Unique parallel decomposition for the π-calculus

Unique Parallel Decomposition for the π-calculus [*]

Matias David Lee
Univ. Lyon, ENS de Lyon, CNRS, UCB Lyon 1, LIP, France.
matias-david.lee@ens-lyon.fr

Bas Luttik
Eindhoven University of Technology, The Netherlands.
s.p.luttik@tue.nl

A (fragment of a) process algebra satisfies unique parallel decomposition if the definable behaviours admit a unique decomposition into indecomposable parallel components. In this paper we prove that finite processes of the π-calculus, i.e. processes that perform no infinite executions, satisfy this property modulo strong bisimilarity and weak bisimilarity. Our results are obtained by an application of a general technique for establishing unique parallel decomposition using decomposition orders.

1 Introduction

A (fragment of a) process algebra has *unique parallel decomposition* (UPD) if all definable behaviours admit a unique decomposition into indecomposable parallel components. In this paper we prove that finite processes definable in the π-calculus satisfy this property modulo strong bisimilarity and modulo weak bisimilarity.

From a theoretical point of view, this property is interesting because it can be used to prove other theoretical properties about process calculi. For instance, relying on unique parallel decomposition, Moller proves in [18, 19] that PA and CCS cannot be finitely axiomatized without auxiliary operations, and Hirshfeld and Jerrum prove in [12] that bisimilarity is decidable for normed PA. Unique parallel decomposition can be also used to define a notion of normal form. Such a notion of normal form is useful in completeness proofs for equational axiomatizations in settings in which an elimination theorem for parallel composition is lacking (see, e.g., [1, 2, 3, 9, 11]). In [13], UPD is used to prove complete axiomatisation and decidability results in the context of a higher-order process calculus.

From a practical point of view, unique parallel decomposition can be used to devise methods for finding the maximally parallel implementation of a behaviour [6], or for improving verification methods [10]. In [8], a unique parallel decomposition result is used as a tool in the comparison of different security notions in the context of electronic voting.

The UPD property has been widely studied for different process calculi and variants of the parallel operator. Milner and Moller were the first to establish a unique parallel decomposition theorem; they proved the property for a simple process calculus that allows the specification of all finite behaviours up to strong bisimilarity and includes parallel composition in the form of pure interleaving without interaction between its components [16]. Moller, in his dissertation [17], extended this result replacing interleaving parallel composition by CCS parallel composition, and then also considering weak bisimilarity. Christensen, also in his dissertation [5], proved unique decomposition for normed behaviours recursively definable modulo strong bisimilarity, and for all behaviours recursively definable modulo

[*] M.D. Lee has been supported by the project ANR 12IS02001 PACE

D. Gebler and K. Peters (Eds.): Combined Workshop on Expressiveness in
Concurrency and Structural Operational Semantics (EXPRESS/SOS 2016).
EPTCS 222, 2016, pp. 45–59, doi:10.4204/EPTCS.222.4

distributed bisimilarity; the proof of the latter result relies on a cancellation law for parallel composition up to distributed bisimilarity, first established by Castellani as [4, Lemma 4.14].

Most of the aforementioned unique parallel decomposition results were established with subsequent refinements of an ingenious proof technique attributed to Milner. In [15], the notion of *decomposition order* is introduced in order to formulate a sufficient condition on commutative monoids that facilitates an abstract version of Milner's proof technique. It is then proved that if a partial commutative monoid can be endowed with a decomposition order, then it has unique decomposition. Thus, an algebraic tool is obtained that allows one to prove UPD for a process calculus by finding a decomposition order.

The tool can deal with most of the settings aforementioned. In this paper, we show how the tool can also be applied to obtain unique parallel decomposition results for finite processes of the π-calculus w.r.t. strong bisimilarity and w.r.t. weak bisimilarity. But, to this end, we do face two complications: The first complication, in the context of the π-calculus is that, as opposed to previous settings, the decomposition order is not directly induced on the commutative monoid of processes by the transition relation. The culprit is that, in general, two parallel components may fuse into a single indecomposable process as a result of scope extrusion. To define the decomposition order we consider a fragment of the transition relation that avoids this phenomenon. The second complication, which arises only in the case of weak bisimilarity, is that certain transitions are deemed unobservable, and that, as a consequence, there are transitions that do not change state (are between weakly bisimilar processes). We demonstrate that a decomposition order can, nevertheless, be obtained by ignoring such *stuttering transitions*.

The paper [7] studies unique parallel decomposition w.r.t. both strong bisimilarity and weak bisimilarity for the applied π-calculus. The applied π-calculus is a variant of the π-calculus that was designed for the verification of cryptographic protocols. Its main feature is that channels can only transmit variables and the values of the variables are set using *active substitutions*. Roughly, active substitution is an extension of the grammar of the π-calculus that works as a 'memory' that save the value of a variable. Because the variables in a transition are observable but the 'memories' are not, it is possible to mask sensitive information. The proof of the result for the strong case in [7] relies on induction over the *norm* of a process and the fact that the norm of the arguments of a parallel composition is less than the norm of the parallel composition. Unfortunately, this property is not true because of the restriction operator (see Section 4 for a counter example). This is the reason why we restrict ourselves to finite processes in the strong setting. The proof of the weak case in [7] follows the proof technique attributed to Milner. The general techniques from [15] cannot be applied directly in the setting of the applied π-calculus due to the active substitutions.

In [14], the second author presented an adaptation of the general result of [15] in order to make it suitable for establishing unique parallel decomposition in settings with a notion of unobservable behaviour. The ensued technique amounts to showing that the transition relation induces a so-called *weak* decomposition order satisfying a property that is called *power cancellation*. In the present paper, we show how, instead of using the adapted technique from [14], the original technique from [15] may be applied in settings with a notion of unobservable behaviour, considering a stutter-free fragment of the transition relation. This method appears to be simpler than the method suggested by the result in [14].

The paper is organized as follows. In Section 2, we briefly recall the abstract framework introduced in [15] to prove UPD results. In Section 3 we recall the syntax and different semantics of the π-calculus. Section 4 is composed of two subsections. In Section 4.1 we introduce the notion of *depth* of a process and we prove some properties of this notion. In Section 4.2 we use these results and the result in Section 2 to prove that finite processes of the π-calculus satisfy unique parallel decomposition w.r.t. strong bisimilarity. Section 5 follows a similar structure. In Section 5.1 we introduce the notion of *processes without stuttering transitions* and we prove some properties of this kind of processes. These properties and the result in Section 2 are used in Section 5.2 to prove that finite processes of the π-calculus satisfy unique parallel decomposition w.r.t. weak bisimilarity. In Section 6 we present some final remarks.

2 Decomposition orders

In this section, we briefly review the theory of unique decomposition for commutative monoids that we shall apply in the remainder of the paper to prove UPD results in the context of the π-calculus.

Definition 1. *A* commutative monoid *is a set M with a distinguished element e and a binary operation on M denoted by $\cdot$ such that for all $x,y,z \in M$:*

- *$x\cdot(y\cdot z) = (x\cdot y)\cdot z$ (associativity);*
- *$x\cdot y = y\cdot x$ (commutativity);*
- *$x\cdot e = e\cdot x = x$ (identity).*

In the remainder of the paper we often suppress the symbol $\cdot$ or use $\|$.

Definition 2. *An element p of a commutative monoid M is called* indecomposable *if $p \neq e$ and $p = xy$ implies $x = e$ or $y = e$.*

Definition 3. *Let M be a commutative monoid. A* decomposition *in M is a finite multi-set $\wr p_1,\ldots,p_k \int$ of indecomposable elements of M such that $p_1\cdot p_2\cdots p_k$ is defined. The element $p_1\cdot p_2\cdots p_k$ in M will be called the* composition associated *with the decomposition $\wr p_1,\ldots,p_k \int$, and, conversely, we say that $\wr p_1,\ldots,p_k \int$ is a* decomposition *of the element $p_1\cdot p_2\cdots p_k$ of M. Decompositions $d = \wr p_1,\ldots,p_k \int$ and $d' = \wr p'_1,\ldots,p'_l \int$ are equivalent in M (notation $d \equiv d'$) if they have the same composition, i.e. $p_1\cdot p_2\cdots p_k = p'_1\cdots p'_l$. A decomposition d in M is* unique *if $d \equiv d'$ implies $d = d'$ for all decompositions d' in M. We say that an element x of M has a* unique decomposition *if it has a decomposition and this decomposition is unique. If every element of M has a unique decomposition, then we say that M has* unique decomposition.

Theorem 1 below gives a sufficient condition to ensure that a commutative monoid M has unique decomposition. It requires the existence of a *decomposition order* for M.

