1992

A.S. Klusener

The silent step in time

Computer Science/Department of Software Technology Report CS-R9221 June

CWI is the research institute of the Stichting Mathematisch Centrum, which was founded on February 11, 1946, as a non-profit institution aiming at the promotion of mathematics, computer science, and their applications. It is sponsored by the Dutch Government through the Netherlands organization for scientific research (NWO).

The Silent Step in Time *

A.S. Klusener

CWI

P.O. Box 4079, 1009 AB Amsterdam, The Netherlands

e-mail: stevenk@cwi.nl

Abstract. In untimed process algebras such as CCS and ACP the silent step enables abstraction from internal actions. Several formalizations of abstraction have been introduced, such as observational congruence, delay bisimulation and branching bisimulation. However, in real time process algebras, such as ACPρ, the silent step has not yet been worked out. In this paper we formalize these semantics regarding the silent step in real time process algebra. We study the characterizing laws, which correspond closely to the untimed laws, and we investigate which of the semantics is appropriate in the context of ACPρ. We restrict ourselves to process terms without recursion and we obtain completeness and decidability for branching bisimulation equivalence.

1985 Mathematics Subject Classification: 68Q60.
1982 CR Categories: D.3.1, F.3.1, J.7.
Key Words & Phrases: Real Time Process Algebra, ACP, Abstraction.

Introduction

Over the years several semantics of the silent step have been introduced. The silent step, normally denoted by the symbol τ, is due to Milner. It enables abstraction from internal activity. As abstraction mechanism it serves for proving that an implementation meets its specification. However, there are several options for defining an equivalence on processes involving silent steps. Milner & Hennessy have defined *observation equivalence* ([HM80]), later Milner defined a slightly stricter equivalence to which we refer as *delay bisimulation* ([Mil81]). Since observation equivalence may be confused with observational equivalence, which coincides with strong bisimulation for finitely branching processes, we refer to observation equivalence by *weak bisimulation.* Van Glabbeek & Weijland have argued in ([GW91]) that neither of these equivalences respect the branching structures of the processes fully and therefore they introduced *branching bisimulation.* Thus branching bisimulation equivalence is contained in delay bisimulation which, on its turn, is contained in weak bisimulation. Finally, Baeten & Van Glabbeek defined η*-bisimulation* in [BG87]. η-bisimulation is contained in weak bisimulation and it contains branching bisimulation but it is incomparable with delay bisimulation. In this paper we will not study η-bisimulation any further.

*An extended abstract of an earlier version of this report has appeared in [Cl92].

Report CS-R9221
ISSN 0169-118X
CWI
P.O. Box 4079, 1009 AB Amsterdam, The Netherlands

Recently, Baeten & Bergstra introduced ACPρ (ACP with real time) in [BB91]. To each occurrence of an atomic action a time stamp (from $[0,\infty]$) is assigned, denoting the time at which the action is being executed. This leads to two languages, one in which the time stamp is interpreted *absolutely*, i.e. from the starting time zero, and one in which it is interpreted *relatively*, i.e. with respect to the previous action. They have also introduced the powerful mechanism of integration, which can be considered as an alternative composition over a continuum of alternatives. We refer to ACPρ with integration by ACPρI. Using integration it can be expressed that an action must be executed somewhere within a certain interval. Moreover, time dependencies between consecutive actions can be expressed. For example

$$\int_{v\in\langle 0,1]} a(v) \quad \text{and} \quad \int_{v\in\langle 0,1]} (a(v)\cdot \int_{w\in\langle v+1,v+3\rangle} b(w)).$$

The process term on the left hand side denotes the process which executes the atomic action a somewhere in the time interval $\langle 0,1]$ after which the process terminates successfully. The process term on the right hand side executes, again, the atomic action a somewhere in the time interval $\langle 0,1]$. After that it idles in between 1 and 3 time units before executing the atomic action b. In this paper we denote a left and/or right open interval with respectively $\langle$ and $\rangle$. Note that this latter term is equivalent to the following term in relative time:

$$\int_{v\in\langle 0,1]} (a[v]\cdot \int_{w\in\langle 1,3\rangle} b[w])$$

In fact, every term in relative time can be translated to a term in absolute time (see [BB91]). The converse is implied by the fact that every term in absolute time (without recursion) can be equated to a *basic* term [Klu91a],[FK92]. Each basic term can be translated directly to a term in relative time. In [BB92] Baeten and Bergstra define each process term in relative time as a function which takes a point in time (the starting point of the process or the point at which the process is being enabled) and delivers a process term in absolute time. In this paper we will consider absolute time only.

In [Klu91a] the restricted syntax of *prefixed integration* without recursion is studied, for this restriction it is proven that the theory of ACPρI characterizes strong bisimulation completely. Finally, Fokkink has proven in [Fok91] that equality in ACPρI is decidable for prefixed integration without recursion. In [FK92] the reports [Klu91a] and [Fok91] are integrated and several improvements have been applied.

In the next section we start with a brief presentation of (untimed) process algebra and recall three bisimulations which involve the silent step. These bisimulations are given in detail together with the rootedness condition on these bisimulations that turns them into congruences. We restrict ourselves to Basic Process Algebra ($\mathrm{BPA}_{\tau\delta}$) without recursion.

Then we present the syntax of real time Basic Process Algebra (BPA$\rho\tau\delta$) in Section 2. Note that the ρ in BPA$\rho\tau\delta$ stands for *real time*. We discuss strong timed bisimulation and its axiomatization. After having presented the silent step in untimed process algebra and strong bisimulation in the timed case we can adapt the definition of untimed branching bisimulation by adding time in Section 3. We see that the silent step may correspond with idling and that we need a rootedness condition in order to have a congruence. We obtain one new law which characterizes completely those identities of rooted branching bisimulation over BPA$\rho\tau\delta$ which are not strong bisimilar. In Section 4 we add prefixed integration. We generalize the real time law which we have found in the section without integration, and we obtain a law which corresponds closely to Van Glabbeek & Weijland's branching bisimulation law. The completeness proof for the calculus with integration is rather involved, but since the proof is based on Fokkink's *normal forms* ([Fok91],[FK92]) we obtain decidability for branching bisimulation for granted.

We do similarly for delay and weak bisimulation in respectively Section 8 and Section 9. Again we have to apply a rootedness condition and for rooted timed delay and weak bisimulation we obtain laws which is very similar to Milner's second and third τ-law. Unfortunately rooted timed weak bisimulation is not a congruence anymore after adding parallel composition and encapsulation, which is essential for an equational theory like ACPρ.

In the original paper of Baeten & Bergstra ([BB91]) the authors suggest to interpret the τ transitions in the operational semantics as idling transitions. This idea is investigated in [Klu91b] for the case without integration, resulting in a timed τ-equivalence. In Section 8 we will see that this equivalence coincides with rooted timed delay bisimulation.

We do not discuss recursion. This may seem a very severe restriction, but it is our idea that the calculus without recursion has to be understood sufficiently before adding recursion.

Contents

1 The Silent Step without Time

1.1 Basic Process Algebra and Strong Bisimulation
1.2 Semantics for the Silent Step
1.3 Strongly Rootedness

2 Adding Time

3 Branching Bisimulation and Time

3.1 Rooted Timed Branching Bisimulation Equivalence
3.2 One Law for the Silent Step

4 Branching Bisimulation and Integration

4.1 Introduction
4.2 The Time Domain
4.3 Bounds and Intervals
4.4 Process Terms
4.5 α-conversion
4.6 SOS and Strong Bisimulation
4.7 Branching Bisimulation and Integration
4.8 A first generalization of ATB
4.9 The Timed Branching Law
4.10 Embedding untimed branching bisimulation into the timed one

5 Basic and Conditional Terms and their Strong Normal Forms

5.1 Introduction
5.2 Basic terms
5.3 Conditions and conditional Terms

5.4 The theory of CTA
5.5 Basic Terms and free time variables
5.6 Strong Normal Froms

6 Branching Normal Forms and the Completeness Proof
6.1 Introduction
6.2 The Conditional Branching Law
6.3 Rewrite Rules for Branching Normal Forms
6.3.1 Introducing τ's for each moment of choice
6.3.2 Partitioning a term
6.3.3 Removing τ's
6.4 The construction of Branching Normal Forms
6.5 The Unicity Theorem for Branching Bisimulation
6.6 Rooted Branching Normal Forms and Completeness

7 An Operational Semantics with Idle Transitions
7.1 The Action Rules and Strong Bisimulation
7.2 The correspondence between the two operational semantics
7.3 Branching Bisimulation

8 Delay Bisimulation and Time
8.1 Rooted Delay Bisimulation Equivalence
8.2 The Timed Delay Law
8.3 The embedding of untimed into timed rooted delay bisimulation

9 Weak Bisimulation and Time
9.1 Its Definition and associated Law
9.2 Weak Bisimulation is not a congruence over ACPρ

10 Conclusions

1 The Silent Step without Time

1.1 Basic Process Algebra and Strong Bisimulation

In this section we introduce some main notions of process algebra. Our point of departure is Basic Process Algebra with τ and δ, abbreviated by BPA$\tau\delta$. The symbol τ denotes the silent step and the constant δ denotes inaction, comparable with the *NIL* and the 0 of Milner ([Mil80],[Mil89]) and Hoare's *STOP* ([Hoa85]). This section is based on [BW90], while the part on bisimulations-with-silent-step is based as well on [GW91].

We have an alphabet A, not containing τ and δ. $A \cup \{\tau, \delta\}$ is abbreviated by $A_{\tau\delta}$. The set of terms over BPA$\tau\delta$ is denoted by $\mathcal{T}_u$ (the subscript u refers to *untimed*) and it has typical elements p, p_i.

$$p ::= a \mid p_1 + p_2 \mid p_1 \cdot p_2 \quad (a \in A_{\tau\delta})$$

Here, $p_1 + p_2$ is the alternative composition of p_1, p_2 while $p_1 \cdot p_2$ is the sequential composition of p_1, p_2. An element of $\mathcal{T}_u$ is referred to as a term or a process term.

The semantics of a process term p is a labeled transition system with p as root. The states of such a transition system are taken from $\mathcal{T}_u \cup \{\surd\}$. The symbol $\surd$ denotes termination. These transition systems are defined as the least relations satisfying the action rules of Table 1. This style of giving operational semantics is due to Plotkin ([Plo81]).

We may identify process terms of which the transition systems represent the same behavior. However, there are several options of formalizing the idea of "having the same behavior", especially when dealing with the silent step.

The first equivalence we give is strong bisimulation equivalence, denoted by $\leftrightarrow_s$, it does not yet regard the special role of the silent action. $p \leftrightarrow_s q$ means roughly that every transition of p can be mimicked by q such that the resulting pair is strongly bisimilar again and vice versa. First we need the definition of a strong bisimulation. In the sequel $p\mathcal{R}q$ abbreviates that the binary relation $\mathcal{R}$ contains the pair (p, q).

$$a \xrightarrow{a} \surd$$

$$\frac{p \xrightarrow{a} p'}{p \cdot q \xrightarrow{a} p' \cdot q} \qquad \frac{p \xrightarrow{a} \surd}{p \cdot q \xrightarrow{a} q}$$

$$\frac{p \xrightarrow{a} p'}{p + q \xrightarrow{a} p'} \qquad \frac{p \xrightarrow{a} p'}{q + p \xrightarrow{a} p'}$$

$$\frac{p \xrightarrow{a} \surd}{p + q \xrightarrow{a} \surd} \qquad \frac{p \xrightarrow{a} \surd}{q + p \xrightarrow{a} \surd}$$

Table 1: Action Rules for $\mathcal{T}_u$ $(a \in A_\tau)$

Definition 1.1 $\mathcal{R} \subset \mathcal{T}_u \times \mathcal{T}_u$ *is a* strong bisimulation *if whenever* $p\mathcal{R}q$ *then*

1. $p \xrightarrow{a} p'$ *implies* $\exists q'$ *such that* $q \xrightarrow{a} q'$ *and* $p'\mathcal{R}q'$.
2. $p \xrightarrow{a} \surd$ *implies* $q \xrightarrow{a} \surd$.
3. *Respectively (1) and (2) with the role of p and q interchanged.*

Strong bisimulation equivalence is now defined by

Definition 1.2 $p \,\underline{\leftrightarrow}_s\, q$ *iff there is a strong bisimulation* $\mathcal{R}$ *relating p and q.*

This equivalence is known to be a congruence. Hence, the transition system model modulo strong bisimulation is a well-defined algebra; in fact it can be completely axiomatized by the axiom system BPAδ of Table 2.

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	$(X \cdot Y) \cdot Z$	$=$	$X \cdot (Y \cdot Z)$

BPA = A1 – A5

A6	$X + \delta$	$=$	X
A7	$\delta \cdot X$	$=$	δ

Table 2: BPAδ = BPA + A6 + A7

1.2 Semantics for the Silent Step

We consider three different bisimulation equivalences and their rooted verions which regard the silent step: branching bis.([GW91]), delay bis. [Mil83] and weak bis. ([Mil89]).

$$\begin{array}{ccccc} \underline{\leftrightarrow}_b & \subset & \underline{\leftrightarrow}_d & \subset & \underline{\leftrightarrow}_w \\ \cup & & \cup & & \cup \\ \underline{\leftrightarrow}_{rb} & \subset & \underline{\leftrightarrow}_{rd} & \subset & \underline{\leftrightarrow}_{rw} \end{array}$$

Each of these bisimulation equivalences allow that an a-transition on one side may be mimicked by a a-transition possibly preceded or followed by silent steps on the other side. This is shown in Figure 1, their formal definitions are given below.

We have one predicate on $\mathcal{T}_u \cup \{\surd\}$ which is denoted by $\surd$ as well; $\surd(p)$ holds iff all maximal paths starting in p consist of τ's only and end in $\surd$. Note that $\surd(\surd)$.

In the following $p \Longrightarrow p'$ denotes that there is a path $p \xrightarrow{\tau} \ldots \xrightarrow{\tau} p'$ of length zero or more.

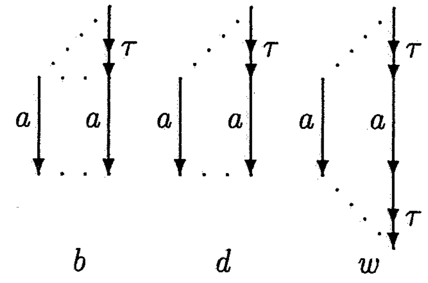


Figure 1: Three bisimulations with τ

Definition 1.3 *$\mathcal{R} \subset \mathcal{T}_u \times \mathcal{T}_u$ is an untimed* branching *bisimulation if whenever $p\mathcal{R}q$ then*

1. *If $p \xrightarrow{a} p'$ then either $a = \tau$ and $p'\mathcal{R}q$ or $\exists z, q'$ such that $q \Longrightarrow z \xrightarrow{a} q'$, $p\mathcal{R}z$ and $p'\mathcal{R}q'$.*
2. *If $p \xrightarrow{a} \surd$ then $\exists z, z'$ such that $q \Longrightarrow z \xrightarrow{a} z'$ with $\surd(z')$ and $p\mathcal{R}z$.*
3. *Respectively (1) and (2) with the role of p and q interchanged.*

Definition 1.4 *$\mathcal{R} \subset \mathcal{T}_u \times \mathcal{T}_u$ is an untimed* delay *bisimulation if whenever $p\mathcal{R}q$ then*

1. *If $p \xrightarrow{a} p'$ then either $a = \tau$ and $p'\mathcal{R}q$ or $\exists z, q'$ such that $q \Longrightarrow z \xrightarrow{a} q'$ and $p'\mathcal{R}q'$.*
2. *If $p \xrightarrow{a} \surd$ then $\exists z, z'$ such that $q \Longrightarrow z \xrightarrow{a} z'$ with $\surd(z')$.*
3. *Respectively (1) and (2) with the role of p and q interchanged.*

Definition 1.5 *$\mathcal{R} \subset \mathcal{T}_u \times \mathcal{T}_u$ is an untimed* weak *bisimulation if whenever $p\mathcal{R}q$ then*

1. *If $p \xrightarrow{a} p'$ then either $a = \tau$ and $p'\mathcal{R}q$ or $\exists z, z', q'$ such that $q \Longrightarrow z \xrightarrow{a} z' \Longrightarrow q'$ and $p'\mathcal{R}q'$.*
2. *If $p \xrightarrow{a} \surd$ then $\exists z, z'$ such that $q \Longrightarrow z \xrightarrow{a} z' \Longrightarrow \surd$.*
3. *Respectively (1) and (2) with the role of p and q interchanged.*

We need the predicate $\surd$ in this paper to express that "τ-stuttering" afterwards is allowed. Hence, we obtain the law $X \cdot \tau = X$. For $* \in \{b, d, w\}$ we define untimed $*$-bisimulation equivalence.

Definition 1.6 *$p \leftrightarrow_* q$ iff there is an untimed $*$-bisimulation relating p and q.*

None of these equivalences is a congruence over $\mathcal{T}_u$. We have to restrict these equivalences to obtain congruences by imposing a rootedness condition on the bisimulation relations.

Definition 1.7 *A relation $\mathcal{R}$ is* rooted *w.r.t. p and q if $p\mathcal{R}q$ and if it does not relate p or q with states which can be reached from either p or q in one or more steps.*

We obtain untimed *rooted* $*$-bisimulation equivalences, denoted by $p \leftrightarrow_{r*} q$, by requiring that there is a untimed rooted $*$-bisimulation relating p and q. Now we have for each $* \in \{b, d, w\}$ that:

Proposition 1.8 $\leftrightarrow_{r*}$ *is a congruence*

T1	$X \cdot \tau$	$=$	X	B1
	$Z \cdot (\tau \cdot (X + Y) + X)$	$=$	$Z \cdot (X + Y)$	B2
T2	$\tau \cdot X$	$=$	$\tau \cdot X + X$	
T3	$a \cdot (\tau \cdot X + Y)$	$=$	$a \cdot (\tau \cdot X + Y) + a \cdot X$	

Table 3: The untimed τ laws

The laws T1-T3 are taken from Milner ([Mil89]). B2 is Van Glabbeek & Weijland's branching bisimulation law, note that A3+T1+T2 $\vdash$ B2. Each rooted bisimulation equivalence can be axiomatized completely by its corresponding theory.

Theorem 1.9 $p, q \in \mathcal{T}_u$

$$p \leftrightarrow_{rb} q \iff \text{BPA}\delta + \text{B1} + \text{B2} \vdash p = q$$
$$p \leftrightarrow_{rd} q \iff \text{BPA}\delta + \text{T1} + \text{T2} \vdash p = q$$
$$p \leftrightarrow_{rw} q \iff \text{BPA}\delta + \text{T1} + \text{T2} + \text{T3} \vdash p = q$$

In [GW91] the completeness is proven for branching bisimulation equivalence first. From this result the other completeness results can be found easily.

1.3 Strongly Rootedness

In the timed case we will come across a stronger rootedness condition. There it is required that a rooted bisimulation acts from the root nodes as a strong bisimulation.

Definition 1.10 *A relation $\mathcal{R}$ is* strongly rooted *w.r.t. p and q if*

1. $p\mathcal{R}q$
2. $p \xrightarrow{a} p'$ *implies* $\exists q'$ *such that* $q \xrightarrow{a} q'$ *and* $p'\mathcal{R}q'$.
3. $p \xrightarrow{a} \surd$ *implies* $q \xrightarrow{a} \surd$.
4. *Respectively (2) and (3) with the role of p and q interchanged.*

In this way we obtain *strongly rooted branching bisimulation* ($\leftrightarrow_{srb}$), *strongly rooted delay bisimulation* ($\leftrightarrow_{srd}$) and *strongly rooted weak bisimulation* ($\leftrightarrow_{srw}$).

For branching bisimulation strongly rootedness is not strictly stronger than rootedness:

Proposition 1.11 $p, q \in \mathcal{T}_u$

$$p \leftrightarrow_{rb} q \iff p \leftrightarrow_{srb} q$$

Proof. We proof only $\Longrightarrow$, the other direction is trivial. Assume $\mathcal{R}$ is a rooted branching bisimulation w.r.t. p and q. We prove that $\mathcal{R}$ is a strongly rooted branching bisimulation w.r.t. p and q as well.

Consider $p \xrightarrow{a} p'$, then it cannot be the case that $a = \tau$ and $p'\mathcal{R}q$ since $\mathcal{R}$ is rooted. Hence there is a z and a q' such that $q \Longrightarrow z \xrightarrow{a} q'$, $p\mathcal{R}z$ and $p'\mathcal{R}q'$. Since $\mathcal{R}$ is rooted z may not be reachable from q in one or more steps, thus $z \equiv q$ and thus $q \xrightarrow{a} q'$.

Similarly we can prove that $p \xrightarrow{a} \surd$ implies $q \xrightarrow{a} \surd$. □

This is certainly not the case for delay bisimulation, as is shown by the following example:

Example 1.12

$$\tau \cdot a \leftrightarrow_{rd} \tau \cdot a + a \quad \text{but} \quad \tau \cdot a \not\leftrightarrow_{srd} \tau \cdot a + a$$

For an axiomatization of $\leftrightarrow_{srd}$ we have to reformulate the second τ law since T2 is sound only in a context.

$\mathrm{T2}_{sr}$	$Z \cdot (\tau \cdot X + Y) = Z \cdot (\tau \cdot X + X + Y)$

And we have the following Theorem.

Theorem 1.13 $p, q \in \mathcal{T}_u$

$$p \leftrightarrow_{srd} q \quad \Longleftrightarrow \quad \mathrm{BPA}\delta + \mathrm{T1} + \mathrm{T2}_{sr} \vdash p = q$$

Proof. Omitted

2 Adding Time

We follow Baeten & Bergstra ([BB91]) in adding time stamps to the atomic actions. We take a from $A_{\tau\delta}$ and t from $[0,\infty]$. Three auxiliary operators are added, they are appropriate in the axiomatization and they will be discussed later. The real time extension of BPA$\tau\delta$ is denoted by BPA$\rho\tau\delta$. The associated set of terms is denoted by $\mathcal{T}_\rho$.

$$p ::= a(t) \mid p_1 + p_2 \mid p_1 \cdot p_2 \mid t \gg p \mid p \gg t \mid p@t \quad (a \in A_{\tau\delta}, t \in [0,\infty])$$

We assume that $\gg$ and @ bind the strongest and that + binds the weakest. Thus

$$t \gg p \cdot q + p \text{ is parsed as } ((t \gg p) \cdot q) + p$$

Syntactic equivalence is denoted by $\equiv$. Syntactic equivalence modulo A1 and A2 (associativity and commutativity of the +) is denoted by $\simeq$. We apply the following abbreviations:

$$\begin{array}{lll} X \ggeq t & \text{abbreviates} & X \gg t + X@t \\ t \ggeq X & \text{abbreviates} & t \gg X + X@t \end{array}$$

The term $\delta(0)$ is abbreviated by δ, it denotes the process which can not do any action nor can it idle to any point in time.

The time stamps are interpreted absolutely, thus from the start time zero. Hence, some parts of a process term may be inaccessible since they are not allowed anymore by the elapse of time caused by preceding actions:

$$\begin{array}{lcl} a(2) \cdot (b(1) + c(3)) & = & a(2) \cdot c(3) \\ a(2) \cdot b(1) & = & a(2) \cdot \delta \end{array}$$

The atomic action $b(1)$ cannot be executed after the execution of $a(2)$, since after $a(2)$ only actions can be executed with a timestamp greater than 2. Therefore, after the execution of the $a(1)$ the process $a(2) \cdot b(1)$ reaches a state of deadlock; nothing can be done anymore, even time cannot progress anymore, in other words the process has terminated unsuccessfully. Hence $a(2) \cdot b(1)$ equals $a(2) \cdot \delta$.

The process $t \gg p$ is that part of p which starts after t, the process $p \gg t$ is that part of p which starts before t. Finally, $p@t$ is that part of p which starts exactly at t and $p \ggeq t$ is that part of p which starts at or before t. (In the original paper [BB91] the operator $p \gg t$ is part of ACPρ and not of BPA$\rho\delta$, but we need this operator to define the τ-laws later on. The operator $p@t$ and the abbreviations $p \ggeq t$ and $t \ggeq X$ are new in the context of BPA$\rho\delta$. The notation @ occurs as well in [Wan90]). The operators $\gg$ and @ are auxiliary operators for the axiomatization of the other operators. For example, the sequential composition is partly axiomatized by

$$a(t) \cdot X = a(t) \cdot (t \gg X)$$

saying that the process $a(t) \cdot X$ can execute after $a(t)$ only that part of X which starts after t. The operator $\gg$ itself is defined inductively.

Apart from these operators we have two functions which are defined inductively on the structure of the terms. The *ultimate delay* of a term p, denoted by $U(p)$, is the least upper bound of points in time to which p can idle. The ultimate delay originates from

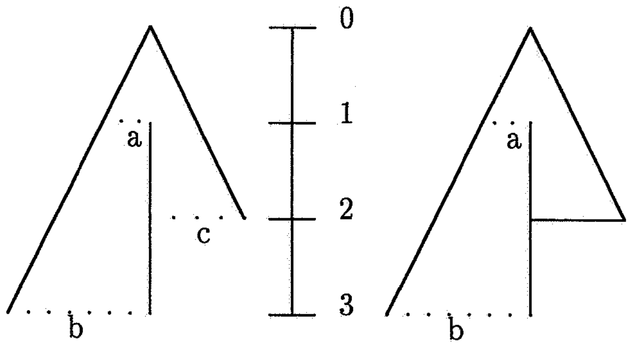


Figure 2: Process diagrams for $a(1) \cdot b(3) + c(2)$ and $a(1) \cdot b(3) + \delta(2)$

[BB91]. It is similar to the *system delay* of Moller & Tofts [MT90]. Moreover, we have the *latest starttime*, denoted by $L(p)$, it is the maximum of points in time at which p can perform an (initial) action. The *(earliest) starttime*, denoted by $S(p)$, is the minimum of points in time at which p can perform an (initial) action. The $S(p)$ is not used until Section 6.

