CCSL denotational semantics

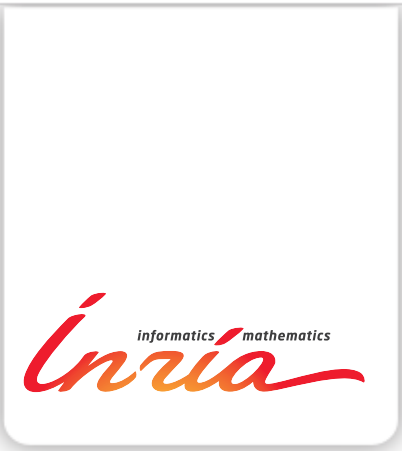
Julien Deantoni, Charles André, Régis Gascon

CCSL denotational semantics

Julien DeAntoni, Charles André, Régis Gascon

RESEARCH REPORT
N° 8628
November 2014
Project-Team Aoste

ISSN 0249-6399 ISRN INRIA/RR--8628--FR+ENG



CCSL denotational semantics

Julien DeAntoni, Charles André, Régis Gascon*

Project-Team Aoste

Research Report n° 8628 — November 2014 — 26 pages

Abstract: The Clock Constraint Specification Language (CCSL) has been informally introduced in the specifications of the UML Profile for Modeling and Analysis of Real-Time and Embedded systems (MARTE). In a previous report entitled "Syntax and Semantics of the Clock Constraint Specification Language", we equipped a kernel of CCSL with an operational semantics. In the present report we pursue this clarification effort by giving a mathematical characterization to each CCSL constructs.

Key-words: CCSL, UML, time constraints, semantics, denotational

* Université de Nice Sophia Antipolis

RESEARCH CENTRE
SOPHIA ANTIPOLIS – MÉDITERRANÉE

2004 route des Lucioles - BP 93
06902 Sophia Antipolis Cedex

Sémantique dénotationnelle de CCSL

Résumé : Le langage de Spécification de Contraintes d'Horloges (connu sous le nom de Clock Constraint Specification Language ou sous l'acronyme CCSL) a été introduit de façon informelle dans le document de spécification du profile UML pour la modélisation et l'analyse des systèmes temps réel et embarqués (MARTE). Dans un précédent rapport intitulé "Syntax and Semantics of the Clock Constraint Specification Language" nous avons défini une sémantique opérationnelle pour un noyau de CCSL. Le présent rapport poursuit cet effort de formalisation en donnant une caractérisation mathématique précise à chaque élément du langage CCSL.

Mots-clés : CCSL, UML, contraintes temporelles, sémantique, dénotationnelle

1 Introduction

Modeling of distributed systems, as well as electronic systems with multi-cores or multi-clock domains, needs multiple time bases. The UML profile for *Modeling and Analysis of Real-Time and Embedded* systems [1] (MARTE) addresses this modeling issue through its rich *model of Time*. MARTE also introduces the concept of *clock constraints* and proposes the non-normative language CCSL (short for *Clock Constraint Specification Language*) for specifying such constraints.

The MARTE time model has been first presented at the 10^{th} international conference on *Model Driven Engineering Languages and Systems* [2]. This paper introduced the concepts of time bases, *clocks* and clock constraints. CCSL appeared only in a few illustrative examples. On the other hand, the OMG specification of MARTE contains neither a precise syntax nor a formal semantics for CCSL; only an informal (English) semantics is given. This lack of formal description of CCSL has been partially filled in our research report entitled *"Syntax and Semantics of the Clock Constraint Specification Language"* [3], which described a kernel of CCSL and provided a structural operational semantics for this kernel. The present report contibutes further to the semantics of CCSL by giving full mathematical characterization of the clock relations and clock expressions of CCSL.

CCSL has also been used in *verification of time requirements* [4, 5]. The contribution of the present report is to define a denotational semantics for CCSL. This semantics introduce formally the notion of birth and death of both clock constraints and clocks. The goal of this formalization is to pave the road of using CCSL with different modes, where each mode is a node of a an automaton.

The report consists of three main sections. The first section introduces the *multiform logical* time, the *clocks*, and the *clock constraints*. A clock constraint is a *clock relation* that applies to two *clock expressions*. In this section, all the clock relations from the kernel CCSL are mathematically defined. The primitive clock expressions of the kernel CCSL are characterized in the same way.

The second section first explains how libraries can be constructed over the CCSL kernel. The third section describes the perspectives opened by such a formal modeling approach.

2 Clock Constraint Specification Language

2.1 Multiform logical time

MARTE Time model deals with both discrete and dense times. In MARTE, a *clock* gives access to a *time structure* made of *time bases*, which are themselves ordered sets of *instants*. A clock can be either *dense* or *discrete*. This report focuses on the structural relations between clocks and these relations do not differentiate between dense and discrete clocks. However, some relations only apply to discrete clocks and others apply to both discrete and dense clocks. Logical clocks refer to discrete-time logical clocks and represent *logical time*.

Leslie Lamport [6] introduced logical clocks in the late 70's. The logical clocks associate numbers (logical timestamps) with events in a distributed system, such that there exists a consistent total ordering of all the events of the system. These clocks can be implemented by counters with no actual timing mechanisms. In the 80's, the synchronous languages [7] introduced their own concept of logical time. This logical time shares with Lamport's time the fact that they need not actually refer to physical time. Logical time only relies on (partial or total) ordering of instants. In what follows, we consider logical time in the sense of synchronous languages. In the synchronous language Signal [8], a *signal* s is an infinite totally ordered sequence $(s_t)_{t\in\mathbb{N}}$ of typed elements. Index t denotes a *logical instant*. At each logical instant of *its* clock, a signal is present and carries a unique value. Signal is a multi-clock (or polychronous) language: it does not assume

the existence of a *global clock*. Instead, it allows multiple logical clocks. Signal composition is ruled by operators which are either mono-clock operators (composing signals defined on a same clock) or multi-clock operators (allowing composition of signals having different clocks).

Indeed, a logical clock can be associated with any event. This point of view has been adopted in the MARTE time model [1, Chap. 10]. A logical clock "ticks" with each new occurrence of its associated event. Synchronous languages like Esterel exploit this property. In an Esterel program, time may be counted in seconds, meters, laps... (see the Berry's RUNNER program [9] which describes the training of a runner). This variety of events supporting time leads to the concept of *multiform time*. More technical examples can be found in automotive applications. For instance, the electronic ignition is driven by the angular position of the crankshaft rather than by a chronometric time (see our study of a knock controller in a 4-stroke engine [10]).

In this report, we consider both dense and logical clocks and their relationships through *clock constraints*.

Because the notion of instant and the ordering of such instants are the core of the proposition, we first introduce the relations we defined on the instants.

2.2 Instants and Time Structure

Instants are the basic elements which are manipulated all along this report. An instant can be seen as an event occurrence, occurring at a specific (logical) time. Expliciting the fact that an instant as a specific logical time implies that instants can be partially ordered. To do so, we use specific instant relations. An instant relation between two instants reflects a specific ordering between the considered instants. We defined five kinds of instant relations, derived from two main relations.

Causal and chronological relations over a set of instant are defined in a *time structure*. A time structure $\mathcal{T}$ is a triple $\langle \mathcal{I}, \prec, \equiv \rangle$ made of a set of instants $\mathcal{I}$ and two main relations on instants verying the conditions below.

- The *precedence relation* $\prec$ is a strict order relation on I (irreflexive, asymmetric and transitive)[1].
- The *coincidence relation* $\equiv$ is an equivalence relation (reflexive, symmetric and transitive). It reflects the fact that two instants have the exactly same logical time.
- The precedence and coincidence relations verify the following properties: for every $i_1, i_2, i_3 \in I$
  - if $i_1 \prec i_2$ and $i_2 \equiv i_3$ then $i_1 \prec i_3$,
  - if $i_1 \prec i_2$ and $i_1 \equiv i_3$ then $i_3 \prec i_2$.

From these two relations, one can define additional relations associated to any time structure:

- The *causality relation* $\preccurlyeq$ is defined as the union of the precedence relation and the coincidence relation.
$$\forall i_1, i_2 \in I,\ i_1 \preccurlyeq i_2 \Leftrightarrow (i_1 \prec i_2 \text{ or } i_1 \equiv i_2).$$
Note that $\preccurlyeq$ is not really a partial order. By definition, this relation is reflexive and transitive, but it is not anti-symmetric. If $i_1 \preccurlyeq i_2$ and $i_2 \preccurlyeq i_1$ we can deduce that $i_1 \equiv i_2$ but coincidence does not imply equality.

[1] Classical properties of binary relations are recalled in Appendix A.1.

- The *exclusion relation* $\#$ is defined as the union of the precedence relation and its converse relation. Formally, we have

$$\forall i_1, i_2 \in I,\ i_1 \# i_2 \Leftrightarrow (i_1 \prec i_2 \text{ or } i_2 \prec i_1).$$

  This impose that instants i_1 and i_2 must not be coincident.

- Finally, the independence relation $\|$ corresponds to the absence of other relations.

$$\forall i_1, i_2 \in I,\ i_1 \| i_2 \Leftrightarrow \text{ neither } (i_1 \prec i_2) \text{ nor } (i_2 \prec i_1) \text{ nor } (i_1 \equiv i_2).$$

  This relation can be used to express concurrency between two events.

