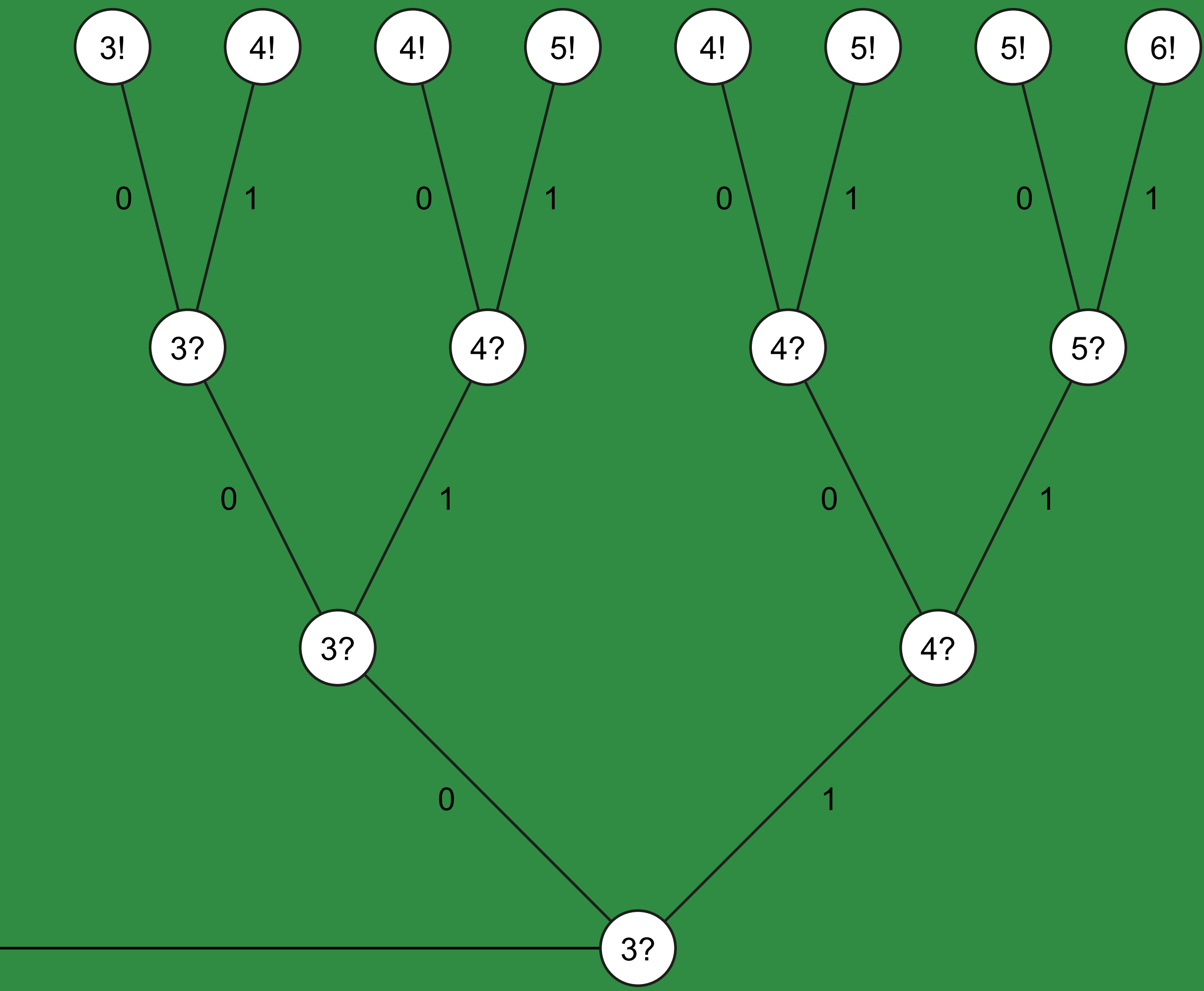
Computability Models and Realizability Toposes



Jetze Zoethout

Computability Models and Realizability Toposes

Thesis committee:
Prof. dr. Steve Awodey, Carnegie Mellon University
Dr. Pieter Hofstra, University of Ottawa
Prof. dr. Giuseppe Rosolini, Università di Genova
Prof. dr. Alex K. Simpson, Univerza v Ljubljani
Prof. dr. Thomas Streicher, Technische Universität Darmstadt

The thesis cover shows a finite part of a tree representing an oracle computation. If the algorithm is presented with oracle α and input n, it iterates the function $x \mapsto x + \alpha(x)$ exactly n times, with initial value n.

ISBN: 978-90-393-7454-2
DOI: 10.33540/1241
Printing: Ridderprint | `www.ridderprint.nl`

Computability Models and Realizability Toposes

Modellen van berekenbaarheid en realizeerbaarheidstoposSen

(met een samenvatting in het Nederlands)

Proefschrift

ter verkrijging van de graad van doctor aan de Universiteit Utrecht op gezag van de rector magnificus, prof. dr. H.R.B.M. Kummeling, ingevolge het besluit van het college voor promoties in het openbaar te verdedigen op maandag 30 mei 2022 des middags te 12.15 uur

door

Jetze Zoethout

geboren op 22 september 1994 te Leeuwarden

Promotor:
Prof. dr. I. Moerdijk

Copromotor:
Dr. J. van Oosten

This thesis was made possible by financial support from the Nederlandse Organisatie voor Wetenschappelijk Onderzoek through the 2012 Spinoza prize (grant number SPI 61-638) awarded to prof. dr. I. Moerdijk.

Contents

Chapter 1

Introduction

This thesis concerns itself with the interplay between two worlds: the world of (abstract) computability theory[1] on the one hand, and the world of category and topos theory, on the other. Before we proceed to exhibit the subject matter of the thesis more precisely in Section 1.2 below, we will first describe the historical developments that led to this research in Section 1.1. Next, we demarcate the contents of the thesis more specifically by discussing the scope of the research, and some limitations, in Section 1.3. Finally, we provide a brief overview of the thesis (Section 1.4), and set forth some conventions and notation that will be used throughout the thesis (Section 1.5).

1.1 Historical overview

The subject matter of this thesis finds its origins in Stephen Cole Kleene's groundbreaking paper [Kle45] on number realizability, where he proposes an interpretation of intuitionistic arithmetic. This section aims to explain how Kleene's work eventually led to research on computability theory and topos theory, and their rich interplay. First, in Section 1.1.1, we describe the birth of realizability in Kleene's paper [Kle45], and some developments that led up to this. Next, Section 1.1.2 describes the subsequent research on realizability up until the late 1970s. Then, in Section 1.1.3, topos theory enters the scene

[1] This discipline also goes by the name 'recursion theory'. The latter finds its origin in the theory of the primitive recursive, and later, the general recursive functions. Since this thesis is mostly concerned with more abstract notions of computability, the name 'computability theory' seems most appropriate for our purposes. However, we will also use the term 'recursion theory' occasionally.

with Martin Hyland's construction of the effective topos, and his collaboration with Peter Johnstone and Andrew Pitts in developing tripos theory. Finally, Section 1.1.4 describes some major strands of further research on realizability toposes that will be relevant for us in Section 1.2.

An overview article of the history of realizability in the twentieth century is the paper [vO02], which serves as the most important secondary source for the first three sections below. A more recent overview article on categorical approaches to computability theory (of which the study of realizability toposes is only a part) is the paper [HS21]. The book [vO08] is a monograph on categorical approaches to realizability specifically.

1.1.1 The origins of realizability

Often in mathematics, a worthwile new strand of research is created by combining ideas from fields that are, up to that point, unrelated, or at least whose relation is not clear. In the case of Kleene's paper [Kle45], these fields are: intuitionistic mathematics, and in particular its metamathematics, on the one hand, and recursion theory on the other. Intuitionistic mathematics was developed in the first quarter of the twentieth century by Luitzen Egbertus Jan Brouwer and is most (in)famous for its rejection of the Law of the Excluded Middle (LEM). It is important to note, though, that this rejection is merely a consequence of Brouwer's philosophy of mathematics, which is grounded in the *a priori* intuition of time. In this sense, Brouwer's philosophy can be said to be broadly Kantian; unlike Kant, however, Brouwer does *not* take the intuition of space to be *a priori* as well.[2] Thus, mathematics is, in the first instance, a mental activity that proceeds exclusively in time. Indeed, according to Brouwer, 'the objects of mathematics are mental constructions in the mind of the (ideal) mathematician' [Tro11, p.157]. One result of this philosophy is that mathematics is not a formal endeavour. So while Brouwer aims to rebuild mathematics on an intuitionistic basis, he does not formulate this intuitionistic basis as any kind of formal system. Instead, this intuitionistic basis consists of basic principles (some of which not only reject, but blatantly contradict LEM) that mostly receive a philosophical motivation.

The task of giving a more formal treatment of intuitionistic mathematics was taken up by others, most prominent among whom is Brouwer's student Arend Heyting. This led to the development of formal systems for intuitionistic propositional and predicate logic, arithmetic and analysis. Attempts were also made to elucidate the notion of intuitionistic proof, the most well-known among these being the Brouwer-Heyting-Kolmogorov ('BHK') interpretation. For example, according to the BHK interpretation, to prove a conditional statement of the form $A \to B$ is to give a construction, or a procedure that could in principle be carried out, transformating any proof of A into a proof of B. However, these developments left plenty to be desired for the mathematician who wishes to

[2] More generally, in the twentieth century, Kant's conception of Euclidean geometry as an *a priori* synthetic science grounded in the intuition of space had fallen out of favor. This was largely due to the discovery of non-Euclidean geometry and its applications in physics.

reason classically. Indeed, the aforementioned clause of the BHK interpretation refers to a primitive notion of 'construction' or 'procedure' that can arguably be understood only by someone already on board with the intuitionistic project. Even worse, any attempt to clarify this primitive notion will likely use implications, which must be understood intuitionistically. This would make the account circular, from the point of view of the classical mathematician. The formal systems of intuitionistic mathematics at least allow the classical mathematician to (formally) prove theorems in intuitionistic mathematics. However, especially at the level of analysis, it also allows one to prove theorems that are clasically *refutable*, the most famous example being the theorem that every function from the reals to the reals is continuous. This could lead to worries about the *consistency* of such formal systems; indeed, the classical mathematician will be hard-pressed to obtain models of formal systems that refute LEM! As we will see below, realizability provides one way to address this worry.

Another result of Brouwer's philosophy of mathematics is the insistence on the fact that mathematical objects can (in principle) be *constructed*. Indeed, Brouwer's intuitionism is a species of constructive mathematics.[3] The constructive nature of intuitionistic mathematics also shows itself in the BHK interpretation, where proofs of certain statements should be *constructions* of some kind. But as we saw above, this is a primitive, informal notion of construction. However, towards the end of the 1930s, mathematical logic had developed a *formal*, mathematical notion of construction, or at least of a computation or algorithm on the natural numbers. Proposals for such a notion were made by, among others, Turing (Turing machines), Church (the lambda calculus) and Kleene himself (the partial recursive functions). They proved the equivalence of these notions, thereby obtaining a robust definition of a computable partial function on the natural numbers. The Church-Turing Thesis states that any partial function on the natural numbers which is (in principle) intuitively computable, is a partial recursive function.

Around 1940, the constructive nature of intuitionistic mathematics led Kleene to conjecture the following (see also [Kle73, p.4]). Suppose a sentence of the form $\forall x \, \exists y \, A(x, y)$ is provable in a formal system of intuitionistic arithmetic. Then there must be a computable function f on the natural numbers such that for each natural number n, the sentence $A(n, f(n))$ is true. Using the Church-Turing Thesis, one can require this function f to be recursive, which turns the conjecture into a mathematical statement (cf. Church's Rule in Section 1.1.2 below). In order to prove this conjecture, Kleene devised, for each natural number e and sentence A in the language of arithmetic, a notion 'e realizes A'. Before we can give Kleene's definition, which is by induction on A, we need some notation. Let $\langle \cdot, \cdot \rangle \colon \mathbb{N}^2 \to \mathbb{N}$ be a primitive recursive coding of pairs of natural numbers, such that the decoding functions, that we will denote by $(\cdot)_0$ and $(\cdot)_1$, are also

[3] The adjectives 'intuitionistic' and 'constructive' are sometimes used interchangably, but it seems more appropriate to say Brouwer's intuitionism is a *kind* of constructive mathematics. It is certainly not the only kind, nor, arguably, the first (see also [Tro11]). Moreover, nowadays 'intuitionistic' can refer more broadly to *predicative* mathematics (e.g., Martin-Löf's type theory), which can be understood both constructively and classically.

primitive recursive. Moreover, for a natural number e, write φ_e for the partial recursive function with Gödel number e. The definition of number realizability is then as follows:

- If A is atomic, then e realizes A iff A is true. In particular, no natural number realizes $\bot$.
- e realizes $A \wedge B$ iff e_0 realizes A and e_1 realizes B.
- e realizes $A \vee B$ iff either $e_0 = 0$ and e_1 realizes A, or $e_0 = 1$ and e_1 realizes B.
- e realizes $A \to B$ iff, for any realizer a of A, we have that $\varphi_e(a)$ is defined and a realizer of B.
- e realizes $\forall x\, A(x)$ iff, for every natural number n, we have that $\varphi_e(n)$ is defined and a realizer of $A(n)$.
- e realizes $\exists x\, A(x)$ iff e_1 realizes $A(e_0)$.

The reader will notice that the clause for implication is reminiscent of the corresponding clause of the BHK interpretation: a realizer of $A \to B$ should code an *algorithm* that turns realizers of A into realizers of B. In fact, the other clauses of the realizability are similarly analogous to the corresponding clauses of the BHK interpretation. It should be mentioned, however, that Kleene's definition has the advantage over the BHK interpretation that, unlike the latter, it is intelligible to the classical mathematician without further ado. Interestingly, the BHK interpretation was *not* what inspired Kleene's definition of number realizability, as transpires from his remarks in [Kle73].

The relevant system of intuitionistic arithmetic is nowadays known as Heyting Arithmetic, and denoted by **HA**. In [Kle45], Kleene proved the following theorem: if a sentence A is provable in Heyting Arithmetic (denoted $\mathbf{HA} \vdash A$), then A is realized by some natural number. In particular, if $\mathbf{HA} \vdash \forall x\, \exists y\, A(x, y)$, then $\forall x\, \exists y\, A(x, y)$ must have a realizer. From this is follows easily that there exists a recursive function f such that $A(n, f(n))$ is realizable for every natural number n. This is not quite Kleene's conjecture yet, since not every realizable sentence is actually true (see also below)! But Kleene was able to adjust his realizability definition to obtain the following result: if $\mathbf{HA} \vdash \forall x\, \exists y\, A(x, y)$, then there exists a recursive function f such that $\mathbf{HA} \vdash A(n, f(n))$ for each natural number n. This does yield Kleene's conjecture, since every theorem of **HA** is true.

Realizability can be used for other purposes besides proving Kleene's conjecture. For example, one may construct a formula $A(x)$ in the language of arithmetic such that $\neg\forall x\, (A(x) \vee \neg A(x))$ has a realizer. (Here we use $\neg B$ as an abbreviation of $B \to \bot$.) From this it follows that $\mathbf{HA} + \neg\forall x\, (A(x) \vee \neg A(x))$ is consistent, for this particular formula $A(x)$. Therefore, we see that realizability can establish the consistency of theories that refute classical logic. Moreover, from Kleene's adjusted realizability definition mentioned above, one can also derive the disjunction and existence properties for **HA**, which had previously been established for intuitionistic predicate logic (without axioms). These state, respectively: if $\mathbf{HA} \vdash A \vee B$, then $\mathbf{HA} \vdash A$ or $\mathbf{HA} \vdash B$; and if $\mathbf{HA} \vdash \exists x\, A(x)$, then for some natural number n, we have $\mathbf{HA} \vdash A(n)$. So we see that realizability can also be used to establish properties of intuitionistic formal systems.

1.1.2 Subsequent developments in realizability

As we saw in the previous section, realizability can be used to prove the consistency of (classically refutable) principles, and to establish properties of formal intuitionistic systems. In the decades following Kleene's paper [Kle45], many variations on realizability were introduced in order to obtain results of a similar character.

As was already realized by Kleene himself, the definition of 'e realizes A' for any given sentence A can be formalized in Heyting Arithmetic itself; see [Kle45], or [Nel47] by Kleene's student David Nelson for a more detailed exposition. Indeed, the clauses for number realizability above mention a primitive recursive pairing of the natural numbers, which can be respresented in **HA**, and recursive function application (in terms of their Gödel numbers). The latter can be formulated using Kleene's T-predicate and U-function. The predicate $T(x, y, z)$ expresses 'z codes a (finished) computation performed by the Turing machine with Gödel number x when presented with input y', and the U-function extracts the final output result from such a computation z. Both T and U are primitive recursive, hence can be represented in **HA** as well. Let us write $xy{\downarrow} \wedge A(xy)$ as an abbreviation of $\exists z\,(T(x, y, z) \wedge A(U(z)))$. Now, we may, for each *formula* A in the language of arithmetic, define a new formula $x \mathbf{r} A$, where x does not occur free in A, and where the free variables of $x \mathbf{r} A$ are among x and the free variables of A. The definition is by induction on A and mimics the definition of number realizability. For example, the clause for universal quantification reads:

$$x \mathbf{r} \forall y\, A(y) \equiv \forall y\,(xy{\downarrow} \wedge xy \mathbf{r} A(y)).$$

The main result is then that every theorem of **HA** is *provably* realizable in **HA**. That is, if $\mathbf{HA} \vdash A$, then $\mathbf{HA} \vdash \exists x\,(x \mathbf{r} A)$. Using a variation of formalized realizability called **q**-realizability, one may obtain what is known as *Church's Rule* for **HA**:

$$\text{if } \mathbf{HA} \vdash \forall x\, \exists y\, A(x, y), \text{ then } \mathbf{HA} \vdash \exists z \forall x\,(zx{\downarrow} \wedge A(x, zx)).$$

This is stronger than Kleene's conjecture above, since it implies that we may take f to be provably total in **HA**. Let us list some further variations on realizability; we will not spell out any technical details, but only give the main ideas and/or the main results derived from these notions of realizability.

Modified realizability was introduced by Kreisel in 1959 [Kre59]. In order to explain its main idea, consider a sentence of the form

$$\forall x\, \exists y\, A(x, y) \to B, \tag{1.1}$$

where for simplicity, we assume that A is atomic. Realizers of $\forall x\, \exists y\, A(x, y)$ are essentially Gödel numbers of total recursive functions that assign to each x a witness y of $A(x, y)$ (and a realizer for $A(x, y)$, but this gives no information if A is atomic). A realizer e of the formula in (1.1) must then send such Gödel numbers to realizers of B. Here we 'forget' that the input of e is supposed to code a total recursive function. Modified realizability takes this into account. Not only must a realizer e of (1.1) send realizers of $\forall x\, \exists y\, A(x, y)$ to realizers of B. Also, if we feed e the Gödel number of a total recursive function (which can be said to be of the 'appropriate form' to realize a sentence like $\forall x\, \exists y\, A(x, y)$), then the result must be defined and, in fact, be of the 'appropriate form' to realize B (which may be a complex sentence). So we see that modified realizability 'remembers' more about the structure of the formula to be realized than ordinary realizability. Modified realizablity can be used to show, among other things, that Markov's Principle:

$$\forall x\, (A(x) \vee \neg A(x)) \wedge \neg\neg \exists x\, A(x) \to \exists x\, A(x)$$

is not derivable in **HA**.

We should mention that the notion of modified realizability explained above is not Kreisel's original notion, which he devised for a version of Heyting Arithmetic with *higher types*, denoted by $\mathbf{HA}^{\omega}$. The system $\mathbf{HA}^{\omega}$ has a model known as the Hereditarily Recursive Operations (HRO), which is definable in **HA**. This yields an interpretation of Kreisel's definition in untyped intuitionistic arithmetic, which is what we have explained here.

Extensional realizability was first defined in the early 1980s [Pit81]. Consider, again, the sentence in (1.1). If the sentence $\forall x\, \exists y\, A(x, y)$ has a realizer, then it will have many realizers: indeed, any total recursive function has infinitely many indices. But two realizers of $\forall x\, \exists y\, A(x, y)$ could also code *different* recursive functions, for $A(x, y)$ may have multiple witnesses y. Such realizers can be said to differ in an 'extensional' way. The idea of extensional realizability is to take the extensional behavior of the realizers into account. Thus, a realizer of (1.1) should send 'extensionally equivalent' (i.e., coding the same function) realizers of $\forall x\, \exists y\, A(x, y)$ to 'extensionally equivalent' realizers of B. It turns out that there are several non-equivalent ways of doing this. In [vO91], two versions are discussed. One is based on 'collapsing' HRO into an extensional type structure, denoted by $\mathrm{HRO}^{\mathrm{E}}$, while the other is based on the type structure of Hereditarily Effective Operations (HEO). Interestingly, while $\mathrm{HRO}^{\mathrm{E}}$ and HEO are equivalent as type structures (this was first proved in [Bez85]; see also [LN15, Section 9.2] for a modern treatment), the resulting two notions of extensional realizability are not equivalent.

Lifschitz realizability was introduced in the late 1970s [Lif79], where Lifschitz used it to show that the scheme known as *Church's Thesis* (CT_0):

$$\forall x\, \exists y\, A(x, y) \to \exists z \forall x\, (zx{\downarrow} \wedge A(x, zx))$$

does not follow from the related scheme $\mathrm{CT}_0!$:

$$\forall x\, \exists! y\, A(x, y) \to \exists z \forall x\, (zx{\downarrow} \wedge A(x, zx))$$

Here the quantifier ∃! stands for 'there exists a *unique* ...'. Note that, while $\mathrm{CT}_0!$ roughly states that every definable total function is recursive, CT_0 states that every definable total relation *contains* a recursive function, so the latter principle asserts some kind of (countable) choice.

As we explained above, Kreisel's original notion of realizability was designed for intuitionistic arithmetic in higher types, which lands one in the realm of intuitionistic analysis. Another notion of realizability designed for analysis is Kleene's *function realizability.* While versions of it occurred earlier in the literature, the standard reference for function realizability is [KV65]. Here, Kleene and Vesley introduce a system of intuitionistic analysis which has some highly non-classical consequences, such as the fact that every real function is continuous. It is shown that every theorem of the system is realizable with respect to function realizability, which in particular establishes the consistency of the system. In function realizability, the realizers are not natural numbers, but *functions* from the natural numbers to the natural numbers. In addition, in order for a sentence to count as realizable, it should not merely have a realizer, but it should have a *recursive* realizer. This is in fact a very early example of a general pattern that has become known as relative realizability, more on which in Section 1.1.4 below.

The notions of realizability discussed above all exploit some notion of computation on natural numbers, or in the case of function realizability, computation on functions. One may also consider more 'abstract' notions of computability. One example of such a more abstract notion is the λ-calculus. While this can certainly be used to represent numerical functions, we can also treat it as an abstract calculus and then consider various of its models. There is in fact an older such calculus, known as combinatory logic, defined by Moses Schönfinkel in [Sch24] (see [Sch67] for an English translation) and studied later by Haskell Curry in [Cur30]. Combinatory logic contains certain special constant symbols, called combinators, that allow one to perform computations. In fact, every model of the λ-calculus is also a model of combinatory logic. In the 1970s, John Staples gave an abstract notion of realizability where the realizers are terms of combinatory logic [Sta73]. A few years later, Solomon Feferman introduced a partial version of combinatory logic [Fef75], where the 'partiality' takes into account that the application of an algorithm to an input may not terminate. The models of Feferman's calculus are known as *partial combinatory algebras* (PCAs), or sometimes as 'models of computation', or 'Schönfinkel algebras' [Joh13]. It is useful to think of the elements of a PCA simultaneously as (codes of) algorithms and as inputs to such algorithms. We can apply an element of a PCA (thought of as an algorithm) to another element (thought of as an input), and this may or may not yield an outcome. As we will see below, these very general models of computation give rise to the class of realizability toposes. In fact, function realizability arises from a partial combinatory algebra, which is now known as Kleene's second model.

The list of variations on realizability given here is certainly not exhaustive. For example, Gödel's *Dialectica interpretation* [Göd58] is decidedly realizability-like. More recently, Krivine has devised a realizability notion for classical set theory [Kri09]. A monumental overview of all the realizability notions existing at the time, and their applications, is [Tro73].

Besides the development of variations on realizability, some attention was also directed at studying the properties of realizability itself. One result in this area is that for any sentence A, we have $\mathbf{HA} \vdash \exists x\,(x\ \mathbf{r}\ A) \leftrightarrow \exists y\,(y\ \mathbf{r}\ \exists x\,(x\ \mathbf{r}\ A))$, i.e., 'realizability is idempotent' [Tro73, Theorem 3.2.16]. Another result is the *axiomatization* of realizability. Given the notion of formalized realizability, a natural question to ask is which sentences are *provably* realizable in **HA**. As we have seen above, this includes at least the theorems of **HA**. But this is not all, for there is a scheme, slightly more general than CT_0 and known as Extended Church's Thesis (ECT_0) which is not provable in **HA**, but is provably realizable in **HA**. It turns out that adding ECT_0 to **HA** completely axiomatizes realizability, in the sense that $\mathbf{HA} \vdash \exists x\,(x\ \mathbf{r}\ A)$ if and only if $\mathbf{HA} + \mathrm{ECT}_0 \vdash A$. Moreover, we have $\mathbf{HA} + \mathrm{ECT}_0 \vdash A \leftrightarrow \exists x\,(x\ \mathbf{r}\ A)$, so from the point of view of $\mathbf{HA} + \mathrm{ECT}_0$, truth and realizability coincide. These results can be found, e.g., in [Tro73, Theorem 3.2.18].

1.1.3 The effective topos and tripos theory

As we have seen in the previous section, realizability and its many variations can be used to establish various results on intuitionistic formal systems. As such, up to the late 1970s, realizability was primarily a proof-theoretic tool, and treatments of realizability were thus of a rather syntactical nature. Around 1980, Hyland, together with Johnstone and Pitts, created a wholly new, more semantical approach to realizability, by combining realizability with topos theory [Hyl82].

Topos theory originated from Alexander Grothendieck's pioneering work in algebraic geometry, which led to the notion of a category of sheaves on a site, also known as a Grothendieck topos. These toposes are sometimes referred to as 'generalized spaces', because they vastly generalize point-set topology. At the same time, Grothendieck toposes behave very much like the category of sets, and can thus be regarded as a 'universe for doing mathematics'. More precisely, every Grothendieck topos can be regarded a model for higher-order many-sorted predicate logic. Crucially, the internal logic of a Grothendieck topos need not be classical, so Grothendieck toposes yields models of *constructive* mathematics. Around 1970, William Lawvere and Myles Tierney isolated a number of elementary properties of Grothendieck toposes that suffice for interpreting constructive mathematics, leading to the definition of an elementary topos. Here 'elementary' refers to the fact that the definition of an elementary topos, unlike that of a Grothendieck topos, does not depend on the category of sets. In fact, a Grothendieck topos can also be defined as an elementary topos that satisfies some additional properties that depend on set theory, and in particular, on a notion of 'smallness'.

The paper [Hyl82] constructs an elementary topos where the internal logic is governed by recursive function application. For this reason, this topos is known as the *effective topos*, and is usually denoted by $\mathcal{E}\!f\!f$. This topos has a natural numbers object, which means that we can interpret higher-order intuitionistic arithmetic in $\mathcal{E}ff$. For first-order arithmetic, one gets that the sentences that are true internally in $\mathcal{E}ff$ are precisely those that are realizable in Kleene's original sense. In this way, the effective topos can be said to be a 'topos for number realizability'. But $\mathcal{E}ff$ is much more than that: being a topos, it yields a model of higher-order arithmetic and analysis. For example, we know that the axiom scheme CT_0 is true in $\mathcal{E}ff$, since every instance is realizable. Using quantification over functions, we can also formulate a version of Church's Thesis as a single axiom, which is known as CT:

$$\forall f^1\, \exists e^0\, \forall x^0\, (ex{\downarrow} \wedge ex = f(x)).$$

Here the superscript indicates the type of a variable, i.e., e and x are meant to range over the natural numbers, whereas f ranges over *functions* from the set of natural numbers to itself. Internally in $\mathcal{E}ff$, CT is true, as is the theorem that every real function is continuous. Another important example of a classically false result that holds in $\mathcal{E}ff$ was described in [Hyl88] and [HRR90]. Here it is shown that one may construct an internal category in $\mathcal{E}ff$ which is complete (in a suitable, internal sense), but not a preorder.

Crucially, the effective topos is not a Grothendieck topos, which means that it yields a genuinely new model of constructive mathematics. However, the construction of $\mathcal{E}ff$ bears some resemblance to the construction of certain Grothendieck toposes, namely the *localic* Grothendieck toposes. These are toposes of sheaves on a complete Heyting algebra $\mathcal{H}$, which may alternatively be described as the category of $\mathcal{H}$-valued sets [FS79]. The construction can be decomposed into two steps. First, one constructs a model of many-sorted intuitionistic predicate logic *without equality*, where the sorts are sets. The possible predicates on a set X come from the set $\mathcal{H}^X$, which is also a complete Heyting algebra when ordered pointwise. The Heyting structure is used to interpret the connectives, while the completeness of the Heyting algebra is used to interpret the quantifiers. In the second stage, one 'adds equality', which yields the topos of $\mathcal{H}$-valued sets. The construction of the effective topos proceeds similarly, but here the possible predicates on X are in $\mathcal{P}(\mathbb{N})^X$. This set is preordered as follows: $\varphi \leq \psi$ iff there is an $e \in \mathbb{N}$ such that, for all $x \in X$ and $n \in \varphi(x)$, we have $en{\downarrow}$ and $en \in \psi(x)$. With this preorder, $\mathcal{P}(\mathbb{N})^X$ becomes a Heyting prealgebra, but it is not a complete one. However, it is still possible to interpret the quantifiers, and adding equality in a way analogous to the case of $\mathcal{H}$-valued sets yields exactly the effective topos.

This led to the following question: 'is there a common generalisation, with useful properties, of the constructions of $\mathcal{H}$-valued sets and of the effective topos?' [Pit02, p.265]. The answer to this question was given in [HJP80] and [Pit81], which introduce the notion of a tripos.[4] This notion generalizes the first step in the construction of both $\mathcal{H}$-valued sets and the effective topos, and for any tripos, one can 'add equality', producing a corresponding topos. The

[4] Officially, this is an acronym for Topos Representing Indexed Partially Ordered Set, but unofficially, it is a pun on the name of the famous mathematics exam in Cambridge.

construction of the tripos underlying $\mathcal{E}ff$ (which is known as the effective tripos) works equally well when $\mathcal{P}(\mathbb{N})$ is replaced by $\mathcal{P}(A)$, where A is a partial combinatory algebra. By the general theory of triposes, each PCA A then gives rise to a topos, which is called its realizability topos and denoted by $\mathsf{RT}(A)$.

Since the discovery of the effective topos and the more general class of realizability toposes, many variations on realizability have found a 'second home' in the world of topos theory as well. As we mentioned above, function realizability actually arises from a *relative* PCA, more on which below. We briefly discuss the other variations on realizability we mentioned.

- The paper [vO97b] describes a topos for modified realizability. It turns out that this topos is a subtopos of $\mathcal{E}ff(\mathsf{Set}^{\to})$, the realizability topos constructed over the Sierpiński topos $\mathsf{Set}^{\to}$ rather than the topos of sets. In fact, the picture can be painted in a bit more detail: the effective topos is an open subtopos of $\mathcal{E}ff(\mathsf{Set}^{\to})$, and the modified realizability topos is its closed complement. Alternatively, the modified realizability topos can also be described as a genuine realizability topos (i.e., of the form $\mathsf{RT}(A)$) for a relative *ordered* PCA (more on which below); see [Hof06, p.253].
- The thesis [Pit81] also describes a topos for the 'HEO version' of extensional realizability. In [vO97a], it is shown that this extensional realizability topos is a subtopos of a realizability topos for an ordered PCA. On the other hand, Pitts' extensional realizability topos cannot itself be of the form $\mathsf{RT}(A)$.
- The thesis [vO91] describes a topos for Lifschitz realizability, and shows that it is a subtopos of $\mathcal{E}ff$. The Lifschitz realizability topos cannot itself be of the form $\mathsf{RT}(A)$ because the axiom of countable choice, which holds in every realizability topos, fails in it. Compare this with Lifschitz' original application of his realizability notion discussed on page 6.

Even a result such as the idempotence of realizability received a topos theoretic version in the form of the effective monad [Pit81, chapter 7].

In the next section, we will proceed to discuss further research on realizability toposes. We should mention, however, that $\mathcal{E}ff$ specifically has also been studied far more extensively than we have discussed here. There is also research on the topos theory of $\mathcal{E}ff$, e.g., its subtoposes. The paper [Hyl82] already shows that the lattice of Turing degrees embeds into the lattice of Lawvere-Tierney topologies on $\mathcal{E}ff$; see also [Pho89]. The topos of sets sits inside $\mathcal{E}ff$ as the subtopos of $\neg\neg$-sheaves, and we saw above that Lifschitz realizability provides another subtopos of $\mathcal{E}ff$. Yet another one is described in [Pit81, example 5.8], and studied further in [vO14]. A systematic study of the subtoposes of $\mathcal{E}ff$ is [LvO13], and recently, [Kih21].

1.1.4 Realizability toposes

In this section, we describe several strands of research on realizability toposes that have emerged since their introduction in the 1980s. These strands are not wholly independent of each other, and in fact, all of them will play in a role in this thesis.

Before we proceed, we record some general facts about $\mathsf{RT}(A)$. Unlike Grothendieck toposes, which have a canonical geometric morphism to the topos of sets, realizability toposes come with a canonical *inclusion* of the topos of sets, denoted $\Gamma \dashv \nabla \colon \mathsf{Set} \hookrightarrow \mathsf{RT}(A)$. In fact, as we already saw in the case of $\mathcal{E}ff$, this inclusion is the subtopos of $\neg\neg$-sheaves. The inverse image Γ is the global sections functor, while the direct image ∇ is known as the constant objects functor. Another interesting full subcategory of $\mathsf{RT}(A)$ consists of the objects that are *separated* for the $\neg\neg$-topology, or equivalently, the subobjects of objects in the image of ∇. This subcategory is known as the category of assemblies, and denoted by $\mathsf{Asm}(A)$. The category of assemblies is somewhat easier to understand than the realizability topos, and already has quite some structure. Indeed, it is a quasitopos, but not a topos; as we shall see below, it fails to be an exact category.

Relative realizability

As we have seen above, a PCA A is an abstract model of computation, where we may 'apply' one element to another. In relative realizability, one additionally equips the PCA A with a privileged subset $A^{\#}$. The intuition behind this is as follows: while all the elements of A still represent computations, the elements of $A^{\#}$ represent computations that can actually be carried out, or implemented. As we mentioned above, function realizability is an early example of this. While all functions $\mathbb{N} \to \mathbb{N}$ can act as realizers (these make up the PCA A), a sentence counts as realizable only if it is realized by a *recursive* function (these form the privileged subset $A^{\#}$). The theme of actually computable operations acting on a wider class of possibly non-computable things is present in other work of Kleene as well, most notably his work on higher-order computability [Kle59].

Given a relative PCA, which we will denote by $(A, A^{\#})$, one can construct a corresponding realizability tripos, hence also a realizability topos $\mathsf{RT}(A, A^{\#})$. The paper [ABS02] shows that there is a logical functor $\mathsf{RT}(A, A^{\#}) \to \mathsf{RT}(A)$. Moreover, $A^{\#}$ is a PCA in its own right, and there exists a local geometric morphism $\mathsf{RT}(A, A^{\#}) \to \mathsf{RT}(A^{\#})$. In particular, this means that $\mathsf{RT}(A^{\#})$ sits inside $\mathsf{RT}(A, A^{\#})$ as a subcategory in two different ways. In [BvO02], this situation is analyzed further. For example, it is shown that the logical functor $\mathsf{RT}(A, A^{\#}) \to \mathsf{RT}(A)$ is an instance of a general construction on toposes known as a filter quotient.

As we will see below and further on in the thesis, the theory of relative realizability toposes $\mathsf{RT}(A, A^{\#})$ can at times be a bit more cumbersome than the theory of ordinary realizability toposes $\mathsf{RT}(A)$. For example, while relative realizability toposes still carry an inclusion of Set, the inverse image is *not* in general the global sections functor anymore. Nevertheless, relative realizability is a very useful generalization of realizability, and it tends to turn up naturally. The latter, for example, is advocated by the paper [Hof06], which we will discuss in more detail on page 14 below. As another example, when we discuss [Ste13] in Chapter 4, we will see that slicing of realizability toposes forces us to consider relative notions of realizability.

Applicative morphisms

Thus far, we have discussed PCAs, but not what would constitute a *morphism* between PCAs. On the other hand, for the categories $\mathsf{Asm}(A)$ and $\mathsf{RT}(A)$, we do have notions of a morphism: functors that preserve certain structure, and in the case of $\mathsf{RT}(A)$, geometric morphisms. Therefore, we can ask the question: does there exist a notion of morphism between PCAs that corresponds to certain 'nice' functors between the categories of assemblies and/or the realizability toposes? Such a notion, known as *applicative morphisms*, was introduced by John Longley in [Lon94]. Crucially, an applicative morphism between PCAs $A \to B$ is not a 'structure-preserving map' or a 'homomorphism of models of partial combinatory logic' in any sense. Instead, an applicative morphism $A \to B$ can be seen as a 'simulation' of the model of computation A inside the model B. This fits the intuition behind PCAs as models of computation quite well, and there are many interesting examples of applicative morphisms. In addition, Longley's notion has strong mathematical credentials: in [Lon94], it is shown that the following are equivalent (in a more precise sense than we specify here):

- an applicative morphism $A \to B$;
- a regular functor $\mathsf{Asm}(A) \to \mathsf{Asm}(B)$ that commutes with the Γ-functors;
- a regular functor $\mathsf{Asm}(A) \to \mathsf{Asm}(B)$ that commutes with the ∇-functors;
- a regular functor $\mathsf{RT}(A) \to \mathsf{RT}(B)$ that commutes with the ∇-functors.

It should be noted that this list must be adjusted if we allow relative PCAs; we will discuss this in Section 1.2 below, and in more detail in Chapter 3.

As was shown in [Lon94], PCAs and applicative morphisms form a category PCA, which is in fact enriched over preorders. In particular, given an adjunction $A \underset{\longleftarrow}{\overset{\longrightarrow}{\perp}} B$ in PCA, we also get an adjunction $\mathsf{RT}(A) \underset{\longleftarrow}{\overset{\longrightarrow}{\perp}} \mathsf{RT}(B)$, that is, a geometric morphism $\mathsf{RT}(B) \to \mathsf{RT}(A)$. In this case, the direct image functor $\mathsf{RT}(B) \to \mathsf{RT}(A)$ is always regular, since it arises from an applicative morphism $B \to A$. Therefore, we can ask more generally: when is $\mathsf{RT}(A) \to \mathsf{RT}(B)$ the inverse image of a geometric morphism, i.e., when does it have a right adjoint that is not necessarily regular? The answer to this question was given in [HvO03], where it is shown that $\mathsf{RT}(A) \to \mathsf{RT}(B)$ has a right adjoint if and only if $A \to B$ satisfies a condition called 'computational density'. This paper also refines Longley's study of applicative morphisms by employing the more general notion of an *ordered* PCA. While the definition of ordered PCAs occurs already in [vO97a], the paper [HvO03] is the first to fully develop the theory of ordered PCAs. Hofstra and Van Oosten define a monad T on the category of ordered PCAs, and recover Longley's category PCA as the full subcategory of $\mathrm{Kl}(T)$ on the 'ordinary' PCAs. In other words, an applicative morphism is the same thing as a morphism of order-PCAs $A \to TB$, or a T-algebra morphism $TA \to TB$. In fact, [FvO14] shows that, for computationally dense $A \to B$, the right adjoint $\mathsf{RT}(B) \to \mathsf{RT}(A)$ arises from a morphism of OPCAs $TB \to TA$ that is *not* necessarily a T-algebra morphism. In this way, the more general framework of ordered PCAs allows us to reconstruct the adjunction $\mathsf{RT}(A) \underset{\leftarrow}{\overset{\to}{\perp}} \mathsf{RT}(B)$ at the level of PCAs after all, even when the direct image is not regular.

Finally, Johnstone has shown in [Joh13] that for *any* geometric morphism $\mathsf{RT}(B) \to \mathsf{RT}(A)$, the inverse image commutes with the ∇-functors. Since this inverse image is also regular, Longley's results imply that it must arise from an applicative morphism $A \to B$. Putting all these results together, we see that there is a correspondence between geometric morphisms $\mathsf{RT}(B) \to \mathsf{RT}(A)$ on the one hand, and computationally dense applicative morphisms $A \to B$ on the other.

The class of realizability toposes

Realizability toposes, like Grothendieck toposes, have been defined by means of the way they are *constructed*; the former are constructed on the basis of a PCA, while the latter are toposes of sheaves on a site.[5] As we mentioned, we can give an alternative characterization of Grothendieck toposes as elementary toposes that satisfy some 'size conditions'. Now we may ask: can we similarly describe realizability toposes in terms of their properties, rather than the way they are constructed?

The first step on the journey towards such a description, is research from the late 1980s describing realizability toposes as the result of a universal construction. The paper [CC82] describes how to turn a left exact category $\mathcal{C}$ into an exact category in a universal way. The resulting exact category is called the ex/lex completion of $\mathcal{C}$, and denoted by $\mathcal{C}_{\text{ex/lex}}$. The universal property of the construction can be expressed by saying that it is left adjoint, in a suitable 2-categorical sense, to the inclusion $\mathsf{EX} \hookrightarrow \mathsf{LEX}$, where EX denotes the category of exact categories and regular functors, and LEX denotes the category of left exact catgeories and functors. The original category $\mathcal{C}$ sits inside $\mathcal{C}_{\text{ex/lex}}$ as a full subcategory, namely as the full subcategory of the projective objects of $\mathcal{C}_{\text{ex/lex}}$. Moreover, an exact category $\mathcal{E}$ is an ex/lex completion if and only if its full subcategory on the projective objects is closed under finite limits, and every object of $\mathcal{E}$ is covered by a projective object. In a realizability topos $\mathsf{RT}(A)$, the projective objects form a subclass of the assemblies, known as the *partitioned* assemblies. One can show that these are closed under finite limits, and that every object of $\mathsf{RT}(A)$ can be covered by a partitioned assembly. This implies that that $\mathsf{RT}(A)$ is the ex/lex completion of its full subcategory on the partitioned assemblies. These results are proved, for the case of $\mathcal{E}ff$, in [RR90].

[5] There is a conceptual difference between the two cases, however. While non-equivalent sites may very well give rise to equivalent sheaf toposes, Longley's results imply that equivalent realizability toposes must arise from equivalent PCAs. On the other hand, it is not very easy to recognize when two PCAs are equivalent!

In fact, there is a bit more to the story. The inclusion $\mathsf{EX} \hookrightarrow \mathsf{LEX}$ can be factored as $\mathsf{EX} \hookrightarrow \mathsf{REG} \hookrightarrow \mathsf{LEX}$, where REG denotes the category of regular categories and functors. Accordingly, the ex/lex completion can be performed in two stages, where one first completes a left exact category into a regular category (known as the reg/lex completion) and then into an exact category (known as the ex/reg completion). It turns out that the reg/lex completion of the category of partitioned assemblies is $\mathsf{Asm}(A)$, which means that $\mathsf{RT}(A)$ can also be described as $\mathsf{Asm}(A)_{\mathrm{ex/reg}}$ (see [CFS88] and [Car95]). In fact, it can be shown that every topos arising from a tripos is an ex/reg completion.

It is somewhat surprising that the exact completion, which is only meant to yield exact categories, delivers so much more in this case, namely toposes. The conditions under which this happens have been studied by Matías Menni in [Men00].

Another important step towards characterizing realizability toposes is taken in Hofstra's paper [Hof06]. This paper introduces the notion of a Basic Combinatorial Object (BCO), which is a poset Σ together with a privileged set of partial functions from Σ to itself that count as 'realizable' or 'computable'. Every relative ordered PCA gives rise to a BCO, where the poset is A and the set of partial functions consists of those that are represented by an element of $A^{\#}$. Every BCO gives rise to an indexed preorder, where the predicates on a set X are functions on X whose values are downwards closed subsets of Σ. The main result of [Hof06] is that this indexed preorder is a tripos if and only if the BCO arises from a relative ordered PCA. Thus, relative ordered PCAs are isolated as a natural class of objects that give rise to triposes, and from there, to toposes.

Recently, Jonas Frey, drawing on the research on exact completions, and Hofstra's work on BCOs described above, was able to give a characterization of realizability toposes purely in terms of their properties. The characterization and its proof can be found in [Fre19]; a much more extensive survey, which also treats relative and typed PCAs, is [Fre14].

The results above all concern realizability toposes constructed over the base category of sets, and to a certain extent, they depend on specific properties of this base category; most notably the Axiom of Choice (AC). We can wonder what happens if we extend our possible bases to a wider class of categories, which possibly do not satisfy AC or even LEM. In fact, we have already seen several examples of realizability toposes constructed over other bases. The papers [vO97b] and [BvO02] construct realizability toposes over the Sierpiński topos $\mathsf{Set}^{\to}$, and Pitts' effective monad involves constructing the effective topos over arbitrary base toposes with a natural numbers object. A very general project of considering non-standard bases is undertaken by Wouter Stekelenburg in [Ste13], which considers PCAs internal to a Heyting category, and their corresponding categories of assemblies, which are called 'realizability categories'. In a spirit similar to Frey's results, Stekelenburg also arrives at a kind of characterization of these realizability categories [Ste13, Theorem 2.2.17], even though this characterization still mentions the presence of the underlying PCA in the realizability category itself.

Oracle and higher-order computation

In the first instance, recursion theory concerns computations on the natural numbers and the resulting partial recursive functions. From there, one can also consider notions of computation involving non-computable functions. An early example of such a notion is computation with a Turing machine equipped with an *oracle* for a not necessarily computable function or set. Later, Kleene ([Kle59]; see also [KM77]) developed the rich theory of *higher-order* recursion theory, where computations not only take numbers as inputs, but also functions. We can wonder whether this theme of 'computability with functions' can be studied using the abstract approach to computability theory provided by the theory of PCAs. An example of this is given by the PCA underlying function realizability, whose elements are indeed functions rather than numbers. A related PCA, whose elements are *partial* functions on the natural numbers, was defined in [vO99]. A comprehensive survey of higher-order computability theory using abstract computability models, also in the typed setting, is [LN15].

We can also approach the matter slightly differently and, given an arbitrary PCA A, try to construct new PCAs expressing computability notions involving functions on A. A first construction of this kind is [vO06], which, given a PCA A and a (partial) function $\alpha\colon A \to A$, defines a new PCA $A[\alpha]$ expressing 'A-computability with an oracle for α'. Applying this construction to the PCA of natural numbers and recursive function application yields something equivalent to the original notion of computation with an oracle. There is an applicative morphism $A \to A[\alpha]$, which is the 'universal solution' to making α computable. This applicative morphism is computationally dense, so it yields a geometric morphism $\mathsf{RT}(A[\alpha]) \to \mathsf{RT}(A)$. In fact, as in the case of $\mathcal{E}\!f\!f$, this is a geometric *inclusion*. The construction of $A[\alpha]$ was generalized to functionals of second order in [FvO16]; applications of the construction $A[\alpha]$ to certain specific PCAs can be found in [vOV18]. Other constructions in this theme can be found in [vO11], which generalizes Kleene's second model and the PCA of partial functions from [vO99] to arbitrary PCAs.

1.2 Main results

In the previous section, we have discussed the development of an intimate connection between abstract computability theory and category theory, via Kleene's notion of realizability. More precisely, we have abstract models of comptability, called partial combinatory algebras, and for each PCA A, we can construct a category of assemblies $\mathsf{Asm}(A)$ and a realizability topos $\mathsf{RT}(A)$. Given this connection, we can ask how the world of PCAs on the one hand, and the world of categories and toposes on the other, interact with one another. The following, still very broadly formulated, question constitutes the main research question of this thesis.

Question 1.2.1. *(i) Consider a construction on categories and/or toposes, e.g., taking the product of two categories. Are categories of assemblies and/or realizability toposes closed under this construction? If not, then what kind of category/topos do these constructions yield? If so, can we describe the construction 'downstairs', that is, at the level of PCAs?*

Similarly, how do certain functors between categories of assemblies and/or realizability toposes correspond to morphisms between their 'underlying' PCAs?

(ii) Conversely, given a construction on PCAs, how does this construction manifest itself in the corresponding categories of assemblies and/or realizability toposes? Can we describe this in familiar categorical terms?

In order to make this more concrete, let us give a few examples of research from the literature that can be motivated using Question 1.2.1. An example for Question 1.2.1(i) is Longley's notion of an applicative morphism between PCAs. One can certainly provide a compelling motivation for this notion purely in terms of PCAs. Nevertheless, the strongest evidence that applicative morphisms are the 'correct' morphisms between PCAs is the fact that they correspond exactly to 'nice' functors between categories of assemblies and realizability toposes. Thus, the notion of an applicative morphism receives at least part of its motivation through the world of category theory. Another example is the notion of a *computationally dense* applicative morphism, which was introduced to characterize the geometric morphisms between realizability toposes. It should be noted that applicative morphisms and computational density turned out to have perfectly nice properties 'in isolation', that is, without reference to their categorical motivation. In Chapter 2 below, we will develop a rather rich theory of applicative morphisms and of computaional density *before* introducing Asm and RT in Chapter 3. An example for Question 1.2.1(ii) is the construction of the 'oracle PCA' $A[\alpha]$ we encountered at the end of Section 1.1.4. As we mentioned, the corresponding realizability topos $\mathsf{RT}(A[\alpha])$ is a subtopos of $\mathsf{RT}(A)$. Thus, $A[\alpha]$ corresponds to a certain Lawvere-Tierney topology on $\mathsf{RT}(A)$ - a familiar concept in topos theory.

Before we proceed to describe the main results of the thesis, we need to explain in a bit more detail how we will organize the world of partial combinatory algebras. In fact, we will introduce three distinct categories of PCAs, which all agree on their objects, but differ in their morphisms. In all three categories, the objects will be *relative ordered* PCAs.

Convention 1.2.2. From now on, when we say 'partial combinatory algebra' or 'PCA', we will always mean a *relative ordered* PCA. Accordingly, we denote the realizability topos over a (by default, relative) PCA A simply by $\mathsf{RT}(A)$ rather than $\mathsf{RT}(A, A^\#)$, and similarly for the category of assemblies. If we want to make it plain that A is not relative (that is, $A^\# = A$), we say that A is *absolute*. Similarly, if we want to discuss the case where A is not ordered (that is, carries the discrete order), then we say that A is *discrete*.

First, we introduce the category OPCA, which first occurs in [HvO03] and whose arrows are called *morphisms of PCAs*. Next, we introduce a category OPCA_T whose arrows are Longley's applicative morphisms. By restricting OPCA_T to the absolute discrete PCAs, one obtains Longley's original category PCA. Moreover, we introduce another category OPCA_D, whose arrows are *partial* applicative morphisms. The notion of a partial applicative morphism is specific to relative PCAs. Recall from Section 1.1.4 that an applicative morphism $A \to B$ can be viewed as a 'simulation' of A inside B. A *partial* applicative morphism $A \to B$ is also a simulation of A inside B, except that elements outside $A^\#$ may be omitted from the simulation.

Thus, we introduce three different notions of an arrow between PCAs, and we introduce them in ascending order of generality. That is, if A and B are PCAs, then there are inclusions $\mathsf{OPCA}(A, B) \hookrightarrow \mathsf{OPCA}_T(A, B) \hookrightarrow \mathsf{OPCA}_D(A, B)$, and this yields faithful functors $\mathsf{OPCA} \to \mathsf{OPCA}_T \to \mathsf{OPCA}_D$. Using the notion of a partial applicative morphism, we will be able to generalize Longley's results discussed in Section 1.1.4, to the relative setting. The following are equivalent:

- a *partial* applicative morphism $A \to B$;
- a regular functor $\mathsf{Asm}(A) \to \mathsf{Asm}(B)$ that commutes with the ∇-functors;
- a regular functor $\mathsf{RT}(A) \to \mathsf{RT}(B)$ that commutes with the ∇-functors.

In comparison with the list on page 12, we have replaced 'applicative morphism' by 'partial applicative morphism', and the second item has disappeared. In fact, the following are also equivalent:

- an applicative morphism $A \to B$;
- a regular functor $\mathsf{Asm}(A) \to \mathsf{Asm}(B)$ that commutes with the Γ-functors.

Thus, in the relative case the picture is a bit more refined than in the case of absolute PCAs.

Now we will list the main results of the thesis, with references to their precise formulations in the main text. We have classified the results of the thesis in three categories:

A. Positive and limitative results on (co)products (Sections 2.4 and 4.2).

B. Products and slices of realizability toposes (Sections 4.3 and 4.4).

C. Higher-type functionals (Chapter 5).

The results in categories A and B answer questions in the spirit of Question 1.2.1(i), while the results in category C concern Question 1.2.1(ii).

A. Positive and limitative results on (co)products

A very elementary construction on categories is taking the product of two categories. If the categories are toposes, then this yields a *co*product in the category of toposes and geometric morphisms. Thus, we can ask: are realizability toposes closed under such coproducts? It turns out that this is not necessarily the correct question to ask. As we mentioned in Section 1.1.4, every realizability topos carries a canonical inclusion of the topos Set of sets. Thus, a topos of the form $\mathsf{RT}(A) \times \mathsf{RT}(B)$ has a canonical inclusion of Set^2, rather than Set. Therefore, it seems more appropriate to take the coproduct *over* Set, that is, to take a pushout of the span $\mathsf{RT}(A) \hookleftarrow \mathsf{Set} \hookrightarrow \mathsf{RT}(B)$. For this kind of pushout, we will prove both a negative and a positive result. In order to obtain these results, we also need to consider products and coproducts 'downstairs', that is, in the various categories of PCAs. The following summarizes our results on (co)products of PCAs and pushouts of realizability toposes.

Theorem A.

(i) The category OPCA *has finite biproducts and small products.*

(ii) The category OPCA_T *has finite coproducts and a terminal object. On the other hand, two PCAs* never *have a product in* OPCA_T*, unless one of the PCAs is equivalent to the terminal object.*

(iii) The pushout, over Set*, of two realizability toposes* $\mathsf{RT}(A)$ *and* $\mathsf{RT}(B)$ *is* never *a realizability topos, unless one of* $\mathsf{RT}(A)$ *and* $\mathsf{RT}(B)$ *is equivalent to* Set.

(iv) Dense subtoposes of realizability toposes are closed under pushout over Set.

For the precise formulations of these results, we refer to: (i) Proposition 2.4.6, Corollary 2.4.10; (ii) Corollary 2.4.17, Theorem 2.4.21; (iii) Theorem 4.2.4; (iv) Theorem 4.2.5.

B. Products and slices of realizability toposes

We can also approach the matter of taking products of realizability toposes differently, and ask whether $\mathsf{RT}(A) \times \mathsf{RT}(B)$ really *is* a realizability topos, but constructed over the 'base topos' Set^2 rather than Set. In this way, we can address the construction of products of realizability toposes directly, by allowing the 'base' to vary.

Another important construction in topos theory is that of taking *slices* of toposes. Thus, we can ask: are slice toposes of realizability toposes again realizability toposes? It turns out that, here as well, we need to allow the base to vary: if $I \in \mathsf{RT}(A)$, then $\mathsf{RT}(A)/I$ carries a canonical inclusion of $\mathsf{Set}/\Gamma I$, rather than Set. We will address both products and slices of realizability toposes by considering a notion of PCA internal to a general base category.

The main ideas of this general notion of PCA are taken from [Ste13]. Given a regular[6] category $\mathcal{C}$, an *internal PCA* (or IPCA) over $\mathcal{C}$ is a pair (A, ϕ). Here A is an object of $\mathcal{C}$ with certain properties, and ϕ, which is called an *external filter* on A, is a set of 'realizing objects'. For more precise definitions, we refer to Section 4.3.1. As explained in [Ste13], there is a notion of applicative morphism between IPCAs over a given base $\mathcal{C}$, which yields a category $\mathsf{IPCA}_{\mathcal{C}}$. Moreover, we introduce a notion of 'base change' between different bases, thereby obtaining a larger category IPCA of IPCAs over (arbitrary) regular base categories.

If (A, ϕ) is an IPCA, then we can construct a category $\mathsf{Asm}(A, \phi)$ of assemblies. In [Ste13, Corollary 2.2.18], it is shown that such categories are closed under slicing. We will prove a number of additional results concerning categories of the form $\mathsf{Asm}(A, \phi)$; see (i) and (iii) below. Moreover, if the base category of an IPCA (A, ϕ) is a topos, then we may also construct a realizability topos $\mathsf{RT}(A, \phi)$. The following summarizes our results on products and slices of categories of assemblies and realizability toposes over IPCAs.

Theorem B.

(i) Categories of assemblies over IPCAs are closed under small products. Moreover, the underlying IPCA of $\prod_i \mathsf{Asm}(A_i, \phi_i)$ *is the product, in* IPCA*, of the* (A_i, ϕ_i)*.*

(ii) Realizability toposes over IPCAs are closed under small products (as categories).

(iii) If $I \in \mathsf{Asm}(A, \phi)$*, then the underlying IPCA of* $\mathsf{Asm}(A, \phi)/I$ *can be constructed explicitly using base change and taking a finite extension of the external filter.*

(iv) Realizability toposes over IPCAs are closed under slicing over assemblies.

For the precise formulations of these results, we refer to: (i) Proposition 4.4.1, Theorem 4.4.3; (ii) Corollary 4.4.33; (iii) Theorem 4.4.9; (iv) Corollary 4.4.37.

C. Higher-type functionals

At the beginning of this section, we mentioned the construction of the 'oracle PCA' $A[\alpha]$ for a partial function α on a PCA A. In [FvO16], this construction is generalized to type-2 functionals, which take functions on A as inputs. In this thesis, we will prove several results concerning type-3 functionals, i.e., (partial) functions that take type-2 functionals as inputs. The main strategy for obtaining such results is to view a type-3 functional on A as a type-2 functional on a related PCA $\mathcal{B}A$, whose elements are partial functions on A. This PCA $\mathcal{B}A$ was first introduced in [vO11], as an absolute discrete PCA. For our purposes, we need to view $\mathcal{B}A$ as a relative ordered PCA. Moreover, we need to generalize the constructions from [vO06] and [FvO16] to relative ordered PCAs. The

[6] It should be noted that [Ste13] only allows *Heyting* categories as base categories, so our setup is more general.

following summarizes our results on the computability of higher-type functionals. The first three items generalize existing constructions from the literature to the relative ordered setting. The final two items are new results on type-3 functionals.

Theorem C.
Let A be a (by default, relative and ordered) PCA.

(i) (Cf. [vO11, Section 5]) There is a PCA $\mathcal{B}A$ whose elements are partial functions on A, such that there exists a local geometric morphism $\mathsf{RT}(\mathcal{B}A) \twoheadrightarrow \mathsf{RT}(A)$.

(ii) (Cf. [vO06, Theorem 2.2]) For each $\alpha \in \mathcal{B}A$, there is a PCA $A[\alpha]$ which is the 'universal solution' to making α computable.

(iii) (Cf. [FvO16, Theorem 3.1]) If the PCA A satisfies a condition called chain-completeness *(Definition 5.3.1) and F is a type-2 functional on A, then there is a PCA $A[F]$ which is the 'universal solution' to making F computable.*

(iv) If A is chain-complete and Φ is a type-3 functional on A, then there is a PCA $A[\Phi]$, which is a 'lax universal solution' to making Φ computable.

(v) Suppose that A is discrete and that Φ is a type-3 functional on A. Then there exists an $\alpha \in \mathcal{B}A$ with the following property: the set of partial functions on A that are forced to be computable if Φ is computable, can be described as the set of partial functions on A that are computable using an oracle for α.

For the precise formulations of these results, we refer to: (i) Proposition 5.2.7, Proposition 5.2.13; (ii) Theorem 5.1.17; (iii) Theorem 5.3.13; (iv) Theorem 5.4.2; (v) Corollary 5.4.8.

1.3 Scope and limitations

As announced in Convention 1.2.2, PCAs will always be relative and ordered, unless stated otherwise. There are several reasons for this choice, which we explain here.

First of all, the notion of a relative ordered PCA seems to be exceedingly natural - more natural, in fact, than the notion of an absolute discrete PCA. We already mentioned one piece of evidence for this view: the paper [Hof06] isolates relative ordered PCAs as a natural class of objects that give rise to triposes in a canonical way. Another manifestation of the 'naturality' of ordered PCAs is the fact that the categories OPCA_T and OPCA_D may be obtained as the Kleisli categories for monads T and D on the category OPCA. These monads can only be defined on the category of *ordered* PCAs, because TA and DA are not discrete PCAs. The definitions of OPCA_T and OPCA_D as such Kleisli categories has many conceptual advantages. For example, many results about (partial) applicative morphisms, that is, about the categories OPCA_T and OPCA_D, are inherited more or less directly from OPCA; see Section 2.3.2 below. This allows a very elegant treatment of the theory of (partial) applicative morphisms, since the category OPCA is easier to work with. Moreover, while (partial) applicative morphisms only allow us to characterize *regular* functors between categories of assemblies (see Corollary 3.3.14 below), the monads T and D can be used to, additionally, characterize *left exact* functors between categories of assemblies (see Theorem 3.3.13 below). This is useful for characterizing geometric morphisms between categories of assemblies, since the direct image of such a geometric morphism, while left exact, is not necessarily regular. Finally, let us direct some attention specifically to the monad D. As we explained in Section 1.2, the notion of a *partial* applicative morphism is only useful in the setting of relative PCAs. Since partial applicative morphisms are arrows of $\mathsf{OPCA}_D = \mathrm{Kl}(D)$, this means that the monad D should also be specific to the relative setting. Indeed, if we view DA as an absolute PCA, then, since DA has a least element, it would always be equivalent to the terminal PCA, making the monad D trivial. On the other hand, in the relative setting, the monad D is quite useful. If A is a PCA, then its realizability tripos is canonically represented by DA. By making DA into a PCA, we can obtain a correspondence between tripos maps and arrows of OPCA; see Proposition 3.3.16 below.

Second, the machinery of relative ordered PCAs is necessary to obtain several of the main results of the thesis, even if one is only interested in these results for absolute discrete PCAs. Consider, for example, Theorem A: items (iii) and (iv) also hold for discrete PCAs. (For item (iii), this follows immediately, and for item (iv), this follows easily by inspecting the proof of Theorem 4.2.5.) Nevertheless, the proofs of (iii) and (iv) use Theorem A(i) and (ii), which mention OPCA, and thus depend on the notion of an ordered PCA. So, even if one is only interested in discrete PCAs, one needs the theory of ordered PCAs to obtain Theorem A(iii) and (iv). For the other main results of the thesis, similar remarks apply. For IPCAs (A, ϕ), there is also a notion of 'absolute' IPCA: (A, ϕ) is absolute iff the external filter ϕ is as large as possible (see Definition 4.3.3). While categories of assemblies over IPCAs are closed under slicing, this is *not* true for categories of assemblies over absolute IPCAs. Thus, even if one is initially only interested in absolute PCAs, the construction of slicing forces one to consider relative PCAs as well. Finally, the results in Theorem C depend crucially on the fact that we view $\mathcal{B}A$ as a relative ordered PCA, even when A is absolute and discrete. Even the proof of item (v), which only holds for discrete PCAs, uses the theory of relative ordered PCAs in an indispensable way.

Third and last, it seems useful to have a treatment of PCAs and their corresponding realizability toposes in this general setting of relative ordered PCAs, since such a treatment does not seem to be available in the literature. The monograph [vO08] also treats the theory of PCAs and realizability toposes very extensively, but primarily for absolute discrete PCAs. The paper [FvO14] proves a number of new results on functors between realizability toposes over ordered PCAs, but it does not treat relative PCAs. As we have seen above, incorporating relative PCAs requires some adjustments to Longley's results. This thesis is the first source to explain the full picture of Longley's results in the setting of relative PCAs.

There is another possible generalization of PCAs that we do not discuss here at all, namely that of a *typed* PCA. The idea behind typed PCAs is as follows. As we have seen, the elements of a PCA act simultaneously as (codes of) algorithms and as inputs to such algorithms. In a typed PCA A, we specify a set T of types, equipped with a binary operation $\to$, and for each $\sigma \in T$, a set $A(\sigma)$. The elements of $A(\sigma \to \tau)$ then represent algorithms taking inputs from $A(\sigma)$ and yielding outputs in $A(\tau)$. More explicitly, for each pair of types $\sigma, \tau \in T$, we have a partial application map $A(\sigma \to \tau) \times A(\sigma) \rightharpoonup A(\tau)$. Every untyped PCA can also be viewed as a typed PCA, by taking T to be a singleton.

There are two reasons for not considering typed PCAs in this thesis. First, as we will see in Section 2.1.4 below, an untyped PCA is automatically equipped with a lot of 'computational' structure: pairing, definition by cases, recursion, fixpoints, etc. For typed PCAs, this structure is no longer automatic, but has to be imposed as an additional requirement. Second, for typed PCAs, the notion of a realizability topos is no longer available. One can define a category of assemblies $\mathsf{Asm}(A)$ for a typed PCA A, but $\mathsf{Asm}(A)_{\mathrm{ex/reg}}$ is not necessarily a topos. In fact, [LS02, Theorem 4.2] shows that $\mathsf{Asm}(A)_{\mathrm{ex/reg}}$ is a topos if and only if A is 'essentially untyped', meaning that A is equivalent to an untyped PCA.

On the other hand, typed PCAs seem to be the natural home of *higher-order* computation, which is why the authors of [LN15] choose typed PCAs as their primary abstract model of computation. As we shall see in Chapter 5 below, studying higher-type functionals in the untyped setting presents serious obstacles. One may view Chapter 5 as an investigation into how far one can go with higher-order computation in the untyped setting.

Regarding Question 1.2.1(i) above, we are mainly interested in slices and coproducts (over Set or outright) of realizability toposes. There are certainly other constructions that can be studied here; we offer one suggestion below. We should also mention that, while there seems to be a rich theory of subtoposes of realizability toposes, and in particular of the effective topos ([LvO13], [Kih21]), subtoposes of realizability toposes play only a minor role in this thesis.

Finally, let us offer two suggestions for further research. First of all, it is known that geometric morphisms between realizability toposes are always localic. This is shown, for absolute discrete PCAs, in [Joh13, Lemma 2.4], but the proof carries over to the relative ordered case. In particular, a geometric morphism $f\colon \mathsf{RT}(A) \to \mathsf{RT}(B)$ is always bounded, which means that pullbacks along f, in the category of toposes and geometric morphisms, always exist. The literature does not contain many results on such pullbacks. Second, while Chapter 5 offers several new results on computing with type-3 functionals, the case of functionals of type 4 and higher remains entirely mysterious.

1.4 Overview of the thesis

Here we offer a brief, linear overview of the contents of the thesis.

In Chapter 2, we treat the theory of PCAs and the various kinds of morphism between them. First, in Section 2.1, we define PCAs, describe some elementary constructions that can be performed in any PCA, and give a few examples. In Section 2.2, we introduce the category OPCA, consider some special properties that arrows of OPCA can have, and give some examples of arrows of OPCA. In Section 2.3, we introduce the Kleisli categories OPCA_T and OPCA_D, yielding the notion of a (partial) applicative morphism. Again, we discuss some special properties and examples of (partial) applicative morphisms. Moreover, we discuss the existence of right adjoints in the three categories of PCAs. Finally, in Section 2.4, we establish Theorem A(i) and (ii).

In Chapter 3, we treat the categories associated to a PCA A: the category of assemblies $\mathsf{Asm}(A)$ and the realizability topos $\mathsf{RT}(A)$. We define these in Section 3.1, and discuss some important properties in Section 3.2. In Section 3.3, we generalize the results from [Lon94] and [FvO14] to the relative setting, characterizing various kinds of functors between categories of assemblies and realizability toposes. Finally, in Section 3.4, we discuss geometric morphisms between categories of assemblies and realizability toposes.

In Chapter 4, we discuss categorical constructions on categories of the form $\mathsf{Asm}(A)$ or $\mathsf{RT}(A)$. Section 4.1 describes various difficulties that arise when one wishes to take products or slices of realizability toposes. In Section 4.2, we establish Theorem A(iii) and (iv). The framework of IPCAs is introduced in Section 4.3, and subsequently, in Section 4.4, we establish the statements in Theorem B.

Chapter 5 concerns computation with oracles and higher-type functionals, and contains all statements from Theorem C. First, in Section 5.1, we generalize the construction of the 'oracle PCA' from [vO06] to the relative ordered setting. Section 5.2 introduces $\mathcal{B}A$ as a relative ordered PCA, and establishes some of its important properties. In Section 5.3, we introduce chain-completeness and we treat type-2 functionals. Finally, Section 5.4 contains our results on type-3 functionals.

1.5 Conventions and notation

In this final section of the introduction, we set forth some conventions and notation that will be used throughout the thesis.

Possibly undefined expressions. Since partial combinatory algebras carry a *partial* application map, we will frequently have to deal with expressions that are possibly undefined. We adopt the convention that such an expression can only be defined if all its subexpressions are defined as well. This will be especially relevant in Construction 2.1.28. If e is a possibly undefined expression, then we write $e\downarrow$ to indicate that e is in fact defined. Moreover, if e and e'

are two possibly undefined expressions, then we write $e \simeq e'$ for the following statement: $e\downarrow$ iff $e'\downarrow$, and in the case that e and e' are both defined, they denote the same value. This relation is known as **Kleene equality** between possibly undefined expressions. On the other hand, we reserve $e = e'$ for the stronger statement that e and e' are defined, and equal to each other. Thus, one might say that we use the equality sign as an 'existence-entailing relation'.

Arrows. By default, we use the usual arrow sign $\to$ to denote functions between sets, and arrows in categories. There will be a few expections to this rule. If we want to emphasize that a function is *partial*, then we will use the harpoon sign $\rightharpoonup$. Since we will discuss three categories whose objects are PCAs, it will be advantageous to distinguish the three kinds of morphism between PCAs notationally. If A and B are PCAs, then the usual arrow sign $A \to B$ will signify a morphism in OPCA. On the other hand, an applicative morphism from A to B will be denoted by $A \multimap B$. We choose the lollipop arrow here because an applicative morphism can be viewed as a multi-valued function. A *partial* applicative morphism from A to B will be denoted by $A \leftharpoonup\!\!\circ\; B$. Finally, we may use the hooked arrow symbol $\hookrightarrow$ if we want to emphasize that an arrow is, depending on the situation, injective, mono, or an inclusion. Similarly, we use the two-headed arrow symbol $\twoheadrightarrow$ to indicate that an arrow is surjective, epi, or a quotient.

Special categories. Above, we have already mentioned left exact, regular, and exact categories. For the sake of definiteness, let us explicitly define these terms. A left exact category is a finitely complete category, that is, a category with all finite limits. A left exact functor is a functor that preserves all finite limits. A regular category is a left exact category that has pullback-stable regular epi-mono factorizations, and a regular functor is a left exact functor that preserves regular epimorphisms. By an exact category, we mean a category which is exact in the sense of Barr. An exact category can be defined as a regular category in which every internal equivalence relation is a kernel pair.

A cartesian closed category is a category with finite products and exponentials. A *locally* cartesian closed category is a category with a terminal object, such that all its slice categories are cartesian closed. Note that a locally cartesian closed category is, in particular, cartesian closed; there are versions of the definition for which this is not true.

Finally, by a topos we will always mean an *elementary* topos, that is, a left exact, cartesian closed category which has a subobject classifier.

2-categorical terminology. We will have to deal with various 2-dimensional categories, that is, categories that have 2-cells in addition to having objects and arrows. Since the terminology regarding 2-dimensional categories is not consistent throughout the literature, we treat our use of the terminology here. First of all, by a 2-category, we mean a 2-dimensional category which is also an ordinary (1-)category. That is, the unit and associativity laws for 1-cells (i.e., arrows) should hold on the nose. Modulo size constraints, a 2-category is a Cat-enriched category, where Cat denotes the category of small categories and functors. A preorder-enriched category is a category enriched over the category of preorders. Alternatively, it is a (locally small) 2-category such that there exists at most one 2-cell between any given pair of 1-cells. A bicategory is a 2-dimensional category in which the axioms for a 1-category need only hold up to invertible 2-cells. Of course, these 2-cells need to be coherent in the appropriate sense, but we do not spell this out. Usually, we need not worry about this, because our homcategories are preorders. We use the term 'preorder-enriched bicategory', which strictly speaking does not make sense, to emphasize that we have a bicategory whose homcategories are preorders.

If we say that F is a pseudofunctor between 2-dimensional categories, then we mean that F must preserve identity 1-cells and composition of 1-cells up to invertible 2-cells. If these invertible 2-cells are identities, that is, F preserves identity 1-cells and composition of 1-cells on the nose, then we say that F is a 2-functor. Note that it is possible to have a 2-functor between proper bicategories; we will meet examples in Chapters 3 and 4 below. If we drop 'invertible' from the definition of a pseudofunctor, then we obtain the notion of a lax or oplax functor, depending on the direction in which the 2-cells go. For natural transformations, we adopt similar definitions. Thus, a 2-natural transformation is natural on the nose, a pseudonatural transformation is natural up to invertible 2-cells, and a(n op)lax natural transformation merely has 2-cells filling the naturality squares. A pseudomonad will be a triple (T, η, μ), where T is a pseudofunctor, and η and μ are pseudonatural transformations satisfying the monad laws up to invertible modification. By an algebra for a pseudomonad, we really mean a *pseudo*algebra, i.e., the algebra laws need only hold up to invertible 2-cells.

Similarly, by a pseudolimit, we mean to employ a 'fully weak' notion of limit. That is, cones only need to commute up to invertible 2-cells, and the universal property of a pseudolimit is expressed by an equivalence of categories. For example, a pseudoproduct of X and Y in a 2-dimensional category $\mathcal{C}$ is a span $X \xleftarrow{\pi_0} X \times Y \xrightarrow{\pi_1} Y$ such that for each object Z,

$$\mathcal{C}(Z, X \times Y) \xrightarrow{(\pi_0 \circ -, \pi_1 \circ -)} \mathcal{C}(Z, X) \times \mathcal{C}(Z, Y)$$

is an equivalence of categories. In the case of pseudoproduct, we need of course not worry about whether cones commute on the nose or not. But note that for, e.g., pseudopullbacks, we need to specifiy *three* projection arrows, rather than the usual two. In contrast, for 2-limits, cones must commute on the nose, and the universal property must be expressed by an *isomorphism* of categories. For colimits, we adopt similar conventions. Moreover, we say that a pseudoinitial object 0 is *strict* if every arrow $X \to 0$ is an equivalence; the notion of a strict pseudoterminal object is defined dually.

Finally, we say that an arrow $f\colon X \to Y$ is a pseudomono if for each object Z, the map $f \circ -\colon \mathcal{C}(Z, X) \to \mathcal{C}(Z, Y)$ is an equivalence of categories. If it is always an isomorphism, then we may say that f is a 2-mono. For epis, we adopt the dual convention.

The Axiom of Choice. In Chapters 2, 3 and 5, the category Set of sets will serve as our 'base category'. We will assume freely that the Axiom of Choice holds in Set, but we will indicate the occasions where we actually use it.

Smallness. In general, we will not worry too much about size issues. At several occasions, however, it is better, for the sake of sanity, to make some smallness assumptions. When discussing the internal logic of a category $\mathcal{C}$, we will implicitly assume that $\mathcal{C}$ is well-powered. Since, in such a situation, $\mathcal{C}$ will have finite products, this also implies that $\mathcal{C}$ is locally small. As a result, in the internal language of $\mathcal{C}$, there will be only a set of relation symbols of a given type, and only a set of arrow symbols of any given type. Similarly, when discussing triposes over a base category $\mathcal{C}$, we implicitly assume that $\mathcal{C}$ is locally small.

Typographical conventions. On page 211, the reader will find a comprehensive index of all notation used in the thesis. Here, we give some general maxims that we followed in choosing notation.

- Elements of a PCA A are typically denoted by a, b, c, or a, a', a''. If we are dealing with elements from $A^{\#}$ specifically, we use r, s. For variables, we use x, y, z.

- The combinators associated to each PCA are denoted in lower case sans serif math font: $\mathsf{k}, \mathsf{s}, \mathsf{i}$, etc. In Chapter 4, the 'combinators' will be certain objects, rather than elements. Here we will use upper case: K, S, I, etc.

- A general category will be denoted in calligraphic math font, e.g., 'let $\mathcal{C}$ be a regular category'. In contrast, specific categories, or a specific class of categories, are denoted in sans serif font, e.g., Set, $\mathsf{Asm}(A)$, $\mathsf{RT}(A)$, LEX, REG, EX. The only exception to this rule is the effective topos, which is traditionally denoted in calligraphic font by $\mathcal{E}ff$. Functors are denoted by upper case Roman letters F, G, and natural transformations are denoted by lower case Greek letters μ, ν.

- Given a PCA A, 'higher-order' objects on A will be denoted by lower case Greek letters α, β, γ. By such higher-order objects, we mean: subsets of A, (partial) functions on A, or higher-type functionals on A. If we have a function whose inputs are such higher-order objects, then we may use upper case Roman letters F, G. If we want to go one level higher, we use upper case Greek letters Φ, Ψ.

- Arrows between PCAs (of any of the three kinds) and arrows of $\mathsf{Asm}(A)$ are usually denoted by lower case Roman letters f, g, h, but we will also sometimes use lower case Greek letters.

- Formulas in a formal language are denoted by lower case Greek letters φ, ψ, χ.

Chapter 2

Partial Combinatory Algebras

In the first three sections of this chapter, we treat the theory of PCAs (Section 2.1) and various categories having PCAs as objects (Section 2.2 and Section 2.3). These sections contain no essentially new material, except for two new notions needed to deal with *relative* PCAs.

1. We introduce the notion of a *partial* applicative morphism, which first occurs in [Zoe21b]. A traditional applicative morphism f from A to B assigns to each $a \in A$ a *nonempty* downset $f(a)$ of B. For partial applicative morphisms, $f(a)$ is allowed to be empty, but only for $a \in A$ that lie outside $A^{\#}$. We should mention that the notion of a partial applicative morphism is implicit in [Ste13] (e.g., Definition 2.3.20 and Definition 2.4.22).

2. The paper [HvO03] introduces a notion of *computationally dense* applicative morphism. This notion can be used to characterize left adjoints in the category of PCAs and applicative morphisms. In our case, we have another category of PCAs, where the morphisms are *partial* applicative morphisms, and accordingly, we will need two notions to characterize left adjoints in these categories. We relativize the notion of computational density from [HvO03], which yields a characterization of left adjoints in the category of partial applicative morphisms. In addition, we introduce the stronger notion of *density*, which characterizes left adjoints in the category of applicative morphisms. The relevant results are Theorem 2.3.14 and Corollary 2.3.15 below.

Finally, Section 2.4 discusses products and coproducts in the various categories of PCAs; this material is from [Zoe21a].

2.1 Definition of partial combinatory algebras

Let us recall that by a 'partial combinatory algebra', we always mean a *relative ordered* PCA. This means that a PCA will be a set A equipped with three pieces of structure: a partial binary operation, a partial order, and a privileged subset $A^{\#}$ (which will be called a *filter*). The intuition behind these is as follows.

- The elements of A simultaneously play the role of (codes or Gödel numbers of) computations, and of inputs that may be fed to these computations. For $a, b \in A$, we think of the image of the pair (a, b) under the partial binary operation as the result, if any, when the computation a is applied to the input b.
- For $a, a' \in A$, we think of $a' \leq a$ as saying that a' is a refinement or specialization of a, or that a' gives more information than a.
- As we mentioned in the introduction, the elements in $A^{\#}$ can be thought of as the computations that can actually be implemented, or carried out. Alternatively, one may think of the elements of $A^{\#}$ as computable data, and of the elements outside $A^{\#}$ as non-computable data. In the ordered context, however, it is more appropriate to say that the elements of $A^{\#}$ may be *refined* to a computable element. Accordingly, the set $A^{\#}$ will be upwards closed with respect to the partial order.

Of course, this structure will have to satisfy certain axioms that capture the intuition that the elements of A represent computations. Below, we will define PCAs in detail, establish some basic properties, and finally, give a few examples.

2.1.1 Partial applicative posets

In this section, we introduce the notion of a partial applicative poset. While this notion does not really occur in the literature, we believe it is useful to introduce because it is the minimal structure needed to define *filters*, which will play a central role in the theory developed below.

Definition 2.1.1. *A* ***partial applicative poset*** *(abbreviated PAP) is a triple* $A = (A, \cdot, \leq)$ *where* $(A, \leq)$ *is a poset and* $\cdot$ *is a partial binary map* $A \times A \rightharpoonup A$*, called the* ***application map****, such that:*

(A) the application map has downwards closed domain and preserves the order, that is, if $a' \leq a$*,* $b' \leq b$ *and* $a \cdot b$ *is defined, then* $a' \cdot b'$ *is defined as well, and* $a' \cdot b' \leq a \cdot b$*.*

We say that A *is* ***total*** *if the application map is a total function, and* ***discrete*** *if the order* $\leq$ *is the discrete order.*

Axiom (A) fits the intuition that $a' \leq a$ expresses that a' contains more information than a: if a' and b' contain more information than a and b, and $a \cdot b$ is already defined, then $a' \cdot b'$ should also be defined and contain at least as much information.

Convention 2.1.2. When working with PAPs, we adopt the following two conventions.

(i) When this creates no confusion, we omit the dot for application, writing ab instead of $a \cdot b$.

(ii) The application map will not, in general, be associative. Therefore, we adopt the convention that application associates to the left, writing abc as an abbreviation for $(ab)c$. Thus, one can read abc as: first apply a to b, then apply the result to c.

In the introduction, we already introduced Kleene equality for dealing with possibly undefined expressions. In combination with order on A, the following notation will be exceedingly useful.

Notation 2.1.3. Let A be a PAP, and let e and e' be possibly undefined expressions that, if defined, assume values in A. We write $e' \preceq e$ for the following statement: if $e\downarrow$, then $e'\downarrow$ and $e' \leq e$. We call this relation **Kleene inequality**.

As for equality, we reserve the statement $e' \leq e$ for the case where e' and e are actually defined, and satisfy $e' \leq e$. Note that the Kleene equality $e \simeq e'$ is equivalent to: $e \preceq e'$ and $e' \preceq e$. Using Kleene inequality, we can define a partial order on the set of n-ary partial functions on A, as follows.

Definition 2.1.4. *Let A be a PAP and let $\alpha, \beta \colon A^n \rightharpoonup A$ be partial functions. We write $\alpha \leq \beta$ if, for all $\vec{a} \in A^n$, we have $\alpha(\vec{a}) \preceq \beta(\vec{a})$.*

It is not hard to check that this indeed defines a partial order on the set of partial functions $A^n \rightharpoonup A$. The case $n = 1$ will be especially interesting: in Chapter 5 we will see that, if A is a PCA, then a certain subset of partial functions $A \rightharpoonup A$ can be made into a PCA as well. The order on this PCA will be the order defined above.

Remark 2.1.5. We should warn the reader that various sources treating only discrete PCAs adopt the 'opposite' of Notation 2.1.3, writing $e \preceq e'$ where we write $e' \preceq e$. The reason for this is that, in the discrete case, the order in Definition 2.1.4 matches the *reverse* subfunction relation, i.e., $\alpha \leq \beta$ iff β is a subfunction of α. By reversing Notation 2.1.3, the order on partial functions $A \rightharpoonup A$ matches the subfunction relation. In the ordered setting, however, Notation 2.1.3 is really the right one to adopt. Indeed, in the case where all expressions are defined, we would like $e' \preceq e$ to imply $e' \leq e$, and not $e \leq e'$, which would be highly confusing!

In Section 2.1.5 below, we will see many examples of PCAs, so in particular, of PAPs. Here we treat two more elementary examples.

Example 2.1.6. If $(A, \leq)$ is a poset with finite meets, then it can be made into a total PAP by setting $ab = a \wedge b$.

Example 2.1.7. Let A be a PAP. We write TA for the set of *nonempty, downwards closed* subsets of A, i.e., the set of all nonempty $\alpha \subseteq A$ satisfying: if $a' \leq a$ and $a \in \alpha$, then $a' \in \alpha$ as well. We define an application map on TA as follows: if $\alpha, \beta \in TA$, then we say that $\alpha\beta\downarrow$ if and only if $ab\downarrow$ for all $a \in \alpha$ and $b \in \beta$. In this case, $\alpha\beta$ is defined as $\downarrow\{ab \mid a \in \alpha, b \in \beta\}$, i.e., the downwards closure of $\{ab \mid a \in \alpha, b \in \beta\}$. This makes $(TA, \cdot, \subseteq)$ into a PAP, which we will simply denote by TA. As we shall see later, T yields a monad structure on one of our categories of PCAs, called the *nonempty downset monad*.

Similarly, we have the poset DA consisting of *all* downsets of A, including $\emptyset$, again ordered by inclusion. We can define an application on DA similar to the one defined on TA, yielding a PAP $DA = (DA, \cdot, \subseteq)$.

2.1.2 Filters and partial applicative structures

In this section, we define partial applicative structures, which will be PAPs equipped with a privileged subset $A^{\#}$. This subset should satisfy two conditions, that we encapsulate in the definition of a filter.

Definition 2.1.8. *Let A be a PAP. A **filter** on A is a subset $F \subseteq A$ which is:*

(i) closed under defined application, that is, if $a, b \in F$ and $ab\downarrow$, then $ab \in F$ as well;

(ii) upwards closed, that is, if $a \leq b$ and $a \in F$, then $b \in F$ as well.

Example 2.1.9. If $(A, \leq)$ is a poset with finite meets, then a nonempty filter on $(A, \wedge, \leq)$ is a filter on $(A, \leq)$ in the usual sense.

Example 2.1.10. (i) If A is a PAP, and F is a filter on A, then F can also be made into a PAP, by restricting both the application map and the order to F. This new PAP will be denoted by $(F, \cdot, \leq)$, or simply by F.

(ii) If A is a PAP, F is a filter on A, and G is a filter on the PAP F, then G is also a filter on A.

Example 2.1.11. Let A be a PAP.

(i) The set TA is a filter on the PAP DA.

(ii) Let F be a filter on A. Then the set

$$\{\alpha \in TA \mid \alpha \cap F \in TA\} = \{\alpha \in TA \mid \exists \beta \in TF\, (\beta \subseteq \alpha)\} = \uparrow(TF) \subseteq TA$$

is a filter on TA. By (i) and Example 2.1.10(ii), this set will also be a filter on DA, and in DA this filter can also be described as the upset of TF. Observe that the filter TA of DA can be retrieved as a special case of this, by taking $F = A$.

Since a filter is defined as a set with certain closure properties, we can consider the notion of a generated filter, which will play an important role in this thesis.

Definition 2.1.12. *Let A be a PAP and let A_0 be a subset of A. We define $\langle A_0 \rangle$ as the smallest filter on A extending A_0, and we call this the **filter generated by** A_0.*

In the case of filters on posets with finite meets, one can always generate a filter by first taking all possible finite meets, and then closing upwards. In the current case, a similar description of generated filters is available. Before we can formulate it, we need the notion of a term, which will also be central to the treatment of PCAs in the next section.

Definition 2.1.13. *Let A be a PAP. The set of **terms** over A is defined recursively as follows:*

(i) We assume given a countably infinite set of distinct variables, and these are all terms.

(ii) For every $a \in A$, we assume that we have a constant symbol for a, and this is a term. The constant symbol for a is simply denoted by a.

(iii) If t_0 and t_1 are terms, then so is $(t_0 \cdot t_1)$.

*We say that a term is **pure** if it contains no constant symbols.*

When dealing with terms, we will omit the dot and brackets as much as possible, subject to the same conventions as in Convention 2.1.2. If $t = t(\vec{x})$ is a term whose variables are among the sequence $\vec{x}$, then we can assign an obvious, possibly undefined, interpretation $t(\vec{a}) \in A$ to an input sequence $\vec{a} \in A$. In this way, every term $t(\vec{x})$ yields a partial function $\lambda\vec{a}.t(\vec{a})\colon A^n \rightharpoonup A$, where n is the length of the sequence $\vec{x}$.

We have the following alternative descriptions of generated filters; the proof is easy and omitted.

Lemma 2.1.14. *Let A be a PAP and $A_0 \subseteq A$. Then*

$$\langle A_0 \rangle = \uparrow\{t(\vec{a}) \mid t(\vec{x}) \text{ a pure term}, \vec{a} \in A_0 \text{ and } t(\vec{a})\downarrow\}.$$

A partial applicative structure is just a partial applicative poset equipped with a filter.

Definition 2.1.15. *A **partial applicative structure** (abbreviated PAS) is a quadruple $A = (A, A^\#, \cdot, \leq)$, where $(A, \cdot, \leq)$ is a PAP, that is to say:*

(A) the application map has downwards closed domain and preserves the order;

and $A^\#$ is a filter on $(A, \cdot, \leq)$, that is to say:

(B) $A^\#$ is closed under defined application;

(C) $A^\#$ is upwards closed.

*The PAS A is called **absolute** if $A^\# = A$.*

We will say that a PAS is total iff its underlying PAP is, and similarly for discreteness. However, for filters on a PAS, we use the following definition.

Definition 2.1.16. *Let A be a PAS. A* ***filter*** *on A will be a filter F on the PAP $(A, \cdot, \leq)$ satisfying $A^{\#} \subseteq F$.*

In other words, a filter on a PAS A is a filter on the underlying PAP that extends $A^{\#}$.

Example 2.1.17. Let A be a PAS and let F be a filter on A. Then $(F, A^{\#}, \cdot, \leq)$ is also a PAS. When no confusion can arise, we will denote this PAS simply by F. In particular, we have the absolute PAS $A^{\#} = (A^{\#}, A^{\#}, \cdot, \leq)$. Of course, $(A, F, \cdot, \leq)$ is also a PAS. In particular, we have the absolute PAS $A_{\text{abs}} = (A, A, \cdot, \leq)$.

Example 2.1.18. If A is a PAP, then the PAP TA (Example 2.1.7) can be made into a PAS by setting

$$(TA)^{\#} = \uparrow(TA^{\#}) = \{\alpha \in TA \mid \alpha \cap A^{\#} \in TA\}.$$

Similarly, DA can be made into a PAS by $(DA)^{\#} = (TA)^{\#}$.

2.1.3 PCAs and combinatory completeness

The partial applicative structures introduced in the previous section are rather 'algebraic' and do not yet capture the intended intuition of PCAs as models of computation. In order to capture this intuition, we impose some further axioms on partial applicative structures.

Definition 2.1.19. *A* ***partial combinatory algebra*** *(abbreviated PCA) is a PAS $A = (A, A^{\#}, \cdot, \leq)$, that is to say:*

(A) the application map has downwards closed domain and preserves the order;

(B) $A^{\#}$ is closed under defined application;

(C) $A^{\#}$ is upwards closed;

for which there exist $\mathsf{k}, \mathsf{s} \in A^{\#}$ such that:

(D) $\mathsf{k}ab \leq a$;

(E) $\mathsf{s}ab\downarrow$;

(F) $\mathsf{s}abc \preceq ac(bc)$,

for all $a, b, c \in A$.