Definition 4. *Let M be a commutative monoid; a partial order $\preccurlyeq$ on M is a* decomposition order *if*

1. *it is* well-founded, *i.e., for every non-empty subset $\hat{M} \subseteq M$ there is $m \in \hat{M}$ such that for all $m' \in M$, $m' \preccurlyeq m$ implies $m' = m$. In this case, we say that m is a $\preccurlyeq$-minimal element of $\hat{M}$;*
2. *the identity element e of M is the least element of M with respect to $\preccurlyeq$, i.e., $e \preccurlyeq x$ for all x in M;*
3. *$\preccurlyeq$ is* strictly compatible, *i.e., for all $x,y,z \in M$ if $x \prec y$ (i.e. $x \preccurlyeq y$ and $x \neq y$) and yz is defined, $xz \prec yz$;*
4. *it is* precompositional, *i.e., for all $x,y,z \in M$ $x \preccurlyeq yz$ implies $x = y'z'$ for some $y' \preccurlyeq y$ and $z' \preccurlyeq z$; and*
5. *it is* Archimedean, *i.e., for all $x,y \in M$ $x^n \preccurlyeq y$ for all $n \in \mathbb{N}_0$ implies that $x = e$.*

Theorem 1 ([15]). *Every commutative monoid M with a decomposition order has unique decomposition.*

3 The π-calculus

We recall the syntax of the π-calculus and the rules to define the transition relation [20]. We assume a set of *names* or *channels* $\mathcal{V}$. We use a,b,c,x,y,z to range over $\mathcal{V}$.

$$\frac{}{\overline{x}y.P \xrightarrow{\overline{x}y} P}\ \text{(Out)} \qquad \frac{}{x(z).P \xrightarrow{xy} P\{y/z\}}\ \text{(Inp)} \qquad \frac{}{\tau.P \xrightarrow{\tau} P}\ \text{(Tau)} \qquad \frac{\pi.P \xrightarrow{\alpha} P}{[x=x]\pi.P \xrightarrow{\alpha} P}\ \text{(Mat)}$$

$$\frac{P \xrightarrow{\alpha} P'}{P+Q \xrightarrow{\alpha} P'}\ \text{(Sum-L)} \qquad \frac{P \xrightarrow{\alpha} P'}{P \mid Q \xrightarrow{\alpha} P' \mid Q}\ bn(\alpha)\cap fn(Q)=\varnothing \quad \text{(Par-L)}$$

$$\frac{P \xrightarrow{\overline{x}y} P' \quad Q \xrightarrow{xy} Q'}{P \mid Q \xrightarrow{\tau} P' \mid Q'}\ \text{(Comm-L)} \qquad \frac{P \xrightarrow{\overline{x}(z)} P' \quad Q \xrightarrow{xz} Q'}{P \mid Q \xrightarrow{\tau} \nu z(P' \mid Q')}\ z \notin fn(Q) \quad \text{(Close-L)}$$

$$\frac{P \xrightarrow{\alpha} P'}{\nu z(P) \xrightarrow{\alpha} \nu z(P')}\ z \notin n(\alpha) \quad \text{(Res)} \qquad \frac{P \xrightarrow{\overline{x}z} P'}{\nu z(P) \xrightarrow{\overline{x}(z)} P'}\ z \neq x \quad \text{(Open)} \qquad \frac{P \xrightarrow{\alpha} P'}{!P \xrightarrow{\alpha} P' \mid !P}\ \text{(Rep-Act)}$$

$$\frac{P \xrightarrow{\overline{x}y} P' \quad P \xrightarrow{xy} P''}{!P \xrightarrow{\tau} (P' \mid P'') \mid !P}\ \text{(Rep-Comm)} \qquad \frac{P \xrightarrow{\overline{x}(z)} P' \quad P \xrightarrow{xz} P''}{!P \xrightarrow{\tau} (\nu z(P' \mid P'')) \mid !P}\ z \notin fn(P) \quad \text{(Rep-Close-L)}$$

Table 1: Transition rules for the π-calculus

Definition 5. *The* processes, summations *and* prefixes *of the π-calculus are given respectively by*

$$\begin{array}{rcl} P & ::= & M \quad \mid \quad P \mid P' \quad \mid \quad \nu z.P \quad \mid \quad !P \\ M & ::= & \mathbf{0} \quad \mid \quad \pi.P \quad \mid \quad M+M' \\ \pi & ::= & \overline{x}y \quad \mid \quad x(z) \quad \mid \quad \tau \quad \mid \quad [x=y]\pi \end{array}$$

We denote with Π the set of processes of the π-calculus.

An occurrence of a name $z \in \mathcal{V}$ is *bound* in a process P if it is in the scope of a restriction νz or of an input $a(z)$. A name $a \in \mathcal{V}$ is *free* in a process P if there is at least one occurrence of a that is not bound. We write $bn(P)$ and $fn(P)$ to denote, respectively, the set of bound names and free names of a process P. The *set of names* of a process P is defined by $n(P) = bn(P) \cup fn(P)$. We employ the following convention of the π-calculus w.r.t. names.

Convention 1. *[20, P.47] In any discussion, we assume that the bound names of any processes or actions under consideration are chosen to be different from the names free in any other entities under consideration, such as processes, actions, substitutions, and sets of names. This convention is subject to the limitation that in considering a transition $P \xrightarrow{\overline{x}(z)} Q$, the name z that is bound in $\overline{x}(z)$ and in P may occur free in Q. This limitation is necessary for expressing scope extrusion.*

The transition relation associated to a term is defined by the rules in Table 1, where we have omitted the symmetric version of the rules (Sum-L), (Par-L), (Comm-L) and (Close-L). We denote with A the set of *visible actions* that can be executed by a process $P \in \Pi$, i.e. $A = \{xy \mid x,y \in \mathcal{V}\} \cup \{\overline{x}y \mid x,y \in \mathcal{V}\} \cup \{\overline{x}(z) \mid x,z \in \mathcal{V}\}$. The action τ is the *internal action*. We define $A_\tau = A \cup \{\tau\}$. For $P,P' \in \Pi$ and $\alpha \in A_\tau$, we write $P \xrightarrow{\alpha} P'$ if there is a derivation of $P \xrightarrow{\alpha} P'$ with rules in Table 1.

Definition 6. Strong bisimilarity *is the largest symmetric relation over Π, notation $\sim$, such that whenever $P \sim Q$, if $P \xrightarrow{\alpha} P'$ then there is Q' s.t. $Q \xrightarrow{\alpha} Q'$ and $P' \sim Q'$.*

The relation $\sim$ is not compatible with input prefix: We have that

$$\bar{z}x \mid a(y) \sim \bar{z}x.a(y) + a(y).\bar{z}x \ ,$$

whereas

$$b(a).(\bar{z}x \mid a(y)) \not\sim b(a).(\bar{z}x.a(y) + a(y).\bar{z}),$$

because when z is received over the channel a, we have

$$(\bar{z}x \mid a(y))\{a/z\} \not\sim (\bar{z}x.a(y) + a(y).\bar{z})\{a/z\} \ .$$

Hence, $\sim$ is not a congruence for the full syntax of the π-calculus. It is, however, a so-called *non-input congruence* (see [20, Theorem 2.2.8]): it is compatible with all the other constructs in the syntax. In the present paper we shall only use the fact that $\sim$ is compatible with parallel composition, i.e., if $P_1 \sim Q_1$ and $P_2 \sim Q_2$, then $P_1 \mid P_2 \sim Q_1 \mid Q_2$.

We recall now the weak variant of bisimilarity. We write $P \Longrightarrow P'$ if $P = P'$ or if there are $P_0,\ldots,P_n$ with $n > 0$ s.t. $P = P_0 \xrightarrow{\tau} \ldots \xrightarrow{\tau} P_n = P'$. We write $P \overset{\alpha}{\Longrightarrow} Q$ with $\alpha \in A_\tau$ if there are P',Q' s.t. $P \Longrightarrow P' \xrightarrow{\alpha} Q' \Longrightarrow Q$. Notice the difference between $P \Longrightarrow P'$ and $P \overset{\tau}{\Longrightarrow} P'$, in the second case, at least one τ-transition is executed.

Definition 7. Weak bisimilarity *is the largest symmetric relation over* Π*, notation* $\approx$*, such that whenever* $P \approx Q$*, (i) if* $P \xrightarrow{\alpha} P'$ *with* $\alpha \in A$ *then there is* Q' *s.t.* $Q \overset{\alpha}{\Longrightarrow} Q'$ *and* $P' \approx Q'$ *and (ii) if* $P \xrightarrow{\tau} P'$ *then there is* Q' *s.t.* $Q \Longrightarrow Q'$ *and* $P' \approx Q'$*.*

Like strong bisimilarity, it is only possible to prove that $\approx$ is a congruence for non-input contexts (see [20, Theorem 2.4.22]).

4 Unique decomposition with respect to strong bisimilarity

In this section, we shall use the result presented in Section 2 to prove that every *finite* π-calculus process has a unique parallel decomposition w.r.t. strong bisimilarity. In Section 4.1 we introduce the definition of depth of a process and some of its properties. We also explain why we restrict our development to finite processes. In Section 4.2 we present the unique decomposition result.

4.1 The depth of a process

Given a set X, we denote with X^* the set of finite sequences over X, where $\varepsilon \in X^*$ is the empty sequence. For $\omega = \alpha_1\alpha_2\cdots\alpha_n \in A_\tau^*$ with $n > 0$, we write $P \xrightarrow{\omega} P'$ if there are processes $P_0,P_1,\ldots,P_n$ s.t. $P = P_0 \xrightarrow{\alpha_1} P_1 \xrightarrow{\alpha_2} \ldots \xrightarrow{\alpha_n} P_n = P'$. If $\omega = \varepsilon$, then $P \xrightarrow{\omega} P'$ implies $P' = P$. If we are not interested in P', we write $P \xrightarrow{\omega}$. In addition, we write $P\downarrow$ if for all $\alpha \in A_\tau$, $P \not\xrightarrow{\alpha}$.

