τ-angelic choice for process algebra
(Revised version)

Pedro R. D'Argenio
LIFIA, Depto. de Informática, Fac. Cs. Exactas,
Universidad Nacional de La Plata,
CC 11, 1900 La Plata, Buenos Aires, Argentina.
pedro@info.unlp.edu.ar

Abstract

The τ-angelic choice is an operator that captures the behaviour of the external choice of CSP in a branching time setting. The idea of the τ-angelic choice is to delay any choice until an observable action happens. In this way, this new operator avoids preemption introduced by internal actions (τ actions). It is studied in theories with abstraction, more precisely, branching bisimulation and τ-bisimulation and failure semantics. In addition, an ilustrative example of application is given.

Keywords: theory of concurrency, semantics of reactive systems, process algebra, non-determinism, specification and verification methodologies

1 Introduction

The aim of this article is to introduce a new choice operator for process algebras [BW90] in order to deal with preemptive contexts. This operator is called *τ-angelic choice* and denoted by $\Box$. The τ-angelic choice is strongly based in the external choice of CSP [BHR84, Hoa85] which was presented in a failure model. I showed in [D'A94] that external choice preserves several bisimulation equivalences. It motivates the study of such operator into a branching time setting. Since some characteristical properties of the external choice do not hold in finer equivalences, I decided to change its name to avoid confusion. So, in this article, the τ-angelic choice is axiomatizatized in the context of process algebras considering bismulation [Par81], branching bismulation [GW89] and τ-bismulation [Mil89] semantics. Besides, it is also studied in the context of failure semantics in order to compare with the CSP external choice.

Preemption is the property of forbidding activity (see [Mil89]). Sometimes, one needs to describe a behaviour which always allows the environment to choose. In this case, preemption is not a good factor. The choice operator introduced in this article avoids preemption. Since this phenomenon is a consequence of internal activity, which is represented by τ, $\Box$ behaves angelically when a τ is around it, that is, the choice is delayed while internal activity is performed.

Consider, for instance, the process $P = \mathsf{left} + \mathsf{right}$, where $+$ is the usual choice of CCS and ACP. When running P, one can observe that either left or right are able to be executed. Hence, each one of them can be chosen to execute. Now, suppose $\tau \cdot \mathsf{left} + \tau \cdot \mathsf{right}$ is running. Since τ represents the internal (and so, non-observable) activity, after some time, one can

observes that left is about to execute or that right is about to execute, but not both at the same time. This fact is caused by the preemption introduced by τ. On the contrary, the process $\tau \cdot \mathsf{left} \,\Box\, \tau \cdot \mathsf{right}$ allows to choose between left and right as in the case of P, even when internal activity is present.

Following this concepts, an example of application of the τ-angelic choice in verification is given. A verification in process algebra consists of a proof that the *actual behaviour* of a system *equals* to its *intended behaviour*. The actual behaviour is represented by the implementation specification, and the intended behaviour is modelled by the requirements specification. Often, the structure of the implementation specification is quite complex due to the presence of internal choices that shows implementations details, part of which may be irrelevant to prove correctness. These less interesting choices among internal actions can be filtered by mean of the τ-angelic choice.

This paper is organized as follows. Section 2 briefly explains preliminary concepts of process algebra and semantics of processes. Section 3 introduces the τ-angelic choice in the above mentioned settings. Several properties are studied in section 4. Associativity, commutativity and neutrality of the inaction are stated among others. Finally, an application example is developed in section 5.

Acknowledgements. I would like to thank Jos Baeten for his helpful technical suggestions that make this work improve considerably, and Juan Echagüe for his careful reading and suggestions of earlier versions of this article. I also would like to acknowledge suggestions and helps given by Javier Blanco, Sussan Doniz, Sjouke Mauw and Jan Joris Vereijken.

2 Basic process algebra with empty process

This section introduces the algebra of sequential processes [BW90], more precisely the basic process algebra with empty processes ($\mathrm{BPA}_{\delta\varepsilon}$). Concrete processes and several characterizations of abstraction will be considered: branching bisimulation [GW89], τ bisimulation [Mil89] and failure equivalence [BHR84, BKO86].

2.1 The equational theories

The signature of the several algebras is parametrized by a set of constants $\mathbf{A} = \{a, b, \ldots\}$ called *atomic actions*. There are three distinguished constants not belonging to $\mathbf{A}$: δ, called *deadlock* or *inaction*, that denotes a process that stopped executing action and cannot proceed; ε, the *empty process*, that denotes the process that does nothing but terminates succesfully; and τ, the *silent action* that is a special action having the meaning of internal activity. Besides, the signature has two binary operators: the *alternative composition* ($+$) that, in $x + y$, executes x or y but not both; and the *sequential composition* ($\cdot$) that, given $x \cdot y$, first executes x and, upon completion, starts with the execution of y. The sequential composition is usually omited, writing xy instead of $x \cdot y$. Furthermore, it is used the convention that $\cdot$ binds stronger than all other operators in this article while $+$ binds weaker. The set of closed terms in this signature is denoted by $\mathcal{T}$.

Equations for $\mathrm{BPA}_{\delta\varepsilon}$, the basic proces algebra with empty process for concrete proces, are given in Table 1. It is worthwhile to notice that in this article the silent action belongs to the

signature of $\mathrm{BPA}_{\delta\varepsilon}$ (which is not included in the original definition, see [BW90]) but, in this case, it is managed as any other action.

A1	$x + y = y + x$	A6	$\delta + x = x$
A2	$(x + y) + z = x + (y + z)$	A7	$\delta x = x$
A3	$x + x = x$		
A4	$(x + y)z = xz + yz$	A8	$\varepsilon x = x$
A5	$(xy)z = x(yz)$	A9	$x\varepsilon = x$

Table 1: Axioms for $\mathrm{BPA}_{\delta\varepsilon}$

Abstraction will be consider up to diferent view points (see subsection 2.2). By adding equation BE in Table 2 to those in Table 1, the set of equations for $\mathrm{BPA}^{\tau}_{\delta\varepsilon}$ [BW90] is obtained. The set of equations for $\mathrm{BPA}_{\delta\varepsilon\tau}$ [BK85, BW90] is defined by adding instead axioms in Table 3. Finally, the set of equations for $\mathrm{BPA}^{F}_{\delta\varepsilon}$ [BKO86] is defined by adding axioms in Table 4 to those in Table 1. Notice that $\mathrm{BPA}^{F}_{\delta\varepsilon} \vdash \{\mathrm{T2}, \mathrm{T3}\}$ and $\mathrm{BPA}_{\delta\varepsilon\tau} \vdash \mathrm{BE}$.

BE	$a(\tau(x + y) + x) = a(x + y)$

Table 2: Additional axiom for $\mathrm{BPA}^{\tau}_{\delta\varepsilon}$ ($a \in \mathbf{A} \cup \{\tau\}$)

T1	$a\tau = a$
T2	$\tau x + x = \tau x$
T3	$a(\tau x + y) = a(\tau x + y) + ax$

Table 3: Additional axioms for $\mathrm{BPA}_{\delta\varepsilon\tau}$ ($a \in \mathbf{A} \cup \{\tau\}$)

R	$a(bx + u) + a(by + v) = a(bx + by + u) + a(bx + by + v)$
T1	$a\tau = a$
TF	$\tau x + y = \tau x + \tau(x + y)$

Table 4: Additional axioms for $\mathrm{BPA}^{F}_{\delta\varepsilon}$ ($a, b \in \mathbf{A} \cup \{\tau\}$)

2.2 Structured operational semantics

Table 5 defines the operational semantics of $\mathrm{BPA}_{\delta\varepsilon}$ terms following the structured style of Plotkin [Plo81], that is, predicates of a term are stated in function of the predicates of its sub-terms. In this case, two kind of predicates are considered. Predicate $\downarrow (\subseteq \mathcal{T})$ expresses that a process may terminate succesfully. For every action $a \in \mathbf{A} \cup \{\tau\}$, predicate $\xrightarrow{a}$ $(\subseteq \mathcal{T} \times \mathcal{T})$ expresses that the first argument can perform action a and becomes the second argument.

