Fibrational Semantics of Dataflow Networks*

Eugene W. Stark

Department of Computer Science, State University of New York at Stony Brook,
Stony Brook, NY 11794 USA

Abstract. Beginning with the category **Dom** of Scott domains and continuous maps, we introduce a syntax for dataflow networks as "systems of inequalities," and provide an associated operational semantics. We observe that, under this semantics, a system of inequalities determines a two-sided fibration in **Dom**. This leads to the introduction of a certain class of cartesian arrows of spans as a notion of morphism for systems. The resulting structure **Sys**, consisting of domains, systems, and morphisms, forms a bicategory that embeds **Dom** up to equivalence and is suitable as a semantic model for nondeterministic networks. Isomorphism in **Sys** amounts to a notion of system equivalence "up to deterministic internal computations."

1 Introduction

Since the seminal paper of Kahn [Kah74], it has been known that networks of concurrently and asynchronously executing deterministic processes, communicating with each other by sending data values over unbounded FIFO communication channels, admit a simple and elegant semantics in which the function computed by a network of processes is determined via a least fixed point construction from the functions computed by the component processes. However, when one introduces the possibility that processes may make nondeterministic choices, the way in which to generalize this structure, so as to preserve best its spirit, is not immediately evident. In spite of the large number of papers (see *e.g.* [Bc94, JK89, Mis89] for pointers to earlier references) that have been published on the subject of nondeterministic dataflow networks, in the opinion of this author, we still cannot say that we have a fundamental understanding of the algebra of such networks and the precise way in which this algebra extends or generalizes the deterministic case.

The main thrust of the present paper is to explore whether a continuous function semantics for deterministic dataflow networks can in a certain sense be completed to yield a semantics for nondeterministic networks, so that the new semantics embeds the original one via an embedding that preserves important structure of the deterministic case. More precisely, we ask whether there is a way of embedding the locally posetal bicategory [Bén67] **Dom** of Scott domains and continuous maps, into a larger bicategory **Sys** whose 1-cells (arrows) can serve

* This research was supported in part by NSF Grant CCR-9320846.

as interpretations for nondeterministic dataflow networks, via a homomorphism of bicategories **Dom** $\rightarrow$ **Sys** that respects the network-forming operations of series composition, parallel composition, and feedback. The main result of the paper is that this can be done, resulting in a bicategory **Sys** that is in a sense equivalent to a certain bicategory of two-sided fibrations in **Dom**, with cartesian arrows of spans as 2-cells [Gra66, Str74, Str80].

In more explicit detail, our construction starts with **Dom** and produces a new structure **Sys**, which has as its objects the objects of **Dom** (*i.e.* the Scott domains), as its 1-cells (arrows) certain *systems of inequalities*, which are syntactic objects denoting nondeterministic networks, and as its 2-cells suitable *morphisms* of systems of inequalities. There is, in addition, an important third dimension to **Sys**: for given systems (1-cells) $\mathcal{S}$ and $\mathcal{S}'$, the collection of morphisms (2-cells) from $\mathcal{S}$ to $\mathcal{S}'$ is a dCPO, in which the ordering relationships (the 3-cells of **Sys**) reflect information about the progress of computation, including nondeterministic choice. The structure **Sys** embeds **Dom**, not just in the sense that each object of **Dom** corresponds to an object of **Sys** and each arrow of **Dom** to a 1-cell of **Sys**, but also in the stronger sense that ordering relationships between arrows in **Dom** manifest themselves as unique 2-cells between the corresponding 1-cells of **Sys**, so that the embedding of **Dom** into **Sys** becomes a homomorphism of bicategories [Bén67].

The relationship between **Sys** and **Dom** is analogous to, but more complicated than, the relationship between the category **Set** of sets and functions and the bicategory **Rel** of sets, binary relations between sets (1-cells), with inclusion relationships between binary relations as (2-cells), or more generally, the relationship between a regular category **C** and the bicategory of relations in **C** [CKS84]. The homomorphism from **Dom** into **Sys** is analogous to the homomorphism of bicategories that takes each function between sets to the relation (1-cell of **Rel**) that is its graph. The bicategory **Sys** is generated by **Dom**, in analogy to the way in which **Rel** is generated by **Set**. Though analogous, the relationship between **Dom** and **Sys** is of necessity more complicated than the case of **Set** and **Rel**, because, as is well known [BA81], ordinary binary relations do not support a denotational semantics for nondeterministic dataflow networks that gives results in agreement with the intuitively correct operational semantics for such networks. More generally, we observe that discrete fibrations are also inadequate for such a semantics. However, the bicategory **Sys** *can* be regarded as a bicategory of "generalized relations" between domains, if we expand our concept of relation to include the possibility that a single input value a and output value b can be related in more than one way, and in addition the set of all ways of relating a and b may have some additional structure (*e.g.* that of a Scott domain).

Finally, we come to the relationship with fibrations. We show that, for each pair of domains A and B, the ordered category $\mathbf{Sys}(A, B)$ is equivalent to a full subcategory of the category of two-sided *fibrations* [Str74] from A to B in **Dom**, with cartesian arrows of spans as morphisms. We use the term *systemic fibration* to refer to fibrations that correspond to systems, and we obtain a characterization of the systemic fibrations. We can then show that the syntactic operation of series composition of systems of inequalities corresponds to the classical *fibrational composite*, or "tensor product of bimodules" [Str74, Str80] of the corresponding systemic fibrations, and that parallel composition of systems corresponds simply to a cartesian product of fibrations. Thus, the entire structure **Sys** can be regarded as equivalent to the structure having domains as objects, systemic fibrations as 1-cells, cartesian arrows of spans as 2-cells, extensional ordering relationships between cartesian arrows of spans as 3-cells, in which 1-cells are composed (vertically) by fibrational composite and horizontally by cartesian product, and "comma objects" [Str74] serve as the identities for the vertical composition. We are also able to show that feedback of systems of inequalities can be characterized abstractly as a construction on systemic fibrations, though this result is outside the scope of the present paper.

The results of this paper are a continuation of the author's previous work [Sta89a, Sta89b, Sta90, Sta91], in which the relationship between dataflow networks and fibrations was observed. The main new contributions of the present paper are the formulation of the syntactic notion of systems of inequalities, and the identification of a suitable notion of morphism for such systems, together with a related notion of "deterministic equivalence" of systems. The latter is what permits us to construct a bicategory **Sys** that is "sufficiently abstract" to embed **Dom** up to equivalence. These results seem to justify the appropriateness of a systematic study of bicategory of systemic fibrations in **Dom** as a model of nondeterministic dataflow networks. In particular, a characterization of this bicategory along the lines of those given in [CKS84] for "bicategories of spans" and "bicategories of relations" would be interesting, as would a kind of axiomatic description like that used by [CW87]. The result of such investigations would be an understanding of how best to strengthen and improve systems for reasoning about dataflow networks, such as those presented in [Bc94, Sta92, Mis89].

