Spatial P systems

**Roberto Barbuti · Andrea Maggiolo-Schettini · Paolo Milazzo ·
Giovanni Pardini · Luca Tesei**

Abstract We present Spatial P systems, a variant of P systems which embodies the concept of space and position inside a membrane. Objects in membranes are associated with positions. Rules specify, in the usual way, the objects which are consumed and the ones which are produced; in addition, they can specify the positions of the produced objects. Objects belong to two different sets: the set of *ordinary* objects and the set of *mutually exclusive* objects. Every position inside a membrane can accommodate an arbitrary number of ordinary objects, but at most one mutually exclusive object. We prove that Spatial P systems are universal even if only non-cooperating rules are allowed. We also show how Spatial P systems can be used to model the evolution of populations in presence of geographical separations.

Keywords Membrane computing · P systems · Spatial modeling · Universality

1 Introduction

P systems were introduced by Păun ([2000](#)) as distributed parallel computing devices inspired by the structure and the functioning of a living cell. A P system consists of a

R. Barbuti · A. Maggiolo-Schettini · P. Milazzo · G. Pardini (✉)
Dipartimento di Informatica, Università di Pisa, Largo Pontecorvo 3, 56127 Pisa, Italy
e-mail: pardinig@di.unipi.it

R. Barbuti
e-mail: barbuti@di.unipi.it

A. Maggiolo-Schettini
e-mail: maggiolo@di.unipi.it

P. Milazzo
e-mail: milazzo@di.unipi.it

L. Tesei
School of Science and Technology, Università di Camerino, Via Madonna delle Carceri 9,
62032 Camerino, MC, Italy
e-mail: luca.tesei@unicam.it

⚛ Springer

hierarchy of membranes, each of them containing a multiset of *objects*, representing molecules, a set of *evolution rules*, representing chemical reactions, and possibly other membranes. For each evolution rule there are two multisets of objects, describing the reactants and the products of the chemical reaction. A rule in a membrane can be applied only to objects in the same membrane. Some objects produced by the rule remain in the same membrane, others are sent *out* of the membrane, others are sent *into* the inner membranes, which are identified by their labels. Evolution rules can be applied more than once to different objects, with *maximal parallelism*, namely it cannot happen that some evolution rule is not applied when the objects needed for its triggering are available and not consumed by the application of any other rule.

Many variants and extensions of P systems exist that include features which increase their expressiveness and which are based on different evolution strategies. A basic extension is that of P systems with dissolution rules that allow a membrane to disappear and release in the surrounding region all the objects it contains. P systems with priorities provides a priority relationship among the evolution rules of each membrane and can influence the applicability of such rules. In P systems with promoters and inhibitors the applicability of evolution rules depends on the presence of at least one occurrence and on the absence of a specific object, respectively. We also mention two variants of P systems dealing with objects crossing membranes. The first extension provides symport/antiport rules, which allow simultaneous trans-membrane transportation of objects either in the same direction (symport) or in opposite directions (antiport). The second extension provides Membrane Channels (PMC Systems), in which the passage of objects through membranes is allowed only through specific channels associated with membranes. Finally, we mention Metabolic P systems by Manca et al. (2004), a quantitative extension of P system form modeling metabolic processes.

See Păun (2002); Bottoni et al. (2002) and Barbuti et al. (2008, 2009b) for the definition of variants of P systems, and P Systems web page (2009) for a complete bibliography.

In this paper we introduce Spatial P systems, an extension of P systems with a concept of space. Membranes and objects are positioned in a two-dimensional discrete space. Evolution rules are associated with membranes, and take into account the positions of objects. At any time during system evolution, in a position there may be a number of *ordinary* objects, but *only* one object from the set of *mutually exclusive* objects. We prove that Spatial P systems are universal, even when evolution rules are restricted to *non-cooperating* ones. As an example of modeling, we present a model for the evolution of "ring species", where populations of a same species evolve independently in presence of a geographic barrier.

The need of representation of space in Biology has brought to include spatial features in formalisms for modeling biological systems (John et al. 2008; Barbuti et al. 2009a; Cardelli and Gardner 2009; Bartocci et al. 2009). As regards P systems, we mention the extension of P systems proposed by Besozzi et al. (2008), used to describe the behavior of "metapopulations". Metapopulations are local populations living in spatially separated areas (called "patches"), where populations can interact, and individuals can disperse from a patch to nearby patches. Models for metapopulations aim at discovering how the fragmented habitat influences local and global population behavior. In the proposed model, objects are used to model different species (predators and preys), and patches are represented as elementary membranes in a flat membrane structure. Patches form the nodes of an undirected weighted graph, with edges describing a neighborhood relation between patches, modeling spatial proximity. Costs associated with edges model the effort of individuals for migrating from a patch to another.

In computer science, the computational formalism of Cellular Automata (CA) (Neumann [1966]) uses a representation of the space analogous to that of Spatial P systems. CAs were developed as a computational tool inspired by biological behaviors, and have been later used for modeling biological systems. A cellular automaton is composed of a finite grid of *cells*, where each cell has an associated state taken from a finite set of different states. Time is discrete, and at each step of the evolution all the cells change state in accordance with a *rule*, which is characteristic of the particular cellular automaton model. The rule is deterministic and "local", in the sense that the new state of a cell is determined only on the basis of the previous states of cell itself and of nearby cells. In Spatial P system, in contrast, cells contain objects, and interactions occur between objects. For this reason, each position in a Spatial P system can accommodate any number of objects, thus possibly representing an infinite number of different "states".

2 Spatial P systems

In this section we recall the definition of standard P systems, and then we present its extension with spatial features. We denote multisets over a finite alphabet as strings. More precisely, let V^* be the set of all strings over an alphabet V, including the empty one, denoted by λ. For $a \in V$ and x in V^* we denote by $|x|_a$ the number of occurrences of a in x. If $V = \{a_1,\ldots,a_n\}$ (the ordering is important), then the Parikh mapping of x is defined by $\Psi_V(x) = (|x|_{a_1}, \ldots, |x|_{a_n})$, called Parikh vector. The definition is extended to languages in the natural way. A string x represents the multiset over V with the multiplicities of objects $a_1,\ldots,a_n$ as given by $\Psi_V(x)$.

