The convex powerdomain in a category of posets realized by cpos

Citation for published version:
Simpson, A 1995, The convex powerdomain in a category of posets realized by cpos. in D Pitt, D Rydeheard & P Johnstone (eds), *Category Theory and Computer Science: 6th International Conference, CTCS '95 Cambridge, United Kingdom, August 7–11, 1995 Proceedings.* vol. 953, Lecture Notes in Computer Science, vol. 953, Springer-Verlag GmbH, pp. 117-145. https://doi.org/10.1007/3-540-60164-3_23

Digital Object Identifier (DOI):
10.1007/3-540-60164-3_23

Link:
Link to publication record in Edinburgh Research Explorer

Document Version:
Peer reviewed version

Published In:
Category Theory and Computer Science

The convex powerdomain in a category of posets realized by cpos

Alex K. Simpson

LFCS, Department of Computer Science, University of Edinburgh,
JCMB, The King's Buildings, Edinburgh, EH9 3JZ
Email: Alex.Simpson@dcs.ed.ac.uk

Abstract. We construct a powerdomain in a category whose objects are posets of data equipped with a cpo of "intensional" representations of the data, and whose morphisms are those monotonic functions between posets that are "realized" by continuous functions between the associated cpos. The category of cpos is contained as a full subcategory that is preserved by lifting, sums, products and function spaces. The construction of the powerdomain uses a cpo of binary trees, these being intensional representations of nondeterministic computation. The powerdomain is characterized as the free semilattice in the category. In contrast to the other type constructors, the powerdomain does not preserve the subcategory of cpos. Indeed we show that the powerdomain has interesting computational properties that differ from those of the usual convex powerdomain on cpos. We end by considering the solution of recursive domain equations. The surprise here is that the limit-colimit coincidence fails. Nevertheless, by moving to a setting in which one considers "realizability" at the level of functors, algebraic compactness is achieved.

1 Introduction

In a recent paper [2], Anderson and Power suggest using certain binary trees as primitive models of nondeterministic computation. These trees are labelled at the leaves with the possible results of computation, and the branching represents the nondeterministic choices encountered along the way. When the output domain is a cpo, the set of such trees also forms a natural cpo. However, the elements of the cpo only provide "intensional" representations of nondeterminism in the sense that "extensionally" equivalent computations, i.e. ones with the same sets of possible outputs, have many different representations. Semantically one does not want to distinguish between different representations of the same extensional computation. Anderson and Power make the desired identifications by (essentially) quotienting the cpo of trees in the category of cpos. In doing so they recover the standard convex powerdomain construction on cpos.

In this paper we consider an alternative approach for dealing with such cpos of intensional representations of computational behaviour. Rather than quotienting by the desired extensional equivalence, we retain the existing intensional cpo and we equip it with its intended equivalence relation as extra structure. Actually, in this paper we assume that the desired equivalence relation is derived from a more primitive preorder. This is intuitively reasonable. In many situations, one thinks of an extensional notion of computational behaviour as being determined by a class of observable tests on computations. Such tests determine, in the first instance, a preorder defined by $x \precsim y$ if and only if y satisfies any test that x satisfies. A natural notion of behavioural equivalence is then easily derived from the preorder.

The above ideas lead to the consideration of a category each of whose objects is a cpo equipped with a (suitable) preorder. There is a natural notion of morphism between such objects, corresponding to the idea that a program should compute with intensional representations, and it should do so in an extensionally meaningful way. Thus a program should determine to a continuous function between the underlying cpos that respects the extensional preorder. Further, we do not wish to distinguish between two programs that have the same observational behaviour on extensionally equivalent data. Therefore a morphism should be a function between equivalence classes under extensional equivalence that is "realized" by some continuous function that preserves the preorder.

In Section 2 we give a formal presentation of the category motivated above. It turns out to have all the basic structure that one would expect of a category of domains. Moreover, the category of cpos is included as a full subcategory. Therefore our category extends the usual universe of denotational semantics. Further, the subcategory of cpos is closed under lifting, sums, products and function spaces in the larger category. Indeed deterministic programs are given their usual semantics when interpreted in our category.

However, the main goal of this paper is to treat nondeterministic computation in this way. For this we base our construction on the cpo of binary trees discussed above. In Section 3 we define an appropriate preorder over this cpo, thereby obtaining a powerdomain, which is characterized, in Section 4, as giving the free semilattice in our category. Then in Section 5 we consider some of its computational properties. In contrast to the other type constructors, the powerdomain is shown not to preserve the subcategory of cpos. It seems that this fact has direct computational relevance to issues concerning a semantic treatment of nondeterministic computability. We also consider the nondeterministic version of PCF investigated by Sieber [16]. There he showed that full abstraction fails for an interpretation of the language using the standard convex powerdomain on cpos. We show that Sieber's counterexample is not available when the language is interpreted using our powerdomain. Thus it appears that our powerdomain may help with issues of full abstraction. However, we do not know if our powerdomain does give a fully abstract model of Sieber's language.

In Section 6 we investigate the solution of recursive domain equations. Such solutions cannot be constructed as "bilimits" of ω-chains as the limit-colimit coincidence fails in our category. Nevertheless, other techniques are available for constructing solutions. Indeed our category is algebraically compact in an appropriate sense, which involves extending the notions of "realizability" to the level of categories and functors between them.

Finally, in Section 7 we discuss possible developments of our work.

2 The realizability categories

In this section we define the realizability categories we are interested in, and establish their basic structure. First some preliminaries. By *pointed poset* we mean a poset with least element, for which we usually write $\bot$. A monotonic function between two pointed posets is said to be *strict* if it preserves the least element. A monotonic function in two arguments is said to be *bistrict* if it is strict in each argument separately. We write **Pposet** for the category of pointed posets and monotonic functions, and $\mathbf{Pposet}_\bot$ for its subcategory of strict monotonic functions. A monotonic function between posets is said to be *continuous* if it preserves existing least-upper-bounds (lubs) of ascending ω-chains. By a *cpo* we mean a pointed poset for which every ascending ω-chain has a lub. A subset of a cpo is said to be *ω-inductive* if it is closed under lubs of ascending ω-chains. A binary relation on a cpo X is said to be *ω-inductive* if it forms an ω-inductive subset of $X \times X$. We write **Cpo** for the category of cpos and continuous functions, and $\mathbf{Cpo}_\bot$ for the subcategory of strict continuous functions. We assume that the reader has a basic knowledge of enriched category theory [11]. We shall enrich over various categories, always taking the monoidal structure to be given by cartesian product.

For denotational semantics, the essential properties of **Cpo** are that it is cartesian closed and that it has a least-fixed-point operator characterized as the unique fixed-point operator satisfying a condition known as uniformity. The essential properties of $\mathbf{Cpo}_\bot$ are: it is symmetric monoidal closed, it is bicartesian, it has a strong "lift" comonad for which **Cpo** is isomorphic to the co-Kleisli category (all the structure so far gives $\mathbf{Cpo}_\bot$ as a model of intuitionistic linear type theory [3]), and it is algebraically compact for a wide class of endofunctors.

The goal of this section is to establish analogues of **Cpo** and of $\mathbf{Cpo}_\bot$ based on the idea of equipping cpos of intensional representations of data with extensional preorders. These categories will retain the essential properties of **Cpo** and $\mathbf{Cpo}_\bot$ highlighted above. Our analogous categories will be called $\mathbf{Q}\boldsymbol{\omega}\mathbf{P}$ (Quotients of ω-inductive Preorders) and $\mathbf{Q}\boldsymbol{\omega}\mathbf{P}_\bot$ respectively.

As motivated in the introduction, an object A of either of these categories will consist of a cpo, $(\|A\|, \sqsubseteq_A)$, of intensional realizers together with a preorder $\precsim_A$ corresponding to the order induced by extensional observations. We require that $\precsim_A$ satisfy the following properties (omitting subscripts):

1. $x \sqsubseteq y$ implies $x \precsim y$,
2. $\precsim$ is ω-inductive, and
3. $x \precsim \bot$ implies $x = \bot$ (where $\bot$ is the least element of $\|A\|$).

We call any preorder $\precsim$ on $\|A\|$ satisfying the above properties *admissible*. Conditions 1 and 2 on admissibility are fundamental to the technical development throughout the paper. In contrast, condition 3 may be omitted without loss. It is included only because it leads to certain minor technical simplifications.

As motivated in the introduction, we want the morphisms from A to B in $\mathbf{Q}\boldsymbol{\omega}\mathbf{P}$ to be determined by those continuous functions f from $\|A\|$ to $\|B\|$ that preserve the preorder (i.e. such that $x \precsim_A y$ implies $f(x) \precsim_B f(y)$). However, we would like to identify morphisms induced by different continuous functions whose behaviour is extensionally indistinguishable. More precisely, we want any f and g for which $x \approx_A y$ implies $f(x) \approx_B g(y)$ (where we write $\approx$ for the equivalence relation induced by $\precsim$) to determine the same morphism. Thus the morphisms from A to B should be equivalence classes of preorder-preserving continuous functions modulo the stated equivalence. However, we prefer to adopt an equivalent viewpoint in which morphisms are functions. Note that each equivalence class of preorder-preserving continuous functions determines a distinct function from $\|A\|/\approx_A$ to $\|B\|/\approx_A$. Thus it is natural to take the morphisms from A to B to be those functions from $\|A\|/\approx_A$ to $\|B\|/\approx_A$ that arise in this way.

In fact, we shall adopt a slightly different definition of $\mathbf{Q\omega P}$, giving a category equivalent to that sketched above. The only difference is that, in order to avoid working with quotiented sets and equivalence classes, we allow $\|A\|/\approx_A$ to be represented by a chosen set $|A|$. For this we require a "quotient" function q_A from $\|A\|$ to $|A|$ which is surjective and such that $x \approx_A y$ if and only if $q_A(x) = q_A(y)$. As q_A is surjective, $\precsim_A$ induces an obvious partial order $\leq_A$ on $|A|$. Further, we can recover $\precsim_A$ from $\leq_A$ because $x \precsim_A y$ if and only if $q_A(x) \leq_A q_A(y)$. Indeed we shall consider $|A|$, $\leq_A$ and q_A as the primitive structure on objects of $\mathbf{Q\omega P}$, and we shall derive $\precsim_A$ as above. To this end we give an intrinsic characterization of the appropriate structures. Given a pointed partial order $(|A|, \leq_A)$, a cpo $(\|A\|, \sqsubseteq_A)$ and a function q_A from $\|A\|$ to $|A|$ we say that q_A is an *admissible quotient* if it is surjective, monotonic, continuous[1] and reflects the least element. The following proposition, whose straightforward proof is omitted, shows that the concept of admissibility for q_A coincides with that for $\precsim_A$.

Proposition 2.1

1. *Given a cpo $(\|A\|, \sqsubseteq_A)$ and an admissible preorder $\precsim_A$ on $\|A\|$, the function mapping an element of $(\|A\|, \sqsubseteq_A)$ to its equivalence class in the partial order $(\|A\|/\approx, \precsim_A/\approx)$ is an admissible quotient.*
2. *Given a partial order $(|A|, \leq_A)$, a cpo $(\|A\|, \sqsubseteq_A)$ and a surjective function q_A from $\|A\|$ onto $|A|$, define $x \precsim_A y$ whenever $q_A(x) \leq_A q_A(y)$. Then $\precsim_A$ is admissible if and only if q_A is an admissible quotient.*

This motivates the definition of the structures that will form the objects of $\mathbf{Q\omega P}$.

Definition 2.2 (Realized poset) *A realized poset is a structure*

$$A \quad = \quad ((|A|, \leq_A), (\|A\|, \sqsubseteq_A), q_A)$$

where: $(|A|, \leq_A)$ is a pointed poset, $(\|A\|, \sqsubseteq_A)$ is a cpo and $q_A : \|A\| \rightarrow |A|$ is an admissible quotient.

[1] Warning! As $(|A|, \leq_A)$ need not be a cpo, we are using continuity in the wider sense between posets defined earlier.

We use the term *realized poset* because we think of the poset $(|A|, \leq_A)$ as giving the "elements" of the structure, whereas the cpo $(\|A\|, \sqsubseteq_A)$ gives hidden "realizability" information of how one may compute with the elements. We say that an element $a \in |A|$ is *realized* by any element $x \in \|A\|$ for which $q_A(x) = a$ (we also say that x *realizes* a). Clearly each x realizes a unique a (although in general a has many realizers). Given a realized poset, we derive $\precsim_A$ as in Proposition 2.1(2). Henceforth we freely use Proposition 2.1 to move between the admissibility of q_A and the admissibility of $\precsim_A$ without further comment.

The objects of **Q$\boldsymbol{\omega}$P** will be the realized posets. As discussed above, the morphisms from A to B are to be determined by the continuous functions from $\|A\|$ to $\|B\|$ that preserve the $\precsim$ preorder. Now any preorder-preserving function f is easily seen to induce a unique monotonic function ϕ from $(|A|, \leq_A)$ to $(|B|, \leq_B)$ such that $\phi \circ q_A = q_B \circ f$. Moreover, for any monotonic ϕ and continuous f such that $\phi \circ q_A = q_B \circ f$ it holds that f is preorder-preserving. Thus we are led to consider the following functions between realized posets.

Definition 2.3 (Realized function) *A realized function from a realized poset A to another B is a monotonic function ϕ from $(|A|, \leq_A)$ to $(|B|, \leq_B)$ for which there exists a continuous f from $(\|A\|, \sqsubseteq_A)$ to $(\|B\|, \sqsubseteq_B)$ such that the diagram below commutes.*

$$\begin{tikzcd}
\|A\| \arrow[r, "f"] \arrow[d, "q_A"'] & \|B\| \arrow[d, "q_B"] \\
|A| \arrow[r, "\phi"] & |B|
\end{tikzcd}$$

We say that a morphism ϕ is *realized* by any continuous f making the diagram commute, and that f *realizes* ϕ. Clearly if f realizes ϕ then ϕ is strict if and only if f is strict (this is one of the technical conveniences of requirement 3 on admissible preorders). We say that f is a *realizing function* if it is continuous and it realizes some (necessarily unique) ϕ. By the above discussion, a continuous f is a realizing function if and only if it preserves the induced preorders.

