



Centrum voor Wiskunde en Informatica

REPORTRAPPORT

Completeness of Timed muCRL

J.F. Groote, M.A. Reniers, J.J. van Wamel, M.B. van der Zwaag

Software Engineering (SEN)

SEN-R0034 November 30, 2000

Report SEN-R0034
ISSN 1386-369X

CWI
P.O. Box 94079
1090 GB Amsterdam
The Netherlands

CWI is the National Research Institute for Mathematics and Computer Science. CWI is part of the Stichting Mathematisch Centrum (SMC), the Dutch foundation for promotion of mathematics and computer science and their applications.

SMC is sponsored by the Netherlands Organization for Scientific Research (NWO). CWI is a member of ERCIM, the European Research Consortium for Informatics and Mathematics.

Copyright © Stichting Mathematisch Centrum
P.O. Box 94079, 1090 GB Amsterdam (NL)
Kruislaan 413, 1098 SJ Amsterdam (NL)
Telephone +31 20 592 9333
Telefax +31 20 592 4199

Completeness of Timed μ CRL *

J.F. Groote¹ M.A. Reniers¹ J. van Wamel² M.B. van der Zwaag²

{J.F.Groote@tue.nl, M.A.Reniers@tue.nl, Jos.van.Wamel@cw.nl, Mark.van.der.Zwaag@cw.nl}

¹*Section Technical Applications
Eindhoven University of Technology
Eindhoven, The Netherlands*

²*Cluster Software Engineering, CWI
Amsterdam, The Netherlands*

ABSTRACT

In [9], a straightforward extension of the process algebra μ CRL was proposed to explicitly deal with time. The process algebra μ CRL has been designed especially to deal with data in a process algebraic context. Using the features for data, only a minor extension of the language was needed to obtain a very expressive variant of time. But [9] contains syntax, operational semantics and axioms characterising timed μ CRL. It did not contain an in-depth analysis of timed μ CRL. This paper fills this gap by providing soundness and completeness results. The main tool to establish these is a mapping of timed to untimed μ CRL and employing the completeness results obtained for untimed μ CRL.

2000 Mathematics Subject Classification: 68Q45; 68Q60; 68Q70; 68Q85

Keywords & Phrases: Real-time, Process Algebra, Completeness, μ CRL

Note: Work carried out under project SEN2.1 "Process Specification and Analysis".

1 Introduction

Process algebras are very nice tools to study fundamental concepts such as actions and interactions, non determinism and parallelism, behavioural equivalences and internal or hidden actions. In their plain form (e.g. CCS [24], CSP [19] or ACP [5]), these languages are not very expressive in the sense that only very simple protocols and distributed systems can properly be described. This is the reason why these plain process algebras have been extended with data ([17, 12]). The most expressive [10] and by far the most developed is μ CRL [12], being the process algebra ACP extended with equational abstract datatypes. The process algebra μ CRL is the basis of several novel proof methodologies [14, 13] and numerous verifications (see [11]), which for a large part have been computer checked. Also, it has been the basis for several fundamental studies about for instance expressiveness and decidability [10], and visualisation of huge state spaces [27]. Finally, a toolset around μ CRL has been constructed which is not based on finite state spaces, but on linear process operators [6], allowing automatic treatment of processes with huge or infinite state spaces.

As has widely been recognised, time is an important feature for interacting systems, and henceforth many timed process algebras have been developed. In order to use the advantages that μ CRL offers in a timed setting, timed μ CRL has been defined [9]. In essence, it only consists of the addition of a single operator p^t , for p a process and t a value indicating a moment in time. Using the conditional operator $(_ \triangleleft _ \triangleright _)$ and the sum (\sum) construct, we could express every timing aspect of systems

*The authors thank all persons who assisted in developing μ CRL throughout the years. For timed μ CRL they have benefited a lot from comments from Jos Baeten, Jan Bergstra, Bas Luttik, Radu Mateescu, Alban Ponse, Jan Springintveld, Jan-Joris Vereijken and Tim Willemsse.

that we encountered, including time-outs, deadlines and intervals. Somewhat surprisingly, this timed extension even turned out to be able to express and analyse hybrid systems [15], in which continuous signals and discrete events are combined.

The definition of timed μCRL has been guided by simplicity, elegance and its suitability to adapt many of the existing proof technologies to the timed setting. In [9] the language was explained, syntax and operational semantics were given and an equational characterisation of the language was provided. There were no completeness results in the paper and a claim of soundness turned out to be not completely justified. This paper fills these gaps by providing soundness and completeness theorems, relative to some completeness assumptions on the datatypes. For simplicity and generality, the datatypes are assumed to be given as algebras satisfying a number of elementary properties, contrary to the definition in [9], where equational, inductive datatypes were employed.

The proof of completeness follows the approach of [26], where a timed μCRL is translated to un-timed μCRL , in such a way that the completeness results of [10] can directly be applied to achieve completeness. Besides providing a completeness result in a relatively easy way, there is an additional advantage of this approach, which may turn out to be very helpful in the analysis of timed systems. As mentioned above, there are many analysis techniques and a very capable tool for un-timed μCRL . Transferring these to the timed setting may be a costly and hard operation. However, if we translate timed μCRL expressions to un-timed μCRL , everything achieved for un-timed μCRL carries over automatically. We have especially high hopes for the toolset, as the translation can be done automatically in this case.

Also, we hope that this approach may help to solve one of the longer standing open problems in timed process algebra, namely a complete axiomatisation of branching or weak bisimulation in a setting as expressive as timed μCRL . There are numerous completeness results for weak bisimulations in timed settings with restricted expressivity (e.g. [8]). The only attempt that we know where the expressiveness of the language comes close to that of μCRL can be found in [21]. Here, an axiom for branching bisimulation is given, which is claimed to be sound and complete. Although we do not doubt soundness, the completeness claim has not been appropriately justified, and turns out not to be easily reconstructable.

Timed process algebras have received a lot of attention, and many different formalisms have been proposed. However, very few of these include data (see for an overview [28, 4]). The few that do so, have a very different focus than μCRL .

In the first place, there are timed automata [1] which have been extended to hybrid automata [23]. The typical difference with timed μCRL is that timed automata are a semantical concept, much less algebraical, and as such there is no set of characteristic axioms or even a precisely defined set of operators to construct these. Timed automata are in a sense just finite state machines extended with clocks. This formalism sparked off a lot of research for instance in timed model checking and equivalence checking (using simulation relations). Using a technique known as convex regions, it is possible to capture the infinite flow of time into a finite automaton. This result has been the basis for various toolsets, such as HyTech [18], Uppaal [22] and Kronos [29], of which Uppaal allows simple datatypes as natural numbers and lists.

At a different end of the spectrum, there are the timed specification languages of which extended LOTOS [7] and LOTOS NT [25] are among the most ambitious, although older languages such as SDL [20] also combine time within an expressive context. The goal of these languages is to allow to express distributed timed systems as elegantly as possible. As such, the languages are rich in syntax and much of the effort around these languages is in building tools around it. The difference with timed μCRL is that these languages are not apt, and not intended for more fundamental research.

2 The axiom system $p\text{CRL}_t$

The axiom system $p\text{CRL}_t$ for *pico* CRL with time is presented. It serves as the basic framework for our studies. We work in a setting without the silent step τ , and without abstraction or general operators for renaming. The addition of operators for parallelism in Section 5 leads to full μCRL_t . We define a notion of *basic terms* and prove that all terms over the signature $\Sigma(p\text{CRL}_t)$ without process variables are derivably equal to basic terms.

2.1 Abstract data types

The processes described in the language μCRL generally exchange data. For the specification of data we use (equational) abstract data types with an explicit distinction between constructor functions and ‘normal’ functions. Moreover, all properties of a data type must be explicitly declared, which makes it clear which assumptions can be used for proving properties of data or processes.

In this paper, we do not treat the syntactical details of the data language in depth. We simply assume the existence of a data signature. A data signature consists of a set S of sort symbols and a set F of function declarations. We assume disjoint infinite sets of variables V_s for the sort symbols $s \in S$. Let $V = \bigcup_{s \in S} V_s$. The set of terms of sort s is denoted by T_s . Furthermore, we assume the existence of a sort symbol B for the booleans with function declarations \mathbf{t} and \mathbf{f} , and the usual connectives \neg , \wedge , and \vee . For each sort symbol s , we assume a data algebra with universe \mathcal{D}_s . The set \mathcal{D}_B has two elements: the interpretation of \mathbf{t} and the interpretation of \mathbf{f} .

Finally, we assume the existence of a sort symbol T of time elements and we require that it is totally ordered and that it contains a least element. The total order is denoted \leq and the least element is denoted $\mathbf{0}$. Throughout this paper we abbreviate terms $\neg(u \leq t)$ to $t < u$. A term of the form $t[e/d]$ denotes the term t with variable d replaced by term e .

Besides the above requirements one is free to specify the kind of time domain one requires. This is done to accommodate those preferring discrete time, and others who prefer a notion of dense time. One can for instance define that \mathcal{D}_T has only a finite number of elements, or one can define an ordinal-like structure on it.

2.2 The syntax of $p\text{CRL}_t$

The signature of the theory $p\text{CRL}_t$ consists of a data signature and a process signature. The process signature consists of the sort symbol P and the following function declarations:

1. *action declarations* $\mathbf{a} : s_1 \times \cdots \times s_n \rightarrow P$ for sort symbols s_i ($i = 0, \dots, n$) from the data signature.
2. *deadlock* $\delta : \rightarrow P$. Timed μCRL contains a constant δ , which can be used to express that from now on, no action can be performed any more. It models inaction, for instance in the case where a number of computers are waiting for each other, and the whole system is blocked.
3. *alternative composition* $_ + _ : P \times P \rightarrow P$. The process represented by $p + q$ behaves like p or q , depending on which of the two performs the first action.
4. *sequential composition* $_ \cdot _ : P \times P \rightarrow P$. The process represented by $p \cdot q$ first performs the actions of p , until p terminates, and then continues with the actions in q .
5. *conditional operator* $_ \triangleleft _ \triangleright _ : P \times B \times P \rightarrow P$. The “ $_ \triangleleft _ \triangleright _$ ”-operator is the conditional operator of μCRL , and it operates precisely as a *then_if_else*-construct. The process term $p \triangleleft b \triangleright q$ behaves like p if b is equal to \mathbf{t} , and if b is equal to \mathbf{f} it behaves like q .

6. *alternative quantification* $\sum_v _ : P \rightarrow P$ for each data variable $v \in V$. The sum operator $\sum_v p$ behaves like $p[d_1/v] + p[d_2/v] + \dots$, i.e., as the possibly infinite choice between $p[d_i/v]$ for any data term d_i of the sort of v .
7. *at-operator* $_ _ : P \times T \rightarrow P$. A key feature of timed μCRL is that it can be expressed at which time certain actions must take place. This is done using the “at”- operator. The process $p _ t$, behaves like the process p , with the restriction that the first action of p must take place at time t .
8. *initialisation operator* $_ \gg _ : T \times P \rightarrow P$. The process $t \gg p$ behaves as the process p as if it was started at time t . Thus the initialisation operator can be used to restrict the behaviour of a process to the part that can idle until the specified moment of time.

In the sequel, we will call these function declarations operators. For action declarations we usually write $\mathbf{a} : s_1 \times \dots \times s_n$ instead of $\mathbf{a} : s_1 \times \dots \times s_n \rightarrow P$. The set of all action declarations is denoted Act . *Action terms* are terms of the form $\mathbf{a}(d_1, \dots, d_n)$, where $\mathbf{a} : s_1 \times \dots \times s_n \in Act$ and d_i a data term of sort s_i (for $i = 0, \dots, n$); the set of all action terms is denoted AT . We write AT_δ for $AT \cup \{\delta\}$.

Furthermore, we assume the existence of an infinite set of process variables V_P which is disjoint from the set of data variables V .

Process terms are built from action terms, data terms, variables and process operators. The set of all process terms is denoted T_P . For all operators in the language, characterising axioms are provided following the process algebraic tradition, in order to define which equivalences hold between processes. In fact, axiomatic reasoning with processes forms the essence of μCRL . Examples of equational manipulations with processes are given throughout this paper. Whereas the axioms provide a more syntactical perspective, operational semantics are used to interpret process terms in terms of potential behaviours of a system (see Section 3.2).

We list the various operators of $p\text{CRL}_t$ in decreasing binding strength:

$$_ _ , _ _ , _ \gg _ , _ \triangleleft _ , _ \triangleright _ , _ \sum _ , _ + _ .$$

2.3 The theory $p\text{CRL}_t$

The axioms of $p\text{CRL}_t$ are the axioms given by the user for the data types such as the booleans B and the time domain T , and the axioms given in Table 2. As proof theory we use generalised equational logic with a congruence rule for binders. For a precise expository on this proof theory we refer to [10]. In Table 1, we present the proof theory for $p\text{CRL}_t$. Here, E represents the set of axioms of $p\text{CRL}_t$ and Σ is the signature of the theory $p\text{CRL}_t$.

Process-closed terms are terms without process variables, but possibly with bound and free data variables.

The two elementary operators to construct processes are the sequential composition operator, written as \cdot and the alternative composition operator, written as $+$. The axioms A1–A5 in Table 2 describe the elementary properties of the sequential and alternative composition operators. For instance, the axioms A1, A2 and A3 express that $+$ is commutative, associative and idempotent. These laws motivate why in some cases parentheses may be omitted. For instance, it is allowed to write $a + b + c$, because, due to A2, it does not matter how brackets are put. The process algebra that consists of atomic actions, alternative and sequential composition only and for which A1–A5 are the only axioms is usually called *Basic Process Algebra* [5].

In the calculations in this paper we work modulo associativity and commutativity of $+$, and we do not mention the use of simple properties of the functions on data such as \neg , \vee , \wedge , \leq , and eq .

In Table 2, $p\text{CRL}_t$ axioms for the sum operator are listed. The sum operator is a difficult operator, because it acts as a binder. This introduces a range of problems with substitutions. When we substitute a process p for a variable x in a term q , then no free variable in p may become bound.

$t = t'$	for every $t = t' \in E$
$\frac{t = t'}{t[e/v] = t'[e/v]}$	for every $v \in V_s$ and $e \in T_s$
$\frac{t_1 = t'_1 \cdots t_n = t'_n}{F(t_1, \dots, t_n) = F(t'_1, \dots, t'_n)}$	for every $F : s_1 \times \cdots \times s_n \rightarrow s' \in \Sigma$
$\frac{t = t'}{\sum_v t = \sum_v t'}$	
$t = t$	$\frac{t = t'}{t' = t}$
	$\frac{t_1 = t_2 \quad t_2 = t_3}{t_1 = t_3}$

Table 1: Generalised equational logic for $p\text{CRL}_t$.

If this appears to happen, we must first rename all bound occurrences of that variable in q into a variable that does not occur free in p . This renaming is called α -conversion. We consider processes modulo α -conversion, so the terms $\sum_v p$ and $\sum_w p[w/v]$ are equal if w does not occur freely in v . Consequently, we may only substitute the action $\mathbf{a}(v)$ for x in the left hand side of axiom SUM1, after renaming the bound variable v into w . So, SUM1 is a concise way of saying that if v does not appear in p , then we may omit the sum operator in $\sum_v p$.

As another example, consider axiom SUM4. It says that we may distribute the sum operator over an alternative composition operator.

The process $p \cdot t$ behaves like the process p , with the restriction that the first action of p must take place at time t . So, if we assume that the natural numbers denote moments in time, the process described by $a \cdot 1 \cdot b \cdot 2 \cdot c \cdot 3$ specifies that the actions a , b and c must take place at times 1, 2 and 3, respectively.

2.4 Time deadlocks

If an action happens at time t , then a subsequent action can take place at time t or afterwards. This means that in $a \cdot 2 \cdot b \cdot 3$ and $a \cdot 2 \cdot b \cdot 2$ the action b can happen, and in $a \cdot 2 \cdot b \cdot 1$ the action b is blocked.

Actually, the last example above is equivalent to $a \cdot 2 \cdot \delta \cdot 2$, which says that after action a we have a deadlock at time 2. In order to let b take place as prescribed, we have to reverse time. As this is clearly in conflict with reality, we choose to stop time at time 2. A process $\delta \cdot t$ is called a *time deadlock*.

Whenever a specification prescribes timing behaviour that cannot be realised, it will exhibit a time deadlock, i.e., time is stopped at a certain point. Specifications with time inconsistencies can clearly not be implemented.

