

Department of Computer Science  
University of Aarhus

# Towards an Operational Understanding of Presheaf Models

Mikkel Nygaard

Progress Report

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>A metalanguage for concurrency</b>	<b>2</b>
2.1	Syntax . . . . .	3
2.2	Types . . . . .	3
2.3	Denotational semantics . . . . .	6
2.4	Properties . . . . .	8
<b>3</b>	<b>Operational semantics</b>	<b>9</b>
3.1	Patterns and paths . . . . .	10
3.2	Path equations . . . . .	12
3.3	Environments . . . . .	15
3.4	Proof of equivalence . . . . .	16
<b>4</b>	<b>Event structures</b>	<b>20</b>
4.1	Constructions . . . . .	21
4.2	Representing profunctors . . . . .	22
4.3	Composition . . . . .	24
4.4	A new denotational semantics . . . . .	26
<b>5</b>	<b>Future work</b>	<b>30</b>
<b>A</b>	<b>Formation rules and interpretations</b>	<b>31</b>
<b>B</b>	<b>Composition, extension and lifting</b>	<b>32</b>
<b>C</b>	<b>Proofs of restructuring lemmas</b>	<b>34</b>
<b>D</b>	<b>Operational rules</b>	<b>35</b>
<b>E</b>	<b>Proof of type-correctness</b>	<b>36</b>
<b>F</b>	<b>Cases of the equivalence proof</b>	<b>38</b>
<b>G</b>	<b>Representation of composition</b>	<b>41</b>

# 1 Introduction

This document reports on progress towards an operational understanding of presheaf models. With their built in notion of open-map bisimulation, presheaf models have been put forward as providing a domain theory for concurrency. An accompanying metalanguage in which terms are interpreted by profunctors between partial orders (with closed terms denoting presheaves) was proposed by Winskel in [21]. Our contributions fall in two parts: First, we give an SOS style operational semantics to a first-order fragment of the metalanguage and establish an isomorphism between derivations and elements of presheaves. Second, by representing the profunctors that interpret open terms with prime event structures, a new denotational semantics of the same fragment is given. With this representation, elements of presheaves can be understood as configurations of event structures with prime configurations corresponding to the derivations of the operational semantics.

The context of this work is an ongoing effort to unify models of concurrency and their notions of bisimulation into one framework, a domain theory based on presheaf models and open map bisimulation. The endeavor was initiated by Joyal, Nielsen and Winskel in [15] and pursued further in two PhD theses [3, 11] and a number of papers including [6, 20, 5, 4, 13, 21, 9, 12, 7].

Presheaf models are instantiations of the following situation: Let  $\mathbb{P}$  be a small category in which objects are viewed as computation path shapes with morphisms saying how paths can be extended. The path category  $\mathbb{P}$  embeds via Yoneda in the category  $\widehat{\mathbb{P}} = [\mathbb{P}^{\text{op}}, \mathbf{Set}]$  of presheaves over  $\mathbb{P}$  in which each object can be seen as a set of computation paths from  $\mathbb{P}$  glued together by identifying subpaths to form a process with non-deterministic branching.

The presheaf category comes with a built in notion of bisimulation, obtained from open maps [15]. Further, it can be characterised abstractly as the free colimit completion of  $\mathbb{P}$ , and so the class of path categories can be naturally assembled as the objects of a 2-category called **Cocont** with arrows colimit-preserving functors between the associated presheaf categories, and with natural transformations as 2-cells. The 2-category is equivalent to a bicategory **Prof** with the same objects, but with profunctors (which are often easier to work with) as arrows [7, 1, 2]. It was shown in [6] that open-map bisimulation is preserved along the arrows of **Cocont**, giving rise to congruence results “for free” when one interprets terms of a process language as such arrows. In fact, the arrows need only preserve

*connected* colimits to preserve open-map bisimulation, and this is exploited in the metalanguage to give meaning to operations of prefixing and parallel composition. We write **Conn** for the 2-category with this relaxed notion of arrows.

In [15] it was shown how traditional interleaving models like synchronisation trees arise as a particular instantiation of the above scheme, taking path categories to be partial orders of strings over some alphabet with prefix ordering, and that to encompass an independence model like event structures, one can replace the strings with pomsets. This is in contrast with the traditional domain-theoretic approach of powerdomains where one is limited to interleaving models because of the inherent focus on non-determinism rather than concurrency [17]. One might object that independence is “smuggled in” via the path category and so is not really a feature of the presheaf approach either. However, a new insight arising from our work representing profunctors as event structures is that the profunctors (and so presheaves) definable in the metalanguage can be seen as structures of independence in their own right.

Section 2 introduces the metalanguage and gives a brief account of its denotational semantics. In Section 3 an SOS style operational semantics is developed along with equational laws expressing its relation to the denotational semantics. Reporting on work in progress, Section 4 gives the details of our representation of profunctors using event structures and the new denotational semantics. Finally, Section 5 discusses future work.

## 2 A metalanguage for concurrency

In this section, we give a brief account of the metalanguage of [21], its syntax, type system and denotational semantics, setting the scene for an exploration of its operational content.

Terms of the language are interpreted as arrows of **Prof** of a certain type, namely profunctors  $\mathbb{P}_\perp \dashv\vdash \mathbb{Q}$  with  $\mathbb{P}$  and  $\mathbb{Q}$  being partial orders. Because of the equivalence  $\mathbf{Prof}(\mathbb{P}_\perp, \mathbb{Q}) \cong \mathbf{Conn}(\mathbb{P}, \mathbb{Q})$  of categories [7], we shall use these profunctors and arrows of **Conn** interchangeably. One can regard **Conn** as obtained via a kind of coKleisli construction (using a lifting comonad) from **Prof**. The latter can be seen as a model of Girard’s linear logic [10], giving rise to a rich type discipline. **Conn** itself is a model of affine linear logic, and weakening is exploited to interpret prefixing and parallel composition.

We refer to Appendix B for notation and concepts related to profunctors.

## 2.1 Syntax

The raw syntax of terms of the metalanguage is given by

$t, u, v ::=$	$  \begin{array}{l}  x, y, \dots \\    \lambda x. t \mid t \cdot u \\    \text{rec } x. t \mid \sum_{i \in I} t_i \\    a. t \mid (t, u) \mid t \otimes u \\    [t > p \Rightarrow u]  \end{array}  $	<p>terms:</p> <ul style="list-style-type: none"> <li>variables</li> <li>abstraction and application</li> <li>recursion and sum</li> <li>prefix, product and tensor</li> <li>match</li> </ul>
$p, q ::=$	$  \begin{array}{l}  x, y, \dots \\    a. p \\    (p, -) \mid (-, q) \\    p \otimes q  \end{array}  $	<p>patterns:</p> <ul style="list-style-type: none"> <li>variables</li> <li>prefix</li> <li>product</li> <li>tensor</li> </ul>

We write  $\emptyset$  for the term  $\sum_{i \in I} t_i$  when the indexing set  $I$  is empty. In a match term, the variables of the pattern  $p$  bind later occurrences of the variables in the body  $u$ . We call a pattern *strict* if it does not contain a subterm of the form  $a.p$  and otherwise *non-strict*. The reason for this terminology will become clear later when we interpret patterns as functors between partial orders with least elements.

## 2.2 Types

A type discipline is enforced on the above syntax. The possible types are described syntactically by the grammar

$$\mathbb{T} ::= \bigoplus_{\alpha} \mathbb{T}_{\alpha \perp} \mid \mathbb{T}_1 \& \mathbb{T}_2 \mid \mathbb{T}_1 \otimes \mathbb{T}_2 \mid \mathbb{T}_1 \multimap \mathbb{T}_2 \mid P \mid \mu_j \vec{P}. \vec{\mathbb{T}}$$

where we have suppressed the indexing set in  $\bigoplus_{\alpha} \mathbb{T}_{\alpha \perp}$  to avoid cluttering the notation.  $P$  stands for a type variable used in the definition of recursive types,  $\mu_j \vec{P}. \vec{\mathbb{T}}$ .

Type expressions are interpreted as partial orders on which we need the operation of *lifting*: A partial order  $\mathbb{P}$  is lifted by adjoining a new least element, so that the resulting partial order  $\mathbb{P}_{\perp}$  has underlying set  $\perp \cup \{ \lfloor P \rfloor : P \in \mathbb{P} \}$  ordered by  $\perp \leq \lfloor P \rfloor$  for all  $P \in \mathbb{P}$  and  $\lfloor P \rfloor \leq \lfloor P' \rfloor$  iff.  $P \leq_{\mathbb{P}} P'$ .

Let  $\mathbb{T}$  be any type expression and  $\zeta$  an environment mapping the free variables of  $\mathbb{T}$  to partial orders. We define the partial order interpreting  $\mathbb{T}$  in  $\zeta$  as follows:

**Function space** Let  $\mathbb{P}_1, \mathbb{P}_2$  be the interpretations in  $\zeta$  of  $\mathbb{T}_1$  and  $\mathbb{T}_2$  respectively. For any partial order  $\mathbb{P}$ , we write  $\mathbb{P}^{\text{op}}$  for the partial order with the same underlying set, but with the reverse order. The partial order  $\mathbb{P}_1 \multimap \mathbb{P}_2$  interpreting  $\mathbb{T}_1 \multimap \mathbb{T}_2$  in  $\zeta$  has underlying set  $(\mathbb{P}_{1\perp})^{\text{op}} \times \mathbb{P}_2$  with componentwise ordering. Note that  $\widehat{\mathbb{P}_1 \multimap \mathbb{P}_2} \cong \mathbf{Prof}(\mathbb{P}_{1\perp}, \mathbb{P}_2) \cong \mathbf{Conn}(\mathbb{P}_1, \mathbb{P}_2)$ .

**Lifted sum** Let  $\mathbb{P}_\alpha$  be the interpretation in  $\zeta$  of  $\mathbb{T}_\alpha$  for each  $\alpha \in A$ . The partial order  $\bigoplus_\alpha \mathbb{P}_{\alpha\perp}$  interpreting  $\bigoplus_\alpha \mathbb{T}_{\alpha\perp}$  in  $\zeta$  has underlying set the disjoint union of  $\mathbb{P}_{\alpha\perp}$ , ordered disjointwise. If  $A$  is empty, we get the empty partial order  $\mathbb{O}$ .

**Product** Let  $\mathbb{P}_1, \mathbb{P}_2$  be the interpretations in  $\zeta$  of  $\mathbb{T}_1$  and  $\mathbb{T}_2$  respectively. The partial order  $\mathbb{P}_1 \& \mathbb{P}_2$  interpreting  $\mathbb{T}_1 \& \mathbb{T}_2$  in  $\zeta$  has underlying set the disjoint union of  $\mathbb{P}_1$  and  $\mathbb{P}_2$  with disjointwise ordering.

**Tensor** Let  $\mathbb{P}_1, \mathbb{P}_2$  be the interpretations in  $\zeta$  of  $\mathbb{T}_1$  and  $\mathbb{T}_2$  respectively. The partial order  $\mathbb{P}_1 \otimes \mathbb{P}_2$  interpreting the expression  $\mathbb{T}_1 \otimes \mathbb{T}_2$  has underlying set  $(\mathbb{P}_{1\perp} \times \mathbb{P}_{2\perp}) \setminus \{(\perp, \perp)\}$  ordered componentwise.

**Variables** The partial order interpreting  $P$  in  $\zeta$  is just  $\zeta P$ .

**Recursion** Following [21], we can regard the partial orders as information systems (see eg. [19]) with all finite subsets consistent and entailment given by the partial order. The usual ordering on such systems induce the ordering

$$\mathbb{P} \sqsubseteq \mathbb{Q} \iff \mathbb{P} \subseteq \mathbb{Q} \wedge (\forall P, P' \in \mathbb{P}. P \leq_{\mathbb{P}} P' \iff P \leq_{\mathbb{Q}} P').$$

on partial orders, so that they form a (large) cpo with least element the empty partial order. With respect to this cpo, the operations above are all continuous, so it makes sense to take least fixed points<sup>1</sup>.

The partial order interpreting the expression  $\mu_j \vec{P}. \vec{T}$  (which abbreviates  $\mu_j P_1, \dots, P_k. (\mathbb{T}_1, \dots, \mathbb{T}_k)$ ) in  $\zeta$  is the  $j$ th component of the simultaneous least fixed point of the mapping  $\vec{I} \mapsto \vec{P}$  where  $\vec{P}$  is the interpretation of  $\vec{T}$  in  $\zeta[\vec{I}/\vec{P}]$ . We write  $\mu \vec{P}. \vec{T}$  for  $(\mu_1 \vec{P}. \vec{T}, \dots, \mu_k \vec{P}. \vec{T})$ .

We'll confuse closed type expressions with notation for the partial orders they denote. Typing judgments for terms have the form

$$x_1 : \mathbb{P}_1, \dots, x_n : \mathbb{P}_n \vdash t : \mathbb{P}$$

where the  $x_i$ 's are distinct variables. The environment must contain the free variables of  $t$ , and possibly more. Such a judgment will be interpreted

---

<sup>1</sup>For another approach, see Section 4

as a profunctor

$$t : (\mathbb{P}_1 \otimes \cdots \otimes \mathbb{P}_n)_\perp \cong \mathbb{P}_{1\perp} \times \cdots \times \mathbb{P}_{n\perp} \dashrightarrow \mathbb{P}$$

If  $n = 0$ , the tensor product is  $\mathbb{O}$ , which is unit for tensor. Judgments for patterns have a similar form

$$x_1 : \mathbb{P}_1, \dots, x_n : \mathbb{P}_n \Vdash p : \mathbb{P}$$

where again the  $x_i$ 's are distinct variables, but this time, the  $x_i$ 's all occur in  $p$  and they each occur exactly once. Such a judgment will be interpreted as a functor

$$p : \mathbb{P}_{1\perp} \times \cdots \times \mathbb{P}_{n\perp} \rightarrow \mathbb{P}_\perp.$$

We use  $\Gamma, \Delta, \Sigma$  to range over term environments and  $\Pi, \Lambda$  to range over pattern environments. Formally, we should write eg.  $\llbracket \Gamma \vdash t : \mathbb{P} \rrbracket$  for the profunctor interpreting  $\Gamma \vdash t : \mathbb{P}$ . However, when no confusion may arise, we'll write just  $\llbracket t \rrbracket$  or, indeed,  $t$  for the interpretation.

For terms  $x : \mathbb{P} \vdash t : \mathbb{Q}$  and  $\vdash u : \mathbb{P}$  one expects a substitution lemma saying that  $\llbracket t\{u/x\} \rrbracket$  is given in some way by the application of  $\llbracket t \rrbracket$  to  $\llbracket u \rrbracket$ . But the language being affinely linear means that copying and so substitution is restricted. As an example, no “diagonal” map  $\mathbb{P} \rightarrow \mathbb{P} \otimes \mathbb{P}$  sending  $X \in \widehat{\mathbb{P}}$  to  $X \otimes X$  exists in **Conn**, implying that we cannot give a meaning to the term  $x \otimes x$  in such a way that  $\llbracket x \otimes x \rrbracket$  applied to  $\llbracket u \rrbracket$  is given by  $\llbracket u \otimes u \rrbracket$ . However, one can avoid such situations by enforcing a simple syntactic condition on the occurrences of variables in terms.

**Definition 2.1** Variables  $x$  and  $y$  are said to be *crossed* in a term  $t$  if there is a subterm of  $t$  of one of the forms

$$t_1 \cdot t_2 \quad \text{or} \quad t_1 \otimes t_2 \quad \text{or} \quad [t_1 > p \Rightarrow t_2]$$

such that  $t$  has free occurrences of  $x$  and  $y$  in both  $t_1$  and  $t_2$ . □

If  $x$  is not crossed with itself in  $t$ , we can define  $\llbracket t \rrbracket$  in such a way that the substitution lemma holds. More generally, if  $x$  and  $y$  are not crossed in  $\Gamma, x : \mathbb{P} \vdash t : \mathbb{Q}$  for any  $y$  in  $\Gamma$ , and  $\Gamma \vdash u : \mathbb{P}$ , then the substitution  $t\{u/x\}$  is well-behaved in the same manner.

The formation rules for terms and patterns are listed in Appendix A. One can show by rule induction that whenever  $\Gamma, x : \mathbb{P} \vdash t : \mathbb{Q}$ , we have that  $x$  is not crossed with itself in  $t$ .

## 2.3 Denotational semantics

So terms of the metalanguage are interpreted as arrows of **Conn** between partial orders that are definable in the type system above. The denotational semantics is summarised in Appendix A using notation and concepts defined in the following.

**Lifting** Given a profunctor  $F : \mathbb{P}_\perp \dashv\vdash \mathbb{Q}$ , we take its lifting to be the profunctor  $\mathbb{P}_\perp \dashv\vdash \mathbb{Q}_\perp$ , given by  $\lfloor F \rfloor$  (see Appendix B).

**Identity, weakening and exchange** The 2-category **Conn** of course has identity arrows,  $1_{\mathbb{P}} : \mathbb{P} \rightarrow \mathbb{P}$ , which are given by the strict Yoneda embedding. The terminal object  $\mathbb{O}$  induces arrows  $0_{\mathbb{P}} : \mathbb{P} \rightarrow \mathbb{O}$  for each  $\mathbb{P}$ , namely the empty profunctor from  $\mathbb{P}$ . To interpret weakening and exchange we then just need the isomorphisms  $\mathbb{P} \otimes \mathbb{O} \cong \mathbb{P}$  and  $\mathbb{P} \otimes \mathbb{Q} \cong \mathbb{Q} \otimes \mathbb{P}$  between partial orders and the induced arrows of **Conn**.

**Tensor** Given two profunctors  $F : \mathbb{P}'_\perp \dashv\vdash \mathbb{P}$  and  $G : \mathbb{Q}'_\perp \dashv\vdash \mathbb{Q}$ , we construct their tensor product  $F \otimes G : \mathbb{P}'_\perp \times \mathbb{Q}'_\perp \dashv\vdash \mathbb{P} \otimes \mathbb{Q}$  as follows:

$$(F \otimes G)(P', Q')(P, Q) = \lfloor FP' \rfloor P \times \lfloor GQ' \rfloor Q$$

**Function space** A profunctor  $(\mathbb{P} \otimes \mathbb{Q})_\perp \dashv\vdash \mathbb{R}$  is represented by a bifunctor  $\mathbb{P}_\perp \times \mathbb{Q}_\perp \times (\mathbb{R}_\perp)^{\text{op}} \rightarrow \mathbf{Set}$  and since

$$\mathbb{P}_\perp \times \mathbb{Q}_\perp \times (\mathbb{R}_\perp)^{\text{op}} \cong \mathbb{P}_\perp \times ((\mathbb{Q}_\perp)^{\text{op}} \times \mathbb{R}_\perp)^{\text{op}} \cong \mathbb{P}_\perp \times (\mathbb{Q} \multimap \mathbb{R})^{\text{op}}$$

we get an isomorphism

$$\mathbf{Conn}(\mathbb{P} \otimes \mathbb{Q}, \mathbb{R}) \cong \mathbf{Conn}(\mathbb{P}, \mathbb{Q} \multimap \mathbb{R})$$

whose associated maps curry and uncurry yield interpretations of abstraction and application, respectively. In more detail, application of the arrow  $F : \mathbb{P} \rightarrow (\mathbb{Q} \multimap \mathbb{R})$  to  $G : \mathbb{S} \rightarrow \mathbb{Q}$  is obtained as the composition  $(\text{uncurry } F) \circ (1_{\mathbb{P}} \otimes G)$ . As a profunctor  $\mathbb{P}_\perp \times \mathbb{S}_\perp \dashv\vdash \mathbb{R}$  it is given by the coend

$$(P, S) \mapsto \int^{Q \in \mathbb{Q}_\perp} \lfloor GS \rfloor Q . F(P, Q)$$

**Sum** Given a family  $(F_i)_{i \in I}$  of arrows  $\mathbb{P} \rightarrow \mathbb{Q}$  of **Conn**, we define their sum, written  $\sum_{i \in I} F_i$ , to be the colimit  $\int^{i \in I} F_i$ , an arrow between the same objects. We write  $\emptyset$  for this colimit when  $I = \emptyset$  and typically write  $F_1 + \cdots + F_k$  when  $I$  is finite.