Definition 2.4 (Q$\boldsymbol{\omega}$P and Q$\boldsymbol{\omega}$P$_\perp$) ***Q$\boldsymbol{\omega}$P*** *is the category whose objects are realized posets and whose morphisms are realized functions, with the obvious identity and composition.* ***Q$\boldsymbol{\omega}$P$_\perp$*** *is the subcategory of strict realized functions.*

We note various facts about **Q$\boldsymbol{\omega}$P** and **Q$\boldsymbol{\omega}$P$_\perp$**. Under the pointwise ordering on hom-sets they form **Pposet**-categories. They do not however form **Cpo**-categories under this ordering (counterexamples will be given later). However, the hidden realizer structure in the categories does give rise to natural **Cpo**-categories "sitting above" **Q$\boldsymbol{\omega}$P** and **Q$\boldsymbol{\omega}$P$_\perp$**.

Definition 2.5 ($\boldsymbol{\omega}$P and $\boldsymbol{\omega}$P$_\perp$) ***$\boldsymbol{\omega}$P*** *is the category whose objects are realized posets and whose morphisms are realizing functions, with the obvious identity and composition.* ***$\boldsymbol{\omega}$P$_\perp$*** *is the subcategory of strict realizing functions.*

A morphism in $\boldsymbol{\omega}\mathbf{P}$ (or $\boldsymbol{\omega}\mathbf{P}_\perp$) is thus a continuous function from $(\|A\|, \sqsubseteq_A)$ to $(\|B\|, \sqsubseteq_B)$. It is easily checked that, under the pointwise ordering, the hom-sets are cpos (using the ω-inductivity of $\precsim_B$) and that composition in each case is continuous. Thus $\boldsymbol{\omega}\mathbf{P}$ and $\boldsymbol{\omega}\mathbf{P}_\perp$ are indeed **Cpo**-categories.

It is worth commenting on the different roles of the categories. The categories of main interest for semantics are $\mathbf{Q}\boldsymbol{\omega}\mathbf{P}$ and $\mathbf{Q}\boldsymbol{\omega}\mathbf{P}_\perp$, as in these categories the equality of morphisms corresponds to identical extensional behaviour. In contrast, $\boldsymbol{\omega}\mathbf{P}$ and $\boldsymbol{\omega}\mathbf{P}_\perp$ distinguish between intensionally different functions with the same extensional behaviour. Nevertheless, operations on $\mathbf{Q}\boldsymbol{\omega}\mathbf{P}$ and $\mathbf{Q}\boldsymbol{\omega}\mathbf{P}_\perp$ are often conveniently considered as being induced by operations on $\boldsymbol{\omega}\mathbf{P}$ and $\boldsymbol{\omega}\mathbf{P}_\perp$. This view will prove essential when we consider the solution of recursive domain equations in Section 6.

There are some useful functors between the different categories. There is an evident forgetful functor $U : \boldsymbol{\omega}\mathbf{P} \to \mathbf{Cpo}$ mapping A to $(\|A\|, \sqsubseteq_A)$. This has a left adjoint $I : \mathbf{Cpo} \to \boldsymbol{\omega}\mathbf{P}$ which maps any cpo $(X, \sqsubseteq)$ to the realized poset $((X, \sqsubseteq), (X, \sqsubseteq), 1_X)$. I is injective on objects and full and faithful, and thus exhibits **Cpo** as a full coreflective subcategory of $\boldsymbol{\omega}\mathbf{P}$. Further, these functors are **Cpo**-enriched and the adjunction holds in the enriched sense. There is also a **Pposet**-functor $Q : \boldsymbol{\omega}\mathbf{P} \to \mathbf{Q}\boldsymbol{\omega}\mathbf{P}$ which is the identity on objects and which maps any realizing $f : \|A\| \to \|B\|$ to the unique $\phi : |A| \to |B|$ it realizes. Q is full and it is faithful on those objects of $\boldsymbol{\omega}\mathbf{P}$ that lie in the image of I. Thus **Cpo** is also contained as a full subcategory of $\mathbf{Q}\boldsymbol{\omega}\mathbf{P}$. However, Q is certainly not faithful in general. All the functors described above cut down to functors between the relevant strict subcategories with the same properties. To complete the picture, there are also the evident inclusion functors from the strict categories to their containing categories. These all have left-adjoints giving "lift" functors in the different categories.

We now turn to the categorical structure of $\mathbf{Q}\boldsymbol{\omega}\mathbf{P}$ and $\mathbf{Q}\boldsymbol{\omega}\mathbf{P}_\perp$ useful for interpreting the usual type constructors. We assume the reader is familiar with the basic constructions on cpos [14]. We write: $\times$ for binary product in both **Cpo** and $\mathbf{Cpo}_\perp$; $\mathbf{0}_\perp$ for the terminal object in **Cpo** and zero object in $\mathbf{Cpo}_\perp$; $+$ for binary coproduct in $\mathbf{Cpo}_\perp$ (coalesced sum); $\otimes$ for smash product in $\mathbf{Cpo}_\perp$; $\mathbf{1}_\perp$ for its unit (Sierpinski space); $\mathbb{N}_\perp$ for the natural number object in $\mathbf{Cpo}_\perp$; and finally L for the lift functor on both **Cpo** and $\mathbf{Cpo}_\perp$. We note that all these operations are inherited from their obvious counterparts in **Pposet** and $\mathbf{Pposet}_\perp$, and we shall use the same notation for the associated operations there.

We now define the analogous operations on realized posets, and again we retain the same notation. The objects $\mathbf{0}_\perp$, $\mathbf{1}_\perp$ and $\mathbb{N}_\perp$ are given as realized posets by applying $I : \mathbf{Cpo} \to \boldsymbol{\omega}\mathbf{P}$ to their cpos. For the other constructions we make the following definitions (using self-explanatory notation).

$A \times B$ is defined by

$$\begin{aligned}
(|A \times B|, \leq_{A\times B}) &= (|A|, \leq_A) \times (|B|, \leq_B) \text{ in } \mathbf{Pposet}_\perp, \\
(\|A \times B\|, \sqsubseteq_{A\times B}) &= (\|A\|, \sqsubseteq_A) \times (\|B\|, \sqsubseteq_B) \text{ in } \mathbf{Cpo}_\perp, \\
q_{A\times B}(\langle x, y\rangle) &= \langle q_A(x), q_B(y)\rangle.
\end{aligned}$$

$A + B$ is defined by

$$\begin{aligned}
(|A+B|, \leq_{A+B}) &= (|A|, \leq_A) + (|B|, \leq_B) \text{ in } \mathbf{Pposet}_\perp, \\
(\|A+B\|, \sqsubseteq_{A+B}) &= (\|A\|, \sqsubseteq_A) + (\|B\|, \sqsubseteq_B) \text{ in } \mathbf{Cpo}_\perp, \\
q_{A+B}(z) &= \begin{cases} \perp & \text{if } z = \perp, \\ \mathit{inl}(q_A(x)) & \text{If } z = \mathit{inl}(x), \text{ where } x \neq \perp, \\ \mathit{inr}(q_B(y)) & \text{If } z = \mathit{inr}(x), \text{ where } x \neq \perp. \end{cases}
\end{aligned}$$

$A \otimes B$ is defined by:

$$\begin{aligned}
(|A \otimes B|, \leq_{A\otimes B}) &= (|A|, \leq_A) \otimes (|B|, \leq_B) \text{ in } \mathbf{Pposet}_\perp, \\
(\|A \otimes B\|, \sqsubseteq_{A\otimes B}) &= (\|A\|, \sqsubseteq_A) \otimes (\|B\|, \sqsubseteq_B) \text{ in } \mathbf{Cpo}_\perp, \\
q_{A\otimes B}(z) &= \begin{cases} \perp & \text{if } z = \perp, \\ \langle q_A(x), q_B(y)\rangle & \text{if } z = \langle x, y\rangle, \text{ where } x, y \neq \perp. \end{cases}
\end{aligned}$$

LA is defined by:

$$\begin{aligned}
(|LA|, \leq_{LA}) &= L(|A|, \leq_A) \text{ in } \mathbf{Pposet}_\perp, \\
(\|LA\|, \sqsubseteq_{LA}) &= L(\|A\|, \sqsubseteq_A) \text{ in } \mathbf{Cpo}_\perp, \\
q_{LA}(z) &= \begin{cases} \perp & \text{if } z = \perp, \\ \lceil q_A(x) \rceil & \text{if } z = \lceil x \rceil. \end{cases}
\end{aligned}$$

The above definitions are easily checked to be good, i.e. the defined q functions are indeed all admissible quotients.

All the above operations were defined on realized posets using the associated constructions in **Pposet** and **Cpo**. This is not the case for the two function space constructors. We shall write $A \Rightarrow B$ for the realized poset of realized functions from A to B and $A \Rightarrow_\perp B$ for that of strict realized functions. These are given by the definitions below, which use the **Pposet**-enriched structure of $\mathbf{Q}\boldsymbol{\omega}\mathbf{P}$ and $\mathbf{Q}\boldsymbol{\omega}\mathbf{P}_\perp$, and the **Cpo**-enriched structure of $\boldsymbol{\omega}\mathbf{P}$ and $\boldsymbol{\omega}\mathbf{P}_\perp$.

$A \Rightarrow B$ is defined by:

$$\begin{aligned}
(|A \Rightarrow B|, \leq_{A\Rightarrow B}) &= \mathbf{Q}\boldsymbol{\omega}\mathbf{P}(A, B), \\
(\|A \Rightarrow B\|, \sqsubseteq_{A\Rightarrow B}) &= \boldsymbol{\omega}\mathbf{P}(A, B), \\
q_{A\Rightarrow B}(f) &= \text{the unique } \phi \text{ realized by } f.
\end{aligned}$$

$A \Rightarrow_\perp B$ is defined by:

$$\begin{aligned}
(|A \Rightarrow_\perp B|, \leq_{A\Rightarrow_\perp B}) &= \mathbf{Q}\boldsymbol{\omega}\mathbf{P}_\perp(A, B), \\
(\|A \Rightarrow_\perp B\|, \sqsubseteq_{A\Rightarrow_\perp B}) &= \boldsymbol{\omega}\mathbf{P}_\perp(A, B), \\
q_{A\Rightarrow B}(f) &= \text{the unique } \phi \text{ realized by } f.
\end{aligned}$$

In order to check that these are good definitions it is convenient to work with the induced preorders. For example, in the case of $A \Rightarrow B$ one shows first that $f \precsim_{A\Rightarrow B} g$ if and only if, for all $x \in \|A\|$, it holds that $f(x) \precsim_B g(x)$. It is then straightforward to show that $\precsim_{A\Rightarrow B}$ is admissible, using the admissibility of $\precsim_B$.

Note that the objects in the image of $I : \mathbf{Cpo} \to \boldsymbol{\omega}\mathbf{P}$ are preserved under the above operations. Thus none of the operations take one outside of the world of cpos. However, the operations are well defined on the larger universe of realized posets and, as the theorem below shows, they have the desired universal properties there.

Theorem 2.6

1. *$\mathbf{Q}\boldsymbol{\omega}\mathbf{P}$ (resp. $\boldsymbol{\omega}\mathbf{P}$) is a cartesian-closed category with finite products given by $\mathbf{0}_\perp$ and $A \times B$ and with exponentials given by $A \Rightarrow B$.*
2. *$\mathbf{Q}\boldsymbol{\omega}\mathbf{P}_\perp$ (resp. $\boldsymbol{\omega}\mathbf{P}_\perp$) has: a natural number object $\mathbb{N}_\perp$; finite products given by $\mathbf{0}_\perp$ and $A \times B$; finite coproducts given by $\mathbf{0}_\perp$ and $A + B$; a symmetric monoidal structure given by $\mathbf{1}_\perp$ and $A \otimes B$; and a closed (relative to $\otimes$) structure given by $A \Rightarrow_\perp B$.*
3. *L is the functor part of a strong comonad on $\mathbf{Q}\boldsymbol{\omega}\mathbf{P}_\perp$ (resp. $\boldsymbol{\omega}\mathbf{P}_\perp$) whose co-Kleisli category is isomorphic to $\mathbf{Q}\boldsymbol{\omega}\mathbf{P}$ (resp. $\boldsymbol{\omega}\mathbf{P}$).*
4. *The above statements all hold in their **Pposet**-enriched (resp. **Cpo**-enriched) versions.*
5. *$Q : \boldsymbol{\omega}\mathbf{P} \to \mathbf{Q}\boldsymbol{\omega}\mathbf{P}$ and $Q : \boldsymbol{\omega}\mathbf{P}_\perp \to \mathbf{Q}\boldsymbol{\omega}\mathbf{P}_\perp$ preserve all the above structure.*

The proof, although lengthy, is just a matter of checking the details.

To conclude this section we consider the canonical fixed-point operators in $\mathbf{Q}\boldsymbol{\omega}\mathbf{P}$ and $\boldsymbol{\omega}\mathbf{P}$. Consider the usual continuous function $\mathit{fix} : \|A \Rightarrow A\| \to \|A\|$ defined by:

$$\mathit{fix}(f) \quad = \quad \textstyle\bigsqcup_i f^i(\perp).$$

It is easily seen that $f \precsim_{A\Rightarrow B} g$ implies $\mathit{fix}(f) \precsim_A \mathit{fix}(g)$, using the ω-inductivity of $\precsim_A$. Thus $(A \Rightarrow A) \xrightarrow{\mathit{fix}} A$ is a morphism in $\boldsymbol{\omega}\mathbf{P}$. We write *Qfix* for the $\mathbf{Q}\boldsymbol{\omega}\mathbf{P}$ morphism that it realizes.

Theorem 2.7 *For any $\phi \in |A \Rightarrow A|$, it holds that $\mathit{Qfix}(\phi)$ is the least-fixed-point of ϕ (under $\leq_A$). Further Qfix is "uniform" in the sense that, for any $A \xrightarrow{\phi} A$, $B \xrightarrow{\psi} B$ and strict $A \xrightarrow{\theta} B$ for which the diagram below commutes,*

$$\begin{tikzcd}
A \arrow[r, "\phi"] \arrow[d, "\theta"'] & A \arrow[d, "\theta"] \\
B \arrow[r, "\psi"] & B
\end{tikzcd}$$

it holds that $\mathit{Qfix}(\psi) = \theta(\mathit{Qfix}(\phi))$.

The proof is entirely standard and hence omitted. Exactly the same properties hold of *fix* in $\boldsymbol{\omega}\mathbf{P}$. We mention that *Qfix* (and *fix*) are characterized by the property of uniformity. However, we shall not need this fact (whose proof is again standard).

