



HAL
open science

Spatio-temporal domains: an overview

David Janin

► **To cite this version:**

| David Janin. Spatio-temporal domains: an overview. 2018. hal-01634897v2

HAL Id: hal-01634897

<https://hal.science/hal-01634897v2>

Preprint submitted on 17 Jul 2018

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 and analysis of heterogeneous spatio-temporal computations as they occur more and more in modern computerized systems of systems. It appears that certain fibrations of posets into posets, called here spatio-temporal domains, eventually provide a fully featured category that extends to space and time the category of cpos and continuous functions, aka Scott Domains, used in classical denotational semantics.

1 Introduction

Research context. Program semantics is classically divided between two complementary approaches : *denotational semantics* and *operational semantics*. Denotational semantics generally refers to *what* the partial functions encoded by programs are : what is the relationship between (models of) their input values (or input memory state) and their output values (or output memory state). Operational semantics refers instead to *when* and *where* these values are read, transformed and eventually produced. To some extent, operational semantics defines effective models for implementing programs. It provides tools for analyzing the space and time behavior of programs therefore analyzing their complexity. On the other hand, denotational semantics provides instead methods for analyzing the (partial) correction of programs.

This suggests that semantic features can be distributed between typical operational features (time and space values) and denotational features (other data values). However, in many modern computerized systems such as, for instance, interactive music or animation systems [14,18,2,3], timing or spacing information plays a crucial rôle in the definition of system's inputs and outputs. There, many data values are implicitly parameterized by some space and/or time information: think of an augmented music system taking as input the melody played by a musician dancing on a stage. How the resulting spaced-and-timed signals can be read, combined and transformed in both an efficient and a sound way is one of the central questions of numerous domain specific language proposals such as, for instance, Fran [14] for animation or Euterpea [18] for music.

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

In all the underlying semantic models, compositionality is a key issue as it allows efficient, structure driven, development and analysis techniques. In fact, with compositionality, properties of complex systems/programs can be derived from certain combinations of the properties of their (simpler) components. As an immediate consequence, semantic models can also be studied and developed *per se*, adequate (domain specific) programming languages *deriving* a posteriori from the algebraic/combinatorial properties of these models.

Our contribution. Following such a model-driven development of programming language, we consider the possibility of lifting space or time information into typical denotational models: Scott Domains. It happens that this can be done by restricting certain constructions known in topos and fibration theory to posets. We thus provide in this paper an elementary description of these constructions and illustrate their applicability by interpreting the induced algebras in terms of typical spaced or timed programming constructs.

Detailed structure. Technically, we define spatio-temporal domain as certain discrete fibrations of posets into posets (Section 2). Simply said, elements of these domains are (partially ordered) computation histories indexed over space-time scales. Relationships with other known semantic models are detailed in Section 3. Spatio-temporal morphisms are defined in Section 4 as monotone functions between spatio-temporal domains that *uniformly* act on the underlying spacetime scales. This eventually yields a fully featured category that extends to space and time the categories of posets/cpos/domains and monotone/continuous functions typically used in classical denotational semantics.

The proposed approach yields two layers of program constructs that have been long identified in Globally-Asynchronous Locally-Synchronous (GALS) system design [10,30]. More precisely, there appear:

- (1) a *synchronous layer* of programming constructs available when components' inputs and outputs are located and timed on the same spacetime scale and received or produced in a synchronous way (Section 5),
- (2) an *asynchronous layer* of programming constructs available when component's inputs and outputs are located and timed on possibly distinct spacetime scales (Section 6).

At the border of these two layers, when inputs and outputs are located on the same scale but without any synchronicity assumption, we show that feedback loop constructs are available for defining non trivial (least) fixpoints (Section 8), that is, infinite signals. Continuously spaced and timed posets are also studied in Section 7 where a tight relationship with various key concepts from Domain Theory is established.

It is known that fibration theory [29] is used in denotational semantics of higher-order dependently typed lambda calculus [20]. Restricting ourselves to discrete fibrations over posets yields a notion of types that weakly depends on spacetime scales (or clocks). These possibilities were already studied in language extension proposals [12,11] of the Synchronous Programming Language fam-

ily [7,5]. Our approach provides a sound mathematical framework for the formal study of these possible extensions.

Although we aim at providing elementary descriptions of all concepts, the reader is still expected to know the (basic) definitions of a partially ordered set (poset), a category and the associated notions of (possibly continuous) functors. All other concepts are defined in the text when needed. Concrete examples defined over spaced and timed signals are detailed throughout, illustrating most concepts and their applicability to spacetime program's semantics.

Some notations. A partially ordered set, or *poset*, is a set P equipped with a partial order relation, that is, a reflexive, symmetric and transitive binary relation. Given a poset P , we use the notation $x, y \leq z$ (resp. $z \leq x, y$) or even $X \leq z$ (resp. $z \leq X$) 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 it is non empty and 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 called *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 directed subset $X \subseteq P$, if $\bigvee X$ 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 every 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 have $\mu(f) = \bigvee_{n \in \omega} f^n(\perp)$.

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 (possibly with non strict functions) 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. Details about these category theoretical concepts are given in the text when needed.

2 Timed posets

Throughout this text, an arbitrary poset T may be interpreted as a *spatio-temporal scale*. This means that for every element $u, v \in T$, we say that u lies *before* v , or v lays *after* u , when $u \leq v$, and that u lays *beside* v when u and v are incomparable. In other words, there is the *time dimension* that increases with the order, and, there is the *space dimension* where incomparable elements are also related.

Example 2.1 (N -distributed spacetime). As an example, let N be a set of mobile agents, each with its own clock timed over, say, the positive real line \mathbb{R}^+ . As a variant of message timestamps in distributed computing, one can define a *spacetime location* as any pair (S, C) with $S \subseteq N$ a non empty set of *synchronized* agents and $C : N \rightarrow \mathbb{R}^+$ their *shared knowledge* of all clocks. This interpretation yields a partial order defined by $(S', C') < (S, C)$ when $C' \leq C$ (point-wise) and $C'(n) < C(n)$ for all $n \in S$. Then it can be shown that every sub-poset of space time locations induces a canonical communication scenario between the agents that justifies the order between these locations.