Of course, we will say that a PCA is total iff it is a total PAS, and similarly for discreteness and absoluteness. A filter on a PCA will simply be a filter on the PAS.

Remark 2.1.20. (i) Let us warn the reader that k and s are not taken to be part of the structure of the PCA A; they are merely required to exist.

(ii) In axiom (F), we require that $\mathsf{s}abc$ is defined if $ac(bc)$ is defined, but not conversely. Some sources (e.g., [vO08]) give a definition of discrete PCAs that require $\mathsf{s}abc$ to be defined *exactly* when $ac(bc)$ is defined; we may call such a discrete PCA 'strict'. In [FvO16], it is shown that there is no essential difference between these two definitions of discrete PCAs (Theorem 5.1). For ordered PCAs, the version of axiom (F) given here is standard.

In Section 2.1.5 below, we will give many interesting examples of PCAs. For now, let us note a few elementary examples.

Example 2.1.21. Any PAS $(A, A^{\#}, \wedge, \leq)$, where $(A, \leq)$ is a poset with finite meets, is automatically a PCA, provided $A^{\#} \neq \emptyset$. Indeed, any element of $A^{\#}$ can serve as both k and s.

Example 2.1.22. If A is a PCA, then so are TA and DA (Example 2.1.18). In both cases, $\downarrow\{\mathsf{k}\}$ and $\downarrow\{\mathsf{s}\}$ satisfy (D)–(F), as we leave for the reader to check.

Example 2.1.23. If A is a PCA and F is a filter on A, then $(A, F, \cdot, \leq)$ and $F = (F, A^{\#}, \cdot, \leq)$ are also PCAs, as can be seen by taking the same k and s. In particular, A_{abs} and $A^{\#}$ are PCAs.

The elements k and s represent certain basic computations in A, and are usually called *combinators*, as they correspond to certain constants from Schönfinkel's combinatory logic [Sch24]. The letter k stands for *Konstanzfunktion*: $\mathsf{k}a$ can be seen as (a code for) the constant function with value a. The combinator s corresponds to Schönfinkel's *Verschmelzungsfunktion*. We can think of it as follows: if $a, b \in A$ depend on a further parameter c, then $\mathsf{s}ab$ is (a code for) an algorithm that, on input c, returns (at least as much information as) ac applied to bc, if defined.

The most important consequence of the existence of the combinators k and s is that every partial function obtained by repeatedly applying the application map is present as a computation in A itself. In order to make this statement precise, we use the terms introduced in the previous section. As we mentioned, every term defines a partial function $A^n \rightharpoonup A$. The key fact about PCAs is that such partial functions are 'computable' using an element from A itself.

Proposition 2.1.24 (Combinatory completeness)**.** *Let A be a PCA. There exists a map that assigns to each nonempty sequence $\vec{x}, y$ of distinct variables and term $t = t(\vec{x}, y)$, an element $\lambda^*\vec{x}, y.t \in A$, satisfying:*

(i) $(\lambda^*\vec{x}, y.t)\vec{a}\downarrow$ (where $\vec{a}$ has the same length as $\vec{x}$);

(ii) $(\lambda^*\vec{x}, y.t)\vec{a}b \preceq t(\vec{a}, b)$;

(iii) if all the constants occurring in t are from $A^{\#}$, then $\lambda^*\vec{x}, y.t \in A^{\#}$ as well.

Proof. Define the element $\mathsf{i} \in A^{\#}$ as skk. We will give a slightly more general construction than required for the proposition. For a variable u and a term s, we define a new term $\lambda^* u.s$ with the following properties:

- the free variables of $\lambda^* u.s$ are those of s minus u;
- if $\vec{v}$ contains the free variables of $\lambda^* u.s$, then the substitution instance $(\lambda^* u.s)[\vec{b}/\vec{v}]$ is defined for all $\vec{b} \in A$;
- moreover, if $a \in A$, then $(\lambda^* u.s)[\vec{b}/\vec{v}] \cdot a \preceq s[\vec{b}/\vec{v}, a/u]$;
- if all the constants occurring in s are from $A^{\#}$, then the same holds for $\lambda^* u.s$.

We define this new term recursively:

- If s is a constant or a variable distinct from u, then $\lambda^* u.s$ is $\mathsf{k}s$.
- If s is the variable u, then $\lambda^* u.s$ is i.
- If s is $s_0 s_1$, then $\lambda^* u.s$ is $\mathsf{s}(\lambda^* u.s_0)(\lambda^* u.s_1)$.

We leave the verification of the stated properties to the reader.

Now, if $\vec{x} = x_0, \ldots, x_{n-1}$, then we define $\lambda^* \vec{x}, y.t$ as the interpretation of the closed term:

$$\lambda^* x_0.(\cdots(\lambda^* x_{n-1}.(\lambda^* y.t))\cdots).$$

The verification of the properties (i), (ii) and (iii) is also left to the reader. □

Remark 2.1.25. (i) Note the k and s can be seen as special cases of combinatory completeness, for the terms $t(x, y) = x$ and $t(x, y, z) = xz(yz)$.

(ii) Note that the order of the variables matters here, i.e., $\lambda^* xy.t(x, y)$ will not be the same as $\lambda^* yx.t(x, y)$.

(iii) The notation λ^* is, of course, reminiscent of the λ-calculus. However, we should note that the λ^*-operator does not obey the usual rules of the λ-calculus, in particular not β-conversion. For examples, we refer to [vO08, p.4–5].

2.1.4 Some basic constructions in PCAs

Combinatory completeness allows us to perform constructions from classical recursion theory inside an arbitrary PCA. Many such constructions correspond to constants of combinatory logic, and are therefore also called combinators. In this section, we introduce a number of such combinators that will be useful throughout the thesis.

Construction 2.1.26 (Identity)**.** We have already seen the identity combinator $\mathsf{i} = \mathsf{skk} \in A^{\#}$, which satisfies $\mathsf{i}a \leq a$. If we write $\overline{\mathsf{k}} = \mathsf{ki} \in A^{\#}$, then we have $\overline{\mathsf{k}}ab \leq b$, so $\overline{\mathsf{k}}$ is the 'dual' of k.

Construction 2.1.27 (Pairing)**.** Let $\mathsf{p} = \lambda^* xyz.zxy$, $\mathsf{p}_0 = \lambda^* x.x\mathsf{k}$ and $\mathsf{p}_1 = \lambda^* x.x\overline{\mathsf{k}}$. These combinators satisfy $\mathsf{p}_0(\mathsf{p}ab) \leq a$ and $\mathsf{p}_1(\mathsf{p}ab) \leq b$. In particular, $\mathsf{p}ab$ is always defined, and we think of this element as (coding) the *pair* (a, b). Accordingly, p is called the pairing combinator, and p_0 and p_1 are known as the unpairing combinators.

Construction 2.1.28 (Booleans)**.** There exist $\mathsf{C}, \top, \bot \in A^\#$ such that $\mathsf{C}\top ab \leq a$ and $\mathsf{C}\bot ab \leq b$; the elements $\top$ and $\bot$ are called booleans. Indeed, we may simply take $\mathsf{C} = \mathsf{i}$, $\top = \mathsf{k}$ and $\bot = \overline{\mathsf{k}}$.

We must beware that, if we have terms $t_0(\vec{x})$, $t_1(\vec{x})$ and $t_2(\vec{x})$, then the term $t := \mathsf{C}t_0t_1t_2$ does not behave as one would expect at first glance. In particular, if $t_0(\vec{a}) \leq \top$ and $t_1(\vec{a})\downarrow$, then it does *not* follow that $t(\vec{a})$ is defined. Indeed, it may happen that $t_2(\vec{a})$ fails to be defined and, since t_2 is a subterm of t, this prevents $t(\vec{a})$ from being defined. We clearly do not want this, since we are not interested in the value (if any) of $t_2(\vec{a})$ when $t_0(\vec{a}) \leq \top$. Therefore, we introduce a *strong case distinction* (see also [LN15, Proposition 3.3.7]). If $t_0(\vec{x})$, $t_1(\vec{x})$ and $t_2(\vec{x})$ are terms, then we define a new term $t'(\vec{x})$ as:

$$\mathsf{C}t_0(\lambda^* y.t_1)(\lambda^* y.t_2)\mathsf{i}.$$

where y is not among the $\vec{x}$. One can easily check that this term has the following property: if $t_0(\vec{a}) \leq \top$, then $t'(\vec{a}) \preceq t_1(\vec{a})$, whereas if $t_0(\vec{a}) \leq \bot$, then $t'(\vec{a}) \preceq t_2(\vec{a})$. We will denote the term t' above by: $\mathsf{if}\ t_0\ \mathsf{then}\ t_1\ \mathsf{else}\ t_2$. Observe that, if all parameters from t_0, t_1 and t_2 are in $A^\#$, then the same holds for $\mathsf{if}\ t_0\ \mathsf{then}\ t_1\ \mathsf{else}\ t_2$.

Note that (strong) case distinction is really only useful when $t_0(\vec{a})$ cannot *simultaneously* lie below both $\top$ and $\bot$. We introduce a special name for PCAs in which this can happen.

Definition 2.1.29. *A PCA A is called* ***semitrivial*** *if the booleans $\top$ and $\bot$ have a common lower bound.*

Note that, in a semitrivial PCA, any two elements have a common lower bound. Indeed, if u is a common lower bound of $\top$ and $\bot$, and $a, b \in A$, then $\mathsf{C}uab$ is a common lower bound of a and b. In semitrivial PCAs, (strong) case distinction does not work well because we cannot really distinguish true and false. Therefore, certain constructions involving case distinctions will only work for PCAs that are not semitrivial. Examples of semitrivial PCAs are the PCAs from Example 2.1.21, and every PCA of the form DA (Example 2.1.22).

Construction 2.1.30 (Fixpoints)**.** Let $u = \lambda^* xy.y(xxy)$ and $\mathsf{y} = uu$, both of which are in $A^\#$. Then one easily verifies that $\mathsf{y}a \preceq a(\mathsf{y}a)$, so we can say that $\mathsf{y}a$, if defined, represents a fixpoint of the computation represented by a. The caveat 'if defined' is important here, however: if $\mathsf{y}a$ if not defined, then $\mathsf{y}a \preceq a(\mathsf{y}a)$ is trivially true, since both sides are undefined. As a result, the fixpoint combinator y is really only useful in total PCAs. In not necessarily total PCAs, one usually needs to use a *guarded*[1] fixpoint combinator, which is defined

[1] This terminology is taken from [LN15, Section 3.3.5].

as follows. Let $v = \lambda^* xyz.y(xxy)z$ and $\mathsf{z} = vv$, both of which are in $A^\#$ again. Then we have that $\mathsf{z}a$ is always defined, and $\mathsf{z}ab \preceq a(\mathsf{z}a)b$. So even though we do not necessarily have $\mathsf{z}a \preceq a(\mathsf{z}a)$ here, we do have $\lambda x.\mathsf{z}ax \leq \lambda x.a(\mathsf{z}a)x$ in the sense of Definition 2.1.4. We can use the combinator z to create self-referential definitions in a PCA. Explicitly, if $t(x, y)$ is a term, then setting $a := \lambda^* xy.t$ yields an element $T := \mathsf{z}a$ with the property that $Tb \preceq aTb \preceq t(T, b)$ for all $b \in A$. Moreover, if all the parameters from t are in $A^\#$, then $T \in A^\#$ as well. Obviously, this construction can be generalized to more variables in the place of b, either by adjusting the definition of z or by using the pairing combinators.

Construction 2.1.31 (Numerals)**.** As in the λ-calculus, we can represent the natural numbers in any PCA A. More precisely, for each natural number n we define a numeral $\overline{n} \in A^\#$ by $\overline{0} = \mathsf{i}$ and $\overline{n+1} = \mathsf{p}\bot\overline{n}$. The elements $\mathsf{zero} = \mathsf{p}_0$, $\mathsf{suc} = \lambda^* x.\mathsf{p}\bot x$ and $\mathsf{pred} = \lambda^* x.\mathsf{p}_0 x \mathsf{i}(\mathsf{p}_1 x)$ from $A^\#$ satisfy:

$$\mathsf{zero}\cdot\overline{0} \leq \top, \quad \mathsf{zero}\cdot\overline{n+1} \leq \bot, \quad \mathsf{suc}\cdot\overline{n} \leq \overline{n+1},$$
$$\mathsf{pred}\cdot\overline{0} \leq \overline{0} \quad \text{and} \quad \mathsf{pred}\cdot\overline{n+1} \leq \overline{n}.$$

Moreover, using the guarded fixpoint operator, we may construct a recursor $\mathsf{rec} \in A^\#$ such that

$$\mathsf{rec}ab\overline{0} \leq a \quad \text{and} \quad \mathsf{rec}ab\cdot\overline{n+1} \preceq b\overline{n}(\mathsf{rec}ab\overline{n}),$$

for all $n \in \mathbb{N}$ and $a, b \in A$. Using these combinators and the guarded fixpoint combinator from the previous construction, one sees that every partial recursive function can be represented in A. More precisely, this means that for every partial recursive function $f\colon \mathbb{N}^k \rightharpoonup \mathbb{N}$, there exists an element $r \in A^\#$ such that $r\overline{n_1}\cdots\overline{n_k} \preceq \overline{f(n_1, \ldots, n_k)}$, for all $n_1, \ldots, n_k \in \mathbb{N}$.

Let us note that, if A is not semitrivial, then no two numerals (for distinct natural numbers) have a common lower bound. Indeed, suppose that $\overline{m}$ and $\overline{n}$ have a common lower bound $u \in A$, where $m < n$. By applying pred to u exactly m times, we find a lower bound u' of $\overline{0}$ and $\overline{n-m}$. But then $\mathsf{zero}\cdot u'$ is a lower bound of $\top$ and $\bot$, so A is semitrivial.

In particular, any PCA which is not semitrivial must be infinite.

Construction 2.1.32 (Sequences)**.** Since we have a 'pairing' function given by $\mathsf{p} \in A^\#$, we can also code longer tuples in A. More precisely, we can define, for each $n \geq 0$, a total function $j^n\colon A^n \to A$ by:

- $j^0() = \mathsf{i}$;
- $j^{n+1}(a_0, \ldots, a_n) = \mathsf{p}a_0 \cdot j^n(a_1, \ldots, a_n)$.

Using these functions and the numerals constructed above, we can devise a coding of finite sequences in A. If $a_0, \ldots, a_{n-1}$ is a sequence, then we define its code by:

$$[a_0, \ldots, a_{n-1}] := \mathsf{p}\overline{n} \cdot j^n(a_0, \ldots, a_{n-1}).$$

Observe that $[a_0, \ldots, a_{n-1}]$ is built using the a_i, combinators from $A^\#$ and application. In particular, if all the a_i are from $A^\#$, then so is the code $[a_0, \ldots, a_{n-1}]$. Using the combinators defined thus far, one can mimick the standard recursion-theoretic arguments to show that all elementary operations on sequences are computable in terms of their codes. For example, there exist $\mathsf{lh}, \mathsf{read}, \mathsf{concat} \in A^\#$ such that:

- $\mathsf{lh} \cdot [a_0, \ldots, a_{n-1}] \leq \overline{n}$;
- $\mathsf{read} \cdot [a_0, \ldots, a_{n-1}] \cdot \overline{i} \leq a_i$ if $i < n$;
- $\mathsf{concat} \cdot [a_0, \ldots, a_{n-1}] \cdot [b_0, \ldots, b_{m-1}] \leq [a_0, \ldots, a_{n-1}, b_0, \ldots, b_{m-1}]$.

More combinators acting on sequences will be introduced as we have need for them.

In particular, a finite number of elements of A can be stored as a single element, namely, as a (code of a) sequence. This has the following consequence. Suppose that $a_0, \ldots, a_{n-1} \in A$. Then the least filter on A that contains $a_0, \ldots, a_{n-1}$ is equal to $\langle A^\# \cup \{a_0, \ldots, a_{n-1}\}\rangle$, which is the same as $\langle A^\# \cup \{[a_0, \ldots, a_{n-1}]\}\rangle$. We give a special name to this situation.

Definition 2.1.33. *A filter F on a PCA A is said to be* ***finitely generated*** *if it is of the form $\langle A^\# \cup \{a\}\rangle$ for a certain $a \in A$. Moreover, if $a \in A$, then we write $A[a] = (A, \langle A^\# \cup \{a\}\rangle, \cdot, \leq)$ (which is a PCA, by Example 2.1.23).*

We can think of $A[a]$ as the result of forcing the element a to be computable. Note that, in order for Definition 2.1.33 as stated here to be sensible, we really need that A is a PCA. Indeed, without the combinators, we have no way of storing a finite set of elements of A as a single element of A.

Remark 2.1.34. Of course, the combinators constructed above are far from unique. On the other hand, all of them may be constructed using only the elements k and s. When working with a PCA, we will assume that we have made an explicit choice for k and s, and as a result, a choice for all the combinators mentioned above.

2.1.5 Examples of PCAs

Thus far, we have not provided many interesting examples of PCAs, and in the examples that we gave, it is not clear that any kind of computation is going on. In this section, we give some more exciting examples of PCAs. We will not, for each of these examples, rigorously prove that the structure in question is indeed a PCA, but rather provide some intuition or refer to other sources.

Example 2.1.35 (Kleene's first model)**.** The archetypical example of a PCA is the absolute discrete PCA given by $\mathbb{N}$ with partial recursive function application. That is, we set $mn \simeq \varphi_m(n)$, where φ_m is the partial recursive function with Gödel number m. Combinatory completeness is an immediate consequence of the Smn-theorem. This PCA is called Kleene's first model, and is denoted by $\mathcal{K}_1$.

Example 2.1.36 (Oracles)**.** A variation on Kleene's first model is given by oracle computation. Let $f\colon \mathbb{N} \rightharpoonup \mathbb{N}$ be a (not necessarily recursive) partial function, and let $\varphi^f_{(-)}$ be some coding of the partial functions that are recursive relative to an oracle for f. Then $mn \simeq \varphi^f_m(n)$ also defines a (absolute discrete) PCA structure on $\mathbb{N}$, that we denote by $\mathcal{K}_1^f$.

This construction has been generalized to arbitrary absolute discrete PCAs in [vO06]. We will discuss this construction in detail in Section 5.1.2 below.

Example 2.1.37 (Kleene's second model)**.** As we mentioned in the introduction, there is a PCA underlying function realizability, which is now known as Kleene's second model. Its domain is $\mathbb{N}^\mathbb{N}$, the set of all functions $\mathbb{N} \to \mathbb{N}$. Before we can define the application map, we need to introduce some notation. Fix a recursive coding of finite sequences $(a_0, \dots, a_{n-1}) \mapsto [a_0, \dots, a_{n-1}] \in \mathbb{N}$. For $\alpha \in \mathbb{N}^\mathbb{N}$ and $n \in \mathbb{N}$, we write $\alpha|_n = [\alpha(0), \dots, \alpha(n-1)]$. Moreover, we write $[n] * \alpha$ for the function α' defined by $\alpha'(0) = n$ and $\alpha'(i+1) = \alpha(i)$.

Each function $\alpha \in \mathbb{N}^\mathbb{N}$ determines a partial function $F_\alpha\colon \mathbb{N}^\mathbb{N} \rightharpoonup \mathbb{N}$ as follows. We set $F_\alpha(\beta) = m$ if and only if there exists an $n \in \mathbb{N}$ such that:

- for all $i < n$, we have $\alpha(\beta|_i) = 0$;
- $\alpha(\beta|_n) = m + 1$.

We can think of this definition as the following process. The function α interrogates the function β, successively demanding more values until it comes up with an output. More precisely, at each stage, α either asks for an additional value of β by returning 0, or presents the final output m by returning $m+1$. Note that F_α is in general a partial function because it may happen that $\alpha(\beta|_n) = 0$ for all n. In this case, α keeps demanding values of β, never coming up with a final output.

The application map of Kleene's second model is now defined as follows. For $\alpha, \beta \in \mathbb{N}^\mathbb{N}$, we say that $\alpha\beta\downarrow$ iff $F_\alpha([n] * \beta)\downarrow$ for all $n \in \mathbb{N}$, and in this case, $\alpha\beta$ is defined by $\alpha\beta(n) = F_\alpha([n] * \beta)$. So in the computation of $\alpha\beta(n)$, the interrogator α is first presented with the input n, and can then proceed to inspect as many values of β as necessary. This definition of application yields a absolute discrete PCA, which is called Kleene's second model and denoted by $\mathcal{K}_2$. For a proof of combinatory completeness, we refer to [vO08, Section 1.4.3].[2]

With this definition of application, computability of total functions coincides with continuity. More precisely, a function $(\mathbb{N}^\mathbb{N})^n \to \mathbb{N}^\mathbb{N}$ is of the form $(\beta_1, \dots, \beta_n) \mapsto \alpha\beta_1 \cdots \beta_n$ for some $\alpha \in \mathbb{N}^\mathbb{N}$ if and only if this function is continuous with respect to the Baire space topology on $\mathbb{N}^\mathbb{N}$.

Function realizability does not arise precisely from this version of Kleene's second model, but from a *relative* version, where $\mathcal{K}_2^\# = \mathcal{K}_2^{\mathrm{rec}}$ consists of the total *recursive* functions. By Example 2.1.23, $\mathcal{K}_2^{\mathrm{rec}}$ is also an absolute PCA in its own right.

[2] [vO08] uses the 'strict' definition of PCAs mentioned in Remark 2.1.20(ii). For this definition, the proof of combinatory completeness given in [vO08] is not entirely correct, as is acknowledged and repaired in [vO11]. For our definition of PCAs, however, the proof in [vO08] works fine.

Example 2.1.38 (Van Oosten model)**.** The Van Oosten model[3] $\mathcal{B}$, first introduced in [vO99], is like Kleene's second model, except that its domain consists of all *partial* functions $\mathbb{N} \rightharpoonup \mathbb{N}$. For $\alpha \in \mathcal{B}$, we will define a function $F_\alpha \colon \mathcal{B} \rightharpoonup \mathbb{N}$ in a similar spirit as in the previous example. There is a complication, however, due to the fact that functions are partial. Suppose the interrogator α is interested in the value (if any) of $\beta(42)$. Then it cannot simply successively ask initial values of β until it hits $\beta(42)$, since $\beta(i)$ could be undefined for some $i < 42$, sending the interrogation astray. So in addition to declaring *that* it wants to see another value of β, the interrogator must also specify *which* value it wants to see. This leads to the following definition of F_α. We say that $F_\alpha(\beta) = m$ iff there exists a finite sequence $u_0, \ldots, u_{n-1}$ of natural numbers such that:

- for all $i < n$, there is a k_i such that $\alpha([u_0, \ldots, u_{i-1}]) = 2k_i + 1$ and $\beta(k_i) = u_i$;
- $\alpha([u_0, \ldots, u_{n-1}]) = 2m$.

Clearly, the sequence $u_0, \ldots, u_{n-1}$ is unique if it exists, which means that $F_\alpha(\beta) = m$ for at most one m, i.e., F_α is indeed a partial function. As in the previous example, $F_\alpha(\beta)$ can fail to be defined of α keeps demanding values from β forever. In the current case, there is also the additional possibility that α tries to interrogate β on a value k_i outside the domain of β, which also causes $F_\alpha(\beta)$ to be undefined.

The application map is defined by: $\alpha\beta(n) \simeq F_\alpha([n] * \beta)$, where $[n] * \beta$ is defined analogously to the previous example. Note that in this case, we do not have to require that $F_\alpha([n] * \beta)$ is always defined, because $\alpha\beta$ is allowed to be a partial function. As a result, this makes $\mathcal{B}$ into a *total* PCA. The proof of combinatory completeness of $\mathcal{B}$, and the generalization to other PCAs, will be treated in Section 5.2.1 below.

Like $\mathcal{K}_2$, the Van Oosten model has a relative version, where $\mathcal{B}^\# = \mathcal{B}^{\mathrm{pr}}$ consists of all partial recursive functions. Again, $\mathcal{B}^{\mathrm{pr}}$ is a PCA in its own right. The Van Oosten model can be regarded as an ordered PCA by setting $\alpha \leq \beta$ iff α is a superfunction of β, cf. Remark 2.1.5. If we also want to regard this ordered version as a relative PCA, then we need to set $\mathcal{B}^\# = {\uparrow}\mathcal{B}^{\mathrm{pr}}$, the set of all *subfunctions* of partial recursive functions.

Example 2.1.39 (Scott's graph model)**.** Scott's graph model, introduced in [Sco75], is another example of a total PCA. Its domain is $\mathcal{P}\omega$, the set of sets of natural numbers. Let us fix a recursive bijective pairing function $\langle \cdot, \cdot \rangle \colon \mathbb{N} \times \mathbb{N} \to \mathbb{N}$, and write $e_{(-)}$ for the bijection $\mathbb{N} \to \mathcal{P}_{\mathrm{fin}}(\mathbb{N})$ given by $e_n = p$ iff $n = \sum_{i \in p} 2^i$. Now we set

$$AB = \{m \mid \exists n\,(e_n \subseteq B \text{ and } \langle n, m \rangle \in A)\}.$$

As for $\mathcal{K}_2$, there is a connection with continuity here. The set $\mathcal{P}\omega$ can be equipped with the Scott topology, whose basic opens are those sets of the form

[3] Even though this model was introduced in 1999, it only received this name in [LN15, Section 3.2.4].

$U_p = \{A \in \mathcal{P}\omega \mid p \subseteq A\}$ for *finite* sets p. A function $(\mathcal{P}\omega)^n \to \mathcal{P}\omega$ is of the form $(B_1, \ldots, B_n) \mapsto AB_1 \cdots B_n$ for some $A \in \mathcal{P}\omega$ iff it is continuous w.r.t. this topology. Since the application map $(A, B) \mapsto AB$ is itself continuous, this immediately implies that $\mathcal{P}\omega$ is combinatorially complete, so it is indeed a total PCA.

Scott's graph model also has a relative version, where $(\mathcal{P}\omega)^\# = (\mathcal{P}\omega)^{\text{re}}$ consists of all recursively enumerable sets of natural numbers.

2.2 Morphisms of PCAs

Thus far, we have only discussed PCAs as isolated objects. In this section, we discuss morphisms between PCAs. These are not yet Longley's applicative morphisms mentioned in the Introduction; we will treat those in Section 2.3. First, we introduce morphisms and show that they form a category enriched over preorders. Then, we consider some special properties that morphisms may enjoy, and give a number of examples of morphisms.

2.2.1 The category of PCAs

A morphism of PCAs $A \to B$ will be a function between the underlying sets, but not one that 'preserves the structure'. Instead, we require that the structure is preserved 'up to a realizer'. For example, we do not require $f(aa')$ to be literally (Kleene) equal to $f(a) \cdot f(a')$. Instead, we ask for an algorithm from $B^\#$ that 'simulates' the application from A inside B, and similarly for the order. Let us make this precise in the following definition.

Definition 2.2.1. *Let A and B be PCAs. A* ***morphism of PCAs*** *$A \to B$ is a function $f\colon A \to B$ satisfying:*

(i) f restricts to a function $A^\# \to B^\#$, i.e., $f(a) \in B^\#$ for all $a \in A^\#$;

(ii) there exists an element $t \in B^\#$ such that $t \cdot f(a) \cdot f(a') \preceq f(aa')$;

(iii) there exists a $u \in B^\#$ such that $u \cdot f(a') \leq f(a)$ whenever $a' \leq a$.

We say that the morphism f is ***realized*** *by $t, u \in B^\#$. We may also say that f preserves application up to t, and preserves the order up to u.*

The set of functions $A \to B$, and in particular, the set of morphisms of PCAs, carries a preorder. In the same spirit as in Definition 2.2.1, we do not define this order as the pointwise order, but rather as the pointwise order 'up to a realizer'.

Definition 2.2.2. *If A and B are PCAs and $f, f'\colon A \to B$ are functions, then we say that $f \leq f'$ iff there exists an $s \in B^\#$ such that $s \cdot f(a) \leq f'(a)$. Such an element s is said to* ***realize*** *the inequality $f \leq f'$. Moreover, we say that f and f' are* ***isomorphic****, written $f \simeq f'$, if both inequalities $f \leq f'$ and $f' \leq f$ hold.*

Proposition 2.2.3. *Partial combinatory algebras, morphisms of PCAs, and inequalities between morphisms of PCAs form a preorder-enriched category, which we denote by* OPCA*.*

Proof. If A is a PCA, then id_A preserves both application and the order 'on the nose'. In particular, id_A preserves both application and the order up to $\mathsf{i} \in A^\#$.

Suppose we have morphisms of PCAs $A \xrightarrow{f} B \xrightarrow{g} C$. Let $t, u \in B^\#$ realize f and let $t', u' \in C^\#$ realize g. Then gf preserves application up to

$$t'' := \lambda^* xy.u'(t'(t' \cdot g(t) \cdot x)y) \in C^\#.$$

Indeed, for $a, a' \in A$, we have:

$$\begin{aligned} t'' \cdot g(f(a)) \cdot g(f(a')) &\preceq u'(t'(t' \cdot g(t) \cdot g(f(a))) \cdot g(f(a'))) \\ &\preceq u'(t' \cdot g(t \cdot f(a)) \cdot g(f(a'))) \\ &\preceq u' \cdot g(t \cdot f(a) \cdot f(a')) \\ &\preceq g(f(aa')), \end{aligned}$$

as desired. Moreover, gf preserves the order up to $\lambda^* x.u'(t' \cdot g(u) \cdot x)$, as a similar computation shows. This proves that OPCA is a category.

If $f \colon A \to B$ is a morphism of PCAs, then $\mathsf{i} \in B^\#$ realizes $f \leq f$. Now suppose that we have morphisms of PCAs $f, f', f'' \colon A \to B$, and elements $s, s' \in B^\#$ realizing $f \leq f'$ and $f' \leq f''$. Then $\lambda^* x.s'(sx)$ realizes $f \leq f''$, so we can conclude that $\leq$ is a preorder on each homset. It remains to show that composition preserves inequality.

Suppose we have morphisms of PCAs $A \underset{f'}{\overset{f}{\rightrightarrows}} B \xrightarrow{g} C$ and an $s \in B^\#$ realizing $f \leq f'$. If $t, u \in C^\#$ realize g, then $gf \leq gf'$ is realized by $\lambda^* x.u(t \cdot g(s) \cdot x)$.

Finally, suppose we have morphisms of PCAs $A \xrightarrow{f} B \underset{g'}{\overset{g}{\rightrightarrows}} C$. Then any realizer of $g \leq g'$ also realizes $gf \leq g'f$. □

Remark 2.2.4. The first letter in OPCA stands for 'ordered'. The reason we don't simply use PCA here (after all, PCAs are ordered by default) is that in the literature, PCA invariably stands for the category of discrete PCAs and *applicative* morphisms, as introduced by Longley in [Lon94].

When working with generated filters, the following lemma will be useful. Essentially, it says that we only need to check requirement (i) of Definition 2.2.1 on a generating set of $A^\#$.

Lemma 2.2.5. *Let A and B be PCAs and suppose that $A^\# = \langle A_0 \rangle$ for some $A_0 \subseteq A$. Let $f \colon A \to B$ be a function satisfying (ii) and (iii) of Definition 2.2.1, and also:*

(i)' if $a \in A_0$, then $f(a) \in B^\#$.

Then f is a morphism of PCAs $A \to B$.

Proof. Let $t, u \in B^\#$ satisfy requirements (ii) and (iii), and consider $f^{-1}(B^\#) = \{a \in A \mid f(a) \in B^\#\}$. We claim that $f^{-1}(B^\#)$ is a filter on the PAP A. First, if $a, a' \in f^{-1}(B^\#)$ are such that $aa'\downarrow$, then we have $t \cdot f(a) \cdot f(a') \leq f(aa')$. Since $t, f(a), f(a') \in B^\#$, this implies that $f(aa') \in B^\#$, as desired. The argument that $f^{-1}(B^\#)$ is upwards closed is similar, using $u \in B^\#$.

Requirement (i)' says that $A_0 \subseteq f^{-1}(B^\#)$. Since $f^{-1}(B^\#)$ is a filter, this implies $A^\# = \langle A_0 \rangle \subseteq f^{-1}(B^\#)$, which is requirement (i) from Definition 2.2.1. □

2.2.2 Properties of morphisms of PCAs

In this section, we introduce some special properties that a morphism of PCAs may have.

Definition 2.2.6. *Let $f\colon A \to B$ be a morphism of PCAs.*

*(i) We say that f is **decidable** if there exists a $d \in B^\#$ (a **decider**) such that $d \cdot f(\top) \leq \top$ and $d \cdot f(\bot) \leq \bot$.*

*(ii) We say that f is **computationally dense** (c.d. for short) if there exists an $n \in B^\#$ satisfying:*

$$\forall s \in B^\# \, \exists r \in A^\# \, (n \cdot f(r) \leq s). \tag{cd}$$

*(iii) We say that f is **dense** if f is c.d. and there exists an $n \in B^\#$ satisfying:*

$$\forall s \in B \, \exists r \in A \, (n \cdot f(r) \leq s). \tag{d}$$

Note that, if f is dense, the ns satisfying (cd) and (d) are not required to be the same element; but see also Lemma 2.2.9 below. Clearly, if A and B are both absolute PCAs, then computational density and density coincide, and in this case, it also coincides with the original notion from [HvO03]. However, the definition we have given here is, even in the absolute case, not quite the definition from [HvO03], but rather a simplification due to Johnstone [Joh13]. We will formulate the original definition from [HvO03] (or rather, two versions of it) and prove its equivalence to Definition 2.2.6(ii) and (iii) in Lemma 2.2.12 below.

The following definition formulates an extremely strict map from A to B, where all the structure should be preserved and reflected 'on the nose'. As we mentioned, this is not the 'correct' notion of a morphism between PCAs, but we will need the definition for technical purposes.

Definition 2.2.7. *A function $f\colon A \to B$ is called an **elementary inclusion** if it satisfies the following three conditions:*

(i) $B^\# = \uparrow(f(A^\#))$;

(ii) $f(aa') \simeq f(a) \cdot f(a')$*;*

(iii) $a \leq a'$ *iff* $f(a) \leq f(a')$*.*

Moreover, we say an elementary inclusion f *is* ***essentially surjective*** *if* $B = \uparrow(f(A))$*.*

Clearly, an elementary inclusion of PCAs will automatically be a morphism of PCAs. Moreover, it will always be a pseudomono in OPCA, as we leave to the reader to verify.

The following lemmata establish some useful properties of the notions from Definition 2.2.6 and Definition 2.2.7.

Lemma 2.2.8. *Let* $f: A \to B$ *be a morphism of PCAs.*

(i) If f *is dense, then* f *is c.d.*

(ii) If f *is c.d., then* f *is decidable.*

(iii) If f *is an elementary inclusion, then* f *is c.d.*

(iv) If f *is an essentially surjective elementary inclusion, then* f *is dense.*

Proof. (i) holds by definition.

(ii). Suppose that f is c.d. Let $t, u \in B^\#$ realize f, and suppose that $n \in B^\#$ satisfies (cd). Then we may find $r_0, r_1 \in A^\#$ such that $n \cdot f(r_0) \leq \top$ and $n \cdot f(r_1) \leq \bot$. Now consider the element $r = \lambda^* x.\mathsf{C}xr_0r_1 \in A^\#$, so that $r\top \leq r_0$ and $r\bot \leq r_1$. Finally, set $d = \lambda^* x.n(u(t \cdot f(r) \cdot x))$. Then we have:

$$d \cdot f(\top) \preceq n(u(t \cdot f(r) \cdot f(\top))) \preceq n(u \cdot f(r\top)) \preceq n \cdot f(r_0) \leq \top,$$

and similarly, $d \cdot f(\bot) \leq \bot$, so d is a decider for f.

(iii). If f is an elementary inclusion, then by requirement (i) from Definition 2.2.7, $\mathsf{i} \in B^\#$ satisfies (cd) for f.

(iv) follows similarly. □

Lemma 2.2.9. *If* $f: A \to B$ *is dense, then there exists an* $n \in B^\#$ *satisfying* both (cd) *and* (d)*.*

Proof. Let $t, u \in B^\#$ realize f, and let $d \in B^\#$ be a decider for f, which exists by Lemma 2.2.8. Let $n_0, n_1 \in B^\#$ satisfy (cd) and (d) respectively, and define $n \in B^\#$ as:

$$\lambda^* x.\mathsf{if}\ d(u(t \cdot f(\mathsf{p}_0) \cdot x))\ \mathsf{then}\ n_0(u(t \cdot f(\mathsf{p}_1) \cdot x))\ \mathsf{else}\ n_1(u(t \cdot f(\mathsf{p}_1) \cdot x)).$$

We claim that n satisfies both (cd) and (d). Indeed, let $s \in B^\#$, and take an $r \in A^\#$ such that $n_0 \cdot f(r) \leq s$. If we define $r' = \mathsf{p}\top r \in A^\#$, then we see that

$$d(u(t \cdot f(\mathsf{p}_0) \cdot f(r'))) \preceq d(u \cdot f(\mathsf{p}_0 r')) \preceq d \cdot f(\top) \leq \top,$$

so

$$n \cdot f(r') \preceq n_0(u(t \cdot f(\mathsf{p}_1) \cdot f(r'))) \preceq n_0(u \cdot f(\mathsf{p}_1 r')) \preceq n_0 \cdot f(r) \leq s,$$

so n satisfies (cd). Similarly, if $s \in B$, then take $r \in A$ such that $n_1 \cdot f(r) \leq s$ and define $r' = \mathsf{p}\bot r$. A similar calculation shows that $n \cdot f(r') \leq s$, so n also satisfies (d). □

Lemma 2.2.10. *Let $A \xrightarrow{f} B \xrightarrow{g} C$ be morphisms of PCAs.*

(i) If f and g are both decidable / c.d. / dense, then so is gf.

(ii) If gf is decidable / c.d. / dense, then so is g.

(iii) If g is an elementary inclusion and gf is decidable / c.d., then f is also decidable / c.d. If, moreover, g is essentially surjective and gf is dense, then f is also dense.

Proof. Let $t, u \in C^{\#}$ realize g.

(i). If $d \in B^{\#}$ and $d' \in C^{\#}$ are deciders for f resp. g, then a decider for gf is given by $\lambda^* x.d'(u(t \cdot g(d) \cdot x)) \in C^{\#}$. The proofs for (computational) density are analogous.

(ii). Suppose $d \in C^{\#}$ is a decider for gf, and set $r = \lambda^* x.\mathsf{C} \cdot x \cdot f(\top) \cdot f(\bot) \in B^{\#}$. Then $\lambda^* x.d(u(t \cdot g(r) \cdot x))$ is a decider for g. Moreover, if gf is c.d., then any $n \in C^{\#}$ satisfying (cd) for gf also satisfies (cd) for g; similarly for the case of density.

(iii). If gf is decidable, then using a decider for gf, we may construct a $c \in C^{\#}$ such that $c \cdot g(f(\top)) \leq g(\top)$ and $c \cdot g(f(\bot)) \leq g(\bot)$. We may find a $d \in B^{\#}$ such that $g(d) \leq c$. Now we have:

$$g(d \cdot f(\top)) \simeq g(d) \cdot g(f(\top)) \preceq c \cdot g(f(\top)) \leq g(\top),$$

so $d \cdot f(\top) \leq \top$. Similarly, we find $d \cdot f(\bot) \leq \bot$, so d is a decider for f. The other statements follows similarly. □

Lemma 2.2.11. *(i) All notions from Definition 2.2.6 are downwards closed, that is, if we have morphisms of PCAs $f, f' \colon A \to B$ such that $f' \leq f$ and f is decidable / c.d. / dense, then f' is also decidable / c.d. / dense.*

(ii) Let $A \underset{g}{\overset{f}{\rightleftarrows}} B$ be morphisms of PCAs. If $fg \leq \mathrm{id}_B$, then f is dense. In particular, left adjoints are dense.

Proof. (i) Let $s \in B^{\#}$ be a realizer of $f' \leq f$. If $d \in B^{\#}$ is a decider for f, then $\lambda^* x.d(sx)$ is a decider for f'. The proofs for (computational) density are analogous.

(ii) Clearly, id_B is dense, so by (i), fg is also dense. The statement now follows from Lemma 2.2.10(ii). □

The following lemma offers alternative characterizations of (computational) density, which are equivalent to the original definiton from [HvO03] in the absolute case. Note that for these definitions, it would be a lot harder to establish the properties of (computational) density established in the lemmata above! The reason for formulating this lemma, is that we will need the alternative characterization of (computational) density in the proof of Theorem 2.3.14 below.

Lemma 2.2.12 (Hofstra, Van Oosten, Johnstone)**.** *Let $f: A \to B$ be a morphism of PCAs.*

(i) f is computatonally dense if and only if there exists an $m \in B^\#$ satisfying:

$$\forall s \in B^\# \, \exists r \in A^\# \, \forall a \in A \; (m \cdot f(ra) \preceq s \cdot f(a)). \tag{cdm}$$

(ii) If f is c.d., then f is dense if and only if there exists an $m \in B^\#$ satisfying:

$$\forall s \in B \, \exists r \in A \forall a \in A \; (m \cdot f(ra) \preceq s \cdot f(a)). \tag{dm}$$

In fact, any element satisfying (cdm) *also satisfies* (cd)*, and any element satisfying* (dm) *also satisfies* (d)*. Moreover, if f is dense, then there exists an $m \in B^\#$ satisfying both* (cdm) *and* (dm)*.*

Proof. Let $t, u \in B^\#$ realize f.

(i) First, suppose that f is c.d. and that $n \in B^\#$ satisfies (cd). Consider:

$$m = \lambda^* x.n(u(t \cdot f(\mathsf{p}_0) \cdot x))(u(t \cdot f(\mathsf{p}_1) \cdot x)) \in B^\#. \tag{2.1}$$

Let $s \in B^\#$, and find a $r_0 \in A^\#$ such that $n \cdot f(r_0) \leq s$. Finally, set $r = \mathsf{p}r_0 \in A^\#$. Then for all $a \in A$, we have $\mathsf{p}_0(ra) \leq r_0$ and $\mathsf{p}_1(ra) \leq a$, which yields:

$$\begin{aligned} m \cdot f(ra) &\preceq n(u(t \cdot f(\mathsf{p}_0) \cdot f(ra)))(u(t \cdot f(\mathsf{p}_1) \cdot f(ra))) \\ &\preceq n(u \cdot f(\mathsf{p}_0(ra)))(u \cdot f(\mathsf{p}_1(ra))) \\ &\preceq n \cdot f(r_0) \cdot f(a) \\ &\preceq s \cdot f(a), \end{aligned}$$

as desired.

Conversely, suppose that $m \in B^\#$ satisfies (cdm), and consider $s \in B^\#$. Then we also have $\mathsf{k}s \in B^\#$, so we may find an $r_0 \in A^\#$ such that $m \cdot f(r_0 a) \preceq \mathsf{k}s \cdot f(a)$ for all $a \in A$. Finally, set $r = r_0\mathsf{k} \in A^\#$. Then we have $m \cdot f(r) \simeq m \cdot f(r_0\mathsf{k}) \preceq \mathsf{k}s \cdot f(\mathsf{k}) \leq s$, as desired.

(ii) follows by exactly the same constructions. Moreover, if f is dense, then by Lemma 2.2.9, there is an $n \in B^\#$ satifying both (cd) and (d). The element $m \in B^\#$ defined in (2.1) will satisfy both (cdm) and (dm). □

2.2.3 Examples of morphisms of PCAs

Example 2.2.13. If $f: A \to B$ is a morphism of PCAs, then restricting f to $A^\#$ yields a morphism of PCAs $f^\#: A^\# \to B^\#$. Moreover, f can also be seen as a morphism of PCAs $f_{\text{abs}}: A_{\text{abs}} \to B_{\text{abs}}$.

Example 2.2.14. Let A be a PCA and F be a filter on A. As we have seen, $F = (F, A^{\#}, \cdot, \leq)$ is a PCA in its own right. The inclusion $F \hookrightarrow A$ is an elementary inclusion, so in particular, it is a c.d. morphism of PCAs. Such a morphism will be called an *inclusion of a filter*. Note that the inclusion $TA \hookrightarrow DA$ is an example of an inclusion of a filter.

Example 2.2.15. Again, let A be a PCA and F be a filter on A. Then $(A, F, \cdot, \leq)$ is also a PCA, and the identity on A is a morphism of PCAs $A \to (A, F, \cdot, \leq)$.

In particular, if F is finitely generated by $a \in A$, we have a morphism $\iota_a : A \to A[a]$, where $A[a]$ is as in Definition 2.1.33. We show that this morphism is dense. Clearly, it suffices to show that ι_a is c.d. Let $n = \lambda x.xa \in A[a]^{\#}$. If $s \in A[a]^{\#}$, then by Lemma 2.1.14, there exists a pure term $t(\vec{x}, y)$ such that $t(\vec{b}, a) \leq s$ for certain $\vec{b} \in A^{\#}$. Now consider the element $r = \lambda^* y.t(\vec{b}, y) \in A^{\#}$. Then we have $nr \preceq ra \preceq t(\vec{b}, a) \leq s$, so n satisfies (cd) for ι_a, as desired.

We will revisit this example in the next section, after introducing applicative morphisms; this will provide another way of showing that ι_a is dense.

Example 2.2.16. A trivial example of a morphism of PCAs is the morphism $f : A \to B$ given by $f(a) = \mathsf{k}$. This is the largest morphism of PCAs $A \to B$. Indeed, if $f' : A \to B$, then kk realizes $f' \leq f$.

Example 2.2.17. Let $f : A \to B$ be a morphism of PCAs. Then we can define a morphism $Tf : TA \to TB$ by $Tf(\alpha) = \downarrow(f(\alpha)) = \downarrow\{f(a) \mid a \in \alpha\}$. Let us check that this is indeed a morphism. First of all, if $\alpha \in (TA)^{\#}$, then there exists some $a \in \alpha \cap B^{\#}$, and we have $f(a) \in Tf(\alpha) \cap B^{\#}$, so $Tf(\alpha) \in (TB)^{\#}$. Moreover, Tf preserves the order on the nose, and if f preserves application up to $t \in A^{\#}$, then Tf preserves application up to $\downarrow\{t\} \in (TA)^{\#}$.

We can say even more, namely that T is a pseudofunctor. Indeed, suppose we have morphisms of PCAs $A \xrightarrow{f} B \xrightarrow{g} C$. Then we have

$$\begin{aligned} T(gf)(\alpha) &= \downarrow\{g(f(a)) \mid a \in \alpha\}, \\ Tg(Tf(\alpha)) &= \downarrow\{g(b) \mid b \in \downarrow f(\alpha)\} = \downarrow\{g(b) \mid \exists a \in \alpha\, (b \leq f(a))\}. \end{aligned}$$

These are not in general equal, but we do have $T(gf)(\alpha) \subseteq Tg(Tf(\alpha))$, which yields $T(gf) \leq Tg \circ Tf$. Conversely, if g preserves the order up to $u \in C^{\#}$, then $\downarrow\{u\} \in (TC)^{\#}$ realizes $Tg \circ Tf \leq T(gf)$. Moreover, we easily see that $T(\mathrm{id}_A)$ is the identity on TA. Finally, if $f, f' : A \to B$ and $s \in B^{\#}$ realizes $f \leq f'$, then $\downarrow\{s\} \in (TB)^{\#}$ realizes $Tf \leq Tf$'.

In a completely similar fashion, we see that D can be made into a pseudofunctor $\mathsf{OPCA} \to \mathsf{OPCA}$. The inclusions $TA \hookrightarrow DA$ then constitute a natural transformation $T \Rightarrow D$.

The following two examples define pseudomonad structures on T and D, which we will study more closely in Section 2.3.1 below.

Example 2.2.18. For each PCA A, there is a morphism of PCAs $\delta_A : A \to TA$ which sends $a \in A$ to the principal downset $\downarrow\{a\}$. It is easy to check that δ is in fact an elementary inclusion which is essentially surjective; in particular, δ_A is dense. Moreover, δ constitutes a pseudonatural transformation $\mathrm{id}_{\mathsf{OPCA}} \Rightarrow T$. Indeed, if $f\colon A \to B$, then

$$\delta_B(f(a)) = {\downarrow}\{f(a)\} \quad \text{and} \quad Tf(\delta_A(a)) = {\downarrow}\{f(a') \mid a' \leq a\}.$$

Again, these are not in general equal, but we do have $\delta_B(f(a)) \subseteq Tf(\delta_A(a))$, which yields $\delta_B \circ f \leq Tf \circ \delta_A$. In the other direction, if f preserves the order up to $u \in B^{\#}$, then ${\downarrow}\{u\} \in (TB)^{\#}$ realizes $Tf \circ \delta_A \leq \delta_B \circ f$.

Similarly, we have $\delta'_A\colon A \to DA$ given by $a \mapsto {\downarrow}\{a\}$, which forms a natural transformation $\delta'\colon \mathrm{id}_{\mathsf{OPCA}} \Rightarrow D$. In this case, δ'_A is an elementary inclusion, so it is c.d., but it is not dense!

Example 2.2.19. For each PCA A, there is a morphism of PCAs $\bigcup_A\colon TTA \to TA$ sending $\mathcal{A} \in TTA$ to $\bigcup \mathcal{A}$. Since $\bigcup({\downarrow}\{\alpha\}) = \alpha$, i.e., $\bigcup_A \circ \delta_{TA} = \mathrm{id}_{TA}$, we see that $\bigcup_A$ is dense. Moreover, $\bigcup$ constitutes a natural transformation $TT \Rightarrow T$; in this case, one easily checks that naturality holds on the nose.

Similarly, we have a dense morphism of PCAs $\bigcup'_A\colon DDA \to DA$ given by union, yielding a natural transformation $\bigcup'\colon DD \Rightarrow D$.

Finally, let us give some examples of morphisms of PCAs involving the PCAs from Section 2.1.5.

Example 2.2.20. Let A be a PCA. Then we have a morphism of PCAs $\mathcal{K}_1 \to A$ which sends $n \in \mathbb{N}$ to its Curry numeral $\overline{n} \in A^{\#}$. In order to see that this is indeed a morphism, note that $(m, n) \mapsto mn$ is a partial recursive function of two variables. Now the final remark of Construction 2.1.31 tells us that there exists an $r \in A^{\#}$ such that $r \cdot \overline{m} \cdot \overline{n} \preceq \overline{mn}$, so application is preserved up to r.

If $f\colon \mathbb{N} \rightharpoonup \mathbb{N}$ is a partial function, then $n \mapsto \overline{n}$ is a morphism of PCAs $\mathcal{K}_1^f \to A$ iff there exists an $s \in A^{\#}$ such that $s \cdot \overline{n} \preceq \overline{f(n)}$. We will prove this, in greater generality, in Section 5.1.2.

Example 2.2.21. Consider the relative versions of Kleene's second model $\mathcal{K}_2 = (\mathcal{K}_2, \mathcal{K}_2^{\mathrm{rec}}, \cdot, =)$ and Scott's graph model $\mathcal{P}\omega = (\mathcal{P}\omega, (\mathcal{P}\omega)^{\mathrm{re}}, \cdot, =)$. There is a morphism of PCAs $f\colon \mathcal{K}_2 \to \mathcal{P}\omega$ which sends $\alpha \in \mathcal{K}_2$ to its graph $f(\alpha) = \{\langle n, \alpha(n)\rangle \mid n \in \mathbb{N}\}$. Let us see that this is indeed a morphism. First of all, if α is a recursive function, then its graph is a decidable set, so it is certainly r.e. It remains to prove that f preserves the application up to some realizer. Note that, if $\alpha\beta(n) = m$, then this depends on only finitely many values of α and β. Moreover, the relation $R(a, b, n, m)$ expressing 'e_a and e_b code graphs of finite functions p and q, and whenever $p \subseteq \alpha$ and $q \subseteq \beta$, we have $\alpha\beta(n) = m$' is r.e. Now let A be the r.e. set consisting of all $\langle a, \langle b, \langle n, m\rangle\rangle\rangle$ such that $R(a, b, n, m)$. Then it is easily checked that $A \cdot f(\alpha) \cdot f(\beta) = f(\alpha\beta)$ when $\alpha\beta{\downarrow}$, as desired.

We have a similar example involving the (relative, but discrete) Van Oosten model $\mathcal{B} = (\mathcal{B}, \mathcal{B}^{\mathrm{pr}}, \cdot, =)$. Again, we define an applicative morphism $f\colon \mathcal{B} \to \mathcal{P}\omega$ which sends a function $\alpha \in \mathcal{B}$ to its graph $f(\alpha) = \{\langle n, \alpha(n)\rangle \mid n \in \operatorname{dom}\alpha\}$. If α is partial recursive, then its graph is recursively enumerable, and the construction of a realizer proceeds similarly as above.

2.3 Applicative morphisms

In this section, we define Longley's notion of an applicative morphism between PCAs. We follow the treatment from [HvO03], which shows that T is a monad on OPCA and recovers the notion of an applicative morphism as a morphism in the Kleisli category of T. Moreover, we introduce a related notion of *partial* applicative morphism, which arises similarly from D. After introducing (partial) applicative morphisms, we will extend the notions from Definition 2.2.6 to (partial) applicative morphisms and investigate when (partial) applicative morphisms have a right adjoint. Finally, we will give a few examples of (partial) applicative morphisms.

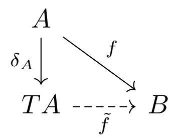
2.3.1 Two more categories of PCAs

As we have seen in the previous section, T and D are pseudofunctors on OPCA. The natural transformations $\delta, \bigcup$ resp. $\delta', \bigcup'$ we introduced for these functors yield two pseudomonad structures.

Proposition 2.3.1. *The triples* $(T, \delta, \bigcup)$ *and* $(D, \delta', \bigcup')$ *are pseudomonads on* OPCA. *Moreover, both pseudomonads are KZ, meaning that* $T\delta \dashv \bigcup \dashv \delta T$ *resp.* $D\delta' \dashv \bigcup' \dashv \delta' D$.

Proof. The proof is exactly the same as for the (nonempty) downset monad on the category of posets. ☐

It is well-known that for KZ-pseudomonads, algebra structures on A are left adjoint to the unit, and are thus unique. For a characterization of the T-algebras, we refer to [HvO03, Section 4]. For each PCA A, we have the free T-algebra $(TA, \bigcup_A)$. If $x: TB \to B$ is another T-algebra, then any morphism of PCAs $f: A \to B$ lifts to an essentially unique morphism of T-algebras:



Explicitly, we have $\tilde{f} \simeq x \circ Tf$. Since T is a KZ-pseudomonad, we can say a bit more: $\tilde{f}$ is the *least* morphism of PCAs $TA \to B$ that makes the diagram above commute. Indeed, suppose that $g: TA \to B$ is any morphism of PCAs such that $g \circ \delta_A \simeq f$. Then we have

$$\tilde{f} \simeq x \circ Tf \simeq x \circ Tg \circ T\delta_A \leq x \circ Tg \circ \delta_{TA} \simeq x \circ \delta_B \circ g \simeq g, \tag{2.2}$$

where we used that a KZ-pseudomonad satisfies $T\delta \leq \delta T$. For the monad D, similar remarks apply.

Definition 2.3.2. *(i) The preorder-enriched bicategory* OPCA_T *is defined as the Kleisli category of the pseudomonad* $(T, \delta, \bigcup)$. *An arrow of this category is called an* ***applicative morphism***. *We write* $f: A \multimap B$ *to indicate* *that f is an applicative morphism from A to B; this means that f is a morphism of PCAs $A \to TB$.*

(ii) The preorder-enriched bicategory OPCA_D *is defined as the Kleisli category of the pseudomonad $(D, \delta', \bigcup')$. An arrow of this category is called a **partial applicative morphism**. We write $f\colon A \leftharpoondown\!\!\circ\ B$ to indicate that f is a aprtial applicative morphism from A to B; this means that f is a morphism of PCAs $A \to DB$.*

Restricting the category OPCA_T to the absolute discrete PCAs yields Longley's category PCA introduced in [Lon94]. The category OPCA_D will mainly serve an auxilliary purpose in this thesis, especially when studying adjunctions between applicative morphisms. In Section 2.3.3 below, we will see that certain applicative morphisms have a right adjoint in OPCA_D, but not in OPCA_T.

In OPCA_T, the identity map on A is given by $\delta_A\colon A \multimap A$, i.e., $a \mapsto {\downarrow}\{a\}$. If $A \overset{f}{\multimap} B \overset{g}{\multimap} C$, then $gf\colon A \multimap C$ is equal to the composition

$$A \xrightarrow{f} TB \xrightarrow{Tg} TTC \xrightarrow{\bigcup} TC,$$

meaning that $(gf)(a) = \bigcup_{b \in f(a)} g(b)$. As we have seen above, if $f\colon A \multimap B$, then there is an essentially unique pseudofactorization

$$\begin{array}{ccc} A & & \\ {\scriptstyle \delta_A}\downarrow & \searrow^{f} & \\ TA & \underset{\tilde{f}}{\dashrightarrow} & TB \end{array} \tag{2.3}$$

where $\tilde{f}$ is an *algebra* morphism between the free algebras TA and TB. Explicitly, we can describe $\tilde{f}$ as $\bigcup \circ Tf$, meaning that $\tilde{f}(\alpha) = \bigcup f(\alpha) = \bigcup_{a \in \alpha} f(a)$. The fact that $\tilde{f}$ is an algebra morphism means that $\tilde{f}$ is, up to realizer, *union preserving*. Since every element of TA is a union of principal downsets, it stands to reason that $\tilde{f}$ is essentially determined by its action on the image of δ_A. It is well-known that the assignment $f \mapsto \tilde{f}$ is a pseudofunctor which yields an equivalence between OPCA_T and the category of *free* T-algebras. For OPCA_D, similar remarks apply. In particular, partial applicative morphisms $f\colon A \leftharpoondown\!\!\circ\ B$ correspond to D-algebra morphisms $\tilde{f}\colon DA \to DB$.

The definition of (partial) applicative morphisms as in Definition 2.3.2 above can be a bit cumbersome, so it is useful to break this definition down a bit. First of all, let us note that every applicative morphism $A \multimap B$, i.e., morphism of PCAs $A \to TB$, can also be seen as a partial appplicative morphism $A \leftharpoondown\!\!\circ\ B$, namely as $A \to TB \hookrightarrow DB$. Since $TB \hookrightarrow DB$ is a pseudomono, it is easy to see that this presents OPCA_T as a preorder-enriched sub-bicategory of OPCA_D. In particular, what we will say below on the definition of partial applicative morphisms will also hold for applicative morphisms.

If $f\colon A \leftharpoondown\!\!\circ\ B$ is a partial applicative morphism, then f must preserve application up to some $\tau \in (DB)^{\#}$. If $t \in \tau \cap B^{\#}$, then we see that f also preserves application up to $\downarrow\{t\} \in (DB)^{\#}$. In order to work with these kind of realizers efficiently, we introduce the following notation.

Notation 2.3.3. Let A be a PAP. If $a \in A$ and $\alpha \in DA$, then we write

$$a \cdot \alpha :\simeq \downarrow\{a\} \cdot \alpha \simeq \downarrow\{aa' \mid a' \in \alpha\}.$$

Note that the second Kleene equality here is a consequence of axiom (A) of Definition 2.1.1.

In particular, if $\alpha, \beta \in DA$, then $a \cdot \alpha \subseteq \beta$ amounts to showing that $aa' \in \beta$ for all $a' \in \alpha$. Now we can reformulate the definition of a partial applicative morphism as follows. A function $f\colon A \to DB$ is a partial applicative morphism if it satisfies the following conditions.

(i) If $a \in A^{\#}$, then $f(a) \cap B^{\#}$ is nonempty.

(ii) There exists a $t \in B^{\#}$ such that $t \cdot f(a) \cdot f(a') \preceq f(aa')$. By abuse of terminology, we say that f preserves application up to t.

(iii) There exists a $u \in B^{\#}$ such that $u \cdot f(a') \subseteq f(a)$ whenever $a' \leq a$. Again, we say that f preserves the order up to u.

We will also say that $t, u \in B^{\#}$ realize f. The intuition behind these clauses is now as follows. We think of a partial applictive morphism as a simulation of A inside B, and we think of the statement $b \in f(a)$ as 'b represents the element a'. Thus each element has a (down)set of representatives in B. Contrary to the case of morphisms of PCAs, this set can contain more than one element, or in the case of *partial* applicative morphisms, none at all. The clauses (i)-(iii) above then state:

(i) Every element of $A^{\#}$ is represented by at least one element of $B^{\#}$.

(ii) There is an algorithm $t \in B^{\#}$ that simulates the application on A. That is, if b, b' represent a, a' and aa' is defined, then tbb' is also defined and a representative of aa'.

(ii) There is an algorithm $u \in B^{\#}$ that simulates the order on A. That is, if b' represents a' and $a' \leq a$, then ub' is defined and a representative of a.

Moreover, we have $f \leq f'$ if and only if there is an $s \in B^{\#}$ such that $s \cdot f(a) \subseteq f'(a)$; and we say that s realizes $f \leq f'$. Intuitively, s turns representatives of a w.r.t. f into representatives of a w.r.t. f'.

Remark 2.3.4. It is worth reflecting on the fact that we defined OPCA_T and OPCA_D as preorder-enriched *bi*categories. This is due to the fact that T and D are only *pseudo*monads. But in fact, OPCA_T and OPCA_D are both 'almost' preorder-enriched categories. The only axiom for 1-categories that does not hold on the nose in either OPCA_T or OPCA_D is $f \circ \mathrm{id} \cong f$. In fact, if f is a (partial) applicative morphism, then $f \circ \mathrm{id} = f$ holds on the nose iff f preserves the order on the nose, that is, $a' \leq a$ implies $f(a') \subseteq f(a)$. As we shall see below in Lemma 2.3.5, it is no real restriction to consider only such f.

It is also worth noting that, if for $f\colon A \multimap B$ we define $\tilde{f}\colon TA \to TB$ as $\bigcup \circ Tf$, then the assignment $f \mapsto \tilde{f}$ becomes *2*-functorial, rather than pseudo-functorial.

The following was shown in [HvO03, Lemma 3.3].

Lemma 2.3.5. *Suppose that $f\colon A \mathrel{\leftharpoonup\!\!\circ} B$ is a partial applicative morphism. Then there exists an $f'\colon A \mathrel{\leftharpoonup\!\!\circ} B$ such that $f \simeq f'$ and f' preserves the order on the nose.*

Proof. Define f' as the composition $A \overset{\mathrm{id}}{\multimap} A \overset{f}{\multimap} B$, or more explicitly, $f'(a) = \bigcup_{a' \leq a} f(a')$. Then of course, we have $f' \simeq f$, and from the explicit description of the morphism f' it is also clear that f' preserves the order on the nose. □

Thus, one could restrict OPCA_T and OPCA_D to the order-preserving (partial) applicative morphisms to obtain actual preorder-enriched categories; but we will not take this approach here.

The name 'partial applicative morphism' deserves some explanation. For this, we need the following definition.

Definition 2.3.6. *Let $f\colon A \mathrel{\leftharpoonup\!\!\circ} B$ be a partial applicative morphism. Then the **domain** of f is defined as:*

$$\operatorname{dom} f := \{a \in A \mid f(a) \in TB\} = \{a \in A \mid f(a) \neq \emptyset\}.$$

*We say that f is **total** if $\operatorname{dom} f = A$, equivalently, if $f\colon A \to DB$ factors through $TB \hookrightarrow DB$.*

Note that the inclusion $\mathsf{OPCA}_T \hookrightarrow \mathsf{OPCA}_D$ identifies applicative morphisms with the total partial applicative morphisms.

Lemma 2.3.7. *If $f\colon A \mathrel{\leftharpoonup\!\!\circ} B$ is a partial applicative morphism, then $\operatorname{dom} f$ is a filter on A.*

Proof. Note that $\operatorname{dom} f = f^{-1}(TA)$, and TA is a filter on DA, so we can use the same argument as in the proof of Lemma 2.2.5. □

In particular, $\operatorname{dom} f = (\operatorname{dom} f, A^{\#}, \cdot, \leq)$ is itself a PCA. Now we can view $f|_{\operatorname{dom} f}\colon \operatorname{dom} f \to TB$ as an applicative morphism $\operatorname{dom} f \multimap B$, and it is easily verified that

$$\begin{array}{ccc} \operatorname{dom} f & \xrightarrow{f|_{\operatorname{dom} f}} & TB \\ \big\downarrow & & \big\downarrow \\ A & \xrightarrow{\quad f \quad} & DB \end{array}$$

is a pseudopullback in OPCA. Conversely, if F is a filter on A and $f\colon F \multimap B$ is an applicative morphism, then we can (uniquely) extend f to a partial applicative morphism $f^{+}\colon A \mathrel{\leftharpoonup\!\!\circ} B$ such that $\operatorname{dom} f^{+} = F$. In this way, we can view partial applicative morphisms as partial maps from A to TB, where the admissible domains are inclusions of a filter.

2.3.2 Properties of (partial) applicative morphisms

In this section, we extend the notions from Definition 2.2.6 to (partial) applicative morphisms.

Definition 2.3.8. *(i) An applicative morphism $f\colon A \multimap B$ is called* ***decidable / c.d. / dense*** *iff $f\colon A \to TB$ is decidable / c.d. / dense as a morphism of PCAs.*

(ii) A partial applicative morphism $f\colon A \rightharpoonup\!\!\!\circ\, B$ is called ***decidable / c.d.*** *iff $f\colon A \to DB$ is decidable / c.d. as a morphism of PCAs.*

Since $\mathsf{OPCA}_T \hookrightarrow \mathsf{OPCA}_D$, we should check the following.

Lemma 2.3.9. *If $f\colon A \multimap B$, then f is decidable / c.d. iff $f\colon A \rightharpoonup\!\!\!\circ\, B$ is decidable / c.d. as a total partial applicative morphism.*

Proof. Given Lemma 2.2.8(iii) and Lemma 2.2.10(i) and (iii), this follows immediately from the diagram

$$\begin{array}{ccc} A & \xrightarrow{\;f\;} & TB \\ & \searrow & \big\downarrow \\ & & DB \end{array}$$

and the fact that $TB \hookrightarrow DB$ is an elementary inclusion. $\square$

Remark 2.3.10. We have refrained from defining a notion of density for partial applicative morphisms, for the following reason. If we say that $A \rightharpoonup\!\!\!\circ\, B$ is dense iff $A \to DB$ is dense as a morphism of PCAs, then the analogue of Lemma 2.3.9 for density would fail, which would be highly confusing. In fact, this notion of density for partial applicative morphism is not very intersting at all. If a morphism $f\colon A \to DB$ is c.d., then it is dense iff $\emptyset$ is in the range of f, which is to say that $f\colon A \rightharpoonup\!\!\!\circ\, B$ is not total.

As we did for Definition 2.3.2, we can break down the notions from Definition 2.3.8 in terms of realizers from $B^\#$ rather than $(DB)^\#$. If $f\colon A \rightharpoonup\!\!\!\circ\, B$ is an applicative morphism, then:

- f is decidable iff there is a $d \in B^\#$ (also called a decider for f) such that $d \cdot f(\top) \subseteq \downarrow\{\top\}$ and $d \cdot f(\bot) \subseteq \downarrow\{\bot\}$;
- f is c.d. iff there exists an $n \in B^\#$ such that

$$\forall s \in B^\# \, \exists r \in A^\# \, (n \cdot f(r) \subseteq \downarrow\{s\}); \tag{cd'}$$

- if f is total, then f is dense iff it is c.d. and there exists an $n \in B^\#$ such that

$$\forall s \in B \, \exists r \in A \, (n \cdot f(r) \subseteq \downarrow\{s\}). \tag{d'}$$

Note that Lemma 2.2.8(i) and (ii) and Lemma 2.2.11(i) carry over automatically to OPCA_T, and to OPCA_D insofar as they do not involve density. Moreover, Lemma 2.2.9 tells us that for a dense $f\colon A \multimap B$, we may select an $n \in B^\#$ satisfying both (cd') and (d'). Using Lemma 2.2.12, we can deduce that $f\colon A \hookrightarrow B$ is c.d. iff there exists an $m \in B^\#$ such that

$$\forall s \in B^\# \exists r \in A^\# \forall a \in A(m \cdot f(ra) \preceq s \cdot f(a)). \qquad \text{(cdm')}$$

If f is total, then f is dense iff f is c.d. and there exists an $m \in B^\#$ such that

$$\forall s \in B \exists r \in A \forall a \in A(m \cdot f(ra) \preceq s \cdot f(a)), \qquad \text{(dm')}$$

and we may find an $m \in B^\#$ satisfying both (cdm') and (dm'). Moreover, any $m \in B^\#$ satisfying (cdm') resp. (dm') also satisfies (cd') resp. (dm').

For a result like Lemma 2.2.10, we need to be a bit more careful, since we have to deal with composition. We will need the following result.

Lemma 2.3.11. *(i) If $f\colon A \multimap B$ is an applicative morphism, then f is decidable / c.d. / dense iff $\tilde{f}\colon TA \to TB$ is decidable / c.d. / dense.*

(ii) If $f\colon A \hookrightarrow B$ is a partial applicative morphism, then f is decidable / c.d. iff $\tilde{f}\colon DA \to DB$ is decidable / c.d.

Proof. (i) follows immediately from diagram (2.3) on page 49, along with the fact that δ_A is dense. The proof of (ii) is similar. □

Now Lemma 2.2.10 and Lemma 2.2.11(ii) also carry over to OPCA_T, and to OPCA_D insofar as they do not involve density. For example, if $A \overset{f}{\multimap} B \overset{g}{\multimap} C$ are applicative morphisms such that gf is dense, then $\widetilde{gf} \simeq \tilde{g} \circ \tilde{f}$ is dense, so $\tilde{g}$ is dense, so g is dense. In particular, left adjoints in OPCA_T are dense. On the other hand, left adjoints in OPCA_D can only be guaranteed to be *computationally* dense.

We know from the theory of monads that $(f\colon A \to B) \mapsto (\delta_B f\colon A \multimap B)$ is a functor $\mathsf{OPCA} \to \mathsf{OPCA}_T$. Since δ_B is always a pseudomono, this presents OPCA as a preorder-enriched subcategory of OPCA_T. We give a special name to the morphisms in the image of this inclusion.

Definition 2.3.12. *An applicative morphism $f\colon A \multimap B$ is called **projective** iff it is in the essential image of $\mathsf{OPCA} \hookrightarrow \mathsf{OPCA}_T$, i.e., there is a morphism of PCAs $f_0\colon A \to B$ such that $f \simeq \delta_B f_0$. Equivalently, f is projective iff $\tilde{f}\colon TA \to TB$ lies in the essential image of T.*

In fact, for $f\colon A \multimap B$ to be projective, it suffices that there be a *function* $f_0\colon A \to B$ such that $f \simeq \delta_B f_0$; this function will then automatically be a morphism of PCAs. Similarly to Lemma 2.3.9, we should prove the following result.

Lemma 2.3.13. *A morphism of PCAs $f_0\colon A \to B$ is decidable / c.d. / dense iff $\delta_B f_0\colon A \multimap B$ is decidable / c.d. / dense.*

Proof. This follows immediately from the fact that δ_B is an essentially surjective elementary inclusion. □

2.3.3 Right adjoints of (partial) applicative morphisms

Computational density was introduced in [HvO03] in order to study right adjoints of functors between categories of assemblies (see Section 3.3 below). In this section, we study the existence of right adjoints within the categories OPCA_T and OPCA_D. First of all, let us note that any partial applicative morphism which has a right adjoint in OPCA_D must actually be total. Indeed, suppose we have $f\colon A \rightharpoonup\!\circ\, B$ and $g\colon B \rightharpoonup\!\circ\, A$ with $f \dashv g$. Then in particular, we have $\mathrm{id}_A \leq gf$, which yields:

$$A = \mathrm{dom}(\mathrm{id}_A) \subseteq \mathrm{dom}(gf) \subseteq \mathrm{dom}\, f,$$

so f is total. Therefore, we really are interested when an applicative morphism $f\colon A \multimap B$ has a right adjoint in either OPCA_T or OPCA_D. The following theorem answers this question. For the absolute case, this was shown in [FvO14, Corollary 1.15]. However, [FvO14] arrives at this result through studying functors between categories of assemblies, whereas we will prove it directly.

Theorem 2.3.14. *Let $f\colon A \multimap B$ be an applicative morphism. Then the following are equivalent:*

(i) f has a right adjoint in OPCA_D.

(ii) f is projective and c.d.

Moreover, the following are also equivalent:

(iii) f has a right adjoint in OPCA_T.

(iv) f is projective and dense.

Proof. First, suppose that f has a right adjoint $g\colon B \rightharpoonup\!\circ\, A$. We have already observed that this implies that f is c.d. For projectivity, suppose that $r \in A^\#$ realizes $\mathrm{id}_A \leq gf$ and $s \in B^\#$ realizes $fg \leq \mathrm{id}_B$. Then for all $a \in A$, we have that $ra\downarrow$ and $ra \in gf(a) = \bigcup_{b \in f(a)} g(b)$. By the Axiom of Choice, there exists a function $f_0\colon A \to B$ such that $f_0(a) \in f(a)$ and $ra \in g(f_0(a))$ for all $a \in A$. We claim that $f \simeq \delta_B \circ f_0$. First of all, we have that $\downarrow\{f_0(a)\} \subseteq f(a)$, so the identity combinator realizes $\delta_B \circ f_0 \leq f$. The converse inequality is realized by $s' := \lambda^* x.s(tr'x) \in B^\#$, where r' is an element from $f(r) \cap B^\#$ and f preserves application up to $t \in B^\#$. Indeed, if $b \in f(a)$, then $tr'b \in f(ra) \subseteq \bigcup_{a' \in g(f_0(a))} f(a') = fg(f_0(a))$. So we see that $s'b \preceq s(tr'b)$, which is defined and an element of $\mathrm{id}_B(f_0(a)) = \downarrow\{f_0(a)\}$, as desired.

Conversely, suppose that $f \simeq \delta_B \circ f_0$, where $f_0\colon A \to B$ is a c.d. morphism of PCAs. Pick an $m \in B^\#$ satisfying (cdm) for f_0. Let $t, u \in B^\#$ realize f_0. We define $g\colon B \rightharpoonup\!\circ\, A$ by:

$$g(b) = \downarrow\{a \in A \mid m \cdot f_0(a) \leq b\}. \tag{2.4}$$

First of all, let us prove that g is a partial applicative morphism. Clearly, g preserves the order on the nose. We know from Lemma 2.2.12 that m also satisfies (cd) for f_0, so if $b \in B^{\#}$, then $g(b)$ contains an element from $A^{\#}$. Now define

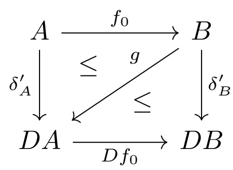
$$s = \lambda^* x.m(u(t \cdot f_0(\mathsf{p}_0) \cdot x))(m(u(t \cdot f_0(\mathsf{p}_1) \cdot x))) \in B^{\#}.$$

Take $r \in A^{\#}$ such that $m \cdot f_0(ra) \preceq s \cdot f_0(a)$ for all $a \in A$, and define $t' = \lambda^* xy.r(\mathsf{p}xy) \in A^{\#}$. We claim that g preserves application up to t'. Suppose that $a \in g(b)$ and $a' \in g(b')$. Then $t'aa' \preceq r(\mathsf{p}aa')$, and:

$$\begin{aligned} m \cdot f_0(r(\mathsf{p}aa')) &\preceq s \cdot f_0(\mathsf{p}aa') \\ &\preceq m(u(t \cdot f_0(\mathsf{p}_0) \cdot f_0(\mathsf{p}aa')))(m(u(t \cdot f_0(\mathsf{p}_1) \cdot f_0(\mathsf{p}aa')))) \\ &\preceq m(u \cdot f_0(\mathsf{p}_0(\mathsf{p}aa')))(m(u \cdot f_0(\mathsf{p}_1(\mathsf{p}aa')))) \\ &\preceq m \cdot f_0(a)(m \cdot f_0(a')) \\ &\preceq bb'. \end{aligned}$$

In particular, if $bb'\downarrow$, then $taa' \in g(bb')$, as desired.

It remains to show that g is right adjoint to f. Note that the composition $A \overset{f}{\multimap} B \overset{g}{\multimap} A$ is isomorphic to $A \overset{f_0}{\to} B \overset{g}{\to} DB$, and that the composition $B \overset{g}{\multimap} A \overset{f}{\multimap} B$ is isomorphic to $B \overset{g}{\to} DA \overset{Df_0}{\longrightarrow} DB$. This means we should show the inequalities in OPCA as in the diagram:



By (cdm), there exists an $r \in A^{\#}$ such that $m \cdot f_0(ra) \preceq \mathsf{i} \cdot f_0(a) \leq f_0(a)$ for all $a \in A$. This means that $ra\downarrow$ and $ra \in g(f_0(a))$, so r realizes $\delta_A \leq gf_0$.

For the other inequality, consider $s = \lambda^* x.m(ux) \in B^{\#}$. Let $b \in B$, and suppose that we have a $b' \in Df_0(g(b))$. Then there exists an $a' \in g(b)$ such that $b' \leq f_0(a')$. And $a' \in g(b)$ means that there exists an $a \geq a'$ such that $m \cdot f_0(a) \leq b$. This yields:

$$sb' \preceq m(ub') \preceq m(u \cdot f_0(a')) \preceq m \cdot f_0(a) \leq b,$$

so sb' is defined and an element of $\delta_B(b)$, i.e., s realizes $Df_0 \circ g \leq \delta_B$.

For the equivalence of (iii) and (iv), if f has a right adjoint in OPCA_T, then f must be dense, and by (i)⇒(ii) above, f is projective.

Conversely, if f is dense, then we may select an $m \in B^{\#}$ satisfying both (cdm) and (dm). Then we see that the right adjoint g defined in (2.4) is actually total, since m satisfies (d), as desired. □

As an immediate corollary, we see that the notion of equivalence of PCAs is the same in all three categories OPCA, OPCA_T and OPCA_D. Indeed, suppose that $f\colon A \multimap B$ and $g\colon B \multimap A$ constitute an equivalence in OPCA_D. Then the remark preceding Theorem 2.3.14 tells us that f and g are total, so the equivalence already exists in OPCA_T. Moreover, Theorem 2.3.14 tells us that f and g are projective, so the equivalence already exists in OPCA.

As another corollary of this theorem, we can obtain characterizations of (computationally) dense (partial) applicative morphisms. In the absolute case, this was shown in [FvO14, Corollary 2.3].

Corollary 2.3.15. *Let $f\colon A \rightharpoonup\!\circ\, B$ be a partial applicative morphism. Then the following are equivalent:*

(i) f is c.d.;

(ii) there exists a partial applicative morphism $h\colon B \rightharpoonup\!\circ\, A$ such that $fh \leq \mathrm{id}_B$;

(iii) $\tilde{f}\colon DA \to DB$ has a right adjoint in OPCA.

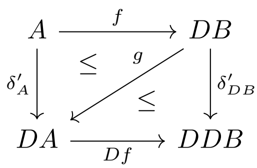
If f is total, then the following are also equivalent:

(iv) f is dense;

(v) there exists an applicative morphism $h\colon B \multimap A$ such that $fh \leq \mathrm{id}_B$;

(vi) $\tilde{f}\colon TA \to TB$ has a right adjoint in OPCA.

Proof. (i)⇒(iii). We apply Theorem 2.3.14 to the projective and c.d. applicative morphism $\delta_{DB} \circ f\colon A \multimap DB$ to obtain a right adjoint $g\colon DB \rightharpoonup\!\circ\, A$ in OPCA_D. That is, $g\colon DB \to DA$ is a morphism of PCAs such that:



Now we see that $\tilde{f}g \simeq \bigcup \circ Df \circ g \leq \bigcup \circ \delta'_{DB} \simeq \mathrm{id}_{DB}$. Moreover, we have $\delta'_A \leq gf \simeq g\tilde{f}\delta'_A$. From this, we'd like to conclude that $\mathrm{id}_{DA} \leq g\tilde{f}$, but we should be careful, since $g\tilde{f}\colon DA \to DA$ is not necessarily a D-*algebra* morphism. By applying (2.2) (or rather, its version for D) for $g\tilde{f}$ instead of g, we find:

$$\mathrm{id}_{DA} \simeq \widetilde{\delta'_A} \leq \widetilde{g\tilde{f}\delta'_A} \leq g\tilde{f},$$

as desired.

(iii)⇒(ii). If $\tilde{f}$ has a right adjoint $g\colon DB \to DA$, then define $h\colon B \rightharpoonup\!\circ\, A$ as $g \circ \delta_B$. The composition $B \overset{h}{\rightharpoonup\!\circ} A \overset{f}{\rightharpoonup\!\circ} B$ is isomorphic to the composition $B \overset{h}{\to} DA \overset{\tilde{f}}{\to} DB$. In OPCA, we have $\tilde{f}h = \tilde{f}g\delta_B \leq \delta_B$, which means that $fh \leq \mathrm{id}_B$ in OPCA_D.

(ii)⇒(i) is immediate.

The equivalence of (iv)-(vi) follows similarly. ∎

For future reference, it will be useful to have an explicit description of the right adjoint of $\tilde{f}\colon DA \to DB$. If $f\colon A \rightharpoonup\!\circ\, B$ is c.d. and $m \in B^{\#}$ satisfies (cdm'), then $\downarrow\{m\} \in (DB)^{\#}$ satifies (cdm) for $\tilde{f}\colon DA \to DB$, so by (2.4) on page 54, the right adjoint $g\colon DB \to DA$ can be described as

$$g(\beta) = \downarrow\{a \in A \mid m \cdot f(a) \subseteq \beta\} \quad \text{for } \beta \in DB. \tag{2.5}$$

If we assume, as we may by Lemma 2.3.5, that f preserves the order on the nose, then the downset sign in (2.5) can be removed. For the right adjoint of $\tilde{f}\colon TA \to TB$, we can use the same formula for an $m \in B^{\#}$ satisfying both (cdm') and (dm').

When discussing functors between categories of assemblies in the next chapter, the following lemma, wich analyzes the situation in Corollary 2.3.15 further, will be needed.

Lemma 2.3.16. *Let $f\colon A \rightharpoonup\!\circ\, B$ be a c.d. partial applicative morphism, and let $g\colon DB \to DA$ be the right adjoint of $\tilde{f}$. If we consider g as a partial applicative morphism $DB \rightharpoonup\!\circ\, A$, then:*

(i) f is total iff $\operatorname{dom} g \subseteq TB$*;*

(ii) if f is total, then f is dense iff $\operatorname{dom} g = TB$*.*

Proof. We use the explicit description of g as in (2.5).

(i) Suppose that f is total and $\beta \in \operatorname{dom} g$, i.e., $g(\beta) \neq \emptyset$. If $a \in g(\beta)$, then by the totality of f, there exists a $b' \in f(a)$, and we see that $mb' \in m \cdot f(a) \subseteq \beta$, so $\beta \in TB$. Conversely, if f is not total, then there exists an $a \in A$ such that $f(a) = \emptyset$. Then we also have $a \in g(\emptyset)$, so $\emptyset \in \operatorname{dom} g$, but $\emptyset \notin TB$.

(ii) Suppose that f is total and dense. Then by (i), we have $\operatorname{dom} g \subseteq TB$. For the converse inclusion, we note that we can take $m \in B^{\#}$ to satisfy both (cdm') and (dm'), and then (2.5) immediately implies that $g(\beta) \neq \emptyset$ if $\beta \neq \emptyset$, that is, $TB \subseteq \operatorname{dom} g$.

Conversely, if $\operatorname{dom} g = TB$, then by (2.5), we see that m satisfies (dm'), so f is dense. □

2.3.4 Examples of (partial) applicative morphisms

In this section, we give examples of (partial) applicative morphisms that are not projective, and therefore do not arise from a morphism of PCAs. In fact, all of the examples below are right adjoints of a projective applicative morphism that we have already encountered!

Example 2.3.17. Let A be a PCA and let F be a filter on A. Then the inclusion $F \hookrightarrow A$ can be viewed as a projective c.d. applicative morphism $F \multimap A$, so it should have a right adjoint in OPCA_D. This right adjoint is simply given by $A \rightharpoonup\!\circ\, F$ sending $a \in A$ to $\downarrow\{a\} \cap F$. Note that the composition $F \to A \rightharpoonup\!\circ\, F$ is the identity applicative morphism on F; in particular, $A \rightharpoonup\!\circ\, F$ is c.d. as well.

If $f\colon A \rightharpoonup\!\!\circ\, B$ is a partial applicative morphism and F is a filter on B, then the domain of the composition $A \rightharpoonup\!\!\circ\, B \rightharpoonup\!\!\circ\, F$ is equal to

$$\{a \in A \mid f(a) \cap F \neq \emptyset\}.$$

In particular, this is always a filter on A.

In the special case where $F = B^{\#}$, we will use the following notation.

Definition 2.3.18. *Let $f\colon A \rightharpoonup\!\!\circ\, B$ be a partial applicative morphism. We define*

$$\mathrm{dom}^{\#} f = \{a \in A \mid f(a) \cap B^{\#} \neq \emptyset\}.$$

By Example 2.3.17, $\mathrm{dom}^{\#} f$ is always a filter on A.

Example 2.3.19. Consider the situation from Example 2.3.17, where F was a filter on A, and we defined the right adjoint $f\colon A \rightharpoonup\!\!\circ\, F$ of $F \to A$ by: $\mathrm{dom}\, f = F$ and $f(a) = {\downarrow}\{a\} \cap F$. This partial applicative morphism is itself c.d., since $\mathsf{i} \in A^{\#}$ witnesses (cd'). This means that $\tilde{f}\colon DA \to DF$, which sends $\alpha \in DA$ to $\alpha \cap F$, should have a right adjoint as well, and we can use (2.5) to define it explicitly.

One should be a bit careful, though. It is tempting to think that $\mathsf{i} \in A^{\#}$ also witnesses (cdm'), since $f(ra) \simeq r \cdot f(a)$ for all $r \in A^{\#}$ and $a \in A$. The latter is not true, however, since it could happen that $ra \in F$, while $a \notin F$. Instead, a suitable witness for (cdm') is given by $\lambda^* x.\mathsf{p}_0 x(\mathsf{p}_1 x) \in A^{\#}$, as can be read off from the proof of Lemma 2.2.12. This means that the right adjoint $g\colon DF \to DA$ can be defined as:

$$g(\alpha) = \{a \in A \mid \text{if } a \in F, \text{ then } \mathsf{p}_0 a(\mathsf{p}_1 a) \in \alpha\} \quad \text{for } \alpha \in DF.$$

If $F \neq A$, then f is not total, and, in accordance with Lemma 2.3.16(i), we see that $g(\emptyset) = A \backslash F \neq \emptyset$.

In the case $F = A^{\#}$, then the g we defined here is isomorphic to the tripos transformation ∇ introduced in [ABS02, Section 3] (see also Proposition 3.3.16 below). Our definition (which we read off from the proof of Corollary 2.3.15) seems to be a bit simpler than the definition in [ABS02].

Example 2.3.20. Let A be a PCA, $a \in A$, and consider the dense morphism of PCAs $\iota_a\colon A \to A[a]$. Viewing this as a projective dense applicative morphism $\iota_a\colon A \multimap A[a]$, we know it must have a right adjoint $g\colon A[a] \multimap A$ in OPCA_T. Explicitly, we can define g by:

$$g(b) = \{c \in A \mid ca \leq b\} \quad \text{for } b \in A.$$

Let us verify that this is indeed an applicative morphism. First of all, $\mathsf{k}b \in g(b)$ for all $b \in B$, so $g(b) \neq \emptyset$. Note that g preserves application up to $\mathsf{s} \in A^{\#}$, and that it preserves the order on the nose. By Lemma 2.2.5, in order to conclude that g is an applicative morphism, it suffices to show that $A^{\#} \cup \{a\}$ is contained in $\mathrm{dom}^{\#} g$. For $r \in A^{\#}$, we have $\mathsf{k}r \in g(r) \cap A^{\#}$. Moreover, we have $\mathsf{i} \in g(a) \cap A^{\#}$, which completes the proof that g is an applicative morphism.

Now it is easily checked that $\mathsf{k} \in A^{\#} \subseteq A[a]^{\#}$ realizes both $\mathrm{id}_A \leq g\iota_a$ and $\mathrm{id}_{A[a]} \leq \iota_a g$, and moreover, $\lambda^* x.xa \in A[a]^{\#}$ realizes $\iota_a g \leq \mathrm{id}_{A[a]}$. This proves that $\iota_a \dashv g$. Note that this provides an alternative way of proving that ι_a is dense.

Example 2.3.21. Consider the relative version $\mathcal{K}_2 = (\mathcal{K}_2, \mathcal{K}_2^{\mathrm{rec}}, \cdot, =)$ of Kleene's second model. There is a partial applicative morphism $g\colon \mathcal{K}_2 \rightharpoonup \mathcal{K}_1$ defined by $g(\alpha) = \{e \in \mathbb{N} \mid \varphi_e = \alpha\}$. Working this out involves showing that g preserves application up to a realizer. This means we should, given indices $e, e' \in \mathbb{N}$ of total recursive functions $\alpha, \alpha' \in \mathbb{N}^{\mathbb{N}}$ such that $\alpha\alpha' \downarrow$, find an index of $\alpha\alpha'$ recursively in e and e'. This is an easy exercise in recursion theory, and we omit it.

Recall from Example 2.2.20 that there always exists a morphism $\mathcal{K}_1 \to A$ sending n to its numeral in A. In the case where $A = \mathcal{K}_2$, this morphism is isomorphic to the function $f\colon \mathcal{K}_1 \to \mathcal{K}_2$ that sends n to the constant function with value n, which we denote by $\hat{n}$. Viewing f as a projective applicative morphism $\mathcal{K}_1 \multimap \mathcal{K}_2$, we have $f \dashv g$, and in particular, f is c.d. First of all, note that $gf(n)$ is the set of indices of the constant function $\hat{n}$. Using this description, it is clear that $gf \simeq \mathrm{id}_{\mathcal{K}_1}$. For the other inequality, find a total recursive funtion ρ such that: if u is a coded sequence of natural numbers of length n, then:

- $\rho(u) = 0$ if $n < 2$, or the computation of $\varphi_{u_1}(u_0)$ has not halted after at most n steps;
- $\rho(u) = \varphi_{u_1}(u_0) + 1$ if the computation of $\varphi_{u_1}(u_0)$ has halted after at most n steps.

Then, if e is a recursive index of the function $\alpha \in \mathbb{N}^{\mathbb{N}}$, we easily see that $\rho \cdot \hat{e} = \alpha$, which means that $\rho \in \mathcal{K}_2^{\mathrm{rec}}$ realizes $fg \leq \mathrm{id}_{\mathcal{K}_2}$.

For $\mathcal{B} = (\mathcal{B}, \mathcal{B}^{\mathrm{pr}}, \cdot, =)$, we similarly have $g\colon \mathcal{B} \rightharpoonup \mathcal{K}_1$, which is defined by $g(\alpha) = \{e \in \mathbb{N} \mid \varphi_e = \alpha\}$. This partial applicative morphism is also right adjoint to the projective applicative morphism $f\colon \mathcal{K}_1 \multimap \mathcal{B}$ given by sending n to $\hat{n}$. In fact, here is is a bit easier to construct a realizer of $fg \leq \mathrm{id}_{\mathcal{B}}$. Indeed, we can simply take a partial recursive function ρ satisfying $\rho([u]) = 1$ and $\rho([u, v]) \simeq 2 \cdot \varphi_v(u)$.