$\varepsilon\downarrow$	$\dfrac{x\downarrow}{x+y\downarrow \quad y+x\downarrow}$	$\dfrac{x\downarrow \quad y\downarrow}{x\cdot y\downarrow}$	
$a\xrightarrow{a}\varepsilon$	$\dfrac{x\xrightarrow{a}x'}{x+y\xrightarrow{a}x' \quad y+x\xrightarrow{a}x'}$	$\dfrac{x\xrightarrow{a}x'}{x\cdot y\xrightarrow{a}x'\cdot y}$	$\dfrac{x\downarrow \quad y\xrightarrow{a}y'}{x\cdot y\xrightarrow{a}y'}$

Table 5: Operational semantics for the basic operators ($a \in \mathbf{A} \cup \{\tau\}$)

The reflexive transitive closure of $\xrightarrow{\tau}$ is denoted by $\Longrightarrow$. For every $\sigma \in \mathbf{A}^*$ the relation $\overset{\sigma}{\Longrightarrow} \in \mathcal{T} \times \mathcal{T}$ is defined by

$$p \overset{\sigma}{\Longrightarrow} q \overset{\text{def}}{\Longleftrightarrow} \sigma \equiv a_1 \ldots a_n \text{ and } p \Longrightarrow \xrightarrow{a_1} \Longrightarrow \cdots \Longrightarrow \xrightarrow{a_n} \Longrightarrow q.$$

Furthermore, $p \overset{a}{\not\longrightarrow}$ means that for no $q \in \mathcal{T}$, $p \xrightarrow{a} q$.

As it was expresed above, predicates give the idea of a certain executed operation that takes a system from some state to another. Now, a criterion to state whether two systems show the same behaviour is needed. Such a criterion is defined by means of the so-called *semantic equivalences*, which vary according to the way of observation of those systems (see [Gla90] and [Gla93]). The semantic equivalences used in this article are defined below. They are slight modifications of the original ones.

Definition 2.1 (Bisimulation semantics) [Par81] A *bisimulation* is a symmetric relation $S \subseteq \mathcal{T} \times \mathcal{T}$ satisfying, for all $a \in \mathbf{A} \cup \{\tau\}$:

- if pSq and $p \xrightarrow{a} p'$, then $\exists q' \in \mathcal{T} :\ q \xrightarrow{a} q'$ and $p'Sq'$; and
- if pSq then $p\downarrow \iff q\downarrow$.

Two processes p and q are *bisimilar* (notation $p \leftrightarroweq q$), if there exists a bisimulation S with pSq. ■

Definition 2.2 (Branching bisimulation semantics) [GW89] A *branching bisimulation* is a symmetric relation $S \subseteq \mathcal{T} \times \mathcal{T}$ satisfying, for all $a \in \mathbf{A} \cup \{\tau\}$:

- if pSq and $p \xrightarrow{a} p'$, then $\begin{cases} a = \tau \text{ and } p'Sq, \text{ or} \\ \exists q'', q' \in \mathcal{T} :\ q \Longrightarrow q'' \xrightarrow{a} q' \text{ with } pSq'' \text{ and } p'Sq'; \text{ and} \end{cases}$
- if pSq and $p\downarrow$ then $\exists q' \in \mathcal{T} :\ q \Longrightarrow q'\downarrow$ and pSq'.

Two processes p and q are *branching bisimilar* (notation $p \underline{\leftrightarrow}_b q$), if there exists a branching bisimulation S with pSq.

Two processes p and q are *rooted branching bisimilar*, (notation $p \underline{\leftrightarrow}_{rb} q$) if for all $a \in \mathbf{A} \cup \{\tau\}$:

- $p \xrightarrow{a} p'$ implies $\exists q' : q \xrightarrow{a} q'$ and $p' \underline{\leftrightarrow}_b q'$;
- $q \xrightarrow{a} q'$ implies $\exists p' : p \xrightarrow{a} p'$ and $p' \underline{\leftrightarrow}_b q'$; and
- $p \downarrow \iff q \downarrow$.

■

Definition 2.3 (τ-bisimulation semantics) [Mil80, Mil89] A *τ-bisimulation* is a symmetric relation $S \subseteq \mathcal{T} \times \mathcal{T}$ satisfying:

- for all $a \in \mathbf{A}$, if pSq and $p \xrightarrow{a} p'$, then $\exists q' \in \mathcal{T} : q \Longrightarrow \xrightarrow{a} \Longrightarrow q'$ and $p'Sq'$;
- if pSq and $p \xrightarrow{\tau} p'$, then $\exists q' \in \mathcal{T} : q \Longrightarrow q'$ and $p'Sq'$; and
- if pSq and $p \downarrow$ then $\exists q' \in \mathcal{T} : q \Longrightarrow q' \downarrow$.

Two processes p and q are *τ-bisimilar* (notation $p \underline{\leftrightarrow}_\tau q$), if there exists a τ-bisimulation S with pSq.

Two processes p and q are *rooted τ-bisimilar*, (notation $p \underline{\leftrightarrow}_{r\tau} q$) if $p \underline{\leftrightarrow}_\tau q$ and:

- $p \xrightarrow{\tau} p'$ implies $\exists q'', q' \in \mathcal{T} : q \xrightarrow{\tau} q'' \Longrightarrow q'$ and $p' \underline{\leftrightarrow}_\tau q'$;
- $q \xrightarrow{\tau} q'$ implies $\exists p'', p' \in \mathcal{T} : p \xrightarrow{\tau} p'' \Longrightarrow p'$ and $p' \underline{\leftrightarrow}_\tau q'$;

■

Definition 2.4 (Failure semantics) [BHR84, Hoa85] Let $I(p) = \{a \in \mathbf{A} \mid \exists q \in \mathcal{T} : p \Longrightarrow \xrightarrow{a} \Longrightarrow q\} \cup \{\surd \mid \exists q \in \mathcal{T} : p \Longrightarrow q \downarrow\}$ be the set of *initial actions* of $p \in \mathcal{T}$. Define the set $F(p) \subseteq (\mathbf{A}^* \times \wp(\mathbf{A} \cup \{\surd\})) \cup \{\sigma\surd \mid \sigma \in \mathbf{A}^*\} \cup \{\tau\}$ of *failures* of p as the least set satisfying:

- $(\sigma, X) \in F(p)$ if $\exists q \in \mathcal{T} : p \stackrel{\sigma}{\Longrightarrow} q \stackrel{\tau}{\not\longrightarrow}$ and $X \cap I(q) = \emptyset$; and
- $\sigma\surd \in F(p)$ if $\exists q \in \mathcal{T} : p \stackrel{\sigma}{\Longrightarrow} q \downarrow$; and
- $\tau \in F(p)$ if $\exists q \in \mathcal{T} : p \xrightarrow{\tau} q$.

Two processes p and q are *failure equivalent* (notation $p \approx_F q$) if $F(p) \backslash \{\tau\} = F(q) \backslash \{\tau\}$. They are *failure congruent* (notation $p =_F q$) if $F(p) = F(q)$. ■

The relations above are ordered by set inclusion as follows:

$$\begin{array}{ccccccc} & \subset & \underline{\leftrightarrow}_{rb} & \subset & \underline{\leftrightarrow}_{r\tau} & \subset & =_F \\ \underline{\leftrightarrow} & & \cap & & \cap & & \cap \\ & \subset & \underline{\leftrightarrow}_b & \subset & \underline{\leftrightarrow}_\tau & \subset & \approx_F \end{array}$$

The following results are standard (see [BK85, BKO86, BW90])

Theorem 2.1 (The term models)

1. $\mathrm{BPA}_{\delta\varepsilon}$ *is a complete axiomatization for* $\mathcal{T}/\underline{\leftrightarrow}$.
2. $\mathrm{BPA}^{\tau}_{\delta\varepsilon}$ *is a complete axiomatization for* $\mathcal{T}/\underline{\leftrightarrow}_{rb}$.
3. $\mathrm{BPA}_{\delta\varepsilon\tau}$ *is a complete axiomatization for* $\mathcal{T}/\underline{\leftrightarrow}_{r\tau}$.
4. $\mathrm{BPA}^{F}_{\delta\varepsilon}$ *is a complete axiomatization for* $\mathcal{T}/=_F$.

3 The τ angelic choice

3.1 Equational theory

This subsection introduces a new choice operator where the choice of the process to be performed is uniquely made by means of observable actions. Thus, the selection is always delayed until an observable action may engage. Because of its behaviour with respect to the silent action, this new operator is called *τ-angelic choice*.

