On the Reachability Problem in P Systems with Mobile Membranes

Bogdan Aman[1] and Gabriel Ciobanu[1,2]

[1] Romanian Academy, Institute of Computer Science
Blvd. Carol I no.8, 700505 Iaşi, Romania
[2] "A.I.Cuza" University of Iaşi, Faculty of Computer Science
Blvd. Carol I no.11, 700506 Iaşi, Romania
baman@iit.tuiasi.ro, gabriel@info.uaic.ro

Summary. We investigate the problem of reaching a configuration from another configuration in mobile membranes, and prove that the reachability can be decided by reducing it to the reachability problem of a version of pure and public ambient calculus without the capability open.

1 Introduction

Membrane systems (called also P systems) are introduced by Gh.Păun in [8, 9] as a class of parallel computing devices inspired by biology. The definition of this computing model starts from the observation that any biological system is a complex hierarchical structure, with a flow of materials and information which underlies their functioning. The membrane computing deals with the evolution of systems composed by objects, rules and membranes nested in other membranes. The P systems with mobile membranes [5] is a model which expresses mobility by the movement of membranes in such a system. The movement is given mainly by two operations: exocytosis and endocytosis.

Ambient calculus is a formalism introduced in [3] to describe concurrent and mobile computation. In contrast with other formalisms for mobile processes such as the π-calculus [7] (whose computational model is based on the notion of communication), the ambient calculus is based on the notion of movement. An ambient is a named location, and represents a unit of movement. Ambients mobility is controlled by the capabilities in, out, and open; the mobile ambients describe the migration of processes between certain boundaries.

The membrane systems and mobile ambients have similar structures and common concepts. Both have a hierarchical structure, work mainly with a notion of location, and are used to model various aspects on the distributed systems. The distributed features of mobile ambients are described in [3], and distributed algorithms for membrane systems are presented in [4].

In this paper we investigate the problem of reaching a certain configuration in mobile membranes starting from a given configuration. We prove that reachability in mobile membranes can be decided by reducing it to the reachability problem of a version of pure and public ambient calculus from which the open capability has been removed. In [1] it is proven that the reachability for this fragment of ambient calculus is decidable by reducing it to marking reachability for Petri nets, which is proven to be decidable in [6].

The structure of the paper is as follows. In Section 2 we present the mobile membrane systems, whereas in Section 3 we present a version of pure and public mobile ambients without the capability open. The core of the paper is represented by Section 4, where we investigate the reachability problem for mobile membranes. Conclusions and references end the paper.

2 Mobile Membranes Systems

Definition 1. *A* mobile membrane system *is a construct*

$$\prod = (V \cup \overline{V}, H \cup \overline{H}, \mu, w_1, \ldots, w_n, R), \quad \textit{where:}$$

1. $n \geq 1$ *(the initial* degree *of the system);*
2. $V \cup \overline{V}$ *is an alphabet (its elements are called* objects*), where* $V \cap \overline{V} = \emptyset$*;*
3. $H \cup \overline{H}$ *is a finite set of* labels *for membranes, where* $H \cap \overline{H} = \emptyset$*;*
4. μ *is a membrane structure, consisting of* n *membranes, labeled (not necessarily in a one-to-one manner) with elements of* H*;*
5. $w_1, w_2, \ldots, w_n$ *are* multisets of objects *placed in the* n *membranes of the system;*
6. R *is a finite set of* developmental rules*, of the following forms:*
   - *a)* $\overline{a}{\downarrow} \to \overline{a}{\downarrow}\ a{\downarrow}$, *for* $a{\downarrow} \in V$, $\overline{a}{\downarrow} \in \overline{V}$*;* replication rule
     The objects $\overline{a}{\downarrow}$ *are used to create new objects* $a{\downarrow}$ *without being consumed.*
   - *b)* $\overline{a}{\uparrow} \to \overline{a}{\uparrow}\ a{\uparrow}$, *for* $a{\uparrow} \in V$, $\overline{a}{\uparrow} \in \overline{V}$*;* replication rule
     The objects $\overline{a}{\uparrow}$ *are used to create new objects* $a{\uparrow}$ *without being consumed.*
   - *c)* $[\ a{\downarrow}\ ]_h\ [\ ]_a \to [\ [\ ]_h\ ]_a$, *for* $a, h \in H, a{\downarrow} \in V$*;* endocytosis
     An elementary membrane labeled h *enters the adjacent membrane labeled* a*, under the control of object* $a{\downarrow}$*. The labels* h *and* a *remain unchanged during this process; however the object* $a{\downarrow}$ *is consumed during the operation. Membrane* a *is not necessarily elementary.*
   - *d)* $[\ [\ a{\uparrow}\ ]_h\ ]_a \to [\ ]_h\ [\ ]_a$, *for* $a, h \in H, a{\uparrow} \in V$*;* exocytosis
     An elementary membrane labeled h *is sent out of a membrane labeled* a*, under the control of object* $a{\uparrow}$*. The labels of the two membranes remain unchanged; the object* $a{\uparrow}$ *of membrane* h *is consumed during the operation. Membrane* a *is not necessarily elementary.*
   - *e)* $[\ ]_{\overline{h}} \to [\ ]_{\overline{h}}[\ ]_h$ *for* $h \in H$, $\overline{h} \in \overline{H}$ division rules
     An elementary membrane labeled $\overline{h}$ *is divided into two membranes labeled by* $\overline{h}$ *and* h *having the same objects.*

