Titre: Title:	On the Scott-continuity of tagged signal processes
Auteurs: Authors:	Laurent Moss, & Guy Bois
Date:	2009
Type:	Rapport / Report
Référence: Citation:	Moss, L., & Bois, G. (2009). On the Scott-continuity of tagged signal processes (Technical Report n° EPM-RT-2009-01). https://publications.polymtl.ca/2640/

URL de PolyPublie: PolyPublie URL:	https://publications.polymtl.ca/2640/
Version:	Version officielle de l'éditeur / Published version
Conditions d'utilisation: Terms of Use:	Tous droits réservés

Institution:	École Polytechnique de Montréal
Numéro de rapport: Report number:	EPM-RT-2009-01
URL officiel: Official URL:	
Mention légale: Legal notice:	

EPM–RT–2009-01

ON THE SCOTT-CONTINUITY OF TAGGED SIGNAL PROCESES

Laurent Moss et Guy Bois
Département de Génie informatique et génie logiciel
École Polytechnique de Montréal

Avril 2009

Poly

ÉCOLE
POLYTECHNIQUE
MONTRÉAL

EPM-RT-2009-01

On the Scott-continuity of tagged signal processes

Laurent Moss et Guy Bois
Département de génie informatique et génie logiciel
École Polytechnique de Montréal

Avril 2009

Dépôt légal :
Bibliothèque nationale du Québec, 2009
Bibliothèque nationale du Canada, 2009

EPM-RT-2009-01
On the Scott-continuity of tagged signal processes
par : **Laurent Moss, Guy Bois**
Département de génie informatique et génie logiciel,
École Polytechnique de Montréal

École Polytechnique de Montréal
Bibliothèque – Service de fourniture de documents
Case postale 6079, Succursale «Centre-Ville»
Montréal (Québec)
Canada H3C 3A7

Téléphone :	(514) 340-4846
Télécopie :	(514) 340-4026
Courrier électronique :	biblio.sfd@courriel.polymtl.ca

On the Scott-continuity of tagged signal processes

Laurent Moss[a], Guy Bois[a]

[a] *Department of Computer and Software Engineering, École Polytechnique de Montréal, P.O. Box 6079 Centre-Ville, Montréal, Québec, Canada H3C 3A7*

Abstract

Process networks are frequently used to model signal processing and multimedia applications. It is important to ensure that a process network has a uniquely defined behavior in order to correctly model such deterministic systems. Furthermore, a constructive procedure to determine this unique behavior is necessary for its simulation and execution.

By the Kahn principle, the unique behavior of a process network is known to be the least fixed point of the network functional when every process computes a Scott-continuous function. The Kahn principle is used in a recent timed extension of the least fixed point semantics of Kahn process networks, using the tagged signal model. In this extension, processes compute a function from input signals to output signals, where a signal is defined as a partial function from a down set of tags to a set of values. However, it is often tedious to formally prove that a tagged signal process is Scott-continuous.

This paper presents theorems on Scott-continuity that simplify such proofs. Thus, a general theorem on the necessary and sufficient conditions for the Scott-continuity of tagged signal processes is proven. Scott-continuity is then proven for broad classes of processes, namely the classes of exactly causal processes and of domain-warping processes, which include stateless processes, delays as well as a subset of discrete-event processes.

Key words: Process networks, Denotational semantics, Timed systems, Discrete events, Dataflow

Email addresses: `moss@grm.polymtl.ca` (Laurent Moss), `bois@grm.polymtl.ca` (Guy Bois)

1. Introduction

Process networks, which are composed of several concurrently executing processes that communicate asynchronously only through point-to-point FIFO channels, are frequently used to model signal processing and multimedia applications. It is important to ensure that a process network has a uniquely defined behavior in order to correctly model such deterministic systems. Furthermore, a constructive procedure to determine this unique behavior is necessary for its simulation and execution.

It has been shown that the existence and uniqueness of behaviors of process networks can be guaranteed when every process is a Scott-continuous function from the history of its inputs to the history of its outputs. By the Kahn principle, the function computed by the process network is then the least fixed point of the network functional. The existence of a least fixed point is guaranteed by the Kleene fixed-point theorem [1, 2] and, because it is a constructive theorem, the unique behavior can be obtained by the iterative application of the network functional.

The Kahn principle has first been used in Kahn process networks [2], which is an untimed model of computation. Recently, Liu and Lee have developed a timed extension of the least fixed point semantics of Kahn process networks [5, 6], using the tagged signal model. In this extension, processes compute a function from input signals to output signals, where a signal is defined as a partial function from a down set of tags to a set of values. An example of such a signal is shown in Figure 1.

The requirement that every process be Scott-continuous is reasonable as it means, informally, that a process does not need infinite input to produce its output. However, it is often tedious to formally prove that a process is Scott-continuous. The purpose of this paper is to present theorems on Scott-continuity that simplify such proofs in the tagged signal model. Section 2 presents a brief overview of the tagged signal model. Section 3 proves a general theorem on the necessary and sufficient conditions for the Scott-continuity of tagged signal processes. In Section 4, the Scott-continuity of a broad class of processes is shown, namely the class of exactly causal processes, which includes the class of stateless processes as well as a subset of discrete-event processes. Section 5 presents sufficient conditions for the Scott-continuity of domain-warping processes such as delays. Section 6 shows an example of a complex Scott-continuous tagged signal process composed of several simpler processes. Section 7 concludes and presents future directions.

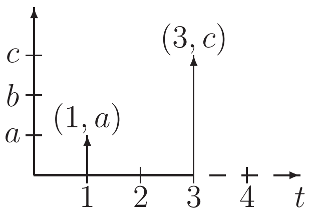

Figure 1: A timed signal defined on domain [0,3] with two present events at $t = 1$ and $t = 3$. The signal takes an absent value ε for $t \in [0, 3] \setminus \{1, 3\}$.

2. Tagged signal model

In order to make this paper self-contained, a brief overview of the tagged signal model is presented here. For more information, see [5, 6].

Signals are the basic building block of the tagged signal model. Each signal represents a communication flow on a channel and has a value for each tag in its domain of definition, where tags may represent time or causality. Each signal is defined on a down set of tags, which means that if a signal is defined for a given tag, then it is also defined for each tag less than or equal to the given tag. Formally,

Definition 2.1 (Partially ordered set)**.** Let X be a set and $\leq$ be a binary relation over X. The tuple $(X, \leq)$ is a *partially ordered set* if, for each $a, b, c \in X$, we have that:

(i) $a \leq a$ (reflexivity);

(ii) if $a \leq b$ and $b \leq a$, then $a = b$ (antisymmetry);

(iii) if $a \leq b$ and $b \leq c$, then $a \leq c$ (transitivity).

Definition 2.2 (Down set)**.** Let $(T, \leq)$ be a partially ordered set. A set $T' \subseteq T$ is a *down set* of T if, for each $t \in T$ and $t' \in T'$, we have $t \leq t' \Rightarrow t \in T'$.

Definition 2.3 (Signal)**.** A *signal* s is a partial function $s : T \rightarrow V$ such that:

(i) $(T, \leq)$ is a partially ordered set (the tag set);

(ii) V is an alphabet of values;

(iii) Its domain of definition, denoted $\operatorname{dom}(s)$, is a down set of T.

The set of signals with tag set $(T, \leq)$ and values V is denoted $S(T, V)$. Common choices for the tag set include the natural numbers ($\mathbb{N}$) and the non-negative real numbers ($\mathbb{R}_0^+$). More complex tag sets, such as $\mathbb{R}_0^+ \times \mathbb{N}$, are also used to model super-dense time (e.g. delta-cycles in hardware description languages).

In the tagged signal model, a process is a function from its input signals to its output signals. For example, a process with one input and one output has the following form: $f : S(T_i, V_i) \to S(T_o, V_o)$. When modeling a timed process network, the tag set is a totally ordered set (such that $t_1 \leq t_2$ or $t_2 \leq t_1$ for each $t_1, t_2 \in T$) and all signals share the same tag set. The sample process would then have the following form: $f : S(T, V_i) \to S(T, V_o)$. Signals in a timed process network can have a special value (ε), which represents the absence of a regular value. Formally,

Definition 2.4 (Timed signal)**.** A *timed signal* $s \in S(T, V)$ is a signal represented as a tuple $(\operatorname{dom}(s), E)$ such that:

(i) $(T, \leq)$ is a totally ordered set;

(ii) V is an alphabet of values such that $\varepsilon \in V$ represents an absent value;

(iii) $E = \{(t, s(t)) \,|t \in \operatorname{dom}(s), s(t) \neq \varepsilon\}$ is the set of present events.

For example, if s is the signal illustrated in Figure 1, then $\operatorname{dom}(s) = [0, 3]$ and $E = \{(1, a), (3, c)\}$. The signal s has the absent value ε for each $t \in [0, 3]$ except for $t = 1$ and $t = 3$.

Signals can be partially ordered under the prefix order, where a signal r is a prefix of s if both signals are identical up to a certain tag t, and then only s may be defined for tags greater than t. Formally,

Definition 2.5 (Prefix order on signals)**.** Let $r, s \in S(T, V)$, r is a *prefix* of s, denoted $r \sqsubseteq s$, if:

(i) $\operatorname{dom}(r) \subseteq \operatorname{dom}(s)$;

(ii) $r(t) = s(t), \forall t \in \operatorname{dom}(r)$.

It has been shown in [5] that, for each $(T, \leq)$ and V, the set of signals $S(T, V)$ with the prefix order $\sqsubseteq$ is a complete partial order. This allows the application of the Kahn principle to tagged signal process networks, where the output signal of a process may be the input signal of another process. If every tagged signal process is a Scott-continuous function from its input signals to its output signals, then the least fixed point of the network functional exists and is the unique behavior of the process network. The rest of the paper focuses on the Scott-continuity of tagged processses.

