Under consideration for publication in Math. Struct. in Comp. Science

A Relational Model of Non-Deterministic Dataflow

Thomas T. Hildebrandt[1], Prakash Panangaden[2] and Glynn Winskel[3]

[1] *IT University of Copenhagen, Denmark,*
[2] *McGill University, Montreal, Canada,*
[3] *University of Cambridge, UK.*

Received 8 May 2003

We recast dataflow in a modern categorical light using profunctors as a generalisation of relations. The well known causal anomalies associated with relational semantics of indeterminate dataflow are avoided, but still we preserve much of the intuitions of a relational model. The development fits with the view of categories of models for concurrency and the general treatment of bisimulation they provide. In particular it fits with the recent categorical formulation of feedback using traced monoidal categories. The payoffs are: (1) explicit relations to existing models and semantics, especially the usual axioms of monotone IO automata are read off from the definition of profunctors, (2) a new definition of bisimulation for dataflow, the proof of the congruence of which benefits from the preservation properties associated with open maps and (3) a treatment of higher-order dataflow as a biproduct, essentially by following the geometry of interaction programme.

Introduction

A fundamental dichotomy in concurrency is the distinction between *asynchronous* communication and *synchronous* communication. In the present paper we unify the analysis of these situations in the framework of a categorical presentation of models for concurrency as initiated by Winskel and Nielsen (Winskel and Nielsen, 1995). In particular we have given a treatment of indeterminate dataflow networks in terms of (a special kind of) profunctors which is very close to the treatment of synchronous communication. This new semantical treatment has a number of benefits

1. the general functoriality and naturality properties of presheaves *automatically* imply the usually postulated axioms for asynchronous, monotone automata (Panangaden and Stark, 1988; Selinger, 1997)
2. we get a notion of bisimulation, which is crucial when allowing synchronous, CCS-like primitives,
3. it is both closely connected to the extant models (Jonsson, 1989) expressed in terms of trace sets, and a direct generalisation of Kahn's model of determinate dataflow (Kahn, 1974), thus providing a relational viewpoint which allows one to think of composing network components as a (kind of) relational composition,
4. it gives a semantics of higher-order networks almost for "free" by using the passage from traced monoidal categories to compact-closed categories (Abramsky, 1996; Joyal, Street and Verity, 1996b) (the "geometry of interaction" construction).

The categorical presentation is critical for all these points. Without the realization that Kahn processes can be described as a traced monoidal category and knowledge of the results in (Abramsky, 1996; Joyal, Street and Verity, 1996b) it would be hard to see how one could have proposed our model of higher-order processes. It is notable that the profunctor semantics of dataflow yields automatically the axioms for monotone port automata used in modelling dataflow (Panangaden and Stark, 1988) in contrast to the work in (Stark, 1989a). At the same time we have to work to get a correct operation on profunctors to model the dataflow feedback; "the obvious" choice of modelling feedback by coend doesn't account for the subtle causal constraints which plague dataflow semantics.

The background for this paper includes work done on presenting models for concurrency as categories, as summarised in (Winskel and Nielsen, 1995). This enabled a sweeping definition of bisimulation based on open maps applicable to any category of models equipped with a distinguished subcategory of paths (Joyal, Nielsen and Winskel, 1996a). It also exposed a new space of models: presheaves. Presheaf categories possess a canonical choice of open maps and bisimulation, and can themselves be related in the bicategory of profunctors. This yields a form of domain theory but boosted to the level of using categories rather than partial orders as the appropriate domains.

One argument for the definition of bisimulation based on open maps is the powerful preservation properties associated with it. Notable is the result of (Cattani and Winskel, 1997) that any colimit preserving functor between presheaf categories preserves bisimulation, which besides obvious uses in relating semantics in different models with different notions of bisimulation is, along with several other general results, useful in establishing congruence properties of process languages. By understanding dataflow in terms of profunctors we are able to exploit the framework not just to give a definition of bisimulation between dataflow networks but also in showing it to be a congruence with respect to the standard operations of dataflow.

A difficulty has been in understanding the operational significance of the bisimulation which comes from open maps for higher-order process languages (where for example processes themselves can be passed as values). Another gap, more open and so more difficult to approach, is that whereas both interleaving models and independence models like event structures can be recast as presheaf models, as soon as higher-order features appear, the presheaf semantics at present reduce concurrency to nondeterministic interleaving. A study of nondeterministic dataflow is helpful here as its compositional models are forced to account for causal dependency using ideas familiar from independence models; at the same time the models are a step towards understanding higher-order as they represent nondeterministic functions from input to output.

The idea that non-deterministic dataflow can be modelled by some kind of generalised relations fits with that of others, notably Stark in (Stark, 1989a; Stark, 1989b; Stark, 1998). Bisimulation for dataflow is studied in (Stark, 1992). That dataflow should fit within a categorical account of feedback accords for instance with (Katis et al., 1997; Abramsky, 1996). But in presenting a semantics of dataflow as profunctors we obtain the benefits to be had from placing nondeterministic dataflow centrally within categories of models for concurrency, and in particular within presheaf models.

Structure of the paper: In Sec. 1 and Sec. 2 we review the well known causal anomalies associated with relational semantics of indeterminate dataflow and recall the notion of traced monoidal categories. In Sec. 3 we present the classical model of non-deterministic dataflow based on sets of execution traces as an instance of a traced monoidal category, and describe the relationship to the so-called *history-model* of input-output relations categorically. We then proceed in Sec. 4 to present the bicategory of *profunctors* as a categorical generalisation of input-output relations and provide a concrete, operational reading of profunctors as (monotone) port automata. We end the section by identifying the subcategory of *stable* port profunctors and provide a characterisation of such profunctors as stable monotone port automata. In Sec. 5 we define a trace for the category of stable port profunctors, and give two characterisations of the trace in terms of port automata and an abstract characterisation of the trace as a colimit. The details of the proof that the trace satisfies the axioms of a traced monoidal category (up to isomorphism) can be found in App. A. Finally, Sec. 6 briefly goes through some of the consequences of the categorical semantics of dataflow; the relationship to *stable domain theory*, a *bisimulation congruence* obtained from the theory of *open maps*, and finally a model of *(linear) higher-order dataflow* by the *geometry of interaction construction*. The paper is a revised version of ch. 8 in (Hildebrandt, 1999a), a short version appears in the proceedings of Concur '98.

1. Models for Indeterminate Dataflow

The Dataflow paradigm for parallel computation, originated in work of Jack Dennis and others in the mid-sixties (Kahn, 1974; Dennis, 1974). The essential idea is that data flows between asynchronous computing agents, that are interconnected by communication channels acting as unbounded buffers. Traditionally, the *observable behaviour* is taken to be the *input-output* relation between sequences of values on respectively input and output channels, sometimes referred to as the *history model* (Jonsson, 1989). If $\mathcal{V}$ is the set of values, the set of *histories on ports* A is defined to be the set of functions $(\mathcal{V}^*)^A$ from the set of port names A to the set of sequences of values $\mathcal{V}^*$. An *IO-relation* for a dataflow with input ports A and output ports B is then a relation $R \subseteq (\mathcal{V}^*)^A \times (\mathcal{V}^*)^B$.

For dataflow networks built from only *deterministic* nodes, Kahn (Kahn, 1974) has observed that their behaviour can be captured *denotationally* in the history model, defining network composition by the least fixed point of a set of equations describing the components, which was later shown formally by several authors, e.g. Faustini (Faustini, 1982), Lynch and Stark (Lynch and Stark, 1989). Subsequently, different semantics have been described as satisfying *Kahn's principle* when they are built up compositionally along similar lines (Abramsky, 1990).

For *indeterminate* networks, the situation is not so simple. Brock and Ackerman (Brock and Ackerman, 1981) showed the fact, referred to as the "Brock-Ackerman anomaly", that for networks containing the nondeterministic primitive *fair merge*, the history model preserves too little information about the structure of the networks to support a compositional semantics. Later, Trakhtenbrot and Rabinovich, and independently, Russell gave examples of anomalies showing that this is true even for the simplest nondeterministic primitive[†] the ordinary *bounded choice*. We present a similar example to illustrate what additional information is needed. It works by giving two simple examples of automata $\mathcal{A}_1$ and $\mathcal{A}_2$, which have the same input-output relation,

[†] See (Panangaden and Stark, 1988; Panangaden, 1995) for a detailed study of indeterminate dataflow primitives.

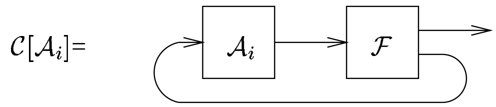


Fig. 1. The automata $\mathcal{A}_i$ inserted in context $\mathcal{C}[-]$ consisting of a fork process $\mathcal{F}$ and a feedback loop

and a context $\mathcal{C}[-]$ in which they behave differently. The context consists of a fork process $\mathcal{F}$ (a process that copies every input to two outputs), through which the output of the automata $\mathcal{A}_i$ is fed back to the input channel, as shown in Fig. 1. Automaton $\mathcal{A}_1$ has the following (deterministic) behaviour: It outputs a token; waits for a token on input and then outputs another token. Automaton $\mathcal{A}_2$ has the choice between two behaviours: Either it outputs a token and stops, *or* it waits for an input token, then outputs two tokens. For both automata, the IO-relation relates empty input to zero or one output token, and non-empty input to zero, one or two output tokens, but $\mathcal{C}[\mathcal{A}_1]$ can output two tokens, whereas $\mathcal{C}[\mathcal{A}_2]$ can only output a single token, choosing the first behaviour of $\mathcal{A}_2$. This example shows the need for a model that records a more detailed causality relation between individual data tokens than the history model.

Jonsson (Jonsson, 1989) and Kok (Kok, 1987) have independently given fully abstract models for indeterminate dataflow. Jonsson's model is based on trace‡ sets, which are sets of possible interaction sequences, finite or infinite, between a process and its environment. Kok's model turned out to be equivalent. Rabinovich and Traktenbrot analyzed the same issues from the point of view of finite observations and came up with general conditions under which a Kahn-like principle would hold (Rabinovich and Trakhtenbrot, 1988; Rabinovich and Trakhtenbrot, 1989; Rabinovich and Trakhtenbrot, 1990). Abramsky has generalised Kahn's principle to a family of generalised trace set models for indeterminate networks, giving conditions for when a trace set computes a set of functions. The trace set for a composite network is shown to compute a set of least fixed points of functions computed by the constituent networks (Abramsky, 1990).

2. Traced Monoidal Categories

The notion of traced monoidal category abstracts the notion of trace of a matrix from multilinear algebra. However it has emerged in a variety of new contexts including the study of feedback systems (Bainbridge, 1976), knot theory (Jones, 1985) and recursion (Hasegawa, 1997b). The axiomatization presented below is the definition of Joyal, Street and Verity (Joyal, Street and Verity, 1996b), slightly simplified and specialized as in (Hasegawa, 1997b) to the context of (strict) symmetric monoidal categories so that the axioms appear simpler; in particular we do not consider braiding or twists. In the Joyal, Street and Verity paper the fact that trace models feedback (or iteration) is attributed to Bloom, but as far back as 25 years ago Bainbridge had been studying trace in the context of feedback in systems and control theory. Indeed Bainbridge had noticed that there were two kinds of trace (associated with two different monoidal structures)

‡ This word commonly used in the literature unfortunately clashes with "trace" in linear algebra. Normally this is not a problem but the present paper uses this word in both senses, we hope the reader will be able to disambiguate from the context.

in Rel, the category of sets and binary relations. Furthermore he noted that one of the traces corresponds to feedback in what are essentially memoryless Kahn networks. [§]

Below we recall the axioms for a strict symmetric monoidal category equipped with a trace. We assume that the reader is familiar with the notion of a (strict) symmetric tensor product. We write $\otimes$ for the tensor product and $\sigma_{XY} : X \otimes Y \to Y \otimes X$ for the natural isomorphism (the symmetry) in this case. Throughout the paper we will use the notation $f; g \colon X \to Z$ for the composition of arrows $f \colon X \to Y$ and $g \colon Y \to Z$.

Definition 2.1. A **trace** for a symmetric monoidal category $\mathcal{C}$ is a family of functions

$$\mathsf{Tr}^U_{X,Y} : \mathcal{C}(X \otimes U, Y \otimes U) \to \mathcal{C}(X, Y)$$

satisfying the following conditions

1 *Bekic*: $f \colon X \otimes U \otimes V \to Y \otimes U \otimes V$ and $g \colon X \to Y$

$$\mathsf{Tr}^{U \otimes V}_{X,Y}(f) = \mathsf{Tr}^U_{X,Y}(\mathsf{Tr}^V_{X \otimes U, Y \otimes U}(f)) \quad \text{and } \mathsf{Tr}^I_{X,Y}(g) = g \ .$$

2 *Yanking*: $\mathsf{Tr}^U_{U,U}(\sigma_{UU}) = I_U$.

3 *Superposing*: Given $f \colon X \otimes U \to Y \otimes U$

$$\mathsf{Tr}^U_{Z \otimes X, Z \otimes Y}(I_Z \otimes f) = I_Z \otimes \mathsf{Tr}^U_{X,Y}(f) \ .$$

4 *Naturality*: Given $g \colon Z \otimes U \to Y \otimes U$, $f \colon X \to Z$ and $h \colon Y \to W$

$$\mathsf{Tr}^U_{X,W}((f \otimes I_U); g; (h \otimes I_U)) = f; \mathsf{Tr}^U_{Z,Y}(g); h \ .$$

5 *Dinaturality*: Given $f \colon X \otimes U \to Y \otimes V$ and $g \colon V \to U$

$$\mathsf{Tr}^U_{X,Y}(f; (I_Y \otimes g)) = \mathsf{Tr}^V_{X,Y}((I_X \otimes g); f) \ .$$

The intuition of a traced symmetric monoidal category as a category with *feedback* becomes clear when the axioms are presented graphically as in Fig. 2. The symmetry is indicated by a cross inside the box (swapping the wires) and identities just as straight arrows.

The following proposition is an easy consequence of the yanking and naturality conditions, keeping in mind functoriality of $\otimes$ and naturality of symmetries. It shows how composition can be defined from trace and tensor as illustrated by Fig. 3.

Proposition 2.1. Given $g \colon U \to Y$ and $f \colon X \to U$ we have

$$\mathsf{Tr}^U_{X,Y}((f \otimes g); \sigma_{UY}) = f; g \ .$$

This can be viewed as a generalisation of the yanking condition.

It is instructive to consider the two well-known examples of trace in the category of sets and binary relations. In the first case one takes the tensor product to be the cartesian product of the underlying sets and in the second case one takes the tensor product to be disjoint union of sets (with the evident action on relations); we call these structures $(\mathsf{Rel}, \times)$ and $(\mathsf{Rel}, +)$ respectively. The trace in $(\mathsf{Rel}, \times)$ is given by

$$\mathsf{Tr}^U_{X,Y}(R)(x, y) = \exists u \in U. R(x, u, y, u) \ ,$$

[§] We are indebted to Samson Abramsky for pointing this reference out to us.

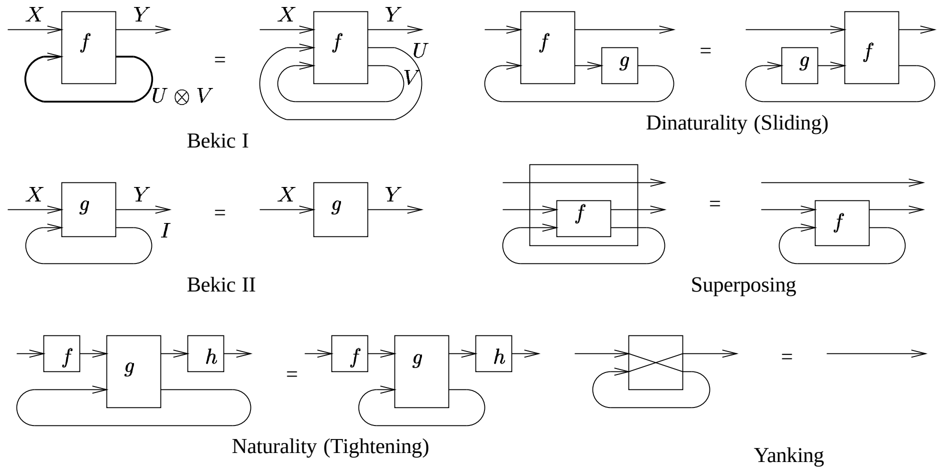


Fig. 2. Graphical presentation of the axioms for a traced strict symmetric monoidal category

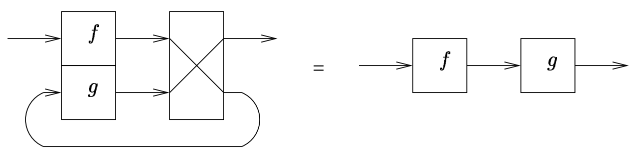


Fig. 3. The generalised Yanking property

where R is a binary relation from $X \times U$ to $Y \times U$. Now for the other structure one proceeds as follows. Let R be a binary relation from $X \uplus U$ to $Y \uplus U$. This can be seen as consisting of 4 pieces, namely the relations R_{XY}, R_{XU}, R_{UY} and R_{UU}. For example we say that $R_{XY}(x, y)$ holds for $x \in X, y \in Y$ iff $R(x, y)$ holds. Now the trace is given by

$$\mathsf{Tr}^{U}_{X,Y}(R) = R_{XY} \cup R_{XU}; R^{*}_{UU}; R_{UY} \quad ,$$

where we are using the standard relational algebra concepts; $*$ for reflexive, transitive closure, ; for relational composition and $\cup$ for union of the sets of pairs in the relation. Intuitively this is the formula expressing feedback: either x and y are directly related or x is related to some u and that u is related to y (once around the feedback loop) or, more generally, we can go around the "feedback loop" an indefinite number of times.

3. A Traced Monoidal Category of Kahn Process

The basic intuitions behind Kahn networks are, of course, due to Kahn (Kahn, 1974) and a formal operational semantics in terms of coroutines is due to Kahn and MacQueen (Kahn and MacQueen, 1977). The particular axiomatisation presented here builds on the ideas of Stark (Stark, 1989a) but using the formalism of traces presented in (Panangaden and Shanbhogue, 1992). No originality is claimed for the trace model; it was Bengt Jonsson (Jonsson, 1989) who showed that traces form a fully abstract model of dataflow networks and there were several others with similar ideas at the time.

We assume a fixed set $\mathcal{V}$ of *values*. A dataflow network processes values on a finite set of input ports producing values on a finite set of output ports. Following (Stark, 1992) we will assume that ports are indexed by natural numbers. By doing this we can smoothly avoid problems with name clashes and in addition work with the simpler categorical structures of *strict* monoidal categories. For $m \leq n \in \omega$ let $[m \dots n] = \{k \mid m \leq k \wedge k < n\}$ and $[n] = [0 \dots n]$, i.e. the set of indexes below n.

An *event* is a triple $\langle s, a, v \rangle$ where $s \in \{\mathsf{i}, \mathsf{o}\}$ is the polarity, $a \in \omega$ is the index of the port and $v \in \mathcal{V}$ the value. We say that $\langle a, v \rangle$ is the *label* of the event $\langle s, a, v \rangle$. An event of the form $\langle \mathsf{o}, a, v \rangle$ is called an *output* event and one of the form $\langle \mathsf{i}, a, v \rangle$ is called an *input* event. We consider sequences of these events, writing $\leq$ for the prefix order on sequences and ϵ for the empty sequence. We write $\mathcal{I}_n$ for the set $\{\mathsf{i}\} \times [n] \ \times \mathcal{V}$ of all input events on ports below n and similarly $\mathcal{O}_n$ for $\{\mathsf{o}\} \times [n] \ \times \mathcal{V}$. We write $\mathcal{L}_n$ for the set $[n] \ \times \mathcal{V}$ of labels on ports below n. If α is a sequence of events we write $l(\alpha)$ for the corresponding sequence of labels. We write $\alpha|_o$ (or $\alpha|_i$) for the sequence of output (or input) events obtained from α by discarding the input (or output) events.