In 3, $H \cap \overline{H} = \emptyset$ states that the membranes having labels from the set $\overline{H}$ can participate only in rules of type (e). Similarly, $V \cap \overline{V} = \emptyset$ in 2 states that the objects from $\overline{V}$ can participate only in rules of type (a) and (b)

The rules are applied using the following principles:

1. In biological systems molecules are divided into classes of different types. We make the same decision here and split the objects into four classes: $a \downarrow$ - objects that control the *endocytosis*, $a \uparrow$ - objects that control the *exocytosis*, and $\overline{a} \downarrow$, $\overline{a} \uparrow$ - objects that produce new objects from the first two classes without being consumed.
2. All the rules of type $(c), (d)$ are applied in parallel, non-deterministically choosing the rules, the membranes, and the objects, but in such a way that the parallelism is maximal; this means that in each step we apply a set of rules such that no further rule of type $(c), (d)$ can be added to the set, no further membranes and objects can evolve at the same time.
3. The membrane a from each rules of type $(c), (d)$ is said to be passive, while the membrane h is said to be active. In any step of a computation, any object and any active membrane can be involved in at most one rule, but the passive membranes are not considered involved in the use of rules (hence they can be used by several rules at the same time as passive membranes).
4. When a membrane is moved across another membrane, by endocytosis or exocytosis, its whole content (its objects) are moved.
5. If a membrane is divided, then its content is replicated in the two new copies.
6. The skin membrane can never be divided.
7. Not all the rules of type $(a), (b), (e)$ are applied whenever it is possible; we choose non-deterministically whether the rules of these types are applied.

According to these rules, we get transitions among the configurations of the system. For two mobile membrane systems M and N we say that M reduces to N if there is a sequence of rules applicable in the membrane system M in order to obtain the membrane system N.

3 Mobile Ambients

We describe a variant of pure and public mobile ambients (mobile ambients in which communication and name restriction are omitted); more details can be found in [1]. Given an infinite set of names $\mathcal{N}$ (ranged over by $m, n, \dots$), we define the set $\mathcal{A}$ of mobile ambients (denoted by $A, A', B, \dots$) together with their capabilities (denoted by $C, C', \dots$) as follows:

C	$::= \ in\ n \ \mid \ out\ n$	**Capabilities**
A	$::= \ A \mid B \ \mid \ C.\,A \ \mid \ n[\,A\,] \ \mid \ !A$	**Processes**

A movement $C.\,A$ is provided by the capability C, followed by the execution of process A. An entry capability $in\ n$ instructs the surrounding ambient to enter a sibling ambient labeled by n, while an exit capability *out* n intructs the surrounding ambient to exit its parent ambient labeled by n. An ambient $n[\ A\ ]$ represents a bounded place labeled by n in which a process A is executed. $A \,|\, B$ is a parallel composition of processes A and B. The process $!A$ denotes the unbounded replication of process A.