3. A general Scott-continuity theorem

A function $f : X \to Y$ between two partially ordered sets $(X, \leq)$ and $(Y, \leq)$ is said to be *Scott-continuous* if it preserves the supremum of all directed subsets. A *directed subset* $D \subseteq X$ is a set such that, for each $a, b \in D$, there exists $c \in D$ with $a \leq c$ and $b \leq c$. The element c is then an upper bound of set $\{a, b\}$. More generally, u is an *upper bound* of set D if $d \leq u$ for each $d \in D$. If s is an upper bound of set D and we have $s \leq u$ for each upper bound u of D, then s is the *least upper bound* or *supremum* of D and is denoted $\bigsqcup D$. If each directed subset D of a partially ordered set $(X, \leq)$ has a supremum $\bigsqcup D$, then $(X, \leq)$ is a *directed complete partial order*. Scott-continuity is thus defined formally as follows:

Definition 3.1 (Scott-continuity)**.** Let $f : X \to Y$ be a function between two directed complete partial orders $(X, \leq)$ and $(Y, \leq)$. The function f is *Scott-continuous* if, for each directed subset $D \subseteq X$, we have that $\bigsqcup f(D)$ exists and $f(\bigsqcup D) = \bigsqcup f(D)$, where $f(D) = \{f(d) \,|d \in D\}$.

Proving that such a function $f : X \to Y$ is Scott-continuous is often tedious, since one must consider all possible directed subsets, which may include infinite directed subsets. Generally, the first step consists in proving that the function is *monotonic*, meaning that $a \leq b$ implies $f(a) \leq f(b)$ for each $a, b \in X$. As shown in Appendix A, if f is monotonic and D is a directed set, then $f(D)$ is also a directed set and $\bigsqcup f(D)$ exists with $\bigsqcup f(D) \leq f(\bigsqcup D)$. The following lemma can thus be derived, as proven in Appendix A:

Lemma 3.2. *Let $(X, \leq)$ and $(Y, \leq)$ be directed complete partial orders. A function $f : X \to Y$ is Scott-continuous if and only if the following conditions hold:*

(i) f *is monotonic.*

(ii) $f\left(\bigsqcup D\right) \leq \bigsqcup f(D)$ *for each directed subset* $D \subseteq X$*.*

For each $(T, \leq)$ and V, the set of signals $(S(T,V), \sqsubseteq)$ is a complete partial order [5], and thus also a directed complete partial order (a *complete partial order* is a directed complete partial order with a least element). We therefore have the following corollary:

Corollary 3.3. *Let* $(X, \leq)$ *be a directed complete partial order. A function* $f : X \to S(T_o, V_o)$ *is Scott-continuous if and only if the following conditions hold:*

(i) f *is monotonic.*

(ii) $f\left(\bigsqcup D\right) \sqsubseteq \bigsqcup f(D)$ *for each directed subset* $D \subseteq X$*.*

It turns out that condition (ii) can be weakened such that only the domains of suprema need to be considered. This significantly simplifies proofs of Scott-continuity for tagged signal processes.

Theorem 3.4. *Let* $(X, \leq)$ *be a directed complete partial order. A function* $f : X \to S(T_o, V_o)$ *is Scott-continuous if and only if the following conditions hold:*

(i) f *is monotonic.*

(ii) $\mathrm{dom}\left(f\left(\bigsqcup D\right)\right) \subseteq \mathrm{dom}\left(\bigsqcup f(D)\right)$*, for each directed subset* $D \subseteq X$*.*

Proof. By Corollary 3.3, if condition (i) does not hold, then f is not Scott-continuous. If (ii) does not hold, then there exists a directed set D such that $\mathrm{dom}\left(f\left(\bigsqcup D\right)\right) \not\subseteq \mathrm{dom}\left(\bigsqcup f(D)\right)$ and, by Definition 2.5, we have $f\left(\bigsqcup D\right) \not\sqsubseteq \bigsqcup f(D)$. Therefore, f is not Scott-continuous by Corollary 3.3. Thus, f is Scott-continuous only if both conditions (i) and (ii) hold. We next show that if both conditions (i) and (ii) hold, then f is Scott-continuous.

If (i) holds, then we have $\bigsqcup f(D) \sqsubseteq f\left(\bigsqcup D\right)$ for each directed set $D \subseteq X$. By Definition 2.5, we thus have $\bigsqcup(f(D))(t) = (f\left(\bigsqcup D\right))(t)$ for each $t \in \mathrm{dom}(\bigsqcup f(D))$. If (ii) also holds, then we have $(f\left(\bigsqcup D\right))(t) = \bigsqcup(f(D))(t)$ for each $t \in \mathrm{dom}(f\left(\bigsqcup D\right))$ and thus $f\left(\bigsqcup D\right) \sqsubseteq \bigsqcup f(D)$. Since $\sqsubseteq$ is antisymmetric, this implies $f\left(\bigsqcup D\right) = \bigsqcup f(D)$ and f is thus Scott-continuous. □

Several corollaries follow immediately from this general theorem. The first one defines when a function with a single input signal and a single output signal is Scott-continuous:

Corollary 3.5. *A function $f : S(T_i, V_i) \to S(T_o, V_o)$ is Scott-continuous if and only if the following conditions hold:*

(i) f *is monotonic.*

(ii) $\mathrm{dom}\left(f\left(\bigsqcup D\right)\right) \subseteq \mathrm{dom}\left(\bigsqcup f\left(D\right)\right)$*, for each directed signal subset* $D \subseteq S(T_i, V_i)$.

A more general corollary covers functions with $m \in \mathbb{N}$ input signals and one output signal. Such functions have the form $f : S(T_{i,1}, V_{i,1}) \times S(T_{i,2}, V_{i,2}) \times \cdots \times S(T_{i,m}, V_{i,m}) \to S(T_o, V_o)$. Following [1, Proposition 3.2.2], $S(T_{i,1}, V_{i,1}) \times S(T_{i,2}, V_{i,2}) \times \cdots \times S(T_{i,m}, V_{i,m})$ is a directed complete partial order where tuples of signals are ordered by the pointwise prefix order (meaning that $(a_1, a_2, \ldots, a_m) \sqsubseteq (b_1, b_2, \ldots, b_m)$ is equivalent to $a_j \sqsubseteq b_j$ for each $1 \le j \le m$).

Corollary 3.6. *A function on signals $f : S(T_{i,1}, V_{i,1}) \times S(T_{i,2}, V_{i,2}) \times \cdots \times S(T_{i,m}, V_{i,m}) \to S(T_o, V_o)$ is Scott-continuous if and only if the following conditions hold:*

(i) f *is monotonic.*

(ii) $\mathrm{dom}\left(f\left(\bigsqcup D\right)\right) \subseteq \mathrm{dom}\left(\bigsqcup f\left(D\right)\right)$*, for each directed signal subset* $D \subseteq S(T_{i,1}, V_{i,1}) \times S(T_{i,2}, V_{i,2}) \times \cdots \times S(T_{i,m}, V_{i,m})$.

A still more general corollary covers functions with $m \in \mathbb{N}$ input signals and $n \in \mathbb{N}$ output signals, having the form $f : S(T_{i,1}, V_{i,1}) \times S(T_{i,2}, V_{i,2}) \times \cdots \times S(T_{i,m}, V_{i,m}) \to S(T_{o,1}, V_{o,1}) \times S(T_{o,2}, V_{o,2}) \times \cdots \times S(T_{o,n}, V_{o,n})$. Such a function can be decomposed into n separate functions such that $f_j : S(T_{i,1}, V_{i,1}) \times S(T_{i,2}, V_{i,2}) \times \cdots \times S(T_{i,m}, V_{i,m}) \to S(T_{o,j}, V_{o,j})$ for each $1 \le j \le n$. Thus, the function f is Scott-continuous if and only if f_j is Scott-continuous for each $1 \le j \le n$. Corollary 3.6 can then be applied separately to each f_j. If we define the pointwise domain operator such that $\mathrm{dom}(x_1, x_2, \ldots, x_n) = (\mathrm{dom}(x_1), \mathrm{dom}(x_2), \ldots, \mathrm{dom}(x_n))$ and the pointwise subset operator such that $(a_1, a_2, \ldots, a_n) \subseteq (b_1, b_2, \ldots, b_n)$ is equivalent to $a_j \subseteq b_j$ for each $1 \le j \le n$, then we get the following corollary:

Corollary 3.7. *A function $f : S(T_{i,1}, V_{i,1}) \times S(T_{i,2}, V_{i,2}) \times \cdots \times S(T_{i,m}, V_{i,m}) \to S(T_{o,1}, V_{o,1}) \times S(T_{o,2}, V_{o,2}) \times \cdots \times S(T_{o,n}, V_{o,n})$ is Scott-continuous if and only if the following conditions hold:*

(i) f *is monotonic.*

(ii) $\mathrm{dom}\,(f\,(\bigsqcup D)) \subseteq \mathrm{dom}\,(\bigsqcup f\,(D))$*, for each directed signal subset* $D \subseteq S(T_{i,1}, V_{i,1}) \times S(T_{i,2}, V_{i,2}) \times \cdots \times S(T_{i,m}, V_{i,m})$*.*

This section proved a general theorem on the Scott-continuity of signal-producing functions and showed it could be applied to tagged signal processes with an arbitrary number of input and output signals. The following sections apply this theorem and its corollaries to specific yet broad classes of tagged signal processes.

4. Exactly causal processes are Scott-continuous

In this section, we prove the Scott-continuity of a broad class of processes: exactly causal processes, which are monotonic and whose output signals are defined only for all tags where all input signals are defined.

4.1. Exactly causal processes

A causal process is usually defined as a process whose outputs depend only on past or current values of its inputs, but not on their future values. In the tagged signal model, a *causal* process is thus a monotonic process whose output signals are defined for all tags where all input signals are defined. Thus, all input and output signals of a causal process must share the same tag set. If each output signal of a causal process is also always defined for at least one later tag, then this process is a *strictly causal process.* An *exactly causal process* is a causal process whose outputs are defined nowhere except for all tags where all input signals are defined. Formally,

Definition 4.1 (Exact causality)**.** A function on signals $f : S(T, V_{i,1}) \times S(T, V_{i,2}) \times \cdots \times S(T, V_{i,m}) \to S(T, V_{o,1}) \times S(T, V_{o,2}) \times \cdots \times S(T, V_{o,n})$ is *exactly causal* if the following conditions hold:

(i) f is monotonic;

(ii) For each $x_1, x_2, \ldots, x_m$ and $y_1, y_2, \ldots, y_n$ such that $(y_1, y_2, \ldots, y_n) = f(x_1, x_2, \ldots, x_m)$, we have $\mathrm{dom}\,(y_k) = \bigcap_{1 \le j \le m} \mathrm{dom}\,(x_j)$, for $1 \le k \le n$.