Definition 3.1. A **process** of **sort** (n, m), is a non-empty prefix closed set S of finite sequences over the alphabet $\mathcal{I}_n \cup \mathcal{O}_m$ satisfying the following closure properties:

K1. If $\alpha\langle \mathsf{o}, b, v \rangle\langle \mathsf{i}, a, u \rangle\beta \in S$ then $\alpha\langle \mathsf{i}, a, u \rangle\langle \mathsf{o}, b, v \rangle\beta \in S$ (Output/Input Independence).
K2. If $\alpha\langle \mathsf{o}, b, v \rangle\langle \mathsf{o}, b', u \rangle\beta \in S$ and $b \neq b'$ then $\alpha\langle \mathsf{o}, b', u \rangle\langle \mathsf{o}, b, v \rangle\beta \in S$ (Output Independence).
K3. If $\alpha\langle \mathsf{i}, a, u \rangle\langle \mathsf{i}, a', v \rangle\beta \in S$ and $a \neq a'$ then $\alpha\langle \mathsf{i}, a', v \rangle\langle \mathsf{i}, a, u \rangle\beta \in S$ (Input Independence).
K4. If $\alpha \in S$ then $\alpha\langle \mathsf{i}, a, v \rangle \in S$ for all $\langle a, v \rangle \in \mathcal{L}_n$ (Receptivity),

where α and β are sequences of events. We say that n is the *input arity* of S and m is the *output arity* of S, and let $S \colon n \to m$ denote that S is a process of sort (n, m).

The first three conditions express *independence* between events occurring at different ports. Note the asymmetry in the first condition: Output events are independent of later input events, but not (necessarily) of earlier input events. If an output occurs after an input then it may be in response to the input. The *receptivity* condition expresses that a process could receive any data on its input ports - unlike with synchronous processes. The first independence condition and the receptivity condition can be seen as a monotonicity condition.

As we will see in the end of this section, monotonicity implies that the IO-relations of Kahn processes are *buffered* in a formal sense. This makes them reasonable assumptions for the type of networks we consider. The restriction to *prefix closed* sets of *finite* sequences is a simplification that ensures consistency with the model presented in the following section, but it is not necessary for the results in this section. It is worth stressing, that it rules out the possiblity of expressing (non-continuous) processes such as fair merge and poll, but not non-deterministic processes built from simpler primitives as e.g. bounded choice.

Given processes as sets of sequences we define a strict symmetric monoidal category of Kahn processes. First we need some notations for restricting sequences of events. For $n \in \omega$ we write $\alpha|^{<n}$ (respectively $\alpha|^{\geq n}$) for the sequence obtained from α by keeping only the input events on the ports below n (respectively higher or equal n, subsequently re-indexed by subtracting n from the port index). We can then define $\alpha|^n = (\alpha|^{\geq n})|^{<1}$, i.e. the re-indexed sequence of input

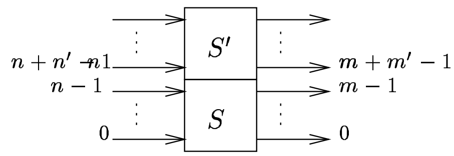


Fig. 4. The shuffle $S' \otimes S$ (parallel composition) of processes $S\colon n \to m$ and $S'\colon n' \to m'$.

events on port n. Formally, define $\epsilon|^{<n} = \epsilon|^{\geq n} = \epsilon$ and for $\alpha = \langle s, a, v\rangle\alpha'$

$$\alpha|^{<n} = \begin{cases} \langle s, a, v\rangle\alpha'|^{<n} & \text{if } s = i \wedge a < n, \\ \alpha'|^{<n} & \text{otherwise,} \end{cases} \tag{1}$$

and

$$\alpha|^{\geq n} = \begin{cases} \langle s, a-n, v\rangle\alpha'|^{\geq n} & \text{if } s = i \wedge a \geq n, \\ \alpha'|^{\geq n} & \text{otherwise.} \end{cases} \tag{2}$$

We define $\alpha|_{<n}$, $\alpha|_{\geq n}$ and $\alpha|_n$ similarly just for output events. Finally, we allow all combinations of these restrictions, e.g. $\alpha|^{\leq n}_{\geq m}$ is the sequence obtained by keeping the input events on ports below n and output events on ports above or equal m, and subsequently re-indexing the ports of the output events by subtracting m from the port number. We extend these notations to sets of sequences.

We can now define for any $n \in \omega$ the *identity process* 1_n of sort (n, n) by

$$1_n = \{\alpha \in (\mathcal{I}_n \cup \mathcal{O}_n)^* \mid \forall \alpha' \leq \alpha \forall k \in [n]. l(\alpha'|_k) \leq l(\alpha'|^k)\} \ .$$

We then define the *shuffle* of two processes, which corresponds to a parallel composition of two processes as illustrated in Fig. 4.

Definition 3.2. For two processes $S\colon n \to m$ and $S'\colon n' \to m'$, define the process

$$S' \otimes S\colon (n + n') \to (m + m')$$

(read, S' **shuffle** S) by

$$S' \otimes S = \{\gamma \in (\mathcal{I}_{n+n'} \cup \mathcal{O}_{m+m'})^* \mid \gamma|^{<n}_{<m} \in S \wedge \gamma|^{\geq n}_{\geq m} \in S'\} \ .$$

For $n, m \in \omega$, the *symmetry process* $\sigma_{n,m}$ of sort $(n + m, m + n)$ is then given by

$$\sigma_{n,m} = \{\alpha \in (\mathcal{I}_{n+m} \cup \mathcal{O}_{m+n})^* \mid l(\alpha|^{\leq n}_{\geq m}) \in 1_n \wedge l(\alpha|^{\geq n}_{<m}) \in 1_m\} \ .$$

We then define sequential composition of two processes to be the shuffle of the two processes, from which we have picked the sequences with the right causal precedence of events on the "internal" connected ports and then discarded these events.

Definition 3.3. For two processes $S\colon n \to m$ and $S'\colon m \to p$, define the composite process

$$S; S'\colon n \to p$$

by

$$S; S' = \{\delta \in S \otimes S' \mid l(\delta|^{<m}) = l(\delta|_{\geq p}) \wedge \forall \delta' \leq \delta.\, l(\delta'|^{<m}) \leq l(\delta'|_{\geq p})\}|^{\geq m}_{<p} \ .$$

It is not difficult to verify that the above definitions allow us to define a strict symmetric monoidal category of Kahn processes as follows.

Definition 3.4. Let Kahn be the strict symmetric monoidal category of Kahn processes, with the natural numbers as objects, processes of sort (n, m) as morphisms from n to m and the identities defined above. The tensor product of the strict monoidal structure is given by sum on objects (with unit 0) and shuffle on arrows and symmetries as defined in Def. 3.2. Composition of morphisms are defined as in Def. 3.3.

The trace construction is as follows.

Definition 3.5. Given $S\colon n + p \to m + p$ we define $Tr^p_{n,m}(S)\colon n \to m$ by

$$\begin{aligned} Tr^p_{n,m}(S) \quad = \{\gamma \in (\mathcal{I}_n \cup \mathcal{O}_m)^* \mid \exists \delta \in S. \quad & \delta|^{\geq p}_{\geq p} = \gamma \\ & l(\delta|^{<p}) = l(\delta|_{<p}) \wedge \\ & \forall \delta' \leq \delta.\ l(\delta'|^{<p}) \leq l(\delta'|_{<p})\} \end{aligned}$$

The condition $\delta|^{\geq p}_{\geq p} = \gamma$ simply says that the projection to input and output events on ports above p gives the sequence γ. The other two conditions express that each of the sequences of values on output ports below p match the sequence of values on the corresponding input ports below p, and that any such output appears before its corresponding input.

The following lemma intuitively expresses that we may allow the feedback ports to be buffered.

Lemma 3.1. Given $S\colon n + p \to m + p$, then

$$\begin{aligned} Tr^p_{n,m}(S) \quad = \{\gamma \in (\mathcal{I}_n \cup \mathcal{O}_m)^* \mid \exists \delta \in S. \quad & \delta|^{\geq p}_{\geq p} = \gamma \wedge \forall k \in [p]. \\ & \big(l(\delta|^k) = l(\delta|_k) \wedge \\ & \forall \delta' \leq \delta.\ l(\delta'|^k) \leq l(\delta'|_k)\big)\} \end{aligned}$$

Proof. (Sketch) We must show that for all processes $S\colon n + p \to m + p$ and $\gamma \in (\mathcal{I}_n \cup \mathcal{O}_m)^*$

$$\exists \delta \in S.\ \delta|^{\geq p}_{\geq p} = \gamma \wedge l(\delta|^{<p}) = l(\delta|_{<p}) \wedge \forall \delta' \leq \delta.\ l(\delta'|^{<p}) \leq l(\delta'|_{<p}) \tag{3}$$

if and only if

$$\exists \delta \in S.\ \delta|^{\geq p}_{\geq p} = \gamma \wedge \forall k \in [p].\big(l(\delta|^k) = l(\delta|_k) \wedge \forall \delta' \leq \delta.\ l(\delta'|^k) \leq l(\delta'|_k)\big) \ . \tag{4}$$

That (3) implies (4) follows directly from the definition of $\delta|^{<p}$ and $\delta|^k$. To show that (4) implies (3) we use the axioms $\mathsf{K1}$ and $\mathsf{K3}$. Assume $\delta \in S$ and $\delta|^{\geq p}_{\geq p} = \gamma \wedge \forall k \in [p].\big(l(\delta|^k) = l(\delta|_k) \wedge \forall \delta' \leq \delta.\ l(\delta'|^k) \leq l(\delta'|_k)\big)$. Let $\delta = \delta'\delta''$ such that δ' is the longest prefix of δ satisfying $l(\delta'|^{<p}) \leq l(\delta'|_{<p})$. We proceed showing (3) holds by induction in the length $|\delta''|$ of δ''. If $|\delta''| = 0$, i.e. $\delta = \delta'$, then $\forall \delta' \leq \delta.\ l(\delta'|^{<p}) \leq l(\delta'|_{<p})$, and then δ also satisfies the condition $l(\delta|^{<p}) = l(\delta|_{<p})$ in (3) and we are done. Now assume that $|\delta''| = n + 1$. Then it follows, that $\delta = \delta'\langle \mathsf{i}, k', v\rangle \delta_1 \langle \mathsf{i}, k, v\rangle \delta_2$ for some $k, k' \in [p]$ and sequence $\langle \mathsf{i}, k', v\rangle \delta_1$ where $\langle \mathsf{i}, k', v\rangle \delta_1|^k = \epsilon$ and $l(\delta'\langle \mathsf{i}, k, v\rangle|^{<p}) \leq l(\delta'\langle \mathsf{i}, k, v\rangle|_{<p})$. Using the axioms $\mathsf{K1}$ and $\mathsf{K3}$ we infer that $\delta'\langle \mathsf{i}, k, v\rangle\langle \mathsf{i}, k', v\rangle \delta_1\delta_2 \in S$. It is easy to see that $\delta'\langle \mathsf{i}, k, v\rangle\langle \mathsf{i}, k', v\rangle \delta_1\delta_2$ satisfies (4) and since $|\langle \mathsf{i}, k', v\rangle \delta_1\delta_2| = n$ we conclude by the induction hypothesis that (3) holds. □

From the alternative definition of the trace given in Lem. 3.1 above it follows quite easily that the trace makes Kahn a strict symmetric monoidal category.

Theorem 3.2. Kahn is a traced strict symmetric monoidal category.

Proof. (Sketch) Follows from a tedious but straightforward verification of the axioms in Fig. 2 using Lem. 3.1. For the Bekic I axiom, we use the equalities $(\delta|^{\geq q}_{\geq q})|^{\geq p}_{\geq p} = \delta|^{\geq p+q}_{\geq p+q}$, $(\delta|^{\geq q}_{\geq q})|^{k} = \delta|^{k+q}$ and $(\delta|^{\geq q}_{\geq q})|_{k} = \delta|_{k+q}$. □

The generalised yanking property can be interpreted in this category as saying that composition can be obtained as a combination of parallel composition (that is, shuffling) and feedback. This is a well-known fact in dataflow folklore.

3.1. *From Kahn Processes to Input-output Relations*

The category of Kahn processes can be related to the history model by a functor from Kahn to a category of buffered relations between histories. As mentioned previously, the set $\mathsf{H_n}$ of *histories on ports* $[n]$, is traditionally defined to be the set $(\mathcal{V}^*)^{[n]}$ of functions from the set of port names $[n]$ to the set of sequences of values $\mathcal{V}^*$ and IO-relations are relations of the form $R \subseteq \mathsf{H_n} \times \mathsf{H_m}$ for $n, m \in \omega$. We use an alternative but equivalent definition of histories, inspired by Mazurkiewicz trace languages, which facilitates the map from Kahn processes to histories. For $n \in \omega$, let $\eqsim_n$ denote the smallest equivalence relation on $\mathcal{L}^*_n$, such that $\alpha\langle a', v'\rangle\langle a, v\rangle\beta \eqsim \alpha\langle a, v\rangle\langle a', v'\rangle\beta \in \mathcal{L}^*_n$ if $a \neq a' \in [n]$. For $\alpha \in \mathcal{L}^*_n$, let $\overline{\alpha}$ denote the equivalence class of α. We then define $\mathsf{H_n}$ to be the quotient set $\mathcal{L}^*_n/\eqsim$ ordered by $\overline{\alpha} \sqsubseteq \overline{\beta} \in \mathcal{L}^*_n/\eqsim$ iff $\exists\gamma \in \mathcal{L}^*_n$ such that $\overline{\alpha\gamma} = \overline{\beta}$ (Winskel and Nielsen, 1995). This is also known as the free partially commutative monoid (Diekert and Métivier, 1997) over $(\mathcal{L}_n, \eqsim_n)$. Viewed as functions, the order corresponds to the standard pointwise order of functions induced by the prefix order on sequences.

For any Kahn process $S\colon n \to m$ we trivially get an IO-relation $\mathcal{H}(S) = \{\, (\overline{\gamma}, \overline{\delta}) \in \mathsf{H_n} \times \mathsf{H_m} \mid \exists\alpha \in S.\ \gamma = l(\alpha|_i) \wedge \delta = l(\alpha|_o)\}$. This mapping preserves composition.

Lemma 3.3. Let $S\colon n \to m$ and $S'\colon m \to m'$ be Kahn processes. Then $\mathcal{H}(S;S') = \mathcal{H}(S);\mathcal{H}(S')$.

Proof. We only show that $\mathcal{H}(S;S') \subseteq \mathcal{H}(S);\mathcal{H}(S')$, the other direction follows by reversing all implications. Assume $(\overline{\gamma}, \overline{\delta}) \in \mathcal{H}(S;S')$. Then there exists $\alpha \in S;S'$ such that

$$\gamma = l(\alpha|_i) \wedge \delta = l(\alpha|_o). \tag{5}$$

By Def. 3.3 there exists $\phi \in S \otimes S'$ such that

$$\alpha = \phi|^{\geq m}_{<p}, \tag{6}$$

$$l(\phi|^{<m}) = l(\phi|_{\geq p}) \qquad \text{and} \tag{7}$$

$$\forall\phi' \leq \phi.\ l(\phi'|^{<m}) \leq l(\phi'|_{\geq p}).$$

But this implies by Def. 3.2 that $\phi|^{<m}_{\geq p} \in S'$ and $\phi|^{\geq m}_{\geq p} \in S$. Let $\phi_{S'} = \phi|^{<m}_{<p}$, $\phi_S = \phi|^{\geq m}_{\geq p}$ and $\beta = l(\phi|^{<m})$. Now from (5) and (6) above it follows that $l(\phi_S|_i) = \gamma$ and $l(\phi_{S'}|_o) = \delta$. From (7) it follows that $l(\phi_{S'}|_i) = l(\phi_S|_o) = \beta$. But then $(\overline{\gamma}, \overline{\beta}) \in \mathcal{H}(S)$ and $(\overline{\beta}, \overline{\delta}) \in \mathcal{H}(S')$ so $(\overline{\gamma}, \overline{\delta}) \in \mathcal{H}(S;S')$. □

By considering the receptivity condition in Def. 3.1, it is easy to see that the mapping given above does not preserve identities, and thus fails to be functorial. However, this can be remedied by considering *buffered* IO-relations. For $n \in \omega$ we define the buffer B_n on $\mathsf{H_n}$ by $B_n = \mathcal{H}(1_n) = \{(\overline{\alpha}, \overline{\beta}) \in H_n \times H_n \mid \overline{\beta} \sqsubseteq \overline{\alpha}\}$. Inspired by (Selinger, 1997), we then say that an IO-relation $R \subseteq H_n \times H_m$ is *buffered* if it satisfies that $R = B_n; R$ and $R = R; B_n$, that is, the relation is closed under composition with buffers. The lemma below gives a characterisation of buffered IO-relations.

Lemma 3.4. Let $R \subseteq H_n \times H_m$. Then the relation R is buffered if and only if
— If $(\overline{\alpha}, \overline{\beta}) \in R$ and $\overline{\alpha} \sqsubseteq \overline{\alpha}'$ then $(\overline{\alpha}', \overline{\beta}) \in R$.
— If $(\overline{\alpha}, \overline{\beta}) \in R$ and $\overline{\beta'} \sqsubseteq \overline{\beta}$ then $(\overline{\alpha}, \overline{\beta}') \in R$.

Proof. Straightforward. □

Noting that $B_n; B_n = B_n$ for any $n \in \omega$ it is easy to see that buffered IO-relations form a category with buffers as identities.

Definition 3.6. Let Hist be the category with objects $\mathsf{H_n}$ for $n \in \omega$ and morphisms buffered IO-relations.

A tensor product is defined on objects by $\mathsf{H_m} \otimes \mathsf{H_n} = \mathsf{H_{m+n}}$ and on arrows by "shuffling" of relations. Define $\epsilon_{|<n} = \epsilon_{|\geq n} = \epsilon$ and for $\alpha = \langle a, v\rangle \alpha' \in \mathcal{L}_n^*$ define

$$\alpha_{|<n} = \begin{cases} \langle a, v\rangle \alpha'_{|<n} & \text{if } a < n, \\ \alpha'_{|<n} & \text{otherwise} \end{cases} \tag{8}$$

and

$$\alpha_{|\geq n} = \begin{cases} \langle a-n, v\rangle \alpha'_{|\geq n} & \text{if } a \geq n, \\ \alpha'_{|\geq n} & \text{otherwise.} \end{cases} \tag{9}$$

We can now define a tensor on arrows formally by $R' \otimes R = \{(\overline{\delta}, \overline{\gamma}) \in \mathsf{H_{n'+n}} \times \mathsf{H_{m'+m}} \mid (\overline{\delta}_{|\geq n}, \overline{\gamma}_{|\geq m}) \in R' \wedge (\overline{\delta}_{|<n}, \overline{\gamma}_{|<m}) \in R\}$ for $R\colon \mathsf{H_n} \to \mathsf{H_m}$ and $R'\colon \mathsf{H_{n'}} \to \mathsf{H_{m'}}$.

Lemma 3.5. The category $(\mathsf{Hist}, \otimes)$ is a strict symmetric monoidal category.

From Lem. 3.3 and the definition of buffers it follows easily that $\mathcal{H}$ defines a symmetric monoidal functor from the category of Kahn processes to the category of buffered IO-relations.

Proposition 3.6. There is a symmetric monoidal functor $\mathcal{H}\colon \mathsf{Kahn} \to \mathsf{Hist}$ on objects defined by $\mathcal{H}(n) = H_n$ and on arrows by $\mathcal{H}(S) = \{\, (\overline{\gamma}, \overline{\delta}) \mid \exists \alpha \in S.\ \gamma = l(\alpha|_i) \wedge \delta = l(\alpha|_o)\}$, for $S\colon n \to m$.

Proof. Functoriality follows from the definition of identities in the two categories and Lem. 3.3. It is easy to verify that the symmetric monoidal structure is preserved. □

4. Generalised Relations

Kahn processes are typical of the solutions to the problem of obtaining a compositional semantics for nondeterministic dataflow. A correct compositional semantics is got by representing processes as interaction sequences, keeping track of the causal dependency between events. However, this seems far removed from the relational history model. In this section we will describe another solution that *contains* the Kahn processes, which comes about as a natural (categorical) extension of the history model. Moreover, it gives a *branching* semantics, which opens the way to combining the asynchronous dataflow communication primitives with synchronous communication primitives.

Our first observation is that buffered IO-relations between $\mathsf{H_n}$ and $\mathsf{H_m}$ correspond exactly to functors $\mathsf{H_n} \times {\mathsf{H_m}}^{\mathsf{op}} \to \mathbf{2}$, where $\mathbf{2}$ is the category consisting of two objects 0 and 1 and only one non-identity arrow $0 \to 1$. This is an immediate categorical analogy to characteristic functions $\mathsf{H_n} \times \mathsf{H_m} \to \{0, 1\}$ of relations. Viewing the buffered IO-relations in this way, composition of $R \colon \mathsf{H_n} \times {\mathsf{H_m}}^{\mathsf{op}} \to \mathbf{2}$ and $R' \colon \mathsf{H_m} \times {\mathsf{H_p}}^{\mathsf{op}} \to \mathbf{2}$ can be written as

$$R; R'(\overline{\alpha}, \overline{\gamma}) = \bigvee_{\overline{\beta} \in \mathsf{H_m}} R(\overline{\alpha}, \overline{\beta}) \wedge R'(\overline{\beta}, \overline{\gamma}) \ , \tag{10}$$

where we make use of the obvious join and meet operations on $\mathbf{2}$.

