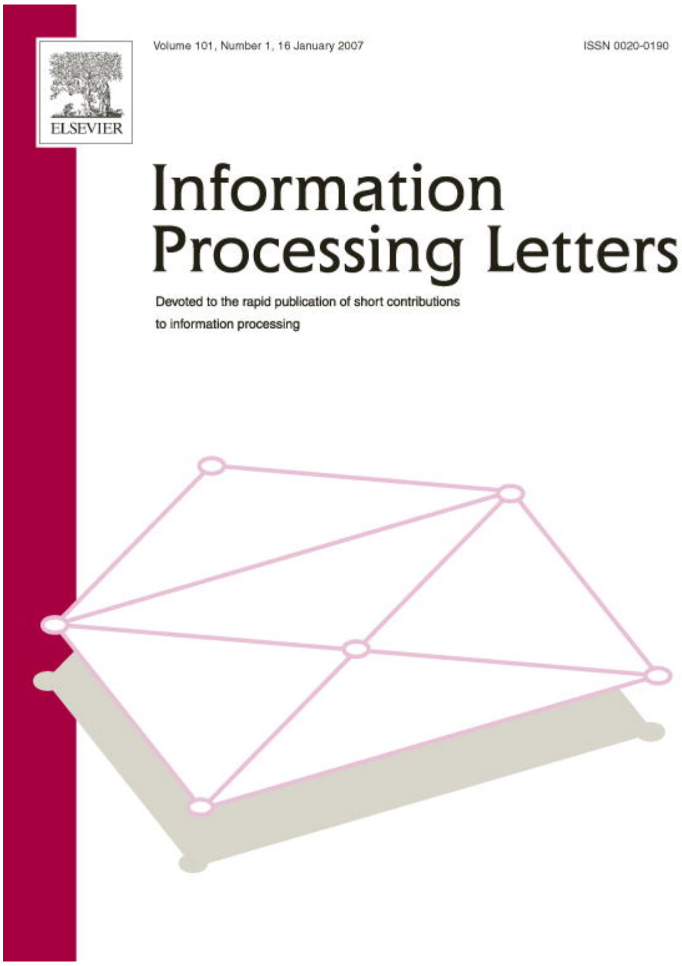

Volume 101, Number 1, 16 January 2007
ISSN 0020-0190

Information
Processing Letters

Devoted to the rapid publication of short contributions
to information processing

Available online at www.sciencedirect.com
ScienceDirect
ELSEVIER
Information Processing Letters 101 (2007) 1–5
Information Processing Letters
www.elsevier.com/locate/ipl

An exercise in structural congruence

Joost Engelfriet [*], Tjalling Gelsema

LIACS, Leiden University, P.O. Box 9512, 2300 RA Leiden, The Netherlands

Received 14 May 2006; received in revised form 1 August 2006; accepted 1 August 2006

Communicated by J.L. Fiadeiro

Abstract

Milner's structural congruence is decidable for the pi-calculus without restriction, in exponential space.

Keywords: Concurrency; Pi-calculus; Petri net; Commutative semigroup

0. Introduction

In the most basic and elegant version of the π-calculus, introduced by Milner in [13] and incorporated in [14,17], replication is viewed as a structural rather than behavioural operation on processes. The structure of a replicated process $!P$ is determined by the structural law $!P \equiv !P \mid P$ which unfolds $!P$ into 'an unlimited number of copies of P able to run concurrently' [14, p. 90]. Clearly, any notion of structure that satisfies this law is necessarily infinite. This raises the question [14, Exercise 9.15] whether the resulting structural congruence relation is decidable, i.e., whether it is decidable whether or not two π-calculus processes have the same structure.

Despite the fact that [14] considers the decidability of structural congruence an exercise (for 'experienced logicians', admittedly), to our knowledge this question is still open. In [6,7] we have shown that adding four new structural laws, the resulting 'extended congruence' is decidable (see [10] for an alternative proof, with one law less, and see [5] for a similar result for the ambient calculus [4]). It should be noted that this addition does not change the behaviour of processes, only their structure. In [8] the same result was shown for 'middle congruence', which is much closer to the original congruence of [13].

