Multiset Rewriting for the Verification of Depth-Bounded Processes with Name Binding

Fernando Rosa-Velardo*, María Martos-Salgado

Sistemas Informáticos y Computación, Universidad Complutense de Madrid
Facultad de Informática, C/Prof. José García Santesmases, s/n. 28040 Madrid (Spain)

Abstract

We combine the two existing approaches to the study of concurrency by means of multiset rewriting: multiset rewriting with existential quantification (MSR) and constrained multiset rewriting. We obtain ν-MSR, where we rewrite multisets of atomic formulae, in which terms can only be pure names, where some names can be restricted. We consider the subclass of depth-bounded ν-MSR, for which the interdependence of names is bounded. We prove that they are strictly Well Structured Transition Systems, so that coverability, termination and boundedness are all decidable for depth-bounded ν-MSR. This allows us to obtain new verification results for several formalisms with name binding that can be encoded within ν-MSR, namely polyadic ν-PN (Petri nets with tuples of names as tokens), the π-calculus, MSR or Mobile Ambients.

Key words: Multiset rewriting, depth-boundedness, WSTS, verification, decidability, Petri nets, process algebrae

1. Introduction

MSR. Dynamic name generation has been thoroughly studied in the last decade, mainly in the fields of security [11, 1] and mobility [27]. The paper [11] presents a meta-notation for the specification and analysis of security protocols. This meta-notation involves facts and transitions, where facts are first-order atomic formulae and transitions are given by means of rewriting rules, with a precondition and a postcondition. For instance, the rule

$$A_0(k), Ann(k') \rightarrow \exists x.(A_1(k,x), N(enc(k', \langle x,k\rangle)), Ann(k'))$$

specifies the first rule of the Needham-Schroeder protocol, in which a principal A with key k ($A_0(k)$) decides to talk to another principal, with a key k' that

*Corresponding author

Email addresses: fernandorosa@sip.ucm.es (Fernando Rosa-Velardo), mrmartos@estumail.ucm.es (María Martos-Salgado)

Authors partially supported by the Spanish projects DESAFIOS10 TIN2009-14599-C03-01 and PROMETIDOS S2009/TIC-1465.

has been announced ($Ann(k')$), for which it creates a nonce x and sends to the network the pair $\langle x, k\rangle$ ciphered under k'. This notation gave rise to the specification language for security protocols MSR [10].

CMRS. In [14] *Constraint Multiset Rewriting Systems* (CMRS) are defined. As in [11], facts are first-order atomic formulae, but the terms that can appear as part of such formulae must belong to a *constraint system.* For instance, the rule $count(x), visit \to count(x+1), enter(x+1)$ could be used to count the number of visits to a web site. For a comprehensive survey of CMRS see [16]. In CMRS, there is no mechanism for name binding or name creation, so that it has to be simulated using the order in the constraint system (for instance, simulating the creation of a fresh name by taking a value greater than any of the values that have appeared so far). Thus, in an unordered version of CMRS, in which only the equality predicate between atoms is used, there is no way of ensuring that a name is fresh.

Our goal. It is our goal in this paper to find a minimal set of primitives that allows us to specify concurrent formalisms with name binding. This specification may be achieved by means of some encoding, provided this encoding preserves concurrency and name topology. This will allow us to obtain new decidability results for those concurrent formalisms in a common framework.

We combine the features of the meta-notation MSR and CMRS, obtaining ν-MSR. On the one hand, we maintain the existential quantifications in [11] to keep a compositional approach, closer to that followed in process algebra with name binding. On the other hand, we restrict terms in atomic formulae to be pure names, that can only be compared with equality or inequality, unlike the arbitrary terms over some syntax, as in [11], or terms in a constraint system, as in CMRS.

Depth boundedness. In the field of process algebra, there are many recent works that look for subclasses of the π-calculus for which some properties, such as termination, are decidable [6, 32, 31, 33, 4]. In this paper we will consider the results in [31] about depth-bounded π-calculus processes.

Depth-boundedness is a semantic restriction on π-calculus processes. Intuitively, a process is depth-bounded whenever the interdependence of names is bounded in any process reachable from it. As a simple example, and assuming that the reader is familiar with the following π-calculus syntax, if starting from some process P the processes

$$\nu a_1.\dots.a_n.(a_1\langle a_2\rangle \mid a_2\langle a_3\rangle \mid \cdots \mid a_i\langle a_{i+1}\rangle \mid \cdots \mid a_{n-1}\langle a_n\rangle) \mid Q_n$$

are reachable for every $n > 0$, then P is a depth-unbounded process. However, the fact that processes

$$\nu a.a_1.\dots.a_n.(a\langle a_1\rangle \mid a\langle a_2\rangle \mid \cdots \mid a\langle a_i\rangle \mid \cdots \mid a\langle a_n\rangle) \mid Q_n$$

can be reached from P for every n does not allow us to conclude that P is depth-unbounded, since though an unbounded number of names can appear in reachable processes, those names do not depend one another, as happened in the previous example.

Meyer proved in [31] that depth-bounded π-calculus processes are WSTS. In this paper we adapt those results to ν-MSR. More precisely, we will consider depth-bounded ν-MSR, that is, ν-MSR for which the interdependence of bound names is bounded in every reachable term. We will prove that this subclass of ν-MSR is well structured by following the same steps followed in [31]. Unfortunately, we will see that this property itself is undecidable for ν-MSR (and also for the π-calculus).

Then we will study the complexity of the decision procedures for depth-bounded ν-MSR, proving that they are all non-primitive recursive, thus rising the exponential space lower bound given in [31].

Models of concurrency with names. Two of the most well established models for concurrency are Petri nets and process algebra. The π-calculus is the paradigmatic example of process algebra with name binding. Names in the π-calculus can be used to build a dynamic communication topology. Two approaches to the dynamic generation of names in the field of Petri nets are ν-PNs [36] and Data Nets [29].

In ν-PNs, tokens are pure names that can move along the places of the net, be used to restrict the firing of transitions to happen only when some names match, and be created fresh. ν-PNs are (strictly) Well Structured Transition Systems (WSTS) [40, 21], but $p\nu$-PNs, its polyadic version, in which tokens are *tuples* of pure names, are not. Actually, $p\nu$-PNs are Turing-complete [37], even in the binary case.

In Data Nets, tokens are taken from a linearly ordered and dense domain, and whole-place operations (like transfers or resets) are allowed. However, in Data Nets (which are also WSTS), fresh name creation has to be simulated using the linear order, as happens in CMRS. Actually, CMRS and Data Nets are equivalent up to coverability languages (with coverability as accepting condition), even if the former cannot perform whole-place operations [3].

Though ν-PN have better decidability properties than $p\nu$-PN, some works need to use the model of $p\nu$-PN to model features like instance isolation in architectures with multiple concurrent conversations [13] or transactions in data bases [28]. We will prove that ν-MSRs are equivalent to $p\nu$-PNs. We will see that this equivalence is a rather strong one (isomorphism between the transitions systems). Moreover, the subclass of monadic ν-MSRs is equivalent to ν-PNs, so that coverability, boundedness and termination are decidable for them.

Next, we will see that processes of the π-calculus can be simulated, in a very natural way, by ν-MSRs. This translation is inspired by the results by Meyer about *structural stationary* π-calculus processes, that can be mapped to P/T nets [32]. As a corollary, depth-bounded π-calculus processes are well structured, which was already known. Finally, we apply the same techniques to other formalisms, like MSR [11] and Mobile Ambients [9]. Up to our knowledge, this is the first time that decidability results for the verification of safety properties of MSR are obtained. In the case of Mobile Ambients (MA), we obtain in particular the decidability of the name convergence problem, which is undecidable in general, even for the subclass without name restriction and in which ambients cannot be opened.

The rest of the paper is organized as follows. In Section 2 we introduce some basic definitions and notations we will use throughout the paper. Section 3 defines ν-MSR. In Section 4 we study depth-boundedness for ν-MSR. In Section 5 the equivalence between ν-MSRs and $p\nu$-PNs is proved. Section 6 presents the encoding of π-calculus terms within ν-MSR. Section 7 encodes other formalisms within ν-MSR, thus obtaining new decidability results for them. Finally, Section 8 presents our conclusions and some directions for future work. This paper is a revised and extended version of the papers [38, 39].

2. Preliminaries

A *quasi order* in A is a reflexive and transitive binary relation on A. A quasi order $\leq$ is said to be a *well-quasi order* (wqo) if for every infinite sequence $s_0, s_1, \ldots$ there are i and j, with $i < j$, such that $s_i \leq s_j$. Equivalently, it is a wqo if every infinite sequence has an increasing subsequence. Note that the equality relation is a wqo in any finite set.

Given an arbitrary set A, we will denote by $\mathcal{MS}(A)$ the set of *finite multisets* of A, that is, the set of mappings $m : A \to \mathbb{N}$ such that $supp(m) = \{a \in A \mid m(a) > 0\}$ (the *support* of m) is finite. We denote by $m_1 + m_2$, $m_1 \subseteq m_2$ and $m_1 - m_2$ the multiset addition, inclusion, and subtraction, respectively. Given $f : A \to B$ and $m \in \mathcal{MS}(A)$ then we can define $f(m) \in \mathcal{MS}(B)$ by $f(m)(b) = \sum_{f(a)=b} m(a)$.

Every quasi order $\leq$ defined in A induces a quasi order $\sqsubseteq$ in $\mathcal{MS}(A)$, given by $\{a_1, \ldots, a_n\} \sqsubseteq \{b_1, \ldots, b_m\}$ if there is some $h : \{1, \ldots, n\} \to \{1, \ldots, m\}$ injective such that $a_i \leq b_{h(i)}$ for all $i \in \{1, \ldots, n\}$. It is a well known fact that the multiset order induced by a wqo is also a wqo.

The set $\mathcal{T}(A)$ of *trees* over A is defined by

$$T ::= a \mid (a, \{T_1, \ldots, T_n\})$$

where a ranges over A. An order $\leq$ over A induces the *rooted tree embedding* [31] $\preceq$ over $\mathcal{T}(A)$, given by $a \preceq a'$ if $a \leq a'$, and $(a, \mathcal{A}) \preceq (a', \mathcal{A}')$ if $a \leq a'$ and $\mathcal{A} \sqsubseteq \mathcal{A}'$, where $\sqsubseteq$ is the multiset order induced by $\preceq$. The mapping *height*(T) is defined as *height*$(a) = 0$ and *height*$(a, \{T_1, \ldots, T_n\}) = 1 + max\{$*height*$(T_i) \mid i = 1, \ldots, n\}$. If we denote by $\mathcal{T}(A)_n$ the set of trees of height less or equal than n, then $(\mathcal{T}(A)_n, \preceq)$ is a wqo provided $(A, \leq)$ is a wqo [31].

A *hypergraph* is a tuple $\mathcal{G} = (V, E, inc)$, where V is the set of vertices, E is the set of edges and for each $e \in E$, $inc(e)$ is the set of vertices that incide in e. There is an arc between $v \in V$ and $e \in E$ whenever $v \in inc(e)$.

A *transition system* is a tuple $(S, \to, s_0)$, where S is a (possibly infinite) set of states, $s_0 \in S$ is the initial state and $\to \subseteq S \times S$. We denote by $\to^*$ the reflexive and transitive closure of $\to$. The *reachability* problem in a transition system consists in deciding for a given state s_f whether $s_0 \to^* s_f$. The *termination* problem consists in deciding whether there is an infinite sequence $s_0 \to s_1 \to s_2 \to \cdots$. The boundedness problem consists in deciding whether the set of reachable states is finite. For any transition system $(S, \to, s_0)$ endowed with a quasi order $\leq$ we can define the *coverability* problem, that consists in deciding, given a state s_f, whether there is $s \in S$ reachable such that $s_f \leq s$.

A *Well Structured Transition System* (WSTS) is a tuple $(S, \to, \leq, s_0)$, where $(S, \to, s_0)$ is a transition system and $\leq$ is a decidable wqo compatible with $\to$, meaning that $s_1' \geq s_1 \to s_2$ implies that there is $s_2' \geq s_2$ with $s_1' \to s_2'$. We will refer to this property as monotonicity of $\to$ with respect to $\leq$. For WSTS, the termination problem is decidable [21]. A WSTS is said to be *strict* if it satisfies the following strict compatibility condition: $s_1' > s_1 \to s_2$ implies that there is $s_2' > s_2$ with $s_1' \to s_2'$. For strict WSTS, also the boundedness problem is decidable [21]. A WSTS satisfies the *effective pred-basis* property if for every $s \in S$, the set of minimal elements in $\{s' \in S \mid s' \to s'' \geq s\}$ (which is always finite) is computable. For WSTS that satisfy the effective pred-basis property, also the coverability problem is decidable.

3. ν-MSR

Let us now define ν-MSR. We fix a denumerable set of predicate symbols $\mathcal{P}$, a denumerable set *Id* of names and a denumerable set *Var* of variables. We use $a, b, c, \ldots$ to range over *Id*, $x, y, \ldots$ to range over *Var*, and $\eta, \eta' \ldots$ to range over $Id \cup Var$.