The τ-angelic choice is denoted by the operator $\Box$ in order to keep Hoare's notation (see [BHR84, Hoa85]), and its definition is given in Table 6. In order to give a finite axiomatization,

BX	$x \Box y = x \sqsubset y + y \sqsubset x$		
LB1	$\varepsilon \sqsubset x = \varepsilon$	LB3	$ax \sqsubset y = ax$
LB2	$\tau x \sqsubset y = \tau(x \Box y)$	LB4	$(x+y) \sqsubset z = x \sqsubset z + y \sqsubset z$

Table 6: Additional axioms for the τ-angelic choice ($a \in \mathbf{A} \cup \{\delta\}$)

the auxiliar operator $\sqsubset$ (*left box*) is introduced. $x \sqsubset y$ choose to execute process x if it can terminate or do some observable action. If x executes instead some silent action, such an action is executed and then, a choice between y and the remaining execution of x is made. Now, $x \Box y$ considers two cases: the first one observes the initial actions of x and decides what will be executed ($x \sqsubset y$); the second one do the same but observing y ($y \sqsubset x$).

Adding the set of equations in Table 6 to $\mathrm{BPA}_{\delta\varepsilon}$, $\mathrm{BPA}^{\tau}_{\delta\varepsilon}$, $\mathrm{BPA}_{\delta\varepsilon\tau}$ and $\mathrm{BPA}^{F}_{\delta\varepsilon}$ the equational specifications $\mathrm{BPA}_{\delta\varepsilon} + \Box$, $\mathrm{BPA}^{\tau}_{\delta\varepsilon} + \Box$, $\mathrm{BPA}_{\delta\varepsilon\tau} + \Box$ and $\mathrm{BPA}^{F}_{\delta\varepsilon} + \Box$ are respectively obtained. Let $\mathbf{B}^{\Box} = \{\mathrm{BPA}_{\delta\varepsilon} + \Box, \mathrm{BPA}^{\tau}_{\delta\varepsilon} + \Box, \mathrm{BPA}_{\delta\varepsilon\tau} + \Box, \mathrm{BPA}^{F}_{\delta\varepsilon} + \Box\}$. Furthermore, the set of closed term in the new (extended) signature are denoted by $\mathcal{T}^{\Box}$.

Example 3.1 Some simple examples are introduced in order to make clear the behaviour of the τ-angelic choice. Let a, b and c be distinct actions in $\mathbf{A}$.

$$\begin{aligned}
\mathrm{BPA}_{\delta\varepsilon} + \Box &\vdash \tau(a+b) \Box \tau(a+c) = \tau(a+b+\tau(a+b+c)) + \tau(a+c+\tau(a+b+c)) \\
\mathrm{BPA}^{\tau}_{\delta\varepsilon} + \Box &\vdash \tau(a+b) \Box \tau(a+c) = \tau(a+b+c) \\
\mathrm{BPA}^{\tau}_{\delta\varepsilon} + \Box &\vdash (\tau(a+b)+c) \Box (\tau(a+c)+a+b) = \tau(a+b+c) + a + b + c \\
\mathrm{BPA}_{\delta\varepsilon\tau} + \Box &\vdash (\tau(a+b)+c) \Box (\tau(a+c)+a+b) = \tau(a+b+c) \\
\mathrm{BPA}_{\delta\varepsilon\tau} + \Box &\vdash (\tau a + \tau b) \Box \tau c = \tau(\tau(a+b) + \tau(a+c)) \\
\mathrm{BPA}^{F}_{\delta\varepsilon} + \Box &\vdash (\tau a + \tau b) \Box \tau c = \tau(a+b) + \tau(a+c)
\end{aligned}$$

■

3.2 Structured operational semantics

Predicates $\downarrow$ and $\xrightarrow{a}$ are extended to the set $\mathcal{T}^{\Box}$. Rules in Table 7 define the operational semantics for the τ-angelic choice. The whole rule system (Tables 5 and 7) is in *path* for-

mat [BV93]. It is clear that the notions of equivalences introduced in the previous pargraph

$$\frac{x\downarrow}{x\Box y\downarrow \quad y\Box x\downarrow} \qquad \frac{x\xrightarrow{a}x'}{x\Box y\xrightarrow{a}x' \quad y\Box x\xrightarrow{a}x'} \qquad \frac{x\xrightarrow{\tau}x'}{x\Box y\xrightarrow{\tau}x'\Box y \quad y\Box x\xrightarrow{\tau}y\Box x'}$$

$$\frac{x\downarrow}{x\sqsubset y\downarrow} \qquad \frac{x\xrightarrow{a}x'}{x\sqsubset y\xrightarrow{a}x'} \qquad \frac{x\xrightarrow{\tau}x'}{x\sqsubset y\xrightarrow{\tau}x'\Box y}$$

Table 7: Operational semantics for the τ-angelic choice ($a \in \mathbf{A}$)

can be easily extended to $\mathcal{T}^{\Box}$.

3.3 Soundness and Completeness

The aim of this section is to prove that the extended equational theories are sound and complete axiomatization of the several semantics equivalences. In order to do that term rewriting techniques are used. So, consider axioms A3 to A9 in Table 1 and all axioms in Table 6 with the left-to-right orientation as rewrite rules, i.e., if $s = t$ is one of those axioms, the respective rewrite rule will be $s \rightarrow t$. Nevertheless, this term rewriting system is not confluent (modulo A1, A2) since axiom A9 is sometimes needed in the opposite direction. So, the term rewriting system is completed with rules in Table 8 and it will be called TRS. Note that each new rewrite rule is derivable from $\mathrm{BPA}_{\delta\varepsilon}$ axioms.

LB2' $\quad \tau\sqsubset x \rightarrow \tau(\varepsilon\Box x)$	LB3' $\quad a\sqsubset x \rightarrow a$

Table 8: Additional rewrite rules ($a \in \mathbf{A} \cup \{\delta\}$)

Theorem 3.1 TRS *is strongly normalyzing.*

Proof. This can be proved by appliyng the method of the lexicographical path ordering [KL80, Klo92]. There is a complication: in the reduction of $\Box$, the operator $\sqsubset$ appears, but when reducing $\tau x\sqsubset y$ it happens the other way around. The solution is to weigh these operators by using semantic labelling [BK85, Zan93].

Thus, the operators are ordered as follow:

$$\mathbf{A}\cup\{\delta,\varepsilon,\tau\} < + < \cdot < \sqsubset_1 < \Box_1 < \cdots < \sqsubset_n < \Box_n < \sqsubset_{n+1} < \cdots$$

Besides, $\cdot$ has the lexicographical status for the left argument. What remains is routine calculations. ■

Definition 3.1 (Basic terms) Let B be the class of *basic terms* over any of the algebras in $\mathbf{B}^{\Box}$ defined as the smalest class satisfying:

1. $\tau, \varepsilon, \delta \in B$, $\mathbf{A} \subseteq A$
2. $a \in \mathbf{A}, t \in B \implies a \cdot t \in B$ and $\tau \cdot t \in B$
3. $t, s \in B \implies s + t \in B$

■

The next theorem states that for every closed term in the extended signature there exists a basic term (not containing $\Box$ or $\boxdot$) such that they can be proved equal in all algebras in $\mathbf{B}^{\Box}$.

Theorem 3.2 (Elimination theorem) *Let* $\mathrm{BPA}_{\delta\varepsilon}{}^{*} + \Box$ *be one of the equational theories in* $\mathbf{B}^{\Box}$. *Let* t *be a closed* $\mathrm{BPA}_{\delta\varepsilon}{}^{*} + \Box$ *term. Then, there is a basic term* s *such that* $\mathrm{BPA}_{\delta\varepsilon}{}^{*} + \Box \vdash t = s$.

Proof. Due to Theorem 3.1, t has normal form s. Moreover, such an s is a basic term as it is shown in the following.

Firstly, take into account that rules A3–A9 rewrite a closed $\mathrm{BPA}_{\delta\varepsilon}$ into a basic one (see [BW90]). Now, if s contains a $\Box$ then rule BX can be applied and so s is not in normal form which contradicts the assumption. If s contains a $\boxdot$, take a smallest sub-term containing it, say $s_1 \boxdot s_2$. Since s_1 and s_2 do not contains $\Box$ and $\boxdot$, they are already basic terms and hence, one of the rules LB1–LB4, LB2' or LB3' can be applied. This concludes the proof. ■

Theorem 3.3 (Congruence) $\underline{\leftrightarrow}$, $\underline{\leftrightarrow}_{rb}$, $\underline{\leftrightarrow}_{r\tau}$ *and* $=_F$ *are congruences for* $\Box$ *and* $\boxdot$.