3 Construction of the powerdomain

In this section we construct, for each realized poset A, a realized poset $\mathcal{P}(A)$, representing the domain of nondeterministic computations which, if they terminate, produce values in A.

We begin the construction of by describing the cpo of realizers, $\|\mathcal{P}(A)\|$. The idea is to have $\|\mathcal{P}(A)\|$ as a cpo of intensional representations of nondeterministic computations. It is natural to represent such computations as possibly infinite binary trees. The branching of the trees represents the possible nondeterministic choices encountered during the computation. The leaves of the trees represent points beyond which no more nondeterministic choices are encountered. From such a point the computation proceeds deterministically, either eventually terminating with a value in $\|A\|\backslash\{\bot\}$ (we take $\bot$ as representing nontermination) or continuing for ever. The above account distinguishes between nondeterministic nontermination — exemplified by the infinite leafless tree; and deterministic nontermination. Because of our requirement on admissible preorders that undefinedness have exactly one intensional representation, we must identify the different forms of nontermination. For technical convenience, we take the infinite leafless tree as the canonical representation of nontermination. This gives us the same class of trees considered in [2].

So far we have discussed only the elements of $\|\mathcal{P}(A)\|$. These elements do indeed have a natural partial order forming a cpo. Indeed we shall see that $(\|\mathcal{P}(A)\|, \sqsubseteq_A)$ is determined up to isomorphism as the initial solution in $\mathbf{Cpo}_\bot$ of the recursive domain equation:

$$\|\mathcal{P}(A)\| \;\;\cong\;\; \|A\| + (\|\mathcal{P}(A)\| \times \|\mathcal{P}(A)\|).$$

However, we shall require a concrete description of $\|\mathcal{P}(A)\|$. This we now develop.

We shall index the nodes of binary trees by elements of $\{0,1\}^*$ (the set of finite sequences of elements of $\{0,1\}$). We use $\sigma, \tau, \ldots$ to range over such sequences. We write: ϵ for the empty sequence; σi (where $i \in \{0,1\}$) and $\sigma\tau$ for the evident concatenated sequences; $\sigma \leq \tau$ (resp. $\sigma < \tau$) to mean σ is a prefix (resp. proper prefix) of τ; and $|\sigma|$ for the length of σ.

We give a slightly cryptic definition of the trees we are interested in. Recall that an *antichain* in a poset is a subset in which any two distinct elements are incomparable. An *($\|A\|$-labelled) computation tree, t,* is a partial function from $\{0,1\}^*$ to $\|A\|\backslash\{\bot\}$ whose domain is an antichain in $(\{0,1\}^*, \leq)$. The domain of t represents the set of leaves of the tree, and we write $\mathit{Leaves}(t)$ for this set. The set of nodes of t is recovered by:

$$\mathit{Nodes}(t) \;\;=\;\; \{\sigma \mid \text{there does not exist } \tau \in \mathit{Leaves}(t) \text{ with } \tau < \sigma\}.$$

One sees that the computation trees do indeed correspond to the trees described informally earlier.

Henceforth, we use $s, t, \ldots$ to range over computation trees. We say that t is *finite* if $\mathit{Nodes}(t)$ is finite. We say that t is *finitely generated* if $\mathit{Leaves}(t)$ is finite. Clearly finite implies finitely generated, but not vice-versa.

Define:

$$\begin{array}{rcl} \|\mathcal{P}(A)\| & = & \text{the set of } \|A\|\text{-labelled computation trees,} \\ s \sqsubseteq_{\mathcal{P}(A)} t & \text{if} & \mathit{Leaves}(s) \subseteq \mathit{Leaves}(t) \text{ and, for all } \sigma \in \mathit{Leaves}(s),\ s(\sigma) \sqsubseteq_A t(\sigma). \end{array}$$

$\|\mathcal{P}(A)\|$ is indeed a cpo with this ordering. The least element is given by the unique tree with the emptyset of leaves. Given a chain $t_0 \sqsubseteq_{\mathcal{P}(A)} t_1 \sqsubseteq_{\mathcal{P}(A)} \ldots$, its lub is defined by[2]

$$t_\omega \quad = \quad \{(\sigma, x) \mid \text{for some } i,\ \sigma \in \mathit{Leaves}(t_i) \text{ and } x = \textstyle\bigsqcup_{j \geq i} t_j(\sigma)\}.$$

It is readily checked that t_ω is indeed both a computation tree and the lub of the ascending sequence.

A useful fact is that every computation tree is the lub of an ascending sequence of finitely generated computation trees. Specifically, for any $n \geq 0$, define:

$$t\lceil_n \quad = \quad \{(\sigma, x) \in t \ \mid\ |\sigma| < n\},$$

which is obviously finitely generated. It is easily seen $t\lceil_0 \sqsubseteq_{\mathcal{P}(A)} t\lceil_1 \sqsubseteq_{\mathcal{P}(A)} \ldots$ is an ascending chain and that $t = \bigsqcup_i t\lceil_i$. Note that the map $t \mapsto t\lceil_n$ is continuous. Indeed it is the projection from $\|\mathcal{P}(A)\|$ to its n-th iterate as a solution of the recursive domain equation given earlier. Thus the equation $t = \bigsqcup_i t\lceil_i$ establishes that indeed $\|\mathcal{P}(A)\|$ is the initial solution of this equation (see [18]).

It remains to consider the additional structure on $\mathcal{P}(A)$, the partial order of extensional elements and its associated quotient map. We shall define these by first determining the desired extensional preorder $\precsim_{\mathcal{P}(A)}$ on $\|\mathcal{P}(A)\|$.

Fundamentally, we want to identify those computation trees that give the same set of possible results (including nontermination). Thus we begin by defining the set of results of a computation tree. The set of *(intensional) results of t* is the subset of $\|A\|$ defined by:

$$\mathit{Res}(t) \quad = \quad \begin{cases} \{t(\sigma) \mid \sigma \in \mathit{Leaves}(t)\} & \text{if } t \text{ is finite,} \\ \{t(\sigma) \mid \sigma \in \mathit{Leaves}(t)\} \cup \{\bot\} & \text{if } t \text{ is infinite.} \end{cases}$$

The second case includes bottom because, by König's Lemma, an infinite tree must have an infinite branch corresponding to a possible infinite execution sequence. We write $\mathcal{R}es$ for the family $\{\mathit{Res}(t) \mid t \in \|\mathcal{P}(A)\|\}$ of all possible result sets. Note that

$$\mathcal{R}es = \{X \subseteq \|\mathcal{P}(A)\| \mid X \text{ is finite nonempty, or } X \text{ is countable and contains } \bot\}.$$

As $\approx_{\mathcal{P}(A)}$ is supposed to be an extensional equivalence we cannot be interested in the particular intensional representations of values in $\mathit{Res}(t)$. Therefore we certainly want to require more of $\approx_{\mathcal{P}(A)}$ than that it equate those s and t for which $\mathit{Res}(s) = \mathit{Res}(t)$. Indeed it is natural to ask that $s \approx_{\mathcal{P}(A)} t$ holds whenever $q_A(\mathit{Res}(s)) = q_A(\mathit{Res}(t))$ (where we extend q_A to act elementwise on sets).

[2] Here and henceforth we define computation trees by giving their graphs.

One might hope to define $\precsim_{\mathcal{P}(A)}$ so that also $s \approx_{\mathcal{P}(A)} t$ only if $q_A(Res(s)) = q_A(Res(t))$. However, certain considerations will prevent us from achieving this. We shall want $\mathcal{P}(A)$ to have an associated nondeterministic choice operator $\mathcal{P}(A) \times \mathcal{P}(A) \xrightarrow{\uplus} \mathcal{P}(A)$, and this must preserve the preorder and have a continuous realizer. The preservation of the preorder forces us to identify sets which have the same "convex closure". The continuity of the realizer (coupled with the ω-inductivity of the preorder) forces us also to identify result sets that have the same set of "limit points". The necessity of making such identities is clearly spelled out by Plotkin in [14], and, for lack of space, we do not repeat the arguments here. However, the naturality of the additional identifications will be made clear by Theorem 4.1. We now turn to each of the two forms of identification in detail.

Given any preorder $\precsim$ on a set Z, we define a preorder $\precsim^{EM}$ (the *Egli-Milner preorder over* $\precsim$) on its powerset, $\wp(Z)$, by defining $X \precsim^{EM} Y$ to hold if:

1. for all $x \in X$ there exists $y \in Y$ such that $x \precsim y$, and
2. for all $y \in Y$ there exists $x \in X$ such that $x \precsim y$.

When $\approx$ is the equivalence relation induced by $\precsim$ we write $\approx^{EM}$ for the equivalence relation induced by $\precsim^{EM}$.[3] A subset $X \subseteq Z$ is called *convex* if, for all $x, y \in X$ and $z \in Z$ we have that $x \precsim z \precsim y$ implies $z \in X$. For any $X \subseteq Z$ define:

$$Conv(X) \quad = \quad \{z \in Z \mid \text{there exist } x, y \in X \text{ such that } x \precsim z \precsim y\}.$$

It is easily checked that $Conv(\cdot)$ is a closure operator mapping any subset $X \subseteq Z$ to the least convex set containing it. It is clear that $X \approx^{EM} Conv(X)$. Also $X \approx^{EM} Y$ if and only if $Conv(X) = Conv(Y)$. Thus $\precsim^{EM}$ partially orders the family of convex sets.

The preorder we are seeking on $\mathcal{P}(A)$ will contain $\precsim_A^{EM}$. It differs from $\precsim_A^{EM}$ only on account of the extra "limit point" identifications referred to above. We say that a subset $X \subseteq \|A\|$ is ω*-convex* if it is both convex under $\precsim_A$ and ω-inductive. It is easily seen that ω-convexity determines a closure operator, ω-$Conv(\cdot)$, assigning to each set X a least ω-convex subset containing it.

The extensional preorder on $\|\mathcal{P}(A)\|$ is defined by

$$s \precsim_{\mathcal{P}(A)} t \quad \text{if} \quad \omega\text{-}Conv(Res(s)) \precsim_A^{EM} \omega\text{-}Conv(Res(t)).$$

Proposition 3.1 *$\precsim_{\mathcal{P}(A)}$ is admissible.*

The importance of the proposition (whose proof is given below) is that we have now determined, up to isomorphism, an object $\mathcal{P}(A)$ of $\mathbf{Q\omega P}$. For a standard

[3] Warning! There is an ambiguity in the notation here. Although any equivalence relation is a preorder, we write $\approx^{EM}$ for the kernel of $\precsim^{EM}$, and not for the Egli-Milner preorder (indeed equivalence relation) generated by $\approx$.

definition we choose $|\mathcal{P}(A)|$ as a subset of $\wp(|A|)$, thus obtaining canonical representations for nondeterministic computations as sets of values. Define:

$$\begin{array}{rcl} |\mathcal{P}(A)| & = & \{q_A(\omega\text{-}Conv(X)) \mid X \subseteq \mathcal{R}es\}, \\ D \leq_{\mathcal{P}(A)} E & \text{if} & D \leq_A^{EM} E, \\ q_{\mathcal{P}(A)}(t) & = & q_A(\omega\text{-}Conv(Res(t))). \end{array}$$

For $\mathcal{P}(A)$ to indeed be an object of $\mathbf{Q\omega P}$, one must check that $s \precsim_{\mathcal{P}(A)} t$ if and only if $q_{\mathcal{P}(A)}(s) \leq_{\mathcal{P}(A)} q_{\mathcal{P}(A)}(t)$, and that $\leq_{\mathcal{P}(A)}$ is indeed a partial order on $|\mathcal{P}(A)|$. This is all routine.

The above definition is perhaps not as good a definition as one might hope for. In particular $|\mathcal{P}(A)|$ is defined crucially using the structure of $\|A\|$ via the definition of ω-convexity. It can be shown that this use of $\|A\|$ is unavoidable in the sense that $|\mathcal{P}(A)|$ cannot be determined from the poset $|A|$ alone, as one can find objects whose underlying posets are isomorphic, but the posets of their powerdomains are not. On the other hand, the partial order on $|\mathcal{P}(A)|$ is defined entirely in terms of the partial order on $|A|$.

One pleasant fact concerning the definition of the powerdomain is that, for an arbitrary object A of $\mathbf{Q\omega P}$, we have achieved a good representation of $\mathcal{P}(A)$ as a family of sets. For the convex powerdomain in **Cpo**, such representations are only known for certain kinds of ω-algebraic cpo [13, 14]. The simplification in our setting is due to every infinite set in $\mathcal{R}es$ containing $\perp$. In Section 5 we shall discuss the computational significance of this fact. Its technical significance is that we avoid needing any of the limiting sequences that are usually dealt with using the Lawson topology (see [13, 14]). Instead, it suffices for us to consider limits of ascending ω-chains, as in the definition of ω-convexity.

We conclude this section with the proof of Proposition 3.1.

Lemma 3.2 *If $X \subseteq \|A\|$ is finite then the following hold.*

1. *For any ascending chain $z_0 \sqsubseteq_A z_1 \sqsubseteq_A \dots$ in $\|A\|$, if, for all i, there exists $x_i \in X$ such that $z_i \precsim_A x_i$ then there exists $x \in X$ such that $\bigsqcup_i z_i \precsim_A x$.*
2. *$Conv(X) = \omega\text{-}Conv(X)$.*

Proof.

1. Let z_i be an ascending chain satisfying the condition. As X is finite, there exists some $x \in X$ such that $x = x_i$ for infinitely many i. For every i, we have $z_i \precsim_A x$, because $\precsim_A$ contains $\sqsubseteq_A$. But $\precsim_A$ is ω-inductive, so indeed $\bigsqcup_i z_i \precsim_A x$.
2. We show that $Conv(X)$ is ω-inductive. Suppose that $z_0 \sqsubseteq_A z_1 \sqsubseteq_A \dots$ is an ascending chain in $Conv(X)$. Then for each z_i there exist y_i, $x_i \in X$ such that $y_i \precsim_A z_i \precsim_A x_i$. By part 1, we have that there exists $x \in X$ such that $\bigsqcup_i z_i \precsim_A x$. Also it is clear that $y_0 \precsim_A \bigsqcup_i z_i$. So indeed $\bigsqcup_i z_i \in Conv(X)$.

