Mathematical Structures in Computer Science (2020), **30**, pp. 458–510
doi:10.1017/S0960129520000079

CAMBRIDGE UNIVERSITY PRESS

PAPER

Implicative algebras: a new foundation for realizability and forcing†

Alexandre Miquel*

Instituto de Matemática y Estadística Prof. Ing. Rafael Laguardia, Facultad de Ingeniería, Universidad de la República, Julio Herrera y Reissig 565, Montevideo C.P. 11300, Uruguay
*Corresponding author. Email: amiquel@fing.edu.uy

(Received 13 July 2019; revised 27 February 2020; accepted 9 March 2020)

Abstract

We introduce the notion of implicative algebra, a simple algebraic structure intended to factorize the model-theoretic constructions underlying forcing and realizability (both in intuitionistic and classical logic). The salient feature of this structure is that its elements can be seen both as truth values and as (generalized) realizers, thus blurring the frontier between proofs and types. We show that each implicative algebra induces a (**Set**-based) tripos, using a construction that is reminiscent from the construction of a realizability tripos from a partial combinatory algebra. Relating this construction with the corresponding constructions in forcing and realizability, we conclude that the class of implicative triposes encompasses all forcing triposes (both intuitionistic and classical), all classical realizability triposes (in the sense of Krivine), and all intuitionistic realizability triposes built from partial combinatory algebras.

Keywords: Realizability, Forcing, Categorical Logic

1. Introduction

In this paper, we introduce the notion of implicative algebra, a simple algebraic structure that is intended to factorize the model-theoretic constructions underlying forcing and realizability, both in intuitionistic and classical logic.

Historically, the method of forcing was introduced by Cohen (1963, 1964) to prove the relative consistency of the negation of the continuum hypothesis with respect to the axioms of set theory. Since then, forcing has been widely investigated, both from a proof-theoretic point of view and from a model-theoretic point of view, and it now constitutes a standard item in the toolbox of set theorists (Jech, 2002). From a model-theoretic point of view, the method of forcing can be understood as a particular way to construct Boolean-valued models of the considered theory (typically: set theory or higher order arithmetic), in which each formula ϕ is interpreted as an element

$$[\![\phi]\!] \in B$$

of a given complete Boolean algebra B. If one is only interested in interpreting intuitionistic theories, one can replace complete Boolean algebras by complete Heyting algebras (HAs), in which

†This work was partly supported by the Uruguayan National Research & Innovation Agency (ANII) under the project "Realizability, Forcing and Quantum Computing," FCE_1_2014_1_104800.

case similar construction methods give us Heyting-valued models, that are essentially equivalent to Kripke (i.e., intuitionistic) forcing or Beth forcing.

As observed by Scott (van Oosten, 2002), there is a strong similarity between (intuitionistic or classical) forcing and the method of realizability, that was introduced by Kleene (1945) to give a constructive semantics to Heyting (i.e., intuitionistic) arithmetic. From a model-theoretic point of view, the method of realizability interprets each closed formula ϕ as a set of realizers

$$[\![\phi]\!] \in \mathfrak{P}(P)$$

where P is a suitable algebra of "programs" (typically, a partial combinatory algebra (PCA)), following the Brouwer–Heyting–Kolmogorov semantics for intuitionistic logic. (Here, the symbol $\mathfrak{P}$ stands for the set-theoretic powerset operator.) Although the method of realizability was initially introduced for intuitionistic first-order arithmetic, it extends to intuitionistic higher order arithmetic and even to intuitionistic Zermelo–Fraenkel set theory (Friedman, 1973; McCarty, 1984; Myhill, 1973).

For a long time, the method of realizability was limited to intuitionistic logic. However, from the mid-90s, Krivine reformulated (Krivine, 2009) the principles of realizability to make them compatible with classical logic, using the correspondence between classical reasoning and control operators discovered by Griffin (1990). Technically, classical realizability departs from intuitionistic realizability by interpreting each formula ϕ not as a set of realizers, but as a set of counter-realizers (a.k.a., a falsity value)

$$[\![\phi]\!] \in \mathfrak{P}(\Pi)$$

where Π is the set of stacks associated to an algebra of classical programs Λ (Krivine, 2011; Streicher, 2013). The corresponding set of realizers (or truth value) is then defined indirectly, as the orthogonal $[\![\phi]\!]^{\perp\!\!\!\perp} \subseteq \Lambda$ of the falsity value $[\![\phi]\!] \subseteq \Pi$ with respect to a particular set of processes $\perp\!\!\!\perp \subseteq \Lambda \times \Pi$ – the pole of the model – that parameterizes the construction. As for intuitionistic realizability, classical realizability extends to higher order arithmetic and even to (classical) Zermelo–Fraenkel set theory (Krivine 2001, 2012), possibly enriched with some weak forms of the axiom of choice.

In spite of their similarity, there is a fundamental difference between forcing and realizability, regarding the treatment of connectives and quantifiers. In forcing, conjunction and disjunction are interpreted as binary meets and joins

$$[\![\phi \wedge \psi]\!] = [\![\phi]\!] \curlywedge [\![\psi]\!] \qquad \text{and} \qquad [\![\phi \vee \psi]\!] = [\![\phi]\!] \curlyvee [\![\psi]\!]$$

(respectively, writing $b \curlywedge c$ and $b \curlyvee c$ the meet and the join of two elements $b, c \in B$), whereas universal and existential quantifications are interpreted by

$$[\![\forall x\, \phi(x)]\!] = \bigcurlywedge_{v \in \mathscr{M}} [\![\phi(v)]\!] \qquad \text{and} \qquad [\![\exists x\, \phi(x)]\!] = \bigcurlyvee_{v \in \mathscr{M}} [\![\phi(v)]\!]\,.$$

So that from the point of view of (intuitionistic or classical) forcing, conjunction and disjunction are just finite forms of universal and existential quantifications. This is definitely not the case in intuitionistic realizability, where conjunctions and disjunctions are interpreted as Cartesian products and direct sums

$$[\![\phi \wedge \psi]\!] = [\![\phi]\!] \times [\![\psi]\!] \qquad \text{and} \qquad [\![\phi \vee \psi]\!] = [\![\phi]\!] + [\![\psi]\!]$$

whereas universal and existential quantifications are still interpreted uniformly[1]:

$$[\![\forall x\, \phi(x)]\!] = \bigcap_{v \in \mathscr{M}} [\![\phi(v)]\!] \qquad \text{and} \qquad [\![\exists x\, \phi(x)]\!] = \bigcup_{v \in \mathscr{M}} [\![\phi(v)]\!]\,.$$

(The situation is slightly more complex in classical realizability, in which existential quantification and disjunction have to be interpreted negatively. But the above picture still holds for conjunctions

https://doi.org/10.1017/S0960129520000079 Published online by Cambridge University Press

and universal quantifications.) In some sense, realizability is more faithful to proof theory, in which proving a universal quantification

$$\frac{\vdash \phi(x)}{\vdash \forall x\, \phi(x)}$$

(i.e., providing a generic proof that holds of all instances of the variable x) is much stronger than proving a (finitary or infinitary) conjunction:

$$\frac{\vdash \phi(t_0) \quad \vdash \phi(t_1) \quad \vdash \phi(t_2) \quad \cdots \quad \vdash \phi(t_n) \quad \cdots}{\vdash \phi(t_0) \land \phi(t_1) \land \phi(t_2) \land \cdots \land \phi(t_n) \land \cdots}$$

(i.e., providing a distinct proof for each instance of the variable x).[2]

But what do have in common an element of a complete HA (or Boolean algebra), a set of realizers (taken in a combinatory algebra (CA) P) or a set of counter-realizers (taken in a set of stacks Π)? The aim of this paper is to show that all these notions of "truth value" pertain to implicative algebras, a surprisingly simple algebraic structure whose most remarkable feature is to use the same set to represent truth values and realizers, thus blurring the frontier between proofs and types. As a matter of fact (Section 2.3), implicative algebras offer a fresh semantic reading of *typing* and *definitional ordering* in terms of *subtyping*, that is now the primitive notion.

However, implicative algebras do not only encompass the various notions of "truth value" underlying forcing and realizability, but they also allow us to factorize the corresponding model-theoretic constructions. For that, we shall place ourselves in the categorical framework of *triposes* (Hyland et al., 1980) that was introduced precisely to compare forcing and realizability in the perspective of constructing categorical models of higher order logic. Intuitively, a tripos is a **Set**-indexed HA of "predicates" $\mathsf{P} : \mathbf{Set}^{\mathrm{op}} \to \mathbf{HA}$ (see Definition 4.7, p. 494) that constitutes a (categorical) model of higher order logic. Triposes can be built from a variety of algebraic structures, such as complete HAs (or Boolean algebra), PCAs, ordered combinatory algebras (OCAs) (van Oosten, 2008), and even abstract Krivine structures (AKSs) (Streicher, 2013). And each tripos can be turned into a topos (i.e., a "Set-like category") via the standard tripos-to-topos construction (Hyland et al., 1980).

As we shall see in Section 4, all the above tripos constructions (as well as the corresponding topos constructions) can be factored through a unique construction, namely, the construction of an *implicative tripos* from a given implicative algebra. Thanks to this factorization, we will be able to characterize forcing in terms of non-determinism (from the point of view of generalized realizers), and we shall prove that classical implicative triposes are equivalent to Krivine's classical realizability triposes.

1.1 Sources of inspiration & related works

The notion of implicative algebra emerged from so many sources of inspirations that it is almost impossible to list them all here. Basically, implicative algebras were designed from a close analysis of the algebraic structure underlying falsity values in Krivine realizability (Krivine, 2009), noticing that this structure is very similar to the one of reducibility candidates (Girard et al., 1989; Parigot, 1997; Tait, 1967; Werner, 1994). Other sources of inspiration are the notion of semantic type in coherence spaces (Miquel, 2000) as well as the notion of fact (or behavior) in phase semantics (Girard, 1987).

The idea of reconstructing λ-terms from implication and infinitary meets came from filter models (Barendregt et al., 1983) that are strongly related to implicative algebras from a technical point of view, although they are not implicative algebras. The same idea appeared implicitly in Streicher's reconstruction of Krivine's tripos (Streicher, 2013) and more explicitly in Ferrer Santos et al. (2017) that introduced many of the ideas that are presented here, but in a slightly different framework, closer to Streicher's. Similar ideas were developed independently by Ruyer, whose applicative lattices (Ruyer, 2006, p. 29) are equivalent to a particular case of implicative structures, namely, to the implicative structures that are compatible with joins (Section 3.8).

1.2 Outline of the paper

The rest of the paper is organized as follows. In Section 2, we introduce the notion of *implicative structure* (as a natural generalization of the notion of complete HA) and show how the elements of such a structure can be used to represent both truth values (or types) and realizers. In Section 3, we introduce the fundamental notion of *separator* (that generalizes the usual notion of filter) as well as the accompanying notion of *implicative algebra*. We show how each separator induces a particular HA (intuitively, the corresponding algebra of propositions) and give a first account on the relationship between forcing and non-determinism (Proposition 3.30, p. 488). Section 4 is devoted to the construction of the *implicative tripos* induced by a particular implicative algebra. We show that implicative triposes encompass many well-known triposes, namely, (intuitionistic and classical) forcing triposes, classical realizability triposes (Streicher, 2013), intuitionistic realizability triposes induced by (total) CAs, and even intuitionistic realizability triposes induced by PCAs (Section 4.7). We also characterize forcing triposes as the non-deterministic implicative triposes (Theorem 4.13, p. 500), and show that classical implicative triposes are equivalent to classical realizability triposes (Theorem 4.19, p. 503).

2. Implicative Structures

2.1 Definition

Definition 2.1 (**Implicative structure**). *An* implicative structure *is a complete meet-semilattice* $(\mathscr{A}, \preccurlyeq)$ *equipped with a binary operation* $(a, b) \mapsto (a \to b)$*, called the* implication of $\mathscr{A}$*, that fulfills the following two axioms:*

(1) Implication is anti-monotonic with respect to its first operand and monotonic with respect to its second operand:

$$\text{if } a' \preccurlyeq a \text{ and } b \preccurlyeq b', \quad \text{then} \quad (a \to b) \preccurlyeq (a' \to b') \qquad (a, a', b, b' \in \mathscr{A})$$

(2) Implication commutes with arbitrary meets on its second operand:

$$a \to \bigcurlywedge_{b \in B} b = \bigcurlywedge_{b \in B} (a \to b) \qquad (a \in \mathscr{A},\ B \subseteq \mathscr{A})$$

Remarks 2.2. (1) By saying that $(\mathscr{A}, \preccurlyeq)$ is a complete meet-semilattice, we mean that every subset of $\mathscr{A}$ has a greatest lower bound (i.e., a *meet*). Such a poset has always a smallest element $\bot = \bigcurlywedge \mathscr{A}$ and a largest element $\top = \bigcurlywedge \varnothing$. More generally, every subset of $\mathscr{A}$ has also a least upper bound (i.e., a *join*), so that a complete meet-semilattice is actually the same as a complete lattice. However, in what follows, we shall mainly be interested in the meet-semilattice structure of implicative structures, so that it is convenient to think that implicative structures are (complete) lattices only by accident.

(2) In the particular case where $B = \varnothing$, axiom (2) states that $(a \to \top) = \top$ for all $a \in \mathscr{A}$. (Recall that $\top = \bigcurlywedge \varnothing$.) In some circumstances, it is desirable to relax this equality, by requiring that axiom (2) holds only for the nonempty subsets B of $\mathscr{A}$. Formally, we call a *quasi-implicative structure* any complete meet-semilattice $\mathscr{A}$ equipped with a binary operation $(a, b) \mapsto (a \to b)$ that fulfills both axioms (1) and (2) of Definition 2.1, the latter being restricted to the case where $B \neq \varnothing$. From this definition, we easily check that a quasi-implicative structure is an implicative structure if and only if $(\top \to \top) = \top$.

2.2 Examples of implicative and quasi-implicative structures

2.2.1 Complete HAs

The most obvious examples of implicative structures are given by complete HAs. Recall that an *HA* is a bounded lattice $(H, \preccurlyeq)$ equipped with a binary operation $(a, b) \mapsto (a \to b)$ (Heyting's implication) characterized by the adjunction

$$(c \curlywedge a) \preccurlyeq b \quad \text{iff} \quad c \preccurlyeq (a \to b) \qquad (a, b, c \in H)$$

Historically, HAs have been introduced as the intuitionistic counterpart of Boolean algebras, and they can be used to interpret intuitionistic provability the same way as Boolean algebras can be used to interpret classical provability. In this framework, conjunction and disjunction are interpreted by binary meets and joins, whereas implication is interpreted by the operation $a \to b$. This interpretation validates all reasoning principles of intuitionistic propositional logic, in the sense that every propositional formula that is intuitionistically valid is denoted by the truth value $\top$.

Boolean algebras are the HAs $(H, \preccurlyeq)$ in which negation is involutive, that is, $\neg\neg a = a$ for all $a \in H$, where negation is defined by $\neg a := (a \to \bot)$. Boolean algebras more generally validate all classical reasoning principles, such as the law of excluded middle $(a \curlyvee \neg a = \top)$ or Peirce's law $((((a \to b) \to a) \to a) = \top)$.

An HA (or Boolean algebra) is *complete* when the underlying lattice is complete. In a complete HA, the interpretation depicted above naturally extends to all formulas of predicate logic, by interpreting universal and existential quantifications as meets and joins of families of truth values indexed over a fixed nonempty set. Again, this (extended) interpretation validates all reasoning principles of intuitionistic predicate logic. It is easy to check that in a complete HA, Heyting's implication fulfills both axioms (1) and (2) of Definition 2.1, so that:

Fact 2.3. Every complete HA is an implicative structure.

In what follows, we shall say that an implicative structure $(\mathscr{A}, \preccurlyeq, \to)$ is a *complete HA* when the underlying lattice $(\mathscr{A}, \preccurlyeq)$ is an (complete) HA, and when the accompanying implication $(a, b) \mapsto (a \to b)$ is Heyting's implication.

2.2.2 Dummy implicative structures

Unlike Heyting's implication, the implication of an implicative structure $\mathscr{A}$ is in general not determined by the ordering of $\mathscr{A}$, and several implicative structures can be defined upon the very same complete lattice structure:

Example 2.4 (Dummy implicative structures). Let $(L, \preccurlyeq)$ be a complete lattice. There are at least two distinct ways to define a dummy implication $a \to b$ on L that fulfills the axioms (1) and (2) of Definition 2.1:

(1) Put $(a \to b) := b$ for all $a, b \in L$.
(2) Put $(a \to b) := \top$ for all $a, b \in L$.

Each of these two definitions induces an implicative structure on the top of the complete lattice $(L, \preccurlyeq)$. From the point of view of logic, these two examples are definitely meaningless, but they will be useful as a source of counter examples.

2.2.3 Quasi-implicative structures induced by partial applicative structures

Another important source of examples is given by the structures underlying intuitionistic realizability (van Oosten, 2008). Recall that a *partial applicative structure* (PAS) is a nonempty set P

https://doi.org/10.1017/S0960129520000079 Published online by Cambridge University Press

equipped with a partial binary operation $(\cdot) : P \times P \rightharpoonup P$, called *application*. Such an operation naturally induces a (total) binary operation $(a, b) \mapsto (a \to b)$ on the subsets of P, called *Kleene's implication*, that is defined for all $a, b \subseteq P$ by

$$a \to b \ := \ \{z \in P : \forall x \in a,\ z \cdot x \downarrow\in b\}$$

(where $z \cdot x \downarrow\in b$ means that $z \cdot x$ is defined and belongs to b). We easily check that:

Fact 2.5. Given a PAS $(P, \cdot)$:

(1) The complete lattice $(\mathfrak{P}(P), \subseteq)$ equipped with Kleene's implication $a \to b$ is a quasi-implicative structure (in the sense of Remark 2.2 (2)).
(2) The quasi-implicative structure $(\mathfrak{P}(P), \subseteq, \to)$ is an implicative structure if and only if the underlying operation of application $(x, y) \mapsto x \cdot y$ is total.

We shall come back to this example in Section 2.7.1.

A variant of the above construction consists to replace the subsets of P by the *partial equivalence relations* (PERs) over P, that is, by the binary relations on P that are both symmetric and transitive – but not reflexive in general. The set of PERs over P, written $\mathrm{PER}(P)$, is clearly closed under arbitrary intersection (in the sense of relations), so that the poset $(\mathrm{PER}(P), \subseteq)$ is a complete meet-semilattice. Kleene's implication naturally extends to PERs, by associating to all $a, b \in \mathrm{PER}(P)$ the relation $(a \to_2 b) \in \mathrm{PER}(P)$ defined by

$$a \to_2 b \ := \ \{(z_1, z_2) \in P^2 : \forall (x_1, x_2) \in a,\ (z_1 \cdot x_1, z_2 \cdot x_2) \downarrow\in b\}\,.$$

Again

Fact 2.6. Given a PAS $(P, \cdot)$:

(1) The complete lattice $(\mathrm{PER}(P), \subseteq)$ equipped with Kleene's implication $a \to_2 b$ is a quasi-implicative structure (in the sense of Remark 2.2 (2)).
(2) The quasi-implicative structure $(\mathrm{PER}(P), \subseteq, \to_2)$ is an implicative structure if and only if the underlying operation of application $(x, y) \mapsto x \cdot y$ is total.

Remark 2.7. The reader is invited to check that the last two examples of (quasi-) implicative structures fulfill the following additional axiom:

$$\Big(\bigcurlyvee_{a \in A} a\Big) \to b \ = \ \bigcurlywedge_{a \in A} (a \to b) \qquad \text{(for all } A \subseteq \mathscr{A} \text{ and } b \in \mathscr{A}\text{)}$$

In what follows, we shall see that this axiom – that already holds in complete HAs – is characteristic from the implicative structures coming from intuitionistic realizability or from (intuitionistic or classical) forcing. (On the other hand, this axiom does not hold in the implicative structures coming from classical realizability, except in the degenerate case of forcing.) We shall come back to this point in Section 3.8.

2.2.4 Quasi-implicative structures of reducibility candidates

Other examples of quasi-implicative structures are given by the various notions of *reducibility candidates* (Girard et al., 1989; Parigot, 1997; Tait, 1967; Werner, 1994) that are used to prove strong normalization. Let us consider, for instance, the case of Tait's saturated sets (Tait, 1967).

Recall that a set S of (possibly open) λ-terms is *saturated* (in the sense of Tait) when it fulfills the following three criteria:

https://doi.org/10.1017/S0960129520000079 Published online by Cambridge University Press

(*i*) $S \subseteq \mathrm{SN}$, where SN is the set of all strongly normalizing terms.
(*ii*) If x is a variable and if $u_1, \ldots, u_n \in \mathrm{SN}$, then $xu_1 \cdots u_n \in S$.
(*iii*) If $t\{x := u_0\}u_1 \cdots u_n \in S$ and $u_0 \in \mathrm{SN}$, then $(\lambda x\,.\,t)u_0u_1 \cdots u_n \in S$.

The set of all saturated sets, written **SAT**, is closed under Kleene's implication, in the sense that for all $S, T \in \mathbf{SAT}$ one has $S \to T = \{t : \forall u \in S,\ tu \in T\} \in \mathbf{SAT}$. Again

Fact 2.8. The triple $(\mathbf{SAT}, \subseteq, \to)$ is a quasi-implicative structure.

The reader is invited to check that the same holds if we replace Tait's saturated sets by other notions of reducibility candidates, such as Girard's reducibility candidates (Girard et al., 1989) or Parigot's reducibility candidates (Parigot, 1997). Let us mention that in each case, we only get a *quasi*-implicative structure, in which we have $(\top \to \top) \neq \top$. The reason is that full implicative structures (which come with the equation $(\top \to \top) = \top$) are actually expressive enough to interpret the full λ-calculus (see Section 2.4), so that they are incompatible with the notion of (weak or strong) normalization.

2.2.5 Implicative structures of classical realizability

The final example – which is the main motivation of this work – is given by classical realizability, as introduced by Krivine (2001, 2003, 2009, 2011, 2012). Basically, classical realizability takes place in a structure of the form $(\Lambda, \Pi, \cdot, \perp\!\!\!\perp)$ where

- Λ is a set whose elements are called *terms* or *realizers*;
- Π is a set whose elements are called *stacks* or *counter-realizers*;
- $(\cdot) : \Lambda \times \Pi \to \Pi$ is a binary operation for *pushing* a term onto a stack;
- $\perp\!\!\!\perp \subseteq \Lambda \times \Pi$ is a binary relation between $\Lambda \times \Pi$, called the *pole*.

(Krivine's classical realizability structures actually contain many other ingredients (cf. Section 2.7.2) that we do not need for now.) From such a quadruple $(\Lambda, \Pi, \cdot, \perp\!\!\!\perp)$, we let

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

writing $a^{\perp\!\!\!\perp} := \{t \in \Lambda : \forall \pi \in a,\ (t, \pi) \in \perp\!\!\!\perp\} \in \mathfrak{P}(\Lambda)$ the *orthogonal* of the set $a \in \mathfrak{P}(\Pi)$ with respect to the pole $\perp\!\!\!\perp \subseteq \Lambda \times \Pi$. Again, it is easy to check that

Fact 2.9. The triple $(\mathscr{A}, \preccurlyeq, \to)$ is an implicative structure.

Remark 2.10. The reader is invited to check that Krivine's implication $a \to b = a^{\perp\!\!\!\perp} \cdot b$ fulfills the two additional axioms

$$\Big(\bigcurlywedge_{a\in A} a\Big) \to b \ =\ \bigcurlyvee_{a\in A} (a \to b) \qquad \text{and} \qquad a \to \Big(\bigcurlyvee_{b\in B} b\Big) \ =\ \bigcurlyvee_{b\in B} (a \to b)$$

for all $a, b \in \mathscr{A}$, $A, B \subseteq \mathscr{A}$, $A, B \neq \varnothing$. It is worth to notice that these extra properties are almost never used in classical realizability, thus confirming that only the properties of meets really matter in such a structure.

We shall come back to this example in Section 2.7.2.

https://doi.org/10.1017/S0960129520000079 Published online by Cambridge University Press

2.3 Viewing truth values as generalized realizers: a manifesto

Intuitively, an implicative structure $(\mathscr{A}, \preccurlyeq, \to)$ represents a semantic type system in which the ordering $a \preccurlyeq b$ expresses the relation of subtyping, whereas the operation $a \to b$ represents the arrow type construction. From the point of view of logic, it is convenient to think of the elements of $\mathscr{A}$ as truth values according to some notion of realizability, that is, as sets of realizers enjoying particular closure properties.

Following this intuition, we can always view an actual realizer t as a truth value, namely, as the smallest truth value that contains t. This truth value, written as $[t]$ and called the *principal type* of the realizer t, is naturally defined as the meet of all truth values containing t as an element. Through the correspondence $t \mapsto [t]$[3], the membership relation $t \in a$ rephrases in term of subtyping as $[t] \preccurlyeq a$, so that we can actually manipulate realizers as if they were truth values.

But the distinctive feature of implicative structures is that they allow us to proceed the other way around, that is, to manipulate *all* truth values as if they were realizers. Technically, this is due to the fact that the two fundamental operations of the λ-calculus – application and λ-abstraction – can be lifted to the level of truth values (Section 2.4). Of course, such a possibility definitely blurs the distinction between the particular truth values that represent actual realizers (the principal types) and the other ones. So that the framework of implicative structures actually leads us to perform a surprising identification, between the notion of truth value and the notion of realizer, now using the latter notion in a generalized sense.

Conceptually, this identification relies on the idea that every element $a \in \mathscr{A}$ may also be viewed as a generalized realizer, namely, as the realizer whose principal type is a itself (by convention). In this way, the element a, when viewed as a generalized realizer, is not only a realizer of a, but it is more generally a realizer of any truth value b such that $a \preccurlyeq b$. Of course, there is something puzzling in the idea that truth values are their own (generalized) realizers, since this implies that any truth value is realized, at least by itself. In particular, the bottom truth value $\bot \in \mathscr{A}$, when viewed as a generalized realizer, is so strong that it actually realizes any truth value. But this paradox only illustrates another aspect of implicative structures, which is that they do not come with an absolute criterion of consistency. To introduce such a "criterion of consistency," we shall need to introduce the notion of *separator* (Section 3), which plays the very same role as the notion of filter in HA (or Boolean algebra).

Due to the identification between truth values and (generalized) realizers, the partial ordering $a \preccurlyeq b$ can be given different meanings depending on whether we consider the elements a and b as truth values or as generalized realizers. For instance, if we think of a and b both as truth values, then the ordering $a \preccurlyeq b$ is simply the relation of subtyping. And if we think of a as a generalized realizer and of b as a truth value, then the relation $a \preccurlyeq b$ is nothing but the realizability relation ("a realizes b"). But if we now think of both elements a and b as generalized realizers, then the relation $a \preccurlyeq b$ means that the (generalized) realizer a is at least as powerful as b, in the sense that a realizes any truth value c that is realized by b. In forcing, we would express it by saying that a is a *stronger condition* than b. And in domain theory, we would naturally say that a is *more defined* than b, which we would write $a \sqsupseteq b$.

The latter example is important, since it shows that when thinking of the elements of $\mathscr{A}$ as generalized realizers rather than as truth values, then the reverse ordering $a \succcurlyeq b$ is conceptually similar to the definitional ordering in the sense of Scott. Note that this point of view is consistent with the fact that the theory of implicative structures (see Definition 2.1 and Remark 2.2 (1)) is built around meets that precisely correspond to joins from the point of view of definitional (i.e., Scott) ordering. In what follows, we shall refer to the relation $a \preccurlyeq b$ as the *logical ordering*, whereas the symmetric relation $b \succcurlyeq a$ (which we shall sometimes write $b \sqsubseteq a$) will be called the *definitional ordering*.

Using these intuitions as guidelines, it is now easy to lift all the constructions of the λ-calculus to the level of truth values in an arbitrary implicative structure.

https://doi.org/10.1017/S0960129520000079 Published online by Cambridge University Press

2.4 Interpreting λ-terms

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

Definition 2.11 (**Application**). *Given two points $a, b \in \mathscr{A}$, we call the* application of a to b *and write ab the element of $\mathscr{A}$ that is defined by*

$$ab \;:=\; \bigcurlywedge\{c \in \mathscr{A} : a \preccurlyeq (b \to c)\}\,.$$

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

Thinking in terms of definitional ordering rather than in terms of logical ordering, this definition expresses that ab is the join of all $c \in \mathscr{A}$ such that the implication $b \to c$ (which is analogous to a *step function*) is a lower approximation of a:

$$ab \;:=\; \bigsqcup\{c \in \mathscr{A} : (b \to c) \sqsubseteq a\}\,.$$

Proposition 2.12 (**Properties of application**). *For all $a, a', b, b' \in \mathscr{A}$,*

(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) $ab = \min\{c \in \mathscr{A} : a \preccurlyeq (b \to c)\}$ (Minimum)
(5) $ab \preccurlyeq c$, iff $a \preccurlyeq (b \to c)$ (Adjunction)

Proof. For all $a, b \in \mathscr{A}$, we write $U_{a,b} = \{c \in \mathscr{A} : a \preccurlyeq (b \to c)\}$, so that $ab := \bigcurlywedge U_{a,b}$. (The set $U_{a,b}$ is upwards closed, from the variance of implication.)