Definition 8. *Let length* $: A_\tau^* \to \mathbb{N}_0$ *be the function defined by*

$$\mathit{length}(\omega) = \begin{cases} 0 & \text{if } \omega = \varepsilon, \\ \mathit{length}(\omega') + 1 & \text{if } \omega = \alpha\omega' \text{ and } \alpha \neq \tau \\ \mathit{length}(\omega') + 2 & \text{if } \omega = \alpha\omega' \text{ and } \alpha = \tau \end{cases}$$

Definition 9. *A process $P \in \Pi$ is* normed *if there is $\omega \in A_\tau^*$ such that $P \xrightarrow{\omega} P' \downarrow$. We denote with Π_n the set of normed processes. The depth* $: \Pi_n \to \mathbb{N}_0 \cup \{\infty\}$ *and the norm* $: \Pi_n \to \mathbb{N}_0$ *of a normed process $P \in \Pi$ are defined, respectively, by*

$$depth(P) = \sup(\{length(\omega) \mid P \xrightarrow{\omega} P' \text{ and } P' \downarrow\})$$
$$norm(P) = \inf(\{length(\omega) \mid P \xrightarrow{\omega} P' \text{ and } P' \downarrow\})$$

Where $\sup(X) = \infty$ *whenever X is an infinite set, and* $\inf(\varnothing) = \infty$.

We remark that we have assigned a higher weight to occurrences of the label τ in the definition of the length of a sequence $\omega \in A_\tau^*$. This is to ensure that depth is additive w.r.t. parallel composition (i.e., the depth of a parallel composition is the sum of the depths of its components), as we shall prove in Lemma 6) below. As opposed to other process calculi for which unique decomposition has been established (see, e.g. [15]), due to scope extrusion, norm is not additive for the π-calculus: Consider

$$P = P_0 \mid P_1 = \nu z(\overline{a}z) \mid a(x).!\overline{x}a$$

P is normed because $P \xrightarrow{\tau} \nu z(\mathbf{0} \mid !\overline{z}a) \downarrow$ but P_1 is not because it only performs an execution of infinite length. Then, to ensure this kind of properties, one approach could be just consider normed processes. Unfortunately this is not enough. Consider

$$Q = Q_0 \mid Q_1 = \nu z(\overline{a}z) \mid a(x).\overline{x}a$$

Processes Q, Q_0 and Q_1 are normed and, moreover, they perform no infinite execution. Despite this, we have that $norm(Q) = 2$, because $Q \xrightarrow{\tau} \nu z(\mathbf{0} \mid \overline{z}a) \downarrow$, and $norm(Q_0) + norm(Q_1) = 1 + 2 = 3$. Moreover, notice that the norm of the arguments of a parallel composition is not less than the norm of the parallel composition, i.e. $norm(Q) = 2$ and $norm(Q_1) = 2$. In particular, these examples show that item 4 in Lemma 3 of [7] (norm is additive) is false, and, as a consequence, some proofs in [7] are flawed. The authors of [7] proposed a solution to this problem that we discuss in the conclusion of this paper. So, to facilitate inductive reasoning, we will consider *finite processes* and depth.

Definition 10. *A process $P \in \Pi$ is* finite *if there is $n \in \mathbb{N}_0$ s.t. there is no $\omega = \alpha_1\alpha_2\cdots\alpha_{n+1} \in A_\tau^*$ such that $P \xrightarrow{\omega}$. We denote with Π_f the set of finite processes of Π.*

Following the last example, we have that $Q, Q_0, Q_1 \in \Pi_f$ and $depth(Q) = 3 = depth(Q_0) + depth(Q_1)$.

To conclude this section we present a collection of results including lemmas and theorems. Most of the lemmas are only needed to prove the theorems. The theorems and only few lemmas will be used in the next section. Theorem 2 states that bisimilar processes have the same depth. Theorem 3 states that the depth of a parallel composition of two processes not bisimilar to $\mathbf{0}$ is greater than the depth of each process. Thanks to these results, we will able to extend the notion of depth to equivalence classes and apply inductive reasoning.

Lemma 1. *For all $P \in \Pi_f$, $P \not\sim \mathbf{0}$ implies $depth(P) > 0$.*

Lemma 2. *If $P \in \Pi_f$ and $P \xrightarrow{\alpha} P'$, $\alpha \in A_\tau$, then $P' \in \Pi_f$ and $depth(P) > depth(P')$.*

Theorem 2. *If $P \sim Q$ then $P \in \Pi_f$ iff $Q \in \Pi_f$; moreover, $depth(P) = depth(Q)$.*

Proof. Suppose that $P \sim Q$. Then, clearly, $P \xrightarrow{\omega}$ iff $Q \xrightarrow{\omega}$, and hence $P \in \Pi_f$ iff $Q \in \Pi_f$.

To prove that $\mathit{depth}(P) = \mathit{depth}(Q)$, first note that if $P,Q \notin \Pi_f$, then $\mathit{depth}(P) = \infty = \mathit{depth}(Q)$. In the case that remains, both $\mathit{depth}(P)$ and $\mathit{depth}(Q)$ are natural numbers; we proceed by induction over $\mathit{depth}(P)$. If $\mathit{depth}(P) = 0$ then $P \sim \mathbf{0}$ by Lemma 1 , so $Q \sim \mathbf{0}$ and therefore $\mathit{depth}(Q) = 0$. Suppose now $\mathit{depth}(P) = n > 0$. Assume that the statement holds for processes with depth less than n. Suppose $\mathit{depth}(Q) = m > n$ then there is Q' s.t. $Q \xrightarrow{\alpha} Q'$ and $m = \mathit{length}(\alpha) + \mathit{depth}(Q')$. By definition of ~, we get $P \xrightarrow{\alpha} P'$, $P' \sim Q'$. By Lemma 2, $\mathit{depth}(P') < \mathit{depth}(P)$, therefore $\mathit{depth}(P') = \mathit{depth}(Q')$ and $\mathit{depth}(P) \geq \mathit{depth}(P') + \mathit{length}(\alpha) = m > n = \mathit{depth}(P)$, i.e. we get a contradiction. Similarly, for the case $\mathit{depth}(Q) = m < n$, we can reach a contradiction by considering a transition $P \xrightarrow{\alpha} P'$ with $n = \mathit{length}(\alpha) + \mathit{depth}(P')$. Then we can conclude $\mathit{depth}(P) = \mathit{depth}(Q)$. □

Lemma 3. *Let $P, P' \in \Pi_f$, $\omega = \alpha\omega' \in A_\tau^*$ be such that $P \xrightarrow{\alpha} P' \xrightarrow{\omega'}$ and $\mathit{depth}(P) = \mathit{length}(\omega)$. Then $\mathit{depth}(P') = \mathit{length}(\omega')$ and therefore $\mathit{depth}(P) = \mathit{depth}(P') + \mathit{length}(\alpha)$.*

Lemma 4. *For all $P \in \Pi_f$, $\mathit{depth}(P) \geq \mathit{depth}(\nu z(P))$ for all $z \in \mathcal{V}$.*

Lemma 5. *Let $P, Q \in \Pi_f$ and $\omega \in A_\tau^*$ be such that $P \mid Q \xrightarrow{\omega}$ and $\mathit{length}(\omega) = \mathit{depth}(P \mid Q)$. Then, there are $\omega_1, \omega_2 \in A_\tau^*$ such that $P \xrightarrow{\omega_1}$, $Q \xrightarrow{\omega_2}$ and $\mathit{length}(\omega_1) + \mathit{length}(\omega_2) = \mathit{length}(\omega)$.*

Proof. We proceed by complete induction on $n = \mathit{depth}(P) + \mathit{depth}(Q)$. Suppose that the property holds for parallel compositions of finite processes such that the sum of the depths is smaller than $n > 0$. Let $\omega = \alpha\omega'$ and R be such that $\mathit{length}(\omega) = n$ and $P \mid Q \xrightarrow{\alpha} R \xrightarrow{\omega'}$. We analyse the different ways of deriving the first transition (we omit the symmetric cases).

- Case (Par-L). Then $P \xrightarrow{\alpha} P'$ and $R = P' \mid Q$. By Lemma 3 and induction $\mathit{depth}(P' \mid Q) = \mathit{length}(\omega') < n$ and then there are ω_1 and ω_2 s.t. $P' \xrightarrow{\omega_1}$, $Q \xrightarrow{\omega_2}$ and $\mathit{length}(\omega_1) + \mathit{length}(\omega_2) = \mathit{length}(\omega')$. Then $P \xrightarrow{\alpha\omega_1}$ and $\mathit{length}(\alpha\omega_1) + \mathit{length}(\omega_2) = \mathit{length}(\omega)$.

- Case (Comm-L). Then $P \xrightarrow{\overline{x}y} P'$, $Q \xrightarrow{xy} Q'$ and $R = P' \mid Q'$ and $\alpha = \tau$. By Lemma 3 and induction $\mathit{depth}(P' \mid Q') = \mathit{length}(\omega') < \mathit{length}(\tau) + \mathit{length}(\omega') = n$ and then there are ω_1 and ω_2 s.t. $P' \xrightarrow{\omega_1}$, $Q' \xrightarrow{\omega_2}$ and $\mathit{length}(\omega_1) + \mathit{length}(\omega_2) = \mathit{length}(\omega')$. Then $P \xrightarrow{\overline{x}y\omega_1}$ and $Q \xrightarrow{xy\omega_2}$ and $\mathit{length}(\overline{x}y\omega_1) + \mathit{length}(xy\omega_2) = \mathit{length}(\tau) + \mathit{length}(\omega') = \mathit{length}(\omega)$.