An *atomic formula* has the form $p(\eta_1, \ldots, \eta_n)$, where $p \in \mathcal{P}$ and $\eta_i \in Var \cup Id$ for all i. A *ground atomic formula* has the form $p(a_1, \ldots, a_n)$, where $p \in \mathcal{P}$ and $a_i \in Id$ for all i. We use $X, Y, \ldots$ to range over atomic formulae and $A, B, \ldots$ to range over atomic ground formulae. We denote by $Var(X)$ and $Id(X)$ the set of variables and names appearing in X, respectively. We will write $\tilde{x}$ and $\tilde{a}$ to denote finite sequences of variables and names, respectively, so that we will sometimes write $p(\tilde{x})$ or $p(\tilde{a})$. Moreover, we will sometimes use set notation with these sequences and write, for instance, $x \in \tilde{x}$ or $\tilde{x}_1 \cup \tilde{x}_2$.

Definition 1. A ν-MSR *term* is given by the following grammar:

$$M ::= \mathbf{0} \mid A \mid M_1 + M_2 \mid \nu a.M$$

We denote by $\mathcal{M}$ the set of ν-MSR terms, and use M, M', $M_1, \ldots$ to range over $\mathcal{M}$. We define $fn : \mathcal{M} \to \mathcal{P}(Var)$ as $fn(\mathbf{0}) = \emptyset$, $fn(A) = Id(A)$, $fn(M_1 + M_2) = fn(M_1) \cup fn(M_2)$, and $fn(\nu a.M) = fn(M) \setminus \{a\}$.

Definition 2. A *rule* t is an expression of the form

$$t : X_1 + \ldots + X_n \to \nu\tilde{a}.(Y_1 + \ldots + Y_m)$$

such that if $x \in Var(Y_j)$ for some j then $x \in Var(X_i)$ for some i. A ν-MSR is a tuple $\langle \mathcal{R}, M_0 \rangle$, where M_0 is the initial ν-MSR term and $\mathcal{R}$ is a finite set of rules.

Sometimes in the examples, we will use commas instead of the symbol $+$. For instance, we will write $p(x,y), q(y,y) \rightarrow \nu a.q(x,a)$ instead of $p(x,y)+q(y,y) \rightarrow \nu a.q(x,a)$. Given a rule $t: X_1,\ldots,X_n \rightarrow \nu\tilde{a}.(Y_1,\ldots,Y_m)$, we will write $pre(t) = \bigcup_{i=1}^{n} Var(X_i)$, $post(t) = \bigcup_{j=1}^{m} Var(Y_m)$, and $Var(t) = pre(t) \cup post(t)$. With these notations, every rule t satisfies $post(t) \subseteq pre(t)$.

We will identify ν-MSR terms up to $\equiv$, defined as the least congruence on $\mathcal{M}$ where α-conversion of bound names is allowed, such that $(\mathcal{M},+,\mathbf{0})$ is a commutative monoid and the three following equations hold:

$$\nu a.\nu b.M \equiv \nu b.\nu a.M \qquad \nu a.\mathbf{0} \equiv \mathbf{0}$$

$$\nu a.(M_1 + M_2) \equiv \nu a.M_1 + M_2 \quad if \quad a \notin fn(M_2)$$

The first rule justifies our notation $\nu\tilde{a}.M$. The last rule is usually called *name extrusion* when applied from right to left. A *substitution* for $t: X_1 + \ldots + X_n \rightarrow \nu\tilde{a}.(Y_1+\ldots+Y_m)$ is any mapping $\sigma : Var(t) \rightarrow Id$. We write $pre_t(\sigma) = \sigma(X_1)+\ldots+\sigma(X_n)$, where $\sigma(p(\eta_1,\ldots,\eta_n)) = p(a_1,\ldots,a_n)$, with $a_i = \sigma(\eta_i)$ if $\eta_i \in Var$, or $a_i = \eta_i$ if $\eta_i \in Id$.

In order to define the analogous $post_t(\sigma)$, and to avoid capturing free names, we consider a sequence of pairwise different names $\tilde{b}$ (of the same length as $\tilde{a}$) such that $\sigma(Var(t)) \cap \tilde{b} = \emptyset$. Then, we take $\sigma' = \sigma \circ \{\tilde{a}/\tilde{b}\}$ and $post_t(\sigma) = \nu\tilde{b}.(\sigma'(Y_1)+\ldots+\sigma'(Y_m))$, where $\{\tilde{a}/\tilde{b}\}$ denotes the simultaneous substitution of each $a_i \in \tilde{a}$ by the corresponding $b_i \in \tilde{b}$. Let us define the transition system $(\mathcal{M},\rightarrow,M_0)$, where $\rightarrow$ is the least relation such that:

$$(t)\ \frac{\sigma : Var(t) \rightarrow Id}{pre_t(\sigma) \rightarrow post_t(\sigma)} \qquad \frac{M_1 \equiv M_1' \rightarrow M_2' \equiv M_2}{M_1 \rightarrow M_2}\ (\equiv)$$

$$(+)\ \frac{M_1 \rightarrow M_2}{M_1 + M \rightarrow M_2 + M} \qquad \frac{M_1 \rightarrow M_2}{\nu a.M_1 \rightarrow \nu a.M_2}\ (\nu)$$

Rules $(+)$ and (ν) state that transitions can happen inside a sum or inside a restriction, respectively. Rule $(\equiv)$ is also standard, and formalizes that we are rewriting terms modulo $\equiv$. Then we have a rule (t) for each $t \in \mathcal{R}$. For instance, let $t: p(x), q(x) \rightarrow \nu b.p(b)$ be a rule in $\mathcal{R}$. Then the rewriting $p(a), q(a) \rightarrow \nu b.p(b)$ can take place by taking $\sigma(x) = a$, which satisfies $pre_t(\sigma) = p(a), q(a)$ and $post_t(\sigma) = \nu b.p(b)$. Consider now the term $p(b), q(b)$. In order to apply the previous rule, one must necessarily consider the substitution σ given by $\sigma(x) = b$, that does not satisfy $\sigma(Var(t)) \cap \{b\} = \emptyset$. Therefore, we need to first rename b in the right handside of the rule, obtaining (e.g. if we replace b by c) $\nu c.p(c)$. We denote by $\longrightarrow_{\not\equiv}$ the transition relation obtained as above though without using the rule $(\equiv)$.

As in the π-calculus, we can consider several normal forms, that force a certain rearrangement of bound names.

Definition 3. A term M is in *standard normal form* if there is a set of names $\tilde{a}$ and atomic formulae $A_1,\ldots,A_n$ such that $M = \nu\tilde{a}.(A_1+\ldots+A_n)$.

Clearly, every term is equivalent to some term in standard form. To obtain it, it is enough to apply the extrusion rule (from right to left) as much as necessary, α-converting the bounded names when needed. The standard form is unique up to commutativity and associativity of $+$, α-conversion and commutativity of the names in $\tilde{a}$. Notice that the right handside of rules is always in standard normal form. Anyhow, we can specify a rule with a right handside not in standard normal form if needed, just by *converting* it to an equivalent term in standard form. Moreover, we can prove the following result, that relates the transition relation with the standard normal form.

Proposition 1. *$M_1 \to M_2$ iff the following holds:*

- *$M_i \equiv \nu\tilde{a}_i.(A_1^i + \ldots + A_{n_i}^i + M)$ for $i = 1, 2$, and some $M \in \mathcal{M}$ without restrictions,*
- *there is $t : X_1^1 + \ldots + X_{n_1}^1 \to \nu\tilde{a}.(X_1^2 + \ldots + X_{n_2}^2)$ in $\mathcal{R}$, σ substitution for t and $\tilde{b}$ with $\sigma(Var(t)) \cap \tilde{b} = \emptyset$ such that $\sigma(X_j^1) = A_j^1$, $\sigma(X_j^2)\{\tilde{a}/\tilde{b}\} = A_j^2$ and $\tilde{a}_1 \sqcup \tilde{b} = \tilde{a}_2$.*

Proof. We prove the *if* implication by induction on the rules proving $M_1 \to M_2$.

- If $M_1 = pre_t(\sigma)$ and $M_2 = post_t(\sigma)$ for some rule t and some substitution σ for t, then trivially both M_1 and M_2 are in standard form, and $\tilde{a}_1 = \emptyset$ and $\tilde{a}_2 = \tilde{b}$, so that clearly $\tilde{a}_1 \sqcup \tilde{b} = \tilde{a}_2$.
- Let $M_i = M' + M_i'$ with $M_1' \to M_2'$, so that by the induction hypothesis, $M_i' \equiv \nu\tilde{a}_i.(A_1^i + \ldots + A_{n_i}^i + M)$ and $\tilde{a}_1 \sqcup \tilde{b} = \tilde{a}_2$. We assume $fn(M') \cap \tilde{a}_i = \emptyset$, or we rename the names in $\tilde{a}_i$ that are free in M, obtaining a term that is equivalent modulo $\equiv$. Let $M \equiv \nu\tilde{c}.M''$ in standard form. As before, we can assume that $\tilde{a}_i \cap \tilde{c} = \emptyset$. Then, $M_i = M_i' + M' \equiv \nu\tilde{a}_i.(A_1^i + \ldots + A_{n_i}^i + M) + \nu\tilde{c}.M''$, which by the extrusion rule is equivalent to $\nu\tilde{a}_i, \tilde{c}.(A_1^i + \ldots + A_{n_i}^i + M + M'')$. Moreover, $\tilde{a}_2 \sqcup \tilde{c} = \tilde{a}_1 \sqcup \tilde{b} \sqcup \tilde{c}$ and we conclude.
- The cases for (ν) and $(\equiv)$ are straightforward.

Conversely, $A_1^1 + \ldots + A_{n_1}^i \to \nu\tilde{b}.(A_1^2 + \ldots + A_{n_2}^2)$ holds by rule (t). Rules $(+)$ and $(\equiv)$ for the extrusion, tells us that $A_1^1 + \ldots + A_{n_1}^i + M \to \nu\tilde{b}.(A_1^2 + \ldots + A_{n_2}^2 + M)$, and by successively applying rule (ν) for all the names in $\tilde{a}_1$, we obtain that $\nu\tilde{a}_1.(A_1^1 + \ldots + A_{n_1}^1 + M^1) \to \nu\tilde{a}_2.(A_1^2 + \ldots + A_{n_2}^2 + M^2)$. Finally, again by rule $(\equiv)$ we can conclude that $M_1 \to M_2$.

Since the behavior of ν-MSR systems is specified in terms of a congruent rewriting relation (with respect to all the constructors), modulo the equational theory defined by $\equiv$, which is compatible with respect to the rewriting relation (Prop. 1) the translation from a ν-MSR specification to an equivalent rewrite specification is straightforward [30]. In [38] some of the details of their representation in the Maude system [12] are given. This representation of ν-MSRs within Maude allows us to use all the analysis machinery available for it.

Let us now define in our setting the restricted normal form of a term, which can be seen as the opposite concept to standard form. Intuitively, a term is in restricted form if the scope of its restrictions is minimal, that is, if every expression $\nu a.(A_1 + \ldots + A_m)$ satisfies $a \in fn(A_i)$ for all i, so that no extrusion rule can be applied from left to right.

Definition 4. Let us define $\hat{\equiv}$ as the least congruence on $\mathcal{M}$ such that $(\mathcal{M}, +, \mathbf{0})$ is a commutative monoid, and $\rightsquigarrow$ as the least binary relation on $\mathcal{M}$ such that:

$$\frac{a \notin fn(M_2)}{\nu a.(M_1 + M_2) \rightsquigarrow \nu a.M_1 + M_2} \qquad \frac{M_1 \hat{\equiv} M_1' \rightsquigarrow M_2' \hat{\equiv} M_2}{M_1 \rightsquigarrow M_2}$$

$$\frac{M_1 \rightsquigarrow M_2}{M_1 + M \rightsquigarrow M_2 + M} \qquad \frac{M_1 \rightsquigarrow M_2}{\nu a.M_1 \rightsquigarrow \nu a.M_2}$$

We say M is in *restricted form* if there is no M' with $M \rightsquigarrow M'$. We say a term M in restricted form is a *fragment* if it cannot be decomposed as $M = M_1 + M_2$.

Any M in restricted form satisfies $M = F_1 + \ldots + F_n$ with F_i fragments, and any fragment is either an atomic formula or a term of the form $\nu a.(F_1 + \ldots + F_m)$, with F_i fragments such that $a \in fn(F_i)$, for all i. For instance, $M = \nu a.\nu a_1.\ldots.\nu a_n.(p(a, a_1), \ldots, p(a, a_n)) \rightsquigarrow F = \nu a.(\nu a_1.p(a, a_1), \ldots, \nu a_n.p(a, a_n))$. It holds that F is a fragment, as well as each $\nu a_i.p(a, a_i)$.

Intuitively, within a fragment some bound names are shared. For instance, if $M = F_1 + F_2$, then F_1 and F_2 share no names in M, and if $F = \nu a.(F_1 + F_2)$ then a is the only name shared by F_1 and F_2 in F.

The relation $\rightsquigarrow$ is confluent up to $\hat{\equiv}$. Moreover, if $M \rightsquigarrow M'$ then $M \equiv M'$. Unlike the standard normal form, the restricted normal form is not compatible with the transition relation (that is, we do not have the analogous result to Prop. 1 for the restricted normal form). For instance, for M and F as above, the rule $t : p(x, y_1), p(x, y_2) \rightarrow q(x)$ satisfies $M \xrightarrow{t}_{\not\equiv}$ but not $F \xrightarrow{t}_{\not\equiv}$. However, restricted normal forms give more insight about the topology of pure names in terms. In particular, they are the basis of the proof that depth-bounded ν-MSR terms yield WSTS.

