Towards an algebra for timed behaviours*

Andrea Maggiolo-Schettini
Dipartimento di Informatica, Università di Pisa, Corso Italia 40, 56100 Pisa, Italy

Józef Winkowski
Instytut Podstaw Informatyki PAN, 00-901 Warszawa, PKiN, Skr.p. 22, Poland

Communicated by G. Rozenberg
Received March 1990
Revised November 1990

Abstract

Activities which proceed in a global time, called timed behaviours, are considered. A mathematical model for such activities is developed using a variant of labelled event structures. A compositional method of defining compound timed behaviours by combining simpler ones is presented. A concept of equivalence of timed behaviours is introduced.

1. Introduction

A way of thinking of complex behaviours is to specify what events are possible, in which order they may occur, and how the behaviour may branch.

Behaviours thus understood can be represented by event structures of the form \( \mathcal{E} = (E, \leq, \#) \), where \( E \) is a set of events, \( \leq \) is a causal order or quasiorder of events, and \( \# \) is a conflict relation between events (cf. [6–8]). Each event \( e \in E \) can be regarded as a particular execution of an action \( u \), written as \( \text{label}(e) \), and it can be identified with a pair \( (x, u) \), where \( x \) is a name of the execution. Consequently, the corresponding event structure can be regarded as a labelled event structure with the labelling given by the correspondence \( e \mapsto \text{label}(e) \). Simultaneous relations \( e \preceq f \) and \( f \preceq e \) in the case of causality given by a quasiorder represent a coincidence of events \( e \) and \( f \).

* This work has been supported by Consiglio Nazionale delle Ricerche and Instytut Podstaw Informatyki PAN.

0304-3975/92/$05.00 © 1992—Elsevier Science Publishers B.V. All rights reserved
An equivalent representation can be given by means of families of configurations of event structures, where a configuration of an event structure $E$ is a conflict-free prefix of $E$ or, more precisely, the set of events of such a prefix (cf. [7] and [8]). We illustrate this in Fig. 1 (events of an event structure and events of the corresponding family of configurations are represented by occurrences of the respective action symbols in the corresponding graphical representations).

![Configuration System](image)

Fig. 1. An event structure $E$ (left). The family of configurations of $E$ (right).

The families of configurations of event structures are members of an axiomatically defined class of systems, called configuration systems in this paper (a concept similar to that of families of configurations in [8], but slightly more general).

Complex behaviours can be obtained by combining simpler ones with the aid of operations similar to those of CCS on the corresponding event structures or configuration systems (cf. [7, 8]). One of such operations, called a parallel composition, can be defined with the aid of an operation of composing certain pairs of actions into joint actions.

A similarity of behaviours can be reflected by a suitable concept of behavioural equivalence (cf. [1, 2]). The type of similarity can be made dependent on a notion of visibility (or invisibility) of actions.

The assumed algebraic operation of composing actions and the assumed notion of invisibility can be given in the form of a structure in the universe of possible actions, called a synchronization structure. Such a structure is an analogue of synchronization algebras as in [7, 8].

In this paper we consider not only how events of behaviours follow or exclude each other, but also how they occur in a global time. The behaviours thus viewed are called timed behaviours.

We assume that events of timed behaviours are instantaneous and that timed behaviours are eager in the sense that their events occur as soon as possible, that is without any unjustified delay.

We assume also as a convention that for each timed behaviour time is counted relatively to the beginning of this behaviour.

Timed behaviours are represented by configuration systems with extra information about temporal aspects of events. Such information is given by specifying for each event the interval from enabling to completion, called the interval of waiting. This additional information about events allows us to say which of them can be considered as waiting for completion of events they coincide with, and which can be considered as critical in the sense that they trigger events in their coincidence classes. In
particular, due to this information we are able to formalize the concept of eagerness of timed behaviours.

The configuration systems with a temporal characterization of events can be regarded as specific configuration systems. To this end, it suffices to regard the intervals of waiting of events as features of the corresponding actions rather than of the events themselves. In this manner each event under consideration can be regarded as an execution of a *timed action*, where a timed action is an action together with an interval of waiting. Formally, such a timed action can be represented by a pair consisting of the corresponding action symbol and the interval of waiting. Of course, the causal order or quasiorder of events must be consistent with the completion times given by the intervals of waiting of the corresponding timed actions.

The configuration systems of the above described type are called *timed configuration systems*. We define them for a given synchronization structure. We will do this by constructing a special synchronization structure of timed actions and by considering configuration systems with actions from such a special structure.

For timed configuration systems we define operations similar to the ones for arbitrary configuration systems. This is achieved by modifying the operations defined for arbitrary configuration systems in the proper way, where the modification is done with the aid of the corresponding synchronization structure of timed actions.

The concept of behavioural equivalence for timed configuration systems is also similar to the one for arbitrary configuration systems and we define it by a slight modification of the latter.

The paper is organized with the idea to make it rather complete. Hence we also recall some known notions and results. In Section 2 we present the general concept of configuration systems and introduce timed configuration systems. In Section 3 we present the general definitions of operations on configuration systems and show how to modify them for timed configuration systems. In Section 4 we present a concept of equivalence of configuration systems and show how to modify it in order to obtain a concept suitable for timed configuration systems.

The present paper is an improved version of earlier works [3–5].

2. Configuration systems

2.1. The general concept

Configuration systems are members of an axiomatically definable class of systems of sets.

**Definition 2.1.** A *configuration system* (abbreviated: c-system) is a nonempty set $P$ of sets such that:

1. $\bigcap S \in P$ for each nonempty $S \subseteq P$ which is bounded in $P$ in the sense that some $p \in P$ contains all $s \in S$.
2. $\bigcup S \in P$ for each $S \subseteq P$ which is bounded in $P$. 
Sets $p \in P$ are called configurations of $P$. Members of such sets are called events (or nodes) of $P$. Given a configuration $p \in P$, each $p' \in P$ such that $p' \subseteq p$ is called a subconfiguration of $p$. A configuration which cannot be represented as the union of its proper subconfigurations is said to be indecomposable. Given a configuration $p \in P$ and two events $e, f \in p$, we say that $f$ follows $e$ in $p$ (resp.: $f$ is coincident with $e$ in $p$, $f$ follows strictly $e$ in $p$) iff, for all subconfigurations $p'$ of $p$, the condition $f \in p'$ implies (resp.: is equivalent to, implies but is not equivalent to) the condition $e \in p'$. By $\text{Nil}$ we denote the c-system $\{\emptyset\}$.

In our considerations a configuration system represents the set of possible states of development of a behaviour. Each state of development is characterized by the set of events due to which it has been reached and is represented in the form of a configuration. Each event represents a particular execution of an action.

Note that the relations of succession and coincidence of events are local in the sense that they are defined relatively to particular configurations. However, due to (1) of Definition 2.1, they are compatible with each other in the following way.

**Proposition 2.2.** If $P$ is a c-system, $p, p' \in P$ configurations such that $p \subseteq p'$, and $e, f \in p$, then $f$ follows $e$ in $p$ iff $f$ follows $e$ in $p'$.

In order to define operations on configuration systems, we distinguish a subclass of the class of configuration systems corresponding to a structure called a synchronization structure, a concept similar to that in [7] of a synchronization algebra.

A synchronization structure describes how actions of communicating behaviours compose into actions of a resulting behaviour and which actions are regarded to be invisible. The composability of an action with some others means that this action realizes a communication of a behaviour with its external world. The lack of composability means that the respective action is internal, that is, executable without any participation of external world. The invisibility of an action is a feature which is assigned to this action in order to declare it to be of no real interest in the respective description. This concept is meaningful only for internal actions but it need not coincide with the concept of internality. The reason of considering such an extra concept is that it may play the role of a parameter in defining various equivalences of behaviours.

