma \mid e : B \vdash \Delta}{\Gamma \vdash [e] : \neg B \mid \Delta}\;{\scriptstyle(\vdash\neg)}}{\Gamma \vdash (u,[e]) : A \otimes \neg B \mid \Delta}\;{\scriptstyle(\vdash\otimes)} \qquad \dfrac{}{\Gamma \mid \alpha : (A \otimes \neg B) \vdash \Delta, \alpha : (A \otimes \neg B)}\;{\scriptstyle(ax\vdash)}}{\langle (u,[e]) \,\|\, \alpha\rangle : (\Gamma \vdash \Delta, \alpha : A \otimes \neg B)}\;{\scriptstyle(\textsc{Cut})}}{\Gamma \mid \mu[\alpha].\langle (u,[e]) \,\|\, \alpha\rangle : \neg(A \otimes \neg B) \vdash \Delta}\;{\scriptstyle(\neg\vdash)}}{\Gamma \mid u \cdot e : A \to B \vdash \Delta}\;{\scriptstyle(def)}
$$

- **Case** $t\,u$:

$$\dfrac{\dfrac{\dfrac{\Gamma \vdash t : A \to B \mid \Delta \qquad \dfrac{\Gamma \vdash u : A \mid \Delta \quad \overline{\Gamma \mid \alpha : B \vdash \Delta, \alpha : B}}{\Gamma \mid u \cdot \alpha : A \to B \vdash \Delta, \alpha : B}}{\langle t \,\|\, u \cdot \alpha \rangle : (\Gamma \vdash \Delta, \alpha : B)}\,(\textsc{Cut})}{\Gamma \vdash \mu\alpha.\langle t \,\|\, u \cdot \alpha \rangle : B \mid \Delta}\,(\vdash\mu)}{\Gamma \vdash t\,u : B \mid \Delta}\,(\mathit{def})$$

◀

Besides, the usual rules of β-reduction for the call-by-value evaluation strategy are simulated through the reduction of $\mathrm{L}^{\otimes}$:

▶ **Proposition 87** (β-reduction). *We have the following reduction rules:*

$$\begin{array}{rcl} \langle t\,u \,\|\, e \rangle & \longrightarrow_\beta & \langle t \,\|\, u \cdot e \rangle \\ \langle \lambda x.t \,\|\, u \cdot e \rangle & \longrightarrow_\beta & \big\langle u \,\big\|\, \mu x.\langle t \,\|\, e \rangle \big\rangle \\ \langle V \,\|\, \mu x.c \rangle & \longrightarrow_\beta & c[V/x] \end{array}$$

Proof. The third rule is included in $\mathrm{L}^{\otimes}$ reduction system, the first follows from:

$$\langle tu \,\|\, e \rangle = \big\langle \mu\alpha.\langle t \,\|\, u \cdot \alpha \rangle \,\big\|\, e \big\rangle \longrightarrow_\beta \langle t \,\|\, u \cdot e \rangle$$

For the second rule, we first check that we have:

$$\langle (V, [e]) \,\|\, \mu(x, [\alpha]).c \rangle = \big\langle (V, [e]) \,\big\|\, \mu(x, x').\langle x' \,\|\, \mu[\alpha].c \rangle \big\rangle \longrightarrow_\beta \langle [e] \,\|\, \mu[\alpha].c[V/X] \rangle \longrightarrow_\beta c[V/x][e/\alpha]$$

from which we deduce:

$$\begin{aligned} \langle \lambda x.t \,\|\, u \cdot e \rangle &= \big\langle [\mu(x, [\alpha]).\langle t \,\|\, \alpha \rangle] \,\big\|\, \mu[\alpha].\langle (u, [e]) \,\|\, \alpha \rangle \big\rangle \\ &\longrightarrow_\beta \big\langle (u, [e]) \,\big\|\, \mu(x, [\alpha]).\langle t \,\|\, \alpha \rangle \big\rangle \\ &\longrightarrow_\beta \Big\langle u \,\Big\|\, \mu y.\big\langle (y, [e]) \,\big\|\, \mu(x, [\alpha]).\langle t \,\|\, \alpha \rangle \big\rangle \Big\rangle \\ &\longrightarrow_\beta \big\langle u \,\big\|\, \mu x.\langle t \,\|\, e \rangle \big\rangle \end{aligned}$$

◀

C.2 A realizability model based on the $\mathrm{L}^{\otimes}$-calculus

We briefly recall the definitions necessary to the realizability interpretation *à la* Krivine of $\mathrm{L}^{\otimes}$. Most of the properties being the same as for $\mathrm{L}^{⅋}$, we spare the reader from a useless copy-paste and go straight to the point. A *pole* is again defined as any subset of $\mathcal{C}_0$ closed by anti-reduction. As usual in call-by-value realizability models [25], formulas are primitively interpreted as sets of values, which we call *ground truth values*, while *falsity values* and *truth values* are then defined by orthogonality. Therefore, an existential formula $\exists X.A$ is interpreted by the union over all the possible instantiations for the primitive truth value of the variable X by a set $S \in \mathcal{P}(\mathcal{V}_0)$. We still assume that for each subset S of $\mathcal{P}(\mathcal{V}_0)$, there is a constant symbol $\dot{S}$ in the syntax. The interpretation is given by:

$$\begin{array}{rcll|rcl} |\dot{S}|_V & \triangleq & S & (\forall S \in \mathcal{P}(\mathcal{V}_0)) & |\exists X.A|_V & \triangleq & \bigcup_{S \in \mathcal{P}(\mathcal{V}_0)} |A\{X := \dot{S}\}|_V \\ |A \otimes B|_V & \triangleq & \{(t, u) : t \in |A|_V \wedge u \in |B|_V\} & & \|A\| & \triangleq & \{e : \forall V \in |A|_V, V \perp\!\!\!\perp e\} \\ |\neg A|_V & \triangleq & \{[e] : e \in \|A\|\} & & |A| & \triangleq & \{t : \forall e \in \|A\|, t \perp\!\!\!\perp e\} \end{array}$$

We define again a *valuation* ρ as a function mapping each second-order variable to a primitive falsity value $\rho(X) \in \mathcal{P}(\mathcal{V}_0)$. In this framework, we say that a *substitution* σ is a function mapping each variable x to a closed value $V \in \mathcal{V}_0$ and each variable α to a closed context $e \in \mathcal{E}_0$. We write $\sigma \Vdash \Gamma$ and we say that a substitution σ realizes a context Γ, when for each binding $(x : A) \in \Gamma$, we have $\sigma(x) \in |A|_V$. Similarly, we say that σ realizes a context Δ if for each binding $(\alpha : A) \in \Delta$, we have $\sigma(\alpha) \in \|A\|$.

