INFORMATIQUE THÉORIQUE ET APPLICATIONS

F. CORRADINI
R. GORRIERI
D. MARCHIGNOLI

Towards parallelization of concurrent systems

Informatique théorique et applications, tome 32, nº 4-6 (1998), p. 99-125
<http://www.numdam.org/item?id=ITA_1998__32_4-6_99_0>

Informatique théorique et Applications/Theoretical Informatics and Applications
(vol. 32, n° 4-5-6, 1998, pp. 99-125)

TOWARDS PARALLELIZATION OF CONCURRENT SYSTEMS (*) (1)

by F. CORRADINI (2), R. GORRIERI (3) and D. MARCHIGNOLI (3)

Communicated by W. BRAUER

Abstract. – *A notion of parallelization of concurrent processes is proposed that satisfies some intuitive requirements. Roughly, given a process P, a more parallel version Q of P can be the result of replacing one of the sequential summands S in one of P subterms by a process R, provided that Q is functionally equivalent to P and R is either a parallel term or a summand of S. This defines an equivalence preserving preorder on processes $Q \sqsubseteq P$ according to which in Q parallelism is increased or the amount of redundancy is decreased. We show that our notion has some connection with the notion of factorization proposed by Milner and Moller. Finally, we identify some classes of processes for which the most parallel version is unique.* © Elsevier, Paris

Résumé. – *Nous proposons une notion de parallélisation de processus concurrents qui satisfait quelques exigences intuitives. En gros, si P est un processus, une version plus parallèle Q de P peut être obtenue en remplaçant un des sommants séquentiels S dans un sous-terme de P, pourvu que Q soit fonctionnellement équivalent à P et que R soit ou bien un terme parallèle ou bien un somant de S. Ceci définit une équivalence qui conserve le préordre sur les processus $Q \sqsubseteq P$ suivant lequel ans Q le paralélisme est augmenté ou bien la quantité de redondance est diminuée. Nous montrons que notre notion est reliée à celle de factorisation proposée par Milner et Moller. Finalement, nous identifions quelques classes de processus pour lesquels la version la plus parallèle est unique.* © Elsevier, Paris

1. INTRODUCTION

Given a process P, the problem of defining the most parallel version Q of P has been largely ignored up to now in the literature. To the best of our knowledge, there is no work that studies this problem in a process algebraic setting. Only the definition of *factorization*, proposed by Milner and Moller

(*) Received December 1996; revised September 1997; accepted September 1997.
(1) This first author has been partially funded by EEC within the HCM Project EXPRESS; the second aythor by Esprit project LOMAPS 8130, by CNR and MURST.
(2) Dipartimento di Matematica Pura ed Applicata, Università di l'Aquila, e-mail: flavio@univaq.it
(3) Dipartimento di Scienze dell'informazione, Università di Bologna, e-mail: gorrieri @cs.unibo.it

0988-5004/98/04-05-06/© Elsevier-Paris

[MM93], might be thought of as a possible notion of parallelization (even if it was not proposed to this aim). Intuitively, we would like that the most parallel version Q of a process P satisfies the following requirements.

- **Equivalence preservation:** Once chosen an equivalence notion $\sim$ among processes, we would like that $P \sim Q$, as the parallelization procedure shold not alter the "functional" behaviour of the process. As the behavioural equivalence is a free parameter, we fix it by choosing strong bisimulation [Mil89].
- **Persistency:** During execution, a maximally parallel process P should remain so. To be more precise, if P is maximaly parallel and $P \xrightarrow{\alpha} P'$, then also P' must enjoy the property. This ensure that the property holds for all derivatives.
- **Summation context independence:** If some summands of P can be parallelized, then they can be replaced by their most parallel version. For instance, consider $P = a.b + b.a + a.a$; it is clear that both subterms $a.b + b.a$ and $a.a$ can be made more parallel as $a|b$ and $a|a$. Hence the most parallel versions of P is $Q = a|b + a|a$. In other words, we require that, independently of the choice of the execution path, this is in the most parallel form. Note that this requirement is not the reverse of persistency (which could be read as "if all the non-trivial derivatives are maximally parallel than also the initial process is so"), as P above is a counter-example.
- **Decrease of redundancy:** It seems reasonable to assume that a maximaly parallel process should not offer redundant, less parallel computations. For instance, consider $P = a|b + ab$. It is clear that the non-determinism offered by the subterm ab is useless, as P can do the same in a more parallel way from the subterm $a|b$. Hence, P can be safely reduced to $a|b$.
- **Increase of distribution/efficiency:** We would like to prove that, given a process P, its maximal parallelization Q is more distributed and more efficient than P. And moreover, that there is no way to do better (at least, under some circumstances). Of course, this means that we need to introduce suitable truly concurrent semantics which express distribution and timing information. Even if crucial for assessing the merits of a notion of parallelization, here we only provide some examples that give intuitive evidence that this requirement is met. More can be found in Marchigloli's thesis [Mar96].

Intuitively, our parallelization notions is defined as follows: in a process P we allow the replacement of one of the sequential summands S in one of P subterms by a process R if the resulting process Q is equivalent to P and R is either a parallel term or a summand of S. This defines a preorder on processes $Q \sqsubseteq P$ according to which parallelism is increased in Q or the amount of redundancy is decreased in Q.

Contrary to many preorders in the literature, our preorder is equivalence perserving in order to meet the first requirement above. This choice has been driven essentially by the following consideration. There are two main interpretations that can be given to actions in a process algebra. According to the first, actions are channels or interacting ports; hence, $ab + ba$ is indeed equivalent to $a|b$ as the same interactions are provided by both processes, but the latter is more parallel and possibly efficient. The second interpretation sees actions as operations to be performed (e.g., on a shared memory), equipped by a dependency relation which states which pairs cannot be executed in parallel. Under this interpretation, if a and b are dependent, then $ab + ba$ is equivalent to $a|b$, as the execution of the actions in the latter process can be done only in mutual exclusion; similarly, if they are independent, then $a|b$ should be considered equivalent to ab and to ba, and so also to $ab + ba$ (even if with redundancy). As we do not want to fix any interpretation, we have noticed that the set of possible traces should be preserved while parallelizing. Hence, we need to choose an equivalence relation which is at least as strong as trace equivalence. Bisimulation equivalence, that is stronger than trace equivalence, has been chosen because it is one standard equivalence relation for concrete (i.e., without invisible actions) process algebras and, moreover, it is widely used in the only paper we know that might offer material for a comparison [MM93].

The paper is organized as follows. Section 2 introduces the language we use. Like in [MM93], the language we use is very simple: it contains prefixing, summation and parallel composition without comunication. Some remarks about the extension of the work to other operators are reported in the conclusions. Section 3 recalls the basic definition and results from [MM93], which will be useful in the following, not only for comparison. Section 4 introduces our notions of parallelization, while closure and structural properties are reported in Section 5. Section 6 compares parallelization and Milner & Moller's factorization, showing that the latter cannot be considered an adequate notion of parallelization. Section 7 presents an abstract reduction system to show that the parallelization procedure always terminates. However, it is not confluent; uniqueness of normal forms (up to a structural congruence considering associative and commutative the parallel and alternative composition operators) is guaranteed only for the sublanguage without the + operator and, in the full language, under some conditions (Section 8).

2. PRELIMINARIES

The language used in the paper as a case study is a subset of Milner's CCS. The set of atomic actions is denoted by Act, ranged over by $a, b, \ldots$. The set $\mathbf{P}$ of processes, ranged over by $P, Q, \ldots$, is composed by the terms generated by the grammar below:

$$P := \mathbf{0} \mid a.P \mid P + P \mid P|P \quad \text{where } a \in Act$$

$\mathbf{0}$, called nil, denotes a terminated process. $a.P$ denotes a process which can do an action a and then behaves like P. $P + Q$ denotes alternative composition of P and Q, while $P|Q$ denotes their parallel composition, that does not enable communication. Final nil's are usually omitted; e.g., $a + b$ stands for $a.\mathbf{0} + b.\mathbf{0}$. As usual, the precedence of operators is as follows: $. > \mid > +$.

The standard operational semantics is given in terms of a labelled transition system $(\mathbf{P}, Act, \longrightarrow)$, where the transition relation $\longrightarrow \subseteq \mathbf{P} \times Act \times \mathbf{P}$ is defined through the set of inference rules listed in Table 1 (where the symmetric rules for alternative and parallel compositions are omitted).

