Algebra of Timed Frames

J.A. Bergstra[1,2] W.J. Fokkink[2] C.A. Middelburg[2,3]

[1] *Programming Research Group, University of Amsterdam*
P.O. Box 41882, 1009 DB Amsterdam, The Netherlands

[2] *Department of Philosophy, Utrecht University*
P.O. Box 80126, 3508 TC Utrecht, The Netherlands

[3] *Department of Network & Service Control, KPN Research*
P.O. Box 421, 2260 AK Leidschendam, The Netherlands

E-mail: `janb@fwi.uva.nl` - `fokkink@phil.ruu.nl` - `keesm@phil.ruu.nl`

Abstract

Timed frames are introduced as objects that can form a basis of a model theory for discrete time process algebra. An algebraic setting for timed frames is proposed and results concerning its connection with discrete time process algebra are given. The presented theory of timed frames captures the basic algebraic properties of timed transition systems for the relative time case. Further structure on timed frames is provided by adding signal inserted states and conditional transitions, thus giving a semantic basis for discrete time process algebra with propositional signals. Time conditions are introduced to cover the absolute time case.

Keywords & Phrases: discrete time, frame algebra, process algebra, conditional transitions, signal inserted states, timed frames, σ-bisimulation.

1994 CR Categories: F.1.1, F.3.1, F.3.2, D.1.3, D.3.1.

Contents

1 Introduction

Frame algebra, introduced in [10], is a simple, general algebraic setting for the objects of the kind that generally underlies models for theories concerning process behaviour. Frames are built from states and action labelled transitions. Equipped with a root marker and optionally with termination markers, they make up the objects known as process graphs or transition systems. There is a well-developed tradition of modal formalisms for the description of transition systems (for an overview of these formalisms, see e.g. [16]). Process algebra studies transition systems at a more abstract level: transition systems modulo an appropriate "process equivalence". In [11], the connection between frames and processes is studied in the setting of ACP [5, 8] and moreover signal insertion is added.

In this paper, we extend frame algebra to an algebra of timed frames. We study the connection between timed frames and discrete time processes in the setting of the discrete time extensions of ACP described in [3]. The primary motivation of this study is logic based: a better understanding of the connection facilitates a systematic approach to devise a suitable logic for reasoning about discrete time processes, applying techniques from modal logic (relevant examples of the application of these techniques can be found in [6, 7]). We also investigate the possibility to impose further structure on timed frames that provides a semantic basis for a discrete time extension of process algebra with propositional signals [2]. The choice of process algebra with propositional signals is motivated by the fact that it is the only extension of time free process algebra exhibiting the interplay between the performance of actions and the consequent visible state changes – which is important if we aim at programming.

Timed frames contain two kinds of transitions: action steps – representing the execution of actions – and time steps – representing the passage of (discrete) time. Time determinism, the property that passage of time by itself can not determine a choice, is built into the operational semantics of the versions of discrete time process algebra presented in [3] by preventing that states with more than one outgoing time step occur. An alternative is used here for timed frames: a special kind of bisimulation, called σ-bisimulation, sees to that passage of time by itself can not determine a choice. One of the results concerning the extraction of discrete time processes from timed frames is that σ-bisimilar frames yield bisimilar processes. By this result, a logic for reasoning about timed frames can be considered to be a logic for reasoning about discrete time processes in relative time if its formulae allow to distinguish frames up to σ-bisimulation (cf. [12]). The result further implies that the operational semantics given in [3] could be simplified considerably at the expense of a more complicated notion of bisimulation, viz. σ-bisimulation.

We extend timed frames with signal insertion. That is, we add a signal insertion operation which assigns a propositional formula to the states contained in a timed frame. The propositional formula assigned to a state is considered to hold in that state. Thus a semantic basis for discrete time extensions of modal logics for reasoning about actions and state changes is given. The above-mentioned result concerning process extraction goes through for various extensions of timed frames, including this one. This is very relevant to ongoing work on devising a suitable logic for reasoning about processes described in φSDL, which is an interesting subset of SDL [17] for which a process algebra semantics is given in [9].

We complement signal inserted states with conditional transitions. The conditional transitions are labelled with an action (or σ in the case of time steps) and a propositional formula. The intuition behind these conditional transitions is that in a state only its outgoing conditional transitions can be performed for which the propositional formula used as condition holds in that state. By the addition of signal inserted states and conditional transitions, a semantic basis for discrete time extensions of process algebra with propositional signals [2] is given as well.

The timed frames made mention of before are only adequate to represent discrete time process behaviour in the relative time case. In order to cover the absolute time case as well, we extend timed frames with another kind of conditional transitions where the truth of the condition is time dependent instead of state dependent. This extension can be regarded as a generalization of time stamping of transitions. Both kinds of conditional transitions can easily be integrated.

The structure of this paper is as follows. First of all, we give an overview of time free frame algebra and a brief summary of the ingredients of process algebra used in this paper (Section 2). Next, we illustrate the use of timed frames by means of an example (Section 3.1). Then, we elaborate timed frames in detail and study their connection with discrete time processes (Section 3.2). After that, we consider the extension of timed frames with signal inserted states (Section 3.3) and we introduce conditional transitions as a complement of signal inserted states (Section 4.1). Finally, we describe the extension of timed frames needed to cover, in addition to the relative time case, the absolute time case (Section 4.2).

2 Preliminaries

This section contains a survey of time free frame algebra, including its extension with signal insertion. We refer to [10] and [11] for further details. A brief summary of the ingredients of process algebra used in later sections is given as well. We will suppose that the reader is familiar with them. Appropriate references to the literature are included.

2.1 Simple frames

Frames are built from states and transitions between states. The states are obtained by an embedding of naturals in states, and a pairing function on states. We consider transitions with a label from a finite set A of *actions*.

The signature of *(simple) frames* is as follows:

Sorts:

$\mathbb{N}$	naturals;
$\mathbb{S}$	*states*;
$\mathbb{F}$	*frames*;

Constants & Functions:

$0 \; : \mathbb{N}$	zero;
$S \; : \mathbb{N} \to \mathbb{N}$	successor;
$\imath_{\mathbb{N}} \; : \mathbb{N} \to \mathbb{S}$	embedding of naturals in states;
$\rangle\!\!-\!\!\langle : \mathbb{S}^2 \to \mathbb{S}$	pairing of states;
$\emptyset \; : \mathbb{F}$	*empty frame*;
$\imath_{\mathbb{S}} \; : \mathbb{S} \to \mathbb{F}$	*embedding of states* in frames;
$\xrightarrow{a} : \mathbb{S}^2 \to \mathbb{F}$	*transition construction* (one for each $a \in A$);
$\oplus \; : \mathbb{F}^2 \to \mathbb{F}$	*frame union.*

The signature introduced above is graphically presented in Figure 1. Given the

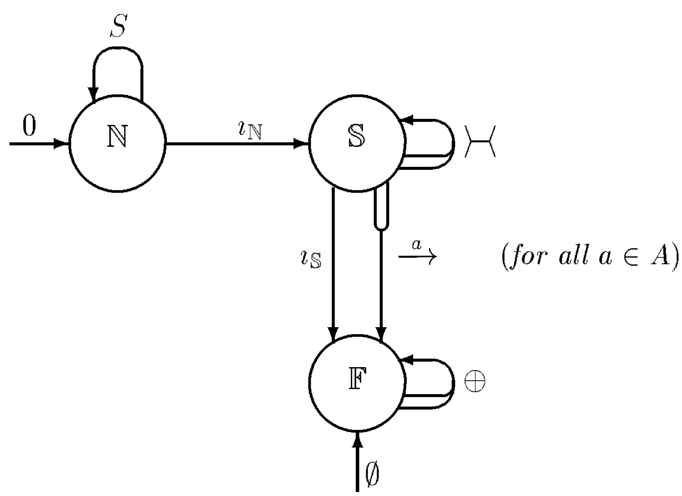

Figure 1: Signature of frames.

signature, (closed) terms are constructed in the usual way. We shall use the meta-variables n and m to stand for arbitrary terms of sort $\mathbb{N}$, the meta-variables s, s' and s'' to stand for arbitrary terms of sort $\mathbb{S}$, and the meta-variables X, Y and Z to stand for arbitrary terms of sort $\mathbb{F}$. We write n instead of $\imath_{\mathbb{N}}(n)$ or $\imath_{\mathbb{S}}(\imath_{\mathbb{N}}(n))$ as well as s instead of $\imath_{\mathbb{S}}(s)$ when this causes no ambiguity. Terms of the forms $\imath_{\mathbb{S}}(s)$ and $s \xrightarrow{a} s'$ denote *atomic* frames, i.e. frames that contain a single state or transition. The constant $\emptyset$ denotes the frame that contains neither states nor transitions. The operator $\oplus$ on frames gives the union of the states and transitions of its arguments. The pairing function $\rangle\!\!-\!\!\langle$ is a simple means to define "fresh" states.[1] The axioms for frames are given in Table 1. These axioms

[1] In [10], the pairing function is used to define a *frame product* function.

(FA1)	$X \oplus Y$	$=$	$Y \oplus X$
(FA2)	$X \oplus (Y \oplus Z)$	$=$	$(X \oplus Y) \oplus Z$
(FA3)	$X \oplus X$	$=$	X
(FA4)	$X \oplus \emptyset$	$=$	X
(FA5)	$s \oplus (s \xrightarrow{a} s')$	$=$	$s \xrightarrow{a} s'$
(FA6)	$s' \oplus (s \xrightarrow{a} s')$	$=$	$s \xrightarrow{a} s'$

Table 1: Axioms for frames.

characterize frames as objects consisting of a finite set of states and a finite set of transitions. In addition, frames are identified if they are the same after addition of the states occurring in the transitions to the set of states (axioms (FA5) and (FA6)). The axioms do not identify frames according to some notion of equivalence that is used to obtain an adequate level of abstraction for processes – such as bisimulation – because frames are intended to provide a lower level of abstraction.