An important property of δ is $\delta \cdot p = \delta$. It says that the process p after the deadlock δ cannot be executed. It is formulated in Table 2 as axiom A7. The constant δ is the ‘classical’ deadlock of process algebra, which cannot execute actions but which does not stop time. A time deadlock $\delta \cdot t$ exists at any moment before t and at t . The deadlock process δ exists at any time; we can derive $\delta = \sum_v \delta \cdot v$. As a consequence, we can not in general have $p + \delta = p$ – an identity that is, in untimed $p\text{CRL}$, valid for every p . Consider for example the process $a \cdot 2 + \delta$. This process can perform the action a at time 2 and then terminates, but it can also let time pass until after time 2. So, contrary to untimed $p\text{CRL}$, δ is not a neutral element for alternative composition in $p\text{CRL}_t$. This role is taken over by the process

A1	$x + y = y + x$	SUM1	$\sum_v x = x$
A2	$x + (y + z) = (x + y) + z$	SUM3	$\sum_v p = \sum_v p + p$
A3	$x + x = x$	SUM4	$\sum_v (p + q) = \sum_v p + \sum_v q$
A4	$(x + y) \cdot z = x \cdot z + y \cdot z$	SUM5	$(\sum_v p) \cdot x = \sum_v (p \cdot x)$
A5	$(x \cdot y) \cdot z = x \cdot (y \cdot z)$	SUM12'	$(\sum_v p) \triangleleft b \triangleright \delta \cdot \mathbf{0} = \sum_v p \triangleleft b \triangleright \delta \cdot \mathbf{0}$
A6 ⁻	$a + \delta = a$	PE	$p \triangleleft eq(v, w) \triangleright \delta \cdot \mathbf{0} = p[v/w] \triangleleft eq(v, w) \triangleright \delta \cdot \mathbf{0}$
A6'	$x + \delta \cdot \mathbf{0} = x$		
A7	$\delta \cdot x = \delta$		
C1	$x \triangleleft \mathbf{t} \triangleright y = x$		
C2	$x \triangleleft \mathbf{f} \triangleright y = y$		
C3	$x \triangleleft b \triangleright y = x \triangleleft b \triangleright \delta \cdot \mathbf{0} + y \triangleleft \neg b \triangleright \delta \cdot \mathbf{0}$		
C4	$(x \triangleleft b_1 \triangleright y) \triangleleft b_2 \triangleright y = x \triangleleft (b_1 \wedge b_2) \triangleright y$		
C5	$x \triangleleft b_1 \triangleright \delta \cdot \mathbf{0} + x \triangleleft b_2 \triangleright \delta \cdot \mathbf{0} = x \triangleleft (b_1 \vee b_2) \triangleright \delta \cdot \mathbf{0}$		
C6	$(x \triangleleft b \triangleright y) \cdot z = x \cdot z \triangleleft b \triangleright y \cdot z$		
C7	$(x + y) \triangleleft b \triangleright z = x \triangleleft b \triangleright z + y \triangleleft b \triangleright z$		
SCA	$(x \triangleleft b \triangleright \delta \cdot \mathbf{0}) \cdot (y \triangleleft b \triangleright \delta \cdot \mathbf{0}) = x \cdot y \triangleleft b \triangleright \delta \cdot \mathbf{0}$		
AT1	$x = \sum_t x^t$		
AT2	$a^t \cdot x = a^t \cdot (t \gg x)$		
ATA1	$a^t u = (a^t \triangleleft u \leq t \triangleright \delta^t) \triangleleft t \leq u \triangleright \delta^t u$		
ATA2	$(x + y)^t = x^t + y^t$		
ATA3	$(x \cdot y)^t = x^t \cdot y$		
ATA4	$(\sum_v p)^t = \sum_v p^t$		
ATA5	$(x \triangleleft b \triangleright y)^t = x^t \triangleleft b \triangleright y^t$		
ATB1	$t \gg a^t u = a^t u \triangleleft t \leq u \triangleright \delta^t$		
ATB2	$t \gg (x + y) = t \gg x + t \gg y$		
ATB3	$t \gg (x \cdot y) = (t \gg x) \cdot y$		
ATB4	$t \gg \sum_v p = \sum_v t \gg p$		
ATB5	$t \gg (x \triangleleft b \triangleright y) = (t \gg x) \triangleleft b \triangleright (t \gg y)$		

Table 2: Axioms of $pCRL_t$: $x, y, z \in V_P$, process-closed $p, q \in T_P$, $a \in AT_\delta$, $t, u \in V_T$, $b, b_1, b_2 \in V_B$ and $v, w \in V$.

$\delta \cdot \mathbf{0}$. This can easily be seen, as every $p\text{CRL}_t$ process exists at time $\mathbf{0}$. In the axiomatisation, we see that the $p\text{CRL}$ axiom A6 ($x + \delta = x$) is absent, and has been replaced by axiom A6'.

In $p\text{CRL}_t$, we have by axiom AT1 that $\delta = \sum_t \delta \cdot t$, so δ models the process that will never do a step, terminate or exhibit a time deadlock. Therefore, a more appropriate name for δ would be *livelock*. We see that if a time deadlock $\delta \cdot u$ occurs in a process term with δ as an alternative, it vanishes:

$$\delta + \delta \cdot t \stackrel{\text{AT1}}{=} \sum_t \delta \cdot t + \delta \cdot t \stackrel{\text{SUM3}}{=} \sum_t \delta \cdot t \stackrel{\text{AT1}}{=} \delta.$$

For processes p that do not refer to time explicitly, we still like to have the identity $p + \delta = p$. Using axiom A6⁻ this identity can be derived for untimed process-closed terms p .

2.5 Simple identities

Calculating in (timed) μCRL generally requires numerous basic identities. In this section, we present some identities as lemmas for easy future reference. In derivations in this section and the following sections, we frequently apply the axioms for booleans and for the time domain. We will do so without explicitly mentioning them.

Axiom SUM3 is typically used to extract a specific alternative from an alternative quantification where the bound variable, say v , is replaced by a data term, say d , of the appropriate sort. The following lemma ensures the correctness of such an extraction.

Lemma 2.1 Let s be an arbitrary sort. For process-closed $p \in T_P$, $v \in V_s$, and $d \in T_s$, it holds that

$$\sum_v p = \sum_v p + p[d/v].$$

Proof. Consider the following derivation:

$$\sum_v p = \left(\sum_v p \right) [d/v] \stackrel{\text{SUM3}}{=} \left(\sum_v p + p \right) [d/v] = \left(\sum_v p \right) [d/v] + p[d/v] = \sum_v p + p[d/v].$$

Lemma 2.2 For $x \in V_P$ and $b \in V_B$, it holds that $x \triangleleft b \triangleright x = x$.

Proof. $x \triangleleft b \triangleright x \stackrel{\text{C3}}{=} x \triangleleft b \triangleright \delta \cdot \mathbf{0} + x \triangleleft \neg b \triangleright \delta \cdot \mathbf{0} \stackrel{\text{C5}}{=} x \triangleleft t \triangleright \delta \cdot \mathbf{0} \stackrel{\text{C1}}{=} x$.

Lemma 2.3 For $t \in V_T$, it holds that $\delta \cdot \mathbf{0} \cdot t = \delta \cdot \mathbf{0}$.

Proof. $\delta \cdot \mathbf{0} \cdot t \stackrel{\text{ATA1}}{=} (\delta \cdot \mathbf{0} \triangleleft t \leq \mathbf{0} \triangleright \delta \cdot \mathbf{0}) \triangleleft \mathbf{0} \leq t \triangleright \delta \cdot t \stackrel{2.2}{=} \delta \cdot \mathbf{0} \triangleleft \mathbf{0} \leq t \triangleright \delta \cdot \mathbf{0} \stackrel{\text{C1}}{=} \delta \cdot \mathbf{0}$.

In the following lemma we present some facts about the initialisation operator.

Lemma 2.4 For $t, u \in V_T$ and $a \in AT_\delta$, it holds that

1. $t \gg (t \gg a \cdot u) = t \gg a \cdot u$;
2. $t \gg \delta \cdot \mathbf{0} = \delta \cdot t$.

Proof.

$$1. t \gg (t \gg a \cdot u) \stackrel{\text{ATB1}}{=} t \gg (a \cdot u \triangleleft t \leq u \triangleright \delta \cdot t) \stackrel{\text{ATB5}}{=} (t \gg a \cdot u) \triangleleft t \leq u \triangleright (t \gg \delta \cdot t) \stackrel{\text{ATB1}}{=} (a \cdot u \triangleleft t \leq u \triangleright \delta \cdot t) \triangleleft t \leq u \triangleright \delta \cdot t \stackrel{\text{C4}}{=} a \cdot u \triangleleft t \leq u \triangleright \delta \cdot t \stackrel{\text{ATB1}}{=} t \gg a \cdot u.$$

$$2. t \gg \delta \cdot \mathbf{0} \stackrel{\text{ATB1}}{=} \delta \cdot \mathbf{0} \triangleleft t \leq \mathbf{0} \triangleright \delta \cdot t = \delta \cdot \mathbf{0} \triangleleft eq(t, \mathbf{0}) \triangleright \delta \cdot t \stackrel{\text{PE}}{=} \delta \cdot t \triangleleft eq(t, \mathbf{0}) \triangleright \delta \cdot t \stackrel{2.2}{=} \delta \cdot t.$$

Lemma 2.5 For $v \in V_T$ and $t \in V_T$, it holds that $\sum_v \delta \cdot v \triangleleft v \leq t \triangleright \delta \cdot \mathbf{0} = \delta \cdot t$.

Proof.

$$\begin{aligned}
& \sum_v \delta \cdot v \triangleleft v \leq t \triangleright \delta \cdot \mathbf{0} \\
& \stackrel{2.1}{=} \sum_v \delta \cdot v \triangleleft v \leq t \triangleright \delta \cdot \mathbf{0} + (\delta \cdot v \triangleleft v \leq t \triangleright \delta \cdot \mathbf{0})[t/v] \\
& = \sum_v \delta \cdot v \triangleleft v \leq t \triangleright \delta \cdot \mathbf{0} + \delta \cdot t \triangleleft t \leq t \triangleright \delta \cdot \mathbf{0} \\
& \stackrel{\text{C1}}{=} \sum_v \delta \cdot v \triangleleft v \leq t \triangleright \delta \cdot \mathbf{0} + \delta \cdot t \\
& \stackrel{\text{AT1}}{=} \sum_v \delta \cdot t \cdot v + \sum_v \delta \cdot v \triangleleft v \leq t \triangleright \delta \cdot \mathbf{0} \\
& \stackrel{\text{ATA1,2.2}}{=} \sum_v \delta \cdot t \triangleleft t \leq v \triangleright \delta \cdot v + \sum_v \delta \cdot v \triangleleft v \leq t \triangleright \delta \cdot \mathbf{0} \\
& \stackrel{\text{C3}}{=} \sum_v (\delta \cdot t \triangleleft t \leq v \triangleright \delta \cdot \mathbf{0} + \delta \cdot v \triangleleft t > v \triangleright \delta \cdot \mathbf{0}) + \sum_v \delta \cdot v \triangleleft v \leq t \triangleright \delta \cdot \mathbf{0} \\
& \stackrel{\text{C5}}{=} \sum_v (\delta \cdot t \triangleleft t \leq v \triangleright \delta \cdot \mathbf{0} + \delta \cdot v \triangleleft t > v \triangleright \delta \cdot \mathbf{0}) \\
& + \sum_v (\delta \cdot v \triangleleft eq(t, v) \triangleright \delta \cdot \mathbf{0} + \delta \cdot v \triangleleft t > v \triangleright \delta \cdot \mathbf{0}) \\
& \stackrel{\text{C5,PE}}{=} \sum_v (\delta \cdot t \triangleleft t < v \triangleright \delta \cdot \mathbf{0} + \delta \cdot t \triangleleft eq(t, v) \triangleright \delta \cdot \mathbf{0} + \delta \cdot v \triangleleft t > v \triangleright \delta \cdot \mathbf{0}) \\
& + \sum_v (\delta \cdot t \triangleleft eq(t, v) \triangleright \delta \cdot \mathbf{0} + \delta \cdot v \triangleleft t > v \triangleright \delta \cdot \mathbf{0}) \\
& \stackrel{\text{SUM4,A3}}{=} \sum_v (\delta \cdot t \triangleleft t < v \triangleright \delta \cdot \mathbf{0} + \delta \cdot t \triangleleft eq(t, v) \triangleright \delta \cdot \mathbf{0} + \delta \cdot v \triangleleft t > v \triangleright \delta \cdot \mathbf{0}) \\
& \stackrel{\text{C5}}{=} \sum_v (\delta \cdot t \triangleleft t \leq v \triangleright \delta \cdot \mathbf{0} + \delta \cdot v \triangleleft t > v \triangleright \delta \cdot \mathbf{0}) \\
& \stackrel{\text{C3}}{=} \sum_v \delta \cdot t \triangleleft t \leq v \triangleright \delta \cdot v \\
& \stackrel{\text{ATA1,2.2}}{=} \sum_v \delta \cdot t \cdot v \\
& \stackrel{\text{AT1}}{=} \delta \cdot t.
\end{aligned}$$

2.6 Basic terms

We provide a basic syntactic format for $p\text{CRL}_t$ -terms. In this format we use the notation $\sum_{\vec{v}}$ to represent a finite sequence of alternative quantifications $\sum_{v_1} \sum_{v_2} \cdots \sum_{v_n}$ ($n \geq 0$).

Definition 2.1 (Basic terms) The set of basic terms is inductively defined as follows:

1. $\sum_{\vec{v}} a \cdot t \triangleleft b \triangleright \delta \cdot \mathbf{0}$, with $a \in AT_\delta$, t a time term and b a boolean term, is a basic term;
2. if p is a basic term, then $\sum_{\vec{v}} a \cdot t \cdot p \triangleleft b \triangleright \delta \cdot \mathbf{0}$, with $a \in AT$, t a time term and b a boolean term, is a basic term;
3. if p and q are basic terms, then $p + q$ is a basic term.

If a basic term is of the first form, then we say it is of *type 1*. Similarly for forms 2 and 3.

Theorem 2.1 (Basic Term Theorem) If q is a process-closed term over $\Sigma(p\text{CRL}_t)$, then there is a basic term p such that $\mu\text{CRL}_t \vdash q = p$.

Proof. We apply induction on the number of symbols in process-closed term q . The following cases can be distinguished based on the structure of process-closed term q .

1. $q \equiv a$ for some $a \in AT_\delta$. Obviously $a \stackrel{\text{AT1}}{=} \sum_t a \cdot t \stackrel{\text{C1}}{=} \sum_t a \cdot t \triangleleft t \triangleright \delta \cdot \mathbf{0}$. This is a type 1 basic term.
2. $q \equiv q_1 + q_2$ for some process-closed terms q_1 and q_2 . By induction we have the existence of basic terms p_1 and p_2 such that $q_1 = p_1$ and $q_2 = p_2$. Hence, $q = q_1 + q_2 = p_1 + p_2$, which is a type 3 basic term.

3. $q \equiv q_1 \cdot q_2$ for some process-closed terms q_1 and q_2 . By induction we have the existence of basic terms p_1 and p_2 such that $q_1 = p_1$ and $q_2 = p_2$. By induction on the structure of basic term p_1 we prove that there exists a basic term p such that $p_1 \cdot p_2 = p$.

- (a) $p_1 \equiv \sum_{\bar{v}} a^c t \triangleleft b \triangleright \delta \cdot \mathbf{0}$. Then $p_1 \cdot p_2 = (\sum_{\bar{v}} a^c t \triangleleft b \triangleright \delta \cdot \mathbf{0}) \cdot p_2 \stackrel{\text{SUM5}}{=} \sum_{\bar{v}} (a^c t \triangleleft b \triangleright \delta \cdot \mathbf{0}) \cdot p_2 \stackrel{\text{C6}}{=} \sum_{\bar{v}} a^c t \cdot p_2 \triangleleft b \triangleright \delta \cdot \mathbf{0} \cdot p_2 \stackrel{\text{ATA3,A7}}{=} \sum_{\bar{v}} a^c t \cdot p_2 \triangleleft b \triangleright \delta \cdot \mathbf{0}$, which is a type 2 basic term.
- (b) $p_1 \equiv \sum_{\bar{v}} a^c t \cdot p'_1 \triangleleft b \triangleright \delta \cdot \mathbf{0}$. By induction we have the existence of a basic term p' such that $p'_1 \cdot p_2 = p'$. Then, $p_1 \cdot p_2 = (\sum_{\bar{v}} a^c t \cdot p'_1 \triangleleft b \triangleright \delta \cdot \mathbf{0}) \cdot p_2 \stackrel{\text{SUM5}}{=} \sum_{\bar{v}} (a^c t \cdot p'_1 \triangleleft b \triangleright \delta \cdot \mathbf{0}) \cdot p_2 \stackrel{\text{C6}}{=} \sum_{\bar{v}} (a^c t \cdot p'_1) \cdot p_2 \triangleleft b \triangleright \delta \cdot \mathbf{0} \cdot p_2 \stackrel{\text{A5,ATA3,A7}}{=} \sum_{\bar{v}} a^c t \cdot p' \triangleleft b \triangleright \delta \cdot \mathbf{0}$, which is a type 2 basic term.
- (c) $p_1 \equiv p'_1 + p''_1$ for some basic terms p'_1 and p''_1 . By induction we have the existence of basic terms p' and p'' such that $p'_1 \cdot p_2 = p'$ and $p''_1 \cdot p_2 = p''$. Then $p_1 \cdot p_2 = (p'_1 + p''_1) \cdot p_2 \stackrel{\text{A4}}{=} p'_1 \cdot p_2 + p''_1 \cdot p_2 = p' + p''$, which is a type 3 basic term.
4. $q \equiv \sum_v q'$ for some process-closed term q' . By induction we have the existence of basic term p' such that $q' = p'$. We proceed by induction on the structure of basic term p' . If p' is of type 1 or type 2, then obviously $\sum_v p'$ is a basic term. If p' is of type 3, e.g. $p' = p'_1 + p'_2$ for some basic terms p'_1 and p'_2 , then we have by induction the existence of basic terms p''_1 and p''_2 such that $\sum_v p'_1 = p''_1$ and $\sum_v p'_2 = p''_2$. Then $q = \sum_v q' = \sum_v p' = \sum_v (p'_1 + p'_2) \stackrel{\text{SUM4}}{=} \sum_v p'_1 + \sum_v p'_2 = p''_1 + p''_2$, which is a basic term.
5. $q \equiv q_1 \triangleleft b \triangleright q_2$ for some process-closed terms q_1 and q_2 . By induction we have the existence of basic terms p_1 and p_2 such that $q_1 = p_1$ and $q_2 = p_2$. Then $q = q_1 \triangleleft b \triangleright q_2 = p_1 \triangleleft b \triangleright p_2 \stackrel{\text{C3}}{=} p_1 \triangleleft b \triangleright \delta \cdot \mathbf{0} + p_2 \triangleleft \neg b \triangleright \delta \cdot \mathbf{0}$. We prove that for every basic term r there exists a basic term s such that $r \triangleleft \alpha \triangleright \delta \cdot \mathbf{0} = s$ by induction on the structure of basic term r .

