Name Generation and Linearity

Glynn Winskel, University of Cambridge Computer Laboratory, England

Abstract

A path-based domain theory for higher-order processes is extended to allow name generation. The original domain theory is built around the monoidal-closed category $\mathbf{Lin}$ *consisting of path orders with join-preserving functions between their domains of path sets. Name generation is adjoined by forming the functor category* $[\mathcal{I}, \mathbf{Lin}]$*, where* $\mathcal{I}$ *consists of finite sets of names and injections. The functor category* $[\mathcal{I}, \mathbf{Lin}]$ *is no longer monoidal-closed w.r.t. the tensor inherited pointwise from* $\mathbf{Lin}$*. However, conditions are given under which function spaces exist. The conditions are preserved by a rich discipline of linear types, including those of new-HOPLA, a recent powerful language for higher-order processes with name generation.*

1. Introduction

The intricate models of distributed computation argue for a new domain theory for concurrency based on computation paths, in which processes denote generalised characteristic functions [9]. This approach encompasses both the view of processes as path sets (by taking truth values 0, 1) and the much richer view of processes as presheaves (by taking truth values to be sets). The ensuing categories of 'domains' are also models of linear logic, which exposes the central role of linearity in distributed computation, where by their very nature processes can often be run only once [2, 9].

A standard way to adjoin name generation (as in the pi-Calculus) to a category of domains is to move to a functor category, so both processes and their types are indexed by the current set of names. But then, for path-based models, not all linear function spaces need exist. For a while now [6, 1] there's been the question of what restrictions are needed both syntactically and mathematically on types and their functor denotations to permit the formation of function spaces. Here a solution is given for the domain theory based on path sets. Sufficient conditions are given for linear function spaces to exist, and their need illustrated through an example. The conditions are liberal enough to allow a rich syntax of types, including those of new-HOPLA [11]. The solution gives a concrete understanding of higher-order paths, so it should facilitate proofs of adequacy and full abstraction for new-HOPLA along the lines of those for HOPLA in [9]. It should also lead the way to understanding those linear function spaces that exist for presheaf models with name generation, extending early work on fully-abstract presheaf semantics of pi-Calculus to higher-order [1] .

Domain theory from path sets

We give a quick review of the domain theory for processes based on path sets [9]. The objects of the category $\mathbf{Lin}$, *path orders*, are preorders $\mathbb{P}$ consisting of computation paths with the preorder $p \le p'$ expressing how a path p extends to a path p'. A path order $\mathbb{P}$ determines a domain $\widehat{\mathbb{P}}$, that of its *path sets*, down-closed sets w.r.t. $\le_{\mathbb{P}}$, ordered by inclusion. Such a domain is a *prime-algebraic complete lattice* [8], in which the (complete) *primes* are precisely those path sets ${\downarrow}p$ generated by individual paths p. The arrows of $\mathbf{Lin}$, *linear* maps, from $\mathbb{P}$ to $\mathbb{Q}$ are join-preserving functions from $\widehat{\mathbb{P}}$ to $\widehat{\mathbb{Q}}$. The category $\mathbf{Lin}$ is monoidal-closed with a tensor given the product $\mathbb{P} \times \mathbb{Q}$ of path orders and a corresponding function space by $\mathbb{P}^{\mathrm{op}} \times \mathbb{Q}$. $\mathbf{Lin}$ has enough structure to form a model of Girard's classical linear logic [5]. The exponential $!\mathbb{P}$ consists of finite elements of $\widehat{\mathbb{P}}$ under inclusion—$!\mathbb{P}$ can be thought of as consisting of compound paths associated with several runs. Its coKleisli category consists of path orders with Scott-continuous functions between the domains of path sets. The idea is that a process denotes a path set, its possible paths within its type, denoted by a path order.

Domain theory for name generation

To cope with name generation we move to a functor category in which domains of path sets are indexed functorially by the current set of names [3, 10]. The category $\mathcal{I}$ consists of finite sets of names related by injections. The functor category $[\mathcal{I}, \mathbf{Lin}]$ has as objects functors $\mathbb{P} : \mathcal{I} \to \mathbf{Lin}$, so path orders $\mathbb{P}(s)$ indexed by finite sets of names s standing for the computation paths possible with that current set of names; its arrows are natural transformations with components in $\mathbf{Lin}$. One important object in $[\mathcal{I}, \mathbf{Lin}]$ is the object of names $\mathbb{N}$ providing the current set of names, so $\mathbb{N}(s)$ is the discrete order s at name set s. There is a tensor got pointwise from the tensor of $\mathbf{Lin}$. Given $\mathbb{P}$ and $\mathbb{Q}$ in $[\mathcal{I}, \mathbf{Lin}]$ we define $\mathbb{P} \otimes \mathbb{Q}$ in $[\mathcal{I}, \mathbf{Lin}]$ so that at $s \in \mathcal{I}$

$$\mathbb{P} \otimes \mathbb{Q}(s) = \mathbb{P}(s) \times \mathbb{Q}(s)\ .$$

The comonad ! on $\mathbf{Lin}$ lifts to a comonad on $[\mathcal{I}, \mathbf{Lin}]$ whose coKleisli category consists of the same objects but where the natural transformations have continuous components.

To support higher-order processes we need function spaces $\mathbb{P} \multimap \mathbb{Q}$ such that

$$[\mathcal{I}, \mathbf{Lin}](\mathbb{R}, \mathbb{P} \multimap \mathbb{Q}) \cong [\mathcal{I}, \mathbf{Lin}](\mathbb{R} \otimes \mathbb{P}, \mathbb{Q})$$

natural in $\mathbb{R}$ and $\mathbb{Q}$. But there is no reason to expect such function spaces to exist in general—the obstacle is there not necessarily being a path order $(\mathbb{P} \multimap \mathbb{Q})s$ at each name set s. This paper provides general conditions which ensure the existence of linear function spaces, and moreover which are preserved by linear function space. The conditions imply that function spaces of the form

$$\mathbb{N} \otimes \cdots \otimes \mathbb{N} \otimes !\mathbb{P}_1 \otimes \cdots \otimes !\mathbb{P}_k \multimap \mathbb{Q}$$

will exist and satisfy the conditions; in particular, all the types of new-HOPLA meet the conditions [11].

2. Towards the linear function space

We are interested in linear function spaces $\mathbb{P} \multimap \mathbb{Q}$, for which there is an isomorphism

$$[\mathcal{I}, \mathbf{Lin}](\mathbb{R}, \mathbb{P} \multimap \mathbb{Q}) \cong [\mathcal{I}, \mathbf{Lin}](\mathbb{R} \otimes \mathbb{P}, \mathbb{Q})\ .$$

Using the Yoneda lemma, we can make an educated guess, that for a name set s_0

$$\begin{aligned} \widehat{(\mathbb{P} \multimap \mathbb{Q})}s_0 &\cong [\mathcal{I}, \mathbf{Lin}](\mathcal{I}(s_0, -), \mathbb{P} \multimap \mathbb{Q}) \\ &\cong [\mathcal{I}, \mathbf{Lin}](\mathcal{I}(s_0, -) \otimes \mathbb{P}, \mathbb{Q}) \\ &\cong [\mathcal{I}, \mathbf{Lin}](\mathbb{P}(s_0 + \_), \mathbb{Q}(s_0 + \_))\ , \end{aligned}$$

where the order is to be pointwise. Given a name set s the hom-functor $\mathcal{I}(s_0, -)$ yields the set $\mathcal{I}(s_0, s)$ which we identify with a discrete path order. The last isomorphism comes about by observing that natural transformations with components $\alpha_s : \mathcal{I}(s_0, s) \otimes \mathbb{P}s \to \mathbb{Q}s$ are in bijection with natural transformations with components $\beta_{s'} : \mathbb{P}(s_0 + s') \to \mathbb{Q}(s_0 + s')$ such that $\beta_{s'}(p) = \alpha_{s_0+s'}(in_0, p)$, where $in_0 : s_0 \to s_0 + s'$ is the obvious injection.

The implied isomorphism

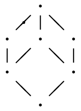
$$\widehat{(\mathbb{P} \multimap \mathbb{Q})}s_0 \cong [\mathcal{I}, \mathbf{Lin}](\mathbb{P}(s_0 + \_), \mathbb{Q}(s_0 + \_))$$

provides an intuitive reading of the domain of the linear function space. It consists of natural transformations between the functors denoting types updated with the current set of names s_0. But of course such isomorphisms only tell us the form of the domain, and not the path order $(\mathbb{P} \multimap \mathbb{Q})s_0$ from which the domain is derived. Obtaining the path order requires insight, or can be impossible, as the following examples show.

Example 2.1 Instantiating $\mathbb{P}$ to the type of names $\mathbb{N}$ we obtain $(\mathbb{N} \multimap \mathbb{Q})s_0 \cong [\mathcal{I}, \mathbf{Lin}](s_0 + \_, \mathbb{Q}(s_0 + \_))$ consisting of natural transformations

$$\begin{tikzcd} s_0 \arrow[r] \arrow[d] & s_0{+}1 \arrow[r] \arrow[d] & s_0{+}s \arrow[d, "\cdots"] \\ \mathbb{Q}(s_0) \arrow[r] & \mathbb{Q}(s_0{+}1) \arrow[r] & \mathbb{Q}(s_0{+}s) \end{tikzcd}$$

From their naturality, such natural transformations are determined by the initial square, so in bijection with path sets over $s_0 \times \mathbb{Q}(s_0) + \mathbb{Q}(s_0 + 1)$—over the first component the path set determines how names in s_0 are mapped and over the second component how a generic ‘new’ name 1 is mapped. We can realise $(\mathbb{N} \to \mathbb{Q})s_0$ as the path order $s_0 \times \mathbb{Q}(s_0) + \mathbb{Q}(s_0 + 1)$.

