Available online at www.sciencedirect.com
ScienceDirect
Electronic Notes in Theoretical Computer Science
ELSEVIER
Electronic Notes in Theoretical Computer Science 239 (2009) 5–15
www.elsevier.com/locate/entcs

Deciding Reachability in Mobile Ambients with Name Restriction

Giorgio Delzanno and Roberto Montagna

Dipartimento di Informatica e Scienze dell'Informazione
Università di Genova, via Dodecaneso 35, 16146 Genova, Italy
e-mail: {giorgio, montagna}@disi.unige.it

Abstract

We investigate the reachability problem for fragments of the Mobile Ambients, a powerful model for distributed and mobile computation. By using a connection with associative-commutative term rewriting, we prove that reachability is decidable in the open-free fragment of pure Mobile Ambients with name restriction and weak reduction semantics. Processes in this model have three sources of infiniteness: depth of ambients, width of parallel composition, and number of restricted names. Our work extends similar results obtained for public fragments of Mobile Ambients.

Keywords: Mobile Ambients, reachability, term rewriting

1 Introduction

The Mobile Ambients (MA) of Cardelli and Gordon [4] is a powerful model of distributed, mobile computation. The basic block of this model is the notion of ambient. An ambient is represented by the expression $n[P]$ where n is a name, and P is a collection of local agents and sub-ambients. Local agents model the possible computations that can take place inside the ambient. The formalism is based on classical operations of process algebra like action prefix $act.P$, parallel composition $P \mid Q$ and replication $!P$. The replication $!P$ denotes an arbitrary number of copies of P in parallel. Its semantics is defined via the axiom $!P \equiv !P|P$. In addition to these operations, the *pure* (i.e. without communication) version of the calculus provides movement capabilities like $in\ n$ ($out\ n$) that allow an ambient A (with any label) to enter (exit from) ambient B with label n. As an example, the process $m[in\ n.P|Q] \mid n[R]$ reduces in one step to $n[m[P|Q]|R]$, whereas $n[m[out\ n.P|Q] \mid R]$ reduces in one step to $n[R] \mid m[P|Q]$. The *open* capability can be used to dissolve the boundary of an ambient. The name restriction $\nu x.P$ (where x is a name that may occur free in P) can be used to assign unique identities to dynamically generated ambients. As an example, the replicated process $P = !\nu x.x[in\ n.\mathbf{0}]$ is equivalent

doi:10.1016/j.entcs.2009.05.027

to the process $\nu x_1 \ldots . \nu x_n . x_1[in\ n.\mathbf{0}] \mid \ldots \mid x_n[in\ n.\mathbf{0}] \mid P$, i.e., to a collection of an arbitrary number of ambients with fresh names and capability *in* n ($\mathbf{0}$ represents the null process). The movement capabilities may generate ambients with arbitrary nesting structure. Thus, MA is an *infinite-state model* with several sources of infiniteness: width of parallel composition, number of restricted names, and depth of ambients.

Reachability in MA

Expressiveness issues and verification problems for dialects of MA have been studied in [2,10,1,3,5,7,8]. In this paper we focus our attention on the *reachability problem*: is there a computation from process P to process Q?. This problem has been studied for public (i.e. without name restriction) fragments of MA in [6,1,3]. Specifically, in [6] Charatonik and Talbot proved the undecidability of reachability in pure public MA. In [1] Boneva and Talbot refined this result by showing that reachability is undecidable in the *open-free* (i.e. without open capability) fragment of public MA. They also proved the Turing completeness of the same fragment by exhibiting an encoding of two counter machines. In this encoding the standard semantics of replication (i.e. $!P \equiv !P|P$) is used for collecting garbage left by the processes performing the simulation of the counter machine. Indeed, in the same paper the authors shown that reachability becomes decidable whenever the replication operation is only used to generate new processes (i.e. $!P \equiv !P|P$ is turned into the oriented reduction rule $!P \rightarrow !P|P$). This semantic restriction is called *weak reduction*. In [3] Busi and Zavattaro proved that reachability is decidable in open-free public MA with standard reduction semantics whenever every occurrence of replication is guarded by a movement capability. (i.e. they admit the use ! only in processes like $!M.P$ where M is either *in* n or *out* n). Interestingly, the open-free public MA fragment with guarded replication is still Turing complete [3,10].

