Synchronous Structures ⋆

David Nowak

Research Center for Information Security (RCIS)
National Institute of Advanced Industrial Science and Technology (AIST)
Akihabara Dai Biru, Room 1102
1-18-13 Sotokanda, Chiyoda-ku, Tokyo 101-0021, Japan

Abstract

Synchronous languages have been designed to ease the development of reactive systems, by providing a methodological framework for assisting system designers from the early stages of requirement specifications to the final stages of code generation or circuit production. Synchronous languages enable a very high-level specification and an extremely modular design of complex reactive systems by structural decomposition of them into elementary processes. We define an order-theoretical model that gives a unified mathematical formalisation of all the above aspects of the synchronous methodology and characterises the essentials of the synchronous paradigm.

Key words: preorder semantics, reactive system, synchronous programming

1 Introduction

The Synchronous Paradigm Synchronous languages, such as Esterel [2], Lustre [3] and Signal [4] have been designed to ease the development of reactive systems. The synchronous hypothesis provides a deterministic notion of concurrency where operations and communications are instantaneous. In a synchronous language, concurrency is meant as a logical

⋆ A preliminary version of this article was presented at the 10^{th} International Conference on Concurrency Theory (CONCUR'99), Eindhoven, The Netherlands, August 1999 [1]. This work was partially supported by Action Coopérative INRIA PRESYSA. This work was done while the author was affiliated with Institut de Recherche en Informatique et Systèmes Aléatoires (IRISA), Rennes, France, and Oxford University Computing Laboratory, United Kingdom.

way to decompose the description of a system into a set of elementary communicating processes. Interaction between concurrent components is conceptually performed by broadcasting events. Synchronous languages enable a very high-level specification and an extremely modular design of complex reactive systems by structurally decomposing them into elementary processes. The use of synchronous languages provides a methodological framework for assisting the users from the early stages of requirement specifications to the final stages of code generation or circuit production while obeying compliance to expressed and implied safety requirements. In that context, the synchronous language SIGNAL is particularly interesting, in that it allows the specification of (early) relational properties of systems which can then be progressively refined in order to obtain an executable specification. All the stages of this design process can easily be modelled and understood in isolation. The purpose of this article is to define the mathematical model of *synchronous structures* which gives a unified formalisation of all the aspects of a synchronous methodology and which contains each of them in isolation.

Related Work There are several ways to characterise the essentials of the synchronous paradigm. In [5], we introduce a co-inductive semantics of SIGNAL, and a library of theorems is developed in the proof assistant COQ [6]. But it is not expressive enough to deal with dependencies. The semantics of a synchronous language can be described in a better way with Symbolic Transition Systems (STSs) [7]. This is a formalism on which fundamental questions can be investigated. But it treats the absence of a signal as a special value. This is not consistent with reality: The presence or the absence of a signal, relatively to another signal, has to be inferred by the program (endochrony [8]). In [8], STSs are extended with preorders and partial orders to model causality relations, schedulings and communications. This preorder-theoretic model is put into practice in the design of BDL [9], a synchronous specification language that uses families of preorders to specify systems. In [10], the problem of characterising synchrony without using a special symbol for absence is addressed in terms of multiple input-output sequential machines. In [11], the language SIGNAL is modelled in interaction categories [12] where morphisms are processes and objects are types of processes.

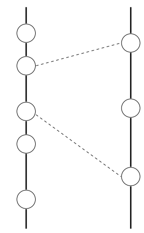
Motivations In 1545, the great Italian mathematician Gerolamo Cardano wrote "Ars Magna" [13], an important and influential treatise on Algebra in which the first complete expression for the solution of a general cubic equation was put forward. Cardano noticed that, in the case of some equation with three real solutions, he was forced to take at a certain stage the square root of a negative number. The imaginary numbers were born. Analogically, we generalise the classical notion of signal [4,8,14] with imaginary signals. Indeed, imaginary signals are obtained by closure under deterministic merge.

This extension has no material counterpart. It is used to compute intermediate results. For instance, in the original trace semantics [15,16], the temporal abstractions of signals (called clocks) have necessary a greatest lower bound but do not always have a (real) least upper bound. In that case, we need to define an imaginary least upper bound. Our model allows to extend the notion of classical clocks with imaginary clocks and to define a boolean lattice of clocks. Indeed, the closure of signals under deterministic merge induces the closure of clocks under least upper bound. In this lattice-theoretical model, temporal relations between signals always have a solution. If the solution contains imaginary signals, it means that the system has no real solution in the classical model and that it does not thus form an executable specification. Imaginary signals and clocks are in fact well suited to model the non-determinism that can be present in a SIGNAL specification. Our model does not treat the absence of a signal as a special value. It is consistent with reality where the presence or the absence of a signal, relatively to another signal, has to be inferred by the program (endochrony [8]). Indeed, it is impossible to check the absence of a signal: Suppose that an input signal is absent. If it comes from a register, then this one will provide its previous value. If it comes from a sensor, then this one will provide a random value. Moreover, synchronous structures can deal elegantly with data dependence and refinement of synchronous specifications.

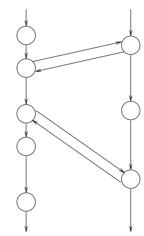
Outline First, in Section 2, we abstract the notion of control dependence in a mathematical structure that we call a synchronous structure in which we define the notions of imaginary signal and clock. Then, in Section 3, we prove that signals and their morphisms define a Cartesian closed category that we can relate with the category of event structures. In Section 4, we extend synchronous structures to deal elegantly with data dependence, with temporal refinement, and with the delay operator of synchronous languages. Finally, in Section 5, we give the semantics of SIGNAL in our model.

2 Synchronous Structure

In this section, we focus on a characterisation of control dependencies i.e. the temporal relations between events or the dates of events relative to some reference of time, not the value of events. Let us informally depict a synchronisation scenario between two sequences of events (i.e. sets of totally ordered events):



They exchange (dotted) synchronisation messages using an asynchronous medium for their communications. This involves a synchronisation relation between events. The natural structure of time of the whole system is that of a preorder. This preorder can be understood as the transitive closure of the union of the temporal relation (an order relation) and the synchronisation relation (an equivalence relation). Then, the example becomes:



In this section, we will formalise the notions involved in this example.

2.1 Synchronous Structure

We define a synchronous structure as a set of events plus a single preorder relation which summarises both the notion of synchrony (induced equivalence) and temporal causality (underlying partial order).

Definition 1 *$(\mathcal{E}, \ll)$ is a synchronous structure if and only if $\mathcal{E}$ is a non-empty set (of events) and $\ll$ is a preorder on $\mathcal{E}$ such that:*

$$\forall x \in \mathcal{E} \cdot \{y \in \mathcal{E} \mid y \leq x\} \text{ is finite, where } \begin{aligned} x \sim y &\Leftrightarrow_{def} x \ll y \wedge y \ll x \\ x < y &\Leftrightarrow_{def} x \ll y \wedge x \not\sim y \\ x \leq y &\Leftrightarrow_{def} x < y \vee x = y \end{aligned}$$

Intuitively, $x \sim y$ means that x and y are synchronous, that is to say the events x and y must occur simultaneously. The partial order $\leq$ is the temporal causality between events: $x \leq y$ states that x must occur before y. For instance, Figure 1 depicts 8 events which define a synchronous structure. To give easier explanations, the events are numbered from 1 to 8. Dotted lines represent the equivalence relation $\sim$ and bold lines represent the strict order relation $<$ as

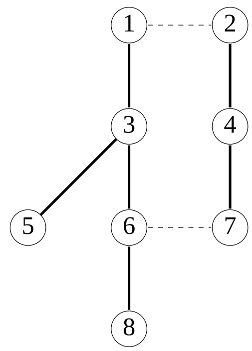
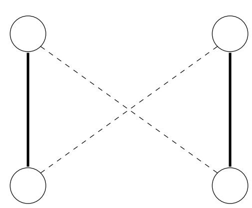


Fig. 1. An example of synchronous structure

a Hasse diagram: $x < y$ if and only if there is a sequence of connected bold line segments moving downwards from x to y.

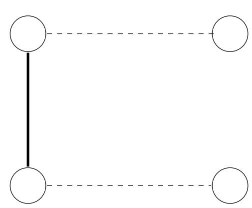
The preorder $\ll$ combines the synchronicity relation and the temporal causality relation. It defines a notion of time for the whole system. The use of a preorder eliminates bad cases.

First, two synchronisations cannot cross each other:

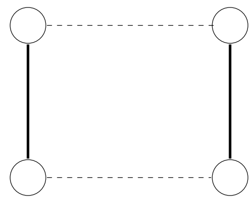


Indeed, by transitivity of the preorder, events are all synchronous in this case:

Second, the necessary temporal relation are induced. Suppose that we have the following set of events:



By transitivity of the preorder, we deduce a missing temporal relation:



More generally, the following property (1) comes directly from the definition of a synchronous structure:

$$\forall x, y_1, y_2, z \in \mathcal{E} \quad \cdot \quad x \ll y_1 \wedge y_1 < y_2 \wedge y_2 \ll z \quad \Rightarrow \quad \neg x \sim z \tag{1}$$

In Figure 1, it is thus guaranteed, for example, that the events numbered 1 and 8 cannot be synchronous.

We say that an event x is covered by an event y, and write $x \prec\!\!\!\cdot\, y$, if and only if $x < y$ and there is no event z satisfying $x < z < y$. From the fact that $\leq$ is well founded, we can deduce the following property:

$$\forall x, y \in \mathcal{E} \quad \cdot \quad x < y \quad \Rightarrow \quad \exists z \in \mathcal{E}, z \prec\!\!\!\cdot\, y \tag{2}$$

Indeed, $(\mathcal{E}, \leq)$ is not dense because $\leq$ is well founded. This property is important to guarantee a discrete model of synchronous programming.

2.2 Signal

Usually, a (real) signal is a totally ordered set of events. This total order implies that two different events cannot be synchronous. We generalise this definition to enable partially ordered sets of events to be (imaginary) signals. A signal just have to satisfy the property that two different events cannot be synchronous. We use this relaxed condition to define internal operations.

Definition 2 *Let X be a subset of $\mathcal{E}$. X is a signal if and only if it satisfies the following axiom:*

$$\forall x, y \in X \quad \cdot \quad x \sim y \quad \Rightarrow \quad x = y \tag{3}$$

Let X be a signal. From (3) we deduce that $\ll$ is antisymmetric on X and then is a partial order on X.

Let $\mathcal{S}_\mathcal{E}$ be the set of signals. For instance, in Figure 1, $\{1, 3, 5, 8\}$ and $\{2, 6, 8\}$ are in $\mathcal{S}_\mathcal{E}$. A real signal is then a particular case of signal which is totally ordered by $\ll$. For instance, in Figure 1, $\{1, 3, 5\}$, $\{2, 6, 8\}$ and $\emptyset$ are real signals

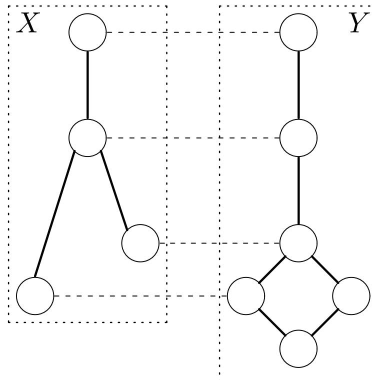


Fig. 2. Preordered signals ($X \preceq Y$)

but not $\{1, 3, 5, 8\}$. An imaginary signal is a signal that is not a real signal. With internal operations on signals, we will see that an imaginary signal enables to represent the lack of synchronisation constraints in an underspecified reactive system. A subspecification is a correct specification with interleaved events, and therefore cannot be executed because its scheduling is not fully determined. It needs to be composed with another specification to remove the non-determinism.

The property (2) is also true in a signal. It means that between two events of a signal there is only a finite numbers of events.

We define a preorder $\preceq$ on $\mathcal{S}_{\mathcal{E}}$. A signal X precedes a signal Y if and only if for any event of X there exists a synchronous event of Y (see, for instance, Figure 2). For all signals X and Y,

$$X \preceq Y \quad \Leftrightarrow_{\text{def}} \quad \forall x \in X \;\cdot\; \exists y \in Y \;\cdot\; x \sim y$$

This preorder gives rise to an equivalence relation $\widehat{=}$. For all signals X and Y,

$$X \widehat{=} Y \quad \Leftrightarrow_{\text{def}} \quad X \preceq Y \;\wedge\; Y \preceq X$$

$X \widehat{=} Y$ states that X and Y are synchronous. i.e. they are present at the same instants. We shall define precisely this notion of instant in Subsection 2.3.