**Definition 2.3.** A synchronization structure is $\Sigma = (U_\Sigma, \otimes_\Sigma, I_\Sigma)$, where $U_\Sigma$ is a set of action symbols, $\otimes_\Sigma$ is a strongly commutative and strongly associative partial binary operation in $U_\Sigma$ (that is an operation such that $u \otimes_\Sigma v = v \otimes_\Sigma u$ whenever either side is defined and $(u \otimes_\Sigma v) \otimes_\Sigma w = u \otimes_\Sigma (v \otimes_\Sigma w)$ whenever either side is defined), and $I_\Sigma$ is a subset of elements of $U_\Sigma$ such that $u \otimes_\Sigma v$ is not defined whenever $u \in I_\Sigma$ or $v \in I_\Sigma$. Action symbols $u, v \in U_\Sigma$ such that $u \otimes_\Sigma v$ is defined are said to be complementary. The action symbols without complementary ones are said to be internal. Those with complementary ones are said to be noninternal. The action symbols belonging to $I_\Sigma$ are said to be invisible.
Example 2.4. Let \( A \) be a set of symbols of data transfers in the handshaking mode, each transfer \( a \in A \) consisting of a send action \(-a\) and a receive action \(+a\). Let \( \tau \) denote an unspecified data transfer. For the actions thus represented we have a synchronization structure \( \Sigma_1 \) with \( U_{\Sigma_1} = -A \cup +A \cup \{\tau\} \), \( \otimes_{\Sigma_1} \) defined by \( (-a) \otimes_{\Sigma_1} (+a) = \tau \) for all \( a \in A \), and \( I_{\Sigma_1} = \{\tau\} \). In this case the concept of invisibility coincides with that of internality.

Example 2.5. Let \( L \) be a set of directed links via which certain objects called tokens can flow from one system to another. For each link \( l \in L \) let \(-l, +l, l\) denote respectively sending a token via \( l \), receiving a token from \( l \), and a complete transfer of a token via \( l \). For such actions we have a synchronization structure \( \Sigma_2 \) with \( U_{\Sigma_2} = -L \cup +L \cup L \), \( (-l) \otimes_{\Sigma_2} (+l) = l \) for all \( l \in L \), and \( I_{\Sigma_2} \) being a subset of \( L \). In this case the choice of \( I_{\Sigma_2} \) specifies the links for which the flow of tokens is of no interest.

For a synchronization structure \( \Sigma \) we define a subclass of configuration systems.

Definition 2.6. Given a synchronization structure \( \Sigma \), a configuration system over \( \Sigma \) (or a labelled c-system with labels from \( \Sigma \)) is a c-system \( P \) such that each configuration \( p \in P \) is a \( U_{\Sigma} \)-valued function (that is a set of pairs \((x, u)\) such that \( u \in U_{\Sigma} \) and the relations \((x, u) \in p \) and \((x, v) \in p \) imply \( u = v \)). For each event (node) \( e = (x, u) \) of \( P \) we write \( u \) as \( \text{label}(e) \). By \( es(\Sigma) \) we denote the universe of c-systems over \( \Sigma \).

Example 2.7. A place of a marked place/transition Petri net can be regarded as a bag into which tokens can be inserted via incoming links and from which residing tokens can be taken via outgoing links. The possible tokens are supposed to not interfere with each other. In particular, more than one token can be inserted or taken at a time. Consequently, the behaviour of a place can be described by numbering tokens which can possibly appear and by specifying for each possible state of the place the events due to which this state has been reached. These events are of four types: (1) emitting a token residing from the beginning (for a set \( X \) of such tokens), (2) receiving a token which remains residing (for a set \( Y \) of such tokens), (3) receiving a token which is next emitted before reaching the considered state (for a set \( Z \) of such tokens), and (4) emitting a token which has been received (for all the tokens belonging to \( Z \)). For a place with \( k \) tokens, incoming links \( a_1, \ldots, a_m \), and outgoing links \( b_1, \ldots, b_n \), this yields a c-system

\[
\text{place}_k(+a_1, \ldots, +a_m, -b_1, \ldots, -b_n)
\]

over a synchronization structure \( \Sigma_2 \) as in Example 2.5, namely the set of functions

\[
p : X \cup Y \cup Z \cup \{0\} \times Z \to U_{\Sigma_2}
\]
such that

1. \( X \subseteq \{1, \ldots, k\} \),
2. \( Y \) and \( Z \) are disjoint finite subsets of \( \{k+1, k+2, \ldots\} \),
3. \( p(x) \in \{-b_1, \ldots, -b_n\} \) for \( x \in X \),
   \( p(y) \in \{+a_1, \ldots, +a_m\} \) for \( y \in Y \),
   \( p(z) \in \{+a_1, \ldots, +a_m\} \) and \( p(0, z) \in \{-b_1, \ldots, -b_n\} \) for \( z \in Z \).

A configuration \( p \in \text{place},(+l_1, +l_2, -l_2, -l_3, -l_4) \) with \( X = \{1\} \), \( Y = \{3\} \), \( Z = \{4\} \), \( p(1) = -l_4 \), \( p(3) = +l_1 \), \( p(4) = +l_2 \) and \( p(0, 4) = -l_4 \) is illustrated in Fig. 2.

![Fig. 2. A configuration of place,(+l1, +l2, -l2, -l3, -l4) with indecomposable subconfigurations.](image)

**Example 2.8.** A transition of a place transition Petri net can be regarded as a procedure of absorbing a collection of tokens available via incoming links and next emitting a collection of tokens via outgoing links. The behaviour of a transition is regarded to consist of its possible executions, all the executions mutually independent. It can be described by numbering possible executions and by specifying for each state of such executions the events due to which this state has been reached. These events constitute collections of coincident events of three types: (1) collections of coincident events of absorbing tokens via all incoming links for executions which have not yet been completed (for a set \( X \) of such executions), (2) collections of coincident events of absorbing tokens via all incoming links for executions which have already been completed (for a set \( Y \) of such executions), and (3) collections of coincident events of emitting tokens via outgoing links for executions which have already been completed (that is for members of \( Y \)). For a transition with incoming links \( a_1, \ldots, a_m \) and outgoing links \( b_1, \ldots, b_n \) such that each execution absorbs one token via each incoming link and emits one token via each outgoing link this yields a c-system

\[
\text{trans}(+a_1, \ldots, +a_m, -b_1, \ldots, -b_n)
\]

over \( \Sigma_2 \) as in Example 2.5, namely the set of functions

\[
p: \{a_1, \ldots, a_m\} \times X \cup \{a_1, \ldots, a_m\} \times Y \cup \{b_1, \ldots, b_n\} \times Y \to U_{\Sigma_2}
\]
such that:

1. \( X \) and \( Y \) are finite disjoint subsets of \( \{1, 2, \ldots\} \),
2. \( p(a_i, x) = +a_i \) for \( x \in X \) and \( i \in \{1, \ldots, m\} \),
   \( p(a_i, y) = +a_i \) and \( p(b_j, y) = -b_j \) for \( y \in Y \), \( i \in \{1, \ldots, m\} \),
   and \( j \in \{1, \ldots, n\} \).

A configuration \( p \in \text{trans}(+l_4, +l_5, -l_1, -l_6) \) with \( X = \{5\} \), \( Y = \{3\} \), \( p(l_4, 5) = +l_4 \),
\( p(l_5, 5) = +l_5 \), \( p(l_4, 3) = +l_4 \), \( p(l_5, 3) = +l_5 \), \( p(l_1, 3) = -l_1 \), \( p(l_6, 3) = -l_6 \) is illustrated in Fig. 3.

Fig. 3. A configuration of \( \text{trans}(+l_4, +l_5, -l_1, -l_6) \) with indecomposable subconfigurations.

2.2. Timed configuration systems

Timed configuration systems are defined for a given synchronization structure \( \Sigma \). This is done by constructing for \( \Sigma \) a synchronization structure \( T(\Sigma) \) of timed actions and by defining timed configuration systems as members of a subclass of the class \( \text{cs}(T(\Sigma)) \) of all configuration systems over \( T(\Sigma) \).

The synchronization structure \( T(\Sigma) \) of timed actions for a synchronization structure \( \Sigma \) describes how actions of \( \Sigma \) may occur in time.