Due to space limitations, we have omitted all proofs, and have provided sketches of proofs only only when they serve to explain critical ideas.

2 Systems of Inequalities

We begin with the category **Dom** of Scott domains, (countably algebraic, bounded-complete, directed-complete partial orders), with continuous maps as morphisms. The hom-sets of this category are again domains under the extensional ordering, which we denote by $\sqsubseteq$, and composition of morphisms in **Dom** is monotone and continuous with respect to this ordering. This structure can be summarized by the statement that **Dom** is a "category enriched in **Dom**," or a "**Dom**-category" [Kel82].

By an *inequality* over **Dom**, we mean an expression of the form $f\mathbf{v} \sqsubseteq g\mathbf{u}$, where $\mathbf{v}$ and $\mathbf{u}$ are formal *variables*, with $\mathbf{u}$ called the *independent variable* and $\mathbf{v}$ the *dependent variable*, and where $f : B \to C$ and $g : A \to C$ are arrows of **Dom**. We say that variables $\mathbf{u}$ and $\mathbf{v}$ have *sorts* A and B, respectively, in the above inequalities. An inequality is called *covariant* if the map f is an identity, so that the inequality may be written in the abbreviated form $\mathbf{v} \sqsubseteq g\mathbf{u}$. Similarly, an inequality is called *contravariant* if the map g is an identity.

A *system of inequalities* over **Dom** consists of a finite set $\mathcal{S}$ of inequalities, together with a partitioning of the set of variables appearing in the inequalities into *covariant* and *contravariant* variables, such that the following conditions are satisfied:

1. Every inequality in $\mathcal{S}$ is either covariant or contravariant.
2. A covariant variable is only permitted to appear as the dependent variable in a covariant inequality, and as the independent variable in a contravariant inequality. In addition, a covariant variable may have at most one dependent occurrence in $\mathcal{S}$.
3. A contravariant variable is only permitted to appear as the dependent variable in a contravariant inequality, and as the independent variable in a covariant inequality.
4. There is a unique variable $\mathbf{i}_{\mathcal{S}}$ that has no dependent occurrence in any of the inequalities of $\mathcal{S}$; we call this variable the *input* variable. There is a unique variable $\mathbf{o}_{\mathcal{S}}$ that has no independent occurrence in any of the inequalities of $\mathcal{S}$; we call this variable the *output variable.* The input and output variables are required to be covariant.
5. All occurrences of the same variable $\mathbf{v}$ in $\mathcal{S}$ have the same sort, which we denote by $|\mathbf{v}|_{\mathcal{S}}$.
6. For every contravariant inequality in $\mathcal{S}$ of the form $f\mathbf{v} \sqsubseteq \mathbf{u}$, the map f is required to be *strict* ($f\bot = \bot$), *additive* ($f(b \sqcup b') = fb \sqcup fb'$, whenever $b \sqcup b'$ exists), and has the following *accessibility* property: for all $b \in |\mathbf{v}|_{\mathcal{S}}$, and for every chain $a_0 \sqsubseteq a_1 \sqsubseteq a_2 \sqsubseteq \ldots$ in $|\mathbf{u}|_{\mathcal{S}}$ with $\sqcup_i a_i = fb$, there exists a chain $b_0 \sqsubseteq b_1 \sqsubseteq b_2 \sqsubseteq \ldots$ in $|\mathbf{v}|_{\mathcal{S}}$ such that $fb_i \sqsubseteq a_i$ for all i and such that $\sqcup_i b_i = b$.

When the system $\mathcal{S}$ is clear from the context, we abbreviate $\mathbf{i}_{\mathcal{S}}$ and $\mathbf{o}_{\mathcal{S}}$ as $\mathbf{i}$ and $\mathbf{o}$, respectively, and we drop the subscript $\mathcal{S}$ from $|\mathbf{v}|_{\mathcal{S}}$. The sort $|\mathbf{i}|$ of the distinguished input variable is called the *input sort* of the system $\mathcal{S}$, the sort $|\mathbf{o}|$ is called the *output sort* of $\mathcal{S}$, and we say that $\mathcal{S}$ is a system *from* $|\mathbf{i}|$ *to* $|\mathbf{o}|$.

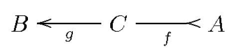
The role of the technical condition (6) will be (*cf.* Proposition 1) to ensure that the "comma poset" $\{(a, b) : fb \sqsubseteq a\}$ is a domain, which embeds "nicely" as a subdomain of the product domain $A \times B$. The embedding will be such that the operational notion "reachability by a computation sequence" exactly coincides with the componentwise ordering on pairs (a, b), and such that every pair (a, b) with $fb \sqsubseteq a$ is reachable from $(\bot, \bot)$ by a computation sequence. These relationships are necessary to maintain a tight correspondence between operational and denotational semantics.

It will often be convenient to describe particular systems of inequalities, or construction on such systems, using a graphical notation. In this notation, a system is represented as a directed graph, with two types of arrows, *covariant*, which we represent by $B \xleftarrow{g} A$, and *contravariant*, which we represent by $B \xrightarrow{f} A$. The nodes of the graph are labeled by objects of **Dom**, and the edges are labeled by arrows of **Dom**, in such a way that if there is a g-labeled covariant edge from a node labeled by A to a node labeled by B, then $g : A \to B$ in **Dom**, and if there is an f-labeled contravariant edge from a node labeled by A to a node labeled by B, then $f : B \to A$ in **Dom**. Each node in the graph corresponds to a distinct formal variable, each covariant edge to a distinct covariant inequality in the system, and each contravariant edge to a contravariant inequality in the system.

Using our graphical notation, we now describe some particular systems of inequalities, and constructions on such systems, which will be of interest to us.

Basic Systems

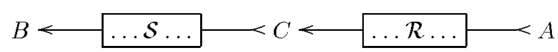
If $f : C \to A$ and $g : C \to B$, then the *basic system* determined by f and g is the three-variable, two-inequality system from A to B described by the graph:



A basic system in which the map f is an identity is called a *basic covariant system*. Similarly, a basic system in which the map g is an identity is called a *basic contravariant system*. The basic systems in which both f and g are identities play a special role. We call them *buffers*.