We define *iterated* frame union by

$$\bigoplus_{i=n}^{k} X_i = \begin{cases} \emptyset & \text{if } k < n, \\ X_n \oplus \bigoplus_{i=n+1}^{k} X_i & \text{otherwise.} \end{cases}$$

Every frame has a finite number of states and transitions, and can be denoted by a term of the form $\bigoplus_{i=1}^{m} X_i$, where the X_i are atomic.

In [10], frame polynomials are introduced to deal with the countably infinite case as well. For completeness sake, this section also informs briefly on frame polynomials. However, because this paper focusses on timed frames corresponding to *regular* discrete time processes, only frames with a finite number of states and transitions are considered in later sections.

We assume a countably infinite set V of variables $x, y, \ldots$ ranging over $\mathbb{N}$. Terms over V are constructed in the usual way. *Frame polynomials* over V are constructed according to the same formation rules and the following additional one: if F is a frame polynomial over V and $x \in V$, then $\bigoplus_x F$ is a frame polynomial. The *generalized frame union* $\bigoplus_x F$ is defined by[2]

$$\bigoplus_x F = F[0/\,x] \oplus F[1/\,x] \oplus F[2/\,x] \oplus \cdots$$

A frame polynomial F is closed if all occurrences of variables in F are bound by an application of generalized frame union. The axioms of closed frame polynomials, are the axioms given in Table 1, understanding that the range of the meta-variables is properly extended, and the additional axioms given in Table 2. The

[2] $F[n/\,x]$ stands for the result of replacing the term n for the occurrences of the variable x in F.

(FP1)	$\bigoplus_x F$	$=$	F	provided x does not occur in F
(FP2)	$\bigoplus_y F$	$=$	$\bigoplus_x F[x/y]$	provided x does not occur in F
(FP3)	$\bigoplus_x \bigoplus_y F$	$=$	$\bigoplus_y \bigoplus_x F$	
(FPA1)	$\bigoplus_x (F \oplus F')$	$=$	$\bigoplus_x F \oplus \bigoplus_x F'$	
(FPA2)	$\bigoplus_x F(x)$	$=$	$F[0/x] \oplus \bigoplus_x F[S(x)/x]$	

Table 2: Additional axioms for frame polynomials.

meta-variables x and y stand for arbitrary variables from V, and the meta-variables F and F' stand for arbitrary frame polynomials. The proviso "x does not occur in F" means that variable x does not occur (free or bound) in F.

2.2 Signal inserted frames

In simple frames, states are not labelled. In signal inserted frames, we consider states with a label from the set of propositional formulae that can be built from a set $\mathbb{P}_{at}$ of *atomic propositions*, t, f, and the connectives $\neg$ and $\rightarrow$. The propositional formula assigned to a state is considered to hold in that state. The further structure on frames provided by adding signal inserted states, gives a semantic basis for modal logics for reasoning about actions as well as state changes. This is further explored in [11].

The signature extension for *signal inserted* frames is as follows:[3]

Sorts:

$\mathbb{P}$		propositions;
$\langle \mathbb{F}, \mathbb{P} \rangle$		*signal inserted frames*;

Constants & Functions:

p	$: \mathbb{P}$	for each $p \in \mathbb{P}_{at}$;
t	$: \mathbb{P}$	true;
f	$: \mathbb{P}$	false;
$\neg$	$: \mathbb{P} \rightarrow \mathbb{P}$	negation;
$\rightarrow$	$: \mathbb{P}^2 \rightarrow \mathbb{P}$	implication;
$\curvearrowright$	$: \mathbb{P} \times \langle \mathbb{F}, \mathbb{P} \rangle \rightarrow \langle \mathbb{F}, \mathbb{P} \rangle$	*signal insertion.*

The signature of signal inserted frames is graphically presented in Figure 2. We shall use the meta-variables ϕ and ψ to stand for arbitrary terms of sort $\mathbb{P}$. As usual, we write $\phi \vee \psi$ for $\neg\phi \rightarrow \psi$, $\phi \wedge \psi$ for $\neg(\neg\phi \vee \neg\psi)$, and $\phi \leftrightarrow \psi$ for $(\phi \rightarrow \psi) \wedge (\psi \rightarrow \phi)$. In Table 3 we give a complete proof system for propositional logic. The signal insertion operation $\curvearrowright$ assigns propositional formulae to the

[3] We will not give a full signature if it can be obtained from an old signature by first renaming one sort and then adding new sort, constant and function names.

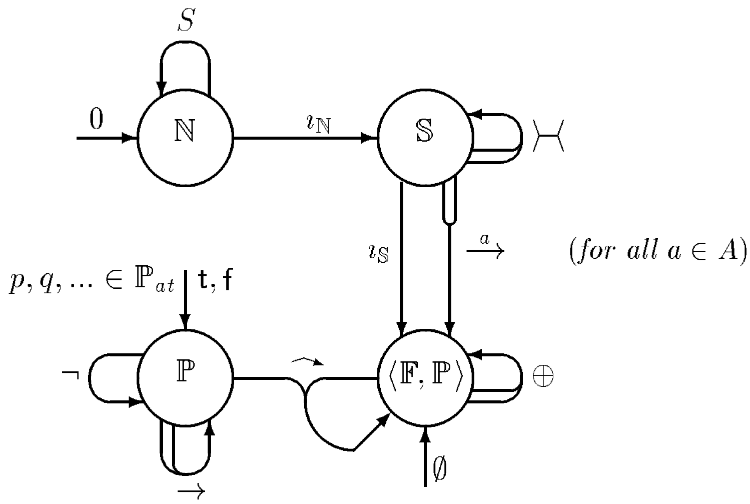

Figure 2: Signature of signal inserted frames.

states contained in frames. The axioms for signal inserted frames are those given in Table 1 (see Section 2.1) and the additional axioms for signal insertion given in Table 4. Additionally, we can use identities $\phi = \psi$ iff $\phi \leftrightarrow \psi$ is provable from the axiom schemas and the inference rule given in Table 3. The axioms (Ins4)–(Ins8) are concerned with combinations of signal insertion ($\curvearrowright$) with frame union ($\oplus$). Note that frame union still gives the union of the states of its arguments, but not always the union of the transitions: a signal inserted state $\mathsf{f} \curvearrowright s$ absorbs all incoming and outgoing transitions of s (axioms (Ins7) and (Ins8)). Note that the equation

$$s \oplus (\phi \curvearrowright s) \quad = \quad \phi \curvearrowright s$$

(reminiscent of the axioms (FA5) and (FA6)) is derivable from the axioms (Ins2)

(P1)	$\phi \rightarrow (\psi \rightarrow \phi)$
(P2)	$(\phi \rightarrow (\psi \rightarrow \xi)) \rightarrow ((\phi \rightarrow \psi) \rightarrow (\phi \rightarrow \xi))$
(P3)	$(\neg\phi \rightarrow \neg\psi) \rightarrow (\psi \rightarrow \phi)$
(P4)	$\mathsf{t} \leftrightarrow (p \rightarrow p)$
(P5)	$\mathsf{f} \leftrightarrow \neg\, \mathsf{t}$
(MP)	$\dfrac{\phi \quad \phi \rightarrow \psi}{\psi}$

Table 3: A proof system for propositional logic.

(Ins1)	$\phi \curvearrowright \emptyset$	$=$	$\emptyset$
(Ins2)	$\mathsf{t} \curvearrowright X$	$=$	X
(Ins3)	$\phi \curvearrowright (\psi \curvearrowright X)$	$=$	$(\phi \wedge \psi) \curvearrowright X$
(Ins4)	$(\phi \curvearrowright X) \oplus (\psi \curvearrowright X)$	$=$	$(\phi \wedge \psi) \curvearrowright X$
(Ins5)	$\phi \curvearrowright (X \oplus Y)$	$=$	$(\phi \curvearrowright X) \oplus (\phi \curvearrowright Y)$
(Ins6)	$\phi \curvearrowright (s \xrightarrow{a} s')$	$=$	$(\phi \curvearrowright s) \oplus (s \xrightarrow{a} s') \oplus (\phi \curvearrowright s')$
(Ins7)	$(\mathsf{f} \curvearrowright s) \oplus (s \xrightarrow{a} s')$	$=$	$(\mathsf{f} \curvearrowright s) \oplus s'$
(Ins8)	$(s \xrightarrow{a} s') \oplus (\mathsf{f} \curvearrowright s')$	$=$	$s \oplus (\mathsf{f} \curvearrowright s')$

Table 4: Additional axioms for signal insertion.

and (Ins4).

2.3 Process algebra

This subsection gives a brief summary of the ingredients of process algebra which will be used in later sections. We will make use of a discrete time extension of BPA (Basic Process Algebra) for relative time presented in [3], viz. $\mathrm{BPA}^{-}_{\mathrm{drt}}$. In BPA [5, 8], processes can be composed by sequential composition, written $P \cdot Q$, and alternative composition, written $P + Q$. The discrete time extensions are based on the division of time into slices indexed by natural numbers. These time slices represent time intervals of a length which corresponds to the time unit used. In $\mathrm{BPA}^{-}_{\mathrm{drt}}$, we have the constants $\mathsf{cts}(a)$ (one for each action a), $\mathsf{cts}(\delta)$ and $\dot{\delta}$. The $\mathsf{cts}(a)$ stand for a in the current time slice, $\mathsf{cts}(\delta)$ stands for a deadlock at the end of the current time slice, and $\dot{\delta}$ stands for an immediate deadlock (a process that comes to a full stop instantly, even in the middle of a time slice). In $\mathrm{BPA}^{-}_{\mathrm{drt}}$, we have, in addition to the sequential and alternative composition operators, the delay operator σ_{rel}. The process $\sigma_{\mathsf{rel}}(P)$ will start P in the next time slice. We will also make use of the initialization operator $\gg$ and the initial abstraction operator $\surd_{\mathsf{d}}$ for discrete time as in [3]. The process $n \gg P$ is the absolute time process obtained by initializing relative (or parametric) time process P at absolute time n, and $\surd_{\mathsf{d}}\, x \,.\, P$ is the parametric time process defined by $n \gg (\surd_{\mathsf{d}}\, x \,.\, P) = n \gg P[n/x]$. We refer to [3] for further details on $\mathrm{BPA}^{-}_{\mathrm{drt}}$ and the additional operators $\gg$ and $\surd_{\mathsf{d}}$. A model, extending the model for $\mathrm{BPA}^{-}_{\mathrm{drt}}$ given in [3] to deal with recursion, in which every set of guarded process equations has a unique solution can be obtained (see e.g. [15]). We will assume such a model.