Internal Operations on Signals We define two operations on signals: The down-sampling $\otimes$ and the deterministic merge $\oplus$.

$X \otimes Y$ selects the events of X which are synchronous with an event of Y (see, for instance, Figure 3). For all signals X and Y,

$$X \otimes Y \;=_{\text{def}}\; \{x \in X \mid \exists y \in Y \;\cdot\; x \sim y\}$$

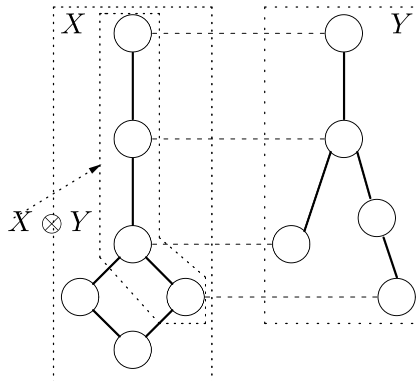


Fig. 3. Example of down-sampling ($X \otimes Y$)

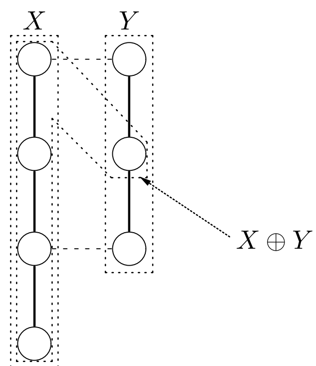


Fig. 4. Example of deterministic merge ($X \oplus Y$)

$\otimes$ is an internal operation on $\mathcal{S}_{\mathcal{E}}$ i.e. for all signals X and Y, $X \otimes Y$ is a signal. Indeed, $X \otimes Y$ is a subset of X, therefore (3) holds. Note that, in Figure 3, although X and Y are imaginary signals, the result $X \otimes Y$ is a real signal in this example.

$X \oplus Y$ is the union of the sets X and Y minus the events of Y that are synchronous with an event of X. In other words, if X are Y are present at a same instant then the priority is given to the left signal X (see, for instance, Figure 4). For all signals X and Y,

$$X \oplus Y \ =_{\text{def}} \ X \cup \{y \in Y \ \mid \ \forall x \in X \ \cdot \ \neg x \sim y\}$$

By construction, $\oplus$ is an internal operation on $\mathcal{S}_{\mathcal{E}}$ i.e. for all signals X and Y, $X \oplus Y$ is a signal. Note that, in Figure 4, although X and Y are real signals, their deterministic merge $X \oplus Y$ is an imaginary signal because its events are not totally ordered by $\ll$. This imaginary signal comes from the lack of synchronisation constraints between X and Y. More generally, it is easy to see that imaginary signals are obtained by closure under deterministic merge.

Algebraic Properties First, the following properties are clear from the definitions of $\otimes$ and $\oplus$:

$$(X \otimes Y) \preceq X \qquad X \preceq (X \oplus Y)$$
$$(X \otimes Y) \preceq Y \qquad Y \preceq (X \oplus Y)$$

It is also clear that $\otimes$ and $\oplus$ are not commutative but they satisfy Proposition 1 stated below. In order to prove this proposition, we first prove the two following lemmas.

Lemma 1 *For all signals X, Y and Z, $Z \preceq X \wedge Z \preceq Y \Rightarrow Z \preceq X \otimes Y$.*

Proof. Let X, Y and Z be three signals such that $Z \preceq X$ and $Z \preceq Y$. Let $z \in Z$. There exists an $x \in X$ such that $x \sim z$ and a $y \in Y$ such that $y \sim z$. Thus, by transitivity of $\sim$, $x \sim y$. Thus $x \in X \otimes Y$. Thus $Z \preceq X \otimes Y$. $\square$

Lemma 2 *For all signals X, Y and Z, $X \preceq Z \wedge Y \preceq Z \Rightarrow X \oplus Y \preceq Z$.*

Proof. Let X, Y and Z be three signals such that $X \preceq Z$ and $Y \preceq Z$. Let $x \in X \oplus Y$. Two cases are possible:

1. If $x \in X$, then, from $X \preceq Z$, we deduce there exists a $z \in Z$ such that $x \sim z$.
2. If $x \in Y$, then, from $Y \preceq Z$, we deduce there exists a $z \in Z$ such that $x \sim z$.

Thus $X \oplus Y \preceq Z$. $\square$

Proposition 1 *For all signals X and Y, $X \otimes Y \mathrel{\widehat{=}} Y \otimes X$ and $X \oplus Y \mathrel{\widehat{=}} Y \oplus X$.*

Proof. We know that $X \otimes Y \preceq Y$ and $X \otimes Y \preceq X$. From Lemma 1, we thus deduce that $X \otimes Y \preceq Y \otimes X$. And similarly, we can prove $Y \otimes X \preceq X \otimes Y$. Thus $X \otimes Y \mathrel{\widehat{=}} Y \otimes X$.

We know that $Y \preceq X \oplus Y$ and $X \preceq X \oplus Y$. From Lemma 2, we thus deduce that $Y \oplus X \preceq X \oplus Y$. And similarly, we can prove $X \oplus Y \preceq Y \oplus X$. Thus $X \oplus Y \mathrel{\widehat{=}} Y \oplus X$. $\square$

2.3 Instant and Trace

In this subsection, we define the notions of instant and trace and we prove other algebraic properties of $\otimes$ and $\oplus$ through their translation into the trace semantics.

Instant Logical instants are modelled by equivalence classes of synchronous events. The set of instants is the quotient of $\mathcal{E}$ by $\sim$:

$$\mathcal{I}_{\mathcal{E}} \ =_{\text{def}} \ \mathcal{E}/\!\sim$$

For any event x, we write $\widetilde{x}$ its equivalence class that we call its instant. The preorder $\ll$ on $\mathcal{E}$ gives rise to a partial order on $\mathcal{I}_{\mathcal{E}}$. For all events x and y,

$$\widetilde{x} \rhd \widetilde{y} \ \Leftrightarrow_{\text{def}} \ x \ll y$$

Trace In order to define traces, we need to prove Lemma 3 relating signals and instants.

Lemma 3 *For any signal $X \in \mathcal{S}_{\mathcal{E}}$, for any instant $i \in \mathcal{I}_{\mathcal{E}}$,*

$$X \cap i = \emptyset \ \vee \ \exists x \in X \ \cdot \ X \cap i = \{x\}$$

Proof. Let $X \in \mathcal{S}_{\mathcal{E}}$ and $i \in \mathcal{I}_{\mathcal{E}}$. Suppose that $X \cap i = \{x, x'\}$. $x \in i$ and $x' \in i$, thus $x \sim x'$. From (3), we thus know that $x = x'$. $\square$

A signal X is said absent at the instant i if and only if $X \cap i = \emptyset$. Otherwise it is said present at the instant i.

We define $t_X(i)$ to be the unique event at the intersection of X and i if X is present at the instant i. Or else it is the special value $\perp \notin \mathcal{E}$ if the signal is absent at the instant i. This function t_X is well defined thanks to Lemma 3.

$$\begin{aligned} t_X : \mathcal{I}_{\mathcal{E}} &\longrightarrow \mathcal{E}_{\perp} \\ i &\longmapsto \begin{cases} x \text{ if } X \cap i = \{x\} \\ \perp \text{ otherwise} \end{cases} \end{aligned}$$

where $\mathcal{E}_{\perp} \ =_{\text{def}} \ \mathcal{E} \cup \{\perp\}$.

t_X is called the trace of X. Now we relate trace properties with signal properties. We first prove the following lemma.

Lemma 4 *For any signal $X \in \mathcal{S}_{\mathcal{E}}$, for any event $x \in X$, $X \cap \widetilde{x} = \{x\}$.*

Proof. Let $X \in \mathcal{S}_{\mathcal{E}}$ and $x \in X$. $x \in \widetilde{x}$ thus $x \in X \cap \widetilde{x}$. But $X \cap \widetilde{x}$ is a singleton, thus $X \cap \widetilde{x} = \{x\}$. $\square$

We can then prove the following proposition.

Proposition 2 *For all signals X and Y, $X = Y \Leftrightarrow t_X = t_Y$.*

Proof. Let $(X, Y) \in \mathcal{S}_{\mathcal{E}}^2$.

1. If $X = Y$, then $t_X(i) = t_Y(i)$ for any i by reflexivity of the equality.
2. Suppose that $t_X = t_Y$. Let $x \in X$. $t_X(\widetilde{x}) = t_Y(\widetilde{x})$ i.e. $X \cap \widetilde{x} = Y \cap \widetilde{x}$. But $X \cap \widetilde{x} = \{x\}$ from Lemma 4. Thus $Y \cap \widetilde{x} = \{x\}$. Thus $x \in Y$. And similarly for $y \in Y$, we prove that $y \in X$. □

To ease proof of properties of operators $\otimes$ and $\oplus$, we translate them into the trace model.

Let $\cdot$ be the operator on traces defined by:

$$\begin{aligned} t_X \cdot t_Y : \mathcal{I}_{\mathcal{E}} &\longrightarrow \mathcal{E}_{\perp} \\ i &\longmapsto \begin{cases} t_X(i) \text{ if } t_X(i) \neq \perp \text{ and } t_Y(i) \neq \perp \\ \perp \text{ otherwise} \end{cases} \end{aligned}$$

The following proposition relates the operators $\cdot$ and $\otimes$.

Proposition 3 *For all signals X and Y, $t_{X \otimes Y} = t_X \cdot t_Y$.*

Proof. Let $i \in \mathcal{I}_{\mathcal{E}}$. Two cases are possible:

1. If $(X \otimes Y) \cap i = \{x_0\}$, then $t_{X \otimes Y}(i) = x_0$. And $\{x \in X \mid \exists y \in Y, x \sim y\} = \{x_0\}$. Thus $X \cap i = \{x_0\}$ and there exists $y_0 \in Y$ such that $x_0 \sim y_0$. Thus $Y \cap i = \{y_0\}$. Thus $t_X(i) = x_0 \neq \perp$ and $t_Y(i) \neq \perp$. Finally, $t_X.t_Y(i) = x_0 = t_{X \otimes Y}(i)$.
2. If $(X \otimes Y) \cap i = \emptyset$, then $t_{X \otimes Y}(i) = \perp$. And $X \cap i = \emptyset$, thus $t_X(i) = \perp = t_{X \otimes Y}(i)$. □

Let $+$ be the operator on traces defined by:

$$\begin{aligned} t_X + t_Y : \mathcal{I}_{\mathcal{E}} &\longrightarrow \mathcal{E}_{\perp} \\ i &\longmapsto \begin{cases} t_X(i) \text{ if } t_X(i) \neq \perp \\ t_Y(i) \text{ otherwise} \end{cases} \end{aligned}$$

The following proposition relates the operators $+$ and $\oplus$.