2.1 P systems

A P system consists of a *hierarchy of membranes* that do not intersect, with a distinguishable membrane, called the *skin membrane*, surrounding all of them. As usual, we assume membranes to be labeled by natural numbers. Given a set of objects V, a membrane m contains a multiset of *objects* in V^*, a set of *evolution rules*, and possibly other membranes, called *child* membranes (m is also called the *parent* of its child membranes). Objects represent molecules swimming in a chemical solution, and evolution rules represent chemical reactions that may occur inside the membrane containing them. For each evolution rule there is a multiset of objects representing the reactants, and a multiset of objects representing the products of the chemical reaction. A rule in a membrane m can be applied only to objects in m, meaning that its reactants are contained in m, and not in its child membranes. The rule must contain target indications, specifying the membranes where the new objects produced by applying the rule are sent. The new objects either remain in m, or can be sent out of m, or can be sent into one of its child membranes, precisely identified by its label.

Formally, the products of a rule are denoted with a multiset of *messages* of the forms: (i) v_{here}, meaning that the multiset of objects v produced by the rule remains in the same membrane m; (ii) v_{out}, meaning that the multiset of objects v produced by the rule are sent out of m; (iii) v_{in_l}, meaning that the multiset of objects v produced by the rule are sent into the child membrane l.

Let *TAR* be the set of message targets $\{here, out\} \cup \{in_i \mid i \in \mathbb{N}\}$. Given a set of objects O we denote with O_{tar} the corresponding set of messages $O \times TAR$. Hence, we denote with V_{tar} the set of all messages and we can define an evolution rule as a rule $u \to v$ such that

$u \in V^*$ and $v \in V^*_{tar}$. The length of the left-hand side u of an evolution rule is called the *radius* of such a rule. If a P system contains rules of radius greater than one, then it is called a *cooperating* system. Otherwise, it is called *non-cooperating*.

Application of evolution rules is done with maximal parallelism, namely at each evolution step a multiset of instances of evolution rules is chosen non-deterministically such that no other rule can be applied to the system obtained by removing all the objects necessary to apply all the chosen rules.

A P system has a tree-structure in which the skin membrane is the root and the membranes containing no other membranes are the leaves. We assume membrane labels to be unique. A membrane structure can be represented as a balanced sequence of labeled brackets and, graphically, as a Venn diagram.

Definition 1 A *P system* is a tuple $\Pi = (V, \mu, w_1, \ldots, w_n, R_1, \ldots, R_n)$, where:

- V is a finite *alphabet* whose elements are called *objects*;
- $\mu \subset \mathbb{N} \times \mathbb{N}$ describes the tree-structure of membranes, where $(i, j) \in \mu$ denotes that the membrane labeled by j is contained in the membrane labeled by i;
- w_i, with $1 \leq i \leq n$, are strings from V^* representing multisets over V associated with membranes 1, 2,…,n of μ;
- R_i, with $1 \leq i \leq n$, are finite sets of *evolution rules* associated with membranes 1, 2,…,n of μ.

A sequence of transitions between configurations of a given P system Π is called a *computation*. A computation is *successful* if and only if it reaches a configuration in which no rule is applicable. The result of a successful computation is the multiset of objects sent out of the skin membrane during the computation. Unsuccessful computations (computations which never halt) yield no result. Given a P system Π whose set of objects is V, let $U \subseteq V$ be the set of objects that can be sent out of the skin membrane, namely objects that appear in the right-end side of a rule in R_1 with target *out*. The result $x \in U^*$ of a computation of Π can be mapped into a vector of natural numbers by the Parikh mapping $\Psi_U(x)$. The set of all vectors of natural numbers computed by Π is denoted $Ps(\Pi)$. Following Păun (2002), let us denote by $P_n(\alpha)$ the class of P Systems with at most $n \geq 1$ membranes and using rules of type α, where $\alpha = coo$ indicates that cooperating rules are allowed and $\alpha = ncoo$ that only non-cooperating rules are used. When the number of membranes is not bounded we replace n with *. Let $PsP_n(\alpha)$ be the family of sets of vectors of natural numbers computed by the P systems of class $P_n(\alpha)$.

2.2 Extension with space

We extend P systems by embedding membranes and objects into the two-dimensional space with natural coordinates $\mathbb{N}^2$. Membranes have rectangular shape and, as for normal P systems, can be nested. The spatial description of a membrane is given in terms of the following parameters: (i) the position $p \in \mathbb{N}^2$ of its bottom-left corner with respect to the parent membrane, (ii) the membrane extents along the two dimensions, i.e. its width w and height h. Width and height cannot be null: $w, h \in \mathbb{N}^+$. There is always a distinguished skin membrane, which contains all other membranes and objects. We assume the skin membrane to be labeled with 1, and be positioned in (0, 0) with respect to the global coordinate system. Nesting of membranes has to satisfy some intuitive constraints: sibling membranes must not overlap, and membranes cannot exceed the bounds of their parent membranes. Moreover, we do not allow membrane edges to be *adjacent*, namely the Manhattan

distance between any two edges must be at least 1. (The Manhattan distance between two positions $(x_1, y_1), (x_2, y_2) \in \mathbb{Z}^2$ is $|x_1 - x_2| + |y_1 - y_2|$.)

