



HAL
open science

Spatio-temporal domains: an overview

David Janin

► **To cite this version:**

| David Janin. Spatio-temporal domains: an overview. 2017. hal-01634897v1

HAL Id: hal-01634897

<https://hal.science/hal-01634897v1>

Preprint submitted on 14 Nov 2017 (v1), last revised 17 Jul 2018 (v2)

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Spatio-temporal domains: an overview

David Janin*

UMR LaBRI, Bordeaux INP
Université de Bordeaux
janin@labri.fr

Abstract

We consider the possibility of defining a general mathematical framework for the homogeneous modeling of heterogeneous spatio-temporal computations as they more and more occur in modern computerized systems of systems. It appears that the notion of element categories of certain sheaves over cpos, called timed domains, and their associated transformations eventually provide a fully featured category for such a purpose that extends to space and time the category of cpos and continuous functions used in classical denotational semantics.

1 Introduction

Research context. Time is what is indicated by a clock. By such an aphorism, Einstein states that the proper time between two events is indicated by a clock present at both events. This implies that time is inherently linked with space. Though defined in physics, such a principle also applies to modern computerized systems: from networked multi-player entertainment applications to semi-automatic navigation systems or high performance computers.

In fact, modern systems are often composed of heterogeneous sub-systems which individual behaviors are based on events observed in some locally (and sometimes continuously) defined space and time. Much like in physics, there is no longer any global notion of time nor space in such a system of systems. Instead, subsystems interact through hybrid interfaces that translate between the local space and time information measured by each of these subsystems and maintain the global coherence of their local behavior. For instance, the notion of Globally-Asynchronous Locally-Synchronous (GALS) systems [10] that appears in the mid 80s already put the emphasis on the necessity of combining locally linear behaviors with event based, therefore generally non linear, causal dependencies.

In order to program these systems many proposals have been made. To name but a few, the development of the synchronous programming language family [31, 6, 4] is an example. The Functional Reactive Programming API [14, 15] with its numerous instances in typed functional programming languages is another example. In theoretical computer science, various mathematical frameworks have also been proposed for modeling the behavior of such complex evolving systems, ranging from domain theory [30, 22] to dataflow model of computations [21], ranging from generalized ultra-metric spaces [28, 25] to category theoretical models [8, 9], ranging from (timed) transducer theory [23] to co-algebraic constructions [3]. By targeting specific usage, such as timed programming with Khan synchronous networks [7] or concurrent programming with Winskel event structures [33], each of these approaches eventually lead to the definition of sound semantic models. However, no proposed mathematical framework yet seems general enough to model in a unified and homogeneous way the heterogeneity of modern computerized systems.

*Work partially supported by Inria center Bordeaux-Sud-Ouest, from September 2016 to February 2017.

Main result. Keeping a view on the inherent link between time and space mentioned above, we consider the possibility of defining such a uniform and homogeneous mathematical framework into which the spatio-temporal semantics of the behavior of heterogeneous systems of systems and the necessarily hybrid interfaces between their components can be modeled, analyzed and eventually certified. For such a purpose, we develop the notion of timed domains and timed morphisms that offer a versatile and sound mathematical framework for a timed denotational semantics where every non trivial computation step necessarily takes some non zero time. As a result, we obtain a somewhat fully featured extension to space and time of the category of cpos and continuous functions as used in classical denotational semantics.

Paper content. Timed posets are defined as certain posets indexed over time scales (Section 2). Various concepts deriving from such notions allow for relating timed posets and domains with, among others, generalized ultra-metric approaches followed in timed system semantics or event structures used in concurrent system semantics (Section 3). Timed morphisms are then defined as monotonic functions between timed posets that uniformly act on the underlying time scales (Section 4).

In the restricted synchronous case, over a given time scale, the resulting categories are shown to be topoi. Each associated categorical constructs can be interpreted in terms of known notions of *synchronous* transformations of timed resources (Section 5). In the general case of timed posets and timed morphisms the resulting categories are shown to be bi-cartesian closed. In this case, the associated categorical constructs are interpreted instead in terms of *asynchronous* transformations (Section 6).

Timed domain are eventually defined as certain continuous timed posets over complete time scale (Section 7). Inductive or co-inductive systems of timed domain fixpoint equations are shown to be solvable whenever their projections over time scales are (Sections 6 and 7). Least fixpoints computation of timed morphisms are also shown to be computable by means of least fixpoint operators which are, themselves, timed morphisms (Section 8).

The applicability of our proposal is illustrated throughout the paper by sketches of interpretation and application examples that are related to existing works.

Some notations. A partially ordered set, or *poset*, is a set P equipped with a partial order relation, that is, for all $x, y, z \in P$ we have $x \leq x$ (reflexivity), if both $x \leq y$ and $y \leq z$ then $x \leq z$ (transitivity) and if both $x \leq y$ and $y \leq x$ then $x = y$ (anti-symmetry). Given a poset P , we use the notation $x, y \leq z$ (resp. $z \leq x, y$) or even $z \leq X$ (resp. $x \leq X$) to denote the fact that both $x, y \in P$ are smaller (resp. greater) than $z \in P$, or the fact that $z \in P$ is an upper bound (resp. a lower bound) of the subset $X \subseteq P$.

The *downward closure* $\downarrow X \subseteq P$ of $X \subseteq P$ is defined by $\downarrow X = \{y \in P : \exists x \in X, y \leq x\}$. By extension, for every element $x \in P$ we also use the notation $\downarrow x = \{y \in P : y \leq x\}$. We say that X is *downward closed* when $\downarrow X = X$. When it exists, its *greatest lower bound* (resp. *least upper bound*) is denoted by $\bigwedge X$ (resp. $\bigvee X$). The poset P is a *meet-semilattice* when every two elements $x, y \in P$ have a meet, that is, a greatest lower bound denoted by $x \wedge y$.

A subset $X \subseteq P$ is *directed* when for all $x, y \in X$ there is $z \in X$ such that $x, y \leq z$. The poset P is a (directed) *complete partially ordered set*, or complete poset, or even *cpo*, when every directed subset $X \subseteq P$ has a least upper bound $\bigvee X$.

A function $f : P \rightarrow Q$ between two posets is *monotone* when $x \leq y$ implies $f(x) \leq f(y)$ for every $x, y \in X$. Whenever needed, the function f is extended point-wise to every subset $X \subseteq P$ by $f(X) = \{f(x) \in Q : x \in X\}$. The function f is *continuous* when, for every subset $X \subseteq P$, if X is directed and $\bigvee X$ is defined then $\bigvee f(X) = f(\bigvee X)$. A continuous function is necessarily monotone and, as such, maps every directed subsets to a directed subset.

In a cpo P with a least element \perp , we know from Knaster-Tarski that a homogeneous monotone function $f : P \rightarrow P$ has a least fixpoint $\mu(f) = \bigvee_{\lambda \in \Lambda} f^\lambda(\perp)$ with Λ any ordinal which cardinal is strictly greater than the cardinal of P . In the case f is continuous, we know from Scott that $\mu(f) = \bigvee_{n \in \omega} f^n(\perp)$.

Last, we shall consider various categorical concepts that are detailed when needed. We may also consider the categories: *Poset* with partially ordered sets and monotone functions, *SLattice* with meet-semilattices and meet-preserving functions and *Cpo* with complete posets and continuous functions, as well as their restriction to posets with a least element, the resulting (full) subcategories being denoted by $Poset_\perp$, $SLattice_\perp$ and Cpo_\perp . All these categories are known to be cartesian closed and, possibly extended with the empty set, bi-cartesian.

2 Timed posets

Throughout this text, we may interpret an arbitrary poset T as a *spatio-temporal scale*. This means that for every element $u, v \in T$, we may say that u lays *before* v when $u \leq v$, that u lays *after* v when $v \leq u$, or that u lays *aside* v when u and v are incomparable. In other words, there is the *time dimension* that increases with the order, and, by lack of a better name, there is the *space dimension* where incomparable elements are also related. Having said so, throughout the rest of this presentation, a spatio-temporal scale is simply be referred to as a *time scale*, its elements being called *instants*.

Definition 2.1 (Timed poset). Let T be a time scale. A poset *timed* over T is a poset P equipped with a *temporal projection* $\pi : P \rightarrow T$ such that:

(IN1) if $x \leq y$ then $\pi(x) \leq \pi(y)$, i.e. the temporal projection is monotone,

(IN2) for every $u \leq \pi(x)$ there is a unique element $x \downarrow u \in P$ such that $\pi(x \downarrow u) = u$ and $x \downarrow u \leq x$,

for all $x, y \in P$. The element $x \downarrow u$ is called the *temporal cut*, or simply the *cut* of x at instant u . This situation is depicted in Figure 1.

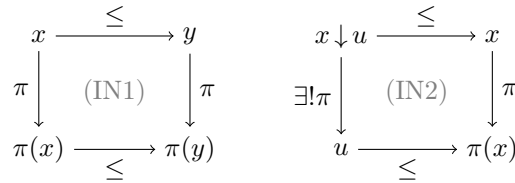


Figure 1: Temporal projection and temporal cut properties.

Remark 2.2. Property (IN1) states that π is a poset *functor*. With property (IN2) the projection π is even a poset *fibration*.

Example 2.3 (Self-timed poset). Every poset T is a timed poset over itself with identity id as temporal projection.

Example 2.4 (Sub-timed poset). Let P be a poset timed over T . Let $X \subseteq P$ be a downward closed subset of P . Let $inc_X : X \rightarrow P$ be the inclusion function. Then the set X ordered as in P with $\pi \circ inc_X$ as temporal projection is also a timed poset.

Example 2.5 (Timed impulses). Let T be a time scale and E be a set of event value extended into $E_\perp = E \cup \{\perp\}$. The set $Imp(T, E) = T \times E_\perp$ is turned into a timed poset by equipping it with the temporal projection and partial order defined by $\pi(u, e) = u$ and $(t, e) \leq (t', e')$ when either $(t, e) = (t', e')$ or $t < t'$ and $e = \perp$.

An element $(t, e) \in Imp(T, E)$ with $e \neq \perp$ is a *timed impulse* with value e at instant t . Observe there are no element strictly above (or after) such an impulse and all elements strictly below (or before) bears no value as they are of the form (t', \perp) for some $t' < t$.

Rephrased in terms of temporal cut, for all element $(t, e) \in Imp(T, E)$ and instant $t' \leq t = \pi(t, e)$, two cases are possible. Either $t' = t$ and we have $(t, e) \downarrow t' = (t, e)$, or $t' < t$ and we have $(t, e) \downarrow t' = (t', \perp)$.

Example 2.6 (Timed signals). Let again T be a time scale and E be the set of event values. A *timed signal* is defined as a relation $S \subseteq T \times E$ that maps every instant $t \in T$ to the set $S(t) = \{e \in E : (t, e) \in S\}$ of event values received at instant t . A *partial timed signal* is then a pair $(u, S) \in T \times \mathcal{P}(T \times E)$ such that for all $(t, e) \in S$ we have $t \leq u$.