TABLE 1

The Rules for the Operational Semantics (symmetric rules omitted).

$$Act \quad \frac{-}{a.P \xrightarrow{a} P} \qquad Sum \quad \frac{P \xrightarrow{a} P'}{P + Q \xrightarrow{a} P'}$$

$$Par \quad \frac{P \xrightarrow{a} P'}{P|Q \xrightarrow{a} P'|Q}$$

The semantic congruence we consider is strong bisimilarity $\sim$ [Mil89]. It is defined as the largest binary symmetric relation on $\mathbf{P}$ such that $P \sim Q$ if and only if, for all $a \in Act$:

$P \xrightarrow{\alpha} P'$ implies $Q \xrightarrow{\alpha} Q'$ for some Q' such that $P' \sim Q'$.

In the rest of the paper we will use = to denote the syntactic identity and $\equiv$ to denote the structural congruence generated by the set of equations below

$$P + Q \doteq Q + P \qquad P|Q \doteq Q|P$$
$$P + (Q + R) \doteq (P + Q) + R \qquad P|(Q|R) \doteq (P|Q)|R$$

This structural congruence is convenient to define a flexible notion of subterm, letting us to interpret + and | as if they were n-very operators. Let $P \leq Q$ denote that P is a subterm of Q and let consider process $P = (a + b) + c$; we would like to have, for instance, the following: $(a + b) \leq P$, $a \leq P$, $b \leq P$, $c \leq P$, but also $a + c \leq P$ and $b + c \leq P$.

Relation $\leq$ is the reflexive (up to $\equiv$) and transitive closure of the binary $<$ over P. Formally: $\leq \stackrel{\mathrm{def}}{=} \cup_{i \in N} <^{i}$, where $<^{0} \stackrel{\mathrm{def}}{=} \equiv$ and $<^{i+1} \stackrel{\mathrm{def}}{=} <^{i} \circ <$ and, finally, $<$ is defined by the following rules:

$$P < \alpha.P \quad P < P + Q \quad P < P|Q$$

We say that Q is a strict subterm of P if $Q \leq P$ but $Q \not\equiv P$. In the following we will always refer to the definition of subterm as given above.

PROPOSITION 2.1: *$P \equiv Q$ implies $P \sim Q$ for all P, $Q \in \mathbf{P}$.*

Proof: Obvious, as all the equations for $\equiv$ are sound w.r.t. bisimulation equivalence. □

The proposition above justifies the use of the structural congruence, as all the terms in the same class have the same semantics. In virtue of this, an alternative representation of a congruence class of terms is a term where the operators + and | are n-vary. For this reason, we sometimes represent any term of the congruence class in a form where parentheses are fogotten as in $P_1 + \ldots + P_n$ and $P_1 | \ldots | P_n$. Sometimes we will abbreviate $P_1 + \ldots + P_n$ with $\sum_{i \in I} P_i$ and $Q_1 | \ldots | Q_n$ with $\prod_{i \in I} Q_i$, $I = [1..n]$, understanding that each P_i does not have + as its outermost operator and that each Q_i does not have | as its outermost operator. These conditions are useful, as they give the precise number of summands or parallel components, respectively.

Some of the proofs in this paper will proceed by induction on the size of processes, denoted by $|P|$; that is defined by induction as follows:

$$|0| = 0 \qquad |a.P| = 1 + |P|$$
$$|P + Q| = \max\{|P|, |Q|\} \qquad |P|Q| = |P| + |Q|$$

It is not difficult to see that bisimilar processes have the same size. Among the consequences of this fact, we mention that $|P| > 0$ iff $P \not\sim \mathbf{0}$.

3. FACTORIZATION OF PROCESSES

In this section we briefly review the main concepts and results behind Milner and Moller's work on factorization of processes [MM93]. In no way we intend to say that Milner and Moller proposed factorization as a possible solution to the problem we are facing here. Nonetheless, their notion of factorization could be read as a form of parallelization, and so we feel authorized to make a (rather ingenerous) comparison. Moreover, another reason for recalling factorization is that it is also used in another part of the paper, namely in the proof of Proposition 8.1.

DEFINITION 3.1: *A process P is prime if $P \not\sim \mathbf{0}$ and whenever $P \sim P_1|P_2$ we have that either $P_1 \sim \mathbf{0}$ or $P_2 \sim \mathbf{0}$.*

DEFINITION 3.2: *A process $P \equiv P_1|\ldots|P_n$ is factorized if every P_i is prime.*

Note that we could define $\mathbf{0}$ as the parallel composition of zero terms. By this way we can consider $\mathbf{0}$ as factorized, even if not prime.

In [MM93] it has been shown that for any process P there is a unique (up to $\sim$) multiset $\{P_1, \ldots, P_n\}$ of primes for which $P \sim P_1|\ldots|P_n$.

THEOREM 3.1: *Any process can be expressed uniquely, up to $\sim$, as a parallel composition of primes.*

Now we want to see if the notion of factorization may be a sensible candidate to match our intuition (still quite informal) of maximal parallelism. Unfortunately, this is not the case, as the following examples show that not all the requirements discussed in the Introduction are met.

Clearly (see Theorem 3.1) the factorization of a process is strongly bisimilar to the process itself, thus equivalence preservation holds. Factorization does not always satisfy the requirement of distribution/efficiency increase: consider the factorized process $P = a.(b.c + c.b)$; however, $Q = a.(c|b)$ seems, intuitively, to be more parallel than P (confirmed also by the fact that $Q \sqsubseteq_p P$ and $Q \sqsubseteq_l P$ where $\sqsubseteq_p$ and $\sqsubseteq_l$ are the performance preorder and the location preorder studied, respectively, in [CGR95, CGR97] and [BCHK92]). Process P also shows that persistency does not hold because, after the execution of action a, the resulting process is not factorized.

Factorization does not even satisfy the requirement of summation context independence: the factorized process $P = (a|b) + a.a$ can be turned into the more parallel process $(a|b) + (a|a) \sim P$.

Finally we show that also the decrease of redundancy does not hold for factorization, as they are disjont concepts; We can have factorized, minimal redundant terms (e.g. $a|b$) as well as factorized redundant terms (e.g. $a|b + a.b + a$, that is more redundant than $a|b + a$).

4. MAXIMALLY PARALLEL PROCESSES

In this section we present a notion of reducible process that takes into account the maximal parallelism that a process can show up. We will show in Section 5 that the new notion satisfies all requirements discussed in the Introduction and prove some pleasant properties it enjoys.

Trying to obtain the new notions by refining factorization, the obvious extension that immediately springs in mind is that of requiring that each subprocess is also factorized. In this way, for instance, we would obtain a persistent property. However, also this notion – apparently more suitable – is not completely satisfactory. Process $P \equiv (a|b|c) + (c.a.b) + (b.c.a)$ is such that every subprocess is factorized even if $P \sim (a|b|c) + (c|a.b) + (b|c.a)$ and the right-hand side is clearly more parallel than P.

We need some preliminary definitions. First we introduce choice processes, that are processes not having parallel composition as the outermost operator.

DEFINITION 4.1: *A process P is a choice process if it is generated by the following grammar*

$$S ::= a.P \mid P + Q \quad \text{with } a \in Act,\ P,\ Q \in \mathbf{P}$$

Choice processes will be ranged over by S (possibly indexed).

The next definition introduces our notion of completely reduced (or maximally parallel) processes. The definition is given inductively on the syntactical structure of process. Intuitively, it states when a process cannot be turned into a more parallel version by replacing some sequential subpart with a parallel term.

DEFINITION 4.2: (Completely Reduced or maximally Parallel Process) *A process P is completely reduced if and only if one of the following holds:*

1) $P = \mathbf{0}$;
2) $P = a.P'$ and both conditions below hold:
   i) P' is completely reduced,

ii) $\nexists P_1 \not\sim \mathbf{0}$, $\nexists P_2 \not\sim \mathbf{0}$ with P_1 and P_2 completely reduced such that $P_1|P_2 \sim P$;

3) $P = P' + P''$ and $\forall P_1, P_2$ such that $P \equiv P_1 + P_2$, the four conditions below hold:

i) P_1 and P_2 are completely reduced,

ii) $\nexists P_1' \not\sim \mathbf{0}$, $\nexists P_2' \not\sim \mathbf{0}$ with P_1' and P_2' completely reduced such that $P_1'|P_2' \sim P_1 + P_2$,

iii) if $P_1 \equiv S$, then $\nexists P_1' \not\sim \mathbf{0}$, $\nexists P_2' \not\sim \mathbf{0}$ with P_1' and P_2' completely reduced such that $(P_1'|P_2') + P_2 \sim P_1 + P_2$,

iv) $P_1 \not\sim P_1 + P_2$,

4) $P = P'|P''$ and P' and P'' are completely reduced.