Processes of this calculus are grouped into equivalence classes, up to trivial syntactic restructuring, by the following structural congruence relation, $\equiv$, which is the least congruence satisfying the following requirements:

$$\begin{array}{ll}
A \,|\, B \equiv B \,|\, A & A \equiv B \text{ implies } A \,|\, A' \equiv B \,|\, A' \\
(A \,|\, B) \,|\, A' \equiv A \,|\, (B \,|\, A') & A \equiv B \text{ implies } !A \equiv !B \\
A \equiv A & A \equiv B \text{ implies } n[A] \equiv n[B] \\
A \equiv B \text{ implies } B \equiv A & A \equiv B \text{ implies } C.A \equiv C.B \\
A \equiv B,\ B \equiv A' \text{ implies } A \equiv A' &
\end{array}$$

The operational semantics of the mobile ambients is defined in terms of a reduction relation $\Rightarrow$ by the following axioms and rules:

Axioms:

(In) $n[\ in\ m.\, A \mid A'\ ] \mid m[\ B\ ] \ \Rightarrow\ m[\ n[\ A \mid A'\ ] \mid B\ ]$;

(Out) $m[\ n[\ out\ m.\, A \mid A'\ ] \mid B\ ] \ \Rightarrow\ n[\ A \mid A'\ ] \mid m[\ B\ ]$;

(Repl) $!A \Rightarrow\ A \mid !A$.

Rules:

(Comp) $\dfrac{A \ \Rightarrow\ A'}{A \mid B \ \Rightarrow\ A' \mid B}$ **(Amb)** $\dfrac{A \ \Rightarrow\ A'}{n[\ A\ ] \ \Rightarrow\ n[\ A'\ ]}$

(Struc) $\dfrac{A \equiv A',\ A' \ \Rightarrow\ B',\ B' \equiv B}{A \ \Rightarrow\ B}$.

The **Axioms** represent the one-step reductions for *in* and *out*, and the unfolding of replication. The **Rules** propagate reduction across ambient nesting, parallel composition and allow the use of equivalence during reduction. The **In** and **Out** rules are applied as soon as possible in a maximal parallel manner. We denote by $\Rightarrow^*$ the reflexive and transitive closure of the binary relation $\Rightarrow$.

4 Reachability Problem

In this section we prove that the problem of reaching a configuration (membranes and objects) starting from a certain configuration is decidable for the special class of mobile membranes systems introduced in Section 2.

Theorem 1. *For two arbitrary mobile membranes M_1 and M_2, it is decidable whether M_1 reduces to M_2.*

The main steps of the proof are as follows:

1. we reduce mobile membranes systems to pure and public mobile ambients without the capability *open*.

2. we show that the reachability problem for two arbitrary mobile membranes can be expressed as the reachability problem for the corresponding mobile ambients.
3. we use the result that the reachability problem is decidable for a fragment of pure and public mobile ambients without the capability *open*.

The following subsections are devoted to the proof of Theorem 1.

4.1 From Mobile Membranes to Mobile Ambients

We use the following translation steps:

1. any object $a\downarrow$ is translated into a capability *in a*;
2. any object $a\uparrow$ is translated into a capability *out a*;
3. any object $\overline{a}\downarrow$ is translated into a replication $!in\ a$
4. any object $\overline{a}\uparrow$ is translated into a replication $!out\ a$
5. a membrane h is translated into an ambient h
6. an elementary membrane $\overline{h}$ is translated into a replication $!h[\ ]$ where all the objects inside membrane h are translated into capabilities in ambient h using the above steps.

A correspondence exists between the rules from mobile membranes and the reduction rules from mobile ambients as follows:

- rule (c) corresponds to rule **(In)**
- rule (d) corresponds to rule **(Out)**
- rules $(a), (b), (e)$ correspond to instances of rule **(Repl)**

If we start with a mobile membrane system M, we denote by $\mathcal{T}(M)$ the mobile ambient obtained using the above translation steps. For example, starting from the membrane system $M = [m\downarrow\ m\uparrow]_n[\ ]_m$ we obtain $\mathcal{T}(M) = n[in\ m \mid out\ m] \mid m[\ ]$.