Series Composition

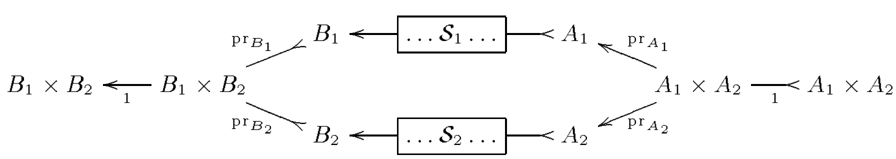
If $\mathcal{R}$ is a system from A to C, and $\mathcal{S}$ is a system from C to B, then the *series composition* of $\mathcal{R}$ and $\mathcal{S}$ is the system $\mathcal{S} \cdot \mathcal{R}$ described by the graph:



where it is understood that the labeled boxes stand for the graphs of $\mathcal{R}$ and $\mathcal{S}$.

Parallel Composition

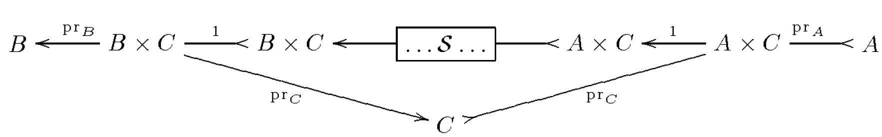
If $\mathcal{S}_1$ is a system from A_1 to B_1, and $\mathcal{S}_2$ is a system from A_2 to B_2, then the *parallel composition* of $\mathcal{S}_1$ and $\mathcal{S}_2$ is the system $\mathcal{S}_1 \times \mathcal{S}_2$ described by the graph:



where the pr_X denote the evident projections. Note that "buffers" have been inserted to satisfy the requirement that the input and output variables of a system be covariant.

Feedback

If $\mathcal{S}$ is a system from $A \times C$ to $B \times C$, then the *feedback* of $\mathcal{S}$ by C is the system $\mathcal{S}_{\star C}$ from A to B described by the graph:



The *regular* (resp. *regular covariant*) systems of inequalities are those that can be constructed from basic (resp. basic covariant) systems using the operations of series composition, parallel composition, and feedback.

The idea that systems of inequalities could be used to describe nondeterministic dataflow networks was first proposed by Misra [Mis89]. The operational semantics we give in the next section is essentially a refinement and elaboration of Misra's "smooth solution" idea.

3 Operational Semantics

An *assignment* for a system of inequalities $\mathcal{S}$ consists of a collection of elements $q_{\mathbf{v}} \in |\mathbf{v}|$, one for each variable $\mathbf{v}$ appearing in the inequalities in $\mathcal{S}$. The set of all assignments for $\mathcal{S}$, with the componentwise ordering, is a Scott domain, which we denote by $\mathrm{Asgt}_{\mathcal{S}}$. Note that $\mathrm{Asgt}_{\mathcal{S}}$ is simply the cartesian product of the sorts of all variables in $\mathcal{S}$. As a special role will be played by the mappings that take an assignment q of $\mathcal{S}$ to the value $q_{\mathbf{i}}$ of the input variable and the $q_{\mathbf{o}}$ of the output variable, it will be convenient to use the symbols $\mathbf{i}$ and $\mathbf{o}$ to denote these mappings. Thus, $\mathbf{i} : \mathrm{Asgt}_{\mathcal{S}} \to A$ and $\mathbf{o} : \mathrm{Asgt}_{\mathcal{S}} \to B$, and we write $\mathbf{i}q$ for $q_{\mathbf{i}} \in A$ and $\mathbf{o}q$ for $q_{\mathbf{o}} \in B$. It is easy to see that the maps $\mathbf{i}$ and $\mathbf{o}$ are continuous. Thus, the triple $(\mathbf{o}, \mathrm{Asgt}_{\mathcal{S}}, \mathbf{i})$ is a *span* from A to B in the category **Dom**:

$$B \xleftarrow{\ \mathbf{o}\ } \mathrm{Asgt}_{\mathcal{S}} \xrightarrow{\ \mathbf{i}\ } A$$

In general, a *span* in a category consists of an object and two arrows in the configuration shown above.

The system $\mathcal{S}$ is said to be *satisfied* by an assignment q if for each inequality $g\mathbf{v} \sqsubseteq f\mathbf{u}$ of $\mathcal{S}$ the relationship $gq_{\mathbf{v}} \sqsubseteq fq_{\mathbf{u}}$ holds in **Dom**. The satisfying assignments for a system of inequalities are called the *configurations* of the system. Let $\mathrm{Conf}_{\mathcal{S}}$ denote the set of all configurations of system $\mathcal{S}$.

If $\mathcal{S}$ is a system from A to B, then a *transition* of $\mathcal{S}$ is a pair of configurations $q \sqsubseteq r$, such that for all inequalities $f\mathbf{v} \sqsubseteq g\mathbf{u}$ in $\mathcal{S}$, we have $fr_{\mathbf{v}} \sqsubseteq gq_{\mathbf{u}}$. We write $q \Rightarrow r$ to denote a transition from q to r. A transition $q \Rightarrow r$ is called *finitary* if $r = q \sqcup c$ for some compact element c of $\mathrm{Asgt}_{\mathcal{S}}$.

A *finite computation sequence from q to q'* for a system of inequalities $\mathcal{S}$ is a sequence of transitions: $q_0 \Rightarrow q_1 \Rightarrow q_2 \Rightarrow \ldots \Rightarrow q_n$ with $q_0 = q$ and $q_n = q'$. An *infinite computation sequence from q to q'* is a sequence of transitions: $q_0 \Rightarrow q_1 \Rightarrow q_2 \Rightarrow \ldots$ with $q_0 = q$ and $\sqcup_i q_i = q'$. A finite or infinite computation sequence is called *finitary* if each transition $q_i \Rightarrow q_{i+1}$ is a finitary transition. We say that a configuration q' is *finitarily reachable from q* if there exists a finite, finitary computation sequence from q to q', *reachable from q* if there exists a finite computation sequence from q to q', *ultimately finitarily reachable from q* if there exists an infinite finitary computation sequence from q to q', and *ultimately reachable from q* if there exists an infinite computation sequence from q to q'.

A configuration of a system $\mathcal{S}$ is called a *finitary state* if it is finitarily reachable from $\bot$, and it is called a *state* if it is ultimately finitarily reachable from

$\perp$. We use $\mathrm{State}_{\mathcal{S}}$ to denote the set of states of system $\mathcal{S}$, ordered by ultimate reachability, and we shall refer to intervals in $\mathrm{State}_{\mathcal{S}}$ as *computations.*

The following result says that ultimate reachability between states of a system of inequalities coincides with the extensional ordering on assignments. Thus, the computations of a system $\mathcal{S}$ already "live" within the domain $\mathrm{Asgt}_{\mathcal{S}}$ of assignments for $\mathcal{S}$. The result also says that "ultimate finitary reachability from $\perp$" coincides with the simpler notion "ultimate reachability from $\perp$." The proof of this result depends crucially on the technical condition (6) in the definition of a system of inequalities.

