# Weak Metrics on Configurations of a P System

**Andrés CORDÓN-FRANCO, Miguel A. GUTIÉRREZ-NARANJO,**
**Mario J. PÉREZ-JIMÉNEZ, Agustín RISCOS-NÚÑEZ**

Research Group on Natural Computing
Department of Computer Science and Artificial Intelligence
University of Sevilla
Avda. Reina Mercedes s/n, 41012 Sevilla, Spain
E-mail: {acordon,magutier,marper, ariscosn}@us.es

**Abstract.** The evolution of a P system generates a tree of computation potentially infinite where it is very difficult to set the degree of closeness between two configurations. The problem is specially hard if we want to quantify that proximity in order to make useful comparisons. In this paper we propose some weak metrics on configurations of a P system with a fixed structure of membranes and briefly discuss their advantages and drawbacks.

## 1 Introduction

In [2], a new model of computation within the framework of *Natural Computing* was introduced, called *P Systems*[1]. It starts from the assumption that the processes taking place in the compartmental structure of a living cell can be interpreted as computations.

Roughly speaking, a P system consists of a cell-like membrane structure, in the compartments of which one places multisets of objects which evolve according to given rules in a synchronous, parallel, and non-deterministic manner.

The *membrane structure* of a P system is a hierarchical arrangement of membranes embedded in a *skin* membrane, the one which separates the system from its *environment.* A membrane without any membrane inside is called *elementary.* Each membrane defines a *region* (the closed space delimited by a membrane and by the membranes immediately inside it).

The *membrane structure* of a P system is used to enclose *computing cells* in order to make them independent computing units. Also, a membrane serves as a communication channel between a given cell and other cells *adjacent* to it. The objects can pass through membranes and the membranes can be dissolved, divided, or created.

A *configuration* is the instantaneous description of the current membrane structure and the multisets of objects associated with the membranes. In each time unit a transformation of a configuration of the system takes place by applying the rules of each region in a non-deterministic and maximally parallel manner. In this way, one gets transitions between the configurations of the system and a sequence of transitions is called a computation.

In certain circumstances, we need to know how different two configurations of a P system are. They can be different in many senses and the problem turns extremely hard

[1] A layman-oriented introduction can be found in [3] and further bibliography at [5].

when the configurations do not correspond to the same P system. If we have three configurations $\mathcal{C}_1$, $\mathcal{C}_2$ and $\mathcal{C}_3$, is $\mathcal{C}_1$ *more different* from $\mathcal{C}_2$ than from $\mathcal{C}_3$? Is it possible to quantify this degree of similarity and give it an algebraic treatment? In this paper we study the differences among configurations and we propose a way to quantify the degree of difference.

We offer some solutions to the problem of finding appropriate metrics for P systems. We focus our attention only on finding metrics on the configurations of a P system with a fixed membrane structure. This involves a fixed alphabet and a fixed set of rules. In this case, two configurations may only differ in the multisets associated with the membranes. This difference can be measured between configurations not necessarily in the same branch of a computation or in the same step.

We propose two models of defining metrics. The first one is based on the distance between regions. This gives us a very natural way of defining the distance according to the difference between multisets, but it does not consider the set of rules of the P system. The second model is based on the dependency graph associated with the rules of a P system and is based on the shortest paths in this directed graph.

The paper is organized as follows. Section 2 recalls some ideas about metrics and weak metrics in a general setup. In Section 3 two metrics on configurations of P systems based on the different multisets of regions are presented. In Section 4 a new concept in P system theory is defined: the dependency graph of a P system. This dependency graph is used in Section 5 to define a weak metric on configurations. The paper finish with an example (Section 6) and some final remarks.

## 2 Metrics

Sometimes it is necessary to reduce the relation between two objects to a number in order to make comparisons and also perform algebraic operations with them. This number is used to be called a *distance* and it allows us to differentiate between pairs of objects in a simple way. So, for instance, we say that two towns $A$ and $B$ are *closer* than the towns $C$ and $D$ if the length of the shortest path from $A$ to $B$ (i.e., the *distance* which separates them) is less than the length of the shortest path from $C$ to $D$.

Analogously, that distance can measure the time elapsed between two events, the amount of necessary combustible to cover a route, or the number of pieces which are left to complete a puzzle.