**Definition 2.9.** Given a synchronization structure \( \Sigma = (U, \otimes, I) \), the synchronization structure of timed actions for \( \Sigma \) is a synchronization structure \( T(\Sigma) = (U_{T(\Sigma)}, \otimes_{T(\Sigma)}, I_{T(\Sigma)}) \), where

1. \( U_{T(\Sigma)} \) is the set of pairs \( u = (\text{as}(u), \text{wait}(u)) \), called timed actions, such that
   1.1. \( \text{as}(u) \) is an element of \( U \) (an action symbol),
   1.2. \( \text{wait}(u) \) is a closed interval of nonnegative real numbers (a waiting interval) with a left end \( \text{entime}(u) \geq 0 \) (enabling time) and a right end \( \text{cptime}(u) \geq \text{entime}(u) \) (completion time);
2. \( u \otimes_{T(\Sigma)} v \) is defined whenever \( \text{as}(u) \otimes_{\Sigma} \text{as}(v) \) is defined and \( \text{cptime}(u) = \text{cptime}(v) \), and then \( u \otimes_{T(\Sigma)} v = w \), where
   \( \text{as}(w) = \text{as}(u) \otimes_{\Sigma} \text{as}(v) \),
   \( \text{entime}(w) = \max(\text{entime}(u), \text{entime}(v)) \),
   \( \text{cptime}(w) = \text{cptime}(u) = \text{cptime}(v) \);
3. \( I_{T(\Sigma)} = \{ u \in U_{T(\Sigma)} ; \text{as}(u) \in I \} \).
Timed configuration systems for a synchronization structure $\Sigma$ are defined as members of an axiomatically defined subclass of c-systems over $T(\Sigma)$, the synchronization structure of timed actions for $\Sigma$.

**Definition 2.10.** Given a synchronization structure $\Sigma$, a **timed configuration system** for $\Sigma$ is a c-system $P$ over $T(\Sigma)$ (the synchronization structure of timed actions for $\Sigma$) such that:

1. $\text{ctime}(\text{label}(e)) \leq \text{ctime}(\text{label}(f))$ whenever $f$ follows $e$ in some $p \in P$,
2. each configuration $p \in P$ is **eager** in the sense that each $e \in p$ with $\text{as}(\text{label}(e))$ being internal and $\text{entime}(\text{label}(e)) < \text{ctime}(\text{label}(e))$ is coincident with some $f \in p$ such that $\text{as}(\text{label}(f))$ is not internal or satisfies $\text{entime}(\text{label}(f)) = \text{ctime}(\text{label}(f))$.

For each event (node) $e$ of $P$ we define $\text{time}(e) = \text{ctime}(\text{label}(e))$, say that $e$ is **internal** iff $\text{as}(\text{label}(e))$ is internal, and say that $e$ is **critical** iff $\text{entime}(\text{label}(e)) = \text{ctime}(\text{label}(e))$. By $\text{tcs}(\Sigma)$ we denote the universe of timed c-systems for $\Sigma$.

The meaning of (1) is that an event which is a consequence of another one cannot precede it in time. The eagerness in (2) means that each event is either critical in the sense that it occurs when only enabled, or it is coincident with an event which is either critical or noninternal. The existence of a critical event in a coincidence class of events corresponds to triggering all the events of this class by an event which is enabled last and then executed immediately. The existence of a noninternal event (that is of an event which is not internal) can also be interpreted in a similar way since each noninternal event can be seen as a local image of an event executed with a participation of environment, and hence it can potentially be critical. Consequently, the requirement of eagerness prevents from unjustified delays in completing events.

Of course, we have $\text{tcs}(\Sigma) \subseteq \text{cs}(T(\Sigma))$.

**Example 2.11.** Suppose that $\Sigma 2$ is a synchronization structure as in Example 2.5. The timed behaviour or a place of a marked place/transition Petri net as in Example 2.7 can be represented by a timed c-system

$t\text{-place}_e(+a_1, \ldots, +a_m, -b_1, \ldots, -b_n) \in \text{tcs}(\Sigma 2)$,

namely by the set of functions

$p : X \cup Y \cup Z \cup \{0\} \times Z \to U_{Y(\Sigma 2)}$

such that

1. $X \subseteq \{1, \ldots, k\}$,
2. $Y$ and $Z$ are disjoint finite subsets of $\{k + 1, k + 2, \ldots\}$,
3. $\text{as}(p(x)) \subseteq \{-b_1, \ldots, -b_n\}$ and $\text{entime}(p(x)) = 0$ for $x \in X$,
4. $\text{as}(p(\xi)) \subseteq \{+a_1, \ldots, +a_m\}$ and $\text{entime}(p(\xi)) = 0$ for $\xi \in Y \cup Z$,
5. $\text{as}(p(0, z)) \subseteq \{-b_1, \ldots, -b_n\}$ and $\text{ctime}(p(z)) = \text{entime}(p(0, z)) \leq \text{ctime}(p(0, z))$ for $z \in Z$,

where the meanings of the sets $X, Y, Z$ are as in Example 2.7 (see Figs. 2 and 4).
Example 2.12. Suppose that each execution of a transition of a place-transition Petri net as in Example 2.8 takes a number $d$ of units of time. Suppose that $\Sigma 2$ is a synchronization structure as in Example 2.5. The the timed behaviour of the respective transition can be represented by timed c-system
\[
t\text{-trans}_d(+a_1,\ldots,+a_m,-b_1,\ldots,-b_n) \in \text{tcs}(\Sigma 2),
\]
namely by the set of functions
\[
p: \{a_1,\ldots,a_m\} \times X \cup \{a_1,\ldots,a_m\} \times Y \cup \{b_1,\ldots,b_n\} \times Y \to U_{t\Sigma 2},
\]
such that:

1. $X$ and $Y$ are finite disjoint subsets of $\{1,2,\ldots\}$.
2. \[\text{as}(p(a_i,\xi)) = +a_i \quad \text{and} \quad \text{entime}(p(a_i,\xi)) = 0\]
   and \[\text{cptime}(p(a_i,\xi)) = a(\xi) \quad \text{with some} \quad a(\xi) \geq 0\]
   for $\xi \in X \cup Y$ and $i \in \{1,\ldots,m\}$,
3. \[\text{as}(p(b_j,y)) = -b_j \quad \text{and} \quad \text{entime}(p(b_j,y)) = \text{cptime}(p(b_j,y)) = \text{cptime}(p(a_i,y)) + d\]
   for $y \in Y$, $i \in \{1,\ldots,m\}$, and $j \in \{1,\ldots,n\}$,

where the meanings of the sets $X$ and $Y$ are as in Example 2.8 (see Figs. 3 and 5).

Example 2.13. Consider a traffic light which can be set with the aid of a switch to one of two possible states, say $G$ (for “Green”) and $R$ (for “Red”), and which exhibits its current state such that this state can be received by certain users, say cars. Consider a synchronization structure $\Sigma 3$ such that
\[Wc, +Wc, -Rc, +Rc \in U_{\Sigma 3},\]

\((-Wc) \otimes \Sigma_3\) \((+Wc) = \tau\) and \((-Rc) \otimes \Sigma_3\) \((+Rc) = \tau\) for \(c \in \{G, R\}\), and \(I_{\Sigma_3} = \{\tau\}\), where \(+Wc\) stands for emitting an order \("go to c\) by the switch, \(-Wc\) stands for accepting an order \("go to c\) by the traffic light, \(-Rc\) stands for emitting \(c\), the current state, by the traffic light, and \(+Rc\) stands for receiving \(c\), the current state of the traffic light, by a car.

The timed behaviour of the traffic light with an initial state \(c_0\) can be represented by a timed \(c\)-system \(tlight(c_0) \in \text{tes}(\Sigma_3)\), namely the timed \(c\)-system which consists of functions \(p : X \cup Y \rightarrow U_{T_{\Sigma_3}}\), such that

1. \(X\) is a finite subset of \(\{1, 2, \ldots\}\),
2. \(Y\) is a finite subset of \(\{0, 1, \ldots\} \times \{0, 1, \ldots\}\),
3. \(i \in X\) or \((i + 1, j) \in Y\) implies \(\{1, \ldots, i\} \subseteq X\),
4. \(\text{entime}(p(1)) = 0\),
5. \(\text{as}(p(i)) = +Wc_i\) and \(\text{as}(p(i, j)) = -Rc_i\) with \(c_i \in \{G, R\}\)
   for all \(i \in X\) and \((i, j) \in Y\),
6. \(\text{entime}(p(i)) < \text{ctime}(p(i)) = \text{entime}(p(i+1))\) for all \(i \in X\).
7. \(\text{entime}(p(i, j)) = \text{entime}(p(i+1))\) for all \((i, j) \in Y\),
8. \(\text{ctime}(p(i-1, k)) < \text{ctime}(p(i)) < \text{ctime}(p(i, j))\)
   for all \(i, j, k \in \{0, 1, \ldots\}\) with \((i, j) \in Y\) and \((i-1, k) \in Y\).