Lemma 3.3

1. *If $Res(t)$ is finite then $s \precsim_{\mathcal{P}(A)} t$ if and only if $Res(s) \precsim_A^{EM} Res(t)$.*

2. *If $\bot \in Res(t)$ then $s \precsim_{\mathcal{P}(A)} t$ if and only if $\bot \in Res(s)$ and $Res(s) \subseteq \omega\text{-}Conv(Res(t))$.*
3. *$s \precsim_{\mathcal{P}(A)} t$ if and only if $Res(s) \precsim_A^{EM} \omega\text{-}Conv(Res(t))$.*

Proof.

1. Suppose $Res(t)$ is finite. Then $Res(t) \approx_A^{EM} Conv(Res(t)) = \omega\text{-}Conv(Res(t))$, the equality by Lemma 3.2(2). So we need only show that $\omega\text{-}Conv(Res(s)) \precsim_A^{EM} Res(t)$ if and only if $Res(s) \precsim_A^{EM} Res(t)$.
When $Res(s)$ is finite this is trivial as $Res(s) \approx_A^{EM} \omega\text{-}Conv(Res(s))$, as above. Suppose $Res(s)$ is infinite. Then $\bot \in Res(s)$. So $Res(s) \precsim_A^{EM} \omega\text{-}Conv(Res(s))$. Thus we have that $\omega\text{-}Conv(Res(s)) \precsim_A^{EM} Res(t)$ implies $Res(s) \precsim_A^{EM} Res(t)$. It remains to prove the converse. Suppose $Res(s) \precsim_A^{EM} Res(t)$. As $\bot \in Res(s)$, it suffices to show that $\omega\text{-}Conv(Res(s)) \subseteq \{x \in \|A\| \mid \exists y \in Res(t) \text{ s.t. } x \precsim_A y\}$. But this holds because the right-hand set is ω-convex (ω-inductivity follows from Lemma 3.2(1)) and contains $Res(s)$ (as $Res(s) \precsim_A^{EM} Res(t)$).
2. Suppose $\bot \in Res(t)$. Then $\omega\text{-}Conv(Res(s)) \precsim_A^{EM} \omega\text{-}Conv(Res(t))$ if and only if $\bot \in \omega\text{-}Conv(Res(s))$ and $\omega\text{-}Conv(Res(s)) \subseteq \omega\text{-}Conv(Res(t))$. But, by elementary properties of the closure operator $\omega\text{-}Conv(\cdot)$, this holds if and only if $\bot \in Res(s)$ and $Res(s) \subseteq \omega\text{-}Conv(Res(t))$ as required.
3. Follows easily from the above.

Lemma 3.4 *$s \sqsubseteq_{\mathcal{P}(A)} t$ implies $Res(s) \precsim_A^{EM} Res(t)$.*

Proof. Suppose that $s \sqsubseteq_{\mathcal{P}(A)} t$.

For any $x \in Res(s)$ we must find $y \in Res(t)$ such that $x \precsim_A y$. This is trivial if $x = \bot$. Otherwise, $x = s(\sigma)$ for some σ. But $s(\sigma) \sqsubseteq_A t(\sigma) \in Res(t)$. So $t(\sigma)$ is the required y.

Conversely, for any $y \in Res(t)$ we must find $x \in Res(s)$ such that $x \precsim_A y$. If $\bot \in Res(s)$ this is trivial. Otherwise $y \neq \bot$ (as $\bot \notin Res(t)$) so $y = t(\sigma)$ for some σ. But then $\sigma \in Leaves(s)$ (as $\bot \notin Res(s)$) and $s(\sigma) \sqsubseteq_A t(\sigma)$. Thus $s(\sigma)$ is the required x.

Proof of Proposition 3.1. There are three conditions to verify.

1. Suppose $s \sqsubseteq_{\mathcal{P}(A)} t$. We must show that $s \precsim_{\mathcal{P}(A)} t$.
By Lemma 3.4, we have that $Res(s) \precsim_A^{EM} Res(t)$. So, when $Res(t)$ is finite we have $s \precsim_{\mathcal{P}(A)} t$ by Lemma 3.3(1). When $Res(t)$ is infinite we have $\bot \in Res(t)$. So $\bot \in Res(s)$ and $Res(s) \subseteq Conv(Res(t))$. But $Conv(Res(t)) \subseteq \omega\text{-}Conv(Res(t))$. So indeed $s \precsim_{\mathcal{P}(A)} t$ by Lemma 3.3(2).
2. For the ω-inductivity of $\precsim_{\mathcal{P}(A)}$, it suffices to show that, for any ascending chain $s_0 \sqsubseteq_{\mathcal{P}(A)} s_1 \sqsubseteq_{\mathcal{P}(A)} \ldots$, and for any t such that, for all i, we have $s_i \precsim_{\mathcal{P}(A)} t$, it holds that $\bigsqcup_i s_i \precsim_{\mathcal{P}(A)} t$. Suppose then that s_i and t are as above. Define $s_\omega = \bigsqcup_i s_i$. We use Lemma 3.3 to show that $s_\omega \precsim_{\mathcal{P}(A)} t$.
Case 1: t is finite. We show that $Res(s_\omega) \precsim_A^{EM} Res(t)$.
Suppose $x \in Res(s_\omega)$. We must show that there exists $y \in Res(t)$ such that $x \precsim_A y$. If $x = \bot$ then any $y \in Res(t)$ will do. Otherwise $x = s_\omega(\sigma)$ for some $\sigma \in Leaves(s_\omega)$. But then $x = \bigsqcup\{s_i(\sigma) \mid \sigma \in Leaves(s_i)\}$, and for each such

$s_i(\sigma)$ we have that there exists $y_i \in \mathit{Res}(t)$ such that $s_i(\sigma) \precsim_A y_i$ (because $s_i \precsim_A^{EM} t$). So, by Lemma 3.2(1), there indeed exists $y \in \mathit{Res}(t)$ such that $x \precsim_A y$.
Now suppose $y \in \mathit{Res}(t)$. We must show that there exists $x \in \mathit{Res}(s_\omega)$ such that $x \precsim_A y$. This is trivial if $\bot \in \mathit{Res}(s_\omega)$. Otherwise $\mathit{Leaves}(s_\omega)$ is finite. Let n be such that, for all $i \geq n$, it holds that $\mathit{Leaves}(s_i) = \mathit{Leaves}(s_\omega)$. For all $i \geq n$, we have $s_i \precsim_{\mathcal{P}(A)} t$, so there exists $\sigma_i \in \mathit{Leaves}(s_\omega)$ such that $s_i(\sigma_i) \precsim_A y$. As $\mathit{Leaves}(s_\omega)$ is finite, some $\sigma \in \mathit{Leaves}(s_\omega)$ equals σ_i for infinitely many i. But then, for all $i \geq n$, we have $s_i(\sigma) \precsim_A y$. So, as $\precsim_A$ is ω-inductive, $\bigsqcup_{i \geq n} s_i(\sigma)$ is the sought x.
Case 2: t is infinite. We show $\bot \in \mathit{Res}(s_\omega)$ and $\mathit{Res}(s_\omega) \subseteq \omega\text{-}\mathit{Conv}(\mathit{Res}(t))$. By Lemma 3.3(2), we have $\bot \in \mathit{Res}(s_i)$ and $\mathit{Res}(s_i) \subseteq \omega\text{-}\mathit{Conv}(\mathit{Res}(t))$. It follows easily that $\bot \in \mathit{Res}(s_\omega)$. Consider now any $x \neq \bot$ in $\mathit{Res}(s_\omega)$. Then $x = s_\omega(\sigma)$ for some $\sigma \in \mathit{Leaves}(s_\omega)$. Also $x = \bigsqcup\{s_i(\sigma) \mid \sigma \in \mathit{Leaves}(s_i)\}$, and each such $s_i(\sigma)$ is in $\omega\text{-}\mathit{Conv}(\mathit{Res}(t))$. So, as $\omega\text{-}\mathit{Conv}(\mathit{Res}(t))$ is ω-inductive, we have that $x \in \omega\text{-}\mathit{Conv}(\mathit{Res}(t))$ as required.

3. It is easily checked that $t \precsim_{\mathcal{P}(A)} \bot$ implies $t = \bot$.

4 Characterization of the powerdomain

In this section we consider some of the basic operations associated with $\mathcal{P}(A)$. Following [9], two operations are given as primitive and the others are derived using a characterization of $\mathcal{P}(A)$ as the free semilattice in $\mathbf{Q\boldsymbol{\omega}P}_\bot$. This characterization underlines the naturality of the construction given in the previous section.

One basic operation associated with $\mathcal{P}(A)$ is the singleton function from A to $\mathcal{P}(A)$, which maps a to a deterministic computation whose result is a. This is the $\mathbf{Q\boldsymbol{\omega}P}$ morphism $A \xrightarrow{\{\cdot\}} \mathcal{P}(A)$ given by the function mapping $a \in |A|$ to $\{a\}$. It is realized by the continuous function $l : \|A\| \to \|\mathcal{P}(A)\|$ defined by:

$$l(x) \quad = \quad \{(\epsilon, x) \mid x \neq \bot\}.$$

The other primitive operation is the nondeterministic choice operator. This is the $\mathbf{Q\boldsymbol{\omega}P}$ morphism $\mathcal{P}(A) \times \mathcal{P}(A) \xrightarrow{\uplus} \mathcal{P}(A)$ defined by:

$$D \uplus E \quad = \quad \omega\text{-}\mathit{Conv}(D \cup E),$$

where, for $D \subseteq |A|$ we write $\omega\text{-}\mathit{Conv}(D)$ to mean $q_A(\omega\text{-}\mathit{Conv}(q_A^{-1}(D)))$. Actually, one can replace $\omega\text{-}\mathit{Conv}$ with Conv in the above formula, as can be shown by a case analysis on the forms of D and E. This is interesting as $\mathit{Conv}(D \cup E)$ can be calculated using the poset $(|A|, \leq_A)$ alone. A realizer for $\uplus$ is given by the continuous function $\langle \cdot, \cdot \rangle : \|\mathcal{P}(A)\| \times \|\mathcal{P}(A)\| \to \|\mathcal{P}(A)\|$ defined by:

$$\langle s, t \rangle \quad = \quad \{(0\sigma, x) \mid (\sigma, x) \in s\} \cup \{(1\tau, x) \mid (\tau, x) \in t\}.$$

Note that both $\{\cdot\}$ and $\uplus$ are strict morphisms.

The other operations of interest will be obtained via a universal property characterizing $\mathcal{P}(A)$ as a free algebra. We now consider the general form of such algebras. A *realized semilattice* (henceforth just semilattice) in $\mathbf{Q\omega P}$ is given by an object B together with a morphism $B \times B \xrightarrow{\vee} B$ such that, for all $a, b, c \in |B|$ we have that:

1. $a \vee a = a$,
2. $a \vee b = b \vee a$, and
3. $a \vee (b \vee c) = (a \vee b) \vee c$.

Note that equation 1 implies that $\vee$ is strict. A *linear morphism* from one semilattice $(B, \vee)$ to another $(B', \vee')$ is a $\mathbf{Q\omega P}$ morphism ϕ from B to B' such that, for all $b, b' \in |B|$, it holds that $\phi(b \vee b') = \phi(b) \vee' \phi(b')$.

Any object $\mathcal{P}(A)$ forms a semilattice with the required map given by $\uplus$ (the equalities are easily checked). Indeed, the next theorem characterizes $\mathcal{P}(A)$ as the free semilattice in the category $\mathbf{Q\omega P}_\perp$ of strict maps.

Theorem 4.1 *Let $(B, \vee)$ be any semilattice. For any strict morphism $A \xrightarrow{\phi} B$, there is a unique strict linear morphism $(\mathcal{P}(A), \uplus) \xrightarrow{\phi^\dagger} (B, \vee)$ such that the diagram below commutes.*

$$\begin{tikzcd}
\mathcal{P}(A) \arrow[r, "\phi^\dagger"] & B \\
A \arrow[u, "\{\cdot\}"] \arrow[ur, dotted, "\phi"'] &
\end{tikzcd}$$

Moreover, there is a strict morphism $(A \Rightarrow_\perp B) \xrightarrow{(\cdot)^\dagger} (\mathcal{P}(A) \Rightarrow_\perp B)$ mapping any ϕ to its associated $\phi^\dagger$.

It is interesting to note that the condition of strictness cannot be dropped from the theorem, as there are examples of non-strict ϕ for which there exist no linear $\phi^\dagger$ (strict or non-strict) making the diagram commute. Thus, unlike the convex powerdomain in $\mathbf{Cpo}$ [9, 14], $\mathcal{P}(A)$ is not the free semilattice in $\mathbf{Q\omega P}$. This situation arises because the definition of $\mathcal{P}(A)$ treats the least element of A as a distinguished value representing nontermination, whereas the non-strict maps of $\mathbf{Q\omega P}$ treat it like any other value. It appears that no free semilattice exists in $\mathbf{Q\omega P}$.

The construction of $\phi^\dagger$ from ϕ is via an operation on realizers. Given $(B, \vee)$, let $\sqcup$ be a chosen realizer for $\vee$. Let f be any morphism in $\boldsymbol{\omega}\mathbf{P}_\perp(A, B)$. Define $f^\dagger$ to be the least solution in $\mathbf{Cpo}_\perp(\|\mathcal{P}(A)\|, \|B\|)$ of:

$$f^\dagger(t) \;=\; \begin{cases} f(x) & \text{if } t = l(x), \\ f^\dagger(s_1) \sqcup f^\dagger(s_2) & \text{if } t = \langle s_1, s_2 \rangle. \end{cases}$$

It is easy to check that $f^\dagger$ is well-defined, but note that the strictness of f is needed for the equation to be consistent when $\mathit{Leaves}(t) = \emptyset$.

Proposition 4.2 *For any $f, g \in \boldsymbol{\omega P}_\perp(A,B)$ and $s,t \in \|\mathcal{P}(A)\|$, if $f \precsim_{A \Rightarrow_\perp B} g$ and $s \precsim_{\mathcal{P}(A)} t$ then $f^\dagger(s) \precsim_B g^\dagger(t)$.*

We delay the proof until the end of the section. An immediate consequence of the proposition is that if f realizes ϕ then $f^\dagger$ realizes a $\mathbf{Q\boldsymbol{\omega}P}_\perp$ morphism from $\mathcal{P}(A)$ to B. Define $\phi^\dagger$ to be this morphism.