The set $Sig(T, V)$ of partial timed signals is turned into a timed poset by equipping it with the temporal projection defined by $\pi(u, S) = u$ and the partial order relation defined by $(u, S) \leq (u', S')$ when $u \leq u'$ and $S(t) = S'(t)$ for all $t \leq u$, for all $(u, S), (u', S') \in Sig(T, V)$. Following [28], a pair $(u, S) \in Sig(T, V)$ represents the observation of a timed until the instant u . In some sense, it is the *temporal trace* of some computation until that instant.

Rephrased in terms of temporal cut, for all partial signal $(u, S) \in Sig(T, V)$ and instant $v \leq u = \pi(u, S)$ we have $(u, S) \downarrow v = (v, S')$ with $S' = \{(t, e) \in S : t \leq v\}$. Such a temporal cut $(u, S) \downarrow v$ models the observation of the partial signal (u, S) until v that occurs before u .

Definition 2.7 (Local cut). Let P be a poset timed over T . For every $x \in P$, property (IN2) induces a (total) function $cut_x : \downarrow \pi(x) \rightarrow P$, called a *local cut*, defined by $cut_x(u) = x \downarrow u$ over every $u \in T$ such that $u \leq \pi(x)$.

Lemma 2.8 (Local cut properties). *Let $x, y \in P$ and $u, v \in T$ such that $u, v \leq \pi(x)$. Then:*

- (1) $x \leq y$ if and only if $\pi(x) \leq \pi(y)$ and $x = y \downarrow \pi(x)$,
- (2) $u \leq v$ if and only if $x \downarrow u \leq x \downarrow v$.

In particular, the (sub)posets $\downarrow \pi(x) \subseteq T$ and $\downarrow x \subseteq P$ are isomorphic via π .

Proof. (1). Assume that $x \leq y$. By (IN1) we have $\pi(x) \leq \pi(y)$ hence by (IN2), $x = y \downarrow \pi(x)$ since $\pi(y \downarrow \pi(x)) = \pi(x)$. Conversely, assume that $\pi(x) \leq \pi(y)$ and $x = y \downarrow \pi(x)$. Then, by definition of local cut, we have $x \leq y$.

(2). Let $u, v \leq \pi(x)$. By (IN2) we have $\pi(x \downarrow u) = u$ and $\pi(x \downarrow v) = v$. Assuming that $u \leq v$ we thus have $u \leq \pi(x \downarrow v)$ hence there is $y = (x \downarrow v) \downarrow u$ such that $\pi(y) = u$ and $y \leq x \downarrow v$. But $x \downarrow v \leq x$ hence $y \leq x$ and thus, by unicity (IN2), we have $y = x \downarrow u$ therefore $x \downarrow u \leq x \downarrow v$. Conversely, assuming that $x \downarrow u \leq x \downarrow v$ by (IN1) we have $\pi(x \downarrow u) \leq \pi(x \downarrow v)$ hence $u \leq v$ since, by (IN2), we have $\pi(x \downarrow u) = u$ and $\pi(x \downarrow v) = v$. \square

Corollary 2.9. *An element $x \in P$ is minimal (resp. minimum) in P if and only if its temporal projection $\pi(x) \in T$ is minimal (resp. minimum) in T .*

On the contrary, maximal elements in a timed poset can have arbitrary temporal projection as shown by the impulse timed poset described in Example 2.5.

Remark 2.10. Observe that the notation \downarrow is used both for cuts or for downward closure. There is a potential risk of ambiguity or confusion.

However, one can first check that for all $x \in P$ and all $u \leq \pi(x)$ we have $\downarrow(x \downarrow u) = (\downarrow x) \downarrow u$ as soon as we extend the (partially defined) cut point-wise. It follows that we can even write $\downarrow x \downarrow u$ without parenthesis at all without any ambiguity.

Moreover, the downward closure can itself be seen as a cut over subsets of posets, defining for every $X \subseteq P$ and $x \in P$ the cut $X \downarrow x = \{y \in X : y \leq x\}$. In fact, a cut is sort of a (partial) action of a poset over set.

3 Derived notions

We review below several notions that derived from the notion of timed posets and that allows for relating this notion with other known concepts appearing in computer system modeling or programming language semantics.

Temporal coherence. The following notion is inspired by Girard notion of coherent space in linear logic [16]. A similar notion, even more closely related with timed domain, also appears in Winskel notion of event structures [33].

Definition 3.1. Let P be a poset timed over T . Let $x, y \in P$. We say that x and y are *coherent*, a property denoted by $x \circ y$, when, for every $x', y' \in P$ such that $x' \leq x$ and $y' \leq y$, if $\pi(x') = \pi(y')$ then $x' = y'$, or, equivalently, for all $u \leq \pi(x), \pi(y)$, $x \downarrow u = y \downarrow u$. By extension, a subset $X \subseteq P$ is a *coherent subset* when $x \circ y$ for all $x, y \in X$.

Lemma 3.2. *Let $x, y \in P$. Then $x \leq y$ if and only if $\pi(x) \leq \pi(y)$ and $x \circ y$.*

Proof. Assume that $x \leq y$ then we have $\pi(x) \leq \pi(y)$ hence $x = y \downarrow \pi(x)$. Let $u \leq \pi(x), \pi(y)$. We have $x \downarrow u \leq x \leq y$ hence, since $\varphi(x \downarrow u) = u$, by (IN2), we have $x \downarrow u = y \downarrow u$. This proves that $x \circ y$. Conversely, assume that $x \circ y$ and $\pi(x) \leq \pi(y)$. This implies that $x \downarrow \pi(x) = y \downarrow \pi(x)$ hence $x = y \downarrow \pi(x)$ and thus $x \leq y$. \square

In other words, coherence offer an alternative to the definition of timed poset via an order relation.

Remark 3.3. In some sense, every coherent subset $X \subseteq P$ can be sense as the *trace* of a spatio-temporal computation that has been observed at the instants $\pi(X)$. One can even check that the (sub)posets X and $\pi(X)$ are necessarily isomorphic via the restriction of π to X . Also, the downward closure $\downarrow X$ of a coherent subset is itself coherent.

Remark 3.4. Observe also that a timed poset with its induced coherence relation is a particular case of an event structure [33], though dropping the finite history requirement, and the same computational interpretation can be done.

Two elements x and y in P describe *concurrent* computations when they are coherent and incomparable: they can both appear in the same computation trace as suggested above. On the contrary, two elements x and y in P describe *conflicting* computations when there are incoherent: they cannot appear in the same computation trace.

To be more precise, by definition of coherence, when $x \not\leq y$ there is $u \in T$ in the past of both $\pi(x)$ and $\pi(y)$ such that $x \downarrow u \neq y \downarrow u$. This means that two distinct computations occur at the same instant (and position) u ; a case that is interpreted as forbidden/impossible.

Remark 3.5. Relating timed posets with event structures, one can also observe that the equivalence induced by π , defined by $x \sim_\pi y$ when $\pi(x) = \pi(y)$ for every $x, y \in P$ is almost¹ a *symmetry* as recently defined by Winskel in event structures [34]. This observation relates even more closely timed posets with the concepts introduced by Winskel in concurrency theory.

However, the notion of timed poset is more restrictive than the notion of event structure. In a timed poset, coherence is *uniformly* defined via temporal projection. In an event structure, coherence is given as part of the definition of that event structure.

Problem 3.6. The possible conditions under which sets equipped with a coherence relation can be viewed as timed posets is left as an open question.

Temporal distance. Timed concurrent system semantics can also be modeled by means of generalized ultrametric distance as already developed quite in the depth by Lee, Liu et Matsikoudis [28, 25]. The following definition shows that timed posets also induce such distances over their elements.

Definition 3.7. Let P be a poset timed over T . Let $\mathcal{P}^\downarrow(T)$ be the set of downward closed subsets of T ordered by reverse inclusion therefore with T as least element. The *distance* induced by π over P is the function $d : P \times P \rightarrow \mathcal{P}^\downarrow(T)$ defined by:

$$d(x, y) = \{t \in T : t \leq \pi(x), \pi(y), x \downarrow t = y \downarrow t\}$$

when $x \neq y$ and by $d(x, y) = T$ when $x = y$.

One can easily check that $d(x, y) = \downarrow d(x, y)$ for all $x, y \in P$ hence the above definition is sound.

Lemma 3.8. *The function $d : P \times P \rightarrow \mathcal{P}^\downarrow(T)$ is a generalized ultrametric, that is, we have:*

- (1) $d(x, y) = T$ if and only if $x = y$ (*separation*),
- (2) $d(x, y) = d(y, x)$ (*symmetry*),
- (3) $d(x, y) \supseteq d(x, z) \cap d(z, y)$ (*ultra-metric inequality*),

for all $x, y, z \in P$.

Proof. Properties (1) and (2) are immediate from the definition. Let $x, y, z \in P$ and let $u \in d(x, z) \cap d(z, y)$. This implies that $x \downarrow u = z \downarrow u$ with $u \leq \pi(x), \pi(z)$ and $z \downarrow u = y \downarrow u$ with $u \leq \pi(z), \pi(x)$ hence $x \downarrow u = y \downarrow u$ with $u \leq \pi(x), \pi(y)$ that is $u \in d(x, y)$. \square

Problem 3.9. Conversely, under which condition a generalized ultra-metric space can be turned into a timed poset is left as open question.

Temporal presheaves. One last connection of timed posets with an existing notion, now complete, goes via the notion of categorical presheaves, a notion already used in concurrency theory especially by Catani, Stark et Winskel for modeling process calculi like CCS and the π -calculus [9, 8].

Let T be a poset. A *presheaf* on T is a functor $F : T^{op} \rightarrow Set$ of the category obtained from T reversing the order relation into the category Set of sets and functions. Then the following

¹For \sim_π to be a symmetry, the timed poset P must be completed with sorts of non evolving elements of the form (u, x) with $x \leq (u, x)$ and $\pi(u, x) = u$, defined for all $x \in P$ and $u \in T$ such that there is no y above x such that $\pi(y) = u$.

lemmas state that poset timed over T are isomorphic with what are known in category theory as the *element categories* of presheaves over T . This shows that timed posets and presheaves are essentially equivalent notions.

Lemma 3.10. *Let $F : T^{op} \rightarrow Set$ be a presheaf over T . Then the set $P_F = \Sigma_{t \in T} F(t) = \{(t, x) : t \in T, x \in F(t)\}$, equipped with the temporal projection defined by $\pi(u, x) = u$ and the order relation defined by $(u, x) \leq (v, y)$ when $u \leq v$ and $F(u \leq v)(y) = x$ for all $(u, x), (v, y) \in P_F$, is a timed poset, also known as the element category of F .*

Proof. The fact P_F equipped with \leq is a poset goes as follows. Reflexivity follows from reflexivity of the order in T and the fact that $F(u \leq u) = id_{F(u)}$ for all $u \in T$. Transitivity follows from the transitivity of the order in T and the fact that $F(u \leq v) \circ F(v \leq w) = F(u \leq w)$ for all $u, v, w \in T$ with $u \leq v \leq w$. Anti-symmetry follows the anti-symmetry of the order in T . Indeed, whenever $(u, x) \leq (v, y)$ and $(v, y) \leq (u, x)$, by definition, we must have $u \leq v$ and $v \leq u$ hence $u = v$ and thus $x = id_{F(u)}(y)$ hence $x = y$.

