10- Implicative algebras

In this chapter, we present Alexandre Miquel's *implicative algebras*[1], which aim at providing an algebraic framework for classical realizability. We first introduce the notion of *implicative structures* on which implicative algebras rely. Then, we will show that most of the structures we introduced in Chapter 9 (Complete Heyting/Boolean algebras, AKSs, OCAs) are particular cases of implicative structures. Next, we show how to embed both the λ_c-calculus in a manner which is adequate with its second-order type system. Finally, we introduce the notion of separators and implicative algebras, and show how they induce realizability triposes.

Most of the results in this chapter are supported by a Coq development• [122]. All along the chapter, we use the bullet to denote the statements that are formalized.

10.1 Implicative structures

10.1.1 Definition

Intuitively, *implicative structures* are tailored to represent both the formulas of second-order logic and realizers arising from Krivine's λ_c-calculus. We shall see in the sequel how they indeed allow us to define λ-terms, but let us introduce them by focusing on their logical facet. We are interested in formulas of second-order logic, that is to say of system F, which are defined by a simple grammar:

$$A, B ::= X \mid A \Rightarrow B \mid \forall X.A$$

Implicative structures are therefore defined as meet-complete lattices (for the universal quantification) with an internal binary operation satisfying the properties of the implication:

Definition• 10.1. An *implicative structure* is a complete meet-semilattice $(\mathcal{A}, \preccurlyeq)$ equipped with a binary operation $(a, b) \mapsto (a \to b)$, called the *implication* of $\mathcal{A}$, that fulfills the following axioms:

1. Implication is anti-monotonic with respect to its first operand and monotonic with respect to its second operand, in the sense that for all $a, a_0, b, b_0 \in \mathcal{A}$:

   (Variance) if $a_0 \preccurlyeq a$ and $b \preccurlyeq b_0$ then $(a \to b) \preccurlyeq (a_0 \to b_0)$

2. Arbitrary meets distribute over the second operand of implication, in the sense that for all $a \in \mathcal{A}$ and for all subsets $B \subseteq \mathcal{A}$:

   (Distributivity) $$\bigwedge_{b \in B} (a \to b) = a \to \bigwedge_{b \in B} b$$

⌟

[1]We insist on the fact that all the results presented in this chapters are his. Most of them are given in [121]. Independently, structures that are very similar to implicative structures can be found in Frédéric Ruyer's Ph.D. thesis [147] under the name of *applicative lattices*.

Remark 10.2. 1. The distributivity axiom of implicative structures should not be confused with the property of distributivity for lattices (see the definition of Boolean algebras). In general, the underlying lattice of an implicative structure does not have to be distributive.

2.• In the particular case where $B = \emptyset$, the axiom of distributivity states that $a \to \top = \top$ for all $a \in \mathcal{A}$.

⌟

10.1.2 Examples of implicative structures

10.1.2.1 Complete Heyting algebras

The first example of implicative structures is given by complete Heyting algebras. Indeed, the axioms of implicative structures are intuitionistic tautologies verified by any complete Heyting algebra:

Proposition 10.3. *If $(\mathcal{H}, \preccurlyeq, \to)$ is a complete Heyting algebra, then for all $a, a', b, b', c \in \mathcal{H}$ and for all subsets $B \subseteq \mathcal{H}$, the following holds:*

1.• *if $a \preccurlyeq a'$, then $a' \to b \preccurlyeq a \to b$;*
2.• *if $b \preccurlyeq b'$, then $a \to b \preccurlyeq a \to b'$;*
3.• $a \curlywedge c \preccurlyeq b \Leftrightarrow a \preccurlyeq c \to b$
4.• $a \to \curlywedge_{b \in B} b = \curlywedge_{b \in B} (a \to b)$.

Proof. Observe first that since $\mathcal{H}$ is complete, by definition we have $a \to b = \curlyvee \{x \in \mathcal{H} : a' \wedge x \preccurlyeq b\}$.

1. Let $a, a', b \in \mathcal{H}$ be fixed. Using this observation above for $a' \to b$, it suffices to show that $a \to b$ is an upper bound of the set $\{x \in \mathcal{H} : a' \curlywedge x \preccurlyeq b\}$. Let then $x \in \mathcal{H}$ be such that $a' \curlywedge x \preccurlyeq b$. To show that $x \preccurlyeq a \to b$, it suffices to show that $a \curlywedge x \preccurlyeq b$. This follows from the transitivity of the order: $a \curlywedge x \preccurlyeq a' \curlywedge x \preccurlyeq b$.
2. Similar to 1.
3. Let $a, b, c \in \mathcal{H}$ be fixed. The left-to-right implication is trivial from the observation above. From right to left, we show that $a \curlywedge c \preccurlyeq c \curlywedge (c \to b) \preccurlyeq b$. The first inequality follows from the monotonicity of $\curlywedge$, the second one follows from the definition of $c \to b$.
4. Let $a \in \mathcal{H}$ and $B \subseteq \mathcal{H}$ be fixed. By definition, this amounts to showing that:

$$\curlyvee \{x \in \mathcal{H} : a \curlywedge x \preccurlyeq \bigcurlywedge_{b \in B} b\} = \bigcurlywedge_{b \in B} \curlyvee \{x \in \mathcal{H} : a \curlywedge x \preccurlyeq b\}$$

which we show by anti-symmetry. To show that the term on the left hand-side term is inferior to the one on the right-hand side, it suffices to show that $\curlyvee \{x \in \mathcal{H} : a \curlywedge x \preccurlyeq \curlywedge_{b \in B} b\} \preccurlyeq a \to b$ for any $b \in B$. Let thus $x \in \mathcal{H}$ be such that $a \curlywedge x \preccurlyeq \curlywedge_{b \in B} b$, we need to show that $x \preccurlyeq a \to b$. This follows from the third item and the inequality $a \curlywedge x \preccurlyeq \curlywedge_{b \in B} b \preccurlyeq b$. The converse inequality is proved similarly.

□

We deduce that every complete Heyting algebra induces an implicative structure with the same arrow:

Proposition• 10.4. *Every complete Heyting algebra is an implicative structure.*

The converse is obviously false, since the implication of an implicative structure $\mathcal{A}$ is in general not determined by the lattice structure of $\mathcal{A}$.

10.1.2.2 Complete Boolean algebras

Since any (complete) Boolean algebra is in particular a (complete) Heyting algebra, *a fortiori* any complete Boolean algebra induces an implicative structure:

Proposition• 10.5. *If $\mathcal{B}$ is a (complete) Boolean algebra, then $\mathcal{B}$ is a (complete) Heyting algebra where the implication is defined for all $a,b \in \mathcal{B}$ by $a \to b \triangleq (\neg a) \curlyvee b$.*

Proof. Let $a,b \in \mathcal{B}$ be fixed. We show that $(\neg a) \curlyvee b$ is the supremum of $\{x \in \mathcal{B} : a \curlywedge x \preccurlyeq b\}$, *i.e.* that it belongs to this set and that it is an upper bound of the same. The first part is trivial, since the distributivity implies that $a \curlywedge (\neg a \curlyvee b) = (a \curlywedge \neg a) \curlyvee (a \curlywedge b) = a \curlywedge b \preccurlyeq b$. For the second part of the statement, let $c \in \mathcal{B}$ be such that $a \curlywedge c \preccurlyeq b$. Then we have: $c = (c \curlywedge \neg a) \curlyvee (c \curlywedge a) \preccurlyeq (c \curlywedge \neg a) \curlyvee b \preccurlyeq \neg a \curlyvee b$, which concludes the proof. □

Proposition• 10.6. *If $\mathcal{B}$ is a (complete) Boolean algebra, then $\mathcal{B}$ induces an implicative structure where the implication is defined for all $a,b \in \mathcal{B}$ by $a \to b \triangleq \neg a \curlyvee b$.*

10.1.2.3 Dummy structures

Given a complete lattice $\mathcal{L}$, there are at least two possible definitions of dummy implicative structures:

Proposition 10.7. *If $\mathcal{L}$ is a complete lattice, the following definitions give rise to implicative structures:*

1.• $a \to b \triangleq \top$ for all $a,b \in \mathcal{L}$ *2.• $a \to b \triangleq b$ for all $a,b \in \mathcal{L}$*

Proof. Trivial in both cases. □

Both definitions induce implicative structures which are meaningless from the point of view of logic. Nonetheless, they will provide us with useful counter-examples.

10.1.2.4 Ordered combinatory algebras

Any ordered combinatory algebra (see Definition 9.25) also induces an implicative structure, whose definition is related with the definition of the realizability tripos. Indeed, remember that given an OCA $\mathcal{A}$ and a set X, the ordering on predicates of $\mathcal{P}(\mathcal{A})^X$ is defined by:

$$\varphi \vdash_X \psi \quad \triangleq \quad \exists r \in \mathcal{A}.\forall x \in X.\forall a \in \mathcal{A}.(a \in \varphi(x) \Rightarrow ra \in \psi(x))$$

where r is broadly a *realizer* of $\forall x \in X.\varphi(x) \Rightarrow \psi(x)$. Similarly, we can define an implication on the complete lattice $\mathcal{P}(\mathcal{A})$ which give rise to an implicative structure:

Proposition 10.8. *If $\mathcal{A}$ is an ordered combinatory algebra, then the complete lattice $\mathcal{P}(\mathcal{A})$ equipped with the implication:*

$$A \to B \quad \triangleq \quad \{r \in \mathcal{A} : \forall a \in A.ra \in B\} \qquad (\forall A,B \subseteq \mathcal{A})$$

is an implicative structure

Proof. Both conditions (variance and distributivity) are trivial from the definition. □

In particular, the powerset of any ${}^{\mathcal{I}}$OCA or ${}^{\mathcal{K}}$OCA induces an implicative structure with the same construction.

10.1.2.5 Implicative structure of classical realizability

Our final example of implicative structure—which is the main motivation of this work—is given by classical realizability. As we saw in Chapter 9, the construction of classical realizability models, whether it be from Krivine's realizability algebras [98, 99, 100] in a set-theoretic like fashion or in Streicher's AKS [151], takes place in a structure of the form $(\mathbf{\Lambda}, \mathbf{\Pi}, \cdot, \perp\!\!\!\perp)$ where:

- $\mathbf{\Lambda}$ is the set of realizers;
- $\mathbf{\Pi}$ is the set of stacks (or opponents);
- $(\cdot) : \mathbf{\Lambda} \times \mathbf{\Pi} \to \mathbf{\Pi}$ is a binary operation for pushing a realizer onto a stack;
- $\perp\!\!\!\perp \subseteq \mathbf{\Lambda} \times \mathbf{\Pi}$ is the pole.

Given such a quadruple, we can define:

- $\mathcal{A} \triangleq \mathcal{P}(\mathbf{\Pi})$;
- $a \preccurlyeq b \triangleq a \supseteq b$ (for all $a, b \in \mathcal{A}$)
- $a \to b \triangleq a^{\perp\!\!\!\perp} \cdot b = \{t \cdot \pi : t \in a^{\perp\!\!\!\perp}, \pi \in b\}$ (for all $a, b \in \mathcal{A}$)

where as usual $a^{\perp\!\!\!\perp}$ is $\{t \in \mathbf{\Lambda} : \forall \pi \in a, (t, \pi) \in \perp\!\!\!\perp\} \in \mathcal{P}(\mathbf{\Lambda})$, the orthogonal set of $a \in \mathcal{P}(\mathbf{\Pi})$ with respect to the pole $\perp\!\!\!\perp$. Here again, it is easy to verify that this defines an implicative structure.