The graphical representation of these different instant relations is given in Table 1.

instant relation	symbol	graphical representation
precedence	$\prec$	
causality	$\preccurlyeq$	
coincidence	$\equiv$	
exclusion	$\#$	
independence	$\|\|$	no link

Table 1: Instant relations

% clocks.tex

2.3 Clocks

We adopt a simplified model compared to the one defined in MARTE. However, this model is still linked to MARTE and has already been used in our papers (see for instance [11]). This model considers that a clock is an ordered set of instants. A clock has a lifetime delimited by a birth instant and a death instant.

Formally, a *Clock* c is a 5-tuple $\langle \mathcal{I}_c, \prec_c, c^{\uparrow}, c^{\downarrow}, \equiv_{c^{\downarrow}} \rangle$ where $\mathcal{I}_c$ is a possibly infinite set of instants, $c^{\uparrow} \notin \mathcal{I}_c$ is the birth instant of c, and $c^{\downarrow} \notin \mathcal{I}_c$ is the death instant of c, $\equiv_{c^{\downarrow}}$ is a coincidence relation and $\prec_c$ is an order relation on $\mathcal{I}_c \uplus \{c^{\uparrow}, c^{\downarrow}\}$ satisfying the following conditions:

- The restriction of $\prec_c$ on $\mathcal{I}_c$ is a strict order relation.

- All instants of $\mathcal{I}_c$ are strictly ordered ($\prec_c$ is total):

$$\forall i, j \in \mathcal{I}_c,\ (i \neq j) \Rightarrow (i \prec_c j) \text{ or } (j \prec_c i).$$

- The birth strictly precedes all the other instants of the clock:

$$\forall i \in \mathcal{I}_c \uplus \{c^{\downarrow}\},\ c^{\uparrow} \prec_c i.$$

- Every instant precedes the death but the last one that may coincide with it:

$$\forall i \in \mathcal{I}_c \uplus \{c^{\uparrow}\},\ ((i \prec_c c^{\downarrow}) \vee (i \equiv_{c^{\downarrow}} c^{\downarrow}))$$

The set of instants $\mathcal{I}_c$ represents the occurrences or *ticks* of the clock c. This is why birth and death does not belong to this set. A clock can have a finite or infinite number of instants. If $\mathcal{I}_c$ is infinite then the death instant is not necessary.

A *discrete-time clock* c is a clock with a discrete set of instants $\mathcal{I}_c$. In that case, $\mathcal{I}_c$ can be indexed by natural numbers in a fashion that respects the ordering $\prec_c$: we define $\text{idx}_c : \mathcal{I}_c \to \mathbb{N}^\star$ ($\mathbb{N}^\star = \mathbb{N} \setminus \{0\}$) such that $\forall i \in \mathcal{I}_c$, $\text{idx}(i) = k$ iff i is the k^{th} instant in $\mathcal{I}_c$ wrt. $\prec_c$. By convention we will consider that the first instant of c is indexed by 1.

For any discrete time clock $c \triangleq \langle \mathcal{I}_c, \prec_c, c^\uparrow, c^\downarrow, \equiv_{c^\downarrow} \rangle$, $c[k]$ denotes the k^{th} instant in $\mathcal{I}_c$ (*i.e.*, $\text{idx}_c(c[k]) = k$). For any instant $i \in \mathcal{I}_c$ of a discrete time clock c, $^\circ i$ is the unique immediate predecessor of i in $\mathcal{I}_c \uplus \{c^\uparrow\}$. We assume that the predecessor of $c[1]$ is the birth $c^\uparrow$. Similarly we denote the unique immediate successor of i in $\mathcal{I}_c$ as i°, if any.

The number of instants preceding a specific instant i of a clock c is retrieved by $\chi(c)@i$. It only makes sense on discrete clock since it always return ∞ on dense clock. Note that the use of this function needs i to be strictly ordered on the set of instants it refers (*i.e.*,$\mathcal{I}_c$ in the previous example). From the previous definition we give the following lemmas:

$$
\begin{aligned}
&\langle \mathcal{I}, \prec, \equiv \rangle \models \chi(c) && \Leftrightarrow \\
&\quad 0)\ (i \in \mathcal{I}) \wedge (\mathcal{I}_c \in \mathcal{I}) \wedge (\exists j, k \in \mathcal{I}_c \uplus \{c^\uparrow\}) \wedge (j \prec k) \wedge (j \prec i) \wedge (i \prec k) && \wedge \\
&\quad 1)\ \big(c \in C \wedge \chi(c)@i = k\big) \Leftrightarrow c[k] \preccurlyeq i \prec c[k+1] && \wedge \\
&\quad 2)\ \big(c \in C, \forall k \in [1; |\mathcal{I}_c|]\big) \Leftrightarrow \chi(c)@c[k] = k - 1 && \wedge \\
&\quad 3)\ \chi(c)@c^\uparrow = 0 && \wedge \\
&\quad 4)\ \chi(c)@c^\downarrow = |\mathcal{I}_c|
\end{aligned}
\tag{1}
$$

The reader can notice that the χ function is defined only if i is ordered with the set of instant of c (cf. equation 1.0).

2.4 Clock Constraints Specification

We have introduced the concept of *clock constraints* in the MARTE specification (chapter 9) and also a dedicated language for expressing such constraints: CCSL [1, Annex C.3]. This language is non normative (the MARTE profile implementors are not obliged to support it). The semantics of CCSL given in the MARTE specification is informal. A first formal semantics, based on mathematical expressions has been proposed in a paper [12] and a research report [13], which is an extended version of the paper. A precise definition of the syntax of a *kernel* of CCSL along with a structural operational semantics is now available [3]. This semantics is the golden reference for the CCSL constraint solver implemented in TimeSquare [14][2], the software environment that supports CCSL and the MARTE time profile.

Clock constraints are classified into two main categories:

1. clock relations that constrain the order of the instant of two or more existing clocks;
2. clock expressions that define a new clock from a set of parameters.

As for clocks, every relation and expression has a timelife defined by the instants of its *birth* and its *death*. We define in more details these elements in the following.

A CCSL specification $\mathcal{S}$ is a triple $\langle C, Expr, Rel \rangle$ such that C is a set of clocks, *Expr* is a set of *clock definitions* and *Rel* is a set of *clock relations*. Both *Expr* and *Rel* state constraints on the elements of C.

[2] `http:timesquare.inria.fr`

The models of CCSL specification are time structures. Let $\mathcal{S} = \langle C, Expr, Rel \rangle$ be a CCSL specification and $\mathcal{T} = \langle I, \prec, \equiv \rangle$ a time structure. Informally, $\mathcal{T}$ satisfies the specification $\mathcal{S}$ iff the following conditions holds:

- The set of instants I includes all the instants of the different clocks in C.
- The orders of the different clocks in C are preserved by $\prec$.
- The different constraints induced by the elements of *Expr* and *Rel* are satisfied.

The satisfaction of the different clock relations and definitions will be defined in the following sections and depends of the relations $\prec$ and $\equiv$.

2.5 Definition of a specific subset of instants

Since relations and expressions are alive from their birth instant and possibly until their death instant, it is useful to define the set of instants of a clock between these two instants. An interval of clock c (*i.e.*,a subset of $\mathcal{I}_c$) characterized by two instants α and β is denoted as $\mathcal{I}_c^{\alpha..\beta}$. It is defined by the following equation:

$$\mathcal{I}_c^{\alpha..\beta} \triangleq \{i \in \mathcal{I}_c | (\alpha \preccurlyeq i) \wedge \neg(\beta \prec i)\} \tag{2}$$

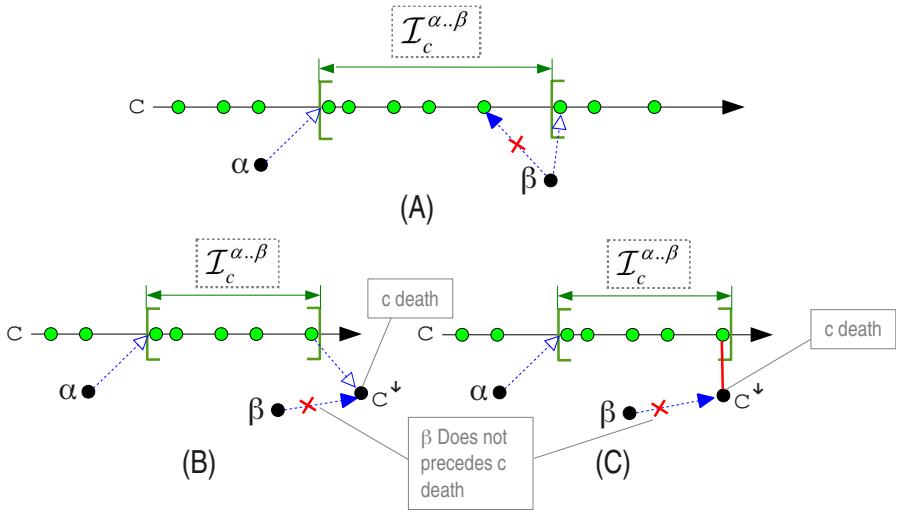


Figure 1: Clock interval definitions

The interval starts with the first instant of $\mathcal{I}_c$ such that α precedes this instant. The interval ends at the first instant of $\mathcal{I}_c$ which is strictly preceded by β; this instant *is not in* the interval. The latter constraint is expressed by $\neg(\beta \prec i)$ for all instants of $\mathcal{I}_c$ in the interval. Note that since $\preccurlyeq$ is a partial order, $\neg(\beta \prec i)$ is not equivalent to $(i \preccurlyeq \beta)$.