We prove that all exactly causal processes are Scott-continuous by applying Corollary 3.6 and a lemma and corollary on the domains of supremum of signals.

Lemma 4.2. *If $D \subseteq S(T, V)$ is a directed signal subset, then* $\mathrm{dom}\,(\bigsqcup D) = \bigcup_{d \in D} \mathrm{dom}\,(d)$.

Proof. See Appendix A. □

Corollary 4.3. *Let $(X, \leq)$ be a partially ordered set. If $f : X \to S(T_o, V_o)$ is a monotonic function and $D \subseteq X$ is a directed subset, then* $\mathrm{dom}\,(\bigsqcup f\,(D)) = \bigcup_{d \in D} \mathrm{dom}\,(f\,(d))$.

Proof. Let $Y = f\,(D)$. Because f is monotonic, Y is a directed subset of $S(T_o, V_o)$. By Lemma 4.2, $\mathrm{dom}\,(\bigsqcup Y) = \bigcup_{y \in Y} \mathrm{dom}\,(y)$. By definition, for each $y \in Y$, there exists $d \in D$ such that $y = f\,(d)$. Therefore, $\bigcup_{y \in Y} \mathrm{dom}\,(y) = \bigcup_{d \in D} \mathrm{dom}\,(f\,(d))$. □

Theorem 4.4. *If $f : S(T, V_{i,1}) \times S(T, V_{i,2}) \times \cdots \times S(T, V_{i,m}) \to S(T, V_o)$ is an exactly causal function, then it is Scott-continuous.*

Proof. Each directed subset $D \subseteq S(T, V_{i,1}) \times S(T, V_{i,2}) \times \cdots \times S(T, V_{i,m})$ can be decomposed pointwise into m subsets such that $D = D_1 \times D_2 \times \cdots \times D_m$ and $D_j \subseteq S(T, V_{i,j})$ for $1 \leq j \leq m$. By [1, Proposition 3.2.2], we have that $\bigsqcup D$ exists and is equal to $(\bigsqcup D_1, \bigsqcup D_2, \ldots, \bigsqcup D_m)$.

By Corollary 4.3, we have $\mathrm{dom}\,(\bigsqcup f(D)) = \bigcup_{d \in D} \mathrm{dom}\,(f\,(d))$. Because f is exactly causal, this is equal to $\bigcup_{d \in D} \left(\bigcap_{1 \leq j \leq m} \mathrm{dom}\,(d_j) \right)$ with $d = (d_1, d_2, \ldots, d_m)$. Because the union and intersection operators are distribituve on sets, this equals $\bigcap_{1 \leq j \leq m} \left(\bigcup_{d_j \in D_j} \mathrm{dom}\,(d_j) \right)$ which, by Lemma 4.2, gives $\bigcap_{1 \leq j \leq m} \mathrm{dom}\,(\bigsqcup D_j)$. As f is exactly causal, this equals $\mathrm{dom}\,(f\,(\bigsqcup D))$ and thus $\mathrm{dom}\,(f\,(\bigsqcup D)) = \mathrm{dom}\,(\bigsqcup f\,(D))$. Therefore, f is Scott-continuous by Corollary 3.6. □

As in the previous Section, this theorem can be easily extended to exactly causal functions with m inputs and n outputs by decomposing such functions into n separate functions each having m inputs and one output.

Corollary 4.5. *If $f : S(T, V_{i,1}) \times S(T, V_{i,2}) \times \cdots \times S(T, V_{i,m}) \to S(T, V_{o,1}) \times S(T, V_{o,2}) \times \cdots \times S(T, V_{o,n})$ is an exactly causal function, then it is Scott-continuous.*

Therefore, all exactly causal processes are Scott-continuous. For all processes whose output signals are defined only for all tags where all input signals are defined, proving the monotonicity of the process is thus sufficient to prove its Scott-continuity.

Corollary 4.6. *Let $f : S(T, V_{i,1}) \times S(T, V_{i,2}) \times \cdots \times S(T, V_{i,m}) \to S(T, V_{o,1}) \times S(T, V_{o,2}) \times \cdots \times S(T, V_{o,n})$. If, for each $x_1, x_2, \ldots, x_m$ and $y_1, y_2, \ldots, y_n$ such that $(y_1, y_2, \ldots, y_n) = f(x_1, x_2, \ldots, x_m)$, we have $\mathrm{dom}\,(y_k) = \bigcap_{1 \le j \le m} \mathrm{dom}\,(x_j)$, for $1 \le k \le n$, then the following statements are equivalent:*

(i) f is monotonic;

(ii) f is exactly causal;

(iii) f is Scott-continuous.

4.2. Stateless processes

The value of each output signal of a stateless process at a given tag depends only on the values of its input signals at the same tag. These processes are called stateless because their output values at a given tag do not depend on past input values. Stateless processes are commonly used to model combinatorial components in the tagged signal model. We show that stateless processes are monotonic and are a subclass of exactly causal processes. Stateless processes are thus Scott-continuous.

Definition 4.7 (Statelessness)**.** A function $f : S(T, V_{i,1}) \times S(T, V_{i,2}) \times \cdots \times S(T, V_{i,m}) \to S(T, V_{o,1}) \times S(T, V_{o,2}) \times \cdots \times S(T, V_{o,n})$ is *stateless* if the following conditions hold:

(i) For each $x_1, x_2, \ldots, x_m$ and $y_1, y_2, \ldots, y_n$ such that $(y_1, y_2, \ldots, y_n) = f(x_1, x_2, \ldots, x_m)$, we have $\mathrm{dom}\,(y_k) = \bigcap_{1 \le j \le m} \mathrm{dom}\,(x_j)$, for $1 \le k \le n$;

(ii) For $1 \le k \le n$ and for each $t \in \mathrm{dom}(y_k)$, we have $g_k : V_{i,1} \times V_{i,2} \times \cdots \times V_{i,m} \to V_{o,k}$ such that $y_k(t) = g_k(x_1(t), x_2(t), \ldots, x_m(t))$.

Theorem 4.8. *If $f : S(T, V_{i,1}) \times S(T, V_{i,2}) \times \cdots \times S(T, V_{i,m}) \to S(T, V_{o,1}) \times S(T, V_{o,2}) \times \cdots \times S(T, V_{o,n})$ is a stateless function, then it is monotonic, exactly causal and Scott-continuous.*

Proof. Let $(a_1, a_2, \ldots, a_m)$ and $(b_1, b_2, \ldots, b_m)$ be tuples of input signals such that $(a_1, a_2, \ldots, a_m) \sqsubseteq (b_1, b_2, \ldots, b_m)$. The tuples of output signals are then $(r_1, r_2, \ldots, r_n) = f(a_1, a_2, \ldots, a_m)$ and $(s_1, s_2, \ldots, s_n) = f(b_1, b_2, \ldots, b_m)$. By Definition 2.5, we have $\mathrm{dom}(a_j) \subseteq \mathrm{dom}(b_j)$ for $1 \leq j \leq m$ and, by condition (i) of Definition 4.7, this implies $\mathrm{dom}(r_k) \subseteq \mathrm{dom}(s_k)$ for $1 \leq k \leq n$.

By condition (ii) of Definition 4.7, $r_k(t) = g_k(a_1(t), a_2(t), \ldots, a_m(t))$ for each $t \in \mathrm{dom}(r_k)$ and $s_k(t) = g_k(b_1(t), b_2(t), \ldots, b_m(t))$ for each $t \in \mathrm{dom}(s_k)$. For $1 \leq j \leq m$, we have $a_j(t) = b_j(t)$ for each $t \in \mathrm{dom}(a_j)$ because $a_j \sqsubseteq b_j$. By condition (i) of Definition 4.7, we have that $\mathrm{dom}(r_k) \subseteq \mathrm{dom}(a_j)$ for $1 \leq j \leq m$ and $1 \leq k \leq n$. Thus, we have $a_j(t) = b_j(t)$ for each $t \in \mathrm{dom}(r_k)$ and $r_k(t) = s_k(t)$ for each $t \in \mathrm{dom}(r_k)$. This implies that $r_k \sqsubseteq s_k$ for each $1 \leq k \leq n$ and thus $(r_1, r_2, \ldots, r_n) \sqsubseteq (s_1, s_2, \ldots, s_n)$. Therefore f is monotonic and it follows from Corollary 4.6 that f is also exactly causal and Scott-continuous. □

Exemples of stateless processes include the merge process [5, 6], the add process [5] and the decimator process, which respectively merge, add and decimate together two data streams.

Example 4.9. A merge process produces an output signal in which the present events of its two input signals are merged, such that only the present events of the first signal are transmitted if both signals are present at exactly the same tag. Formally, a merge is a function $merge : S(T, V_1) \times S(T, V_2) \to S(T, V_1 \cup V_2)$ such that, for each $z = merge\,(x, y)$, we have:

(i) $\mathrm{dom}\,(z) = \mathrm{dom}\,(x) \cap \mathrm{dom}\,(y)$;

(ii) For each $t \in \mathrm{dom}\,(z)$, $z\,(t) = \begin{cases} x\,(t) & \text{if } x\,(t) \neq \varepsilon \\ y\,(t) & \text{otherwise} \end{cases}$

Let $g : V_1 \times V_2 \to V_1 \cup V_2$ such that $g\,(v, w) = \begin{cases} v & \text{if } v \neq \varepsilon \\ w & \text{otherwise} \end{cases}$

We then have $z\,(t) = g\,(x\,(t)\,, y\,(t))$ and, by Definition 4.7, a merge is a stateless process. By Theorem 4.8, it is thus monotonic, exactly causal and Scott-continuous.