Proposition 10.9. *The triple* $(\mathcal{A}, \preccurlyeq, \to)$ *is an implicative structure.*

Proof. The proof is again trivial. Variance conditions correspond to the usual monotonicity of truth and falsity values, while the distributivity follows directly by unfolding the definitions. □

Remark 10.10.

1. Actually, in this particular case the implication satisfies two additional laws:

$$(\bigwedge_{a \in A} a) \to b = \bigvee_{a \in A} (a \to b) \qquad \text{and} \qquad a \to (\bigvee_{b \in B} b) = \bigvee_{b \in B} (a \to b)$$

   for all $a, b \in \mathcal{A}, A, B \subseteq \mathcal{A}$. These extra properties also follow directly from the definition, however, they are almost never used in classical realizability.

2. Unlike Streicher's definition of the OCA used for the construction of Krivine's tripos (see Proposition 9.29), where $\mathcal{A}$ is defined as $\mathcal{P}_{\perp\!\!\!\perp}(\mathbf{\Pi})$, we consider $\mathcal{A}$ to be all the sets of $\mathcal{P}(\mathbf{\Pi})$. In this sense, we are in line with Krivine's usual definitions, where falsity values are not necessarily closed by double orthogonal. We will see that this presents an advantage over Streicher's OCAs (and thus Ferrer *et al.* ${}^{\mathcal{I}}$OCAs and ${}^{\mathcal{K}}$OCAs), namely that we will have the full adjunction:

$$a \leq b \to c \quad \Leftrightarrow \quad ab \leq c \qquad (\forall a, b, c \in \mathcal{A})$$

   On the contrary, in ${}^{\mathcal{I}}$OCAs and ${}^{\mathcal{K}}$OCAs an adjunctor $\boldsymbol{e}$ is required for the right-to-left implication, which becomes:

$$ab \leq c \quad \Rightarrow \quad \boldsymbol{e}a \leq b \to c \qquad (\forall a, b, c \in \mathcal{A})$$

⌟

10.2 Interpreting the λ-calculus

10.2.1 Interpretation of λ-terms

We motivated the definition of implicative structures with the aim of obtaining a common framework for the interpretation both of types and programs. We shall now see how λ-terms can indeed be defined in implicative structures.

From now on, let $\mathcal{A} = (\mathcal{A}, \preccurlyeq, \to)$ denotes an arbitrary implicative structure.

Definition• 10.11 (Application)**.** Given two elements $a, b \in \mathcal{A}$, we call the *application* of a to b and write ab the element of $\mathcal{A}$ that is defined by

$$ab \triangleq \bigwedge\{c \in \mathcal{A} : a \preccurlyeq (b \to c)\}.$$

As usual, we write $ab_1b_2 \cdots b_n$ for $((ab_1)\, b_2) \cdots b_n$ (for all $a, b_1, b_2, \ldots, b_n \in \mathcal{A}$). ⌟

If we think of the order relation $a \preccurlyeq b$ as "a is more precise than b", the above definition actually defines the application ab as the meet of all the elements c such that $b \to c$ is an approximation of a. This definition fulfills the usual properties of the λ-calculus:

Proposition 10.12 (Properties of application)**.** *For all $a, a', b, b', c \in \mathcal{A}$:*

1.• *If $a \preccurlyeq a'$ and $b \preccurlyeq b'$, then $ab \preccurlyeq a'b'$* *(Monotonicity)*
2.• $(a \to b)a \preccurlyeq b$ *(β-reduction)*
3.• $a \preccurlyeq (b \to ab)$ *(η-expansion)*
4.• $ab = \min\{c \in A : a \preccurlyeq (b \to c)\}$ *(Minimum)*
5.• $ab \preccurlyeq c \quad \Leftrightarrow \quad a \preccurlyeq (b \to c)$ *(Adjunction)*

Proof. For all $a, b \in \mathcal{A}$, let us write $\mathsf{App}_{a,b} = \{c \in \mathcal{A} : a \preccurlyeq (b \to c)\}$, so that $ab = \bigwedge \mathsf{App}_{a,b}$.

1. We prove the monotonicity w.r.t. to the left operand a, the monotonicity w.r.t. to the right one is very similar. Let a, a', b be elements of $\mathcal{A}$, and assume that $a \preccurlyeq a'$. We want to prove:

   $$\bigwedge \mathsf{App}_{a,b} \preccurlyeq \bigwedge \mathsf{App}_{a',b}$$

   It is thus enough to show that $\mathsf{App}_{a,b} \subseteq \mathsf{App}_{a',b}$, which is trivial.
2. For any $a, b \in \mathcal{A}$, we have by definition that $b \in \mathsf{App}_{a \to b, a}$, thus $\bigwedge \mathsf{App}_{a \to b, a} \preccurlyeq b$.
3. Let a, b be elements of $\mathcal{A}$. By distributivity, we have $b \to \bigwedge \mathsf{App}_{a,b} = \bigwedge\{b \to c : c \in \mathsf{App}_{a,b}\}$. To prove the desired inequality, it is enough to show that for any $c \in \mathsf{App}_{a,b}$, we have $a \preccurlyeq b \to c$, which is a tautology.
4. Follows from 3.
5. From left to right, we prove that $a \preccurlyeq (b \to ab) \preccurlyeq (b \to c)$ using 3 and the covariance of the implication. From right to left, it is clear that if $c \in \mathsf{App}_{a,b}$, then $ab = \bigwedge \mathsf{App}_{a,b} \preccurlyeq c$.

□

Remark• 10.13 (Galois connection)**.** The adjunction $ab \preccurlyeq c \Leftrightarrow a \preccurlyeq (b \to c)$ expresses the existence of a family of Galois connections $f_b \dashv g_b$ indexed by all $b \in \mathcal{A}$, where the left and right adjoints $f_b, g_b : \mathcal{A} \to \mathcal{A}$ are defined by:

$$f_b : a \mapsto ab \qquad \text{and} \qquad g_b : c \mapsto (b \to c) \qquad \text{(for all } a, b, c \in \mathcal{A})$$

Recall that in a Galois connection, the left adjoint is fully determined by the right one (and vice-versa, see Proposition 9.20). In the particular case of a complete Heyting algebra $(\mathcal{H}, \preccurlyeq, \to)$, this implies that the application is characterized by $ab = a \curlywedge b$ for all $a, b \in \mathcal{H}$. Indeed, in any Heyting algebra, the adjunction $a \curlywedge b \preccurlyeq c \Leftrightarrow a \preccurlyeq (b \to c)$ holds for all $a, b, c \in \mathcal{H}$ (Proposition 10.3), by uniqueness of the left adjoint, ab and $a \curlywedge b$ are thus equal. ⌟

Definition• 10.14 (Abstraction)**.** Given a function $f : \mathcal{A} \to \mathcal{A}$, we call *abstraction* of f and write λf the element of $\mathcal{A}$ defined by:

$$\lambda f \triangleq \bigwedge_{a \in \mathcal{A}} (a \to f(a))$$

⌟

Once again, if we think of the order relation $a \preccurlyeq b$ as "a is more precise than b", the meet of the elements of a set S is an element containing the union of all the informations given by the elements of S. With this in mind, the above definition sets λf as the union of all the step functions $a \to f(a)$. This definition, together with the definition of the application, fulfills again properties expected from the λ-calculus:

Proposition 10.15 (Properties of the abstraction)**.** *The following holds for any $f, g : \mathcal{A} \to \mathcal{A}$:*

1.• *If for all $a \in \mathcal{A}$, $f(a) \preccurlyeq g(a)$, then $\lambda f \preccurlyeq \lambda g$.* *(Monotonicity)*
2.• *For all $a \in \mathcal{A}$, $(\lambda f)a \preccurlyeq f(a)$.* *($\beta$-reduction)*
3.• *For all $a \in \mathcal{A}$, $a \preccurlyeq \lambda(x \mapsto ax)$.* *($\eta$-expansion)*

Proof. Let $a \in \mathcal{A}$ be fixed.

1. By hypothesis, we have for all $b \in \mathcal{A}$ that $\curlywedge_{a \in \mathcal{A}}(a \to f(a)) \preccurlyeq b \to f(b) \preccurlyeq b \to g(b)$. We can thus conclude that $\lambda f = \curlywedge_{a \in \mathcal{A}}(a \to f(a)) \preccurlyeq \curlywedge_{a \in \mathcal{A}}(a \to g(a)) = \lambda g$.
2. By definition of the application, in order to show that $(\lambda f)a \preccurlyeq f(a)$ it is enough to prove the inequality $\lambda f \preccurlyeq a \to f(a)$, which is obvious.
3. By definition of the abstraction, to show that $a \preccurlyeq \lambda(x \mapsto ax)$ it is enough to show that for any $x \in A$ we have $a \preccurlyeq x \to ax$. By distributivity, we have:

$$x \to ax = x \to \bigwedge \{b \in \mathcal{A} : a \preccurlyeq (x \to b)\} = \bigwedge_{x, b \in \mathcal{A}} \{x \to b : a \preccurlyeq (x \to b)\}$$

   We conclude by proving that a is a lower bound of the set on the right hand-side, which is a tautology.

□

We call a *λ-term with parameters (in $\mathcal{A}$)* any term defined from the following grammar:

$$t, u ::= x \mid a \mid \lambda x.t \mid tu$$

where x is a variable and a is an element of $\mathcal{A}$. We can thus associate to each closed λ-term with parameters t an element $t^{\mathcal{A}}$ of $\mathcal{A}$, defined by induction on the size of t as follows:

$$\begin{array}{rcll} a^{\mathcal{A}} & \triangleq & a & (\text{if } a \in \mathcal{A}) \\ (tu)^{\mathcal{A}} & \triangleq & (t^{\mathcal{A}})u^{\mathcal{A}} & \\ (\lambda x.t)^{\mathcal{A}} & \triangleq & \lambda(a \mapsto (t[a/x])^{\mathcal{A}}) & \end{array}$$

Thanks to the properties of the application and of the abstraction in implicative structures that we proved, we can check that the embedding of λ-term is sound with respect to the β-reduction and the η-expansion:

$$\frac{(x:a)\in\Gamma}{\Gamma\vdash x:a}\ (\mathrm{Ax}) \qquad \frac{}{\Gamma\vdash a:a}\ (\mathcal{A}) \qquad \frac{FV(t)\subseteq \mathrm{dom}(\Gamma)}{\Gamma\vdash t:\top}\ (\top)$$

$$\frac{\Gamma\vdash t:a \quad a\preccurlyeq a'}{\Gamma\vdash t:a'}\ (\preccurlyeq) \qquad \frac{\Gamma\vdash t:a \quad \Gamma'\preccurlyeq\Gamma}{\Gamma'\vdash t:a}\ (w) \qquad \frac{\Gamma,x:a\vdash t:b}{\Gamma\vdash \lambda x.t:a\to b}\ (\lambda)$$

$$\frac{\Gamma\vdash t:a\to b \quad \Gamma\vdash u:a}{\Gamma\vdash tu:b}\ (@) \qquad \frac{\Gamma\vdash t:a_i \quad \text{for all } i\in I}{\Gamma\vdash t:\curlywedge_{i\in I}a_i}\ (\curlywedge)$$

Figure 10.1: Semantic typing rules

Lemma 10.16. *The substitution of variable by parameter is monotonic, that is to say: for each λ-term t with free variables $x_1,\dots,x_n$, and for all parameters $a_1,b_1,\dots,a_n,b_n$, if $a_i\preccurlyeq b_1$ for all $i\le n$, then:*

$$(t[a_1/x_1,\dots,a_n/x_n])^{\mathcal{A}}\preccurlyeq(t[b_1/x_1,\dots,b_n/x_n])^{\mathcal{A}}$$

Proof. By induction on the structure of t, using Propositions 10.12 and 10.15. □

Proposition 10.17. *For all closed λ-terms t and u with parameters in $\mathcal{A}$, the following holds:*

1. *If $t\to_\beta u$, then $t^{\mathcal{A}}\preccurlyeq u^{\mathcal{A}}$.*
2. *If $t\to_\eta u$, then $u^{\mathcal{A}}\preccurlyeq t^{\mathcal{A}}$.*

Proof. Straightforward from Proposition 10.15 and Lemma 10.16. □

Again, if we think of the order relation $a\preccurlyeq b$ as "a is more precise than b", it makes sense that the β-reduction $t\to_\beta u$ is reflected in the ordering $t^{\mathcal{A}}\preccurlyeq u^{\mathcal{A}}$: the result of a computation contains indeed less information than the computation itself[2].