When β is not given, the interval (denoted as $\mathcal{I}_a^{\alpha\cdot\cdot}$) is defined as follows:

$$\mathcal{I}_c^{\alpha\cdot\cdot} \triangleq \{i \in \mathcal{I}_c | (\alpha \preccurlyeq i)\} \tag{3}$$

Fig. 2 illustrates various cases: without the clock death (Fig. 1.A) or with the clock death (Fig. 1.B and C).

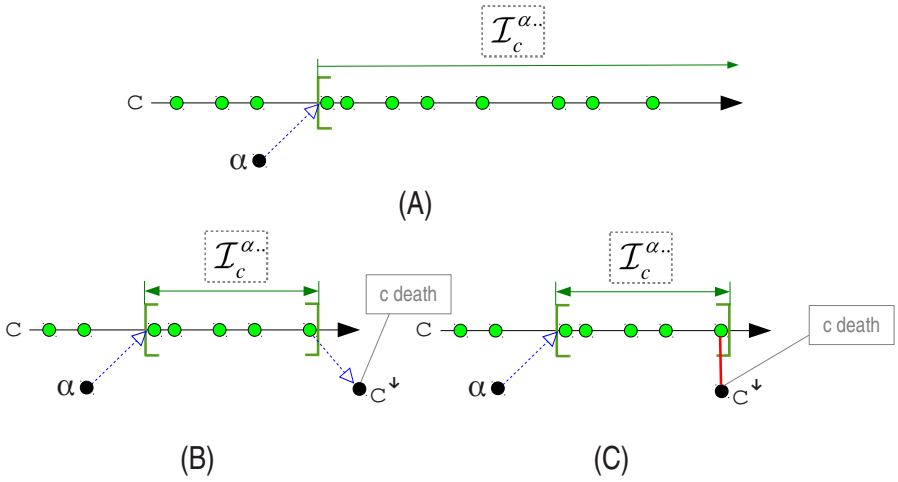


Figure 2: Clock interval when only α is given

These intervals are used in the following to represent the instants to be considered during the life of a specific relation or expression. Consequently, the set of instants of a clock c during the life of a relation r is noted $\mathcal{I}_c^r$. Similarly the set of instants of a clock c during the life of an expression is noted $\mathcal{I}_c^{ce}$ where ce is the clock defined by the expression. More generally, we denote $\mathcal{I}_c^x$ the set of instants of c that is defined on an interval bounded by $x^{\uparrow}$ and, either $c^{\downarrow}$ or $x^{\downarrow}$. More formally, $\mathcal{I}_c^x$ is defined by the following equation:

$$\mathcal{I}_c^x \triangleq \{(\exists x^{\downarrow} \in x \Rightarrow \mathcal{I}_c^x = \mathcal{I}_c^{x^{\uparrow}..x^{\downarrow}}) \vee (\nexists x^{\downarrow} \in x \Rightarrow \mathcal{I}_c^x = \mathcal{I}_c^{x^{\uparrow}..})\} \tag{4}$$

Predicate dies_in Let dies_in be a predicate on *clock* × *constraint* such that

$$c \text{ dies\_in } (r) \Leftrightarrow (r^{\uparrow} \preccurlyeq c^{\downarrow}) \wedge \left(\exists r^{\downarrow} \in r \Rightarrow \neg(r^{\downarrow} \prec c^{\downarrow})\right) \tag{5}$$

2.6 Clock relations

The clock relations presented in this subsection form the kernel all binary clock relations. Let a and b two clocks. Five primitive relations on clocks are defined. For each of them the definition is split into a first part named 'a)' defining the semantics of the relation and a second part 'b)' specifying the relations between deaths of clocks in the relations.

Subclocking: $a \overset{r}{\boxed{\subset}} b$ is a synchronous clock relation. a is said to be a sub-clock of b, and b a super-clock of a. The r represents the entity that gives the birth and the death of the relation (*i.e.*,the interval on which the relation applies). If r is not specified, the relation applies for the whole system life.

$$\begin{aligned} &\langle \mathcal{I}, \prec, \equiv \rangle \models a \overset{r}{\boxed{\subset}} b \Leftrightarrow \\ &(a) \quad \forall i_a \in \mathcal{I}_a^r,\ \exists i_b \in \mathcal{I}_b^r,\ i_a \equiv i_b \quad \wedge \\ &(b) \quad b \text{ dies\_in } (r) \Rightarrow (a^{\downarrow} \preccurlyeq b^{\downarrow}). \end{aligned} \tag{6}$$

Equation 6-a defines the constraints between instants of $\mathcal{I}_a$ and $\mathcal{I}_b$. It means that each instant of a must coincide with an instant of b. Equation 6-b states that a must die when the superclock dies .

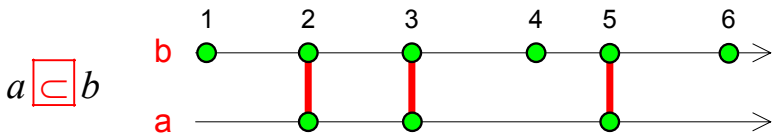


Figure 3: Example of subclocking.(For a sake of simplicity $\chi(a)@r^{\uparrow} = 0$ and $\chi(b)@r^{\uparrow} = 0$)

Note that an equivalent definition of this relation is: there is an order preserving mapping $h : \mathcal{I}_a^r \to \mathcal{I}_b^r$ such that for every $i \in \mathcal{I}_a^r$ we have $i \equiv h(i)$. However, the properties of the relations $\prec$ and $\equiv$ simplify the definition. Additionally, note that the death of a coincides with the one of b if b dies in r and a is not killed by another relation in the time structure. Because of the possibly to be killed by another relation, the death of a precedes the death of b.

Strict precedence: $a \overset{r}{\boxed{\prec}} b$ is an asynchronous clock relation. a is said to be strictly faster than b, and b strictly slower than a.

$$
\begin{aligned}
&\langle \mathcal{I}, \prec, \equiv \rangle \models a \overset{r}{\boxed{\prec}} b \Leftrightarrow \\
&(a) \quad \exists h : \mathcal{I}_b^r \to \mathcal{I}_a^r, \\
&\qquad \big(\forall i \in \mathcal{I}_b^r,\ (h(i) \prec i) \ \wedge\ \forall i, j \in \mathcal{I}_b^r,\ (i \prec j) \Rightarrow (h(i) \prec h(j))\big) \quad \wedge \\
&(b) \quad (\neg(b^{\downarrow} \preccurlyeq r^{\uparrow}) \wedge a \ \mathsf{dies\_in}\ (r) \wedge h \text{ is bijective}) \Rightarrow b \ \mathsf{dies\_in}\ (r).
\end{aligned}
\tag{7}
$$

Equation 8-a imposes that each instant of $\mathcal{I}_b^r$ (slowest clock) is preceded by a distinct instant of $\mathcal{I}_a^r$. Equation 8-b implies that b dies in r if a is dead and b is not late any more (and if b was not dead before entering the interval). The condition "h is bijective" is to ensure that one has the same number of a an b instants; even in the dense case.

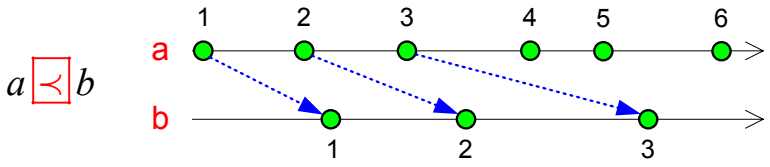


Figure 4: Exemple of precedence.(For a sake of simplicity $\chi(a)@r^{\uparrow} = 0$ and $\chi(b)@r^{\uparrow} = 0$)

Causality: $a \overset{r}{\boxed{\preccurlyeq}} b$ is similar to the previous one but considers the causality relation instead. a is said to cause b, and b depends on a.

$$
\begin{aligned}
&\langle \mathcal{I}, \prec, \equiv \rangle \models a \overset{r}{\boxed{\preccurlyeq}} b \Leftrightarrow \\
&(a) \quad \exists h : \mathcal{I}_b^r \to \mathcal{I}_a^r, \\
&\qquad \big(\forall i \in \mathcal{I}_b^r,\ (h(i) \preccurlyeq i) \ \wedge\ \forall i,j \in \mathcal{I}_b^r,\ (i \prec j) \Rightarrow (h(i) \prec h(j))\big) \quad \wedge \\
&(b) \quad (\neg(b^{\downarrow} \preccurlyeq r^{\uparrow}) \wedge a \ \mathsf{dies\_in}\ (r) \wedge h \text{ is bijective}) \Rightarrow b \ \mathsf{dies\_in}\ (r).
\end{aligned}
\tag{8}
$$

The constraints are similar to precedence.

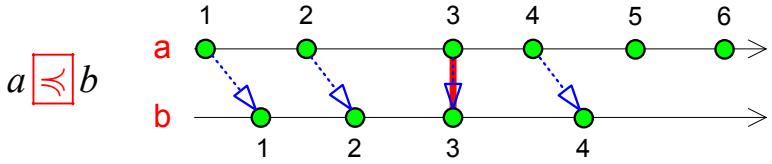


Figure 5: Example of causality relation.
(For a sake of simplicity $\chi(a)@r^{\uparrow} = 0$ and $\chi(b)@r^{\uparrow} = 0$)

In this example, note that $a[3]$ and $b[3]$ are coincident.