Due to the existence of these decidable versions of structural congruence, the original version seems to be of historical (and recreational) interest only. For instance, extended congruence was taken as the basis for the semantics of the spatial logic of [3].

In the present paper, as an exercise, we prove the decidability of Milner's structural congruence for a special case, viz. for the π-calculus without restriction. Essentially, we solve the word problem for a free commutative semigroup with a replication operation, as suggested in [15] (where, by definition, a replication operation is a unary operation ! satisfying $!x = !x \circ x$, for the semigroup operation $\circ$). We reduce this problem to the reachability problem for Petri nets, which was shown to be decidable in [11] (see also [16]). More precisely, we reduce it to the reachability problem for *reversible*

---

[*] Corresponding author.

E-mail addresses: engelfri@liacs.nl (J. Engelfriet), tgsa@cbs.nl (T. Gelsema).

0020-0190/$ – see front matter © 2006 Elsevier B.V. All rights reserved.
doi:10.1016/j.ipl.2006.08.001

Petri nets, which is the same as the uniform word problem for (finitely presented) commutative semigroups. As shown in [12], the uniform word problem for commutative semigroups is complete in exponential space. The existence of an exponential space algorithm was essentially already known in the twenties of the 20th century, as explained in [12] (the algorithm is based on techniques from linear algebra to solve linear equations over the ring of polynomials with rational coefficients). It leads to an exponential space algorithm for deciding Milner's structural congruence, for the π-calculus without restriction.

The result of this paper, without the complexity bound, was obtained independently by Sobociński [18]. In the unpublished technical report [9] we have shown the decidability of a larger subset of the π-calculus of [13], but without the complexity bound and with a more complicated proof.

We note that this paper does not rely on the concepts and results of [6–8], where Petri nets are used (as, e.g., in [1,2]) to express the full behaviour of π-calculus processes rather than just their structure.

1. Composition and replication

We start with a version of the π-calculus that has no restriction, no guarding, and no null process. Guarding and the null process will be added in the next section. Instead of guarding, this version of the π-calculus has 'actions'.

The set of *actions* is denoted by $\mathbf{A}$. In this section, a *process* is a term obtained by the syntactical description

$$P ::= a,\ \ P \mid P,\ \ !P$$

where a is an action from $\mathbf{A}$, $P \mid Q$ is the parallel composition of the processes P and Q, and $!P$ is the replication of P.

By $\equiv$ we denote the smallest congruence satisfying the *structural laws*

$$P \mid Q \equiv Q \mid P,$$
$$P \mid (Q \mid R) \equiv (P \mid Q) \mid R \quad \text{and}$$
$$!P \equiv\, !P \mid P.$$

Thus, $P \equiv Q$ if and only if $P \to^* Q$, where $P \to Q$ means that Q is obtained from P by the application of one of the structural laws (in either direction) to a subterm of P, and where, as usual, $\to^*$ is the transitive reflexive closure of $\to$. The first two laws will be called the *composition laws*, and the last one the *replication law*. Congruence $\equiv$ will also be referred to as *Milner's structural congruence*. Note that, modulo $\equiv$, the processes form the free commutative semigroup with a replication operation (as discussed in the introduction) generated by the set $\mathbf{A}$ of actions, with parallel composition as the semigroup operation.

Any process obtained from $(\cdots((P_1 \mid P_2) \mid P_3) \mid \cdots \mid P_{m-1}) \mid P_m$ by applying the second composition law (of associativity of composition) will be denoted $P_1 \mid \cdots \mid P_m$. We will say that a process is a *web* if it is not of the form $Q \mid R$, i.e., it is an action or a replication. Thus, every process P is uniquely of the form $P = P_1 \mid \cdots \mid P_m$, $m \geqslant 1$, with P_i a web for all i: the *webform* of P.

For a process P, we let $\rho(P)$ be the *replication depth* of P, i.e., the nesting depth of replications in P: $\rho(a) = 0$, $\rho(P \mid Q) = \max(\rho(P), \rho(Q))$, and $\rho(!P) = \rho(P) + 1$. It is easy to see that congruent processes have the same replication depth, i.e., if $P \equiv Q$ then $\rho(P) = \rho(Q)$.