- Case (Close-L). Then $P \xrightarrow{\overline{x}(z)} P', Q \xrightarrow{xz} Q'$, $\alpha = \tau$ and $R = \nu z(P' \mid Q')$. The side condition of (Close-L) allows us to use the rules (Par-L) and its symmetric version, then $P \mid Q \xrightarrow{\overline{x}(z)xz} P' \mid Q'$. On one hand, by Lemma 4, $\mathit{depth}(P' \mid Q') \geq \mathit{depth}(\nu z(P' \mid Q'))$. On the other hand, $\mathit{depth}(P' \mid Q') \leq \mathit{depth}(\nu z(P' \mid Q'))$ because $\omega = \tau\omega'$ is a maximal execution, $\mathit{length}(\tau) = \mathit{length}(\overline{x}zxz)$, $\nu z(P' \mid Q') \xrightarrow{\omega'}$ and by Lemma 3. Then $\mathit{depth}(P' \mid Q') = \mathit{depth}(\nu z(P' \mid Q')) < n$. Moreover, there is ω'' such that $P' \mid Q' \xrightarrow{\omega''}$ and $\mathit{length}(\overline{x}zxz\omega'') = \mathit{depth}(P \mid Q)$. Then $P \mid Q \xrightarrow{\overline{x}(z)} P' \mid Q \xrightarrow{xz\omega''}$ with $\mathit{length}(xz\omega'') < n$. From this point we can repeat the proof of the first case.

□

Lemma 6. *For all processes $P, Q \in \Pi_f$, $\mathit{depth}(P \mid Q) = \mathit{depth}(P) + \mathit{depth}(Q)$.*

Proof. By Lemma 5 we can ensure $\mathit{depth}(P \mid Q) \leq \mathit{depth}(P) + \mathit{depth}(Q)$. On the other hand, by Convention 1, we have that $P \xrightarrow{\omega}$ or $Q \xrightarrow{\omega}$ implies $P \mid Q \xrightarrow{\omega}$. This allows us to conclude $\mathit{depth}(P \mid Q) \geq \mathit{depth}(P) + \mathit{depth}(Q)$ and therefore $\mathit{depth}(P \mid Q) = \mathit{depth}(P) + \mathit{depth}(Q)$. □

Lemma 7. *For all $P,Q \in \Pi$, $P,Q \in \Pi_f$ iff $P \mid Q \in \Pi_f$.*

Theorem 3. *If $P,Q,R \in \Pi_f$, $P \not\rightarrow \mathbf{0}$, $Q \not\rightarrow \mathbf{0}$ and $P \mid Q \sim R$ then $depth(P) < depth(R)$ and $depth(Q) < depth(R)$.*

Proof. By Lemma 1, $depth(P) > 0$, $depth(Q) > 0$. By Theorem 2, $depth(R) = depth(P \mid Q)$. By Lemma 6, $depth(R) = depth(P) + depth(Q)$ and we conclude $depth(P) < depth(R)$ and $depth(Q) < depth(R)$. □

4.2 Unique decomposition

The commutative monoid associated with Π_f modulo $\sim$ is defined by

- $\mathbf{P}_\sim = \{[P]_\sim : P \in \Pi_f\}$ where $[P]_\sim = \{P' : P' \sim P\}$
- $e = [\mathbf{0}]_\sim \in \mathbf{P}_\sim$.
- $\parallel : \mathbf{P}_\sim \times \mathbf{P}_\sim \rightarrow \mathbf{P}_\sim$ is such that $[P]_\sim \parallel [Q]_\sim = [P \mid Q]_\sim$

By Lemma 7 we have that the definition of $\parallel$ is sound. By Theorem 2 we have that all $P' \in [P]_\sim$ have the same depth. Then we can lift function *depth* to $\mathbf{P}_\sim$, i.e. $depth([P]_\sim) = depth(P)$.

Lemma 8. $\mathbf{P}_\sim$ *with neutral element $[\mathbf{0}]_\sim$ and binary operation $\parallel$ is a commutative monoid. I.e., $\parallel \subseteq \mathbf{P}_\sim \times \mathbf{P}_\sim$ satisfies the associativity, commutativity and identity properties.*

In order to use the Theorem 1 we need to define on $\mathbf{P}_\sim$ a decomposition order. In [15, 14], it is shown that the transition relation directly induces a decomposition order on a commutative monoid of processes. In the case of the π-calculus, however, the order induced by the transition relation cannot be directly used, as is illustrated by the following example: Define a binary relation $\rightsquigarrow \subseteq \mathbf{P}_\sim \times \mathbf{P}_\sim$ by $[R]_\sim \rightsquigarrow [S]_\sim$ if there is $R' \in [R]_\sim$ and $S' \in [S]_\sim$ such that $R' \xrightarrow{\alpha} S'$. We denote the inverse of the reflexive-transitive closure of $\rightsquigarrow$ by $\preccurlyeq_\rightsquigarrow$, i.e., $\preccurlyeq_\rightsquigarrow = (\rightsquigarrow^*)^{-1}$. The order $\preccurlyeq_\rightsquigarrow$ is not precompositional. Consider the processes $P = \nu z.(\overline{a}z.\overline{z}c.\overline{c}a)$ and $Q = a(x).x(y).\overline{y}b$. Then

- $P \mid Q = \nu z.(\overline{a}z.\overline{z}c.\overline{c}a) \mid a(x).x(y).\overline{y}b \xrightarrow{\tau} \nu z.(\overline{z}c.\overline{c}a \mid z(y).\overline{y}b) = R$ and therefore
- $[P]_\sim \parallel [Q]_\sim = [P \parallel Q]_\sim \rightsquigarrow [R]_\sim = [\nu z.(\overline{z}c.\overline{c}a \mid z(y).\overline{y}b)]_\sim$.
- Note that R executes only one transition, i.e. $\nu z.(\overline{z}c.\overline{c}a \mid z(y).\overline{y}b) \xrightarrow{\tau} \nu z(\overline{c}a \mid \overline{c}b)$, then it is clear that there are no processes P' and Q' s.t.

$$[\nu z.(\overline{a}z.\overline{z}c.\overline{c}a)]_\sim \rightsquigarrow^* [P']_\sim \qquad [a(x).x(y).\overline{y}b]_\sim \rightsquigarrow^* [Q']_\sim \qquad [P']_\sim \parallel [Q']_\sim = [\nu z(\overline{z}c.\overline{c}a \mid z(y).\overline{y}b)]_\sim$$

The particularity of this example is the *scope extrusion*. We need to define an order based on a fragment of the transition relation that avoids this phenomenon. We shall define the partial order $\preccurlyeq$ over $\mathbf{P}_\sim$ as the reflexive-transitive closure of the relation $\rightarrow \subseteq \mathbf{P}_\sim \times \mathbf{P}_\sim$, which is, in turn, defined as follows:

$$\rightarrow_0 = \{([P]_\sim, [Q]_\sim) : P \xrightarrow{\alpha} Q, \alpha \in A_\tau \text{ and } \nexists P_0, P_1 \in \Pi_f \text{ s.t. } P_0 \not\rightarrow \mathbf{0}, P_1 \not\rightarrow \mathbf{0}, P_0 \mid P_1 \sim P\}$$

$$\begin{aligned}\rightarrow_{k+1} &= \{([P_0 \mid P_1]_\sim, [Q_0 \mid P_1]_\sim) : [P_0]_\sim \rightarrow_k [Q_0]_\sim, P_1 \in \Pi_f\}\\ &\cup \{([P_0 \mid P_1]_\sim, [P_0 \mid Q_1]_\sim) : [P_1]_\sim \rightarrow_k [Q_1]_\sim, P_0 \in \Pi_f\}\end{aligned}$$

$$\rightarrow = \bigcup_{k=0}^{\infty} \rightarrow_n \quad .$$

The partial order $\preccurlyeq$ is defined as the inverse of the reflexive-transitive closure of $\rightarrow$ i.e., $\preccurlyeq = (\rightarrow^*)^{-1}$. We write $[P]_\sim \prec [Q]_\sim$ if $[P]_\sim \preccurlyeq [Q]_\sim$ and $[P]_\sim \neq [Q]_\sim$. Notice that the definition of $\rightarrow$ avoids any kind of communications between the arguments of the parallel operator, this ensures that the scope extrusion is also avoided.

Lemma 9. *If* $[P]_\sim \to [Q]_\sim$ *then* $depth([Q]_\sim) < depth([P]_\sim)$.

Lemma 10. $\preccurlyeq$ *is a partial order.*

Proof. We have to prove that $\preccurlyeq$ is reflexive, antisymmetric, and transitive. $\preccurlyeq$ is reflexive and transitive because it is the reflexive-transitive closure of $\to$. To prove that $\preccurlyeq$ is antisymmetric notice that $[P]_\sim \prec [Q]_\sim$ implies $[Q]_\sim = [P_n]_\sim \to \ldots \to [P_1]_\sim \to [P_0]_\sim = [P]_\sim$ for $n > 0$ and then, by Lemma 9, $depth([P]_\sim) < depth([Q]_\sim)$. Therefore $[P]_\sim \preccurlyeq [Q]_\sim$ and $[Q]_\sim \preccurlyeq [P]_\sim$ implies $[P]_\sim = [Q]_\sim$. □

In Lemma 12, we prove that $\preccurlyeq$ is a decomposition order. To prove this result we need to add a last auxiliary result, Lemma 11.