Proof of Theorem 4.1. By its definition $\phi^\dagger$ is strict. By the definition of $f^\dagger$ we have that $f^\dagger(\langle s,t\rangle) = f^\dagger(s) \sqcup f^\dagger(t)$ and $f^\dagger(l(x)) = f(x)$. Thus $\phi^\dagger(D \uplus E) = \phi^\dagger(D) \vee \phi^\dagger(E)$ and $\phi^\dagger(\{a\}) = \phi(a)$. So $\phi^\dagger$ is linear and the diagram commutes.

For uniqueness suppose we have a strict, linear ψ making the diagram commute. Let h be a realizer for ψ. The linearity of ψ gives us that:

$$h(\langle s,t\rangle) \quad \approx_B \quad h(s) \sqcup h(t),$$

and the commutativity of the diagram gives us:

$$h(l(x)) \quad \approx_B \quad f(x).$$

We must show that $h(t) \approx_B f^\dagger(t)$ for all t.

This is proved first for finitely generated t. An important observation is that all the finitely generated computation trees are generated from trees of the form $l(a)$ by a finite number of applications of $\langle\cdot,\cdot\rangle$. Using this fact, it is easy to show that $h(t) \approx_B f^\dagger(t)$ for finitely generated t, by induction on the structure of t using the two equivalences above.

For arbitrary t we have, as in Section 3, that $t = \bigsqcup_i t\lceil_i$ where each $t\lceil_i$ is finitely generated. But then

$$h(t) \quad = \quad \bigsqcup_i h(t\lceil_i) \quad \approx_B \quad \bigsqcup_i f^\dagger(t\lceil_i) \quad = \quad f^\dagger(t),$$

because h and $f^\dagger$ are continuous and $\approx_B$ is ω-inductive.

It remains to show that $(A \Rightarrow_\perp B) \xrightarrow{(\cdot)^\dagger} (\mathcal{P}(A) \Rightarrow_\perp B)$ is a morphism in $\mathbf{Q\boldsymbol{\omega}P}_\perp$. This follows from Proposition 4.2. ⊠

Theorem 4.1 implies that $\mathcal{P}(.)$ is the action on objects of the functor part of a strong monad on $\mathbf{Q\boldsymbol{\omega}P}_\perp$. We now derive the associated operations, giving explicit descriptions of the action of their underlying functions and associated realizers.

The multiplication of the monad, $\biguplus : \mathcal{P}(\mathcal{P}(A)) \longrightarrow \mathcal{P}(A)$, is obtained as $(1_{\mathcal{P}(A)})^\dagger$. Concretely, it is the expected function,

$$\biguplus(\boldsymbol{D}) \quad = \quad \omega\text{-}Conv(\bigcup \boldsymbol{D}),$$

which is realized by the function mapping $\boldsymbol{s} \in \|\mathcal{P}(\mathcal{P}(A))\|$ to:

$$\{(\sigma\tau,\, x) \mid (\sigma, t) \in \boldsymbol{s} \text{ and } (\tau, x) \in t\}.$$

Incidentally, as with $\uplus$ above, one can replace ω-*Conv* with *Conv* in the formula defining $\biguplus(\boldsymbol{D})$.

The operation of the $\mathcal{P}(\cdot)$ functor on strict morphisms works by mapping $A \xrightarrow{\phi} B$ to $(\{\cdot\} \circ \phi)^\dagger$. Concretely, this is the strict morphism $\mathcal{P}(A) \xrightarrow{\mathcal{P}(\phi)} \mathcal{P}(B)$ defined by:

$$\mathcal{P}(\phi)(D) \quad = \quad \omega\text{-}Conv(\phi(D)).$$

If f realizes ϕ then $\mathcal{P}(\phi)$ is realized by the function mapping $s \in \|\mathcal{P}(A)\|$ to:

$$\{(\sigma, f(x)) \mid (\sigma, x) \in s \text{ and } f(x) \neq \bot\}.$$

In contrast to the previous operations, it is not in general possible to replace ω-*Conv* with *Conv* in the definition of $\mathcal{P}(\phi)(D)$.

The strength of the monad, $A \otimes \mathcal{P}(B) \xrightarrow{st} \mathcal{P}(A \otimes B)$, is obtained from $(A \Rightarrow_\bot B) \xrightarrow{(\cdot)^\dagger} (\mathcal{P}(A) \Rightarrow_\bot B)$ in the usual way (see Kock [12]). Concretely it is the unique strict map satisfying:

$$st(\langle a, D\rangle) \quad = \quad \{\langle a, b\rangle \mid b \in D\backslash\{\bot\}\} \,\cup\, \{\bot \mid \bot \in D\}$$

(note that this set is automatically ω-convex). The strength is realized by the unique strict function mapping $\langle x, t\rangle \in \|A \otimes \mathcal{P}(B)\|$ to

$$\{(\sigma, \langle x, y\rangle) \mid (\sigma, y) \in t\}.$$

The remainder of the section is devoted to the promised proof of Proposition 4.2. Throughout the proof we use f, g to range over elements of $\boldsymbol{\omega}\mathbf{P}_\bot(A, B)$ and s, t to range over elements of $\|\mathcal{P}(A)\|$.

Lemma 4.3 *For finitely generated s, t, if $Res(s) = Res(t)$ then $f^\dagger(s) \approx_B f^\dagger(t)$.*

Proof. This follows easily from the semilattice axioms, using the fact that any finitely generated tree is obtained from trees of the form $l(a)$ by a finite number of applications of $\langle\cdot,\cdot\rangle$.

Lemma 4.4 *If $x \in \omega\text{-}Conv(Res(t))$ then $f^\dagger(t) \approx_B f^\dagger(\langle t, l(x)\rangle)$.*

Proof. We show that $X = \{x \in \|A\| \mid f^\dagger(t) \approx_B f^\dagger(\langle t, l(x)\rangle)\}$ is ω-convex and contains $Res(t)$.

Suppose $x \in Res(t)$. Let n be such that $x \in Res(t\lceil_n)$. Then, for all $i \geq n$, $Res(t\lceil_i) = Res(\langle t, l(x)\rangle\lceil_{i+1})$. So

$$f^\dagger(t) \quad = \quad \textstyle\bigsqcup_{i \geq n} f^\dagger(t\lceil_i) \quad \approx_B \quad \bigsqcup_{i \geq n} f^\dagger(\langle t, l(x)\rangle\lceil_{i+1}) \quad = \quad f^\dagger(\langle t, l(x)\rangle),$$

because $f^\dagger$ is continuous, $\approx_B$ is ω-inductive and $f^\dagger(t\lceil_i) \approx_B f^\dagger(\langle t, l(x)\rangle\lceil_{i+1})$ by Lemma 4.3. Therefore $Res(t) \subseteq X$.

For convexity, suppose we have $y, z \in X$ with $y \precsim_A x \precsim_A z$. Then

$$f^\dagger(t) \quad \approx_B \quad f^\dagger(\langle t, l(y)\rangle) \quad = \quad f^\dagger(t) \sqcup f(y) \quad \precsim_B \quad f^\dagger(t) \sqcup f(x) \quad = \quad f^\dagger(\langle t, l(x)\rangle)$$

because $y \in X$, and f and $\sqcup$ respect the $\precsim$ preorders. A similar argument using z shows that $f^\dagger(\langle t, l(x)\rangle) \precsim_B f^\dagger(t)$. Thus indeed $x \in X$.

For ω-inductivity, suppose we have $x_0 \sqsubseteq_A x_1 \sqsubseteq_A \ldots$ in X. Then indeed

$$f^\dagger(t) \quad \approx_B \quad \bigsqcup_i f^\dagger(\langle t, l(x_i)\rangle) \quad = \quad f^\dagger(\langle t, l(\bigsqcup_i x_i)\rangle)$$

because $\approx_B$ is ω-inductive and all the functions are continuous.
Proof of Proposition 4.2. We must show that if $f \precsim_{A \Rightarrow_\perp B} g$ and $s \precsim_{\mathcal{P}(A)} t$ then $f^\dagger(s) \precsim_B f^\dagger(t)$.

Case 1: s and t finitely generated.
Suppose $s \precsim_{\mathcal{P}(A)} t$. Then $Res(s) \precsim_A^{EM} Res(t)$, by Lemma 3.3(1). So we can write $Res(s)$ as $\{x_1, \ldots, x_k\}$ and $Res(t)$ as $\{y_1, \ldots, y_k\}$ where $x_i \precsim_A y_i$ for each i $(1 \leq i \leq k)$. So, by Lemma 4.3, we have:

$$\begin{array}{lcl} f^\dagger(s) & \approx_B & f^\dagger(\langle l(x_1), \ldots, l(x_k)\rangle) \quad = \quad f(x_1) \sqcup \ldots \sqcup f(x_k), \text{ and} \\ g^\dagger(t) & \approx_B & g^\dagger(\langle l(y_1), \ldots, l(y_k)\rangle) \quad = \quad g(y_1) \sqcup \ldots \sqcup g(y_k). \end{array}$$

But also:

$$f(x_1) \sqcup \ldots \sqcup f(x_k) \quad \precsim_B \quad g(y_1) \sqcup \ldots \sqcup g(y_k)$$

because $\sqcup$ respects the preorder and $f \precsim_{A \Rightarrow_\perp B} g$. Thus indeed $f^\dagger(s) \precsim_B g^\dagger(t)$.

Case 2: s finitely generated and t infinite.
Suppose $s \precsim_{\mathcal{P}(A)} t$. Then $\perp \in Res(s)$ and $Res(s) \subseteq \omega\text{-}Conv(Res(t))$, by Lemma 3.3(2). Suppose $Res(s) = \{x_1, \ldots, x_k\}$. Define $t' = \langle t, l(x_1), \ldots, l(x_k)\rangle$. Clearly $Res(s) \subseteq Res(t')$. So for some n we have that $Res(s) \subseteq Res(t'\lceil_i)$ for all $i \geq n$. As $\perp \in Res(s)$ we have, by Lemma 3.3(2), that $s \precsim_{\mathcal{P}(A)} t'\lceil_i$ for such i. Hence, by Case 1 above, $f^\dagger(s) \precsim_B g^\dagger(t'\lceil_i)$. But $t' = \bigsqcup_{i \geq n} t'\lceil_i$. So, by the ω-inductivity of $\precsim_B$ and continuity of $g^\dagger$, we have $f^\dagger(s) \precsim_B g^\dagger(t')$. However, by the definition of t', it follows from k iterates of Lemma 4.4 that $g^\dagger(t) \approx_B g^\dagger(t')$. Thus indeed $f^\dagger(s) \precsim_B g^\dagger(t)$.

Case 3: s and t arbitrary.
Suppose $s \precsim_{\mathcal{P}(A)} t$. Then, for any i, we have $s\lceil_i \precsim_{\mathcal{P}(A)} t$. So, by the appropriate case above, $f^\dagger(s\lceil_i) \precsim_B g^\dagger(t)$. But $s = \bigsqcup_i s\lceil_i$. So by the ω-inductivity of $\precsim_B$ and continuity of $f^\dagger$, we have $f^\dagger(s) \precsim_B g^\dagger(t)$ as required.

5 Properties of the powerdomain

In this section we consider some additional properties of our powerdomain, which expose its differences from the convex powerdomain on **Cpo**.

First we show that the powerdomain does not preserve the objects in the image of $I : \mathbf{Cpo} \to \boldsymbol{\omega}\mathbf{P}$. In other words, the powerdomain takes one outside the world of cpos. A simple example is given by $\mathcal{P}(\mathbb{N}_\perp \Rightarrow_\perp \mathbb{N}_\perp)$. For $k \geq 0$ define $I_k \subseteq |\mathbb{N}_\perp \Rightarrow_\perp \mathbb{N}_\perp|$ by:

$$I_k \; = \; \{f \in |\mathbb{N}_\perp \Rightarrow_\perp \mathbb{N}_\perp| \;\mid\; f(n) \in \{0, 1\} \text{ if } 0 \leq n < k \text{ and } f(n) = \perp \text{ otherwise}\}.$$

Then, for all k, we have $I_k \in |\mathcal{P}(\mathbb{N}_\perp \Rightarrow_\perp \mathbb{N}_\perp)|$, as is easily checked. Further, as $k \leq l$ implies $I_k \leq^{EM}_{\mathbb{N}_\perp \Rightarrow_\perp \mathbb{N}_\perp} I_l$, we have that $I_0 \leq_{\mathcal{P}(\mathbb{N}_\perp \Rightarrow_\perp \mathbb{N}_\perp)} I_1 \leq_{\mathcal{P}(\mathbb{N}_\perp \Rightarrow_\perp \mathbb{N}_\perp)} \cdots$ is an ascending chain in $|\mathcal{P}(\mathbb{N}_\perp \Rightarrow_\perp \mathbb{N}_\perp)|$. We claim that this chain has no lub (indeed no upper bound). Thus the underlying poset of $\mathcal{P}(\mathbb{N}_\perp \Rightarrow_\perp \mathbb{N}_\perp)$ is not a cpo, and hence $\mathcal{P}(\mathbb{N}_\perp \Rightarrow_\perp \mathbb{N}_\perp)$ is not isomorphic to any object in the image of $I : \mathbf{Cpo} \to \boldsymbol{\omega}\mathbf{P}$. This shows that, as stated earlier, $\mathbf{Q}\boldsymbol{\omega}\mathbf{P}$ and $\mathbf{Q}\boldsymbol{\omega}\mathbf{P}_\perp$ are indeed not **Cpo**-enriched under the pointwise order on hom-sets.

To justify the claim, suppose that $D \in |\mathcal{P}(\mathbb{N}_\perp \Rightarrow_\perp \mathbb{N}_\perp)|$ is an upper bound of the chain. Thus, for each k, it holds that $I_k \leq^{EM}_{\mathbb{N}_\perp \Rightarrow_\perp \mathbb{N}_\perp} D$. We show that these inequalities imply that D is an infinite subset of

$$I_\omega \quad = \quad \{f \in |\mathbb{N}_\perp \Rightarrow_\perp \mathbb{N}_\perp| \mid f(n) \in \{0,1\} \text{ for all } n\}.$$

First we show that $D \subseteq I_\omega$. Take any $f \in D$. Then, for each n, there exists $g \in I_{n+1}$ such that $g \leq_{\mathbb{N}_\perp \Rightarrow_\perp \mathbb{N}_\perp} f$. But then $g(n) \in \{0,1\}$ and so $f(n) \in \{0,1\}$ as required. Now suppose that D were finite. Take any k such that $|D| < 2^k$. There must be some $g \in I_k$ for which there is no $f \in D$ with $g \leq_{\mathbb{N}_\perp \Rightarrow_\perp \mathbb{N}_\perp} f$. But this contradicts D being an upper bound of the chain.