Proposition 4 *For all signals X and Y, $t_{X \oplus Y} = t_X + t_Y$.*

t_X	t_Y	t_Z	$t_X \cdot t_Y$	$(t_X \cdot t_Y) \cdot t_Z$	$t_Y \cdot t_Z$	$t_X \cdot (t_Y \cdot t_Z)$
$\perp$	$\perp$	$\perp$	$\perp$	$\perp$	$\perp$	$\perp$
$\perp$	$\perp$	z	$\perp$	$\perp$	$\perp$	$\perp$
$\perp$	y	$\perp$	$\perp$	$\perp$	$\perp$	$\perp$
$\perp$	y	z	$\perp$	$\perp$	y	$\perp$
x	$\perp$	$\perp$	$\perp$	$\perp$	$\perp$	$\perp$
x	$\perp$	z	$\perp$	$\perp$	$\perp$	$\perp$
x	y	$\perp$	x	$\perp$	$\perp$	$\perp$
x	y	z	x	x	y	x

Fig. 5. Associativity of $\cdot$

Proof. Let $i \in \mathcal{I}_{\mathcal{E}}$. Two cases are possible:

1. If $X \cap i = \{x_0\}$, then $t_X(i) = x_0 \neq \perp$, and thus $t_X + t_Y(i) = t_X(i) = x_0$. From $X \subseteq X \oplus Y$ and $X \cap i = \{x_0\}$, we deduce that $t_{X \oplus Y}(i) = (X \oplus Y) \cap i = \{x_0\} = t_X + t_Y(i)$.
2. If $X \cap i = \emptyset$, then $t_X(i) = \perp$. And thus $t_X + t_Y(i) = t_Y(i)$.
   (a) If $Y \cap i = \{y_0\}$, then $t_Y(i) = y_0$ and $(X \oplus Y) \cap i = \{y_0\}$. Thus $t_{X \oplus Y}(i) = y_0 = t_Y(i) = t_X + t_Y(i)$.
   (b) If $Y \cap i = \emptyset$, then $t_Y(i) = \perp$ and $(X \oplus Y) \cap i = \emptyset$. Thus $t_{X \oplus Y}(i) = \perp = t_Y(i) = t_X + t_Y(i)$. $\square$

We can now easily prove some algebraic laws of $\otimes$ and $\oplus$.

Proposition 5 *For all signals X, Y and Z,*

$$\begin{aligned} t_X \cdot (t_Y \cdot t_Z) &= (t_X \cdot t_Y) \cdot t_Z \\ t_X + (t_Y + t_Z) &= (t_X + t_Y) + t_Z \end{aligned}$$

Proof. It is sufficient to prove these equalities for any instant. Let $X, Y, Z \in \mathcal{S}_{\mathcal{E}}$. Let $i \in \mathcal{I}_{\mathcal{E}}$. $t_X(i) = \perp \vee \exists x \in X, t_X(i) = x$, $t_Y(i) = \perp \vee \exists y \in Y, t_Y(i) = y$ and $t_X(i) = \perp \vee \exists z \in Z, t_Z(i) = z$. We thus only need to enumerate the 8 possible cases to establish the associativity relations. It is done in Figures 5 and 6. $\square$

Corollary 1 *For all signals X, Y and Z,*

$$\begin{aligned} X \otimes (Y \otimes Z) &= (X \otimes Y) \otimes Z \\ X \oplus (Y \oplus Z) &= (X \oplus Y) \oplus Z \end{aligned}$$

t_X	t_Y	t_Z	$t_X + t_Y$	$(t_X + t_Y) + t_Z$	$t_Y + t_Z$	$t_X + (t_Y + t_Z)$
$\perp$	$\perp$	$\perp$	$\perp$	$\perp$	$\perp$	$\perp$
$\perp$	$\perp$	z	$\perp$	z	z	z
$\perp$	y	$\perp$	y	y	y	y
$\perp$	y	z	y	y	y	y
x	$\perp$	$\perp$	x	x	$\perp$	x
x	$\perp$	z	x	x	z	x
x	y	$\perp$	x	x	y	x
x	y	z	x	x	y	x

Fig. 6. Associativity of +

Proof. It follows from Propositions 2 and 5. □

Proposition 6 *For all signals X, Y and Z,*

$$\begin{aligned} t_X \cdot (t_Y + t_Z) &= (t_X \cdot t_Y) + (t_X \cdot t_Z) \\ t_X + (t_Y \cdot t_Z) &= (t_X + t_Y) \cdot (t_X + t_Z) \\ (t_X + t_Y) \cdot t_Z &= (t_X \cdot t_Z) + (t_Y \cdot t_Z) \end{aligned}$$

Proof. It is sufficient to prove these equalities for any instant. Let $X, Y, Z \in \mathcal{S}_{\mathcal{E}}$. Let $i \in \mathcal{I}_{\mathcal{E}}$. $t_X(i) = \perp \vee \exists x \in X, t_X(i) = x$, $t_Y(i) = \perp \vee \exists y \in Y, t_Y(i) = y$ and $t_X(i) = \perp \vee \exists z \in Z, t_Z(i) = z$. We thus only need to enumerate the 8 possible cases to establish the distributivity relations. It can be done using tables as in the previous proposition. □

Remark 1 *$+$ is not distributive to the right. Indeed, if $t_X(i) = x$, $t_Y(i) = \perp$ and $t_Z(i) = z$, then $((t_X \cdot t_Y) + t_Z)(i) = z$ and $((t_X + t_Z) \cdot (t_Y + t_Z))(i) = x$.*

Corollary 2 *For all signals X, Y and Z,*

$$\begin{aligned} X \otimes (Y \oplus Z) &= (X \otimes Y) \oplus (X \otimes Z) \\ X \oplus (Y \otimes Z) &= (X \oplus Y) \otimes (X \oplus Z) \\ (X \oplus Y) \otimes Z &= (X \otimes Z) \oplus (Y \otimes Z) \end{aligned}$$

Proof. It follows from Propositions 2 and 6. □

Traces can be used to relate the semantics of the synchronous language Signal given in Section 5 with the original trace semantics of Signal [15,16].

In order to study the temporal relations between signals, we define the equivalence classes of signals by $\widehat{=}$. The set of clocks $\mathcal{C}_\mathcal{E}$ is the quotient of $\mathcal{S}_\mathcal{E}$ by $\widehat{=}$:

$$\mathcal{C}_\mathcal{E} \;=_{\text{def}}\; \mathcal{S}_\mathcal{E}/\widehat{=}$$

For any signal X, we write $\widehat{X}$ its equivalence class that we call its clock. $\widehat{\emptyset}$ is called the null clock. The clock of a real (resp. imaginary) signal is a real (resp. imaginary) clock.

The preorder $\preceq$ on $\mathcal{S}_\mathcal{E}$ gives rise to an order $\sqsubseteq$ on $\mathcal{C}_\mathcal{E}$. For all signals X and Y,

$$\widehat{X} \sqsubseteq \widehat{Y} \;\Leftrightarrow_{\text{def}}\; X \preceq Y$$

The Boolean Lattice of Clocks Intuitively, it is clear that a clock should be related to a set of instants and conversely. We show that the set of clocks $\mathcal{C}_\mathcal{E}$ and the powerset $\mathfrak{P}(\mathcal{I}_\mathcal{E})$ of $\mathcal{I}_\mathcal{E}$ are isomorphic. To prove this isomorphism, we need the Axiom of Choice.

Axiom 1 (Axiom of Choice) *For any set E and any equivalence relation R on E, there exists a function $c_R : E \longrightarrow E$ such that*

- $\forall x \in E \;\cdot\; x \mathrel{R} c_R(x)$, *and*
- $\forall x, y \in E \;\cdot\; x \mathrel{R} y \;\Rightarrow\; c_R(x) = c_R(y)$.

From this axiom we extract the two useful choice functions $c_\sim : \mathcal{E} \longrightarrow \mathcal{E}$ and $c_{\widehat{=}} : \mathcal{S}_\mathcal{E} \longrightarrow \mathcal{S}_\mathcal{E}$.

Theorem 1 $(\mathcal{C}_\mathcal{E}, \sqsubseteq)$ *and* $(\mathfrak{P}(\mathcal{I}_\mathcal{E}), \subseteq)$ *are isomorphic.*

Proof. let $f : \mathfrak{P}(\mathcal{I}_\mathcal{E}) \longrightarrow \mathcal{C}_\mathcal{E}$ be a function which associates a clock to any set of instants:

$$\begin{aligned} f : \mathfrak{P}(\mathcal{I}_\mathcal{E}) &\longrightarrow \mathcal{C}_\mathcal{E} \\ I &\longmapsto \widehat{X} \text{ with } X = c_\sim\langle\{x \in i \mid i \in I\}\rangle \end{aligned}$$

We first prove that this definition is legal. X is a set of events (It is the image set of $\{x \in i \mid i \in I\}$ by $c_\sim$). From the Axiom of Choice, (3) is established i.e. two synchronous events of X are necessarily equal. X is thus a signal and we can take its clock $\widehat{X}$.

Let us prove that f is monotonic. Let I and J be two sets of instants such that $I \subseteq J$. Thus $\{x \in i \mid i \in I\} \subseteq \{x \in i \mid i \in J\}$. Let $X = c_{\sim}\langle\{x \in i \mid i \in I\}\rangle$ and $Y = c_{\sim}\langle\{x \in i \mid i \in J\}\rangle$. $X \subseteq Y$, thus $X \preceq Y$, thus $\widehat{X} \sqsubseteq \widehat{Y}$ i.e. $f(I) \sqsubseteq f(J)$.

Let $g : \mathcal{C}_{\mathcal{E}} \longrightarrow \mathfrak{P}(\mathcal{I}_{\mathcal{E}})$ be a function which associates a set of instants to any clock:

$$\begin{aligned} g : \mathcal{C}_{\mathcal{E}} &\longrightarrow \mathfrak{P}(\mathcal{I}_{\mathcal{E}}) \\ C &\longmapsto \{\widetilde{x} \in \mathcal{I}_{\mathcal{E}} \mid x \in X\} \text{ with } \{X\} = c_{\widehat{=}}\langle C\rangle \end{aligned}$$

We first prove that this definition is legal. C is an equivalence class of synchronous signals. From the Axiom of Choice, the image of C by $c_{\widehat{=}}$ is a singleton, and thus its unique element is called X.

Let us prove that g is monotonic. Let C and D be two clocks such that $C \sqsubseteq D$. Let X and Y be the respective representatives of C and D chosen by the choice function $c_{\widehat{=}}$ i.e. $\{X\} = c_{\widehat{=}}(C)$ and $\{Y\} = c_{\widehat{=}}(D)$. From the definition of $\sqsubseteq$, we deduce $X \preceq Y$. Let $i \in g(C)$ and $x \in X$ be the representative of i in X. From $X \preceq Y$ we deduce that there exists $y \in Y$ such that $x \sim y$. Thus $y \in i$. Thus $i \in g(D)$ (from the definition of g). Finally $g(C) \subseteq g(D)$.

Let us prove that $g \circ f = \mathbf{Id}_{\mathfrak{P}(\mathcal{I}_{\mathcal{E}})}$. Let $I \subseteq \mathcal{I}_{\mathcal{E}}$ be a set of instants. Let $X = c_{\sim}\langle\{x \in i \mid i \in I\}\rangle$. We have $f(I) = \widehat{X}$. Let $\{Y\} = c_{\widehat{=}}\langle\widehat{X}\rangle$. We have $g \circ f(I) = g(f(I)) = g(\widehat{X}) = \{\widetilde{y} \in \mathcal{I}_{\mathcal{E}} \mid y \in Y\}$. Thus, we just need to prove that $\{\widetilde{y} \in \mathcal{I}_{\mathcal{E}} \mid y \in Y\} = I$.

1. Let $i \in \{\widetilde{y} \in \mathcal{I}_{\mathcal{E}} \mid y \in Y\}$. There exists $y \in i$ such that $y \in Y$. But $X \widehat{=} Y$, thus there exists $x \in X$ such that $x \sim y$. Thus $i = \widetilde{x} = \widetilde{y} \in I$.
2. Let $i \in I$. By definition of X, there exists $x \in X$ such that $x \in i$. But $X \widehat{=} Y$, thus there exists $y \in Y$ such that $x \sim y$. Thus $i = \widetilde{x} = \widetilde{y} \in \{\widetilde{y} \in \mathcal{I}_{\mathcal{E}} \mid y \in Y\}$.