More precisely, let T_N be the poset of spacetime locations induced by N . We assume the following communication model between agents. There may be *two-way synchronizations* among a group of agent $S \subseteq N$ when these agents are, say, in the same space location. In this case, they share all their knowledge of local clock values, taking the max of clock values of unsynchronized agents. There may also be *one-way messages* sent by one agent, with a copy of its own local clock value, to (any member of) a synchronized group of agents.

Then it is an easy observation that any discrete sub-poset $X \subseteq T_N$ of spacetime locations induces a canonical communication scenario define in the obvious way for two-way synchronization and as follows for one-way messages. For every spacetime location $(S, C) \in X$, for every $n \notin S$, either there is $(S', C') \leq (S, C)$ such that $C'(n) = C(n)$, or there is a one-way message from n to S sent from n at (local) instant $C(n)$ and received by one/any $m \in S$ at (local) instant $C(m)$.

An example of such a communication scenario is illustrated in Figure 1 in the case of $N = \{1, 2, 3, 4\}$. The selected subset of spacetime locations is detailed on

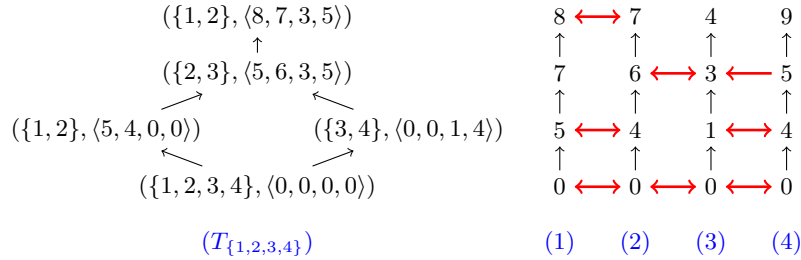


Fig. 1. A communication scenario.

the left of the figure, with functions from N into \mathbb{R}^+ represented by vectors of length four. The resulting communication scenario is detailed on the right, with two-way (resp. one-way) transversal arrows modeling two-way synchronizations (resp. one-way messages). Arrows resulting from transitivity of two-way synchronizations have been omitted.

One can observe that such a model extends to arbitrary collection of posets taken as (local) time scales since local time scales are not required to be compatible in any sense.

Having said so, throughout the rest of this presentation, a spatio-temporal scale is simply referred to as a *time scale* and its elements are simply called *instants*.

Definition 2.2 (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 $t \leq \pi(x)$ there is a unique element $x \downarrow t \in P$ such that $\pi(x \downarrow t) = t$ and $x \downarrow t \leq x$,

for all $x, y \in P$. The element $x \downarrow t$ is called the *temporal cut*, or simply the *cut* of x at instant t . This situation is depicted in (1). The cut function $(x, t) \mapsto x \downarrow t$ can be seen as sort of a (partial) action of the poset T over the poset P .

$$\begin{array}{ccc}
 x \xrightarrow{\leq} y & & x \downarrow t \xrightarrow{\leq} x \\
 \pi \downarrow \quad \text{(IN1)} \quad \downarrow \pi & & \exists! \pi \downarrow \quad \text{(IN2)} \quad \downarrow \pi \\
 \pi(x) \xrightarrow{\leq} \pi(y) & & t \xrightarrow{\leq} \pi(x)
 \end{array} \tag{1}$$

Property (IN1) states that temporal projections are poset *functors*. Property (IN2) states that these functors are *discrete fibrations*¹.

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 a set of event values 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 $e = \perp$.

¹ Discrete fibrations and the related notion of categories of elements of a functor on *Set* are presented in [4] (Section 12.2) or in [27] (Section 1.5).

An element $(t, e) \in \text{Imp}(T, E)$ with $e \neq \perp$ is a *timed impulse* with value e at instant t . Observe there are no elements strictly above (or after) such an impulse and all elements strictly below (or before) bear no value as they are of the form (t', \perp) for some $t' < t$. Rephrased in terms of temporal cut, for every element $(t, e) \in \text{Imp}(T, E)$ and instant $t' \leq t$, 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). With T and E as above, 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 . Then a *partial timed signal* is 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 $\text{Sig}(T, E)$ 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 $(u, S), (u', S') \in \text{Sig}(T, E)$. Following [28], a pair $(u, S) \in \text{Sig}(T, E)$ represents the observation of a timed signal 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 every partial signal $(u, S) \in \text{Sig}(T, E)$ and instant $v \leq u = \pi(u, S)$ we have $(u, S) \downarrow v = (v, S \downarrow v)$ with $S \downarrow v = \{(t, e) \in S : t \leq v\}$. Such a temporal cut $(u, S) \downarrow v$ models the observation of the partial signal (u, S) until instant v that lies before instant 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 $\text{cut}_x : \downarrow \pi(x) \rightarrow P$, called *local cut*, defined by $\text{cut}_x(u) = x \downarrow u$ for every $u \in T$ such that $u \leq \pi(x)$.

Remark 2.8. In the definition above and later in the text, we use the notation $\downarrow X \subseteq P$ for the *downward closure* of a subset $X \subseteq P$ of a poset P defined by $\downarrow X = \{y \in P : \exists x \in X, y \leq x\}$ that extends to an element $x \in P$ by putting $\downarrow x = \{y \in P : y \leq x\}$. At first sight, this notation may clash with our chosen notation for cuts. However, 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 extent the (partially defined) cut point-wise. It follows that we can even write $\downarrow x \downarrow u$ without any parenthesis with no 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 $X \downarrow x = \{y \in X : y \leq x\}$. Then we have $\downarrow x = P \downarrow x$.