We will further use the signal emission operators $\curvearrowright$ and $\curvearrowleft$ as in [2]. The expression $\phi \curvearrowright P$ is the process P where the proposition ϕ is made to hold at its start and the expression $P \overset{\bullet}{\frown} \phi$ is the process P where the proposition ϕ is made to hold at its termination. We refer to [2] for further details.

Besides, we will use the one-armed conditional operator $:\!\!:\!\!\rightarrow$. The expression $b :\!\!:\!\!\rightarrow P$ is to be read as "if b then P". The operator $:\!\!:\!\!\rightarrow$ can best be defined in terms of the two-armed conditional operator $\triangleleft\cdot\triangleright$, with the defining equations $P \triangleleft \mathsf{t} \triangleright Q = P$ and $P \triangleleft \mathsf{f} \triangleright Q = Q$, added to BPA in [1]. The one-armed conditional is then defined by $b :\!\!:\!\!\rightarrow P = P \triangleleft b \triangleright \dot{\delta}$.

We will also use the following abbreviation. Let $(P_i)_{i \in I}$ be an indexed set of process expressions where $I = \{i_1, \ldots, i_n\}$. Then, we write:

$$\sum_{i \in I} P_i \quad \text{for} \quad P_{i_1} + \ldots + P_{i_n}$$

We further use the convention that empty sums stand for $\dot{\delta}$.

3 Timed frames

In this section, we introduce timed frames. Simple timed frames differ from simple frames by containing two kinds of transitions: action steps and time steps. We study the connection between simple timed frames and discrete time processes in the setting of $\mathrm{BPA}^{-}_{\mathrm{drt}}$. We also extend simple timed frames with signal insertion, like in the untimed case.

The timed frames introduced in this section are only adequate to represent discrete time process behaviour in the relative time case. In Section 4, an extension is introduced to cover the absolute time case as well.

3.1 Example

First of all, we give a small example to illustrate the use of timed frames. The example concerns a simple telephone answering machine. We use the extension of frame algebra to timed frames for the description of the control component of the telephone answering machine. The example is based on a specification in φSDL due to Mauw [13].

In order to control the telephone answering, the control component of the answering machine has to communicate with the recorder component of the answering machine, the telephone connected with the answering machine, and the telephone network. When an incoming call is detected, the answering is not started immediately. If the incoming call is broken off or the receiver of the telephone is lifted within a period of 10 time units, answering is discontinued. Otherwise, an off-hook signal is issued to the network when this period has elapsed and a pre-recorded message is played. Upon termination of the message, a beep signal is issued to the network and the recorder is started. The recorder is stopped when the call is broken off, or when 30 time units have passed in case the call has not been broken off earlier. Thereafter, an on-hook signal is issued to the network.

It is obvious that the behaviour of the controller is time dependent. We will use time steps to describe this behaviour. Time steps represent the passage of time. They are denoted by terms of the form $s \xrightarrow{\sigma} s'$. Time steps are elaborated formally in Section 3.2. The behaviour of the controller is represented by the timed frame *TAMC* defined by:

$$
\begin{aligned}
&TAMC \;= \\
&\quad (0 \xrightarrow{\sigma} 0) \oplus (0 \xrightarrow{r(incoming\ call)} 1) \oplus \\
&\quad \textstyle\bigoplus_{i=1}^{10} ((i \xrightarrow{\sigma} S(i)) \oplus (i \xrightarrow{r(rcv\ lifted)} 0) \oplus (i \xrightarrow{r(end\ call)} 0)) \oplus \\
&\quad (11 \xrightarrow{s(off\text{-}hook)} 12) \oplus (12 \xrightarrow{s(play\ msg)} 13) \oplus \\
&\quad (13 \xrightarrow{\sigma} 13) \oplus (13 \xrightarrow{r(end\ msg)} 14) \oplus (13 \xrightarrow{r(end\ call)} 47) \oplus \\
&\quad (14 \xrightarrow{s(beep)} 15) \oplus (15 \xrightarrow{s(start\ rec)} 16) \oplus \\
&\quad \textstyle\bigoplus_{j=16}^{45} ((j \xrightarrow{\sigma} S(j)) \oplus (j \xrightarrow{r(end\ call)} 46)) \oplus \\
&\quad (46 \xrightarrow{s(stop\ rec)} 47) \oplus (47 \xrightarrow{s(on\text{-}hook)} 0)
\end{aligned}
$$

The following are some of the time related properties of the telephone answering machine that should be respected by this behaviour:

1. if an incoming call leads to playing the pre-recorded message, the receiver of the telephone has not been lifted during the first 10 time units from the detection of the incoming call;
2. if the recorder of the answering machine is started, it will always be stopped after at most 30 time units.

A suitable logic for reasoning about timed frames should allow to check such properties. By means of the process extraction operation defined in Section 3.2, timed frames can also be subject to process algebraic reasoning. Checking properties like the above-mentioned ones by process algebraic reasoning requires the use of an abstraction mechanism for discrete time processes such as the one presented in [4].

In case the telephone answering machine has to work together with another piece of telecommunications equipment, it is conceivable that, in order to co-operate properly, this piece of equipment has to inspect whether the state of the answering machine is one of playing, recording or otherwise. The representation of such inspection behaviour is possible if signal inserted states (see Section 3.3) and conditional transitions (see Section 4.1) are added to timed frames. Using signal insertion to assign to each state of *TAMC* a propositional formula that indicates whether it is a state of playing, recording or otherwise, we get the signal inserted timed frame $TAMC'$ defined by:

$$\begin{aligned} TAMC' \;=\; & \\ & TAMC \oplus \bigoplus_{i=0}^{12}((\neg playing \wedge \neg recording) \curvearrowright i) \oplus \\ & ((playing \wedge \neg recording) \curvearrowright 13) \oplus \bigoplus_{j=14}^{15}((\neg playing \wedge \neg recording) \curvearrowright j) \oplus \\ & \bigoplus_{k=16}^{46}((\neg playing \wedge recording) \curvearrowright k) \oplus ((\neg playing \wedge \neg recording) \curvearrowright 47) \end{aligned}$$

$TAMC'$ reveals some details of the states of the controller while $TAMC$ does not reveal any detail of them. This means that $TAMC'$ represents the behaviour of the controller at a lower level of abstraction than $TAMC$.

3.2 Simple timed frames

In the untimed case, there is only one kind of transitions, which we will call action steps. They represent the execution of actions. In the timed case, we consider an additional kind of transitions, which we will call time steps. They represent the passage of time. This fits in very well with the two phase notation for discrete time process algebra used in [3]. Without further extensions, simple timed frames are only adequate for relative timing. An extension for absolute timing and parametric timing is introduced in Section 4.2.

The signature extension for (simple) *timed* frames is as follows:

Sorts:
$\mathbb{F}_\mathrm{t}$ *timed frames*;
Functions:
$\xrightarrow{\sigma} : \mathbb{S}^2 \to \mathbb{F}_\mathrm{t}$ *time step construction.*

The axioms for simple timed frames are the axioms given in Table 1 (see Section 2.1) and the axioms given in Table 5. The axioms (TFA1) and (TFA2) are

(TFA1)	$s \oplus (s \xrightarrow{\sigma} s')$	$=$	$s \xrightarrow{\sigma} s'$
(TFA2)	$s' \oplus (s \xrightarrow{\sigma} s')$	$=$	$s \xrightarrow{\sigma} s'$

Table 5: Additional axioms for timed frames.

simply the counterparts of the axioms (FA5) and (FA6) for time steps. The axioms do not identify timed frames representing the same process behaviour if time determinism is assumed (in the face of states with more than one outgoing time step). Time determinism is assumed in the versions of discrete time process algebra presented in [3], where it corresponds to the axiom (DRT1) in the case of discrete time process algebra with relative timing: $\sigma_{\mathsf{rel}}(P) + \sigma_{\mathsf{rel}}(Q) = \sigma_{\mathsf{rel}}(P + Q)$ (also called the time factorization axiom). It is also assumed in that paper that a process will not become (dead)locked in the current time slice if there is the choice to proceed executing actions in the current time slice or any subsequent time slice. This property, called time persistency, is closely related to time determinism. The crucial axiom is (DRT4): $\mathsf{cts}(a) + \mathsf{cts}(\delta) = \mathsf{cts}(a)$, but it follows that $P + \mathsf{cts}(\delta) = P$ for all closed terms P except $\underset{\bullet}{\delta}$. Seeing that timed frames are intended to underlie models for theories concerning discrete time process behaviour, the axioms of timed frames should not be concerned with properties that are primarily relevant to the higher level of abstraction provided by discrete time processes. Therefore, time determinism and time persistency are not anticipated in the axioms for timed frames. Consequently, time steps are not treated different from action steps in the axioms for simple timed frames. However, the distinction between action steps and time steps is of vital importance to relate timed frames to discrete time processes.

In order to investigate the connection with discrete time process algebra, we introduce in Definition 3.3 a special kind of bisimulation, called σ-bisimulation, which takes into account the identifications due to time determinism and time persistency. That definition and subsequent ones need some conditions that are related to the transitions contained in a given frame. These frame conditions are as follows.