In this way, if the distance from $A$ to $B$ is less than the distance from $A$ to $C$, we think that the relation between $A$ and $B$ is narrower than the relation between $A$ and $C$.

Given a set $X$, if we associate to every pair of elements $(x, y) \in X \times X$ its distance, we get a mapping $d : X \times X \to \mathbb{R}$. But, obviously, not every mapping $d : X \times X \to \mathbb{R}$ is a distance. What properties does a mapping $d : X \times X \to \mathbb{R}$ have to satisfy in order to be a distance? It is clear that the criterion has to be weak enough to be common to the different distances of geometric intuition and strong enough to settle a solid theory which allows us to deal with the concept of distance in abstract situations.

It was M. Fréchet in his Ph.D. dissertation [1] who stated that it was sufficient that the mapping satisfied

- $(\forall x, y \in X)\, d(x, y) = 0 \Leftrightarrow x = y$,
- $(\forall x, y \in X)\, d(x, y) = d(y, x)$ *(the condition of symmetry)*,

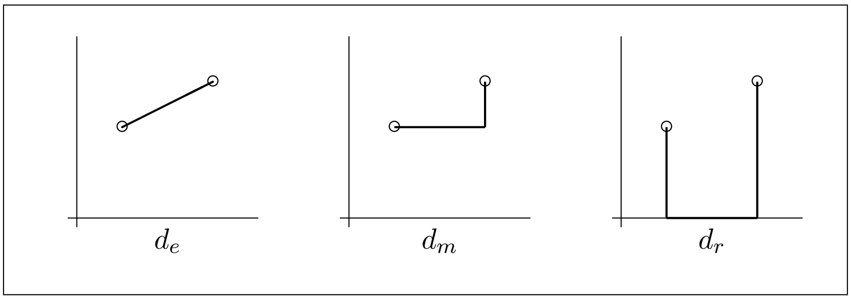

Figure 1: Several metrics

- $(\forall x, y, z \in X)\, d(x,z) \leq d(x,y) + d(y,z)$ *(the triangle inequality)*,

to develop a theory of metric spaces and, since then, they have been considered the basic pillars of the theory.

As examples of distances based on the geometric intuition, we can cite three well-known distances in $\mathbb{R}^2$. Let $A = (x_1, y_1)$ and $B = (x_2, y_2)$ be two points of the plane.

- **Euclidean distance** $(d_e)$**:** Given two points $A$ and $B$ in $\mathbb{R}^2$, the distance $d_e$ measures the length of the segment which joins $A$ and $B$, i.e., of the shortest path from $A$ to $B$, assuming that there are no obstacles in the plane

$$d(A,B) = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}.$$

- **Manhattan distance** $(d_m)$**:** In this case, we also measure the shortest path from $A$ to $B$, but in contrast to the Euclidean distance, in the Manhattan distance we suppose that the moves can only be horizontal or vertical ones, simulating the movement of a vehicle through streets with a grid form.

$$d_m(A,B) = |x_1 - x_2| + |y_1 - y_2|.$$

- **Distance of the rainforest** $(d_r)$**:** An example, perhaps less known, of distance in $\mathbb{R}^2$ is this *distance of the rainforest* which also measures the length of the shortest path between two points. It receives this name because it stands in $\mathbb{R}^2$ for the situation of a tribe in a rainforest with a river in $y = 0$. The people of the tribe, to reach the water, have done breaches perpendicular to the river. Due to the thick rainforest, if someone wants to go from $A$ to $B$, the only path is by the breaches or on the bank.

$$d_r(A,B) = \begin{cases} |y_1 - y_2| & \text{if } x_1 = x_2, \\ |y_1| + |y_2| + |x_1 - x_2| & \text{if } x_1 \neq x_2. \end{cases}$$

At this point, it makes sense to wonder why it is necessary to define several distances on the same set. The answer is clear. Every distance is adapted to an earlier structure in the set. If we are only interested in endowing the set with a mapping which satisfies the Fréchet's conditions and we do not consider any other previous relation among the members of the set, we can always consider the *discrete distance*

$$d_d(A,B) = \begin{cases} 0 & \text{if } A = B \\ 1 & \text{if } A \neq B \end{cases}$$

which satisfies the Fréchet's conditions to be a distance, but it would hardly have a practical usefulness.