Let us prove that $f \circ g = \mathbf{Id}_{\mathcal{C}_{\mathcal{E}}}$. Let C be a clock. Let X be a representative of C chosen by the choice function $c_{\widehat{=}}$ i.e. $\{X\} = c_{\widehat{=}}(C)$. We have $g(C) = \{\widetilde{x} \in \mathcal{I}_{\mathcal{E}} \mid x \in X\}$. Let $Y = c_{\sim}\langle\{y \in i \mid i \in g(C)\}\rangle$. We have $f \circ g(C) = \widehat{Y}$. We thus just need to prove that $\widehat{Y} = C$, which is equivalent to $X \widehat{=} Y$. By substitution, we obtain $Y = c_{\sim}\langle\{y \in i \mid i \in \{\widetilde{x} \in \mathcal{I}_{\mathcal{E}} \mid x \in X\}\}\rangle$. Thus $Y = c_{\sim}\langle\{y \in \widetilde{x} \mid x \in X\}\rangle$. Thus $X \widehat{=} Y$. $\square$

Using this isomorphism, we define the operator $\backslash$ on clocks which is the counterpart of the operator $\backslash$ on sets of instants which subtracts a set from another. We can also define the complementary of a clock. Let f be the isomorphism from $\mathcal{C}_{\mathcal{E}}$ to $\mathfrak{P}(\mathcal{I}_{\mathcal{E}})$. For all clocks C and D,

$$C \backslash D =_{\text{def}} f^{-1}(f(C) \backslash f(D))$$

$$\overline{C} =_{\text{def}} f^{-1}(\mathcal{I}_\mathcal{E} \backslash f(C))$$

The complementary of a signal X is a "chosen" signal $\overline{X}$ (using the Axiom of Choice) of clock $\overline{\widehat{X}}$.

$$\{\overline{X}\} = c_{\widehat{=}}\langle \overline{\widehat{X}} \rangle$$

$(\mathfrak{P}(\mathcal{I}_\mathcal{E}), \subseteq)$ is a boolean lattice. From the isomorphism, we deduce that $(\mathcal{C}_\mathcal{E}, \sqsubseteq)$ is also a boolean lattice i.e. it is complete, distributive, and there exists a null element $\widehat{\emptyset}$ and a universal element $\mathbf{1}_\mathcal{E}$ equal to $f^{-1}(\mathcal{I}_\mathcal{E})$.

The operator $\otimes$ on $\mathcal{S}_\mathcal{E}$ gives rise to the greatest lower bound operator $\sqcap$ on $\mathcal{C}_\mathcal{E}$. For all signals X and Y,

$$\widehat{X} \sqcap \widehat{Y} =_{\text{def}} \widehat{X \otimes Y}$$

Proposition 7 *$C \sqcap D$ is the greatest lower bound of clocks C and D.*

Proof. Let C and D be two clocks. Let $X \in C$ and $Y \in D$ be the respective representative signals. We have $X \otimes Y \preceq X$ and $X \otimes Y \preceq Y$, thus, by definition of $\sqsubseteq$, $\widehat{X \otimes Y} \sqsubseteq \widehat{X}$ and $\widehat{X \otimes Y} \sqsubseteq \widehat{Y}$. $\widehat{X \otimes Y}$ is thus a lower bound of $\{\widehat{X}, \widehat{Y}\}$.

Let us prove that it is the greatest lower bound. Let $\widehat{Z}$ be a lower bound of $\{\widehat{X}, \widehat{Y}\}$ such that $\widehat{X \otimes Y} \sqsubseteq \widehat{Z}$. Let $z \in Z$. From $Z \preceq X$ and $X \preceq Y$, we deduce that there exists a pair of events $(x, y) \in X \times Y$ such that $x \sim z$ and $y \sim z$. By transitivity of $\sim$, we have $x \sim y$, and thus $x \in X \otimes Y$. Thus $Z \preceq X \otimes Y$ i.e. $Z \sqsubseteq X \otimes Y$. Thus $Z = X \otimes Y$. □

The operator $\oplus$ on $\mathcal{S}_\mathcal{E}$ gives rise to the least upper bound operator $\sqcup$ on $\mathcal{C}_\mathcal{E}$. For all signals X and Y,

$$\widehat{X} \sqcup \widehat{Y} =_{\text{def}} \widehat{X \oplus Y}$$

Proposition 8 *$C \sqcup D$ is the least upper bound of clocks C and D.*

Proof. Let C and D be two clocks. Let $X \in C$ and $Y \in D$ be the respective representative signals. We have $X \preceq X \oplus Y$ and $Y \preceq X \oplus Y$ thus, by definition of $\sqsubseteq$, $\widehat{X} \sqsubseteq \widehat{X \oplus Y}$ and $\widehat{Y} \sqsubseteq \widehat{X \oplus Y}$. $\widehat{X \oplus Y}$ is thus an upper bound of $\{\widehat{X}, \widehat{Y}\}$.

Let us show that it is the least upper bound. Let $\widehat{Z}$ be an upper bound of $\{\widehat{X}, \widehat{Y}\}$ such that $\widehat{Z} \sqsubseteq \widehat{X \oplus Y}$. Let $x \in X \oplus Y$. If $x \in X$, then, from $X \preceq Z$, we deduce that there exists an event $z \in Z$ such that $x \sim z$. Otherwise $x \in Y$, and thus, from $Y \preceq Z$, we deduce that there exists an event $z \in Z$ such that $x \sim z$. Thus $X \oplus Y \preceq Z$ i.e. $\widehat{X \oplus Y} \sqsubseteq \widehat{Z}$. □

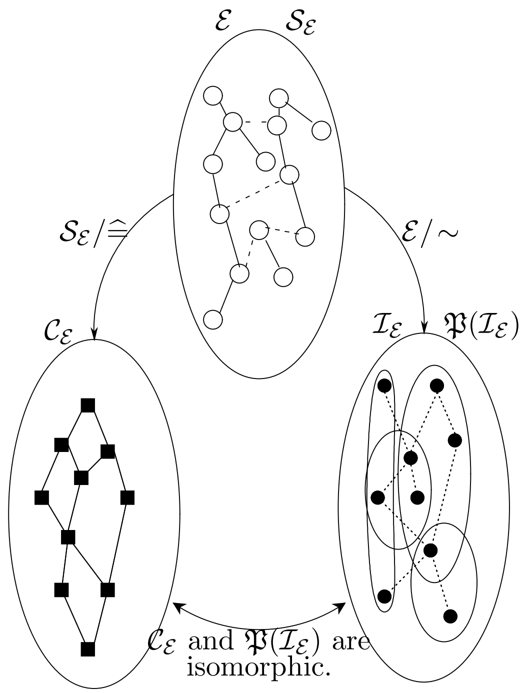


Fig. 7. Summary

We can see that the closure of the set of signals under deterministic merge induces the closure of the set of clocks under least upper bound.

Figure 7 summarises the relations between the different involved sets.

2.5 Configuration

Synchronous structures do not allow explicit representation of the states of a system. However, we can define a notion of computation state. A configuration is a computation state of the system described by the synchronous structure i.e. the set of events which have occurred in the computation. Formally, a configuration is a downward closed subset of a synchronous structure. Let $(\mathcal{E}, \ll)$ be a synchronous structure. A subset c of $\mathcal{E}$ is a configuration of $\mathcal{E}$ if and only if

$$\forall x, y \in \mathcal{E} \quad \cdot \quad x \ll y \ \wedge \ y \in c \quad \Rightarrow \quad x \in c$$

If an event is in a configuration then it is clear that all its predecessors and synchronous events are also in this configuration.

Proposition 9 *Let $(\mathcal{E}, \ll)$ be a synchronous structure. A configuration of $\mathcal{E}$*

is a subset c of $\mathcal{E}$ such that:

- $\forall (x,y) \in \mathcal{E}^2, x \leq y \wedge y \in c \Rightarrow x \in c$, *and*
- $\forall (x,y) \in \mathcal{E}^2, x \sim y \wedge y \in c \Rightarrow x \in c$.

Let $\mathcal{D}_\mathcal{E}$ be the set of configurations of $\mathcal{E}$ and $\mathcal{D}^0_\mathcal{E} \subseteq \mathcal{D}_\mathcal{E}$ be the subset of finite configurations. These sets are partially ordered by inclusion $\subseteq$.

We write $\lceil x \rceil$ the set $\{x' \in \mathcal{E} \mid x' \ll x\}$. By definition, $\lceil x \rceil$ is a configuration.

Lemma 5 *Let $(\mathcal{E}, \ll)$ be a synchronous structure. Let $x \in \mathcal{E}$. $\lceil x \rceil \backslash \widetilde{x}$ is a configuration.*

Proof. Let $y, y' \in \mathcal{E}$ be such that $y' \in \lceil x \rceil \backslash \widetilde{x}$ and $y \ll y'$. We have $y' \in \lceil x \rceil$ (thus $y' \ll x$) and $y' \not\sim x$. By transitivity, $y \ll x$ and $y \not\sim x$. Thus $y \in \lceil x \rceil \backslash \widetilde{x}$. $\square$

Lemma 6 *Let $(\mathcal{E}, \ll)$ be a synchronous structure. Let $c_1, c_2 \in \mathcal{D}_\mathcal{E}$ be a pair of configurations. $c_1 \cup c_2$ is a configuration.*

Proof. Let $x, x' \in \mathcal{E}$ such that $x' \in c_1 \cup c_2$ and $x \ll x'$. Two cases (non exclusive) are possible:

1. $x' \in c_1$: By definition of a configuration, $x \in c_1$. Thus $x \in c_1 \cup c_2$.
2. $x' \in c_2$: By definition of a configuration, $x \in c_2$. Thus $x \in c_1 \cup c_2$. $\square$

The preorder associated with synchronous structure can be recovered from its finite configurations.

Proposition 10 *For all events x and y,*

$$x \ll y \Leftrightarrow \forall c \in \mathcal{D}^0_\mathcal{E}, y \in c \Rightarrow x \in c$$

Proof. Let $x, y \in \mathcal{E}$.

($\Rightarrow$) Suppose that $x \ll y$. Let $c \in \mathcal{D}^0_\mathcal{E}$ be a finite configuration such that $y \in c$. We obtain $x \in c$ by definition of a configuration.

($\Leftarrow$) By reducing to the absurd, suppose that $\forall c \in \mathcal{D}^0_\mathcal{E}, y \in c \Rightarrow x \in c$ and $x \not\ll y$. Let us take $c = \lceil y \rceil$. As $x \not\ll y$, $x \notin \lceil y \rceil$, which is a contradiction. $\square$

We can regard configurations and instants as respectively states and labels in a labelled transition system. The set of states is the set of configurations $\mathcal{D}_\mathcal{E}$. The initial state is the empty configuration $\emptyset \in \mathcal{D}_\mathcal{E}$. The set of labels is the set of instants $\mathcal{I}_\mathcal{E}$. The transition relation is a ternary relation over $\mathcal{D}_\mathcal{E} \times \mathcal{I}_\mathcal{E} \times \mathcal{D}_\mathcal{E}$ such that for any $i \in \mathcal{I}_\mathcal{E}$ and for any $(c, c') \in {\mathcal{D}_\mathcal{E}}^2$,

$$c \xrightarrow{i} c' \Leftrightarrow_{\text{def}} i \nsubseteq c \wedge c' = c \cup i$$

Theorem 2 *Let i_1 and i_2 be two instants. The following propositions are equivalent.*

(i) *There exists configurations c_0, c_1, c_2 et c_3 such that:*

$$\begin{tikzcd} & c_3 & \\ c_1 \arrow[ur, "i_2"] & & c_2 \arrow[ul, "i_1"'] \\ & c_0 \arrow[ul, "i_1"] \arrow[ur, "i_2"'] & \end{tikzcd}$$

(ii) *i_1 et i_2 are not comparable.*

Proof.

$(i) \Rightarrow (ii)$ Suppose that Proposition (i) holds. Let us prove by reducing to the absurd that i_1 and i_2 are comparable. Suppose that $i_1 \triangleright i_2$. We have $i_2 \subseteq c_2$. But $i_1 \triangleright i_2$, thus $i_1 \subseteq c_2$. The transition $c_2 \xrightarrow{i_1} c_3$ is thus not possible. Symmetrically, if $i_2 \triangleright i_1$ then the transition $c_1 \xrightarrow{i_2} c_3$ is not possible. It is a contradiction, thus i_1 and i_2 are not comparable.

$(ii) \Rightarrow (i)$ Suppose that i_1 and i_2 are not comparable. Let $x_1 \in i_1$ and $x_2 \in i_2$. We just have to take:

$$\begin{aligned} c_0 &= (\lceil x_1 \rceil \backslash i_1) \cup (\lceil x_2 \rceil \backslash i_2) \\ c_1 &= \lceil x_1 \rceil \cup (\lceil x_2 \rceil \backslash i_2) \\ c_2 &= (\lceil x_1 \rceil \backslash i_1) \cup \lceil x_2 \rceil \\ c_3 &= \lceil x_1 \rceil \cup \lceil x_2 \rceil \end{aligned}$$

These definitions are correct thanks to Lemma 5 and Lemma 6. □

3 The Category of Signals

Another way to study temporal relations between signals is to define a category of signals in which a morphism describes the temporal relation between two signals. The purpose of this categorical formalisation is to provide a nice and convenient way to relate our work with event structures [17]. Indeed, category theory is a convenient formalism for relating models. We show that the category of signals can be related to the category of event structures by a pair of specification structures [18]. Moreover, we show that this category is Cartesian closed.