4. Depth-bounded ν-MSR

We now consider depth-bounded ν-MSR. Intuitively, a ν-MSR is depth-bounded if names cannot appear *linked* in an arbitrarily long way. Thus, if every term of the form

$$\nu a_1, \ldots, \nu a_n.(p(a_1, a_2), p(a_2, a_3), \ldots, p(a_{n-1}, a_n))$$

can be reached, then the ν-MSR is not depth-bounded. However, reaching all terms of the form $\nu a_1, \ldots, \nu a_n, \nu a.(p(a, a_1), \ldots, p(a, a_n))$ does not allow us

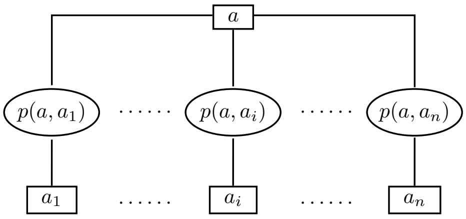

Figure 1: Hypergraph of the fragment $F = \nu a_1 . \ldots \nu a_n . \nu a . (p(a, a_1), \ldots, p(a, a_n))$, or its equivalent $F' = \nu a . (\nu a_1 . p(a, a_1) + \nu a_2 . p(a, a_2) + \ldots + \nu a_n . p(a, a_n))$

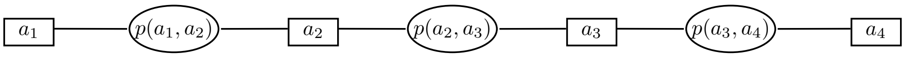

Figure 2: Hypergraph of the fragment $F = \nu a_1, a_2, a_3, a_4 . (p(a_1, a_2), p(a_2, a_3), p(a_3, a_4))$, or its equivalent $F' = \nu a_2 . (\nu a_1 . p(a_1, a_2) + \nu a_3 . (p(a_2, a_3) + \nu a_4 . p(a_3, a_4)))$

to conclude that the ν-MSR is depth-unbounded. In order to define depth-boundedness for ν-MSR, we define a function $nest_\nu$, that measures the nesting of restrictions (occurrences of the operator ν) in a term.

Definition 5. We define $nest_\nu(M)$ by structural induction on M:

- $nest_\nu(A) = nest_\nu(\mathbf{0}) = 0$,
- $nest_\nu(M_1 + M_2) = max(nest_\nu(M_1), nest_\nu(M_2))$,
- $nest_\nu(\nu a.M) = 1 + nest_\nu(M)$.

We take $depth(M) = min\{nest_\nu(M') \mid M \equiv M'\}$. A ν-MSR is *k-bounded* if $depth(M) \leq k$ for any reachable M, and *depth-bounded* if it is k-bounded for some $k \geq 0$.

As explained in [31], *depth* measures the interdependence of restricted names. The fragment $F = \nu a_1 . \ldots . \nu a_n . \nu a . (p(a, a_1), \ldots, p(a, a_n))$, satisfies $nest_\nu(F) = n+1$ and is equivalent to $F' = \nu a . (\nu a_1 . p(a, a_1), \ldots, \nu a_n . p(a, a_n))$, which satisfies $nest_\nu(F') = 2$. In fact, it can be easily checked that $depth(F) = 2$. Therefore, fragments that are equivalent modulo $\equiv$ do not necessarily have the same nesting, because $\equiv$ allows to rearrange the binding operator. However, $\hat{\equiv}$ does not allow those rearrangements, so that the following holds.

Lemma 1. *If $F \hat{\equiv} G$ then $nest_\nu(F) = nest_\nu(G)$.*

Proof. Obvious.

As in [31], we use the graph-theoretic interpretation of fragments. A fragment can be seen as a hypergraph with its atomic formulae as vertices, and its names as arcs, that link all the formulae that contain that name.

Definition 6. For a term $M \equiv \nu\tilde{a}.(A_1 + \ldots + A_m)$ we define the hypergraph $\mathcal{G}(M) = (V, E, inc)$, where $V = \{A_1, \ldots, A_m\}$, $E = \tilde{a}$ and for $e \in E$, $inc(e)$ is the set of atomic formulae in V in which e occurs.

Fragments correspond to connected components. $M_1 \equiv M_2$ implies that $\mathcal{G}(M_1)$ and $\mathcal{G}(M_2)$ are isomorphic. For $F = \nu a_1 \ldots \nu a_n \nu a.(p(a, a_1), \ldots, p(a, a_n))$ and $F' = \nu a.(\nu a_1.p(a, a_1), \ldots, \nu a_n(p(a, a_n)))$, since $F \equiv F'$ the hypergraphs obtained for them are isomorphic (see Fig. 1). Another example is shown in Fig. 2.

A *path* in a hypergraph is a finite sequence $\rho = A_1 a_1 A_2 a_2 \cdots a_n A_{n+1}$ with $A_i, A_{i+1} \in inc(a_i)$ for each $i \in \{1, \ldots, n\}$. The length of ρ is $|\rho| = n$, and ρ is *simple* whenever $a_i \neq a_j$ for $i \neq j$. A simple path in the hypergraph in Fig. 1 is for instance

$$\rho_1 = p(a, a_1)\ a_1\ p(a, a_1)\ a\ p(a, a_2)\ a_2\ p(a, a_2)$$

with length 3. Any attempt to extend that simple path results in a path that is no longer simple (since a and a_2 already occur in it). Indeed, it can be checked that the length of every single path is at most 3. In the case of the hypergraph in Fig. 2 the longest simple path, with length 4, is

$$\rho_2 = p(a_1, a_2)\ a_1\ p(a_1, a_2)\ a_2\ p(a_2, a_3)\ a_3\ p(a_3, a_4)\ a_4\ p(a_3, a_4)$$

In [39] we proved, by following the same steps as in [31], that a ν-MSR is depth-bounded if and only if the length of the simple paths in every reachable fragment is bounded. Next we define an order over terms, that will endow ν-MSR with a well-structure.

Definition 7. We define $\sqsubseteq_F$ as the least binary relation over fragments such that $A \sqsubseteq_F A$, $\nu a.(\sum_{i=1}^{n} F_i) \sqsubseteq_F \nu a.(\sum_{i=1}^{n} G_i + \sum_{i=1}^{n'} G'_i)$ provided $F_i \sqsubseteq_F G_i$ for all $i \in \{1, \ldots, n\}$, and $F \sqsubseteq_F G$ provided $F \equiv F' \sqsubseteq_F G' \equiv G$. We also define $M_1 \sqsubseteq M_2$ if $M_i \equiv \sum_{j=1}^{n_i} F_j^i$, $n_1 \leq n_2$ and $F_i^1 \sqsubseteq_F F_i^2$ for $i \in \{1, \ldots, n_1\}$.

The order $\sqsubseteq$ over terms can be seen as the multiset order induced by $\sqsubseteq_F$ over fragments. In turn, $\sqsubseteq_F$ can be intuitively characterized using standard forms.

Lemma 2. *Given two fragments F and G, $F \sqsubseteq_F G$ holds if and only if $F \equiv \nu\tilde{a}.(A_1 + \ldots + A_m)$ and $G \equiv \nu\tilde{a}.(A_1 + \ldots + A_m + M)$ for some $M \in \mathcal{M}$ without restrictions.*

Proof. Let F and G such that $F \sqsubseteq_F G$. We proceed by induction on the rules used to derive $F \sqsubseteq_F G$. For $F = A \sqsubseteq_F A = G$ it is trivial. Suppose now that $F = \nu a.(F_1 + \ldots + F_n)$ and $G = \nu a.(G_1 + \ldots + G_n + G'_1 + \ldots + G'_m)$ with $F_i \sqsubseteq_F G_i$. The induction hypothesis tells us that $F_i \equiv \nu\tilde{a}_i.(\sum A_j^i)$ and $G_i \equiv \nu\tilde{a}_i.(\sum A_j^i + M_i)$. Then, $F \equiv \nu a, \tilde{a}_1, \ldots, \tilde{a}_n.(\sum A_j^i)$ and $G \equiv \nu a, \tilde{a}_1, \ldots, \tilde{a}_n.(\sum A_j^i + \sum G'_i + \sum M_i)$, which satisfy the thesis. Finally, if $F \equiv F' \sqsubseteq_F G' \equiv G$ the induction

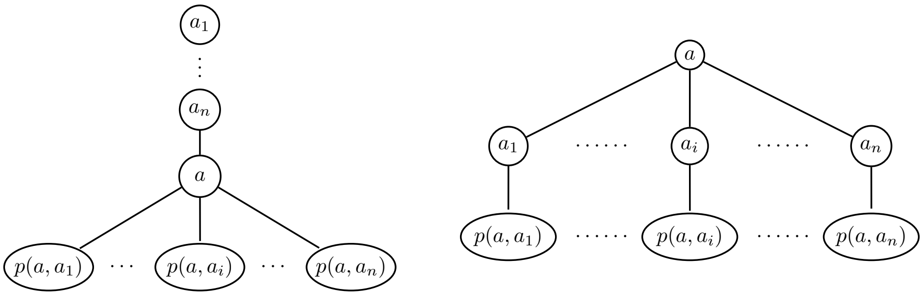

Figure 3: Trees of the fragments F (left) and F' (right) in Fig. 1

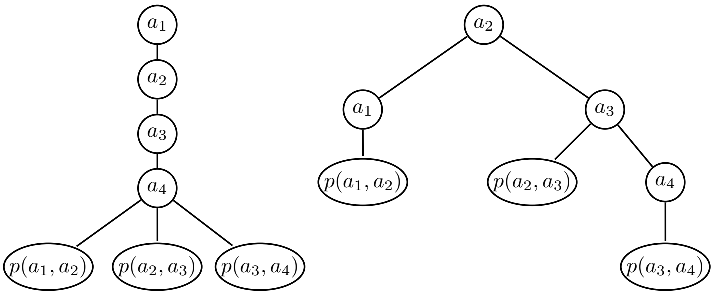

Figure 4: Trees of the fragments F (left) and F' (right) in Fig. 2

hypothesis tells us that $F' \equiv \nu\tilde{a}.(A_1+\ldots+A_m)$ and $G' \equiv \nu\tilde{a}.(A_1+\ldots+A_m+M)$ and because $\equiv$ is transitive, the same holds for F and G.

Conversely, if $F \equiv \nu\tilde{a}.(A_1+\ldots+A_m)$ and $G \equiv \nu\tilde{a}.(A_1+\ldots+A_m+M)$, trivially $A_i \sqsubseteq_F A_i$, so that $\nu\tilde{a}.(A_1+\ldots+A_m) \sqsubseteq_F \nu\tilde{a}.(A_1+\ldots+A_m+M)$ and we can conclude by rule ($\equiv$) that $F \sqsubseteq_F G$.

Let us see that depth-bounded ν-MSR are WSTS with respect to that order. In order to see that the order is a wqo, we map fragments to trees as follows.

Definition 8. Let Δ be the set of names and atomic formulae. We define $\mathcal{T}$ that maps fragments to trees in $\mathcal{T}(\Delta)$ as follows:

- $\mathcal{T}(A) = A$,
- $\mathcal{T}(\nu a.(F_1+\ldots+F_n)) = (a, \{\mathcal{T}(F_1),\ldots,\mathcal{T}(F_n)\})$.

Figure 3 and Fig. 4 show the trees corresponding to the fragments considered in Fig. 1 and Fig. 2, respectively. The following lemma is easy to prove.

Lemma 3. $nest_\nu(F) = height(\mathcal{T}(F))$

PROOF. Clearly, $nest_\nu(A) = 0 = height(A) = height(\mathcal{T}(A))$. For a fragment $F = \nu a.(F_1+\ldots+F_n)$, $nest_\nu(F) = 1 + max\{nest_\nu(F_i) \mid i = 1,\ldots,n\}$. By the induction hypothesis, $nest_\nu(F_i) = height(\mathcal{T}(F_i))$. Then, $nest_\nu(F) = 1 + max\{height(\mathcal{T}(F_i)) \mid i = 1,\ldots,n\} = height((a, \{\mathcal{T}(F_1),\ldots,\mathcal{T}(F_n)\})) = height(\mathcal{T}(F))$.

Moreover, the corresponding orders are preserved by $\mathcal{T}$ in the following sense:

Proposition 2. *If $\mathcal{T}(F_1) \preceq \mathcal{T}(F_2)$ then $F_1 \sqsubseteq_F F_2$.*

Proof. We proceed by induction on the rules used to derive $\mathcal{T}(F_1) \preceq \mathcal{T}(F_2)$. If $\mathcal{T}(F_1) = A \preceq A = \mathcal{T}(F_2)$ then $F_1 = F_2 = A$ and trivially, $F_1 \sqsubseteq_F F_2$. Otherwise, $\mathcal{T}(F_1) = (a, \{T_1, \ldots, T_n\}) \preceq (a, \{T'_1, \ldots, T'_{n'}\}) = \mathcal{T}(F_2)$ and $\{T_1, \ldots, T_n\} \sqsubseteq \{T'_1, \ldots, T'_{n'}\}$, so that we can assume without loss of generality that $T_i \preceq T'_i$ for all $i \in \{1, \ldots, n\}$. Then, $F_1 = \nu a.(F^1_1 + \ldots + F^1_n)$ with $\mathcal{T}(F^1_i) = T_i$, and $F_2 = \nu a.(F^2_1 + \ldots + F^2_{n'})$ with $\mathcal{T}(F^2_i) = T'_i$. The induction hypothesis tells us that $F^1_i \sqsubseteq_F F^2_i$, which allows us to conclude that $F_1 \sqsubseteq_F F_2$.