Definition 2.1 *(Ultimate Delay, Latest Starttime and (Earliest) Starttime)*

$$
\begin{array}{lll}
U(a(t)) & = & t \\
U(\delta(t)) & = & t \\
U(p \cdot q) & = & U(p) \\
U(p+q) & = & max(U(p), U(q)) \\
U(r \gg p) & = & max(r, U(p)) \\
U(p \gg r) & = & min(r, U(p)) \\
U(p@r) & = & r
\end{array}
\qquad
\begin{array}{lll}
initact(a(t)) & = & \{t\} \\
initact(\delta(t)) & = & \emptyset \\
initact(p \cdot q) & = & initact(p) \\
initact(p+q) & = & initact(p) \cup initact(q) \\
initact(r \gg p) & = & \{t \in initact(p) | t > r\} \\
initact(p \gg r) & = & \{t \in initact(p) | t < r\} \\
initact(p@r) & = & \{t \in initact(p) | t = r\} \\
L(p) & = & max(initact(p)) \\
S(p) & = & min(initact(p))
\end{array}
$$

Before giving the operational semantics we give some *process diagrams* of process terms as proposed by Baeten and Bergstra. In Figure 2 two examples are given. The intuition behind a process diagram is that a process starts in the top-point. It can idle by going to a lower point without crossing any line, whereas the execution of an action a at 1 is reflected by going to a dashed line at level 1. Only dashed lines may be crossed, after landing on them. We see in the process diagram of $a(1) \cdot b(3) + c(2)$ that this process can idle from its start state to a state with time between 1 and 2. From that state the action a at 1 can not be executed anymore and it is clear that the c will be executed at time 2. The process $a(1) \cdot b(3) + \delta(2)$ can idle as well from its start state to a state with time in between 1 and 2; but then no action can be executed at all. This process can idle till 2 without the option of doing anything at 2 nor can it progress to a point in ti.ne at or after 2. In other words the process has a *time stop* at 2. In general, the process $\delta(t)$ may be regarded as the process with an unspecified behavior at time t. The terminology time stop can be found as well in ([Jef91]). A timed deadlock at t will always be avoided if something else is possible at or after t. This is shown in Figure 3; in the process diagram

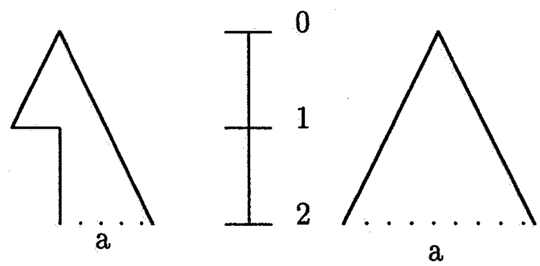


Figure 3: Process diagrams for $\delta(1) + a(2)$ and $a(2)$

of $a(2) + \delta(1)$ it is always possible to go to the dashed line labelled with a at level 2 without crossing any line.

There are several options for defining the operational semantics. Baeten and Bergstra have given an operational semantics in which all possible idlings are explicit in the transition system; in section 7 we will present a similar semantics which will be called the *idle* semantics.

However, in this section we will present an operational semantics in which the idlings of the processes are not considered explicitly. The reason for not considering the idlings explicitly is that less inference rules are needed for the definition of the operational semantics and that it is easier to reason formally about the transition systems of the processes.

The price to pay is the introduction of the δ-transitions. We require that $a(2) + \delta(1)$ is bisimilar with $a(2)$ and that $a(1) + \delta(2)$ is not bisimilar with $a(1)$. Therefore we add to each state with a timestop (which can not be avoided) a δ-transition. The condition for such a transition is defined in terms of $U()$ and $L()$; if a process can idle to a point in time at which no actions are possible anymore (in other words if $U(p) > L(p)$) then an δ-transition with time $U(p)$ is added. The semantics of a process term is again a labeled transition system, the action rules and strong bisimulation equivalence have been taken from [Klu91a]. The labels are now timed actions (in $A_{\tau\delta}$). The states are taken from $\mathcal{T}_u \cup \{\surd\}$. The root state of the transition system of a term p is the state p. We have the following transitions:

$$a(r) \xrightarrow{a(r)} \surd, \quad a(r) + p \xrightarrow{a(r)} \surd \quad \text{and} \quad a(r) \cdot p \xrightarrow{a(r)} r \gg p$$

The right most transition shows us that the course of time is encoded by $r \gg ....$ The action rules are given in Table 2. We define a strong bisimulation for timed transition systems.

Definition 2.2 *Strong Bisimulation*
$\mathcal{R} \subset \mathcal{T}_\rho \times \mathcal{T}_\rho$ *is a* strong bisimulation *if whenever* $p\mathcal{R}q$ *then*

1. $p \xrightarrow{a(r)} p'$ *implies* $\exists q'$ *such that* $q \xrightarrow{a(r)} q'$ *and* $p'\mathcal{R}q'$.

2. $p \xrightarrow{a(r)} \surd$ *implies* $q \xrightarrow{a(r)} \surd$.

3. *Respectively (1) and (2) with the role of* p *and* q *interchanged.*

$a(r) \xrightarrow{a(r)} \surd$		$\dfrac{U(p) > L(p)}{p \xrightarrow{\delta(U(p))} \surd}$	
$\dfrac{p \xrightarrow{a(r)} p'}{p+q \xrightarrow{a(r)} p'}$	$\dfrac{p \xrightarrow{a(r)} \surd}{p+q \xrightarrow{a(r)} \surd}$	$\dfrac{r > s \quad p \xrightarrow{a(r)} p'}{s \gg p \xrightarrow{a(r)} p'}$	$\dfrac{r > s \quad p \xrightarrow{a(r)} \surd}{s \gg p \xrightarrow{a(r)} \surd}$
$\dfrac{p \xrightarrow{a(r)} p'}{q+p \xrightarrow{a(r)} p'}$	$\dfrac{p \xrightarrow{a(r)} \surd}{q+p \xrightarrow{a(r)} \surd}$	$\dfrac{r < s \quad p \xrightarrow{a(r)} p'}{p \gg s \xrightarrow{a(r)} p'}$	$\dfrac{r < s \quad p \xrightarrow{a(r)} \surd}{p \gg s \xrightarrow{a(r)} \surd}$
$\dfrac{p \xrightarrow{a(r)} p'}{p \cdot q \xrightarrow{a(r)} p' \cdot q}$	$\dfrac{p \xrightarrow{a(r)} \surd}{p \cdot q \xrightarrow{a(r)} r \gg q}$	$\dfrac{p \xrightarrow{a(r)} p'}{p@r \xrightarrow{a(r)} p'}$	$\dfrac{p \xrightarrow{a(r)} \surd}{p@r \xrightarrow{a(r)} \surd}$

Table 4: Action Rules ($a \in A_\tau$ and $r \in \langle 0, \infty \rangle$, $s \in [0, \infty]$)

And we have strong bisimulation equivalence.

Definition 2.3 *Strong Bisimulation Equivalence*
$p \leftrightarrow_s q$ *iff there is a strong bisimulation* $\mathcal{R}$ *such that* $p\mathcal{R}q$.

This equivalence is completely axiomatized by the theory BPA$\rho\delta$ ([BB91],[FK92]), which is given in Table 5. In our operational semantics the course of time is encoded in an application of the $\gg$ operator. For later use we define the function *time* to extract the course of time from a state.

Definition 2.4 *The function time* : $\mathcal{T}_\rho \longrightarrow [0, \infty]$

$$\begin{aligned} time(X+Y) &= 0 \\ time(X \gg r) &= 0 \\ time(X@r) &= 0 \\ time(r \gg X) &= r \\ time(X \cdot Y) &= time(X) \end{aligned}$$

The definition of *time* is motivated by the following proposition.

Proposition 2.5 $p \in \mathcal{T}_\rho$

$$p \xrightarrow{a(r)} p' \quad \Longrightarrow \quad time(p') = r$$

Finally, we have the following simple facts.

Proposition 2.6 $p, q \in \mathcal{T}_\rho$

$$\begin{array}{rrcl} & p & \leftrightarrow_s & time(p) \gg p \\ t < r & t \gg p & \leftrightarrow_s & t \gg q \qquad \Longrightarrow \qquad r \gg p \leftrightarrow_s r \gg q \\ & p & \leftrightarrow_s & q \qquad \Longleftrightarrow \quad \forall t \;\; t \gg p \leftrightarrow_s t \gg q \end{array}$$

	$a(0) = \delta(0)$
	$a(\infty) = \delta(\infty)$
	$\delta(t) \cdot X = \delta(t)$
$t \leq s$	$\delta(t) + \delta(s) = \delta(s)$
	$a(t) + \delta(t) = a(t)$
	$a(t) \cdot X = a(t) \cdot (t \gg X)$
	$t \gg (X + Y) = (t \gg X) + (t \gg Y)$
	$t \gg (X \cdot Y) = (t \gg X) \cdot Y$
$t < s$	$t \gg a(s) = a(s)$
$t \geq s$	$t \gg a(s) = \delta(t)$
	$(X + Y) \gg t = (X \gg t) + (Y \gg t)$
	$(X \cdot Y) \gg t = (X \gg t) \cdot Y$
$t < s$	$a(s) \gg t = \delta(t)$
$t \geq s$	$a(s) \gg t = a(s)$
	$(X + Y)@t = (X@t) + (Y@t)$
	$(X \cdot Y)@t = (X@t) \cdot Y$
$t = s$	$a(s)@t = a(s)$
$t \neq s$	$a(s)@t = \delta(t)$

Table 5: An Axiom System for BPA$\rho\delta$ ($a \in A_{\tau\delta}$) (on top of BPA)

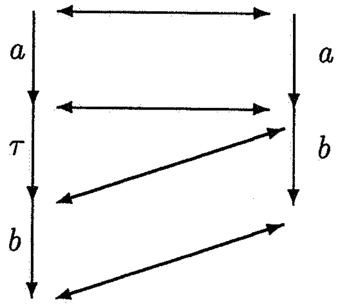
3 Branching Bisimulation and Time

3.1 Rooted Timed Branching Bisimulation Equivalence

In untimed branching bisimulation it is allowed that

a τ-transition on one side may be matched
with no transition at all at the other side.

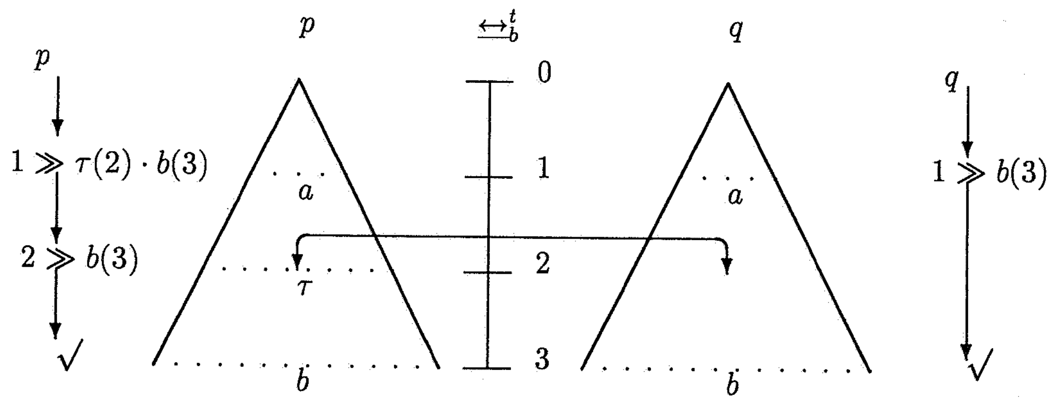
if this τ-transition does not determine a choice. Take for example:



In our real time context we can only relate states with the same *time* value. Therefore the statement above becomes

a (timed) τ-transition on one side may be matched
with an idling at the other side.

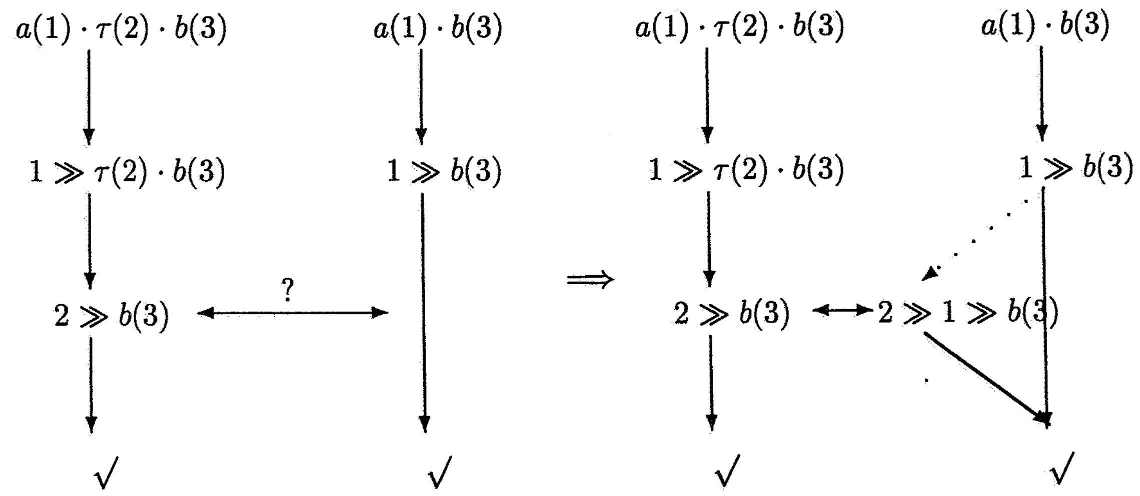
For example take $p \equiv a(1) \cdot \tau(2) \cdot b(3)$ and $q \equiv a(1) \cdot b(3)$:



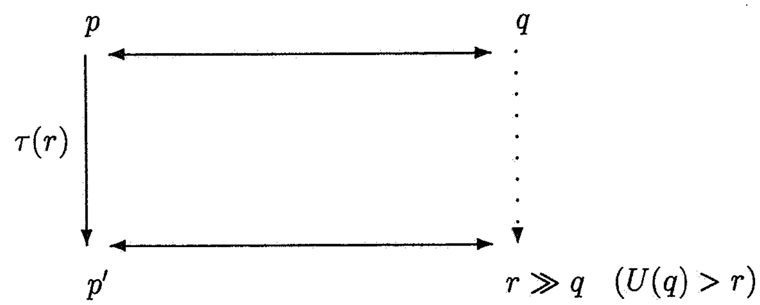
In the process diagrams we see that the τ at 2 (within p) can be matched with an idling to 2 (within q). However, since we do not consider idlings explicitly in the transition systems there is no transition in the transition system of q which corresponds with

$$1 \gg (\tau(2) \cdot b(3)) \xrightarrow{\tau(2)} 2 \gg b(3).$$

But, if necessary, we can make an idling explicit by the use of the $\gg$ operator:



For an untimed branching bisimulation $p\mathcal{R}q$ and $p \xrightarrow{\tau} p'$ might imply that $p'\mathcal{R}q$. A (timed) branching bisimulation $\mathcal{R}$ must increase the time in q accordingly to the transition $p \xrightarrow{\tau(r)} p'$, resulting in $p'\mathcal{R}(r \gg q)$. Some care is needed, since it must be guaranteed that q is able to idle till r, which is formalized by $U(q) > r$.

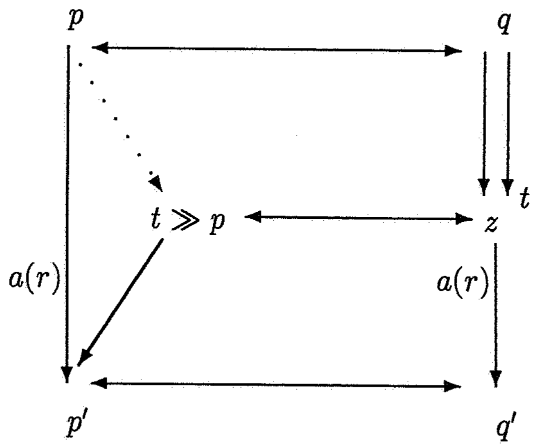


In the second part of the first clause in the definition of untimed branching bisimulation the "intermediate" state z (where $q \Longrightarrow z \xrightarrow{a} q'$) in the transition system of q must be related to p. To translate this situation to the real time case we first have to define $z \Longrightarrow_t z'$ inductively:

$$z \Longrightarrow_{time(z)} z$$

$$\text{if} \quad z \Longrightarrow_t z' \xrightarrow{\tau(t')} z'' \quad \text{then} \quad z \Longrightarrow_{t'} z''$$

In real time it will always be the case that the time in z (where $q \Longrightarrow_t z \xrightarrow{a(r)} q'$) is greater than the time in q, since z is a state after q. Hence, in stead of $p\mathcal{R}z$ we have $(t \gg p)\mathcal{R}z$ where $t = time(z)$.

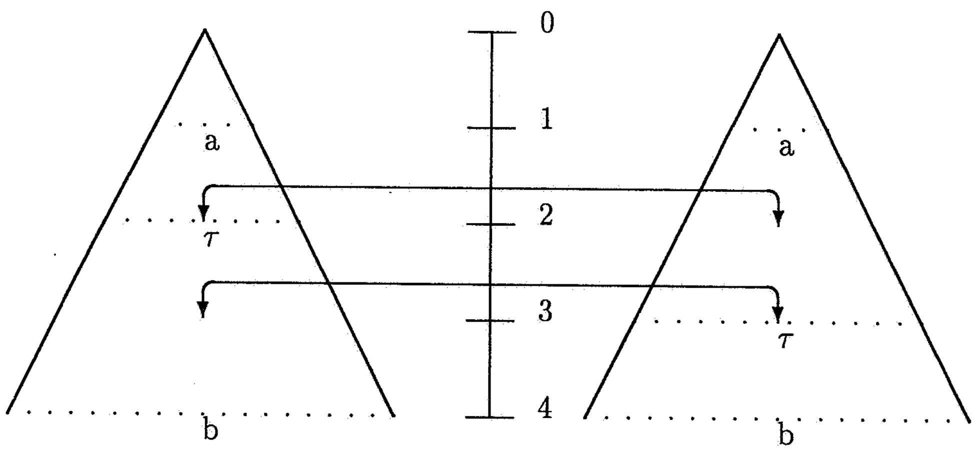


In timed branching bisimulation no "τ-stuttering" is allowed afterwards; we enforce that two bisimilar process terms terminate at the same points in time. If we take the definition of an untimed branching bisimulation then these considerations could imply the following definition, at a first sight. However, we will see that this definition is not yet totally correct.

Definition 3.1 *Naive Timed Branching Bisimulation (not yet correct!)*
$\mathcal{R} \subset \mathcal{T}_p \times \mathcal{T}_p$ *is a* branching bisimulation *if whenever* $p\mathcal{R}q$ *then*

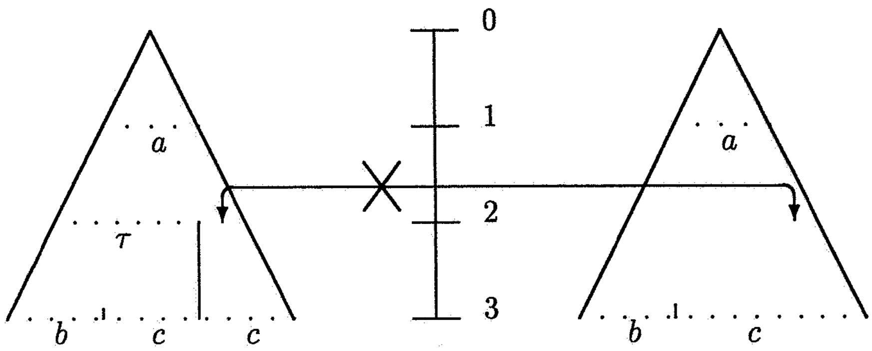
1. $p \xrightarrow{a(r)} p'$ *implies*
   - *either* $a = \tau$, $U(q) > r$ *and* $p'\mathcal{R}(r \gg q)$
   - *or* $\exists z, q'$ *and* $\exists t$ *such that* $q \Longrightarrow_t z \xrightarrow{a(r)} q'$ *and* $(t \gg p)\mathcal{R}z$ *and* $p'\mathcal{R}q'$.
2. $p \xrightarrow{a(r)} \surd$ *implies* $\exists z$ *and* $\exists t$ *such that* $q \Longrightarrow_t z \xrightarrow{a(r)} \surd$ *and* $(t \gg p)\mathcal{R}z$.
3. *Respectively (1) and (2) with the role of p and q interchanged.*

The problem with this definition that in case $p \xrightarrow{\tau(r)} p'$ is matched with an idling of q then q is not allowed to do internal actions before r. Hence, it is not possible to relate the term $a(1) \cdot \tau(2) \cdot b(4)$ with $a(1) \cdot \tau(3) \cdot b(4)$.



So we must allow τ-transitions at the other side as well.

However, this is still not sufficient. We have to distinguish the terms $a(1) \cdot (\tau(2) \cdot (b(3) + c(3)) + c(3))$ and $a(1) \cdot ((b(3) + c(3))$, since in the first term it may be the case that at 2 it is chosen that the c will be executed at 3 while in the latter term the choice between the b and the c at 3 can not be done earlier than 3.

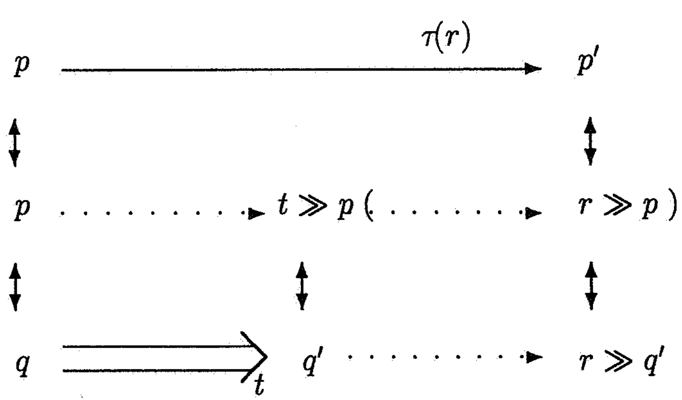


Definition 3.2 *Timed Branching Bisimulation*
$\mathcal{R} \subset \mathcal{T}_\rho \times \mathcal{T}_\rho$ *is a* branching bisimulation *if whenever* $p\mathcal{R}q$ *then*

1. $p \xrightarrow{a(r)} p'$ *implies*
   - *either* $a = \tau$, $\exists q'$ *with* $U(q') > r$ *and* $\exists t$ *such that* $q \Longrightarrow_t q'$ *and* $(t \gg p)\mathcal{R}q'$, $p'\mathcal{R}(r \gg q')$ *and* $U(p) > r$ *implies* $p'\mathcal{R}r \gg p$.
   - *or* $\exists z, q'$ *and* $\exists t$ *such that* $q \Longrightarrow_t z \xrightarrow{a(r)} q'$ *and* $(t \gg p)\mathcal{R}z$ *and* $p'\mathcal{R}q'$.
2. $p \xrightarrow{a(r)} \surd$ *implies* $\exists z$ *and* $\exists t$ *such that* $q \Longrightarrow_t z \xrightarrow{a(r)} \surd$ *and* $(t \gg p)\mathcal{R}z$.
3. *Respectively (1) and (2) with the role of p and q interchanged.*

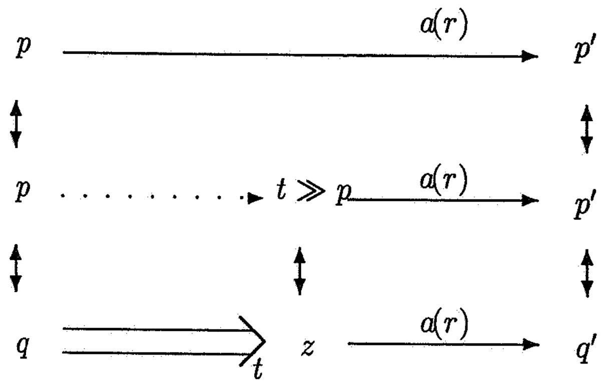
In the sequel we will often use pictures to refer to the two clauses of matching a transition.
$p \xrightarrow{a(r)} p'$ implies

- either $a = \tau$, $\exists q'$ with $U(q') > r$ and $\exists t$ such that



(Note that the part $U(p) > r$ implies $p'\mathcal{R}r \gg p$ is denoted by the use of the brackets. Depending of the context we will omit the brackets and print or leave out $r \gg p$.)

- or $\exists z, q'$ and $\exists t$ such that



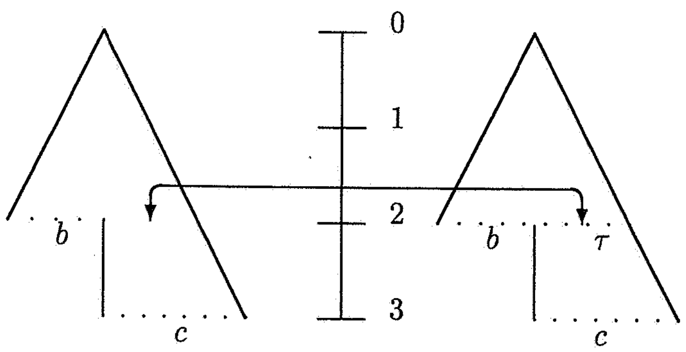
Using this definition it is possible to construct a branching bisimulation which relates $a(1)\cdot\tau(2)\cdot b(4)$ with $a(1)\cdot\tau(3)\cdot b(4)$, while $a(1)\cdot(\tau(2)\cdot(b(3)+c(3))+c(3))$ is distinguished from $a(1)\cdot((b(3)+c(3))$.

In the Section 7 we will introduce a slightly different semantics in which the transition systems have idle transitions, denoted by $q \xrightarrow{\iota(r)} q'$. Instead of the requirement $U(q) > r$ and $p'\mathcal{R}(r \gg q)$ we then have the requirement $q \xrightarrow{\iota(r)} q'$ and $p'\mathcal{R}q'$.

Definition 3.3 *Timed Branching Bisimulation Equivalence*
$p \leftrightarrow_b^t q$ *iff there is a branching bisimulation* $\mathcal{R}$ *such that* $p\mathcal{R}q$.

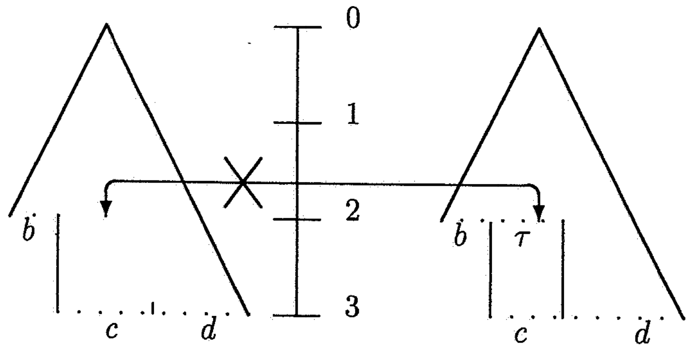
A typical example is :

$$b(2) + c(3) \quad \leftrightarrow_b^t \quad b(2) + \tau(2)\cdot c(3)$$



And again, this equivalence is not a congruence:

$$b(2) + c(3) + d(3) \quad \not\leftrightarrow_b^t \quad b(2) + \tau(2)\cdot c(3) + d(3)$$



We need a *rootedness* condition which is similar to the strongly rootedness condition for untimed bisimulations as given in Definition 1.10.