Novel Contribution

In this paper we extend the decidability result of Boneva and Talbot in [1] by proving that reachability remains decidable when adding *name restriction* to the open-free fragment of MA with weak reduction, i.e., when adding the third source of infiniteness to the fragment considered in [1]. We call the resulting fragment pMA_w^{-o}. To prove this result, we exploit a link between MA and AC term rewriting. Specifically, we show that the reachability problem in pMA_w^{-o} can be reduced to a reachability problem for ground terms and rewrite rules with multiset-variables. The resulting rewrite rules satisfy the syntactic restrictions proposed in [9] (*structure preserving rewriting*) under which reachability is decidable.

This reduction requires some preliminary transformation on the pMA_w^{-o} semantics. More in detail, we first introduce a new reduction relation working on pMA_w^{-o} processes in a special syntactic form called *prenex form*. A process is in prenex form when all the name restrictions occurring outside the scope of a replication are moved at the top level (i.e. it has the form $\nu\vec{x}.P$ where P has no restrictions

outside replications). This transformation requires some work because the congruence relation $\equiv$ of MA does not provide a direct way to move restrictions through movement capabilities, e.g., $in\ n.\nu\ x.P \equiv \nu\ x.in\ n.P$ for $x \neq n$ is not an axiom for $\equiv$. As a second step, we show how to reduce a reachability problem in the new pMA_w^{-o} reduction semantics to a reachability problem for ground terms. Terms are built by mapping an ambient $n[P_1 \mid \ldots \mid P_n]$ to a compound term $n\langle t_1 \mid \ldots \mid t_n\rangle$, where t_i is a term associated to P_i, and $\mid$ is an associative-commutative term constructor. Local agents like $!P$ are encoded by constants like $q_{!P}$. The key point in the encoding consists in showing that it is enough to consider a finite set of node constructors and constants to model a reachability problem in pMA_w^{-o}. Since we work in AC term rewriting, the finiteness of the set of constants does not imply the finiteness of the set of ground terms we have to deal with. As an example, if $q_{\mathbf{0}}$ is the constant representing the null process $\mathbf{0}$, then we have to deal with the infinite set of ground terms of the form $q_{\mathbf{0}} \mid \ldots \mid q_{\mathbf{0}}$. This makes the encoding from pMA_w^{-o} to term rewriting non trivial.

As mentioned before, our result extends the result of Boneva and Talbot in [1] formulated for the *public* fragment of pMA_w^{-o}. To our knowledge, this is the first positive result for reachability in non trivial fragments of MA with name restriction.

Related work

In [9] we have studied the relationship between public fragments of MA and a fragment of associative and commutative (AC) term rewriting, we called TUC. Indeed, the computational mechanisms of public MA can be naturally expressed using rewriting systems working on terms with multiset-variables. In [9] we have shown that reachability between ground terms (but for a set of rules with multiset-variables) is decidable for the *structure preserving* fragment of TUC, called TUC^{SP}. Structure preserving rules cannot remove *internal nodes* of a tree term. However, they can still produce and consume leaves. In the same paper we have shown that the decidability of reachability in TUC^{SP} generalizes the results obtained for the fragments of Mobile Ambients in [1,3]. Indeed, the semantic and syntactic restrictions for MA studied in [1,3] can be reformulated in a uniform way using a set of structure preserving TUC rewrite rules. Interestingly, TUC^{SP} has a different nature from other fragments of AC rewriting like PRS [12] and AC ground rewriting [11]. Indeed, to express the movement operations of MA, we need rewrite rules (like the one in the previous example) that synchronize tree terms with multiset-variables. This kind of synchronization rules are restricted to ground terms in PRS and ground AC term rewriting (the interested reader can refer to [9] for a more detailed discussion on this point).