Proof. Since the rule system is in *path* format, the case of $\underline{\leftrightarrow}$ follows immediately from [BV93]. In other cases, the proof for $\Box$ follows the same lines of the proof of the external choice of CSPin [D'A94] and proof for $\boxdot$ is quite similar. ■

In addition, $\Box$ also preserves $\underline{\leftrightarrow}_b$, $\underline{\leftrightarrow}_\tau$ and $\approx_F$, although it is not the case of $\boxdot$. For instance, we know that $a \underline{\leftrightarrow}_b \tau a$, but $a = a \boxdot b \not\approx_F \tau a \boxdot b = \tau(a + b)$.

Theorem 3.4 (Soundness)

$$\mathcal{T}^{\Box}/\underline{\leftrightarrow} \models \mathrm{BPA}_{\delta\varepsilon} + \Box \qquad \mathcal{T}^{\Box}/\underline{\leftrightarrow}_{r\tau} \models \mathrm{BPA}_{\delta\varepsilon\tau} + \Box$$
$$\mathcal{T}^{\Box}/\underline{\leftrightarrow}_{rb} \models \mathrm{BPA}^{\tau}_{\delta\varepsilon} + \Box \qquad \mathcal{T}^{\Box}/=_F \models \mathrm{BPA}^{F}_{\delta\varepsilon} + \Box$$

Proof. As usual. For every axiom $s = t$ having free variables in X, define the relation $R = \{(\sigma(s), \sigma(t)) |\ \sigma \text{ substitutes variables in } X \text{ to closed terms}\} \cup Id$. It is not difficult to prove that R is a bisimulation, rooted branching bisimulation or rooted τ bisimulation according to the soundness property one wants to prove.

In order to prove soundness in the case of failure congruence, it is enough to prove that for every axiom $s = t$ having free variables in X, for every substitution σ from X to closed terms, $F(\sigma(s)) = F(\sigma(t))$. ■

Theorem 3.5 (Conservative extension)

1. $\mathrm{BPA}_{\delta\varepsilon} + \Box$ *is a conservative extension of* $\mathrm{BPA}_{\delta\varepsilon}$.
2. $\mathrm{BPA}^{\tau}_{\delta\varepsilon} + \Box$ *is a conservative extension of* $\mathrm{BPA}^{\tau}_{\delta\varepsilon}$.
3. $\mathrm{BPA}_{\delta\varepsilon\tau} + \Box$ *is a conservative extension of* $\mathrm{BPA}_{\delta\varepsilon\tau}$.
4. $\mathrm{BPA}^{F}_{\delta\varepsilon} + \Box$ *is a conservative extension of* $\mathrm{BPA}^{F}_{\delta\varepsilon}$.

Proof. The proof applies general results of conservative extension of [Ver94] (see also [BV95]). The operational conservative extension follows since the rules are pure, well founded and positive, and no new rule for an old operator is introduced. This implies operational conservative extension up to $\underline{\leftrightarrow}$, $\underline{\leftrightarrow}_{rb}$, $\underline{\leftrightarrow}_{r\tau}$ and $=_F$. Since $\mathrm{BPA}_{\delta\varepsilon}$, $\mathrm{BPA}^{\tau}_{\delta\varepsilon}$, $\mathrm{BPA}_{\delta\varepsilon\tau}$ and $\mathrm{BPA}^{F}_{\delta\varepsilon}$ are complete axiomatizations (Theorem 2.1), and $\mathrm{BPA}_{\delta\varepsilon} + \Box$, $\mathrm{BPA}^{\tau}_{\delta\varepsilon} + \Box$, $\mathrm{BPA}_{\delta\varepsilon\tau} + \Box$ and $\mathrm{BPA}^{F}_{\delta\varepsilon} + \Box$ are sound (Theorem 3.4), equational conservative extension follows from [Ver94, BV95]. ∎

Theorem 3.6 (Completeness)

1. $\mathrm{BPA}_{\delta\varepsilon} + \Box$ *is a complete axiomatization of* $\mathcal{T}^{\Box}/\underline{\leftrightarrow}$.
2. $\mathrm{BPA}^{\tau}_{\delta\varepsilon} + \Box$ *is a complete axiomatization of* $\mathcal{T}^{\Box}/\underline{\leftrightarrow}_{rb}$.
3. $\mathrm{BPA}_{\delta\varepsilon\tau} + \Box$ *is a complete axiomatization of* $\mathcal{T}^{\Box}/\underline{\leftrightarrow}_{r\tau}$.
4. $\mathrm{BPA}^{F}_{\delta\varepsilon} + \Box$ *is a complete axiomatization of* $\mathcal{T}^{\Box}/=_F$.

Proof. Again, following [Ver94, BV95] and considering Theorem 3.2, this theorem is a corollary of the previous one. ∎

4 Properties

4.1 Expansion laws

The aim of an expansion theorem is to break down expression containing $\Box$ and to deal with the immediate context to be executed. Two expansion theorems are given following two difernt syles.

Theorem 4.1 (The expansion theorem)

a) For all terms $p_1, \ldots, p_n$, *in all equational theories in* $\mathbf{B}^{\Box}$, *it holds that*

$$p_1 \Box \ldots \Box p_n = \sum_{i=1}^{n} p_i \sqsubset (p_1 \Box \ldots p_{i-1} \Box p_{i+1} \ldots \Box p_n)$$

b) Assume $p = \sum_i a_i p_i + \sum_j \tau p_j + \sum_k \varepsilon$ *and* $q = \sum_m b_m q_m + \sum_n \tau q_n + \sum_l \varepsilon$ *where* $a_i \neq \tau$ *and* $b_m \neq \tau$ *for all* i *and* m. *Then, in all equational theories in* $\mathbf{B}^{\Box}$, *it holds that*

$$p \sqsubset q = \textstyle\sum_i a_i p_i + \sum_j \tau(p_j \Box q) + \sum_k \varepsilon$$

$$p \Box q = \textstyle\sum_i a_i p_i + \sum_m b_m q_m + \sum_j \tau(p_j \Box q) + \sum_n \tau(p \Box q_n) + \sum_{k+l} \varepsilon$$

Proof. a) In order to avoid notation $\overset{n}{\underset{i=1}{\Box}} p_i$ will be written instead of $p_1 \Box \ldots \Box p_n$. Induction on n is applied. Thus, if $n = 2$ the theorem follows from axiom BX. If $k > 2$ then

$$
\begin{array}{lll}
p_1 \Box \ldots \Box p_{n+1} & = & (\overset{n}{\underset{i=1}{\Box}} p_i) \Box p_{n+1} \\
& \overset{\text{BX}}{=} & (\overset{n}{\underset{i=1}{\Box}} p_i) \sqsubset p_{n+1} + p_{n+1} \sqsubset (\overset{n}{\underset{i=1}{\Box}} p_i) \\
& \overset{\text{IH}}{=} & \left(\sum_{j=1}^{n} p_j \sqsubset (\overset{n}{\underset{i=1,i\neq j}{\Box}} p_i) \right) \sqsubset p_{n+1} + p_{n+1} \sqsubset (\overset{n}{\underset{i=1}{\Box}} p_i) \\
& \overset{\text{A4}}{=} & \left(\sum_{j=1}^{n} \left(p_j \sqsubset (\overset{n}{\underset{i=1,i\neq j}{\Box}} p_i) \right) \sqsubset p_{n+1} \right) + p_{n+1} \sqsubset (\overset{n}{\underset{i=1}{\Box}} p_i) \\
& \overset{\text{T.4.3}}{=} & \sum_{j=1}^{n} p_j \sqsubset \left((\overset{n}{\underset{i=1,i\neq j}{\Box}} p_i) \Box p_{n+1} \right) + p_{n+1} \sqsubset (\overset{n}{\underset{i=1}{\Box}} p_i) \\
& \overset{\text{T.4.2, T.4.3}}{=} & \sum_{j=1}^{n} p_j \sqsubset (\overset{n+1}{\underset{i=1,i\neq j}{\Box}} p_i) + p_{n+1} \sqsubset (\overset{n}{\underset{i=1}{\Box}} p_i) \\
& \overset{\text{A1, A2}}{=} & \sum_{j=1}^{n+1} p_j \sqsubset (\overset{n+1}{\underset{i=1,i\neq j}{\Box}} p_i)
\end{array}
$$

b) Straightforward from the $\text{BPA}_{\delta\varepsilon} + \Box$ axioms. ∎

4.2 Choice laws

$\Box$ satisfies some classical properties of choice operators, especially, commutativity and associativity. Besides, δ is the neutral element for $\Box$. These properties will be proved in the following.