Definition 3.4 *Rooted Timed Branching Bisimulation Equivalence*
$p \leftrightarrow_{rb} q$ *iff*

1. $p \xrightarrow{a(r)} p'$ *implies* $\exists q'$ *such that* $q \xrightarrow{a(r)} q'$ *and* $p' \leftrightarrow_b q'$.
2. $p \xrightarrow{a(r)} \surd$ *implies* $q \xrightarrow{a(r)} \surd$.
3. *Respectively (1) and (2) with the role of p and q interchanged.*

Note that $a(2) + b(3) \not\leftrightarrow_b^t a(2) + \tau(2) \cdot b(3)$,

In the following we do not use the adjective *timed* anymore when referring to timed branching bisimulation. Similarly, if it is clear that $\leftrightarrow_b^t$ is mentioned we allow ourselves the freedom to write $\leftrightarrow_b$.

This rootedness condition for timed branching bisimulation corresponds with the strongly rootedness condition for the untimed case (Definition 1.10). For steps starting from the root it acts as strong bisimulation equivalence. As soon as an action (from A_τ) is executed τ-transitions cannot be distinguished anymore from idlings.

One may think of a machine which is not yet active at time zero, only when an action is executed it becomes active and it remains active until termination. Following this idea it makes sense to distinguish idling and internal activity from the root state; the process $a(2) + b(3)$ can be inactive between 2 and 3 while $a(2) + \tau(2) \cdot b(3)$ cannot.

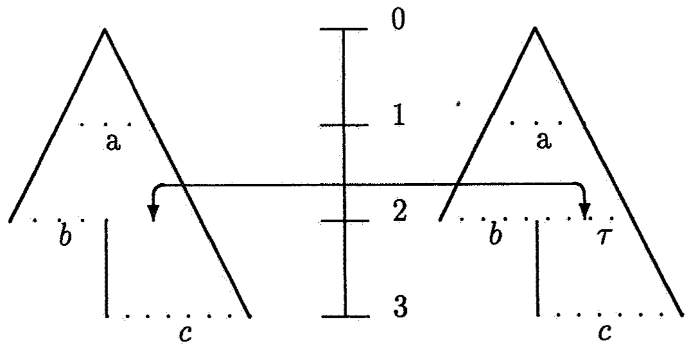
Proposition 3.5 $\leftrightarrow_{rb}$ *is a congruence for* BPA$\rho\delta\tau$

The proof of this proposition is postponed to section 4.

3.2 One Law for the Silent Step

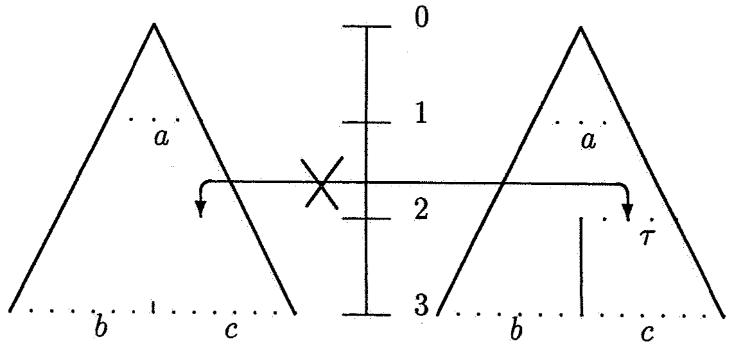
A typical identity is the following

$$a(1) \cdot (b(2) + c(3)) \quad \leftrightarrow_b^t \quad a(1) \cdot (b(2) + \tau(2) \cdot c(3))$$



The choice for the $c(3)$ is determined at time 2, with or without $\tau(2)$ in front of it. This identity depends on the time stamp of the b action. For if we change it to 3 then there is no identity anymore:

$$a(1)\cdot(b(3)+c(3)) \quad \not\leftrightarrow^t_b \quad a(1)\cdot(b(3)+\tau(2)\cdot c(3))$$



We can axiomatize the above identities by the following law, which occurs already in [Klu91b]:

ATB′	$r < t < U(X) \wedge U(Y) \leq t$	:	$a(r)\cdot(\tau(t)\cdot X + Y) = a(r)\cdot(t \gg X + Y)$

However the conditions are rather involved, and we prefer a slight reformulation.

In general, if p can idle till t we may split p into two parts, one which starts before or at a time t (denoted by $p \ggg t$) and the other part which starts after t (denoted by $t \gg p$). Whenever $t < U(p)$ we have

$$p \quad \leftrightarrow_s \quad p \ggg t + t \gg p \quad \leftrightarrow_b \quad p \ggg t + \tau(t)\cdot p$$

In order to have a corresponding equivalence in $\leftrightarrow_{rb}$ we have to put it in a context $a(r)\cdot(..)$ where $r < t$. We can express this algebraically by the following real time τ-law (AT stands for *Absolute Time* and the B for *Branching*).

ATB	$r < t < U(X)$	:	$a(r)\cdot X = a(r)\cdot(X \ggg t + \tau(t)\cdot X)$

The following examples show the need for the condition $r < t < U(X)$, without this condition ATB would identify $a(1)\cdot\tau(1)\cdot b(3)$ with $a(1)\cdot b(3)$ and $a(1)\cdot\tau(3)\cdot b(3)$ with $a(1)\cdot b(3)$.

$$\begin{array}{lclcl} a(1)\cdot\delta & \leftrightarrow_s & a(1)\cdot\tau(1)\cdot b(3) & \not\leftrightarrow_b & a(1)\cdot b(3) \\ a(1)\cdot\tau(3)\cdot\delta & \leftrightarrow_s & a(1)\cdot\tau(3)\cdot b(3) & \not\leftrightarrow_b & a(1)\cdot b(3) \end{array}$$

Finally we show that ATB′ is indeed derivable from ATB, at least for closed terms. First we need a proposition.

Proposition 3.6 *$X, Y, Z \in \mathcal{T}_p$ where $U(Y) \leq t < U(X)$*

$$\begin{array}{lcl} t \gg (t \gg Z) & = & t \gg Z \\ (t \gg Z) \ggg t & = & \delta(t) \\ t \gg X + \delta(t) & = & t \gg X \\ Y \ggg t & = & Y \\ t \gg Y & = & \delta(t) \end{array}$$

Proof. Each identity can be proven by induction on the structure of the term. □

Then we have

Proposition 3.7

$$\text{BPA}\rho\delta + \text{ATB} \vdash \text{ATB}'$$

Proof. Take X, Y and t such that $U(Y) \leq t < U(X)$ then we have

$$\begin{aligned}(t \gg X + Y) \ggcurly t &= (t \gg X) \ggcurly t + Y \ggcurly t \\ &= \delta(t) + Y\end{aligned}$$

$$\begin{aligned}t \gg X + t \gg Y &= t \gg X + \delta(t) \\ &= t \gg X\end{aligned}$$

And finally

$$\begin{aligned}a(r) \cdot (t \gg X + Y) &= a(r) \cdot ((t \gg X + Y) \ggcurly t + \tau(t) \cdot (t \gg X + Y)) \\ &= a(r) \cdot (\delta(t) + Y + \tau(t) \cdot (t \gg (t \gg X + Y))) \\ &= a(r) \cdot (Y + \tau(t) \cdot (t \gg X + t \gg Y)) \\ &= a(r) \cdot (Y + \tau(t) \cdot t \gg X) \\ &= a(r) \cdot (Y + \tau(t) \cdot X)\end{aligned}$$

□

We close this section with one theorem.

Theorem 3.8 *(Soundness and Completeness)* $p, q \in \mathcal{T}_\rho$

$$\text{BPA}\rho\delta\tau + \text{ATB} \vdash p = q \quad \Longleftrightarrow \quad p \leftrightarroweq_{rb} q$$

The proof of this theorem is postponed to section 6.

4 Branching Bisimulation and Integration

The subsections 4.1 till 4.7 are taken from ([FK92]).

4.1 Introduction

Integration is the alternative composition over a continuum of alternatives ([BB91]). We denote the process that can perform an action somewhere in the interval $[1,2\rangle$ (1 included, 2 excluded) by the process term

$$\int_{v\in[1,2\rangle} a(v).$$

Each integral binds a *time variable*, which may occur later on:

$$\int_{v\in[1,2\rangle}(a(v)\cdot\int_{w\in[v+1,v+2]} b(w)) \quad \text{and} \quad \int_{v\in\langle 0,\infty\rangle}(a(v)\cdot b(2\cdot v)).$$

Thus, the bounds of an interval may depend on time variables.

In this paper we take a more restrictive view on integration than in [BB91], called *prefixed* integration ([Klu91a],[FK92]). We require that every integral $\int_{v\in V}$ is directly followed by an action $a(v)$. Moreover, time variables are chosen from intervals of which the bounds are liniar expressions. Examples of terms which are not in this from are

$$\int_{v\in\{n|n \text{ is prime}\}} a(v) \quad \text{and} \quad \int_{v\in\langle 1,2\rangle} a(1)\cdot b(v).$$

4.2 The Time Domain

For technical reasons we require that each element of our time domain has a unique finite representation. This is clearly not the case for the collection of real numbers. As a time domain we assume from now on a countable subset D of $\mathbb{R}^{\geq 0}$ such that

- $\mathbb{Q}_{\geq 0} \cup \{\infty\} \subseteq D$
- $+, \dot{-} : D \times D \to D$
- $\cdot : \mathbb{Q}_{>0} \times D \to D$

By abuse of notation we may refer to D by $[0,\infty]$.

4.3 Bounds and Intervals

The definition of a bound becomes

$$b ::= t \mid v \mid b+b \mid b \dot{-} b \mid r\cdot b$$

where $t \in D$, $r \in \mathbb{Q}_{>0}$ and $v \in TVar$.

TVar denotes an infinite, countable set of *time variables.* Let $t \in [0,\infty]$, $r \in \langle 0,\infty\rangle$ and $v \in TVar$. The set *Bound* of *bounds*, with typical elements b, b_1, b_2, is defined by

$$b \ ::= \ t \mid v \mid b_1 + b_2 \mid b_1 \dot{-} b_2 \mid r\cdot b$$

where $\dot{-}$ denotes the monus function, i.e. if $t_0 \leq t_1$ then $t_0 \dot{-} t_1 = 0$. In the sequel $\langle\!\![$ and $]\!\!\rangle$ are elements of $\{\langle, [\}$ and $\{\rangle,]\}$ respectively. An interval V is of the form $\langle\!\![b_1, b_2]\!\!\rangle$ with b_1, b_2 bounds. For $b \in Bound$ the set of time variables occurring in b is denoted by $tvar(b)$. Of course $tvar(\langle\!\![b, c]\!\!\rangle) = tvar(b) \cup tvar(c)$. The set of intervals is denoted by Int, its subset of *time-closed* intervals, i.e. intervals for which $tvar(V) = \emptyset$, is denoted by Int^{cl}. On Int two operators sup, inf are defined. Let $V = \langle\!\![b_0, b_1]\!\!\rangle$

$$\begin{array}{lll} \text{if } V \neq \emptyset & \text{then} & \inf(V) = b_0 \text{ and } \sup(V) = b_1 \\ \text{if } V = \emptyset & \text{then} & \inf(V) = \sup(V) = 0 \end{array}$$

A *condition* is a logical expression built from bounds, for example $v < 1$ or $2v \leq w+4$. We may consider expressions like $b \in V$ and $V = \emptyset$ as abbreviations for conditions. Similarly, we may consider $b < V$ as a condition which reduces to true if b is smaller than every point in V. For example, $1 < [2,3]$ and $v < \langle v, v+1]$. Finally, we may write $V < W$ where V and W are both intervals; meaning that it is always true that an arbitrary point in W is larger than every point in V. We require in this case as well that W is not empty.

For example $\langle v, v+1\rangle < \langle 2v, 10]$ reduces to the condition $1 < v < 5$.

4.4 Process Terms

Let $a \in A_\delta$, $t \in [0, \infty\rangle$, $v \in TVar$, V an interval and b a bound. The set $\mathcal{T}_{\mathrm{BPA}\rho\mathrm{I}\tau\delta}$ of *process terms*, is abbreviated by $\mathcal{T}_I$.

$$p ::= a(t) \mid \int_{v \in V} a(v) \mid \int_{v \in V} (a(v) \cdot p) \mid p_1 + p_2 \mid p_1 \cdot p_2 \mid b \gg p \mid p \gg b \mid p@b$$

We use a *scope convention*, saying that if we do not write scope brackets, then the scope of an integral is as large as possible. Thus we write $\int_{v \in V} a(v) \cdot p$ for $\int_{v \in V}(a(v) \cdot p)$. We can define inductively the collection $FV(p)$ of free time variables appearing in a process term p. FV(p) contains those variables which are not bound by any of the integrals in p. A term p with $FV(p) = \emptyset$ is called a *time-closed* term. The subset of time closed terms is denoted by $\mathcal{T}_I^{cl}$. We (re)define

$$\begin{array}{lll} U(\int_{v \in V} a(v)) & = & \sup(V) \\ U(\int_{v \in V}(a(v) \cdot p)) & = & \sup(V) \\ \\ initact(\int_{v \in V} a(v)) & = & V(\text{ if } a \neq \delta) \\ initact(\int_{v \in V} \delta(v)) & = & \emptyset \\ initact(\int_{v \in V}(a(v) \cdot p)) & = & V(\text{ if } a \neq \delta) \\ initact(\int_{v \in V}(\delta(v) \cdot p)) & = & \emptyset \\ \\ L(p) & = & \sup(initact(p)) \\ S(p) & = & \inf(initact(p)) \\ \\ time(\int_{v \in V} a(v)) & = & 0 \\ time(\int_{v \in V}(a(v) \cdot p)) & = & 0 \end{array}$$

We define *(syntactic) summand inclusion*, denoted by $\sqsubseteq$, by taking the transitive closure over the following clauses.

$$\begin{array}{lcl} & & \delta \sqsubseteq p \\ p \simeq p' & \Longrightarrow & p \sqsubseteq p' \\ V \subseteq V' & \Longrightarrow & \int_{v \in V} a(v) \sqsubseteq \int_{v \in V'} a(v) \\ V \subseteq V' & \Longrightarrow & \int_{v \in V}(a(v) \cdot p) \sqsubseteq \int_{v \in V'}(a(v) \cdot p) \\ p \sqsubseteq q, p' \sqsubseteq q & \Longrightarrow & p + p' \sqsubseteq q \\ p \sqsubseteq q & \Longrightarrow & p \sqsubseteq q + q' \end{array}$$

It is quite easy to determine whether $p \sqsubseteq p'$ or not.

4.5 α-conversion

Process terms are considered modulo α-conversion; we extend the definition of $\simeq$ as follows.

$$\begin{array}{rcl} \int_{v \in V} a(v) & \simeq & \int_{w \in V} a(w) \\ \text{if } w \text{ does not occur in } p \text{, then } \int_{v \in V} a(v) \cdot p & \simeq & \int_{w \in V} a(w) \cdot p[w/v] \end{array}$$

We take the transitive closure of this relation.

$$\frac{r \in V}{\int_{v \in V} a(v) \xrightarrow{a(r)} \surd} \qquad \frac{r \in V}{\int_{v \in V} a(v) \cdot p \xrightarrow{a(r)} r \gg p[r/v]}$$

Table 6: Additional Action Rules for Integral Construct

4.6 SOS and Strong Bisimulation

The action rules for the integral are given in Table 6. These rules express exactly that an integral is a continuum of alternatives. Definition 2.3 of strong bisimulation and Definition 3.3 of branching bisimulation still apply when we restrict ourselves to time closed terms. In [Klu91a] it is proven that $\leftrightarrow_s$ is a congruence over $\mathcal{T}_I^{cl}$. Moreover, it is proven that $\leftrightarrow_s$ over $\mathcal{T}_I^{cl}$ is completely axiomatized by the theory BPAρI$\tau\delta$ which is given in Table 7. In Table 7 it is assumed that $p, q \in \mathcal{T}$ with $FV(p+q) \subseteq \{v\}$ and $X, Y \in \mathcal{T}^{cl}$ and P is of the form $a(v)$ or $a(v) \cdot p$.

4.7 Branching Bisimulation and Integration

We extend the definitions of $\leftrightarrow_b$ and $\leftrightarrow_{rb}$ to $\mathcal{T}_I^{cl} \times \mathcal{T}_I^{cl}$ in the obvious way.

Proposition 4.1 $\leftrightarrow_{rb}$ *is a congruence over* $\mathcal{T}_I^{cl}$

INT0		$\int_{v\in[t,t]} a(v) = a(t)$
INT1	$V = V_0 \cup V_1$	$\int_{v\in V_0} P + \int_{v\in V_1} P = \int_{v\in V} P$
INT2		$\int_{v\in\emptyset} P = \delta$
INT3a		$\int_{v\in V}(a(v)) \cdot Y = \int_{v\in V} a(v) \cdot Y$
INT3b		$\int_{v\in V}(a(v) \cdot p) \cdot Y = \int_{v\in V} a(v) \cdot (p \cdot Y)$
INT4		$\forall t \in V \;\; X + a(t) \cdot p[t/v] = X \implies X + \int_{v\in V} a(v) \cdot p = X$
ATA1a		$a(0) = \delta$
ATA1b		$a(\infty) = \delta(\infty)$
AI2		$\int_{v\in V} \delta(v) = \delta(sup(V))$
AI3		$\int_{v\in V} \delta(v) \cdot p = \delta(sup(V))$
AI4	$t \leq \sup(V)$	$\int_{v\in V} P + \delta(t) = \int_{v\in V} P$
AI5		$\int_{v\in V} a(v) \cdot p = \int_{v\in V} a(v) \cdot (v \gg p)$
AI7		$t \gg \int_{v\in V} P = \int_{v\in V\cap(t,\infty]} P + \delta(t)$
AI8		$t \gg (X + Y) = (t \gg X) + (t \gg Y)$
AI9		$\int_{v\in V} P \gg t = \int_{v\in V\cap(0,t)} P + \delta(\min(\sup(V), t))$
AI10		$(X + Y) \gg t = (X \gg t) + (Y \gg t)$
AI11		$\int_{v\in V} P@t = \int_{v\in V\cap[t,t]} P + \delta(t)$
AI12		$(X + Y)@t = (X@t) + (Y@t)$

Table 7: An axiom system for BPA$\rho\delta$I

Proof. The check that $\underline{\leftrightarrow}_{rb}$ is indeed an equivalence is left to the reader. We have to show for $p_1, p_2, q_1, q_2 \in \mathcal{T}_I^{cl}$ that

$$\begin{array}{lll} p_1 \underline{\leftrightarrow}_{rb} q_1 \wedge p_2 \underline{\leftrightarrow}_{rb} q_2 & \implies & p_1 + p_2 \underline{\leftrightarrow}_{rb} q_1 + q_2, \;\; p_1 \cdot p_2 \underline{\leftrightarrow}_{rb} q_1 \cdot q_2, \\ p_1 \underline{\leftrightarrow}_{rb} q_1 & \implies & r \gg p_1 \underline{\leftrightarrow}_{rb} r \gg q_1, \;\; p_1 \gg r \underline{\leftrightarrow}_{rb} q_1 \gg r \text{ and} \\ & & p_1@r \underline{\leftrightarrow}_{rb} q_1@r \end{array}$$

Assume $\mathcal{R}_1 : p_1 \underline{\leftrightarrow}_{rb} q_1$ and $\mathcal{R}_2 : p_2 \underline{\leftrightarrow}_{rb} q_2$ then we give for each of the cases above a branching bisimulation $\mathcal{R}$. The check that $\mathcal{R}$ is indeed a branching bisimulation for each case is left to the reader.

- $p_1 + p_2 \underline{\leftrightarrow}_{rb} q_1 + q_2$

$$\mathcal{R} = \mathcal{R}_1 \cup \mathcal{R}_2$$

- $p_1 \cdot p_2 \underline{\leftrightarrow}_{rb} q_1 \cdot q_2$

$$\begin{array}{rcll} \mathcal{R} & = & \{(p \cdot p_2, q \cdot q_2) \mid p\mathcal{R}_1 q\} & \cup \\ & & \{(r \gg p_2, r \gg q_2) \mid r > 0\} & \cup \\ & & \mathcal{R}_2 & \end{array}$$

- $r \gg p_1 \leftrightarrow_{rb} r \gg q_1$
$$\mathcal{R} = \mathcal{R}_1 \cup \{(r \gg p_1, r \gg q_1)\}$$
- $p_1 \gg r \leftrightarrow_{rb} q_1 \gg r$
$$R = \mathcal{R}_1 \cup \{(p_1 \gg r, q_1 \gg r)\}$$
- $p_1@r \leftrightarrow_{rb} q_1@r$
$$R = \mathcal{R}_1 \cup \{(p_1@r, q_1@r)\}$$

□

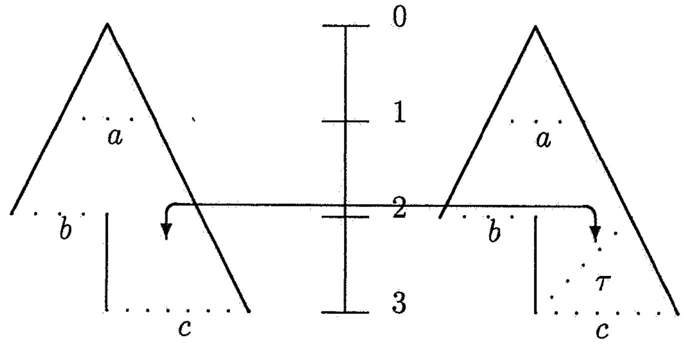
4.8 A first generalization of ATB

First we give a few examples of (non) rooted branching bisimilar pairs. In the calculus without integrals we have the following typical example

$$a(1) \cdot (b(2) + c(3)) \quad \leftrightarrow_{rb} \quad a(1) \cdot (b(2) + \tau(2) \cdot c(3))$$

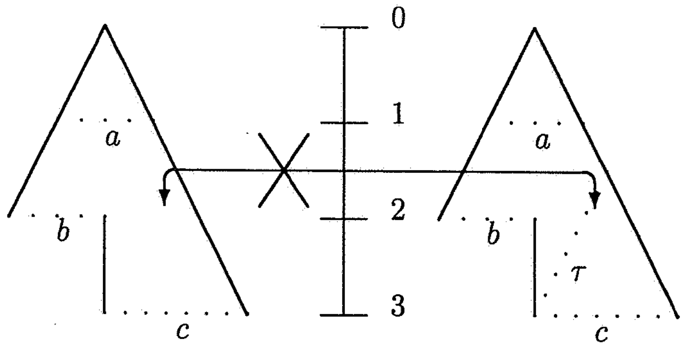
We can adapt this example to

$$a(1) \cdot (b(2) + c(3)) \qquad \leftrightarrow_{rb} \qquad a(1) \cdot (b(2) + \textstyle\int_{v \in \langle 2,3 \rangle} \tau(v) \cdot c(3))$$



Again, if we can make a choice for the $c(3)$ before time 2 (by allowing a τ before 2) this pair is not bisimilar anymore:

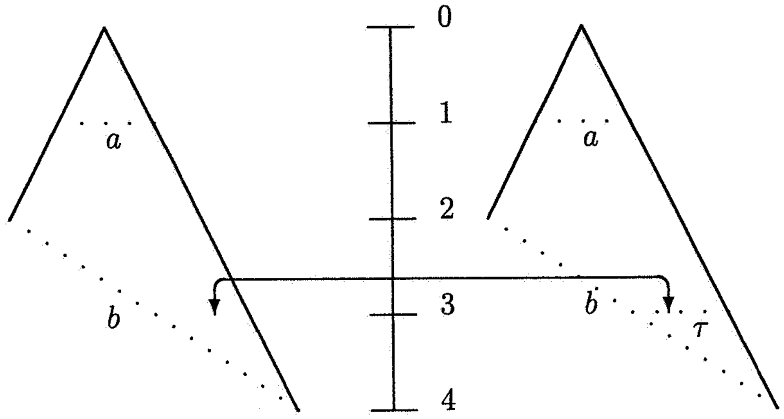
$$a(1) \cdot (b(2) + c(3)) \qquad \not\leftrightarrow_{b} \qquad a(1) \cdot (b(2) + \textstyle\int_{v \in \langle 1.5,3 \rangle} \tau(v) \cdot c(3))$$



We have to redefine the conditions for the timed τ laws, since they have to deal with intervals. ATB is first generalized to ATBI'; the I in the name refers to *integration* and we need the prime since we will generalize ATBI' even further.