10.2.2 Adequacy

We now dispose of a structure in which we can interpret types and λ-terms. We saw that the interpretation of terms was intuitively sound with respect to the β-reduction. We shall now prove that the typing rules of System F are adequate with respect to the interpretation of terms, that is to say that if t is a closed λ-terms of type T, then $t^{\mathcal{A}}\preccurlyeq T^{\mathcal{A}}$. The last statement can again be understood as the fact that a term (*i.e.* a computation) carries more information than its type, just like a realizer of a formula is more informative about the formula than the formula itself.

10.2.2.1 Semantic typing rules

To this aim, we start by defining a semantic type system, that is a set of inference rules where terms are typed with elements of $\mathcal{A}$. Typing judgments are thus of the shape $\Gamma\vdash t:a$ where:

- t a λ-term with parameters;
- a is an element of the implicative structure $\mathcal{A}$;
- Γ is a finite list of the shape $\Gamma\equiv x_1:a_1,\dots,x_n:a_n$, where the x_i are variables and the a_i are elements of $\mathcal{A}$.

[2]For instance, 0 contains less information than $15-(3\times5)$ or than $\mathbb{1}_{\mathbb{Q}}(\sqrt{(2)})$.

Since elements of $\mathcal{A}$ are also their own realizers, we can also identify typing contexts with substitutions whose values are in $\mathcal{A}$. The ordering relation naturally extends to typing contexts: we write $\Gamma' \preccurlyeq \Gamma$ when for every binding $(x : a) \in \Gamma$, there exists a binding $(x : a') \in \Gamma'$ such that $a' \preccurlyeq a$. In other words, the relation $\Gamma' \preccurlyeq \Gamma$ means that $\mathrm{dom}(\Gamma) \subseteq \mathrm{dom}(\Gamma')$ and that Γ' restricted to $\mathrm{dom}(\Gamma)$ is lower than Γ component-wise.

Using the notation $t[\Gamma]$ to denote the term t under the substitution Γ, we can finally define the sequents $\Gamma \vdash t : a$ as shorthands for:

$$\Gamma \vdash t : a \quad \triangleq \quad FV(t) \subseteq \mathrm{dom}(\Gamma) \ \wedge \ (t[\Gamma])^{\mathcal{A}} \preccurlyeq a$$

We can now prove that:

Proposition 10.18 (Semantic typing)**.** *The typing rules in Figure 10.1 are sound,* i.e. *for each inference rule, we can deduce the conclusion from its hypotheses.*

Proof. Simple proof by case analysis.

- **Cases** (Ax),($\mathcal{A}$),($\top$). Obvious from the definition.

- **Case** ($\preccurlyeq$). Direct by transitivity of the order: if $(t[\Gamma])^{\mathcal{A}} \preccurlyeq a$ and $a \preccurlyeq a'$ then $(t[\Gamma])^{\mathcal{A}} \preccurlyeq a'$.

- **Case** (w). Follows from the definition of $\Gamma' \preccurlyeq \Gamma$ and the monotonicity of the substitution(Lemma 10.16).

- **Case** (λ). Assume that t is a term, that a, b are elements of $\mathcal{A}$ and that Γ is a context such that $FV(t) \subseteq \mathrm{dom}(\Gamma) \cup \{a\}$ and $(t[\Gamma, x : a])^{\mathcal{A}} \preccurlyeq b$. Then we have:

$$(\lambda x.t[\Gamma])^{\mathcal{A}} = \bigwedge_{c \in \mathcal{A}} (c \to (t[\Gamma, x : c])^{\mathcal{A}}) \preccurlyeq a \to (t[\Gamma, x : a])^{\mathcal{A}} \preccurlyeq a \to b$$

- **Case** (@). Assume that t, u are terms, that a, b are elements of $\mathcal{A}$, and that Γ is a context such that:

$$FV(t), FV(u) \subseteq \mathrm{dom}(\Gamma) \qquad (t[\Gamma])^{\mathcal{A}} \preccurlyeq a \to b \qquad (u[\Gamma])^{\mathcal{A}} \preccurlyeq u$$

Then by definition and adjunction, we have:

$$(tu[\Gamma])^{\mathcal{A}} = (t[\Gamma])^{\mathcal{A}}(u[\Gamma])^{\mathcal{A}} \qquad \text{and} \qquad (t[\Gamma])^{\mathcal{A}}(u[\Gamma])^{\mathcal{A}} \preccurlyeq b \Leftrightarrow (t[\Gamma])^{\mathcal{A}} \preccurlyeq (u[\Gamma])^{\mathcal{A}} \to b$$

We conclude by anti-monotonicity of the implication:

$$(t[\Gamma])^{\mathcal{A}} \preccurlyeq a \to b \preccurlyeq (u[\Gamma])^{\mathcal{A}} \to b$$

- **Case** ($\wedge$). This case is obvious since the meet is the greatest lower bound. □

This finally formalizes the intuition that $t \preccurlyeq a$ could be read as "t realizes a". Indeed, if t is a closed λ-term, and A a formula of system F, the adequacy lemma (Proposition 3.14) of Krivine classical realizability gives us that $t \Vdash A$, while the previous corollary somewhat gives us $t^{\mathcal{A}} \preccurlyeq A^{\mathcal{A}}$. Nonetheless, to justify formally such a statement, we should define an embedding of formulas and to prove the adequacy of the translations of terms and types with respect to the typing rules of System F.

$$\frac{(x:A)\in\Gamma}{\Gamma\vdash x:A}\ (\text{Ax}) \qquad \frac{\Gamma,x:A\vdash t:B}{\Gamma\vdash\lambda x\,.\,t:A\to B}\ (\to_I) \qquad \frac{\Gamma\vdash t:A\to B \qquad \Gamma\vdash t:A}{\Gamma\vdash tu:B}\ (\to_E)$$

$$\frac{\Gamma\vdash t:A \quad X\notin FV(\Gamma)}{\Gamma\vdash t:\forall X.A}\ (\forall_I) \qquad \frac{\Gamma\vdash t:\forall X.A}{\Gamma\vdash t:A\{X:=B\}}\ (\forall_E) \qquad \frac{}{\Gamma\vdash \boldsymbol{cc}:((A\to B)\to A)\to A}\ (\boldsymbol{cc})$$

Figure 10.2: Type system• for the λ_c-calculus

10.2.2.2 Adequacy of the interpretation

For the formalization of the former result, we chose a slightly different approach that we shall now sketch. First, we extend the usual formulas of System F by defining second-order formulas with parameters as:

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

We can then embed closed formulas with parameters into the implicative structure $\mathcal{A}$. The embedding is trivially defined by:

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

We define a type system for the λ_c-calculus with parameters[3] (that is λ-terms with parameter plus an instruction $\boldsymbol{cc}$). Typing contexts• are defined as usual by finite lists of hypotheses of the shape $(x:A)$ where x is a variable and A a formula with parameters. The inference rules, given in Figure 10.2, are the same as in System F (with the extended syntaxes of terms and formulas with parameters), plus the additional rules for *cc*.

In order to prove the adequacy of the type system with respect to the embedding, we define substitutions•, which we write σ, as functions mapping variables (of terms and types) to element of $\mathcal{A}$:

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

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

Theorem• 10.19. *The typing rules of Figure 10.2 are adequate with respect to the interpretation of terms and formulas: if t is a λ_c-term with parameters, A a formula with parameters and Γ a typing context such that $\Gamma\vdash t:A$ then for all substitutions $\sigma\Vdash\Gamma$, we have $(t[\sigma])^{\mathcal{A}}\preccurlyeq(A[\sigma])^{\mathcal{A}}$.*

Proof. The proof resembles the usual proof of adequacy in classical realizability, and most of the cases are very similar to cases of Proposition 10.18. The additional case for the instruction $\boldsymbol{cc}$ is trivial since we define $\boldsymbol{cc}^{\mathcal{A}}\triangleq\bigwedge_{a,b\in\mathcal{A}}(((a\to b)\to a)\to a) = (\forall XY.(((X\Rightarrow Y)\Rightarrow X)\Rightarrow X))^{\mathcal{A}}$ (we shall come back later to this definition). □

In the particular case where t is a closed term typed by A in the empty context, we obtain that $t^{\mathcal{A}}\preccurlyeq A^{\mathcal{A}}$. This result will be fundamental in the next section.

Corollary• 10.20. *For all λ-terms t, if $\vdash t:A$, then $t^{\mathcal{A}}\preccurlyeq A^{\mathcal{A}}$.*

[3] In practice, we use Charguéraud's locally nameless representation [23] for terms and formulas. Without giving too much details, we actually define pre-terms• and pre-types• which allow both for names (for free variables) and De Bruijn indices (for bounded variables). Terms• and types• are then defined as pre-terms and pre-types without free De Bruijn indices. Such a representation is particularly convenient to prevent from name clashes to arise.