Exclusion: $a \overset{r}{\boxed{\#}} b$ means that a and b have no coincident instants.

$$
\begin{aligned}
&\langle \mathcal{I}, \prec, \equiv \rangle \models a \overset{r}{\boxed{\#}} b \Leftrightarrow \\
&\qquad \forall i \in \mathcal{I}_a^r,\ \forall j \in \mathcal{I}_b^r,\ i \# j.
\end{aligned}
\tag{9}
$$

The reader can notice that there is no death propagation in this relation.

Equality: $a \overset{r}{\boxed{=}} b$ is a typical synchronous clock relation that means that each instant of a coincides with one instant of b. This relation holds when both a is a sub-clock of b, and b is a sub-clock of a.

$$
\begin{aligned}
&\langle \mathcal{I}, \prec, \equiv \rangle \models a \overset{r}{\boxed{=}} b \Leftrightarrow \\
&(a) \quad \forall i \in \mathcal{I}_a^r,\ \exists j \in \mathcal{I}_b^r,\ i \equiv j \\
&\qquad \wedge\ \forall i \in \mathcal{I}_b^r,\ \exists j \in \mathcal{I}_a^r,\ i \equiv j \qquad\qquad \wedge \\
&(b) \quad \big((a \ \mathsf{dies\_in}\ r) \vee (b \ \mathsf{dies\_in}\ r)\big) \Rightarrow (a^{\downarrow} \equiv b^{\downarrow}).
\end{aligned}
\tag{10}
$$

Note that 10-a expresses that a is subclock of b and vice versa. Equation 10-b states that the death of a and the death of b are coincident.

2.7 Clock expressions

A clock expression defines a new implicit clock. In order to assign this result to an explicit clock, we used clock definitions of the form $c \triangleq expr$ where *expr* is one of the expressions whose semantics is defined in this section. Consequently, the corresponding expression born and death respectively coincides with the born and the death of c. For all expressions, if other expressions or relations are used for their definition, their birth and their death respectively coincide with the birth and the death of c.

2.7.1 Index independent clock expressions

Clock union: $a + b$ represents a clock that ticks whenever a or b ticks. This new clock is the minimal (w.r.t. $\preccurlyeq$) super clock of both a and b.

$$\begin{array}{ll} \langle \mathcal{I}, \prec, \equiv \rangle \models c \triangleq a + b \Leftrightarrow & \\ (a) \quad \forall i_a \in \mathcal{I}_a^c,\ \exists i \in \mathcal{I}_c,\ (i \equiv i_a)) \wedge \forall i_b \in \mathcal{I}_b^c,\ \exists i \in \mathcal{I}_c,\ (i \equiv i_b)) & \\ (b) \quad \wedge\ \forall i \in \mathcal{I}_c,\ (\exists i_a \in \mathcal{I}_a^c,\ (i \equiv i_a)) \vee (\exists i_b \in \mathcal{I}_b^c,\ (i \equiv i_b)) & \\ (c) \quad \wedge\ c^{\downarrow} \preccurlyeq \max_{\preccurlyeq}\{a^{\downarrow}, b^{\downarrow}\}. & \end{array} \tag{11}$$

The function $\max_{\preccurlyeq}$ is defined as follows: $\max_{\preccurlyeq}\{i, j\}$ is equal to i if $j \preccurlyeq i$, to j if $i \prec j$. If i and j are idependent both solution are valid. The function $\min_{\preccurlyeq}$ is defined similarly.

Equation 11-a states that both a and b are subclocks of c and 11-b that every instants of c coincides either with a or b (minimal subclcok). Equation 11-c expresses that c dies iff both a and b are dead. The death of c coicindes with the death of the last clock. Note that a consequence of this definition is that in any time structure satisfying $c \triangleq a + b$, the clocks a and b are totally ordered since the instants of c are.

Clock intersection: $a * b$ defines a clock that ticks whenever both a or b tick.

$$\begin{array}{ll} \langle \mathcal{I}, \prec, \equiv \rangle \models c \triangleq a * b \Leftrightarrow & \\ (a) \quad \forall i \in \mathcal{I}_c,\ \exists i_a \in \mathcal{I}_a^c,\ \exists i_b \in \mathcal{I}_b^c,\ \big((i \equiv i_a) \wedge (i \equiv i_b)\big) & \\ (b) \quad \forall i_a \in \mathcal{I}_a^c,\ \forall i_b \in \mathcal{I}_b^c,\ \big((i_a \equiv i_b) \Rightarrow (\exists i \in \mathcal{I}_c,\ (i \equiv i_a)\big) & \\ (c) \quad \wedge\ c^{\downarrow} \preccurlyeq \min_{\preccurlyeq}\{a^{\downarrow}, b^{\downarrow}\}. & \end{array} \tag{12}$$

Equation 12-a state that c is a sub-clock of both a and b. Equation 12-b makes c maximal with respect to the subclocking relation. Equation 12-c states that c must die as soon as either a or b dies.

Figure 6 illustrates the two clock expressions just defined. note that in this case the death of c coincides with the death of a clock in the parameters but nothing prevents something else (*i.e.*,another constraint) to kill the clock before.

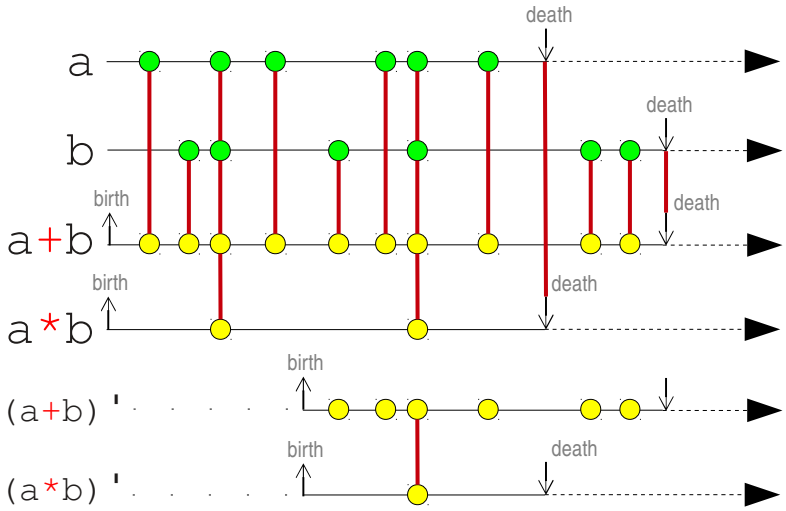


Figure 6: Exemples of clock expressions union and intersection.

2.7.2 Index dependent clock expressions

Sup: $a \vee b$ defines a clock that is the fastest among all the clocks slower than a and b (equation 13). In other terms, the resulting clock always coincides with the slowest clock among a and b, in an opportunistic manner.

Let $\mathcal{T} = \langle \mathcal{I}, \prec, \equiv \rangle$ and $\mathcal{T}' = \langle \mathcal{I}', \prec', \equiv' \rangle$ be two time structures. In the equation below, $\mathcal{T} \subseteq \mathcal{T}'$ means that $\mathcal{I} \subseteq \mathcal{I}'$ and the relations $\prec'$ and $\equiv'$ respectively extend $\prec$ and $\equiv$. Also, we note $d \in \mathcal{T}$ for a clock d iff $\mathcal{I}$ includes the instants of d and $\prec$ respects their ordering.

$$
\begin{array}{ll}
\mathcal{T} \models c \triangleq a \vee b \Leftrightarrow & \\
(a) & \mathcal{T} \models a \preccurlyeq c \wedge \mathcal{T} \models b \preccurlyeq c \\
(b) & \wedge\ \forall \mathcal{T}' \text{ st. } \mathcal{T} \subseteq \mathcal{T}',\ \forall d \in \mathcal{T}', \\
 & \quad \big((\mathcal{T}' \models a \preccurlyeq d) \wedge \mathcal{T}' \models b \preccurlyeq d)\big) \Rightarrow \mathcal{T}' \not\models c \prec d. \\
(c) & \wedge |\mathcal{I}_c| = min(|\mathcal{I}_a^c|, |\mathcal{I}_b^c|) \\
 & \wedge\big(\ (|\mathcal{I}_c| = |\mathcal{I}_a^c| \wedge a \text{ dies\_in } c) \Rightarrow c^{\downarrow} \preccurlyeq max_{\preccurlyeq}(c[|\mathcal{I}_a^c|], a^{\downarrow}) \\
 & \quad \vee (|\mathcal{I}_c| = |\mathcal{I}_b^c| \wedge b \text{ dies\_in } c) \Rightarrow c^{\downarrow} \preccurlyeq max_{\preccurlyeq}(c[|\mathcal{I}_b^c|], b^{\downarrow})\ \big)
\end{array}
\tag{13}
$$

Equation 13-b means that a and b are faster than c. Equation 13-b states that there is no extension of $\mathcal{T}$ including a clock d being slower that a and be b and fsater than c. We need to introduce extensions of $\mathcal{T}$ to ensure that d can have distinct elements from the other clocks. Finally, 13-c express that c dies with the same number of ticks than the clock with the minimum number of ticks. Death of c occurs either at the death of this clock if c was following it or when it has the same number of ticks in the other case. Note that the death propagation makes sense only if a and b are discrete clocks.