Here \(i\) stands for going to the \(i\)th subsequent state and \((i, j)\) stands for one of a possible number of mutually independent acts of delivering the \(i\)th subsequent state to particular cars (see Fig. 6).

Fig. 6. A configuration of \(tlight(R)\) with subconfigurations.
For the switch we may assume any timed behaviour capable of setting a state of the traffic light. Such a behaviour can be represented by a timed c-system \texttt{switch} \in \texttt{tcs} (\Sigma 3) with configurations containing events of executing timed actions of the form \((-W_c, [x, y])\) with \(c \in \{G, R\}\) (see Fig. 7).

Finally, for a car we may assume any timed behaviour capable of looking at the traffic light and thus receiving the current state of the light. Such a behaviour can be represented by a timed c-system \texttt{car} \in \texttt{tcs} (\Sigma 3) with configurations containing sets of events of the form

\[
\{(0, (+RR, [t_0, t_0])), \ldots, (n-1, (+RR, [t_{n-1}, t_{n-1}])), (n, (+RG, (t_n, t_n)))\}
\]

(see Fig. 8).

Fig. 7. A configuration of \texttt{switch} with subconfigurations.

Fig. 8. A configuration of \texttt{car} with subconfigurations.

3. Operations

3.1. The general case

Operations on configuration systems over a synchronization structure \(\Sigma\) can be introduced on the basis of the following proposition (cf. Definition 3.3 and the respective comments).

**Proposition 3.1.** For all c-systems \(P, P_0, P_1 \in \texttt{cs}(\Sigma)\), each \(K \subseteq U_\Sigma\) with all internal \(u \in U_\Sigma\) in \(K\), and each injective endomorphism \(b\) of \(\Sigma\), where an injective endomorphism is an injection \(b : U_\Sigma \rightarrow U_\Sigma\) which preserves \(\oplus_\Sigma\), internality, visibility and invisibility, we have the following c-systems over \(\Sigma\)

1. \(P_0|K\), the result of restricting \(P_0\) to \(K\), where
   
   \[p \in P_0|K\iff p \in P_0\text{ and label}(e) \in K\text{ for all }e \in p.\]
(2) $P_{ob}$, the result of relabelling $P_0$ according to $b$, where
\[ p \in P_{ob} \iff p\{(e, b(\text{label}(e))): e \in p_0\} \text{ for some } p_0 \in P_0.\]

(3) $P_0; P_1$, the result of prefixing $P_0$ to $P_1$, where
\[ p \in P_0; P_1 \iff p = \{((0, e), \text{label}(e)): e \in p_0\} \text{ for some } p_0 \in P_0 \text{ or } \]
\[ p = \{((0, e), \text{label}(e)): e \in p_0\} \cup \{((1, f), \text{label}(f)): f \in p_1\} \]
\[ \text{for a maximal } p_0 \in P_0 \text{ and some } p_1 \in P_1.\]

(4) $P_0 + P_1$, the sum of $P_0$ and $P_1$, where
\[ p \in P_0 + P_1 \iff p = \{((0, e), \text{label}(e)): e \in p_0\} \text{ for some } p_0 \in P_0 \text{ or } \]
\[ p = \{((1, f), \text{label}(f)): f \in p_1\} \text{ for some } p_1 \in P_1.\]

(5) $P_0 \parallel P_1$, the parallel composition of $P_0$ and $P_1$, where
\[ p \in P_0 \parallel P_1 \iff p \text{ consists of some } p_0 \in P_0 \text{ and } p_1 \in P_1 \text{ in the sense that } \]
\[ p = \alpha_0(p_0) \cup \alpha_1(p_1).\]

Here, $\alpha_0$ and $\alpha_1$ are defined by
\[
\alpha_0(e) = ((0, e), \text{label}(e)) \text{ for } e \in p_0 - \alpha^{-1}(p_1),
\]
\[
\alpha_1(f) = ((1, f), \text{label}(f)) \text{ for } f \in p_1 - \alpha(p_0),
\]
\[
\alpha_0(e) = \alpha_1(f) = \alpha_1(f) = (((0, e), (1, f)), \text{label}(e) \otimes_2 \text{label}(f)) \text{ for } (e, f) \in \alpha,
\]
for a one-to-one correspondence $\alpha \subseteq p_0 \times p_1$, called an association of $p_0$ with $p_1$, such that

(5.1) $\alpha$ is a set of pairs of events $e \in p_0$ and $f \in p_1$ such that $\text{label}(e)$ and $\text{label}(f)$ are complementary in the sense that $\text{label}(e) \otimes_2 \text{label}(f)$ is defined, and

(5.2) $p$ does not contain any nontrivial causal cycle, i.e. any sequence $e_0, e_1, \ldots, e_n, e_{n+1}$ with $e_{n+1} = e_0$ such that, for all $i \in \{0, 1, \ldots, n\},$

either $\alpha_0^{-1}(e_{i+1})$ follows $\alpha_0^{-1}(e_i)$ in $p_0$
or $\alpha_1^{-1}(e_{i+1})$ follows $\alpha_1^{-1}(e_i)$ in $p_1$,
and, for some $j \in \{0, 1, \ldots, n\},$

either $\alpha_0^{-1}(e_{j+1})$ follows strictly $\alpha_0^{-1}(e_j)$ in $p_0$
or $\alpha_1^{-1}(e_{j+1})$ follows strictly $\alpha_1^{-1}(e_j)$ in $p_1$. 
Towards an algebra for timed behaviours

Moreover, for \( p \in P_0 \parallel P_1 \) as in (5) and \( e, f \in p, f \) follows \( e \) in \( p \) iff there exists a causal chain from \( e \) to \( f \), i.e. a sequence

\[ e_1 = e, e_2, \ldots, e_n, e_{n+1} = f \]

such that, for each \( i \in \{1, \ldots, n\} \),

- either \( \alpha_0^{-1}(e_{i+1}) \) follows \( \alpha_0^{-1}(e_i) \) in \( p_0 \)
- or \( \alpha_1^{-1}(e_{i+1}) \) follows \( \alpha_1^{-1}(e_i) \) in \( p_1 \).

**Proof (outline).** The proofs of (1)–(4) are trivial. For (5) we proceed as follows.

Let \( P = P_0 \parallel P_1 \). Consider a nonempty \( S \subseteq P \) with an upper bound \( p \in P \) as in (5). Then \( S_0 = \{ \alpha_0^{-1}(s) : s \in S \} \) and \( S_1 = \{ \alpha_1^{-1}(s) : s \in S \} \) are nonempty bounded subsets of \( P_0 \) and \( P_1 \), resp., and \( s \in S \) iff \( s \) consists of some \( s_0 \in S_0 \) and \( s_1 \in S_1 \) with the association \( \alpha_s = \alpha_0 \cap (s_0 \times s_1) \). Moreover, \( \alpha_0^{-1}(\bigcap S) = \bigcap S_0 \) and \( \alpha_1^{-1}(\bigcap S) = \bigcap S_1 \), and the fact that \( P_0 \) and \( P_1 \) are c-systems implies \( \bigcap S \subseteq P_0 \) and \( \bigcap S \subseteq P_1 \). On the other hand, each \( p' \subseteq p \) with \( \alpha_0^{-1}(p') \in P_0 \) and \( \alpha_1^{-1}(p') \in P_1 \) belongs to \( P \). Hence \( \bigcap S \) consists of \( \bigcap S_0 \) and \( \bigcap S_1 \) with the association \( \bigcap (\alpha_s : s \in S) \) and thus \( \bigcap S \subseteq P \). Similarly \( \bigcup S \subseteq P \) for each bounded \( S \subseteq P \).