We denote by $\mathcal{F}_n$ the set of fragments with depth less or equal than n, and analogously, we define $\mathcal{M}_n$ as the set of terms with depth less or equal than n. Let us consider a depth-bounded ν-MSR, with depth n. Let $Id_n = \{a_0, ..., a_n\}$ and $\Delta(n)$ the set Id_n together with all the atomic formulae that can be built using names in Id_n and predicates in the ν-MSR (that is, in its initial term and in its rules). Notice that $\Delta(n)$ is a finite set. We will assume that fragments do not contain free names (otherwise, we just have to add those free names to Id_n).

Lemma 4. *If $F \in \mathcal{F}_n$ then there is $F' \equiv F$ such that $\mathcal{T}(F') \in \mathcal{T}(\Delta(n))_n$.*

Proof. Since $F \in \mathcal{F}_n$ there is $F'' \equiv F$ such that $height(\mathcal{T}(F'')) = nest_\nu(F'') = depth(F) \leq n$. The fragment F'' may use names not in Id_n, but we can α-convert them. Indeed, let $F' = F''\sigma^0$, where each σ^i is defined recursively: $A\sigma^i = A$ for A atomic, $\nu a.(F_1 + ... + F_m)\sigma^i = \nu a_i.(F_1\{a/a_i\}\sigma^{i+1} + ... + F_m\{a/a_i\}\sigma^{i+1})$. By induction we can check that $0 \leq i \leq n - nest_\nu(G)$ holds in every recursive call. In particular, $0 \leq i \leq n$, so that all the names used are in Id_n, which implies that $\mathcal{T}(F') \in \mathcal{T}(\Delta(n))_n$ and we conclude.

Proposition 3. *$(\mathcal{F}_n, \sqsubseteq_F)$ and $(\mathcal{M}_n, \sqsubseteq)$ are wqos.*

Proof. Let (F_i) be an infinite sequence of fragments in $\mathcal{F}_n$. By the previous lemma, for each i there is $F'_i \equiv F_i$ such that $\mathcal{T}(F'_i) \in \mathcal{T}(\Delta(n))_n$. Since $\Delta(n)$ is finite, the equality is a wqo in it, and therefore so is $\mathcal{T}(\Delta(n))_n$. Then there are $i < j$ such that $\mathcal{T}(F'_i) \preceq \mathcal{T}(F'_j)$. By Prop. 2, $F'_i \sqsubseteq_F F'_j$. Finally, because $F_i \equiv F'_i$ and $F_j \equiv F'_j$ we can conclude that $F_i \sqsubseteq_F F_j$. $(\mathcal{M}_n, \sqsubseteq)$ is also a wqo because $\sqsubseteq_F$ is a wqo and $\sqsubseteq$ is the multiset order induced by $\sqsubseteq_F$.

The proof of the previous result makes use of the fact that the order $\preceq$ in trees is a wqo. Therefore, if a ν-MSR is depth-bounded by n, then the set of reachable terms is contained in $\mathcal{M}_n$, which is a wqo with its order. In order to see that they are WSTS, we still have to see that the transition relation is monotonic with respect the considered order.

Proposition 4. *Depth-bounded ν-MSR are strict WSTS.*

Proof. We have to see that the defined order is strictly monotonic with respect to the rewriting relation. This fact follows from the compatibility of the transition relation with respect to the standard normal form (Prop. 1) and Lemma 2.

Since termination and boundedness are decidable for strict WSTS [21, 2], we obtain the following result as a corollary.

Corollary 1. *Boundedness and termination are decidable for the class of depth-bounded ν-MSR.*

In [31], where depth-bounded π-calculus processes are proved to be WSTS, the decidability of termination for the π-calculus was already discussed. For boundedness, it is enough to prove that the WSTS is actually strict, as done above. However, decidability of coverability was not considered. The most common way to face this problem when WSTS are studied in the literature, is by performing a backwards analysis, that computes a finite representation of the set $\{s \in S \mid s \rightarrow^* s' \geq s_f\}$, where s_f is the state whose coverability we want to decide, and then checking whether the initial state is in that set. In order to perform this backwards analysis, the WSTS must satisfy the effective pred-basis property.

In our case, when the system is k-bounded, we need to consider the set $\{M' \in \mathcal{M}_k \mid M' \rightarrow M'' \sqsupseteq M\}$ for any M. If the bound k is known we can compute that set using standard techniques (notice that in particular it is possible to decide whether a given M is in $\mathcal{M}_k$). Then the generic backwards analysis in [2, 21] is feasible and coverability is decidable.

However, knowing a bound for the depth of a ν-MSR is not always possible, that is, it may be the case that the system is depth-bounded, but we do not know the particular bound k. In this case, the generic forward algorithms deciding boundedness and termination are still applicable, so that Cor. 1 holds even if the bound is not known. However, it is no longer possible to compute the set of predecessors, so that we do not have the effective pred-basis property. In [43] they solve this problem for the π-calculus by introducing an ADL (Adequate Domain of Limits) to represent downward closed sets, and performing a forward analysis deciding coverability. Instead of defining an ADL or any other representation for downward closed sets [23, 24], we consider the algorithm proposed in [25], which does not need a defined ADL nor a known bound for the depth.

Corollary 2. *Coverability is decidable for depth-bounded ν-MSR, even if the bound is not known.*

Proof. It is enough to apply the abstract interpretation forward algorithm in [25] (Algorithm 1). Effectiveness of (2) in the algorithm (Prop. 6 and Cor. 1) is proved assuming the WSTS satisfies the "effective Pred-basis" property. However, Cor. 1 still holds even if such property is not satisfied, as in our case, so the algorithm is still correct.

In Section 6 we will prove that π-calculus processes can be directly encoded into ν-MSR. Moreover, depth-bounded π-calculus processes correspond to depth-bounded ν-MSR, so that they are WSTS, which was already known in [31]. The novelty of our results lies in the fact that we can apply Prop. 4 to other formalisms that can be easily encoded within ν-MSR. This is the case for $p\nu$-PN. In particular, as a corollary of the results in the next section, we can prove that ν-MSR are Turing-complete. Instead, we prove it directly with an encoding of Counter Machines, which will give us also the undecidability of depth-boundedness.

A Minsky machine with two counters [34], or Two Counter Machine (TCM) consists of a finite set of states $\mathcal{S} = \{s_0, \ldots, s_k\}$ and a finite set of instructions $\mathcal{I} = Inc \cup Dec \cup Zero$: $Inc(i, s, t) \in Inc$ increases counter c_i by one when at state s, and moves to state t; $Dec(i, s, t) \in Dec$ when in state s decreases the counter c_i by one and moves to state t if $c_i > 0$; $Zero(i, s, t)$ moves to state t when in s and $c_i = 0$. Configurations are of the form $\langle s, n_1, n_2\rangle$, where $s \in \mathcal{S}$ and n_i is the value of the counter c_i. The boundedness problem for TCM is that of deciding if the values of the counters are bounded in every reachable configuration, which is undecidable [34].

Proposition 5. *ν-MSR are Turing complete and depth-boundedness is undecidable for ν-MSR.*

Proof. We simulate TCM by means of ν-MSR. Moreover, we do it in such a way that we reduce the boundedness problem for TCM to depth-boundedness in ν-MSR.

We consider 0-ary predicates $s_0, ..., s_k$, unary predicates z_i, l_i and binary predicates c_i, for $i = 1, 2$. We represent a configuration $\langle s, n_1, n_2\rangle$ by means of the ν-MSR term $[\![\langle s, n_1, n_2\rangle]\!]$

$$s + \sum_{i=1}^{2} \nu a_0^i, ..., a_{n_i}^i.(z_i(a_0^i), c_i(a_0^i, a_1^i), ..., c_i(a_{n_i-1}^i, a_{n_i}), l_i(a_{n_i}))$$

with $a_l^i \neq a_r^i$ for $l \neq r$. Instructions can be simulated as rules as follows:

$Inc(i, s, t) : s, l_i(x) \rightarrow \nu a.(t, c_i(x, a), l_i(a))$,

$Dec(i, s, t) : s, c_i(x, y), l_i(y) \rightarrow t, l_i(x)$,

$Zero(i, s, t) : s, z_i(x), l_i(x) \rightarrow t, z_i(x), l_i(x)$.

This simulation establishes an isomorphism between the reachability graphs, that is, $[\![\ ]\!]$ is a bijection between the reachable configurations such that $C_1 \rightarrow C_2$ in the TCM if and only if $[\![C_1]\!] \rightarrow [\![C_2]\!]$. Moreover, the depth of $[\![\langle s, n_1, n_2\rangle]\!]$ is $max\{n_1, n_2\} + 1$, so that the TCM is bounded iff its simulation is depth-bounded, and we conclude.

To conclude this section let us study the complexity of the problems we have proved to be decidable. In the next section we will prove that ν-MSR is equivalent to $p\nu$-PN, Petri nets in which tokens are tuples of pure names. Moreover, monadic ν-MSR (in which every predicate has arity at most 1) is equivalent to ν-PN (for which tokens are pure names). This fact allows us to obtain the following hardness results.[2]

Proposition 6. *Coverability, boundedness and termination are not primitive recursive for depth-bounded ν-MSR.*

Proof. In the first place, notice that a monadic ν-MSR is 1-bounded. Then, it is enough to consider that those properties are not primitive recursive for ν-PN [40], which are equivalent to them.

5. ν-MSRs and $p\nu$-PNs

Let us now consider a class of Petri nets with name creation. A $p\nu$-PN [37] is a Petri net in which tokens are tuples of pure names. Arcs are labelled by tuples of variables (or multiset of such tuples, if we allow weights) that specify how tokens flow from preconditions to postconditions. Variables are taken from a set Var. Some of the variables in postarcs can be in the set of special variables $\Upsilon \subset \mathit{Var}$ that can only be instantiated to names that do not occur in the current marking, thus creating fresh names. We use ν, ν', $\nu_1, \ldots$ to range over Υ. We take $\mathcal{L} = \bigcup_{i>0} \mathit{Var}^i$, that is, the set of tuples of variables of arbitrary length. We will sometimes use set notation for tuples, so that we will write, for instance, $x \in (x, y)$. Moreover, we will use an arbitrary set Id of names.

Definition 9. A $p\nu$-PN is a tuple $N = (P, T, F)$, where P and T are finite disjoint sets of elements called places and transitions, respectively,

$$F : (P \times T) \cup (T \times P) \to \mathcal{MS}(\mathcal{L})$$

is such that for every $t \in T$, $\mathit{pre}(t) \cap \Upsilon = \emptyset$, and $\mathit{post}(t) \setminus \Upsilon \subseteq \mathit{pre}(t)$, where $\mathit{pre}(t) = \bigcup_{p \in P}\{x \in \mathit{Var} \mid x \in \ell \in F(p,t)\}$, $\mathit{post}(t) = \bigcup_{p \in P}\{x \in \mathit{Var} \mid x \in \ell \in F(t,p)\}$ and $\mathit{Var}(t) = \mathit{pre}(t) \cup \mathit{post}(t)$.

Let us denote by $\mathcal{T}$ the set of tuples of names of arbitrary length, that is, $\mathcal{T} = \bigcup_{i>0} \mathit{Id}^i$. The tokens of a p$\nu$-PN are taken from $\mathcal{T}$. We will use φ, φ', $\varphi_1, \ldots$ to range over tokens.

Definition 10. A *marking* of a pν-PN $N = (P, T, F)$ is any $M : P \to \mathcal{MS}(\mathcal{T})$.

We define $\mathit{Id}(M) = \bigcup_{p \in P}\{a \in \mathit{Id} \mid a \in \varphi \in M(p)\} \subset \mathit{Id}$, the set of all the names appearing in some token in some place, according to the marking M.

Transitions are fired with respect to a mode, that chooses which tokens are taken from preconditions and which are put in postconditions. Given a transition t of a net N, a mode of t is a mapping $\sigma : \mathit{Var}(t) \to \mathit{Id}$, that instantiates each variable involved in the firing of t to an identifier. We will use $\sigma, \sigma', \sigma_1 \ldots$ to range over modes. We extend modes to tuples of variables by taking $\sigma((x_1, \ldots, x_n)) = (\sigma(x_1), \ldots, \sigma(x_n))$.

[2] Though we will not prove that equivalence until the next section, we prefer to "look ahead" and consider here this hardness result.

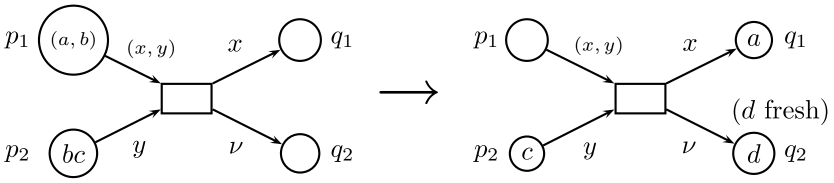

Figure 5: A simple *pν*-PN

Definition 11. Let N be a pν-PN, M a marking of N, t a transition of N and σ a mode of t. We say t is *enabled* with mode σ if $\sigma(\nu) \notin Id(M)$ for all $\nu \in Var(t) \cap \Upsilon$, and $\sigma(F(p,t)) \subseteq M(p)$ for all $p \in P$. The reached state of N after the *firing* of t with mode σ is the marking M', given by

$$M'(p) = (M(p) - \sigma(F(p,t))) + \sigma(F(t,p)) \ \ \forall p \in P$$

We will write $M \xrightarrow{t(\sigma)} M'$ if M' is reached from M when t is fired with mode σ. We also define the relations $\rightarrow$ and $\rightarrow^*$, as usual. Fig. 5 depicts a simple example of a pν-PN and the firing of its only transition. Notice that the transition can be fired because the second component of the pair (a,b) in p_1 matches one of the names in p_2, as demanded by the labels in the arcs.