This defines a category BRel of buffered relations, with path categories as objects, arrows being relations and composition as defined above. The category BRel can be equipped with a strict symmetric monoidal structure as in Hist, and as stated below, BRel is just an alternative presentation of the category Hist given in the previous section.

Proposition 4.1. The category Hist is (strict symmetric monoidal) equivalent to the category BRel.

Proof. Follows easily from Lem. 3.4. □

A trace in BRel can be defined as in $(\mathsf{Rel}, \times)$, that is for $R \colon \mathsf{H_{n+p}} \times {\mathsf{H_{m+p}}}^{\mathsf{op}} \to \mathbf{2}$, define for $(\overline{\alpha}, \overline{\beta})$ in $\mathsf{H_n} \times {\mathsf{H_m}}^{\mathsf{op}}$,

$$\mathsf{Tr}^{\mathsf{H_p}}_{\mathsf{H_n},\mathsf{H_m}}(R)(\overline{\alpha}, \overline{\gamma}) = \bigvee_{\overline{\beta} \in \mathsf{H_p}} R((\overline{\alpha}, \overline{\beta}), (\overline{\gamma}, \overline{\beta})) \ , \tag{11}$$

where we have implicitly used the isomorphisms $\mathsf{H_{n+p}} \cong \mathsf{H_n} \times \mathsf{H_p}$ and $\mathsf{H_{m+p}} \cong \mathsf{H_m} \times \mathsf{H_p}$.

However, the anomaly given in Sec. 1 shows that there is no way of defining a trace on BRel such that the functor $\mathcal{H}$ given in the last section preserves the trace of Kahn. It must be possible to represent *different* dependencies between input and output for a particular input-output pair in the relation. This is precisely what moving to the bicategory of *profunctors* does for us.

4.1. *Profunctors*

The bicategory Prof of profunctors, also referred to as bimodules, or distributors (Borceux, 1994), is a categorical generalisation of sets and relations. The objects of Prof are small categories and arrows are profunctors; profunctors are like the buffered relations above but with the category $\mathbf{2}$ replaced by Set.

Definition 4.1. Let P and Q be small categories. A profunctor X from P to Q is a bifunctor $X \colon \mathsf{P} \times \mathsf{Q}^{\mathsf{op}} \to \mathsf{Set}$ (or equivalently, a presheaf in $\widehat{\mathsf{P}^{\mathsf{op}} \times \mathsf{Q}}$), and will be written as $X \colon \mathsf{P} \,\mathrlap{\to}{+}\, \mathsf{Q}$.

Composition of profunctors $X : \mathsf{P} \relbar\!\mapsto \mathsf{U}$ and $Y : \mathsf{U} \relbar\!\mapsto \mathsf{Q}$ is given by the *coend* (Mac Lane, 1971)

$$X;Y(p,q) = \int^{u} X(p,u) \times Y(u,q) \quad , \tag{12}$$

(which is functorial in its parameters). This defines composition only to within isomorphism, explaining why we get a *bi*category. Note how (12) generalises the expression for relational composition given by (10) earlier. Identities $I_P : \mathsf{P} \relbar\!\mapsto \mathsf{P}$ are given by hom-functors as

$$I_P(p,p') = \mathsf{P}[p',p].$$

The tensor product is given by the product of categories on objects and set-theoretic product on arrows. This defines a symmetric monoidal structure (Day and Street, 1997) on Prof.

Definition 4.2. Let P,P' and Q,Q' be small categories and $X : \mathsf{P} \relbar\!\mapsto \mathsf{Q}$, $Y : \mathsf{P}' \relbar\!\mapsto \mathsf{Q}'$ profunctors. Define $\mathsf{P} \otimes \mathsf{P}' = \mathsf{P} \times \mathsf{P}'$ and $X \otimes Y = X \times Y : \mathsf{P} \otimes \mathsf{P}' \relbar\!\mapsto \mathsf{Q} \otimes \mathsf{Q}'$, so $(X \otimes Y)(p,p',q,q') = X(p,q) \times Y(p',q')$. This defines a (pseudo-) functorial tensor product with unit the one object (and one arrow) category $\mathbf{1}$. The symmetry $\sigma_{PP'} : \mathsf{P} \otimes \mathsf{P}' \relbar\!\mapsto \mathsf{P}' \otimes \mathsf{P}$ is defined in the obvious way from hom-functors, so $\sigma_{PP'}((p,p'),(q',q)) = \mathsf{P} \times \mathsf{P}'[(q,q'),(p,p')]$.

The bicategory Prof is in fact compact closed (Street, 2003), and thus have an essentially unique choice of trace $\int_{P,Q}^{U}(X)$ for a profunctor $X : \mathsf{P} \otimes \mathsf{U} \relbar\!\mapsto \mathsf{Q} \otimes \mathsf{U}$ given by the coend

$$\textstyle\int_{P,Q}^{U}(X)(p,q) = \displaystyle\int^{u} X((p,u),(q,u)) \quad , \tag{13}$$

which satisfies the properties of a trace up to isomorphism. In particular, we can prove the generalised yanking property.

Proposition 4.2. Given $X : \mathsf{P} \relbar\!\mapsto \mathsf{U}$ and $Y : \mathsf{U} \relbar\!\mapsto \mathsf{Q}$ we have a (natural) isomorphism

$$\textstyle\int_{P,Q}^{U}((X \otimes Y);\sigma_{UQ}) \cong X;Y \quad .$$

Proof. By unfolding the definitions we get

$$\begin{aligned}
\textstyle\int_{P,Q}^{U}((X \otimes Y);\sigma_{UQ})(p,q) &= \int^{u} (X \otimes Y);\sigma_{UQ}((p,u),(q,u)) \\
&= \int^{u}\int^{(u',q')} X \otimes Y((p,u),(u',q')) \times \sigma_{UQ}((u',q'),(q,u)) \\
&= \int^{u}\int^{(u',q')} X \otimes Y((p,u),(u',q')) \times \mathsf{U} \times \mathsf{Q}[(u,q),(u',q')] \\
&= \int^{u} (X \otimes Y); I_{U \otimes Q}((p,u),(u,q)) \\
&\cong \int^{u} X(p,u) \times Y(u,q) \quad ,
\end{aligned}$$

where the isomorphism comes from the (natural) isomorphism for composition with the identity in Prof. □

Since we are working with functors into Set, the coend in Eq. (13) has an explicit definition. For p and q objects of respectively P and Q, we have

$$\int^{u} X(p,u,q,u) \cong \biguplus_{u \in \mathsf{U}} \{\, x \in X((p,u),(q,u)) \}_{/\sim} \ , \tag{14}$$

where $\sim$ is the symmetric, transitive closure of the relation $\rightsquigarrow$ defined as follows. For $x \in X(p,u,q,u)$ and $x' \in X(p,u',q,u')$, let $x \rightsquigarrow x'$ if

$$\exists m\colon u \to u' \text{ and } y \in X(p,u,q,u') \text{ such that } X(p,u,q,m)y = x \text{ and } X(p,m,q,u')y = x'.$$

For $f\colon p \to p'$ and $g\colon q' \to q$ arrows of respectively P and Q, we have

$$\int^{u} X((f,u),(g,u))[x]_{\sim} = [X((f,1_u),(g,1_u))x]_{\sim} \quad \text{for } x \in X((p,u),(q,u)). \tag{15}$$

For our purpose, we focus on the subbicategory PProf of Prof generalising the buffered IO-relations, with arrows being profunctors $X\colon \mathsf{H_n} \to\!\!\!\!\!+\!\!\!\!\to \mathsf{H_m}$ between path categories. We refer to such profunctors as *port profunctors*.

A tensor product on PProf is given by $\mathsf{H_n} \otimes \mathsf{H_m} = \mathsf{H_{n+m}}$ with unit $I = \mathsf{H_0}$. Via the isomorphism $\mathsf{H_{n+m}} \cong \mathsf{H_n} \times \mathsf{H_m}$ the category PProf inherits the traced symmetric monoidal structure (up to iso) of Prof. We will refer to the symmetries by $\sigma_{n,m}\colon \mathsf{H_n} \otimes \mathsf{H_m} \to \mathsf{H_m} \otimes \mathsf{H_n}$. Below we will see that this category is a promising candidate for a model of non-deterministic dataflow. However, first we will note that the trace as given by the coend fails to satisfy the causal constraints of feedback, that a token must appear as output before it appears as input on a feedback channel, as stated in the third requirement of the trace in Kahn. This is not surprising, bearing in mind the close relationship to the trace in $(\mathsf{Rel}, \times)$.

Consider the fork process $\mathcal{F}\colon \mathsf{H_1} \to\!\!\!\!\!+\!\!\!\!\to \mathsf{H_1} \otimes \mathsf{H_1}$ used in the example of Sect. 1, which is just a buffer copying each input to two output channels. The port profunctor corresponding to $\mathcal{F}$ is constructed from hom-functors, on objects defined by $\mathcal{F}\big(\overline{\alpha}, (\overline{\beta}, \overline{\gamma})\big) = \mathsf{H_1}[\overline{\beta}, \overline{\alpha}] \times \mathsf{H_1}[\overline{\gamma}, \overline{\alpha}]$. Connecting one of the output channels to the input channel should result in a process with no input channels and one output channel, that can output *nothing* but the empty trace. This is indeed the result in Kahn. However, from the explicit definition of the coend given in (14) it is not difficult to compute that

$$\begin{aligned} \textstyle\int^{\mathsf{H_1}}_{\mathsf{H_0},\mathsf{H_1}} (\mathcal{F})(*, \overline{\beta}) &\cong \biguplus_{\overline{\alpha} \in \mathsf{H_1}} \{\, x \in \mathsf{H_1}[\overline{\beta}, \overline{\alpha}] \times \mathsf{H_1}[\overline{\alpha}, \overline{\alpha}] \}_{/\sim} \\ &\cong \{ \overline{\alpha} \mid \overline{\beta} \leq \overline{\alpha} \in \mathsf{H_1} \} \ , \end{aligned} \tag{16}$$

where $*$ denote the unique history in $\mathsf{H_0}$. This means that $\int^{\mathsf{H_1}}_{\mathsf{H_0},\mathsf{H_1}} (\mathcal{F})(*, \overline{\beta})$ is non-empty (in fact infinite) for *any* output history β, in other words, *any* output history is contained in the (generalised) IO-relation.

For reasons that will become clear in the following section, we will restrict attention further to the sub(bi)category $\mathsf{PProf_r}$ of PProf induced by the *rooted* port profunctors $X\colon \mathsf{H_n} \to\!\!\!\!\!+\!\!\!\!\to \mathsf{H_m}$, which are the port profunctors satisfying that $X(\overline{\alpha}, \overline{\epsilon})$ is the singleton set for any $\overline{\alpha}$ in $\mathsf{H_n}$.

4.2. *An Operational Reading*

The bicategory $\mathsf{PProf_r}$ of rooted port profunctors can be given an operational reading via a generalised Grothendieck construction (Street, 1980), thereby representing profunctors as particular *port-automata* (Panangaden and Stark, 1988; Lynch and Stark, 1989). This is a direct generalisation of the representation of rooted presheaves as *synchronisation trees* given in (Joyal, Nielsen and Winskel, 1996a; Winskel and Nielsen, 1996). In particular, the hom-category $\mathsf{PProf_r}[I, \mathsf{H_1}]$ is equivalent to the category of rooted presheaves over $\mathsf{H_1}$, which is equivalent to the category $\mathsf{ST}_{\mathcal{V}}$ of *synchronisation trees* with label set $\mathcal{V}$. In this way, the hom-categories in $\mathsf{PProf_r}$ can be viewed as generalising the presheaf models used in giving semantics to *synchronously* comunicating systems (Cattani and Winskel, 1997), to be able to represent multiple output ports and asynchronous input.

Port automata are particular examples of *I/O automata* (Lynch and Tuttle, 1987). An I/O automaton is a labelled transition system for which the set of actions is partioned in sets of input, output and internal actions. A port automaton is then an I/O automaton for which the sets of input and output actions consist of respectively input events and output events for specified sets of input and output ports. We need only the special case where the set of internal actions is empty and where the port sets are index sets.

Definition 4.3. Let $n, m \in \omega$. An (n, m)-*port automaton* $\mathcal{A}$ is a quintuple $(S, r, \longrightarrow, [n]\ , [m]\ )$, where

— S is a set of *states*,
— $r \in S$ is the *initial state*,
— $[n]$ and $[m]$ are the sets of resp. *input ports* and *output ports*, and
— $\longrightarrow \subseteq S \times Act \times S$, for $Act = \mathcal{I}_n \cup \mathcal{O}_m$, is the *transition relation*.

The transition relation is extended to sequences of actions as follows. We write $s \xrightarrow{\phi} s'$ if $\phi \in Act^k$ for some $k \in \omega$ and $s \xrightarrow{\phi_0} s_1 \xrightarrow{\phi_1} s_2 \ldots \xrightarrow{\phi_{k-1}} s_k = s'$. We write $s \xrightarrow{\mathsf{i}\,\alpha} s'$ if $\alpha \in ([n]\ \times \mathcal{V})^k$ for some $k \in \omega$ and $s \xrightarrow{\mathsf{i}\,\alpha_0} s_1 \xrightarrow{\mathsf{i}\,\alpha_1} s_2 \ldots \xrightarrow{\mathsf{i}\,\alpha_{k-1}} s_k = s'$, and similarly for output transitions. Define $Seq(\mathcal{A}) = \{\phi \in Act^* \mid \exists s \in S.\ r \xrightarrow{\phi} s\}$, i.e. the set of finite sequences of events labelling sequences of transitions of $\mathcal{A}$ beginning at the initial state r.

We define morphisms between (n, m)-port automata as for labelled transition systems (Winskel and Nielsen, 1995).

Definition 4.4. Let $n, m \in \omega$ and let $\mathcal{A}_i = (S_i, r_i, \longrightarrow_i, [n]\ , [m]\ )$ be two (n, m)-port automata. A morphism $f : \mathcal{A}_1 \to \mathcal{A}_2$ consists of a function $f : S_1 \to S_2$ such that

— $f(r_1) = r_2$,
— if $s \xrightarrow{\phi}_1 s'$ then $f(s) \xrightarrow{\phi}_2 f(s')$

It is easy to check that the above definition makes (n, m)-port automata into a category, which we will denote by $\mathsf{PAut}[n, m]$.

Below we define the associated (n, m)-port automaton for a port profunctor. If $\overline{\alpha} \sqsubseteq \overline{\gamma}$ in $\mathsf{H_n}$, we write $[\overline{\alpha}, \overline{\gamma}] : \overline{\alpha} \to \overline{\gamma}$ and $[\overline{\gamma}, \overline{\alpha}] : \overline{\gamma} \to \overline{\alpha}$ for the unique arrows in respectively $\mathsf{H_n}$ and $\mathsf{H_n}^{\mathsf{op}}$.

Definition 4.5. Let $X : \mathsf{H_n} \mathrel{-\!\!\mapsto} \mathsf{H_m}$ be a rooted port profunctor. Define its *associated* (n,m)*-port automaton* by $\mathcal{A}(X) = (S, r, \longrightarrow, [n]\ , [m]\ )$, where

— $S = \{\big((\overline{\alpha},\overline{\beta}),x\big) \mid x \in X(\overline{\alpha},\overline{\beta})\}$,
— $r = \big((\epsilon,\epsilon),x\big)$ for $x \in X(\epsilon,\epsilon)$, and
— $\longrightarrow \subseteq S \times \mathit{Act} \times S$ is given by the rules by

– $\big((\overline{\alpha},\overline{\beta}),x\big) \stackrel{\mathsf{i}\,a,v}{\longrightarrow} \big((\overline{\alpha\langle a,v\rangle},\overline{\beta}),y\big)$, if $X([\overline{\alpha},\overline{\alpha\langle a,v\rangle}],\overline{\beta})x = y$,

– $\big((\overline{\alpha},\overline{\beta}),x\big) \stackrel{\mathsf{o}\,b,v}{\longrightarrow} \big((\overline{\alpha},\overline{\beta\langle b,v\rangle}),y\big)$, if $X(\overline{\alpha},[\overline{\beta\langle b,v\rangle},\overline{\beta}])y = x$.

We will write $\mathit{Seq}(X)$ for $\mathit{Seq}\big(\mathcal{A}(X)\big)$.

The port automaton obtained from a port profunctor defines the action on objects of a functor from the hom-category $\mathsf{PProf_r}[\mathsf{H_n},\mathsf{H_m}]$ of rooted profunctors between $\mathsf{H_n}$ and $\mathsf{H_m}$ to the category $\mathsf{PAut}[n,m]$.

Remarkably, axioms similar to those usually postulated for monotone port automata (Panangaden and Stark, 1988) follow for port automata of profunctors simply by functoriality.

Proposition 4.3. Let $X : \mathsf{H_n} \mathrel{-\!\!\mapsto} \mathsf{H_m}$ be a rooted port profunctor and $\mathcal{A}(X) = (S, r, \longrightarrow, [n]\ , [m]\ )$ its associated port automaton. Then

A1. *For all* $(a,v) \in [n]\ \times \mathcal{V}$*, and* $s \in S$ *there exists a unique* $s' \in S$ *such that* $s \stackrel{\mathsf{i}\,a,v}{\longrightarrow} s'$ *(Receptivity),*

A2. *For all* $s \stackrel{\mathsf{o}\,b,v}{\longrightarrow} t$ *and* $s \stackrel{\mathsf{i}\,a,v'}{\longrightarrow} t'$ *there exists a unique* $u \in S$ *such that* $t \stackrel{\mathsf{i}\,a,v'}{\longrightarrow} u$ *and* $t' \stackrel{\mathsf{o}\,b,v}{\longrightarrow} u$*,*

pictorially $$\begin{array}{ccc} & t' & \\ {\scriptstyle \mathsf{i}\,a,v'}\nearrow & & \searrow{\scriptstyle \mathsf{o}\,b,v} \\ s & \Rightarrow \exists! & u \\ {\scriptstyle \mathsf{o}\,b,v}\searrow & & \nearrow{\scriptstyle \mathsf{i}\,a,v'} \\ & t & \end{array}$$ *(Monotonicity),*

A3. *For all* $a \neq a' \in [n]$ *, if* $s \stackrel{\mathsf{i}\,a,v}{\longrightarrow} t$ *and* $t \stackrel{\mathsf{i}\,a',v'}{\longrightarrow} u$ *then there exists a unique* $t' \in S$ *such that* $s \stackrel{\mathsf{i}\,a',v'}{\longrightarrow} t'$ *and* $t' \stackrel{\mathsf{i}\,a,v}{\longrightarrow} u$*, pictorially* $$\begin{array}{ccc} & u & \\ {\scriptstyle \mathsf{i}\,a',v'}\nearrow & & \nwarrow{\scriptstyle \mathsf{i}\,a,v} \\ t & \Rightarrow \exists! & t' \\ {\scriptstyle \mathsf{i}\,a,v}\nwarrow & & \nearrow{\scriptstyle \mathsf{i}\,a',v'} \\ & s & \end{array}$$ *(Input Commutativity),*

A4. *For all* $b \neq b' \in [m]$ *, if* $s \stackrel{\mathsf{o}\,b,v}{\longrightarrow} t$ *and* $t \stackrel{\mathsf{o}\,b',v'}{\longrightarrow} u$ *then there exists a unique* $t' \in S$ *such that* $s \stackrel{\mathsf{o}\,b',v'}{\longrightarrow} t'$ *and* $t' \stackrel{\mathsf{o}\,b,v}{\longrightarrow} u$*, pictorially* $$\begin{array}{ccc} & u & \\ {\scriptstyle \mathsf{o}\,b',v'}\nearrow & & \nwarrow{\scriptstyle \mathsf{o}\,b,v} \\ t & \Rightarrow \exists! & t' \\ {\scriptstyle \mathsf{o}\,b,v}\nwarrow & & \nearrow{\scriptstyle \mathsf{o}\,b,v'} \\ & s & \end{array}$$ *(Output Commutativity),*

The monotonicity axiom A2 says that an input transition can never disable an output transition. It can equivalently be stated as an asymmetric commutativity axiom between input and output arrows.