- (a) $r \equiv \sum_{\bar{v}} a^c t \triangleleft b \triangleright \delta \cdot \mathbf{0}$. Then, $r \triangleleft \alpha \triangleright \delta \cdot \mathbf{0} = (\sum_{\bar{v}} a^c t \triangleleft b \triangleright \delta \cdot \mathbf{0}) \triangleleft \alpha \triangleright \delta \cdot \mathbf{0} \stackrel{\text{SUM12}'}{=} \sum_{\bar{v}} (a^c t \triangleleft b \triangleright \delta \cdot \mathbf{0}) \triangleleft \alpha \triangleright \delta \cdot \mathbf{0} \stackrel{\text{C4}}{=} \sum_{\bar{v}} a^c t \triangleleft b \wedge \alpha \triangleright \delta \cdot \mathbf{0}$, which is a basic term.
- (b) $r \equiv \sum_{\bar{v}} a^c t \cdot r' \triangleleft b \triangleright \delta \cdot \mathbf{0}$. Then, $r \triangleleft \alpha \triangleright \delta \cdot \mathbf{0} = (\sum_{\bar{v}} a^c t \cdot r' \triangleleft b \triangleright \delta \cdot \mathbf{0}) \triangleleft \alpha \triangleright \delta \cdot \mathbf{0} \stackrel{\text{SUM12}'}{=} \sum_{\bar{v}} (a^c t \cdot r' \triangleleft b \triangleright \delta \cdot \mathbf{0}) \triangleleft \alpha \triangleright \delta \cdot \mathbf{0} \stackrel{\text{C4}}{=} \sum_{\bar{v}} a^c t \cdot r' \triangleleft b \wedge \alpha \triangleright \delta \cdot \mathbf{0}$, which is a basic term.
- (c) $r \equiv r_1 + r_2$. By induction we have the existence of basic terms s_1 and s_2 such that $r_1 \triangleleft \alpha \triangleright \delta \cdot \mathbf{0} = s_1$ and $r_2 \triangleleft \alpha \triangleright \delta \cdot \mathbf{0} = s_2$. Then $r \triangleleft \alpha \triangleright \delta \cdot \mathbf{0} = (r_1 + r_2) \triangleleft \alpha \triangleright \delta \cdot \mathbf{0} \stackrel{\text{C7}}{=} r_1 \triangleleft \alpha \triangleright \delta \cdot \mathbf{0} + r_2 \triangleleft \alpha \triangleright \delta \cdot \mathbf{0} = s_1 + s_2$, which is a basic term.
6. $q \equiv t \gg q'$ for some process-closed term q' . By induction we have the existence of basic term p' such that $q' = p'$. By induction on the structure of basic term p' we prove the existence of a basic term p'' such that $t \gg p' = p''$.

- (a) $p' \equiv \sum_{\bar{v}} a^c u \triangleleft b \triangleright \delta \cdot \mathbf{0}$. Then,

$$\begin{aligned}
t \gg p' &= t \gg (\sum_{\bar{v}} a^c u \triangleleft b \triangleright \delta \cdot \mathbf{0}) \\
&\stackrel{\text{ATB4}}{=} \sum_{\bar{v}} t \gg (a^c u \triangleleft b \triangleright \delta \cdot \mathbf{0}) \\
&\stackrel{\text{ATB5}}{=} \sum_{\bar{v}} (t \gg a^c u) \triangleleft b \triangleright (t \gg \delta \cdot \mathbf{0}) \\
&\stackrel{\text{C3,SUM4,2.4.2}}{=} \sum_{\bar{v}} (t \gg a^c u) \triangleleft b \triangleright \delta \cdot \mathbf{0} + \sum_{\bar{v}} \delta \cdot t \triangleleft \neg b \triangleright \delta \cdot \mathbf{0}.
\end{aligned}$$

The second summand is a basic term. We continue with the first summand:

$$\begin{aligned}
& \sum_{\bar{v}}(t \gg a^c u) \triangleleft b \triangleright \delta \cdot \mathbf{0} \\
\stackrel{\text{ATB1}}{=} & \sum_{\bar{v}}(a^c u \triangleleft t \leq u \triangleright \delta \cdot t) \triangleleft b \triangleright \delta \cdot \mathbf{0} \\
\stackrel{\text{C3}}{=} & \sum_{\bar{v}}(a^c u \triangleleft t \leq u \triangleright \delta \cdot \mathbf{0} + \delta \cdot t \triangleleft t > u \triangleright \delta \cdot \mathbf{0}) \triangleleft b \triangleright \delta \cdot \mathbf{0} \\
\stackrel{\text{C7,SUM4}}{=} & \sum_{\bar{v}}(a^c u \triangleleft t \leq u \triangleright \delta \cdot \mathbf{0}) \triangleleft b \triangleright \delta \cdot \mathbf{0} + \sum_{\bar{v}}(\delta \cdot t \triangleleft t > u \triangleright \delta \cdot \mathbf{0}) \triangleleft b \triangleright \delta \cdot \mathbf{0} \\
\stackrel{\text{C4}}{=} & \sum_{\bar{v}} a^c u \triangleleft t \leq u \wedge b \triangleright \delta \cdot \mathbf{0} + \sum_{\bar{v}} \delta \cdot t \triangleleft t > u \wedge b \triangleright \delta \cdot \mathbf{0},
\end{aligned}$$

which is a basic term.

- (b) $p' \equiv \sum_{\bar{v}} a^c u \cdot p'_1 \triangleleft b \triangleright \delta \cdot \mathbf{0}$. Similar to the previous case only axiom ATB3 has to be applied just before applying ATB1.
- (c) $p' \equiv p'_1 + p'_2$. By induction we have the existence of basic terms p''_1 and p''_2 such that $t \gg p'_1 = p''_1$ and $t \gg p'_2 = p''_2$. Then, $t \gg p' = t \gg (p'_1 + p'_2) \stackrel{\text{ATB2}}{=} t \gg p''_1 + t \gg p''_2 = p''_1 + p''_2$, which is a basic term.
7. $q \equiv q' t$ for some process-closed term q' . By induction we have the existence of basic term p' such that $q' = p'$. We prove that for every basic term p' there exists a basic term p'' such that $p' t = p''$ by induction on the structure of basic term p' .

- (a) $p' \equiv \sum_{\bar{v}} a^c u \triangleleft b \triangleright \delta \cdot \mathbf{0}$. Then,

$$\begin{aligned}
p' t &= (\sum_{\bar{v}} a^c u \triangleleft b \triangleright \delta \cdot \mathbf{0}) t \\
\stackrel{\text{ATA4}}{=} & \sum_{\bar{v}} (a^c u \triangleleft b \triangleright \delta \cdot \mathbf{0}) t \\
\stackrel{\text{ATA5}}{=} & \sum_{\bar{v}} a^c u t \triangleleft b \triangleright \delta \cdot \mathbf{0} t \\
\stackrel{\text{ATA1,2,3}}{=} & \sum_{\bar{v}} ((a^c u \triangleleft t \leq u \triangleright \delta \cdot u) \triangleleft u \leq t \triangleright \delta \cdot t) \triangleleft b \triangleright \delta \cdot \mathbf{0} \\
\stackrel{\text{C3}}{=} & \sum_{\bar{v}} ((a^c u \triangleleft t \leq u \triangleright \delta \cdot \mathbf{0} + \delta \cdot u \triangleleft t > u \triangleright \delta \cdot \mathbf{0}) \triangleleft u \leq t \triangleright \delta \cdot \mathbf{0} \\
& \quad + \delta \cdot t \triangleleft u > t \triangleright \delta \cdot \mathbf{0}) \triangleleft b \triangleright \delta \cdot \mathbf{0} \\
\stackrel{\text{C7,SUM4}}{=} & \sum_{\bar{v}} ((a^c u \triangleleft t \leq u \triangleright \delta \cdot \mathbf{0} + \delta \cdot u \triangleleft t > u \triangleright \delta \cdot \mathbf{0}) \triangleleft u \leq t \triangleright \delta \cdot \mathbf{0}) \triangleleft b \triangleright \delta \cdot \mathbf{0} \\
& \quad + \sum_{\bar{v}} (\delta \cdot t \triangleleft u > t \triangleright \delta \cdot \mathbf{0}) \triangleleft b \triangleright \delta \cdot \mathbf{0} \\
\stackrel{\text{C7,SUM4}}{=} & \sum_{\bar{v}} ((a^c u \triangleleft t \leq u \triangleright \delta \cdot \mathbf{0}) \triangleleft u \leq t \triangleright \delta \cdot \mathbf{0}) \triangleleft b \triangleright \delta \cdot \mathbf{0} \\
& \quad + \sum_{\bar{v}} ((\delta \cdot u \triangleleft t > u \triangleright \delta \cdot \mathbf{0}) \triangleleft u \leq t \triangleright \delta \cdot \mathbf{0}) \triangleleft b \triangleright \delta \cdot \mathbf{0} \\
& \quad + \sum_{\bar{v}} (\delta \cdot t \triangleleft u > t \triangleright \delta \cdot \mathbf{0}) \triangleleft b \triangleright \delta \cdot \mathbf{0} \\
\stackrel{\text{C4}}{=} & \sum_{\bar{v}} a^c u \triangleleft t \leq u \wedge u \leq t \wedge b \triangleright \delta \cdot \mathbf{0} + \sum_{\bar{v}} \delta \cdot u \triangleleft t > u \wedge u \leq t \wedge b \triangleright \delta \cdot \mathbf{0} \\
& \quad + \sum_{\bar{v}} \delta \cdot t \triangleleft u > t \wedge b \triangleright \delta \cdot \mathbf{0} \\
= & \sum_{\bar{v}} a^c u \triangleleft eq(t, u) \wedge b \triangleright \delta \cdot \mathbf{0} + \sum_{\bar{v}} \delta \cdot u \triangleleft t > u \wedge b \triangleright \delta \cdot \mathbf{0} \\
& \quad + \sum_{\bar{v}} \delta \cdot t \triangleleft t < u \wedge b \triangleright \delta \cdot \mathbf{0},
\end{aligned}$$

which is a basic term.

- (b) $p' \equiv \sum_{\bar{v}} a^c u \cdot p'_1 \triangleleft b \triangleright \delta \cdot \mathbf{0}$. Similar to the previous case: only axiom ATA3 has to be applied just before applying ATA1.
- (c) $p' \equiv p'_1 + p'_2$. By induction we have the existence of basic terms p''_1 and p''_2 such that $p'_1 t = p''_1$ and $p'_2 t = p''_2$. Then, $p' t = (p'_1 + p'_2) t \stackrel{\text{ATA2}}{=} p''_1 t + p''_2 t = p''_1 + p''_2$, which is a basic term.

The main virtue of the basic term theorem is that in proving an identity for process-closed terms, we can restrict the proof to basic terms. Instead of using induction on the structure of process-closed terms we can use induction on the structure of basic terms. This limits the cases to be considered considerably. Another example of such a proof is the proof of the following lemma. In Section 3, we frequently use induction on basic terms.

Lemma 2.6 For process-closed terms p we have $t \gg (t \gg p) = t \gg p$.

Proof. Without loss of generality we may assume that p is a basic term. We prove the lemma by induction on the structure of basic term p .

1. $p \equiv \sum_{\bar{v}} a^c u \triangleleft b \triangleright \delta \cdot \mathbf{0}$. This case is similar to the next case albeit that axiom ATB3 does not have to be applied.
2. $p \equiv \sum_{\bar{v}} a^c u \cdot p' \triangleleft b \triangleright \delta \cdot \mathbf{0}$. Then,

$$\begin{aligned}
t \gg (t \gg p) &= t \gg (t \gg \sum_{\bar{v}} a^c u \cdot p' \triangleleft b \triangleright \delta \cdot \mathbf{0}) \\
&\stackrel{\text{ATB4}}{=} \sum_{\bar{v}} t \gg (t \gg (a^c u \cdot p' \triangleleft b \triangleright \delta \cdot \mathbf{0})) \\
&\stackrel{\text{ATB5}}{=} \sum_{\bar{v}} t \gg (t \gg a^c u \cdot p' \triangleleft b \triangleright t \gg (t \gg \delta \cdot \mathbf{0})) \\
&\stackrel{\text{ATB3}}{=} \sum_{\bar{v}} (t \gg (t \gg a^c u)) \cdot p' \triangleleft b \triangleright t \gg (t \gg \delta \cdot \mathbf{0}) \\
&\stackrel{2.4.1}{=} \sum_{\bar{v}} (t \gg a^c u) \cdot p' \triangleleft b \triangleright t \gg \delta \cdot \mathbf{0} \\
&\stackrel{\text{ATB3}}{=} \sum_{\bar{v}} t \gg a^c u \cdot p' \triangleleft b \triangleright t \gg \delta \cdot \mathbf{0} \\
&\stackrel{\text{ATB5}}{=} \sum_{\bar{v}} t \gg (a^c u \cdot p' \triangleleft b \triangleright \delta \cdot \mathbf{0}) \\
&\stackrel{\text{ATB4}}{=} t \gg \sum_{\bar{v}} a^c u \cdot p' \triangleleft b \triangleright \delta \cdot \mathbf{0} \\
&= t \gg p.
\end{aligned}$$

3. $p \equiv p' + p''$. This part follows easily from the induction hypothesis and axiom ATB2.

3 Semantics of timed p CRL

In this section, we present a model in which processes parameterised with data can be interpreted. With respect to the abstract data types we assume the existence of a data algebra with universe \mathcal{D}_s for each sort symbol s . For instance, for the booleans we have that the universe contains the interpretation of \mathbf{t} and the interpretation of \mathbf{f} .

A valuation ν is a function from the set of variables V to the set of values $\bigcup_{s \in S} \mathcal{D}_s$ such that $\nu(v) \in \mathcal{D}_s$ if and only if $v \in V_s$. This means that a variable can only be mapped to a value of the right data algebra. For valuations ν and ν' and variable $v \in V$, we write $\nu[v]\nu'$, if for all $u \in V$, $u \neq v$ implies $\nu(u) = \nu'(u)$. Thus $\nu[v]\nu'$ means that the valuations ν and ν' are the same for all variables except possibly for the variable v . Given a valuation ν , we write $\llbracket t \rrbracket^\nu$ for the interpretation of a term t .

The process terms are mapped to processes and on these we define a suitable notion of *strong timed bisimulation*. Soundness of the axioms is proved w.r.t. the model. Completeness of the axioms is proved under the assumption that the data is completely axiomatised and adheres to certain other properties. This notion of completeness will be called *relative completeness*.

3.1 Interpretation of process terms

Definition 3.1 Given some $p\text{CRL}_t$ specification with a set of action declarations Act , the set of *actions* is defined by

$$\mathcal{A} = \{ \mathbf{a}(e_1, \dots, e_n) \mid \mathbf{a} : s_1 \times \dots \times s_n \in Act, e_i \in \mathcal{D}_{s_i} \}.$$

The set $\mathcal{A} \cup \{\delta\}$ is denoted by \mathcal{A}_δ . The set $\mathcal{P} = \mathcal{P}^\omega$ of *processes* is obtained by the following recursion:

$$\begin{aligned}
\mathcal{P}^0 &= \mathcal{A}_\delta, \\
\mathcal{P}^{n+1} &= \mathcal{P}^n \cup \{ p \cdot q, \sum P', p^c t, t \gg p \mid p, q \in \mathcal{P}^n, P' \neq \emptyset, P' \subseteq \mathcal{P}^n, t \in \mathcal{D}_T \}.
\end{aligned}$$

$a \xrightarrow{a}_t \checkmark$	$\frac{p \xrightarrow{a}_t \checkmark}{\sum(\{p\} \cup P) \xrightarrow{a}_t \checkmark}$	$\frac{p \xrightarrow{a}_t p'}{\sum(\{p\} \cup P) \xrightarrow{a}_t p'}$	$\frac{p \xrightarrow{a}_t \checkmark}{p \cdot t \xrightarrow{a}_t \checkmark}$	$\frac{p \xrightarrow{a}_t p'}{p \cdot t \xrightarrow{a}_t p'}$
$\frac{p \xrightarrow{a}_t \checkmark}{p \cdot q \xrightarrow{a}_t t \gg q}$	$\frac{p \xrightarrow{a}_t p'}{p \cdot q \xrightarrow{a}_t p' \cdot q}$	$\frac{p \xrightarrow{a}_t \checkmark} \quad t' \leq t}{t' \gg p \xrightarrow{a}_t \checkmark}$	$\frac{p \xrightarrow{a}_t p'} \quad t' \leq t}{t' \gg p \xrightarrow{a}_t p'}$	

Table 3: Transition rules for the $p\text{CRL}_t$ operators, where $a \in \mathcal{A}$, $p, p', q \in \mathcal{P}$, $t, t' \in \mathcal{D}_T$ and $P \subseteq \mathcal{P}$.

Definition 3.2 The interpretation of process-closed terms under a valuation ν is inductively defined by: for $\mathbf{a} : s_1 \times \dots \times s_n \in \text{Act}$ and $d_i \in T_{s_i}$, process-closed terms $p, q \in T_{\mathcal{P}}$, $b \in T_B$, and $t \in T_T$

$$\begin{aligned}
\llbracket \mathbf{a}(d_1, \dots, d_n) \rrbracket^\nu &= \mathbf{a}(\llbracket d_1 \rrbracket^\nu, \dots, \llbracket d_n \rrbracket^\nu), \\
\llbracket \delta \rrbracket^\nu &= \delta, \\
\llbracket p + q \rrbracket^\nu &= \sum \{ \llbracket p \rrbracket^\nu, \llbracket q \rrbracket^\nu \}, \\
\llbracket p \cdot q \rrbracket^\nu &= \llbracket p \rrbracket^\nu \cdot \llbracket q \rrbracket^\nu, \\
\llbracket p \cdot t \rrbracket^\nu &= \llbracket p \rrbracket^\nu \cdot \llbracket t \rrbracket^\nu, \\
\llbracket t \gg p \rrbracket^\nu &= \llbracket t \rrbracket^\nu \gg \llbracket p \rrbracket^\nu, \\
\llbracket p \triangleleft b \triangleright q \rrbracket^\nu &= \begin{cases} \llbracket p \rrbracket^\nu & \text{if } \llbracket b \rrbracket^\nu = \llbracket \mathbf{t} \rrbracket^\nu, \\ \llbracket q \rrbracket^\nu & \text{if } \llbracket b \rrbracket^\nu = \llbracket \mathbf{f} \rrbracket^\nu, \end{cases} \\
\llbracket \sum_v p \rrbracket^\nu &= \sum \{ \llbracket p \rrbracket^{\nu'} \mid \nu[v]\nu' \}.
\end{aligned}$$

This summation operator, which takes a set of processes as argument, is taken from [10].