It remains to prove that π is a temporal projection. Let $(u, x), (v, y) \in P_F$. If $(u, x) \leq (v, y)$, then, by definition of the order relation, we must have $\pi(u, x) = u \leq v = \pi(v, y)$ which proves (IN1). Given $w \leq u = \pi(u, x)$ then there is $z = F(w \leq u)(x)$ with, by definition of the order $(w, z) \leq (u, x)$ and if $(w, z') \leq (u, x)$ then, again by definition of the order, we must have $z' = F(w \leq u)(x)$ therefore $z' = z$. This proves (IN2). \square

Lemma 3.11. *Let P be poset timed over T . Then the presheaf $F_P : T^{op} \rightarrow Set$ defined by $F_P(t) = \{x \in P : \pi(x) = t\}$ and $F_P(u \leq v)(x) = x \downarrow u$ for all $t, u, v \in T$ with $u \leq v$ and $x \in F_P(v)$, has an element category isomorphic to P .*

Proof. We first observe that whenever $u \leq v$ we indeed have $u \leq \pi(x)$ for all $x \in F_P(v)$ hence $x \downarrow u$ is well-defined. Then, proving that F_P is a functor just amounts to proving that $F_P(u \leq u) = id_{F_P(u)}$ and $F_P(u \leq v) \circ F_P(v \leq w) = F_P(u \leq w)$ for all $u, v, w \in T$ such that $u \leq v \leq w$ which makes no difficulty. The stated isomorphism between P and the element category of F_P is also straightforward. \square

See Theorem 4.10 below for a stronger statement.

4 Timed morphisms

In this section, we define a class of timed morphisms between timed posets, called Δ -synchronous functions, as monotonic functions between timed posets that uniformly act on the underlying time scales. In some sense, if we consider the time scales over which posets as part of their types, then our proposed definition of timed morphisms allows to type them not only by the type of their domain and codomain but also by the uniform transformation of time scales they induces.

Δ -synchronous function. Throughout this section, let P and, resp. Q , be two posets timed over the time scales U and, resp. V .

Definition 4.1 (Δ -synchronous functions). A function $f : P \rightarrow Q$ is Δ -synchronone when there is a monotone function $\delta : U \rightarrow V$, called the *temporal projection* of f , such that:

$$(SD1) \quad \pi(f(x)) = \delta \circ \pi(x),$$

$$(SD2) \quad \text{if } x \leq y \text{ then } f(x) \leq f(y),$$

$$\text{(SD3)} \quad f(x \downarrow u) = f(x) \downarrow \delta(u),$$

for all $x, y \in P$ and all $u \in U$ such that $u \leq \pi(x)$. This situation is depicted in Figure 2.

$$\begin{array}{ccc} U & \xleftarrow{\pi_P} & P \\ \delta \downarrow & \text{(SD1)} & \downarrow f \\ V & \xleftarrow{\pi_Q} & Q \end{array} \quad \begin{array}{ccc} P & \xrightarrow{\leq_P} & P \\ f \downarrow & \text{(SD2)} & \downarrow f \\ Q & \xrightarrow{\leq_Q} & Q \end{array}$$

Figure 2: Δ -synchronous conditions.

Remark 4.2. Observe that as soon a $u \leq \pi(x)$ then, by monotony of δ we have $\delta(u) \leq \delta \circ \pi(x)$ hence, by (SD1), $\delta(u) \leq \pi(f(x))$ therefore property (SD3) is sound. Property (SD2) states that Δ -synchronous functions are poset functors. Property (SD1) formalizes the statement that they uniformly acts on the underlying time scales.

Example 4.3 (Self-synchronous functions). Every monotone function $\delta : U \rightarrow V$ is a Δ -synchronone between the self-timed posets U and V , with itself as temporal projection.

Example 4.4 (Scale changes on timed signals). Continuing example 2.6, let $\delta : U \rightarrow V$ be a monotone function between two time scales. Let E be a set of event values. Then there is the function $f_\delta : \text{Sig}(U, E) \rightarrow \text{Sig}(V, E)$ defined par

$$f_\delta(u, S) = (\delta(u), \{(\delta(t), e) \in V \times E : (t, e) \in S\})$$

for all partial signal $(u, S) \in \text{Sig}(U, E)$. It is an easy observation that f_δ is a Δ -synchronone with temporal projection δ .

Lemma 4.5. *Under hypothesis (SD1), property (SD2) and (SD3) are equivalent.*

Proof. Let $f : P \rightarrow Q$ be a function and let $\delta : U \rightarrow V$ be a monotone function.

Assume (SD1) and (SD2). Let $x \in P$ and $u \leq \pi(x)$. By (IN2) we have $\pi(x \downarrow u) = u$ hence $\pi(f(x \downarrow u)) = \delta(u)$ by (SD1). By (SD2) we also have $f(x \downarrow u) \leq f(x)$ hence, by (IN1), we have $\delta(u) = \pi(f(x \downarrow u)) \leq \pi(f(x))$ and thus $\delta(u) \leq \pi(f(x))$. It follows, that $f(x \downarrow u) = f(x) \downarrow \delta(u)$ by (IN2). This proves (SD3).

Conversely, assume (SD1) and (SD3). Let $x, y \in P$ such that $x \leq y$. By (IN2) we have $x = y \downarrow \pi(x)$ hence $f(x) = f(y) \downarrow \delta \circ \pi(x)$ by (SD3) and thus $f(x) \leq f(y)$. This proves (SD2). \square

On coherence preservation. It is easy to see that the Δ -synchronous image of a coherent subset bounded above is coherent. However, without the boundedness condition this is no longer true as shown by the following example.

Example 4.6. Let $U = \{\perp, a, b\}$, self-timed, with minimum element \perp and a and b incomparable. Let $P = U$ ordered just as U but timed over $V = \{0, 1\}$ by $\pi(\perp) = 0$, $\pi(a) = 1$ and $\pi(b) = 1$. The function $f : U \rightarrow P$ defined by $f(\perp) = \perp$, $f(a) = a$ and $f(b) = b$ is Δ -synchronous with temporal projection $\delta : U \rightarrow V$ defined by $\delta(\perp) = 0$, $\delta(a) = 1$ and $\delta(b) = 1$. Although $a \supset b$ in U we have $f(a) = a \not\leq b = f(b)$ in P . The function f does not preserve coherence.

Lemma 4.7. *Assume that both U and V are meet-semilattices. Let $\delta : U \rightarrow V$ be meet-preserving function and let $f : P \rightarrow Q$ be a Δ -synchronous function with temporal projection δ . Then f is coherence preserving. Moreover, both P and Q are conditional meet-semilattice and the function f preserves contional meet, i.e. it is stable in the sense of Berry [5].*

Proof. We first prove that f preserves coherence. Let $x, y \in P$ such that $x \circ y$. We aim at proving that $f(x) \circ f(y)$. Let $v \leq \pi(f(x), \pi(f(y)))$. By (SD1), both U and V are meet-semilattices and δ is meet-preserving, we have $v \leq \delta(\pi(x) \wedge \pi(y))$. Given $u = \pi(x) \wedge \pi(y)$, since $x \circ y$ we have $x \downarrow u = y \downarrow u$ hence, by (SD3), $f(x) \downarrow \delta(u) = f(y) \downarrow \delta(u)$. It follows, since $v \leq \delta(u)$ that we have $f(x) \downarrow v = f(y) \downarrow v$. This concludes the proof that $f(x) \circ f(y)$.

Let $x, y, z \in P$ such that $x, y \leq z$. This means that $x = z \downarrow \pi(x)$ and $y = z \downarrow \pi(y)$ therefore, as routine checking shows, we have $x \wedge y = z \downarrow \pi(x) \wedge \pi(y)$. Indeed, if there is any $z' \leq x, y$ then we have $\pi(z') \leq \pi(x) \wedge \pi(y)$ therefore $z' = z \downarrow \pi(z) \leq z \downarrow \pi(x) \wedge \pi(y)$.

By definition, a function is stable when it preserves pullbacks, that is, with timed posets, conditional meets. Let again $x, y, z \in P$ such that $x, y \leq z$. Since $x \wedge y = z \downarrow \pi(x) \wedge \pi(y)$, by (SD3), we have $f(x \wedge y) = f(z) \downarrow \delta(\pi(x) \wedge \pi(y))$. Since δ is meet preserving, together with (SD1), this yields $f(x \wedge y) = f(z) \downarrow \pi(f(x)) \wedge \pi(f(y))$ therefore $f(x \wedge y) = f(x) \wedge f(y)$ since V is also a meet-semilattice. This concludes the proof that f is stable. \square

Remark 4.8. In example 4.6, both U and V are meet semi-lattices, but the function $\delta : U \rightarrow V$ is indeed not meet-preserving since, $a \wedge b = \perp$ while $\delta(\perp) = 0 < 1 = \delta(a) \wedge \delta(b)$.

Synchronous functions. We restrict our attention to a smaller class of timed morphisms we call *synchronous* for they produce output at the same time they read input.

Definition 4.9 (Synchronous function). A Δ -synchronous function $f : P \rightarrow Q$ with temporal projection $\delta : U \rightarrow V$ is a *synchronous function* when $U = V$ and $\delta = id_U$. In other words, f is synchronous when:

$$(SI1) \quad \pi(f(x)) = \pi(x),$$

$$(SI2) \quad \text{if } x \leq y \text{ then } f(x) \leq f(y),$$

$$(SI3) \quad f(x \downarrow u) = f(x) \downarrow u,$$

for all $x \in P$ and $u \in U$ such that $u \leq \pi(x)$.

Theorem 4.10. *The category $TPoset(T)$ of posets timed over T and synchronous functions is equivalent to the category $Psh(T)$ of presheaves over T and natural transformations.*

More precisely, the functor $\varphi : TPoset(T) \rightarrow Psh(T)$ defined, for every timed poset $P \in TPoset(T)$, by $\varphi(P) = F_P$ (see 3.11) and, for all synchronous function $f : P \rightarrow Q$ in $TPoset(T)$, by $\varphi(f) = \alpha$ with $\alpha_t(x) = f(x)$ for all $t \in T$ and $x \in F_P(t)$, is a categorical equivalence.

Proof. A pseudo inverse functor for φ is the functor $\psi : Psh(T) \rightarrow TPoset(T)$ defined, for all presheaf $F \in Psh(T)$, by $\psi(F) = P_F$ (see 3.10), and, for all natural transformation $\alpha : F \Rightarrow G$ in $Psh(T)$, by $\psi(\alpha) = f$ with $f(t, x) = (t, \alpha_t(x))$ for all $(t, x) \in P_F$. \square

We shall see in the next section that this implies that the category $TPoset(T)$ is a topos.

From Δ -synchronous to synchronous functions. Synchronous functions eventually provide a double characterization of Δ -synchronous functions as stated in the next theorem which generalizes in some sense Example 4.4.