This example even works for the ordered version $\mathcal{B} = (\mathcal{B}, \mathcal{B}^{\mathrm{pr}}, \cdot, \supseteq)$ of the Van Oosten model, but we should adjust the definition of $g(\alpha)$ to $\{e \in \mathbb{N} \mid \varphi_e \supseteq \alpha\}$. We will generalize this example to arbitrary PCAs in Chapter 5.

Example 2.3.22. Consider Kleene's second model $\mathcal{K}_2 = (\mathcal{K}_2, \mathcal{K}_2^{\mathrm{rec}}, \cdot, =)$ and Scott's graph model $\mathcal{P}\omega = (\mathcal{P}\omega, (\mathcal{P}\omega)^{\mathrm{re}}, \cdot, =)$. There is an applicative morphism $g\colon \mathcal{P}\omega \to \mathcal{K}_2$ defined by: $g(A) = \{\alpha \in \mathcal{K}_2 \mid A = \{n \mid n+1 \in \mathrm{Im}\,\alpha\}\}$. Informally, $g(A)$ consists of all the functions whose positive outcomes 'enumerate' A. If A is r.e., then we may clearly find a total recursive function that does this. So in order to show that g is an applicative morphism, we should show that application is preserved up to a realizer. We describe such a realizer informally. Suppose that we are given α, β such that $A = \{n \mid n+1 \in \mathrm{Im}\,\alpha\}$ and $B = \{n \mid n+1 \in \mathrm{Im}\,\beta\}$. The task is to construct a function whose positive values enumerate AB. On input $\langle x, y\rangle$, request the first x values of α and the first y values of β. Then we check whether $\alpha(x-1)$ is of the form $\langle n, m\rangle + 1$. If not, output 0. If so, check whether $e_n \subseteq \{k \mid k+1 \in \{\beta(0), \ldots, \beta(y-1)\}\}$. If not, output 0, but if so, output $m+1$.

Recall from Example 2.2.21 that we have a morphism of PCAs $f\colon \mathcal{K}_2 \to \mathcal{P}\omega$ sending a function $\alpha \in \mathcal{K}_2$ to its graph. Considering f as a projective applicative morphism, we have $f \dashv g$. Given $\alpha \in \mathcal{K}_2$, the set $gf(\alpha)$ consists of all functions whose positive values enumerate the graph of α. We have $\mathrm{id}_{\mathcal{K}_2} \leq gf$, since, given α and input n, we can request the first $n+1$ values of α and output $\langle n, \alpha(n)\rangle + 1$. Conversely, given a function β such that $\{n \mid n+1 \in \mathrm{Im}\,\beta\}$ is the graph of α, we can retrieve α. Indeed, on input n, keep requesting values of β until you hit a value of the form $\langle n, m\rangle + 1$, and then output m. So we have $gf \simeq \mathrm{id}_{\mathcal{K}_2}$. For the other inequality, let B be the (recursive) set $\{\langle e_{\{\langle x,y\rangle\}}, y-1\rangle \mid y > 0\}$. Then we easily see that $B \cdot f(\alpha) = A$ for any $\alpha \in g(A)$, meaning that B realizes $fg \leq \mathrm{id}_{\mathcal{P}\omega}$.

For the Van Oosten model $(\mathcal{B}, \mathcal{B}^{\mathrm{pr}}, \cdot, =)$, we similarly have $g\colon \mathcal{P}\omega \multimap \mathcal{B}$ which is right adjoint to $f\colon \mathcal{B} \to \mathcal{P}\omega$ from Example 2.2.21. Since the functions in $\mathcal{B}$ are partial, we can simply put $g(A) = \{\alpha \in \mathcal{B} \mid A = \mathrm{Im}\,\alpha\}$. We leave the verification of this example to the reader. It is worth noting that in this case, we do *not* have $gf \leq \mathrm{id}_{\mathcal{B}}$.

2.4 Products and coproducts

In this section, we investigate the existence of pseudoproducts and pseudocoproducts in the two categories of PCAs of primary interest, namely OPCA and OPCA_T. We will show that OPCA has small 2-products and finite pseudobiproducts. The category OPCA_T, on the other hand, has finite pseudocoproducts, but no nontrivial pseudoproducts.

2.4.1 Products and coproducts in OPCA

We start with generalizing a result by J. Longley [Lon94, Proposition 2.1.7] to the ordered setting.

Proposition 2.4.1. *The category* OPCA *has a pseudozero object.*

Proof. The required pseudozero object is the absolute discrete PCA $\mathbf{1} = \{*\}$, where $** = *$. For every PCA A, there is only one function $!\colon A \to \mathbf{1}$, and this is clearly a morphism of PCAs, so $\mathbf{1}$ is in fact a 2-terminal object. Conversely, every element $r \in A^{\#}$ yields a morphism of PCAs $¡\colon 1 \to A$ with $¡(*) = r$. Clearly, these are all isomorphic, so $\mathbf{1}$ is also a pseudoinitial object. □

The existence of a pseudozero object means that we also have *zero morphisms*.

Definition 2.4.2. *A morphism of PCAs $A \to B$ is called a **zero morphism** if it factors, up to isomorphism, through* $\mathbf{1}$.

The following lemma provides two alternative characterizations of zero morphisms, which shows that the morphisms from Example 2.2.16 are exactly the zero morphisms. We leave the proof to the reader.

Lemma 2.4.3. *For a morphism of PCAs $f\colon A \to B$, the following are equivalent:*

(i) f is a zero morphism;

(ii) there exists an $r \in B^{\#}$ such that $r \leq f(a)$ for all $a \in A$;

(iii) f is a top element of $\mathsf{OPCA}(A, B)$.

It follows from (iii) that OPCA is even enriched over preorders with a top element. Before we continue, we characterize the PCA $\mathbf{1}$ up to equivalence in a number of ways.

Lemma 2.4.4. *Let A be a PCA. The following are equivalent:*

(i) A is equivalent to $\mathbf{1}$;

(ii) A is absolute and has a least element;

(iii) id_A is a zero morphism;

(iv) $¡\colon \mathbf{1} \to A$ is dense.

Definition 2.4.5. *A PCA A is **trivial** if it satisfies the equivalent conditions of Lemma 2.4.4.*

If A is a PCA, then $! \circ ¡$ is isomorphic to the identity $\mathrm{id}_{\mathbf{1}}$. On the other hand, $¡ \circ !$ is, by definition, a zero morphism, so we also have $\mathrm{id}_A \leq ¡ \circ !$. This means that $! \dashv ¡$.

In [HvO03, Remark (2) on p.450], it is observed that OPCA has binary products. This construction generalizes to products of arbitrary (small) size, given choice on the index set.

Proposition 2.4.6. *The category OPCA has small pseudoproducts.*

Proof. Suppose we have an I-indexed family of PCAs $(A_i)_{i\in I}$. We make the product $A = \prod_{i\in I} A_i$ into a PAS by defining all the structure coordinatewise. That is, if $a = (a_i)_{i\in I}$ and $b = (b_i)_{i\in I}$ are elements of A, then we set:

- $a \leq b$ iff $a_i \leq b_i$ for all $i \in I$;
- $ab{\downarrow}$ iff $a_ib_i{\downarrow}$ for all $i \in I$, and in this case, $ab = (a_ib_i)_{i\in I}$;
- $a \in A^{\#}$ iff $a_i \in A_i^{\#}$ for all $i \in I$, that is, $\left(\prod_{i\in I} A_i\right)^{\#} = \prod_{i\in I} A_i^{\#}$.

Axioms (A)-(C) clearly hold for A, since they hold coordinatewise. For all $i \in I$, we may (using AC) pick suitable combinators k_i and s_i for A_i. Then it is not hard to check that $\mathsf{k} = (\mathsf{k}_i)_{i \in I}$ and $\mathsf{s} = (\mathsf{s}_i)_{i \in I}$ are suitable combinators for A, so A is a PCA.

It is easy to show that the projections $\pi_i \colon A \to A_i$ are morphisms of PCAs, and to verify that this makes A even into the 2-product of the A_i. We leave this to the reader. □

If $f_i \colon B \to A_i$ are morphisms of PCAs, then we denote their amalgamation by $\langle f_i \rangle_{i \in I}$. The projections π_i are clearly dense, so if an amalgamation $\langle f_i \rangle_{i \in I}$ is dense, then so are all the f_i, and similarly for decidability and computational density. The following proposition is a partial converse to this.

Proposition 2.4.7. *Let $(A_i)_{i \in I}$ be an indexed family of PCAs. Suppose that we have a morphism of PCAs $f_i \colon B \to A_i$ for each $i \in I$, and denote their amalgamation $B \to \prod_{i \in I} A_i$ by f.*

(i) If all the f_i are decidable, then so is f.

(ii) If I is finite and all the f_i are c.d. / dense, then f is also c.d. / dense.

Proof. (i) Using a decider for f_i, we can construct an element $d_i \in A_i^{\#}$ such that $d_i \cdot f_i(\top) \leq \pi_i(\top)$ and $d_i \cdot f_i(\bot) \leq \pi_i(\bot)$. Now we easily see that $(d_i)_{i \in I}$ is a decider for f.

(ii) It suffices to treat the nullary and the binary case. The nullary case states that $! \colon B \to \mathbf{1}$ is always dense, which follows from the adjunction $! \dashv$ ¡.

For the binary case, suppose we have c.d. morphisms $f_0 \colon B \to A_0$ and $f_1 \colon B \to A_1$. Let $t_i, u_i \in A_i^{\#}$ realize f_i, and let $n_i \in A_i^{\#}$ satisfy (cd) for f_i. We define $n_i' = \lambda^* x. n_i(u_i(t_i \cdot f_i(\mathsf{p}_i) \cdot x)) \in A_i$. We claim that $n = (n_0', n_1') \in (A_0 \times A_1)^{\#}$ satisfies (cd) for $f \colon B \to A_0 \times A_1$.

In order to prove this, let $s = (s_0, s_1) \in (A_0 \times A_1)^{\#} = A_0^{\#} \times A_1^{\#}$. Then we know that there exist $r_i \in B^{\#}$ such that $n_i \cdot f_i(r_i) \leq s_i$. Now define $r = \mathsf{p} r_0 r_1 \in B^{\#}$. Then

$$n_i' \cdot f_i(r) \preceq n_i(u_i(t_i \cdot f_i(\mathsf{p}_i) \cdot f_i(r))) \preceq n_i(u_i \cdot f_i(\mathsf{p}_i r)) \preceq n_i \cdot f(r_i) \leq s_i,$$

so $n \cdot f(r) \leq s$, as desired. The proof for density is analogous. □

Example 2.4.8. Let A be a PCA that is not semitrivial, and let I be a set such that $2^{|I|} > |A^{\#}|$. Then a morphism $f \colon A \to A^I$ is never c.d., where A^I denotes the I-fold product of A. Indeed, suppose for the sake of contradiction that f is c.d., witnessed by $n \in (A^{\#})^I$. Then every element of $(A^{\#})^I$ is bounded from below by an element of $X = \{n \cdot f(r) \mid r \in A^{\#}, n \cdot f(r)\downarrow\}$. This set X has cardinality at most $|A^{\#}|$. However, the subset

$$\{a \in (A^{\#})^I \mid \forall i \in I \, (a_i \in \{\top, \bot\})\}$$

of $(A^{\#})^I$, which has cardinality $2^{|I|} > |A^{\#}| \geq |X|$, has the property that every two distinct elements do not have a common lower bound in A^I: contradiction.

In particular, the diagonal $\delta\colon A \to A^I$ is not c.d., hence also not dense, which means that Proposition 2.4.7(ii) does not hold for infinite I, for either density or computational density.

Just as the 2-terminal object $\mathbf{1}$ is also pseudoinitial, *finite* 2-products in OPCA also serve as pseudocoproducts.

Theorem 2.4.9. *The category* OPCA *has finite pseudocoproducts.*

Proof. It suffices to treat the binary case. Let A_0 and A_1 be PCAs. Then there is a morphism of PCAs $\kappa_0\colon A_0 \to A_0 \times A_1$ given by $\kappa_0(a) = (a, \mathsf{i})$. Similarly, we have $\kappa_1\colon A_1 \to A_0 \times A_1$ given by $\kappa_1(a) = (\mathsf{i}, a)$. We claim that this is a pseudocoproduct diagram. This means that we should show that the map

$$(- \circ \kappa_0, - \circ \kappa_1)\colon \mathsf{OPCA}(A_0 \times A_1, B) \to \mathsf{OPCA}(A_0, B) \times \mathsf{OPCA}(A_1, B).$$

is an equivalence of preorders, for each PCA B. If suffices to prove that this map is essentially surjective and full; it is automatically faithful.

For essential surjectivity, suppose we have morphisms of PCAs $f_0\colon A_0 \to B$ and $f_1\colon A_1 \to B$. Let $t_i, u_i \in B^\#$ realize f_i. We define $f = [f_0, f_1]\colon A_0 \times A_1 \to B$ by $f(a_0, a_1) = \mathsf{p}\cdot f_0(a_0)\cdot f_1(a_1)$. Then f preserves application up to the following element of $B^\#$:

$$\lambda^* xy.\mathsf{p}(t_0(\mathsf{p}_0 x)(\mathsf{p}_0 y))(t_1(\mathsf{p}_1 x)(\mathsf{p}_1 y)).$$

The calculation is straightfowrward, and we omit it. Similarly, one can show that f preserves the order up to the element $\lambda^* x.\mathsf{p}(u_0(\mathsf{p}_0 x))(u_1(\mathsf{p}_1 x))$ of $B^\#$. Finally, we clearly have $f(a_0, a_1) \in B^\#$ if $(a_0, a_1) \in (A_0 \times A_1)^\#$, so f is a morphism of PCAs. We have $f(\kappa_0(a)) = \mathsf{p}a\mathsf{i}$, so $\mathsf{p}_0 \in B^\#$ realizes $f\kappa_0 \le f_0$ and $\lambda^* x.\mathsf{p}x\mathsf{i} \in B^\#$ realizes $f_0 \le f\kappa_0$. Similarly, one shows that $f\kappa_1 \simeq f_1$.

For fullness, suppose we have morphisms $g, g'\colon A_0 \times A_1 \to B$ such that $g\kappa_0 \le g'\kappa_0$ and $g\kappa_1 \le g'\kappa_1$. Let $s_i \in B^\#$ realize $g\kappa_i \le g'\kappa_i$, let $t, u \in B^\#$ realize g, and let $t', u' \in B^\#$ realize g'. We claim that $g \le g'$ is realized by $s \in B^\#$ defined as:

$$\lambda^* x.u'(t'(t' \cdot g'(\mathsf{k}, \overline{\mathsf{k}}) \cdot (s_0(u(t \cdot g(\mathsf{i}, \mathsf{ki}) \cdot x))))(s_1(u(t \cdot g(\mathsf{ki}, \mathsf{i}) \cdot x)))).$$

Let $(a_0, a_1) \in A_0 \times A_1$. Then we have:

$$\begin{aligned} s_0(u(t \cdot g(\mathsf{i}, \mathsf{ki}) \cdot g(a_0, a_1))) &\preceq s_0(u \cdot g(\mathsf{i}a_0, \mathsf{ki}a_1)) \\ &\preceq s_0 \cdot g(a_0, \mathsf{i}) \\ &\simeq s_0 \cdot g(\kappa_0(a_0)) \\ &\le g'(\kappa_0(a_0)) \\ &= g'(a_0, \mathsf{i}), \end{aligned}$$

and similarly, $s_1(u(t \cdot g(\mathsf{ki}, \mathsf{i}) \cdot g(a_0, a_1))) \le g'(\mathsf{i}, a_1)$. This yields:

$$\begin{aligned} s \cdot g(a_0, a_1) &\preceq u'(t'(t' \cdot g'(\mathsf{k}, \overline{\mathsf{k}}) \cdot g'(a_0, \mathsf{i})) \cdot g'(\mathsf{i}, a_1)) \\ &\preceq u'(t' \cdot g'(\mathsf{k}a_0, \overline{\mathsf{k}}\mathsf{i}) \cdot g'(\mathsf{i}, a_1)) \\ &\preceq u' \cdot g'(\mathsf{k}a_0\mathsf{i}, \overline{\mathsf{k}}\mathsf{i}a_1) \\ &\le g'(a_0, a_1), \end{aligned}$$

as desired. □

Corollary 2.4.10. *The category* OPCA *has finite pseudobiproducts*

Proof. The only thing left to check is that $A_0 \xrightarrow{\kappa_0} A_0 \times A_1 \xrightarrow{\pi_0} A_0$ is isomorphic to id_{A_0}, and that $A_0 \xrightarrow{\kappa_0} A_0 \times A_1 \xrightarrow{\pi_1} A_1$ is a zero morphism. Both are immediate. □

Note that Lemma 2.2.10(ii) immediately yields the following.

Corollary 2.4.11. *If $f_0\colon A_0 \to B$ and $f_1\colon A_1 \to B$ are morphisms of PCAs and f_0 is decidable / c.d. / dense, then $[f_0, f_1]\colon A_0 \times A_1 \to B$ is also decidable / c.d. / dense.*

In analogy with ordinary coproducts, we say that finite pseudocoproducts are *disjoint* if, for every pseudocoproduct diagram $A_0 \to A_0 \sqcup A_1 \leftarrow A_1$, the coprojections are pseudomonos, and

$$\begin{array}{ccc} 0 & \longrightarrow & A_1 \\ \downarrow & \searrow & \downarrow \\ A_0 & \longrightarrow & A_0 \sqcup A_1 \end{array}$$

is a pseudopullback, where 0 denotes the pseudoinitial object.

Proposition 2.4.12. *The finite pseudocoproducts in* OPCA *are disjoint.*

Proof. Since $\pi_i\kappa_i \simeq \mathrm{id}_{A_i}$, it is immediate that the κ_i are pseudomonos. In order to establish the required pseudopullback, we need to show the following: if we have morphisms $f_0\colon B \to A_0$ and $f_1\colon B \to A_1$ such that $\kappa_0 f_0 \simeq \kappa_1 f_1$, then f_0 and f_1 are both zero morphisms. Let $s = (s_0, s_1) \in A_0^{\#} \times A_1^{\#}$ realize $\kappa_0 f_0 \leq \kappa_1 f_1$. Then for all $b \in B$, we have $(s_0 \cdot f_0(b), s_1\mathsf{i}) \simeq s \cdot \kappa_0(f_0(b)) \leq \kappa_1(f_1(b)) = (\mathsf{i}, f_1(b))$. In particular, we have $s_1\mathsf{i} \leq f_1(b)$ for all $b \in B$, so since $s_1\mathsf{i} \in B^{\#}$, we get that f_1 is a zero morphism. The proof that f_0 is a zero morphism proceeds analogously. □

The 'dual' result to Proposition 2.4.12 also holds; this will be useful in Section 2.4.2.

Proposition 2.4.13. *If A_0 and A_1 are PCAs, then $\pi_i\colon A_0 \times A_1 \to A_i$ is a pseudoepi and*

$$\begin{array}{ccc} A_0 \times A_1 & \longrightarrow & A_1 \\ \downarrow & \searrow & \downarrow \\ A_0 & \longrightarrow & \mathbf{1} \end{array}$$

is a pseudopushout diagram.

Proof. Since $\pi_i\kappa_i \simeq \mathrm{id}_{A_i}$, we know that π_i is indeed pseudoepi.

For the pseudopushout, we need to show the following: if $f_0 \colon A_0 \to B$ and $f_1 \colon A_1 \to B$ are morphisms such that $f_0\pi_0 \simeq f_1\pi_1$, then f_0 and g_0 are both zero morphisms. If $s \in B^{\#}$ realizes $f_0\pi_0 \leq f_1\pi_1$, then we have $s \cdot f_0(a_0) \leq f_1(a_1)$ for all $a_0 \in A_0$ and $a_1 \in A_1$. In particular, we have $s \cdot f_0(\mathsf{i}) \leq f_1(a_1)$ for all $a_1 \in A_1$, so f_1 is a zero morphism. The proof that f_0 is a zero morphism again proceeds analogously. □

We close this section by investigating coproducts in a category related to OPCA.

Definition 2.4.14. *The preorder-enriched category* $\mathsf{OPCA}_{\mathrm{adj}}$ *is defined as follows.*

- *Its objects are PCAs.*
- *An arrow* $f \colon A \to B$ *is a pair of morphisms* $f^* \colon B \to A$ *and* $f_* \colon A \to B$ *with* $f^* \dashv f_*$*.*
- *If* $f, g \colon A \to B$*, then we say that* $f \leq g$ *if* $f^* \leq g^*$*; equivalently, if* $g_* \leq f_*$*.*

Proposition 2.4.15. *The category* $\mathsf{OPCA}_{\mathrm{adj}}$ *has finite pseudocoproducts. Moreover, the pseudoinitial object is strict, and pseudocoproducts are disjoint.*

Proof. We have already seen that there are essentially unique morphisms $! \colon A \to \mathbf{1}$ and $¡ \colon \mathbf{1} \to A$ satisfying $! \dashv ¡$, yielding the (essentially) unique arrow $\mathbf{1} \to A$ in $\mathsf{OPCA}_{\mathrm{adj}}$. Moreover, if we have an arrow $A \to \mathbf{1}$ in $\mathsf{OPCA}_{\mathrm{adj}}$, then also $¡ \dashv !$, so $!$ and $¡$ form an equivalence between A and $\mathbf{1}$, meaning that $\mathbf{1}$ is indeed strict.

Now consider two PCAs A and B. We have the product and coproduct diagrams:

$$A \xleftarrow{\pi_A} A \times B \xrightarrow{\pi_B} B \qquad\qquad A \xrightarrow{\kappa_A} A \times B \xleftarrow{\kappa_B} B$$

We have already remarked that $\pi_A\kappa_A \simeq \mathrm{id}_A$. Moreover, it is easily computed that $\mathrm{id}_{A\times B} \leq \kappa_A\pi_A$, which means that $\pi_A \dashv \kappa_A$ is an arrow $A \to A \times B$ of $\mathsf{OPCA}_{\mathrm{adj}}$. Similarly, we have the arrow $\pi_B \dashv \kappa_B \colon B \to A \times B$. In order to show that this yields a pseudocoproduct diagram in $\mathsf{OPCA}_{\mathrm{adj}}$, we need to show the following: if $f \colon A \to C$ and $g \colon B \to C$ are arrows of $\mathsf{OPCA}_{\mathrm{adj}}$, then $h^* = \langle f^*, g^*\rangle$ is left adjoint to $h_* = [f_*, g_*]$. First of all, we may easily compute that $h_*(h^*(c)) = \mathsf{p} \cdot f_*(f^*(c)) \cdot g_*(g^*(c))$. So, if $r, s \in C^{\#}$ realize $\mathrm{id}_C \leq f_*f^*$ and $\mathrm{id}_C \leq g_*g^*$ respectively, then $\lambda^* x.\mathsf{p}(rx)(sx) \in C^{\#}$ realizes $\mathrm{id}_C \leq h_*h^*$. The other inequality can be obtained completely from universal properties. We have (recall that $\pi_A\kappa_B$ is a zero morphism):

$$\pi_A h^* h_* \kappa_A \simeq f^* f_* \leq \mathrm{id}_A \simeq \pi_A\kappa_A \quad \text{and} \quad \pi_A h^* h_* \kappa_B \simeq f^* g_* \leq \pi_A\kappa_B,$$

so from the universal property of the coproduct $A \times B$, it follows that $\pi_A h^* h_* \leq \pi_A$. Similarly, we obtain $\pi_B h^* h_* \leq \pi_B$, and the universal property of the product $A \times B$ yields $h^* h_* \leq \mathrm{id}_{A\times B}$, as desired.

For disjointness, we first note that $\pi_A \dashv \kappa_A$ is a pseudomono because $\pi_A\kappa_A \simeq \mathrm{id}_A$. Now suppose we have arrows $f\colon C \to A$ and $g\colon C \to B$ of $\mathsf{OPCA}_{\mathrm{adj}}$ such that $\kappa_A f_* \simeq \kappa_B g_*$. Then we know from Proposition 2.4.12 that f_* and g_* are both zero morphisms. From $\mathrm{id}_C \geq f^* f_*$, it follows that id_C is also a zero morphism, i.e., C is trivial. Now it is immediate that $\mathbf{1}$ is the pseudopullback of $A \to A \times B \leftarrow B$ in $\mathsf{OPCA}_{\mathrm{adj}}$. □

For a PCA A, the codiagonal $\varepsilon\colon A\times A \to A$ can be defined as $\varepsilon(a, a') = \mathsf{p}aa'$. Proposition 2.4.15 tells us in particular that ε is right adjoint to the diagonal $\delta\colon A \to A \times A$. Together with the fact that $!\colon A \to 1$ has a right adjoint ¡, we deduce that every PCA is a cartesian object in the preorder-enriched category OPCA (compare with the internal finite meets of BCOs in [Hof06, p.246]). Moreover, if $f, g\colon A \to B$ are morphisms of PCAs, then the composition

$$A \xrightarrow{\langle f,g\rangle} B \times B \xrightarrow{\varepsilon_B} B$$

is readily seen to be the meet of f and g in $\mathsf{OPCA}(A, B)$. From this, it is easy to deduce that OPCA is even enriched over preorders with finite meets.

Remark 2.4.16. We have seen that OPCA is enriched over preorders, preorders with a top element, and preorders with finite meets. For pseudo(co)limits in OPCA, it does not matter which of these enrichments we consider. The reason for this is that all these enrichments equip the homsets with the same *structure* (namely, a preorder), and differ only in which *properties* they ascribe to this structure.

2.4.2 Products and coproducts in OPCA_T

In this section, we investigate to what extent the results from Section 2.4.1 carry over to the category OPCA_T. For pseudocoproducts, this is quite easy.

Corollary 2.4.17. *The pseudofunctor $\mathsf{OPCA} \to \mathsf{OPCA}_T$ preserves finite pseudocoproducts. In particular, OPCA_T has all finite pseudocoproducts.*

Proof. For every PCA A, we have $\mathsf{OPCA}_T(\mathbf{1}, A) \simeq \mathsf{OPCA}(\mathbf{1}, TA)$, which we know to be equivalent to the one-element preorder. Similarly, if A_0, A_1 and B are PCAs, then

$$\begin{aligned}\mathsf{OPCA}_T(A_0 \times A_1, B) &\simeq \mathsf{OPCA}(A_0 \times A_1, TB)\\ &\simeq \mathsf{OPCA}(A_0, TB) \times \mathsf{OPCA}(A_1, TB)\\ &\simeq \mathsf{OPCA}_T(A_0, B) \times \mathsf{OPCA}_T(A_1, B),\end{aligned}$$

finishing the proof. □

Explicitly, if $f_0\colon A_0 \multimap B$ and $f_1\colon A_1 \multimap B$ are applicative morphisms, then the mediating arrow $[f_0, f_1]\colon A_0 \times A_1 \multimap B$ is given by:

$$[f_0, f_1](a_0, a_1) = \mathsf{p} \cdot f_0(a_0) \cdot f_1(a_1) = {\downarrow}\{\mathsf{p}b_0b_1 \mid b_0 \in f_0(a_0) \text{ and } b_1 \in f_1(a_1)\}.$$

By Lemma 2.2.10(ii) (or rather, its counterpart for OPCA_T), we immediately have the following corollary.

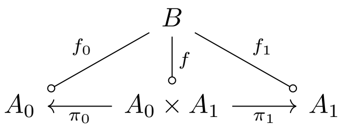
Corollary 2.4.18. *If $f_0\colon A_0 \multimap B$ and $f_1\colon A_1 \multimap B$ are applicative morphisms and f_0 is decidable / c.d. / dense, then $[f_0, f_1]\colon A_0 \times A_1 \multimap B$ is also decidable / c.d. / dense.*

Since $T\mathbf{1} \simeq \mathbf{1}$, we have that $\mathbf{1}$ is not only pseudoinitial in OPCA_T, but also 2-terminal. Therefore, we also define zero morphisms in OPCA_T, by saying that $f\colon A \multimap B$ is a zero morphism iff it factors (in OPCA_T) through $\mathbf{1}$. This is in fact equivalent to $f\colon A \to TB$ being a zero morphism in OPCA, which is equivalent to $B^\# \cap \bigcap_{a\in A} f(a) \neq \emptyset$. The proof of the following proposition is now completely analogous to the proof Proposition 2.4.12, and is therefore omitted.

Proposition 2.4.19. *Pseudocoproducts in OPCA_T are disjoint.*

If we want to show that $A_0 \times A_1$ is also the pseudoproduct of A_0 and A_1 in OPCA_T, then we should show that $T(A_0 \times A_1) \simeq TA_0 \times TA_1$. However, it turns out that this is *not* true in general, and that OPCA_T does not have finite pseudoproducts. On the other hand, $A_0 \times A_1$ is still a product of A_0 and A_1 in OPCA_T in a weak sense. Explicitly, if $f_0\colon B \multimap A_0$ and $f_1\colon B \multimap A_1$, then there exists a *largest* mediating arrow $f\colon B \multimap A_0 \times A_1$. Using the theory developed in Section 2.4.1, we can tie things together quite nicely in the following proposition.

Proposition 2.4.20. *If $f_0\colon B \multimap A_0$ and $f_1\colon B \multimap A_1$ are applicative morphisms, then there exists a* largest *mediating applicative morphism $f\colon B \multimap A_0 \times A_1$:*



Moreover, if $g, g'\colon B \multimap A_0 \times A_1$ are such that $\pi_i g \leq \pi_i g'$ for $i = 0, 1$, and g' is projective, then $g \leq g'$.

Proof. First of all, let us reformulate the proposition in terms of the category OPCA:

(i) Given $f_0\colon B \to TA_0$ and $f_1\colon B \to TA_1$, there should exist a largest $f\colon B \to T(A_0 \times A_1)$ such that $T\pi_i \circ f \simeq f_i$ for $i = 0, 1$.

(ii) Moreover, if we have $g, g'\colon A \to T(A_0 \times A_1)$ such that $T\pi_i \circ g \leq T\pi_i \circ g'$ for $i = 0, 1$, and g' factors though $\delta_{A_0\times A_1}$, then $g \leq g'$.

Because T is a pseudofunctor, we have arrows

$$T\pi_0 \dashv T\kappa_0\colon TA_0 \to T(A_0 \times A_1) \quad \text{and} \quad T\pi_1 \dashv T\kappa_1\colon TA_1 \to T(A_0 \times A_1)$$

of $\mathsf{OPCA}_{\mathrm{adj}}$. By Proposition 2.4.15, there exists an essentially unique mediating arrow

$$h^* \dashv h_*\colon TA_0 \times TA_1 \to T(A_0 \times A_1),$$

where:

$$\begin{aligned} h_* &= [T\kappa_0, T\kappa_1]\colon TA_0 \times TA_1 \to T(A_0 \times A_1), \\ h^* &= \langle T\pi_0, T\pi_1\rangle\colon T(A_0 \times A_1) \to TA_0 \times TA_1. \end{aligned} \tag{2.6}$$

The direct image part h_* is more conveniently described by $h_*(\alpha_0, \alpha_1) = \alpha_0 \times \alpha_1$. One easily computes that h^*h_* is in fact isomorphic to $\mathrm{id}_{TA_0 \times TA_1}$. (This also follows from the fact that $T\pi_i \circ T\kappa_i \simeq \mathrm{id}_{TA_i}$, whereas $T\pi_j \circ T\kappa_i$ is a zero morphism for $i \neq j$.)

In order to establish (i) above, we define f as the composition:

$$B \xrightarrow{\langle f_0, f_1\rangle} TA_0 \times TA_1 \xrightarrow{h_*} T(A_0 \times A_1)$$

Indeed, we have $h^*f = h^*h_* \circ \langle f_0, f_1\rangle \simeq \langle f_0, f_1\rangle$, and if $f'\colon B \to T(A_0 \times A_1)$ is any arrow such that $h^*f' \simeq \langle f_0, f_1\rangle$, then $f' \leq h_*h^*f' \simeq h_* \circ \langle f_0, f_1\rangle = f$.

For (ii), we suppose that $h^*g \leq h^*g'$, and that $g' \simeq \delta_{A_0 \times A_1} \circ g_0'$ for some $g_0\colon B \to A_0 \times A_1$. It is easily verified that $\delta_{A_0 \times A_1} \simeq h_* \circ (\delta_{A_0} \times \delta_{A_1})$, so g' factors through h^*, which implies that $h_*h^*g' \simeq g'$. Now we obtain $g \leq h_*h^*g \leq h_*h^*g' \simeq g'$, as desired. □

Now let us turn to the existence of genuine pseudoproducts in OPCA_T. Obviously, if A_0 (resp. A_1) is trivial, then the pseudoproduct of A_0 and A_1 exists in OPCA_T, and it is equivalent to A_1 (resp. A_0). Using the morphism h^* above, we can show that this is the *only* situation in which A_0 and A_1 have a product in OPCA_T.

Theorem 2.4.21. *If A_0 and A_1 are PCAs that have a pseudoproduct in OPCA_T, then at least one of A_0 and A_1 is trivial.*

Proof. The proof is divided into two parts.

1. First, we show that, if A_0 and A_1 have a pseudoproduct in OPCA_T, then $h^*\colon T(A_0 \times A_1) \to TA_0 \times TA_1$ as defined in (2.6) has a left adjoint.

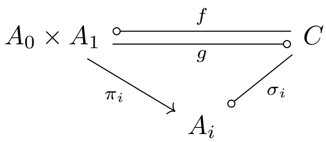
2. Second, we show that h^* cannot have a left adjoint if A_0 and A_1 are both nontrivial.

For the first part, denote the product projections $TA_0 \times TA_1 \to TA_i$ by ρ_i; then h^* is the essentially unique morphism of PCAs such that

$$\begin{array}{ccc} T(A_0 \times A_1) & \xrightarrow{\quad h^* \quad} & TA_0 \times TA_1 \\ & {\scriptstyle T\pi_i} \searrow \quad \swarrow {\scriptstyle \rho_i} & \\ & TA_i & \end{array}$$

commutes up to isomorphism, for $i = 0, 1$.

Suppose that C is a pseudoproduct of A_0 and A_1 in OPCA_T, with projections $\sigma_i\colon C \multimap A_i$. Then, by Proposition 2.4.20, σ_0 and σ_1 induce a maximal mediating applicative morphism $f\colon C \multimap A_0 \times A_1$. On the other hand, π_0 and π_1, seen as projective applicative morphisms, induce an essentially unique mediating applicative morphism $g\colon A_0 \times A_1 \multimap C$ by the universal property of the pseudoproduct C. So for $i = 0, 1$ we get a diagram in OPCA_T:



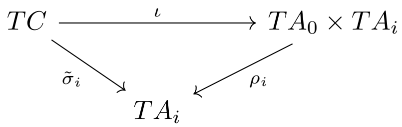
(2.7)

where the triangles commute up to isomorphism. Since C is a pseudoproduct, we have $gf \simeq \mathrm{id}_C$. Moreover, we have $\pi_i fg \simeq \sigma_i g \simeq \pi_i \simeq \pi_i \circ \mathrm{id}_{A_0\times A_1}$ for $i = 0, 1$, and since $\mathrm{id}_{A_0\times A_1}$ is certainly projective, this yields $fg \leq \mathrm{id}_{A_0\times A_1}$, by Proposition 2.4.20. We can conclude that $f \dashv g$.

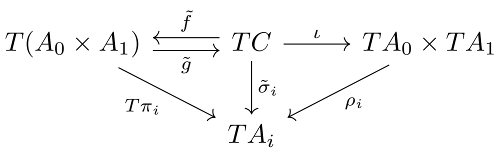
For every PCA B, we have natural equivalences

$$\begin{aligned}\mathsf{OPCA}(B, TC) &\simeq \mathsf{OPCA}_T(B, C)\\ &\simeq \mathsf{OPCA}_T(B, A_0) \times \mathsf{OPCA}_T(B, A_1)\\ &\simeq \mathsf{OPCA}(B, TA_0) \times \mathsf{OPCA}_T(B, TA_1),\end{aligned}$$

so $TA_0 \xleftarrow{\tilde\sigma_0} TC \xrightarrow{\tilde\sigma_1} TA_1$ is a product diagram in OPCA. This means there exists an equivalence $\iota\colon TC \to TA_0 \times TA_1$ such that the diagram



commutes up to isomorphism for $i = 0, 1$. Taking the image of the diagram (2.7) under the equivalence between OPCA_T and free T-algebras, we get the diagram



in OPCA for $i = 0, 1$, where all triangles commute up to isomorpism. In particular, $\rho_i \iota \tilde g \simeq \tilde\sigma_i \tilde g \simeq T\pi_i$, so $\iota\tilde g$ must be isomorphic to h^*. Since $f \dashv g$, we also have $\tilde f \dashv \tilde g$, hence also $\tilde f \iota^{-1} \dashv \iota \tilde g \simeq h^*$, which concludes the proof of the first part.

For the second part, suppose that A_0 and A_1 are both nontrivial, and that h^* has a left adjoint $h_!\colon TA_0 \times TA_1 \to T(A_0 \times A_1)$. Consider the set

$$X = \{\alpha \in T(A_0 \times A_1) \mid h^*(\alpha) = (A_0, A_1)\}.$$

We claim that $(A_0\times A_1)^\# \cap \bigcap X$ is empty. Let $(a_0, a_1) \in (A_0\times A_1)^\# = A_0^\# \times A_1^\#$ be arbitrary, and consider the downset

$$\alpha = \{(b_0, b_1) \in A_0 \times A_1 \mid a_0 \not\leq b_0 \text{ or } a_1 \not\leq b_1\}$$

of $A_0 \times A_1$. Since $a_0 \in A_0^\#$ and A_0 is nontrivial, we see that a_0 cannot be the least element of A_0, so there exists a $b_0 \in A_0$ such that $a_0 \not\leq b_0$. This implies that $\{b_0\} \times A_1 \subseteq \alpha$, so α is nonempty and satisfies $T\pi_1(\alpha) = A_1$. Similarly, we show that $T\pi_0(\alpha) = A_0$, so $\alpha \in X$. On the other hand, we clearly do *not* have $(a_0, a_1) \in \alpha$, so $(a_0, a_1) \notin \bigcap X$. Since this holds for all $(a_0, a_1) \in (A_0 \times A_1)^\#$, we can conclude that $(A_0 \times A_1)^\# \cap \bigcap X = \emptyset$.

Now let $s \in (A_0 \times A_1)^\#$ realize $h_! h^* \leq \mathrm{id}_{T(A_0 \times A_1)}$. Since (A_0, A_1) is clearly in $(TA_0 \times TA_1)^\#$, we must have $h_!(A_0, A_1) \in (T(A_0 \times A_1))^\#$. Now, if $\alpha \in X$, then

$$s \cdot h_!(A_0, A_1) \simeq s \cdot h_!(h^*(\alpha)) \subseteq \alpha.$$

The set X is clearly nonempty, so in particular, $s \cdot h_!(A_0, A_1)$ is defined and an element of $(T(A_0 \times A_1))^\#$. Moreover, the above shows that $s \cdot h_!(A_0, A_1) \subseteq \bigcap X$, so $\bigcap X \in (T(A_0 \times A_1))^\#$ as well. However, we also have $(A_0 \times A_1)^\# \cap \bigcap X = \emptyset$, which is a contradiction. This finishes the proof of the second part. □

We close this section by investigating, in analogy with $\mathsf{OPCA}_{\mathrm{adj}}$, the category $\mathsf{OPCA}_{T,\mathrm{adj}}$.

Definition 2.4.22. *The preorder-enriched category* $\mathsf{OPCA}_{T,\mathrm{adj}}$ *is defined as follows.*

- *Its objects are PCAs.*
- *An arrow* $f \colon A \to B$ *is a pair of applicative morphisms* $f^* \colon B \multimap A$ *and* $f_* \colon A \multimap B$ *with* $f^* \dashv f_*$.
- *If* $f, g \colon A \to B$, *then we say that* $f \leq g$ *if* $f^* \leq g^*$*; equivalently, if* $g_* \leq f_*$.

From Theorem 2.3.14, we know that $\mathsf{OPCA}_{T,\mathrm{adj}}$ is actually equivalent to $\mathsf{OPCA}_{\mathrm{dense}}^{\mathrm{op}}$, where $\mathsf{OPCA}_{\mathrm{dense}}$ denotes the wide subcategory of OPCA on the dense morphisms, and $(\cdot)^{\mathrm{op}}$ indicates a reversal of the *1-cells*. The following result is now immediate.

Corollary 2.4.23. *The category* $\mathsf{OPCA}_{T,\mathrm{adj}}$ *has finite pseudocoproducts. Moreover, the pseudoinitial object is strict, and pseudocoproducts are disjoint.*

Proof. It suffices to prove the dual statements in $\mathsf{OPCA}_{\mathrm{dense}}$. By Proposition 2.4.7, $\mathsf{OPCA}_{\mathrm{dense}}$ has *finite* pseudoproducts. Moreover, by Lemma 2.4.4, the terminal object is strict in $\mathsf{OPCA}_{\mathrm{dense}}$. The final statement follows from Proposition 2.4.13. □

Chapter 3

Assemblies and the Realizability Topos

In this chapter, we discuss the categories that may be constructed using partial combinatory algebras. As mentioned in the Introduction, each PCA A gives rise to a realizability topos $\mathsf{RT}(A)$ and a subcategory $\mathsf{Asm}(A)$ of $\mathsf{RT}(A)$ known as the category of assemblies. We introduce these in Section 3.1 and establish some of their properties in Section 3.2. In Section 3.3, we discuss results from [Lon94], [HvO03] and [FvO14] characterizing left exact and regular functors between categories of assemblies and realizability toposes. Finally, Section 3.4 treats geometric morphisms between realizability toposes.

This chapter contains two new improvements over the existing literature.

1. As in the previous chapter, we treat everything in the relative setting. As we will see below, this requires some adjustments compared to the absolute setting, especially in Section 3.3 and Section 3.4.

2. In Section 3.4.3, we describe when a geometric morphism induced by a c.d. applicative morphism is a geometric surjection. As far as we are aware, such a characterization has not occurred in the literature before.

3.1 Categories associated to PCAs

In this section, we introduce, for each PCA A, its category of assemblies, its realizability tripos and its realizability topos. Especially for the latter two, a much more detailed account can be found in, e.g., [vO08]. Here, we only discuss as much material as is needed for the rest of the thesis.

3.1.1 The category of assemblies

Intuitively, an assembly over a PCA A can be viewed as a datatype that has been implemented in A. More precisely, an assembly X will consist of a set $|X|$ and a function E_X that assigns to each $x \in |X|$ a nonempty downset $E_X(x) \in TA$. We think of the elements of $E_X(x)$ as those elements that 'represent' the element x inside A. This intuition should be familiar from the previous chapter. Indeed, an applicative morphism $A \multimap B$ is, among other things, an assembly over B with underlying set A. We will exploit this fact, in the slightly more general setting of partial applicative morphisms, in the proof of Theorem 3.3.13 below.

A morphism of assemblies $X \to Y$ will be a function $|X| \to |Y|$ between the underlying sets, for which there exists an algorithm that transforms representers of x into representers of $f(x)$. Let us make this precise in the following definition.

Definition 3.1.1. *Let A be a PCA.*

*(i) An **assembly** over A is a pair $X = (|X|, E_X)$, where $|X|$ is a set and $E_X \colon |X| \to TA$ is a function.*

*(ii) A **morphism of assemblies** $f \colon X \to Y$ is function $f \colon |X| \to |Y|$ for which there exists a $t \in A^\#$, called a **tracker** of f, such that $t \cdot E_X(x) \subseteq E_Y(f(x))$ for all $x \in |X|$.*

Recall from Notation 2.3.3 that $t \cdot E_X(x) \subseteq E_Y(f(x))$ means: for all $a \in E_X(x)$, we have that ta is defined and an element of $E_Y(f(x))$. As we mentioned in the Introduction, assemblies and morphisms between them form a category, which is a quasitopos $\mathsf{Asm}(A)$. In the following proposition, we prove part of this statement, namely $\mathsf{Asm}(A)$ is a regular category. The proof serves mainly to record the construction of finite limits and regular epimorphisms in $\mathsf{Asm}(A)$, as this will be important in Section 3.3 below.

Proposition 3.1.2. *Assemblies over A and morphisms between them form a regular category that we denote by $\mathsf{Asm}(A)$. Moreover, there is an adjunction*

$$\mathsf{Set} \underset{\nabla}{\overset{\Gamma}{\underset{\longrightarrow}{\overset{\longleftarrow}{\bot}}}} \mathsf{Asm}(A)$$

where Γ and ∇ are both regular, and $\Gamma\nabla = \mathrm{id}_{\mathsf{Set}}$.

Proof. If X is an assembly, then $\mathsf{i} \in A^\#$ tracks the identity on $|X|$ as a morphism $X \to X$. Moreover, if $X \xrightarrow{f} Y \xrightarrow{g} Z$ are tracked by $t, s \in A^\#$ respectively, then the composite function $gf \colon |X| \to |Z|$ is tracked as a morphism $X \to Z$ by $\lambda^* x.s(tx) \in A^\#$. This shows that $\mathsf{Asm}(A)$ is a category, and that there is a forgetful functor $\Gamma \colon \mathsf{Asm}(A) \to \mathsf{Set}$ with $\Gamma X = |X|$.

For a set Y, we define the assembly $\nabla Y = (Y, y \mapsto A)$. If X is an assembly, then every function $|X| \to Y$ is automatically a morphism of assemblies $X \to \nabla Y$, so we have

$$\mathsf{Set}(\Gamma X, Y) = \mathsf{Asm}(A)(X, \nabla Y).$$

This makes it clear that ∇ can be extended to a right adjoint of Γ, by letting $\nabla f = f$ for functions f. Moreover, we have $\Gamma\nabla = \mathrm{id}_{\mathsf{Set}}$ by definition.

Now let us turn to finite limits in $\mathsf{Asm}(A)$. First of all, it is clear from the above that $\nabla 1$ is a terminal object of $\mathsf{Asm}(A)$. If X and Y are assemblies, then their product is given by:

$$|X \times Y| = |X| \times |Y| \quad \text{and} \quad E_{X\times Y}(x, y) = \mathsf{p} \cdot E_X(x) \cdot E_Y(y).$$

If $f, g\colon A \to Y$ are parallel morphisms of assemblies, then their equailizer is given by $m\colon U \hookrightarrow X$, where $m\colon |U| \hookrightarrow |X|$ is the equalizer of $f, g\colon |X| \to |Y|$ in Set and E_U is the restriction of E_X to $|U|$, that is, $E_U(u) = E_X(m(u))$. From this description of finite limits, it is clear that Γ is left exact.

For regularity, we first note that $\mathsf{Asm}(A)$ has all coequalizers. Indeed, if $f, g\colon X \to Y$ are parallel morphisms of assemblies, then their coequalizer is given by $e\colon Y \twoheadrightarrow Q$, where $e\colon |Y| \to |Q|$ is the coequalizer of $f, g\colon |X| \to |Y|$ in Set, and $E_Q(q) = \bigcup_{e(y)=q} E_Y(y)$. From this construction of coequalizers, it is easy to see that in general, a morphism of assemblies $f\colon X \to Y$ is a regular epi iff there exists an $r \in A^{\#}$ such that for $r \cdot E_Y(y) \subseteq \bigcup_{f(x)=y} E_X(x)$ for all $y \in |Y|$. Note that the latter automatically implies that $f\colon |X| \to |Y|$ is surjective. Using this description of regular epis, it is easy to see that regular epis are stable under pullback, so $\mathsf{Asm}(A)$ is regular.

Finally, ∇f is clearly a regular epi whenever f is a surjective function. Since ∇, being a right adjoint, preserves finite limits, we see that ∇ is regular. Moreover, we have already remarked that Γ is left exact and since Γ is also a left adjoint, it follows that Γ is regular as well. $\square$

In particular, Proposition 3.1.2 tells us that ∇ is fully faithful. Moreover, it is clear that Γ is faithful, so Γ reflects monos and epis. Since Γ preserves finite limits and colimits, we know that Γ also preserves monos and epi. This means that $f\colon X \to Y$ is mono resp. epi iff $f\colon |X| \to |Y|$ is injective resp. surjective. Another consequence of Γ being faithful is that the unit of $\Gamma \dashv \nabla$ is mono at each coordinate. In fact, if X is an assembly, then the unit $X \hookrightarrow \nabla\Gamma X$ is the identity on $|X|$, which is certainly injective.

In the sequel, we will need the following two notions.

Definition 3.1.3. *Let A be a PCA.*

*(i) An assembly X is called a **constant object** if X is isomorphic to an object in the image of ∇. Accordingly, we call ∇ the **constant object functor**.*

*(ii) A morphism of assemblies $f\colon X \to Y$ is called **prone** if the naturality square*

$$\begin{array}{ccc} X & \xrightarrow{\;f\;} & Y \\ \downarrow & & \downarrow \\ \nabla\Gamma X & \xrightarrow{\;\nabla\Gamma f\;} & \nabla\Gamma Y \end{array}$$

is a pullback diagram.

Note that, if X is a constant object, then X must be isomorphic to $\nabla\Gamma X$. From this, it follows that X is constant iff the unit $X \hookrightarrow \nabla\Gamma X$ is an isomorphism. In particular, X is constant iff the unique morphism $X \to 1$ is prone. We will revisit this connection between constant objects and proneness when discussing slicing in Section 4.4.2. The following lemma gives a more direct description of the notions from Definition 3.1.3; the proof is left to the reader.

Lemma 3.1.4. *Let A be a PCA.*

(i) An assembly X is constant iff $\bigcap_{x\in|X|} E_X(x) \in (TA)^{\#}$.

*(ii) A morphism of assemblies $f\colon X \to Y$ is prone iff there exists a $t' \in A^{\#}$ such that $t' \cdot E_Y(f(x)) \subseteq E_X(x)$ for all $x \in |X|$. Such a t' will be called a **reverse tracker** of f.*

Using (ii), it is easy to see that the regular monos in $\mathsf{Asm}(A)$ are precisely the prone monos.

Finally, the following lemma records a fact noted in Section 1.1.4.

Lemma 3.1.5. *Let A be a PCA. Then $\Gamma\colon \mathsf{Asm}(A) \to \mathsf{Set}$ is isomorphic to the global sections functor iff A is an absolute PCA.*

Proof. Suppose that A is absolute. If X is an assembly, then every $x \in |X|$ determines a global section $1 \to X$, which is tracked by $\mathsf{k}a$, where a is any element of $E_X(x)$. This shows that Γ is isomorphic to the global sections functor.

Conversely, suppose that A is not absolute, and take $a \in A\backslash A^{\#}$. Then the assembly 1_a defined by $|1_a| = \{*\}$ and $E_{1_a}(*) = \downarrow\{a\}$ has no global sections. □

Example 3.1.6. Let us give a few important examples of assemblies.

(i) We have the *object of nonempty downsets* $T_A \in \mathsf{Asm}(A)$, defined by $|T_A| = TA$ and $E_{T_A}(\alpha) = \alpha$. Note that the global sections of TA correspond to the elements of $(TA)^{\#}$.

(ii) We have the *object of realizers* $R_A \in \mathsf{Asm}(A)$, defined by $|R_A| = A$ and $E_{R_A}(a) = \delta_A(a) = \downarrow\{a\}$. Note that R_A is a regular subobject of T_A, and that the global sections of R_A correspond to the elements of $A^{\#}$.

(iii) We have the assembly $N \in \mathsf{Asm}(A)$ defined by $|N| = \mathbb{N}$ and $E_N(n) = \downarrow\{\overline{n}\}$, which is a regular subobject of R_A. Using the recursor from Construction 2.1.31, one easily shows that N is a natural numbers object in $\mathsf{Asm}(A)$.

3.1.2 The realizability tripos

In the previous section, we have introduced the category of assemblies $\mathsf{Asm}(A)$. For introducing the realizability topos $\mathsf{RT}(A)$, there are two equivalent routes we may take. As we explained in the introduction, the realizability topos was first defined using the notion of a tripos, but it may alternatively be described as the ex/reg completion of $\mathsf{Asm}(A)$. As the two definitions each have their own merits, we will treat both of them. In this section, we explain the necessary background on tripos theory. Far more extensive accounts may be found in the original sources [HJP80] and [Pit81], or in the more recent exposition [vO08, Chapter 2].

Before we can define the notion of a tripos, we need some auxilliary concepts. By a **Heyting prealgebra**, we mean a preorder which, viewed as category, is finitely complete and cocomplete, and cartesian closed. Equivalently, a Heyting prealgebra is a preorder whose poset reflection is a Heyting algebra. Accordingly, we will employ the usual logical symbols to denote finite limits $(\top, \wedge)$, finite colimits $(\bot, \vee)$ and exponentials $(\rightarrow)$. We write $\mathsf{HeytPre}$ for the preorder-enriched category of Heyting prealgebras and order-preserving functions that preserve all the Heyting structure, up to isomorphism. On the other hand, we write PreOrd for the preorder-enriched category of preorders and order-preserving maps.

Definition 3.1.7. *Let $\mathcal{C}$ be a cartesian closed category. A **tripos** over $\mathcal{C}$ is a pseudofunctor $\mathsf{P}\colon \mathcal{C}^{\mathrm{op}} \to \mathsf{HeytPre}$ such that:*

*(i) For each morphism $f\colon X \to Y$ in $\mathcal{C}$, there exist $\exists_f, \forall_f\colon \mathsf{P}X \to \mathsf{P}Y$ in PreOrd such that $\exists_f \dashv \mathsf{P}f \dashv \forall_f$, and the **Beck-Chevalley Condition** (BCC) holds: whenever*

$$\begin{array}{ccc} W & \xrightarrow{\;q\;} & Z \\ {\scriptstyle p}\big\downarrow & & \big\downarrow{\scriptstyle g} \\ X & \xrightarrow{\;f\;} & Y \end{array}$$

is a pullback diagram in $\mathcal{C}$, we have that $\mathsf{P}g \circ \exists_f$ and $\exists_q \circ \mathsf{P}p$ are isomorphic.

*(ii) There is a **generic predicate**, that is, an object Σ of $\mathcal{C}$ and an element $\sigma \in \mathsf{P}\Sigma$, such that, for all objects X of $\mathcal{C}$ and $\phi \in \mathsf{P}X$, there exists an arrow $[\phi]\colon X \to \Sigma$ such that $\mathsf{P}[\phi](\sigma)$ and ϕ are isomorphic.*

Intuitively, we view $\mathsf{P}X$ as the Heyting prealgebra of predicates on X. We write the order as $\vdash_X$, and we read $\phi \vdash_X \psi$ as 'the predicate ϕ entails the predicate ψ'. Moreover, we write isomorphism in $\mathsf{P}X$ as $\dashv\vdash_X$, so the isomorphism in (ii) above can be written as $\mathsf{P}[\phi](\sigma) \dashv\vdash_X \phi$.

For $f\colon X \to Y$, we view $\mathsf{P}f\colon \mathsf{P}Y \to \mathsf{P}X$ as pulling back along f, or from a logical point of view, as substituting $f(x)$ for y. Accordingly, we will usually write $\mathsf{P}f$ as f^* when P is understood or can be determined from the context. The BCC can now be written as $g^* \circ \exists_f \dashv\vdash \exists_q \circ p^*$, understanding that the isomorphism should hold pointwise in $\mathsf{P}Z$. Note also that the 'genericity' of σ can now be interpreted as the fact that every predicate is a pullback of σ.

Let us make a few additional remarks on Definition 3.1.7.

Remark 3.1.8. (i) The fact that P is a *pseudo*functor means that it only needs to preserve identity and composition 'up to isomorphism'. That is, we require $(\mathrm{id}_X)^* \dashv\vdash \mathrm{id}_{\mathsf{P}X}$ and $(gf)^* \dashv\vdash f^* \circ g^*$.

(ii) The analogue of BCC for $\forall$ also holds, as can be seen by taking the right adjoint of BCC for $\exists$.

(iii) While f^* is required to preserve all the Heyting structure, its adjoints $\exists_f$ and $\forall_f$ are not. As is well-known, the fact that f^* preserves $\to$ is equivalent to the **Frobenius condition**:

$$\exists_f(f^*\psi \wedge \phi) \dashv\vdash \psi \wedge \exists_f \phi,$$

for $f\colon X \to Y$, $\phi \in \mathsf{P}X$ and $\psi \in \mathsf{P}Y$.

(iv) The definition of a tripos also works when $\mathcal{C}$ is merely left exact (as opposed to cartesian closed), but one needs to replace (ii) by a 'parametrized' version. This is comparable to the definition of a natural numbers object in a category that has finite limits, but is not cartesian closed. Since we will only work with triposes over toposes, we do not need to state the more general definition here.

Let us give a few examples of triposes; of course, the second example will be of the most importance to us.

Example 3.1.9. Let $\mathcal{H}$ be a complete Heyting (pre)algebra. The tripos of $\mathcal{H}$-valued predicates $\mathsf{P}_{\mathcal{H}}$ over Set is defined as follows. For a set X, we let $\mathsf{P}_{\mathcal{H}}X$ be $\mathcal{H}^X$ with the pointwise order, which is a Heyting (pre)algebra. If $f\colon X \to Y$ is a function, then f^* is simply composition with f. Moreover, the completeness of $\mathcal{H}$ allows us to define the adjoints of f^*. If $\phi \in \mathcal{H}^X$, then

$$\exists_f(\phi)(y) = \bigvee_{f(x)=y} \phi(x) \quad \text{and} \quad \forall_f(\phi)(y) = \bigwedge_{f(x)=y} \phi(x).$$

Finally, the generic predicate is $\mathrm{id}_{\mathcal{H}} \in \mathcal{H}^{\mathcal{H}}$.

Example 3.1.10. Let A be a PCA. The **realizability tripos** P_A over Set is defined as follows. For a set X, we let $\mathsf{P}_A X$ be $(DA)^X$, but *not* with a pointwise ordering. Instead, if $\phi, \psi \in (DA)^X$, then we say that $\phi \vdash_X \psi$ if there exists an $r \in A^\#$ such that $r \cdot \phi(x) \subseteq \psi(x)$ for all $x \in X$. In other words, r transforms elements of $\phi(x)$ into elements of $\psi(x)$, *uniformly* in x. The Heyting structure on $\mathsf{P}_A X$ can be defined as follows. The top and bottom elements of $\mathsf{P}_A X$ are given by $\top(x) = A$ and $\bot(x) = \emptyset$. If $\phi, \psi \in (DA)^X$, then we set

$$\begin{aligned}(\phi \wedge \psi)(x) &= \mathsf{p} \cdot \phi(x) \cdot \psi(x),\\ (\phi \vee \psi)(x) &= (\mathsf{pk} \cdot \phi(x)) \cup (\mathsf{p}\overline{\mathsf{k}} \cdot \psi(x)),\\ (\phi \to \psi)(x) &= \{a \in A \mid a \cdot \phi(x) \subseteq \psi(x)\}.\end{aligned}$$

Again, if $f\colon X \to Y$ is a function, then f^* is composition with f. Since the Heyting operations are computed pointwise, it is clear that f^* preserves the Heyting structure. Even though $\mathsf{P}_A X$ is a Heyting prealgebra, it is not necessarily complete, but we can still define adjoints of f^* as follows:

$$\exists_f(\phi)(y) = \bigcup_{f(x)=y} \phi(x),$$
$$\forall_f(\phi)(y) = \{a \in A \mid \forall b \in A \forall x \in f^{-1}(y)\,(ab{\downarrow} \wedge ab \in \phi(x))\}.$$

We leave the proof of the adjunctions $\exists_f \dashv f^* \dashv \forall_f$ and of the BCC to the reader. Finally, the generic predicate is $\mathrm{id}_{DA} \in (DA)^{DA}$.

Remark 3.1.11. In the first instance, the reader might expect the definition of $\forall_f$ above to read: $\forall_f(x) = \bigcap_{f(x)=y} \phi(x)$. However, this only yields the required adjunction $f^* \dashv \forall_f$ when f is a *surjective* function.

Example 3.1.12. Every (locally small) elementary topos $\mathcal{E}$ gives rise to the **subobject tripos** $\mathrm{Sub}_{\mathcal{E}}$ over $\mathcal{E}$. To an object X of $\mathcal{E}$, it assigns the subobject lattice $\mathrm{Sub}_{\mathcal{E}}(X)$ of X, which is always a Heyting algebra. If $f\colon X \to Y$ is an arrow of $\mathcal{E}$, then $f^*\colon \mathrm{Sub}_{\mathcal{E}}(Y) \to \mathrm{Sub}_{\mathcal{E}}(X)$ is defined as pullback along f. It is well-known that f^* commutes with all the Heyting structure and has both adjoints, satisfying BCC. Finally, the generic predicate is the subobject classifier of $\mathcal{E}$.

Every tripos gives an interpretation of typed higher-order intuitionistic logic *without equality.* In order to make this more explicit, we introduce the notion of a sequent.

Definition 3.1.13. *Let $\mathcal{L}$ be a language for a typed logic (e.g., typed regular logic or typed predicate logic).*

*(i) A **context** is a sequence of distinct typed variables.*

(ii) If Γ is a context, then a formula φ of $\mathcal{L}$ is said to be in context Γ if all its free variables occur in Γ.

*(iii) A **sequent** is an expression of the form $\varphi \vdash_\Gamma \psi$, where Γ is a context and φ and ψ are $\mathcal{L}$-formulae in context Γ.*

Let P be a tripos over a cartesian closed category $\mathcal{C}$, and let $\mathcal{L}$ be the language for typed higher-order intuitionistic logic without equality, where:

- the types are the objects of $\mathcal{C}$;
- function symbols of type $X_0, \ldots, X_{n-1} \to Y$ are arrows $X_0 \times \cdots \times X_{n-1} \to Y$ of $\mathcal{C}$.
- relation symbols of type $X_0, \ldots, X_{n-1}$ are elements of $\mathsf{P}(X_0 \times \cdots \times X_{n-1})$.

If $\Gamma = x_0 : X_0, \ldots, x_{n-1} : X_{n-1}$ is a context, then we can assign, to each $\mathcal{L}$-formula in context Γ, an interpretation $[\![\varphi]\!] \in \mathsf{P}(X_0 \times \cdots \times X_{n-1})$. We will not give a precise definition of this interpretation as, especially for the first-order fragment, it is standard (see, e.g., [Pit81] or [vO08]). Broadly speaking:

- The Heyting structure on the $\mathsf{P}X$ and the pullback functors allow us to interpret the propositional connectives.
- The adjoints $\exists_f$ and $\forall_f$ allow us to interpret the quantifiers.
- The generic predicate and the cartesian closed structure of $\mathcal{C}$ allow us to interpret higher-order logic. In particular, Σ plays the role of the type of propositions.

Definition 3.1.14. *A sequent $\varphi \vdash_\Gamma \psi$, where $\Gamma = x_0 : X_0, \ldots, x_{n-1} : X_{n-1}$, is **valid** in P if $[\![\varphi]\!] \vdash_{X_0 \times \cdots \times X_{n-1}} [\![\psi]\!]$. We write $\mathsf{P} \colon \varphi \vDash_\Gamma \psi$, or $\varphi \vDash_\Gamma \psi$ if P is understood.*

Of course, this interpretation is sound with respect to a proof system for typed higher-order intuitionistic logic without equality, formulated in terms of sequents. That is, if the sequent s is derivable from a set of sequents S, and all members of S are valid in P, then s will be valid in P as well.

Naturally, there is also a notion of morphism between triposes.

Definition 3.1.15. *Let P and Q be triposes over $\mathcal{C}$. A **transformation of triposes** $f \colon \mathsf{P} \to \mathsf{Q}$ is an arrow $\mathsf{P} \Rightarrow \mathsf{Q}$ in the functor category $[\mathcal{C}^{\mathrm{op}}, \mathsf{PreOrd}]$. We say that f is **left exact** if f_X preserves finite meets for every object X of $\mathcal{C}$. Finally, if $f, f' \colon P \to Q$, then we say that $f \leq f'$ iff $f_X \vdash_X f'_X$ for every object X of $\mathcal{C}$.*

It is easy to check that this makes the triposes over $\mathcal{C}$ into a preorder-enriched category, that we denote by $\mathsf{Trip}(\mathcal{C})$. A transformation $f \colon \mathsf{P} \to \mathsf{Q}$ specifies, for each object X of $\mathcal{C}$, an order-preserving function $f_X \colon \mathsf{P}X \to \mathsf{Q}X$, such that for every arrow $g \colon X \to Y$, we have $f_X \circ g^* \dashv\vdash g^* \circ f_Y$. It is easy to see that f is completely determined, up to isomorphism, by $f_\Sigma(\sigma)$.

Before we introduce realizability toposes, let us note the following connection between the realizability tripos P_A and the category of assemblies A. Since monos in $\mathsf{Asm}(A)$ are injective functions, a subobject of an assembly X is given by a subset $|Y| \subseteq |X|$ and a function $E_Y \colon |Y| \to TA$. We may extend E_Y to a function $\tilde{E}_Y \colon |X| \to DA$, by setting $\tilde{E}_Y(x) = \emptyset$ for x outside $|Y|$. Thus, we obtain a predicate $\tilde{E}_Y \in \mathsf{P}_A|X|$, and the fact that the inclusion $|Y| \subseteq |X|$ is a morphism of assemblies $Y \hookrightarrow X$ means precisely that $\tilde{E}_Y \vdash_{|X|} E_X$. Now it is not hard to see that the poset of subobjects $\mathrm{Sub}(X)$ is equivalent to $\downarrow\{E_X\} \subseteq \mathsf{P}_A|X|$. Moreover, if $f \colon X' \to X$ is a morphism of assemblies, then the pullback function $f^* \colon \mathrm{Sub}(X) \to \mathrm{Sub}(X')$ coincides, modulo this equivalence, with the restriction of $f^* \colon \mathsf{P}_A|X| \to \mathsf{P}_A|X'|$ to $\downarrow\{E_X\}$.

This yields an alternative description of the realizability tripos P_A. Indeed, P_A is equivalent (in $\mathsf{Trip}(\mathsf{Set})$) to $\mathrm{Sub}_{\mathsf{Asm}(A)}(\nabla(-))$, where $\mathrm{Sub}_{\mathsf{Asm}(A)}(\nabla X)$ is the subobject poset of ∇X, and $\mathrm{Sub}_{\mathsf{Asm}(A)}(\nabla f)$ is pullback along ∇f.

3.1.3 The realizability topos

The construction of the realizability topos is an instance of a more general construction, which assigns, to each tripos, a corresponding topos.

Definition 3.1.16 (Tripos-to-topos). *Let* P *be a tripos over* $\mathcal{C}$. *The category* $\mathcal{C}[\mathsf{P}]$ *is defined as follows.*

(i) An object of $\mathcal{C}[\mathsf{P}]$ *is a pair* $X = (|X|, \sim_X)$*, where* $|X|$ *is an object of* $\mathcal{C}$ *and* $\sim_X \in \mathsf{P}(|X| \times |X|)$ *is:*

- *symmetric:* $x \sim_X x' \vDash_{x,x'} x' \sim_X x$;
- *transitive:* $x \sim_X x' \wedge x' \sim_X x'' \vDash_{x,x',x''} x \sim_X x''$.

(ii) If X *and* Y *are objects of* $\mathcal{C}[\mathsf{P}]$*, then a* ***functional relation*** *from* X *to* Y *is an* $F \in \mathsf{P}(|X| \times |Y|)$ *that is:*

- *strict:* $F(x,y) \vDash_{x,y} x \sim_X x \wedge y \sim_Y y$;
- *relational:* $F(x,y) \wedge x \sim_X x' \wedge y \sim_Y y' \vDash_{x,x',y,y'} F(x', y')$;
- *single-valued:* $F(x,y) \wedge F(x,y') \vDash y \sim_Y y'$;
- *total:* $x \sim_X x \vDash_x \exists y\, F(x,y)$.

Moreover, we say that two functional relations are isomorphic if they are isomorphic in $\mathsf{P}(|X| \times |Y|)$.

(iii) An arrow $X \to Y$ *of* $\mathcal{C}[\mathsf{P}]$ *is an equivalence class of functional relations from* X *to* Y*. If* $X \to Y$ *is the equivalence class of the functional relation* F*, then we also say that* F *represents the arrow* $X \to Y$*.*

An object X of $\mathcal{C}[\mathsf{P}]$ can be viewed as an object $|X|$ equipped with an equality predicate $\sim_X$. Accordingly, we write $x \sim_X x'$ rather than $\sim_X(x, x')$, and we think of this statement as expressing that x is identical to x'. Note that we do *not* require that $\vDash_x x \sim_X x$, so $\sim_X$ is really a *partial* equivalence relation. We think of the statement $x \sim_X x$ ('x is identical to itself') as expressing that x *exists*.

Similarly, if F represents the arrow $f \colon X \to Y$, then we view $F(x,y)$ as the statement that f sends x to y. The first two requirements make sure that F is well-behaved w.r.t. equality: $F(x,y)$ should only hold for x and y that exist, and F should respect equality. The other two requirements state that F behaves like a function: if f sends x to both y and y', then y and y' should be identical. Moreover, if x exists, then $f(x)$ must exist as well.

In order to see that $\mathcal{C}[\mathsf{P}]$ is indeed a category, we note that $\sim_X$ represents the identity on X, and if $X \to Y \to Z$ are represented by $F \in \mathsf{P}(|X| \times |Y|)$ and $G \in \mathsf{P}(|Y| \times |Z|)$, then their composition $X \to Z$ is represented by:

$$H := [\![\exists y\, (F(x,y) \wedge G(y,z))]\!] \in \mathsf{P}(|X| \times |Z|).$$

The sequents expressing that H is a functional relation are derivable from the sequents expressing that F and G are functional relations. So by soundness, we know that H thus defined is actually a functional relation from X to Z. Similarly, the sequents expressing that $\sim_X$ is a functional relation from X to itself are derivable from the symmetry and transitivy of $\sim_X$. The axioms for a category can also easily be deduced by reasoning internally in P. In fact, $\mathcal{C}[\mathsf{P}]$ is always a topos. We will not prove this here, but refer to [Pit81] or [vO08].

Proposition 3.1.17 (Pitts)**.** *If* P *is a tripos over* $\mathcal{C}$*, then* $\mathcal{C}[\mathsf{P}]$ *is an elementary topos.*

Example 3.1.18. If $\mathcal{H}$ is a complete Heyting algebra, then $\mathsf{Set}[\mathsf{P}_{\mathcal{H}}]$ is called the topos of $\mathcal{H}$-valued sets, and it is equivalent to the topos of sheaves over $\mathcal{H}$.

Example 3.1.19. If $\mathcal{E}$ is a topos, then $\mathcal{E}[\mathrm{Sub}_{\mathcal{E}}]$ is equivalent to $\mathcal{E}$ itself.

Definition 3.1.20. *For a PCA A, we call* $\mathsf{Set}[\mathsf{P}_A]$ *the* ***realizability topos*** *of A, and we denote it by* $\mathsf{RT}(A)$*. Traditionally,* $\mathsf{RT}(\mathcal{K}_1)$ *is called the* ***effective topos****, and denoted by* $\mathcal{E}ff$*.*

Thus, an object X of $\mathsf{RT}(A)$ is a pair $(|X|, \sim_X)$, where $|X|$ is a set and $\sim_X$ is a function $|X| \times |X| \to DA$. We think of the elements of $x \sim_X x'$ as *evidence* for the fact that x and x' are equal, or as *realizers* of the statement that x and x' are equal. In the case where $x = x'$, we think of the elements of $x \sim_X x$ as realizers of the existence of x, or simply: realizers of x. There should be an algorithm realizing the symmetry of $\sim_X$, meaning that there is an $s \in A^{\#}$ such that $s \cdot (x \sim_X x') \subseteq (x' \sim_X x)$ for all $x, x' \in |X|$. Similarly for transitivity, there should be a $t \in A^{\#}$ such that $t \cdot (x \sim_X x') \cdot (x' \sim_X x'') \subseteq (x \sim_X x'')$ for all $x, x', x'' \in |X|$. The requirements for functional relations can similarly be described in terms of elements of $A^{\#}$. The advantage of the tripos perspective is that, in order to see that $\mathsf{RT}(A)$ is indeed a category (and subsequently, a topos), we do not have to perform all kinds of explicit constructions inside A. Instead, we can simply refer to the internal logic of the realizability tripos P_A.

Definition 3.1.21. *Let* P *be a tripos over* $\mathcal{C}$*. The* ***constant object functor*** $\nabla_{\mathsf{P}} \colon \mathcal{C} \to \mathcal{C}[\mathsf{P}]$ *is defined as follows. If* $X \in \mathcal{C}$*, then* $\nabla_{\mathsf{P}} X$ *is* $(X, \exists_\delta \top)$*, where* $\delta \colon X \hookrightarrow X \times X$ *is the diagonal. Moreover, if* $f \colon X \to Y$ *is an arrow of* $\mathcal{C}$*, then* $\nabla_{\mathsf{P}} f$ *is represented by* $\exists_{\langle \mathrm{id}_X, f \rangle} \top$*.*

The constant object functor is always left exact (see [Pit81, Proposition 3.4] or [vO08, Proposition 2.4.1]). Let us describe the constant object functor in the examples above.

Example 3.1.22. If $\mathcal{H}$ is a complete Heyting algebra, then the corresponding constant object functor $\mathsf{Set} \to \mathsf{Set}[\mathsf{P}_{\mathcal{H}}]$ is the inverse image of the (unique) geometric morphism $\mathsf{Set}[\mathsf{P}_{\mathcal{H}}] \to \mathsf{Set}$.

Example 3.1.23. If A is a PCA, then we will denote ∇_{P_A} by $\hat{\nabla}_A$, or simply $\hat{\nabla}$ if the PCA A is clear from the context. We use $\hat{\nabla}$ here to avoid confusion with the constant object functor $\nabla \colon \mathsf{Set} \to \mathsf{Asm}(A)$ we defined in the previous section.

If X is a set, then $\hat{\nabla} X = (X, \sim)$, where

$$(x \sim x') = \begin{cases} A & \text{if } x = x'; \\ \emptyset & \text{if } x \neq x'. \end{cases}$$

Moreover, if $f \colon X \to Y$, then $\hat{\nabla} f$ is represented by $F \in \mathsf{P}(X \times Y)$, where $F(x, y) = A$ if $f(x) = y$, and $F(x, y) = \emptyset$ otherwise. We will revisit $\hat{\nabla}$ in the next section, where we will also explain its relation to $\nabla \colon \mathsf{Set} \to \mathsf{Asm}(A)$.

Example 3.1.24. If $\mathcal{E}$ is a topos, then the constant object functor for $\mathrm{Sub}_{\mathcal{E}}$ is one half of the equivalence $\mathcal{E} \simeq \mathcal{E}[\mathrm{Sub}_{\mathcal{E}}]$.

Remark 3.1.25. Even though we certainly need the tripos structure to obtain Proposition 3.1.17, we have defined $\mathcal{C}[\mathsf{P}]$ and ∇_{P} using only finite limits in $\mathcal{C}$ and formulas of regular logic. Thus, these definitions make sense for any pseudofunctor $\mathsf{P}\colon \mathcal{C}^{\mathrm{op}} \to \mathsf{PreOrd}$. where $\mathcal{C}$ is left exact and P soundly interprets regular logic.

Consider a left exact transformation $f\colon \mathsf{P} \to \mathsf{Q}$ of triposes over $\mathcal{C}$ (or indeed, pseudofunctors $\mathcal{C}^{\mathrm{op}} \to \mathsf{PreOrd}$ interpreting regular logic). Suppose that f commutes with existential quatification, that is, $f_Y \circ \exists_g \dashv\vdash \exists_g \circ f_X$ for $g\colon X \to Y$. Then f preserves the interpretation of regular logic, so we may call such an f **regular**. Now we can define a functor $\overline{f}\colon \mathcal{C}[\mathsf{P}] \to \mathcal{C}[\mathsf{Q}]$ by $\overline{f}(X) = (|X|, f(\sim_X))$, and by sending an arrow represented by the functional relation F to the arrow represented by $f(F)$. Since f preserves regular logic, $\overline{f}$ is a well-defined regular functor that satisfies $\overline{f}\nabla_{\mathsf{P}} \cong \nabla_{\mathsf{Q}}$.

3.2 Properties of the realizability topos

In this section, we record some general properties shared by all realizability toposes. Since all of these are well-known, we will be rather brief on proofs. This section is certainly not exhaustive, in the sense that there are many more properties of assemblies and realizability toposes that are not discussed here.

3.2.1 $\mathsf{Asm}(A)$ as a subcategory of $\mathsf{RT}(A)$

First of all, we will explain how the category of assemblies $\mathsf{Asm}(A)$ can be seen as a subcategory of the realizability topos $\mathsf{RT}(A)$.

Proposition 3.2.1. *There is a fully faithful functor $i\colon \mathsf{Asm}(A) \to \mathsf{RT}(A)$.*

Proof. For an assembly X, define $i(X)$ as the object $(|X|, \sim)$, where

$$(x \sim x') = \begin{cases} E_X(x) & \text{if } x = x'; \\ \emptyset & \text{if } x \neq x'. \end{cases}$$

Moreover, if $f\colon X \to Y$ is a morphism of assemblies, then we let $i(f)$ be represented by F, where $F(x,y) = E_X(x)$ if $f(x) = y$, and $F(x,y) = \emptyset$ otherwise. It is easy to check that F is functional relation if and only if f has a tracker, and that the functor i thus defined is faithful. For fullness, if $f\colon i(X) \to i(Y)$ is represented by F, then we can define $f'\colon |X| \to |Y|$ by $f'(x) = y$ iff $F(x,y) \neq \emptyset$, and one easily checks that $i(f') = f$. □

By a slight abuse of terminology, we will say that an object of $\mathsf{RT}(A)$ is an assembly if it is isomorphic to an object in the image of $i\colon \mathsf{Asm}(A) \to \mathsf{RT}(A)$. Note that the composition $\mathsf{Set} \xrightarrow{\nabla} \mathsf{Asm}(A) \xrightarrow{i} \mathsf{RT}(A)$ coincides with the constant object functor $\hat{\nabla}$ from Example 3.1.23. Thus, no real confusion can arise by calling both these functors the constant object functor. Note also that $\hat{\nabla}$, being the composition of two fully faithful functors, is fully faithful.

There is another description of the assemblies in $\mathsf{RT}(A)$, but before we can give it, we need to have a closer look at subobjects in $\mathsf{RT}(A)$, or more generally, in a topos of the form $\mathcal{C}[\mathsf{P}]$. In the previous section, we saw that functional relations, besides behaving like a function, should also be well-behaved w.r.t. identity. We generalize this to arbitrary predicates on an object of $\mathcal{C}[\mathsf{P}]$.

Definition 3.2.2. *Let* P *be a tripos over* $\mathcal{C}$ *and let* X *be an object of* $\mathcal{C}[\mathsf{P}]$*. If* $\phi \in \mathsf{P}|X|$*, then we say that* ϕ *is a:*

(i) ***strict*** *predicate on* X *if* $\phi(x) \vDash_x x \sim_X x$*;*

(ii) ***relational*** *predicate on* X *if* $\phi(x) \wedge x \sim_X x' \vDash_{x,x'} \phi(x')$*;*

Lemma 3.2.3. *Let* P *be a tripos over* $\mathcal{C}$ *and let* X *be an object of* $\mathcal{C}[\mathsf{P}]$*. Then* $\mathrm{Sub}_{\mathcal{C}[\mathsf{P}]}(X)$ *is equivalent to the preorder of strict and relational predicates* $\phi \in \mathsf{P}|X|$ *on* X*.*

Sketch of proof. If $\phi \in \mathsf{P}|X|$ is strict and relational for X, then we have the object $(|X|, \sim_X^\phi)$ of $\mathcal{C}[\mathsf{P}]$, where $\sim_X^\phi$ is $[\![x \sim_X x' \wedge \phi(x)]\!]$. Moreover, $\sim_X^\phi$ also represents a monomorphism $(|X|, \sim_X^\phi) \hookrightarrow X$. If $\phi, \psi \in \mathsf{P}|X|$ are strict and relational for X, then it is easily verified that $(|X|, \sim_X^\phi) \le (|X|, \sim_X^\psi)$ as subobjects of X iff we have $\phi \vdash_{|X|} \psi$. Finally, if $Y \hookrightarrow X$ is a mono represented by F, then $\phi := [\![\exists y\, F(y,x)]\!] \in \mathsf{P}|X|$ is strict and relational for X, and Y and $(X, \sim_X^\phi)$ are isomorphic subobjects of X. □

We warn the reader, though, that the Heyting structure on $\mathrm{Sub}(X)$ does not coincide with the Heyting structure on $\mathsf{P}|X|$. For example, if ϕ, ψ are strict and relational for X, then $(|X|, \sim_X^\phi) \to (|X|, \sim_X^\psi)$ in $\mathrm{Sub}(X)$ is not given by $\phi \to \psi \in \mathsf{P}|X|$. Indeed, while $\phi \to \psi$ is relational, it is not necessarily strict. Instead, $(|X|, \sim_X^\phi) \to (|X|, \sim_X^\psi)$ corresponds to the strict and relational predicate $[\![(\phi(x) \to \psi(x)) \wedge x \sim_X x]\!] \in \mathsf{P}|X|$.

If X is a set, then every predicate $\phi \in \mathsf{P}_A X$ is strict and relational for $\hat{\nabla} X$, so $\mathrm{Sub}(\hat{\nabla} X)$ is equivalent to $\mathsf{P}_A X$. If $\phi \in \mathsf{P}_A X$, then the subobject of $\hat{\nabla} X$ corresponding to ϕ is isomorphic to the assembly Y, which is given by $|Y| = \{x \in X \mid \phi(x) \neq \emptyset\}$ and $E_Y(y) = \phi(y)$. Conversely, if Y is an assembly, then Y is isomorphic to the subobject of $\hat{\nabla}|Y|$ given by the predicate $E_Y \in \mathsf{P}_A|Y|$. Thus, we see that the assemblies in $\mathsf{RT}(A)$ are precisely the subobjects of the objects in the image of $\hat{\nabla} \colon \mathsf{Set} \to \mathsf{RT}(A)$, i.e., the subobjects of the constant objects. In particular, since $\hat{\nabla} \colon \mathsf{Set} \to \mathsf{RT}(A)$ is left exact, it follows that the assemblies are closed under finite limits in $\mathsf{RT}(A)$. Moreover, it follows that the assemblies are closed under subobjects in $\mathsf{RT}(A)$. Together, these two facts imply that the regular structure of $\mathsf{Asm}(A)$ coincides with the regular structure it inherits from $\mathsf{RT}(A)$, i.e., the inclusion $i \colon \mathsf{Asm}(A) \to \mathsf{RT}(A)$ is regular. In particular, $\hat{\nabla} \colon \mathsf{Set} \to \mathsf{RT}(A)$, being the composition of two regular functors, is regular.

As we mentioned in Section 1.1.4, $\mathsf{RT}(A)$ can be described as a universal construction in two ways. We will now discuss the first of these, which also yields an alternative way of constructing the realizability topos.

Proposition 3.2.4. *Let $\mathcal{R}$ be a regular category. There exists a fully faithful regular functor $i\colon \mathcal{R} \to \mathcal{R}_{\text{ex/reg}}$ with $\mathcal{R}_{\text{ex/reg}}$ an exact category, called the **ex/reg completion** of $\mathcal{R}$, such that for each exact category $\mathcal{E}$, composition with i yields an equivalence of categories:*

$$\mathsf{REG}(\mathcal{R}_{\text{ex/reg}}, \mathcal{E}) \simeq \mathsf{REG}(\mathcal{R}, \mathcal{E}).$$

Sketch of proof. Consider the subobject functor $\text{Sub}_{\mathcal{R}}\colon \mathcal{R}^{\text{op}} \to \mathsf{PreOrd}$. This is certainly not in general a tripos, but it does soundly interpret regular logic, since $\mathcal{R}$ is a regular category. As per Remark 3.1.25, we can define $\nabla_{\text{Sub}_{\mathcal{R}}}\colon \mathcal{R} \to \mathcal{R}[\text{Sub}_{\mathcal{R}}]$ as we did for triposes. This is the ex/reg completion of $\mathcal{R}$. □

Remark 3.2.5. In the particular case of $\text{Sub}_{\mathcal{R}}$, every object of $\mathcal{R}[\text{Sub}_{\mathcal{R}}]$ is isomorphic to an object X where $\sim_X$ is a genuine equivalence relation, i.e., $\vDash_x x \sim_X x$. This leads to the description of $\mathcal{R}_{\text{ex/reg}}$ one more commonly finds in the literature:

- objects are pairs (X, R), where X is an object of $\mathcal{R}$ and $R \hookrightarrow X \times X$ is an (internal) equivalence relation on X;
- arrows $(X, R) \to (Y, S)$ are subobjects $F \hookrightarrow X \times Y$ that are relational, single-valued and total relative to R and S (note that strictness is empty in this case).

Proposition 3.2.6. *If A is a PCA, then $i\colon \mathsf{Asm}(A) \to \mathsf{RT}(A)$ is the ex/reg completion of $\mathsf{Asm}(A)$.*

Sketch of proof. We will describe how to extend a regular functor $F\colon \mathsf{Asm}(A) \to \mathcal{E}$, with $\mathcal{E}$ exact, to a regular functor $\mathsf{RT}(A) \to \mathcal{E}$. For the time being, suppose that $\mathcal{E}$ is only regular. First of all, note that we have a regular functor $F\nabla\colon \mathsf{Set} \to \mathcal{E}$. Second, if X is a set, then we get a function

$$f_X : \mathsf{P}_A X \simeq \text{Sub}_{\mathsf{Asm}(A)}(\nabla X) \xrightarrow{F} \text{Sub}_{\mathcal{E}}(F\nabla X).$$

The f_X do not form a regular transformation $\mathsf{P}_A \Rightarrow \text{Sub}_{\mathcal{E}}$, since we are working over different bases. On the other hand, the f_X do form a regular transformation $\mathsf{P}_A \Rightarrow \text{Sub}_{\mathcal{E}}$ 'over $F\nabla$', meaning that $f_X \circ g^* \dashv\vdash (F\nabla g)^* \circ f_Y$ for every function $g\colon X \to Y$. This still allows us to define the regular functor $\overline{f}\colon \mathsf{RT}(A) = \mathsf{Set}[\mathsf{P}_A] \to \mathcal{E}[\text{Sub}_{\mathcal{E}}]$, and it makes the following diagram commute:

$$\begin{array}{ccc} \mathsf{Asm}(A) & \xrightarrow{F} & \mathcal{E} \\ {\scriptstyle i}\downarrow & & \downarrow{\scriptstyle \nabla} \\ \mathsf{RT}(A) & \xrightarrow{\overline{f}} & \mathcal{E}[\text{Sub}_{\mathcal{E}}] \end{array}$$

If we assume that $\mathcal{E}$ is exact, then $\mathcal{E}$ is equivalent to its own ex/reg completion, which means that ∇ above is an equivalence. This yields the desired extention of F. □

Remark 3.2.7. Proposition 3.2.6 above can be formulated more generally for triposes. If P is a tripos, then by Lemma 3.2.3, P is equivalent to $\mathrm{Sub}_{\mathcal{C}[\mathsf{P}]}(\nabla_{\mathsf{P}}(-))$. Now let us define $\mathsf{Asm}(\mathsf{P})$ as the full subcategory of $\mathcal{C}[\mathsf{P}]$ on the objects that embed into a constant object. Then by the same arguments as given for the case of the realizability tripos, we can show that $\mathsf{Asm}(\mathsf{P})$ inherits the structure of a regular category from $\mathcal{C}[\mathsf{P}]$, and that $\mathsf{Asm}(\mathsf{P}) \hookrightarrow \mathcal{C}[\mathsf{P}]$ is its ex/reg completion.

In the proof of Proposition 3.2.6, we defined a functor on a topos of the form $\mathcal{C}[\mathsf{P}]$ in terms of a transformation on the underlying tripos P. These ideas are already present in [Pit81]. On the other hand, Proposition 3.2.6 was not stated until [CFS88] (for the case of $\mathcal{E}\mathit{ff}$).

3.2.2 Set as the subtopos of ¬¬-sheaves

Since $\Gamma\colon \mathsf{Asm} \to \mathsf{Set}$ is regular and Set is exact, Proposition 3.2.6 allows us to extend Γ to a regular functor $\hat{\Gamma}\colon \mathsf{RT}(A) \to \mathsf{Set}$, i.e., we have $\hat{\Gamma}i \cong \Gamma$. Explicitly, if X is an object of $\mathcal{C}[\mathsf{P}]$, then $\hat{\Gamma}X$ is the quotient of $|X|$ under the *partial* equivalence relation R defined by $R(x, x')$ iff $(x \sim_X x') \neq \emptyset$. That is, if we write $[x]_{\sim_X}$ for the set of all $x' \in |X|$ such that $(x \sim_X x') \neq \emptyset$, then

$$\hat{\Gamma}X = \{[x]_{\sim_X} : x \in |X| \text{ and } [x]_{\sim_X} \neq \emptyset\}.$$

It is easily checked that, in analogy with Lemma 3.1.5, $\hat{\Gamma}$ is the global sections functor iff A is an absolute PCA. Now consider the pair of functors:

$$\mathsf{Set} \underset{\hat{\nabla}}{\overset{\hat{\Gamma}}{\rightleftarrows}} \mathsf{RT}(A).$$

We have $\hat{\Gamma}\hat{\nabla} = \hat{\Gamma}i\nabla \cong \Gamma\nabla = \mathrm{id}_{\mathsf{Set}}$. Moreover, by Proposition 3.2.6, the unit $\eta\colon \mathrm{id}_{\mathsf{Asm}(A)} \Rightarrow \nabla\Gamma$ can be extended to a natural transformation $\hat{\eta}\colon \mathrm{id}_{\mathsf{RT}(A)} \Rightarrow \hat{\nabla}\hat{\Gamma}$. Again using Proposition 3.2.6, we can verify that the isomorphism $\hat{\Gamma}\hat{\nabla} \Rightarrow \mathrm{id}_{\mathsf{Set}}$ and $\hat{\eta}\colon \mathrm{id}_{\mathsf{RT}(A)} \Rightarrow \hat{\nabla}\hat{\Gamma}$ satisfy the triangle identities, so we have $\hat{\Gamma} \dashv \hat{\nabla}$. Since the counit of this adjunction is an isomorphism and $\hat{\Gamma}$ preserves finite limits, this presents Set as a subtopos of $\mathsf{RT}(A)$. If fact, this is always the inclusion of ¬¬-sheaves, as was observed already in [Hyl82, Proposition 4.4] for the case of the effective topos. In order to give the proof, we first need the notion of a dense subtopos (which is unrelated to the notion of density from the previous chapter).