Cases 1, 2 and 4 are obvious. More interesting is case 3: for all choices of P_1 and P_2 such that their summation is structurally congruent to P, we require that i) they are both maximally parallel and, more importantly, that ii) $P_1 + P_2$ cannot be transformed in a term with $|$ as outermost operator – as well as iii) P_1, if it is a choice process; finally, the fourth condition avoids presence of redundancy. Note that condition ii) is independent of the choice of P_1 ad P_2, as for every such P_1, P_2 we have that $P = P' + P'' \sim P_1 + P_2$, and so condition ii) can be moved outside the universal quantification.

The following of this section is devoted to give a characterization of maximally parallel processes that will be useful in the following. It is based on the reducibility property of sequential processes given below.

DEFINITION 4.3: (Reducible process) *A process P is reducible if one of the following three conditions holds (otherwise P is irreducible).*

i) P is a choice process and $\exists R_1 \not\sim \mathbf{0}$, $\exists R_2 \not\sim \mathbf{0}$ with R_1 and R_2 completely reduced such that $R_1|R_2 \sim P$. $R_1|R_2$ is said *contraction* of P in P;

ii) $P \equiv S + P''$ and $\exists R_1 \not\sim \mathbf{0}$, $\exists R_2 \not\sim \mathbf{0}$ with R_1 and R_2 completely reduced such that $(R_1|R_2) + P'' \sim P$. $R_1|R_2$ is said *contraction* of S in $S + P''$;

iii) $P \equiv P' + P''$ and $P' \sim P' + P''$. P' is said *contraction of* $P' + P''$ in $P' + P''$.

Q is a *reduction* of P if it is obtained from P by applying the contraction; in other words, $Q = R_1|R_2$ for i), $Q = (R_1|R_2) + P''$ for ii) and $Q = P'$ for iii).

Observe that items 2.ii), 3.ii) and 3.iii) of Definition 4.2 and items i) and ii) of Definition 4.3 share a similar side condition of the form "with P_1 and P_2 completely reduced". It is not difficult to see that, if we remowe such a condition in every item of Definition 4.3, we would get an equivalent definition of completely reduced process and of reducible process. The current definition has the advantage of permitting shorter proofs of the propositions in Section 5 and 7.

Let us see some examples. Process $a|a$ is a contraction of $a.a$, so that the latter is reducible. And also a is a contraction of $a + a$. Hence $a + a$ is reducible. To give a more interesting example consider process $P \equiv a+a.b+b.a$. Given $S = a.b+b.a$ and $Q = a$, we have $S+Q \sim a|b+Q$ thus P is reducible. Process $a|b + a$ is now irreducible. Note that the notion of irreducibility is quite different of that of prime given in the previous section. Indeed, a parallel process (e.g. $a|b$) is always irreducible, but it cannot be prime. More on this comparison in Section 6.

The example below show that, in a contraction, the substituted terms can even be not bisimilar.

Example 4.1: In process $P \equiv a|b|c + a.c.b + c.a.b$ there is no contraction R bisimilar to any summand of P. Nevertheless P is reducible; in fact, by letting $S = c.a.b$, $Q = a|b|c + a.c.b$ and $R = c|a.b$, we have $P \equiv S + Q \sim R + Q \equiv c|a.b + a|b|c + a.c.b$. Again, process $R + Q$ is reducible. Similarly as above, let $S \sim a.c.b$ we have $P \sim a|b|c+c.b|a+c|a.b$.

Now we want to relate the notion of maximal parallelism with the notion of reducibility.

PROPOSITION 4.1: *A process P is completely reduced if and only if each of its subterms is irreducible.*

Proof: Assume P completely reduced. We prove that $\nexists P' \leq P$ such that either item i), ii) or iii) of Definition 4.3 holds. We proceed by induction on the syntactic structure of P.

a) $P = \mathbf{0}$, then the statement immediately follows.

b) $P = a.P_1$. We distinguish two cases:

i) $P' \leq P_1$. As P is completely reduced P_1 is so too. Then, the thesis follows by induction.

ii) $P' \equiv a.P_1$. Since P is completely reduced, $\nexists P_1' \not\sim \mathbf{0}$, $\nexists P_2' \not\sim \mathbf{0}$ with P_1' and P_2' completely reduced such that $P_1'|P_2' \sim a.P_1$. And so there is no contraction of $a.P_1$.

c) $P = P_1 + P_2$. Again we have two cases to consider:

i) P' is a strict subterm of $P_1 + P_2$. Hence, P' is a subterm of some P_1' such that there exists a P_2' with $P_1 + P_2 \equiv P_1' + P_2'$. Note that

P'_1 is completely reduced by item 3.i) of Definition 4.2. Hence, P' is irreducible by induction hypothesis.

ii) $P' = P'_1 + P'_2 \equiv P_1 + P_2$. By definition of completely reduced we have

1) $\nexists R_1 \not\sim \mathbf{0}$, $\nexists R_2 \not\sim \mathbf{0}$ with R_1 and R_2 completely reduced such that $R_1|R_2 \sim P'$.

2) If $P'_1 \equiv S$ then $\nexists R_1 \not\sim \mathbf{0}$, $\nexists R_2 \not\sim \mathbf{0}$ with R_1 and R_2 completely reduced such that $(R_1|R_2) + P'_2 \sim P'$.

3) $P'_1 \not\sim P'_1 + P'_2$.

Items 1) 2) and 3), corresponding to the similar items of Definition 4.3, show that it is not possible to find any contraction in P'.

d) $P = P_1|P_2$. The subterms of P are the subterms P'_i of P_i for $i = 1, 2$, as well as their parallel composition. By definition of completely reduced, each P_i is completely reduced. By induction, each P'_i is irreducible, and so are also their parallel compositions $P'_1|P'_2$.

Conversely, let us now assume that $\nexists P' \leq P$ such that either item i), ii) or iii) of the definition of reducible process holds. Then, we prove that P is completely reduced. Also in this case, we proceed by induction on the syntactic structure of P.

a) $P = \mathbf{0}$, then P is completely reduced by definition.

b) $P = a.P_1$. As P_1 is a strict subterm of P, then each of P_1 subterms is irreducible. By induction hypothesis, P_1 is completely reduced, and so we have matched the first requirement for $a.P_1$ to be completely reduced. Now, we also know that $a.P_1$ is irreducible. As only item i) of the Definition 4.3 could be applicable in this case, we know that $\nexists R_1 \not\sim \mathbf{0}$, $\nexists R_2 \not\sim \mathbf{0}$ with R_1 and R_2 completely reduced such that $R_1|R_2 \sim a.P_1$. And this is the second requirement for $a.P_1$ to be completely reduced.

c) $P = P_1 + P_2$. We prove that the four items of the definition of completely reduced hold. Consider a generic $P' = P'_1 + P'_2 \equiv P_1 + P_2$. Item i) is proved as follows: As all the subterms of P are irreducible, then also P'_1 and P'_2 are so, as they are subterms. Hence, by induction hypothesis, P'_1 and P'_2 are completely reduced. To prove items ii), iii) and iv), let us note that, as P' is a subterm of P, it is irreducible. Then, item ii) is a consequence of item i) of Definition 4.3; similarly, item iii) follows by item ii) of Definition 4.3, as well as item iv) by item iii) of Definition 4.3.

d) $P = P_1|P_2$. As all the subterms of P are irreducible, then P_1 and P_2 are irreducible. By induction hypothesis, we have that P_1 and P_2 are completely reduced; hence, so is also P. □

We end this section with some examples.

Example 4.2: $P = a.(a.b + b.a)$ is irreducible but it is not completely reduced; indeed its subterm $a.b+b.a$ is reducible, as we saw before. Similarly $Q \equiv a + a.b + b.a$ is not completely reduced. Note that both P and Q are factorized.

Example 4.3: $P = a|b + a|b$ is reducible; indeed $a|b$ is a (bisimilar) contraction for P in P. Clearly P and $a|b$ have exactly the same degree of parallelism, but in some way the latter is less redundant than the former.

Example 4.4: Process $Q \equiv a.b|a|b + a.b|a.b + a$ is not completely reduced because subterm $a.b|a|b + a.b|a.b$ is reducible. Indeed $Q \sim a.b|a|b + a$.

This example is particularly interesting because it shows how our notion of complete reduction permits to obtain processes that are more parallel and, at the same time, more deterministic, using only rule ii) of Definition 4.3. This suggests that – while parallelizing – it is sometimes necessary to decrease redundancy.

5. SOME PROPERTIES OF MAXIMALLY PARALLEL PROCESSES

5.1. Closure properties

Maximal parallelism is not preserved by prefixing and alternative composition, as the following example establishes, while it is preserved by parallel composition.

Example 5.1: Consider process $P = a$; obviously, P is completely reduced but $a.P$ is not so. Furthermore, processes $P = a.b$ and $Q = b.a$ are completely reduced, while $P + Q$ is not.