Lemma 11. *If* $P \in \Pi_f$ *and* $depth(P) > 0$ *then there is* Q *s.t.* $[P]_\sim \to [Q]_\sim$.

Proof. We proceed by complete induction over $n = depth(P)$. Assume that the hypothesis holds for values less than $n \geq 1$. Suppose there are no $P_0, P_1 \in \Pi_f$ such that $P \sim P_0 \mid P_1$, $P_0 \not\sim \mathbf{0}$ and $P_1 \not\sim \mathbf{0}$. Given that $n \geq 1$ then there is $\alpha \in A_\tau$ s.t. $P \xrightarrow{\alpha} P'$. Then $[P]_\sim \to_0 [P']_\sim$ and therefore $[P]_\sim \to [P']_\sim$. Finally, we can define $[Q]_\sim = [P']_\sim$.

Suppose there are $P_0, P_1 \in \Pi_f$ such that $P \sim P_0 \mid P_1$, $P_0 \not\sim \mathbf{0}$ and $P_1 \not\sim \mathbf{0}$, then $[P]_\sim = [P_0 \mid P_1]_\sim$. By Theorem 3 $depth(P_0) < depth(P)$ and by Lemma 1 $depth(P_0) > 0$. By induction there is Q_0 s.t. $[P_0]_\sim \to [Q_0]_\sim$ and therefore there is k s.t. $[P_0]_\sim \to_k [Q_0]_\sim$ By definitions of $\to_{k+1}$ and $\to$, $[P_0 \mid P_1]_\sim \to_{k+1} [Q_0 \mid P_1]_\sim$ and $[P_0 \mid P_1]_\sim \to [Q_0 \mid P_1]_\sim$. Therefore if we define $[Q]_\sim = [Q_0 \parallel P_1]_\sim$ the proof is complete. □

Lemma 12. $\preccurlyeq \subseteq \mathbf{P}_\sim \times \mathbf{P}_\sim$ *is a decomposition order.*

Proof.

1. $\preccurlyeq$ is well-founded. We have to prove that every non-empty subset of $\mathbf{P}_\sim$ has a $\preccurlyeq$-minimal element. Let $X \subseteq \mathbf{P}_\sim$ with $X \neq \varnothing$. Let $[P]_\sim$ be s.t. $[P]_\sim \in X$ and $depth([P]_\sim) = \min\{depth([Q]_\sim) \mid [Q]_\sim \in X\}$, then $[P]_\sim$ is a minimal element of X by Lemma 9 and definition of $\preccurlyeq$.

2. $[\mathbf{0}]_\sim$ is the least element of $\mathbf{P}_\sim$ w.r.t. $\preccurlyeq$. We consider $[P]_\sim$ and proceed by induction on $depth(P)$. If $depth(P) = 0$, then $P \sim \mathbf{0}$ and therefore $[P]_\sim = [\mathbf{0}]_\sim$. Suppose that $depth(P) = n > 0$. By Lemma 11 there is Q s.t. $[P]_\sim \to [Q]_\sim$. By Lemma 9, $depth(Q) < depth(P)$. By induction and definition of $\preccurlyeq$, $[\mathbf{0}]_\sim \preccurlyeq [Q]_\sim \preccurlyeq [P]_\sim$.

3. $\preccurlyeq$ is strictly compatible. Suppose $[Q]_\sim \prec [P]_\sim$ and consider $[P]_\sim \parallel [S]_\sim$. By definition of $\prec$ there are $P_0, \ldots, P_n \in \Pi_f$, with $n > 0$, s.t. $[P]_\sim = [P_0]_\sim \to [P_1]_\sim \to \ldots \to [P_n]_\sim = [Q]_\sim$. By definition of $\to$, for each $i = 0, \ldots n-1$ there is k_i s.t. $[P_i]_\sim \to_{k_i} [Q_i]_\sim$. Define $k = \max\{k_i : i = 0, \ldots n-1\}$. Then

$$[P]_\sim \parallel [S]_\sim = [P_0]_\sim \parallel [S]_\sim = [P_0 \mid S]_\sim \to_{k+1} \cdots \to_{k+1} [P_n \mid S]_\sim = [P_n]_\sim \parallel [S]_\sim = [Q]_\sim \parallel [S]_\sim$$

By definition of $\to$,

$$[P]_\sim \parallel [S]_\sim = [P_0]_\sim \parallel [S]_\sim = [P_0 \mid S]_\sim \to \ldots \to [P_n \mid S]_\sim = [P_n]_\sim \parallel [S]_\sim = [Q]_\sim \parallel [S]_\sim$$

By Lemma 9 and $n > 0$, $depth([P]_\sim \parallel [S]_\sim) > depth([Q]_\sim \parallel [S]_\sim)$. Then $[Q]_\sim \parallel [S]_\sim \prec [P]_\sim \parallel [S]_\sim$.

4. $\preccurlyeq$ is precompositional. Suppose $[P]_\sim \preccurlyeq [Q]_\sim \parallel [R]_\sim$, we have to prove there are $[Q']_\sim \preccurlyeq [Q]_\sim$ and $[R']_\sim \preccurlyeq [R]_\sim$ s.t. $[P]_\sim = [Q']_\sim \parallel [R']_\sim$. If $Q \sim R \sim \mathbf{0}$ then $Q' \sim R' \sim \mathbf{0}$ and the conditions are satisfied. Suppose that only one of both processes is bisimilar to $\mathbf{0}$. W.l.o.g. suppose $Q \not\sim \mathbf{0}$ and $R \sim \mathbf{0}$. In this case, $[P]_\sim \preccurlyeq [Q]_\sim \parallel [R]_\sim = [Q]_\sim$, then if we define $[Q']_\sim = [P]_\sim$ and $[R']_\sim = [\mathbf{0}]_\sim$, the conditions are also satisfied. Suppose now that $Q \not\sim \mathbf{0}$ and $R \not\sim \mathbf{0}$. By definition of $\preccurlyeq$ there are $n \geq 0$ and

processes $S_n,\ldots,S_0$ s.t. $[Q]_\sim \parallel [R]_\sim = [Q \mid R]_\sim = [S_n]_\sim \to \ldots \to [S_0]_\sim = [P]_\sim$. The proof proceed by induction on n. Suppose that the hypothesis holds for n, we prove the case $n+1$. Given that $[S_{n+1}]_\sim = [Q \mid R]_\sim = [Q]_\sim \parallel [R]_\sim \to [S_n]_\sim$, by definition of $\to$, there is T s.t. either $[Q]_\sim \to [T]_\sim$ and $[S_n]_\sim = [T \mid R]_\sim$ or, $[R]_\sim \to [T]_\sim$ and $[S_n]_\sim = [Q \mid T]_\sim$. (We have omitted the sub-index of $\to$ because it does not play any role.) W.l.o.g. suppose that $[Q]_\sim \to [T]_\sim$ and $[S_n]_\sim = [T \mid R]_\sim$. Then $[P]_\sim \preccurlyeq [T \mid R]_\sim = [T]_\sim \parallel [R]_\sim$. By induction there are $[T']_\sim$ and $[R']_\sim$ s.t. $[T']_\sim \preccurlyeq [T]_\sim$, $[R']_\sim \preccurlyeq [R]_\sim$ and $[P]_\sim = [T']_\sim \parallel [R']_\sim$. Because $[T]_\sim \preccurlyeq [Q]_\sim$ and $\preccurlyeq$ is a partial order, we have that $[T']_\sim \preccurlyeq [Q]_\sim$ and we can conclude the proof.

5. $\preccurlyeq$ is Archimedean. Suppose that $[P]_\sim, [Q]_\sim \in \mathbf{P}_\sim$ are s.t. $[P]^n_\sim \preccurlyeq [Q]_\sim$ for all $n \in \mathbb{N}_0$. By Lemma 6, $depth(P^n) = n \cdot depth(P)$. Given that $depth(Q) \in \mathbb{N}_0$ we can conclude that $depth(P) = 0$ and therefore $[P]_\sim = [\mathbf{0}]_\sim$.

□

By Theorem 1, it follows that $\mathbf{P}_\sim$ has unique decomposition.

Corollary 1. *The commutative monoid $\mathbf{P}_\sim$ has unique decomposition.*

5 Unique parallel decomposition with respect to weak bisimilarity

To prove the result of unique parallel decomposition w.r.t. strong bisimilarity, we relied on the definition of depth and on the properties that are satisfied when we take into account strong bisimilarity. In particular, we proved that all strongly bisimilar processes have the same depth. For the weak bisimilarity we do not have the same property. Consider the following processes

$$P = \overline{x}y.\mathbf{0} \qquad P' = \tau.\overline{x}y.\mathbf{0} \qquad P'' = \tau.\tau.\overline{x}y.\mathbf{0}$$

Notice that $P \approx P' \approx P''$, despite this, $depth(P) < depth(P') < depth(P'')$. To avoid this problem and to adapt the ideas behind results in the previous section, we will consider processes without *stuttering transitions*. A transition $P \xrightarrow{\alpha} P'$ is a stuttering transition if $\alpha = \tau$ and $P \approx P'$.