**Recursion** Let  $F : \mathbb{P} \otimes \mathbb{Q} \rightarrow \mathbb{Q}$  be an arrow of **Conn** and let  $\sigma : \mathbb{P}_\perp \rightarrow (\mathbb{P} \otimes \mathbb{P})_\perp$  be the arrow in **Cocont** taking  $Y \in \widehat{\mathbb{Q}_\perp}$  to  $Y \otimes \emptyset + \emptyset \otimes Y$ . We can then define an endomorphism  $\phi_F$  in **Conn** on the object  $\mathbb{P} \multimap \mathbb{Q}$  exploiting the equivalence  $\widehat{\mathbb{P} \multimap \mathbb{Q}} \cong \mathbf{Conn}(\mathbb{P}, \mathbb{Q})$ : For any  $G : \mathbb{P} \rightarrow \mathbb{Q}$  of **Conn** let  $\phi_F(G) = F \circ (1_{\mathbb{P}} \otimes G) \circ \sigma$ . Since the  $\widehat{\mathbb{P} \multimap \mathbb{Q}}$  has an initial object, the empty presheaf, and is cocomplete, it allows a least fixed point of  $\phi_F$ , written  $\text{fix } \phi_F$ .

**Lifted sum** Given an arrow  $F : \mathbb{P} \rightarrow \mathbb{Q}_a$  of **Conn**, where  $a \in A$ , we can obtain an arrow  $a.F : \mathbb{P} \rightarrow \bigoplus_\alpha \mathbb{Q}_{\alpha\perp}$  by the composition  $\text{in}_a \circ [F]$ :

$$a.F = \text{in}_a \circ [F] \quad (a.F)PQ_b = \begin{cases} [FP]Q_b & \text{if } a \equiv b \\ \emptyset & \text{if } a \not\equiv b. \end{cases} \quad \text{for } Q_b \in \mathbb{Q}_b$$

**Product** Given two arrows  $F_1 : \mathbb{P} \rightarrow \mathbb{Q}_1$  and  $F_2 : \mathbb{P} \rightarrow \mathbb{Q}_2$  of **Conn**, we can tuple them together and obtain an arrow  $(F_1, F_2) : \mathbb{P} \rightarrow \mathbb{Q}_1 \& \mathbb{Q}_2$ :

$$(F_1, F_2)PQ_i = F_iPQ_i \quad \text{for } Q_i \in \mathbb{Q}_i, i = 1, 2$$

**Match** Let  $F : \mathbb{P}'_\perp \multimap \mathbb{P}$  and  $G : \mathbb{Q}'_\perp \times \mathbb{R}_\perp \multimap \mathbb{Q}$  be profunctors and let  $H : \mathbb{R}_\perp \rightarrow \mathbb{P}_\perp$  be a functor between partial orders. We define the profunctor  $[F > H \Rightarrow G] : \mathbb{Q}'_\perp \times \mathbb{P}'_\perp \multimap \mathbb{Q}$  as the coend

$$[F > H \Rightarrow G](Q', P') = \int^{R \in \mathbb{R}_\perp} [FP'](HR) \cdot G(Q', R)$$

Notice that if  $H$  is the identity, then  $[F > H \Rightarrow G]$  is just the composition  $G \circ (1_{\mathbb{Q}'} \otimes F)$ .

**Patterns** The identity functor on the partial order  $\mathbb{P}_\perp$  is used to interpret the pattern  $x : \mathbb{P} \vdash x : \mathbb{P}$ . Let  $H : \mathbb{P}_\perp \rightarrow \mathbb{Q}_{a\perp}$  for some  $a \in A$  and  $H_i : \mathbb{P}_{i\perp} \rightarrow \mathbb{Q}_{i\perp}$  for  $i = 1, 2$  be functors between partial orders. We can then define the functors

$$\begin{aligned} a.H & : \mathbb{P}_\perp \rightarrow (\bigoplus_\alpha \mathbb{Q}_{\alpha\perp})_\perp & P & \mapsto [\text{in}_a HP] \\ (H_1, -) & : \mathbb{P}_{1\perp} \rightarrow (\mathbb{Q}_1 \& \mathbb{Q}_2)_\perp & P & \mapsto \begin{cases} \perp & \text{if } H_1 P = \perp \\ [\text{in}_1 P'] & \text{if } H_1 P = [P'] \end{cases} \\ (-, H_2) & : \mathbb{P}_{2\perp} \rightarrow (\mathbb{Q}_1 \& \mathbb{Q}_2)_\perp & P & \mapsto \begin{cases} \perp & \text{if } H_2 P = \perp \\ [\text{in}_2 P'] & \text{if } H_2 P = [P'] \end{cases} \\ H_1 \otimes H_2 & : \mathbb{P}_{1\perp} \times \mathbb{P}_{2\perp} \rightarrow (\mathbb{Q}_1 \otimes \mathbb{Q}_2)_\perp & (P_1, P_2) & \mapsto (H_1 P_1, H_2 P_2) \end{aligned}$$

## 2.4 Properties

For terms  $\Gamma, x : \mathbb{P} \vdash t : \mathbb{Q}$  and  $\Delta \vdash u : \mathbb{P}$  such that the variables of  $\Gamma$  and  $\Delta$  are disjoint, we have  $\llbracket t \cdot u \rrbracket \cong \llbracket t\{u/x\} \rrbracket$  from [21]. If  $\mathbb{P} = \mathbb{Q}$  and  $x$  and  $y$  are not crossed in  $t$  for any  $y$  of  $\Gamma$ , we similarly have  $\llbracket \text{rec } x.t \rrbracket \cong \llbracket t\{\text{rec } x.t/x\} \rrbracket$ . We shall not need these results and so omit their proofs.

**Lemma 2.2** *Suppose  $\Gamma \vdash t : \mathbb{P}$  and  $\mathbb{P} \Vdash \vec{x} : \mathbb{P}$ . Then  $\Gamma \vdash [t > \vec{x} \Rightarrow \vec{x}] : \mathbb{P}$  and there is an isomorphism*

$$t \cong [t > \vec{x} \Rightarrow \vec{x}]$$

*between profunctors.*

**Proof:** The term  $\vec{x}$  is interpreted as the profunctor  $1_{\mathbb{P}}$  which is the identity for composition.  $\square$

**Lemma 2.3** *Suppose  $\Delta, \Gamma \vdash [t > p \Rightarrow u] : \mathbb{Q}$  with  $\Pi \Vdash p : \mathbb{P}$ . Let  $C$  be a colimit-preserving map  $\widehat{\mathbb{Q}} \rightarrow \widehat{\mathbb{R}}$ . Then there is an isomorphism*

$$C([t > p \Rightarrow u]) \cong [t > p \Rightarrow C(u)]$$

*between profunctors. If  $p$  is strict,  $C$  only needs to preserve connected colimits.*

**Proof:** The coend interpreting  $[t > p \Rightarrow u]$  can be rewritten as a colimit which is connected if  $p$  is strict. Let  $\delta \in \Delta_{\perp}$  and  $\gamma \in \Gamma_{\perp}$ . Then

$$\begin{aligned} & [t > p \Rightarrow u](\delta, \gamma) \\ &= \int^{P \in \Pi_{\perp}} [t\gamma](pP) \cdot u(\delta, P) \\ &\cong \int^{P \in \Pi_{\perp}} \int^{e \in [t\gamma](pP)} u(\delta, P) \\ &\cong \int^{(e, P) \in \text{els}[t\gamma](p-)} u(\delta, P) \end{aligned}$$

Assume that  $p$  is strict and so sends  $\perp$  to  $\perp$ . Composing with the rooted presheaf  $[t\gamma]$  then yields another rooted presheaf and so the domain of the diagram of the last-mentioned colimit above has an initial object. Hence, the diagram is connected.  $\square$

Notice that for  $p$  strict we can let  $C$  be (the interpretation of) any context of the metalanguage, so long as  $C([t > p \Rightarrow u])$  is well-formed and no variables of  $p$  occur free in  $C$ .

The remaining two lemmas say how to restructure pattern matches. The proofs can be found in Appendix C.

**Lemma 2.4** Assume  $\Sigma, \Delta, \Gamma \vdash [t > s \Rightarrow [u > q \Rightarrow v]] : \mathbb{R}$  with

$$\begin{array}{ll} \Gamma \vdash t : \mathbb{P} & \mathbb{P}_1, \mathbb{P}_2 \Vdash s : \mathbb{P} \\ \Delta, \mathbb{P}_1 \vdash u : \mathbb{Q} & \wedge \Vdash q : \mathbb{Q} \\ \Sigma, \mathbb{P}_2, \wedge \vdash v : \mathbb{R} & \end{array}$$

and where  $s$  is strict and the variables of  $\mathbb{P}_1, \mathbb{P}_2$  and  $\wedge$  are distinct. If  $\vec{y}$  are the variables of  $\mathbb{P}_2$ , then  $\Sigma, \Delta, \Gamma \vdash [[t > s \Rightarrow u \otimes \vec{y}] > q \otimes \vec{y} \Rightarrow v] : \mathbb{R}$  and there is an isomorphism

$$[t > s \Rightarrow [u > q \Rightarrow v]] \cong [[t > s \Rightarrow u \otimes \vec{y}] > q \otimes \vec{y} \Rightarrow v].$$

between profunctors.

**Lemma 2.5** Assume  $\Delta, \Gamma \vdash [t > p \Rightarrow [x > q \Rightarrow u]] : \mathbb{R}$  with

$$\begin{array}{ll} \Gamma \vdash t : \mathbb{P} & \mathbb{P}_1, x : \mathbb{Q}, \mathbb{P}_2 \Vdash p : \mathbb{P} \\ \Delta, \mathbb{P}_1, \wedge, \mathbb{P}_2 \vdash u : \mathbb{R} & \wedge \Vdash q : \mathbb{Q} \end{array}$$

and where the variables of  $p$  and  $q$  are disjoint. Then  $\Delta, \Gamma \vdash [t > p\{q/x\} \Rightarrow u] : \mathbb{R}$  and there is an isomorphism

$$[t > p \Rightarrow [x > q \Rightarrow u]] \cong [t > p\{q/x\} \Rightarrow u]$$

between profunctors.

### 3 Operational semantics

A closed term  $\vdash t : \mathbb{P}$  of the metalanguage is interpreted as a presheaf  $t$  over  $\mathbb{P}$ . According to the understanding of presheaves as processes, we should look to the category of elements of  $t$ ,  $\text{els}(t)$ , for an operational understanding of this term. The category of elements can be seen as a transition system (with possibly multiple initial states) in which each branch corresponds to an element of  $\mathbb{P}$ . By lifting  $t$ , we adjoin a new initial element to  $\text{els}(t)$ , thereby obtaining a transition system with a designated initial state. It would be natural to attempt writing down an operational semantics in small step SOS style [16] in such a way that the generated transition system  $T$  can be somehow related to  $\text{els}[t]$ .

The simplest possible  $T$  has states just the (closed) terms of the metalanguage with  $t$  itself the initial state. What transitions should there be out of  $t$ ? For each object  $(x, P)$  of  $\text{els}(t)$ , there is a unique arrow

$$(*, \perp) \xrightarrow{\perp \leq [P]} (x, P)$$

in  $\text{els}[t]$ . The element  $P \in \mathbb{P}$  induces a presheaf  $P^*t$  over  $(\mathbb{P}\downarrow P)$ , obtained by restricting  $t$ :

$$P^*t(Q \geq P) = tQ.$$

This presheaf has a rooted component,  $X = [X']$ , induced by  $x$ :

$$X(Q \geq P) = \{y \in tQ \mid t(Q \geq P)y = x\}.$$

So having performed the  $P$ -computation  $x$ ,  $t$  continues to behave like  $X'$ . The hope is that  $X'$  is itself expressible in the language as a term  $u$ , and in that case the transitions out of  $t$  should be

$$\{t \xrightarrow{P} u : [u] \text{ is a rooted component of } P^*t\} \quad (*)$$

However, having just terms as states,  $T$  collapses into one state all those objects  $(x, P)$  of  $\text{els}[t]$  that induce the same  $u$ , and so is clearly not isomorphic to  $\text{els}[t]$ . To remedy this, states of  $T$  could instead take the form  $(x, u)$  with  $x$  being the element of  $[t]$  that induces  $u$ . However, then we must face the question of how to treat such elements operationally.

This question will be answered partially in this and the following section. To start with, we need to generalise the operation  $P^*$  in order to handle strict matches.

### 3.1 Patterns and paths

In terms of the form  $[t > p \Rightarrow u]$  with  $p$  strict (for concreteness, suppose  $p \equiv x \otimes y$ ), the open term  $u$  may be able to perform a computation regardless of the capabilities of  $t$ . For instance, the interpretation of the term  $[\text{rec } z.z > x \otimes y \Rightarrow a.x]$  is isomorphic to that of  $a.\emptyset$ . And notice that we cannot expect to obtain from  $t$  a term of the form  $t_1 \otimes t_2$  that can be substituted for  $x$  and  $y$  in  $u$ . So it seems that we need to be able to talk about the  $P$ -computations of open terms in an environment. It will therefore be helpful to extend the operation  $P^*$  to a functor that can be precomposed with rooted profunctors  $\mathbb{P}_\perp \leftrightarrow \mathbb{P}_\perp$ .

An object  $P$  of a partial order  $\mathbb{P}$  induces a functor  $\tilde{P} : (\mathbb{P}\downarrow P) \rightarrow \mathbb{P}$  embedding the category of arrows out of  $P$  inside  $\mathbb{P}$ . By postcomposing with  $y_{\mathbb{P}}$  and taking the right adjoint of the resulting profunctor's left Kan extension, we obtain the functor

$$P^* : \widehat{\mathbb{P}} \rightarrow \widehat{(\mathbb{P}\downarrow P)} \quad X \mapsto \widehat{\mathbb{P}}(y_{\mathbb{P}}(\tilde{P}-), X) \stackrel{\text{Yoneda}}{\cong} X(\tilde{P}-).$$

We see that it really is an extension of the operation from above. But in fact, we do not need this extra class of embeddings  $\tilde{P}$  into ground types of the metalanguage, because they are closely related to the following patterns:

$$p, q ::= a.x \mid a.p \mid p \otimes y \mid x \otimes q \mid p \otimes q \mid (p, -) \mid (-, q)$$

– the only difference being that the codomain of the patterns are lifted. All these patterns are non-strict and any strict sub-pattern is a variable. We call such patterns *paths* because of the correspondence with elements of the types. Using the same construction as above, we obtain from  $p : \mathbb{P}_\perp \rightarrow \mathbb{P}_\perp$  an “inverse” functor

$$p^* : \widehat{\mathbb{P}_\perp} \rightarrow \widehat{\mathbb{P}_\perp} \quad X \mapsto \widehat{\mathbb{P}_\perp}(y_{\mathbb{P}_\perp}(p-), X) \stackrel{\text{Yoneda}}{\cong} X(p-).$$

The functor  $p^*$  is itself a left adjoint and so preserves arbitrary colimits. We’ll write  $p^*[t]$  for the composition  $p^* \circ \llbracket \llbracket \Gamma \vdash t : \mathbb{P} \rrbracket \rrbracket$ , any judgment  $\Gamma \vdash t : \mathbb{P}$ .

We are thus aiming at an operational semantics in which  $t \xrightarrow{p} t'$  is derivable for a closed term  $t$  and some path  $p$ , if  $[t']$  is a rooted component of  $p^*[t]$ . It will turn out (Lemma 3.1) that  $p^*[t]$  is isomorphic to the pattern match  $[t > p(\vec{y}) \Rightarrow [\vec{y}]]$ , indicating that matching  $t$  against a path  $p$  can be treated operationally by having  $t$  make a transition  $t \xrightarrow{p} t'$ .

This illustrates the satisfying economy of the approach, but notice that we do not have any patterns that can be used with higher types, and it is at present unclear to us how to define them because of the contravariance of the left component of  $\mathbb{P} \multimap \mathbb{Q}$ . We return to this issue in Section 5.

Another unfortunate consequence of the approach is that the uniformity of the denotational semantics regarding the different forms of patterns cannot be carried over into the operational semantics because we need to distinguish paths from general patterns. A clean and non-trivial language is obtained by restricting to the *tensor-fragment* of the metalanguage with terms

$$t, u ::= x \mid \text{rec } x.t \mid \sum_{i \in I} t_i \mid a.t \mid t \otimes u \mid [t > s \Rightarrow u] \mid [t > p \Rightarrow u]$$

and patterns either tensors of variables or paths:

$$\begin{array}{ll} s ::= x \mid s \otimes s & \text{strict patterns} \\ p, q ::= a.x \mid a.p \mid p \otimes y \mid x \otimes q \mid p \otimes q & \text{paths} \end{array}$$

We’ll write strict patterns as just  $\vec{x}$ , exploiting associativity of  $\otimes$  to suppress the grouping. Similarly, we’ll write eg.  $p \otimes \vec{x}$  for the path obtained by regrouping.

One can easily add to the tensor fragment product terms  $(t, u)$  and projections  $\text{fst}(t), \text{snd}(t)$  in place of the matches  $[t > (x, -) \Rightarrow x]$  and  $[t > (-, x) \Rightarrow x]$ , respectively, thereby regaining the expressiveness of the ground type language, but doing so gives no new insight, adding just trivial cases to the proofs that follow.

### 3.2 Path equations

So we are interested in the rooted components of the profunctor  $p^*[t]$  where  $p$  is a path. As said this is closely related to pattern matching in the metalanguage:

**Lemma 3.1** *Assume  $\Gamma \vdash t : \mathbb{P}$  and  $\Pi \Vdash p : \mathbb{P}$  with  $\vec{y}$  the variables of  $\Pi$ . Then*

$$p^*[t] \cong [t > p(\vec{y}) \Rightarrow [\vec{y}]]$$

**Proof:** Let  $\gamma \in \Gamma_{\perp}$ . Then, since  $y_{\Pi_{\perp}} \cong [j_{\Pi_{\perp}} -] = [\vec{y} -]$ ,

$$\begin{aligned} p^*[t\gamma] & \\ &\cong [t\gamma](p-) \\ &\cong \int^{P \in \Pi_{\perp}} [t\gamma](pP) \cdot y_{\Pi_{\perp}} P \\ &\cong \int^{P \in \Pi_{\perp}} [t\gamma](pP) \cdot [\vec{y}P] \\ &\cong [t > p(\vec{y}) \Rightarrow [\vec{y}]]\gamma \end{aligned}$$

– as wanted. □

As an easy corollary, we get

**Corollary 3.2** *Assume  $\Gamma \vdash t : \mathbb{P}$  and that  $\mathbb{P} \Vdash \vec{x} : \mathbb{P}$ . Then,*

$$\vec{x}^*[t] \cong [t].$$

**Proof:** By Lemma 3.1,  $\vec{x}^*[t] \cong [t > \vec{x} \Rightarrow [\vec{x}]]$ . Since  $\vec{x}$  is strict and  $[-]$  preserves connected colimits, the result follows by Lemma 2.3 and Lemma 2.2. □

These results are helpful but to write down an SOS-style operational semantics, we need to know how to obtain the rooted components of  $p^*[t]$  in terms of the rooted components of  $q^*[u]$  where  $u$  is structurally smaller than  $t$ . This is the purpose of the following five lemmas.