For a set $\mathcal{R}$ of processes, we denote by $\equiv_{\mathcal{R}}$ the smallest congruence satisfying the structural laws, but such that the replication law is valid for $P \in \mathcal{R}$ only. The one-step relation $\to_{\mathcal{R}}$ is defined as above.

For a process P, we define $\mathrm{desc}(P) = \{R \mid\, !R$ is a subterm of $P\}$: the set of *descendants* of P. We will need the following two lemmas, which hold for all processes P and Q, and every set $\mathcal{R}$ of processes.

Lemma 1. *If $P \equiv_{\mathcal{R}} Q$ and $T \in \mathrm{desc}(Q)$, then there exists $S \in \mathrm{desc}(P)$ such that $S \equiv_{\mathcal{R}} T$.*

Proof. Clearly, if $P \to_{\mathcal{R}} Q$ and $T \in \mathrm{desc}(Q)$, then either $T \in \mathrm{desc}(P)$ or there exists $S \in \mathrm{desc}(P)$ such that $S \to_{\mathcal{R}} T$. □

Lemma 2. *$!P \equiv_{\mathcal{R}}\, !Q$ if and only if $P \equiv_{\mathcal{R}} Q$.*

Proof. Assume that $!P \equiv_{\mathcal{R}}\, !Q$. Note that $\rho(P) = \rho(Q)$. Since Q is a descendant of $!Q$, there is a descendant S of P such that $S \equiv_{\mathcal{R}} Q$, by Lemma 1. Hence $\rho(S) = \rho(Q) = \rho(P)$. Since P is the only descendant of $!P$ with the same replication depth as P, $S = P$ and $P \equiv_{\mathcal{R}} Q$. □

We will show the decidability of $P \equiv Q$ in exponential space by proving first that, for finite $\mathcal{R}$, $P \equiv_{\mathcal{R}} Q$ is decidable in exponential space, and then that $P \equiv Q$ if and only if $P \equiv_{\mathrm{desc}(P)} Q$. In the first proof we use the notion of a Petri net, which can be viewed as a vector addition system (see, e.g., [16]) or as a multiset transition system (see, e.g., [6]). A *multiset* v is a finite set D_{v}, the underlying set of v, together with a mapping $\phi_{\mathsf{v}} : D_{\mathsf{v}} \to \{1, 2, 3, \ldots\}$ that defines the multiplicity of the elements of D_{v} in v. By convention, we let $\phi_{\mathsf{v}}(x) = 0$

for $x \notin D_{\mathsf{v}}$. The *empty* multiset $\emptyset$ satisfies $D_\emptyset = \emptyset$ and $\phi_\emptyset = \emptyset$. The *union* of multisets is obtained by adding the multiplicity of each element: $\mathsf{v}_1 \cup \mathsf{v}_2$ is the multiset with $D_{\mathsf{v}_1 \cup \mathsf{v}_2} = D_{\mathsf{v}_1} \cup D_{\mathsf{v}_2}$ and $\phi_{\mathsf{v}_1 \cup \mathsf{v}_2}(d) = \phi_{\mathsf{v}_1}(d) + \phi_{\mathsf{v}_2}(d)$ for every $d \in D_{\mathsf{v}_1 \cup \mathsf{v}_2}$. For a finite set D, v is a multiset *over* D if $D_{\mathsf{v}} \subseteq D$. Note that such a v can be viewed as an n-dimensional vector of nonnegative integers, where n is the cardinality of D, and that union is just vector addition. Note also that the set of nonempty multisets over D is the free commutative semigroup generated by the elements of D, with union as the semigroup operation. We use the Parikh mapping Φ to turn a sequence into a multiset, as usual, e.g., $\Phi(b, b, c, a, b, a)$ is the multiset v with $D_{\mathsf{v}} = \{a, b, c\}$, $\phi_{\mathsf{v}}(a) = 2$, $\phi_{\mathsf{v}}(b) = 3$, and $\phi_{\mathsf{v}}(c) = 1$.