Theorem 4.11 (Left Kan extension). *Let $\delta : U \rightarrow V$ be a monotone function. Then there are two categorical functors*

$$\delta_! : TPoset(U) \rightarrow TPoset(V) \quad \text{and} \quad \delta^* : TPoset(V) \rightarrow TPoset(U)$$

and, for every timed poset $P \in TPoset(U)$ and $Q \in TPoset(V)$, two Δ -synchronous functions

$$\alpha_P : P \rightarrow \delta_!(P) \quad \text{and} \quad \omega_Q : \delta^*(Q) \rightarrow Q$$

both with temporal projection δ , such that every Δ -synchronous $f : P \rightarrow Q$ with temporal projection δ we both have:

- (1) there is a unique synchronous function $f_! : \delta_!(P) \rightarrow Q$ such that $f = f_! \circ \alpha_P$,
- (2) there is a unique synchronous function $f^* : P \rightarrow \delta^*(Q)$ such that $f = \omega_Q \circ f^*$.

i.e. f uniquely factorizes through α_P or ω_Q . In particular, there is the categorical adjunction $\delta_! \dashv \delta^*$. This situation is depicted in Figure 3.

$$\begin{array}{ccccc}
 & & & \exists! f_! & \\
 & & \delta_!(P) & \xrightarrow{\quad} & Q \\
 & \delta \uparrow & \alpha_P \uparrow & \searrow \forall f & \uparrow \omega_Q \\
 & U & P & \xrightarrow{\quad} & \delta^*(Q) \\
 & & & \exists! f^* &
 \end{array}$$

Figure 3: Δ -synchronous via synchronous.

Proof. The functor δ^* is defined as follows. Let $Q \in TPoset(V)$. We define $\delta^*(Q) = \{(u, y) \in U \times Q : \pi(y) = \delta(u)\}$ ordered point-wise with projection $\pi(u, y) = u$ for all $(u, y) \in \delta^*(Q)$. Clearly π is monotone, moreover, for every $(u, y) \in \delta^*(Q)$ and $u' \leq u$ we have $(u, y) \downarrow u' = (u', y \downarrow \delta(u'))$ which proves that $\delta^*(Q) \in TPoset(V)$.

Let $h : Q \rightarrow Q'$ synchronous into some $Q' \in TPoset(V)$. We define $\delta^*(h) : \delta^*(Q) \rightarrow \delta^*(Q')$ by $\delta^*(h)(u, y) = (u, h(y))$ for all $(u, y) \in \delta^*(Q)$. Clearly $\delta^*(h)$ is synchronous, and δ^* defined this way is indeed a functor.

Let then $\omega_Q : \delta^*(Q) \rightarrow Q$ simply defined by $\omega_Q(u, y) = y$ for every $(u, y) \in \delta^*(Q)$. By definition, we have $\pi(y) = \delta(u)$ for all $(u, y) \in \delta^*(Q)$ and since ω_Q is clearly monotone this proves it is Δ -synchronous.

Given Δ -synchronous $f : P \rightarrow Q$ with temporal projection δ , we define $f^* : P \rightarrow \delta^*(Q)$ by $f^*(x) = (\pi(x), f(x))$. Given $x \in P$, by (SD1) we have $(\pi(x), f(x)) \in \delta^*(Q)$ and $\pi(f^*(x)) = \pi(x)$. Given $u \leq \pi(x)$, we also have $f^*(x \downarrow u) = (u, f(x \downarrow u))$ hence, by (SD2), $f^*(x \downarrow u) = (u, f(x) \downarrow \delta(u))$ hence $f^*(x \downarrow u) = f^*(x) \downarrow u$. This proves that f^* is synchronous.

Last, by construction, we have $f = \omega_Q \circ f^*$. The unicity of f^* follows from the fact that, though not injective, ω_Q is a mono since, when $f = \omega_Q \circ g$, for every $x \in P$, the value of $f(x)$ in Q fixes the right projection of $g(x)$ and synchronicity fixes its left projection.

The functor $\delta_!$ is defined as follows. Let $P \in TPoset(U)$. We define $x \uparrow v = \{y \in P : \exists z \in P, v \leq \delta \circ \pi(z), z \leq x, y\}$ for every $x \in P$ and $v \in V$. The set $X \uparrow v$ is the connected component of x above v via δ . This allows for defining $\delta_!(P) = \{(v, X \uparrow v) \in V \times \mathcal{P}(P) : v \in V, x \in P, v \leq \pi(x)\}$ ordered point-wise with reverse inclusion over $\mathcal{P}(P)$ and with projection $\pi(v, X) = v$ for every $(v, X) \in \delta_!(P)$. We check that $(v, X) \downarrow w = (w, X \uparrow w)$ for all $w \leq v$ and $x \in X$.

Let $g : P \rightarrow P'$ be a synchronous function from into $P' \in TPoset(U)$. We define $\delta_!(g) : \delta_!(P) \rightarrow \delta_!(P')$ by $\delta_!(g)(v, X) = (v, g(x) \uparrow v)$ for any choice of $x \in X$. If we chose another $y \in X$, then there is $z \in P$ such that $v \leq \delta \circ \pi(z)$ and $z \leq x, y$. By monotony of g we thus have $f(z) \leq g(x), g(y)$ with $\pi(f(z)) = \pi(z)$ since g is synchronous therefore $f(y) \in f(x) \uparrow v$ hence, as observed above, $f(y) \uparrow v = f(x) \uparrow v$, i.e. our definition is sound.

We easily check that $\delta_!(g)$ is synchronous and proving that $\delta_!$ defined as above is a functor makes no difficulty.

Let then $\alpha_P : P \rightarrow \delta_!(P)$ simply defined by $\alpha_P(x) = (\delta \circ \pi(x), x \uparrow \delta \circ \pi(x))$ for every $x \in P$. We easily check that α_P is monotone and since $\pi(\alpha_P(x)) = \delta \circ \pi(x)$ for all $x \in P$, it is Δ -synchronous. The fact $\delta_!$ defined as above is a functor makes no difficulties.

Given Δ -synchronous $f : P \rightarrow Q$ with temporal projection δ , we define $f_! : \delta_!(P) \rightarrow Q$ by $f_!(v, X) = f(x) \downarrow v$ for every $(v, X) \in \delta_!(P)$ and any choice of $x \in X$. We check that we indeed have $v \leq \delta \circ \pi(x)$ hence, by (SD1), $v \leq \pi(f(x))$. If we chose another $y \in X$, then there is $z \in P$ such that $v \leq \delta \circ \pi(z)$ and $z \leq x, y$. This implies that $f(z) \leq f(x), f(y)$ hence, since $v \leq \pi(f(z))$, we have $f(z) \downarrow v = f(x) \downarrow v$ and $f(z) \downarrow v = f(y) \downarrow v$ hence $f(x) \downarrow v = f(y) \downarrow v$, i.e. our definition is sound.

Given $(v, X) \in \delta_!(P)$, by (SD1), we clearly have $v = \pi(f_!(v, X))$ and since $f_!$ is obviously monotone this proves it is synchronous.

The fact $f_! \circ \alpha_P = f$ follows from the definition. The unicity of $f_!$ follows from the fact that α_P , though not surjective, is an epi since every element $(v, X) \in \delta_!(P)$ lays below $\alpha_P(y)$ for some $y \in P$ with $X \leq y$ hence $f_!(v, X)$ is fully determined by $f_! \circ \alpha_P(y) = f(y)$. \square

Remark 4.12. Every choice of a time scale provides a granularity at which a system behavior can be observed. The above result can thus be seen as a tool box that allows for analyzing the behavior of a timed system at various granularity. Though further work is required for a deeper understanding of how such a fact can be used in concrete application cases, there appears some potential links with abstraction/refinement techniques for system design [1] and abstract interpretation techniques for system analysis [13].

5 More on synchronous functions

Theorem 4.11 shows that Δ -synchronous functions are inherently linked with synchronous functions. Theorem 4.10 ensures that the category $TPoset(T)$ of poset timed over T and synchronous functions is a topos. We review below the concrete constructions over timed posets that derive from such a result. With a view towards system modeling, each of these constructions have a fairly intuitive interpretation.

Lemma 5.1 (The clock). *Let P be a poset timed over T . Then the temporal projection $\pi : P \rightarrow T$ is the unique synchronous function from P into T . In particular, the self-timed poset T is terminal in $TPoset(T)$.*

Remark 5.2. In some sense, the time scale T seen as a sel-timed poset, can be seen as a clock. Indeed, every synchronous function $c : T \rightarrow P$ can be viewed as a timed constant which is produced, pieces after pieces, as time is passing.

Observe however that the existence of such a function implies that $c(T)$ is a subset of P isomorphic to T . There are posets timed over T with no such a subset as show by any strict downward closed subset of T (see Example 2.4).

In other words, timed posets may also contain timed constant which evolution in time may stop at some instant as if they are timed over a smaller (sub) time scale.

Definition 5.3 (Synchronous product). Let P, Q be two posets timed over T . The *synchronous product* of P and Q is defined as the set $P \otimes Q = \{(x, y) \in P \times Q : \pi_P(x) = \pi_Q(y)\}$ ordered point-wise and equipped with the temporal projection defined by $\pi(x, y) = \pi(x) = \pi(y)$ for all $(x, y) \in P \otimes Q$.

Remark 5.4. Though we do not propose (yet) a programming language associated to timed poset, the interpretation of the above synchronous product shall be obvious. Thanks to the fact it is a categorical product (see below), every pair of synchronous functions $f : R \rightarrow P$ and $g : R \rightarrow Q$ uniquely factorizes through the gluing $f \times g : R \rightarrow P \otimes Q$ of the functions f and g . This combinator over functions is much useful in data flow programming that allows for using synchronous functions in an arrow programming style [18].

Lemma 5.5. *Then $P \otimes Q$ is a timed poset over T and, with projections $p_1 : P \otimes Q \rightarrow P$ and $p_2 : P \otimes Q \rightarrow Q$, it is the categorical product of P and Q in $T\text{Poset}(T)$.*

Proof. Given $f : R \rightarrow P$ and $g : R \rightarrow Q$, we have to show that $f \times g : R \rightarrow P \otimes Q$ defined for all $z \in R$ by $f \times g(z) = (f(z), g(z))$ is the unique synchronous function $h : R \rightarrow P \otimes Q$ through which f and g factorizes, i.e. we have $f = p_1 \circ h$ and $g = p_2 \circ h$, as depicted in Figure 4.

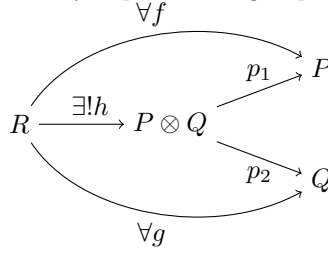


Figure 4: Categorical product diagram.

This property is routine to check. □

Remark 5.6. One can check that the disjoint sum² $P \oplus Q$ of two timed posets with canonical injection eventually leads to the definition of categorical co-product of P and Q in $T\text{Poset}(T)$. Then the empty timed poset is the initial object in $T\text{Poset}(T)$. This shows that the category $T\text{Poset}(T)$ is bi-cartesian.