Theorem 4.2 (Commutativity) *For all terms x and y, in all equational theories in $\mathbf{B}^{\Box}$, it holds that $x \Box y = y \Box x$*

Proof. $x \Box y \overset{\text{BX}}{=} x \sqsubset y + y \sqsubset x \overset{\text{A1}}{=} y \sqsubset x + x \sqsubset y \overset{\text{BX}}{=} y \Box x$. ∎

Theorem 4.3 (Associativity) *For all closed terms p,q and r, in all equational theories in $\mathbf{B}^{\Box}$, it holds that:*

1. $(p \sqsubset q) \sqsubset r = p \sqsubset (q \Box r)$, *and*
2. $(p \Box q) \Box r = p \Box (q \Box r)$.

Proof. Since p, q and r are closed, without lost of generality it will be assumed they are basic term (see Theorem 3.2). 1. and 2. will be proved simultaneously by induction on the sum k of the number of symbols of p, q and r. Case $k = 3$ is left to the reader.

Suppose $k > 3$. Suppose $p = \sum_i a_i p_i + \sum_j \tau p_j + \sum_l \varepsilon$ with $a_i \in \mathbf{A}$. Then

1. $(p \sqsubset q) \sqsubset r$

$$
\begin{array}{lll}
\overset{\text{T.4.1.b}}{=} & (\sum_i a_i p_i + \sum_j \tau(p_j \Box q) + \sum_l \varepsilon) \sqsubset r & \overset{\text{T.4.1.b}}{=} \quad \sum_i a_i p_i + \sum_j \tau((p_j \Box q) \Box r) + \sum_l \varepsilon \\
\overset{\text{IH}}{=} & \sum_i a_i p_i + \sum_j \tau(p_j \Box (q \Box r)) + \sum_l \varepsilon & \overset{\text{T.4.1.b}}{=} \quad (\sum_i a_i p_i + \sum_j \tau p_j + \sum_l \varepsilon) \sqsubset (q \Box r) \\
= & p \sqsubset (q \Box r) &
\end{array}
$$

2. $(p\Box q)\Box r$

$$
\begin{array}{lll}
\overset{\mathrm{BX}}{=} & (p\Box q)\sqsubset r + r\sqsubset(p\Box q) & \overset{\mathrm{BX}}{=} \quad (p\sqsubset q + q\sqsubset p)\sqsubset r + r\sqsubset(p\Box q) \\
\overset{\mathrm{LB4,\ T.4.2}}{=} & (p\sqsubset q)\sqsubset r + (q\sqsubset p)\sqsubset r + r\sqsubset(q\Box p) & \overset{(1)}{=} \quad p\sqsubset(q\Box r) + q\sqsubset(p\Box r) + (r\sqsubset q)\sqsubset p \\
\overset{\mathrm{T.4.2}}{=} & p\sqsubset(q\Box r) + q\sqsubset(r\Box p) + (r\sqsubset q)\sqsubset p & \overset{(1)}{=} \quad p\sqsubset(q\Box r) + (q\sqsubset r)\sqsubset p + (r\sqsubset q)\sqsubset p \\
\overset{\mathrm{LB4}}{=} & p\sqsubset(q\Box r) + (q\sqsubset r + r\sqsubset q)\sqsubset p & \overset{\mathrm{BX}}{=} \quad p\sqsubset(q\Box r) + (q\Box r)\sqsubset p \\
\overset{\mathrm{BX}}{=} & p\Box(q\Box r) &
\end{array}
$$

■

Theorem 4.4 (δ is the neutral element for $\Box$) *For all closed term p, in all equational theories in $\mathbf{B}^{\Box}$, it holds that:*

1. $p\sqsubset\delta = p$, *and*
2. $p\Box\delta = \delta\Box p = p$.

Proof. As before, suppose p is a basic term. 1. and 2. are proved by simultaneous induction on the number of symbols of p. Case $k = 1$ is straightforward.

Suppose $k > 1$ and assume $p = \sum_i a_i p_i + \sum_j \tau p_j + \sum_l \varepsilon$ with $a_i \in \mathbf{A}$. Then

1. $p\sqsubset\delta \overset{\mathrm{T.4.1.b}}{=} \sum_i a_i p_i + \sum_j \tau(p_j\Box\delta) + \sum_l \varepsilon \overset{\mathrm{IH}}{=} \sum_i a_i p_i + \sum_j \tau p_j + \sum_l \varepsilon = p$

2. $p\Box\delta \overset{\mathrm{BX}}{=} p\sqsubset\delta + \delta\sqsubset p \overset{(1),\ \mathrm{A9,\ LB3}}{=} p + \delta(\varepsilon\Box p) \overset{\mathrm{A7,\ A6}}{=} p$

Finally $\delta\Box p = p$ holds by Theorem 4.2 ■

It was already stated that $\Box$ is commutative and associative and has δ as neutral element. However, the τ angelic choice is not idempotent in $\mathrm{BPA}_{\delta\varepsilon} + \Box$, $\mathrm{BPA}^{\tau}_{\delta\varepsilon} + \Box$ and $\mathrm{BPA}_{\delta\varepsilon\tau} + \Box$. Furthermore, the several distributivity laws do not hold in none of the equational specifications introduced here.

Fact 4.5 *The following equation does* **not** *hold in none of the initial algebras of* $\mathrm{BPA}_{\delta\varepsilon} + \Box$, $\mathrm{BPA}^{\tau}_{\delta\varepsilon} + \Box$ *and* $\mathrm{BPA}_{\delta\varepsilon\tau} + \Box$*:*

$$x = x\Box x$$

Proof. $(\tau + a)\Box(\tau + a) = \tau(\varepsilon + \tau\varepsilon + a) + a$ but $\tau(\varepsilon + \tau\varepsilon + a) + a \not\leftrightarrows_{\tau\tau} \tau + a$. The fact follows by the relation of the semantic equivalences and Theorem 3.6. ■

Fact 4.6 *The following equations do* **not** *hold in the initial algebras of every equational specification in* $\mathbf{B}^{\Box}$*:*

1. $(x + y)\Box z = (x\Box z) + (y\Box z)$
2. $(x\Box y) + z = (x + z)\Box(y + z)$
3. $(x\Box y)z = xz\Box yz$
4. $z(x\Box y) = zx\Box zy$

Proof. In every case it is proved that the equations do not hold in $\mathrm{BPA}_{\delta\varepsilon}^{F} + \Box$ taking into account the term model. The fact follows from the relation among the semantic equivalences and Theorem 3.6.

1. $Lf = (a + \tau b)\Box\tau c = a + \tau(b + \tau(b + c)) + \tau(a + c + \tau(b + c))$
   $= \tau(a + c + \tau(b + c)) = \tau(b + c) + \tau(a + b + c)$
   while $Rt = (a\Box\tau c) + (\tau b\Box\tau c) = a + \tau(a + c) + \tau(b + \tau(b + c)) + \tau(c + \tau(b + c))$
   $= \tau(a + c) + \tau(b + c)$.
   Now, the pair $(\langle\rangle, \{b\})$ is in $F(Rt)$ but not in $F(Lf)$ ($\langle\rangle$ is the empty sequence).

2. $Lf = (a\Box\tau b) + c = a + \tau(a + b) + c = \tau(a + b) + \tau(a + b + c)$
   while $Rt = (a + c)\Box(\tau b + c) = a + c + \tau(a + b + c) = \tau(a + b + c)$.
   Now, $(\langle\rangle, \{c\})$ is in $F(Lf)$ but not in $F(Rt)$.

3. $Lf = (a\Box\varepsilon)\tau = (a + \varepsilon)\tau = a\tau + \varepsilon\tau = a + \tau$
   while $Rt = a\tau\Box\varepsilon\tau = a\Box\tau = a + \tau(\varepsilon + a) = \tau(\varepsilon + a)$.
   Then, $(\langle\rangle, \{a\})$ is in $F(Rt)$ but not in $F(Lf)$.

4. $Lf = a(b\Box c) = a(b + c)$
   while $Rt = ab\Box ac = ab + ac$,
   and $(\langle a\rangle, \{c\})$ is in $F(Rt)$ but not in $F(Lf)$.