3.2 Operational semantics and strong bisimulation

In this section, we define an operational semantics of processes in terms of the timed execution of actions. For this purpose we introduce *action relations* $_ \xrightarrow{a}_t _ \checkmark$ and $_ \xrightarrow{a}_t _ _$. By $p \xrightarrow{a}_t \checkmark$ we express that the process p can perform an action a at time t , and then terminate successfully at t . By $p \xrightarrow{a}_t p'$ we express that the process p evolves into process p' by performing action a at time t .

Definition 3.3 The action relations $_ \xrightarrow{a}_t _ \checkmark \subseteq \mathcal{P} \times \mathcal{A} \times \mathcal{D}_T$ and $_ \xrightarrow{a}_t _ _ \subseteq \mathcal{P} \times \mathcal{A} \times \mathcal{D}_T \times \mathcal{P}$ are defined by the transition rules in Table 3.

The execution of an action does not take time: it is possible that the process p' can subsequently perform other actions at t , e.g.

$$a \cdot t \cdot b \cdot t \xrightarrow{a}_t t \gg b \cdot t \xrightarrow{b}_t \checkmark.$$

Given the action relations we have a good understanding of the steps which can be performed by a process. An obvious question is when two processes are equivalent. Such a notion is required to justify the axioms. Strong bisimulation is a commonly used process equivalence. Therefore, we provide a timed variant of strong bisimulation, called strong timed bisimulation. However, bisimulation cannot be defined in terms of actions only, e.g. the processes $\delta \cdot t$ and $\delta \cdot u$ are considered equivalent only if $t = u$. We introduce the *delay relation* $U_t(p)$, which expresses that p can at least idle until time t . It has the same purpose as the ultimate delay in [2, 3, 21].

Definition 3.4 The delay relation $U_{_} \subseteq \mathcal{D}_T \times \mathcal{P}$ is defined by the transition rules in Table 4.

$U_t(a)$	$\frac{U_t(p)}{U_t(p \cdot q)}$	$\frac{U_t(p)}{U_t(\sum P \cup \{p\})}$	$\frac{U_t(p) \quad t \leq t'}{U_t(p \cdot t')}$	$\frac{U_t(p)}{U_t(t' \gg p)}$	$\frac{t \leq t'}{U_t(t' \gg p)}$
----------	---------------------------------	---	--	--------------------------------	-----------------------------------

Table 4: Transition rules for the delay relation, where $a \in \mathcal{A}_\delta$, $p, q \in \mathcal{P}$, $t, t' \in \mathcal{D}_T$ and $P \subseteq \mathcal{P}$.

Definition 3.5 A symmetric relation $R \subseteq \mathcal{P} \times \mathcal{P}$ is a *strong timed bisimulation* if for all $(p, q) \in R$ it holds that

1. if $p \xrightarrow{a}_t \checkmark$, then $q \xrightarrow{a}_t \checkmark$,
2. if $p \xrightarrow{a}_t p'$, then there is a q' such that $q \xrightarrow{a}_t q'$ and $(p', q') \in R$, and
3. if $U_t(p)$, then $U_t(q)$,

for all $a \in \mathcal{A}$, $t \in \mathcal{D}_T$ and $p' \in \mathcal{P}$.

Processes p and q are *strongly timed bisimilar*, notation $p \Leftrightarrow_t q$, if and only if there is a strong timed bisimulation R such that $(p, q) \in R$. Observe that the union of any set of strong timed bisimulations is again a strong timed bisimulation.

Lemma 3.1 (Congruence) Strong timed bisimilarity is a congruence with respect to the operators defined on \mathcal{P} .

Proof. It is not hard to prove that strong timed bisimilarity is an equivalence. We omit the proof. Next, we give bisimulations that witness the congruence of strong timed bisimilarity with respect to the operators \gg , \cdot , \sum , and \cdot . For \cdot we give a more detailed proof.

1. \gg . Let $R : p \Leftrightarrow_t q$. Let $t \in \mathcal{D}_T$. Obviously, the relation $\{(t \gg p, t \gg q), (t \gg q, t \gg p)\} \cup R$ is a strong timed bisimulation.
2. \cdot . Let $R_1 : p_1 \Leftrightarrow_t q_1$ and $R_2 : p_2 \Leftrightarrow_t q_2$. Define $R = \{(p'_1 \cdot p_2, q'_1 \cdot q_2), (q'_1 \cdot q_2, p'_1 \cdot p_2), (t \gg p_2, t \gg q_2), (t \gg q_2, t \gg p_2) \mid t \in \mathcal{D}_T \wedge (p'_1, q'_1) \in R_1\} \cup R_2$. We do not have to consider pairs in R that are also in R_2 as for these we have already assumed that they are strongly timed bisimilar. We also do not have to consider the pairs in R of the form $(t \gg p_2, t \gg q_2)$ and vice versa with $(p_2, q_2) \in R_2$ as the strong timed bisimulation from the proof of congruence of \gg is contained in R . Thus, we only have to consider pairs of the form $(p'_1 \cdot p_2, q'_1 \cdot q_2)$ with $(p'_1, q'_1) \in R_1$. For pairs of this form it is easy to prove that these satisfy the conditions of the definition of strong timed bisimulations. Hence the relation R is a strong timed bisimulation.
3. \sum . Let R be a strong timed bisimulation such that for all $p \in P$ there exists $q \in Q$ such that $(p, q) \in R$ and vice versa for all $q \in Q$ there exists $p \in P$ such that $(q, p) \in R$. The relation $R' = \{(\sum P, \sum Q), (\sum Q, \sum P)\} \cup R$ is obviously a strong timed bisimulation.
4. \cdot . Let $R : p \Leftrightarrow_t q$. Let $t \in \mathcal{D}_T$. Obviously the relation $\{(p \cdot t, q \cdot t), (q \cdot t, p \cdot t)\} \cup R$ is a strong timed bisimulation.

Without further proof we state that the axioms are sound with respect to the model of closed terms modulo strong timed bisimilarity. This means that any two derivably equal process-closed terms are strongly timed bisimilar.

Theorem 3.1 (Soundness) With respect to strong timed bisimulation $p\text{CRL}_t$ is a sound axiom system.

4 Relative completeness

In this section, we prove completeness of the axiomatisation with respect to the operational semantics modulo strong timed bisimilarity. For $p\text{CRL}$ a completeness result is given in [10], where it is shown that the result can only be partial. It relies on the existence of built-in equality and Skolem functions for all data sorts involved.

Definition 4.1 A data algebra has *built-in equality* if, for every sort s , it has a function declaration $eq : s \times s \rightarrow B$, such that for all terms t_1 and t_2 of sort s , it holds that $\vdash t_1 = t_2$ if and only if $\vdash eq(t_1, t_2) = \mathbf{t}$.

Definition 4.2 A data algebra \mathcal{D} has *built-in Skolem functions* if for every first-order formula ϕ with free variables $\{x, y_1, \dots, y_n\}$ there exists a term $t_\phi(y_1, \dots, y_n)$ such that $\mathcal{D} \models (\exists x)\phi$ implies $\mathcal{D} \models \phi(t_\phi(y_1, \dots, y_n), y_1, \dots, y_n)$.

For a more detailed treatment of these notions we refer to [10].

As $p\text{CRL}_t$ is an extension of $p\text{CRL}$ we also need these assumptions about the data sorts for the completeness of $p\text{CRL}_t$. In fact, as $p\text{CRL}_t$ is a syntactic extension of $p\text{CRL}$, the proof of completeness for $p\text{CRL}_t$ contains the proof of completeness of $p\text{CRL}$. We do not repeat the proof presented in [10], but formulate our proof of completeness in such a way that the proof of completeness of $p\text{CRL}$ is reused.

4.1 Well-timedness and deadlock-saturation

The main difference between $p\text{CRL}$ and $p\text{CRL}_t$ is the addition of the at-operator. This can easily be seen by considering the basic terms. We can even conclude that the only difference is the application of the at-operator on atomic actions (including deadlock).

To reuse the proof of completeness of $p\text{CRL}$, we translate timed process terms as untimed process terms in such a way that strongly timed bisimilar process terms are strongly bisimilar after translating them, and vice versa. Furthermore, two untimed process terms that are derivably equal in $p\text{CRL}$ are also derivably equal after applying the inverse of the translation.

The only way to translate the timed process terms as untimed process terms is to consider the time assignment to atomic actions as an additional data parameter of the atomic action. For example, the timed atomic action $\mathbf{a}(d)\cdot 3$ might be translated to the untimed atomic action $\mathbf{a}(d, 3)$. There are, however, two problems with such a translation.

First, there is the problem of ill-timedness. In $p\text{CRL}_t$ we have the following identity: $\mathbf{a}(d)\cdot 3 \cdot (\mathbf{b}(e)\cdot 2 + \mathbf{c}(f)\cdot 4) = \mathbf{a}(d)\cdot 3 \cdot \mathbf{c}(f)\cdot 4$. Translating the timing assignment as an additional parameter yields the identity $\mathbf{a}(d, 3) \cdot (\mathbf{b}(e, 2) + \mathbf{c}(f, 4)) = \mathbf{a}(d, 3) \cdot \mathbf{c}(f, 4)$. This identity does not hold in $p\text{CRL}$. The reason for this is that timing assignments in $p\text{CRL}_t$ influence the behaviour in such a way that data parameters attached to atomic action cannot simulate in $p\text{CRL}$. Our solution to this problem is the introduction of the notion of well-timedness. We will prove that for every basic term there exists a derivably equal well-timed basic term.

Second, there is the more technical problem that deadlock cannot have a parameter in $p\text{CRL}$. Hence it is not clear how to translate $\delta\cdot 3$. We cannot simply translate $\delta\cdot 3$ as δ since in that case $\delta\cdot 3$ and $\delta\cdot 4$ are translated both as δ . Our solution to this problem comes from the following observation. In a timed process term, there are many time deadlocks implicitly included. For example, the process term $a\cdot 3$ implicitly has time deadlock summands $\delta\cdot t$ for each $t \leq 3$. If we can succeed in making all implicit deadlocks explicitly visible, then we can translate a time deadlock as a special atomic action with one data parameter representing the time assignment of that deadlock. That is, we will translate $\delta\cdot 3$ as $\Delta(3)$. Making all implicit time deadlocks explicitly available is called deadlock-saturation. We show that for each basic term a deadlock-saturated basic terms exists that is derivably equal, and hence, that the restriction to deadlock-saturated basic terms is allowed.

By restricting our attention to deadlock-saturated well-timed basic terms we can prove that the translation sketched above satisfies the criteria mentioned.

Definition 4.3 (Well-timedness) The set of well-timed basic terms is inductively defined as follows:

1. $\sum_{\bar{v}} a^c t \triangleleft b \triangleright \delta \cdot \mathbf{0}$ is well-timed;
2. if r is a well-timed basic term and $t \gg r = r$, then $\sum_{\bar{v}} a^c t \cdot r \triangleleft b \triangleright \delta \cdot \mathbf{0}$ is well-timed;
3. if r and r' are well-timed basic terms, then $r + r'$ is well-timed.

The following theorem states that every basic term is derivably equal to a well-timed basic term. Combined with the Basic Term Theorem we thus have that every process-closed term is derivably equal to a well-timed basic term. Thus, in cases where we want to prove an equality over process-closed terms, we can restrict ourselves to a proof that the equality holds for well-timed basic terms.

Theorem 4.1 (Well-timedness) For every basic term r there exists a well-timed basic term s such that $r = s$.

Proof. We prove this theorem by induction on the structure of basic term r .

1. $r \equiv \sum_{\bar{v}} a^c u \triangleleft b \triangleright \delta \cdot \mathbf{0}$. Then r is well-timed by definition.
2. $r \equiv \sum_{\bar{v}} a^c u \cdot r' \triangleleft b \triangleright \delta \cdot \mathbf{0}$. By induction we have the existence of a well-timed basic term s' such that $r' = s'$. First we prove by induction on the structure of well-timed basic term r , that for every well-timed basic term r and term t of sort T , there exists a well-timed basic term s such that $t \gg r = s$.
 - (a) $r \equiv \sum_{\bar{v}} a^c u \triangleleft b \triangleright \delta \cdot \mathbf{0}$. Then,

$$\begin{aligned}
t \gg r &= t \gg (\sum_{\bar{v}} a^c u \triangleleft b \triangleright \delta \cdot \mathbf{0}) \\
&\stackrel{\text{ATB4}}{=} \sum_{\bar{v}} t \gg (a^c u \triangleleft b \triangleright \delta \cdot \mathbf{0}) \\
&\stackrel{\text{ATB5}}{=} \sum_{\bar{v}} (t \gg a^c u) \triangleleft b \triangleright (t \gg \delta \cdot \mathbf{0}) \\
&\stackrel{\text{ATB1,2.4.2}}{=} \sum_{\bar{v}} (a^c u \triangleleft t \leq u \triangleright \delta \cdot t) \triangleleft b \triangleright \delta \cdot t \\
&\stackrel{\text{C4}}{=} \sum_{\bar{v}} a^c u \triangleleft t \leq u \wedge b \triangleright \delta \cdot t \\
&\stackrel{\text{C3,SUM4}}{=} \sum_{\bar{v}} a^c u \triangleleft t \leq u \wedge b \triangleright \delta \cdot \mathbf{0} + \sum_{\bar{v}} \delta \cdot t \triangleleft \neg(t \leq u \wedge b) \triangleright \delta \cdot \mathbf{0},
\end{aligned}$$

which is a well-timed basic term.

- (b) $r \equiv \sum_{\bar{v}} a^c u \cdot r' \triangleleft b \triangleright \delta \cdot \mathbf{0}$. As r is well-timed we have $u \gg r' = r'$. Then,

$$\begin{aligned}
t \gg r &= t \gg (\sum_{\bar{v}} a^c u \cdot r' \triangleleft b \triangleright \delta \cdot \mathbf{0}) \\
&\stackrel{\text{ATB4}}{=} \sum_{\bar{v}} t \gg (a^c u \cdot r' \triangleleft b \triangleright \delta \cdot \mathbf{0}) \\
&\stackrel{\text{ATB5,ATB3}}{=} \sum_{\bar{v}} (t \gg a^c u) \cdot r' \triangleleft b \triangleright (t \gg \delta \cdot \mathbf{0}) \\
&\stackrel{\text{ATB1,2.4.2}}{=} \sum_{\bar{v}} (a^c u \triangleleft t \leq u \triangleright \delta \cdot t) \cdot r' \triangleleft b \triangleright \delta \cdot t \\
&\stackrel{\text{C6,ATA3,A7}}{=} \sum_{\bar{v}} (a^c u \cdot r' \triangleleft t \leq u \triangleright \delta \cdot t) \triangleleft b \triangleright \delta \cdot t \\
&\stackrel{\text{C4}}{=} \sum_{\bar{v}} a^c u \cdot r' \triangleleft t \leq u \wedge b \triangleright \delta \cdot t \\
&\stackrel{\text{C3,SUM4}}{=} \sum_{\bar{v}} a^c u \cdot r' \triangleleft t \leq u \wedge b \triangleright \delta \cdot \mathbf{0} + \sum_{\bar{v}} \delta \cdot t \triangleleft \neg(t \leq u \wedge b) \triangleright \delta \cdot \mathbf{0}.
\end{aligned}$$

Obviously, this is a well-timed basic term.

- (c) $r \equiv r' + r''$. By induction we have the existence of well-timed basic terms s' and s'' such that $t \gg r' = s'$ and $t \gg r'' = s''$. Then, $t \gg r = t \gg (r' + r'') \stackrel{\text{ATB2}}{=} t \gg r' + t \gg r'' = s' + s''$ which is a well-timed basic term.

Now, we have the existence of a well-timed basic term s'' such that $u \gg s' = s''$. Then,

$$\begin{aligned} r &= \sum_{\bar{v}} a^\epsilon u \cdot r' \triangleleft b \triangleright \delta \bullet \mathbf{0} \\ &= \sum_{\bar{v}} a^\epsilon u \cdot s' \triangleleft b \triangleright \delta \bullet \mathbf{0} \\ &\stackrel{\text{AT2}}{=} \sum_{\bar{v}} a^\epsilon u \cdot u \gg s' \triangleleft b \triangleright \delta \bullet \mathbf{0} \\ &= \sum_{\bar{v}} a^\epsilon u \cdot s'' \triangleleft b \triangleright \delta \bullet \mathbf{0}. \end{aligned}$$

By Lemma 2.6 we have that $u \gg s'' = u \gg (u \gg s') = u \gg s' = s''$, and hence the above term is well-timed.

3. $r \equiv r' + r''$. By induction we have the existence of well-timed basic terms s' and s'' such that $r' = s'$ and $r'' = s''$. Thus, $r = r' + r'' = s' + s''$ which is a well-timed basic term.

The fact that we can restrict ourselves to well-timed basic terms instead of arbitrary basic terms brings us closer to our goal, the completeness result. As explained before we wish to embed timed processes into untimed processes. One of the major difficulties in this embedding is that time restricts behaviour. For well-timed basic terms, however, time is reduced to just an annotation with an action term. In the embedding we will simply add it to the parameter list of a corresponding action term.

In this section, we will present an even more specific notion of basic terms. To each basic term as many deadlocks will be added as possible while keeping the result equal to the original. The reason for this deadlock-saturation is that later we will embed the timed process terms into untimed process terms by replacing timed deadlocks by a special atomic action with a parameter representing the time stamp.