(1) If $a \preccurlyeq a'$ and $b \preccurlyeq b'$, then $U_{a',b'} \subseteq U_{a,b}$ (from the variance of implication); hence, we get $ab = \bigcurlywedge U_{a,b} \preccurlyeq \bigcurlywedge U_{a',b'} = a'b'$.
(2) It is clear that $b \in U_{a\to b,a}$, hence $(a \to b)a = \bigcurlywedge U_{a\to b,a} \preccurlyeq b$.
(3) We have $(b \to ab) = (b \to \bigcurlywedge U_{a,b}) = \bigcurlywedge_{c\in U_{a,b}}(b \to c) \succcurlyeq a$, from the definition of $U_{a,b}$.
(4) From (3), it is clear that $ab \in U_{a,b}$, hence $ab = \min(U_{a,b})$.
(5) Assuming that $ab \preccurlyeq c$, we get $a \preccurlyeq (b \to ab) \preccurlyeq (b \to c)$ from (3). Conversely, assuming that $a \preccurlyeq (b \to c)$, we have $c \in U_{a,b}$ and thus $ab = \bigcurlywedge U_{a,b} \preccurlyeq c$. □

Corollary 2.13 (**Application in a complete HA**). *In a complete HA $(H, \preccurlyeq, \to)$, application is characterized by $ab = a \curlywedge b$ for all $a, b \in H$.*

Proof. For all $c \in \mathscr{A}$, we have $ab \preccurlyeq c$ iff $a \preccurlyeq (b \to c)$ by Proposition 2.12 (5). But from Heyting's adjunction, we also have $a \preccurlyeq (b \to c)$ iff $a \curlywedge b \preccurlyeq c$. Hence, $ab \preccurlyeq c$ iff $a \curlywedge b \preccurlyeq c$ for all $c \in \mathscr{A}$, and thus $ab = a \curlywedge b$. □

Corollary 2.14 (**Application in a total applicative structure**). *In the implicative structure $(\mathfrak{P}(P), \subseteq, \to)$ induced by a total applicative structure $(P, \cdot)$ (cf. Fact 2.5, p. 463), application is characterized by $ab = \{x \cdot y : x \in a,\ y \in b\}$ for all $a, b \in \mathfrak{P}(P)$.*

Proof. Let $a \cdot b = \{x \cdot y : x \in a,\ \ y \in b\}$. It is clear that for all $c \in \mathfrak{P}(P)$, we have $a \cdot b \subseteq c$ iff $a \subseteq (b \to c)$. Therefore, $a \cdot b = ab$, by adjunction. □

https://doi.org/10.1017/S0960129520000079 Published online by Cambridge University Press

Definition 2.15 (**Abstraction**). *Given an arbitrary function* $f : \mathscr{A} \to \mathscr{A}$, *we write* $\boldsymbol{\lambda} f$ *the element of* $\mathscr{A}$ *defined by*

$$\boldsymbol{\lambda} f \ := \ \bigwedge_{a \in \mathscr{A}} (a \to f(a)) \,.$$

(Note that we do not assume that the function f is monotonic.)

Again, if we think in terms of definitional ordering rather than in terms of logical ordering, then it is clear that this definition expresses that $\boldsymbol{\lambda} f$ is the join of all the step functions of the form $a \to f(a)$, where $a \in \mathscr{A}$:

$$\boldsymbol{\lambda} f \ := \ \bigsqcup_{a \in \mathscr{A}} (a \to f(a)) \,.$$

Proposition 2.16 (**Properties of abstraction**). *For all* $f, g : \mathscr{A} \to \mathscr{A}$ *and* $a \in \mathscr{A}$,

(1) *if* $f(a) \preccurlyeq g(a)$ *for all* $a \in \mathscr{A}$, *then* $\boldsymbol{\lambda} f \preccurlyeq \boldsymbol{\lambda} g$ *(Monotonicity)*
(2) $(\boldsymbol{\lambda} f)a \preccurlyeq f(a)$ *(β-reduction)*
(3) $a \preccurlyeq \boldsymbol{\lambda}(b \mapsto ab)$ *(η-expansion)*

Proof.

(1) Obvious from the variance of implication.
(2) From the definition of $\boldsymbol{\lambda} f$, we have $\boldsymbol{\lambda} f \preccurlyeq (a \to f(a))$. Applying Proposition 2.12 (5), we get $(\boldsymbol{\lambda} f)a \preccurlyeq f(a)$.
(3) Follows from Proposition 2.12 (3), taking the meet for all $b \in \mathscr{A}$. □

We call a λ*-term with parameters in* $\mathscr{A}$ any λ-term (possibly) enriched with constants taken in the set $\mathscr{A}$ – the “parameters.” Such enriched λ-terms are equipped with the usual notions of β- and η-reduction, considering parameters as inert constants.

To every closed λ-term t with parameters in $\mathscr{A}$, we associate an element of $\mathscr{A}$, written as $t^{\mathscr{A}}$ and defined by induction on the size of t by

$$\begin{aligned} a^{\mathscr{A}} &:= a && (\text{if } a \in \mathscr{A}) \\ (tu)^{\mathscr{A}} &:= t^{\mathscr{A}} u^{\mathscr{A}} && (\text{application in } \mathscr{A}) \\ (\lambda x \,.\, t)^{\mathscr{A}} &:= \boldsymbol{\lambda}(a \mapsto (t\{x := a\})^{\mathscr{A}}) && (\text{abstraction in } \mathscr{A}) \end{aligned}$$

Proposition 2.17 (**Monotonicity of substitution**). *For each* λ*-term* t *with free variables* $x_1, \ldots, x_k$ *and for all parameters* $a_1 \preccurlyeq a'_1, \ldots, a_k \preccurlyeq a'_k$, *we have*

$$(t\{x_1 := a_1, \ldots, x_k := a_k\})^{\mathscr{A}} \ \preccurlyeq \ (t\{x_1 := a'_1, \ldots, x_k := a'_k\})^{\mathscr{A}}$$

(where $t\{x_1 := a_1, \ldots, x_k := a_k\}$ *denotes a simultaneous substitution).*

Proof. By induction on t, using Propositions 2.12 (1) and 2.16 (1). □

Proposition 2.18 (β **and** η). *For all closed* λ*-terms* t *and* u *with parameters in* $\mathscr{A}$,

(1) *if* $t \twoheadrightarrow_{\beta} u$, *then* $t^{\mathscr{A}} \preccurlyeq u^{\mathscr{A}}$;
(2) *if* $t \twoheadrightarrow_{\eta} u$, *then* $t^{\mathscr{A}} \succcurlyeq u^{\mathscr{A}}$;

Proof. Obvious from Propositions 2.16 (2), (3) and 2.17. □

https://doi.org/10.1017/S0960129520000079 Published online by Cambridge University Press

Remark 2.19. It is important to observe that an implicative structure is in general *not* a denotational model of the λ-calculus, since the inequalities of Proposition 2.18 are in general not equalities, as shown in Example 2.20. Let us recall that in a denotational model D of the λ-calculus (where $t =_{\beta\eta} u$ implies $t^D = u^D$), the interpretation function $t \mapsto t^D$ is either trivial, or injective on $\beta\eta$-normal forms. This is no longer the case in implicative structures, where some $\beta\eta$-normal terms may collapse, while others do not. We shall come back to this problem in Section 2.7.

Example 2.20 (**Dummy implicative structure**). Let us consider the dummy implicative structure (cf. Example 2.4 (2)) constructed on the top of a complete lattice $(L, \preccurlyeq)$ by putting $a \to b := \top$ for all $a, b \in \mathscr{A}$. In this structure, we observe that

- $ab = \curlywedge\{c \in \mathscr{A} : a \preccurlyeq (b \to c)\} = \curlywedge \mathscr{A} = \bot$ for all $a, b \in \mathscr{A}$;
- $\boldsymbol{\lambda} f = \curlywedge_{a \in \mathscr{A}} (a \to f(a)) = \top$ for all functions $f : \mathscr{A} \to \mathscr{A}$.

So that for any closed λ-term t, we immediately get

$$t^{\mathscr{A}} \;=\; \begin{cases} \top & \text{if } t \text{ is an abstraction} \\ \bot & \text{if } t \text{ is an application} \end{cases}$$

(The reader is invited to check that the above characterization is consistent with the inequalities of Proposition 2.18.) In particular, letting $\mathbf{I} := \lambda x\,.\,x$, we observe that

- $\mathbf{I}\,\mathbf{I} \to_\beta \mathbf{I}$, but $(\mathbf{I}\,\mathbf{I})^{\mathscr{A}}\ (= \bot) \neq \mathbf{I}^{\mathscr{A}}\ (= \top)$;
- $\lambda x\,.\,\mathbf{I}\,\mathbf{I}\,x \to_\eta \mathbf{I}\,\mathbf{I}$, but $(\lambda x\,.\,\mathbf{I}\,\mathbf{I}\,x)^{\mathscr{A}}\ (= \top) \neq (\mathbf{I}\,\mathbf{I})^{\mathscr{A}}\ (= \bot)$.

Proposition 2.21 (**λ-terms in a complete HA**). *If $(\mathscr{A}, \preccurlyeq, \to)$ is a complete HA, then for all (pure) λ-terms with free variables $x_1, \ldots, x_k$ and for all parameters $a_1, \ldots, a_k \in \mathscr{A}$, we have*

$$(t\{x_1 := a_1, \ldots, x_k := a_k\})^{\mathscr{A}} \;\succcurlyeq\; a_1 \curlywedge \cdots \curlywedge a_k\,.$$

In particular, for all closed λ-terms t, we have $t^{\mathscr{A}} = \top$.

Proof. Let us write $\vec{x} = x_1, \ldots, x_k$ and $\vec{a} = a_1, \ldots, a_k$. We reason by induction on t, distinguishing the following cases:

- $t = x$ (variable). This case is obvious.
- $t = t_1 t_2$ (application). In this case, we have

$$\begin{aligned} (t\{\vec{x} := \vec{a}\})^{\mathscr{A}} &= (t_1\{\vec{x} := \vec{a}\})^{\mathscr{A}} (t_2\{\vec{x} := \vec{a}\})^{\mathscr{A}} \\ &= (t_1\{\vec{x} := \vec{a}\})^{\mathscr{A}} \curlywedge (t_2\{\vec{x} := \vec{a}\})^{\mathscr{A}} && \text{(by Corollary 2.13)} \\ &\succcurlyeq a_1 \curlywedge \cdots \curlywedge a_k && \text{(by IH)} \end{aligned}$$

- $t = \lambda x_0\,.\,t_0$ (abstraction). In this case, we have

$$\begin{aligned} (t\{\vec{x} := \vec{a}\})^{\mathscr{A}} &= \mathop{\curlywedge}_{a_0 \in \mathscr{A}} \big(a_0 \to (t_0\{x_0 := a_0, \vec{x} := \vec{a}\})^{\mathscr{A}}\big) \\ &\succcurlyeq \mathop{\curlywedge}_{a_0 \in \mathscr{A}} \big(a_0 \to a_0 \curlywedge a_1 \curlywedge \cdots \curlywedge a_k\big) && \text{(by IH)} \\ &\succcurlyeq a_1 \curlywedge \cdots \curlywedge a_k \end{aligned}$$

using the relation $b \preccurlyeq (a \to a \curlywedge b)$ (for all $a, b \in \mathscr{A}$) in the last inequality. □

https://doi.org/10.1017/S0960129520000079 Published online by Cambridge University Press

Remark 2.22. The above result is reminiscent from the fact that in forcing (in the sense of Kripke or Cohen), all (intuitionistic or classical) tautologies are interpreted by the top element (i.e., the weakest condition). This is clearly no longer the case in (intuitionistic or classical) realizability, as well as in implicative structures more generally.

2.5 Semantic typing

Any implicative structure $\mathscr{A} = (\mathscr{A}, \preccurlyeq, \to)$ naturally induces a *semantic type system* whose types are the elements of $\mathscr{A}$.

In this framework, a *typing context* is a finite (unordered) list $\Gamma = x_1 : a_1, \ldots, x_n : a_n$, where $x_1, \ldots, x_n$ are pairwise distinct λ-variables and where $a_1, \ldots, a_n \in \mathscr{A}$. Thinking of the elements of $\mathscr{A}$ as realizers rather than as types, we may also view every typing context $\Gamma = x_1 : a_1, \ldots, x_n : a_n$ as the substitution $\Gamma = x_1 := a_1, \ldots, x_n := a_n$.

Given a typing context $\Gamma = x_1 : a_1, \ldots, x_n : a_n$, we write $\mathrm{dom}(\Gamma) = \{x_1, \ldots, x_n\}$ its *domain*, and the concatenation Γ, Γ' of two typing contexts Γ and Γ' is defined as expected, provided $\mathrm{dom}(\Gamma) \cap \mathrm{dom}(\Gamma') = \varnothing$. Given two typing contexts Γ and Γ', we write $\Gamma' \preccurlyeq \Gamma$ when for every declaration $(x : a) \in \Gamma$, there is a type $b \preccurlyeq a$ such that $(x : b) \in \Gamma'$. (Note that the relation $\Gamma' \preccurlyeq \Gamma$ implies that $\mathrm{dom}(\Gamma') \supseteq \mathrm{dom}(\Gamma)$.)

Given a typing context Γ, a λ-term t with parameters in $\mathscr{A}$, and an element $a \in \mathscr{A}$, we define the (semantic) typing judgment $\Gamma \vdash t : a$ as the following shorthand:

$$\Gamma \vdash t : a \quad :\Leftrightarrow \quad FV(t) \subseteq \mathrm{dom}(\Gamma) \ \text{ and } \ (t[\Gamma])^{\mathscr{A}} \preccurlyeq a$$

(using Γ as a substitution in the right-hand side inequality). From this semantic definition of typing, we easily deduce that

Proposition 2.23 (**Semantic typing rules**). *For all typing contexts Γ, Γ', for all λ-terms t, u with parameters in $\mathscr{A}$ and for all $a, a', b \in \mathscr{A}$, the following “semantic typing rules” are valid*

- *if* $(x : a) \in \Gamma$, *then* $\Gamma \vdash x : a$; *(Axiom)*
- $\Gamma \vdash a : a$; *(Parameter)*
- *if* $\Gamma \vdash t : a$ *and* $a \preccurlyeq a'$, *then* $\Gamma \vdash t : a'$; *(Subsumption)*
- *if* $\Gamma' \preccurlyeq \Gamma$ *and* $\Gamma \vdash t : a$, *then* $\Gamma' \vdash t : a$; *(Context subsumption)*
- *if* $FV(t) \subseteq dom(\Gamma)$, *then* $\Gamma \vdash t : \top$; *($\top$-intro)*
- *if* $\Gamma, x : a \vdash t : b$, *then* $\Gamma \vdash \lambda x . t : a \to b$; *($\to$-intro)*
- *if* $\Gamma \vdash t : a \to b$ *and* $\Gamma \vdash u : a$, *then* $\Gamma \vdash tu : b$. *($\to$-elim)*

Moreover, for every family $(a_i)_{i \in I}$ of elements of $\mathscr{A}$ indexed by a set (or a class) I,

$$\text{if} \quad \Gamma \vdash t : a_i \quad (\text{for all } i \in I), \quad \text{then} \quad \Gamma \vdash t : \bigcurlywedge_{i \in I} a_i \qquad \textit{(Generalization)}$$

Proof. Axiom, Parameter, Subsumption, $\top$-intro: Obvious.

Context subsumption: Follows from Proposition 2.17 (monotonicity of substitution).

$\to$-intro: Let us assume that $FV(t) \subseteq \mathrm{dom}(\Gamma, x := a)$ and $(t[\Gamma, x := a])^{\mathscr{A}} \preccurlyeq b$. It is clear that $FV(\lambda x . t) \subseteq \mathrm{dom}(\Gamma)$ and $x \notin \mathrm{dom}(\Gamma)$, so that

$$\begin{aligned}((\lambda x . t)[\Gamma])^{\mathscr{A}} = (\lambda x . t[\Gamma])^{\mathscr{A}} &= \bigcurlywedge_{a_0 \in \mathscr{A}} \Big(a_0 \to \big(t[\Gamma, x := a_0]\big)^{\mathscr{A}}\Big) \\ &\preccurlyeq a \to \big(t[\Gamma, x := a]\big)^{\mathscr{A}} \preccurlyeq a \to b\,.\end{aligned}$$

https://doi.org/10.1017/S0960129520000079 Published online by Cambridge University Press

$\to$-*elim*: Let us assume that $FV(t), FV(u) \subseteq \mathrm{dom}(\Gamma)$, $(t[\Gamma])^{\mathscr{A}} \preccurlyeq a \to b$, and $(u[\Gamma])^{\mathscr{A}} \preccurlyeq a$. It is clear that $FV(tu) \subseteq \mathrm{dom}(\Gamma)$, and from Proposition 2.12 (2), we get

$$\big((tu)[\Gamma]\big)^{\mathscr{A}} = \big(t[\Gamma]\big)^{\mathscr{A}} \big(u[\Gamma]\big)^{\mathscr{A}} \preccurlyeq (a \to b)a \preccurlyeq b\,.$$

Generalization: Obvious, by taking the meet. □

2.6 Some combinators

Let us now consider the following combinators (using Curry's notation):

$$\begin{aligned} \mathbf{I} &= \lambda x\,.\,x & \mathbf{K} &= \lambda xy\,.\,x \\ \mathbf{B} &= \lambda xyz\,.\,x(yz) & \mathbf{W} &= \lambda xy\,.\,xyy \\ \mathbf{C} &= \lambda xyz\,.\,xzy & \mathbf{S} &= \lambda xyz\,.\,xz(yz) \end{aligned}$$

It is well known that in any polymorphic type assignment system, the above λ-terms can be given the following (principal) types:

$$\begin{aligned} \mathbf{I} &: \forall\alpha\,(\alpha \to \alpha) \\ \mathbf{B} &: \forall\alpha\,\forall\beta\,\forall\gamma\,((\alpha \to \beta) \to (\gamma \to \alpha) \to \gamma \to \beta) \\ \mathbf{K} &: \forall\alpha\,\forall\beta\,(\alpha \to \beta \to \alpha) \\ \mathbf{C} &: \forall\alpha\,\forall\beta\,\forall\gamma\,((\alpha \to \beta \to \gamma) \to \beta \to \alpha \to \gamma) \\ \mathbf{W} &: \forall\alpha\,\forall\beta\,((\alpha \to \alpha \to \beta) \to \alpha \to \beta) \\ \mathbf{S} &: \forall\alpha\,\forall\beta\,\forall\gamma\,((\alpha \to \beta \to \gamma) \to (\alpha \to \beta) \to \alpha \to \gamma) \end{aligned}$$

Turning the above syntactic type judgments into semantic type judgments (Section 2.5) using the typing rules of Proposition 2.23, it is clear that in any implicative structure $\mathscr{A} = (\mathscr{A}, \preccurlyeq, \to)$, we have the following inequalities:

$$\mathbf{I}^{\mathscr{A}} \preccurlyeq \bigwedge_{a \in \mathscr{A}} (a \to a), \qquad \mathbf{K}^{\mathscr{A}} \preccurlyeq \bigwedge_{a,b \in \mathscr{A}} (a \to b \to a),$$
$$\mathbf{S}^{\mathscr{A}} \preccurlyeq \bigwedge_{a,b,c \in \mathscr{A}} ((a \to b \to c) \to (a \to b) \to a \to c), \qquad \text{etc.}$$

A remarkable property of implicative structures is that the above inequalities are actually equalities, for each one of the six combinators **I**, **B**, **K**, **C**, **W**, and **S**:

Proposition 2.24. *In any implicative structure $(\mathscr{A}, \preccurlyeq, \to)$, we have*

$$\begin{aligned} \mathbf{I}^{\mathscr{A}} &= \bigwedge_{a \in \mathscr{A}} (a \to a) & \mathbf{B}^{\mathscr{A}} &= \bigwedge_{a,b,c \in \mathscr{A}} ((a \to b) \to (c \to a) \to c \to b) \\ \mathbf{K}^{\mathscr{A}} &= \bigwedge_{a,b \in \mathscr{A}} (a \to b \to a) & \mathbf{C}^{\mathscr{A}} &= \bigwedge_{a,b,c \in \mathscr{A}} ((a \to b \to c) \to b \to a \to c) \\ \mathbf{W}^{\mathscr{A}} &= \bigwedge_{a,b \in \mathscr{A}} ((a \to a \to b) \to a \to b) & \mathbf{S}^{\mathscr{A}} &= \bigwedge_{a,b,c \in \mathscr{A}} ((a \to b \to c) \to b \to a \to c) \end{aligned}$$

Proof. Indeed, we have

https://doi.org/10.1017/S0960129520000079 Published online by Cambridge University Press

- $\mathbf{I}^{\mathscr{A}} = (\lambda x\,.\,x)^{\mathscr{A}} = \bigwedge_{a\in\mathscr{A}} (a \to a)$ (by definition)
- $\mathbf{K}^{\mathscr{A}} = (\lambda xy\,.\,x)^{\mathscr{A}} = \bigwedge_{a\in\mathscr{A}} \Big(a \to \bigwedge_{b\in\mathscr{A}} (b \to a)\Big) = \bigwedge_{a,b\in\mathscr{A}} (a \to b \to a)$ (by axiom (2))
- By semantic typing, it is clear that

$$\mathbf{S}^{\mathscr{A}} \;=\; \big(\lambda xyz\,.\,xz(yz)\big)^{\mathscr{A}} \;\preccurlyeq\; \bigwedge_{a,b,c\in\mathscr{A}} ((a \to b \to c) \to (a \to b) \to a \to c).$$

Conversely, we have

$$\begin{aligned}
&\bigwedge_{a,b,c\in\mathscr{A}} ((a \to b \to c) \to (a \to b) \to a \to c)\\
&\preccurlyeq \bigwedge_{a,d,e\in\mathscr{A}} \big((a \to ea \to da(ea)) \to (a \to ea) \to a \to da(ea)\big)\\
&\preccurlyeq \bigwedge_{a,d,e\in\mathscr{A}} \big((a \to da) \to e \to a \to da(ea)\big) && \text{(using Proposition 2.12 (3) twice)}\\
&\preccurlyeq \bigwedge_{a,d,e\in\mathscr{A}} (d \to e \to a \to da(ea)) && \text{(using Proposition 2.12 (3) again)}\\
&= \bigwedge_{d\in\mathscr{A}} \Big(d \to \bigwedge_{e\in\mathscr{A}} \Big(e \to \bigwedge_{a\in\mathscr{A}} (a \to da(ea))\Big)\Big) \;=\; \big(\lambda xyz\,.\,xz(yz)\big)^{\mathscr{A}} \;=\; \mathbf{S}^{\mathscr{A}}
\end{aligned}$$

- The proofs for **B**, **W**, and **C** proceed similarly. □

Remarks 2.25. (1) The above property does not generalize to typable terms that are not in β-normal form. For instance, the term $\mathbf{II} = (\lambda x\,.\,x)(\lambda x\,.\,x)$ has the principal polymorphic type $\forall\alpha\,(\alpha \to \alpha)$, but in the dummy implicative structure used in Example 2.20 (where $a \to b = \top$ for all $a, b \in \mathscr{A}$), we have seen that

$$\mathbf{II}\;(=\bot) \quad\neq\quad \bigwedge_{a\in\mathscr{A}} (a \to a) \quad (=\mathbf{I} = \top)\,.$$

However, we conjecture that in any implicative structure $(\mathscr{A}, \preccurlyeq, \to)$, the interpretation of each closed λ-term in β-normal form is equal to the interpretation of its principal type in a polymorphic type system with binary intersections (Coppo et al., 1980; Ronchi della Rocca and Venneri, 1984).

(2) Combining the definitions of $\mathbf{K}^{\mathscr{A}}$ and $\mathbf{S}^{\mathscr{A}}$ with Proposition 2.18, it is clear that

$$\mathbf{K}^{\mathscr{A}} ab \preccurlyeq a \qquad \text{and} \qquad \mathbf{S}^{\mathscr{A}} abc \preccurlyeq ac(bc) \qquad (\text{for all } a, b, c \in \mathscr{A})$$

These inequalities actually mean that each implicative structure $\mathscr{A}$ is also an OCA (van Oosten, 2008), and even an intuitionistic OCA (^{I}OCA) in the sense of Ferrer Santos et al. (2017).

2.6.1 Interpreting call/cc

Since Griffin's seminal work (Griffin, 1990), it is well known that the control operator cc ("call/cc," for: *call with current continuation*) can be given the type $((\alpha \to \beta) \to \alpha) \to \alpha$ that corresponds to Peirce's law. In classical realizability (Krivine, 2009), the control operator cc (that naturally realizes Peirce's law) is the key ingredient to bring the full expressiveness of classical logic into the realm of realizability.

https://doi.org/10.1017/S0960129520000079 Published online by Cambridge University Press

By analogy with Proposition 2.24, it is possible to interpret the control operator œ in any implicative structure $\mathscr{A} = (\mathscr{A}, \preccurlyeq, \to)$ by *identifying* it with Peirce's law, thus letting

$$\text{œ}^{\mathscr{A}} := \bigcurlywedge_{a,b \in \mathscr{A}} (((a \to b) \to a) \to a) \qquad \text{(Peirce's law)}$$

$$= \bigcurlywedge_{a \in \mathscr{A}} ((\neg a \to a) \to a)$$

where negation is defined by $\neg a := (a \to \bot)$ for all $a \in \mathscr{A}$. (The second equality easily follows from the properties of meets and from the variance of implication.)

Of course, the fact that it is possible to interpret the control operator œ in any implicative structure does not mean that any implicative structure is well suited for classical logic, since it may be the case that $\text{œ}^{\mathscr{A}} = \bot$, as shown in the following example:

Example 2.26 (**Dummy implicative structure**). Let us consider the dummy implicative structure (cf. Example 2.4 (1)) constructed on the top of a complete lattice $(L, \preccurlyeq)$ by putting $a \to b := b$ for all $a, b \in L$. In this structure, we have

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

The interpretation $t \mapsto t^{\mathscr{A}}$ of pure λ-terms naturally extends to all λ-terms containing the constant œ, by interpreting the latter by $\text{œ}^{\mathscr{A}}$.

Proposition 2.27 (**œ in a complete HA**). *Let $(\mathscr{A}, \preccurlyeq, \to)$ be a complete HA. Then, the following are equivalent:*

(1) $(\mathscr{A}, \preccurlyeq, \to)$ is a (complete) Boolean algebra;
(2) $\text{œ}^{\mathscr{A}} = \top$;
(3) $t^{\mathscr{A}} = \top$ for all closed λ-terms with œ.

Proof. Let us assume that $(\mathscr{A}, \preccurlyeq, \to)$ is a complete HA.

(1) $\Rightarrow$ (2). In the case where $(\mathscr{A}, \preccurlyeq, \to)$ is a Boolean algebra, Peirce's law is valid in $\mathscr{A}$, so that $((\neg a \to a) \to a) = \top$ for all $a \in \mathscr{A}$. Hence, $\text{œ}^{\mathscr{A}} = \top$, taking the meet.

(1) $\Rightarrow$ (3). Let us assume that $\text{œ}^{\mathscr{A}} = \top$. Given a closed λ-term t with œ, we have $t = t_0\{x := \text{œ}\}$ for some pure λ-term t_0 such that $FV(t_0) \subseteq \{x\}$. From Proposition 2.21, we thus get $t^{\mathscr{A}} = (t_0\{x := \text{œ}^{\mathscr{A}}\})^{\mathscr{A}} \succcurlyeq \text{œ}^{\mathscr{A}} = \top$, hence $t^{\mathscr{A}} = \top$.

(3) $\Rightarrow$ (1). From (3), it is clear that $\text{œ}^{\mathscr{A}} = \top$, hence $((\neg a \to a) \to a) = \top$ for all $a \in \mathscr{A}$. Therefore, $(\neg\neg a \to a) = ((\neg a \to \bot) \to a) \succcurlyeq ((\neg a \to a) \to a) = \top$, hence $(\neg\neg a \to a) = \top$ for all $a \in \mathscr{A}$, which means that $(\mathscr{A}, \preccurlyeq, \to)$ is a Boolean algebra. □

2.7 The problem of consistency

Although it is possible to interpret all closed λ-terms (and even the control operator œ) in any implicative structure $(\mathscr{A}, \preccurlyeq, \to)$, the counter examples given in Examples 2.20 and 2.26 should make clear to the reader that not all implicative structures are well suited to interpret intuitionistic or classical logic. In what follows, we shall say that

https://doi.org/10.1017/S0960129520000079 Published online by Cambridge University Press

Definition 2.28 (**Consistency**). *An implicative structure* $(\mathscr{A}, \preccurlyeq, \to)$ *is*

— intuitionistically consistent *when* $t^{\mathscr{A}} \neq \bot$ *for all closed* λ*-terms;*
— classically consistent *when* $t^{\mathscr{A}} \neq \bot$ *for all closed* λ*-terms with* $\mathtt{c\!c}$.