Lemma 2.9 (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 $\downarrow x \subseteq P$ and $\downarrow \pi(x) \subseteq T$ are isomorphic posets as shown by the restriction of the cut to $\downarrow x$ and, as inverse, the temporal projection.

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 $\pi(x \downarrow v) = v$.

Corollary 2.10. *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 other hand, maximal elements in a timed poset can have arbitrary temporal projection as shown by timed impulse timed posets (see 2.5).*

3 Derived notions

We review below several notions that derive from the notion of timed posets and that allow this notion to be related with other known concepts appearing in computer system modeling or programming language semantics.

Temporal coherence. The following notion is inspired by Girard's notion of coherent space in linear logic [16]. A similar notion, even more closely related with timed domains, also appears in Winskel's notion of event structures [31].

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 $x \leq y$.

In other words, coherence offers 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 interpreted as the *trace* of a spatio-temporal computation that has been observed at the instants $\pi(X)$. The (sub)posets X and $\pi(X)$ are even isomorphic via the restriction of π to X and the downward closure $\downarrow X$ of a coherent subset is itself coherent.

Remark 3.4. The structure $(P, \leq, \not\prec)$ defined for every timed poset P is a special case of an event structure [31], though dropping the finite history requirement. The same computational interpretations are possible. 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 other hand, two incoherent elements x and y in P describe *conflicting* computations: they cannot appear in the same computation trace. More precisely, by the definition of coherence, when $x \not\prec y$ 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 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 $x, y \in P$, essentially is a *symmetry*² as recently defined by Winskel in event structures [32].

This observation relates timed posets even more closely 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 a given as part of the definition of that event structure.

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 [28,26,25]. The following definition shows that timed posets also induce such distances over their elements.

Definition 3.6. 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 itself as least element. The *distance* induced by π over P is the function

$$d : P \times P \rightarrow \mathcal{P}^\downarrow(T) \quad \text{defined by} \quad d(x, y) = \{t \in T : t \leq \pi(x), \pi(y), x \downarrow t = y \downarrow t\}$$

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 3.7. *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),

² Strictly speaking, for \sim_π to be a symmetry, the timed poset P must be completed with sorts of “passing time” 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 with $\pi(y) = u$.

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 $u \leq \pi(z), \pi(y)$ hence $x \downarrow u = y \downarrow u$ with $u \leq \pi(x), \pi(y)$ that is $u \in d(x, y)$.

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 and 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 $F : T^{op} \rightarrow Set$ from the category T^{op} , obtained from T by reversing the order relation, into the category Set of sets and functions. Then the following lemmas state that posets timed over T are isomorphic with what are known in category theory as the *categories of elements* of presheaves over T . This shows that timed posets and presheaves are essentially equivalent notions, an equivalence later stated as a categorical equivalence (see 4.10 below).

Lemma 3.8. *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 category of elements 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).

Lemma 3.9. *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 its category of elements 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 category of elements of F_P is also straightforward.

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 their underlying time scales. Somehow generalizing the approach of Colaço et al. [12,11], with timed posets interpreted as timed types, this uniformity requirement allows timed morphisms types to be defined not only as the domains and codomains of these morphisms, as in the simply-typed definition of function types, but also as the uniform transformation of time scales they induce.

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 $f : P \rightarrow Q$ together with a *monotone function* $\delta : U \rightarrow V$, called the *temporal projection* of f , is Δ -synchronous when, as depicted in (2):

- (SD1) $\pi(f(x)) = \delta \circ \pi(x)$,
- (SD2) if $x \leq y$ then $f(x) \leq f(y)$,
- (SD3) $f(x \downarrow u) = f(x) \downarrow \delta(u)$,

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

$$\begin{array}{ccc}
 U & \xleftarrow{\pi_P} & P \\
 \delta \downarrow & \text{(SD1)} & \downarrow f \\
 V & \xleftarrow{\pi_Q} & Q
 \end{array}
 \qquad
 \begin{array}{ccc}
 P & \xrightarrow{\leq_P} & P \\
 f \downarrow & \text{(SD2)} & \downarrow f \\
 Q & \xrightarrow{\leq_Q} & Q
 \end{array}
 \tag{2}$$

Remark 4.2. As soon a $u \leq \pi(x)$ then, by monotonicity 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). Every monotone function $\delta : U \rightarrow V$ is a Δ -synchronous function between the self-timed posets U and V , with itself as temporal projection.

Lemma 4.4. *Under hypothesis (SD1), properties (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).

On coherence preservation. It is easy to see that the Δ -synchronous image of a coherent subset bounded above is coherent. With a view towards application, this coherence preservation property is probably enough. Nevertheless, we show that without the boundedness condition this is no longer true and we provide sufficient additional conditions on time scale changes for coherence preservation.

Example 4.5. Let $U = \{\perp, a, b\}$, self-timed, with minimum element \perp and a and b incomparable. Let $P = U$ ordered the same 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 $a \circ b$ in U we have $f(a) = a \not\leq b = f(b)$ in P . The function f does not preserve coherence.

Lemma 4.6. *Assume that both U and V are meet-semilattices. Let $\delta : U \rightarrow V$ be a 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 conditional meet, i.e. it is stable in the sense of Berry [6].*

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.

Corollary 4.7. *Let $\text{Coh}(P)$ and, resp. $\text{Coh}(Q)$ be the set of coherent subsets of P and, resp. of Q , ordered by inclusion. Then, equipped with the point-wise*

extension of their temporal projection, they both are posets timed over $\mathcal{P}(U)$ and $\mathcal{P}(V)$ respectively. Moreover, the function $f : \text{Coh}(P) \rightarrow \text{Coh}(Q)$ defined by extending f point-wise is Δ -synchronous, stable, and event linear in the sense of Girard [16], i.e. it is well-defined and we have $f(X \cap Y) = f(X) \cap f(Y)$ and $f(X \cup Y) = f(X) \cup f(Y)$ for all $X, Y \in \text{Coh}(P)$.