▶ **Proposition 88** (Adequacy [32]). *Let Γ, Δ be typing contexts, ρ be a valuation and σ be a substitution which verifies that $\sigma \Vdash \Gamma[\rho]$ and $\sigma \Vdash \Delta[\rho]$. We have:*

1. *If $\vdash V : A \mid$, then $V \in |A|_V$.*
2. *If $\mid e : A \vdash$, then $e \in \|A\|$.*
3. *If $\vdash t : A \mid$, then $t \in |A|$.*
4. *If $c : (\vdash)$, then $c \in \perp\!\!\!\perp$.*

C.3 Conjunctive structures

▶ **Proposition 89** (Commutations). *In any $L^{\otimes}$ realizability model, if $X \notin FV(B)$ the following equalities hold:*

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

Proof. **1.** Assume the $X \notin FV(B)$, then we have:

$$\begin{aligned}
|\exists X.(A \otimes B)|_V &= S \in \mathcal{P}(\mathcal{V}_0) \sqcup |A\{X := \dot{S}\} \otimes B|_V \\
&= S \in \mathcal{P}(\mathcal{V}_0) \sqcup \{(V_1, V_2) : V_1 \in |A\{X := \dot{S}\}|_V \wedge V_2 \in |B|_V\} \\
&= \{(e_1, e_2) : e_1 \in S \in \mathcal{P}(\mathcal{V}_0) \sqcup |A\{X := \dot{S}\}|_V \wedge e_2 \in |B|_V\} \\
&= \{(e_1, e_2) : e_1 \in |\exists X.A|_V \wedge e_2 \in \|B\|\} \quad = \quad |(\exists X.A) \otimes B|_V
\end{aligned}$$

2. Identical.

3. The proof is again a simple unfolding of the definitions:

$$\begin{aligned}
|\neg(\exists X.A)|_V = \{[t] : t \in |\exists X.A|\} \quad &= \quad \{[t] : t \in \bigcap_{S \in \mathcal{P}(\mathcal{V}_0)} |A\{X := \dot{S}\}|\} \\
= \bigcap_{S \in \mathcal{P}(\mathcal{V}_0)} \{[t] : t \in |A\{X := \dot{S}\}]|\} &= \bigcap_{S \in \mathcal{P}(\mathcal{V}_0)} |\neg A\{X := \dot{S}\}|_V
\end{aligned}$$

◀

▶ **Example 22** (Realizability models). *As for the disjunctive case, we can abstract the structure of the realizability interpretation of $L^{\otimes}$ into a structure of the form $(\mathcal{T}_0, \mathcal{E}_0, \mathcal{V}_0, (\cdot,\cdot), [\cdot], \perp\!\!\!\perp)$, where $(\cdot,\cdot)$ is a map from $\mathcal{T}_0^2$ to $\mathcal{T}_0$ (whose restriction to $\mathcal{V}_0$ has values in $\mathcal{V}_0$), $[\cdot]$ is an operation from $\mathcal{E}_0$ to $\mathcal{V}_0$, and $\perp\!\!\!\perp \subseteq \mathcal{T}_0 \times \mathcal{E}_0$ is a relation. From this sextuple we can define:*

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

$(\forall a, b \in \mathcal{A})$

▶ **Proposition 90**. *The quadruple $(\mathcal{A}, \preccurlyeq, \otimes, \neg)$ is a conjunctive structure.*

Proof. We show that the axioms of Definition 19 are satisfied.

1. Anti-monotonicity. Let $a, a' \in \mathcal{A}$, such that $a \preccurlyeq a'$ ie $a \subseteq a'$. Then $a'^{\perp\!\!\!\perp} \subseteq a^{\perp\!\!\!\perp}$ and thus
$$\neg a' = \{[t] : t \in a'^{\perp\!\!\!\perp}\} \subseteq \{[t] : t \in a^{\perp\!\!\!\perp}\} = \neg a$$
i.e. $\neg a' \preccurlyeq \neg a$.
2. Covariance of the conjunction. Let $a, a', b, b' \in \mathcal{A}$ such that $a' \subseteq a$ and $b' \subseteq b$. Then we have
$$a \otimes b = \{(t, u) : t \in a \wedge u \in b\} \subseteq \{(t, u) : t \in a' \wedge u \in b'\} = a' \otimes b'$$
i.e. $a \otimes b \preccurlyeq a' \otimes b'$
3. Distributivity. Let $a \in \mathcal{A}$ and $B \subseteq \mathcal{A}$, we have:
$$\curlyvee_{b \in B} (a \otimes b) = \curlyvee_{b \in B} \{(v, u) : t \in a \wedge u \in b\} = \{(t, u) : t \in a \wedge u \in \curlyvee_{b \in B} b\} = a \otimes (\curlyvee_{b \in B} b)$$
4. Commutation. Let $B \subseteq \mathcal{A}$, we have (recall that $\curlywedge_{b \in B} b = \bigcap_{b \in B} b$):
$$\curlywedge_{b \in B} \{\neg b\} = \curlywedge_{b \in B} \{[t] : t \in b^{\perp\!\!\!\perp}\} = \{[t] : t \in \curlywedge_{b \in B} b^{\perp\!\!\!\perp}\} = \{[t] : t \in (\curlyvee_{b \in B} b)^{\perp\!\!\!\perp}\} = \neg(\curlyvee_{b \in B} b)$$

◀

C.4 Interpreting $\mathrm{L}^{\otimes}$ terms

We shall now see how to embed $\mathrm{L}^{\otimes}$ commands, contexts and terms into any conjunctive structure. For the rest of the section, we assume given a conjunctive structure $(\mathcal{A}, \preccurlyeq, \otimes, \neg)$.