ATBI'	$v \in V \Longrightarrow v < \sup(W) \ \wedge\ W < U(X)$
	$\int_{v \in V} a(v) \cdot X = \int_{v \in V} a(v) \cdot (X \gg \inf(W) + \int_{w \in W} \tau(w) \cdot X)$

So, we have $a(1) \cdot \int_{v \in \langle 2,4 \rangle} b(v) \quad \underline{\leftrightarrow}_{rb} \quad a(1) \cdot (\int_{v \in \langle 2,3]} b(v) + \tau(3) \cdot \int_{v \in \langle 3,4 \rangle} b(v))$



This identity can be derived as follows. First we show that the condition

$$v \in V \Longrightarrow v < \sup(W) \ \wedge\ W < U(X)$$

for $V = [1,1]$,$W = [3,3]$ and $U(X) = 4$ reduces to true.

$$v \in [1,1] \Longrightarrow v < 3 \ \wedge\ [3,3] < 4$$

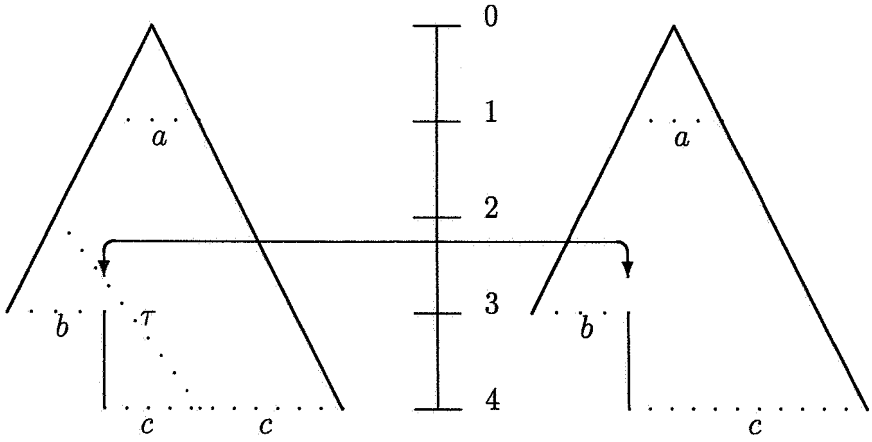
reduces to $1 < 3 < 4$ which is obviously true. Now we may apply ATBI':

$$\begin{aligned} a(1) \cdot \textstyle\int_{v \in \langle 2,4 \rangle} b(v) &= a(1) \cdot (\textstyle\int_{v \in \langle 2,4 \rangle} b(v) \gg 3 + \tau(3) \cdot \textstyle\int_{v \in \langle 2,4 \rangle} b(v)) \\ &= a(1) \cdot (\textstyle\int_{v \in \langle 2,4 \rangle \cap \langle 0,3]} b(v) + \tau(3) \cdot (3 \gg \textstyle\int_{v \in \langle 2,4 \rangle} b(v)) \\ &= a(1) \cdot (\textstyle\int_{v \in \langle 2,3]} b(v) + \tau(3) \cdot \textstyle\int_{v \in \langle 2,4 \rangle \cap \langle 3,\infty]} b(v)) \\ &= a(1) \cdot (\textstyle\int_{v \in \langle 2,3]} b(v) + \tau(3) \cdot \textstyle\int_{v \in \langle 3,4 \rangle} b(v)) \end{aligned}$$

4.9 The Timed Branching Law

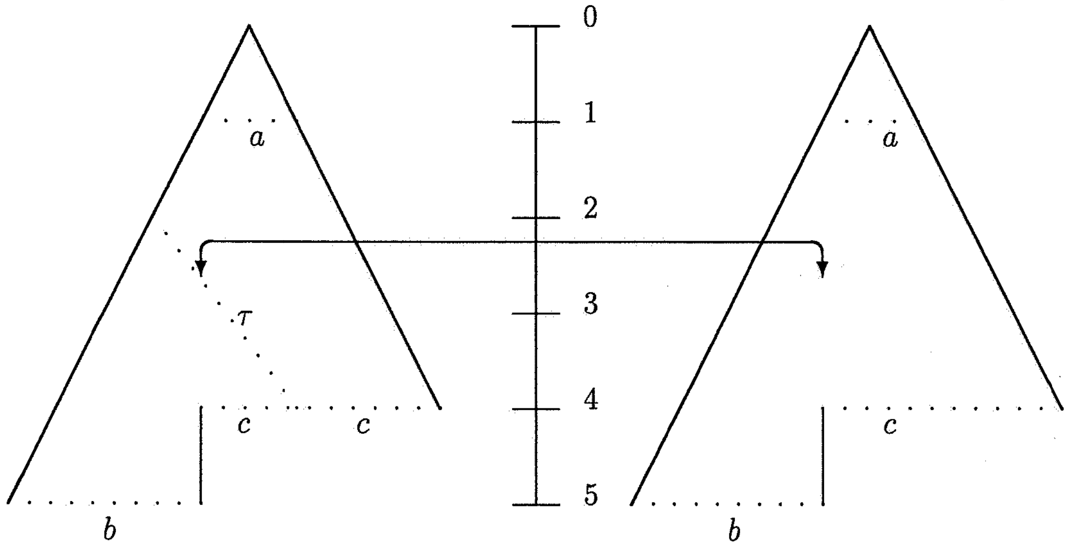
However, not all identities can be covered by ATBI'.

$a(1) \cdot (\int_{v \in \langle 2,4 \rangle} \tau(v) \cdot (b(3) + c(4)) + c(4)) \underline{\leftrightarrow}_{rb} \qquad a(1) \cdot (b(3) + c(4))$



and

$a(1) \cdot (\int_{v \in \langle 2,4 \rangle} \tau(v) \cdot (b(5) + c(4)) + c(4)) \underline{\leftrightarrow}_{rb} \qquad a(1) \cdot (b(5) + c(4))$

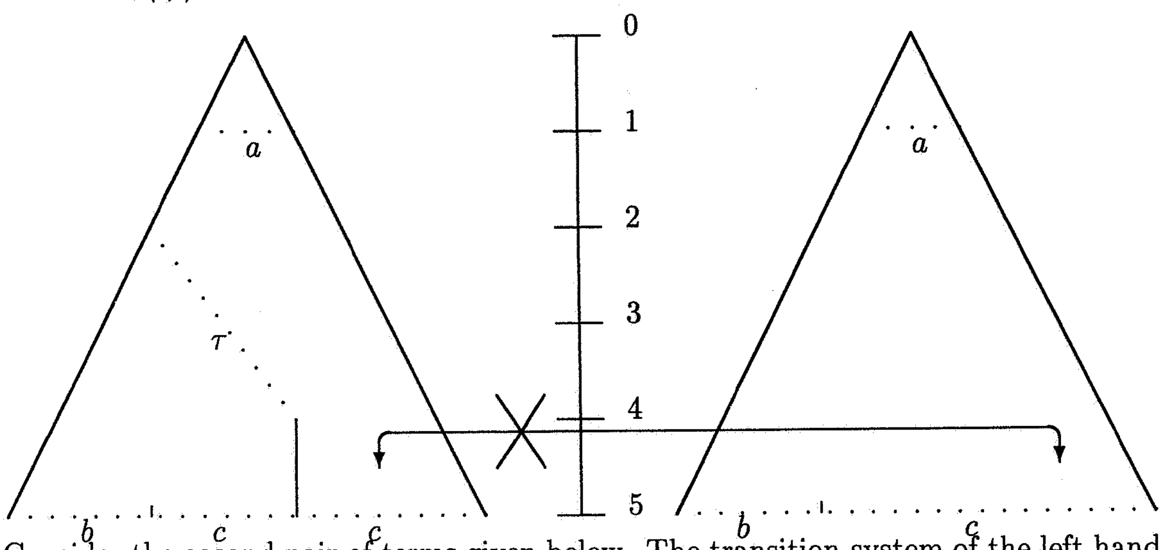


These identities looke like an instance of the second (untimed) τ-law:

$$\text{T2} \quad Z \cdot (\tau \cdot (X + Y) + X) = Z \cdot (X + Y)$$

When adding time to this law we have to be very careful with the conditions on all the intervals as is shown in the following example which differs only with the previous example in that repect that in the term on the left hand side both components $b(5)$ and $c(5)$ can idle after $\inf(\langle 2, 4])$.

$a(1) \cdot (\int_{v \in \langle 2,4 \rangle} \tau(v) \cdot (b(5) + c(5)) + c(5)) \not\leftrightarrow_{rb} \quad a(1) \cdot (b(5) + c(5))$



Consider the second pair of terms given below. The transition system of the left hand side of the second pair has a deadlock at 4 caused by the possible execution of the τ at 4. The transition system of the right hand side of the first pair does not have such a deadlock.

$$a(1) \cdot (\int_{v \in \langle 2,4 \rangle} \tau(v) \cdot (b(3) + c(4)) + c(4)) \quad \leftrightarrow_{rb} \quad a(1) \cdot (b(3) + c(4))$$
$$a(1) \cdot (\int_{v \in \langle 2,4]} \tau(v) \cdot (b(3) + c(4)) + c(4)) \quad \not\leftrightarrow_{rb} \quad a(1) \cdot (b(3) + c(4))$$

Consider the following two pairs, of which the first is a bisimilar one. In the transition system of the left hand side of the second pair the choice for doing the b might be done at 10, while at the right hand side it may be postponed until 11.

$$a(1) \cdot (\int_{v \in \langle 1,10 \rangle} \tau(v) \cdot (\int_{w \in \langle 1,20]} b(w) + \int_{z \in \langle 0,10 \rangle} c(z)) + \int_{w \in \langle 1,20]} b(w))$$
$$\leftrightarrow_{rb}$$
$$a(1) \cdot (\int_{w \in \langle 1,20]} b(w) + \int_{z \in \langle 0,10 \rangle} c(z))$$

$$a(1) \cdot (\int_{v \in \langle 1,10 \rangle} \tau(v) \cdot (\int_{w \in \langle 1,20]} b(w) + \int_{z \in \langle 0,11 \rangle} c(z)) + \int_{w \in \langle 1,20]} b(w))$$
$$\not\leftrightarrow_{rb}$$
$$a(1) \cdot (\int_{w \in \langle 1,20]} b(w) + \int_{z \in \langle 0,11 \rangle} c(z))$$

ATBI	$v \in V \Longrightarrow v < \sup(W) \wedge W < U(X+Y) \wedge \min(U(X), U(Y)) \leq \sup(W)$
	$\int_{v \in V} a(v) \cdot ((\int_{w \in W} \tau(w)) \cdot (X+Y) + X) = \int_{v \in V} a(v) \cdot (X + \inf(W) \gg Y)$

Theorem 4.2 *(Soundness)* $p, q \in \mathcal{T}_I^{cl}$

$$\mathrm{BPA}\rho\mathrm{I}\delta + \mathrm{ATBI} \vdash p = q \quad \Longleftrightarrow \quad p \leftrightarrow_{rb} q$$

Proof. Omitted □

In Section 6 we will prove the completeness, that is $p \leftrightarrow_{rb} q$ implies BPA$\rho\delta$I + ATBI $\vdash p = q$.

4.10 Embedding untimed branching bisimulation into the timed one

Baeten & Bergstra have given an embedding of BPAδ into BPA$\rho\delta$ which interpretes every (untimed) atomic action a as $\int_{v \in \langle 0,\infty \rangle} a(v)$. Let us denote this translation by

$$RT : \mathcal{T}_u \longrightarrow \mathcal{T}_\rho.$$

We have

$$\forall p, q \in \mathcal{T}_u \qquad p \leftrightarrows_s q \quad \Longleftrightarrow \quad RT(p) \leftrightarrows_s RT(q).$$

However this does not hold any more after adding the τ and working with branching bisimulation equivalence. For example, in the untimed case we have $a \cdot \tau \leftrightarrows_{rb} a$ but

$$\int_{v \in \langle 0,\infty \rangle} a(v) \cdot \int_{v \in \langle 0,\infty \rangle} \tau(v) \quad \not\leftrightarrows_{rb} \quad \int_{v \in \langle 0,\infty \rangle} a(v)$$

This is caused by a difference in the design decisions of BPAδ respectively BPA$\rho\delta$. In BPAδ the intuition is that the execution of an action may take some time. This is exactly the motivation behind the first τ law $X \cdot \tau = X$. But in real time process algebra $\int_{v \in V} a(v)$ executes an a at some point $r \in V$ and terminates successfully at r. A possible embedding of BPA$\tau\delta$ into BPA$\rho\tau\delta$ is now given by interpreting an untimed atomic action a as $\int_{v \in \langle 0,\infty \rangle} a(v) \cdot \int_{v \in \langle 0,\infty \rangle} \tau(v)$, expressing that the action a is executed somewhere in time after which it terminates some time later. This translation has been pointed out by Jos Baeten ([Bae92]). Let us denote this translation by RT_τ, then we have the following Theorem.

Proposition 4.3 $p, q \in \mathcal{T}_u$

$$p \leftrightarrows_{rb} q \quad \Longleftrightarrow \quad RT_\tau(p) \leftrightarrows^t_{rb} RT_\tau(q)$$

Proof. Omitted

5 Basic and Conditional Terms and their Strong Normal Forms

5.1 Introduction

In this section we present some concepts which are necessary to prove the completeness for branching bisimulation in the next section.

Basic terms, conditional terms and the theory CTA have been introduced in [Klu91a]. Strong normal forms have been introduced in [Fok91]. However, this section is mainly taken from [FK92], which is an improved compilation of [Klu91a] and [Fok91]. Only Section 5.5 is new here.

5.2 Basic terms

First we define a notion of a *widest scope term*, that is a term p without subterms of the form $q \cdot q'$. In [Klu91a] it is shown that every term can be rewritten into a widest scope term. The term $(\int_{v\in V} a(v)) \cdot p$ is first rewritten into $(\int_{w\in V} a(w)) \cdot p$ where $w \notin FV(p)$ and then it is rewritten into $\int_{w\in V}(a(w) \cdot p)$. Note that in this latter term p is in the scope of the $\int_{w\in V}$. Finally we use the *scope convention* by not writing the scope brackets anymore, thus obtaining $\int_{w\in V} a(w) \cdot p$.

A basic term will be a widest scope term that does not contain "redundant" parts. For example, the terms $a(5) \cdot b(6) + \delta(1)$ and $a(5) \cdot (b(6) + c(4))$ contain parts that can be removed by application of the axioms of BPA$\rho\delta$; they are derivably equal to the term $a(5) \cdot b(6)$, which does not contain any redundant parts. In [Klu91a],[FK92] completeness for strong bisimulation is proven for basic terms, since the transitions of a basic term corresponds directly with its syntax. By showing that each process term is equal to a basic term, completeness follows for all closed terms.

Basic terms have two properties:

- **Basic terms have ascending time stamps.** For the term $a(t) \cdot p$ to be a basic term, it is required that p is a basic term that either starts after t or is equal to δ. So if $\mathcal{B}$ denotes the set of basic terms, then

$$\begin{array}{llll} a(1) \cdot b(2) & \in \mathcal{B} & a(1) \cdot (b(2) + c(1)) & \notin \mathcal{B} \\ a(1) \cdot \delta & \in \mathcal{B} & a(1) \cdot \delta(1) & \notin \mathcal{B} \end{array}$$

- **Basic terms do not contain redundant deadlocks.** A basic term will not contain timed deadlocks that do not contribute to its deadlock behaviour. In other words, if $p + \delta(t)$ is a basic term then it can do a $\delta(t)$-transition to $\surd$. For example:

$$\begin{array}{llll} a(1) + \delta(2) & \in \mathcal{B} & a(1) + \delta(2) + \delta(1) & \notin \mathcal{B} \\ \delta(3) + \delta(3) & \in \mathcal{B} & a(3) + \delta(3) & \notin \mathcal{B} \end{array}$$

Note that $\mathcal{B} \subseteq \mathcal{T}^{cl}$. In a basic term only prefixed multiplication is used and the time shift $\gg$ does not occur. If p is a basic term, then we allow $p + \delta$ to be a basic term. For convenience of notation we extend the definition of $\simeq$ (equality modulo A1,2) by putting $p \simeq p + \delta$.

5.3 Conditions and conditional Terms

Until now we have mostly considered time-closed terms. But if we want to prove that every time-closed term has a basic form, we have to consider terms with free time variables as well. Therefore we will introduce the notion of a *conditional* term. A conditional term determines for each substitution of real values for the free time variables a time-closed term. For example, if we consider the term $a(5) \cdot b(v)$, which has a free time variable v, we will associate to it the following conditional term:

if the context assigns a value $t \leq 5$ to v, then deliver $a(5) \cdot \delta$
if the context assigns a value $t > 5$ to v, then deliver $a(5) \cdot b(t)$

This conditional term will be denoted as follows (the notation $:\!\longrightarrow$ is taken from [BB90]):

$$p \simeq \begin{array}{l} \{v \leq 5 \quad :\!\longrightarrow \quad a(5) \cdot \delta\} \\ + \\ \{v > 5 \quad :\!\longrightarrow \quad a(5) \cdot b(v)\} \end{array}$$

For every substitution σ that assigns a real number to v we have $\sigma(p) \in \mathcal{B}$. A substitution either satifies a condition α or not.

A conditional basic term will be of the form $\sum_i \alpha_i :\!\longrightarrow p_i$ where $\{\alpha_i\}$ form a partition. That means that $\cup_i \alpha_i = tt$ and $i \neq i'$ implies $\alpha_i \ \wedge \ \alpha_{i'} = ff$. Thus each substitution σ satisfies exactly one condition in the partition, say α_i, and we require that $\sigma(p) = \sigma(p_i)$ yields a basic term.

All these notions have been defined formally in [FK92].

5.4 The theory of CTA

The equational theory CTA (Conditional Terms Algebra) consists of 'conditional variants' of axioms of BPA$\rho\delta$I together with six new axioms CON1-6. For example the law

$$t \leq \sup(V) \quad \int_{v \in V} P + \delta(t) = \int_{v \in V} P$$

is now formulated by

$$\int_{v \in V} P + \delta(b) = \begin{array}{lll} \{b \leq sup(V) & :\!\longrightarrow & \int_{v \in V} P\} \\ + & & \\ \{o.w. & :\!\longrightarrow & \int_{v \in V} P + \delta(b)\} \end{array}$$

Examples of typical conditional axioms are

$$\begin{array}{ll} \text{CON1} & \{tt \ :\!\longrightarrow \ p\} = p \\ \text{CON2} & \{ff \ :\!\longrightarrow \ p\} = \delta \end{array}$$

There are two axioms which allow us to combine the conditions and to form a partition:

CON3 $\{\alpha :\!\longrightarrow \Sigma_i\{\beta_i :\!\longrightarrow p_i\}\} = \Sigma_i\{\alpha \wedge \beta_i :\!\longrightarrow p_i\}$

CON4 $\{\alpha :\!\longrightarrow p\} + \{\beta :\!\longrightarrow q\} = \{\alpha \wedge \beta :\!\longrightarrow p + q\} + \{\alpha \wedge \neg\beta :\!\longrightarrow p\} + \{\neg\alpha \wedge \beta :\!\longrightarrow q\}$

Finally we have an axiom which allows us to lift conditions over an integral.

CON6 If $\{\alpha_i \wedge v \in W_i\}$ is a partition and $v \notin tvar(\alpha_i) \cup tvar(W_i)$, then

$$\textstyle\int_{v\in V} a(v) \cdot \Sigma_i\{\alpha_i \wedge v \in W_i :\to p_i\} = \Sigma_i\{\alpha_i :\to \int_{v\in V\cap W_i} a(v) \cdot p_i\}$$

We have to require that $\{\alpha_i \wedge v \in W_i\}$ is a partition; otherwise we would change the branching structure; i.e. we would introduce the right distributivity $Z \cdot (X + Y) = Z \cdot X + Z \cdot Y$.

Using these conditional axioms it is possible to reason about terms containing free time variables; we can prove now that each process term can be reduced to a basic conditional term (see [FK92]).

In order to use CON6 for general conditional terms we need the following lemma. Note that it would not be true if we would allow expressions like v^2 to be bounds. A condition α is refined by $\{\beta_j\}$ if $\cup_j \beta_j = \alpha$ and $j \neq j'$ implies $\beta_j \wedge \beta_{j'} = ff$.

Lemma 5.1 *Let $v \in TVar$. A condition α always has a refinement of the form $\{\beta_j \wedge v \in V_j\}$, where $tvar(\beta_j) \cup tvar(V_j) \subseteq tvar(\alpha)\backslash\{v\}$.*

Proof. See [FK92].

An example of an identity in CTA is:

Example 5.2

$$\mathrm{CTA} \vdash \int_{v\in V} a(v) \cdot p = \int_{v\in V} a(v) \cdot \{v \in V :\to p\}$$

In the sequel we abbreviate

$$\mathrm{CTA} \vdash \{\alpha :\to p\} = \{\alpha :\to q\} \quad \text{by} \quad \mathrm{CTA}, \alpha \vdash p = q$$

Furthermore, the term $\{\alpha :\to p\}+\{\neg\alpha :\to q\}$ may be denoted by $\{\alpha :\to p\}+\{o.w. :\to q\}$.

5.5 Basic Terms and free time variables

In the previous section we have seen that some τ's may be removed by the branching law. In fact in the completeness proof we use a rewriting in which for example the basic term $a(1) \cdot \int_{v\in\langle 2,3\rangle} \tau(v) \cdot \int_{w\in\langle v,5\rangle} b(w)$ is rewritten into $a(1) \cdot \int_{w\in\langle 2,5\rangle} b(w)$. We make use of the following identity

$$\int_{v\in V} a(v) \cdot p = \int_{v\in V} a(v) \cdot p[\inf(V)/v]$$

which is derivable if the variable v occurs only as lower bound of initial integrals of p. To formalize this we define $FV^*(p) \subseteq FV(p)$ where p is supposed to be a widest scope term which does not contain bounds like $b + 0$ or $b + b' - b'$.

$$\begin{array}{llll}
FV^*(\int_{v\in\langle\!|b,b'|\!\rangle} a(v)) & = & tvar(b') & \text{if} \quad b \in TVar \\
FV^*(\int_{v\in\langle\!|b,b'|\!\rangle} a(v)) & = & tvar(b + b') & \text{o.w} \\
FV^*(\int_{v\in\langle\!|b,b'|\!\rangle} a(v) \cdot p) & = & tvar(b') \cup FV(p)\backslash\{v\} & \text{if} \quad b \in TVar \\
FV^*(\int_{v\in\langle\!|b,b'|\!\rangle} a(v) \cdot p) & = & tvar(b + b') \cup FV(p)\backslash\{v\} & \text{o.w} \\
FV^*(p + q) & = & FV^*(p) + FV^*(q) &
\end{array}$$

Proposition 5.3 *Let p be a widest scope term where $v \notin FV^*(p)$ then*

$$\mathrm{CTA} \vdash \int_{v \in V} a(v) \cdot p = \int_{v \in V} a(v) \cdot p[\inf(V)/v]$$

Proof. Omitted

5.6 Strong Normal Froms

In this section the machinery is introduced to reduce each process term p to its *strong normal form*, denoted by $snf(p)$. Process terms are considered modulo commutativity and associativity of the $+$, which is denoted by $\simeq$. For each time-closed process term its strong normal form will again be a time-closed process term. In the next section we will prove that if $p, q \in \mathcal{T}^{cl}$ with $\mathrm{BPA}\rho\delta\mathrm{I} \vdash p = q$, then $snf(p) \simeq snf(q)$. Thus it is possible to check in a finite number of steps whether two terms $p, q \in \mathcal{T}^{cl}$ are equal in $\mathrm{BPA}\rho\delta\mathrm{I}$ or not; first p and q are reduced to strong normal form, and then a finite computation is carried out to see if $snf(p) \simeq snf(q)$ or not.

Below we discuss four (groups) of rewrite rules which will be expressions of the form $p \longrightarrow q$ with $p, q \in \mathcal{C}$. Every term is reduced to its strong normal form using these rewrite rules. This strong normal form will again be a basic conditional term. For details we refer to [FK92].

- **Reducing conditions.** We define four rewrite rules that reduce conditions, they are based on the conditional axioms CON1-CON5 and CON6, see Subsection 5.4.
- **Reducing bounds.** In order to reduce time-closed process terms to a unique strong normal form, it is necessary to reduce bounds. For example, the equation $2v \dot{-} 1 = (v \dot{-} 1) + v$ holds for $v \geq 1$, but is untrue for $0 \leq v < 1$. So the equality

  $$\int_{v \in \langle t, \infty \rangle} a(v) \cdot \int_{w \in \langle 2v \dot{-} 1, 2v \rangle} b(w) \;=\; \int_{v \in \langle t, \infty \rangle} a(v) \cdot \int_{w \in \langle (v \dot{-} 1) + v, 2v \rangle} b(w)$$

  holds for $t \geq 1$, but not for $0 \leq t < 1$.

  In [FK92] the notion of a bound is generalized to that of a *conditional* bound, which allows expressions of the form $\{\alpha :\rightarrow b\}$ with α a condition and b a bound. Furthermore, it is described how a bound can be reduced to a normal form, which is a conditional bound with all its monus signs replaced by minus signs. This is done by applying the rewrite rule

  $$b \dot{-} c \;\longrightarrow\; \{b > c :\rightarrow b - c\} \;+\; \{b \leq c :\rightarrow 0\}$$

  Now we can give rewrite rule 5, which enables us to reduce the bounds occurring in a process term. Let b be a bound that, using the construction described in [FK92], is reduced to $\Sigma_i\{\alpha_i :\rightarrow b_i\}$. Then

5.	$\int_{v \in \langle\!\lbrack b, c \rbrack\!\rangle} P \;\longrightarrow\; \Sigma_i\{\alpha_i :\rightarrow \int_{v \in \langle\!\lbrack b_i, c \rbrack\!\rangle} P\}$

  We have a similar rewrite rule for if b has normal form b' and symmetric rules for if c has normal form $\Sigma_i\{\alpha_i :\rightarrow c_i\}$ or c'.

- **Substituting redundant variables.** A variable occurring in a process term can be *redundant* in the sense that only one value can be substituted for it. For example:

$$\int_{v\in[1,1]} a(v)\cdot\int_{w\in\langle v,v+1\rangle} b(w) = \int_{v\in[1,1]} a(v)\cdot\int_{w\in\langle 1,2\rangle} b(w)$$

  So in order to reduce time-closed process terms to a unique strong normal form, it is necessary to substitute the only possible value for such a redundant variable.

  The following rewrite rule reduces process terms of the form $\int_{v\in[b,b]} a(v)\cdot p$. Ensure by applying α-conversion that $v \notin tvar(b)$ and also that none of the variables in $tvar(b)$ are bound by integral signs occurring in p. Then

$$6. \quad \int_{v\in[b,b]} a(v)\cdot p \longrightarrow \int_{v\in[b,b]} a(v)\cdot p[b/v]$$

- **Reducing double terms.** The main problem of reducing time-closed process terms to a unique strong normal form is getting rid of the 'double terms'. We first give two rewrite rules, based on axiom INT1, to deal with this problem.

  Let '$V_0 \sim V_1$' denote that '$V_0 \cup V_1$ is an interval'. Note that this is a condition, i.e. it can be described by a finite number of (in)equalities between bounds.

$$7a. \quad \int_{v\in V_0} a(v) + \int_{v\in V_1} a(v) \longrightarrow$$

$$\{V_0 \sim V_1 \;:\!\longrightarrow\; \int_{v\in V_0\cup V_1} a(v)\} + \{o.w. \;:\!\longrightarrow\; \int_{v\in V_0} a(v) + \int_{v\in V_1} a(v)\}$$

$$7b. \quad \int_{v\in V_0} a(v)\cdot p + \int_{v\in V_1} a(v)\cdot p \longrightarrow$$

$$\{V_0 \sim V_1 \;:\!\longrightarrow\; \int_{v\in V_0\cup V_1} a(v)\cdot p\} + \{o.w. \;:\!\longrightarrow\; \int_{v\in V_0} a(v)\cdot p + \int_{v\in V_1} a(v)\cdot p\}$$

However, this rule is not sufficient in all cases. Consider the following two examples.

Example 5.4

$$\int_{v\in\langle 0,1\rangle} a(v)\cdot\int_{w\in\langle v,v+1\rangle} b(w) \;+\; \int_{v\in[1,2\rangle} a(v)\cdot\int_{w\in\langle v,2\rangle} b(w)$$

$$= \int_{v\in\langle 0,1]} a(v)\cdot\int_{w\in\langle v,v+1\rangle} b(w) \;+\; \int_{v\in\langle 1,2\rangle} a(v)\cdot\int_{w\in\langle v,2\rangle} b(w)$$

Although these terms are equal, they can not be rewritten by rule 7*b*.