With a view towards application, such a co-product can also be used for modeling alternative. However, it is very likely that such an alternative must be solved “at initialization time” before any instant in T . Indeed, when T is totally ordered, then a coherent subset of $P \otimes Q$ can only be (the embedding of) a coherent subset of P or coherent subset of Q . In this case, every mixed subset with elements from both P and Q is time incoherent.

Definition 5.7 (Temporal cut of a synchronous function). Let $f : P \rightarrow Q$ be a synchronous function on the time scale T . Let $u \in T$. The *temporal cut* of f at u is the function $f \downarrow u : P \downarrow u \rightarrow Q \downarrow u$ defined by $P \downarrow u = \{x \in P : \pi(x) \leq u\}$, $Q \downarrow u = \{y \in Q : \pi(y) \leq u\}$ and $(f \downarrow u)(x) = f(x)$ for all $x \in P \downarrow u$.

Observe that both $P \downarrow u$ and $Q \downarrow u$ are downward closed hence (see Example 2.4) they both are posets timed over T . Moreover, since f is synchronous, property (SI1) ensures that $f(x) \in Q \downarrow u$ for all $x \in P \downarrow u$ therefore $f \downarrow u$ is a well defined synchronous function from $P \downarrow u$ into $Q \downarrow u$.

²Possibly gluing minimal elements when considering the subcategory of timed posets with a minimum elements

Definition 5.8 (Synchronous exponent). Let P and Q be two posets timed over T . The *synchronous exponent* of Q by P is defined to be the set $[P \rightarrow_T Q]$ of all pair (u, h) with $u \in T$ and synchronous functions $h : P \downarrow u \rightarrow Q \downarrow u$, with temporal projection defined by $\pi(u, h) = u$ and partial order defined by $(u_1, h_1) \leq (u_2, h_2)$ when $u_1 \leq u_2$ and $h_1 = h_2 \downarrow u_1$ for all $(u, h), (u_1, h_1), (u_2, h_2) \in [P \rightarrow_T Q]$.

Lemma 5.9. *Let P, Q, R be three posets timed over T . Then $[Q \rightarrow_T R]$ is a poset timed over T . Moreover, there is also the synchronous function $eval : Q^R \otimes Q \rightarrow R$ defined by $eval((u, h), y) = h(y)$ for all $((u, h), y) \in R^Q \otimes Q$ therefore with $\pi(y) = u$.*

The timed poset $[Q \rightarrow_T R]$ with function $eval$ is the categorical exponent of R by Q in $T\text{Poset}(T)$, i.e. for all synchronous function $g : P \otimes Q \rightarrow R$, the function $g^ : P \rightarrow R^Q$ defined, for all $x \in P$, by $g^*(x) = (\pi(x), \lambda y. g(x \downarrow \pi(y), y))$ is the unique synchronous function from P into R^Q such that $g(x, y) = eval(g^*(x), y)$ for all $(x, y) \in P \otimes Q$.*

Proof. The above situation is depicted in Figure 5.

$$\begin{array}{ccc}
 P & & P \otimes Q \\
 \exists! g^* \downarrow & & g^* \times id \downarrow \\
 R^Q & & R^Q \otimes Q \\
 & & \xrightarrow{\quad eval \quad} \\
 & & R
 \end{array}
 \quad
 \begin{array}{c}
 \\
 \\
 \\
 \searrow \forall g \\
 \\
 \\
 \end{array}$$

Figure 5: Categorical exponent diagram.

The proof of this lemma essentially goes by first checking that both $eval$ and g^* are indeed well defined synchronous functions. In particular, when $y \in P \downarrow \pi(x)$ then we indeed have $\pi(y) \leq \pi(x)$ therefore $eval(g^*(x), y) = f(x \downarrow \pi(y), y)$ is indeed well-defined. The fact that g^* is the unique synchronous function such that $g(x, y) = eval(g^*(x), y)$ for all $(x, y) \in P \otimes Q$ is then routine to check. \square

Remark 5.10. In other words, such a result states that synchronous functions can themselves be represented by (maximal) coherent subsets of a timed poset that can be transmitted piece by piece in time. The synchronous function $eval$ thus encodes the *on-the-fly* application of such a function to its argument. Observe moreover that, by definition of the synchronous product, whenever the argument is fully received then such an on-the-fly application ends since the possible remaining values of the function cannot be synchronized with any further received argument.

The following lemma comes as a complement of the construction of timed posets from downward closed subsets of timed posets (see 2.4), by characterizing subobjects³ of the category $T\text{Poset}(T)$ precisely as these downward closed subsets.

Lemma 5.11 (Timed subobjects). *Let $f : P \rightarrow Q$ be a synchronous function. Then $f(P) = \downarrow f(P)$, moreover, f is injective if and only if $P \simeq f(P)$ as posets.*

Proof. Let $x \in P$ and let $y \leq f(x)$. This means that $y = f(x) \downarrow \pi(y)$. By (SI3) we thus have $y = f(x \downarrow \pi(y))$ hence $y \in f(P)$.

Given then $x, y \in P$. If $x \leq y$ then, by (SI2), we have $f(x) \leq f(y)$. Conversely, assume that $f(x) \leq f(y)$ hence $f(x) = f(y) \downarrow \pi(f(x))$. By (SI1) and monotony of π we thus have

³One can easily verify that the monomorphisms in $T\text{Poset}(T)$ are the injective synchronous functions. Then, as a consequence of the lemma, every injective synchronous function $f : Q \rightarrow P$ is equivalent (as sub-object) with the inclusion synchronous function $inc_{f(Q)} : f(Q) \rightarrow P$.

$\pi(x) \leq \pi(y)$. Then, (SI3) we have $f(y \downarrow \pi(x)) = f(y) \downarrow \pi(x)$ hence, by (SI1), $f(y \downarrow \pi(x)) = f(y) \downarrow \pi(f(x))$. Together, we thus have $f(y \downarrow \pi(x)) = f(x)$ hence $x = y \downarrow \pi(x)$ since f is injective and thus $x \leq y$. \square

In the category *Set*, there is the powerset construction $\mathcal{P}(E)$ of subsets of a set E . In $T\text{Poset}(T)$ the analogous power object is defined below.

Definition 5.12 (Synchronous power). Let P be a poset timed over T . The *synchronous power* of T is defined as the set $\Omega^P = \{(u, X) \in T \times \mathcal{P}(P) : X = \downarrow X, \pi(X) \leq u\}$ with temporal projection defined by $\pi(u, X) = u$ and partial order defined by $(u, X) \leq (v, Y)$ when $u \leq v$ and $X = Y \downarrow u$ for all $(u, X), (v, Y) \in \Omega^P$.

Lemma 5.13. *The synchronous power Ω^P of a timed poset P is a poset timed over T and the powerobject of P in $T\text{Poset}(P)$.*

Proof. We first prove that Ω^T is the subobject classifier in $T\text{Poset}(T)$ as stated in Remark 5.15. For such a purpose, given a poset P timed over T , we easily check that the characteristic function $\mathcal{X}_X : P \rightarrow \Omega^T$ of a downward closed subset $X \subset P$ is uniquely defined by $\mathcal{X}_X(x) = (\pi(x), \pi(\downarrow x \cap X))$ for every $x \in P$.

Then we prove that that $\Omega^P \simeq [P \rightarrow \Omega^T]$ which, by known result in topos theory proves that Ω^P is indeed the power object of P . The expected isomorphism $f : \Omega^P \rightarrow [P \rightarrow \Omega^T]$ is defined, for every $(u, X) \in \Omega^P$, by $f(u, X) = (u, \mathcal{X}_X \downarrow u) \in [P \rightarrow \Omega^T]$. The inverse image of every pair $(u, h) \in [P \rightarrow \Omega^T]$ is defined by $f^{-1}(u, h) = (u, \{x \in P \downarrow u : h(x) = \text{true}(x)\})$ with *true* defined in Remark 5.15. \square

Remark 5.14. The interpretation of Ω^P in terms of temporal traces is also fairly obvious. It allows to represent arbitrarily many computation traces in P ran synchronously. One can observe however, much like is sets by opposition to multisets that there two sub-traces identical at some instant are considered as a single one. In other words, the number of “processors” onto which these traces are run are kept minimum at every instant.

Extending this construction to allow “weighted” runs of traces, either with integer weight as with multisets or positive reals as with probability distributions distribution is surely possible but goes out of the scope of this overview.

Observe also that we have given above a parallel interpretation of the timed powerobject. It would also make sense to interpret it as a non deterministic choice as done with power domain constructions. The powerobject construct behaves much like the powerset construct. What it should actually model is of course a matter of interpretation.

Remark 5.15. Following topos theory [2, 26], there is $\Omega^T = \{(u, V) \in T \times \mathcal{P}(T) : V = \downarrow V \leq u\}$, with the synchronous function *true* : $T \rightarrow \Omega^T$ defined by *true*(t) = ($t, \downarrow t$) for all $t \in T$ which form the *subobject classifier* in $T\text{Poset}(T)$. In other words, as depicted in Figure 6, for

$$\begin{array}{ccc} P & \xrightarrow{\pi} & T \\ \forall f \downarrow & & \downarrow \text{true} \\ Q & \xrightarrow{\exists! \mathcal{X}_f} & \Omega^T \end{array}$$

Figure 6: Subobject classifier diagram.

all poset P timed over T , for all downward closed subset $X \subseteq P$, there is a unique synchronous

function $\mathcal{X}_X : P \rightarrow \Omega$, the characteristic morphism of X , such that, given the synchronous inclusion $inc_X : X \rightarrow P$ we have $\mathcal{X}_f \circ inc_X = true \circ \pi$.

As an immediate consequence of Theorem 4.10 or gathering the results stated in this section⁴ we have:

Theorem 5.16. *The category $TPoset(T)$ is a topos, i.e. it is cartesian closed, finitely complete and well-powered.*

6 More on Δ -synchronous functions

There are several candidates as relevant categories of timed posets Δ -synchronous function. In indeed, one can select the underlying category \mathcal{C} of time scales and time scale transformations such as the category $Poset$ for the most general case, or, as observed in 4.7, the category $SLattice$ for preserving coherent subsets.

The resulting categories of timed posets and Δ -synchronous functions are denoted by $TPoset(\mathcal{C})$ where \mathcal{C} is the chosen category of time scales.

Lemma 6.1 (One instant clock). *Let $\mathbf{1} = \{*\}$ be the one element poset timed over itself. Then $\mathbf{1}$ is the terminal element in $TPoset(\mathcal{C})$.*

Let $P \in TPoset(\mathcal{C})$ timed over $U \in \mathcal{C}$ and let $Q \in TPoset(\mathcal{C})$ timed over $U \in \mathcal{C}$.

Definition 6.2 (Asynchronous product). The *asynchronous product* of P and Q is defined as the cartesian product $P \times Q$ with temporal projection $\pi : P \times Q \rightarrow U \times V$ by $\pi(x, y) = (\pi(x), \pi(y))$, ordered point wise.