Example 2.2 In general $[\mathcal{I}, \mathbf{Lin}](\mathbb{P}, \mathbb{Q})$ need not be distributive, so definitely not representable by a path order $(\mathbb{P} \multimap \mathbb{Q})\emptyset$. For instance, suppose $\mathbb{Q}$ is constantly $\mathbf{1}$, the path order with one point. Suppose $\mathbb{P}\emptyset$ is the order $a_0 < a_1$ and $\mathbb{P}s$ is the discrete order with two elements b_1, b_2, for $s \neq \emptyset$. Let $\mathbb{P}(\emptyset \hookrightarrow s) : a_1 \mapsto \{b_1, b_2\}; a_0 \mapsto \emptyset$; and give the identity elsewhere. Then naturality constrains $[\mathcal{I}, \mathbf{Lin}](\mathbb{P}, \mathbb{Q})$ ordered pointwise to have Hasse diagram , which is not distributive.

As Example 2.1 makes clear, it can be a matter of some thought to determine the path order $(\mathbb{P} \multimap \mathbb{Q})s_0$. Especially so as the existence of this path order requires extra conditions on the type functor $\mathbb{P}$—Example 2.2. In future we will use the characterisation

$$\widehat{(\mathbb{P} \multimap \mathbb{Q})}s_0 \cong [\mathcal{I}, \mathbf{Lin}](\mathcal{I}(s_0, -) \otimes \mathbb{P}, \mathbb{Q})$$

as this exposes the functorial action of $\mathbb{P} \multimap \mathbb{Q}$ most clearly. The first step is to exhibit a path order $(\mathbb{P} \multimap \mathbb{Q})s_0$ with domain of path sets isomorphic to the pointwise order on natural transformations $\mathcal{I}(s_0, -) \otimes \mathbb{P} \Rightarrow \mathbb{Q}$. This will be under the conditions that $\mathbb{P}$ and $\mathbb{Q}$ are *type functors*, where $\mathbb{P}$ in addition *respects primes*. (I presently have no example to show that the latter additional restriction is necessary, *i.e.* conceivably $\mathbb{P} \multimap \mathbb{Q}$ exists for all type functors $\mathbb{P}$ and $\mathbb{Q}$.)

3. Type functors

Definition 3.1 Say $\mathbb{Q} : \mathcal{I} \to \mathbf{Lin}$ is a *type functor* when it respects finites, nonempty meets, and pullbacks, in the sense of the following definitions:

Respects finites: The function $\mathbb{Q}f : \widehat{\mathbb{Q}s} \to \widehat{\mathbb{Q}s'}$ sends finite elements of the domain $\widehat{\mathbb{Q}s}$ to finite elements of $\widehat{\mathbb{Q}s'}$ for all $f : s \to s'$ in $\mathcal{I}$; being an arrow in $\mathbf{Lin}$, the function $\mathbb{Q}f$ preserves joins, so it is sufficient that $\mathbb{Q}f$ sends all primes to finite elements.

Respects nonempty meets: The function $\mathbb{Q}f : \widehat{\mathbb{Q}s} \to \widehat{\mathbb{Q}s'}$ preserves nonempty meets for all $f : s \to s'$ in $\mathcal{I}$; the empty meet, the maximum path set consisting of all $\mathbb{Q}s$ need not be sent to the maximum path set consisting of all $\mathbb{Q}s'$.

Supposing $\mathbb{Q}$ respects nonempty meets, and letting $y \in \widehat{\mathbb{Q}s'}$, we define

$$\min(\mathbb{Q}f, y) = \bigcap \{x \in \widehat{\mathbb{Q}s} \mid y \subseteq \mathbb{Q}f(x)\}$$

provided there exists x such that $y \subseteq \mathbb{Q}f(x)$, and to be undefined otherwise. If defined, $\min(\mathbb{Q}f, y)$ is the minimum input to $\mathbb{Q}f$ which yields y. We have $\min(\mathbb{Q}f, \bigcup_{j\in J} y_j) = \bigcup_{j\in J} \min(\mathbb{Q}f, y_j)$, one side being defined iff the other is. If y is prime, then if defined $\min(\mathbb{Q}f, y)$ is necessarily also prime—this follows because $\mathbb{Q}f$ is join preserving. If $\min(\mathbb{Q}f, {\downarrow}q')$ is defined for $q' \in \mathbb{Q}s'$, then

$$\min(\mathbb{Q}f, {\downarrow}q') = {\downarrow}q$$

for some $q \in \mathbb{Q}s$. We often identify a path q with its associated prime ${\downarrow}q$, and *e.g.* write the above as $\min(\mathbb{Q}f, q') = q$. The partial function $\min$ respects composition.

Respects pullbacks: Let

$$\begin{tikzcd} & s_1 \arrow[dr, "g_1"] & \\ s_0 \arrow[ur, "h_1"] \arrow[dr, "h_2"'] & & s_3 \\ & s_2 \arrow[ur, "g_2"'] & \end{tikzcd}$$

be a pullback in $\mathcal{I}$. If $x_1 \in \widehat{\mathbb{Q}s_1}$, $x_2 \in \widehat{\mathbb{Q}s_2}$, and $x_3 \in \widehat{\mathbb{Q}s_3}$ where

$$x_2 = \min(\mathbb{Q}g_2, x_3) \text{ and } x_3 = \mathbb{Q}g_1(x_1)\ ,$$

then there exists $x_0 \in \widehat{\mathbb{Q}s_0}$ such that

$$x_0 = \min(\mathbb{Q}h_1, x_1) \text{ and } x_2 = \mathbb{Q}h_2(x_0)\ .$$

Observe that if x_1 is a join then the pullback condition holds for x_1 iff it holds for every component of the join. Consequently, in verifying the pullback condition it is sufficient to assume that x_1 is prime—then x_0 will also have to be prime. When verifying the pullback condition it is helpful to note that under its assumptions it requires:

(1) the existence of $x_0 = \min(\mathbb{Q}h_1, x_1)$—this already ensures that $\mathbb{Q}h_2(x_0) \supseteq x_2$ as $\mathbb{Q}g_2(\mathbb{Q}h_2(x_0))$ includes x_3 and so must dominate $\min(\mathbb{Q}g_2, x_3)$;

(2) that $x_2 = \mathbb{Q}h_2(x_0)$, for which it is now seen to be sufficient to verify that $\mathbb{Q}h_2(x_0) \subseteq x_2$.

We will sometimes need to strengthen the conditions on a functor $\mathbb{Q} : \mathcal{I} \to \mathbf{Lin}$ and say it

Respects primes: The function $\mathbb{Q}f : \widehat{\mathbb{Q}s} \to \widehat{\mathbb{Q}s'}$ sends primes of the domain $\widehat{\mathbb{Q}s}$ to primes of $\widehat{\mathbb{Q}s'}$ for all $f : s \to s'$ in $\mathcal{I}$.

Proposition 3.2 *Suppose a functor $\mathbb{Q} : \mathcal{I} \to \mathbf{Lin}$ respects pullbacks. Then,*

(i) $\min(\mathbb{Q}f, \mathbb{Q}f(x)) = x$ for all arrows $f : s \to s'$ in $\mathcal{I}$ and $x \in \widehat{\mathbb{Q}s}$;

(ii) for all arrows $f : s \to s'$ in $\mathcal{I}$ and $q \in \mathbb{Q}s$ there exists $q' \in \mathbb{Q}s'$ such that $\min(\mathbb{Q}f, {\downarrow}q') = {\downarrow}q$;

(iii) $\mathbb{Q}f$ is an order monic for all arrows $f : s \to s'$ in $\mathcal{I}$, i.e. if $\mathbb{Q}f(x) \subseteq \mathbb{Q}f(y)$, then $x \subseteq y$ for all $x, y \in \widehat{\mathbb{Q}s}$;

(iv) $\mathbb{Q}$ preserves pullbacks.

Proposition 3.3 *Let s_0 be a name set. The Yoneda functor $\mathcal{I}(s_0, -)$ is a type functor which respects primes. If $\mathbb{R}$ and $\mathbb{Q}$ are type functors, so is the functor $\mathbb{R} \otimes \mathbb{Q}$; if further $\mathbb{R}$ and $\mathbb{Q}$ respect primes, then so does $\mathbb{R} \otimes \mathbb{Q}$. If $\mathbb{Q}$ is a type functor, then so is $!\mathbb{Q}$; moreover, $!\mathbb{Q}$ respects primes.*

From Proposition 3.3, $\mathcal{I}(s_0, -) \otimes \mathbb{P}$ is a prime-respecting type functor if $\mathbb{P}$ is. For this reason finding a path order for the linear function space at a name set s_0 will involve us with the path-order representation of natural transformations from a prime-respecting type functor to a type functor.

4. Natural transformations as path sets

Throughout this section let $\mathbb{P}$ and $\mathbb{Q}$ be type functors where in addition $\mathbb{P}$ is assumed to respect primes. We first concentrate on showing that $[\mathcal{I}, \mathbf{Lin}](\mathbb{P}, \mathbb{Q})$, natural transformations from $\mathbb{P}$ to $\mathbb{Q}$ ordered pointwise, can be represented as the domain of path sets of a path order $(\mathbb{P} \Rightarrow \mathbb{Q})$.

Definition 4.1 Let $\mathbb{P}$ and $\mathbb{Q}$ be type functors. An IO-tuple for $\mathbb{P}, \mathbb{Q}$ comprises (s, p, q) where s is a name set, $p \in \mathbb{P}s$, and $q \in \mathbb{Q}s$. We shall often write such a tuple as spq.