C.4.1 Commands

Following the same intuition as for the embedding of $\mathrm{L}^{\parr}$ into disjunctive structures, we define the *commands* $\langle a \,\|\, b \rangle$ of the conjunctive structure $\mathcal{A}$ as the pairs (a, b), and we define the pole $\perp\!\!\!\perp$ as the ordering relation $\preccurlyeq$. We write $\mathcal{C}_{\mathcal{A}} = \mathcal{A} \times \mathcal{A}$ for the set of commands in $\mathcal{A}$ and $(a, b) \in \perp\!\!\!\perp$ for $a \preccurlyeq b$.

We consider the same relation $\trianglelefteq$ over $\mathcal{C}_{\mathcal{A}}$, which was defined by:
$$c \trianglelefteq c' \quad \triangleq \quad \text{if } c \in \perp\!\!\!\perp \text{ then } c' \in \perp\!\!\!\perp \qquad\qquad (\forall c, c' \in \mathcal{C}_{\mathcal{A}})$$
Since the definition of commands only relies on the underlying lattice of $\mathcal{A}$, the relation $\trianglelefteq$ has the same properties as in disjunctive structures and in particular it defines a preorder (see Appendix B.3.1).

C.4.2 Terms

The definitions of terms are very similar to the corresponding definitions for the dual contexts in disjunctive structures.

▶ **Definition 91** (Pairing). *For all $a, b \in \mathcal{A}$, we let $(a, b) \triangleq a \otimes b$.*

▶ **Definition 92** (Boxing). *For all $a \in \mathcal{A}$, we let $[a] \triangleq \neg a$.*

▶ **Definition 93** (μ^+).
$$\mu^+.c \triangleq \curlywedge_{a \in \mathcal{A}} \{a : c(a) \in \perp\!\!\!\perp\}$$
We have the following properties for μ^+:, whose proofs are trivial:

▶ **Proposition 94** (Properties of μ^+). *For any functions $c, c' : \mathcal{A} \to \mathcal{C}_\mathcal{A}$, the following hold:*

1. *If for all $a \in \mathcal{A}$, $c(a) \trianglelefteq c'(a)$, then $\mu^+.c' \preccurlyeq \mu^+.c$* *(Variance)*
2. *For all $t \in \mathcal{A}$, then $t = \mu^+.(a \mapsto \langle t \,\|\, a\rangle)$* *($\eta$-expansion)*
3. *For all $e \in \mathcal{A}$, then $\langle \mu^+.c \,\|\, e\rangle \trianglelefteq c(e)$* *($\beta$-reduction)*

Proof. **1.** Direct consequence of Proposition 54.
2,3. Trivial by definition of μ^+.
◀

C.4.3 Contexts

Dually to the definitions of the (positive) contexts μ^+ as a meet, we define the embedding of (negative) terms, which are all binders, by arbitrary joins:

▶ **Definition 95** (μ^-). *For all $c : \mathcal{A} \to \mathcal{C}_\mathcal{A}$, we define:*

$$\mu^-.c \triangleq \bigvee_{a \in \mathcal{A}} \{a : c(a) \in \perp\!\!\!\perp\}$$

▶ **Definition 96** ($\mu^{()}$). *For all $c : \mathcal{A}^2 \to \mathcal{C}_\mathcal{A}$, we define:*

$$\mu^{()}.c \triangleq \bigvee_{a,b \in \mathcal{A}} \{a \otimes b : c(a,b) \in \perp\!\!\!\perp\}$$

▶ **Definition 97** ($\mu^{[]}$). *For all $c : \mathcal{A} \to \mathcal{C}_\mathcal{A}$, we define:*

$$\mu^{[]}.c \triangleq \bigvee_{a \in \mathcal{A}} \{\neg a : c(a) \in \perp\!\!\!\perp\}$$

Again, these definitions satisfy variance properties with respect to the preorder $\trianglelefteq$ and the order relation $\preccurlyeq$. Observe that the $\mu^{()}$ and μ^- binders, which are negative binders catching positive terms, are contravariant with respect to these relations while the $\mu^{[]}$ binder, which catches a negative context, is covariant.

▶ **Proposition 98** (Variance). *For any functions c, c' with the corresponding arities, the following hold:*

1. *If $c(a) \trianglelefteq c'(a)$ for all $a \in \mathcal{A}$, then $\mu^-.c' \preccurlyeq \mu^-.c$*
2. *If $c(a,b) \trianglelefteq c'(a,b)$ for all $a, b \in \mathcal{A}$, then $\mu^{()}.c' \preccurlyeq \mu^{()}.c$*
3. *If $c(a) \trianglelefteq c'(a)$ for all $a \in \mathcal{A}$, then $\mu^{[]}.c \preccurlyeq \mu^{[]}.c'$*

Proof. Direct consequences of Proposition 54. ◀

The η-expansion is also reflected by the ordering relation $\preccurlyeq$:

▶ **Proposition 99** (η-expansion). *For all $t \in \mathcal{A}$, the following holds:*

1. *$\mu^-.(a \mapsto \langle t \,\|\, a\rangle) = t$*
2. *$\mu^{()}.(a, b \mapsto \langle t \,\|\, (a,b)\rangle) \preccurlyeq t$*
3. *$\mu^{[]}.(a \mapsto \langle t \,\|\, [a]\rangle) \preccurlyeq t$*

Proof. Trivial from the definitions. ◀

The β-reduction is again reflected by the preorder $\trianglelefteq$ as the property of subject reduction:

▶ **Proposition 100** (β-reduction). *For all $e, e_1, e_2, t \in \mathcal{A}$, the following holds:*

1. $\langle \mu^{-}.c \,\|\, e\rangle \trianglelefteq c(e)$
2. $\langle \mu^{()}.c \,\|\, (e_1, e_2)\rangle \trianglelefteq c(e_1, e_2)$
3. $\langle \mu^{[]}.c \,\|\, [t]\rangle \trianglelefteq c(t)$