Example 5.5

$$\int_{v\in\langle 0,2\rangle} a(v \cdot \int_{w\in\langle v,v+1\rangle} b(w) + \int_{v\in[1,1]} a(v)\cdot\int_{w\in\langle 1,2\rangle} b(w) = \int_{v\in\langle 0,2\rangle} a(v)\cdot\int_{w\in\langle v,v+1\rangle} b(w)$$

Again b· ch terms can not be rewritten by rule 7*b*.

A logical solution for avoiding such situations seems to be to allow only integration over open intervals and over intervals consisting of one point. However, the following example shows that this restraint does not work.

Example 5.6

$$\int_{v\in\langle 0,1\rangle} a(v)\cdot\int_{w\in\langle v,v+1\rangle} b(w) + \int_{v\in[1,1]} a(v)\cdot\int_{w\in\langle 1,2\rangle} b(w) + \int_{v\in\langle 1,2\rangle} a(v)\cdot\int_{w\in\langle v,v+1\rangle} b(w)$$

$$= \int_{v\in\langle 0,2\rangle} a(v)\cdot\int_{w\in\langle v,v+1\rangle} b(w)$$

Both terms are equal and satisfy the restraint on intervals, but they can not be rewritten by rule 7*b*. We therefore introduce two rewrite rules to deal with Examples 5.4 and 5.5.

Example 5.4 shows that we need a reduction

$$\int_{v\in\langle b,c\rangle\!\rangle} a(v)\cdot p + \int_{v\in V} a(v)\cdot q \longrightarrow \int_{v\in[b,c\rangle\!\rangle} a(v)\cdot p + \int_{v\in V} a(v)\cdot q$$

if the following two statements are true:

1. $b \in V$.
2. $p[b/v] \simeq q[b/v]$.

The first statement is clearly a condition. In [FK92] it is shown that $p[b/v] \simeq q[b/v]$ is a condition as well, the idea is that only finitely many bounds have to be compared.

Let $p, q \in \mathcal{T}$, $b \in \mathit{Bound}$ and $v \in \mathit{TVar}$. Ensure by applying α-conversion that $v \notin \mathit{tvar}(b)$ and that none of the variables occurring in b are bound by integrals occurring in p and q. Example 5.4 can be reduced by the following rewrite rule.

8. $\int_{v\in\langle b,c\rangle\!\rangle} a(v)\cdot p + \int_{v\in V} a(v)\cdot q \quad\longrightarrow$

$$\{p[b/v] \simeq q[b/v] \ \wedge\ b \in V \ :\rightarrow\ \int_{v\in[b,c\rangle\!\rangle} a(v)\cdot p \ +\ \int_{v\in V} a(v)\cdot q\}$$
$$+$$
$$\{o.w \ :\rightarrow\ \int_{v\in\langle b,c\rangle\!\rangle} a(v)\cdot p + \int_{v\in V} a(v)\cdot q\}$$

We also have a symmetric version of this rewrite rule in order to reduce the process term $\int_{v\in\langle\!\langle c,b\rangle} a(v)\cdot p + \int_{v\in V} a(v)\cdot q$.

Similarly, Example 5.5 can be reduced by

$$\begin{array}{lll} 9. & \int_{v\in V} a(v)\cdot p \;+\; \int_{v\in[b,b]} a(v)\cdot q & \longrightarrow \\ & \{p[b/v] \simeq q \;\wedge\; b \in V \;:\!\rightarrow\; \int_{v\in V} a(v)\cdot p\} & \\ & + & \\ & \{o.w. \;:\!\rightarrow\; \int_{v\in V} a(v)\cdot p + \int_{v\in[b,b]} a(v)\cdot q\} & \end{array}$$

In [FK92] an algorithme, based on these rewrite rules, is given which reduces each term p to its strong normal form $snf(p)$.

The normal form of the bound $b[t/v]$ is denoted by $b[t/v]*$, by abuse of notation we allow ourselves to denote this bound in normal form by $b(t)$ as well, whenever the variable v is clear from the context. Note that if two bounds b, b' in normal form are not of the same form, then there is at most one t such that $b(t) \simeq b'(t)$. Here $\simeq$ denotes syntactical equivalence on bounds modulo associativity and commutativity of the $+$.

In the sequel $p[t/v]*$ denotes the term $p[t/v]$ in which all bounds (in which v occurs) have been rewritten to their normal form. For example $a(v+1)[2/v]* \simeq a(3)$.

The following two Lemma's will be used in the proof of Theorem 5.9 and of Theorem 6.13. A process term that is a subterm of a strong normal form is called a *subnormal form.*

Lemma 5.7 *Let p and q be subnormal forms. If $p[t/v]* \simeq q[t/v]*$ for some $v \in TVar$ and infinitely many $t \in [0,\infty\rangle$, then $p \simeq q$.*

Proof. See [FK92].

Lemma 5.8 *Let $\int_{v\in V} a(v)\cdot p$ be a strong normal form. Then $p[t/v]*$ is a strong normal form for all $t \in V$.*

Proof. See [FK92].

Theorem 5.9 $p, q \in \mathcal{T}^{cl}$.

$$\mathrm{BPA}\rho\delta\mathrm{I} \vdash p = q \quad\Longrightarrow\quad snf(p) \simeq snf(q)$$

Proof. See [FK92].

6 Branching Normal Forms and the Completeness Proof

6.1 Intro uction

In this section we extend the techniques which are presented in the previous section by taking branching bisimulation into account.

First we give a conditional version of the branching law. Then we give three rewrite rules which, together with the rewrite rules for strong normal forms, give for each term p its *branching normal form* which is denoted by $bnf(p)$. We prove that if two branching normal forms are branching bisimilar then they must be isomorphic. Finally we prove the completeness.

6.2 The Conditional Branching Law

The branching law

$$\text{ATBI} \quad \int_{v\in V} a(v)((\int_{w\in W} \tau(w)) \cdot (X+Y)+X) = \int_{v\in V} a(v) \cdot (X + \inf(W) \gg Y)$$

has as condition that

$$v \in V \Longrightarrow v < \sup(W) \ \wedge \ W < U(X+Y) \ \wedge \ \min(U(X), U(Y)) \leq \sup(W)$$

reduces to tt. We can turn this into a conditional law:

$$\begin{array}{ll} \text{BC} & \int_{v\in V} a(v) \cdot ((\int_{w\in W} \tau(w)) \cdot (X+Y)+X) \\ = & \int_{v\in V} a(v) \cdot \left(\begin{array}{l} \{v < \sup(W) \ \wedge \ W < U(X+Y) \ \wedge \ \min(U(X), U(Y)) \leq \sup(W) \\ \qquad :\to \ X + \inf(W) \gg Y\} \\ + \\ \{o.w. \ :\to \ (\int_{w\in W} \tau(w)) \cdot (X+Y)+X\} \end{array} \right) \end{array}$$

This condition is quite involved, however if we restrict ourselves to basic terms then it can be reduced. If p is a widest scope term then the statement that p is a basic term is a condition, which can be used as a premise in a derivation.

Proposition 6.1 *Let* $z \simeq (\int_{w\in W} \tau(w)) \cdot ((p+q)+p)$ *then*

$$\int_{v\in V} a(v) \cdot z \textit{ is basic} \quad \Longrightarrow \quad v < \sup(W) \ \wedge \ W < U(p+q)$$

Proof. Omitted.

The conditional axiom BC can also be applied the other way around:

Proposition 6.2

CTA + BC $\vdash$

$$\textstyle\int_{v\in V} a(v)\cdot(X+\inf(W)\gg Y)$$

$$= \textstyle\int_{v\in V} a(v)\cdot\left(\begin{array}{l}\{v<\sup(W)\ \wedge\ W<U(X+Y)\ \wedge\ \min(U(X),U(Y))\leq\sup(W)\\ \qquad :\rightarrow\ (\int_{w\in W}\tau(w))\cdot(X+Y)+X\}\\ +\\ \{o.w.\ \ :\rightarrow\ \ X+\inf(W)\gg Y\}\end{array}\right)$$

Proof. Omitted

6.3 Rewrite Rules for Branching Normal Forms

6.3.1 Introducing τ's for each moment of choice

In order to rewrite each term into a canonical form we have to rewrite all hree terms below into the same form.

Example 6.3

$$a(1)\cdot(\tau(2)\cdot b(3)+c(3))=a(1)\cdot(\tau(2)\cdot b(3)+\tau(2)\cdot c(3))=a(1)\cdot(b(3)+\tau(2)\cdot c(3))$$

This example shows us that it is not possible to have as least as possible τ's in a term. Therefore we add a τ for each moment of choice; in the above example boith outermost terms are rewritten into the term in the middle.

In case of integrals we have to do some m ore work. If we consider the term p where

$$p\simeq\int_{v\in\langle 0,5\rangle}a(v)+\int_{v\in[2,6\rangle}b(v)$$

then we can split $\langle 0,6\rangle$ into $\langle 0,2]$, $\langle 2,5]$ and $\langle 5,6\rangle$, such that the potential of p does not change by idling within one of the resulting intervals. In order to reflect these intervals of potential at the syntactical level we introduce a τ for each moment of choice. A moment of choice is the upper or lower bound of one of these intervals. (Only for $U(p)$ and $S(p)$ we do not add a τ.) Thus p is rewritten into

$$p'\simeq\int_{v\in\langle 0,2]}a(v)+b(2)+\tau(2)\cdot(\int_{v\in\langle 2,5\rangle}a(v)+\int_{v\in\langle 2,5]}b(v)+\tau(5)\cdot\int_{v\in\langle 5,6\rangle}b(v)))$$

In case of a term with free time variables such as

$$q\simeq\int_{w\in\langle v,v+1\rangle}a(w)+\int_{w\in\langle v,2v\rangle}b(w)$$

we introduce a partition. Each condition in this partition is an assumption on the ordering of the bounds of the term q. Therefore each condition determines which τ's have to be added.

Example 6.4

$$q' \simeq \left(\begin{array}{lll} \{v+1 = 2v & :\rightarrow & \int_{w\in\langle v,v+1\rangle} a(w) + \int_{w\in\langle v,v+1\rangle} b(w)\} \\ + & & \\ \{v+1 < 2v & :\rightarrow & \int_{w\in\langle v,v+1\rangle} a(w) + \int_{w\in\langle v,v+1]} b(w) + \tau(v+1)\cdot\int_{w\in\langle v+1,2v\rangle} b(w)\} \\ + & & \\ \{v+1 > 2v & :\rightarrow & \int_{w\in\langle v,2v]} a(w) + \int_{w\in\langle v,2v\rangle} b(w) + \tau(2v)\cdot\int_{w\in\langle 2v,v+1\rangle} b(w)\} \end{array} \right)$$

In the rewrite step we consider a variable v from which we assume that it is the time variable which is bound by the integral in which scope the term itself will be put. Therefore we take only that part of the term which starts after v according to the associated condition.

6.3.2 Partitioning a term

As we have seen in Example 6.4 we have to consider all possible orderings on the bounds of a term p. Therefore we need the notion of an *ordered partition* of a finite set S. The sequence $\langle S_1, \ldots, S_n\rangle$ of subsets of S is an ordered partition if $\{S_1, \ldots, S_n\}$ is a partition of S.

Consider a set B of bounds. We construct the set of conditions on B, each defining an ordering on the bounds.

- Construct all possible ordered partitions of B, let us assume that there are m different partitions of B.
- For each ordered partition $\langle B_1, ..., B_n\rangle$ we construct a condition α. If $b, b' \in B_i$ for some i then α contains $b = b'$. If $b \in B_i$ and $b' \in B_{i'}$ where $i < i'$ then α contains $b < b'$. In this way we obtain the partition $\{\alpha_j\} = \{\alpha_1, \ldots, \alpha_m\}$ associated to B.

In the following we will denote α by $B_1 < ... < B_n$ whenever α is constructed from the ordered partition $\langle B_1, ..., B_n\rangle$ of B.

Assume p is of the form

$$\sum_i \int_{v\in \{\!|_i s_i, u_i |\!\}_i} P_i$$

where each P_i is of the form $a(v)$ or $a(v)\cdot p'$. Consider $\alpha = B_1 < \ldots < B_n$ where $\langle B_1, \ldots, B_n\rangle$ is one of the ordered partitions of $B = bounds(p) \cup \{v\}$. For each B_l we take a representant $b_l \in B_l$. We construct $\tilde{p}(\alpha, l)$ for $l < n$:

$$\begin{array}{rlll} \tilde{p}(\alpha,l) \simeq & \sum_{i:\alpha\Rightarrow s_i\le b_l \,\wedge\, u_i > b_{l+1}} & \int_{v\in\langle b_l, b_{l+1}]} & P_i \\ + & \sum_{i:\alpha\Rightarrow s_i\le b_l \,\wedge\, u_i = b_{l+1}} & \int_{v\in\langle b_l, b_{l+1}|\!\}_i} & P_i \\ + & \sum_{i:\alpha\Rightarrow s_i = b_{l+1} = u_i \,\wedge\, \{\!|_i = [\,\wedge\, |\!\}_i =]} & \int_{v\in[b_{l+1}, b_{l+1}]} & P_i \\ + & \sum_{i:\alpha\Rightarrow s_i = b_{l+1} < u_i \,\wedge\, \{\!|_i = [} & \int_{v\in[b_{l+1}, b_{l+1}]} & P_i \end{array}$$

And for $l \ge n$ we take $\tilde{p}(\alpha, l) \simeq \delta$. We have the following proposition.

Proposition 6.5 $l < n$

$$\mathrm{CTA}, \alpha \vdash \tilde{p}(\alpha, l) = b_l \gg p \gtrsim b_{l+1}$$

Proof. By construction.
Next, we define

$$\begin{array}{rcll} v(\alpha) & = & l & \text{such that } v \in B_l \\ u(\alpha, p) & = & l & \text{such that } l \text{ is the smallest index } \geq v(\alpha) \\ & & & \quad \text{with } l' \geq l \text{ implies } \tilde{p}(\alpha, l') \simeq \delta \end{array}$$

Note that $\alpha \Longrightarrow U(v \gg p) = b_{u(\alpha,p)}$. Finally we define $p(\alpha, l)$.

$$\begin{array}{rcll} p(\alpha, l) & \simeq & \delta & \text{if } l \geq u(\alpha, p) \\ & \simeq & \tilde{p}(\alpha, l) & \text{if } l = u(\alpha, p) - 1 \\ & \simeq & \tilde{p}(\alpha, l) + \tau(b_{l+1}) \cdot p(\alpha, l+1) & \text{if } l < u(\alpha, p) - 1 \end{array}$$

The rewrite rule which adds all possible τ's is

$$\boxed{10. \quad p \quad \longrightarrow \quad \Sigma_j \{\alpha_j \; :\!\longrightarrow \; \tilde{p}(\alpha_j, v(\alpha_j))\}}$$

where $\{\alpha_j\}$ is the partition associated to $bounds(p) \cup \{v\}$.

6.3.3 Removing τ's

We have to introduce a rewrite rule which handles the following examples.

Example 6.6

$$a(1) \cdot (\int_{w \in \langle 1,3 \rangle} \tau(w) \cdot (\int_{z \in \langle w,3 \rangle} a(z) + b(3)) + \int_{w \in \langle 1,3 \rangle} a(w) + b(3)) \quad = \quad a(1) \cdot (\int_{w \in \langle 1,3 \rangle} a(w) + b(3))$$

A similar example is the following term which has a $\tau(2)$ which does not determine a choice.

Example 6.7

$$\begin{array}{l} a(1) \cdot (\int_{w \in \langle 1,2 \rangle} \tau(w) \cdot (\int_{z \in \langle w,3 \rangle} a(z) + b(3)) + \int_{w \in \langle 1,2 \rangle} a(w) + \tau(2) \cdot (\int_{z \in \langle 2,3 \rangle} a(z) + b(3))) \\ \qquad = \quad a(1) \cdot (\int_{z \in \langle 1,3 \rangle} a(z) + b(3)) \end{array}$$

Therefore we introduce the rule

$$\boxed{\begin{array}{lll} 11 \quad (w \notin FV^*(p)) \quad \int_{w \in \langle\!|b,b'|\!\rangle} \tau(w) \cdot p + p' + \tau(b') \cdot q & \longrightarrow & \\ \qquad \{p' + \tau(b') \cdot q \sqsubseteq p[b/w] & :\!\longrightarrow & p\} \\ \qquad + & & \\ \qquad \{p' + q \sqsubseteq p[b/w] & :\!\longrightarrow & p\} \\ \qquad + & & \\ \qquad \{o.w. & :\!\longrightarrow & \int_{w \in \langle\!|b,b'|\!\rangle} \tau(w) \cdot p + p' + \tau(b') \cdot q\} \end{array}}$$

Example 6.6 is rewritten according to the clause $p' + \tau(b') \cdot q \sqsubseteq p[b/w]$ and Example 6.7 is rewritten according to the clause $p' + q \sqsubseteq p[b/w]$.

Note that p' may be δ. We have a similar rule for the case without the summand $\tau(b') \cdot q$.

Finally we n‹ ed a rule to remove all τ's which do not determine a moment of choice really. So in ca ɜ for each summand $\int_{v \in V} P$ of p there is an "adjacent" summand $\int_{v \in V'} P$ of q such that ` $\sim V'$ then $p + \tau(b) \cdot q$ is rewritten into the term in which each summand of p has been glued to its adjacent summand of q. We have to be carefull that the resulting term z has only terms of the form $\int_{v \in \langle\!|S(z),U(z)|\!\rangle} P$ or $\int_{v \in [U(z),U(z)]} P$.

Example 6.8

$\int_{v \in \langle 1,2]} a(v) + \tau(2) \cdot \int_{v \in \langle 2,3\rangle} a(v)$	is rewritten into	$\int_{v \in \langle 1,3\rangle} a(v)$
$\int_{v \in \langle 1,2]} a(v) + \tau(2) \cdot (\int_{v \in \langle 2,3\rangle} a(v) + b(3))$	is rewritten into	$\int_{v \in \langle 1,3\rangle} a(v) + b(3)$
$\int_{v \in \langle 1,2]} a(v) + b(2) + \tau(2) \cdot (\int_{v \in \langle 2,3\rangle} a(v) + \int_{v \in \langle 2,3\rangle} b(v))$	is not rewritten	

$p \sim q$ denotes that for each summand $\int_{v \in V} P$ of p there is a summand $\int_{v \in V'} P$ of q such that $V \sim V'$ and vice versa. $p \sim q$ is clearly a condition but it is not the yet the right one. We change the condition $p \sim q$ slightly, obtaining $p \sim^* q$, by requiring that in case $S(p) < U(p)$ p does not have a summand $\int_{v \in [U(p),U(p)]} P$. Moreover summands of the form $\int_{v \in [U(q),U(q)]} P$ of q are not considered. Clearly $p \sim^* q$ is again a condition.

The term $(p+q) \sim$ denotes the term $p+q$ in which every summand of p is taken together with its adjacent summand of q. If $p \sim^* q$ then $(p+q) \sim$ is well defined.

$$12 \quad p + \tau(b) \cdot q \quad \longrightarrow \quad \begin{array}{l} \{p \sim^* q \quad :\longrightarrow \quad (p+q)_\sim\} \\ + \\ \{o.w. \quad :\longrightarrow \quad p + \tau(b) \cdot q\} \end{array}$$

6.4 The construction of Branching Normal Forms

Now we have to combine these rewrite rules together with the construction for strong normal forms. We take a basic term p and we perform the steps given below. Each step is concluded by taking all the conditions together into a partition ; this will not be stated explicitly anymore. Therefore, at the beginning of each step below, except the first one, the term under consideration is of the form $\sum_i \{\alpha_i :\longrightarrow p_i\}$ where $\{\alpha_i\}$ is a partition.

1. We replace every summand $\int_{v \in V} a(v) \cdot p'$ by $\int_{v \in V} a(v) \cdot bnf(p')$. Next we lift the conditions (which occur only at the toplevel of $bnf(p')$ and which form a partition) over $\int_{v \in V}$.

2. We apply the same steps of the construction for normal forms, so we reduce the bounds of the intervals to their normal forms, we substitute redundant variables and we take double summands together.

3. We apply rule 10 on each p_i. Assume that we have introduced a term of the form

$$q_0 + [\tau(b_1) \cdot (q_1 + \tau(b_2) \cdot ... \tau(b_n) \cdot p_n)]$$

(where $p[+p']$ denotes a term which is either of the form p or $p + p'$) then for each level $k \geq 0$ we have to do the following

(a) Rule 10 may have introduced redundant variables (see the construction of $\tilde{p}(\alpha, l)$) which have to be removed by Rule 6.

(b) Apply Rule 11 and 12.

(c) We lift the resulting partition over the $\tau(b_k)$ whenever $k > 0$.

Let CTA_{br} denote the axiom system obtained by extending CTA with BC and the rewrite rules for strong normal forms which are not based on axioms from CTA but from BPA$\rho\delta$I.

Proposition 6.9

$$\mathrm{CTA}_{br}, \int_{v\in V} a(v)\cdot p \textit{ is basic} \vdash \int_{v\in V} a(v)\cdot p = \int_{v\in V} a(v)\cdot \sum_j \{\alpha_j :\rightarrow \tilde{p}^{\alpha_j}_{v(\alpha_j)}\}$$

Proof. Take an arbitrary j and take $\alpha = \alpha_j$. According to proposition 5.1 there is a refinement $\{\beta_i \wedge v \in V_i'\}$ of α such that v does not occur in any of the β_j's or V_i''s. Take $V \cap V_i' = W_i$. It is sufficient to show that

$$\beta_i \vdash \int_{v\in W_i} a(v)\cdot p = \int_{v\in W_i} a(v)\cdot p(\alpha, v(\alpha))$$

Note that $\alpha \Longrightarrow \alpha'$ implies that $\mathrm{CTA}_{br} \vdash p = \{\alpha' :\rightarrow p\}$ and we have:

$$\begin{array}{l} \beta_i \vdash \\ \int_{v\in W_i} a(v)\cdot p \\ = \int_{v\in W_i}(a(v)\cdot\{\beta_i \wedge v\in V_i :\rightarrow p\}) \\ = \int_{v\in W_i}(a(v)\cdot\{\beta_i \wedge v\in V_i :\rightarrow \{\alpha :\rightarrow p\}\}) \\ = \int_{v\in W_i}(a(v)\cdot\{\alpha :\rightarrow p\}) \end{array}$$

We use as well $(b = U(p)) \vdash b \gg p = \delta(b)$.

First we prove that for l with $v(\alpha) \leq l \leq u(\alpha, p)$

$$\beta_i \vdash \int_{v\in W_i} a(v)\cdot p = \int_{v\in W_i} a(v)\cdot (p \gtrsim b_l + p(\alpha, l))$$

by induction on $u(\alpha,p) - l$. There are three cases to consider.

1. $v(\alpha) \leq l = u(\alpha, p)$. This case is trivial since $\alpha \Longrightarrow U(p) = b_l$ and $p \gtrsim U(p) = p$. Moreover $p(\alpha, u(\alpha, p)) = \delta$.

2. $v(\alpha) \leq l = u(\alpha, p) - 1$.

$$\begin{array}{l} \mathrm{CTA}_{br}, \beta_i \vdash \\ \int_{v\in W_i} a(v)\cdot p \\ = \int_{v\in W_i} a(v)\cdot p \gtrsim b_l + b_l \gg p \gtrsim b_{l+1} \\ = \int_{v\in W_i} a(v)\cdot p \gtrsim b_l + \tilde{p}(\alpha, l) \\ = \int_{v\in W_i} a(v)\cdot p \gtrsim b_l + p(\alpha, l) \end{array}$$

3. $v(\alpha) \leq l < u(\alpha,p) - 1$. Note that $\alpha \Longrightarrow U(p(\alpha, l+1)) = b_{l+2} > b_{l+1}$

$$\begin{aligned}
&\text{CTA}_{br}, \beta_i \vdash \\
&\textstyle\int_{v \in W_i} a(v) \cdot p \\
&= \textstyle\int_{v \in W_i} a(v) \cdot (p \gtrsim b_{l+1} + p(\alpha, l+1)) \\
&= \textstyle\int_{v \in W_i} a(v) \cdot ((p \gtrsim b_{l+1} + p(\alpha, l+1)) \gtrsim b_{l+1} + \tau(b_{l+1}) \cdot (p \gtrsim b_{l+1} + p(\alpha, l+1))) \\
&= \textstyle\int_{v \in W_i} a(v) \cdot (p \gtrsim b_{l+1} + \tau(b_{l+1}) \cdot p(\alpha, l+1)) \\
&= \textstyle\int_{v \in W_i} a(v) \cdot (p \gtrsim b_l + b_l \gg p \gtrsim b_{l+1} + \tau(b_{l+1}) \cdot p(\alpha, l+1)) \\
&= \textstyle\int_{v \in W_i} a(v) \cdot (p \gtrsim b_l + \tilde{p}(\alpha, l) + \tau(b_{l+1}) \cdot p(\alpha, l+1)) \\
&= \textstyle\int_{v \in W_i} a(v) \cdot (p \gtrsim b_l + p(\alpha, l))
\end{aligned}$$

We continue

$$\begin{aligned}
&\text{CTA}_{br}, \beta_i \vdash \\
&\textstyle\int_{v \in W_i} a(v) \cdot p \\
&= \textstyle\int_{v \in W_i} a(v) \cdot (p \gtrsim v + p(\alpha, v(\alpha))) \\
&= \textstyle\int_{v \in W_i} a(v) \cdot (v \gg p \gtrsim v + v \gg p(\alpha, v(\alpha))) \\
&= \textstyle\int_{v \in W_i} a(v) \cdot p(\alpha, v(\alpha))
\end{aligned}$$

□

We have a similar proposition for the Rules 11 and 12.

Proposition 6.10 *If $z \longrightarrow z'$ by Rule 11 or 12 then*

$$\text{CTA}_{br}, \textstyle\int_{v \in V} a(v) \cdot z \textit{ is basic } \vdash \textstyle\int_{v \in V} a(v) \cdot z = \textstyle\int_{v \in V} a(v) \cdot z'$$

Proof. Omitted

Proposition 6.11

$$\text{CTA}_{br}, \int_{v \in V} a(v) \cdot p \textit{ is basic } \vdash \int_{v \in V} a(v) \cdot p = \int_{v \in V} a(v) \cdot bnf(p)$$

Proof. We have to prove that each step in the construction of a branching normal form is a derivation in CTA_{br}.

In the first step we replace each summand $\int_{v \in V} a(v) \cdot p'$ by $\int_{v \in V} a(v) \cdot bnf(p')$ for which there is a derivation by induction. Then we lift the conditions, using CON6.

In the second step we reduce the term to a strong normal form for which there is a derivation since for each of the Rules 1-10 there is an associated axiom in CTA_{br}.