**Lemma 3.3** *Assume  $\Gamma \vdash \sum_{i \in I} t_i : \mathbb{P}$  and that  $\Pi \Vdash p : \mathbb{P}$  is non-strict. Then,*

$$p^*[\sum_{i \in I} t_i] \cong \sum_{i \in I} p^*[t_i].$$

**Proof:** Let  $\gamma \in \Gamma_{\perp}$  and  $P' \in \mathbb{P}_{\perp}$  with  $pP' = \lfloor P \rfloor$ . Then  $\lfloor t \rfloor(pP') \cong tP$  for any profunctor  $t$ , and so, since colimits are obtained pointwise,

$$\begin{aligned} & (p^* \lfloor \Sigma_{i \in I} t_i \gamma \rfloor) P' \\ & \cong \lfloor \Sigma_{i \in I} t_i \gamma \rfloor (pP') \\ & \cong \Sigma_{i \in I} (t_i \gamma) P \\ & \cong \Sigma_{i \in I} \lfloor t_i \gamma \rfloor (pP') \\ & \cong \Sigma_{i \in I} (p^* \lfloor t_i \gamma \rfloor) P' \end{aligned}$$

as wanted. □

**Lemma 3.4** Assume  $\Gamma \vdash a.t : \bigoplus_{\alpha} \mathbb{P}_{\alpha \perp}$  and that  $\mathbb{P} \Vdash b.p : \bigoplus_{\alpha} \mathbb{P}_{\alpha \perp}$ . Then,

$$b.p^* \lfloor a.t \rfloor \cong \begin{cases} p^* \lfloor t \rfloor & \text{if } a \equiv b \\ \emptyset & \text{if } a \not\equiv b. \end{cases}$$

**Proof:** Let  $\gamma \in \Gamma_{\perp}$  and  $P' \in \mathbb{P}_{\perp}$ . Then,

$$\begin{aligned} & (b.p^* \lfloor (a.t) \gamma \rfloor) P' \\ & \cong \lfloor (a.t) \gamma \rfloor ((b.p) P') \\ & = \lfloor \text{in}_a \lfloor t \gamma \rfloor \rfloor \lfloor \text{in}_b (pP') \rfloor \\ & \cong \text{in}_a \lfloor t \gamma \rfloor (\text{in}_b (pP')) \\ & \cong \begin{cases} \lfloor t \gamma \rfloor (pP') \cong (p^* \lfloor t \gamma \rfloor) P' & \text{if } a \equiv b \\ \emptyset & \text{if } a \not\equiv b \end{cases} \end{aligned}$$

as wanted. □

**Lemma 3.5** Assume  $\Gamma, \Delta \vdash t \otimes u : \mathbb{P} \otimes \mathbb{Q}$  and let  $\mathbb{P}, \Delta \Vdash p \otimes q : \mathbb{P} \otimes \mathbb{Q}$ . Then,

$$(p \otimes q)^* \lfloor t \otimes u \rfloor \cong \Sigma_{t'} \Sigma_{u'} \lfloor t' \otimes u' \rfloor$$

where the sums are taken over  $\lfloor t' \rfloor$  in  $p^* \lfloor t \rfloor$  and  $\lfloor u' \rfloor$  in  $q^* \lfloor u \rfloor$ , respectively.

**Proof:** Let  $\gamma \in \Gamma_{\perp}$ ,  $\delta \in \Delta_{\perp}$ ,  $P' \in \mathbb{P}_{\perp}$  and  $Q' \in \mathbb{Q}_{\perp}$ . Then,

$$\begin{aligned} & (p \otimes q)^* \lfloor (t \otimes u)(\gamma, \delta) \rfloor (P', Q') \\ & \cong \lfloor (t \otimes u)(\gamma, \delta) \rfloor (pP', qQ') \\ & = \lfloor t \gamma \rfloor (pP') \times \lfloor u \delta \rfloor (qQ') \\ & \cong (p^* \lfloor t \gamma \rfloor) P' \times (q^* \lfloor u \delta \rfloor) Q' \\ & \cong \Sigma_{t'} \lfloor t' \gamma \rfloor P' \times \Sigma_{u'} \lfloor u' \delta \rfloor Q' \\ & \cong \Sigma_{t'} \Sigma_{u'} \lfloor t' \gamma \rfloor P' \times \lfloor u' \delta \rfloor Q' \\ & = \Sigma_{t'} \Sigma_{u'} \lfloor (t' \otimes u')(\gamma, \delta) \rfloor (P', Q') \end{aligned}$$

– as wanted. □

Given two profunctors  $t : \Gamma_{\perp} \dashv\vdash \mathbb{P}_{\perp}$  and  $u : \Delta_{\perp} \dashv\vdash \mathbb{Q}_{\perp}$ , we shall use the notation  $t \times_{\perp} u$  for the profunctor  $(\Gamma \otimes \Delta)_{\perp} \dashv\vdash \mathbb{P} \otimes \mathbb{Q}$  given by

$$\lambda(\gamma, \delta). \lambda(P, Q). (t\gamma)P \times (u\delta)Q$$

This notation allows us to write the above equivalence as  $(p \otimes q)^* [t \otimes u] \cong p^* [t] \times_{\perp} q^* [u]$  which is sometimes more convenient.

**Lemma 3.6** *Assume  $\Delta, \Gamma \vdash [v > s \Rightarrow t] : \mathbb{P}$  with  $s$  strict. Then,*

$$p^* [[v > s \Rightarrow t]] \cong [v > s \Rightarrow p^* [t]].$$

**Proof:** The map  $p^* [-]$  preserves connected colimits. The result then follows from Lemma 2.3.  $\square$

This lemma justifies the informal reasoning at the start of Section 3.1, that the “flow of control” of  $[v > s \Rightarrow t]$  with  $s$  strict moves to  $t$  first. The opposite situation arises when the pattern is a path:

**Lemma 3.7** *Assume  $\Delta, \Gamma \vdash [t > p \Rightarrow u] : \mathbb{Q}$  with  $p = p(\vec{x})$  a path and typings  $\Delta \vdash t : \mathbb{P}, \Pi \Vdash p : \mathbb{P}$  and  $\Gamma, \mathbf{x} : \mathbb{P}' \vdash u : \mathbb{Q}$ . Then,*

$$[t > p(\vec{x}) \Rightarrow u] \cong \sum_{t'} [t' > \vec{x} \Rightarrow u]$$

where the sum is over  $[t']$  in  $p^* [t]$ .

**Proof:** Let  $\delta \in \Delta_{\perp}, \gamma \in \Gamma_{\perp}$  and  $Q \in \mathbb{Q}$ . Then,

$$\begin{aligned} & [t > p \Rightarrow u](\delta, \gamma)Q \\ & \cong \int^{P' \in \mathbb{P}_{\perp}} [t\delta](pP') \times u(\gamma, P')Q \\ & \cong \int^{P' \in \mathbb{P}_{\perp}} p^* [t\delta]P' \times u(\gamma, P')Q \\ & \cong \int^{P' \in \mathbb{P}_{\perp}} (\sum_{t'} [t'\delta])P' \times u(\gamma, P')Q \\ & \cong \int^{P' \in \mathbb{P}_{\perp}} \sum_{t'} ([t'\delta]P' \times u(\gamma, P')Q) \\ & \cong \sum_{t'} \int^{P' \in \mathbb{P}_{\perp}} [t'\delta]P' \times u(\gamma, P')Q \quad \text{Fubini} \\ & \cong \sum_{t'} [t' > \vec{x} \Rightarrow u](\delta, \gamma)Q \end{aligned}$$

– as wanted.  $\square$

Disregarding matches and recursion, the above lemmas immediately suggest the following operational rules for closed terms (ns abbreviates non-strict):

$$\frac{}{t \xrightarrow{x} t} \quad \frac{t_j \xrightarrow{p} t' \quad j \in I \quad p \text{ ns}}{\sum_{i \in I} t_i \xrightarrow{p} t'} \quad \frac{t \xrightarrow{p} t'}{a.t \xrightarrow{a.p} t'} \quad \frac{t \xrightarrow{p} t' \quad u \xrightarrow{q} u'}{t \otimes u \xrightarrow{p \otimes q} t' \otimes u'}$$

In fact, because each result states an isomorphism between profunctors, one is able to prove a strong relationship between this operational semantics and the denotational semantics for this small fragment:

**Proposition 3.8** *Let  $t$  be any well-formed, closed term of the restricted language and  $p$  any path. Then, summing over all derivations  $d$  with conclusion of the form  $t \xrightarrow{p} t'$ , for some  $t'$ , we have  $p^*[t] \cong \Sigma_d[t']$ .*

**Proof:** By structural induction on  $t$ . □

In other words, the two semantics are equivalent with derivations corresponding exactly to elements of presheaves. An obvious rule for recursion is

$$\frac{t\{\text{rec } x.t/x\} \xrightarrow{p} t'}{\text{rec } x.t \xrightarrow{p} t'}$$

and one should be able to extend the equivalence proof without too much trouble using standard techniques. But for now, we are more interested in covering matches and for that we unfortunately have to complicate the operational semantics.

### 3.3 Environments

Consider a strict match  $[v > s \Rightarrow t]$ . Because of Lemma 3.6 it is necessary in the operational semantics to consider  $v > s$  as an *environment* for the execution of  $t$ , providing bindings to the free variables of that term. It will be convenient to write any closed term as  $[\varepsilon \Rightarrow t]$  where  $\varepsilon$  is a possibly empty environment.

We face then the usual problem of explicit sequencing when two subterms of  $t$  refer to the environment. Lemma 3.7 together with Lemma 3.6 say that in the case of  $[\varepsilon \Rightarrow [t > p \Rightarrow u]]$  with  $p$  a path, the sequencing should be  $t$  first, then  $u$ . Tensor, on the other hand, is completely symmetric, but we can evade the problem by restricting ourselves to paths that are non-strict in exactly one component. Then, the path itself determines in which component to move:

$$\frac{[\varepsilon \Rightarrow t] \xrightarrow{p} [\varepsilon' \Rightarrow t']}{[\varepsilon \Rightarrow t \otimes u] \xrightarrow{p \otimes y} [\varepsilon' \Rightarrow t' \otimes u]} \quad \frac{[\varepsilon \Rightarrow u] \xrightarrow{q} [\varepsilon' \Rightarrow u']}{[\varepsilon \Rightarrow t \otimes u] \xrightarrow{x \otimes q} [\varepsilon' \Rightarrow t \otimes u']}$$

Another issue with the environments is how to handle the cases where the demand on a term yields a demand on a subterm which does not necessarily contain all the free variables of the whole term. To be concrete,

consider the left tensor rule above. To prove that it preserves the equivalence of derivations and rooted components, we must be able to show that for *any* term  $u$  such that  $[\varepsilon \Rightarrow t \otimes u]$  is well-formed,

$$(p \otimes y)^* [[\varepsilon \Rightarrow t \otimes u]] \cong \Sigma_d [[\varepsilon' \Rightarrow t' \otimes u]] \quad (*)$$

where the sum is over derivations  $d \vdash [\varepsilon \Rightarrow t] \xrightarrow{p} [\varepsilon' \Rightarrow t']$ . The environment  $\varepsilon'$  must thus provide suitable bindings for the free variables of  $u$  even though  $u$  does not occur in  $d$ .

This consideration is built into the operational semantics listed in Appendix D, in that bindings for variables are never lost. The isomorphism (\*) above is reflected in the induction hypothesis of the proof of the following proposition, saying that the operational semantics is type correct:

**Proposition 3.9** *Assume  $\vdash [\varepsilon \Rightarrow t] : \mathbb{P}$ . If  $[\varepsilon \Rightarrow t] \xrightarrow{p} [\varepsilon' \Rightarrow t']$ , then  $p$  is a path with  $\sqcap \Vdash p : \mathbb{P}$  for some  $\sqcap$ , and  $\vdash [\varepsilon' \Rightarrow t'] : \sqcap$ .*

**Proof:** By rule induction using the induction hypothesis

$\mathcal{Q}([\varepsilon \Rightarrow t])$ : Let  $r$  be any term such that  $\vdash [\varepsilon \Rightarrow t \otimes r] : \mathbb{P} \otimes \mathbb{R}$ . If  $[\varepsilon \Rightarrow t] \xrightarrow{p} [\varepsilon' \Rightarrow t']$  then  $p$  is a path with  $\sqcap \Vdash p : \mathbb{P}$  for some  $\sqcap$ , and  $\vdash [\varepsilon' \Rightarrow t' \otimes r] : \sqcap \otimes \mathbb{R}$ .

The straightforward proof is deferred to Appendix E. □

### 3.4 Proof of equivalence

Clearly, we cannot use structural induction to prove a result like Proposition 3.8 for the operational semantics of Appendix D. However, it is possible to define a well-founded order  $\succ$  on (equivalence classes of  $\alpha$ -equivalent) terms that will enable a proof by well-founded induction. Let  $\succ$  be given as the smallest transitive relation generated by the following rules:

$$\frac{t \text{ structurally larger than } t'}{t \succ t'} \qquad \frac{p \text{ a path}}{[t > p(\vec{x}) \Rightarrow u] \succ [t > \vec{x} \Rightarrow u]}$$

$$\frac{t \succ t'}{C(t) \succ C(t')} \qquad \frac{t \succ t'}{[\varepsilon \Rightarrow t] \succ [\varepsilon' \Rightarrow t']}$$

– where  $C$  ranges over syntactic contexts where the hole is not part of a sum. Notice that if  $[\varepsilon \Rightarrow t] \succ [\varepsilon' \Rightarrow t']$  then either  $t \succ t'$  or  $t \equiv t'$  and

$$\varepsilon \equiv v > s \quad \text{and} \quad \varepsilon' \equiv v' > s \quad \text{and} \quad v \succ v'.$$

We have the following result:

**Lemma 3.10** *If  $[\varepsilon \Rightarrow t] \xrightarrow{p} [\varepsilon' \Rightarrow t']$  then  $[\varepsilon \Rightarrow t] \succ [\varepsilon' \Rightarrow t']$ .*

**Proof:** By rule induction. In the case of path match, the rule is:

$$\frac{[\varepsilon \Rightarrow t] \xrightarrow{p} [\varepsilon' \Rightarrow t'] \quad [\varepsilon' \Rightarrow [t' > \vec{x} \Rightarrow u]] \xrightarrow{q} [\varepsilon'' \Rightarrow u']}{[\varepsilon \Rightarrow [t > p(\vec{x}) \Rightarrow u]] \xrightarrow{q} [\varepsilon'' \Rightarrow u']}$$

By the induction hypothesis,  $[\varepsilon \Rightarrow t] \succ [\varepsilon' \Rightarrow t']$ . There are two cases. If  $t \succ t'$ , we have

$$[\varepsilon \Rightarrow [t > p(\vec{x}) \Rightarrow u]] \succ [\varepsilon' \Rightarrow [t' > p(\vec{x}) \Rightarrow u]].$$

Otherwise,  $t \equiv t'$  and  $\varepsilon \equiv v > s$  and  $\varepsilon' \equiv v' > s$  for some  $v, v', s$  with  $v \succ v'$ . Then,

$$\begin{aligned} [\varepsilon \Rightarrow [t > p(\vec{x}) \Rightarrow u]] &\equiv [v > s \Rightarrow [t > p(\vec{x}) \Rightarrow u]] \\ &\succ [v' > s \Rightarrow [t > p(\vec{x}) \Rightarrow u]] \\ &\equiv [\varepsilon' \Rightarrow [t' > p(\vec{x}) \Rightarrow u]] \end{aligned}$$

Either way,

$$\begin{aligned} [\varepsilon \Rightarrow [t > p(\vec{x}) \Rightarrow u]] &\succ [\varepsilon' \Rightarrow [t' > p(\vec{x}) \Rightarrow u]] \\ &\succ [\varepsilon' \Rightarrow [t' > \vec{x} \Rightarrow u]] \\ &\succ [\varepsilon'' \Rightarrow u'] \quad \text{by IH} \end{aligned}$$

– as wanted. The other rules are handled in a similar way.  $\square$

**Theorem 3.11 (Equivalence)** *Let  $t$  be any well-formed, closed term and  $p$  any path with exactly one non-strict tensor component. Then, summing over all derivations  $d$  with conclusion of the form  $t \xrightarrow{p} t'$ , for some  $t'$ , we have  $p^*[t] \cong \Sigma_d[t']$ .*

**Proof:** By well-founded induction using the induction hypothesis

$Q([\varepsilon \Rightarrow t])$ : Let  $r$  be any term such that  $[\varepsilon \Rightarrow t \otimes r]$  is well-formed and closed. Let  $p$  be any path with exactly one non-strict tensor-component. Then, summing over all derivations  $d$  with conclusion of the form  $[\varepsilon \Rightarrow t] \xrightarrow{p} [\varepsilon' \Rightarrow t']$ , some  $\varepsilon', t'$ , we have

$$(p \otimes \vec{z})^* [[\varepsilon \Rightarrow t \otimes r]] \cong \Sigma_d [[\varepsilon' \Rightarrow t' \otimes r]]$$

Because of Proposition 3.9, we need not concern ourselves with questions of well-formedness in what follows. We proceed by case analysis on  $t$  and  $p$ , deferring the simpler cases to Appendix F:

**Term  $x$**  There is one possible last rule:

$$\frac{v \xrightarrow{s\{p/x\}} v'}{[v > s \Rightarrow x] \xrightarrow{p} [v' > s \Rightarrow x]}$$

Since  $v$  is structurally smaller than  $[v > s \Rightarrow x]$ , we have  $[v > s \Rightarrow x] \succ v$  and so by induction hypothesis,

$$s\{p/x\}^* [v] \cong \Sigma_{d_v} [v'],$$

Let  $r$  be any term such that  $[v > s \Rightarrow x \otimes r]$  is well-formed and closed. We get:

$$\begin{aligned} & (p \otimes \vec{z})^* [[v > s \Rightarrow x \otimes r]] \\ & \cong [v > s \Rightarrow p^* [x] \times_{\perp} \vec{z}^* [r]] \quad (3.6),(3.5) \\ & \cong [v > s \Rightarrow [x > p(\vec{y}) \Rightarrow [\vec{y}]] \times_{\perp} [r]] \quad (3.1),(3.2) \end{aligned}$$

The context  $(\cdot \times_{\perp} [r])$  preserves arbitrary colimits and so according to Lemma 2.3, we can continue with

$$\begin{aligned} & \cong [v > s \Rightarrow [x > p(\vec{y}) \Rightarrow [\vec{y}]] \times_{\perp} [r]] \quad (3.1) \\ & \cong [v > s\{p(\vec{y})/x\} \Rightarrow [\vec{y}]] \times_{\perp} [r] \quad (2.5) \\ & \cong [v > s\{p(\vec{y})/x\} \Rightarrow [\vec{y} \otimes r]] \quad (3.5) \\ & \cong \Sigma_{v'} [v' > s\{\vec{y}/x\} \Rightarrow [\vec{y} \otimes r]] \quad (3.7) \\ & \cong \Sigma_{d_v} [v' > s\{\vec{y}/x\} \Rightarrow [\vec{y} \otimes r]] \\ & \cong \Sigma_{d_v} [[v' > s\{\vec{y}/x\} \Rightarrow \vec{y} \otimes r]] \quad (2.3) \\ & \cong \Sigma_{d_v} [[v' > s \Rightarrow x \otimes r]] \quad \text{renaming } \vec{y} \text{ to } x \end{aligned}$$

– as wanted.