Proposition 1. *For mobile membrane systems M and N, M reduces to N by applying one rule if and only if $\mathcal{T}(M)$ reduces to $\mathcal{T}(N)$ by applying only one reduction rule.*

Proof (Sketch). Since M reduces N by applying one rule, then one of the rules of type $(a), \ldots, (e)$ is applied. We treat only the case when a rule of type (a) is applied, the others being treated in a similar manner.

If a rule $\overline{a}\downarrow \rightarrow \overline{a}\downarrow\ a\downarrow$ is applied, only one object from the membrane system M is used, namely $\overline{a}\downarrow$, to create a new object $a\downarrow$, thus obtaining the membrane system N. By translating the membrane system M into $\mathcal{T}(M)$ we have that $\overline{a}\downarrow$ is translated in $!in\ a$. By applying the reduction rule corresponding to (a) (namely the rule **(Repl)**) to $!in\ m$, then we have that $!in\ a \Rightarrow !in\ a \mid in\ a$, namely a new capability in a is created. We observe that $\mathcal{T}(a\downarrow) = in\ a$, which means that the obtained mobile ambient is in fact $\mathcal{T}(N)$.

According to Proposition 1 the reachability problem for mobile membranes can be reduced to a similar problem for mobile ambients.

4.2 From Mobile Ambients to Petri Nets

After translating the mobile membranes into a fragment of mobile ambients known to be decidable, we present for our fragment of mobile ambients the algorithm used in [1] to translate mobile ambients into a fragment of Petri nets, which is known to be decidable from [6]. The fragment of mobile ambients used here is a subset of the fragment of mobile ambients used in [1] and the difference is provided by the extra-rule $!A \Rightarrow !A \mid !A$ used in [1].

We observe that applying a reduction rule over a process either increases the number of ambients or leaves it unchanged. The only reduction rule that increases the number of ambients when applied is the rule **(Repl)**, while the other reduction rules leave the number of ambients unchanged. If we reach process B starting from process A, then the number of ambients of process B is known. Therefore, we can use this information to know how many times the reduction rule **(Repl)** is applied to replicate ambients. A similar argument does not hold for capabilities as they can be consumed by the reduction rules **(In)** and **(Out)**.

An ambient context $\mathcal{C}$ is a process in which may occur some holes (denoted by $\Box$). Starting from this observations we split a process into two parts: one will be a context containing ambients, whereas the other one will be a process without ambients.

In order to uniquely identify all the occurrences of replication, ambient, capability or hole $\Box$ within an ambient context or a process, we introduce a labeling system. Using a countable set of labels, we say that a process A or an ambient context $\mathcal{C}$ is well-labeled if any label occurs at most once in A or $\mathcal{C}$. We denote by $Amb(\mathcal{C})$ the multiset of ambients occurring in an ambient context $\mathcal{C}$.

I) labeled Transition System

For the reachability problem $A \Rightarrow^* B$, we denote by $\mathcal{C}_A$ a well-labeled ambient context and with θ_A a mapping from the set of holes in $\mathcal{C}_A$ to some labeled processes without replicable ambients such that $\theta_A(\mathcal{C}_A)$ is well-labeled, and $\theta_A(\mathcal{C}_A) = A$ ignoring labels.

A labeled transition system $L_{A,B}$ describes all possible reductions for the context $\mathcal{C}_A$: this includes reductions of replications and capabilities contained in $\mathcal{C}_A$ and in the processes associated with the holes of the context. The states of the labeled transition system $L_{A,B}$ are associative-commutative equivalent classes of ambient contexts, and for simplicity, we often identify a state as one of the representatives of its class.

We define a mapping $\theta_{L_{A,B}}$ which extends the mapping θ_A. Initially, $L_{A,B}$ contains (the equivalence class of) $\mathcal{C}_A$ as a unique state and we have $\theta_{L_{A,B}} = \theta_A$. We present in what follows the construction steps of $\theta_{L_{A,B}}$, where cap stands for in or out:

1. for any ambient context $\mathcal{C}$ from $L_{A,B}$ and for any labeled capability $\mathsf{cap}^w\mathsf{n}$ in $\mathcal{C}$ if this capability can be executed using one of the rules **(In)** or **(Out)** leading

to some ambient context $\mathcal{C}'$, then the state $\mathcal{C}'$ and a transition from $\mathcal{C}$ to $\mathcal{C}'$ labeled by $\mathsf{cap}^w\mathsf{n}$ are added to $L_{A,B}$.