A different situation is settled when the pre–existing relation between the objects is not symmetric. The number of kilometers which separate a town $A$ on the coast from a another at the top of a mountain is independent of the direction of the journey. But if our idea of *distance* is the number of calories spent by a cyclist from a town to the other, then the condition of symmetry is lost in our definition of *distance*. A more extreme case is the passage of time. When January 1st 2005 arrives, we will have to wait for 365 days to January 1st 2006, but when January 1st 2005 arrives, it will not make sense to wait for the arrival of the year 2004.

Another real life situation in which Fréchet's conditions must be weakened occurs when we go shopping. A good pointer to estimate the difference between two items can be the price, but this is not exactly a distance: We can find two distinct items with the same price.

# 3 Metrics on Regions

In this section we propose two metrics on configurations based on the different multiset of the regions in each configuration. For that, we consider a P system with alphabet $\mathcal{L}$ and a fixed membrane structure, i.e., dissolution or duplication of membranes are not allowed. Since the membrane structure does not change along different configurations, we also consider that we can identify the same membranes in different configurations[2]. The metrics are based on the difference between the multisets.

Firstly, we define the distance between two regions as the cardinality of the symmetrical difference of their associated multisets. We will use this definition to measure the distance between two occurrences of the same membrane in two different configurations.

**Definition 3.1** *Let us consider a region $R$ and $\mathcal{L}$ the alphabet of the P system. The multiset associated with the region $R$, $\mathcal{M}_{\mathcal{R}}$, can be characterized as the mapping $\mathcal{M}_{\mathcal{R}} : \mathcal{L} \to \mathbb{N}$. The distance $d_R$ between the regions $R_1$ and $R_2$ is defined as*

$$d_R(R_1, R_2) = \sum_{x \in \mathcal{L}} |\mathcal{M}_{R_1}(x) - \mathcal{M}_{R_2}(x)|,$$

*where $|.|$ is the function absolute value.*

**Theorem 3.1** *$d_R$ is a (weak) metric between regions.*

## 3.1 Plain Metric

With the help of the distance $d_R$ between regions, the definition of the distance between configurations is pretty natural. As set of regions, the difference between configurations is the sum of the differences between their regions.

Let $\Pi$ be a P system in which the structure of membranes does not change during the computation. In this P system, two configurations $C^1$ and $C^2$ only differ on the multisets associated to the regions, and therefore, if $R^i_j$ is the region delimited by the membrane

[2] This can be done by considering labels, positions or some type of enumeration.

$m_j$ in the configuration $C^i$ (with $j \in \{1, \dots, k\}$ and $i \in \{1, 2\}$), then we can consider the additive distance based on the distance between regions

$$d^+(C_1, C_2) = \sum_{1 \le j \le k} d_R(R_j^1, R_j^2).$$

**Theorem 3.2** *$d^+$ is a (weak) metric between configurations.*

### 3.2 Biased Metric

The (weak) metric defined above does not consider the tree structure of the set of membranes. We assume that every membrane has the same importance to measure the closeness between two configurations, so every element has the same weight regardless in which membrane it occurs.

Nevertheless, sometimes we can have another point of view. Sometimes, we design P systems where the inner membranes work as parallel devices, sending out to the skin the output of each computation. From this point of view, the objects in the skin, i.e., the result of the parallel computation, are more important than the objects in each inner membrane, since these objects only have a *local* function.

The (weak) metric defined below follows this idea. Firstly we define a recursive distance $d_B$ among regions: If $\{m_{j1}, \dots, m_{js_j}\}$ are the children of the membrane $m_j$, then we define

$$d_B(R_j^1, R_j^2) = d_R(R_j^1, R_j^2) + C_j \cdot \sum_{i=1}^{s_j} d_R(R_{ji}^1, R_{ji}^2),$$

where $C_j$ is a constant of bias associated with the membrane $m_j$. As a particular case of this definition, we have the situation in which $m_j$ is a leave, i.e., $m_j$ has no children; then

$$d_B(R_j^1, R_j^2) = d^+(R_j^1, R_j^2).$$

Finally, to define a mapping in order to quantify the closeness between configurations, we only have to consider the distance between their skins[3]. If $m_s$ is the skin membrane, then

$$d_B(C^1, C^2) = d_B(R_s^1, R_s^2).$$

Note that if all the constants of bias are equal to 1, then $d_r(C^1, C^2) = d^+(C^1, C^2)$.