Definition 3.1.

$$[s \xrightarrow{a} s']_F = \begin{cases} \mathsf{t} & \text{if } (s \xrightarrow{a} s') \oplus F = F \\ \mathsf{f} & \text{otherwise} \end{cases}$$

$$[s \xrightarrow{\sigma} s']_F = \begin{cases} \mathsf{t} & \text{if } (s \xrightarrow{\sigma} s') \oplus F = F \\ \mathsf{f} & \text{otherwise} \end{cases}$$

$$[s \rightarrow s']_F = \begin{cases} \mathsf{t} & \text{if } [s \xrightarrow{a} s']_F = \mathsf{t} \text{ for some } a \text{ or } [s \xrightarrow{\sigma} s']_F = \mathsf{t} \\ \mathsf{f} & \text{otherwise} \end{cases}$$

$$[s \rightarrow_S^* s']_F = \begin{cases} \mathsf{t} & \text{if } [s \rightarrow s']_F = \mathsf{t} \text{ or} \\ & \quad [s \rightarrow s'']_F = \mathsf{t} \text{ and } [s'' \rightarrow_S^* s']_F = \mathsf{t} \text{ for some } s'' \in S \\ \mathsf{f} & \text{otherwise} \end{cases}$$

In the sequel, we will write $[s \xrightarrow{a} s']_F$ instead of $[s \xrightarrow{a} s']_F = \mathsf{t}$, $[s \xrightarrow{\sigma} s']_F$ instead of $[s \xrightarrow{\sigma} s']_F = \mathsf{t}$, etc. when this causes no ambiguity. We write $|F|$ for $\{s \in \mathbb{S} \mid \imath_{\mathbb{S}}(s) \oplus F = F\}$. For $s' \in |F|$, we write $[\not\xrightarrow{\sigma} s']_F$ to indicate that there exists no $s \in |F|$ such that $[s \xrightarrow{\sigma} s']_F$.

Below bisimulation and σ-bisimulation are defined as equivalences on pointed frames, i.e. frames equipped with a root marker and a termination marker. Pointed frames, which are closely related to transition systems, are defined first.

Definition 3.2. A *pointed* timed frame is a triple (F, p, q) where F is a timed frame and $p, q \in |F|$.

In the definition of σ-bisimulation given below, a relation on sets of states is used instead of a relation on states (as is usual). Rules 1–3 are the normal rules for (strong) bisimulation in the untimed case lifted to sets of states. The non-singleton sets are due to rule 4. This rule is the main rule for time steps. In a well-defined sense, it takes care of consistently identifying states reachable from the same state via one time step. In this way, time determinism is taken into account. However empty sets of states, standing for no state at all, may occur. Without rule 5, this would allow to relate two (sets of) states where the one has an outgoing time step and the other has no outgoing time step, provided that the time step concerned ends in a state without outgoing transitions. This should not be generally allowed. Rule 5 allows it only if the state without an outgoing time step has an outgoing action step. Thus time persistency is taken into account as well.

Definition 3.3. Let F and F' be timed frames, and let $p, q \in |F|$ and $p', q' \in |F'|$. The pointed timed frames (F, p, q) and (F', p', q') are *σ-bisimilar*, written $(F, p, q) \overset{\sigma}{\leftrightarrows} (F', p', q')$, if there exists a relation R on $\mathcal{P}(|F|) \times \mathcal{P}(|F'|)$ such that:

1. $R(\{p\}, \{p'\})$;
2. if $R(S, T)$ and $[s \xrightarrow{a} s']_F$ for some $s \in S$ and $s' \in |F| \setminus \{q\}$, then $[t \xrightarrow{a} t']_{F'}$ and $R(\{s'\}, \{t'\})$ for some $t \in T$ and $t' \in |F'| \setminus \{q'\}$;

2^c. rule 2 vice versa;

3. if $R(S, T)$ and $[s \xrightarrow{a} q]_F$ for some $s \in S$, then $[t \xrightarrow{a} q']_{F'}$ for some $t \in T$;

3^c. rule 3 vice versa;

4. if $R(S, T)$, then $R(S', T')$ where $S' = \{s' \in |F| \setminus \{q\} \mid \exists s \in S \cdot [s \xrightarrow{\sigma} s']_F\}$ and $T' = \{t' \in |F'| \setminus \{q'\} \mid \exists t \in T \cdot [t \xrightarrow{\sigma} t']_{F'}\}$;
5. if $R(S, T)$ and $[s \xrightarrow{\sigma} s']_F$ for some $s \in S$ and $s' \in |F|$, then $[t \rightarrow t']_{F'}$ for some $t \in T$ and $t' \in |F'|$;

5^c. rule 5 vice versa.

(F, p, q) and (F', p', q') are *bisimilar*, written $(F, p, q) \leftrightarrows (F', p', q')$, if there exists a relation R that satisfies, in addition to the above-mentioned conditions, the following one:

6. if $R(S, T)$, then $card(S) = card(T) \leq 1$.

We also define time determinism and time persistency for frames, because together they characterize the kind of frames that corresponds to the timed transition systems that underlie the model of discrete time process algebra with relative timing presented in [3]. Frames of this kind are called proper timed frames.

Definition 3.4. A timed frame F is *σ-deterministic* if it satisfies:

if $[s \xrightarrow{\sigma} t]_F$ and $[s \xrightarrow{\sigma} t']_F$ for some $s, t, t' \in |F|$, then $t = t'$.

A timed frame F is *σ-persistent* if it satisfies:

if $[s \xrightarrow{a} t]_F$ and $[s \xrightarrow{\sigma} t']_F$ for some $s, t, t' \in |F|$, then $[t' \rightarrow t'']_F$ for some $t'' \in |F|$.

A timed frame F is *proper* if it is σ-deterministic and σ-persistent.

In [3], discrete time process algebra with relative timing is based on transition systems corresponding to pointed frames (F, p, q) where F is proper and q has no incoming time steps (i.e. $[\xrightarrow{\sigma}\!\!\!\!\not\;\; q]_F$). For pointed frames satisfying these conditions, the definition of bisimulation given here is equivalent to the one given in that paper. Three kinds of termination states can be distinguished in the transition systems considered in [3]: states representing successful termination, states representing immediate deadlock, and states representing deadlock in the current time slice. States of the last kind are no termination states in pointed frames; they are modelled as states with one outgoing transition, being a time step, to an immediate deadlock state. Thus, deadlock in the current time slice is identified with immediate deadlock at the beginning of the next time slice. This is in accordance with [3] where it corresponds to the axiom (DRT3): $\sigma_{\mathsf{rel}}(\dot{\delta}) = \mathsf{cts}(\delta)$. Besides, pointed frames have at most one successful termination state. This does not give any loss of generality in case of relative timing.

According to the following two lemmas, every pointed frame is σ-bisimilar to one of the pointed frames that correspond to the transition systems that underlie the model of discrete time process algebra with relative timing presented in [3]. This fact will be used in the proof of Lemmas 3.9 and 3.10.

Lemma 3.5. *Every pointed timed frame (F, p, q) is σ-bisimilar to a pointed timed frame (F', p, q) where $[\xrightarrow{\sigma}\!\!\!\!\not\;\; q]_{F'}$.*

Proof. We show the existence of such a pointed frame by means of a transformation of (F, p, q). Add a fresh state. Replace q in each of its incoming time steps by the fresh state. Let F' be the frame obtained in this way. It follows immediately from the transformation that $[\xrightarrow{\sigma}\!\!\!\!\not\;\; q]_{F'}$ and that $(F, p, q) \overset{\sigma}{\leftrightarrow} (F', p, q)$. □

Lemma 3.6. *Every pointed timed frame (F, p, q) where $[\xrightarrow{\sigma}\!\!\!\!\not\;\; q]_F$ is σ-bisimilar to a pointed timed frame (F', p, q) where F' is proper and $[\xrightarrow{\sigma}\!\!\!\!\not\;\; q]_{F'}$.*

Proof. We show the existence of such a pointed timed frame by means of a transformation of (F, p, q). First identify the states reachable from the root state p via one time step. Then remove the remaining time step if it ends in a state without outgoing transitions, provided that p has outgoing action steps as well. Repeat this for all states reachable from p via one transition in the frame obtained in this manner, and so on. Let F' be the resulting timed frame. It follows immediately that this frame is proper and $[\xrightarrow{\sigma}\!\!\!\!\not\;\; q]_{F'}$. Besides, it follows directly from the transformation that $(F, p, q) \overset{\sigma}{\leftrightarrow} (F', p, q)$. □

According to the following lemma, $\overset{\sigma}{\leftrightarrow}$ and $\leftrightarrow$ coincide for the pointed frames that correspond to the transition systems that underlie the model of discrete time process algebra with relative timing presented in [3]. This fact will also be used in the proof of Lemmas 3.9 and 3.10.

Lemma 3.7. *For proper timed frames F and F', $p, q \in |F|$ and $p', q' \in |F'|$ such that $[\xrightarrow{\sigma}\!\!\!\!\!/\; q]_F$ and $[\xrightarrow{\sigma}\!\!\!\!\!/\; q']_{F'}$, $(F, p, q) \overset{\sigma}{\leftrightarrows} (F', p', q')$ iff $(F, p, q) \leftrightarrows (F', p', q')$.*

Proof. $\leftrightarrows$ implies $\overset{\sigma}{\leftrightarrows}$ by definition. For the other direction, we use the fact that there exists a relation satisfying conditions 1–5 of Definition 3.3. Let R be such a relation. Then R has a subset R' defined by $R'(S, T)$ iff $R(S, T)$, $card(S), card(T) \leq 1$ and $card(S) = card(T)$. It follows that R' satisfies conditions 1–5 of Definition 3.3 as well: if $R(S, T)$ but not $R'(S, T)$ then $R(S, T)$ is not necessary to satisfy conditions 1–5, seeing that the necessity would imply that either F or F' does not fulfil both conditions for properness and would thus imply contradiction. Here we use that the need for $R(S, T)$ where not $card(S), card(T) \leq 1$ implies that F or F' is not σ-deterministic, and that the need for $R(S, T)$ where not $card(S) = card(T)$ likewise implies that F or F' is not σ-persistent. In addition, it is immediate that R' satisfies condition 6 of Definition 3.3. Consequently, $(F, p, q) \leftrightarrows (F', p', q')$. □

Below a process extraction operation is defined on pointed timed frames. It is defined such that σ-bisimilarity of the pointed frames coincides with bisimilarity of the extracted processes (see further Lemma 3.9). Process extraction is such that all states without an outgoing transition are interpreted as states representing immediate deadlock if it is not the state marked as termination state. For incoming action steps, the termination state is interpreted as a state representing successful termination. For incoming time steps, it is interpreted as a state representing immediate deadlock (in this case, it does not make sense to interpret it as a state representing successful termination).