The following proposition – implicitly used in a few places in the proof of Proposition 4.1 – follows directly by item 4) of Definition 4.2.

PROPOSITION 5.1: *Let P, Q completely reduced processes. Then $P|Q$ is completely reduced.*

Finally, the proposition below guarantees that we can always work modulo the structural congruence $\equiv$.

PROPOSITION 5.2: *Let P be completely reduced and $P \equiv Q$. Then Q is completely reduced.*

Proof: Observe that, for all the equations defining $\equiv$, if the lhs is completely reduced, then so is the rhs. Indeed, the case of $+$-associativity follows by the universal quantification in item 3) of Definition 4.2, while the $|$-associativity follows by a simple inductive argument. □.

5.2. Structural properties

PROPOSITION 5.3: *If Q is the reduction of P, then $P \sim Q$.*

Together with Theorem 5.1, this proposition, that follows by Definition 4.3, ensures that the first requirement (*equivalence preservation*) is actually met.

PROPOSITION 5.4: *Each subterm of a completely reduced process is completely reduced.*

The proposition above follows directly by definition of completely reduced process, because $\mathbf{0}$ has no strict subterms, 2.i) and 3.i) requires this property to hold for all proper subterms of a choice process; finally, 4) asks this for the immediate components of a parallel composition, whence the thesis by induction and by Proposition 5.1. This proposition states that our notion satisfies the requirement of *summation context independence*: In order to be completely reduced, each summand of a choice process has to be completely reduced.

The following proposition shows that the notion of maximal parallelism satisfies also the *persistency* requirement. Indeed, we prove that each derivative of a completely reduced process is completely reduced.

PROPOSITION 5.5: *Let P be a completely reduced process. Then $P \xrightarrow{a} P'$ implies P' is completely reduced.*

Proof: By induction on the depth of the derivation tree of transition $P \xrightarrow{a} P'$. We proceed by cases analysis on the syntactic structure of P.

- Let $P = a.P'$. Then $a.P' \xrightarrow{a} P'$. Since P' is a subterm of P, P' is completely reduced by Proposition 5.4.
- Let $P = P_1 | P_2$. Assume $P_1 \xrightarrow{a} P_1'$ (the symmetric case is similar). Note that processes P_1 and P_2 are completely reduced because they are subterms of P. Thus, by induction hypothesis, also P_1' is completely reduced and, by Proposition 5.1, also $P_1' | P_2$ is completely reduced.

- Let $P = P_1 + P_2$. As above, assume $P_1 \xrightarrow{a} P_1'$ (the symmetric case is similar). Note that processes P_1 and P_2 are completely reduced as they are subterms of P. Then, by induction hypothesis. P_1' is completely reduced too. □

Now we want to show another property of completely reduced processes, related to the degree of nondeterminism of processes. More precisely, we prove that every completely reduced process shows only the "unavoidable" nondeterminism, in the sense given by the following definition.

DEFINITION 5.1 (*Multiple Branches*): A process P has multiple branches if there exists a subterm R of P with $R \equiv R_1 + R_2$ such that $R \sim R_1$; otherwise it is without multiple branches (or wmb for short).

A wmb process shows, in a sense, only unavoidable nondeterminism or, equivalently, offers the minimum amount of redundancy. As an example, consider the process $P = a|b + a.b$. Let $P_1 = a|b$ and $P_2 = a.b$. We have $P = P_1 + P_2 \sim P_1$, and hence P has multiple branches, while P_1 is wmb.

The following proposition shows that completely reduced processes are without multiple branches, hence proving that our parallelization notion satisfies the requirement on *Decrease of Redundancy*.

PROPOSITION 5.6: *Every completely reduced process P is wmb.*

Proof: By contradiction, assume that P has multiple branches. Then, by definition, there exists a subterm R of P with $R \equiv R_1 + R_2$ such that $R \sim R_1$. Therefore, R_1 is a contraction of R in R. Hence R is reducible and P is not completely reduced, contradicting the hypothesis. □

Note that, for a given process P, different contractions could be possible. Some of them lead to wmb processes, while others do not.

Example 5.2: Consider process $a.b + b.a$. Clearly $a|b$ is a contraction of both $a.b + b.a$ and $a.b$ in $a.b + b.a$. We would obtain respectively: i) $a|b$, which is wmb, ii) $a|b + b.a$ which has multiple branches.

5.3. Existence of completely reduced processes

In this section we show that every process P has at least one maximally parallel version. First, we need a preliminary result.

LEMMA 5.1: *Let $S \equiv \sum_{i \in I} P_i$ be not completely reduced, with P_i completely reduced for every $i \in I$ and $|I| \geq 2$. Then $\exists P_1', \exists P_2'$ with $S \equiv P_1' + P_2'$ such that either ii), or iii) or iv) of item 3) of Definition 4.2 does not hold.*

Proof: By induction on $|I|$. The base case is when $|I| = 2$. Consider generic R_1 and R_2 such that $S \equiv R_1 + R_2$. Since in R_1 and R_2 the operator $+$ does not appear at the outermost position, we can only have $R_1 \equiv P_1$ and $R_2 \equiv P_2$ or $R_1 \equiv P_2$ and $R_2 \equiv P_1$. Moreover, since P_1 and P_2 are completely reduced, and $\equiv$ preserves completely reduced processes (Proposition 5.2), we have that also R_1 and R_2 are completely reduced. As S is not completely reduced, either i), ii), iii) or iv) of item 3) of the definition of completely reduced process should not hold. Since for every R_1, R_2 such that $S \equiv R_1 + R_2$, R_1 and R_2 are completely reduced, then i) holds. Necessarily, for R_1 or R_2 one of ii), iii) or iv) does not hold.

Inductively, assume that the statement holds for $|I| < n$; we now prove it for $|I| = n$. By contradiction, assume that for every R_1 and R_2 such that $S \equiv R_1 + R_2$ then all of ii), iii) and iv) of item 3) of the definition of completely reduced process hold. Nonetheless, S is not completely reduced and since every P_i such that $S \equiv \sum_{i \in I} P_i$ is completely reduced, we can only have that there exist R_1 and R_2 such that $S \equiv R_1 + R_2$ and R_1 or R_2 is not completely reduced. W.l.o.g. assume R_1 not completely reduced. Now we have two cases:

1) If the outermost operator of R_1 is not $+$, then R_1 is one of the P_i's. This contradicts the hypothesis that all summands are completely reduced.

2) If the outermost operator of R_1 is $+$, then it is a choice process with less than n summands. Hence, the induction hypothesis can be applied. So, we have that there are R'_1 and R'_2 such that $R_1 \equiv R'_1 + R'_2$ and either ii), iii) or iv) of item 3) of Definition 4.2 does not hold. We distinguish the following three cases:

- Item ii) does not hold. Then $\exists \overline{P}'_3 \not\sim \mathbf{0}$, $\exists \overline{P}'_4 \not\sim \mathbf{0}$ with $\overline{P}'_3$ and $\overline{P}'_4$ completely reduced such that $\overline{P}'_3 | \overline{P}'_4 \sim R'_1 + R'_2 \sim R_1$. But then $S \equiv R_1 + R_2 \sim \overline{P}'_3 | \overline{P}'_4 + R_2$. Thus R_1 in $R_1 + R_2$ would be reducible and item iii) of item 3) of the definition of completely reduced process can be applied to contradict the hypothesis.
- Item iii) does not hold. Then $R'_1 \equiv S_1$ and $\exists \overline{P}'_3 \not\sim \mathbf{0}$, $\exists \overline{P}'_4 \not\sim \mathbf{0}$ with $\overline{P}'_3$ and $\overline{P}'_4$ completely reduced such that $\overline{P}'_3 | \overline{P}'_4 + R'_2 \sim R'_1 + R'_2 \sim R_1$. Now $S \equiv R_1 + R_2 \equiv (R'_1 + R'_2) + R_2 \equiv R'_1 + (R'_2 + R_2) \sim (\overline{P}'_3 | \overline{P}'_4) + (R'_2 + R_2)$. Thus by taking $P'_1 = R'_1$ and $P'_2 = R'_2 + R_2$ we have that $S \equiv P'_1 + P'_2$ and P'_1 can be reduced and item iii) of 3) of the definition of completely reduced process can be used to contradict the hypothesis.