An example of Spatial P system is shown in Fig. 1. Membrane 1, having width 8 and height 5, contains two membranes labeled 2 and 3. The spatial extension of membrane 2 is described by its bottom-left corner in (1, 1), its width 3 and height 3. For membrane 3, its position is (5, 3), and its dimensions are (2, 1). Three objects a are contained in membrane 2, at positions (1, 1), (2, 1) and (2, 0). Membrane 3 contains objects a and c in position (0, 0). All other positions, in all regions, are empty. Figure 2 shows a fragment of an invalid Spatial P system structure, in which the bottom edge of membrane 2 is adjacent to the bottom edge of the parent membrane, and the top edge of membrane 2 is adjacent to the bottom edge of membrane 3.

The structure of membranes can be seen as a partition of the space bounded by the skin membrane, where a position belongs to a membrane if and only if it is contained within its bounds and not contained in any other child membrane. The set of all positions belonging to a membrane is called a *region*. For example, with respect to Fig. 2, region of membrane 2 is the set $reg(2) = \{(x, y) | 1 \leq x, y \leq 3\}$, region of membrane 3 is $reg(3) = \{(5, 3), (6, 3)\}$, and region of membrane 1 contains the remaining positions: $reg(1) = \{(x, y) | 0 \leq x \leq 7, 0 \leq y \leq 4\} \setminus reg(2) \setminus reg(3)$.

Each object in a Spatial P system model is associated with a position in the region of a membrane. There are two kinds of objects, *ordinary* objects and *mutually exclusive* (*ME*) objects, which are represented by two disjoint sets V and E, respectively. The difference between them is that the positions of *ME* objects are constrained, since two *ME* objects are not allowed to occupy the same position at the same time. No constraints are imposed to ordinary objects.

A set of evolution rules is associated with each membrane. Evolution rules are either of the form (i) $u \rightarrow v$ or (ii) $u_1 - u_2 \rightarrow v_1 - v_2$, where u_1 and u_2 are strings over objects, and v_1 and v_2 are strings of messages (where each message is composed of a multiset of objects, and a target indication). A rule of the form (i), $u \rightarrow v$, is meant to be applied to each position forming the membrane region, provided that all the objects u appear in the position. String v specifies the products of the rule and their resulting positions. A rule of the form (ii), $u_1 - u_2 \rightarrow v_1 - v_2$, describes a simultaneous application of two rules, $u_1 \rightarrow v_1$ and $u_2 \rightarrow v_2$, to two adjacent positions inside the membrane region. Two positions are adjacent if and only if their Manhattan distance is exactly 1.

Besides the abilities of sending an object either into an inner membrane or out of the membrane, rules have to specify for the objects which remain in the same membrane, their resulting position inside the membrane region. Messages are of the following forms:

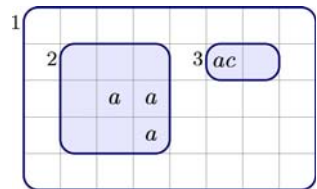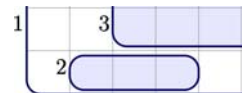**Fig. 1** An example of Spatial P system

Fig. 2 Example of adjacent edges (not allowed)

- $v_{\delta p}$, with $\delta p \in \mathbb{Z}^2$, meaning that the multisets of objects v are added to position $p + \delta p$ relative to the position p in which the rule is applied;
- v_{out}, meaning that the multisets of objects v are to be sent out of the membrane;
- v_{in_l}, meaning that the multisets of objects v are to be sent into the child membrane identified by l.

Note that there is not an explicit target *here*, used in conventional P systems for messages of the form v_{here}. In Spatial P systems, target *here* means "in the current position" (inside membrane region), and thus it has a narrower meaning than in conventional P systems. It can be defined as an alias for the null position $here = (0, 0)$. We also use the following abbreviations for the relative positions δp denoting adjacent positions: $N = (0, 1)$, $S = (0, -1)$, $E = (1, 0)$, $W = (-1, 0)$.

For example, the rule $A \rightarrow (b)_{(2,0)}(c)_{out}(d)_{in_2}$ can be applied to an object A, which results in an object b in position $p + (2, 0)$ relative to the current position p; an object c being sent out of the membrane; and an object d being sent into the inner membrane labeled 2.

Formally, let *TAR* denote the set of message targets $\mathbb{Z}^2 \cup \{out\} \cup \{in_i \mid i \in \mathbb{N}\}$. Given a set of objects O we denote with O_{tar} the corresponding set of messages $O \times TAR$. Hence, in evolution rules (i) $u \rightarrow v$ and (ii) $u_1 - u_2 \rightarrow v_1 - v_2$, we have $u, u_1, u_2 \in (V \cup E)^*$ and $v, v_1, v_2 \in ((V \cup E)_{tar})^*$. For Spatial P Systems, *cooperating* rules are both rules of type (i) with radius greater than one, and rules of type (ii).

Definition 2 A *Spatial P system* is a tuple $(V, E, \mu, \sigma, W^{(1)}, \ldots, W^{(n)}, R_1, \ldots, R_n)$ where:

- V and E are disjoint *alphabets*: elements of V are called *ordinary objects*, while elements of E are called *mutually exclusive objects* (*ME* objects);
- $\mu \subset \mathbb{N} \times \mathbb{N}$ describes the tree-structure of membranes, where $(i, j) \in \mu$ denotes that the membrane labeled by j is contained in the membrane labeled by i;
- $\sigma : \{1, \ldots, n\} \rightarrow \mathbb{N} \times \mathbb{N} \times \mathbb{N}^+ \times \mathbb{N}^+$ describes spatial position and dimensions of membranes; formally $\sigma(i) = (x_i, y_i, w_i, h_i)$ gives the position (x_i, y_i) of bottom left corner of membrane i, its width w_i and its height h_i (for the skin membrane, labeled 1, $\sigma(1) = (0, 0, w_1, h_1)$ for some w_1, h_1); Function σ must satisfy the following constraints:

  - $\forall(k, i) \in \mu. (0 < x_i < w_k - w_i) \wedge (0 < y_i < h_k - h_i)$;
  - $\forall(k, i), (k, j) \in \mu.$

    $$\neg(x_j \leq x_i \leq x_j + w_j) \wedge \neg(x_j \leq x_i + w_i \leq x_j + w_j) \wedge$$
    $$\neg(y_j \leq y_i \leq y_j + h_j) \wedge \neg(y_j \leq y_i + h_i \leq y_j + h_j);$$