It may seem counterintuitive that this first-come, first-served merge is monotonic and Scott-continuous. Indeed, first-come, first-served merge is a special case of fair merge, which has been proven to be neither monotonic nor Scott-continuous in untimed dataflow [7]. First-come, first-served merge has also been shown to be neither monotonic nor Scott-continuous in the timed dataflow model of Yates [8] (a similar model is used in an earlier version of the tagged signal model [4, 3]). The critical difference is that the prefix order of signals in the (new) tagged signal model takes into account their domains of definition. Table 1 illustrates the behavior of a first-come, first-served merge in untimed dataflow, Yates dataflow and the tagged signal model. In each case, we have outputs $z1, z2$ such that $z_1 \not\sqsubseteq z_2$. In both untimed and Yates dataflow, we have inputs $x1, x2$ such that $x_1 \sqsubseteq x_2$, which shows that first-come, first-served merge is not monotonic, and thus not Scott-continuous, in these models. However, in the tagged signal model, we have that $x_1 \not\sqsubseteq x_2$ because, while both signals have the same domain, we have $x_1(2) = \varepsilon$ and $x_2(2) = b$. If instead, we define $x_3 = ([0,1], \{(1,a)\})$, such that $x_3 \sqsubseteq x_2$, then we have $z_3 = ([0,1], \{(1,a)\})$ and $z_3 \sqsubseteq z_2$. This behavior is consistent with the monotonocity of first-come, first-served merge in the tagged signal model.

Table 1: Signals of first-come, first-served merge, with $z_1 = \mathit{merge}(x_1, y)$ and $z_2 = \mathit{merge}(x_2, y)$

Model	**Inputs** x_1, x_2	**Input** y	**Outputs** z_1, z_2
Untimed	(a)	(c)	(a, c)
dataflow	(a, b)	(c)	(a, b, c)
Yates	$\{(1, a)\}$	$\{(3, c)\}$	$\{(1, a), (3, c)\}$
dataflow	$\{(1, a), (2, b)\}$	$\{(3, c)\}$	$\{(1, a), (2, b), (3, c)\}$
Tagged	$([0, 3], \{(1, a)\})$	$([0, 3], \{(3, c)\})$	$([0, 3], \{(1, a), (3, c)\})$
signal model	$([0, 3], \{(1, a), (2, b)\})$	$([0, 3], \{(3, c)\})$	$([0, 3], \{(1, a), (2, b), (3, c)\})$

Example 4.10. An add process behaves similarly to a merge process, except when events are present at the same tag in the two input signals. The output signal at this tag then takes the value of the sum of the inputs. Formally, an add is a function $\mathit{add} : S(T, V) \times S(T, V) \to S(T, V)$ such that $+$ is a binary operation over V and, for each $z = \mathit{add}\,(x, y)$, we have:

(i) $\mathrm{dom}\,(z) = \mathrm{dom}\,(x) \cap \mathrm{dom}\,(y)$;

(ii) For each $t \in \mathrm{dom}\,(z)$, $z\,(t) = \begin{cases} x\,(t) + y\,(t) & \text{if } x\,(t) \neq \varepsilon \text{ and } y\,(t) \neq \varepsilon \\ x\,(t) & \text{if } x\,(t) \neq \varepsilon \text{ and } y\,(t) = \varepsilon \\ y\,(t) & \text{otherwise} \end{cases}$

Let $g : V \times V \to V$ such that $g\,(v, w) = \begin{cases} v + w & \text{if } v \neq \varepsilon \text{ and } w \neq \varepsilon \\ v & \text{if } v \neq \varepsilon \text{ and } w = \varepsilon \\ w & \text{otherwise} \end{cases}$

We then have $z\,(t) = g\,(x\,(t)\,, y\,(t))$ and, by Definition 4.7, an add is a stateless process. It is thus monotonic, exactly causal and Scott-continuous.

Example 4.11. A decimator process produces an output signal which contains all present events of the first input signal, except for tags where the second signal is present. At these tags, the output signal is always absent and the second input signal is said to decimate the first one. Formally, a decimator is a function $dec : S(T, V_1) \times S(T, V_2) \to S(T, V_1)$ such that, for each $z = dec\,(x, y)$, we have:

(i) $\mathrm{dom}\,(z) = \mathrm{dom}\,(x) \cap \mathrm{dom}\,(y)$;

(ii) For each $t \in \mathrm{dom}\,(z)$, $z\,(t) = \begin{cases} x\,(t) & \text{if } y\,(t) = \varepsilon \\ \varepsilon & \text{otherwise} \end{cases}$

Let $g : V_1 \times V_2 \to V_1$ such that $g\,(v, w) = \begin{cases} v & \text{if } w = \varepsilon \\ \varepsilon & \text{otherwise} \end{cases}$

We then have $z\,(t) = g\,(x\,(t)\,, y\,(t))$ and, by Definition 4.7, a decimator is a stateless process. It is thus monotonic, exactly causal and Scott-continuous.

4.3. Exactly causal discrete-event processes

Not all exactly causal processes are stateless. An exactly causal process may have state, in that the current values of its output signals depend not only on the current values of its input signals, but also on their histories. In this section, we consider stateful processes which are discrete-event processes. Both the input and output signals of such processes are discrete-event signals. A discrete-event signal is a timed signal such that there is a finite number of present events between any two given tags in its domain of definition. Discrete-event signals were formally defined in [5, 6] as follows:

Definition 4.12 (Preimage)**.** Let X and Y be two sets and let $f : X \to Y$. The *preimage* of set $R \subseteq Y$ is $f^{-1}(R) = \{x \in X | f(x) \in R\}$.

Definition 4.13 (Discrete-event signal)**.** Let $s \in S(T, V)$ be a timed signal and let $P = s^{-1}(V \setminus \{\varepsilon\})$ be the set of tags with present values. The signal s is a *discrete-event signal* if the following conditions hold:

(i) There exists $K \subseteq \mathbb{N}$ and a function $f : K \to P$ such that:

   (a) K is a down set of $\mathbb{N}$;

   (b) $i \leq j \Leftrightarrow f(i) \leq f(j)$ (f is an order-isomorphism between P and a down set of $\mathbb{N}$);

(ii) If P is an infinite set, then $\mathrm{dom}(s) = \bigcup_{t \in P} D(t)$, where $D(t) = \{t' \in T | t' \leq t\}$.

The set of discrete-event signals with tag set $(T, \leq)$ and values V is denoted $S_d(T, V)$ and was shown in [5] to be a complete partial order and a down set of $S(T, V)$. Since the order-isomorphism in condition (i) of Definition 4.13 is a function from a down set of $\mathbb{N}$ to a set of tags, it is a member of $S(\mathbb{N}, T)$. We define the function $\tau : S_d(T, V) \to S(\mathbb{N}, T)$ as an operator which maps a discrete-event signal to the order-isomorphism between a down set of $\mathbb{N}$ and its set of tags with present values. We define another function $\nu : S_d(T, V) \to S(\mathbb{N}, V)$ as an operator which maps a discrete-event signal to an enumeration of all present values ordered by their tag. Thus, for each $s \in S_d(T, V)$, we have $\mathrm{dom}(\nu(s)) = \mathrm{dom}(\tau(s))$ and $\nu(s)(i) = s(\tau(s)(i))$ for each $i \in \mathrm{dom}(\tau(s))$. It is easy to see that τ and ν are monotonic functions. The operators τ and ν are illustrated by examples in Table 2.

Table 2: Operators τ and ν applied to sample signals

Signal s	$\tau(s)$	$\nu(s)$
$([0, 10], \{(1, a), (3, c), (6, b)\})$	$([0, 2], \{(0, 1), (1, 3), (2, 6)\})$	$([0, 2], \{(0, a), (1, c), (2, b)\})$
$([0, 10], \{(0.5, r), (2.84, s)\})$	$([0, 1], \{(0, 0.5), (1, 2.84)\})$	$([0, 1], \{(0, r), (1, s)\})$

We model a stateful discrete-event process as the composition of two functions. The first function determines when present events occur on the output signal, by computing an output sequence of tags from the sequences of tags of the input signals. The second function determines the value of each of these present events in the output signal. Finally, the constraint on the output signal's domain ensures that if the process is monotonic, then it is also exactly causal and Scott-continuous.

Definition 4.14 (Strictly increasing signal)**.** A signal $s \in S(T, V)$ is *strictly increasing* if, for each $t_1, t_2 \in T$ such that $t_1 < t_2$, we have $s(t_1) < s(t_2)$.

Definition 4.15 (Sequence-based function)**.** A *sequence-based function* is a function $f : S_d(T, V_{i,1}) \times S_d(T, V_{i,2}) \times \cdots \times S_d(T, V_{i,m}) \to S_d(T, V_o)$ with $g : S(\mathbb{N}, T)^m \to S(\mathbb{N}, T)$ and $h : S(\mathbb{N}, V_{i,1}) \times S(\mathbb{N}, V_{i,2}) \times \cdots \times S(\mathbb{N}, V_{i,m}) \to S(\mathbb{N}, V_o)$ such that, for each $x = (x_1, x_2, \ldots, x_m)$ and $y = f(x)$, we have:

(i) $\mathrm{dom}(y) = \bigcap\limits_{1 \leq j \leq m} \mathrm{dom}\,(x_j)$;

(ii) $y(t) = \begin{cases} h(\nu(x))(n) & \text{if } \exists n \in \mathbb{N} \text{ such that } g(\tau(x))(n) = t \\ \varepsilon & \text{otherwise} \end{cases}$

(iii) g is closed on strictly increasing signals.

(iv) $\mathrm{dom}(g(\tau(s))) \subseteq \mathrm{dom}(h(\nu(s)))$ for each $s \in S_d(T, V_{i,1}) \times S_d(T, V_{i,2}) \times \cdots \times S_d(T, V_{i,m})$.

If s is a discrete-event signal, then $\tau(s)$ is a strictly increasing signal. Condition (iii) ensures that the sequence of tags in the output signal is also strictly increasing.

The following theorem gives sufficient conditions for the monotonicity, exact causality and Scott-continuity of a sequence-based function. For $a \in S(\mathbb{N}, T)$, we note $t \in a$ if $a^{-1}(\{t\}) \neq \emptyset$ and $t \notin a$ if $a^{-1}(\{t\}) = \emptyset$.

Theorem 4.16. *A sequence-based function $f : S_d(T, V_{i,1}) \times S_d(T, V_{i,2}) \times \cdots \times S_d(T, V_{i,m}) \to S_d(T, V_o)$, with $g : S(\mathbb{N}, T)^m \to S(\mathbb{N}, T)$ and $h : S(\mathbb{N}, V_{i,1}) \times S(\mathbb{N}, V_{i,2}) \times \cdots \times S(\mathbb{N}, V_{i,m}) \to S(\mathbb{N}, V_o)$, is monotonic, exactly causal and Scott-continuous if the following conditions hold:*

(i) h is monotonic;

(ii) For each $r, s \in S_d(T, V_{i,1}) \times S_d(T, V_{i,2}) \times \cdots \times S_d(T, V_{i,m})$ such that $r \sqsubseteq s$, we have:

(a) $t \notin g(\tau(r))$ and $t \in \mathrm{dom}\,(f(r)) \Rightarrow t \notin g(\tau(s))$;

(b) $t \in g(\tau(r))$ and $t \in \mathrm{dom}\,(f(r)) \Rightarrow t \in g(\tau(s))$;

Proof. Let $x = f(r)$ and $y = f(s)$. If $r \sqsubseteq s$, then $\text{dom}(r) \subseteq \text{dom}(s)$ and, by condition(i) of Definition 4.15, this means that $\text{dom}(x) \subseteq \text{dom}(y)$.