A2'. *For all* $s \stackrel{\mathsf{o}\,b,v}{\longrightarrow} t$ *and* $t \stackrel{\mathsf{i}\,a,v'}{\longrightarrow} u$ *there exists a unique* $t' \in S$ *such that* $s \stackrel{\mathsf{i}\,a,v'}{\longrightarrow} t'$ *and* $t' \stackrel{\mathsf{o}\,b,v}{\longrightarrow} u$*,*

pictorially $$\begin{array}{ccc} & u & \\ {\scriptstyle \mathsf{i}\,a,v'}\nearrow & & \nwarrow{\scriptstyle \mathsf{o}\,b,v} \\ t & \Rightarrow \exists! & t' \\ {\scriptstyle \mathsf{o}\,b,v}\nwarrow & & \nearrow{\scriptstyle \mathsf{i}\,a,v'} \\ & s & \end{array}$$ *(Asymmetric I/O Commutativity).*

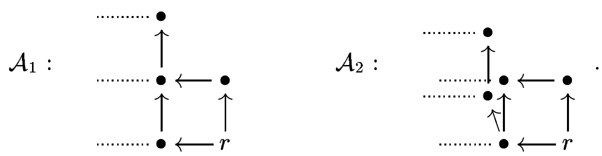


Fig. 5. The initial parts of two port automata representing the networks given in Sec. 1

Proposition 4.4. Axiom A1 together with axiom A2 of proposition 4.3 are equivalent to axiom A1 together with axiom A2'.

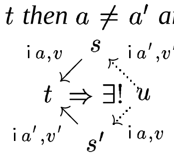
Proof. Assume axiom A1 and A2. If $s \xrightarrow{\mathsf{o}\, b,v} t$ and $t \xrightarrow{\mathsf{i}\, a,v'} u$, then by axiom A1 one gets that $\exists! t' \in S.\ s \xrightarrow{\mathsf{i}\, a,v'} t'$ and from axiom A2 it follows that $\exists u' \in S.\ t' \xrightarrow{\mathsf{o}\, b,v} u'$ and $t \xrightarrow{\mathsf{i}\, a,v'} u'$. Axiom A1 implies that $u = u'$. For the other direction, assume axiom A1 and A2'. If $s \xrightarrow{\mathsf{o}\, b,v} t$ and $s \xrightarrow{\mathsf{i}\, a,v'} t'$, then by axiom A1 one gets $\exists u \in S.\ t \xrightarrow{\mathsf{i}\, a,v'} u$ and from axiom A2' it follows that $\exists! t'' \in S.\ s \xrightarrow{\mathsf{i}\, a,v'} t''$ and $t'' \xrightarrow{\mathsf{o}\, b,v} u$. Axiom A1 gives that $t' = t''$. □

As a corollary, we get a mapping from port profunctors to Kahn processes.

Corollary 4.5. Let $X \colon \mathsf{H_n} \nrightarrow \mathsf{H_m}$ be a rooted port profunctor. Then $Seq(X)$ is a Kahn process of sort (n, m).

Figure 5 shows (the initial parts of) two $(1,1)$-port automata representing the networks described in Sect. 1 (where we assumed that the set of values $\mathcal{V}$ consisted of just a single "token"). All vertical arrows are output transitions (producing a token on the unique output port) and horisontal arrows are input transitions (receiving a token on the unique input port). The dotted transitions indicate that the automata can receive more input tokens. Notice the two runs in automaton $\mathcal{A}_2$, with the same action sequence, but different dependencies.

The operational reading makes rooted port profunctors look promising as a model of non-deterministic dataflow. However, in order to define the trace operation we restrict attention to the class of *stable (rooted) port profunctors*. These are the profunctors for which the associated port automaton satisfies the additional axiom

A5. *For all $s \neq s' \in S$, if $s \xrightarrow{\mathsf{i}\, a,v} t$ and $s' \xrightarrow{\mathsf{i}\, a',v'} t$ then $a \neq a'$ and there exists a unique $u \in S$ such that $u \xrightarrow{\mathsf{i}\, a',v'} s$ and $u \xrightarrow{\mathsf{i}\, a,v} s'$, pictorially*  *(Stability).*

Categorically, axiom A5 is equivalent to requiring that the profunctor (when regarded as a functor $X \colon \mathsf{H_n} \to \widehat{\mathsf{H_m}}$) preserves pullbacks. In our setting, this amounts to preserving meets of bounded (or consistent) pairs. As explained in Sec. 6.1, stable profunctors can be viewed as a direct generalisation of the stable continuous functions of Berry (Berry, 1979; Gunter, 1992). Stable rooted port profunctors define a sub symmetric monoidal bicategory of $\mathsf{PProf_r}$, which we will refer to as $\mathsf{PProf_s}$. We will use the notation $X \colon \mathsf{H_n} \nrightarrow_s \mathsf{H_m}$ to indicate that X is a profunctor in $\mathsf{PProf_s}$.

For any Kahn process S there exists a stable rooted port profunctor $P(S)$ such that $Seq(P(S)) = S$, i.e. the set of sequences for $P(S)$ as defined in Def. 4.5 equals the original process S. The profunctor $P(S)$ can be defined by $P(S)(\overline{\alpha}, \overline{\beta}) = \{\gamma \in S \mid \gamma|_i \in \overline{\alpha} \ \wedge \ \gamma|_o \in \overline{\beta}\}$. This construction gives a functor P from hom-categories $\mathsf{Kahn}[n, m]$ to $\mathsf{PProf_s}[\mathsf{H_n}, \mathsf{H_m}]$ with Seq as left inverse. However, it does not map identities to identities, so it does not extend to a functor between the full categories. It remains an open question if there exists a functor from Kahn to $\mathsf{PProf_s}$ having Seq as left inverse.

We end this section giving a complete characterisation of stable port profunctors in terms of port automata. Port automata associated to port profunctors will always have a reachable and acyclic transition relation with at most one transition between any two states. The possibility of two transitions having the same codomain is restricted by axiom A5 and further by the following two "unfolding" axioms.

U1. *If* $s \neq s' \in S$, $s \xrightarrow{\mathsf{o}\,b,v} t$ *and* $s' \xrightarrow{\mathsf{o}\,b',v'} t$ *then* $b \neq b'$ *and there exists a unique* $u \in S$ *such that* $u \xrightarrow{\mathsf{o}\,b',v'} s$ *and* $u \xrightarrow{\mathsf{o}\,b,v} s'$*, pictorially*

$$\begin{array}{ccc} & s & \\ {}^{\mathsf{o}\,b,v}\swarrow & & \nwarrow^{\mathsf{o}\,b',v'} \\ t & \Rightarrow \exists! & u \\ {}_{\mathsf{o}\,b',v'}\nwarrow & & \swarrow_{\mathsf{o}\,b,v} \\ & s' & \end{array} \quad \textit{(Output Unfold).}$$

U2. *If* $s \neq s' \in S$, $s \xrightarrow{\mathsf{i}\,a,v} u$ *and* $s' \xrightarrow{\mathsf{o}\,b,v'} u$ *then there exists a unique* $u \in S$ *such that* $u \xrightarrow{\mathsf{o}\,b,v'} s$ *and* $u \xrightarrow{\mathsf{i}\,a,v} s'$*, pictorially*

$$\begin{array}{ccc} & s & \\ {}^{\mathsf{i}\,a,v}\swarrow & & \nwarrow^{\mathsf{o}\,b,v'} \\ t & \Rightarrow \exists! & u \\ {}_{\mathsf{o}\,b,v'}\nwarrow & & \swarrow_{\mathsf{i}\,a,v} \\ & s' & \end{array} \quad \textit{(Input/Output Unfold).}$$

Together with the stability axiom, the two unfolding axioms tell that port automata associated to port profunctors are essentially trees, except for the diamonds required by the commutativity axioms A2', A3 and A4.

Theorem 4.6. Let $n, m \in \omega$. Then the hom category $\mathsf{PProf_s}[\mathsf{H_n}, \mathsf{H_m}]$ is equivalent to the full sub category of $\mathsf{PAut}[n, m]$ induced by (n, m)-port automata satisfying axiom A1-A5 and U1-U2, and having a reachable and acyclic transition relation, with at most one transition between any two states.

We can recover the *category of elements* of the presheaf X from its associated port automaton $\mathcal{A}(X)$, which thus determines X up to isomorphism (Mac Lane and Moerdijk, 1992). This allows us to work with the more concrete representation when convenient, and we will freely confuse elements $x \in X(\overline{\alpha}, \overline{\beta})$ with their corresponding states in $\mathcal{A}(X)$.

5. A Trace for Stable Profunctors

The trace as given by the coend in (13) is not well defined in $\mathsf{PProf_s}$, since the coend will not always be a rooted profunctor. This fact is illustrated by the example given in the end of section 4.1, which gives a profunctor with infinitely many roots. Below we will define a trace in $\mathsf{PProf_s}$ which intuitively speaking restricts the coend to *causally secured states*. Observe that the relation $\leadsto$ defined in the explicit definition of the coend given by (14) can be interpreted as a relation between states of the associated port automaton, expressing that two states are connected by a chain of *internal communications*. More precisely, if x and x' are states of a profunctor $X\colon \mathsf{H_n} \otimes \mathsf{H_p} \longrightarrow\!\!\!\!\!+\, \mathsf{H_m} \otimes \mathsf{H_p}$ (or rather, its associated port automaton) and $(c, v) \in [p] \times \mathcal{V}$, we will let $x \overset{c,v}{\Longrightarrow}_p x'$ denote that $x \overset{\mathsf{o}\,c,v}{\longrightarrow}\overset{\mathsf{i}\,c,v}{\longrightarrow} x'$, i.e. x' is reachable from x by two transitions, the first outputs a value on port c and the second inputs the same value on the corresponding input port. We will write $x \Longrightarrow_p x'$ if $x \overset{c,v}{\Longrightarrow}_p x'$ for some $(c, v) \in [p] \times \mathcal{V}$ and let $\Longrightarrow_p^*$ denote the reflexive, symmetric and transitive closure of $\Longrightarrow_p$. If we now take P, Q and U in Eq. (14) to be respectively $\mathsf{H_n}$, $\mathsf{H_m}$ and $\mathsf{H_p}$ we get that $x \rightsquigarrow x'$ if and only if $x \Longrightarrow_p^* x'$. This leads to the following definition.

Definition 5.1. Let $X : \mathsf{H_n} \otimes \mathsf{H_p} \mathrel{-\!\!\mapsto_s} \mathsf{H_m} \otimes \mathsf{H_p}$ and $\mathcal{A}(X) = (S, r, \longrightarrow, [n + p], [m + p])$. We say that $s \in S$ is *p-secured* if

$$r \overset{\mathsf{i}\alpha}{\longrightarrow}\Longrightarrow_p^* \overset{\mathsf{o}\beta}{\longrightarrow} s \ ,$$

for some $\alpha \in ([p \ldots n + p] \times \mathcal{V})^*$ and $\beta \in ([p \ldots m + p] \times \mathcal{V})^*$, i.e. a state is p-secured if it is reachable from the root by a sequence of input transitions on ports not in $[p]$, followed by a sequence of internal communication transitions on ports in $[p]$ and ended by a sequence of output transitions on ports not in $[p]$.

Observation 5.1. Let $X : \mathsf{H_n} \otimes \mathsf{H_p} \mathrel{-\!\!\mapsto_s} \mathsf{H_m} \otimes \mathsf{H_p}$ and $\mathcal{A}(X) = (S, r, \longrightarrow, [n + p], [m + p])$. If $s \Longrightarrow_p^* t$ for $s = \big((\overline{\alpha}, \overline{\gamma}_s, \overline{\beta}, \overline{\gamma}_s), x\big)$ and $t = \big((\overline{\alpha}, \overline{\gamma}_t, \overline{\beta}, \overline{\gamma}_t), x'\big)$ then $\overline{\gamma}_s \sqsubseteq \overline{\gamma}_t$, and if $\overline{\gamma}_s = \overline{\gamma}_t$ then $s = t$.

Lemma 5.2. Let $X : \mathsf{H_n} \otimes \mathsf{H_p} \mathrel{-\!\!\mapsto_s} \mathsf{H_m} \otimes \mathsf{H_p}$ and $\mathcal{A}(X) = (S, r, \longrightarrow, [n + p], [m + p])$. Then $\Longrightarrow_p^*$ is a partial order satisfying the descending chain condition.

Proof. Follows from Obs. 5.1 and the fact that the partial order $\mathsf{H_p}$ satisfy the descending chain condition. □

Lemma 5.3. Let $X : \mathsf{H_n} \otimes \mathsf{H_p} \mathrel{-\!\!\mapsto_s} \mathsf{H_m} \otimes \mathsf{H_p}$ and $\mathcal{A}(X) = (S, r, \longrightarrow, [n + p], [m + p])$. Then if $s \neq s'$ and $s \Longrightarrow_p t$ and $s' \Longrightarrow_p t$, there exists a (unique) state $u \in S$ such that $u \Longrightarrow_p s$ and $u \Longrightarrow_p s'$.

Proof. Follows from Obs. 5.1 and the axioms U1, U2 and A5. □

Let $\sim_p$ be the least equivalence relation including $\Longrightarrow_p$.

Lemma 5.4. Let $X : \mathsf{H_n} \otimes \mathsf{H_p} \mathrel{-\!\!\mapsto_s} \mathsf{H_m} \otimes \mathsf{H_p}$ and $\mathcal{A}(X) = (S, r, \longrightarrow, [n + p], [m + p])$. Then if $t \sim_p t'$, for $t = \big((\overline{\alpha}, \overline{\gamma}_t, \overline{\beta}, \overline{\gamma}_t), x\big) \in S$ and $t' = \big((\overline{\alpha}, \overline{\gamma}_{t'}, \overline{\beta}, \overline{\gamma}_{t'}), x'\big) \in S$, then there exists a state $z = \big((\overline{\alpha}, \overline{\gamma}_t \wedge \overline{\gamma}_{t'}, \overline{\beta}, \overline{\gamma}_t \wedge \overline{\gamma}_{t'}), x''\big) \in S$ such that $z \Longrightarrow_p^* t$ and $z \Longrightarrow_p^* t'$.

Proof. Induction using Lem. 5.3. □

We are now ready to show a crucial property for stable port profunctors exploited in this paper: any $\sim_p$-equivalence class has a least element with respect to the partial order $\Longrightarrow_p^*$.

Proposition 5.5. Let $X : \mathsf{H_n} \otimes \mathsf{H_p} \mathrel{-\!\!\mapsto_s} \mathsf{H_m} \otimes \mathsf{H_p}$ and $\mathcal{A}(X) = (S, r, \longrightarrow, [n+p], [m+p])$. Then for any $s = \big((\overline{\alpha}, \overline{\gamma}, \overline{\beta}, \overline{\gamma}), x\big) \in S$ the equivalence class $[s]_{\sim_p}$ has a least element with respect to the partial order $\Longrightarrow_p^*$, which we denote by $s_{\triangleright p}$.

Proof. By Obs. 5.1 and Lem. 5.4 we get that if $s \sim_p t$ for $s = \big((\overline{\alpha}, \overline{\gamma}, \overline{\beta}, \overline{\gamma}), x\big)$ and $t = \big((\overline{\alpha}, \overline{\gamma}, \overline{\beta}, \overline{\gamma}), x'\big)$ then $s = t$. This implies that the number of elements of any $\sim_p$-equivalence class is bounded by the number of objects in $\mathsf{H_p}$, which is countable. Let $[s]_{\sim p} = \{s_i\}_{i \in \omega}$. By induction in i, we define (using Lem. 5.4) a set $\{t_i\}_{i\in\omega}$ such that $t_i \Longrightarrow_p^* s_j$ and $t_i \Longrightarrow_p^* t_j$ for $j \leq i$. For the basis, let $t_0 = s_0$. For the induction step, assume we have defined $\{t_j\}_{0\leq j\leq i}$. Then let t_{i+1} be the element given by Lem. 5.4 such that $t_{i+1} \Longrightarrow_p^* t_i$ and $t_{i+1} \Longrightarrow_p^* s_{i+1}$. This completes the definition of $\{t_i\}_{i\in\omega}$. Using Lem. 5.2 we conclude from $t_i \Longrightarrow_p^* t_j$ for $j \leq i$ that $\{t_i\}_{i\in\omega}$ has a (unique) least element and from $t_i \Longrightarrow_p^* s_j$ for $j \leq i$ that this element also is the (unique) least element of $[s]_{\sim p}$. □

The securedness condition has some useful equivalent formulations.

Lemma 5.6. Let $X\colon \mathsf{H_n} \otimes \mathsf{H_p} \rightarrow\!\!\!\!+\!\!\!\rightarrow_s \mathsf{H_m} \otimes \mathsf{H_p}$ and $\mathcal{A}(X) = (S, r, \longrightarrow, [n+p], [m+p])$. Let $N = [p \ldots n+p]$ and $N = [p \ldots m+p]$. Then the following statements are equivalent

1) $s \in S$ is *p-secured,*
2) $r \xrightarrow{\mathsf{i}N}{}^* \sim_p \xrightarrow{\mathsf{o}M}{}^* s$,
3) $r(\xrightarrow{\mathsf{i}N} \cup \sim_p \cup \xrightarrow{\mathsf{o}M})^* s$,
4) there exists a sequence of transitions $r \longrightarrow s_0 \longrightarrow s_1 \longrightarrow \ldots \longrightarrow s_n = s$, such that if $s_i = \big((\overline{\alpha}_i, \overline{\gamma}_i, \overline{\beta}_i, \overline{\delta}_i), x_i\big)$ then $\overline{\gamma}_i \sqsubseteq \overline{\delta}_i$ and $\overline{\gamma}_n = \overline{\delta}_n$.
5) $r \xrightarrow{\mathsf{i}N}{}^* s$ or $\exists s'.s'$ is p-secured and $s' \Longrightarrow_p \xrightarrow{\mathsf{o}M}{}^* s$

Proof. It follows directly that 1) implies 2), that 2) implies 3) and that 1) implies 4). That 3) implies 2) follows by repeated use of the axioms A2, A3 and A4. To show that 2) implies 1), assume that $r \xrightarrow{\mathsf{i}N}{}^* t \sim_p t' \xrightarrow{\mathsf{o}M}{}^* s$. Then, by Lem. 5.4 there exists z such that $z \Longrightarrow_p^* t$ and $z \Longrightarrow_p^* t'$. Since $t = \big((\overline{\alpha}, \epsilon, \epsilon, \epsilon), x\big)$ for some $\overline{\alpha}$ and $x \in X(\overline{\alpha}, \epsilon, \epsilon, \epsilon)$, it follows from Obs. 5.1 that $t = z$. That 4) implies 3) can be shown by an induction in the length of $\overline{\gamma}_n$, using axioms A2, A3 and A4. That 5) is equivalent to 1) follows from a simple induction proof. □

We will say that a state s of an automaton $(S, r, \longrightarrow, [n+p], [m+p])$ is *pre* p-secured if there exists a sequence $r \longrightarrow s_0 \longrightarrow s_1 \longrightarrow \ldots \longrightarrow s_n = s$, such that if $s_i = \big((\overline{\alpha}_i, \overline{\gamma}_i, \overline{\beta}_i, \overline{\delta}_i), x_i\big)$ then $\overline{\gamma}_i \sqsubseteq \overline{\delta}_i$. By 4) above, p-securedness clearly implies pre p-securedness, moreover, for any pre p-secured state $s = \big((\overline{\alpha}, \overline{\gamma}, \overline{\beta}, \overline{\delta}), x\big)$ there exists a (unique) secured state $s' = \big((\overline{\alpha}, \overline{\delta}, \overline{\beta}, \overline{\delta}), x'\big)$ such that $s \xrightarrow{\mathsf{i}[p]}{}^* s'$.

The lemma below says that (pre) p-securedness is closed under forward and backward transitions on ports above p.

Lemma 5.7. Let $X\colon \mathsf{H_n} \otimes \mathsf{H_p} \rightarrow\!\!\!\!+\!\!\!\rightarrow_s \mathsf{H_m} \otimes \mathsf{H_p}$ and $\mathcal{A}(X) = (S, r, \longrightarrow, [n+p], [m+p])$. Let $N = [p \ldots n+p]$ and $M = [p \ldots m+p]$. If $s \in S$ is (pre) p-secured and $s \xrightarrow{\mathsf{i}N} s'$ or $s' \xrightarrow{\mathsf{i}N} s$ or $s \xrightarrow{\mathsf{o}M} s'$ or $s' \xrightarrow{\mathsf{o}M} s$ then s' is (pre) p-secured.