1. $P|\mathbf{0} \equiv P$
2. $P|Q \equiv Q|P$
3. $(P|Q)|R \equiv P|(Q|R)$
4. $if\ P \equiv_\alpha Q\ then\ P \equiv Q$
5. $\nu n.(\nu m.P) \equiv \nu m.(\nu n.P)$
6. $\nu x.\mathbf{0} \equiv \mathbf{0}$
7. $\nu x.(P|Q) \equiv P|\nu x.Q \quad for \quad x \notin fn(P)$
8. $\nu x.(n[P]) \equiv n[\nu x.P] \quad for \quad x \neq n$

Fig. 1. Congruence relation.

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

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

$$open\ n.P|n[Q] \ \rightarrow\ P|Q \qquad (open)$$

$$P \mid !P \ \rightarrow\ !P \qquad (abs)$$

$$!P \ \rightarrow\ P \mid !P \qquad (gen)$$

$$\frac{P \rightarrow Q}{C[P] \rightarrow C[Q]}\ (context) \qquad \frac{P' \equiv P \quad P \rightarrow Q \quad Q \equiv Q'}{P' \rightarrow Q'}\ (congr)$$

Fig. 2. Reduction relation: $C[\bullet]$ is either $\bullet|R$, $n[\bullet]$, or $\nu x.\bullet$

2 Pure Mobile Ambients (MA)

Given a denumerable set of ambient names Amb, the set of MA process terms is the smallest set generated by the following grammar.

$$P, Q \quad ::= \quad \mathbf{0} \mid \ P|Q \ \mid !P \mid \ \nu x.P \ \mid n[P] \mid \ in\ n.P \ \mid \ out\ n.P \ \mid open\ n.P$$

The term $n[P]$ denotes an *ambient* with name n. *Local agents* are processes in one of the following forms: $\mathbf{0}$, $!P$, $in\ n.P$, $out\ n.P$, and $open\ n.P$. In the rest of the paper we use $P \equiv_\alpha Q$ to denote that P and Q are equivalent modulo α-conversion, and $fn(P)$ to denote the set of free names in P (all names occurring in P that are not binded by a name restriction). The structural congruence $\equiv$ is the smallest congruence relation satisfying the equations listed in Fig. 1. Notice that for any P and $x \notin fn(P)$, we have that $\nu x.P \equiv P$. We call this kind of restrictions *useless*. Furthermore, we call *active* any occurrence of a term/operator outside the scope of a replication. The operational semantics of the language is given via a reduction relation $\rightarrow$ defined as the smallest relation satisfying the axioms and rules of Fig. 2. Differently from the standard presentation of Mobile Ambients, the equivalence $!P \equiv !P \mid P$ is split into two reduction axioms, namely *gen* (generate) and *abs* (absorb). This presentation simplifies the definition of the fragments studied in the following section. We use $\rightarrow^*$ to denote the reflexive and transitive closure of $\rightarrow$.

Definition 2.1 Given processes P and Q, the *reachability problem* $RP(P, Q)$ consists in deciding if $P \rightarrow^* Q$.

2.1 Reachability in open-free Mobile Ambients

As mentioned in the introduction, in [1] Boneva and Talbot proposed a weak reduction semantics for a fragment without *open* and without name restriction. According to the weak reduction of [1], replication in this fragment can only be used to generate new processes. The result in [1] is based on the following property: if $P \rightarrow^* Q$ in this fragment, then the tree structure of Q gives us an upper bound on the number of ambients (occurring outside a replication) that may occur in the processes appearing in a derivation from P to Q. Indeed, without *open* and with weak reduction it is not possible to consume ambients in a derivation. In this paper we study the reachability problem for an extension, named pMA_w^{-o}, of the fragment with weak reduction of Boneva-Talbot defined as follows.

Definition 2.2 The fragment pMA_w^{-o} is obtained by forbidding the use of the *open* capability in the definition of a process (open-free), and by removing *abs* and *open* from the rules of Fig. 2 (weak reduction).