- $W^{(i)} = \{ w_{x,y}^{(i)} \}$ with $1 \leq i \leq n$ are sets of strings $w_{x,y}^{(i)} \in (V \cup E)^*$, with $0 \leq x < w_i$, $0 \leq y < h_i$, where each string $w_{x,y}^{(i)}$ represents a multiset over $V \cup E$ associated with position (x, y) inside membrane $i \in \{1, 2, \ldots, n\}$ of μ, where (x, y) corresponds to a position in the region of i;
- R_i, with $1 \leq i \leq n$, are finite sets of *evolution rules* associated with membranes $1, 2, \ldots, n$ of μ.

Given a position $p = (x, y)$ in membrane i, an evolution rule $u \rightarrow v$ is *p-enabled* iff:

- if the rule specifies any *out* target, then p is adjacent to an edge of membrane i; formally, given the membrane width w_i and height h_i, the rule is enabled iff $x \in \{0, w_i - 1\}$ or $y \in \{0, h_i - 1\}$;

– if the rule specifies a target in_j, then j is a child membrane of i, and p is adjacent to it; formally, given $\sigma_j = (x_j, y_j, w_j, h_j)$, the rule is enabled iff $x \in \{x_j - 1, x_j + w_j\}$ $\wedge y_j \leq y < y_j + h_j$ or $y \in \{y_j - 1, y_j + h_j\} \wedge x_j \leq x < x_j + w_j$;
– for any target position δp, the resulting position $p' = p + \delta p$ with respect to the current position p is contained in membrane region (i.e. p' is inside membrane bounds and does not overlap with an inner membrane).A multiset of evolution rules of the form $u \to v$ is *applicable* to a position p inside a region of the system iff: each evolution rule is p-enabled, and all reactant objects (with their multiplicities) are present in p.

In each step of the evolution of a Spatial P system, some evolution rules are chosen and applied to the system state, by removing all reactant objects and adding all the products. In particular, in each step, for every membrane i, the following multisets of evolution rules are chosen non-deterministically:

– for every position p in membrane region, a multiset $M_1^{(p)}$ of evolution rules of the form $u \to v$ is chosen;
– for every pair of adjacent positions p, q in membrane region, a multiset $M_2^{(p,q)}$ of evolution rules of the form $u_1 - u_2 \to v_1 - v_2$ is chosen.

The chosen multisets of rules must satisfy some constraints. First of all, they must be applicable on the whole, namely for each position x in membrane region, the multiset of rules $M_1^{(x)} \cup \{u_1 \to v_1 \mid \exists q.\, u_1 - u_2 \to v_1 - v_2 \in M_2^{(x,q)}\} \cup \{u_2 \to v_2 \mid \exists q.\, u_1 - u_2 \to v_1 - v_2 \in M_2^{(q,x)}\}$ must be applicable. Moreover, they are required to be *valid*, namely two ME objects are forbidden to end up occupying the same position p' at the end of the step. However, note that, during the step, a ME object can disappear from a position and another one can take its place. Finally, as with standard P systems, the chosen multisets of rules must be *maximal*. Maximality means that, considering the system obtained by removing the multiset of reactant objects from all the chosen evolution rules, no other rule can be applied to that system while still preserving the validity.

For a message v_{in_j}, the objects are placed in the nearest position p' in the region of j, with respect to the current position. For a message v_{out}, the objects are placed in one of the nearest positions outside the membrane. In case the output position for a v_{out} message is not unique, as is the case when the rule is applied to a vertex position, then the output position is chosen non-deterministically along the horizontal and vertical direction. Figure 3 shows a fragment of Spatial P system, with three objects a in different positions inside a membrane. Arrows indicate the resulting position of the products, after the application of a rule $a \to a_{out}$ to the two objects a adjacent to membrane edges. The objects sent out of the skin membrane disappear from the system.

The definitions of *computation* and successful computation from standard P systems also apply to Spatial P systems. Therefore, the result of a successful computation is represented by the multiset of objects sent out of the skin membrane during the evolution, which are described by a Parikh vector. The set of all vectors computed by a Spatial P

Fig. 3 Possible outside positions for rule $a \to a_{out}$
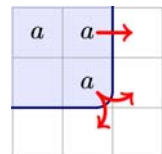
system Π is denoted $Ps(\Pi)$. Let us denote by $SP_n(\alpha,\beta)$ the class of Spatial P Systems with at most $n \geq 1$ membranes and using rules of type α, where $\alpha \in \{coo, ncoo\}$ as above, and $\beta \in \{me, nme\}$. We indicate with me that ME objects can be used, and with nme that their use is not allowed. Analogously to what done for P systems, when the number of membranes is not bounded we replace n with $*$. Let $PsSP_n(\alpha,\beta)$ be the family of sets of vectors of natural numbers computed by the Spatial P systems of class $SP_n(\alpha,\beta)$.

3 Universality of Spatial P systems

In this section we first prove that Spatial P systems are not universal when only non-cooperating rules are used and ME objects are not allowed. Then we prove that if ME objects can be used then universality is reached.

Theorem 1 $PsSP_*(ncoo, nme) \subseteq PsP_*(ncoo)$.

Proof   We show how to translate a Spatial P system with only non-cooperating rules and no ME objects $(V, \varnothing, \mu, \sigma, W^{(1)},\ldots,W^{(n)}, R_1,\ldots,R_n)$ into an equivalent classical P system $(V', \mu, w_1,\ldots,w_n, R'_1,\ldots,R'_n)$ with only non-cooperating rules.

The idea is to use the spatial information to translate each evolution rule into a set of rules which take into account the position of objects (that becomes part of the object name as a superscript) and the spatial membrane structure.