Definition 3.8. Let F be a timed frame and $s, t \in \mathbb{S}$. Then $s \curvearrowright t F$ is the process X_s given by the following finite set of process equations:

$$\{X_{s'} = P_{s'} \mid s' = s \text{ or } ([s \rightarrow^*_{|F| \setminus \{t\}} s']_F \text{ and } s' \neq t)\}$$

where

$$P_{s'} = \sum_{a \in A} \left(\begin{array}{l} [s' \xrightarrow{a} t]_F :\!\!:\!\!\rightarrow \mathsf{cts}(a) + \\ [s' \xrightarrow{\sigma} t]_F :\!\!:\!\!\rightarrow \mathsf{cts}(\delta) + \\ \displaystyle\sum_{s'' \in |F| \setminus \{t\}} ([s' \xrightarrow{a} s'']_F :\!\!:\!\!\rightarrow \mathsf{cts}(a) \cdot X_{s''}) + \\ \displaystyle\sum_{s'' \in |F| \setminus \{t\}} ([s' \xrightarrow{\sigma} s'']_F :\!\!:\!\!\rightarrow \sigma_{\mathsf{rel}}(X_{s''})) \end{array} \right)$$

Recall that we assume a model in which every set of guarded process equations has a unique solution. Note further that a set of linear process equations can be obtained here by rewriting each term of the form $b :\!\!:\!\!\rightarrow P$ in accordance with the value of the condition b in the frame F.

The following two lemmas are about two very desirable properties of the process extraction operation defined above. The first one is that σ-bisimilarity of the pointed timed frames coincides with bisimilarity of the extracted processes. The second property is that for each regular relative time process there is a timed frame of which it is the extracted process (up to bisimilarity). These properties give a clear picture of the connection between timed frames and discrete time processes.

Lemma 3.9. *For timed frames F and F', $p, q \in |F|$ and $p', q' \in |F'|$, $(F, p, q) \overset{\sigma}{\underline{\leftrightarrow}} (F', p', q')$ iff $p \curvearrowright qF \underline{\leftrightarrow} p' \curvearrowright q'F'$.*

Proof. Let $\mathrm{C}(F, p, q)$ and $\mathrm{C}(F', p', q')$ be the pointed frames obtained by applying the transformations described in the proofs of Lemmas 3.5 and 3.6, in that order, to (F, p, q) and (F', p', q'), respectively. By these lemmas and Lemma 3.7, $(F, p, q) \overset{\sigma}{\underline{\leftrightarrow}} (F', p', q')$ iff $\mathrm{C}(F, p, q) \underline{\leftrightarrow} \mathrm{C}(F', p', q')$. So it remains to be proved that $p \curvearrowright qF \underline{\leftrightarrow} p' \curvearrowright q'F'$ iff $\mathrm{C}(F, p, q) \underline{\leftrightarrow} \mathrm{C}(F', p', q')$. This follows directly from the observation that $\curvearrowright$ and C are defined such that, for each extracted process $p \curvearrowright qF$, the pointed frame $\mathrm{C}(F, p, q)$ is the canonical process graph determined by the finite set of linear process equations obtained from the equations for $p \curvearrowright qF$ according to Definition 3.8 as described above. □

Lemma 3.10. *A relative time process P is regular iff $P \underline{\leftrightarrow} p \curvearrowright qF$ for some (finite) timed frame F and some states $p, q \in |F|$.*

Proof. Each regular process P is the solution of a finite set of linear process equations E. The set E determines a canonical process graph (F, p, q), which conversely determines a finite set of linear process equations E'. The set E' in its turn determines a canonical process graph (F', p', q') which is bisimilar to (F, p, q). Hence, the solutions of E and E' are bisimilar. In addition, we have that $p \curvearrowright qF$ is the solution of E'. Consequently, $P \underline{\leftrightarrow} p \curvearrowright qF$. The other direction is trivial because the timed frames considered here have a finite number of states and transitions. □

We envisage a systematic approach to devise suitable logics for reasoning about discrete time processes, which starts with full predicate logic over timed frames. Obviously, this simple logic permits to express properties that are not at all relevant to discrete time processes. The way to remedy this is to determine (modal) fragments that distinguish up to an equivalence that provides an adequate level of abstraction for discrete time processes (for examples in the time free case, see e.g. [6]). Lemma 3.9 supports our claim that σ-bisimulation is a suitable equivalence for this purpose. Lemma 3.10 makes clear that timed frames are adequate to represent all regular discrete relative time processes.

Note that by means of process extraction, pointed timed frames can be subject to process algebraic reasoning. Process extraction yields a process which is given by a set of process equations. It is frequently the case that the extracted process can also be given by a process term of a certain form that provides a better starting point for the process algebraic reasoning. The rule given below takes this up. It allows to extract processes given by a process term of the form $\lfloor P \rfloor^{\omega}$ if certain conditions are satisfied by the pointed frame concerned. The *unbounded* *start delay* operator $\lfloor\cdot\rfloor^\omega$ originates from ATP [14]. The process $\lfloor P\rfloor^\omega$ will start P in the current time slice or any future time slice. Equations in the setting of $\mathrm{BPA}^-_{\mathrm{drt}}$ are given in [4]. The rule for extracting processes with an unbounded start delay is as follows:

$$\frac{\forall s \in |F| \cdot [p \xrightarrow{\sigma} s]_F = \mathsf{f} \qquad [p \to^*_{|F|\setminus\{q\}} p]_F = \mathsf{f} \qquad p \neq q}{p \curvearrowright q\ ((p \xrightarrow{\sigma} p) \oplus F) = \lfloor p \curvearrowright qF\rfloor^\omega}$$

Note further that this rule provides a clear-cut characterization of unbounded start delay from the viewpoint of timed frames. Rules concerned with other operators on discrete time processes can be devised as well.

3.3 Signal inserted timed frames

Timed frames are extended with signal insertion like in the untimed case by adding propositions and the signal insertion operation $\curvearrowright$:

Sorts:

$\mathbb{P}$		propositions;
$\langle \mathbb{F}_\mathrm{t}, \mathbb{P}\rangle$		*signal inserted timed frames*;

Constants & Functions:

p	$: \mathbb{P}$	for each $p \in \mathbb{P}_{at}$;
t	$: \mathbb{P}$	true;
f	$: \mathbb{P}$	false;
$\neg$	$: \mathbb{P} \to \mathbb{P}$	negation;
$\to$	$: \mathbb{P}^2 \to \mathbb{P}$	implication;
$\curvearrowright$	$: \mathbb{P} \times \langle \mathbb{F}_\mathrm{t}, \mathbb{P}\rangle \to \langle \mathbb{F}_\mathrm{t}, \mathbb{P}\rangle$	*signal insertion.*

The axioms for signal inserted timed frames are the axioms given in Table 1 (see Section 2.1), Table 4 (see Section 2.2), Table 5 (see Section 3.2) and the axioms given in Table 6. The axiom (TIns1) is simply the counterpart of the

(TIns1)	$\phi \curvearrowright (s \xrightarrow{\sigma} s')$	$=$	$(\phi \curvearrowright s) \oplus (s \xrightarrow{\sigma} s') \oplus (\phi \curvearrowright s')$
(TIns2)	$(\phi \curvearrowright s) \oplus (s \xrightarrow{\sigma} s')$	$=$	$(s \xrightarrow{\sigma} s') \oplus (\phi \curvearrowright s')$

Table 6: Additional axioms for signal inserted timed frames.

axiom (Ins6) for time steps. The axiom (TIns2) reflects the intuition that the passage of time cannot change the propositions that hold in the current state of a process. This is in accordance with discrete relative time process algebra with propositional signals where it corresponds to the axiom $\phi \curvearrowright \sigma_{\mathsf{rel}}(x) = \sigma_{\mathsf{rel}}(\phi \curvearrowright x)$. Axiom (TIns2) entails that inconsistent states, i.e. states where f holds, remain inconsistent with progress of time. Thus, one inconsistent state would render all states inconsistent if there were also counterparts of the axioms (Ins7) and (Ins8) for time steps.

Below, we will use an operation that extracts from a frame F the propositional formula assigned to a state $s \in |F|$. The proposition extraction operation χ is defined in Table 7. We will write $\chi(F, S)$, where $S \subseteq |F|$, for $\bigwedge_{s \in S} \chi(F, s)$.

(Ext1)	$\chi(\emptyset, s)$	$=$	t	
(Ext2)	$\chi(s' \oplus X, s)$	$=$	$\chi(X, s)$	
(Ext3)	$\chi((s' \xrightarrow{a} s'') \oplus X, s)$	$=$	$\chi(X, s)$	
(Ext4)	$\chi((s' \xrightarrow{\sigma} s'') \oplus X, s)$	$=$	$\chi((\chi(X, s'') \curvearrowright s') \oplus (\chi(X, s') \curvearrowright s'') \oplus X, s)$	
(Ext5)	$\chi((\phi \curvearrowright s) \oplus X, s)$	$=$	$\phi \wedge \chi(X, s)$	
(Ext6)	$\chi((\phi \curvearrowright s') \oplus X, s)$	$=$	$\chi(X, s)$	if $s' \neq s$ and $s' \oplus X \neq X$

Table 7: Axioms for proposition extraction.

The right-hand side of axiom (Ext4) is not simply $\chi(X, s)$ because, according to axiom (TIns2), a time step $s' \xrightarrow{\sigma} s''$ implies that the propositions holding in state s' hold in state s'' as well, and vice versa. Note that, because of possible incoming or outgoing time steps of s' in X, $\chi((\phi \curvearrowright s') \oplus X, s)$ and $\chi(X, s)$ could be wrongly identified if the condition $s' \oplus X \neq X$ had been omitted in axiom (Ext6). In case $s' \oplus X = X$ because X contains subterms of the form $\psi \curvearrowright s'$, axiom (Ins4) becomes essential to reduce the term $\chi((\phi \curvearrowright s') \oplus X, s)$ to a term that does not contain applications of proposition extraction.