■

4.3 Other properties

This paragraph introduces properties mainly related with the operational behaviour of the processes.

A term x *may terminate immediately* in a process algebra if it can be proved that $x = x + \varepsilon$ in such an algebra. Thus, the following theorem states that if one of the terms involved in a τ-angelic composition may terminate, so does the whole term.

Theorem 4.7 (Termination of a τ-angelic composition) *Let* $\mathrm{BPA}_{\delta\varepsilon}{}^{*} + \Box \in \mathbf{B}^{\Box}$. *For all* $\mathrm{BPA}_{\delta\varepsilon}{}^{*} + \Box$ *terms* x *and* y, *if* x *may terminate immediately in* $\mathrm{BPA}_{\delta\varepsilon}{}^{*} + \Box$, *so does* $x\Box y$ *and* $y\Box x$.

Proof. Straightforward calculations using $\mathrm{BPA}_{\delta\varepsilon} + \Box$ axioms. ■

I decided to include this theorem because it is not always true that a choice composition preserves immediate termination; consider for instance, the internal choice of CSP [Hoa85] or the static choice of [BB94].

As it was shown before, $\Box$ is not generally idempotent. Nevertheles, the following theorem can be easily proved.

Theorem 4.8 *In all equational specifications in* $\mathbf{B}^{\Box}$ *it holds that* $x\Box x = x \sqsubset x$.

A process has local deadlock if it can reach an inaction state without performing any visible action. Formally:

Definition 4.1 (Local deadlock) *$p \in \mathcal{T}^{\Box}$ has local deadlock* if and only if there exisits a q such that $p \Longrightarrow q$, $q \stackrel{a}{\not\longrightarrow}$ for all $a \in \mathbf{A} \cup \{\tau\}$ and it is not the case that $q \downarrow$. ■

Because of completness of the term models, it is not difficult to prove the following theorem:

Theorem 4.9 *Let $p \in \mathcal{T}^{\Box}$. p has local deadlock if and only if in any equational theory in $\mathbf{B}^{\Box}$ it can be proved that $p = \delta$ or $p = p + \tau.q$ for some q which has local deadlock.*

Notice that $p+q$ has local deadlock if at least one of the summands has local deadlock. On the contrary, $p \Box q$ requires both operands has local deadlock. That is, $\Box$ avoids local deadlock whenever it is possible.

Theorem 4.10 (Local deadlock avoidance) *Let $p, q \in \mathcal{T}^{\Box}$. $p \Box q$ has local deadlock if and only if p and q have local deadlock.*

Proof. It is enough to prove that $p \Longrightarrow p'$ and $q \Longrightarrow q' \iff p \Box q \Longrightarrow p' \Box q'$. Recalling that $\Longrightarrow \; = \; \stackrel{\tau}{\longrightarrow}^*$, it is an straightforward induction on the amount of τs. ■

Preemption is an immediate consequence of internal activity. This phenomenon introduces "unstability" of processes. A process p is said to be *stable* if $p \stackrel{\tau}{\not\longrightarrow}$, or, similarly, if p does not have a summand $\tau p'$. Thus, the following theorem can be easily proved:

Theorem 4.11 ($\Box$ is idempotent for stable processes) *Let $p \in \mathcal{T}^{\Box}$ be a stable process. In all equational theories in $\mathbf{B}^{\Box}$, it holds that*

1. *$p = p \sqsubset x$, for all x, and*
2. *$p = p \Box p$.*

4.4 Choice laws in $\mathrm{BPA}^F_{\delta\varepsilon} + \Box$

There are some choices laws that only hold in $\mathrm{BPA}^F_{\delta\varepsilon} + \Box$. They are: idempotency and distributivity of $\Box$ in preemptive contexts.

The next proposition is a simple corollary of results in [Bro83].

Proposition 4.12 *For all unstable closed term p, there exists a closed term q having the form $\sum_i \tau \cdot q_i$, where each q_i is stable, such that $\mathrm{BPA}^F_{\delta\varepsilon} + \Box \vdash p = q$.*

Theorem 4.13 (Idempotency in $\mathrm{BPA}^F_{\delta\varepsilon} + \Box$) *For all closed term p, $p \Box p = p$ holds in $\mathrm{BPA}^F_{\delta\varepsilon} + \Box$.*

Proof. If p is stable, it is the case of Theorem 4.11. On the other hand, since theorem 4.8, it is enough to prove that $p \sqsubset p = p$. Because of Proposition 4.12, assume $p = \sum_i \tau p_i$, where each p_i is stable. In addition, it can be proved that:

$$\mathrm{BPA}^F_{\delta\varepsilon} + \Box \vdash \tau(x + \tau y) = \tau(x + y) + \tau y \qquad (\star)$$

$$\mathrm{BPA}^F_{\delta\varepsilon} + \Box \vdash \tau x + \tau y + \tau(x + y) = \tau x + \tau y \qquad (\star\star)$$

Consider that i and i' range over the same index set, then

$$
\begin{array}{lll}
p \sqsubset p & \overset{\text{T.4.1.b}}{=} & \sum_i \tau \cdot (p_i \Box p) \\
& \overset{\text{BX}}{=} & \sum_i \tau \cdot (p_i \sqsubset p + p \sqsubset p_i) \\
& \overset{\text{T.4.11}}{=} & \sum_i \tau \cdot (p_i + p \sqsubset p_i) \\
& \overset{\text{T.4.1.b}}{=} & \sum_i \tau \cdot (p_i + \sum_{i'} \tau \cdot (p_{i'} \Box p_i)) \\
& \overset{\text{BX, T.4.11}}{=} & \sum_i \tau \cdot (p_i + \sum_{i'} \tau \cdot (p_{i'} + p_i)) \\
& \overset{(\star)}{=} & \sum_i (\tau \cdot (p_i + \sum_{i'} (p_{i'} + p_i)) + \sum_{i'} \tau \cdot (p_{i'} + p_i)) \\
& \overset{\text{A3}}{=} & \sum_i (\tau \cdot (\sum_{i'} p_{i'}) + \sum_{i'} \tau \cdot (p_{i'} + p_i)) \\
& \overset{\text{A3}}{=} & \sum_i (\tau \cdot (\sum_{i'} p_{i'}) + \sum_{i'} \tau p_{i'} + \sum_{i'} \tau \cdot (p_{i'} + p_i)) \\
& \overset{(\star\star)}{=} & \sum_i (\tau \cdot (\sum_{i'} p_{i'}) + \sum_{i'} \tau p_{i'}) \\
& \overset{(\star\star)}{=} & \sum_i \sum_{i'} \tau p_{i'} \\
& \overset{\text{A3}}{=} & \sum_{i'} \tau p_{i'} \\
& = & p
\end{array}
$$

■

Another property that holds on $\mathrm{BPA}_{\delta\varepsilon}^{F} + \Box$ is that $\Box$ is distributive with respect to preemptive contexts.

Theorem 4.14 (Distributivity of $\Box$ w.r.t. preemptive contexts in $\mathrm{BPA}_{\delta\varepsilon}^{F} + \Box$) *For all closed term p, $\mathrm{BPA}_{\delta\varepsilon}^{F} + \Box \vdash (\tau x + \tau y)\Box p = \tau(x \Box p) + \tau(y \Box p)$.*

Proof. It will be proved by induction on the size of p. Suppose p is a basic term. If $k = 1$ then, if $p = \delta$ the proof follows by Theorem 4.4. If $p = \varepsilon$, the prove is a bit difficult. Notice that axiom T2 is provable from A6 and TF. Thus,

$$
\begin{array}{l}
(\tau x + \tau y)\Box\varepsilon \\
\quad \begin{array}{lll}
\overset{\text{T.4.1.a}}{=} & \tau(x\Box\varepsilon) + \tau(y\Box\varepsilon) + \varepsilon & \overset{\text{BX, LB1}}{=} \quad \tau(x\Box\varepsilon) + \tau(y \sqsubset \varepsilon + \varepsilon) + \varepsilon \\
\overset{\text{T2, A3}}{=} & \tau(x\Box\varepsilon) + \tau(y \sqsubset \varepsilon + \varepsilon) & \overset{\text{LB1, BX}}{=} \quad \tau(x\Box\varepsilon) + \tau(y\Box\varepsilon)
\end{array}
\end{array}
$$