2. for any ambient context $\mathcal{C}$ from $L_{A,B}$ and for any labeled replication $!^w$ in $\mathcal{C}$ such that the reduction rule **(Repl)** is applied, we define the ambient context $\mathcal{C}'$ as follows: $\mathcal{C}'$ is identical to $\mathcal{C}$ except that the subcontext $!^w\mathcal{C}_a$ in $\mathcal{C}$ is replaced by $!^w\mathcal{C}_a \mid \gamma(\mathcal{C}_a)$ in $\mathcal{C}'$; the mapping γ relabels $\mathcal{C}_a$ with fresh labels, such that $\mathcal{C}'$ is well-labeled. If $Amb(\mathcal{C}') \subseteq Amb(B)$ then the state $\mathcal{C}'$ and a transition from $\mathcal{C}$ to $\mathcal{C}'$ labeled by $!^w$ is added to $L_{A,B}$. Additionally, we define $\theta'_{L_{A,B}}$ as an extension of $\theta_{L_{A,B}}$ such that for all $\Box^{w'}$ in $\mathcal{C}_a$ we have:
   (i) $\theta'_{L_{A,B}}(\gamma(\Box^{w'}))$ and $\theta_{L_{A,B}}(\Box^{w'})$ are identical ignoring labels,
   (ii) labels in $\theta'_{L_{A,B}}(\gamma(\Box^{w'}))$ are fresh in the currently built transition system $L_{A,B}$
   (iii) $\theta'_{L_{A,B}}(\gamma(\Box^{w'}))$ is well-labeled.
   As a final step, we set $\theta_{L_{A,B}}$ to be $\theta'_{L_{A,B}}$.
3. for any ambient context $\mathcal{C}$ from $L_{A,B}$ and for any labeled hole $\Box^w$ in $\mathcal{C}$ and for any capability $\mathsf{cap}^w\mathsf{n}$ in the process $\theta_{L_{A,B}}(\Box^w)$, we consider the ambient context $\mathcal{C}_m$ identical to $\mathcal{C}$ except that $\Box^w$ in $\mathcal{C}$ has been replaced by $\Box^w \mid cap^w n$ in $\mathcal{C}_m$. If this capability $cap^w n$ can be consumed in $\mathcal{C}_m$ using one of the rules **(In)** or **(Out)** leading to some ambient context $\mathcal{C}'$, then the state $\mathcal{C}'$ and a transition from $\mathcal{C}$ to $\mathcal{C}'$ labeled by $cap^w n$ are added to transition system $L_{A,B}$.
4. for any ambient context $\mathcal{C}$ from $L_{A,B}$ and for any labeled hole $\Box^w$ in $\mathcal{C}$ associated by $\theta_{L_{A,B}}$ with a process of the form $!^{w'}A'$, if the replication $!^{w'}$ can be reduced in the process $\theta_{L_{A,B}}(\mathcal{C})$ using the rule **(Repl)**, then for any replication $!^{w''}$ in $\theta_{L_{A,B}}(\Box^w)$, a transition from $\mathcal{C}$ to itself, labeled by $!^{w''}$ is added to $L_{A,B}$.

In step 2 the reduction of a replication contained in the ambient context by means of the rule **(Repl)** is done only when the number of ambients in the resulting process is smaller than the number of ambients in the target process B, namely $Amb(\mathcal{C}') \subseteq Amb(B)$. This requirement is crucial as it implies that the transition system $L_{A,B}$ has only finitely many states.

As an example, we give in Figure 1 the labeled transition system associated with the process $n[!^1 in\ m.!^2 out\ m] \mid m[\ ]$ (we omit in this process unnecessary labels). We use the labeled replications $!^1$ and $!^2$ to distinguish between different replication operators which appear in the process.

We observe that the labeled transitions in $L_{A,B}$ for replications and capabilities from an ambient context correspond to reductions performed on processes. As shown in steps 3 and 4 the transitions applied for any capabilities or replications associated with the holes are independently of the fact that they are effectively at this point available to perform a transition.