Definition 4.2 Let spq be an IO-tuple for $\mathbb{P}$, $\mathbb{Q}$. Say spq is *feasible* for $\mathbb{P}, \mathbb{Q}$ iff for all $f : s' \to s$ in $\mathcal{I}$, $p' \in \mathbb{P}s'$

$$p \leq \mathbb{P}f(p') \Rightarrow \exists q' \in \mathbb{Q}s'.\ q \in \mathbb{Q}f(q')\ .$$

Let spq and $s'p'q'$ be IO-tuples for $\mathbb{P}$, $\mathbb{Q}$.

Define $spq \xrightarrow{f}_F s'p'q'$ iff $f : s \to s'$ in $\mathcal{I}$ and

$$\downarrow p' = \mathbb{P}f(p) \ \& \ q' \in \mathbb{Q}f(q) \ .$$

Define $s'p'q' \lesssim_F spq$ iff there exists $f : s \to s'$ such that $spq \xrightarrow{f}_F s'p'q'$.
Define $s'p'q' \xrightarrow{f}_B spq$ iff $f : s' \to s$ in $\mathcal{I}$ and

$$\mathbb{P}f(p') = \downarrow p \ \& \ \downarrow q' = \min(\mathbb{Q}f, q) \ .$$

Define $s'p'q' \lesssim_B spq$ iff there exists $f : s' \to s$ such that $s'p'q' \xrightarrow{f}_B spq$.
Define $s'p'q' \lesssim_M spq$ iff $s = s'$ & $q = q'$ & $p \leq p'$ in $\mathbb{P}s$.
Define $\lesssim$ to be the least preorder including $\lesssim_F$, $\lesssim_B$ and $\lesssim_M$.

Observe that if $s'p'q' \xrightarrow{f}_B spq$ then automatically $s'p'q' \xrightarrow{f}_F spq$; so the relation $\lesssim_B$ is included in $\gtrsim_F$, the converse of the relation $\lesssim_F$. Note that F-arrows compose, as do B-arrows.

Proposition 4.3 *Let spq and $s'p'q'$ be IO-tuples for $\mathbb{P}$, $\mathbb{Q}$ at s_0. If spq is feasible and $s'p'q' \lesssim spq$, then $s'p'q'$ is feasible.*

Proof. It suffices to show the required property for $\lesssim_F$, $\lesssim_B$ and $\lesssim_M$. For $\lesssim_B$ and $\lesssim_M$ the proof is obvious, while for $\lesssim_F$ we make use of the fact that $\mathbb{P}$ respects pullbacks. □

Definition 4.4 Define $(\mathbb{P} \Rightarrow \mathbb{Q})$ to be the path order consisting of feasible IO-tuples for $\mathbb{P}$, $\mathbb{Q}$ under $\lesssim$.

Theorem 4.5

(i) Let $\alpha : \mathbb{P} \Rightarrow \mathbb{Q}$. Then its graph

$$\mathcal{G}(\alpha) = \{spq \mid q \in \alpha_s(p)\}$$

is a $\lesssim$-down-closed subset of $(\mathbb{P} \Rightarrow \mathbb{Q})$.

(ii) Let U be a $\lesssim$-down-closed subset of $(\mathbb{P} \Rightarrow \mathbb{Q})$. Then $\sigma(U)$ comprising a family $\sigma(U)_s : \mathbb{P}s \to \mathbb{Q}s$ such that

$$\sigma(U)_s(p) = \{q \mid spq \in U\}$$

for $s \in \mathcal{I}$, is a natural transformation $\sigma(U) : \mathbb{P} \Rightarrow \mathbb{Q}$.

(iii) The functions $\mathcal{G}$ and σ are mutual inverses determining an order isomorphism between natural transformations $[\mathcal{I}, Lin](\mathbb{P}, \mathbb{Q})$, ordered pointwise, and the domain $\widehat{(\mathbb{P} \Rightarrow \mathbb{Q})}$, ordered by inclusion.

Proof.
(i) Let $\alpha : \mathbb{P} \Rightarrow \mathbb{Q}$. First notice that $\mathcal{G}(\alpha)$ must consist of feasible tuples; otherwise we would violate the naturality of α. We check $\mathcal{G}(\alpha)$ is $\lesssim$-down-closed. For this it suffices to show that $\mathcal{G}(\alpha)$ is down-closed w.r.t. $\lesssim_B$, $\lesssim_F$ and $\lesssim_M$.

($\lesssim_B$) Let $f : s' \to s$. It is associated with the naturality square

$$\begin{tikzcd}
\mathbb{P}s' \arrow[r, "\alpha_{s'}"] \arrow[d, "f"'] & \mathbb{Q}s' \arrow[d, "\mathbb{Q}f"] \\
\mathbb{P}s \arrow[r, "\alpha_s"] & \mathbb{Q}s \ .
\end{tikzcd}$$

Suppose $spq \in \mathcal{G}(\alpha)$, *i.e.* $q \in \alpha_s(p)$, and $\mathbb{P}f(p') = \downarrow p$ and $\downarrow q' = \min(\mathbb{Q}f, q)$. Then $q \in (\alpha_s \circ \mathbb{P}f)(p')$. Hence by naturality, $q \in (\mathbb{Q}f)(\alpha_{s'}(p'))$. But then $\downarrow q' = \min(\mathbb{Q}f, q) \subseteq \alpha_{s'}(p')$. Hence $s'p'q' \in \mathcal{G}(\alpha)$.

($\lesssim_F$) Let $f : s \to s'$. Again it is associated with a naturality square

$$\begin{tikzcd}
\mathbb{P}s \arrow[r, "\alpha_s"] \arrow[d, "f"'] & \mathbb{Q}s \arrow[d, "\mathbb{Q}f"] \\
\mathbb{P}s' \arrow[r, "\alpha_{s'}"] & \mathbb{Q}s' \ .
\end{tikzcd}$$

Suppose $spq \in \mathcal{G}(\alpha)$ with $\downarrow p' = \mathbb{P}f(p)$ and $q' \in \mathbb{Q}f(q)$. This time, $q' \in (\mathbb{Q}f) \circ \alpha_s(p)$. So by naturality,

$$q' \in \alpha_{s'}(p').$$

Hence $s'p'q' \in \mathcal{G}(\alpha)$.

($\lesssim_M$) Closure under this condition follows directly from the monotonicity of each component α_s.

(ii) Let U be a $\lesssim$-down-closed subset of feasible tuples. By $\lesssim_M$-downclosure each component $\sigma(U)_s$ is monotonic and so (corresponds to) a linear map. We need naturality. Let $f : s \to s'$ in $\mathcal{I}$. We require that

$$\begin{tikzcd}
\mathbb{P}s \arrow[r, "\sigma(U)_s"] \arrow[d, "\mathbb{P}f"'] & \mathbb{Q}s \arrow[d, "\mathbb{Q}f"] \\
\mathbb{P}s' \arrow[r, "\sigma(U)_{s'}"] & \mathbb{Q}s' \ .
\end{tikzcd}$$

commutes. Suppose $p \in \mathbb{P}s$. Then, proceeding both ways round the square,

$$\sigma(U)_{s'}(\mathbb{P}f(p)) = \{q' \mid (s', \mathbb{P}f(p), q') \in U\}, \text{ and}$$
$$\mathbb{Q}f(\sigma(U)_s(p)) = \bigcup \{\mathbb{Q}f(q) \mid spq \in U\} \ .$$

Hence, if $q' \in \sigma(U)_{s'}(\mathbb{P}f(p))$, then $(s', Pf(p), q') \in U$ and is therefore a feasible tuple. Thus $q = \min(\mathbb{Q}f, q')$ exists. Consequently,

$$spq \lesssim_B (s', Pf(p), q') \in U \ .$$

Thus $spq \in U$. It follows that $q' \in \mathbb{Q}f(\sigma(U)_s(p))$.

Conversely, if $q' \in \mathbb{Q}f(\sigma(U)_s(p))$, then $q' \in \mathbb{Q}f(q)$ and $spq \in U$, for some $q \in \mathbb{Q}s$. But now,

$$(s', \mathbb{P}f(p), q') \lesssim_F spq \in U \ ,$$

where $(s', \mathbb{P}f(p), q')$ is necessarily feasible by Proposition 4.3. It follows that $q' \in \sigma(U)_{s'}(\mathbb{P}f(p))$.

Hence the required naturality square commutes and we have established that $\sigma(U) : \mathbb{P} \Rightarrow \mathbb{Q}$.

(iii) Directly from the definitions of $\mathcal{G}$ and σ:

$$\begin{aligned}
\sigma(\mathcal{G}(\alpha))_s(p) &= \{q \mid spq \in \mathcal{G}(\alpha)\} \\
&= \{q \mid q \in \alpha_s(p)\} \\
&= \alpha_s(p) \ , \text{ and} \\
\mathcal{G}(\sigma(U)) &= \{spq \mid q \in \sigma(U)_s(p)\} \\
&= \{spq \mid spq \in U\} \\
&= U \ .
\end{aligned}$$

Hence the functions $\mathcal{G}$ and σ are mutual inverses.

Suppose for down-closed sets, $U_1 \subseteq U_2$. Then $\sigma(U_1)_s(p) \subseteq \sigma(U_2)_s(p)$ for all $s \in \mathcal{I}$ and $p \in \mathbb{P}s$, ensuring $\sigma(U_1) \sqsubseteq \sigma(U_2)$ in the pointwise order.

Suppose $\alpha^1 \sqsubseteq \alpha^2$ pointwise, *i.e.* for all $s \in \mathcal{I}$ and $p \in \mathbb{P}s$ we have $\alpha^1_s(p) \subseteq \alpha^2_s(p)$. Then clearly, $\mathcal{G}(\alpha^1) \subseteq \mathcal{G}(\alpha^2)$.