We have seen that complete HA/Boolean algebra are particular cases of implicative structures. From Propositions 2.21 and 2.27, it is clear that

Proposition 2.29 (**Consistency of complete HA/Boolean algebra**). *All non-degenerate complete HAs (respectively, Boolean) are intuitionistically (respectively, classically) consistent, as implicative structures.*

2.7.1 The case of intuitionistic realizability

Let us recall (van Oosten, 2008) that

Definition 2.30 (**Partial combinatory algebra**). *A partial combinatory algebra (or* PCA*, for short) is a PAS* $(P, \cdot)$ *(Section 2.2.3) with two elements* $\mathtt{k}, \mathtt{s} \in P$ *satisfying the following properties for all* $x, y, z \in P$*:*

(1) $(\mathtt{k} \cdot x)\downarrow$, $(\mathtt{s} \cdot x)\downarrow$ *and* $((\mathtt{s} \cdot x) \cdot y)\downarrow$*;*
(2) $(\mathtt{k} \cdot x) \cdot y \simeq x$*;*
(3) $((\mathtt{s} \cdot x) \cdot y) \cdot z \simeq (x \cdot z) \cdot (y \cdot z)$.

(As usual, the symbol $\simeq$ *indicates that either both sides of the equation are undefined, or that they are both defined and equal.)*

Let $(P, \cdot, \mathtt{k}, \mathtt{s})$ be a PCA. In Section 2.2.3, we have seen (Fact 2.5) that the underlying PAS $(P, \cdot)$ induces a quasi-implicative structure $(\mathfrak{P}(P), \subseteq, \to)$ based on Kleene's implication. Since we are only interested here in full implicative structures (in which $(\top \to \top) = \top$), we shall now assume that the operation of application $(\cdot) : P^2 \to P$ is total, so that the above axioms on $\mathtt{k}, \mathtt{s} \in P$ simplify to

$$(\mathtt{k} \cdot x) \cdot y = x \qquad \text{and} \qquad ((\mathtt{s} \cdot x) \cdot y) \cdot z = (x \cdot z) \cdot (y \cdot z) \qquad\qquad (\text{for all } x, y, z \in P)$$

The quadruple $(P, \cdot, \mathtt{k}, \mathtt{s})$ is then called a (total) *CA*.

We want to show that the implicative structure $\mathscr{A} = (\mathfrak{P}(P), \subseteq, \to)$ induced by any (total) CA $(P, \cdot, \mathtt{k}, \mathtt{s})$ is intuitionistically consistent, thanks to the presence of the combinators $\mathtt{k}$ and $\mathtt{s}$. For that, we call a *closed combinatory term* any closed λ-term that is either $\mathbf{K}$ $(= \lambda xy\,.\,x)$, either $\mathbf{S}$ $(= \lambda xyz\,.\,xz(yz))$, or the application $t_1 t_2$ of two closed combinatory terms t_1 and t_2. Each closed combinatory term t is naturally interpreted in the set P by an element $t^P \in P$ that is recursively defined by

$$\mathbf{K}^P := \mathtt{k}, \qquad \mathbf{S}^P := \mathtt{s}, \qquad \text{and} \qquad (t_1 t_2)^P := t_1^P \cdot t_2^P\,.$$

We then easily check that

Lemma 2.31. *For each closed combinatory term t, we have* $t^P \in t^{\mathscr{A}}$.

Proof. By induction on t, distinguishing the following cases:

https://doi.org/10.1017/S0960129520000079 Published online by Cambridge University Press

- $t = \mathbf{K}$. In this case, we have

$$\mathbf{K}^P = \mathtt{k} \in \bigcap_{a,b \in \mathfrak{P}(P)} (a \to b \to a) = \mathbf{K}^{\mathscr{A}} \qquad \text{(by Proposition 2.24)}$$

- $t = \mathbf{S}$. In this case, we have

$$\mathbf{S}^P = \mathtt{s} \in \bigcap_{a,b,c \in \mathfrak{P}(P)} ((a \to b \to c) \to (a \to b) \to a \to c) = \mathbf{S}^{\mathscr{A}} \qquad \text{(by Proposition 2.24)}$$

- $t = t_1 t_2$, where t_1, t_2 are closed combinatory terms. By IH, we have $t_1^P \in t_1^{\mathscr{A}}$ and $t_2^P \in t_2^{\mathscr{A}}$, hence $t^P = t_1^P \cdot t_2^P \in t_1^{\mathscr{A}} t_2^{\mathscr{A}} = t^{\mathscr{A}}$, by Corollary 2.14. □

From the above observation, we immediately get that

Proposition 2.32 (**Consistency**). *The implicative structure $(\mathfrak{P}(P), \subseteq, \to)$ induced by any (total) CA $(P, \cdot, \mathtt{k}, \mathtt{s})$ is intuitionistically consistent.*

Proof. Let t be a closed λ-term. From the theory of the λ-calculus (Barendregt, 1984, Theorem 7.3.10(i)), there is a closed combinatory term t_0 such that $t_0 \twoheadrightarrow_\beta t$. We have $t_0^P \in t_0^{\mathscr{A}}$ (by Lemma 2.31) and $t_0^{\mathscr{A}} \subseteq t^{\mathscr{A}}$ (by Proposition 2.18), hence $t^{\mathscr{A}} \neq \varnothing\ (= \bot)$. □

(The implicative structure $(\mathfrak{P}(P), \subseteq, \to)$ is not classically consistent, in general.)

2.7.2 The case of classical realizability

Definition 2.33 (**Abstract Krivine structure**). *An AKS is any structure of the form* $\mathcal{K} = (\Lambda, \Pi, @, \cdot, \mathtt{k}\_, \mathtt{K}, \mathtt{S}, \mathtt{cc}, \mathrm{PL}, \perp\!\!\!\perp)$, *where*

- *Λ and Π are nonempty sets, whose elements are, respectively, called the $\mathcal{K}$-*terms *and the $\mathcal{K}$-*stacks *of the AKS $\mathcal{K}$;*
- *$@ : \Lambda \times \Lambda \to \Lambda$ ("application") is an operation that associates to each pair of $\mathcal{K}$-terms $t, u \in \Lambda$ a $\mathcal{K}$-term $@(t, u) \in \Lambda$, usually written tu (by juxtaposition);*
- *$(\cdot) : \Lambda \times \Pi \to \Pi$ ("push") is an operation that associates to each $\mathcal{K}$-term $t \in \Lambda$ and to each $\mathcal{K}$-stack $\pi \in \Pi$ a $\mathcal{K}$-stack $t \cdot \pi \in \Pi$;*
- *$\mathtt{k}\_ : \Pi \to \Lambda$ is a function that turns each $\mathcal{K}$-stack $\pi \in \Pi$ into a $\mathcal{K}$-term $\mathtt{k}_\pi \in \Pi$, called the* continuation *associated to π;*
- *$\mathtt{K}, \mathtt{S}, \mathtt{cc} \in \Lambda$ are three distinguished $\mathcal{K}$-terms;*
- *$PL \subseteq \Lambda$ is a set of $\mathcal{K}$-terms, called the set of* proof-like $\mathcal{K}$-terms, *that contains the three $\mathcal{K}$-terms $\mathtt{K}$, $\mathtt{S}$, and $\mathtt{cc}$, and that is closed under application;*
- *$\perp\!\!\!\perp \subseteq \Lambda \times \Pi$ is a binary relation between $\mathcal{K}$-terms and $\mathcal{K}$-stacks, called the* pole *of the AKS $\mathcal{K}$, that fulfills the following axioms:*

$$\begin{array}{llll}
t \perp\!\!\!\perp u \cdot \pi & \textit{implies} & tu & \perp\!\!\!\perp \pi \\
t \perp\!\!\!\perp \pi & \textit{implies} & \mathtt{K} & \perp\!\!\!\perp t \cdot u \cdot \pi \\
t \perp\!\!\!\perp v \cdot uv \cdot \pi & \textit{implies} & \mathtt{S} & \perp\!\!\!\perp t \cdot u \cdot v \cdot \pi \\
t \perp\!\!\!\perp \mathtt{k}_\pi \cdot \pi & \textit{implies} & \mathtt{cc} & \perp\!\!\!\perp t \cdot \pi \\
t \perp\!\!\!\perp \pi & \textit{implies} & \mathtt{k}_\pi & \perp\!\!\!\perp t \cdot \pi'
\end{array}$$

for all $t, u, v \in \Lambda$ and $\pi, \pi' \in \Pi$.

https://doi.org/10.1017/S0960129520000079 Published online by Cambridge University Press

Remarks 2.34. (1) The above closure conditions on the pole $\perp\!\!\!\perp \subseteq \Lambda \times \Pi$ actually express that it is closed by *anti-evaluation*, in the sense of the evaluation rules

$$\begin{array}{ll} tu \star \pi & \succ t \star u \cdot \pi \\ \mathtt{K} \star t \cdot u \cdot \pi & \succ t \star \pi \\ \mathtt{S} \star t \cdot u \cdot v \cdot \pi & \succ t \star v \cdot uv \cdot \pi \\ \mathtt{cc} \star t \cdot \pi & \succ t \star \mathtt{k}_\pi \cdot \pi \\ \mathtt{k}_\pi \star t \cdot \pi' & \succ t \star \pi \end{array}$$

(writing $t \star \pi = (t, \pi)$ the *process* formed by a $\mathcal{K}$-term t and a $\mathcal{K}$-stack π).

(2) The notion of AKS – which was introduced by Streicher (2013) – is very close to the notion of *realizability structure* such as introduced by Krivine (2011), the main difference being that the latter notion introduces more primitive combinators, essentially to mimic the evaluation strategy of the λ_c-calculus (Krivine, 2009). However, in what follows, we shall not need such a level of granularity, so that we shall stick to Streicher's definition.

In Section 2.2.5, we have seen (Fact 2.9) that the quadruple $(\Lambda, \Pi, \cdot, \perp\!\!\!\perp)$ underlying any AKS $\mathcal{K} = (\Lambda, \Pi, @, \cdot, \mathtt{k\_}, \mathtt{K}, \mathtt{S}, \mathtt{cc}, \mathrm{PL}, \perp\!\!\!\perp)$ induces an implicative structure $\mathscr{A} = (\mathscr{A}, \preccurlyeq, \to)$ that is defined by

- $\mathscr{A} \ := \ \mathfrak{P}(\Pi)$
- $a \preccurlyeq b \ :\Leftrightarrow \ a \supseteq b$ (for all $a, b \in \mathscr{A}$)
- $a \to b \ := \ a^{\perp\!\!\!\perp} \cdot b \ = \ \{t \cdot \pi : t \in a^{\perp\!\!\!\perp},\ \pi \in b\}$ (for all $a, b \in \mathscr{A}$)

where $a^{\perp\!\!\!\perp} := \{t \in \Lambda : \forall \pi \in a,\ \ (t, \pi) \in \perp\!\!\!\perp\} \in \mathfrak{P}(\Lambda)$ is the orthogonal of the set $a \in \mathfrak{P}(\Pi)$ with respect to the pole $\perp\!\!\!\perp \subseteq \Lambda \times \Pi$.

Note that since the ordering of subtyping $a \preccurlyeq b$ is defined here as the relation of *inverse inclusion* $a \supseteq b$ (between two sets of stacks $a, b \in \mathfrak{P}(\Pi)$), the smallest element of the induced implicative structure $\mathscr{A} = (\mathscr{A}, \preccurlyeq, \to)$ is given by $\bot = \Pi$.

Remark 2.35. In Streicher (2013), Streicher only considers sets of stacks $a \in \mathfrak{P}(\Pi)$ such that $a^{\perp\!\!\!\perp\perp\!\!\!\perp} = a$, thus working with a smaller set of "truth values" $\mathscr{A}'$ given by

$$\mathscr{A}' \ := \ \mathfrak{P}_{\perp\!\!\!\perp}(\Pi) \ = \ \{a \in \mathfrak{P}(\Pi) : a^{\perp\!\!\!\perp\perp\!\!\!\perp} = a\}\,.$$

Technically, such a restriction requires to alter the interpretation of implication, by adding another step of bi-orthogonal closure:

$$a \to' b \ := \ \left(a^{\perp\!\!\!\perp} \cdot b\right)^{\perp\!\!\!\perp\perp\!\!\!\perp} \qquad \text{(for all } a, b \in \mathscr{A}'\text{)}$$

However, the resulting triple $(\mathscr{A}', \preccurlyeq, \to')$ is in general not an implicative structure, since it does not fulfill axiom (2) of Definition 2.1.[4] For this reason, we shall follow Krivine by considering all sets of stacks as truth values in what follows.

The basic intuition underlying Krivine's realizability is that each set of $\mathcal{K}$-stacks $a \in \mathfrak{P}(\Pi)$ represents the set of *counter-realizers* (or *attackers*) of a particular formula, whereas its orthogonal $a^{\perp\!\!\!\perp} \in \mathfrak{P}(\Lambda)$ represents the set of *realizers* (or *defenders*) of the same formula[5]. In this setting, the realizability relation is naturally defined by

$$t \Vdash a \ :\Leftrightarrow \ t \in a^{\perp\!\!\!\perp} \qquad \text{(for all } t \in \Lambda,\ a \in \mathscr{A}\text{)}$$

However, when the pole $\perp\!\!\!\perp \subseteq \Lambda \times \Pi$ is not empty, we can observe that

https://doi.org/10.1017/S0960129520000079 Published online by Cambridge University Press

Fact 2.36. Given a fixed $(t_0, \pi_0) \in \perp\!\!\!\perp$, we have $\quad \mathtt{k}_{\pi_0} t_0 \Vdash a \quad$ for all $a \in \mathscr{A}$.

so that *any* element of the implicative structure is actually realized by some $\mathcal{K}$-term (which does not even depend on the considered element of $\mathscr{A}$). This is the reason why Krivine introduces an extra parameter, the set of *proof-like* ($\mathcal{K}$)-terms $\mathrm{PL} \subseteq \Lambda$, whose elements are (by convention) the realizers that are considered as valid certificates of the truth of a formula. (The terminology "proof-like" comes from the fact that all realizers that come from actual proofs belong to the subset $\mathrm{PL} \subseteq \Lambda$.)

Following Krivine, we say that a truth value $a \in \mathscr{A}$ is *realized* when it is realized by a proof-like term, that is

$$\begin{aligned} a \text{ realized} \quad &:\Leftrightarrow \quad \exists t \in \mathrm{PL},\ t \Vdash a \\ &\Leftrightarrow \quad a^{\perp\!\!\!\perp} \cap \mathrm{PL} \neq \varnothing \end{aligned}$$

More generally, we say that the AKS $\mathcal{K} = (\Lambda, \Pi, \ldots, \mathrm{PL}, \perp\!\!\!\perp)$ is *consistent* when the smallest truth value $\bot = \Pi$ is not realized, that is

$$\mathcal{K} \text{ consistent} \quad :\Leftrightarrow \quad \Pi^{\perp\!\!\!\perp} \cap \mathrm{PL} = \varnothing\,.$$

We now need to check that Krivine's notion of consistency is consistent with the one that comes with implicative structures (Definition 2.28). For that, we call a *closed classical combinatory term* any closed λ-term with cc that is either $\mathbf{K}$ $(= \lambda xy\,.\,x)$, either $\mathbf{S}$ $(= \lambda xyz\,.\,xz(yz))$, either the constant cc, or the application $t_1 t_2$ of two closed classical combinatory terms t_1 and t_2. Each closed classical combinatory term t is naturally interpreted by an element $t^\Lambda \in \Lambda$ that is recursively defined by

$$\mathbf{K}^\Lambda := \mathtt{K}, \qquad \mathbf{S}^\Lambda := \mathtt{S}, \qquad \text{cc}^\Lambda := \text{cc}, \qquad \text{and} \qquad (t_1 t_2)^\Lambda := t_1^\Lambda t_2^\Lambda\,.$$

From the closure properties of the set PL of proof-like terms, it is clear that $t^\Lambda \in \mathrm{PL}$ for each closed classical combinatory term t. Moreover,

Lemma 2.37. *For each closed classical combinatory term t, we have* $\quad t^\Lambda \Vdash t^{\mathscr{A}}$.

Proof. By induction on t, distinguishing the following cases:

- $t = \mathbf{K}, \mathbf{S}, \text{cc}$. In this case, combining standard results of classical realizability (Krivine, 2011) with the properties of implicative structures, we get

$$\mathbf{K}^\Lambda \;=\; \mathtt{K} \;\Vdash\; \bigwedge_{a,b \in \mathscr{A}} (a \to b \to a) \;=\; \mathbf{K}^{\mathscr{A}} \qquad \text{(by Proposition 2.24)}$$

$$\mathbf{S}^\Lambda \;=\; \mathtt{S} \;\Vdash\; \bigwedge_{a,b,c \in \mathscr{A}} ((a \to b \to c) \to (a \to b) \to a \to c) \;=\; \mathbf{S}^{\mathscr{A}} \qquad \text{(by Proposition 2.24)}$$

$$\text{cc}^\Lambda \;=\; \text{cc} \;\Vdash\; \bigwedge_{a,b \in \mathscr{A}} (((a \to b) \to a) \to a) \;=\; \text{cc}^{\mathscr{A}} \qquad \text{(by definition)}$$

- $t = t_1 t_2$, where t_1 are t_2 are closed classical combinatory terms. In this case, we have $t_1^\Lambda \Vdash t_1^{\mathscr{A}}$ and $t_2^\Lambda \Vdash t_2^{\mathscr{A}}$ by IH. And since $t_1^{\mathscr{A}} \preccurlyeq (t_2^{\mathscr{A}} \to t_1^{\mathscr{A}} t_2^{\mathscr{A}})$ (from Proposition 2.12 (3)), we also have $t_1^\Lambda \Vdash t_2^{\mathscr{A}} \to t_1^{\mathscr{A}} t_2^{\mathscr{A}}$ (by subtyping), so that we get $t^\Lambda = t_1^\Lambda t_2^\Lambda \Vdash t_1^{\mathscr{A}} t_2^{\mathscr{A}} = t^{\mathscr{A}}$ (by modus ponens). □

We can now conclude

https://doi.org/10.1017/S0960129520000079 Published online by Cambridge University Press

Proposition 2.38. *If an AKS $\mathcal{K} = (\Lambda, \Pi, \ldots, PL, \perp\!\!\!\perp)$ is consistent (in the sense that $\Pi^{\perp\!\!\!\perp} \cap PL = \varnothing$), then the induced implicative structure $\mathscr{A} = (\mathfrak{P}(\Pi), \supseteq, \to)$ is classically consistent (in the sense of Definition 2.28).*

Proof. Let us assume that $\Pi^{\perp\!\!\!\perp} \cap \mathrm{PL} = \varnothing$. Given a closed λ-term t with cc, there exists a closed classical combinatory term t_0 such that $t_0 \twoheadrightarrow_\beta t$. So that we have $t_0^\Lambda \Vdash t_0^{\mathscr{A}}$ (by Lemma 2.37) and $t_0^{\mathscr{A}} \preccurlyeq t^{\mathscr{A}}$ (by Proposition 2.18), hence $t_0^\Lambda \Vdash t^{\mathscr{A}}$ (by subtyping). But this implies that $t^{\mathscr{A}} \neq \bot$ $(= \Pi)$, since $t^\Lambda \in (t^{\mathscr{A}})^{\perp\!\!\!\perp} \cap \mathrm{PL} \neq \varnothing$. □

Note that the converse implication does not hold in general. The reason is that the criterion of consistency for the considered AKS depends both on the pole $\perp\!\!\!\perp$ and on the conventional set PL of proof-like terms. (In particular, it should be clear to the reader that the larger the set PL, the stronger the corresponding criterion of consistency.) On the other hand, the construction of the induced implicative structure $\mathscr{A} = (\mathfrak{P}(\Pi), \supseteq, \to)$ does not depend on the set PL, so that the criterion of classical consistency of Definition 2.28 – which does not depend on PL either – can only be regarded as a minimal criterion of consistency.

In order to reflect more faithfully Krivine's notion of consistency at the level of the induced implicative structure, it is now time to introduce the last ingredient of implicative algebras: the notion of *separator*.

3. Separation

3.1 Separators and implicative algebras

Let $\mathscr{A} = (\mathscr{A}, \preccurlyeq, \to)$ be an implicative structure.

Definition 3.1 (**Separator**). *We call a* separator *of $\mathscr{A}$ any subset $S \subseteq \mathscr{A}$ that fulfills the following conditions for all $a, b \in \mathscr{A}$:*

(1) if $a \in S$ and $a \preccurlyeq b$, then $b \in S$ *(S is upwards closed)*
(2) $\mathbf{K}^{\mathscr{A}} = (\lambda xy \,.\, x)^{\mathscr{A}} \in S$ and $\mathbf{S}^{\mathscr{A}} = (\lambda xyz \,.\, xz(yz))^{\mathscr{A}} \in S$ *(S contains **K** and **S**)*
(3) if $(a \to b) \in S$ and $a \in S$, then $b \in S$ *(S is closed under modus ponens).*

A separator $S \subseteq \mathscr{A}$ is said to be

— consistent *when $\bot \notin S$;*
— classical *when $\text{cc}^{\mathscr{A}} \in S$.*

Remark 3.2. In the presence of condition (1) (upwards closure), condition (3) (closure under modus ponens) is actually equivalent to

(3′) If $a, b \in S$, then $ab \in S$ (closure under application)

Proof. Let $S \subseteq \mathscr{A}$ be an upwards closed subset of $\mathscr{A}$.

- (3) ⇒ (3′) Suppose that $a, b \in S$. Since $a \preccurlyeq (b \to ab)$ (from Proposition 2.12 (3)), we get $(b \to ab) \in S$ by upwards closure, hence $ab \in S$ by (3).
- (3′) ⇒ (3) Suppose that $(a \to b), a \in S$. By (3′), we have $(a \to b)a \in S$, and since $(a \to b)a \preccurlyeq b$ (from Proposition 2.12 (2)), we get $b \in S$ by upwards closure. □

https://doi.org/10.1017/S0960129520000079 Published online by Cambridge University Press

Intuitively, each separator $S \subseteq \mathscr{A}$ defines a particular "criterion of truth" within the implicative structure $\mathscr{A} = (\mathscr{A}, \preccurlyeq, \to)$. In implicative structures, separators play the very same role as filters in HAs, and it is easy to check that

Proposition 3.3 (**Separators in a complete HA**). *If $\mathscr{A} = (\mathscr{A}, \preccurlyeq, \to)$ is a complete HA, then a subset $S \subseteq \mathscr{A}$ is a separator (in the sense of implicative structures) if and only if S is a filter (in the sense of HAs).*

Proof. Indeed, when the implicative structure $\mathscr{A} = (\mathscr{A}, \preccurlyeq, \to)$ is a complete HA, the conditions (1), (2), and (3′) defining separators simplify to

(1) if $a \in S$ and $a \preccurlyeq b$, then $b \in S$ (upwards closure)
(2) $\top\ (= \mathbf{K}^{\mathscr{A}} = \mathbf{S}^{\mathscr{A}}) \in S$ (from Proposition 2.21)
(3′) if $a, b \in S$, then $a \curlywedge b\ (= ab) \in S$ (from Corollary 2.13)

which is precisely the definition of the notion of a filter. □

However, separators are in general *not* filters, since they are not closed under binary meets (i.e., $a \in S$ and $b \in S$ do not necessarily imply that $a \curlywedge b \in S$). Actually, one of the key ideas we shall develop in the rest of this paper is that the difference between (intuitionistic or classical) realizability and forcing (in the sense of Kripke or Cohen) lies precisely in the difference between separators and filters.

Proposition 3.4. *If $S \subseteq \mathscr{A}$ is a separator, then for all λ-terms t with free variables $x_1, \ldots, x_n$ and for all parameters $a_1, \ldots, a_n \in S$, we have*

$$(t\{x_1 := a_1, \ldots, x_n := a_n\})^{\mathscr{A}} \in S\,.$$

In particular, for all closed λ-terms t, we have $t^{\mathscr{A}} \in S$.

Proof. Let t be a λ-term with free variables $x_1, \ldots, x_n$, and let $a_1, \ldots, a_n$ be parameters taken in S. From the theory of the λ-calculus, there exists a closed combinatory term t_0 such that $t_0 \twoheadrightarrow_\beta \lambda x_1 \cdots x_n\,.\,t$. It is clear that $t_0^{\mathscr{A}} a_1 \cdots a_n \in S$ from the conditions (2) and (3′) on the separator S. Moreover, by Proposition 2.18, we have

$$t_0^{\mathscr{A}} a_1 \cdots a_n \ \preccurlyeq\ (\lambda x_1 \cdots x_n\,.\,t)^{\mathscr{A}} a_1 \cdots a_n \ \preccurlyeq\ (t\{x_1 := a_1, \ldots, x_n := a_n\})^{\mathscr{A}}\ ,$$

so that we get $(t\{x_1 := a_1, \ldots, x_n := a_n\})^{\mathscr{A}} \in S$, by upwards closure. □

Definition 3.5 (**Implicative algebra**). *We call an* implicative algebra *any implicative structure $(\mathscr{A}, \preccurlyeq, \to)$ equipped with a separator $S \subseteq \mathscr{A}$. An implicative algebra $(\mathscr{A}, \preccurlyeq, \to, S)$ is said to be* consistent *(respectively,* classical*) when the underlying separator $S \subseteq \mathscr{A}$ is* consistent *(respectively,* classical*).*

3.2 Examples

3.2.1 Complete HAs.

We have seen that a complete HA $(H, \preccurlyeq)$ can be seen as an implicative structure $(H, \preccurlyeq, \to)$ where implication is defined by

$$a \to b \ :=\ \max\{c \in H : (c \curlywedge a) \preccurlyeq b\} \qquad \text{(for all } a, b \in H)$$

The complete HA $(H, \preccurlyeq)$ can also be seen as an implicative *algebra*, by endowing it with the trivial separator $S = \{\top\}$ (i.e., the smallest filter of H).

https://doi.org/10.1017/S0960129520000079 Published online by Cambridge University Press

3.2.2 Implicative algebras of intuitionistic realizability

Let $(P, \cdot, \mathtt{k}, \mathtt{s})$ be a (total) CA. In Section 2.7.1, we have seen that such a structure induces an implicative structure $(\mathfrak{P}(P), \subseteq \rightarrow)$, whose implication is defined by

$$a \rightarrow b \;:=\; \{z \in P : \forall x \in a,\; z \cdot x \in b\} \qquad \text{(for all } a, b \in \mathfrak{P}(P))$$

The above implicative structure is naturally turned into an implicative algebra by endowing it with the separator $S = \mathfrak{P}(P) \setminus \{\varnothing\}$ formed by all truth values that contain at least a realizer. In this case, the separator $S = \mathfrak{P}(P) \setminus \{\varnothing\}$ is not only consistent (in the sense of Definition 3.1), but it is also a *maximal separator* (see Section 3.6).

Remark 3.6. In an arbitrary implicative structure $(\mathscr{A}, \preccurlyeq, \rightarrow)$, we can observe that the subset $\mathscr{A} \setminus \{\bot\} \subset \mathscr{A}$ is in general *not* a separator. (Counter example: consider the Boolean algebra with four elements.) The property that $\mathscr{A} \setminus \{\bot\}$ is a separator is thus a specific property of the implicative structures induced by (total) CAs, and the existence of such a separator that is trivially consistent explains why there is no need to introduce a notion of proof-like term in intuitionistic realizability.

3.2.3 Implicative algebras of classical realizability

Let

$$\mathcal{K} \;=\; (\Lambda, \Pi, @, \cdot, \mathtt{k\_}, \mathrm{K}, \mathrm{S}, \text{æ}, \mathrm{PL}, \perp\!\!\!\perp)$$

be an AKS (Definition 2.33, p. 474). We have seen (Section 2.7.2) that such a structure induces an implicative structure $(\mathscr{A}, \preccurlyeq, \rightarrow)$ where

- $\mathscr{A} \;:=\; \mathfrak{P}(\Pi)$;
- $a \preccurlyeq b \;:\Leftrightarrow\; a \supseteq b$ (for all $a, b \in \mathscr{A}$);
- $a \rightarrow b \;:=\; a^{\perp\!\!\!\perp} \cdot b \;=\; \{t \cdot \pi : t \in a^{\perp\!\!\!\perp},\; \pi \in b\}$ (for all $a, b \in \mathscr{A}$);

Using the set PL of proof-like terms, we can now turn the former implicative structure into an implicative algebra $(\mathscr{A}, \preccurlyeq, \rightarrow, S)$, letting

$$S \;:=\; \{a \in \mathscr{A} : a^{\perp\!\!\!\perp} \cap \mathrm{PL} \neq \varnothing\}\,.$$

Proposition 3.7. *The subset $S = \{a \in \mathscr{A} : a^{\perp\!\!\!\perp} \cap PL \neq \varnothing\} \subseteq \mathscr{A}$ is a classical separator of the implicative structure $(\mathscr{A}, \preccurlyeq, \rightarrow)$.*

Proof. By construction, we have $S = \{a \in \mathscr{A} : \exists t \in \mathrm{PL},\; t \Vdash a\}$.