Definition 3.2.8. *A geometric inclusion $i\colon \mathcal{E} \to \mathcal{F}$ between toposes is called **dense** if $0 \in \mathcal{F}$ is a sheaf for this inclusion. We also say that $\mathcal{E}$ is a dense subtopos of $\mathcal{F}$.*

In other words, i is dense iff 0 is isomorphic to its sheafification $i_*(i^*(0)) \cong i_*(0)$, that is, iff i_* preserves the initial object. This, in turn, is equivalent to saying that i^* *reflects* the initial object. Indeed, suppose that $i_*(0) \cong 0$ and $i^*(X) \cong 0$ for a certain $X \in \mathcal{F}$. Then it follows that $X \to i_*(i^*(X)) \cong i_*(0) \cong 0$, and the initial object is strict in a topos, so $X \cong 0$. Conversely, if i^* reflects the initial object, then $i^*(i_*(0)) \cong 0$ implies that $i_*(0) \cong 0$.

The following is proposition is treated in [Joh77, Proposition 5.18].

Proposition 3.2.9. *If $\mathcal{E}$ is a topos, then $\neg\neg$ is the largest dense topology on $\mathcal{E}$. Accordingly, the $\neg\neg$-sheaves of $\mathcal{E}$ form the smallest dense subtopos of $\mathcal{E}$. In particular, if $\mathcal{E}$ is boolean, then $\mathcal{E}$ has no proper dense subtoposes.*

Using this proposition, we can prove:

Proposition 3.2.10. *The geometric inclusion $\hat{\Gamma} \dashv \hat{\nabla} \colon \mathsf{Set} \to \mathsf{RT}(A)$ is equivalent to the inclusion of $\neg\neg$-sheaves of $\mathsf{RT}(A)$. In particular, an object of $\mathsf{RT}(A)$ is a $\neg\neg$-sheaf iff it is a constant object, and $\neg\neg$-separated iff it is an assembly.*

Proof. Let $\mathsf{RT}(A)_{\neg\neg} \hookrightarrow \mathsf{RT}(A)$ denote the subtopos of $\neg\neg$-sheaves. It is easy to check that $\hat{\nabla}\emptyset$ is the initial object of $\mathsf{RT}(A)$, which means that $\hat{\Gamma} \dashv \hat{\nabla} \colon \mathsf{Set} \to \mathsf{RT}(A)$ is dense. In particular, $\mathsf{RT}(A)_{\neg\neg}$ must be a dense subtopos of Set. But Set is boolean, so we can conclude that $\mathsf{RT}(A)_{\neg\neg}$ and Set coincide. ☐

3.2.3 Projective objects

In this section, we discuss the other way in which $\mathsf{RT}(A)$ can be seen as a universal construction. First, we need to introduce the notion of a projective object.

Definition 3.2.11. *Let $\mathcal{C}$ be a regular category.*

(i) An object P of $\mathcal{C}$ is called ***projective*** *if every regular epi with codomain P splits.*

(ii) We say that $\mathcal{C}$ has ***enough projectives*** *if every object is covered by a projective object, that is, for every object X there exists a regular epi $P \twoheadrightarrow X$ with P projective.*

One of the main goals of this section is to show that the projective objects in $\mathsf{RT}(A)$ are the objects introduced in the following definition.

Definition 3.2.12. *An assembly X is called* ***partitioned*** *if $E_X(x) \colon |X| \to TA$ factors through δ_A, i.e., $E_X(x)$ is a principal downset of A, for all $x \in |X|$. We write $\mathsf{PAsm}(A)$ for the full subcategory of $\mathsf{Asm}(A)$ on the partitioned assemblies.*

Example 3.2.13. The object of realizers R_A and the natural numbers object N from Example 3.1.6 are both partitioned. In fact, an assembly X is isomorphic to a partitioned assembly if and only if it allows a prone morphism of assemblies $X \to R_A$.

We can give the following equivalent description of $\mathsf{PAsm}(A)$: its objects are pairs $X = (|X|, e_X)$ where $e_X : |X| \to A$, and arrows $f : X \to Y$ are functions $|X| \to |Y|$ for which there exists a tracker $t \in A^{\#}$ such that $t \cdot e_X(x) \leq e_Y(f(x))$ for all $x \in |X|$.

Example 3.2.14. As the description of PAsm above makes clear, we have $\mathsf{Asm}(A) \cong \mathsf{PAsm}(TA)$.

Lemma 3.2.15. *Let A be a PCA.*

(i) Every constant object is isomorphic to a partitioned assembly.

(ii) The partitioned assemblies are closed under finite limits in $\mathsf{RT}(A)$*.*

(iii) Every object of $\mathsf{RT}(A)$ *is covered by a partitioned assembly.*

It should be noted that the property of being partitioned is not stable under isomorphism, so $\mathsf{PAsm}(A)$ is not a *replete* subcategory of either $\mathsf{Asm}(A)$ or $\mathsf{RT}(A)$. Thus, by statement (ii), we really mean: given a finite diagram in $\mathsf{PAsm}(A)$, its limit, as computed in $\mathsf{RT}(A)$, is *isomorphic to* a partitioned assembly. We can also formulate this as: the fully faithful functor $\mathsf{PAsm}(A) \to \mathsf{RT}(A)$ preserves finite limits.

Proof of Lemma 3.2.15. (i) If X is a set, then ∇X is isomorphic to the partitioned assembly (X, E), where $E(x) = {\downarrow}\{\mathsf{i}\}$.

(ii) Since $i : \mathsf{Asm}(A) \to \mathsf{RT}(A)$ preserves finite limits, it suffices to prove that partitioned assemblies are closed under finite limits in $\mathsf{Asm}(A)$. This follows easily by inspecting the construction of finite limits in $\mathsf{Asm}(A)$ in the proof of Proposition 3.1.2.

(iii) Let Y be an object of $\mathsf{RT}(A)$. Then Y is covered by the partitioned assembly X given by $|X| = \{(y, a) \mid y \in |Y| \text{ and } a \in (y \sim_Y y)\}$, and $E_X(y, a) = {\downarrow}\{a\}$. The cover $X \twoheadrightarrow Y$ is represented by $((y, a), y') \mapsto \mathsf{p}a \cdot (y \sim_Y y')$. □

Now we are ready to prove that the projective objects of $\mathsf{RT}(A)$ are, up to isomorphism, exactly the partitioned assemblies. This result requires, and is in fact equivalent to, the Axiom of Choice.

Proposition 3.2.16. *An object of* $\mathsf{RT}(A)$ *is projective if and only if it is isomorphic to a partitioned assembly.*

Proof. The challenging part, which is also the part that needs AC, is showing that every partitioned assembly is projective. Let X be a partitioned assembly, and let $Y \twoheadrightarrow X$ be a regular epi represented by $F \in \mathsf{P}(|Y| \times |X|)$. The fact that F represents a regular epi means that $E_X(x) \vDash_x \exists y\, F(y, x)$ holds, so there exists an $r \in A^{\#}$ such that $r \cdot E_X(x) \subseteq \bigcup_{y \in |Y|} F(y, x)$ for all $x \in |X|$. Now write $E_X(x) = {\downarrow}\{e_X(x)\}$, where $e_X : |X| \to A$. Then in particular, we have $r \cdot e_X(x) \in \bigcup_{y \in |Y|} F(y, x)$, so by AC, there is a function $g : |X| \to |Y|$ such that $r \cdot e_X(x) \in F(g(x), x)$ for all $x \in |X|$. This implies that $r \cdot E_X(x) \subseteq F(g(x), x)$

for all $x \in |X|$. Now $[\![E_X(x) \wedge g(x) \sim_Y y]\!] \in \mathsf{P}(|X| \times |Y|)$ is a functional relation from X to Y, and it represents a section of $Y \twoheadrightarrow X$.

The converse is a folklore argument that uses only elementary category theory (see, e.g., [CV98, Proposition 3]). Suppose that P is a projective object of $\mathsf{RT}(A)$. In view of Lemma 3.2.15(iii), there exists a regular epi $e\colon X \twoheadrightarrow P$, where X is a partitioned assembly. Since P is projective, e splits by means of an $m\colon P \hookrightarrow X$. Now

$$P \xhookrightarrow{\;m\;} X \underset{me}{\overset{\mathrm{id}}{\rightrightarrows}} X$$

is an equalizer diagram, and since partitioned assemblies are closed under finite limits (Lemma 3.2.15(ii)), P must be isomorphic to a partitioned assembly. □

Clearly, Lemma 3.2.15 and Proposition 3.2.16 also hold when $\mathsf{RT}(A)$ is replaced by $\mathsf{Asm}(A)$.

Remark 3.2.17. In order to see that Proposition 3.2.16 is equivalent to AC, we note that, in the adjunction $\hat{\Gamma} \dashv \hat{\nabla}$, the right adjoint $\hat{\nabla}$ is regular. It is a well-known result in category theory that this implies that the left adjoint $\hat{\Gamma}$ preserves projectives. If X is a set, then $\hat{\nabla}X$, being a constant object, is projective. It follows that $X \cong \hat{\Gamma}\hat{\nabla}X$ is also projective, for every set X; and this is the Axiom of Choice.

Now we can present the other way in which $\mathsf{RT}(A)$ is a free construction. The following results can all be found in [Car95].

Proposition 3.2.18 (Carboni, Celia Magno, Rosolini)**.** *Let $\mathcal{C}$ be a left exact category.*

*(i) There exists a fully faithful left exact functor $i\colon \mathcal{C} \to \mathcal{C}_{\mathrm{reg/lex}}$ with $\mathcal{C}_{\mathrm{reg/lex}}$ a regular category, called the **reg/lex completion** of $\mathcal{C}$, such that for each regular category $\mathcal{R}$, composition with i yields an equivalence of categories:*

$$\mathsf{REG}(\mathcal{C}_{\mathrm{reg/lex}}, \mathcal{R}) \simeq \mathsf{LEX}(\mathcal{C}, \mathcal{R}).$$

(ii) A fully faithful left exact functor $j\colon \mathcal{C} \to \mathcal{R}$, with $\mathcal{R}$ regular, is equivalent to the reg/lex completion of $\mathcal{C}$ if and only if: the projective objects of $\mathcal{R}$ are precisely those isomorphic to an object in the image of j, every object of $\mathcal{R}$ can be embedded into a projective, and $\mathcal{R}$ has enough projectives.

*(iii) There exists a fully faithful left exact functor $i\colon \mathcal{C} \to \mathcal{C}_{\mathrm{ex/lex}}$ with $\mathcal{C}_{\mathrm{ex/lex}}$ an exact category, called the **ex/lex completion** of $\mathcal{C}$, such that for each exact category $\mathcal{E}$, composition with i yields an equivalence of categories:*

$$\mathsf{REG}(\mathcal{C}_{\mathrm{ex/lex}}, \mathcal{E}) \simeq \mathsf{LEX}(\mathcal{C}, \mathcal{E}).$$

(iv) A fully faithful left exact functor $j\colon \mathcal{C} \to \mathcal{E}$, with $\mathcal{E}$ exact, is equivalent to the ex/lex completion of $\mathcal{C}$ if and only if: the projective objects of $\mathcal{E}$ are precisely those isomorphic to an object in the image of j, and $\mathcal{E}$ has enough projectives.

Note that (iii) follows by combining (i) with the construction in Proposition 3.2.4. Using the results obtained in this section, we immediately obtain the following. The second part of the corollary was first observed in [RR90].

Corollary 3.2.19. *Let A be a PCA. Then:*

(i) the inclusion $\mathsf{PAsm}(A) \to \mathsf{Asm}(A)$ *is the reg/lex completion of* $\mathsf{PAsm}(A)$*;*

(ii) the fully faithful functor $\mathsf{PAsm}(A) \to \mathsf{RT}(A)$ *is the ex/lex completion of* $\mathsf{PAsm}(A)$*.*

Proof. By Lemma 3.2.15(ii), $\mathsf{PAsm}(A) \to \mathsf{Asm}(A)$ and $\mathsf{PAsm}(A) \to \mathsf{RT}(A)$ are both left exact. In both $\mathsf{Asm}(A)$ and $\mathsf{RT}(A)$, there are enough projectives, and the projectives are, up to isomorphism, precisely the partitioned assemblies. Moreover, every assembly X embeds into the projective assembly $\nabla|X|$ via the unit of $\Gamma \dashv \nabla$. □

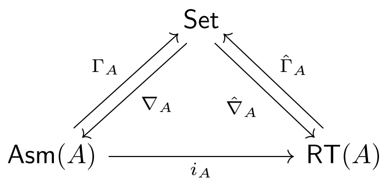
Example 3.2.20. By Example 3.2.14, $\mathsf{Asm}(TA)$ is the reg/lex completion of $\mathsf{Asm}(A)$. This is an example of the phenomenon that, when $\mathcal{C}$ is already regular, $\mathcal{C}_{\mathrm{reg/lex}}$ need not be equivalent to $\mathcal{C}$ itself. This is due to the fact that the inclusion $\mathsf{REG} \to \mathsf{LEX}$ is not *full*.

Remark 3.2.21. As we have stated it here, Corollary 3.2.19 requires the Axiom of Choice, because it requires Proposition 3.2.16. There are results very similar to Corollary 3.2.19, however, that do *not* require AC; see [Hof04] and [Fre14].

3.3 Functors between realizablity toposes

In this section, we study functors between categories of assemblies and realizability toposes. On the one hand, every partial applicative morphism $A \multimap B$ yields functors $\mathsf{Asm}(A) \to \mathsf{Asm}(B)$ and $\mathsf{RT}(A) \to \mathsf{RT}(B)$. On the other hand, as Longley first showed in the absolute case, we can characterize exactly which functors arise in this way.

In this section and the next, we will be working with multiple PCAs simultaneously. To avoid confusion, we will provide the functors introduced in Section 3.1 with subscripts as in this diagram:



3.3.1 Functors arising from morphisms of PCAs

In [FvO14, Theorem 2.2], it is shown that morphisms of PCAs $TA \to TB$ correspond to certain left exact functors $\mathsf{Asm}(A) \to \mathsf{Asm}(B)$. In the relative context, we will be a bit more liberal and allow the morphism of PCAs to assume values in DB, rather than TB. First, we will assign, to each morphism of PCAs $TA \to DB$, a left exact functor $\mathsf{Asm}(A) \to \mathsf{Asm}(B)$. Note that, if $f\colon TA \to DB$ is a morphism of PCAs, then f can be extended to a morphism $DA \to DB$ by simply setting $f(\emptyset) = \emptyset$. Thus, we see that $\mathsf{OPCA}(TA, DB)$ is isomorphic to the preorder of $f \in \mathsf{OPCA}(DA, DB)$ such that $f(\emptyset) = \emptyset$. If we regard f as a partial applicative morphism $DA \multimap B$, then this is equivalent to saying that $\operatorname{dom} f \subseteq TA$. We will say that such an f is **bottom-preserving**, and we denote the preorder consisting of all bottom-preserving f by $\mathsf{OPCA}(DA, DB)_{\mathrm{bp}}$, so that $\mathsf{OPCA}(TA, DB) \cong \mathsf{OPCA}(DA, DB)_{\mathrm{bp}}$.

Construction 3.3.1. For a morphism $f \in \mathsf{OPCA}(DA, DB)_{\mathrm{bp}}$, we define the functor $\mathsf{Asm}_0(f)\colon \mathsf{Asm}(A) \to \mathsf{Asm}(B)$ as follows:

- If X is an assembly over A, then $\mathsf{Asm}_0(f)(X)$ is the assembly over B defined by:

  $$|\mathsf{Asm}_0(f)(X)| = \{x \in |X| : f(E_X(x)) \neq \emptyset\} = \{x \in |X| : E_X(x) \in \operatorname{dom} f\},$$

  and $E_{\mathsf{Asm}_0(f)}(x) = f(E_X(x))$ for $x \in |\mathsf{Asm}_0(f)(X)|$.
- If $g\colon X \to Y$ is a morphism of assemblies over A, then $\mathsf{Asm}_0(f)(g)$ is the restriction of $g\colon |X| \to |Y|$ to $|\mathsf{Asm}_0(f)(X)| \subseteq |X|$.

Proposition 3.3.2. *If $f \in \mathsf{OPCA}(DA, DB)_{\mathrm{bp}}$, then $\mathsf{Asm}_0(f)$ is a well-defined left exact functor that satisfies $\mathsf{Asm}_0(f) \circ \nabla_A \cong \nabla_B$. This is part of a functor*

$$\mathsf{Asm}_0\colon \mathsf{OPCA}(DA, DB)_{\mathrm{bp}} \to \mathsf{LEX}(\mathsf{Asm}(A), \mathsf{Asm}(B)).$$

Moreover, the assignment Asm_0 is functorial, that is, we have $\mathsf{Asm}_0(\mathrm{id}_{DA}) = \mathrm{id}_{\mathsf{Asm}(A)}$ and $\mathsf{Asm}_0(gf) = \mathsf{Asm}_0(g) \circ \mathsf{Asm}_0(f)$.

Proof. Let $t, u \in B^{\#}$ realize f as a partial applicative morphism $DA \multimap B$. For the sake of readability, we will write F instead of $\mathsf{Asm}_0(f)$.

First of all, we need to see that that F is well-defined. If $g\colon X \to Y$ is a morphism of assemblies over A, then the restriction of g to $|FX|$ should land in $|FY|$, and this should yield a morphism of assemblies $FX \to FY$ over B. Let $r \in A^{\#}$ track g. If $x \in |X|$, then we have

$$u(t \cdot f(\downarrow\{r\}) \cdot f(E_X(x))) \preceq u \cdot f(r \cdot E_X(x)) \subseteq f(E_Y(g(x))).$$

Now, if $r' \in f(\downarrow\{r\}) \cap B^{\#}$ and $s = \lambda^* x.u(trx) \in B^{\#}$, then we get $s \cdot f(E_X(x)) \subseteq f(E_Y(g(x)))$. In particular, if $E_X(x) \in \operatorname{dom} f$, then $E_Y(g(x)) \in \operatorname{dom} f$ as well, so g indeed restricts to a map $|FX| \to |FY|$, and s tracks this map as a morphism of assemblies $FX \to FY$. It is immediate that F preserves identities and composition, so F is indeed a functor.

If X is a set, then $F(\nabla_A X) = (X, x \mapsto f(A))$, and since $f(A) \in (DB)^{\#}$, we see that $F(\nabla_A X)$ is isomorphic to the constant object $\nabla_B X$, so we indeed have $F\nabla_A \cong \nabla_B$. In particular, F preserves the terminal object. For binary products, let X and Y be assemblies over A. By the universal property of the product, we have that $|F(X \times Y)| \subseteq |FX \times FY|$, and that this inclusion is a morphism of assemblies $F(X \times Y) \to FX \times FY$. Conversely, suppose that $(x, y) \in |FX \times FY| = |FX| \times |FY|$. Then $E_{FX \times FY}(x, y) = \mathsf{p} \cdot f(E_X(x)) \cdot f(E_Y(y))$, so

$$\begin{aligned} & t(t \cdot f(\downarrow\{\mathsf{p}\}) \cdot (\mathsf{p}_0 \cdot E_{FX \times FY}(x, y))) \cdot (\mathsf{p}_1 \cdot E_{FX \times FY}(x, y)) \\ \preceq\ & t(t \cdot f(\downarrow\{\mathsf{p}\}) \cdot f(E_X(x))) \cdot f(E_Y(y)) \\ \preceq\ & t \cdot f(\mathsf{p} \cdot E_X(x)) \cdot f(E_Y(y)) \\ \subseteq\ & f(\mathsf{p} \cdot E_X(x) \cdot E_Y(y)) \\ =\ & f(E_{X \times Y}(x, y)). \end{aligned}$$

If $p \in f(\downarrow\{\mathsf{p}\}) \cap B^{\#}$, then the element $s = \lambda^* x.t(tp(\mathsf{p}_0 x))(\mathsf{p}_1 x)$ of $B^{\#}$ satisfies $s \cdot E_{FX \times FY}(x, y) \subseteq f(E_{X \times Y}(x, y))$. In particular, we see that $E_{X \times Y}(x, y) \in \operatorname{dom} f$ for all $(x, y) \in |FX \times FY|$, so we get $|FX| \times |FY| = |F(X \times Y)|$, and $s \in B^{\#}$ tracks the identity as a morphism $FX \times FY \to F(X \times Y)$. This shows that F preserves binary products. The proof that F preserves equalizers is easy, and we omit it.

Now let us first describe the action of Asm_0 on inequalities between morphisms. Suppose we have another $f' \in \mathsf{OPCA}(DA, DB)_{\mathrm{bp}}$ such that $f \leq f'$ is realized by $s \in B^{\#}$; we will write F' for $\mathsf{Asm}_0(f')$. If X is an assembly over A, then we have $s \cdot f(E_X(x)) \subseteq f'(E_X(x))$ for all $x \in |X|$. This shows that $|FX| \subseteq |F'X|$ and that $s \in B^{\#}$ tracks the inclusion $|FX| \subseteq |F'X|$ as a morphism of assemblies $FX \hookrightarrow F'X$. These constitute the required natural transformation $\mathsf{Asm}_0(f \leq f') \colon F \Rightarrow F'$, and it is obvious that this makes Asm_0 into a functor $\mathsf{OPCA}(DA, DB)_{\mathrm{bp}} \to \mathsf{LEX}(\mathsf{Asm}(A), \mathsf{Asm}(B))$.

Finally, the functoriality of Asm_0 is easy (but see also Remark 3.3.3 below) and left to the reader. □

Remark 3.3.3. Even though the definition of $\mathsf{Asm}_0(f)$ works just as well if we do not require that f preserves $\emptyset$, we need this requirement to ensure the functoriality of Asm_0. Indeed, suppose we have $f \colon DA \to DB$ that sends some $\alpha \in TA$ to $\emptyset$, and a $g \colon DB \to DC$ such that $g(\emptyset) \neq \emptyset$. Let $1_\alpha \in \mathsf{Asm}(A)$ be the assembly such that $|1_\alpha| = \{*\}$ and $E_{1_\alpha}(*) = \alpha$. Then $|\mathsf{Asm}_0(gf)(1_\alpha)| = \{*\}$, but $|\mathsf{Asm}_0(g)(\mathsf{Asm}_0(f)(1_\alpha))| = \emptyset$!

Even though $\mathsf{Asm}_0(f)$ commutes with the constant objects functors, it does not necessarily commute with the forgetful functors Γ. On the other hand, the inclusions $|\mathsf{Asm}_0(f)(X)| \subseteq |X|$ constitute a natural transformation $\Gamma_B \circ \mathsf{Asm}_0(f) \Rightarrow \Gamma_A$. The following result is now immediate.

Proposition 3.3.4. *Let $f \in \mathsf{OPCA}(DA, DB)_{\mathrm{bp}}$. Then $\Gamma_B \circ \mathsf{Asm}_0(f) \Rightarrow \Gamma_A$ is an isomorphism if and only if* $\operatorname{dom} f = TA$.

Proof. If $\operatorname{dom} f = TA$ and $X \in \mathsf{Asm}(A)$, then $E_X(x) \in TA = \operatorname{dom} f$ for all $x \in |X|$, so $|\mathsf{Asm}_0(f)(X)| = |X|$. For the converse, recall the assembly T_A from Example 3.1.6(i). The fact that $|\mathsf{Asm}_0(f)(T_A)| = |T_A| = TA$ means precisely that $\operatorname{dom} f = TA$. □

Proposition 3.3.5. *Let $f \in \mathsf{OPCA}(DA, DB)$. Then f is a morphism of D-algebras if and only if: $f(\emptyset) = \emptyset$ and $\mathsf{Asm}_0(f)$ is a regular functor.*

Proof. We assume (per Lemma 2.3.5) that f preserves the order on the nose, and we write F for $\mathsf{Asm}_0(f)$. Recall from the proof of Proposition 3.1.2 that a morphism $e\colon X \to Y$ of $\mathsf{Asm}(A)$ is a regular epimorphism if and only if there exists an $r \in A^\#$ that $r \cdot E_Y(y) \subseteq \bigcup_{e(x)=y} E_X(x)$ for all $y \in |Y|$. We also note that f is a D-algebra morphism iff the diagram

$$\begin{array}{ccc} DDA & \xrightarrow{Df} & DDB \\ {\scriptstyle \bigcup'_A}\downarrow & & \downarrow{\scriptstyle \bigcup'_B} \\ DA & \xrightarrow{f} & DB \end{array}$$

commutes up to isomorphism. But since D is a KZ-monad, we always have $\bigcup'_B \circ Df \leq f \circ \bigcup'_A$, so f is a D-algebra morphism iff $f \circ \bigcup'_A \leq \bigcup'_B \circ Df$. The latter means that there exists a $v \in B^\#$ such that $v \cdot f\left(\bigcup \mathcal{A}\right) \subseteq \bigcup_{\alpha \in \mathcal{A}} f(\alpha)$ for all $\mathcal{A} \in DDA$.

First, suppose that such a $v \in B^\#$ exists. Applying this for $\mathcal{A} = \emptyset$ yields $v \cdot f(\emptyset) = v \cdot f(\bigcup \emptyset) \subseteq \bigcup \emptyset = \emptyset$, which can only be true if $f(\emptyset) = \emptyset$. Now suppose that $e\colon X \to Y$ is a regular epi, and that $r \in A^\#$ is such that $r \cdot E_Y(y) \subseteq \bigcup_{x \in |X|} E_X(x)$ for all $y \in |Y|$. Now let $y \in |Y|$ and take $\mathcal{A} \in DDA$ equal to ${\downarrow}\{E_X(x) \mid e(x) = y\}$. Since we assumed that f preserves the order on the nose, we have:

$$\begin{aligned} v(t \cdot f({\downarrow}\{r\}) \cdot f(E_Y(y))) \preceq v \cdot f(r \cdot E_Y(y)) \preceq v \cdot f\left(\bigcup \{E_X(x) \mid e(x) = y\}\right) \\ \simeq v \cdot f\left(\bigcup \mathcal{A}\right) \subseteq \bigcup_{\alpha \in \mathcal{A}} f(\alpha) = \bigcup_{e(x)=y} f(E_X(x)). \end{aligned}$$

Thus, if $r' \in f({\downarrow}\{r\}) \cap B^\#$, then $s = \lambda^* x.v(tr'x)$ satisfies $s \cdot f(E_Y(y)) \subseteq \bigcup_{e(x)=y} f(E_X(x))$, for all $y \in |Y|$. Applying this for the $y \in |FY|$, we see that $Fe\colon FX \to FY$ is also a regular epi.

Conversely, suppose that $f(\emptyset) = \emptyset$ and that F is regular. Consider the assemblies X and Y over A given by:

$$\begin{aligned} |X| &= \left\{\mathcal{A} \in DDA \mid \bigcup \mathcal{A} \neq \emptyset\right\} \quad \text{and} \quad E_X(\mathcal{A}) = \bigcup \mathcal{A}, \\ |Y| &= \{(\alpha, \mathcal{A}) \in TA \times |X| : \alpha \in \mathcal{A}\} \quad \text{and} \quad E_Y(\alpha, \mathcal{A}) = \alpha. \end{aligned}$$

There is an obvious projection $\pi\colon |Y| \to |X|$, and for all $\mathcal{A} \in |X|$, we have:

$$\bigcup_{(\alpha,\mathcal{A}) \in |Y|} E_Y(\alpha, \mathcal{A}) = \bigcup_{\alpha \in \mathcal{A} \cap TA} \alpha = \bigcup \mathcal{A} = E_X(\mathcal{A}),$$

which means that $\pi\colon Y \to X$ is a regular epimorphism of assemblies. Since we assumed that $\operatorname{dom} f \subseteq TA$, we have that $(F\pi)^{-1}(\mathcal{A}) = \{(\alpha, \mathcal{A}) \mid \alpha \in \mathcal{A} \cap \operatorname{dom} f\}$

for $\mathcal{A} \in |FX|$. The fact that $F\pi$ should be a regular epi thus means that there should be a $v \in B^{\#}$ such that:

$$v \cdot f\left(\bigcup \mathcal{A}\right) \subseteq \bigcup_{\alpha \in \mathcal{A} \cap \operatorname{dom} f} f(\alpha) = \bigcup_{\alpha \in \mathcal{A}} f(\alpha),$$

for all $\mathcal{A} \in |FX|$. If $\mathcal{A} \in DDA$ is outside $|FX|$, then we either have $\mathcal{A} \in |X|$, which implies that $f\left(\bigcup \mathcal{A}\right) = \emptyset$, or we have $\bigcup \mathcal{A} = \emptyset$, in which case also $f\left(\bigcup \mathcal{A}\right) = f(\emptyset) = \emptyset$. So we can conclude that $v \cdot f\left(\bigcup \mathcal{A}\right) \subseteq \bigcup_{\alpha \in \mathcal{A}} f(\alpha)$ for all $\mathcal{A} \in DDA$, as desired. □

As we know, D-algebra morphisms correspond to partial applictive morphisms. This gives the following construction.

Construction 3.3.6. For a partial applicative morphism $f\colon A \multimap B$ with corresponding $\tilde{f}\colon DA \to DB$, we define the functor $\mathsf{Asm}(f)\colon \mathsf{Asm}(A) \to \mathsf{Asm}(B)$ as $\mathsf{Asm}_0(\tilde{f})$.

Corollary 3.3.7. *For every partial applicative morphism $f\colon A \multimap B$, the functor $\mathsf{Asm}(f)$ is regular and satisfies $\mathsf{Asm}(f) \circ \nabla_A \cong \nabla_B$. This makes Asm into a 2-functor:*

$$\mathsf{Asm}\colon \mathsf{OPCA}_D \to \mathsf{REG},$$

Moreover, there is a canonical natural transformation $\Gamma_B \circ \mathsf{Asm}(f) \Rightarrow \Gamma_A$, which is an isomorphism if and only if f is total.

Note that, even though OPCA_D is merely a bicategory (Remark 2.3.4), we can still say that Asm is a 2-functor because both $\widetilde{(\cdot)}$ and Asm_0 preserve identities and composition on the nose. This has the following consequence: if $f\colon A \multimap B$ is a partial applicative morphism, then: $\mathsf{Asm}(f \circ \mathrm{id}_A) = \mathsf{Asm}(f) \circ \mathsf{Asm}(\mathrm{id}_A) = \mathsf{Asm}(f) \circ \mathrm{id}_{\mathsf{Asm}(A)} = \mathsf{Asm}(f)$. In other words, in the definition of $\mathsf{Asm}(f)$, it does not matter whether we use f itself or its 'order-preserving version' as defined in the proof of Lemma 2.3.5.

By using Proposition 3.2.6, we can extend this construction to realizability toposes 'for free'.

Construction 3.3.8. For a partial applicative morphism $f\colon A \multimap B$, we define $\mathsf{RT}(f)\colon \mathsf{RT}(A) \to \mathsf{RT}(B)$ as the essentially unique regular functor making the diagram

$$\begin{array}{ccc} \mathsf{Asm}(A) & \xrightarrow{\mathsf{Asm}(f)} & \mathsf{Asm}(B) \\ {\scriptstyle i_A}\downarrow & & \downarrow{\scriptstyle i_B} \\ \mathsf{RT}(A) & \xrightarrow{\mathsf{RT}(f)} & \mathsf{RT}(B) \end{array}$$

commute up to natural isomorphism.

By Proposition 3.2.6 and Corollary 3.3.7, this makes RT into a pseudofunctor $\mathsf{OPCA}_D \to \mathsf{REG}$ (but see also Remark 3.3.10 below). If $f\colon A \multimap B$, then writing $F = \mathsf{Asm}(f)$ and $\hat{F} = \mathsf{RT}(f)$, we see that

$$\hat{F} \circ \hat{\nabla}_A = \hat{F} \circ i_A \circ \nabla_A \cong i_B \circ F \circ \nabla_A \cong i_B \circ \nabla_B = \hat{\nabla}_B,$$

so $\hat{F}$ also commutes with the constant object functors. Moreover, we have

$$\hat{\Gamma}_B \circ \hat{F} \circ i_A \cong \hat{\Gamma}_B \circ i_B \circ F \cong \Gamma_B \circ F \Rightarrow \Gamma_A \cong \hat{\Gamma}_A \circ i_A,$$

so by Proposition 3.2.6, we get a lift $\hat{\Gamma}_B \circ \hat{F} \Rightarrow \hat{\Gamma}_A$. It is not true that this natural transformation is mono at each component. On the other hand, Proposition 3.2.6 does tell us that $\hat{\Gamma}_B \circ \hat{F} \Rightarrow \hat{\Gamma}_A$ is an isomorphism iff $\Gamma_B \circ F \Rightarrow \Gamma_A$ is an isomorphism, i.e., iff f is total.

Example 3.3.9. Recall from Proposition 2.4.1 that $\mathbf{1}$ denotes the zero object of OPCA. Its realizability tripos $\mathsf{P}_{\mathbf{1}}$ is simply $\mathrm{Sub}_{\mathsf{Set}}$, which means that $\mathsf{RT}(\mathbf{1}) \simeq \mathsf{Set}$, and also $\mathsf{Asm}(\mathbf{1}) \simeq \mathsf{Set}$. This leads to the following degenerate examples of Construction 3.3.6 and Construction 3.3.8. If A is a PCA, then we have $!\colon A \to \mathbf{1}$ and $¡\colon \mathbf{1} \to A$. Modulo the equivalances $\mathsf{RT}(\mathbf{1}) \simeq \mathsf{Asm}(\mathbf{1}) \simeq \mathsf{Set}$, we have $\mathsf{Asm}(!) \cong \Gamma_A$ and $\mathsf{RT}(!) \cong \hat{\Gamma}_A$, and similarly, $\mathsf{Asm}(¡) \cong \nabla_A$ and $\mathsf{RT}(¡) \cong \hat{\nabla}_A$.

Remark 3.3.10. We can give a slightly more explicit description of $\hat{F}$ using triposes. There is a regular transformation of triposes $\mathsf{P}_A \to \mathsf{P}_B$, which, for a set X, is given by:

$$\mathsf{P}_A X \simeq \mathrm{Sub}(\nabla_A X) \xrightarrow{F} \mathrm{Sub}(F(\nabla_A X)) \simeq \mathrm{Sub}(\nabla_B X) \simeq \mathsf{P}_B X.$$

It is easy to give an explicit description of this transformation: it sends $\phi \in P_A X$ to $\tilde{f}\phi \in \mathsf{P}_B X$. Now by Remark 3.1.25, we get a corresponding functor $\mathsf{RT}(A) \to \mathsf{RT}(B)$. This functor agrees with F on the assemblies, and must therefore be isomorphic to $\hat{F}$. If we take this to be the definition of $\mathsf{RT}(f)$, then we see that $\mathsf{RT}(f)$ actually becomes *2*-functiorial.

More generally, if $g\colon DA \to DB$ is any morphism of PCAs (but not necessarily a D-algebra morphism), then $g \circ (-)$ is a left exact (but not necessarily regular) transformation $\mathsf{P}_A \to \mathsf{P}_B$. We leave the proof of this to the reader; it is similar to the proof Proposition 3.3.2.

3.3.2 Left exact Γ- and ∇-functors

In the previous section, we defined, for morphisms between PCAs, corresponding functors between the categories of assemblies and the realizability toposes. These functors always commute with the constant object functors, and, in some circumstances, also with the Γ's or $\hat{\Gamma}$'s. Longley has shown [Lon94, Section 2.3] that, for absolute discrete PCAs, a regular functor $\mathsf{Asm}(A) \to \mathsf{Asm}(B)$ commutes with the ∇'s iff it commutes with the Γ's. This cannot be true in our setting, for if $f\colon A \multimap B$ is a partial applicative morphism which is not total, then $\mathsf{Asm}(f)\colon \mathsf{Asm}(A) \to \mathsf{Asm}(B)$ will commute with the ∇'s, but not with the Γ's. This is largely due to the fact that, for relative PCAs, the functor Γ is not the global sections functor (Lemma 3.1.5). The goal of this (rather technical) section is to explain to what extent Longley's results carry over to relative PCAs. We will show that, for any left exact functor $F\colon \mathsf{Asm}(A) \to \mathsf{Asm}(B)$, we have $F\nabla_A \cong \nabla_B$ iff there exists a natural transformation $\Gamma_B F \Rightarrow \Gamma_A$ that is injective at each component. Moreover, if $F, F' \colon \mathsf{Asm}(A) \to \mathsf{Asm}(B)$ are two such functors, then there exists at most one natural transformation $F \Rightarrow F'$, cf. [Lon94, Proposition 2.2.19].

Thus far, we have been saying that a functor 'commutes with the constant object functors'. We can be a bit more precise about this: let us denote by $\mathsf{Set}/\mathsf{LEX}$ the pseudocoslice of LEX under Set, viewed as a 2-category. That is:

- objects are left exact functors $F \colon \mathsf{Set} \to \mathcal{C}_F$, with $\mathcal{C}_F$ left exact;
- arrows from $f \colon F \to F'$ are left exact functors $G \colon \mathcal{C}_F \to \mathcal{C}_{F'}$, with a specified natural isomorphism $GF \cong F'$;
- if $G, G' \colon F \to F'$, then a 2-cell $G \to G'$ is a natural transformation $\mu \colon G \Rightarrow G'$ such that μF is the natural isomorphism $GF \cong F' \cong G'F$.

Similarly, we have the pseudoslice $\mathsf{LEX}/\mathsf{Set}$ over Set, and we have the pseudo-coslice $\mathsf{Set}/\mathsf{REG}$ and the pseudoslice $\mathsf{REG}/\mathsf{Set}$. With this notation, Asm_0 can be seen as a functor

$$\mathsf{OPCA}(DA, DB)_{\mathrm{bp}} \to (\mathsf{Set}/\mathsf{LEX})(\nabla_A, \nabla_B).$$

Moreover, Proposition 3.3.4 tells us that Asm_0 restricts to a functor

$$\mathsf{OPCA}(TA, TB) \to (\mathsf{LEX}/\mathsf{Set})(\Gamma_A, \Gamma_B).$$

Finally, it is easily checked that Asm and RT become pseudofunctors $\mathsf{OPCA}_D \to \mathsf{Set}/\mathsf{REG}$, and both restrict to pseudofunctors $\mathsf{OPCA}_T \to \mathsf{REG}/\mathsf{Set}$.

For our analysis of Longley's results in the relative setting, we need the following folklore lemma.

Lemma 3.3.11. *If $F \colon \mathsf{Set} \to \mathsf{Set}$ is a functor such that $F1 \cong 1$, then there is at most one natural transformation $\mathrm{id}_{\mathsf{Set}} \Rightarrow F$.*

Proof. Since $\mathrm{id}_{\mathsf{Set}} \cong \mathrm{Hom}(1, -)$, this follows from Yoneda's Lemma. □

Lemma 3.3.12. *Let A and B be PCAs and let $F \colon \mathsf{Asm}(A) \to \mathsf{Asm}(B)$ be left exact.*

(*i*) *There is a bijection*

$$\mathsf{Nat}(F\nabla_A, \nabla_B) \cong \mathsf{Nat}(\Gamma_B F, \Gamma_A),$$

natural in F.

(*ii*) *For $\kappa \colon F\nabla_A \Rightarrow \nabla_B$ with corresponding $\kappa' \colon \Gamma_B F \Rightarrow \Gamma_A$, the following are equivalent:*

(*a*) *κ is an isomorphism;*

(*b*) *κ' is pointwise injective;*

(*c*) *$\Gamma_B\kappa$ is an isomorphism, and F preserves prone morphisms.*

(iii) There is at most one isomorphism $F\nabla_A \cong \nabla_B$.

Now let $F' \colon \mathsf{Asm}(A) \to \mathsf{Asm}(B)$ *be another left exact functor, and assume that* $F\nabla_A \cong \nabla_B \cong F'\nabla_A$.

(iv) If $\mu \colon F \Rightarrow F'$, *then the following triangles commute:*

$$\begin{array}{ccc} F\nabla_A & \xrightarrow{\mu\nabla_A} & F'\nabla_A \\ & \searrow \quad \swarrow & \\ & \nabla_B & \end{array} \qquad \begin{array}{ccc} \Gamma_B F & \xrightarrow{\Gamma_B\mu} & \Gamma_B F' \\ & \searrow \quad \swarrow & \\ & \Gamma_A & \end{array}$$

(v) There exists at most one natural transformation $F \Rightarrow F'$.

Proof. (i). Given $\kappa \colon F\nabla_A \Rightarrow \nabla_B$, we define $\kappa' \colon \Gamma_B F \Rightarrow \Gamma_A$ as the transpose of

$$F \xRightarrow{F\eta} F\nabla_A\Gamma_A \xRightarrow{\kappa\Gamma_A} \nabla_B\Gamma_A$$

across the adjunction $\Gamma_B \dashv \nabla_B$. Conversely, given $\lambda \colon \Gamma_B F \Rightarrow \Gamma_A$, define $\lambda' \colon F\nabla_A \Rightarrow \nabla_B$ as the transpose of

$$\Gamma_B F\nabla_A \xRightarrow{\lambda\nabla_A} \Gamma_A\nabla_A = \!\!=\!\!= \mathrm{id}_{\mathsf{Set}}$$

across the adjunction $\Gamma_B \dashv \nabla_B$. A tedious but straightforward diagram chase shows that these two operations are inverse, and the naturality of these bijections is obvious.

(ii). First, we prove that (a) implies (b) and (c). Suppose that $\kappa \colon F\nabla_A \cong \nabla_B$. The associated κ' is equal to the composition

$$\Gamma_B F \xRightarrow{\Gamma_B F\eta} \Gamma_B F\nabla_A\Gamma_A \xRightarrow{\Gamma_B\kappa\Gamma_A} \Gamma_B\nabla_B\Gamma_A = \!\!=\!\!= \Gamma_A \tag{3.1}$$

Since η is pointwise mono and $\Gamma_B F$ is left exact, the first of these is pointwise injective. The other natural transformation is by assumption an isomorphism, so κ' is indeed pointwise injective. As for (c), it is immediate that $\Gamma_B\kappa$ is iso if κ is. Now suppose that $f \colon X \to Y$ in $\mathsf{Asm}(A)$ is prone. Then the naturality square

$$\begin{array}{ccc} \Gamma_B FX & \overset{\kappa'_X}{\hookrightarrow} & \Gamma_A X \\ {\scriptstyle \Gamma_B Ff}\downarrow & & \downarrow{\scriptstyle \Gamma_A f} \\ \Gamma_B FY & \overset{\kappa'_Y}{\hookrightarrow} & \Gamma_A Y \end{array} \tag{3.2}$$

is a pullback diagram. Indeed, (3.1) allows us to write this as the composition of two pullback squares. Moreover, the outer square in:

$$\begin{array}{ccccc} FX & \xrightarrow{F\eta_X} & F\nabla_A\Gamma_A X & \xrightarrow{\kappa_{\Gamma_A X}} & \nabla_B\Gamma_A X \\ \downarrow{\scriptstyle Ff} & & \downarrow{\scriptstyle F\nabla_A\Gamma_A f} & & \downarrow{\scriptstyle \nabla_B\Gamma_A f} \\ FY & \xrightarrow[F\eta_Y]{} & F\nabla_A\Gamma_A Y & \xrightarrow[\kappa_{\Gamma_A Y}]{} & \nabla_B\Gamma_A Y \end{array}$$

is a pullback, since both inner squares are. By the definition of κ', this outer square can also be decomposed as:

$$\begin{array}{ccccc}
FX & \xrightarrow{\eta_{FX}} & \nabla_B\Gamma_B FX & \xhookrightarrow{\nabla_B\kappa'_X} & \nabla_B\Gamma_A X \\
\big\downarrow{\scriptstyle Ff} & & \big\downarrow{\scriptstyle \nabla_B\Gamma_B Ff} & & \big\downarrow{\scriptstyle \nabla_B\Gamma_A f} \\
FY & \xrightarrow[\eta_{FY}]{} & \nabla_B\Gamma_B FY & \xhookrightarrow[\nabla_B\kappa'_Y]{} & \nabla_B\Gamma_A Y
\end{array}$$

which must therefore also be a pullback square. By (3.2), the right-hand square is a pullback as well, so it follows that the left-hand square is also a pullback, i.e., Ff is prone, as desired.

Now we show that (c) implies (a). If $\Gamma_B\kappa$ is iso, then κ itself is iso iff $F\nabla_A X$ is a constant object, for every set X. In other words, F should preserve constant objects. But this is immediate, since an assembly Y is constant iff the unique arrow $Y \to 1$ is prone.

Finally, we show that (b) implies (a). The proof is a generalization of the proof of [Lon94, Proposition 2.3.3], and in particular, it is nonconstructive. Suppose that we have $\kappa' \colon \Gamma_B F \Rightarrow \Gamma_A$ consisting of injections. It is easy to show that, by replacing F by an isomorphic functor, we may assume w.l.o.g. that $\Gamma_B FX \subseteq \Gamma_A X$ for every $X \in \mathsf{Asm}(A)$, i.e., that κ' consists of *subset inclusions*.

First, we show that κ' is the identity on constant objects. Let X be a set. For all $x \in X$, we have a global section $x\colon 1 \to \nabla_A X$ in $\mathsf{Asm}(A)$. Since $\Gamma_B F$ and Γ_A are both left exact, we know that κ'_1 is the unique isomorphism $\Gamma_B F1 \cong 1 \cong \Gamma_A 1$. The naturality square

$$\begin{array}{ccc}
\Gamma_B F1 & \xrightarrow{\Gamma_B Fx} & \Gamma_B F\nabla_A X \\
{\scriptstyle \kappa'_1}\big\downarrow & & \big\downarrow{\scriptstyle \kappa'_{\nabla_A X}} \\
\Gamma_A 1 & \xrightarrow{\Gamma_A x} & \Gamma_A\nabla_A X
\end{array}$$

now tells us that $x \in \Gamma_B F\nabla_A X \subseteq \Gamma_A\nabla_A X = X$. But this holds for all $x \in X$, so $\Gamma_B F\nabla_A = \Gamma_A\nabla_A = \mathrm{id}_{\mathsf{Set}}$, and $\kappa'\nabla_A$ is the identity natural transformation on $\mathrm{id}_{\mathsf{Set}}$. We have $\Gamma_B\kappa = \kappa'\nabla_A$, so in order to show that κ is iso, it remains to prove that $F\nabla_A X$ is a constant object, for every set X.

If X is empty, then this is clear, so suppose that X is not empty. Let Y be a set such that $\operatorname{card} Y > \operatorname{card} X \times \operatorname{card} B^{\#}$. For all $y \in Y$, consider the global section $y\colon 1 \to Y$ in Set. Since $\Gamma_B F\nabla_A$ is the identity on Set, we know that the underlying function of $F\nabla_A y\colon F\nabla_A 1 \to F\nabla_A Y$ is again $y\colon 1 \to Y$. Moreover, $F\nabla_A 1$ is a terminal object of $\mathsf{Asm}(B)$, so $F\nabla_A y$ is a global section of $F\nabla_A Y$, which means that $E_{F\nabla_A Y}(y) \cap B^{\#}$ is nonempty. For $s \in B^{\#}$, let us write $Y_s = \{y \in Y \mid s \in E_{F\nabla_A Y}(y)\}$. By what we have just shown, $Y = \bigcup_{s\in B^{\#}} Y_s$, so by our choice of Y, there must be an $s_0 \in B^{\#}$ such that $\operatorname{card} Y_{s_0} > \operatorname{card} X$. Now we pick a surjection $e\colon Y \twoheadrightarrow X$ such that $e(Y_{s_0}) = X$. Then the underlying function of $F\nabla_A e\colon F\nabla_A Y \to F\nabla_A X$ is e. If t tracks $F\nabla_A e$, then by our choice of e, we have $ts_0 \in \bigcap_{x\in X} E_{F\nabla_A X}(x)$. Since $ts_0 \in B^{\#}$, this shows that $F\nabla_A X$

is indeed constant, as desired.

(iii). If $\kappa\colon F\nabla_A \Rightarrow \nabla_B$ is iso, then $\Gamma_B\kappa^{-1}\colon \mathrm{id}_{\mathsf{Set}} = \Gamma_B\nabla_B \Rightarrow \Gamma_B F\nabla_A$ is uniquely determined in virtue of Lemma 3.3.11. Since Γ_B is faithful, this determines κ^{-1}, which determines κ.

(iv). The naturality of the bijection in (i) means precisely that the two triangles are equivalent, so it suffices to prove that the left-hand triangle commutes. First of all, we observe that this triangle consists of isomorphisms. Since $\Gamma_B\nabla_B = \mathrm{id}_{\mathsf{Set}}$, Lemma 3.3.11 tells us that the image of the triangle under Γ_B must commute. Since Γ_B is faithful, the result follows.

(v). Consider the right-hand triangle in (iv). Since $\Gamma_B F' \Rightarrow \Gamma_A$ is pointwise injective, this triangle determines $\Gamma_B\mu$. Since Γ_B is faithful, this determines μ. □

The most important consequences of this lemma may be summarized as follows. First of all, an arrow $\nabla_A \to \nabla_B$ of $\mathsf{Set}/\mathsf{LEX}$, or $\Gamma_A \to \Gamma_B$ of $\mathsf{LEX}/\mathsf{Set}$, is determined by its functor part, since the accompanying natural isomorphism is unique anyway. Moreover, there are full inclusions

$$(\mathsf{LEX}/\mathsf{Set})(\Gamma_A, \Gamma_B) \hookrightarrow (\mathsf{Set}/\mathsf{LEX})(\nabla_A, \nabla_B) \hookrightarrow \mathsf{LEX}(\mathsf{Asm}(A), \mathsf{Asm}(B)),$$

and the first two categories are actually preorders.

3.3.3 Longley's correspondence theorem generalized

In this section, we generalize Longley's result [Lon94, Theorem 2.3.4] characterizing regular functors $\mathsf{Asm}(A) \to \mathsf{Asm}(B)$ that commute with the ∇'s, or equivalently for absolute PCAs, with the Γ's. We also generalize the result by Faber and Van Oosten [FvO14, Theorem 2.2] that characterizes left exact such functors.

Theorem 3.3.13. *Let A and B be PCAs. Then*

$$\mathsf{Asm}_0\colon (DA, DB)_{\mathrm{bp}} \to (\mathsf{Set}/\mathsf{LEX})(\nabla_A, \nabla_B)$$

is an equivalence of categories.

Proof. First, we show that Asm_0 is essentially surjective. Let $F\colon \mathsf{Asm}(A) \to \mathsf{Asm}(B)$ be a left exact functor such that $F\nabla_A \cong \nabla_B$. We know that there is a pointwise injective natural transformation $\kappa'\colon \Gamma_B F \Rightarrow \Gamma_A$. By replacing F by an isomorphic functor, we can assume that κ' consists of subset inclusions, i.e., $|FX| \subseteq |X|$ for all $X \in \mathsf{Asm}(A)$. By Lemma 3.3.12(ii), F preserves prone morphisms. Moreover, in the proof of Lemma 3.3.12(ii) (diagram (3.2)), we saw that, if $g\colon X \to Y$ is prone, the naturality square

$$\begin{array}{ccc} |FX| & \subseteq & |X| \\ {\scriptstyle Fg}\downarrow & & \downarrow{\scriptstyle g} \\ |FY| & \subseteq & |Y| \end{array}$$

is a pullback diagram.

Consider the assembly T_A as in Example 3.1.6. Then FT_A is an assembly over B with $|FT_A| \subseteq |T_A| = TA$ and $E_{FT_A}\colon |FT_A| \to TB$. We can extend E_{FT_A} to a function $f\colon DA \to DB$ such that $\operatorname{dom} f := \{\alpha \in DA \mid f(\alpha) \neq \emptyset\} = |TF_A|$. We claim that f is a morphism of PCAs $DA \to DB$.

First of all, if $\alpha \in (DA)^{\#}$, then $1 \stackrel{\alpha}{\to} T_A$ is a morphism of assemblies. Since $F1 \cong 1$, we also get a global section $1 \cong F1 \to FT_A$ whose underlying function is $1 \stackrel{\alpha}{\to} TA$. This implies that $f(\alpha) \cap B^{\#} = E_{FT_A}(\alpha) \cap B^{\#}$ is nonempty, so $f(\alpha) \in (DB)^{\#}$. In order to show that f preserves application up to a realizer, consider the prone subobject $P \hookrightarrow T_A \times T_A$ given by $|P| = \{(\alpha, \alpha') \mid \alpha\alpha'\downarrow\} \subseteq TA \times TA$ and $E_P(\alpha, \alpha') = \mathsf{p}\alpha\alpha'$. Then

$$\begin{array}{ccc} |FP| & \subseteq & |P| \\ \downarrow & & \downarrow \\ |FT_A| \times |FT_A| & \subseteq & |T_A| \times |T_A| \end{array}$$

is a pullback, which means that

$$|FP| = |P| \cap (|FT_A| \times |FT_A|) = \{(\alpha, \alpha') \in \operatorname{dom} f \times \operatorname{dom} f \mid \alpha\alpha'\downarrow\}.$$

Moreover, since $FP \hookrightarrow FT_A \times FT_A$ is prone, we can also assume that

$$E_{FP}(\alpha, \alpha') = E_{FT_A \times FT_A}(\alpha, \alpha') = \mathsf{p} \cdot f(\alpha) \cdot f(\alpha')$$

for $(\alpha, \alpha') \in |FP|$. Now we observe that $\mathrm{app}\colon P \to T_A$ defined by $\mathrm{app}(\alpha, \alpha') = \alpha\alpha'$ is a morphism of assemblies, tracked by $\lambda^* x.\mathsf{p}_0 x(\mathsf{p}_1 x)$. Now $F\mathrm{app}\colon FP \to FT_A$, which is the restriction of app to $|FP|$, must be a morphism of assemblies as well. Let $t \in B^{\#}$ be a tracker of $F\mathrm{app}$, and define $t' = \lambda^* xy.t(\mathsf{p}xy) \in B^{\#}$. Then for all $(\alpha, \alpha') \in |FP|$, we have

$$t' \cdot f(\alpha) \cdot f(\alpha) \preceq t \cdot (\mathsf{p} \cdot f(\alpha) \cdot f(\alpha')) \simeq t \cdot E_{FP}(\alpha, \alpha') \subseteq E_{FT_A}(\alpha\alpha') = f(\alpha\alpha'),$$

so f preserves the application up to t'. Similarly, we can define an assembly O by $|O| = \{(\alpha, \alpha') \mid \alpha \subseteq \alpha'\} \subseteq TA \times TA$ and $E_O(\alpha, \alpha') = \alpha$. Then the first projection $\pi_0\colon |O| \to TA$ is clearly a prone morphism of assemblies $O \to T_A$. This implies that $|FO| = \{(\alpha, \alpha') \in |O| : \alpha \in \operatorname{dom} f\}$, and the fact that $F\pi_0$ is prone means that we can assume that $E_{FO}(\alpha, \alpha') = E_{T_A}(\alpha) = f(\alpha)$ for $(\alpha, \alpha') \in |FO|$. The second projection $\pi_1\colon |O| \to TA$ is a morphism of assemblies, for it is tracked by i. Now $F\pi_1$, which is the restriction of π_1 to

$|FO|$, is a morphism of assemblies $FO \to FT_A$. If $u \in B^\#$ tracks $F\pi_1$, then f preserves the order up to u. This completes the proof that f is a morphism of PCAs.

We have $f(\emptyset) = \emptyset$ by definition, so it remains to show that $\mathsf{Asm}_0(f) \cong F$. If X is an assembly, then $E_X : |X| \to TA$ is a prone morphism of assemblies $X \to T_A$. First of all, this implies that $|FX| = \{x \in |X| : E_X(x) \in |FT_A|\} = \{x \in |X| : E_X(x) \in \mathrm{dom}\, f\}$. Moreover, the fact that $FE_X : FX \to FT_A$ is also prone means precisely that the identity on $|FX|$ is an isomorphism of assemblies $FX \cong \mathsf{Asm}_0(f)(X)$. This completes the proof of essential surjectivity.

It remains to show that Asm_0 is fully faithful. Since $(\mathsf{Set}/\mathsf{LEX})(\nabla_A, \nabla_B)$ is a preorder, it suffices to prove the following: if $f, f' \in \mathsf{OPCA}(DA, DB)_{\mathrm{bp}}$ and there exists a natural transformation $\mu : \mathsf{Asm}_0(f) \Rightarrow \mathsf{Asm}_0(f')$, then $f \le f'$. By Lemma 3.3.12(iv), μ_X must be the inclusion $|\mathsf{Asm}_0(f)(X)| \subseteq |\mathsf{Asm}_0(f')(X)|$. Now any tracker of μ_{T_A} is easily seen to be a realizer of $f \le f'$. This completes the proof. □

Combining this theorem with the results from Section 3.3.1 yields the following corollaries.

Corollary 3.3.14. *Let A and B be PCAs. Then the following are equivalences of categories:*

(i) $\mathsf{Asm}_0 : \mathsf{OPCA}(TA, TB) \to (\mathsf{LEX}/\mathsf{Set})(\Gamma_A, \Gamma_B)$ *(cf. [FvO14, Theorem 2.2]);*

(ii) $\mathsf{Asm} : \mathsf{OPCA}_D(A, B) \to (\mathsf{Set}/\mathsf{REG})(\nabla_A, \nabla_B)$*;*

(iii) $\mathsf{Asm} : \mathsf{OPCA}_T(A, B) \to (\mathsf{REG}/\mathsf{Set})(\Gamma_A, \Gamma_B)$ *(cf. [Lon94, Theorem 2.3.4]).*

In particular, $\mathsf{Asm} : \mathsf{OPCA}_D \to \mathsf{Set}/\mathsf{REG}$ *and* $\mathsf{Asm} : \mathsf{OPCA}_T \to \mathsf{REG}/\mathsf{Set}$ *are local equivalences.*

Proof. For (i): by Proposition 3.3.5, the equivalence from Theorem 3.3.13 restricts to an equivalence between the full preorder of $\mathsf{OPCA}(DA, DB)_{\mathrm{bp}}$ consisting of the morphisms f such that $\mathrm{dom}\, f = TA$, and $(\mathsf{LEX}/\mathsf{Set})(\Gamma_A, \Gamma_B)$. But the former is simply $\mathsf{OPCA}(TA, TB)$, as desired. The equivalence in (ii) is a composition of equivalences:

$$\mathsf{OPCA}_D(A, B) \xrightarrow{f \mapsto \tilde{f}} D\text{–}\mathsf{Alg}(DA, DB) \xrightarrow{\mathsf{Asm}_0} (\mathsf{Set}/\mathsf{REG})(\nabla_A, \nabla_B),$$

Here the second map is an equivalence by Theorem 3.3.13 and Corollary 3.3.7. Finally, Corollary 3.3.7 also tells us that the equivalence in (ii) restricts to the equivalence in (iii). □

Finally, the result in (ii) above can be extended to realizability toposes.

Corollary 3.3.15. *The 2-functor* $\mathsf{RT} : \mathsf{OPCA}_D \to \mathsf{Set}/\mathsf{REG}$ *is a local equivalence, that is, for PCAs A and B, we have an equivalence:*

$$\mathsf{RT} : \mathsf{OPCA}_D \to (\mathsf{Set}/\mathsf{REG})(\hat{\nabla}_A, \hat{\nabla}_B).$$

Proof. By Corollary 3.3.14(ii) and Proposition 3.2.6, it suffices to show that every regular functor $F\colon \mathsf{RT}(A) \to \mathsf{RT}(B)$ with $F\hat{\nabla}_A \cong \hat{\nabla}_B$ restricts to a functor $\mathsf{Asm}(A) \to \mathsf{Asm}(B)$. But this is clear, since the assemblies are the subobjects of the constant objects, F commutes with the constant object functors, and F preserves monos. □

Note that we cannot get an extention of Corollary 3.3.14(iii) in the same way. Even though, by Proposition 3.2.6, every functor in $(\mathsf{REG}/\mathsf{Set})(\Gamma_A, \Gamma_B)$ lifts to a functor in $(\mathsf{REG}/\mathsf{Set})(\hat{\Gamma}_A, \hat{\Gamma}_B)$, we cannot conversely guarantee that every functor in $(\mathsf{REG}/\mathsf{Set})(\hat{\Gamma}_A, \hat{\Gamma}_B)$ restricts to a functor on the assemblies.

As a consequence of Corollary 3.3.15, we see that every regular transformation between realizability triposes is actually of the form $\tilde{f} \circ (-)$ for a partial applicative morphism f. In fact, we have the following result.

Proposition 3.3.16. *Denote by* $\mathsf{Trip}_{\mathrm{lex}}(\mathsf{Set})$ *the preorder-enriched subcategory of* $\mathsf{Trip}(\mathsf{Set})$ *containing only the left exact transformations of triposes. Then* $f \mapsto f \circ (-)$ *is an equivalence of categories*

$$\mathsf{OPCA}(DA, DB) \to \mathsf{Trip}_{\mathrm{lex}}(\mathsf{Set})(\mathsf{P}_A, \mathsf{P}_B).$$

Proof. Since id_{DA} is a generic predicate for P_A, any transformation $\mathsf{P}_A \to \mathsf{P}_B$ is given, up to isomorphism, by $f \circ (-)$ for some function $f\colon DA \to DB$. So what we need to show is that f is a morphism of PCAs iff $f \circ (-)$ is a left exact transformation, and that $f \leq f'$ iff $f \circ (-) \leq f' \circ (-)$. The proof of this is similar to the proof of Theorem 3.3.13, and we omit it. □

3.4 Geometric morphisms

In the previous section, we studied functors between categories of assemblies and realizability toposes. Between toposes, one usually considers not just functors, but geometric morphisms. In this section, we study geometric morphisms between realizability toposes, and in particular, we determine for which partial applicative morphism f, the functor $\mathsf{RT}(f)$ is the inverse image of geometric morphism. This continues research from [HvO03], [Joh13] and [FvO14], but as we shall see, not all the results from these papers generalize to the case of relative PCAs.

3.4.1 Geometric morphisms between categories of assemblies

Before we turn our attention to realizability toposes, we first study geometric morphisms between categories of assemblies. Since categories of assemblies are not in general toposes, let us, for the sake of completeness, define what we mean by this.

Definition 3.4.1. *Let $\mathcal{C}$ and $\mathcal{D}$ be left exact categories. A **geometric morphism** $f\colon \mathcal{C} \to \mathcal{D}$ is an adjunction:*

$$\mathcal{C} \underset{f_*}{\overset{f^*}{\underset{\longrightarrow}{\overset{\longleftarrow}{\bot}}}} \mathcal{D}$$

*where f^* preserves finite limits. The functors f_* and f^* are called the **direct image** and **inverse image** of f, respectively. We say that f is a **geometric inclusion** if the counit of $f^* \dashv f_*$ is an isomorphism; equivalently, if f_* is fully faithful. Morover, if $f, g\colon \mathcal{C} \to \mathcal{D}$ are geometric morphisms, then a **geometric transformation** $f \Rightarrow g$ is a natural transformation $f^* \Rightarrow g^*$; equivalently, a natural transformation $g_* \Rightarrow f_*$. We write* GEOM *for the 2-category of left exact categories, geometric morphisms and geometric transformations.*

Theorem 3.4.2. *Let $f\colon A \multimap B$ be an applicative morphism. Then $\mathsf{Asm}(f)$ has a right adjoint if and only if f is computationally dense. Moreover, if $\mathsf{Asm}(f)$ has a right adjoint G, then $\Gamma_A G \cong \Gamma_B$ if and only if f is dense.*

Proof. Write $\tilde{f}\colon DA \to DB$ for the D-algebra morphism corresponding to f, and write $F = \mathsf{Asm}(f) = \mathsf{Asm}_0(\tilde{f})$. First, suppose that f is c.d. Then by Theorem 2.3.14, $\tilde{f}$ has a right adjoint $g\colon DB \to DA$, and by Lemma 2.3.16(i), $g(\emptyset) = \emptyset$. Thus, we can define $G = \mathsf{Asm}_0(g)$, and the adjunction $\tilde{f} \dashv g$ yields an adjunction $F \dashv G$.

Conversely, suppose that F has a right adjoint G. The isomorphism $\Gamma_B F \cong \Gamma_A$ yields an isomorphism $G\nabla_B \cong \nabla_A$ by taking right adjoints. Moreover, G, being a right adjoint, is left exact. Thus, by Theorem 3.3.13, we must have $G \cong \mathsf{Asm}_0(g)$ for some $g \in \mathsf{OPCA}(DA, DB)_{\mathrm{bp}}$. Moreover, by Lemma 3.3.12(iv), the unit and counit of $F \dashv G$ must be arrows of $(\mathsf{Set}/\mathsf{LEX})(\nabla_A, \nabla_B)$, so by Theorem 3.3.13 again, the adjunction $F \dashv G$ must arise from an adjunction $\tilde{f} \dashv g$. By Theorem 2.3.14, this implies that f is c.d.

Finally, we know from Proposition 3.3.4 that $\Gamma_A G \cong \Gamma_B$ iff $\mathrm{dom}(g) = TB$. By Lemma 2.3.16(ii), this is true iff f is dense. □

Using the category GEOM, we get the following corollary of Theorem 3.4.2.

Corollary 3.4.3. *Let $\mathsf{OPCA}_{T,\mathrm{cd}}$ denote the category of PCAs and c.d. applicative morphisms. There is a local equivalence $\mathsf{OPCA}_{T,\mathrm{cd}}^{\mathrm{op}} \to \mathsf{Set}/\mathsf{GEOM}$ that sends a PCA A to the geometric inclusion $\Gamma_A \dashv \nabla_A\colon \mathsf{Set} \hookrightarrow \mathsf{Asm}(A)$.*

Proof. For a c.d. applicative morphism $f\colon A \multimap B$, we get a geometric morphism $\mathsf{Asm}(B) \to \mathsf{Asm}(A)$ whose inverse image is $\mathsf{Asm}(f)$, and every inequality $f \leq f'$ yields a geometric transformation $\mathsf{Asm}(f) \Rightarrow \mathsf{Asm}(f')$. In order to see that this defines a local equivalence, it suffices to show that, for any commutative triangle

$$\begin{array}{ccc} & \mathsf{Set} & \\ \swarrow & & \searrow \\ \mathsf{Asm}(B) & \xrightarrow{\quad g \quad} & \mathsf{Asm}(A) \end{array}$$

in GEOM, we have $g^* \cong \mathsf{Asm}(f)$ for some $f\colon A \multimap B$. But g^* is regular, and the triangle tells us that $\Gamma_A g \cong \Gamma_B$, so this follows from Corollary 3.3.14(iii). □

3.4.2 Geometric morphisms between realizability toposes

In this section, we study geometric morphisms between realizability toposes. This topic is most naturally approached using tripos theory. Let us first define the notion of a geometric morphism of triposes.

Definition 3.4.4. *If* P *and* Q *are triposes over* $\mathcal{C}$*, then a* ***geometric morphism*** $f\colon \mathsf{P} \to \mathsf{Q}$ *is an adjunction*

$$\mathsf{P} \underset{f_*}{\overset{f^*}{\underset{\perp}{\rightleftarrows}}} \mathsf{Q}$$

where f^* *is left exact. The transformations* f_* *and* f^* *are called the* ***direct image*** *and* ***inverse image*** *of* f*, respectively. We say that* f *is a* ***geometric inclusion*** *if* $f^*f_* \cong \mathrm{id}_{\mathsf{P}}$.

If $f\colon \mathsf{P} \to \mathsf{Q}$ is a geometric morphism, then the fact that f_* is a right adjoint implies that f_* is also left exact. Moreover, the fact that f^* is a left adjoint implies that it commutes with $\exists$, and thus that f^* is a regular transformation. In particular, we get a regular functor $\overline{f^*}\colon \mathcal{C}[\mathsf{Q}] \to \mathcal{C}[\mathsf{P}]$. Since f_* is not necessarily regular, we cannot lift it to a functor between toposes as in Remark 3.1.25. In spite of this, Pitts was able to obtain the following result.

Proposition 3.4.5 (Pitts). *If* $f\colon \mathsf{Q} \to \mathsf{P}$ *is a regular transformation of triposes over* $\mathcal{C}$*, then* f *has a right adjoint* g *iff* $\overline{f}\colon \mathcal{C}[\mathsf{Q}] \to \mathcal{C}[\mathsf{P}]$ *has a right adjoint* $\overline{g}$*. In this case,* $\overline{f} \dashv \overline{g}\colon \mathcal{C}[\mathsf{P}] \to \mathcal{C}[\mathsf{Q}]$ *is an inclusion iff* $f \dashv g\colon \mathsf{P} \to \mathsf{Q}$ *is an inclusion.*

Proof. See [Pit81, Theorem 4.8 and Remark 5.2(iii)]. □

In view of Proposition 3.3.16, geometric morphisms $\mathsf{P}_B \to \mathsf{P}_A$ correspond to arrows $DB \to DA$ of $\mathsf{OPCA}_{\mathrm{adj}}$. The following result is now immediate.

Theorem 3.4.6. *Let* $f\colon A \hookrightarrow\!\!\!\!\multimap B$ *be a partial applicative morphism. Then* $\mathsf{RT}(f)$ *has a right adjoint iff* f *is c.d. If* $\mathsf{RT}(f)$ *has a right adjoint* $\hat{G}$*, then* f *is total iff the triangle*

$$\begin{array}{ccc} & \mathsf{Set} & \\ \swarrow & & \searrow \\ \mathsf{RT}(B) & \xrightarrow[\mathsf{RT}(f) \dashv \hat{G}]{} & \mathsf{RT}(A) \end{array}$$

commutes.

Proof. By Proposition 3.4.5, $\mathsf{RT}(f)$ has a right adjoint iff $\tilde{f} \circ (-)\colon \mathsf{P}_A \to \mathsf{P}_B$ has a right adjoint. This is true iff $\tilde{f}\colon DA \to DB$ has a right adjoint, and by Corollary 2.3.15, this is true iff f is c.d. Finally, the triangle says that $\hat{\Gamma}_B \circ \mathsf{RT}(f) \cong \hat{\Gamma}_A$, and we know that this holds iff f is total, by the observation preceding Example 3.3.9. □

Let us consider the case where $f\colon A \multimap B$ is a c.d. applicative morphism, and write $F = \mathsf{Asm}(f)$ and $\hat{F} = \mathsf{RT}(f)$. Then by the theorem above, we have a geometric morphism $\hat{F} \dashv \hat{G}\colon \mathsf{RT}(B) \to \mathsf{RT}(A)$ that commutes with the inclusions of $\neg\neg$-sheaves, and thus, we have $\hat{\Gamma}_B\hat{F} \cong \hat{\Gamma}_A$ and $\hat{G}\hat{\nabla}_B \cong \hat{\nabla}_A$. In particular, $\hat{G}$ restricts to a functor $G\colon \mathsf{Asm}(B) \to \mathsf{Asm}(A)$. Since the inclusion of assemblies into the realizability topos is fully faithful, the adjunction $\hat{F} \dashv \hat{G}$ restricts to an adjunction $F \dashv G$, so G is the right adjoint of F as provided by Theorem 3.4.2.

Since $\hat{F}$ is defined as $\mathsf{RT}(f)$, we have $\hat{F}\hat{\nabla}_A \cong \hat{\nabla}_B$. Now we expect the 'remaining' isomorphism $\hat{\Gamma}_A\hat{G} \cong \hat{\Gamma}_B$ to hold iff $\Gamma_A G \cong \Gamma_B$ holds, i.e., iff f is dense. However, we do not directly get this from Proposition 3.2.6, since $\hat{\Gamma}_A\hat{G}$ could fail to be regular. In the case of absolute PCAs, we automatically have $\hat{\Gamma}_A\hat{G} \cong \hat{\Gamma}_B$. Indeed, in this case, $\hat{\Gamma}$ is the global sections functor, so

$$\hat{\Gamma}_A\hat{G}X \cong \mathrm{Hom}(1, \hat{G}X) \cong \mathrm{Hom}(\hat{F}1, X) \cong \mathrm{Hom}(1, X) \cong \hat{\Gamma}_B X.$$

We can adjust this argument to obtain the following result.

Proposition 3.4.7. *Let $f\colon A \multimap B$ be a c.d. applicative morphism, and let $\hat{G}\colon \mathsf{RT}(B) \to \mathsf{RT}(A)$ be the right adjoint of $\mathsf{RT}(f)$. Then $\hat{\Gamma}_A\hat{G} \cong \hat{\Gamma}_B$ if and only if f is dense.*

Proof. In view of Theorem 3.4.2 and the remarks above, it suffices to show that, if we have a natural isomorphism $\hat{\Gamma}_A\hat{G}X \cong \hat{\Gamma}_B X$ for *assemblies* X, then we get an isomorphism $\hat{\Gamma}_A\hat{G} \cong \hat{\Gamma}_B$ on the whole of $\mathsf{RT}(B)$.

In order to show this, let us give the following alternative description of $\hat{\Gamma}$. Let X be an object of the realizability topos, and suppose that $U \hookrightarrow X$ is a subobject of X, with U subterminal but nonempty, i.e., $U \not\cong 0$. Applying $\hat{\Gamma}$ yields $1 \cong \hat{\Gamma}U \hookrightarrow \hat{\Gamma}X$, that is, an element of $\hat{\Gamma}X$. Moreover, two nonempty subterminal subobjects U and V of X yield the same element of $\hat{\Gamma}X$ iff $\hat{\Gamma}U \cap \hat{\Gamma}V \neq \emptyset$ (as subobjects of $\hat{\Gamma}_X$). Since $\hat{\Gamma}$ is left exact, and preserves and reflects the initial object, this is true iff $U \cap V \not\cong 0$. Moreover, if $[x]_{\sim_X} \neq \emptyset$ is an element of $\hat{\Gamma}_X$, then the predicate ϕ defined by

$$\phi(x') = \begin{cases} x' \sim_X x' & \text{if } x' \in [x]_{\sim_X}; \\ \emptyset & \text{otherwise,} \end{cases}$$

is strict and relational for X, and it defines a nonempty subterminal subobject of X whose image under $\hat{\Gamma}$ corresponds to $[x]_{\sim_X}$. Thus, we see that $\hat{\Gamma}X$ is naturally isomorphic to the set of nonempty subterminal subobjects of X, modulo the equivalence relation $U \cap V \not\cong 0$.

If $\hat{\Gamma}_A\hat{G}X \cong \hat{\Gamma}_B X$ for assemblies X, then in particular, we have $\hat{G}U \cong 0$ iff $U \cong 0$ for subterminals U in $\mathsf{RT}(B)$. This means that composition with $\hat{G}$ sends nonempty subterminal subobjects of X to nonempty subterminal subobjects of $\hat{G}X$, and that this operation preserves and reflects the equivalence relation $U \cap V \not\cong 0$. Thus we get a natural transformation $\hat{\Gamma}_B \Rightarrow \hat{\Gamma}_A\hat{G}$ consisting of injections. In order to show that $\hat{\Gamma}_B X \to \hat{\Gamma}_A\hat{G}X$ is also surjective, suppose we have $U \hookrightarrow \hat{G}X$ with U subterminal and nonempty. Then by transposing across the adjunction $\hat{F} := \mathsf{RT}(f) \dashv \hat{G}$, we get $\hat{F}U \hookrightarrow X$, and $\hat{F}U$ is also nonempty and subterminal. Moreover, we have a commutative diagram

$$\begin{array}{ccc} U & & \\ \downarrow & \searrow & \\ \hat{G}\hat{F}U & \hookrightarrow & \hat{G}X \end{array}$$

so $\hat{G}(\hat{F}U \hookrightarrow X)$ and $U \hookrightarrow \hat{G}X$ represent the same element of $\hat{\Gamma}_A\hat{G}X$, which completes the proof. □

Finally, let us note the following corollary to Theorem 3.4.6; this generalizes [Joh13, Corollary 3.3] to relative ordered PCAs.

Corollary 3.4.8. *Suppose that $f\colon A \to B$ and $g\colon B \rightharpoonup A$ form a coreflection in OPCA_D, i.e., $fg \leq \mathrm{id}_B$ and $gf \simeq \mathrm{id}_A$. Then $\mathsf{RT}(f) \dashv \mathsf{RT}(g)$ is a local geometric morphism $\mathsf{RT}(B) \to \mathsf{RT}(A)$.*

Note that we write $f\colon A \to B$ since the adjunction $f \dashv g$ in OPCA_D implies that f is projective anyway.

Proof of Corollary 3.4.8. We have $\mathsf{RT}(g) \circ \mathsf{RT}(f) \cong \mathsf{RT}(gf) \cong \mathsf{RT}(\mathrm{id}_A) \cong \mathrm{id}_{\mathsf{RT}(A)}$, so it remains to show that $\mathsf{RT}(g)$ has a right adjoint. But $gf \simeq \mathrm{id}_A$ implies that g is c.d., so this follows from Theorem 3.4.6. □

3.4.3 Inclusions and surjections

Consider a c.d. partial applicative morphism $f\colon A \rightharpoonup B$, so that $\mathsf{RT}(f)$ is the inverse image part of a geometric morphism $\mathsf{RT}(B) \to \mathsf{RT}(A)$. In this section, we investigate when this geometric morphism is an inclusion resp. a surjection.

Let us start with studying inclusions. The following proposition is a generalization of [vO08, Proposition 2.6.2] and [FvO14, Corollary 2.5]. We note that our equation (im) below is called (in) in [vO08]. The advantage of our formulation of (in) is that it does not mention the witness m of (cdm') from Section 2.3.2, which can, in general, be somewhat complicated.

Proposition 3.4.9. *Let $f\colon A \rightharpoonup B$ be a c.d. partial applicative morphism. Then the following are equivalent:*

(i) $\mathsf{RT}(f)$ *is the inverse image of a geometric inclusion* $\mathsf{RT}(B) \to \mathsf{RT}(A)$*;*

(ii) the right adjoint g of $\tilde{f}\colon DA \to DB$ satisfies $\tilde{f}g \simeq \mathrm{id}_{DB}$;

(iii) there exists a partial applicative morphism $h\colon B \rightharpoonup A$ such that $fh \simeq \mathrm{id}_B$;

(iv) there exist $s, e \in B^\#$ satisfying:

$$\forall b \in B \, \exists a \in A (eb \in f(a) \wedge s \cdot f(a) \subseteq \downarrow\{b\}); \qquad \text{(in)}$$

(v) *whenever* $m \in B^{\#}$ *satisfies* (cdm'), *there exists an* $e \in B^{\#}$ *satisfying:*

$$\forall b \in B\, \exists a \in A\, (eb \in f(a) \wedge m \cdot f(a) \subseteq {\downarrow}\{b\}). \tag{im}$$

Moreover, if f *is total, then (i)-(v) are also equivalent to:*

(vi) $\mathsf{Asm}(f)$ *is the inverse image of a geometric inclusion* $\mathsf{Asm}(B) \to \mathsf{Asm}(A)$*;*

(vii) *there exists an applicative morphism* $h\colon B \multimap A$ *such that* $fh \simeq \mathrm{id}_B$*;*

Finally, if f *is total and (i)-(vii) hold, then* f *is dense.*

Proof. (i)⇔(ii). We know from Proposition 3.4.5 that (i) holds iff the geometric morphism of triposes $\tilde{f} \circ (-) \dashv g \circ (-)\colon \mathsf{P}_B \to \mathsf{P}_A$ is an inclusion. In view of Proposition 3.3.16, this is true iff the counit of $\tilde{f} \dashv g$ is an isomorphism, i.e., (ii) holds.

(ii)⇒(iii). As in the proof of Corollary 2.3.15, this follows by defining $h = g\delta'_B\colon B \to DA$.

(iii)⇒(iv). Let $s, e \in B^{\#}$ realize $fh \leq \mathrm{id}_B$ and $\mathrm{id}_B \leq fh$ respectively. If $b \in B$, then eb is defined and an element of $fh(b) = \bigcup_{a \in h(b)} f(a)$, which means there must be an $a \in h(b)$ such that $eb \in f(a)$. Moreover, since $f(a) \subseteq fh(b)$, we have $s \cdot f(a) \preceq s \cdot fh(b) \subseteq \mathrm{id}_B(b) = {\downarrow}\{b\}$.

(iv)⇒(v). Let $m \in B^{\#}$ satisfy (cdm') and take $s, e \in B^{\#}$ such that (in) holds. According to (cdm'), we may find an $r \in A^{\#}$ such that

$$\forall a \in A\, (m \cdot f(ra) \preceq s \cdot f(a)).$$

Let f preserve application up to $t \in B^{\#}$, pick an element $r' \in f(r) \cap B^{\#}$ and define $e' = \lambda^* x.tr'x$.

Now let $b \in B$, and find an $a \in A$ such that eb is defined and in $f(a)$, and $s \cdot f(a) \subseteq {\downarrow}\{b\}$. It follows that $m \cdot f(ra) \preceq s \cdot f(a) \subseteq {\downarrow}\{a\}$, so $m \cdot f(ra)$ is defined and a subset of ${\downarrow}\{b\}$. Moreover, we have $e'b \preceq tr'b$, which is defined and an element of $f(ra)$. This means that $e'b$ is also defined and an element of $f(ra)$. which establishes (im).

(v)⇒(ii). Recall from (2.5) that $g\colon DB \to DA$ can be defined by

$$g(\beta) = {\downarrow}\{a \in A \mid m \cdot f(a) \subseteq \beta\},$$

where m satisfies (cdm'). If $e \in B^{\#}$ satisfies (im), then it is immediate that e realizes $\mathrm{id}_{DB} \leq \tilde{f}g$.

For the remainder of the proof, suppose that f is total. Then the right adjoint of $\mathsf{Asm}(f) = \mathsf{Asm}_0(\tilde{f})$ is $\mathsf{Asm}_0(g)$, and by Theorem 3.3.13, (ii) and (vi) are equivalent. Moreover, we clearly have that (vii) implies (iii). Conversely, suppose that (i)-(v) hold, and pick $m \in B^{\#}$ satisfying (cdm'). By (im), m also satisfies (dm'), so f is dense. By Lemma 2.3.16(ii), we have $\operatorname{dom} g = TB$. Thus, the $h\colon B \multimap A$ defined by $g\delta'_A$ is in fact total, so (vii) follows. □

Remark 3.4.10. Note that, if f is not total, then $\mathsf{RT}(f)$ does not commute with the Γ-functors, which means that the right adjoint G of $\mathsf{RT}(f)$ does not commute with the constant object functors. Thus, we do not know whether G restricts to a functor $\mathsf{Asm}(B) \to \mathsf{Asm}(A)$, and therefore, we do not get that $\mathsf{Asm}(f)$ has a right adjoint as asserted in (vi) above.

Now let us turn to surjections. We have not defined geometric surjections yet, because the equivalent definitions for geometric morphisms between toposes are not equivalent for categories that are merely left exact. We summarize the situation in the following lemma, which is of course well-known, but included for the sake of completeness.

Lemma 3.4.11. *Let $F\colon \mathcal{C} \to \mathcal{D}$ be a left exact functor between left exact categories. Then the following are equivalent:*

(i) F is conservative;

(ii) F reflects the order on all subobject posets;

and if $\mathcal{C}$ is balanced, then these are also equivalent to:

(iii) F is faithful.

Proof. (i)⇒(ii). Let U and V be two subobjects of X such that $FU \leq FV$ as subobjects of FX. Consider the two pullback squares

$$\begin{array}{ccc} U \cap V & \hookrightarrow & V \\ \downarrow & & \downarrow \\ U & \hookrightarrow & X \end{array} \qquad\qquad \begin{array}{ccc} FU \cap FV & \hookrightarrow & FV \\ \downarrow & & \downarrow \\ FU & \hookrightarrow & FX \end{array}$$

Since F preserves pullbacks, the right-hand square is the image of the left-hand square under F. By our assumption, the inclusion $FU \cap FV \hookrightarrow FU$ is an isomorphism, which means that $U \cap V \hookrightarrow U$ is iso as well, i.e., $U \leq V$.

(ii)⇒(i). Consider an arrow $f\colon X \to Y$ such that Ff is iso. Let $K \hookrightarrow X \times X$ be the kernel pair of f and let $\Delta \hookrightarrow X \times X$ be the diagonal on X. Since F preserves finite limits, we know that FK is the kernel pair of Ff and that $F\Delta$ is the diagonal on FX. Since Ff is mono, we have $FK \cong F\Delta$ as subobjects of $FX \times FX$, which implies that $K \cong \Delta$ as subobjects of $X \times X$, i.e., f is mono. This means we can regard f as (representing) a subobject of Y. But we also know that Ff is the maximal subobject of FY, which means that f is already the maximal subobject of Y, i.e., f is iso.

(i)⇒(iii) in fact holds without the assumption that $\mathcal{C}$ is balanced. Indeed, let $f, g\colon X \to Y$ be arrows such that $Ff = Fg$. If $m\colon U \hookrightarrow X$ is the equalizer of f and g, then $Fm\colon FU \hookrightarrow FX$ is the equalizer of Ff and Fg. By our assumption, Fm is an isomorphism, implying that m is also iso, so $f = g$.

Finally, assume that $\mathcal{C}$ is balaned.

(iii)⇒(i). Let f be an arrow in $\mathcal{C}$ such that Ff is iso. Then in particular, Ff is both mono and epi. Since F is faithful, this implies that f is mono and epi as well; and since $\mathcal{C}$ is balanced, this means that f is iso. □

Toposes are balanced, and we say that a geometric morphism f between toposes is a surjection iff f^* satisfies the equivalent conditions of Lemma 3.4.11. By (iii), these are also equivalent to requiring the unit of $f^* \dashv f_*$ to be pointwise mono.

For realizability toposes, we are thus interested in the following question: if $f\colon A \hookrightarrow\!\!\circ\, B$ is a partial applicative morphism, when is $\mathsf{RT}(f)\colon \mathsf{RT}(A) \to \mathsf{RT}(B)$ conservative? First, let us note that, if $\mathsf{RT}(f)$ is conservative, then f must be total. Indeed, suppose that we have an $a \in A$ such that $f(a) = \emptyset$, and consider the assembly 1_a given by $|1_a| = \{*\}$ and $E_{1_a}(*) = \downarrow\{a\}$. Then we have $\mathsf{RT}(f)(1_a) \cong 0 \cong \mathsf{RT}(f)(0)$, but $1_a \not\cong 0$, so $\mathsf{RT}(f)$ is not conservative. Therefore, we can restrict our attention to applicative morphisms $f\colon A \multimap B$.

Proposition 3.4.12. *Let $f\colon A \multimap B$ be an applicative morphism. Then the following are equivalent:*

(i) $\mathsf{RT}(f)$ *is conservative;*

(ii) $\mathsf{Asm}(f)$ *is conservative;*

(iii) for every set X, the map $\tilde{f} \circ (-)\colon \mathsf{P}_A X \to \mathsf{P}_B X$ reflects the order;

(iv) f satisfies:

$$\forall s \in B^{\#}\, \exists r \in A^{\#}\, \forall a \in A \forall \alpha \in DA\, (s \cdot f(a) \subseteq \tilde{f}(\alpha) \to ra \in \alpha). \qquad \text{(sur)}$$

If (i)-(iv) hold, then $\tilde{f}$ is a pseudomono in OPCA*, and f is a pseudomono in* OPCA_D*. Moreover, if f is c.d., then (i)-(iv) are also equivalent to:*

(v) the right adjoint $g\colon DB \to DA$ of $\tilde{f}$ satisfies $g\tilde{f} \simeq \mathrm{id}_{DA}$.

Moreover, if f is projective, say $f \simeq \delta_B f_0$ for some $f_0\colon A \to B$, then (i)-(iv) are also equivalent to:

(vi) f_0 satisfies:

$$\forall s \in B^{\#}\, \exists r \in A^{\#}\, \forall a, a' \in A\, (s \cdot f_0(a) \le f_0(a') \to ra \le a'). \qquad \text{(surp)}$$

Finally, if f is projective <u>and</u> c.d., then (i)-(vi) are also equivalent to:

(vii) the right adjoint $h\colon B \hookrightarrow\!\!\circ\, A$ of f in OPCA_D *satisfies $hf \simeq \mathrm{id}_A$.*

Proof. First, let us show that (i), (ii) and (iii) are equivalent. Since $\mathsf{Asm}(f)$ is the restriction of $\mathsf{RT}(f)$ to $\mathsf{Asm}(A)$, and $\mathsf{Asm}(A)$ is closed under subobjects, it is immediate (given Lemma 3.4.11) that (i) implies (ii). Moreover, if (ii) holds, then in particular, $\mathsf{Asm}(f)\colon \mathrm{Sub}_{\mathsf{Asm}(A)}(\nabla_A X) \to \mathrm{Sub}_{\mathsf{Asm}(B)}(\nabla_B X)$ reflects the order, which yields (iii). Finally, if (iii) holds, then (i) follows from Lemma 3.2.3 and Lemma 3.4.11.

Before we continue, let us show that (iii) implies that $\tilde{f}$ is a pseudomono in OPCA. Let $k, \ell\colon C \to DA$ be morphisms of PCAs such that $\tilde{f}k \le \tilde{f}\ell$. Viewing

k and ℓ as elements of $\mathsf{P}_A C$ yields $\tilde{f}k \vdash \tilde{f}\ell$, so by (iii), $k \vdash \ell$, which means that $k \leq \ell$ in OPCA. The argument that f is a pseudomono in OPCA_D, given that (iii) holds, is exactly the same.

(iii)⇒(iv). Suppose for simplicity that f preserves the order on the nose. Let $s \in B^\#$ and consider the set

$$X = \{(a, \alpha) \in A \times DA \mid s \cdot f(a) \subseteq \tilde{f}(\alpha)\}.$$

Define $\phi, \psi \colon X \to DA$ by $\phi(a, \alpha) = {\downarrow}\{a\}$ and $\psi(a, \alpha) = \alpha$. Then s realizes $\tilde{f}\phi \vdash \tilde{f}\psi$. This implies that $\phi \vdash \psi$ as well, and if $r \in A^\#$ realizes this inequality, then (sur) is clearly satisfied.

(iv)⇒(iii). Let X be a set, and suppose we have $\phi, \psi \colon X \to DA$ such that $\tilde{f}\phi \vdash \tilde{f}\psi$. Let $s \in B^\#$ realize this inequality, and find $r \in A^\#$ as in (sur). We claim that r realizes $\phi \vdash \psi$. Indeed, let $x \in X$, let $a \in \phi(x)$ and define $\alpha = \psi(x)$. Then $f(a) \subseteq \bigcup_{a' \in \phi(x)} f(a') = \tilde{f}(\phi(x))$, which means that $s \cdot f(a)$ defined and $s \cdot f(a) \subseteq \tilde{f}(\psi(x)) = \tilde{f}(\alpha)$ We conclude that ra is defined and an element of $\alpha = \psi(x)$, as desired.

Now suppose that f is c.d., and let $g \colon DB \to DA$ be the right adjoint of $\tilde{f}$. If (i)-(iv) hold, then f is a pseudomono. Since $\tilde{f} \dashv g$, we have $\tilde{f}g\tilde{f} \simeq \tilde{f}$, so it follows that $g\tilde{f} \simeq \mathrm{id}_{DA}$, which is (v). Conversely, it is immediate that (v) implies (iii).

Now suppose that $f \simeq \delta_B f_0$ is projective; for simplicity, we assume that $f = \delta_B f_0$ on the nose.

(iv)⇒(vi). Let $s \in B^\#$ and take $r \in A^\#$ as in (sur). Suppose that we have $a, a' \in A$ with $s \cdot f_0(a) \leq f_0(a')$. Then it follows that:

$$s \cdot f(a) \simeq s \cdot {\downarrow}\{f_0(a)\} \simeq {\downarrow}\{s \cdot f_0(a)\} \subseteq {\downarrow}\{f_0(a')\} = f(a') \subseteq \tilde{f}({\downarrow}\{a'\}).$$

So by (sur), with $\alpha = {\downarrow}\{a'\}$, we have $ra \in {\downarrow}\{a'\}$, i.e., $ra \leq a'$, as desired.