More formally, we discard the spatial description σ, but maintain the membrane structure μ. The objects of the translated P system are $V' = V \cup \{a^{x,y}|a \in V, (x,y) \in \{0,\ldots,h_1 - 1\} \times \{0,\ldots,w_1 - 1\}\}$. Every initial object $a_{x,y} \in W^{(i)}$ is mapped into the object $a^{x,y} \in w_i$. Given an evolution rule $a \rightarrow v_1 \ldots v_k \in R_i$, it is mapped into at most $h_i \cdot w_i$ rules by the partial function $\mathcal{T}_{\mathrm{rule}}(a \rightarrow v_1 \ldots v_k, i, x, y, \mu, \sigma)$, applied to all positions $(x, y) \in reg(i)$. $\mathcal{T}_{\mathrm{rule}}$ is defined as follows:

$$\mathcal{T}_{\mathrm{rule}}(a \rightarrow v_1 \ldots v_k, i, x, y, \mu, \sigma) = \begin{cases} a^{x,y} \rightarrow v'_1 \ldots v'_k & \text{if } cond(v_1 \ldots v_k, i, x, y, \mu, \sigma) \\ \bot & \text{otherwise;} \end{cases}$$

where $v'_j = \mathcal{T}_{\mathrm{tar}}(v_j, i, x, y, \mu, \sigma)$ for every $j = 1,\ldots,k$, and $cond(v_1 \cdots v_k, i, x, y, \mu, \sigma) \equiv \forall j \in \{1,\ldots,k\}. \mathcal{T}_{\mathrm{tar}}(v_j, i, x, y, \mu, \sigma) \neq error$.

The function $\mathcal{T}_{\mathrm{tar}}$ translates an object v_j into a new object v'_j whose superscripts and target are derived from the position (x, y) and the spatial information. It returns an error when the evolution rule $a \rightarrow v_1 \ldots v_k$ is not (x, y)-enabled because of the target of v_j. For instance, $\mathcal{T}_{\mathrm{tar}}(v_{(x',y')}, i, x, y, \mu, \sigma) = v_{here}^{x+x',y+y'}$ provided that position $(x + x', y + y') \in reg(i)$, otherwise the translation results in an error. The other cases of $\mathcal{T}_{\mathrm{tar}}$ are similar.

By definition of $\mathcal{T}_{\mathrm{rule}}$, a rule $a \rightarrow v_1 \ldots v_k$ is translated for a position (x, y) if and only if the rule is (x, y)-enabled, otherwise the rule is not translated for that position.

Finally, the rules sending objects out of the skin membrane must drop their superscript, making the output exactly equal to that of the original Spatial P System. $\square$

Since P systems with only non-cooperating rules are not universal (Păun 2002), Theorem 3 says that also Spatial P systems with only non-cooperating rules and no ME objects are not universal. In the following we show that we can reach universality by allowing ME objects.

In the proof of this result we show that any matrix grammar with appearance checking can be simulated by a Spatial P system with non-cooperating rules. As a consequence, before giving the result and its proof, we recall from Păun (2002) the definition of such variant of matrix grammars and some related notions.

Matrix grammars with appearance checking A (context-free) matrix grammar with appearance checking is a tuple $G = (N, T, S, M, F)$, where N and T are disjoint alphabets of non–terminals and terminals, respectively, $S \in N$ is the axiom, M is a finite set of matrices, namely sequences of the form $(A_1 \rightarrow x_1,\ldots,A_n \rightarrow x_n)$ of context–free rules over $N \cup T$ with $n \geq 1$, and F is a set of occurrences of rules in the matrices of M. For a string w, a matrix $m : (r_1,\ldots,r_n)$ can be executed by applying its rules to w sequentially in the order in which they appear in m. Rules of a matrix occurring in F can be skipped during the execution of the matrix if they cannot be applied, namely if the symbol in their left-hand side is not present in the string.

Formally, given $w, z \in (N \cup T)^*$, we write $w \Longrightarrow z$ if there is a matrix $(A_1 \rightarrow x_1,\ldots,A_n \rightarrow x_n)$ in M and the strings $w_i \in (N \cup T)^*$ with $1 \leq i \leq n + 1$ such that $w = w_1, z = w_{n+1}$ and, for all $1 \leq i \leq n$, either (1) $w_i = w_i' A_i w_i''$ and $w_{i+1} = w_i' x_i w_i''$, for some $w_i', w_i'' \in (N \cup T)^*$, or (2) $w_i = w_{i+1}$, A_i does *not* appear in w_i and the rule $A_i \rightarrow x_i$ appears in F. Thus, in case (2) a matrix can be applied even if some of its rules are not applicable, provided that these rules are listed in F. We remark that F consists of *occurrences* of rules in M, that is, if the same rule appears several times in the matrices, it is possible that only some of these occurrences are contained in F.

The language generated by a matrix grammar G is $L(G) = \{w \in T^* | S \Longrightarrow^* w\}$, where $\Longrightarrow^* w$ is the reflexive and transitive closure of $\Longrightarrow$. It is known (see Păun ([2002](#)) for details) that matrix grammars with appearance checking are universal.

Let $|x|$ denote the length of the string x. A matrix grammar with appearance checking $G = (N, T, S, M, F)$ is said to be in *binary normal form* if $N = N_1 \cup N_2 \cup \{S, \#\}$, with these sets mutually disjoint, and the matrices in M are of the forms:

1. $(S \rightarrow XA)$, with $X \in N_1$, $A \in N_2$;
2. $(X \rightarrow Y, A \rightarrow x)$, with $X, Y \in N_1$, $A \in N_2$, $x \in (N_2 \cup T)^*$, $|x| \leq 2$;
3. $(X \rightarrow Y, A \rightarrow \#)$, with $X, Y \in N_1$, $A \in N_2$;
4. $(X \rightarrow \lambda, A \rightarrow x)$, with $X \in N_1$, $A \in N_2$, $x \in T^*$, $|x| \leq 2$.