Proposition 1. *Let $\mathcal{S}$ be a system of inequalities. Then a configuration q of $\mathcal{S}$ is a state if and only if it is ultimately reachable from $\perp$. The set* $\mathrm{State}_{\mathcal{S}}$ *of states, ordered by ultimate reachability, is a normal subdomain ([GS90], p. 642) of* $\mathrm{Asgt}_{\mathcal{S}}$*, the compact elements of which are precisely the finitary states. Moreover,* $\mathrm{State}_{\mathcal{S}}$ *is* transition-closed *in* $\mathrm{Asgt}_{\mathcal{S}}$*, in the sense that whenever $q \in \mathrm{State}_{\mathcal{S}}$ and $q \Rightarrow r$ is a transition, then $r \in \mathrm{State}_{\mathcal{S}}$ as well.*

A state q of a system $\mathcal{S}$ is called *completed* if whenever q' is a state of $\mathcal{S}$ such that $q \sqsubseteq q'$ and $\mathbf{i}q = \mathbf{i}q'$, then $q = q'$. Intuitively, completed states represent states in which all computation that is enabled by the available input has already occurred. The *input/output relation* of a system $\mathcal{S}$ from A to B is defined to be the set $R_{\mathcal{S}}$ of all pairs $(\mathbf{i}q, \mathbf{o}q)$, such that q is a state of $\mathcal{S}$. The *completed input/output relation* of $\mathcal{S}$ is the set $\overline{R}_{\mathcal{S}}$ of all pairs $(\mathbf{i}q, \mathbf{o}q)$ such that q is a completed state of $\mathcal{S}$. Both the input/output relation and the completed input/output relation give basic information about the input/output behavior of a system of inequalities. The input/output relation simply gives the set of input/output pairs that can be observed as the results of computations from $\perp$. The completed input output relation gives only those pairs corresponding to computations that are "completed" in the sense of having made "maximal progress," given the available input.

To illustrate the expressive power of the systems of inequalities model, we now consider briefly an example, originally given by Misra [Mis89]. Let V be a set of *data values*, which we assume contains at the two distinct elements 0 and 1. Let V^∞ denote the domain of finite and infinite sequences of elements of V, with the prefix ordering. Let $V + V = (V \times \{0\}) \cup (V \times \{1\})$ be the disjoint union of two copies of V, and let $(V+V)^\infty$ be the domain of finite and infinite sequences of elements of $V + V$.

Let the maps

$$\mathrm{proj}_0,\ \ \mathrm{proj}_1 : (V+V)^\infty \to V^\infty$$

be the projection maps that take a sequence of elements of $V+V$ and extract the subsequences of values in the left summand and the right summand, respectively. Define

$$\mathrm{deal} = \langle \mathrm{proj}_0, \mathrm{proj}_1 \rangle : (V+V)^\infty \to V^\infty \times V^\infty \ \ ;$$

that is, deal is a map that distributes a "tagged" input sequence onto two untagged output sequences, using the tags to determine the destination of each value in the input sequence. Let strip : $(V+V)^\infty \to V^\infty$ be the map that "strips tags" from its input sequence. Now, consider the system:

$$V^\infty \xleftarrow[\text{strip}]{} (V+V)^\infty \xrightarrow[\text{deal}]{} V^\infty \times V^\infty$$

By examining the possible computation sequences for this system, one sees that it nondeterministically performs a "tagged merge" of two input sequences, then strips the tags before outputting the resulting sequence. This nondeterministic system corresponds to what has generally been called "angelic merge" in the literature on dataflow networks (though [Mis89] calls it "fair merge"). See [PS92, PS88] for further discussion on various types of nondeterministic merging that have been considered.

4 Properties of the State Space

The state space $\text{State}_{\mathcal{S}}$ of a system of inequalities $\mathcal{S}$ has a number of special properties, which we explore in this section. Our objective here is not just to make a list of properties that are simple consequences of properties of partial orders, but rather to forge a connection between the concrete, syntactic systems of inequalities model and its associated operational semantics on the one hand, and the abstract theory of fibrations on the other hand. This development leads directly to (1) the identification of an appropriate notion of morphism of systems, so that domains, systems, and morphisms become a bicategory **Sys**, and isomorphism in **Sys** turns out to be an appropriate and useful notion of system equivalence; (2) the characterization of **Sys** up to equivalence as a bicategory of fibrations in **Dom**, with corresponding characterizations of the operations of sequential composition, parallel composition, and feedback. Ultimately, we hope to achieve an axiomatization of **Sys** as a "bicategory of fibrations," within which reasoning about dataflow networks could be carried out categorically.

Proposition 2. *Suppose S is a system from A to B. Then the maps $\mathbf{i} : \text{State}_{\mathcal{S}} \to A$ and $\mathbf{o} : \text{State}_{\mathcal{S}} \to B$ are strict, additive, and accessible.*

Proposition 3. *Suppose $\mathcal{S}$ is a system from A to B. If q is a state of $\mathcal{S}$, then:*

1. *For all $a \sqsupseteq \mathbf{i}q$, there exists a least state $q \sqcup a$ of S such that $a \sqsubseteq \mathbf{i}(q \sqcup a)$ and such that $q \sqsubseteq q \sqcup a$. Moreover, $\mathbf{i}(q \sqcup a) = a$.*
2. *For all $b \sqsubseteq \mathbf{o}q$, there exists a greatest state $b \sqcap q$ of S such that $\mathbf{o}(b \sqcap q) \sqsubseteq b$ and such that $b \sqcap q \sqsubseteq q$. Moreover, $\mathbf{o}(b \sqcap q) = b$.*
3. *For all $a \sqsupseteq \mathbf{i}q$ and all $b \sqsubseteq \mathbf{o}q$ we have $b \sqcap (q \sqcup a) = (b \sqcap q) \sqcup a$.*

A partially ordered set S, equipped with monotone maps $\mathbf{i} : S \to A$ and $\mathbf{o} : S \to B$ having the properties stated in Proposition 3 above is called a (two-sided) *fibration* from A to B [Gra66, Str74]. We can thus restate Proposition 3 as follows: If $\mathcal{S}$ is a system from A to B, then the span:

$$B \xleftarrow{\;\mathbf{o}\;} \text{State}_{\mathcal{S}} \xrightarrow{\;\mathbf{i}\;} A$$

is a fibration from A to B in the 2-category of posets and monotone maps.