Hence the pair $\mathcal{G}, \sigma$ forms an order isomorphism. □

Corollary 4.6
(i) An IO-tuple spq for $\mathbb{P}$, $\mathbb{Q}$ is feasible iff there is a natural transformation $\alpha : \mathbb{P} \Rightarrow \mathbb{Q}$ for which $q \in \alpha_s(p)$.
(ii) Let spq and $s'p'q'$ be IO-tuples for $\mathbb{P}$, $\mathbb{Q}$. Let spq be feasible. Then,

$$s'p'q' \lesssim spq$$

iff for all natural transformations $\alpha : \mathbb{P} \Rightarrow \mathbb{Q}$

$$q \in \alpha_s(p) \Rightarrow q' \in \alpha_{s'}(p') \ .$$

4.1. Properties of $\lesssim$

The main result of this section is a standard form for $\lesssim$ in Lemma 4.11. It plays an important role later in showing that the linear function space respects finites and pullbacks.

Definition 4.7 Let spq and $s'p'q'$ be IO-tuples for $\mathbb{P}$, $\mathbb{Q}$. Define $s'p'q' \xrightarrow{f}_{MB} spq$ iff $f : s' \to s$ in $\mathcal{I}$ and

$$p' = \min(\mathbb{P}f, p) \ \& \ \ q' = \min(\mathbb{Q}f, q) \ .$$

Define $s'p'q' \lesssim_{MB} spq$ iff $s'p'q' \xrightarrow{f}_{MB} spq$ for some $f : s' \to s$.

MB-arrows compose, and the relation $\lesssim_{MB}$ is a preorder included in $\lesssim$. An arrow $t' \xrightarrow{f}_{MB} t$ factors as $t' \xrightarrow{f}_B t_0 \lesssim_M t$, for some t_0; as the following proposition makes precise, t_0 is $\lesssim_M$-maximum from which such a B-arrow is possible.

Proposition 4.8 *If $t_1 \lesssim_M t$ and $t'_1 \xrightarrow{f}_B t_1$, then there exists a unique t' such that $t'_1 \lesssim_M t'$ and $t' \xrightarrow{f}_{MB} t$:*

$$\begin{tikzcd}
t' \arrow[r, dotted, "f", "MB"'] \arrow[d, dotted, no head, "\lesssim_M" description] & t \arrow[d, no head, "\lesssim_M" description] \\
t'_1 \arrow[r, "f", "B"'] & t_1
\end{tikzcd}$$

Proof. Suppose $sp_1q \lesssim_M spq$ and $s'p'_1q' \xrightarrow{f}_B sp_1q$. Then $p \le p_1 = \mathbb{P}f(p'_1)$ and $q' = \min(\mathbb{Q}f, q)$. Hence $p' = \min(\mathbb{P}f, p)$ is defined, so that $s'p'_1q' \lesssim_M s'p'q'$ and $s'p'q' \xrightarrow{f}_{MB} spq$. The minimality constraints on tuple t' for which $t' \xrightarrow{f}_{MB} spq$ ensures the uniqueness of $s'p'q'$. □

Proposition 4.8 gives the sense in which B-moves contributing to the order $\lesssim$ are bounded by canonical MB-moves.

Lemma 4.9 *Let*

$$\begin{tikzcd}
 & s_1 \arrow[dr, "g_1"] & \\
s \arrow[ur, "h_1"] \arrow[dr, "h_2"'] & & s_3 \\
 & s_2 \arrow[ur, "g_2"'] &
\end{tikzcd}$$

be a pullback in $\mathcal{I}$. Assume that $s_1p_1q_1 \in (\mathbb{P} \Rightarrow \mathbb{Q})$ and

$$s_1p_1q_1 \xrightarrow{g_1}_F s_3p_3q_3 \ \textit{ and } \ s_2p_2q_2 \xrightarrow{g_2}_{MB} s_3p_3q_3 \ .$$

Then, there is $spq \in (\mathbb{P} \Rightarrow \mathbb{Q})$ such that $spq \xrightarrow{h_1}_F s_1p_1q_1$ and $spq \xrightarrow{h_2}_{MB} s_2p_2q_2$:

$$\begin{tikzcd}
 & s_1p_1q_1 \arrow[dr, "g_1", "F"'] & \\
spq \arrow[ur, dotted, "h_1", "MB"'] \arrow[dr, dotted, "h_2"', "F"] & & s_3p_3q_3 \\
 & s_2p_2q_2 \arrow[ur, "g_2"', "MB"] &
\end{tikzcd}$$

Proof. The assumptions yield

$$\mathbb{P}g_1(p_1) = p_3 \text{ and } p_2 = \min(\mathbb{P}g_2, p_3) \ .$$

As $\mathbb{P}$ respects pullbacks, there is $p \in \mathbb{P}s$ such that

$$\mathbb{P}h_2(p) = p_2 \text{ and } p = \min(\mathbb{P}h_1, p_1) \ .$$

Now the tuple $s_1p_1q_1$ is feasible. Hence there exists $q \in \mathbb{Q}s$ such that

$$q = \min(\mathbb{Q}h_1, q_1) \ .$$

It follows that $spq \xrightarrow{h_1}_{MB} s_1p_1q_1$. In addition, $q_2 = \min(\mathbb{Q}g_2, q_3)$ while $q_3 \in (\mathbb{Q}g_1h_1)(q) = \mathbb{Q}g_2(\mathbb{Q}h_2(q))$, which implies that $q_2 \in \mathbb{Q}h_2(q)$. Thus $spq \xrightarrow{h_2}_F s_2p_2q_2$. □

Proposition 4.10 *If $t_1 \lesssim_M t$ and $t_1 \xrightarrow{f}_F t'_1$, then there exists a unique t' such that $t'_1 \lesssim_M t'$ and $t \xrightarrow{f}_F t'$:*

$$\begin{tikzcd}
t \arrow[r, dotted, "f", "F"'] \arrow[d, dash, "\lesssim_M" description] & t' \arrow[d, dotted, dash, "\lesssim_M" description] \\
t_1 \arrow[r, "f", "F"'] & t'_1
\end{tikzcd}$$

Proof. Suppose $sp_1q \lesssim_M spq$ and $sp_1q \xrightarrow{f}_F s'p'_1q'$. Then $p \le p_1$ and $p'_1 = \mathbb{P}f(p_1)$ and $q' \in \mathbb{Q}f(q)$. Take $p' = \mathbb{P}f(p)$. By the monotonicity of $\mathbb{P}f$, we obtain $s'p'_1q' \lesssim_M s'p'q'$, while $spq \xrightarrow{f}_F s'p'q'$. These two relations determine $s'p'q'$ uniquely. □

The next lemma gives an important standard form in which the order $\lesssim$ can be obtained.

Lemma 4.11 *Assume tuples $t, t' \in (\mathbb{P} \Rightarrow \mathbb{Q})$. Then, $t' \lesssim t$ in $(\mathbb{P} \Rightarrow \mathbb{Q})$ iff there are arrows g, f in $\mathcal{I}$ with a common source and tuples $t_1, t_2 \in (\mathbb{P} \Rightarrow \mathbb{Q})$ such that $t_1 \xrightarrow{g}_{MB} t$ & $t_1 \xrightarrow{f}_F t_2$ & $t' \lesssim_M t_2$:*

$$\begin{tikzcd}
t_1 \arrow[rr, "f" description, "F"'] \arrow[dr, "g" description, "MB"'] & & t_2 \arrow[d, dash, "\lesssim_M" description] \\
& t \arrow[r, dotted, dash, "\gtrsim" description] & t'
\end{tikzcd}$$

Proof. We show the property illustrated above by induction along the length of the chain of $\lesssim_F$, $\lesssim_B$ and $\lesssim_M$ links establishing $t' \lesssim t$. The addition of a link $t'' \lesssim_M t'$ clearly maintains the property. The addition of a link $t'' \lesssim_F t'$ maintains the property by Proposition 4.10. If the additional link is $t'' \lesssim_B t'$, so $t'' \xrightarrow{h}_B t'$ for some h, we first use Proposition 4.8 to obtain

$$\begin{tikzcd}
t_1 \arrow[rr, "f" description, "F"'] \arrow[dr, "g" description, "MB"'] & & t_2 \arrow[d, dash, "\lesssim_M" description] & t_3 \arrow[l, "h"', "MB"] \arrow[d, dash, "\lesssim_M" description] \\
& t \arrow[r, dotted, dash, "\gtrsim" description] & t' & t'' \arrow[l, "h"', "B"]
\end{tikzcd}$$

whereupon we can use Lemma 4.9 w.r.t. $t_1 \xrightarrow{f}_F t_2$ and $t_3 \xrightarrow{h}_{MB} t_2$ to obtain the desired property for t''. □

Definition 4.12 *Let $U \subseteq (\mathbb{P} \Rightarrow \mathbb{Q})$. Say $U_0 \subseteq (\mathbb{P} \Rightarrow \mathbb{Q})$ FM-*generates *U iff*

$$U = \{t' \in (\mathbb{P} \Rightarrow \mathbb{Q}) \mid \exists t \in U_0, t''.\, t' \lesssim_M t'' \lesssim_F t\} \ .$$

Corollary 4.13 *Let $t \in (\mathbb{P} \Rightarrow \mathbb{Q})$. There is a finite subset of $(\mathbb{P} \Rightarrow \mathbb{Q})$ which FM-generates $\downarrow t$.*

For a name set s', the contribution of t at s',

$$\mathrm{C}ontrib(t, s') =_{def} \{(p', q') \mid s'p'q' \lesssim t\}$$

is a finite element of the domain $\widehat{(\mathbb{P}s')^{\mathrm{op}} \times \mathbb{Q}s'}$.