**Theorem 3.3** *$d_B$ is a (weak) metric between configurations.*

## 4 Dependency Graphs

In this section we explore a new variant of metrics between configurations based on the dependence among elements of the alphabet with respect to the set of rules of the P system. To this aim, we consider the rules of a P system with a new representation and we define the concept of confluence of computations in a more general way than the standard one.

The rules of a non-cooperative P system, without dissolution nor division fit into the following schema

$$(e_0, \mu_1) \to (e_1, \mu_2), (e_2, \mu_2), \dots, (e_n, \mu_2)$$

[3] For the sake of simplicity, we keep the same notation $d_B$ also for configurations.

which can be interpreted as follows: *The occurrence of the element* $e_0$ *in the membrane* $\mu_1$ *triggers the rule and provokes the apparition of the multiset* $e_1 e_2 \dots e_n$ *into the membrane* $\mu_2$. Obviously, if $\mu_1 = \mu_2$, then we have an *evolution* rule, if $n = 1$ and $\mu_1$ is a father of $\mu_2$, then we have a send-in communication rule, and if $\mu_1$ is a child of $\mu_2$, then we have a send-out communication rule. The pair $(e_0, \mu_1)$ is the *left side* of the rule and the multiset of pairs $(e_1, \mu_2), (e_2, \mu_2), \dots, (e_n, \mu_2)$ is the *right side* of the rule.

Next, we define the graph of dependence of a P system based on this new representation of the rules.

**Definition 4.1** *The* dependency graph *of a P system* $\Pi$ *is a pair* $G_\Pi = \langle V_\Pi, E_\Pi \rangle$ *such that* $V_\Pi$ *is the set of all the pairs* $(e, \mu)$ *where e is an element of the language and* $\mu$ *is a membrane and* $E_\Pi$ *is the set of all the ordered pairs of elements of* $V_\Pi$, $\langle (e_1, \mu_1), (e_2, \mu_2) \rangle$ *such that* $(e_1, \mu_1)$ *is the left side of a rule and* $(e_2, \mu_2)$ belongs *to the right side of a rule.*

We illustrate this definition with an example. Let us consider the next toy P system $\Pi$, with alphabet $\Gamma = \{a, b, c, d, z\}$, membrane structure $[_s [_e ]_e ]_s$ and set of rules:

**Rule 1:** $[_e a]_e \rightarrow a[_e ]_e$
**Rule 2:** $[_s a]_s \rightarrow a[_s ]_s$
**Rule 3:** $[_e a]_e \rightarrow [_e bz]_e$
**Rule 4:** $[_e b]_e \rightarrow c[_e ]_e$
**Rule 5:** $[_s c]_s \rightarrow [_s dz]_s$
**Rule 6:** $[_s d]_s \rightarrow a[_s ]_s$

In order to define the dependency graph, we have to consider the set of membranes $\{e, s\}$, and since the elements can be sent out of the system (rules **2** and **6**), we will consider a new region *outside* as a place where the elements can stand, so the set of regions becomes $\{e, s, outside\}$. Finally, with the new representation, the rules can be written as follows:

**Rule 1:** $(a, e) \rightarrow (a, s)$
**Rule 2:** $(a, s) \rightarrow (a, outside)$
**Rule 3:** $(a, e) \rightarrow (b, e), (z, e)$
**Rule 4:** $(b, e) \rightarrow (c, s)$
**Rule 5:** $(c, s) \rightarrow (d, s), (z, s)$
**Rule 6:** $(d, s) \rightarrow (a, outside)$

Therefore, the dependency graph of $\Pi$, $G_\Pi = \langle V_\Pi, E_\Pi \rangle$ is defined by the following sets:

$$V_\Pi = \left\{ \begin{array}{lllll} (a, e) & (b, e) & (c, e) & (d, e) & (z, e) \\ (a, s) & (b, s) & (c, s) & (d, s) & (z, s) \\ (a, outside) & (b, outside) & (c, outside) & (d, outside) & (z, outside) \end{array} \right\}$$