A *Petri net* (more precisely, a P/T net) is a tuple $N = (P, T)$, where P is a finite set of *places* and T is a finite set of *transitions*. Each transition is of the form $\mathsf{u} \to \mathsf{v}$ where u and v are nonempty multisets over P. A multiset over P is called a *marking* of N. For markings x, y we write $\mathsf{x} \Rightarrow_N \mathsf{y}$ if there exist markings $\mathsf{u}, \mathsf{v}, \mathsf{w}$ such that $\mathsf{u} \to \mathsf{v}$ is a transition in T, $\mathsf{x} = \mathsf{u} \cup \mathsf{w}$, and $\mathsf{y} = \mathsf{v} \cup \mathsf{w}$. The *reachability problem* asks, for a Petri net N and markings x and y, whether or not $\mathsf{x} \Rightarrow_N^* \mathsf{y}$; it is shown to be decidable in [11]. The Petri net N is *reversible* if T is symmetric, meaning that if $\mathsf{u} \to \mathsf{v}$ is in T then $\mathsf{v} \to \mathsf{u}$ is in T. A reversible Petri net is the same as a finite presentation of a commutative semigroup, and the reachability problem for reversible Petri nets is therefore the same as the uniform word problem for commutative semigroups. It is shown in [12] that this problem is decidable (and even complete) in exponential space, i.e., in space c^n for some constant c, where n is the size of the problem instance. The precise complexity of the general reachability problem is open.

We now translate processes into nonempty multisets. Let $[P]_{\mathcal{R}}$ denote the equivalence class containing P of the congruence $\equiv_{\mathcal{R}}$, i.e., $[P]_{\mathcal{R}} = \{Q \mid Q \equiv_{\mathcal{R}} P\}$. For a process P, the *decomposition of* P is $\mathsf{d}_{\mathcal{R}}(P) = \Phi([P_1]_{\mathcal{R}}, \ldots, [P_m]_{\mathcal{R}})$, where $P = P_1 \mid \cdots \mid P_m$ and P_i is a web. Thus, $\mathsf{d}_{\mathcal{R}}(P)$ is the multiset of the equivalence classes of the webs in the webform of P. By $\pi_{\mathcal{R}}(P)$ we denote the underlying set of these equivalence classes, i.e., $\pi_{\mathcal{R}}(P) = \{[P_1]_{\mathcal{R}}, \ldots, [P_m]_{\mathcal{R}}\}$. We will need a few simple properties of decomposition.

Lemma 3. *For processes P and Q, set $\mathcal{R}$ of processes, and multiset* w,

(1) $\mathsf{d}_{\mathcal{R}}(P \mid Q) = \mathsf{d}_{\mathcal{R}}(P) \cup \mathsf{d}_{\mathcal{R}}(Q)$,
(2) *if* $\mathsf{d}_{\mathcal{R}}(P) = \mathsf{d}_{\mathcal{R}}(Q)$, *then* $P \equiv_{\mathcal{R}} Q$, *and*
(3) *if* $\mathsf{d}_{\mathcal{R}}(P) = \mathsf{d}_{\mathcal{R}}(Q) \cup \mathsf{w}$ *with* $\mathsf{w} \neq \emptyset$, *then there exists a process R such that* $P \equiv_{\mathcal{R}} Q \mid R$ *and* $\mathsf{d}_{\mathcal{R}}(R) = \mathsf{w}$.

Lemma 4. *It is decidable in exponential space, for processes P and Q and for a finite set $\mathcal{R}$ of processes, whether or not $P \equiv_{\mathcal{R}} Q$.*

Proof. We may assume that $\rho(P) = \rho(Q)$ (because otherwise they are not congruent), and that $\rho(R) < \rho(P)$ for every $R \in \mathcal{R}$ (because every R with $\rho(R) \geqslant \rho(P)$ can be dropped from $\mathcal{R}$: the corresponding instance $!R \equiv\ !R \mid R$ of the replication law cannot be applied to P or to any process congruent to P).