In order to capture the intuition that the names in Id are pure, we work modulo $\equiv_\alpha$, which allows consistent renaming of names in markings. Accordingly, the order $\sqsubseteq_\alpha$ that induces coverability for pν-PNs is defined as follows: $M \sqsubseteq_\alpha M'$ if there is an injection $\iota : Id(M) \rightarrow Id(M')$ such that for every place $p \in P$, $\iota(M(p)) \subseteq M'(p)$, where $\iota((a_1,\ldots,a_n)) = (\iota(a_1),\ldots,\iota(a_n))$.

Proposition 7. *For any pν-PN N we can compute a ν-MSR $[\![N]\!]$ such that N and $[\![N]\!]$ are isomorphic (as transition systems).*

Proof. Let $N = (P,T,F)$ with initial marking M_0 be a *pν*-PN. For every $t \in T$, if $\tilde{\nu}$ is a sequence formed by the special variables in postarcs of t, let us take any sequence (of the same length) of arbitrary names $\tilde{a}$, and let us define the rule

$$[\![t]\!] : \sum_{p\in P} \sum_{\tilde{x}\in F(p,t)} p(\tilde{x}) \rightarrow \nu\tilde{a}. \sum_{p\in P} \sum_{\tilde{x}\in F(t,p)} p(\tilde{x}\{\tilde{a}/\tilde{\nu}\})$$

For every marking M with $\tilde{b} = Id(M)$, we define $[\![M]\!]$ as the ν-MSR term $\nu\tilde{b}.(\sum_{p\in P}\sum_{\tilde{a}_i\in M(p)} p(\tilde{a}_i))$. Then, we define $\mathcal{R} = \{[\![t]\!] \mid t \in T\}$ and $[\![N]\!] = \langle \mathcal{R}, [\![M_0]\!]\rangle$. For two markings M_1 and M_2 with $M_1 \rightarrow M_2$, it holds that $[\![M_1]\!] \rightarrow [\![M_2]\!]$. On the other hand, for two ν-MSR terms M_1 and M_2 such that $M_1 \rightarrow M_2$, Prop. 1 tells us that $M_i \equiv \nu\tilde{a}_i(A^i_1 + \ldots + A^i_{n_i} + M)$, so that M_i is equivalent to a $[\![M'_i]\!]$ for some markings M'_i and M'_2. Moreover, $M'_1 \rightarrow M'_2$ and the thesis follows.

For instance, consider the *pν*-PN in Fig. 5. The previous construction yields the ν-MSR given by the rule $t : p_1(x,y), p_2(y) \rightarrow \nu a.(q_1(x), q_2(a))$. The initial marking is represented by $\nu a, b.(p_1(a,b), p_2(b), p_2(c))$, which evolves to $\nu a, c, d.(q_1(a), p_2(c), q_2(d))$.

Therefore, $p\nu$-PNs can be just thought of as a graphical representation of ν-MSRs. However, since $p\nu$-PNs lack a name binding operator, intuitively they always work with terms in their standard normal form. Indeed, for a marking M, the term $[\![M]\!]$ is in standard form. Let us now prove the converse result.

Proposition 8. *For any ν-MSR S we can compute a $p\nu$-PN S^* such that S and S^* are isomorphic (as transition systems).*

Proof. Let $S = \langle \mathcal{R}, M_0 \rangle$ be a ν-MSR using predicates in $\mathcal{P}'$ (finite). We define $G(S) = (\mathcal{P}', \mathcal{R}, F, [\![M_0]\!])$ as follows. Let $t : \sum_{i=1}^{n} p_i(\tilde{x}_i) \to \nu\tilde{a}.(\sum_{i=1}^{m} q_i(\eta_i))$ be a rule in $\mathcal{R}$. We assume for the sake of readability that no name appears in the tuples of the left handside of the rule, and that the only names appearing in the right handside are those in $\tilde{a}$. Let $\tilde{\nu}$ be a sequence (of the same length of $\tilde{a}$) of pairwise different special variables. We define $F(p,t) = \sum\limits_{p=p_i} \tilde{x}_i$ and $F(t,p) = \sum\limits_{p=q_i} \eta_i\{\tilde{\nu}/\tilde{a}\}$. For a ν-MSR term $M \equiv \nu\tilde{a}.(\sum_{i=1}^{n} p_i(\tilde{a}_i))$ we define M^* as the marking given by $M^*(p) = \{\tilde{a}_i \mid p = p_i\}$. As in the previous result, for two terms M_1 and M_2, thanks to Prop. 1, it holds that $M_1^* \to M_2^*$. Moreover, for two markings M_1 and M_2 such that $M_1 \to M_2$, $M_i = M_i'^*$ for some terms M_1' and M_2', with $M_1' \to M_2'$.

We have seen that ν-MSR are Turing-complete (and therefore, so are $p\nu$-PN, though this fact was already proved in [37]). It is easy to devise some decidable subclasses of ν-MSR. For instance, if a ν-MSR S is monadic, that is, if atomic formulae have the form $p(\eta)$, then the $p\nu$-PN S^* obtained in Prop. 8 is a ν-PN [36], that is, a Petri net in which tokens are pure names. In [40] we proved that coverability, termination and boundedness are decidable for them, so that they are also decidable for monadic ν-MSRs. Moreover, if we consider a ν-MSR with only binary predicates and so that for every formula $p(a,b)$ there are only finitely many b_i such that $p(a,b_i)$ appears in any reachable term, then S^* is a restricted binary $p\nu$-PN [37], for which these properties are also decidable. We claim that these results could have been obtained directly for the restricted classes of ν-MSRs, so that the corresponding results for Petri nets could have been obtained as a corollary instead. Finally, let us remark that in the case of ordinary P/T nets (that are a subclass of ν-PNs, in which only one element of *Id* is used) our translation yields a ν-MSR that coincides with the rewriting logic specification obtained in [42].

Let us now use the subclass of depth-bounded ν-MSR to obtain a new decidable subclasses of $p\nu$-PN. We say a $p\nu$-PN is k-bounded if for any reachable marking M and for any sequence $A_1, \ldots, A_n$ of tokens in M such that for every i, there is a different name a_i in A_i and A_{i+1}, then necessarily $n \leq k$. A $p\nu$-PN is depth-bounded if it is k-bounded for some k. Depth-bounded ν-PN correspond to depth-bounded ν-MSR.

Corollary 3. *Depth-bounded $p\nu$-PN are strict WSTS.*

Therefore, boundedness and termination are decidable for the class of depth-bounded $p\nu$-PN, as well as coverability. However, in the case of coverability one needs to be a bit more careful. Indeed, since coverability is decidable for depth-bounded ν-MSR, so is coverability for $p\nu$-PN when considering the order induced by the order $\sqsubseteq$ between ν-MSR terms. More precisely, if for two markings M_1 and M_2 we define $M_1 \ll M_2 \Leftrightarrow [\![M_1]\!] \sqsubseteq [\![M_2]\!]$, then coverability induced by $\ll$ (that we can call $\ll$-coverability) is decidable directly from Prop. 7 and Corollary 3.

However, the natural order $\sqsubseteq_\alpha$ in $p\nu$-PN is slightly different from $\ll$, so that coverability is different from $\ll$-coverability. Intuitively, $M_1 \sqsubseteq_\alpha M_2$ holds when M_2 has more tokens than M_1 (possibly renaming some names).

Example 1. Let M_1 be a marking with only two tokens a and b in a place p. Let M_2 be the result of adding a token (a,b) in q. Then it holds that $M_1 \sqsubseteq_\alpha M_2$ (without need to rename), but $M_1 \not\ll M_2$ Indeed, $[\![M_1]\!] = \nu a, b.(p(a), p(b)) \equiv \nu a.p(a) + \nu b.p(b) \equiv F + F$ with $F = \nu a.p(a)$, and $[\![M_2]\!] = \nu a, b.(p(a), p(b), q(a,b)) \equiv G$ with $G = \nu a.(p(a), \nu b.(p(b), q(a,b))$. Since $[\![M_1]\!]$ is composed by two fragments but $[\![M_2]\!]$ is composed by only one fragment, we have $[\![M_1]\!] \not\sqsubseteq [\![M_2]\!]$ and, therefore, $M_1 \not\ll M_2$. Therefore, if we want to check that M_1 can be $\ll$-covered, reaching M_2 would not provide us a positive answer.

Fortunately, we still have decidability for coverability with our natural order.

Proposition 9. *Termination, boundedness and coverability are decidable for depth-bounded $p\nu$-PN.*

Proof. Termination and boundedness follow immediately from Cor. 3. From that result we also obtain decidability of $\ll$-coverability. Let us see that coverability of a marking M can be decided by solving a finite number of $\ll$-coverability problems. Let $[\![M]\!] \equiv F_1 + ... + F_n$ and each F_i has a different name a_i in the topmost restriction. Let M' such that $\nu\tilde{a}.M'$ is equivalent to $F_1 + ... + F_n$ in standard form, without renaming the a_is. We consider a new 0-ary predicate ok and for each $1 \leq m \leq n$ we consider a new predicate $link_m$ of arity m. Intuitively, we will add an atomic formula $link_m(a_1, \ldots, a_m)$ in order to merge some fragments $F_{i_1}, ..., F_{i_m}$ into a single fragment. Notice that we will only merge up to n fragments (otherwise, if we could merge arbitrarily many fragments, the resulting ν-MSR could no longer be depth-bounded).

Let $\mathcal{I} = (I_i)_{i=1}^{l}$ be a partition of $\{1, ..., n\}$, that is, $I_i \neq \emptyset$ for all i, $\bigcup_{i=1}^{l} I_i = \{1, ..., n\}$ and $I_i \cap I_j = \emptyset$ for each $i \neq j$. Assuming $I_i = \{j_1^i, ..., j_{n_i}^i\}$, so that $|I_i| = n_i$, we consider the following new rule $ok, M' \to M', part(\mathcal{I})$, where $part(\mathcal{I}) = link_{n_1}(\tilde{a}_i), ..., link_{n_l}(\tilde{a}_l)$ and $\tilde{a}_i = (a_{j_1^i}, ..., a_{j_{n_i}^i})$. Then, the marking M can be covered from M_0 in the $p\nu$-PN N iff there is some partition $\mathcal{I}$ such that the ν-MSR term $M', part(\mathcal{I})$ can be covered from $[\![M_0]\!], ok$ in the ν-MSR $[\![N]\!]$.

In Example 1, the construction would yield two new rules, one for the partition $\mathcal{I}_1 = \{I\}$ with $I = \{1, 2\}$, and another one for the partition $\mathcal{I}_2 = \{I_1, I_2\}$

with $I_1 = \{1\}$ and $I_2 = \{2\}$. The rule for $\mathcal{I}_1$ is

$$ok, p(a), p(b) \rightarrow p(a), p(b), link_2(a, b)$$

and the rule for $\mathcal{I}_2$ is

$$ok, p(a), p(b) \rightarrow p(a), p(b), link_1(a), link_2(b)$$

If we reached M_2 in the $p\nu$-PN, then we can reach the ν-MSR term

$$ok, \nu a, b.(p(a), p(b), q(a, b))$$

that can evolve with the first of the new rules to

$$\nu a, b.(p(a), p(b), q(a, b), link_2(a, b))$$

which is greater than $\nu a, b.(p(a), p(b), q(a, b))$, while $[\![M_2]\!]$ is not.

6. ν-MSRs and the π-calculus

Let us see that ν-MSRs can simulate any π-calculus process in a simple way. We use the monadic version of the π-calculus used in [32, 31, 41], with parameterized recursion. The prefixes of the π-calculus are defined by $\pi ::= x\langle y\rangle \mid x(y) \mid \tau$. In order to define parameterized recursive processes we use process identifiers K, K'.... The set of the π-calculus processes is defined by

$$P ::= \sum_{i=1}^{n} \pi_i.P_i \mid P_1 \mid P_2 \mid \nu a.P \mid K\lfloor\tilde{a}\rfloor$$

The empty sum (with $n = 0$) is denoted as $\mathbf{0}$. As usual, we identify processes up to $\equiv$, which is the least congruence that allows α-conversion of bound names, such that $+$ and $\mid$ are commutative and associative with $\mathbf{0}$ as identity, and the following equations hold: $\nu a.0 \equiv 0$, $\nu a.\nu b.P \equiv \nu b.\nu a.P$ and $\nu a.(P \mid Q) \equiv \nu a.P \mid Q$, if $a \notin fn(Q)$, where $fn(P)$ is the set of names that occur *free* in P. If a name in P is not free then it is *bound*. As usual, we omit pending $\mathbf{0}$ in the examples. The reaction relation is defined by the following rules:

$$\tau.P + M \rightarrow P \qquad x(y).P + M \mid x\langle z\rangle.Q + N \rightarrow P\{z/y\} \mid Q$$

$$K\lfloor\tilde{a}\rfloor \rightarrow P\{\tilde{a}/\tilde{x}\}, \quad \mathit{if} \quad K(\tilde{x}) := P$$

$$\frac{P \rightarrow P'}{P \mid Q \rightarrow P' \mid Q} \qquad \frac{P \rightarrow P'}{\nu a.P \rightarrow \nu a.P'} \qquad \frac{P \equiv Q \rightarrow Q' \equiv P'}{P \rightarrow P'}$$

We will use the notion of *derivatives* of a process introduced in [32]. For a process P with recursive definitions $K_i(\tilde{x}_i) := P_i$ for $i = 1, \ldots, n$, we define $\mathit{derivatives}(P) = \mathit{der}(P) \cup \bigcup_{i=1}^{n} \mathit{der}(P_i)$, where $\mathit{der}(\mathbf{0}) = \emptyset$, $\mathit{der}(K\lfloor\tilde{a}\rfloor) = \{K\lfloor\tilde{a}\rfloor\}$, $\mathit{der}(\sum_{i=1}^{n} \pi_i.P_i) = \{\sum_{i=1}^{n} \pi_i.P_i\} \cup \bigcup_{i=1}^{n} \mathit{der}(P_i)$ for $n > 0$, $\mathit{der}(P_1 \mid P_2) = \mathit{der}(P_1) \cup \mathit{der}(P_2)$, and $\mathit{der}(\nu a.P) = \mathit{der}(P)$.