10.2.3 Combinators

The previous results indicates that any closed λ-terms is, through the interpretation, lower than the interpretation of its principal type. We give here some examples of closed λ-terms which are in fact equal to their principal types through the interpretation in $\mathcal{A}$. Let us now consider the following combinators:

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

It is well-known that these combinators can be given the following polymorphic types:

$$\begin{array}{rcl} \mathbf{I} & : & \forall X.X \Rightarrow X \\ \mathbf{K} & : & \forall XY.X \Rightarrow Y \Rightarrow X \\ \mathbf{S} & : & \forall XYZ.(X \Rightarrow Y \Rightarrow Z) \Rightarrow (X \Rightarrow Y) \Rightarrow X \Rightarrow Z \\ \mathbf{W} & : & \forall XY.(X \Rightarrow X \Rightarrow Y) \Rightarrow X \Rightarrow Y \end{array}$$

Through the interpretation these combinators are identified with their types:

Proposition 10.21. *The following equalities hold in any implicative structure $\mathcal{A}$:*

1. $\boldsymbol{\imath}^{\mathcal{A}} = \bigwedge_{a \in \mathcal{A}} (a \to a)$
2. $\boldsymbol{\kappa}^{\mathcal{A}} = \bigwedge_{a,b \in \mathcal{A}} (a \Rightarrow b \Rightarrow a)$
3. $\boldsymbol{s}^{\mathcal{A}} = \bigwedge_{a,b,c \in \mathcal{A}} ((a \to b \to c) \to (a \to b) \to a \to c)$
4. $\boldsymbol{w}^{\mathcal{A}} = \bigwedge_{a,b,c \in \mathcal{A}} ((a \to a \to b) \to a \to b)$

Proof. The inequality from left to right are consequences of the adequacy.

1. By definition, $\mathbf{I}^{\mathcal{A}} = (\lambda x.x)^{\mathcal{A}} = \bigwedge_{a \in \mathcal{A}} (a \to a)$
2. By definition, $\mathbf{K}^{\mathcal{A}} = (\lambda xy.x)^{\mathcal{A}} = \bigwedge_{a \in \mathcal{A}} (a \to (\lambda y.a)^{\mathcal{A}}) = \bigwedge_{a \in \mathcal{A}} (a \to (\bigwedge_{b \in \mathcal{A}} (b \to a))$. We obtain the desired equality by distributivity.
3. By definition, $\mathbf{S}^{\mathcal{A}} = (\lambda xyz.xy(zy))^{\mathcal{A}} = \bigwedge_{x,y,z \in \mathcal{A}} (x \to y \to z \to xz(yz))$. We thus need to show that for any $x, y, z \in \mathcal{A}$, we have:

$$\bigwedge_{a,b,c \in \mathcal{A}} ((a \to b \to c) \to (a \to b) \to a \to c) \preccurlyeq x \to y \to z \to xz(yz)$$

   We use the transitivity to show that (the other inequality is trivial):

$$\bigwedge_{c \in \mathcal{A}} ((z \to yz \to c) \to (z \to yz) \to z \to c) \preccurlyeq x \to y \to z \to xz(yz) = \bigwedge_{c \in \mathcal{A}:\ xz \preccurlyeq yz \to c} (x \to y \to z \to c)$$

   where we obtain the equality by unfolding the definition of the application and by using the distributivity. We conclude by showing that for any $c \in \mathcal{A}$ such that $xz \preccurlyeq yz \to c$, we have:

$$(z \to yz \to c) \to (z \to yz) \to z \to c \preccurlyeq x \to y \to z \to c$$

   This follows from the monotony of the arrow, using the adjunction of the implication. For instance, we have:

$$x \preccurlyeq (z \to yz \to c) \quad \Leftrightarrow \quad xz \preccurlyeq yz \to c$$

4. The case for $\mathbf{W}$ is similar.

□

Finally, in the spirit of the previous equality, we define the interpretation of $\boldsymbol{cc}$ by the interpretation of its principal type, that is:

$$\boldsymbol{cc}^{\mathcal{A}} \triangleq \mathbf{CC} = \bigwedge_{a,b} (((a \to b) \to a) \to a)$$

Remark 10.22. It is not always the case that a term is equal to its principal type. Consider for instance a dummy implicative structure $\mathcal{A}$ where $a \to b = \top$ for all elements $a, b \in \mathcal{A}$. Suppose in addition that $\mathcal{A}$ has at least two distinct elements, so that $\bot \neq \top$. Then the following holds:

1.• For any $a, b \in \mathcal{A}$, we have $ab = \bigwedge\{c : a \preccurlyeq b \to c\} = \bigwedge \mathcal{A} = \bot$.

2.• For any $f : \mathcal{A} \to \mathcal{A}$, we have $\lambda f = \bigwedge_{a \in \mathcal{A}}(a \to f(a)) = \bigwedge_{a \in \mathcal{A}} \top = \top$.

3.• $\mathbf{II} : \forall X.X \to X$, yet $(\mathbf{II})^{\mathcal{A}} = \bot \neq \top = (\forall X.X \to X)^{\mathcal{A}}$.

4.• $\mathbf{I}^{\mathcal{A}} = \top \neq \bot = (\mathbf{SKK})^{\mathcal{A}}$.

⌟

10.2.4 The problem of consistency

The last remark shows us that not all implicative structures are suitable for interpreting intuitionistic or classical logic. We thus need to introduce a criterion of consistency:

Definition 10.23 (Consistency)**.** We say that an implicative structure is:

- *intuitionistically consistent* if $t^{\mathcal{A}} \neq \bot$ for all closed λ-terms;
- *classically consistent* if $t^{\mathcal{A}} \neq \bot$ for all closed λ_c-terms.

⌟

We verify that non trivial complete Heyting algebras are consistent as implicative algebras. To this aim, we first show that:

Proposition 10.24. *In any complete Heyting algebra $\mathcal{A}$, all closed pure λ-terms t arei nterpreted as the maximal element: $t^{\mathcal{A}} = \top$.*

Proof. Remember from Remark 10.13 that the application in the associated implicative structure is characterized by $ab = a \curlywedge b$ for all $a, b \in \mathcal{H}$. We prove a more general proposition, namely that for any closed λ-term t with parameters $a_1, \dots, a_n \in \mathcal{A}$, we have:

$$t^{\mathcal{A}} \succcurlyeq a_1 \curlywedge \dots \curlywedge a_n$$

In the particular case where t is a pure λ-term (*i.e.* without any parameter), it indeed implies that $t^{\mathcal{A}} = \top$. We proceed by induction on t. The cases for the application and parameters are trivial, for the abstraction we have:

$$(\lambda x.t)^{\mathcal{A}} = \bigwedge_{a \in \mathcal{A}} (a \to (t[a/x])^{\mathcal{A}}) \succcurlyeq \bigwedge_{a \in \mathcal{A}} (a \to a \curlywedge a_1 \curlywedge \dots \curlywedge a_n)$$

We conclude by showing that for any a, we have:

$$a_1 \curlywedge \dots \curlywedge a_n \preccurlyeq a \to a \curlywedge a_1 \curlywedge \dots \curlywedge a_n$$

which follows by adjunction. □

The proposition above enforces the observation (see Example 9.32) that Heyting algebras and Boolean algebras provide us with an interpretation of logic that is degenerated with respect to the computation. In other words, all proofs collapse to the maximal element $\top$. Nonetheless, this ensures that any non-degenerated Heyting algebra induces an intuitionistically consistent implicative structure:

Proposition 10.25. *Every non-degenerated Heyting algebras gives rise to an intuitionistically consistent implicative structure.*

We shall now relate the previous definition to the usual definition of consistency in classical realizability. Recall that any abstract Krivine structures $\mathcal{K} = (\Lambda, \Pi, \mathsf{app}, \mathsf{push}, \mathsf{k}_{\_}, \boldsymbol{k}, \boldsymbol{s}, \boldsymbol{cc}, \mathbf{PL}, \perp\!\!\!\perp)$ induces an implicative structure $(\mathcal{A}, \preccurlyeq, \to)$ where $\mathcal{A} = \mathcal{P}(\Pi)$, $a \preccurlyeq b \Leftrightarrow a \supseteq b$ and $a \to b = a^{\perp\!\!\!\perp} \cdot b$. Remember that a realizability model is said to be consistent when there is no proof-like term realizing $\bot$. Rephrased in terms of abstract Krivine structures, a falsity value $a \in \mathcal{P}(\Pi)$ is said to be realized by $t \in \mathbf{PL}$, which we write $t \Vdash a$, if $t \in a^{\perp\!\!\!\perp}$. The consistency can then be expressed by this simple criterion:

$$\mathcal{K} \text{ is consistent} \quad \text{if and only if} \quad \{\bot\}^{\perp\!\!\!\perp} \cap \mathbf{PL} = \Pi^{\perp\!\!\!\perp} \cap \mathbf{PL} = \emptyset$$

We thus need to check that this criterion of consistency for the AKS implies the consistency of the induced implicative algebra, *i.e.* that if t is a closed λ_c-term, then $t^{\mathcal{A}} \neq \bot$. By definition of the implicative algebra $\mathcal{A}$ induced the AKS, we have that $t^{\mathcal{A}} \in \mathcal{A} = \mathcal{P}(\Pi)$. Therefore, $t^{\mathcal{A}}$ is a falsity value from the point of view of the AKS. To ensure that it is not equal to $\bot$ (*i.e.* Π), it is enough to find a realizer of $t^{\mathcal{A}}$ in the AKS. The consistency of the AKS precisely states that $\bot$ does not have any realizer.

Our strategy to find a realizer for $t^{\mathcal{A}}$ in the AKS is to use t itself. First, we reduce the problem to the set of terms that are identifiable with the combinatory terms of the AKS. We call a *combinatory term* any term that is obtained by combination of the previous combinators. To each combinatory term t we associate a term t^{Λ} in Λ, whose definition by induction is trivial:

$$\mathbf{k}^{\Lambda} \triangleq \boldsymbol{k} \qquad \mathbf{s}^{\Lambda} \triangleq \boldsymbol{s} \qquad \mathbf{cc}^{\Lambda} \triangleq \boldsymbol{cc} \qquad (tu)^{\Lambda} \triangleq \mathsf{app}(t^{\Lambda}, u^{\Lambda})$$

Since the set $\mathbf{PL}$ is closed under application, for any combinatory term t, its interpretation t^{Λ} is in $\mathbf{PL}$. The combinatory completeness of $(\mathbf{k}, \mathbf{s}, \mathbf{cc})$ with respect to closed λ_c-terms ensures us that there exists a combinatory term t_0 (viewed as a λ-term) such that $t_0 \to_\beta t$. By Proposition 10.17, we thus have $t_0^{\mathcal{A}} \preccurlyeq t^{\mathcal{A}}$. It is thus enough to show that $t_0^{\mathcal{A}} \neq \bot$: we reduced the original problem for closed λ_c-terms to combinatory terms.

It thus only remains to show that for any combinatory term t_0, its interpretation $t_0^{\mathcal{A}}$ is not $\bot$. For the reason detailed above, it is sufficient to prove that $t_0^{\mathcal{A}}$ is realized. We prove that $t_0^{\mathcal{A}}$ is in fact realized by t_0^{Λ}:

Lemma 10.26. *For any combinatory term t, t^{Λ} realizes $t^{\mathcal{A}}$*, i.e. $t^{\Lambda} \Vdash t^{\mathcal{A}}$

Proof. We proceed by induction on the structure of t, by combining usual results of classical realizability and properties of the implicative structures:

- For the three combinators $\mathbf{k}, \mathbf{s}, \mathbf{cc}$, we have that their interpretations in $\mathcal{A}$ are equal to their principal types (see Proposition 10.21), which their associated combinators in the AKS realize. For instance, $\mathbf{k}^{\mathcal{A}} = \bigwedge_{a,b \in \mathcal{A}}(a \to b \to a)$ and $\mathbf{k}^{\Lambda} = \boldsymbol{k} \Vdash \|\forall AB.A \to B \to A\|$. By definition of the implicative structures, we have $\bigwedge_{a,b \in \mathcal{A}}(a \to b \to a) = \|\forall AB.A \to B \to A\|$. Thus $\mathbf{k}^{\Lambda} \Vdash \mathbf{k}^{\mathcal{A}}$.
- If $t = t_1 t_2$, we have by induction hypothesis $t_1^{\Lambda} \Vdash t_1^{\mathcal{A}}$. By η-expansion (Proposition 10.17), we get that $t_1^{\mathcal{A}} \preccurlyeq t_2^{\mathcal{A}} \to t_1^{\mathcal{A}} t_2^{\mathcal{A}}$, and thus by subtyping $t_1^{\Lambda} \Vdash t_2^{\mathcal{A}} \to t_1^{\mathcal{A}} t_2^{\mathcal{A}}$. Since we have $t_2^{\Lambda} \Vdash t_2^{\mathcal{A}}$ by induction hypothesis, we can conclude that $t_1^{\Lambda} t_2^{\Lambda} \Vdash t_1^{\mathcal{A}} t_2^{\mathcal{A}}$.