Inf: $a \wedge b$ defines a clock that is the slowest among all the clocks faster than a and b (equation 14). The resulting clock always coincides with the fastest clock among a and b, in an

opportunistic manner.

$$
\begin{array}{ll}
\mathcal{T} \models c \triangleq a \wedge b \Leftrightarrow & \\
(a) \quad \mathcal{T} \models c \preccurlyeq a \wedge \ \mathcal{T} \models c \preccurlyeq b & \\
(b) \quad \wedge\, \forall \mathcal{T}' \text{ st. } \mathcal{T} \subseteq \mathcal{T}',\ \forall d \in \mathcal{T}', & \\
\qquad \big((\mathcal{T}' \models d \preccurlyeq a) \wedge\ (\mathcal{T}' \models d \preccurlyeq b)\big) \Rightarrow \mathcal{T}' \not\models d \prec c. & \\
(c) \quad \wedge\, c^{\downarrow} \preccurlyeq \max_{\preccurlyeq}\{a^{\downarrow}, b^{\downarrow}\}. &
\end{array}
\tag{14}
$$

The different parts of this equation are similar to the previous case but the death propagation where it is not necessary to delay the death depending on the clock c was following.

Figure 7 shows the sup and inf of two clocks.

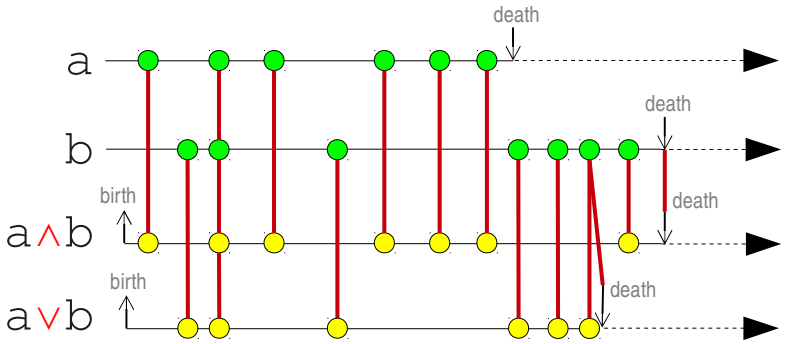


Figure 7: Exemples of clock expressions sup and inf.

Strict sampling: $a \Downarrow b$ is a mixed clock expression (*i.e.*,based on both precedence and coincidence). It defines a subclock of b that ticks whenever clock a has ticked at least once since the previous tick of b. For this expression to make sense, b must be a discrete clock.

Figure 8 shows the corresponding instant ordering.

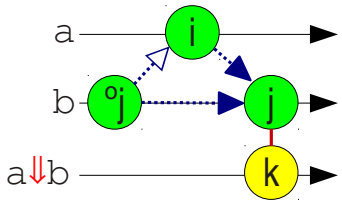


Figure 8: Strict sampling instant ordering.

$$
\begin{array}{ll}
\langle \mathcal{I}, \prec, \equiv \rangle \models c \triangleq a \Downarrow b \Leftrightarrow & \\
(a) \quad \forall i \in \mathcal{I}_c,\ \exists i_a \in \mathcal{I}_a^c,\ i_b \in \mathcal{I}_b^c,\ (^\circ i \prec i_a) \wedge (i_a \prec i) \wedge (i \equiv i_b) & \\
(b) \quad \wedge\, c^{\downarrow} \preccurlyeq \min_{\preccurlyeq}\{b^{\downarrow},\ i_b \text{ st. } i_b \in \mathcal{I}_b^c \wedge (^\circ i_b \prec a^{\downarrow}) \wedge (a^{\downarrow} \prec i_b)\}. &
\end{array}
\tag{15}
$$

Equation 15-a says that c is a subclock of b and that there is always an occurrence of a between two occurrences of c (cf. Figure 8).

Non strict sampling: $a \Downarrow\!\!\downarrow b$ is also a mixed clock expression, similar to the previous one, just changing the precedence relations to non-strict ones. The clock b still must be a discrete clock.

Figure 9 shows the corresponding instant ordering.

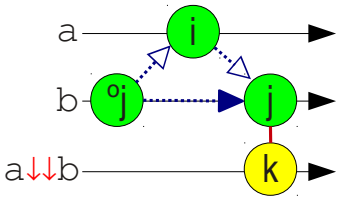


Figure 9: Non strict sampling instant ordering.

$$
\begin{aligned}
&\langle \mathcal{I}, \prec, \equiv \rangle \models c \triangleq a \Downarrow\!\!\downarrow b \Leftrightarrow \\
&(a) \quad \forall i \in \mathcal{I}_c,\ \exists i_a \in \mathcal{I}_a^c,\ i_b \in \mathcal{I}_b^c,\ (^\circ i_b \prec i_a) \wedge (i_a \preccurlyeq i_b) \wedge (i \equiv i_b) \\
&(b) \quad \wedge\ c^\downarrow \preccurlyeq \min_{\preccurlyeq}\{b^\downarrow,\ i_b \text{ st. } i_b \in \mathcal{I}_b^c \wedge (^\circ i_b \prec a^\downarrow) \wedge (a^\downarrow \preccurlyeq i_b)\}.
\end{aligned}
\tag{16}
$$

Figure 10 highlights the different behaviors of the strict and non strict clock expression samplings.

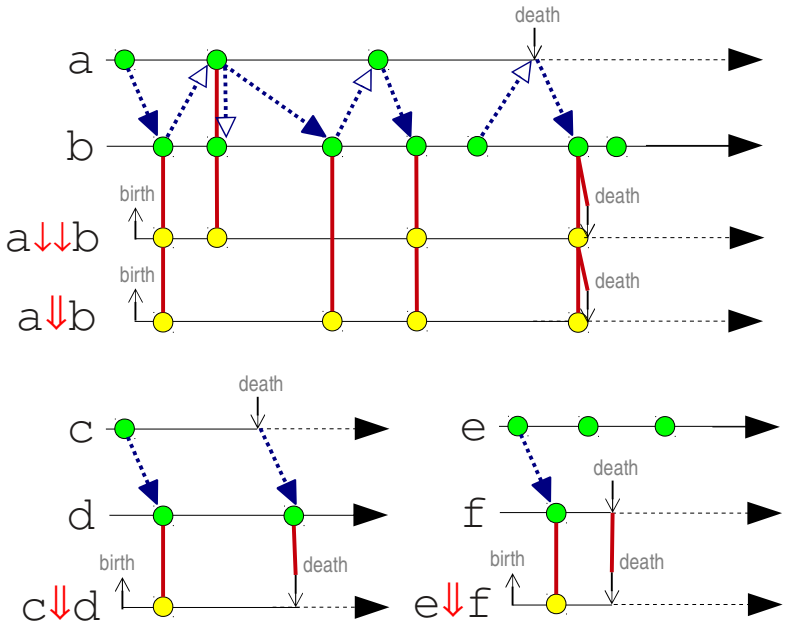


Figure 10: Exemple of clock sampling.

2.7.3 Clock expressions involving the c clock death

Upto: $a \downarrow b$ defines a clock that ticks whenever a ticks upto the first tick of b. As of this tick, the resulting clock dies. To make sence, we require that b is a discrete clock.

$$\begin{array}{ll} \langle \mathcal{I}, \prec, \equiv \rangle \models c \triangleq a \downarrow b \Leftrightarrow & \\ (a) \quad \forall i \in \mathcal{I}_c,\ \exists i_a \in \mathcal{I}_a^c,\ (i \equiv i_a) & \\ (b) \quad \wedge\, \forall i_a \in \mathcal{I}_a^c,\ (\forall i_b \in \mathcal{I}_b^c,\ \neg(i_b \preccurlyeq i_a)) \Rightarrow (\exists i \in \mathcal{I}_c,\ (i \equiv i_a) & \\ (c) \quad c^{\downarrow} \preccurlyeq \min_{\preccurlyeq}\{a^{\downarrow},\ i_b \text{ st. } i_b \in \mathcal{I}_b^c \wedge \forall j_b \in \mathcal{I}_b^c,\ (i_b \preccurlyeq j_b)\}. & \end{array} \tag{17}$$

Equation 17-a enforce c to be coincident with a and 17-b that every instants of a that is not preceded by the first instant of b corresponds to an instant of c. Equation 17-c means that c dies either with the death of a or the first occurrence of b if it precedes the death of a.

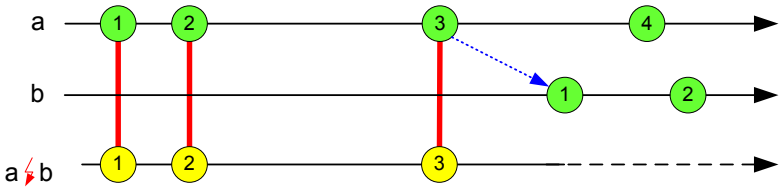


Figure 11: Exemples of clock expression upto.
(For a sake of simplicity $\chi(a)@c^{\uparrow} = 0$ and $\chi(b)@c^{\uparrow} = 0$)

CE clock, is defined on page 15.