For each $t \in \text{dom}(x)$, if $x(t) = \varepsilon$, then $t \notin g(\tau(r))$ and, by condition (ii.a), we have $t \notin g(\tau(s))$ and thus $y(t) = \varepsilon$.

If $x(t) \neq \varepsilon$, then $t \in g(\tau(r))$ and, by condition (ii.b), we have $t \in g(\tau(s))$ and thus $y(t) \neq \varepsilon$. In this case, we have $x(t) = h(\nu(r))(m)$ and $y(t) = h(\nu(s))(n)$ with $t \in \text{dom}(x)$ and $t = g(\tau(r))(m) = g(\tau(s))(n)$. By condition (ii) and the fact that g is closed on strictly increasing signals, we have that $g(\tau(r))$ and $g(\tau(s))$ are equal for all tags in $\text{dom}(x)$ and we thus have $m = n$. By condition (i), we have $h(\nu(r)) \sqsubseteq h(\nu(s))$ and thus $h(\nu(r))(m) = h(\nu(s))(n)$. Therefore, $x(t) = y(t)$ for each $t \in \text{dom}(x)$ and thus $x \sqsubseteq y$. We have shown that $r \sqsubseteq s$ implies $f(r) \sqsubseteq f(s)$ and f is thus monotonic. It follows from Corollary 4.6 that f is also exactly causal and Scott-continuous. □

Again, this theorem can be easily extended to sequence-based functions with m inputs and n outputs by decomposing such functions into n separate functions each having m inputs and one output.

Example 4.17. The first example is a Kahn FIFO, which is an infinite FIFO with non-blocking writes and blocking reads. As shown in Figure 2, a Kahn FIFO has a data input signal (x) and a data output signal (y), as well as an explicit request signal (r). When an event with the special value α is present on the request signal, this means that the reading process is requesting a new data value. As shown in Table 3, a Kahn channel answers a request for a data value only once it has received this data value. The time when a Kahn channel outputs a data value is thus the maximum of the time when it receives this data value and the time when it receives a request for it. A Kahn channel therefore does a pointwise maximum on the sequences of tags of its inputs. Formally, a Kahn FIFO is a function $kahn : S_d(T, V) \times S_d(T, V_\alpha) \rightarrow S_d(T, V)$ with $V_\alpha = \{\alpha, \varepsilon\}$ such that, for each $y = kahn(x, r)$, we have:

(i) $\text{dom}(y) = \text{dom}(x) \cap \text{dom}(r)$;

(ii) $$y(t) = \begin{cases} \nu(x)(n) & \text{if } \exists n \in \mathbb{N} \text{ such that } \max(\tau(x)(n), \tau(r)(n)) = t \\ \varepsilon & \text{otherwise} \end{cases}$$

By (i), a Kahn FIFO meets condition (i) of Definition 4.15. Let $g : S(\mathbb{N}, T)^2 \rightarrow S(\mathbb{N}, T)$ such that, for each $a, b, c \in S(\mathbb{N}, T)$ with $c = g(a, b)$, we have $\text{dom}(c) = \text{dom}(a) \cap \text{dom}(b)$ and, for each $n \in \text{dom}(c)$, we have $c(n) = \max(a(n), b(n))$. Let $h : S(\mathbb{N}, V) \times S(\mathbb{N}, V_\alpha) \rightarrow S(\mathbb{N}, V)$ such that

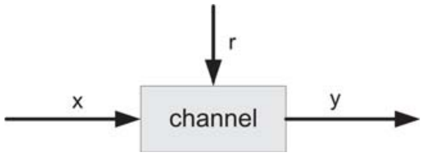

Figure 2: A Kahn FIFO channel with data and request inputs

Table 3: Sequence of events for a Kahn channel

Channel name	**Sequence of events**
x	$((1,a),(5,b),(19,c),(20,d))$
r	$((0,\alpha),(6,\alpha),(12,\alpha),(18,\alpha))$
y	$((1,a),(6,b),(19,c),(20,d))$

$h(a,b) = a$ for each $(a,b) \in S(\mathbb{N},V) \times S(\mathbb{N},V_\alpha)$. Thus, a Kahn FIFO also meets conditions (ii) and (iv) of Definition 4.15 with the defined g and h. Let $a, b \in S(\mathbb{N},T)$ be two strictly increasing signal and let $c = g(a,b)$. For each $i, j \in \text{dom}(c)$ such that $i < j$, we have $x(i) < x(j)$ and $y(i) < y(j)$, as well as $x(j) \leq z(j)$ and $y(j) \leq z(j)$. By transitivity, we have $x(i) < z(j)$ and $y(i) < z(j)$. Because $z(i)$ is equal to $x(i)$ or $y(i)$, we have $z(i) < z(j)$. A Kahn FIFO thus meets condition (iii) of Definition 4.15 and is a sequence-based function.

Since h is trivially monotonic, a Kahn FIFO meets condition (i) of Theorem 4.16. Let $c = g(a,b)$ and $z = g(x,y)$ such that $a \sqsubseteq x$ and $b \sqsubseteq y$. Then $\text{dom}(a) \subseteq \text{dom}(x)$ and $\text{dom}(b) \subseteq \text{dom}(y)$ imply $\text{dom}(c) \subseteq \text{dom}(z)$. For each $j \in \text{dom}(c)$, we have $a(j) = x(j)$ and $b(j) = y(j)$ which leads to $c(j) = z(j)$ and therefore $c \sqsubseteq z$. Thus, g is monotonic and it follows immediately that a Kahn FIFO meets condition (ii.b) of Theorem 4.16. Let $y_1 = kahn(x_1, r_1)$ and $y_2 = kahn(x_2, r_2)$ such that $(x_1, r_1) \sqsubseteq (x_2, r_2)$. Assume there exists $t \in T$ such that $t \in \tau(y_2)$ and $t \notin \tau(y_1)$. Because $\tau(y_1) \sqsubseteq \tau(y_2)$, there exists $n \in \text{dom}(\tau(y_2))$ such that $n \notin \text{dom}(\tau(y_1))$ and $\tau(y_2)(n) = t$. Thus, we have first that $n \in \text{dom}(\tau(x_2))$ and $n \in \text{dom}(\tau(r_2))$, and second that $n \notin \text{dom}(\tau(x_1))$ or $n \notin \text{dom}(\tau(r_1))$. Therefore, $t \notin \text{dom}(x_1)$ or $t \notin \text{dom}(r_1)$ which leads to $t \notin \text{dom}(y_1)$. A Kahn FIFO thus meets condition (ii.a) of Theorem 4.16 and it is monotonic, exactly causal and Scott-continuous.

Example 4.18. An empty detector is a channel with a data input signal (x) and an explicit request signal (r). As shown in Table 4, an empty detector writes the special value α to the output signal (p) every time a read request

Table 4: Sequence of events for an empty detector

Channel name	**Sequence of events**
x	$((1, a), (5, b), (19, c), (20, d))$
r	$((0, \alpha), (6, \alpha), (12, \alpha), (18, \alpha))$
$p = \mathit{empty}(x, r)$	$((0, \alpha), (18, \alpha))$

is made when the channel is empty. For example, if a read request is made on an empty detector such that $\tau(r)(0) < \tau(x)(0)$, or if $0 \notin \mathrm{dom}(\tau(x))$, then $\tau(p)(0) = \tau(r)(0)$ is the time of the first request made on an empty channel. If $\tau(r)(1) < \tau(x)(0)$, then $\tau(p)(1) = \tau(r)(1)$ is the time of the second failed request, and so on and so forth.

Formally, denote the set of all subsets of T (the *power set* of T) by $\mathcal{P}(T)$. Let $G : S(\mathbb{N}, T)^2 \to S(\mathbb{N}, \mathcal{P}(T))$ such that, for each $E = G(\tau_x, \tau_r)$ with $\tau_x, \tau_r \in S(\mathbb{N}, T)$ and for each $t \in T$, we have $t \in E(n)$ if there exists $j \in \mathrm{dom}(\tau_r)$ such that (i)$t = \tau_r(j)$; (ii)$j \geq n$; (iii) $t < \tau_x(j - n)$ or $(j - n) \notin \mathrm{dom}(\tau_x)$. Also denote the set of all partial functions from $\mathbb{N}$ to t by $[\mathbb{N} \to t]$ and let $g : S(\mathbb{N}, T)^2 \to [\mathbb{N} \to t]$ such that, for each $e = g(\tau_x, \tau_r)$ with $\tau_x, \tau_r \in S(\mathbb{N}, T)$ and $n \in \mathbb{N}$, we have $n \in \mathrm{dom}(e)$ if $E(n) \neq \emptyset$ and $e(n) = \min(E(n))$, where $E = G(\tau_x, \tau_r)$.

Then, an empty detector is a function $f : S_d(T, V) \times S_d(T, V_\alpha) \to S_d(T, V_\alpha)$ with $V_\alpha = \{\alpha, \varepsilon\}$ such that, for each $y = \mathit{empty}(x, r)$, we have:

(i) $\mathrm{dom}(y) = \mathrm{dom}(x) \cap \mathrm{dom}(r)$;

(ii) $y(t) = \begin{cases} \alpha & \text{if } t \in g(\tau(x), \tau(r)) \\ \varepsilon & \text{otherwise} \end{cases}$

By (i), an empty detector meets condition (i) of Definition 4.15. Let $h : S(\mathbb{N}, V) \times S(\mathbb{N}, V_\alpha) \to S(\mathbb{N}, V_\alpha)$ such that, for each $c = h(a, b)$ with $(a, b) \in S(\mathbb{N}, V) \times S(\mathbb{N}, V_\alpha)$, we have $\mathit{dom}(c) = \mathbb{N}$ and $c(n) = \alpha$ for each $n \in \mathbb{N}$. Then, if g is closed on strictly increasing signals, an empty detector meets conditions (ii), (iii) and (iv) of Definition 4.15. We show by induction that g is closed on strictly increasing signals.