Remark 4.8. In example 4.5, 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* in the sense that their outputs are timed the same way as their inputs.

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) $\pi(f(x)) = \pi(x)$,
- (SI2) if $x \leq y$ then $f(x) \leq f(y)$,
- (SI3) $f(x \downarrow u) = f(x) \downarrow u$,

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

Theorem 4.10. *The category $\text{TPoset}(T)$ of posets timed over T and synchronous functions is equivalent to the category $\text{Psh}(T)$ of presheaves over T and natural transformations. More precisely, the functor $\varphi : \text{TPoset}(T) \rightarrow \text{Psh}(T)$ defined, for every timed poset $P \in \text{TPoset}(T)$, by $\varphi(P) = F_P$ (see 3.9) and, for all synchronous function $f : P \rightarrow Q$ in $\text{TPoset}(T)$, 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 $\psi : \text{Psh}(T) \rightarrow \text{TPoset}(T)$ defined, for all presheaf $F \in \text{Psh}(T)$, by $\psi(F) = P_F$ (see 3.8), and, for all natural transformation $\alpha : F \Rightarrow G$ in $\text{Psh}(T)$, by $\psi(\alpha) = f$ with $f(t, x) = (t, \alpha_t(x))$ for all $(t, x) \in P_F$.

This implies that the category $\text{TPoset}(T)$ is a Grothendieck topos. In the next section, we shall make explicit the constructions that prove it is, as a consequence, an elementary topos.

Δ -synchronous vs synchronous functions. We show that every time scale change δ induces a (simple) contravariant time scale change functor in such a way that every Δ -synchronous function f with temporal projection δ uniquely (and uniformly) factorizes into a synchronous function followed by a (simple) time scale change, this functor having a (less simple) left adjoint.

Theorem 4.11 (Left Kan extension). *Let $\delta : U \rightarrow V$ be a monotone function. Then there are two categorical functors $\delta^* : \text{TPoset}(V) \rightarrow \text{TPoset}(U)$ and $\delta_! : \text{TPoset}(U) \rightarrow \text{TPoset}(V)$ and, for every $P \in \text{TPoset}(U)$ and $Q \in \text{TPoset}(V)$,*

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

i.e. f uniquely factorizes through α_P or ω_Q . In particular, there is the categorical adjunction $\delta_! \dashv \delta^*$. The functor $\delta_!$ is the left Kan extension operation along δ . This situation is depicted in (3).

$$\begin{array}{ccccc}
 & & \delta_!(P) & \xrightarrow{\exists! f_!} & Q \\
 & & \uparrow \alpha_P & \nearrow f & \uparrow \omega_Q \\
 V & & P & \xrightarrow{\exists! f^*} & \delta^*(Q) \\
 \delta \uparrow & & & & \\
 U & & & &
 \end{array} \tag{3}$$

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 $\delta^*(h)(u, y) = (y, h(y))$ 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 $X \in P$.

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 $f_! \circ \alpha_P(y) = f(y)$.

Example 4.12 (Timed signals). Continuing examples on signals (see 2.6), with $P = \text{Sig}(U, A)$ and $Q = \text{Sig}(V, B)$, the “lower” part of the above theorem can be defined within a slight extension of the notion signals as follows. We define $\delta^*(Q) = \text{Sig}^*(\delta, B)$ as the set of all pairs $(u, Y) \in U \times \mathcal{P}(V \times B)$ such that $v \leq \delta(u)$ for all $(v, b) \in Y$, with the cut defined by

$$(u, Y) \downarrow u' = (u', \{(v, b) \in Y : v \leq \delta(u')\})$$

for all $(u, Y) \in \text{Sig}^*(\delta, B)$ and $u' \leq u$. Then there is the Δ -synchronous function $\omega_Q : \text{Sig}^*(\delta, B) \rightarrow \text{Sig}(V, B)$ defined by $\omega_Q(u, Y) = (\delta(u), Y)$ for all $(u, Y) \in \text{Sig}^*(\delta, B)$ and, for every Δ -synchronous $f : \text{Sig}(U, A) \rightarrow \text{Sig}(V, B)$ with projection δ , the synchronous function $f^* : \text{Sig}(U, A) \rightarrow \text{Sig}^*(\delta, B)$ defined for all $(u, X) \in \text{Sig}(U, A)$ by $f^*(u, X) = (u, Y)$ when $f(u, X) = (\delta(u), Y)$ that uniquely factorizes f as above.

For the “upper” part, one can define over $\text{Sig}(U, A)$ the least equivalence \simeq_δ such that $(u, X) \simeq_\delta (u', X')$ whenever there is $u'' \in U$ such that $u, u' \leq u''$, $\delta(u) = \delta(u')$ and $X \downarrow u' = X' \downarrow u$. Then, one can check if $(u, X) \simeq_\delta (u', X')$ then $f(u, X) = f(u', X')$ for any Δ -synchronous function f with temporal projection δ . This means that $\text{Sig}(U, A) / \simeq_\delta$ can be used for defining $\delta_!(Q)$. However, in general, it does not seem that such a quotient is itself embeddable into (a sort of) a signal timed poset as done for the “lower” part above.

Remark 4.13. In some sense, in the above example, the function ω_Q acts as a *scheduler* that plans, within the time scale U , the events that will be emitted within the time scale V . Somehow dually, in the case both $\delta_1(Q)$ and α_P are definable over timed signals, for instance when δ is injective, then it can be observed that function α_P acts as a *buffer* that delays events timed on U until they are necessary for computation. More general conditions under which such an interpretation makes sense can be defined within timed signals, but we failed yet to find any general enough to be worth being detailed.