2.7.4 Clock expressions with non-clock parameters

In the previous expressions, all the parameters were clock. We consider here expressions with other kind of parameter type, for instance *Integer* or *IntegerWords* (see Annex B). In the following expressions non-clock parameters have the meaning of a schedule that are used to create a list of events or tasks and the times at which each one should happen or be done. More precisely, the schedule associated with an expression contains the future times at which the resulting clock should tick in coincidence with a parameter clock. evaluated

Awaiting: $a \hat{}\, n$, where $n \in \mathbb{N}^{\star}$, is a synchronous clock expression. This expression waits for the n^{th} strictly future tick of a. On this occurrence, the resulting clock ticks and dies. As one have to count the occurrences of a, the clock must be discrete.

$$\begin{array}{ll} \langle \mathcal{I}, \prec, \equiv \rangle \models c \triangleq a \hat{}\, n \Leftrightarrow & \\ (a) \quad (\exists i_a \in \mathcal{I}_a^c,\ idx(i_a) = n) \Rightarrow (\mathcal{I}_c = \{i_c\} \wedge (i_c \equiv a[n]) \wedge c^{\downarrow} \equiv a[n]) & \\ (b) \quad \wedge\, (\nexists i_a \in \mathcal{I}_a^c,\ idx(i_a) = n) \Rightarrow (\mathcal{I}_c = \{\} \wedge c^{\downarrow} \preccurlyeq a^{\downarrow}) & \end{array} \tag{18}$$

Equation 18-a defines the case where $a[n]$ exists: in that case $\mathcal{I}_c$ is a singleton whose element coincides with $a[n]$, and the death of c also coincides with $a[n]$. If $a[n]$ does not exist (18-b), $\mathcal{I}_c$ is empty and c dies with a.

figure 12).

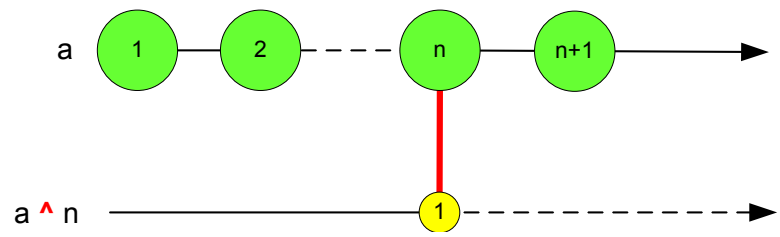


Figure 12: Exemple of clock awaiting.(For a sake of simplicity $\chi(a)@c^{\uparrow} = 0$)

Defer: $a \overset{(w)}{\rightsquigarrow} b$ is also a mixed clock expression that deals with multiple future scheduled ticks. w is an IntegerWord defined, coinjointly with notations over w,in the annex B. Defer is a complex expression. A simplist explanation is to consider that each ticks of the clock a specifies that a ticks of the resulting clock will coincides after n ticks of b; where n is taken in w (cf. 19-a below). For this expression to make sense, a and b must be discrete.

$$
\begin{aligned}
&\langle \mathcal{I}, \prec, \equiv \rangle \models c \triangleq a \overset{(w)}{\rightsquigarrow} b \Leftrightarrow \\
&(a) \quad \exists h : \mathcal{I}_c \to \mathcal{I}_b, \\
&\qquad \forall i \in \mathcal{I}_c,\ (i \equiv h(i)) \\
&\qquad \wedge\ \forall i_a \in \mathcal{I}_a,\ \exists i_c \in \mathcal{I}_c \exists X \subseteq \mathcal{I}_b,\ \big((|X| = w(idx(i_a) - 1)) \\
&\qquad \wedge\ \forall i_b \in \mathcal{I}_b,\ (i_b \in X \Leftrightarrow i_a \prec i_b \preccurlyeq i_c)\big) \\
&(b) \quad c^{\downarrow} \equiv \min_{\preccurlyeq}\{b^{\downarrow},\ c[\min\{|w|, |\mathcal{I}_a|\}]\}.
\end{aligned}
\tag{19}
$$

Equation 19-b means that the death of c coincindes with the first of the following instant:

- the death of b (since c is subclock of b),
- the $|w|^{\text{th}}$ instant of c when w is finite (since there is no future scheduled occurence of c),
- the $|\mathcal{I}_a|^{\text{th}}$ of c (since no future scheduled occurence of c can be selected afterward).

Figure 13 shows clock expression 'defer' when n is a constant. In this case, the clock expression is also known as clock expression 'delayedFor'.

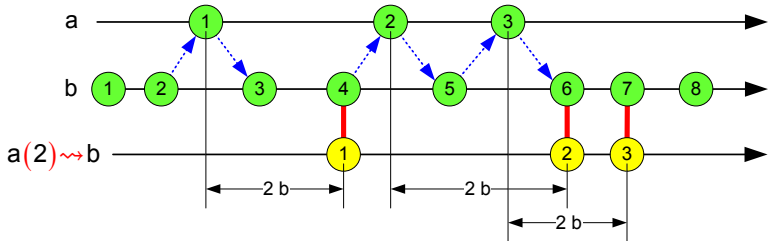


Figure 13: Exemple of clock expression 'defer'.

2.7.5 Clock expression concatenation

Will be defined in a future version

3 Conclusion

We presented in this document the denotational semantics of the CCSL language. Compared to older document, this report introduce an explicit notion of birth and death in the definition of a time system. The birth and death notions are implemented in TimeSquare [14] and used in the definition of the MoCCML operational semantics [15].

References

[1] OMG. *UML Profile for MARTE, v1.0*. Object Management Group, November 2009. Document number: formal/2009-11-02.

[2] C. André, F. Mallet, and R. de Simone. Modeling time(s). In G. Engels, B. Opdyke, D. C. Schmidt, and F. Weil, editors, *MoDELS*, volume 4735 of *Lecture Notes in Computer Science*, pages 559–573. Springer, 2007.

[3] Charles André. Syntax and semantics of the clock constraint specification language (CCSL). Research Report 6925, INRIA, 05 2009.

[4] Frédéric Mallet, Marie-Agnès Peraldi-Frati, and Charles André. Marte CCSL to execute East-ADL timing requirements. In *ISORC* [16], pages 249–253.

[5] Charles André and Frédéric Mallet. Specification and verification of time requirements with CCSL and esterel. In *LCTES*, pages 167–176. ACM, June 2009.

[6] Leslie Lamport. Time, clocks, and the ordering of events in a distributed system. *Commun. ACM*, 21(7):558–565, 1978.

[7] A. Benveniste and G. Berry. The synchronous approach to reactive and real-time systems. *Proceeding of the IEEE*, 79(9):1270–1282, September 1991.

[8] A. Benveniste, P. Le Guernic, and C. Jacquemot. Synchronous programming with events and relations: the SIGNAL language and its semantics. *Sci. Comput. Program.*, 16(2):103–149, 1991.

[9] Gérard Berry. *The Esterel Language Primer, version v5_91*. Ecole des Mines de Paris, CMA, INRIA, July 2000.

[10] C. André, F. Mallet, and M.-A. Peraldi-Frati. A multiform time approach to real-time system modeling: Application to an automotive system. In Universidade Nova de Lisboa, editor, *Int. Symp. on Industrial Embedded Systems*, pages 234–241, Lisboa, Portugal, July 2007. IEEE.

[11] Marie-Agnès Peraldi-Frati and Julien DeAntoni. Scheduling multi clock real time systems: From requirements to implementation. In *ISORC*, pages 50–57. IEEE Computer Society, 2011.

[12] Frédéric Mallet and Charles André. On the semantics of UML/marte clock constraints. In *ISORC* [16], pages 305–312.

[13] Charles André and Frédéric Mallet. Clock constraint specification language in UML/-MARTE CCSL. Research Report 6540, INRIA, 05 2008.

[14] Julien Deantoni and Frédéric Mallet. TimeSquare: Treat your Models with Logical Time. In Carlo A. Furia, Sebastian Nanz, editor, *TOOLS - 50th International Conference on Objects, Models, Components, Patterns - 2012*, volume 7304, pages 34–41, Prague, Czech Republic, May 2012. Czech Technical University in Prague, in co-operation with ETH Zurich, Springer.

[15] Julien Deantoni, Papa Issa Diallo, Joël Champeau, Benoit Combemale, and Ciprian Teodorov. Operational Semantics of the Model of Concurrency and Communication Language. Research Report RR-8584, I3S/INRIA AOSTE, September 2014.

[16] IEEE. *12th IEEE Int. Symp. on Object-Oriented Real-Time Distributed Computing (ISORC 2009)*. IEEE Computer Society, March 2009.

[17] Carl Adam Petri. Concurrency theory. In W. Brauer, W. Reisig and G. Rozenberg, editor, *Petri Nets: Central Models and their Properties*, volume 254 of *LNCS*. Springer-Verlag, 1987.

A Mathematical notions and notations

A.1 Binary relations

A binary relation R over X and Y is a subset of $X \times Y$:

$$R \subset X \times Y \tag{20}$$

X is called the *domain* of R; Y its *codomain*.