Suppose that X and Y are two signals such that $X \preceq Y$. Thus, for any event $x \in X$, there exists an event $y \in Y$ such that $x \sim y$, by definition of $\preceq$. This event y is unique by definition of a signal. Hence, we can define a total function $[Y]_X$, called signal morphism, from X to Y:

$$\begin{aligned} [Y]_X : X &\longrightarrow Y \\ x &\longmapsto y \text{ such that } x \sim y \end{aligned}$$

By definition, $[Y]_X$ is the unique morphism from X to Y. For any signal X, the automorphism $[X]_X$ is the identity function on X.

Proposition 11 *For all signals X and Y such that $X \preceq Y$,*

1. *$[Y]_X$ is injective: $\forall x, x' \in X \cdot [Y]_X(x) = [Y]_X(x') \Rightarrow x = x'$*
2. *$[Y]_X$ is strictly monotonic: $\forall x, x' \in X \cdot x < x' \Rightarrow [Y]_X(x) < [Y]_X(x')$*

Proof. Let X and Y be two signals such that $X \preceq Y$.

1. Let x and x' be two events of X. Suppose that $y = [Y]_X(x) = [Y]_X(x')$. Thus, by definition of $[Y]_X$, $x \sim y$ and $x' \sim y$. By symmetry and transitivity $\sim$, we have $x \sim x'$. Finally, by definition of a signal, we obtain $x = x'$.
2. Let x and x' be two events of X such that $x < x'$. Suppose that $[Y]_X(x') \leq [Y]_X(x)$. Two cases are possible. First, if $[Y]_X(x') = [Y]_X(x)$, thus, by injectivity of $[Y]_X$, $x = x'$. It is a contradiction. Second, let us take $[Y]_X(x') < [Y]_X(x)$. From $x' \sim [Y]_X(x')$ we deduce $x' \ll [Y]_X(x')$. From $[Y]_X(x') < [Y]_X(x)$ we deduce $[Y]_X(x') \ll [Y]_X(x)$. From $x \sim [Y]_X(x)$ we deduce $[Y]_X(x) \ll x$. By transitivity of $\ll$, we obtain $x' \ll x$. From the hypothesis $x < x'$, we also have $x \ll x'$. Thus $x \sim x'$. Finally, by definition of a signal, we obtain $x = x'$. It is a contradiction. Thus $[Y]_X(x) < [Y]_X(x')$ $\square$

Proposition 12 *$[Y]_X$ is bijective (with $[Y]_X^{-1} = [X]_Y$) if and only if $X \hat{=} Y$.*

Proof.

1. Let X and Y be two synchronous signals. We can thus define the total functions $[Y]_X$ and $[X]_Y$. Let us prove that $[X]_Y$ is the converse of $[Y]_X$ i.e. $[X]_Y \circ [Y]_X = [X]_X$ and $[Y]_X \circ [X]_Y = [Y]_Y$.
   (a) Let x be an event of X. $[X]_Y \circ [Y]_X(x) \in X$ and $[X]_Y \circ [Y]_X(x) \sim x$, thus, by definition of a signal, $x = x'$.

(b) Let y be an event of Y. $[Y]_X \circ [X]_Y(y) \in Y$ and $[Y]_X \circ [X]_Y(y) \sim y$, thus, by definition of a signal, $y = y'$.

Thus $[Y]_X$ is bijective.

2. Let X and Y be two signals such that $[Y]_X$ is bijective. $X \preceq Y$ because $[Y]_X$ exists. *A fortiori* $[Y]_X$ is surjective i.e. for any $y \in Y$ there exists $x \in X$ such that $y = [Y]_X(x)$, and thus $y \sim x$. Thus $Y \preceq X$. $\square$

A signal isomorphism is a bijective signal morphism such that its converse is also a signal morphism. From Proposition 12, two signals are isomorphic if and only if they are synchronous.

Signal morphisms can be composed.

Proposition 13 *For all signals X, Y and Z such that $X \preceq Y \preceq Z$,*

$$[Z]_Y \circ [Y]_X = [Z]_X$$

Proof. Let us prove this proposition by reducing to the absurd. Suppose that $[Z]_X \neq [Z]_Y \circ [Y]_X$. Thus there exists an $x_0 \in X$ such that $[Z]_X(x_0) \neq [Z]_Y \circ [Y]_X(x_0)$. We have $[Z]_X(x_0) \sim x_0$ and $x_0 \sim [Z]_Y \circ [Y]_X(x_0)$. By transitivity, $[Z]_X(x_0) \sim [Z]_Y \circ [Y]_X(x_0)$. But $[Z]_X(x_0)$ and $[Z]_Y \circ [Y]_X(x_0)$ are both in Z. they are thus equal by definition of a signal. It is a contradiction. $\square$

3.2 The Category of Signals

The set of signals and the set of morphisms define a small (preorder) category $\mathsf{Sig}_{\mathcal{E}}$ with product $\otimes$ and coproduct $\oplus$ [19].

More precisely, let X and Y be two objects (i.e. signals) of the category $\mathsf{Sig}_{\mathcal{E}}$. The product object $X \otimes Y$ and the two projections $[X]_{X\otimes Y}$ and $[Y]_{X\otimes Y}$ are a product of X and Y. These data satisfy the property that, for any object Z and all morphisms $f : Z \longrightarrow X$ and $g : Z \longrightarrow Y$, there exists a unique morphism $\langle f, g\rangle : Z \longrightarrow X \otimes Y$ such that the following equations hold:

$$[X]_{X\otimes Y} \circ \langle f, g\rangle = f \tag{4}$$
$$[Y]_{X\otimes Y} \circ \langle f, g\rangle = g \tag{5}$$

Indeed, necessarily $f = [X]_Z$ (resp. $g = [Y]_Z$) because $[X]_Z$ (resp. $[Y]_Z$) is the unique morphism from Z to X (resp. Y). The unique morphism $\langle f, g\rangle$ from Z to $X \otimes Y$ is $[X \otimes Y]_Z$ which exists (because $Z \preceq X$ and $Z \preceq Y$, thus $Z \preceq X \otimes Y$) and establishes (4) and (5) according to Proposition 13.

This definition amounts to saying that the following diagram commutes:

$$\begin{tikzcd}
X & X \otimes Y \arrow[l, "{[X]_{X\otimes Y}}"'] \arrow[r, "{[Y]_{X\otimes Y}}"] & Y \\
& Z \arrow[ul, "{f=[X]_Z}"] \arrow[u, "{\langle f,g\rangle=[X\otimes Y]_Z}"] \arrow[ur, "{g=[Y]_Z}"'] &
\end{tikzcd}$$

The coproduct object $X \oplus Y$ and the two injections $[X \oplus Y]_X$ and $[X \oplus Y]_Y$ are a coproduct of X and Y. These data satisfy the property that, for any object Z and all morphisms $f : X \longrightarrow Z$ and $g : Y \longrightarrow Z$, there exists a unique morphism $[f, g] : X \oplus Y \longrightarrow Z$ such that the following equations hold:

$$[f, g] \circ [X \oplus Y]_X = f \tag{6}$$
$$[f, g] \circ [X \oplus Y]_Y = g \tag{7}$$

Indeed, necessary $f = [Z]_X$ (resp. $g = [Z]_Y$) because $[Z]_X$ (resp. $[Z]_Y$) is the unique morphism from X (resp. Y) to Z. The unique morphism $[f, g]$ from $X \oplus Y$ to Z is $[Z]_{X\oplus Y}$ which exists (because $X \preceq Z$ and $Y \preceq Z$, thus $X \oplus Y \preceq Z$) and establishes (6) and (7) according to Proposition 13.

This definition amounts to saying that the following diagram commutes:

$$\begin{tikzcd}
X \arrow[r, "{[X\oplus Y]_X}"] \arrow[dr, "{f=[Z]_X}"'] & X \oplus Y \arrow[d, "{[f,g]=[Z]_{X\oplus Y}}"] & Y \arrow[l, "{[X\oplus Y]_Y}"'] \arrow[dl, "{g=[Z]_Y}"] \\
& Z &
\end{tikzcd}$$

The signal $\emptyset$ is the unique initial object of the category $\mathsf{Sig}_{\mathcal{E}}$ i.e. for any object X of $\mathsf{Sig}_{\mathcal{E}}$ there exists a unique morphism $[X]_\emptyset : \emptyset \longrightarrow X$. And the coproduct $\oplus$ is defined for each ordered pair of objects of $\mathsf{Sig}_{\mathcal{E}}$. Hence the category $\mathsf{Sig}_{\mathcal{E}}$ has finite coproducts.

It is also possible to construct a terminal object. Let C be the clock corresponding to the set of all instants $\mathcal{I}_{\mathcal{E}}$. Let $Y \in C$ be a signal of clock C. This signal Y is a terminal object i.e. for any object X of $\mathsf{Sig}_{\mathcal{E}}$ there exists a unique morphism $[Y]_X : X \longrightarrow Y$. And the product $\otimes$ is defined for each ordered pair of objects of $\mathsf{Sig}_{\mathcal{E}}$. Hence the category $\mathsf{Sig}_{\mathcal{E}}$ has finite products.

Let $Y \Rightarrow Z$ be the object $\overline{Y} \oplus Z$ and $\mathsf{Apply}_{Y,Z} : (Y \Rightarrow Z) \otimes Y \longrightarrow Z$ be the morphism $[Z]_{(Y\Rightarrow Z)\otimes Y}$. $\mathsf{Apply}_{Y,Z}$ is correctly defined because $(Y \Rightarrow Z)\otimes Y \preceq Z$.

Indeed:

$$
\begin{aligned}
(Y \Rightarrow Z) \otimes Y &= (\overline{Y} \oplus Z) \otimes Y \\
&= (\overline{Y} \otimes Y) \oplus (Z \otimes Y) \\
&= \emptyset \oplus (Z \otimes Y) \\
&= (Z \otimes Y) \\
&\preceq Z
\end{aligned}
$$

In addition, $(Y \Rightarrow Z) \otimes Y = (Z \otimes Y)$. Therefore $\mathsf{Apply}_{Y,Z} = [Z]_{Z \otimes Y}$.

$\mathsf{Sig}_{\mathcal{E}}$ is Cartesian closed i.e. for all objects Z and each morphism $f : X \otimes Y \longrightarrow Z$ there exists a unique morphism $\lambda(f) = [Y \Rightarrow Z]_X : X \longrightarrow (Y \Rightarrow Z)$ such that the following diagram[1] commutes:

$$
\begin{tikzcd}
(Y \Rightarrow Z) \otimes Y \arrow[r, "\mathsf{Apply}_{Y,Z}"] & Z \\
X \otimes Y \arrow[u, "\lambda(f) \otimes [Y]_Y"] \arrow[ur, "f"'] &
\end{tikzcd}
$$

The morphism $\lambda(f) = [Y \Rightarrow Z]_X$ is correctly defined. Indeed, let $x \in X$. Two cases are possible:

1. There exists an event $y \in Y$ such that $x \sim y$. Thus $x \in X \otimes Y$. The existence of f implies $X \otimes Y \preceq Z$. Thus there exists an event $z \in Z$ such that $x \sim z$. we thus have a $z' \in \overline{Y} \oplus Z$ or rather $z' \in (Y \Rightarrow Z)$. Thus $X \preceq (Y \Rightarrow Z)$.
2. There exists an event $y \in \overline{Y}$ such that $x \sim y$. Thus $y \in \overline{Y} \oplus Z$ or rather $y \in (Y \Rightarrow Z)$. Thus $X \preceq (Y \Rightarrow Z)$.