The set of vertices $V_\Pi$ has 15 elements, but 7 of them are isolated vertices: only 8 vertices occur in some edge (see Figure 2).

$$E_\Pi = \left\{ \begin{array}{l} \langle (a, e), (b, e) \rangle, \langle (a, e), (z, e) \rangle, \langle (a, e), (a, s) \rangle, \\ \langle (a, s), (a, outside) \rangle, \\ \langle (b, e), (c, s) \rangle, \\ \langle (c, s), (d, s) \rangle, \langle (c, s), (z, s) \rangle, \\ \langle (d, s), (a, outside) \rangle \end{array} \right\}$$

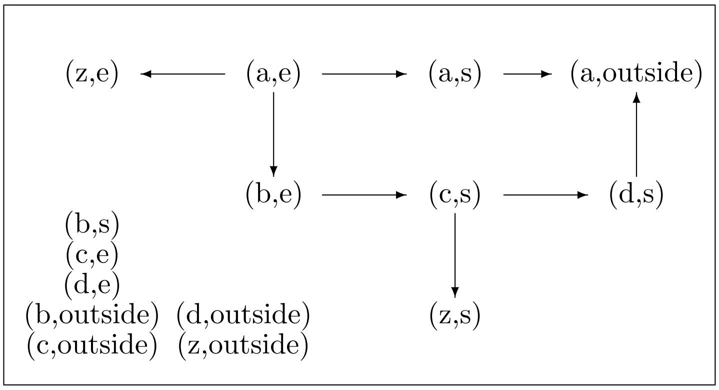

Figure 2: The dependency graph

Note that the dependency graph only depends on the membrane structure and the set of rules of the P system and not on the elements of the membranes at the initial moment.

The example will help us to introduce a new definition of *confluence*, more general than the usual one. We know the structure of membranes and the set of rules of our toy P system. At the beginning we will consider the skin empty and the inner membrane containing only copies of the element $a$. The intended computation sends out of the system, in several steps, as many copies of $a$ as introduced at the beginning in the membrane $e$. Figure 3 shows the computation tree of the P system when two copies of $a$ are introduced in the membrane $e$. The system is non-deterministic. In the first step the rules **1** and **3** can be triggered. This produces three different branches. The three branches end and the final configuration is different in all the cases, but always in the end of the computation the P system sends out as many copies of $a$ as introduced in the inner membrane. As a computational device, we can think that the P system works, as every branch returns the correct number of $a$. This leads us to define a more general definition of confluence than the classical one[4]: the confluence with respect to a property.

**Definition 4.2** *A P system is called* confluent with respect to a property *if all the branches of the computation tree end, and all the final configurations satisfy the property.*

With this definition, we can say that the P system in the example (with $a^2$ in the membrane $e$ at the beginning) is confluent with respect to the property: *The number of objects a in the environment in the final configuration is two.*

Note that the three branches in the example end with a *correct* configuration, but the *number of steps* is not the same in all them. This suggests us a way to compute how far from each other two configurations are.

Before giving the definition of the weak metric on configurations, we need some previous definitions.

**Definition 4.3** *Given a P system, an* L-configuration *of the P system is a multiset of pairs $(s, m)$ where $s$ is an element of the alphabet and $m$ is a membrane of the P system. We will say that an L-configuration is* total *when for all symbol $s$ of the alphabet and for all membrane $m$, the multiplicity of $s$ in $m$ is the same as the multiplicity of the pair*

[4] See, for example, [4].