(vi)⇒(iv). Let $s \in B^\#$ and take $r \in A^\#$ as in (surp). Suppose that we have $a \in A$ and $\alpha \in DA$ with $s \cdot f(a) \subseteq \tilde{f}(\alpha)$. Then in particular, we have:

$$s \cdot f_0(a) \in s \cdot {\downarrow}\{f_0(a)\} = s \cdot f(a) \subseteq \tilde{f}(\alpha),$$

which means that there is an $a' \in \alpha$ such that $s \cdot f_0(a) \in f(a') = {\downarrow}\{f_0(a')\}$. This means that $s \cdot f_0(a) \leq f_0(a')$, so by (surp), we have $ra \leq a' \in \alpha$, hence $ra \in \alpha$ as well, as desired.

Finally, suppose that f is projective and c.d. If $h \colon B \multimap A$ is the right adjoint of f, then we must also have $\tilde{f} \dashv \tilde{h}$, and it immediately follows that (v) and (vii) are equivalent. □

Remark 3.4.13. Part of Proposition 3.4.12 holds more generally in the context of triposes. More precisely, if $f \colon \mathsf{P} \to \mathsf{Q}$ is a geometric morphism of triposes over $\mathcal{C}$, then the following (corresponding to (i), (iii) and (v) above) are equivalent:

- $\overline{f^*} \colon \mathcal{C}[\mathsf{Q}] \to \mathcal{C}[\mathsf{P}]$ is conservative, i.e., the corresponding $\mathcal{C}[\mathsf{P}] \to \mathcal{C}[\mathsf{Q}]$ is a surjection;

- $f_X^* \colon \mathsf{Q}X \to \mathsf{P}X$ reflects the order, for every object X of $\mathcal{C}$;
- $f_* f^* \simeq \mathrm{id}_{\mathsf{Q}}$.

This is related to the discussion of connected geometric morphisms between triposes and their corresponding toposes in [Bie08, Chapter 1]; in particular, Proposition 1.16 of that chapter is a special case of the equivalence we have stated here.

Let us note the following remarkable corollary to Proposition 3.4.12.

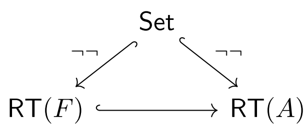
Corollary 3.4.14. *Let $f \colon A \multimap B$ be a projective and c.d. applicative morphism. If the corresponding geometric morphism $\mathsf{RT}(B) \to \mathsf{RT}(A)$ is surjective, then it is local.*

Proof. By (vii) of Proposition 3.4.12, f and its right adjoint h form a coreflection in OPCA_D, so this follows from Corollary 3.4.8. □

To close this chapter, let us revisit the examples from Section 2.3.4. All of these involved adjunctions in OPCA arising from c.d. partial applicative morphisms, and thus should give rise to geometric morphisms between the realizability toposes.

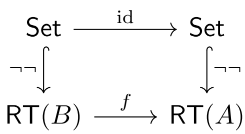
Example 3.4.15. In Example 2.3.17 and Example 2.3.19, we considered a PCA A and a filter F, and we saw that the inclusion $F \hookrightarrow A$ has a right adjoint, yielding a coreflection in OPCA_D. In particular, we have a local geometric morphism $\mathsf{RT}(A) \to \mathsf{RT}(F)$. This was first observed, for the case $F = A^{\#}$, in [ABS02].

Since $\mathsf{RT}(A) \to \mathsf{RT}(F)$ is local, we also have a geometric inclusion $\mathsf{RT}(F) \hookrightarrow \mathsf{RT}(A)$, whose inverse image is $\mathsf{RT}(f)$, with $f \colon A \leftharpoondown\!\!\circ\, F$ the right adjoint of $F \hookrightarrow A$. If $F \neq A$, then f is not total, which means that the diagram



does **not** commute.

[Joh13, Lemma 2.1] states that, for absolute discrete PCAs A and B and any geometric morphism $f \colon \mathsf{RT}(B) \to \mathsf{RT}(A)$, the square



is a pullback in the category of toposes and geometric morphisms. The current example shows that this result does not extend to relative PCAs.

Example 3.4.16. In Example 2.3.20, we saw that $\iota\colon A \to A[a]$ has a right adjoint, yielding a reflection in OPCA_T. In particular, we have a geometric inclusion $\mathsf{RT}(A[a]) \hookrightarrow \mathsf{RT}(A)$. In the next chapter, we will see that this inclusion is *open*.

Example 3.4.17. By Example 2.3.21, there are local geometric morphisms $\mathsf{RT}(\mathcal{K}_2) \to \mathcal{E}\!f\!f$ and $\mathsf{RT}(\mathcal{B}) \to \mathcal{E}\!f\!f$. Moreover, Example 2.3.22 provides a local geometric morphism $\mathsf{RT}(\mathcal{P}\omega) \to \mathsf{RT}(\mathcal{K}_2)$, and a geometric morphism $\mathsf{RT}(\mathcal{P}\omega) \to \mathsf{RT}(\mathcal{B})$.

Chapter 4

Products and Slicing

As we have seen in the previous chapter, each PCA A gives rise to a category of assemblies $\mathsf{Asm}(A)$ and a realizability topos $\mathsf{RT}(A)$. In this chapter, we will consider products and slices of these categories. We know that toposes are closed under both products and slices, the latter being known as the 'fundamental theorem of topos theory'. Thus, it makes sense to ask the following questions:

Question 4.0.1. *Are realizability toposes closed under (finite) products and/or slicing? That is, is A and B are PCAs, is $\mathsf{RT}(A) \times \mathsf{RT}(B)$ again of the form $\mathsf{RT}(C)$ for some PCA C? And if $I \in \mathsf{RT}(A)$, is $\mathsf{RT}(A)/I$ again of the form $\mathsf{RT}(C)$ for some PCA C?*

In Section 4.1, we present a number of reasons why the answers to these questions should be *no*. Therefore, most of our attention will be directed at the following question.

Question 4.0.2. *Can we isolate a natural class of 'realizability-like' toposes that includes all realizability toposes and is closed under (finite) products and/or slicing?*

Of course, we may ask similar questions for categories of assemblies instead of realizability toposes. In Section 4.1 below, we shall see that the most prominent obstacle for taking products of realizability toposes is that it involves a 'base change'. Before turning to Question 4.0.2 above, we will first, in Section 4.2, consider a construction that avoids base change, namely the construction of *pushouts* of realizability toposes over Set. For these pushouts, we will pose and answer the analoga of Question 4.0.1 and Question 4.0.2 above. This material is adapted from [Zoe21a, Section 6].

Next, in Section 4.3, we introduce a general framework for dealing with products and slicing, which is adapted from W. Stekelenburg's PhD thesis [Ste13]. It has the following two characteristics. First, we construct PCAs over more general 'base categories' than the category of sets. Second, we allow a very liberal notion of 'relative PCA', given by the notion of an *external filter*. Let us note a new differences between [Ste13] and our setup in Section 4.3 below:

- Whereas the 'base categories' in [Ste13] are Heyting categories, our approach will be more general and allow regular categories as base categories.
- At the same time, our approach is slightly less general than [Ste13], because we will require that the external filters only contain inhabited objects. The reason for doing so is that it allows a much simpler description of categories of assemblies, and working with this restriction is sufficient for our purposes. In fact, [Ste13] shows that our constraint does not really impose a restriction of the categories of assemblies we consider.
- Moreover, whereas [Ste13] introduces a notion of applicative morphism (and even of *partial* applicative morphism) over a fixed base, we will define a notion of applicative morphism that allows a change of base. We will see that this has applications in the constructions of products and slices.

Finally, in Section 4.4, we apply the framework from Section 4.3 to the problem of products and slices. We will direct most of our attention to categories of assemblies, but obtain some partial results for realizability toposes as well. Finally, we discuss the relation between products and slices, and computational density. The material from Section 4.3 and Section 4.4 is adapted from [Zoe20].

4.1 Products and slicing: obstacles

In Section 3.2 above, we established various properties shared by all realizability toposes. Thus, if realizability toposes are to be closed under products and/or slicing, then taking products and/or slicing should preserve these properties. In this section, we present one property that is not perserved by either products or slicing, and another one which is not preserved by slicing. These can be seen as 'obstacles' to taking products and slices of realizability toposes, and these obstacles must be overcome if we are to answer Question 4.0.2 on the previous page.

Base change

Consider two PCAs A and B. Then the two geometric inclusions $\neg\neg\colon \mathsf{Set} \to \mathsf{RT}(A)$ and $\neg\neg\colon \mathsf{Set} \to \mathsf{RT}(B)$ combine into a geometric inclusion:

$$\mathsf{Set}^2 \underset{\hat{\nabla}_A \times \hat{\nabla}_B}{\overset{\hat{\Gamma}_A \times \hat{\Gamma}_B}{\rightleftarrows}} \mathsf{RT}(A) \times \mathsf{RT}(B) \tag{4.1}$$

which is again the inclusion of ¬¬-sheaves. Thus, we see that $\mathsf{RT}(A) \times \mathsf{RT}(B)$ lives 'over Set^2' rather than Set. Similarly, if I is an object of $\mathsf{RT}(A)$, then we get a canonical geometric inclusion:

$$\mathsf{Set}/\hat{\Gamma}_A I \underset{\hat{\eta}_I^* \circ \hat{\nabla}_A}{\overset{\hat{\Gamma}_A}{\rightleftarrows}} \mathsf{RT}(A)/I \tag{4.2}$$

where the direct image part consists of applying $\hat{\nabla}_A$, followed by pulling back along $\hat{\eta}_I \colon I \to \hat{\nabla}_A \hat{\Gamma}_A I$. This suggests that $\mathsf{RT}(A)/I$ lives over $\mathsf{Set}/\hat{\Gamma} A_I$ rather than Set. (Moreover, in the case of slicing, we cannot guarantee that (4.2) is the inclusion of ¬¬-sheaves if I is not an assembly.) This suggests that, in order to obtain closure under products and slices, we need to work over more general 'bases' than Set. In other words, we should consider PCAs that are constructed internally in base categories other than Set.

Let us examine the morphism (4.1) a bit more closely. The product, as categories, of two toposes is their pseudo*co*product in the 2-category of toposes, geometric morphisms and geometric transformations. The inclusion in (4.1) is simply the geometric inclusion $\mathsf{Set} + \mathsf{Set} \hookrightarrow \mathsf{RT}(A) + \mathsf{RT}(B)$ induced by the two inclusions $\mathsf{Set} \hookrightarrow \mathsf{RT}(A)$ and $\mathsf{Set} \hookrightarrow \mathsf{RT}(B)$. Viewed like this, it is not at all surprising that taking the pseudocoproduct of realizability toposes involves taking the pseudocoproduct of the 'base categories' as well. If we want to keep working over Set, then it would make more sense to consider the pseudopushout of the span:

$$\mathsf{RT}(A) \hookleftarrow \mathsf{Set} \hookrightarrow \mathsf{RT}(B)$$

which exists by [Joh77, Proposition 4.26]. We will cover this construction in Section 4.2 below.

Global sections

Next, we discuss the behavior of the global sections functors in relation to the morphisms in (4.1) and (4.2). This is not really an 'obstacle' in the sense specified at the beginning of this section, but for the case of slicing, it is nevertheless telling. We will see that, even if one is only interested in absolute PCAs, slicing will likely force one to consider relative PCAs as well.

In Section 3.2.2, we saw that a PCA A is absolute iff the inverse image part $\hat{\Gamma}_A$ of $\mathsf{Set} \hookrightarrow \mathsf{RT}(A)$ is the global sections functor. Of course, if our base category is not Set, then this criterion is not available. On the other hand, if A and B are absolute PCAs, then the inverse image part of (4.1) still *commutes* with the global sections functors:

$$\begin{array}{ccc} \mathsf{RT}(A) \times \mathsf{RT}(B) & \xrightarrow{\hat{\Gamma}_A \times \hat{\Gamma}_B} & \mathsf{Set}^2 \\ & {\scriptstyle \mathrm{Hom}(1,-)} \searrow \quad \swarrow {\scriptstyle \mathrm{Hom}(1,-)} & \\ & \mathsf{Set} & \end{array}$$

Thus, it seems reasonable to expect the following: if A and B are absolute PCAs, then $\mathsf{RT}(A) \times \mathsf{RT}(B)$ should be a realizability topos over an absolute PCA internal to Set^2. For slicing, we do not have the analogous result. That is, if A is an absolute PCA, and I is an object of $\mathsf{RT}(A)$, then it does not follow that the diagram

$$\begin{array}{ccc} \mathsf{RT}(A)/I & \xrightarrow{\hat{\Gamma}_A} & \mathsf{Set}/\hat{\Gamma}_A I \\ & {\scriptstyle \mathrm{Hom}(1,-)} \searrow \quad \swarrow {\scriptstyle \mathrm{Hom}(1,-)} & \\ & \mathsf{Set} & \end{array}$$

commutes.

Example 4.1.1. For an explicit counterexample, let A be an absolute PCA which is not semitrivial, and take $I = \hat{\nabla}_A 2$. In $\mathsf{RT}(A)$, the coproduct $1+1$ is the assembly X with $|X| = 2$, $E_X(0) = \downarrow\{\top\}$ and $E_X(1) = \downarrow\{\bot\}$. Now consider the object $\hat{\eta}_{1+1} \colon 1+1 \to \hat{\nabla}_A 2$ of $\mathsf{RT}(A)/\hat{\nabla}_A 2$. Then $\hat{\Gamma}_A \hat{\eta}_{1+1}$ is an isomorphism, so it is the terminal object of $\mathsf{Set}/2$, which means that it has exactly one global section. On the other hand, $\hat{\eta}_{1+1}$ does not have any global sections in $\mathsf{RT}(A)/\hat{\nabla}_A 2$. Indeed, such a global section is a section of $\hat{\eta}_{1+1}$, meaning that $\hat{\eta}_{1+1}$ must be split epi. But $\hat{\eta}_{1+1}$ is also mono (since $1+1$ is an assembly), so this implies that $\hat{\eta}_{1+1}$ is iso, i.e., that $1+1$ is a constant object. If A is not semitrivial, then this is clearly not the case.

Thus, even if we start with an absolute PCA A, then we should still expect $\mathsf{RT}(A)/I$ to arise from a *relative* PCA internal to $\mathsf{Set}/\hat{\Gamma}_A I$.

Projectives

In Section 3.2.3, we saw that the projective objects in $\mathsf{RT}(A)$ are precisely the partitioned assemblies. In particular, the terminal object of $\mathsf{RT}(A)$ is always projective. For slices of $\mathsf{RT}(A)$, this is not the case. Indeed, if I is an object of $\mathsf{RT}(A)$, then the forgetful functor $\mathsf{RT}(A)/I \to \mathsf{RT}(A)$ has a right adjoint $(-)\times I$, which is a regular functor. As we mentioned in Remark 3.2.17, this implies that $\mathsf{RT}(A)/I \to \mathsf{RT}(A)$ preserves projectives. In particular, if the terminal object of $\mathsf{RT}(A)/I$ is projective, then I must be projective as well. Contrapositively, if I is not projective, then the terminal object of $\mathsf{RT}(A)/I$ will not be projective either.

Thus, we must allow 'realizability-like' toposes in which the terminal object is not necessarily projective. In order to see how this can be done, consider the PCA TA introduced in Chapter 2: it is equipped with the filter $(TA)^\# = \uparrow(TA^\#) = \{\alpha \in TA \mid \alpha \cap A^\# \neq \emptyset\}$. As a result, realizers from $\alpha \in (TA)^\#$ are really realizers from $A^\#$ in disguise, a fact we have exploited frequently above. It also plays a crucial role in the proof that every partitioned assembly is projective; in order to split a regular epi $g \colon Y \twoheadrightarrow X$ with X a partitioned assembly, we use that there exists a *single* $r \in A^\#$ such that $r \cdot E_X(x) \subseteq \bigcup_{y \in |Y|} G(y,x)$ (where G represents g).

We can also consider PCAs of the form $B = (TA, \phi, \cdot, \subseteq)$, where ϕ is a filter which is not necessarily of the form $\uparrow(TF) = \{\alpha \in TA \mid \alpha \cap F \neq \emptyset\}$ for some filter F on the PAS $(A, \cdot, \leq)$. Now we can define a 'realizability-like' tripos $\mathsf{P}_{A,\phi}$ as follows. For a set X, the underlying set of $\mathsf{P}_{A,\phi}X$ is still $(DA)^X$, but now we say that $\psi \vdash_X \chi$ iff there exists a $\rho \in \phi$ such that $\rho \cdot \psi(x) \subseteq \chi(x)$ for all $x \in X$. It is easy to check that this indeed yields a tripos, and therefore we have a corresponding topos $\mathsf{RT}(A, \phi)$. Its category of assemblies $\mathsf{Asm}(A, \phi)$ (that is, the full subcategory of the subobjects of constant objects, see Remark 3.2.7) can be described as follows. The objects are simply assemblies over A, but a function $f\colon |X| \to |Y|$ is a morphism of assemblies iff there exists a $\rho \in \phi$ such that $\rho \cdot E_X(x) \subseteq E_Y(f(y))$ for all $x \in |X|$. In other words, every element of ρ should be a tracker of $f\colon X \to Y$ in the usual sense.

In this setting, if X is a partitioned assembly and $g\colon Y \twoheadrightarrow X$ in $\mathsf{RT}(A, \phi)$ is regular epi, then there exists a $\rho \in \phi$ such that $\rho \cdot E_X(x) \subseteq \bigcup_{y \in |Y|} G(y, x)$, rather than a single element r. Because of this, the proof of Proposition 3.2.16 no longer goes through. In fact, we can give an explicit example of such a 'realizability-like' topos $\mathsf{RT}(A, \phi)$ in which the terminal object is not projective.

Example 4.1.2. Let A be a PCA that is not absolute, and let $B = (TA)[X]$ (as in Example 2.2.15), where $X = A \backslash A^\# \in TA$. That is, B is the PAP $(TA, \cdot, \subseteq)$, equipped with the filter $\phi = \langle (TA)^\# \cup \{X\} \rangle$. As in Example 2.2.15, we see that $\alpha \in TA$ is in ϕ iff there exists a $\rho \in (TA)^\#$ such that $\rho \cdot X \subseteq \alpha$, that is, iff there exists an $r \in A^\#$ such that $r \cdot X \subseteq \alpha$.

Consider the assembly Y given by $|Y| = X$ and $E_Y(x) = \downarrow\{x\}$. Then the unique arrow $Y \to 1$ is a regular epi, since $\rho := \mathsf{k} \cdot X \in F$ satisfies $\rho \cdot E_1(*) \subseteq X = \bigcup_{x \in |Y|} E_Y(x)$. If $1 \in \mathsf{RT}(A, F)$ is projective, this implies that Y has a global section, i.e., there exists an $a \in X$ such that $1 \xrightarrow{a} Y$ is a morphism of assemblies. This implies in particular that $\downarrow\{a\} \in F$, so there exists an $r \in A^\#$ such that $rx \leq a$ for every $x \in X$.

Now let A be the relative version of Scott's graph model $(\mathcal{P}\omega, (\mathcal{P}\omega)^{\mathrm{re}}, \cdot, =)$. Then the above says that there exist an r.e. set r and a non-r.e. set a such that $rx = a$ for every non-r.e. set x. In particular, for every $m \in a$, there must exist an n such that $\langle n, m \rangle \in r$. Conversely, suppose that we have a coded pair $\langle n, m \rangle \in r$. If x is any non-r.e. set such that $e_n \subseteq x$, then we see that $m \in rx = a$. Thus, we can conclude that $a = \{m \mid \exists n(\langle n, m \rangle \in r)\}$. But since r is r.e., this implies that a is r.e. as well, contradiction.

This example depends crucially on the fact that the PCA A is relative. However, we will see that, if we combine the ideas here with base change, then we will also be able to 'break' the projectivity of 1 if we start with absolute PCAs.

Remark 4.1.3. The description of $\mathsf{Asm}(A, F)$ above Example 4.1.2 shows that $\mathsf{Asm}(A, F)$ is actually $\mathsf{PAsm}(B)$. In particular, $\mathsf{PAsm}(B)$ is a regular category, which means that B is a T-algebra; see [HvO03, Theorem 4.2]. We also see that, even though every object of $\mathsf{PAsm}(B)$ *becomes* projective via $\mathsf{PAsm}(B) \hookrightarrow$

$\mathsf{Asm}(B)$, this does not mean that every object of $\mathsf{PAsm}(B)$ itself is projective, even if $\mathsf{PAsm}(B)$ is a regular category.

4.2 Pushouts

In this section, we investigate the pseudopushout, over Set, of two toposes of the form $\mathsf{RT}(A)$. We will prove both a negative and a positive result, corresponding to the analoga of Question 4.0.1 resp. Question 4.0.2 for pseudopushouts. The negative result is that, barring trivial cases, such a pseudopushout is *never* a realizability topos (Theorem 4.2.4). We will see that this is a consequence of the nonexistence of pseudoproducts in OPCA_T. On the positive side, such a pseudopushout is a *subtopos* of a realizability topos. This leads to the positive result that certain subtoposes of realizability toposes, namely, those that include Set, are closed under pseudoposuhout over Set (Theorem 4.2.5). We will establish these results in Section 4.2.2, while Section 4.2.1 collects some necessary background on pseudopushouts of geometric inclusions, and dense subtopses.

4.2.1 Pushouts and dense subtoposes

First of all, let us describe pseudopushouts of geometric inclusions. The proof of the following proposition serves mainly to record the parts of the construction that will be important for our purposes.

Proposition 4.2.1 ([Joh77, Proposition 4.26]). *In the 2-category of toposes, geometric morphisms and geometric transformations, pseudopushouts of pairs of inclusions exist. Moreover, inclusions are stable under pseudopushouts over inclusions.*

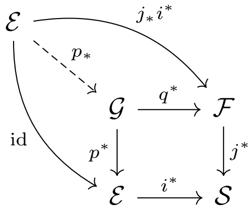
Sketch of proof. Suppose we have geometric inclusions $\mathcal{E} \overset{i}{\hookleftarrow} \mathcal{S} \overset{j}{\hookrightarrow} \mathcal{F}$. Its pseudopushout

$$\begin{array}{ccc} \mathcal{S} & \overset{j}{\hookrightarrow} & \mathcal{F} \\ {\scriptstyle i}\big\downarrow & & \big\downarrow{\scriptstyle q} \\ \mathcal{E} & \overset{p}{\hookrightarrow} & \mathcal{G} \end{array} \tag{4.3}$$

is constructed by constructing the pseudo*pullback* in CAT:

$$\begin{array}{ccc} \mathcal{G} & \overset{q^*}{\longrightarrow} & \mathcal{F} \\ {\scriptstyle p^*}\big\downarrow & & \big\downarrow{\scriptstyle j^*} \\ \mathcal{E} & \overset{i^*}{\longrightarrow} & \mathcal{S} \end{array}$$

That is, $\mathcal{G}$ is the iso-comma object of i^* and j^*, and p^* and q^* are the obvious projections. The direct image of p is the essentially unique arrow such that the diagram



commutes up to isomorphism. In particular, p is an inclusion, and similarly for q. Now suppose we have a commutative diagram

$$\begin{array}{ccc} \mathcal{S} & \overset{j}{\hookrightarrow} & \mathcal{F} \\ {\scriptstyle i}\downarrow & & \downarrow{\scriptstyle g} \\ \mathcal{E} & \overset{f}{\longrightarrow} & \mathcal{H} \end{array} \tag{4.4}$$

of toposes (or even in GEOM). Then by the definition of $\mathcal{G}$, there is a unique $h^*\colon \mathcal{H} \to \mathcal{G}$ such that $p^*h^* \cong f^*$ and $q^*h^* \cong g^*$, which is left exact. Its right adjoint h^* can be defined by requiring that there is a commutative diagram

$$\begin{array}{ccc} h^* & \longrightarrow & g_*q^* \\ \downarrow & & \downarrow \\ f_*p^* \longrightarrow f_*i_*i^*p^* & \cong & g_*j_*j^*q^* \end{array} \tag{4.5}$$

which is a pullback square at each coordinate. The final thing to show, which is of course the challenging part, is that $\mathcal{G}$ is actually a topos. This is a consequence of the glueing construction in topos theory; for details, we refer to [Joh77, Section 4.2]. □

In the remainder of this section, we prove two lemmata concerning pseudopushouts of geometric inclusions. These lemmata may be known, but we have not been able to find a reference for them. Therefore, we state and prove them for the sake of completeness.

Note that, as a consequence of the definition of p_* in the proof above, we have $q^*p_* \cong j_*i^*$, which is known as the Beck-Chevalley Condition for the square (4.3). In the same way, we also have $p^*q_* \cong i_*j^*$, even though, for general squares, these two BCCs are not equivalent.

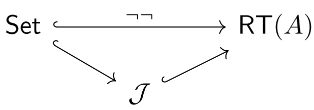
Lemma 4.2.2. *Suppose given a pseudopushout of toposes as in* (4.3) *and a commutative diagram as in* (4.4) *such that:*

- *f and g are both inclusions;*
- *the square* (4.4) *satisfies both BCCs: $g^*f_* \cong j_*i^*$ and $f^*g_* \cong i_*j^*$.*

Then the unique mediating arrow $h\colon \mathcal{G} \to \mathcal{H}$ is an inclusion.

Proof. The fact that $f^*g_* \cong i_*j^*$ factors through j^* implies that $f^*g_* \to f^*g_*j_*j^*$ is an isomorphism. Taking the image of the diagram (4.5) under f^* yields that $f^*h^* \to f^*f_*p^*$ is also an isomorphism (since the square is a pullback at each coordinate). This yields $p^*h^*h_* \cong f^*h_* \cong f^*f_*p^* \cong p^*$. Similarly, we find $q^*h^*h_* \cong q^*$, and these two isomorphisms combine into an isomorphism $h^*h_* \cong \mathrm{id}_{\mathcal{G}}$, as desired. ☐

As we have seen in Section 3.2.2, the subtopos of $\neg\neg$-sheaves in $\mathsf{RT}(A)$ is equivalent to Set. If A is absolute, then this is in fact the *only* inclusion $\mathsf{Set} \hookrightarrow \mathsf{RT}(A)$, as was shown by Johnstone in [Joh13, Corollary 1.4]. In the setting of relative PCAs, however, this is no longer true. Indeed, the noncommutative triangle in Example 3.4.15 above provides a counterexample. On the other hand, we know that $\neg\neg\colon \mathsf{Set} \hookrightarrow \mathsf{RT}(A)$ is the smallest dense subtopos of $\mathsf{RT}(A)$. That is, a subtopos $\mathcal{J} \hookrightarrow \mathsf{RT}(A)$ is dense iff there exists a factorization (which is automatically essentially unique):



In particular, any *dense* inclusion $\mathsf{Set} \hookrightarrow \mathsf{RT}(A)$ must be equivalent to the inclusion of $\neg\neg$-sheaves.

Lemma 4.2.3. *Dense geometric inclusions are stable under pseudopushouts over arbitrary inclusions. That is, in the diagram* (4.3)*, if i is dense, then q is also dense.*

Proof. Assume that i is dense and let $X \in \mathcal{G}$ be such that $q^*(X) \cong 0$. Then

$$i^*(p^*(X)) \cong j^*(q^*(X)) \cong j^*(0) \cong 0,$$

so by our assumption, $p^*(X) \cong 0$ as well. Therefore, we have $q^*(X) \cong 0 \cong q^*(0)$ and $p^*(X) \cong 0 \cong p^*(0)$, and these two isomorphisms combine into an isomorphism $X \cong 0$ in $\mathcal{G}$, as desired. ☐

In particular, if, in the diagram (4.3), i and j are both dense, then $pi \cong jq$ is also dense, since dense inclusions are closed under composition.

4.2.2 Pushouts of realizability toposes

First, we prove the promised negative result, stating that the pseudopushout, over Set, of two realizability toposes is almost never a realizability topos.

Theorem 4.2.4. *Let A_0 and A_1 be PCAs. If the pseudopushout of $\mathsf{Set} \hookrightarrow \mathsf{RT}(A_0)$ and $\mathsf{Set} \hookrightarrow \mathsf{RT}(A_1)$ is a realizability topos, then at least one of A_0 and A_1 is trivial.*

Proof. Suppose we have a pseudopushout diagram

$$\begin{array}{ccc} \mathsf{Set} & \xhookrightarrow{\neg\neg} & \mathsf{RT}(A_1) \\ {\scriptstyle \neg\neg}\big\downarrow & & \big\downarrow{\scriptstyle p_1} \\ \mathsf{RT}(A_0) & \xhookrightarrow{p_0} & \mathsf{RT}(C) \end{array}$$

of toposes. Then by Lemma 4.2.3, the composite $\mathsf{Set} \hookrightarrow \mathsf{RT}(C)$ is dense, and therefore it must be the inclusion of $\neg\neg$-sheaves. In particular, we have the pseudopullback

$$\begin{array}{ccc} \mathsf{RT}(C) & \xrightarrow{p_1^*} & \mathsf{RT}(A_1) \\ {\scriptstyle p_0^*}\big\downarrow & \searrow^{\hat{\Gamma}_C} & \big\downarrow{\scriptstyle \hat{\Gamma}_{A_1}} \\ \mathsf{RT}(A_0) & \xrightarrow[\hat{\Gamma}_{A_0}]{} & \mathsf{Set} \end{array}$$

of categories.

Since p_i is an inclusion, we have $p_i^*\hat{\nabla}_C \cong p_i^*(p_i)_*\hat{\nabla}_{A_i} \cong \hat{\nabla}_{A_i}$. Moreover, the resulting isomorphism $p_i^*\hat{\nabla}_C\hat{\Gamma}_C \cong \hat{\nabla}_{A_i}\hat{\Gamma}_{A_i}p_i^*$ identifies $p_i^*\hat{\eta}\colon p_i^* \to p_i^*\hat{\nabla}_C\hat{\Gamma}_C$ with $\hat{\eta}p_i^*\colon p_i^* \to \hat{\nabla}_{A_i}\hat{\Gamma}_{A_i}p_i^*$. Now, if $X \in \mathsf{RT}(C)$, we see that: X is an assembly iff $\hat{\eta}_X$ is mono; iff $p_i^*\hat{\eta}_X$ is mono for $i = 0, 1$; iff $\hat{\eta}_{p_i^*X}$ is mono for $i = 0, 1$; iff p_i^*X is an assembly for $i = 0, 1$. Thus, the pseudopullback above restricts to a pseudopullback of categories:

$$\begin{array}{ccc} \mathsf{Asm}(C) & \longrightarrow & \mathsf{Asm}(A_1) \\ \big\downarrow & \searrow^{\Gamma_C} & \big\downarrow{\scriptstyle \Gamma_{A_1}} \\ \mathsf{Asm}(A_0) & \xrightarrow[\Gamma_{A_0}]{} & \mathsf{Set} \end{array}$$

Since all the categories and functors in this diagram are regular, this is even a pseudopullback in REG, which means that Γ_C is the pseudoproduct of Γ_{A_0} and Γ_{A_1} in $\mathsf{REG}/\mathsf{Set}$.

To finish the proof, let B be an arbitrary PCA. By Corollary 3.3.14(iii) and the above, we have natural equivalences:

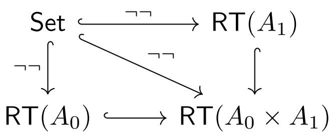
$$\begin{aligned} \mathsf{OPCA}_T(B, C) &\simeq (\mathsf{REG}/\mathsf{Set})(\Gamma_B, \Gamma_C) \\ &\simeq (\mathsf{REG}/\mathsf{Set})(\Gamma_B, \Gamma_{A_0}) \times (\mathsf{REG}/\mathsf{Set})(\Gamma_B, \Gamma_{A_1}) \\ &\simeq \mathsf{OPCA}_T(B, A_0) \times \mathsf{OPCA}_T(B, A_1). \end{aligned}$$

We conclude that C is a pseudoproduct of A_0 and A_1 in OPCA_T, so the result follows by Theorem 2.4.21. □

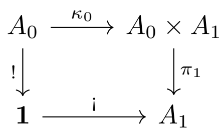
As we have seen in the previous section, for every dense subtopos $\mathcal{J} \hookrightarrow \mathsf{RT}(A)$, there is an essentially unique inclusion $\mathsf{Set} \hookrightarrow \mathcal{J}$ such that the composite $\mathsf{Set} \hookrightarrow \mathcal{J} \hookrightarrow \mathsf{RT}(A)$ is the inclusion of $\neg\neg$-sheaves. Thus, we can ask what the pseudopushout, over Set, of two such dense subtoposes looks like. The following theorem shows that it must also be a dense subtopos of a realizability topos.

Theorem 4.2.5. *Dense subtoposes of realizability toposes are closed under pseudopushout over* Set.

Proof. Let A_0 and A_1 be PCAs. First, we will show that the pseudopushout $\mathcal{E}$ of the span $\mathsf{RT}(A_0) \hookleftarrow \mathsf{Set} \hookrightarrow \mathsf{RT}(A_1)$ is a dense subtopos of a realizability topos. Recall from Section 2.4.1 that there is an arrow $\pi_i \dashv \kappa_i : A_i \to A_0 \times A_1$ in $\mathsf{OPCA}_{\mathrm{adj}}$, which satisfies $\pi_i \kappa_i \simeq \mathrm{id}_{A_i}$. Regarding π_i and κ_i as projective applicative morphisms yields a geometric inclusion $\mathsf{RT}(\pi_i) \dashv \mathsf{RT}(\kappa_i) : \mathsf{RT}(A_i) \hookrightarrow \mathsf{RT}(A_0 \times A_1)$. Thus, we get a square of inclusions:

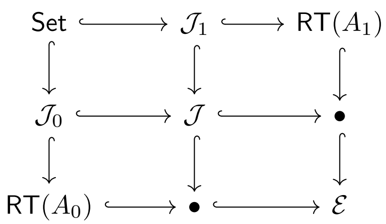


Moreover, we know that $\pi_1 \kappa_0$ is a zero morphism, i.e., the diagram



commutes. Taking the image of this diagram under RT yields $\mathsf{RT}(\pi_1) \circ \mathsf{RT}(\kappa_0) \cong \hat{\nabla}_{A_1} \hat{\Gamma}_{A_0}$ (see Example 3.3.9). Similarly, we find $\mathsf{RT}(\pi_0) \circ \mathsf{RT}(\kappa_1) \cong \hat{\nabla}_{A_0} \hat{\Gamma}_{A_1}$, so our square of geometric inclusions satisfies both BCCs. By Lemma 4.2.2, the mediating arrow $\mathcal{E} \to \mathsf{RT}(A_0 \times A_1)$ is an inclusion. Moreover, $\neg\neg : \mathsf{Set} \to \mathsf{RT}(A_0 \times A_1)$ factors through this inclusion, which shows that $\mathcal{E}$ is a dense subtopos of $\mathsf{RT}(A_0 \times A_1)$.

For the general case, suppose we have dense inclusions $\mathsf{Set} \hookrightarrow \mathcal{J}_i \hookrightarrow \mathsf{RT}(A_i)$ for $i = 0, 1$. Then we construct pseudopushout squares:



So we see that the pseudopushout $\mathcal{J}$ of $\mathcal{J}_0$ and $\mathcal{J}_1$ over Set is a dense subtopos of $\mathcal{E}$, which is a dense subtopos of $\mathsf{RT}(A_0 \times A_1)$. This completes the proof. □

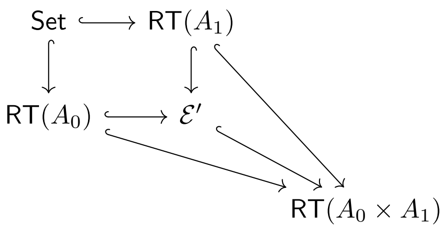
We can describe the situation in Theorem 4.2.5 a bit more explicitly. In Proposition 3.4.5, we saw that every geometric morphism of triposes $f : \mathsf{P} \to \mathsf{Q}$ yields a geometric morphism of toposes $\mathcal{C}[\mathsf{P}] \to \mathcal{C}[\mathsf{Q}]$, and that the latter is an inclusion iff the former is an inclusion, i.e., $f^* f_* \simeq \mathrm{id}_{\mathsf{P}}$. Now consider the tripos morphism $j := f_* f^* : \mathsf{Q} \to \mathsf{Q}$; it is left exact, and moreover, it satisfies

$\mathrm{id}_{\mathsf{Q}} \leq j$ and $jj \simeq j$. In other words, j satisfies analogous conditions to those of (Lawvere-Tierney) topologies on a topos, and accordingly, such morphisms j are called topologies on the tripos Q. It can be shown that *every* subtopos of $\mathcal{C}[\mathsf{Q}]$ arises in this way, and in particular, corresponds to a topology on Q; see [Pit81, Chapter 5].

For a realizability tripos P_A, topologies on P_A correspond to idempotent morphisms of PCAs $j\colon DA \to DA$ such that $\mathrm{id}_{DA} \leq j$. Now consider the pseudopushout $\mathcal{E}$ of the span $\mathsf{RT}(A_0) \hookleftarrow \mathsf{Set} \hookrightarrow \mathsf{RT}(A_1)$. As we have shown in Theorem 4.2.5, $\mathcal{E}$ is a subtopos of $\mathsf{RT}(A_0 \times A_1)$, so it must correspond to a topology on $\mathsf{P}_{A_0 \times A_1}$, i.e., a map $j\colon D(A_0 \times A_1) \to D(A_0 \times A_1)$. First of all, we note that $\mathsf{RT}(A_i) \hookrightarrow \mathsf{RT}(A_0 \times A_1)$ is given by $D(\kappa_i\pi_i)\colon D(A_0 \times A_1) \to D(A_0 \times A_1)$. Explicitly, we can describe $D(\kappa_0\pi_0)$ as:

$$D(\kappa_0\pi_0)(\alpha) = D\pi_0(\alpha) \times A_1 = \{(a_0, a_1) \mid \exists a_1' \in A_1\,((a_0, a_1') \in \alpha)\},$$

and similarly for $D(\kappa_1\pi_1)$. Since $\mathsf{RT}(A_i)$ is contained in $\mathcal{E}$, we must have $j \leq D(\kappa_i\pi_i)$ for $i = 0, 1$. Conversely, suppose that $j'\colon D(A_0 \times A_1) \to D(A_0 \times A_1)$ is a topology such that $j' \leq D(\kappa_i\pi_i)$ for $i = 0, 1$. If $\mathcal{E}' \hookrightarrow \mathsf{RT}(A_0 \times A_1)$ is the subtopos corresponding to j', then we get a commutative diagram:



where the square commutes since $\mathcal{E}' \hookrightarrow \mathsf{RT}(A_0 \times A_1)$ is an inclusion. But since $\mathcal{E}$ is the pseudopushout of the $\mathsf{RT}(A_i)$ over Set, this means that $\mathcal{E}$ is contained in $\mathcal{E}'$, that is, $j' \leq j$. We can conclude that j is the *largest* topology $D(A_0 \times A_1) \to D(A_0 \times A_1)$ such that $j \leq D(\kappa_i\pi_i)$ for $i = 0, 1$. But that is simply to say that j is the meet of $D(\kappa_0\pi_0)$ and $D(\kappa_1\pi_1)$, which we can describe, up to isomorphism, by:

$$j(\alpha) = D\pi_0(\alpha) \times D\pi_1(\alpha).$$

Geometrically, we can think of $j(\alpha)$ as the smallest 'rectangle' that contains α. Note also that, if $h\colon TA_0 \times TA_1 \to T(A_0 \times A_1)$ is the map of $\mathsf{OPCA}_{\mathrm{adj}}$ defined in the proof of Proposition 2.4.20, then $j(\alpha) = h_* h^*(\alpha)$ for $\alpha \in T(A_0 \times A_1)$, and $j(\emptyset) = \emptyset$.

For the general case, suppose we have dense subtoposes $\mathcal{J}_i \hookrightarrow \mathsf{RT}(A_i)$ corresponding to topologies $j_i\colon DA_i \to DA_i$. Then a similar argument shows that the pseudopushout $\mathcal{J}$ of $\mathcal{J}_0$ and $\mathcal{J}_1$ over Set corresponds to the topology $j\colon D(A_0 \times A_1) \to D(A_0 \times A_1)$ given by:

$$j(\alpha) = j_0(D\pi_0(\alpha)) \times j_1(D\pi_1(\alpha)).$$

4.3 PCAs internal to a regular category

In this section, we treat a framework of generalized PCAs suitable for handling products and slices of categories of assemblies. As we mentioned at the beginning of this chapter, this framework considers PCAs constructed over general base categories, and allows a very liberal notion of relative PCA, which can be motivated by Example 4.1.2.

Before we proceed, we introduce some notation concerning the internal logic of regular categories. If $\mathcal{C}$ is a regular category, then $\mathrm{Sub}_{\mathcal{C}} \colon \mathcal{C} \to \mathsf{PreOrd}^{\mathrm{op}}$ soundly interprets typed regular logic, as discussed in Section 3.1.2. Thus, if φ is a formula of regular logic in context $\Gamma = x_0 : X_0, \ldots, x_{n-1} : X_{n-1}$ (with the X_i objects of $\mathcal{C}$), then φ receives an interpretation $[\![\varphi]\!] \in \mathrm{Sub}_{\mathcal{C}}(X_0 \times \cdots \times X_{n-1})$. Since we are working with subobjects, we will write this more suggestively as:

$$\{(x_0, \ldots, x_{n-1}) \in X_0 \times \cdots \times X_{n-1} \mid \varphi\} \subseteq X_0 \times \cdots \times X_{n-1}.$$

In particular, we will use the usual subset sign to denote subobjects in $\mathcal{C}$. We will often write $U(x) :\equiv \varphi(x)$ to indicate that we define the subobject $U \subseteq X$ as $\{x \in X \mid \varphi(x)\}$, and similarly for contexts with multiple variables.

Of course, if φ and ψ are regular formulas in context Γ, then we say that the sequent $\varphi \vdash_\Gamma \psi$ is **valid** in $\mathcal{C}$ if:

$$\{(x_0, \ldots, x_{n-1}) \in X_0 \times \cdots \times X_{n-1} \mid \varphi\} \subseteq \{(x_0, \ldots, x_{n-1}) \in X_0 \times \cdots \times X_{n-1} \mid \psi\}.$$

In this case, we write $\mathcal{C} \colon \varphi \vDash_\Gamma \psi$, or $\varphi \vDash_\Gamma \psi$ if $\mathcal{C}$ is understood. This interpretation is sound in the following sense: if the sequent s is derivable (in some proof system for typed regular logic) from a set of sequents S, and all members of S are valid in $\mathcal{C}$, then s will be valid in $\mathcal{C}$ as well. We will often signal an application of soundness by writing 'reason inside $\mathcal{C}$', or 'work internally in $\mathcal{C}$'.

4.3.1 Internal PCAs

Throughout this section, $\mathcal{C}$ denotes a regular category which serves as our 'base category'.

Definition 4.3.1 (Cf. [Ste13, Definition 1.2.1]). *A **partial applicative poset internal to** $\mathcal{C}$, or IPAP over $\mathcal{C}$, is an object A of $\mathcal{C}$ equipped with:*

- *a partial order $\leq \subseteq A \times A$;*
- *a partial binary **application map** $A \times A \rightharpoonup A$, that is, a subobject $D \subseteq A \times A$ and an arrow $D \to A \colon (a, b) \mapsto a \cdot b$,*

satisfying:

(iA) $a' \leq a \wedge b' \leq b \wedge D(a, b) \vDash_{a,a',b,b':A} D(a', b') \wedge a' \cdot b' \leq a \cdot b.$

*We say that A is **total** if $D = A \times A$, and **discrete** if $\leq$ is the discrete order (i.e., the diagonal $\langle \mathrm{id}, \mathrm{id}\rangle \colon A \hookrightarrow A \times A$).*

As for ordinary PAPs, we will omit the dot for application whenever possible, and adopt the convention that application associates to the left. Moreover, in the sequel we will write $ab\downarrow$ for the formula $D(a, b)$. In this way, we see that axiom (iA) above expresses axiom (A) from Definition 2.1.1 'internally in $\mathcal{C}$'.

If $t(\vec{x})$ is a pure term in n variables, then we get a partial map $\lambda\vec{x}.t\colon A^n \rightharpoonup A$ in the obvious way. The domain of $\lambda\vec{x}.t$ can be expressed by a regular formula involving D and the application map. We abbreviate this formula by $t(\vec{a})\downarrow$, where $\vec{a}\colon A^n$. For example, $abc\downarrow$ may be expressed as $D(a, b) \wedge D(ab, c)$.

Remark 4.3.2. The reader may object to this, and to the formulation of axiom (iA) above, that the function symbol for application is a unary function symbol with domain D, rather than a binary function symbol taking inputs from A. We can circumvent this difficulty by saying that officially, the application map is a tertiary single-valued relation symbol on A expressing '$ab = c$'. In this setup, we can define D as $\{(a, b) \mid \exists c(ab = c)\}$. The formula $abc\downarrow$ is then actually an abbreviation of $\exists w : A(ab = w \wedge D(w, c))$. Likewise, if $t(\vec{a})\downarrow$, then we can still freely use the expression $t(\vec{a})$ in our formulas. E.g., we may write $abc\downarrow \wedge\, \varphi(abc)$, which should really be read as $\exists v, w : A(ab = w \wedge wc = v \wedge \varphi(v))$. Other solutions may also be employed as well; in particular, it is possible to treat formulas involving the application function without using existential quantification at all, therefore staying within the realm of *cartesian* logic. However, we will need existential quantification in the sequel of this chapter anyway (e.g., in Definition 4.3.3(ii) and Definition 4.3.12 below), so the approach presented here is satisfactory.

Next, we need to define internal partial applicative structures and combinatory completeness. At first glance, it would seem reasonable to equip an IPAP A with a subobject $A^\#$ satisfying internal versions of axioms (B) and (C) from Definition 2.1.15. However, with Example 4.1.2 in mind, we really want to be talking about filters on the PCA TA of nonempty *downsets* of A. We cannot do this inside $\mathcal{C}$ if $\mathcal{C}$ is merely a regular category. Of course, we can evade this issue by simply assuming that $\mathcal{C}$ is a topos. There is an alternative solution, however, which only requires $\mathcal{C}$ to be regular. This solution is taken from [Ste13]; the key idea is to view TA as an object *external* to $\mathcal{C}$.

Definition 4.3.3 (Cf. [Ste13, Definitions 1.3.11 and 1.3.14])**.** *Let A be an IPAP over $\mathcal{C}$.*

(i) We write $\mathcal{T}A$ for the set of all inhabited downsets of A. That is, $\mathcal{T}A$ consists of all subobjects $U \subseteq A$ such that:

- $\vDash \exists a : A(U(a))$ *(that is, $U \to 1$ is regular epi);*
- $a' \le a \wedge U(a) \vDash_{a,a':A} U(a')$.

(ii) We make $\mathcal{T}A$ into a PAP $(\mathcal{T}A, \cdot, \subseteq)$ as follows. For $U, V \in \mathcal{T}A$, we say that $UV\downarrow$ iff $U \times V \subseteq D$, and if this is the case, we set:

$$UV = \{a \in A \mid \exists b, c : A(U(b) \wedge V(c) \wedge a \le bc)\}.$$

*(iii) An **external filter** on A is a filter on $\mathcal{T}A$ in the sense of Definition 2.1.8.*

*(iv) A **partial applicative structure internal to** $\mathcal{C}$, or IPAS over $\mathcal{C}$, is pair (A, ϕ), where A is an IPAP over $\mathcal{C}$ and ϕ is an external filter on A. We say that (A, ϕ) is **absolute** if $\phi = \mathcal{T}A$.*

Even though we qualify the entirety of (A, ϕ) as 'internal to $\mathcal{C}$', we emphasize that really only A is internal. Indeed, $\mathcal{T}A$ is an *external* PAP, that is, simply a PAP in the sense of Definition 2.1.1. In particular, ϕ is a *set*, namely, a subset of $\mathcal{T}A$, and not something living inside $\mathcal{C}$.

Finally, we need to say what it means for (A, ϕ) to be an internal PCA. Of course, our realizers will be elements of ϕ, so in particular, they will be external objects, but they will need to realize terms *internally* in $\mathcal{C}$.

Definition 4.3.4. *Let (A, ϕ) be an IPAS over $\mathcal{C}$.*

*(i) Suppose that $t(\vec{x}, y)$ is a pure term. We say that $U \in \mathcal{T}A$ **realizes** $\lambda^*\vec{x}, y.t$ if:*

- $U(r) \vDash_{r,\vec{a}:A} r\vec{a}\downarrow$ *(where $\vec{a}$ has the same length as $\vec{x}$);*
- $U(r) \wedge t(\vec{a}, b)\downarrow \vDash_{r,\vec{a},b:A} r\vec{a}b\downarrow \wedge\, r\vec{a}b \leq t(\vec{a}, b)$.

*(ii) If ϕ is an external filter on A, then we say that the IPAS (A, ϕ) is a **partial combinatory algebra internal to** $\mathcal{C}$, or IPCA over $\mathcal{C}$, if there exist $\mathsf{K}, \mathsf{S} \in \phi$ realizing $\lambda^*xy.x$ and $\lambda^*xyz.xz(yz)$, respectively.*

Let us make a few remarks concerning Definition 4.3.4.

Remark 4.3.5. (i) We warn the reader that in the current setting, the isolated expression $\lambda^*\vec{x}, y.t$ does not mean anything (but see also item (ii) below). This is in contrast with the situation for ordinary PCAs, where $\lambda^*\vec{x}, y.t$ denotes an element of $A^{\#}$. In the setting of IPCAs, we can only use $\lambda^*\vec{x}, y.t$ in the expression 'realizer of $\lambda^*\vec{x}, y.t$'.

(ii) There are two important differences between [Ste13] and our setup. First of all, we require the external filter ϕ to consist of *inhabited* subobjects of A. In [Ste13], there is no such requirement (but see also Remark 4.3.32 below). Second, in [Ste13], the base categories are *Heyting* categories, which means they interpret full (typed) first-order logic. This allows a slightly more elegant treatment of the definition of an IPCA. For a pure term $t(\vec{x}, y)$, we can then *define*:

$$\lambda^*\vec{x}, y.t = \{r \in A \mid \forall \vec{a}, b\colon A(r\vec{a}\downarrow \wedge (t(\vec{a}, b)\downarrow \to r\vec{a}b\downarrow \wedge\, r\vec{a}b \leq t(\vec{a}, b)))\}.$$

Then ϕ contains a realizer of $\lambda^*\vec{x}, y.t$ in the sense of Definition 4.3.4(i) iff it contains the *element* $\lambda^*\vec{x}, y.t$ defined above; see also [Ste13, Definition 1.2.4]. If $\mathcal{C}$ is merely regular, however, then we need to use the notion 'realizer of $\lambda^*\vec{x}, y.t$', because we cannot define the *object* $\lambda^*\vec{x}, y.t$ inside $\mathcal{C}$ as we did above.

(iii) If (A, ϕ) is an IPCA, then in particular, $(\mathcal{T}A, \phi, \cdot, \subseteq)$ is a PCA, as is witnessed by $\mathsf{K}, \mathsf{S} \in \phi$. We will denote this PCA simply by $\mathcal{T}(A, \phi)$.

We have the following analogue of Proposition 2.1.24. The construction is similar, so we omit the proof.

Proposition 4.3.6. *If (A, ϕ) is an IPCA over $\mathcal{C}$, then $\lambda^* \vec{x}, y.t$ has a realizer in ϕ, for every pure term $t(\vec{x}, y)$.*

The reader may have noticed that, thus far, we have only talked about pure terms, and not about terms with parameters. In fact, since A is not a set but an object of $\mathcal{C}$, it is not entirely clear what a term with parameters from A should be. On the other hand, $\mathcal{T}A$ *is* a set, and it will be useful to have a notion of 'realizer of $\lambda^* \vec{x}, y.t$', where t contains parameters from $\mathcal{T}A$. In order to achieve this, we note that a term with parameters from $\mathcal{T}A$ can be written as $t(\vec{U}, \vec{x}, y)$, where $t(\vec{z}, \vec{x}, y)$ is a pure term and $\vec{U} \in \mathcal{T}A$.

Definition 4.3.7. *Let A be an IPAP over $\mathcal{C}$, let $t(\vec{z}, \vec{x}, y)$ be a pure term and let $\vec{U} \in \mathcal{T}A$ (of the same length as $\vec{z}$). We say that $V \in \mathcal{T}A$ **realizes** $\lambda^* \vec{x}, y.t(\vec{U}, \vec{x}, y)$ if there exists a realizer W of $\lambda^* \vec{z}\vec{x}y.t$ such that $V \subseteq W\vec{U}$.*

Clearly, Proposition 4.3.6 above can be extended to terms with parameters from ϕ, since ϕ is closed under defined application.

Remark 4.3.8. Definition 4.3.7 will occasionally create an ambiguity. For example, if we say that V realizes $\lambda^* x.UU$, then this could mean two things. We could mean that $V \subseteq WU$, where W realizes $\lambda^* yx.yy$, or we could mean that $V \subseteq WUU$, where W realizes $\lambda^* yzx.yz$. Therefore, we adopt the following convention: if we use the parameter U more than once in a term, then we assume we have substituted all these occurrences for the same variable, that is, we go with the first option. This is only for the sake of definiteness; in practice it does not matter which option one uses, except for the fact that the first option introduces fewer variables.

In addition to K and S, we also introduce a few other useful combinators, in the same way as in Section 2.1.4. We have the identity combinator I realizing $\lambda x.x$, a combinator $\overline{\mathsf{K}}$ realizing $\lambda xy.y$, and pairing and unpairing combinators P, P_0 and P_1 realizing $\lambda xyz.zxy$, $\lambda x.x\mathsf{K}$ and $\lambda x.x\overline{\mathsf{K}}$ respectively. For any choice of these pairing and unpairing combinators, we have:

$$\mathsf{P}(p) \wedge \mathsf{P}_0(p_0) \vDash_{p,p_0,a,b:A} p_0(pab)\downarrow \wedge\ p_0(pab) \leq a,$$
$$\mathsf{P}(p) \wedge \mathsf{P}_1(p_1) \vDash_{p,p_1,a,b:A} p_1(pab)\downarrow \wedge\ p_1(pab) \leq b.$$

We prove the first of these in detail, since it elucidates how Definition 4.3.7 works. By definition, there exists a realizer U of $\lambda^* yx.xy$ such that $\mathsf{P}_0 \subseteq U\mathsf{K}$. Now reason internally in $\mathcal{C}$ and suppose that $\mathsf{P}(p)$ and $\mathsf{P}_0(p_0)$. Then there exist $r, k : A$ such that $U(r)$, $\mathsf{K}(k)$ and $p_0 \leq rk$. We know that kab is defined and $\mathsf{k}ab \leq a$. This implies that $pabk$ is defined as well, and $pabk \leq kab \leq a$. This, in turn, implies that $rk(pab)$ is defined and $rk(pab) \leq pabk \leq a$. Finally, this implies that $p_0(pab)$ is defined and $p_0(pab) \leq rk(pab) \leq a$, as desired.

Because $\mathcal{T}A$ is simply a PAP in the sense of Chapter 2, the notion of a generated filter is available to us. That is, if $G \subseteq \mathcal{T}A$, then there is a least external filter $\langle G \rangle$ on A extending G. We treat two definitions involving the notion of a generated filter, both of which will be needed in the sequel of the chapter, espacially when we consider slicing in Section 4.4.2.

Definition 4.3.9 (Cf. [Ste13, Definition 2.4.1])**.** *Let A be an IPAP over $\mathcal{C}$.*

(i) We define a function $\delta_A \colon \mathrm{Hom}(1, A) \to \mathcal{T}A$ as follows. If $a \colon 1 \to A$ is a global section of A, then $\delta_A(a) := \{b \in A \mid b \leq a\} \in \mathcal{T}A$.

*(ii) If ϕ is an external filter on A, then we say that ϕ is **generated by singletons** if, for every $U \in \phi$, there exists an $a \colon 1 \to A$ such that $\vDash U(a)$ and $\delta_A(a) \in \phi$.*

Intuitively, if ϕ is generated by singletons, then everything that is realized by some member of ϕ is also realized by a 'principal downset' in ϕ. The following proposition explains the name 'generated by singletons'.

Proposition 4.3.10. *Let (A, ϕ) be an IPCA over $\mathcal{C}$. Then ϕ is generated by singletons iff there exists a set $C \subseteq \mathrm{Hom}(1, A)$ such that $\phi = \langle \{\delta_A(a) \mid a \in C\} \rangle$.*

Proof. If ϕ is generated by singletons, then $\phi = \uparrow\{\delta_A(a) \mid a \in C\}$, where $C = \{a \colon 1 \to A \mid \delta_A(a) \in \phi\}$, so in particular, we also have $\phi = \langle \{\delta_A(a) \mid a \in C\} \rangle$.

Conversely, suppose that $\phi = \langle \{\delta_A(a) \mid a \in C\} \rangle$ for some $C \subseteq \mathrm{Hom}(1, A)$. Note the IPAP structure on A makes $\mathrm{Hom}(1, A)$ into a PAP, and the function $\delta_A \colon \mathrm{Hom}(1, A) \to \mathcal{T}A$ preserves application and the order on the nose. Now define the generated filter $\langle C \rangle \subseteq \mathrm{Hom}(1, A)$. Then an easy argument shows that:

$$\uparrow\{\delta_A(a) \mid a \in \langle C \rangle\} = \langle \{\delta_A(a) \mid a \in C\} \rangle = \phi,$$

so in particular, ϕ is generated by singletons. □

An IPAP over Set is, of course, simply a PAP A, and we have $\mathcal{T}A = TA$. Thus, every PCA in the sense of Chapter 2 can be regarded as an IPCA $(A, (TA)^{\#})$ over Set. With Definition 4.3.9, we can even say a bit more: PCAs are exactly IPCAs over Set whose filters are generated by singletons. It is certainly possible to have IPCAs over Set whose filters are *not* generated by singletons. In fact, Example 4.1.2 is an example of such an IPCA, as we shall see shortly. The following definition generalizes the construction used in Example 4.1.2.

Definition 4.3.11. *Let (A, ϕ) be an IPCA over $\mathcal{C}$ and let $X \in \mathcal{T}A$. We define the IPCA $(A, \phi)[X]$ as $(A, \langle \phi \cup \{X\} \rangle)$, and we say that the filter $\langle \phi \cup \{X\} \rangle$ is **finitely generated**.*

4.3.2 Applicative morphisms

In this section, we will consider applicative morphisms between IPCAs. First, we will define a notion of applicative morphism between IPCAs over a given base; this definition also occurs in [Ste13]. Next, we introduce a notion of base change, thus obtaining a bicategory of IPCAs over arbitrary regular base categories. This approach allows us to compare IPCAs constructed over different bases, which will be useful when discussing products and slicing in the next section.

For ordinary PCAs, an applicative morphism $A \multimap B$ is a function $A \to TB$. Such a definition is not available here, because the 'PCA of inhabited downsets' $\mathcal{T}B$ lives outside $\mathcal{C}$. But note that a function $A \to TB$ can also be viewed as a *relation* between the sets A and B, with certain properties. In fact, when applicative morphisms were first defined in [Lon94, Definition 2.1.1], they were defined as relations, and not as functions. A regular category $\mathcal{C}$ has a well-behaved calculus of relations, so this approach seems promising.

For the time being, fix a regular base category $\mathcal{C}$. We first define a notion of applicative morphism between IPCAs over $\mathcal{C}$.

Definition 4.3.12 (Cf. [Ste13, Definition 2.3.20])**.** *Let (A, ϕ) and (B, ψ) be IPCAs over $\mathcal{C}$. An* ***applicative morphism*** *$f\colon (A, \phi) \to (B, \psi)$ is a subobject $f \subseteq A \times B$ with the following properties:*

(i) $\vDash_{a:A} \exists b\colon B\, (f(a, b))$;

(ii) $f(a, b) \wedge b' \leq b \vDash_{a:A;b,b':B} f(a, b')$;

(iii) if $V \in \phi$, *then* $f(V) := \{b \in B \mid \exists a\colon A(V(a) \wedge f(a, b))\} \in \psi$;

(iv) there is a $T \in \psi$ *such that*

$$T(t) \wedge f(a, b) \wedge f(a', b') \wedge aa'{\downarrow} \vDash_{a,a':A;t,b,b':B} tbb'{\downarrow} \wedge\ f(aa', tbb');$$

(v) there is a $U \in \psi$ *such that*

$$U(u) \wedge f(a, b) \wedge a \leq a' \vDash_{a,a':A;u,b:B} ub{\downarrow} \wedge\ f(a', ub).$$

Item (i) says that f is a total relation, i.e., that '$f(a)$ is inhabited', for each a (but of course, we cannot talk about $f(a)$ directly inside $\mathcal{C}$). This requirement is absent in [Ste13, Definition 2.3.20], which can therefore we viewed as a version of what we have called *partial* applicative morphisms. For the purposes of this chapter, the notion of applicative morphism will suffice, which is why we include (i). Item (ii) says that f is 'downwards closed on the right', and amounts to requiring that '$f(a)$ is a downset'. Items (iii)-(v) correspond to the requirements for applicative morphisms between ordinary PCAs. Of course, we will also say that $T, U \in \psi$ realize $f\colon (A, \phi) \to (B, \psi)$, or more specifically, that f preserves application up to T and preserves the order up to U.

Remark 4.3.13. (i) Note that $V \mapsto f(V)$ as defined in item (iii) makes f into a morphism of PCAs $\mathcal{T}(A, \phi) \to \mathcal{T}(B, \psi)$.

(ii) Since applicative morphisms are the only morphisms we will consider between IPCAs, we use the ordinary arrow symbol to denote $f\colon (A,\phi) \to (B,\psi)$, rather than $\multimap$.

Definition 4.3.14. *Let (A,ϕ) and (B,ψ) be IPCAs over $\mathcal{C}$. If $f, f' \subseteq A \times B$, then we say that $f \leq f'$ if there exists a $V \in \psi$ such that*

$$V(s) \wedge f(a,b) \vDash_{a:A;s,b:B} sb\downarrow \wedge f'(a,sb).$$

*Such an element V is said to **realize** the inequality $f \leq f'$, and we write $f \simeq f'$ if both inequalities $f \leq f'$ and $f' \leq f$ hold.*

Proposition 4.3.15. *IPCAs over $\mathcal{C}$, applicative morphisms, and inequalities between applicative morphisms form a preorder-enriched bicategory, which we denote by $\mathsf{IPCA}_{\mathcal{C}}$.*

Proof. The identity on (A,ϕ) is given by $\{(a,a') \in A \times A \mid a \geq a'\}$. If $(A,\phi) \overset{f}{\to} (B,\psi) \overset{g}{\to} (C,\chi)$ are applicative morphisms, then gf is their composition as relations, i.e., $gf = \{(a,c) \in A \times C \mid \exists b\colon B\,(f(a,b) \wedge g(b,c))\}$. Let us verify in detail that gf preserves application up to a realizer, and leave the remaining properties of gf to the reader. Let f and g preserve application up to $T \in \phi$ resp. $T' \in \chi$. Now let $T'' \in \chi$ be a realizer of $\lambda^* xy.T'(T' \cdot g(T) \cdot x)y$. That is, we suppose that $T'' \subseteq WT' \cdot g(T)$, where W realizes $\lambda^* zz'xy.z(zz'x)y$. Now reason inside $\mathcal{C}$, and suppose that $T''(t'')$, $gf(a,c)$, $gf(a',c')$ and $aa\downarrow$. Then there exist $r, t', t_0 \colon C$ such that $W(r)$, $T'(t')$, $g(T)(t_0)$ and $t'' \leq rt't_0$. Moreover, there are $b, b'\colon B$ such that $f(a,b)$, $g(b,c)$, $f(a',b')$ and $g(b',c')$, and finally, there is also a $t : B$ such that $T(t)$ and $g(t,t_0)$. From these data, we can conclude that $tbb'\downarrow$ and $f(aa',tbb')$. In particular, $tb\downarrow$, so $t't_0c\downarrow$ as well, and $g(tb, t't_0c)$. This yields that $t'(t't_0c)c'\downarrow$ and $g(tbb', t'(t't_0c)c')$. Thus, $rt't_0cc'\downarrow$ as well, and $rt't_0cc' \leq t'(t't_0c)c'$, so $g(tbb', rt't_0cc')$. This, in turn, gives that $t''cc'\downarrow$, and $t''cc' \leq rt't_0cc'$, so $g(tbb', t''cc')$. Combining this with $f(aa', tbb')$ finally yields $gf(aa', t''cc')$, as desired.

That the axioms for a 1-category hold up to isomorphism, follows by reasoning internally in $\mathcal{C}$. Suppose we have relations $f, f', f''\colon (A,\phi) \to (B,\psi)$. Then $\mathsf{I} \in \psi$ realizes $f \leq f$, and if $V, V' \in \psi$ realize $f \leq f'$ resp. $f' \leq f''$, then a realizer of $\lambda^* x.V'(Vx)$ also realizes $f \leq f''$.

Now suppose we have relations $f, f' \subseteq A \times B$ such that $V \in \psi$ realizes $f \leq f'$, and an applicative morphism $g\colon (B,\psi) \to (C,\chi)$ that preserves application up to $T \in \chi$. Then any realizer of $\lambda^* x.T \cdot g(V) \cdot x$ also realizes $gf \leq gf'$. Finally, if $f \subseteq A \times B$ and $g, g' \subseteq B \times C$ are relations, then any realizer of $g \leq g'$ also realizes $gf \leq g'f$. ☐

Remark 4.3.16. In the construction of T'' above, the parameter T' occurred twice, so we used the convention from Remark 4.3.8. Had we chosen the other convention, then two elements from T' would have played a role in the internal argument in $\mathcal{C}$, rather than one. This does not make an essential difference to the argument, but it introduces even more variables.

In the internal reasoning that T'' works as desired, we really do the following: first, we unpack all the existential quantifiers, and then, we give the 'usual' argument that composition in OPCA_T is well-defined, but internally inside $\mathcal{C}$. Moreover, this internal argument has to be formulated somewhat cumbersomely because we only have regular logic at our disposal. In the sequel, we will usually only specify the term to be realized, and omit the internal argument that the corresponding realizer works as desired.

Again, we needed to say that $\mathsf{IPCA}_\mathcal{C}$ is a *bi*category since $f \circ \mathrm{id} = f$ iff f preserves the order on the nose, i.e., $f(a,b) \wedge a \leq a' \vDash_{a,a':A;b:B} f(a',b)$. As in Lemma 2.3.5 we see that $f \circ \mathrm{id} = \{(a,b) \in A \times B \mid \exists a' : A(a' \leq a \wedge f(a',b))\}$ is isomorphic to f and preserves the order on the nose.

Example 4.3.17. Suppose that A is an IPAP over $\mathcal{C}$ and that $\phi, \psi \subseteq \mathcal{T}A$ are two combinatorially complete external external filters on A such that $\phi \subseteq \psi$. Then we have an applicative morphism $f\colon (A,\phi) \to (A,\psi)$ given by $\{(a,a') \in A \times A \mid a \geq a'\}$. In the special case where ψ is a finite extension of ϕ, say $\psi = \langle \phi \cup \{X\}\rangle$ with $X \in \mathcal{T}A$, we will denote this f by ι_X, in analogy with Example 2.2.15.

Next, we will define, for each regular functor $p\colon \mathcal{C} \to \mathcal{D}$ between regular categories, a 2-functor $p^*\colon \mathsf{IPCA}_\mathcal{C} \to \mathsf{IPCA}_\mathcal{D}$. Before we can do so, we need a few auxilliary results. The first of these generalizes Lemma 2.2.5.

Lemma 4.3.18. *Let (A,ϕ) and (B,ψ) be IPCAs over $\mathcal{C}$, and suppose that $\phi = \langle G\rangle$ for a certain $G \subseteq \mathcal{T}A$. Let $f \subseteq A \times B$ be a relation satisfying (i), (ii), (iv) and (v) from Definition 4.3.12, and also:*

(iii)' if $U \in G$, then $f(U) \in \phi$.

Then f is an applicative morphism $(A,\phi) \to (B,\psi)$.

Proof. Apply Lemma 2.2.5 to the PCAs $\mathcal{T}(A,\phi)$ and $\mathcal{T}(B,\psi)$, and the function $f\colon \mathcal{T}A \to \mathcal{T}B$. □

If $p\colon \mathcal{C} \to \mathcal{D}$ is a regular functor and A is an IPAP over $\mathcal{C}$, then $p(A)$ is an IPAP over $\mathcal{D}$ in the obvious way. Note also that p yields an order-preserving function $\mathcal{T}A \to \mathcal{T}(p(A))$, that we also denote by p. If $\vec{U} = U_0, \ldots, U_{n-1}$ is a sequence of elements of $\mathcal{T}A$, then we write $p(\vec{U})$ as a shorthand for the sequence $p(U_0), \ldots, p(U_{n-1}) \in \mathcal{T}(p(A))$.

Lemma 4.3.19. *Let $p\colon \mathcal{C} \to \mathcal{D}$ be a regular functor, let A be an IPAP over $\mathcal{C}$.*

- *(i) If $t(\vec{x})$ is a pure term and $\vec{U} \in \mathcal{T}A$ are such that $t(\vec{U})\downarrow$, then $t(p(\vec{U}))$ is defined as well, and equal to $p(t(\vec{U}))$.*
- *(ii) If $G \subseteq \mathcal{T}A$, then $\langle p(\langle G\rangle)\rangle = \langle p(G)\rangle$.*

Proof. (i) is an easy induction on t.

For (ii), we first note that $p(G) \subseteq p(\langle G\rangle) \subseteq \langle p(\langle G\rangle)\rangle$, which implies $\langle p(G)\rangle \subseteq \langle p(\langle G\rangle)\rangle$. For the converse, suppose that we have an element $V \in p(\langle G\rangle)$, that is, $V = p(V')$ for some $V \in \langle G\rangle$. By Lemma 2.1.14, there are a pure term $t(\vec{x})$ and $\vec{U} \in G$ such that $t(\vec{U})\downarrow$ and $t(\vec{U}) \subseteq V'$. By (i), $t(p(\vec{U}))$ is also defined, and we have $t(p(\vec{U})) = p(t(\vec{U})) \subseteq p(V') = V$, which shows that $V \in \langle p(G)\rangle$. We conclude that $p(\langle G\rangle) \subseteq \langle p(G)\rangle$, hence also $\langle p(\langle G\rangle)\rangle \subseteq \langle p(G)\rangle$. □

Remark 4.3.20. The converse of Lemma 4.3.19(i) does *not* hold. Indeed, for $U, V \in \mathcal{T}A$, it may very well be the case that $p(U)\cdot p(V)\downarrow$, but UV is undefined. For an example of this phenomenon, consider the case where $\mathcal{D}$ is the terminal category.

Construction 4.3.21. Let $p\colon \mathcal{C} \to \mathcal{D}$ be a regular functor. We define a 2-functor $p^*\colon \mathsf{IPCA}_\mathcal{C} \to \mathsf{IPCA}_\mathcal{D}$ as follows.

- If (A, ϕ) is an IPCA over $\mathcal{C}$, then $p^*(A, \phi) := (p(A), \langle p(\phi)\rangle)$.
- If $f\colon (A, \phi) \to (B, \psi)$ is an applicative morphism, then $p^*(f)$ is the relation $p(f) \subseteq p(A) \times p(B)$.

Proposition 4.3.22. *If $p\colon \mathcal{C} \to \mathcal{D}$ is a regular functor, then p^* as defined in Construction 4.3.21 is a well-defined 2-functor $\mathsf{IPCA}_\mathcal{C} \to \mathsf{IPCA}_\mathcal{D}$. Moreover, Construction 4.3.21 is functorial in the sense that $\mathrm{id}^*_\mathcal{C} = \mathrm{id}_{\mathsf{IPCA}_\mathcal{C}}$ and $(qp)^* = q^*p^*$.*

Proof. First, let us show that $p^*(A, \phi)$ is an IPCA over $\mathcal{D}$. In general, if $U \in \mathcal{T}A$ realizes $\lambda^*\vec{x}, y.t$ w.r.t. the IPAP A, then $p(U) \in \mathcal{T}(p(A))$ realizes $\lambda^*\vec{x}, y.t$ w.r.t. the IPAP $p(A)$, since p preserves regular logic. In particular, if $\mathsf{K}, \mathsf{S} \in \phi$ witness the fact that (A, ϕ) is an IPCA over $\mathcal{C}$, then $p(\mathsf{K}), p(\mathsf{S}) \in p(\phi) \subseteq \langle p(\phi)\rangle$ witness the fact that $p^*(A, \phi)$ is an IPCA over $\mathcal{D}$.

Now suppose that $f\colon (A, \phi) \to (B, \psi)$ is an applicative morphism. Since p preserves regular logic, we have that $p(f) \subseteq p(A) \times p(B)$ satisfies the first two requirements from Definition 4.3.12. Moreover, if f preserves application and the order up to $T, U \in \psi$, then $p(f)$ preserves application and the order up to $p(T), p(U)$, again because p preserves regular logic. Since $T, U \in \psi$, we have $p(U), p(T) \in p(\psi) \subseteq \langle p(\psi)\rangle$, so it remains to check that $p(f)$ maps $\langle p(\phi)\rangle$ to $\langle p(\psi)\rangle$. For this, we use Lemma 4.3.18. Suppose that $V \in p(\phi)$, so $V = p(V')$ for some $V' \in \phi$. Then $f(V') \in \psi$, and since p is regular, we have $p(f)(V) = p(f)(p(V')) = p(f(V')) \in p(\psi) \subseteq \langle p(\psi)\rangle$, as desired.

In order to see that p^* preserves the order on homsets, we note: if $f, f'\colon A \times B$ are relations such that $V \in \psi$ realizes $f \leq f'$, then $p(V) \in p(\psi) \subseteq \langle p(\psi)\rangle$ realizes $p(f) \leq p(f')$.

Using the fact that p is regular once more, it is clear that p^* preserves identities and composition. For the final statement, the only nontrivial thing to check is that $q^*(p^*(A, \phi))$ and $(qp)^*(A, \phi)$ are equipped with the same external filter. This follows from Lemma 4.3.19(ii), which tells us that:

$$\langle q(\langle p(\phi)\rangle)\rangle = \langle q(p(\phi))\rangle = \langle (qp)(\phi)\rangle.$$ □

Remark 4.3.23. Consider the map $\mathcal{T}(A, \phi) \to \mathcal{T}(p^*(A, \phi))$ given by $V \mapsto p(V)$. By Lemma 4.3.19(i), we have $p(U) \cdot p(V) \preceq p(UV)$, which implies that this map is a morphism of PCAs.

Can we extend Construction 4.3.21 to natural transformations, thus obtaining a 2-functor from REG to the category of preorder-enriched bicategories? If $p, q \colon \mathcal{C} \to \mathcal{D}$ are regular functors and μ is a natural transformation $p \Rightarrow q$, then we can define:

$$\overline{\mu}_A = \{(a, b) \in p(A) \times q(A) \mid \mu_A(a) \geq b\} \tag{4.6}$$

for IPAPs A over $\mathcal{C}$. If (A, ϕ) is an IPCA over $\mathcal{C}$, then $\overline{\mu}_A$ does not in general seem to be an applicative morphism $p^*(A, \phi) \to q^*(A, \phi)$. Specifically, if $U \in \phi$, the applying the naturality of μ to the inclusion $U \subseteq A$ tells us that $\overline{\mu}_A(p(U)) \subseteq q(U)$. However, if we want to show that $\overline{\mu}_A(p(U)) \in q(\psi)$, then we would need the *reverse* inequality, which does not seem to hold in general. Still, $\overline{\mu}_A$ is a relation between $p(A)$ and $q(A)$, and relations can be composed. The following lemma says that the naturality of $\overline{\mu}$ is lax.

Lemma 4.3.24. *Let $f \colon (A, \phi) \to (B, \psi)$ be an arrow of $\mathsf{IPCA}_{\mathcal{C}}$, and let $\mu \colon p \Rightarrow q$, where $p, q \colon \mathcal{C} \to \mathcal{D}$ are regular functors. Then $\overline{\mu}_B \circ p(f) \subseteq q(f) \circ \overline{\mu}_A$.*

Proof. Applying the naturality of μ to the inclusion $f \subseteq A \times B$ shows that $p(f)(a, b) \vDash_{a:p(A);b:p(B)} q(f)(\mu_A(a), \mu_B(b))$. Now the statement of the lemma follows by an easy internal argument in $\mathcal{D}$. □

In any case, Construction 4.3.21 defines a 1-functor from REG to preorder-enriched bicategories. In such a situation (ignoring for the moment that $\mathsf{OPCA}_{\mathcal{C}}$ is merely a bicategory), we can perform (the opposite of) the Grothendieck construction and obtain a category $\int \mathsf{IPCA}_{(-)}$ that is opfibred over REG. Since Construction 4.3.21 does not seem to extend to natural transformations, it is not *a priori* clear that $\int \mathsf{IPCA}_{(-)}$ will be a bicategory, but it turns out that $\int \mathsf{IPCA}_{(-)}$ is, in fact, a bicategory. First, let us give the full definition of $\int \mathsf{IPCA}_{(-)}$; we will denote this category simply by IPCA.

Definition 4.3.25. *The bicategory IPCA is defined as follows.*

(i) The objects are triples $(\mathcal{C}, A, \phi)$, where (A, ϕ) is an IPCA over the regular category $\mathcal{C}$. We will usually just write (A, ϕ) instead of $(\mathcal{C}, A, \phi)$.

(ii) If (A, ϕ) and (B, ψ) are IPCAs over $\mathcal{C}$ and $\mathcal{D}$ respectively, then an arrow $(A, \phi) \to (B, \psi)$ is a pair (p, f), where $p \colon \mathcal{C} \to \mathcal{D}$ is a regular functor and $f \colon p^*(A, \phi) \to (B, \psi)$ is an applicative morphism.

(iii) A 2-cell $(p, f) \Rightarrow (q, g)$ is a natural transformation $\mu \colon p \Rightarrow q$ such that $f \leq g \circ \overline{\mu}_A$.

*An arrow of IPCA is also called an **applicative morphism**, whereas a 2-cell is called an **applicative transformation**.*

In order to prove that this is indeed a bicategory, we need to have a closer look at the proof of Proposition 4.3.15. First of all, we have shown that, if (A,ϕ) and (B,ψ) are IPCAs over $\mathcal{C}$, the set of *all* relations from A to B form a preorder. That is, we do not need that the relations are applicative morphisms. Moreover, the inference $g \le g' \implies gf \le g'f$ holds for all relations f, g, g'. On the other hand, the inference $f \le f' \implies gf \le gf'$ holds for all relations f and f', but we do need that g is an applicative morphism. Furthermore, in the proof of Proposition 4.3.22, we saw that a regular functor $p\colon \mathcal{C} \to \mathcal{D}$ preserves the order on *all* relations between IPCAs over $\mathcal{C}$, not merely on applicative morphisms.

Theorem 4.3.26. *The bicategory* IPCA *as defined in Definition 4.3.25 is indeed a bicategory.*

Proof. The identity on (A,ϕ) is $(\mathrm{id}_{\mathcal{C}}, \mathrm{id}_{(A,\phi)})$. The composition of two applicative morphisms $(A,\phi) \stackrel{(p,f)}{\longrightarrow} (B,\psi) \stackrel{(q,g)}{\longrightarrow} (C,\chi)$ is given by $(qp, g\circ q^*f) = (qp, g\circ q(f))$. It is easily verified that the axioms for a 1-category hold up to isomorphism. Again, the only identity that does not hold on the nose is $(p,f)\circ(\mathrm{id},\mathrm{id}) \cong (p,f)$, which holds up to the invertible 2-cell id_p.

We define the vertical and horizontal composition of 2-cells as in REG. Suppose that we have parallel applicative morphisms $(p,f),(q,g),(r,h)\colon (A,\phi) \to (B,\psi)$ and that

$$(p,f) \stackrel{\mu}{\Rightarrow} (q,g) \stackrel{\nu}{\Rightarrow} (r,h)$$

are applicative transformations. Then $\overline{\nu}_A \circ \overline{\mu}_A = \overline{\mu\nu}_A$, as follows by an easy internal argument in $\mathcal{C}$ (using the fact that $\nu_A\colon q(A) \to r(A)$ is order-preserving). Now we see that $\nu\mu$ is an applicative transformation $(p,f) \Rightarrow (r,h)$ since $f \le g\circ\overline{\mu}_A \le h\circ\overline{\nu}_A\circ\overline{\mu}_A = h\circ\overline{\nu\mu}_A$.

Now suppose that $(p,f),(q,g)\colon (A,\phi) \to (B,\psi)$ are applicative morphisms and that μ is an applicative transformation $(p,f) \Rightarrow (q,g)$. Let $(r,h)\colon (B,\psi) \to (C,\chi)$ be another applicative morphism. Since r is left exact, we have $r(\overline{\mu}_A) = \overline{r\mu}_A$. Now we see that

$$h\circ r(f) \le h\circ r(g\circ\overline{\mu}_A) = h\circ r(g)\circ r(\overline{\mu}_A) = h\circ r(g)\circ\overline{r\mu}_A,$$

so $r\mu$ is an applicative transformation $(r,h)\circ(p,f) \Rightarrow (r,h)\circ(q,g)$.

On the other hand, if $(r,h)\colon (C,\chi) \to (A,\phi)$ is another applicative morphism, then, using Lemma 4.3.24:

$$f\circ p(h) \le g\circ\overline{\mu_A}\circ p(h) \le g\circ q(h)\circ\overline{\mu}_{r(C)} = g\circ q(h)\circ\overline{\mu r}_C,$$

so μr is an applicative transformation $(p,f)\circ(r,h) \Rightarrow (q,g)\circ(r,h)$.

Finally, all the required coherence conditions follow trivially from the fact that REG is a 2-category. □

Remark 4.3.27. Consider an applicative morphism $(p,f)\colon (A,\phi) \to (B,\psi)$. Combining Remark 4.3.13(i) and Remark 4.3.23, we see that the map $\mathcal{T}(A,\phi) \to \mathcal{T}(B,\psi)$ given by $U \mapsto f(p(U))$ is a morphism of PCAs, and we will denote it by

$\mathcal{T}(p, f)$. Moreover, if $\mu\colon (p, f) \Rightarrow (q, g)$ is an applicative transformation, then any realizer of $f \leq g \circ \overline{\mu}_A$ also realizes $\mathcal{T}(p, f) \leq \mathcal{T}(q, g)$. It is easy to check that this makes $\mathcal{T}$ into a 2-functor $\mathsf{IPCA} \to \mathsf{PCA}$.

Before we proceed to define the category of assemblies in the next section, let us have a closer look at the IPCA $p^*(A, \phi) = (p(A), \langle p(\phi)\rangle)$. By Lemma 2.1.14, the external filter $\langle p(\phi)\rangle$ can be described as:

$$\uparrow \left\{ t(p(\vec{U})) \mid t(\vec{x}) \text{ a pure term}, \vec{U} \in \phi \text{ and } t(p(\vec{U}))\downarrow \right\}.$$

But as we saw in Remark 4.3.20, the fact that $t(p(\vec{U}))$ is defined does *not* imply that $t(\vec{U})\downarrow$. In the sequel, it will be convenient to have an alternative description of $\langle p(\phi)\rangle$ and its finite extensions as defined in Definition 4.3.11.

Lemma 4.3.28. *Let (A, ϕ) be an IPCA over $\mathcal{C}$, let $p\colon \mathcal{C} \to \mathcal{D}$ be a regular functor, and let $X \in \mathcal{T}(p(A))$. Then:*

$$\langle p(\phi) \cup \{X\}\rangle = \uparrow\{p(U) \cdot X \mid U \in \phi \textit{ and } p(U) \cdot X\downarrow\}.$$

Proof. Clearly, any filter on $\mathcal{T}(p(A))$ containing $p(\phi) \cup \{X\}$ must contain the set $\uparrow\{p(U) \cdot X \mid U \in \phi \text{ and } p(U) \cdot X\downarrow\}$, so it remains to show that the latter is itself a filter on $\mathcal{T}(p(A))$. Upwards closure holds by definition, so we need to verify closure under defined application. Consider $V, V' \in \mathcal{T}(A)$ such that $VV'\downarrow$, and suppose that we have $U, U' \in \phi$ such that $p(U) \cdot X \subseteq V$ and $p(U') \cdot X \subseteq V'$. As observed in the proof of Proposition 4.3.15, if $\mathsf{S} \in \phi$ realizes $\lambda^* xyz.xz(yz)$ w.r.t. A, then $p(\mathsf{S})$ realizes $\lambda^* xyz.xz(yz)$ w.r.t. $p(A)$. Now we see that $U'' = \mathsf{S}UU'$ is defined and an element of ϕ, and:

$$p(U'') \cdot X \simeq p(\mathsf{S}) \cdot p(U) \cdot p(U') \cdot X \preceq p(U) \cdot X \cdot (p(U') \cdot X) \subseteq VV',$$

as desired. □

Corollary 4.3.29. *If (A, ϕ) is an IPCA over $\mathcal{C}$ and $p\colon \mathcal{C} \to \mathcal{D}$ is a regular functor, then:*

$$\langle p(\phi)\rangle = \uparrow\{p(U) \cdot p(A) \mid U \in \phi \textit{ and } p(U) \cdot p(A)\downarrow\}.$$

Proof. Since $A \in \phi$, we have $p(A) \in p(\phi)$, so $\langle p(\phi)\rangle = \langle p(\phi) \cup \{p(A)\}\rangle$. Now apply Lemma 4.3.28. □

4.3.3 Assemblies and the realizability topos

In this section, we primarily discuss the category of assemblies for an IPCA (A, ϕ), and regular functors between these categories of assemblies. We will also, for the case where the base category is a topos, define the realizability topos.

In the case of PCAs, an assembly is a set $|X|$ equipped with a function $E_X : |X| \to TA$. In the current case, this definition is not available, because TA cannot be defined inside $\mathcal{C}$. We use the same solution as we did for applicative morphisms, that is, we view E_X as a relation between $|X|$ and A.

Definition 4.3.30. *Let (A, ϕ) be an IPCA over $\mathcal{C}$.*

*(i) An **assembly** over (A, ϕ) is a pair $X = (|X|, E_X)$, where $|X|$ is an object of $\mathcal{C}$ and $E_X \subseteq |X| \times A$ satisfies:*

$$\vDash_{x:|X|} \exists a\colon A(E_X(x,a)) \quad \text{and} \quad E_X(x,a) \wedge a' \leq a \vDash_{x:|X|;a,a':A} E_X(x,a').$$

*(ii) A **morphism of assemblies** $f\colon X \to Y$ is a function $f\colon |X| \to |Y|$ for which there exists a $T \in \phi$, called a **tracker** of f, such that:*

$$T(t) \wedge E_X(x,a) \vDash_{x:|X|;t,a:A} ta{\downarrow} \wedge E_Y(f(x), ta).$$

As for the case of ordinary PCAs, the category of assemblies has the following properties.

Proposition 4.3.31. *Assemblies over (A, ϕ) and morphisms between them form a regular category that we denote by $\mathsf{Asm}(A, \phi)$. Moreover, there is an adjunction*

$$\mathcal{C} \underset{\nabla}{\overset{\Gamma}{\underset{\longrightarrow}{\overset{\longleftarrow}{\bot}}}} \mathsf{Asm}(A, \phi)$$

where Γ and ∇ are both regular, and $\Gamma\nabla = \mathrm{id}_{\mathcal{C}}$.

Proof. The proof is of course similar to the proof of Proposition 3.1.2, with appropriate adjustments. We give the main constructions required for the proof.

- Γ is the obvious forgetful functor given by $\Gamma X = |X|$ and $\Gamma f = f$.
- If X is an object of $\mathcal{C}$, the $\nabla X = (X, X \times A)$.
- If X and Y are assemblies, then their product is given by $|X \times Y| = |X| \times |Y|$, and:

$$E_{X\times Y}(x,y,a) :\equiv \exists b,c,d\colon A(\mathsf{P}(b) \wedge E_X(x,c) \wedge E_Y(y,d) \wedge a \leq bcd).$$

- If $f, g\colon X \to Y$ are morphisms, then their equalizer $m\colon U \hookrightarrow X$ is constructed by first taking the equalizer $m\colon |U| \hookrightarrow |X|$ of $f, g\colon |X| \to |Y|$ in $\mathcal{C}$, and then putting $E_U(u,a) :\equiv E_X(m(u), a)$.
- A morphism $e\colon X \to Y$ is regular epi iff there exists a $U \in \phi$ such that

$$U(r) \wedge E_Y(y,a) \vDash_{y:|Y|;r,a:A} ra{\downarrow} \wedge \exists x\colon |X|(e(x) = y \wedge E_X(x, ra)). \qquad \square$$

Of course, we will say that an assembly X is **constant** if it is isomorphic to an object in the image of ∇. This is equivalent to saying that there exists a $U \in \phi$ such that $U(a) \vDash_{x:|X|,a:A} E_X(x,a)$, or more succinctly, $|X| \times U \subseteq E_X$. Moreover, we say that a morphism of assemblies is **prone** if its naturality square for $\eta\colon \mathrm{id}_{\mathsf{Asm}(A,\phi)} \Rightarrow \nabla\Gamma$ is a pullback. This is equivalent to the existence of a **reverse tracker** $T' \in \phi$ satisfying:

$$T'(t') \wedge E_Y(f(x), a) \vDash_{x:|X|;t',a:A} t'a{\downarrow} \wedge E_X(x, t'a).$$

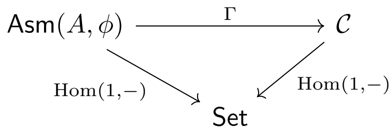
Now we see that $m\colon X \hookrightarrow Y$ is a regular mono iff $m\colon |X| \hookrightarrow |Y|$ is a regular mono in $\mathcal{C}$, and m is prone.

Remark 4.3.32. The definition of $\mathsf{Asm}(A, \phi)$ only works well under the assumption that every member of ϕ is inhabited. As we mentioned in Remark 4.3.5(ii), this is not assumed in [Ste13], but it is assumed in our setup. For an example of why this is important, consider the characterization of regular epis given at the end of the proof of Proposition 4.3.31. Since Γ is supposed to be a regular functor, this characterization had better imply that $e \colon |X| \to |Y|$ is also regular epi in $\mathcal{C}$. And this is indeed the case, as follows by reasoning internally in $\mathcal{C}$, but the argument uses that U is inhabited.

For the more general case where members of ϕ need not be inhabited, one can also define the category of assemblies $\mathsf{Asm}(A, \phi)$. In [Ste13, Theorem 2.2.14], it is shown that every category of the form $\mathsf{Asm}(A, \phi)$ is equivalent to one where every member of ϕ is inhabited. So we see that our assumption that ϕ consists of inhabited downsets is not very restrictive.

We have the following analogue of Lemma 3.1.5.

Lemma 4.3.33. *If (A, ϕ) is an IPCA over $\mathcal{C}$, then (A, ϕ) is absolute iff the diagram*



commutes up to isomorphism.