The proof is by induction on the replication depth n of P and Q. Thus, we assume that $P' \equiv_{\mathcal{R}} Q'$ is decidable for processes P', Q' of replication depth $< n$. For given P and Q of replication depth n we construct a Petri net N with the following places and transitions. The set of places is

$$\pi_{\mathcal{R}}(P) \cup \pi_{\mathcal{R}}(Q) \cup \bigcup \{\pi_{\mathcal{R}}(!R \mid R) \mid R \in \mathcal{R}\},$$

and the set of transitions consists of all

$$\mathsf{d}_{\mathcal{R}}(!R) \to \mathsf{d}_{\mathcal{R}}(!R \mid R) \quad \text{and}$$
$$\mathsf{d}_{\mathcal{R}}(!R \mid R) \to \mathsf{d}_{\mathcal{R}}(!R)$$

with $R \in \mathcal{R}$. Note that, by definition, the Petri net N is reversible. It is important to realize that N can indeed be constructed, due to the induction hypothesis. Each place is the equivalence class of either an action $a \in \mathbf{A}$ or a replication $!S$ with $\rho(S) < n$. Since, obviously, $[a]_{\mathcal{R}} = \{a\}$ for every action a, equality of these equivalence classes can be decided by the induction hypothesis and by Lemma 2.

We now claim that $P \equiv_{\mathcal{R}} Q$ if and only if $\mathsf{d}_{\mathcal{R}}(P) \Rightarrow_N^* \mathsf{d}_{\mathcal{R}}(Q)$. By the decidability of the reachability problem of Petri nets, this implies that $P \equiv_{\mathcal{R}} Q$ is decidable.

For the only-if direction of the claim it suffices to show that if $S \to_{\mathcal{R}} S'$ and $\mathsf{d}_{\mathcal{R}}(S)$ is a marking of N, then $\mathsf{d}_{\mathcal{R}}(S) = \mathsf{d}_{\mathcal{R}}(S')$ or $\mathsf{d}_{\mathcal{R}}(S) \Rightarrow_N \mathsf{d}_{\mathcal{R}}(S')$. The first conclusion clearly holds when a composition law is applied to (a subterm of) S or when the replication law is applied to a proper subterm of a web of the webform of S. Thus, it remains to consider $S =\ !R \mid T$ and $S' =\ !R \mid R \mid T$ with $R \in \mathcal{R}$, where T may also be absent. By Lemma 3(1), $\mathsf{d}_{\mathcal{R}}(S) = \mathsf{d}_{\mathcal{R}}(!R) \cup \mathsf{w}$ and $\mathsf{d}_{\mathcal{R}}(S') = \mathsf{d}_{\mathcal{R}}(!R \mid R) \cup \mathsf{w}$, where $\mathsf{w} = \mathsf{d}_{\mathcal{R}}(T)$ or $\mathsf{w} = \emptyset$ (when T is absent). Hence $\mathsf{d}_{\mathcal{R}}(S) \Rightarrow_N \mathsf{d}_{\mathcal{R}}(S')$.

For the if direction of the claim it suffices, by Lemma 3(2), to show that if $\mathsf{d}_{\mathcal{R}}(S) \Rightarrow_N \mathsf{y}$ then there exists a process S' such that $\mathsf{d}_{\mathcal{R}}(S') = \mathsf{y}$ and $S \equiv_{\mathcal{R}} S'$. Let $\mathsf{d}_{\mathcal{R}}(S) = \mathsf{d}_{\mathcal{R}}(!R) \cup \mathsf{w}$ and $\mathsf{y} = \mathsf{d}_{\mathcal{R}}(!R \mid R) \cup \mathsf{w}$ with $R \in \mathcal{R}$ (the reverse case is similar). Assume that $\mathsf{w} \neq \emptyset$ (the case that $\mathsf{w} = \emptyset$ is similar, using Lemma 3(2)). By Lemma 3(3), there exists T such that $S \equiv_{\mathcal{R}}\ !R \mid T$ and

$\mathsf{d}_{\mathcal{R}}(T) = \mathsf{w}$. Then $S' = {!R} \mid R \mid T$ satisfies the requirements.