To extend the result of [1] to pMA_w^{-o}, we need some considerations on the semantics of restrictions and movement. Consider a process $\nu n.P'$ occurring in a derivation from P to Q and suppose that we use α conversion to avoid clashing with other restrictions occurring in P. Then, we have three possible situations. (1) If n does not occur in P' then $\nu n.P'$ is equivalent to P'. (2) If n occurs in a subterm $n[Q]$ of P', then n occur in all successive configurations (weak reduction does not allow the consumption of active ambients). (3) Finally, the more subtle case is when n occurs in a subterm $in/out\ n.Q$ of P' while it does not occur in a subterm $n[Q']$ of P'. Potentially, we could consume the name n by executing the corresponding capability. However, $in\ n$ and $out\ n$ require the presence of an ambient named n to be executed. Thus, in the latter case the processes $in\ n.Q$ and $out\ n.Q$ never be executed (they are deadlocked). The previous properties show us that if $P \rightarrow^* Q$ in pMA_w^{-o} then Q contains at least one occurrence (either in a term $n[Q]$ or in a deadlocked process $in\ n.P$ or $out\ n.P$) of every newly generated ambient name. Thus, the tree structure of Q together with the set of ambient names occurring in Q outside the scope of a replication can be used to have an upper bound on the size of tree structures and on the number of names we have to consider to solve $RP(P, Q)$. To make this observation into a formal argument, we introduce a special class of terms, we call prenex forms, in which all restrictions are moved at the top level (i.e. we extrude their scope as much as possible), and we only keep restrictions that bind names occurring somewhere in the process.

2.2 Prenex form

The prenex-form of a term P is a term P' (structurally equivalent to P) in which all the name restrictions have been pushed up (renamed if necessary) in the tree-structure of P as much as possible. To find the prenex form we use a rewriting relation $\rightharpoonup$ defined as the smallest relation satisfying the rules in Fig. 3. The reduction in prenex form is defined as follows.

$$\nu x.P \rightharpoonup P \qquad if\ x \notin fn(P)$$

$$(\nu x.P) \mid Q \rightharpoonup \nu x.(P \mid Q) \qquad if\ x \in fn(P)\ and\ x \notin fn(P)$$

$$n[\nu x.P] \rightharpoonup \nu x.n[P] \qquad if\ x \in fn(P)\ and\ x \neq n$$

$$in\ n.(\nu x.P) \rightharpoonup \nu x.(in\ n.P) \qquad if\ x \in fn(P)\ and\ x \neq n$$

$$out\ n.(\nu x.P) \rightharpoonup \nu x.(out\ n.P) \qquad if\ x \in fn(P)\ and\ x \neq n$$

$$\frac{P \rightharpoonup Q}{C[P] \rightharpoonup C[Q]}\ (context) \qquad \frac{P \equiv_{1-5} P' \rightharpoonup Q' \equiv_{1-5} Q}{P \rightharpoonup Q}\ (congr)$$

Fig. 3. Relation $\rightharpoonup$: $C[\bullet]$ is either $\bullet|R$, $n[\bullet]$, $M.\bullet$, or $\nu x.\bullet$.

$$m[in\ n.P \mid Q] \mid n[R] \rightarrow_\emptyset n[m[P \mid Q] \mid R]$$

$$n[m[out\ n.P \mid Q] \mid R] \rightarrow_\emptyset m[P \mid Q] \mid n[R]$$

$$P \Rrightarrow \nu\vec{y} \triangleright P' \qquad P \rightarrow_{\vec{y}} Q \qquad if\ \vec{y} \cap fn(R) = \emptyset$$
$$!P \rightarrow_{\vec{y}} !P|P' \qquad P \mid R \rightarrow_{\vec{y}} P \mid R$$

$$\frac{P \rightarrow_{\vec{y}} Q}{n[P] \rightarrow_{\vec{y}} n[Q]}\ if\ n \notin \vec{y} \qquad P \equiv_{1-5} P' \rightarrow_{\vec{y}} Q' \equiv_{1-5} Q \Rightarrow P \rightarrow_{\vec{y}} Q$$

$$P \rightarrow_{\vec{y}} Q \Rightarrow \nu\vec{x} \triangleright P \mapsto \nu\vec{x} \cup \vec{y} \triangleright Q \qquad if\ \vec{y} \cap \vec{x} = \emptyset$$

Fig. 4. Restricted Reduction Relation.

Definition 2.3 $P \Rrightarrow P'$ if $P \rightharpoonup^* P' \not\rightharpoonup$.