- Item iv) does not hold, i.e., $R'_1 \sim R'_1 + R'_2$. Then $R'_1 + R_2 \sim R'_1 + R'_2 + R_2 \equiv R'_1 + R_2 + R'_2$. But $R'_1 + R_2 + R'_2 \equiv S$. Thus we contradict the hypothesis that item iv) of the definition of completely reduced process holds. □

THEOREM 5.1: *Let P be a $\mathbf{P}$ process. Then there exists $\overline{P}$ completely reduced such that $P \sim \overline{P}$.*

Proof: By induction on the size of P. Assume the statement true for $|P| < n$ and prove it for $|P| = n$. We proceed by case analysis on the structure of P.

a) $P = \mathbf{0}$, then we take $\overline{P} = \mathbf{0}$.

b) $P = a.P_1$. By induction hypothesis (as $|P_1| < n$) there exists $\overline{P}_1$ completely reduced such that $P_1 \sim \overline{P}_1$ and $a.P_1 \sim a.\overline{P}_1$. Now, if $\nexists P'_1 \not\sim \mathbf{0}$, $\nexists P'_2 \not\sim \mathbf{0}$ such that $P'_1|P'_2 \sim a.P_1$ then $a.\overline{P}_1$ is completely reduced and the statement follows. Otherwise, assume $\exists P'_1 \not\sim \mathbf{0}$, $\exists P'_2 \not\sim \mathbf{0}$ such that $P'_1|P'_2 \sim a.P_1$. Note that $|P'_1| > 0$ and $|P'_2| > 0$ because $P'_1 \not\sim \mathbf{0}$ and $P'_2 \not\sim \mathbf{0}$; moreover, $|P'_1| < n$ and $|P'_2| < n$ as $|P'_1|P'_2| = |P'_1| + |P'_2| = 1 + |P_1| = n$. Hence the induction hypothesis can be applied: $\exists \overline{P}'_1$, $\exists \overline{P}'_2$ completely reduced such that $\overline{P}'_1 \sim P'_1$ and $\overline{P}'_2 \sim P'_2$. Also $\overline{P}'_1|\overline{P}'_2 \sim a.P_1$ and $\overline{P}'_1|\overline{P}'_2$ is completely reduced.

c) $P = P_1|P_2$. If one of the two subprocesses (assume w.l.o.g. P_1) is bisimilar to $\mathbf{0}$, then $P \sim P_2$ and we proceed on the structure of P_2. Otherwise, for both P_1 and P_2 the size is less than n. By induction hypothesis, there are $\overline{P}_1$ and P'_2 completely reduced such that $P_1 \sim \overline{P}_1$ and $P_2 \sim \overline{P}_2$. Then, process $\overline{P}_1|\overline{P}_2$ is completely reduced and $\overline{P}_1|\overline{P}_2 \sim P_1|P_2$.

d) $P \equiv \sum_{i \in I} P_i$ with $|I| \geq 2$. Each P_i is in one of the following forms: $\mathbf{0}$, or $a.P'$ or $P'|P''$. In any case, there exists a completely reduced bisimilar term for each P_i by a), or b) or c) above. Hence, we can assume already completely reduced each summand of P. We prove by induction on $|I|$ that there exists $\overline{P}$ completely reduced such that $P \sim \overline{P}$.

Assume $|I| = 2$. There are two cases:

1) $P_1 + P_2$ is completely reduced. Then this is the process we were looking for.

2) $P_1 + P_2$ is not completely reduced. By Lemma 5.1, there exist P'_1, P'_2 such that $P \equiv P'_1 + P'_2$ and ii), iii) or iv) does not hold. Let us consider the three cases in the following order: iv), ii) and then iii),

meaning that one case is explored only if the previous one does hold. If iv) does not hold, then $P_1' \sim P_1' + P_2'$: since P_1' is completely reduced, we have done. Similarly if ii) does not hold: $\exists \overline{P}_3 \not\sim \mathbf{0}$, $\exists \overline{P}_4 \not\sim \mathbf{0}$ with $\overline{P}_3$ and $\overline{P}_4$ completely reduced such that $\overline{P}_3|\overline{P}_4 \sim P_1' + P_2'$. If iii) does not hold, then the situation is a bit more complex. We have that $\exists \overline{P}_3 \not\sim \mathbf{0}$, $\exists \overline{P}_4 \not\sim \mathbf{0}$ with $\overline{P}_3$ and $\overline{P}_4$ completely reduced such that $(\overline{P}_3|\overline{P}_4) + P_2' \sim P_1' + P_2'$. However, we have no guarantee that $(\overline{P}_3|\overline{P}_4) + P_2'$ is completely reduced, even if $\overline{P}_3|\overline{P}_4$ and P_2' are so. In such a bad case, again item ii) or iii) or iv) does not hold. In the first case, this means that also for $P_1' + P_2'$ item ii) does not hold, and so item iii) should have not bee considered. Similarly if item iv) does not hold. In case iii) does not hold for the second time, then we can only have: $\exists \overline{P}_5 \not\sim \mathbf{0}$, $\exists \overline{P}_6 \not\sim \mathbf{0}$ with $\overline{P}_5$ and $\overline{P}_6$ completely reduced such that $(\overline{P}_3|\overline{P}_4) + (\overline{P}_5|\overline{P}_6) \sim (\overline{P}_3|\overline{P}_4) + P_2'$. This produced term is completely reduced. Thus, the statement follows when $|I| = 2$.
Assume true the thesis for $|I| < n$ and prove it for $|I| = n$. Assume $P \equiv \sum_{i \in I} P_i$ not completely reduced. By Lemma 5.1, $\exists P_1'$, $\exists P_2'$ with $P = S \equiv P_1' + P_2'$ such that ii), iii) or iv) of the definition of completely reduced process does not hold. If item ii) fails, then $\exists \overline{P}_3 \not\sim \mathbf{0}$, $\exists \overline{P}_4 \not\sim \mathbf{0}$ with $\overline{P}_3$ and $\overline{P}_4$ completely reduced such that $\overline{P}_3|\overline{P}_4 \sim P_1' + P_2'$. Thus $\overline{P}_3|\overline{P}_4$ is completely reduced and $\overline{P}_3|\overline{P}_4 \sim P$. Similarly if iv) fails, i.e., $P_1' \sim P_1' + P_2'$. Clearly, $P_1' \equiv \sum_{j \in J} P_j$ with $J \subset I$ and hence, since I is finite, $|J| < |I| = n$. By induction hypothesis, there exists $\overline{P}$ completely reduced such that $\overline{P} \sim P_1' \sim P_1' + P_2' \sim P$. And so $\overline{P}$ is the required process. Finally, assume that item iii) fails (if neither ii) nor iv) fails). Then $P_1' \equiv S$ and $\exists \overline{P}_3 \not\sim \mathbf{0}$, $\exists \overline{P}_4 \not\sim \mathbf{0}$ with $\overline{P}_3$ and $\overline{P}_4$ completely reduced such that $(\overline{P}_3|\overline{P}_4) + P_2' \sim P_1' + P_2'$. Of course, $P_1' \equiv \sum_{j \in J} P_j$ with $J \subseteq I$. Among all the possible pairs P_1', P_2' that makes item iii) fail, consider the one which maximizes $|J|$ (i.e., the "maximal" contraction). We have two cases:
1) If $|J| > 1$, then we have that P is equivalent to a new sumform where the number of summands is less than $n : P \sim (\overline{P}_3|\overline{P}_4) + P_2'$. Therefore, by induction hypothesis, there exists $\overline{P}$ completely reduced such that $\overline{P} \sim (\overline{P}_3|\overline{P}_4) + P_2' \sim P$.
2) If $|J| = 1$, then P_1' is necessarily of the form $a.\overline{P}_1'$. This means that $(\overline{P}_3|\overline{P}_4) + P_2'$ is a bisimilar form of P with the same number of summands, but with one sequential addend $a.\overline{P}_1'$ replaced by a parallel addend $\overline{P}_3|\overline{P}_4$, which is also completely reduced. Hence $P \sim \sum_{h \in H} P_h$, where $|H| = |I|$ and all the P_h are completely

reduced. Thus we have found a new term which is in the same theorem hypothesis of P, except for the fact that one addend – namely $\overline{P}_3|\overline{P}_4$ – cannot be replaced by a more parallel version. Then we can try to apply several times the reduction above [1] until one of the following happens:
2.1) for the obtained term item iii) does hold. What we gain is the completely reduced term we were looking for [2].
2.2) no more summands of the form $a.\overline{P}'_h$ occur in the obtained term, but item iii) fails again. This is not possible as – for item iii) to fail – we need at least two summands in this case, contradicting the hypothesis of having chosen the maximal contraction the first time. □

6. ON THE RELATIONSHIP BETWEEN REDUCTION AND FACTORIZATION

In this section we compare completely reduced processes and factorized processes. We will prove that every completely reduced process is factorized and so are its subterms.