For the stated characterization of the relation of following an event by another in a configuration of a parallel composition let us consider \( p \in P \) as in (5) and \( e, f \in P \). As \( \alpha_0^{-1}(p') \in P_0 \) and \( \alpha_1^{-1}(p') \in P_1 \) for each subconfiguration \( p' \) of \( p \), the existence in \( p \) of a causal chain from \( e \) to \( f \) implies immediately \( e \in p' \) for each subconfiguration \( p' \) of \( p \) with \( f \in p' \), that is that \( f \) follows \( e \). In order to see the converse implication notice that, for each \( g \in p \), the least subconfiguration \( p' \) of \( p \) with \( g \in p' \) consists of \( p'_0 \) and \( p'_1 \), where \( p'_1 \) denotes the union over \( g' \in p \) with a causal chain to \( g \) of the least subconfigurations of \( p \), containing \( \alpha_1^{-1}(g') \), hence it contains exactly those \( g'' \in p \) which there is a causal chain from \( g'' \) to \( g \). Consequently, the relation \( f \) follows \( e \) implies that the least subconfiguration \( p' \) of \( p \) such that \( f \in p' \) must contain \( e \), and hence the existence of a causal chain from \( e \) to \( f \), as required. \( \square \)

**Example 3.2.** The behaviour of the part shown in Fig. 9 of a marked place/transition Petri net can be represented as

\[ Q = P|_1 \{ U_{2,2} - \{-l1, +l1, -l4, +l4\} \}, \]

![Fig. 9. A part of a marked place-transition Petri net.](image_url)
where
\[ P = \text{place}_1(+l_1, +l_2, -l_3, -l_4) \parallel \text{trans}(+l_4, +l_5, -l_1, -l_6). \]

In Fig. 10 we show two ways of combining the configuration in Fig. 2 of \( \text{place}_1(+l_1, +l_2, -l_3, -l_4) \) and the configuration in Fig. 3 of \( \text{trans}(+l_4, +l_5, -l_1, -l_6) \). Only in the second case the resulting configuration does not contain any event \( e \) with \( \text{label}(e) \in \{-l_1, +l_1, -l_4, +l_4\} \) and thus it belongs to \( Q \).

**Definition 3.3.** The operations
\[
\begin{align*}
P_0 &\rightarrow P_0|K, \\
P_0 &\rightarrow P_0b, \\
P_1 &\rightarrow P_0; P_1, \\
(P_0, P_1) &\rightarrow P_0 + P_1, \\
(P_0, P_1) &\rightarrow P_0 \parallel P_1,
\end{align*}
\]

where \( P_0, P_1, K, b \) are as in Proposition 3.1, are called **basic operations** on c-systems.

The assumptions about \( K \) and \( b \) in Proposition 3.1 are made in order to guarantee that the corresponding operations preserve the considered equivalences of c-systems. The prefixing is regarded as an operation with respect to the second argument only, the first one playing the role of a parameter. This is motivated by the lack of continuity with respect to the first argument in the sense to be defined. Note that prefixing of an action symbol \( u \in U_2 \) to a c-system \( P_1 \) is a particular variant of such an operation with \( u \) represented by a one-event c-system with an event with the label \( u \).

For c-systems we have a natural prefix relation.
Definition 3.4. Given two c-systems $P$ and $Q$, we say that $P$ is a prefix of $Q$, written as $P \preceq Q$, iff $P \subseteq Q$ and, for each $q \in Q$, $q \subseteq \bigcup P$ implies $q \in P$.

The following property is a simple consequence of definition.

Proposition 3.5. The relation $\preceq$ is a chain-complete partial order on $\text{cs}(\Sigma)$ with $\text{Nil}$ playing the role of least element and the supremum of each countable chain $P_0 \preceq P_1 \preceq \cdots$ being $\bigcup(P_i : i \in \omega)$, where $\omega = \{0, 1, \ldots \}$. The basic operations are continuous with respect to this order, that is, they preserve the suprema of countable chains in the respective cartesian powers of $\text{cs}(\Sigma)$.

From the known properties of complete partial orders we obtain the following result.

Proposition 3.6. Let $F : (\text{cs}(\Sigma))^{m+n} \to (\text{cs}(\Sigma))^m$ be a continuous mapping which transforms each pair $(P, Q)$ with $P \in (\text{cs}(\Sigma))^{m+n}$ and $Q \in (\text{cs}(\Sigma))^n$ into some $R = F(P, Q) \in (\text{cs}(\Sigma))^n$. Then we have:

1. the fixed-point equation $P = F(P, Q)$ has a least solution written as $\text{fix}_P F(P, Q)$,
2. the solution $\text{fix}_P F(P, Q)$ is given by $\bigcup(P_i : i \in \omega)$, where $P_0 = \text{Nil}^m$ and $P_{i+1} = F(P_i, Q)$ for $i \in \omega$.
3. the correspondence $Q \mapsto \text{fix}_P F(P, Q)$ is a continuous mapping from $(\text{cs}(\Sigma))^n$ to $(\text{cs}(\Sigma))^m$.

We call the correspondence between $F$ and $Q \mapsto \text{fix}_P F(P, Q)$ a fixed-point operator. Due to this result we can define a large variety of operations on c-systems.

Definition 3.7. The operations on c-systems which can be obtained by combining basic operations with the aid of superpositions and fixed-point operators are called definable operations.

For example, the operation $Q \mapsto \text{fix}_P (P \parallel Q)$ is definable.

From Proposition 3.6 and Definition 3.7 we obtain the following result.

Proposition 3.8. Definable operations on c-systems are continuous.

3.2. Operations on timed configuration systems

As timed c-systems are c-systems of a particular type, we may combine them with the aid of operations on c-systems. For some operations (like restrictions and summation) we obtain in this manner again timed c-systems. For some others (such as relabellings, prefixing, composition) the results are not necessarily timed c-systems in the sense of Definition 2.10 and, in order to achieve this, we have to modify the operations. The details are as follows.
Proposition 3.9. The universe $\text{tcs}(\Sigma)$ is closed under the following basic operations on $c$-systems:

$$
P \rightarrow P \mid K \quad \text{for } K \subseteq U_{T(\Sigma)} \text{, with all internal } u \in U_{T(\Sigma)} \text{ in } K$$

(restriction),

$$(P_0, P_1) \rightarrow P_0 + P_1 \quad \text{(summation)}.$$

The proof is immediate.

For relabellings we have the following obvious results.

Proposition 3.10. Let $\Sigma$ be a synchronization structure, $b$ an injective endomorphism of $\Sigma$, and $\Delta$ a nonnegative real number. Then the injection $\text{rs}(b, \Delta): U_{T(\Sigma)} \rightarrow U_{T(\Sigma)}$, where

$$\text{rs}(b, \Delta)(u, [x, y]) = (b(a), [x + \Delta, y + \Delta])$$

for $u = (a, x, y) \in U_{T(\Sigma)}$,

is an injective endomorphism of the synchronization structure $T(\Sigma)$ of timed actions. We call it a rename-and-shift endomorphism of $T(\Sigma)$.

Proposition 3.11. The universe $\text{tcs}(\Sigma)$ is closed under rename-and-shift relabellings (i.e., the relabellings corresponding to rename-and-shift endomorphisms of $T(\Sigma)$).

Prefixing for timed $c$-systems and the parallel composition differ slightly from those for arbitrary $c$-systems, though the main idea remains the same. The respective concepts can be obtained easily as follows.

Proposition 3.12. For all timed $c$-systems $P_0, P_1 \in \text{tcs}(\Sigma)$ we have the following timed $c$-systems belonging to $\text{tcs}(\Sigma)$:

1. $P_0 \cdot P_1$, the result of prefixing the timed $c$-system $P_0$ to the timed $c$-system $P_1$, where

$$p \in P_0 \cdot P_1 \iff p \in \{(0, e), \text{label}(e)): e \in p_0\} \text{ for some } p_0 \in P_0 \text{ or}$$

$$p \in \{(0, e), \text{label}(e)): e \in p_0\}$$

$$\cup \{(1, f), (\text{as}(u), [\text{entime}(u) + \Delta, \text{cptime}(u) + \Delta])):$$

$$f \in P_1, u = \text{label}(f)\}$$

for a maximal $p_0 \in P_0$, some $p_1 \in P_1$, and $\Delta = \max(\text{time}(e): e \in p_0)$,

2. $P_0 \parallel P_1$, the parallel composition of timed $c$-systems $P_0$ and $P_1$, where

$$p \in P_0 \parallel P_1 \iff p \in P_0 \parallel P_1 \text{ and } p \text{ is eager in the sense of (2) of Definition 2.10.}$$
The proof of (1) is straightforward. The proof of (2) can be carried out easily by exploiting the properties of the parallel composition of arbitrary c-systems, stated in Proposition 3.1.

The parallel composition of timed c-systems corresponds to the eager composition in [3] and it has similar properties.