3.3 Relation with Event Structures

Event structures [17] are a fundamental model for concurrency. As synchronous structures, they are based on the notion of event. A category of prime event

[1] Let $f : X \longrightarrow Y$ and $g : X' \longrightarrow Y'$ be two morphisms. The morphism $f \otimes g : X \otimes X' \longrightarrow Y \otimes Y'$ is defined by:

$$f \otimes g = \langle f \circ [X]_{X \otimes X'}, g \circ [X']_{X \otimes X'} \rangle$$

structures can be defined of which the morphisms model a synchronisation between two event structures. Indeed, these morphisms are partial functions η such that $\eta(x) = x'$ states that the occurrence of x implies the simultaneous occurrence of x'. Event structures have a conflict relation which does not exist between signals. This conflict relation can be seen as an enrichment of the category of signals that we will model with a pair of specification structures [18].

Specification Structures Specification structures formalise the idea of enriching a semantic universe with a refined notion of property. Let $\mathbb{C}$ be a category. A specification structure S over $\mathbb{C}$ is defined by the following data:

- a set PA of "properties over A" , for each object A of $\mathbb{C}$,
- a relation $R_{A,B} \subseteq PA \times \mathbb{C}(A, B) \times PB$ for each pair of objects A, B of $\mathbb{C}$.

We write $\varphi\{f\}\psi$ for $(\varphi, f, \psi) \in R_{A,B}$ ("Hoare triples"). This relation is required to satisfy the following axioms, for all morphisms $f : A \longrightarrow B$, $g : B \longrightarrow C$, and for all "properties" $\varphi \in PA$, $\psi \in PB$, and $\theta \in PC$:

$$\varphi\{\mathbf{Id}_A\}\varphi \tag{8}$$

$$\varphi\{f\}\psi \wedge \psi\{g\}\theta \Rightarrow \varphi\{g \circ f\}\theta \tag{9}$$

In fact, these axioms are typed versions of the standard Hoare logic axioms for "sequential composition" and "skip".

With a category $\mathbb{C}$ and a specification structure S over $\mathbb{C}$, we can define a new category $\mathbb{C}_S$. Its objects are pairs (A, φ) with A an object of $\mathbb{C}$ and $\varphi \in PA$. Its morphisms $f : (A, \varphi) \longrightarrow (B, \psi)$ are morphisms $f : A \longrightarrow B$ in $\mathbb{C}_S$ such that $\varphi\{f\}\psi$. Composition and identities are inherited from $\mathbb{C}$. (8) and (9) ensure that $\mathbb{C}_S$ is a category.

Moreover, there is an evident faithful functor from $\mathbb{C}_S$ to $\mathbb{C}$ such that the image of (A, φ) is A. In fact, the notion of specification structure over $\mathbb{C}$ is coextensive with that of faithful functor into $\mathbb{C}$. Indeed, given a faithful functor $F : \mathbb{D} \longrightarrow \mathbb{C}$, we can define a specification structure by:

$$PA =_{\text{def}} \{\varphi \in \mathsf{Obj}(\mathbb{D}) \mid F(\varphi) = A\}$$

$$\varphi\{f\}\psi \Leftrightarrow_{\text{def}} \exists \alpha \in \mathbb{D}(\varphi, \psi) \cdot F(\alpha) = f$$

For instance, if $\mathbb{C} =_{\text{def}} \mathsf{Set}$, $PX =_{\text{def}} X$, and $x\{f\}y \Leftrightarrow_{\text{def}} f(x) = y$, then the category $\mathbb{C}_S$ is the category of pointed sets.

From signals to event structures We cannot define a specification structure directly from the category of signals $\mathsf{Sig}_{\mathcal{E}}$ to the category of prime event structures, because the morphisms of the former are total functions whereas the ones of the latter are partial functions. Hence, we first define an intermediate category $\mathsf{PSig}_{\mathcal{E}}$ whose objects are those of $\mathsf{Sig}_{\mathcal{E}}$, and morphisms $f : X \longrightarrow Y$ are, for all signals X and Y, partial functions defined by:

$$\begin{aligned} f : X &\longrightarrow Y \\ x &\longmapsto \begin{cases} [Y]_X(x) \text{ if } x \in X \otimes Y, \\ \text{not defined otherwise.} \end{cases} \end{aligned}$$

It is clear that the identities are morphisms and morphisms can be composed (as in the category Pfn of sets and partial functions).

We relate these three categories by a pair of specification structures.

First, we define a specification structure S_1 from $\mathsf{PSig}_{\mathcal{E}}$ to the category of signals $\mathsf{Sig}_{\mathcal{E}}$:

- $\mathbb{C} =_{\text{def}} \mathsf{PSig}_{\mathcal{E}}$
- $PX =_{\text{def}} \{*_X\}$
- $*_X\{f\}*_Y \Leftrightarrow_{\text{def}} f$ is total

Second, we define a specification structure S_2 from $\mathsf{PSig}_{\mathcal{E}}$ to the category of prime event structures ES (using the notations from [17]):

- $\mathbb{C} =_{\text{def}} \mathsf{PSig}_{\mathcal{E}}$
- $PX =_{\text{def}} \{\#_X \subseteq X^2 \mid \forall (x, x', x'') \in X, (x, x') \in \#_X \wedge x' \leq x'' \Rightarrow (x, x'') \in \#_X\}$
- $\#_X\{f\}\#_Y \Leftrightarrow_{\text{def}} \begin{cases} \forall x \in X, f(x) \text{ is defined} \Rightarrow \lceil f(x) \rceil \subseteq f(\lceil x \rceil) \\ \forall (x_1, x_2) \in X^2, f(x_1) \veebar\!\vee f(x_2) \Rightarrow x_1 \veebar\!\vee x_2 \end{cases}$

By construction, we obtain that $\mathbb{C}_{S_2}$ is the category of prime event structures.

4 Data Dependence

In this section, we extend the notion of synchronous structure to deal with data dependence.

We associate a data dependency relation $\rightarrow$ to synchronous structures such that a data dependence cannot come from the future.

Definition 3 *A dependent synchronous structure is a triple* $(\mathcal{E}, \ll, \rightarrow)$ *such that* $(\mathcal{E}, \ll)$ *is a synchronous structure and* $\rightarrow$ *is a partial order included in* $\ll$ *i.e.*

$$\forall x, y \in \mathcal{E} \ \cdot \ x \rightarrow y \ \Rightarrow \ x \ll y$$

The inclusion of the data dependency relation in the preorder of the event structure guarantees that the value of an event cannot depend on the value of a future event. Indeed, the preorder represent the global time of the whole system. The data dependencies of an event can only come from past or present values of other events.

Temporal Refinement The synchronous paradigm is a good abstraction for the design and the verification of reactive systems. Particularly, the synchronous language Signal allows the specification of early relational properties of systems which can then be progressively refined in order to obtain an executable specification. Actually, we have to cut into the logical instants of the specification with respect to temporal and data dependence. This transformation is called a temporal refinement. This notion models the search (by the Signal compiler for instance) for an execution order of synchronous events.

Definition 4 *Let* $(\mathcal{E}_1, \ll_1, \rightarrow_1)$ *and* $(\mathcal{E}_2, \ll_2, \rightarrow_2)$ *be dependent synchronous structures. A bijective function* f *from* $\mathcal{E}_1$ *to* $\mathcal{E}_2$ *is a temporal refinement morphism if and only if for any pair of events* x, y *of* $\mathcal{E}_1$,

$$x \sim_1 y \Rightarrow f(x) \sim_2 f(y) \vee f(x) <_2 f(y) \vee f(y) <_2 f(x) \tag{10}$$

$$x <_1 y \Rightarrow f(x) <_2 f(y) \tag{11}$$

$$x \rightarrow_1 y \Rightarrow f(x) \rightarrow_2 f(y) \tag{12}$$

The atomic properties of events are respected by any temporal refinement morphism f. An event is *indivisible* i.e. an event cannot be cut into distinct events because f is a function. An event cannot be *destroyed* because f is total. Distinct events cannot be *joined* because f is injective. An event cannot be *spontaneously created* because f is surjective. (10) enables to order synchronous events. (11) and (12) respectively guarantee that temporal dependence and data dependence are respected.

Temporal refinement morphisms give rise to a partial order $\rightsquigarrow$ over dependent synchronous structures (We identify isomorphic ones). For all dependent synchronous structures $(\mathcal{E}_1, \ll_1, \rightarrow_1)$ and $(\mathcal{E}_2, \ll_2, \rightarrow_2)$,

$$\begin{aligned}&(\mathcal{E}_1, \ll_1, \rightarrow_1) \rightsquigarrow (\mathcal{E}_2, \ll_2, \rightarrow_2) \quad \Leftrightarrow_{\text{def}} \\ &\qquad \text{(There exists a temporal refinement morphism } f : \mathcal{E}_1 \longrightarrow \mathcal{E}_2\text{)}\end{aligned}$$

Dependent synchronous structures such that their events are all synchronous are minimal elements. Dependent synchronous structures such that distinct events cannot be synchronous are maximal elements.

Conditional Dependency We define a ternary relation, called conditional dependency. Intuitively, $X \xrightarrow{C} Y$ states that, at the instants of the clock C, there are dependencies $\rightarrow$ from an event of X to an event of Y in the same instant. That is to say, in this relation, we are only interested in instantaneous dependencies. Practically this relation is used to schedule computations that have to be done in the same logical instant. A set of conditional dependencies is called a scheduling specification. For all signals X, Y and Z,

$$X \xrightarrow{\widehat{Z}} Y \quad \Leftrightarrow_{\text{def}} \quad \forall x \in X \otimes Z \ \cdot \ \exists y \in Y \ \cdot \ x \sim y \ \wedge \ x \rightarrow y$$

It follows from this definition that if $X \xrightarrow{\widehat{Z}} Y$, then $X \otimes Z \preceq Y$; that is to say, for any event of $X \otimes Z$, there exists a synchronous event of Y.

The following theorem enables to compute the transitive closure of a scheduling specification.

Theorem 3 *For all signals X, Y and Z, for all clocks C and D,*

$$\begin{aligned}&X \xrightarrow{C} Y \wedge Y \xrightarrow{D} Z \Rightarrow X \xrightarrow{C \sqcap D} Z \quad \textit{and} \\ &X \xrightarrow{C} Y \wedge X \xrightarrow{D} Y \Rightarrow X \xrightarrow{C \sqcup D} Y\end{aligned}$$

Proof. Let V, W, X, Y and Z be signals.

1. Suppose that $X \xrightarrow{\widehat{V}} Y$ and $Y \xrightarrow{\widehat{W}} Z$. Let $x \in X \otimes (V \otimes W)$. By associativity, we have $x \in (X \otimes V) \otimes W$. Thus $x \in (X \otimes V)$ from the definition of $\otimes$. But $X \xrightarrow{\widehat{V}} Y$, thus there exists $y \in Y$ such that $x \sim y$ and $x \rightarrow y$. From $x \in (X \otimes V) \otimes W$ and $x \sim y$, we deduce that $y \in Y \otimes W$. But $Y \xrightarrow{\widehat{W}} Z$. Thus there exists $z \in Z$ such that $y \sim z$ and $y \rightarrow z$. By transitivity of $\sim$ and $\rightarrow$, we respectively have $x \sim z$ and $x \rightarrow z$. Finally $X \xrightarrow{\widehat{V \otimes W}} Z$ or rather $X \xrightarrow{\widehat{V} \sqcap \widehat{W}} Z$.

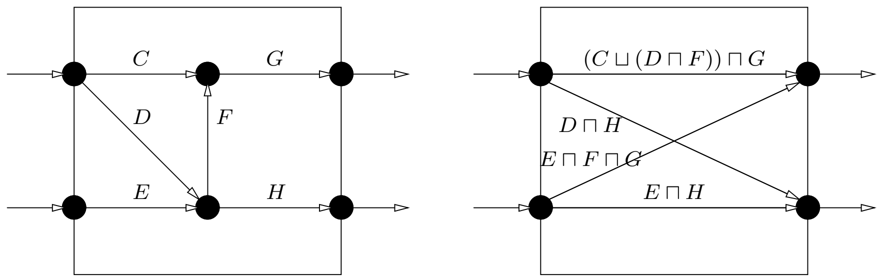


Fig. 8. Abstraction of Scheduling Specifications

2. Suppose that $X \xrightarrow{\widehat{V}} Y$ and $X \xrightarrow{\widehat{W}} Y$. Let $x \in X \otimes (V \oplus W)$. By left distributivity, we obtain $x \in (X \otimes V) \oplus (X \otimes W)$. Two cases are then possible:
   (a) $x \in X \otimes V$: From $X \xrightarrow{\widehat{V}} Y$, we deduce there exists $y \in Y$ such that $x \sim y$ and $x \rightarrow y$.
   (b) $x \in X \otimes W$: From $X \xrightarrow{\widehat{W}} Y$, we deduce there exists $y \in Y$ such that $x \sim y$ and $x \rightarrow y$.