A.1.1 Relations over a set

If $X = Y$, $R \subset X^2$ is a binary relation over X (or *endorelation* over X). Some important classes of binary relations over X are:

$$R \text{ is reflexive } :\Leftrightarrow \forall x \in X, x \; R \; x \tag{21}$$

$$R \text{ is irreflexive } :\Leftrightarrow \forall x \in X, x \not R \; x \tag{22}$$

$$R \text{ is symmetric } :\Leftrightarrow \forall x, y \in X, x \; R \; y \Rightarrow y \; R \; x \tag{23}$$

$$R \text{ is antisymmetric } :\Leftrightarrow \forall x, y \in X, (x \; R \; y) \wedge (y \; R \; x) \Rightarrow x = y \tag{24}$$

$$R \text{ is asymmetric } :\Leftrightarrow \forall x, y \in X, (x \; R \; y) \Rightarrow \neg(y \; R \; x) \tag{25}$$

$$R \text{ is transitive } :\Leftrightarrow \forall x, y, z \in X, (x \; R \; y) \wedge (y \; R \; z) \Rightarrow (x \; R \; z) \tag{26}$$

$$R \text{ is total } :\Leftrightarrow \forall x, y \in X, (x \; R \; y) \vee (y \; R \; x) \tag{27}$$

$$R \text{ is an equivalence relation over } X :\Leftrightarrow R \text{ is reflexive, symmetric, and transitive} \tag{28}$$

$$R \text{ is a partial order relation over } X :\Leftrightarrow R \text{ is reflexive, antisymmetric, and transitive} \tag{29}$$

$$R \text{ is a total order relation over } X :\Leftrightarrow R \text{ is a partial order which is total} \tag{30}$$

A total order is also known as a Linear order.

A.1.2 Operations on Binary Relations

If R is a binary relation over X and Y, then the following is a binary relation over Y and X:

$$R^{-1} \text{ the converse of relation } R \text{ over } X \text{ and } Y \text{ is such that } R^{-1} = \{(y,x)|(x,y) \in R\} \tag{31}$$

If R is a binary relation over X, then each of the following relations is a binary relation over X:

$$\begin{aligned} &R^{=} \text{ the reflexive closure of relation } R \text{ over } X \text{ is defined as} \\ &\quad R^{+} = \{(x,x)|x \in X \in R\} \cup R \end{aligned} \tag{32}$$

$$\begin{aligned} &R^{+} \text{ the transitive closure of relation } R \text{ over } X \text{ is such that} \\ &\quad \text{the smallest transitive relation over } X \text{ containing } R. \end{aligned} \tag{33}$$

If R and S are binary relations over X and Y, then each of the following relations is a binary relation over Y and X:

$$R \cup S \subseteq X \times Y \text{ (union) is defined as } R \cup S = \big\{(x,y)|\big((x,y) \in R\big) \vee \big((x,y) \in S\big)\big\} \tag{34}$$

$$R \cap S \subseteq X \times Y \text{ (intersection) is defined as } R \cap S = \big\{(x,y)|\big((x,y) \in R\big) \wedge \big((x,y) \in S\big)\big\} \tag{35}$$

If R is a binary relation over X and Y, and S a binary relation over Y and Z, then the following relation is a binary relation over X and Z:

$$\begin{aligned} &R \circ S \subseteq X \times Z \text{ (composition) is defined as} \\ &\qquad R \circ S = \big\{(x,z)|\exists y \in Z, \big((x,y) \in R\big) \wedge \big((y,z) \in S\big)\big\} \end{aligned} \tag{36}$$

A.1.3 Other types of Binary Relations

If R and S are binary relations over X and Y:

$$R \text{ is injective } :\Leftrightarrow \forall x, z \in X, \forall y \in Y, (x\ R\ y) \wedge (z\ R\ y) \Rightarrow (x = z) \tag{37}$$

$$R \text{ is functional } :\Leftrightarrow \forall x \in X, \forall y, z \in Y, (x\ R\ y) \wedge (x\ R\ z) \Rightarrow (y = z) \tag{38}$$

$$R \text{ is one-to-one } :\Leftrightarrow R \text{ is both injective and functional} \tag{39}$$

$$R \text{ is left-total } :\Leftrightarrow \forall x \in X, \exists y \in Y, x\ R\ y \tag{40}$$

$$R \text{ is surjective } :\Leftrightarrow \forall y \in Y, \exists x \in X, x\ R\ y \tag{41}$$

$$R \text{ is a correspondence } :\Leftrightarrow R \text{ is both left-total and surjective} \tag{42}$$

$$R \text{ is a function } :\Leftrightarrow R \text{ is both left-total and functional} \tag{43}$$

$$R \text{ is a bijection } :\Leftrightarrow R \text{ is a one-to-one correspondence} \tag{44}$$

A.1.4 Strict partial order

In his paper entitled "Concurrency theory" [17] Petri defines a *strict partial order*.

A strict partial order: $\mathrm{SPO}(X, <)$ is defined as:

$$\begin{aligned}\mathrm{SPO}(X,<) :\Leftrightarrow\ & 1)\ \forall x,y,(x<y)\Rightarrow(x\in X)\wedge(y\in X)\\ \wedge\ & 2)\ \forall x\in X,\neg(x<x)\\ \wedge\ & 3)\ \forall x,y,z\in X,(x<y)\wedge(y<z)\Rightarrow(x<z)\end{aligned} \tag{45}$$

in words $<$ is an irreflexive, transitive relation in X; in shorthhand:

$$\begin{aligned}\mathrm{SPO}(X,<) :\Leftrightarrow\ & 1)\ <\ \subseteq X\times X\\ \wedge\ & 2)\ <\cap\,\mathrm{Id}=\emptyset\\ \wedge\ & 3)\ <^2\subseteq <\end{aligned} \tag{46}$$

In any strict partial order, we can define the disorder relation "neither $x < y$ nor $y < x$".

$$x\ \mathsf{co}\ y\Leftrightarrow(x,y\in X)\wedge\neg(x<y)\wedge\neg(y<x) \tag{47}$$

or for short $\mathsf{co} := \overline{<\cup>}$

It follows that co is (totally) reflexive and symmetric: $\mathrm{Id}\subseteq\mathsf{co}$ and $\mathsf{co}=\mathsf{co}^{-1}$. A structure (X,co) with these properties is called a *similarity*.

We now assert that, in general, co is not transitive: $\mathsf{co}^2-\mathsf{co}\neq\emptyset$.

Remark: a transitive similarity is an equivalence relation.

A.2 Functions

Functions are a special class of binary relations that are left-total and functional (Eq. 44).

Let A and B be sets.

$$\begin{aligned}&\text{A function from } X \text{ to } Y, \text{ denoted } f:X\to Y, \text{ is a relation from } X \text{ to } Y \text{ such that}\\ &\qquad\forall x\in X,\exists!y\in Y,(x,y)\in f\end{aligned} \tag{48}$$

If $(x, y) \in f$, then we write $f(x) = y$.

$$\begin{aligned}&\text{Let } f \text{ be a function from } X \text{ to } Y \text{ and let } X' \text{ be a subset of } X\\ &\quad f(X') \text{ denotes a subset of } Y, \text{ called the image of } X' \text{ under } f, \text{ such that}\\ &\quad f(X')=\{f(x)\mid x\in X'\}\end{aligned} \tag{49}$$

A.2.1 Special classes of functions

Let f be a function $f : X \to Y$.

$$f \text{ is surjective (onto) if } f(X)=Y \tag{50}$$

$$f \text{ is injective (one-to-one) if } \forall x,x'\in X, x\neq x'\Rightarrow f(x)\neq f(x') \tag{51}$$

$$f \text{ is bijective (one-to-one and onto) if } f \text{ is both surjective and injective} \tag{52}$$

A.3 Time related concepts

A.3.1 Clocks

$$\begin{array}{l}\text{A clock } c \text{ is a 5-tuple } \langle \mathcal{I}_c, \prec_c, c^{\uparrow}, c^{\downarrow}, \equiv_{c^{\downarrow}} \rangle \text{where} \\ \qquad \mathcal{I}_c \text{ is a possibly infinite set of instants,} \\ \qquad \prec_c \text{ is an order relation on } \mathcal{I}_c \uplus \{c^{\uparrow}, c^{\downarrow}\} \\ \qquad c^{\uparrow} \notin \mathcal{I}_c \text{ is the birth instant of } c, \text{ and} \\ \qquad c^{\downarrow} \notin \mathcal{I}_c \text{ is the death instant of } c \\ \qquad \equiv_{c^{\downarrow}} \text{ is a coincidence relation satisfying the following conditions:}\end{array} \tag{53}$$

- The restriction of $\prec_c$ on $\mathcal{I}_c$ is a strict order relation.
- All instants of $\mathcal{I}_c$ are strictly ordered ($\prec_c$ is total):

$$\forall i, j \in \mathcal{I}_c,\ (i \neq j) \Rightarrow (i \prec_c j) \text{ or } (j \prec_c i).$$

- The birth strictly precedes all the other instants of the clock:

$$\forall i \in \mathcal{I}_c \uplus \{c^{\downarrow}\},\ c^{\uparrow} \prec_c i.$$

- Every instants precedes the death but the last one that may coincide with it:

$$\forall i \in \mathcal{I}_c \uplus \{c^{\uparrow}\},\ ((i \prec_c c^{\downarrow}) \vee (i \equiv_{c^{\downarrow}} c^{\downarrow}))$$

.

A.3.2 Sets of instants

Let $\mathcal{C}$ be the set of clocks.

- Set of (actual) instants:

$$\mathcal{I} = \bigcup_{c \in \mathcal{C}} \mathcal{I}_c \tag{54}$$

- Extended set of instants (includes virtual ones):

$$\mathcal{I}^{+} = \mathcal{I}_c \uplus \{c^{\uparrow}, c^{\downarrow}\} \tag{55}$$

where $c^{\downarrow}$ is a virtual instant, greatest element of $\mathcal{I}^{+}$ (*i.e.*,$\forall i \in \mathcal{I}^{+}, i \preccurlyeq c^{\downarrow}$). $c^{\downarrow}$ stands for the (virtual) death instant of clocks c.