Let $E = G(\tau_x, \tau_r)$ and $e = g(\tau_x, \tau_r)$ with two strictly increasing signals $\tau_x, \tau_r \in S(\mathbb{N}, T)$. If $\mathrm{dom}(e) = \emptyset$ or $\mathrm{dom}(e) = \{0\}$, then $e \in S(\mathbb{N}, T)$ is a strictly increasing signal. If not, then let $i \in \mathrm{dom}(e)$ such that $i \geq 1$. There exists $k \in \mathrm{dom}(\tau_r)$ such that $e(i) = \tau_r(k)$ with $\tau_r(k) < \tau_x(k - i)$ or $(k - i) \notin \mathrm{dom}(\tau_x)$. In the first case, we have $\tau_r(k) < \tau_x(k - i)$, thus

$\tau_r(k-1) < \tau_x(k-i)$ and $\tau_r(k-1) \in E(i-1)$. In the second case, we have $(k-i) \notin \mathrm{dom}(\tau_x)$ and thus $(k-1)-(i-1) \notin \mathrm{dom}(\tau_x)$ and $\tau_r(k-1) \in E(i-1)$. In both cases, this implies both that $(i-1) \in \mathrm{dom}(e)$ and $e(i-1) < e(i)$. By induction, if e is defined on any $i \in \mathbb{N}$, then it is defined on any $n \leq i$ and $e(i) < e(j)$ for each $i, j \in \mathrm{dom}(e)$ such that $i < j$. Thus, e is defined on a down set of $\mathbb{N}$ and is a strictly increasing signal. An empty detector is therefore a sequence-based function.

Since h is trivially monotonic, an empty detector meets condition (i) of Theorem 4.16. Let $y_1 = \mathit{empty}(x_1, r_1)$ and $y_2 = \mathit{empty}(x_2, r_2)$ such that $x_1 \sqsubseteq x_2$ and $r_1 \sqsubseteq r_2$. Let $e_1 = g(\tau_{x1}, \tau_{r1})$, $e_2 = g(\tau_{x2}, \tau_{r2})$, $E_1 = G(\tau_{x1}, \tau_{r1})$ and $E_2 = G(\tau_{x2}, \tau_{r2})$ such that $\tau_{x1} = \tau(x_1)$, $\tau_{x2} = \tau(x_2)$, $\tau_{r1} = \tau(r_1)$ and $\tau_{r2} = \tau(r_2)$. We show that for each $t \in T$ such that $t \in e_2$ and $t \notin e_1$, we have $t \notin \mathrm{dom}(y_1)$. Let $t \in T$ such that $t \in e_2$ and $t \notin e_1$. Then there exists $n \in \mathrm{dom}(e_2)$ and $k \in \mathrm{dom}(\tau_{r2})$ such that $t = e_2(n)$ and $t = \tau_{r2}(k)$. Thus we have $\tau_{r2}(k) < \tau_{x2}(k-n)$ or $(k-n) \notin \mathrm{dom}(\tau_{x2})$.

Because τ_{r1} is strictly increasing and $\tau_{r1} \sqsubseteq \tau_{r2}$, then we have $t = \tau_{r1}(k)$ if $k \in \mathrm{dom}(\tau_{r1})$ and $t \notin \mathrm{dom}(r_1)$ if not. If $t \notin \mathrm{dom}(r_1)$, then $t \notin \mathrm{dom}(y_1)$, which meets condition (ii.a) of Theorem 4.16. If $t = \tau_{r1}(k)$ and $\tau_{r2}(k) < \tau_{x2}(k-n)$, then we have $\tau_{r1}(k) < \tau_{x1}(k-n)$ or $(k-n) \notin \mathrm{dom}(\tau_{x1})$ and thus $t \in E_1(n)$. If $t = \tau_{r1}(k)$ and $(k-n) \notin \mathrm{dom}(\tau_{x2})$, then $(k-n) \notin \mathrm{dom}(\tau_{x1})$ and $t \in E_1(n)$. In both cases, since $t \notin e_1$ and $t = e_2(n)$, there exists $i \in \mathrm{dom}(\tau_{r1})$ with $i < k$ such that $(i-n) \notin \mathrm{dom}(\tau_{x1})$ and $\tau_{x2}(i-n) \leq \tau_{r2}(i)$. Thus, $\tau_{x2}(i-n) \notin \mathrm{dom}(x_1)$. Because $\tau_{x2}(i-n) < \tau_{r2}(k)$, we have $t \notin \mathrm{dom}(x_1)$ and thus $t \notin \mathrm{dom}(y_1)$. An empty detector thus meets condition (ii.a) of Theorem 4.16.

We next show that condition (ii.b) of Theorem 4.16 holds. Let $t \in T$ such that $t \in e_1$ and $t \in \mathrm{dom}(y_1)$. Then there exists $n \in \mathrm{dom}(e_1)$ and $k \in \mathrm{dom}(\tau_{r1})$ such that $t = e_1(n)$ and $t = \tau_{r1}(k)$. Thus we have $\tau_{r1}(k) < \tau_{x1}(k-n)$ or $(k-n) \notin \mathrm{dom}(\tau_{x1})$ and, for each $i \in \mathrm{dom}(\tau_{r1})$ such that $i < k$, we have $(i-n) \in \mathrm{dom}(\tau_{x1})$ and $\tau_{x1}(i-n) \leq \tau_{r1}(i)$. If $\tau_{r1}(k) < \tau_{x1}(k-n)$, we thus have that $\tau_{r2}(k) < \tau_{x2}(k-n)$ and $e_2(n) = t$. If $(k-n) \notin \mathrm{dom}(\tau_{x1})$ and $(k-n) \notin \mathrm{dom}(\tau_{x2})$, then we also have $e_2(n) = t$. If $(k-n) \notin \mathrm{dom}(\tau_{x1})$ and $(k-n) \in \mathrm{dom}(\tau_{x2})$, then $\tau_{x2}(k-n) \notin \mathrm{dom}(x_1)$ and thus $\tau_{x2}(k-n) \notin \mathrm{dom}(y_1)$. Because $t \in \mathrm{dom}(y_1)$, we have that $t < \tau_{x2}(k-n)$ and thus $\tau_{r2}(k) < \tau_{x2}(k-n)$ and $t = e_2(n)$. An empty detector thus meets all conditions of Theorem 4.16 and is monotonic, exactly causal and Scott-continuous.

5. Domain-warping processes

In the previous section, we considered processes whose output signals are defined only for all tags where all input signals are defined. The output of these processes thus preserves the domain of their inputs. In this section, we rather consider domain-warping processes, whose output signal generally does not have the same domain as their input signals. Such domain-warping processes can in particular be used to model delays.

The domain of a signal $s \in S(T, V)$ is a down of set of T. Let $\mathcal{D}(T)$ denote the set of all down sets on T. Then we have $\mathrm{dom}(s) \in \mathcal{D}(T)$ for each $s \in S(T, V)$. It has been shown [5] that:

(i) $\mathcal{D}(T)$ is closed under union and intersection;

(ii) $\mathcal{D}(T)$ ordered by set inclusion $\subseteq$ is a complete partial order;

(iii) For each directed subset $D \subseteq \mathcal{D}(T)$, we have $\bigsqcup D = \bigcup\limits_{d \in D} d$

Functions of the form $h : \mathcal{D}(T) \to \mathcal{D}(T)$ are used to warp the domain of a signal. As usual, h is monotonic if, for each $r, s \in \mathcal{D}(T)$ such that $r \subseteq s$, we have $h(r) \subseteq h(s)$. Also h is Scott-continuous if, for each directed subset $D \subseteq \mathcal{D}(T)$, we have $h(\bigsqcup D) = \bigsqcup h(D)$ and thus $h\left(\bigcup\limits_{d \in D} d\right) = \bigcup\limits_{d \in D} h(d)$.

Both continuous-time and discrete-event domain-warping processes are considered in the following subsections.

5.1. Continuous-time domain-warping processes

The principle behind continuous-time domain-warping processes is simple: the value of the input signal for tag $t \in T$ becomes the value of the output signal for tag $g(t)$, if $g(t)$ is in the domain of the output signal, where $g : T \to T$. Of course, g needs to be injective so that each tag of the output signal takes at most one value from the input signal. Formally, we have:

Definition 5.1 (Left inverse)**.** Let X, Y be two sets and let $f : X \to Y$ be an injective function. The *left inverse* of f is a partial function $f^{-1} : Y \to X$ with domain of definition $\{f(x) | x \in X\}$ such that $f^{-1}(f(x)) = x$ for each $x \in X$.

Definition 5.2 (Continuous-time domain-warping function)**.** A *continuous-time domain-warping function* is a function $f : S(T, V) \rightarrow S(T, V)$ with $g : T \rightarrow T$ and $h : \mathcal{D}(T) \rightarrow \mathcal{D}(T)$ such that g is injective, g^{-1} is the left inverse of g and, for each $x \in S(T, V)$ and $y = f(x)$, we have:

(i) $\mathrm{dom}(y) = h(\mathrm{dom}(x))$;

(ii) $y(t) = \begin{cases} x(g^{-1}(t)) & \text{if } t \in \mathrm{dom}(g^{-1}) \text{ and } g^{-1}(t) \in \mathrm{dom}(x) \\ \varepsilon & \text{otherwise} \end{cases}$

Lemma 5.3. *A continuous-time domain-warping function $f : S(T, V) \rightarrow S(T, V)$ with $g : T \rightarrow T$ and $h : \mathcal{D}(T) \rightarrow \mathcal{D}(T)$ is monotonic if the following conditions hold:*

(i) h is monotonic;

(ii) For each $s \in S(T, V)$ and for each $t \in h(\mathrm{dom}(s))$, there exists $t' \in \mathrm{dom}(s)$ such that $g^{-1}(t) \leq t'$.

Proof. Let $r, s \in S(T, V)$ with $x = f(r)$ and $y = f(s)$. If $r \sqsubseteq s$, then $\mathrm{dom}(r) \subseteq \mathrm{dom}(s)$ and, by condition (i), we have $\mathrm{dom}(x) \subseteq \mathrm{dom}(y)$.