First of all, we want to relate irreducible terms and prime terms. Consider process $\mathbf{0}$: this is irreducible but not prime. Further, consider a factorized term $P \equiv P_1|...|P_n$; obviously, we have P irreducible but not prime (provided that $n > 1$). The cases discussed above outline an important difference between prime and irreducible terms. In general, a prime term is a term that cannot be decomposed into simpler, non empty processes; differently, an irreducible term is a term that cannot be rewritten to obtain a more deterministic or more parallel one.

PROPOSITION 6.1: *Let S be a non prime, choice process. If $S \not\sim \mathbf{0}$, then S is reducible.*

Proof: By the fact that S is not prime and is not (bisimilar to) $\mathbf{0}$, there are $R_1, R_2 \not\sim \mathbf{0}$ such that $S \sim R_1|R_2$. By Theorem 5.1 there are R_3, R_4 completely reduced such that $R_3 \sim R_1$, $R_4 \sim R_2$. Hence, $S \sim R_1|R_2 \sim R_3|R_4$, with $R_3|R_4$ completely reduced.

The reverse of the above proposition could be something like: "If $S \not\sim \mathbf{0}$ is reducible, then S is non prime". This is false as $a+\mathbf{0}$ is reducible but prime.

(1) Remember that only one (prefixed) addend can be transformed into a parallel addend, as we have assumed to choose P'_1, P'_2 that maximazes $|J|$, if we started with $|J| = 1$, then all the following cannot have a larger size

(2) Remind that items ii) and iv) cannot be applicable, as we were considering item iii) only if both ii) and iv) failed, moreover, in no step we introduce a possibility for ii) or iv) to be applicable again

PROPOSITION 6.2: *Let S be a completely reduced, choice process. If $S \not\sim \mathbf{0}$, then S is prime.*

Proof: We need to prove that if S is decomposable as $P_1|P_2$ then either $P_1 \sim \mathbf{0}$ or $P_2 \sim \mathbf{0}$. By ii) of item 2) of Definition 4.2, if S is in prefix form, then S is prime. By ii) of item 3) of Definition 4.2, if S is in sumform, then S is prime.

From the fact that $\mathbf{0}$ is completely reduced, it follows immediately that, given P factorized and completely reduced, $P|\mathbf{0}$ is still completely reduced, even if it is no more factorized. Hence, it is not true, in general, that a completely reduced term is also factorized. However, this implication is false only for this trivial case. We first need to introduce the notion of cleaned process. Let us call $clean(P)$ the process P where the occurrence of $\mathbf{0}$ (in general, the sequential subprocesses bisimilar to $\mathbf{0}$) are removed if they are argument of a parallel composition operator. For instance, if $P \equiv (a + \mathbf{0})|\mathbf{0}|b.(c|\mathbf{0})$ then $clean(P) \equiv (a + \mathbf{0})|b.c$.

PROPOSITION 6.3: *Let P be a completely reduced process. Then* clean(P) *is factorized.*

Proof: Consider a generic completely reduced process $P = P_1|...|P_n$ where each P_i does not contain a parallel composition as the outermost operator. Then we have $clean(P) = \prod_{i \in I} clean(P_i)$, where $I \subseteq \{1, ..., n\}$ contains the indexes of the P_i's not bisimilar to $\mathbf{0}$. Moreover, every $clean(P_i)$ is a completely reduced, choice term. Thus, every $clean(P_i)$ is prime by Proposition 6.2, and so $clean(P)$ is factorized.

We complete the comparison with the following proposition.

COROLLARY 6.1: *The cleaned version of every subterm of a completely reduced process is factorized.*

Proof: Every subterm of a completely reduced process is, by definition, a completely reduced process. Thus, by Proposition 6.3, its cleaned version is also factorized. □

The vice versa of Proposition 6.3 and Corollary 6.1 could be something like: "If all the subterms of a factorized process P are factorized, then P is completely reduced". Unfortunately, this does not hold in general. Consider $P = a|b|c + acb + cab$ and note that each of its subterms is decomposed as the parallel composition of primes, even if P is not completely reduced; see Example 4.1.

7. AN ABSTRACT REDUCTION SYSTEM FOR PROCESSES

The definition of contracting given in Section 4 will be used to define an abstract reduction system for processes (see [Klo90]). We will prove that it is strongly normalizing and that its normal forms are exactly the completely reduced processes.

DEFINITION 7.1 (*Reduction Relation*): Let $\Rrightarrow$ be the least binary relation over $\mathbf{P}$ defined by the following rules:

1. $P \Rrightarrow R$ if $P \equiv S$ and R is a contraction of S in S
2. $P \Rrightarrow R + Q$ if $P \equiv S + Q$ and R is a contraction of S in $S + Q$
3. $\alpha.P \Rrightarrow \alpha.P'$ if $P \Rrightarrow P'$
4. $P + Q \Rrightarrow R$ if $P + Q \equiv P_1 + P_2$ and $P_1 \Rrightarrow P_1'$ and $R \equiv P_1' + P_2$
5. $P|Q \Rrightarrow R$ if $P|Q \equiv P_1|P_2$ and $P_1 \Rrightarrow P_1'$ and $R \equiv P_1'|P_2$

The reduction relation $\Rrightarrow$ given in Definition 7.1 together with $\mathbf{P}$ defines an abstract reduction system. It will be denoted by $\langle \mathbf{P}, \Rrightarrow \rangle$. Relation $\Rrightarrow$ defines also a partial ordering on processes: $Q \sqsubseteq P$ if and only if $P \Rrightarrow^* Q$, where $\Rrightarrow^*$ is the reflexive and transitive closure of the reduction relation. Relation $\sqsubseteq$ is indeed a partial ordering as $\Rrightarrow$ is acyclic, as we will prove later on.

PROPOSITION 7.1: *$\Rrightarrow$ is sound with respect to $\sim$.*

Proof: By induction on the depth of the proof of $P \Rrightarrow P'$. The base cases are rules 1 and 2, for which we can resort to Proposition 5.3. All the other cases follows by induction hypothesis and by the congruence property of $\sim$.

A normal form of the reduction system $\langle \mathbf{P}, \Rrightarrow \rangle$ is a process P such that there exists no Q for which $P \Rrightarrow Q$. Note that, by Proposition 7.1, if $P \Rrightarrow^* Q$ and Q is a normal form, then $P \sim Q$. Hence, if the normal forms are the completely reduced processes, this proposition offers an alternative way to prove that the first requirement on equivalence preservation is met. Indeed, we have the following:

PROPOSITION 7.2: *A process P is a normal form if and only if it is completely reduced.*

Proof: Assume P is a normal form. Then there exists no Q such that $P \Rrightarrow Q$. This means that rules 1, 2, 3, 4 and 5 can never be applied. It follows that there is no subterm P' of P which is reducible. By Proposition 4.1, P is completely reduced. Conversely, assume P completely reduced. Again by Proposition 4.1, none of its subterms is reducible; thus, we never can apply rules 1 and 2 above, hence neither can apply it in any context (rules 3, 4 and 5). □

We want to prove formally that the abstract reduction system $\langle \mathbf{P}, \Rightarrow \rangle$ is strongly normalizing. To do this we need some new notation and results. A reduciton sequence is a sequence $P_1 \Rightarrow P_2 \ldots \Rightarrow P_n \ldots$ of reductions $P_k \Rightarrow P_{k+1}$ for $k \in \mathbf{N}$. The next lemma states an obvious fact.

LEMMA 7.1: *Let P be a* **P** *process such that every reduction sequence is finite and $P \Rightarrow \overline{P}$. Then $\overline{P}$ has only finite reduction sequences.*

Now we call *inner* those reduction sequences where each contraction has always been applied to one single summand.

DEFINITION 7.2: Let $Q_k \equiv \sum_{i \in I} P_i^k$ for $k \in \mathbf{N}$. A reduction sequence $Q_0 \Rightarrow Q_1 \Rightarrow Q_2 \ldots$ is inner if for each k we have that

1) $P_i^k \Rightarrow P_i^{k+1}$ for some $i \in I$;

2) $P_l^k \not\Rightarrow P_l^{k+1}$ for every $l \in I$ and $l \neq i$.

LEMMA 7.2: *Let $S \equiv \sum_{i \in I} P_i$ such that $|I| \geq 2$ and every P_i has only finite reduction sequences. Then every inner reduction sequence from S is finite.*

Proof: It trivially follows from the fact that an inner reduction sequence from S is an "interleaving" of the (finite) reduction sequences from the (finite) summands of S. □

The lemma below generalizes the previous result to all reduction sequences.

LEMMA 7.3: *Let $S \equiv \sum_{i \in I} P_i$ such that $|I| \geq 2$ and every P_i has only finite reduction sequences. Then every reduction sequence from S is finite.*

Proof: Let r be the number of P_i's with $i \in I$ of the form $a.P_i'$. The proof is by induction on $r + |I|$.