Proof. Trivial from the definitions. ◀

C.5 Adequacy

We shall now prove that the interpretation of $\mathrm{L}^{\otimes}$ is adequate with respect to its type system. Again, we extend the syntax of formulas to define second-order formulas with parameters by:

$$A, B ::= a \mid X \mid \neg A \mid A \otimes B \mid \exists X.A \qquad\qquad (a \in \mathcal{A})$$

This allows us to define an embedding of closed formulas with parameters into the conjunctive structure $\mathcal{A}$;

$$\begin{array}{rcl} a^{\mathcal{A}} & \triangleq & a \qquad\qquad (\text{if } a \in \mathcal{A}) \\ (\neg A)^{\mathcal{A}} & \triangleq & \neg A^{\mathcal{A}} \\ (A \otimes B)^{\mathcal{A}} & \triangleq & A^{\mathcal{A}} \otimes B^{\mathcal{A}} \\ (\exists X.A)^{\mathcal{A}} & \triangleq & \curlyvee_{a \in \mathcal{A}} (A\{X := a\})^{\mathcal{A}} \end{array}$$

As in the previous chapter, we define substitutions, which we write σ, as functions mapping variables (of terms, contexts and types) to element of $\mathcal{A}$:

$$\sigma ::= \varepsilon \mid \sigma[x \mapsto a] \mid \sigma[\alpha \mapsto a] \mid \sigma[X \mapsto a] \qquad\qquad (a \in \mathcal{A},\ x, X \text{ variables})$$

We say that a substitution σ realizes a typing context Γ, which write $\sigma \Vdash \Gamma$, if for all bindings $(x : A) \in \Gamma$ we have $\sigma(x) \preccurlyeq (A[\sigma])^{\mathcal{A}}$. Dually, we say that σ realizes Δ if for all bindings $(\alpha : A) \in \Delta$, we have $\sigma(\alpha) \succcurlyeq (A[\sigma])^{\mathcal{A}}$.

▶ **Theorem 101** (Adequacy). *The typing rules of $L^{\otimes}$ (Figure 3) are adequate with respect to the interpretation of terms (contexts,commands) and formulas: for all contexts Γ, Δ, for all formulas with parameters A and for all substitutions σ such that $\sigma \Vdash \Gamma$ and $\sigma \Vdash \Delta$, we have:*

1. *For any term t, if $\Gamma \vdash t : A \mid \Delta$, then $(t[\sigma])^{\mathcal{A}} \preccurlyeq A[\sigma]^{\mathcal{A}}$;*
2. *For any context e, if $\Gamma \mid e : A \vdash \Delta$, then $(e[\sigma])^{\mathcal{A}} \succcurlyeq A[\sigma]^{\mathcal{A}}$;*
3. *For any command c, if $c : (\Gamma \vdash \Delta)$, then $(c[\sigma])^{\mathcal{A}} \in \perp\!\!\!\perp$.*

Proof. By induction on the typing derivations. Since most of the cases are similar to the corresponding cases for the adequacy of the embedding of $\mathrm{L}^{\parr}$ into disjunctive structures, we only give some key cases.

Case ($\vdash \otimes$).

Assume that we have:

$$\frac{\Gamma \vdash t_1 : A_1 \mid \Delta \quad \Gamma \vdash t_2 : A_2 \mid \Delta}{\Gamma \vdash (t_1, t_2) : A_1 \otimes A_2 \mid \Delta}\ (\vdash \otimes)$$

By induction hypotheses, we have that $(t_1[\sigma])^{\mathcal{A}} \preccurlyeq (A_1[\sigma])^{\mathcal{A}}$ and $(t_2[\sigma])^{\mathcal{A}} \preccurlyeq (A_2[\sigma])^{\mathcal{A}}$. Therefore, by monotonicity of the $\otimes$ operator, we have:

$$((t_1, t_2)[\sigma])^{\mathcal{A}} = (t_1[\sigma], t_2[\sigma])^{\mathcal{A}} = (t_1[\sigma])^{\mathcal{A}} \otimes (t_2[\sigma])^{\mathcal{A}} \preccurlyeq (A_1[\sigma])^{\mathcal{A}} \parr (A_2[\sigma])^{\mathcal{A}} .$$

Case $(\otimes\vdash)$.

Assume that we have:

$$\frac{c : \Gamma, x_1 : A_1, x_2 : A_2 \vdash \Delta}{\Gamma \mid \mu(x_1,x_2).c : A_1 \otimes A_2 \vdash \Delta}\ (\otimes\vdash)$$

By induction hypothesis, we get that $(c[\sigma, x_1 \mapsto (A_1[\sigma])^{\mathcal{A}}, x_2 \mapsto (A_2[\sigma])^{\mathcal{A}}])^{\mathcal{A}} \in \perp\!\!\!\perp$. Then by definition we have

$$((\mu(x_1,x_2).c)[\sigma])^{\mathcal{A}} = \bigcurlyvee_{a,b\in\mathcal{A}} \{a \parr b : (c[\sigma, x_1 \mapsto a, x_2 \mapsto b])^{\mathcal{A}} \in \perp\!\!\!\perp\} \succcurlyeq (A_1[\sigma])^{\mathcal{A}} \otimes (A_2[\sigma])^{\mathcal{A}}.$$

Case $(\exists\vdash)$.

Assume that we have:

$$\frac{\Gamma \mid e : A \vdash \Delta \quad X \notin FV(\Gamma,\Delta)}{\Gamma \mid e : \exists X.A \vdash \Delta}\ (\exists\vdash)$$

By induction hypothesis, we have that for all $a \in \mathcal{A}$, $(e[\sigma])^{\mathcal{A}} \succcurlyeq ((A)[\sigma, x \mapsto a])^{\mathcal{A}}$. Therefore, we have that $(e[\sigma])^{\mathcal{A}} \succcurlyeq \bigcurlyvee_{a\in\mathcal{A}}(A\{X := a\}[\sigma])^{\mathcal{A}}$.

Case $(\vdash\exists)$.