(1) Upwards closure: obvious, by subtyping.
(2) We have seen in Section 2.7.2 (Proof of Lemma 2.37) that $\mathrm{K} \Vdash \mathbf{K}^{\mathscr{A}}$, $\mathrm{S} \Vdash \mathbf{S}^{\mathscr{A}}$ and $\text{æ} \Vdash \text{æ}^{\mathscr{A}}$, and since $\mathrm{K}, \mathrm{S}, \text{æ} \in \mathrm{PL}$, we get $\mathbf{K}^{\mathscr{A}}, \mathbf{S}^{\mathscr{A}}, \text{æ}^{\mathscr{A}} \in S$.
(3) Suppose that $(a \rightarrow b), a \in S$. From the definition of S, we have $t \Vdash a \rightarrow b$ and $u \Vdash a$ for some $t, u \in \mathrm{PL}$, so that $tu \Vdash b$, where $tu \in \mathrm{PL}$. Hence, $b \in S$. □

Moreover, it is obvious that

Proposition 3.8 (**Consistency**). *The classical implicative algebra $(\mathscr{A}, \preccurlyeq, \rightarrow, S)$ induced by the AKS $\mathcal{K} = (\Lambda, \Pi, \ldots, PL, \perp\!\!\!\perp)$ is consistent (in the sense of Definition 3.5) if and only if $\mathcal{K}$ is consistent (in the sense that $\Pi^{\perp\!\!\!\perp} \cap PL = \varnothing$).*

Proof. Indeed, we have $\bot \notin S$ iff $\bot^{\perp\!\!\!\perp} \cap \mathrm{PL} = \varnothing$, that is, iff $\Pi^{\perp\!\!\!\perp} \cap \mathrm{PL} = \varnothing$. □

https://doi.org/10.1017/S0960129520000079 Published online by Cambridge University Press

3.3 Generating separators

Let $\mathscr{A} = (\mathscr{A}, \preccurlyeq, \to)$ be an implicative structure. For each subset $X \subseteq \mathscr{A}$, we write

- $\uparrow X = \{a \in \mathscr{A} : \exists a_0 \in X,\ a_0 \preccurlyeq a\}$ the *upwards closure* of X in $\mathscr{A}$;
- $@(X)$ the *applicative closure of* X, defined as the smallest subset of $\mathscr{A}$ containing X (as a subset) and closed under application;
- $\Lambda(X)$ the λ*-closure of* X, formed by all elements $a \in \mathscr{A}$ that can be written $a = (t\{x_1 := a_1, \ldots, x_n := a_n\})^{\mathscr{A}}$ for some pure λ-term t with free variables $x_1, \ldots, x_n$ and for some parameters $a_1, \ldots, a_n \in X$.

Note that in general, the sets $@(X)$ and $\Lambda(X)$ are not upwards closed, but we obviously have the inclusion $@(X) \subseteq \Lambda(X)$.

Proposition 3.9 (**Generated separator**). *Given any subset $X \subseteq \mathscr{A}$, we have*

$$\uparrow\Lambda(X) \ = \ \uparrow @(X \cup \{\mathbf{K}^{\mathscr{A}}, \mathbf{S}^{\mathscr{A}}\})\,.$$

By construction, the above set is the smallest separator of $\mathscr{A}$ that contains X as a subset; it is called the separator generated by X*, and written* $Sep(X)$.

Proof. The inclusion $\uparrow @(X \cup \{\mathbf{K}^{\mathscr{A}}, \mathbf{S}^{\mathscr{A}}\}) \subseteq \uparrow\Lambda(X)$ is obvious, and the converse inclusion follows from Proposition 2.18 using the fact each λ-term is the β-contracted of some combinatory term constructed from variables, **K**, **S**, and application. The set $\uparrow\Lambda(X)$ is clearly a separator (closure under application follows from Proposition 2.12 (1)), and from Proposition 3.4, it is included in any separator containing X as a subset. □

An important property of first-order logic is the deduction lemma, which states that an implication $\phi \Rightarrow \psi$ is provable in a theory $\mathscr{T}$ if and only if the formula ψ is provable in the theory $\mathscr{T} + \phi$ that is obtained by enriching $\mathscr{T}$ with the axiom ϕ. Viewing separators $S \subseteq \mathscr{A}$ as theories, this naturally suggests the following semantic counterpart:

Lemma 3.10 (**Deduction in a separator**). *For each separator $S \subseteq \mathscr{A}$, we have*

$$(a \to b) \in S \qquad \textit{iff} \qquad b \in Sep(S \cup \{a\}) \qquad\qquad (\textit{for all } a, b \in \mathscr{A})$$

Proof. Suppose that $(a \to b) \in S$. Then, $(a \to b) \in \mathrm{Sep}(S \cup \{a\})$ (by inclusion), and since $a \in \mathrm{Sep}(S \cup \{a\})$ (by construction), we get $b \in \mathrm{Sep}(S \cup \{a\})$ (by modus ponens). Conversely, let us suppose that $b \in \mathrm{Sep}(S \cup \{a\})$. From the definition of the separator $\mathrm{Sep}(S \cup \{a\})$, this means that there are a λ-term t with free variables $x_1, \ldots, x_n$ and parameters $a_1, \ldots, a_n \in S \cup \{a\}$ such that $(t\{x_1 := a_1, \ldots, x_n := a_n\})^{\mathscr{A}} \preccurlyeq b$. Without loss of generality, we can assume that $a_1 = a$ and $a_2, \ldots, a_n \in S$ (with $n \geq 1$). Letting $c := (\lambda x_1 \,.\, t\{x_2 := a_2, \ldots, x_n := a_n\})^{\mathscr{A}}$, we observe that $c \in S$, by Proposition 3.4. Moreover, we have $ca \preccurlyeq (t\{x_1 := a_1, x_2 := a_2, \ldots, x_n := a_n\})^{\mathscr{A}} \preccurlyeq b$ by Proposition 2.18. And by adjunction, we deduce that $c \preccurlyeq (a \to b)$, hence $(a \to b) \in S$. □

In what follows, we shall say that a separator $S \subseteq \mathscr{A}$ is *finitely generated* when $S = \mathrm{Sep}(X)$ for some finite subset $X \subseteq \mathscr{A}$. Two important examples of finitely generated separators of an implicative structure $\mathscr{A} = (\mathscr{A}, \preccurlyeq, \to)$ are

— the *intuitionistic core* of $\mathscr{A}$, defined by $S_J^0(\mathscr{A}) := \mathrm{Sep}(\varnothing)$;
— the *classical core* of $\mathscr{A}$, defined by $S_K^0(\mathscr{A}) := \mathrm{Sep}(\{œ^{\mathscr{A}}\})$.

https://doi.org/10.1017/S0960129520000079 Published online by Cambridge University Press

By definition, the set $S^0_J(\mathscr{A})$ (respectively, $S^0_K(\mathscr{A})$) is the smallest separator (respectively, the smallest classical separator) of $\mathscr{A}$; and from Proposition 3.9, it is clear that the implicative structure $\mathscr{A} = (\mathscr{A}, \preccurlyeq, \to)$ is intuitionistically consistent (respectively, classically consistent) in the sense of Definition 2.28 if and only if $\bot \notin S^0_J(\mathscr{A})$ (respectively, $\bot \notin S^0_K(\mathscr{A})$).

3.4 Interpreting first-order logic

3.4.1 Conjunction and disjunction

Each implicative structure $\mathscr{A} = (\mathscr{A}, \preccurlyeq, \to)$ describes a particular logic from the interaction between implication $a \to b$ and universal quantification, seen as a meet with respect to the ordering $a \preccurlyeq b$ of subtyping. In such a framework, conjunction (notation: $a \times b$) and disjunction (notation: $a + b$) are naturally defined using the standard encodings of minimal second-order logic (Girard, 1972; Girard et al., 1989):

$$a \times b \;:=\; \bigcurlywedge_{c \in \mathscr{A}} ((a \to b \to c) \to c)$$

$$a + b \;:=\; \bigcurlywedge_{c \in \mathscr{A}} ((a \to c) \to (b \to c) \to c)$$

Finally, negation and logical equivalence are defined as expected, letting $\neg a := (a \to \bot)$ and $a \leftrightarrow b := (a \to b) \times (b \to a)$. We easily check that

Proposition 3.11. *When* $(\mathscr{A}, \preccurlyeq, \to)$ *is a complete HA:*

$$a \times b \;=\; a \curlywedge b \qquad \textit{and} \qquad a + b \;=\; a \curlyvee b \qquad\qquad (\textit{for all } a, b \in \mathscr{A})$$

(The proof is left as an exercise to the reader.)

In the general case, the introduction and elimination rules of conjunction and disjunction are naturally expressed as semantic typing rules (see Section 2.5) using the very same proof terms as in Curry-style system F (Leivant, 1983; van Bakel et al., 1994):

Proposition 3.12 **(Typing rules for $\times$ and $+$).** *The semantic typing rules*

$$\frac{\Gamma \vdash t : a \quad \Gamma \vdash u : b}{\Gamma \vdash \lambda z\,.\,z\,t\,u : a \times b} \qquad \frac{\Gamma \vdash t : a \times b}{\Gamma \vdash t\,(\lambda xy\,.\,x) : a} \qquad \frac{\Gamma \vdash t : a \times b}{\Gamma \vdash t\,(\lambda xy\,.\,y) : b}$$

$$\frac{\Gamma \vdash t : a}{\Gamma \vdash \lambda zw\,.\,z\,t : a + b} \qquad \frac{\Gamma \vdash t : b}{\Gamma \vdash \lambda zw\,.\,w\,t : a + b}$$

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

are valid in any implicative structure.

(Recall that $\Gamma \vdash t : a$ means: $FV(t) \subseteq \mathrm{dom}(\Gamma)$ and $(t[\Gamma])^{\mathscr{A}} \preccurlyeq a$.)

Following the spirit of Proposition 2.24, we can notice that via the interpretation $t \mapsto t^{\mathscr{A}}$ of pure λ-terms into the implicative structure $\mathscr{A}$ (Section 2.4), the pairing construct $\langle t, u\rangle := \lambda z\,.\,z\,t\,u$ appears to be the same as conjunction itself:

Proposition 3.13. *For all* $a, b \in \mathscr{A}$: $\langle a, b\rangle^{\mathscr{A}} = (\lambda z\,.\,z\,a\,b)^{\mathscr{A}} = a \times b$.

Proof. Same proof technique as for Proposition 2.24. □

https://doi.org/10.1017/S0960129520000079 Published online by Cambridge University Press

3.4.2 Quantifiers

In any implicative structure $\mathscr{A} = (\mathscr{A}, \preccurlyeq, \to)$, the universal quantification of a family of truth values $(a_i)_{i\in I} \in \mathscr{A}^I$ is naturally defined as its meet:

$$\bigforall_{i\in I} a_i \ := \ \bigcurlywedge_{i\in I} a_i \,.$$

It is obvious that

Proposition 3.14 (**Rules for $\forall$**). *The following semantic typing rules*

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

are valid in any implicative structure.

In such a framework, it would be quite natural to define existential quantification dually, that is, as a join. Alas, this interpretation does not fulfill (in general) the elimination rule for $\exists$ – remember that joins only exist by accident. As for conjunction and disjunction, we shall use the corresponding encoding in second-order minimal logic (Girard, 1972; Girard et al., 1989), letting

$$\bigexists_{i\in I} a_i \ := \ \bigcurlywedge_{c\in\mathscr{A}} \Big(\bigcurlywedge_{i\in I} (a_i \to c) \ \to \ c \Big) .$$

Again, we easily check that

Proposition 3.15. *When $(\mathscr{A}, \preccurlyeq, \to)$ is a complete HA:*

$$\bigexists_{i\in I} a_i \ = \ \bigcurlyvee_{i\in I} a_i \qquad\qquad (\textit{for all } (a_i)_{i\in I} \in \mathscr{A}^I)$$

Coming back to the general case:

Proposition 3.16 (**Rules for $\exists$**). *The following semantic typing rules*

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

are valid in any implicative structure.

3.4.3 Leibniz equality

Given any two objects α and β, the *identity* of α and β (in the sense of Leibniz) is expressed by the truth value $\mathbf{id}^{\mathscr{A}}(\alpha, \beta) \in \mathscr{A}$ defined by

$$\mathbf{id}^{\mathscr{A}}(\alpha, \beta) \ := \ \begin{cases} \mathbf{I}^{\mathscr{A}} & \text{if } \alpha = \beta \\ \top \to \bot & \text{if } \alpha \neq \beta \end{cases}$$

It is a straightforward exercise to check that when α and β belong to a given set M, the above interpretation of Leibniz equality amounts to the usual second-order encoding:

https://doi.org/10.1017/S0960129520000079 Published online by Cambridge University Press

Proposition 3.17. *For all sets M and for all $\alpha, \beta \in M$, we have*

$$\mathbf{id}^{\mathscr{A}}(\alpha, \beta) = \bigcurlywedge_{p \in \mathscr{A}^M} (p(\alpha) \to p(\beta))\,.$$

Moreover,

Proposition 3.18 (**Rules for $\mathbf{id}^{\mathscr{A}}$**). *Given a set M, a function $p : M \to \mathscr{A}$ and two objects $\alpha, \beta \in M$, the following semantic typing rules are valid:*

$$\frac{}{\Gamma \vdash \lambda x\,.\,x : \alpha = \alpha} \qquad\qquad \frac{\Gamma \vdash t : \mathbf{id}^{\mathscr{A}}(\alpha, \beta) \quad \Gamma \vdash u : p(\alpha)}{\Gamma \vdash t\,u : p(\beta)}$$

3.4.4 Interpreting a first-order language

Let $\mathscr{A} = (\mathscr{A}, \preccurlyeq, \to)$ be an implicative structure. An $\mathscr{A}$-valued interpretation of a first-order language $\mathscr{L}$ is defined by

— a domain of interpretation $M \neq \varnothing$;
— an M-valued function $f^M : M^k \to M$ for each k-ary function symbol of $\mathscr{L}$;
— a truth-value function $p^{\mathscr{A}} : M^k \to \mathscr{A}$ for each k-ary predicate symbol of $\mathscr{L}$.

As usual, we call a *term with parameters in M* (respectively, a *formula with parameters in M*) any first-order term (respectively, any formula) of the first-order language $\mathscr{L}$ enriched with constant symbols taken in M. Each closed term t with parameters in M is naturally interpreted as the element $t^M \in M$ defined from the equations

$$a^M = a \quad \text{(if } a \text{ is a parameter)} \qquad\qquad f(t_1, \ldots, t_k)^M = f^M(t_1^M, \ldots, t_k^M)$$

whereas each closed formula ϕ with parameters in M is interpreted as the truth value $\phi^{\mathscr{A}} \in \mathscr{A}$ defined from the equations:

$$\begin{array}{rclrcl}
(t_1 = t_2)^{\mathscr{A}} & := & \mathbf{id}^{\mathscr{A}}(t_1^M, t_2^M) & (p(t_1, \ldots, t_k))^{\mathscr{A}} & := & p^{\mathscr{A}}(t_1^M, \ldots, t_k^M) \\
(\phi \Rightarrow \psi)^{\mathscr{A}} & := & \phi^{\mathscr{A}} \to \psi^{\mathscr{A}} & (\neg\phi)^{\mathscr{A}} & := & \phi^{\mathscr{A}} \to \bot \\
(\phi \wedge \psi)^{\mathscr{A}} & := & \phi^{\mathscr{A}} \times \psi^{\mathscr{A}} & (\phi \vee \psi)^{\mathscr{A}} & := & \phi^{\mathscr{A}} + \psi^{\mathscr{A}} \\
(\forall x\, \phi(x))^{\mathscr{A}} & := & \displaystyle\bigforall_{\alpha \in M} (\phi(\alpha))^{\mathscr{A}} & (\exists x\, \phi(x))^{\mathscr{A}} & := & \displaystyle\bigexists_{\alpha \in M} (\phi(\alpha))^{\mathscr{A}}
\end{array}$$

Proposition 3.19 (**Soundness**). *If a closed formula ϕ of the language $\mathscr{L}$ is an intuitionistic tautology (respectively, a classical tautology), then*

$$\phi^{\mathscr{A}} \in S_J^0(\mathscr{A}) \qquad \textit{(respectively, } \phi^{\mathscr{A}} \in S_K^0(\mathscr{A}))$$

where $S_J^0(\mathscr{A})$ (respectively, $S_K^0(\mathscr{A})$) is the intuitionistic core (respectively, the classical core) of $\mathscr{A}$.

Proof. By induction on the derivation d of the formula ϕ (in natural deduction), we construct a closed λ-term t (possibly containing the constant cc when the derivation d is classical) such that $\vdash t : \phi^{\mathscr{A}}$, using the semantic typing rules given in Propositions 2.23, 3.12, 3.14, 3.16, and 3.18. So that $t^{\mathscr{A}} \preccurlyeq \phi^{\mathscr{A}}$. We conclude by Proposition 3.4. □

https://doi.org/10.1017/S0960129520000079 Published online by Cambridge University Press

3.5 Entailment and the induced HA

Let $(\mathscr{A}, \preccurlyeq, \to)$ be an implicative structure. Each separator $S \subseteq \mathscr{A}$ induces a binary relation of *entailment*, written $a \vdash_S b$ and defined by

$$a \vdash_S b \quad :\Leftrightarrow \quad (a \to b) \in S \qquad \text{(for all } a, b \in \mathscr{A})$$

Proposition 3.20. *The relation $a \vdash_S b$ is a preorder on $\mathscr{A}$.*

Proof. Reflexivity: given $a \in \mathscr{A}$, we have $\mathbf{I}^{\mathscr{A}} \preccurlyeq (a \to a) \in S$. Transitivity: given $a, b, c \in \mathscr{A}$ such that $(a \to b) \in S$ and $(b \to c) \in S$, we observe that $\mathbf{B}^{\mathscr{A}} = (\lambda xyz\,.\,x(yz))^{\mathscr{A}} \preccurlyeq (b \to c) \to (a \to b) \to a \to c \in S$, hence $(a \to c) \in S$, by modus ponens. □

In what follows, we shall write $\mathscr{A}/S = (\mathscr{A}/S, \leq_S)$ the *poset reflection* of the preordered set $(\mathscr{A}, \vdash_S)$, where

- $\mathscr{A}/S := \mathscr{A}/{\dashv\vdash_S}$ is the quotient of $\mathscr{A}$ by the equivalence relation $a \dashv\vdash_S b$ induced by the preorder $a \vdash_S b$, which is defined by
$$a \dashv\vdash_S b \quad :\Leftrightarrow \quad (a \to b) \in S \wedge (b \to a) \in S \qquad \text{(for all } a, b \in \mathscr{A})$$
- $\alpha \leq_S \beta$ is the order induced by the preorder $a \vdash_S b$ in the quotient set $\mathscr{A}/S$, which is characterized by
$$[a] \leq_S [b] \quad \Leftrightarrow \quad a \vdash_S b \qquad \text{(for all } a, b \in \mathscr{A})$$
writing $[a]$, $[b]$ the equivalence classes of $a, b \in \mathscr{A}$ in the quotient $\mathscr{A}/S$.

Proposition 3.21 (**Induced HA**). *For each separator $S \subseteq \mathscr{A}$, the poset reflection $H := (\mathscr{A}/S, \leq_S)$ of the preordered set $(\mathscr{A}, \vdash_S)$ is an HA whose operations are given for all $a, b \in \mathscr{A}$ by*

$$\begin{aligned} [a] \to_H [b] &= [a \to b] \\ [a] \wedge_H [b] &= [a \times b] \qquad \top_H = [\top] = S \\ [a] \vee_H [b] &= [a + b] \qquad \bot_H = [\bot] = \{c \in \mathscr{A} : (\neg c) \in S\} \end{aligned}$$

(writing $[a]$ the equivalence class of a). If, moreover, the separator $S \subseteq \mathscr{A}$ is classical, then the induced HA $H = (\mathscr{A}/S, \leq_S)$ is a Boolean algebra.

In what follows, the quotient poset $H := (\mathscr{A}/S, \leq_S)$ is called the *HA induced by the implicative algebra* $(\mathscr{A}, \preccurlyeq, \to, S)$.

Proof. Given $a, b \in \mathscr{A}$, we observe the following.

- For all $c \in \mathscr{A}$, we have $\mathbf{I}^{\mathscr{A}} \preccurlyeq (\bot \to c) \in S$, hence $[\bot] \leq_S [c]$.
- For all $c \in \mathscr{A}$, we have $(c \to \top) = \top \in S$, hence $[c] \leq_S [\top]$.
- $(\lambda z\,.\,z\,(\lambda xy\,.\,x))^{\mathscr{A}} \preccurlyeq (a \times b \to a) \in S$ and $(\lambda z\,.\,z\,(\lambda xy\,.\,y))^{\mathscr{A}} \preccurlyeq (a \times b \to b)$, hence $[a \times b] \leq_S [a]$ and $[a \times b] \leq_S [b]$. Conversely, if $c \in \mathscr{A}$ is such that $[c] \leq_S [a]$ and $[c] \leq_S [b]$, we have $(c \to a) \in S$ and $(c \to b) \in S$. From Propositions 3.4 and 2.12 (2), we get $(\lambda zw\,.\,w\,((c \to a)\,z)\,((c \to b)\,z))^{\mathscr{A}} \preccurlyeq (c \to a \times b) \in S$, hence $[c] \leq_S [a \times b]$. Therefore, $[a \times b] = \inf_H([a], [b]) = [a] \wedge_H [b]$.
- $(\lambda xzw\,.\,z\,x)^{\mathscr{A}} \preccurlyeq (a \to a + b) \in S$ and $(\lambda yzw\,.\,w\,y)^{\mathscr{A}} \preccurlyeq (b \to a + b) \in S$, hence $[a] \leq_S [a + b]$ and $[b] \leq_S [a + b]$. Conversely, if $c \in \mathscr{A}$ is such that $[a] \leq_S [c]$ and $[b] \leq_S [c]$, we have $(a \to c) \in S$ and $(b \to c) \in S$. From Proposition 3.4, we get $(\lambda z\,.\,z\,(a \to c)\,(b \to c))^{\mathscr{A}} \preccurlyeq (a + b \to c) \in S$, hence $[a + b] \leq_S [c]$. Therefore, $[a + b] = \sup_H([a], [b]) = [a] \vee_H [b]$.

https://doi.org/10.1017/S0960129520000079 Published online by Cambridge University Press

- For all $c \in \mathscr{A}$, we have $(\lambda wz\,.\,z\,w)^{\mathscr{A}} \preccurlyeq ((c \to a \to b) \to c \times a \to b) \in S$ and $(\lambda wxy\,.\,w\,\langle x, y\rangle)^{\mathscr{A}} \preccurlyeq ((c \times a \to b) \to c \to a \to b) \in S$. Hence, the equivalence $(c \to a \to b) \in S$ iff $(c \times a \to b) \in S$, that is, $[c] \leq_S [a \to b]$ iff $[c \times a] \leq_S [b]$. Therefore, $[a \to b] = \max\{\gamma \in H : \gamma \wedge_H [a] \leq_S [b]\} = [a] \to_H [b]$.

So that the poset $(\mathscr{A}/S, \leq_S)$ is an HA. If, moreover, the separator $S \subseteq \mathscr{A}$ is classical, then we have $\text{cc}^{\mathscr{A}} \preccurlyeq (\neg\neg a \to a) \in S$ for all $a \in \mathscr{A}$, so that $\neg_H \neg_H [a] = [\neg\neg a] \leq_S [a]$, which means that $(\mathscr{A}/S, \leq_S)$ is a Boolean algebra. □

Remarks 3.22. (1) In the particular case where $(\mathscr{A}, \preccurlyeq, \to)$ is a complete HA (Section 2.2.1), the separator $S \subseteq \mathscr{A}$ is a filter, and the above construction amounts to the usual construction of the quotient $\mathscr{A}/S$ in HAs.

(2) Coming back to the general framework of implicative structures, it is clear that the induced HA $H = (\mathscr{A}/S, \leq_S)$ is non-degenerate (i.e., $[\top] \neq [\bot]$) if and only if the separator $S \subseteq \mathscr{A}$ is consistent (i.e., $\bot \notin S$).

(3) When the separator $S \subseteq \mathscr{A}$ is classical (i.e., when $\text{cc}^{\mathscr{A}} \in S$), the induced HA is a Boolean algebra. The converse implication does not hold in general, and we shall see a counter example in Section 3.6 (Remark 3.26).

(4) In general, the induced HA $(\mathscr{A}/S, \leq_S)$ is not complete – so that it is not an implicative structure either. A simple counter example is given by the complete Boolean algebra $\mathfrak{P}(\omega)$ (which is also an implicative structure) equipped with the Fréchet filter $F = \{a \in \mathfrak{P}(\omega) : a \text{ cofinite}\}$ (which is also a classical separator of $\mathfrak{P}(\omega)$), since the quotient Boolean algebra $\mathfrak{P}(\omega)/F$ is not complete (Koppelberg, 1989, Chapter 2, Section 5.5).

3.6 Ultraseparators

Let $\mathscr{A} = (\mathscr{A}, \preccurlyeq, \to)$ be an implicative structure. Although the separators of $\mathscr{A}$ are in general not filters, they can be manipulated similarly to filters. By analogy with the notion of ultrafilter, we define the notion of ultraseparator:

Definition 3.23 (**Ultraseparator**). *We call an* ultraseparator *of $\mathscr{A}$ any separator $S \subseteq \mathscr{A}$ that is both consistent and maximal among consistent separators (w.r.t. $\subseteq$).*

From Zorn's lemma, it is clear that

Lemma 3.24. *For each consistent separator $S_0 \subseteq \mathscr{A}$, there exists an ultraseparator $S \subseteq \mathscr{A}$ such that $S_0 \subseteq S$.*

Proposition 3.25. *For each separator $S \subseteq \mathscr{A}$, the following are equivalent.*

(1) S is an ultraseparator of $\mathscr{A}$.
(2) The induced HA $(\mathscr{A}/S, \leq_S)$ is the two-element Boolean algebra.

Proof. (1) ⇒ (2) Assume that $S \subseteq \mathscr{A}$ is an ultraseparator. Since S is consistent, we have $\bot \notin S$ and thus $[\bot] \neq [\top]$ $(= S)$. Now, take $a_0 \in \mathscr{A}$ such that $[a_0] \neq [\bot]$, and let $S' = \{a \in \mathscr{A} : [a_0] \leq_S [a]\} = \{a \in \mathscr{A} : (a_0 \to a) \in S\}$ be the preimage of the principal filter $\uparrow[a_0] \subseteq \mathscr{A}/S$ via the canonical surjection $[\,\cdot\,] : \mathscr{A} \to \mathscr{A}/S$. Clearly, the subset $S' \subseteq \mathscr{A}$ is a consistent separator such that $S \subseteq S'$ and $a_0 \in S'$. By maximality, we have $S' = S$, so that $a_0 \in S$ and thus $[a_0] = [\top]$. Therefore, $\mathscr{A}/S = \{[\bot], [\top]\}$ is the two-element HA, that is also a Boolean algebra.

https://doi.org/10.1017/S0960129520000079 Published online by Cambridge University Press

(2) ⇒ (1) Let us assume that $\mathscr{A}/S$ is the two-element Boolean algebra (so that $\mathscr{A}/S = \{[\bot], [\top]\}$), and consider a consistent separator $S' \subseteq \mathscr{A}$ such that $S \subseteq S'$. For all $a \in S'$, we have $\neg a \notin S$ (otherwise, we would have $a, \neg a \in S'$, and thus $\bot \in S'$), hence $a \notin [\bot]$ and thus $a \in [\top] = S$. Therefore, $S' = S$. □

Remark 3.26. It is important to notice that a maximal separator is not necessarily classical, although the induced HA is always the trivial *Boolean* algebra. Indeed, we have seen in Section 3.2.2 that any total CA $(P, \cdot, \mathtt{k}, \mathtt{s})$ induces an implicative algebra $(\mathscr{A}, \preccurlyeq, \to, S) = (\mathfrak{P}(P), \subseteq, \to, \mathfrak{P}(P) \setminus \{\varnothing\})$ whose separator $S := \mathfrak{P}(P) \setminus \{\varnothing\} = \mathscr{A} \setminus \{\bot\}$ is obviously an ultraseparator. But when the set P has more than one element, it is easy to check that

$$\text{cc}^{\mathscr{A}} \preccurlyeq \bigcurlywedge_{a \in \mathscr{A}} (\neg\neg a \to a) = \bot \, (= \varnothing)$$

so that $\text{cc}^{\mathscr{A}} = \bot \notin S$. On the other hand, the induced HA $\mathscr{A}/S$ is the trivial Boolean algebra, which corresponds to the well-known fact that, in intuitionistic realizability, one of both formulas ϕ and $\neg\phi$ is realized for each *closed* formula ϕ. So that all the closed instances of the law of excluded middle are actually realized. Of course, this does not imply that the law of excluded middle itself – that holds for all *open* formulas – is (uniformly) realized. By the way, this example also shows that a non-classical separator $S \subseteq \mathscr{A}$ may induce a Boolean algebra (see Remark 3.22 (3)).

3.7 Separators, filters, and non-deterministic choice

Like filters, separators are upwards closed and nonempty, but they are not closed under binary meets in general. In this section, we shall now study the particular case of separators that happen to be filters.