Proof. Suppose $t = spq$. By Lemma 4.11, if $s'p'q' \lesssim t$, then

$$\begin{tikzcd}
s_1p_1q_1 \arrow[rr, "f" description, "F"'] \arrow[dr, "g" description, "MB"'] & & s'p_2q_2 \arrow[d, dash, "\lesssim_M" description] \\
& t \arrow[r, dotted, dash, "\gtrsim" description] & s'p'q'
\end{tikzcd}$$

for suitable tuples and maps $f : s_1 \to s'$, $g : s_1 \to s$ in $\mathcal{I}$ with a common source. For each n, with $0 \le n \le |s|$, choose $\bar{n}$ a name set of size n. For each name set s_1, with $|s_1| \le |s|$, fix a choice of isomorphism $\overline{|s_1|} \cong s_1$. In the diagram above, the MB-F-span of arrows $s_1p_1q_1 \xrightarrow{g}_{MB} t$ and $s_1p_1q_1 \xrightarrow{f}_F s'p_2q_2$ can be replaced by a span $\bar{n}\bar{p}_1\bar{q}_1 \xrightarrow{\bar{g}}_{MB} t$ and $\bar{n}\bar{p}_1\bar{q}_1 \xrightarrow{\bar{f}}_F s'p_2q_2$ which relocates the source of g, f to a chosen name set $\bar{n}$ via the choice of isomorphism $\bar{n} \cong s_1$:

$$\begin{tikzcd}
\bar{n}\bar{p}_1\bar{q}_1 \arrow[rr, "\bar{f}" description, "F"'] \arrow[dr, "\bar{g}" description, "MB"'] & & s'p_2q_2 \arrow[d, dash, "\lesssim_M" description] \\
& t \arrow[r, dotted, dash, "\gtrsim" description] & s'p'q'
\end{tikzcd}$$

Define

$$U_0 = \{\bar{n}p_1q_1 \mid 0 \le n \le |s| \ \&\ \bar{n}p_1q_1 \lesssim_{MB} t\} \ .$$

This is a finite set which clearly FM-generates $\downarrow t$.

Because $\downarrow t$ is $\lesssim_F$- and $\lesssim_M$-down-closed, $\mathrm{C}ontrib(t, s')$ is a down-closed subset of $(\mathbb{P}s')^{\mathrm{op}} \times \mathbb{Q}s'$. It is generated by the set

$$\{(p_2, q_2) \mid \exists \bar{f},\ \bar{n}p_1q_1 \in U_0.\ \bar{n}p_1q_1 \xrightarrow{\bar{f}}_F s'p_2q_2\} \ .$$

Each $\mathbb{Q}\bar{f}(q_1)$ is a finite element of $\widehat{\mathbb{Q}s'}$. This, together with U_0 being finite, ensures that $\mathrm{C}ontrib(t, s')$ is a finite element. □

5. Linear function space

Let $\mathbb{P}$ and $\mathbb{Q}$ be type functors where $\mathbb{P}$ also respects primes. Let s_0 be a name set. By Proposition 3.3, $\mathcal{I}(s_0, -) \otimes \mathbb{P}$ is a type functor which respects primes. Hence we can define the path order

$$(\mathbb{P} \multimap \mathbb{Q})s_0 = (\mathcal{I}(s_0, -) \otimes \mathbb{P} \Rightarrow \mathbb{Q}) \ ,$$

and inherit many of its properties from the previous sections.

This section is devoted to the functorial action of the function space, to the definition and properties of

$$(\mathbb{P} \multimap \mathbb{Q})g : (\mathbb{P} \multimap \mathbb{Q})s_0 \to (\mathbb{P} \multimap \mathbb{Q})s_1$$

an arrow in $\mathbf{Lin}$, for $g : s_0 \to s_1$ an arrow in $\mathcal{I}$. Recall that

$$\widehat{(\mathbb{P} \multimap \mathbb{Q})}(s_0) \cong [\mathcal{I}, \mathbf{Lin}](\mathcal{I}(s_0, -) \times \mathbb{P}, \mathbb{Q})$$

which determines the functorial action. Regarded as a map on natural transformations, $(\mathbb{P} \multimap \mathbb{Q})g$ takes

$$\alpha^0 : \mathcal{I}(s_0, -) \times \mathbb{P} \Rightarrow \mathbb{Q}$$

to

$$\alpha^1 : \mathcal{I}(s_1, -) \times \mathbb{P} \Rightarrow \mathbb{Q}$$

with components at $s \in \mathcal{I}$ given by the commuting triangle

$$\begin{tikzcd}
\mathcal{I}(s_1, s) \times \mathbb{P}s \arrow[r, "\alpha^1_s"] \arrow[d, "(-\circ g)\times \mathrm{id}"'] & \mathbb{Q}s \, . \\
\mathcal{I}(s_0, s) \times \mathbb{P}s \arrow[ur, "\alpha^0_s"'] &
\end{tikzcd} \qquad (\dagger)$$

It is a routine matter to show that the map $\alpha^0 \mapsto \alpha^1$ preserves joins, but to complete the proof that $\mathbb{P} \multimap \mathbb{Q}$ is itself a type functor we require a more careful analysis of its functorial action.

The functorial action of $\mathbb{P} \multimap \mathbb{Q}$ described above on natural transformations fixes the action of $(\mathbb{P} \multimap \mathbb{Q})g$ on path sets. Recall the isomorphism $\mathcal{G}$ between natural transformations and path sets of Theorem 4.5. From the commuting triangle $(\dagger)$ we observe that

$$si_1pq \in \mathcal{G}(\alpha^1) \text{ iff } si_0pq \in \mathcal{G}(\alpha^0) \;\&\; i_0 = i_1 \circ g \, ,$$

which is simply a rephrasing of the fact that

$$q \in \alpha^1_s(i_1, p) \text{ iff } q \in \alpha^0_s(i_0, p) \;\&\; i_0 = i_1 \circ g \, .$$

So for $g : s_0 \to s_1$ an arrow in $\mathcal{I}$, we define

$$(\mathbb{P} \multimap \mathbb{Q})g(U) = \{ si_1pq \mid i_1 : s_1 \to s \;\&\; s(i_1g)pq \in U \} \, ,$$

for any $U \in \widehat{(\mathbb{P} \multimap \mathbb{Q})}s_0$. Note that if $s(i_1g)pq \in (\mathbb{P} \multimap \mathbb{Q})s_0$, then $si_1pq \in (\mathbb{P} \multimap \mathbb{Q})s_1$. This is because the feasibility of si_1pq at s_1 follows from the feasibility of $s(i_1g)pq$ at s_1. This is seen by considering the commuting triangle $(\dagger)$ above defining the action of $(\mathbb{P} \multimap \mathbb{Q})g$ on natural transformations, taking a natural transformation α^0 to a natural transformation α^1. Clearly if $q \in \alpha^1_s(i_1g, p)$, then $q \in \alpha^0_s(i_1, p)$.

We can understand the effect of $(\mathbb{P} \multimap \mathbb{Q})g$ on a path set U as an inverse image with respect to a partial function Fg:

$$(\mathbb{P} \multimap \mathbb{Q})g(U) = (Fg)^{-1}U$$

where

$$Fg : (\mathbb{P} \multimap \mathbb{Q})s_1 \rightharpoonup (\mathbb{P} \multimap \mathbb{Q})s_0$$

is the partial function taking si_1pq to $s(i_1g)pq$ provided $s(i_1g)pq \in (\mathbb{P} \multimap \mathbb{Q})s_0$, and undefined otherwise—in the case where $s(i_1g)pq$ is not feasible at s_0. This has two immediate consequences. The first that $\mathbb{P} \multimap \mathbb{Q}$ respects nonempty meets is essential for being a type functor. The second is a useful characterisation of the $\min$-partial function for maps $(\mathbb{P} \multimap \mathbb{Q})g$.

Proposition 5.1 *For g an arrrow in $\mathcal{I}$, the function $(\mathbb{P} \multimap \mathbb{Q})g$ preserves non-empty meets.*

Proof. From the general fact that inverse image of a partial function preserves nonempty intersections. □

Proposition 5.2 *Let g be an arrrow in $\mathcal{I}$. If $\min((\mathbb{P} \multimap \mathbb{Q})g, si_1pq)$ is defined, then*

$$\min((\mathbb{P} \multimap \mathbb{Q})g, si_1pq) = {\downarrow} s(i_1g)pq \, .$$

Proof. Clear from the definition of $(\mathbb{P} \multimap \mathbb{Q})g$. □

Given an arrow $g : s_0 \to s_1$ in $\mathcal{I}$, an order relation, whether it be $\lesssim_B, \lesssim_F, \lesssim_M$ or $\lesssim$, in $(\mathbb{P} \multimap \mathbb{Q})s_1$ projects under Fg to the corresponding order relation in $(\mathbb{P} \multimap \mathbb{Q})s_0$. The order $\lesssim$ in $(\mathbb{P} \multimap \mathbb{Q})s_0$ is generally more refined than the image of that in $(\mathbb{P} \multimap \mathbb{Q})s_1$.

6. Function space is a type functor

Let $\mathbb{P}$ and $\mathbb{Q}$ be type functors where $\mathbb{P}$ respects primes. The main result of this section is:

Theorem 6.1 *The functor $\mathbb{P} \multimap \mathbb{Q}$ is a type functor.*

We have already seen that $\mathbb{P} \multimap \mathbb{Q}$ respects meets (Proposition 5.1), so to prove this theorem it remains to show that finites and pullbacks are respected.