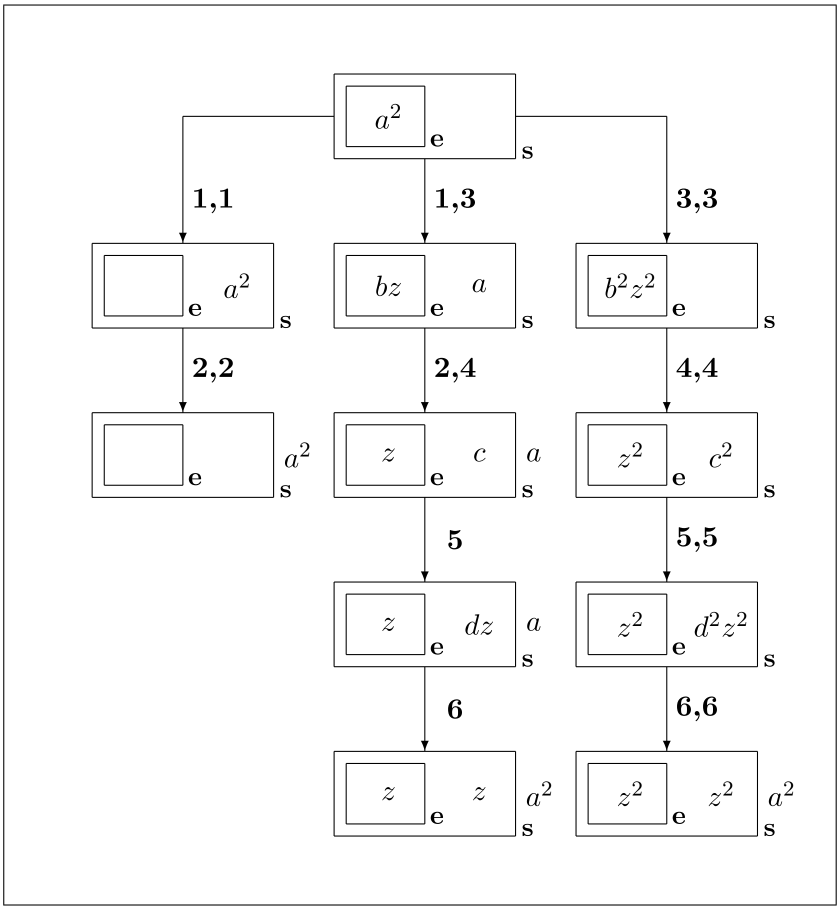

Figure 3: The computation tree

*$(s, m)$ in the configuration. Any proper submultiset of a total L-configuration is a* partial *L-configuration.*

The distance between two nodes of the dependency graph is defined in the natural way:

**Definition 4.4** *Given a directed graph (as a dependency graph), a path from two vertices $a$ and $b$ is a finite sequence $v_0, v_1, \ldots, v_n$ of vertices such that $v_0 = a$, $v_n = b$ and for all $i \in \{0, \ldots, n-1\}$, $(v_i, v_{i+1})$ is an edge of the graph. The sequence of vertices with an unique vertex is also considered a path. The length of a path is the number of vertices of the sequence minus one.*

*Given a vertex $v$, we define the* set of initial vertices of $v$, *$I_v$ as the set of all the vertex $a$ of the graph such that there exists a path from $a$ to $v$. Given a set of vertices $S$, we define the* set of initial vertices of $S$, *$I_S$ as the set of all the vertex $a$ of the graph such that there exists a vertex $v$ in $S$ and a path from $a$ to $v$.*

**Definition 4.5** *Given a P system $\Pi$ and its dependency graph $G_\Pi$, the* distance *between two nodes $v_1$ and $v_2$ of $G_\Pi$ is the length of the shortest path that connect $v_1$ and $v_2$ and infinite if there is no path from $v_1$ to $v_2$.*

# 5 Weak Metrics Based on the Dependency Graph

## 5.1 First Approach

In non-deterministic P systems, given a configurations there (potentially) exist several configurations which can be reached. If the P system is confluentin the classical sense, from the point of view of correctness, it is not important the branch we follow, because the final result is the same, but from a computational point of view, the cost measured as the number of steps in the computation can be different, so it can be interesting to define some kind of measure of how far a configuration is from the final configuration.

Next, let us consider a total L-configuration $\mathcal{C}$, which represents an intermediate step of the computation, and a partial L-configuration $\mathcal{F}$, which represents the property of a possible final L-configuration. How can we measure the closeness between them? One way is by using the minimum number of steps of computation between them in the natural way.