The set of derivatives of a process is always finite, and it essentially corresponds to the set of its sequential subprocesses, but disregarding name restriction. As proved in [32], every reachable process can be built up by composing derivatives with its free names renamed.

Proposition 10. [32, Proposition 3] *Let P be a π-calculus process. Every Q reachable from P is structurally congruent to $\nu\tilde{a}.(Q_1\sigma_1 \mid \cdots \mid Q_n\sigma_n)$, where $Q_i \in derivatives(P)$ and $\sigma_i : fn(Q_i) \to fn(P) \cup \tilde{a}$.*

We will heavily rely on this result for our simulation of π-calculus processes by means of ν-MSRs. More precisely, we will consider the finite set of derivatives as predicates. If a derivative p has $\tilde{x}$ as free names, then we will write $p(\tilde{a})$ to represent the derivative $p\{\tilde{x}/\tilde{a}\}$.

Definition 12. Given a process P_0, we define $[\![P]\!] \in \mathcal{M}$ for every P reachable from P_0 recursively as follows:

- $[\![0]\!] = 0$
- $[\![P_1 \mid P_2]\!] = [\![P_1]\!] + [\![P_2]\!]$
- $[\![\nu n.P]\!] = \nu n.[\![P]\!]$
- $[\![D(\tilde{a})]\!] = D(\tilde{a})$ if $D \in \textit{derivatives}(P_0)$.

Lemma 5. *If $P \equiv Q$ then $[\![P]\!] \equiv [\![Q]\!]$.*

Proof. It is easily proved by induction on the rules used to derive $P \equiv Q$, considering that every rule for $\equiv$ in the π-calculus is mimicked by a rule for $\equiv$ in ν-MSR.

Let us now define the set of rules that simulate the behavior of a process.

Definition 13. Given a process P_0 we define the ν-MSR $\langle \mathcal{R}, [\![P_0]\!] \rangle$, where for all D, D_1 and D_2 in *derivatives*(P_0), $\mathcal{R}$ contains:

- $D(\tilde{x}) \to [\![P]\!]$ if $D(\tilde{x}) \equiv \tau.P + M$, or $D(\tilde{x}) = K(\tilde{x})$ with $K\lfloor\tilde{x}\rfloor := P$,
- $D_1(\tilde{x}), D_2(\tilde{y}) \to [\![P\{y/y'\}]\!], [\![Q]\!]$ if $D_1(\tilde{x}) \equiv x(y).P + M$ and $D_2(\tilde{y}) \equiv x\langle y' \rangle.Q + N$.

Notice that if $[\![P]\!]$ is not in standard normal form, we can always convert it to an equivalent term in standard form.

Proposition 11. *Let P_0 be a π-calculus process. Then P_0 and $[\![P_0]\!]$ have isomorphic transition systems.*

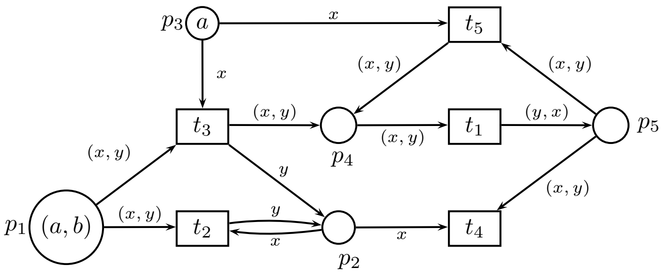

Figure 6: $p\nu$-PN simulating the process in Example 2

Proof. Let us see that if P_1 is a reachable process and $P_1 \to P_2$ then $[\![P_1]\!] \to [\![P_2]\!]$. We prove it by induction on the rules used to derive $P_1 \to P_2$. If $P_1 = K\lfloor\tilde{a}\rfloor$ and $P_2 = P\{\tilde{x}/\tilde{a}\}$ with $K(\tilde{x}) ::= P$ then there is a derivative D such that $[\![P_1]\!] = D(\tilde{a})$. By construction, we have a rule $D(\tilde{x}) \to [\![P]\!]$, that can be applied for $[\![P_1]\!] = D(\tilde{a})$, producing $[\![P_2]\!]$.

Let us now consider the case in which $P_1 = x(y).P_1' + M \mid x\langle z\rangle.P_2' + N$ and $P_2 = P_1'\{y/z\} \mid P_2'$. Then there are derivatives D_1 and D_2 such that $P_1 = D_1\{\tilde{x}_1/\tilde{a}_1\} \mid D_2\{\tilde{x}_2/\tilde{a}_2\}$. By construction, there is a rule $D_1(\tilde{x}_1), D_2(\tilde{x}_2) \to [\![P_1']\!], [\![P_2']\!]$, that can be instantiated for $[\![P_1]\!] = D_1(\tilde{a}_1), D_2(\tilde{a}_2)$, yielding the term $[\![P_2]\!]$.

The rules for parallel composition and restriction are easy to check because they correspond to rules $(+)$ and (ν), respectively. The rule for $\equiv$ follows from the previous lemma and rule $(\equiv)$.

The converse implication is proved analogously by induction on the rules used to derive $[\![P_1]\!] \to [\![P_2]\!]$. Moreover, any term reachable from $[\![P_0]\!]$ is of the form $[\![P]\!]$ for some process P, which concludes the proof.

Example 2. Let us consider $P = \nu b.a\langle b\rangle.b(x) \mid a(y).K\lfloor a, y\rfloor$, where $K(x, y) := y\langle x\rangle$. The set of derivatives of P (renaming its free names for clarity) is $\{p_1, p_2, p_3, p_4, p_5\}$, where $p_1 = x\langle y\rangle.y(z)$, $p_2 = x(y)$, $p_3 = x(y).K\lfloor x, y\rfloor$, $p_4 = K\lfloor x, y\rfloor$, and $p_5 = x\langle y\rangle$. The ν-MSR term corresponding to P is $[\![P]\!] = \nu b.p_1(a, b), p_3(a)$.

These derivatives give rise to the following rules:

$$t_1 : p_4(x, y) \to p_5(y, x)$$

$$t_2 : p_1(x, y), p_2(x) \to p_2(y)$$

$$t_3 : p_1(x, y), p_3(x) \to p_2(y), p_4(x, y)$$

$$t_4 : p_2(x), p_5(x, y) \to 0$$

$$t_5 : p_3(x), p_5(x, y) \to p_4(x, y)$$

In turn, according to Prop. 8, we can write these rules as a $p\nu$-PN, which is depicted in Fig. 6. Its initial marking corresponds to the term $[\![P]\!]$, with a token

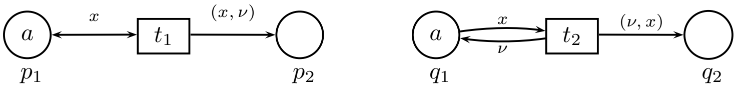

Figure 7: $p\nu$-PNs simulating the processes in Example 3

(a,b) in p_1 and a token a in p_3. Actually, the rules (and the net) obtained are the same for any process with derivatives in $p_1,\ldots,p_5$. Indeed, starting from the process P, one can check that the derivatives p_1 and p_2, or p_3 and p_5, will never be in parallel. Our construction is safe, so that it does consider the reaction rules t_2 and t_5, though they will never be enabled. Thus, any process whose set of derivatives coincides with that of P, is simulated by the same net, though with a different initial marking. Finally, notice that the resulting net does not have any arc labelled with any special variable, so that the names appearing in any reachable markings are taken from the finite set of names in the initial markings. In this situation, the net can be flattened to an equivalent P/T net.

Example 3. Let us consider the processes $P_1 = \nu a.L\lfloor a\rfloor$ and $P_2 = \nu a.K\lfloor a\rfloor$, with $L(x) := \nu b.(x\langle b\rangle \mid L\lfloor x\rfloor)$ and $K(x) := \nu a.(a\langle x\rangle \mid K\lfloor a\rfloor)$. We have $\mathit{derivatives}(P_1) = \{L\lfloor x\rfloor, y\langle z\rangle\}$ and $\mathit{derivatives}(P_2) = \{K\lfloor x\rfloor, y\langle z\rangle\}$. Let us take $p_1(x) = L\lfloor x\rfloor$ and $p_2(y,z) = y\langle z\rangle$ for P_1 and $q_1(x) = K\lfloor x\rfloor$ and $q_2(y,z) = y\langle z\rangle$ for P_2. The only rules obtained are

$$p_1(x) \to \nu b.(p_1(x), p_2(x,b))$$

for P_1 and

$$q_1(x) \to \nu a.(q_1(a), q_2(a,x))$$

for P_2. The corresponding ν-MSR gives rise to two $p\nu$-PNs, which are shown in Fig. 7. Since each process has only two derivatives, the corresponding nets have two places, and since they can only react in one way, only one transition is produced for each.

Finally, if $\mathit{nest}_\nu(P)$ is as defined in [31], we have $\mathit{nest}_\nu(P) = \mathit{nest}_\nu([\![P]\!])$. Then, thanks to the previous results, and as a corollary of Prop. 4 we can obtain the following corollary.

Corollary 4. *Depth-bounded π-calculus processes are strict WSTS. Therefore, coverability, termination and boundedness are decidable for depth-bounded π-calculus processes.*

As already mentioned, the non-strict well-structuredness was already obtained in [31] for depth-bounded π-calculus processes, so that decidability of termination was proved. Now we add for depth-bounded π-calculus processes the decidability of boundedness and coverability. In particular, we can specify the control reachability problem in the π-calculus [15] in terms of coverability. Given a process P and a derivative $D \in \mathit{derivatives}(P)$, the *control reachability* problem is that of deciding whether $P \to^* \nu\tilde{a}.(D(\tilde{b}) \mid Q)$.[3] In [15] a sound but possibly non-terminating procedure deciding control reachability is given for the asynchronous choice-free π-calculus. Here we have seen that it is decidable for depth-bounded processes, even in the synchronous case with choice. Indeed, it is enough to check whether the term $[\![D]\!]$ can be covered from $[\![P]\!]$.

Finally, the complexity of the decision procedures is not discussed in [31]. In [43] the authors argue that since depth-bounded π-calculus processes subsume Petri Nets, then we have an exponential space lower bound for coverability [35], and state that the exact complexity for coverability is open. Since depth-bounded π-calculus processes subsume not only Petri nets, but also ν-PN, we can conclude as follows.

Corollary 5. *Termination, boundedness and coverability are not primitive recursive for depth-bounded π-calculus processes.*

7. Applications to other formalisms

We have seen how the results in ν-MSR can be applied to other concurrent formalisms with name binding, namely the π-calculus and $p\nu$-PN. The same technique can be applied to other models for concurrency. As done for the π-calculus, we could translate spi-calculus processes [1] into ν-MSR, thus bringing together two different approaches for the specification and analysis of security protocols, the spi-calculus and MSR. This translation would be completely analogous to the one followed for the π-calculus, so that we will not do it here.

Now we show it for other models for concurrency: MSR [11], that "extends" ν-MSR with terms over an arbitrary signature, and Mobile Ambients [9].

7.1. ν-MSR and MSR

Now we consider the meta-notation for specification of security protocols MSR. We are interested in translating it into ν-MSR in order to apply our techniques to MSR. As we defined previously, in ν-MSR an atomic formula is of the form $p(\eta_1, \ldots, \eta_n)$, where p is a predicate symbol and for all $i \in \{1, \ldots, n\}$, η_i is either a variable or a name. The atomic formulae of ν-MSR correspond to facts $p(t_1, \ldots, t_n)$ of MSR, in which the predicates are not applied to names, but to terms. The set of terms in MSR is defined as the free algebra over a signature of symbols of a given arity. The rules are of the form $F_1, \ldots, F_n \to \exists a_1, \ldots, a_i(G_1, \ldots, G_j)$, where $F_1, \ldots, F_n, G_1, \ldots, G_n$ are facts and $a_1, \ldots, a_n$ are new names. Intuitively, a state S can evolve to another state S' by applying this rule if S contains the facts $F_1, \ldots, F_n$, and S' is obtained from S by removing $F_1, \ldots, F_n$ and adding $G_1, \ldots, G_n$ where $a_1, \ldots, a_n$ are names.

Let us show that ν-MSRs can simulate any MSR. For that purpose, we define two functions which translate MSR terms and facts, respectively, into ν-MSR terms. Essentially, we have to represent the trees which define the terms of an

[3] This is a slightly different, yet equivalent, definition to the one used in [15].

MSR. For that purpose we identify each subterm of a term with a pure name which will be in the set of parameters of the term and the subterm.