Proof. Assume $s \in S$ is (pre) p-secured. If $s \xrightarrow{\mathsf{i}N} s'$ or $s \xrightarrow{\mathsf{o}M} s'$ it follows directly from Lem. 5.6 that s' is (pre) p-secured. If $s' \xrightarrow{\mathsf{i}N} s$ (or $s' \xrightarrow{\mathsf{o}M} s$), then one may show by induction in n using axiom U2 and axiom A5 (or axiom U1) that if there exists a sequence $r \longrightarrow s_0 \longrightarrow s_1 \longrightarrow \ldots \longrightarrow s_n = s$, such that if $s_i = \big((\overline{\alpha}_i, \overline{\gamma}_i, \overline{\beta}_i, \overline{\delta}_i), x_i\big)$ then $\overline{\gamma}_i \sqsubseteq \overline{\delta}_i$ then there exists a sequence $r \longrightarrow s'_0 \longrightarrow s'_1 \longrightarrow \ldots \longrightarrow s'_{n'} = s'$, such that if $s'_i = \big((\overline{\alpha}'_i, \overline{\gamma}'_i, \overline{\beta}'_i, \overline{\delta}'_i), x'_i\big)$ then $\overline{\gamma}'_i \sqsubseteq \overline{\delta}'_i$. It then follows from 4) above that s' is (pre) p-secured. □

We will now finally define a trace on $\mathsf{PProf_s}$ satisfying the causal constraints of feedback. The definition is given as a restriction on the explicit definition of the coend and that it yields a profunctor follows from Lem. 5.7. Rootedness follows easily from Prop. 5.5. The proof of functoriality and that the trace yields a *stable* port profunctor is postponed.

Definition 5.2. Let $X\colon \mathsf{H_n} \otimes \mathsf{H_p} \multimap\!\!\!\rightarrow_s \mathsf{H_m} \otimes \mathsf{H_p}$. Define $\mathsf{Tr}^{\mathsf{H_p}}_{\mathsf{H_n},\mathsf{H_m}}(X)\colon \mathsf{H_n} \multimap\!\!\!\rightarrow \mathsf{H_m}$, the trace of X to be given by

$$\mathsf{Tr}^{\mathsf{H_p}}_{\mathsf{H_n},\mathsf{H_m}}(X)(\overline{\alpha},\overline{\beta}) = \biguplus_{\overline{\gamma}\in\mathsf{H_p}} \{\, x \in X(\overline{\alpha},\overline{\gamma},\overline{\beta},\overline{\gamma}) \mid x \text{ is } p\text{-secured}\}_{/\sim_p}, \tag{17}$$

where the action on arrows is defined as for the coend. We will often abbreviate $\mathsf{Tr}^{\mathsf{H_p}}_{\mathsf{H_n},\mathsf{H_m}}$ to $\mathsf{Tr}^p_{n,m}$ or just Tr^p.

Below we give two characterisations of the trace in terms of port automata.

Lemma 5.8. Let $X\colon \mathsf{H_n} \otimes \mathsf{H_p} \multimap\!\!\!\rightarrow_s \mathsf{H_m} \otimes \mathsf{H_p}$ and $\mathcal{A}(X) = (S, r, \longrightarrow, [n+p], [m+p])$. Then $\mathcal{A}(\mathsf{Tr}^p_{n,m}(X)) \cong (S_{\sim p}, [r]_{\sim p}, \longrightarrow_{\sim p}, [n]\ , [m]\ )$, where $S_{\sim p} = \{s \in S \mid s \text{ is } p\text{-secured}\}_{/\sim p}$ and

$$[s]_{\sim p} \xrightarrow{\mathsf{i}\,a,v}_{\sim p} [s']_{\sim p} \text{ if } s \xrightarrow{\mathsf{i}\,a',v} s', \text{ for } a' = a+p \text{ and } a \in [n] \text{ and}$$

$$[s]_{\sim p} \xrightarrow{\mathsf{o}\,b,v}_{\sim p} [s']_{\sim p} \text{ if } s \xrightarrow{\mathsf{o}\,b',v'} s', \text{ for } b' = b+p\ b \in [m].$$

The operational reading of Lem. 5.8 is not clear. Intuitively, any computation in the traced automaton should correspond to a computation of the original automaton, consisting of internal communication on ports in $[p]$ and output and input transitions on ports not in $[p]$. Using Prop. 5.5 we can give an alternative characterisation by replacing the equivalence classes of states with their minimal representatives, which has the intuitive operational reading.

Lemma 5.9. Let $X\colon \mathsf{H_n} \otimes \mathsf{H_p} \multimap\!\!\!\rightarrow_s \mathsf{H_m} \otimes \mathsf{H_p}$ and $\mathcal{A}(X) = (S, r, \longrightarrow, [n+p], [m+p])$.

$$s{\sim_p} \xrightarrow{\mathsf{i}\,a',v} {\sim_p}s', \text{ for } a \in [n] \text{ if and only if } s_{\triangleright p} \xrightarrow{\mathsf{i}\,a',v} s'_{\triangleright p}, \text{ for } a' = a+p \text{ and } a \in [n] \text{ and} \tag{18}$$

$$s{\sim_p} \xrightarrow{\mathsf{o}\,b',v} {\sim_p}s', \text{ for } b \in [m] \text{ if and only if } s_{\triangleright p} \Longrightarrow^*_p \xrightarrow{\mathsf{o}\,b',v} s'_{\triangleright p}, \text{ for } b' = b+p \text{ and } b \in [m]. \tag{19}$$

Proof. The *if* case of both (18) and (19) is immediate. The *only if* case of (18): From the axioms $\mathsf{A2}$ and $\mathsf{A3}$ we get that $s{\sim_p} \xrightarrow{\mathsf{i}\,a+p,v} {\sim_p}s'$ implies $s_{\triangleright p} \xrightarrow{\mathsf{i}\,a+p,v} {\sim_p}s'$. Now, assume $t \xRightarrow{p}_p s'$. Then by axioms $\mathsf{U1}$, $\mathsf{U2}$ it follows that there exists $s' \xRightarrow{p}_p s_{\triangleright p}$, contradicting the minimality of $s_{\triangleright p}$. So we can conclude that $s' = s'_{\triangleright p}$. The *only if* case of (19) follows in a similar way, using axiom $\mathsf{A4}$. □

Proposition 5.10. Let $X\colon \mathsf{H_n} \otimes \mathsf{H_p} \multimap\!\!\!\rightarrow_s \mathsf{H_m} \otimes \mathsf{H_p}$ and $\mathcal{A}(X) = (S, r, \longrightarrow, [n+p], [m+p])$. Then $\mathcal{A}(\mathsf{Tr}^p_{n,m}(X)) \cong (S_p, r, \longrightarrow_p, [n]\ , [m]\ )$, $S_p = \{s_{\triangleright p} \in S \mid s \text{ is } p\text{-secured}\}$, where $s_{\triangleright p}$ is given as in Prop. 5.5 and

$$s_{\triangleright p} \xrightarrow{\mathsf{i}\,a,v}_p s'_{\triangleright p} \text{ if } s_{\triangleright p} \xrightarrow{\mathsf{i}\,a+p,v} s'_{\triangleright p}, \text{ for } a \in [n] \text{ and}$$

$$s_{\triangleright p} \xrightarrow{\mathsf{o}\,b,v}_p s'_{\triangleright p} \text{ if } s_{\triangleright p} \xRightarrow{p}{}^*_p \xrightarrow{\mathsf{o}\,b+p,v'} s'_{\triangleright p}, \text{ for } b \in [m].$$

This characterisation can be read as a *demand driven* feedback, since internal communication does only occur when needed for producing output.

Using the port automata characterisation of the trace, we now prove that the trace as defined in Def. 5.2 is functorial and yields a *stable* port profunctor.

Lemma 5.11. The trace given in Def. 5.2 above defines the action on objects for a functor $\mathsf{Tr}^{\mathsf{H_p}}_{\mathsf{H_n},\mathsf{H_m}} \colon \mathsf{PProf_s}[\mathsf{H_n} \otimes \mathsf{H_p}, \mathsf{H_m} \otimes \mathsf{H_p}] \to \mathsf{PProf_s}[\mathsf{H_n}, \mathsf{H_m}]$.

Proof. Let $X \colon \mathsf{H_n} \otimes \mathsf{H_p} \mathrel{\mapstochar\!\!\!\longrightarrow}_s \mathsf{H_m} \otimes \mathsf{H_p}$ and $\mathcal{A}(X) \cong (S, r, \longrightarrow, [n+p], [m+p])$. We first show that the automaton $(S_p, r, \longrightarrow_p, [m]\ , [n]\ )$ associated to $\mathsf{Tr}^p_{n,m}(X)$ as given in Prop. 5.10 satisfies axiom $\mathsf{A5}$. Assume $s_{\triangleright p} \neq s'_{\triangleright p}, s_{\triangleright p} \xrightarrow{\mathsf{i}\,a,v}_p t_{\triangleright p}$ and $s'_{\triangleright p} \xrightarrow{\mathsf{i}\,a',v'}_p t_{\triangleright p}$. Then $s_{\triangleright p} \xrightarrow{\mathsf{i}\,a+p,v} t_{\triangleright p}$ and $s'_{\triangleright p} \xrightarrow{\mathsf{i}\,a'+p,v'} t_{\triangleright p}$. By stability of X, $a+p \neq a'+p$ and $\exists ! u.\ u \xrightarrow{\mathsf{i}\,a'+p,v'} s_{\triangleright p}$ and $u \xrightarrow{\mathsf{i}\,a+p,v} s'_{\triangleright p}$. By Lem. 5.7, u is p-secured, and as in the proof of Lem. 5.9 it follows that $u = u_{\triangleright p}$. We conclude that $\exists ! u_{\triangleright p}.\ u_{\triangleright p} \xrightarrow{\mathsf{i}\,a',v'}_p s_{\triangleright p}$ and $u_{\triangleright p} \xrightarrow{\mathsf{i}\,a,v}_p s'_{\triangleright p}$.

Functoriality of the trace is given as for the coend. Assume $\tau \colon X \dot{\to} Y$ is a natural transformation between X and Y in $\mathsf{PProf_s}[\mathsf{H_n} \otimes \mathsf{H_p}, \mathsf{H_m} \otimes \mathsf{H_p}]$. Then $\mathsf{Tr}^p_{n,m}(\tau) \colon \mathsf{Tr}^p_{n,m}(X) \dot{\to} \mathsf{Tr}^p_{n,m}(Y)$ is given by $\mathsf{Tr}^p_{n,m}(\tau)_{\overline{\alpha},\overline{\beta}}[x]_{\sim p} = [\tau_{\overline{\alpha},\overline{\gamma},\overline{\beta},\overline{\gamma}} x]_{\sim p}$, for $\overline{\alpha} \in \mathsf{H_n}$, $\overline{\beta} \in \mathsf{H_m}$ and $\overline{\gamma} \in \mathsf{H_p}$. To check that this is well defined we just need to verify that if x is p-secured then $\tau_{\overline{\alpha},\overline{\gamma},\overline{\beta},\overline{\gamma}} x$ is p-secured as well. This is a simple consequence of the definition. □

As for the trace, we can characterise composition and the tensor product directly in terms of automata.

Lemma 5.12. Let $X \colon \mathsf{H_n} \mathrel{\mapstochar\!\!\!\longrightarrow}_s \mathsf{H_m}$ and $\mathcal{A}(X) = (S, r, \longrightarrow, [n], [m])$. Then for all $s = \big((\overline{\alpha}, \overline{\beta}), x\big)$ in S there exists a unique state $\lfloor s \rfloor$ in $[s]_{\sim_p}$ such that $\lfloor s \rfloor \xrightarrow{\mathsf{i}N}{}^* s$ and for all $s' \in S$. if $s' \xrightarrow{\mathsf{i}N}{}^* s$ then $\lfloor s \rfloor \xrightarrow{\mathsf{i}N}{}^* s'$.

Proof. By induction in $|\overline{\alpha}|$ using axiom $\mathsf{A5}$. □

Lemma 5.13. Let $X_i \colon \mathsf{H_{n_i}} \to \mathsf{H_{m_i}}$ and $\mathcal{A}(X_i) = (S_i, r_i, \longrightarrow_i, [n_i], [m_i])$ for $i \in \{1, 2\}$ such that $m_1 = n_2$. Then $\mathcal{A}(X_1; X_2) \cong (S_1 \times \lfloor S_2 \rfloor, (r_1, r_2), \longrightarrow_{1\times 2}, [n_1], [m_2])$, where $\lfloor S_2 \rfloor = \{(s_1, s_2) \in S_1 \times S_2 \mid s_2 = \lfloor s_2 \rfloor\}$ and the transition relation $\longrightarrow_{1;2}$ is defined by

— $(s_1, s_2) \xrightarrow{\mathsf{i}\,av}_{1;2} (s'_1, s'_2)$ if $s_1 \xrightarrow{\mathsf{i}\,a'v}_1 s'_1$ and $s_2 = s'_2$

— $(s_1, s_2) \xrightarrow{\mathsf{o}\,bv}_{1;2} (s'_1, s'_2)$ if $s_1 \xrightarrow{\mathsf{o}\,c_1 v_1}_1 \xrightarrow{\mathsf{o}\,c_2 v_2}_1 \cdots \xrightarrow{\mathsf{o}\,c_n v_n}_1 s'_1$ and $s_2 \xrightarrow{\mathsf{i}\,c_1 v_1}_2 \xrightarrow{\mathsf{i}\,c_2 v_2}_2 \cdots \xrightarrow{\mathsf{i}\,c_n v_n}_1 s''_2 \xrightarrow{\mathsf{o}\,bv}_1 s'_2$ and $s'_2 = \lfloor s'_2 \rfloor$.

Lemma 5.14. Let $X_i \colon \mathsf{H_{n_i}} \to \mathsf{H_{m_i}}$ and $\mathcal{A}(X_i) = (S_i, r_i, \longrightarrow_i, [n_i], [m_i])$ for $i \in \{1, 2\}$. Then $\mathcal{A}(X_1 \otimes X_2) \cong (S_1 \times S_2, (r_1, r_2), \longrightarrow_{1\times 2}, [n_1 + n_2], [m_1 + m_2])$, where the transition relation $\longrightarrow_{1\times 2}$ is defined by

— $(s_1, s_2) \xrightarrow{\mathsf{i}\,av}_{1\times 2} (s'_1, s'_2)$ if $(s_1 \xrightarrow{\mathsf{i}\,a'v}_1 s'_1$ and $a = a' + n_2)$ or $s_2 \xrightarrow{\mathsf{i}\,av}_2 s'_2$,

— $(s_1, s_2) \xrightarrow{\mathsf{o}\,bv}_{1\times 2} (s'_1, s'_2)$ if $(s_1 \xrightarrow{\mathsf{o}\,b'v}_1 s'_1$ and $b = b' + m_2)$ or $s_2 \xrightarrow{\mathsf{o}\,bv}_2 s'_2$,

We are now ready to state the main theorem, that the trace operator given in Def. 5.2 satisfy the axioms of a traced monoidal category up to isomorphism. The proof can be found in App. A.

Theorem 5.15. With the trace operator given in Def. 5.2, $\mathsf{PProf_s}$ satisfies (up to isomorphism) the axioms of a traced monoidal category.

Proposition 5.16. The map Seq given in Def. 4.5 defines the action on arrows of a pseudofunctor $Seq\colon \mathsf{PProf_s} \to \mathsf{Kahn}$, that preserves the traced monoidal structure, on objects mapping $\mathsf{H_n}$ to n.

Proof. (Sketch) Use Lem. 5.6 formulation 4) of securedness and Prop. 5.10. □

5.1. *The Trace as a Colimit*

The secured trace can be defined as the composition of two functors on hom-categories; first a functor restricting to secured states and then a *colimit*, hiding internal communication. We will benefit from this formulation of the secured trace in proving that bisimulation is a congruence with respect to trace.

We begin with the standard construction of the *subdivision category* (Mac Lane, 1971) for a category $\mathsf{H_p}$. Note the definition here is the dual to that in (Mac Lane, 1971) since we are concerned with coends and not ends. For a category $\mathsf{H_p}$, the subdivision category ${\mathsf{H_p}}^{\S}$ has as objects all arrows $f\colon \overline{\alpha} \to \overline{\beta}$ of $\mathsf{H_p}$ (i.e. $f = [\overline{\alpha}, \overline{\beta}]$.) For each such object f, it has two arrows $f_{\mathsf{o}}\colon f \to 1_{\alpha}$ and $f_{\mathsf{i}}\colon f \to 1_{\beta}$, i.e. $[\overline{\alpha},\overline{\alpha}] \xleftarrow{f_{\mathsf{o}}} [\overline{\alpha},\overline{\beta}] \xrightarrow{f_{\mathsf{i}}} [\overline{\beta},\overline{\beta}]$. These are the only non-identity arrows. Now we define a functor $\mathcal{S}\colon \mathsf{PProf_s}[\mathsf{H_n}\otimes\mathsf{H_p},\mathsf{H_m}\otimes\mathsf{H_p}] \to \mathsf{Prof}[\mathsf{H_n}\times{\mathsf{H_p}}^{\S},\mathsf{H_m}]$ as in the standard construction, except we restrict to secured states. Let X be a profuncor in $\mathsf{PProf_s}[\mathsf{H_n}\otimes\mathsf{H_p},\mathsf{H_m}\otimes\mathsf{H_p}]$. Define a functor $\mathcal{S}(X)\colon \mathsf{H_n}\times{\mathsf{H_p}}^{\S}\times{\mathsf{H_m}}^{\mathsf{op}} \to \mathsf{Set}$ as follows. For $(\overline{\gamma},[\overline{\alpha},\overline{\beta}],\overline{\delta})$ an object in $\mathsf{H_n}\times{\mathsf{H_p}}^{\S}\times{\mathsf{H_m}}^{\mathsf{op}}$ define

$$\mathcal{S}(X)(\overline{\gamma},[\overline{\alpha},\overline{\beta}],\overline{\delta}) = \{x \in X\big((\overline{\alpha},\overline{\gamma}),(\overline{\beta},\overline{\delta})\big) \mid x \text{ is pre } p\text{-secured}\},$$

implicitly using the equivalence between $\mathsf{Prof}[\mathsf{H_{n+p}},\mathsf{H_{m+p}}]$ and $\mathsf{Prof}[\mathsf{H_n}\times\mathsf{H_p},\mathsf{H_m}\times\mathsf{H_p}]$.

For $h\colon \overline{\gamma} \to \overline{\gamma}'$ and $j\colon \overline{\delta}' \to \overline{\delta}$ arrows of respectively $\mathsf{H_n}$ and $\mathsf{H_m}$, and $x \in \mathcal{S}(X)(\overline{\gamma},[\overline{\alpha},\overline{\beta}],\overline{\delta})$ define

$$\mathcal{S}(X)(h,[\overline{\alpha},\overline{\beta}]_{\mathsf{o}},j)x = X\big((h,1_{\overline{\alpha}}),([\overline{\beta},\overline{\alpha}],j)\big)x \quad \text{and}$$

$$\mathcal{S}(X)(h,[\overline{\alpha},\overline{\beta}]_{\mathsf{i}},j)x = X\big((h,[\overline{\alpha},\overline{\beta}]),(1_{\overline{\beta}},j)\big)x.$$

For $\tau\colon X \dashrightarrow Y$ a natural transformation between profunctors X,Y in $\mathsf{PProf_s}[\mathsf{H_n}\otimes\mathsf{H_p},\mathsf{H_m}\otimes\mathsf{H_p}]$ define a natural transformation $\mathcal{S}(\tau)\colon \mathcal{S}(X) \dashrightarrow \mathcal{S}(Y)$ by

$$\mathcal{S}(\tau)_{(\overline{\gamma},[\overline{\alpha},\overline{\beta}],\overline{\delta})}x = \tau_{(\overline{\gamma},\overline{\alpha},\overline{\beta},\overline{\delta})}x,$$

for $(\overline{\gamma},[\overline{\alpha},\overline{\beta}],\overline{\delta})$ an object in $\mathsf{H_n}\times{\mathsf{H_p}}^{\S}\times{\mathsf{H_m}}^{\mathsf{op}}$.

Well definedness follows from Lem. 5.7 and the fact that if $x \in X\big((\overline{\gamma},\overline{\alpha}),(\overline{\beta},\overline{\delta})\big)$ is pre p-secured then $X\big((1_{\overline{\gamma}},[\overline{\alpha},\overline{\beta}]),(1_{\overline{\beta}},1_{\overline{\delta}})\big)x$ is pre p-secured too, (in fact p-secured).