Remark 4.14. 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 the behavior of a timed system to be analyzed at various granularity. There appear 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 $T\text{Poset}(T)$ of posets timed over T and synchronous functions is a Grothendieck topos (see 5.18) therefore an elementary topos. We review below the concrete constructions over timed posets that derive from such a result and, with a view towards system modeling, describe their fairly intuitive interpretation.

Lemma 5.1 (Clock ticks). *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 other words, the self-timed poset T is terminal in $T\text{Poset}(T)$.*

Remark 5.2. In some sense, the time scale T , seen as a self-timed poset, can be understood as a clock. Indeed, every synchronous function $c : T \rightarrow P$ defines 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 shown by any strict downward closed subset of T (see 2.4). In other words, timed posets may also contain timed constants in which evolution in time may stop at some instant as if they were 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 $(x, y) \in P \otimes Q$.

Example 5.4 (Timed signals). One can check that we have $\text{Sig}(T, A) \otimes \text{Sig}(T, B) \cong \text{Sig}(T, A \oplus B)$.

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 $h : RT \rightarrow P \otimes Q$ 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 2.

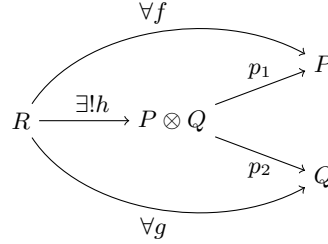


Fig. 2. Categorical product diagram.

This property is routine to check.

Remark 5.6. The interpretation of the above synchronous product shall be obvious. Thanks to the fact it is a categorical product, every pair of synchronous functions $f : R \rightarrow P$ and $g : R \rightarrow Q$ uniquely factorizes through some synchronous gluing $f \times g : R \rightarrow P \otimes Q$ of the functions f and g . Such a combinator over synchronous functions could be used as an arrow programming style [19].

Remark 5.7 (On coproduct). One can check that the coproduct³ $P \oplus Q$ of two timed posets with canonical injection eventually leads to the definition of categorical coproduct of P and Q in $TPoset(T)$. Then the empty timed poset is the initial object in $TPoset(T)$. This shows that the category $TPoset(T)$ is bi-cartesian.

Such a coproduct can be used as a timed alternative. However, such an alternative is very likely to be solved at initialization time, before any instant in T . Indeed, as soon as T has a minimum, every coherent subset of $P \oplus Q$ is necessarily either the embeddings of a coherent subsets of P or the embedding of a coherent subset of Q . No mixed subset is coherent.

Definition 5.8 (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 $Q \downarrow u$.

³ Possibly gluing minimal elements when considering the subcategory of timed posets with a minimum elements

Definition 5.9 (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 pairs (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 5.10. 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 $eval : R^Q \otimes Q \rightarrow R$ 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 $TPoset(T)$, 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 3.

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

Fig. 3. 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.

Remark 5.11. This result states that every synchronous function $f : P \rightarrow Q$ can itself be represented by coherent subset $\{f \downarrow u\}_{u \in T}$ of exponent Q^P which can be transmitted and applied *on-the-fly* over the pieces $\{x \downarrow u\}_{u \leq \pi(x)}$ of an argument $x \in P$. Then, as soon as the instant $\pi(x)$ is reached, such an on-the-fly application stops since the remaining values $\{f \downarrow u\}_{u \not\leq \pi(x)}$ of the functions cannot be synchronized with any further argument. This property could perhaps be used designing a timed programming language where resources are indeed freed whenever a timed (sub)computation terminates.

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 sub-objects⁴ of the category $TPoset(T)$ precisely as these downward closed subsets.

⁴ One can easily verify that the monomorphisms in $TPoset(T)$ are the injective synchronous functions. Then, as a consequence of the lemma, every injective syn-

Lemma 5.12 (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 $\pi(x) \leq \pi(y)$. 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 $x \leq y$.

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

Definition 5.13 (Synchronous power). Let P be a poset timed over T . The *synchronous power* of P 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 5.14. *The synchronous power Ω^P of a timed poset P is a poset timed over T and the power object of P in $TPoset(P)$.*

Proof. We first prove that Ω^T is the subobject classifier in $TPoset(T)$ as stated in Remark 5.17. 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 $f^{-1}(u, h) = (u, \{x \in P \downarrow u : h(x) = true(x)\})$ with *true* defined in Remark 5.17.

Example 5.15 (Timed signals). Continuing Examples 2.5 and 2.6 we have $\Omega^{Imp(T, E)} \cong Sig(T, E)$. Indeed, every timed signal $(u, X) \in Sig(T, E)$ is a collection of (non trivial) timed impulses arrived before or at instant u .

Remark 5.16. The above example illustrates how power objects can be interpreted in terms of parallelism: a computation trace in Ω^P models arbitrarily many computation traces in P that are run synchronously. Moreover, just like subsets, two sub-traces identical at some instant are eventually merged into a single one. Alternatively, it also makes sense to interpret traces in the power object as pending nondeterministic choices much like in power domain constructions.

chronous function $f : Q \rightarrow P$ is equivalent (as sub-object) with the inclusion synchronous function $inc_{f(Q)} : f(Q) \rightarrow P$.

Remark 5.17. Following topos theory [4,27], the timed domain $\Omega^T = \{(u, V) \in T \times \mathcal{P}(T) : V = \downarrow V \leq u\}$, with the synchronous function $true : T \rightarrow \Omega^T$ defined by $true(t) = (t, \downarrow t)$ for all $t \in T$ is the *subobject classifier* in $T\text{Poset}(T)$. In other words, for every poset P timed over T , for every downward closed subset $X \subseteq P$, there is a unique synchronous function $\mathcal{X}_X : P \rightarrow \Omega$, the *characteristic synchronous function* of X , such that, given the synchronous inclusion $inc_X : X \rightarrow P$ we have $\mathcal{X}_f \circ inc_X = true \circ \pi$ and this is a pullback square, i.e. for all synchronous $f : Q \rightarrow P$ such that $\mathcal{X}_f \circ f = true \circ \pi$ we necessarily have $f(Q) = X$ therefore f uniquely factorizes through inc_X .