Firstly, we consider an element $b \in \mathcal{F}$. The element $b$ has to be reachable from the elements in $\mathcal{C}$, and we are interested in the shortest path, so we consider

$$\min_{a \in \mathcal{C} \cap I_b} d(a, b),$$

where $\mathcal{C} \cap I_{\mathcal{F}}$ is the intersection of the L-configuration $\mathcal{C}$ with the set of initial vertices of $\mathcal{F}$, in other words, is the multiset of all the elements $a$ of $\mathcal{C}$ such that such that there exists a path from $a$ to $b$. If this set is empty, the minimum is infinite. Finally, to compute the distance, we have to consider the longest of these shortest paths.

**Definition 5.1** *Given two L-configurations $\mathcal{C}$ and $\mathcal{F}$, the quasi-metric from $\mathcal{C}$ to $\mathcal{F}$ is defined as*

$$d(\mathcal{C}, \mathcal{F}) = \max_{b \in \mathcal{F}} \{ \quad \min_{a \in \mathcal{C} \cap I_b} d(a, b) \quad \}.$$

**Theorem 5.1** *$d$ is a (weak) metric between L-configurations.*

The metric $d_h$ induced by this quasi-metric,

$$d_h(\mathcal{C}_1, \mathcal{C}_2) = \max\{d(\mathcal{C}_1, \mathcal{C}_2), d(\mathcal{C}_2, \mathcal{C}_1)\},$$

is the Hausdorff metric on the L-configurations.

# 6 Example

In our example, in the first step of the computation three new configurations are possible (see Figure 3). They can be represented as the following multisets

$$\begin{aligned}
\mathcal{C}_1 &= \{(a, s), (a, s)\}, \\
\mathcal{C}_2 &= \{(b, e), (z, e), (a, s)\}, \\
\mathcal{C}_3 &= \{(b, e), (b, e), (z, e), (z, e)\},
\end{aligned}$$

and the partial L-configuration which characterizes all the final configurations is

$$\mathcal{F} = \{(a, outside), (a, outside)\};$$

following the definitions, we get

$$I_{(a,outside)} = \{(a,e),(a,s),(b,e),(c,s),(d,s),(a,outside)\}$$

and $d(\mathcal{C}_1,\mathcal{F}) = 1$, $d(\mathcal{C}_3,\mathcal{F}) = 3$ and $d(\mathcal{C}_2,\mathcal{F}) = ...$

## 6.1 Second Approach

Unfortunately, $d(\mathcal{C}_2,\mathcal{F}) = \min\{1,3\} = 1$, and this is not the intuitive idea that we want to formalize. The problem arises from the fact that we are dealing with multisets, not with sets, so we must adapt the definitions. We begin with a new definition of dependency graph.

**Definition 6.1** *The* dependency graph *of a P system* $\Pi$ *is a labeled graph* $G_\Pi = \langle V_\Pi, E_\Pi\rangle$ *such that* $V_\Pi$ *is the set of all the pairs* $(e,\mu)$ *where* $e$ *is an element of the language and* $\mu$ *is a membrane and* $E_\Pi$ *is the set of all the labeled pairs of elements of* $V_\Pi$*,* $(e_1,\mu_1) \xrightarrow{n_i} (e_i,\mu_2)$ *such that* $(e_1,\mu_1)$ *is the left side of a rule and* $(e_i,\mu_2)^{n_i}$ belongs *to the right side of a rule.*

For example, the set of rules

| | |
|---|---|
| **Rule 1:** | $[_e a]_e \to a[_e]_e$ |
| **Rule 2:** | $[_s a]_s \to a[_s]_s$ |
| **Rule 3:** | $[_e a]_e \to [_e bz^2]_e$ |
| **Rule 4:** | $[_e b]_e \to c[_e]_e$ |
| **Rule 5:** | $[_s c]_s \to [_s dz^2]_s$ |
| **Rule 6:** | $[_s d]_s \to a[_s]_s$ |

produces a graph whose edges are

$$E_\Pi = \left\{ \begin{array}{l} (a,e) \xrightarrow{\mathbf{1}} (b,e), (a,e) \xrightarrow{\mathbf{2}} (z,e), (a,e) \xrightarrow{\mathbf{1}} (a,s), \\ (a,s) \xrightarrow{\mathbf{1}} (a, outside), \\ (b,e) \xrightarrow{\mathbf{1}} (c,s), \\ (c,s) \xrightarrow{\mathbf{1}} (d,s), (c,s) \xrightarrow{\mathbf{2}} (z,s), \\ (d,s) \xrightarrow{\mathbf{1}} (a, outside) \end{array} \right\}$$

Figure 4 shows the labeled graph.