Proposition 5.17. Let $X : \mathsf{H_n}\otimes\mathsf{H_p} \mathrel{\rightarrowtail\!\!\!\!\!\!\!+}_{s} \mathsf{H_m}\otimes\mathsf{H_p}$. Then

$$\mathsf{Tr}^{p}_{n,m}(X) \cong \mathrm{Colim}_{{\mathsf{H_p}}^{\S}}\mathcal{S}(X), \tag{20}$$

where ${\mathsf{H_p}}^{\S}$ is the *subdivision category* of $\mathsf{H_p}$ as defined above.

Proof. As for the standard construction. □

6. Some Consequences

We will briefly go through some of the consequences of the categorical semantics of dataflow given in the two previous sections.

6.1. *Generalised Scott Domains*

We have chosen to introduce the category $\mathsf{PProf_r}$ as a categorical generalisation of relations. Alternatively, $\mathsf{PProf_r}$ could have been introduced as a generalisation of the category of *Scott-continuous functions* between algebraic domains $\bar{D}$, to the category of *connected colimit preserving functors* between presheaf categories $\widehat{D^+}$ (where $\bar{D}$ is the well-known ideal completion of the basis D). In this sense, the *stable* port profunctors is a direct generalisation of the stable continuous functions of Berry (Berry, 1979; Gunter, 1992). As described in (Cattani, 1999; Cattani and Winskel, 1999), the *free connected-colimit completion* of a category P with initial object $\perp$ can be represented by the strict extension of the Yoneda embedding, $\mathcal{Y}^\circ_{\mathsf{P}} \colon \mathsf{P} \hookrightarrow \widehat{\mathsf{P}^+}$, mapping the initial object $\perp$ to the empty presheaf. The Kan extension of a functor $X \colon \mathsf{H_n} \longrightarrow \widehat{\mathsf{H}^+_\mathsf{m}}$ along $\mathcal{Y}^\circ_{\mathsf{H_n}} \colon \mathsf{H_n} \longrightarrow \widehat{\mathsf{H}^+_\mathsf{n}}$ gives a connected colimit preserving functor $X \colon \widehat{\mathsf{H}^+_\mathsf{n}} \longrightarrow \widehat{\mathsf{H}^+_\mathsf{m}}$, which extends to an equivalence between the category $\mathsf{PProf_r}$ and the category of connected-colimit preserving functors between presheaf categories of the form $\widehat{\mathsf{H}^+_\mathsf{n}}$ for $n \in \omega$. This generalises the ideal completion in domain-theory. Connected-colimit preserving functors satisfy a very strong property (Cattani and Winskel, 2000): they preserve the canonical notion of open maps in presheaf categories - and thus the canonical notion of bisimulation equivalence. We will return to this issue below.

6.2. *A Bisimulation Congruence*

The presentation of models for concurrency as categories allows us to apply a general notion of bisimulation from spans of open maps proposed in (Joyal, Nielsen and Winskel, 1996a) and arising from work of Joyal and Moerdijk (Joyal and Moerdijk, 1994). The general idea is to identify a functor $\mathcal{P} \colon \mathsf{P} \to \mathsf{M}$ from a *path category* P (which in (Joyal, Nielsen and Winskel, 1996a) was assumed to be the inclusion of a subcategory) to the model M, with objects identifying *runs* (or histories) and morphisms compatible extensions of these. A morphism is said to be $\mathcal{P}$*-open* if it satisfies the *path-lifting* property, i.e. it *reflects* extensions of histories. For a presheaf model $\hat{\mathsf{P}}$, the *canonical* choice of path category is the category $\mathsf{P}_\perp$, which is obtained from P by adding a new initial object $\perp$ and embeds in $\widehat{\mathsf{P}}$ by the strict extension of the Yoneda embedding $\mathcal{Y}^\circ_{P_\perp} \colon \mathsf{P}_\perp \to \widehat{\mathsf{P}}$, mapping $\perp$ to the empty presheaf. The definition of $\mathcal{Y}^\circ_{P_\perp}$-open maps in presheaf categories can be given conveniently as follows.

Definition 6.1. Let $f \colon X \to Y$ be a morphism in $\hat{\mathsf{P}}$. Then f is $\mathcal{Y}^\circ_{P_\perp}$*-open* if f is surjective and for any arrow $e \colon P \to Q$ of P the square below is a quasi-pullback.

$$\begin{array}{ccc} X(Q) & \xrightarrow{Xe} & X(P) \\ {\scriptstyle f_Q}\downarrow & & \downarrow{\scriptstyle f_P} \\ Y(Q) & \xrightarrow{Ye} & Y(P) \end{array}$$

Two objects are said to be $\mathcal{P}$-*open map bisimilar* if they are connected by a span of $\mathcal{P}$-open maps. We will usually omit the reference to $\mathcal{Y}^{\circ}_{P_\perp}$ when refering to the canonical open maps and open map bisimulation for presheaf categories. Note that the canonical open maps in the present setting are exactly the *surjective* open maps of (Joyal and Moerdijk, 1994).

Now, since a port profunctor $X : \mathsf{H_n} \relbar\!\!\mapsto_s \mathsf{H_m}$ can be viewed as a presheaf in $\widehat{\mathsf{H_n}^{\mathsf{op}} \times \mathsf{H_m}}$, we get a canonical notion of bisimulation from open maps for port profunctors as defined above. As for the presheaves as transition systems in (Winskel and Nielsen, 1996), the open map bisimulation can be characterised as a back-&-forth bisimulation between the associated port automata.

Proposition 6.1. Let $X_i \colon \mathsf{H_n} \relbar\!\!\mapsto_s \mathsf{H_m}$ and $\mathcal{A}(X_i) = (S_i, r_i, \longrightarrow_i, [n]\ , [m]\ )$ for $i \in \{1, 2\}$. X_1 and X_2 are open map bisimilar iff $\mathcal{A}(X_1)$, $\mathcal{A}(X_2)$ are *back-&-forth bisimilar*: There exists a relation $R \subseteq S_1 \times S_2$ such that $(r_1, r_2) \in R$ and

— $(s, s') \in R \wedge t \xrightarrow{\phi}_1 s \Rightarrow \exists t'.\ t' \xrightarrow{\phi}_2 s' \wedge (t, t') \in R$,
— $(s, s') \in R \wedge s \xrightarrow{\phi}_1 t \Rightarrow \exists t'.\ s' \xrightarrow{\phi}_2 t' \wedge (t, t') \in R$,
— $(s, s') \in R \wedge t' \xrightarrow{\phi}_2 s' \Rightarrow \exists t.\ t \xrightarrow{\phi}_1 s \wedge (t, t') \in R$,
— $(s, s') \in R \wedge s' \xrightarrow{\phi}_2 t' \Rightarrow \exists t.\ s \xrightarrow{\phi}_1 t \wedge (t, t') \in R$.

As noted in the previous section, the hom-category $\mathsf{PProf_s}[I, \mathsf{H_1}]$ is equivalent to the category of synchronisation trees with label set $\mathcal{V}$, indeed the associated port automata are trees labelled with $\{\mathsf{o}\} \times \{0\} \times \mathcal{V}$. The bisimulation in this case reduces to the standard strong bisimulation.

It is important to check that our notion of bisimulation on $\mathsf{PProf_s}$ is a congruence with respect to the operations tensor and trace. Here we can exploit some general properties of open maps and so bisimulation on presheaves: the product of (surjective) open maps in a presheaf category is (surjective) open (Joyal and Moerdijk, 1994); any colimit-preserving functor between presheaf categories preserves (surjective) open maps (Cattani and Winskel, 1997). The proof that trace on $\mathsf{PProf_s}$ preserves bisimulation uses the latter property, exploiting the fact that trace can be expressed as a colimit, first showing that $\mathcal{S}$ as a functor between presheaf categories preserves open maps. The proof of the corresponding result for tensor rests on a construction of tensor from more basic functors, which are all colimit-preserving and so preserve (surjective) open maps. Using the simple fact that bisimulation is a congruence with respect to sequential composition with a symmetry together with the generalised yanking property we can conclude that bisimulation is a congruence with respect to all of the operations on networks as stated in the theorem below. The details of the proof can be found in App. 5.1.

Theorem 6.2. Open map bisimulation in $\mathsf{PProf_s}$ is a congruence with respect to sequential composition, tensor and trace.

The congruence property allows us to work with the quotiont category with bisimulation classes of profunctors as arrows. This is exploted in the following section. Also, by placing dataflow within profunctors and the broader class of presheaf models, constructions of dataflow could be mixed with constructions from other paradigms of concurrent computation such as those traditionally from CCS-like process calculi. As an example, a pairwise synchronous join of output ports between two port profunctors can be represented by the product of presheaves. In this richer world of constructions bisimulation would appear to be the more suitable equivalence.

Fig. 6. A process with bi-directional I/O implemented by an uni-directional process. Dotted lines indicate channels that play the opposite role in the higher-order model

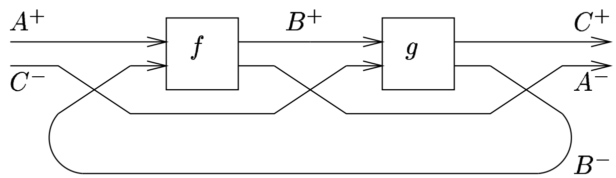


Fig. 7. Implementation of composition in the higher-order model using symmetries and trace

6.3. *Linear Higher-order Dataflow via Geometry of Interaction*

The Geometry of Interaction program was invented by Girard in his analysis of the fine structure of cut elimination (Girard, 1989a; Girard, 1989b). His basic insight was that higher order structure could be understood in terms of trace but this understanding was hidden in the mathematical setting - Hilbert spaces and traces of operators - that he used. Several people had the idea that Girard's Geometry of Interaction construction could be understood in terms of a trace operator on a monoidal category. Two well known accounts of this appear in (Joyal, Street and Verity, 1996b) and (Abramsky, 1996).

We will just give the main definition, for more details see (Joyal, Street and Verity, 1996b; Abramsky, 1996). Essentially, one obtains a (linear) higher-order model by working with processes with bi-directional "input" and "output". These processes are implemented by uni-directional processes of the underlying category in the obvious way, regarding negative, i.e. reversed, input channels as output channels and negative output as input.

Definition 6.2. Given a traced monoidal category $\mathcal{C} = (\mathcal{C}, \otimes, I, \sigma, \mathsf{Tr})$ we define a new category $\mathcal{G}(\mathcal{C})$ as follows. The objects of $\mathcal{G}(\mathcal{C})$ are pairs of objects (A^+, A^-) of $\mathcal{C}$. A morphism $f : (A^+, A^-) \to (B^+, B^-)$ of $\mathcal{G}(\mathcal{C})$ is a $\mathcal{C}$-morphism $f : A^+ \otimes B^- \to B^+ \otimes A^-$, as illustrated in Fig. 6. Composition is implemented using composition, trace and symmetries of $\mathcal{C}$ to connect channels with same polarity, ie. for $g : (B^+, B^-) \to (C^+, C^-)$, $f; g$ is implemented, as illustrated in Fig. 7, by

$$\mathsf{Tr}^{B^-}_{A^+\otimes C^-, C^+\otimes A^-}((I_{A^+} \otimes \sigma); (f \otimes I_{C^-}); (I_{B^+} \otimes \sigma'); (g \otimes I_{A^-}); (I_{C^+} \otimes \sigma'')) \ ,$$

for the appropriate symmetries σ, σ' and σ''.

Note that $\mathcal{C}$ embeds into $\mathcal{G}(\mathcal{C})$ as arrows with no negative flow, mapping objects A to (A, I). A symmetric monoidal structure $\odot$ is defined on objects by $(A^+, A^-) \odot (B^+, B^-) = (A^+ \otimes B^+, B^- \otimes A^-)$. The unit of $\odot$ is the pair (I, I). An obvious duality is defined on objects by $(A^+, A^-)^* = (A^-, A^+)$, and on arrows by swapping the roles of channels as illustrated in Fig. 8. This defines a contravariant functor $(-)^* : \mathcal{G}(\mathcal{C}) \to \mathcal{G}(\mathcal{C})$. Internal hom sets are given

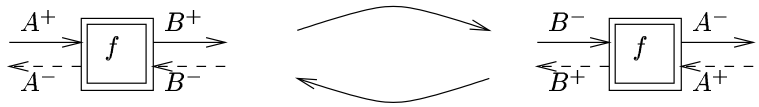


Fig. 8. The duality is obtained by swapping the roles of the channels

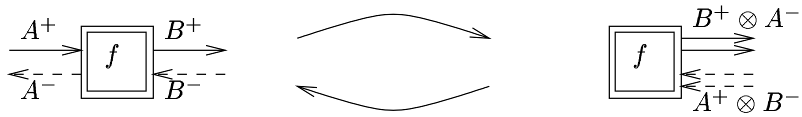


Fig. 9. Involution

by $(A^+, A^-) \multimap (B^+, B^-) = (B^+, B^-) \odot (A^+, A^-)^*$, giving an involution as illustrated by Fig. 9.

We can now directly apply the above construction to the category Kahn, obtaining a category $\mathcal{G}(\mathsf{Kahn})$ of *(linear) higher order Kahn processes*. Since $\mathsf{PProf_s}$ is a bicategory, only satisfying the axioms of a TMC up to isomorphism, $\mathcal{G}(\mathsf{PProf_s})$ will not be a category, e.g. composition is only associative up to isomorphism. There may be several ways around this problem. It is likely to be the case, that by making precise what it means to be a traced monoidal bicategory one can show that $\mathcal{G}(\mathsf{PProf_s})$ is a compact closed bicategory. This should be related to the work in (Katis et al., 1997). Another possibility is to consider the quotient of $\mathsf{PProf_s}$ with respect to open map bisimulation, analogous to the definition of the category $\mathcal{ASP}roc$ in (Abramsky et al., 1994; Gay, 1995). That is, instead of $\mathsf{PProf_s}$ use the category with objects being path categories as usual, but taking arrows to be equivalence classes with respect to open map bisimulation. By Thm. 6.2 this is indeed well defined and it is easy to check that we get a traced (strict) symmetric monoidal category. Since the pseudofunctor $\mathcal{S}eq$ from $\mathsf{PProf_s}$ to Kahn preserves tensor and trace, it extends to a (possibly pseudo)functor from $\mathcal{G}(\mathsf{PProf_s})$ to $\mathcal{G}(\mathsf{Kahn})$ using $\mathcal{S}eq$ on the base category.

The higher order structure of $\mathcal{G}(\mathsf{Kahn})$ and $\mathcal{G}(\mathsf{PProf_s})$ has a very intuitive interpretation in Kahn and $\mathsf{PProf_s}$ as plugging networks into contexts. As an example, consider the fork process $\mathcal{F}\colon \mathsf{H}_1 \mathrel{-\!\!+\!\!\!\rightarrow}_s \mathsf{H}_1 \otimes \mathsf{H}_1$ which was used in defining the context $\mathcal{C}[-]$ of Sec. 1. It implements the higher order process $\mathcal{F}\colon (\mathsf{H}_1, \mathsf{H}_1) \to (\mathsf{H}_1, I)$, that is, a process $\mathcal{F}\colon (\mathsf{H}_1 \multimap \mathsf{H}_1) \to \mathsf{H}_1$ writing H_1 as short for (H_1, I). The processes $\mathcal{A}_i\colon \mathsf{H}_1 \mathrel{-\!\!+\!\!\!\rightarrow}_s \mathsf{H}_1$ implements higher order processes $\mathcal{A}_i\colon (\mathsf{H}_1, I) \to (\mathsf{H}_1, I)$ which by involution can be regarded as processes $\mathcal{A}_i\colon I \to (\mathsf{H}_1 \multimap \mathsf{H}_1)$, again writing H_1 as short for (H_1, I). Now the processes $\mathcal{C}[\mathcal{A}_i]$ are simply the processes $\mathcal{A}_i; \mathcal{F}$ obtained by composition as illustrated in Fig. 10.

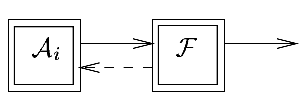


Fig. 10. The fork process $\mathcal{F}$ regarded as a higher-order process, applied to the automata $\mathcal{A}_i$

7. Concluding Remarks

The upshot of the work in this paper is a treatment of non-deterministic dataflow within the framework of categorical models for concurrency that unifies different phenomena - asynchrony and synchrony in our case - and different viewpoints of dataflow networks: dataflow composition as relational composition, dataflow processes as categorical constructs and the concrete views of dataflow networks as port automata and as sequences of events encoding causality. In particular, we obtain a bisimulation congruence from the bisimulation from open maps approach, and dataflow feedback is shown as an instance of a trace operation in a category. This allows one to adapt the ideas from the geometry of interaction program to give a smooth treatment of higher-order processes. The development was carried out for dataflow networks between simple stream domains for ease of presentation. However, the results only depend on the domains having a countable basis satisfying the descending chain condition.

It remains to explore systematically the full family of models for dataflow, relating automata, event structure and traces-based models to the relational model, following the pattern set in (Winskel and Nielsen, 1995). This analysis would include the models of *scenarios* studied in (Brock, 1983; Brock and Ackerman, 1981) and the generalized models in (Abramsky, 1990).

The seemingly close relationship between the model of stable port profunctors and the much more concrete model in (Staples and Nguyen, 1985) should also be investigated.

It is an interesting question to find a more abstract characterisation of the trace in the profunctor model; the present one relies much on the concrete representation of profunctors as port-automata. A step in this direction is taken in (Hildebrandt, 2000), where it is shown that the trace satisfies a *uniformity* property suggested in the thesis of Hasegawa (Hasegawa, 1997a) and also studied in (Selinger, 1999; Hasegawa, 2003) as a generalization of uniformity for fixed-point operators. Another possibility is to explore the relationship to stable domains and generalise the approach in (Abramsky, 1990).

The simplification to disregard fairness helped identifying the category of profunctors as a relational model of indeterminate dataflow. We hope to incorporate fairness into the profunctor model along the lines in (Hildebrandt, 1999a; Hildebrandt, 1999b) using *separated* presheaves over path categories including *completed*, i.e. possibly infinite computations.

Appendix A. Traced Monoidal Properties

We show here that the trace as given by Def. 5.2 satisfies all the properties of a traced monoidal category up to isomorphism.

We first introduce a more general notion of trace, in which two arbitrary ports can be connected. For $X\colon \mathsf{H}_\mathsf{n} \relbar\!\!\mapsto_s \mathsf{H}_\mathsf{m}$, we specify which ports are connected by a partial isomorphism $f\colon [n] \overset{\sim}{\rightharpoonup} [m]$. Writing $do(f)$ for the domain of f and $co(f)$ for the codomain of f, the intuition is that if $c \in do(f)$ the input port c is connected to output port $f(c)$ by a feedback wire.

For the definition of the generalised trace we generalise the communication relation as follows. Let $X\colon \mathsf{H}_\mathsf{n} \relbar\!\!\mapsto_s \mathsf{H}_\mathsf{m}$ and $\mathcal{A}(X) = (S, r, \longrightarrow, [n], [m])$, and let $f\colon [n] \overset{\sim}{\rightharpoonup} [m]$ be a partial isomorphism. We then let $s \overset{c,v}{\Longrightarrow}_f s'$, *communication via* f, denote that $s \xrightarrow{\mathsf{o}\, c',v} \xrightarrow{\mathsf{i}\, c,v} s'$ for some $c \in do(f)$ where $f(c) = c'$, i.e. a communication via f consists of a transition that outputs a value on a port in the codomain of f followed by a transition that inputs that value on the, via f, corresponding input port. Let $\sim_f$ denote the least equivalence relation including $\overset{f}{\Longrightarrow}$. Now, we say that a state $s \in S$ is f*-secured* if

$$r \xrightarrow{\mathsf{i}\alpha}\Longrightarrow_f^* \xrightarrow{\mathsf{o}\beta} s \quad ,$$

for $\alpha \in ([n]\backslash do(f) \times \mathcal{V})^*$ and $\beta \in ([m]\backslash co(f) \times \mathcal{V})^*$.

As for the securedness defined in Sec. 4.2, f-securedness has some useful equivalent formulations. In particular, if $P = [n] \backslash do(f)$ and $Q = [m] \backslash co(f)$, then a state $s \in S$ is f-secured if and only if $r(\xrightarrow{\mathsf{i}P}\sim_f\xrightarrow{\mathsf{o}Q})^* s$, where $x \xrightarrow{\mathsf{i}P} x'$ denote that $x \xrightarrow{\mathsf{i}\,a,v} x'$ for some $a \in P$ and $x \xrightarrow{\mathsf{o}Q} x'$ denote that $x \xrightarrow{\mathsf{o}\,b,v} x'$ for some $b \in Q$.