6.1. Function space respects finites

In ‘bounding’ the images of finite elements we’ll make use of Leifer and Milner’s relative pushouts [7]:

Proposition 6.2 *A* relative pushout (rpo) *for arrows g_1, g_2 relative to f_1, f_2, where $f_1 \circ g_1 = f_2 \circ g_2$, comprises arrows h_1, h_2, h such that $h_1 \circ g_1 = h_2 \circ g_2$ and $f_1 = h \circ h_1$ and $f_2 = h \circ h_2$*

$$\begin{tikzcd}
 & s_2 \arrow[dr, dotted, "h_2"] \arrow[rr, bend left=10, "f_2"] & & \\
s_0 \arrow[ur, "g_2"] \arrow[dr, "g_1"'] & & s'' \arrow[r, dotted, "h"] & s' \, , \\
 & s_1 \arrow[ur, dotted, "h_1"] \arrow[rru, "f_1"'] & &
\end{tikzcd}$$

with the universal property that for any arrows h_1', h_2', h' such that $h_1' \circ g_1 = h_2' \circ g_2$ and $f_1 = h' \circ h_1'$ and $f_2 =$

$h' \circ h'_2$ there is a unique arrow u such that $h = h' \circ u$ and $h'_1 = u \circ h_1$ and $h'_2 = u \circ h_2$.

Above, h_1, h_2, id is itself an rpo for the commuting square g_1, g_2 relative to h_1, h_2 called an idem pushout *(ipo).*

Any commuting square in $\mathcal{I}$ has an rpo. Any pair of arrows g_1, g_2 in $\mathcal{I}$ with a common source have only finitely many ipos to within isomorphism.

Proof. The basic facts about rpos can be found in [7]. In particular, by universality, an rpo is determined to within isomorphism. The existence of rpos in $\mathcal{I}$ for g_1, g_2 relative to f_1, f_2, as in the diagram above, follows by *e.g.* taking s'' to be the union of the images of s_1 and s_2 in s', and h to be the inclusion map. Clearly, there are only finitely many ways in which images of s_1 and s_2 can overlap when varying f_1, f_2. It follows that there are only finitely many ipos for g_1, g_2, to within isomorphism. □

Lemma 6.3 *Let $g : s_0 \to s_1$. Assume*

$$si_0pq \xrightarrow{f}_F s'(fi_0)p'q' \textit{ in } (\mathbb{P} \multimap \mathbb{Q})s_0 \textit{ and}$$
$$s'i'_1p'q' \in (\mathbb{P} \multimap \mathbb{Q})g(s'(fi_0)p'q') \,.$$

As $i'_1 g = fi_0$, there are arrows $g' : s \to s''$, $i_1 : s_1 \to s''$ and $f' : s'' \to s'$ forming an rpo of i_0, g relative to f, i'_1 in $\mathcal{I}$. Then $\min(\mathbb{Q}f', q')$ is defined, and writing $p'' = \mathbb{P}g'(p)$ and $q'' = \min(\mathbb{Q}f', q')$,

$$si_0pq \xrightarrow{g'}_F s''(i_1g)p''q'' \textit{ in } (\mathbb{P} \multimap \mathbb{Q})s_0 \textit{ and}$$
$$s''i_1p''q'' \xrightarrow{f'}_F s'i'_1p'q' \textit{ in } (\mathbb{P} \multimap \mathbb{Q})s_1 \;:$$

$$\begin{tikzcd}
si_0pq \arrow[rrr, "f", "F"'] \arrow[dr, dotted, "g'", "F"'] & & & s'(fi_0)p'q' \arrow[d, dash, "(\mathbb{P}\multimap\mathbb{Q})g"] \\
& s''(i_1g)p''q'' \arrow[urr, dotted, "f'", "F"'] \arrow[d, dotted, dash, "(\mathbb{P}\multimap\mathbb{Q})g"'] & & s'i'_1p'q' \\
& s''i_1p''q'' \arrow[urr, dotted, "f'", "F"'] & &
\end{tikzcd}$$

Proof. The initial assumption of the lemma entails that the following diagram in $\mathcal{I}$ commutes

$$\begin{tikzcd}
s_0 \arrow[r, "i_0"] \arrow[d, "g"'] & s \arrow[r, "f"] & s' \\
s_1 \arrow[urr, "i'_1"'] & &
\end{tikzcd}$$

as well as that $p \in \mathbb{P}s$, $q \in \mathbb{Q}s$, $p' \in \mathbb{P}s'$, $q' \in \mathbb{Q}s'$ with $p' = \mathbb{P}f(p)$ and $q' \in \mathbb{Q}f(q)$.

Forming the rpo we obtain the diagram

$$\begin{tikzcd}
s_0 \arrow[r, "i_0"] \arrow[d, "g"'] & s \arrow[r, "f"] \arrow[d, "g'"'] & s' \\
s_1 \arrow[r, "i_1"'] \arrow[urr, "i'_1"] & s'' \arrow[ur, "f'"'] &
\end{tikzcd}$$

in which all subdiagrams commute.

We write $p'' = \mathbb{P}g'(p)$ and $q'' = \min(\mathbb{Q}f', q')$—the latter is defined as $q' \in \mathbb{Q}f(q)$ and $f'g' = f$. Then $p' = \mathbb{P}f'(p'')$, again as $f'g' = f$. So we get

$$si_0pq \xrightarrow{g'}_F s''(i_1g)p''q'' \text{ in } (\mathbb{P} \multimap \mathbb{Q})s_0 \,,$$

using $g'i_0 = i_1g$ and the minimality property of q'', and

$$s''i_1p''q'' \xrightarrow{f'}_F s'i'_1p'q' \text{ in } (\mathbb{P} \multimap \mathbb{Q})s_1 \,,$$

using $i'_1 = f'i_1$ and $q' \in \mathbb{Q}f'(q'')$. □

Corollary 6.4 *The functor $\mathbb{P} \multimap \mathbb{Q}$ respects finites.*

Proof. Let $g : s_0 \to s_1$ in $\mathcal{I}$. Given a prime element in the domain $(\mathbb{P} \multimap \mathbb{Q})s_0$, Lemma 6.3 allows us to produce a finite set which generates the prime's image under $(\mathbb{P} \multimap \mathbb{Q})g$. Let $\downarrow t$ be a prime in $(\mathbb{P} \multimap \mathbb{Q})s_0$. By Corollary 4.13, there is a finite subset U_0 of $(\mathbb{P} \multimap \mathbb{Q})s_0$ which FM-generates $\downarrow t$. The tuples in U_0 have the form $sipq$ and involve only finitely many name sets s, so bounded by a maximum size N. For each such name set s to within isomorphism there are finitely many ipo's

$$\begin{tikzcd}
s_0 \arrow[r, "i_0"] \arrow[d, "g"'] & s \arrow[d, "g'"] \\
s_1 \arrow[r, "i_1"'] & s'' \,.
\end{tikzcd}$$

In such an ipo the name set s'' can have size at most $N + |s_1|$. Let $\bar{n}$ be a choice of name set for each $n \leq N + |s_1|$.

Recall from Corollary 4.13, that $\mathrm{C}\mathit{ontrib}(t, s'')$ is a finite element of $(\widehat{\mathcal{I}(s_0, s'') \times \mathbb{P}s''})^{\mathrm{op}} \times \mathbb{Q}s''$. So there is a finite subset $\mathrm{C}\mathit{ontrib}_0(t, s'')$ generating $\mathrm{C}\mathit{ontrib}(t, s'')$—*i.e.* of which the downclosure in $(\mathcal{I}(s_0, s'') \times \mathbb{P}s'')^{\mathrm{op}} \times \mathbb{Q}s''$ is $\mathrm{C}\mathit{ontrib}(t, s'')$. Define

$$\mathit{Gen} = \bigcup_{n \leq N + |s_1|} \{\bar{n}i''p''q'' \mid i''p''q'' \in \mathrm{C}\mathit{ontrib}_0(t, \bar{n})\} \,,$$

which is clearly a finite set.

By Lemma 6.3, the image $(\mathbb{P} \multimap \mathbb{Q})g(t)$ is FM-generated by the set $(Fg)^{-1}\mathit{Gen}$. Recall the partial function Fg such that $(\mathbb{P} \multimap \mathbb{Q})g = (Fg)^{-1}$; when defined $(Fg)(s''i_1p''q'') = s''(i_1g)p''q''$—see Section 5. Consequently $((Fg)^{-1}\mathit{Gen}$ is finite and generates $(\mathbb{P} \multimap \mathbb{Q})g(t)$, which is therefore a finite element of $\widehat{(\mathbb{P} \multimap \mathbb{Q})}s_1$. □

6.2. Function space respects pullbacks

Lemma 6.5 *The functor $\mathbb{P} \multimap \mathbb{Q}$ respects pullbacks.*

Proof. Assume a pullback square

$$\begin{tikzcd}
 & s_1 \arrow[dr, "g_1"] & \\
s_0 \arrow[ur, "h_1"] \arrow[dr, "h_2"'] & & s_3 \\
 & s_2 \arrow[ur, "g_2"'] &
\end{tikzcd}$$

in $\mathcal{I}$. Suppose $s_1' i_1 pq \in (\mathbb{P} \multimap \mathbb{Q})s_1$, so $i_1 : s_1 \to s_1'$ in $\mathcal{I}$, has the following property: letting $x_1 = \downarrow s_1' i_1 pq$, a prime element of $\widehat{(\mathbb{P} \multimap \mathbb{Q})}s_1$, and $x_3 = ((\mathbb{P} \multimap \mathbb{Q})g_1)(x_1)$ there is $x_2 = \min((\mathbb{P} \multimap \mathbb{Q})g_2, x_3) \in \widehat{(\mathbb{P} \multimap \mathbb{Q})}s_2$. We shall show that $x_0 = \downarrow s_1'(i_1 h_1)pq$ is an element of $\widehat{(\mathbb{P} \multimap \mathbb{Q})}s_0$ which meets the requirements for the function space to respect pullbacks, *viz.*

$$\begin{aligned}
&(1)\ \ x_0 = \min((\mathbb{P} \multimap \mathbb{Q})h_1, x_1) \quad \text{and} \\
&(2)\ \ ((\mathbb{P} \multimap \mathbb{Q})h_2)(x_0) = x_2 \ .
\end{aligned}$$

(1) The existence of x_0 presupposes that $s_1'(i_1 h_1)pq \in (\mathbb{P} \multimap \mathbb{Q})s_0$. So we must check that $s_1'(i_1 h_1)pq$ is feasible. Once $s_1'(i_1 h_1)pq$ is known to be feasible, it will follow directly from Proposition 5.2 that x_0 is the minimum input yielding x_1.