   Finally we obtain $X \xrightarrow{\widehat{V \oplus W}} Y$ or rather $X \xrightarrow{\widehat{V} \sqcup \widehat{W}} Y$. $\square$

In Figure 8, the diagram on the left depicts a scheduling specification involving local variables. These are hidden in the diagram on the right, using Theorem 3.

4.2 Valuated Synchronous Structure

We associate a valuation function v to dependent synchronous structures.

Definition 5 *Let $\mathcal{D}$ be a set (of values). $(\mathcal{E}, \ll, \rightarrow, v)$ is a valuated synchronous structure if and only if $(\mathcal{E}, \ll, \rightarrow)$ is a dependent synchronous structure and v a total function from $\mathcal{E}$ to $\mathcal{D}$.*

Let D be a subset of $\mathcal{D}$. A signal is said of domain D if and only if

$$\forall x \in X \ \cdot \ v(x) \in D$$

We define a preorder $\preceq_v$ over valuated signals. For all signals X and Y,

$$X \preceq_v Y \quad \Leftrightarrow_{\text{def}} \quad X \preceq Y \ \wedge \ \forall x \in X \ \cdot \ v(x) = v([Y]_X(x))$$

This preorder gives rise to an equivalence relation. Two synchronous signals of which events have same values at same instants are equivalent. For all signals

X and Y,

$$X \hat{=}_v Y \quad \Leftrightarrow_{\text{def}} \quad X \preceq_v Y \ \wedge \ Y \preceq_v X$$

Flow In order to study flows of values of a signal, we define a flow as an equivalence class of $\hat{=}_v$. The set of flows $\mathcal{F}_\mathcal{E}$ is the quotient of the set $\mathcal{S}_\mathcal{E}$ by the equivalence relation $\hat{=}_v$.

$$\mathcal{F}_\mathcal{E} =_{\text{def}} \mathcal{S}_\mathcal{E} / \hat{=}_v$$

The equivalence class of a signal X is called the flow of X and is written $|X|$. The preorder $\preceq_v$ over valuated signals gives rise to a partial order on flows. For all signals X and Y,

$$|X| \leq_v |Y| \Leftrightarrow_{\text{def}} X \preceq_v Y$$

Let $v_\perp$ be the extension of v to $\mathcal{E}_\perp$:

$$\begin{aligned} v_\perp : \mathcal{E}_\perp &\longrightarrow \mathcal{D}_\perp \\ x &\longmapsto \begin{cases} v(x) \text{ if } x \in \mathcal{E} \\ \perp \text{ otherwise} \end{cases} \end{aligned}$$

Let X be a valuated signal. Its valuated trace $v_X : \mathcal{I}_\mathcal{E} \longmapsto \mathcal{D}_\perp$ is defined by:

$$v_X = v_\perp \circ t_X$$

For any instant i, $v_X(i)$ is equal to the value of X at the instant i if X is present at this instant, otherwise it is equal to $\perp$.

Lemma 7 *For any signal $X \in \mathcal{S}_\mathcal{E}$, for any event $x \in X$, $v(x) = v_X(\widetilde{x})$.*

Proof. Let $X \in \mathcal{S}_\mathcal{E}$ and $x \in X$.

$$\begin{aligned} v_X(\widetilde{x}) &= v_\perp(t_X(\widetilde{x})) && \text{by definition of } v_X \\ &= v_\perp(x) && \text{from Lemma 4} \\ &= v(x) && \text{by definition of } v_\perp \qquad \square \end{aligned}$$

$|X|$ and v_X are related by the following proposition:

Proposition 14 *For all signals X and Y, $|X| = |Y| \ \Leftrightarrow \ v_X = v_Y$.*

Proof. Let X and Y be signals.

1. Suppose that $|X| = |Y|$. Thus $X \widehat{=}_v Y$. Let $i \in \mathcal{I}_\mathcal{E}$. If $v_X(i) = \perp$, then $v_Y(i) = \perp$ because X and Y are synchronous. If $v_X(i) = v$, then $v_Y(i) = v$ by definition of $\widehat{=}_v$.
2. Suppose that $v_X = v_Y$. It is thus clear that X and Y are synchronous. Let $(x, y) \in X \times Y$ such that $x \sim y$. We have:

$$\begin{aligned} v(x) &= v_X(\widetilde{x}) && \text{from Lemma 7} \\ &= v_Y(\widetilde{x}) && \text{by hypothesis} \\ &= v_Y(\widetilde{y}) && \text{because } x \sim y \\ &= v(y) && \text{from Lemma 7} \qquad \square \end{aligned}$$

Flow function Let $A_1, \ldots, A_n$ and B be subsets of $\mathcal{D}$, and f be a total function from $A_1 \times \cdots \times A_n$ to B. We define its extension $f_\perp$ by:

$$\begin{aligned} f_\perp : A_1 \times \cdots \times A_n \cup \{\perp, \ldots, \perp\} &\longrightarrow B \cup \{\perp\} \\ (x_1, \ldots, x_n) &\longmapsto \begin{cases} f(x_1, \ldots, x_n) \text{ if} \\ (x_1, \ldots, x_n) \in A_1 \times \cdots \times A_n \\ \\ \perp \text{ otherwise} \end{cases} \end{aligned}$$

We write $f|X_1, \ldots, X_n|$ the application of f to all simultaneous values of the signals $X_1, \ldots, X_n$:

$$f|X_1, \ldots, X_n| = \{Y \in \mathcal{F} \mid v_Y = f_\perp \circ \langle v_X \times \cdots \times v_{X_n} \rangle\}$$

where $\langle v_{X_1} \times \cdots \times v_{X_n} \rangle$ is the function such that the image of $x_1, \ldots, x_n \in (X_1 \times \cdots \times X_n)$ is $v_X(x_1), \cdots, v_{X_n}(x_n)$.

Proposition 15 *$f|X_1, \ldots, X_n|$ is a flow.*

Proof. We make the proof for $n = 1$ but it can easily be generalised for any n.

1. Let Y_1 and Y_2 be signals of $f|X|$. $v_{Y_1} = f_\perp \circ v_X$ and $v_{Y_2} = f_\perp \circ v_X$. Thus $v_{Y_1} = v_{Y_2}$. From Proposition 14, we deduce that $|Y_1| = |Y_2|$. Thus $Y_1 \widehat{=}_v Y_2$.
2. Let $Y_1 \in f|X|$ and $Y_2 \in \mathcal{S}_\mathcal{E}$ such that $Y_1 \widehat{=}_v Y_2$. Thus $|Y_1| = |Y_2|$. Thus $v_{Y_2} = v_{Y_1} = f_\perp \circ v_X$. Therefore $Y_2 \in f|X|$. $\square$

Delay The delay enables to move forward the valuations of a real signal. In the case of a real signal, the meaning of the delay is clear. Indeed, the value of an event of a delayed real signal is the value of the previous event if it exists. If it does not exists, then a default value is given. $\mathbf{Pre}^{\mathrm{r}}(u, X, Y)$ states that Y is the delayed real signal of the real signal X, initialised with u.

$$\mathbf{Pre}^{\mathrm{r}}(u, X, Y) \quad \Leftrightarrow_{\text{def}} \quad X \widehat{=} Y \wedge$$
$$\forall y \in Y \; \cdot \; \begin{cases} y \text{ minimal element of Y} \;\Rightarrow\; v(y) = u \\ \exists y^- \in Y \;\cdot\; y^- \prec\!\!\!- y \;\Rightarrow\; v(y) = v([X]_Y(y^-)) \end{cases}$$

This function is well defined because Y is a real signal. Indeed, any event y of Y is either the minimal element of Y or has a unique predecessor in Y.

We can extend the previous definition such that it takes into account data dependence:

$$\mathbf{Pre}^{\mathrm{r}}_{\rightarrow}(u, X, Y) \quad \Leftrightarrow_{\text{def}} \quad X \widehat{=} Y \quad \wedge$$
$$\forall y \in Y \; \cdot \; \begin{cases} y \text{ minimal element of Y} \;\Rightarrow\; v(y) = u \\ \exists y^- \in Y \;\cdot\; y^- \prec\!\!\!- y \;\Rightarrow\; \begin{cases} v(y) = v([X]_Y(y^-)) \wedge \\ [X]_Y(y^-) \rightarrow y \end{cases} \end{cases}$$

In the case of an imaginary signal, where events are only partially ordered "along time", the definition of a delay operator is not obvious. Indeed, there are many ways to delay an imaginary signal. In fact, we wish to have the property that a signal and its delayed signal are synchronous. In an imaginary signal, two events can be in concurrence and if we delay this signal, we have to choose the one that is "pushed forward" by the delay. For instance, on Figure 9, the signal Y is a possible delayed signal of X in which we have chosen to "push forward" events 2, 3 and 5. As this choice would be arbitrary, the delay of imaginary signals is defined as a relation. Let X and Y be signals. Y is a delayed signal of X initialised to u is written $\mathbf{Pre}(u, X, Y)$ and defined by:

$$\mathbf{Pre}(u, X, Y) \Leftrightarrow_{\text{def}}$$
$$\begin{cases} X \widehat{=} Y \wedge \\ \exists X'_M \in \max_{\subseteq}\{X' \subseteq X \mid X' \text{ is totally ordered}\} \;\cdot \\ \qquad \begin{cases} \mathbf{Pre}^{\mathrm{r}}(u, X'_M, [Y]_X\langle X'_M\rangle) \wedge \\ X \backslash X'_M \widehat{=}_v Y \backslash [Y]_X\langle X'_M\rangle \end{cases} \end{cases}$$

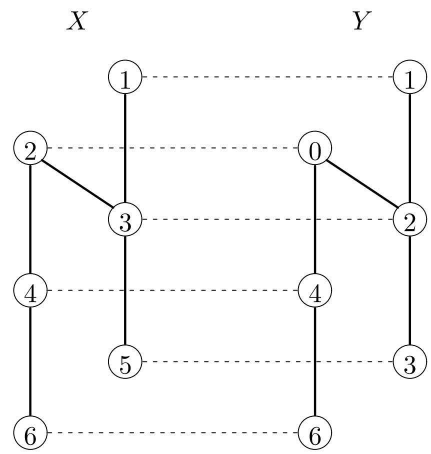


Fig. 9. Delay

where $\max_{\subseteq} E$ is the set of maximal elements of a set E ordered by inclusion. This definition states that X and Y are synchronous, and a real subsignal X'_M of X is delayed while the rest of X does not change. For instance, in Figure 9, X and Y are such that $\mathbf{Pre}(u, X, Y)$.

We can extend the previous definition such that it takes into account data dependence:

$$
\mathbf{Pre}_{\rightarrow}(u, X, Y) \quad \Leftrightarrow_{\text{def}}
\begin{cases}
X \widehat{=} Y \wedge \\
\exists X'_M \in \max_{\subseteq}\{X' \subseteq X \mid X' \text{ is totally ordered}\} \cdot \\
\qquad \begin{cases}
\mathbf{Pre}^{\mathrm{r}}_{\rightarrow}(u, X'_M, [Y]_X\langle X'_M\rangle) \wedge \\
X \backslash X'_M \widehat{=}_v Y \backslash [Y]_X\langle X'_M\rangle \wedge \\
\forall x \in X \backslash X'_M \;\cdot\; x \rightarrow [Y]_X(x)
\end{cases}
\end{cases}
$$

Applied to the example of Figure 9, it gives the graph of Figure 10 where an arrow between two events x and y denotes a data dependence $x \rightarrow y$.