The prenex form P' of P is such that $P \Rrightarrow P'$. In a process in prenex form all active restrictions are moved at the top level. In order to study the properties of processes in prenex form, let us call *tree structure* of a process P the term $ts(P)$ obtained by removing all active occurrences of restriction in P. Formally, $ts(!Q) = !Q$, $ts(\mathbf{0}) = \mathbf{0}$, $ts(\nu x.Q) = ts(Q)$, $ts(n[Q]) = n[ts(Q)]$, $ts(Q|R) = ts(Q)|ts(R)$, $ts(M.Q) = M.ts(Q)$. The following properties hold.

Proposition 2.4 *The relation* $\Rrightarrow$ *modulo* $\equiv_{1-5}$ *is terminating.*

Proposition 2.5 *If* $P_1 \Rrightarrow P_2$*, then there exist a set of names* $\vec{y}$ *such that* $P_2 = \nu\vec{y}.P_3$*,* P_3 *has no active occurrences of restrictions, every name in* $\vec{y}$ *occurs free in* P_3*, and there exists* $P_1' \equiv_\alpha P_1$ *such that* $ts(P_1') = ts(P_2)$*.*

Proposition 2.6 *If* $P_1 \Rrightarrow P_2$ *and* $P_1 \Rrightarrow P_3$*, then* $P_2 \equiv_{1-5} P_3$*.*

In the following we use the notation $\nu\vec{y} \triangleright P$ to identify a term $\nu\vec{y}.P$ in which P does not contain active occurrences of restrictions, e.g., to isolate the block of top-level restrictions of a prenex form. In Fig. 4 we define a new reduction relation

$$Der_N(\mathbf{0}) = \{\mathbf{0}\} \qquad Der_N(n[P]) = Der_N(P)$$
$$Der_N(M.P) = \{M.P\} \cup Der_N(P) \qquad Der_N(\nu x.P) = \bigcup_{n \in N} Der_N(P[n/x])$$
$$Der_N(P \mid Q) = Der_N(P) \cup Der_N(Q)$$
$$Der_N(!P) = \{!P\} \cup Der_N(P') \; if \; P \Rrightarrow P'$$

Fig. 5. Derivatives of a process.

working on terms in prenex form and in which α-renaming is only applied locally to the generation rule ($\emptyset$ denotes the empty vector). The following property relates the new reduction with the standard one.

Proposition 2.7 *For any P, Q, $P \to Q$ iff $P \Rrightarrow P' \longmapsto Q'$ and $Q \Rrightarrow Q'$.*

Let us make some final considerations on the semantics of pMA_w^{-o}. Let us first notice that we can work with a congruence relation applied only to contexts different from $!P$ (as for the reduction semantics). Furthermore, let us reformulate the axiom $P \mid \mathbf{0} \equiv P$ as the following two reduction rules

$$P \to_\emptyset P \mid \mathbf{0} \qquad P \mid \mathbf{0} \to_\emptyset P$$

Since in MA the empty ambient is $n[\mathbf{0}]$ and not $n[\,]$, when using the reduction rules for $\mathbf{0}$ we need to refine the reduction rule for the *out* rule as follows

$$n[m[out\ n.P \mid R] \mid Q] \;\to_\emptyset\; m[P \mid R] \mid n[Q \mid \mathbf{0}]$$

Several computation steps in the resulting reduction may correspond to one computation or congruence step in the original semantics. However, the reachability can be safely checked in the new semantics. From here on, we still use $\mapsto$ to denote the the modified reduction relation. Finally, given a process term P and a finite set of names N, in Fig. 5 we define the set of local agents $Der_N(P)$ that may become active during a computation (*derivatives*) and in which restricted names are replaced by names in N. For a finite set N, $Der_N(P)$ is finite, too. Furthermore, we have the following property.