A fibration can be thought of as a kind of generalized monotone relation, which allows a particular input/output pair (a, b) to be related in more than one way. The set $\{q \in \mathrm{State}_{\mathcal{S}} : \mathbf{o}q = b, \mathbf{i}q = a\}$ of different ways in which a particular pair (a, b) can be related is called the *fiber over b and a*. The sense in which fibrations are "monotone" is that an interval $a \sqsubseteq a'$ in A, representing an increase in input, and an interval $b' \sqsubseteq b$ in B, representing a decrease in output, induce a transformation from the "ways of relating a and b" (the fiber over a and b) to the "ways of relating a' and b'" (the fiber over a' and b'). Specifically, this transformation takes q to $b' \sqcap q \sqcup a'$.

An interesting special case of fibrations occurs when each of the fibers is a discrete partial order (*i.e.* a set); these are called *discrete* fibrations. A discrete fibration corresponds to a kind of generalized input/output relation for which each given pair (a, b) can be related in multiple ways, but for which there is no connection between the different ways of relating b and a. The fibrations determined by systems of inequalities are *not* discrete, in general, because in general there will be nontrivial ordering (ultimate reachability) relationships between ways of relating b and a. These ordering relationships are significant from the point of view of computational intuition. States q and q', in the the fiber over a and b, that are incomparable with respect to the ultimate reachability ordering but nevertheless consistent in the sense of having an upper bound within the same fiber, can be thought of as representing situations that could occur in a single concurrent computation. States q and q' that are inconsistent with respect to the ultimate reachability ordering represent situations that reflect two distinct, incompatible resolutions of some nondeterministic choice, and thus do not represent situations that could occur in a single concurrent computation.

If $\mathcal{S}$ is a system from A to B, then there exist "comma posets":

$$\mathbf{i}_{\mathcal{S}}/A = \{(q, a) : \mathbf{i}q \sqsubseteq a\} \qquad B/\mathbf{o}_{\mathcal{S}} = \{(b, q) : b \sqsubseteq \mathbf{o}q\} \ .$$

Proposition 3 gives us monotone maps:

$$\sqcup : \mathbf{i}_{\mathcal{S}}/A \to \mathrm{State}_{\mathcal{S}} : (q, a) \mapsto q \sqcup a \qquad \sqcap : B/\mathbf{o}_{\mathcal{S}} \to \mathrm{State}_{\mathcal{S}} : (b, q) \mapsto b \sqcap q \ .$$

It can be shown that the posets $\mathbf{i}_{\mathcal{S}}/A$ and $B/\mathbf{o}_{\mathcal{S}}$ are in fact domains, and the maps $\sqcup$ and $\sqcap$ are continuous.

In the theory of fibrations [Gra66, Str74], the above maps $\sqcup$ and $\sqcap$ play a special role. We refer to $\sqcup$ as the *input action* and to $\sqcap$ as the *output action* of the fibration associated with the system $\mathcal{S}$. These maps turn out [Str74] to give $\mathrm{State}_{\mathcal{S}}$ (more precisely, the span $(\mathbf{o}, \mathrm{State}_{\mathcal{S}}, \mathbf{i})$) a structure of algebra for two monads that correspond to the constructions of $\mathbf{i}_{\mathcal{S}}/A$ and $B/\mathbf{o}_{\mathcal{S}}$ from $\mathrm{State}_{\mathcal{S}}$ (to be precise, these have to be viewed as constructions on spans from A to B), and in fact the notion of fibration can actually be characterized in terms of the existence of such structure maps. Thus, $\mathrm{State}_{\mathcal{S}}$ is not just a fibration in the 2-category of posets and monotone maps, but in fact also in the 2-category **Dom**.

In the context of Proposition 3 above, we call computations of the form $q \sqsubseteq q \sqcup a$ *pure-input* computations, and we call computations of the form $b \sqcap q \sqsubseteq b$ *pure-output* computations. (In the standard terminology associated with fibrations [Gra66], these would be called "opcartesian" and "cartesian" morphisms, respectively.) Computations $q \sqsubseteq q'$ such that $\mathbf{i}q = \mathbf{i}q'$ and $\mathbf{o}q = \mathbf{o}q'$ are called *internal* computations. An interesting and useful result that follows from general considerations is that every interval in $\mathrm{State}_{\mathcal{S}}$ has a unique factorization as a pure-input computation, followed by an internal computation, followed by a pure-output computation.

The following result does not hold for fibrations in general, but does hold for fibrations derived from systems of inequalities, due to the syntactic restrictions we have placed on the occurrence of the output variable in such systems.

Proposition 4. *Suppose $\mathcal{S}$ is a system of inequalities from A to B. Then the output action $\sqcap : B/\mathbf{o}_{\mathcal{S}} \to \mathrm{State}_{\mathcal{S}}$ has a right adjoint:*

$$\sqcap^* : \mathrm{State}_{\mathcal{S}} \to B/\mathbf{o}_{\mathcal{S}} : q \mapsto (\mathbf{o}q, \overline{q})$$

with identity counit, where $\overline{q}$ is obtained from q by increasing the value of the output variable until the unique inequality having the output variable as a dependent variable is satisfied exactly.

The above result implies the existence, for any state q, of a largest state $\overline{q}$ for which there exists a pure-output computation $q \sqsubseteq q'$. We call the state $\overline{q}$ in the previous result the *pure-output completion* of state q. This property turns out to be equivalent to the statement that systems of inequalities are "output buffered," in the sense of being equivalent to their series compositions on the output side with a buffer. This property is an important characteristic of dataflow networks.

5 Deterministic Computations

We are not interested in distinctions between systems arising solely from certain details of internal computation. For example, even though the systems

$$B \xleftarrow[g]{} C \xrightarrow[1_C]{} C \xleftarrow[g']{} A \xrightarrow[1_A]{} A \qquad\qquad B \xleftarrow[gg']{} A \xrightarrow[1_A]{} A$$

do not have state spaces that are isomorphic as spans, the difference between the two has only to do with the fact that the former has internal variables whose values in a sense depend functionally on the input variable. This difference is uninteresting from the point of view of input/output behavior. We wish to define and investigate a notion of system equivalence that ignores this type of distinction between systems. Our basic approach in addressing these issues is to characterize a class of internal computations of systems that we regard as "uninteresting", and then to define a notion of isomorphism of systems by essentially arranging for all such uninteresting computations to be mapped to identities. The most obvious candidate class of "uninteresting internal computations" is the class of *all* internal computations. In a formal mathematical sense, it is indeed possible to factor the set of states of a system into classes "connected by internal computation." In fibrational terms, the result of this construction would be a *discrete* fibration, because all the fibers would be reduced to unstructured sets. Since the theory of discrete fibrations is well-developed, this idea, of trying to "discretize" the fibrations derived from systems of inequalities, is very tempting. However, from a computational point of view, it is wrong, as the necessary quotienting construction fails to distinguish systems having distinct completed input/output relations. In particular, if

$$\mathrm{pr}_1 : V^\infty \times V^\infty \to V^\infty : (a, a') \mapsto a \qquad \sqcap : V^\infty \times V^\infty \to V^\infty : (a, a') \mapsto a \sqcap a'$$

then the systems

$$V^\infty \xleftarrow[\sqcap]{} V^\infty \times V^\infty \xrightarrow[\mathrm{pr}_1]{} V^\infty \qquad\qquad V^\infty \xleftarrow[1]{} V^\infty \xrightarrow[1]{} V^\infty$$

have distinct completed input/output relations, but determine the same discrete fibration. Essential information about the completed input/output relation of a system is thus lost if we completely ignore internal computations.