We could not establish UPD for normed processes in the strong setting, because the norm of the arguments of a parallel composition is not necessarily less than the norm of the parallel composition. In the weak setting, it is known that normed processes of Π do not satisfy UPD w.r.t. bisimilarity. Consider the following counter example [7]: define $P = \nu z(\overline{z}c \mid z(x).!\overline{a}b \mid z(y))$. P is normed because $P \xrightarrow{\tau} \nu z(\mathbf{0} \mid z(x).!\overline{a}b \mid \mathbf{0}) \downarrow$ but there is no a unique parallel decomposition of P because $P \approx P \mid P$.

We study processes without stuttering transitions in Section 5.1. Using the results developed in that section and Theorem 1, in Section 5.2 we prove that for finite processes there is a unique parallel decomposition w.r.t. weak bisimilarity.

5.1 Processes without stuttering steps

For $\omega = a_1 a_2 \cdots a_n \in A^*_\tau$ with $n > 0$, we write $P \overset{\omega}{\Longrightarrow} P'$ if there are processes $P_0, P_1, \ldots, P_n$ s.t. $P = P_0 \overset{a_1}{\Longrightarrow} P_1 \overset{a_2}{\Longrightarrow} \ldots \overset{a_n}{\Longrightarrow} P_n = P'$. If $\omega = \varepsilon$, then $P \overset{\omega}{\Longrightarrow} P'$ implies $P \Longrightarrow P'$.

Definition 11. *A process $P \in \Pi_f$ is a process* without stuttering transitions *if there are no $\omega \in A^*$ and $P', P'' \in \Pi_f$ s.t. $P \overset{\omega}{\Longrightarrow} P' \xrightarrow{\tau} P''$ and $P' \approx P''$. We denote with $\Pi_{\not\circlearrowright}$ the set of processes of Π_f without stuttering transitions.*

In Section 4.1 we discussed why we do not consider infinite processes, this discussion also applies for weak bisimilarity. By definition, $\Pi_{\not\circlearrowleft} \subseteq \Pi_f$. This fact and Lemma 14 ensure that we can use processes in $\Pi_{\not\circlearrowleft}$ to define properties over equivalence classes of processes in Π_f w.r.t. weak bisimilarity.

To prove Lemma 14 we need to introduce some notation and Lemma 13. We write $\overline{x}(z).P$ to denote $\nu z.\overline{x}z.P$. We call $\overline{x}(z)$ a *bound-output prefix*. We use λ to range over prefixes, including bound-outputs.

Lemma 13. *For all $P \in \Pi_f$ there are prefixes $\lambda_1,\dots,\lambda_n$ and processes $P_1,\dots,P_n$ such that $P \sim \sum_{i=1}^{n} \lambda_i.P_i$.*

Proof. The proof proceeds by structural induction on P. Cases $P = \mathbf{0}$, $P = \pi.P'$ are straightforward by definition. For the case $P = P_1 + P_2$, by induction hypothesis there are processes $Q_1 = \sum_{i\in I} \lambda_i.P_i$ and $Q_2 = \sum_{j\in J} \lambda_j.P_j$ and bisimulations R_1 and R_2 s.t. $P_k\ R_k\ Q_k$ for $k = 1,2$. It is easy to see that $\{(P, Q_1 + Q_2)\} \cup R_1 \cup R_2$ is a bisimulation. Case $P = P_1 \mid P_2$ is straightforward by induction and the Expansion Lemma for $\sim$ in the π-calculus [20, Lemma 2.2.14]. Thanks this lemma we can state that for all $P = \sum_{i\in I} \lambda_i P_i$ and $Q = \sum_{j\in J} \lambda_j Q_j$ there is $R = \sum_{k\in K} \lambda_k R_k$ s.t. $P \mid Q \sim R$. Case $P = \nu z.P'$ proceeds by structural induction on P'. If $P' = \mathbf{0}$ then $\nu z.P' \approx \mathbf{0} = \sum_{i=1}^{0} \lambda_i.P_i$. If $P' = \pi.P''$ then there are three cases to analyse: (i) $z \notin n(\pi)$ then $P \sim \pi.(\nu z.P'')$, (ii) $\pi = \overline{x}z$ then $\nu z.P'$ can be denoted by $\overline{x}(z)P''$, (iii) $z \in n(\pi)$ and $\pi \neq \overline{x}z$ then $\nu z.\pi.P'' \sim \mathbf{0}$. If $P' = P_0 + P_1$ then notice that $\nu z(P_0 + P_1) \sim \nu z.P_0 + \nu z.P_1$; by induction there are processes $\sum_{i\in I} \lambda_i P_i$ and $\sum_{j\in J} \lambda_j P_j$ s.t. $\nu z.P_0 \sim \sum_{i\in I} \lambda_i P_i$ and $\nu z.P_1 \sim \sum_{j\in J} \lambda_j P_j$. From this point, the proof follows as in the case $P = P_1 + P_2$. Finally, case $P' = P_0 \mid P_1$ can be reduced to the previous case using the Expansion Lemma for $\sim$ ([20, Lemma 2.2.14]). □

Lemma 14. *For every process $P \in \Pi_f$ there is $Q \in \Pi_{\not\circlearrowleft}$ s.t. $P \approx Q$.*

Proof. The proof of the result follows by complete induction on $n = depth(P)$. By Lemma 13 for $P \in \Pi_f$ there are prefixes $\lambda_1,\dots,\lambda_n$ and processes $P_1,\dots,P_n$ such that $P \sim \sum_{i=1}^{n} \lambda_i.P_i$. By induction and Lemma 2, for each i there is $Q_i \in \Pi_{\not\circlearrowleft}$ s.t. $P_i \approx Q_i$. Then if we define

$$Q = \textstyle\sum_{i\in\{1,\dots n\} \text{ and } P \not\approx P_i} \lambda_i.Q_i$$

we get Q s.t. $Q \in \Pi_{\not\circlearrowleft}$ and $P \approx Q$. □

We cannot restrict our attention only to processes in $\Pi_{\not\circlearrowleft}$ because the property of not executing stuttering transitions is not preserved by parallel composition. Consider the processes $P_0 = \nu z(\overline{a}z)$ and $P_1 = a(x).(\overline{x}b + \tau.\overline{c}b)$. Both $P_0, P_1 \in \Pi_{\not\circlearrowleft}$ but $P_0 \parallel P_1 \notin \Pi_{\not\circlearrowleft}$ because

$$P_0 \parallel P_1 = \nu z(\overline{a}z) \parallel a(x).(\overline{x}b + \tau.\overline{c}b) \xrightarrow{\tau} \nu z(\mathbf{0} \parallel (\overline{z}b + \tau.\overline{c}b)) \sim \tau.\overline{c}b \approx \overline{c}b$$

If we compare this fact with the strong setting, we can say that it is not possible to prove a lemma similar to Lemma 7 for processes in $\Pi_{\not\circlearrowleft}$.

As in Section 4.1, we conclude with a collection of theorems and lemmas. Theorems 4 and 5 are equivalent, respectively, to Theorems 2 and 3 but w.r.t. processes in $\Pi_{\not\circlearrowleft}$ and weak bisimilarity. Most of the lemmas are needed to prove these results and only a few of them are used in the next section.

Lemma 15. *If $P \in \Pi_{\not\circlearrowleft}$ and $P \overset{\omega}{\Longrightarrow} Q$ for $\omega \in A_\tau^*$ then $Q \in \Pi_{\not\circlearrowleft}$.*

Proof. Suppose $Q \notin \Pi_{\not\circlearrowleft}$, then there are $\omega' \in A^*$ and $Q', Q'' \in \Pi_f$ s.t. $Q \overset{\omega'}{\Longrightarrow} Q' \xrightarrow{\tau} Q''$ with $Q' \approx Q''$. Let $\tilde{\omega}$ obtained from ω by removing τ's actions. Then $P \overset{\tilde{\omega}\omega'}{\Longrightarrow} Q' \xrightarrow{\tau} Q''$ and therefore $P \notin \Pi_{\not\circlearrowleft}$, which is a contradiction. □

Lemma 16. *If $P, Q \in \Pi_{\not\circlearrowright}$ are s.t. $P \approx Q$ and $P \xrightarrow{\alpha} P'$ with $\alpha \in A_\tau$, then $Q \xRightarrow{\alpha} Q'$, i.e. Q executes at least a transition, and $P' \approx Q'$*

Proof. If $\alpha \neq \tau$ the result is straightforward by Def. 7. If $\alpha = \tau$ and there is no transition $Q_0 \xrightarrow{\tau} Q_1$ s.t. $Q \Longrightarrow Q_0 \xrightarrow{\tau} Q_1 \Longrightarrow Q'$ and $P' \approx Q'$ then $P' \approx Q$ since $P \approx Q$. This implies that $P \approx Q \approx P'$, i.e. $P \xrightarrow{\alpha} P'$ is a stuttering transition. This contradicts $P \in \Pi_{\not\circlearrowright}$. □

Theorem 4. *If $P, Q \in \Pi_{\not\circlearrowright}$ and $P \approx Q$ then $depth(P) = depth(Q)$.*