3.7.1 Non-deterministic choice

Given an implicative structure $\mathscr{A} = (\mathscr{A}, \preccurlyeq, \to)$, we let

$$\pitchfork^{\mathscr{A}} := (\lambda xy\,.\,x)^{\mathscr{A}} \curlywedge (\lambda xy\,.\,y)^{\mathscr{A}} = \bigcurlywedge_{a,b \in \mathscr{A}} (a \to b \to a \curlywedge b)\,.$$

By construction, we have

$$\pitchfork^{\mathscr{A}} a\, b \preccurlyeq a \qquad \text{and} \qquad \pitchfork^{\mathscr{A}} a\, b \preccurlyeq b \qquad\qquad (\text{for all } a, b \in \mathscr{A})$$

so that we can think of $\pitchfork^{\mathscr{A}}$ as the *non-deterministic choice operator* (in $\mathscr{A}$), that takes two arguments $a, b \in \mathscr{A}$ and returns a or b in an non-deterministic way.[6]

From the point of view of logic, recall that the meet $a \curlywedge b$ of two elements $a, b \in \mathscr{A}$ can be seen as a strong form of conjunction. Indeed, it is clear that

$$(\lambda xz\,.\,z\,x\,x)^{\mathscr{A}} \preccurlyeq (a \curlywedge b \to a \times b) \in S$$

for all separators $S \subseteq \mathscr{A}$ and for all $a, b \in \mathscr{A}$, so that we have $a \curlywedge b \vdash_S a \times b$. Seen as a type, the non-deterministic choice operator $\pitchfork^{\mathscr{A}} = \bigcurlywedge_{a,b} (a \to b \to a \curlywedge b)$ precisely expresses the converse implication, and we easily check that

Proposition 3.27 **(Characterizing filters).** *For all separators $S \subseteq \mathscr{A}$, the following assertions are equivalent:*

(1) $\pitchfork^{\mathscr{A}} \in S$;
(2) $[a \curlywedge b]_{/S} = [a \times b]_{/S}$ for all $a, b \in \mathscr{A}$;
(3) S is a filter (w.r.t. the ordering $\preccurlyeq$).

https://doi.org/10.1017/S0960129520000079 Published online by Cambridge University Press

Proof. (1) $\Rightarrow$ (2) For all $a, b \in \mathscr{A}$, it is clear that $[a \curlywedge b]_{/S} \leq_S [a \times b]_{/S}$. And from (1), we get $\big(\lambda z\,.\,z\,⫛^{\mathscr{A}}\big)^{\mathscr{A}} \preccurlyeq (a \times b \to a \curlywedge b) \in S$, hence $[a \times b]_{/S} \leq_S [a \curlywedge b]_{/S}$.

(2) $\Rightarrow$ (3) Let us assume that $a, b \in S$. We have $[a]_{/S} = [b]_{/S} = [\top]_{/S}$, so that by (2) we get $[a \curlywedge b]_{/S} = [a \times b]_{/S} = [\top \times \top]_{/S} = [\top]_{/S}$. Therefore, $(a \curlywedge b) \in S$.

(3) $\Rightarrow$ (1) It is clear that $(\lambda xy\,.\,x)^{\mathscr{A}} \in S$ and $(\lambda xy\,.\,y)^{\mathscr{A}} \in S$, so that from (3) we get $⫛^{\mathscr{A}} = (\lambda xy\,.\,x)^{\mathscr{A}} \curlywedge (\lambda xy\,.\,y)^{\mathscr{A}} \in S$. □

3.7.2 Non-deterministic choice and induction

In second-order logic (Girard et al., 1989; Krivine, 1993), the predicate $\mathsf{N}(x)$ expressing that a given individual x is a natural number[7] is given by

$$\mathsf{N}(x) \;:=\; \forall Z\,(Z(0) \Rightarrow \forall y\,(Z(y) \Rightarrow Z(y+1)) \Rightarrow Z(x))\,.$$

In intuitionistic realizability (Krivine, 1993; van Oosten, 2008) as in classical realizability (Krivine, 2009), it is well known that the (unrelativized) induction principle $\textsc{Ind} := \forall x\,\mathsf{N}(x)$ is not realized in general, even when individuals are interpreted by natural numbers in the model. (Technically, this is the reason why uniform quantifications over the set of natural numbers need to be replaced by quantifications relativized to the predicate $\mathsf{N}(x)$.)

In any implicative structure $\mathscr{A} = (\mathscr{A}, \preccurlyeq, \to)$, the syntactic predicate $\mathsf{N}(x)$ is naturally interpreted by the semantic predicate $\mathsf{N}^{\mathscr{A}} : \omega \to \mathscr{A}$ defined by

$$\mathsf{N}^{\mathscr{A}}(n) \;:=\; \bigcurlywedge_{a \in \mathscr{A}^{\omega}} \Big(a_0 \to \bigcurlywedge_{i \in \omega}\big(a_i \to a_{i+1}\big) \to a_n\Big) \qquad \text{(for all } n \in \omega\text{)}$$

while the (unrelativized) induction scheme is interpreted by the truth value

$$\textsc{Ind}^{\mathscr{A}} \;:=\; \bigcurlywedge_{n \in \omega} \mathsf{N}^{\mathscr{A}}(n)\,.$$

The following proposition states that the unrelativized induction scheme $\textsc{Ind}^{\mathscr{A}}$ and the non-deterministic choice operator $⫛^{\mathscr{A}}$ are intuitionistically equivalent in $\mathscr{A}$:

Proposition 3.28. $\textsc{Ind}^{\mathscr{A}} \dashv\vdash_{S^0_J(\mathscr{A})} ⫛^{\mathscr{A}}$ *(where $S^0_J(\mathscr{A})$ is the intuitionistic core of $\mathscr{A}$).*

Proof. ($\textsc{Ind}^{\mathscr{A}} \vdash_{S^0_J(\mathscr{A})} ⫛^{\mathscr{A}}$) Given $a, b \in \mathscr{A}$, we let $c_0 = a$ and $c_n = b$ for all $n \geq 1$. From an obvious argument of subtyping, we get

$$\textsc{Ind}^{\mathscr{A}} \;\preccurlyeq\; \bigcurlywedge_{n \in \omega}\Big(c_0 \to \bigcurlywedge_{i \in \omega}\big(c_i \to c_{i+1}\big) \to c_n\Big) \;=\; a \to ((a \to b) \curlywedge (b \to b)) \to a \curlywedge b$$

so that $(\lambda nxy\,.\,n\,x\,(\mathbf{K}\,y))^{\mathscr{A}} \preccurlyeq (\textsc{Ind}^{\mathscr{A}} \to a \to b \to a \curlywedge b)$. Now taking the meet for all $a, b \in \mathscr{A}$, we thus get $(\lambda nxy\,.\,n\,x\,(\mathbf{K}\,y))^{\mathscr{A}} \preccurlyeq (\textsc{Ind}^{\mathscr{A}} \to ⫛^{\mathscr{A}}) \in S^0_J(\mathscr{A})$.

($⫛^{\mathscr{A}} \vdash_{S^0_J(\mathscr{A})} \textsc{Ind}^{\mathscr{A}}$) Consider the following pure λ-terms:

$$\begin{aligned}
\mathbf{zero} &:= \lambda xy\,.\,x \\
\mathbf{succ} &:= \lambda nxy\,.\,y\,(n\,x\,y) \\
\mathbf{Y} &:= (\lambda yf\,.\,f\,(y\,y\,f))\,(\lambda yf\,.\,f\,(y\,y\,f)) \\
t[x] &:= \mathbf{Y}\,(\lambda r\,.\,x\,\mathbf{zero}\,(\mathbf{succ}\,r))
\end{aligned}$$

https://doi.org/10.1017/S0960129520000079 Published online by Cambridge University Press

(here, **Y** is Turing's fixpoint combinator). From the typing rules of Proposition 2.23, we easily check that $\mathbf{zero}^{\mathscr{A}} \preccurlyeq \mathsf{N}(0)$ and $\mathbf{succ}^{\mathscr{A}} \preccurlyeq \mathsf{N}(n) \to \mathsf{N}(n+1)$ for all $n \in \omega$. Now, consider the element $\Theta := \big(t[\pitchfork^{\mathscr{A}}]\big)^{\mathscr{A}} \in \mathscr{A}$. From the reduction rule of **Y**, we get

$$\Theta \ \preccurlyeq\ \pitchfork^{\mathscr{A}} \mathbf{zero}^{\mathscr{A}} (\mathbf{succ}^{\mathscr{A}}\, \Theta) \ \preccurlyeq\ \mathbf{zero}^{\mathscr{A}} \curlywedge \mathbf{succ}^{\mathscr{A}}\, \Theta\,.$$

By a straightforward induction on n, we deduce that $\Theta \preccurlyeq \mathsf{N}(n)$ for all $n \in \omega$, hence $\Theta \preccurlyeq \textsc{Ind}^{\mathscr{A}}$. Therefore, $(\lambda x\,.\,t[x])^{\mathscr{A}} \preccurlyeq (\pitchfork^{\mathscr{A}} \to \Theta) \preccurlyeq (\pitchfork^{\mathscr{A}} \to \textsc{Ind}^{\mathscr{A}}) \in S_J^0(\mathscr{A})$. □

3.7.3 Non-deterministic choice and the parallel-or

A variant of the non-deterministic choice operator is the *parallel "or,"* that is defined by

$$\mathbf{p\text{-}or}^{\mathscr{A}} \ := \ (\bot \to \top \to \bot) \curlywedge (\top \to \bot \to \bot)\,.$$

Intuitively, the parallel "or" is a function that takes two arguments – one totally defined and the other one totally undefined – and returns the most defined of both, independently from the order in which both arguments were passed to the function. (Recall that according to the definitional ordering $a \sqsubseteq b :\Leftrightarrow a \succcurlyeq b$, the element $\bot$ represents the totally defined object, whereas $\top$ represents the totally undefined object.)

We observe that

$$\pitchfork^{\mathscr{A}} \ = \ \bigcurlywedge_{a,b \in \mathscr{A}} (a \to b \to a \curlywedge b) \ \preccurlyeq \ (\bot \to \top \to \bot) \curlywedge (\top \to \bot \to \bot)\,,$$

which means that the parallel "or" $\mathbf{p\text{-}or}^{\mathscr{A}}$ is a super-type of the non-deterministic choice operator $\pitchfork^{\mathscr{A}}$. However, both operators are classically equivalent.

Proposition 3.29. *$\mathbf{p\text{-}or}^{\mathscr{A}} \dashv\vdash_{S_K^0(\mathscr{A})} \pitchfork^{\mathscr{A}}$ (where $S_K^0(\mathscr{A})$ is the classical core of $\mathscr{A}$).*

Proof. ($\pitchfork^{\mathscr{A}} \vdash_{S_K^0(\mathscr{A})} \mathbf{p\text{-}or}^{\mathscr{A}}$) Obvious, by subtyping.

($\mathbf{p\text{-}or}^{\mathscr{A}} \vdash_{S_K^0(\mathscr{A})} \pitchfork^{\mathscr{A}}$) Let $t := \lambda zxy\,.\,\text{œ}\,(\lambda k\,.\,z\,(k\,x)\,(k\,y))$. From the semantic typing rules of Proposition 2.23 (and from the type of œ), we easily check that

$$t^{\mathscr{A}} \preccurlyeq (\mathbf{p\text{-}or}^{\mathscr{A}} \to a \to b \to a) \quad \text{and} \quad t^{\mathscr{A}} \preccurlyeq (\mathbf{p\text{-}or}^{\mathscr{A}} \to a \to b \to b)$$

for all $a, b \in \mathscr{A}$, hence $t^{\mathscr{A}} \preccurlyeq (\mathbf{p\text{-}or}^{\mathscr{A}} \to \pitchfork^{\mathscr{A}}) \in S_K^0(\mathscr{A})$. □

3.7.4 The case of finitely generated separators

In Proposition 3.27, we have seen that a separator $S \subseteq \mathscr{A}$ is a filter if and only if it contains the non-deterministic choice operator $\pitchfork^{\mathscr{A}}$. In the particular case where the separator $S \subseteq \mathscr{A}$ is finitely generated (see Section 3.3), the situation is even more dramatic:

Proposition 3.30. *Given a separator $S \subseteq \mathscr{A}$, the following are equivalent.*

(1) S is finitely generated and $\pitchfork^{\mathscr{A}} \in S$.
(2) S is a principal filter of $\mathscr{A}$: $\quad S = \uparrow\{\Theta\} \quad$ for some $\Theta \in S$.
(3) The induced HA $(\mathscr{A}/S, \leq_S)$ is complete, and the canonical surjection $[\,\cdot\,]_{/S} : \mathscr{A} \to \mathscr{A}/S$ commutes with arbitrary meets:

$$\Big[\bigcurlywedge_{i \in I} a_i\Big]_{/S} \ = \ \bigwedge_{i \in I} [a_i]_{/S} \qquad (\text{for all } (a_i)_{i \in I} \in \mathscr{A}^I)$$

https://doi.org/10.1017/S0960129520000079 Published online by Cambridge University Press

Proof. (1) ⇒ (2) Let us assume that $S = \uparrow@(\{g_1, \dots, g_n\})$ for some $g_1, \dots, g_n \in S$ (see Section 3.3, Proposition 3.9), and $⫛^{\mathscr{A}} \in S$. From the latter assumption, we know (by Proposition 3.27) that S is closed under all finite meets, so that for all $k \geq 1$, we have

$$⫛_k^{\mathscr{A}} \;:=\; \bigwedge_{i=1}^{k} (\lambda x_1 \cdots x_k \,.\, x_i)^{\mathscr{A}} \;=\; \bigwedge_{a_1,\dots,a_k \in \mathscr{A}} (a_1 \to \cdots \to a_k \to a_1 \curlywedge \cdots \curlywedge a_k) \;\in\; S\,.$$

Let $\Theta := \big(\mathbf{Y}\,(\lambda r\,.\,⫛_{n+1}^{\mathscr{A}} g_1 \cdots g_n\,(r\,r))\big)^{\mathscr{A}}$, where $\mathbf{Y} := (\lambda yf\,.f\,(y\,y\,f))\,(\lambda yf\,.f\,(y\,y\,f))$ is Turing's fixpoint combinator. Since $g_1, \dots, g_n, ⫛_{n+1}^{\mathscr{A}} \in S$, it is clear that $\Theta \in S$. From the evaluation rule of $\mathbf{Y}$, we have $\Theta \preccurlyeq ⫛_{n+1}^{\mathscr{A}} g_1 \cdots g_n\,(\Theta\Theta) \preccurlyeq g_1 \curlywedge \cdots \curlywedge g_n \curlywedge \Theta\Theta$, hence $\Theta \preccurlyeq g_i$ for all $i \in \{1, \dots, n\}$ and $\Theta \preccurlyeq \Theta\Theta$. By a straightforward induction, we deduce that $\Theta \preccurlyeq a$ for all $a \in @(\{g_1, \dots, g_n\})$ (recall that the latter set is generated from $g_1, \dots, g_n$ by application), and thus $\Theta \preccurlyeq a$ for all $a \in \uparrow@(\{g_1, \dots, g_n\}) = S$ (by upwards closure). Therefore, $\Theta = \min(S)$ and $S = \uparrow\{\Theta\}$ (since S is upwards closed).

(2) ⇒ (3) Let us assume that $S = \uparrow\{\Theta\}$ for some $\Theta \in S$. Let $(\alpha_i)_{i\in I} \in (\mathscr{A}/S)^I$ be a family of equivalence classes indexed by an arbitrary set I, and $(a_i)_{i\in I} \in \prod_{i\in I} \alpha_i$ a system of representatives. Since $\big(\bigwedge_{i\in I} a_i\big) \preccurlyeq a_i$ for all $i \in I$, we have $\big[\bigwedge_{i\in I} a_i\big]_{/S} \leq_S \alpha_i$ for all $i \in I$, hence $\big[\bigwedge_{i\in I} a_i\big]_{/S}$ is a lower bound of $(\alpha_i)_{i\in I}$ in $\mathscr{A}/S$. Now, let us assume that $\beta = [b]_{/S}$ is a lower bound of $(\alpha_i)_{i\in I}$ in $\mathscr{A}/S$, which means that $(b \to a_i) \in S$ for all $i \in I$. But since $S = \uparrow\{\Theta\}$, we have $\Theta \preccurlyeq (b \to a_i)$ for all $i \in I$, hence $\Theta \preccurlyeq \big(b \to \bigwedge_{i\in I} a_i\big)$, so that $\beta = [b]_{/S} \leq_S \big[\bigwedge_{i\in I} a_i\big]_{/S}$. Therefore, $\big[\bigwedge_{i\in I} a_i\big]_{/S}$ is the g.l.b. of the family $(\alpha_i)_{i\in I} = \big([a_i]_{/S}\big)_{i\in I}$ in $\mathscr{A}/S$. This proves that the induced HA $(\mathscr{A}/S, \leq_S)$ is complete, as well as the desired commutation property.

(3) ⇒ (2) ⇒ (1) Let us assume that the HA $(\mathscr{A}/S, \leq_S)$ is complete, and that the canonical surjection $[\,\cdot\,]_{/S} : \mathscr{A} \to \mathscr{A}/S$ commutes with arbitrary meets. Letting $\Theta := \bigwedge S$, we observe that

$$[\Theta]_{/S} \;=\; \Big[\bigwedge_{a\in S} a\Big]_{/S} \;=\; \bigwedge_{a\in S} [a]_{/S} \;=\; [\top]_{/S}\,,$$

hence, $\Theta \in S$. Therefore, $\Theta = \min(S)$ and $S = \uparrow\{\Theta\}$ (since S is upwards closed), which shows that S is the principal filter generated by Θ. But this implies that S is finitely generated (we obviously have $S = \mathrm{Sep}(\{\Theta\})$) and that $⫛^{\mathscr{A}} \in S$ (by Proposition 3.27). □

Remark 3.31. From a categorical perspective, the situation described by Proposition 3.30 is particularly important, since it characterizes the collapse of realizability to forcing. Indeed, we shall see in Section 4.5 (Theorem 4.13) that the tripos induced by an implicative algebra $(\mathscr{A}, \preccurlyeq, \to, S)$ (Section 4.4) is isomorphic to a forcing tripos (induced by some complete HA) if and only if the separator $S \subseteq \mathscr{A}$ is a principal filter of $\mathscr{A}$, that is, if and only if the separator S is finitely generated and contains the non-deterministic choice operator $⫛^{\mathscr{A}}$.

3.8 On the interpretation of existential quantification as a join

In Section 3.4, we have seen that existential quantifications cannot be interpreted by (infinitary) joins in the general framework of implicative structures. (We shall actually present a counter example at the end of this section.) Using the material presented in Section 3.7, we shall now study the particular class of implicative structures where existential quantifications are naturally interpreted by joins.

https://doi.org/10.1017/S0960129520000079 Published online by Cambridge University Press

Formally, we say that an implicative structure $\mathscr{A} = (\mathscr{A}, \preccurlyeq, \to)$ is *compatible with joins* when it fulfills the additional axiom

$$\bigcurlywedge_{a\in A}(a \to b) \;=\; \Big(\bigcurlyvee_{a\in A} a\Big) \to b$$

for all subsets $A \subseteq \mathscr{A}$ and for all $b \in \mathscr{A}$. (Note that the converse relation $\succcurlyeq$ holds in any implicative structure, so that only the direct relation $\preccurlyeq$ matters.)

This axiom obviously holds in any complete HA (or Boolean algebra), as well as in any implicative structure induced by a total CA $(P, \cdot, \mathtt{k}, \mathtt{s})$ (Section 2.7.1). On the other hand, the implicative structures induced by classical realizability (Section 2.7.2) are in general *not* compatible with joins, as we shall see below.

When an implicative structure $\mathscr{A} = (\mathscr{A}, \preccurlyeq, \to)$ is compatible with joins, the existential quantifier can be interpreted as a join

$$\exists_{i\in I} a_i \;:=\; \bigcurlyvee_{i\in I} a_i$$

since the corresponding elimination rule is directly given by the subtyping relation

$$\bigcurlywedge_{i\in I}(a_i \to b) \;\preccurlyeq\; \Big(\bigcurlyvee_{i\in I} a_i\Big) \to b\,.$$

In this situation, we can also observe many simplifications at the level of the defined connectives $\times$ and $+$:

Proposition 3.32. *If an implicative structure $\mathscr{A} = (\mathscr{A}, \preccurlyeq, \to)$ is compatible with joins, then for all $a \in \mathscr{A}$, we have*

$$\begin{array}{rclcrcl} \bot \to a &=& \top & \qquad & \mathbf{p\text{-}or}^{\mathscr{A}} &=& \top \\ a \times \bot &=& \top \to \bot & & a + \bot &=& (\lambda xy\,.\,x\,a)^{\mathscr{A}} \\ \bot \times a &=& \top \to \bot & & \bot + a &=& (\lambda xy\,.\,y\,a)^{\mathscr{A}} \end{array}$$

Proof. Indeed, we have

- $\bot \to a = (\bigcurlyvee \varnothing) \to a = \bigcurlywedge \varnothing = \top$, from the compatibility with joins.
- $\mathbf{p\text{-}or}^{\mathscr{A}} = (\bot \to \top \to \bot) \curlywedge (\top \to \bot \to \bot) = \top \curlywedge (\top \to \top) = \top$.
- $a \times \bot = \bigcurlywedge_c ((a \to \bot \to c) \to c) = \bigcurlywedge_c (\top \to c) = \top \to \bot$.
- $\bot \times a = \bigcurlywedge_c ((\bot \to a \to c) \to c) = \bigcurlywedge_c (\top \to c) = \top \to \bot$.
- By semantic typing, we have

$$(\lambda xy\,.\,x\,a)^{\mathscr{A}} \;\preccurlyeq\; \bigcurlywedge_c ((a \to c) \to (\bot \to c) \to c) \;=\; a + \bot\,.$$

  And conversely,

$$\begin{aligned} a + \bot \;&=\; \bigcurlywedge_c ((a \to c) \to (\bot \to c) \to c) \;=\; \bigcurlywedge_c ((a \to c) \to \top \to c) \\ &\preccurlyeq\; \bigcurlywedge_{d,e} ((a \to da) \to e \to da) \;\preccurlyeq\; \bigcurlywedge_{d,e} (d \to e \to da) \;=\; (\lambda xy\,.\,x\,a)^{\mathscr{A}} \end{aligned}$$

- The equality $\bot + a = (\lambda xy\,.\,y\,a)^{\mathscr{A}}$ is proved similarly. □

https://doi.org/10.1017/S0960129520000079 Published online by Cambridge University Press

In particular, we observe a trivialization of the parallel "or": $\mathbf{p\text{-}or}^{\mathscr{A}} = \top$, so that by Proposition 3.29, we get $⋔^{\mathscr{A}} \in S_K^0(\mathscr{A})$. Therefore, by Proposition 3.27, it is clear that

Proposition 3.33. *If an implicative structure $\mathscr{A} = (\mathscr{A}, \to, \preccurlyeq)$ is compatible with joins, then all its classical separators are filters.*

Of course, this situation is highly undesirable in classical realizability (see Remark 3.31 above), and this explains why classical realizability is not and cannot be compatible with joins in general (except in the degenerate case of forcing).

Remark 3.34 (**The model of threads**). In Krivine (2012), Krivine constructs a model of Zermelo-Fraenkel set theory with the axiom of Dependent Choice (ZF + DC) from a particular AKS (see Section 2.7.2), called the *model of threads*. This particular AKS is defined in such a way that it is consistent, while providing a proof-like term $\theta \in \mathrm{PL}$ that realizes the *negation* of the parallel "or":

$$\theta \Vdash \neg((\bot \to \top \to \bot) \curlywedge (\top \to \bot \to \bot)).$$

In the induced classical implicative algebra $(\mathscr{A}, \preccurlyeq, \to, S)$ (Section 3.2.3), we thus have $\bot \notin S$ and $\neg\mathbf{p\text{-}or}^{\mathscr{A}} \in S$. Hence, $\mathbf{p\text{-}or}^{\mathscr{A}} \notin S$ and thus $⋔^{\mathscr{A}} \notin S$ (by Proposition 3.29), so that S is not a filter (Proposition 3.27). From Proposition 3.33 (by contraposition), it is then clear that the underlying implicative structure $(\mathscr{A}, \preccurlyeq, \to)$ is not compatible with joins.

4. The Implicative Tripos

In Section 3.5, we have seen that any implicative algebra $(\mathscr{A}, \preccurlyeq, \to, S)$ induces an HA $(\mathscr{A}/S, \leq_S)$ that intuitively captures the corresponding logic, at least at the propositional level. In this section, we shall see that this construction more generally gives rise to a (**Set**-based) tripos, called an *implicative tripos*. For that, we first need to present some constructions on implicative structures and on separators.

4.1 Product of implicative structures

Let $(\mathscr{A}_i)_{i\in I} = (\mathscr{A}_i, \preccurlyeq_i, \to_i)_{i\in I}$ be a family of implicative structures indexed by an arbitrary set I. The Cartesian product $\mathscr{A} := \prod_{i\in I} \mathscr{A}_i$ is naturally equipped with the ordering $(\preccurlyeq) \subseteq \mathscr{A}^2$ and the implication $(\to) : \mathscr{A}^2 \to \mathscr{A}$ that are defined componentwise:

$$(a_i)_{i\in I} \preccurlyeq (b_i)_{i\in I} \;:\Leftrightarrow\; \forall i \in I,\ a_i \preccurlyeq_i b_i \qquad \text{(product ordering)}$$

$$(a_i)_{i\in I} \to (b_i)_{i\in I} \;:=\; (a_i \to_i b_i)_{i\in I} \qquad \text{(product implication)}$$

It is straightforward to check that

Proposition 4.1. *The triple $(\mathscr{A}, \preccurlyeq, \to)$ is an implicative structure.*

In the product implicative structure $(\mathscr{A}, \preccurlyeq, \to) = \prod_{i\in I} \mathscr{A}_i$, the defined constructions $\neg a$ (negation), $a \times b$ (conjunction), $a + b$ (disjunction), ab (application), $\text{cc}^{\mathscr{A}}$ (Peirce's law), and $⋔^{\mathscr{A}}$ (non-deterministic choice) are naturally characterized componentwise:

Proposition 4.2. *For all $a, b \in \mathscr{A} = \prod_{i\in I} \mathscr{A}_i$, we have*

$$\neg a = (\neg a_i)_{i\in I} \qquad a \times b = (a_i \times b_i)_{i\in I} \qquad a + b = (a_i + b_i)_{i\in I}$$
$$ab = (a_i b_i)_{i\in I} \qquad \text{cc}^{\mathscr{A}} = \left(\text{cc}^{\mathscr{A}_i}\right)_{i\in I} \qquad ⋔^{\mathscr{A}} = \left(⋔^{\mathscr{A}_i}\right)_{i\in I}$$

https://doi.org/10.1017/S0960129520000079 Published online by Cambridge University Press

Proof. Given $a, b \in \mathscr{A}$, we have

$$a \times b = \bigcurlywedge_{c \in \mathscr{A}} ((a \to b \to c) \to c) = \bigcurlywedge_{c \in \mathscr{A}} \Big((a_i \to b_i \to c_i) \to c_i\Big)_{i \in I}$$

$$= \Big(\bigcurlywedge_{c \in \mathscr{A}_i} ((a_i \to b_i \to c) \to c)\Big)_{i \in I} = (a_i \times b_i)_{i \in I}$$

$$ab = \bigcurlywedge\{c \in \mathscr{A} : a \preccurlyeq (b \to c)\} = \bigcurlywedge \prod_{i \in I} \{c \in \mathscr{A}_i : a_i \preccurlyeq (b_i \to c)\}$$

$$= \Big(\bigcurlywedge\{c \in \mathscr{A}_i : a_i \preccurlyeq (b_i \to c)\}\Big)_{i \in I} = (a_i b_i)_{i \in I}$$

The other equalities are proved similarly. □

Proposition 4.3. *For all pure λ-terms $t(x_1, \ldots, x_k)$ with free variables $x_1, \ldots, x_k$ and for all parameters $a_1, \ldots, a_k \in \mathscr{A} = \prod_{i \in I} \mathscr{A}_i$, we have*

$$t(a_1, \ldots, a_k)^{\mathscr{A}} = \Big(t(a_{1,i}, \ldots, a_{k,i})^{\mathscr{A}_i}\Big)_{i \in I}$$

Proof. By structural induction on the term $t(x_1, \ldots, x_k)$. The case of a variable is obvious, the case of an application follows from the equality $ab = (a_i b_i)_{i \in I}$, so that we only treat the case where $t(x_1, \ldots, x_k) = \lambda x_0 \,.\, t_0(x_0, x_1, \ldots, x_k)$. In this case, we have

$$\begin{aligned}
t(a_1, \ldots, a_k)^{\mathscr{A}} &= (\lambda x_0 \,.\, t_0(x_0, a_1, \ldots, a_k))^{\mathscr{A}} \\
&= \bigcurlywedge_{a_0 \in \mathscr{A}} (a_0 \to t_0(a_0, a_1, \ldots, a_k)^{\mathscr{A}}) \\
&= \bigcurlywedge_{a_0 \in \mathscr{A}} \Big(a_{0,i} \to_i t_0\big(a_{0,i}, a_{1,i}, \ldots, a_{k,i}\big)^{\mathscr{A}_i}\Big)_{i \in I} \qquad \text{(by IH)} \\
&= \Big(\bigcurlywedge_{a_0 \in \mathscr{A}_i} \Big(a_0 \to_i t_0\big(a_0, a_{1,i}, \ldots, a_{k,i}\big)^{\mathscr{A}_i}\Big)\Big)_{i \in I} \\
&= \Big((\lambda x_0 \,.\, t_0(x_0, a_{1,i}, \ldots, a_{k,i}))^{\mathscr{A}_i}\Big)_{i \in I} = \Big(t\big(a_{1,i}, \ldots, a_{k,i}\big)^{\mathscr{A}_i}\Big)_{i \in I}
\end{aligned}$$

□

4.1.1 Product of separators

Given a family of separators $(S_i \subseteq \mathscr{A}_i)_{i \in I}$, it is clear that the Cartesian product $S = \prod_{i \in I} S_i$ is also a separator of $\mathscr{A} = \prod_{i \in I} \mathscr{A}_i$. In the product separator $S = \prod_{i \in I} S_i$, the relation of entailment $a \vdash_S b$ and the corresponding equivalence $a \dashv\vdash_S b$ are characterized by

$$\begin{aligned} a \vdash_S b &\Leftrightarrow \forall i \in I,\ a_i \vdash_{S_i} b_i \\ a \dashv\vdash_S b &\Leftrightarrow \forall i \in I,\ a_i \dashv\vdash_{S_i} b_i \end{aligned} \qquad \text{(for all } a, b \in \mathscr{A}\text{)}$$

For each index $i \in I$, the corresponding projection $\pi_i : \mathscr{A} \to \mathscr{A}_i$ factors into a map

$$\begin{aligned} \tilde{\pi}_i \ :\ \mathscr{A}/S &\to \mathscr{A}_i/S_i \\ [a]_{/S} &\mapsto [a_i]_{/S_i} \end{aligned}$$

that is obviously a morphism of HAs (from Propositions 3.21 and 4.2). In this situation, we immediately get the factorization $\mathscr{A}/S \cong \prod_{i \in I} (\mathscr{A}_i/S_i)$, since:

https://doi.org/10.1017/S0960129520000079 Published online by Cambridge University Press

Proposition 4.4. *The map*

$$\langle \tilde{\pi}_i \rangle_{i\in I} \;:\; \mathscr{A}/S \to \prod_{i\in I} (\mathscr{A}_i/S_i)$$

is an isomorphism of HAs.