Next, we have to prove that whenever p_k is rewritten into p' by Rule 10 then

$$\text{CTA}_{br}, \int_{v \in V} a(v) \cdot p \text{ is basic } \vdash \int_{v \in V} a(v) \cdot \sum_i \{\alpha_i \;:\!\longrightarrow\; p_i\} = \int_{v \in V} a(v) \cdot \sum_i \{\alpha_i \;:\!\longrightarrow\; p_i'\}$$

where $p_i' \simeq p'$ for $i = k$ and $p_i' \simeq p_i$ otherwise.

First we construct the refinement $\{\beta_j \wedge v \in V_j\}_i$ for each α_i. Apply CON6 such that the term is rewritten into one of the form $\sum_l \{\beta_l \;:\!\longrightarrow\; \int_{v \in V \cap V_l} a(v) \cdot q_l\}$. For each l there is a pair (i, j) such that $q_l \equiv p_i$ and $\beta_l = \beta_j^i$. Since we deal with basic terms only, β_l implies that $\int_{v \in V \cap V_l} a(v) \cdot q_l$ is basic. Apply Rule 10 on all q_l such that l is associated to (k, j) for some j. By proposition 6.9 this is a derivation in CTA_{br}. Apply CON6 again, but now in the other direction. We replace the refinement $\{\beta_j \wedge v \in V_j\}_i$ by the original partition $\{\alpha_i\}$.

For Rules 11 and 12 we can prove it similarly by using proposition 6.10. □

So CTA_{br} is the axiom system in which each basic term is reduced to its branching normal form. We relate CTA_{br} with $\mathrm{BPA}\rho\delta\mathrm{I}$ by the following Proposition.

Proposition 6.12 $p, q \in T^{cl}$

$$\mathrm{CTA}_{br} \vdash p = q \quad \Longrightarrow \quad \mathrm{BPA}\rho\delta\mathrm{I} \vdash p = q$$

Proof. As in [FK92].

6.5 The Unicity Theorem for Branching Bisimulation

Theorem 6.13 *If p, q are branching normal forms*

$$p \leftrightarrow_b q \quad \Longrightarrow p \simeq q$$

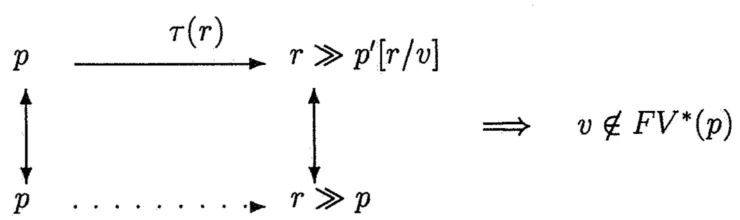
Proof. Firts we prove three Facts for p and q, in the given order.

Take $U(p) = u$.

- Fact 1 $(r < u)$

$$p \xrightarrow{\tau(r)} r \gg p'[r/v] \quad \wedge \quad r \gg p \leftrightarrow_b r \gg p'[r/v] \quad \Longrightarrow \quad v \notin FV^*(p')$$

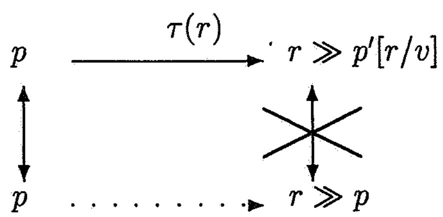
In a picture



- Fact 2 $(r < u)$

$$p \xrightarrow{\tau(r)} r \gg p'[r/v] \quad \Longrightarrow \quad r \gg p \not\leftrightarrow_b r \gg p'[r/v]$$

In a picture



- Fact 3

$$p \leftrightarrow_b q \quad \Longrightarrow \quad p \leftrightarrow_s q$$

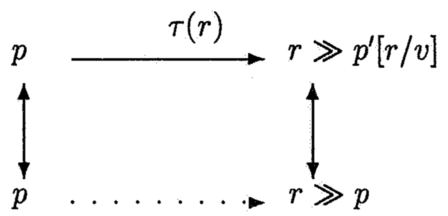
Fact 1 is used in the proof of Fact 2, Fact 2 is used in the proof of Fact 3. The Theorem follows from the Fact 3 and the Unicity Theorem for strong normal forms (Theorem 5.9). In the proofs of Facts 1 and 2 we assume that we have proven the Theorem and its Facts for terms smaller than p. In the proof of Fact 3 we assume that the Theorem and its Facts are proven already for terms smaller than p or q.

If z is a branching normal form and $r < U(z)$ then z_r denotes the term which is constructed from z by replacing each summand $\int_{v \in \langle\!\langle t,t' \rangle\!\rangle} P$ by $\int_{v \in \langle r,t' \rangle\!\rangle} P$ whenever $t < r < t'$. Note that $r \gg z \underline{\leftrightarrow}_s z_r$.

$P(b)$ denotes a term of the form $\int_{v \in [b,b]} a(v) \cdot p$ where $v \notin FV(p)$ or a term of the form $\int_{v \in [b,b]} a(v)$.

- **Proof.** Fact 1

  Assume for $r < u$ that

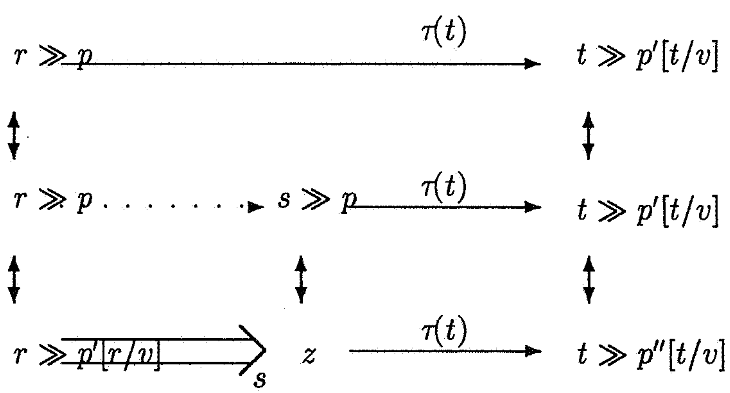


we have to prove that $v \notin FV^*(p')$.

Take an arbitrary $t \in \langle r, u \rangle$ and consider the transition

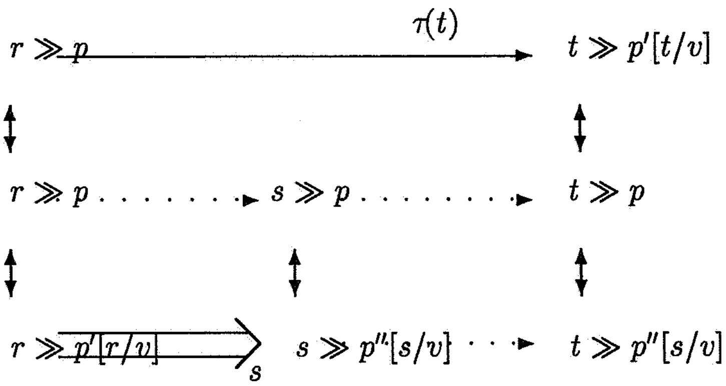
$$r \gg p \xrightarrow{\tau(t)} t \gg p'[t/v],$$

we have to find a corresponding series of transitions starting from $r \gg p'[r/v]$. First we assume that there are z, p'' and s such that



Since p is a basic term $p'[t/v]* \underline{\leftrightarrow}_b p''[t/v]*$, by induction $p'[t/v]* \simeq p''[t/v]*$. But this cannot be the case since $depth(p'') < depth(p')$.

Hence, it must be the case that there are p'' and s such that



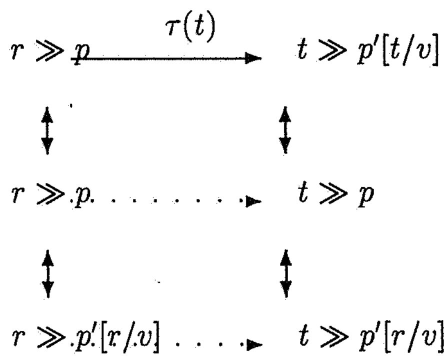
where $U(t \gg p''[s/v]) > t$. Then as well

$$p'[t/v]* \ \underline{\leftrightarrow}_b \ t \gg p'[t/v] \ \underline{\leftrightarrow}_b \ t \gg p''[s/v] \ \underline{\leftrightarrow}_b \ (p''[s/v]*)_t$$

By induction $p'[t/v]* \simeq (p''[s/v]*)_t$, thus ($depth(p') = depth(p'[t/v]*)$ follows from p is a basic term)

$$depth(p') = depth(p'[t/v]*) = depth((p''[s/v]*)_t) = depth(p'')$$

and since $r \gg p' \Longrightarrow_s s \gg p''[s/v]$ the sequence $\Longrightarrow_s$ must be empty. Hence $p'' \equiv p'$ and $s = r$. Thus we have obtained that



Let b, b' be the lowerbound resp. the upperbound of the initial integrals of p', then every summand of p' is either of the form $\int_{v \in \langle\!| b,b' |\!\rangle} P$ or $P(b')$.

There are two cases to consider:

– The case where $b = v$. Note that

$$p'[t/v]* \ \underline{\leftrightarrow}_b \ t \gg p'[t/v] \ \underline{\leftrightarrow}_b \ t \gg p'[r/v] \ \underline{\leftrightarrow}_b \ (p'[r/v]*)_t.$$

By induction $p'[t/v]* \simeq (p'[r/v]*)_t$.
Take an arbitrary summand $\int_{w \in \langle\!| v,b' |\!\rangle} a(w) \cdot z$ of p'. We will show that $v \notin tvar(b') \cup FV(z)$.
Since $p'[t/v]* \simeq (p'[r/v]*)_t$ there is a z^t such that

$$p'[t/v]* \sqsupseteq \int_{w \in \langle\!| t,b'(t) |\!\rangle} a(w) \cdot (z[t/v]*) \simeq \int_{w \in \langle\!| t,b'(r) |\!\rangle} a(w) \cdot (z^t[r/v]*) \sqsubseteq (p'[r/v]*)_t.$$

Then there must be a summand $\int_{w \in \langle\!\langle v,b' \rangle\!\rangle} a(w) \cdot z'$ of p' and an infinite subset $S \subseteq \langle r, u \rangle$ such that

$$\forall t \in S \quad \int_{w \in \langle t, b'(t) \rangle\!\rangle} a(w) \cdot (z[t/v]*) \simeq \int_{w \in \langle t, b'(r) \rangle\!\rangle} a(w) \cdot (z'[r/v]*)$$

from which we conclude that

* $b'(t) = b'(r)$ for more than one t, thus it must be the case that $v \notin tvar(b')$ and thus $b' \in \langle 0, \infty \rangle$.
* $\forall t \in S$ it holds that $z[t/v]* \simeq z'[r/v]*$, hence $z[t/v]* \simeq (z'[r/v]*)[t/v]*$ and by proposition 5.7 $z \simeq z'[r/v]*$. Hence $v \notin FV(z)$.

For summands $\int_{w \in \langle\!\langle v,b' \rangle\!\rangle} a(w)$ and $P(b')$ of p we can conclude similarly that v can not occur in b', resp. in $P(b')$. Hence, $v \notin FV^*(p')$.

- The case where $b \neq v$. Then since p is a basic term $r \leq b(r)$. $r = b(r)$ (for arbitrary r) implies $b = v$ which case already has been considered. So assume $r < b(r)$ then we may assume as well that we have chosen t such that $r < t < b(r)$. Hence $t \gg p'[r/v] \leftrightarrow_b p'[r/v]$. Thus $p'[t/v] * \leftrightarrow_b p'[r/v]*$ and by induction $p'[t/v]* \simeq p'[r/v]*$. Since this holds for all $t \in \langle r, b(r) \rangle$ we conclude $v \notin FV(p')$.

- **Proof.** Fact 2

We assume for $r < u$ that

$$\begin{array}{ccc} p & \xrightarrow{\quad \tau(r) \quad} & r \gg p'[r/v] \\ \updownarrow & & \updownarrow \\ p & \cdots\cdots\cdots\rightarrow & r \gg p \end{array}$$

and we will show that this assumption leads to a contradiction. Take $s = S(p)$.

First we note $v \notin FV^*(p')$ (by Fact 1) and that there is a $z \not\simeq \delta$ such that

$$p \simeq \int_{v \in \langle\!\langle s,u \rangle\!\rangle} \tau(v) \cdot p' + z$$

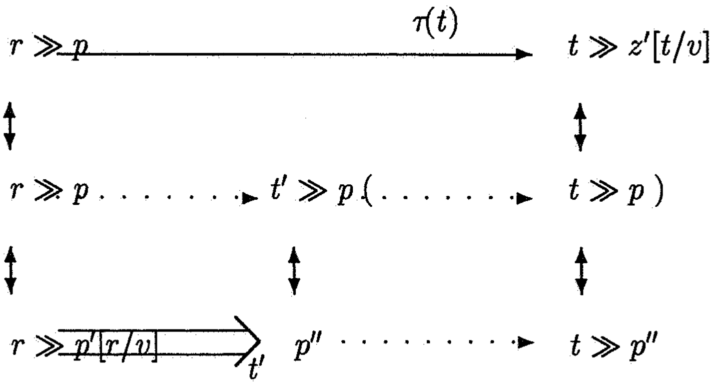
For if $z \simeq \delta$ then Rule 11 could be applied.

Consider

$$z \xrightarrow{a(t)} t \gg z'[t/v] \quad t \in \langle r, u]$$

since we assume $r \gg p'[r/v] \leftrightarrow_b r \gg p$ there must be corresponding series of transitions starting from $r \gg p'[r/v]$.

First we assume that $a = \tau$ and that there is a p'' and t' such that



where $U(t \gg p'') > t$, and where $t \gg p \leftrightarrow_b t \gg z'[t/v]$ is required in case $U(r \gg p) > t$. Consider the transition

$$t' \gg p \xrightarrow{\tau(r')} r' \gg p'[r'/v]$$

for an arbitrary $r' \in \langle t', u \rangle$, which must be matched with a corresponding series of transitions starting from p''. As in the proof of Fact 1 we can deduce, by using induction, that $depth(p') = depth(p'')$. Hence $t' = r$ and $p'' \equiv r \gg p'[r/v]$.

There are two cases to consider, either $s < u$ and $\int_{v \in \langle\!\langle s,u \rangle\!\rangle} \tau(v) \cdot z' \sqsubseteq z$ or $\tau(u) \cdot z' \sqsubseteq z$. In both cases we will show a contradiction.

- If $s < u$ and $\int_{v \in \langle\!\langle s,u \rangle\!\rangle} \tau(v) \cdot z' \sqsubseteq z$.

  Suppose the transition $r \gg p \xrightarrow{\tau(t)} t \gg z'[t/v]$ is matched with an idling, thus $t \gg z'[t/v] \leftrightarrow_b t \gg p'[r/v]$, only for finitely many t. Then there are infinitely many t such that the transition $r \gg p \xrightarrow{\tau(t)} t \gg z'[t/v]$ is matched with a transition $r \gg p'[r/v] \xrightarrow{\tau(t)} t \gg p''[t/v]$, as will be shown in the sequel. Using induction we can show then that $\int_{v \in \langle\!\langle s,u \rangle\!\rangle} \tau(v) \cdot z' \sqsubseteq p'[s/v]$ and thus for all $t \in \langle r, u]$ the transition $r \gg p \xrightarrow{\tau(t)} t \gg z'[t/v]$ can be matched with a transition $r \gg p'[r/v] \xrightarrow{\tau(t)} t \gg p''[t/v]$.

  Hence, we have to consider only the case where the transition $r \gg p \xrightarrow{\tau(t)} t \gg z'[t/v]$ is matched with an idling for infinitely many t. We may assume that $t < u$. Since $v \notin FV^*(p')$ and p is a basic term

  $$p'[t/v] * \leftrightarrow_b t \gg p'[r/v] \leftrightarrow_b t \gg z'[t/v] \leftrightarrow_b z'[t/v]*$$

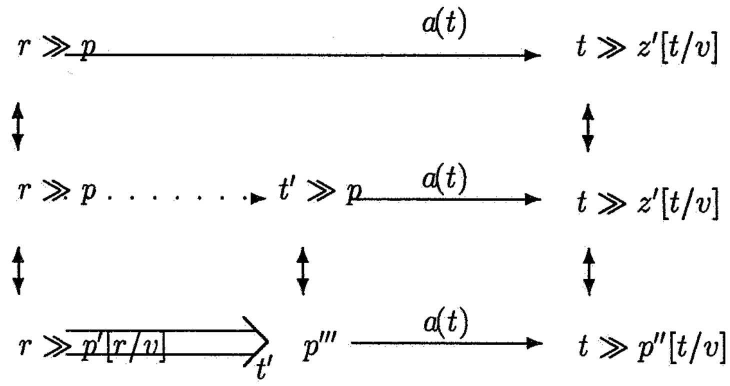
  By induction $p'[t/v]* \simeq z'[t/v] *$. Since this must hold for infinitely many t we have $p' \simeq z'$. But then we have double summands and Rule 7 can be applied. Contradiction.

- If $\tau(u) \cdot z' \sqsubseteq z$ then $t = u$ and since $U(u \gg p'[r/v]) > u$ and $v \notin FV^*(p')$

  $$z' \leftrightarrow_b u \gg z'[u/v] \leftrightarrow_b u \gg p'[r/v] \leftrightarrow_b p'[u/v]*$$

  By induction $z' \simeq p'[u/v]*$ but then the summand $\tau(u) \cdot z'$ and $\int_{v \in \langle\!\langle s,u \rangle\!\rangle} \tau(v) \cdot p'$ can be taken together by either Rule 7*b* or Rule 9. Contradiction.

Thus there must be p'', p''' and t' such that



As in the previous part we can deduce, using induction, that $t' = r$ and $p''' \equiv r \gg p'[r/v]$.

We have obtained that for each transition $r \gg z \xrightarrow{a(t)} t \gg z'[t/v]$ there is a p'' such that $r \gg p'[r/v] \xrightarrow{a(t)} t \gg p''[t/v]$ and $t \gg z'[t/v] \leftrightarrow_b t \gg p''[t/v]$. Since p is a basic term $z'[t/v]* \leftrightarrow_b p''[t/v]*$, by induction $z'[t/v]* \simeq p''[t/v]*$ and thus $t \gg z'[t/v] \leftrightarrow_s t \gg p''[t/v]$. Hence

$$(3) \quad \forall r \in \langle s, u \rangle \quad p'[r/v] + r \gg z \leftrightarrow_s p'[r/v]$$

From which we obtain that

$$p'[s/v] + z \leftrightarrow_s p'[s/v]$$

For if not than there must be a transition

$$z \xrightarrow{a(r')} r' \gg z'[r'/v]$$

which cannot be matched by $p'[s/v]$. But this cannot be possible since we may choose r arbitrary close to s in (3).

Following the reasoning of the Unicity Theorem for strong normal forms we obtain

$$z \sqsubseteq p'[s/v]$$

And $\int_{v \in \langle s,u \rangle} \tau(v) \cdot p' + z$ can be rewritten by Rule 11. Contradiction.

- **Proof.** Fact 3
  Now we are able to prove

$$p \leftrightarrow_b q \quad \Longrightarrow \quad p \leftrightarrow_s q$$

First we prove that for $t < U(p)$

$$p \leftrightarrow_b q \wedge q \Longrightarrow_t q' \quad \wedge \quad t \gg p \leftrightarrow_b q' \quad \Longrightarrow \quad q' \equiv q$$

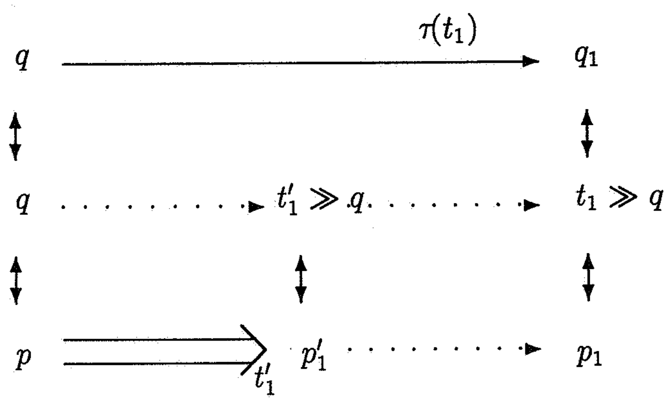
Take $q \xrightarrow{\tau(t_1)} q_1 \ldots \xrightarrow{\tau(t_n)} q_n$ where $q_n \equiv q'$ and assume $n > 0$.

In the sequel $p \Longrightarrow_t^* p'$ denotes that there is a z and s such that $p \Longrightarrow_s z$, $U(z) > t$ and $p' \equiv t \gg z$; thus $\Longrightarrow_t^*$ is nothing more than $\Longrightarrow_t$ where an idling is allowed at the end.

Since $p \leftrightarrow_b q$ we can take $p_1, \ldots, p_n$ such that

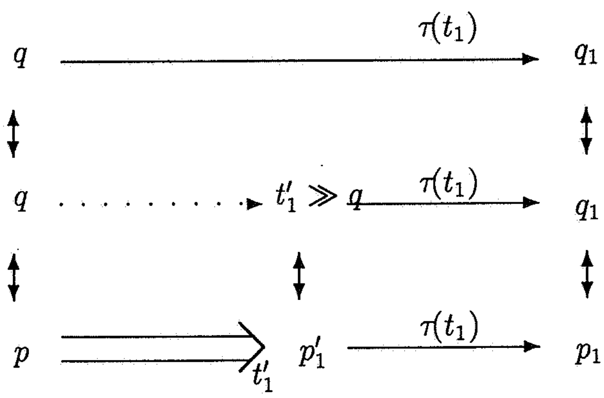
$$p \Longrightarrow_{t_1}^* p_1 \ldots \Longrightarrow_{t_n}^* p_n \quad \wedge \quad p_1 \leftrightarrow_b q_1 \wedge \ldots \wedge p_n \leftrightarrow_b q_n$$

It cannot be the case that there is a p_1' and a t_1' such that



where $p_1 \equiv t_1 \gg p_1'$. Since, together with $t_1 \leq t < U(q)$, it would imply that $t_1 \gg q \leftrightarrow_b q_1$ which is excluded by Fact 2.

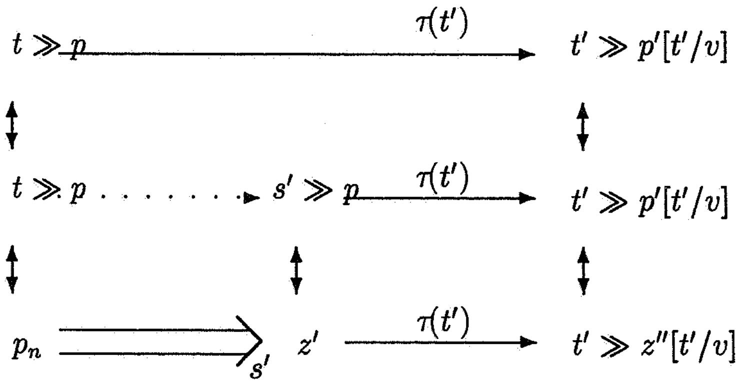
Hence there is a p_1' and a t_1' such that



Take p' such that $p \xrightarrow{\tau(r)} r \gg p'[r/v] \Longrightarrow_{t_1} p_1$ then forall $t' \in \langle t, u \rangle$ as well

$$t \gg p \xrightarrow{\tau(t')} t' \gg p'[t'/v]$$

Since $t \gg p \leftrightarrow_b q' \leftrightarrow_b p_n$ and by Fact 2 there are z', z'' and a s' such that



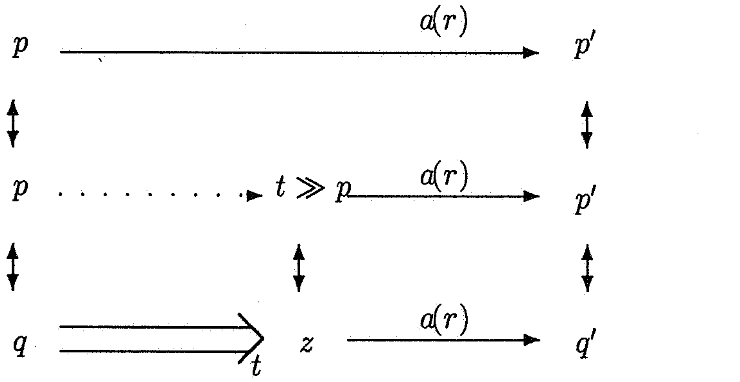
and as well $p'[t'/v] * \leftrightarrow_b z''[t'/v]*$. By induction $p'[t'/v]* \simeq z''[t'/v]*$ but

$$p \xrightarrow{\tau(r)} r \gg p'[r/v] \Longrightarrow^*_{t_n} p_n \Longrightarrow_{s'} z' \xrightarrow{\tau(t')} t' \gg z''[t'/v]$$

implies $depth(p') > depth(z'')$.

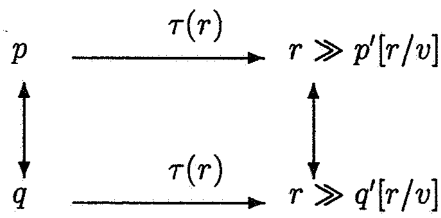
So the assumption that $n > 0$ leads to a contradiction and therefore we conclude that $n = 0$ and thus $q' \equiv q$.

Assume $U(q) = u' \geq u$ and take $s = S(p)$. Consider $p \xrightarrow{a(r)} p'$ where $a \neq \tau$. Then there are z, q' and a t such that



We have proven already that $z \equiv q$. Hence there is a q' such that $q \xrightarrow{a(r)} q'$ and $p' \leftrightarrow_b q'$, by induction $p' \leftrightarrow_s q'$.

It is left to prove that in case of $p \xrightarrow{\tau(r)} r \gg p'[r/v]$ there must be q' such that



For $r \in \langle s, u \rangle$ it is easy to see since

$$\begin{array}{ccc} p & \xrightarrow{\tau(r)} & r \gg p'[r/v] \\ \updownarrow & & \updownarrow \\ q & \dashrightarrow & r \gg q \end{array}$$

is excluded by Fact 2. So assume $r = u$, then there are two cases to consider; either $s < u$ and $\int_{v \in \{\!|s,u]} \tau(v) \cdot p' \sqsubseteq p$ or $\tau(u) \cdot p \sqsubseteq p$.