Proof. We proceed by complete induction over $n = depth(P)$. If $depth(P) = 0$, then $P \sim \mathbf{0}$ and moreover $P \approx \mathbf{0}$. Because $P \approx Q$ and $P \sim \mathbf{0}$, there is no $\alpha \in A$ s.t. $Q \xRightarrow{\alpha}$. Taking into account this fact, if there is Q' s.t. $Q \xrightarrow{\tau} Q'$, Q' is such that $Q' \approx \mathbf{0}$. This creates a contradiction because $Q \xrightarrow{\tau} Q'$ is a stuttering transition and $Q \in \Pi_{\not\circlearrowright}$. Then $Q \sim \mathbf{0}$ and therefore $depth(Q) = 0 = depth(P)$. Suppose $depth(P) = n+1$. Let $\omega = \alpha\omega' \in A_\tau^*$ and P' be s.t. $length(\omega) = n+1$ and $P \xrightarrow{\alpha} P' \xrightarrow{\omega'}$. Because $P \approx Q$ and Lemma 16 there are Q_0, Q_1, Q' s.t. $Q \Longrightarrow Q_0 \xrightarrow{\alpha} Q_1 \Longrightarrow Q'$ and $P' \approx Q'$. By induction $depth(P') = depth(Q')$ and therefore $depth(Q) \geq depth(P) = n+1$. We prove now that when we assume $depth(Q) > n+1$ we reach a contradiction; it then follows that $depth(Q) = n+1$. Assume $depth(Q) > n+1$ and let $\omega = \alpha\omega' \in A_\tau^*$ be such that $Q \xrightarrow{\alpha} \tilde{Q} \xrightarrow{\omega'} \tilde{Q}' \downarrow$ and $length(\omega) = depth(Q)$. Because $P \approx Q$ and Lemma 16 there is $\tilde{P}$ s.t. $P \xRightarrow{\alpha} \tilde{P}$, $\tilde{P} \approx \tilde{Q}$ and $depth(P) = n+1 > depth(\tilde{P})$. By the complete induction $depth(\tilde{P}) = depth(\tilde{Q})$. Then, we reach a contradiction, $n+1 > depth(\tilde{P}) = depth(\tilde{Q}) \geq n+1$. □

Lemma 17. *If $P \in \Pi_f$ and $P \xrightarrow{\alpha} P'$ with $\alpha \neq \tau$ then $P \not\approx P'$.*

Proof. Let $\omega \in A^*$ be the largest sequence s.t. $P' \xRightarrow{\omega} P'' \downarrow$. Then there is no Q s.t. $P' \xRightarrow{\alpha\omega} Q$. On the other hand $P \xRightarrow{\alpha\omega}$, therefore $P \not\approx P'$. □

Lemma 18. *If $P \in \Pi_{\not\circlearrowright}$ is s.t. $P \not\sim \mathbf{0}$ and for all $P' \in \Pi_f, \alpha \in A_\tau$, $P \xrightarrow{\alpha} P' \downarrow$, then there is $\alpha' \neq \tau$ s.t. $P \xrightarrow{\alpha'}$.*

Proof. Because $P \not\sim \mathbf{0}$ there is α s.t. $P \xrightarrow{\alpha}$. If for all $P' \in \Pi_f, \alpha \in A_\tau$, $P \xrightarrow{\alpha} P' \downarrow$ and $\alpha = \tau$ then $P \approx \mathbf{0}$ and therefore all transitions that can be executed by P are stuttering transitions. This contradicts $P \in \Pi_{\not\circlearrowright}$. □

Theorem 5. *If $P, Q, R \in \Pi_{\not\circlearrowright}$, $P \not\approx \mathbf{0}$, $Q \not\approx \mathbf{0}$ and $P \parallel Q \approx R$ then $depth(P) < depth(R)$ and $depth(Q) < depth(R)$.*

Proof. We prove $depth(P) < depth(R)$, the proof that $depth(Q) < depth(R)$ is analogous. Note that, since $Q \not\approx \mathbf{0}$, there is Q' with $depth(Q') = 1$ that is reachable from Q, that is, there exists $\omega \in A^*$ s.t. $Q \xRightarrow{\omega} Q'$ and $Q' \not\sim \mathbf{0}$ (we remark the symbol $\not\sim$) and for all $Q'' \in \Pi_f, \alpha \in A_\tau$, $Q' \xrightarrow{\alpha} Q'' \downarrow$. Then, by Lemma 15, $Q' \in \Pi_{\not\circlearrowright}$ and, by Lemma 18, $Q' \xrightarrow{\alpha}$ with $\alpha \neq \tau$. Furthermore, by Convention 1 and the symmetric version of rule (Par-L) we have that $P \parallel Q \xRightarrow{\omega} P \parallel Q'$. By Lemma 17, $P \parallel Q' \xrightarrow{\alpha} P \parallel \mathbf{0}$ and $\alpha \neq \tau$ imply $P \parallel Q' \not\approx P \parallel \mathbf{0} \approx P$. Because $P \in \Pi_{\not\circlearrowright}$, whenever $S \in \Pi_{\not\circlearrowright}$ and $S \approx P \parallel Q'$, $depth(S) \geq 1 + depth(P)$ $(\star)$. Given that $R \approx P \parallel Q$, $P \parallel Q \xRightarrow{\omega} P \parallel Q'$ implies there is R' s.t. $R \xRightarrow{\omega} R'$ and $R' \approx P \parallel Q'$. By Lemma 15, $R' \in \Pi_{\not\circlearrowright}$. In addition, by $(\star)$, $depth(R') \geq 1 + depth(P)$. Finally $depth(R) \geq depth(R') \geq 1 + depth(P) > depth(P)$. □

Lemma 19. *If $P, Q \in \Pi_{\not\circlearrowright}$, $P \approx Q$ and $P \xRightarrow{\alpha} P'$, with $\alpha \in A_\tau$, then there is Q' s.t. $Q \xRightarrow{\alpha} Q'$.*

5.2 Unique parallel decomposition

The development in this section is similar to the development in Section 4.2, for this reason in some cases we use the same notation. This will not be a problem because both developments are independent. In order to use Theorem 1 we need to define a commutative monoid with a decomposition order. The commutative monoid is defined by

- $\mathbf{P}_\approx = \{[P]_\approx : P \in \Pi_f\}$ where $[P]_\approx = \{P' : P' \approx P\}$
- $e = [\mathbf{0}]_\approx \in \mathbf{P}_\approx$.
- $\parallel \subseteq \mathbf{P}_\approx \times \mathbf{P}_\approx$ is s.t. $[P]_\approx \parallel [Q]_\approx = [P \mid Q]_\approx$

Notice that we cannot ensure that for all $P', P'' \in [P]_\approx$, $depth(P') = depth(P'')$. Then, we extend the notion of depth in the following way. Define $[P]_\approx^{\not\circlearrowleft} = [P]_\approx \cap \Pi_{\not\circlearrowleft}$. For $[P]_\approx \in \mathbf{P}_\approx$, $depth([P]_\approx) = depth(P')$ with $P' \in [P]_\approx^{\not\circlearrowleft}$. This definition is sound because of Lemma 14 and Theorem 4.

Lemma 20. *$\mathbf{P}_\approx$ with neutral element $[\mathbf{0}]_\approx$ and binary operation $\parallel$ is a commutative monoid. I.e., $\parallel \subseteq \mathbf{P}_\approx \times \mathbf{P}_\approx$ satisfies the associativity, commutativity and identity properties.*

We shall define the partial order $\preccurlyeq$ over $\mathbf{P}_\approx$ using the relation $\rightarrow \subseteq \mathbf{P}_\approx \times \mathbf{P}_\approx$ defined as follows:

$$\begin{aligned}
\rightarrow_0 = \{([P]_\approx, [Q]_\approx) : \exists P' \in [P]_\approx^{\not\circlearrowleft}, Q' \in [Q]_\approx^{\not\circlearrowleft} : P' \overset{\alpha}{\Longrightarrow} Q', \alpha \in A_\tau \\
\text{and } \nexists P_0, P_1 \in \Pi_{\not\circlearrowleft} \text{ s.t. } P_0 \not\approx \mathbf{0}, P_1 \not\approx \mathbf{0}, P_0 \mid P_1 \approx P\}
\end{aligned}$$

$$\begin{aligned}
\rightarrow_{k+1} = {} & \{([P_0 \mid P_1]_\approx, [Q_0 \mid P_1]_\approx) : [P_0]_\approx \rightarrow_k [Q_0]_\approx, P_1 \in \Pi_f\} \\
& \cup \{([P_0 \mid P_1]_\approx, [P_0 \mid Q_1]_\approx) : [P_1]_\approx \rightarrow_k [Q_1]_\approx, P_0 \in \Pi_f\}
\end{aligned}$$

$$\rightarrow = \bigcup_{k=0}^{\infty} \rightarrow_k$$

The partial order $\preccurlyeq$ is defined as the inverse of the reflexive-transitive closure of $\rightarrow$ i.e., $\preccurlyeq = (\rightarrow^*)^{-1}$. We write $[P]_\approx \prec [Q]_\approx$ if $[P]_\approx \preccurlyeq [Q]_\approx$ and $[P]_\approx \neq [Q]_\approx$.

Notice that in this case $\rightarrow$ takes into account processes in $\Pi_{\not\circlearrowleft}$ and weak transitions that execute at least one transition. Also notice that we are avoiding communications between the arguments of the parallel composition in order to avoid scope extrusion.