Definition 4.4 (Deadlock-saturation) The set of deadlock-saturated basic terms is inductively defined as follows:

1. $\sum_{\bar{v}} a^\epsilon t \triangleleft b \triangleright \delta \bullet \mathbf{0} + \sum_{\bar{v}, u} \delta^\epsilon u \triangleleft u \leq t \wedge b \triangleright \delta \bullet \mathbf{0}$ is a deadlock-saturated basic term;
2. if r is a deadlock-saturated basic term, then $\sum_{\bar{v}} a^\epsilon t \cdot r \triangleleft b \triangleright \delta \bullet \mathbf{0} + \sum_{\bar{v}, u} \delta^\epsilon u \triangleleft u \leq t \wedge b \triangleright \delta \bullet \mathbf{0}$ is a deadlock-saturated basic term;
3. if r and r' are deadlock-saturated basic terms, then $r + r'$ is a deadlock-saturated basic term.

The following result states that every basic term can be transformed into a deadlock-saturated basic term. Hence, it is allowed to restrict ourselves to deadlock-saturated basic terms in cases where we are only interested in closed terms.

Theorem 4.2 For every basic term p there exists a deadlock-saturated basic term q such that $p\text{CRL}_t \vdash p = q$.

Proof. We prove this theorem by induction on the structure of basic term p .

1. $p \equiv \sum_{\bar{v}} a^\epsilon t \triangleleft b \triangleright \delta \bullet \mathbf{0}$. Let $q = \sum_{\bar{v}} a^\epsilon t \triangleleft b \triangleright \delta \bullet \mathbf{0} + \sum_{\bar{v}, u} \delta^\epsilon u \triangleleft u \leq t \wedge b \triangleright \delta \bullet \mathbf{0}$ where u is a fresh variable. Then,

$$\begin{aligned} q &= \sum_{\bar{v}} a^\epsilon t \triangleleft b \triangleright \delta \bullet \mathbf{0} + \sum_{\bar{v}, u} \delta^\epsilon u \triangleleft u \leq t \wedge b \triangleright \delta \bullet \mathbf{0} \\ &\stackrel{\text{SUM4}}{=} \sum_{\bar{v}} (a^\epsilon t \triangleleft b \triangleright \delta \bullet \mathbf{0} + \sum_u \delta^\epsilon u \triangleleft u \leq t \wedge b \triangleright \delta \bullet \mathbf{0}) \\ &\stackrel{\text{C4}}{=} \sum_{\bar{v}} (a^\epsilon t \triangleleft b \triangleright \delta \bullet \mathbf{0} + \sum_u (\delta^\epsilon u \triangleleft u \leq t \triangleright \delta \bullet \mathbf{0}) \triangleleft b \triangleright \delta \bullet \mathbf{0}) \\ &\stackrel{\text{SUM12}'}{=} \sum_{\bar{v}} (a^\epsilon t \triangleleft b \triangleright \delta \bullet \mathbf{0} + (\sum_u (\delta^\epsilon u \triangleleft u \leq t \triangleright \delta \bullet \mathbf{0})) \triangleleft b \triangleright \delta \bullet \mathbf{0}) \\ &\stackrel{2.5}{=} \sum_{\bar{v}} (a^\epsilon t \triangleleft b \triangleright \delta \bullet \mathbf{0} + \delta^\epsilon t \triangleleft b \triangleright \delta \bullet \mathbf{0}) \\ &\stackrel{\text{C7}}{=} \sum_{\bar{v}} (a^\epsilon t + \delta^\epsilon t) \triangleleft b \triangleright \delta \bullet \mathbf{0} \\ &\stackrel{\text{ATA2}}{=} \sum_{\bar{v}} (a + \delta)^\epsilon t \triangleleft b \triangleright \delta \bullet \mathbf{0} \\ &\stackrel{\text{A6}^-}{=} \sum_{\bar{v}} a^\epsilon t \triangleleft b \triangleright \delta \bullet \mathbf{0} \\ &= p. \end{aligned}$$

2. $p \equiv \sum_{\bar{v}} a^{\circ}t \cdot p' \triangleleft b \triangleright \delta \cdot \mathbf{0}$. By induction we have the existence of a deadlock-saturated basic term q' such that $p\text{CRL}_t \vdash p' = q'$. Let $q = \sum_{\bar{v}} a^{\circ}t \cdot q' \triangleleft b \triangleright \delta \cdot \mathbf{0} + \sum_{\bar{v},u} \delta^{\circ}u \triangleleft u \leq t \wedge b \triangleright \delta \cdot \mathbf{0}$ where u is a fresh variable. Then,

$$\begin{aligned}
q &= \sum_{\bar{v}} a^{\circ}t \cdot q' \triangleleft b \triangleright \delta \cdot \mathbf{0} + \sum_{\bar{v},u} \delta^{\circ}u \triangleleft u \leq t \wedge b \triangleright \delta \cdot \mathbf{0} \\
&\stackrel{\text{SUM4}}{=} \sum_{\bar{v}} (a^{\circ}t \cdot q' \triangleleft b \triangleright \delta \cdot \mathbf{0} + \sum_u \delta^{\circ}u \triangleleft u \leq t \wedge b \triangleright \delta \cdot \mathbf{0}) \\
&\stackrel{\text{C4}}{=} \sum_{\bar{v}} (a^{\circ}t \cdot q' \triangleleft b \triangleright \delta \cdot \mathbf{0} + \sum_u (\delta^{\circ}u \triangleleft u \leq t \triangleright \delta \cdot \mathbf{0}) \triangleleft b \triangleright \delta \cdot \mathbf{0}) \\
&\stackrel{\text{SUM12}'}{=} \sum_{\bar{v}} (a^{\circ}t \cdot q' \triangleleft b \triangleright \delta \cdot \mathbf{0} + (\sum_u (\delta^{\circ}u \triangleleft u \leq t \triangleright \delta \cdot \mathbf{0})) \triangleleft b \triangleright \delta \cdot \mathbf{0}) \\
&\stackrel{2.5}{=} \sum_{\bar{v}} (a^{\circ}t \cdot q' \triangleleft b \triangleright \delta \cdot \mathbf{0} + \delta^{\circ}t \triangleleft b \triangleright \delta \cdot \mathbf{0}) \\
&\stackrel{\text{C7}}{=} \sum_{\bar{v}} (a^{\circ}t \cdot q' + \delta^{\circ}t) \triangleleft b \triangleright \delta \cdot \mathbf{0} \\
&\stackrel{\text{ATA3,ATA2,A7,A4}}{=} \sum_{\bar{v}} ((a + \delta) \cdot q')^{\circ}t \triangleleft b \triangleright \delta \cdot \mathbf{0} \\
&\stackrel{\text{A6}^-, \text{ATA3}}{=} \sum_{\bar{v}} a^{\circ}t \cdot q' \triangleleft b \triangleright \delta \cdot \mathbf{0} \\
&= \sum_{\bar{v}} a^{\circ}t \cdot p' \triangleleft b \triangleright \delta \cdot \mathbf{0} \\
&= p.
\end{aligned}$$

3. $p \equiv p' + p''$. By induction we have the existence of deadlock-saturated basic terms q' and q'' such that $p\text{CRL}_t \vdash p' = q'$ and $p\text{CRL}_t \vdash p'' = q''$. Then, $p\text{CRL}_t \vdash p = p' + p'' = q' + q''$. Obviously, $q' + q''$ is a deadlock-saturated basic term.

As a consequence of the Theorems 4.1 and 4.2, for each basic term there exists a derivably equal deadlock-saturated well-timed basic term.

Corollary 4.1 For every basic term p there exists a deadlock-saturated, well-timed basic term q such that $p\text{CRL}_t \vdash p = q$.

Proof. By Theorem 4.1 there exists a well-timed basic term p' such that $p\text{CRL}_t \vdash p = p'$. Obviously p' itself is a basic term. Hence, by Theorem 4.2, there exists a deadlock-saturated basic term q such that $p\text{CRL}_t \vdash p' = q$. By construction deadlock-saturation preserves well-timedness. Hence, q is both well-timed and deadlock-saturated.

4.2 Time-free abstraction

Next, we present an embedding of well-timed basic terms into basic terms in the theory $p\text{CRL}$. Thereto, we assume that for every action declaration $a : s_1 \times \cdots \times s_n$ in $p\text{CRL}_t$, we have an action declaration $a : s_1 \times \cdots \times s_n \times T$ in $p\text{CRL}$. We prove that this embedding preserves strong bisimilarity in the sense that the embeddings of two strongly timed bisimilar process terms are strongly bisimilar.

Definition 4.5 (Time-free abstraction) The mapping π_{tf} on basic terms is inductively defined as follows:

$$\begin{aligned}
\pi_{\text{tf}}(\sum_{\bar{v}} \mathbf{a}(\bar{e})^{\circ}t \triangleleft b \triangleright \delta \cdot \mathbf{0}) &= \sum_{\bar{v}} \mathbf{a}(\bar{e}, t) \triangleleft b \triangleright \delta, \\
\pi_{\text{tf}}(\sum_{\bar{v}} \delta^{\circ}t \triangleleft b \triangleright \delta \cdot \mathbf{0}) &= \sum_{\bar{v}} \Delta(t) \triangleleft b \triangleright \delta, \\
\pi_{\text{tf}}(\sum_{\bar{v}} \mathbf{a}(\bar{e})^{\circ}t \cdot r \triangleleft b \triangleright \delta \cdot \mathbf{0}) &= \sum_{\bar{v}} \mathbf{a}(\bar{e}, t) \cdot \pi_{\text{tf}}(r) \triangleleft b \triangleright \delta, \\
\pi_{\text{tf}}(r + r') &= \pi_{\text{tf}}(r) + \pi_{\text{tf}}(r').
\end{aligned}$$

Theorem 4.3 For deadlock-saturated well-timed basic terms p and q and arbitrary valuations ν and ξ : if $\llbracket p \rrbracket^{\nu} \Leftrightarrow_t \llbracket q \rrbracket^{\xi}$, then $\llbracket \pi_{\text{tf}}(p) \rrbracket^{\nu} \Leftrightarrow \llbracket \pi_{\text{tf}}(q) \rrbracket^{\xi}$.

Proof. We prove this theorem by induction on the number of symbols of p and q . Suppose that $\llbracket p \rrbracket^\nu \Leftrightarrow_t \llbracket q \rrbracket^\xi$ for some valuations ν and ξ . Let R be defined as follows: if $\llbracket p \rrbracket^\nu \Leftrightarrow_t \llbracket q \rrbracket^\xi$ for some deadlock-saturated well-timed basic terms p and q and arbitrary valuations ν and ξ , then $(\llbracket \pi_{\text{tf}}(p) \rrbracket^\nu, \llbracket \pi_{\text{tf}}(q) \rrbracket^\xi) \in R$.

By induction on the structure of basic term p we can easily prove that the transitions of the process associated with basic term p under valuation ν are of one of the following forms:

1. $\llbracket p \rrbracket^\nu \xrightarrow{a}_t \checkmark$;
2. $\llbracket p \rrbracket^\nu \xrightarrow{a}_t t \gg \llbracket p' \rrbracket^{\nu'}$ for some deadlock-saturated well-timed basic term p' and valuation ν' .

Now, we first prove the following lemmata: for arbitrary deadlock-saturated well-timed basic terms p and p' , $\mathbf{a} \in \text{Act}$, $t \in \mathcal{D}_T$, and valuations χ and χ'

1. if $\mathbf{a} \neq \Delta$ and $\llbracket \pi_{\text{tf}}(p) \rrbracket^\chi \xrightarrow{\mathbf{a}(\vec{d}, t)} \checkmark$, then $\llbracket p \rrbracket^\chi \xrightarrow{\mathbf{a}(\vec{d})}_t \checkmark$;
2. if $\llbracket p \rrbracket^\chi \xrightarrow{\mathbf{a}(\vec{d})}_t \checkmark$, then $\llbracket \pi_{\text{tf}}(p) \rrbracket^\chi \xrightarrow{\mathbf{a}(\vec{d}, t)} \checkmark$;
3. if $\llbracket \pi_{\text{tf}}(p) \rrbracket^\chi \xrightarrow{\Delta(t)} \checkmark$, then $U_t(\llbracket p \rrbracket^\chi)$;
4. if $U_t(\llbracket p \rrbracket^\chi)$, then $\llbracket \pi_{\text{tf}}(p) \rrbracket^\chi \xrightarrow{\Delta(t)} \checkmark$;
5. if $\llbracket \pi_{\text{tf}}(p) \rrbracket^\chi \xrightarrow{\mathbf{a}(\vec{d}, t)} \llbracket \pi_{\text{tf}}(p') \rrbracket^{\chi'}$, then $\llbracket p \rrbracket^\chi \xrightarrow{\mathbf{a}(\vec{d})}_t t \gg \llbracket p' \rrbracket^{\chi'}$ and $t \gg \llbracket p' \rrbracket^{\chi'} \Leftrightarrow_t \llbracket p' \rrbracket^{\chi'}$;
6. if $\llbracket p \rrbracket^\chi \xrightarrow{\mathbf{a}(\vec{d})}_t t \gg \llbracket p' \rrbracket^{\chi'}$, then $\llbracket \pi_{\text{tf}}(p) \rrbracket^\chi \xrightarrow{\mathbf{a}(\vec{d}, t)} \llbracket \pi_{\text{tf}}(p') \rrbracket^{\chi'}$.

Next, we prove that R is a strong timed bisimulation. First, suppose that $\llbracket \pi_{\text{tf}}(p) \rrbracket^\nu \xrightarrow{\mathbf{a}(\vec{d}, t)} \checkmark$. If $\mathbf{a} = \Delta$, then $U_t(\llbracket p \rrbracket^\nu)$. Therefore, $U_t(\llbracket q \rrbracket^\xi)$. Thus, $\llbracket \pi_{\text{tf}}(q) \rrbracket^\xi \xrightarrow{\Delta(t)} \checkmark$. If $\mathbf{a} \neq \Delta$, then $\llbracket p \rrbracket^\nu \xrightarrow{\mathbf{a}(\vec{d})}_t \checkmark$. Therefore, $\llbracket q \rrbracket^\xi \xrightarrow{\mathbf{a}(\vec{d})}_t \checkmark$. Thus, $\llbracket \pi_{\text{tf}}(q) \rrbracket^\xi \xrightarrow{\mathbf{a}(\vec{d}, t)} \checkmark$.

Second, suppose that $\llbracket \pi_{\text{tf}}(p) \rrbracket^\nu \xrightarrow{\mathbf{a}(\vec{d}, t)} \llbracket \pi_{\text{tf}}(p') \rrbracket^{\nu'}$ for some deadlock-saturated well-timed basic term p' and valuation ν' . Then $\llbracket p \rrbracket^\nu \xrightarrow{\mathbf{a}(\vec{d})}_t t \gg \llbracket p' \rrbracket^{\nu'}$. Therefore, $\llbracket q \rrbracket^\xi \xrightarrow{\mathbf{a}(\vec{d})}_t t \gg \llbracket q' \rrbracket^{\xi'}$ for some q' and ξ' such that $t \gg \llbracket p' \rrbracket^{\nu'} \Leftrightarrow_t t \gg \llbracket q' \rrbracket^{\xi'}$. Thus, $\llbracket \pi_{\text{tf}}(q) \rrbracket^\xi \xrightarrow{\mathbf{a}(\vec{d}, t)} \llbracket \pi_{\text{tf}}(q') \rrbracket^{\xi'}$. As p and q are well-timed we also have $t \gg \llbracket p' \rrbracket^{\nu'} \Leftrightarrow_t \llbracket p' \rrbracket^{\nu'}$ and $t \gg \llbracket q' \rrbracket^{\xi'} \Leftrightarrow_t \llbracket q' \rrbracket^{\xi'}$. Thus, $(\llbracket \pi_{\text{tf}}(p) \rrbracket^{\nu'}, \llbracket \pi_{\text{tf}}(q) \rrbracket^{\xi'}) \in R$.

Next, we present a mapping π_t from $p\text{CRL}$ -terms to $p\text{CRL}_t$ -terms and we show that this mapping is the inverse of the mapping π_{tf} for basic terms.

Definition 4.6 The mapping π_t is defined as follows:

$$\begin{aligned}
\pi_t(x) &= x, \\
\pi_t(\delta) &= \delta \cdot \mathbf{0}, \\
\pi_t(\mathbf{a}) &= \delta \cdot \mathbf{0}, \\
\pi_t(\mathbf{a}(\vec{d}, t)) &= \mathbf{a}(\vec{d}) \cdot t, \\
\pi_t(\Delta) &= \delta \cdot \mathbf{0}, \\
\pi_t(\Delta(t)) &= \delta \cdot t, \\
\pi_t(\Delta(\vec{d}, t)) &= \delta \cdot \mathbf{0}, \\
\pi_t(p + q) &= \pi_t(p) + \pi_t(q), \\
\pi_t(p \cdot q) &= \pi_t(p) \cdot \pi_t(q), \\
\pi_t(p \triangleleft b \triangleright q) &= \pi_t(p) \triangleleft b \triangleright \pi_t(q), \\
\pi_t(\sum_v p) &= \sum_v \pi_t(p).
\end{aligned}$$

Lemma 4.1 For basic terms p we have

$$p\text{CRL}_t \vdash \pi_t(\pi_{\text{tf}}(p)) = p.$$

Proof. We prove this lemma by induction on the structure of basic term p .