Definition 14. Given an MSR term $t = f(t_1 \dots t_n)$ and a name $a \notin fn(t)$, we define the ν-MSR term $[\![t]\!]^a$ as

$$\nu\tilde{a}.(f(\alpha_1, \dots, \alpha_n, a) + \sum_{i \in U} [\![t_i]\!]^{a_i})$$

where $I = \{i \in 1, \dots, n \mid t_i \in Id \cup Var\}$, $U = \{1, \dots, n\} - I$, $a_i \in Id \setminus fn(t_i)$ is a different name for each $i \in U$, $\tilde{a}$ is the sequence formed by these a_i, $\alpha_i = t_i$ for all $i \in I$ and $\alpha_i = a_i$ for all $i \in U$.

Intuitively, the parameter a denotes the name of the ν-MSR term representing the MSR term t we are considering. Therefore, we can consider each a_i as the name of the corresponding subterm of t. Now we define the translation function for MSR facts analogously.

Definition 15. Given an MSR formula F, we define the ν-MSR term $[\![F]\!]$ inductively as follows:

- If $F = p(t_1, \dots, t_n)$ is an MSR fact and $I = \{i \in 1, \dots, n \mid t_i \in Id \cup Var\}$, $U = \{1, \dots, n\} - I$, for each $i \in U$, $a_i \in Id \setminus fn(t_i)$ is a different name and $\tilde{a}$ is the sequence formed by these a_i, then

$$[\![F]\!] = \nu\tilde{a}.(p(\alpha_1, \dots, \alpha_n) + \sum_{i \in U} [\![t_i]\!]^{a_i})$$

  where $\alpha_i = t_i$ for all $i \in I$ and $\alpha_i = a_i$ for all $i \in U$.

- $[\![F_1 + F_2]\!] = [\![F_1]\!] + [\![F_2]\!]$,
- $[\![\nu a.F]\!] = \nu a.[\![F]\!]$.

In order to simplify the notation, in the following examples we usually consider the standard normal form of the translated terms and facts.

Example 4. Consider the fact $p(f(b, g(c)), g(b))$. The the corresponding ν-MSR term $[\![p(f(b, g(c)), g(b))]\!]$ is $\nu a_1 a_2 a_3 (p(a_1, a_2) + f(b, a_3, a_1) + g(c, a_3) + g(b, a_2))$.

Finally, we translate rules. Suppose that $p(t_1, \dots, t_n)$ is a MSR-term, and $[\![p(t_1, \dots, t_n)]\!] \equiv \nu\tilde{a}.M$, in standard form (so that M has no restriction). Then we will call $N(p(t_1, \dots, t_n))$ to M. We define the translation function for rules as follows:

Definition 16. Consider an MSR rule $R : F_1, \dots, F_m \to \exists a_1 \dots \exists a_k.G_1, \dots, G_n$. We define $[\![R]\!]$ as the ν-MSR rule

$$N(F_1) + \dots + N(F_m) \to \nu a_1 \dots a_k.([\![G_1]\!] + \dots + [\![G_n]\!])$$

Example 5. Let us consider the rule $q(g(x)) \to \exists y\, p(f(x, g(y)), g(x))$. The corresponding ν-MSR rule is $q(a_1) + g(x, a_1) \to \nu a_2 a_3 a_4 y(p(a_2, a_3) + f(x, a_4, a_2) + g(y, a_4) + g(x, a_3))$.

In this translation we may create garbage when a rule R, such that a variable x is in the terms of its preconditions and does not appear in the terms of the postconditions, is translated and applied by instantiating x to a term t which has some subterms. That is because, in this case, these subterms are not consumed but are not accessible from any fact. Therefore, they cannot be used again by a rule, and therefore remain as garbage. Let us illustrate this creation of garbage with an example:

Example 6. Consider the MSR rule $p(f(x, y), x) \to \exists z p(g(z), x)$. The corresponding ν-MSR rule would be $p(a_1, x) + f(x, y, a_1) \to \nu z, a_2(p(a_2, x) + g(z, a_2))$. Now, consider the MSR state $p(f(a, g(b)), a)$. Its translation is the term $\nu a_1, a_2(p(a_1, a) + f(a, a_2, a_1) + g(b, a_2))$, which can evolve using the previous rule to $\nu a_2, z, a_3(g(b, a_2) + p(a_3, a) + g(z, a_3))$. The term $g(b, a_2)$ remains unreachable from any other term.

Garbage does not affect depth-boundedness because when it is created, it can not grow anymore, and remains inaccessible. Anyhow, we can remove the garbage when created. In order to do that, we can add a new term $\mathit{dispose}(x)$ to the translated rules when x is as explained above and add new rules like $\mathit{dispose}(x) + f(y_1, \ldots, y_n, x) \to \mathit{dispose}(y_1) + \ldots + \mathit{dispose}(y_n)$.

As we defined depth-boundedness for ν-MSR, obtaining interesting results for depth-bounded ν-MSRs, we are interested in having an analogous concept in MSR, that is, having a set of conditions in MSR that ensure that the corresponding ν-MSR is depth-bounded. Given an MSR, the names of the states of its translation are either names in the MSR term or names representing subterms. Therefore, the set of restricted names causing the translation of an MSR to be depth-unbounded, is formed by an infinite amount of restrictions applied to names of one of these types (or both). That is why the translation of an MSR is depth-unbounded if the MSR satisfies at least one of the following conditions:

- We consider the height of a term t of an MSR to be 0 if $t \in Id$ and $1 + \mathit{max}\,\{\mathit{height}(t_i) \mid i = 1, \ldots, n\}$ if $t = f(t_1, \ldots, t_n)$. Then if the depth of the terms of the reachable states of an MSR is unbounded, its translation is depth-unbounded.
- Given a state S of an MSR, we define the hypergraph $\mathcal{G}(S)$ in a similar way as we defined the graph of a ν-MSR term: We consider the facts of S as vertices, and an arc from one node p to another node q if there is $a \in Id$ such that a belongs to both facts which represent p and q. Then, if there is not a bound for the length of the simple paths in any hypergraph built from the graphs of the reachable states of this MSR, then its translation is depth-unbounded.

Processes	
$P, Q ::=$	processes
$(\nu n)P$	restriction
0	inactivity
$P \mid Q$	composition
$!P$	replication
$n[P]$	ambient
$\pi.P$	capability action
Actions	
$\pi ::=$	capabilities
in n	can enter in n
out n	can exit n
open n	can open n

Figure 8: Syntax of Mobile Ambients

The second condition is the same as for ν-MSR. Let us illustrate the first one with an example:

Example 7. Consider an MSR with the rule $p(x) \to p(f(x))$. Then, if the initial state is $\{p(a)\}$, we could reach all states of the form $\{p(f(f(\ldots f(a)\ldots)))\}$. Therefore, the depth of the facts of the reachable states is unbounded, so the MSR is depth-unbounded. If we consider now the corresponding ν-MSR, the only rule is $p(x) \to \nu b(p(b) + f(x, b))$ and the initial state is $p(a)$. Therefore, we can reach every state of the form $p(b_1) + f(b_2, b_1) + \ldots + f(a, b_n)$, for all $n \in \mathbb{N}$, so the ν-MSR is depth-unbounded.

Then we can use our general result to depth-bounded MSR systems.

Proposition 12. *Coverability, termination and boundedness are decidable for depth-bounded MSR systems.*

7.2. ν-MSR and Mobile Ambients

Now we see how ν-MSR can be used to obtain decidable subclasses of Mobile Ambients (MA), a Turing-complete formalism for the specification of concurrent executing in a dynamical hierarchical topology [9]. Its syntax is defined in Fig. 8. The ambient operator is responsible for the creation of the tree topology of processes. The semantics is given by a structural congruence $\equiv$, defined to be the least congruence (for all operators) for which $\mid$ is commutative, associative, has 0 as identity and satisfies the equations in Fig. 9. Transitions can happen inside restrictions, ambients and parallel compositions, and are given by the axioms and rules in Fig. 9

Structural congruence

$(\nu n)(\nu m)P \equiv (\nu m)(\nu n)P$ (Res)

$(\nu n)(P \mid Q) \equiv P \mid (\nu n)Q$ if $n \notin fn(P)$ (Par)

$(\nu n)(m[P]) \equiv m[(\nu n)P]$ if $n \neq m$ (Amb)

$(\nu n)0 \equiv 0$ (ZeroRes)

$!0 \equiv 0$ (ZeroRepl)

Reduction

$n[in\ m.P \mid Q] \mid m[R] \rightarrow m[n[P \mid Q] \mid R]$ (In)

$m[n[out\ m.P \mid Q] \mid R] \rightarrow n[P \mid Q] \mid m[R]$ (Out)

$open\ n.P \mid n[Q] \rightarrow P \mid Q$ (Open)

$!P \rightarrow P \mid !P$ (Spawn)

$P' \equiv P, P \rightarrow Q, Q \equiv Q' \Rightarrow P' \rightarrow Q'$ ($\equiv$)

Figure 9: Operational Semantics of Mobile Ambients

Many papers investigate decidable subclasses of MA [7, 8, 5, 17, 18]. More precisely, these papers investigate subclasses of MA for which reachability is decidable. In [5] the authors drop name restriction and the open capability and consider the so called *weak semantics* of MA (the one we consider here), as opposed to the strong semantics in which we also have the rule $!P \mid P \rightarrow !P$. The paper [7] considers the strong semantics, provided every occurrence of replication is guarded by a prefix, and prove that reachability is still decidable. In [17] the authors study different dialects of MA within the common framework of the *Tree Update Calculus* (TUC), and in [18] extend their work to deal also with name restriction, proving that MA with name restriction, without the open capability and with the weak semantics still has decidable reachability.

Here we will extend these results by considering name restriction, the open capability and the strong semantics (actually, for coverability it is indifferent whether we use strong or weak semantics).

Let us now see how we encode MA inside ν-MSR. We will avoid using yet another intermediate formalism such as TUC, though we will use some ideas in [17, 18]. However, in those papers the translation from MA to TUC is done for a fragment of MA without restriction in the first case, or one in which only finitely-many names must be considered, in the second case.

For any MA process P we define the set of (sequential) processes $Der(P)$ by structural induction of P: $Der(0) = \{0\}$, $Der(n[P]) = Der(P)$, $Der(\pi.P) = \{\pi.P\} \cup Der(P)$, $Der(\nu n.P) = Der(P)$, $Der(P \mid Q) = Der(P) \cup Der(Q)$, and $Der(!P) = \{!P\} \cup Der(P)$. From the set of derivatives of a process we can build every process that can be reached from it. More precisely, if *Seq* is a set of (sequential) processes we define the set of processes $\mathcal{P}(Seq)$ as the least set containing all renamings of processes in Seq, closed under the ambient operator, parallel composition and name restriction, analogously as in Prop. 10 or as in [17]. Then it holds that if $P \rightarrow^* Q$ then $Q \in \mathcal{P}(Der(P))$.

Let us now assume that P_0 is the initial process. We use each $D \in Der(P_0)$ with n free variables as a predicate with arity $n+1$, and we will write $D(\tilde{x})^y$ instead of $D(\tilde{x}, y)$ to highlight the last parameter. Moreover, we use a 3-ary predicate $amb(x, y, z)$, meaning that y is an ambient with name x, inside ambient z. Then we can define the ν-MSR term $[\![P]\!]$ for any MA process P inductively as follows.

- $[\![0]\!]^\eta = 0$
- $[\![D(\tilde{\eta})]\!]^\eta = D(\tilde{\eta})^\eta$
- $[\![P_1 \mid P_2]\!]^\eta = [\![P_1]\!]^\eta, [\![P_2]\!]^\eta$
- $[\![\nu n.P]\!]^\eta = \nu n.[\![P]\!]^\eta$
- $[\![n[P]]\!]^\eta = \nu a(amb(n, a, \eta), [\![P]\!]^a)$

Now let us define ν-MSR rules that encode the transition relation of MA. We will use a 0-ary predicate *ok* and a binary predicate *moveup*.

$$\begin{array}{rl}
[Spawn]: & ok, !P(\tilde{x})^x \rightarrow ok, !P(\tilde{x})^x, [\![P(\tilde{x})]\!]^x \\
[In]: & ok, amb(x, y, z), (in\ t.P)^y, amb(t, u, z) \rightarrow ok, amb(x, y, u), [\![P]\!]^y \\
[Out]: & ok, amb(x, y, z), amb(t, u, y), (out\ x.P)^u \rightarrow ok, amb(t, u, z), [\![P]\!]^u \\
[Open]: & ok, (open\ x.P)^y, amb(x, z, y) \rightarrow moveup(z, y), [\![P]\!]^y
\end{array}$$

Moreover, we need to define rules to move up everything that was in the opened ambient to its parent ambient:

$$\begin{array}{rl}
[upSeq]: & moveup(x, y), D(\tilde{x})^x \rightarrow moveup(x, y), D(\tilde{x})^y \\
[upAmb]: & moveup(x, y), amb(z, t, x) \rightarrow moveup(x, y), amb(z, t, y) \\
[upDone]: & moveup(x, y) \rightarrow ok
\end{array}$$

This translation is *lossy*, in the sense that some processes may be lost.More precisely, when an ambient is opened, all the processes it contains must be "moved up" using rules [*upSeq*], [*upAmb*] and [*upDone*]. However, if rule [*upDone*] is fired before all processes are moved, then those processes will remain as garbage, in the sense that they cannot be reached from the rest of the processes (they remain as independent fragments). Again, this garbage does not affect the properties we are interested in. Indeed, if the process with garbage is non-terminating or unbounded, then so is the process without garbage.