**Term  $[t > s \Rightarrow u]$  with  $s$  strict** There is one possible last rule:

$$\frac{[[v > s_1 \Rightarrow t \otimes \vec{y}] > s \otimes \vec{y} \Rightarrow u] \xrightarrow{q} [v' > s' \Rightarrow u']}{[v > s_1 \Rightarrow [t > s \Rightarrow u]] \xrightarrow{q} [v' > s' \Rightarrow u']}$$

Let  $r$  be any term such that  $[v > s_1 \Rightarrow [t > s \Rightarrow u] \otimes r]$  is well-formed and closed. Then, by Lemma 2.3 and Lemma 2.4, we get

$$[v > s_1 \Rightarrow [t > s \Rightarrow u] \otimes r] \cong [[v > s_1 \otimes \vec{y}] > s \otimes \vec{y} \Rightarrow u \otimes r]$$

and so  $Q([v > s_1 \Rightarrow [t > s \Rightarrow u]])$  follows directly if  $Q([[v > s_1 \Rightarrow t \otimes \vec{y}] > s \otimes \vec{y} \Rightarrow u])$ . But  $u$  is structurally smaller than  $[t > s \Rightarrow u]$  and so  $[v > s_1 \Rightarrow [t > s \Rightarrow u]] \succ [[v > s_1 \Rightarrow t \otimes \vec{y}] > s \otimes \vec{y} \Rightarrow u]$ . By the induction hypothesis, we are done.

**Term**  $[t > p \Rightarrow u]$  **with**  $p$  **a path** There is one possible last rule:

$$\frac{[\varepsilon \Rightarrow t] \xrightarrow{p} [\varepsilon' \Rightarrow t'] \quad [\varepsilon' \Rightarrow [t' > \vec{x} \Rightarrow u]] \xrightarrow{q} [\varepsilon'' \Rightarrow u']}{[\varepsilon \Rightarrow [t > p(\vec{x}) \Rightarrow u]] \xrightarrow{q} [\varepsilon'' \Rightarrow u']}$$

Let  $\vec{y}$  be the variables bound by  $\varepsilon$  that are not free in  $t$ . Since  $t$  is structurally smaller than  $[t > p \Rightarrow u]$ , we have  $[\varepsilon \Rightarrow [t > p \Rightarrow u]] \succ [\varepsilon \Rightarrow t]$  and so, by the induction hypothesis:  $(p \otimes \vec{y})^* [[\varepsilon \Rightarrow t \otimes \vec{y}]] \cong \Sigma_{d_t} [[\varepsilon' \Rightarrow t' \otimes \vec{y}]]$ . For any such  $\varepsilon', t'$  it follows from the proof of Lemma 3.10 that  $[\varepsilon \Rightarrow [t > p(\vec{x}) \Rightarrow u]] \succ [\varepsilon' \Rightarrow [t' > \vec{x} \Rightarrow u]]$  and so by the induction hypothesis

$$(q \otimes \vec{z})^* [[\varepsilon' \Rightarrow [t' > \vec{x} \Rightarrow u] \otimes r]] \cong \Sigma_{d_u} [[\varepsilon'' \Rightarrow u' \otimes r]]$$

for any  $r$  such that  $[\varepsilon \Rightarrow [t > p \Rightarrow u] \otimes r]$  is well-formed and closed. Now,

$$\begin{aligned} [\varepsilon \Rightarrow [t > p \Rightarrow u] \otimes r] & \cong [\varepsilon \Rightarrow [[t > y \Rightarrow y] > p \Rightarrow u] \otimes r] & (2.2), y \text{ fresh} \\ & \cong [\varepsilon \Rightarrow [t > y \Rightarrow [y > p \Rightarrow u]] \otimes r] & (2.4) \\ & \cong [\varepsilon \Rightarrow [t > y \Rightarrow [y > p \Rightarrow u] \otimes r]] & (2.3) \\ & \cong [[\varepsilon \Rightarrow t \otimes \vec{y}] > y \otimes \vec{y} \Rightarrow [y > p \Rightarrow u] \otimes r] & (2.4) \end{aligned}$$

Therefore,

$$\begin{aligned} (q \otimes \vec{z})^* [[\varepsilon \Rightarrow [t > p \Rightarrow u] \otimes r]] & \cong (q \otimes \vec{z})^* [[[\varepsilon \Rightarrow t \otimes \vec{y}] > y \otimes \vec{y} \Rightarrow [y > p \Rightarrow u] \otimes r]] \\ & \cong [[[\varepsilon \Rightarrow t \otimes \vec{y}] > y \otimes \vec{y} \Rightarrow (q \otimes \vec{z})^* [[y > p \Rightarrow u] \otimes r]]] & (3.6) \\ & \cong [[[\varepsilon \Rightarrow t \otimes \vec{y}] > y \otimes \vec{y} \Rightarrow q^* [[y > p \Rightarrow u]] \times_{\perp} \vec{z}^* [r]]] & (3.5) \end{aligned}$$

We may assume that the free variables of  $u$  and  $r$  are disjoint – possibly by renaming variables of  $p, u$ . The context  $q^* [-] \times_{\perp} \vec{z}^* [r]$  preserves all colimits as  $q$  is non-strict, and so, by Lemma 2.3, we can continue with

$$\begin{aligned} & \cong [[[\varepsilon \Rightarrow t \otimes \vec{y}] > y \otimes \vec{y} \Rightarrow [y > p \Rightarrow q^* [u] \times_{\perp} \vec{z}^* [r]]]] & (2.5) \\ & \cong [[[\varepsilon \Rightarrow t \otimes \vec{y}] > p \otimes \vec{y} \Rightarrow q^* [u] \times_{\perp} \vec{z}^* [r]]] & (2.5) \\ & \cong [[[\varepsilon \Rightarrow t \otimes \vec{y}] > p \otimes \vec{y} \Rightarrow (q \otimes \vec{z})^* [u \otimes r]]] & (3.5) \\ & \cong \Sigma_{\varepsilon', t'} [[\varepsilon' \Rightarrow t' \otimes \vec{y}] > \vec{x} \otimes \vec{y} \Rightarrow (q \otimes \vec{z})^* [u \otimes r]] & (3.7) \\ & \cong \Sigma_{d_t} (q \otimes \vec{z})^* [[[\varepsilon' \Rightarrow t' \otimes \vec{y}] > \vec{x} \otimes \vec{y} \Rightarrow u \otimes r]] & (2.4) \\ & \cong \Sigma_{d_t} (q \otimes \vec{z})^* [[\varepsilon' \Rightarrow [t' > \vec{x} \Rightarrow u \otimes r]]] & (2.4) \\ & \cong \Sigma_{d_t} (q \otimes \vec{z})^* [[\varepsilon' \Rightarrow [t' > \vec{x} \Rightarrow u] \otimes r]] & (2.3) \\ & \cong \Sigma_{d_t} \Sigma_{d_u} [[\varepsilon'' \Rightarrow u' \otimes r]] \end{aligned}$$

– as wanted.

The proof is complete.  $\square$

As for the restricted language of Section 3.2, we should be able to extend the above result to cover recursion. However, a more pressing issue is that to obtain an isomorphism between  $\text{els}[t]$  and the transition system obtained from the operational semantics, we need transitions with labels of the form  $p \otimes q$ , for  $p, q$  proper paths. Such transitions can be obtained by composing derived transitions with labels  $p \otimes y$  and  $x \otimes q$ . Consider a term  $[\varepsilon \Rightarrow t]$  of tensor type. Deriving both

$$[\varepsilon \Rightarrow t] \xrightarrow{p \otimes y} [\varepsilon_1 \Rightarrow t_1] \quad \text{and} \quad [\varepsilon_1 \Rightarrow t_1] \xrightarrow{x \otimes q} [\varepsilon' \Rightarrow t']$$

we can obtain a transition  $[\varepsilon \Rightarrow t] \xrightarrow{p \otimes q} [\varepsilon' \Rightarrow t']$ . Naturally, we ought to be able to obtain the same transition by taking the  $x \otimes q$  step first with intermediate state represented by a term  $[\varepsilon_2 \Rightarrow t_2]$ . How to give an operational account of when two such pairs of derivations should be identified is not so clearcut, but the fact that both combinations are possible and that either choice should lead to the same state suggests regarding the transitions  $[\varepsilon \Rightarrow t] \xrightarrow{p \otimes y} [\varepsilon_1 \Rightarrow t_1]$  and  $[\varepsilon \Rightarrow t] \xrightarrow{x \otimes q} [\varepsilon_2 \Rightarrow t_2]$  as “independent” or “concurrent”. In other words, it might be helpful to view processes definable in the metalanguage as event structures.

## 4 Event structures

This section reports on work in progress on interpreting both types and first-order terms of the tensor fragment using the prime event structures of [18], henceforth just called event structures. For simplicity, we use binary conflict rather than the more general consistency predicate of [18]. We may later need to revert to the general definition.

**Definition 4.1** An event structure is a triple  $A = (A, \leq, \smile)$  where  $A$  is a set of *events* upon which a partial order  $\leq$  of *causality* and a binary, symmetric, irreflexive relation  $\smile$  of *conflict* are defined. This data must satisfy

$$\begin{aligned} [e] &=_{\text{def}} \{e' \mid e' \leq e\} \text{ is finite for every } e \in A \\ e \smile e' \leq e'' &\Rightarrow e \smile e'' \text{ for all } e, e', e'' \in A \end{aligned}$$

A *configuration* of  $A$  is a finite subset  $a$  of  $A$  which is down-closed:  $\forall e \in a. [e] \subseteq a$  and consistent:  $\forall e, e' \in a. e \circ e'$ .<sup>2</sup> The partial order of configurations of  $A$  ordered by inclusion is written  $\mathcal{L}A$ . Two configurations  $a, a' \in \mathcal{L}A$  are *compatible*, written  $a \uparrow a'$ , if they have an upper bound in  $\mathcal{L}A$ .  $\square$

<sup>2</sup>We use Girard’s notation:  $\circ$  is the complement of  $\smile$ , the reflexive closure of which is written  $\smile$  with complement  $\frown$ . Specifying one relation clearly determines all of them.

Note that the empty set belongs to  $\mathcal{L}A$  as does the set  $[e]$  for each  $e \in A$ . Event structures can be assembled in a large cpo using a substructure ordering:

**Definition 4.2** Let  $A = (A, \leq_A, \smile_A)$  and  $B = (B, \leq_B, \smile_B)$  be event structures.  $A$  is a *substructure* of  $B$  if

$$\begin{aligned} A \subseteq B \quad \wedge \quad & \forall a \in A. [a]_A = [a]_B \\ & \wedge \quad \forall a, a'. a \smile_A a' \iff a, a' \in A \wedge a \smile_B a' \end{aligned}$$

In that case, we write  $A \trianglelefteq B$ . □

The  $\trianglelefteq$ -least event structure is the empty structure  $\emptyset = (\emptyset, \emptyset, \emptyset)$ . An  $\omega$ -chain  $A_0 \trianglelefteq A_1 \trianglelefteq \dots \trianglelefteq A_n \trianglelefteq \dots$  of event structures  $A_n = (A_n, \leq_n, \smile_n)$  has as least upper bound the event structure  $(\bigcup_n A_n, \bigcup_n \leq_n, \bigcup_n \smile_n)$ .

## 4.1 Constructions

We list some basic constructions on event structures:

**Lifting** Let  $P = (P, \leq_P, \smile_P)$  be an event structure. Then  $P_\perp$  is given by

$$\begin{aligned} \text{events} \quad & \{*\} \cup \{[e] : e \in P\} \\ \text{causality} \quad & * \leq e, \text{ all events } e, \text{ and } [e] \leq [e'] \iff e \leq_P e' \\ \text{conflict} \quad & * \circ e, \text{ all events } e, \text{ and } [e] \smile [e'] \iff e \smile_P e' \end{aligned}$$

We have  $\mathcal{L}(P_\perp) \cong (\mathcal{L}P)_\perp$ .

**Sum** Let  $A$  be a set and assume for each  $\alpha \in A$  that  $P_\alpha = (P_\alpha, \leq_\alpha, \smile_\alpha)$  is an event structure. Then  $\bigoplus_\alpha P_\alpha$  is given by

$$\begin{aligned} \text{events} \quad & \bigcup_{\alpha \in A} \{\alpha\} \times P_\alpha \\ \text{causality} \quad & (\alpha, e) \leq (\beta, e') \iff \alpha = \beta \wedge e \leq_\alpha e' \\ \text{conflict} \quad & (\alpha, e) \smile (\beta, e') \iff \alpha \neq \beta \vee (\alpha = \beta \wedge e \smile_\alpha e') \end{aligned}$$

The partial order  $\mathcal{L}(\bigoplus_\alpha P_\alpha)$  is the coalesced sum of the partial orders  $\mathcal{L}P_\alpha$  and the partial order  $\mathcal{L}(\bigoplus_\alpha P_{\alpha_\perp})$  is the lifting of the separated sum of the partial orders  $\mathcal{L}P_\alpha$ .

**Tensor** Let  $P_i = (P_i, \leq_i, \smile_i)$  for  $i = 1, 2$  be event structures. Then  $P_1 \otimes P_2$  is given by

$$\begin{aligned} \text{events} \quad & \{1\} \times P_1 \cup \{2\} \times P_2 \\ \text{causality} \quad & (i, e) \leq (j, e') \iff i = j \wedge e \leq_i e' \\ \text{conflict} \quad & (i, e) \smile (j, e') \iff i = j \wedge e \smile_i e' \end{aligned}$$

We have  $\mathcal{L}(P_1 \otimes P_2) \cong \mathcal{L}P_1 \times \mathcal{L}P_2$ .

**Fixpoint** Let  $F_1, \dots, F_n$  be operations on event structures that are continuous in each argument separately with regard to  $\sqsubseteq$ . Then the components of the simultaneous least fixed point  $\mu \vec{P}.(F_1 \vec{P}, \dots, F_n \vec{P})$  are event structures. Lifting, sum and tensor are all continuous in each argument separately.

Using the constructions above (specifically, using separated sum to represent the lifted sum constructor) we have defined an event-structure  $P$  for each type  $\mathbb{P}$  of the tensor fragment such that  $\mathcal{L}P \cong \mathbb{P}_\perp$ . We'll therefore confuse the configurations of  $P$  with objects  $P \in \mathbb{P}_\perp$ .

## 4.2 Representing profunctors

To represent the terms of the tensor fragment as well as the types, we need the notion of a strict morphism of event structures:

**Definition 4.3** Let  $A_1 = (A_1, \leq_1, \smile_1)$  and  $A_2 = (A_2, \leq_2, \smile_2)$  be two event structures. A *strict morphism*  $\eta : A_1 \rightarrow A_2$  is a function  $\eta : A_1 \rightarrow A_2$  that for all  $e, e' \in A_1$  satisfies  $\eta[e] = [\eta e]$  and  $e \smile_1 e' \Rightarrow \eta e \smile_2 \eta e'$ .  $\square$

**Lemma 4.4** Let  $\eta : A_1 \rightarrow A_2$  be a strict morphism. Then (i)  $\eta$  sends configurations of  $A_1$  to configurations of  $A_2$ , (ii)  $\eta$  is injective on configurations:  $\forall a_1 \in \mathcal{L}A_1. \forall e, e' \in a_1. \eta e = \eta e' \Rightarrow e = e'$ , and (iii) a smaller “output” configuration is obtained from a unique smaller “input” configuration:

$$\forall a_1 \in \mathcal{L}A_1. \forall a_2 \in \mathcal{L}A_2. a_2 \subseteq \eta a_1 \Rightarrow \exists! a'_1 \in \mathcal{L}A_1. a'_1 \subseteq a_1 \wedge \eta a'_1 = a_2.$$

**Proof:** (i) and (ii) are standard results, see [18]. As for (iii) suppose  $a_1 \in \mathcal{L}A_1$  and  $a_2 \in \mathcal{L}A_2$  with  $a_2 \subseteq \eta a_1$ . Consider the subset

$$a'_1 =_{\text{def}} \{e \in a_1 : \eta e \in a_2\}$$

of  $a_1$ . If  $e' \leq_1 e \in a'_1$ , then by definition,  $e \in a_1$  and  $\eta e \in a_2$ .  $a_1$  being a configuration implies  $e' \in a_1$  and from  $\eta[e] = [\eta e]$  we get  $\eta e' \in [\eta e] \subseteq a_2$ . So  $e' \in a'_1$  and  $a'_1$  is a down-closed subset of a configuration, and therefore a configuration itself. The equality  $\eta a'_1 = a_2$  follows from the definition of  $a'_1$  and the assumption  $a_2 \subseteq \eta a_1$ . To prove uniqueness, assume  $a''_1 \subseteq a_1$  is another configuration of  $A_1$ , satisfying  $\eta a''_1 = a_2$ . Then for each  $e' \in a'_1$ , there is an  $e'' \in a''_1$  such that  $\eta e' = \eta e''$ . But since both  $e'$  and  $e''$  belong to  $a_1$ , we get  $e' = e''$  by (ii). Hence,  $a'_1 \subseteq a''_1$  and by symmetry,  $a'_1 = a''_1$ .  $\square$

A strict morphism  $A \xrightarrow{\eta} Q$  to an event structure  $Q$  with  $\mathcal{L}Q \cong \mathbb{Q}_\perp$  induces a presheaf  $Y = Y(A, \eta) \in \widehat{\mathbb{Q}_\perp}$  as follows:

$$YQ = \{a \in \mathcal{L}A : \eta a = Q\} \quad \text{and} \quad Y(Q' \subseteq Q) : a \in YQ \mapsto a' \in YQ'$$

where  $a'$  is the unique configuration guaranteed to exist by Lemma 4.4, item (iii). Observe that  $Y$  is rooted since  $Y\emptyset = \{\emptyset\}$  (so  $Y$  can equally well be regarded as a presheaf over  $\mathbb{Q}$ ). There is an obvious isomorphism between  $\mathcal{L}A$  and  $\text{els } Y$ , the category of elements of  $Y$ . However, the exact class of presheaves represented in this way is as yet unknown to us.