Proof. For all $a, b \in \mathscr{A}$, we have

$$[a] \leq_S [b] \;\Leftrightarrow\; (a \to b) \in S \;\Leftrightarrow\; (\forall i \in I)\,(a_i \to b_i) \in S_i \;\Leftrightarrow\; (\forall i \in I)\,[a_i] \leq_{S_i} [b_i]$$

which proves that the map $\langle \tilde{\pi}_i \rangle_{i\in I} : \mathscr{A}/S \to \prod_{i\in I} (\mathscr{A}_i/S_i)$ is an embedding of the poset $(\mathscr{A}/S, \leq_S)$ into the product poset $\prod_{i\in I} (\mathscr{A}_i/S_i, \leq_{S_i})$. Moreover, the map $\langle \tilde{\pi}_i \rangle_{i\in I}$ is clearly surjective (from the axiom of choice); therefore, it is an isomorphism of posets, and thus an isomorphism of HAs. □

4.2 The uniform power separator

Let $\mathscr{A} = (\mathscr{A}, \preccurlyeq, \to)$ be a fixed implicative structure. For each set I, we write

$$\mathscr{A}^I \;=\; (\mathscr{A}^I, \preccurlyeq^I, \to^I) \;:=\; \prod_{i\in I} (\mathscr{A}, \preccurlyeq, \to)$$

the corresponding power implicative structure, which is a particular case of the product presented in Section 4.1. Each separator $S \subseteq \mathscr{A}$ induces two separators in $\mathscr{A}^I$:

- The *power separator* $S^I := \prod_{i\in I} S \subseteq \mathscr{A}^I$.
- The *uniform power separator* $S[I] \subseteq \mathscr{A}$, that is defined by

$$S[I] \;:=\; \{a \in \mathscr{A} : \exists s \in S,\ \forall i \in I,\ s \preccurlyeq a_i\} \;=\; \uparrow\!\mathrm{img}(\delta_I)\,,$$

  where $\delta_I : \mathscr{A} \to \mathscr{A}^I$ is defined by $\delta(a) = (i \mapsto a)$ for all $a \in \mathscr{A}$.

From the definition, it is clear that $S[I] \subseteq S^I \subseteq \mathscr{A}$. The converse inclusion $S^I \subseteq S[I]$ does not hold in general, and we easily check that

Proposition 4.5. *For all separators $S \subseteq \mathscr{A}$, the following are equivalent.*

(1) $S[I] = S^I$.
(2) *S is closed under all I-indexed meets.*

Proof. (1) ⇒ (2) Let $(a_i)_{i\in I}$ be an I-indexed family of elements of S, that is, an element of S^I. By (1), we have $(a_i)_{i\in I} \in S[I]$, so that there is $s \in S$ such that $s \preccurlyeq a_i$ for all $i \in I$. Therefore, $s \preccurlyeq \big(\bigcurlywedge_{i\in I} a_i\big) \in S$ (by upwards closure).

(2) ⇒ (1) Let $(a_i)_{i\in I} \in S^I$. By (2), we have $s := \big(\bigcurlywedge_{i\in I} a_i\big) \in S$, and since $s \preccurlyeq a_i$ for all $i \in I$, we get that $(a_i)_{i\in I} \in S[I]$ (by definition). Hence, $S^I = S[I]$. □

Thanks to the notion of uniform separator, we can also characterize the intuitionistic and classical cores (Section 3.3) of the power implicative structure $\mathscr{A}^I$.

Proposition 4.6. $S^0_J(\mathscr{A}^I) = S^0_J(\mathscr{A})[I]$ *and* $S^0_K(\mathscr{A}^I) = S^0_K(\mathscr{A})[I]$.

https://doi.org/10.1017/S0960129520000079 Published online by Cambridge University Press

Proof. Recall that:
$$S_J^0(\mathscr{A}) = \uparrow\{(t)^{\mathscr{A}} : t \text{ closed } \lambda\text{-term}\}$$
$$S_J^0(\mathscr{A}^I) = \uparrow\{(t)^{\mathscr{A}^I} : t \text{ closed } \lambda\text{-term}\}$$
$$S_J^0(\mathscr{A})[I] = \{a \in \mathscr{A}^I : \exists s \in S_J^0(\mathscr{A}),\ \forall i \in I,\ s \preccurlyeq a_i\}.$$

Since $S_J^0(\mathscr{A}^I)$ is the smallest separator of $\mathscr{A}^I$, we have $S_J^0(\mathscr{A}^I) \subseteq S_J^0(\mathscr{A})[I]$. Conversely, take $a \in S_J^0(\mathscr{A})[I]$. By definition, there is $s \in S_J^0(\mathscr{A})$ such that $s \preccurlyeq a_i$ for all $i \in I$. And since $s \in S_J^0(\mathscr{A})$, there is a closed λ-term t such that $(t)^{\mathscr{A}} \preccurlyeq s$, hence $(t)^{\mathscr{A}} \preccurlyeq a_i$ for all $i \in I$. From Proposition 4.3, we deduce that $(t)^{\mathscr{A}^I} = \big((t)^{\mathscr{A}}\big)_{i\in I} \preccurlyeq (a_i)_{i\in I}$ (in $\mathscr{A}^I$), hence $(a_i)_{i\in I} \in S_J^0(\mathscr{A}^I)$. The equality $S_K^0(\mathscr{A}^I) = S_K^0(\mathscr{A})[I]$ is proved similarly, using closed λ-terms with cc instead of pure λ-terms. □

In the rest of this section, we shall see that, given a separator $S \subseteq \mathscr{A}$, the correspondence $I \mapsto \mathscr{A}^I/S[I]$ (from unstructured sets to HAs) is functorial and actually constitutes a tripos.

4.3 Triposes

4.3.1 The category of HAs

Given two HAs H and H', a function $F : H \to H'$ is called a *morphism of HAs* when

$$\begin{aligned} F(a \wedge_H b) &= F(a) \wedge_{H'} F(b) & F(\top_H) &= \top_{H'} \\ F(a \vee_H b) &= F(a) \vee_{H'} F(b) & F(\bot_H) &= \bot_{H'} \\ F(a \to_H b) &= F(a) \to_{H'} F(b) \end{aligned} \qquad \text{(for all } a, b \in H)$$

(In other words, a morphism of HAs is a morphism of bounded lattices that also preserves Heyting's implication. Note that such a function is always monotonic.)

The category of HAs is the category whose objects are the HAs and whose arrows are the morphisms of HAs; it is a (non-full) subcategory of the category of posets (notation: **Pos**). This category also enjoys some specific properties that will be useful in the following:

(1) An arrow is an isomorphism in **HA** if and only if it is an isomorphism in **Pos**.
(2) Any injective morphism of HAs $F : H \to H'$ is also an embedding of posets, in the sense that: $a \le b$ iff $F(a) \le F(b)$ (for all $a, b \in H$).
(3) Any bijective morphism of HAs is also an isomorphism.

4.3.2 **Set**-based triposes

In this section, we recall the definition of **Set**-based triposes, such as initially formulated by Hyland et al. in (1980). For the general definition of triposes, where the base category **Set** can be replaced by an arbitrary Cartesian category, see for instance (Pitts 1981, 2002).

Definition 4.7 (**Set-based tripos**). *A* **Set**-based tripos *is a functor* $\mathsf{P} : \mathbf{Set}^{op} \to \mathbf{HA}$ *that fulfills the following three conditions:*

(1) For each function $f : I \to J$, the corresponding map $\mathsf{P}f : \mathsf{P}J \to \mathsf{P}I$ has left and right adjoints in **Pos** *that are monotonic maps $\exists f, \forall f : \mathsf{P}I \to \mathsf{P}J$ such that*

$$\begin{aligned} \exists f(p) \le q &\Leftrightarrow p \le \mathsf{P}f(q) \\ q \le \forall f(p) &\Leftrightarrow \mathsf{P}f(q) \le p \end{aligned} \qquad (\textit{for all } p \in \mathsf{P}I,\ q \in \mathsf{P}J)$$

https://doi.org/10.1017/S0960129520000079 Published online by Cambridge University Press

(2) *Beck–Chevalley condition. Each pullback square in* **Set** *(on the left-hand side) induces the following two commutative diagrams in* **Pos** *(on the right-hand side):*

$$\begin{array}{ccc} I & \xrightarrow{f_1} & I_1 \\ {\scriptstyle f_2}\downarrow & \lrcorner & \downarrow{\scriptstyle g_1} \\ I_2 & \xrightarrow[g_2]{} & J \end{array} \quad \Rightarrow \quad \begin{array}{ccc} \mathsf{P}I & \xrightarrow{\exists f_1} & \mathsf{P}I_1 \\ {\scriptstyle \mathsf{P}f_2}\uparrow & & \uparrow{\scriptstyle \mathsf{P}g_1} \\ \mathsf{P}I_2 & \xrightarrow[\exists g_2]{} & \mathsf{P}J \end{array} \qquad \begin{array}{ccc} \mathsf{P}I & \xrightarrow{\forall f_1} & \mathsf{P}I_1 \\ {\scriptstyle \mathsf{P}f_2}\uparrow & & \uparrow{\scriptstyle \mathsf{P}g_1} \\ \mathsf{P}I_2 & \xrightarrow[\forall g_2]{} & \mathsf{P}J \end{array}$$

That is, $\exists f_1 \circ \mathsf{P}f_2 = \mathsf{P}g_1 \circ \exists g_2$ *and* $\forall f_1 \circ \mathsf{P}f_2 = \mathsf{P}g_1 \circ \forall g_2$.

(3) *The functor* $\mathsf{P} : \mathbf{Set}^{op} \to \mathbf{HA}$ *has a* generic predicate*, that is, a predicate* $\mathsf{Tr} \in \mathsf{P}\,Prop$ *(for some set Prop) such that for all sets I, the following map is surjective:*

$$\begin{array}{rcl} Prop^I & \to & \mathsf{P}I \\ f & \mapsto & \mathsf{P}f(\mathsf{Tr}) \end{array}$$

Remarks 4.8 (**Intuitive meaning of the definition**). Intuitively, each **Set**-based tripos $\mathsf{P} : \mathbf{Set}^{\mathrm{op}} \to \mathbf{HA}$ describes a particular model of intuitionistic higher order logic, in which higher order types are modeled by sets. In this framework,

(1) the functor $\mathsf{P} : \mathbf{Set}^{\mathrm{op}} \to \mathbf{C}$ associates to each "type" $I \in \mathbf{Set}$ the poset $\mathsf{P}I$ of all *predicates* over I. The ordering on $\mathsf{P}I$ represents *inclusion* of predicates (in the sense of the considered model), whereas equality represents *extensional equality* (or *logical equivalence*). In what follows, it is convenient to think that predicates $p, q, \ldots \in \mathsf{P}I$ represent abstract formulas $p(x), q(x), \ldots$ depending on a variable $x : I$, so that

$$p \leq q \quad \text{means} \quad (\forall x \in I)(p(x) \Rightarrow q(x))$$

whereas

$$p = q \quad \text{means} \quad (\forall x \in I)(p(x) \Leftrightarrow q(x))\,.$$

The fact that $\mathsf{P}I$ is an HA simply expresses that the predicates over I can be composed using all the connectives of intuitionistic logic and that these operations fulfill the laws of intuitionistic propositional logic.

(2) the functoriality of P expresses that each function $f : I \to J$ induces a *substitution map* $\mathsf{P}f : \mathsf{P}J \to \mathsf{P}I$ that intuitively associates to each predicate $q \in \mathsf{P}J$ its "preimage" $\mathsf{P}f(q) =$ "$q \circ f$" $\in \mathsf{P}I$. Again, if we think that the predicate $q \in \mathsf{P}J$ represents an abstract formula $q(y)$ depending on a variable $y : J$, then the predicate $\mathsf{P}f(q)$ represents the substituted formula $q(y)\{y := f(x)\} \equiv q(f(x))$ (that now depends on $x : I$). The fact that the substitution map $\mathsf{P}f : \mathsf{P}J \to \mathsf{P}I$ is a morphism of HAs expresses that substitution commutes with all the logical connectives.

(3) Given a function $f : I \to J$, the left and right adjoints $\exists f, \forall f : \mathsf{P}I \to \mathsf{P}J$ represent existential and universal quantifications along the function $f : I \to J$. By this, we mean that if a predicate $p \in \mathsf{P}I$ represents a formula $p(x)$ (depending on $x : I$), then

$$\exists f(p) \quad \text{represents the formula} \quad (\exists x : I)(f(x) = y \wedge p(x))$$

whereas

$$\forall f(p) \quad \text{represents the formula} \quad (\forall x : I)(f(x) = y \Rightarrow p(x))$$

(where both right-hand side formulas depend on $y : J$). Both "quantified" predicates $\exists f(p), \forall f(p) \in \mathsf{P}J$ are characterized by the adjunctions

$$\exists f(p) \leq q \quad \text{iff} \quad p \leq \mathsf{P}f(q)$$

and

$$q \leq \forall f(p) \quad \text{iff} \quad \mathsf{P}f(q) \leq p$$

https://doi.org/10.1017/S0960129520000079 Published online by Cambridge University Press

(for all $q \in \mathsf{P}J$), which express the logical equivalences

$$(\forall y : J)[(\exists x : I)(f(x) = y \wedge p(x)) \Rightarrow q(y)] \quad \Leftrightarrow \quad (\forall x : I)[p(x) \Rightarrow q(f(x))]$$

and
$$(\forall y : J)[q(y) \Rightarrow (\forall x : I)(f(x) = y \Rightarrow p(x))] \quad \Leftrightarrow \quad (\forall x : I)[q(f(x)) \Rightarrow p(x)]\,.$$

Note that unlike the substitution map $\mathsf{P}f : \mathsf{P}J \to \mathsf{P}I$, the two adjoints $\exists f, \forall f : \mathsf{P}I \to \mathsf{P}J$ are only monotonic maps (i.e., arrows in **Pos**); they are not morphisms of HAs in general. (Quantifiers do not commute will all connectives!) Nevertheless, left adjoints commute with joins (and $\bot$), whereas right adjoints commute with meets (and $\top$):

$$\begin{array}{rclrcl} \exists f(p_1 \vee p_2) & = & \exists f(p_1) \vee \exists f(p_2) & \qquad \exists f(\bot) & = & \bot \\ \forall f(p_1 \wedge p_2) & = & \forall f(p_1) \wedge \forall f(p_2) & \qquad \forall f(\top) & = & \top \end{array} \qquad \text{(for all } p_1, p_2 \in \mathsf{P}I\text{)}$$

Using left adjoints, we can also define the equality predicate

$$(=_I) \;:=\; \exists\delta_I(\top_I) \;\in\; \mathsf{P}(I \times I) \qquad \text{(for each } I \in \mathbf{Set}\text{)}$$

writing $\delta_I : I \to I \times I$ the duplication function and $\top_I$ the top element of $\mathsf{P}I$. From what precedes, it should be clear to the reader that this predicate represents the formula $(\exists x \in I)(\delta(x) = (x_1, x_2) \wedge \top)$ (depending on $x_1, x_2 : I$), that is equivalent to $x_1 = x_2$.

(4) The Beck–Chevalley condition expresses a property of commutation between substitution and quantifications. It is typically used with pullback squares of the form

$$\begin{array}{ccc} I \times K & \xrightarrow{\pi_{I,K}} & I \\ {\scriptstyle f \times \mathrm{id}_K}\downarrow & \lrcorner & \downarrow{\scriptstyle f} \\ J \times K & \xrightarrow[\pi_{J,K}]{} & J \end{array}$$

where the adjoints $\exists\pi_{I,K}, \forall\pi_{I,K} : \mathsf{P}(I \times K) \to \mathsf{P}I$ and $\exists\pi_{J,K}, \forall\pi_{J,K} : \mathsf{P}(J \times K) \to \mathsf{P}J$ represent "pure" quantifications over an abstract variable $z : K$ (in the contexts $x : I$ and $y : J$, respectively). In this case, the induced equalities

$$\exists\pi_{I,K} \circ \mathsf{P}(f \times \mathrm{id}_K) = \mathsf{P}f \circ \exists\pi_{J,K} \quad \text{and} \quad \forall\pi_{I,K} \circ \mathsf{P}(f \times \mathrm{id}_K) = \mathsf{P}f \circ \forall\pi_{J,K}$$

$$\begin{array}{ccc} \mathsf{P}(I \times K) & \xrightarrow{\exists\pi_{I,K}} & \mathsf{P}I \\ {\scriptstyle \mathsf{P}(f \times \mathrm{id}_K)}\uparrow & & \uparrow{\scriptstyle \mathsf{P}f} \\ \mathsf{P}(J \times K) & \xrightarrow[\exists\pi_{J,K}]{} & \mathsf{P}J \end{array} \qquad \begin{array}{ccc} \mathsf{P}(I \times K) & \xrightarrow{\forall\pi_{I,K}} & \mathsf{P}I \\ {\scriptstyle \mathsf{P}(f \times \mathrm{id}_K)}\uparrow & & \uparrow{\scriptstyle \mathsf{P}f} \\ \mathsf{P}(J \times K) & \xrightarrow[\forall\pi_{J,K}]{} & \mathsf{P}J \end{array}$$

express for each predicate $q \in \mathsf{P}(J \times K)$ the logical equivalences

$$(\forall x : I)[(\exists z : K)(q(y, z)\{y := f(x), z := z\}) \Leftrightarrow ((\exists z : K)q(y, z))\{y := f(x)\}]$$

and
$$(\forall x : I)[(\forall z : K)(q(y, z)\{y := f(x), z := z\}) \Leftrightarrow ((\forall z : K)q(y, z))\{y := f(x)\}]$$

describing the behavior of substitution with respect to quantifiers.

(5) The Beck–Chevalley condition requires that the diagrams

$$\begin{array}{ccc} \mathsf{P}I & \xrightarrow{\exists f_1} & \mathsf{P}I_1 \\ {\scriptstyle \mathsf{P}f_2}\uparrow & & \uparrow{\scriptstyle \mathsf{P}g_1} \\ \mathsf{P}I_2 & \xrightarrow[\exists g_2]{} & \mathsf{P}J \end{array} \qquad \text{and} \qquad \begin{array}{ccc} \mathsf{P}I & \xrightarrow{\forall f_1} & \mathsf{P}I_1 \\ {\scriptstyle \mathsf{P}f_2}\uparrow & & \uparrow{\scriptstyle \mathsf{P}g_1} \\ \mathsf{P}I_2 & \xrightarrow[\forall g_2]{} & \mathsf{P}J \end{array}$$

https://doi.org/10.1017/S0960129520000079 Published online by Cambridge University Press

commute for all pullback squares $\begin{array}{ccc} I & \xrightarrow{f_1} & I_1 \\ {\scriptstyle f_2}\downarrow & \lrcorner & \downarrow{\scriptstyle g_1} \\ I_2 & \xrightarrow[g_2]{} & J \end{array}$ in **Set**.

However, both commutation properties are equivalent up to the symmetry with respect to the diagonal (by exchanging the indices 1 and 2 in the above pullback square). By this, we mean that the $\exists$-diagram associated to the initial pullback square commutes if and only if the $\forall$-diagram associated to the symmetric pullback square (obtained by exchanging the indices 1 and 2) commutes, that is,

$$\exists f_1 \circ \mathsf{P} f_2 = \mathsf{P} g_1 \circ \exists g_2 \qquad \text{iff} \qquad \forall f_2 \circ \mathsf{P} f_1 = \mathsf{P} g_2 \circ \forall g_1 \,.$$

(This equivalence is easily derived from the adjunctions defining the monotonic maps $\exists f$ and $\forall f$.) So that in order to prove the Beck–Chevalley condition, we only need to check that all $\exists$-diagrams commute, or that all $\forall$-diagrams commute.

(6) Finally, the set *Prop* represents the *type of propositions*, whereas the generic predicate $\mathsf{Tr} \in \mathsf{P}\,\mathit{Prop}$ represents the formula asserting that a given proposition is true. Thanks to this predicate, we can turn any *functional proposition* into a predicate via the map

$$\begin{array}{rcl} \mathit{Prop}^I & \to & \mathsf{P}I \\ f & \mapsto & \mathsf{P}f(\mathsf{Tr}) \end{array} \qquad (I \in \mathbf{Set})$$

We require that this map is surjective for all sets I, thus ensuring that each predicate $p \in \mathsf{P}I$ is represented by (at least) a functional proposition $f \in \mathit{Prop}^I$.

Remark 4.9 (**Non-uniqueness of the generic predicate**). It is important to observe that in a **Set**-based tripos P, the generic predicate is never unique.

(1) Indeed, given a generic predicate $\mathsf{Tr} \in \mathsf{P}\,\mathit{Prop}$ and a surjection $h : \mathit{Prop}' \to \mathit{Prop}$, we can always construct another generic predicate $\mathsf{Tr}' \in \mathsf{P}\,\mathit{Prop}'$, letting $\mathsf{Tr}' = \mathsf{P}h(\mathsf{Tr})$.[8]

(2) More generally, if $\mathsf{Tr} \in \mathsf{P}\,\mathit{Prop}$ and $\mathsf{Tr}' \in \mathsf{P}\,\mathit{Prop}'$ are two generic predicates of the same tripos P, then there always exist two conversion maps $h : \mathit{Prop}' \to \mathit{Prop}$ and $h' : \mathit{Prop} \to \mathit{Prop}'$ such that $\mathsf{Tr}' = \mathsf{P}h(\mathsf{Tr})$ and $\mathsf{Tr} = \mathsf{P}h'(\mathsf{Tr})$. Intuitively, the sets *Prop* and *Prop′* represent distinct implementations of the type of propositions (they do not need to have the same cardinality), whereas the conversion functions $h : \mathit{Prop}' \to \mathit{Prop}$ and $h' : \mathit{Prop} \to \mathit{Prop}'$ implement the corresponding changes in representation.

Example 4.10 (**Forcing tripos**). Given a complete HA $(H, \leq)$, the functor $\mathsf{P} : \mathbf{Set}^{\mathrm{op}} \to \mathbf{HA}$ defined for all $I, J \in \mathbf{Set}$ and $f : I \to J$ by

$$\mathsf{P}I := H^I \qquad \text{and} \qquad \mathsf{P}f := (h \mapsto h \circ f) : H^J \to H^I$$

is a **Set**-based tripos, in which left and right adjoints $\exists f, \forall f : \mathsf{P}I \to \mathsf{P}J$ are given by

$$\exists f(p) := \Big(\bigvee_{i \in f^{-1}(j)} p_i\Big)_{j \in J} \qquad \text{and} \qquad \forall f(p) := \Big(\bigwedge_{i \in f^{-1}(j)} p_i\Big)_{j \in J}$$

(for all $f : I \to J$ and $p \in \mathsf{P}I = H^I$), and whose generic predicate $(\mathit{Prop}, \mathsf{Tr})$ is given by

$$\mathit{Prop} := H \qquad \text{and} \qquad \mathsf{Tr} := \mathrm{id}_H \in \mathsf{P}\mathit{Prop}\,.$$

Such a tripos is called a *Heyting tripos*, or a *forcing tripos*.

https://doi.org/10.1017/S0960129520000079 Published online by Cambridge University Press

4.4 Construction of the implicative tripos

Theorem 4.11 **(Implicative tripos).** *Let $\mathscr{A} = (\mathscr{A}, \preccurlyeq, \to, S)$ be an implicative algebra. For each set I, we write $\mathsf{P}I = \mathscr{A}^I/S[I]$. Then:*

(1) The correspondence $I \mapsto \mathsf{P}I$ induces a (contravariant) functor $\mathsf{P} : \mathbf{Set}^{\mathrm{op}} \to \mathbf{HA}$
(2) The functor $\mathsf{P} : \mathbf{Set}^{\mathrm{op}} \to \mathbf{HA}$ is a $\mathbf{Set}$-based tripos.

Proof. It is clear that for each set I, the poset $(\mathscr{A}^I/S[I], \leq_{S[I]})$ is an HA, namely, the HA induced by the implicative algebra $(\mathscr{A}^I, \preccurlyeq^I, \to^I, S[I])$.

Functoriality Let $I, J \in \mathbf{Set}$. Each function $f : I \to J$ induces a reindexing map $\mathscr{A}^f : \mathscr{A}^J \to \mathscr{A}^I$ defined by $\mathscr{A}^f(a) = a \circ f$ for all $a \in \mathscr{A}^J$. Now, let us consider two families $a, b \in \mathscr{A}^J$ such that $a \dashv\vdash_{S[J]} b$, that is, such that $\bigcurlywedge_{j\in J}(a_j \leftrightarrow b_j) \in S$. Since $\bigcurlywedge_{j\in J}(a_j \leftrightarrow b_j) \preccurlyeq \bigcurlywedge_{i\in I}(a_{f(i)} \leftrightarrow b_{f(j)})$, we deduce that $\bigcurlywedge_{i\in I}(a_{f(i)} \leftrightarrow b_{f(i)}) \in S$, so that $\mathscr{A}^f(a) \dashv\vdash_{S[I]} \mathscr{A}^f(b)$. Therefore, through the quotients $\mathsf{P}J = \mathscr{A}^J/S[J]$ and $\mathsf{P}I = \mathscr{A}^I/S[I]$, the reindexing map $\mathscr{A}^f : \mathscr{A}^J \to \mathscr{A}^I$ factors into a map $\mathsf{P}f : \mathsf{P}J \to \mathsf{P}I$. We now need to check that the map $\mathsf{P}f : \mathsf{P}J \to \mathsf{P}I$ is a morphism of HAs. For that, given predicates $p = [a]_{/S[J]} \in \mathsf{P}J$ and $q = [b]_{/S[J]} \in \mathsf{P}J$, we observe that

$$\begin{aligned}\mathsf{P}f(p \wedge q) &= \mathsf{P}f\big([a \times^J b]_{/S[J]}\big) = \mathsf{P}f\big([(a_j \times b_j)_{j\in J}]_{/S[J]}\big)\\ &= \big[(a_{f(i)} \times b_{f(i)})_{i\in I}\big]_{/S[I]} = \big[(a_{f(i)})_{i\in I} \times^I (b_{f(i)})_{i\in I}\big]_{/S[I]}\\ &= \big[(a_{f(i)})_{i\in I}\big]_{/S[I]} \wedge \big[(b_{f(i)})_{i\in I}\big]_{/S[I]} = \mathsf{P}f(p) \wedge \mathsf{P}f(q)\end{aligned}$$

(The case of the other connectives $\vee$, $\to$, $\bot$, and $\top$ is similar.) The contravariant functoriality of the correspondence $f \mapsto \mathsf{P}f$ is obvious from the definition.

Existence of right adjoints Let $f : I \to J$. For each family $a \in \mathscr{A}^I$, we let

$$\forall^0_f(a) = \Big(\bigcurlywedge_{f(i)=j} a_i\Big)_{j\in J} \qquad (\in \mathscr{A}^J)$$

We observe that for all $a, b \in \mathscr{A}^I$ and $s \in S$,

$$s \preccurlyeq \bigcurlywedge_{i\in I}(a_i \to b_i) \quad \text{implies} \quad s \preccurlyeq \bigcurlywedge_{j\in J}(\forall^0_f(a)_j \to \forall^0_f(b)_j)\,.$$

Therefore, $\quad a \vdash_{S[I]} b \quad$ implies $\quad \forall^0_f(a) \vdash_{S[J]} \forall^0_f(b)$,

and thus $\quad a \dashv\vdash_{S[I]} b \quad$ implies $\quad \forall^0_f(a) \dashv\vdash_{S[J]} \forall^0_f(b)$.