Down-sampling clock For any signal X of domain $\{\mathit{false}, \mathit{true}\}$, the clock $[X]$ is isomorphic to the set of instants where X is present with the value *true*:

$$[X] = \widehat{X'} \text{ with } X' = \{x \in X \mid v(X) = \mathit{true}\}$$

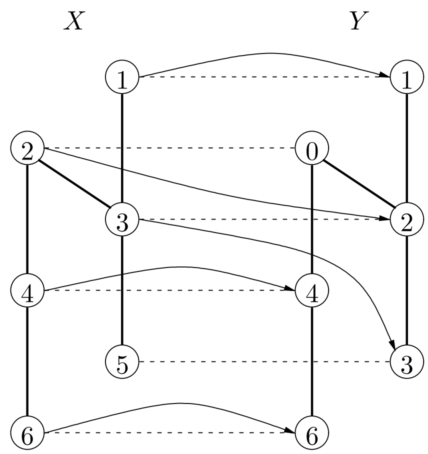


Fig. 10. Delay with dependence

5 Application to Signal

In this section, we give a denotational semantics of the synchronous language Signal into valuated synchronous structures. A process is denoted by a class of dependent synchronous structures. The notation $[\![.]\!]$ is used for different denotation functions because it is clear from the context which one it is.

Let $\mathcal{D}$ be a set of values. Primitive functions are denoted by function over $\mathcal{D}$ (Note that primitive values $\mathtt{v}$ are considered as primitive functions of arity 0):

$$[\![\mathtt{f}]\!] \in \mathcal{D} \times \cdots \times \mathcal{D} \longrightarrow \mathcal{D}$$

Signal variables are evaluated with the signal environment ρ:

$${}^{\mathcal{E}}[\![\mathtt{x}]\!]_\rho = \rho(\mathtt{x}) \in \mathcal{S}_\mathcal{E}$$

Primitive processes are denoted by relations on flows and also by dependence relations from the signals involved in the right part of an equation to the signal of the left part, at the clock of the latter (see Fig 11).

Parallel composition and hiding are respectively denoted by the logical "and" and the existential quantifier of the underlying logic:

$${}^{\mathcal{E}}[\![P_1 \,|\, P_2]\!]^\phi_\rho = {}^{\mathcal{E}}[\![P_1]\!]^\phi_\rho \wedge {}^{\mathcal{E}}[\![P_2]\!]^\phi_\rho \qquad {}^{\mathcal{E}}[\![P/\mathtt{x}]\!]^\phi_\rho = \exists X \in \mathcal{S}_\mathcal{E}, {}^{\mathcal{E}}[\![P]\!]^\phi_{\rho,\mathtt{x}\mapsto X}$$

Process variables are evaluated with the process environment ϕ:

$${}^{\mathcal{E}}[\![\mathtt{p(x1,\ldots,xn)}]\!]^\phi_\rho = \left(\exists \mathcal{E}' \in \phi(\mathtt{p}), \mathcal{E}' = {}^{\mathcal{E}}[\![\mathtt{x1}]\!]_\rho \cup \ldots \cup {}^{\mathcal{E}}[\![\mathtt{xn}]\!]_\rho\right)$$

$$
\begin{aligned}
{}^{\mathcal{E}}[\![\texttt{y:=f(x1,...,xn)}]\!]^{\phi}_{\rho} = \; & |{}^{\mathcal{E}}[\![\texttt{Y}]\!]_{\rho}| = f|{}^{\mathcal{E}}[\![\texttt{x1}]\!]_{\rho}, \ldots, {}^{\mathcal{E}}[\![\texttt{xn}]\!]_{\rho}| \quad \wedge \\
& \textstyle\bigwedge_{i=1}^{n} {}^{\mathcal{E}}[\![\texttt{xi}]\!]_{\rho} \xrightarrow{\widehat{{}^{\mathcal{E}}[\![\texttt{y}]\!]_{\rho}}} {}^{\mathcal{E}}[\![\texttt{y}]\!]_{\rho}
\end{aligned}
$$

$$
\begin{aligned}
{}^{\mathcal{E}}[\![\texttt{z:=x when y}]\!]^{\phi}_{\rho} = \; & {}^{\mathcal{E}}[\![\texttt{z}]\!]_{\rho} \mathrel{\widehat{=}_v} {}^{\mathcal{E}}[\![\texttt{x}]\!]_{\rho} \otimes {}^{\mathcal{E}}[\![\texttt{y}]\!]_{\rho} \quad \wedge \\
& {}^{\mathcal{E}}[\![\texttt{x}]\!]_{\rho} \xrightarrow{\widehat{{}^{\mathcal{E}}[\![\texttt{x}]\!]_{\rho}} \sqcap [{}^{\mathcal{E}}[\![\texttt{y}]\!]_{\rho}]} {}^{\mathcal{E}}[\![\texttt{z}]\!]_{\rho}
\end{aligned}
$$

$$
\begin{aligned}
{}^{\mathcal{E}}[\![\texttt{z:=x default y}]\!]^{\phi}_{\rho} = \; & {}^{\mathcal{E}}[\![\texttt{z}]\!]_{\rho} \mathrel{\widehat{=}_v} {}^{\mathcal{E}}[\![\texttt{x}]\!]_{\rho} \oplus {}^{\mathcal{E}}[\![\texttt{y}]\!]_{\rho} \quad \wedge \\
& {}^{\mathcal{E}}[\![\texttt{x}]\!]_{\rho} \xrightarrow{\widehat{{}^{\mathcal{E}}[\![\texttt{x}]\!]_{\rho}}} {}^{\mathcal{E}}[\![\texttt{z}]\!]_{\rho} \quad \wedge \\
& {}^{\mathcal{E}}[\![\texttt{y}]\!]_{\rho} \xrightarrow{\widehat{{}^{\mathcal{E}}[\![\texttt{y}]\!]_{\rho}} \setminus \widehat{{}^{\mathcal{E}}[\![\texttt{x}]\!]_{\rho}}} {}^{\mathcal{E}}[\![\texttt{z}]\!]_{\rho}
\end{aligned}
$$

$$
{}^{\mathcal{E}}[\![\texttt{y:=x\$ init v}]\!]^{\phi}_{\rho} = \mathbf{Pre}_{\rightarrow}([\![\texttt{v}]\!], {}^{\mathcal{E}}[\![\texttt{x}]\!]_{\rho}, {}^{\mathcal{E}}[\![\texttt{y}]\!]_{\rho})
$$

Fig. 11. Semantics of primitive processes

Process schemes are denoted by the enrichment of environments ρ and ϕ with heading declarations (Note that ρ and ϕ are empty for the main process):

$[\![$`(?x1,...,xm !y1,...,yn)` P `where`

`process p1=`D_1`;`

...

`process pp=`D_p

`end`$]\!]^{\phi}_{\rho} =$

$$
\left\{ \begin{aligned} & \mathcal{E} \mid \exists (x_1, \ldots, x_m, y_1, \ldots, y_n) \in \mathcal{S}_{\mathcal{E}},\, \mathcal{E} = (\textstyle\bigcup_{i=1}^{m} x_i) \cup (\bigcup_{j=1}^{n} y_j) \quad \wedge \\ & \qquad\qquad\qquad\qquad [\![P]\!]^{\phi, \texttt{p1} \mapsto [\![D_1]\!]^{\phi}_{\rho}, \ldots, \texttt{pp} \mapsto [\![D_p]\!]^{\phi}_{\rho}}_{\rho, \texttt{x1} \mapsto x_1, \ldots, \texttt{xm} \mapsto x_m, \texttt{y1} \mapsto y_1, \ldots, \texttt{yn} \mapsto y_n} \end{aligned} \right\}
$$

One will find the original trace semantics of SIGNAL [15,16] by restricting our semantics to real signals. This semantics improves the original trace semantics by adding a notion of least upper bound for non-deterministic processes. It can then deal elegantly with data dependence and refinement of SIGNAL processes.

6 Conclusions

We have defined a unified model which formalises all aspects of the development of a reactive system using the underlying programming methodology of synchronous languages. This model uses basic notions of preorder theory and category theory and has been partially specified and validated using the COQ theorem prover [6]. Synchronous structures allow to model non-determinism with imaginary signals and clocks. The set of clocks is completed with imaginary clocks to form a boolean lattice. Thus, any pair of clocks always has a least upper bound. In our model, absence is not treated as a special value: It is consistent with reality. Synchronous structures can also deal elegantly with data dependence and refinement of synchronous specifications to model the compilation of a synchronous language.

Acknowledgment We are grateful to Jean-Pierre Talpin and Paul Le Guernic for working on an earlier version of this work, to Paul Caspi, Simon Gay and Rajagopal Nagarajan for useful discussions, and to anonymous reviewers for their comments.

References

[1] D. Nowak, J.-P. Talpin, P. Le Guernic, Synchronous structures, in: CONCUR '99: Concurrency Theory, 10th International Conference, Eindhoven, The Netherlands, August 24-27, 1999, Proceedings, Vol. 1664 of Lecture Notes in Computer Science, Springer, 1999, pp. 494–509.

[2] G. Berry, G. Gonthier, The Esterel synchronous programming language: Design, semantics, implementation, Science of Computer Programming 19 (2) (1992) 87–152.

[3] N. Halbwachs, P. Caspi, P. Raymond, D. Pilaud, The synchronous dataflow programming language LUSTRE, Proceedings of IEEE 79 (9) (1991) 1305–1320.

[4] A. Benveniste, P. L. Guernic, C. Jacquemot, Synchronous programming with events and relations: The SIGNAL language and its semantics, Science of Computer Programming 16 (2) (1991) 103–149.

[5] D. Nowak, J.-R. Beauvais, J.-P. Talpin, Co-inductive axiomatization of a synchronous language, in: Proceedings of the 11th International Conference on Theorem Proving in Higher Order Logics (TPHOLs'98), Vol. 1479 of Lecture Notes in Computer Science, Springer, 1998, pp. 387–399.

[6] The Coq Development Team, The coq proof assistant reference manual, version 6.2, Tech. rep., INRIA (1998).

[7] A. Pnueli, N. Shankar, E. Singerman, Fair synchronous transition systems and their liveness proofs, in: Formal Techniques in Real-Time and Fault-Tolerant Systems, Vol. 1486 of Lecture Notes in Computer Science, Springer-Verlag, 1998, pp. 198–209.

[8] Benveniste, Caillaud, L. Guernic, Compositionality in dataflow synchronous languages: Specification and distributed code generation, Information and Computation 163 (1) (2000) 125–171.

[9] J.-P. Talpin, A. Benveniste, B. Caillaud, C. Jard, Z. Bouziane, H. Canon, BDL, a language of distributed reactive objects, in: Proceedings of the International Symposium on Object-Oriented Real-time Distributed Computing (ISORC 98), 1998, pp. 196–205.

[10] P. Caspi, Clocks in dataflow languages, Theoretical Computer Science 94 (1) (1992) 125–140.

[11] S. J. Gay, R. Nagarajan, Modelling Signal in Interaction Categories, in: Theory and Formal Methods 1993: Proceedings of the First Imperial College Department of Computing Workshop on Theory and Formal Methods, Springer-Verlag Workshops in Computer Science, 1993, pp. 148–158.

[12] S. Abramsky, Interaction Categories (Extended Abstract), in: Theory and Formal Methods 1993: Proceedings of the First Imperial College Department of Computing Workshop on Theory and Formal Methods, Springer-Verlag Workshops in Computer Science, 1993, pp. 57–70.

[13] G. Cardano, *Ars Magna*, 1545.

[14] N. Halbwachs, Synchronous programming of reactive systems, Kluwer Academic Publishers, 1993.

[15] I. M. Smarandache, Transformations affines d'horloges: application au codesign de systèmes temps-réel en utilisant les langages SIGNAL et ALPHA, Ph.D. thesis, Université de Rennes 1 (Oct. 1998).

[16] D. Nowak, Spécification et preuve de systèmes réactifs, Ph.D. thesis, Université de Rennes 1, 176 pages (Oct. 1999).

[17] G. Winskel, Event structures, Petri nets: applications and relationships to other models of concurrency, Springer Lecture Notes in Computer Science 255 (1987) 325–392.

[18] S. Abramsky, S. Gay, R. Nagarajan, Specification structures and propositions-as-types for concurrency, j-LECT-NOTES-COMP-SCI 1043 (1996) 5–??

[19] S. M. Lane, Categories for the Working Mathematician, Vol. 5 of Graduate Texts in Mathematics, Springer-Verlag, New York, NY, 1971.