So, if we are to ensure that completed input/output relations are respected by system equivalence, we must stop short of identifying *all* states that are connected by internal computations. In light of this fact, it is appropriate to look for a suitable class of uninteresting internal computations that is smaller than the full class of internal computations. A class of internal computations with particularly nice properties is the class of "deterministic internal computations," which we now define and investigate.

Formally, suppose $\mathcal{S}$ is a system of inequalities from A to B. An endomorphism $\rho : \mathrm{State}_{\mathcal{S}} \to \mathrm{State}_{\mathcal{S}}$ in **Dom** is called *increasing* if the inequality $1_A \sqsubseteq \rho$ holds. It is called a *reflexive* if it is increasing and also *idempotent* ($\rho\rho = \rho$). The map ρ is called an *arrow of spans* if it preserves input and output; that is, if $\mathbf{i}\rho = \mathbf{i}$ and $\mathbf{o}\rho = \mathbf{o}$ hold. An internal computation $q \sqsubseteq q'$ of $\mathcal{S}$ is called *deterministic* if there exists an increasing arrow of spans ρ on $\mathrm{State}_{\mathcal{S}}$ such that $\rho q = \rho q'$. Define states q and q' of $\mathcal{S}$ to be *deterministically equivalent* if $\rho q = \rho q'$ for some increasing arrow of spans ρ on $\mathrm{State}_{\mathcal{S}}$.

It can be shown that any two *increasing* arrows of spans on $\mathrm{State}_{\mathcal{S}}$ have a least upper bound (by taking the colimit of a "tower" of composites). It follows from this, using directed completeness and continuity, that there always exists a *largest* increasing arrow of spans on $\mathrm{State}_{\mathcal{S}}$, which is necessarily a reflexive.

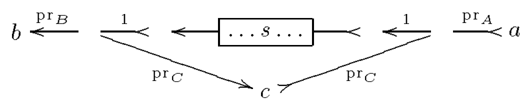
Proposition 5. *Suppose $\mathcal{S}$ is a system of inequalities. Then the following are equivalent statements about an internal computation $q \sqsubseteq q'$ of $\mathcal{S}$:*

1. *$\rho q = \rho q'$, where ρ is the largest increasing arrow of spans on* $\mathrm{State}_{\mathcal{S}}$.
2. *$\rho q = \rho q'$ for some reflexive arrow of spans* $\rho : \mathrm{State}_{\mathcal{S}} \to \mathrm{State}_{\mathcal{S}}$.
3. *$\rho q = \rho q'$, for some increasing arrow of spans* $\rho : \mathrm{State}_{\mathcal{S}} \to \mathrm{State}_{\mathcal{S}}$.

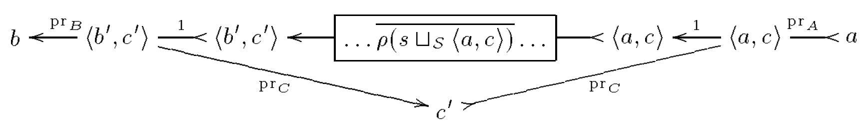
Proposition 5 can be used to establish that the class of deterministic internal computations is the largest class of internal computations that is stable under pushout along arbitrary internal computations. We prefer the abstract definition in terms of reflexives because it lends itself more readily to categorical reasoning.

We can characterize the class of deterministic internal computations of systems of inequalities formed by series composition, parallel composition, and feedback. This is done by characterizing the largest increasing arrows of spans on such systems. The next result gives this characterization in the case of feedback, which is the most interesting case. The cases of series and parallel composition are similar, but simpler, and are omitted.

Proposition 6. *Suppose $\mathcal{S}$ is a system from $A \times C$ to $B \times C$. Let ρ be the largest increasing arrow of spans on* $\mathrm{State}_{\mathcal{S}}$*. Then the largest increasing arrow of spans on* $\mathrm{State}_{\mathcal{S}_{\star C}}$ *is the colimit $\sqcup_i \rho^i$ of the chain $1 \sqsubseteq \rho \sqsubseteq \rho\rho \sqsubseteq \ldots$, where ρ is the map that takes a state of the form*



to the state



where $\langle b', c' \rangle = \mathbf{o}(\overline{\rho(s \sqcup_{\mathcal{S}} \langle a, c \rangle)})$.

In the preceding result, we have extended the use of our graphical notation for systems in an obvious way to serve as a notation for representing assignments for a system.

6 Morphisms of Systems

Suppose $\mathcal{S}$ and $\mathcal{S}'$ are systems of inequalities from A to B. A *weak morphism* from $\mathcal{S}$ to $\mathcal{S}'$ is an arrow of spans: $h : \mathrm{State}_{\mathcal{S}} \to \mathrm{State}_{\mathcal{S}'}$ such that the following conditions are satisfied:

1. h is *input quasi-cartesian*: for all $q \in \mathrm{State}_{\mathcal{S}}$ and all $a \in A$ with $\mathbf{i}q \sqsubseteq a$, the internal computation $(hq) \sqcup a \sqsubseteq h(q \sqcup a)$ of $\mathrm{State}_{\mathcal{S}'}$ is deterministic.
2. h is *output quasi-cartesian*: for all $q \in \mathrm{State}_{\mathcal{S}}$ and all $b \in B$ with $b \sqsubseteq \mathbf{o}q$, the computation $h(b \sqcap q) \sqsubseteq b \sqcap (hq)$ in $\mathrm{State}_{\mathcal{S}'}$ is deterministic.
3. h *preserves deterministic computations*: whenever $q \sqsubseteq r$ is a deterministic computation of $\mathcal{S}$, then $hq \sqsubseteq hr$ is a deterministic computation of $\mathcal{S}'$.