For each predicate $p = [a]_{/S[I]} \in \mathsf{P}I$, we can now let $\forall f(p) = \big[\forall^0_f(a)\big]_{/S[J]} \in \mathsf{P}J$. Given $p = [a]_{/S[I]} \in \mathsf{P}I$ and $q = [b]_{/S[J]} \in \mathsf{P}J$, it remains to check that

$$\begin{aligned}\mathsf{P}f(q) \leq p \text{ iff } &\bigcurlywedge_{i\in I}(b_{f(i)} \to a_i) \in S \quad \text{iff} \quad \bigcurlywedge_{j\in J}\bigcurlywedge_{f(i)=j}(b_j \to a_i) \in S\\ \text{iff } &\bigcurlywedge_{j\in J}\Big(b_j \to \bigcurlywedge_{f(i)=j} a_i\Big) \in S \quad \text{iff} \quad \bigcurlywedge_{j\in J}(b_j \to \forall^0_f(a)_j) \in S\\ \text{iff } &q \leq \forall f(p)\end{aligned}$$

Existence of left adjoints Let $f : I \to J$. For each family $a \in \mathscr{A}^I$, we let

$$\exists^0_f(a) \;=\; \Big(\bigexists_{f(i)=j} a_i\Big)_{j\in J} \;=\; \Big(\bigcurlywedge_{c\in\mathscr{A}}\Big(\bigcurlywedge_{f(i)=j}(a_i \to c) \;\to\; c\Big)\Big)_{j\in J} \qquad (\in \mathscr{A}^J)$$

We observe that for all $a, b \in \mathscr{A}^I$ and $s \in S$,

$$s \preccurlyeq \bigcurlywedge_{i\in I}(a_i \to b_i) \qquad \text{implies} \qquad s' \preccurlyeq \bigcurlywedge_{j\in J}(\exists^0_f(a)_j \to \exists^0_f(b)_j)\,,$$

where $s' := (\lambda xy\,.\,x\,(\lambda z\,.\,y\,(s\,z)))^{\mathscr{A}} \in S$.

Therefore, $\quad a \vdash_{S[I]} b \quad$ implies $\quad \exists^0_f(a) \vdash_{S[J]} \exists^0_f(b)\,,$

and thus $\quad a \dashv\vdash_{S[I]} b \quad$ implies $\quad \exists^0_f(a) \dashv\vdash_{S[J]} \exists^0_f(b)\,.$

For each predicate $p = [a]_{/S[I]} \in \mathsf{P}I$, we can now let $\exists f(p) = \big[\exists^0_f(a)\big]_{/S[J]} \in \mathsf{P}J$. Given $p = [a]_{/S[I]} \in \mathsf{P}I$ and $q = [b]_{/S[J]} \in \mathsf{P}J$, it remains to check that

$$\begin{aligned}
p \le \mathsf{P}f(q) \quad & \text{iff} \quad \bigcurlywedge_{i\in I}(a_i \to b_{f(i)}) \in S \quad \text{iff} \quad \bigcurlywedge_{j\in J}\bigcurlywedge_{f(i)=j}(a_i \to b_j) \in S \\
& \text{iff} \quad \bigcurlywedge_{j\in J}\Big(\Big(\bigexists_{f(i)=j} a_i\Big) \to b_j\Big) \in S \quad \text{iff} \quad \bigcurlywedge_{j\in J}(\exists^0_f(a)_j \to b_j) \in S \\
& \text{iff} \quad \exists f(p) \le q
\end{aligned}$$

(Here, the third "iff" follows from the soundness of the elimination rule of $\exists$.)

Beck–Chevalley condition Let us now check that the Beck–Chevalley condition holds for the functor $\mathsf{P} : \mathbf{Set}^{\mathrm{op}} \to \mathbf{HA}$. For that, we consider an arbitrary pullback square in the category **Set**

$$\begin{array}{ccc}
I & \xrightarrow{f_1} & I_1 \\
{\scriptstyle f_2}\downarrow & \lrcorner & \downarrow{\scriptstyle g_1} \\
I_2 & \xrightarrow[g_2]{} & J
\end{array}$$

and we want to show that the following two diagrams commute (in **Pos**):

$$\begin{array}{ccc}
\mathsf{P}I & \xrightarrow{\exists f_1} & \mathsf{P}I_1 \\
{\scriptstyle \mathsf{P}f_2}\uparrow & & \uparrow{\scriptstyle \mathsf{P}g_1} \\
\mathsf{P}I_2 & \xrightarrow[\exists g_2]{} & \mathsf{P}J
\end{array}
\qquad\qquad
\begin{array}{ccc}
\mathsf{P}I & \xrightarrow{\forall f_1} & \mathsf{P}I_1 \\
{\scriptstyle \mathsf{P}f_2}\uparrow & & \uparrow{\scriptstyle \mathsf{P}g_1} \\
\mathsf{P}I_2 & \xrightarrow[\forall g_2]{} & \mathsf{P}J
\end{array}$$

Since both commutation properties are equivalent up to the symmetry with respect to the diagonal (Remarks 4.8 (5)), we only need to prove the second commutation property. And since the correspondence $f \mapsto \forall f$ is functorial, we can assume without loss of generality that

- $I = \{(i_1, i_2) \in I_1 \times I_2 : g_1(i_1) = g_2(i_2)\}$
- $f_1(i_1, i_2) = i_1$, for all $(i_1, i_2) \in I$
- $f_2(i_1, i_2) = i_2$, for all $(i_1, i_2) \in I$

using the fact that each pullback diagram in **Set** is of this form, up to a bijection. For all $p = [a] = \big[(a_i)_{i\in I_2}\big] \in \mathsf{P}I_2$, we check that

$$\begin{aligned}(\forall f_1 \circ \mathsf{P}f_2)(p) &= \forall f_1\big(\big[(a_{f_2(i_1,i_2)})_{(i_1,i_2)\in I}\big]\big) = \forall f_1\big(\big[(a_{i_2})_{(i_1,i_2)\in I}\big]\big)\\ &= \Bigg[\Bigg(\bigcurlywedge_{\substack{(i_1,i_2)\in I\\ f_1(i_1,i_2)=i_1'}} a_{i_2}\Bigg)_{i_1'\in I_1}\Bigg] = \Bigg[\Bigg(\bigcurlywedge_{\substack{i_2\in I_2\\ g_2(i_2)=g_1(i_1)}} a_{i_2}\Bigg)_{i_1\in I_1}\Bigg]\\ &= \Big[\Big(\big(\forall^0_{g_2}(a)\big)_{g_1(i_1)}\Big)_{i_1\in I_1}\Big] = \mathsf{P}g_1\big(\big[\forall^0_{g_2}(a)\big]\big)\\ &= (\mathsf{P}g_1 \circ \forall g_2)(p)\end{aligned}$$

The generic predicate Let us now take $Prop := \mathscr{A}$ and $\mathsf{Tr} := [\mathrm{id}_{\mathscr{A}}]_{/S[\mathscr{A}]} \in \mathsf{P}\,Prop$. Given a set $I \in$ **Set** and a predicate $p = \big[(a_i)_{i\in I}\big]_{/S[I]} \in \mathsf{P}I$, we take $f := (a_i)_{i\in I} : I \to \mathscr{A}$ and check that

$$\mathsf{P}f(\mathsf{Tr}) = \mathsf{P}f\big(\big[(a)_{a\in\mathscr{A}}\big]_{/S[\mathscr{A}]}\big) = \big[(a_i)_{i\in I}\big]_{/S[I]} = p\,. \qquad \square$$

Example 4.12 (**Particular case of a complete HA**). In the particular case where the implicative algebra $(\mathscr{A}, \preccurlyeq, \to, S)$ is a complete HA (which means that $\to$ is Heyting's implication, whereas the separator is trivial: $S = \{\top\}$), we can observe that for each set I, the equivalence relation $\dashv\vdash_{S[I]}$ over $\mathscr{A}^I$ is discrete (each equivalence class has one element), so that we can drop the quotient:

$$\mathsf{P}I = \mathscr{A}^I/S[I] \sim \mathscr{A}^I\,.$$

Up to this technical detail, the implicative tripos associated to the implicative algebra $(\mathscr{A}, \preccurlyeq, \to, S)$ is thus the very same as the forcing tripos associated to the underlying complete HA $(\mathscr{A}, \preccurlyeq)$ (cf. Example 4.10).

4.5 Characterizing forcing triposes

Example 4.12 shows that forcing triposes are particular cases of implicative triposes. However, it turns out that many implicative algebras that are not complete HAs nevertheless induce a tripos that is isomorphic to a forcing tripos. The aim of this section is to characterize them, by proving the following:

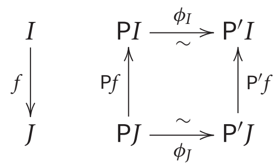
Theorem 4.13 (**Characterizing forcing triposes**). *Let* $\mathsf{P} : \mathbf{Set}^{\mathrm{op}} \to \mathbf{HA}$ *be the tripos induced by an implicative algebra* $(\mathscr{A}, \preccurlyeq, \to, S)$*. Then, the following are equivalent.*

(1) P *is isomorphic to a forcing tripos.*
(2) The separator $S \subseteq \mathscr{A}$ *is a principal filter of* $\mathscr{A}$*.*
(3) The separator $S \subseteq \mathscr{A}$ *is finitely generated and* $⫛^{\mathscr{A}} \in S$*.*

Before proving the theorem, let us recall that:

Definition 4.14. *Two* **Set***-based triposes* $\mathsf{P}, \mathsf{P}' : \mathbf{Set}^{\mathrm{op}} \to \mathbf{HA}$ *are* isomorphic *when there exists a natural isomorphism* $\phi : \mathsf{P} \Rightarrow \mathsf{P}'$*, that is, a family of isomorphisms* $\phi_I : \mathsf{P}I \xrightarrow{\sim} \mathsf{P}'I$ *(indexed by* $I \in$ **Set***) such that the following diagram commutes:*

https://doi.org/10.1017/S0960129520000079 Published online by Cambridge University Press



(*for all* $f : I \to J$)

(Note that here, the notion of isomorphism can be taken indifferently in the sense of **HA** *or* **Pos**, *since* $\phi_I : \mathsf{P}I \to \mathsf{P}'I$ *is an iso in* **HA** *if and only if it is an iso in* **Pos**.*)*

Remarks 4.15. The above definition does not take care of generic predicates, since any natural isomorphism $\phi : \mathsf{P} \Rightarrow \mathsf{P}'$ automatically maps each generic predicate $\mathsf{Tr} \in \mathsf{P}\,\mathit{Prop}$ (for the tripos P) into a generic predicate $\mathsf{Tr}' := \phi_{\mathit{Prop}}(\mathsf{Tr}) \in \mathsf{P}'\mathit{Prop}$ (for the tripos P').

4.5.1 The fundamental diagram

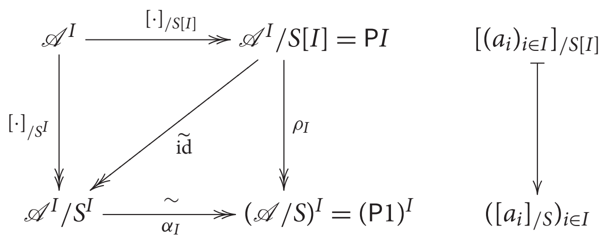
Given an implicative algebra $\mathscr{A} = (\mathscr{A}, \preccurlyeq, \to, S)$ and a set I, we have seen (Section 4.2) that the separator $S \subseteq \mathscr{A}$ induces two separators

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

in the power implicative structure $\mathscr{A}^I$, where

$$S[I] := \left\{(a_i)_{i\in I} \in \mathscr{A}^I \;:\; \exists s \in S,\ \forall i \in I,\ s \preccurlyeq a_i\right\} \qquad \text{(uniform power separator)}$$

We thus get the following (commutative) diagram



where

- $[\,\cdot\,]_{/S[I]} : \mathscr{A}^I \to A^I/S[I]$ $(= \mathsf{P}I)$ is the quotient map associated to $A^I/S[I]$;
- $[\,\cdot\,]_{/S^I} : \mathscr{A}^I \to A^I/S^I$ is the quotient map associated to A^I/S^I;
- $\tilde{\mathrm{id}} : \mathscr{A}^I/S[I] \to \mathscr{A}^I/S^I$ is the (surjective) map that factors the identity of $\mathscr{A}^I$ through the quotients $\mathscr{A}^I/S[I]$ and $\mathscr{A}^I/S^I$ (remember that $S[I] \subseteq S^I$);
- $\alpha_I = \langle\tilde{\pi}_i\rangle_{i\in I} : \mathscr{A}^I/S^I \to (\mathscr{A}/S)^I$ is the canonical isomorphism (Proposition 4.4) between the HAs $\mathscr{A}^I/S^I$ and $(\mathscr{A}/S)^I$ $(= (\mathsf{P}1)^I)$;
- $\rho_I : \mathscr{A}^I/S[I] \to (\mathscr{A}/S)^I$ is the (surjective) map that is defined by $\rho_I := \alpha_I \circ \tilde{\mathrm{id}}$, so that for all $(a_i)_{i\in I} \in \mathscr{A}^I$, we have

$$\rho_I\big([(a_i)_{i\in I}]_{/S[I]}\big) = \big([a_i]_{/S}\big)_{i\in I}.$$

Proposition 4.16. *The following are equivalent.*

1. *The map* $\rho_I : \mathsf{P}I \to (\mathsf{P}1)^I$ *is injective.*
2. *The map* $\rho_I : \mathsf{P}I \to (\mathsf{P}1)^I$ *is an isomorphism of HAs.*
3. *Both separators* $S[I]$ *and* S^I *coincide:* $S[I] = S^I$.
4. *The separator* $S \subseteq \mathscr{A}$ *is closed under all* I*-indexed meets.*

https://doi.org/10.1017/S0960129520000079 Published online by Cambridge University Press

Proof. (1) $\Leftrightarrow$ (2) Recall that a morphism of HAs is an isomorphism (in **HA**) if and only if the underlying map (in **Set**) is bijective. But since ρ_I is a surjective morphism of HAs, it is clear that ρ_I is an isomorphism (in **HA**) if and only the underlying map (in **Set**) is injective.

(2) $\Leftrightarrow$ (3) It is clear that ρ_I is an iso iff $\widetilde{\mathrm{id}}$ is an iso, that is, iff $S[I]=S^I$.

(3) $\Leftrightarrow$ (4) See Proposition 4.5, p. 493. □

We can now present:

Proof of Theorem 4.13. We have already proved that (2) $\Leftrightarrow$ (3) (Proposition 3.30, Section 3.7.4), so that it only remains to prove that (1) $\Leftrightarrow$ (2).

(2) $\Rightarrow$ (1) When $S\subseteq\mathscr{A}$ is a principal filter, the HA $H:=\mathsf{P}1=\mathscr{A}/S$ is complete (Proposition 3.30). Moreover, since S is closed under arbitrary meets, the arrow $\rho_I:\mathsf{P}I\to(\mathsf{P}1)^I$ is an isomorphism (Proposition 4.16) for all sets I. It is also clearly natural in I, so that the family $(\rho_I)_{I\in\mathbf{Set}}$ is an isomorphism between the implicative tripos P and the forcing tripos $I\mapsto H^I$ (where $H=\mathsf{P}1=\mathscr{A}/S$).

(1) $\Rightarrow$ (2) Let us now assume that there is a complete HA H together with a natural isomorphism $\phi_I:\mathsf{P}I\xrightarrow{\sim}H^I$ (in I). In particular, we have $\phi_1:\mathsf{P}1\xrightarrow{\sim}H^1=H$, so that $\mathscr{A}/S=\mathsf{P}1\sim H$ is a complete HA. Now, fix a set I and write $c_i:=\{0\mapsto i\}:1\to I$ for each element $i\in I$. Via the two (contravariant) functors $\mathsf{P},H^{(-)}:\mathbf{Set}^{\mathrm{op}}\to\mathbf{HA}$, we easily check that the arrow $c_i:1\to I$ is mapped to

$$\mathsf{P}c_i=\rho_i\ :\mathsf{P}I\to\mathsf{P}1$$

and

$$H^{c_i}=\pi'_i\ :H^I\to H$$

where ρ_i is the ith component of the surjection $\rho_I:\mathsf{P}I\twoheadrightarrow(\mathsf{P}1)^I$ and where π'_i is the ith projection from H^I to H. We then observe that the two diagrams

$$\begin{array}{ccc} \mathscr{A}/S & \xrightarrow[\sim]{\phi_1} & H \\ {\scriptstyle \mathsf{P}c_i=\rho_i}\uparrow & & \uparrow{\scriptstyle \pi'_i=H^{c_i}} \\ \mathscr{A}^I/S[I] & \xrightarrow[\phi_I]{\sim} & H^I \end{array} \qquad \begin{array}{ccc} (\mathscr{A}/S)^I & \xrightarrow[\sim]{\phi_1^I} & H^I \\ {\scriptstyle \rho_I=\langle\rho_i\rangle_{i\in I}}\uparrow & & \sim\uparrow{\scriptstyle \mathrm{id}_{H^I}=\langle\pi'_i\rangle_{i\in I}} \\ \mathscr{A}^I/S[I] & \xrightarrow[\phi_I]{\sim} & H^I \end{array}$$

are commutative. Indeed, the first commutation property comes from the naturality of ϕ, and the second commutation property follows from the first commutation property, by gluing the arrows ρ_i and π'_i for all indices $i\in I$. From the second commutation property, it is then clear that the arrow $\rho_I:\mathsf{P}I\to(\mathsf{P}1)^I$ is an isomorphism for all sets I, so that by Proposition 4.16, the separator $S\subseteq\mathscr{A}$ is closed under arbitrary meets, which precisely means that it is a principal filter of $\mathscr{A}$. □

Remarks 4.17. Intuitively, Theorem 4.13 expresses that forcing is the same as non-deterministic realizability (both in intuitionistic and classical logic).

4.6 The case of classical realizability

In Sections 2.2.5 and 3.2.3, we have seen that each AKS $\mathcal{K}=(\Lambda,\Pi,\ldots)$ can be turned into a classical implicative algebra $\mathscr{A}_{\mathcal{K}}=(\mathfrak{P}(\Pi),\ldots)$. By Theorem 4.11, the classical implicative algebra $\mathscr{A}_{\mathcal{K}}$ induces in turn a (classical) tripos, which we shall call the *classical realizability tripos* induced by the AKS $\mathcal{K}$.

https://doi.org/10.1017/S0960129520000079 Published online by Cambridge University Press

Remark 4.18. In Streicher (2013), Streicher shows how to construct a classical tripos (which he calls a *Krivine tripos*) from an AKS, using a very similar construction. Streicher's construction is further refined in Ferrer Santos et al. (2017), which already introduces some of the main ideas underlying implicative algebras. Technically, the main difference between Streicher's construction and ours is that Streicher works with a smaller algebra $\mathscr{A}'_{\mathcal{K}}$ of truth values, that only contains the sets of stacks that are closed under bi-orthogonal:

$$\mathscr{A}'_{\mathcal{K}} = \mathfrak{P}_{\perp\!\!\!\perp}(\Pi) = \{S \in \mathfrak{P}(\Pi) : S = S^{\perp\!\!\!\perp\perp\!\!\!\perp}\}.$$

Although Streicher's algebra $\mathscr{A}'_{\mathcal{K}}$ is not an implicative algebra (it is a *classical ordered combinatory algebra*, following the terminology of Ferrer Santos et al. 2017), it nevertheless gives rise to a classical tripos, using a construction that is very similar to ours. In Ferrer and Malherbe (2017), it is shown that Streicher's tripos is actually isomorphic to the implicative tripos that is constructed from the implicative algebra $\mathscr{A}_{\mathcal{K}}$.

The following theorem states that AKSs generate the very same class of triposes as classical implicative algebras, so that both structures (AKSs and classical implicative algebras) have actually the very same logical expressiveness:

Theorem 4.19 (**Universality of AKS**). *For each classical implicative algebra $\mathscr{A}$, there exists an AKS $\mathcal{K}$ that induces the same tripos, in the sense that the classical realizability tripos induced by $\mathcal{K}$ is isomorphic to the implicative tripos induced by $\mathscr{A}$.*

The proof of Theorem 4.19 is a consequence of the following lemma:

Lemma 4.20 (**Reduction of implicative algebras**). *Let $\mathscr{A} = (\mathscr{A}, \preccurlyeq_{\mathscr{A}}, \to_{\mathscr{A}}, S_{\mathscr{A}})$ and $\mathscr{B} = (\mathscr{B}, \preccurlyeq_{\mathscr{B}}, \to_{\mathscr{B}}, S_{\mathscr{B}})$ be two implicative algebras. If there exists a surjective map $\psi : \mathscr{B} \to \mathscr{A}$ (a "reduction from $\mathscr{B}$ onto $\mathscr{A}$") such that*

(1) $\psi\big(\bigcurlywedge_{i\in I} b_i\big) = \bigcurlywedge_{i\in I} \psi(b_i)$ *(for all $I \in$ **Set** and $b \in \mathscr{B}^I$)*
(2) $\psi(b \to_{\mathscr{B}} b') = \psi(b) \to_{\mathscr{A}} \psi(b')$ *(for all $b, b' \in \mathscr{B}$)*
(3) $b \in S_{\mathscr{B}}$ *iff* $\psi(b) \in S_{\mathscr{A}}$ *(for all $b \in \mathscr{B}$)*

then the corresponding triposes $\mathsf{P}_{\mathscr{A}}, \mathsf{P}_{\mathscr{B}} : \mathbf{Set}^{\mathrm{op}} \to \mathbf{HA}$ are isomorphic.

Proof. For each set I, we consider the map $\psi^I : \mathscr{B}^I \to \mathscr{A}^I$ defined by $\psi^I(b) = \psi \circ b$ for all $b \in \mathscr{B}^I$. Given two points $b, b' \in \mathscr{B}^I$, we observe that

$$\begin{aligned} b \vdash_{S_{\mathscr{B}}[I]} b' \quad &\text{iff} \quad \textstyle\bigcurlywedge_{i\in I} (b_i \to_{\mathscr{B}} b'_i) \in S_{\mathscr{B}} \\ &\text{iff} \quad \psi\big(\textstyle\bigcurlywedge_{i\in I} (b_i \to_{\mathscr{B}} b'_i)\big) \in S_{\mathscr{A}} \\ &\text{iff} \quad \textstyle\bigcurlywedge_{i\in I} (\psi(b_i) \to_{\mathscr{A}} \psi(b'_i)) \in S_{\mathscr{A}} \\ &\text{iff} \quad \psi^I(b) \vdash_{S_{\mathscr{A}}[I]} \psi^I(b') \end{aligned}$$

From this, we deduce that

(1) The map $\psi^I : \mathscr{B}^I \to \mathscr{A}^I$ is compatible with the preorders $\vdash_{S_{\mathscr{B}}[I]}$ (on $\mathscr{B}^I$) and $\vdash_{S_{\mathscr{A}}[I]}$ (on $\mathscr{A}^I$), and thus factors into a monotonic map $\tilde{\psi}_I : \mathsf{P}_{\mathscr{B}} I \to \mathsf{P}_{\mathscr{A}} I$ through the quotients $\mathsf{P}_{\mathscr{B}} I = \mathscr{B}^I / S_{\mathscr{B}}[I]$ and $\mathsf{P}_{\mathscr{A}} I = \mathscr{A}^I / S_{\mathscr{A}}[I]$.
(2) The monotonic map $\tilde{\psi}_I : \mathsf{P}_{\mathscr{B}} I \to \mathsf{P}_{\mathscr{A}} I$ is an embedding of partial orderings, in the sense that $p \leq p'$ iff $\tilde{\psi}_I(p) \leq \tilde{\psi}_I(p')$ for all $p, p' \in \mathsf{P}_{\mathscr{B}} I$.

https://doi.org/10.1017/S0960129520000079 Published online by Cambridge University Press

Moreover, since $\psi : \mathscr{B} \to \mathscr{A}$ is surjective, the maps $\psi^I : \mathscr{B}^I \to \mathscr{A}^I$ and $\tilde{\psi}_I : \mathsf{P}_{\mathscr{B}} I \to \mathsf{P}_{\mathscr{A}} I$ are surjective too, so that the latter is actually an isomorphism in **Pos**, and thus an isomorphism in **HA**. The naturality of $\tilde{\psi}_I : \mathsf{P}_{\mathscr{B}} I \to \mathsf{P}_{\mathscr{A}} I$ (in I) follows from the naturality of $\psi^I : \mathscr{A}^I \to \mathscr{B}^I$ (in I), which is obvious by construction. □

Proof of Theorem 4.19. Let $\mathscr{A} = (\mathscr{A}, \preccurlyeq, \to, S)$ be a classical implicative algebra. Following Ferrer Santos et al. (2017), we define $\mathcal{K} = (\Lambda, \Pi, @, \cdot, \mathtt{k}\_, \mathtt{K}, \mathtt{S}, \mathtt{cc}, \mathrm{PL}, \perp\!\!\!\perp)$ by

- $\Lambda = \Pi := \mathscr{A}$
- $a@b := ab, \quad a \cdot b := a \to b \quad \text{and} \quad \mathtt{k}_a := a \to \bot$ (for all $a, b \in \mathscr{A}$)
- $\mathtt{K} := \mathbf{K}^{\mathscr{A}}, \quad \mathtt{S} := \mathbf{S}^{\mathscr{A}} \quad \text{and} \quad \mathtt{cc} := \mathbf{cc}^{\mathscr{A}}$
- $\mathrm{PL} := S \quad \text{and} \quad \perp\!\!\!\perp := (\preccurlyeq_{\mathscr{A}}) = \{(a, b) \in \mathscr{A}^2 : a \preccurlyeq b\}$

It is a routine exercise to check that the above structure is an AKS. Note that in this AKS, the orthogonal $\beta^{\perp\!\!\!\perp} \subseteq \Lambda$ of a set of stacks $\beta \subseteq \Pi$ is characterized by

$$\beta^{\perp\!\!\!\perp} = \{a \in \mathscr{A} : \forall b \in \beta,\ a \preccurlyeq b\} = \downarrow\!\{\textstyle\bigcurlywedge \beta\}$$

From the results of Sections 2.2.5 and 3.2.3, the AKS $\mathcal{K}$ induces in turn a classical implicative algebra $\mathscr{B} = (\mathscr{B}, \preccurlyeq_{\mathscr{B}}, \to_{\mathscr{B}}, S_{\mathscr{B}})$ that is defined by

- $\mathscr{B} := \mathfrak{P}(\Pi) = \mathfrak{P}(\mathscr{A})$
- $\beta \preccurlyeq_{\mathscr{B}} \beta' :\Leftrightarrow \beta \supseteq \beta'$ (for all $\beta, \beta' \in \mathscr{B}$)
- $\beta \to_{\mathscr{B}} \beta' := \beta^{\perp\!\!\!\perp} \cdot \beta' = \{a \to a' : a \preccurlyeq \bigcurlywedge \beta,\ a' \in \beta'\}$ (for all $\beta, \beta' \in \mathscr{B}$)
- $S_{\mathscr{B}} := \{\beta \in \mathscr{B} : \beta^{\perp\!\!\!\perp} \cap \mathrm{PL} \neq \varnothing\} = \{\beta \in \mathfrak{P}(\mathscr{A}) : \bigcurlywedge \beta \in S_{\mathscr{A}}\}$

Let us now define $\psi : \mathscr{B} \to \mathscr{A}$ by $\psi(\beta) = \bigcurlywedge \beta$ for all $\beta \in \mathscr{B}$ $(= \mathfrak{P}(\mathscr{A}))$. We easily check that $\psi : \mathscr{B} \to \mathscr{A}$ is a reduction from the implicative algebra $\mathscr{B}$ onto the implicative algebra $\mathscr{A}$ (in the sense of Lemma 4.20), so that by Lemma 4.20, the triposes induced by $\mathscr{A}$ and $\mathscr{B}$ are isomorphic. □

4.7 The case of intuitionistic realizability

In Sections 2.2.3 and 2.7.1, we have seen that each PCA $P = (P, \cdot, \mathtt{k}, \mathtt{s})$ induces a quasi-implicative structure $\mathscr{A}_P = (\mathfrak{P}(P), \subseteq, \to)$ based on Kleene's implication. In intuitionistic realizability (Hyland et al., 1980; Pitts, 2002; van Oosten, 2008), this quasi-implicative structure $\mathscr{A}_P$ is the logical seed from which the corresponding realizability tripos is constructed. Indeed, recall that the intuitionistic realizability tripos $\mathsf{P} : \mathbf{Set}^{\mathrm{op}} \to \mathbf{HA}$ induced by a PCA $P = (P, \cdot, \mathtt{k}, \mathtt{s})$ is defined by

$$\mathsf{P}I := \mathfrak{P}(P)^I / \dashv\vdash_I \qquad \text{(for all } I \in \mathbf{Set})$$

where $\dashv\vdash_I$ is the equivalence relation associated to the preorder of entailment $\vdash_I$ on the set $\mathfrak{P}(P)^I$ $(= \mathscr{A}_P^I)$, which is defined by

$$a \vdash_I a' \quad \text{iff} \quad \bigcap_{i \in I} (a_i \to a'_i) \neq \varnothing \qquad \text{(for all } I \in \mathbf{Set} \text{ and } a, a' \in \mathfrak{P}(P)^I)$$

In the particular case where the PCA $P = (P, \cdot, \mathtt{k}, \mathtt{s})$ is a (total) CA, the induced quasi-implicative structure $\mathscr{A}_P$ turns out to be a full implicative structure (Fact 2.5 (2), p. 463), and it should be clear to the reader that the above construction coincides with the construction of the implicative tripos induced by the implicative algebra $(\mathscr{A}_P, \mathfrak{P}(P) \setminus \{\varnothing\})$, which is obtained by endowing the implicative structure $\mathscr{A}_P$ with the separator formed by all nonempty truth values. In other words:

https://doi.org/10.1017/S0960129520000079 Published online by Cambridge University Press

Proposition 4.21. *For each (total) CA $P = (P, \cdot, k, s)$, the corresponding intuitionistic realizability tripos is an implicative tripos, namely, the implicative tripos induced by the implicative algebra $(\mathscr{A}_P, \mathfrak{P}(P) \setminus \{\varnothing\})$ induced by P.*

However, there are many interesting intuitionistic realizability triposes (for instance, Hyland's effective tripos) that are induced by PCAs whose application is not total. To see how such realizability triposes fit in our picture, it is now time to make a detour toward the notion of *quasi*-implicative algebra and the corresponding tripos construction.