**Example 3.13.** The timed behaviour of the part shown in Fig. 9 of a marked place/transition Petri net with a transition whose execution takes two units of time can be represented as

\[ S = R \upharpoonright \{ u \in U_{T,2} : \text{as}(u) \notin \{-l1, +l1, -l4, +l4\} \}, \]

where

\[ R = t\text{-place}(_{+l1, +l2, -l3, -l4}) \upharpoonright t\text{-trans}(_{+l4, +l5, -l1, -l6}). \]

In Fig. 11 we show a way of obtaining a configuration of \( S \) by combining the configuration in Fig. 4 of \( t\text{-place}(_{+l1, +l2, -l3, -l4}) \) and the configuration in Fig. 5 of \( t\text{-trans}(_{+l4, +l5, -l1, -l6}) \). The eagerness of the configuration which can be obtained in this way follows from the fact that in each coincidence class it contains an occurrence of a noninternal timed action.

**Example 3.14.** The timed behaviour of a system as in Example 2.13 consisting of a traffic light with the initial state \( R \), a switch, and two cars, can be represented by the timed c-system

\[ W = (t\text{light}(R) \upharpoonright \text{switch} \upharpoonright \text{car} \upharpoonright \text{car}) \]

\[ \{ u \in U_{T,2} : \text{as}(u) \notin \{-WR, +WR, -WG, +WG\} \}. \]

A configuration of such a timed c-system can be obtained by combining configurations of component timed c-systems as shown in Fig. 12 (cf. Figs. 6–8).
Finally, we come to the following concept of basic operations on timed c-systems.

**Definition 3.15.** The operations

\[ P_0 \mapsto P_0 \mid K, \quad P_0 \mapsto P_0 \text{rs}(b, \Delta), \quad P_1 \mapsto P_0 \cdot P_1, \]

\[ (P_0, P_1) \mapsto P_0 + P_1, \quad (P_0, P_1) \mapsto P_0 \parallel P_1, \]

where \( P_0, P_1, K, b, \Delta \) are as in Propositions 3.9, 3.10, and 3.12, are called basic operations on timed c-systems.

As the suprema of countable chains of timed c-systems are also timed c-systems, we obtain easily the following results and concepts.

**Proposition 3.16.** The basic operations on timed c-systems are continuous.

**Proposition 3.17.** For each continuous mapping \( F : (\text{tcs}(\Sigma))^m \times n \to (\text{tcs}(\Sigma))^m \), the operation \( Q \mapsto \text{fix}_P F(P, Q) \) is a continuous operation on timed c-systems.

**Definition 3.18.** The operations on timed c-systems which can be obtained by combining basic operations on timed c-systems with the aid of superpositions and fixed-point operators are called definable operations on timed c-systems.
Proposition 3.19. Definable operations on timed c-systems are continuous.

4. Equivalence

4.1. Arbitrary configuration systems

Equivalences of c-systems are essentially as for labelled event structures. In particular, for c-systems we have analogues of such important equivalences of labelled event structures as the pomset bisimulation equivalence, the weakly history preserving equivalence, and the history preserving equivalence (cf. [1] for the concepts). In order to be able to make use of such equivalences in an algebraic style we have to prove that they are congruences for definable operations on c-systems. An idea of the proof is presented in details for a common refinement of pomset bisimulation equivalence and weakly history preserving equivalence. The idea applies to the pomset bisimulation equivalence, to the weakly history preserving equivalence, and to the history preserving equivalence, as well.

The concept of equivalence of c-systems is defined for c-systems over a synchronization structure. It is determined by the corresponding concept of invisibility. The latter can be chosen arbitrarily provided that only internal actions are declared as invisible. Due to the freedom in the choice of invisibility we are able to define equivalences of c-systems depending on the problems under consideration.

The equivalence of c-systems over a synchronization structure is defined and studied with the aid of suitable morphisms, called simulations. These morphisms relate configurations of c-systems in a manner which reflects the identity of what is visible in suitable parts of related configurations and their subconfigurations.

We start with several simple facts and notions.

Proposition 4.1. Given a synchronization structure $\Sigma$, for each c-system $P \in \text{cs}(\Sigma)$ which has a greatest configuration, we have a unique c-system image of $P$, called the image of $P$, where

$$q \in \text{image}_\Sigma(P) \quad \text{iff} \quad q = \{e \in p : \text{label}(e) \notin I_\Sigma\} \quad \text{for some} \quad p \in P.$$

Proposition 4.2. For each c-system $P$ and arbitrary configurations $p, q \in P$ such that $p \subseteq q$ the set

$$\downarrow q - p = \{r : r = p' - p \text{ for some } p' \in P \text{ with } p \subseteq p' \subseteq q\}$$

is a c-system with a greatest configuration, namely $q - p$.

Definition 4.3. Given a synchronization structure $\Sigma$ and a c-system $P \in \text{cs}(\Sigma)$, an increment of $P$ is a triple $p \Rightarrow_A q$, where $p$ and $q$ are configurations of $P$ such that $p \subseteq q$ and $A = \text{image}_\Sigma(\downarrow q - p)$. 
Definition 4.4. An isomorphism from a c-system $P \in \text{cs}(\Sigma)$ to a c-system $Q \in \text{cs}(\Sigma)$ is a bijection $b : \bigcup P \rightarrow \bigcup Q$ such that, for all $e, f, p, q$,

- $f = b(e)$ implies $\text{label}(f) = \text{label}(e)$,
- $p \in P$ implies $b(p) \in Q$, and
- $q \in Q$ implies $b^{-1}(q) \in P$.

Definition 4.5. By a simulation of a c-system $P \in \text{cs}(\Sigma)$ in a c-system $Q \in \text{cs}(\Sigma)$ we mean a triple $r : P \rightarrow Q$, written also as $P \xrightarrow{r} Q$, where:

1. $r \subseteq P \times Q$,
2. $(\emptyset, \emptyset) \in r$,
3. for all $(p, q) \in r$, the c-systems $\text{image}_\Sigma(\downarrow p \emptyset)$ and $\text{image}_\Sigma(\downarrow q \emptyset)$ are isomorphic,
4. for each $(p, q) \in r$ and each increment $p \Rightarrow A p'$ of $P$ there exists an increment $q \Rightarrow B q'$ of $Q$ such that $(p', q') \in r$ and the c-systems $A$ and $B$ are isomorphic.

If $(\emptyset, q) \in r$ only for $q = \emptyset$, then we call $r : P \rightarrow Q$ a rooted simulation. If $r^{op} : Q \rightarrow P$, where $r^{op} = \{(q, p) : (p, q) \in r\}$, is also a simulation then we call $r : P \rightarrow Q$ a bisimulation.

Example 4.6. $\text{id}_P : P \rightarrow Q$ with $P \preceq Q$ and $\text{id}_P$ denoting the identity in $P$, written also as $P \overset{\text{id}_P} \rightarrow Q$, is a rooted simulation. Similarly, $B_P : P \rightarrow P + P$ with

$$(p, q) \in B_P \text{ iff } q = \{((0, e), \text{label}(e)) : e \in p\}$$

or $q = \{((1, e), \text{label}(e)) : e \in p\}$

is a rooted bisimulation.

From the definition we have the following properties and concept.

Proposition 4.7. If $r : P \rightarrow Q$ and $s : Q \rightarrow R$ are simulations (resp.: rooted simulations, bisimulations) then $r \circ s : P \rightarrow R$ with

$$r \circ s = \{(p, r) : (p, q) \in r \text{ and } (q, r) \in s \text{ for some } q \in Q\}$$

is also a simulation (resp.: a rooted simulation, a bisimulation).

Proposition 4.8. The binary relation defined by

$$P \approx Q \text{ iff there exists a rooted bisimulation } r : P \rightarrow Q$$

is an equivalence. We call it the behavioural equivalence of c-systems. If $P \approx Q$ then we say that $P$ and $Q$ are behaviourally equivalent (or simply equivalent).

For example, for arbitrary c-systems we have

- $P + \text{Nil} \approx P$,
- $P + P \approx P$,
- $P + Q \approx Q + P$,
- $(P + Q) + R \approx P + (Q + R)$,
- $P \parallel Q \approx Q \parallel P$,
- $(P \parallel Q) \parallel R \approx P \parallel (Q \parallel R)$.