Notice that presheaves  $Y \in \widehat{\mathbb{Q}_\perp}$  correspond to profunctors  $Y : \mathbb{O}_\perp \dashrightarrow \mathbb{Q}_\perp$ . To represent a general profunctor  $F : \mathbb{P}_\perp \dashrightarrow \mathbb{Q}_\perp$ , we might reasonably use a span  $\mathbb{P} \xleftarrow{\mu} A \xrightarrow{\eta} \mathbb{Q}$  where  $\eta$  is a strict morphism of event structures. But what should  $\mu$  do? Since the codomain of  $F$  is lifted,  $F$  can be seen as a process that can read more and more input, enabling more and more output. Following this intuition, more (consistent) output should demand more (consistent) input. So we take  $\mu$  to be a map  $\mu : A \rightarrow \mathcal{L}\mathbb{P}$  satisfying

$$\begin{aligned} e \leq_A e' &\Rightarrow \mu e \subseteq \mu e' \text{ for all } e, e' \in A \\ e \circ_A e' &\Rightarrow \mu e \uparrow_{\mathbb{P}} \mu e' \text{ for all } e, e' \in A \end{aligned}$$

$\mu$  can be extended to a map  $\mu^\dagger : \mathcal{L}A \rightarrow \mathcal{L}\mathbb{P}$  taking  $a$  to  $\bigcup \mu a$ . We now obtain a profunctor  $F = F(A, \mu, \eta) : \mathbb{P}_\perp \dashrightarrow \mathbb{Q}_\perp$  as follows:

$$\begin{aligned} F(P, Q) &= \{a \in \mathcal{L}A : \mu^\dagger a \subseteq P \wedge \eta a = Q\} \\ F(P \subseteq P', Q) &: a \in F(P, Q) \mapsto a \in F(P', Q) \\ F(P, Q' \subseteq Q) &: a \in F(P, Q) \mapsto a' \in F(P, Q') \end{aligned}$$

where again,  $a'$  is the unique configuration of  $A$  guaranteed to exist by Lemma 4.4, item (iii). It is easy to see that  $F$  thus defined is a profunctor. As for  $Y$  above, we can observe that for any  $P \in \mathcal{L}\mathbb{P}$ ,  $F(P, \emptyset) = \{\emptyset\}$ , meaning  $F$  is rooted and so corresponds to a profunctor  $\mathbb{P}_\perp \dashrightarrow \mathbb{Q}$  of the kind used to interpret terms of the metalanguage. The associated functor  $\bar{F} : \mathbb{P}_\perp \rightarrow \widehat{\mathbb{Q}}$  turns out to be *stable*, ie. pullback-preserving, and this property can be used to characterise the partial order  $\mathcal{L}A$  in terms of  $F$ :

**Proposition 4.5** *Let  $F$  be the profunctor  $\mathbb{P}_\perp \dashrightarrow \mathbb{Q}_\perp$  obtained in the way described from the event structure  $A$ . Then (i) the functor  $\bar{F} : \mathbb{P}_\perp \rightarrow \widehat{\mathbb{Q}}$  given as  $\bar{F}PQ = F(P, \lfloor Q \rfloor)$  preserves pullbacks and (ii)  $\mathcal{L}A$  is isomorphic to the category of elements of  $\int^{P \in \mathbb{P}_\perp} \bar{F}P$ .*

**Proof:** Since limits in presheaves are obtained pointwise, we only need to show that each pullback diagram of  $\mathbb{P}_\perp$  is sent by  $(\bar{F}-)Q$  to a pullback diagram in **Set** for all  $Q \in \mathbb{Q}$ . Now,  $(\bar{F}-)Q$  sends arrows of  $\mathbb{P}_\perp$  to injections in **Set** by definition of  $F$ . Notice also that the pullback object in  $\mathbb{P}_\perp$  is a binary intersection when viewed as a configuration of  $\mathbb{P}$ . (i) then follows easily.

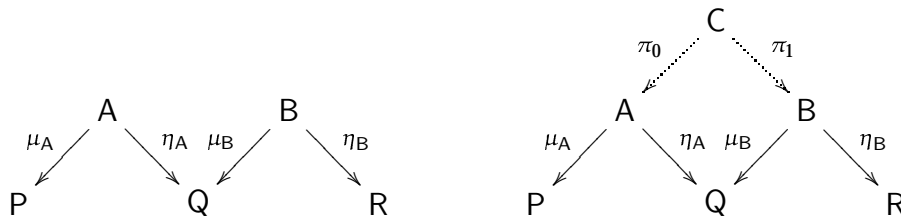
As for (ii), pullback-preservation and the fact that  $\mathbb{P}_\perp$  has all pullbacks implies that if  $a \in F(P, Q)$ , then there is a unique minimal  $P_0 \subseteq P$  such that  $a \in F(P_0, Q)$ . But this means that there is a canonical choice of representative for each element of the presheaf  $\int^{P \in \mathbb{P}_\perp} FP$ , namely the triple  $(a, P_0, Q)$ . They can therefore be viewed as the objects of  $\text{els}(\int^{P \in \mathbb{P}_\perp} FP)$  and the arrows then have the form  $(a', P'_0, Q') \rightarrow (a, P_0, Q)$  where  $(\int^{P \in \mathbb{P}_\perp} FP)(Q' \subseteq Q)a = a'$ , implying  $a' \subseteq a$ . Mapping the arrows to the implied inclusion in  $\mathcal{L}A$  provides one direction of the wanted isomorphism. In the other direction, we map  $a \in \mathcal{L}A$  to  $(a, P_0, Q)$  if  $\eta a = Q$  and where  $P_0$  is least such that  $a \in F(P_0, Q)$ . An inclusion  $a' \subseteq a$  is mapped to the corresponding arrow  $(a', P'_0, Q') \rightarrow (a, P_0, Q)$ , where  $\eta a' = Q' \subseteq Q$ .  $\square$

Again, the exact class of profunctors represented in this way is unknown to us at present.

For any pair  $P, Q$ , the large cpo of event structures induces a large cpo  $\mathbf{Spans}_{P, Q}$  with objects the class of spans  $P \xleftarrow{\mu} A \xrightarrow{\eta} Q$ , ordered by  $(A, \mu_A, \eta_A) \trianglelefteq_{P, Q} (B, \mu_B, \eta_B)$  if  $A \trianglelefteq B$  and the maps  $\mu_A, \eta_A$  are restrictions of the maps  $\mu_B, \eta_B$ .

### 4.3 Composition

Representing profunctors is not enough to give an event structure semantics for the metalanguage. We also need to compose those represented profunctors. We already know how to do that using coends, but then we are faced with the problem of how to represent the composition with an event structure. We therefore proceed to define composition directly on the representations. To compose the spans on the left below, it seems natural to “adjoin a pullback” as on the right:



In details, consider  $\eta_A : \mathcal{L}A \rightarrow \mathcal{L}Q$  and  $\mu_B : B \rightarrow \mathcal{L}Q$  as arrows in  $\mathbf{Set}$ . Their pullback is the set  $\{(a, e) \in \mathcal{L}A \times B : \eta_A a = \mu_B e\}$  with associated projections  $\pi_0 : (a, e) \mapsto a$  and  $\pi_1 : (a, e) \mapsto e$ . Following this, we define

an event structure  $C$  with

$$\begin{array}{ll} \text{events} & \{(a, e) \in \mathcal{L}A \times B : \eta_A a = \mu_B e\} \\ \text{causality} & (a, e) \leq (a', e') \iff a \subseteq a' \wedge e \leq_B e' \\ \text{conflict} & (a, e) \smile (a', e') \iff a \not\subseteq_A a' \vee e \smile_B e' \end{array}$$

– and obtain the wanted span  $P \xleftarrow{\mu_C} C \xrightarrow{\eta_C} R$  by

$$\begin{aligned} \mu_C(a, e) &= (\mu_A^\dagger \circ \pi_0)(a, e) = \mu_A^\dagger a \\ \eta_C(a, e) &= (\eta_B \circ \pi_1)(a, e) = \eta_B e \end{aligned}$$

We must then verify that  $(C, \mu_C, \eta_C)$  is indeed an event structure representation of a profunctor  $\mathbb{P}_\perp \dashv\vdash \mathbb{R}_\perp$ . This is elementary, but tedious, and deferred to Appendix G. The operation of composition is a map  $\mathbf{Spans}_{P,Q} \times \mathbf{Spans}_{Q,R} \rightarrow \mathbf{Spans}_{P,R}$  which can be shown to be continuous in each argument separately.

The definition of  $C$  implies that we can view the partial order  $\mathcal{L}C$  as given as the pullback object in the category of partial orders and monotone mappings of the two arrows  $\eta_A : \mathcal{L}A \rightarrow \mathcal{L}Q$  and  $\mu_B^\dagger : \mathcal{L}B \rightarrow \mathcal{L}Q$ , ie.

**Lemma 4.6** *The partial order  $\mathcal{L}C$  is isomorphic to  $\{(a, b) \in \mathcal{L}A \times \mathcal{L}B : \eta_A a = \mu_B^\dagger b\}$  with componentwise ordering.*

Let  $F = F(A, \mu_A, \eta_A)$ ,  $G = G(B, \mu_B, \eta_B)$  and  $H = H(C, \mu_C, \eta_C)$  be the profunctors represented by the three spans above. The obvious question is whether  $H$  is equivalent to  $G \circ F$ , defined using the coend

$$(G \circ F)(P, R) = \int^{Q \in \mathbb{Q}_\perp} F(P, Q) \times G(Q, R).$$

Via an understanding of coends in  $\mathbf{Set}$  and by stability of  $G$ , we can describe  $G \circ F$  as the bifunctor

$$(P, R) \mapsto \{(a, b) : a \in F(P, Q_0) \wedge b \in G(Q_0, R)\}$$

where  $Q_0$  is least among  $Q \in \mathbb{Q}_\perp$  such that  $b \in G(Q, R)$ . In other words,  $Q_0 = \mu_B^\dagger b$ . We can thus rewrite the bifunctor as

$$(P, R) \mapsto \{(a, b) : \mu_A^\dagger a \subseteq P \wedge \eta_A a = Q_0 = \mu_B^\dagger b \wedge \eta_B b = R\}$$

and therefore, by Lemma 4.6, it is isomorphic to

$$(P, R) \mapsto \{c \in \mathcal{L}C : \mu_C^\dagger c \subseteq P \wedge \eta_C c = R\}$$

which is just the definition of  $H$ .

It should be possible to define a bicategory  $\mathbf{ES}$  in which objects are event structures and arrows are spans of the above kind and prove it equivalent to a sub-bicategory of  $\mathbf{Prof}$ . This is left for future work; here, we'll give a new denotational semantics to the tensor fragment of the metalanguage by interpreting terms as spans.

## 4.4 A new denotational semantics

Assume the partial orders  $\mathbb{F}, \mathbb{P}$  are represented by event structures  $\Gamma, P$  such that  $\mathbb{F}_\perp \cong \mathcal{L}\Gamma$  and  $\mathbb{P}_\perp \cong \mathcal{L}P$ . We'll then interpret a typing judgment  $\mathbb{F} \vdash t : \mathbb{P}$  as a span  $\Gamma \xleftarrow{\mu} E_t \xrightarrow{\eta} P$  that represents a rooted profunctor  $\mathbb{F}_\perp \dashv\vdash \mathbb{P}_\perp$ , intended to be isomorphic to  $\lfloor t \rfloor$  (with  $t$  being, as usual, the profunctor  $\mathbb{F}_\perp \dashv\vdash \mathbb{P}$  interpreting  $\mathbb{F} \vdash t : \mathbb{P}$  according to the denotational semantics of Section 2).

**Identity** Consider the rule

$$\frac{}{x : \mathbb{P} \vdash x : \mathbb{P}}$$

Define  $E_{1_{\mathbb{P}}} = P = (P, \leq_P, \cup_P)$ ,  $\mu e = [e]$  and  $\eta e = e$ . So  $\mu^\dagger = 1_{\mathcal{L}P}$  and  $\eta = 1_P$ . This clearly defines a span

$$P \xleftarrow{\mu} E_x \xrightarrow{\eta} P$$

of the right kind. It represents a rooted profunctor  $F : \mathbb{P}_\perp \dashv\vdash \mathbb{P}_\perp$  given by

$$\begin{aligned} F(P_1, P_2) &= \{P \in \mathcal{L}P : \mu^\dagger P \subseteq P_1 \wedge \eta P = P_2\} \\ &\cong \{P \in \mathbb{P}_\perp : P \leq_{\mathbb{P}_\perp} P_1 \wedge P = P_2\} \\ &\cong \mathbb{P}_\perp(P_2, P_1) \\ &= \lfloor j_{\mathbb{P}_\perp} P_1 \rfloor P_2 \end{aligned}$$

as wanted.

**Sum** Consider the rule

$$\frac{\mathbb{F} \vdash t_i : \mathbb{P} \quad \text{all } i \in I}{\mathbb{F} \vdash \sum_{i \in I} t_i : \mathbb{P}}$$

Assuming that  $\lfloor t_i \rfloor$  is represented by the span  $\Gamma \xleftarrow{\mu_i} E_{t_i} \xrightarrow{\eta_i} P$  for each  $i \in I$ , we define  $E_{\sum_{i \in I} t_i} = \bigoplus_i E_{t_i}$ , and take  $\mu(i, e) = \mu_i e$  and  $\eta(i, e) = \eta_i e$ , for each  $i \in I$ . The span

$$\Gamma \xleftarrow{\mu} E_{\sum_{i \in I} t_i} \xrightarrow{\eta} P$$

is of the right form and represents a rooted profunctor  $F : \mathbb{F}_\perp \dashv\vdash \mathbb{P}_\perp$  given by

$$\begin{aligned} F(\gamma, \lfloor P \rfloor) &= \{c \in \mathcal{L}E_{\sum_{i \in I} t_i} : \mu^\dagger c \subseteq \gamma \wedge \eta c = \lfloor P \rfloor\} \\ &\cong \biguplus_{i \in I} \{c \in \mathcal{L}E_{t_i} : \mu_i^\dagger c \subseteq \gamma \wedge \eta_i c = \lfloor P \rfloor\} \\ &\cong \biguplus_{i \in I} \lfloor t_i \gamma \rfloor \lfloor P \rfloor \\ &\cong \sum_{i \in I} (t_i \gamma) P \\ &\cong \lfloor \sum_{i \in I} t_i \gamma \rfloor \lfloor P \rfloor \end{aligned}$$

as wanted.

**Tensor** Consider the rule

$$\frac{\Gamma_1 \vdash t_1 : \mathbb{P}_1 \quad \Gamma_2 \vdash t_2 : \mathbb{P}_2}{\Gamma_1, \Gamma_2 \vdash t_1 \otimes t_2 : \mathbb{P}_1 \otimes \mathbb{P}_2}$$

Assuming that  $\lfloor t_1 \rfloor$  and  $\lfloor t_2 \rfloor$  are represented by the spans

$$\Gamma_i \xleftarrow{\mu_i} E_{t_i} \xrightarrow{\eta_i} P_i \quad \text{for } i = 1, 2,$$

we define  $E_{t_1 \otimes t_2} = E_{t_1} \otimes E_{t_2}$ , and take  $\mu(i, e) = \text{in}_i(\mu_i e)$  and  $\eta(i, e) = (i, \eta_i e)$ , for  $i = 1, 2$ . The span

$$\Gamma_1 \otimes \Gamma_2 \xleftarrow{\mu} E_{t_1 \otimes t_2} \xrightarrow{\eta} P_1 \otimes P_2$$

is of the right form and represents a rooted profunctor  $F : (\Gamma \otimes \Delta)_\perp \dashv\dashv (\mathbb{P}_1 \otimes \mathbb{P}_2)_\perp$  given by

$$\begin{aligned} F((\gamma_1, \gamma_2), (P_1, P_2)) &= \{c \in \mathcal{L}E_{t_1 \otimes t_2} : \mu^\dagger c \subseteq (\gamma_1, \gamma_2) \wedge \eta c = (P_1, P_2)\} \\ &\cong \{(c_1, c_2) \in \mathcal{L}E_{t_1} \times \mathcal{L}E_{t_2} : \mu_i^\dagger c_i \subseteq \gamma_i \wedge \eta_i c_i = P_i, i = 1, 2\} \\ &\cong \prod_{i=1,2} \{c_i \in \mathcal{L}E_{t_i} : \mu_i^\dagger c_i \subseteq \gamma_i \wedge \eta_i c_i = P_i\} \\ &\cong \lfloor t_1 \gamma_1 \rfloor P_1 \times \lfloor t_2 \gamma_2 \rfloor P_2 \\ &= \lfloor t_1 \otimes t_2 (\gamma_1, \gamma_2) \rfloor (P_1, P_2) \end{aligned}$$

as wanted.

**Recursion** Consider the rule

$$\frac{\Gamma, x : \mathbb{P} \vdash t : \mathbb{P} \quad x, y \text{ not crossed in } t, \text{ any } y \text{ in } \Gamma}{\Gamma \vdash \text{rec } x.t : \mathbb{P}}$$

Assume that  $\lfloor t \rfloor$  is represented by the span  $\Gamma \otimes P \xleftarrow{\mu_t} E_t \xrightarrow{\eta_t} P$ . The arrow  $\sigma : \Gamma_\perp \rightarrow (\Gamma \otimes \Gamma)_\perp$  of **Cocont** used in the definition of  $\llbracket \text{rec } x.t \rrbracket$  in Section 2 is represented by the span

$$\Gamma \xleftarrow{\mu_\sigma} \Gamma \oplus \Gamma \xrightarrow{\eta_\sigma} \Gamma \otimes \Gamma$$

Here,  $\mu_\sigma(i, e) = \lceil e \rceil$  and  $\eta_\sigma(i, e) = (i, e)$  for  $i = 1, 2$ . For any span  $\Gamma \xleftarrow{\mu_u} E_u \xrightarrow{\eta_u} P$  representing the lifting of a profunctor  $u : \Gamma_\perp \dashv\dashv \mathbb{P}$ , we can employ the tensor construction above to obtain a span

$$\Gamma \otimes \Gamma \xleftarrow{\mu'} E_{1 \otimes u} \xrightarrow{\eta'} \Gamma \otimes P$$

representing  $\lfloor 1_\Gamma \otimes u \rfloor$ . Now, composing this span with the spans for  $t$  and  $\sigma$  yields a span representing  $\lfloor t \circ (1_\Gamma \otimes u) \circ \sigma \rfloor$ .

To obtain a representation of  $\lfloor \text{rec } x.t \rfloor$ , we just need a fixed point of this operation  $\phi_t$  on spans in the cpo **Spans** $_{\Gamma, P}$ . To make sense of this, we must verify that  $\phi_t$  is  $\triangleleft_{\Gamma, P}$ -continuous, but this follows from both composition and tensoring being continuous operations.

**Prefix** Consider the rule

$$\frac{\Gamma \vdash t : \mathbb{P}_a \quad \text{where } a \in A}{\Gamma \vdash a.t : \bigoplus_{\alpha} \mathbb{P}_{\alpha \perp}}$$

Assuming that  $[t]$  is represented by the span  $\Gamma \xleftarrow{\mu_t} E_t \xrightarrow{\eta_t} P_a$ , we define  $E_{a.t} = E_{t \perp}$  and take  $\mu^* = \emptyset$ ,  $\mu[e] = \mu_t e$  and  $\eta^* = (a, *)$ ,  $\eta[e] = (a, [\eta_t e])$ . The span

$$\Gamma \xleftarrow{\mu} E_{a.t} \xrightarrow{\eta} \bigoplus_{\alpha} P_{\alpha \perp}$$

represents a rooted profunctor  $F : \Gamma_{\perp} \dashv \rightarrow (\bigoplus_{\alpha} \mathbb{P}_{\alpha \perp})_{\perp}$  given by

$$\begin{aligned} F(\gamma, [\text{in}_a P]) &= \{c \in \mathcal{L}E_{a.t} : \mu^{\dagger} c \subseteq \gamma \wedge \eta c = [\text{in}_a P]\} \\ &\cong \{c \in \mathcal{L}E_t : \mu_t^{\dagger} c \subseteq \gamma \wedge \eta_t c = P\} \\ &\cong [t\gamma][P] \end{aligned}$$

as wanted.