Moreover, there is only one matrix of type 1 and F consists exactly of all rules $A \rightarrow \#$ appearing in matrices of type 3. We remark that $\#$ is a trap symbol, namely once introduced it cannot be removed, and a matrix of type 4 is used only once, in the last step of a derivation.

For each matrix grammar (with or without appearance checking) there exists an equivalent matrix grammar in binary normal form. A matrix grammar with appearance checking in binary normal form is always given as $G = (N, T, S, M, F)$, with $N = N_1 \cup N_2 \cup \{S, \#\}$ and with $n + 1$ matrices in M, injectively labeled with $m_0, m_1,\ldots,m_n$. The matrix $m_0 : (S \rightarrow X_{init} A_{init})$ is the initial one, with X_{init} a given symbol from N_1 and A_{init} a given symbol from N_2; the next k matrices are without appearance checking rules, $m_i : (X \rightarrow \alpha, A \rightarrow x)$, with $1 \leq i \leq k$, where $X \in N_1$, $\alpha \in N_1 \cup \{\lambda\}$, $A \in N_2$, $x \in (N_2 \cup T)^*$, $|x| \leq 2$ (if $\alpha = \lambda$, then $x \in T^*$); the last $n - k$ matrices have rules to be applied in the appearance checking mode, $m_i : (X \rightarrow Y, A \rightarrow \#)$, with $k + 1 \leq i \leq n$, $X, Y \in N_1$, and $A \in N_2$.

We remark that in matrix grammars in binary normal form we can assume that all symbols $X \in N_1$ and $A \in N_2$ appear as the left-hand side of a rule from a matrix: otherwise, the derivation is blocked after introducing such symbols, hence we can remove these symbols and the matrices involving them.

The following theorem shows that Spatial P systems with *ME* objects and using only non–cooperating rules are universal. We denote by *PsRE* the family of Parikh images of all recursively enumerable languages.

Theorem 2 $PsSP_1(ncoo, me) = PsRE$.

Proof   It is enough to show that for a grammar G in binary normal form there is a Spatial P system Π_G with non-cooperating evolution rules and *ME* objects such that $Ps(\Pi_G) = \Psi_T(L(G))$. We build Π_G as a system with only a root membrane, whose geometry is depicted in Fig. 4. The non-mutually exclusive objects of Π_G are the symbols of G plus some control objects: $V = N \cup T \cup \{c_1, c_1', c_2, c_2', c_\#, c'_\#, \#\}$. All objects corresponding to grammar symbols will reside in position p_{home}. The other positions can be seen as control positions to simulate the application of the matrices of the grammar. The *ME* objects $E = \{e_0, e_1, e_2, e_3\}$ do not move, and they expire after the given time, i.e. e_i is canceled after i time steps, according to rules in Fig. 5. The other rules for the membrane are defined below.

The execution consists of a repetition of cycles of four time steps. Each cycle selects non-deterministically and then applies one of the matrices $m_1,\dots,m_n$. If during the cycle something goes wrong, i.e. either the application of the selected matrix is not possible or a matrix $(X \to Y, A \to \#)$ of type 3 is selected when a symbol A is present, then the trap symbol $\#$ is introduced yielding a non-terminating system, which corresponds to aborting the computation.

Let us define how matrices are mapped into rules. A matrix $m_i: (X \to Y, A \to x)$, $1 \leq i \leq k$, i.e. without appearance checking, is mapped into two rules:

$$X \to (Y)_{here}(e_3)_{p_1} \cdots (e_3)_{p_{i-1}}(e_3)_{p_{i+1}} \cdots (e_3)_{p_n}(c_1)_{p_c}(e_2)_{p_{c1}} \tag{1}$$

$$A \to (x)_{here}(e_3)_{p_A}(e_2)_{p_i}(c_2)_{p_c}(e_1)_{p_{c2}} \tag{2}$$

Note that, for readability, we omit the subscript *home* and use the message p_j to denote the vector δp from position p_{home} to position p_j. Similarly for the other positions. Rule (1) is used to select the matrix to be applied: since it puts a *ME* object in all positions p_j, with $j \neq i$, only one can be applied among the n available. Note that the occupied positions will be free exactly after 3 time steps (the object e_3 is used) respecting the duration of the cycle. A control object c_1 is placed in position p_c and position p_{c1} is occupied for two time steps (object e_2). Moreover, there is a rule $c_1 \to c_1'$ to let object c_1 pass from step 1 to step 2. Rule (2) cannot be applied at step 1 of the cycle because position p_A is not initially free (it contains e_1) and the rule is trying to put an *ME* object there. Thus, the rule can be applied only at the second step of the cycle. It transforms the non-terminal A and puts an *ME* object e_2 in position p_i, the only one still free after the first step. Note that other rules without appearance checking cannot be applied at the second step because the positions in which

Fig. 4 Initial step of the simulation

p_{home} X_{init} A_{init}			$\cdots$		$p_\#$	p_{c2}
p_A e_1	p_1	p_2	$\cdots$	p_n	p_c	p_{c1}

Fig. 5 Rules for e_i objects and trap symbol $\#$

$$e_3 \to e_2$$
$$e_2 \to e_1$$
$$e_1 \to e_0$$
$$e_0 \to \lambda$$
$$\# \to \#$$

they try to put the *ME* object is occupied. Note also that object e_3 is put in p_A, thus freeing the position only at the second step of the next cycle. A control object c_2 is put in position p_c and position p_{c2} is occupied for one time step.