A.3.3 Relations on instants

Let $\prec$ (precedence) and $\equiv$ (coincidence) be two binary relations over $\mathcal{I}^{+}$.

$$\begin{array}{l}\text{Precedence is a strict partial order over } \mathcal{I}^{+} \\ \qquad \textit{i.e.,}\ \forall x, y, z \in \mathcal{I}+ : \\ \qquad\quad \neg(x \prec x) \text{ (irreflexivity)}, \\ \qquad\quad x \prec y \Rightarrow \neg(y \prec x) \text{ (asymmetry), and} \\ \qquad\quad x \prec y \wedge y \prec z \Rightarrow x \prec z \text{ (transitivity)}\end{array} \tag{56}$$

$$\begin{array}{l}\text{Coincidence is an equivalence relation over } \mathcal{I}^+\\ \quad i.e., \forall x,y,z \in \mathcal{I}+ :\\ \qquad (x \equiv x) \text{ (reflexivity)},\\ \qquad x \equiv y \Rightarrow (y \equiv x) \text{ (symmetry), and}\\ \qquad x \equiv y \wedge y \equiv z \Rightarrow x \equiv z \text{ (transitivity)}\end{array} \tag{57}$$

From these relations we derive three new relations over $\mathcal{I}^+$:

$$\text{Causality } \preccurlyeq \triangleq \prec \cup \equiv \tag{58}$$

$$\text{Exclusion } \# \triangleq \prec \cup \prec^{-1} \tag{59}$$

$$\text{Independence } \| \triangleq \overline{\preccurlyeq \cup \preccurlyeq^{-1}} \tag{60}$$

The graphical representation of instant relations is given in Table 2.

instant relation	symbol	graphical representation
precedence	$\prec$	
causality	$\preccurlyeq$	
coincidence	$\equiv$	
exclusion	#	
independence	$\|\|$	no link

Table 2: Instant relations

Remark

$$\forall c \in \mathcal{C}, \prec_c = \prec \cap (\mathcal{I}_c \times \mathcal{I}_c) \tag{61}$$

A.3.4 Intervals

Let α and β two instants of $\mathcal{I}^+$, and c a clock. We define special subsets of $\mathcal{I}_c$:

$$\mathcal{I}_c^{\alpha..\beta} \triangleq \{i \in \mathcal{I}_c | (\alpha \preccurlyeq i) \wedge \neg(\beta \preccurlyeq i) \wedge \left(\exists c^\dagger, \neg(\beta \preccurlyeq c^\dagger) \Rightarrow (i \preccurlyeq c^\dagger)\right)\} \tag{62}$$

When β is not given, the interval (denoted as $\mathcal{I}_a^{\alpha\cdot\cdot}$) is defined as follows:

$$\mathcal{I}_c^{\alpha\cdot\cdot} \triangleq \{i \in \mathcal{I}_c | (\alpha \preccurlyeq i) \wedge \left(\exists c^\dagger \Rightarrow (i \preccurlyeq c^\dagger)\right)\} \tag{63}$$

Note that elements of $\mathcal{I}_c^{\alpha..\beta}$ and $\mathcal{I}_c^{\alpha\cdot\cdot}$ are all actual instants (*i.e.*,members of $\mathcal{I}$).

Supremum of an interval

$$\begin{array}{l}s = \bigvee \mathcal{I}_c^{\alpha..\beta} \text{ is the supremum of } \mathcal{I}_c^{\alpha..\beta} \text{ iff}\\ \qquad 1)\ s \in \mathcal{I}^+\\ \wedge\ 2)\ s \text{ is an upper bound for } \mathcal{I}_c^{\alpha..\beta}\\ \qquad i.e., \forall i \in \mathcal{I}_c^{\alpha..\beta}, i \preccurlyeq s\\ \wedge\ 3)\ s \text{ is a least upper bound for } \mathcal{I}_c^{\alpha..\beta}\\ \qquad i.e., \forall s' \text{ upper bound of } \mathcal{I}_c^{\alpha..\beta}, s \preccurlyeq s'\end{array} \tag{64}$$

Note that because the coincidence relation is not the identity relation, the supremum is not necessarily unique: all the instants of a coincidence set can meet the above conditions.

A similar definition holds for $\mathcal{I}_c^{\alpha\cdot\cdot}$. $\bigvee \mathcal{I}_c^{\alpha..\beta}$ always exists thanks to the assumed existence of a greatest element (†) in $\mathcal{I}^+$.

Fig 14 shows different cases. In the discrete time case, the supremum of a non empty interval is the last instant of the interval. In the dense time cas, the supremum is the first instant of the next interval (*i.e.,*an instant not in the interval). When the interval is shortened by a death, then the (virtual) death instant is the supremum of the interval. Of course, the instant may coincide with an instant of the interval.

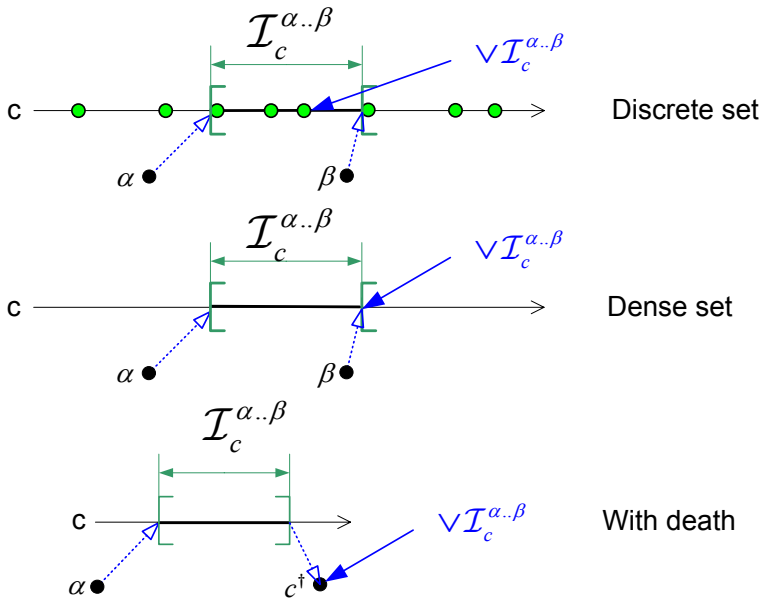


Figure 14: Supremum in different cases

B Integer Words

B.1 Finite/infinite words

Definition B.1 (Finite word)**.** A *finite word* is a word of $\forall i \in \mathbb{N}^\star, (i)^*$.

Definition B.2 (Infinite word)**.** An *infinite word* is a word of $\forall i \in \mathbb{N}^\star, (i)^\omega$.

Definition B.3 (Periodic word)**.** A *periodic word* is an infinite word defined by the following grammar:

$$\begin{aligned} & w ::= u\,(v)^\omega \\ \exists i, j \in \mathbb{N}^\star, & \\ & u ::= \varepsilon \mid i \bullet u \\ & v ::= i \bullet v \mid j \bullet v \end{aligned} \tag{65}$$

u is called the *prefix* of w, v is the *period* of w, and $(v)^\omega = \lim_n v^n$ denotes the infinite repetition of v. ε is the empty word. In order to avoid confusion between parentheses denoting periodic words and usual parentheses, the former are colored red. The associated ω symbol is also red.

A periodic word with an empty prefix is called a *strictly periodic word* ($w = (v)^\omega$).

A periodic word has infinitely many representations:

$$\text{Let } b \in \mathbb{N}^\star, u, v \in (\mathbb{N}^\star)^*, u \bullet b\,(v \bullet b)^\omega = u\,(b \bullet v)^\omega \tag{66}$$

Notation B.1 (Length of a word). $|w|$ denotes the length of the word w.

Notation B.2. $|w|_b$ denotes the number of integers set to $b \in \mathbb{N}^\star$ in the word w.

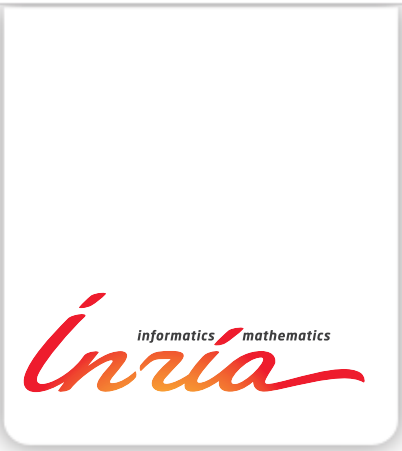
Notation B.3. $w\,[k]$ denotes the k^{th} integer of the word w.

Notation B.4. $w[k..l]$ denotes the (sub) word from w starting at the k^{th} integer upto the l^{th} integer included.

Notation B.5. $w[k..]$ denotes the (sub) word from w starting at the k^{th} integer. Possibly infinite.

$$\begin{aligned}\text{Let } k \in \mathbb{N}^\star, b \in \mathbb{B},\ w \text{ a word}&\\ w \downarrow 0 &\triangleq 0\\ b \bullet w \downarrow k &= b + (w \downarrow (k-1))\end{aligned} \tag{67}$$

Contents



RESEARCH CENTRE
SOPHIA ANTIPOLIS – MÉDITERRANÉE

2004 route des Lucioles - BP 93
06902 Sophia Antipolis Cedex

Publisher
Inria
Domaine de Voluceau - Rocquencourt
BP 105 - 78153 Le Chesnay Cedex
inria.fr

ISSN 0249-6399