Similarly, assume that we have:

$$\frac{\Gamma \vdash t : A\{X := B\} \mid \Delta}{\Gamma \vdash t : \exists X.A \mid \Delta}\ (\vdash\exists)$$

By induction hypothesis, we have that $(t[\sigma])^{\mathcal{A}} \preccurlyeq (A[\sigma, X \mapsto (B[\sigma])^{\mathcal{A}}])^{\mathcal{A}}$. Therefore, we have that $(t[\sigma])^{\mathcal{A}} \preccurlyeq \curlyvee_{b\in\mathcal{A}}\{A\{X := b\}[\sigma]^{\mathcal{A}}\}$. ◀

C.6 Conjunctive algebras

If we analyse the tensorial calculus underlying $\mathrm{L}^{\otimes}$ type system and try to inline all the typing rules involving commands and contexts within the one for terms, we are is left with the following four rules:

$$\frac{}{\Gamma, A \vdash A} \qquad \frac{\Gamma \vdash A \quad \Gamma \vdash B}{\Gamma \vdash A \otimes B} \qquad \frac{\Gamma, A, B \vdash C \quad \Gamma, A, B \vdash \neg C}{\Gamma \vdash \neg(A \otimes B)} \qquad \frac{\Gamma, A \vdash C \quad \Gamma, A \vdash \neg C}{\Gamma \vdash \neg A}$$

This emphasizes that this positive fragment is a *calculus of contradiction*: both deduction rules have a negated formula as a conclusion. This justifies considering the following deduction rule in the separator:

$$\frac{\neg(a \otimes b) \in \mathcal{S} \quad a \in \mathcal{S}}{\neg b \in \mathcal{S}}$$

rather than the modus ponens. The latter can be retrieved when assuming that the separator also satisfies that if $\neg\neg a \in \mathcal{S}$ then $a \in \mathcal{S}$. Computationally, this corresponds to the intuition that when composing values, one essentially gets a computation $(\neg\neg a)$ rather than a value. Extracting the value from a computation requires some extra computational power that is provided by classical control.

C.6.1 Internal logic

▶ **Proposition 102** (Preorder). *For any $a, b, c \in \mathcal{A}$, we have:*

1. $a \vdash_{\mathcal{S}} a$ *(Reflexivity)*
2. *if* $a \vdash_{\mathcal{S}} b$ *and* $b \vdash_{\mathcal{S}} c$ *then* $a \vdash_{\mathcal{S}} c$ *(Transitivity)*

Proof. We deduce (2) from its variant defined in terms of $a \vdash^{\neg} b \triangleq \neg(a \otimes b) \in \mathcal{S}$: if $a \vdash^{\neg} b$ and $\neg b \vdash^{\neg} c$ then $a \vdash^{\neg} c$. As for disjunctive algebras, this is proven by applying twice the deduction rule (3) of separators to prove instead that:

$$\neg(\neg(\neg b \otimes c) \otimes \neg(a \otimes b) \otimes a \otimes c) \in \mathcal{S}$$

This follows directly from the fact that $\boldsymbol{s}_4^{\otimes} \in \mathcal{S}$. ◀

▶ **Proposition 103** (Implicative negation). *For all $a \in \mathcal{A}$, the following holds:*

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

Proof. Easy manipulations of the algebras, see Coq proofs. We sketch the two last to give an idea:

3. Observe that $\boldsymbol{s}_4^{\otimes}$ implies that if $a \vdash_{\mathcal{S}} b$ and $(\neg(c \otimes b)) \in \mathcal{S}$ then $(\neg(c \otimes a)) \in \mathcal{S}$. Then the claim follows from the fact that $\neg\neg(a \otimes \neg\bot) \vdash_{\mathcal{S}} a \otimes \neg\bot$ (by 2) and that $(\neg a) \otimes a \otimes \neg\bot \in \mathcal{S}$ (using $\boldsymbol{s}_2^{\otimes}$).
4. We first prove that $\neg a \vdash_{\mathcal{S}} \neg b$ implies $b \vdash_{\mathcal{S}} a$ and that $\neg\bot = \top$. Then $a \vdash_{\mathcal{S}} a \otimes \top$ follows by upward closure from $\boldsymbol{s}_1^{\otimes}$. ◀

For technical reasons, we define:

$$a \Diamond b \triangleq \bigcurlyvee \{c : a \preccurlyeq \neg(b \otimes c)\}$$

We first show that:

▶ **Lemma 104** (Adjunction). *For any $a, b, c \in \mathcal{A}$, $c \preccurlyeq a \Diamond b$ iff $a \preccurlyeq \neg(b \otimes c)$.*

Proof. ($\Rightarrow$) Assume $c \preccurlyeq a \Diamond b$. We use the transitivity to prove that:

$$a \preccurlyeq \neg(b \otimes (a \Diamond b)) \qquad \neg(b \otimes (a \Diamond b)) \preccurlyeq \neg(b \otimes c)$$

The right hand side follows directly from the assumption, while the left hand side is a consequences of distribution laws:

$$\neg(b \otimes (a \Diamond b)) = \neg(b \otimes \bigcurlyvee \{c : a \preccurlyeq \neg(b \otimes c)\}) = \bigcurlywedge \{\neg(b \otimes c) : a \preccurlyeq \neg(b \otimes c)\} \succcurlyeq a$$

($\Leftarrow$) Trivial by definition of $a \Diamond b$.

◀

▶ **Proposition 105** . *If $a \in \mathcal{S}$ and $b \in \mathcal{S}$ then $ab \in \mathcal{S}$.*

Proof. First, observe that we have:

$$ab = \bigcurlywedge \{\neg\neg c : a \preccurlyeq b \overset{\otimes}{\rightarrow} c\} = \neg \bigcurlyvee \{\neg c : a \preccurlyeq b \overset{\otimes}{\rightarrow} c\}$$

Then one can easily show that:

$$\neg(a \Diamond b) \preccurlyeq ab$$

and therefore it suffices to show that $\neg(a \Diamond b) \in \mathcal{S}$. To that end, we use the deduction rule of the separator and show that

$$\neg(b \otimes (a \Diamond b)) \in \mathcal{S}$$

This is now easy using the previous lemma and that $a \in \mathcal{S}$, since we have:

$$a \preccurlyeq \neg(b \otimes (a \Diamond b)) \quad \text{iff} \quad a \Diamond b \preccurlyeq a \Diamond b$$ ◀

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

▶ **Proposition 106** . $(\lambda f)a \preccurlyeq \neg\neg f(a)$

We show that Hilbert's combinators $\mathbf{k}$ and $\mathbf{s}$ belong to any conjunctive separator:

▶ **Proposition 107** ($\mathbf{k}$ and $\mathbf{s}$). *We have:*

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

Proof. We only sketch these proofs are quite involved and require several auxiliary results (see the Coq files).