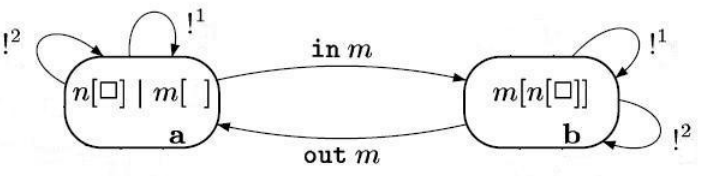

Fig. 1. A labeled transition system for the process $n[!^1\,\mathbf{in}\ m.!^2\mathbf{out}\ m] \mid m[\ ]$

II) From Processes Without Ambients to Petri Nets

In what follows we show how to build a Petri net from a labeled process without ambients. We denote by $\mathcal{E}(E)$ the set of all multisets that can be built with elements from the set E.

We recall that a Petri net is given by a 5-tuple $(\mathcal{P}, \mathcal{P}_i, \mathcal{T}, Pre, Post)$ with

- $\mathcal{P}$ a finite set of places;
- $\mathcal{P} \subseteq \mathcal{P}_i$ a set of initial places;
- $\mathcal{T}$ a finite set of transitions;
- $Pre, Post : \mathcal{T} \to \mathcal{E}(\mathcal{P})$ mappings from transitions to multisets of places.

We say that an ambient-free process is rooted if it is of the form $cap^w n.A'$ or of the form $!^w A'$. We define $PN_{A'}$ the Petri net associated with some rooted process A' as follows: places for $PN_{A'}$ are precisely rooted subprocesses of A', and A' itself is the unique initial place. Transitions are defined as the set of all capabilities $in^w n$, $out^{w'} n$ and replications $!^w$ occurring in A'. Finally, Pre and $Post$ are defined for all transitions as follows:

- $Pre(cap^w n) = \{cap^w n\}$ and $Post(cap^w n) = \emptyset$ if $cap^w n$ is a place in $PN_{A'}$.
- $Pre(cap^w n) = \{cap^w n.(A_1 \mid \ldots \mid A_k)\}$ and $Post(cap^w n) = \{A_1 \mid \ldots \mid A_k\}$ if $cap^w n.(A_1 \mid \ldots \mid A_k)$ is a place in $PN_{A'}$ $(A_1 \mid \ldots \mid A_k$ being rooted processes).
- $Pre(!^w) = Pre(!^w) = \{!^w A'\}$, $Post(!^w) = \{!^w A', A'\}$ and $Post(!^w) = \{!^w A'\}$ if $!^w A'$ is a place in $PN_{A'}$.

For $!^1 in\ m.!^2 out\ m$, we obtain the Petri net given in Figure 2.

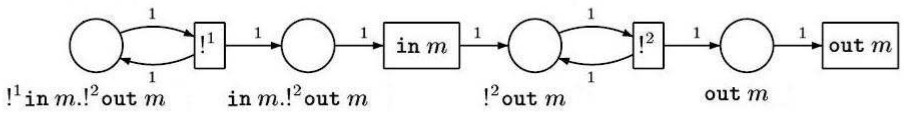

Fig. 2. A Petri net for the process $!^1\,\mathbf{in}\ m.!^2\mathbf{out}\ m$

We will denote by $PN_{\Box^w}$ the Petri net $PN(\theta_{L_{A,B}}(\Box^w))$, that is, the Petri net corresponding to the rooted ambient-free process associated with $\Box^w$ by $\theta_{L_{A,B}}$. In what follows we show how to combine the transition system $L_{A,B}$ and the Petri nets $PN_{\Box^w}$ into one single Petri net.

III) Combining the Transition System and Petri Nets

We first turn the labeled transition system $L_{A,B}$ into a Petri net $PN_L = (\mathcal{P}_L, \mathcal{P}^i_L, \mathcal{T}_L, Pre_L, Post_L)$ with:

- $\mathcal{P}_L$ a set of states of $L_{A,B}$;
- $\mathcal{P}^i_L$ a singleton set containing the state corresponding to $\mathcal{C}_A$, the ambient context of A;
- $\mathcal{T}_L$ the set of transitions of the form (s, l, s'), with
  - s and s' states from $L_{A,B}$;
  - l transition from s to s' in $L_{A,B}$.
- $Pre(t) = s$ and $Post(t) = \{s'\}$, for all transitions $t = (s, l, s')$.