- $s < u$ and $\int_{v \in \{\!|s,u]} \tau(v) \cdot p'$ is a summand of p. We have proven already for each $t \in \langle s, u \rangle$ that there is a q^t such that

  $$\begin{array}{ccc} p & \xrightarrow{\tau(t)} & t \gg p'[t/v] \\ \updownarrow & & \updownarrow \\ q & \xrightarrow{\tau(t)} & t \gg q^t[t/v] \end{array}$$

  Since there are infinitely many t there must be a summand $\int_{v \in \{\!|s,u'\rrbracket} \tau(v) \cdot q'$ of q and an infinite subset S of $\langle s, u \rangle$ such that

  $$\forall t \in S \quad p'[t/v] \underline{\leftrightarrow}_b t \gg p'[t/v] \underline{\leftrightarrow}_b t \gg q'[t/v] \underline{\leftrightarrow}_b q'[t/v]$$

  By induction

  $$\forall t \in S \quad p'[t/v]* \simeq q'[t/v]*$$

  and since S is infinite $p' \simeq q'$. But then as well

  $$p'[u/v]* \simeq q'[u/v]*$$

  From which we obtain directly

  $$u \gg p'[u/v] \underline{\leftrightarrow}_b u \gg q'[u/v]$$

  and we are ready since by $p' \simeq q'$ as well

  $$u \gg p'[u/v] \underline{\leftrightarrow}_s u \gg q'[u/v].$$

  It is left to show that it must be the case that indeed $q \xrightarrow{\tau(u)} u \gg q'[u/v]$, thus we have to show that $u \in \{\!|s, u'\rrbracket$. In case $u' > u$ or $u' = u \wedge \rrbracket =]$ then it is obvious.

  Assume $u' = u$ and $\rrbracket = \rangle$. Then there must be a q'' such that $q \xrightarrow{\tau(u)} \gg q''[u/v]$ and $u \gg p'[u/v] \underline{\leftrightarrow}_b u \gg q''[u/v]$. Thus q has as well a summand $\int_{v \in \{\!|s,u'\rrbracket'} \tau(v) \cdot q''$ or $\tau(u') \cdot q''$. By induction $p'[u/v]* \simeq q''[u/v]*$. But then Rule 7*b* or Rule 9 can be applied on q. This is a contradiction and thus it cannot be the case that $u' = u$ and $\rrbracket = \rangle$.

- $\tau(u) \cdot p' \sqsubseteq p$. Then the transition $p \xrightarrow{\tau(u)} u \gg p'$ is matched either with a transition, thus there is a q' such that $q \xrightarrow{\tau(u)} u \gg q'[u/v]$ and $u \gg p' \underline{\leftrightarrow}_b u \gg$

$q'[u/v]$, or the transition is matched by an idling, thus $u \gg p' \leftrightarrow_b u \gg q$. We have to show that the latter is not possible.

In case of $U(q) = u$ the transition $p \xrightarrow{\tau(u)} u \gg p'$ cannot be matched by an idling at all and we are ready.

We will show that it cannot be the case that $U(q) > u$. Take $U(q) = u'$.

Assume there is a q' such that $q \xrightarrow{\tau(u)} u \gg q'[u/v]$ and $u \gg p' \leftrightarrow_b u \gg q'[u/v]$. By induction $p' \simeq q'[u/v]*$. Moreover $\int_{v \in [\![s,u' \rangle\!\rangle} \tau(v) \cdot q' \sqsubseteq q$ and by reasoning as in the previous part we obtain that $\int_{v \in [\![s,u]} \tau(v) \cdot q' \sqsubseteq p$ as well. But then $\tau(u) \cdot p'$ can be removed by Rule 9, which is a contradiction. Hence there is no such q' and thus it must be the case that $u \gg p' \leftrightarrow_b u \gg q$.

Assume q is of the form $\sum_i \int_{v \in \langle s,u' \rangle\!\rangle_i} Q + \sum_j Q'(u')$, where $u' > u$, and take $q_u \simeq \sum_i \int_{v \in \langle u,u' \rangle\!\rangle_i} Q + \sum_j Q'(u')$. And we have

$$q_u \leftrightarrow_b u \gg q \leftrightarrow_b u \gg p' \leftrightarrow_b p'$$

By induction $q_u \simeq p'$, hence $U(p') = u'$ and every summand $\int_{v \in \langle u,u' \rangle\!\rangle} P$ of p' is also a summand of q_u hence also $\int_{v \in \langle s,u' \rangle\!\rangle} P \sqsubseteq q$. Idem for summands $P(u')$ of p'.

There must be a z such that $p \simeq z + \tau(u) \cdot p'$. By reasoning as in the previous case we obtain $\int_{v \in \langle s,u \rangle\!\rangle} Q \sqsubseteq z$ for each summand $\int_{v \in \langle s,u \rangle\!\rangle} Q$ of q. In other words, for every summand $\int_{v \in \langle u,u' \rangle\!\rangle} P$ of p' there is an adjacent summand $\int_{v \in \langle s,u]} P$ in z. Moreover, each summand of z must be of the form $\int_{v \in \langle s,u]} P$ for which there is a summand $\int_{v \in \langle u,u' \rangle\!\rangle} P$ in p'. Hence $p \sim^* q$ and Rule 12 reduces $z + \tau(u) \cdot p'$ to $(z + p')_\sim$. Contradiction.

□

6.6 Rooted Branching Normal Forms and Completeness

We construct for each basic term its *rooted branching normal form*, denoted by $rbnf(p)$:

1. We replace every summand $\int_{v \in V} a(v) \cdot p'$ by $\int_{v \in V} a(v) \cdot bnf(p')$ and we lift the conditions.

2. We apply the same steps of the construction for normal forms, so we reduce the bounds of the intervals to their normal forms, we substitute redundant variables and we take double summands together.

3. The resulting term p is of the form $\sum_i \{\alpha_i :\!\rightarrow p_i\}$. If we started with a time closed term then p is a time closed term as well and there must be exactly one i such that $\alpha_i = tt$ and we deliver p_i. In case we didn't start with a time closed term then we deliver p itself.

So the Rules 10, 11 and 12 are not applied at the root level.
By construction $\mathrm{CTA}_{br} \vdash p = rbnf(p)$ and thus (by using Corollary 6.12) $\mathrm{BPA}\rho\delta\mathrm{I} \vdash p = rbnf(p)$ as well.

Lemma 6.14 $p, q \in \mathcal{T}^{cl}$

$$p \underline{\leftrightarrow}_{rb}\, q \quad \Longrightarrow \quad rbnf(p) \simeq rbnf(q)$$

Proof.

$$\begin{array}{lll} & & \mathrm{BPA}\rho\mathrm{I} + \mathrm{BI} \vdash p = rbnf(p), q = rbnf(q) \\ \text{soundness} & \Longrightarrow & p \underline{\leftrightarrow}_{rb}\, rbnf(p), q \underline{\leftrightarrow}_{rb}\, rbnf(q) \\ \text{transitivity of } \underline{\leftrightarrow}_b & \Longrightarrow & rbnf(p) \underline{\leftrightarrow}_{rb}\, rbnf(q) \\ \text{Unicity Theorem} & \Longrightarrow & rbnf(p) \simeq rbnf(q) \end{array}$$

□

Corollary 6.15 $p, q \in \mathcal{T}^{cl}$ *(Completeness)*

$$p \underline{\leftrightarrow}_{rb}\, q \quad \Longrightarrow \quad \mathrm{BPA}\rho\mathrm{I} + \mathrm{BI} \vdash p = q$$

Proof. Direct by the previous Lemma.

7 An Operational Semantics with Idle Transitions

In this section we give an operational semantics in the style of Baeten & Bergstra in Table 8. Each state is a pair of a time closed term and a point in time. The state consisting of the term p at time t is denoted by $< p, t >$. An idle transition is a transition which increases the course of time without changing the process part, in this paper we label an idle transition with the greek letter iota (ι). Thus if the time of the state $< p, t >$ can be increased to r (since $U(p) > r$) then we have $< p, t > \xrightarrow{\iota(r)} < p, r >$. Baeten & Bergstra do not have labels on idle transitions. We refer to this operational semantics by *idle semantics.*

On the transition systems we define again *strong bisimulation* and *(rooted) branching bisimulation.* To distinguish these definitions of earlier definitions we add an index ι which stands for *idle.* We denote these new equivalences by resp. $\underline{\leftrightarrow}_s^\iota$ and $\underline{\leftrightarrow}_{(r)b}^\iota$ and we will prove that they coincide with the earlier definitions.

7.1 The Action Rules and Strong Bisimulation

The root state of the transition system of a process p is the state $< p, 0 >$, denoting that time starts at zero. If from $< p, t >$ an initial action at time r can be executed, then there is an idle transition to every point between t and r. Note that $< p, t > \xrightarrow{a(r)} < \surd, r >$ implies that $a \neq \iota$. We can define strong bisimulation and strong bisimulation equivalence in the standard way.

Definition 7.1 *Strong Bisimulation*
$\mathcal{R} \subset (\mathcal{T} \times [0, \infty\rangle)^2$ *is a* strong bisimulation *if whenever* $< p, t > \mathcal{R} < q, t >$ *then*

1. $< p, t > \xrightarrow{a(r)} < p', r >$ *implies that there is a* q' *such that* $< q, t > \xrightarrow{a(r)} < q', r >$ *and* $< p', r > \mathcal{R} < q', r >$.
2. $< p, t > \xrightarrow{a(r)} \surd$ *implies that* $< q, t > \xrightarrow{a(r)} \surd$.
3. *Respectively (1) and (2) with the role of p and q interchanged.*

Definition 7.2 *Strong Bisimulation Equivalence*
$< p, t > \underline{\leftrightarrow}_s^\iota < q, t >$ *iff there is a strong bisimulation* $\mathcal{R}$ *such that* $< p, t > \mathcal{R} < q, t >$.

In the sequel we will show that this equivalence corresponds with the strong equivalence we have defined before; we will show that $p \underline{\leftrightarrow}_s q$ iff $< p, 0 > \underline{\leftrightarrow}_s^\iota < q, 0 >$.

7.2 The correspondence between the two operational semantics

In this section we will discuss the correspondence between the idle semantics and abstract semantics. It is mainly taken from [Klu91a]. Consider

In abstract semantics	$(a(r) \cdot p) \cdot q$	$\xrightarrow{a(r)}$	$(r \gg p) \cdot q$
In idle semantics	$< (a(r) \cdot p) \cdot q, 0 >$	$\xrightarrow{a(r)}$	$< p \cdot q, r >$

$$\frac{t < r}{< b(r), t > \xrightarrow{b(r)} \surd} \qquad \frac{t < r < s}{< c(s), t > \xrightarrow{\iota(r)} < c(s), r >}$$

$$\frac{t < r \quad r \in V}{< \int_{v \in V} b(v), t > \xrightarrow{b(r)} \surd} \qquad \frac{t < r < sup(V)}{< \int_{v \in V} c(v), t > \xrightarrow{\iota(r)} < \int_{v \in V} c(v), r >}$$

$$\frac{< p, t > \xrightarrow{b(r)} < p', r >}{< p + q, t > \xrightarrow{b(r)} < p', r >} \qquad \frac{< p, t > \xrightarrow{b(r)} \surd}{< p + q, t > \xrightarrow{b(r)} \surd}$$

$$\frac{< p, t > \xrightarrow{b(r)} < p', r >}{< q + p, t > \xrightarrow{b(r)} < p', r >} \qquad \frac{< p, t > \xrightarrow{b(r)} \surd}{< q + p, t > \xrightarrow{b(r)} \surd}$$

$$\frac{< p, t > \xrightarrow{\iota(r)} < p, r >}{< p + q, t > \xrightarrow{\iota(r)} < p + q, r >} \qquad \frac{< p, t > \xrightarrow{\iota(r)} < p, r >}{< q + p, t > \xrightarrow{\iota(r)} < q + p, r >}$$

$$\frac{< p, t > \xrightarrow{a(r)} < p', r >}{< p \cdot q, t > \xrightarrow{a(r)} < p' \cdot q, r >} \qquad \frac{< p, t > \xrightarrow{a(r)} \surd}{< p \cdot q, t > \xrightarrow{a(r)} < q, r >}$$

$$\frac{r > s \quad < p, t > \xrightarrow{a(r)} < p', r >}{< s \gg p, t > \xrightarrow{a(r)} < p', r >} \qquad \frac{r > s \quad < p, t > \xrightarrow{a(r)} \surd}{< s \gg p, t > \xrightarrow{a(r)} \surd}$$

$$\frac{r < s \quad < p, t > \xrightarrow{b(r)} < p', r >}{< p \gg s, t > \xrightarrow{b(r)} < p', r >} \qquad \frac{r < s \quad < p, t > \xrightarrow{a(r)} \surd}{< p \gg s, t > \xrightarrow{a(r)} \surd}$$

$$\frac{< p, t > \xrightarrow{b(s)} < p', s >}{< p@s, t > \xrightarrow{b(s)} < p', s >} \qquad \frac{< p, t > \xrightarrow{a(s)} \surd}{< p@s, t > \xrightarrow{a(s)} \surd}$$

$$\frac{r < s \quad < p, t > \xrightarrow{\iota(r)} < p, r >}{< s \gg p, t > \xrightarrow{\iota(r)} < s \gg p, r >} \qquad \frac{r < s \quad < p, t > \xrightarrow{\iota(r)} < p, r >}{< p \gg s, t > \xrightarrow{\iota(r)} < p \gg s, r >}$$

$$\frac{t < r < s}{< p@s, t > \xrightarrow{\iota(r)} < p@s, r >}$$

Table 8: Action Rules ($a \in A_{\tau,\iota}, b \in A_\tau, c \in A_{\tau,\delta}$ and $r, s, t \in [0, \infty]$)

We see that in the abstract semantics the course of time is encoded by an occurrence of $r \gg$ in the prefix, while in the idle semantics the course of time is reflected by the time component in the state.

In order to relate the states of the (abstract semantics) transition system of p with the states of the (idle semantics) transition $< p, 0 >$, we define the function *strip* inductively. It is the counterpart of the function *time* which is defined earlier.

$$\begin{array}{lcl}
strip(\int_{v \in V} a(v)) & = & \int_{v \in V} a(v) \\
strip(\int_{v \in V} a(v) \cdot p) & = & \int_{v \in V} a(v) \cdot p \\
strip(X + Y) & = & X + Y \\
strip(X \gg r) & = & X \gg r \\
strip(X @ r) & = & X @ r \\
strip(r \gg X) & = & X \\
strip(X \cdot Y) & = & strip(X) \cdot Y
\end{array}$$

Using the functions *time* (Definition 2.4) and *strip* we can state some simple Propositions concerning the (abstract) transition system :

Proposition 7.3 $p, p' \in \mathcal{T}$ $a \in A$ $r \in [0, \infty]$

$$p \xrightarrow{a(r)} p' \quad \Longleftrightarrow \quad r > time(p) \wedge strip(p) \xrightarrow{a(r)} p'$$

The following proposition states a correspondence between the transition system of p in the abstract operational semantics and the transition system of $< strip(p), time(p) >$ in the idle semantics:

Proposition 7.4 $\forall p, q \in \mathcal{T}$ $\forall a \in A_\tau$ $\forall r \in [0, \infty]$

$$\begin{array}{lcl}
p \xrightarrow{a(r)} \surd & \Longleftrightarrow & < strip(p), time(p) > \xrightarrow{a(r)} < \surd, r > \\
p \xrightarrow{a(r)} q & \Longrightarrow & < strip(p), time(p) > \xrightarrow{a(r)} < strip(q), r > \\
< strip(p), time(p) > \xrightarrow{a(r)} < q, r > & \Longrightarrow & \\
 & & \exists q' \quad strip(q') \equiv q \wedge time(q') = r \wedge p \xrightarrow{a(r)} q'
\end{array}$$

Furthermore, w.r.t. the original semantics we know that

$$U(p) = \sup\{s | < p, t > \xrightarrow{\iota(s)} < p, s >\}$$

and

$$U(p) > L(p) \quad \Longrightarrow \quad \forall a \in A_{\tau,\iota} \quad < p, t > \stackrel{a(U(p))}{\not\longrightarrow}$$

Now we are ready to prove the following:

Lemma 7.5 $p, q \in \mathcal{T}_I^{cl}$

$$p \leftrightarrow_s q \quad \Longleftrightarrow \quad < p, 0 > \leftrightarrow_s < q, 0 >$$

Proof.

- $\Longrightarrow$ Assume $\mathcal{R} \subseteq \mathcal{T}_I^{cl} \times \mathcal{T}_I^{cl}$ is a strong bisimulation relation relating p and q. Note that for all $p', q' \in \mathcal{T}_I^{cl}$ $p'\mathcal{R}q'$ implies $U(p') = U(q')$.
Construct $\mathcal{R}^* \subseteq (\mathcal{T}_I^{cl} \times [0,\infty\rangle) \times (\mathcal{T}_I^{cl} \times [0,\infty\rangle)$:

$$\begin{array}{rcl} \mathcal{R}^* & = & \{ \ (< strip(p'), r >, < strip(q'), r >) \mid (p', q') \in \mathcal{R} \ \wedge \ time(p') \leq r < U(p') \ \} \\ & \cup & \{ \ (< p, r >, < q, r >) \mid 0 \leq r < U(p) \ \} \end{array}$$

The proof that $\mathcal{R}^*$ is a strong bisimulation relation relating $< p, 0 >$ and $< q, 0 >$ is left to the reader.

- $\Longleftarrow$ Assume $\mathcal{R}^* \subseteq (\mathcal{T}_I^{cl} \times [0,\infty\rangle) \times (\mathcal{T}_I^{cl} \times [0,\infty\rangle)$ is a strong bisimulation relation containing $(< p, 0 >, < q, 0 >)$, then construct $\mathcal{R} \subseteq \mathcal{T}_I^{cl} \times \mathcal{T}_I^{cl}$:

$$\begin{array}{rcl} \mathcal{R} & = & \{ \ (p', q') \mid (< strip(p'), time(p') >, < strip(q'), time(q') >) \in \mathcal{R}^* \ \} \\ & \cup & \{ \ (p, q) \ \} \end{array}$$

Again, the proof that $\mathcal{R}$ is a strong bisimulation relation relating p and q is left to the reader. □

7.3 Branching Bisimulation

Before giving the definition of branching bisimulation for these transition systems we have to define $< p, t > \Longrightarrow_r < q, r >$:

Definition 7.6 $\Longrightarrow \subset ((\mathcal{T} \times [0,\infty\rangle) \times [0,\infty\rangle \times (\mathcal{T} \times [0,\infty\rangle))$ *is defined as the least relation satisfying:*

- $< p, t > \Longrightarrow_t < p, t >$
- *if* $< p, t > \Longrightarrow_r < q, r >$ *and* $< q, r > \xrightarrow{\tau(r')} < q', r' >$ *then* $< p, t > \Longrightarrow_{r'} < q', r' >$

It would have been possible to allow idle transitions as well in a series of steps $< p, t > \Longrightarrow_r < p', t >$. However, since

$$< p_0, t_0 > \xrightarrow{\iota_1} < p_1, t_1 > \ldots \xrightarrow{\iota_n} < p_n, t_n > \xrightarrow{\tau(r)} < p', r > \quad \Longrightarrow \quad < p_0, t_0 > \xrightarrow{\tau(r)} < p', r >$$

we prefer the above definition. $< p, t > \Longrightarrow_r^* < p', r >$ denotes that $< p, t > \Longrightarrow_r < p', r >$ or that there is a s such that $< p, t > \Longrightarrow_s < p', s > \xrightarrow{\iota(r)} < p', r >$.

Within abstract semantics $p\mathcal{R}q$ and $p \xrightarrow{\tau(r)}$ might imply that $U(q) > r$ and $p'\mathcal{R}(r \gg q)$. In the idle semantics of this section of this section this situation is formulated by saying that a τ transition may be matched with a ι transition; i.e. if

$$< p, t > \mathcal{R} < q, r > \text{ and } < p, t > \xrightarrow{\tau(r)} < p', r >$$

then it might be the case that

$$< p', r > \mathcal{R} < q, r > \text{ and } < q, t > \xrightarrow{\iota(r)} < q, r > .$$

We obtain the following definitions of idle branching bisimulation and idle branching bisimulation equivalence.

Definition 7.7 *Idle Branching Bisimulation*
$\mathcal{R} \subset (\mathcal{T} \times [0,\infty))^2$ *is an* idle branching bisimulation *if whenever* $<p,t> \mathcal{R} <q,t>$ *then*

1. $<p,t> \xrightarrow{a(r)} <p',r>$ *($a \in A_{\tau,\iota}$) implies that there are z, q', a' and s such that*
   - $<q,t> \Longrightarrow_s <z,s> \xrightarrow{a'(r)} <q',r>$,
   - *if $a \in A$ then $a' = a$ else $a' \in \{\tau, \iota\}$,*
   - $<p,s> \mathcal{R} <z,s>$ *and* $<p',r> \mathcal{R} <q',r>$.
2. $<p,t> \xrightarrow{a(r)} \surd$ *($a \in A_\tau$) implies that there are z and s such that*
   - $<q,t> \Longrightarrow_s <z,s> \xrightarrow{a(r)} \surd$,
   - $<p,s> \mathcal{R} <z,s>$.
3. *Respectively (1) and (2) with the role of p and q interchanged.*

Note that $<p,t> \xrightarrow{a(r)} \surd$ implies that $a \in A_\tau$.

Definition 7.8 *Idle Branching Bisimulation Equivalence*
$<p,t> \underline{\leftrightarrow}_b^\iota <q,t>$ *iff there is a idle branching bisimulation $\mathcal{R}$ such that* $<p,t> \mathcal{R} <q,t>$.

As usual $\underline{\leftrightarrow}_b^\iota$ itself is not a congruence, therefore we have *rooted* idle branching bisimulation equivalence.

Definition 7.9 *Rooted Idle Branching Bisimulation Equivalence*
$<p,t> \underline{\leftrightarrow}_{rb}^\iota <q,t>$ *iff*

1. $<p,t> \xrightarrow{a(r)} <p',r>$ *with $a \in A_\tau$ implies that there is a q' such that* $<q,t> \xrightarrow{a(r)} <q',r>$ *and* $<p',r> \underline{\leftrightarrow}_b^\iota <q',r>$.
2. $<p,t> \xrightarrow{\iota(r)} <p,r>$ *implies that* $<q,t> \xrightarrow{\iota(r)} <q,r>$ *and* $<p,r> \underline{\leftrightarrow}_{rb}^\iota <q,r>$.
3. $<p,t> \xrightarrow{a(r)} \surd$ *implies that* $<q,t> \xrightarrow{a(r)} \surd$.
4. *Respectively (1),(2) and (3) with the role of p and q interchanged.*

Note that the equivalence does not change by omitting the second clause.

Lemma 7.10 $\underline{\leftrightarrow}_{rb}^\iota$ *is a congruence over BPAρδI*

Proof. Omitted

Lemma 7.11

$$\mathrm{BPA}\rho\delta\mathrm{I} \vdash p = q \quad \Longrightarrow \quad <p,0> \underline{\leftrightarrow}_{rb}^\iota <q,0>$$

Proof. Omitted

For proving the correspondence between $\underline{\leftrightarrow}_b^\iota$ and $\underline{\leftrightarrow}_b$ we have to introduce *branching bisimulation inclusion*, a kind of semantical summand inclusion, and *primed idle branching bisimulation equivalence.*

Definition 7.12 *Branching Bisimulation Inclusion*
$<p,t> \subseteq_{br} <q,t>$ *iff*

1. $<p,t> \xrightarrow{a(r)} <p',r>$ *implies that there are* z, q', a' *and* s *such that*
   - $<q,t> \Longrightarrow_s <z,s> \xrightarrow{a'(r)} <q',r>$,
   - *if* $a \in A$ *then* $a' = a$ *else* $a' \in \{\tau, \iota\}$,
   - $<p,s> \subseteq_{br} <z,s>$ *and* $<p',r> \leftrightarrow_b^{\iota} <q',r>$.
2. $<p,t> \xrightarrow{a(r)} \surd$ *implies that there are* z *and* s *such that*
   - $<q,t> \Longrightarrow_s <z,s> \xrightarrow{a(r)} \surd$ *and*
   - $<p,s> \subseteq_{br} <z,s>$.

Proposition 7.13

$$<p,t> \subseteq_{br} <q,t> \ \wedge \ <q,t> \subseteq_{br} <p,t> \iff <p,t> \leftrightarrow_b^{\iota} <q,t>$$
$$<p,t> \subseteq_{br} <p',t> \subseteq_{br} <p'',t> \implies <p,t> \subseteq_{br} <p'',t>$$

Proof. Direct by definition. □

Proposition 7.14 *Let* p *be a branching normal form and* $r < U(p)$.

$$<p,t> \xrightarrow{\tau(r)} <p',r> \implies <p',r> \subseteq_{br} <p,r>$$

Proof. Omitted

Proposition 7.15 *Let* p, q *be branching normal forms and* $t < r < \min(U(p), U(q))$.

$$<p,t> \leftrightarrow_b^{\iota} <q,t> \implies <p,r> \leftrightarrow_b^{\iota} <q,r>$$

Proof. $<p,t> \xrightarrow{\iota(r)} <p,r>$, and by the definition of $\leftrightarrow_b^{\iota}$ there is a $<q',r>$ such that $<q,t> \Longrightarrow_r^* <q',r>$ and $<p,r> \leftrightarrow_b^{\iota} <q',r>$. By the previous Proposition $<q',r> \subseteq_{br} <q,r>$, hence $<p,r> \subseteq_{br} <q,r>$.

Similarly we can deduce that $<q,r> \subseteq_{br} <p,r>$ and thus $<p,r> \leftrightarrow_b^{\iota} <q,r>$. □

Definition 7.16 *Primed Idle Branching Bisimulation*
$\mathcal{R} \subset (\mathcal{T} \times [0,\infty))^2$ *is a* primed idle branching bisimulation *if whenever* $<p,t> \mathcal{R} <q,t>$ *then*

1. $<p,t> \xrightarrow{a(r)} <p',r>$ *(*$a \in A_\tau$*) implies*
   - *either* $a = \tau$, $\exists q'$ *and* $\exists s$ *such that* $<q,t> \Longrightarrow_s <q',s> \xrightarrow{\iota(r)} <q',r>$ *and* $<p,s> \mathcal{R} <q',s>$, $<p',r> \mathcal{R} <q',r>$. *Moreover,* $<p,t> \xrightarrow{\iota(r)} <p,r>$ *implies* $<p',r> \mathcal{R} <p,r>$.

- *or* $\exists z, q'$ *and* $\exists s$ *such that* $< q,t > \Longrightarrow_s < z,s > \xrightarrow{a(r)} < q',r >$ *and* $< p,s > \mathcal{R} < z,s >$ *and* $< p',r > \mathcal{R} < q',r >$.

2. $< p,t > \xrightarrow{\iota(r)} \surd$ *implies* $\exists z$ *and* $\exists s$ *such that* $< q,t > \Longrightarrow_s < z,s > \xrightarrow{a(r)} \surd$ *and* $< p,s > \mathcal{R} < z,s >$.