The definition of σ-bisimulation for simple timed frames (Definition 3.3) must be adapted. Rule 1 is unchanged. Rule 2 is added to prevent abstraction from the propositional formulae assigned to states. Rules 3–6 are the rules 2–5 from the original definition adapted to take into account that an inconsistent state can neither be entered nor left in spite of possible incoming and outgoing transitions. In this definition (and also in Definition 4.1), we write $\phi = \psi$ to indicate that for all valuations $v : \mathbb{P}_{at} \to \{\mathsf{t}, \mathsf{f}\}$, $v(\phi) = \mathsf{t}$ iff $v(\psi) = \mathsf{t}$. Furthermore, we write $\phi \neq \mathsf{f}$ to indicate that there exists a valuation v such that $v(\phi) = \mathsf{t}$.

Definition 3.11. Let F and F' be signal inserted timed frames, and let $p, q \in |F|$ and $p', q' \in |F'|$. The pointed frames (F, p, q) and (F', p', q') are σ-bisimilar, written $(F, p, q) \overset{\sigma}{\leftrightarrow} (F', p', q')$ if there exists a relation R on $\mathcal{P}(|F|) \times \mathcal{P}(|F'|)$ such that:

1. $R(\{p\}, \{p'\})$;
2. if $R(S, T)$, then $\chi(F, S) = \chi(F', T)$;
3. if $R(S, T)$, $\chi(F, S) \neq \mathsf{f}$ and $[s \xrightarrow{a} s']_F$ for some $s \in S$ and $s' \in |F| \setminus \{q\}$, then $[t \xrightarrow{a} t']_{F'}$ and $R(\{s'\}, \{t'\})$ for some $t \in T$ and $t' \in |F'| \setminus \{q'\}$;

3^c. rule 3 vice versa;

4. if $R(S,T)$, $\chi(F,S) \neq \mathsf{f}$ and $[s \xrightarrow{a} q]_F$ for some $s \in S$, then $[t \xrightarrow{a} q']_{F'}$ for some $t \in T$ and $\chi(F,q) = \chi(F',q') \neq \mathsf{f}$;

4^c. rule 4 vice versa;

5. if $R(S,T)$ and $\chi(F,S) \neq \mathsf{f}$, then $R(S',T')$ where $S' = \{s' \in |F| \setminus \{q\} \mid \exists s \in S \cdot [s \xrightarrow{\sigma} s']_F\}$ and $T' = \{t' \in |F'| \setminus \{q'\} \mid \exists t \in T \cdot [t \xrightarrow{\sigma} t']_{F'}\}$;

6. if $R(S,T)$, $\chi(F,S) \neq \mathsf{f}$ and $[s \xrightarrow{\sigma} s']_F$ for some $s \in S$ and $s' \in |F|$, then $[t \to t']_{F'}$ for some $t \in T$ and $t' \in |F'|$;

6^c. rule 6 vice versa.

The definition of process extraction for simple timed frames (Definition 3.8) must also be adapted to take into account the propositional formulae assigned to states.

Definition 3.12. Let F be a signal inserted timed frame and $s, t \in \mathbb{S}$. Then $s \curvearrowright tF$ is the process X_s given by the following finite set of process equations:

$$\{X_{s'} = P_{s'} \mid s' = s \text{ or } ([s \to^*_{|F| \setminus \{t\}} s']_F \text{ and } s' \neq t)\}$$

where

$$P_{s'} \;=\; \chi(F,s') \curvearrowright \sum_{a \in A} \left(\begin{array}{l} [s' \xrightarrow{a} t]_F :\!:\!\to (\mathsf{cts}(a) \curvearrowleft \chi(F,t)) + \\ [s' \xrightarrow{\sigma} t]_F :\!:\!\to \;\mathsf{cts}(\delta) + \\ \displaystyle\sum_{s'' \in |F| \setminus \{t\}} ([s' \xrightarrow{a} s'']_F :\!:\!\to \mathsf{cts}(a) \cdot X_{s''}) + \\ \displaystyle\sum_{s'' \in |F| \setminus \{t\}} ([s' \xrightarrow{\sigma} s'']_F :\!:\!\to \sigma_{\mathsf{rel}}(X_{s''})) \end{array} \right)$$

The definitions of bisimulation and proper timed frames for signal inserted timed frames are as for simple timed frames. Lemmas 3.5–3.7, 3.9 and 3.10, which concern simple timed frames, go through for signal inserted timed frames.

The following remarks are worth mentioning. In Section 3.2, the notation $|F|$ was introduced for the set of all states contained in frame F. It is not difficult to define a corresponding operation $|\cdot|$ on frames by means of equational axioms instead. In case of signal inserted timed frames, the operation concerned removes transitions and undoes signal insertions. An operation $|\cdot|_{\mathsf{s}}$ that only removes transitions can be defined analogously. Using this operation, axiom (Ext6) can be replaced by

$$\text{(Ext6}'\text{)} \qquad \chi((\phi \curvearrowright s') \oplus |X|_{\mathsf{s}}, s) \;=\; \chi(|X|_{\mathsf{s}}, s) \quad \text{if } s' \neq s$$

4 Conditional transitions

In this section, we introduce two kinds of conditional transitions in the setting of timed frames. In Section 3, we already added signal inserted states to timed frames. Here we complement signal inserted states with conditional transitions where the truth of the condition is state dependent. The condition is a propositional formula that may hold in a signal inserted state; its truth is determined by the propositional formula assigned to the starting state of the transition concerned. We also add conditional transitions to timed frames where the truth of the condition is time dependent. With the further structure provided by the latter kind of conditional transitions, timed frames are also adequate to represent discrete time process behaviour in the absolute time case. In order to distinguish the latter kind of conditional transitions from the former one, we will use the prefix *time* for conditional transitions of the latter kind. The integration of both kinds of conditional transitions is not carried out because it is so trivial.

Conditional transitions should not be confused with conditional frames. Conditional frames do not introduce new kinds of states or transitions; they are just a means to describe frames conditionally. Conditional frames are briefly treated in an appendix.

4.1 State dependent conditional transitions

In this subsection, we complement signal inserted states with conditional transitions. These transitions are labelled with an action (or σ in the case of time steps) and a propositional formula. Whether an outgoing conditional transition of a state can be performed or not in that state depends on the propositional formula assigned to the state. The further structure provided by adding conditional transitions to signal inserted timed frames gives a semantic basis for discrete time extensions of process algebra with propositional signals [2].

The signature extension for signal inserted timed frames with *conditional transitions* is as follows:

Sorts:

$\langle \mathbb{F}_t, \mathbb{P} \rangle_c$	signal inserted timed frames with *conditional transitions*;

Functions:

$\xrightarrow{\cdot,a}$	$: \mathbb{P} \times \mathbb{S}^2 \to \langle \mathbb{F}_t, \mathbb{P} \rangle_c$	*conditional action step construction* (one for each $a \in A$);
$\xrightarrow{\cdot,\sigma}$	$: \mathbb{P} \times \mathbb{S}^2 \to \langle \mathbb{F}_t, \mathbb{P} \rangle_c$	*conditional time step construction.*

The signature of signal inserted timed frames with conditional transitions is graphically presented in Figure 3. The additional axioms for conditional transitions are given in Table 8. The following equations are derivable from the axioms (Ins4) (see Table 4) and (Con1)–(Con4):

$$
\begin{aligned}
((\phi \wedge \psi) \curvearrowright s) \oplus (s \xrightarrow{\phi,a} s') &= ((\phi \wedge \psi) \curvearrowright s) \oplus s \xrightarrow{a} s' \\
(\phi \curvearrowright s) \oplus (s \xrightarrow{\phi \vee \psi,a} s') &= (\phi \curvearrowright s) \oplus s \xrightarrow{a} s' \\
(\phi \curvearrowright s) \oplus (s \xrightarrow{\neg\phi,a} s') &= (\phi \curvearrowright s) \oplus s \oplus s'
\end{aligned}
$$

These equations make clear that a conditional action step $s \xrightarrow{\phi,a} s'$ can only

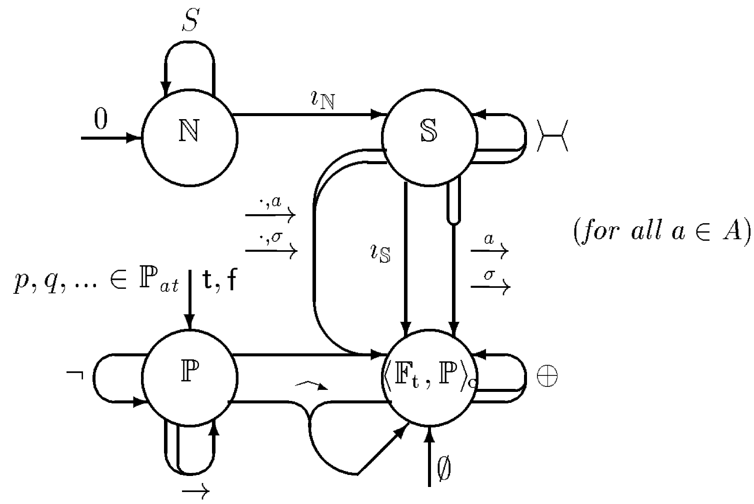

Figure 3: Signature of signal inserted timed frames with conditional transitions.

be performed if ϕ holds in s. The corresponding equations for conditional time steps are derivable from the axioms (Ins4) and (TCon1)–(TCon4).