Proof. First, suppose that $\phi = \mathcal{T}A$; we need to show that, if X is an assembly, every global section $x \colon 1 \to |X|$ is also a global section $1 \to X$. If we define $U := \{a \in A \mid E_X(x, a)\} \in \mathcal{T}A$, then $\mathsf{K}U$ realizes $x \colon 1 \to X$, as desired.

Conversely, suppose that there exists a $U \in \mathcal{T}A$ such that $U \notin \phi$. Then we can define an assembly 1_U with $|1_U| = 1$ and $E_{1_U} = U \subseteq A \cong 1 \times A$, which has no global sections. $\square$

Next, we define, for each applicative morphism $(A, \phi) \to (B, \psi)$, a regular functor $\mathsf{Asm}(A, \phi) \to \mathsf{Asm}(B, \psi)$. As we did in the previous chapter, we will write Γ_A and ∇_A to disambiguate the various functors involved. Of course, we should really write $\Gamma_{(A,\phi)}$ and $\nabla_{(A,\phi)}$, but this would clutter notation too much.

Construction 4.3.34. Let $(p, f) \colon (A, \phi) \to (B, \psi)$ be an applicative morphism, where $p \colon \mathcal{C} \to \mathcal{D}$.

(i) We define a functor $\mathsf{Asm}(p, f) \colon \mathsf{Asm}(A, \phi) \to \mathsf{Asm}(B, \psi)$ as follows. If X is an object of $\mathsf{Asm}(A, \phi)$, then $|\mathsf{Asm}(p, f)(X)| := p|X|$ and

$$E_{\mathsf{Asm}(p,f)(X)}(x, b) :\equiv \exists a : p(A) \, (p(E_X)(x, a) \wedge f(a, b)).$$

If $g \colon X \to Y$ is a morphism of assemblies, then $\mathsf{Asm}(p, f)(g) := p(g)$.

(ii) If $\mu \colon (p, f) \Rightarrow (q, g)$ is an applicative transformation, then we define the natural transformation $\mathsf{Asm}(\mu) \colon \mathsf{Asm}(p, f) \Rightarrow \mathsf{Asm}(q, g)$ by $\mathsf{Asm}(\mu)_X = \mu_{|X|} \colon p|X| \to q|X|$.

Proposition 4.3.35. *If* $(p,f)\colon (A,\phi)\to(B,\psi)$ *is an applicative morphism, then* $\mathsf{Asm}(p,f)$ *is a well-defined regular functor* $\mathsf{Asm}(A,\phi)\to\mathsf{Asm}(B,\psi)$*, and we have* $\Gamma_B\circ\mathsf{Asm}(p,f)\cong p\circ\Gamma_A$ *and* $\mathsf{Asm}(p,f)\circ\nabla_A\cong\nabla_B\circ p$*. Moreover, if* $\mu\colon(p,f)\Rightarrow(q,g)$ *is an applicative transformation, then* $\mathsf{Asm}(\mu)$ *is a well-defined natural transformation* $\mathsf{Asm}(p,f)\Rightarrow\mathsf{Asm}(q,g)$*. Finally, this makes* Asm *into a 2-functor* $\mathsf{IPCA}\to\mathsf{REG}$*.*

Proof. We easily verify that $\mathsf{Asm}(p,f)=\mathsf{Asm}(\mathrm{id}_{\mathcal{D}},f)\circ\mathsf{Asm}(p,\mathrm{id}_{p^*(A,\phi)})$. Thus, in order to verify that $\mathsf{Asm}(p,f)$ is well-defined and regular, it suffices to check that both $\mathsf{Asm}(\mathrm{id}_{\mathcal{D}},f)$ and $\mathsf{Asm}(p,\mathrm{id}_{p^*(A,\phi)})$ are well-defined and regular. For $\mathsf{Asm}(\mathrm{id}_{\mathcal{D}},f)$, this follows of course by an 'internalization' of the proof that $\mathsf{Asm}(f)$ is well-defined and regular for arrows f of OPCA_T. We omit this part of the proof, and check that $\mathsf{Asm}(p,\mathrm{id}_{p^*(A,\phi)})$ is well-defined and regular.

We will abbreviate $\mathsf{Asm}(p,\mathrm{id}_{p^*(A,\phi)})$ by F, so that $|FX|=|X|$, $E_{FX}=p(E_X)$ and $F(g)=p(g)$ for morphisms $g\colon X\to Y$. First, we need to check that $p(g)$ is also a morphism of assemblies. Suppose that $T\in\phi$ tracks $g\colon X\to Y$, that is:

$$T(t)\wedge E_X(x,a)\vDash_{x:|X|;t,a:A} ta{\downarrow}\wedge E_Y(g(x),ta)$$

Since p preserves regular logic, this implies:

$$p(T)(t)\wedge p(E_X)(x,a)\vDash_{x:p|X|;t,a:p(A)} ta{\downarrow}\wedge p(E_Y)(p(g)(x),ta),$$

so $p(T)$ tracks $p(g)\colon FX\to FY$. Since $p(T)\in p(\phi)\subseteq\langle p(\phi)\rangle$, this shows that $p(g)$ is a morphism of assemblies $p(g)$. For finite limits, the only nontrivial thing to check is that F preserves binary products. If X and Y are assemblies over (A,ϕ), then since p preserves regular logic, we have $p(E_{X\times Y})(x,y,a)$ iff:

$$\exists b,c,d{:}p(A)\,(p(\mathsf{P})(b)\wedge p(E_X)(x,c)\wedge p(E_Y)(y,d)\wedge a\leq bcd).$$

Again because p preserves regular logic, we know that $p(\mathsf{P})\in\langle p(\phi)\rangle$ realizes $\lambda^*xyz.zxy$, which shows that $\mathrm{id}_{p|X|\times p|Y|}$ is an isomorphism between $F(X\times Y)$ and $FX\times FY$. The regularity of F follows by a similar argument, using the characterization of regular epis given at the end of the proof of Proposition 4.3.31.

The fact that $\mathsf{Asm}(p,f)$ is compatible with the Γ- and ∇-functors is easily verified. Now let $\mu\colon(p,f)\Rightarrow(q,g)$ be an applicative transformation. We need to check that $\mu_{|X|}$ is a morphism of assemblies $\mathsf{Asm}(p,f)(X)\to\mathsf{Asm}(q,g)(X)$. Applying the naturality of μ to the inclusion $E_X\subseteq|X|\times A$ yields:

$$p(E_X)(x,a)\vDash_{x:p|X|;a:p(A)} q(E_X)(\mu_{|X|}(x),\mu_A(a)).$$

Now it easily follows that any realizer of $f\leq g\circ\overline{\mu}_A$ also tracks $\mu_{|X|}$ as a morphism $\mathsf{Asm}(p,f)(X)\to\mathsf{Asm}(q,g)(X)$, as desired. Finally, the functoriality of Asm is easily verified. □

Note that, analogously to $\mathsf{Asm}\colon\mathsf{OPCA}_T\to\mathsf{REG}$, we have that $\mathsf{Asm}\colon\mathsf{IPCA}\to\mathsf{REG}$ is a 2-functor even though IPCA is merely a bicategory.

As in the case of ordinary PCAs, we can show that *all* regular functors between categories of assemblies that are compatible with both Γ and ∇, must arise from an applicative morphism. The proof of the following proposition is of course similar to the proof of Theorem 3.3.13, but we need to make a few adjustments. Most importantly, the object of nonempty downsets T_A used in the proof of Theorem 3.3.13 is not available in the current setting. On the other hand, we do still have the object of realizers $R_A := (A, \geq)$, which will play a vital role.

Proposition 4.3.36. *Let (A, ϕ) and (B, ψ) be IPCAs over $\mathcal{C}$ resp. $\mathcal{D}$, and suppose that $p\colon \mathcal{C} \to \mathcal{D}$ and $F\colon \mathsf{Asm}(A, \phi) \to \mathsf{Asm}(B, \psi)$ are regular functors such that $\Gamma_B F \cong p\Gamma_A$ and $F\nabla_A \cong \nabla_B p$. Then there exists an applicative morphism $(p, f)\colon (A, \phi) \to (B, \psi)$ such that $F \cong \mathsf{Asm}(p, f)$.*

Proof. We may assume that $\Gamma_B F = p\Gamma_A$ on the nose, that is, $|FX| = p|X|$ for every assembly X over (A, ϕ). First, we show that F preserves prone morphisms, using a trick by Longley [Lon94, Proposition 1.4.4]. Consider the naturality diagram:

$$\begin{array}{ccc} F & \xrightarrow{\quad \eta F \quad} & \nabla_B\Gamma_B F \\ {\scriptstyle F\eta}\downarrow & & \downarrow{\scriptstyle \nabla_B\Gamma_B F\eta} \\ F\nabla_A\Gamma_A & \xrightarrow{\eta F\nabla_A\Gamma_A} & \nabla_B\Gamma_B F\nabla_A\Gamma_A \end{array}$$

Since $\Gamma_B F \cong p\Gamma_B$, we have that $\nabla_B\Gamma_B F\eta$ is an isomorphism, and since $F\nabla_A \cong \nabla_B p$, we have that $\eta F\nabla_A\Gamma_A$ is an isomorphism. Thus, we have an isomorphism $F\nabla_A\Gamma_A \cong \nabla_B\Gamma_B F$ that identifies $F\eta$ and ηF. Since F is also left exact, this implies that F preserves prone morphisms.

Now consider $FR_A \in \mathsf{Asm}(B, \psi)$: we have $|FR_A| = p|R_A| = p(A)$, which means that $f := E_{FR_A}$ is a subobject of $p(A) \times B$. We claim that (p, f) is the desired applicative morphism.

Requirements (i) and (ii) from Definition 4.3.12 hold by definition. In order to see that f preserves application up to a realizer, consider the prone subobject P of $R_A \times R_A$ with $|P| = D \subseteq A \times A$ and the morphism of assemblies $\mathrm{app}\colon P \to R_A$ given by the application map. Then FP is a prone subobject of $FR_A \times FR_A$, and from a tracker of $F(\mathrm{app})$ we then easily construct the required realizer. Similarly, in order to see that f preserves the order up to a realizer, we can consider the assembly O with $|O| = \{(a, a') \in A \times A \mid a \leq a'\}$ and $E_O(a, a', a'') :\equiv a \geq a''$. Then the first projection $\pi_0\colon O \to R_A$ is prone, and the second projection $\pi_1\colon O \to R_A$ is a morphism of assemblies. Now, using the fact that $F\pi_0$ also prone, and the fact that $F\pi_1$ is a morphism of assemblies, one easily constructs the desired realizer. For the final requirement on f, we use Lemma 4.3.18. Thus, suppose that $U \in \phi$; we need to show that $f(p(U)) \in \psi$. Consider the prone subobject R_U of R_A with $|R_U| = U$. Then $R_U \to 1$ is regular epi, so $FR_U \to F1 \cong 1$ is regular epi as well. Combining this with the fact that FR_U is a prone subobject of FR_A, one easily deduces that $f(p(U)) \in \psi$.

It remains to show that $F \cong \mathsf{Asm}(p, f)$. Let X be an assembly over (A, ϕ). Consider the prone subobject Y of $\nabla|X| \times R_A$ given by $|Y| = E_X \subseteq |X| \times A = |\nabla|X|| \times |R_A|$. Then the first projection $\pi\colon |Y| \to |X|$ is easily seen to be a regular epimorphism $Y \twoheadrightarrow X$. Now, using the diagram:

$$\begin{array}{ccc} FY & \xhookrightarrow{\text{prone}} & F\nabla|X| \times FR_A \;\cong\; \nabla p|X| \times FR_A \\ \big\downarrow & & \\ FX & & \end{array}$$

we can deduce that $\mathrm{id}_{p|X|}$ is an isomorphism of assemblies $FX \cong \mathsf{Asm}(p,f)(X)$, which yields $F \cong \mathsf{Asm}(p,f)$, as desired. □

In the case of applicative morphisms between ordinary PCAs, we showed that Asm yields a local equivalence from OPCA_T to $\mathsf{REG}/\mathsf{Set}$. This result of course required that every regular functor $\mathsf{Asm}(A) \to \mathsf{Asm}(B)$ that commutes with the Γs, also commutes with the ∇s. The proof of this, which was given in Lemma 3.3.12(ii), was nonconstructive and thus does not seem to be available in the current setting. However, we may still present Asm as a local equivalence by considering functors that commute with *both* Γ and ∇. Longley uses the notion of a '$\nabla\Gamma$-functor' in [Lon94]. We will formulate the result a bit differently, in terms of a functor category.

Let $\mathcal{I}$ be the 2-category with two objects 0 and 1, which is freely generated by 1-cells $\ell\colon 1 \to 0$ and $r\colon 0 \to 1$ and a 2-cell $\eta\colon \mathrm{id}_1 \Rightarrow rl$, subject to the equations $lr = \mathrm{id}_0$, $\ell\eta = \mathrm{id}_\ell$ and $\eta r = \mathrm{id}_r$. Then a pseudofunctor $\mathcal{I} \to \mathsf{REG}$ is simply a geometric inclusion between regular categories. Moreover, these pseudofunctors form a 2-category $\mathsf{REG}^{\mathcal{I}}$ in the straightforward way, where the 1-cells are pseudonatural transformations and the 2-cells are modifications.

Theorem 4.3.37. *The 2-functor* $\mathsf{IPCA} \to \mathsf{REG}^{\mathcal{I}}$ *that sends:*

- *an IPCA* (A,ϕ) *over* $\mathcal{C}$ *to* $\Gamma \dashv \nabla\colon \mathcal{C} \hookrightarrow \mathsf{Asm}(A,\phi)$*;*
- *an applicative morphism* (p,f) *to the pair* $(p, \mathsf{Asm}(p,f))$*, equipped with the natural isomorphisms from Proposition 4.3.35;*
- *an applicative transformation* $\mu\colon (p,f) \Rightarrow (q,g)$ *to the pair* $(\mu, \mathsf{Asm}(\mu))$*,*

is a local equivalence.

Proof. It is readily checked that $\mathsf{IPCA} \to \mathsf{REG}^{\mathcal{I}}$ as given in the theorem is well-defined and a 2-functor. Now let (A,ϕ) and (B,ψ) be IPCAs over $\mathcal{C}$ resp. $\mathcal{D}$. We need to check that

$$\mathsf{IPCA}((A,\phi),(B,\psi)) \to \mathsf{REG}^{\mathcal{I}}(\mathcal{C} \hookrightarrow \mathsf{Asm}(A,\phi), \mathcal{D} \hookrightarrow \mathsf{Asm}(B,\psi))$$

is an equivalence of categories. Essential surjectivity follows from Proposition 4.3.36, and faithfulness holds automatically. For fullness, we need to check the following: if $(p,f),(q,g)\colon (A,\phi) \to (B,\psi)$ and $\mu\colon p \Rightarrow q$ are such that $X \mapsto \mu_{|X|}$ is a natural transformation $\mathsf{Asm}(p,f) \Rightarrow \mathsf{Asm}(q,g)$, then μ is an applicative transformation $(p,f) \Rightarrow (q,g)$. If $R_A = (A, \geq)$ is the object of realizers, then $\mathrm{id}_{p(A)}$ is an isomorphism $\mathsf{Asm}(p,f)(R_A) \cong (p(A), f)$, and similarly, $\mathrm{id}_{q(A)}$

is an isomorphism $\mathsf{Asm}(B,\psi)(R_A) \cong (q(A), g)$. Thus, μ_A must be a morphism of assemblies $(p(A), f) \to (q(A), g)$, and a tracker of this morphism is precisely a realizer of $f \leq g \circ \overline{\mu}_A$. □

Finally, let us treat realizability toposes for internal PCAs. If (A,ϕ) is an IPCA over $\mathcal{C}$, then we expect the realizability topos for (A,ϕ) to arise from a tripos $\mathsf{P}_{A,\phi}$ on $\mathcal{C}$. If X is an object of $\mathcal{C}$, then an element of $\mathsf{P}_{A,\phi}X$ should be a subobject $U \subseteq X \times A$ such that $U(x,a) \wedge a' \leq a \vDash_{x:X;a,a':A} U(x,a')$. This certainly yields a pseudofunctor $\mathsf{P}_{A,\phi}\colon \mathcal{C} \to \mathsf{PreOrd}^{\mathrm{op}}$ (where substitution is given by pullback), but if $\mathcal{C}$ is merely regular, $\mathsf{P}_{A,\phi}$ will not be a tripos. In particular, if $\mathsf{P}_{A,\phi}$ is to have a generic predicate, then the 'object of downsets of A' must be present inside $\mathcal{C}$ itself. The most obvious way to achieve this is to assume that $\mathcal{C}$ is a topos. In this case, $\mathsf{P}_{A,\phi}$ is indeed a tripos, which means that $\mathcal{C}[\mathsf{P}_{A,\phi}]$ is a topos. Moreover, its category of assemblies, that is, the full subcategory on the subobjects of the image of $\nabla_{\mathsf{P}_{A,\phi}}\colon \mathcal{C} \to \mathcal{C}[\mathsf{P}_{A,\phi}]$, is equivalent to $\mathsf{Asm}(A,\phi)$. The proof of this fact is similar to the proof we gave in the case of ordinary PCAs in Section 3.2.1, but one has to replace 'nonempty' by 'inhabited' everywhere.

For the official definition of the realizability topos for IPCAs, we use the alternative approach to toposes-from-triposes given in Remark 3.2.7.

Definition 4.3.38. *If (A,ϕ) is an IPCA over the* topos *$\mathcal{C}$, then we define its **realizability topos** $\mathsf{RT}(A,\phi)$ as $\mathsf{Asm}(A,\phi)_{\mathrm{ex/reg}}$.*

Let us write $\mathsf{IPCA}_{\mathrm{top}}$ for the full (on 1- and 2-cells) sub-bicategory of IPCA on the IPCAs constructed over toposes. Then RT automatically extends to a pseudofunctor $\mathsf{IPCA}_{\mathrm{top}} \to \mathsf{REG}$, by means of Proposition 3.2.4. Moreover, as in Chapter 3, we write i for the inclusion $\mathsf{Asm}(A,\phi) \to \mathsf{RT}(A,\phi)$, we let $\hat{\nabla} = i\nabla$, and we write $\hat{\Gamma}$ for the essentially unique extension of Γ along i. Then as in Chapter 3, $\hat{\Gamma} \dashv \hat{\nabla}$ is a geometric inclusion $\mathcal{C} \hookrightarrow \mathsf{RT}(A,\phi)$.

4.4 Applications

In the previous section, we introduced the framework of IPCAs, and the notions of applicative morphism and assembly. In this section, we treat several applications of this framework. The most important application is that categories of the form $\mathsf{Asm}(A,\phi)$ are closed under small (2-)products and under slicing. Moreover, we will connect products and slices of categories of the form $\mathsf{Asm}(A,\phi)$ to the notion of computational density. Finally, we will partially extend these results to realizability toposes, for the case where the base category is a topos.

4.4.1 Products of categories of assemblies

In this section, we show that categories of the form $\mathsf{Asm}(A,\phi)$ are closed under arbitrary (small) products. Moreover, we investigate the existence of pseudocoproducts in IPCA, and their interaction with 2-products.

In Section 2.4, we established that, while OPCA has all small 2-products, OPCA_T does not have any nontrivial pseudo- or 2-products. On the other hand, the possibility of base change allows us to construct all small 2-products in IPCA.

Proposition 4.4.1. *The bicategory* IPCA *has small 2-products.*

Proof. Let I be a set and suppose that for each $i \in I$, we have an IPCA (A_i, ϕ_i) over $\mathcal{C}_i$. Consider the product category $\mathcal{C} = \prod_{i \in I} \mathcal{C}_i$, which is the 2-product of the $\mathcal{C}_i$ in REG. The object $A = (A_i)_{i \in I}$ is an IPAS over $\mathcal{C}$ in the obvious way. Moreover, an element of $\mathcal{T}A$ is an I-indexed collection $(U_i)_{i \in I}$, where $U_i \in \mathcal{T}A_i$ for each $i \in I$. Thus, we can define an external filter ϕ on A by:

$$\phi = \{(U_i)_{i \in I} \mid \forall i \in I\, (U_i \in \phi_i)\}.$$

If, for each $i \in I$, we pick suitable combinators $\mathsf{K}_i, \mathsf{S}_i \in \phi_i$ for A_i, then $(\mathsf{K}_i)_{i \in I}, (\mathsf{S}_i)_{i \in I} \in \phi$ are suitable combinators for A, so (A, ϕ) is an IPCA over $\mathcal{C}$. Moreover, for each $i \in I$, we have the projection map $\pi_i \colon \mathcal{C} \to \mathcal{C}_i$, which satisfies $\pi_i^*(A, \phi) = (A_i, \phi_i)$, so in particular, we have an applicative morphism $(\pi_i, \mathrm{id}) \colon (A, \phi) \to (A_i, \phi_i)$. We claim that this makes (A, ϕ) into the 2-product of the (A_i, ϕ_i).

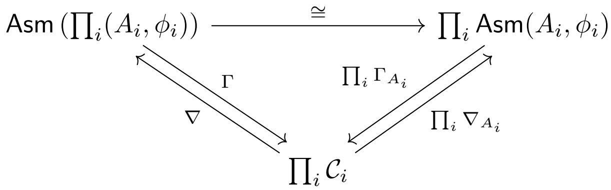
First, suppose that (B, ψ) is an IPCA over $\mathcal{D}$ and that, for each $i \in I$, we have an applicative morphism $(p_i, f_i) \colon (B, \psi) \to (A_i, \phi_i)$. The p_i have a unique amalgamation $p \colon \mathcal{D} \to \mathcal{C}$ such that $\pi_i \circ p = p_i$ for all $i \in I$. Moreover, there exists a unique relation $f \subseteq p(B) \times A$ such that $\pi_i(f) \subseteq p_i(B) \times A_i$ is equal to f_i. It is easily seen that (p, f) is an applicative morphism $(B, \psi) \to (A, \phi)$; again, we need to pick realizers $T_i, U_i \in \phi_i$ for each f_i, which then combine into realizers $T, U \in \phi$ of f. Finally, it is immediate that $(\pi_i, \mathrm{id}) \circ (p, f) = (p_i, f_i)$, so (p, f) is an amalgamation of the (p_i, f_i).

Finally, suppose that $(q, g), (r, h) \colon (B, \psi) \to (A, \phi)$ are applicative morphisms, and $\mu \colon q \Rightarrow r$. We need to show that, if μ_i is an applicative transformation $(q_i, g_i) \Rightarrow (r_i, h_i)$ for each $i \in I$, then μ is an applicative transformation $(q, g) \Rightarrow (r, h)$. But this is immediate, since we can pick a realizer $U_i \in \phi$ of $g_i \leq h_i \circ (\overline{\mu_i})_B$ for each $i \in I$, so that $(U_i)_{i \in I} \in \phi$ realizes $g \leq h \circ \overline{\mu}_B$. □

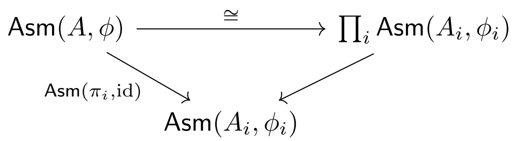
Remark 4.4.2. At various occasions in the proof of Proposition 4.4.1, we needed choice on the index set I, because we needed to pick, for each $i \in I$, an element of ϕ_i that realizes something. If the base categories $\mathcal{C}_i$ are *Heyting* categories, then the use of AC can actually be avoided. Indeed, then we can *define*, for each $i \in I$, the desired element of ϕ_i using first-order logic inside $\mathcal{C}$; cf. Remark 4.3.5(ii).

Now we can state and prove the main result of this section.

Theorem 4.4.3. *The 2-functor* $\mathsf{Asm} \colon \mathsf{IPCA} \to \mathsf{REG}$ *preserves small 2-products, and in particular, categories of the form* $\mathsf{Asm}(A, \phi)$ *are closed under small 2-products of categories. If, for each* $i \in I$*, we have an IPCA* (A_i, ϕ_i) *over* $\mathcal{C}_i$*,* *then the isomorphism* $\mathsf{Asm}\left(\prod_i (A_i, \phi_i)\right) \cong \prod_i \mathsf{Asm}(A_i, \phi_i)$ *makes the triangles in following diagram commute:*



Proof. Let us write $\mathcal{C} = \prod_i \mathcal{C}_i$ and $(A, \phi) = \prod_i (A_i, \phi_i)$, and denote the projection $(A, \phi) \to (A_i, \phi_i)$ by (π_i, id). Now, an object X of $\mathsf{Asm}(A, \phi)$ consists of an object $|X| = (|X|_i)_{i \in I}$ of $\mathcal{C}$, and a subobject $E_X \subseteq |X| \times A$, that is, a subobject $(E_X)_i \subseteq |X|_i \times A_i$ for each $i \in I$. The fact that X is an assembly over (A, ϕ) means precisely that $(|X|_i, (E_X)_i)$ is an assembly over (A_i, ϕ_i), for each i. Moreover, a morphism $X \to Y$ of $\mathsf{Asm}(A, \phi)$ is precisely a family of functions $(f_i \colon |X|_i \to |Y|_i)_{i \in I}$ such that f_i is a morphism of assemblies $(|X|_i, (E_X)_i) \to (|Y|_i, (E_Y)_i)$. Thus, we have an obvious isomorphism $\mathsf{Asm}\left(\prod_i A_i, \phi_i)\right) \cong \prod_i \mathsf{Asm}(A_i, \phi_i)$, and we readily check that it makes the diagram



commute, for each $i \in I$. This establishes the claim that Asm preserves small 2-products. Moreover, from the triangle above one easily deduces that the triangles from the statement of the theorem commute as well. □

Now let us turn to pseudocoproducts. The following proposition generalizes Corollary 2.4.17.

Proposition 4.4.4. *For every regular category* $\mathcal{C}$*, the category* $\mathsf{IPCA}_{\mathcal{C}}$ *has finite pseudocoproducts. Moreover, if* $p \colon \mathcal{C} \to \mathcal{D}$ *is regular, then* $p^* \colon \mathsf{IPCA}_{\mathcal{C}} \to \mathsf{IPCA}_{\mathcal{D}}$. *preserves finite pseudocoproducts.*

Proof. The pseudoinitial object of $\mathsf{IPCA}_{\mathcal{C}}$ is of course $(1, \mathcal{T}1 = \{1\})$. Now suppose that (A, ϕ) and (B, ψ) are IPCAs over $\mathcal{C}$. Then we can make $A \times B$ into an IPAP over $\mathcal{C}$ by defining application and the order coordinatewise. Let us write $\phi \times \psi = \{U \times V \mid U \in \phi, V \in \psi\} \subseteq \mathcal{T}(A \times B)$. Then $(A \times B, \langle \phi \times \psi \rangle)$ is an IPCA, since suitable combinators are given by $\mathsf{K} \times \mathsf{K}$ and $\mathsf{S} \times \mathsf{S}$.

There is an applicative morphism $i \colon (A, \phi) \to (A \times B, \langle \phi \times \psi \rangle)$ given by $i = \{(a, a', b) \in A \times (A \times B) \mid a \geq a'\}$, and similarly, $j \colon (B, \psi) \to (A \times B, \langle \phi \times \psi \rangle)$ given by $j = \{(b, a, b') \in B \times (A \times B) \mid b \geq b'\}$. Now, if $f \colon (A, \phi) \to (C, \chi)$ and $g \colon (B, \psi) \to (C, \chi)$ are applicative morphisms, then their amalgamation $h \colon (A \times B, \langle \phi \times \psi \rangle) \to (C, \chi)$ is given by:

$$h(a, b, c) :\equiv \exists d, d', d'' \colon C \, (\mathsf{P}(d) \wedge f(a, d') \wedge g(b, d'') \wedge c \leq dd'd'').$$

The remainder of the proof that $(A \times B, \langle \phi \times \psi \rangle)$ is the pseudocoproduct of (A, ϕ) and (B, ψ) is similar to the proof of Theorem 2.4.9, and is omitted. Note that, in order to show that h as defined above is an applicative morphism, we need to use Lemma 4.3.18, because $(A \times B, \langle \phi \times \psi \rangle)$ carries a *generated* filter.

For the second statement, we in fact have that $p^*(A \times B, \langle \phi \times \psi \rangle)$ is *equal* to the pseudocoproduct of $p^*(A, \phi)$ and $p^*(B, \psi)$. The only nontrivial part is showing that both are equipped with the same filter, i.e., that $\langle p(\langle \phi \times \psi \rangle) \rangle = \langle \langle p(\phi) \rangle \times \langle p(\psi) \rangle \rangle$. By Lemma 4.3.19(ii), it suffices to show that $\langle p(\phi \times \psi) \rangle = \langle \langle p(\phi) \rangle \times \langle p(\psi) \rangle \rangle$. In one direction, since p is left exact, we have:

$$p(\phi \times \psi) = p(\phi) \times p(\psi) \subseteq \langle p(\phi) \rangle \times \langle p(\psi) \rangle \subseteq \langle \langle p(\phi) \rangle \times \langle p(\psi) \rangle \rangle,$$

which yields $\langle p(\phi \times \psi) \rangle \subseteq \langle \langle p(\phi) \rangle \times \langle p(\psi) \rangle \rangle$. For the other direction, suppose we have $U \in \langle p(\phi) \rangle$ and $V \in \langle p(\psi) \rangle$. By Corollary 4.3.29, there exist $U' \in \phi$ and $V' \in \psi$ such that $p(U') \cdot p(A) \subseteq U$ and $p(V') \cdot p(B) \subseteq V$. Now we see that

$$p(U' \times V') \cdot p(A \times B) = (p(U') \cdot p(A)) \times (p(V') \cdot p(B)) \subseteq U \times V,$$

which implies $U \times V \in \langle p(\phi \times \psi) \rangle$. Thus, we can conclude that $\langle p(\phi) \rangle \times \langle p(\psi) \rangle \subseteq \langle p(\phi \times \psi) \rangle$, hence also that $\langle \langle p(\phi) \rangle \times \langle p(\psi) \rangle \rangle \subseteq \langle p(\phi \times \psi) \rangle$. ☐

For an opfibration between 1-categories, we have the following result: if the base has finite coproducts, all the fibers have finite coproducts, and the reindexing functors preserve finite coproducts, then the total category has finite coproducts as well. Since REG has finite pseudocoproducts, Proposition 4.4.4 seems to be the kind of result we need to conclude that IPCA has finite pseudocoproducts as well. However, $\mathsf{IPCA} \to \mathsf{REG}$ is not an opfibration in a 2-categorical sense, since 2-cells $\mu \colon p \Rightarrow q$ of REG fail to lift to *arrows* $p^*(A, \phi) \to q^*(A, \phi)$. On the other hand, if μ is a natural *isomorphism*, then the functors $p^*, q^* \colon \mathsf{IPCA}_{\mathcal{C}} \to \mathsf{IPCA}_{\mathcal{D}}$ will clearly be isomorphic as well. This allows us to obtain a partial result concerning coproducts, in the following way. Given a bicategory $\mathcal{B}$, we can make it into a 1-category $\mathcal{B}|_1$ in the following way. The objects of $\mathcal{B}|_1$ are simply the objects of $\mathcal{B}$, but the arrows of $\mathcal{B}|_1$ are *isomorphism classes* of arrows of $\mathcal{B}$. Proposition 4.4.4 suffices to ensure the existence of coproducts in $\mathsf{IPCA}|_1$, and in fact, they are biproducts.

Corollary 4.4.5. *The 1-category* $\mathsf{IPCA}|_1$ *has finite biproducts.*

Proof. The zero object of $\mathsf{IPCA}|_1$ is the unique IPCA over the terminal category $\mathbf{1}$. Now suppose that (A_0, ϕ_0) and (A_1, ϕ_1) are IPCAs over $\mathcal{C}_0$ resp. $\mathcal{C}_1$. We have the functor $\kappa_0 \colon \mathcal{C}_0 \to \mathcal{C}_0 \times \mathcal{C}_1$ given by $\kappa_0 X = (X, 1)$, and similarly, we have a functor $\kappa_1 \colon \mathcal{C}_1 \to \mathcal{C}_0 \times \mathcal{C}_1$. Note that $\kappa_0^*(A_0, \phi_0) = ((A_0, 1), \{(U, \{1\}) \mid U \in \phi\})$, and similarly for $\kappa_1^*(A_1, \phi_1)$. Now it is easy to check that the pseudocoproduct of $\kappa_0^*(A_0, \phi_0)$ and $\kappa_1^*(A_1, \phi_1)$ in $\mathsf{IPCA}_{\mathcal{C} \times \mathcal{D}}$ is isomorphic to the 2-product $(A_0, \phi_0) \times (A_1, \phi_1) = ((A_0, A_1), \{(U_0, U_1) \mid U_i \in \phi_i\})$. The pseudocoproduct inclusion $i_0 \colon \kappa_0^*(A_0, \phi_0) \to (A_0, \phi_0) \times (A_1, \phi_1)$ is given by $i_0 = (\{(a, a') \mid a \geq a'\}, 1 \times A_1) \subseteq (A_0 \times A_0, 1 \times A_1)$, and similarly for i_1. Now it is easy to show that

$$(A_0, \phi_0) \xrightarrow{(\kappa_0, i_0)} (A_0, \phi_0) \times (A_1, \phi_1) \xleftarrow{(\kappa_1, i_1)} (A_1, \phi_1)$$

yields a coproduct diagram in $\mathsf{IPCA}|_1$. Moreover, denoting the 2-product projection $(A_0, \phi_0) \times (A_1, \phi_1) \to (A_1, \phi_1)$ by (π_i, id), we easily compute that $(\pi_0, \mathrm{id}) \circ (\kappa_0, i_0) = \mathrm{id}_{(A_0, \phi_0)}$, whereas $(\pi_1, \mathrm{id}) \circ (\kappa_0, i_0)$ is a zero morphism. This completes the proof. □

4.4.2 Slicing categories of assemblies

In this section, we show that categories of the form $\mathsf{Asm}(A, \phi)$ are closed under slicing. This result is already obtained in [Ste13, Corollary 2.2.18], but in an 'indirect' way. More precisely, [Ste13] first offers a characterization of categories of the form $\mathsf{Asm}(A, \phi)$ (Theorem 2.2.17), and then shows that categories satisfying this characterization are closed under slicing. Thus, the proof does not describe the 'underlying' IPCA of a slice category $\mathsf{Asm}(A, \phi)/I$, where I is an assembly over (A, ϕ). The main additional contribution of this section is that we do offer an explicit description of the underlying IPCA of $\mathsf{Asm}(A, \phi)/I$. More precisely, we show that the underlying IPCA of $\mathsf{Asm}(A, \phi)/I$ arises from combining base change (Construction 4.3.21) with the notion of a finitely generated filter (Definition 4.3.11). This explicit description will have two main applications. First, we will be able to give a simple presentation of slice categories of the form $\mathsf{Asm}(A)/I$, where A is a PCA. Moreover, in the next section, we will see that there is a connection between slicing and the notion of computational density.

Before we turn to slicing, let us see that, in the current setup, the terminal object of $\mathsf{Asm}(A, \phi)$ is not necessarily projective.

Proposition 4.4.6. *Let (A, ϕ) be an IPCA over $\mathcal{C}$. Then $1 \in \mathsf{Asm}(A, \phi)$ is projective iff $1 \in \mathcal{C}$ is projective and ϕ is generated by singletons.*

Proof. First, suppose that $1 \in \mathsf{Asm}(A, \phi)$ is projective. Since $\Gamma\colon \mathsf{Asm}(A, \phi) \to \mathcal{C}$ has a regular right adjoint, we know that Γ preserves projectives (see Remark 3.2.17), so $\Gamma 1 \cong 1 \in \mathcal{C}$ is projective as well. Now suppose we have a $U \in \phi$, and define the prone subobject R_U of the object of realizers R_A with $|R_U| = U \subseteq A$. Then $R_U \to 1$ is a regular epimorphism, which means that R_U has a global section $a\colon 1 \to R_U$. In particular, a is a global section $1 \to A$ such that $\vDash U(a)$. Moreover, the fact that $a\colon 1 \to R_U$ is a morphism of assemblies implies that $\delta_A(a) \in \phi$, as desired.

For the converse, suppose that $1 \in \mathcal{C}$ is projective and that ϕ is generated by singletons. First of all, we observe that there must exist a global section $a_0\colon 1 \to A$ such that $\delta_A(a_0) \in \phi$. Now we can describe the terminal object of $\mathsf{Asm}(A, \phi)$ as the assembly 1 with $|1| = 1$ and $E_1 = \delta_A(a_0) \subseteq A \cong 1 \times A$. In order to show that this assembly 1 is projective, suppose we have a regular epimorphism $X \twoheadrightarrow 1$. Then there exists a $U \in \phi$ such that:

$$U(r) \wedge a \leq a_0 \vDash_{r, a : A} ra{\downarrow} \wedge \exists x{:}|X|(E_X(x, ra))).$$

By our assumtion, there exists a global section $a_1\colon 1 \to A$ such that $\vDash U(a_1)$ and $\delta(a_1) \in \phi$. We can conclude that $a_1 a_0{\downarrow}$, and $\vDash \exists x{:}|X|(E_X(x, a_1 a_0))$. Since

$1 \in \mathcal{C}$ is assumed to be projective, this implies that there exists a global section $x_0 \colon 1 \to |X|$ such that $\vDash E_X(x_0, a_1 a_0)$, and $\delta_A(a_1) \in \phi$ tracks x_0 as a morphism $1 \to X$, as desired. □

Example 4.4.7. In Example 4.1.2, we defined an IPCA (A, ϕ) over Set such that $1 \in \mathsf{RT}(A, \phi)$ is not projective. The same argument shows that $1 \in \mathsf{Asm}(A, \phi)$ is not projective, which means that (A, ϕ) is not generated by singletons.

If (A, ϕ) is an IPCA over $\mathcal{C}$ and I is an assembly, then our goal is to show that $\mathsf{Asm}(A, \phi)/I$ is equivalent to the category of assemblies for an IPCA over $\mathcal{C}/|I|$. Therefore, we first make some general remarks on IPCAs internal to a slice of $\mathcal{C}$.

Thus, let (A, ϕ) be an IPCA over $\mathcal{C}$ and let J be an object of $\mathcal{C}$. We consider the regular category $\mathcal{C}/J$ and the pullback functor $J^* \colon \mathcal{C} \to \mathcal{C}/J$ sending X to $J \times X \xrightarrow{\pi_0} J$. This pullback functor is regular, so $J^*(A)$ is an IPAP over $\mathcal{C}/J$. Note that $J^*(A) \times_J J^*(A)$ is simply $J \times A \times A \to J$. The order on $J^*(A)$ is given by $\{(j, a, a') \in J \times A \times A \mid a \leq a'\}$, the domain of the application map is $J \times D \subseteq J \times A \times A$, and the application map $J \times D \to J \times A$ is the product of id_J with the application map $D \to A$. We will spell out what an IPCA $(J^*(A), \psi)$ over $\mathcal{C}/J$ is in terms of the internal logic of $\mathcal{C}$ (rather than $\mathcal{C}/J$).

First, let us describe $\mathcal{T}(J^*(A))$. Since an arrow is mono in $\mathcal{C}/J$ if and only if it is mono in $\mathcal{C}$, we see that the subobjects of $J^*(A)$ in $\mathcal{C}/J$ are the subobjects of $J \times A$ in $\mathcal{C}$. A subobject $U \subseteq J \times A$ is inhabited in $\mathcal{C}/J$ if and only if it is fiberwise inhabited in $\mathcal{C}$, i.e., $\vDash_{j:J} \exists a : A(U(j, a))$. Moreover, U is downwards closed iff is if fiberwise downwards closed, i.e., $U(j, a) \wedge a' \leq a \vDash_{j:J; a, a':A} U(j, a')$. The application of two such subobjects $U, V \subseteq J \times A$ is defined iff $U \times_J V \subseteq J \times D$, and in this case, UV is the image of the map $U \times_J V \subseteq J \times D \to J \times A$. In other words,

$$UV = \{(j, a) \in J \times A \mid \exists b, c : A(U(j, b) \wedge V(j, c) \wedge a \leq bc)\}.$$

A filter ψ on $J^*(A)$ is then a subset of $\mathcal{T}(J^*(A))$ that is upwards closed and closed under the application displayed above.

Suppose that an IPCA $(J^*(A), \psi)$ over $\mathcal{C}/J$ is given. An assembly X over this IPCA is an object $|X| \xrightarrow{k_X} J$ of $\mathcal{C}/J$, together with a subobject $E_X \subseteq |X| \times_J J^*(A)$ satisfying the two conditions from Definition 4.3.30(i). But note that $|X| \times_J J^*(A) = |X| \times A$, and that the requirements on E_X simply say that $(|X|, E_X)$ is an assembly over (A, ϕ).

Now consider another assembly Y, and suppose that $f \colon |X| \to |Y|$ is an arrow of $\mathcal{C}/J$, that is, $k_Y \circ f = k_X$. Then $U \subseteq J \times A$ tracks $f \colon X \to Y$ iff

$$U(k_X(x), r) \wedge E_X(x, a) \vDash_{x:|X|; r, a:A} ra{\downarrow} \wedge E_Y(f(x), ra).$$

In other words, if $U(j, r)$, then r should track f as if it were a morphism $(|X|, E_X) \to (|Y|, E_Y)$ in $\mathsf{Asm}(A, \phi)$, but only for those $x \in |X|$ that lie in the fiber of j.

Now let I be an assembly over $\mathsf{Asm}(A, \phi)$. Then $E_I \subseteq |I| \times A$ is by definition fiberwise (over $|I|$) inhabited and downwards closed, that is, $E_I \in \mathcal{T}(|I|^*(A))$. This is the key insight behind the the following definition. This definition uses base change along the functor $|I|^* \colon \mathcal{C} \to \mathcal{C}/|I|$. The resulting 2-functor $\mathsf{IPCA}_{\mathcal{C}} \to \mathsf{IPCA}_{\mathcal{C}/|I|}$ should be denoted by $(|I|^*)^*$, which is slightly unfortunate. Therefore, we denote this 2-functor using the slightly longer notation $\mathsf{IPCA}_{|I|^*} \colon \mathsf{IPCA}_{\mathcal{C}} \to \mathsf{IPCA}_{\mathcal{C}/|I|}$.

Definition 4.4.8. *Let (A, ϕ) be an IPCA over $\mathcal{C}$ and let I be an assembly over (A, ϕ). We define the IPCA $(A, \phi)/I$ over $\mathcal{C}/|I|$ as:*

$$\big(\mathsf{IPCA}_{|I|^*}(A, \phi)\big)\,[E_I] = (|I|^*(A), \langle |I|^*(\phi) \cup \{E_I\}\rangle).$$

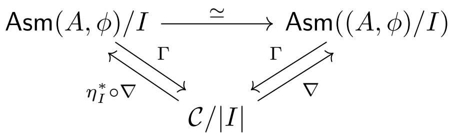
We will denote the external filter $\langle |I|^*(\phi) \cup \{E_I\}\rangle$ on $|I|^*(A)$ by ϕ_I, so that $(A, \phi)/I = (|I|^*(A), \phi_I)$.

By Lemma 4.3.28, we have:

$$\phi_I = {\uparrow}\{|I|^*(U) \cdot E_I \mid U \in \phi \text{ and } |I|^*(U) \cdot E_I{\downarrow}\}. \tag{4.7}$$

Now we are ready to prove the main result on slicing.

Theorem 4.4.9. *Let (A, ϕ) be an IPCA over $\mathcal{C}$, and let I be an assembly over (A, ϕ). Then there exists an equivalence of categories $\mathsf{Asm}(A, \phi)/I \simeq \mathsf{Asm}((A, \phi)/I)$ that makes the triangles in the following diagram commute (up to isomorphism):*



In particular, categories of the form $\mathsf{Asm}(A, \phi)$ are closed under slicing.

Proof. When working with $\mathsf{Asm}((A, \phi)/I)$, we employ the notation introduced above, with $J = |I|$. That is, an assembly over $(A, \phi)/I$ will be denoted as $(k_X \colon |X| \to |I|, E_X)$, where $(|X|, E_X)$ is an assembly over (A, ϕ). Moreover, we will denote objects of $\mathsf{Asm}(A, \phi)/I$ by $\ell_X \colon X \to I$, where X is an assembly over (A, ϕ) and ℓ_X is a morphism of assemblies.

We define the desired equivalence $F \colon \mathsf{Asm}(A, \phi)/I \to \mathsf{Asm}((A, \phi)/I)$ as follows. Given $\ell_X \colon X \to I$, we define $FX \in \mathsf{Asm}((A, \phi)/I)$ simply by $|FX| = |X|$, $k_{FX} = \ell_X$ and $E_{FX} = E_X$, and on morphisms f, we set $F(f) = f$. In order to see that f is well-defined, we need to check: if $f \colon X \to Y$ is a morphism of assemblies over (A, ϕ) such that $\ell_Y \circ f = \ell_X$, then f is also a morphism of assemblies $FX \to FY$ over $(A, \phi)/I$. But this is easy, since if $T \in \phi$ tracks $f \colon X \to Y$, then $|I|^*(T) \in |I|^*(\phi) \subseteq \phi_I$ tracks $f \colon FX \to FY$.

In order to see that F is full, we should check the converse, that is: if $f \colon |X| \to |Y|$ is an arrow such that $\ell_Y \circ f = \ell_X$ and f is a morphism of assemblies

$FX \to FY$ over $\mathsf{Asm}(A,\phi)/I$, then f is already a morphism of assemblies $X \to Y$ over (A,ϕ). Let $T \in \phi_I$ track $f\colon FX \to FY$, that is:

$$T(\ell_X(x),t) \wedge E_X(x,a) \vDash_{x:|X|;t,a:A} ta{\downarrow} \wedge E_Y(f(x),ta).$$

By (4.7), there exists a $T' \in \phi$ such that $|I|^*(T') \cdot E_I \subseteq T$, which is to say:

$$T'(t') \wedge E_I(i,a) \vDash_{i:|I|;t',a:A} t'a{\downarrow} \wedge T(i,t'a).$$

Now let $U \in \phi$ be a tracker of $\ell_X\colon X \to I$, and let $V \in \phi$ be a realizer of $\lambda^* x.T'(Ux)x$. We claim that V tracks $f\colon X \to Y$. By definition, we have $V \subseteq WT'U$ for some realizer W of $\lambda^* yzx.y(zx)x$. Now reason inside $\mathcal{C}$, and suppose that $V(r)$ and $E_I(x,a)$. Then there exist $r',t',u\colon A$ such that $W(r')$, $T'(t')$, $U(u)$ and $r \leq r't'u$. We see that ua is defined and $E_I(\ell_X(x),ua)$. This implies that $t'(ua)$ is defined as well, and $T(\ell_X(x),t'(ua))$. This implies that $t'(ua)a$ is defined as well, and $E_Y(f(x),t'(ua)a)$. Finally, we see that $r't'ua$ and ra are defined as well, that $ra \leq r't'ua \leq t'(ua)a$, and thus $E_Y(f(x),ra)$, as desired.

Clearly, F is faithful, so in order to show that F is an equivalence, it remains to establish essential surjectivity. Let $X = (k_X\colon |X| \to |I|, E_X)$ be an assembly over $(A,\phi)/I$. We define the morphism of assemblies $\ell_{X'}\colon X' \to I$ by $|X'| = |X|$, $\ell_{X'} = k_X$, but:

$$E_{X'}(x,a) :\equiv \exists b,c,d\colon A\,(\mathsf{P}(b) \wedge E_I(k_X(x),c) \wedge E_X(x,d) \wedge a \leq bcd).$$

Note that $\ell_{X'}$ is indeed a morphism $X' \to I$, for it is tracked by P_0. In order to see that $FX' \cong X$, we note that $|I|^*(\mathsf{P})\cdot E_I \in \phi_I$ tracks $\mathrm{id}_{|X|}$ as a morphism $X \to F'X$, whereas $|I|^*(\mathsf{P}_1) \in \phi_I$ tracks $\mathrm{id}_{|X|}$ as a morphism $F'X \to X$.

From the definition of F, it is clear that F commutes (strictly) with the Γ-functors. But now, since F is an equivalence, it easily follows that

$$\mathcal{C}/|I| \xrightarrow{\eta_I^* \circ \nabla} \mathsf{Asm}(A,\phi)/I \xrightarrow{F} \mathsf{Asm}((A,\phi)/I)$$

is right adjoint to $\Gamma\colon \mathsf{Asm}((A,\phi)/I) \to \mathcal{C}/|I|$, so we get the other triangle as well. □

Remark 4.4.10. In the proof of the essential surjectivity of F above, we cannot simply take $X' = (|X|, E_X)$, since k_X may fail to be a morphism of assemblies $(|X|, E_X) \to I$. On the other hand, k_X is always a morphism $(|X|,E_X) \to \nabla|I|$, and X' is obtained via the pullback square:

$$\begin{array}{ccc} X' & \longrightarrow & (|X|, E_X) \\ \downarrow & & \downarrow \\ I & \xrightarrow{\eta_I} & \nabla|I| \end{array}$$

Note that the replete image of $\eta_I^* \circ \nabla\colon \mathcal{C}/|I| \to \mathsf{Asm}(A,\phi)/I$ consists precisely of the prone morphisms of $\mathsf{Asm}(A,\phi)$ with codomain I. Thus, we see that the constant objects in $\mathsf{Asm}(A,\phi)/I \simeq \mathsf{Asm}((A,\phi)/I)$ are precisely the prone arrows with codomain I.

Example 4.4.11. Let (A, ϕ) be an IPCA over $\mathcal{C}$ and let $f\colon I \to J$ be a morphism of assemblies. The pullback functor $f^*\colon \mathsf{Asm}(A,\phi)/J \to \mathsf{Asm}(A,\phi)/I$ sends prone arrows with codomain J to prone arrows with codomain I. Under the equivalences of Theorem 4.4.9, we get a functor $F\colon \mathsf{Asm}((A,\phi)/J) \to \mathsf{Asm}((A,\phi)/I)$ that preserves constant objects. Now we easily see that both squares in

$$\begin{array}{ccc} \mathsf{Asm}((A,\phi)/J) & \xrightarrow{F} & \mathsf{Asm}((A,\phi)/I) \\ \nabla\uparrow\downarrow\Gamma & & \Gamma\downarrow\uparrow\nabla \\ \mathcal{C}/|J| & \xrightarrow{f^*} & \mathcal{C}/|I| \end{array}$$

commute. Since F is also regular, we must have $F \cong \mathsf{Asm}(f^*, g)$ for some applicative morphism $(f^*, g)\colon (A,\phi)/J \to (A,\phi)/I$. Note that $\mathsf{IPCA}_{f^*}((A,\phi)/J)$ is $f^*(|J|^*A) = |I|^*(A)$, equipped with the external filter:

$$\langle f^*(\phi_J)\rangle = \langle f^*(|J|^*(\phi) \cup \{E_J\})\rangle = \langle |I|^*(\phi) \cup \{f^*(E_J)\}\rangle,$$

where we used Lemma 4.3.19(ii). (Note that we also wrote IPCA_{f^*} to avoid writing $(f^*)^*$.) If $T \in \phi$, then T tracks $f\colon I \to J$ if and only if $|I|^*(T) \cdot E_I \subseteq f^*(E_J)$ holds in $\mathcal{T}(|I|^*(A))$. Thus, we see that the fact that f is a morphism $I \to J$ means precisely that $f^*(E_J) \in \phi_I$, which is equivalent to saying that $\langle f^*(\phi_J)\rangle \subseteq \phi_I$. In particular, ϕ_I is a finite extension of $\langle f^*(\phi_J)\rangle$ by means of the element $E_I \in \mathcal{T}(|I|^*(A))$. If we let $g = \iota_{E_I}$ as in Example 4.3.17, then a direct calculation shows that $\mathsf{Asm}(f^*, g) \cong F$.

In particular, if we take $J = 1$, then we get the applicative morphism $(|I|^*, \iota_{E_I})\colon (A,\phi) \to (A,\phi)/I$, such that

$$\mathsf{Asm}(A,\phi) \xrightarrow{\mathsf{Asm}(|I|^*, \iota_{E_I})} \mathsf{Asm}((A,\phi)/I) \simeq \mathsf{Asm}(A,\phi)/I$$

is isomorphic to the functor that pulls back along I.

Example 4.4.12. Suppose that the IPCA (A,ϕ) over $\mathcal{C}$ is generated by singletons (Definition 4.3.9). Consider a *partitioned* assembly I over $\mathsf{Asm}(A,\phi)$, that is, we suppose there exists an arrow $e_I\colon |I| \to A$ such that $E_I(i,a) :\equiv a \le e_I(i)$. In [Ste13], it is shown that, when ϕ is generated by singletons, the projectives in $\mathsf{Asm}(A,\phi)$ are reasonably well-behalved. More precisely, an assembly X is projective in $\mathsf{Asm}(A,\phi)$ iff X is partitioned and $|X|$ is projective in $\mathcal{C}$.

Note that global sections of $|I|^*(A)$ in the slice category $\mathcal{C}/|I|$ are simply functions $|I| \to A$. If $a\colon 1 \to A$ is a global section, then $|I|^*(\delta_A(a)) = \delta_{|I|^*(A)}(|I|^*(a))$, and $|I|^*(a)$ is the global section $|I| \to 1 \xrightarrow{a} A$ of $|I|^*A$. The object E_I, viewed as an element of $\mathcal{T}(|I|^*(A))$, is equal to $\delta_{|I|^*(A)}(e_I)$. Thus, if ϕ is generated by $\{\delta_A(a) \mid a \in C\}$ for some $C \subseteq \mathrm{Hom}(1,A)$, then (using Lemma 4.3.19(ii)) we see that ϕ_I is generated by $\{\delta_{|I|^*(A)}(f) \mid f \in D\}$, where:

$$D = \{|I| \to 1 \xrightarrow{a} A \mid a \in C\} \cup \{e_I\} \subseteq \mathrm{Hom}(|I|, A).$$

In fact, we can be a bit more explicit here. As we saw in the proof of Proposition 4.3.10, we may take $C' = \{a \in \mathrm{Hom}(1,A) \mid \delta_A(a) \in \phi\}$, so that

$\phi = \uparrow\{\delta_A(a) \mid a \in C'\}$. Now $U \subseteq |I| \times A$ is in ϕ_I iff there exists a $V \in \phi$ such that $|I|^*(V) \cdot E_I \subseteq U$, which holds iff there exists an $a \in C'$ such that $|I|^*(\delta_A(a)) \cdot E_I \subseteq U$. In the internal logic of $\mathcal{C}$, this is equivalent to $\vDash_{i:|I|} a \cdot e_I(i)\downarrow \wedge U(i, a \cdot e_I(i))$. Thus, we see that $\phi_I = \uparrow\{\delta_{|I|^*(A)}(f) \mid f \in D'\}$, where:

$$D' = \{f \in \mathrm{Hom}(|I|, A) \mid \exists a \in C' \,(\vDash_{i:|I|} a \cdot e_I(i)\downarrow \wedge\, a \cdot e_I(i) \leq f(i))\}.$$

Example 4.4.13. Let A be a PCA in the sense of Chapter 2. Then we can view A as the IPCA $(A, (TA)^\#)$ over Set, whose external filter is generated by singletons. Now let I be an assembly over A. First of all, let us note that $\mathsf{Set}/|I|$ is equivalent to $\mathsf{Set}^{|I|}$ and that, modulo this equivalence, the pullback functor $|I|^*$ is simply the diagonal functor $\mathsf{Set} \to \mathsf{Set}^{|I|}$. Applying base change along this functor, we obtain the IPAP $(A)_{i\in|I|}$ over $\mathsf{Set}^{|I|}$. An object of $\mathcal{T}((A)_{i\in|I|})$ is simply a sequence $(U_i)_{i\in|I|}$, where each U_i is in TA, and a global section of $(A)_{i\in|I|}$ is simply a sequence $(a_i)_{i\in|I|}$, where each a_i is in A. Now $\mathsf{Asm}(A)/I$ is equivalent to $\mathsf{Asm}((A)_{i\in|I|}, \phi_I)$, where:

$$\phi_I = \{(U_i)_{i\in|I|} : \exists r \in A^\# \,\forall i \in |I|\,(r \cdot E_I(i)\downarrow \wedge\, r \cdot E_I(i) \subseteq U_i)\}.$$

Now suppose that I is partitioned, so that $E_I(i) = \delta_A(e_I(i))$ for some function $e_I \colon |I| \to A$. Then by the previous example, ϕ_I is also generated by singletons. Indeed, a suitable generating set $C \subseteq A^{|I|}$ is $\{(r)_{i\in I} \mid r \in A^\#\} \cup \{e_I\}$, and we can also describe ϕ_I as:

$$\{(U_i)_{i\in|I|} : \exists r \in A^\# \,\forall i \in |I|\,(r \cdot e_I(i)\downarrow \wedge\, r \cdot e_I(i) \in U_i)\}.$$

It follows that $\mathsf{Asm}(A)/I$ may be described as follows. Its objects are families $X = (X_i)_{i\in|I|}$, where each X_i is an assembly over A. An arrow $X \to Y$ is a family of morphisms of assemblies $f_i \colon X_i \to Y_i$ for which there exists an $r \in A^\#$ such that $r \cdot e_I(i)$ tracks f_i, for all $i \in |I|$.

Example 4.4.14. Again, let A be a PCA in the sense of Chapter 2, and let I be a subterminal assembly over $(A, (TA)^\#)$ such that I is nonempty, i.e., $I \not\cong 0$. Then $|I| = 1$, which means that $\mathsf{Asm}(A)/I$ is equivalent to the category of assemblies for an IPCA over Set again. We can write I as 1_X, where $|1_X| = \{*\}$ and $E_{1_X}(*) = X \in TA$. Then by Theorem 4.4.9, $\mathsf{Asm}(A)/I$ is equivalent to $\mathsf{Asm}((A, (TA)^\#)[X]) = \mathsf{Asm}(A, \langle (TA)^\# \cup \{X\}\rangle)$. Thus, slices of $\mathsf{Asm}(A)$ over nonempty subterminals correspond to finite extensions of the filter $(TA)^\# \subseteq TA$.

Now suppose that I is projective. Then I must be isomorphic to a certain *partitioned* nonempty subterminal 1_a, where $|1_a| = \{*\}$ and $E_{1_a}(*) = \downarrow\{a\}$. Then $\mathsf{Asm}(A)/I$ is equivalent to the category of assemblies for an IPCA over Set which is generated by singletons; that is, a PCA in the sense of Chapter 2. Its underlying PAP is A itself, and by Example 4.4.13 above, its filter is generated by $A^\# \cup \{a\}$. We have seen this PCA before: it is the PCA $A[a]$ from Example 2.2.15. Thus, we can conclude that $\mathsf{Asm}(A)/1_a \simeq \mathsf{Asm}(A[a])$, and we see: slices of $\mathsf{Asm}(A)$ over *projective* nonempty subterminals correspond to finite extensions of the filter $A^\# \subseteq A$.

Example 4.4.15. Let A be a PCA in the sense of Chapter 2, and consider a constant object ∇I, where I is a set. Then $\mathsf{Asm}(A)/\nabla I \simeq \mathsf{Asm}((A)_{i\in I}, \phi_{\nabla I})$, where:

$$\phi_{\nabla I} = \left\{(U_i)_{i\in I} \mid A^{\#} \cap \bigcap_{i\in I} U_i \neq \emptyset\right\}$$

Thus, we see that a morphism $X \to Y$ of $\mathsf{Asm}(A)/\nabla I$ is a family of morphisms of assemblies $f_i \colon X_i \to Y_i$ for which there exists an $r \in A^{\#}$ that tracks all the f_i *simultaneously*.

Now let us, for simplicity, assume that $I = 2$, so that $\mathsf{Asm}(A)/\nabla 2 \simeq \mathsf{Asm}((A, A), \phi_{\nabla 2})$, where $\phi_{\nabla 2} = \{(U, V) \in TA \times TA \mid A^{\#} \cap U \cap V \neq \emptyset\}$. We see that, even if A is an absolute PCA, that is, $A = A^{\#}$, the IPCA $((A, A), \phi_{\nabla 2})$ over Set^2 will not be an absolute IPCA as well, provided that A is not semitrivial; cf. Example 4.1.1.

Moreover, while we know that $\phi_{\nabla 2}$ is generated by singletons, it does not correspond to a *subobject* of (A, A) in Set^2 in any obvious way. Thus, while we originally introduced the notion of an external filter to allow for slicing over non-projective assemblies, we see that it is actually required to accommodate slices over projective assemblies as well.

Example 4.4.16. Let A be a PCA in the sense of Chapter 2, and consider $1+1$, which is the projective assembly defined by $|1 + 1| = \{0, 1\}$, $E_{1+1}(0) = \downarrow\{\top\}$ and $E_{1+1}(1) = \downarrow\{\bot\}$. Then $\mathsf{Asm}(A)/(1 + 1) \simeq \mathsf{Asm}((A, A), \phi_{1+1})$, where ϕ_{1+1} consists of all (U, V) for which there exists an $r \in A^{\#}$ such that $r\top \in U$ and $r\bot \in V$. But this is just to say that U and V both contain an element from $A^{\#}$, so we see that $((A, A), \phi_{1+1})$ is the 2-product, in IPCA, of A with itself. In particular, we see that $\mathsf{Asm}(A)/(1 + 1) \simeq \mathsf{Asm}(A)^2$, which we could also have derived from the fact that $\mathsf{Asm}(A)$ is a quasitopos.

Example 4.4.17. We consider a few examples of slicing over the natural numbers object N of $\mathsf{Asm}(A)$.

(i) First, consider Kleene's first model $\mathcal{K}_1$, and the natural numbers object $N \in \mathsf{Asm}(\mathcal{K}_1)$ given by $|N| = \mathbb{N}$ and $E_N(n) = \{n\}$. Now we see that the arrows of $\mathsf{Asm}(\mathcal{K}_1)/N \simeq \mathsf{Asm}(\mathcal{K}_1/N)$ are sequences of morphisms of assemblies $f_n \colon X_n \to Y_n$ over $\mathcal{K}_1$ for which there exists a total recursive function g such that $g(n)$ tracks f_n, for all $n \in \mathbb{N}$.

(ii) Second, consider the relative version of Kleene's second model $\mathcal{K}_2 = (\mathcal{K}_2, \mathcal{K}_2^{\mathrm{rec}}, \cdot, =)$. The natural numbers object N of $\mathsf{Asm}(\mathcal{K}_2)$ is given by $|N| = \mathbb{N}$ and $E_N(n) = \{\hat{n}\}$, where $\hat{n}$ denotes the constant function with value n. Thus, the arrows of $\mathsf{Asm}(\mathcal{K}_2)/N \simeq \mathsf{Asm}(\mathcal{K}_2/N)$ are sequences of morphisms of assemblies $f_n \colon X_n \to Y_n$ over $\mathcal{K}_2$ for which there exists a total recursive function g such that $g \cdot \hat{n}$ tracks f_n, for all $n \in \mathbb{N}$. In general, if $h \colon \mathbb{N}^2 \to \mathbb{N}$, then there exists a total recursive g such that $(g \cdot \hat{n})(m) = h(n, m)$ iff h itself is recursive. Thus, we can also say that there should exist a total recursive function $h \colon \mathbb{N}^2 \to \mathbb{N}$ such that $h(n, -)$ tracks f_n, for all $n \in \mathbb{N}$.

We also have the absolute version $(\mathcal{K}_2)_{\text{abs}}$ of Kleene's second model. The arrows of $\mathsf{Asm}((\mathcal{K}_2)_{\text{abs}})/N \simeq \mathsf{Asm}((\mathcal{K}_2)_{\text{abs}}/N)$ are now sequences of morphisms of assemblies $f_n : X_n \to Y_n$ over $\mathcal{K}_2$ for which there exists *any* function g such that $g \cdot \hat{n}$ tracks f_n, for all $n \in \mathbb{N}$. But this is just to say that each f_n has a tracker, so we see that $\mathsf{Asm}((\mathcal{K}_2)_{\text{abs}})/N \simeq \mathsf{Asm}((\mathcal{K}_2)_{\text{abs}})^{\mathbb{N}}$. In fact, in $\mathsf{Asm}((\mathcal{K}_2)_{\text{abs}})$, the natural numbers object N is isomorphic to the countable coproduct of copies of 1.

(iii) Finally, we consider the relative version of Scott's graph model $\mathcal{P}\omega = (\mathcal{P}\omega, (\mathcal{P}\omega)^{\text{re}}, \cdot, =)$. The natural numbers object N of $\mathsf{Asm}(\mathcal{P}\omega)$ is given by $|N| = \mathbb{N}$ and $E_N(n) = \{\{n\}\}$. Thus, the arrows of $\mathsf{Asm}(\mathcal{P}\omega)/N \simeq \mathsf{Asm}(\mathcal{P}\omega/N)$ are sequences of morphisms of assemblies $f_n : X_n \to Y_n$ over $\mathcal{P}\omega$ for which there exists an r.e. set A such that $A \cdot \{n\}$ tracks f_n, for all $n \in \mathbb{N}$. For a set $B \in \mathcal{P}\omega$, let us write $B_n = \{m \mid \langle n, m\rangle \in B\}$. Now it is easy to show that there exists an r.e. set A such that $A \cdot \{n\} = B_n$ for all n iff B itself is r.e. Thus, we can also say that there should exist an r.e. set B such that B_n tracks f_n, for all $n \in \mathbb{N}$.

If we do *not* require that A above is r.e., then for *any* set B, there exists an A such that $A \cdot \{n\} = B_n$. From this, we can deduce that $\mathsf{Asm}((\mathcal{P}\omega)_{\text{abs}})/N \simeq \mathsf{Asm}((\mathcal{P}\omega)_{\text{abs}})^{\mathbb{N}}$, and indeed, inside $\mathsf{Asm}((\mathcal{P}\omega)_{\text{abs}})$ we have $N \cong \bigsqcup_{n\in\mathbb{N}} 1$.

Example 4.4.18. We consider an example of slicing over a *non-partitioned* assembly. Consider Kleene's first model $\mathcal{K}_1$ and define the assembly Σ by $|\Sigma| = \{0, 1\}$, $E_\Sigma(0) = \{n \mid \varphi_n(n)\downarrow\}$ and $E_\Sigma(1) = \mathbb{N} - E_\Sigma(0) = \{n \mid \varphi_n(n)\uparrow\}$. This assembly is also known as the *r.e. subobject classifier*; see, e.g., [vO08, Section 3.2.7]. It is easy to show that this assembly cannot be isomorphic to a partitioned assembly. We will show an equivalent fact, namely that ϕ_Σ is not generated by singletons. Note that ϕ_Σ consists of all pairs (U, V) of sets of natural numbers for which there exists a (total) recursive function f satisfying: if $n \in E_\Sigma(0)$, then $f(n) \in U$, whereas if $n \in E_\Sigma(1)$, then $f(n) \in V$. This means that the pair $(E_\Sigma(0), E_\Sigma(1))$ is certainly in ϕ_Σ, so if ϕ_Σ is generated by singletons, there must exist $a \in E_\Sigma(0)$ and $b \in E_\Sigma(1)$ such that $(\{a\}, \{b\}) \in \phi_\Sigma$. Thus, there exists a recursive function $f : \mathbb{N} \to \mathbb{N}$ such that $f(n) = a$ for all $n \in E_\Sigma(0)$ and $f(n) = b$ for all $n \in E_\Sigma(1)$. Since $a \neq b$, this implies that $E_\Sigma(0)$ is decidable, which we know to be false. (In fact, this argument is very similar to the argument needed to show that Σ cannot be partitioned.)

4.4.3 Computational density

In this section, the notion of computational density, which we have defined for arrows of OPCA, OPCA_T and OPCA_D, is generalized to arrows of IPCA.

Definition 4.4.19. *We say that an applicative morphism $(p, f) : (A, \phi) \to (B, \psi)$ is **computationally dense** (c.d. for short) if the morphism of PCAs $\mathcal{T}(p, f) : \mathcal{T}(A, \phi) \to \mathcal{T}(B, \psi)$ is c.d. Explicitly, there should exist an $N \in \psi$ such that for all $V \in \psi$, there exists a $U \in \phi$ with $N \cdot f(p(U)) \subseteq V$.*

Since $\mathcal{T}$ is a 2-functor, all the results for c.d. morphisms of PCAs we established in Chapter 2 carry over automatically to morphisms of IPCA. Let us state these properties explicitly.

Proposition 4.4.20. *Let* $(A, \phi) \xrightarrow{(p,f)} (B, \psi) \xrightarrow{(q,g)} (C, \chi)$ *be applicative morphisms.*

(i) If (p, f) *and* (q, g) *are both c.d., then* $(q, g) \circ (p, f)$ *is c.d. as well.*

(ii) If $(q, g) \circ (p, f)$ *is c.d., then* (q, g) *is c.d. as well.*

(iii) Computational density is downwards closed, that is, if there exists an applicative transformation $\mu\colon (p', f') \Rightarrow (p, f)$ *and* (p, f) *is c.d., then* (p', f') *is c.d. as well.*

In particular, left adjoints in IPCA *are c.d.*

Let us consider a few examples of c.d. applicative morphisms. Some of these will have right adjoints in IPCA.

Example 4.4.21. Let (A, ϕ) be an IPCA over $\mathcal{C}$, let $p\colon \mathcal{C} \to \mathcal{D}$ be a regular functor, and let $X \in \mathcal{T}(p(A))$. Then the applicative morphism $(p, \iota_X)\colon (A, \phi) \to (p^*(A, \phi))[X]$ is c.d. Indeed, by Lemma 4.3.28, we can take $N \in \langle p(\phi) \cup \{X\}\rangle$ to be a realizer of $\lambda^* y.yX$.

In particular, if we take $X = p(A)$, we see that the applicative morphism $(p, \mathrm{id})\colon (A, \phi) \to p^*(A, \phi)$ is always c.d. This has the following consequence. Note that any applicative morphism $(p, f)\colon (A, \phi) \to (B, \psi)$ may be factored, up to isomorphism, as a 'horizontal' part followed by a 'vertical' part:

$$\begin{array}{ccc} (A,\phi) & \xrightarrow{(p,\mathrm{id})} & p^*(A,\phi) \\ & \underset{(p,f)}{\searrow} & \big\downarrow{\scriptstyle (\mathrm{id},f)} \\ & & (B,\psi) \end{array}$$

Using Proposition 4.4.20, we see that an applicative morphism is c.d. iff its 'vertical' part is c.d.

Example 4.4.22. Let (A_0, ϕ_0) and (A_1, ϕ_1) be IPCAs, and let $((A_0, A_1), \phi) = (A_0, \phi_0) \times (A_1, \phi_1)$ denote their 2-product. Then the applicative morphism $(\pi_0, \mathrm{id})\colon ((A_0, A_1), \phi) \to (A_0, \phi_0)$ is c.d. by the previous example. In fact, (π_0, id) has a right adjoint inside IPCA. In the notation of the proof of Corollary 4.4.5, we have $(\kappa_0, i_0)\colon (A_0, \phi_0) \to ((A_0, A_1), \phi)$, and we already verified that $(\pi_0, \mathrm{id}) \circ (\kappa_0, i_0)$ is the identity on (A_0, ϕ_0). Note that $\pi_0 \dashv \kappa_0$ is a geometric inclusion $\mathcal{C}_0 \to \mathcal{C}_0 \times \mathcal{C}_1$, whose unit $\eta\colon \mathrm{id} \Rightarrow \kappa_0\pi_0$ is defined by $\eta_{(X_0,X_1)} = (\mathrm{id}, !)\colon (X_0, X_1) \to (X_0, 1) = \kappa_0\pi_0(X_0, X_1)$. If $(1, \mathcal{T}(1))$ denotes the pseudoinitial object of $\mathsf{IPCA}_{\mathcal{D}}$, then it is easily verified that

$$(\kappa_0\pi_0)^*((A_0, A_1), \phi) = ((A_0, 1), \{(U, 1) \mid U \in \phi_0\}) = (A_0, \phi_0) \times (1, \mathcal{T}(1)),$$

and that $(\kappa_0, i_0) \circ (\pi_0, \mathrm{id}) = (\kappa_0\pi_0, (\mathrm{id}_{(A_0,\phi_0)}, ¡))$. Moreover, $\overline{\eta}_{(A_0,A_1)}$ is the applicative morphism $(\mathrm{id}_{(A_0,\phi_0)}, !)\colon ((A_0, A_1), \phi) \to (A_0, \phi_0) \times (1, \mathcal{T}(1))$. Now it easily follows that η is an applicative transformation $\mathrm{id} \Rightarrow (\kappa_0, i_0) \circ (\pi_0, \mathrm{id})$, which shows that $(\pi_0, \mathrm{id}) \dashv (\kappa_0, i_0)$.

Example 4.4.23. If (A, ϕ) is an IPCA and $I \in \mathsf{Asm}(A, \phi)$, then the applicative morphism $(|I|^*, \iota_{E_I})\colon (A, \phi) \to (A, \phi)/I$ is c.d., by Example 4.4.21. Similarly, if $f\colon I \to J$ is a morphism of assemblies, then $(f^*, \iota_{E_I})\colon (A, \phi)/J \to (A, \phi)/I$ is c.d.

Example 4.4.24. Let (A_0, ϕ_0) and (A_1, ϕ_1) be IPCAs over the same base $\mathcal{C}$, and consider their 2-product $((A_0, A_1), \phi)$ in IPCA. We have the product functor $\times\colon \mathcal{C} \times \mathcal{C} \to \mathcal{C}$, which is regular, and $\times^*((A_0, A_1), \phi) = (A_0 \times A_1, \langle \phi_0 \times \phi_1 \rangle)$ is the pseudo*co*product of (A_0, ϕ_0) and (A_1, ϕ_1) in the fiber $\mathsf{IPCA}_{\mathcal{C}}$. In particular, we have the applicative morphism $(\times, \mathrm{id})\colon ((A_0, A_1), \phi) \to (A_0 \times A_1, \langle \phi_0 \times \phi_1 \rangle)$, which is c.d., by Example 4.4.21. However, $(\times, \mathrm{id})$ does not in general have a right adjoint in IPCA, since that would require the product functor $\times$ to have a right adjoint.

On the other hand, the product functor does have a left adjoint, namely, the diagonal functor $\Delta\colon \mathcal{C} \to \mathcal{C} \times \mathcal{C}$. Let us denote the unit and counit of this adjunction by η resp. ε. Now we have $(\Delta, f)\colon (A_0 \times A_1, \langle \phi_0 \times \phi_1 \rangle) \to ((A_0, A_1), \phi)$, where f is given by:

$$\Delta^*(A_0 \times A_1, \langle \phi_0 \times \phi_1 \rangle) = (\Delta \circ \times)^*((A_0, A_1), \phi) \xrightarrow{\overline{\varepsilon}_{(A_0,A_1)}} ((A_0, A_1), \phi)$$

It is easily verified that (Δ, f) is indeed an applicative morphism, and that it is c.d. In fact, we have $(\Delta, f) \dashv (\times, \mathrm{id})$. In order to see this, we first note that ε is an applicative transformation $(\Delta, f) \circ (\times, \mathrm{id}) \Rightarrow \mathrm{id}$ as a result of the way we defined f. Moreover, the diagram of relations

$$\begin{array}{ccc} A_0 \times A_1 & \xrightarrow{\quad\quad \geq \quad\quad} & A_0 \times A_1 \\ & {\scriptstyle \overline{\eta}_{A_0 \times A_1}} \searrow \qquad \nearrow {\scriptstyle \times(f)} & \\ & (A_0 \times A_1) \times (A_0 \times A_1) & \end{array}$$

commutes on the nose, as a result of a triangle identity for $\Delta \dashv \times$. This implies that η is an applicative transformation $\mathrm{id} \Rightarrow (\times, \mathrm{id}) \circ (\Delta, f)$, as desired.

We will also need a formulation of computational density in the spirit of (cdm) from Lemma 2.2.12. In order to formulate a suitable internal version of (cdm), we need to assume a bit more stucture on our base categories than we have done thus far. More precisely, if $(p, f)\colon (A, \phi) \to (B, \psi)$ is an applicative morphism, we will assume that the base category of (B, ψ) is a Heyting category and thus interprets full (typed) first-order logic. Now a statement live $UV\downarrow$, where $U, V \in \mathcal{T}B$, can be expressed internally as $\forall u, v{:}B\,(U(u) \wedge V(v) \to uv\downarrow)$. If $a{:}p(A)$, then we may even consider a statement like $U \cdot f(a)\downarrow$, which should be read as $\forall u, b{:}B\,(U(u) \wedge f(a,b) \to ub{\downarrow})$. In the sequel, we will often use $f(a)$ as a standalone term, even though $\mathcal{D}$ is not assumed to be a topos, trusting that the reader can translate the statements into first-order logic.

Lemma 4.4.25. *Let $(p,f)\colon (A,\phi) \to (B,\psi)$ be an applicative morphism, where $p\colon \mathcal{C} \to \mathcal{D}$ and $\mathcal{D}$ is a Heyting category. Then (p,f) is c.d. if and only if there exists an $M \in \psi$ with the following property:*

$$\text{for all } V \in \psi, \text{ there exists a } U \in \phi \text{ such that } UA{\downarrow} \text{ and:}$$
$$p(U)(r) \wedge V \cdot f(a){\downarrow} \vDash_{r,a:p(A)} M \cdot f(ra) \subseteq V \cdot f(a). \qquad \text{(icdm)}$$

Remark 4.4.26. Note that, while (cdm) only requires ra to be defined if $s \cdot f(a)$ is defined, (icdm) requires UA to be defined always. This move is necessary, because UA lives inside $\mathcal{C}$, whereas $V \cdot f(a)$ is a statement inside $\mathcal{D}$. Note also that the proof of Lemma 2.2.12 implies that in (cdm), we can also require that ra is always defined.

Proof of Lemma 4.4.25. For simplicity, we assume that f preserves the order on the nose. First, suppose we have $M \in \psi$ satisfying (icdm). Given $V \in \psi$, apply (icdm) with $V' = \mathsf{K}V \in \psi$, and find $U' \in \phi$ such that $U'A{\downarrow}$ and $p(U')(r) \vDash_{r,a:p(A)} M \cdot f(ra) \subseteq V$. If $U := U'A \in \phi$, then we have $M \cdot f(p(U)) \subseteq V$ by an easy argument inside $\mathcal{D}$. This shows that (p,f) is c.d.

Conversely, suppose that (p,f) is c.d., say that $N \in \psi$ satisfies (cd) for $\mathcal{T}(p,f)$, and let f preserve application up to $T \in \psi$. Let $M \in \psi$ be a realizer of:

$$\lambda^* x.N(T \cdot f(\mathsf{P}_0) \cdot x)(T \cdot f(\mathsf{P}_1) \cdot x).$$

Given $V \in \psi$, find $U' \in \phi$ such that $N \cdot f(p(U')) \subseteq V$, and define $U = \mathsf{P}U'$. Then $UA = \mathsf{P}U'A{\downarrow}$, and $p(U) = p(\mathsf{P}) \cdot p(U')$, where $p(\mathsf{P})$ is a pairing combinator (that is, realizes $\lambda^* xyz.zxy$) w.r.t. $p(A)$. Now the desired sequent $p(U)(r) \wedge V \cdot f(a){\downarrow} \vDash_{r,a:p(A)} M \cdot f(ra) \subseteq V \cdot f(a)$ follows by a straightforward argument inside $\mathcal{D}$. ☐

Now we can show that, even under the weak assumptions that $\mathcal{C}$ is a regular category and $\mathcal{D}$ is a Heyting category, the functor $\mathsf{Asm}(p,f)$ has a right adjoint, provided that p has one.

Theorem 4.4.27. *Let $(p,f)\colon (A,\phi) \to (B,\psi)$ be an applicative morphism, where $p\colon \mathcal{C} \to \mathcal{D}$, and $\mathcal{D}$ is a Heyting category. If p has a right adjoint and (p,f) is c.d., then $\mathsf{Asm}(p,f)$ has a right adjoint.*

Proof. Let $q\colon \mathcal{D} \to \mathcal{C}$ be the right adjoint of p, and denote the unit and counit of $p \dashv q$ by $\eta\colon \mathrm{id} \Rightarrow qp$ resp. $\varepsilon\colon pq \Rightarrow \mathrm{id}$. For simplicity, we assume that f preserves the order on the nose. Let $M \in \psi$ satisfy (icdm). For the sake of readability, we write F for $\mathsf{Asm}(p,f)$.

We define the right adjoint G of F as follows. Let X be an assembly over (B,ψ), and define $E'_X \subseteq |X| \times A$ by $E'_X(x,a) :\equiv (M \cdot f(a) \subseteq E_X(x))$. Note that here we also use $E_X(x)$ as a standalone term while it really isn't, so the formula $M \cdot f(a) \subseteq E_X(x)$ should be read as:

$$\forall m, b{:}B\,(M(m) \wedge f(a,b) \to mb{\downarrow} \wedge E_X(x, mb)).$$

Now $q(E'_X) \subseteq q(|X|) \times qp(A)$, and we define the subobject $E'_{GX} \subseteq q(|X|) \times A$ as $(\mathrm{id} \times \eta_A)^*(q(E'_X))$, that is, $E'_{GX}(x,a) :\equiv q(E'_X)(x, \eta_A(a))$. Note that, since we assumed that f preserves the order on the nose, E'_X is downwards closed on the right. Since q is left exact, this implies that $q(E'_X)$ is downwards closed on the right, and since η_A preserves the order, this means that E'_{GX} is downwards closed on the right. Now we can define the assembly GX over (A, ϕ) by setting

$$|GX| = \{x \in q(|X|) \mid \exists a : A\,(E'_{GX}(x,a))\} \subseteq q(|X|),$$

and by letting E_{GX} be the restriction of E'_{GX} to $|GX| \times A$.

Before we continue, we first formulate the following lemma.

Lemma 4.4.28. *For every assembly X over (B, ψ), there is a commutative diagram:*

$$\begin{array}{ccc} pE'_{GX} & \longrightarrow & E'_X \\ \big\downarrow & & \big\downarrow \\ pq|X| \times p(A) & \xrightarrow{\varepsilon \times \mathrm{id}} & |X| \times p(A) \end{array} \tag{4.8}$$

Proof of Lemma 4.4.28. The object E'_{GX} is defined by the pullback diagram

$$\begin{array}{ccc} E'_{GX} & \longrightarrow & qE'_X \\ \big\downarrow & & \big\downarrow \\ q|X| \times A & \xrightarrow{\mathrm{id} \times \eta} & q|X| \times qpA \end{array}$$

This means that we can obtain the diagram (4.8) by pasting the following squares:

$$\begin{array}{ccccc} pE'_{GX} & \longrightarrow & pqE'_X & \xrightarrow{\varepsilon} & E'_X \\ \big\downarrow & & \big\downarrow & & \big\downarrow \\ pq|X| \times pA & \xrightarrow{\mathrm{id} \times p\eta} & pq|X| \times pqpA & \xrightarrow{\varepsilon \times \varepsilon} & |X| \times pA \end{array}$$

and using the triangle identity for the bottom composition. □

Now suppose that $g \colon X \to Y$ is an arrow in $\mathsf{Asm}(B, \psi)$, tracked by $T \in \psi$. Let $V \in \psi$ be a realizer of $\lambda x.T(Mx)$, and find a $U \in \phi$ as in (icdm). We claim that

$$\mathcal{C} \colon U(r) \wedge E'_{GX}(x,a) \vDash_{x:q(|X|);r,a:A} E'_{GY}(q(g)(x), ra). \tag{4.9}$$

To this end, we first prove that

$$\mathcal{D} \colon p(U)(r) \wedge E'_X(x,a) \vDash_{x:|X|;r,a:p(A)} E'_Y(g(x), ra). \tag{4.10}$$

Reason inside $\mathcal{D}$ and suppose that we have $x : |X|$ and $r, a : p(A)$ such that $p(U)(r)$ and $E'_X(x,a)$. Then by the definition of E'_X, we have $M \cdot f(a) \subseteq E_X(x)$. This implies that $T(M \cdot f(a)) \subseteq E_Y(g(x))$, hence also $V \cdot f(a) \subseteq E_Y(g(x))$. Since $p(U)(r)$, this yields $M \cdot f(ra) \subseteq V \cdot f(a) \subseteq E_Y(g(x))$, that is, $E'_Y(g(x), ra)$, which proves (4.10).

Now we obtain a commutative diagram

$$\begin{array}{ccccc} pU \times pE'_{GX} & \longrightarrow & pU \times E'_X & \longrightarrow & E'_Y \\ \big\downarrow & & \big\downarrow & & \big\downarrow \\ pU \times pq|X| \times pA & \xrightarrow{\mathrm{id} \times \varepsilon \times \mathrm{id}} & pU \times |X| \times pA & \xrightarrow{\;*\;} & |Y| \times pA \end{array}$$

in $\mathcal{D}$, where $*$ is the arrow sending (r, x, a) to $(g(x), ra)$. Indeed, the left-hand square exists by diagram (4.8), and the right-hand square expresses (4.10). Transposing this diagram yields the diagram

$$\begin{array}{ccc} U \times E'_{GX} & \longrightarrow & qE'_Y \\ \big\downarrow & & \big\downarrow \\ U \times q|X| \times A & \xrightarrow{\;**\;} & q|Y| \times qpA \end{array}$$

in $\mathcal{C}$, where $**$ is the arrow sending (r, x, a) to $(q(g)(x), \eta_A(ra))$. (Observe that, since $UA\!\downarrow$, we know that the application map $p(U) \times p(A) \to p(A)$ is the image of the application map $U \times A \to A$ under p.) This diagram tells us that

$$\mathcal{C}\colon U(r) \wedge E'_{GX}(x, a) \vDash_{x:q(|X|);r,a:A} E'_Y(q(g)(x), \eta_A(ra)).$$

from which (4.9) immediately follows. Since U is inhabited, (4.9) implies that

$$\mathcal{C}\colon \exists a\!:\!A(E'_{GX}(x, a)) \vDash_{x:q(|X|)} \exists a : A(E'_{GY}(q(g)(x), a)),$$

which means that $q(g)$ restricts to an arrow $G(g)\colon |GX| \to |GY|$. Moreover, (4.9) implies that U tracks $G(g)$ as a morphism $GX \to GY$. It is immediate that G is a functor, so it remains to show that $F \dashv G$.

Applying (icdm) to $\mathsf{I} \in \psi$, we see that there exists a $U \in \phi$ such that $UA\!\downarrow$ and

$$\mathcal{D}\colon p(U)(r) \vDash_{r,a:p(A)} M \cdot f(ra) \subseteq f(a).$$

We claim that for every assembly $X \in \mathsf{Asm}(A, \phi)$:

$$\mathcal{D}\colon p(U)(r) \wedge p(E_X)(x, a) \vDash_{x:p(|X|);r,a:p(A)} E'_{FX}(x, ra). \tag{4.11}$$

Indeed, reason inside $\mathcal{D}$ and suppose that we have $x\!:\!p(|X|)$ and $r, a\!:\!p(A)$ such that $p(U)(r)$ and $p(E_X)(x, a)$. Then we find $M \cdot f(ra) \subseteq f(a) \subseteq E_{FX}(x)$, which means precisely that $E'_{FX}(x, ra)$, so (4.11) indeed holds.

The validity of (4.11) can be expressed by a diagram

$$\begin{array}{ccc} pU \times pE_X & \longrightarrow & E'_{FX} \\ \big\downarrow & & \big\downarrow \\ pU \times p|X| \times pA & \xrightarrow{\;*\;} & p|X| \times pA \end{array}$$

in $\mathcal{D}$, where $*$ is the arrow sending (r, x, a) to (x, ra). Transposing this diagram yields a diagram

$$\begin{array}{ccc} U \times E_X & \longrightarrow & qE'_{FX} \\ \downarrow & & \downarrow \\ U \times |X| \times A & \xrightarrow{**} & qp|X| \times qpA \end{array}$$

in $\mathcal{C}$, where $**$ is the arrow sending (r, x, a) to $(\eta_{|X|}(x), \eta_A(ra))$. This diagram tells us that $\mathcal{C}\colon U(r) \wedge E_X(x, a) \vDash_{x:|X|;r,a:A} q(E'_{FX})(\eta_{|X|}(x), \eta_A(ra))$, which implies:

$$\mathcal{C}\colon U(r) \wedge E_X(x, a) \vDash_{x:|X|;r,a:A} E'_{GFX}(\eta_{|X|}(x), ra)). \tag{4.12}$$

Since U is inhabited and E_X is total, (4.12) implies that

$$\mathcal{C}\colon \vDash_{x:|X|} \exists a\colon A\,(E'_{GFX}(\eta_{|X|}(x), a)),$$

so the image of $\eta_{|X|}\colon |X| \to qp(|X|) = q(|FX|)$ is contained in $|GFX|$. This means that we have an arrow $\tilde{\eta}_X\colon |X| \to |GFX|$, and (4.12) tells us that U tracks it as a morphism $X \to GFX$. The naturality of η implies that $\tilde{\eta}$ is natural transformation $\mathrm{id} \Rightarrow GF$.

Now consider an assembly $X \in \mathsf{Asm}(B, \psi)$. We have $|FGX| = p(|GX|) \subseteq pq(|X|)$, so $\varepsilon_{|X|}\colon pq(|X|) \to |X|$ restricts to an arrow $\tilde{\varepsilon}_X\colon |FGX| \to |X|$. We will show that M tracks $\tilde{\varepsilon}_X$ as a morphism $FGX \to X$, so that $\tilde{\varepsilon}$ is a natural transformation $FG \Rightarrow \mathrm{id}$. To this end, reason inside $\mathcal{D}$ and suppose we have $x\colon|FGX| = p(|GX|)$ and $m, b\colon B$ such that $M(m)$ and $E_{FGX}(x, b)$. Then there exists an $a\colon p(A)$ such that $p(E_{GX})(x, a)$ and $f(a, b)$. By the diagram (4.8), it follows that $E'_X(\varepsilon_{|X|}(x), a)$. By the definition of E'_X, we conclude that $mb\downarrow$, and $E_X(\varepsilon_{|X|}(x), mb)$, as desired.

Finally, the triangle identities for ε and η yield that the triangle identities hold for $\tilde{\varepsilon}$ and $\tilde{\eta}$ as well, so $F \dashv G$. This completes the proof. □

Our examples of computationally dense applicative morphisms thus yield the following examples of geometric morphisms between categories of assemblies. Of course, in the cases where we already have an adjunction in IPCA, we do not need to appeal to Theorem 4.4.27.

Example 4.4.29. Let (A_0, ϕ_0) and (A_1, ϕ_1) be IPCAs, and denote their 2-product by $((A_0, A_1), \phi)$. In Example 4.4.22, we saw that the projection morphism $(\pi_0, \mathrm{id})\colon ((A_0, A_1), \phi) \to (A_0, \phi_0)$ has a right adjoint (κ_0, i_0), and that the counit of this adjunction is an isomorphism. According to Theorem 4.4.3, the composition

$$\mathsf{Asm}(A_0, \phi_0) \times \mathsf{Asm}(A_1, \phi_1) \simeq \mathsf{Asm}((A_0, A_1), \phi) \xrightarrow{\mathsf{Asm}(\pi_0, \mathrm{id})} \mathsf{Asm}(A_0, \phi)$$

is the projection functor. Thus, we see that the adjunction $(\pi_0, \mathrm{id}) \dashv (\kappa_0, i_0)$ in IPCA gives rise to the pseudocoproduct inclusion $\mathsf{Asm}(A_0, \phi_0) \hookrightarrow \mathsf{Asm}(A_0, \phi_0) \times \mathsf{Asm}(A_1, \phi_1)$ in GEOM.