1. We first show $(\lambda xy.x)^{\mathcal{A}} = \bigcurlywedge_{a,b}(\neg(a \otimes b \otimes \neg b))$. Then we conclude by transitivity by showing that $\bigcurlywedge_{a,b}(\neg(a \otimes \neg b \otimes b)) \in \mathcal{S}$ and $\bigcurlywedge_{a,b} \neg((\neg b \otimes b) \otimes (b \otimes \neg b)) \in \mathcal{S}$. The latter follows from $\boldsymbol{s}_3^{\otimes}$, while the former follows from $\boldsymbol{s}_2^{\otimes}$ modulo the facts that the tensor is associative and commutative with respect to the separator.
2. We first show that the interpretation of the type of $\mathbf{s}$ belongs to $\mathcal{S}$:
$$\bigcurlywedge_{a,b,c}]((a \overset{\otimes}{\to} b \overset{\otimes}{\to} c) \overset{\otimes}{\to} (a \overset{\otimes}{\to} b) \overset{\otimes}{\to} a \overset{\otimes}{\to} c) \in \mathcal{S} \tag{$S1$}$$
We then mimic the proof that $\mathbf{s}$ and its type are the same in implicative structure, replacing the ordering relation by the entailment. We begin by showing (using distribution laws) that $\mathbf{s} \in \mathcal{S}$ can be deduced from:
$$\bigcurlywedge\{\neg(((a \otimes b) \otimes c) \otimes \neg d) : ac \preccurlyeq bc \overset{\otimes}{\to} d\} \in \mathcal{S} \tag{$S2$}$$
Then, after showing that:
$$\bigcurlywedge_{a,b}(a \overset{\otimes}{\to} (b \overset{\otimes}{\to} ab)) \in \mathcal{S} \tag{$S3$}$$
we prove that the $(S3)$ can be deduced from:
$$\bigcurlywedge_{b,c,d}((c \overset{\otimes}{\to} bc \overset{\otimes}{\to} d) \overset{\otimes}{\to} (c \overset{\otimes}{\to} bc) \overset{\otimes}{\to} c \overset{\otimes}{\to} d) \in \mathcal{S} \tag{$S4$}$$
which follows from $(S1)$.

◀

In the case where the separator is classical[26], we can prove that it contains the interpretation of all closed λ-terms:

▶ **Theorem 108** (λ-calculus). *If $\mathcal{S}$ is classical and t is a closed λ-term, then $t^{\mathcal{A}} \in \mathcal{S}$.*

Proof. By combinatorial completeness, we have the existence of a combinatorial term t_0 (*i.e.* a composition of $\mathbf{k}$ and $\mathbf{s}$) such that $t_0 \to^* t$. Since $\mathbf{k} \in \mathcal{S}$, $\mathbf{s} \in \mathcal{S}$ and $\mathcal{S}$ is closed under application, $t_0^{\mathcal{A}} \in \mathcal{S}$. For each step $t_n \to t_{n+1}$, we have $t_n^{\mathcal{A}} \preccurlyeq \neg\neg t_{n+1}^{\mathcal{A}}$, and thus $t_n^{\mathcal{A}} \in \mathcal{S}$ implies $t_{n+1}^{\mathcal{A}} \in \mathcal{S}$. We conclude by induction on the length of the reduction $t_0 \to^* t$. ◀

[26] 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 $\neg\neg t^{\mathcal{A}} \in \mathcal{S}$.

Induced Heyting algebra

As in the implicative case, the entailment relation induces a structure of (pre)-Heyting algebra, whose conjunction and disjunction are naturally given $a + b \triangleq \neg(\neg a \otimes \neg b)$ and $a \times b \triangleq a \otimes b$.

▶ **Proposition 109** (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* $a \times b \vdash_{\mathcal{S}} c$

Proof. Easy manipulation of conjunctive algebras, see the Coq proofs. ◀

Conjunctive tripos

We will need the following lemma:

▶ **Lemma 110** (Adjunction). *If $a \preccurlyeq b \overset{\otimes}{\to} c$ then $ab \preccurlyeq \neg\neg c$.*

In order to obtain a conjunctive tripos, we define:

$$\exists_{i \in I} a_i \triangleq \bigcurlyvee_{i \in I} a_i \qquad\qquad \forall_{i \in I} a_i \triangleq \neg(\bigcurlyvee_{i \in I} \neg a_i)$$

▶ **Theorem 111** (Conjunctive tripos). *Let $(\mathcal{A}, \preccurlyeq, \to, \mathcal{S})$ be a classical conjunctive algebra. The following functor (where $f : J \to I$):*

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

defines a tripos.

Proof. The proof mimics the proof in the case of implicative algebras, see Theorem 111.
We verify that $\mathcal{T}$ satisfies all the necessary conditions to be a tripos.