The base case is when $r + |I| = 2$, with $r = 0$ and $|I| = 2$ because S has at least two summands. Hence, neither P_1 nor P_2 is in prefix form. By Lemma 7.2 every inner reduction sequence from S is finite. So, we focus on non inner reduction sequences. After (zero or more) steps of an inner reduction sequence from S, we reach a process $Q_k = P_1^k + P_2^k$ such that there exists a reduction involving both summands. Let $S \Rightarrow Q_1 \Rightarrow Q_2 \Rightarrow Q_3 \ldots \Rightarrow Q_k \Rightarrow \overline{P}$ be such a reduction sequence, where $Q_k \Rightarrow \overline{P}$ is the first non inner reduction, which involves both summands. As the reduction sequence is inner until Q_k, both P_1^k and P_2^k have finite reduction sequences by Lemma 7.1. Now if a contraction is applied to the whole Q_k, then rule 1 of the reduction relation has been used. Moreover, if $\overline{P}$ is a contraction of $P_1^k + P_2^k$, only i) or iii) of Definition 4.3 can be applied. Note that i) reduces $P_1^k + P_2^k$ to the parallel composition of completely reduced processes, and so the reduction sequence ends with $\overline{P}$; similarly, item iii) reduces $P_1^k + P_2^k$ to P_1^k, that has finite reduction sequences. Summing up, in all the cases, a reduction sequence is finite.

Assume now the thesis hold for $r + |I| < n$ and prove it for $r + |I| = n$ (with $n > 2$). By Lemma 7.2 every inner reduction sequence from S is finite. So, we focus on non inner reduction sequences. After (zero or more) steps of an inner reduction sequence from S, we reach a process $Q_k = P_1^k + P_2^k$ such that there exists a reduction which involves either more than one summand, or only one in a +-context such that rule 2) is applicable. Let $Q_k \Rightarrow \overline{P}$ be such a non inner reduction. Assume that rule 1 was applied; then, $\overline{P}$ is a contraction of Q_k in Q_k. Thus, only i) or iii) of Definition 4.3 can be applied. Note that i) reduces $P_1^k + P_2^k$ to the parallel composition of completely reduced processes, and so the reduction sequence ends with $\overline{P}$; similarly, item iii) reduces $P_1^k + P_2^k$ to P_1^k, that has finite reduction sequences. Assume instead that rule 2 was applied for $Q_k \Rightarrow \overline{P}$; then, $Q_k \equiv S' + Q'$, $\overline{P} = R' + Q'$ and R' is a contraction of S' in Q_k. Then only item ii) of Definition 4.3 can be applied. Depending on the form of S', this contraction reduces either a choice process $S_1 + S_2$ as the parallel composition of completely reduced processes, or the number of processes of the form $a.P'$ decreases. In both cases, for the produced term $\overline{P}$ the number of summands in prefix form plus the total number of summands is decreased w.r.t. the corresponding value for Q_k. Hence, induction hypothesis is sufficient to prove that $\overline{P}$ has finite reduction sequences. □

PROPOSITION 7.3: *Let P be a* $\mathbf{P}$ *process. Then, every reduction sequence from P is finite.*

Proof: By induction on the syntactic structure of P.

a) $P = \mathbf{0}$ then $P \not\Rightarrow$.

b) $P = a.P'$. By induction, every reduction sequence for P' is finite. Hence, applying rule 3, we can obtain only finite reduction sequences for $a.P'$. The only other possible reduction for $a.P'$ is by rule 1, which gives a completely reduced process in one step.

c) $P = P' + P''$. By induction hypothesis, both P' and P'' have finite reduction sequences only. By Lemma 7.3 also P has only finite reduction sequences.

d) $P = P'|P''$. By induction hypothesis, every reduction of both P' and P'' is finite. The statement follows by observing that any reduction sequence of P is – roughly speaking – the "interleaving" of reductions sequences of P' and P'', respectively. □

COROLLARY 7.1: *Reduction system* $\langle \mathbf{P}, \Rightarrow \rangle$ *is strongly normalizing.*

We will see later an example of a process which admits more than one normal form. Refer to Example 8.2. Finally, another obvious consequence is that the reduction relation is acyclic, hence:

COROLLARY 7.2: $\sqsubseteq$ *is a partial ordering on* $\mathbf{P}$.

8. UNIQUE NORMAL FORMS

This section addresses the problem of finding conditions under which normal forms are unique, up to the structural congruence. As we want to prove this result independently of the definition of the abstract reduction system for processes given in Section 7, new definitions and results are needed.

DEFINITION 8.1: Let P, Q be processes such that $P \sim Q$. We say that P and Q are structural bisimilar (denoted $P \cong Q$) if one of the following holds:

i) $P \equiv Q \equiv \mathbf{0}$;

ii) $P \equiv \alpha.P'$, $Q \equiv \alpha.Q'$ and $P' \sim Q'$;

iii) $P \equiv \sum_{i \in I} P_i$, $Q \equiv \sum_{i \in I} Q_i$ and $P_i \sim Q_i$ for all $i \in I$;

iv) $P \equiv \prod_{i \in I} P_i$, $Q \equiv \prod_{i \in I} Q_i$ and $P_i \sim Q_i$ for all $i \in I$.

Recall that in (iii) above none of the P_i, Q_i is a sum of processes; and also in (iv) above none of the P_i, Q_i is a parallel composition of processes. The following holds trivially.

LEMMA 8.1: $P \equiv Q$ *implies* $P \cong Q$.

However, the reverse does not hold, as shown by the following example.

Example 8.1: Given $P = a.(b|c)$ and $Q = a.(b.c + c.b)$, we have that $P \cong Q$ but $P \not\equiv Q$.

Intuitively, structural bisimilarity implies structural congruence whenever the former is inductively required under all possible subterms. If we could prove that bisimilar, completely reduced terms are structurally bisimilar, we would also have that they are in the relation $\equiv$. Hence, since subterms of completely reduced terms are also completely reduced, the problem of uniqueness of normal forms could be equivalently stated in terms of structural bisimilarity. The following (meta-)theorem formally defines the above.

LEMMA 8.2: *Let $\mathcal{C}$ be a property over* $\mathbf{P}$ *such that* $\forall P,\ P' \in \mathbf{P},\ P' \leq P$ *and* $\mathcal{C}(P)$ *imply* $\mathcal{C}(P')$. *Let* P, Q *be processes and consider the following two statements:*

1) $\forall P,Q$ *such that* $P \sim Q, \mathcal{C}(P)$ *and* $\mathcal{C}(Q)$ *imply* $P \cong Q$;

2) $\forall P,Q$ *such that* $P \sim Q, \mathcal{C}(P)$ *and* $\mathcal{C}(Q)$ *imply* $P \equiv Q$.

Then $\mathcal{C}$ satisfies condition 1) if and only if it satisfies condition 2).

Proof: Let $\mathcal{C}$ be a property for which 1) holds. Let P, Q be such that $P \sim Q$ and $\mathcal{C}(P)$, $\mathcal{C}(Q)$. We prove that $P \equiv Q$ by induction on the structure of P.

- $P \equiv 0$. Clearly $Q \equiv P$.
- $P \equiv \alpha.P'$. By 1) follows $Q \equiv \alpha.Q'$ with $P' \sim Q'$. By hypothesis $\mathcal{C}(P')$ and $\mathcal{C}(Q')$, and hence by induction hypothesis $P' \equiv Q'$, and also $P \equiv Q$.
- $P \equiv \sum_{i \in I} P_i$. By item 1) follows $Q \equiv \sum_{i \in I} Q_i$ such that $P_i \sim Q_i$ for all $I \in I$. By hypothesis we have that $\mathcal{C}(P_i)$ and $\mathcal{C}(Q_i)$ for every $i \in I$. Induction hypothesis ensures that $P_i \equiv Q_i$, and also $P \equiv Q$.
- $P \equiv \prod_{i \in I} P_i$. Similar to the previous case.

Now we prove the other implication by contradiction. Assume that $\mathcal{C}$ does not hold 1) for some pair of bisimilar processes P and Q. We prove that $P \not\equiv Q$. If $\mathcal{C}$ does not satisfy 1), then we have that $P \sim Q$, $\mathcal{C}(P)$ and $\mathcal{C}(Q)$ but P and Q that are not structurally bisimilar. Hence, by Lemma 8.1, $P \not\equiv Q$.