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

Fig. 4. Subobject classifier pullback square.

As an immediate consequence of Theorem 4.10, or gathering the results stated in this section⁵, we have:

Theorem 5.18. *The category $T\text{Poset}(T)$ is an elementary topos, i.e. it is cartesian closed, finitely complete and has all powerobjects.*

6 More on Δ -synchronous functions

Clearly, timed posets identities are synchronous functions and every composition of two Δ -synchronous functions is itself Δ -synchronous with temporal projection the composition of their temporal projections. It follows that every choice of a category $\mathbf{C} \subseteq \text{Poset}$ of time scales and time scale transformations yields the category $T\text{Poset}(\mathbf{C})$ of posets timed over time scales in \mathbf{C} and Δ -synchronous functions with temporal projections that are morphisms in \mathbf{C} . In the most general case, one can choose for \mathbf{C} the category Poset .

Throughout the rest of the section, we assume that the chosen time scale category $\mathbf{C} \subseteq \text{Poset}$ is cartesian closed with terminal poset $\{*\} \in \mathbf{C}$. We also assume some time scales U and $V \in \mathbf{C}$ and some posets P and, resp., $Q \in T\text{Poset}(\mathbf{C})$ timed over U and, resp., V .

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 $T\text{Poset}(\mathbf{C})$.*

⁵ additionally proving that $T\text{Poset}(T)$ also has all equalizers, which is easy since they are essentially defined as in *Set*.

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 $U \times V \in \mathbf{C}$. 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 \mathbf{C} and $\pi(p_2) = p_2 : U \times V \rightarrow V$ in \mathbf{C} . Together, they form the categorical product of P and Q in $T\text{Poset}(\mathbf{C})$.*

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 in $V^U \in \mathbf{C}$ just as already defined in 4.1.

Lemma 6.5. *Then Q^P is a poset timed over $V^U \in \mathbf{C}$. 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 \mathbf{C} . Together, they form the exponent of Q by P in $T\text{Poset}(\mathbf{C})$.*

Gathering all the above lemmas, we have:

Theorem 6.6. *The category $T\text{Poset}(\mathbf{C})$ is cartesian closed whenever \mathbf{C} is.*

Remark 6.7. The empty poset timed over the empty time scale is the initial object. The disjoint sum is the coproduct in $T\text{Poset}(\mathbf{C})$ therefore $T\text{Poset}(\mathbf{C})$ is even bi-cartesian closed whenever \mathbf{C} is. The next theorem generalizes such a remark.

Definition 6.8 (Projection of diagram functor). Let $G = \langle V, E, s, t \rangle$ be a graph with vertices V , edges E , source and target functions $s, t : E \rightarrow V$. Let $F : G \rightarrow T\text{Poset}(\mathbf{C})$ be a diagram functor⁶. The temporal projection of F is defined to be the diagram functor $H : G \rightarrow \mathbf{C}$ 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 tranformation $H(e)$ is the temporal projection $\pi \circ F(e)$ of the Δ -synchronous function $F(e)$.

Theorem 6.9 (Limit and colimit). *A diagram functor $F : G \rightarrow T\text{Poset}(\mathbf{C})$ has a limit (resp. a co-limit) in $\text{Poset}(\mathbf{C})$ whenever its temporal projection $H : G \rightarrow \mathbf{C}$ has a limit (resp. a colimit) in \mathbf{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 $T \in \mathbf{C}$ and, for each $v \in V$, a function $\delta_v : T \rightarrow H(v)$ in \mathbf{C} . By definition, we have $\delta_{t(e)} = H(e) \circ \delta_{s(e)}$

⁶ We 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 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 cone given by $T \in \mathbf{C}$ and, for each $v \in V$, a function $\delta_v : H(v) \rightarrow T$ in \mathbf{C} . By definition, we have $\delta_{s(e)} = \delta_{t(e)} \circ H(e)$ for every $e \in E$. Then we can define

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

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

$$\iota_v(x) = (\delta_v \circ \pi_v(x), v, x)$$

for all $x \in F(v)$ and we have $P = \bigcup_{v \in V} \iota_v(F(v))$.

Let then \simeq be the least equivalence over P such that we have

$$\iota_{s(e)}(x) \simeq \iota_{t(e)}(F(e)(x))$$

for all $e \in E$ and $x \in F(s(e))$. 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 for all $x, y \in P$ by $[x] \leq [y]$ when there is $x' \in [x]$ such that $x' \leq y$ and the temporal projection defined for all $x \in P$ by $\pi([x]) = \pi(x)$.

Then P/\simeq is a poset timed over T and, for every $v \in V$, the function $j_v : F(v) \rightarrow P/\simeq$, defined by

$$j_v(x) = [\iota_v(x)]$$

for all $x \in F(v)$ is Δ -synchronous with temporal projection δ_v . By construction, we have $j_{s(e)} = j_{t(e)} \circ F(e)$ for every $e \in E$. Then, it is routine to check that P/\simeq equipped with the functions $\{j_v\}_{v \in V}$ form the colimit of F .

As a particular case, a fixpoint equation of timed posets (or, as defined below, timed domains) has an inductive (resp. co-inductive) solution whenever the projection of this equation over time scales has an inductive (resp. co-inductive) solution.

7 Timed domains

Timed domains ought to be cpos timed over cpos with continuous projections and cuts. Such a definition is formalized via the notion of pre-continuous timed posets so that timed domains can precisely be defined as pre-continuous timed posets timed over cpos.

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

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

for all $X \subseteq P$.