We define the Petri net $PN_{A,B} = (\mathcal{P}_{A,B}, \mathcal{P}^i_{A,B}, \mathcal{T}_{A,B}, Pre_{A,B}, Post_{A,B})$ as:

- places (resp. initial places) from $PN_{A,B}$ are the union of places (resp. initial places) of PN_L and of each of the Petri nets $PN_{\Box^w}$ (for $\Box^w$ occurring in one of the states of $L_{A,B}$).
- transitions of $PN_{A,B}$ are precisely the transitions of PN_L.
- The mappings $Pre_{A,B}$ and $Post_{A,B}$ are defined for all transitions $t = (a, f, b)$:
  (i) $Pre_{A,B}(t) = \{a\}$ and $Post_{A,B}(t) = \{b\}$, if f does not occur as a transition in any $PN_{\Box^w}$ (for $\Box^w$ occurring in one of the states of $L_{A,B}$)
  (ii) $Pre_{A,B}(t) = \{a\} \cup Pre_{\Box^w}(f)$ and $Post_{A,B}(t) = \{b\} \cup Post_{\Box^w}(f)$, if f is a transition of $PN_{\Box^w}$ ($Pre_{\Box^w}$(resp. $Post_{\Box^w}$) being the mapping Pre (resp. $Post$) of $PN_{\Box^w}$).

4.3 Deciding Reachability

We recall that for a Petri net $PN = (\mathcal{P}, \mathcal{P}^i, \mathcal{T}, Pre, Post)$, a marking m is a multiset from $\mathcal{E}(P)$. A transition t is enabled by a marking m if $Pre(t) \subseteq m$. Executing an enabled transition t for a marking m gives a marking m' defined as $m' = (m \backslash Pre(t)) \cup Post(t)$ (where $\backslash$ stands for the multiset difference). A marking m' is reachable from m if there exists a sequence $m_0, \ldots, m_k$ of markings such that $m_0 = m$, $m_k = m'$ and for each m_i, m_{i+1}, there exists an enabled transition for m_i whose execution gives m_{i+1}.

Theorem 2 ([6]). *For all Petri nets P, for all markings m, m' for P, one can decide whether m' is reachable from m.*

For the reachability problem $A \Rightarrow^* B$ over ambients, we consider the Petri net $PN_{A,B}$ and the initial marking m_A defined as $m_A = \mathcal{P}^i_{A,B}$. In Figure 3 is depicted the initial marking for the process $n[!^1 in\ m.!^2 out\ m] \mid m[\,]$ as a combination of the labeled transition system from Figure 1 and the Petri net from Figure 2.

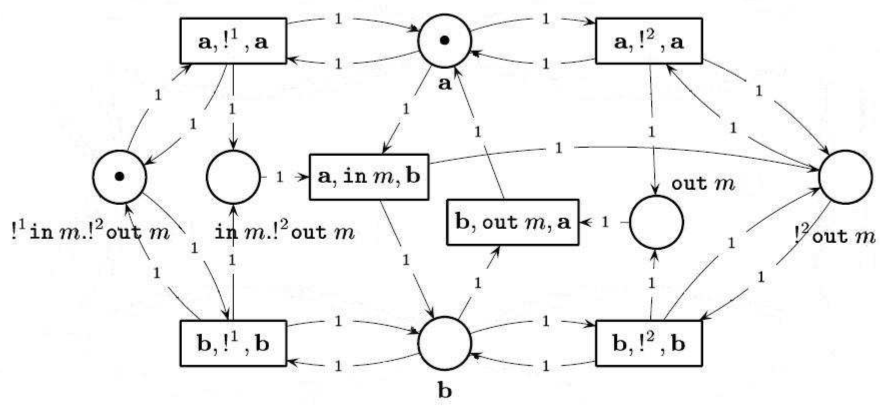

Fig. 3. The Petri net for the labeled process $n[!^1\mathbf{in}\ m.!^2\mathbf{out}\ m\ ] \mid m[\ ]$

It should be noticed that for any marking m reachable from m_A, m contains exactly one occurrence of a place from $\mathcal{P}_L$. Roughly speaking, to any reachable marking corresponds exactly one ambient context. Moreover, the execution of one transition in the Petri net $PN_{A,B}$ simulates a reduction from $\Rightarrow$.