4.7.1 Quasi-implicative algebras

Quasi-implicative structures (Remark 2.2 (2)) differ from (full) implicative structures in that the commutation property

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

only holds for the nonempty subsets $B \subseteq \mathscr{A}$, so that in general we have $(\top \to \top) \neq \top$.

In practice, quasi-implicative structures are manipulated essentially the same way as (full) implicative structures, the main difference being that, in the absence of the equation $(\top \to \top) = \top$, the operation of application $(a, b) \mapsto ab$ (Definition 2.11), and the interpretation $t \mapsto t^{\mathscr{A}}$ of pure λ-terms (Section 2.4) are now *partial* functions. Formally,

Definition 4.22 (**Interpretation of λ-terms in a quasi-implicative structure**). *Let $(\mathscr{A}, \preccurlyeq, \to)$ be a quasi-implicative structure.*

(1) Given $a, b \in \mathscr{A}$, we let $U_{a,b} = \{c \in \mathscr{A} : a \preccurlyeq (b \to c)\}$. When $U_{a,b} \neq \varnothing$, application is defined as $ab := \bigcurlywedge U_{a,b}$; otherwise, the notation ab is undefined.

(2) Given a partial *function $f : \mathscr{A} \rightharpoonup \mathscr{A}$, the abstraction $\boldsymbol{\lambda} f$ is (always) defined by*

$$\boldsymbol{\lambda} f := \bigcurlywedge_{a \in dom(f)} (a \to f(a)) \,.$$

(3) The partial *function $t \mapsto t^{\mathscr{A}}$ (from the set of closed λ-terms with parameters in $\mathscr{A}$ to $\mathscr{A}$) is defined by the equations*

$$a^{\mathscr{A}} := a, \qquad (tu)^{\mathscr{A}} := t^{\mathscr{A}} u^{\mathscr{A}}, \qquad (\lambda x \,.\, t)^{\mathscr{A}} := \boldsymbol{\lambda}(a \mapsto (t\{x := a\})^{\mathscr{A}})$$

whenever the right-hand side is well defined.

(4) The above interpretation naturally extends to λ-terms with $\mathfrak{cc}$, letting $\mathfrak{cc}^{\mathscr{A}} := \bigcurlywedge_{a,b \in \mathscr{A}} (((a \to b) \to a) \to a)$ (as before).

(The reader is invited to check that when the equation $(\top \to \top) = \top$ holds, the above functions are total and coincide with the ones defined in Section 2.4.)

In the broader context of quasi-implicative structures,

— All the semantic typing rules of Proposition 2.23, p. 469 remain valid (except the $\top$-introduction rule), provided we adapt the definition of semantic typing to partiality, by requiring that well-typed terms have a well-defined interpretation:

$$\Gamma \vdash t : a \quad :\equiv \quad FV(t) \subseteq \mathrm{dom}(\Gamma), \;\; (t[\Gamma])^{\mathscr{A}} \text{ well defined and } (t[\Gamma])^{\mathscr{A}} \preccurlyeq a \,.$$

— The identities of Proposition 2.24, p. 470 (about combinators $\mathbf{K}^{\mathscr{A}}$, $\mathbf{S}^{\mathscr{A}}$, etc.) remain valid.

— Separators are defined the same way as for implicative structures (Definition 3.1, p. 477).

https://doi.org/10.1017/S0960129520000079 Published online by Cambridge University Press

— A *quasi-implicative algebra* is a quasi-implicative structure $(\mathscr{A}, \preccurlyeq, \to)$ equipped with a separator $S \subseteq \mathscr{A}$. As before, each quasi-implicative algebra $(\mathscr{A}, \preccurlyeq, \to, S)$ induces an HA written $\mathscr{A}/S$, that is defined as the poset reflection of the preordered set $(\mathscr{A}, \vdash_S)$, where $\vdash_S$ is defined by $a \vdash_S b \ :\equiv\ (a \to b) \in S$ for all $a, b \in \mathscr{A}$.

Given a quasi-implicative algebra $(\mathscr{A}, \preccurlyeq, \to, S)$, we more generally associate to each set I the poset $\mathsf{P}I := \mathscr{A}^I/S[I]$, where $S[I]$ is the corresponding uniform power separator (same definition as before). It is then a routine exercise to check that:

Proposition 4.23 (**Quasi-implicative tripos**).

(1) The correspondence $I \mapsto \mathsf{P}I$ *induces a (contravariant) functor* $\mathsf{P} : \mathbf{Set}^{\mathrm{op}} \to \mathbf{HA}$.
(2) The functor $\mathsf{P} : \mathbf{Set}^{\mathrm{op}} \to \mathbf{HA}$ *is a* **Set***-based tripos.*

From what precedes, it is now clear that:

Proposition 4.24. *Given a PCA* $P = (P, \cdot, \Bbbk, \mathtt{s})$*:*

(1) The quadruple $(\mathfrak{P}(P), \subseteq, \to, \mathfrak{P}(P) \setminus \{\varnothing\})$ *is a quasi-implicative algebra.*
(2) The tripos induced by the quasi-implicative algebra $(\mathfrak{P}(P), \subseteq, \to, \mathfrak{P}(P) \setminus \{\varnothing\})$ *is the intuitionistic realizability tripos induced by the PCA* $P = (P, \cdot, \Bbbk, \mathtt{s})$.

At this point, the reader might wonder why we focused our study on the notion of implicative algebra rather than on the more general notion of quasi-implicative algebra. The reason is that from the point of view of logic, quasi-implicative algebras bring no expressiveness with respect to implicative algebras, due to the existence of a simple completion mechanism that turns any quasi-implicative algebra into a full implicative algebra, without changing the underlying tripos.

4.7.2 Completion of a quasi-implicative algebra

Given a quasi-implicative structure $\mathscr{A} = (\mathscr{A}, \preccurlyeq_{\mathscr{A}}, \to_{\mathscr{A}})$, we consider the triple $\mathscr{B} = (\mathscr{B}, \preccurlyeq_{\mathscr{B}}, \to_{\mathscr{B}})$ that is defined by

- $\mathscr{B} := \mathscr{A} \cup \{\top_{\mathscr{B}}\}$, where $\top_{\mathscr{B}}$ is a new element;
- $b \preccurlyeq_{\mathscr{B}} b'$ iff $b, b' \in \mathscr{A}$ and $b \preccurlyeq_{\mathscr{A}} b'$, or $b' = \top_{\mathscr{B}}$ (for all $b, b' \in \mathscr{B}$)
- $b \to_{\mathscr{B}} b' := \begin{cases} b \to_{\mathscr{A}} b' & \text{if } b, b' \in \mathscr{A} \\ \top_{\mathscr{A}} \to b' & \text{if } b = \top_{\mathscr{B}} \text{ and } b' \in \mathscr{A} \\ \top_{\mathscr{B}} & \text{if } b' = \top_{\mathscr{B}} \end{cases}$ (for all $b, b' \in \mathscr{B}$)

Fact 4.25. The triple $\mathscr{B} = (\mathscr{B}, \preccurlyeq_{\mathscr{B}}, \to_{\mathscr{B}})$ is a full implicative structure.

In what follows, we shall say that the implicative structure $\mathscr{B}$ is the *completion* of the quasi-implicative structure $\mathscr{A}$[9]. Intuitively, this completion mechanism simply consists to add to the source quasi-implicative structure $\mathscr{A}$ a new top element $\top_{\mathscr{B}}$ (or, from the point of view of definitional ordering: a new bottom element) that fulfills the equation $(\top_{\mathscr{B}} \to \top_{\mathscr{B}}) = \top_{\mathscr{B}}$ by construction.

Writing $\phi : \mathscr{A} \to \mathscr{B}$ the inclusion of $\mathscr{A}$ into $\mathscr{B}$, we easily check that:

https://doi.org/10.1017/S0960129520000079 Published online by Cambridge University Press

Lemma 4.26.

(1) $a \preccurlyeq_{\mathscr{A}} a'$ *iff* $\phi(a) \preccurlyeq_{\mathscr{B}} \phi(a')$ *(for all $a, a' \in \mathscr{A}$)*

(2) $\phi\Big(\bigwedge_{i\in I} a_i\Big) = \bigwedge_{i\in I} \phi(a_i)$ *(for all $I \neq \varnothing$ and $a \in \mathscr{A}^I$)*

(3) $\phi(a \to_{\mathscr{A}} a') = \phi(a) \to_{\mathscr{B}} \phi(a')$ *(for all $a, a' \in \mathscr{A}$)*

(4) $\phi(\mathbf{K}^{\mathscr{A}}) = \mathbf{K}^{\mathscr{B}}$ *and* $\phi(\mathbf{S}^{\mathscr{A}}) = \mathbf{S}^{\mathscr{B}}$

Proof. Items (1), (2), and (3) are obvious from the definition of $\preccurlyeq_{\mathscr{B}}$ and $\to_{\mathscr{B}}$. (Note that (2) only holds when $I \neq \varnothing$.) To prove (4), we observe that

$$\begin{aligned}
\mathbf{K}^{\mathscr{B}} &= \bigwedge_{a,b\in\mathscr{B}} (a \to_{\mathscr{B}} b \to_{\mathscr{B}} a) \\
&= \bigwedge_{a,b\in\mathscr{A}} (a \to_{\mathscr{B}} b \to_{\mathscr{B}} a) \;\;\curlywedge\;\; \bigwedge_{a\in\mathscr{A}} (a \to_{\mathscr{B}} \top_{\mathscr{B}} \to_{\mathscr{B}} a) \;\;\curlywedge \\
&\quad \bigwedge_{b\in\mathscr{A}} (\top_{\mathscr{B}} \to_{\mathscr{B}} b \to_{\mathscr{B}} \top_{\mathscr{B}}) \;\;\curlywedge\;\; (\top_{\mathscr{B}} \to_{\mathscr{B}} \top_{\mathscr{B}} \to_{\mathscr{B}} \top_{\mathscr{B}}) \\
&= \bigwedge_{a,b\in\mathscr{A}} (a \to_{\mathscr{A}} b \to_{\mathscr{A}} a) \;\;\curlywedge\;\; \bigwedge_{a\in\mathscr{A}} (a \to_{\mathscr{A}} \top_{\mathscr{A}} \to_{\mathscr{A}} a) \;\;\curlywedge\;\; \top_{\mathscr{B}} \;\;\curlywedge\;\; \top_{\mathscr{B}} \\
&= \bigwedge_{a,b\in\mathscr{A}} (a \to_{\mathscr{A}} b \to_{\mathscr{A}} a) \;=\; \phi(\mathbf{K}^{\mathscr{A}})
\end{aligned}$$

The equality $\phi(\mathbf{S}^{\mathscr{A}}) = \mathbf{S}^{\mathscr{B}}$ is proved similarly. □

From the above lemma, we immediately deduce that:

Proposition 4.27. *If $S_{\mathscr{A}}$ is a separator of the quasi-implicative structure $\mathscr{A}$, then the set $S_{\mathscr{B}} := S_{\mathscr{A}} \cup \{\top_{\mathscr{B}}\}$ is a separator of the implicative structure $\mathscr{B}$.*

Now, given a quasi-implicative algebra $\mathscr{A} = (\mathscr{A}, \preccurlyeq_{\mathscr{A}}, \to_{\mathscr{A}}, S_{\mathscr{A}})$, we can define its completion as the full implicative algebra $\mathscr{B} = (\mathscr{B}, \preccurlyeq_{\mathscr{B}}, \to_{\mathscr{B}}, S_{\mathscr{B}})$, where

- $(\mathscr{B}, \preccurlyeq_{\mathscr{B}}, \to_{\mathscr{B}})$ is the completion of the quasi-implicative structure $(\mathscr{A}, \preccurlyeq_{\mathscr{A}}, \to_{\mathscr{A}})$;
- $S_{\mathscr{B}} := S_{\mathscr{A}} \cup \{\top_{\mathscr{B}}\}$ is the extension of the separator $S_{\mathscr{A}} \subseteq \mathscr{A}$ into $\mathscr{B}$ (Proposition 4.27).

Writing $\mathsf{P}_{\mathscr{A}} : \mathbf{Set}^{\mathrm{op}} \to \mathbf{HA}$ and $\mathsf{P}_{\mathscr{B}} : \mathbf{Set}^{\mathrm{op}} \to \mathbf{HA}$ the triposes induced by $\mathscr{A}$ and $\mathscr{B}$, respectively, it now remains to check that:

Theorem 4.28. *The triposes $\mathsf{P}_{\mathscr{A}}, \mathsf{P}_{\mathscr{B}} : \mathbf{Set}^{\mathrm{op}} \to \mathbf{HA}$ are isomorphic.*

Proof. For each set I, we observe that the inclusion map $\phi : \mathscr{A} \to \mathscr{B}$ induces an inclusion map $\phi^I : \mathscr{A}^I \to \mathscr{B}^I$ (given by $\phi^I(a) = \phi \circ a$ for all $a \in \mathscr{A}^I$). Given $a, a' \in \mathscr{A}^I$, let us now check that

$$a \vdash_{S_{\mathscr{A}}[I]} a' \quad \text{iff} \quad \phi^I(a) \vdash_{S_{\mathscr{B}}[I]} \phi^I(a')\,.$$

https://doi.org/10.1017/S0960129520000079 Published online by Cambridge University Press

Indeed, the equivalence is clear when $I=\varnothing$, since $a=a'$ (for $\mathscr{A}^I$ is a singleton). And in the case where $I\neq\varnothing$, we have

$$\begin{aligned}
a\vdash_{S_{\mathscr{A}}[I]}a' \ \text{ iff }\ & \bigcurlywedge_{i\in I}(a_i\to_{\mathscr{A}}a'_i)\in S_{\mathscr{A}} \\
\text{iff }\ & \phi\Big(\bigcurlywedge_{i\in I}(a_i\to_{\mathscr{A}}a'_i)\Big)\in S_{\mathscr{B}} && (\text{since } S_{\mathscr{A}}=S_{\mathscr{B}}\cap\mathscr{A}=\phi^{-1}(S_{\mathscr{B}})) \\
\text{iff }\ & \bigcurlywedge_{i\in I}(\phi(a_i)\to_{\mathscr{B}}\phi(a'_i))\in S_{\mathscr{B}} && (\text{by Lemma 4.26 (2), (3)}) \\
\text{iff }\ & \phi^I(a)\vdash_{S_{\mathscr{B}}[I]}\phi^I(a')
\end{aligned}$$

From the above equivalence, it is clear that

(1) The map $\phi^I:\mathscr{A}^I\to\mathscr{B}^I$ is compatible with the preorders $\vdash_{S_{\mathscr{A}}[I]}$ (on $\mathscr{A}^I$) and $\vdash_{S_{\mathscr{B}}[I]}$ (on $\mathscr{B}^I$), and thus factors into a monotonic map $\tilde{\phi}_I:\mathsf{P}_{\mathscr{A}}I\to\mathsf{P}_{\mathscr{B}}I$ through the quotients $\mathsf{P}_{\mathscr{A}}I:=\mathscr{A}^I/S_{\mathscr{A}}[I]$ and $\mathsf{P}_{\mathscr{B}}I:=\mathscr{B}^I/S_{\mathscr{B}}[I]$.
(2) The monotonic map $\tilde{\phi}_I:\mathsf{P}_{\mathscr{A}}I\to\mathsf{P}_{\mathscr{B}}I$ is an embedding of partial orderings, in the sense that $p\leq p'$ iff $\tilde{\phi}_I(p)\leq\tilde{\phi}_I(p')$ for all $p,p'\in\mathsf{P}_{\mathscr{A}}I$.
(3) The embedding $\tilde{\phi}_I:\mathsf{P}_{\mathscr{A}}I\to\mathsf{P}_{\mathscr{B}}I$ is natural in $I\in\mathbf{Set}$.

To conclude that the embedding $\tilde{\phi}_I:\mathsf{P}_{\mathscr{A}}I\to\mathsf{P}_{\mathscr{B}}I$ is an isomorphism in **Pos** – and thus an isomorphism in **HA** –, it only remains to prove that it is surjective. For that, we consider the map $\psi:\mathscr{B}\to\mathscr{B}$ that is defined by

$$\psi(b)\ :=\ \bigcurlywedge_{c\in\mathscr{B}}((b\to c)\to c)\qquad(\text{for all } b\in\mathscr{B})$$

as well as the family of maps $\psi^I:\mathscr{B}^I\to\mathscr{B}^I$ ($I\in\mathbf{Set}$) defined by $\psi^I(b):=\psi\circ b$ for all $I\in\mathbf{Set}$ and $b\in\mathscr{B}^I$. Now, given $I\in\mathbf{Set}$ and $b\in\mathscr{B}^I$, we observe that

(1) $\psi^I(b)\dashv\vdash_{S_{\mathscr{B}}[I]}b$, since

$$(\lambda xz\,.\,zx)^{\mathscr{B}}\ \preccurlyeq\ \bigcurlywedge_{i\in I}(b_i\to\psi(b_i))\quad\text{and}\quad(\lambda y\,.\,y\,\mathbf{I})^{\mathscr{B}}\ \preccurlyeq\ \bigcurlywedge_{i\in I}(\psi(b_i)\to b_i)\,.$$

(2) $\psi^I(b)\in\mathscr{A}^I$, since for all $i\in I$, we have

$$\psi^I(b)_i\ =\ \psi(b_i)\ \preccurlyeq\ (b\to_{\mathscr{B}}\bot)\to_{\mathscr{B}}\bot\ \preccurlyeq\ \bot\to_{\mathscr{B}}\bot\ =\ \bot\to_{\mathscr{A}}\bot\ \preccurlyeq\ \top_{\mathscr{A}}\,.$$

Therefore, $[b]_{/S_{\mathscr{B}}[I]}=[\psi^I(b)]_{/S_{\mathscr{B}}[I]}=\tilde{\phi}_I\big([\psi^I(b)]_{/S_{\mathscr{A}}[I]}\big)$. Hence, $\tilde{\phi}_I$ is surjective. □

From the above discussion, we can now conclude that

Theorem 4.29. *For each PCA $P=(P,\cdot,\Bbbk,\mathbb{s})$, the intuitionistic realizability tripos induced by P is isomorphic to an implicative tripos, namely, to the implicative tripos induced by the completion of the quasi-implicative algebra $(\mathfrak{P}(P),\subseteq,\to,\mathfrak{P}(P)\setminus\{\varnothing\})$.*

Notes

1 Here, we put aside the case of numeric (or arithmetic) quantifications that can always be decomposed as a uniform quantification followed by a relativization: $(\forall x\in\mathbb{N})\phi(x)\equiv\forall x\,(x\in\mathbb{N}\Rightarrow\phi(x))$ and $(\exists x\in\mathbb{N})\phi(x)\equiv\exists x\,(x\in\mathbb{N}\wedge\phi(x))$.
2 The distinction between uniform constructions (e.g., intersection and union types) and non-uniform constructions (Cartesian product and direct sum) has always been overlooked in model theory, although it is at the core of the phenomenon

https://doi.org/10.1017/S0960129520000079 Published online by Cambridge University Press

of incompleteness in logic. Indeed, Gödel's undecidable sentence is a Π^0_1-formula $G \equiv \forall x\, \phi(x)$ that is built from a particular Δ^0_0-predicate $\phi(x)$ that has no generic proof, although each closed instance $\phi(n)$ ($n \in \mathbb{N}$) has.
3 Note that this correspondence automatically identifies realizers that have the same principal type. But since such realizers are clearly interchangeable in the "logic" of $\mathscr{A}$, this identification is harmless.
4 As a consequence, the constructions presented in Streicher (2013), Ferrer Santos et al. (2017) only fulfill half of the adjunction of Proposition 2.12 (5), the missing implication being recovered only up to a step of η-expansion, by inserting the combinator $\mathsf{E} = \lambda xy\,.\,xy$ appropriately (see Ferrer Santos et al. 2017; Streicher 2013 for the details).
5 This is why sets of stacks are sometimes called *falsity values*, as in Miquel (2011, 2011).
6 In classical realizability, it can be shown (Guillermo and Miquel, 2015) that the universal realizers of the second-order formula $\forall\alpha\,\forall\beta\,(\alpha \to \beta \to \alpha \cap \beta)$ (where $\alpha \cap \beta$ denotes the intersection of α and β) are precisely the closed terms t with the non-deterministic computational rules $t \star u \cdot v \cdot \pi \succ u \star \pi$ and $t \star u \cdot v \cdot \pi \succ v \star \pi$ for all closed terms u, v and for all stacks π. Recall that Krivine's abstract machine (Krivine, 2009) can be extended with extra instructions at will (for instance, an instruction $\pitchfork$ with the aforementioned non-deterministic behavior), so that such realizers may potentially exist.
7 Here, we recognize Dedekind's construction of natural numbers, as the elements of a fixed Dedekind-infinite set that are reached by the induction principle (seen as a local property).
8 To prove that $\mathsf{Tr}' \in \mathsf{P}\,Prop'$ is another generic predicate of the tripos P, we actually need to pick a right inverse of $h : Prop' \to Prop$, which exists by the Axiom of Choice (AC). Without (AC), the same argument works by replacing "surjective" with "having a right inverse."
9 Here, the terminology of completion is a bit abusive, since $\mathscr{B}$ always extends $\mathscr{A}$ with one point, even when $\mathscr{A}$ is already a full implicative algebra.

References

Barendregt, H. (1984). *The Lambda Calculus: Its Syntax and Semantics*, Studies in Logic and The Foundations of Mathematics, vol. 103, North-Holland, Elsevier.

Barendregt, H., Coppo, M. and Dezani-Ciancaglini, M. (1983). A filter lambda model and the completeness of type assignment. *Journal of Symbolic Logic* **48** (4) 931–940.

Cohen, P. J. (1963). The independence of the continuum hypothesis. *Proceedings of the National Academy of Sciences of the United States of America* **50** (6) 1143–1148.

Cohen, P. J. (1964). The independence of the continuum hypothesis II. *Proceedings of the National Academy of Sciences of the United States of America* **51** (1) 105–110.

Coppo, M., Dezani-Ciancaglini, M. and Venneri, B. (1980). Principal type schemes and lambda-calculus semantics. In: Hindley, R. and Seldin, G. (eds.) *To H. B. Curry. Essays on Combinatory Logic, Lambda-Calculus and Formalism*, London, Academic Press, 480–490.

Ferrer, W. and Malherbe, O. (2017). The category of implicative algebras and realizability. ArXiv e-prints.

Ferrer Santos, W., Frey, J., Guillermo, M., Malherbe, O. and Miquel, A. (2017). Ordered combinatory algebras and realizability. *Mathematical Structures in Computer Science* **27** (3) 428–458.

Friedman, H. (1973). Some applications of Kleene's methods for intuitionistic systems. In: Mathias, A. R. D. and Rogers, H. (eds.), *Cambridge Summer School in Mathematical Logic*, Lecture Notes in Mathematics, vol. 337, Berlin-Heidelberg-New York, Springer-Verlag, 113–170.

Girard, J. (1987). Linear logic. *Theoretical Computer Science* **50**, 1–102.

Girard, J.-Y. (1972). *Interprétation fonctionnelle et élimination des coupures de l'arithmétique d'ordre supérieur*. Doctorat d'État, Université Paris VII.

Girard, J.-Y., Lafont, Y. and Taylor, P. (1989). *Proofs and Types*. Cambridge, Cambridge University Press.

Griffin, T. (1990). A formulae-as-types notion of control. In: *Principles of Programming Languages (POPL'90)*, 47–58.

Guillermo, M. and Miquel, A. (2015). Specifying Peirce's law. *Mathematical Structures in Computer Science* **26** (7) 1269–1303.

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

Jech, T. (2002). *Set Theory, Third Millennium Edition (Revised and Expanded)*. Berlin-Heidelberg-New York, Springer.

Kleene, S. C. (1945). On the interpretation of intuitionistic number theory. *Journal of Symbolic Logic* **10** 109–124.

Koppelberg, S. (1989). *Handbook of Boolean Algebras*, vol. 1, North-Holland.

Krivine, J. L. (1993). *Lambda-Calculus, Types and Models*. Ellis Horwood. Out of print, now available at the author's web page at: https://www.irif.univ-paris-diderot.fr/~krivine/articles/Lambda.pdf.

Krivine, J.-L. (2001). Typed lambda-calculus in classical Zermelo-Fraenkel set theory. *Archive for Mathematical Logic* **40** (3) 189–205.

Krivine, J.-L. (2003). Dependent choice, 'quote' and the clock. *Theoretical Computer Science* **308** (1–3) 259–276.

Krivine, J.-L. (2009). Realizability in classical logic. In: *Interactive Models of Computation and Program Behaviour*, Panoramas et synthèses, vol. 27, Société Mathématique de France, 197–229.

Krivine, J.-L. (2011). Realizability algebras: a program to well order R. *Logical Methods in Computer Science* **7** 1–47.

https://doi.org/10.1017/S0960129520000079 Published online by Cambridge University Press

Krivine, J.-L. (2012). Realizability algebras II: new models of ZF + DC. *Logical Methods for Computer Science* **8** (1:10) 1–28.
Leivant, D. (1983). Polymorphic type inference. In: *Proceedings of the 10th ACM Symposium on Principles of Programming Languages*, 88–98.
McCarty, D. (1984). *Realizability and Recursive Mathematics*. Phd thesis, Oxford University.
Miquel, A. (2000). A model for impredicative type systems, universes, intersection types and subtyping. In: *LICS*, 18–29.
Miquel, A. (2011). Existential witness extraction in classical realizability and via a negative translation. *Logical Methods for Computer Science* **7** (2) 1–47.
Miquel, A. (2011). Forcing as a program transformation. In: *LICS*, IEEE Computer Society, 197–206.
Myhill, J. (1973). Some properties of intuitionistic Zermelo-Fraenkel set theory. *Lecture Notes in Mathematics* **337** 206–231.
Parigot, M. (1997). Proofs of strong normalisation for second order classical natural deduction. *Journal of Symbolic Logic* **62** (4) 1461–1479.
Pitts, A. M. (1981). *The Theory of Triposes*. Phd thesis, University of Cambridge.
Pitts, A. M. (2002). Tripos theory in retrospect. *Mathematical Structures in Computer Science* **12** (3) 265–279.
Ronchi della Rocca, S. and Venneri, B. (1984). Principal type schemes for an extended type theory. *Theoretical Computer Science* **28** 151–169.
Ruyer, F. (2006). *Preuves, types et sous-types*. Thèse de doctorat, Université Savoie Mont Blanc.
Streicher, T. (2013). Krivine's classical realisability from a categorical perspective. *Mathematical Structures in Computer Science* **23** (6) 1234–1256.
Tait, W. (1967). Intensional interpretation of functionals of finite type I. *Journal of Symbolic Logic* **32** (2), 198–212.
van Bakel, S., Liquori, L., Ronchi della Rocca, S. and Urzyczyn, P. (1994). Comparing cubes. In: Nerode, A. and Matiyasevich, Y. V. (eds.), *Proceedings of LFCS'94. Third International Symposium on Logical Foundations of Computer Science*, Lecture Notes in Computer Science, vol. 813, Springer-Verlag, 353–365.
van Oosten, J. (2002). Realizability: a historical essay. *Mathematical Structures in Computer Science* **12** (3) 239–263.
van Oosten, J. (2008). *Realizability, an Introduction to its Categorical Side*. Amsterdam, Elsevier.
Werner, B. (1994). *Une théorie des Constructions Inductives*. Phd thesis, Université Paris VII.

Cite this article: Miquel A (2020). Implicative algebras: a new foundation for realizability and forcing. *Mathematical Structures in Computer Science* **30**, 458–510. https://doi.org/10.1017/S0960129520000079