Note that the existence of the computations referred to in conditions (1) and (2) is ensured by the universal properties of $\sqcup$ and $\sqcap$. In conditions (1) and (2) of the above definition, if the computation $(hq) \sqcup a \sqsubseteq h(q \sqcup a)$ (resp. $h(b \sqcap q) \sqsubseteq b \sqcap (hq)$) is an identity, so that $(hq) \sqcup a = h(q \sqcup a)$ (resp. $h(b \sqcap q) = b \sqcap (hq)$), then h is called *input cartesian* (resp. *output cartesian*). We call a map *quasi-cartesian* if it is both input and output quasi-cartesian, and *cartesian* if it is both input and output cartesian.

A *morphism* from $\mathcal{S}$ to $\mathcal{S}'$ is a weak morphism h that satisfies the additional property:

4. h is *deterministically complete*: if $\rho' : \text{State}_{\mathcal{S}'} \to \text{State}_{\mathcal{S}'}$ is the largest increasing arrow of spans on $\text{State}_{\mathcal{S}'}$, then $\rho' h = h$.

Note that in fact property (4) implies property (3).

Proposition 7. *Suppose $\mathcal{S}$ is a system of inequalities from A to B. Then*

1. *Every increasing arrow of spans on* $\text{State}_{\mathcal{S}}$ *is input quasi-cartesian.*
2. *The largest increasing arrow of spans on* $\text{State}_{\mathcal{S}}$ *is output cartesian.*

Though the proof of (1) above is reasonably straightforward, the proof of (2) is nontrivial, and it involves using the right adjoint to the output action $\sqcap_{\mathcal{S}}$ to construct an increasing arrow of spans σ such that if ρ denotes the largest increasing arrow of spans on $\text{State}_{\mathcal{S}}$, then all computations $\sigma(b \sqcap q) \sqsubseteq \sigma(b \sqcap \rho q)$ are identities.

Proposition 8. *Suppose $h : \text{State}_{\mathcal{S}} \to \text{State}_{\mathcal{S}'}$ is a weak morphism. If ρ' denotes the largest increasing arrow of spans on $\text{State}_{\mathcal{S}'}$, then $\rho' h$ is a morphism.*

The next result states that, if we restrict our attention to basic covariant systems, morphisms correspond exactly to the extensional ordering $\sqsubseteq$.

Proposition 9. *Consider the basic covariant system: $\mathcal{S} :\ B \xleftarrow[g]{} A \xrightarrow[1_A]{} A$. If $\mathcal{S}'$ is an arbitrary system from A to B, then there can be at most one morphism $h : \mathcal{S}' \to \mathcal{S}$. Moreover, if $\mathcal{S}'$ is the basic covariant system $B \xleftarrow[g']{} A \xrightarrow[1_A]{} A$, then there is a morphism $h : \mathcal{S}' \to \mathcal{S}$ if and only if $g' \sqsubseteq g$.*

Let $\mathbf{dCPO}_{\perp}$ denote the category of directed-complete posets with least element.

Theorem 1 *The systems of inequalities from A to B are the objects of a $\mathbf{dCPO}_{\perp}$-category $\mathbf{Sys}(A, B)$, whose arrows are the morphisms of systems, with ordinary function composition as composition of morphisms, and with the largest increasing arrow of spans on $\text{State}_{\mathcal{S}}$ as the identity morphism of $\mathcal{S}$. The ordering on homs is the extensional ordering. The full subcategory determined by the* *basic covariant systems is a poset isomorphic to* $\mathbf{Dom}(A, B)$*. The syntactic operations of series composition, parallel composition and feedback of systems of inequalities extend to locally continuous functors:*

$$\begin{aligned} \text{-} \cdot \text{-} &: \mathbf{Sys}(C, B) \times \mathbf{Sys}(A, C) \to \mathbf{Sys}(A, B) \\ \text{-} \times \text{-} &: \mathbf{Sys}(A_1, B_1) \times \mathbf{Sys}(A_2, B_2) \to \mathbf{Sys}(A_1 \times A_2, B_1 \times B_2) \\ (\text{-})_{\star C} &: \mathbf{Sys}(A \times C, B \times C) \to \mathbf{Sys}(A, B) \ . \end{aligned}$$

We call systems $\mathcal{S}$ and $\mathcal{S}'$ from A to B *deterministically equivalent* if they are isomorphic objects in the category $\mathbf{Sys}(A, B)$.

Proposition 10. *If systems* $\mathcal{S}$ *and* $\mathcal{S}'$ *are deterministically equivalent, then they have the same completed input/output relation.*

Proposition 11. *Every system of inequalities* $\mathcal{S}$ *from* A *to* B *is deterministically equivalent to a basic system* $B \xleftarrow[\mathrm{o}_{\mathcal{S}}]{} \mathrm{State}_{\mathcal{S}} \xrightarrow[\mathrm{i}_{\mathcal{S}}]{} A$

Theorem 2 *If* $\mathcal{S}$ *and* $\mathcal{S}'$ *are regular covariant systems from* A *to* B*, then* $\mathcal{S}$ *and* $\mathcal{S}'$ *are deterministically equivalent if and only if they denote the same function under the classical (Kahn) semantics, in which series composition of systems corresponds to function composition, parallel composition of systems to cartesian product of functions, and feedback of systems to the usual least fixed point construction.*

We are now able to organize domains, systems of inequalities, morphisms of systems, and their computations into a single algebraic structure **Sys**, which might be called a "bicategory with homs in $\mathbf{dCPO}_\perp$-**Cat**."

Theorem 3 *The* $\mathbf{dCPO}_\perp$*-categories* $\mathbf{Sys}(A, B)$ *are the homs of a bicategory* **Sys***, where composition is given by the series composition functors*

$$\text{-} \cdot \text{-} : \mathbf{Sys}(C, B) \times \mathbf{Sys}(A, C) \to \mathbf{Sys}(A, B)$$

and for each domain A*, the identity element of* $\mathbf{Sys}(A, A)$ *is the "*A*-buffer":*

$$A \xleftarrow[1_A]{} A \xrightarrow[1_A]{} A \ .$$

Moreover, there is a homomorphism of bicategories $(\text{-})_* : \mathbf{Dom} \to \mathbf{Sys}$ *that takes each function* $g : A \to B$ *to the basic covariant system* g_**:*

$$B \xleftarrow[g]{} A \xrightarrow[1_A]{} A$$

and takes each ordering relationship $g \sqsubseteq g'$ *to the unique morphism:* $g_* \to g'_*$*.*

It is natural to ask what happens if we factor the fibration $\mathrm{State}_{\mathcal{S}}$ associated with a system of inequalities $\mathcal{S}$ by splitting the largest reflexive arrow of spans on $\mathrm{State}_{\mathcal{S}}$. It turns out that we can do this, and the result is again a fibration. Let us call a fibration *systemic* if it results by this splitting construction from a system of inequalities. Systemic fibrations can be characterized as spans in the subcategory of strict, additive and accessible maps in **Dom**, which are in addition fibrations in **Dom** whose output actions have right adjoints with identity counit, and which admit no nontrivial reflexive arrows of spans. A systemic fibration $B \xleftarrow[g]{} S \xrightarrow[f]{} A$ from A to B thus determines a basic system $B \xleftarrow[g]{} S \underset{f}{\longrightarrow\!\!<} A$. Series composition of systems translates under the splitting construction to the classical "fibrational composite" or "tensor product of bimodules" [Str74, Str80]. Morphisms of systems translate to "cartesian arrows of spans," which are the standard notion of morphism for fibrations.