Example 4.4.30. In Section 4.3.3, we showed that $\mathsf{Asm}(A,\phi)$ is a regular category, where (A,ϕ) is an IPCA over a regular category $\mathcal{C}$. If the base category $\mathcal{C}$ has some further structure, then $\mathsf{Asm}(A,\phi)$ will have more structure as well. We can give one example of this using the theory of slicing and of computational density. Suppose that $\mathcal{C}$ is a Heyting category which is cartesian closed. We will show that $\mathsf{Asm}(A,\phi)$ is cartesian closed as well. Let $I \in \mathsf{Asm}(A,\phi)$ be an assembly. Then we have the applicative morphism $(|I|^*, \iota_{E_I})\colon (A,\phi) \to (A,\phi)/I$, which is c.d. according to Example 4.4.23. Moreover, $\mathcal{C}/|I|$ is also a Heyting category, and if $\mathcal{C}$ is cartesian closed, then $|I|^*$ has a right adjoint, so by Theorem 4.4.27, $\mathsf{Asm}(|I|^*, \iota_{E_I})$ has a right adjoint as well. Under the equivalence $\mathsf{Asm}((A,\phi)/I) \simeq \mathsf{Asm}(A,\phi)/I$, this means that $I^*\colon \mathsf{Asm}(A,\phi) \to \mathsf{Asm}(A,\phi)/I$ has a right adjoint, so $\mathsf{Asm}(A,\phi)$ is indeed cartesian closed.

By a similar argument, if $\mathcal{C}$ is a Heyting category which is *locally* cartesian closed, then $\mathsf{Asm}(A,\phi)$ is locally cartesian closed as well.

Example 4.4.31. Let (A_0,ϕ_0) and (A_1,ϕ_1) be IPCAs over the same base $\mathcal{C}$. In Example 4.4.24, we saw that there is an adjunction

$$((A_0,A_1),\phi) \underset{(\times,\mathrm{id})}{\overset{(\Delta,f)}{\underset{\longrightarrow}{\overset{\longleftarrow}{\bot}}}} (A_0 \times A_1, \langle \phi_0 \times \phi_1 \rangle)$$

in IPCA, which yields a geometric morphism $\mathsf{Asm}(A_0,\phi_0) \times \mathsf{Asm}(A_1,\phi_1) \to \mathsf{Asm}(A_0 \times A_1, \langle \phi_0 \times \phi_1 \rangle)$.

Finally, we provide a partial converse to Theorem 4.4.27, for which we assume that $\mathcal{D}$ is a topos.

Proposition 4.4.32. *Let $(p,f)\colon (A,\phi) \to (B,\psi)$ be an applicative morphism, where $p\colon \mathcal{C} \to \mathcal{D}$, and $\mathcal{D}$ is a topos. If $\mathsf{Asm}(p,f)$ has a right adjoint, then (p,f) is c.d.*

Proof. Again, we write F for $\mathsf{Asm}(A,\phi)$. Suppose that F has a right adjoint G, and let ε be the counit of $F \dashv G$. Since $\mathcal{D}$ is assumed to be a topos, we can view $\mathcal{T}B$ as an object of $\mathcal{D}$. Now consider the assembly $T_B \in \mathsf{Asm}(B,\psi)$, where $|T_B| = \mathcal{T}B$ and $E_{T_B} \subseteq \mathcal{T}B \times B$ is the element relation, and let $N \in \psi$ track $\varepsilon_{T_B}\colon FGT_B \to T_B$.

Suppose that $V \in \psi$. Then the global section $V\colon 1 \to \mathcal{T}B$ is also a morphism of assemblies $1 \to T_B$, for it is tracked by $\mathsf{K}V$. Since $F1 \simeq 1$, this morphism can be transposed to an arrow $\tilde{V}\colon 1 \to GT_B$ of $\mathsf{Asm}(A,\phi)$. Then $F\tilde{V}$ is a global section $1 \simeq F1 \to FGT_B$, and by the adjunction, we have $\varepsilon_{T_B}(F\tilde{V}) = V$.

Now define $U = \{a \in A \mid E_{GT_B}(\tilde{V},a)\} \in \mathcal{T}A$. If $W \in \phi$ tracks $\tilde{V}\colon 1 \to GT_B$, then WA is defined and a subobject of U, which implies that $U \in \phi$. Now $f(p(U)) = \{b \in B \mid E_{FGT_B}(F\tilde{V},b)\}$, which means that $\mathsf{N} \cdot f(p(U))$ is defined and a subobject of

$$\{b \in B \mid E_{T_B}(\varepsilon_{T_B}(F\tilde{V}),b)\} = \{b \in B \mid E_{T_B}(V,b)\} = V,$$

as desired. □

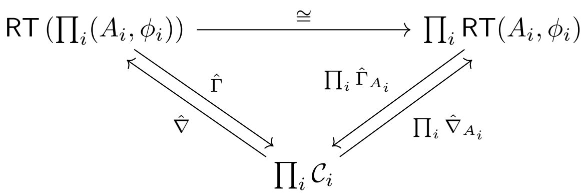
4.4.4 Extension to realizability toposes

Thus far, we have discussed products and slicing for categories of assemblies over IPCAs. In this section, we investigate to what extent these results generalize to realizability toposes, in the case where the base categories are toposes. Here the definition of $\mathsf{RT}(A, \phi)$ as $\mathsf{Asm}(A, \phi)_{\mathrm{ex/reg}}$ is particularly helpful.

Recall from Remark 3.2.5 that, if $\mathcal{R}$ is a regular category, its ex/reg completion $\mathcal{R}_{\mathrm{ex/reg}}$ may be described as follows. Its objects are pairs (X, R), where X is an object of $\mathcal{R}$ and $R \subseteq X \times X$ is an equivalence relation on X. An arrow $(X, R) \to (Y, S)$ is a subobject $F \subseteq X \times Y$ which is relational, total and single-valued. Moreover, $\mathcal{R}$ sits inside $\mathcal{R}_{\mathrm{ex/reg}}$ via the fully faithful functor that sends $X \in \mathcal{R}$ to X equipped with the diagonal relation. In this section, we will denote this functor $\mathcal{R} \hookrightarrow \mathcal{R}_{\mathrm{ex/reg}}$ by $\eta_{\mathcal{R}}$.

Now suppose that we have two regular categories $\mathcal{R}$ and $\mathcal{S}$. Because the internal logic of the regular category $\mathcal{R} \times \mathcal{S}$ is computed 'coordinatewise', it is immediately clear from the description of ex/reg completions above that there is an isomorphism $(\mathcal{R} \times \mathcal{S})_{\mathrm{ex/reg}} \cong \mathcal{R}_{\mathrm{ex/reg}} \times \mathcal{S}_{\mathrm{ex/reg}}$ which identifies $\eta_{\mathcal{R}\times\mathcal{S}}$ with $\eta_{\mathcal{R}} \times \eta_{\mathcal{S}}$. Similar results hold for products of more than two regular categories, so the following result follows immediately from Theorem 4.4.3.

Corollary 4.4.33. *The pseudofunctor* $\mathsf{RT}\colon \mathsf{IPCA}_{\mathrm{top}} \to \mathsf{REG}$ *preserves small 2-products, and in particular, realizability toposes over IPCAs are closed under small 2-products of categories. If, for each* $i \in I$*, we have an IPCA* (A_i, ϕ_i) *over a topos* $\mathcal{C}_i$*, then the isomorphism* $\mathsf{RT}\left(\prod_i A_i, \phi_i)\right) \cong \prod_i \mathsf{RT}(A_i, \phi_i)$ *makes the triangles in following diagram commute:*



Example 4.4.34. Continuing Example 4.4.22 and Example 4.4.29, we see that the pesudocoproduct inclusion $\mathsf{RT}(A_0, \phi_0) \to \mathsf{RT}(A_0, \phi_0) \times \mathsf{RT}(A_1, \phi_1)$ arises from an adjunction in IPCA, provided (A_0, ϕ_0) and (A_1, ϕ_1) are both IPCAs over a topos.

Example 4.4.35. Continuing Example 4.4.24 and Example 4.4.31, let (A_0, ϕ_0) and (A_1, ϕ_1) be IPCAs over the same base topos $\mathcal{C}$. Then we get a geometric morphism $\mathsf{RT}(A_0, \phi_0) \times \mathsf{RT}(A_1, \phi_1) \to \mathsf{RT}(A_0 \times A_1, \langle \phi_0 \times \phi_1 \rangle)$.

Now take $\mathcal{C} = \mathsf{Set}$, and suppose that A_0 and A_1 are ordinary PCAs, seen as IPCA over Set that are generated by singletons. Then the pseudocoproduct of A_0 and A_1 in the fiber $\mathsf{IPCA}_{\mathsf{Set}}$ is easily seen to coincide with the pseudocoproduct $A_0 \times A_1$ of A_0 and A_1 in OPCA. Thus, we get a geometric morphism $\mathsf{RT}(A_0) \times \mathsf{RT}(A_1) \to \mathsf{RT}(A_0 \times A_1)$. This geometric morphism is the amalgamation of the inclusions $\mathsf{RT}(A_i) \hookrightarrow \mathsf{RT}(A_0 \times A_1)$ we used in the proof of Theorem 4.2.5.

For slicing, we can only obtain a partial result: realizability toposes over IPCAs are closed under slicing *over assemblies*. First, let us give a description of the ex/reg completion of a slice category $\mathcal{R}/I$, where I is an object of $\mathcal{R}$. This description may be known, but we have not been able to find a reference for it, so we include it for the sake of completeness.

An object of $(\mathcal{R}/I)_{\mathrm{ex/reg}}$ is an object $k_X \colon X \to I$ of $\mathcal{R}/I$, equipped with an equivalence relation R on X that lies over I in the sense that $R \subseteq X \times_I X$. We may also express this final requirement in logical terms as $R(x, x') \vDash_{x,x':X} k_X(x) = k_X(x')$. Similarly, an arrow $(X, R) \to (Y, S)$ of $(\mathcal{R}/I)_{\mathrm{ex/reg}}$ is an arrow $F \colon (X, R) \to (Y, S)$ of $\mathcal{R}_{\mathrm{ex/reg}}$, with the additional property that $F \subseteq X \times_I Y$, that is, $F(x, y) \vDash_{x:X;y:Y} k_X(x) = k_Y(y)$.

Lemma 4.4.36. *Let $\mathcal{R}$ be a regular category and let I be an object of $\mathcal{R}$. Then there is an isomorphism $(\mathcal{R}/I)_{\mathrm{ex/reg}} \cong (\mathcal{R}_{\mathrm{ex/reg}})/\eta_{\mathcal{R}} I$ that makes the following diagram commute:*

$$\begin{array}{ccc} & \mathcal{R}/I & \\ {\scriptstyle \eta_{\mathcal{R}/I}}\swarrow & & \searrow{\scriptstyle \eta_{\mathcal{R}}} \\ (\mathcal{R}/I)_{\mathrm{ex/reg}} & \xrightarrow{\ \cong\ } & (\mathcal{R}_{\mathrm{ex/reg}})/\eta_{\mathcal{R}} I \end{array}$$

Proof. We define the isomorphism $\Phi \colon (\mathcal{R}/I)_{\mathrm{ex/reg}} \to (\mathcal{R}_{\mathrm{ex/reg}})/\eta_{\mathcal{R}} I$ as follows. If (X, R) is an object of $(\mathcal{R}/I)_{\mathrm{ex/reg}}$, then we let $\Phi(X, R)$ be the arrow $(X, R) \to \eta_{\mathcal{R}} I$ of $\mathcal{R}_{\mathrm{ex/reg}}$ given by the relation $k_X(x) = i$. This relation is clearly total and single-valued, and relationality follows from the sequent $R(x, x') \vDash_{x,x':X} k_X(x) = k_X(x')$.

Before we define Φ on arrows, let us check that Φ is bijective on objects. Injectivity is clear, so suppose we have an arrow $K \colon (X, R) \to \eta_{\mathcal{R}} I$ of $\mathcal{R}_{\mathrm{ex/reg}}$. Then K is total, i.e., $\vDash_{x:X} \exists i : I\, (K(x, i))$, and K is single-valued, i.e., $K(x, i) \wedge K(x, i') \vDash_{x:X;i,i':I} i = i'$. This implies that there is an *arrow* $k_X \colon X \to I$ such that K is the relation $k_X(x) = i$. Moreover, the relationality of K then means precisely that $R(x, x') \vDash_{x,x':X} k_X(x) = k_X(x')$, so we can conclude that $K \colon (X, R) \to \eta_{\mathcal{R}} I$ is in the image of Φ.

For arrows, we note that an arrow $\Phi(X, R) \to \Phi(Y, S)$ of $(\mathcal{R}_{\mathrm{ex/reg}})/\eta_{\mathcal{R}} I$ is an arrow $F \colon (X, R) \to (Y, S)$ of $\mathcal{R}_{\mathrm{ex/reg}}$ such that the diagram of relations

$$\begin{array}{ccc} X & \xrightarrow{\ F\ } & Y \\ {\scriptstyle k_X(x)=i}\searrow & & \swarrow{\scriptstyle k_Y(y)=i} \\ & I & \end{array}$$

commutes. This diagram is equivalent to $F(x, y) \vDash_{x:X;y:Y} k_X(x) = k_Y(y)$, so we see that setting $\Phi(F) = F$ makes Φ into a well-defined, fully faithful functor. Finally, the diagram stated in the lemma is easy to check. $\square$

From Theorem 4.4.9 and Lemma 4.4.36, we now immediately get the following corollary.

Corollary 4.4.37. *Let (A, ϕ) be an IPCA over a topos $\mathcal{C}$, and let I be an* assembly *over (A, ϕ). Then there exists an equivalence of categories $\mathsf{RT}(A, \phi)/I \simeq \mathsf{RT}((A, \phi)/I)$ that makes the triangles in the following diagram commute (up to isomorphism):*

$$\begin{array}{ccc} \mathsf{RT}(A,\phi)/I & \xrightarrow{\;\simeq\;} & \mathsf{RT}((A,\phi)/I) \\ {\scriptstyle \hat{\eta}_I^* \circ \hat{\nabla}} \nwarrow \searrow {\scriptstyle \hat{\Gamma}} & & {\scriptstyle \hat{\Gamma}} \swarrow \nearrow {\scriptstyle \hat{\nabla}} \\ & \mathcal{C}/|I| & \end{array}$$

In particular, realizability toposes over IPCAs are closed under slicing over assemblies.

Example 4.4.38. An open subtopos of a topos $\mathcal{E}$ is a subtopos of the form $\mathcal{E}/U$, where U is a subterminal object. In a realizability topos $\mathsf{RT}(A, \phi)$, subterminals are assemblies, so by Corollary 4.4.37 above, open subtoposes of $\mathsf{RT}(A, \phi)$ are again realizability toposes over an IPCA.

Now suppose that A is a PCA. Continuing Example 4.4.14, we see that every nontrivial open subtopos of $\mathsf{RT}(A)$ is of the form $\mathsf{RT}(A, \langle (TA)^{\#} \cup \{X\} \rangle)$ for some $X \in TA$. Thus, nontrivial open subtoposes correspond to finite extensions of the filter $(TA)^{\#} \subseteq TA$. We also see that, if 1_a denotes the partitioned subterminal assembly with $|1_a| = \{*\}$ and $E_{1_a}(*) = \downarrow\{a\}$, then $\mathsf{RT}(A)/1_a \simeq \mathsf{RT}(A[a])$.

Chapter 5

Computing with Oracles and Higher-Order Functionals

In this chapter, we investigate computation with oracles and higher-order functionals in an arbitrary PCA A. The paper [vO06] describes how to 'force' a partial function on A to become computable, using a general notion of oracle computation. The paper [FvO16] generalizes this to type-2 functionals, i.e., functions that assume values in A, but whose inputs are functions on A. The main goal of this chapter is to obtain similar results about type-3 functionals, i.e., functions whose domain consists of type-2 functionals. Our main strategy will be to view a type-3 functional on A as a *type-2* functional on another PCA $\mathcal{B}A$, whose elements are partial functions $A \rightharpoonup A$. This PCA $\mathcal{B}A$ is a generalization of the Van Oosten model $\mathcal{B}$ from Example 2.1.38, and was first described in [vO11]. In order for this strategy to work, our treatment must diverge substantially from [vO11] in the following respect. While [vO11] only discusses absolute discrete PCAs, we will view $\mathcal{B}A$ as a relative ordered PCA, *even when A itself is absolute and discrete.* As a consequence of this, we will also need to generalize the material from [vO06] and [FvO16], which also discuss absolute discrete PCAs, to relative ordered PCAs.

Thus, before we can treat the type-3 case in Section 5.4 below, we must first revisit the constructions of $A[\alpha]$, $A[F]$ and $\mathcal{B}A$. Let us describe in a bit more detail how these constructions occur in the literature, and what our contribution is.

1. The paper [vO06] constructs, for an absolute discrete PCA A and partial function $\alpha\colon A \rightharpoonup A$, a new PCA $A[\alpha]$ whose application map is given by 'computation with an oracle for α'. This PCA $A[\alpha]$ has the following property: every applicative morphism $A \multimap B$ that 'makes α computable'

(in a precise sense to be specified below) factors, essentially uniquely, through $A[\alpha]$ (see [vO06, Theorem 2.2]). Thus, $A[\alpha]$ may be viewed as the 'universal solution' to the problem of making α computable. In Section 5.1, we construct a PCA $A[\alpha]$ with a similar universal property, but in the more general setting where A is a relative ordered PCA.

2. The paper [vO11] constructs, for each absolute discrete PCA A, a new PCA $\mathcal{B}A$. As we mentioned, $\mathcal{B}A$ generalizes the Van Oosten PCA $\mathcal{B}$ from Example 2.1.38, and its underlying set consists of all partial functions $A \rightharpoonup A$. Its application map, like the application map on $A[\alpha]$, can be seen as a computation with oracles. In Section 5.2, we construct a version of $\mathcal{B}A$ which is relative and ordered, even when A is absolute and discrete.

3. The paper [FvO16] generalizes the construction of $A[\alpha]$ to partial functions $F\colon A^A \rightharpoonup A$, which, as we explained above, can be viewed as 'type-2' functionals. For each such a type-2 functional F, there is a new PCA $A[F]$ such that every applicative morphism $A \multimap B$ that 'makes F computable' factors essentially uniquely through $A[F]$; this is [FvO16, Theorem 3.1]. It turns out, however, that the construction of $A[F]$ is not quite as universal as the construction of $A[\alpha]$; see Remark 5.3.15 below. We will obtain a generalization of [FvO16, Theorem 3.1] in Section 5.3.2 below (Theorem 5.3.13). This generalization will enable us to apply the construction of $A[F]$ to the PCA $\mathcal{B}A$ in Section 5.4.

The fact that all our PCAs are relative and ordered presents a few challenges that are absent in the absolute, discrete case:

- In the case of ordered PCAs, not all partial functions $\alpha\colon A \rightharpoonup A$ are suited to serve as oracles in the construction of $A[\alpha]$. Thus, we will only perform this construction for certain partial functions α. Similarly, $\mathcal{B}A$ will in general be a proper subset of the set of all partial functions $A \rightharpoonup A$; see Definition 5.1.1 below. Moreover, the definition of application in $A[\alpha]$ and in $\mathcal{B}A$ must be adjusted slightly to suit the ordered context; see Remark 5.1.12 below.

- The construction of $A[F]$ will not work for arbitrary ordered PCAs. Thus, we introduce a notion of *chain-completeness* (Definition 5.3.1), and show that $A[F]$ may be constructed for all PCAs A whose order is chain-complete.

As we explained, the main application of having the constructions of $A[\alpha]$, $\mathcal{B}A$ and $A[F]$ for relative ordered PCAs is to obtain results for the case of type-3 functionals. Additionally, we get a number of other results that are absent in the theory of absolute discrete PCAs:

- There is a morphism of PCAs $i\colon A \to \mathcal{B}A$. In the setup of [vO11], this morphism is not c.d., since the cardinality of $\mathcal{B}A$ is larger than the cardinality of A. In our setup, on the other hand, the morphism i *is* c.d., and thus yields a geometric morphism $\mathsf{RT}(\mathcal{B}A) \to \mathsf{RT}(A)$. We will see that this geometric morphism is local (Proposition 5.2.13).

- We show that the constructions of $A[\alpha]$ and $A[F]$ are connected to $\mathcal{B}A$ in a way that has not been observed before (see Proposition 5.2.15 and the discussion preceding Theorem 5.3.13). These connections depend crucially on the fact that $\mathcal{B}A$ is a relative ordered PCA.

5.1 Oracle computation

In this section, we give the construction of $A[\alpha]$ for relative ordered PCAs A and (certain) partial functions α on A. First, we give a precise definition of the notion that 'α is computable', and that a morphism $A \to B$ 'makes α computable'. We will also treat the case of higher-type functionals, which we will need later in the chapter. Then, we treat the construction of $A[\alpha]$ and prove its universal property.

5.1.1 Representing functions and functionals

As announced above, the construction of $A[\alpha]$ will not work for every partial function $\alpha\colon A \rightharpoonup A$, but only for those α that are 'compatible' with the order on A. The following definition makes this precise.

Definition 5.1.1. *Let A be a PCA. The set $\mathcal{B}A$ is defined as the set of all partial functions $\alpha\colon A \rightharpoonup A$ such that: if $a' \le a$, then $\alpha(a') \preceq \alpha(a)$. We view $\mathcal{B}A$ as a poset with the order from Definition 2.1.4, i.e., $\alpha \le \beta$ iff $\alpha(a) \preceq \beta(a)$ for all $a \in A$.*

Note that $\alpha \in \mathcal{B}A$ tells us two things. First, the domain of α, as a partial function $A \rightharpoonup A$, must be downwards closed, and second, α must be order-preserving on its domain. Moreover, we have $\alpha \le \beta$ iff both $\operatorname{dom}\alpha \supseteq \operatorname{dom}\beta$, and $\alpha(a) \le \beta(a)$ for all $a \in \operatorname{dom}\beta$. As we shall see in the next section, $\mathcal{B}A$ can be equipped with a partial applicative structure, making it into a PCA. In this section, we will view $\mathcal{B}A$ simply as a poset.

Definition 5.1.2. *Let A be a PCA and $\alpha \in \mathcal{B}A$.*

*(i) We say that an element $r \in A$ **represents** α if $ra \preceq \alpha(a)$ for all $a \in A$. The function α is called **representable** if it is represented by an $r \in A^{\#}$.*

*(ii) Now let $f\colon A \to B$ be a morphism of PCAs. We say that an element $s \in B$ **represents** α w.r.t. f if $s \cdot f(a) \preceq f(\alpha(a))$ for all $a \in A$. The function α is called **representable** w.r.t. f if it is represented w.r.t. f by some $s \in B^{\#}$.*

Note that (i) is actually a special case of (ii), if we let f be id_A.

Remark 5.1.3. If $r \in A$, then the partial function $a \mapsto ra$ is always in $\mathcal{B}A$, as follows from axiom (A) from Definition 2.1.1. This means that the set of all representable α can be described as $\uparrow\{a \mapsto ra \mid r \in A^\#\} \subseteq \mathcal{B}A$. This will be relevant in the next section.

Lemma 5.1.4. *Let $A \xrightarrow{f} B \xrightarrow{g} C$ be morphisms of PCAs and let $\alpha \in \mathcal{B}A$.*

(i) If α is representable w.r.t. f, then α is representable w.r.t. gf as well.

(ii) If g is an elementary inclusion (Definition 2.2.7) and α is representable w.r.t. gf, then α is representable w.r.t. f as well.

In particular, if $\alpha \in \mathcal{B}A$ is representable, then it is representable w.r.t. any morphism of PCAs $f\colon A \to B$.

Proof. (i) Let $t, u \in C^\#$ realize g and suppose that $s \in B^\#$ represents α w.r.t. f. If we define $s' \in C^\#$ as $\lambda^* x.u(t \cdot g(s) \cdot x)$, then

$$s' \cdot g(f(a)) \preceq u(t \cdot g(s) \cdot g(f(a))) \preceq u \cdot g(s \cdot f(a)) \preceq g(f(\alpha(a))),$$

so s' represents α w.r.t. gf.

(ii) Suppose that $s \in C^\#$ respresents α w.r.t. gf, and find an $s' \in B^\#$ such that $g(s') \leq s$. Then:

$$g(s' \cdot f(a)) \simeq g(s') \cdot g(f(a)) \preceq s \cdot g(f(a)) \preceq g(f(\alpha(a))),$$

hence $s' \cdot f(a) \preceq f(\alpha(a))$, as desired. □

We will also need a notion of representability w.r.t. (partial) applicative morphisms.

Definition 5.1.5. *If $f\colon A \multimap B$ is an applicative morphism (resp. $f\colon A \hookrightarrow\!\!\!\!\!\multimap B$ a partial applicative morphism) and $\alpha \in \mathcal{B}A$, then we say that α is **representable** w.r.t. f if α is representable w.r.t. $f\colon A \to TB$ (resp. $f\colon A \to DB$).*

Note that, by Lemma 5.1.4(ii), $\alpha \in \mathcal{B}A$ is representable w.r.t. an applicative morphism $f\colon A \multimap B$ iff α is representable w.r.t. f, considered as a partial applicative morphism $A \hookrightarrow\!\!\!\!\!\multimap B$. Thus, we can say that α is representable w.r.t. f without specifying whether we view f as an arrow of OPCA_T or of OPCA_D. Similarly, if $f\colon A \to B$ is a morphism of PCAs, then $\alpha \in \mathcal{B}A$ is representable w.r.t. $f\colon A \to B$ iff α is representable w.r.t. the projective applicative morphism $\delta_B f\colon A \multimap B$. This means we can say that α is representable w.r.t f without specifying whether we view f as an arrow of OPCA or of OPCA_T.

Let us also remark that Lemma 5.1.4(i) automatically extends to (partial) applicative morphisms. That is, if $\alpha \in \mathcal{B}A$ is representable w.r.t. $f\colon A \multimap B$, and we have $g\colon B \multimap C$, then α is also representable w.r.t. $gf\colon A \multimap C$. Indeed, $gf\colon A \multimap C$ can be written as $A \xrightarrow{f} TB \xrightarrow{\tilde{g}} TC$, so this follows fron Lemma 5.1.4(i). The case of partial applicative morphisms is analogous.

If $\alpha \in \mathcal{B}A$ is representable w.r.t. a partial applicative morphism $f\colon A \hookrightarrow\!\!\!\!\!\multimap B$, then there must exist an $s \in B^\#$ such that $s \cdot f(a) \subseteq f(\alpha(a))$ for all $a \in \operatorname{dom}\alpha$.

By the usual abuse of terminology, we say that such an s represents α, even though the representer is really $\downarrow\{s\}$. By axiom (A) from Definition 2.1.1, the set of all representers of α forms a downset of A, for which we introduce the following notation.

Definition 5.1.6. *Let $f\colon A \multimap B$ be a partial applicative morphism and let $\alpha \in \mathcal{B}A$. We define:*

$$\mathrm{rep}^f(\alpha) = \{s \in B \mid \forall a \in \mathrm{dom}\,\alpha\,(s \cdot f(a) \subseteq f(\alpha(a)))\} \in DB.$$

Note that $\mathrm{rep}^f(\alpha) \in DB$ is the largest representer of α w.r.t. $f\colon A \to DB$, and that α is representable w.r.t. f iff $\mathrm{rep}^f(\alpha) \in (DB)^\#$. In Section 5.2, we will show that $\mathcal{B}A$ can be equipped with a PCA structure, and that $\mathrm{rep}^f\colon \mathcal{B}A \to DB$ is always a partial applicative morphism $\mathcal{B}A \multimap B$.

Now let us treat the representability of *higher-type functionals*, which we will need in Section 5.3 and Section 5.4.

Definition 5.1.7. *For a PCA A, we define posets $\mathcal{B}_nA$, where $n \geq 0$, by induction on n:*

- $\mathcal{B}_0A = (A, \leq)$.
- *$\mathcal{B}_{n+1}A$ consists of all partial functions $F\colon \mathcal{B}_nA \rightharpoonup A$ such that $\alpha \leq \beta$ implies $F(\alpha) \preceq F(\beta)$ for all $\alpha, \beta \in \mathcal{B}_nA$. Moreover, if $F, G \in \mathcal{B}_{n+1}A$, then we set $F \leq G$ iff $F(\alpha) \preceq G(\alpha)$ for all $\alpha \in \mathcal{B}_nA$.*

Note that $\mathcal{B}_1A$ coincides with $\mathcal{B}A$ as defined in Definition 5.1.1. Now let us give the definition of representability for higher-type functionals. We phrase the definition in terms of a 'downset of representers' as in Definition 5.1.5 above.

Definition 5.1.8. *Let $f\colon A \multimap B$ be a partial applicative morphism. We define, for each $n \geq 0$, a function $\mathrm{rep}^f_n\colon \mathcal{B}_nA \to DB$, by induction on n.*

- *If $n = 0$ and $a \in A$, then we set $\mathrm{rep}^f_0(a) = f(a)$.*
- *For $F \in \mathcal{B}_{n+1}A$, we set:*

$$\mathrm{rep}^f_{n+1}(F) = \{s \in B \mid \forall \alpha \in \mathrm{dom}\,F\,(s \cdot \mathrm{rep}^f_n(\alpha) \subseteq f(F(\alpha)))\} \in DB.$$

If $\alpha \in \mathcal{B}_nA$, then we say that $s \in B$ ***represents*** *α if $s \in \mathrm{rep}^f_n(\alpha)$, and we say that α is* ***representable*** *w.r.t. f if $\mathrm{rep}^f_n(\alpha) \in (DB)^\#$, that is, α is represented w.r.t f by means of some $s \in B^\#$. Moreover, we say that $r \in A$ represents α (without qualification) if r represents α w.r.t. $\mathrm{id}_A\colon A \multimap A$ and similarly, we say that α is representable (without qualification) if α is representable w.r.t. id_A.*

Note that here, we only define representability w.r.t. partial applicative morphisms. Representability w.r.t. morphisms of PCAs and applicative morphisms can be taken to be special cases of Definition 5.1.8.

For $n = 1$, we have that $\mathrm{rep}_1^f(\alpha) = \mathrm{rep}^f(\alpha)$, thus the notions of representability for $\alpha \in \mathcal{B}_1 A = \mathcal{B}A$ from Definition 5.1.5 and Definition 5.1.8 coincide. For $n = 0$, the notions from Definition 5.1.8 are somewhat degenerate. If $f\colon A \multimap B$, then $s \in B$ represents a w.r.t. f iff $s \in f(a)$, and $a \in A$ is representable w.r.t. f iff $a \in \mathrm{dom}^\# f$, where $\mathrm{dom}^\#$ is as in Definition 2.3.18. Now let us revisit Example 2.3.20.

Example 5.1.9. Let A be a PCA and let $a \in A$. Then we have the PCA $A[a] = (A, \langle A^\# \cup \{a\}\rangle, \cdot, \leq)$, and the morphism of PCAs $\iota_a\colon A \to A[a]$, which is simply the identity on A. This is the universal partial applicative morphism that makes a representable in the 'degenerate' sense above. More precisely, for each PCA B, composition with ι_a yields an isomorphism of preorders:

$$\mathsf{OPCA}_D(A[a], B) \cong \{f \in \mathsf{OPCA}_D(A, B) \mid a \in \mathrm{dom}^\# f\}.$$

This follows simply by observing that, if $f\colon A \multimap B$ is such that $a \in \mathrm{dom}^\# f$, then f is also a partial applicative morphism $A[a] \multimap B$, by Lemma 2.2.5.

For $n > 1$, we do not have a result analogous to Lemma 5.1.4(i). In other words, representability of higher-type functionals cannot be 'transferred' along morphisms. Let us give an example of this phenomenon.

Example 5.1.10. Let A be Kleene's first model $\mathcal{K}_1$. In this case, $\mathcal{B}A$ is simply the set of all partial functions $\mathbb{N} \rightharpoonup \mathbb{N}$. Moreover, the minimal elements of $\mathcal{B}A$ are the *total* functions of $\mathbb{N} \to \mathbb{N}$. This means that any function $\mathbb{N}^\mathbb{N} \to \mathbb{N}$ is automatically in $\mathcal{B}_2\mathcal{K}_1$. Consider the functional $F\colon \mathbb{N}^\mathbb{N} \to \mathbb{N}$ defined by:

$$F(\alpha) = \begin{cases} 0 & \text{if } \alpha \text{ is recursive;} \\ 1 & \text{otherwise.} \end{cases}$$

Note that $e \in \mathbb{N}$ represents a total function α iff $\varphi_e = \alpha$. In particular, only recursive functions have a representer at all, which means that any index for the constant function with value 0 is a representer of F.

In Example 2.3.21, we saw that there is a morphism of PCAs $f\colon \mathcal{K}_1 \to \mathcal{K}_2$ that sends n to the constant function $\hat{n}$ with value n. Suppose that F is representable w.r.t. f. Define a recursive function $\rho\colon \mathbb{N} \to \mathbb{N}$ such that $\rho([[x]]) = 1$ and:

$$\rho([[x, y], u_0, \ldots, u_{i-1}]) = \begin{cases} 0 & \text{if } i \leq y; \\ u_y + 2 & \text{if } i > y. \end{cases}$$

Then it is easily verified that $\rho\alpha\hat{n}(m) = \alpha(n)$ for all $\alpha \in \mathbb{N}^\mathbb{N}$ and $m, n \in \mathbb{N}$. Thus, we have $\rho\alpha\hat{n} = \widehat{\alpha(n)}$, which means that $\rho\alpha \in \mathrm{rep}^f(\alpha)$ for all $\alpha \in \mathbb{N}^\mathbb{N}$. Using ρ and a representer of F w.r.t. f, we can construct a recursive function σ such that $\sigma\alpha = \widehat{F(\alpha)}$ for all $\alpha \in \mathbb{N}^\mathbb{N}$. This yields:

$$\sigma\alpha(0) = \widehat{F(\alpha)}(0) = F(\alpha) = \begin{cases} 0 & \text{if } \alpha \text{ is recursive;} \\ 1 & \text{otherwise,} \end{cases}$$

for all $\alpha \in \mathbb{N}^{\mathbb{N}}$. In particular, we have $\sigma\hat{0}(0) = 0$. This implies that there exists an $N \in \mathbb{N}$ such that $\sigma\alpha(0) = 0$ for all α that agree with $\hat{0}$ on the inputs $0, 1, \ldots, N-1$. In other words, whenever $\alpha(n) = 0$ for all $n < N$, we have that $\sigma\alpha(0) = 0$, and thus that α is recursive. This is clearly a contradiction, so we can conclude that F is not representable w.r.t. f.

5.1.2 Adjoining a function to a PCA

The goal of this section is to define, for $\alpha \in \mathcal{B}A$, a PCA $A[\alpha]$ with a similar universal property as in Example 5.1.9 above. More precisely, partial applicative morphisms $A \multimap B$ with respect to which α is representable should correspond to partial applicative morphisms $A[\alpha] \multimap B$. The fact that our PCAs are relative and ordered presents a few challenges. As we already mentioned, we can only construct $A[\alpha]$ for partial functions α that behave well w.r.t. the order on A, namely those α that are in $\mathcal{B}A$. Moreover, due to the fact that A is ordered, we need to be slightly more careful when defining the application map on $A[\alpha]$; see Remark 5.1.12 below. For dealing with relativity, the notion of a generated filter will once again be useful.

In order for the construction of $A[\alpha]$ to work, we need to assume that A is not semitrivial. In fact, from this point onwards, we will assume that all PCAs we consider are not semitrivial. The underlying set of $A[\alpha]$ will simply be A itself, and the order on $A[\alpha]$ will simply be the order on A. However, we equip $A[\alpha]$ with a new application operation. Informally, a computation in $A[\alpha]$ will be a computation in A with an oracle for α. That is, the computation can feed a finite number of inputs to α before coming up with the final result. In order to distinguish this new application from the original one, we will write it as $a \odot b$, or $a \odot_\alpha b$ if α is not clear from the context.

Definition 5.1.11. *Let A be a PCA and let $\alpha \in \mathcal{B}A$. We define the PAP $A[\alpha] = (A, \odot, \leq)$ as follows. For $a, b, c \in A$, we say that $a \odot b = c$ if and only if there exists a (possibly empty) sequence $u_0, \ldots, u_{n-1} \in A$ such that:*

- *for all $i < n$, we have:*

$$\mathsf{p}_0(a \cdot [b, u_0, \ldots, u_{i-1}]) \leq \bot \quad \text{and} \quad \alpha(\mathsf{p}_1(a \cdot [b, u_0, \ldots, u_{i-1}])) = u_i;$$

- $\mathsf{p}_0(a \cdot [b, u_0, \ldots, u_{n-1}]) \leq \top$ *and* $\mathsf{p}_1(a \cdot [b, u_0, \ldots, u_{n-1}]) = c$.

*The sequence $u_0, \ldots, u_{n-1}$ is called a b-**interrogation** of α by a.*

Intuitively, the coefficients in the interrogation sequence are the values the oracle returns in the course of the computation of $a \odot b$. At each stage of the computation, the algorithm a is allowed to consult the input b and the values $u_0, \ldots, u_{i-1}$ obtained from the oracle so far. Formally, this means that we let a act on the coded sequence $[b, u_0, \ldots, u_{i-1}]$. We view the result as carrying two pieces of information. The first piece is a boolean, which tells us whether the computation has gathered enough oracle values to output a result. If not, then the second piece of information is fed to the oracle; if the oracle need not be consulted anymore, then this second piece is the output.

Since A is not semitrivial, there is at most one b-interrogation of α by a, which also means that $a \odot b = c$ for at most one $c \in A$. Observe that $a \odot b$ may fail to be defined in several ways. First of all, one of the applications in A could be undefined. In addition, $\mathsf{p}_0(a \cdot [b, u_0, \ldots, u_{i-1}])$ could fail to lie below either $\top$ or $\bot$, or $\mathsf{p}_1(a \cdot [b, u_0, \ldots, u_{i-1}])$ could lie outside the domain of α (i.e., the oracle fails to return a value). Finally, it could happen that the computation keeps feeding inputs to the oracle indefinitely, never coming up with a final output. For example, if $a = \mathsf{k}(\mathsf{p}\bot\bot)$, then $a \odot b$ will always be undefined, even if A itself is a total PCA and α is a total function.

Remark 5.1.12. In the original definition of $a \odot b$ from [vO06, Theorem 2.2], which is meant for discrete PCAs, the sequence $u_0, \ldots, u_{n-1}$ should satisfy:

- for all $i < n$, there exists a $v_i \in A$ such that $a \cdot [b, u_0, \ldots, u_{i-1}] = \mathsf{p}\bot v_i$ and $\alpha(v_i) = u_i$;
- there exists a $c \in A$ such that $a \cdot [b, u_0, \ldots, u_{n-1}] = \mathsf{p}\top c$,

in which case $a \odot b = c$. Since we are working with ordered PCAs, however, we cannot hope to get *equalities* between elements from A, since all the available combinators only yield inequalities. We do have the following: if there are $\bar{u}_0, \ldots, \bar{u}_{n-1}, c \in A$ such that:

- for all $i < n$, there exists a $v_i \in A$ such that $a \cdot [b, \bar{u}_0, \ldots, \bar{u}_{i-1}] \leq \mathsf{p}\bot v_i$ and $\alpha(v_i) \leq \bar{u}_i$;
- $a \cdot [b, \bar{u}_0, \ldots, \bar{u}_{n-1}] \leq \mathsf{p}\top c$,

then $a \odot b$ is defined and $a \odot b \leq c$. (We write $\bar{u}_i$ rather than u_i because this sequence need not be the actual b-interrogation of α by a.)

Of course, we should show that $A[\alpha]$ is actually a PAP, which is also one of the points where we need that $\alpha \in \mathcal{B}A$. Suppose we have $a' \leq a$ and $b' \leq b$ such that $a \odot b\downarrow$, and let $u_0, \ldots, u_{n-1}$ be the b-interrogation of α by a. Then by induction, one easily shows that there exist $u'_i \leq u_i$ such that $u'_0, \ldots, u'_{n-1}$ is a b'-interrogation of α by a'; and from this, we get that $a' \odot b'$ is defined and $a' \odot b' \leq a \odot b$, as desired.

In order to complete the definition of $A[\alpha]$ as a PAS, it remains to define $A[\alpha]^\#$. Note that we cannot simply take $A[\alpha]^\# = A^\#$, since $A^\#$ could fail to be closed under (defined) $\odot$. The following definition remedies this.

Definition 5.1.13. *Let A be a PCA and let $\alpha \in \mathcal{B}A$. The PAP $A[\alpha]$ is made into a PAS by setting $A[\alpha]^\# := \langle A^\# \rangle$, where the generated filter is taken in the PAP $A[\alpha]$, rather than A.*

This, by definition, makes $A[\alpha]$ into a PAS, which we now show to be a PCA. The construction of the required k- and s-combinators for $A[\alpha]$ is the same as in [vO06, Theorem 2.2], but adapted to the ordered setting.

Proposition 5.1.14. *For each PCA A and $\alpha \in \mathcal{B}A$, the quadruple $A[\alpha] = (A, A[\alpha]^{\#}, \odot, \leq)$ is a PCA.*

Proof. We need to exhibit suitable combinators k_α and s_α for $A[\alpha]$. Let us define $\mathsf{fst} \in A^{\#}$ as $\lambda^* x.\mathsf{read} \cdot x \cdot \overline{0}$, where read is as in Construction 2.1.32. Note that $\mathsf{fst} \cdot [a_0, \ldots, a_n] \leq a_0$ for a coded sequence $[a_0, \ldots, a_n]$. As k_α, we can take

$$\lambda^* x.\mathsf{p}\top(\lambda^* y.\mathsf{p}\top(\mathsf{fst}x)) \in A^{\#} \subseteq A[\alpha]^{\#}.$$

Indeed, if $a, b \in A$, then $\mathsf{k}_\alpha \odot a \leq (\lambda^* y.\mathsf{p}\top(\mathsf{fst}x))[[a]/x]$, and

$$(\mathsf{k}_\alpha \odot a) \cdot [b] \preceq (\mathsf{p}\top(\mathsf{fst}x))[[a]/x, [b]/y] = \mathsf{p}\top(\mathsf{fst}[a]) \leq \mathsf{p}\top a,$$

which means that $\mathsf{k}_\alpha \odot a \odot b \leq a$, as desired. Observe that k_α does not, in fact, depend on α, and that the computation of $\mathsf{k}_\alpha \odot a \odot b$ does not consult the oracle at all.

The definition of s_α (which will also not depend on α) is a little more involved, and defining it explicilty using the combinators from $A^{\#}$ would merely obscure the main idea of the construction. Therefore, we informally describe how to construct s_α. Using the fixpoint operator from $A^{\#}$, elementary operations on sequences and numerals, and the $\mathsf{if} \ldots \mathsf{then} \ldots \mathsf{else} \ldots$ construction, we may construct an element $S \in A^{\#}$ such that for all a, b and $u = [u_0, \ldots, u_n]$ from A, we have that $Sab\downarrow$, and:

- If $\mathsf{p}_0(a \cdot [u_0, \ldots, u_i]) \leq \bot$ for all $i \leq n$, then $Sabu \preceq au$;
- Suppose there is a least $i \leq n$ such that $\mathsf{p}_0(a \cdot [u_0, \ldots, u_i]) \leq \top$, and set $d := \mathsf{p}_1(a \cdot [u_0, \ldots, u_i])$.
  - If $\mathsf{p}_0(b \cdot [u_0, u_{i+1}, \ldots, u_j]) \leq \bot$ for all j with $i \leq j \leq n$, then:
  $$Sabu \preceq b \cdot [u_0, u_{i+1}, \ldots, u_n].$$
  - Suppose there is a least j such that $\mathsf{p}_0(b \cdot [u_0, u_{i+1}, \ldots, u_j]) \leq \top$ and $i \leq j \leq n$, and set $e := \mathsf{p}_1(b \cdot [u_0, u_{i+1}, \ldots, u_j])$. Then:
  $$Sabu \preceq d \cdot [e, u_{j+1}, \ldots, u_n].$$

Even more informally, the computation $Sabu$ does the following. First, we check whether, for some i, the sequence $u_1, \ldots, u_i$ is a u_0-interrogation of α by a. However, since we are working in A, the oracle α is not available, so we have to believe 'on faith' that each next value of the sequence u is the correct value returned by the oracle α. If the us run out before we find such an i, then the next query to α that the computation of $a \odot u_0$ was supposed to make, is our output. If we *do* find such an i, then we start looking for a $j \geq i$ such that $u_{i+1}, \ldots, u_j$ is a u_0-interrogation of α by b. Again, if the us run out before we find such a j, then the next query to α in the computation of $b \odot u_0$ is our output. If we *do* find such a j, then we know $a \odot u_0$ and $b \odot u_0$, and we use the remaining part of the sequence u to mimick the computation of $a \odot u_0 \odot (b \odot u_0)$.

Suppose we have $a, b, c \in A$ such that $a \odot c$, $b \odot c$ and $a \odot c \odot (b \odot c)$ are all defined. Then from the informal description of S given above, it follows that $Sab \odot c$ is also defined, and $Sab \odot c \le a \odot c \odot (b \odot c)$. Now we can set

$$\mathsf{s}_\alpha = \lambda^* x.\mathsf{p}\top(\lambda^* y.\mathsf{p}\top(S(\mathsf{fst}x)(\mathsf{fst}y))) \in A^\# \subseteq A[\alpha]^\#.$$

In the same way as we did for k_α, one can verify that $\mathsf{s}_\alpha \odot a \odot b \le Sab$, so we can conclude that $A[\alpha]$ is a PCA. □

Similarly to Example 2.2.15, we define $\iota_\alpha \colon A \to A[\alpha]$ as the identity on A.

Proposition 5.1.15. *The map ι_α is a morphism of PCAs $A \to A[\alpha]$, and α is representable w.r.t. ι_α. Moreover, ι_α has a right adjoint $g \colon A[\alpha] \multimap A$ in OPCA_T satisfying $\iota_\alpha g \simeq \mathrm{id}_{A[\alpha]}$. In particular, ι_α is dense.*

Proof. Since $A^\# \subseteq A[\alpha]^\#$, it is clear that ι_α satisfies the first requirement from Definition 2.2.1, and it is also obvious that ι_α preserves the order on the nose. If we define

$$t := \lambda^* x.\mathsf{p}\top(\lambda^* y.\mathsf{p}\top(\mathsf{fst}x(\mathsf{fst}y))),$$

where fst is as above, then an easy calculation shows that $t \odot a \odot b \preceq ab$. Since $t \in A^\# \subseteq A[\alpha]^\#$, this completes the proof that ι_α is a morphism of PCAs.

Now let

$$r := \lambda^* x.\mathsf{if}\ \mathsf{zero}(\mathsf{pred}(\mathsf{lh}x))\ \mathsf{then}\ \mathsf{p}\bot(\mathsf{fst}x)\ \mathsf{else}\ \mathsf{p}\top(\mathsf{read}x1).$$

Then $r \in A^\# \subseteq A[\alpha]^\#$, and for $a \in \mathrm{dom}\,\alpha$, we have:

- $r \cdot [a] \le \mathsf{p}\bot a$ and $\alpha(a)\downarrow$;
- $r \cdot [a, \alpha(a)] \le \mathsf{p}\top\alpha(a)$,

which means that $r \odot a \le \alpha(a)$. We conclude that r_α represents α w.r.t. ι_α.

We define the required right adjoint $g \colon A[\alpha] \multimap A$ by:

$$g(a) := \{b \in A \mid b \odot \mathsf{i} \le a\},$$

where $\mathsf{i} \in A^\#$ is the identity combinator for A. It is clear that $g(a)$ is a downset, and that g preserves the order on the nose. Now consider the $S \in A^\#$ constructed in the proof of Proposition 5.1.14. If $b \in g(a)$ and $b' \in g(a')$, then $Sbb' \odot \mathsf{i} \preceq (b \odot \mathsf{i}) \odot (b' \odot \mathsf{i}) \preceq a \odot a'$, so if $a \odot a' \downarrow$, then $Sbb' \in g(a \odot a')$. In other words, g preserves application up to $S \in A^\#$. In order to see that g is a partial applicative morphism, Lemma 2.2.5 tells us that it suffices to show that $A^\# \subseteq \mathrm{dom}^\# g$. But this follows immediately from the observation that $\mathsf{k}(\mathsf{p}\top a) \in g(a)$ for all $a \in A$, which also shows that g is total.

We also see that $\lambda^* x.\mathsf{k}(\mathsf{p}\top x) \in A^\#$ realizes $\mathrm{id}_A \le g\iota_\alpha$. Finally, $\mathrm{id}_{A[\alpha]} \le \iota_\alpha g$ and $\iota_\alpha g \le \mathrm{id}_{A[\alpha]}$ are realized by k_α and $\lambda^* x.x \odot \mathsf{i}$, respectively. □

Before we proceed to establish the universal property of ι_α, let us make a few remarks.

Remark 5.1.16. (i) The PCA $A[\alpha]$ is always not semitrivial, given that A is not semitrivial. Indeed, if $A[\alpha]$ were semitrivial, then any two elements of $A[\alpha]$ would have a lower bound. But as a poset, $A[\alpha]$ is simply A, so every two elements of A would have a lower bound, which we assumed not to be the case.

(ii) In general, the filter $A[\alpha]^\#$ is strictly larger than $A^\#$. This makes sense given the intuition that $A^\#$ consists of those elements of A that are actually computable (or can be refined into such a computable element). In $A[\alpha]$, the function α is computable, so if a certain $a \in A$ is computable, then $\alpha(a)$ should be computable as well. However, we have not required that $A^\#$ is closed under defined application of α, so making α computable will also force the set of 'computable elements' to become larger. In fact, $A[\alpha]$ can alternatively be described as the least filter on the PCA A that is closed under defined application of α; we leave the proof to the reader.

(iii) Since ι_α is dense, it is also decidable and c.d. Note that the computational density of ι_α is not *a priori* obvious, since $A[\alpha]^\#$ can be larger than $A^\#$.

(iv) In analogy with Example 3.4.16, we see that $\mathsf{RT}(A[\alpha])$ is a subtopos of $\mathsf{RT}(A)$.

Now we are ready to prove the universal property of $\iota_\alpha \colon A \to A[\alpha]$. The main part of the proof is adapted from [vO06, Theorem 2.2]. In the proof of the following theorem, we will use combinators $\mathsf{ext}, \mathsf{unit} \in A^\#$ with the following properties: $\mathsf{ext} \cdot [a_0, \ldots, a_{n-1}] \cdot a' \leq [a_0, \ldots, a_{n-1}, a']$ and $\mathsf{unit} \cdot a \leq [a]$.

Theorem 5.1.17. *Let $f \colon A \to B$ be a decidable morphism of PCAs, and let $\alpha \in \mathcal{B}A$. Then f factors through ι_α if and only if α is representable w.r.t. f.*

Proof. The 'only if' statement is clear, given Lemma 5.1.4(i) and Proposition 5.1.15. For the converse, we need to show that, if α is representable w.r.t. f, then f is also a morphism of PCAs $A[\alpha] \to B$.

Let $t, u \in B^\#$ realize $f \colon A \to B$, let $d \in B^\#$ be a decider for f, and let $s \in B^\#$ represent α w.r.t. f. For $i = 0, 1$, we define $\mathsf{p}'_i = \lambda^* x.u(t \cdot f(\mathsf{p}_i) \cdot x) \in B^\#$, so that: if $\mathsf{p}_i a \leq a'$, then $\mathsf{p}'_i \cdot f(a) \leq f(a')$. Similarly, we may define an element $\mathsf{unit}' \in B^\#$ with the property that $\mathsf{unit}' \cdot f(a) \leq f([a])$. Since ext is intended to take two arguments, we set:

$$\mathsf{ext}' := \lambda^* xy.u(t(t \cdot f(\mathsf{ext}) \cdot x)y) \in B^\#,$$

so that $\mathsf{ext}' \cdot f([a_0, \ldots, a_{n-1}]) \cdot f(a') \leq f([a_0, \ldots, a_{n-1}, a'])$.

Using the fixed point operator in $B^\#$, we may construct an element $T \in B^\#$ satisfying:

$$Tbv \preceq \text{if } d(\mathsf{p}'_0(tbv)) \text{ then } \mathsf{p}'_1(tbv) \text{ else } Tb\big(\mathsf{ext}'v(s(\mathsf{p}'_1 \cdot (tbv)))\big). \tag{5.1}$$

Suppose that $a, a' \in A$ are such that $a \odot a' \downarrow$, and let $u_0, \ldots, u_{n-1}$ be the a'-interrogation of α by a. First of all, we claim that

$$T \cdot f(a) \cdot f([a', u_0, \ldots, u_{i-1}]) \preceq T \cdot f(a) \cdot f([a', u_0, \ldots, u_i]) \tag{5.2}$$

for all $i < n$. Indeed, apply (5.1) with $b = f(a)$ and $v = f([a', u_0, \dots, u_{i-1}])$. We have $tbv \leq f(a \cdot [a', u_0, \dots, u_{i-1}])$, and since $\mathsf{p}_0(a \cdot [a', u_0, \dots, u_{i-1}]) \leq \bot$, this yields $\mathsf{p}'_0(tbv) \leq f(\bot)$, hence $d(\mathsf{p}'_0(tbv)) \leq \bot$. This means that the 'else' clause of (5.1) applies, in other words, we have $Tbv \preceq Tb\big(\mathsf{ext}'v(s(\mathsf{p}'_1 \cdot (tbv)))\big)$. In order to evaluate the latter, we note that $\mathsf{p}'_1(tbv) \leq f(\mathsf{p}_1(a \cdot [a', u_0, \dots, u_{i-1}]))$. Since s represents α, this yields $s(\mathsf{p}'_1(tbv)) \leq f(\alpha(\mathsf{p}_1(a \cdot [a', u_0, \dots, u_{i-1}]))) = f(u_i)$. This implies that $Tbv \preceq Tb\big(\mathsf{ext}'v \cdot f(u_i)\big) \preceq Tb \cdot f([a', u_0, \dots, u_{i-1}, u_i]))$, as desired.

Moreover, we have:

$$T \cdot f(a) \cdot f([a', u_0, \dots, u_{n-1}]) \leq f(a \odot a'). \tag{5.3}$$

Indeed, if we employ (5.1) with $b = f(a)$ and $v = f([a', u_0, \dots, u_{n-1}])$, then this time we have $d(\mathsf{p}'_0(tbv)) \leq \top$. This means that the 'then' clause of (5.1) applies, so $Tbv \preceq \mathsf{p}'_1 \cdot (tbv) \leq f(\mathsf{p}_1(a \cdot [a', u_0, \dots, u_{n-1}])) = f(a \odot a')$.

Combining (5.2) and (5.3), we get

$$T \cdot f(a) \cdot f([a']) \preceq f(a \odot a'),$$

for all $a, a' \in A$. Now it follows $f \colon A[\alpha] \to B$ preserves application up to $\lambda^* xy.Tx(\mathsf{unit}'y) \in B^\#$.

Clearly, $f \colon A[\alpha] \to B$ also preserves the order up to $u \in B^\#$. Since f is a morphism $A \to B$, we have $f(a) \in B^\#$ for all $a \in A^\#$, and now Lemma 2.2.5 tells us that f is also a morphism $A[\alpha] \to B$, as desired. □

We can reformulate Theorem 5.1.17 in the following way. Let us write $\mathsf{OPCA}_{\mathrm{dec}}$ for the wide subcategory of OPCA consisting of only the decidable morphisms of PCAs. We define $\mathsf{OPCA}_{T,\mathrm{dec}}$ and $\mathsf{OPCA}_{D,\mathrm{dec}}$ similarly.

Corollary 5.1.18. *Let $\mathcal{C}$ be any of the categories $\mathsf{OPCA}_{\mathrm{dec}}$, $\mathsf{OPCA}_{T,\mathrm{dec}}$ and $\mathsf{OPCA}_{D,\mathrm{dec}}$. If A, B are PCAs and $\alpha \in \mathcal{B}A$, then composition with $\iota_\alpha \colon A \to A[\alpha]$ yields an isomorphism of preorders:*

$$\mathcal{C}(A[\alpha], B) \cong \{f \in \mathcal{C}(A, B) \mid \alpha \textit{ is representable w.r.t. } f\}.$$

Proof. For $\mathcal{C} = \mathsf{OPCA}_{\mathrm{dec}}$, we note that Lemma 2.2.10, Lemma 5.1.4(i) and Theorem 5.1.17 imply that composition with ι_α yields a well-defined bijection between the preorders from the theorem. Moreover, it is immediate that composition with ι_α preserves and reflects the order.

For $\mathsf{OPCA}_{T,\mathrm{dec}}$ and $\mathsf{OPCA}_{D,\mathrm{dec}}$, the result follows by applying the theorem for $\mathsf{OPCA}_{\mathrm{dec}}$ with TB resp. DB in the place of B. □

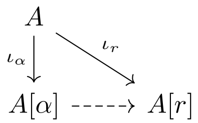
Remark 5.1.19. By Lemma 2.2.10, we see that Corollary 5.1.18 is still true if we restrict to c.d. morphisms, or - in the case of OPCA and OPCA_T - dense morphisms.

Example 5.1.20. In Remark 5.1.16(iii), we explained that it is not surprising that $A[\alpha]^\#$ is, in general, larger than $A^\#$ because 'adding' a new representable function α can also make more elements of A computable. Similarly, adding α as a representable function makes other functions besides α representable as well. By Corollary 5.1.18, a function β is representable w.r.t. ι_α iff the identity on A is a morphism of PCAs $A[\beta] \to A[\alpha]$.

Example 5.1.21. The construction of $A[\alpha]$ is a generalization of oracle computations for classical Turing computability. Indeed, if $f\colon \mathbb{N} \rightharpoonup \mathbb{N}$ is a partial function, then $g\colon \mathbb{N} \rightharpoonup \mathbb{N}$ is representable w.r.t. $\iota_f\colon \mathcal{K}_1 \to \mathcal{K}_1[f]$ iff g is Turing computable relative to an oracle for f. This implies that the identity on $\mathbb{N}$ is an isomorphism $\mathcal{K}_1[f] \cong \mathcal{K}_1^f$; see also Corollary 2.3 in [vO06].

Example 5.1.22. The construction of $A[\alpha]$ can be seen as a higher-order version of the construction of $A[a]$ from Example 2.2.15. On the other hand, the construction of $A[a]$ can be seen as a special case of the construction $A[\alpha]$. Indeed, consider $a \in A$ and denote the constant function with value a, which is an element of $\mathcal{B}A$, by $\hat{a}$. It is easy to see that, for any $f\colon A \multimap B$, we have that $\hat{a}$ is representable w.r.t. f iff $a \in \mathrm{dom}^{\#} f$. It follows that $A[a]$ and $A[\hat{a}]$ are isomorphic PCAs.

Example 5.1.23. Of course, if $\alpha \in \mathcal{B}A$ is already representable in A itself, then $\iota_\alpha\colon A \to A[\alpha]$ will be an isomorphism of PCAs. Now suppose that α is represented by an element $r \in A$ (but not necessarily $r \in A^{\#}$). Then α is representable w.r.t. $\iota_r\colon A \to A[r]$, so we get a factorization:



It is worth observing that the mediating arrow $A[\alpha] \to A[r]$ is *not*, in general, an isomorphism. Indeed, consider a PCA with $A^{\#} \neq A$, and take a $b \in A \backslash A^{\#}$. Then the partial function $\alpha \in \mathcal{B}A$ defined by

$$\alpha(a) = \begin{cases} \mathsf{p}_1 a & \text{if } \mathsf{p}_0 a \leq \top; \\ \text{undefined} & \text{else} \end{cases}$$

is representable, e.g., by means of $\mathsf{p}_1 \in A^{\#}$. This means that ι_α is an isomorphism. But α is *also* represented by $r := \lambda^* x.\mathsf{if}\ \mathsf{p}_0 x\ \mathsf{then}\ \mathsf{p}_1 x\ \mathsf{else}\ b$. Moreover, we have $r \notin A^{\#}$, because $r(\mathsf{p}\bot\bot) \leq b$, so $r \in A^{\#}$ would imply $b \in A^{\#}$. This means that ι_r is *not* an isomorphism, so $A[\alpha] \to A[r]$ cannot be an isomorphism either. We see that the point here is, really, that a function $\alpha \in \mathcal{B}A$ can have many representers.

Above, we have adjoined one function α to a PCA A, but of course, we can also adjoin multiple functions at once. This will be useful in the next section, so we give the construction explicitly. In fact, we can do this in several ways: we can apply the construction of $A[\alpha]$ repeatedly, we can code a finite sequence of functions into one funtion (cf. the remarks before Definition 2.1.33), or we can adjust the definition of $\odot$ directly. Here we choose the latter approach, because it is the least heavy on coding.

Construction 5.1.24. If $\vec{\alpha} = \alpha_1, \ldots, \alpha_\ell \in \mathcal{B}A$, then we define a new PCA $A[\vec{\alpha}] = (A, \odot = \odot_{\vec{\alpha}}, \leq, (A[\vec{\alpha}])^\#)$ as follows. If $a, b, c \in A$, then we say that $a \odot b = c$ iff there exists a b-interrogation of $\vec{\alpha}$ by a, that is, a sequence $u_0, \ldots, u_{n-1} \in A$, such that:

- for all $i < n$, there exists a $k \in \{1, \ldots, \ell\}$ such that:

$$\mathsf{p}_0(a \cdot [b, u_0, \ldots, u_{i-1}]) \leq \overline{k} \quad \text{and} \quad \alpha_k(\mathsf{p}_1(a \cdot [b, u_0, \ldots, u_{i-1}])) = u_i$$

- $\mathsf{p}_0(a \cdot [b, u_0, \ldots, u_{n-1}]) \leq \overline{0}$ and $\mathsf{p}_1(a \cdot [b, u_0, \ldots, u_{n-1}]) = c$.

Note that this application is well-defined if A is not semitrivial, because, as remarked in Construction 2.1.31, no two numerals in A have a common lower bound. Of course, we let $A[\vec{\alpha}]^\#$ be the filter generated by $A^\#$ under this new application. This makes $A[\vec{\alpha}]$ into a PCA, and the identity on A is a morphism of PCAs $\iota_{\vec{\alpha}} \colon A \to A[\vec{\alpha}]$ which is the 'universal solution' to making all of $\alpha_1, \ldots, \alpha_\ell$ representable. More explicitly, every decidable morphism of PCA $f \colon A \to B$ with respect to which $\alpha_1, \ldots, \alpha_k$ are representable, factors through $\iota_{\vec{\alpha}}$. The proof uses the classical fact (see, e.g., [Lon94, Proposition 2.4.16]) that every decidable morphism of PCAs also 'preserves numerals'.

For $\ell = 1$, this yields a slighty different definition of $A[\alpha]$, but of course, the two definitions yield isomorphic PCAs, since they have the same universal property. In the case $\ell = 0$, we have that $\vec{\alpha}$ is the empty sequence ϵ, and it is clear that $\iota_\epsilon \colon A \to A[\epsilon]$ is an isomorphism.

5.2 The PCA of partial functions

5.2.1 Construction of the PCA

In this section, we show how to turn the set $\mathcal{B}A$ from Definition 5.1.1 into a PCA. For the order, we take the order defined in Definition 2.1.4. In particular, the empty function is the largest element of $\mathcal{B}A$. It is worth noting that the order on $\mathcal{B}A$ is *not* discrete even if the order on A is. Indeed, if A is discrete, then $\mathcal{B}A$ consists of all partial functions $A \rightharpoonup A$, and the order is the *reverse* subfunction relation. In this case, the total functions are minimal elements of $\mathcal{B}A$. In the general case, the total functions in $\mathcal{B}A$ form a downwards closed set.

The application on $\mathcal{B}A$ will, in a sense, generalize the $A[\alpha]$ for $\alpha \in \mathcal{B}A$ all at once. As for the construction of $A[\alpha]$, we need the assumption that A is not semitrivial. An important thing to note about the application on $\mathcal{B}A$ is that it will be *total*.

Definition 5.2.1. *Let A be a PCA. We define the PAP $\mathcal{B}A = (\mathcal{B}A, \cdot, \leq)$ as follows. The order on $\mathcal{B}A$ is as in Definition 2.1.4. For $\alpha, \beta \in \mathcal{A}$ and $a, b \in A$, we say that $\alpha\beta(a) = b$ if and only if there are $u_0, \ldots, u_{n-1}$ such that:*

- *for all* $i < n$, *we have:*

$$\mathsf{p}_0 \cdot \alpha([a, u_0, \ldots, u_{i-1}]) \leq \bot \quad \textit{and} \quad \beta(\mathsf{p}_1 \cdot \alpha([a, u_0, \ldots, u_{i-1}])) = u_i;$$

- $\mathsf{p}_0 \cdot \alpha([a, u_0, \ldots, u_{n-1}]) \leq \top$ *and* $\mathsf{p}_1 \cdot \alpha([a, u_0, \ldots, u_{i-1}]) = b$.

The sequence $u_0, \ldots, u_{n-1}$ *is called an* a-***interrogation*** *of* β *by* α.

By the assumption that A is nontrivial, the sequence $u_0, \ldots, u_{n-1}$ is unique if it exists, so we see that $\alpha\beta(a) = b$ for at most one b, which means that $\alpha\beta$ is indeed a partial function $A \rightharpoonup A$. Table 5.1 compares the definition of $\alpha\beta(a)$ with the definition of $a \odot_\alpha b$ from Definition 5.1.11.

	$a \odot_\alpha b$	$\alpha\beta(a)$
Interrogator	a	α
Input	b	a
Oracle	α	β

Table 5.1: Application in $A[\alpha]$ versus $\mathcal{B}A$.

In particular, we see that the function $(x \mapsto ax)\cdot\alpha$ coincides with the function $x \mapsto a \odot_\alpha x$.

Of course, we need to check that $\alpha\beta$ is actually an element of $\mathcal{B}A$, and that the resulting application map on $\mathcal{B}A$ satisfies axiom (A) from Definition 2.1.1. This is similar to the verification of axiom (A) for $A[\alpha]$, and we leave it to the reader.

In order to make $\mathcal{B}A$ into a PAS, we will let $(\mathcal{B}A)^\#$ consist of all representable functions. In order for this to work, we must show that the set of representable functions is closed under the application map defined above. The proof will use the same technique as the proof of Theorem 5.1.17; in particular, we will employ the combinators ext and unit again.

Lemma 5.2.2. *Let* A *be a PCA. If* $\alpha, \beta \in \mathcal{B}A$ *are representable, then* $\alpha\beta \in \mathcal{B}A$ *is representable as well.*

Proof. Suppose that $r, s \in A^\#$ are such that $ra \preceq \alpha(a)$ and $sa \preceq \beta(a)$. Using the fixpoint operator in $A^\#$, we may construct a $T \in A^\#$ such that:

$$Tu \preceq \mathsf{if}\ \mathsf{p}_0(ru)\ \mathsf{then}\ \mathsf{p}_1(ru)\ \mathsf{else}\ T(\mathsf{ext} \cdot u \cdot (s(\mathsf{p}_1(ru)))).$$

Suppose $a \in A$ is such that $\alpha\beta(a)$ is defined, and let $u_0, \ldots, u_{n-1}$ be the a-interrogation of β by α. Then it easily follows that

$$T \cdot [a, u_0, \ldots, u_{i-1}] \preceq T \cdot [a, u_0, \ldots, u_i]$$

for $i < n$, and $T \cdot [a, u_0, \ldots, u_{n-1}] \leq \alpha\beta(a)$. Thus, we get $T \cdot [a] \preceq \alpha\beta(a)$ for all $a \in A$, which means that $\lambda^* x.T(\mathsf{unit} \cdot x) \in A^\#$ represents $\alpha\beta$. $\square$

Definition 5.2.3. *Let A be a PCA. We make $\mathcal{B}A$ into a PAS by letting $(\mathcal{B}A)^{\#}$ consist of all representable functions, that is:*

$$(\mathcal{B}A)^{\#} = \uparrow\{a \mapsto ra \mid r \in A^{\#}\}.$$

In order to show that $\mathcal{B}A$ is a PCA, it remains to construct the appropriate combinators. In [vO11], this is shown by giving an alternative characterization of representable functions $\mathcal{B}A \to \mathcal{B}A$, using trees. Here, we will prove that $\mathcal{B}A$ is a PCA using the argument that $\mathcal{B}$ is a PCA given in [vO08, Section 1.4.5]. Recall from Construction 5.1.24 the morphism of PCAs $\iota_{\vec{\alpha}} \colon A \to A[\vec{\alpha}]$. We need the following two rather technical lemmata. The first lemma says that, given representers of β and γ w.r.t. $\iota_{\vec{\alpha}}$, we can compute a representer of $\beta\gamma$ w.r.t. $\iota_{\vec{\alpha}}$, and we can do this *uniformly* in the αs.

Lemma 5.2.4. *Let $\ell \geq 0$ be a natural number. There exists an element $t_\ell \in A^{\#}$ such that, for all $\vec{\alpha} = \alpha_1, \ldots, \alpha_\ell \in \mathcal{B}A$ and $\beta, \gamma \in \mathcal{B}A$, we have:*

$$t_\ell \cdot \mathrm{rep}^{\iota_{\vec{\alpha}}}(\beta) \cdot \mathrm{rep}^{\iota_{\vec{\alpha}}}(\gamma) \subseteq \mathrm{rep}^{\iota_{\vec{\alpha}}}(\beta\gamma).$$

Proof. The construction of t_ℓ is similar to the construction of S in the proof of Proposition 5.1.14. For r, s and $u = [u_0, \ldots, u_n]$, we should have that $t_\ell rs$ is defined, and the computation of $t_\ell rsu$ does the following. In the course of the computation, we construct natural numbers i_p and j_p, and elements $v_p \in A$ for $p \geq 0$. Initially, we set $p = 0$ and $i_0 = 0$. Then we repeatedly run through the following loop.

- For $i_p \leq j \leq n$, let us write $w_j = [[u_0, v_0, \ldots, v_{p-1}], u_{i_p+1}, \ldots, u_j]$. If for all j with $i_p \leq j \leq n$, we have that $\mathsf{p}_0(rw_j) \leq \overline{k}$ for some $k > 0$, then:

$$t_\ell rsu \preceq rw_n \simeq r \cdot [[u_0, v_0, \ldots, v_{p-1}], u_{i_p+1}, \ldots, u_n].$$

  Suppose, on the other hand, that there is a least j_p such that $i_p \leq j_p \leq n$ and $\mathsf{p}_0(rw_{j_p}) \leq \overline{0}$. Set $x := \mathsf{p}_1(rw_{j_p})$.

  - If $\mathsf{p}_0 x \leq \top$, then $t_\ell rsu \preceq \mathsf{p}\overline{0}(\mathsf{p}_1 x)$.
  - Suppose that $\mathsf{p}_0 x \leq \bot$. If, for all i with $j_p \leq i \leq n$, we have $\mathsf{p}_0(s \cdot [\mathsf{p}_1 x, u_{j_p+1}, \ldots, u_i]) \leq \overline{k}$ for some $k > 0$, then:

$$t_\ell rsu \preceq s \cdot [\mathsf{p}_1 x, u_{j_p+1}, \ldots, u_n].$$

    Suppose, on the other hand, that there is a least i_{p+1} such that $j_p \leq i_{p+1} \leq n$ and $\mathsf{p}_0(s \cdot [\mathsf{p}_1 x, u_{j_p+1}, \ldots, u_{i_{p+1}}]) \leq \overline{0}$. Then set $v_p = \mathsf{p}_1(s \cdot [\mathsf{p}_1 x, u_{j_p+1}, \ldots, u_{i_{p+1}}])$, increase p by 1, and restart the loop.

Now, if $r, s \in A^{\#}$ represent β resp. γ, and $\beta\gamma(a)$ is defined, then it follows that $t_\ell rs \odot_{\vec{\alpha}} a$ is defined as well, and $t_\ell rs \odot_{\vec{\alpha}} a \leq \beta\gamma(a)$. Thus, we have $t_\ell rs \in \mathrm{rep}^{\iota_{\vec{\alpha}}}(\beta\gamma)$, as desired. □

The following lemma says that, if we can compute a certain function β relative to oracles $\alpha_1, \dots, \alpha_{\ell+1}$, then we can effectively find a new function β', computable relative to $\alpha_1, \dots, \alpha_\ell$, such that $\beta'\alpha_{\ell+1} \leq \beta$; and this can be done uniformly in the αs. Since $\beta'\alpha_{\ell+1}$ is, in effect, a computation with an oracle for $\alpha_{\ell+1}$, this is simply a matter of rearranging the oracle calls appropriately. Thus, while the proof of the following lemma is again rather technical, the idea of the construction is quite simple.

Lemma 5.2.5. *Let $\ell \geq 0$ be a natural number. There exists an element $s_\ell \in A^\#$ with the following property: if $\vec{\alpha} = \alpha_1, \dots, \alpha_{\ell+1} \in \mathcal{B}A$ and $r \in A$, then $s_\ell r\downarrow$ and:*

$$(x \mapsto s_\ell r \odot_{\alpha_1,\dots,\alpha_\ell} x) \cdot \alpha_{\ell+1} \leq (x \mapsto r \odot_{\vec{\alpha}} x).$$

Proof. Again, we give an informal description of s_ℓ. Let r and $u = [u_0, \dots, u_n]$ be elements of A, where u_0 is itself a coded sequence $[v_0, \dots, v_m]$. We should have that $s_\ell r$ is defined, and that the computation of $s_\ell r u$ does the following. In the course of the computation, we construct elements w_p of A for $p \geq 0$. Initially, we set $p = 0$ and $i = j = 1$. Then repeatedly run through the following loop.

Let $x := r \cdot [v_0, w_0, \dots, w_{p-1}]$.

- If $\mathsf{p}_0 x \leq \overline{0}$, then $s_\ell r u \preceq \mathsf{p}\overline{0}(\mathsf{p}\top(\mathsf{p}_1 x))$.
- Suppose $\mathsf{p}_0 x \leq \overline{k}$ for some $k \in \{1, \dots, \ell\}$. If $i \leq n$, set $w_p = u_i$, increase both p and i by 1, and restart the loop. If $i > n$, then $s_\ell r u \leq x$.
- Suppose that $\mathsf{p}_0 x \leq \overline{\ell + 1}$. If $j \leq m$, set $w_p = v_j$, increase both p and j by 1, and restart the loop. If $j > m$, then $s_\ell r u \preceq \mathsf{p}\overline{0}(\mathsf{p}\bot(\mathsf{p}_1 x))$.

If $a \in A$ is such that $r \odot_{\vec{\alpha}} a$ is defined, then $\big((x \mapsto s_\ell r \odot_{\alpha_1,\dots,\alpha_\ell} x) \cdot \alpha_{\ell+1}\big)(a)$ will be defined as well, and $\big((x \mapsto s_\ell r \odot_{\alpha_1,\dots,\alpha_\ell} x) \cdot \alpha_{\ell+1}\big)(a) \leq r \odot_{\vec{\alpha}} a$, as desired. □

Remark 5.2.6. If $\alpha_1, \dots, \alpha_{\ell+1} \in \mathcal{B}A$, then the identity on A is an isomorphism f of PCAs as in the following diagram:

$$\begin{array}{ccc} A & \xrightarrow{\iota} & A[\alpha_1, \dots, \alpha_{\ell+1}] \\ {\scriptstyle \iota}\downarrow & & \downarrow{\scriptstyle f} \\ A[\alpha_1, \dots, \alpha_\ell] & \xrightarrow{\iota} & A[\alpha_1, \dots, \alpha_\ell][\alpha_{\ell+1}] \end{array}$$

This follows by observing that $A[\alpha_1, \dots, \alpha_{\ell+1}]$ and $A[\alpha_1, \dots, \alpha_\ell][\alpha_{\ell+1}]$ have the same universal property, namely, that they make all of $\alpha_1, \dots, \alpha_{\ell+1}$ representable. By the remark below Table 5.1, we see that the existence of s_ℓ as in Lemma 5.2.5 implies in particular that f preserves application up to a realizer. But of course, Lemma 5.2.5 says much more than that: the crucial point of the lemma is that s_ℓ does not depend on the αs.

Proposition 5.2.7. *If A is a non-semitrivial PCA, then $\mathcal{B}A$ is also a PCA.*

Proof. Let $t = t(x_1, \ldots, x_\ell)$ be a pure term in ℓ variables. Then there exists an $r \in A^\#$ such that r represents $t(\vec{\alpha})$ w.r.t. $\iota_{\vec{\alpha}}$, for all $\vec{\alpha} = \alpha_1, \ldots, \alpha_\ell \in \mathcal{B}A$. This follows by induction on t; the base case is similar to the construction of r in the proof of Proposition 5.1.15, and the induction step follows by Lemma 5.2.4.

Now let $s_0, \ldots s_{\ell-1} \in A^\#$ be as in Lemma 5.2.5. Define the sequence $r_\ell, r_{\ell-1}, \ldots, r_1, r_0 \in A^\#$ by $r_\ell = r$ and $r_i = s_i r_{i+1}$ for $i < \ell$. Then:

$$\begin{aligned} t(\alpha_1, \ldots, \alpha_\ell) &\geq (x \mapsto r_\ell \odot_{\alpha_1, \ldots, \alpha_\ell} x) \\ &\geq (x \mapsto r_{\ell-1} \odot_{\alpha_1, \ldots, \alpha_{\ell-1}} x) \cdot \alpha_\ell \\ &\geq \cdots \\ &\geq (x \mapsto r_1 \odot_{\alpha_1} x) \cdot \alpha_2 \cdots \alpha_\ell \\ &\geq (x \mapsto r_0 \odot_\epsilon x) \cdot \alpha_1 \cdots \alpha_\ell. \end{aligned}$$

Since $r_0 \in A^\#$ and $\iota_\epsilon \colon A \to A[\epsilon]$ is an isomorphism, the function $\rho \in \mathcal{B}A$ defined by $\rho(x) \simeq r_0 \odot_\epsilon x$ is in $(\mathcal{B}A)^\#$. Thus, we see that, for each term $t(\vec{x})$, there exists a $\rho \in (\mathcal{B}A)^\#$ such that $\rho\vec{\alpha} \leq t(\vec{\alpha})$ for all $\vec{\alpha} \in \mathcal{B}A$, which shows that $\mathcal{B}A$ is indeed a PCA. ☐

Remark 5.2.8. (i) Note that the PCA $\mathcal{B}A$ is not semitrivial, given that A is not semitrivial. Indeed, if $\mathcal{B}A$ were semitrivial, then the two constant functions with values $\top$ and $\bot$ would have a lower bound $\alpha \in \mathcal{B}A$. But then for any $a \in A$, we get that $\alpha(a)$ is a lower bound of $\top$ and $\bot$, which cannot be true.

(ii) Let us remark that $\mathcal{B}\mathcal{K}_1$ is not quite the Van Oosten PCA $\mathcal{B} = (\mathcal{B}, \mathcal{B}^{\mathrm{pr}}, \cdot, =)$ from Example 2.1.38, but rather its (relative and) ordered version mentioned at the end of Example 2.1.38.

(iii) The paper [vO11] also considers a generalization of Kleene's second model to arbitrary PCAs. It gives a construction that yields, for each absolute discrete PCA A, a new absolute discrete PCA $\mathcal{K}_2 A$ consisting of all total functions $A \to A$. This construction does not seem to have a 'relative' version, where $(\mathcal{K}_2 A)^\#$ consists of the total representable functions. For each term t, we can certainly construct a $\rho \in (\mathcal{B}A)^\#$ such that $\rho\vec{\alpha} \preceq t(\vec{\alpha})$ as above. However, such a ρ is not necessarily total. We can extend ρ to a total function by assigning a default value on inputs outside the domain of ρ, but of course, this may destroy the representability of ρ.

5.2.2 Universal property

Now that we established $\mathcal{B}A$ to be a relative ordered PCA, we will prove some of its important properties. In particular, we will show that $\mathcal{B}A$ has a certain (lax) universal property (Theorem 5.2.10). It is worth noting that almost none of the results in this section are true if we view $\mathcal{B}A$ as an absolute discrete PCA!

For $a \in A$, let $\hat{a} \in \mathcal{B}A$ denote the constant function with value a.

Proposition 5.2.9. *There is a c.d. morphism of PCAs $i\colon A \to \mathcal{B}A$, defined by $i(a) = \hat{a}$.*

Proof. If $a \in A^{\#}$, then $\hat{a}$ is clearly represented by $\mathsf{k}a \in A^{\#}$, so $\hat{a} \in (\mathcal{B}A)^{\#}$, meaning that i satisfies the first requirement from Definition 2.2.1. Moreover, i clearly preserves the order on the nose, so it remains to show that i preserves application up to a realizer. Using the elementary operations on sequences, we may contruct a $t \in A^{\#}$ satisfying:

- $t \cdot [[x]] \leq \mathsf{p}\top(\mathsf{p}\bot\mathsf{i})$;
- $t \cdot [[x, y]] \leq \mathsf{p}\bot\mathsf{i}$;
- $t \cdot [[x, y], z] \preceq \mathsf{p}\top(\mathsf{p}\top(zy))$.

If $\tau \in (\mathcal{B}A)^{\#}$ is defined by $\tau(x) \simeq tx$, then it is straightforward to check that $\tau\hat{a}\hat{b}(c) \preceq ab$ for all $a, b, c \in A$, so i preserves application up to τ.

For computational density, we take an $n \in A^{\#}$ satisfying:

- $n \cdot [x] \leq \mathsf{p}\bot\mathsf{i}$;
- $n \cdot [x, y] \preceq \mathsf{p}\top(yx)$.

If $\nu \in (\mathcal{B}A)^{\#}$ is defined by $\nu(a) \simeq na$, then it is easy to check that $\nu\hat{r}(a) \preceq ra$ for all $a, r \in A$. So, if $\rho \in (\mathcal{B}A)^{\#}$ is represented by $r \in A^{\#}$, then $\nu\hat{r} \leq \rho$, showing that ν satisfies (cd). ☐

Now we will prove the 'lax universal property' of $i\colon A \to \mathcal{B}A$. In [vO11, Theorem 5.1], this universal property states: if $f\colon A \multimap B$ is a decidable applicative morphism such that every $\alpha \in \mathcal{B}A$ is representable w.r.t. f, then f factors through i in a maximal way. Since we defined $\mathcal{B}A$ as a *relative* PCA, and since we have the notion of a *partial* applicative morphism, we can formulate the universal property of $\mathcal{B}A$ in a simpler way: every decidable partial applicative morphism $A \leftharpoonup\!\!\circ\, B$ factors through i in a maximal way.

Theorem 5.2.10. *Let $f\colon A \leftharpoonup\!\!\circ\, B$ be a decidable partial applicative morphism and consider the function $\mathrm{rep}^f : \mathcal{B}A \to DB$ from Definition 5.1.6.*

(i) rep^f is a partial applicative morphism $\mathcal{B}A \leftharpoonup\!\!\circ\, B$ that satisfies $\mathrm{rep}^f \circ i \simeq f$.

(ii) Moreover, rep^f is the largest such partial applicative morphism, i.e., if $g\colon \mathcal{B}A \leftharpoonup\!\!\circ\, B$ satisfies $gi \simeq f$, then $g \leq \mathrm{rep}^f$.

Proof. (i) If $\alpha \in (\mathcal{B}A)^{\#}$, then α is representable w.r.t. id_A, so by Lemma 5.1.4(i), α is also representable w.r.t. f, i.e., $\mathrm{rep}^f(\alpha) \cap B^{\#} \neq \emptyset$. Now assume, without loss of generality, that f preserves the order on the nose. Then it is readily seen that rep^f also preserves the order on the nose. Let f preserve application up to $t \in B^{\#}$, and let $d \in B^{\#}$ be a decider for f. For $j = 0, 1$, we define $\mathsf{p}'_j = tp_j \in B^{\#}$, where p_j is some element of $f(\mathsf{p}_j) \cap B^{\#}$, so that: if $\mathsf{p}_j a \leq a'$, then $\mathsf{p}'_j \cdot f(a) \subseteq f(a')$. Similarly, we may define $\mathsf{unit}' \in B^{\#}$ such that $\mathsf{unit}' \cdot f(a) \subseteq$

$f([a])$, and we set $\mathsf{ext}' = \lambda^* xy.t(tex)y \in B^\#$, where $e \in f(\mathsf{ext}) \cap B^\#$, so that $\mathsf{ext}' \cdot f([a_0, \ldots, a_{n-1}]) \cdot f(a') \subseteq f([a_0, \ldots, a_{n-1}, a'])$.

Using the fixpoint operator in $B^\#$, we may construct an element $T \in B^\#$ such that:

$$Txyv \preceq \text{ if } d(\mathsf{p}_0'(xv)) \text{ then } \mathsf{p}_1'(xv) \text{ else } Txy(\mathsf{ext}'v(y(\mathsf{p}_1'(xv)))).$$

If $\alpha, \beta \in \mathcal{B}A$ and $a \in A$ are such that $\alpha\beta(a)\downarrow$, then analogously to the proof of Theorem 5.1.17, we find that $T \cdot \mathrm{rep}^f(\alpha) \cdot \mathrm{rep}^f(\beta) \cdot f([a]) \subseteq f(\alpha\beta(a))$. We conclude that rep^f preserves application up to $\lambda^* xyz.Txy(\mathsf{unit}'z) \in B^\#$.

For $a \in A$, we have:

$$(\mathrm{rep}^f \circ i)(a) = \mathrm{rep}^f(\hat{a}) = \{b \in B \mid \forall a' \in A\,(b \cdot f(a') \subseteq f(a))\}.$$

Now it is easy to see that $\mathsf{k} \in B^\#$ realizes $f \leq \mathrm{rep}^f \circ i$, and if $j \in f(\mathsf{i}) \cap B^\#$, then $\lambda^* x.xj \in B^\#$ realizes $\mathrm{rep}^f \circ i \leq f$. So we indeed have $\mathrm{rep}^f \circ i \simeq f$.

(ii) Suppose we have $g\colon \mathcal{B}A \to B$ such that $gi \simeq f$. Assume, without loss of generality, that g preserves the order on the nose, and suppose that g preserves application up to $t' \in B^\#$. Moreover, let $r_0, r_1 \in B^\#$ realize $gi \leq f$ and $f \leq gi$ respectively. Now construct an element $p \in A^\#$ such that:

- $p \cdot [[x]] \leq \mathsf{p}\top(\mathsf{p}\bot\mathsf{i})$;
- $p \cdot [[x, y]] \leq \mathsf{p}\bot y$;
- $p \cdot [[x, y], z] \leq \mathsf{p}\top(\mathsf{p}\top z)$,

If $\rho \in (\mathcal{B}A)^\#$ is defined by $\rho(x) \simeq px$, then it is readily verified that $\rho\alpha\hat{a}(x) \preceq \alpha(a)$ for all $\alpha \in \mathcal{B}A$ and $a, x \in A$. In other words, we have $\rho\alpha\hat{a} \preceq \widehat{\alpha(a)}$. Let q be an element from $g(\rho) \cap B^\#$. Now we claim that the element $s := \lambda^* xy.r_0(t'(t'qx)(r_1 y))$ from $B^\#$ realizes $g \leq \mathrm{rep}^f$.

Let $\alpha \in \mathcal{B}A$ and $a \in A$ such that $\alpha(a)\downarrow$, and consider $b \in g(\alpha)$ and $c \in f(a)$. Then $r_1 c \in gi(a) = g(\hat{a})$, which yields that $t'(t'qb)(r_1 c) \in g(\rho\alpha\hat{a}) \subseteq g\big(\widehat{\alpha(a)}\big) = gi(\alpha(a))$. This gives $sbc \leq r_0(t'(t'qb)(r_1 c)) \in f(\alpha(a))$, so we can conclude that $s \cdot g(\alpha) \cdot f(a) \subseteq f(\alpha(a))$. Since this holds for all $a \in \operatorname{dom}\alpha$, we can conclude $s \cdot g(\alpha) \subseteq \mathrm{rep}^f(\alpha)$, as desired. □

Note that, by Lemma 2.2.10(ii), rep^f is decidable, since f is decidable.

Remark 5.2.11. The largest part of the proof of Theorem 5.2.10(i) consists of showing that rep^f preserves application up to a realizer. For $f = \mathrm{id}_A$, this is Lemma 5.2.2. For $f = \iota_{\vec{\alpha}}$, this follows from Lemma 5.2.4. But of course, Lemma 5.2.4 is stronger, because it also asserts that this realizer does not depend on the $\vec{\alpha}$.

Corollary 5.2.12. *Let $f\colon A \multimap B$ be a decidable partial applicative morphism. Then:*

$$\{\alpha \in \mathcal{B}A \mid \alpha \text{ is representable w.r.t. } f\}$$

is a filter on $\mathcal{B}A$.

Proof. The set in the corollary is $\mathrm{dom}^{\#}(\mathrm{rep}^f)$. □

Since $i\colon A \to \mathcal{B}A$ is c.d., we know in particular that it is decidable. This means we can extend the construction $A \mapsto \mathcal{B}A$ to decidable partial applicative morphisms $f\colon A \multimap B$, by defining $\mathcal{B}f\colon \mathcal{B}A \multimap \mathcal{B}B$ as $\mathrm{rep}^{i_B f}$. By Theorem 5.2.10, this yields a 'lax' 1-functor in the sense that $\mathrm{id}_{\mathcal{B}A} \leq \mathcal{B}\,\mathrm{id}_A$ and $\mathcal{B}g \circ \mathcal{B}f \leq \mathcal{B}(gf)$. In fact, Proposition 5.2.13(i) below says that the first of these inequalities is always an isomorphism. Note that there is no obvious way to extend $\mathcal{B}$ to inequalities between morphisms. Indeed, if $f, g\colon A \multimap B$ are decidable and satisfy $f \leq g$, then this does not give any means of comparing $\mathrm{rep}^f(\alpha)$ and $\mathrm{rep}^g(\alpha)$, for $\alpha \in \mathcal{B}A$.

Proposition 5.2.13. *Consider $i\colon A \to \mathcal{B}A$.*

(i) We have $\mathrm{rep}^i \simeq \mathrm{id}_{\mathcal{B}A}$.

(ii) The morphism i has a right adjoint $j\colon \mathcal{B}A \multimap A$ in OPCA_D which satisfies $ji \simeq \mathrm{id}_A$.

In particular, there is a local geometric morphism $\mathsf{RT}(\mathcal{B}A) \twoheadrightarrow \mathsf{RT}(A)$.

Proof. (i) Since $\mathrm{id}_{\mathcal{B}A} \circ i = i$, Theorem 5.2.10 tells us that $\mathrm{id}_{\mathcal{B}A} \leq \mathrm{rep}^i$. For the converse inequality, construct an $r \in A^{\#}$ such that:

- $r \cdot [x] \leq \mathsf{p}\bot[\mathsf{i}]$;
- $r \cdot [x, u_0, \ldots, u_i] \leq$ if $\mathsf{p}_0 u_i$ then $\mathsf{p}\top(\mathsf{p}_1 u_i)$ else $\mathsf{p}\bot[\mathsf{i}, \underbrace{x, \ldots, x}_{i+1 \text{ times}}]$.