From Proposition 4.7 we conclude the following fact which suggests a way of studying the behavioural equivalence of c-systems.
Proposition 4.9. Given a synchronization structure $\Sigma$, the c-systems over $\Sigma$ and their simulations constitute a category which we denote by $\text{CS}(\Sigma)$. For each cardinal $m$ we have the cartesian power $(\text{CS}(\Sigma))^m$ of this category.

The categories thus obtained have a property which is important for our studies.

Proposition 4.10. The category $\text{CS}(\Sigma)$ has colimits of countable chains. The colimit of each chain $P_0 \ll P_1 \ll \cdots$ coincides with the supremum $P = \bigcup (P_i : i \in \omega)$ and similarly for the cartesian powers of $\text{CS}(\Sigma)$. Moreover, $r = \bigcup (r_i : i \in \omega)$ for each commutative diagram as in Fig. 13 with the unique $r : P \to Q$ resulting from the universal property of colimits.

Proof (outline). It suffices to notice that the commutativity of the diagram in Fig. 13 means that each $r_i$ with $i < j$ is the restriction of $r_j$ to $P_i$ and that $r = \bigcup (r_i : i \in \omega)$ is the unique relation such that the diagram in Fig. 13 commutes.  \[ \square \]

For the categories of c-systems and their simulations and for cartesian powers of such categories we consider functors with some particular properties.

Definition 4.11. Let $F : (\text{CS}(\Sigma))^m \to (\text{CS}(\Sigma))^m$ be a functor. We say that $F$ is continuous iff it preserves colimits. We say that $F$ preserves the prefix order iff, for all $P$ and $Q$, $P \ll Q$ implies $F(P) \ll F(Q)$ and the coincidence of $F(P) \overset{\mathcal{F}(\ll)}{\to} F(Q)$ with $F(P) \overset{\mathcal{F}(\ll)}{\to} F(Q)$. Finally, we say that $F$ preserves rooted bisimulations iff the simulation $F(P) \overset{\mathcal{F}(\ll)}{\to} F(Q)$ is a rooted bisimulation for each rooted bisimulation $r : P \to Q$.

The following property of definable operations on c-systems is crucial for our purposes.

Proposition 4.12. Each definable operation on c-systems can be extended in a canonical way to a continuous functor which preserves the prefix order and rooted bisimulations.

Proof (outline). For the basic operations the proof is straightforward. For example, for $r_0 : P_0 \to Q_0$ and $r_1 : P_1 \to Q_1$, we define $r_0 \parallel r_1 : P_0 \parallel P_1 \to Q_0 \parallel Q_1$ by

- $(p, q) \in r_0 \parallel r_1$ iff $p$ consists of $p_0$ and $p_1$, $q$ consists of $q_0$ and $q_1$,
- $(p_0, q_0) \in r_0$, $(p_1, q_1) \in r_1$. 

Fig. 13.
In this way we obtain a functor whose continuity and other required properties follow easily from Propositions 3.16 and 4.10.

In order to extend the proof on all definable operations it suffices to consider a continuous functor $F : (CS(\Sigma))^m \rightarrow (CS(\Sigma))^n$ which preserves the prefix order and rooted bisimulations and to prove that the operation $f : Q^* \rightarrow \text{fix}_P F(P, Q)$ extends to a continuous functor which preserves the prefix order and rooted bisimulations.

Suppose that $r : Q \rightarrow Q'$ is a simulation and consider the least solutions $f(Q)$ and $f(Q')$ of the respective fixed-point equations $P = F(P, Q)$ and $P = F(P, Q')$. As $F$ is continuous, we obtain the commutative diagram in Fig. 14 with a unique simulation $s : f(Q) \rightarrow f(Q')$. From the uniqueness of $s$ and Proposition 4.10 it follows that the correspondence $r \mapsto s$ defines a functor $f : r \mapsto s$, where $f(r) = \bigcup \{ r_i : i \in \omega \}$ with $r_0 = \emptyset^n$ and $r_{i+1} = F(r_i, r)$ for $i \in \omega$, and that this functor is continuous. It is also easy to see that this functor preserves the prefix order and rooted bisimulations.

From Proposition 4.12 and the definition of the equivalence of c-systems we obtain the desired result.

![Fig. 14.](image)

**Proposition 4.13.** The behavioural equivalence of c-systems is a congruence for all definable operations on c-systems.

The concept of equivalence of c-systems extends in a natural way to a concept of equivalence of operations on c-systems.

**Definition 4.14.** Functors $F : (CS(\Sigma))^m \rightarrow (CS(\Sigma))^n$ and $G : (CS(\Sigma))^m \rightarrow (CS(\Sigma))^n$ are said to be equivalent, written as $F \equiv G$, iff there exists a natural transformation $r : F \rightarrow G$ which consists of rooted bisimulations, i.e. a family

$$r = (r(P) : F(P) \rightarrow G(P) : P \in (CS(\Sigma))^n)$$

of rooted bisimulations such that, for each simulation $s : P \rightarrow Q$, a diagram as in Fig. 15 commutes. Two definable operations on c-systems are said to be equivalent iff their canonical extensions to functors which preserve the prefix order and rooted bisimulations are equivalent.
Towards an algebra for timed behaviours

For example, the operations \( P \rightarrow P \) and \( P \rightarrow P + P \) are equivalent with the equivalence given by

\[
(B_P : P \rightarrow P : P \in \text{cs}(\Sigma))
\]

with \( B_P \) as in Example 4.6. Similarly, the following operations are equivalent:

\[
\begin{align*}
(P, Q) \rightarrow P + Q \quad &\text{and} \quad (P, Q) \rightarrow Q + P \\
(P, Q) \rightarrow P \parallel Q \quad &\text{and} \quad (P, Q) \rightarrow Q \parallel P \\
(P, Q, R) \rightarrow (P + Q) + R \quad &\text{and} \quad (P, Q, R) \rightarrow P + (Q + R) \\
(P, Q, R) \rightarrow (P \parallel Q) \parallel R \quad &\text{and} \quad (P, Q, R) \rightarrow P \parallel (Q \parallel R).
\end{align*}
\]

For the equivalence of definable operations on c-systems we have the following result.

**Proposition 4.15.** If two definable operations on c-systems are constructed in the same manner from equivalent definable operations then they are equivalent.

**Proof (outline).** The only nontrivial part of the proof is that about operations defined by fixed-point equations. Thus it suffices to consider two equivalent continuous functors which preserve the prefix order and rooted bisimulations, say \( F : (\text{cs}(\Sigma))^n \rightarrow (\text{cs}(\Sigma))^m \) and \( G : (\text{cs}(\Sigma))^n \rightarrow (\text{cs}(\Sigma))^m \), with the equivalence given by a family \( r \) of rooted bisimulations \( r(P, Q) \) and find a suitable family of \( s(Q) : f(Q) \rightarrow g(Q) \) for \( f(Q) = \text{fix}_P F(P, Q) \) and \( f(Q) = \text{fix}_P G(P, Q) \). To this end, we consider the diagram in Fig. 16 which is commutative due to the continuity and the other properties of \( F \) and \( G \) and due to the fact that \( r \) is a natural transformation from \( F \) to \( G \). By Proposition 4.10 we obtain a unique rooted bisimulation \( s(Q) : f(Q) \rightarrow g(Q) \) which completes this diagram to a commutative one. From the uniqueness of \( s(Q) \) we obtain that the family

\[
s = (s(Q) : f(Q) \rightarrow g(Q) : Q \in (\text{cs}(\Sigma))^n)
\]

is a natural transformation as required. \( \square \)
4.2. Timed c-systems

An equivalence of timed c-systems can be introduced by slightly modifying the concept of equivalence of arbitrary c-systems. As we want this equivalence to be a congruence for definable operations on timed c-systems, we have to take into account the fact that the parallel composition of timed c-systems depends on the existence of critical or noninternal events in each coincidence class of the component timed c-systems. Consequently, in order to have the parallel composition of timed c-systems depending only on the equivalence classes of these c-systems, we have to modify the concept of visible image of a c-system with a greatest configuration by adding information about the existence in coincidence classes of invisible critical events. As only the existence of such events is essential, we represent it in each case by a single event with an extra action symbol. This leads us to replacing the notion of image of a c-system with a greatest configuration by a notion suitable for timed c-systems. After such a replacement all the remaining concepts concerning the equivalence apply to timed c-systems and all the results and proofs about the equivalence of arbitrary c-systems and the equivalence of definable operations on arbitrary c-systems remain valid. The respective formulations are as follows.