It only remains to show that $|\mathcal{P}(\mathbb{N}_\perp \Rightarrow_\perp \mathbb{N}_\perp)|$ contains no infinite subsets of I_ω. As I_ω does not contain $\perp_{\mathbb{N}_\perp \Rightarrow_\perp \mathbb{N}_\perp}$ we have that any subset $E \subseteq I_\omega$ for which $E \in |\mathcal{P}(\mathbb{N}_\perp \Rightarrow_\perp \mathbb{N}_\perp)|$ must be of the form $\omega\text{-}Conv(E')$ for some finite $E' \subseteq E$. But it is easily checked that any finite subset of I_ω is already ω-convex. Therefore E must be finite. Thus indeed the I_i chain has no upper bound.

It is interesting to note that the above example has direct computational relevance. The elements of $\mathcal{P}(\mathbb{N}_\perp \Rightarrow_\perp \mathbb{N}_\perp)$ should be viewed as denotations of nondeterministic programs which, if they terminate, output a deterministic program giving a (somewhere defined) partial function on the natural numbers. By König's lemma, any such program that necessarily terminates may only output a finite number of such partial functions. Each of the sets I_k corresponds to an intuitively computable, necessarily terminating nondeterministic program. As argued above, any upper bound of the chain would be an infinite subset of I_ω. This could not be the denotation of a program because it would correspond to a necessarily terminating computation (as $\perp_{\mathbb{N}_\perp \Rightarrow_\perp \mathbb{N}_\perp}$ is not in the set) with an infinite number of possible outputs. Thus it is computationally reasonable that no upper bound exists.

Of course, one would expect $\mathcal{P}(\mathbb{N}_\perp \Rightarrow_\perp \mathbb{N}_\perp)$ to contain many elements that cannot be the denotation of any program, so one may question the relevance of whether or not an upper bound to the I_i exists. The relevance is that, ideally, one would like a notion of "computable element", singling out those elements that a program could possibly denote. In **Cpo** such questions of computability are handled using effective ω-algebraic cpos, and the standard convex powerdomain operates upon such cpos. One defines a "computable element" of a cpo to be the lub of a recursive chain of compact elements. However, I_i defines a recursive ascending chain of compact elements in the appropriate cpo. So in the conventional setting this chain has a lub which is deemed to be "computable". Our avoidance of such elements suggests that our powerdomain might provide a more appropriate setting for defining a notion of "computability" for denotations of nondeterministic computation.

The König's lemma argument that a nondeterministic program should be represented by a set of results that if infinite necessarily contains $\perp$, apparently relies upon the assumption that a nonterminating program has no observable behaviour. This assumption is not always valid. For example, suppose we have a class of nonterminating nondeterministic programs that can output natural numbers during execution. One might be tempted to model such programs as elements of $\mathcal{P}(\mathit{Streams})$ where *Streams* is (inherited from) the initial solution (in $\mathbf{Cpo}_\perp$) of the recursive domain equation:

$$\mathit{Streams} \;\cong\; \mathbb{N}_\perp \otimes L(\mathit{Streams}).$$

Observe that $\mathcal{P}(\mathit{Streams})$ contains an evident analogue of the I_i chain, but this time one would like a lub to exist, for this should represent the computation that repeatedly chooses between outputting a 0 and outputting a 1. As before, this lub is available if one uses the standard convex powerdomain on cpos, but not with our powerdomain. Thus it might seem that our powerdomain cannot be used to model standard features such as mid-execution output. However, this is not the case. Rather than taking $\mathcal{P}(\mathit{Streams})$ as the domain of denotations, one should instead take the initial solution in $\mathbf{Q}\boldsymbol{\omega}\mathbf{P}_\perp$ of the recursive domain equation:

$$A \;\cong\; \mathcal{P}(\mathbb{N}_\perp \otimes LA)$$

(we will show how to solve such domain equations in Section 6). This equation amounts to modelling *bisimilarity* between programs, whereas $\mathcal{P}(\mathit{Streams})$ attempted to model *trace equivalence*. It should also be possible to model trace equivalence by moving to the category of (realized) semilattices and linear maps and solving a recursive domain equation involving the tensor product that classifies bilinear maps, as in [9].

The above differences between our powerdomain and the classical one, involved the existence and nonexistence of infinite behaviours. Next we point out an entirely finitary difference, adapting an example due to Sieber [16]. We show that there is no $\mathbf{Q}\boldsymbol{\omega}\mathbf{P}$ morphism $(\mathbf{1}_\perp \times \mathbf{1}_\perp) \xrightarrow{\phi} \mathcal{P}(\mathbf{1}_\perp \times \mathbf{1}_\perp \times \mathbf{1}_\perp \times \mathbf{1}_\perp)$ representing the function:

$$\begin{array}{lcl}
\phi(\perp,\perp) & = & \{\langle *,\perp,\perp,\perp\rangle,\ \langle \perp,\perp,\perp,*\rangle\},\\
\phi(\perp,*) & = & \{\langle *,\perp,*,\perp\rangle,\ \langle \perp,*,\perp,*\rangle\},\\
\phi(*,\perp) & = & \{\langle *,*,\perp,\perp\rangle,\ \langle \perp,\perp,*,*\rangle\},\\
\phi(*,*) & = & \{\langle *,*,\perp,*\rangle,\ \langle *,\perp,*,*\rangle\}.
\end{array}$$

Suppose, for contradiction, that f realizes ϕ. Then there must be some $\sigma \in \mathit{Leaves}(f(\perp,\perp))$ such that $f(\perp,\perp)(\sigma) = \langle *,\perp,\perp,\perp\rangle$. So, by the monotonicity of f, it must hold that $f(\perp,*)(\sigma) = \langle *,\perp,*,\perp\rangle$ and hence $f(*,*)(\sigma) = \langle *,\perp,*,*\rangle$. Similarly, $f(*,\perp)(\sigma) = \langle *,*,\perp,\perp\rangle$ and hence $f(*,*)(\sigma) = \langle *,*,\perp,*\rangle$. We now have two contradictory values for $f(*,*)$, so indeed ϕ has no realizer. On the other hand, as ϕ is monotonic, it does give a morphism if the convex powerdomain in $\mathbf{Cpo}$ is used.

Above, we have considered examples using domains such as $\mathcal{P}(\mathbb{N}_\perp \Rightarrow_\perp \mathbb{N}_\perp)$ and $\mathcal{P}(\mathbf{1}_\perp \times \mathbf{1}_\perp \times \mathbf{1}_\perp \times \mathbf{1}_\perp)$, which do not correspond to natural domains for interpreting nondeterministic computation. We chose these domains in order to illustrate the various phenomena in simple settings. Similar phenomena do also arise in more natural domains for nondeterministic programs, for example, in the domains used to interpret Sieber's nondeterministic, call-by-value version of PCF [16]. To interpret this language in $\mathbf{Q}\boldsymbol{\omega}\mathbf{P}_\perp$, one would interpret types by:

$$\begin{aligned} [\![\iota]\!] &= \mathbb{N}_\perp, \\ [\![\sigma \to \tau]\!] &= L\,([\![\sigma]\!] \Rightarrow_\perp [\![\mathcal{P}(\tau)]\!]), \end{aligned}$$

and the denotation of a program of type σ would be an element of $\mathcal{P}([\![\sigma]\!])$. It is straightforward to interpret all of Sieber's language including his **exists** operator. Sieber interpreted his language using the standard convex powerdomain on cpos. He showed that full abstraction fails because there is a non-definable finite element in $[\![(\iota \to \iota) \to (\iota \to \iota)]\!]$. This non-definable element is a variant of the function ϕ defined above, and, as with ϕ above, it does not exist if our powerdomain is used. Thus Sieber's counterexample to full abstraction fails when the language is interpreted in $\mathbf{Q}\boldsymbol{\omega}\mathbf{P}_\perp$. We do not know if the interpretation in $\mathbf{Q}\boldsymbol{\omega}\mathbf{P}_\perp$ gives rise to a fully abstract model. It would be interesting to investigate this question, but the analysis might well involve developing a theory of "algebraic" or even "bifinite" realized posets. It would also be interesting to investigate the possibility of obtaining a universality theorem. As with the earlier discussion for $\mathcal{P}(\mathbb{N}_\perp \Rightarrow_\perp \mathbb{N}_\perp)$, it seems that our powerdomain offers the possibility of a better account of nondeterministic computability in the interpretations of the PCF types. But such an investigation would probably require a theory of "effective algebraic" realized posets. We leave such developments for future work.

6 Recursive domain equations

In this section we show that $\mathbf{Q}\boldsymbol{\omega}\mathbf{P}_\perp$ is algebraically compact for a good class of endofunctors, allowing us to find canonical solutions to recursive domain equations.

First we make an observation concerning the construction of solutions to recursive domain equations in $\mathbf{Q}\boldsymbol{\omega}\mathbf{P}_\perp$. Consider the ω-chain of morphisms:

$$\begin{tikzcd} \mathbf{0}_\perp \arrow[r, "\perp"] & L\mathbf{0}_\perp \arrow[r, "L\perp"] & L^2\mathbf{0}_\perp \arrow[r, "L^2\perp"] & \dots \end{tikzcd}$$

The colimit of this diagram in $\mathbf{Q}\boldsymbol{\omega}\mathbf{P}_\perp$ is the object A defined (up to isomorphism) by:

$$\begin{aligned} \|A\| &\;=\; \{\langle m,n\rangle \mid m,n \geq 1,\ m \leq n\} \cup \{\perp\}, \\ p \sqsubseteq_A q &\quad\text{if}\quad \text{either } p = \perp \text{ or: } p = \langle m,n\rangle,\ q = \langle m,n'\rangle \text{ and } n \leq n', \\ p \precsim_A q &\quad\text{if}\quad \text{either } p = \perp \text{ or: } p = \langle m,n\rangle,\ q = \langle m',n'\rangle \text{ and } n \leq n'. \end{aligned}$$

It is easily checked that A is not isomorphic to LA. Thus one cannot solve a recursive domain equation in $\mathbf{Q\omega P}_\perp$ by taking the colimit of the usual ω-chain. Note also that the underlying poset of A is not a cpo. This gives another example of how $\mathbf{Q\omega P}$ and $\mathbf{Q\omega P}_\perp$ are not **Cpo**-enriched under the pointwise ordering.

Now, consider the ω^{op}-chain:

$$\begin{tikzcd} \mathbf{0}_\perp & L\mathbf{0}_\perp \arrow[l, "!"'] & L^2\mathbf{0}_\perp \arrow[l, "L!"'] & \ldots \arrow[l, "L^2!"'] \end{tikzcd}$$

The limit of this diagram in $\mathbf{Q\omega P}_\perp$ is the object B defined by:

$$\begin{array}{rcl} \|B\| & = & \omega+1, \\ \alpha \sqsubseteq_B \beta & \text{if} & \alpha \leq \beta, \\ \alpha \precsim_B \beta & \text{if} & \alpha \leq \beta, \end{array}$$

which is indeed the expected solution to $B \cong LB$. Thus the colimit of the chain obtained by iterating the functor L is not isomorphic to the limit of its associated co-chain.

Recall that in any **Pposet**-category an *embedding* is a morphism $A \xrightarrow{f} B$ for which there exists a morphism $B \xrightarrow{g} A$ such that $g \circ f = 1_A$ and $f \circ g \leq 1_B$. Any g that arises from an embedding in this way is called a *projection*. An embedding determines its associated projection and vice-versa. In the above example, one sees that the original ω-chain is a chain of embeddings, and the ω^{op}-chain consists of the associated projections. Thus in $\mathbf{Q\omega P}_\perp$ we do not have the limit-colimit coincidence of classical domain theory [18].

The above observations show that one cannot expect to construct solutions to recursive domain equations as ω-colimits in $\mathbf{Q\omega P}_\perp$, but leave open the possibility that solutions to recursive domain equations could be constructed as ω^{op}-limits. However, although we lack a counterexample, this does not seem to be the correct way to proceed. Instead we make use of the fact that all the functors of interest on $\mathbf{Q\omega P}_\perp$ are induced by **Cpo**-functors on $\boldsymbol{\omega}\mathbf{P}_\perp$. As $\boldsymbol{\omega}\mathbf{P}_\perp$ is a **Cpo**-category, recursive domain equations can be solved there in the usual way. These solutions are transported by the functor Q to $\mathbf{Q\omega P}_\perp$. Although the limiting and colimiting properties of the solution are not preserved by Q, other relevant universal properties are.

The relevant universal properties concern initial algebras and terminal coalgebras. We review the basic facts about these. Given any endofunctor F on a category, an *F-algebra* is any morphism $FA \xrightarrow{a} A$. An *F-algebra homomorphism* from $FA \xrightarrow{a} A$ to $FB \xrightarrow{b} B$ is any morphism $A \xrightarrow{x} B$ such that $x \circ a = b \circ F(x)$. An *initial F-algebra* is an initial object in the category of F-algebras and homomorphisms. A famous lemma of Lambek shows that when $FA \xrightarrow{a} A$ is an initial F-algebra then a is an isomorphism (see e.g. [18]).

An important fact about initial algebras is that they can be found functorially in the following sense. Suppose that $G : \mathcal{D} \times \mathcal{C} \to \mathcal{C}$ is a functor such that, for any object A of $\mathcal{D}$, the endofunctor $G(A, \perp)$ has an initial algebra $G(A, C_A) \xrightarrow{a_A} C_A$ in $\mathcal{C}$. Then there exists a canonical initial-algebra-finding functor $G^\dagger : \mathcal{D} \to \mathcal{C}$

that maps each A to C_A. On morphisms $A \xrightarrow{f} B$ in $\mathcal{D}$, one defines $G^{\dagger}(f)$ to be the unique morphism in $\mathcal{C}$ (given by the initiality of a_A) that makes the diagram below commute.

$$\begin{tikzcd}
G(A, C_A) \arrow[rr, "a_A"] \arrow[d, "G(A, G^{\dagger}(f))"'] & & C_A \arrow[d, "G^{\dagger}(f)"] \\
G(A, C_B) \arrow[r, "G(f, C_B)"] & G(B, C_B) \arrow[r, "a_B"] & C_B
\end{tikzcd} \qquad (1)$$

The functoriality of $G^{\dagger}$ follows from the uniqueness of the morphism.