For each, $t \in \mathrm{dom}(x)$, we have:

$$x(t) = \begin{cases} r(g^{-1}(t)) & \text{if } t \in \mathrm{dom}(g^{-1}) \text{ and } g^{-1}(t) \in \mathrm{dom}(r) \\ \varepsilon & \text{otherwise} \end{cases} \quad \textit{and}$$

$$y(t) = \begin{cases} s(g^{-1}(t)) & \text{if } t \in \mathrm{dom}(g^{-1}) \text{ and } g^{-1}(t) \in \mathrm{dom}(s) \\ \varepsilon & \text{otherwise} \end{cases}$$

By condition (ii), for each $t \in \mathrm{dom}(x)$ such that $t \in \mathrm{dom}(g^{-1})$, there exists $t' \in \mathrm{dom}(r)$ such that $g^{-1}(t) \leq t'$ and we thus have $g^{-1}(t) \in \mathrm{dom}(r)$ and $g^{-1}(t) \in \mathrm{dom}(s)$. Because $r \sqsubseteq s$, we have $x(t) = y(t)$ for each $t \in \mathrm{dom}(x)$. Therefore, $x \sqsubseteq y$ and f is thus monotonic. □

Theorem 5.4. *A continuous-time domain-warping function $f : S(T, V) \rightarrow S(T, V)$ with $g : T \rightarrow T$ and $h : \mathcal{D}(T) \rightarrow \mathcal{D}(T)$ is Scott-continuous if the following conditions hold:*

(i) h is Scott-continuous;

(ii) For each $s \in S(T, V)$ and for each $t \in h(\mathrm{dom}(s))$, there exists $t' \in \mathrm{dom}(s)$ such that $g^{-1}(t) \leq t'$.

Proof. By defintion, h is monotonic and f is thus monotonic by Lemma 5.3. Let $D \subseteq S(T,V)$ be a directed subset. Then we have that $\text{dom}(f(\bigsqcup D)) = h(\text{dom}(\bigsqcup D))$. By Lemma 4.2, this is equal to $h\left(\bigcup_{d \in D} \text{dom}\,(d)\right)$ and, because h is Scott-continuous, to $\bigcup_{d \in D} h\,(\text{dom}(d))$. By Defintion 5.2, this gives $\bigcup_{d \in D} \text{dom}(f(d))$ and, by Corollary 4.3, $\text{dom}(\bigsqcup f(D))$. Therefore, $\text{dom}(f(\bigsqcup D)) = \text{dom}(\bigsqcup f(D))$ and f is Scott-continuous by Corollary 3.5. □

Example 5.5. Two examples of continuous-time domain-warping processes are the lookahead and delay processes. These processes respectively shift to the past and to the future every event in their input signal by a fixed tag amount [5, 6]. Formally, let $(T^{'}, \leq)$ be a totally ordered tag superset and let $+$ be a binary operation over $T^{'}$ such that $(T^{'}, +)$ is a group (i.e. $+$ is associative, there exists an identy element $0 \in T^{'}$ and, for each $t \in T^{'}$, there exists an inverse element $-t$). Let $T \subseteq T^{'}$ be an interval, meaning that, for each $a, b, c \in T^{'}$, if $a, c \in T$ with $a \leq b$ and $b \leq c$, then we also have $b \in T$.

For $a \in T$ such that $a \geq 0$, a lookahead process is a function $f : S(T,V) \to S(T,V)$ such that, for each $y = f(x)$ with $x \in S(T,V)$, we have:

(i) $\text{dom}(y) = \{t \in T | (t + a) \in \text{dom}(x)\}$;

(ii) y(t) = x(t+a)

Let $g : T \to T$ such that $g(t) = t - a$ and $g^{-1}(t) = t + a$, and let $h : \mathcal{D}(T) \to \mathcal{D}(T)$ such that $h(d) = \{t \in T | (t + a) \in d\}$. A lookahead is thus a continuous-time domain-warping process by Definition 5.2. We have that $h\left(\bigcup_{d \in D} d\right) = \{t \in T | (t + a) \in \bigcup_{d \in D} d\}$ and, by distributivity, this equals $\bigcup_{d \in D} \{t \in T | (t + a) \in d\}$. This is equal to $\bigcup_{d \in D} h(d)$: h is thus Scott-continuous and a lookahead meets condition (i) of Theorem 5.4. Let $g^{-1}(h(\text{dom}(x))) = \{g^{-1}(t) | t \in \ h(\text{dom}(x))\}$. This is equal to $\{t + a | t \in h(\text{dom}(x))\}$, which gives $\{t + a \in T | (t + a) \in \text{dom}(x)\} = \text{dom}(x)$. Therefore, $g^{-1}(h(\text{dom}(x))) \subseteq \text{dom}(x)$ and a lookahead meets condition (ii) of Theorem 5.4. A lookahead is thus monotonic and Scott-continuous.

A delay is defined similarly, for $a \in T$ such that $a \geq 0$, as a function $f : S(T,V) \to S(T,V)$ such that, for each $y = f(x)$ with $x \in S(T,V)$, we have:

(i) $\mathrm{dom}(y) = \{t \in T | (t-a) \in \mathrm{dom}(x) \text{ or } (t-a) \notin T\}$;

(ii) $y(t) = \begin{cases} x(t-a) & \text{if } (t-a) \in T \\ \varepsilon & \text{otherwise} \end{cases}$

A delay is a continuous-time domain-warping process and it can be shown, with a proof similar to the one for the lookahead, that it is monotonic and Scott-continuous.

5.2. Discrete-event domain-warping processes

A discrete-event domain-warping process is similar to a sequence-based process, except that the domain of the output signal is not necessarily the domain of the input signal and that only the tags but not the values of present events may be changed by the process. Formally, we have:

Definition 5.6 (Injective signal)**.** A signal $s \in S(T,V)$ is *injective* if, for each $t_1, t_2 \in T$ such that $t_1 \neq t_2$, we have $s(t_1) \neq s(t_2)$.

Definition 5.7 (Discrete-event domain-warping function)**.** A *discrete-event domain-warping function* is a function $f : S_d(T,V) \to S_d(T,V)$ with $g : S(\mathbb{N},T) \to S(\mathbb{N},T)$ and $h : \mathcal{D}(T) \to \mathcal{D}(T)$ such that, for each $x \in S_d(T,V)$ and $y = f(x)$, we have:

(i) $\mathrm{dom}(y) = h(\mathrm{dom}(x))$;

(ii) $y(t) = \begin{cases} \nu(x)(n) & \text{if } \exists n \in \mathbb{N} \text{ such that } g(\tau(x))(n) = t \\ \varepsilon & \text{otherwise} \end{cases}$

(iii) g is closed on injective signals;

(iv) $\mathrm{dom}(g(\tau(s))) \subseteq \mathrm{dom}(\tau(s))$ for each $s \in S(T,V)$.

For each discrete-event signal $s \in S_d(T,V)$, we have that $\tau(s)$ is an injective signal. Condition (iii) ensures that $g(\tau(s))$ is also an injective signal and thus that two different tags in the input signal are never mapped to the same tag in the output signal.

Lemma 5.8. *A discrete-event domain-warping function $f : S_d(T,V) \to S_d(T,V)$ with $g : S(\mathbb{N},T) \to S(\mathbb{N},T)$ and $h : \mathcal{D}(T) \to \mathcal{D}(T)$ is monotonic if the following conditions hold:*

(i) h is monotonic;

(ii) *g is closed on strictly increasing signals;*

(iii) *For each $r, s \in S_d(T, V)$ such that $r \sqsubseteq s$, we have:*

> *(a)* *$t \notin g(\tau(r))$ and $t \in h(\mathrm{dom}\,(r)) \Rightarrow t \notin g(\tau(s))$;*
>
> *(b)* *$t \in g(\tau(r))$ and $t \in h(\mathrm{dom}\,(r)) \Rightarrow t \in g(\tau(s))$;*

Proof. Let $x = f(r)$ and $y = f(s)$. If $r \sqsubseteq s$, then $\mathrm{dom}(r) \subseteq \mathrm{dom}(s)$ and, by condition (i), we have $\mathrm{dom}(x) \subseteq \mathrm{dom}(y)$. With conditions (ii) and (iii), we can then use the same procedure than in the proof of Theorem 4.16 to prove that $x(t) = y(t)$ for each $t \in \mathrm{dom}(x)$ and thus $x \sqsubseteq y$. □

Theorem 5.9. *A discrete-event domain-warping function $f : S_d(T, V) \rightarrow S_d(T, V)$ with $g : S(\mathbb{N}, T) \rightarrow S(\mathbb{N}, T)$ and $h : \mathcal{D}(T) \rightarrow \mathcal{D}(T)$ is Scott-continuous if the following conditions hold:*

(i) *h is Scott-continuous;*

(ii) *g is closed on strictly increasing signals;*

(iii) *For each $r, s \in S_d(T, V)$ such that $r \sqsubseteq s$, we have:*

> *(a)* *$t \notin g(\tau(r))$ and $t \in h(\mathrm{dom}\,(r)) \Rightarrow t \notin g(\tau(s))$;*
>
> *(b)* *$t \in g(\tau(r))$ and $t \in h(\mathrm{dom}\,(r)) \Rightarrow t \in g(\tau(s))$;*

Proof. It follows immediately that h is monotonic and, by Lemma 5.8 with conditions (ii) and (iii), f is also monotonic. With condition (i), we can then use the same procedure than in the proof of Theorem 5.4 to prove that $\mathrm{dom}(f(\bigsqcup D)) = \mathrm{dom}(\bigsqcup f(D))$ and that f is Scott-continuous by Corollary 3.5. □

6. Composite processes

In the previous sections, we proved the Scott-continuity of exactly causal processes and, under certain conditions, of domain-warping processes. Following the Kahn principle, if every process in a process network is Scott-continuous, then the process network is itself a Scott-continuous function that maps input signals to the least fixed point of the network equations. Therefore, a process network built from the Scott-continuous processes defined in the previous sections is also a Scott-continuous process, even if the resulting composite process is neither an exactly causal process nor a domain-warping process.