Lemma 6.3. *Then $P \times Q$ is a poset timed over $U \times V \in \mathcal{C}$. Both projections $p_1 : P \times Q \rightarrow P$ and $p_2 : P \times Q \rightarrow Q$ are Δ -synchronous with temporal projection $\pi(p_1) = p_1 : U \times V \rightarrow U$ in \mathcal{C} and $\pi(p_2) = p_2 : U \times V \rightarrow V$ in \mathcal{C} . Together, they form the categorical product of P and Q in $TPoset(\mathcal{C})$.*

Definition 6.4 (Asynchronous exponent). The *asynchronous exponent* of Q by P is defined as the set Q^P of Δ -synchronous function from P into Q with temporal projections as already defined.

Lemma 6.5. *Then Q^P is a poset timed over $V^U \in \mathcal{C}$. The evaluation mapping $eval : Q^P \times P \rightarrow Q$ defined, for all $f \in Q^P$ and $x \in P$, by $eval(f, x) = f(x)$ is Δ -synchronous with, thanks to (SD1), temporal projection $\pi(eval) = eval$ in \mathcal{C} . Together, they form the exponent of Q by P in $TPoset(\mathcal{C})$.*

Gathering all the above lemmas, we have:

Theorem 6.6. *The category $TPoset(\mathcal{C})$ is cartesian closed.*

Remark 6.7. In all the above cases, the empty poset timed over the empty time scale is the initial object. The disjoint sum is the co-product in $TPoset(\mathcal{C})$ therefore $TPoset(\mathcal{C})$ is even bi-cartesian closed. The next theorem generalizes such a remark.

⁴additionally proving that $TPoset(T)$ also has all equalizers, which is easy since they are essentially defined as in *Set*.

Definition 6.8 (Temporal projection of diagram functor). Let $\mathcal{C} \subseteq \text{Poset}$ be a sub-category. Let $G = \langle V, E, s, t \rangle$ be a graph with vertices V , edges E , source $s : E \rightarrow V$ and target $t : E \rightarrow V$. Let $F : G \rightarrow \text{TPoset}(\mathcal{C})$ be a diagram functor⁵. The temporal projection of F is defined to be the diagram functor $H : G \rightarrow \mathcal{C}$ defined by, for all $v \in V$, the poset $H(v)$ is the time scale over which $F(v)$ is timed and, for all $e \in E$, the time scale transformation $H(e)$ is the temporal projection $\pi \circ F(e)$ of the Δ -synchronous function $F(e)$.

Theorem 6.9 (Existence of limit and colimit). *A diagram functor $F : G \rightarrow \text{TPoset}(\mathcal{C})$ has a limit (resp. a co-limit) in $\text{Poset}(\mathcal{C})$ whenever its temporal projection $H : G \rightarrow \mathcal{C}$ has a limit (resp. a colimit) in \mathcal{C} .*

Proof. Throughout this proof, for each $v \in V$, let $\pi_v : F(v) \rightarrow H(v)$ be the temporal projection of the poset $F(v)$ timed over $H(v)$.

The limit case. Assume H has a limit cone given by $T \in \mathcal{C}$ and, for each $v \in V$, a fonction $\delta_v : T \rightarrow F(v)$ in \mathcal{C} . By definition, we have $\delta_{t(e)} = H(e) \circ \delta_{s(e)}$ for every $e \in E$. Let then

$$P = \bigcup_{t \in T} \left(\{t\} \times \prod_{v \in V} \pi_v^{-1}(\delta_v(t)) \right)$$

ordered point-wise with first projection as temporal projection. It is a poset timed over T . There are also the Δ -synchronous projection $p_v : P \rightarrow F(v)$, one for each $v \in V$, defined by

$$p_v((t, \{x_v\}_{v \in V})) = x_v$$

for every $(t, \{x_v\}_{v \in V}) \in P$ with temporal projection $\pi(p_v) = \delta_v$. Then it is routine to check that the subset

$$X = \{(t, \{x_v\}_{v \in V}) \in P : \forall e \in E, x_{t(e)} = F(e)(x_{s(e)})\}$$

is downward closed therefore a sub-timed poset of P and, equipped with the (restriction of) the projection $\{p_v\}_{v \in V}$ is the limit of F .

The colimit case. Assume that H has a colimit cocone given by $T \in \mathcal{C}$ and, for each $v \in V$, a fonction $\delta_v : F(v) \rightarrow T$ in \mathcal{C} . By definition, we have $\delta_{s(e)} = \delta_{t(e)} \circ H(e)$ for every $e \in E$. Then we can define

$$P = \bigcup_{t \in T} \left(\{t\} \times \sum_{v \in V} \delta_v^{-1}(t) \right)$$

ordered point-wise with first projection as temporal projection. It is a poset timed over T . Moreover, there are the Δ -synchronous injection $i_v : F(v) \rightarrow P$ defined by

$$i_v(x) = (\delta_v(x), v, x)$$

for all $x \in F(v)$ with temporal projection δ_v and we have $P = \bigcup \{i_v(x) \in P : v \in V, x \in F(v)\}$.

Let then \simeq be the least equivalence over P such that for all $e \in E$ and $x, y \in F(s(e))$ we have $x \simeq F(e)(x)$ and if $x \simeq y$ then $F(e)(x) \sim F(e)(y)$.

It is routine checking that for all $x, y \in P$, if $x \simeq y$ then $\pi(x) = \pi(y)$ and for all $t \in T$ such that $t \leq \pi(x)$ we have $x \downarrow t \simeq y \downarrow t$. It follows that P / \simeq can be equipped with the order defined

⁵We call here a diagram functor a functor from the category freely generated by a graph G . As such a functor is fully determined by its value on graph vertices and edges it can simply be seen as a graph morphism from G into (the graph of) its codomain category.

for all $x, y \in P$ by $[x] \leq [y]$ if and only if there is $x' \in [x]$ such that $x' \leq y$ and the temporal projection $\pi([x]) = \pi(x)$.

Then P/\simeq is a poset timed over T and the functions $j_v : F(v) \rightarrow P/\simeq$, one for each $v \in V$, defined for all $x \in F(v)$ by $j_v(x) = [i_v(x)]$ are Δ -synchronous. By construction, for every $e \in E$ we have $j_{s(e)} = j_{t(e)} \circ F(e)$. Then, it is routine to check that P/\simeq equipped with the functions $\{j_v\}_{v \in V}$ form the colimit of F . \square

In other words, fixpoint equations of timed domains have an inductive (resp. co-inductive) solution whenever the projection of these equations over time scales have an inductive (resp. co-inductive) solution.

7 Timed domains

We define and study in this section the notion of timed domain, simply defined as cpos timed over cpos. For such a purpose we define the notion of continuous timed poset in such a way that timed domains exactly corresponding to continuous timed poset that are timed over cpos.

Definition 7.1 (Continuous timed posets). Let P be a poset timed over T . We say that P is a continuous timed poset when

(IN3) if X is directed and $\bigvee \pi(X)$ is defined then so is $\bigvee X$,

for all $X \subseteq P$.

Lemma 7.2. *Let P be a continuous timed poset timed over T . Then, for every $x \in P$ the local cut $cut_x : \downarrow \pi(x) \rightarrow P$ (see 2.7) is continuous.*

Proof. Let $x \in P$ and let $U \subseteq T$, directed, such that $U \leq \pi(x)$. Assume that there is $v = \bigvee U$. Proving the continuity of cut_x amounts to proving that $\bigvee x \downarrow U = x \downarrow v$.

Since the local cut is monotone the set $x \downarrow U$ is directed with $\pi(x \downarrow U) = U$. By (IN3), there is thus $y \in P$ such that $\bigvee (x \downarrow U) = y \leq x$. Moreover, by (IN1), we have $\pi(y) \leq \pi(x)$ hence, by (IN2), we have $y = x \downarrow \pi(y)$.

Since $\pi(x \downarrow u) = u$, by monotony of the temporal projection, we have $\bigvee U = v \leq \pi(y)$. Moreover, since $\bigvee U = v$ we have $x \downarrow U \leq x \downarrow v$ hence, by definition of least upper bound we have $y = x \downarrow \pi(y) \leq x \downarrow v$ hence $\pi(y) \leq v$. This proves that $\pi(y) = v$ hence $y = x \downarrow v$. \square

Remark 7.3. Observe that, in general, the continuity of all local cut in a timed poset *does not* imply the continuity of that timed poset as shown by the example $P_1 = \mathbb{N}$ timed as a timed subset of $T_1 = \overline{\mathbb{N}} = \mathbb{N} \cup \{\infty\}$ self-timed.

Also, the continuity of a timed poset *does not* imply the continuity of its temporal projection as shown by the example $P_2 = \overline{\mathbb{N}}$ timed as a timed subset of $T_2 = \overline{\mathbb{N}} \cup \{\infty'\}$ self-timed with ∞' another upper bound of \mathbb{N} distinct and incomparable with ∞ .

Lemma 7.4. *Let P be a continuous timed poset, timed over T . Assume that T is complete. Then P is complete and its temporal projection is continuous.*

Proof. Let $X \subseteq P$ directed. By monotony of the temporal projection, the set $\pi(X) \subseteq T$ is directed. Since T is complete, there is $v = \bigvee \pi(X)$. Since P is continuous, by (IN3), there is $y = \bigvee X$. This concludes the proof that P is complete.

For every $x \in X$, we have $x \leq y$ therefore we have $x = y \downarrow \pi(x)$. Since $\pi(X)$ is directed, by continuity of cut_y (Lemma 7.2) we have $\bigvee x = y \downarrow \bigvee \pi(X)$ that is $y = y \downarrow v$ hence $v = \pi(y)$. This concludes the proof that π is continuous. \square

Definition 7.5 (Timed domain). A *timed domain* is a continuous timed poset timed over a complete time scale, hence, as proved above also a complete poset with continuous local cuts and temporal projection.

Lemma 7.6 (Δ -synchronous vs continuous). *Let P and Q be two timed domains. Let $f : P \rightarrow Q$ be a Δ -synchronous function with temporal projection $\delta : U \rightarrow V$. Assume that δ is continuous. Then f is continuous.*

Proof. Let $X \subseteq P$ directed and, by completeness of P , let $y = \bigvee X$. For every $x \in X$ we have $x \leq y$ hence $x = y \downarrow \pi(x)$ and thus, by (SD3), $f(x) = f(y) \downarrow \delta \circ \pi(x)$ with $\delta \circ \pi(x) \leq \pi(f(y))$. Since the temporal projection π , the function δ and the local cut $cut_{f(y)}$ are all continuous this ensures that $\bigvee f(X) = f(y) \delta \circ \pi(\bigvee X)$. By continuity of π we have $\pi(\bigvee X) = \pi(y)$. By (SD1) we have $\pi(f(y)) = \delta \circ \pi(y)$ therefore by (IN2) we have $f(y) \delta \circ \pi(\bigvee X) = f(y)$ hence $\bigvee f(X) = f(y)$. \square

Remark 7.7. As a special case of the above lemma every synchronous function between timed domains is continuous.