Consider the following property

$$\mathcal{C}_1(P) = (P \text{ is completely reduced})$$

and note that by proposition 5.4, $\mathcal{C}_1(P)$ satisfies the hypothesis of Lemma 8.2. Unfortunately, it does not satisfy 1); indeed bisimilar and completely reduced processes not necessarily are also structural bisimilar as the following example shows.

Example 8.2: Let P be the process

$$a.(c+d)+b.(c+d)+c.(a+d)+d.(a+b)+a.c+c.a+a.d+d.a+b.c+c.b+b.d+d.b.$$

It is easy to convince that both processes

$$Q \equiv a|(c+d) + b|(c+d) + c|(a+b) + d|(a+b)$$

and

$$R \equiv (a+b)|(c+d) + a|c + a|d + b|c + b|d$$

are bisimilar to P. However, both Q and R are completely reduced.

In order to obtain a unique normal form for processes, we single out some properties that ensure condition 1) of Lemma 8.2, together with its premises. First of all, consider the subset of sum-free concurrent processes, that is the terms built with prefixing and parallel composition only. Let

$$\mathcal{C}_2(P) = (P \text{ is sum-free and completely reduced}).$$

The following proposition shows that $\mathcal{C}_2$ satisfies both 1) and the hypothesis above.

PROPOSITION 8.1: *If $P \sim Q$ and $\mathcal{C}_2(P)$ and $\mathcal{C}_2(Q)$, then $P \cong Q$. Moreover, $\mathcal{C}_2(P)$ implies $\mathcal{C}_2(P')$ for every $P' \leq P$.*

Proof: We prove only the first part of the statement, as the second one is trivial. Consider P, Q be completely reduced processes with $P \sim Q$. The proof proceeds by case analysis. If $P \equiv \alpha.P'$, then necessarily, it has to be $Q \equiv \alpha.Q'$. Otherwise, we could have for instance $Q \equiv \prod_i Q_i$ contradicting the hypothesis that P is completely reduced. Similarly if $P \equiv \prod_i P_i$ then Q is forced to have parallel composition as its outermost operator, and by Milner and Moller's unique factorization theorem, it is possible to prove that the primes of P and Q are pairwise bisimilar. □

Now we want to introduce another suitable property, this time on the whole language; it states that a process P has a unique normal form if any state it reaches has never more than two possible alternatives. Let

$$\mathcal{C}_3(P) = (P \text{ is completely reduced and has at most two choices}).$$

We first need to formalize the condition on the number of choices.

DEFINITION 8.2: Let P be a process. Define $succ(P) = \{(\alpha, [P']_\sim) : P \xrightarrow{\alpha} P'\}$, where $[P]_\sim$ is the congruence class of P w.r.t. $\sim$.

DEFINITION 8.3: Let P be a process and $n \in \mathbf{N}$. We say that P has at most n choices if for every P', $s \in Act^*$ such that $P \xrightarrow{s}{}^* P'$ is $|succ(P')| \leq n$.

PROPOSITION 8.2: *$\mathcal{C}_3$ satisfies condition 1) of Lemma 8.2 and its hypothesis.*

Proof: We first prove the hypothesis of the Lemma, that is $C_3(P)$ implies $C_3(P')$ for every $P' \leq P$. Case $P = 0$ is trivial. Consider case $P \equiv \alpha.P'$. If P' would have more two choices then the same holds for P contradicting the hypothesis. Consider case $P \equiv \sum_{i \in I} P_i$ and note that $|succ(P)| \geq |succ(\sum_{j \in J} P_j)|$ for every $J \subseteq I$. Similarly if $P \equiv \prod_{i \in I} P_i$.

Now let us prove condition 1). Assume $P \sim Q, C_3(P)$ and $C_3(Q)$. Cases $P \equiv 0$, $P \equiv \alpha.P'$ and $P \equiv \prod_i P_i$ are similar to those in Proposition 8.1. Take case $P = \sum_{i \in I} P_i$. This forces $|I| = 2$; indeed if $|I| > 2$ we have two possible cases to distinguish: either P has not at most two choices, or P is not completely reduced because it has multiple branches. Both of them contradict the hypothesis. Thus $P \equiv P_1 + P_2$. If Q would be of the form $Q \equiv \alpha.Q'$ then P would not be completely reduced. If Q would be of the form $Q \equiv \prod_j Q_j$ then P would be reducible and hence not completely reduced. Thus Q is forced to be of the form $Q \equiv Q_I + Q_2$ (similar reasonings of P hold also for Q). As each of the P_i's and of the Q_i's can offer exactly one choice (otherwise either P and Q have extra branches or they would not be completely reduced), it is possible to set the required bijection among them. □

9. CONCLUSION AND FURTHER RESEARCH

We have addressed the problem of finding the maximally parallel version of a given process from a theoretical point of view. This means that in our investigation we have not bound the number of available processors which the parallel components can be mapped onto. This is also what we need in the case of extending our work to recursive processes. To explain why, let us consider process P that repeats the same action a forever: $P = a.P$. The same computations of P can be performed by $P|P$, or by $P|P|P$, or even with an infinity of finite processes, each executing one single action a. It is clear that this forms a chain of processes, each to be preferred to the previous one because it increases distribution and efficiency. Hence, if one wants to maximize parallelism for infinite processes, one has to cope with infinite terms. Further study should be devoted to a constrained version of this problem, where the maximum number of available processors (i.e., number of parallel components) is fixed *a priori*; a solution to such a problem could be useful practically, also in case of recursion.

The problem of finding the maximally parallel version of a given process has not been widely studied up to now. Maybe this is due to the fact that, despite of its expected intuitive simplicity, the parallelization of a process is actually a delicate notion that can be influenced by several parameters as shown in [Mar96]. These have to do mainly with the interplay between increase of parallelization and decrease of non-determinism. In particular, we show that these two are conficiting requests! For instance, consider agent $P = (a.a.b + c)|b$. It is easy to see that P is completely reduced. Nonetheless, the non completely reduced agent $Q = P + a.b|a.b$ may be, intuitively, faster in some cases as the two occurrences of action a are not sequentialized. In other words, Q is faster but has multiple branches, while P is wmb but at the price of being, in some cases, slower. The definition have presented here is the one we have found more intuitively appealing; see [Mar96] for a detailed discussion on possible variations.

Concerning the requirement of *Increase of distribution/efficiency*, we would like to prove that the most parallel version Q of a process P is more distributed and more efficient than P. And moreover, that there is no way to do better (at least, under some circumstances). Of course, this means that we need to introduce suitable truly concurrent semantics which express distribution and timing information. These are crucial criteria for assessing the merits of a notion of parallelization; hence, the problem we are facing sheds new light on the role of truly concurrent semantics, which should be used to check structural properties of systems, rather than their functionality (already expressed by the interleaving semantics). Indeed, in [Mar96] our parallelization preorder is compared with the location preorder [BCHK92] and with the performance preorder [CGR95, CGR97]. These two preorders are, as ours, bisimulation-based and interleaving bisimulation equivalence preserving. These comparisons are not included here because they require an extensive treatment which goes outside the scope of this introductory paper.

The language we have used is certainly simple. Besides the extension to recursion (already discussed), we would like to mention the problems arising when communication and estriction are included. Consider the completely reduced process $P = a.a.b$. However, if we allow communication, $P \sim Q = (a.c.b)|(a.\bar{c}) \setminus c$, which is more parallel than P. Nonetheless, besides the technical problem of singling out such communications, it is not clear that the increase in distribution will cause an increase of efficiency as communication is usually a costly operation. Furthermore, the presence of unobservable communications (as in CCS), forces to choose a different underlying equivalence, e.g. weak bisimulation.

REFERENCES

[BCHK92] G. Boudol, I. Castellani, M. Hennessy and A. Kiehn, A Theory of Processes with Localities, *Formal Aspects of Computing*, 1992, *6*, pp. 165-200.

[CGR95] F. Corradini, R. Gorrieri and M. Roccetti, Performance Preorder: Ordering Processes with Respect to Speed. In the Proceedings of MFCS'95, Springer-Verlag LNCS, 1995, *969*, pp. 444-453.

[CGR97] F. Corradini, R. Gorrieri and M. Roccetti, Performance Preorder and Competitive Equivalence, *Acta Informatica*, November 1997, Vol. 34, (11).

[Klo90] J. W. Klop, Term Rewriting System: A Tutorial, Bulletin of EATCS, 1990.

[Mar96] D. Marchignoli, Decomposizione Parallela di Processi Concorrenti, Master Thesis (in Italian), University of Bologna, March 1996.

[Mil89] R. Milner, *Communication and Concurrency*, Prentice Hall, Englewood Cliffs, NJ, 1989.

[MM93] R. Milner and F. Moller, Unique decomposition of processes, *Theoretical Computer Science*, 1993, *107*, pp. 357-363.