An F*-coalgebra* is just an F^{op}-algebra in $\mathcal{C}^{op}$. The above facts about initial F-algebras are easily dualized to *terminal* F*-coalgebras*. Following Freyd [7, 8], we shall be interested in structures that are simultaneously initial algebras and terminal coalgenras. A *free* F*-algebra* is an isomorphism $FA \xrightarrow{a} A$ such that a is an initial F-algebra and $a^{\perp 1}$ is a terminal F-coalgebra. $\mathcal{C}$ is said to be *algebraically compact* if every endofunctor has a free algebra, where "every" usually ranges over an understood class of functors, for example over suitably enriched functors [5]. Algebraically compact categories allow the construction of canonical solutions to recursive domain equations involving bifunctorial type constructors [7, 5].

Our goal in this section is to show that $\mathbf{Q\boldsymbol{\omega}P}_{\perp}$ is algebraically compact relative to a good class of endofunctors including the powerdomain as well as all functors derived from the type constructors considered in Section 2. More generally one wants to solve recursive domain equations for multi-arity mixed-variance functors, which means having to work with functors from $(\mathbf{Q\boldsymbol{\omega}P}_{\perp} \times \mathbf{Q\boldsymbol{\omega}P}_{\perp}^{op})^k$ to $\mathbf{Q\boldsymbol{\omega}P}_{\perp}$. In order to obtain a setting in which one can treat such generalizations of simple endofunctors uniformly, we consider a class of "realized" categories, which includes all categories $(\mathbf{Q\boldsymbol{\omega}P}_{\perp} \times \mathbf{Q\boldsymbol{\omega}P}_{\perp}^{op})^k$ as well as $\mathbf{Q\boldsymbol{\omega}P}_{\perp}$ itself, and an associated class of "realized" functors between them, which includes the bifunctors of interest. It will turn out that all realized categories are algebraically compact relative to the class of realized endofunctors.

Definition 6.1 (Realized category) *A realized category is a* ***Pposet****-category, $R\mathcal{C}$, together with a* ***Cpo****-category $\mathcal{C}$ and a functor $R : \mathcal{C} \to R\mathcal{C}$ satisfying:*

1. *$\mathcal{C}$ and $R\mathcal{C}$ have the same objects and R is the identity on objects.*
2. *For all objects A, B the functions $R_{AB} : \mathcal{C}(A, B) \to R\mathcal{C}(A, B)$ are admissible quotients.*
3. *Composition in $\mathcal{C}$ is a bistrict morphism in* ***Cpo****.*
4. *$\mathcal{C}$ has a terminal object and limits of all ω^{op}-chains of projections.*

If $R\mathcal{C}$ and $S\mathcal{D}$ are realized categories then so is $R\mathcal{C} \times S\mathcal{D}$ in the evident way. It is more interesting that $(R\mathcal{C})^{op}$ is also a realized category. Here the non-obvious point is that $\mathcal{C}^{op}$ satisfies condition 4. $\mathcal{C}^{op}$ has a terminal object because the bistrictness of composition means that the terminal object in $\mathcal{C}$ is in fact a zero object. Also, the limit of an ω^{op}-chain of projections in $\mathcal{C}^{op}$ corresponds to the colimit of an ω-chain of embeddings in $\mathcal{C}$. This colimit is given as the limit of the induced ω^{op}-chain of projections in $\mathcal{C}$, by the limit-colimit coincidence for **Cpo**-categories [18]. Lastly, $\mathbf{Q}\boldsymbol{\omega}\mathbf{P}_\perp$ is a realized category, its associated **Cpo**-category is $\boldsymbol{\omega}\mathbf{P}_\perp$ and its associated functor is $Q : \boldsymbol{\omega}\mathbf{P}_\perp \to \mathbf{Q}\boldsymbol{\omega}\mathbf{P}_\perp$. Condition 4 holds because the forgetful from $\boldsymbol{\omega}\mathbf{P}_\perp$ to **Cpo** creates limits.

Definition 6.2 (Realized functor) *A realized functor from $R\mathcal{C}$ to $S\mathcal{D}$ is a* ***Pposet****-functor $\Phi : R\mathcal{C} \to S\mathcal{D}$ for which there exists a* ***Cpo****-functor $F : \mathcal{C} \to \mathcal{D}$ making the diagram below commute.*

$$\begin{tikzcd}
\mathcal{C} \arrow[r, "F"] \arrow[d, "R"'] & \mathcal{D} \arrow[d, "S"] \\
R\mathcal{C} \arrow[r, "\Phi"] & R\mathcal{D}
\end{tikzcd}$$

All the type constructors we have on $\mathbf{Q}\boldsymbol{\omega}\mathbf{P}_\perp$ are indeed realized functors between appropriate realized categories. Specifically: $\times$, $+$ and $\otimes$ are realized functors from $\mathbf{Q}\boldsymbol{\omega}\mathbf{P}_\perp \times \mathbf{Q}\boldsymbol{\omega}\mathbf{P}_\perp$ to $\mathbf{Q}\boldsymbol{\omega}\mathbf{P}_\perp$; L and $\mathcal{P}(\cdot)$ are realized functors from $\mathbf{Q}\boldsymbol{\omega}\mathbf{P}_\perp$ to $\mathbf{Q}\boldsymbol{\omega}\mathbf{P}_\perp$; and $\Rightarrow$ and $\Rightarrow_\perp$ are realized functors from $\mathbf{Q}\boldsymbol{\omega}\mathbf{P}_\perp^{op} \times \mathbf{Q}\boldsymbol{\omega}\mathbf{P}_\perp$ to $\mathbf{Q}\boldsymbol{\omega}\mathbf{P}_\perp$. For all constructions except $\mathcal{P}(\cdot)$ this follows from Theorem 2.6. For the powerdomain it is clear from the action of $\mathcal{P}(\cdot)$ on realizers described in Section 4.

We can now state the main theorem of this section. This shows that any realized category is algebraically compact relative to the class of realized endofunctors. Further, it shows that for a class of realized endofunctors given parametrically in another realized category, the initial-algebra-finding functor defined earlier is also a realized functor. This gives parameterized algebraic compactness in the sense of Fiore [5].

Theorem 6.3 (Parameterized algebraic compactness)

1. *Every realized endofunctor $\Phi : R\mathcal{C} \to R\mathcal{C}$ has a free algebra in $R\mathcal{C}$.*
2. *For any realized functor $\Psi : S\mathcal{D} \times R\mathcal{C} \to R\mathcal{C}$ the initial-algebra-finding functor $\Psi^\dagger : S\mathcal{D} \to R\mathcal{C}$ is realized.*

Theorem 6.3 is all that is needed to solve recursive domain equations. More specifically, one uses the parameterized algebraic compactness of $\mathbf{Q}\boldsymbol{\omega}\mathbf{P}_\perp \times \mathbf{Q}\boldsymbol{\omega}\mathbf{P}_\perp^{op}$, as all type constructors give rise to multi-arity functors on this category. Moreover the "symmetry" of such functors allows the solutions to be found "on the diagonal". The reader is referred to Fiore's thesis [5, Ch. 6] for full details.

The remainder of this section is devoted to the proof of Theorem 6.3. Let Φ be any realized endofunctor on $R\mathcal{C}$, and let F be any functor realizing Φ. The conditions on $\mathcal{C}$ in the definition of realized category are enough to guarantee that $\mathcal{C}$ is algebraically compact relative to all **Cpo**-enriched endofunctors [5, Ch. 7]. Thus $\mathcal{C}$ has a free F-algebra $FA \xrightarrow{a} A$. To prove Theorem 6.3(1), we would like to show that $\Phi A \xrightarrow{R(a)} A$ is a free Φ-algebra in $R\mathcal{C}$. This does not follow by elementary diagram chasing, but it does follow using an alternative equational characterization of free algebras as *special invariant objects.* These were introduced in the **Cpo**-enriched case by Freyd [6]. Our treatment, which fits into the general setting of [17], involves regarding $\mathcal{C}$ as an $\boldsymbol{\omega}\mathbf{P}$-enriched category and $R\mathcal{C}$ as a $\mathbf{Q}\boldsymbol{\omega}\mathbf{P}$-enriched one.

To emphasize the enrichment under consideration, we shall write $\boldsymbol{\omega}\mathbf{P}$-$\mathcal{C}$ for the $\boldsymbol{\omega}\mathbf{P}$-enriched version of $\mathcal{C}$ and $\mathbf{Q}\boldsymbol{\omega}\mathbf{P}$-$R\mathcal{C}$ for the $\mathbf{Q}\boldsymbol{\omega}\mathbf{P}$-enriched version of $R\mathcal{C}$. The hom-objects $\boldsymbol{\omega}\mathbf{P}$-$\mathcal{C}(A,B)$ and $\mathbf{Q}\boldsymbol{\omega}\mathbf{P}$-$R\mathcal{C}(A,B)$ are the same realized poset $(R\mathcal{C}(A,B),\, \mathcal{C}(A,B),\, R_{AB})$. The identity and composition maps are obvious. Note that composition in $\boldsymbol{\omega}\mathbf{P}$-$\mathcal{C}$ is a bistrict morphism in $\boldsymbol{\omega}\mathbf{P}$, and composition in $\mathbf{Q}\boldsymbol{\omega}\mathbf{P}$-$\mathcal{C}$ is a bistrict morphism in $\mathbf{Q}\boldsymbol{\omega}\mathbf{P}$. Note also that $Q : \boldsymbol{\omega}\mathbf{P} \to \mathbf{Q}\boldsymbol{\omega}\mathbf{P}$ maps the objects, identities and composition of $\boldsymbol{\omega}\mathbf{P}$-$\mathcal{C}$ to those of $\mathbf{Q}\boldsymbol{\omega}\mathbf{P}$-$R\mathcal{C}$. Thus $\mathbf{Q}\boldsymbol{\omega}\mathbf{P}$-$R\mathcal{C}$ is the $\mathbf{Q}\boldsymbol{\omega}\mathbf{P}$-category $Q_*\,\boldsymbol{\omega}\mathbf{P}$-$\mathcal{C}$ in the notation of Eilenberg and Kelly [4]. It follows that Q induces an evident (ordinary) functor (called $Q_{0\mathcal{C}}$ in [4]) from the underlying category $\mathcal{C}$ of $\boldsymbol{\omega}\mathbf{P}$-$\mathcal{C}$ to the underlying category $R\mathcal{C}$ of $\mathbf{Q}\boldsymbol{\omega}\mathbf{P}$-$R\mathcal{C}$. This functor is none other than $R : \mathcal{C} \to R\mathcal{C}$.

Next we consider how the enrichment extends to realized functors. Let $\Phi : R\mathcal{C} \to S\mathcal{D}$ be realized by $F : \mathcal{C} \to \mathcal{D}$. One sees that the continuous function $F_{AB} : \mathcal{C}(A,B) \to \mathcal{D}(FA,FB)$ realizes the monotone function $\Phi_{AB} : R\mathcal{C}(A,B) \to S\mathcal{D}(\Phi A, \Phi B)$ from the realized poset $(R\mathcal{C}(A,B),\, \mathcal{C}(A,B),\, R_{AB})$ to the realized poset $(S\mathcal{D}(A,B),\, \mathcal{D}(A,B),\, S_{AB})$. It follows that F enriches to an $\boldsymbol{\omega}\mathbf{P}$-functor from $\boldsymbol{\omega}\mathbf{P}$-$\mathcal{C}$ to $\boldsymbol{\omega}\mathbf{P}$-$\mathcal{D}$. Similarly, Φ enriches to a $\mathbf{Q}\boldsymbol{\omega}\mathbf{P}$-functor from $\mathbf{Q}\boldsymbol{\omega}\mathbf{P}$-$R\mathcal{C}$ to $\mathbf{Q}\boldsymbol{\omega}\mathbf{P}$-$S\mathcal{D}$. Further, $Q : \boldsymbol{\omega}\mathbf{P} \to \mathbf{Q}\boldsymbol{\omega}\mathbf{P}$ maps the action of F to the action of Φ. Thus Φ is the $\mathbf{Q}\boldsymbol{\omega}\mathbf{P}$-functor Q_*F in the notation of [4].

We can now define the alternative characterization of free algebras. Suppose that $\Phi : R\mathcal{C} \to R\mathcal{C}$ is realized by $F : \mathcal{C} \to \mathcal{C}$. Given an isomorphism $FA \xrightarrow{a} A$ in $\mathcal{C}$, the $\boldsymbol{\omega}\mathbf{P}$-enriched structure of $\mathcal{C}$ and F gives us an evident $\boldsymbol{\omega}\mathbf{P}$ morphism:

$$\boldsymbol{\omega}\mathbf{P}\text{-}\mathcal{C}(A,A) \xrightarrow{a \circ F(\perp) \circ a^{\perp 1}} \boldsymbol{\omega}\mathbf{P}\text{-}\mathcal{C}(A,A).$$

We say that $FA \xrightarrow{a} A$ is *special F-invariant* if $\mathit{fix}(a \circ F(\perp) \circ a^{\perp 1}) = 1_A$. Similarly, given an isomorphism $\Phi A \xrightarrow{\alpha} A$ in $R\mathcal{C}$, the $\mathbf{Q}\boldsymbol{\omega}\mathbf{P}$-enriched structure of $R\mathcal{C}$ and Φ gives us an evident $\mathbf{Q}\boldsymbol{\omega}\mathbf{P}$ morphism:

$$\mathbf{Q}\boldsymbol{\omega}\mathbf{P}\text{-}R\mathcal{C}(A,A) \xrightarrow{\alpha \circ \Phi(\perp) \circ \alpha^{\perp 1}} \mathbf{Q}\boldsymbol{\omega}\mathbf{P}\text{-}R\mathcal{C}(A,A).$$

We say that $\Phi A \xrightarrow{\alpha} A$ is *special Φ-invariant* if $\mathit{Qfix}(\alpha \circ \Phi(\perp) \circ \alpha^{\perp 1}) = 1_A$.