Theorem 7.8. *Let T be a cpo. Then the category $TCpo(T)$ of timed domain over T and, when T has a least element, the category $TCpo_{\perp}(T)$ of timed domain over T with least element, with, in both cases, synchronous (and continuous) functions between them, are topoi.*

Proof (Sketch of). The proof goes by rephrasing Scott topology in terms of a Grothendieck topology J over posets in such a way that (the element categories of the) sheaves in $Sh(T, J)$ are the continuous timed posets timed over T .

As a reminder, a subset X of a poset E is Scott closed when it is downward closed and for every directed $Y \subseteq X$, if $\bigvee Y$ is defined then $\bigvee Y \in X$. Then, for every $t \in T$, we define $J(t)$ be the set of all downward closed subsets $U \subseteq T$ such that $U \leq t$, i.e. U is a sieves on t , and such that the Scott closure \bar{U} of U equals $\downarrow t$.

One can easily check that J is a Grothendieck topology (see [26] p 110). Moreover, one can also check that a timed poset $P \in TPoset(T)$ is continuous if and only if it associated presheaf $F_P : T^{op} \rightarrow Set$ (see 3.11) is a sheaf for J (see [26] p 121). This ensures that the categorical equivalence between $TPoset(T)$ and $Psh(T)$ (see Theorem 4.10) also defines a categorical equivalence between the subcategories $TCpo(T)$ and $Sh(T, J)$. It follows that $TCpo(T)$ is also a topos since $Sh(T, J)$ is.

For the category $TCpo_{\perp}(T)$ the argument is similar though taking instead the topology J' defined from J by letting $J'(t) = J(t)$ when $t > \perp$ and $J'(\perp) = \{\{\perp\}, \emptyset\}$. This forces all sheaves in $Sh(T, J')$ to be singletons on \perp therefore their element categories to have a least element. \square

Remark 7.9. In both cases, terminal object, product and exponent in $TCpo(T)$ or $TCpo_{\perp}(T)$ are defined just in the same way as in $TPoset(T)$. The power object Ω^P differs from $TPoset(T)$ by the fact that it only contains pairs of the form $(u, X) \in T \times \mathcal{P}(P)$ were X is not only downward closed but also Scott closed. This follows from the fact that in both $TCpo(T)$ or $TCpo_{\perp}(T)$, subobjects correspond to Scott closed subsets of timed domains.

Theorem 7.10. *Both categories $TCpo(Cpo)$ of timed domain or $TCpo_{\perp}(Cpo_{\perp})$ of timed domain with least elements, and, in both case, Δ -synchronous function with continuous temporal projection (therefore themselves continuous) are cartesian closed categories.*

Proof. Since Cpo and, resp. Cpo_{\perp} are cartesian closed, it is routine to check that terminal objects, asynchronous products and asynchronous exponents both in $TCpo(Cpo)$ or in $TCpo_{\perp}(Cpo_{\perp})$ just as in $TPoset(Poset)$ (see Theorem 6.6). \square

Remark 7.11. Extending these categories with the empty timed domain over the empty time scale they both have an initial object. Then, it can be shown that both categories have co-products: the disjoint sum in $TCpo(Cpo)$ and the coalescent sum in $TCpo_{\perp}(Cpo_{\perp})$. In other words, both categories are bi-cartesian closed.

Theorem 7.12. *Given a graph G as in Theorem 6.9, given $\mathcal{C} \subseteq Cpo$ and $F : G \rightarrow TCpo(Cpo)$ (or $F : G \rightarrow TCpo_{\perp}(\mathcal{C})$). Then if the projection $H : G \rightarrow \mathcal{C}$ of G has a limit (resp. a colimit) then so does F and its limit (resp. colimit) is in $TCpo(\mathcal{C})$ (or in $TCpo_{\perp}(\mathcal{C})$) in \mathcal{C} .*

8 Timed fixpoints

We examined in this section the least fixpoints of homogeneous Δ -synchronous on timed domains. Observe that all synchronous functions have \perp as least fixpoint. This fact alone justifies the definition of the more general notion of Δ -synchronous functions.

Lemma 8.1. *Let $P \in TCpo_{\perp}(Cpo_{\perp})$ a timed domain with least element \perp_P timed over a complete time scale T with least element \perp_T . Let $f : P \rightarrow P$ be a Δ -synchronous function with continuous temporal projection $\delta : T \rightarrow T$. Then both least fixpoint $\mu_P(f) \in P$ and $\mu_T(\delta) \in T$ are defined. Moreover, we have $\pi \circ \mu_P(f) = \mu_T(\delta)$.*

Proof. By Scott continuity, we have $\mu_P(f) = \bigvee_{n \in \omega} f^n(\perp_P)$ and $\mu_T(\delta) = \bigvee_{n \in \omega} \delta^n(\perp_T)$. Moreover, by (SD1) and induction over $n \in \omega$, since $\pi(\perp_P) = \perp_T$ we easily prove that $\pi \circ f^n(\perp_P) = \delta^n(\perp_T)$ for all $n \in \omega$. By continuity of π (7.4) we have $\pi \circ \mu_P(f) = \bigvee_{n \in \omega} \delta^n(\perp_T)$ and thus $\pi \circ \mu_P(f) = \mu_T(\delta)$. \square

Theorem 8.2. *Let then P^P be the exponent of the object P by itself in $TCpo_{\perp}(Cpo_{\perp})$ and let T^T be the exponent of the object T by itself in Cpo_{\perp} . Then the least-fixpoint mapping $\mu_P : P^P \rightarrow P$ is a Δ -synchronous functions with continuous temporal projection $\pi(\mu_P) = \mu_T : T^T \rightarrow T$.*

Proof. The fact $\mu_P : P^P \rightarrow P$ is Δ -synchronous follows from the fact that μ_P is monotone (SD2) and, by Lemma 8.1, we have $\pi \circ \mu_P(f) = \mu_T \circ \pi(f)$ for every $f \in P^P$ (SD1). This means that $\pi(\mu_P) = \mu_T$. Finally, as it is a known fact from domain theory that $\mu_T : T^T \rightarrow T$ is continuous this proved that μ_P is indeed in $TCpo_{\perp}(TCpo_{\perp})$. \square

In other words, the computation of a least fixpoint computation can be performed on-the-fly.

Remark 8.3. Computing the fixpoint of a function f implicitly mean that the output of that function is sort of rewired on its input. In signal processing, there is a feedback loop. Since both inputs and outputs lay in the same timed domain, their temporal projections lay in the same time scale and can thus be compared. This eventually leads us to the notion of causality.

Definition 8.4 (Timed causality). Let P be a poset timed over T with least element $\perp_P \in P$ therefore also a least element $\perp_T \in T$. Let $f : P \rightarrow P$ be a Δ -synchronous function with temporal projection $\delta : T \rightarrow T$. We say that f is *globally causal* when $\pi(x) \leq \pi(f(x))$ for every $x \in P$. Equivalently, by (SD1), when $t \leq \delta(t)$ for all $t \in \pi(P)$.

Remark 8.5. In Theorem 8.2, quite surprisingly at first sight, we do not require functions to be causal. Of course, every Δ -synchronous function is in some sense *locally causal* since for each $n \in \omega$ given $x_n = f^n(\perp_P)$ we indeed have $\pi(x_n) \leq \pi(f(x_n))$.

Restricting to $TCpo_{\perp}(SLattice_{\perp})$ so that Δ -synchronous function preserves coherence, the above general fixpoints property fails since nothing ensures that μ_T is meet-preserving.

Theorem 8.6. *In the category $TCpo_{\perp}(CSLattice_{\perp})$ of domain timed over continuous meet-semilattices and Δ -synchronous functions with meet-preserving temporal projections, the fixpoint mapping $\mu_P : P^P \rightarrow P$ belongs to $TCpo_{\perp}(CSLattice_{\perp})$ provided we restrict ourselves to causal Δ -synchronous functions.*

Proof. Applying Theorem 8.2 it suffices to prove that, restricted to (self-synchronous) meet-preserving causal functions, the least fixpoint function $\mu T : T^T \rightarrow T$ is itself meet preserving.

For such a purpose, let $\delta_1, \delta_2 : T \rightarrow T$ be two causal meet-preserving functions. Remember that $\delta_1 \wedge \delta_2 : T \rightarrow T$ is simply defined by $(\delta_1 \wedge \delta_2)(u) = \delta_1(u) \wedge \delta_2(u)$ for every $u \in T$. Since both δ_1 and δ_2 are causal then $\delta_1 \wedge \delta_2$ is also causal therefore the restriction to causal functions still yields a meet-semilattice.

Since δ_2 is causal, we have $\delta_1 \leq \delta_2 \circ \delta_1$. But we also have $id \leq \delta_2$ hence, by monotony of δ_1 , we have $\delta_1 \leq \delta_1 \circ \delta_2$. Rephrased in a single line we this have proved that

$$\delta_1 \leq \delta_1 \circ \delta_2, \delta_2 \circ \delta_1 \quad (1)$$

By symmetry, we prove similarly that

$$\delta_2 \leq \delta_1 \circ \delta_2, \delta_2 \circ \delta_1 \quad (2)$$

Let then $A = \{1, 2\}$ seen as a two letter alphabet. Let A^* be the set of finite strings over A with ϵ the empty string. Let $\{\delta_w : T \rightarrow T\}_{w \in A^*}$ be the family of function inductively defined by $\delta_{\epsilon}(u) = u$, $\delta_{1w} = \delta_1 \circ \delta_w(u)$ and $\delta_{2w} = \delta_2 \circ \delta_w(u)$ for all $u \in T$ and all $w \in A^*$. For any string $w \in A^*$, let also denote by $|w|_1$ the number of 1 in w and by $|w|_2$ the number of 2 in w .

Let $n \in \mathbb{N}$. Since both δ_1 and δ_2 are meet preserving, we observe that we have

$$(\delta_1 \wedge \delta_2)^{2n} = \bigwedge_{w \in A^{2n}} \delta_w$$

Let then $w \in A^{2n}$. Two cases are possible. In the case $|w|_2 \leq |w|_1$, by applying $|w|_2$ times the inequality (1) we have $\delta_1^{|w|_1} \leq \delta_w$ hence, since δ_1 is causal and $n \leq |w|_1$ we have $\delta_1^n \leq \delta_w$. In the symmetric case $|w|_1 \leq |w|_2$, by applying $|w|_1$ times the inequality (2) we have $\delta_2^{|w|_2} \leq \delta_w$ hence, since δ_2 is causal and $n \leq |w|_2$ we have $\delta_2^n \leq \delta_w$. Applied to the above finite meet, since on of the two above cases must occur in each member, we have

$$\delta_1^n \wedge \delta_2^n \leq (\delta_1 \wedge \delta_2)^{2n}$$

As this holds for every $n \in \mathbb{N}$, applied to the minimal element \perp , we thus have

$$\bigvee_{n \in \mathbb{N}} \delta_1^n(\perp) \wedge \delta_2^n(\perp) \leq \bigvee_{n \in \mathbb{N}} (\delta_1 \wedge \delta_2)^{2n}(\perp)$$

Now, by continuity of the meet, we have $\mu(\delta_1) \wedge \mu(\delta_2)$ on the left of that inequality. Moreover, since the sequence $\{(\delta_1 \wedge \delta_2)^n(\perp)\}$ is monotonic it has the same greatest lower bound than its even subsequence, we thus have $\mu(\delta_1 \wedge \delta_2)$ on the right of the inequality. It follows that $\mu(\delta_1) \wedge \mu(\delta_2) \leq \mu(\delta_1 \wedge \delta_2)$.