It remains to show the existence of an exponential space algorithm. Let MM be the exponential space algorithm of [12] for the reachability problem for reversible Petri nets. For given $P, Q, \mathcal{R}$, let D be the set $\mathrm{desc}(!P) \cup \mathrm{desc}(!Q) \cup \bigcup\{\mathrm{desc}(!R) \mid R \in \mathcal{R}\}$, with cardinality d. It should be clear from the inductive proof above, that there is an algorithm for deciding $P \equiv_{\mathcal{R}} Q$ that, for $i = 0, \ldots, n$, decides for every two processes $S, T \in \mathsf{D}$ of replication depth i whether or not $S \equiv_{\mathcal{R}} T$, and stores that information in its workspace. Since this algorithm calls MM at most d^2 times, for subterms of P, Q, or $R \in \mathcal{R}$, it clearly works in exponential space. □

Lemma 5. *If $P \equiv Q$ then $P \equiv_{\mathrm{desc}(P)} Q$.*

Proof. We show that $P \to^n Q$ implies $P \equiv_{\mathrm{desc}(P)} Q$, by induction on n. The case $n = 0$ is trivial. Assume now that $P \to^n Q \to Q'$. By induction, $P \equiv_{\mathrm{desc}(P)} Q$. We only need to consider the case that $Q \to Q'$ by an application of the replication law. Suppose that $Q = T[!R]$ and $Q' = T[!R \mid R]$, where $T[\cdot]$ is a context. Since $R \in \mathrm{desc}(Q)$, there exists $S \in \mathrm{desc}(P)$ such that $S \equiv_{\mathrm{desc}(P)} R$ by Lemma 1. Then $P \equiv_{\mathrm{desc}(P)} Q = T[!R] \equiv_{\mathrm{desc}(P)} T[!S]$. In one step $T[!S] \equiv_{\mathrm{desc}(P)} T[!S \mid S]$, because $S \in \mathrm{desc}(P)$. And finally $T[!S \mid S] \equiv_{\mathrm{desc}(P)} T[!R \mid R] = Q'$. Thus $P \equiv_{\mathrm{desc}(P)} Q'$. The case that $Q = T[!R \mid R]$ and $Q' = T[!R]$ is similar. □

2. Guards and null

Next we consider the π-calculus without restriction and still without the null process. In other words, we add guards (but keep the actions).

The set of *names* is denoted by $\mathbf{N}$. The rule $P ::= g.P$ is added to the syntax, where the guard g can appear in two forms: as an input guard $x(y)$, or as an output guard $\bar{x}z$, with $x, y, z \in \mathbf{N}$. In a guarded term $x(y).P$ the name y is *bound* in P. As usual, process P is α-congruent to process Q, denoted by $P \equiv_\alpha Q$, if P can be converted to Q by a change of bound names. The structural laws are the same as before, plus the *α-law* which says that $P \equiv Q$ whenever $P \equiv_\alpha Q$.

For a process P, we let $\gamma(P)$ be the *guarding depth* of P, i.e., the nesting depth of guards in P: $\gamma(a) = 0$, $\gamma(P \mid Q) = \max(\gamma(P), \gamma(Q))$, $\gamma(!P) = \gamma(P)$, and $\gamma(g.P) = \gamma(P) + 1$. Obviously, congruent processes have the same guarding depth, i.e., if $P \equiv Q$ then $\gamma(P) = \gamma(Q)$. Moreover, for a process P, we define the set $\mathrm{gua}(P) = \{[g.R] \mid g.R \text{ is an outer subterm of } P\}$, where 'outer' means that $g.R$ is not in the context of another guard of P, and where $[g.R]$ is the equivalence class of $g.R$ with respect to $\equiv$. It is easy to see that if $P \equiv Q$ then $\mathrm{gua}(P) = \mathrm{gua}(Q)$.