□

We can thus conclude that the consistency of the AKS induces the one (in the sense of Definition 10.23) of the associated implicative structures:

Proposition 10.27. *If $\mathcal{K}$ is a consistent abstract Krivine structure, then the implicative structure it induces is classically consistent.*

Proof. Let t be any closed λ_c-term. We want to show that $t^{\mathcal{A}} \neq \bot = \Pi$. We show that $t^{\mathcal{A}}$, which belongs to $\mathcal{P}(\Pi)$ is realized by a proof-like term □

It is worth noting that the previous reasoning also applies to Krivine ordered combinatory algebras, since they induce abstract Krivine structures. Besides, the criterion of consistency is defined in both case with respect to the set **PL** (the filter for $^{\mathcal{K}}$OCAs, recall that both are identified through the passage from $^{\mathcal{K}}$OCA to AKS). Beyond that, this set (together with the pole in the case of AKS) is the key ingredient in the definition of the realizability tripos. It is already at the heart of the definition of Krivine's realizabilty models, where valid formulas are precisely the formulas realized by a proof-like term. We shall then introduce the corresponding ingredient for implicative structures.

10.3 Implicative algebras

10.3.1 Separation

Definition• 10.28 (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. $\mathbf{k}^{\mathcal{A}} \in \mathcal{S}$, and $\mathbf{s}^{\mathcal{A}} \in \mathcal{S}$. (Combinators)
2. If $a \in \mathcal{S}$ and $a \preccurlyeq b$, then $b \in \mathcal{S}$. (Upwards closure)
3. If $(a \to b) \in \mathcal{S}$ and $a \in \mathcal{S}$, then $b \in \mathcal{S}$. (Closure under modus ponens)

A separator $\mathcal{S}$ is said to be *classical* if besides $\mathbf{cc}^{\mathcal{A}} \in \mathcal{S}$ and *consistent* if $\bot \notin \mathcal{S}$. ⌟

Remark• 10.29 (Alternative definition)**.** In presence of condition (2), condition (3) is equivalent to the following condition:

(3') If $a \in \mathcal{S}$ and $b \in \mathcal{S}$ then $ab \in \mathcal{S}$ (Closure under application)

The proof uses basic properties of application:

- (3) ⇒(3'): If $a \in \mathcal{S}$ and $b \in \mathcal{S}$, since $a \preccurlyeq b \to ab$ (Proposition 10.17) by upward closure we have $b \to ab \in \mathcal{S}$, and thus $ab \in \mathcal{S}$ by modus ponens .
- (3')⇒(3): If $a \in \mathcal{S}$ and $a \to b \in \mathcal{S}$, then $(a \to b)a \in \mathcal{S}$ by closure under application. Since $(a \to b)a \preccurlyeq b$ (Proposition 10.17) by upward closure we conclude that $b \in \mathcal{S}$.

⌟

Intuitively, thinking of elements of an implicative structure as truth values, a separator should be understood as the set which distinguishes the valid formulas. Considering the elements as terms, it should rather be viewed as the set of valid realizers. Indeed, conditions (1) and (3') ensure that all λ-terms are in any separator. Reading $a \preccurlyeq b$ as "*the formula a is a subtype of the formula b*", condition (2) ensures the validity of semantic subtyping. Thinking of the ordering as "*a is a realizer of the formula b*", condition (2) states that if a formula is realized, then it is in the separator.

Definition• 10.30 (Implicative algebra)**.** We call *implicative algebra* any quadruple $(\mathcal{A}, \preccurlyeq, \to, \mathcal{S})$ where $(\mathcal{A}, \preccurlyeq, \to)$ is an implicative structure and $\mathcal{S}$ is a separator over $\mathcal{A}$. We say that an implicative algebra is *classical* if its separator is. ⌟

Example• 10.31 (Complete Boolean algebras)**.** If $\mathcal{B}$ is a complete Boolean algebra, then $\mathcal{B}$ induces an implicative structure. Besides, the interpretation of any closed λ-term is equal to $\top$ (Proposition 10.24), and it is easy to verify that for all $a, b \in \mathcal{B}$, $(((a \to b) \to a) \to a) = (\neg(\neg(\neg a \curlyvee b) \curlyvee a) \curlyvee a) = \top$, so that in particular $\mathbf{cc}^{\mathcal{B}} = \top$. Therefore, the singleton $\{\top\}$ is a classical separator for the induced implicative structure (it is obviously closed under modus ponens and upward closure). Any non-degenerated complete Boolean algebras thus induces a classically consistent implicative algebra.

Alternatively, any filter for $\mathcal{B}$ defines a separator: a filter is upward closed and closed under (binary) meets by definition. Since the application ab in Boolean algebras coincide with the binary meet $a \curlywedge b$ (Remark 10.13), any filter satisfies conditions (2) and (3') ⌟

Example 10.32 (Abstract Krivine structure)**.** Recall that any AKS $(\mathbf{\Lambda}, \mathbf{\Pi}, \mathsf{app}, \mathsf{push}, \mathsf{k}_{\_}, \boldsymbol{k}, \boldsymbol{s}, \boldsymbol{cc}, \mathbf{PL}, \perp\!\!\!\perp)$ induces an implicative structure $(\mathcal{A}, \preccurlyeq, \to)$ where $\mathcal{A} = \mathcal{P}(\mathbf{\Pi})$, $a \preccurlyeq b \Leftrightarrow a \supseteq b$ and $a \to b = a^{\perp\!\!\!\perp} \cdot b$. The sets of realized formulas, namely $\mathcal{S} = \{a \in \mathcal{A} : a^{\perp\!\!\!\perp} \cap \mathbf{PL} \neq \emptyset\}$, defines a valid separator. The condition of upward-closure is obvious by subtyping and we saw in Lemma 10.26 that $\boldsymbol{\kappa}^{\mathcal{A}}, \mathbf{s}^{\mathcal{A}}, \mathbf{cc}^{\mathcal{A}}$ were realized respectively by $\boldsymbol{k}, \boldsymbol{s}$ and $\boldsymbol{cc}$. As for the closure under modus ponens, for any $a, b \in \mathcal{A}$, if $(a \to b) \in \mathcal{S}$ and $a \in \mathcal{S}$, by definition there exist $t, u \in \mathbf{\Lambda}$ such that $t \Vdash a \to b$ and $u \Vdash a$. Therefore, $tu \Vdash b$ and thus $b \in S$. ⌟

10.3.2 λ_c-terms

The first property that we shall state about classical separators is that they contain the interpretation of all closed λ_c-terms. This follows again from the combinatorial completeness of the basis $(\boldsymbol{\kappa}, \mathbf{s}, \mathbf{cc})$ for the λ_c-calculus. Indeed, if $\mathcal{S}$ is a classical separator over an implicative structure $(\mathcal{A}, \preccurlyeq, \to)$, it is clear that any combinatory term is in the separator. Again, by combinatory completeness, if t is a closed λ_c-term, there exists a combinatory term t_0 such that $t_0 \to_\beta t$, and therefore $t_0^{\mathcal{A}} \preccurlyeq t^{\mathcal{A}}$ (by Proposition 10.17). By upward closure of separators, we deduce that:

Proposition• 10.33. *If $(\mathcal{A}, \preccurlyeq, \to, \mathcal{S})$ is a (classical) implicative algebra and t is a closed λ-term (resp. λ_c-term), then $t^{\mathcal{A}} \in \mathcal{S}$.*

From the previous proposition and the adequacy of second-order typing rules for the λ_c-calculus (Theorem 10.19), we obtain that:

Corollary• 10.34. *If $(\mathcal{A}, \preccurlyeq, \to, \mathcal{S})$ is a (classical) implicative algebra, t is a closed λ-term (resp. λ_c-term) and A is a formula such that $\vdash t : A$, then $A^{\mathcal{A}} \in \mathcal{S}$.*

Remark 10.35. The latter corollary provides us with a methodology for proving that an element of a given implicative algebra is in the separator. In the spirit of realizability, where the standard methodology to prove that a formula is realized consists in using typed terms and adequacy as much as possible, we can use typed terms to prove automatically that the corresponding formulas belongs to the separator. We shall use this methodology[4] abundantly in the sequel. ⌟

10.3.3 Internal logic

In order to be able to define triposes from implicative algebras, we first need to equip them with a structure of Heyting algebra. To this end, we begin with defining an entailment relation in the spirit of filtered OCAs. We then define quantifiers and connectives as usual in classical realizability (see Section 3.3.1.1), and we verify that they satisfy the usual logical rules. This will lead us to the definition of the implicative tripos.

10.3.3.1 Entailment

In the rest of this section, we work within a fixed implicative algebra $(\mathcal{A}, \preccurlyeq, \to, \mathcal{S})$.

Definition• 10.36 (Entailment)**.** For all $a, b \in \mathcal{A}$, we say that *a entails b* and write $a \vdash_{\mathcal{S}} b$ if $a \to b \in \mathcal{S}$. We say that a and b are equivalent and write $a \cong_{\mathcal{S}} b$ if $a \vdash_{\mathcal{S}} b$ and $b \vdash_{\mathcal{S}} a$. ⌟

Proposition 10.37 (Properties of $\vdash_{\mathcal{S}}$)**.** *For any $a, b, c \in \mathcal{A}$, the following holds:*

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)*

[4] In the Coq development, this corresponds to the tactic called `realizer•` which we indeed use a lot.

3.• *if* $a \preccurlyeq b$ *then* $a \vdash_{\mathcal{S}} b$ *(Subtyping)*

4.• *if* $a \cong_{\mathcal{S}} b$ *then* $a \in \mathcal{S}$ *if and only if* $b \in \mathcal{S}$ *(Closure under* $\cong_{\mathcal{S}}$*)*

5.• *if* $a \vdash_{\mathcal{S}} b \to c$ *then* $a \curlywedge b \vdash_{\mathcal{S}} c$ *(Half-adjunction property)*

6.• $\bot \vdash_{\mathcal{S}} a$ *(Ex falso quod libet)*

7.• $a \vdash_{\mathcal{S}} \top$ *(Maximal element)*

Proof. 2. We go once and for all through all the steps of the methodology described in Remark 10.35. If $a \vdash b$ and $b \vdash c$, we have by definition that $a \to b \in \mathcal{S}$ and $b \to c \in \mathcal{S}$. We use the closure under modus ponens and prove that $(a \to b) \to (b \to c) \to (a \to c) \in \mathcal{S}$. Besides, let us define $t \triangleq \lambda xyz.y(xz)$. It is clear that we can derive $\vdash t : \forall abb'.(a \to b) \to (b \to c) \to (a \to c)$ in System F, whence by Theorem 10.19 we have:

$$t^{\mathcal{A}} \preccurlyeq \min_{a,b,c \in \mathcal{A}} (a \to b) \to (b \to c) \to (a \to c)$$

Since $t^{\mathcal{A}} \in \mathcal{S}$ (Proposition 10.33) and $\mathcal{S}$ is upward closed, we get the expected result. In the sequel, we shall simply say that the formula is realized by $\lambda xyz.y(xz)$.