Remark 7.2. In general, neither $Imp(T, E)$ nor $Sig(T, E)$ (see 2.5 and 2.6) are pre-continuous. Indeed, as soon as there is $t = \bigvee U$ with directed $U \subseteq T$ and $t \notin U$, then, in $Imp(T, E)$, we have $\{(t, e) \downarrow u\}_{u \in U} = \{(t, \perp) \downarrow u\}_{u \in U}$ therefore this set as no upper bound. A remedy to this fact is proposed below.

Example 7.3 (Observable timed signals). Let \ll be the relation called here *way before*, defined for every instant $t, u \in T$ by $t \ll u$ when for every directed subset $U \subseteq X$ such that $\bigvee U = u$ there must exists $u' \in U$ such that $t \leq u'$. Let $Sig_C(T, E) \subseteq Sig(T, E)$ be the set of timed signals (u, X) such that $t \ll u$ for all $(t, e) \in X$, i.e. every event $(t, e) \in X$ is *observable* in the sense that it can be observed in any series of observations performed as any (directed) set of instants U such that $\bigvee U = u$. Then, with the cut defined by $(u, X) \downarrow u' = (u', \{(t, e) \in X : t \ll u'\})$ for all $u' \leq u$, the resulting set of signals $Sig_C(T, E)$ is a pre-continuous timed poset though not, in general, a sub-timed poset of $Sig(T, E)$.

Lemma 7.4. *Let P be a pre-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 $y = x \downarrow v$.

Remark 7.5. In general, the continuity of all local cuts 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 7.6. *Let P be a pre-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 pre-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 7.4) 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.

Definition 7.7 (Timed domain). A *timed domain* is a pre-continuous timed poset timed over a complete time scale, hence, as proved above also a complete poset with continuous local cuts and temporal projection.

Remark 7.8. As a consequence of 7.4, a timed domain P is continuous/algebraic in the sense of Scott if and only if T itself is continuous/algebraic. Indeed, it can be shown that for all $x, y \in P$, we have $x \ll y$ if and only if $\pi(x) \ll \pi(y)$.

Lemma 7.9 (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 $\bigvee f(X) = f(y)$.

Remark 7.10. As a special case of the above lemma every synchronous function between timed domains is continuous.

Theorem 7.11. *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 the Scott topology in terms of a Grothendieck topology J over posets in such a way that (the categories of

elements of the) sheaves in $Sh(T, J)$ are the pre-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 $J(t)$ to be the set of all downward closed subsets $U \subseteq T$ such that $U \leq t$, i.e. U is a sieve on t , and their Scott closure \bar{U} equals $\downarrow t$.

One can easily check that J is a Grothendieck topology (see [27] p 110). Moreover, one can also check that a timed poset $P \in TPoset(T)$ is pre-continuous if and only if its associated presheaf $F_P : T^{op} \rightarrow Set$ (see 3.9) is a sheaf for J (see [27] 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 every sheaf in $Sh(T, J')$ to be a singleton on \perp , and therefore its category of elements to have a least element. \square

Remark 7.12. The terminal object, products and exponentials in both $TCpo(T)$ and $TCpo_{\perp}(T)$ 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 $(u, X) \in T \times \mathcal{P}(P)$ where 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 7.13. *Both categories $TCpo(Cpo)$ of timed domains or $TCpo_{\perp}(Cpo_{\perp})$ of timed domains with least elements, and, in both cases, Δ -synchronous functions with continuous temporal projections (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 $TCpo_{\perp}(Cpo_{\perp})$ are defined just as in $TPoset(Poset)$ (see Theorem 6.6).

Remark 7.14. Both categories above have an initial object when extended with the empty timed domain over the empty time scale. They also have coproducts: the disjoint sum in $TCpo(Cpo)$ and the coalescent sum in $TCpo_{\perp}(Cpo_{\perp})$. In other words, both (slightly extended) categories are bi-cartesian closed. More generally, it can be shown that Theorem 6.9 still holds when restricted to the category $TCpo_{\perp}(Cpo_{\perp})$.

8 Timed fixpoints and causality

In the category $TCpo_{\perp}(Cpo_{\perp})$ of timed domains with least elements, every Δ -synchronous function has a least fixpoint. Following the footsteps of Matsikoudis and Lee [28], we examine below the property of the induced least fixpoint operators.

Lemma 8.1. *Let $P \in TCpo_{\perp}(Cpo_{\perp})$ be 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 fixpoints $\mu_P(f) \in P$ 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 (7.6) we have $\pi \circ \mu_P(f) = \bigvee_{n \in \omega} \delta^n(\perp_T)$ and thus $\pi \circ \mu_P(f) = \mu_T(\delta)$.

Example 8.2 (Timed signals). Let $T = \overline{\mathbb{R}^+}$ be the time scale of positive reals completed with a maximum element ∞ . Let $f : Sig(T, E) \rightarrow Sig(T, E)$ defined for all $(u, X) \in Sig(T, E)$ by $f(u, X) = (u + 2, \{(0, e_0)\} \cup \{(t + 2, e) : (t, e) \in X\})$ for some fixed event value $e_0 \in E$. Then f is Δ -synchronous with temporal projection $t \mapsto t + 2$ with ∞ as least fixpoint. Then we have the least fixpoint $\mu(f) = (\infty, \{(2 * n, e_0) : n \in \mathbb{N}\})$.

Theorem 8.3. *Let P^P be the exponent in $TCpo_{\perp}(Cpo_{\perp})$ of the timed domain P by itself 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 function 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 $TCpo_{\perp}(TCpo_{\perp})$.

Remark 8.4. When computing the fixpoint of a function f 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 lie in the same timed domain, their temporal projections lie in the same time scale and can thus be compared. By default, every Δ -synchronous function is *locally causal* in the sense that, by monotonicity, for every $n \in \omega$, with $x_n = f^n(\perp_P)$, we have $x_n \leq f(x_n)$ and $\pi(x_n) \leq \pi(f(x_n))$. This suffices for Theorem 8.3 to hold. However, intuition suggests that we could require these functions to be globally causal as defined and studied below.