In order to show that $s_1'(i_1 h_1)pq$ is feasible, suppose that arrows $i_0 : s_0 \to s_0'$ and $h_1' : s_0' \to s_1'$ in $\mathcal{I}$ and $p' \in \mathbb{P}s_0'$ satisfy

$$i_1 h_1 = h_1' i_0 \ \&\ p \leq (\mathbb{P}h_1')p' \ .$$

We need to exhibit $q_0 \in \mathbb{Q}s_0'$ such that $q \in (\mathbb{Q}h_1')q_0$.

The arrows introduced so far form a diagram in $\mathcal{I}$, consisting of the solid arrows below, in which the two solid squares commute:

$$\begin{tikzcd}
 & s_1 \arrow[r, "i_1"] \arrow[dr, "g_1"] & s_1' \arrow[dr, dotted, "g_1'"] & \\
s_0 \arrow[ur, "h_1"] \arrow[r, "i_0"'] \arrow[dr, "h_2"'] & s_0' \arrow[ur, "h_1'"] \arrow[dr, dotted, "h_2'"'] & s_3 \arrow[r, dotted, "i_3"'] & s_3' \\
 & s_2 \arrow[ur, "g_2"] \arrow[r, dotted, "i_2"'] & s_2' \arrow[ur, dotted, "g_2'"'] &
\end{tikzcd}$$

Taking advantage of the special nature of $\mathcal{I}$, we have completed the diagram (the dotted arrows) to one in which all squares commute and s_0', h_1', h_2' is a pullback.

Let q_3 be an arbitrary path in the image $\mathbb{Q}g_1'(q)$. Let $p_3 = (\mathbb{P}g_1')(p)$. As

$$s_3'(i_3 g_1)p_3 q_3 = s_3'(g_1' i_1)p_3 q_3 \lesssim_F s_1' i_1 pq \ ,$$

the image of x_1 under $(\mathbb{P} \multimap \mathbb{Q})g_1$ contains $s_3' i_3 p_3 q_3$. Thus $s_3' i_3 p_3 q_3 \in x_3$. By assumption, $\min((\mathbb{P} \multimap \mathbb{Q})g_2, x_3)$ exists. Hence $\min((\mathbb{P} \multimap \mathbb{Q})g_2, s_3' i_3 p_3 q_3)$ exists, and has to be the prime generated by $s_3'(i_3 g_2)p_3 q_3 \in (\mathbb{P} \multimap \mathbb{Q})s_2$. Now we use the feasibility of the tuple $s_3'(i_3 g_2)p_3 q_3$: letting $p_2 = (\mathbb{P}h_2')(p')$ in $\mathbb{P}s_2'$ and observing that $g_2' i_2 = i_3 g_2$, there must exist $q_2 = \min(\mathbb{Q}g_2', q_3)$.

But q_3 was an arbitrary path in the image $\mathbb{Q}g_1'(q)$, so $\min(\mathbb{Q}g_2', \mathbb{Q}g_1'(q))$ exists. Now as $\mathbb{Q}$ respects pullbacks, there must exist q_0 such that in particular $q_0 = \min(\mathbb{Q}h_1', q)$. *A fortiori* $q \in (\mathbb{Q}h_1')q_0$.

(2) It suffices to show that $((\mathbb{P} \multimap \mathbb{Q})h_2)(x_0) \subseteq x_2$.

We first characterise certain tuples $s_2' i_2 p_2 q_2 \in ((\mathbb{P} \multimap \mathbb{Q})h_2)(x_0)$ sufficient to generate $((\mathbb{P} \multimap \mathbb{Q})h_2)(x_0)$.

Recall that $x_0 = \downarrow s_1'(i_1 h_1)pq$. By Lemma 4.11, any tuple in x_0 is $\lesssim_M$-below some $s_2'(h_2' i_0)p_2 q_2$ such that

$$\begin{aligned}
&s_0' i_0 p_0 q_0 \xrightarrow{h_2'}_F s_2'(h_2' i_0)p_2 q_2 \ \ \text{and} \\
&s_0' i_0 p_0 q_0 \xrightarrow{h_1'}_{MB} s_1'(i_1 h_1)pq \ \text{in } (\mathbb{P} \multimap \mathbb{Q})s_0
\end{aligned}$$

for suitable tuples and arrows $h_1' : s_0' \to s_1'$, $h_2' : s_0' \to s_2'$ in $\mathcal{I}$. Consequently images of such tuples $s_2'(h_2' i_0)p_2 q_2$ under $(\mathbb{P} \multimap \mathbb{Q})h_2$ generate $((\mathbb{P} \multimap \mathbb{Q})h_2)(x_0)$. If $i_2 : s_2 \to s_2'$ is an arrow in $\mathcal{I}$ for which $i_2 h_2 = h_2' i_0$, then $s_2' i_2 p_2 q_2 \in ((\mathbb{P} \multimap \mathbb{Q})h_2)(s_2'(h_2' i_0)p_2 q_2)$. The image $((\mathbb{P} \multimap \mathbb{Q})h_2)(x_0)$ is generated by tuples $s_2' i_2 p_2 q_2$ obtained in such a way. Without loss of generality we can assume that q_2 is maximal in the finite element $\mathbb{Q}h_2'(q_0)$.

Thus it is suffices to show that every such tuple $s_2' i_2 p_2 q_2$, obtained as above with q_2 maximal in $\mathbb{Q}h_2'(q_0)$, is in x_2.

It is helpful to summarise the name sets and maps as the solid arrows in the following diagram in $\mathcal{I}$:

$$\begin{tikzcd}
 & s_1 \arrow[r, "i_1"] \arrow[dr, "g_1"] & s_1' \arrow[dr, dotted, "g_1'"] & \\
s_0 \arrow[ur, "h_1"] \arrow[r, "i_0"'] \arrow[dr, "h_2"'] & s_0' \arrow[ur, "h_1'"] \arrow[dr, "h_2'"'] & s_3 \arrow[r, dotted, "i_3"'] & s_3' \\
 & s_2 \arrow[ur, "g_2"] \arrow[r, "i_2"'] & s_2' \arrow[ur, dotted, "g_2'"'] &
\end{tikzcd}$$

We have filled out the diagram to one where all squares commute and s_0', h_1', h_2' is a pullback.

Define $p' = \mathbb{P}h_1'(p_0)$, so $p \leq p'$, and $p_3 = \mathbb{P}g_1'(p')$, so $p_3 = \mathbb{P}g_2'(p_2)$.

We first show the existence of $q_3 \in \mathbb{Q}g_1'(q)$ such that

$$q_2 = \min(\mathbb{Q}g_2', q_3) \ .$$

As $\mathbb{Q}$ respects pullbacks,

$$\mathbb{Q}h'_2(q_0) = \min(\mathbb{Q}g'_2, \mathbb{Q}g'_1(q)) \ .$$

It follows that there exists $q_3 \in \mathbb{Q}g'_1(q)$ such that

$$q_2 \leq \min(\mathbb{Q}g'_2, q_3) \ .$$

But q_2 is maximal in $\mathbb{Q}h'_2(q_0)$. So the converse order also holds, yielding the required equality $q_2 = \min(\mathbb{Q}g'_2, q_3)$.

Now observe that

$$s'_3(i_3 g_1)p_3 q_3 = s'_3(g'_1 i_1)p_3 q_3 \lesssim_F s'_1 i_1 p' q \in x_1 \ .$$

Thus

$$s'_3 i_3 p_3 q_3 \in x_3 \ ,$$

the image of x_1 under $(P \multimap \mathbb{Q})g_1$.

By assumption $\min((\mathbb{P} \multimap \mathbb{Q})g_2, x_3) = x_2$. So $\min((\mathbb{P} \multimap \mathbb{Q})g_2, s'_3 i_3 p_3 q_3)$ must be defined and equal to $s'_3(i_3 g_2)p_3 q_3$. Therefore $s'_3(i_3 g_2)p_3 q_3 \in x_2$. But

$$s'_2 i_2 p_2 q_2 \lesssim_B s'_3(g'_2 i_2)p_3 q_3 = s'_3(i_3 g_2)p_3 q_3 \ ,$$

from which we obtain $s'_2 i_2 p_2 q_2 \in x_2$, as required. □

6.3. The defining adjunction

To conclude it only remains to check the required adjunction for the linear function space:

Theorem 6.6 *Let* $\mathbb{P}$ *be a type functor which respects primes. Then,*

$$[\mathcal{I}, \mathbf{Lin}](\mathbb{R}, \mathbb{P} \multimap \mathbb{Q}) \cong [\mathcal{I}, \mathbf{Lin}](\mathbb{R} \otimes \mathbb{P}, \mathbb{Q}) \ ,$$

natural in type functors $\mathbb{R}$ *and* $\mathbb{Q}$.