1. $p \equiv \sum_{\vec{v}} \mathbf{a}(\vec{d}) \cdot t \triangleleft b \triangleright \delta \cdot \mathbf{0}$. Then,

$$\begin{aligned} p\text{CRL}_t \vdash \pi_t(\pi_{\text{tf}}(p)) &= \pi_t(\pi_{\text{tf}}(\sum_{\vec{v}} \mathbf{a}(\vec{d}) \cdot t \triangleleft b \triangleright \delta \cdot \mathbf{0})) \\ &= \pi_t(\sum_{\vec{v}} \mathbf{a}(\vec{d}, t) \triangleleft b \triangleright \delta) \\ &= \sum_{\vec{v}} \pi_t(\mathbf{a}(\vec{d}, t) \triangleleft b \triangleright \delta) \\ &= \sum_{\vec{v}} \pi_t(\mathbf{a}(\vec{d}, t)) \triangleleft b \triangleright \pi_t(\delta) \\ &= \sum_{\vec{v}} \mathbf{a}(\vec{d}) \cdot t \triangleleft b \triangleright \delta \cdot \mathbf{0} \\ &= p. \end{aligned}$$

2. $p \equiv \sum_{\vec{v}} \mathbf{a}(\vec{d}) \cdot t \cdot p' \triangleleft b \triangleright \delta \cdot \mathbf{0}$. By induction we have $p\text{CRL}_t \vdash \pi_t(\pi_{\text{tf}}(p')) = p'$. Then,

$$\begin{aligned} p\text{CRL}_t \vdash \pi_t(\pi_{\text{tf}}(p)) &= \pi_t(\pi_{\text{tf}}(\sum_{\vec{v}} \mathbf{a}(\vec{d}) \cdot t \cdot p' \triangleleft b \triangleright \delta \cdot \mathbf{0})) \\ &= \pi_t(\sum_{\vec{v}} \mathbf{a}(\vec{d}, t) \cdot \pi_{\text{tf}}(p') \triangleleft b \triangleright \delta) \\ &= \sum_{\vec{v}} \pi_t(\mathbf{a}(\vec{d}, t) \cdot \pi_{\text{tf}}(p') \triangleleft b \triangleright \delta) \\ &= \sum_{\vec{v}} \pi_t(\mathbf{a}(\vec{d}, t) \cdot \pi_{\text{tf}}(p')) \triangleleft b \triangleright \pi_t(\delta) \\ &= \sum_{\vec{v}} \mathbf{a}(\vec{d}) \cdot t \cdot \pi_t(\pi_{\text{tf}}(p')) \triangleleft b \triangleright \delta \cdot \mathbf{0} \\ &= \sum_{\vec{v}} \mathbf{a}(\vec{d}) \cdot t \cdot p' \triangleleft b \triangleright \delta \cdot \mathbf{0} \\ &= p. \end{aligned}$$

3. $p \equiv p' + p''$. By induction we have $p\text{CRL}_t \vdash \pi_t(\pi_{\text{tf}}(p')) = p'$ and $p\text{CRL}_t \vdash \pi_t(\pi_{\text{tf}}(p'')) = p''$. Then, $p\text{CRL}_t \vdash \pi_t(\pi_{\text{tf}}(p)) = \pi_t(\pi_{\text{tf}}(p' + p'')) = \pi_t(\pi_{\text{tf}}(p') + \pi_{\text{tf}}(p'')) = \pi_t(\pi_{\text{tf}}(p')) + \pi_t(\pi_{\text{tf}}(p'')) = p' + p'' = p$.

The following result indicates that the mapping π_t preserves derivability.

Lemma 4.2 For $p\text{CRL}$ -terms p and q we have

$$p\text{CRL} \vdash p = q \implies p\text{CRL}_t \vdash \pi_t(p) = \pi_t(q).$$

Proof. As the proof systems underlying $p\text{CRL}$ and $p\text{CRL}_t$ are identical, all we have to prove is that the π_t -image of every axiom of $p\text{CRL}$ is derivable from the axioms of $p\text{CRL}_t$. The axioms of $p\text{CRL}$ are given in Table 5.

This is easily established. Consider for example axiom A7. We must prove that $p\text{CRL}_t \vdash \pi_t(\delta \cdot x) = \pi_t(\delta)$. This is not difficult: $p\text{CRL}_t \vdash \pi_t(\delta \cdot x) = \pi_t(\delta) \cdot \pi_t(x) = \delta \cdot \mathbf{0} \cdot x \stackrel{\text{ATA3}}{=} (\delta \cdot x) \cdot \mathbf{0} \stackrel{\text{A7}}{=} \delta \cdot \mathbf{0} = \pi_t(\delta)$. As another example consider axiom Cond6:

$$\begin{aligned} p\text{CRL}_t \vdash \pi_t((x \triangleleft b \triangleright \delta) \cdot y) &= \pi_t(x \triangleleft b \triangleright \delta) \cdot \pi_t(y) \\ &= (\pi_t(x) \triangleleft b \triangleright \pi_t(\delta)) \cdot \pi_t(y) \\ &= (x \triangleleft b \triangleright \delta \cdot \mathbf{0}) \cdot y \stackrel{\text{C6}}{=} x \cdot y \triangleleft \delta \cdot \mathbf{0} \cdot y \\ &\stackrel{\text{ATA3}}{=} x \cdot y \triangleleft (\delta \cdot y) \cdot \mathbf{0} \\ &\stackrel{\text{A7}}{=} x \cdot y \triangleleft b \triangleright \delta \cdot \mathbf{0} \\ &= \pi_t(x) \cdot \pi_t(y) \triangleleft b \triangleright \pi_t(\delta) \\ &= \pi_t(x \cdot y) \triangleleft b \triangleright \pi_t(\delta) \\ &= \pi_t(x \cdot y \triangleleft b \triangleright \delta). \end{aligned}$$

A1	$x + y = y + x$
A2	$x + (y + z) = (x + y) + z$
A3	$x + x = x$
A4	$(x + y) \cdot z = x \cdot z + y \cdot z$
A5	$(x \cdot y) \cdot z = x \cdot (y \cdot z)$
A6	$x + \delta = x$
A7	$\delta \cdot x = \delta$
PE _a	$\mathbf{a}(\vec{x}) \triangleleft eq(\vec{x}, \vec{y}) \triangleright \delta = \mathbf{a}(\vec{y}) \triangleleft eq(\vec{x}, \vec{y}) \triangleright \delta$
Cond1	$x \triangleleft \mathbf{t} \triangleright y = x$
Cond2	$x \triangleleft \mathbf{f} \triangleright y = y$
Cond3	$x \triangleleft b \triangleright y = x \triangleleft b \triangleright \delta + y \triangleleft \neg b \triangleright \delta$
Cond4	$(x \triangleleft b_1 \triangleright \delta) \triangleleft b_2 \triangleright \delta = x \triangleleft b_1 \wedge b_2 \triangleright \delta$
Cond5	$x \triangleleft b_1 \triangleright \delta + x \triangleleft b_2 \triangleright \delta = x \triangleleft b_1 \vee b_2 \triangleright \delta$
Cond6	$(x \triangleleft b \triangleright \delta) \cdot y = x \cdot y \triangleleft b \triangleright \delta$
Cond7	$(x + y) \triangleleft b \triangleright \delta = x \triangleleft b \triangleright \delta + y \triangleleft b \triangleright \delta$
SCA	$(x \triangleleft b \triangleright \delta) \cdot (y \triangleleft b \triangleright \delta) = x \cdot y \triangleleft b \triangleright \delta$
Sum1	$\sum_v y = y$
Sum3	$\sum_v p = \sum_v p + p$
Sum4	$\sum_v (p + q) = \sum_v p + \sum_v q$
Sum5	$(\sum_v p) \cdot x = \sum_v p \cdot x$
Sum12	$(\sum_v p) \triangleleft b \triangleright \delta = \sum_v p \triangleleft b \triangleright \delta$

Table 5: Axioms of p CRL.

Theorem 4.4 (Relative completeness) With respect to strong timed bisimulation $p\text{CRL}_t$ is a complete axiom system for process-closed process terms under the assumptions that the data algebra is completely axiomatised and that the data algebra has built-in equality and Skolem functions.

Proof. Without loss of generality we may assume by Corollary 4.1 that p and q are deadlock-saturated well-timed basic terms. We use the completeness result of [10] for untimed $p\text{CRL}$. Suppose that $\llbracket p \rrbracket^\nu \Leftrightarrow_t \llbracket q \rrbracket^\nu$ for all valuations ν . Then, by Theorem 4.3, we have $\llbracket \pi_{\text{tf}}(p) \rrbracket^\nu \Leftrightarrow \llbracket \pi_{\text{tf}}(q) \rrbracket^\nu$ for all valuations ν . Here we use the completeness of $p\text{CRL}$ to obtain $p\text{CRL} \vdash \pi_{\text{tf}}(p) = \pi_{\text{tf}}(q)$. By Lemma 4.2 we thus have $p\text{CRL}_t \vdash \pi_t(\pi_{\text{tf}}(p)) = \pi_t(\pi_{\text{tf}}(q))$. Using Lemma 4.1 we have $p\text{CRL}_t \vdash p = q$.

5 The axiom system μCRL_t

One of the major strengths of timed μCRL is that it is possible to specify time-dependent, parallel processes with data. In this section, we incorporate operators for parallelism in the language and introduce the axiom system μCRL_t .

5.1 Axioms for parallelism

Concurrency is described by the binary operator \parallel . A process $p\parallel q$, the parallel execution of p and q , can first perform an action of p , first perform an action of q , or start with a communication, or synchronisation, between p and q . Hence, the first axiom for this operator is $x\parallel y = x \parallel y + y \parallel x + x \mid y$ (see axiom CM1 in Table 6), where the auxiliary *left merge* operators \parallel and the *communication merge* \mid are defined below. The process $p\parallel q$ exists at time t only if both p and q exist at time t .

The process $p \parallel\!\!\! \parallel q$ is as $p\parallel q$, but the first action that is performed comes from p . As was the case for parallel composition, the action can only be performed if the other party still exists at that time. For the axiomatisation of the left merge $\parallel\!\!\! \parallel$ the auxiliary *before*-operator is defined; $p \ll q$ should be interpreted as the part of process p that starts before q gets definitely disabled. Otherwise $p \ll q$ becomes a time deadlock at time t_0 . Although the operator differs slightly from the similarly named operator in [2], we have decided to give it the same symbol. A small example may facilitate the understanding of the “ \ll ”-operator. For example, the process represented by $(a\cdot 2 + b\cdot 4) \ll c\cdot 3$ cannot choose to execute action b at time 4 since its first action must be executed not later than time 3. Using the axioms we can derive that $(a\cdot 2 + b\cdot 4) \ll c\cdot 3 = a\cdot 2$.

The process $p \mid q$ also behaves as the process $p\parallel q$, except that the first action must be a communication between its left and right operand. Furthermore, these actions must occur at the same time. The action resulting from such a communication is defined by the binary, commutative and associative function γ , which is only defined on *action declarations*. In order for a communication to occur between action terms $\mathbf{a}(d), \mathbf{a}'(e) \in AT$, $\gamma(\mathbf{a}, \mathbf{a}')$ should be defined, and the data parameters d and e of the action terms should match according to axiom CF in Table 6.

The axioms for these operators are given in Table 6 and Table 7. These axioms are designed to eliminate the parallel operators in favour of the alternative and the sequential operator.

Sometimes we want to express that certain actions cannot happen, and must be blocked. Generally, this is only done when we want to force actions into a communication. The encapsulation operator ∂_H ($H \subseteq Act$) is specially designed for this task. In $\partial_H(p)$ it prevents all actions of which the action declaration is mentioned in H from happening by renaming the corresponding action terms into δ . Consider, for example, the process $a\parallel b = a\cdot b + b\cdot a + c$. Often, this is not quite what is desired, as the intention generally is that a and b do not happen separately. Therefore, the encapsulation operator can be used. The process represented by $\partial_{\{a,b\}}(a\parallel b)$ is equal to c .

The axioms of μCRL_t are the axioms of $p\text{CRL}_t$, combined with the axioms in the Tables 7 and 6. The signature $\Sigma(\mu\text{CRL}_t)$ is as $\Sigma(p\text{CRL}_t)$, extended with the operators for parallelism and the “ \ll ”-operator.

The various operators of $\Sigma(\mu\text{CRL}_t)$ are listed in order of decreasing binding strength:

$${}^c, \quad \partial_H, \quad \cdot, \quad \{\gg, \ll\}, \quad \{\triangleleft, \triangleright\}, \quad \parallel, \quad \underline{\parallel}, \quad | \}, \quad \sum, \quad +.$$

Parentheses are omitted from terms according to this convention.

CM1	$x \parallel y = x \underline{\parallel} y + y \underline{\parallel} x + x y$
CM2	$a^c t \underline{\parallel} x = (a^c t \ll x) \cdot x$
CM3	$a^c t \cdot x \underline{\parallel} y = (a^c t \ll y) \cdot (t \gg x \parallel y)$
CM4	$(x + y) \underline{\parallel} z = x \underline{\parallel} z + y \underline{\parallel} z$
SUM6	$(\sum_v p) \underline{\parallel} x = \sum_v (p \underline{\parallel} x)$
H18	$(x \triangleleft b \triangleright y) \underline{\parallel} z = (x \underline{\parallel} z) \triangleleft b \triangleright (y \underline{\parallel} z)$
CF	$\mathbf{a}(\vec{d}) \mathbf{a}'(\vec{e}) = \begin{cases} \gamma(\mathbf{a}, \mathbf{a}')(\vec{d}) \triangleleft eq(\vec{d}, \vec{e}) \triangleright \delta \\ \delta \quad \text{if } \gamma(\mathbf{a}, \mathbf{a}') \text{ defined} \\ \delta \quad \text{otherwise} \end{cases}$
CD1	$\delta a = \delta$
CD2	$a \delta = \delta$
CM5	$a \cdot x a' = (a a') \cdot x$
CM6	$a a' \cdot x = (a a') \cdot x$
CM7	$a \cdot x a' \cdot y = (a a') \cdot (x \parallel y)$
CM8	$(x + y) z = x z + y z$
CM9	$x (y + z) = x y + x z$
ATA7	$(x y)^c t = x^c t y$
ATA8	$(x y)^c t = x y^c t$
SUM7	$(\sum_v p) x = \sum_v (p x)$
SUM7'	$x (\sum_v p) = \sum_v (x p)$
H8	$x (y \triangleleft b \triangleright z) = (x y) \triangleleft b \triangleright (x z)$
H8'	$(x \triangleleft b \triangleright y) z = (x z) \triangleleft b \triangleright (y z)$

Table 6: Axioms for parallelism of μCRL_t , where $x, y, z \in V_P$, process-closed $p \in T_P$, $a, a' \in AT_\delta$, $\mathbf{a}, \mathbf{a}' \in Act$, $b \in V_B$, $t \in V_T$, $v \in V$ and $H \subseteq Act$.

Example 5.1. Data transfer between parallel components occurs very often. We describe as an example a simplified instance of data transfer. One process sends a natural number n via action \mathbf{s} , and another process reads it via action \mathbf{r} and then announces it via action \mathbf{a} . Using an encapsulation operator we force the actions \mathbf{s} and \mathbf{r} to communicate: $\gamma(\mathbf{s}, \mathbf{r}) = \mathbf{c}$.

$$p = \partial_{\{\mathbf{r}, \mathbf{s}\}}(\mathbf{s}(n) \parallel \sum_m \mathbf{r}(m) \cdot \mathbf{a}(m))$$

The process represented by p behaves the same as $\mathbf{c}(n) \cdot \mathbf{a}(n)$, so the n is transferred to action \mathbf{a} , as expected. \square

Processes that are put in parallel can have time constraints on their actions that are used for communication. It may happen that these constraints are incompatible. For instance, a process can be able to perform a read action r at time 2, whereas the corresponding send action occurs at time 3. This is expressed by $\partial_{\{\mathbf{s}, \mathbf{r}\}}(r^c 2 \parallel s^c 3)$. This process is equivalent to $\delta^c 2$, expressing that up till time

DD	$\partial_H(\delta) = \delta$	
D1	$\partial_H(\mathbf{a}(\vec{d})) = \mathbf{a}(\vec{d})$	if $\mathbf{a} \notin H$
D2	$\partial_H(\mathbf{a}(\vec{d})) = \delta$	if $\mathbf{a} \in H$
D3	$\partial_H(x + y) = \partial_H(x) + \partial_H(y)$	
D4	$\partial_H(x \cdot y) = \partial_H(x) \cdot \partial_H(y)$	
D5	$\partial_H(x \triangleleft b \triangleright y) = \partial_H(x) \triangleleft b \triangleright \partial_H(y)$	
D6	$\partial_H(\sum_v p) = \sum_v \partial_H(p)$	
D7	$\partial_H(x)^{\cdot t} = \partial_H(x^{\cdot t})$	
ATC1	$x \ll a^{\cdot t} = \sum_u x^{\cdot u} \triangleleft u \leq t \triangleright x^{\cdot t}$	
ATC2	$x \ll (y + z) = x \ll y + x \ll z$	
ATC3	$x \ll y \cdot z = x \ll y$	
ATC4	$x \ll \sum_v p = \sum_v x \ll p$	
ATC5	$x \ll (y \triangleleft b \triangleright z) = (x \ll y) \triangleleft b \triangleright (x \ll z)$	

Table 7: Time related axioms of μCRL_t , where $x, y, z \in V_P$, process-closed $p \in T_P$, $\mathbf{a} \in \text{Act}$, $a \in AT_\delta$, $b \in V_B$, $t, u \in V_T$, $v \in V$ and $H \subseteq \text{Act}$.

2, the process can go on without violating any timing constraint, but to reach any moment in time larger than 2, a timing constraint of an individual process must be ignored. Hence, this process does not exist at time 3.

5.2 Elimination of parallel operators

In this section, we consider processes with parallel operators, and show that all process-closed terms are derivably equal to $\Sigma(p\text{CRL}_t)$ -terms. In [16], the elimination of the parallel operators has been provided for the version of timed μCRL_t , as presented in [9]. Although this version of μCRL_t differs from the version presented here the results are similar.

Lemma 5.1 Let p and q be basic terms. Then, $\partial_H(p)$ and $p \ll q$ are derivably equal to basic terms.

Proof. First, we prove the existence of a basic term that is derivably equal to $\partial_H(p)$ by induction on the structure of basic term p .