Proposition 4.16. Given a synchronization structure \( \Sigma \), for each timed c-system \( P \in \text{tcs}(\Sigma) \) which has a greatest configuration we have a unique timed c-system \( \text{t-image}_\Sigma(P) \), called the timed image of \( P \), where \( q \in \text{t-image}(P) \) iff there exists \( p \in P \) such that

\[
q = \{((0, e), \text{label}(e)) : e \in p : \text{as(label}(e)) \notin I_\Sigma \}
\]

\[
\cup \{((1, E), (\text{invisible}, \text{wait(label}(f)))) : \text{Ex}(p, f, E)\}
\]
with

- $\text{Ex}(p, f, E)$ iff $f \in p$, and $f$ is critical, and $\text{as}(\text{label}(f)) \in I_\Sigma$, and in the coincidence class of $f$ there are $e$ with $\text{as}(\text{label}(e)) \notin I_\Sigma$ none of which is critical, and $E$ is the set of $e$ such that $e$ is coincident with $f$ and $e$ is critical and $\text{as}(\text{label}(e)) \in I_\Sigma$,

- $\text{invisible}$ being a special symbol not in $U_\Sigma$.

The proof of this proposition is straightforward if we think of each event of the form

$$((1, E), (\text{invisible, wait}(\text{label}(f))))$$

as of a construct representing the existence of critical invisible events in a coincidence class of events such that all visible events in this class are not critical.

**Example 4.17.** By combining configurations as shown in Fig. 11 (cf. Example 3.13) we obtain a timed c-system $P \in \text{tcs}(\Sigma 2)$ with a greatest configuration as shown in Fig. 17. In the case of invisible $l_1$ and $l_4$ (that is $l_1$ and $l_4$ in $I_\Sigma$) we obtain the \text{t-image} of $P$ which we show in Fig. 18.

With the idea of a timed image we come to a concept of increment and a concept of simulation for timed c-systems.

**Definition 4.18.** Given a synchronization structure $\Sigma$ and a timed c-system $P \in \text{tcs}(\Sigma)$, a **timed increment** (or briefly a **t-increment**) of $P$ is a triple $p \leadsto A q$, where $p$ and $q$ are configurations of $P$ such that $p \subseteq q$ and $A = \text{t-image}_\Sigma(\downarrow q-p)$ (we recall that $\downarrow q-p$ denotes the set $\{r: r = p'-p$ for some $p' \in P$ with $p \subseteq p' \subseteq q\}$). □
Definition 4.19. By a simulation of a timed c-system \( P \in \text{tcs}(\Sigma) \) in a timed c-system \( Q \in \text{tcs}(\Sigma) \) we mean a triple \( r: P \rightarrow Q \), written also as \( P \overset{r}{\rightarrow} Q \), where:

1. \( r \subseteq P \times Q \),
2. \( (0,0) \in r \),
3. for all \( (p,q) \in r \), the c-systems \( \text{t-image}_p(\downarrow p-0) \) and \( \text{t-image}_q(\downarrow q-0) \) are isomorphic,
4. for each \( (p,q) \in r \) and each timed increment \( p \rightarrow_A p' \) of \( P \) there exists a timed increment \( q \rightarrow_B q' \) of \( Q \) such that \( (p',q') \in r \) and the c-systems \( A \) and \( B \) are isomorphic.

If \( (0,q) \in r \) only for \( q = 0 \) then we call \( r: P \rightarrow Q \) a rooted simulation. If \( r^{op}: Q \rightarrow P \), where \( r^{op} = \{(q,p): (p,q) \in r\} \), is also a simulation when we call \( r: P \rightarrow Q \) a bisimulation.

Example 4.20. Let \( P \) denote the timed behaviour \( t\text{-trans}((+l4, +l5, -l1, -l6)) \) of a transition \( t \) of duration 2 of a timed place/transition Petri net. According to Example 2.12, this behaviour consists of independent components like the one shown in Fig. 19. Let \( Q \) be the timed behaviour of a part \( N \) as in Fig. 20 of a marked timed

Fig. 19. A transition \( t \) (left). A component of the behaviour of \( t \) (right).

Fig. 20. A part of \( N \) of a net (left). A component of the behaviour of \( N \) (right).
place/transition Petri net with each transition taking one unit of time. This timed behaviour can be obtained by combining the timed behaviours of the places and transitions of \( N \) as described in Example 3.13. It consists of independent components as shown in Fig. 20. Assuming that \( k_1, k_2, k_3, k_4 \) are invisible and relating the components of \( P \) and \( Q \) as shown in Fig. 21 we obtain a rooted bisimulation \( r: P \to Q \).

The results about simulations of arbitrary c-systems and the related notions apply also to timed c-systems. In particular, in a manner as for arbitrary c-systems we obtain what follows.

**Proposition 4.21.** If \( r: P \to Q \) and \( s: Q \to R \) are simulations of timed c-systems then \( r \circ s: P \to R \) is also a simulation of timed c-systems.

**Proposition 4.22.** The following binary relation between timed c-systems is an equivalence:

\[ P \approx Q \text{ iff there exists a rooted bisimulation } r: P \to Q. \]

We call it the behavioural equivalence (or simply equivalence) of timed c-systems.

**Proposition 4.23.** Given a synchronization structure \( \Sigma \), the timed c-systems for \( \Sigma \) and their simulations constitute a category \( \text{TCS}(\Sigma) \). For each cardinal \( m \) we have the
cartesian power \((TCS(\Sigma))^m\) of this category. The category \(TCS(\Sigma)\) has colimits of countable chains and these colimits coincide with the suprema. Similarly for the cartesian powers of \(TCS(\Sigma)\).

**Definition 4.24.** The continuity of a functor \(F : (TCS(\Sigma))^m \rightarrow (TCS(\Sigma))^n\), preservation of the prefix order, and preservation of rooted bisimulations, are defined as for functors between cartesian powers of categories of arbitrary \(c\)-systems and their simulations (cf. Definition 4.11). Similarly for the equivalence of functors between the cartesian powers of \(TCS(\Sigma)\) (cf. Definition 4.14).

**Proposition 4.25.** Each definable operation on timed \(c\)-systems can be extended in a canonical way to a continuous functor which preserves the prefix order and rooted bisimulations.

**Proposition 4.26.** The behavioural equivalence of timed \(c\)-systems is a congruence for all definable operations on timed \(c\)-systems.

**Proposition 4.27.** If two definable operations on timed \(c\)-systems are constructed in the same manner from equivalent definable operations on timed \(c\)-systems (that is, from operations whose canonical extensions are equivalent functors) then they are equivalent (that is, their canonical extensions are equivalent functors).

5. Final remarks

The general tendency in modelling complex behaviours is to specify their branching structure and causal order.

The information about branching and causality is a minimum one needs for dealing with dynamical properties of behaviours. However, not all properties can be expressed with such information. On one hand, one may need to describe behaviours like those of real time systems, communication protocols, etc., where keeping time into account is substantial. On the other hand, there are properties which cannot be described without some information about the lapse of time, like a fairness which reflects assumptions about relative speeds of system components, or like inevitability of events known to necessarily occur in a certain period of time. In order to cover cases like these we need definition tools powerful enough to deal with time.

A suitable descriptive power can be achieved by considering timed behaviours and modelling them with the aid of timed configuration systems. In this case we combine the structure of branching and causality with that of time. The structure of branching and the causal order are given by configurations and how the configurations contain each other. The structure of time is given by timed actions.
Towards an algebra for timed behaviours

Combining the structure of causality with that of time reveals that configurations do not necessarily correspond to real states of development of the represented behaviour. For example, the one-event configuration consisting of the occurrence of \((b, 0, 2]\) does not correspond to any real state of the behaviour represented by the timed c-system in Fig. 22.

![Fig. 22. A timed c-system.](image)

The configurations corresponding to real states of a behaviour can be distinguished among all configurations by means of the information on completion times of events. Consequently, we can formally define the possible runs of the represented behaviour and how they develop in time. This may help in reasoning about dynamical properties like safety, liveness, inevitability of some states, etc.

It is important to realize that considering time does not necessarily mean requiring more information than one usually has at his disposal or may assume. For instance, in order to have a fairness, it may be sufficient to know only that delays between certain events have a positive lower bound and a finite upper bound.

References