- The functoriality of $\mathcal{T}$ is clear.
- For each $I \in \mathbf{Set}$, the image of the corresponding diagonal morphism $\mathcal{T}(\delta_I)$ associates to any element $[(a_{ij})_{(i,j) \in I \times I}] \in \mathcal{T}(I \times I)$ the element $[(a_{ii})_{i \in I}] \in \mathcal{T}(I)$. We define

$$(=_I) \;:\; i, j \mapsto \begin{cases} \bigcurlywedge_{a \in \mathcal{A}} (a \to a) & \text{if } i = j \\ \bot \to \top & \text{if } i \neq j \end{cases}$$

and we need to prove that for all $[a] \in \mathcal{T}(I \times I)$:

$$[\top]_I \preccurlyeq_{\mathcal{S}[I]} \mathcal{T}(\delta_I)(a) \qquad \Leftrightarrow \qquad [=_I] \preccurlyeq_{S[I \times I]} [a]$$

Let then $[(a_{ij})_{i,j \in I}]$ be an element of $\mathcal{T}(I \times I)$.
From left to right, assume that $[\top]_I \preccurlyeq_{\mathcal{S}[I]} \mathcal{T}(\delta_I)(a)$, that is to say that there exists $s \in \mathcal{S}$ such that for any $i \in I$, $s \preccurlyeq \top \to a_{ii}$. We would like to reproduce the proof in the implicative case, which uses $\lambda z.z(s(\lambda x.x))$. Here, due to the double-negation induced by the application (see Section 4.4), we can only show that:

$$\lambda z.z(s(\lambda x.x)) \preccurlyeq i =_I j \to \neg\neg\neg\neg a_{ij} \qquad (\forall i, j)$$

. Indeed, if $i \neq j$, we have that:

$$\begin{array}{rll} & \lambda z.z(s(\lambda x.x)) \preccurlyeq (\top \to \bot) \to \neg\neg\neg\neg a_{ij} & \\ \Leftarrow & (\top \to \bot) \to (\top \to \bot)(s(\lambda x.x)) \preccurlyeq (\top \to \bot) \to \neg\neg\neg\neg a_{ij} & (\lambda - \text{def}) \\ \Leftarrow & (\top \to \bot)(s(\lambda x.x)) \preccurlyeq \neg\neg\neg\neg a_{ij} & (\text{variance}) \\ \Leftarrow & \top \to \bot \preccurlyeq (s(\lambda x.x)) \to \neg\neg a_{ij} & (\text{adjunction}) \end{array}$$

the last one being true by subtyping. If $i = j$, we have that:

$$\begin{array}{rll} & \lambda z.z(s(\lambda x.x)) \preccurlyeq (\curlywedge a \to a) \to \neg\neg\neg\neg a_{ii} & \\ \Leftarrow & (\curlywedge a \to a) \to (\curlywedge a \to a)(s(\lambda x.x)) \preccurlyeq (\curlywedge a \to a) \to \neg\neg\neg\neg a_{ij} & (\lambda - \text{def}) \\ \Leftarrow & (\curlywedge a \to a)(s(\lambda x.x)) \preccurlyeq \neg\neg\neg\neg a_{ij} & (\text{variance}) \\ \Leftarrow & (\curlywedge a \to a) \preccurlyeq (s(\lambda x.x)) \to \neg\neg a_{ij} & (\text{adjunction}) \\ \Leftarrow & (s(\lambda x.x)) \to (s(\lambda x.x)) \preccurlyeq (s(\lambda x.x)) \to \neg\neg a_{ij} & (\text{variance}) \\ \Leftarrow & s(\lambda x.x) \preccurlyeq \neg\neg a_{ij} & (\text{adjunction}) \\ \Leftarrow & s \preccurlyeq \lambda x.x \to a_{ij} & \end{array}$$

the last one being true by assumption. We conclude using the fact that any λ-terms with parameters in $\mathcal{S}$ belongs to $\mathcal{S}$ using a slight variant of Theorem 108.

From right to left, if there exists $s \in \mathcal{S}$ such that for any $i, j \in I$, $s \preccurlyeq i =_I j \to a_{ij}$, then in particular for all i $\curlywedge_a (a \overset{\otimes}{\to} a) \vdash a_{ii}$. We use the transitivy of $\vdash$ to show that $\top \vdash a_{ii}$ follows from $\curlywedge_a (a \overset{\otimes}{\to} a) \vdash a_{ii}$ and $\top \vdash \curlywedge_a (a \overset{\otimes}{\to} a)$. Writing Id for $\curlywedge_a (a \overset{\otimes}{\to} a)$, the latter is obtained by using the deduction rule:

$$\frac{\vdash \neg(Id \otimes (\top \otimes \neg Id)) \quad \vdash Id}{\vdash \neg(\top \otimes Id}$$

Using $\boldsymbol{s}_5^{\otimes}$ to get $\neg(Id \otimes (\top \otimes \neg Id))$ from $\neg(\neg(Id \otimes \top) \otimes \neg Id)$ which follows from $\boldsymbol{s}_2^{\otimes}$.

For each projection $\pi^1_{I \times J} : I \times J \to I$ in $\mathcal{C}$, the monotone function $\mathcal{T}(\pi^1_{I,J}) : \mathcal{T}(I) \to \mathcal{T}(I \times J)$ has both a left adjoint $(\exists J)_I$ and a right adjoint $(\forall J)_I$ which are defined by:

$$(\forall J)_I\big([(a_{ij})_{i,j \in I \times J}]\big) \triangleq \big[(\underset{j \in J}{\forall} a_{ij})_{i \in I}\big] \qquad\qquad (\exists J)_I\big([(a_{ij})_{i,j \in I \times J}]\big) \triangleq \big[(\underset{j \in J}{\exists} a_{ij})_{i \in I}\big]$$

We only give the case of $\forall$, the case for $\exists$ is easier (it corresponds to this and this Coq lemmas). We need to show that for any $[(b_{ij})_{(i,j) \in I \times J}] \in \mathcal{T}(I \times J)$ and for any $[(a_i)_{i \in I}]$, we have:

$$[(a_i)_{(i,j) \in I \times J}] \preccurlyeq_{\mathcal{S}[I \times J]} [(b_{ij})_{(i,j) \in I}] \quad \Leftrightarrow \quad [(a_i)_{i \in I}] \preccurlyeq_{\mathcal{S}[I]} \big[(\underset{j \in J}{\forall} b_{ij})_{i \in I}\big] = \big[(\neg \bigcurlyvee_{j \in J} \neg b_{ij})_{i \in I}\big]$$

Let us fix some $[a]$ and $[b]$ as above.