**Strict match** Consider the rule

$$\frac{\Gamma \vdash t : \mathbb{P} \quad \Delta, \vec{x} : \mathbb{P} \vdash u : \mathbb{Q}}{\Delta, \Gamma \vdash [t > \vec{x} \Rightarrow u] : \mathbb{Q}}$$

Assume that  $[t]$  and  $[u]$  are represented by the spans

$$\Gamma \xleftarrow{\mu_t} E_t \xrightarrow{\eta_t} P \quad \text{and} \quad \Delta \otimes P \xleftarrow{\mu_u} E_u \xrightarrow{\eta_u} Q.$$

Using the representation of tensor from above, we can then construct a span

$$\Delta \otimes \Gamma \xleftarrow{\mu'} E_{1 \otimes t} \xrightarrow{\eta'} \Delta \otimes Q$$

representing  $[1_{\Delta} \otimes t]$ . So by composing with the span representing  $[u]$ , we get a representation of  $[u \circ (1_{\Delta} \otimes t)]$  as wanted.

**Paths** Let  $\sqcap \Vdash p : \mathbb{P}$  be a path and let  $\Pi$  and  $P$  be the event structures representing  $\sqcap$  and  $\mathbb{P}$ , respectively. Lifting  $\Pi$ , we obtain an event structure  $\Pi_{\perp}$  representing  $\sqcap_{\perp}$ , in which the events are  $*$  together with  $[e]$  for  $e$  an event of  $\Pi$ . The functor  $p : \Pi_{\perp} \rightarrow \mathbb{P}_{\perp}$  interpreting the path  $p$  induces a map  $\bar{p}$  on the events of  $\Pi_{\perp}$  given by  $*$   $\mapsto p\emptyset$  and  $[e] \mapsto p[e]$ . We can then construct a span

$$P \xleftarrow{\bar{p}} \Pi_{\perp} \xrightarrow{1} \Pi_{\perp}$$

which represents a rooted profunctor  $F : \mathbb{P}_\perp \dashv\vdash (\mathbb{P}_\perp)_\perp$  with

$$\begin{aligned} F(P, \lfloor Q \rfloor) &= \{Q' \in (\mathbb{P}_\perp)_\perp : \bar{p}^\dagger \lfloor Q \rfloor \subseteq P \wedge Q' = \lfloor Q \rfloor\} \\ &\cong \mathbb{P}_\perp(\bar{p}^\dagger \lfloor Q \rfloor, P) \\ &\cong \mathbb{P}_\perp(pQ, P) \\ &\cong p^*(P, Q) \end{aligned}$$

So  $F$  is isomorphic to  $\lfloor p^* \rfloor$  viewed as a profunctor  $\mathbb{P}_\perp \dashv\vdash (\mathbb{P}_\perp)_\perp$ . Notice that we cannot represent  $p^*$  directly with the spans we use, because  $p$  is non-strict, and so  $p^*$  is not rooted as a profunctor.

**Path match** Consider the rule

$$\frac{\Gamma \vdash t : \mathbb{P} \quad \Pi \Vdash p : \mathbb{P} \quad \Delta, \Pi \vdash u : \mathbb{Q}}{\Delta, \Gamma \vdash [t > p \Rightarrow u] : \mathbb{Q}}$$

Assume that  $\lfloor t \rfloor$ ,  $\lfloor p^* \rfloor$  and  $\lfloor u \rfloor$  are represented by the spans

$$\Gamma \xleftarrow{\mu_t} E_t \xrightarrow{\eta_t} P \quad P \xleftarrow{\bar{p}} \Pi_\perp \xrightarrow{1} \Pi_\perp \quad \Delta \otimes \Pi \xleftarrow{\mu_u} E_u \xrightarrow{\eta_u} Q.$$

By composing the spans for  $\lfloor t \rfloor$  and  $\lfloor p^* \rfloor$ , we obtain a span

$$\Gamma \xleftarrow{\mu_F} E_F \xrightarrow{\eta_F} \Pi_\perp$$

representing the profunctor  $F = \lfloor p^* \rfloor \circ \lfloor t \rfloor = \lfloor p^* \rfloor \lfloor t \rfloor$ . Consider the set  $X$  of events  $e$  of  $E_F$  satisfying  $\eta_F e = *$ . By strictness of  $\eta_F$  these events must be pairwise in conflict. Furthermore, any event  $e'$  of  $E_F$  causally depends on some  $e \in X$ . We can therefore split  $E_F$  into event structures  $E_e$  for  $e \in X$  by taking the events of  $E_e$  to be  $\{e' : e <_{E_F} e'\}$ . Notice that, by not including  $e$  into  $E_e$ ,  $\eta_F$  will map events of  $E_e$  to events of  $\Pi$ . Hence, restricting  $\mu_F$  and  $\eta_F$  to  $E_e$ , we obtain spans

$$\Gamma \xleftarrow{\mu_e} E_e \xrightarrow{\eta_e} \Pi$$

for  $e \in X$  each representing a different rooted component  $\lfloor t_e \rfloor$  of  $\lfloor p^* \rfloor \lfloor t \rfloor$ . Now, according to Lemma 3.7,

$$\lfloor t > p(\vec{x}) \Rightarrow u \rfloor \cong \sum_{e \in X} \lfloor t_e > \vec{x} \Rightarrow u \rfloor$$

and so we can represent  $\lfloor [t > p \Rightarrow u] \rfloor$  by using the representations above for sums and strict matches.

We expect to be able to show

**Conjecture 4.7** *Let  $\Gamma \vdash t : \mathbb{P}$  be a term of the tensor fragment and  $\Gamma \xleftarrow{\mu} E_t \xrightarrow{\eta} P$  the span that interprets it. Then the profunctor represented by the span is isomorphic to the profunctor interpreting  $\lfloor t \rfloor$  according to the denotational semantics of Section 2.*

## 5 Future work

The work described above both has some loose ends that it is planned to deal with in the near future, and also points out some directions for further research in the longer term.

It has already been noted that it is trivial to add products in the form of pairing and projection terms to the operational semantics, and that recursion can be handled in the proofs by using standard techniques of relating finite unfoldings of  $\text{rec } x.t$  to finite approximations to the fixpoint denoting that term. However, the operational semantics of Section 3 may well be replaced by one taking into account the observation of Section 4 that profunctors definable in the metalanguage are stable. This means that it might be possible to derive operationally for a given profunctor  $t$  the minimal input  $P$  needed for a certain output, represented by a path  $q$ . In other words, one would have derivations  $d$  with conclusions of the form  $t \xrightarrow{P \mapsto q} u$  where  $[u]$  represents the rooted component of  $q^*[t]$  corresponding to  $d$ . Notice that  $t$  and  $u$  are *open* terms and so environments should not be needed. Instead one can aim at a simple operational semantics of the kind presented on Page 14.

In Section 4 the profunctor  $\text{rec } x.t$  was represented by an event structure obtained as a suitable fixpoint. It could possibly also be done by intuitively “short-circuiting” the output with a part of the input. By such a construction, we seem to be able to represent the *trace* of non-deterministic dataflow, treated in [13] using stable profunctors of the same kind used here. This might allow a generalisation of the metalanguage by replacing recursion with a trace operator. A related line of work is on characterising the kind of profunctors representable by our event structure spans, leading to a study of the relationship between profunctors and event structures in general and as models for dataflow.

A main challenge will be to extend the development to higher order with a long term goal of obtaining a general operational understanding of presheaf models and open map bisimulation at higher order. Because of the mixed variance of function space, a reasonable starting point (in the light of Section 4) is to employ *bistructures* which are like event structures but with two partial orders,  $\leq^L$  (associated with input) and  $\leq^R$  (associated with output) instead of just causality [8]. Of course, this approach might not provide the wanted understanding of presheaf models, but one could then hope for some insights, indicating where to break away from presheaves if need be, in the pursuit of a domain theory for concurrency.

## A Formation rules and interpretations

The rules on the left are term formation rules, the rules on the right give the corresponding interpretations. In any environment that appears, all variables are assumed distinct.

### Identity, weakening and exchange

$$\overline{x : \mathbb{P} \vdash x : \mathbb{P}}$$

$$\overline{\mathbb{P} \xrightarrow{1} \mathbb{P}}$$

$$\frac{\Delta \vdash t : \mathbb{P}}{\Gamma, \Delta \vdash t : \mathbb{P}}$$

$$\frac{\Delta \xrightarrow{t} \mathbb{P}}{\Gamma \otimes \Delta \xrightarrow{0 \otimes t} \mathbb{O} \otimes \mathbb{P} \cong \mathbb{P}}$$

$$\frac{\Gamma, x : \mathbb{P}, y : \mathbb{Q}, \Delta \vdash t : \mathbb{R}}{\Gamma, y : \mathbb{Q}, x : \mathbb{P}, \Delta \vdash t : \mathbb{R}}$$

$$\frac{\Gamma \otimes \mathbb{P} \otimes \mathbb{Q} \otimes \Delta \xrightarrow{t} \mathbb{R}}{\Gamma \otimes \mathbb{Q} \otimes \mathbb{P} \otimes \Delta \cong \Gamma \otimes \mathbb{P} \otimes \mathbb{Q} \otimes \Delta \xrightarrow{t} \mathbb{R}}$$

### Recursive types

$$\overline{\Gamma \vdash t : \mathbb{T}_j[\mu \vec{P}. \vec{T} / \vec{P}]}$$

$$\Gamma \vdash t : \mu_j \vec{P}. \vec{T}$$

$$\Gamma \vdash t : \mu_j \vec{P}. \vec{T}$$

$$\overline{\Gamma \vdash t : \mathbb{T}_j[\mu \vec{P}. \vec{T} / \vec{P}]}$$

$$\left. \begin{array}{l} \\ \\ \end{array} \right\}$$

$$\frac{\Gamma \xrightarrow{t} \mu_j \vec{P}. \vec{T} = \mathbb{T}_j[\mu \vec{P}. \vec{T} / \vec{P}]}{\Gamma \xrightarrow{t} \mu_j \vec{P}. \vec{T} = \mathbb{T}_j[\mu \vec{P}. \vec{T} / \vec{P}]}$$

$$\frac{\Gamma \xrightarrow{t} \mu_j \vec{P}. \vec{T} = \mathbb{T}_j[\mu \vec{P}. \vec{T} / \vec{P}]}{\Gamma \xrightarrow{t} \mu_j \vec{P}. \vec{T} = \mathbb{T}_j[\mu \vec{P}. \vec{T} / \vec{P}]}$$

### Abstraction and application

$$\frac{\Gamma, x : \mathbb{P} \vdash t : \mathbb{Q}}{\Gamma \vdash \lambda x. t : \mathbb{P} \multimap \mathbb{Q}}$$

$$\frac{\Gamma \otimes \mathbb{P} \xrightarrow{t} \mathbb{Q}}{\Gamma \xrightarrow{\text{curry } t} \mathbb{P} \multimap \mathbb{Q}}$$

$$\frac{\Gamma \vdash t : \mathbb{P} \multimap \mathbb{Q} \quad \Delta \vdash u : \mathbb{P}}{\Gamma, \Delta \vdash t \cdot u : \mathbb{Q}}$$

$$\frac{\Gamma \xrightarrow{t} \mathbb{P} \multimap \mathbb{Q} \quad \Delta \xrightarrow{u} \mathbb{P}}{\Gamma \otimes \Delta \xrightarrow{(\text{uncurry } t) \circ (1_{\Gamma} \otimes u)} \mathbb{Q}}$$

### Recursion and sum

$$\frac{\Gamma, x : \mathbb{P} \vdash t : \mathbb{P}}{\Gamma \vdash \text{rec } x. t : \mathbb{P}}$$

$$\frac{\Gamma \otimes \mathbb{P} \xrightarrow{t} \mathbb{P}}{\Gamma \xrightarrow{\text{fix } \phi_t} \mathbb{P}}$$

$$\frac{\Gamma \vdash t_i : \mathbb{P} \quad \text{all } i \in I}{\Gamma \vdash \sum_{i \in I} t_i : \mathbb{P}}$$

$$\frac{\Gamma \xrightarrow{t_i} \mathbb{P} \quad \text{all } i \in I}{\Gamma \xrightarrow{\sum_{i \in I} t_i} \mathbb{P}}$$

In the rule for recursion,  $x, y$  must not be crossed in  $t$ , for any  $y$  in  $\Gamma$ .

### Prefix, product and tensor

$$\frac{\Gamma \vdash t : \mathbb{P}_a \quad \text{where } a \in A}{\Gamma \vdash \mathbf{a}.t : \bigoplus_{\alpha} \mathbb{P}_{\alpha \perp}} \quad \frac{\Gamma \xrightarrow{t} \mathbb{P}_a \quad \text{where } a \in A}{\Gamma \xrightarrow{\mathbf{a}.t} \bigoplus_{\alpha} \mathbb{P}_{\alpha \perp}}$$

$$\frac{\Gamma \vdash t : \mathbb{P} \quad \Gamma \vdash u : \mathbb{Q}}{\Gamma \vdash (t, u) : \mathbb{P} \& \mathbb{Q}} \quad \frac{\Gamma \xrightarrow{t} \mathbb{P} \quad \Gamma \xrightarrow{u} \mathbb{Q}}{\Gamma \xrightarrow{(t, u)} \mathbb{P} \& \mathbb{Q}}$$

$$\frac{\Gamma \vdash t : \mathbb{P} \quad \Delta \vdash u : \mathbb{Q}}{\Gamma, \Delta \vdash t \otimes u : \mathbb{P} \otimes \mathbb{Q}} \quad \frac{\Gamma \xrightarrow{t} \mathbb{P} \quad \Delta \xrightarrow{u} \mathbb{Q}}{\Gamma \otimes \Delta \xrightarrow{t \otimes u} \mathbb{P} \otimes \mathbb{Q}}$$

### Match

$$\frac{\Gamma \vdash t : \mathbb{P} \quad \Pi \Vdash p : \mathbb{P} \quad \Delta, \Pi \vdash u : \mathbb{Q}}{\Delta, \Gamma \vdash [t > p \Rightarrow u] : \mathbb{Q}}$$

$$\frac{\Gamma \xrightarrow{t} \mathbb{P} \quad \Pi_{\perp} \xrightarrow{p} \mathbb{P}_{\perp} \quad \Delta \otimes \Pi \xrightarrow{u} \mathbb{Q}}{\Delta \otimes \Gamma \xrightarrow{[t > p \Rightarrow u]} \mathbb{Q}}$$

### Patterns

$$\frac{}{\mathbf{x} : \mathbb{P} \Vdash \mathbf{x} : \mathbb{P}} \quad \frac{}{\mathbb{P}_{\perp} \xrightarrow{1} \mathbb{P}_{\perp}}$$

$$\frac{\Pi \Vdash p : \mathbb{P}_a \quad \text{where } a \in A}{\Pi \Vdash \mathbf{a}.p : \bigoplus_{\alpha} \mathbb{P}_{\alpha \perp}} \quad \frac{\Pi_{\perp} \xrightarrow{p} \mathbb{P}_{a \perp} \quad \text{where } a \in A}{\Pi_{\perp} \xrightarrow{\mathbf{a}.p} (\bigoplus_{\alpha} \mathbb{P}_{\alpha \perp})_{\perp}}$$

$$\frac{\Pi \Vdash p : \mathbb{P}}{\Pi \Vdash (p, -) : \mathbb{P} \& \mathbb{Q}} \quad \frac{\Pi_{\perp} \xrightarrow{p} \mathbb{P}_{\perp}}{\Pi_{\perp} \xrightarrow{(p, -)} (\mathbb{P} \& \mathbb{Q})_{\perp}}$$

$$\frac{\wedge \Vdash q : \mathbb{Q}}{\wedge \Vdash (-, q) : \mathbb{P} \& \mathbb{Q}} \quad \frac{\wedge_{\perp} \xrightarrow{q} \mathbb{Q}_{\perp}}{\wedge_{\perp} \xrightarrow{(-, q)} (\mathbb{P} \& \mathbb{Q})_{\perp}}$$

$$\frac{\Pi \Vdash p : \mathbb{P} \quad \wedge \Vdash q : \mathbb{Q}}{\Pi, \wedge \Vdash p \otimes q : \mathbb{P} \otimes \mathbb{Q}} \quad \frac{\Pi_{\perp} \xrightarrow{p} \mathbb{P}_{\perp} \quad \wedge_{\perp} \xrightarrow{q} \mathbb{Q}_{\perp}}{(\Pi \otimes \wedge)_{\perp} \xrightarrow{p \otimes q} (\mathbb{P} \otimes \mathbb{Q})_{\perp}}$$

## B Composition, extension and lifting

This appendix summarises material of [7]. A profunctor  $F : \mathbb{P} \dashv\vdash \mathbb{Q}$  is a bifunctor  $\mathbb{P} \times \mathbb{Q}^{\text{op}} \rightarrow \mathbf{Set}$ . By currying, it can be regarded also as a functor  $\mathbb{P} \rightarrow \widehat{\mathbb{Q}}$ . In any given context, we shall use the most convenient representation.

**Composition of profunctors** Given  $F : \mathbb{P} \dashrightarrow \mathbb{Q}$  and  $G : \mathbb{Q} \dashrightarrow \mathbb{R}$  we define their composition by

$$(G \circ F)(P, R) = \int^{Q \in \mathbb{Q}} F(P, Q) \times G(Q, R)$$

**Left Kan extension** A profunctor  $F : \mathbb{P} \dashrightarrow \mathbb{Q}$  extends via the Yoneda embedding  $y_{\mathbb{P}}$  to a functor  $F_{\dagger} : \widehat{\mathbb{P}} \rightarrow \widehat{\mathbb{Q}}$

$$F_{\dagger}X = \int^{P \in \mathbb{P}} XP \cdot FP$$

called the *left Kan extension of  $F$  along  $y_{\mathbb{P}}$* . It has a right adjoint  $F^* : \widehat{\mathbb{Q}} \rightarrow \widehat{\mathbb{P}}$ , given by

$$F^*Y = \widehat{\mathbb{Q}}(F-, Y).$$

$F^*$  is also obtained as the left Kan extension along  $y_{\mathbb{Q}}$  of the profunctor  $\mathbb{Q} \dashrightarrow \mathbb{P}$  that takes  $Q$  to  $\mathbb{Q}(F-, Q)$ . So  $F^*$  is a left adjoint and thus preserves arbitrary colimits.

**Lifting** As noted in the introduction,  $\widehat{\mathbb{P}}$  is characterised abstractly as the free colimit completion of  $\mathbb{P}$ . But  $\mathbb{P}$  can also be seen as the free *connected*-colimit completion of  $\mathbb{P}_{\perp}$  associated with the strict Yoneda embedding

$$j_{\mathbb{P}_{\perp}} : \mathbb{P}_{\perp} \rightarrow \widehat{\mathbb{P}} \quad \perp \mapsto \lambda P. \emptyset \quad [P] \mapsto y_{\mathbb{P}}P$$

We'll write  $\lfloor - \rfloor$  for the functor  $j_{\mathbb{P}_{\perp}}^*$ . It preserves connected colimits and so is a map  $\mathbb{P} \rightarrow \mathbb{P}_{\perp}$  of **Conn**.