If $\rho \in (\mathcal{B}A)^{\#}$ is defined by $\rho(a) \simeq ra$, then $\rho\beta(a) \preceq \beta\hat{a}(\mathsf{i})$ for all $a \in A$ and $\beta \in \mathcal{B}A$. Now suppose that $\beta \in \mathrm{rep}^i(\alpha)$, that is, $\beta\hat{a} \preceq \widehat{\alpha(a)}$ for all $a \in A$. Then $\rho\beta(a) \preceq \beta\hat{a}(\mathsf{i}) \preceq \widehat{\alpha(a)}(\mathsf{i}) \simeq \alpha(a)$, so $\rho\beta \leq \alpha$. We conclude that ρ realizes $\mathrm{rep}^i \leq \mathrm{id}_{\mathcal{B}A}$, as desired.

(ii) Let $j = \mathrm{rep}^{\mathrm{id}_A}\colon \mathcal{B}A \multimap A$. Then by Theorem 5.2.10(i), we have $ji \simeq \mathrm{id}_A$. This yields $iji \simeq i$, so by Theorem 5.2.10(ii), we get $ij \leq \mathrm{rep}^i \simeq \mathrm{id}_{\mathcal{B}A}$, which completes the proof of (ii).

The final statement follows from Corollary 3.4.8. □

Remark 5.2.14. Now we can see that the inequality $g \leq \mathrm{rep}^f$ in Theorem 5.2.10 cannot, in general, be an isomorphism. Indeed, if that were the case, then in the proof of Proposition 5.2.13, we could conclude that $ij \simeq \mathrm{id}_{\mathcal{B}A}$, and thus that i and j constitute an equivalence of PCAs. By the remark following the proof of Theorem 2.3.14, this means that j is an arrow of OPCA. In particular, j is total, so every function in $\mathcal{B}A$ has a representer. Now take A to be a (nontrivial) discrete PCA. Then the partial function $A \rightharpoonup A^A$ sending an element of A to the total function it represents (if any), is surjective. But by Cantor's theorem, such a surjection cannot exist.

Since $\mathrm{id}_{\mathcal{B}A} \leq \mathrm{rep}^i$, every $\alpha \in \mathcal{B}A$ has a representer w.r.t. i. On the other hand, since we also have $\mathrm{rep}^i \leq \mathrm{id}_{\mathcal{B}A}$, any function that is *representable* w.r.t. i (i.e., has a representer w.r.t. i in $(\mathcal{B}A)^\#$) must already be in $(\mathcal{B}A)^\#$. So we see that i creates no new representable functions. This also follows from the fact that i has a retraction, along with Lemma 5.1.4(i).

In Example 5.1.20, we saw that a function $\beta \in \mathcal{B}A$ is representable w.r.t. $\iota_\alpha \colon A \to A[\alpha]$ iff the identity on A is a morphism of PCAs $A[\beta] \to A[\alpha]$. Using the machinery developed in this section, we can give a succinct description of the set of all such β. Note that, since α is an element of $\mathcal{B}A$, we can construct the PCA $\mathcal{B}A[\alpha]$ as in Definition 2.1.33. The identity on $\mathcal{B}A$ is a morphism of PCAs $\mathcal{B}A \to \mathcal{B}A[\alpha]$, which we will denote by ι'_α in order to distinguish it from $\iota_\alpha \colon A \to A[\alpha]$.

Proposition 5.2.15. *Let $\alpha \in \mathcal{B}A$.*

(i) We have $\mathrm{rep}^{\iota'_\alpha i} \simeq \iota'_\alpha$.

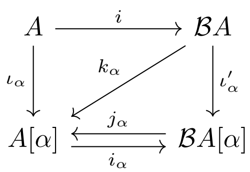
(ii) There is a morphism of PCAs $i_\alpha \colon A[\alpha] \to \mathcal{B}A[\alpha]$, which has a right adjoint $j_\alpha \colon \mathcal{B}A[\alpha] \rightharpoonup A[\alpha]$ in OPCA_D, satisfying $j_\alpha i_\alpha \simeq \mathrm{id}_{A[\alpha]}$.

(iii) We have:

$$\{\beta \in \mathcal{B}A \mid \beta \text{ is representable w.r.t. } \iota_\alpha\} = (\mathcal{B}A[\alpha])^\# = \langle (\mathcal{B}A)^\# \cup \{\alpha\} \rangle.$$

Proof. (i) Note that $\mathrm{rep}^{\iota'_\alpha i}(\beta) = \mathrm{rep}^i(\beta)$, since the application map on $\mathcal{B}A[\alpha]$ is the same as on $\mathcal{B}A$. So this follows immediately from Proposition 5.2.13(i).

(ii) By (i), we get that $\beta \in \mathcal{B}A$ is representable w.r.t. $\iota'_\alpha i$ iff $\beta \in (\mathcal{B}A[\alpha])^\#$. In particular, α itself is representable w.r.t. $\iota'_\alpha i$, so by the universal property of $A[\alpha]$, we obtain $i_\alpha \colon A[\alpha] \to \mathcal{B}A[\alpha]$ such that $i_\alpha \iota_\alpha \simeq \iota'_\alpha i$.



Now define $k_\alpha \colon \mathcal{B}A \rightharpoonup A[\alpha]$ as $\mathrm{rep}^{\iota_\alpha}$, so that $k_\alpha i \simeq \iota_\alpha$. Since α is representable w.r.t. ι_α, we have that $\alpha \in \mathrm{dom}^\#(k_\alpha)$, which means that k_α factors through ι'_α. So we obtain $j_\alpha \colon \mathcal{B}A[\alpha] \rightharpoonup A[\alpha]$ such that $j_\alpha \iota'_\alpha \simeq k_\alpha$. Now we get:

$$j_\alpha i_\alpha \iota_\alpha \simeq j_\alpha \iota'_\alpha i \simeq k_\alpha i \simeq \iota_\alpha,$$

which yields $j_\alpha i_\alpha \simeq \mathrm{id}_{A[\alpha]}$. Moreover, we have:

$$i_\alpha j_\alpha \iota'_\alpha i \simeq i_\alpha k_\alpha i \simeq i_\alpha \iota_\alpha \simeq \iota'_\alpha i.$$

Now Theorem 5.2.10(ii) yields $i_\alpha j_\alpha \iota'_\alpha \leq \mathrm{rep}^{\iota'_\alpha i} \simeq \iota'_\alpha$, and from this, we can conclude that $i_\alpha j_\alpha \leq \mathrm{id}_{\mathcal{B}A[\alpha]}$. This concludes the proof of (ii).

(iii) Since i_α has a retraction j_α, Lemma 5.1.4(i) implies that $\beta \in \mathcal{B}A$ is representable w.r.t. ι_α iff β is representable w.r.t. $i_\alpha \iota_\alpha \simeq \iota'_\alpha i$. By (i), the latter holds iff $\beta \in (\mathcal{B}A[\alpha])^\#$, as desired. ∎

In particular, there is a local geometric morphism $\mathsf{RT}(\mathcal{B}A[\alpha]) \twoheadrightarrow \mathsf{RT}(A[\alpha])$. Recall from Example 4.4.38 that $\mathsf{RT}(\mathcal{B}A[\alpha]) \simeq \mathsf{RT}(\mathcal{B}A)/1_\alpha$, where $1_\alpha \in \mathsf{RT}(\mathcal{B}A)$ is the subterminal assembly given by $|1_\alpha| = \{*\}$ and $E_{1_\alpha}(*) = {\downarrow}\{\alpha\}$. Now the diagram

$$\begin{array}{ccc} A & \xrightarrow{i} & \mathcal{B}A \\ {\scriptstyle \iota_\alpha}\downarrow & & \downarrow{\scriptstyle \iota'_\alpha} \\ A[\alpha] & \xrightarrow{i_\alpha} & \mathcal{B}A[\alpha] \end{array}$$

of c.d. morphisms of PCAs yields the following diagram of toposes:

$$\begin{array}{ccc} \mathsf{RT}(\mathcal{B}A)/1_\alpha & \hookrightarrow & \mathsf{RT}(\mathcal{B}A) \\ \downarrow & & \downarrow \\ \mathsf{RT}(A[\alpha]) & \hookrightarrow & \mathsf{RT}(A) \end{array}$$

Thus, if we cover $\mathsf{RT}(A)$ by $\mathsf{RT}(\mathcal{B}A)$, and then take the open subtopos corresponding to the subterminal 1_α, then we can retrieve $\mathsf{RT}(A[\alpha])$ as the image of the composite geometric morphism $\mathsf{RT}(\mathcal{B}A)/1_\alpha \hookrightarrow \mathsf{RT}(\mathcal{B}A) \twoheadrightarrow \mathsf{RT}(A)$.

5.3 The second-order case

The paper [FvO16] constructs, for each absolute discrete PCA A and partial function $F\colon A^A \rightharpoonup A$, a new PCA $A[F]$ which is the 'universal solution' to making F representable (but see also Remark 5.3.15 below). In this section, we generalize this construction to relative and *some* ordered PCAs. It turns out that the construction of $A[F]$ does not work for all ordered PCAs. Therefore, in Section 5.3.1 below, we will introduce a condition on A, called *chain-completeness*, with the following two properties.

- Every discrete PCA is chain-complete, and for every chain-complete PCA A and $F \in \mathcal{B}_2 A$, we can construct $A[F]$. Thus, our construction is at least as general as the construction in [FvO16].
- If A is chain-complete, then so is $\mathcal{B}A$. This will allow us to apply the construction of $A[F]$ to $\mathcal{B}A$ in Section 5.4 below (provided A is chain-complete).

5.3.1 Chain-completeness and fixpoints

In [FvO16], the PCA $A[F]$ is obtained as a special case of the construction $A[\alpha]$, where α can be thought of as a 'master function' that encodes all relevant information about F. Here, we employ the same strategy, but we show that the required α is an instance of a more general construction, namely of *fixpoints* in $\mathcal{B}A$. First, let us give the definition of chain-completeness.

Definition 5.3.1. *Let A be a PCA.*

*(i) A **chain** in A is a non-empty subset $X \subseteq A$ that is totally ordered by $\leq$.*

*(ii) A is called **chain-complete** if every chain $X \subseteq A$ has a greatest lower bound in A, which we will denote by $\bigwedge X$.*

Now suppose that A is chain-complete.

*(iii) A morphism of PCAs $f: A \to B$ is called **chain-continuous** if, for every chain $X \subseteq A$, the set $f(X) = \{f(a) \in X \mid a \in X\}$ has a greatest lower bound $\bigwedge f(X)$ in B, and $\bigwedge f(X) = f(\bigwedge X)$.*

(iv) An applicative morphism $f: A \multimap B$ (resp. partial applicative morphism $f: A \leftharpoondown\!\!\circ\, B$) is called chain-complete if $f: A \to TB$ (resp. $f: A \to DB$) is chain-complete.

Remark 5.3.2. (i) In Definition 5.3.1(i) above, we exclude the empty chain because we do not, in general, want to require that A has a top element.

(ii) If $f: A \to B$ is chain-complete, then f must preserve the order on the nose. Indeed, if $a' \leq a$, then $\{a, a'\}$ is a chain with greatest lower bound a', and we get $f(a') = f(a) \wedge f(a')$, i.e., $f(a') \leq f(a)$. In particular, if $X \subseteq A$ is a chain, then $f(X) \subseteq B$ is a chain as well.

(iii) As usual, the notion of chain-completeness is invariant across the inclusions $\mathsf{OPCA} \hookrightarrow \mathsf{OPCA}_T \hookrightarrow \mathsf{OPCA}_D$. We leave the proof to the reader.

(iv) In Definition 5.3.1(iv) above, the condition that f is chain-complete can (in both cases) be written as: if X is a chain in A then $f(\bigwedge X) = \bigcap_{a \in X} f(a)$.

Example 5.3.3. Every discrete PCA A is chain-complete, since the only chains are singletons. Thus, every morphism of PCAs $A \to B$ is trivially chain-continuous if A is discrete.

Example 5.3.4. Note that DA is always chain-complete since DA has all greatest lower bounds, given by intersections. On the other hand, TA is never chain-complete if A is not semitrivial. Indeed, the chain $(\downarrow\{\overline{m} \mid m \geq n\})_{n \in \mathbb{N}}$ has empty intersection.

Example 5.3.5. Suppose that A and B are chain-complete PCAs. If we have morphisms of PCAs $A \to B \to C$ that are both chain-continuous, then their composition $A \to C$ is chain-continuous as well. Similarly, if we have chain-continuous $A \to B \leftharpoondown\!\!\circ\, C$, that is, a morphism of PCAs followed by a partial applicative morphism, then their composition $A \leftharpoondown\!\!\circ\, C$ in OPCA_D is chain-continuous as well. On the other hand, the composition of two chain-continuous partial applicative morphisms $A \leftharpoondown\!\!\circ\, B \leftharpoondown\!\!\circ\, C$ is *not* necessarily chain-continuous. For example, let A be a PCA that is not semitrivial, and consider the partial applicative morphism $f: DA \leftharpoondown\!\!\circ\, A$ given by $f(\alpha) = \alpha$, and the unique morphism of PCAs $g: A \to \mathbf{1}$. Both are clearly chain-continuous, but their composition

$gf\colon DA \rightharpoonup \mathbf{1}$ is not. Indeed, we have $gf(\alpha) = \mathbf{1}$ if $\alpha \neq \emptyset$, and $gf(\emptyset) = \emptyset$. Now consider again the chain $(\downarrow\{\overline{m} \mid m \geq n\})_{n\in\mathbb{N}}$ that has empty intersection. Then $\bigcap_{n\geq 0} gf(\downarrow\{\overline{m} \mid m \geq n\}) = \bigcap_{n\geq 0} \mathbf{1} = \mathbf{1} \neq \emptyset = gf(\emptyset)$, so gf does not preserve the greatest lower bound of this chain.

The following two lemmata explain the relation between chain-completeness and the constructions from Section 5.2.

Lemma 5.3.6. *Let A be a chain-complete PCA. Then $\mathcal{B}A$ is chain-complete as well, and $i\colon A \to \mathcal{B}A$ is chain-continuous.*

Proof. Let X be a chain in $\mathcal{B}A$. We define its greatest lower bound $\beta = \bigwedge X$ as follows. First of all, we set:

$$\operatorname{dom}\beta = \bigcup_{\alpha\in X} \operatorname{dom}\alpha = \{a \in A \mid \exists \alpha \in X\,(\alpha(a)\downarrow)\}.$$

For $a \in \operatorname{dom}\beta$, the set $X_a = \{\alpha(a) \mid \alpha \in X \text{ and } a \in \operatorname{dom}\alpha\}$ is a chain in A. This means we can define $\beta(a) = \bigwedge X_a$. We leave it to the reader to show that $\beta \in \mathcal{B}A$, and that β is indeed the greatest lower bound of X.

The final statement says that, for a chain $X \subseteq A$, we have $\bigwedge_{a\in X} \hat{a} = \widehat{\bigwedge X}$. This is easy to show. □

Lemma 5.3.7. *Consider a decidable partial applicative morphism $f\colon A \rightharpoonup B$, where A is chain-complete, and consider $\mathrm{rep}^f\colon \mathcal{B}A \rightharpoonup B$ as in Theorem 5.2.10. Then f is chain-continuous if and only if rep^f is chain-continuous.*

Proof. By Example 5.3.5 and Lemma 5.3.6, we know that f is chain-continuous if rep^f is chain-continuous. Conversely, suppose that f is chain-continuous. We recall that, since f preserves the order on the nose, rep^f does so as well. If X is a chain in $\mathcal{B}A$ with greatest lower bound $\beta = \bigwedge X$, then this already implies that $\mathrm{rep}^f(\beta) \subseteq \bigcap_{\alpha\in X} \mathrm{rep}^f(\alpha)$. So it remains to show the converse inclusion, i.e., if $b \in B$ represents all $\alpha \in X$ w.r.t. f, then b also represents β w.r.t. f.

So suppose that $b \in B$ represents all $\alpha \in X$ w.r.t. f, and consider an $a \in \operatorname{dom}\beta$. Then $a \in \operatorname{dom}\alpha$ for at least one $\alpha \in X$, which implies that $b \cdot f(a)$ is defined. Moreover, for all $\alpha \in X$ such that $a \in \operatorname{dom}\alpha$, we have $b \cdot f(a) \subseteq f(\alpha(a))$. This yields:

$$b \cdot f(a) \subseteq \bigcap_{\substack{\alpha\in X\\ a\in\operatorname{dom}\alpha}} f(\alpha(a)) = f\left(\bigwedge\{\alpha(a) \mid \alpha \in X \text{ and } a \in \operatorname{dom}\alpha\}\right) = f(\beta(a)),$$

as desired. □

The most important feature of chain-complete PCAs that we can construct *fixpoints* of total, order-preserving functions on $\mathcal{B}A$. The construction of such a fixpoint proceeds by transfinite recursion; in order to handle the limit case, we need chain-completeness.

Proposition 5.3.8. *Let A be a chain-complete PCA and let $F \in \mathcal{BB}A$ be a* total *function. Then F has a largest fixpoint in $\mathcal{B}A$.*

Remark 5.3.9. It may seem strange that we construct a *largest* fixpoint, since recursion theory is usually concerned with *smallest* fixpoints. However, we must keep in mind that 'largest' should be read w.r.t. the ordering on $\mathcal{B}A$ as in Definition 2.1.4. If A is discrete, then this is the *reverse* subfunction relation, so what we identify as the largest fixpoint would usually indeed be called the smallest fixpoint.

Proof of Proposition 5.3.8. Note that a total function $F \in \mathcal{BB}A$ is simply an order-preserving function $\mathcal{B}A \to \mathcal{B}A$. We define, recursively, an ordinal-indexed sequence $(\alpha_\gamma)_{\gamma\in\mathrm{ORD}}$ of elements of $\mathcal{B}A$, as follows:

- α_0 is the empty function $\emptyset$;
- $\alpha_{\gamma+1} = F(\alpha_\gamma)$;
- $\alpha_\lambda = \bigwedge_{\kappa<\lambda} \alpha_\kappa$ if $\lambda > 0$ is a limit ordinal.

Using transfinite induction and the fact that F is order-preserving, one may show that $\alpha_\gamma \geq \alpha_\delta$ for $\gamma \leq \delta$, and that the sequence is well-defined. By cardinality considerations, the sequence must stabilize at some point, i.e., there exists an ordinal ζ such that $F(\alpha_\zeta) = \alpha_\zeta$. Then α_ζ is a fixpoint of f. Moreover, if $\beta \in \mathcal{B}A$ is an element satisfying $\beta \leq F(\beta)$, then by transfinite induction, it easily follows that $\beta \leq \alpha_\gamma$ for every ordinal γ, and in particular, $\beta \leq \alpha_\zeta$. We conclude that α_ζ is the largest fixpoint of F. □

As in classical recursion theory, we expect that, if the operation F is 'computable', then so is its largest fixpoint. We will obtain this as a corollary of the following more general proposition.

Proposition 5.3.10. *Let A be a chain-complete PCA, and let $f\colon A \multimap B$ be decidable and chain-continuous. If $F \in \mathcal{BB}A$ is a total function which is representable w.r.t. $\mathrm{rep}^f\colon \mathcal{B}A \multimap B$, then the largest fixpoint of F is representable w.r.t. f.*

Proof. Let $\mathsf{z} \in B^\#$ be the guarded fixpoint combinator as in Construction 2.1.30. We will show that, if $r \in B^\#$ represents F w.r.t. rep^f, then $\mathsf{z}r$ (which is always defined) represents the largest fixpoint of F w.r.t. f. Clearly, this implies the proposition, since $\mathsf{z}r \in B^\#$.

So suppose that $r \in B^\#$ represents F w.r.t. rep^f. Define the sequence $(\alpha_\gamma)_{\gamma\in\mathrm{ORD}}$ as in the proof of Proposition 5.3.8, so that the largest fixpoint of F is α_ζ for some ordinal ζ. We will show, using transfinite induction, that $\mathsf{z}r \in \mathrm{rep}^f(\alpha_\gamma)$ for all ordinals γ. In particular, we will have $\mathsf{z}r \in \mathrm{rep}^f(\alpha_\zeta)$, which means that $\mathsf{z}r$ represents α_ζ w.r.t. f, as desired.

First of all, we have $\mathrm{rep}^f(\alpha_0) = \mathrm{rep}^f(\emptyset) = B$, so the base case is trivial. Now suppose that $\mathsf{z}r \in \mathrm{rep}^f(\alpha_\gamma)$ for a certain ordinal γ. Then $r(\mathsf{z}r) \in \mathrm{rep}^f(F(\alpha_\gamma)) = \mathrm{rep}^f(\alpha_{\gamma+1})$, i.e., $r(\mathsf{z}r)$ represents $\alpha_{\gamma+1}$ w.r.t. f. Now, since $\mathsf{z}rb \preceq r(\mathsf{z}r)b$ for all

$b \in B$, it follows that $\mathsf{z}r$ also represents $\alpha_{\gamma+1}$ w.r.t. f, i.e., $\mathsf{z}r \in \mathrm{rep}^f(\alpha_{\gamma+1})$. Finally, Lemma 5.3.7 tells us that rep^f is chain-continuous, from which the limit case immediately follows. This completes the induction. □

Corollary 5.3.11. *If A is chain-complete and $F \in \mathcal{BB}A$ is total and representable, then the largest fixpoint of F is representable as well.*

Proof. Let α be the largest fixpoint of F. Recall from Proposition 5.2.13(i) that $\mathrm{rep}^i \simeq \mathrm{id}_{\mathcal{B}A}$. Thus, if we apply Proposition 5.3.10 to $i\colon A \to \mathcal{B}A$, we see: if F is representable, then F is representable w.r.t. rep^i, which means that α is reprentable w.r.t. i. But we know that an element of $\mathcal{B}A$ is representable w.r.t. i iff it is representable, so α is representable, as desired. □

Remark 5.3.12. There is an alternative proof of Corollary 5.3.11 that does not require Proposition 5.3.10. Suppose that $F \in \mathcal{BB}A$ is represented by $\rho \in (\mathcal{B}A)^{\#}$, and let $\alpha \in \mathcal{B}A$ be the largest fixpoint of F. If $\mathsf{y} \in (\mathcal{B}A)^{\#}$ is the unguarded fixpoint combinator (Construction 2.1.30), then $\mathsf{y}\rho$ is defined since $\mathcal{B}A$ is a total PCA. Moreover, we have $\mathsf{y}\rho \leq \rho(\mathsf{y}\rho) \leq F(\mathsf{y}\rho)$, which suffices to ensure that $\mathsf{y}\rho \leq \alpha$. Since $\mathsf{y}\rho \in (\mathcal{B}A)^{\#}$, we see that $\alpha \in (\mathcal{B}A)^{\#}$ as well, i.e., α is representable.

5.3.2 Adjoining a type-2 functional

In this section, we construct, for each chain-complete PCA A and $F \in \mathcal{B}_2A$, a morphism of PCAs $\iota_F\colon A \to A[F]$ such that F is representable w.r.t. ι_F. Moreover, if F is representable w.r.t. a decidable and chain-continuous partial applicative morphism $f\colon A \multimap B$, then f will factor through ι_F. Note that, if A is discrete, then A^A is the set of minimal elements of $\mathcal{B}A$, which means that every partial function $A^A \rightharpoonup A$ will be in $\mathcal{B}_2A$. Thus, [FvO16, Theorem 3.1] is a special case of our result.

The construction we give of $A[F]$ is essentially the same as in [FvO16], even though this is not obvious at first glance. First, let us explain the main idea behind the construction. If $F \in B_2A$ is to be representable, then there should exist an element $r \in A^{\#}$ such that $r \cdot \mathrm{rep}(\beta) \subseteq {\downarrow}\{F(\beta)\}$ for all $\beta \in \mathrm{dom}\, F$. We can force such an element to exist by adjoining a *function* $\alpha \in \mathcal{B}A$ (as in Section 5.1.2) that maps elements of $\mathrm{rep}(\beta)$ to elements below $F(\beta)$. Then we would like to say that $\iota_\alpha\colon A \to A[\alpha]$ works as desired, but this does not follow. The reason is that now we need an $r \in A[\alpha]^{\#}$ such that $r \odot_\alpha \mathrm{rep}^{\iota_\alpha}(\beta) \subseteq {\downarrow}\{F(\beta)\}$, and $\mathrm{rep}^{\iota_\alpha}(\beta)$ is not the same as $\mathrm{rep}(\beta)$. Indeed, the whole point of the construction $A[\alpha]$ is to make *more* functions representable! The main idea is to extend α to a new function α' that maps elements of $\mathrm{rep}^{\iota_\alpha}(\beta)$ to elements below $F(\beta)$. Then we are again faced with the same problem, so we do this transfinitely many times, until the function α stabilizes. This will be the 'master function' we alluded to at the beginning of Section 5.3.1. Of course, we have already seen a construction using transfinite recursion in the previous section, in order to construct greatest fixpoints in $\mathcal{B}A$ (Proposition 5.3.8). We will obtain the 'master function' α as a special instance of this fixpoint construction.

Theorem 5.3.13. *Let A be a chain-complete PCA and let $F \in \mathcal{B}_2 A$. Then there exists an $\alpha \in \mathcal{B}A$ such that:*

(i) F is representable w.r.t. $\iota_\alpha \colon A \to A[\alpha]$;

(ii) if $f \colon A \multimap B$ is decidable and chain-continuous, and F is representable w.r.t. f, then f factors, essentially uniquely, through ι_α.

Proof. Define the total function $\tilde{F} \in \mathcal{B}\mathcal{B}A$ by:

$$\tilde{F}(\beta)(a) :\simeq F(x \mapsto a \odot_\beta x) \simeq F((x \mapsto ax) \cdot \beta) \quad \text{for } \beta \in \mathcal{B}A, a \in A,$$

where the second Kleene equality is a consequence of the remark below Table 5.1. First, let us show that $\tilde{F}$ is well-defined. If $b \leq a$, then we have $(x \mapsto bx) \leq (x \mapsto ax)$ by axiom (A) from Definition 2.1.19, so $(x \mapsto bx) \cdot \beta \leq (x \mapsto ax) \cdot \beta$ by axiom (A) for $\mathcal{B}A$. This yields

$$\tilde{F}(\beta)(b) \simeq F((x \mapsto bx) \cdot \beta) \preceq F((x \mapsto ax) \cdot \beta) \simeq \tilde{F}(\beta)(a),$$

since $F \in \mathcal{B}_2 A$, so we have $\tilde{F}(\beta) \in \mathcal{B}A$. Moreover, if $\gamma \leq \beta$, then $(x \mapsto ax) \cdot \gamma \leq (x \mapsto ax) \cdot \beta$ by axiom (A) for $\mathcal{B}A$, which gives:

$$\tilde{F}(\gamma)(a) \simeq F((x \mapsto ax) \cdot \gamma) \preceq F((x \mapsto ax) \cdot \beta) \simeq \tilde{F}(\beta)(a),$$

again since $F \in \mathcal{B}_2 A$. This means that $\tilde{F}(\gamma) \leq \tilde{F}(\beta)$, so we indeed have $\tilde{F} \in \mathcal{B}\mathcal{B}A$. We take α to be the greatest fixpoint of $\tilde{F}$ as provided by Proposition 5.3.8.

(i) Suppose that $a \in \mathrm{rep}^{\iota_\alpha}(\beta)$ for some $\beta \in \mathcal{B}A$. Then $a \odot_\alpha x \preceq \beta(x)$ for all $x \in A$, which means that $(x \mapsto a \odot_\alpha x) \leq \beta$. This gives:

$$\alpha(a) \simeq \tilde{F}(\alpha)(a) \simeq F(x \mapsto a \odot_\alpha x) \preceq F(\beta),$$

since $F \in \mathcal{B}_2 A$. Thus, for all $\beta \in \mathrm{dom}\, F$ and $a \in \mathrm{rep}^{\iota_\alpha}(\beta)$, we have $\alpha(a) \leq F(\beta)$, which implies that any representer of α w.r.t. ι_α also represents F w.r.t. ι_α. Since α is, by construction, representable w.r.t. ι_α, this completes the proof of (i).

(ii) By Theorem 5.1.17, it suffices to show that α is representable w.r.t. f. And Proposition 5.3.10 tells us that, in order to show this, it suffices to show that $\tilde{F}$ is representable w.r.t. $\mathrm{rep}^f \colon \mathcal{B}A \multimap B$.

Let $s \in B^\#$ represent F w.r.t. f, i.e., $s \cdot \mathrm{rep}^f(\beta) \subseteq f(F(\beta))$ for all $\beta \in \mathrm{dom}\, F$. Moreover, let $\mathrm{rep}^f \colon \mathcal{B}A \multimap B$ preserve application up to $t \in B^\#$. Finally, let $r \in B^\#$ be an element such that $r \cdot f(a) \in \mathrm{rep}^f(x \mapsto ax)$ for all $a \in A$; e.g., we can take $r = \lambda^* xy.t'xy$, where f preserves application up to $t' \in B^\#$. Now we claim that the element $q := \lambda^* xy.s(t(ry)x)$ of $B^\#$ represents $\tilde{F}$ w.r.t. rep^f.

Let $\beta \in \mathcal{B}A$ and $a \in A$ such that $\tilde{F}(\beta)(a)\downarrow$, and consider $b \in \mathrm{rep}^f(\beta)$ and $c \in f(a)$. By definition, we have $rc \in \mathrm{rep}^f(x \mapsto ax)$, which yields $t(rc)b \in \mathrm{rep}^f((x \mapsto ax) \cdot \beta)$. Since $(x \mapsto ax) \cdot \beta \in \mathrm{dom}\, F$, this means that $s(t(rc)b)$ is defined and an element of $f(F((x \mapsto ax) \cdot \beta)) = f(\tilde{F}(\beta)(a))$. This gives that qbc is also defined and an element of $f(\tilde{F}(\beta)(a))$. Since this holds for all $c \in f(a)$, we can conclude that $qb \in \mathrm{rep}^f(\tilde{F}(\beta))$ for all $b \in \mathrm{rep}^f(\beta)$, as desired. ☐

Definition 5.3.14. *Let A be a chain-complete PCA, let $F \in \mathcal{B}_2 A$ and suppose that $\alpha \in \mathcal{B}A$ is as in Theorem 5.3.13. Then we denote $\iota_\alpha \colon A \to A[\alpha]$ by $\iota_F \colon A \to A[F]$.*

Remark 5.3.15. As observed in Example 5.1.10, the representability of type-2 functionals is not 'transferable' along partial applicative morphisms. Thus, if we have a partial applicative morphism $A[F] \multimap B$, it does not follow that F is representable w.r.t. the composition $A \xrightarrow{\iota_F} A[F] \multimap B$. This means that we do not have an analogue of Corollary 5.1.18 in the type-2 case.

In Proposition 5.2.13(i), we saw that $\mathrm{rep}^i \simeq \mathrm{id}_{\mathcal{B}A}$, where $i \colon A \to \mathcal{B}A$. Informally, we might say that *elements* of $\mathcal{B}A$ and their *representers* (w.r.t. i) can be used interchangably. An analogous result holds if we take a finite extension of the filter $(\mathcal{B}A)^\#$ (Proposition 5.2.15(i)). The following example shows that the same is *not* true if we equip $\mathcal{B}A$ with an oracle.

Example 5.3.16. Let A be Kleene's first model $\mathcal{K}_1$. Consider the *Kleene functional* $E \colon \mathbb{N}^\mathbb{N} \to \mathbb{N}$ defined by:

$$E(\alpha) = \begin{cases} 0 & \text{if } \forall n \in \mathbb{N}(\alpha(n) = 0); \\ 1 & \text{if } \exists n \in \mathbb{N}(\alpha(n) > 0), \end{cases}$$

which is in $\mathcal{B}_2\mathcal{K}_1$ since its domain consists of total functions. Now define $\hat{E} \in \mathcal{B}\mathcal{B}\mathcal{K}_1$ by $\hat{E}(\alpha) = \widehat{E(\alpha)}$ for $\alpha \colon \mathbb{N} \to \mathbb{N}$. We will now proceed to show the following things about the composition:

$$\mathcal{K}_1 \xrightarrow{i} \mathcal{B}\mathcal{K}_1 \xrightarrow{\iota_{\hat{E}}} \mathcal{B}\mathcal{K}_1[\hat{E}].$$

1. Every total function $\mathbb{N} \to \mathbb{N}$ which is representable w.r.t. $\iota_{\hat{E}} i$, is an arithmetical function, that is, its graph is an arithmetical relation.
2. The composition $\iota_{\hat{E}} i$ does *not* factor through $\iota_E \colon \mathcal{K}_1 \to \mathcal{K}_1[E]$.
3. The type-2 functional E is *not* representable w.r.t. $\iota_{\hat{E}} i$.
4. We have $\iota_{\hat{E}} \not\simeq \mathrm{rep}^{\iota_{\hat{E}} i}$.

For the first claim, we first observe the following. Since the range of $\hat{E}$ is simply $\{\hat{0}, \hat{1}\} \subseteq (\mathcal{B}\mathcal{K}_1)^\#$, we have $\left(\mathcal{B}\mathcal{K}_1[\hat{E}]\right)^\# = (\mathcal{B}\mathcal{K}_1)^\#$ (by Remark 5.1.16(ii)), which is the upwards closure of the set of partial recursive functions. Now, if $\alpha, \beta \in \mathcal{B}\mathcal{K}_1$ are partial functions, then the graph of $\alpha\beta \in \mathcal{B}A$ can be defined arithmetically in terms of the graphs of α and β, as is immediate from the definition of application in $\mathcal{B}\mathcal{K}_1$, and the fact that the relation $\varphi_e(m) = n$ is arithmetical. Moreover, if $\alpha \in \mathcal{B}\mathcal{K}_1$ and $i \in \mathbb{N}$, then $\hat{E}(\alpha) = \hat{i}$ can be expressed arithmetically in terms of i and the graph of α. And we know that all the combinators in $(\mathcal{B}\mathcal{K}_1)^\#$ can be taken to be partial recursive functions, and therefore have arithmetically expressible graphs. Finally, we assume for simplicity that the booleans in $(\mathcal{BK}_1)^\#$ are simply $\hat{0}$ and $\hat{1}$, which we can do, because $i\colon \mathcal{K}_1 \to \mathcal{BK}_1$ is decidable. Now suppose that $\alpha\colon \mathbb{N} \to \mathbb{N}$ is represented w.r.t. $\iota_{\hat{E}} i$ by $\rho \in \left(\mathcal{BK}_1[\hat{E}]\right)^\# = (\mathcal{BK}_1)^\#$. Then we can assume without loss of generality that ρ is partial recursive. Since α is total, we have $\alpha(a) = b$ iff $\rho \odot_{\hat{E}} \hat{a} = \hat{b}$. We see that $\rho \odot_{\hat{E}} \hat{a} = \hat{b}$ holds iff there exists a coded sequence $u = [u_0, \ldots, u_{n-1}]$ of *natural numbers*, such that:

- for all $i < n$, we have
$$\mathsf{p}_0(\rho \cdot [\hat{a}, \hat{u}_0, \ldots, \hat{u}_{i-1}]) = \hat{0} \quad \text{and} \quad \hat{E}\left(\mathsf{p}_1(\rho \cdot [\hat{a}, \hat{u}_0, \ldots, \hat{u}_{i-1}])\right) = \hat{u}_i;$$
- $\mathsf{p}_0(\rho \cdot [\hat{a}, \hat{u}_0, \ldots, \hat{u}_{n-1}]) = \hat{1}$ and $\mathsf{p}_1(\rho \cdot [\hat{a}, \hat{u}_0, \ldots, \hat{u}_{n-1}]) = \hat{b}$.

By the remarks above, this is an arithmetical relation in terms of a and b, so we see that '$\alpha(a) = b$' is arithmetical, as desired.

The paper [FvO16] shows that, when Theorem 5.3.13 is applied to Kleene's first model and a type-2 functional $\mathbb{N}^\mathbb{N} \rightharpoonup \mathbb{N}$, the result is equivalent to Kleene's original notion of computability w.r.t. a higher-order functional [Kle59]. In particular, the functions $\mathbb{N} \to \mathbb{N}$ which are representable w.r.t. ι_E are precisely the *hyper*arithmetical functions ([FvO16], Corollary 4.1). Now, if $\iota_{\hat{E}} i$ were to factor through ι_E, then every hyperarithmetical function $\mathbb{N} \to \mathbb{N}$ would be representable w.r.t. $\iota_{\hat{E}} i$ as well. But claim 1 tells us that this is not the case, since there are certainly total hyperarithmetical functions which are not arithmetical (e.g., the characteristic function of a hyperarithemtical set which is not arithmetical).

Claim 3 immediately follows from claim 2 by Theorem 5.3.13.

Finally, in order to prove claim 4, suppose for the sake of contradiction that $\sigma \in (\mathcal{BK}_1)^\#$ realizes the inequality $\mathrm{rep}^{\iota_{\hat{E}} i} \leq \iota_{\hat{E}}$. Moreover, let $\rho \in (\mathcal{BK}_1)^\#$ represent $\hat{E}$ w.r.t. $\iota_{\hat{E}}$. Then it easily follows that
$$\lambda^* x.\rho \odot_{\hat{E}} (\sigma \odot_{\hat{E}} x) \in (\mathcal{BK}_1)^\#$$
represents E w.r.t. $\iota_{\hat{E}} \circ i$, which is not the case by claim 3.

5.4 The third-order case

In this section, we investigate to what extent Theorem 5.1.17 and Theorem 5.3.13 can be extended to type-3 functionals. We obtain a partial generalization: for every chain-complete PCA A and $\Phi \in \mathcal{B}_3 A$, we may obtain a morphism of PCAs $\iota_\Phi\colon A \to A[\Phi]$ such that:

1. Φ is representable w.r.t. ι_Φ;
2. if Φ is representable w.r.t. a decidable and chain-continuous $f\colon A \multimap B$, then f factors through ι_Φ in a *maximal* (but not unique) way.

Moreover, we show that, under certain conditions, the set of *functions* from $\mathcal{B}A$ that are forced to be representable if Φ is representable, form a finite extension of $(\mathcal{B}A)^\#$.

5.4.1 Adjoining a type-3 functional

As the authors of [FvO16] mention as well, there is a fundamental obstacle when studying the representability of type-3 functionals. A PCA A can only 'talk about' first-order functions by means of their representers. So as far as A is concerned, only functions that have a representer really exist. For representing type-2 functionals F, this is not a problem. On the contrary, is makes the task easier: if a (type-1) function α does not have a representer, then we do not have to worry about α when constructing a representer for F. In other words, from the point of view of A, the domain of F may look smaller than it actually is. This was also the reason why the 'master function' α in Theorem 5.3.13 had to be constructed using a transfinite recursion.

For a third-order functional Φ, on the other hand, the second-order functionals serve as *inputs*, and A may lose information contained in these inputs. More precisely, there may be two distinct $F, G \in \mathcal{B}_2 A$ such that $\Phi(F)$ and $\Phi(G)$ do not have a common lower bound, but which are the same from the point of view of A. Let us give an example of this phenemenon.

Example 5.4.1. Again, we let A be Kleene's first model $\mathcal{K}_1$. Consider the third-order functional $\Phi \in \mathcal{B}_3\mathcal{K}_1$ defined by $\mathrm{dom}\,\Phi = \{F \in \mathcal{B}_2\mathcal{K}_1 \mid \mathbb{N}^\mathbb{N} \subseteq \mathrm{dom}\,F\}$ and

$$\Phi(F) = \begin{cases} 0 & \text{if } \forall \alpha \in \mathbb{N}^\mathbb{N}\,(F(\alpha) = 0); \\ 1 & \text{if } \exists \alpha \in \mathbb{N}^\mathbb{N}\,(F(\alpha) > 0). \end{cases}$$

We leave it to the reader to check that Φ is actually in $\mathcal{B}_3\mathcal{K}_1$. Recall from Example 5.1.10 the functional $F \colon \mathbb{N}^\mathbb{N} \to \mathbb{N}$ defined by

$$F(\alpha) = \begin{cases} 0 & \text{if } \alpha \text{ is recursive;} \\ 1 & \text{otherwise.} \end{cases}$$

Consider also the function $G \in \mathcal{B}_2\mathcal{K}_1$, which is 0 on total functions, and undefined on non-total functions. Then we clearly have $\Phi(F) = 1 \neq 0 = \Phi(G)$. However, from the point of view of $\mathcal{K}_1$, the functionals F and G are equal, and both are represented by an index for the constant 0 function.

This does not yet exclude the possibility that, as for the second-order case, we can construct a partial function $\alpha \colon \mathbb{N} \rightharpoonup \mathbb{N}$ such that Φ becomes representable w.r.t. $\iota_\alpha \colon \mathcal{K}_1 \to \mathcal{K}_1[\alpha]$. However, we can adjust the example above to show that this is impossible as well. Define $F_\alpha \colon \mathbb{N}^\mathbb{N} \to \mathbb{N}$ by:

$$F_\alpha(\beta) = \begin{cases} 0 & \text{if } \beta \text{ is representable w.r.t. } \iota_\alpha; \\ 1 & \text{otherwise.} \end{cases}$$

Once again, we have $\Phi(F_\alpha) = 1 \neq 0 = \Phi(G)$, but $\mathcal{K}_1[\alpha]$ cannot distinguish between F_α and G. This means that Φ is not representable w.r.t. to ι_α for *any* partial function $\alpha \colon \mathbb{N} \rightharpoonup \mathbb{N}$.

This example shows that a construction as in the previous section, where $A[F]$ was of the form $A[\alpha]$ for an $\alpha \in \mathcal{B}A$, simply cannot work in the type-3 case. On the other hand, the example above would clearly be blocked if we move to a PCA in which every (partial) function has a representer. Fortunately, we have such a PCA, namely $\mathcal{B}A$. This leads to the following 'lax' result about the type-3 case.

Theorem 5.4.2. *Let A be a chain-complete PCA and let $\Phi \in \mathcal{B}_3 A$. Then there exists a c.d. and chain-continuous morphism of PCAs $\iota_\Phi \colon A \to A[\Phi]$ such that:*

- *(i) Φ is representable w.r.t. ι_Φ;*
- *(ii) if $f \colon A \multimap B$ is decidable and chain-continuous, and Φ is representable w.r.t. g, then there exists a* largest *$f' \colon A[\Phi] \multimap B$ such that $f' \iota_\Phi \simeq f$.*

Proof. Define $\tilde{\Phi} \in \mathcal{B}_2 \mathcal{B}A$ by:

$$\tilde{\Phi}(F)(a) \simeq \Phi(\alpha \mapsto F(\alpha)(\mathsf{i})) \quad \text{for } F \in \mathcal{B}\mathcal{B}A \text{ and } a \in A.$$

Observe that if $\tilde{\Phi}(F)$ is defined, then it is a constant function. We leave it to the reader to check that $\tilde{\Phi}$ is well-defined and in $\mathcal{B}_2\mathcal{B}A$. If $F \in \mathcal{B}_2 A$, then we define $\hat{F} \in \mathcal{B}\mathcal{B}A$ by $\hat{F}(\alpha) \simeq \widehat{F(\alpha)}$. Since

$$\tilde{\Phi}(\hat{F})(a) \simeq \Phi(\alpha \mapsto \hat{F}(\alpha)(\mathsf{i})) \simeq \Phi(\alpha \mapsto F(\alpha)) \simeq \Phi(F),$$

we have $\tilde{\Phi}(\hat{F}) \simeq \widehat{\Phi(F)}$ for all $F \in \mathcal{B}_2 A$.

We will show that $\iota_\Phi \colon A \to A[\Phi]$ can be taken to be the composition:

$$A \xrightarrow{\quad i \quad} \mathcal{B}A \xrightarrow{\quad \iota_{\tilde{\Phi}} \quad} \mathcal{B}A[\tilde{\Phi}]$$

For the sake of readability, we will just write ι for $\iota_{\tilde{\Phi}}$, and we write $\odot$ for the application in $\mathcal{B}A[\tilde{\Phi}]$.

(i) By construction, $\tilde{\Phi}$ is representable w.r.t. ι by means of a $\rho \in (\mathcal{B}A[\tilde{\Phi}])^\#$. Let $\sigma \in (\mathcal{B}A[\tilde{\Phi}])^\#$ realize $\iota \leq \mathrm{rep}^{\iota \circ i}$, so that $\sigma \odot \alpha$ represents α w.r.t. $\iota \circ i$ for all $\alpha \in \mathcal{B}A$. We will show that

$$\tau = \lambda^* x.\rho \odot (\lambda^* y.x \odot (\sigma \odot y)) \in (\mathcal{B}A[\tilde{\Phi}])^\#$$

represents Φ w.r.t. $\iota \circ i$. In order to establish this claim, let $F \in \mathcal{B}_2 A$ be such that $\Phi(F)$ is defined, and suppose that $\beta \in \mathcal{B}A$ represents F w.r.t. $\iota \circ i$. First of all, we claim that $(\lambda^* y.x \odot (\sigma \odot y))[\beta/x]$ represents $\hat{F}$ w.r.t. ι. In order to show this, let $\alpha \in \mathcal{B}A$ be such that $\hat{F}(\alpha)$ is defined. Then:

$$(\lambda^* y.x \odot (\sigma \odot y))[\beta/x] \odot \alpha \preceq \beta \odot (\sigma \odot \alpha) \preceq \widehat{F(\alpha)} = \hat{F}(\alpha),$$

since $\sigma \odot \alpha$ represents α w.r.t. $\iota \circ i$ and β represents F w.r.t. $\iota \circ i$. This proves the claim, and it follows that

$$\tau \odot \beta \preceq \rho \odot ((\lambda^* y.x \odot (\sigma \odot y))[\beta/x]) \preceq \tilde{\Phi}(\hat{F}) = \widehat{\Phi(F)},$$

as desired.

(ii) Suppose that $s \in B^{\#}$ represents Φ w.r.t. f, and consider $\mathrm{rep}^f : \mathcal{B}A \rightharpoonup\!\circ\, B$. Since rep^f is also decidable and chain-continuous, it suffices to show that $\tilde{\Phi} \in \mathcal{B}_2\mathcal{B}A$ is representable w.r.t. rep^f. Then Theorem 5.3.13 tells us that rep^f is also a morphism $f' : \mathcal{B}A[\tilde{\Phi}] \rightharpoonup\!\circ\, B$, and Theorem 5.2.10 implies that this is the largest partial applicative morphism by means of which f factors through $\iota \circ i$.

We will show that

$$t = \lambda^* x.\mathsf{k}(s(\lambda^* y.xyj)) \in B^{\#}$$

represents $\tilde{\Phi}$ w.r.t. rep^f, where j is any element from $f(\mathsf{i}) \cap B^{\#}$. So let $F \in \mathcal{B}\mathcal{B}A$ be such that $\tilde{\Phi}(F)$ is defined, and suppose that $b \in B$ represents F w.r.t. rep^f. First of all, we claim that $(\lambda^* y.xyj)[b/x]$ represents $\alpha \mapsto F(\alpha)(\mathsf{i})$, as a second-order functional on A, w.r.t. f. In order to prove the claim, let $\alpha \in \mathcal{B}A$ be such that $F(\alpha)(\mathsf{i})$ is defined, and let $c \in B$ represent α w.r.t. f. Then $c \in \mathrm{rep}^f(\alpha)$, which means that bc is defined and in $\mathrm{rep}^f(F(\alpha))$, i.e., bc represents $F(\alpha)$ w.r.t. f. Since $j \in f(\mathsf{i})$, this yields that $((\lambda^* y.xyj)[b/x])\,c \preceq bcj$ is defined and an element of $f(F(\alpha)(\mathsf{i}))$, which proves the claim. Now we find that $s\,((\lambda^* y.xyj)[b/x]) \in f(\Phi(\alpha \mapsto F(\alpha)(\mathsf{i})))$, so it follows that $tb \preceq \mathsf{k}\,(s\,((\lambda^* y.xyj)[b/x]))$ is defined and represents $\tilde{\Phi}(F)$. In other words, we have $tb \in \mathrm{rep}^f(\tilde{\Phi}(F))$, as desired. □

The strategy for proving Theorem 5.4.2 is clearly different from the strategy we used in the type-2 case. The analogous strategy for the type-2 case would be that we try to make $F \in \mathcal{B}_2A$ representable by adjoining $\hat{F} \in \mathcal{B}\mathcal{B}A$ to $\mathcal{B}A$, where $\hat{F}$ is again defined by $\hat{F}(\alpha) \simeq \widehat{F(\alpha)}$. But we already know that this does not work: Example 5.3.16 is a counterexample. Therefore, it may seem strange that a similar strategy *does* work for the type-3 case! Let us explain why this is so. In the type-2 case, the task was to construct, given a representer of $\hat{F}$, a representer of F. Now, a representer of F eats representers of $\alpha \in \mathcal{B}A$, whereas a representer of $\hat{F}$ wants to eat α itself. So the task really is to effectively find, given a representer of α, the function α itself so that it can be fed to the representer of $\hat{F}$. But the problem is exactly that, once we add an *oracle* to $\mathcal{B}A$, this may no longer be possible, as we saw in Example 5.3.16. On the other hand, the *converse* construction obviously does work, i.e., given a representer of F, we can construct a representer of $\hat{F}$. Since we constructed $\tilde{\Phi}$ in such a way that $\tilde{\Phi}(\hat{F}) \simeq \widehat{\Phi(F)}$, this is precisely what we need to construct a representer for Φ, given a representer for $\tilde{\Phi}$. We can also put this as follows: in the business of representing Φ, the representers of type-2 functionals $F \in \mathcal{B}_2A$ are not the things to be constructed, but the things that are *given*. Unfortunately, this also reveals that the current strategy can probably not be pushed beyond the type-3 case, because in the type-4 case, things will be 'the wrong way around' again.[1]

[1]This is related to an observation from [Lon05, Section 4.3.3]. Game semantics for PCF suggests that there exists a 'duality' or 'parity' in higher-order computation, according to whether a functional serves as an *oracle* or as an *input* to an oracle.

5.4.2 The first-order effect

As we mentioned in Remark 5.3.15, the construction of $A[F]$ is not quite as universal as the construction of $A[\alpha]$, since the representability of type-2 (or higher) functionals fails to be transferable along partial applicative morphisms. Still, the PCA $A[F]$ is 'universal' in the sense that Theorem 5.3.13(ii) gives us an essentially *unique* factorization. On the other hand, Theorem 5.4.2 from the previous section does not express a genuine 'universal property' of $A[\Phi]$ in this sense, because item (ii) only says that f factors through ι_Φ in a *largest* way. We might say that this is only a 'lax' universal property (cf. Theorem 5.2.10). Nevertheless, Theorem 5.4.2 allows us to derive an interesting corollary about type-3 functionals, which concerns the representability of (type-1) *functions.*

If we add an oracle $\alpha \in \mathcal{B}A$ to A, then inevitably, other functions become representable as well. In Proposition 5.2.15(iii), we gave a precise description of the set of functions that are representable w.r.t. $\iota_\alpha \colon A \to A[\alpha]$. Using the following definition, we can formulate this result in a slightly different way.

Definition 5.4.3. *Let A be a PCA and let $F \in B_nA$ for some $n \geq 0$. We define* FOE(F) *as the set of all $\alpha \in \mathcal{B}A$ with the following property: if F is representable w.r.t. a decidable partial applicative morphism $f \colon A \rightharpoonup B$, then α is also representable w.r.t. f. We call this set* FOE(F) *the **first-order effect** of F.*

The following corollary now immediately follows from Theorem 5.1.17 and Proposition 5.2.15(iii).

Corollary 5.4.4. *For $\alpha \in \mathcal{B}A$, we have* $\mathrm{FOE}(\alpha) = (\mathcal{B}A[\alpha])^{\#} = \langle (\mathcal{B}A)^{\#} \cup \{\alpha\}\rangle$.

Note that Corollary 5.4.4 is slightly more conceptual than Theorem 5.1.17 and Proposition 5.2.15(iii), because we do not need to mention the construction of $\iota_\alpha \colon A \to A[\alpha]$ in the *formulation* of Corollary 5.4.4. (Of course, the construction of $\iota_\alpha \colon A \to A[\alpha]$ is used in the proof of Corollary 5.4.4.)

By Example 5.1.22, we have $\mathrm{FOE}(a) = (\mathcal{B}A[\hat{a}])^{\#}$ for all $a \in A$. Now let us consider the type-2 case. We will only treat the case where A is a discrete PCA, so that we do not need to worry about chain-completeness and -contintuity. If $F \in \mathcal{B}_2A$, then from Theorem 5.3.13, we can deduce that FOE(F) is of the form $(\mathcal{B}A[\alpha])^{\#}$, where α is the function provided by the theorem. The goal of this section is to derive an analogous result or the type-3 case. First, we need the following definition.

Definition 5.4.5. *Let A be a discrete PCA and let $f \colon A \rightharpoonup B$ be a partial applicative morphism. Then f is called **discrete** if $f(a) \cap f(a') = \emptyset$ for every two* distinct $a, a' \in A$.

If $f \colon A \to B$ is a morphism of PCAs, then of course, we say that f is discrete if f is discrete when considered as an arrow of OPCA_D. Explicitly, this means that $f(a)$ and $f(a')$ should not have a common lower bound for disctinct $a, a' \in A$.

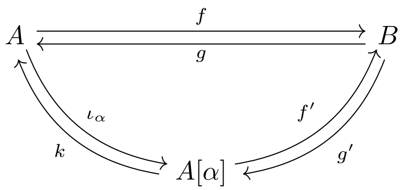
The following describes a general class of morphisms of PCAs for which the set of all $\beta \in \mathcal{B}A$ that are representable w.r.t. f is of the form $(\mathcal{B}A[\alpha])^{\#}$. The proof is based on the proof of [FvO14, Theorem 2.12].

Lemma 5.4.6. *Let A be a discrete PCA and let $f\colon A \to B$ be a morphism of PCAs which is discrete and c.d. Then there exists an $\alpha \in \mathcal{B}A$ such that:*

$$\{\beta \in \mathcal{B}A \mid \beta \textit{ is representable w.r.t. } f\} = (\mathcal{B}A[\alpha])^{\#}.$$

Proof. Throughout the proof, we view f as a projective applicative morphism $A \multimap B$. By Theorem 2.3.14, f has a right adjoint $g\colon B \hookrightarrow\!\!\!\!\multimap A$ in OPCA_D. We claim that the partial applicative morphism $gf\colon A \hookrightarrow\!\!\!\!\multimap A$ is discrete. Suppose we have $a, a' \in A$ such that $gf(a) \cap gf(a')$ is nonempty. Since f preserves the order up to a realizer, it follows that $fgf(a) \cap fgf(a')$ is also nonempty. Since $f \dashv g$, we have $fgf \simeq f$, so this implies that $f(a) \cap f(a')$ is nonempty, so $a = a'$ by the discreteness of f.

Now define the partial function $\alpha\colon A \rightharpoonup A$ by: $\alpha(a) = a'$ if and only if $a \in gf(a')$, which is well-defined by the discreteness of gf. If $\alpha(a)\downarrow$, then $f(a) \subseteq fgf(\alpha(a))$, so if $s \in B^{\#}$ realizes $fgf \leq f$, then s also represents α w.r.t. f. This means that f factors through ι_α by means of an $f'\colon A[\alpha] \to B$, which is defined simply by $f'(a) = f(a)$ for $a \in A$. Now f' is projective and c.d. as well, so it has a right adjoint $g'\colon B \hookrightarrow\!\!\!\!\multimap A[\alpha]$ in OPCA_D. Moreover, we recall from Proposition 5.1.15 that ι_α has a right adjoint $k\colon A[\alpha] \multimap A$ in OPCA_T satisfying $\iota_\alpha k \simeq \mathrm{id}_{A[\alpha]}$.



Since $f = f'\iota_\alpha$, we also have $g \simeq kg'$ by taking right adjoints, hence $\iota_\alpha g \simeq \iota_\alpha k g' \simeq g'$. This means we can assume without loss of generality that $g'(b) = g(b)$ for all $b \in B$. In particular, $g'f'(a) = gf(a)$ for all $a \in A$. But now it is clear that any representer $r \in (A[\alpha])^{\#}$ of α will also realize the inequality $g'f' \leq \mathrm{id}_{A[\alpha]}$. Combining this with $f' \dashv g'$ yields $g'f' \simeq \mathrm{id}_{A[\alpha]}$. This implies that $\beta \in \mathcal{B}A$ is representable w.r.t. $f \simeq f'\iota_\alpha$ if and only if β is representable w.r.t. ι_α; if and only if $\beta \in (\mathcal{B}A[\alpha])^{\#}$, by Proposition 5.2.15(iii). □

Remark 5.4.7. In the notation of the proof of Lemma 5.4.6, we know that $\iota_\alpha \dashv k$ yields a geometric inclusion $\mathsf{RT}(A[\alpha]) \hookrightarrow \mathsf{RT}(A)$. Moreover, we have shown that $f' \dashv g'$ is a coreflection, meaning that $\mathsf{RT}(B) \twoheadrightarrow \mathsf{RT}(A[\alpha])$ is local. Thus, under the hypotheses of Lemma 5.4.6, the image topos of the induced geometric morphism $\mathsf{RT}(B) \to \mathsf{RT}(A)$ is again a realizability topos, namely $\mathsf{RT}(A[\alpha])$; cf. [FvO14, Theorem 2.12].

Corollary 5.4.8. *Let A be a discrete PCA and let $\Phi \in \mathcal{B}_3 A$. Then there exists an $\alpha \in \mathcal{B}A$ such that* $\mathrm{FOE}(\Phi) = (\mathcal{B}A[\alpha])^{\#}$.

Proof. By Theorem 5.4.2, FOE(Φ) consists of all functions $\beta \in \mathcal{B}A$ that are representable w.r.t. ι_Φ. Thus, the statement follows from Lemma 5.4.6 if we can show that ι_Φ is discrete and c.d. Both are easy to check. $\square$

Note that, even though Corollary 5.4.8 concerns discrete PCAs, the theory of ordered PCAs is indispensable for proving it.

Samenvatting

In deze scriptie onderzoeken we de interactie tussen twee deelgebieden van de wiskunde: abstracte recursietheorie aan de ene kant, en categorie- en topostheorie aan de andere kant. Hieronder leggen we kort uit wat deze vakgebieden inhouden, op welke manier ze bij elkaar komen, en wat onze bijdrage is.

Recursietheorie en PCAs

Recursietheorie bestudeert de notie van een *berekening* of *algoritme*. Om deze begrippen vatbaar te maken voor wiskundig onderzoek, is het noodzakelijk een precieze, wiskundige, niet-intuïtieve formulering van deze begrippen te hebben. Meerdere voorstellen voor zo'n precieze notie van algoritme werden in de jaren '30 gedaan door, onder anderen, Church, Kleene en Turing, en al snel bleken hun noties equivalent te zijn. Preciezer gezegd, een partiële functie op de natuurlijke getallen is berekenbaar volgens één van deze noties van algoritme, dan en slechts dan als hij berekenbaar is volgens de andere noties. Dit leidde tot een wiskundig robuuste definitie van een berekenbare partiële numerieke functie; zo'n berekenbare functie noemen we *recursief*.

In 1945 gebruikte Kleene de theorie van recursieve functies om een interpretatie te geven van intuïtionistische rekenkunde, d.w.z., rekenkunde waarin de wet van de uitgesloten derde niet wordt aangenomen. Deze interpretatie staat bekend als de *realizeerbaarheidsinterpretatie*. Met behulp van deze interpretatie bewees Kleene het volgende resultaat. Als een uitspraak van de vorm $\forall x\, \exists y\, A(x, y)$ bewezen kan worden in intuïtionistische rekenkunde, dan bestaat er een recursieve functie $f : \mathbb{N} \to \mathbb{N}$ zodanig dat $A(n, f(n))$, voor ieder natuurlijk getal n. Dit resultaat bevestigt en onderstreept het *constructieve* karakter van intuïtionistische rekenkunde.

In deze scriptie zullen we recursietheorie vanuit een abstracter oogpunt benaderen. In 1975 formuleerde Feferman, voorbouwend op werk van Staples, de notie van een partieel combinatorische algebra (PCA). Een PCA is een verzameling, waarvan we de elementen tegelijktijd beschouwen als algoritmes en als argumenten die aan deze algoritmes gegeven kunnen worden. Om een PCA te vormen, moet de verzameling voldoen aan een aantal axioma's die het computationele karakter van de verzameling uitdrukken. Zodanig kunnen we PCAs beschouwen als algemene modellen van berekenbaarheid, en de bestudering van PCAs als abstracte recursietheorie.

De PCAs die we in deze scriptie beschouwen zijn *relatief* en *geordend*. In een relatieve PCA specificeren we een deelverzameling $A^\# \subseteq A$ bestaande uit die algoritmes die daadwerkelijk uitgevoerd kunnen worden. De elementen buiten $A^\#$ stellen weliswaar ook algoritmes voor, maar zijn niet implementeerbaar. Laten we een voorbeeld geven: er is een relatieve PCA $\mathcal{K}_2$, genaamd Kleene's tweede model, bestaande uit alle functies $\mathbb{N} \to \mathbb{N}$. In dit model codeert iedere functie $\mathbb{N} \to \mathbb{N}$ een algoritme. Echter, alleen de algoritmes gegeven door *recursieve* functies zijn daadwerkelijk uitvoerbaar. In dit model bestaat $\mathcal{K}_2^\#$ dus slechts uit de recursieve functies. In een geordende PCA is er op de verzameling A een partiële ordening gedefinieerd. Hierbij kunnen we 'kleiner' lezen als 'geeft meer informatie'. Laten we opnieuw een voorbeeld bekijken: er is een geordende PCA $T\mathcal{K}_1$ bestaande uit alle niet-lege deelverzamelingen van de natuurlijke getallen. De ordening is hier de voor de hand liggende ordening 'deelverzameling van'. Intuïtief kunnen we als volgt over dit model denken. We werken met een onbekend natuurlijk getal x, en een niet-lege verzameling $\alpha \in T\mathcal{K}_1$ geeft ons de informatie dat x behoort tot α. Hier geldt inderdaad dat, hoe kleiner de verzameling α is, hoe meer informatie we hebben over het onbekende getal x.

Categorietheorie

Een categorie is een abstracte wiskundige structuur bestaande uit objecten en afbeeldingen tussen deze objecten. In deze scriptie zal een speciaal soort categorie, de *topos*, in het bijzonder van belang zijn. Grothendieck introduceerde in de jaren '50 de notie van een *topos van schoven*, tegenwoordig ook wel bekend als een *Grothendieck topos*. Rond 1970 isoleerden Lawvere and Tierney een aantal 'elementaire' eigenschappen van Grothendieck topossen; een topos met deze eigenschappen noemen we een *elementaire topos*. De kwalificatie 'elementair' wil zeggen dat de eigenschappen geformuleerd kunnen worden in zuivere categorietheorie, zonder te verwijzen naar verzamelingenleer. Er bestaat een rijke theorie van elementaire topossen en afbeeldingen tussen deze, de zogeheten *geometrische morfismen*. Bovendien kan iedere elementaire topos gezien worden als een model voor hogere-orde intuïtionistische wiskunde.

De effectieve topos en realizeerbaarheidstopossen

Rond 1980 construeerde Hyland een elementaire topos gebaseerd op de theorie van recursieve functies, de zogeheten *effectieve topos*. Deze topos is *geen* Grothendieck topos, en geeft dus een nieuw model van intuïtionistische wiskunde. Het blijkt dat rekenkunde in de effectieve topos precies overeenkomt met Kleene's interpretatie. Preciezer gezegd, een uitspraak van de rekenkunde is waar in de effectieve topos dan en slechts dan als deze waar is volgens Kleene's realizeerbaarheidsinterpretatie. Op deze wijze kan de effectieve topos gezien worden als een natuurlijke uitbreiding van Kleene's realizeerbaarheid naar hogere-orde rekenkunde.

Zoals hierboven gezegd is de constructie van de effectieve topos gebaseerd op de theorie van recursieve functies. De constructie werkt meer algemeen als men uitgaat van een abstract model van berekenbaarheid, d.w.z., van een partieel combinatorische algebra. Op deze wijze kan men voor iedere PCA A een *realizeerbaarheidstopos* construeren, en deze noteren we met $\mathsf{RT}(A)$. De algemene theorie van deze constructie werd uitgewerkt door Hyland, Johnstone en Pitts.

Tussen (realizeerbaarheids)topossen kunnen we verschillende soorten afbeeldingen bekijken: functoren en de hierboven genoemde geometrische morfismen. Voor PCAs ligt een geschikte notie van morfisme minder voor de hand. In 1994 introduceerde Longley de notie van een *applicatief morfisme* tussen PCAs, en toonde aan dat deze applicatieve morfismen precies corresponderen met bepaalde functoren tussen de bijbehorende realizeerbaarheidstopossen. Op deze wijze kan de constructie van de realizeerbaarheidstopos als functorieel worden beschouwd.

Onderzoeksvragen

In deze scriptie bestuderen we de constructie die aan een relatieve geordende PCA A de realizeerbaarheidstopos $\mathsf{RT}(A)$ toekent. In het bijzonder onderzoeken we hoe constructies in categorietheorie zich manifesteren in de wereld van abstracte recursietheorie, en *vice versa*. Laten we een voorbeeld bekijken. Van twee realizeerbaarheidstopossen kunnen we het product nemen; is dit wederom een realizeerbaarheidstopos? Anders gezegd, als A en B PCAs zijn, is $\mathsf{RT}(A)\times\mathsf{RT}(B)$ dan ook van de vorm $\mathsf{RT}(C)$ voor een PCA C? Als het antwoord ja is, kunnen we dan een beschrijving geven van deze 'onderliggende' PCA C? En als het antwoord nee is, kunnen we dan een andere beschrijving geven van de productcategorie $\mathsf{RT}(A) \times \mathsf{RT}(B)$? Deze vragen worden in dit proefschrift beantwoord. Daarnaast zullen we analoge vragen beschouwen met betrekking tot het nemen van *slices* van realizeerbaarheidstopossen.

In het bovenstaande voorbeeld gaan we uit van een constructie uit de categorietheorie, maar we kunnen ook juist beginnen met een constructie uit de (abstracte) recursietheorie. In deze scriptie bestuderen we berekeningen met *orakels* en *hogere-orde functionalen*. Een orakel voor een PCA A is een partiële functie α op de verzameling A. Een berekening met orakel α kan een willekeurig (maar eindig) aantal keer een $a \in A$ kiezen en de waarde van $\alpha(a)$ opvragen bij het orakel, alvorens met een eindantwoord te komen. Dit geeft een nieuw model van berekening, oftewel een nieuwe PCA, die genoteerd wordt met $A[\alpha]$. Daarnaast bekijken we een PCA $\mathcal{B}A$ bestaande uit alle mogelijke orakels voor A. De PCAs $A[\alpha]$ en $\mathcal{B}A$ werden eerder bestudeerd door Van Oosten.

Een hogere-orde functionaal op A is een partiële functie waarvan de argumenten geen elementen van A zijn, maar functies op A. Bijvoorbeeld, een type-2 functionaal op A is een partiële functie die orakels op A als argumenten neemt, en elementen van A als uitvoer heeft. Een type-3 functionaal neemt vervolgens type-2 functionalen als argumenten, enzovoorts. We onderzoeken of een 'orakel-PCA' zoals $A[\alpha]$ geconstrueerd kan worden voor zulke hogere-orde functionalen; dit zet voort op onderzoek door Faber en Van Oosten.

Overzicht van de scriptie

De introductie (Hoofdstuk 1) bevat een uitgebreidere historische inleiding van het onderzoek, en een meer technische samenvatting van de hoofdresultaten van de scriptie.

In Hoofdstuk 2 definiëren we relatieve geordende PCAs en drie verschillende noties van een morfisme tussen PCAs. Dit resulteert in drie verschillende categorieën van PCAs, die we noteren met OPCA, OPCA_T en OPCA_D. De twee laatstgenoemde worden verkregen als een Kleislicategorie voor een monade op OPCA. Tot slot bestuderen we het bestaan van producten en coproducten in OPCA en OPCA_T.

In Hoofdstuk 3 introduceren we de realizeerbaarheidstopos $\mathsf{RT}(A)$ en behandelen we de belangrijkste eigenschappen van $\mathsf{RT}(A)$. Daarnaast verfijnen we Longley's analyse van de correspondentie tussen applicatieve morfismen en functoren tussen realizeerbaarheidstoposssen. Meer specifiek bestaat onze bijdrage erin dat we Longley's resultaten veralgemeniseren naar *relatieve* PCAs.

In Hoofdstuk 4 behandelen we het vraagstuk van het nemen van producten en *slices* van realizeerbaarheidstoposssen. Hiertoe introduceren we een veralgemeniseerde notie van PCA, die we IPCA noemen, geïnspireerd op het werk van Stekelenburg. Daarnaast leggen we een verband tussen producten in OPCA en OPCA_T, en het nemen van producten van realizeerbaarheidstoposssen *over* de categorie van verzamelingen.

Ten slotte, in Hoofdstuk 5, bestuderen we berekeningen met orakels en hogere-orde functionalen. Eerst introduceren we de orakel-PCA $A[\alpha]$ en de PCA van orakels $\mathcal{B}A$. Vervolgens onderzoeken we berekening met betrekking tot type-2 functionalen, en type-3 functionalen.

Dankwoord

Jaap, wij kenden elkaar al ruim voordat ik aan mijn promotietraject begon. Al tijdens mijn tijd als student vond ik in jou een docent aan het Mathematisch Instituut die dezelfde wiskundige interesses deelde als ik. Daarom was ik ook erg blij toen jij me een positie als jouw PhD-student aanbood. Tijdens de afgelopen (ruim) vier jaar spraken we elkaar bijna wekelijks, en zelfs toen dat 'verboden' werd, hebben we een manier gevonden om daar mee door te gaan. Dat is voor mij ongelooflijk waardevol geweest, en ik wil je heel erg bedanken voor onze gesprekken over wiskunde, het academische leven en het leven meer in het algemeen. Ieke, ik wil ook jou heel erg bedanken voor het optreden als mijn promotor, en voor het mogelijk maken van mijn promotieplaats.

I would also like to thank the members of the thesis committee, Steve Awodey, Pieter Hofstra, Giuseppe Rosolini, Alex Simpson and Thomas Streicher, for taking the time to read this thesis carefully.

The Mathematical Institute is a wonderful place to work, and I have had many fun conversations and experiences with fellow PhD students and other colleagues, both inside the HFG and outside of it. I will not attempt to mention each of you by name, simply for the fear of forgetting someone. There are, however, a few people I would like to thank in particular. First of all, my fellow inhabitants of office 4.10: Jack, Yuqing and Giacomo. I really missed having you around when we were banned from our office. Harry, bedankt voor onze samenwerking tijdens het voorbereiden en geven van Inleiding Meetkunde. Ik heb in die periode veel van je opgestoken, en het vervoeren van een krijtbord dwars door Utrecht is iets dat ik ook niet snel zal vergeten. Barbara, toen duidelijk werd dat er geld vrij kwam voor onderwijsverbeteringen, was ik één van de personen waar jij aan dacht; ik wil je erg bedanken voor die kans. Finally, I should thank the secretaries for helping me with the countless questions I have fired at them during the past years. Without you, the institute would probably have ceased to function a long time ago.

I also want to thank the people over at the Department of Philosophy and

Religious Studies, where I spent half (well, probably more than half) of my time during my studies. Albert, jou wil ik in het bijzonder bedanken; je bent in die tijd een mentorfiguur voor mij geweest.

Tino, Casper, Roos, Melanie, Samuel, Laura, Inge en Demi, jullie hebben als mijn huisgenoten een grote rol gespeeld in mijn leven de afgelopen jaren. Zeker toen de universiteit moest sluiten, was ik erg blij dat ik bij mij thuis in ieder geval nog wel mensen kon zien. Dat ik hier al ontzettend lang woon, is voor een groot deel aan jullie te danken; ik zag geen reden om weg te gaan. Ooit komt het moment dat ik ga verhuizen, en dat zal me ontzettend zwaar vallen. Gedurende de tijd dat ik hier gewoond heb, zijn er ook al veel huisgenoten vertrokken. Elsemiek, Sanne, Nicolai, Mark en Nino, jullie wil ik hier in het bijzonder noemen en bedanken.

Feike, ik wil je bedanken voor onze (bij benadering) maandelijkse videogesprekken, die doen me altijd erg goed.

Daarnaast ben ik dank verschuldigd aan een illuster gezelschap dat bekend staat als de ‘Onnoclan’. Anne, Chris, Claudia, Francis, Geert, Jerryt, Jimi, Joël, Marlies, Onno en Willemijn, hoewel we elkaar tegenwoordig minder zien dan vroeger, hebben jullie een enorme invloed gehad op mijn leven. Zonder jullie was ik niet geworden wie ik nu ben.

Tijdens mijn master heb ik het geluk gehad omgeven te zijn door een grote groep studenten die ook geïnteresseerd waren in logica. Sven, Tom, Menno, Mark, Mireia en Anne, ontzettend bedankt voor de leuke (en soms maar beperkt productieve) tijden die we hebben gehad in de wiskundebibliotheek.

Ik ben ook de Nederlandse Wiskunde Olympiade dankbaar dat ik deel mocht uitmaken van de bevlogen groep mensen die de IMO-training verzorgt. Ik heb daar erg veel van geleerd, maar vooral ook een ontzettend leuke tijd gehad. Birgit, Quintijn, Sietske en Ellen, jullie wil ik bedanken voor de prettige samenwerking bij het organiseren van de EGMO, wat een grote stempel op mijn promotietijd heeft gedrukt. Hoewel we werden geconfronteerd met een zeer onverwachte situatie, hebben we toch iets moois neergezet.

Merlijn, bedankt voor onze eindeloze gesprekken over meer en minder zinnige onderwerpen. Ik ben blij dat dat, ook nu je aan de andere kant van de wereld woont, nog steeds doorgaat.

Opa en Oma, bedankt voor de aanmoediging die jullie me altijd hebben gegeven. Pake en Beppe, foar jimme jildt itselde; ik fyn it spitich dat jimme dit net mear meimeitsje kinne. Heit, Mem en Lisanne, myn tank oan jimme is miskien wol grutter as dy oan alle oaren by elkoar. Sûnder jimme stipe, yn wurd en died, hie ik nea sa fier komme kinne.

Curriculum Vitae

Jetze Zoethout was born on 22 September 1994 in Leeuwarden, the Netherlands. He obtained his high school degree at the Christelijk Gymnasium Beyers Naudé. He participated twice in the International Mathematical Olympiad, obtaining a bronze medal in 2011 (Amsterdam) and a gold medal in 2012 (Mar del Plata, Argentina).

In 2012, he started bachelor programs in Mathematics and Philosophy at Utrecht University, obtaining both degrees *cum laude* in 2015. Subsequently, he enrolled in research master programs in Mathematics and Philosophy in Utrecht, both of which he completed *cum laude*. In February 2018, he started his PhD project under supervision of prof. dr. Ieke Moerdijk and dr. Jaap van Oosten. During his studies, he was also active for the Dutch Mathematical Olympiad, as a tutor for the training program and as an organizer of the European Girls' Mathematical Olympiad 2020.

He defended his PhD thesis, entitled *Computability Models and Realizability Toposes*, on 30 May 2022.

Bibliography

[ABS02] S. Awodey, L. Birkedal, and D. S. Scott. Local realizability toposes and a modal logic for computability. *Math. Struct. in Comp. Science*, 12(3):319–334, 2002.

[Bez85] M. Bezem. Isomorphisms between HEO and $\mathrm{HRO}^{\mathrm{E}}$, ECF and $\mathrm{ICF}^{\mathrm{E}}$. *Journal of Symbolic Logic*, 50(2):359–371, 1985.

[Bie08] B. Biering. *Dialectica interpretations: a categorical analysis.* PhD thesis, IT University of Copenhagen, 2008. Available online.

[BvO02] L. Birkedal and J. van Oosten. Relative and modified relative realizability. *Annals of Pure and Applied Logic*, 118(1–2):115–132, 2002.

[Car95] A. Carboni. Some free constructions in realizability and proof theory. *Journal of Pure and Applied Algebra*, 103(2):117–148, 1995.

[CC82] A. Carboni and R. Celia Magno. The free exact category on a left exact one. *Journal of the Australian Mathematical Society*, 33(3):195–201, 1982.

[CFS88] A. Carboni, P.J. Freyd, and A. Scedrov. A categorical approach to realizability and polymorphic types. In M. Main, A. Melton, M. Mislove, and D. Schmidt, editors, *Mathematical Foundations of Programming Language Semantics*, Lecture Notes in Computer Science, pages 23–42. Springer-Verlag, 1988.

[Cur30] H.B. Curry. Grundlagen der kombinatorischen logik. *American Journal of Mathematics*, 52(3):509–536, 1930.

[CV98] A. Carboni and E. M. Vitale. Regular and exact completions. *Journal of Pure and Applied Algebra*, 125(1–3):79–116, 1998.

[Fef75] S. Feferman. A language and axioms for explicit mathematics. In J. N. Crossley, editor, *Algebra and Logic*, volume 450 of *Lecture Notes in Mathematics*, pages 87–139. Springer-Verlag, 1975.

[Fre14] J. Frey. *A fibrational study of realizability toposes.* PhD thesis, Université Paris Diderot (Paris 7), 2014. Available at arXiv 1403.3672.

[Fre19] J. Frey. Characterizing partitioned assemblies and realizability toposes. *Journal of Pure and Applied Algebra*, 223(5):2000–2014, 2019.

[FS79] M.P. Fourman and D.S. Scott. Sheaves and logic. In M.P. Fourman, D.S. Scott, and C.J. Mulvey, editors, *Applications of Sheaves*, volume 753 of *Lecture Notes in Mathematics*, pages 302–401. Springer-Verlag, 1979.

[FvO14] E. Faber and J. van Oosten. More on geometric morphisms between realizability toposes. *Theory and Applications of Categories*, 29(30):874–895, 2014.

[FvO16] E. Faber and J. van Oosten. Effective operations of type 2 in pcas. *Computability*, 5(2):127–145, 2016.

[Göd58] K. Gödel. Über eine bisher noch nicht benützte Erweiterung des finiten Standpuktes. *Dialectica*, 12(3–4):280–287, 1958.

[HJP80] J. M. E. Hyland, P. T. Johnstone, and A. M. Pitts. Tripos theory. *Math. Proc. Camp. Phil. Soc.*, 88(2):205–232, 1980.

[Hof04] P.J.W. Hofstra. Relative completions. *Journal of Pure and Applied Algebra*, 192(1–3):129–148, 2004.

[Hof06] P.J.W. Hofstra. All realizability is relative. *Math. Proc. Camb. Phil. Soc.*, 141(2):239–264, 2006.

[HRR90] J.M.E. Hyland, E.P. Robinson, and G. Rosolini. The discrete objects in the effective topos. *Proceedings of the London Mathematical Society*, 60(1):1–36, 1990.

[HS21] P.J.W. Hofstra and P. J. Scott. Aspects of categorical recursion theory. In C. Casadio and P. J. Scott, editors, *Joachim Lambek: The Interplay of Mathematics, Logic, and Linguistics*, volume 20 of *Outstanding Contributions to Logic*, pages 219–269. Springer, Cham, 2021.

[HvO03] P.J.W. Hofstra and J. van Oosten. Ordered partial combinatory algebras. *Math. Proc. Camb. Phil. Soc.*, 134(3):445–463, 2003.

[Hyl82] J. M. E. Hyland. The effective topos. In A. S. Troelstra and D. van Dalen, editors, *The L. E. J. Brouwer Centenary Symposium*, volume 110 of *Studies in logic and the foundations of mathematics*, pages 165–216. North-Holland Publishing Company, 1982.

[Hyl88] J.M.E. Hyland. A small complete category. *Annals of Pure and Applies Logic*, 40(2):135–165, 1988.

[Joh77] P. T. Johnstone. *Topos Theory*. Academic Press, 1977. Paperback edition: Dover reprint 2014.

[Joh81] P. T. Johnstone. Factorization theorems for geometric morphisms, I. *Cahiers de topologie et géométrie différentielle catégoriques*, 22(1):3–17, 1981.

[Joh02] P. T. Johnstone. *Sketches of an Elephant: a Topos Theory Compendium*, volume 1&2. Oxford University Press, 2002.

[Joh13] P. T. Johnstone. Geometric morphisms of realizability toposes. *Theory and Applications of Categories*, 28(9):241–249, 2013.

[Kih21] Takayuki Kihara. Lawvere-tierney topologies for computability theorists. Available at arXiv 2106.03061, 2021.

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

[Kle59] S.C. Kleene. Recursive functionals and quatifiers of finite types I. *Trans. Amer. Math. Soc.*, 91(1):1–52, 1959.

[Kle73] S.C. Kleene. Realizability: a retrospective survey. In A.R.D. Mathias and H. Rogers, editors, *Cambridge Summer School in Mathematical Logic*, volume 337 of *Lecture Notes in Mathematics*, pages 95–112. Springer-Verlag, 1973.

[KM77] A. S. Kechris and Y. N. Moschovakis. Recursion in higher types. In J. Barwise, editor, *Handbook of mathematical logic*, volume 90 of *Studies in Logic and the Foundations of Mathematics*, chapter C.6, pages 681–737. North-Holland Publishing Company, 1977.

[Kre59] G. Kreisel. Interpretation of analysis by means of constructive functionals of finite types. In A. Heyting, editor, *Constructivity in mathamatics*, Studies in Logic and the Foundations of Mathematics, pages 101–128. North-Holland Publishing Company, 1959.

[Kri09] J.-L. Krivine. Realizability in classical logic. In *Interactive models of computation and program behavior*, volume 27 of *Panoramas et Synthèses*, pages 197–229. Société mathématique de France, 2009.

[KV65] S.C. Kleene and R.E. Vesley. *The foundations of intuitionistic mathematic, especially in relation to recursive functions*, volume 39 of *Studies in Logic and the Foundations of Mathematics*. North-Holland Publishing Company, 1965.

[Lif79] V. Lifschitz. CT_0 is stronger than CT_0!. *Proceedings of the American Mathematical Society*, 73(1):101–106, 1979.

[LN15] J. Longley and D. Normann. *Higher-Order Computability.* Springer-Verlag Berlin Heidelberg, 2015.

[Lon94] J. Longley. *Realizability Toposes and Language Semantics.* PhD thesis, University of Edinburgh, 1994.

[Lon05] J. Longley. Notions of computability at higher types I. In René Cori, Alexander Razborov, Stevo Todorčević, and Carol Wood, editors, *Logic Colloquium 2000*, volume 19 of *Lecture Notes in Logic*, pages 32–142. Association for Symbolic Logic, 2005.

[LS02] P. Lietz and T. Streicher. Impredicativity entails untypedness. *Math. Struct. in Comp. Science*, 12(3):335–347, 2002.

[LvO13] S. Lee and J. van Oosten. Basis subtoposes of the effective topos. *Annals of Pure and Applied Logic*, 164(9):335–347, 2013.

[Men00] M. Menni. *Exact completions and toposes.* PhD thesis, University of Edinburgh, 2000. Available online.

[Nel47] D. Nelson. Recursive functions and intuitionistic number theory. *Trans. Amer. Math. Soc.*, 61(2):307–368, 1947.

[Pho89] W. Phoa. Relative computability in the effective topos. *Math. Proc. Camb. Phil. Soc.*, 106(3):419–422, 1989.

[Pit81] A.M. Pitts. *The Theory of Triposes.* PhD thesis, University of Cambridge, 1981. Available online.

[Pit02] A.M. Pitts. Tripos theory in retrospect. *Math. Struct. in Comp. Science*, 12(3):265–279, 2002.

[RR90] E. Robinson and G. Rosolini. Colimit completions and the effective topos. *Annals of Pure and Applies Logic*, 50(2):678–699, 1990.

[Sch24] M. Schönfinkel. Über die Bausteine der mathematischen Logik. *Mathematische Annalen*, 92(3–4):305–316, 1924.

[Sch67] M. Schönfinkel. On the building blocks of mathematical logic. In J. van Heijenoort, editor, *From Frege to Gödel: A Source Book in Mathematical Logic, 1879-1931*, pages 355–366. Harvard University Press, 1967. Translated from original German text [Sch24].

[Sco75] D.S. Scott. Lambda calculus and recursion theory. In S. Kanger, editor, *Proceedings of the Third Scandinavian Logic Symposium*, volume 82 of *Studies in Logic and the Foundations of Mathematics*, pages 154–193. North-Holland Publishing Company, 1975.

[Sta73] J. Staples. Combinator realizability of constructive finite type analysis. In A.R.D. Mathias and H. Rogers, editors, *Cambridge Summer School in Mathematical Logic*, volume 337 of *Lecture Notes in Mathematics*, pages 253–273. Springer-Verlag, 1973.

[Ste13] W. P. Stekelenburg. *Realizability Categories*. PhD thesis, Utrecht University, 2013. Available at arXiv 1301.2134.

[Tro73] A. S. Troelstra, editor. *Metamathematical Investigation of Intuitionistic Arithmetic and Analysis*, volume 344 of *Lecture Notes in Mathematics*. Springer-Verlag, 1973.

[Tro98] A.S. Troelstra. Realizability. In *Handbook of Proof Theory*, volume 137 of *Studies in Logic and the Foundations of Mathematics*, pages 407–473. Elsevier Science, 1998.

[Tro11] A.S. Troelstra. History of constructivism in the 20th century. In J. Kennedy and R. Kossak, editors, *Set Theory, Arithmetic, and Foundations of Mathematics: Theorems, Philosophies*, pages 150–179. Cambridge University Press, 2011.

[vO91] J. van Oosten. *Exercises in realizability*. PhD thesis, University of Amsterdam, 1991. Available online.

[vO97a] J. van Oosten. Extensional realizability. *Annals of Pure and Applied Logic*, 84(3):317–349, 1997.

[vO97b] J. van Oosten. The modified realizability topos. *Journal of Pure and Applies Algebra*, 116(1–3):273–289, 1997.

[vO99] J. van Oosten. A combinatory algebra for sequential functionals of finite type. In S.B. Cooper and J.K. Tuss, editors, *Models and Computability*, volume 259 of *London Mathematical Society Lecture Note Series*, pages 389–406. Cambridge University Press, 1999.

[vO02] J. van Oosten. Realizability: a historical essay. *Math. Struct. in Comp. Science*, 12(3):239–263, 2002.

[vO06] J. van Oosten. A general form of relative recursion. *Notre Dame Journal of Formal Logic*, 47(3):311–318, 2006.

[vO08] J. van Oosten. *Realizability: An Introduction to its Categorical Side*, volume 152 of *Studies in Logic and the Foundations of Mathematics*. Elsevier, 2008.

[vO11] J. van Oosten. Partial combinatory algebras of functions. *Notre Dame Journal of Formal Logic*, 52(4):431–448, 2011.

[vO14] J. van Oosten. Realizability with a local operator of A.M. Pitts. *Theoretical Computer Science*, 564:237–243, 2014.

[vOV18] J. van Oosten and N. Voorneveld. Extensions of Scott's graph model and Kleene's second algebra. *Indagationes Mathematicae*, 29(1):5–22, 2018.

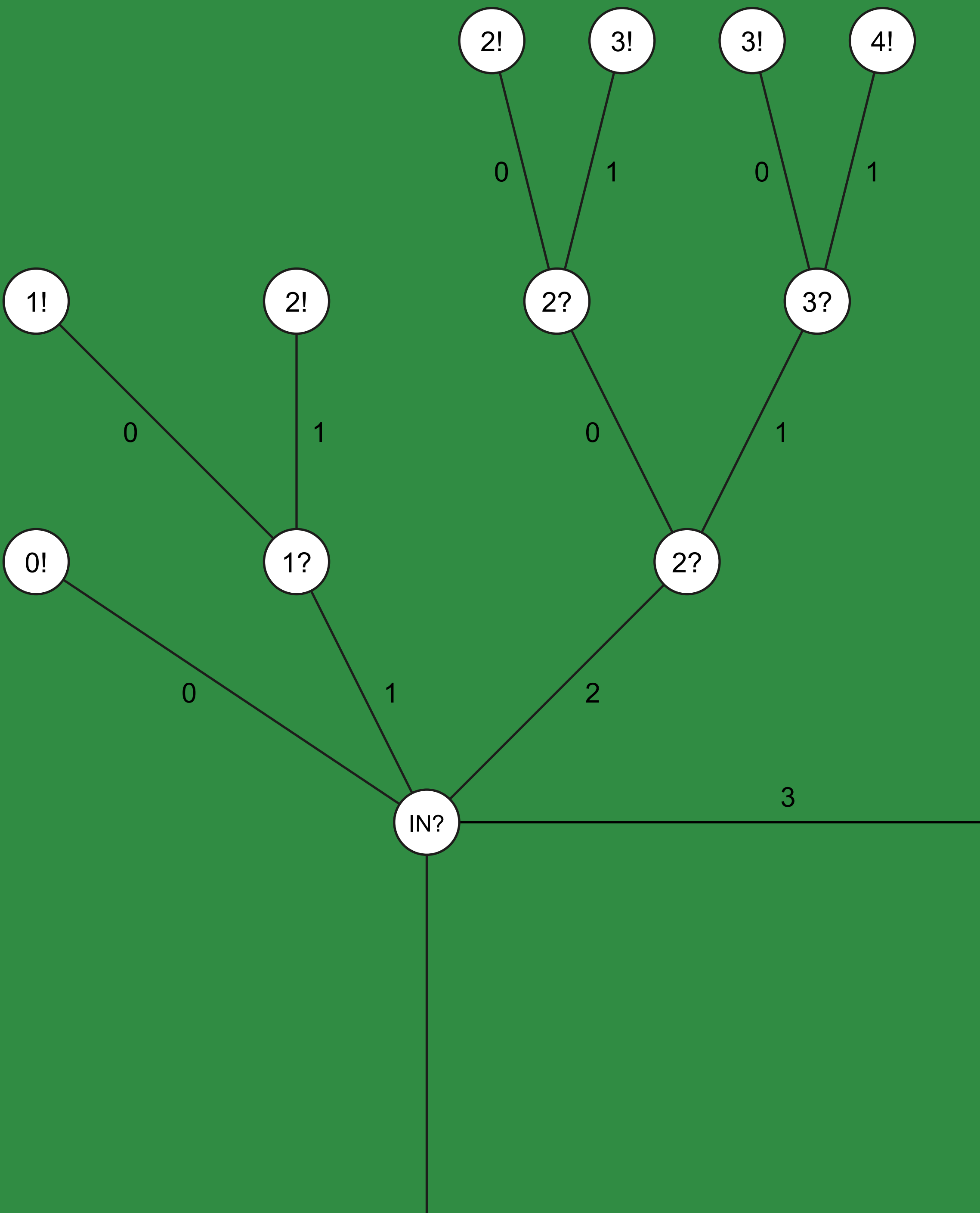
[Zoe20] J. Zoethout. Internal partial combinatory algebras and their slices. *Theory and Applications of Categories*, 35(52):1907–1952, 2020.

[Zoe21a] J. Zoethout. On (co)products of partial combinatory algebras, with an application to pushouts of realizability toposes. *Mathematical Structures in Computer Science*, 31(2):214–233, 2021.

[Zoe21b] J. Zoethout. Third-order functionals on partial combinatory algebras. Available at arXiv 2103.09000, 2021.

Index of symbols

Index of terms



2!
3!
3!
4!
0
1
0
1
1!
2!
2?
3?
0
1
0
1
0!
1?
2?
0
1
2
3
IN?