3. This is realized by the identity (by subtyping).

4. Direct from the definition of $\cong_{\mathcal{S}}$ and the closure under modus ponens.

5. The formula $(a \to b \to c) \to a \curlywedge b \vdash_{\mathcal{S}} c$ is realized (using the fact that $a \curlywedge b \preccurlyeq a, b$) $W = \lambda xy.xyy$.

1,6,7. Direct from 3. □

Besides, the entailment relation is compatible with respect to the monotonicity of the arrow:

Proposition 10.38 (Compatibility with →)**.** *The following hold for all* $a, a', b, b' \in \mathcal{A}$*:*

1.• *If* $b \vdash b'$ *then* $a \to b \vdash a \to b'$ 2.• *If* $a \vdash a'$ *then* $a' \to b \vdash a \to b$

Proof. 1. If $b \vdash b'$, we have by definition $b \to b' \in \mathcal{S}$. We use the closure under modus ponens and prove that $(b \to b') \to (a \to b) \to (a \to b') \in \mathcal{S}$. This formula is realized by $\lambda xyz.x(yz)$.

2. Similarly, we prove that $(a \to b') \to (a' \to b) \to (a \to b) \in \mathcal{S}$ since it is realized by $\lambda xyz.y(xz)$. □

Therefore, the arrow behaves like Heyting's arrow with respect to the preorder relation $\vdash_{\mathcal{S}}$ in terms of monotonicity. Nonetheless, we only have half the adjunction with the meet. Indeed, the other direction (if $a \curlywedge b \vdash_{\mathcal{S}} c$ then $a \vdash_{\mathcal{S}} b \to c$) does not make sense computationally, since the meet does not reflect a logical connective. This should not come as a surprise, since we explained in Section 9.1.1 that in realizability, the conjunction was interpreted by the product type rather than the meet.

10.3.3.2 Negation

Recall that the negation is defined by $\neg a \triangleq a \to \bot$. If additionally the separator is classical, we can prove that for any $a \in \mathcal{A}$, we have:

Proposition 10.39 (Double negation)**.** *If* $\mathcal{S}$ *is a classical separator, the following holds for any* $a \in \mathcal{A}$*:*

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

Proof. 1. Trivial, since it is realized by $\lambda xk.kx$.

2. Follows from the inequality $((a \to \bot) \to a) \to a \preccurlyeq ((a \to \bot) \to \bot) \to a$, whose left member is realized by **cc**.

□

10.3.3.3 Quantifiers

Following the usual definition in classical realizability (see Section 9.1.1), the universal quantification of a family of truth values is naturally defined as its meet. Therefore, we introduce the convenient notation:

$$\bigforall_{i\in I} a_i \triangleq \bigwedge_{i\in I} a_i$$

It is clear that this definition is compatible with the expected semantic rules:

Proposition• 10.40 (Universal quantifier)**.** *The following semantic typing rules are valid in any implicative structures:*

$$\frac{\Gamma \vdash t : a_i \quad \textit{for all } i \in I}{\Gamma \vdash t : \forall_{i\in I}\, a_i} \qquad \frac{\Gamma \vdash t : \forall_{i\in I}\, a_i \quad i_0 \in I}{\Gamma \vdash t : a_{i_0}}$$

Dually, we follow the usual encodings of the existential quantification (see Section 3.3.1.1), and we define:

$$\bigexists_{i\in I} a_i \triangleq \bigwedge_{c\in A} \Big(\bigwedge_{i\in I} (a_i \to c) \to c\Big)$$

While it could have seemed more natural to define existential quantifiers through joins, we should recall that the arrow does not commute with joins in general. We shall see in Section 10.4.4.2 that when it does, the realizability tripos precisely collapses to a forcing tripos. Once more, the expected semantic typing rules are satisfied:

Proposition• 10.41 (Existential quantifier)**.** *The following semantic typing rules are valid in any implicative structures:*

$$\frac{\Gamma \vdash t : a_{i_0} \quad i_0 \in I}{\Gamma \vdash \lambda x.xt : \exists_{i\in I}\, a_i} \qquad \frac{\Gamma \vdash t : \exists_{i\in I}\, a_i \quad \Gamma, x : a_i \vdash u : c \quad (\textit{for all } i \in I)}{\Gamma \vdash t(\lambda x.u) : c}$$

Proof. Straightforward using the adjunction of the application (Proposition 10.12) and lattices properties. For instance, for the introduction rule, assume that $(t[\Gamma])^{\mathcal{A}} \preccurlyeq a_i$ for some $i \in I$. Then we have to prove that $(\lambda x.xt[\Gamma])^{\mathcal{A}} \preccurlyeq \bigwedge_{c\in A}(\bigwedge_{i\in I}(a_i \to c) \to c)$. Let then c be in $\mathcal{A}$, using the adjunction it suffices to prove that:

$$(\lambda x.xt[\Gamma])^{\mathcal{A}}\Big(\bigwedge_{i\in I}(a_i \to c)\Big) \preccurlyeq c$$

Using the property of β-reduction (Proposition 10.17) and the transitivity, it is enough to show that:

$$\Big(\bigwedge_{i\in I}(a_i \to c)(t[\Gamma])^{\mathcal{A}}\Big) \preccurlyeq c \quad \Leftrightarrow \quad \bigwedge_{i\in I}(a_i \to c) \preccurlyeq (t[\Gamma])^{\mathcal{A}} \to c$$

We conclude using the hypothesis for t and the anti-monotonicity of the arrow. The proof for the elimination rule is very similar. Observe that we really consider the elements of the implicative structure as λ-terms, that is to say that we compute with truth values. □

10.3.3.4 Sum and product

We define it by the usual encodings in System F:

$$a \times b \triangleq \bigwedge_{c\in\mathcal{A}} ((a \to b \to c) \to c)$$

Recall that the pair $\langle a, b\rangle$ is encoded by the λ-term $\lambda x.xab$, while first and second projection are respectively defined by $\pi_1 \triangleq \lambda xy.x$ and $\pi_2 \triangleq \lambda xy.y$. We can check that the expected semantic typing rules for pairs are valid

Proposition• 10.42 (Product)**.** *The following semantic typing rules are valid:*

$$\frac{\Gamma \vdash t : a \quad \Gamma \vdash u : b}{\Gamma \vdash \lambda z.ztu : a \times b} \qquad \frac{\Gamma \vdash t : a \times b}{\Gamma \vdash t\pi_1 : a} \qquad \frac{\Gamma \vdash t : a \times b}{\Gamma \vdash t\pi_2 : b}$$

Proof. Straightforward lattice manipulation, similar to the proof for the existential quantifier. □

Similarly, we can define a sum type through the usual encoding:

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

We check again that the expected semantic typing rules for pairs are valid:

Proposition• 10.43 (Sum)**.** *The following semantic typing rules are valid:*

$$\frac{\Gamma \vdash t : a}{\Gamma \vdash \lambda lr.lt : a + b} \qquad \frac{\Gamma \vdash t : b}{\Gamma \vdash \lambda lr.rt : a + b} \qquad \frac{\Gamma \vdash t : a + b \quad \Gamma, x : a \vdash u : c \quad \Gamma, y : b \vdash v : c}{\Gamma \vdash t(\lambda x.u)(\lambda y.v) : c}$$

Proof. Straightforward lattice manipulation. □

We are now ready to verify that the entailment relation together with the sum and products induce a structure of Heyting algebra. We will then focus to the construction of the implicative tripos.

10.4 Implicative triposes

10.4.1 Induced Heyting algebra

The natural candidate which computationally represents a "meet" of a and b is the product type $a \times b$. We can verify that it satisfies the expected property (in Heyting algebras) w.r.t. to the arrow:

Proposition• 10.44 (Adjunction)**.** *For any $a, b, c \in \mathcal{A}$, we have:*

$$a \vdash_{\mathcal{S}} b \to c \qquad \textit{if and only if} \qquad a \times b \vdash_{\mathcal{S}} c$$

Proof. Both directions are proofs using the expected realizer and subtyping: from left to right, we use $\lambda xy.yx$ to realize $(a \to b \to c) \to a \times b \to c$; from right to left, we realize $(a \times b \to c) \to a \to b \to c$ with $\lambda pxy.p(\lambda z.zxy)$. □

Corollary 10.45 (Heyting prealgebra)**.** *For any implicative algebra $(\mathcal{A}, \preccurlyeq, \to, \mathcal{S})$, the induced quintuple $(\mathcal{A}, \vdash_{\mathcal{S}}, \times, +, \to)$ is a Heyting prealgebra.*

The former is only a Heyting prealgebra and not a Heyting algebra because the entailment relation is a preorder (instead of an order). We thus consider the quotient $\mathcal{A}/_{\cong_{\mathcal{S}}}$ of the former Heyting prealgebra by the relation $\cong_{\mathcal{S}}$, which we write $\mathcal{A}/\mathcal{S}$ (and $\mathcal{H}$ hereafter). We equip $\mathcal{H}$ with an order relation:

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

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

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

Proposition 10.46 (Induced Heyting algebra)**.** *The quintuple* $(\mathcal{H}, \leq_{\mathcal{H}}, \wedge_{\mathcal{H}}, \vee_{\mathcal{H}}, \to_{\mathcal{H}})$ *is a Heyting algebra.*

Proof. We first show that $(\mathcal{H}, \leq_{\mathcal{H}}, \wedge_{\mathcal{H}}, \wedge_{\mathcal{H}})$ is a lattice. It is clear that $(\mathcal{H}, \leq_{\mathcal{H}})$ is a poset, we then have to prove that $\wedge_H$ and $\vee_H$ indeed defines binary meets and joins. We thus need to prove that for all $a, b, c \in \mathcal{A}$, we have:

1. $[a] \times [b] \leq_{\mathcal{H}} [a]$ and $[a] \times [b] \leq_{\mathcal{H}} [b]$. In $\mathcal{A}$, the corresponding implications are realized respectively by $\lambda xy.x$ and $\lambda xy.y$.
2. If $[c] \leq_{\mathcal{H}} [a]$ and $[c] \leq_{\mathcal{H}} [b]$, then $[c] \leq_{\mathcal{H}} [a] \times [b]$. Let us assume that $c \to a \in \mathcal{S}$ and $c \to b \in \mathcal{S}$. Then by closure of the separator under modus ponens, it suffices to show that $(c \to a) \to (c \to b) \to c \to (a \times b) \in \mathcal{S}$. This formula is realized by $\lambda tucz.z(tc)(uc)$.
3. $[a] \leq_{\mathcal{H}} [a] + [b]$ and $[b] \leq_{\mathcal{H}} [a] + [b]$. The corresponding implications in $\mathcal{A}$ are realized respectively by $\lambda xtu.tx$ and $\lambda xtu.ux$.
4. If $[a] \leq_{\mathcal{H}} [c]$ and $[b] \leq_{\mathcal{H}} [c]$, then $[c] \leq_{\mathcal{H}} [a] + [b]$. Let us assume that $c \to a \in \mathcal{S}$ and $c \to b \in \mathcal{S}$. Then by closure of the separator under modus ponens, it suffices to show that $(a \to c) \to (b \to c) \to (a + b) \to c \in \mathcal{S}$. This formula is realized by $\lambda xyt.txy$.