For a set of indexes $N \subseteq [n]$ and $j \in N$, we define $j{\downarrow}N = |\{j' \in N \mid j' < j\}|$, i.e. $j{\downarrow}N$ is the number of elements below j in N.

Definition A.1. Let $X\colon \mathsf{H_n} \otimes \mathsf{H_p} \!-\!\!\mapsto_s \mathsf{H_m} \otimes \mathsf{H_p}$ and let $f\colon [n+p] \overset{\sim}{\rightharpoonup} [m+p]$ be a partial isomorphism such that $|do(f)| = p$. Define $\mathsf{Tr}^f_{\mathsf{H_n},\mathsf{H_m}}(X)\colon \mathsf{H_n} \!-\!\!\mapsto \mathsf{H_m}$, the generalised trace of X to be given by

$$\mathsf{Tr}^f_{\mathsf{H_n},\mathsf{H_m}}(X)(\overline{\alpha},\overline{\beta}) = \biguplus_{\overline{\gamma}\in\mathsf{H_p}} \{\, x \in X(\overline{\phi},\overline{\psi}) \mid x \text{ is } f\text{-secured and } Perm_f(\overline{\phi},\overline{\psi},\overline{\alpha},\overline{\beta},\overline{\gamma}) \,\}_{/\sim_f}, \tag{21}$$

where $Perm_f$ is the predicate on $\mathsf{H_{n+p}} \times \mathsf{H_{m+p}} \times \mathsf{H_n} \times \mathsf{H_m} \times \mathsf{H_p}$ defined by

$$Perm_f(\overline{\phi},\overline{\psi},\overline{\alpha},\overline{\beta},\overline{\gamma})$$

if and only if

$$\overline{\phi}(i) = \begin{cases} \overline{\gamma}\big(f(i){\downarrow}co(f)\big) & \text{if } i \in do(f), \\ \overline{\alpha}\big(i{\downarrow}[n+p]\backslash do(f)\big) & \text{if } i \notin do(f). \end{cases}$$

and

$$\overline{\psi}(i) = \begin{cases} \overline{\gamma}\big(i{\downarrow}co(f)\big) & \text{if } i \in co(f), \\ \overline{\beta}\big(i{\downarrow}[m+p]\backslash co(f)\big) & \text{if } i \notin co(f), \end{cases}$$

i.e. the predicate $Perm_f$ garantees that the ports are permutated correctly. The action on arrows is defined as for the standard trace. We will usually abbreviate $\mathsf{Tr}^f_{\mathsf{H_n},\mathsf{H_m}}$ to $\mathsf{Tr}^f_{n,m}$ or just Tr^f.

The following lemma states a generalised Yanking property, the straightforward proof is omitted.

Lemma A.1 (Generalized Yanking). Let $X\colon \mathsf{H_n} \!-\!\!\mapsto_s \mathsf{H_p}$ and $Y\colon \mathsf{H_p} \!-\!\!\mapsto_s \mathsf{H_m}$. Then

$$X;Y \cong \mathsf{Tr}^f(X \otimes Y) \cong \mathsf{Tr}^p_{n,m}(X \otimes Y;\sigma_{p,m}) \quad ,$$

for $f\colon [n+p] \overset{\sim}{\rightharpoonup} [m+p]$ the partial isomorphism defined by $f(i) = i+m$ if $i \in [p]$.

The lemma below is a simple consequence of the definition of generalised securedness.

Lemma A.2. Let $X\colon \mathsf{H_n} \!-\!\!\mapsto_s \mathsf{H_m}$ and $f, g\colon [n] \overset{\sim}{\rightharpoonup} [m]$ partial isomorphisms such that $do(g) \subseteq do(f)$ and codomain $co(g) \subseteq co(f)$. Then a state s of $\mathcal{A}(X)$ is f*-secured* only if it is g*-secured*.

The lemma below is the crucial step in establishing the generalised Bekic property.

Lemma A.3. Let $X\colon \mathsf{H_n} \multimap_s \mathsf{H_m}$ and let $f, g\colon [n] \overset{\sim}{\rightharpoonup} [m]$ partial isomorphisms such that $do(f) \cap do(g) = \varnothing$ and $co(f) \cap co(g) = \varnothing$. Then

$$s \sim_g \overset{\circ f(c),v}{\longrightarrow} \sim_g \overset{\mathsf{i}\, c,v}{\longrightarrow} \sim_g s' \text{ if and only if } s \sim_g \overset{\circ f(c),v}{\longrightarrow}\overset{\mathsf{i}\, c,v}{\longrightarrow} \sim_g s'$$

Proof. The if direction is immediate. For the only if direction, note that by definition

$$s \sim_g \overset{\circ f(c),v}{\longrightarrow} \sim_g \overset{\mathsf{i}\, c,v}{\longrightarrow} \sim_g s'$$

if and only if

$$s \sim_g \overset{\circ f(c),v}{\longrightarrow} (\Longrightarrow_g^{-1} \cup \Longrightarrow_g)^k \overset{\mathsf{i}\, c,v}{\longrightarrow} \sim_g s', \tag{22}$$

for some $k \in \omega$, where $\Longrightarrow_g^{-1}$ is the inverse of $\Longrightarrow_g$, i.e. $s \Longrightarrow_g^{-1} s'$ iff $s' \Longrightarrow_g s$. We proceed by induction in k, showing

$I(k)$ $\quad s \sim_g \overset{\circ f(c),v}{\longrightarrow} (\Longrightarrow_g^{-1} \cup \Longrightarrow_g)^k \overset{\mathsf{i}\, c,v}{\longrightarrow} \sim_g s'$ implies $s \sim_g \overset{\circ f(c),v}{\longrightarrow}\overset{\mathsf{i}\, c,v}{\longrightarrow} \sim_g s'$.

For $k = 0$ the desired result follows immediately. Now assume $k = k' + 1$ and $I(k')$ holds. Then there exists a state s'' such that $s''(\Longrightarrow_g^{-1} \cup \Longrightarrow_g)^{k'} \overset{\mathsf{i}\, c,v}{\longrightarrow} \sim_g s'$ and ($s \sim_g \overset{\circ f(c),v}{\longrightarrow} \Longrightarrow_g^{-1} s''$ or $s \sim_g \overset{\circ f(c),v}{\longrightarrow} \Longrightarrow_g s''$). If $s \sim_g \overset{\circ f(c),v}{\longrightarrow} \Longrightarrow_g^{-1} s''$ then it follows from Ax. U1-U2 and $co(g) \cap co(f) = \varnothing$ that $s \sim_g \Longrightarrow_g^{-1} s''' \overset{\circ f(c),v}{\longrightarrow} s''$, which gives that $s \sim_g s''' \overset{\circ f(c),v}{\longrightarrow} s''(\Longrightarrow_g^{-1} \cup \Longrightarrow_g)^{k'} \overset{\mathsf{i}\, c,v}{\longrightarrow} \sim_g s'$. By $I(k')$ we then get $s \sim_g \overset{\circ f(c),v}{\longrightarrow}\overset{\mathsf{i}\, c,v}{\longrightarrow} \sim_g s'$, which was the desired result. If $s \sim_g \overset{\circ f(c),v}{\longrightarrow} \Longrightarrow_g s''$then it follows from Ax. A2', A3-A4 and $co(g) \cap co(f) = \varnothing$ that $s \sim_g \Longrightarrow_g s''' \overset{\circ f(c),v}{\longrightarrow} s''$. As before it then follows by induction that $s \sim_g \overset{\circ f(c),v}{\longrightarrow}\overset{\mathsf{i}\, c,v}{\longrightarrow} \sim_g s'$. □

The scene is now set to prove the generalized Bekic property.

Proposition A.4 (Generalized Bekic). Let $X\colon \mathsf{H_n} \multimap_s \mathsf{H_m}$, and let $f, g\colon [n] \overset{\sim}{\rightharpoonup} [m]$ partial isomorphisms such that $do(f) \cap do(g) = \varnothing$ and $co(f) \cap co(g) = \varnothing$. Then there is an isomorphism

$$\mathsf{Tr}^{f \cup g}(X) \cong \mathsf{Tr}^{f \setminus g}(\mathsf{Tr}^g(X)),$$

where $f \cup g\colon [n] \overset{\sim}{\rightharpoonup} [m]$ is the obvious union of the two partial isomorphisms, and $f \setminus g\colon [n-p] \overset{\sim}{\rightharpoonup} [m-p]$ for $p = |do(g)|$ is the partial isomorphism defined by $f \setminus g(i) = f(j)$, if there exists $j \in do(f)$ such that $j \downarrow [n] \setminus do(g) = i$.

Proof. Let $\mathcal{A}(X) = (S, r, \longrightarrow, [n], [m])$ and $\mathcal{A}(\mathsf{Tr}^g(X)) = (S', r', \longrightarrow', [n-p], [m-p])$. It is sufficient to show that

$$x \in S \text{ is } f \cup g\text{-secured } \Leftrightarrow [x]_{\sim_g} \in S' \text{ and } [x]_{\sim_g} \text{ is } f \setminus g\text{-secured.} \tag{23}$$

Let $x \Longrightarrow_{f_{\sim g}} x'$ denote that $x \sim_g \overset{\circ c,v}{\longrightarrow} \sim_g \overset{\mathsf{i}\, c,v}{\longrightarrow} \sim_g x'$, for some $c \in do(f)$. The relation $\Longrightarrow_{f_{\sim g}}$ can be read as "communication via f upto back&forth communication via g". Let $P = [n] \setminus do(f \cup g)$ and $Q = [m] \setminus co(f \cup g)$, and let $x \overset{\mathsf{i}P}{\longrightarrow}_{\sim g} x'$ denote that $x \sim_g \overset{\mathsf{i}\, a,v}{\longrightarrow} \sim_g x'$ for some $a \in P$ and $x \overset{\circ Q}{\longrightarrow}_{\sim g} x'$ denote that $x \sim_g \overset{\circ b,v}{\longrightarrow} \sim_g x'$ for some $b \in Q$. Then the right hand side of (23) is equivalent to

$$x \text{ is } g\text{-secured} \quad \text{and} \quad r(\overset{\mathsf{i}P}{\longrightarrow}_{\sim g} \cup \sim_{f \sim g} \cup \overset{\circ Q}{\longrightarrow}_{\sim g})^* x \tag{24}$$

where $\sim_{f\sim g}$ denote the least equivalence relation including $\Longrightarrow_{f_{\sim g}}$ and $\sim_g$. For $\sim_{f\cup g}$ the least equivalence relation including $\Longrightarrow_{f\cup g}$ we now show that $\sim_{f\cup g}=\sim_{f\sim g}$. From the fact that

$$\begin{aligned}\Longrightarrow_{f\cup g} &= (\Longrightarrow_f \cup \Longrightarrow_g)\\ &\subseteq (\Longrightarrow_{f_{\sim g}} \cup \sim_g),\end{aligned}$$

we get that $\sim_{f\cup g}\subseteq\sim_{f\sim g}$. From Lem. A.3 we get that $\Longrightarrow_{f_{\sim g}} = (\sim_g\Longrightarrow_f\sim_g)$. From the fact that $(\sim_g\Longrightarrow_f\sim_g \cup \sim_g) \subseteq\sim_{f\cup g}$ it then follows that $\sim_{f\sim g}\subseteq\sim_{f\cup g}$ and so $\sim_{f\cup g}=\sim_{f\sim g}$.
Now, since $\sim_g\subseteq\sim_{f\cup g}$ we get that (24) is equivalent to

$$x \text{ is } g\text{-secured} \quad \text{and} \quad r(\xrightarrow{\mathsf{i}P} \cup \sim_{f\cup g} \cup \xrightarrow{\mathsf{o}Q})^* x \tag{25}$$

which by Lem. A.2 is equivalent to that x is $f \cup g$-secured. □

We now show a generalised Superposing property.

Lemma A.5 (Generalised Superposing). Let $X_i\colon \mathsf{H}_{\mathsf{n_i}} \multimap\!\!\!\to_s \mathsf{H}_{\mathsf{m_i}}$ for $i \in \{1,2\}$ and let $f\colon [n_1 + n_2] \overset{\sim}{\rightharpoonup} [m_1 + m_2]$ a partial isomorphism such that $do(f) \subseteq [n_2]$ and $co(f) \subseteq [m_2]$. Then there are isomorphisms

$$\mathsf{Tr}^f(X_1 \otimes X_2) \cong X \otimes \mathsf{Tr}^{f_{|n_2}}(X_2) \tag{26}$$

and

$$\mathsf{Tr}^{f'}(X_2 \otimes X_1) \cong \mathsf{Tr}^{f_{|n_2}}(X_2) \otimes X \ \ , \tag{27}$$

where $f_{|n_2}\colon [n_2] \overset{\sim}{\rightharpoonup} [m_2]$ is the restriction of f to the domain $[n_2]$ and $f'(j) = f(j - n_1) + m_1$, if $j - n_1 \in do(f)$.

Proof. We will just prove the existence of the first isomorphism, the proof for the second is analogous. Let $\mathcal{A}(X_i) = (S_i, r_i, \longrightarrow, [n_i], [m_i])$. Assume $\mathcal{A}(X_1 \otimes X_2) \cong (S_1 \times S_2, (r_1, r_2), \longrightarrow, [n_1 + n_2], [m_1 + m_2])$ as given by Lem. 5.14.

It is sufficient to show that for all $(x_1, x_2) \in S_1 \times S_2$

$$r \xrightarrow{\mathsf{i}\alpha}\Longrightarrow_f^*\xrightarrow{\mathsf{o}\beta} (x_1, x_2),$$

for $\alpha \in [n_1 + n_2] \backslash do(f)$ and $\beta' \in [m_1 + m_2] \backslash co(f)$ if and only if

$$r \xrightarrow{\mathsf{i}\alpha'}_2\Longrightarrow^*_{f_{|n_2}}\xrightarrow{\mathsf{o}\beta'}_2 x_2 \text{ and } r \xrightarrow{\mathsf{i}\alpha''}_1\xrightarrow{\mathsf{o}\beta''}_1 x_1,$$

for $\alpha' \in [n_2] \backslash do(f_{|n_2})$ and $\beta' \in [m_2] \backslash co(f_{|n_2})$. □

The notation for the generalised trace can be used for coends as well. Let $X\colon \mathsf{H_n} \multimap\!\!\!\to_s \mathsf{H_m}$ and $f\colon [n] \overset{\sim}{\rightharpoonup} [m]$ be a partial isomorphism defined on a single element $i \in [n]$. Then define

$$\textstyle\int^f(X) = \displaystyle\int^x X(-_0, \ldots, -_{i-1}, x, -_{i+1}, \ldots, -_{n-1})(-_0, \ldots, -_{f(i)-1}, x, -_{f(i)+1}, \ldots, -_{m-1})$$

The definition is generalised in the obvious way to general partial isomorphisms defined on more than one element. Using the following lemma together with the Generalised Bekic, Yanking and Superposing properties, the naturality axioms can then be inferred from isomorphisms known to exist for coends (Hyland, 1995).

Lemma A.6. Let $X\colon \mathsf{H_n} \multimap\!\!\!\to_s \mathsf{H_m}$, $Y\colon \mathsf{H_p} \multimap\!\!\!\to_s \mathsf{H_q}$ and let $f\colon [p + n] \overset{\sim}{\rightharpoonup} [q + m]$ be a partial isomorphism such that either $do(f) \subseteq [p]$ and $co(f) \subseteq [m+q]\backslash[q]$ or $do(f) \subseteq [n+p]\backslash[p]$ and $co(f) \subseteq [q]$. Then there is an isomorphism

$$\mathsf{Tr}^f(X \otimes Y) \cong \int^f (X \otimes Y) \ .$$

Proof. (Sketch) Let $\mathcal{A}(X) = (S, r, \longrightarrow, [n+p], [m+q])$. Using Lem. 5.14 one shows that for all $s, s' \in S$

$$s \stackrel{\mathsf{i}\,c,v}{\longrightarrow}\stackrel{\mathsf{o}\,f(c),v}{\longrightarrow} s' \text{ for some } c \in do(f) \text{ implies that } s \stackrel{\mathsf{o}\,f(c),v}{\longrightarrow}\stackrel{\mathsf{i}\,c,v}{\longrightarrow} s' \ .$$

It follows that the generalized securedness condition in the definition of the trace will hold. for all elements of $X \otimes Y$ □

Appendix B. Congruence Properties of Bisimulation

We here prove that trace and tensor preserves open maps as promised in Sec. 6.2. This implies that open map bisimulation is a congruence with respect to feedback and parallel composition.

The crucial property we will use is that any colimit-preserving functor between presheaf categories preserves (surjective) open maps (Cattani and Winskel, 1997).

We first show that the trace operator preserves open maps. Recall that the notion of open maps apply canonically to *any* presheaf category, why we also get a notion of open map between presheaves over $(\mathsf{H_n} \times \mathsf{H_p}^{\S})^{\mathsf{op}} \times \mathsf{H_m}$. The desired result then follows from Prop. 5.17 if we can show that the functor $\mathcal{S}$ defined in App. 5.1 preserves (surjective) open maps.

Lemma B.1. Let $X, Y \colon \mathsf{H_n} \otimes \mathsf{H_p} \multimap\!\!\!\rightarrow_s \mathsf{H_m} \otimes \mathsf{H_p}$. If $f \colon X \to Y$ is an (surjective) *open* natural transformation then $\mathcal{S}(f) \colon \mathcal{S}(X) \to \mathcal{S}(Y)$ is (surjective) open too.

Proof. Let $f \colon X \to Y$ be an (surjective) *open* natural transformation. This means that (f is surjective and) for any arrow $e \colon P \to Q$ of $(\mathsf{H_n} \times \mathsf{H_p}^{\S})^{\mathsf{op}} \times \mathsf{H_m}$ the square

$$\begin{array}{ccc} X(Q) & \xrightarrow{Xe} & X(P) \\ {\scriptstyle f_Q}\big\downarrow & & \big\downarrow{\scriptstyle f_P} \\ Y(Q) & \xrightarrow{Ye} & Y(P) \end{array} \tag{28}$$

is a quasi-pullback. We must then show that ($\mathcal{S}(f)$ is surjective and) for any arrow $e \colon P \to Q$ of $(\mathsf{H_n} \times \mathsf{H_p}^{\S})^{\mathsf{op}} \times \mathsf{H_m}$ the square

$$\begin{array}{ccc} \mathcal{S}(X)(Q) & \xrightarrow{\mathcal{S}(X)e} & \mathcal{S}(X)(P) \\ {\scriptstyle \mathcal{S}(f)_Q}\big\downarrow & & \big\downarrow{\scriptstyle \mathcal{S}(f)_P} \\ \mathcal{S}(Y)(Q) & \xrightarrow{\mathcal{S}(Y)e} & \mathcal{S}(Y)(P) \end{array} \tag{29}$$

is a quasi-pullback. Subjectiveness is easily verified. For the quasi-pullback property it follows by induction that it is enough to consider a class of arrows from which all arrows can be obtained by finitely many compositions. Consequently, it suffices to consider the cases

1. e is an iso, i.e. $e = (1_{\overline{\gamma}}, 1_{[\overline{\alpha},\overline{\beta}]}, 1_{\overline{\delta}}) \colon (\overline{\gamma}, [\overline{\alpha}, \overline{\beta}], \overline{\delta}) \to (\overline{\gamma}, [\overline{\alpha}, \overline{\beta}], \overline{\delta})$,

2. $e = ([\overline{\gamma}, \overline{\gamma}'], 1_{[\overline{\alpha},\overline{\beta}]}, 1_{\overline{\delta}})\colon (\overline{\gamma}, [\overline{\alpha}, \overline{\beta}], \overline{\delta}) \to (\overline{\gamma}', [\overline{\alpha}, \overline{\beta}], \overline{\delta})$, for $\overline{\gamma}' \leq \overline{\gamma}$ in $\mathsf{H_n}$, or
3. $e = (1_{\overline{\gamma}}, 1_{[\overline{\alpha},\overline{\beta}]}, [\overline{\delta}, \overline{\delta}'])\colon (\overline{\gamma}, [\overline{\alpha}, \overline{\beta}], \overline{\delta}) \to (\overline{\gamma}, [\overline{\alpha}, \overline{\beta}], \overline{\delta}')$, for $\overline{\delta} \leq \overline{\delta}'$ in $\mathsf{H_m}$, or
4. $e = (1_{\overline{\gamma}}, g_{\mathsf{i}}, 1_{\overline{\delta}})\colon (\overline{\gamma}, [\overline{\beta}, \overline{\beta}], \overline{\delta}) \to (\overline{\gamma}, [\overline{\alpha}, \overline{\beta}], \overline{\delta})$, for g_{i} the opposite of $[\overline{\alpha}, \overline{\beta}]_{\mathsf{i}}\colon [\overline{\alpha}, \overline{\beta}] \to [\overline{\beta}, \overline{\beta}]$ in $\mathsf{H_p}^{\S}$ or
5. $e = (1_{\overline{\gamma}}, g_{\mathsf{o}}, 1_{\overline{\delta}})\colon (\overline{\gamma}, [\overline{\alpha}, \overline{\alpha}], \overline{\delta}) \to (\overline{\gamma}, [\overline{\alpha}, \overline{\beta}], \overline{\delta})$, for g_{o} the opposite of $[\overline{\alpha}, \overline{\beta}]_{\mathsf{o}}\colon [\overline{\alpha}, \overline{\beta}] \to [\overline{\alpha}, \overline{\alpha}]$ in $\mathsf{H_p}^{\S}$.