The case of $p = a$ follows the same lines. Case of $p = \tau$ is as follows

$$
\begin{array}{l}
(\tau x + \tau y)\Box\tau \\
\quad \begin{array}{ll}
\overset{\text{T.4.1.a}}{=} & \tau(x\Box\tau) + \tau(y\Box\tau) + \tau(\varepsilon\Box(\tau x + \tau y)) \\
\overset{\text{Case } \varepsilon}{=} & \tau(x\Box\tau) + \tau(y\Box\tau) + \tau(\tau(\varepsilon\Box x) + \tau(\varepsilon\Box y)) \\
\overset{\text{A3,R,T1}}{=} & \tau(x\Box\tau) + \tau(y\Box\tau) + \tau(\varepsilon\Box x) + \tau(\varepsilon\Box y) \\
\overset{\text{BX,A9,LB2}}{=} & \tau((x \sqsubset \tau) + \tau(\varepsilon\Box x)) + \tau((y \sqsubset \tau) + \tau(\varepsilon\Box x)) + \tau(\varepsilon\Box x) + \tau(\varepsilon\Box y) \\
\overset{\text{T2,A3}}{=} & \tau((x \sqsubset \tau) + \tau(\varepsilon\Box x)) + \tau((y \sqsubset \tau) + \tau(\varepsilon\Box x)) \\
\overset{\text{LB2,A9,BX}}{=} & \tau(x\Box\tau) + \tau(y\Box\tau)
\end{array}
\end{array}
$$

If $k > 1$, suppose $p \equiv \sum_i a_i p_i + \sum_j \tau p_j + \sum_l \varepsilon$. Then:

$$
\begin{array}{ll}
(\tau x + \tau y)\Box p & \\
\overset{\text{T.4.1.a}}{=} & \tau(x\Box p) + \tau(y\Box p) + \sum_i a_i p_i + \sum_j \tau(p_j\Box(\tau x + \tau y)) + \sum_l \varepsilon \\
\overset{\text{IH}}{=} & \tau(x\Box p) + \tau(y\Box p) + \sum_i a_i p_i + \sum_j \tau(\tau(p_j\Box x) + \tau(p_j\Box y)) + \sum_l \varepsilon \\
\overset{\text{BX, T.4.1.b}}{=} & \tau\left(x \sqsubset p + \sum_i a_i p_i + \sum_j \tau(p_j\Box x) + \sum_l \varepsilon\right) \\
& \quad + \tau\left(y \sqsubset p + \sum_i a_i p_i + \sum_j \tau(p_j\Box y) + \sum_l \varepsilon\right) \\
& \quad + \sum_i a_i p_i + \sum_j \tau(\tau(p_j\Box x) + \tau(p_j\Box y)) + \sum_l \varepsilon \\
\overset{\text{T2}}{=} & \tau\left(x \sqsubset p + \sum_i a_i p_i + \sum_j \tau(p_j\Box x) + \sum_l \varepsilon\right) \\
& \quad + \tau\left(y \sqsubset p + \sum_i a_i p_i + \sum_j \tau(p_j\Box y) + \sum_l \varepsilon\right) \\
& \quad + \sum_j \tau(\tau(p_j\Box x) + \tau(p_j\Box y)) \\
\overset{\text{A3, R, T1}}{=} & \tau\left(x \sqsubset p + \sum_i a_i p_i + \sum_j \tau(p_j\Box x) + \sum_l \varepsilon\right) \\
& \quad + \tau\left(y \sqsubset p + \sum_i a_i p_i + \sum_j \tau(p_j\Box y) + \sum_l \varepsilon\right) \\
& \quad + \sum_j (\tau(p_j\Box x) + \tau(p_j\Box y)) \\
\overset{\text{A1, A2}}{=} & \tau\left(x \sqsubset p + \sum_i a_i p_i + \sum_j \tau(p_j\Box x) + \sum_l \varepsilon\right) + \sum_j \tau(p_j\Box x) \\
& \quad + \tau\left(y \sqsubset p + \sum_i a_i p_i + \sum_j \tau(p_j\Box y) + \sum_l \varepsilon\right) + \sum_j \tau(p_j\Box y) \\
\overset{\text{T2, A3}}{=} & \tau\left(x \sqsubset p + \sum_i a_i p_i + \sum_j \tau(p_j\Box x) + \sum_l \varepsilon\right) \\
& \quad + \tau\left(y \sqsubset p + \sum_i a_i p_i + \sum_j \tau(p_j\Box y) + \sum_l \varepsilon\right) \\
\overset{\text{T.4.1.b, BX}}{=} & \tau(x\Box p) + \tau(y\Box p)
\end{array}
$$

■

5 An example

This section introduces a simple but representative example. It is borrow from [DM95] where it was introduced to show the use of the delay choice in context with abstraction. I made some simple modifications that show that the τ-angelic choice is a bit more versatile in this kind of applications. Proofs in this section will be done using the $\mathrm{BPA}^{\tau}_{\delta\varepsilon} + \Box$ equational specification. It is immediate that the obtained results also hold in $\mathrm{BPA}_{\delta\varepsilon\tau} + \Box$ and $\mathrm{BPA}^{F}_{\delta\varepsilon} + \Box$. It has no sense to use $\mathrm{BPA}_{\delta\varepsilon}$ since it has no criteria to manage abstraction. In addition, operators and axioms of $\mathrm{ACP}^{\tau}_{\varepsilon}$ will be considered (see [BW90]).

A verification in process algebra consists of proving that the requirement specification *Spec* equals to the observable part of the implementation specification *Imp*, i.e., $Spec = \tau_I(Imp)$, where τ_I is the abstraction operator [BW90] that renames actions from the set I into τ. However, τ_I only removes internal actions and keeps the branching structure of a process. It often happens that one wants to check a very simple specification requirement *Spec* while, after the calculation of $Spec = \tau_I(Imp)$, an expression with an excess of internal choices is obtained. Most of this internal choices represent implementation decisions that may be not relevant for the designer view. At this point, there are two ways to proceed. The first one may be to simply

forget about *Spec* and accept a more implementation directed requirement specification. The second one may be to change for a coarser ponit of view, i.e., a linear time semantics, but it makes one loose all information about branching structure of the requeriments.

I propose to use the τ-angelic choice in order to reduce only the internal branching structure which is irrelevant. Let S^τ be the operator that skips all internal choices until an observable action appears by replacing all occurrences of the alternative composition by the τ-angelic

ST1	$\mathsf{S}^\tau(\varepsilon) = \varepsilon$
ST2	$\mathsf{S}^\tau(\delta) = \delta$
ST3	$\mathsf{S}^\tau(x \cdot y) = \mathsf{S}^\tau(x) \cdot \mathsf{S}^\tau(y)$
ST4	$\mathsf{S}^\tau(x + y) = \mathsf{S}^\tau(x) + \mathsf{S}^\tau(y)$

Table 9: The S^τ operator to skip internal choices

choice as it is defined in Table 9. Now, a mixed linear time/branching time verification consist of proving the equation

$$Spec \;=\; \tau_{I_2} \circ \mathsf{S}^\tau \circ \tau_{I_1}(Imp)$$

where the set I_1 contains all atomic actions which induce non-interesting internal choices, while choices made by actions in I_2 should remain after abstraction.

The example is based in the verification of a *leader election protocol* called *Paint Ball protocol* [DM95].

Suppose that entities E_i $(i \in ID)$ have to elect a leader among themselves. A simple specification of this fact could be say that after certain internal activity a leader is chosen non-deterministically, i.e.,

$$Spec = \tau \cdot \sum_{i \in ID} \tau \cdot \mathsf{leader}(i)$$

The Paint Ball protocol is specified as the parallel composition of all entities:

$$Imp = \partial_H \left(\mathop{||}_{i \in ID} E_i \right)$$

An entity should defeat all the others to become a leader. Thus, each entity E_i is indexed by a set V that contains all other entities which have not yet been defeated by E_i. The entity E_i may send a message (s_{ij}) to defeat one of the other participants, or receive a message (r_{ji}) that defeats it, entering thus in a fail state (F_i). When all except one entities have been defeated, the leader (L_i) informs all failed entities that the election has finished by broadcasting a message (sready), and finally proclaming itself the leader $(\mathsf{leader}(i))$.

$$\begin{array}{lcll} E_i & = & E_i^{ID-\{i\}} & \\ E_i^V & = & \sum_{j \in V} \left(\mathsf{s}_{ij} \cdot E_i^{V-\{j\}} + \mathsf{r}_{ji} \cdot F_i \right) & (V \neq \emptyset) \\ E_i^\emptyset & = & L_i & \\ F_i & = & \sum_{j \in ID-\{i\}} \left(\mathsf{r}_{ji} \cdot F_i + \mathsf{rready} \right) & \\ L_i & = & \left(||_{j \in ID-\{i\}} \, \mathsf{sready} \right) \cdot \mathsf{leader}(i) & \end{array}$$