We already know from Proposition 10.37 that $\top$ and $\bot$ are respectively the maximal and minimal elements of $\mathcal{A}$ for $\leq_{\mathcal{H}}$. Thus $(\mathcal{H}, \leq_{\mathcal{H}}, \wedge_{\mathcal{H}}, \wedge_{\mathcal{H}})$ is a bounded lattice.

Finally, we need to prove that the adjunction $[a] \wedge_{\mathcal{H}} [b] \leq_{\mathcal{H}} [c] \Leftrightarrow [a] \leq_{\mathcal{H}} [b] \to_{\mathcal{H}} [c]$ holds for any $a, b, c \in \mathcal{A}$. This is a direct consequence of the corresponding adjunction that we proved in $\mathcal{A}$ for $\vdash_S$ and $\to$ (Proposition 10.44). □

Remark 10.47. If the implicative algebra is classical, for all $a \in \mathcal{A}$ we have $\neg\neg a \cong_{\mathcal{S}} a$ (Proposition 10.39). Through the quotient, this implies that $\neg\neg[a] = [a]$ for all $a \in \mathcal{A}$. This means that in the case of a classical implicative algebra, the induced Heyting algebra is actually a Boolean algebra. ⌟

We are almost ready to define the implicative tripos. Following the construction of triposes associated to AKSs and $^{\mathcal{K}}$OCAs, we want to define a functor roughly of the form $\mathcal{P} : I \in \mathbf{Set}^{op} \mapsto \mathcal{A}^I$. However, as we saw that the implicative algebra $\mathcal{A}$ gives rise to a Heyting algebra through a quotient by (the equivalence relation induced by) the separator. We first need to check that the indexed family $\mathcal{A}^I$ is an implicative structure. Then we will need to quotient $\mathcal{A}^I$ by an appropriate separator.

10.4.2 Product of implicative structures

Let I be a set and $(A_i)_{i \in I}$ be a family of implicative structures, which we write $(\mathcal{A}_i, \preccurlyeq_i, \to_i)$. The Cartesian product $\mathcal{A} \triangleq \prod_{i \in I} A_i$ is naturally equipped with a structure of implicative structure, using the order and implication defined componentwise:

$$(a_i)_{i \in I} \preccurlyeq (b_i)_{i \in I} \triangleq \forall i \in I.(a_i \preccurlyeq_i b_i) \qquad (a_i)_{i \in I} \to (b_i)_{i \in I} \triangleq (a_i \to_i b_i)_{i \in I}$$

Proposition 10.48 (Product of structures)**.** *The triple* $(\mathcal{A}, \preccurlyeq, \to)$ *is an implicative structure.*

Proof. Straightforward, since the variance and the distributivity are verified for each component. □

Since the order relation is defined componentwise, in particular the meet of a set of family is the family of the meet componentwise:

$$\bigwedge_{(a_i)_{i \in I} \in \mathcal{A}} (a_i)_{i \in I} = \left(\bigwedge_{a_i \in \mathcal{A}_i} a_i \right)_{i \in I}$$

As a consequence, all the definitions are compatible with the corresponding definitions componentwise, namely for all $a, b \in \mathcal{A}$ and any $f = (f_i : \mathcal{A}_i \to \mathcal{A}_i)_{i \in I}$ we have:

$$ab = (a_i b_i)_{i \in I} \qquad \lambda f = (\lambda f_i)_{i \in I} \qquad \mathbf{k}^{\mathcal{A}} = (\mathbf{k}^{\mathcal{A}_i})_{i \in I} \qquad \mathbf{s}^{\mathcal{A}} = (\mathbf{s}^{\mathcal{A}_i})_{i \in I}$$
$$a \times b = (a_i \times b_i)_{i \in I} \qquad a + b = (a_i + b_i)_{i \in I} \qquad \mathbf{cc}^{\mathcal{A}} = (\mathbf{cc}^{\mathcal{A}_i})_{i \in I}$$

In the same spirit, it is clear that if $(\mathcal{S}_i)_{i \in I} \subseteq \mathcal{A}_i$ is a family of separators (*i.e.* for each $i \in I$, $\mathcal{S}_i$ is a separator for $\mathcal{A}_i$), then the Cartesian product $\mathcal{S} = \prod_{i \in I} \mathcal{S}_i$ is a separator for the implicative structure $\mathcal{A}$. Besides, the entailment relation induced by this separator product corresponds again to the induced relation componentwise, since for all $a, b \in \mathcal{A}$ we have:

$$a \vdash_{\mathcal{S}} b \triangleq a \to b \in \mathcal{S} \quad \Leftrightarrow \quad \forall i \in I.(a_i \to_i b_i \in \mathcal{S}_i) \quad \Leftrightarrow \quad \forall i \in I.(a_i \vdash_{\mathcal{S}_i} b_i)$$

10.4.3 Implicative tripos

We are now ready to define the implicative tripos. Let $(\mathcal{A}, \preccurlyeq, \to, \mathcal{S})$ be a fixed implicative algebra. For each set I, the Cartesian product $\mathcal{A}^I$ gives rise to an implicative structure which we write $(\mathcal{A}^I, \preccurlyeq^I, \to^I)$. As explained in the previous section, the Cartesian product $\mathcal{S}^I$ defines a separator for the implicative structure $\mathcal{A}^I$, which we call the *power separator*. By definition, an element a of $\mathcal{A}^I$ belongs to the power separator $\mathcal{S}^I$, if for each $i \in I$, a_i belongs to $\mathcal{S}$. In terms of realizability, this intuitively means that for each $i \in I$, a_i is realized.

As we shall see further, this separator is too permissive in the sense that it contains too many elements and that the corresponding quotient collapses to a forcing tripos. Yet, the separator $\mathcal{S}$ induces another separator, which we write $\mathcal{S}[I]$ and call *uniform separator*, which is defined by:

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

An element $a \in \mathcal{A}$ is thus in the uniform separator if it is uniformly realized by the same s in each component. We clearly have the following inclusion:

$$\mathcal{S}[I] \subseteq \mathcal{S}^I \subseteq \mathcal{A}^I$$

We write $(\mathcal{A}^I/\mathcal{S}[I], \leq_{\mathcal{S}[I]}, \to_{\mathcal{S}[I]})$ the associated Heyting algebra.

Theorem 10.49 (Implicative tripos)**.** *Let $(\mathcal{A}, \preccurlyeq, \to, \mathcal{S})$ be an implicative algebra. The following functor :*

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

defines[5] *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[6]:

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

[5]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.

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

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

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

Let then $[(a_{ij})_{i,j\in I}]$ be an element of $\mathcal{T}(I\times I)$. From left to right, assume that $[\top]_I \leq_{\mathcal{S}[I]} \mathcal{T}(\delta_I)(a)$, 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:

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

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:

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

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 $\mathtt{Prop} \triangleq \mathcal{A}$ and verify that $\mathtt{tr} \triangleq [\mathrm{id}_{\mathcal{A}}] \in \mathcal{T}(\mathtt{Prop})$ 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 $I \to \mathtt{Prop}$), which obviously satisfies that for all $i \in I$:

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

□

10.4.4 Relation with forcing triposes

10.4.4.1 The fundamental diagram

We shall now briefly present a criterion to determine whether an implicative tripos is equivalent to a forcing tripos. By forcing tripos, we refer to a tripos of the shape $\mathcal{T} : I \mapsto \mathcal{H}^I$ where $\mathcal{H}$ is a complete

Heyting (or Boolean in classical case) algebra (see Example 9.22). In particular, recall that in the case of forcing (see Section 9.1.2), we have:

$$\forall = \curlywedge = \wedge$$

while it is worth observing that the definition of the implicative tripos is in adequacy with the usual situation of in realizability, that is to say that we have:

$$\forall = \curlywedge \qquad\qquad \wedge = \times$$

In the case of the implicative tripos, the algebra $\mathcal{T}(I)$ of predicates associated to the set I is defined by $\mathcal{T}(I) = \mathcal{A}^I/\mathcal{S}[I]$, that is: as the quotient of the power implicative algebra $\mathcal{A}^I$ by the uniform power separator $\mathcal{S}[I]$. Note that here, we used the uniform power separator $\mathcal{S}[I]$ and not the pointwise power separator $\mathcal{S}^I$, precisely to avoid a trivialization of the form $\mathcal{A}^I/S^I = (\mathcal{A}/S)^I$ that would amount to a forcing tripos, based on the Heyting algebra $\mathcal{H} = \mathcal{A}/\mathcal{S}$.

Indeed, we saw in Section 10.4.2 that the separator product $\mathcal{S}^I$ also defines a separator for the algebra $\mathcal{A}^I$. We could have considered instead the quotient $\mathcal{A}^I/\mathcal{S}^I$. Since $\mathcal{S}[I] \subseteq \mathcal{S}^I$, in particular we have that if a and b are two elements of $\mathcal{A}^I$ and if besides $a \cong_{S[I]} b$, then $a \cong_{S^I} b$. In other words, the map which associates to each equivalence class w.r.t. $\mathcal{S}[I]$ the equivalence class of its representative w.r.t. $\mathcal{S}^I$:

$$\iota_I : \left\{ \begin{array}{ccc} \mathcal{A}^I/\mathcal{S}[I] & \to & \mathcal{A}^I/\mathcal{S}^I \\ [a]_{/\mathcal{S}[I]} & \mapsto & [a]_{/\mathcal{S}^I} \end{array} \right.$$

is surjective onto $\mathcal{A}^I/\mathcal{S}^I$.