Definition 8.5 (Global 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 $t \in \pi(P)$.

Though not needed in the general case, such a notion appears when restricting to $TCpo_{\perp}(SLattice_{\perp})$ so that Δ -synchronous function preserves arbitrary coherence (4.6). Indeed, in $SLattice_{\perp}$, the fixpoint function μ_T is not in general meet-preserving hence Theorem 8.3 fails.

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

Proof. Applying Theorem 8.3 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

$$\delta_1 \leq \delta_1 \circ \delta_2, \delta_2 \circ \delta_1 \quad (4)$$

By symmetry, we prove similarly that

$$\delta_2 \leq \delta_1 \circ \delta_2, \delta_2 \circ \delta_1 \quad (5)$$

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 (4) 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 (5) 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.

Remark 8.7. Despite such a result, global causality remains quite an ad hoc restriction. We are still in need of some additional restrictions on the notion of Δ -synchronous functions in a category $TCpo_{\perp}(\mathbf{C}_{\perp})$ that would guarantee their combinations to be globally causal whenever applicable. For such a purpose, adjunctions in posets could be a direction to investigate.

9 Conclusion

Along these pages, we have detailed 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.

Bi-cartesian closed with internal fixpoint operators, and essentially all limits or co-limits that may exist in Cpo_{\perp} , the category $TCpo_{\perp}(Cpo_{\perp})$ eventually turned out to be 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, power-objects.

Technically rooted in topos and fibration theory, via the category of elements of sheaves over certain Grothendieck topologies, the resulting definitions and constructions have (mostly) been stated in elementary mathematical terms. This means that, after some more polishing and more detailed application studies, such material could even be taught to standard students in computer science and software engineering.

With a view towards concrete applications, we have not yet developed at all the potential offered by the left Kan extension theorem (see 4.11 and 4.12). When time scales are built from concrete numerical scales such as \mathbb{N} , \mathbb{Q}^+ or \mathbb{R}^+ , it allows us to define sound lifting of operators over these time scales such as delays, projections, stretches, etc., into timed program constructs. This would lead to pursuing the research program initiated by Paul Hudak for an algebraic and programming theory of Polymorphic Temporal Media [17] and the somehow related though earlier proposal of Functional Reactive Programming [14,15]. Links with the related ultrametric [24] or categorical models [21,22] could be investigated.

Examples over timed signals detailed throughout suggest that timed domains also may induce some notion of timed operational semantics, probably deeply linked with the existing state based timed system modeling frameworks such as IO-timed automata theory [23]. This surely necessitates focussing our attention on finite (or finitely representable) spaced and timed functions, a necessity that may benefit from our somewhat strong restriction to timed behaviors that uniformly act on the underlying time scales.

Acknowledgment

The author wishes to express his deep gratitude to Gordon Plotkin et Phil Scott for their early advice to look at the notion of presheaves, to Marek Zawadowski for his help in understanding Grothendieck topologies and sheaves, to referees for their numerous suggestions of improvement, 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. S. Archipoff and D. Janin. Structured reactive programming with polymorphic temporal tiles. In *ACM Work. on Functional Art, Music, Modeling and Design (FARM)*, pages 29–40. ACM Press, 2016.
3. S. Archipoff and D. Janin. Unified media programming: An algebraic approach. In *ACM Work. on Functional Art, Music, Modeling and Design (FARM)*, pages 36–47. ACM, 2017.
4. M. Barr and C. Wells. *Category Theory for Computing Science, 3rd edition*. Centre de Recherche Mathématique (CRM), Montréal, 1999.
5. A. Benveniste, P. Caspi, S. A. Edwards, P. Le Guernic N. Halbwachs, and R. de Simone. The synchronous languages twelve years later. *Proceedings of the IEEE*, 2002.
6. 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.
7. G. Berry and G. Gonthier. The Esterel synchronous programming language: design, semantics, implementation. *Science of Computer Programming*, 19(2):87 – 152, 1992.
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. P. Hudak. *The Haskell School of Music : From signals to Symphonies*. Yale University, Department of Computer Science, 2013.
19. J. Hughes. Programming with arrows. In *Advanced Functional Programming (AFP)*, volume 3622 of *LNCS*, pages 73–129. Springer, 2005.
20. B. Jacobs. *Categorical Logic and Type Theory*. Number 141 in Studies in Logic and the Foundations of Mathematics. North Holland, Amsterdam, 1999.
21. A. Jeffrey. Functional reactive types. In *IEEE Symp. on Logic in Computer Science (LICS)*. ACM, 2014.
22. 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.
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 space-time leaks. In *Int. Conf. Func. Prog. (ICFP)*, 2013.
25. N. R. Krishnaswami and N. Benton. Ultrametric semantics of reactive programs. In *IEEE Symp. on Logic in Computer Science (LICS)*, 2011.
26. X. Liu and E. A. Lee. CPO semantics of timed interactive actor networks. *Theoretical Computer Science*, 409(1):110 – 125, 2008.
27. S. Mac Lane and I. Moerdijk. *Sheaves in Geometry and Logic*. Universitext. Springer, 1992.
28. E. Matsikoudis and E. A. Lee. The fixed-point theory of strictly causal functions. *Theor. Comp. Sci.*, 574:39–77, 2015.
29. T. Streicher. Fibred categories à la Jean Bénabou. Revised notes of a course on fibred categories given at a Spring School in Munich 1999, 2014.
30. P. Teehan, M. R. Greenstreet, and G. G. Lemieux. A survey and taxonomy of GALS design styles. *IEEE Design & Test of Computers*, 24(5):418–428, 2007.
31. G. Winskel. Event structures. In *Advances in Petri Nets*, volume 255 of *LNCS*, pages 325–392. Springer, 1987.
32. G. Winskel. Events, causality and symmetry. In *International Conference on Visions of Computer Science*, pages 111–128. BCS Learning & Development Ltd., 2008.