Lemma 6. *For a guard g and processes P and Q, $g.P \equiv Q$ if and only if there exists a process Q' such that $Q \equiv_\alpha g.Q'$ and $P \equiv Q'$.*

Proof. The if-part is trivial. For the other part we first observe that if $S \equiv T$ then there exists $U \equiv_\alpha T$ such that $S \sim U$, where $\sim$ is the smallest congruence satisfying the laws of $\equiv$ *except* the α-law. This can easily be shown by induction on the number of applications of $\equiv$ laws. Hence, if $g.P \equiv Q$ then there exists U such that $g.P \sim U \equiv_\alpha Q$. Since the α-law is the only law that affects guards, $g.P \sim U$ implies that $U = g.Q'$ for some Q' with $P \sim Q'$ (and so $P \equiv Q'$). □

Clearly, the processes Q' with $Q \equiv_\alpha g.Q'$ are all α-congruent. Thus, to check whether $g.P \equiv Q$ it suffices to consider one such Q' and check whether $P \equiv Q'$. Note that if such a Q' does not exist, then $g.P \not\equiv Q$. Note also that $\gamma(Q') = \gamma(Q) - 1$.

The proof of decidability of $P \equiv Q$ for this version of the π-calculus is by induction on the guarding depth n of P and Q. For $n = 0$ we are back in the case of the previous section. Assume now that $n > 0$ and that $P' \equiv_{\mathcal{R}} Q'$ is decidable for processes P', Q' of guarding depth $< n$. For processes P and Q of guarding depth n, it follows from the induction hypothesis and Lemma 6 that it is decidable whether or not $\mathrm{gua}(P) = \mathrm{gua}(Q)$. If not, then P and Q are not congruent. Assume now that $\mathrm{gua}(P) = \mathrm{gua}(Q)$. Let P' and Q' be obtained from P and Q by replacing each outer subterm $S = g.T$ by the action $a_{[S]}$, where the mapping $[S] \mapsto a_{[S]}$ is injective, and the actions $a_{[S]}$ are 'new', i.e. do not yet occur in P or Q. In other words, every outer guarded subterm is replaced by a new action, in such a way that congruent subterms are replaced by the same action. Observe that if $P' = Q'$ then $P \equiv Q$. We now claim that $P \equiv Q$ if and only if $P' \equiv Q'$, which is decidable by the previous section. In the only-if direction it suffices to show that if $P \to Q$ then $P' = Q'$ or $P' \to Q'$. The first conclusion holds for any application of the α-law, and whenever a law is applied to (a subterm of) an outer subterm $g.R$ of P; otherwise, clearly, the second conclusion holds. In the if-direction it suffices (by the above observation) to show that if $P' \to R$ then there exists T such that $P \to T$ and $T' = R$; that is easy to show.

It should be clear that the above proof leads to an exponential space algorithm, by an argument similar to the one at the end of the proof of Lemma 4.

Finally, we add the null process $\mathbf{0}$. The syntax is extended with the rule $P ::= \mathbf{0}$, and the structural law $P \mid \mathbf{0} \equiv P$ is added: the *zero-law*. This produces the π-calculus without restriction (and still with actions). To show the decidability of $P \equiv Q$, let P' and Q' be obtained from P and Q by iteratively applying the zero-law in the direction from left to right, and then replacing each subterm $!\mathbf{0}$ by $a_{!0}$ and each subterm $g.\mathbf{0}$ by $g.a_0$, where $a_{!0}$ and a_0 are two distinct new actions. It is left to the reader to show that $P \equiv Q$ if and only if $P' \sim Q'$, where $\sim$ is the smallest congruence satisfying all laws of $\equiv$ *except* the zero-law. This brings us back to the previous case of the π-calculus without the null process.

3. Conclusion

We have proved that *Milner's structural congruence is decidable for the π-calculus without restriction, in exponential space*.

The decidability of Milner's structural congruence for the full π-calculus is left as an exercise (for the more experienced logician). To be honest, we ourselves are not sufficiently experienced to do that exercise (and we could even imagine undecidability). The main problem is the intricate 'interplay' between restriction, with its scope extrusion law, and replication. The technique of this paper can be used to simplify the proof of [9], where Milner's structural congruence is shown to be decidable for a fairly large, behaviourally complete subset of the π-calculus (including all restriction-free processes).

Another problem that we are not able to solve is whether or not Milner's structural congruence is complete in exponential space.