Similarly to the strong setting, we need two lemmas, Lemmas 21 and 22, to prove that $\preccurlyeq$ is a partial order (Lemma 23). In addition, to prove that $\preccurlyeq$ is a decomposition order, we need the Lemma 24 that is equivalent to Lemma 11. The proofs of these results follow similarly to their respective counterpart in the strong setting. (The complete proofs are in the appendix.)

Lemma 21. *If $[P]_\approx \rightarrow [Q]_\approx$ then for all $\tilde{P} \in [P]_\approx$ there are $\alpha \in A_\tau$ and $\tilde{Q} \in [Q]_\approx$ s.t. $\tilde{P} \overset{\alpha}{\Longrightarrow} \tilde{Q}$.*

Lemma 22. *If $[P]_\approx \rightarrow [Q]_\approx$ then $depth([P]_\approx) > depth([Q]_\approx)$.*

Lemma 23. *$\preccurlyeq$ is a partial order.*

Lemma 24. *If $P \in \Pi_{\not\circlearrowleft}$ and $depth(P) > 0$ then there is Q s.t. $[P]_\approx \rightarrow [Q]_\approx$.*

We are ready to prove that $\preccurlyeq \subseteq \mathbf{P}_\approx \times \mathbf{P}_\approx$ is a decomposition order. This proof does not present changes w.r.t. proof of Lemma 12 except that for proving $\preccurlyeq$ is Archimedean, we use Theorem 5. Notice that there is no lemma equivalent to Lemma 6 in the weak setting.

Lemma 25. *$\preccurlyeq \subseteq \mathbf{P}_\approx \times \mathbf{P}_\approx$ is a decomposition order.*

By Theorem 1, it follows that $\mathbf{P}_\approx$ has unique decomposition.

Corollary 2. *The commutative monoid $\mathbf{P}_\approx$ has unique decomposition.*

6 Final Remarks

In this paper we have proved that finite processes of the π-calculus satisfy UPD w.r.t. both strong bisimilarity and weak bisimilarity. We have obtained these results using the technique presented in [15] (see Theorem 1) and different properties that are satisfied in each setting. For the strong setting, we had to prove properties related to the depth of processes. For the weak setting, we had to prove properties related to processes that execute no stuttering transitions. Our results show that the abstract framework of [15] can be used in the context of the π-calculus, dealing with the complications that arise from scope extrusion. In addition, the same framework can be used to deal with the weak setting if one considers processes without stuttering transitions. In this way, we have avoided the abstract technique introduced in [14] which is considerably more involved than the technique that we have used in this paper.

In Section 4 we showed with two examples that norm is not additive for the π-calculus and therefore some proofs in [7] are flawed. After pointing out this problem to Dreier et al., they proposed us an alternative definition of norm to solve it. Call this variant *norm′*. Roughly, *norm′* should not consider traces where there is a scope extrusion of processes. We think this solution may work for the applied π-calculus, but is not suitable for the variant of the π-calculus considered in the present paper. We first explain what is the problem in our context, and then why this problem is not present in applied π. In the first example in Section 4, we had $P = P_0 \mid P_1 = \nu z(\overline{a}z) \mid a(x).!\overline{x}a$. Process P is not normed, i.e. $norm'(P) = \infty$, because the only finite trace that the process executes, $P \xrightarrow{\tau} \nu z(\mathbf{0} \mid !\overline{z}a)\downarrow$, goes through a scope extrusion. Now, consider the process

$$P' = \nu z(\overline{a}z).(\mathbf{0} \mid a(x).!\overline{x}a) \ + \ a(x).(\nu z(\overline{a}z) \mid !\overline{x}a) \ + \ \tau.(\nu z(\mathbf{0} \mid !\overline{z}a)) \ ;$$

it would be normed according to the alternative definition suggested above (the τ-transition from P' is not the result of a scope extrusion). Now, since P' is just the expansion of P, it is clear that $P \sim P'$. Thus, we find that the property of being normed is not compatible with bisimilarity. Since the applied π-calculus does not include the construct for non-deterministic choice needed for the expansion, this problem is not present there.

An open question that leaves this paper is related with UPD of the π-calculus w.r.t. *strong full bisimilarity*[20]. Strong full bisimilarity is a stronger notion of bisimulation that is a congruence for all constructs of the π-calculus. We have tried to apply the abstract technique in this setting so far without success. When we tried to repeat the result in Section 4, taking into account the universal quantification in the definition of strong full bisimilarity, a problem arose when we wanted to prove that the order is a decomposition order. Particularly, we were not able to prove that the order is strict compatible. Notice that this problem is not present in the *asynchronous π-calculus*[20], a well-known fragment of the π-calculus, because (strong) bisimilarity and (strong) full bisimilarity coincide.

Acknowledgement. The authors thank Daniel Hirschkoff for discussions, comments and suggestions on various drafts of this paper, and anonymous reviewers for their thorough reviews and good suggestions.

References

[1] L. Aceto, W. Fokkink, A. Ingólfsdóttir & B. Luttik (2005): *CCS with Hennessy's merge has no finite-equational axiomatization*. *Theor. Comput. Sci.* 330(3), pp. 377–405, doi:10.1016/j.tcs.2004.10.003.

[2] L. Aceto, W. Fokkink, A. Ingólfsdóttir & B.. Luttik (2009): *A finite equational base for CCS with left merge and communication merge*. *ACM Trans. Comput. Log.* 10(1), doi:10.1145/1459010.1459016.

[3] L. Aceto, A. Ingólfsdóttir, B. Luttik & P. van Tilburg (2008): *Finite Equational Bases for Fragments of CCS with Restriction and Relabelling*. In: *IFIP 20th World Computer Congress, TC 1, Foundations of Computer Science*, pp. 317–332, doi:10.1007/978-0-387-09680-3_22.

[4] I. Castellani (1998): *Bisimulations for Concurrency*. Ph.D. thesis, University of Edinburgh. Also published as LFCS-88-51.

[5] S. Christensen (1993): *Decidability and Decomposition in Process Algebra*. Ph.D. thesis, University of Edinburgh.

[6] F. Corradini, R. Gorrieri & D. Marchignoli (1998): *Towards parallelization of concurrent systems*. *Informatique théorique et applications* 32(4-6), pp. 99–125.

[7] J. Dreier, C. Ene, P. Lafourcade & Y. Lakhnech (2016): *On the existence and decidability of unique decompositions of processes in the applied π-calculus*. *Theor. Comput. Sci.* 612, pp. 102–125, doi:10.1016/j.tcs.2015.11.033.

[8] J. Dreier, P. Lafourcade & Y. Lakhnech (2012): *Defining Privacy for Weighted Votes, Single and Multi-voter Coercion*. In: *ESORICS 2012*, pp. 451–468, doi:10.1007/978-3-642-33167-1_26.

[9] W. Fokkink & B. Luttik (2000): *An* omega-*Complete Equational Specification of Interleaving*. In: *Automata, Languages and Programming, 27th International Colloquium, ICALP 2000*, pp. 729–743, doi:10.1007/3-540-45022-X_61.

[10] J. Friso Groote & F. Moller (1992): *Verification of Parallel Systems via Decomposition*. In: *CONCUR '92, Third International Conference on Concurrency Theory*, pp. 62–76, doi:10.1007/BFb0084783.

[11] D. Hirschkoff & D. Pous (2008): *A Distribution Law for CCS and a New Congruence Result for the π-calculus*. *Logical Methods in Computer Science* 4(2), doi:10.2168/LMCS-4(2:4)2008.

[12] Y. Hirshfeld & M. Jerrum (1999): *Bisimulation Equivalence Is Decidable for Normed Process Algebra*. In: *Automata, Languages and Programming, 26th International Colloquium, ICALP'99*, pp. 412–421, doi:10.1007/3-540-48523-6_38.

[13] I. Lanese, J. A. Pérez, D. Sangiorgi & A. Schmitt (2011): *On the expressiveness and decidability of higher-order process calculi*. *Inf. Comput.* 209(2), pp. 198–226, doi:10.1016/j.ic.2010.10.001.

[14] B. Luttik (2016): *Unique parallel decomposition in branching and weak bisimulation semantics*. *Theor. Comput. Sci.* 612, pp. 29–44, doi:10.1016/j.tcs.2015.10.013.

[15] B. Luttik & V. van Oostrom (2005): *Decomposition orders another generalisation of the fundamental theorem of arithmetic*. *Theor. Comput. Sci.* 335(2-3), pp. 147–186, doi:10.1016/j.tcs.2004.11.019.

[16] R. Milner & F. Moller (1993): *Unique Decomposition of Processes*. *Theor. Comput. Sci.* 107(2), pp. 357–363, doi:10.1016/0304-3975(93)90176-T.

[17] F. Moller (1989): *Axioms for Concurrency*. Ph.D. thesis, University of Edinburgh.

[18] F. Moller (1990): *The Importance of the Left Merge Operator in Process Algebras*. In: *Automata, Languages and Programming, 17th International Colloquium, ICALP90*, pp. 752–764, doi:10.1007/BFb0032072.

[19] F. Moller (1990): *The Nonexistence of Finite Axiomatisations for CCS Congruences*. In: *Proceedings of LICS '90*, pp. 142–153, doi:10.1109/LICS.1990.113741.

[20] D. Sangiorgi & D. Walker (2001): *π-Calculus: A Theory of Mobile Processes*. Cambridge University Press, New York, NY, USA.