1. $p \equiv \sum_{\vec{v}} a^{\cdot t} \triangleleft b \triangleright \delta \cdot \mathbf{0}$. Then, $\partial_H(p) = \partial_H(\sum_{\vec{v}} a^{\cdot t} \triangleleft b \triangleright \delta \cdot \mathbf{0}) \stackrel{\text{D6}}{=} \sum_{\vec{v}} \partial_H(a^{\cdot t} \triangleleft b \triangleright \delta \cdot \mathbf{0}) \stackrel{\text{D5}}{=} \sum_{\vec{v}} \partial_H(a^{\cdot t}) \triangleleft b \triangleright \partial_H(\delta \cdot \mathbf{0}) \stackrel{\text{D7}}{=} \sum_{\vec{v}} \partial_H(a)^{\cdot t} \triangleleft b \triangleright \partial_H(\delta) \cdot \mathbf{0} \stackrel{\text{DD}}{=} \sum_{\vec{v}} \partial_H(a)^{\cdot t} \triangleleft b \triangleright \delta \cdot \mathbf{0}$. Since $\partial_H(a)$ is an action term or δ we have obtained a process-closed $p\text{CRL}_t$ -term. Hence a basic term exists for $\partial_H(p)$.
2. $p \equiv \sum_{\vec{v}} a^{\cdot t} \cdot p' \triangleleft b \triangleright \delta \cdot \mathbf{0}$. By induction we have the existence of basic term r' such that $\partial_H(p') = r'$. Then, $\partial_H(p) = \partial_H(\sum_{\vec{v}} a^{\cdot t} \cdot p' \triangleleft b \triangleright \delta \cdot \mathbf{0}) \stackrel{\text{D6}}{=} \sum_{\vec{v}} \partial_H(a^{\cdot t} \cdot p' \triangleleft b \triangleright \delta \cdot \mathbf{0}) \stackrel{\text{D5}}{=} \sum_{\vec{v}} \partial_H(a^{\cdot t} \cdot p') \triangleleft b \triangleright \partial_H(\delta \cdot \mathbf{0}) \stackrel{\text{D4}}{=} \sum_{\vec{v}} \partial_H(a^{\cdot t}) \cdot \partial_H(p') \triangleleft b \triangleright \partial_H(\delta \cdot \mathbf{0}) \stackrel{\text{D7}}{=} \sum_{\vec{v}} \partial_H(a)^{\cdot t} \cdot \partial_H(p') \triangleleft b \triangleright \partial_H(\delta) \cdot \mathbf{0} \stackrel{\text{DD}}{=} \sum_{\vec{v}} \partial_H(a)^{\cdot t} \cdot r' \triangleleft b \triangleright \delta \cdot \mathbf{0}$. Since $\partial_H(a)$ is an action term or δ , we have obtained a process-closed $p\text{CRL}_t$ -term. Hence, a basic term exists.
3. $p \equiv p_1 + p_2$. By induction we have the existence of basic terms p'_1 and p'_2 such that $\partial_H(p_1) = p'_1$ and $\partial_H(p_2) = p'_2$. Then, $\partial_H(p) = \partial_H(p_1 + p_2) \stackrel{\text{D3}}{=} \partial_H(p_1) + \partial_H(p_2) = p'_1 + p'_2$, which is a basic term.

Second, we prove the existence of basic term that is derivably equal to $p \ll q$ by induction on the structure of basic term q .

1. $q \equiv \sum_{\bar{v}} a^t \triangleleft b \triangleright \delta \cdot \mathbf{0}$. Then, $p \ll q = p \ll \sum_{\bar{v}} a^t \triangleleft b \triangleright \delta \cdot \mathbf{0} \stackrel{\text{ATC4}}{=} \sum_{\bar{v}} p \ll (a^t \triangleleft b \triangleright \delta \cdot \mathbf{0}) \stackrel{\text{ATC5}}{=} \sum_{\bar{v}} (p \ll a^t) \triangleleft b \triangleright (p \ll \delta \cdot \mathbf{0}) \stackrel{\text{ATC1}}{=} \sum_{\bar{v}} (\sum_u p^c u \triangleleft u \leq t \triangleright p^t) \triangleleft b \triangleright (\sum_u p^c u \triangleleft u \leq \mathbf{0} \triangleright p \cdot \mathbf{0})$. This is a process-closed $p\text{CRL}_t$ -term, and hence it is derivably equal to a basic term.
2. $q \equiv \sum_{\bar{v}} a^t \cdot q' \triangleleft b \triangleright \delta \cdot \mathbf{0}$. Then, $p \ll q = p \ll \sum_{\bar{v}} a^t \cdot q' \triangleleft b \triangleright \delta \cdot \mathbf{0} \stackrel{\text{ATC4}}{=} \sum_{\bar{v}} p \ll (a^t \cdot q' \triangleleft b \triangleright \delta \cdot \mathbf{0}) \stackrel{\text{ATC5}}{=} \sum_{\bar{v}} (p \ll a^t \cdot q') \triangleleft b \triangleright (p \ll \delta \cdot \mathbf{0}) \stackrel{\text{ATC3}}{=} \sum_{\bar{v}} (p \ll a^t) \triangleleft b \triangleright (p \ll \delta \cdot \mathbf{0}) \stackrel{\text{ATC1}}{=} \sum_{\bar{v}} (\sum_u p^c u \triangleleft u \leq t \triangleright p^t) \triangleleft b \triangleright (\sum_u p^c u \triangleleft u \leq \mathbf{0} \triangleright p \cdot \mathbf{0})$. This is a process-closed $p\text{CRL}_t$ -term, and hence it is derivably equal to a basic term.
3. $q \equiv q_1 + q_2$. By induction we have the existence of basic terms p_1 and p_2 such that $p \ll q_1 = p_1$ and $p \ll q_2 = p_2$. Then, $p \ll q = p \ll (q_1 + q_2) \stackrel{\text{ATC2}}{=} p \ll q_1 + p \ll q_2 = p_1 + p_2$, which is a basic term.

Lemma 5.2 Let p and q be well-timed basic terms. Then, $p \ll q$, $p | q$, and $p \parallel q$ are derivably equal to a basic term.

Proof. The three statements are proven simultaneously by induction on the total number of symbols of well-timed basic terms p and q . First, we consider the term $p \ll q$. We distinguish three cases based on the structure of basic term p :

1. $p \equiv \sum_{\bar{v}} a^t \triangleleft b \triangleright \delta \cdot \mathbf{0}$. Then, $p \ll q = (\sum_{\bar{v}} a^t \triangleleft b \triangleright \delta \cdot \mathbf{0}) \ll q \stackrel{\text{SUM6}}{=} \sum_{\bar{v}} (a^t \triangleleft b \triangleright \delta \cdot \mathbf{0}) \ll q \stackrel{\text{H18}}{=} \sum_{\bar{v}} (a^t \ll q) \triangleleft b \triangleright (\delta \cdot \mathbf{0} \ll q) \stackrel{\text{CM2}}{=} \sum_{\bar{v}} (a^t \ll q) \cdot q \triangleleft b \triangleright (\delta \cdot \mathbf{0} \ll q) \cdot q$. This is a μCRL_t -term in which none of the operators \ll , $|$, or \parallel occurs, and hence it is derivably equal to a basic term using the previous lemma and Theorem 2.1.
2. $p \equiv \sum_{\bar{v}} a^t \cdot p' \triangleleft b \triangleright \delta \cdot \mathbf{0}$. By induction we have the existence of basic term r' such that $p' \ll q = r'$. As p is well-timed we have $t \gg p' = p'$. Then, $p \ll q = (\sum_{\bar{v}} a^t \cdot p' \triangleleft b \triangleright \delta \cdot \mathbf{0}) \ll q \stackrel{\text{SUM6}}{=} \sum_{\bar{v}} (a^t \cdot p' \triangleleft b \triangleright \delta \cdot \mathbf{0}) \ll q \stackrel{\text{H18}}{=} \sum_{\bar{v}} (a^t \cdot p' \ll q) \triangleleft b \triangleright (\delta \cdot \mathbf{0} \ll q) \stackrel{\text{CM3,CM2}}{=} \sum_{\bar{v}} (a^t \ll q) \cdot (t \gg p' \ll q) \triangleleft b \triangleright (\delta \cdot \mathbf{0} \ll q) \cdot q = \sum_{\bar{v}} (a^t \ll q) \cdot (p' \ll q) \triangleleft b \triangleright (\delta \cdot \mathbf{0} \ll q) \cdot q = \sum_{\bar{v}} (a^t \ll q) \cdot r' \triangleleft b \triangleright (\delta \cdot \mathbf{0} \ll q) \cdot q$. This is a μCRL_t -term in which none of the operators \ll , $|$, or \parallel occurs, and hence it is derivably equal to a basic term using the previous lemma and Theorem 2.1.
3. $p \equiv p' + p''$. By induction we have the existence of basic terms r' and r'' such that $p' \ll q = r'$ and $p'' \ll q = r''$. Then, $p \ll q = (p' + p'') \ll q \stackrel{\text{CM4}}{=} p' \ll q + p'' \ll q = r' + r''$, which is a basic term.

Second, we consider the term $p | q$. We distinguish cases based on the structure of the well-timed basic terms p and q :

1. $p \equiv \sum_{\bar{v}} a^t \triangleleft b \triangleright \delta \cdot \mathbf{0}$ and $q \equiv \sum_{\bar{w}} a'^t \triangleleft b' \triangleright \delta \cdot \mathbf{0}$. Then, $p | q = (\sum_{\bar{v}} a^t \triangleleft b \triangleright \delta \cdot \mathbf{0}) | (\sum_{\bar{w}} a'^t \triangleleft b' \triangleright \delta \cdot \mathbf{0}) \stackrel{\text{SUM7, SUM7'}}{=} \sum_{\bar{v}, \bar{w}} (a^t \triangleleft b \triangleright \delta \cdot \mathbf{0}) | (a'^t \triangleleft b' \triangleright \delta \cdot \mathbf{0}) \stackrel{\text{H8}}{=} \sum_{\bar{v}, \bar{w}} ((a^t \triangleleft b \triangleright \delta \cdot \mathbf{0}) | a'^t) \triangleleft b' \triangleright ((a^t \triangleleft b \triangleright \delta \cdot \mathbf{0}) | \delta \cdot \mathbf{0}) \stackrel{\text{H8'}}{=} \sum_{\bar{v}, \bar{w}} ((a^t | a'^t) \triangleleft b \triangleright (\delta \cdot \mathbf{0} | a'^t)) \triangleleft b' \triangleright ((a^t | \delta \cdot \mathbf{0}) \triangleleft b \triangleright (\delta \cdot \mathbf{0} | \delta \cdot \mathbf{0})) \stackrel{\text{ATA7, ATA8}}{=} \sum_{\bar{v}, \bar{w}} ((a | a')^t \triangleleft b \triangleright (\delta | a') \cdot \mathbf{0} \triangleleft b' \triangleright ((a | \delta)^t \triangleleft b \triangleright (\delta | \delta) \cdot \mathbf{0} \cdot \mathbf{0})) \stackrel{\text{CD1, CD2}}{=} \sum_{\bar{v}, \bar{w}} ((a | a')^t \triangleleft b \triangleright \delta \cdot \mathbf{0} \triangleleft b' \triangleright (\delta \cdot \mathbf{0} \triangleleft b \triangleright \delta \cdot \mathbf{0} \cdot \mathbf{0}))$. By axiom CF the process $(a | a')$ is equal to an atomic action or δ . Hence, all parallel operators can be eliminated from the term $p | q$. By Theorem 2.1 there exists a basic term r such that $p | q = r$.

2. $p \equiv \sum_{\bar{v}} a^t \triangleleft b \triangleright \delta \cdot \mathbf{0}$ and $q = \sum_{\bar{w}} a'^t \cdot q' \triangleleft b' \triangleright \delta \cdot \mathbf{0}$. Then, $p | q = (\sum_{\bar{v}} a^t \triangleleft b \triangleright \delta \cdot \mathbf{0}) | (\sum_{\bar{w}} a'^t \cdot q' \triangleleft b' \triangleright \delta \cdot \mathbf{0}) \stackrel{\text{SUM7}, \text{SUM7}'}{=} \sum_{\bar{v}, \bar{w}} (a^t \triangleleft b \triangleright \delta \cdot \mathbf{0}) | (a'^t \cdot q' \triangleleft b' \triangleright \delta \cdot \mathbf{0}) \stackrel{\text{H8}}{=} \sum_{\bar{v}, \bar{w}} ((a^t \triangleleft b \triangleright \delta \cdot \mathbf{0}) | a'^t \cdot q') \triangleleft b' \triangleright ((a^t \triangleleft b \triangleright \delta \cdot \mathbf{0}) | \delta \cdot \mathbf{0}) \stackrel{\text{H8}'}{=} \sum_{\bar{v}, \bar{w}} ((a^t | a'^t \cdot q') \triangleleft b \triangleright (\delta \cdot \mathbf{0} | a'^t \cdot q')) \triangleleft b' \triangleright ((a^t | \delta \cdot \mathbf{0}) \triangleleft b \triangleright (\delta \cdot \mathbf{0} | \delta \cdot \mathbf{0})) \stackrel{\text{ATA3}, \text{ATA7}, \text{ATA8}}{=} \sum_{\bar{v}, \bar{w}} ((a | a' \cdot q') \triangleleft b \triangleright (\delta | a' \cdot q') \cdot \mathbf{0} \triangleleft b' \triangleright ((a | \delta) \triangleleft b \triangleright (\delta | \delta) \cdot \mathbf{0} \cdot \mathbf{0})) \stackrel{\text{CM6}, \text{CD1}, \text{CD2}, \text{A7}}{=} \sum_{\bar{v}, \bar{w}} ((a | a' \cdot q') \triangleleft b \triangleright \delta \cdot \mathbf{0} \triangleleft b' \triangleright (\delta \cdot \mathbf{0} \triangleleft b \triangleright \delta \cdot \mathbf{0} \cdot \mathbf{0})) \stackrel{\text{CM6}}{=} \sum_{\bar{v}, \bar{w}} ((a | a') \cdot q') \triangleleft b \triangleright \delta \cdot \mathbf{0} \triangleleft b' \triangleright (\delta \cdot \mathbf{0} \triangleleft b \triangleright \delta \cdot \mathbf{0} \cdot \mathbf{0})$. By axiom CF the process $(a | a')$ is equal to an atomic action or δ . Hence, all parallel operators can be eliminated from the term $p | q$. By Theorem 2.1 there exists a basic term r such that $p | q = r$.
3. $p \equiv \sum_{\bar{v}} a^t \cdot p' \triangleleft b \triangleright \delta \cdot \mathbf{0}$ and $q = \sum_{\bar{w}} a'^t \cdot q' \triangleleft b' \triangleright \delta \cdot \mathbf{0}$. This case is similar to the previous case.
4. $p \equiv \sum_{\bar{v}} a^t \cdot p' \triangleleft b \triangleright \delta \cdot \mathbf{0}$ and $q = \sum_{\bar{w}} a'^t \cdot q' \triangleleft b' \triangleright \delta \cdot \mathbf{0}$. By induction we have the existence of a basic term r' such that $p' || q' = r'$. Then, $p | q = (\sum_{\bar{v}} a^t \cdot p' \triangleleft b \triangleright \delta \cdot \mathbf{0}) | (\sum_{\bar{w}} a'^t \cdot q' \triangleleft b' \triangleright \delta \cdot \mathbf{0}) \stackrel{\text{SUM7}, \text{SUM7}'}{=} \sum_{\bar{v}, \bar{w}} (a^t \cdot p' \triangleleft b \triangleright \delta \cdot \mathbf{0}) | (a'^t \cdot q' \triangleleft b' \triangleright \delta \cdot \mathbf{0}) \stackrel{\text{H8}}{=} \sum_{\bar{v}, \bar{w}} ((a^t \cdot p' \triangleleft b \triangleright \delta \cdot \mathbf{0}) | a'^t \cdot q') \triangleleft b' \triangleright ((a^t \cdot p' \triangleleft b \triangleright \delta \cdot \mathbf{0}) | \delta \cdot \mathbf{0}) \stackrel{\text{H8}'}{=} \sum_{\bar{v}, \bar{w}} ((a^t \cdot p' | a'^t \cdot q') \triangleleft b \triangleright (\delta \cdot \mathbf{0} | a'^t \cdot q')) \triangleleft b' \triangleright ((a^t \cdot p' | \delta \cdot \mathbf{0}) \triangleleft b \triangleright (\delta \cdot \mathbf{0} | \delta \cdot \mathbf{0})) \stackrel{\text{ATA3}, \text{ATA7}, \text{ATA8}}{=} \sum_{\bar{v}, \bar{w}} ((a \cdot p' | a' \cdot q') \triangleleft b \triangleright (\delta | a' \cdot q') \cdot \mathbf{0} \triangleleft b' \triangleright ((a \cdot p' | \delta) \triangleleft b \triangleright (\delta | \delta) \cdot \mathbf{0} \cdot \mathbf{0})) \stackrel{\text{CM5}, \text{CM6}, \text{CM7}}{=} \sum_{\bar{v}, \bar{w}} ((a | a') \cdot (p' || q')) \triangleleft b \triangleright ((\delta | a') \cdot q') \triangleleft b' \triangleright (((a | \delta) \cdot p') \triangleleft b \triangleright (\delta | \delta) \cdot \mathbf{0} \cdot \mathbf{0})) \stackrel{\text{CD1}, \text{CD2}, \text{A7}}{=} \sum_{\bar{v}, \bar{w}} (((a | a') \cdot (p' || q')) \triangleleft b \triangleright \delta \cdot \mathbf{0} \triangleleft b' \triangleright (\delta \cdot \mathbf{0} \triangleleft b \triangleright \delta \cdot \mathbf{0} \cdot \mathbf{0})) = \sum_{\bar{v}, \bar{w}} (((a | a') \cdot r') \triangleleft b \triangleright \delta \cdot \mathbf{0} \triangleleft b' \triangleright (\delta \cdot \mathbf{0} \triangleleft b \triangleright \delta \cdot \mathbf{0} \cdot \mathbf{0}))$. By axiom CF the process $(a | a')$ is equal to an atomic action or δ . Hence, all parallel operators can be eliminated from the term $p | q$. By Theorem 2.1 there exists a basic term r such that $p | q = r$.
5. $p \equiv p' + p''$. By induction we have the existence of basic terms r' and r'' such that $p' | q = r'$ and $p'' | q = r''$. Then, $p | q = (p' + p'') | q \stackrel{\text{CM8}}{=} p' | q + p'' | q = r' + r''$, which is a basic term.
6. $q \equiv q' + q''$. Similar to case 5.