Case 1 is trivial. Intuitively, case 2 and case 3 express respectively that backtracking of input on ports in $\mathsf{H_n}$ and extension of output on ports in $\mathsf{H_m}$ in $\mathcal{S}(Y)$ can be matched in $\mathcal{S}(X)$. Similarly, case 4 and case 5 express respectively that backtracking of input(communication) on ports in $\mathsf{H_p}$ and extension of output(communication) on ports in $\mathsf{H_p}$ in $\mathcal{S}(Y)$ can be matched in $\mathcal{S}(X)$.

For the cases 2-5, we use that the diagram 28 is a quasi-pullback and use axioms U1,U2 repeatedly. We will only go through case 2, the other cases are treated in a similar fashion.

We want to show that

$$\begin{array}{ccc} \mathcal{S}(X)(\overline{\gamma}', [\overline{\alpha}, \overline{\beta}], \overline{\delta}) & \xrightarrow{\mathcal{S}(X)([\overline{\gamma}', \overline{\gamma}], 1_{[\overline{\alpha},\overline{\beta}]}, 1_{\overline{\delta}})} & \mathcal{S}(X)(\overline{\gamma}, [\overline{\alpha}, \overline{\beta}], \overline{\delta}) \\ {\scriptstyle \mathcal{S}(f)_{(\overline{\gamma}', [\overline{\alpha},\overline{\beta}], \overline{\delta})}} \downarrow & & \downarrow {\scriptstyle \mathcal{S}(f)_{(\overline{\gamma}, [\overline{\alpha},\overline{\beta}], \overline{\delta})}} \\ \mathcal{S}(Y)(\overline{\gamma}', [\overline{\alpha}, \overline{\beta}], \overline{\delta}) & \xrightarrow{\mathcal{S}(Y)([\overline{\gamma}', \overline{\gamma}], 1_{[\overline{\alpha},\overline{\beta}]}, 1_{\overline{\delta}})} & \mathcal{S}(Y)(\overline{\gamma}, [\overline{\alpha}, \overline{\beta}], \overline{\delta}) \end{array} \tag{30}$$

is a quasi-pullback. Assume that $x \in \mathcal{S}(X)(\overline{\gamma}, [\overline{\alpha}, \overline{\beta}], \overline{\delta})$, $y \in \mathcal{S}(Y)(\overline{\gamma}, [\overline{\alpha}, \overline{\beta}], \overline{\delta})$ and $y' \in \mathcal{S}(Y)(\overline{\gamma}', [\overline{\alpha}, \overline{\beta}], \overline{\delta})$, such that $\mathcal{S}(f)_{(\overline{\gamma}', [\overline{\alpha},\overline{\beta}], \overline{\delta})} x = y$ and $\mathcal{S}(Y)([\overline{\gamma}', \overline{\gamma}], 1_{[\overline{\alpha},\overline{\beta}]}, 1_{\overline{\delta}}) y' = y$. We must then show that there exists $x' \in \mathcal{S}(X)(\overline{\gamma}', [\overline{\alpha}, \overline{\beta}], \overline{\delta})$ such that

$$\mathcal{S}(X)([\overline{\gamma}', \overline{\gamma}], 1_{[\overline{\alpha},\overline{\beta}]}, 1_{\overline{\delta}}) x' = x \tag{31}$$

and

$$\mathcal{S}(f)_{(\overline{\gamma}, [\overline{\alpha},\overline{\beta}], \overline{\delta})} x' = y'. \tag{32}$$

Now, by using the definition of $\mathcal{S}$ the assumption gives us that

$$x \in X\big((\overline{\gamma}, \overline{\alpha}), (\overline{\beta}, \overline{\delta})\big), y \in Y\big((\overline{\gamma}, \overline{\alpha}), (\overline{\beta}, \overline{\delta})\big) \text{ and } y' \in Y\big((\overline{\gamma}', \overline{\alpha}), (\overline{\beta}, \overline{\delta})\big),$$

such that $f_{(\overline{\gamma}', \overline{\alpha}, \overline{\beta}, \overline{\delta})} x = y$ and $Y\big(([\overline{\gamma}', \overline{\gamma}], 1_{\overline{\alpha}}), (1_{\overline{\beta}}, 1_{\overline{\delta}})\big) y' = y$. Furthermore we have that $w = X\big((1_{\overline{\gamma}}, 1_{\overline{\alpha}}), ([\overline{\beta}, \overline{\alpha}], 1_{\overline{\delta}})\big) x$ is p-secured. By Eq. 28 being a quasi-pullback, there exists an $x' \in X\big((\overline{\gamma}', \overline{\alpha}), (\overline{\beta}, \overline{\delta})\big)$ such that $X\big(([\overline{\gamma}', \overline{\gamma}], 1_{\overline{\alpha}}), (1_{\overline{\beta}}, 1_{\overline{\delta}})\big) x' = x$ and $f_{(\overline{\gamma}, \overline{\alpha}, \overline{\beta}, \overline{\delta})} x' = y'$. If we can show that $x' \in \mathcal{S}(X)(\overline{\gamma}', [\overline{\alpha}, \overline{\beta}], \overline{\delta})$ then Eq. 32 and Eq. 31 follows by definition and the proof is completed. We only need to show that $w' = X\big((1_{\overline{\gamma}'}, 1_{\overline{\alpha}}), ([\overline{\beta}, \overline{\alpha}], 1_{\overline{\delta}})\big) x'$ is p-secured. Now, note that $w \xrightarrow{\mathsf{o}\phi_1} \xrightarrow{\mathsf{o}\phi_2} \cdots \xrightarrow{\mathsf{o}\phi_n} x$ for $\overline{\beta} = \overline{\alpha\phi_1\phi_2 \ldots \phi_n}$, and $x' \xrightarrow{\mathsf{i}\psi_1} \xrightarrow{\mathsf{i}\psi_2} \cdots \xrightarrow{\mathsf{i}\psi_n} x$ for $\overline{\gamma} = \overline{\gamma\psi_1\psi_2 \ldots \psi_n}$. By repeated use of Ax. U2 we get $z \xrightarrow{\mathsf{o}\phi_1} \xrightarrow{\mathsf{o}\phi_2} \cdots \xrightarrow{\mathsf{o}\phi_n} x'$ such that $z \xrightarrow{\mathsf{i}\psi_1} \xrightarrow{\mathsf{i}\psi_2} \cdots \xrightarrow{\mathsf{i}\psi_n} w$. Since w is p-secured it finally follows by repeated use Axioms U1,U2 and A4 that z is p-secured. Finally, since $w' \xrightarrow{\mathsf{o}\phi_1} \xrightarrow{\mathsf{o}\phi_2} \cdot \xrightarrow{\mathsf{o}\phi_n} x'$ it follows by repeated use of Ax. U1 that $w' = z$. □

Proposition B.2. Let $X, Y\colon \mathsf{H_n} \otimes \mathsf{H_p} \multimap\!\!\!\to_s \mathsf{H_m} \otimes \mathsf{H_p}$. If $f\colon X \to Y$ is an *open* natural transformation then $\mathsf{Tr}^p_{n,m}(f)\colon \mathsf{Tr}^p_{n,m}(X) \to \mathsf{Tr}^p_{n,m}(Y)$ is open too.

Proof. Note that $\mathrm{Colim}_{\mathsf{H_p}^\S}$ is obviously a colimit preserving functor between presheaf categories, i.e. it preserves (surjective) open maps. The desired result then follows from Prop. 5.17 and Lem. B.1. □

The proof that tensor preserves (surjective) open maps is simpler than the one for trace, we simply show that tensor can be defined from colimit preserving functors.

Proposition B.3. Let $X, X' : \mathsf{H}_{\mathsf{n}_1} \multimap\!\!\to_s \mathsf{H}_{\mathsf{m}_1}$ and $Y, Y' : \mathsf{H}_{\mathsf{n}_2} \multimap\!\!\to_s \mathsf{H}_{\mathsf{m}_2}$. If $f : X \to X'$ and $g : Y \to Y'$ are *open* natural transformations then $f \otimes g : X \otimes Y \to X' \otimes Y'$ is open too.

Proof. The tensor of $X : \mathsf{H}_{\mathsf{n}_1} \multimap\!\!\to_s \mathsf{H}_{\mathsf{m}_1}$ and $Y : \mathsf{H}_{\mathsf{n}_2} \multimap\!\!\to_s \mathsf{H}_{\mathsf{m}_2}$ can be expressed as a product of presheaves over ${\mathsf{H}_{\mathsf{n}_1}}^{\mathsf{op}} \times \mathsf{H}_{\mathsf{m}_1} \times {\mathsf{H}_{\mathsf{n}_2}}^{\mathsf{op}} \times \mathsf{H}_{\mathsf{m}_2}$ by $X \otimes Y = (\pi_1^* X) \times (\pi_2^* Y)$, where e.g.

$$\pi_1^* : [\mathsf{H}_{\mathsf{n}_1} \times {\mathsf{H}_{\mathsf{m}_1}}^{\mathsf{op}}, \mathsf{Set}] \to [\mathsf{H}_{\mathsf{n}_1} \times {\mathsf{H}_{\mathsf{m}_1}}^{\mathsf{op}} \times \mathsf{H}_{\mathsf{n}_2} \times {\mathsf{H}_{\mathsf{m}_2}}^{\mathsf{op}}, \mathsf{Set}]$$

is obtained by composition with the projection

$$\pi_1 : \mathsf{H}_{\mathsf{n}_1} \times {\mathsf{H}_{\mathsf{m}_1}}^{\mathsf{op}} \times \mathsf{H}_{\mathsf{n}_2} \times {\mathsf{H}_{\mathsf{m}_2}}^{\mathsf{op}} \to \mathsf{H}_{\mathsf{n}_1} \times {\mathsf{H}_{\mathsf{m}_1}}^{\mathsf{op}},$$

so $\pi_1^*(X)(\overline{\alpha}_1, \overline{\beta}_1)(\overline{\alpha}_2, \overline{\beta}_2) = X(\overline{\alpha}_1, \overline{\beta}_1)$. For general reasons π_1^* has a right adjoint (constructed as a right Kan extension - see (Mac Lane, 1971; Cattani and Winskel, 1997)). Thus π_1^* and, similarly, π_2^* are left adjoints and so preserve (surjective) open maps. Combined with the similar fact about product of presheaves we deduce that $\otimes$ preserves (surjective) open maps. □

References

Abramsky, S. (1990). A generalized kahn principle for abstract asynchronous networks. In *MFPS'89*, volume 442 of *LNCS*, pages 1–21. Springer.

Abramsky, S. (1996). Retracing some paths in process algebra. In *CONCUR'96*, volume 1119 of *LNCS*, pages 1–17.

Abramsky, S., Gay, S., and Nagarajan, R. (1994). Interaction categories and the foundations of typed concurrent programming. In *Proc. of the 1994 Marktoberdorf summer school*. Springer.

Bainbridge, E. S. (1976). Feedback and generalized logic. *Information and Control*, 31:75–96.

Berry, G. (1979). *Modèles complètement adéquats et stable des lambda-calculus typés*. PhD thesis, L'université Paris VII.

Borceux, F. (1994). *Handbook of categorical logic*, volume 1. Cambridge University Press.

Brock, J. and Ackerman, W. (1981). Scenarios: A model of non-determinate computation. In Diaz, J. and Ramos, I., editors, *Formalization of Programming Concepts*, volume 107 of *LNCS*. Springer.

Brock, J. D. (1983). *A Formal Model of Non-determinate Dataflow Computation*. PhD thesis, Laboratory for Computer Science, MIT.

Cattani, G. L. (1999). *Presheaf models for concurrency*. PhD thesis, Aarhus University.

Cattani, G. L. and Winskel, G. (1997). Presheaf models for concurrency. In *Computer Science Logic: Tenth international Workshop, CSL'96, Annual Conference of the EACSL. Selected Papers*, number 1258 in Lecture Notes in Computer Science, pages 58–75. Springer-Verlag.

Cattani, G. L. and Winskel, G. (1999). Presheaf models for CCS-like languages. Technical report, BRICS, RS-99-14.

Cattani, G. L. and Winskel, G. (2000). Profunctors, open maps and bisimulation. Manuscript.

Day, B. J. and Street, R. (1997). Monoidal bicategories and hopf algebroids. *Advances in Mathematics*, 129:99–157.

Dennis, J. B. (1974). First version of a dataflow procedure language. In Robinet, B., editor, *Proceedings Colloque sur la Programmation*, volume 19 of *LNCS*, pages 362–376. Springer.

Diekert, V. and Métivier, Y. (1997). *Handbook of Formal Languages.*, chapter Partial Commutation and Traces. Springer.

Faustini, A. A. (1982). An operational semantics for pure dataflow. In *ICALP'82*, volume 140 of *LNCS*, pages 212–224. Springer.

Gay, S. J. (1995). *Linear Types for Communicating Processes*. PhD thesis, University of London, Imperial College of Science, Technology and Medicine, Department of Computing.

Girard, J.-Y. (1989a). Geometry of interaction I: interpretation of system F. In et. al., F., editor, *Proceedings Logic Colloquium 88*, pages 221–260. North-Holland.

Girard, J.-Y. (1989b). Geometry of interaction II: deadlock free algorithms. In Martin-Lof, P. and Mints, G., editors, *Proceedings of COLOG 88*, number 417 in Lecture Notes In Computer Science, pages 76–93. Springer-Verlag.

Gunter, C. A. (1992). *Semantics of Programming Languages.* MIT Press.

Hasegawa, M. (1997a). *Models of Sharing Graphs: A Categorical Semantics of* let *and* letrec. PhD thesis, Department of Computer Science, The University of Edinburgh.

Hasegawa, M. (1997b). Recursion from cyclic sharing: traced monoidal categories and models of cyclic lambda calculi. In *Proceedings of TLCA'97*, volume 1210 of *LNCS*, pages 196–213.

Hasegawa, M. (2003). The uniformity principle on traced monoidal categories. In *Proceedings of CTCS'02*, volume 69 of *ENTCS*.

Hildebrandt, T. T. (1999a). *Categorical Models for Concurrency: Independence, Fairness and Dataflow*. PhD thesis, Department of Computer Scence, University of Aarhus, Denmark.

Hildebrandt, T. T. (1999b). A fully abstract presheaf semantics of SCCS with finite delay. Research Series RS-99-28, BRICS, Department of Computer Science, University of Aarhus. Appears in Proceedings of CTCS'99, ENTCS, Vol. 29.

Hildebrandt, T. T. (2000). A generalized kahn principle for a relational semantics of indeterminate dataflow. Extended abstract presented at MFPS XVI.

Hyland, M. (1995). Category theory. Handwritten lecture notes taken by C. Hillhorst.

Jones, V. F. (1985). A polynomial invariant for links via von neumann algebras. *Bull. Amer. Math. Soc.*, 129:103–112.

Jonsson, B. (1989). A fully abstract trace model for dataflow networks. In *POPL'89*, pages 155–165. ACM.

Joyal, A. and Moerdijk, I. (1994). A completeness theorem for open maps. *Annals of Pure and Applied Logic*, 70(1):51–86.

Joyal, A., Nielsen, M., and Winskel, G. (1996a). Bisimulation from open maps. *LICS '93 special issue of Information and Computation*, 127(2):164–185. Available as BRICS report, RS-94-7.

Joyal, A., Street, R., and Verity, D. (1996b). Traced monoidal categories. In *Math. Proc. Camb. Phil. Soc.*, volume 119, pages 447–468.

Kahn, G. (1974). The semantics of a simple language for parallel programming. In *Information Processeing*, volume 74, pages 471–475.

Kahn, G. and MacQueen, D. (1977). Coroutines and networks of parallel processes. In Gilchrist, editor, *Proceedings of Information Processing*, pages 993–998. North-Holland.

Katis, P., Sabadini, N., and Walters, R. (1997). Bicategories of processes. *Journal of Pure and Applied Algebra*.

Kok, J. (1987). A fully abstract semantics for dataflow nets. In *Proceedings of Parallel Architectures And Languages Europe*, pages 351–368, Berlin. Springer.

Lynch, N. and Tuttle, M. (1987). Hierarchical correctness proofs for distributed algorithms. Technical report, M.I.T. Laboratory for Computer Science.

Lynch, N. A. and Stark, E. W. (1989). A proof of the kahn principle for input/output automata. *Information and Computation*, 82:81–92.

Mac Lane, S. (1971). *Categories for the Working Mathematician*. Springer.

Mac Lane, S. and Moerdijk, I. (1992). *Sheaves in Geometry and Logic: A First Introduction to Topos Theory*. Springer.

Panangaden, P. (1995). The expressive power of indeterminate primitives in asynchronous computations. Technical Report SOCS-95.8, School of Computer Science, McGill.

Panangaden, P. and Shanbhogue, V. (1992). The expressive power of indeterminate dataflow primitives. *Information and Computation*, 98(1):99–131.

Panangaden, P. and Stark, E. W. (1988). Computations, residuals and the power of indeterminacy. In *Proc. of the 15th ICALP*, pages 439–454. Springer.

Rabinovich, A. and Trakhtenbrot, B. A. (1988). Nets of processes and dataflow. In *Proceedings of ReX School on Linear Time, Branching Time and Partial Order in Logics and Models for Concurrency*, volume 354 of *LNCS*, pages 574–602. Springer.

Rabinovich, A. and Trakhtenbrot, B. A. (1989). Nets and data flow interpreters. In *Proceedings of the 4th LICS*, pages 164–174.

Rabinovich, A. and Trakhtenbrot, B. A. (1990). Communication among relations. In Paterson, M. S., editor, *Proc. of the 7th ICALP*, volume 443 of *LNCS*, pages 294–307. Springer.

Selinger, P. (1997). First-order axioms for asynchrony. In Mazurkiewicz, A. and Winkowski, J., editors, *CONCUR'97*, volume 1243 of *LNCS*, pages 376–390. Springer.

Selinger, P. (1999). Categorical structure of asynchrony. *Electronic Notes in Theoretical Computer Science*, 20.

Staples, J. and Nguyen, V. L. (1985). A fixpoint semantics for nondeterministic data flow. *Journal of the ACM*, 32(2):411–444.

Stark, E. (1998). Fibrational semantics of dataflow networks. In *Proceedings of CONCUR '98*, LNCS. Springer.

Stark, E. W. (1989a). Compositional relational semantics for indeterminate dataflow networks. In *CTCS*, volume 389 of *LNCS*, pages 52–74, Manchester, U.K. Springer.

Stark, E. W. (1989b). Dataflow networks are fibrations. Technical report, Dept. of Computer Science, Stony Brook.

Stark, E. W. (1992). A calculus of dataflow networks. In *LICS '92, Proceedings of the 7th Annual Symposium on Logic in Computer Science*, pages 125–136. IEEE Computer Society Press.

Street, R. (1980). Fibrations in bicategories. *Cahiers de topologie et gomtrie diffrentielle*, XXI(2 - 2^{e} trimestre):111–160.

Street, R. (2003). Functorial calculus in monoidal bicategories. *Applied Categorical Structures*. to appear.

Winskel, G. and Nielsen, M. (1995). *Handbook of Logic in Computer Science*, volume IV, chapter Models for concurrency, pages 1–148. OUP.

Winskel, G. and Nielsen, M. (1996). Presheaves as transition systems. In Peled, D., Pratt, V., and Holzmann, G., editors, *POMIV'96*, volume 29 of *DIMACS*, pages 129–140. AMS.