3. *Respectively (1) and (2) with the role of p and q interchanged.*

We define *primed idle branching bis. eq.* ($\underline{\leftrightarrow}_b^{\iota'}$) and *rooted idle branching bis. eq.* ($\underline{\leftrightarrow}_{rb}^{\iota'}$).

Lemma 7.17 $p, q \in \mathcal{T}_I^{cl}$

$$p \underline{\leftrightarrow}_{(r)b}\, q \iff < p,0 > \underline{\leftrightarrow}_{(r)b}^{\iota'} < q,0 >$$

Proof. Similar to the proof of Lemma 7.5. □

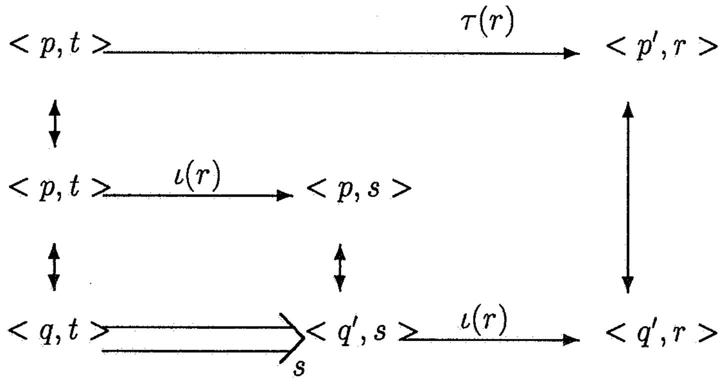
Lemma 7.18 $p, q \in \mathcal{T}_I^{cl}$

$$< p,r > \underline{\leftrightarrow}_b^{\iota'} < q,r > \iff < p,r > \underline{\leftrightarrow}_b^{\iota} < q,r >$$

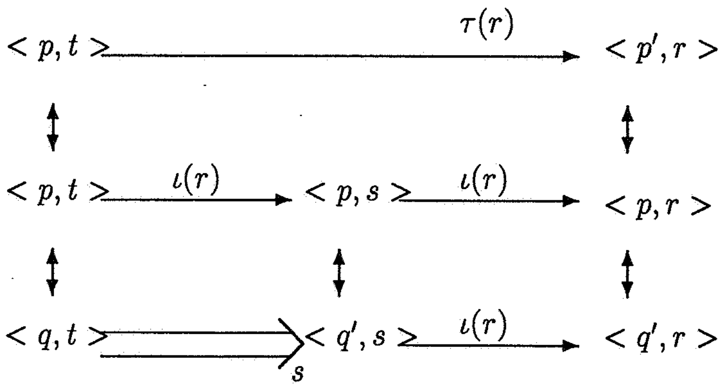
Proof.

- $\Longrightarrow$ Assume $< p,r > \underline{\leftrightarrow}_b^{\iota'} < q,r >$, take an arbitrary $a \in A_\tau$ then $< a(r) \cdot p, 0 > \underline{\leftrightarrow}_{rb}^{\iota'} < a(r) \cdot q, 0 >$. By the previous Lemma $a(r) \cdot p \underline{\leftrightarrow}_{rb}\, a(r) \cdot q$ and by completeness BPA$\rho\delta$I +BI $\vdash p = q$. By soundness of BPA$\rho\delta$I +BI w.r.t. $\underline{\leftrightarrow}_{rb}^{\iota}$ we have $< a(r) \cdot p, 0 > \underline{\leftrightarrow}_{rb}^{\iota} < a(r) \cdot q, 0 >$. Hence $< p,r > \underline{\leftrightarrow}_b^{\iota} < q,r >$.

- $\Longleftarrow$ It is sufficient to show that for p, q being branching normal forms



implies



From Proposition 7.15 and $< p, s > \underline{\leftrightarrow}_b^\iota < q', s >$ we obtain $< p, r > \underline{\leftrightarrow}_b^\iota < q', r >$ as was to be shown.

□

Finally we can state that branching bisimulation as defined in abstract semantics coincides with branching bisimulation defined in idle semantics.

Corollary 7.19 $p, q \in \mathcal{T}^{cl}$

$$p \underline{\leftrightarrow}_{(r)b}\, q \quad \Longleftrightarrow \quad < p, 0 > \underline{\leftrightarrow}_{(r)b}^\iota < q, 0 >$$

8 Delay Bisimulation and Time

8.1 Rooted Delay Bisimulation Equivalence

Now we have studied branching bisimulation in detail it is easy to introduce delay bisimulation. In the untimed case, delay bisimulation can be found by taking branching bisimulation and relaxing one condition (see Figure 1). Let us take the definition of branching bisimulation and do likewise.

Definition 8.1 *Delay Bisimulation(Not correct!)*
$\mathcal{R} \subset \mathcal{T}_I^{cl} \times \mathcal{T}_I^{cl}$ *is a* delay bisimulation *if whenever* $p\mathcal{R}q$ *then*

1. $p \xrightarrow{a(r)} p'$ *implies*
   - *either* $a = \tau$, $\exists z$ *with* $U(q') > r$ *and* $\exists s$ *such that* $q \Longrightarrow_s q'$ *and and* $p'\mathcal{R}(r \gg q')$
   - *or* $\exists z, q'$ *and* $\exists t$ *such that* $q \Longrightarrow_t z \xrightarrow{a(r)} q'$ *and* $p'\mathcal{R}q'$.
2. $p \xrightarrow{a(r)} \surd$ *implies* $\exists z$ *and* $\exists t$ *such that* $q \Longrightarrow_t z \xrightarrow{a(r)} \surd$.
3. *Respectively (1) and (2) with the role of p and q interchanged.*

with this definition we can find a delay bisimulation $\mathcal{R}$ which relates

$$\int_{v\in\langle 1,2\rangle} \tau(v)\cdot a(3) + b(3) \quad \text{and} \quad \int_{v\in\langle 1,3\rangle} \tau(v)\cdot a(3) + b(3)$$

We do not want to relate these two terms; in the term on the left hand side the choice is determined before time 2, while in the term on the right hand side the choice still can be made after 2.

The conclusion is that when dealing with delay bisimulation we have to consider the idle transitions as well. Therefore we will not discuss the above definition of a delay bisimulation in the abstract semantics any more.

What we have to do now is to take the definition of idle branching bisimulation and turn that one into a definition for delay bisimulation.

Definition 8.2 *Delay Bisimulation*
$\mathcal{R} \subset (\mathcal{T} \times [0,\infty\rangle)^2$ *is a* delay bisimulation *if* $< p,t > \mathcal{R} < q,t >$ *and*

1. $< p,t > \xrightarrow{a(r)} < p',r >$ *implies that there are* z, q', a' *and* s *such that*
   - $< q,t > \Longrightarrow_s < z,s > \xrightarrow{a'(r)} < q',r >$,
   - *if* $a \in A$ *then* $a' = a$ *else* $a' \in \{\tau, \iota\}$,
   - $< p',r > \mathcal{R} < q',r >$.
2. $< p,t > \xrightarrow{a(r)} \surd$ *implies that there are* z *and* s *such that*
   - $< q,t > \Longrightarrow_s < z,s > \xrightarrow{a(r)} \surd$,

3. Respectively (1) and (2) with the role of p and q interchanged.

We define *delay bisimulation equivalence* (denoted by $\underline{\leftrightarrow}_d$) and *rooted bisimulation equivalence* (denoted by $\underline{\leftrightarrow}_{rd}$) on $(\mathcal{T}_I^{cl} \times [0,\infty])^2$ in the standard way. Finally we define $\underline{\leftrightarrow}_d$ on $\mathcal{T}_I^{cl} \times \mathcal{T}_I^{cl}$ by putting

$$p \underline{\leftrightarrow}_d q \quad \Longleftrightarrow \quad <p,0> \underline{\leftrightarrow}_d <q,0>$$

Similarly we define $\underline{\leftrightarrow}_{rd}$ on $\mathcal{T}_I^{cl} \times \mathcal{T}_I^{cl}$.

Proposition 8.3 $\underline{\leftrightarrow}_{rd}$ *is a congruence over* $\mathcal{T}_I^{cl}$

The proof of this proposition is left to the reader.

8.2 The Timed Delay Law

We obtain the following new identity:

$$\begin{array}{l} a(1) \cdot (\int_{v\in\langle 1,10\rangle} \tau(v) \cdot \int_{w\in\langle 1,10\rangle} b(w) + c(11)) \quad \underline{\leftrightarrow}_{rd} \\ a(1) \cdot (\int_{v\in\langle 1,10\rangle} \tau(v) \cdot \int_{w\in\langle 1,10\rangle} b(w) + \int_{w\in\langle 1,10\rangle} b(w) + c(11)) \end{array}$$

This identity looks like an instantiation of the (untimed) delay law

$$\text{T2} \qquad \tau \cdot X + X = \tau \cdot X.$$

We can apply this law only within an appropriate context, since the root condition in the timed case corresponds with the strongly rootedness condition of the untimed case (Definition 1.10). The delay law becomes in real time

ATTI2	$v \in V \Longrightarrow U(X) \leq \sup(W)$
	$\int_{v\in V} a(v) \cdot ((\int_{w\in W} \tau(w)) \cdot X + Y) = \int_{v\in V} a(v) \cdot ((\int_{w\in W} \tau(w)) \cdot X + \inf(W) \gg X + Y)$

Note the condition $U(X) \leq \sup(W)$ since:

$$\begin{array}{l} a(1) \cdot (\int_{v\in\langle 1,10\rangle} \tau(v) \cdot \int_{w\in\langle 1,11\rangle} b(w) + c(11)) \quad \not\underline{\leftrightarrow}_{rd} \\ a(1) \cdot (\int_{v\in\langle 1,10\rangle} \tau(v) \cdot \int_{w\in\langle 1,11\rangle} b(w) + \int_{w\in\langle 1,11\rangle} b(w) + c(11)) \end{array}$$

And we have the following Theorem.

Theorem 8.4 *(Soundness)*$p, q \in \mathcal{T}_I^{cl}$

$$\text{BPA}\rho\text{I}\delta + \text{ATBI} + \text{ATTI2} \vdash p = q \quad \Longrightarrow \quad p \underline{\leftrightarrow}_{rd} q$$

Proof. Omitted

In delay bisimulation it is more difficult to reason with τ-transitions in which variables may become bound, since the intermediate states do not have to be related. We have the completeness only for a restricted set of terms. The techniques we have found so far have not been sufficient to deal with the terms in which τ transitions may bind variables non trivially. Before studying proof techniques to deal with these terms in delay bisimulation it has to be determined whether delay bisimulation as defined here is a worthwile notion.

The set $\mathcal{T}_I^{res}$ is defined as follows

$$\mathcal{T}_I^{res} = \{p \in \mathcal{T}_I^{cl} | \int_{v \in V} \tau(v) \cdot p' \text{ is a subterm of } p \text{ mplies } v \notin FV^*(p')\}$$

We then have a completeness theorem.

Theorem 8.5 *(Completeness)*$p, q \in \mathcal{T}_I^{res}$

$$p \underline{\leftrightarrow}_{rd} q \quad \Longrightarrow \quad \text{BPA}\rho\text{I}\delta + \text{ATBI} + \text{ATTI2} \vdash p = q$$

Proof. We give only a sketch here. First we formulate ATTI2 as conditional law TC2 and we prove that for $p, q \in \mathcal{T}^{cl}$ CTA_{br} + TC2$\vdash p = q$ implies BPA$\rho\delta$I +ATBI + ATTI2 $\vdash p = q$ for arbitrary σ.

We may assume that p is a basic term. We construct its *(rooted) delay normal form* by taking the construction of branching normal forms and adding the following Rule to it. This new Rule must be applied right after the step in which smaller terms have been replaced by their delay normal form. Only at the root level of p this Rule is not applied.

$$\begin{array}{lll} \int_{w \in W} \tau(w) \cdot p + q & \longrightarrow & \\ & & \\ \{v \in V \Longrightarrow U(p) = \sup(W) & :\longrightarrow & \int_{w \in W} \tau(w) \cdot p + p[\inf(W)/w] + q\} \\ + & & \\ \{o.w & :\longrightarrow & \int_{w \in W} \tau(w) \cdot p + q\} \end{array}$$

Then we show that if p, q are in rooted delay normal form

$$< p, 0 > \underline{\leftrightarrow}_{rd} < q, 0 > \quad \Longrightarrow \quad < p, 0 > \underline{\leftrightarrow}_{rb}^{\iota} < q, 0 >$$

And we are ready, since we can apply the completeness Theorem for rooted branching bisimulation. □

Next, we have the following proposition which explains why we have not introduced delay bisimulation without integration. This fact may look surprising, but it can easily be seen from the fact that the axiom ATTI2, which characterizes those identities which are not branching bisimilar, is derivable within $\mathcal{T}_\rho$. Note that if $W = [t, t]$ then $U(X) \leq \sup(W)$ implies $\inf(W) \gg X = \delta(t)$.

Proposition 8.6 $p, q \in \mathcal{T}_\rho$

$$p \underline{\leftrightarrow}_{rd} q \quad \Longleftrightarrow \quad p \underline{\leftrightarrow}_{rb} q$$

8.3 The embedding of untimed into timed rooted delay bisimulation

Finally, we note that untimed delay bisimulation can not be embedded straightforwardly into timed delay bisimulation. If we consider the translation RT_τ of section 4.10 then we do lose some identities, for example $RT_\tau(\tau \cdot a) \not\underline{\leftrightarrow}_{rb} RT_\tau(\tau \cdot a + a)$. This is due to our definition of rooted delay branching bisimulation, which is more similar to the notion of strongly rootedness as defined in Definition 1.10. This is reflected by the following Thereom.

Theorem 8.7 $p, q \in \mathcal{T}_u$

$$RT_\tau(p) \mathrel{\underline{\leftrightarrow}}_{rd} RT_\tau(q) \quad \Longleftrightarrow \quad p \mathrel{\underline{\leftrightarrow}}_{srd} q$$

Baeten & Bergstra have suggested to interpret τ-transitions as idle transitions. In order to obtain "well behaved" transition systems one has to apply the transitive closure on idle and step transitions. This idea is investigated further in [Klu91b], this transitive closure is expressed by certain Action Rules which have to be applied on internal states only in order to obtain a congruence. Internal states are distinguished from root states by keeping a boolean value in each state. It can be proven that strong bisimulation equivalence over these transition systems coincides with rooted delay bisimulation equivalence of this paper.

9 Weak Bisimulation and Time

9.1 Its Definition and associated Law

As with delay bisimulation equivalence it doesn't make sense to define it within abstract semantics. Therefore we take the definition of delay bisimulation in the context of idle semantics and turn it into one for weak bisimulation.

Definition 9.1 *Weak Bisimulation*
$\mathcal{R} \subset (\mathcal{T} \times [0,\infty))^2$ *is a* weak bisimulation *if whenever* $<p,t> \mathcal{R} <q,t>$ *then*

1. $<p,t> \xrightarrow{a(r)} <p',r>$ *implies that there are* $z, z', q' \in \mathcal{T}_I^{cl}$, $a' \in A_{\tau,\iota}$ *and* $s, t' \in [0,\infty)$ *such that*
   - $<q,t> \Longrightarrow_s <z,s> \xrightarrow{a'(r)} <z',r> \Longrightarrow_{t'} <q',t'>$,
   - *if* $a \in A$ *then* $a' = a$ *else* $a' \in \{\tau, \iota\}$,
   - *if* $t' > r$ *then* $<p',r> \xrightarrow{\iota(t')} <p',t'>$.
   - $<p',t'> \mathcal{R} <q',t'>$
   - $<p',r> \xrightarrow{a''(t'')} <p'',t''>$ *and* $a'' \in A_\tau$ *implies* $t'' > t'$
2. $<p,t> \xrightarrow{a(r)} \surd$ *implies that there are* $z \in \mathcal{T}_I^{cl}$ *and* $s \in [0,\infty)$ *such that* $<q,t> \Longrightarrow_s <z,s> \xrightarrow{a(r)} \surd$
3. *Respectively (1) and (2) with the role of p and q interchanged.*

We define *weak bisimulation equivalence* (denoted by $\underline{\leftrightarrow}_w$) and *rooted weak bisimulation equivalence* (denoted by $\underline{\leftrightarrow}_{rw}$).

Example 9.2

$$a(1) \cdot (\tau(2) \cdot b(3) + c(2)) \quad \underline{\leftrightarrow}_{rw} \quad a(1) \cdot (\tau(2) \cdot b(3) + c(2)) + a(1) \cdot b(3)$$

This identity looks like an instance of the untimed law T3 $a{\cdot}(\tau{\cdot}X+Y) = a{\cdot}(\tau{\cdot}X+Y)+a{\cdot}X$. And we have the following timed version of T3.

ATTI3 $v \in V \Longrightarrow v < \sup(W) \wedge W < U(X)$
$\int_{v\in V} a(v) \cdot (\int_{w\in W} \tau(w) \cdot X + Y) = \int_{v\in V} a(v) \cdot (\int_{w\in W} \tau(w) \cdot X + Y) + \int_{v\in V} a(v) \cdot (\int_{w\in W} \tau(w) \cdot X)$

We obtain new identities over the calculus without τ as well, these are characterized by combining ATB and ATTI3.

$r < t < U(X) \wedge U(Y) \leq t$
$a(r) \cdot (t \gg X + Y) = a(r) \cdot (t \gg X + Y) + a(r) \cdot (t \gg X)$

Theorem 9.3 *(Soundness)* $p, q \in \mathcal{T}_I^{cl}$

$$\text{BPA}\rho\delta\tau + \text{ATBI} + \text{ATTI2} + \text{ATTI3} \vdash p = q \quad \Longrightarrow \quad p \leftrightarrows_{rw} q$$

Proof. Omitted

Theorem 9.4 *(Completeness)* $p, q \in \mathcal{T}_I^{res}$

$$p \leftrightarrows_{rw} q \quad \Longrightarrow \quad \text{BPA}\rho\delta\tau + \text{ATBI} + \text{ATTI2} + \text{ATTI3} \vdash p = q$$

Proof. Omitted

9.2 Weak Bisimulation is not a congruence over ACPρ

Let us extend BPA$\rho\delta$ to ACPρ by adding communication and encapsulation ([BB91] ,[Klu91a]). We assume $c|\bar{c} = \tau$, then $c(2)|\bar{c}(2) = \tau(2)$. Furthermore we disallow the individual action $c(2)$ to happen by the encapsulation $\partial_{\{c\}}$.

Take $p = a(1) \cdot (b(3) + c(2))$ and $q = a(1) \cdot (b(3) + c(2)) + a(1) \cdot b(3)$. Then we have BPA$\rho\delta$+ ATT4 $\vdash p = q$. But in a context they can be distinguished. In $\partial_{\{c\}}(p \| \bar{c}(2))$ at time 2 a communication of $c(2)$ with $\bar{c}(2)$ is forced since it is the only option for the whole process not to deadlock at 2. However, $\partial_{\{c\}}(q \| \bar{c}(2))$ has a deadlock at time 2.

$$\begin{array}{lcl} \text{ACP}\rho \vdash \ \partial_{\{c\}}(p \| \bar{c}(2)) & = & a(1) \cdot \tau(2) \\ \text{ACP}\rho \vdash \ \partial_{\{c\}}(q \| \bar{c}(2)) & = & a(1) \cdot \tau(2) + a(1) \cdot \delta(2) \end{array}$$

This counterexample is due to Jan Bergstra ([Be92]). Hence, weak bisimulation is not a congruence in ACPρ. Therefore, we think that weak bisimulation is not appropriate for extension with time, at least in the context of ACP and ACPρ.

10 Conclusions

We have defined several formalizations of abstraction from untimed process algebra into real time process algebra. The characterizing laws are very close to the untimed ones. Weak bisimulation is not very appropriate since it is not a congruence for real time ACP, the extension of real time BPA with parallelism and synchronization. We expect that branching bisimulation and delay bisimulation, are congruences for real time ACP. However, we do not have formal proofs for these facts yet.

Timed delay bisimulation was already found along another route in earlier work ([Klu91b]). There it was motivated as a possible interpretation of abstraction in real time by interpreting τ-transition as idle transitions. In this paper it is more motivated as a real time formulation of a untimed delay equivalence. In [Klu91b] a protocol verification has been carried out as well. The law ATB is the only law which is used there, hence, this verification is sound for branching bisimulation as well.

The completeness results for (timed) delay and weak bisimulation hold only for a restricted set of terms, it is unclear yet whether a complete axiomatization can be found.

A lot of work still has to be done. For example, recursion has to be studied for strong, branching and delay bisimulation. Other proposals for introducing the silent step into a timed process algebra can be found in Wang's thesis ([Wan90]) and [MT92]. Both proposals concern timed extensions of CCS. A comparison is not easy due to the subtle differences in syntax and semantics, but it definitely has to be done. For example, our conclusions depend on the fact that the transition systems we study obey the property that every transition increases time. Both Wang and Moller &Tofts have two types of transitions; one for passing the time and one for the execution of actions. They allow consecutive actions at the same point in time. Moreover, we do not have *maximal progress* (the assumption that internal actions happen immediately when possible). However, Wang's work is based on maximal progress. The argument that maximal progress can be handled easily in BPA$\rho\delta$I by extending it with the priority operator ([BW90]) is not a proper one, since the interaction in real time process algebra between the priority operator and branching bisimulation is not trivial, though it is not studied yet.

Finally, it has to be studied whether the theory we have build so far enables the verification of real time protocols. Vaandrager and Lynch ([VL92]) argue that in real time any bisimulation equivalence is to fine since no abstraction is possible from the timings of external actions. This phenomenon can be seen as well in the protocol verification of [Klu91b].

Acknowledgements

Jos Baeten (Eindhoven Technical University) and Jan Bergstra (University of Amsterdam) are thanked for their encouraging and critical comments. Willem Jan Fokkink (CWI) is thanked for the many technical discussions and for the errors he showed me in earlier versions. Frits Vaandrager (CWI) is thanked for some final improvements.

References

[Bae92] J.C.M. Baeten. Personal Communication, 1992.

[BB90] J.C.M. Baeten and J.A. Bergstra. Process algebra with signals and conditions. Report P9008, University of Amsterdam, Amsterdam, 1990.

[BB91] J.C.M. Baeten and J.A. Bergstra. Real time process algebra. *Journal of Formal Aspects of Computing Science*, 3(2):142–188, 1991.

[BB92] J.C.M. Baeten and J.A. Bergstra. Discrete real time process algebra. In [Cl92].

[Be92] J.A. Bergstra. Personal Communication, 1992.

[BG87] J.C.M. Baeten and R.J. van Glabbeek. Another look at abstraction in process algebra. In Th. Ottman, editor, *Proceedings* 14^{th} *ICALP*, Karlsruhe, volume 267 of *Lecture Notes in Computer Science*, pages 84–94. Springer-Verlag, 1987.

[BW90] J.C.M. Baeten and W.P. Weijland. *Process algebra.* Cambridge Tracts in Theoretical Computer Science 18. Cambridge University Press, 1990.

[Cl92] W.R. Cleaveland, editor. *Proceedings CONCUR '92,* Stony Brook, USA. Volume 630 of *Lecture Notes in Computer Science.* Springer-Verlag, 1992.

[FK92] W.J. Fokkink and A.S. Klusener. Real time process algebra with prefixed integration. Report CS-R9219, CWI, Amsterdam, 1992.

[Fok91] W.J. Fokkink. Normal forms in real time process algebra. Report CS-R9149, CWI, Amsterdam, 1991.

[GW91] R.J. van Glabbeek and W.P. Weijland. Branching time and abstraction in bisimulation semantics. Report CS-R9120, CWI, Amsterdam, 1991. An extended abstract of an earlier version has appeared in G.X. Ritter, editor, *Information Processing 89*, North-Holland, 1989.

[HM80] M. Hennessy and R. Milner. On observing nondeterminism and concurrency. In J. de Bakker and J. van Leeuwen, editors, *Proceedings* 7^{th} *ICALP*, volume 85 of *Lecture Notes in Computer Science*, pages 299–309. Springer-Verlag, 1980. This is a preliminary version of *Algebraic laws for nondeterminism and concurrency.* JACM 32(1), pp. 137–161, 1985.

[Hoa85] C.A.R. Hoare. *Communicating Sequential Processes.* Prentice Hall International, 1985.

[Jef91] A. Jeffrey. Discrete timed CSP. Technical Report Memo 78, Chalmers University, Goteborg, 1991.

[Klu91a] A.S. Klusener. Completeness in real time process algebra. Report CS-R9106, CWI, Amsterdam, 1991. An extended abstract appeared in J.C.M. Baeten and J.F. Groote, editors, *Proceedings CONCUR 91*, Amsterdam. Volume 527 of *Lecture Notes in Computer Science*, pages 376–392. Springer-Verlag, 1991.

[Klu91b] A.S. Klusener. Abstraction in real time process algebra. Report CS-R9144, CWI, Amsterdam, 1991. An extended abstract appeared in J.W. de Bakker, C. Huizin· , W.P. de Roever and G. Rozenberg , editors, *Proceedings of the REX works op "Real-Time: Theory in Practice"*. Volume 600 of *Lecture Notes in Com, uter Science*, Springer-Verlag, 1991.

[Mil80] R. Milner. *A Calculus of Communicating Systems*, volume 92 of *Lecture Notes in Computer Science.* Springer-Verlag, 1980.

[Mil81] R. Milner. Modal characterisation of observable machine behaviour. In G. Astesiano and C. Bohm, editors, *Proceedings CAAP 81*, volume 112 of *Lecture Notes in Computer Science*, pages 25–34. Springer-Verlag, 1981.

[Mil83] R. Milner. Calculi for synchrony and asynchrony. *Theoretical Computer Science*, 25:267–310, 1983.

[Mil89] R. Milner. *Communication and concurrency.* Prentice Hall International, 1989.

[MT90] F. Moller and C. Tofts. A temporal calculus of communicating systems. In J.C.M. Baeten and J.W. Klop, editors, *Proceedings CONCUR 90,* Amsterdam, volume 458 of *Lecture Notes in Computer Science*, pages 401–415. Springer-Verlag, 1990.

[MT92] F. Moller and C. Tofts. Behavioural abstraction in TCCS. In *Proceedings ICALP 92,* Vienna, Lecture Notes in Computer Science. Springer-Verlag, 1992.

[Plo81] G.D. Plotkin. A structural approach to operational semantics. Report DAIMI FN-19, Computer Science Department, Aarhus University, 1981.

[VL92] F.W. Vaandrager and N.A. Lynch. Action transducers and timed automata. In [Cl92].

[Wan90] Y. Wang. *A Calculus of Real Time Systems.* PhD thesis, Chalmers University of Technology, Göteborg, 1990.