Similarly to what we saw for MSR, with our encoding, depth-boundedness corresponds to MA in which the interdependence of names is bounded, and in which the height in the hierarchy of ambients is also bounded. However, the breadth of the hierarchical topology can be unbounded, as well as the amount of (non α-equivalent) processes in each ambient. Thus, our fragment of MA still encompasses restriction, unguarded replication and the open capability. To our knowledge, this is a novel subclass of MA for which some verification is still possible. More precisely, termination, boundedness and coverability are decidable for them and, as we have previously seen, we do not need to know in advance the bound on the interdependence of names or in the height of every reachable process.

Proposition 13. *Termination, boundedness and coverability are decidable for depth-bounded MA processes.*

In terms of coverability we can specify the *name convergence* problem in MA, proved to be undecidable in [5], even in the fragment of MA without name restriction and without the open capability. Given an ambient name n and a process P, the name convergence problem is that of deciding whether $P \rightarrow^* n[Q] \mid R$ for some processes Q and R. For depth-bounded processes (even with name restriction and with the open capability) the name convergence problem is decidable. Indeed, it is enough to decide whether $\nu a.amb(n, a, \top)$ can be covered from $[\![P]\!]^\top$.

Analogously, we can decide whether an ambient with a given name appears at all in some reachable marking, or whether a given spatial configuration can be covered.

8. Conclusions and future work

In this paper we have defined ν-MSRs, where MSR stands for MultiSet Rewriting. ν-MSRs encompass the multiset rewriting approach for concurrency, followed in [16], and the multiset rewriting approach for security, or name binding in general, followed in [11, 10].

We have proved that ν-MSRs simulate, in a very natural way, two models of concurrency with name binding, namely $p\nu$-PNs and π-calculus processes. The previous simulations establish that any result obtained for ν-MSRs can be translated both to the π-calculus and $p\nu$-PNs.

In particular, we have adapted the results in [31] in order to prove that a subclass of ν-MSR, that of depth-bounded ν-MSR, in which the interdependence of restricted names is bounded, is a strict Well Structured Transition System. This yields decidability of coverability, termination and also boundedness. Moreover, the decidability of those properties is obtained even when the bound on the depth of the terms is not known, even for coverability, and without need to use an Adequate Domain of Limits to finitely represent arbitrary downward closed sets.

These results can be transferred to any formalism that can be encoded within ν-MSR. We know that π-calculus processes can be easily translated to a ν-MSR system, so that depth-bounded π-calculus processes are WSTS. This was already proved in [31]. However, we can also obtain as a corollary the strict well structuredness of depth-bounded $p\nu$-PN. We claim that the same result holds for spi-calculus processes [1], with an encoding analogous to the one used for the π-calculus.

Then we have shown how other concurrent formalisms with name binding can be encoded within ν-MSR, namely MSR and Mobile Ambients (MA), and we have discussed what depth-boundedness means in each particular case. Thus, we obtain new decidability results for them. Up to our knowledge, this is the first time that decidability of verification for MSR is studied in general. Moreover, for the class of MA, we obtain new decidability results, as that of the name convergence problem.

As future work, it would certainly be interesting to find useful structural (or in any case decidable) sufficient conditions for depth-boundedness of ν-MSR.

In the encoding of MA, and more precisely in the encoding of the execution of an *open* capability, we need to modify an arbitrary amount of predicates, those representing processes inside the opened ambient. Since our rules only deal with a fixed amount of predicates, we need to process those predicates sequentially, perhaps leaving some garbage as explained there. Instead, we could add *broadcast* primitives to ν-MSR, similarly as the transfers or resets in Affine Well Nets [22] or as done for Data Nets [29]. For instance, we could have a rule stating that given names $a, b \in Id$ and predicates $p, q \in \mathcal{P}$, every $p(a, \tilde{a})$ is replaced by $q(b, \tilde{a})$. With such rules, we can for instance encode the open capability in such a way that its execution corresponds to a single application of a rule. Moreover, we claim that our decidability results still hold in that extension of ν-MSR.

In a different line, ν-MSRs establish a clean bridge between Petri nets and process algebra, that could be interesting in order to compare the natural concurrent (process) semantics of Petri nets to π-calculus processes.

Finally, recently new results for the verification of systems that rewrite graphs have been obtained in [19, 20]. We plan to relate our order, inducing the notion of depth-boundedness, to the order used in those papers to obtain well structuredness.

Acknowledgments

The authors would like to thank Pierre Ganty for his suggestion to use the generic abstract interpretation framework in [25] in the case of our depth-bounded terms.

References

[1] Abadi, M., Gordon, A.D.: A Calculus for Cryptographic Protocols: The spi Calculus. Inf. Comput. **148**(1) (1999) 1–70

[2] P.A. Abdulla, K. Cerans, B. Jonsson, Y. Tsay. Algorithmic Analysis of Programs with Well Quasi-ordered Domains. Inf. Comput. 160(1-2): 109-127 (2000)

[3] Abdulla, P.A., Delzanno, G., Begin, L.V.: Comparing the expressive power of well-structured transition systems. In Duparc, J., Henzinger, T.A., eds.: CSL. Volume 4646 of Lecture Notes in Computer Science., Springer (2007) 99–114

[4] Baldan, P., Bonchi, F., Gadducci, F.: Encoding asynchronous interactions using open Petri nets. In: CONCUR 2009: Proceedings of the 20th International Conference on Concurrency Theory, Berlin, Heidelberg, Springer-Verlag (2009) 99–114

[5] Boneva, I., Talbot, J.-M. When Ambients Cannot be Opened! TCS 333(1-2):127-169, 2005.

[6] Busi, N., Gorrieri, R.: Distributed semantics for the pi-calculus based on Petri nets with inhibitor arcs. J. Log. Algebr. Program. **78**(3) (2009) 138–162

[7] Busi, N., Zavattaro, G. Deciding reachability in mobile ambients. In ESOP'05:248-262.

[8] Busi, N., Zavattaro, G. Deciding reachability problems in turing-complete fragments of mobile ambients. Mathematical Structures in Computer Science **19**(6) (2009) 1223–1263

[9] Cardelli, L., Gordon, A.D.: Mobile ambients. Theor. Comput. Sci. **240**(1) (2000) 177–213

[10] Cervesato, I.: Typed MSR: Syntax and Examples. In Gorodetski, V.I., Skormin, V.A., Popyack, L.J., eds.: MMM-ACNS. Volume 2052 of Lecture Notes in Computer Science., Springer (2001) 159–177

[11] Cervesato, I., Durgin, N.A., Lincoln, P., Mitchell, J.C., Scedrov, A.: A meta-notation for protocol analysis. In: CSFW. (1999) 55–69

[12] Clavel, M., Durán, F., Eker, S., Lincoln, P., Martí-Oliet, N., Meseguer, J., Talcott, C.L., (Eds.): All About Maude - A High-Performance Logical Framework, How to Specify, Program and Verify Systems in Rewriting Logic. Lecture Notes in Computer Science 4350. Springer (2007)

[13] G. Decker, and M. Weske. *Instance Isolation Analysis for Service-Oriented Architecture.* In IEEE Int. Conference on Services Computing, SCC'08. IEEE Computer Society, 2008.

[14] Delzanno, G.: An overview of MSR(C): A CLP-based framework for the symbolic verification of parameterized concurrent systems. Electr. Notes Theor. Comput. Sci. **76** (2002)

[15] Delzanno, G.: A Symbolic Procedure for Control Reachability in the Asynchronous π-calculus. In 5th International Workshop on Verification of Infinite-State Systems, INFINITY 2003. Electr. Notes Theor. Comput. Sci. **98** 21-33 (2004)

[16] Delzanno, G.: Constraint multiset rewriting. Technical Report DISI-TR-05-08, University of Genova (2005)

[17] Delzanno, G., Montagna, R. Reachability Analysis of Mobile Ambients in Fragments of AC Term Rewriting. Formal Asp. Comput. 20(4-5): 407-428 (2008)

[18] Delzanno, G., Montagna, R. Deciding Reachability in Mobile Ambients with Name Restriction. Joint Proc. of the 8th, 9th, and 10th Int. Workshops on Verification of Infinite-State Systems, INFINITY 2006, 2007, 2008. ENTCS 239:5-15 (2009)

[19] Delzanno, G., Sangnier, A., Zavattaro, G.: Parameterized Verification of Ad Hoc Networks. In 21st International Conference on Concurrency Theory, CONCUR 2010. Lecture Notes in Computer Science 6269, pp. 313-327. Springer (2010)

[20] Delzanno, G., Sangnier, A., Zavattaro, G.: On the Power of Cliques in the Parameterized Verification of Ad Hoc Networks. In 14th International Conference on Foundations of Software Science and Computational Structures, FOSSACS 2011. Lecture Notes in Computer Science 6604, pp. 441-455. Springer (2011)

[21] Finkel, A., Schnoebelen, P.: Well-structured transition systems everywhere! Theor. Comput. Sci. **256**(1-2) (2001) 63–92

[22] A. Finkel, P. McKenzie, and C. Picaronny. *A well-structured framework for analysing petri net extensions.* Information and Computation, vol. 195(1-2):1-29 (2004).

[23] A. Finkel and J.Goubault-Larrecq. Forward analysis for WSTS, Part I: Completions. *In Proceedings of the 26th International Symposium on Theoretical Aspects of Computer Science, STACS'09* (2009) 433-444.

[24] A. Finkel and J.Goubault-Larrecq. Forward analysis for WSTS, Part II: Complete WSTS. *In 36th International Colloquium on Automata, Languages and Programming, ICALP'09.* LNCS vol. 5556. Springer (2009) 188-199.

[25] P. Ganty, J.-F. Raskin, and L. van Begin. A complete abstract interpretation framework for coverability properties of WSTS. In 7th VMCAI, pages 4964. Springer Verlag LNCS 3855 (2006).

[26] G. Geeraerts, J.-F. Raskin, and L. Van Begin. Expand, Enlarge and Check: New algorithms for the coverability problem of WSTS. J. Comput. Syst. Sci. 72(1): 180-203 (2006)

[27] Gordon, A.D.: Notes on nominal calculi for security and mobility. In Focardi, R., Gorrieri, R., eds.: FOSAD. Volume 2171 of Lecture Notes in Computer Science., Springer (2000) 262–330

[28] K.M. van Hee, M. Sidorova, M. Voorhoeve, and J.M. van der Werf. *Generation of Database Transactions with Petri Nets*. Fundamenta Informaticae 93(1-3):171-184 (2009)

[29] Lazic, R., Newcomb, T., Ouaknine, J., Roscoe, A.W., Worrell, J.: Nets with tokens which carry data. Fundam. Inform. **88**(3) (2008) 251–274

[30] Meseguer, J.: Rewriting logic as a semantic framework for concurrency: a progress report. In Montanari, U., Sassone, V., eds.: CONCUR. Volume 1119 of Lecture Notes in Computer Science., Springer (1996) 331–372

[31] Meyer, R.: On boundedness in Depth in the pi-calculus. In Ausiello, G., Karhumäki, J., Mauri, G., Ong, C.H.L., eds.: IFIP TCS. Volume 273 of IFIP., Springer (2008) 477–489

[32] Meyer, R.: A theory of structural stationarity in the *pi*-calculus. Acta Inf. **46**(2) (2009) 87–137

[33] Meyer, R., Gorrieri, R.: On the relationship between pi-calculus and finite place/transition Petri nets. In Bravetti, M., Zavattaro, G., eds.: CONCUR. Volume 5710 of Lecture Notes in Computer Science., Springer (2009) 463–480

[34] Minsky, M.L. Computation: finite and infinite machines. Prentice-Hall, Inc. (1967)

[35] Rackoff, C. The Covering and Boundedness Problems for Vector Addition Systems. Theor. Comput. Sci. 6: 223-231 (1978)

[36] Rosa-Velardo, F., de Frutos-Escrig, D.: Name creation vs. replication in Petri net systems. Fundam. Inform. **88**(3) (2008) 329–356

[37] Rosa-Velardo, F., de Frutos-Escrig, D.: Decidability problems for Petri nets with name creation and replication. Fundamenta Informaticae 104 (2010) 1–27

[38] Rosa-Velardo, F. Multiset Rewriting: a semantic framework for concurrency with name binding. In 8th International Workshop on Rewriting Logic and its Applications, WRLA 2010. LNCS vol. 6381, pp. 191-207. Springer-Verlag, 2010.

[39] Rosa-Velardo, F. Depth boundedness in multiset rewriting systems with name binding. In 4th Workshop on Reachability Problems, RP 2010. LNCS 6227, pp. 161-175. Springer-Verlag, 2010.

[40] Rosa-Velardo, F., de Frutos-Escrig, D.: Decidability and Complexity of Petri Nets with Unordered Data. Theoretical Computer Science (to appear)

[41] Sangiorgi, D., Walker, D.: The *pi*-calculus: a Theory of Mobile Processes. Cambridge University Press (2001)

[42] Stehr, M.O., Meseguer, J., Ölveczky, P.C.: Rewriting logic as a unifying framework for Petri nets. In Ehrig, H., Juhás, G., Padberg, J., Rozenberg, G., eds.: Unifying Petri Nets. Volume 2128 of Lecture Notes in Computer Science., Springer (2001) 250–303

[43] T. Wies, D. Zufferey, and T. Henzinger. Forward Analysis of Depth-Bounded Processes. 13th Int. Conference Foundations of Software Science and Computational Structures, FOSSACS 2010. LNCS vol. 6014, pp. 94-108. Springer-Verlag, 2010.