Additional frame conditions are needed to extend the definition of σ-bisimulation to conditional transitions:

$$[s \xrightarrow{\phi,a} s']_F = \begin{cases} \mathsf{t} & \text{if } (s \xrightarrow{\phi,a} s') \oplus F = F \\ \mathsf{f} & \text{otherwise} \end{cases}$$

$$[s \xrightarrow{\phi,\sigma} s']_F = \begin{cases} \mathsf{t} & \text{if } (s \xrightarrow{\phi,\sigma} s') \oplus F = F \\ \mathsf{f} & \text{otherwise} \end{cases}$$

(Con1)	$s \xrightarrow{\mathsf{t},a} s'$	$=$	$s \xrightarrow{a} s'$
(Con2)	$s \xrightarrow{\mathsf{f},a} s'$	$=$	$s \oplus s'$
(Con3)	$(s \xrightarrow{\phi \vee \psi,a} s')$	$=$	$(s \xrightarrow{\phi,a} s') \oplus (s \xrightarrow{\psi,a} s')$
(Con4)	$(\phi \rightharpoonup s) \oplus (s \xrightarrow{\phi \wedge \psi,a} s')$	$=$	$(\phi \rightharpoonup s) \oplus (s \xrightarrow{\psi,a} s')$
(TCon1)	$s \xrightarrow{\mathsf{t},\sigma} s'$	$=$	$s \xrightarrow{\sigma} s'$
(TCon2)	$s \xrightarrow{\mathsf{f},\sigma} s'$	$=$	$s \oplus s'$
(TCon3)	$(s \xrightarrow{\phi \vee \psi,\sigma} s')$	$=$	$(s \xrightarrow{\phi,\sigma} s') \oplus (s \xrightarrow{\psi,\sigma} s')$
(TCon4)	$(\phi \rightharpoonup s) \oplus (s \xrightarrow{\phi \wedge \psi,\sigma} s')$	$=$	$(\phi \rightharpoonup s) \oplus (s \xrightarrow{\psi,\sigma} s')$

Table 8: Additional axioms for conditional transitions.

Note that the definition of $[s \rightarrow_S^* s']_F$ actually needs a trivial adaptation to cover conditional transitions as well.

The definition of σ-bisimulation for signal inserted timed frames (Definition 3.11) must be adapted. Rules 1 and 2 are unchanged. Rules 3–6 are generalized for conditional transitions in accordance with the definition of bisimulation in [2]. They take into account that, with conditional transitions, the outgoing transitions of a state can not always be performed.

Definition 4.1. Let F and F' be signal inserted timed frames with conditional transitions, and let $p, q \in |F|$ and $p', q' \in |F'|$. The pointed frames (F, p, q) and (F', p', q') are *σ-bisimilar*, written $(F, p, q) \overset{\sigma}{\leftrightarrow} (F', p', q')$, if there exists a relation R on $\mathcal{P}(|F|) \times \mathcal{P}(|F'|)$ such that:

1. $R(\{p\}, \{p'\})$;
2. if $R(S, T)$, then $\chi(F, S) = \chi(F', T)$;
3. if $R(S, T)$ and $[s \xrightarrow{\phi, a} s']_F$ for some $s \in S$ and $s' \in |F| \setminus \{q\}$, then, for all valuations v such that $v(\chi(F, S)) = \mathsf{t}$ and $v(\phi) = \mathsf{t}$, there exists a proposition ψ such that $v(\psi) = \mathsf{t}$, $[t \xrightarrow{\psi, a} t']_{F'}$ and $R(\{s'\}, \{t'\})$ for some $t \in T$ and $t' \in |F'| \setminus \{q'\}$;

3^c. rule 3 vice versa;

4. if $R(S, T)$ and $[s \xrightarrow{\phi, a} q]_F$ for some $s \in S$, then, for all valuations v such that $v(\chi(F, S)) = \mathsf{t}$ and $v(\phi) = \mathsf{t}$, there exists a proposition ψ such that $v(\psi) = \mathsf{t}$ and $[t \xrightarrow{\psi, a} q']_{F'}$ for some $t \in T$, and $\chi(F, q) = \chi(F', q') \neq \mathsf{f}$;

4^c. rule 4 vice versa;

5. if $R(S, T)$, then $R(S_\phi, T_\psi)$, where $S_\phi = \{s' \in |F| \setminus \{q\} \mid \exists s \in S \cdot [s \xrightarrow{\phi, \sigma} s']_F\}$ and $T_\psi = \{t' \in |F'| \setminus \{q'\} \mid \exists t \in T \cdot [t \xrightarrow{\psi, \sigma} t']_{F'}\}$, for all propositions ϕ and ψ such that, for all valuations v such that $v(\chi(F, S)) = \mathsf{t}$, $v(\phi) = \mathsf{t}$ iff $v(\psi) = \mathsf{t}$;
6. if $R(S, T)$ and $[s \xrightarrow{\phi, \sigma} s']_F$ for some $s \in S$ and $s' \in |F|$, then, for all valuations v such that $v(\chi(F, S)) = \mathsf{t}$ and $v(\phi) = \mathsf{t}$, there exists a proposition ψ such that $v(\psi) = \mathsf{t}$ and either $[t \xrightarrow{\psi, a} t']_{F'}$ or $[t \xrightarrow{\psi, \sigma} t']_{F'}$ for some $a \in A$, $t \in T$ and $t' \in |F'|$;

6^c. rule 6 vice versa.

The definition of process extraction for signal inserted timed frames (Definition 3.12) must also be adapted for the extension with conditional transitions. In this definition (and also in Definition 4.4), we write Φ_F for the finite set $\{\phi \mid \exists s', s'' \in |F| \cdot (\exists a \in A \cdot [s' \xrightarrow{\phi, a} s'']_F) \vee [s' \xrightarrow{\phi, \sigma} s'']_F\}$.

Definition 4.2. Let F be a signal inserted timed frame with conditional transitions and $s, t \in \mathbb{S}$. Then $s \curvearrowright tF$ is the process X_s given by the following finite set of process equations:

$$\{X_{s'} = P_{s'} \mid s' = s \text{ or } ([s \rightarrow_{|F| \setminus \{t\}}^* s']_F \text{ and } s' \neq t)\}$$

where

$$P_{s'} = \chi(F, s') \frown \sum_{a \in A} \sum_{\phi \in \Phi_F} \left(\begin{array}{l} [s' \xrightarrow{\phi,a} t]_F :: \rightarrow ((\phi :: \rightarrow \mathsf{cts}(a)) \frown \chi(F, t)) + \\ [s' \xrightarrow{\phi,\sigma} t]_F :: \rightarrow \ (\phi :: \rightarrow \mathsf{cts}(\delta)) + \\ \displaystyle\sum_{s'' \in |F| \setminus \{t\}} ([s' \xrightarrow{\phi,a} s'']_F :: \rightarrow (\phi :: \rightarrow \mathsf{cts}(a) \cdot X_{s''})) + \\ \displaystyle\sum_{s'' \in |F| \setminus \{t\}} ([s' \xrightarrow{\phi,\sigma} s'']_F :: \rightarrow (\phi :: \rightarrow \sigma_{\mathsf{rel}}(X_{s''}))) \end{array} \right)$$

The finite set Φ_F is used instead of the infinite set of all terms of sort $\mathbb{P}$ since only finite sums are supported in the setting of $\mathrm{BPA}^{-}_{\mathrm{drt}}$.

The definitions of σ-bisimulation and process extraction for the extension with conditional transitions are such that Lemmas 3.9 and 3.10, connecting σ-bisimulation for frames with bisimulation for processes and frames with regular processes, respectively, go through for this extension as well.

4.2 Time dependent conditional transitions

Time stamping of states is the common usage to represent discrete time process behaviour in the absolute time case, but time stamping of transitions could be used as well. In simple timed frames, states nor transitions have time stamps. In this subsection, we extend simple timed frames with conditional transitions where the truth of the condition is time dependent. This can be regarded as a generalization of time stamping of transitions. The atomic time conditions are t, f, $\mathsf{sl}(m)$ and $\mathsf{sl}_{>}(m)$ ($m \in \mathbb{N}_1$). The condition $\mathsf{sl}(m)$ is true during time slice m, and the condition $\mathsf{sl}_{>}(m)$ is true from time slice $m+1$ on. Further time conditions can be built from the atomic time conditions by means of the connectives $\neg$ and $\rightarrow$. Simple timed frames extended with time conditional transitions are adequate to represent discrete time process behaviour in the absolute time case.

The signature extension for simple timed frames with *time* conditional transitions is as follows:

Sorts:

$\mathbb{P}_{\mathrm{t}}$	*time conditions*;
$\langle \mathbb{F}_{\mathrm{t}}, \mathbb{P}_{\mathrm{t}} \rangle_{\mathrm{c}}$	simple timed frames with *time conditional transitions*;

Constants & Functions:

t	$: \mathbb{P}_{\mathrm{t}}$	true;
f	$: \mathbb{P}_{\mathrm{t}}$	false;
sl	$: \mathbb{N} \rightarrow \mathbb{P}_{\mathrm{t}}$	*slice equal*;
$\mathsf{sl}_{>}$	$: \mathbb{N} \rightarrow \mathbb{P}_{\mathrm{t}}$	*slice later than*;
$\neg$	$: \mathbb{P}_{\mathrm{t}} \rightarrow \mathbb{P}_{\mathrm{t}}$	negation;
$\rightarrow$	$: \mathbb{P}_{\mathrm{t}}^2 \rightarrow \mathbb{P}_{\mathrm{t}}$	implication;
$\xrightarrow{\cdot,a}$	$: \mathbb{P}_{\mathrm{t}} \times \mathbb{S}^2 \rightarrow \langle \mathbb{F}_{\mathrm{t}}, \mathbb{P}_{\mathrm{t}} \rangle_{\mathrm{c}}$	*time conditional action step construction* (one for each $a \in A$);
$\xrightarrow{\cdot,\sigma}$	$: \mathbb{P}_{\mathrm{t}} \times \mathbb{S}^2 \rightarrow \langle \mathbb{F}_{\mathrm{t}}, \mathbb{P}_{\mathrm{t}} \rangle_{\mathrm{c}}$	*time conditional time step construction.*

The additional axioms for time conditional transitions are simply the axioms (Con1)–(Con3) and (TCon1)–(TCon3) given in Table 8 (see Section 4.1), understanding that the range of the meta-variables is properly changed.

In [3], discrete time process algebra with absolute timing is based on transition systems where states are time stamped. These transition systems are in addition such that, for each action step $s \xrightarrow{a} s'$, the time stamp of s' must be equal to the time stamp of s; and for each time step $s \xrightarrow{\sigma} s'$, the time stamp of s' must be the successor of the time stamp of s. A state with time stamp n represents a state that can only be entered or left during time slice $n+1$. The conditions on the time stamps of states are needed because an action step does not change the time slice and a time step changes it to the next one. This also permits to time stamp transitions instead or, equivalently, to use time conditional transitions with conditions of the form $\mathsf{sl}(m)$ $(m > 0)$. The intended time restrictions on the performance of this simple kind of time conditional transitions can be explained as follows: in case of an action step $s \xrightarrow{\mathsf{sl}(m),a} s'$, state s must be left during time slice m and state s' must be entered during time slice m; and in case of a time step $s \xrightarrow{\mathsf{sl}(m),\sigma} s'$, state s must be left during time slice m and state s' must be entered during time slice $m+1$. The definition of process extraction from simple timed frames with time conditional transitions reflects this.