If  $X$  is a presheaf over  $\mathbb{P}$ ,  $\lfloor X \rfloor$  maps  $\perp$  to a singleton and  $[P]$  to  $XP$  (via the Yoneda lemma), and so considering the category of elements of  $\lfloor X \rfloor$ , it is obtained from the category of elements of  $X$  by adding a new initial element. A presheaf such as  $X$  mapping  $\perp$  to a singleton is called *rooted*, and so is a profunctor  $\mathbb{Q} \dashrightarrow \mathbb{P}_{\perp}$  that maps all  $Q \in \mathbb{Q}$  to a rooted presheaf.

**Composition in Conn** Using the above, we obtain a way of composing arrows in **Conn**. Let  $F : \mathbb{P}_{\perp} \dashrightarrow \mathbb{Q}$  and  $G : \mathbb{Q}_{\perp} \dashrightarrow \mathbb{R}$  be the profunctors representing two composable arrows. Then, we can define a connected-colimit preserving functor by taking the left Kan extension of  $G$  along  $j_{\mathbb{Q}_{\perp}}$ :

$$G^{\dagger} : \widehat{\mathbb{Q}} \rightarrow \widehat{\mathbb{R}} \quad Y \mapsto \int^{Q \in \mathbb{Q}_{\perp}} \lfloor Y \rfloor Q \cdot GQ$$

so that the composition  $G \circ F$  in  $\mathbf{Prof}(\mathbb{P}_{\perp}, \mathbb{R}) \cong \mathbf{Conn}(\mathbb{P}, \mathbb{R})$  is given by the composition  $G^{\dagger} \circ F$  of functors.

## C Proofs of restructuring lemmas

**Proof of Lemma 2.4** Let  $\sigma \in \Sigma$ ,  $\delta \in \Delta_\perp$  and  $\gamma \in \Gamma_\perp$ . Then with  $P_1, P_2, Q$  ranging over  $\mathbb{P}_{1\perp}, \mathbb{P}_{2\perp}, \mathbb{A}_\perp$ , we can calculate as follows:

$$\begin{aligned} & [t > s \Rightarrow [u > q \Rightarrow v]](\sigma, \delta, \gamma) \\ & \cong \int^{P_1, P_2} [t\gamma](s(P_1, P_2)) \cdot \int^Q [u(\delta, P_1)](qQ) \cdot v(\sigma, P_2, Q) \\ & \cong \int^{P_1, P_2, Q} [t\gamma](s(P_1, P_2)) \cdot [u(\delta, P_1)](qQ) \cdot v(\sigma, P_2, Q) \\ & \cong \int^{P_1, P_2, Q} ([t\gamma](s(P_1, P_2)) \times [u(\delta, P_1)](qQ)) \cdot v(\sigma, P_2, Q) \end{aligned}$$

Consider now the presheaf  $XP_2 = [t\gamma](s(P_1, P_2))$  over  $\mathbb{P}_{2\perp}$ . It can be written as a colimit of representables:

$$X \cong \int^{(x, P'_2) \in \text{els } X} y_{\mathbb{P}_{2\perp}} P'_2$$

and so, since

$$y_{\mathbb{P}_{2\perp}} P'_2 \cong [j_{\mathbb{P}_{2\perp}} P'_2] = [\bar{y}P'_2],$$

the set  $[t\gamma](p(P_1, P_2)) \times [u(\delta, P_1)](qQ)$  from above can be understood as

$$\begin{aligned} & XP_2 \times [u(\delta, P_1)](qQ) \\ & \cong \left( \int^{(x, P'_2) \in \text{els } X} [\bar{y}P'_2] \right) P_2 \times [u(\delta, P_1)](qQ) \\ & \cong \int^{(x, P'_2) \in \text{els } X} [\bar{y}P'_2] P_2 \times [u(\delta, P_1)](qQ) \\ & \cong \int^{(x, P'_2) \in \text{els } X} [(u \otimes \bar{y})(\delta, P_1, P'_2)](qQ, P_2) \\ & \cong \int^{P'_2} XP'_2 \cdot [(u \otimes \bar{y})(\delta, P_1, P'_2)](qQ, P_2) \end{aligned}$$

We can now continue with the original calculation as follows:

$$\begin{aligned} & \cong \int^{P_1, P_2, Q} \left( \int^{P'_2} XP'_2 \cdot [(u \otimes \bar{y})(\delta, P_1, P'_2)](qQ, P_2) \right) \cdot v(\sigma, P_2, Q) \\ & \cong \int^{Q, P_2} \left( \int^{P_1, P'_2} XP'_2 \cdot [(u \otimes \bar{y})(\delta, P_1, P'_2)](qQ, P_2) \right) \cdot v(\sigma, P_2, Q) \\ & \cong \int^{Q, P_2} \left[ \int^{P_1, P'_2} XP'_2 \cdot (u \otimes \bar{y})(\delta, P_1, P'_2) \right](qQ, P_2) \cdot v(\sigma, P_2, Q) \\ & \cong \int^{Q, P_2} [[t > s \Rightarrow u \otimes \bar{y}](\gamma, \delta)](qQ, P_2) \cdot v(\sigma, P_2, Q) \\ & \cong [[t > s \Rightarrow u \otimes \bar{y}] > q \otimes \bar{y} \Rightarrow v](\sigma, \delta, \gamma) \end{aligned}$$

– as wanted.

**Proof of Lemma 2.5** Let  $\delta \in \Delta_{\perp}$  and  $\gamma \in \Gamma_{\perp}$ . If we let  $P_1, Q, P_2$  range over  $\mathbb{P}_{1\perp}, \mathbb{Q}_{\perp}, \mathbb{P}_{2\perp}$ , respectively, and  $Q'$  range over  $\mathbb{A}_{\perp}$ , we can calculate as follows:

$$\begin{aligned}
& [t > p \Rightarrow [x > q \Rightarrow u]](\delta, \gamma) \\
& \cong \int^{P_1, Q, P_2} [t\gamma](p(P_1, Q, P_2)) \cdot \int^{Q'} [j_{\mathbb{Q}_{\perp}} Q](qQ') \cdot u(\delta, P_1, Q', P_2) \\
& \cong \int^{P_1, Q', P_2} (\int^{Q'} [t\gamma](p(P_1, Q, P_2)) \cdot [j_{\mathbb{Q}_{\perp}} Q])(qQ') \cdot u(\delta, P_1, Q', P_2) \\
& \cong \int^{P_1, Q', P_2} (\int^{Q'} [t\gamma](p(P_1, Q, P_2)) \cdot y_{\mathbb{Q}_{\perp}} Q)(qQ') \cdot u(\delta, P_1, Q', P_2) \\
& \cong \int^{P_1, Q', P_2} ([t\gamma](p(P_1, -, P_2)))(qQ') \cdot u(\delta, P_1, Q', P_2) \\
& \cong \int^{P_1, Q', P_2} [t\gamma](p\{q/x\}(P_1, Q', P_2)) \cdot u(\delta, P_1, Q', P_2) \\
& \cong [t > p\{q/x\} \Rightarrow u](\delta, \gamma)
\end{aligned}$$

– as wanted.

## D Operational rules

The operational semantics for the tensor-fragment without recursion is given by the following rules:

$$\begin{array}{c}
\frac{\mathbf{v} \xrightarrow{s\{p/x\}} \mathbf{v}'}{[\mathbf{v} > \mathbf{s} \Rightarrow \mathbf{x}] \xrightarrow{p} [\mathbf{v}' > \mathbf{s} \Rightarrow \mathbf{x}]} \quad \frac{[\varepsilon \Rightarrow t_j] \xrightarrow{p} [\varepsilon' \Rightarrow t'] \quad j \in I}{[\varepsilon \Rightarrow \Sigma_{i \in I} t_i] \xrightarrow{p} [\varepsilon' \Rightarrow t']} \\
\frac{}{[\varepsilon \Rightarrow \mathbf{a}.t] \xrightarrow{\mathbf{a}.x} [\varepsilon \Rightarrow t]} \quad \frac{[\varepsilon \Rightarrow t] \xrightarrow{p} [\varepsilon' \Rightarrow t']}{[\varepsilon \Rightarrow \mathbf{a}.t] \xrightarrow{\mathbf{a}.p} [\varepsilon' \Rightarrow t']} \\
\frac{[\varepsilon \Rightarrow t] \xrightarrow{p} [\varepsilon' \Rightarrow t']}{[\varepsilon \Rightarrow t \otimes \mathbf{u}] \xrightarrow{p \otimes y} [\varepsilon' \Rightarrow t' \otimes \mathbf{u}]} \quad \frac{[\varepsilon \Rightarrow \mathbf{u}] \xrightarrow{q} [\varepsilon' \Rightarrow \mathbf{u}']}{[\varepsilon \Rightarrow t \otimes \mathbf{u}] \xrightarrow{x \otimes q} [\varepsilon' \Rightarrow t \otimes \mathbf{u}']} \\
\frac{[[\mathbf{v} > \mathbf{s}_1 \Rightarrow t \otimes \vec{y}] > \mathbf{s} \otimes \vec{y} \Rightarrow \mathbf{u}] \xrightarrow{q} [\mathbf{v}' > \mathbf{s}' \Rightarrow \mathbf{u}']}{[\mathbf{v} > \mathbf{s}_1 \Rightarrow [t > \mathbf{s} \Rightarrow \mathbf{u}]] \xrightarrow{q} [\mathbf{v}' > \mathbf{s}' \Rightarrow \mathbf{u}']} \\
\frac{[\varepsilon \Rightarrow t] \xrightarrow{p} [\varepsilon' \Rightarrow t'] \quad [\varepsilon' \Rightarrow [t' > \vec{x} \Rightarrow \mathbf{u}]] \xrightarrow{q} [\varepsilon'' \Rightarrow \mathbf{u}']}{[\varepsilon \Rightarrow [t > p(\vec{x}) \Rightarrow \mathbf{u}]] \xrightarrow{q} [\varepsilon'' \Rightarrow \mathbf{u}']}
\end{array}$$

Patterns  $s, s_1, s'$  are strict, whereas  $p, q$  are paths, having exactly one non-strict tensor-component. In the first rule it is assumed that the variables of  $p$  are disjoint from the variables of  $s$ . In the rule for strict match,  $\vec{y}$  are the variables of  $s_1$  that are not free in  $t$ . Renaming of the variables of  $s, u$  is implicitly used to ensure that the variables of  $s$  and  $\vec{y}$  are disjoint.

**Recursion** The obvious rule for recursion is

$$\frac{[\varepsilon \Rightarrow t\{\text{rec } x.t/x\}] \xrightarrow{p} [\varepsilon' \Rightarrow t']}{[\varepsilon \Rightarrow \text{rec } x.t] \xrightarrow{p} [\varepsilon' \Rightarrow t']}$$

but nothing has been proven about it.

## E Proof of type-correctness

The proof of Proposition 3.9 proceeds by rule induction using the induction hypothesis

$Q([\varepsilon \Rightarrow t])$ : Let  $r$  be any term such that  $\vdash [\varepsilon \Rightarrow t \otimes r] : \mathbb{P} \otimes \mathbb{R}$ . If  $[\varepsilon \Rightarrow t] \xrightarrow{p} [\varepsilon' \Rightarrow t']$  then  $p$  is a path with  $\sqcap \Vdash p : \mathbb{P}$  for some  $\sqcap$ , and  $\vdash [\varepsilon' \Rightarrow t' \otimes r] : \sqcap \otimes \mathbb{R}$ .

We look at each rule in turn:

**Variable** The rule is

$$\frac{v \xrightarrow{s\{p/x\}} v'}{[v > s \Rightarrow x] \xrightarrow{p} [v' > s \Rightarrow x]}$$

with the variables of  $p$  and  $s$  disjoint. Assume  $\vdash [v > s \Rightarrow x] : \mathbb{P}$  and let  $r$  be any term such that  $\vdash [v > s \Rightarrow x \otimes r] : \mathbb{P} \otimes \mathbb{R}$ . Then  $\vdash v : \mathbb{Q}$  and  $\wedge_1, x : \mathbb{P}, \wedge_2 \Vdash s : \mathbb{Q}$  for some  $\mathbb{Q}, \wedge_1, \wedge_2$ . By the induction hypothesis,  $\wedge \Vdash s\{p/x\} : \mathbb{Q}$  for some  $\wedge$ , but this means that  $\sqcap \Vdash p : \mathbb{P}$  for some  $\sqcap$  and  $\wedge \equiv \wedge_1, \sqcap, \wedge_2$ . By further use of the induction hypothesis,  $\vdash v' : \wedge$  and so, if we change the type of  $x$  to  $\sqcap$ , we get  $\wedge_1, x : \sqcap, \wedge_2 \Vdash s : \wedge$ , so that  $\vdash [v' > s \Rightarrow x \otimes r] : \sqcap \otimes \mathbb{R}$  as wanted.

**Sum** The rule is

$$\frac{[\varepsilon \Rightarrow t_j] \xrightarrow{p} [\varepsilon' \Rightarrow t'] \quad j \in I}{[\varepsilon \Rightarrow \sum_{j \in I} t_j] \xrightarrow{p} [\varepsilon' \Rightarrow t']}$$

Assume  $\vdash [\varepsilon \Rightarrow \sum_{j \in I} t_j] : \mathbb{P}$  and let  $r$  be any term such that  $\vdash [\varepsilon \Rightarrow \sum_{j \in I} t_j \otimes r] : \mathbb{P} \otimes \mathbb{R}$ . Then for each  $j \in I$ , we have  $\vdash [\varepsilon \Rightarrow t_j \otimes r] : \mathbb{P} \otimes \mathbb{R}$ , and so by the induction hypothesis,  $\sqcap \Vdash p : \mathbb{P}$  for some  $\sqcap$  and  $\vdash [\varepsilon' \Rightarrow t' \otimes r] : \sqcap \otimes \mathbb{R}$ , as wanted.

**Prefix** There are two rules:

$$\frac{}{[\varepsilon \Rightarrow a.t] \xrightarrow{a.x} [\varepsilon \Rightarrow t]} \quad \frac{[\varepsilon \Rightarrow t] \xrightarrow{p} [\varepsilon' \Rightarrow t']}{[\varepsilon \Rightarrow a.t] \xrightarrow{a.p} [\varepsilon' \Rightarrow t']}$$

Assume  $\vdash [\varepsilon \Rightarrow a.t] : \bigoplus_{\alpha} \mathbb{P}_{\alpha \perp}$  and let  $r$  be any term such that  $\vdash [\varepsilon \Rightarrow a.t \otimes r] : \bigoplus_{\alpha} \mathbb{P}_{\alpha \perp} \otimes \mathbb{R}$ .

In the former rule,  $a.x$  has type  $x : \mathbb{P}_a \Vdash a.x : \bigoplus_{\alpha} \mathbb{P}_{\alpha \perp}$  and  $\vdash [\varepsilon \Rightarrow t \otimes r] : \mathbb{P}_a \otimes \mathbb{R}$  as wanted.

In the latter rule,  $\vdash [\varepsilon \Rightarrow t \otimes r] : \mathbb{P}_a \otimes \mathbb{R}$ , so by the induction hypothesis,  $\sqcap \Vdash p : \mathbb{P}_a$  for some  $\sqcap$  and  $\vdash [\varepsilon' \Rightarrow t' \otimes r] : \sqcap \otimes \mathbb{R}$ . Since  $\sqcap \Vdash a.p : \bigoplus_{\alpha} \mathbb{P}_{\alpha \perp}$ , we are done.

**Tensor** There are two rules:

$$\frac{[\varepsilon \Rightarrow t] \xrightarrow{p} [\varepsilon' \Rightarrow t']}{[\varepsilon \Rightarrow t \otimes u] \xrightarrow{p \otimes y} [\varepsilon' \Rightarrow t' \otimes u]} \quad \frac{[\varepsilon \Rightarrow u] \xrightarrow{q} [\varepsilon' \Rightarrow u']}{[\varepsilon \Rightarrow t \otimes u] \xrightarrow{x \otimes q} [\varepsilon' \Rightarrow t \otimes u']}$$

Assume  $\vdash [\varepsilon \Rightarrow t \otimes u] : \mathbb{P} \otimes \mathbb{Q}$  and let  $r$  be any term such that  $\vdash [\varepsilon \Rightarrow (t \otimes u) \otimes r] : (\mathbb{P} \otimes \mathbb{Q}) \otimes \mathbb{R}$ . Then  $\vdash [\varepsilon \Rightarrow t \otimes (u \otimes r)] : \mathbb{P} \otimes (\mathbb{Q} \otimes \mathbb{R})$  so by the induction hypothesis,  $\sqcap \Vdash p : \mathbb{P}$  for some  $\sqcap$  and  $\vdash [\varepsilon' \Rightarrow t' \otimes (u \otimes r)] : \sqcap \otimes (\mathbb{Q} \otimes \mathbb{R})$ . But then  $\sqcap, y : \mathbb{Q} \Vdash p \otimes y : \mathbb{P} \otimes \mathbb{Q}$  and  $\vdash [\varepsilon' \Rightarrow (t' \otimes u) \otimes r] : (\sqcap \otimes \mathbb{Q}) \otimes \mathbb{R}$  as wanted. The proof for the other rule is symmetric.

**Strict match** The rule is:

$$\frac{[[v > s_1 \Rightarrow t \otimes \vec{y}] > s \otimes \vec{y} \Rightarrow u] \xrightarrow{q} [v' > s' \Rightarrow u']}{[v > s_1 \Rightarrow [t > s \Rightarrow u]] \xrightarrow{q} [v' > s' \Rightarrow u']}$$

with  $\vec{y}$  the variables of  $s_1$  that are not free in  $t$  and with the understanding that renaming of variables of  $s, u$  is done to prevent clashes in the pattern  $s \otimes \vec{y}$ . Assume

$$\vdash [v > s_1 \Rightarrow [t > s \Rightarrow u]] : \mathbb{Q}$$

and let  $r$  be any term such that

$$\vdash [v > s_1 \Rightarrow [t > s \Rightarrow u] \otimes r] : \mathbb{Q} \otimes \mathbb{R}.$$

By the renaming of variables of  $s, u$ , we may assume that  $r$  has no free variables also occurring in  $s$ . According to the remark following Lemma 2.3, this means that

$$\vdash [v > s_1 \Rightarrow [t > s \Rightarrow u \otimes r]] : \mathbb{Q} \otimes \mathbb{R}.$$

Lemma 2.4 then yields

$$\vdash [[v > s_1 \Rightarrow t \otimes \vec{y}] > s \otimes \vec{y} \Rightarrow u \otimes r] : \mathbb{Q} \otimes \mathbb{R}$$

and so, by the induction hypothesis,  $\wedge \Vdash q : \mathbb{Q}$  for some  $\wedge$  and  $\vdash [v' > s' \Rightarrow u' \otimes r] : \wedge \otimes \mathbb{R}$  as wanted.