From left to right, assume that for all $i \in I$, $j \in J$, $a_{ij} \vdash b_i$, we want to prove that $\forall i \in I$, we have $a_i \vdash \neg \curlyvee_{j \in J} \neg b_{ij}$. We first show that for any a, b, c, the following rule is valid (it mainly amount to $\boldsymbol{s}_4^{\otimes}$):

$$\frac{a \vdash b \quad \neg(c \otimes b) \in \mathcal{S}}{\neg(c \otimes a) \in \mathcal{S}}$$

and prove instead that $\neg\neg \curlyvee_{j \in J} \neg b_{ij} \vdash \curlyvee_{j \in J} \neg b_{ij}$ and $\neg(a_i \otimes \curlyvee_{j \in J} \neg b_{ij}) \in \mathcal{S}$. The former amount to $\neg\neg a \vdash a$ while we can use commutation rule on the latter to rewrite it as: $\curlywedge_{j \in J} \neg(a_i \otimes \neg b_{ij}) \in \mathcal{S}$ which follows from the assumption.

From right to left, the processus is almost the same and relies on the fact we have for all $i \in I$, $j \in J$, $(a_i \overset{\otimes}{\to} b_i j) \in \mathcal{S}$ if and only if for all $i \in I$, $\curlywedge_j (a_i \overset{\otimes}{\to} b_i j) \in \mathcal{S}$ if and only if for all $i \in I$, $(a_i \otimes \curlyvee_j \neg b_i j) \in \mathcal{S}$. We then use the same lemma with the reverse law $a \vdash \neg\neg a$.

- These adjoints clearly satisfy the Beck-Chevaley condition as in the implicative cases.
- Finally, we define $\mathrm{Prop} \triangleq \mathcal{A}$ and verify that $\mathrm{tr} \triangleq [\mathrm{id}_{\mathcal{A}}] \in \mathcal{T}(\mathrm{Prop})$ is a generic predicate, as in the implicative case.

◀

D The duality of computation

▶ **Proposition 112.** *Let $(\mathcal{A}, \preccurlyeq, \parr, \neg)$ be a disjunctive structure. Let us define:*

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

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

Proof. We check that for all $a, a', b, b' \in \mathcal{A}$ and for all subsets $A \subseteq \mathcal{A}$, we have:

1. If $a \triangleleft a'$ then $\neg a' \triangleleft \neg a$ (Variance)
2. If $a \triangleleft a'$ and $b \triangleleft b'$ then $a \otimes b \triangleleft a' \otimes b'$. (Variance)
3. $(\curlywedge^{\otimes}_{a \in A} a) \otimes b = \curlywedge^{\otimes}_{a \in A}(a \otimes b)$ and $b \otimes (\curlywedge^{\otimes}_{a \in A} a) = \curlywedge^{\otimes}_{a \in A}(b \otimes a)$ (Distributivity)
4. $\neg(\curlyvee^{\otimes}_{a \in A} a) = \curlywedge^{\otimes}_{a \in A}(\neg a)$ (Commutation)

All the proof are trivial from the corresponding properties of disjunctive structures. ◀

▶ **Proposition 113.** *Let $(\mathcal{A}, \preccurlyeq, \otimes, \neg)$ be a conjunctive structure. Let us define:*

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

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

Proof. We check that for all $a, a', b, b' \in \mathcal{A}$ and for all subsets $A \subseteq \mathcal{A}$, we have:

1. If $a \triangleleft a'$ then $\neg a' \triangleleft \neg a$. (Variance)
2. If $a \triangleleft a'$ and $b \triangleleft b'$ then $a \parr b \triangleleft a' \parr b'$. (Variance)
3. $(\curlywedge^{\parr}_{a \in A} a) \parr b = \curlywedge^{\parr}_{a \in A}(a \parr b)$ and $a \parr (\curlywedge^{\parr}_{b \in B} b) = \curlywedge^{\parr}_{b \in B}(a \parr b)$ (Distributivity)
4. $\neg(\curlywedge^{\parr} A) = \curlyvee^{\parr}_{a \in A}(\neg a)$ (Commutation)

All the proof are trivial from the corresponding properties of conjunctive structures. ◀

▶ **Theorem 114.** *Let $(\mathcal{A}^{\otimes}, \mathcal{S}^{\otimes})$ be a conjunctive algebra, the set:*

$$\mathcal{S}^{\parr} \;\triangleq\; \neg^{-1}(\mathcal{S}^{\otimes}) = \{a \in \mathcal{A} : \neg a \in \mathcal{S}^{\otimes}\}$$

is a valid separator for the dual disjunctive structure $\mathcal{A}^{\parr}$.

▶ **Theorem 115.** *Let $(\mathcal{A}^{\parr}, \mathcal{S}^{\parr})$ be a disjunctive algebra. The set:*

$$\mathcal{S}^{\otimes} \;\triangleq\; \neg^{-1}(\mathcal{S}^{\parr}) = \{a \in \mathcal{A} : \neg a \in \mathcal{S}^{\parr}\}$$

is a classical separator for the dual conjunctive structure $\mathcal{A}^{\otimes}$.

Proof. Both proofs rely on the fact that:

$$a \vdash_{\mathcal{S}^{\otimes}} b \Leftrightarrow \neg a \vdash_{\mathcal{S}^{\parr}} \neg b \quad \text{and} \quad a \vdash_{\mathcal{S}^{\parr}} b \Leftrightarrow \neg a \vdash_{\mathcal{S}^{\otimes}} \neg b$$

In particular, to prove that the modus ponens is valid when passing from $\mathcal{A}^{\otimes}$ to $\mathcal{A}^{\parr}$, we need to show that if $a, a \to b \in \neg^{-1}(\mathcal{S}^{\otimes})$, then $b \in \neg^{-1}(\mathcal{S}^{\otimes})$ *i.e.* $\neg b \in \mathcal{S}^{\otimes}$. By hypothesis, we thus have that $\neg a \to \neg b \in \mathcal{S}^{\otimes}$, from which we deduce that $\neg(b \otimes \neg a) \in \mathcal{S}^{\otimes}$ (by internal contraposition). Using the deduction axiom (since $\neg a \in \mathcal{S}^{\otimes}$), we finally get $\neg b \in \mathcal{S}^{\otimes}$. ◀