Example 6.1. A poll FIFO is a FIFO with non-blocking reads. It has a data input signal (x) and a data output signal (y), as well as an explicit request signal (r). If it receives a request while it is empty, it immediately signals the failure to read by sending special value α on its output and it then discards the request. If it receives a request while it is not empty, it sends to the output the earliest (not yet sent to the output) value received on its input. A poll FIFO is the composition of several simpler processes, as shown in Figure 3. The behavior of the poll channel is shown in Table 5 with the same inputs than for the Kahn FIFO.

Table 5: Sequence of events for a poll FIFO

Channel name	**Sequence of events**
x	$((1,a),(5,b),(19,c),(20,d))$
r	$((0,\alpha),(6,\alpha),(12,\alpha),(18,\alpha))$
$p = empty(x,r)$	$((0,\alpha),(18,\alpha))$
$d = dec(r,p)$	$((6,\alpha),(12,\alpha))$
$v = kahn(x,d)$	$((6,a),(12,b))$
$y = merge(v,p)$	$((0,\alpha),(6,a),(12,b),(18,\alpha))$

An empty detector first detects read requests that are made when the FIFO is empty. A decimator then produces a modified request signal in which all requests on an empty FIFO have been removed. This modified request signal controls an inner Kahn FIFO. The output of the Kahn FIFO, which contains the result of all successful read requests, is merged with the output of the empty detector, which contains failed read requests.

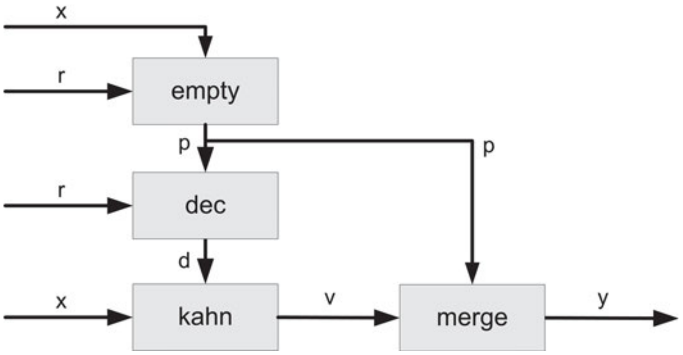

Figure 3: Schema of a poll FIFO

The Scott-continuity of the merge and decimator processes has been shown in Example 4.9 and Example 4.11 whereas the empty detector has been shown to be Scott-continuous in Example 4.18. A poll FIFO is thus a process network in which every process is Scott-continuous and it is therefore a Scott-continuous process.

7. Conclusion

We have demonstrated a general theorem on the necessary and sufficient conditions for the Scott-continuity of tagged signal processes and we have applied it to show the Scott-continuity of broad classes of processes. Thus, we have shown the Scott-continuity of exactly causal processes, which include stateless processes, such as merge and decimator processes, as well as some discrete-event processes, such as a Kahn FIFO and empty detectors. The Scott-continuity, under certain conditions, of domain-warping processes such as delays and lookaheds has also been shown. Finally, we have shown how composite processes can be composed from simpler processes.

With the proofs and theorems given in this paper, it becomes easier to show that a given tagged signal process is Scott-continuous and therefore to build compositional models of computation within the tagged signal model. For example, these theorems can be used to model explicit communication channels in real-time process networks and Scott-continuity guarantees that the resulting process networks are compositional. Future work could also involve developing proofs of Scott-continuity for other classes of processes.

Acknowledgments

The authors thank the Natural Sciences and Engineering Research Council of Canada for its financial help.

References

[1] S. Abramsky and A. Jung, Domain theory, in: S. Abramsky, D. M. Gabbay, and T. S. E. Maibaum (Eds.), Handbook of Logic in Computer Science (vol. 3): Semantic Structures, Oxford University Press, Oxford, UK, 1995, pp. 1–168.

[2] Gilles Kahn, The semantics of a simple language for parallel programming, in: J.L. Rosenfeld (Ed.), Proceedings of the IFIP Congress 74, North-Holland Publishing, 1974, pp. 471–75.

[3] E. A. Lee, Modeling concurrent real-time processes using discrete events, Annals of Software Engineering 7(1-4):25–45, 1999.

[4] E. A. Lee and A. Sangiovanni-Vincentelli, A framework for comparing models of computation, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems 17(12):1217–29, 1998.

[5] X. Liu, Semantic Foundation of the Tagged Signal Model, Ph.D. Thesis, EECS Department, University of California at Berkeley, 2005.

[6] X. Liu and E. A. Lee, CPO semantics of timed interactive actor networks, Theoretical Computer Science 409(1):110–125, 2008.

[7] Prakash Panangaden and Eugene W. Stark. Computations, residuals, and the power of indeterminacy, in: Timo Lepist and Arto Salomaa (Eds.), Proceedings of the 15th International Colloquium on Automata, Languages and Programming, Lecture Notes in Computer Science (vol. 317), Springer-Verlag, London, 1988, pp. 439–454.

[8] R. K. Yates. Networks of real-time processes, in: E. Best (Ed.), Proceedings of the 4th International Conference on Concurrency Theory, Lecture Notes in Computer Science (vol. 715), Springer-Verlag, Germany, 1993, pp. 384–97.

A. Appendix: Additional proofs

A.1. Order theory

Lemma A.1. *Let $(X, \leq)$ and $(Y, \leq)$ be complete partial orders. If a function $f : X \to Y$ is monotonic, then for each directed subset $D \subseteq X$:*

(i) $f(D) = \{f\,(d)\,|d \in D\}$ *is a directed subset of* Y.

(ii) $\bigsqcup f(D) \leq f\,(\bigsqcup D)$.

Proof. For each directed subset $D \subseteq X$ and for each $a, b \in D$, there exists by definition $c \in D$ such that $a \leq c$ and $b \leq c$. If f is monotonic, then we have $f(a) \leq f(c)$ and $f(b) \leq f(c)$ such that $f(a), f(b), f(c) \in f(D)$. Thus, $f(D)$ is a directed subset of Y. There exists $\bigsqcup D \in X$ and $\bigsqcup f(D) \in Y$ because $(X, \leq)$ and $(Y, \leq)$ are complete partial orders. By definition, we have $d \leq \bigsqcup D$ for each $d \in D$ and, because f is monotonic, we have $f(d) \leq f\left(\bigsqcup D\right)$ for each $f(d) \in f(D)$. Thus, $f\left(\bigsqcup D\right)$ is an upper bound of $f(D)$. By definition, $\bigsqcup f(D)$ is the least upper bound of $f(D)$ and we thus have $\bigsqcup f(D) \leq f\left(\bigsqcup D\right)$. □

Lemma A.2. *Let $(X, \leq)$ and $(Y, \leq)$ be complete partial orders. A function $f : X \to Y$ is Scott-continuous if and only if the following conditions hold:*

(i) f is monotonic.

(ii) $f\left(\bigsqcup D\right) \leq \bigsqcup f(D)$ for each directed subset $D \subseteq X$.

Proof. If (i) does not hold, then there exists $r, s \in X$ such that $r \leq s$ but $f(r) \not\leq f(s)$. We have a directed subset $D = \{r, s\}$ such that $\bigsqcup D = s$ and therefore $f\left(\bigsqcup D\right) = f(s)$. However $\bigsqcup f(D) \neq f(s)$ because $f(r) \not\leq f(s)$. Therefore, f is not Scott-continuous.

If (ii) does not hold, then there exists a directed subset D such that $f\left(\bigsqcup D\right) \not\leq \bigsqcup f(D)$. Because of the reflexivity of $\leq$, we have that $f\left(\bigsqcup D\right) \neq \bigsqcup f(D)$ and f is not Scott-continuous. Therefore, f is Scott-continuous only if both conditions (i) and (ii) hold. We next show that if both conditions (i) and (ii) hold, then f is Scott-continuous.

If condition (i) holds, then by Lemma A.1, for each directed subset $D \subseteq X$, we have $\bigsqcup f(D) \leq f\left(\bigsqcup D\right)$. If condition (ii) also holds, then we have $f\left(\bigsqcup D\right) = \bigsqcup f(D)$ because of the antisymmetry of $\leq$ and f is thus Scott-continuous. □

A.2. Tagged signal model

Lemma A.3. *If $D \subseteq S(T, V)$ is a directed signal subset, then* $\mathrm{dom}\left(\bigsqcup D\right) = \bigcup\limits_{d \in D} \mathrm{dom}\,(d)$.

Proof. The proof of this Lemma is the same as the proof of Lemma 2.16 in [5], which showed that each signal set $S(T, V)$ is a complete partial order. It is given here again in order to make this paper self-contained.

Let $d_t \in D$ be any signal such that $t \in \mathrm{dom}(d_t)$ and let s be a signal such that $\mathrm{dom}(s) = \bigcup\limits_{d \in D} \mathrm{dom}\,(d)$ and $s(t) = d_t(t)$. By definition, d_t exists for each $t \in \mathrm{dom}(s)$. Also, since D is a directed set, every signal that could be d_t must have the same value at tag t, or else they would have no common upper bound and this would be a contradiction.

For each $d \in D$, we have that $\mathrm{dom}(d) \subseteq \mathrm{dom}(s)$ and $d(t) = s(t)$ for each $t \in \mathrm{dom}(d)$. The signal s is thus an upper bound of D. Let u be an upper bound of D. Then we have $\mathrm{dom}(d) \subseteq \mathrm{dom}(u)$ for each $d \in D$ and thus $\bigcup\limits_{d \in D} \mathrm{dom}\,(d) \subseteq \mathrm{dom}(u)$. This gives $\mathrm{dom}(s) \subseteq \mathrm{dom}(u)$. Furthermore, we have $d_t(t) = u(t)$ for each $t \in \mathrm{dom}(s)$ and thus $s(t) = u(t)$ for each $t \in dom(s)$. Therefore $s \sqsubseteq u$ and $s = \bigsqcup D$. This proves that $\mathrm{dom}\,(\bigsqcup D) = \bigcup\limits_{d \in D} \mathrm{dom}\,(d)$.

□

L'École Polytechnique se spécialise dans la formation d'ingénieurs et la recherche en ingénierie depuis 1873

École Polytechnique de Montréal

École affiliée à l'Université de Montréal

Campus de l'Université de Montréal
C.P. 6079, succ. Centre-ville
Montréal (Québec)
Canada H3C 3A7

www.polymtl.ca

ÉCOLE
POLYTECHNIQUE
MONTRÉAL