Proposition 2.8 *If $P_0 \mapsto P_1 \ldots \mapsto P_n = \vec{x}_n \triangleright Q_n$, then $Der_{\vec{x}_n}(P_0) \cup \{\mathbf{0}\}$ contains the set of local agents active in P_i for $i : 0, \ldots, n$.*

3 From Mobile Ambients to AC term rewriting

To show that the reachability problem for the fragment of pure Mobile Ambients defined in the previous section is decidable, we use a reduction to reachability in a special fragment of AC term rewriting called *structure preserving*. The latter problem is decidable [9]. We introduce the syntax of structure preserving rewrite rules in the next section.

4 Structure Preserving AC Term Rewriting

We consider a restricted class of rewrite rules defined over TR terms and with variables ranging over multisets of terms. For this purpose we first need to define the shape of restricted terms that can occur in the left- and right-hand side of rules RT_L and RT_R, respectively. Given a denumerable set of variables $\mathcal{V} = \{X, Y, \ldots\}$: ranging over MS-terms:
RT_L is the least set of terms satisfying: $\mathcal{Q} \subseteq RT_L$; if $t_1, \ldots, t_n \in RT_L$, and $X \in \mathcal{V}$, then $n\langle t_1 \mid \ldots \mid t_n \mid X\rangle \in RT_L$ for $n \geq 0$.
Furthermore, RT_R is the least set of terms satisfying: $\mathcal{Q} \subseteq RT_R$; if $t_1, \ldots, t_n \in RT_R$, and $X \in \mathcal{V}$, then $n\langle t_1 \mid \ldots \mid t_n \mid X\rangle \in RT_R$ and $n\langle t_1 \mid \ldots \mid t_n\rangle \in RT_R$.

Given a term t let $IntNds(t)$ denote the number of occurrences of labels in $\mathcal{N}$ (internal nodes/ambients) in t. Formally, $IntNds(t)$ is defined by induction on t as follows: $IntNds(\epsilon) = IntNds(X) = IntNds(q) = 0$ for $X \in \mathcal{V}$ and $q \in \mathcal{Q}$, $IntNds(t_1|\ldots|t_k) = IntNds(t_1|\ldots|t_k|\, X) = \Sigma_{i=1}^{k} IntNds(t_i)$, and $IntNds(n\langle m\rangle) = IntNds(m) + 1$.

A structure preserving rule $l \to r$ is such that

(i) $l = t_1 \mid \ldots \mid t_n$, and $t_i \in RT_L$ for $i : 1, \ldots, n$,

(ii) $r = t'_1 \mid \ldots \mid t'_m$ and $t'_i \in RT_R$ for $i : 1, \ldots, m$;

(iii) l and r have the same set V of variables;

(iv) each variable in V occurs once in l and once in r;

(v) $IntNds(l) \leq IntNds(r)$.

5 Encoding Reachability Problems

Consider the processes $P_0 = \vec{x} \triangleright P'_0$ with $\vec{x} = \langle x_1, \ldots, x_k\rangle$ and $P_1 = \vec{x} \cup \vec{y} \triangleright P'_1$ with $\vec{y} = \langle y_1, \ldots, y_p\rangle$. Furthermore, assume that all free names occurring in P'_0 occur in $\vec{x}$. The reachability problem $RP(P_0, P_1)$ can be encoded into a reachability problem for two ground terms and a finite set $\mathcal{R}$ of term rewrite rules with variables and one associative-commutative constructor. To handle names, we consider a set N_1 of constants associated to the names in $\vec{x}$ and a set N_2, disjoint from N_1, associated to the names in $\vec{y} \setminus \vec{x}$. We define $\mathcal{N} = N_1 \cup N_2$. Let us now describe the encoding of processes into terms.

The set TR of terms used to represent processes is built upon a signature with the following constructors and constant symbols:

- For any $n \in \mathcal{N}$, the ambient $n[\cdot]$ is represented by the constructor $n\langle\cdot\rangle$.
- The parallel composition is represented by an associative and commutative constructor $|$. The constant ϵ is the identity element of $|$. A term $t_1|\ldots|t_n$ can be viewed as a multiset of terms.
- Each derivative R of P_0 is represented by means of a constant q_R.
- To keep track of unused names, we associate a constant q_n to each $n \in N_2$.

$$T(\mathbf{0}) = q_{\mathbf{0}} \quad T(!Q_1) = q_{!Q_1} \quad T(M.Q_1) = q_{M.Q_1}$$
$$T(n[Q_1]) = n\langle T(Q_1)\rangle \qquad T(Q_1|Q_2) = T(Q_1)|T(Q_2)$$

Fig. 6. Encoding of processes into ground terms.