In order to adapt the definition of process extraction to time conditional transitions, we need the intended time dependent valuation of time conditions.

Definition 4.3. For each $n \in \mathbb{N}$ the valuation $\cdot^n : \mathbb{P}_{\mathsf{t}} \to \{\mathsf{t}, \mathsf{f}\}$ is recursively defined by

$$
\begin{aligned}
\mathsf{t}^n &= \mathsf{t} \\
\mathsf{f}^n &= \mathsf{f} \\
(\mathsf{sl}(m))^n &= \begin{cases} \mathsf{t} & \text{if } S(n) = m \\ \mathsf{f} & \text{otherwise} \end{cases} \\
(\mathsf{sl}_{>}(m))^n &= \begin{cases} \mathsf{t} & \text{if } S(n) > m \\ \mathsf{f} & \text{otherwise} \end{cases} \\
(\neg\phi)^n &= \neg(\phi^n) \\
(\phi \to \psi)^n &= \phi^n \to \psi^n
\end{aligned}
$$

We will write $[s \xrightarrow{\phi,a} s']_F^n$ for $[s \xrightarrow{\phi,a} s']_F \wedge \phi^n$. The abbreviation $[s \xrightarrow{\phi,\sigma} s']_F^n$ is used analogously.

The definition of process extraction for simple timed frames (Definition 3.8) can be adapted in two ways: one way yielding absolute time processes and another way yielding parametric time processes.

Definition 4.4. Let F be a simple timed frame with time conditional transitions and $s, t \in \mathbb{S}$. Then $s \overset{\mathsf{a}}{\curvearrowright} tF$ is the absolute time process $0 \gg X_s^0$, where X_s^0 is given by the following finite set of process equations:

$$\{X_{s'}^n = P_{s'}^n \mid s' = s \text{ or } ([s \to_{|F| \setminus \{t\}}^* s']_F \text{ and } s' \neq t), n \in \mathbb{N}, P_{s'}^{n-1} \neq \dot{\delta} \text{ if } n > 0\}$$

where

$$P^n_{s'} = \sum_{a \in A} \sum_{\phi \in \Phi_F} \left(\begin{array}{l} [s' \xrightarrow{\phi,a} t]^n_F :\!:\!\rightarrow \mathsf{cts}(a) + \\ [s' \xrightarrow{\phi,\sigma} t]^n_F :\!:\!\rightarrow \mathsf{cts}(\delta) + \\ \sum_{s'' \in |F| \setminus \{t\}} ([s' \xrightarrow{\phi,a} s'']^n_F :\!:\!\rightarrow \mathsf{cts}(a) \cdot X^n_{s''}) + \\ \sum_{s'' \in |F| \setminus \{t\}} ([s' \xrightarrow{\phi,\sigma} s'']^n_F :\!:\!\rightarrow \sigma_{\mathsf{rel}}(X^{n+1}_{s''})) \end{array} \right)$$

Besides, $s \overset{\mathrm{p}}{\curvearrowright} tF$ is the parametric time process $\surd_{\mathsf{d}} k \,.\, (k \gg X^k_s)$, where X^k_s is given by the set of process equations given above.

The restriction $P^{n-1}_{s'} \neq \dot{\delta}$ if $n > 0$ is used to keep the set of process equations finite. The process $P^n_{s'}$ can be viewed as the behaviour subsequent to entering state s' during time slice $n + 1$. Note that $s \overset{\mathrm{a}}{\curvearrowright} tF = 0 \gg (s \overset{\mathrm{p}}{\curvearrowright} tF)$. It is very straightforward to integrate the conditional transitions introduced here with the conditional transitions introduced in Section 4.1, where the truth of the conditions is state dependent instead of time dependent.

It follows directly from the definitions of regular absolute time processes and regular parametric time processes in [3] that Lemma 3.10 goes through for this extension.

References

[1] J.C.M. Baeten and J.A. Bergstra. Process algebra with signals and conditions. In M. Broy, editor, *Programming and Mathematical Methods*, pages 273–323. NATO ASI Series F88, Springer-Verlag, 1992.

[2] J.C.M. Baeten and J.A. Bergstra. Process algebra with propositional signals. Logic Group Preprint Series 123, Utrecht University, Department of Philosophy, November 1994.

[3] J.C.M. Baeten and J.A. Bergstra. Discrete time process algebra. Report P9208c, University of Amsterdam, Programming Research Group, March 1995. To appear in Formal Aspects of Computing.

[4] J.C.M. Baeten and J.A. Bergstra. Discrete time process algebra with abstraction. In H. Reichel, editor, *Fundamentals of Computation Theory*, pages 1–15. LNCS 965, Springer-Verlag, 1995.

[5] J.C.M. Baeten and W.P. Weijland. *Process Algebra*. Cambridge Tracts in Theoretical Computer Science 18, Cambridge University Press, 1990.

[6] J.F.A.K. van Benthem and J.A. Bergstra. Logic of transition systems. *Journal of Logic, Language and Information*, 3:247–283, 1995.

[7] J.F.A.K. van Benthem, D.J.N. van Eijck, and V. Stebletsova. Modal logic, transition systems and processes. *Journal of Logic and Computation*, 4:811–855, 1994.

[8] J.A. Bergstra and J.W. Klop. Process algebra for synchronous communication. *Information and Control*, 60:109–137, 1984.

[9] J.A. Bergstra and C.A. Middelburg. Process algebra semantics of φSDL. Logic Group Preprint Series 129, Utrecht University, Department of Philosophy, March 1995.

[10] J.A. Bergstra and A. Ponse. Frame algebra with synchronous communication. In R.J. Wieringa and R.B. Feenstra, editors, *Information Systems – Correctness and Reusability*, pages 3–15. World Scientific, 1995.

[11] J.A. Bergstra and A. Ponse. Frame-based process logics. In A. Ponse, M. de Rijke, and Y. Venema, editors, *Modal Logic and Process Algebra*, pages 39–64. CSLI Lecture Notes 53, CSLI Publications, 1995.

[12] M. Hennessy and R. Milner. Algebraic laws for non-determinism and concurrency. *Journal of the ACM*, 32:137–161, 1985.

[13] S. Mauw. Example specifications in φSDL. Unpublished manuscript, 1995.

[14] X. Nicollin and J. Sifakis. The algebra of timed processes ATP: Theory and application. *Information and Computation*, 114:131–178, 1994.

[15] M.A. Reniers and J.J. Vereijken. Completeness in discrete time process algebra, 1995. Draft.

[16] C. Stirling. Modal and temporal logics. In S. Abramsky, D. Gabbay, and T.S.E. Maibaum, editors, *Handbook of Logic and Computer Science, Volume 2*. Clarendon Press, 1992.

[17] Specification and description language (SDL). ITU-T Recommendation Z.100, Revision 1, 1994.

Appendix

In this appendix, we introduce conditional timed frames. They do not introduce new kinds of states or transitions; they are just a means to describe frames conditionally.

The signature extension for *conditional* timed frames is as follows:

Sorts:

$\mathbb{C}$	*conditions*;
$\mathbb{F}_{\mathsf{t}}^{\mathsf{c}}$	*conditional timed frames*;

Constants & Functions:

$c \quad : \mathbb{C}$	for each $c \in \mathbb{C}_{at}$;
$\mathsf{t} \quad : \mathbb{C}$	true;
$\mathsf{f} \quad : \mathbb{C}$	false;
$\neg \quad : \mathbb{C} \to \mathbb{C}$	negation;
$\to \quad : \mathbb{C}^2 \to \mathbb{C}$	implication;
$\lhd \cdot \rhd : \mathbb{P} \times {\mathbb{F}_{\mathsf{t}}^{\mathsf{c}}}^2 \to \mathbb{F}_{\mathsf{t}}^{\mathsf{c}}$	two-armed *conditional*.

The additional axioms for conditional timed frames are given in Table 9. The

(FCA1)	$X \lhd \mathsf{t} \rhd Y$	$=$	X
(FCA2)	$X \lhd (\neg\phi) \rhd Y$	$=$	$Y \lhd \phi \rhd X$
(FCA3)	$X \lhd (\phi \wedge \psi) \rhd Y$	$=$	$(X \lhd \phi \rhd Y) \lhd \psi \rhd Y$
(FCA4)	$X \lhd \phi \rhd Y$	$=$	$(X \lhd \phi \rhd \emptyset) \oplus (\emptyset \lhd \phi \rhd Y)$
(FCA5)	$(X \oplus Y) \lhd \phi \rhd Z$	$=$	$(X \lhd \phi \rhd Z) \oplus (Y \lhd \phi \rhd Z)$
(FCA6)	$(\phi \curvearrowright s) \lhd \psi \rhd \emptyset$	$=$	$(\phi \wedge \psi) \curvearrowright s$
(FCA7)	$(s \xrightarrow{\phi, a} s') \lhd \psi \rhd \emptyset$	$=$	$s \xrightarrow{\phi \wedge \psi, a} s'$
(FCA8)	$(s \xrightarrow{\phi, \sigma} s') \lhd \psi \rhd \emptyset$	$=$	$s \xrightarrow{\phi \wedge \psi, \sigma} s'$

Table 9: Additional axioms for conditional timed frames.

following are some equations derivable from the axioms (FCA1)–(FCA5):

$$X \lhd \phi \rhd X = X$$

$$(X \lhd \phi \rhd Y) \oplus Z = (X \oplus Z) \lhd \phi \rhd (Y \oplus Z)$$

$$(X \lhd \phi \rhd Y) \lhd \psi \rhd Z = (X \lhd \psi \rhd Z) \lhd \phi \rhd (Y \lhd \psi \rhd Z)$$

$$(X \lhd \phi \rhd Y) \lhd \phi \rhd Z = X \lhd \phi \rhd Z$$