A matrix m_i: $(X \rightarrow Y, A \rightarrow \#)$, $k + 1 \leq i \leq n$, i.e. with appearance checking, is also mapped into two rules:

$$X \rightarrow (Y)_{here}(e_3)_{p_1} \cdots (e_3)_{p_{i-1}}(e_3)_{p_{i+1}} \cdots (e_3)_{p_n}(c_\#)_{p_\#} \tag{1'}$$

$$A \rightarrow \#(e_3)_{p_A}(e_2)_{p_i} \tag{2'}$$

The differences in rule (1') w.r.t. rule (1) are that control object c_1 is not needed and that an object $c_\#$ is put in the control position $p_\#$. For this object we add rules $c_\# \rightarrow c'_\#$ and $c'_\# \rightarrow (e_2)_{p_A}$, i.e., we assure the correct continuation of the cycle if the trap symbol is not generated at the second step of the cycle. By rule (2'), if a symbol A is present in p_{home} at the second step of the cycle, after matrix i was selected, the trap symbol is generated and the computation aborted, as it would happen for the grammar. Otherwise, if there are no A symbols the cycle goes towards its end doing nothing and the computation will continue in the next cycle.

In the third step of the cycle we perform a double check, using both control objects c_1' and c_2, that all positions p_i, $1 \leq i \leq n$, are occupied. Otherwise, either one matrix was selected, but the second part was not applied due to a missing object A, or no symbol of N_1 was left (i.e. the state symbol was canceled), but some non-terminal symbol in N_2 remains. In both cases the computation has to be aborted. The rules are: $c_1' \rightarrow \#(e_0)_{p_1}, \ldots, c_1' \rightarrow \#(e_0)_{p_n}$ and $c_2 \rightarrow \#(e_0)_{p_1}, \ldots, c_2 \rightarrow \#(e_0)_{p_n}$.

In the fourth step of the cycle we simply cancel control objects in order to make the next cycle start in a consistent state. Rules are $c_1' \rightarrow (e_0)_{p_{c1}}$ and $c_2 \rightarrow (e_0)_{p_{c2}}$. Note that, since positions p_{c1} and p_{c2} are occupied in step 3, these rules can not be applied, i.e. the control objects are not canceled in step 3.

Finally, for each terminal in T there is a rule sending it out of the membrane. Since all terminals are produced in position p_{home}, the rule can be always applied.

4 An example of application

In this section we show a simple application of our model to a classical example: the evolution of "ring species" based on small changes between geographically contiguous populations (Irwin et al. 2001). A ring species is a species which expanded along two pathways around a geographic barrier, with the forms which gradually diverge along the pathways. The intermediate contiguous forms can interbreed but, when the terminal forms meet on the other side of the barrier, they have accumulated so many changes that they behave like different species. This process results in a complete ring of populations with a single species boundary. An example of a ring species is the Greenish warbler, *Phylloscopus trochiloides*. The Greenish warbler is a small insectivorous bird that breeds in forests over a range spanning much of the Palaearctic. The species consists of six subspecies, five of which form a ring around the Tibetan Plateau. Two of the subspecies coexisted without interbreeding in the Yenisey River valley of central Siberia, with gradual variation through the chain of populations to the south.

In the following we show a simple model of a species which expands around a barrier. The colonization of a new space can be associated with a small change in the genotype of

the moving population. Such small changes do not prevent the possibility for two contiguous populations to interbreed.

Each population is represented by its genotype: a string of three loci each of them having, as possible values (alleles), either 0 or 1. Two populations can interbreed if their genotypes differ in one position at most.

Figure 6a shows the initial situation. The environment is represented by a membrane of size 3×3, and the barrier is represented by an inner membrane of size 1×1. The initial population is located in position (1, 2), and it is composed of three individuals with 010 genotype (010^3). The *ME* object e states that a position is already colonized and it cannot be reached by a different population. The evolution rules are shown in Fig. 7, where $x, y, z \in \{0, 1\}$, an overlined symbol $\overline{x}$ represents the negation of x (as if 0, 1 represent the logical values *false* and *true*), and d denotes a direction $d \in \{N, S, E, W\}$.

Rules of type 1 simply state that individuals can survive, while rules of type 2 say that individuals can die. Type 3 rules describe the reproduction of two individuals in the same position, while rules of type 4–7 describe the reproduction of individuals of two contiguous populations. The contiguous populations, in order to have offspring can have either the same genotype or two genotypes differing in only one locus. Because we assume that all individuals in a population have the same genotype, an offspring is placed in the population of the parent from which it inherits its genotype.

Rules of type 8 describe the expansion of a population (actually of an individual) in a contiguous position which is not already colonized (the expansion is possible only if the *ME* object e can be placed in the target position). Rules of type 9–11 describe expansions associated with small changes in the genotype.

The system can evolve in many directions. Figure 6 shows a possible evolution of the system, starting from the initial state depicted in Fig. 6a. Suppose that, in the first step, rule 11 is applied to an object in position (1, 2), producing object 011 in (0, 2), while rule 1 is applied to the remaining two objects 010. The resulting state is depicted in Fig. 6b. The system can evolve to a state in which all the positions are colonized, such at the one shown
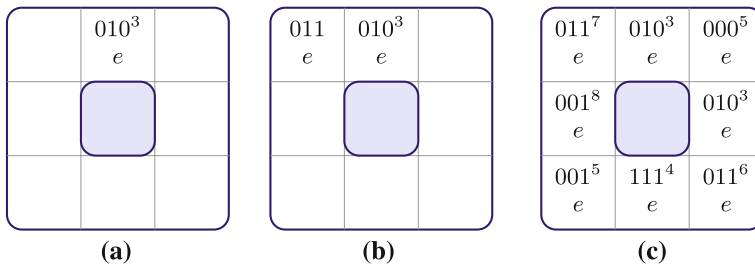