A process P derived from P_0 is mapped to a ground term $T(P)$ (i.e. a term with variables) in TR via the map T defined by induction as shown in Fig. 6. Notice that the map T does not produce constants q_n with $n \in N_2$. We add them to the initial configuration as explained in the next section.

We are ready now to define the set $\mathcal{R}$ of rules modelling the behavior of processes in $Der_{\mathcal{N}}(P_0)$. In the following $X, Y, \ldots$ denote variables ranging over multisets of terms.

- For every $n, m \in \mathcal{N}$, $in\ n.Q, out\ n.Q \in Der_{\mathcal{N}}(P_0)$, $\mathcal{R}$ contains:

$$m\langle q_{in\ n.Q} \mid X\rangle \mid n\langle Y\rangle \longrightarrow n\langle m\langle T(Q) \mid X\rangle \mid Y\rangle$$
$$n\langle m\langle q_{out\ n.Q} \mid X\rangle \mid Y\rangle \longrightarrow m\langle T(Q) \mid X\rangle \mid n\langle q_{\mathbf{0}}|Y\rangle$$

  These rules are a natural reformulation of the movement operations of pMA^{-o}_{w}. The continuation Q is a label in the constant $q_{M.Q}$ occurring in the left-hand side. It becomes a ground term $T(Q)$ in the right-hand side.

- In order to generate a new copy of process $!Q$, we first put it in its prenex form $Q'' = \nu\vec{y} \triangleright Q_1$. Then, we assume that v distinct leaves $q_{a_1}, \ldots, q_{a_v}$ representing unused names float in parallel with $q_{!Q}$. The rule consumes these leaves (i.e. every name in N_2 can be used only once) and generate an instance of Q_1 in which the free names $y_1, \ldots, y_v$ are replaced by $a_1, \ldots, a_v$. Formally, for every $q_{!Q} \in \mathcal{Q}$ and $a_1, \ldots, a_v \in N_2$, $\mathcal{R}$ contains:

$$q_{a_1} \mid \ldots \mid q_{a_v} \mid q_{!Q} \longrightarrow q_{!Q} \mid T(R)$$

  where $Q \Rrightarrow \nu\vec{y} \triangleright Q_1$, $\vec{y} = y_1, \ldots, y_v$, $a_i \notin fn(Q_1)$ for $i : 1, \ldots, v$, and $R = Q_1[a_1/y_1, \ldots, a_v/y_v]$.

- For the previous rule to work, constants that represent unused names must be available inside any ambient when needed. To let constants q_a with $a \in N_2$ move across ambients, for any $m \in \mathcal{N} \setminus \{a\}$, we add to $\mathcal{R}$ the rule

$$q_a \mid m\langle X\rangle \longrightarrow m\langle q_a|X\rangle$$

  If constants associated to N_2 are at the top level in the initial configuration, then the *move* rule allows us to distribute them inside the tree structure of terms in order to be ready to synchronize with a *gen* rule.

- Finally, for any $R \in Der_{\mathcal{N}}(P_0)$ and $n \in \mathcal{N}$ we add to $\mathcal{R}$ the rules

$$q_R \to q_R \mid q_{\mathbf{0}} \quad n\langle X\rangle \to \langle X\rangle \mid q_{\mathbf{0}} \quad q_R \mid q_{\mathbf{0}} \to q_R \quad n\langle X\rangle \mid q_{\mathbf{0}} \to n\langle X\rangle$$

These rules naturally model the congruence $P|\mathbf{0} \equiv P$ independently from the structure of P (ambient $n\langle\ldots\rangle$ or local agent q_R).

Since $Der_{\mathcal{N}}(P_0)$ is a finite set and $\Rightarrow$ is terminating, then we can always pre-compute all the terms needed to define $\mathcal{R}$. Thus, for fixed P_0 and $\mathcal{N}$, $\mathcal{R}$ is a finite set of rewrite rules.