We finally observe that *extended* structural congruence, see [6,7,10], can easily be decided for the π-calculus without restriction. Thus, the complicated algorithm (for the full π-calculus) in [7] is entirely due to the above-mentioned interplay between restriction and replication. Let us briefly discuss how structural congruence can be decided for the processes in Section 1 when the structural laws $!(P \mid Q) \equiv !P \mid !Q$ and $!\,!\,P \equiv !P$ are added. Every process is now congruent with a process of the form $!a_1 \mid \cdots \mid !a_m \mid b_1 \mid \cdots \mid b_n$, where the a_i are distinct actions and the b_j are (not necessarily distinct) actions with $b_j \neq a_i$. This normal form can easily be obtained by first iteratively applying the above two laws from left to right, and then iteratively applying the replication law $!P \mid P \equiv P$ and the derived law $!P \mid !P \equiv !P$, also from left to right (with, of course, the composition laws taken for granted). It is not difficult to show, and easily follows from the proofs in [7,10], that two processes of that form are structurally congruent if and only if they are the same (modulo the two composition laws). The same proof holds after adding the null process $\mathbf{0}$, with the law $P \mid \mathbf{0} \equiv P$ and the additional law $!\mathbf{0} \equiv \mathbf{0}$: view $\mathbf{0}$ as an action, and apply these two laws (from left to right) in the final phase. For the addition of guarding, the proof is entirely the same as in Section 2.

References

[1] R.M. Amadio, C. Meyssonnier, On decidability of the control reachability problem in the asynchronous π-calculus, Nordic J. Comput. 9 (2002) 70–101.

[2] N. Busi, R. Gorrieri, A Petri net semantics for π-calculus, in: Proc. Concur'95, in: Lecture Notes in Computer Science, vol. 962, Springer, Berlin, 1995, pp. 145–159.

[3] L. Caires, L. Cardelli, A spatial logic for concurrency (part I), Inform. and Comput. 186 (2003) 194–235.

[4] L. Cardelli, A.D. Gordon, Mobile ambients, Theoret. Comput. Sci. 240 (2000) 177–213.

[5] S. Dal Zilio, Spatial congruence for ambients is decidable, in: Proc. ASIAN'00—6th Asian Computing Science Conference, in: Lecture Notes in Computer Science, vol. 1961, Springer, Berlin, 2000, pp. 88–103.

[6] J. Engelfriet, A multiset semantics for the pi-calculus with replication, Theoret. Comput. Sci. 153 (1996) 65–94.

[7] J. Engelfriet, T.E. Gelsema, Multisets and structural congruence of the pi-calculus with replication, Theoret. Comput. Sci. 211 (1999) 311–337.

[8] J. Engelfriet, T.E. Gelsema, A new natural structural congruence in the pi-calculus with replication, Acta Inform. 40 (2004) 385–430.

[9] J. Engelfriet, T.E. Gelsema, The decidability of structural congruence for replication restricted pi-calculus processes, LIACS Technical Report 2004-07, May 2004, http://www.liacs.nl/home/engelfri/repres.ps.

[10] D. Hirschkoff, Mise en oeuvre de preuves de bisimulation, Ph.D. thesis, École Nationale des Ponts et Chaussées, 1999.

[11] E.W. Mayr, An algorithm for the general Petri net reachability problem, SIAM J. Comput. 13 (1984) 441–460.

[12] E.W. Mayr, A.R. Meyer, The complexity of the word problems for commutative semigroups and polynomial ideals, Adv. Math. 46 (1982) 305–329.

[13] R. Milner, Functions as processes, Math. Struct. in Comput. Sci. 2 (1992) 119–141.

[14] R. Milner, Communicating and Mobile Systems: the π-Calculus, Cambridge University Press, Cambridge, 1999.

[15] R. Milner, Moca mailing list, May 15, 2001.

[16] C. Reutenauer, The Mathematics of Petri Nets, Prentice-Hall, New York, 1990.

[17] D. Sangiorgi, D. Walker, The π-calculus: A Theory of Mobile Processes, Cambridge University Press, Cambridge, 2001.

[18] P. Sobociński, Unpublished manuscript, 2001.