Finally, we base the elimination of $||$ on the elimination of \lll and $|$ as follows: By the previous parts of the proof we have the existence of basic terms r_1 , r_2 , and r_3 such that $p \lll q = r_1$, $q \lll p = r_2$, and $p | q = r_3$. Hence, $p || q = p \lll q + q \lll p + p | q = r_1 + r_2 + r_3$, which is a basic term.

Theorem 5.1 (Elimination theorem) If q is a process-closed term over $\Sigma(\mu\text{CRL}_t)$, then there is a basic term p such that $\mu\text{CRL}_t \vdash q = p$.

Proof. By Lemma 5.1 and Lemma 5.2 any process-closed term q over $\Sigma(\mu\text{CRL}_t)$ with at most one operator from $\{\lll, \partial_H, ||, \lll, |\}$ is derivably equal to some basic term. Obviously, any term with $n + 1$ operators from this set contains subterms with only one such operator, such that after elimination of one of these, a term with n parallel operators results. By induction on n we conclude that all these operators can be eliminated, so that q is derivably equal to some basic term p .

5.3 Semantics of timed μCRL

Definition 5.1 The set $\mathcal{P} = \mathcal{P}^\omega$ of processes is obtained by the following recursion:

$$\begin{aligned}
\mathcal{P}^0 &= \mathcal{A}_\delta \\
\mathcal{P}^{n+1} &= \mathcal{P}^n \\
&\cup \{ p \cdot q, \sum P', p^t, t \gg p, p || q, p \lll q, p | q, p \lll q, \partial_H(p) \\
&\quad | p, q \in \mathcal{P}^n, P' \neq \emptyset, P' \subseteq \mathcal{P}^n, t \in \mathcal{D}_T, H \subseteq \text{Act} \\
&\quad \}.
\end{aligned}$$

$\frac{p \xrightarrow{a} t \checkmark U_t(q)}{p \ q \xrightarrow{a} t t \gg q}$	$\frac{q \xrightarrow{a} t \checkmark U_t(p)}{p \ q \xrightarrow{a} t t \gg p}$	$\frac{p \xrightarrow{a} t \checkmark U_t(q)}{p \ll q \xrightarrow{a} t t \gg q}$	$\frac{p \xrightarrow{a} t \checkmark U_t(q)}{p \ll q \xrightarrow{a} t \checkmark}$
$\frac{p \xrightarrow{a} t p' U_t(q)}{p \ q \xrightarrow{a} t p' \ t \gg q}$	$\frac{q \xrightarrow{a} t q' U_t(p)}{p \ q \xrightarrow{a} t t \gg p \ q'}$	$\frac{p \xrightarrow{a} t p' U_t(q)}{p \ll q \xrightarrow{a} t p' \ t \gg q}$	$\frac{p \xrightarrow{a} t p' U_t(q)}{p \ll q \xrightarrow{a} t p'}$
$\frac{p \xrightarrow{a(\vec{d})} t \checkmark \quad q \xrightarrow{b(\vec{d})} t \checkmark \quad \gamma(\mathbf{a}, \mathbf{b}) = \mathbf{c}}{p \ q \xrightarrow{c(\vec{d})} t \checkmark}$		$\frac{p \xrightarrow{a(\vec{d})} t \checkmark \quad q \xrightarrow{b(\vec{d})} t q' \quad \gamma(\mathbf{a}, \mathbf{b}) = \mathbf{c}}{p \ q \xrightarrow{c(\vec{d})} t q'}$	
$\frac{p \xrightarrow{a(\vec{d})} t p' \quad q \xrightarrow{b(\vec{d})} t \checkmark \quad \gamma(\mathbf{a}, \mathbf{b}) = \mathbf{c}}{p \ q \xrightarrow{c(\vec{d})} t p'}$		$\frac{p \xrightarrow{a(\vec{d})} t p' \quad q \xrightarrow{b(\vec{d})} t q' \quad \gamma(\mathbf{a}, \mathbf{b}) = \mathbf{c}}{p \ q \xrightarrow{c(\vec{d})} t p' \ q'}$	
$\frac{p \xrightarrow{a(\vec{d})} t \checkmark \quad q \xrightarrow{b(\vec{d})} t \checkmark \quad \gamma(\mathbf{a}, \mathbf{b}) = \mathbf{c}}{p q \xrightarrow{c(\vec{d})} t \checkmark}$		$\frac{p \xrightarrow{a(\vec{d})} t \checkmark \quad q \xrightarrow{b(\vec{d})} t q' \quad \gamma(\mathbf{a}, \mathbf{b}) = \mathbf{c}}{p q \xrightarrow{c(\vec{d})} t q'}$	
$\frac{p \xrightarrow{a(\vec{d})} t p' \quad q \xrightarrow{b(\vec{d})} t \checkmark \quad \gamma(\mathbf{a}, \mathbf{b}) = \mathbf{c}}{p q \xrightarrow{c(\vec{d})} t p'}$		$\frac{p \xrightarrow{a(\vec{d})} t p' \quad q \xrightarrow{b(\vec{d})} t q' \quad \gamma(\mathbf{a}, \mathbf{b}) = \mathbf{c}}{p q \xrightarrow{c(\vec{d})} t p' \ q'}$	

Table 8: Transition rules for the parallel operators, where $a \in \mathcal{A}$, $\mathbf{a}, \mathbf{b}, \mathbf{c} \in Act$, $\vec{d} \in \mathcal{D}_s$, $p, q, p', q' \in \mathcal{P}$, and $t \in \mathcal{D}_T$.

Definition 5.2 The interpretation of process-closed terms under a valuation, from Definition 3.2, is extended with the following clauses: for process-closed $p, q \in TP$, $H \subseteq Act$, and valuation ν

$$\begin{aligned}
\llbracket p \ll q \rrbracket^\nu &= \llbracket p \rrbracket^\nu \ll \llbracket q \rrbracket^\nu, \\
\llbracket p \| q \rrbracket^\nu &= \llbracket p \rrbracket^\nu \| \llbracket q \rrbracket^\nu, \\
\llbracket p \ll q \rrbracket^\nu &= \llbracket p \rrbracket^\nu \ll \llbracket q \rrbracket^\nu, \\
\llbracket p | q \rrbracket^\nu &= \llbracket p \rrbracket^\nu | \llbracket q \rrbracket^\nu, \\
\llbracket \partial_H(p) \rrbracket^\nu &= \partial_H(\llbracket p \rrbracket^\nu).
\end{aligned}$$

The operational semantics of these processes is described in terms of the action relations $\xrightarrow{a} t \checkmark$ and $\xrightarrow{a} t$ and the delay relation U_t . For the new operators the definitions of these relations are extended with the transition rules in the Tables 8, 9, and 10.

Although the set of processes that we consider in this section is larger than the set of processes we considered for $pCRL_t$ we have no reason to change our definition of strong timed bisimilarity as it is defined in terms of the action relations and the delay predicate only.

Theorem 5.2 (Congruence) Strong timed bisimilarity is a congruence with respect to the operators defined on \mathcal{P} .

$\frac{p \xrightarrow{a(\vec{d})}_t \checkmark \quad \mathbf{a} \notin H}{\partial_H(p) \xrightarrow{a(\vec{d})}_t \checkmark} \quad \frac{p \xrightarrow{a(\vec{d})}_t q \quad \mathbf{a} \notin H}{\partial_H(p) \xrightarrow{a(\vec{d})}_t \partial_H(q)}$

Table 9: Transition rules for encapsulation, where $p, q \in \mathcal{P}$, $t \in \mathcal{D}_T$, $H \subseteq \text{Act}$, $\mathbf{a} \in \text{Act}$, and $\vec{d} \in \mathcal{D}_s$.

$\frac{U_t(p) \quad U_t(q)}{U_t(p \ll q)} \quad \frac{U_t(p) \quad U_t(q)}{U_t(p \ q)} \quad \frac{U_t(p) \quad U_t(q)}{U_t(p \parallel q)} \quad \frac{U_t(p) \quad U_t(q)}{U_t(p q)} \quad \frac{U_t(p)}{U_t(\partial_H(p))}$
--

Table 10: Transition rules for the delay relation, where $p, q \in \mathcal{P}$, $t \in \mathcal{D}_T$, and $H \subseteq \text{Act}$.

Proof. We give bisimulations that witness the congruence of strong timed bisimilarity with respect to the operators \ll , ∂_H , \parallel , \parallel , and $|$. In each of these cases it is straightforward to prove that the constructed set R is a strong timed bisimulation.

1. \ll . Let $R_1 : p_1 \Leftrightarrow_t q_1$ and $R_2 : p_2 \Leftrightarrow_t q_2$. Define $R = \{(p_1 \ll p_2, q_1 \ll q_2), (q_1 \ll q_2, p_1 \ll p_2)\} \cup R_2$.
2. ∂_H . Let $R_1 : p \Leftrightarrow_t q$. Define $R = \{(\partial_H(p'), \partial_H(q')) \mid (p', q') \in R_1, H \subseteq \text{Act}\}$.
3. \parallel . Let $R_1 : p_1 \Leftrightarrow_t q_1$ and $R_2 : p_2 \Leftrightarrow_t q_2$. Define $R_{12} = R_1 \cup R_2$. Define $R'_{12} = R_{12} \cup \{(t \gg p', t \gg q') \mid (p', q') \in R_{12}, t \in \mathcal{D}_T\}$. Define $R = R'_{12} \cup \{(p' \parallel p'', q' \parallel q'') \mid (p', q'), (p'', q'') \in R'_{12}\}$.
4. \parallel . Let $R_1 : p_1 \Leftrightarrow_t q_1$ and $R_2 : p_2 \Leftrightarrow_t q_2$. Define $R_{12} = R_1 \cup R_2$ and $R'_{12} = R_{12} \cup \{(t \gg p', t \gg q') \mid (p', q') \in R_{12}, t \in \mathcal{D}_T\}$. Define $R = R'_{12} \cup \{(p_1 \parallel p_2, q_1 \parallel q_2), (q_1 \parallel q_2, p_1 \parallel p_2)\} \cup \{(p' \parallel p'', q' \parallel q'') \mid (p', q'), (p'', q'') \in R'_{12}\}$.
5. $|$. Let $R_1 : p_1 \Leftrightarrow_t q_1$ and $R_2 : p_2 \Leftrightarrow_t q_2$. Define $R_{12} = R_1 \cup R_2$ and $R'_{12} = R_{12} \cup \{(t \gg p', t \gg q') \mid (p', q') \in R_{12}, t \in \mathcal{D}_T\}$. Define $R = R'_{12} \cup \{(p' | p'', q' | q''), (p' \parallel p'', q' \parallel q'') \mid (p', q'), (p'', q'') \in R'_{12}\}$.

5.4 Soundness and completeness

Theorem 5.3 (Soundness) With respect to strong timed bisimilarity, μCRL_t is a sound axiom system.

Theorem 5.4 (Relative completeness) With respect to strong timed bisimilarity, μCRL_t is a complete axiom system for process-closed terms under the assumptions that the data algebra is completely axiomatised and that the data algebra has built-in equality and Skolem functions.

Proof. Consider arbitrary process-closed μCRL_t -terms p and q . Suppose that $\llbracket p \rrbracket^\nu \Leftrightarrow_t \llbracket q \rrbracket^\nu$ for all valuations ν . By the elimination theorem we have the existence of basic terms p' and q' such that $\mu\text{CRL}_t \vdash p = p'$ and $\mu\text{CRL}_t \vdash q = q'$. By the soundness of the axioms we then also have $\llbracket p' \rrbracket^\nu \Leftrightarrow_t \llbracket q' \rrbracket^\nu$ for all valuations ν . By the completeness of $p\text{CRL}_t$ we then have $p\text{CRL}_t \vdash p' = q'$. Using the fact that all axioms of $p\text{CRL}_t$ are also axioms of μCRL_t we find $\mu\text{CRL}_t \vdash p' = q'$. Thus we have $\mu\text{CRL}_t \vdash p = p' = q' = q$.

References

- [1] ALUR, R., AND DILL, D. Automata for modelling real-time systems. In *Proceedings of the International Colloquium on Algorithms, Languages and Programming* (1990), vol. 443 of *Lecture Notes in Computer Science*, pp. 322–335.
- [2] BAETEN, J., AND BERGSTRA, J. Real time process algebra. *Formal Aspects of Computing* 3, 2 (1991), 142–188.
- [3] BAETEN, J., AND BERGSTRA, J. Discrete time process algebra. *Formal Aspects of Computing* 8, 2 (1996), 188–208.
- [4] BAETEN, J., AND MIDDELBURG, C. Process algebra with timing: real time and discrete time. Tech. Rep. CSR 99-11, Eindhoven University of Technology, Department of Computing Science, 1999. To appear in: *Handbook of Process Algebra*, editors, J.A. Bergstra, A. Ponse, S.A. Smolka, Elsevier 2000.
- [5] BAETEN, J., AND WEIJLAND, W. *Process Algebra*, vol. 18 of *Cambridge Tracts in Theoretical Computer Science*. Cambridge University Press, 1990.
- [6] BEZEM, M., AND GROOTE, J. Invariants in process algebra with data. In *CONCUR'94: Concurrency Theory* (Uppsala, Sweden, 1994), B. Jonsson and J. Parrow, Eds., vol. 836 of *Lecture Notes in Computer Science*, Springer-Verlag, pp. 401–416.
- [7] BRINKSMA, E. *On the design of extended LOTOS: a specification language for open distributed systems*. PhD thesis, Twente University, 1988.
- [8] FOKKINK, W. *Clock, Trees and Stars in Process Theory*. PhD thesis, University of Amsterdam, 1994.
- [9] GROOTE, J. The syntax and semantics of timed μ CRL. Tech. Rep. SEN-R9709, CWI, Amsterdam, 1997.
- [10] GROOTE, J., AND LUTTIK, S. Undecidability and completeness results for process algebras with alternative quantification over data. Tech. Rep. SEN-R9806, CWI, Amsterdam, 1998.
- [11] GROOTE, J., MONIN, F., AND VAN DE POL, J. Checking verifications of protocols and distributed systems by computer. In *Proceedings CONCUR'98* (1998), D. Sangiorgi and R. de Simone, Eds., vol. 1466 of *Lecture Notes in Computer Science*, Sophia Antipolis, Springer-Verlag, pp. 629–655.
- [12] GROOTE, J., AND PONSE, A. The syntax and semantics of μ CRL. In *Algebra of Communicating Processes, Utrecht 1994* (1995), A. Ponse, C. Verhoef, and S. v. Vlijmen, Eds., Workshops in Computing, Springer-Verlag, pp. 26–62.
- [13] GROOTE, J., AND SELLINK, M. Confluence for process verification. *Theoretical Computer Science* 170, 1-2 (1996), 47–81.
- [14] GROOTE, J., AND SPRINGINTVELD, J. Focus points and convergent process operators. A proof strategy for protocol verification. Tech. Rep. 142, University Utrecht, Department of Philosophy, 1995.
- [15] GROOTE, J., AND VAN WAMEL, J. Analysis of three hybrid systems in timed μ CRL. Tech. Rep. SEN-R9815, CWI, Amsterdam, 1998. To appear in *Science of Computer Programming*.
- [16] GROOTE, J., AND VAN WAMEL, J. Basic theorems for parallel processes in timed μ CRL. Tech. Rep. SEN-R9808, CWI, Amsterdam, 1998.

- [17] HENNESSY, M., AND LIN, H. Symbolic bisimulations. *Theoretical Computer Science* 138, 2 (1995), 353–389.
- [18] HENZINGER, T., HO, P., AND WONG-TOI, H. Hytech: A model checker for hybrid systems. *International Journal on Software Tools for Technology Transfer* 1, 1-2 (1997), 110–122.
- [19] HOARE, C. *Communicating Sequential Processes*. International Series in Computer Science. Prentice-Hall International, 1985.
- [20] ITU-T. *Recommendation Z.100: Specification and Description Language (SDL)*. ITU-T, Geneva, June 1994.
- [21] KLUSENER, A. *Models and Axioms for a Fragment of Real Time Process Algebra*. PhD thesis, Eindhoven University of Technology, 1993.
- [22] LARSEN, K., PETTERSSON, P., AND YI, W. Uppaal in a nutshell. *International Journal on Software Tools for Technology Transfer* 1, 1-2 (1997), 134–152.
- [23] LYNCH, N., SEGALA, R., VAANDRAGER, F., AND WEINBERG, H. Hybrid I/O automata. In *Hybrid Systems III: Verification and Control (DIMACS/SYCON Workshop on Verification and Control of Hybrid Systems, New Brunswick, New Jersey, October 1995)* (1996), R. Alur, T. Henzinger, and E. Sontag, Eds., vol. 1066 of *Lecture Notes in Computer Science*, Elsevier Science Publishers B.V., pp. 496–510.
- [24] MILNER, R. *A Calculus of Communicating Systems*, vol. 92 of *Lecture Notes in Computer Science*. Springer-Verlag, 1980.
- [25] SIGHIREANU, M. *Contribution à la définition et à l'implémentation de la norme "Extended LOTOS"*. PhD thesis, Université Joseph Fourier, Grenoble, France, 1999.
- [26] VAN DER ZWAAG, M. Time-stamped actions in p CRL algebras. Tech. Rep. SEN-R0002, CWI, Amsterdam, 2000.
- [27] VAN HAM, F. Visualisation of state transition graphs. Master's thesis, Eindhoven University of Technology, 2000.
- [28] VEREIJKEN, J. *Discrete-time process algebra*. PhD thesis, Eindhoven University of Technology, 1997.
- [29] YOVINE, S. Kronos: A verification tool for real-time systems. *International Journal on Software Tools for Technology Transfer* 1, 1-2 (1997).