Now let $\Rightarrow$ denote the standard rewrite relation for ground terms, i.e, $t \Rightarrow t'$ if there exists a ground instance $l \rightarrow r$ of a rule in $\mathcal{R}$ such that l is a subterm of t and t' is obtained by replacing l with r in t. Let $\Rightarrow^*$ be the reflexive and transitive closure of $\Rightarrow$. Then, the following property holds.

Proposition 5.1 *Given $RP(P_0, P_1)$ with $P_0 = \nu\vec{x} \triangleright P_0'$ and $P_1 = \nu\vec{x} \cup \vec{y} \triangleright P_1'$, let $N_1 = \{n_1, \ldots, n_k\}$, $N_2 = \{a_1, \ldots, a_p\}$, $\mathcal{N} = N_1 \cup N_2$ and let $\mathcal{R}$ be the set of rewrite rules associated to $Der_{\mathcal{N}}(P_0)$. Then, we have that*

$$\vec{x} \triangleright P_0' \mapsto^* \vec{x} \cup \vec{y} \triangleright P_1' \quad \textit{iff} \quad q_{a_1} \mid \ldots \mid q_{a_p} \mid T(Q_0) \Rightarrow^* T(Q_1)$$

where $Q_0 = P_0'[n_1/x_1, \ldots, n_k/x_k]$, $Q_1 = P_1'[n_1/x_1, \ldots, n_k/x_k, a_1/y_1, \ldots, a_p/y_p]$.

From Prop. 5.1 it follows that reachability in pMA_w^{-o} can be reduced to a reachability problem for two ground terms and a finite set of AC term rewrite rules with variables ranging over multisets of terms. All rewrite rules used in this reduction satisfy the *structure preserving* syntactic restriction introduced in [9]. The restriction ensures that the application of a structure preserving rewrite rule never removes internal nodes (occurrences of the constructor $n\langle\rangle$) from the current term. For rules of this kind, in [9] we have proved that reachability can be decided by means of a further encoding into Petri net reachability. The following result then holds.

Theorem 5.2 *Reachability is decidable in pMA_w^{-o}.*

Proof. It follows then from the decidability of reachability for ground terms and structure preserving AC term rewriting proved in [9]. □

6 Conclusions

The open-free fragment of MA with weak reduction is a model with different sources of infiniteness: the number of local agents/ambients, the nesting of ambients, and the number of names. In this paper we have proved that reachability is decidable in this infinite-state model. This result extends the decidability result of [1] obtained for pMA_w^{-o} without name restriction.

References

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

[2] N. Busi and G. Zavattaro. On the expressive power of movement and restriction in pure mobile ambients. TCS 322(3): 477-515, 2004.

[3] N. Busi and G. Zavattaro. Deciding Reachability in Mobile Ambients. In ESOP '05: 248-262.

[4] L. Cardelli and A. D. Gordon. Mobile ambients. TCS 240(1): 177-213, 2000.

[5] L. Cardelli and A. D. Gordon. Anytime, anywhere: modal logics for mobile ambients. In POPL '00: 365–377.

[6] W. Charatonik, J.-M. Talbot. The Decidability of Model Checking Mobile Ambients. In CSL 2001: 339-354.

[7] W. Charatonik, S. Dal Zilio, A. D. Gordon, S. Mukhopadhyay, and J.M. Talbot. The Complexity of Model Checking Mobile Ambients. In FoSSaCS 2001: 152-167.

[8] W. Charatonik, S. Dal Zilio, A. D. Gordon, S. Mukhopadhyay, J-M. Talbot. Model Checking Mobile Ambients. TCS 308(1-3): 277-331, 2003.

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

[10] S. Maffeis and I. Phillips. On the computational strength of pure ambient calculi. TCS 330(3): 501-551 (2005)

[11] R. Mayr and M. Rusinowitch. Reachability is decidable for ground AC rewrite systems. In INFINITY'98, 3rd International Workshop on Verification of Infinite State Systems, Aalborg (Denmark), July 1998.

[12] R. Mayr. Process Rewrite Systems. Inf. Comput. 156(1-2): 264-286, 2000.