Proposition 6.4 *Suppose that $\Phi : R\mathcal{C} \to R\mathcal{C}$ is realized by $F : \mathcal{C} \to \mathcal{C}$ and that $FA \xrightarrow{a} A$ is an isomorphism. Consider the following statements.*

1. *$FA \xrightarrow{a} A$ is an initial F-algebra in $\mathcal{C}$.*
2. *$A \xrightarrow{a^{\perp 1}} FA$ is a terminal F-coalgebra in $\mathcal{C}$.*
3. *$FA \xrightarrow{a} A$ is special F-invariant.*

4. $\Phi A \xrightarrow{R(a)} A$ is an initial Φ-algebra in $R\mathcal{C}$.
5. $A \xrightarrow{R(a)^{\perp 1}} \Phi A$ is a terminal Φ-coalgebra in $R\mathcal{C}$.
6. $\Phi A \xrightarrow{R(a)} A$ is special Φ-invariant.

Then $1 \equiv 2 \equiv 3 \Rightarrow 4 \equiv 5 \equiv 6$.

Proof.

3 ⇒ 6. A straightforward consequence of Q preserving the fixed-point operator and all the enriched structure.
4 ⇒ 6. Suppose that $\Phi A \xrightarrow{\alpha} A$ is an initial Φ-algebra in $R\mathcal{C}$. Then it is easily checked that $Qfix(\alpha \circ \Phi(\perp) \circ \alpha^{\perp 1})$ is a Φ-algebra homomorphism from α to itself. But 1_A is the unique such homomorphism.
6 ⇒ 4. Suppose that $\Phi A \xrightarrow{\alpha} A$ is special Φ-invariant. We show that it is an initial Φ-algebra. Given any $\Phi B \xrightarrow{\beta} B$ we have a $\mathbf{Q\boldsymbol{\omega}P}$-morphism:

$$\mathbf{Q\boldsymbol{\omega}P}\text{-}R\mathcal{C}(A,B) \xrightarrow{\beta \circ \Phi(\perp) \circ \alpha^{\perp 1}} \mathbf{Q\boldsymbol{\omega}P}\text{-}R\mathcal{C}(A,B).$$

It is easily checked that $Qfix(\beta \circ \Phi(\perp) \circ \alpha^{\perp 1})$ is a Φ-algebra homomorphism from α to β. For uniqueness, let $A \xrightarrow{x} B$ be any such homomorphism. There is an evident $\mathbf{Q\boldsymbol{\omega}P}$-morphism:

$$\mathbf{Q\boldsymbol{\omega}P}\text{-}R\mathcal{C}(A,A) \xrightarrow{x \circ \perp} \mathbf{Q\boldsymbol{\omega}P}\text{-}R\mathcal{C}(A,B),$$

which, by the bistrictness of composition in $\mathbf{Q\boldsymbol{\omega}P}$-$R\mathcal{C}$, is strict. As x is a Φ-algebra homomorphism, the diagram below commutes:

$$\begin{tikzcd}
\mathbf{Q\boldsymbol{\omega}P}\text{-}R\mathcal{C}(A,A) \arrow[r, "\alpha \circ \Phi(\perp) \circ \alpha^{\perp 1}"] \arrow[d, "x \circ \perp"'] & \mathbf{Q\boldsymbol{\omega}P}\text{-}R\mathcal{C}(A,A) \arrow[d, "x \circ \perp"] \\
\mathbf{Q\boldsymbol{\omega}P}\text{-}R\mathcal{C}(A,B) \arrow[r, "\beta \circ \Phi(\perp) \circ \alpha^{\perp 1}"] & \mathbf{Q\boldsymbol{\omega}P}\text{-}R\mathcal{C}(A,B).
\end{tikzcd}$$

So, by the uniformity of $Qfix$, it holds that $(x \circ \perp) \circ Qfix(\alpha \circ \Phi(\perp) \circ \alpha^{\perp 1}) = Qfix(\beta \circ \Phi(\perp) \circ \alpha^{\perp 1})$. Whence, as α is special Φ-invariant, it follows that $x = Qfix(\beta \circ \Phi(\perp) \circ \alpha^{\perp 1})$ as required. (This argument is from [17].)
5 ≡ 6. Dual to $4 \equiv 6$.
1 ≡ 2 ≡ 3. By a similar proof to $4 \equiv 5 \equiv 6$. ⊠

Part 1 of Theorem 6.3 follows easily. By earlier remarks, $\mathcal{C}$ has a free F-algebra $FA \xrightarrow{a} A$. So, by the proposition, $\Phi A \xrightarrow{R(a)} A$ is a free Φ-algebra in $R\mathcal{C}$.

It remains to prove part 2. Suppose that $\Psi : S\mathcal{D} \times R\mathcal{C} \to R\mathcal{C}$ is realized by $G : \mathcal{D} \times \mathcal{C} \to \mathcal{C}$. For any object A of $\mathcal{D}$, let $G(A, C_A) \xrightarrow{a_A} C_A$ be a chosen free $G(A, \perp)$-algebra in $\mathcal{C}$. This choice determines the initial-alegra-finding functor

$G^\dagger : \mathcal{D} \to \mathcal{C}$. Now, for any $\mathcal{D}$ morphism $A \xrightarrow{f} B$, the $\boldsymbol{\omega}\mathbf{P}$-enriched structure of $\mathcal{C}$, $\mathcal{D}$ and G gives us an $\boldsymbol{\omega}\mathbf{P}$ morphism:

$$\boldsymbol{\omega}\mathbf{P}\text{-}\mathcal{C}(C_A, C_B) \xrightarrow{a_B \circ G(f,\perp) \circ a_A^{\perp 1}} \boldsymbol{\omega}\mathbf{P}\text{-}\mathcal{C}(C_A, C_B).$$

It is easy to check that diagram (1) for $G^\dagger$ still commutes if one replaces $G^\dagger(f)$ with $\mathit{fix}(a_B \circ G(f,\perp) \circ a_A^{\perp 1})$. Therefore $G^\dagger(f) = \mathit{fix}(a_B \circ G(f,\perp) \circ a_A^{\perp 1})$. Thus the following $\boldsymbol{\omega}\mathbf{P}$-morphism shows that $G^\dagger$ is $\boldsymbol{\omega}\mathbf{P}$-enriched, hence **Cpo**-enriched.

$$\boldsymbol{\omega}\mathbf{P}\text{-}\mathcal{D}(A, B) \xrightarrow{f \;\mapsto\; \mathit{fix}(a_B \circ G(f,\perp) \circ a_A^{\perp 1})} \boldsymbol{\omega}\mathbf{P}\text{-}\mathcal{C}(C_A, C_B).$$

By Proposition 6.4, $\Psi(A, C_A) \xrightarrow{R(a_A)} C_A$ is a free $\Psi(A,\perp)$-algebra in $R\mathcal{C}$. This choice determines $\Psi^\dagger : S\mathcal{D} \to R\mathcal{C}$. As above, one shows that, for any $S\mathcal{D}$ morphism $A \xrightarrow{\phi} B$, it holds that $\Psi^\dagger(\phi) = Q\mathit{fix}(\alpha_B \circ \Psi(\phi,\perp) \circ \alpha_A^{\perp 1})$ and that $\Psi^\dagger$ is $\mathbf{Q}\boldsymbol{\omega}\mathbf{P}$-enriched, hence **Pposet**-enriched. Moreover, as Q preserves the fixed-point-operator and enriched structure, it is clear that $G^\dagger$ realizes $\Psi^\dagger$ as required.

7 Further work

In this paper we have constructed and analysed a powerdomain in a new category of domains based on realized posets. Our original motivation for this investigation was to obtain a powerdomain avoiding Sieber's problem with full abstraction for his nondeterministic call-by-value PCF [16]. As discussed in Section 5, we do indeed avoid this particular problem. However, the question of whether we achieve full abstraction remains to be investigated.

Also in [16], Sieber points out the impossibility of achieving full abstraction for a treatment of total correctness based on the upper powerdomain in cpos. It is possible that a variation on our powerdomain, using a cpo of intensional representations of necessarily terminating nondeterministic programs, might help with this problem too. It would be interesting to see if such a treatment arises from the standard algebraic characterization of the upper powerdomain [14]. Partial correctness and the lower powerdomain are perhaps less interesting, as these can already be adequately treated using the existing lower powerdomain in **Cpo** (again see [16]). Nevertheless, for the sake of completeness, it would be worth having an analogous construction on realized posets. It would also be interesting to investigate whether the logical characterizations of the powerdomains using modalities on observable properties (as in [15]) hold in our setting. Indeed, the informal motivation for considering extensional preorders on intensional cpos, given in the introduction, could be best underpinned by a proper theory of intensional and extensional observable properties.

It would also be worthwhile to investigate interpretations of more complicated nondeterministic (and concurrent) behaviours. For this it might be natural to move from $\mathbf{Q}\boldsymbol{\omega}\mathbf{P}_\perp$ to the category of realized semilattices and strict linear morphisms. As mentioned in Section 5, this category should allow one to model trace equivalence. It should also support important variant powerdomains. For example, every realized semilattice has a second associated partial order, the *inclusion order*, determined by the semilattice structure. This order is preserved by every linear morphism. Thus there should be a forgetful functor from the category of realized semilattices to an appropriate category of realized *double posets* (sets with two partial orders). The left adjoint to this forgetful should produce a powerdomain generating the free inclusion-order-preserving semilattice. It appears that such a powerdomain might provide a good (fully abstract?) interpretation of Sieber's nondeterministic PCF without its computationally unnatural **exists** operator.

Another direction for research is to consider variations on the construction of $\mathbf{Q\omega P}$. Rather than working with cpos equipped with ω-inductive preorders, one could use ω-inductive equivalence relations, partial equivalence relations or partial preorders instead. The third of these would give rise to a different category of realized posets, the other two to categories of realized sets. It would be interesting to see all the variations as instances of some general construction. Quite possibly the associated categories of realized categories and realized functors would also be instances, perhaps at the 2-categorical level. It would be remarkable if it were also possible to incorporate the standard realizability examples (such as the category of *modest sets* [10]) into the same theory. It would also be interesting to relate these constructions to the categories of games defined by Abramsky *et al*, in there is a similar passage from intensional representations to extensional equivalence [1].

It appears that the above variations on $\mathbf{Q\omega P}$ may well have interesting applications. For example, sometimes the identification of any two nondeterministic computations whose sets of results have the same convex closure is not desirable. For us such identifications were forced by working with a category of realized posets. It seems that one approach to avoiding them would be to work instead with one of the categories of realized sets mentioned above.

In general, the main benefit of these realizability categories seems to be that quotients are more easily defined than in **Cpo**. For example, in $\mathbf{Q\omega P}$ coequalisers leave the underlying cpo unchanged, modifying only the extensional preorder. There are many situations in denotational semantics in which one easily constructs domains of intensional representations, but has difficulty in quotienting them. It is plausible that realizability categories might help with such problems. Indeed, independently of us, Andy Pitts has been led to consider similar realizability categories in order to resolve similar quotienting problems associated with the interpretation of programs involving local variables (private communication). Another speculative application is to concurrency where one might be able to model weak bisimulation as a quotient of strong bisimulation.

Acknowledgements

I would like to thank Pietro Di Gianantonio, Andy Pitts and Gordon Plotkin for their encouragement and for a number of helpful suggestions. The author is supported by an EPSRC postdoctoral research fellowship.

References

1. S. Abramsky, R. Jagadeesan, and P. Malacaria. Full abstraction for PCF (extended abstract). In *Proceedings of TACS '94*. Springer LNCS 789, 1994.
2. S. O. Anderson and A. J. Power. *A representable approach to nondeterminism.* Submitted to *Proceedings of MFPS '94*, 1994.
3. G. M. Bierman. What is a categorical model of intuitionistic linear logic? In *Typed Lambda Calculi and Applications, Proceedings of TLCA '95*. Springer LNCS 902, 1995.
4. S. Eilenberg and G. M. Kelly. Closed categories. In *Proceedings of the Conference on Categorical Algebra, La Jolla 1965*, pages 421–562. Springer, 1966.
5. M. P. Fiore. *Axiomatic Domain Theory in Categories of Partial Maps.* Ph.D. thesis, Department of Computer Science, University of Edinburgh. Available as ECS-LFCS-94-307, 1994.
6. P. J. Freyd. Recursive types reduced to inductive types. In *Proceedings of 5th Annual Symposium on Logic in Computer Science*, pages 498 – 507, 1990.
7. P. J. Freyd. Algebraically complete categories. In *Category Theory, Proceedings Como 1990*, Springer LNM 1488, 1991.
8. P. J. Freyd. Remarks on algebraically compact categories. In *Applications of Categories in Computer Science*, pages 95–106. LMS Lecture Notes 177, CUP, 1992.
9. M. Hennessy and G. D. Plotkin. Full abstraction for a simple parallel programming language. In *Mathematical Foundations of Computer Science*, pages 108–120. Springer LNCS 74, 1979.
10. J. M. E. Hyland. A small complete category. *Annals of Pure and Applied Logic*, 40:135 – 165, 1988.
11. G. M. Kelly. *Basic Concepts of Enriched Category Theory.* LMS Lecture Notes 64, CUP, 1982.
12. A. Kock. Monads on symmetric monoidal closed categories. *Archiv der Mathematik*, 21:1 – 10, 1970.
13. G. D. Plotkin. A powerdomain construction. *SIAM Journal of Computing*, 5:452–487, 1976.
14. G. D. Plotkin. *Domains.* Lecture notes, Department of Computer Science, University of Edinburgh, 1983.
15. E. P. Robinson. Logical aspects of denotational semantics. In M. Bezem and J. F. Groote, editors, *Category Theory and Computer Science, Proceedings of CTCS '87.* Springer LNCS 283, 1987.
16. K. Sieber. Call-by-value and nondeterminism. In M. Bezem and J. F. Groote, editors, *Typed Lambda Calculi and Applications, Proceedings of TLCA '93*, pages 376–390. Springer LNCS 664, 1993.
17. A. K. Simpson. *Recursive types in Kleisli categories.* Available by FTP from ftp.dcs.ed.ac.uk/pub/als/kleisli.ps.Z, 1992.
18. M.B. Smyth and G. D. Plotkin. The category theory solution of recursive domain equations. *SIAM Journal of Computing*, 11:761 – 783, 1982.