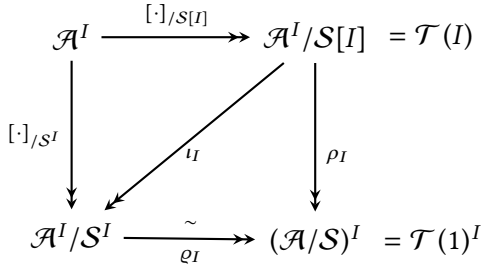
Moreover, we could have directly defined a tripos by taking the quotient $\mathcal{A}/\mathcal{S}$ (which defines a Heyting algebra $\mathcal{H}$), and considered the functor which associates to each I the product $(\mathcal{A}/\mathcal{S})^I$. This situation corresponds precisely to a forcing tripos. Here again, we can define the map which associates to each equivalence class $[(a_i)_{i\in I}]$ w.r.t. $\mathcal{S}[I]$ the sequence of equivalence classes of the a_i w.r.t. $\mathcal{S}$:

$$\rho_I : \left\{ \begin{array}{ccc} \mathcal{A}^I/\mathcal{S}[I] & \to & (\mathcal{A}/\mathcal{S})^I \\ [a]_{/\mathcal{S}[I]} & \mapsto & [a_i]_{/\mathcal{S}} \end{array} \right.$$

which is surjective onto $(\mathcal{A}/\mathcal{S})^I$. Finally, it is clear that $\mathcal{A}^I/\mathcal{S}^I$ and $(\mathcal{A}/\mathcal{S})^I$ are in bijection: in $\mathcal{A}^I/\mathcal{S}^I$, two elements $[(a_i)_{i\in I}]$ and $[(v_i)_{i\in I}]$ are in the same equivalence class if they are equivalent componentwise, that is for all $i \in I$, a_i and b_i are equivalent:

$$[(a_i)_{i\in I}] \cong_{\mathcal{S}^I[(b_i)_{i\in I}]} \quad \Leftrightarrow \quad \forall i \in I.[a_i] \cong_{\mathcal{S}} [b_i]$$

Denoting by ϱ_I the corresponding bijection from $\mathcal{A}^I/\mathcal{S}^I$ to $(\mathcal{A}/\mathcal{S})^I$, the situation can then be summarized by the following diagram:



In this diagram, all the maps are surjective, the top right corner corresponds to the implicative tripos while the bottom right one corresponds to a forcing tripos. We shall now make use of the diagram to precise the situation. To this purpose, we first need to prove a lemma about morphisms of Heyting algebras.

Lemma 10.50. *Let $\mathcal{H},\mathcal{H}'$ be two Heyting algebras. If $f : \mathcal{H} \to \mathcal{H}'$ be a morphism of Heyting algebras, then f is an isomorphism if and only if f is bijective.*

Proof. The left to right implication is trivial, we thus have to prove that if f is a one-to-one morphism, then f^{-1} is a morphism. It is easy to see that f^{-1} preserves the lattice structure and the implication because f does. For instance for the preservation of meets, for all $a,b \in \mathcal{H}'$ we have:

$$f^{-1}(a \curlywedge b) = f^{-1}(f(f^{-1}(a)) \curlywedge f(f^{-1}(b))) = f^{-1}(f(f^{-1}(a) \curlywedge f^{-1}(b))) = f^{-1}(a) \curlywedge f^{-1}(b)$$

As for the preservation of the order, if $a,b \in \mathcal{H}'$ are such that $a \preccurlyeq b$, then $a = a \curlywedge b$ and we have:

$$f^{-1}(a) = f^{-1}(a \curlywedge b) = f^{-1}(a) \curlywedge f^{-1}(b) \preccurlyeq f^{-1}(b)$$

Therefore, we can conclude that $f^{-1}(a) \preccurlyeq f^{-1}(b)$. □

Using the previous lemma, we obtain the following characterization:

Proposition 10.51. *The following are equivalent:*

1. *The map: $\rho_I : (\mathcal{A}^I/\mathcal{S}[I]) \to (\mathcal{A}/\mathcal{S})^I$ is an isomorphism (of Heyting Algebras).*
2. *The map: $\rho_I : (\mathcal{A}^I/\mathcal{S}[I]) \to (\mathcal{A}/\mathcal{S})^I$ is injective.*
3. *$\mathcal{S}[I] = \mathcal{S}^I$.*
4. *The separator $\mathcal{S} \subseteq A$ is closed under all I-indexed meets.*

Proof. The equivalence between the first three conditions follows from the above characterization of isomorphisms in **HA**. If $(a_i)_{i\in I} \in \mathcal{S}^I$ and $\mathcal{S}[I] = \mathcal{S}^I$, then there exists an $s \in \mathcal{S}$ such that for all $i \in I$, $s \preccurlyeq a_i$. Then $s \preccurlyeq \bigcurlywedge_{i\in I} a_i$ and the latter belongs to $\mathcal{S}$ by upward closure. Therefore, $\mathcal{S}$ is closed under I-indexed meets. For the converse direction, it suffices to see that if $(a_i)_{i\in I} \in \mathcal{S}^I$, then by closure under I-indexed meets $\bigcurlywedge_{i\in I} a_i$ is in $\mathcal{S}$ and is a uniform realizer for $(a_i)_{i\in I}$, which thus belongs to $\mathcal{S}[I]$. □

This diagram is thus the cornerstone on the study of implicative tripos. In particular, the most interesting realizability models (*i.e.* those which can not be obtained by forcing) are the ones occurring in the top right corner when the map ρ_I is not an isomorphism.

10.4.4.2 Collapse criteria

We shall briefly present some criteria which characterizes the situations where implicative triposes are isomorphic to forcing triposes. As we do not want to enter into too much detail here (we leave it for the forthcoming paper of Alexandre Miquel on the topic), let us loosely use notions that we do not formally define. Our goal here is mainly to give some intuitions, and to highlight some phenomena that were already known in Krivine realizability algebras.

First of all, as we mentioned in Section 3.5.3, the construction of Krivine's realizability models for the negation of the axiom of choice and the continuum hypothesis deeply relies on the fact that the formula $\textsc{ind} \equiv \forall x.\mathrm{Nat}(x)$ is not realized. In our framework, this formula can be defined by:

$$\textsc{ind}^{\mathcal{A}} \triangleq \bigcurlywedge_{n\in\mathbb{N}} \bigcurlywedge_{a\in\mathcal{A}^{\mathbb{N}}} (a_0 \to \bigcurlywedge_{i\in\mathbb{N}} (a_i \to a_{i+1}) \to a_n)$$

In fact, this can be reduced to the formula called *parallel-or* (**p-or**), which is defined in any implicative structure by:

$$\textbf{p-or}^{\mathcal{A}} \triangleq (\top \to \bot \to \bot) \curlywedge (\bot \to \top \to \bot)$$

in the sense that if this formula is realized if and only if IND is[7]. Besides, in the case where the realizability algebra (*i.e.* the λ_c-calculus) contains an instruction $\pitchfork$ of non-deterministic choice, it is easy to define a realizer for the formula **p-or**. In which case, the realizability models collapses to a forcing model.

This phenomenon can be rephrased directly within implicative algebras. First, the operator $\pitchfork$ is naturally interpreted in any implicative structure $\mathcal{A}$ by:

$$\pitchfork^{\mathcal{A}} \triangleq \bigwedge_{a,b \in A} (a \to b \to a \curlywedge b)$$

and it is an easy exercise of λ_c-calculus to show that:

Proposition 10.52. *If $(\mathcal{A}, \preccurlyeq, \to, \mathcal{S})$ is a classical implicative algebra, then:*

$$\pitchfork^{\mathcal{A}} \cong_{\mathcal{S}} \textbf{\textit{p-or}}^{\mathcal{A}} \cong_{\mathcal{S}} \mathit{IND}^{\mathcal{A}}$$

Then it is possible to show that an implicative tripos is isomorphic to a forcing tripos if and only if its separator contains $\pitchfork^{\mathcal{A}}$ and is *finitely generated* (*i.e.* it is defined as the closure under application and upwards of a finite subset of the implicative structure $\mathcal{A}$).

Theorem 10.53 (Characterization of forcing triposes)**.** *Let $\mathcal{T} : \mathbf{Set}^{op} \to \mathbf{HA}$ be an implicative tripos induced by an implicative algebra $(\mathcal{A}, \preccurlyeq, \to, \mathcal{S})$. The following are equivalent:*

1. *The tripos $\mathcal{T}$ is isomorphic to a forcing tripos*
2. *The separator $\mathcal{S} \subseteq \mathcal{A}$ is a principal filter of $\mathcal{A}$.*
3. *The separator $\mathcal{S} \subseteq \mathcal{A}$ is finitely generated and $\pitchfork \in \mathcal{S}$.*

Proof. See [121]. □

Furthermore, in the case where the arrow commutes with arbitrary joins, that is if for all $b \in \mathcal{A}$ the following holds:

$$\bigwedge_{a \in \mathcal{A}} (a \to b) = (\bigvee_{a \in \mathcal{A}} a) \to b$$

the interpretation of **p-or** belongs to any separator. Indeed, since $\bot = \bigvee \emptyset$, the previous equality implies that $\bot \to a = \top$ for any $a \in \mathcal{A}$, and in particular:

$$\textbf{p-or}^{\mathcal{A}} = (\bot \to \top \to \bot) \curlywedge (\top \to \bot \to \bot) = \top \curlywedge (\top \to \top) = \top \to \top$$

Therefore, in the previous situation, $\pitchfork^{\mathcal{A}}$ and $\mathrm{IND}^{\mathcal{A}}$ also belong to all classical separators. The previous equation is not meaningless, because when it holds, it allows to define the existential quantifier as a join, and it can be read as:

$$\bigvee_{a \in \mathcal{A}} (a \to b) = (\bigwedge_{a \in \mathcal{A}} a) \to b$$

In other words, we can not expect an implicative algebra which is "too" commutative to induce triposes which are not isomorphic to a forcing tripos.

[7] In Krivine's article, the fact that the algebra ∇_2 is not trivial precisely relies on the fact that there is no term which realizes both $\top \to \bot \to \bot$ and $\bot \to \top \to \bot$ (see [99, Theorem 31]).

10.5 Conclusion

We presented in this section the concept of *implicative algebra*, that relies on the primitive notion of implicative structure. These structures are defined as a particular class of meet-complete lattices equipped with an arrow, where this arrow satisfies commutations with arbitrary meets which are the counterpart of the logical commutation between the universal quantification and the implication. We showed that implicative structures are a generalization Streicher's AKSs and Ferrer *et al.*'s $\mathcal{K}$OCAs. In particular, they allow us to defined triposes arising from Krivine classical realizability models, and they provide us with simple criteria to determine whether the induced triposes are equivalent to forcing triposes. As such, implicative algebras appear to be a promising framework for the algebraic analysis of classical realizability.

This presentation is totally in line with Krivine's usual presentation of his realizability models, and in particular it takes position on a presentation of logic through universal quantification and the implication. The computational counterpart of this choice is that the presentation relies on the call-by-name λ_c-calculus. This raises the question of knowing whether it is possible to have alternative presentations with similar structures based on different connectives (and thus different calculi).

In the last two chapters of this manuscript, we will present an attempt in this perspective. Firstly, we will introduce the so-called notion of *disjunctive algebras*, which are primitively defined in disjunctive structures relying on a disjunction $\parr$ and a negation $\neg$. We will relate these connectives to a fragment of Munch-Maccagnoni's System L, which amounts to a call-by-name decomposition of the λ-calculus. In particular, we will see that any disjunctive algebra induces an implicative algebra.

Secondly, we will introduce the dual notion of *conjunctive algebras*, based on conjunctive structures whose connectives are a conjunction $\otimes$ and a negation $\neg$. Here again, this decomposition of the arrow corresponds to a fragment of Munch-Maccagnoni's System L, which amounts to a call-by-value λ-calculus. We will see that such a structure can naturally be obtained by duality from a disjunctive algebra.

These two different presentations are not as accomplished as the study of implicative algebras. In particular, we do not dispose of the full embeddings of the corresponding calculus, and we are still missing some correspondences between the three presentations. Yet, they should rather be taken as a first step toward a complete zoology of the implicative-like algebras. We conjecture that implicative algebras constitute the more general framework.