Proof. We can now identify path sets in $(\mathbb{P} \widehat{\multimap \mathbb{Q}})s$ with natural transformations from $(\mathcal{I}(s, -) \otimes \mathbb{P}$ to $\mathbb{Q}$. The isomorphism θ takes $\alpha : \mathbb{R} \Rightarrow (\mathbb{P} \multimap \mathbb{Q})$ to $\theta\alpha : \mathbb{R} \otimes \mathbb{P} \Rightarrow \mathbb{Q}$ such that $(\theta\alpha)_s(r, p) = (\alpha_s r)_s(\mathrm{id}_s, p)$, for all $r \in \mathbb{R}s$ and $p \in \mathbb{P}s$. Its inverse φ, taking $\beta : \mathbb{R} \otimes \mathbb{P} \Rightarrow \mathbb{Q}$ to $\phi\beta : \mathbb{R} \Rightarrow (\mathbb{P} \multimap \mathbb{Q})$, is defined to satisfy $((\varphi\beta)_{s_0} r)_s(f, p) = \beta_s(((\mathbb{R}f)r) \otimes p)$, for all $r \in \mathbb{R}s_0$, $f : s_0 \to s$ in $\mathcal{I}$ and $p \in \mathbb{P}s$. These can now be checked to give an isomorphism natural in type functors $\mathbb{R}$ and $\mathbb{Q}$. □

Concluding remarks

The explicit description of paths of the function space paves the way for proofs of adequacy and full abstraction for higher-order process languages like new-HOPLA, paralleling the proofs for HOPLA [9]. The construction of the path orders for function space should guide us towards analogous constructions for analogues of $\mathbf{Lin}$ such as the bicategory of profunctors, or the biequivalent 2-category $\mathbf{Cont}$, which has colimit-preserving functors between presheaf categories as arrows [2]—this would solve the limitation acknowledged in [1]. While it is not likely to help proofs of adequacy and full abstraction, one could hope for a more conceptually satisfying method to adjoin name generation to a domain theory, *e.g.* by defining the category of domains internally within some topos like nominal sets which copes with name generation in a global fashion [4]. The 'obvious' approach of defining $\mathbf{Lin}$ internally in nominal sets only yields prime respecting type functors however, and *e.g.* $\mathbb{N} \multimap \mathbb{Q}$ from Example 2.1 does not respect primes.

Acknowledgements

Thanks to Marcelo Fiore, Matthew Hennessy, Martin Hyland, Andy Pitts, Sam Staton, Dave Turner, Francesco Zappa Nardelli for advice and encouragement.

References

[1] G. L. Cattani, I. Stark and G. Winskel. Presheaf models for the π-calculus. In *Proc. CTCS'97*, 1997.

[2] G. L. Cattani and G. Winskel. Profunctors, open maps and bisimulation. Accepted by MSCS, 2004. Available from http://www.cl.cam.ac.uk/ gw104/.

[3] M. Fiore, E. Moggi and D. Sangiorge. A fully-abstract model for the π-calculus. LICS 1996.

[4] M.J. Gabbay and A.M. Pitts. A new approach to abstract syntax involving binders. LICS, 1999.

[5] J.-Y. Girard. Linear logic. *Theoretical Computer Science*, 50(1):1–102, 1987.

[6] M. Hennessy. A fully abstract denotational semantics for the pi-calculus. *Tneoretical Computer Science*: 53–89, 2001.

[7] J.J. Leifer and R. Milner. Deriving bisimulation congruences for reactive systems. CONCUR, 2000.

[8] M. Nielsen, G. Plotkin and G. Winskel. Petri nets, event structures and domains, part I. *Theoretical Computer Science*, 13(1):85–108, 1981.

[9] M. Nygaard and G. Winskel. Domain theory for concurrency. *Theoretical Computer Science* 316: 153–190, 2004.

[10] I. Stark. A fully-abstract domain model for the π-calculus. LICS 1996.

[11] G. Winskel and F. Zappa Nardelli. New-HOPLA: a higher order process language with name generation. TCS2004.

Appendix

Proof of Proposition 3.2:
(i) For $f : s \rightarrow s'$ in $\mathcal{I}$, consider the pullback square

$$\begin{tikzcd}
 & s \arrow[dr, "f"] & \\
s \arrow[ur, "\mathrm{id}"] \arrow[dr, "\mathrm{id}"'] & & s' \\
 & s \arrow[ur, "f"'] &
\end{tikzcd}$$

The fact that $\min(\mathbb{Q}f, \mathbb{Q}f(x)) = x$ now follows as a special case of the pullback respecting property (with x_1 set to x).

(ii) This follows as the special case of (i), when $x = {\downarrow}q$ as $\min$ preserves joins in its second argument.

(iii) Let $f : s \rightarrow s'$ in $\mathcal{I}$. Suppose $\mathbb{Q}f(x) \subseteq \mathbb{Q}f(y)$, for $x, y \in \widehat{\mathbb{Q}s}$. Then

$$x = \min(\mathbb{Q}f, \mathbb{Q}f(x)) \subseteq \min(\mathbb{Q}f, \mathbb{Q}f(y)) = y \ ,$$

showing f is an order monic.

(iv) To show type functors preserve pullbacks we first show that with respect to a pullback square

$$\begin{tikzcd}
 & s_1 \arrow[dr, "g_1"] & \\
s_0 \arrow[ur, "h_1"] \arrow[dr, "h_2"'] & & s_3 \\
 & s_2 \arrow[ur, "g_2"'] &
\end{tikzcd}$$

in $\mathcal{I}$, that assuming $\mathbb{Q}g_1(x_1) = \mathbb{Q}g_2(x_2)$, for $x_1 \in \widehat{\mathbb{Q}s_1}$, $x_2 \in \widehat{\mathbb{Q}s_2}$, there then exists a unique $x_0 \in \widehat{\mathbb{Q}s_0}$ such that $x_1 = \mathbb{Q}h_1(x_0)$ and $x_2 = \mathbb{Q}h_2(x_0)$. Write x_3 for $\mathbb{Q}g_1(x_1) = \mathbb{Q}g_2(x_2)$. Then $x_2 = \min(\mathbb{Q}g_2, x_3)$. As $\mathbb{Q}$ respects pullbacks, we deduce the existence of $x_0 \in \widehat{\mathbb{Q}s_0}$ such that $x_0 = \min(\mathbb{Q}h_1, x_1)$ and $x_2 = \mathbb{Q}h_2(x_0)$. Similarly, as $x_1 = \min(\mathbb{Q}g_1, x_3)$, we deduce the existence of $x_0' \in \widehat{\mathbb{Q}s_0}$ such that $x_0' = \min(\mathbb{Q}h_2, x_2)$ and $x_1 = \mathbb{Q}h_1(x_0')$. But by the minimality properties $x_0 \subseteq x_0'$ and $x_0' \subseteq x_0$, so $x_0 = x_0'$. Hence $x_1 = \mathbb{Q}h_1(x_0)$ and $x_2 = \mathbb{Q}h_2(x_0)$, as required.

Assume arrows $l_1 : \mathbb{P} \rightarrow \mathbb{Q}s_1$ and $l_2 : \mathbb{P} \rightarrow \mathbb{Q}s_2$ in **Lin** for which $(\mathbb{Q}g_1) \circ l_1 = (\mathbb{Q}g_2) \circ l_2$. Let $p \in \mathbb{P}$. Then clearly $\mathbb{Q}g_1(l_1(p)) = \mathbb{Q}g_2(l_2(p))$. Thus there exists a unique $x_0 \in \widehat{\mathbb{Q}s_0}$ such that $l_1(p) = \mathbb{Q}h_1(x_0)$ and $l_2(p) = \mathbb{Q}h_2(x_0)$. Thus to each $p \in \mathbb{P}$ we can associate an $x_0 \in \widehat{\mathbb{Q}s_0}$. Moreover, it is easy to check that this association is monotonic in p. Letting $p \leq p'$, analogously we obtain x_0' such that $l_1(p') = \mathbb{Q}h_1(x_0')$ and $l_2(p') = \mathbb{Q}h_2(x_0')$. As $l_1(p) \leq l_1(p')$, this in particular yields $\mathbb{Q}h_1(x_0) \leq \mathbb{Q}h_1(x_0')$, whence $x_0 \subseteq x_0'$, as $\mathbb{Q}h_1$ is an order monic. Thus we have determined a unique arrow $h : \mathbb{P} \rightarrow \mathbb{Q}s_0$ in **Lin** for which $l_1 = (\mathbb{Q}h_1) \circ h$ and $l_2 = (\mathbb{Q}h_2) \circ h$. □

Proof of Proposition 3.3:
For f an arrow in $\mathcal{I}$, the functorial action $\mathcal{I}(s_0, f) = f \circ \_$ is an injection from primes to primes, so preserves primes and nonempty meets. The remaining property required for $\mathcal{I}(s_0, \_)$ to be a type functor, *viz.* $\mathcal{I}(s_0, \_)$ respects pullbacks, follows straightforwardly from the properties of pullbacks.

It is easily checked that $\otimes$ preserves the properties of type functors, and results in a prime-respecting functor if both components respect primes.

Because the finite elements of $\widehat{\mathbb{Q}s}$ become the primes of $!\widehat{\mathbb{Q}s}$, we see that $\mathbb{Q}$ being a type functor, so respecting finites, ensures that $!\mathbb{Q}$ respects primes. For $f : s \rightarrow s'$ an arrow in $\mathcal{I}$, the function $\mathbb{Q}f : \widehat{\mathbb{Q}s} \rightarrow \widehat{\mathbb{Q}s'}$ is injective (by Proposition 3.2 (iii)), so $!\mathbb{Q}f$ preserves nonempty meets. By the earlier remarks, it sufficient to check the condition that $!\mathbb{Q}$ respects pullbacks for x_1 a prime, but then that $!\mathbb{Q}$ respects pullbacks reduces to $\mathbb{Q}$ respecting pullbacks. □