7 Conclusion

We conclude with a very brief comparison with the closely related paper [HPW98], in this same Proceedings, where the use of *profunctors* is proposed as a model for dataflow. Profunctors between posets are equivalent to discrete fibrations between posets, and the composition of profunctors using coends is equivalent to fibrational composite of discrete fibrations. A fibrational version of feedback can also be given corresponding to the "secured coend" characterization in [HPW98]. We have noted in Sect. 5 the inadequacy of discrete fibrations if completed input/output relations are of interest. The paper [HPW98] avoids this and some other important technicalities by treating only finite computations.

In fibrational terms, the "stability" condition in [HPW98] ensures the existence of minimal representatives in each fiber. In this case, the "discretization mapping" taking $\text{State}_{\mathcal{S}}$ to a discrete fibration (*i.e.* profunctor) amounts to splitting the *least coreflexive* arrow of spans, in contrast to the present paper, where we split the *greatest reflexive*. In intuitive terms, splitting the least coreflexive amounts to equating q and q' if we can reach a common state by "computing backward codeterministically" from both q and q'. Splitting the largest reflexive amounts to identifying q and q' if we can reach the same state by "computing forward deterministically." In the presence of stability, every internal computation is codeterministic. The stability condition would seem to be a significant restriction on the expressiveness of the model, though [HPW98] does not make this clear. For example, can any non-stable functions between domains be expressed as stable profunctors?

References

[BA81] J. D. Brock and W. B. Ackerman. Scenarios: A model of non-determinate computation. In *Formalization of Programming Concepts*, volume 107 of *Lecture Notes in Computer Science*, pages 252–259. Springer-Verlag, 1981.

[Bc94] J. A. Bergstra and Gh. Ştefănescu. Network algebra for synchronous and asynchronous dataflow. Technical Report Logic Group Preprint No 122, Utrecht University Department of Philosophy, October 1994.

[Bén67] J. Bénabou. Introduction to bicategories. In *Reports of the Midwest Category Seminar*, volume 47 of *Lecture Notes in Mathematics*, pages 1–77. Springer-Verlag, 1967.

[CKS84] A. Carboni, S. Kasangian, and R. Street. Bicategories of spans and relations. *Journal of Pure and Applied Algebra*, 33:259–267, 1984.

[CW87] A. Carboni and R. F. C. Walters. Cartesian bicategories I. *Journal of Pure and Applied Algebra*, 49:11–32, 1987.

[Gra66] J. W. Gray. Fibred and cofibred categories. In *Proc. Conference on Categorical Algebra at La Jolla*, pages 21–83. Springer-Verlag, 1966.

[GS90] C. Gunter and D. S. Scott. Semantic domains. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science*, volume Volume B: Formal Models and Semantics of *Lecture Notes in Computer Science*, pages 633–674. MIT Press, 1990.

[HPW98] T. Hildebrandt, P. Panangaden, and G. Winskel. A relational model of non-deterministic dataflow. In R. de Simone and D. Sangiorgi, editors, *Proceedings CONCUR 98,* Nice, Lecture Notes in Computer Science. Springer-Verlag, 1998.

[JK89] B. Jonsson and J. N. Kok. Comparing two fully abstract dataflow models. In *Proc. PARLE 89*, volume 365 of *Lecture Notes in Computer Science*, pages 217–234. Springer-Verlag, 1989.

[Kah74] G. Kahn. The semantics of a simple language for parallel programming. In J. L. Rosenfeld, editor, *Information Processing 74*, pages 471–475. North-Holland, 1974.

[Kel82] G. M. Kelly. *Basic Concepts of Enriched Category Theory*. Number 64 in London Mathematical Society Lecture Note Series. Cambridge University Press, 1982.

[Mis89] J. Misra. Equational reasoning about nondeterministic processes (preliminary version). In *ACM Symposium on Principles of Distributed Computing*, pages 29–44, 1989.

[PS88] P. Panangaden and E. W. Stark. Computations, residuals, and the power of indeterminacy. In T. Lepisto and A. Salomaa, editors, *Automata, Languages, and Programming*, volume 317 of *Lecture Notes in Computer Science*, pages 439–454. Springer-Verlag, 1988.

[PS92] Prakash Panangaden and Vasant Shanbhogue. The expressive power of indeterminate dataflow primitives. *Information and Computation*, 98(1):99–131, May 1992.

[Sta89a] E. W. Stark. An algebra of dataflow networks. *Fundamenta Informaticae*, 22:167–186, 1989.

[Sta89b] E. W. Stark. Compositional relational semantics for indeterminate dataflow networks. In *Category Theory and Computer Science*, volume 389 of *Lecture Notes in Computer Science*, pages 52–74. Springer-Verlag, Manchester, U. K., 1989.

[Sta90] E. W. Stark. A simple generalization of Kahn's principle to indeterminate dataflow networks. In M. Z. Kwiatkowska, M. W. Shields, and R. M. Thomas, editors, *Semantics for Concurrency, Leicester 1990*, pages 157–176. Springer-Verlag, 1990.

[Sta91] E. W. Stark. Dataflow networks are fibrations. In *Category Theory and Computer Science*, volume 530 of *Lecture Notes in Computer Science*, pages 261–281. Springer-Verlag, Paris, France, 1991.

[Sta92] E. W. Stark. A calculus of dataflow networks. In *Logic in Computer Science*, pages 125–136. IEEE Computer Society Press, 1992.

[Str74] R. H. Street. Fibrations and Yoneda's lemma in a 2-category. In *Lecture Notes in Mathematics 420*, pages 104–133. Springer-Verlag, 1974.

[Str80] R. H. Street. Fibrations in bicategories. *Cahier de Topologie et Géometrie Différentielle*, XXI-2:111–159, 1980.

This article was processed using the LaTeX macro package with LLNCS style