We define now $\mathcal{M}_B$, the set of markings of $PN_{A,B}$ corresponding to B. Intuitively, a marking m belongs to $\mathcal{M}_B$ if m contains exactly one occurrence $\mathcal{C}$ of a place from $\mathcal{P}_L$ (that is, representing some ambient context) and in the context $\mathcal{C}$, the holes can be replaced with processes without ambients to obtain B. Each of the processes without replication must correspond to a marking of the sub-Petri net associated with the hole it fills up. $\mathcal{M}_B$ is defined as the set of markings m for $PN_{A,B}$ satisfying:

(i) in m there exists exactly one ambient context $\mathcal{C}_m$;
(ii) ignoring labels, $\sigma_m(\mathcal{C}_m)$ is equal to B modulo associative-commutative, for the substitution σ_m from holes $\Box^w$ occurring in $\mathcal{C}_m$ to processes without ambients defined as: $\sigma_m(\Box_m) = P_1 \mid \ldots \mid P_k$ for $\{P_1, \ldots, P_k\}$ the multiset corresponding to the restriction of m to the places of $PN_{\Box^w}$
(iii)for all holes $\Box^w$ occurring in some state of the transition system $L_{A,B}$ but not in $\mathcal{C}_m$, the restriction of m to places of $PN_{\Box^w}$ is precisely the set of initial places from $PN_{\Box^w}$.

We adapt the results presented in [1] to a restricted fragment of mobile ambients.

Proposition 2. *For a Petri net $PN_{A,B}$, there are only finitely many markings corresponding to the process B, and their set $\mathcal{M}_B$ can be computed.*

The translation correctness is ensured by the following result.

Proposition 3. *For all processes A, B we have that $A \Rightarrow B$ if and only if there exists a marking from $\mathcal{M}_B$ such that m_B is reachable from m_A in $PN_{A,B}$.*

Using the Proposition 3 and Theorem 2, we can decide whether an ambient A can be reduced to an ambient B.

Theorem 3. *For two arbitrary ambients A and B from our restricted fragment, it is decidable whether A reduces to B.*

5 Conclusions

In this paper we have investigated the problem of reaching a configuration in mobile membranes starting from another configuration. In order to do this we use [1] where the reachability problem for a fragment of ambient calculus is proven to be decidable, namely the pure and public ambient calculus except the *open* capability. The same problem is tackled in [2], but the authors do not take into account the replication of ambients, which is used to simulate the division rules in mobile membranes. We proved that the reachability can be decided by reducing this problem to the reachability problem of a version of pure and public ambient calculus from which the open capability has been removed.

References

1. I. Boneva, J.-M. Talbot. When Ambients Cannot be Opened. *Foundations of Software Science and Computation Structures*, Lecture Notes in Computer Science vol.2620, Springer, 169-184, 2003.
2. N. Busi, G. Zavattaro. Deciding Reachability in Mobile Ambients. *Proceedings of European Symposium on Programming*, Lecture Notes in Computer Science vol.3444, Springer, 248-262, 2005.
3. L. Cardelli, A. Gordon. Mobile Ambients. *Foundations of Software Science and Computation Structures*, Lecture Notes in Computer Science vol.1378, Springer, 140-155, 1998.
4. G. Ciobanu. Distributed Algorithms over Communicating Membrane Systems. *BioSystems* vol.70, 123-133, 2003.
5. S.N. Krishna, Gh. Păun. P Systems with Mobile Membranes. *Natural Computing*, vol.4(3), 255-274, 2005.
6. E.W. Mayr. An Algorithm for the General Petri Net Reachability Problem. *SIAM Journal of Computing*, vol.13(3), 441-460, 1984.
7. R. Milner. *Communicating and mobile systems: the π-calculus.* Cambridge University Press, 1999.
8. Gh. Păun. Computing with membranes. *Journal of Computer and System Sciences*, vol.61(1), 108-143, 2000.
9. Gh. Păun. *Membrane Computing. An Introduction.* Springer, 2002.