The comunication function and the encapsulation set are defined as usual: $\mathsf{r}_{ij}|\mathsf{s}_{ij} = \mathsf{c}_{ij}$, $\mathsf{rready}|\mathsf{sready} = \mathsf{cready}$ and $H = \{\mathsf{r}_{ij}, \mathsf{s}_{ij}|i,j \in ID\} \cup \{\mathsf{rready}, \mathsf{sready}\}$. Clearly, the set of internal action is $I = \{\mathsf{c}_{ij}|i,j \in ID\} \cup \{\mathsf{cready}\}$. After several calculations the implementation specification is reduced to $\tau_I(Imp) = P^{ID}$ where

$$\begin{array}{lcll} P^V & = & \sum_{i \in V} \tau \cdot P^{V-\{i\}} & (|V| > 1) \\ P^{\{i\}} & = & \mathsf{leader}(i) & \end{array}$$

which cannot be further reduced. In this specification, some internal choices denote that some entity i is removed from the list of candidates during the execution of the protocol. This goes on until only one candidate remains. According to the original requirements specification, these implementation details are not required. Hence, using the strategy above proposed, one can consider that the actions of defeating an entity are not relevant. So, define $I_1 = \{\mathsf{c}_{ij}|i,j \in ID\}$. Then

$$\mathsf{S}^\tau \circ \tau_{I_1}(Imp) \;=\; \tau \cdot \sum_{i \in ID} \left(\left(\mathop{\Vert}_{j \in ID-\{i\}} \mathsf{cready} \right) \cdot \mathsf{leader}(i) \right)$$

Now, let $I_2 = \{\mathsf{cready}\}$. Finally,

$$\tau_{I_2} \circ \mathsf{S}^\tau \circ \tau_{I_1}(Imp) \;=\; \tau \cdot \sum_{i \in ID} \tau \cdot \mathsf{leader}(i) \;=\; Spec$$

which proves the desired equality.

6 Concluding remarks

The CSP external choice has been redefined in a branching time setting. Complete axiomatizations for bisimulation, rooted branching bisimulation and rooted τ-bisimulation equivalences were given. It was also proved that the τ-angelic choice satisfies standard properties for choice operators (e.g. commutativity, associativity, existence of neutral element) as well as some other relevant properties as local deadlock avoidance. Moreover, several properties have been studied in the failure setting. They showed that in this context, the τ-angelic choice behaves exactly as the CSP external choice. In addition an interesting application case in verification was presented.

However, the τ-angelic choice is not totally angelic. An angelic choice [MO91] (say @) delays the choice as far as possible; thus, it tries to avoid deadlock even after performing visible actions. For instance, the process $a\delta$ has deadlock but not local deadlock. So, for any process p, $a\delta \Box p$, may deadlock after performing a. Particularly $a\delta \Box a\varepsilon$, may deadlock, although the process $a\delta @ a\varepsilon$ does not have dedlock at all. An study of the angelic choice in a branching and concrete setting was made by *Baeten & Mauw* in [BM95]. There, it receives the more appropriate name of delayed choice. *Mauw & I* [DM95] extend the delayed choice to deal with abstraction by borrowing the treatment given by the τ-angelic choice. *Baeten & Bergstra* [BB94] defined an operator similar to the τ-angelic choice in the context of concrete process algebras. They called it *dynamic sum* and also observed that idempotency does not generally hold for the dynamic sum.

I showed in [D'A94] that the external choice of CSP can be mapped to the τ-angelic choice preserving bisimulation equivalence.

References

[BB94] J.C.M. Baeten and J.A. Bergstra. Process algebra with partial choice. In B. Jhonsson and J. Parrow, editors, *Proceedings of CONCUR'94*, pages 465–480, Uppsala, 1994. LNCS 836, Springer-Verlag.

[BHR84] S.D. Brookes, C.A.R. Hoare, and A.W. Roscoe. A theory of communicating sequential processes. *Journal of the ACM*, 31(3):560–599, July 1984.

[BK85] J.A. Bergstra and J.W. Klop. Algebra of communicating processes with abstraction. *Theoretical Computer Science*, 37:77–121, 1985.

[BKO86] J.A. Bergstra, J.W. Klop, and E.-R. Olderog. Failure semantics with fair abstraction. Report CS-R8609, Centrum voor Wiskunde en Informatica, Amsterdam, January 1986.

[BM95] J.C.M. Baeten and S. Mauw. Delayed choice: an operator for joining Message Sequence Charts. In D. Hogrefe and S. Leue, editors, *Formal Description Techniques VII*. Chapman & Hall, 1995. To appear.

[Bro83] S.D. Brookes. On the relationship of CCS an CSP. In J. Diaz, editor, *Proceedings 10th. ICALP*, pages 83–96, Barcelona, 1983. LNCS 154, Springer-Verlag.

[BV93] J.C.M. Baeten and C. Verhoef. A congruence theorem for structured operational semantics with predicates. In E. Best, editor, *Proceedings of CONCUR'93*, pages 477–492, Hildesheim, 1993. LNCS 715, Springer-Verlag.

[BV95] J.C.M. Baeten and C. Verhoef. Concrete process algebra. Report 95/03, Department of Mathematics and Computer Science, Eindhoven University of Technology, 1995.

[BW90] J.C.M. Baeten and W.P. Weijland. *Process algebra*. Cambridge University Press, 1990.

[D'A94] Pedro R. D'Argenio. A comparative analisys of concurrency theories. Graduation thesis, Departamento de Informática, Facultad de Ciencias Exactas, Universidad Nacional de La Plata, September 1994.

[DM95] Pedro R. D'Argenio and Sjouke Mauw. Delayed choice for process algebras with abstraction, 1995. Submitted for publication. An extended version to appear as Report, Dept. of Computer Science, Eindhoven University of Technology.

[Gla90] R.J. van Glabbeek. The linear time - branching time specrtrum. In J.C.M. Beten and J.W. Klop, editors, *Proceedings of CONCUR'90*, pages 278–297, Amsterdam, 1990. LNCS 458, Springer-Verlag.

[Gla93] R.J. van Glabbeek. The linear time - branching time specrtrum II. (extended abstract). In E. Best, editor, *Proceedings of CONCUR'93*, pages 308–323, Hildesheim, 1993. LNCS 715, Springer-Verlag.

[GW89] R.J. van Glabbeek and W.P. Weijland. Branching time and abstraction in bisimulation semantics (extended abstract). In G.X. Ritter, editor, *Information Processing 89*, pages 613–618. North-Holland, 1989.

[Hoa85] C.A.R. Hoare. *Communicating sequential process*. Prentice Hall, 1985.

[KL80] S. Kamin and J.-J. Lévy. Two generalizations of the recursive path ordering, 1980. Unpublished manuscript.

[Klo92] J.W. Klop. Term rewriting systems. In S. Abramsky, D. Gabbay, and T. Maibaum, editors, *Handbook of Logic in Computer Science*, volume II, pages 1–116. Oxford University Press, 1992.

[Mil80] Robin Milner. *A calculus of communicating systems*, volume 92 of *LNCS*. Springer-Verlag, 1980.

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

[MO91] M.W. Mislove and F.J. Oles. A simple language supporting angelic nondeterminism and parallel composition. In S. Brookes, M. Main, A. Melton, M. Mislove, and D. Schmidt, editors, *Proceedings of Mathematical Foundations of Programming Semantics, 7th. International Conference*, pages 77–101, Pittsburgh, 1991. LNCS 598, Springer-Verlag.

[Par81] D.M.R. Park. Concurrency and automata on infinite sequence. In P. Deussen, editor, *Proceedings 5th. GI Conference*, pages 167–183. LNCS 104, Springer-Verlag, 1981.

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

[Ver94] C. Verhoef. A general conservative extension theorem in process algebra. In E.-R. Olderog, editor, *Proceedings of PROCOMET'94, IFIP 2 Working Conference*, pages 149–168, San Miniato, 1994. North-Holland.

[Zan93] H. Zantema. Termination of term rewriting by semantic labelling. Report RUU-CS-93-24, Department of Computer Science, Utrecht University, 1993.