By causality, $\mu(\delta_1) \wedge \mu(\delta_2)$ is a fixpoint of $\delta_1 \wedge \delta_2$ and $\mu(\delta_1 \wedge \delta_2)$ is its least fixpoint hence we also have $\mu(\delta_1 \wedge \delta_2) \leq \mu(\delta_1) \wedge \mu(\delta_2)$. It follows that $\mu(\delta_1) \wedge \mu(\delta_2) = \mu(\delta_1 \wedge \delta_2)$ which concludes the proof that μ is meet-preserving over non expanding functions. \square

Remark 8.7. In other words, quite surprisingly, the causality requirement that seems required for modeling purpose, is a sufficient restriction for computing fixpoints within $TCpo_{\perp}(SLattice_{\perp})$. This raises the question of how causality should be guaranteed. As Δ -synchronous functions are expected to change the underlying time scale, there seems to be no condition over these functions to ensure that, in fine, when eventually composed into a time homogeneous Δ -synchronous function, the resulting function would be causal.

Remark 8.8. Another approach to handle the non preservation of coherence is to assume that, when defining a Δ -synchronous function that do not preserve coherence, one knows the reason for this and it shall be accepted. One way to formalize this is to lift every Δ -synchronous function $f : P \rightarrow Q$ with temporal projection δ into a function $\Omega^f : \Omega^P \rightarrow \Omega^Q$ defined by $\Omega^f(u, X) = (\delta(u), \downarrow f(X))$ which now preserves coherence. In other words, thanks to the fact that $TCpo_{\perp}(T)$ is a topos for every time scale T , coherence preservation is not necessarily a drastic issue and semantics modeling can still be done within the most general category $TCpo_{\perp}(Cpo_{\perp})$.

9 Conclusion

Along these pages, we have briefly presented a possible mathematical framework for the modeling of spatio-temporal system behaviors that extends to space and time the classical notion of cpos and continuous functions used in denotational semantics.

Cartesian closed with internal fixpoint operators, with essentially all limits or co-limits that may exist in Cpo_{\perp} , the category $TCpo_{\perp}(Cpo_{\perp})$ looks like a fairly general and fully featured category for defining and analyzing the behaviors of timed programs with both synchronous and asynchronous versions of typical categorical constructs such as sum, products or exponents, as well as, in the synchronous case, powerobjects for modeling unbounded (synchronous) parallelism or, depending on the interpretation, unbounded non determinism.

Technically rooted in topos theory, via the element categories of sheaves over certain Grothendieck topologies, the resulting definitions and constructions have (mostly) been stated in elementary mathematical terms. This suggest that, after some more polishing and a lot more detailed application cases, such a material could also be taught to standard student in computer science and software engineering.

Observe also that we have not developed at all the potential offered by the left Kan extension theorem (see 4.11) that, in the case time scales are based on concrete numerical scales such a \mathbb{N} , \mathbb{Q}^+ or \mathbb{R}^+ , allows for defining sound lifting of operators over time scales such as delays, projections, stretches, etc..., into timed program constructs. In particular, this may allow developing and pursuing the research program initiated by Paul Hudak for an algebraic and programming theory of Polymorphic Temporal Media [17] and the somehow related though anterior proposal of Functional Reactive Programing [14, 15].

Although we have not yet defined any programming language associated to the proposed framework, most constructions convey effective program transformations or combinations. Following [11, 12] such a programming language would surely consists in including clocks and clock transformations into types. Also, since the temporal projection of a Δ -synchronous function can also be seen as a type modality that tells how the underlying time scales in a timed domain is altered by the function, this would certainly lead to generalizing the reactive and discrete modal type systems developed by Krishnaswami [24] and the related categorical models [19, 20].

Concerning numerical time scales, it must be mentioned that, there is also the (very) recent work of Schultz and Spivak [29] that proposes a Temporal Type Theory which develops quite in

the depth a notion of time domains over some notions of real intervals time scale. The potential links between their approach and the one proposed here need however to be investigated.

Since our time scales are not totally ordered, the choice of a time scale also specifies how data dependencies are defined and may evolve as time is passing. Some links with sessions types and beyond linear types [32], which allows specifying such evolving dependencies, could perhaps be developed. Links with polymorphic reactive programming language such as ReactiveML [27] should also be considered.

Last, it shall be clear that every timed semantic model has a strong flavor of operational semantics. However, such an operational semantics still need to be made explicit, especially in link with automata based timed operational semantics such as IO-timed automata theory [23].

Acknowledgment

The author wishes to express his deep gratitude to Gordon Plotkin et Phil Scott who both advise him to look at the notion of presheaves, to Marek Zawadowski for his help in understanding Grothendieck topologies and sheaves, and to Simon Archipoff, Michail Raskin and Bernard Serpette for many fruitful discussions on various aspects of this work.

References

- [1] J.-R. Abrial. *Modeling in Event-B - System and Software Engineering*. Cambridge University Press, Cambridge, 2010.
- [2] M. Barr and C. Wells. *Category Theory for Computing Science, 3rd edition*. Centre de Recherche Mathématique (CRM), Montréal, 1999.
- [3] H. Basold, M. M. Bonsangue, H. H. Hansen, and J. Rutten. (co)algebraic characterizations of signal flow graphs. In *Horizons of the Mind. A Tribute to Prakash Panangaden*, volume 8464 of *LNCS*. Springer, 2014.
- [4] A. Benveniste, P. Caspi, S. A. Edwards, P. Le Guernic N. Halbwegs, and R. de Simone. The synchronous languages twelve years later. *Proceedings of the IEEE*, 2002.
- [5] G. Berry. Stable models of typed lambda-calculi. In *Int. Col. on Aut., Lang. and Programming (ICALP)*, volume 62 of *LNCS*, pages 72–89. Springer, 1978.
- [6] G. Berry and G. Gonthier. The Esterel synchronous programming language: design, semantics, implementation. *Science of Computer Programming*, 19(2):87 – 152, 1992.
- [7] P. Caspi and M. Pouzet. Synchronous Kahn networks. In *Int. Conf. Func. Prog. (ICFP)*, pages 226–238, 1996.
- [8] G. L. Cattani, I. Stark, and G. Winskel. Presheaf models for the π -calculus. In *Category Theory and Computer Science (CTCT)*. Springer, 1997.
- [9] G. L. Cattani and G. Winskel. Presheaf models for CCS-like languages. *Theor. Comp. Sci.*, 300(1):47 – 89, 2003.
- [10] Daniel M. Chapiro. *Globally-Asynchronous Locally-Synchronous Systems*. PhD thesis, Department of Computer Science, Stanford University, 1985.
- [11] J.-L. Colaço, A. Girault, G. Hamon, and M. Pouzet. Towards a higher-order synchronous data-flow language. In *Int. Conf. On Embedded Software (EMSOFT)*, pages 230–239. ACM, 2004.
- [12] J.-L. Colaço and M. Pouzet. Clocks as first class abstract types. In *Int. Conf. On Embedded Software (EMSOFT)*, pages 134–155. ACM, 2003.
- [13] P. Cousot, R. Cousot, and L. Mauborgne. Logical abstract domains and interpretations. In S. Nanz, editor, *The Future of Software Engineering*, pages 48–71. Springer-Verlag, 2010.

- [14] C. Elliott and P. Hudak. Functional reactive animation. In *Int. Conf. Func. Prog. (ICFP)*. ACM, 1997.
- [15] C. M. Elliott. Push-pull functional reactive programming. In *Symp. on Haskell*, pages 25–36. ACM, 2009.
- [16] J.-Y. Girard. Linear logic. *Theor. Comp. Sci.*, 50:1–102, 1987.
- [17] P. Hudak. A sound and complete axiomatization of polymorphic temporal media. Technical Report RR-1259, Department of Computer Science, Yale University, 2008.
- [18] J. Hughes. Programming with arrows. In *Advanced Functional Programming, 5th International School, AFP 2004, Revised lecture*, volume 3622 of *Lecture Notes in Computer Science*, pages 73–129. Springer, 2005.
- [19] A. Jeffrey. Functional reactive types. In *IEEE Symp. on Logic in Computer Science (LICS)*. ACM, 2014.
- [20] W. Jeltsch. An abstract categorical semantics for functional reactive programming with processes. In *Workshop on Programming Languages Meets Program Verification*, pages 47–58. ACM, 2014.
- [21] G. Kahn. The semantics of simple language for parallel programming. In *IFIP Congress*, pages 471–475, 1974.
- [22] G. Kahn and G. D. Plotkin. Concrete domains. *Theor. Comp. Sci.*, 121(1&2):187–277, 1993.
- [23] D. K. Kaynar, N. Lynch, R. Segala, and F. Vaandrager. *The Theory of Timed I/O Automata*. Morgan & Claypool Publishers, 2006.
- [24] N. R. Krishnaswami. Higher-order functional reactive programming without spacetime leaks. In *Int. Conf. Func. Prog. (ICFP)*, 2013.
- [25] X. Liu and E. A. Lee. CPO semantics of timed interactive actor networks. *Theoretical Computer Science*, 409(1):110 – 125, 2008.
- [26] S. Mac Lane and I. Moerdijk. *Sheaves in Geometry and Logic*. Universitext. Springer, 1992.
- [27] L. Mandel and M. Pouzet. ReactiveML, a reactive extension to ML. In *Int. Symp. on Principles and Practice of Declarative Programming (PPDP)*. ACM, 2005.
- [28] E. Matsikoudis and E. A. Lee. The fixed-point theory of strictly causal functions. *Theor. Comp. Sci.*, 574:39–77, 2015.
- [29] P. Schultz and D. I. Spivak. Temporal type theory : A topos-theoretic approach to systems and behavior. Technical report, ArXiv:1710.10258, October 2017.
- [30] D. S. Scott. Data types as lattices. In *International Summer Institute and Logic Colloquium, Kiel*, volume 499 of *LNM*, pages 579–651. Springer, 1975.
- [31] R. de Simone, J.-P. Talpin, and D. Potop-Butucaru. The synchronous hypothesis and synchronous languages. In *Embedded Systems Handbook*. CRC Press, 2005.
- [32] P. Wadler. Propositions as sessions. *J. Funct. Program.*, 24(2-3):384–418, 2014.
- [33] G. Winskel. Event structures. In *Advances in Petri Nets*, volume 255 of *LNCS*, pages 325–392. Springer, 1987.
- [34] G. Winskel. Events, causality and symmetry. In *International Conference on Visions of Computer Science*, pages 111–128. BCS Learning & Development Ltd., 2008.