Fig. 6 A possible evolution of the system. **a** Initial population. **b** First step. **c** Final population

$$(1) \quad xyz \to xyz \qquad (2) \quad xyz \to \lambda \qquad (3) \quad xyz^2 \to xyz^3$$
$$(4) \quad xyz - xyz \to xyz^2 - xyz \qquad (5) \quad xyz - \overline{x}yz \to xyz^2 - \overline{x}yz$$
$$(6) \quad xyz - x\overline{y}z \to xyz^2 - x\overline{y}z \qquad (7) \quad xyz - xy\overline{z} \to xyz^2 - xy\overline{z}$$
$$(8) \quad xyz \to xyz_d \ e_d \qquad (9) \quad xyz \to \overline{x}yz_d \ e_d$$
$$(10) \quad xyz \to x\overline{y}z_d \ e_d \qquad (11) \quad xyz \to xy\overline{z}_d \ e_d$$

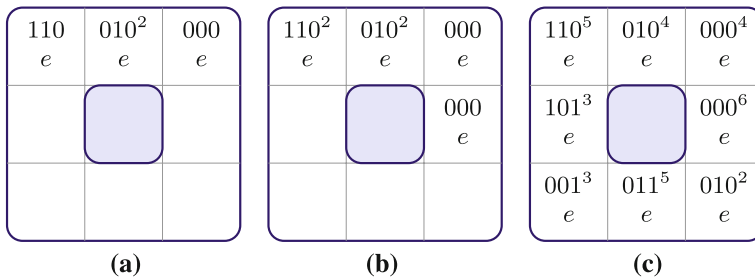**Fig. 7** Evolution rules for the model of ring species

Fig. 8 Another possible evolution of the system, with initial state as Fig. 6a. **a** First step. **b** Second step. **c** Final population

in Fig. 6c. In this case, it is easy to see that the expansion followed two pathways around the barrier. Each population can interbreed with the contiguous ones apart from the populations in positions (0, 0) and (1, 0). The two populations have accumulated so many changes that their genotypes are incompatible.

Figure 8 shows another possible evolution of the system, for the same starting state as before (Fig. 6a). Figure 8a shows a different possible state reached after the first step, corresponding to the application of rule 9 producing 110 in (0, 2), rule 10 producing 000 in (2, 2), and rule 1 deleting one object 010 from position (1, 2). After another step, the system reaches the state shown in Fig. 8b. In this case, by rule 8, species 000 colonizes position (2, 1); by rule 5, another object 110 is created in (0, 2); while the other 010 in (1, 2) is kept unmodified by the application of rule 1. Figure 8c shows a reachable state in which, as in the previous example, all position are colonized, and populations 110 in (0, 2), and 101 in (0, 1), cannot interbreed since their genotypes are too different.

5 Conclusions

We have presented Spatial P systems, an extension of P systems which provides an explicit representation of space inside membranes. The considered space is the discrete two-dimensional space $\mathbb{N}^2$. Evolution rules are associated with membranes, while objects are associated with positions inside membranes. Evolution rules are extended to allow objects to be moved to different positions. Objects belonging to the special kind of *mutually exclusive objects* are subjected to the constraint that no two objects of this kind can occupy the same position at the same time.

We have proved that Spatial P systems are universal even if only non-cooperating rules are allowed, and that universality is achieved due to the presence of mutually exclusive objects. We have also given an example of application of Spatial P systems for modeling the evolution of "ring species".

As future work, we plan to continue the study of Spatial P systems as a modeling tool for describing population dynamics in ecosystems. Moreover, in order to improve the usefulness of models, we plan to develop various extensions of the calculus. For example, a probabilistic or stochastic version would be able to describe more faithfully the behavior of systems than the non-deterministic semantics, and also to pave the way for quantitative simulations (see, for example, Cardona et al. (2009)). Another possible extension of Spatial P systems would be to three-dimensional space.

References

Barbuti R, Maggiolo-Schettini A, Milazzo P, Tini S (2008) Compositional semantics and behavioral equivalences for P Systems. Theor Comput Sci 395(1):77–100

Barbuti R, Maggiolo-Schettini A, Milazzo P, Pardini G (2009a) Spatial calculus of looping sequences. ENTCS 229(1):21–39, In: FBTC 2008

Barbuti R, Maggiolo-Schettini A, Milazzo P, Tini S (2009b) P systems with transport and diffusion membrane channels. Fundamenta Informaticae 93(1–3):17–31

Bartocci E, Di Berardini MR, Corradini F, Emanuela M, Tesei L (2009) A shape calculus for biological processes. In: Preproceedings of ICTCS 2009, Politecnico di Milano, pp 30–33

Besozzi D, Cazzaniga P, Pescini D, Mauri G (2008) Modelling Metapopulations with stochastic membrane systems. Biosystems 91(3):499–514

Bottoni P, Martín-Vide C, Păun G, Rozenberg G (2002) Membrane systems with promoters/inhibitors. Acta Informatica 38(10):695–720

Cardelli L, Gardner P (2009) Processes in space. Imperial College Technical Report DTR09-4

Cardona M, Colomer MA, Pérez-Jiménez MJ, Sanuy D, Margalida A (2009) Modeling ecosystems using P systems: the bearded vulture, a case study. In: Corne DW, Frisco P, Păun G, Rozenberg G, Salomaa A (eds) Membrane computing: 9th international workshop, Lecture notes in computer science, vol 5391, pp 137–156

Irwin DE, Irwin JH, Price TD (2001) Ring species as bridges between microevolution and speciation. Genetica 112–113(1):223–243

John M, Ewald R, Uhrmacher AM (2008) A spatial extension to the π-calculus. ENTCS 194(3):133–148, In: FBTC 2007

Manca V, Bianco L, Fontana F (2004) Evolution and oscillation in P systems: applications to biological phenomena. In: Mauri G, Păun G, Pérez-Jiménez MJ, Rozenberg G, Salomaa A (eds) Workshop on membrane computing, Springer, Lecture notes in computer science, vol 3365, pp 63–84

Neumann JV (1966) Theory of self-reproducing automata. University of Illinois Press, Champaign, IL

P Systems web page (2009) http://ppage.psystems.eu

Păun G (2000) Computing with membranes. J Comput Syst Sci 61(1):108–143

Păun G (2002) Membrane computing. An introduction. Springer, Berlin