**Path match** The rule is:

$$\frac{[\varepsilon \Rightarrow t] \xrightarrow{p} [\varepsilon' \Rightarrow t'] \quad [\varepsilon' \Rightarrow [t' > \vec{x} \Rightarrow u]] \xrightarrow{q} [\varepsilon'' \Rightarrow u']}{[\varepsilon \Rightarrow [t > p(\vec{x}) \Rightarrow u]] \xrightarrow{q} [\varepsilon'' \Rightarrow u']}$$

Assume  $\vdash [\varepsilon \Rightarrow [t > p \Rightarrow u]] : \mathbb{Q}$  and let  $r$  be any term such that

$$\vdash [\varepsilon \Rightarrow [t > p \Rightarrow u] \otimes r] : \mathbb{Q} \otimes \mathbb{R}.$$

Letting  $\vec{y}$  be the variables bound by  $\varepsilon$  that are not free in  $t$ , we have  $\vdash [\varepsilon \Rightarrow t \otimes \vec{y}] : \mathbb{P} \otimes \vec{\mathbb{P}}$  for some  $\vec{\mathbb{P}}$ . By the induction hypothesis, we therefore have  $\sqcap \Vdash p : \mathbb{P}$  for some  $\sqcap$  and  $[\varepsilon' \Rightarrow t' \otimes \vec{y}] : \sqcap \otimes \vec{\mathbb{P}}$ . But this means that

$$\vdash [\varepsilon' \Rightarrow [t' > \vec{x} \Rightarrow u] \otimes r] : \mathbb{Q} \otimes \mathbb{R}$$

and by further use of the induction hypothesis, we get  $\wedge \Vdash q : \mathbb{Q}$  for some  $\wedge$  and  $\vdash [\varepsilon'' \Rightarrow u' \otimes r] : \wedge \otimes \mathbb{R}$  as wanted.

By rule induction, the proof is complete.

## F Cases of the equivalence proof

The missing cases in the proof of Theorem 3.11 are listed below:

**Term**  $\Sigma_{i \in I} t_i$  There is one possible last rule:

$$\frac{[\varepsilon \Rightarrow t_j] \xrightarrow{p} [\varepsilon' \Rightarrow t'] \quad j \in I}{[\varepsilon \Rightarrow \Sigma_{i \in I} t_i] \xrightarrow{p} [\varepsilon' \Rightarrow t']}$$

Let  $r$  be any term such that  $[\varepsilon \Rightarrow \Sigma_{i \in I} t_i \otimes r]$  is well-formed and closed. For each  $j \in I$ , we may argue as follows:  $t_j$  is structurally smaller than  $\Sigma_{i \in I} t_i$  (because of our convention of not using the sum notation when  $I$  is a singleton). So  $[\varepsilon \Rightarrow \Sigma_{i \in I} t_i] \succ [\varepsilon \Rightarrow t_j]$  and therefore, by the induction hypothesis,

$$(p \otimes \vec{z})^* [[\varepsilon \Rightarrow t_j \otimes r]] \cong \Sigma_{d_j} [[\varepsilon' \Rightarrow t' \otimes r]].$$

We then get:

$$\begin{aligned}
& (\mathbf{p} \otimes \vec{\mathbf{z}})^* [[\varepsilon \Rightarrow \Sigma_{i \in I} \mathbf{t}_i \otimes r]] \\
& \cong [\varepsilon \Rightarrow \mathbf{p}^* [\Sigma_{i \in I} \mathbf{t}_i] \times_{\perp} \vec{\mathbf{z}}^* [r]] \quad (3.6), (3.5) \\
& \cong [\varepsilon \Rightarrow (\Sigma_{i \in I} \mathbf{p}^* [\mathbf{t}_i]) \times_{\perp} \vec{\mathbf{z}}^* [r]] \quad (3.3) \\
& \cong [\varepsilon \Rightarrow \Sigma_{i \in I} (\mathbf{p}^* [\mathbf{t}_i] \times_{\perp} \vec{\mathbf{z}}^* [r])] \\
& \cong \Sigma_{i \in I} (\mathbf{p} \otimes \vec{\mathbf{z}})^* [[\varepsilon \Rightarrow \mathbf{t}_i \otimes r]] \quad (3.5), (2.3) \\
& \cong \Sigma_{i \in I} \Sigma_{d_i} [[\varepsilon' \Rightarrow \mathbf{t}' \otimes r]]
\end{aligned}$$

– as wanted.

**Term  $a.t$  and path  $b.x$**  There is one possible last rule:

$$\frac{}{[\varepsilon \Rightarrow a.t] \xrightarrow{a.x} [\varepsilon \Rightarrow t]}$$

Let  $r$  be any term such that  $[\varepsilon \Rightarrow a.t \otimes r]$  is well-formed and closed. Then,

$$\begin{aligned}
& (\mathbf{b}.x \otimes \vec{\mathbf{z}})^* [[\varepsilon \Rightarrow a.t \otimes r]] \\
& \cong [\varepsilon \Rightarrow \mathbf{b}.x^* [a.t] \times_{\perp} \vec{\mathbf{z}}^* [r]] \quad (3.6), (3.5) \\
& \cong \begin{cases} [\varepsilon \Rightarrow \mathbf{x}^* [t] \times_{\perp} \vec{\mathbf{z}}^* [r]] & \text{if } a \equiv b \\ [\varepsilon \Rightarrow \emptyset \times_{\perp} [r]] & \text{if } a \not\equiv b \end{cases} \quad (3.4) \\
& \cong \begin{cases} [\varepsilon \Rightarrow [t] \times_{\perp} [r]] & \text{if } a \equiv b \\ [\varepsilon \Rightarrow \emptyset \times_{\perp} [r]] & \text{if } a \not\equiv b \end{cases} \quad (3.2) \\
& \cong \begin{cases} [[\varepsilon \Rightarrow t \otimes r]] & \text{if } a \equiv b \\ \emptyset & \text{if } a \not\equiv b \end{cases} \quad (3.5), (2.3)
\end{aligned}$$

– as wanted.

**Term  $a.t$  and path  $b.p$**  There is one possible last rule:

$$\frac{[\varepsilon \Rightarrow t] \xrightarrow{p} [\varepsilon' \Rightarrow t']}{[\varepsilon \Rightarrow a.t] \xrightarrow{a.p} [\varepsilon' \Rightarrow t']}$$

Let  $r$  be any term such that  $[\varepsilon \Rightarrow a.t \otimes r]$  is well-formed and closed. Since  $t$  is structurally smaller than  $a.t$ , we have  $[\varepsilon \Rightarrow a.t] \succ [\varepsilon \Rightarrow t]$ . Hence, by the induction hypothesis,

$$(\mathbf{p} \otimes \vec{\mathbf{z}})^* [[\varepsilon \Rightarrow t \otimes r]] \cong \Sigma_{d_t} [[\varepsilon' \Rightarrow t' \otimes r]].$$

We now get:

$$\begin{aligned}
& (\mathbf{b.p} \otimes \bar{\mathbf{z}})^* [[\varepsilon \Rightarrow \mathbf{a.t} \otimes \mathbf{r}]] \\
& \cong [\varepsilon \Rightarrow \mathbf{b.p}^* [\mathbf{a.t}] \times_{\perp} \bar{\mathbf{z}}^* [\mathbf{r}]] \quad (3.6), (3.5) \\
& \cong \begin{cases} [\varepsilon \Rightarrow \mathbf{p}^* [\mathbf{t}] \times_{\perp} \bar{\mathbf{z}}^* [\mathbf{r}]] & \text{if } \mathbf{a} \equiv \mathbf{b} \\ [\varepsilon \Rightarrow \emptyset \times_{\perp} \bar{\mathbf{z}}^* [\mathbf{r}]] & \text{if } \mathbf{a} \not\equiv \mathbf{b} \end{cases} \quad (3.4) \\
& \cong \begin{cases} (\mathbf{p} \otimes \bar{\mathbf{z}})^* [[\varepsilon \Rightarrow \mathbf{t} \otimes \mathbf{r}]] & \text{if } \mathbf{a} \equiv \mathbf{b} \\ \emptyset & \text{if } \mathbf{a} \not\equiv \mathbf{b} \end{cases} \quad (3.5), (2.3) \\
& \cong \begin{cases} \Sigma_{d_t} [[\varepsilon' \Rightarrow \mathbf{t}' \otimes \mathbf{r}]] & \text{if } \mathbf{a} \equiv \mathbf{b} \\ \emptyset & \text{if } \mathbf{a} \not\equiv \mathbf{b} \end{cases}
\end{aligned}$$

– as wanted.

**Term  $t \otimes u$  and path  $p \otimes y$**  There is one possible last rule:

$$\frac{[\varepsilon \Rightarrow \mathbf{t}] \xrightarrow{p} [\varepsilon' \Rightarrow \mathbf{t}']}{[\varepsilon \Rightarrow \mathbf{t} \otimes \mathbf{u}] \xrightarrow{p \otimes y} [\varepsilon' \Rightarrow \mathbf{t}' \otimes \mathbf{u}]}$$

Let  $r$  be any term such that  $[\varepsilon \Rightarrow (\mathbf{t} \otimes \mathbf{u}) \otimes \mathbf{r}]$  is well-formed and closed. Since  $t$  is structurally smaller than  $t \otimes u$ , we have  $[\varepsilon \Rightarrow \mathbf{t} \otimes \mathbf{u}] \succ [\varepsilon \Rightarrow \mathbf{t}]$ . Therefore, by the induction hypothesis,

$$(\mathbf{p} \otimes \bar{\mathbf{z}})^* [[\varepsilon \Rightarrow \mathbf{t} \otimes (\mathbf{u} \otimes \mathbf{r})]] \cong \Sigma_{d_t} [[\varepsilon' \Rightarrow \mathbf{t}' \otimes (\mathbf{u} \otimes \mathbf{r})]].$$

We then have:

$$\begin{aligned}
& ((\mathbf{p} \otimes \mathbf{y}) \otimes \bar{\mathbf{z}})^* [[\varepsilon \Rightarrow (\mathbf{t} \otimes \mathbf{u}) \otimes \mathbf{r}]] \\
& \cong [\varepsilon \Rightarrow (\mathbf{p}^* [\mathbf{t}] \times_{\perp} \mathbf{y}^* [\mathbf{u}]) \times_{\perp} \bar{\mathbf{z}}^* [\mathbf{r}]] \quad (3.6), (3.5) \\
& \cong [\varepsilon \Rightarrow \mathbf{p}^* [\mathbf{t}] \times_{\perp} (\mathbf{y}^* [\mathbf{u}] \times_{\perp} \bar{\mathbf{z}}^* [\mathbf{r}])] \\
& \cong (\mathbf{p} \otimes \bar{\mathbf{z}})^* [[\varepsilon \Rightarrow \mathbf{t} \otimes (\mathbf{u} \otimes \mathbf{r})]] \quad (3.6), (3.5) \\
& \cong \Sigma_{d_t} [[\varepsilon \Rightarrow \mathbf{t}' \otimes (\mathbf{u} \otimes \mathbf{r})]] \\
& \cong \Sigma_{d_t} [[\varepsilon \Rightarrow (\mathbf{t}' \otimes \mathbf{u}) \otimes \mathbf{r}]]
\end{aligned}$$

– as wanted. The case with path  $x \otimes q$  is symmetric.

## G Representation of composition

The following verifies that the span  $P \xleftarrow{\mu_C} C \xrightarrow{\eta_C} R$  is a representation of a profunctor as wanted:

**C is an event structure** The relation  $\leq_C$  is a partial order since both  $\subseteq$  and  $\leq_B$  are, and the relation  $\smile_C$  is symmetric and irreflexive since both  $\not\smile_A$  and  $\smile_B$  are. Finiteness of  $a \in \mathcal{L}A$  and  $\lceil e \rceil \in \mathcal{L}B$  implies finiteness of  $\lceil (a, e) \rceil$  for all  $(a, e) \in C$ . Finally, if  $(a, e) \smile_C (a', e') \leq_C (a'', e'')$  then either  $a \not\smile_A a' \subseteq a''$  implying  $a \not\smile_A a''$  or  $e \smile_B e' \leq_B e''$  implying  $e \smile_B e''$ . So, in either case,  $(a, e) \smile_C (a'', e'')$  as wanted.

**$\mu_C$  is of the right form** If  $(a, e) \leq_C (a', e')$ , then  $a \subseteq a'$  and so clearly  $\mu_C(a, e) = \mu_A^\dagger a \subseteq \mu_A^\dagger a' = \mu_C(a', e')$ . If  $(a, e) \circ_C (a', e')$ , then by definition,  $a \uparrow_A a'$  so that events of  $a$  and  $a'$  are pairwise consistent. The map  $\mu_A$  sends such pairs of events to compatible configurations of  $P$  and it follows that  $\mu_A^\dagger a \uparrow_P \mu_A^\dagger a'$ .

**$\eta_C$  is a strict morphism** Let  $(a, e) \in C$ . We'll first show that  $\lceil \eta_C(a, e) \rceil \subseteq \eta_C \lceil (a, e) \rceil$ . So let  $\bar{e} \in \lceil \eta_C(a, e) \rceil$  which equals  $\lceil \eta_B e \rceil$  by definition and  $\eta_B \lceil e \rceil$  because  $\eta_B$  is strict. Therefore, there is an  $e' \leq_B e$  such that  $\eta_B e' = \bar{e}$ . The map  $\mu_B$  sends  $e' \leq_B e$  to configurations  $\mu_B e' \subseteq \mu_B e = \eta_A a$  of  $Q$ , and so applying Lemma 4.4, item (iii) to  $\eta_A$ , there is a (unique) configuration  $a' \subseteq a$  of  $\mathcal{L}A$  such that  $\eta_A a' = \mu_B e'$ . In other words,  $(a', e') \in C$  and since  $(a', e') \leq_C (a, e)$  we get  $\bar{e} = \eta_B e' = \eta_C(a', e') \in \eta_C \lceil (a, e) \rceil$  as wanted.

For the converse, suppose  $\bar{e} \in \eta_C \lceil (a, e) \rceil$ . Then  $\bar{e} = \eta_C(a', e')$  for some  $(a', e') \leq_C (a, e)$ . We need to show that  $\eta_C(a', e') = \eta_B e' \leq_R \eta_B e = \eta_C(a, e)$ . But this follows from  $e' \leq_B e$  and the fact that  $\eta_B$  is strict.

Finally, we must show that for all  $(a, e), (a', e') \in C$ ,  $(a, e) \frown_C (a', e')$  implies  $\eta_C(a, e) \frown_R \eta_C(a', e')$ . Assuming  $(a, e) \frown_C (a', e')$ , there are two cases. If  $e \neq e'$ , then we have  $e \frown_B e'$  so that, by strictness of  $\eta_B$ , we get  $\eta_C(a, e) = \eta_B e \frown_R \eta_B e' = \eta_C(a', e')$ . On the other hand, if  $e = e'$  we must have  $a \neq a'$  and  $a \uparrow_A a'$ . Let  $a'' \in \mathcal{L}A$  be a superset to both. Since  $(a, e), (a', e) \in C$ , we have  $\eta_A a = \eta_A a' = \mu_B e$ , but  $\eta_A$  is strict and so, according to Lemma 4.4, item (iii), there is a *unique* configuration included in  $a''$  whose image under  $\eta_A$  is  $\mu_B e$ . We have a contradiction with  $a \neq a'$  and we are done.

## References

- [1] Francis Borceaux. *Handbook of Categorical Algebra I*, volume 50 of *Encyclopedia of Mathematics and its Applications*. Cambridge University Press, 1994.
- [2] Mario Caccamo and Glynn Winskel. Lecture notes in category theory. Notes for a course given by Glynn Winskel based on Martin Hyland's [14]. Available at <http://www.brics.dk/~mcaccamo/>, 2000.
- [3] Gian Luca Cattani. *Presheaf Models for Concurrency*. PhD thesis, University of Aarhus, 1999.
- [4] Gian Luca Cattani, Marcelo P. Fiore, and Glynn Winskel. A theory of recursive domains with applications to concurrency. In *LICS'98, Proceedings of the Thirteenth Annual IEEE Symposium on Logic in Computer Science*, pages 214–225. IEEE Computer Society Press, 1998.
- [5] Gian Luca Cattani, Ian Stark, and Glynn Winskel. Presheaf models for the  $\pi$ -calculus. In *CTCS'97, Proceedings of the 7th International Conference on Category Theory and Computer Science*, number 1290 in Lecture Notes in Computer Science, pages 106–126. Springer-Verlag, 1997.
- [6] Gian Luca Cattani and Glynn Winskel. Presheaf models for concurrency. In *Proceedings of CSL'96*, number 1258 in Lecture Notes in Computer Science, pages 58–75. Springer-Verlag, 1997.
- [7] Gian Luca Cattani and Glynn Winskel. Profunctors, open maps and bisimulation. Manuscript, 2000.
- [8] Pierre-Louis Curien, Gordon Plotkin, and Glynn Winskel. *Bistructures, Bidomains, and Linear Logic*, chapter 1, pages 21–54. Foundations of Computing Series. The MIT Press, 2000.
- [9] Marcelo Fiore, Gian Luca Cattani, and Glynn Winskel. Weak bisimulation and open maps (extended abstract). In *LICS'99, Proceedings of the Fourteenth Annual IEEE Symposium on Logic in Computer Science*, pages 67–76. IEEE Computer Society Press, 1999.
- [10] Jean-Yves Girard. Linear logic. *Theoretical Computer Science*, 50:1–102, 1987.
- [11] Thomas T. Hildebrandt. *Categorical Models for Concurrency: Independence, Fairness and Dataflow*. PhD thesis, University of Aarhus, 1999.

- [12] Thomas T. Hildebrandt. A fully abstract presheaf semantics for SCCS with finite delay. In *CTCS'99: Proceedings of the 8th International Conference on Category Theory and Computer Science*, number 29 in Electronic Notes in Theoretical Computer Science. Elsevier, 1999.
- [13] Thomas T. Hildebrandt, Prakash Panangaden, and Glynn Winskel. Relational semantics of non-deterministic dataflow. In *CONCUR'98: Proceedings of the 9th International Conference on Concurrency Theory*, number 1466 in Lecture Notes in Computer Science, pages 613–628. Springer-Verlag, 1998.
- [14] Martin Hyland. Category theory. Lecture notes taken by Cocky Hillhorst of Martin Hyland's course, 1995.
- [15] André Joyal, Mogens Nielsen, and Glynn Winskel. Bisimulation from open maps. *Information and Computation*, 127(2):164–185, June 1996.
- [16] Gordon D. Plotkin. A structural approach to operational semantics. DAIMI FN 19, Computer Science Department, Aarhus University, 1981.
- [17] Gordon D. Plotkin. Domains. Lecture notes on domain theory (the "Pisa notes"). L<sup>A</sup>T<sub>E</sub>X-version by Yugo Kashiwagi and Hidetaka Kondoh, November 1992, 1983.
- [18] Glynn Winskel. *Event structures*, pages 325–392. Number 255 in Lecture Notes in Computer Science. Springer Verlag, 1987.
- [19] Glynn Winskel. *The Formal Semantics of Programming Languages: An Introduction*. Foundations of Computing Series. The MIT Press, 1993. Second printing, 1994.
- [20] Glynn Winskel. A presheaf semantics of value-passing processes (extended abstract). In *CONCUR'96, Proceedings of the 7th International Conference on Concurrency Theory*, number 1119 in Lecture Notes in Computer Science, pages 98–114. Springer-Verlag, 1996.
- [21] Glynn Winskel. A linear metalanguage for concurrency. In *Algebraic Methodology and Software Technology: 7th International Conference, AMAST '98: Proceedings*, number 1548 in Lecture Notes in Computer Science, pages 42–58. Springer-Verlag, 1998.