The problem now is to define the mapping $d(\mathcal{C},\mathcal{F})$ among L-configurations. Given an element $b \in \mathcal{F}$ with multiplicity $m$, we have to find $n$ elements in $\mathcal{C}$, $\{e_1,\ldots,e_n\}$ $(n \le m)$, with multiplicities $\{m_1,\ldots,m_n\}$, respectively, such that for every $i \in \{1,\ldots,n\}$ there exists a path from $e_i$ to $b$ and

$$\sum_{1\le i\le n} r_i \cdot c_i \ge m,$$

where $c_i$ is the coeficient obtained by multiplying the labels of the shortest path from $e_i$ to $b$; in other words, the $r_i$ occurrences of the element $e_i$ in $\mathcal{C}$ provoke the apparition of $r_i \cdot c_i$ occurrences of $b$ in $\mathcal{F}$.

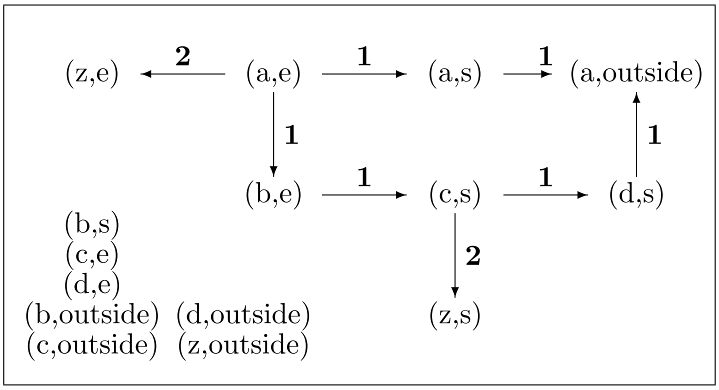

Figure 4: The labeled dependence graph

The set $\mathcal{M} = \{e_1, \ldots, e_r\}$ verifying these properties is probably not unique, and for each $b \in \mathcal{F}$ we have to consider the set

$$\mathcal{K}_b = \left\{ \begin{array}{lll} \{e_1^1 & \cdots & e_{l_1}^1\} \\ \{e_1^2 & \cdots & e_{l_2}^2\} \\ \cdots & & \cdots \\ \{e_1^{s_b} & \cdots & e_{l_{s_b}}^{s_b}\} \end{array} \right\}$$

of all the sets $\mathcal{M}_i = \{e_1^i, \ldots, e_{l_i}^i\}$ with $i \in \{1, \ldots, s_b\}$ verifying the properties.

With the help of this set $\mathcal{K}_b$ we can define a mapping which quantifies the closeness among L-configurations $\mathcal{C}$ and $\mathcal{F}$ by considering the multiplicities of the elements.

**Definition 6.2** *With the previous notations, the (weak) metric between the L-configurations $\mathcal{C}$ and $\mathcal{F}$ is defined as*

$$D(\mathcal{C}, \mathcal{F}) = \max_{b \in \mathcal{F}} \min_{1 \le i \le s_b} \max_{1 \le j \le l_i} d(e_j^i, b).$$

With the L-configurations of the example

$$\begin{aligned} \mathcal{C}_1 &= \{(a,s), (a,s)\}, \\ \mathcal{C}_2 &= \{(b,e), (z,e), (a,s)\}, \\ \mathcal{C}_3 &= \{(b,e), (b,e), (z,e), (z,e), \} \end{aligned}$$

and

$$\mathcal{F} = \{(a, outside), (a, outside)\},$$

we have the following.

- To compute $D(\mathcal{C}_1, \mathcal{F})$, we only have to consider $\mathcal{K}_{(a,outside)} = \{\mathcal{M}_1\}$ with $\mathcal{M}_1 = \{(a,s)\}$, i.e., $s_{(a,outside)} = 1$ and $l_1 = 1$, therefore

$$D(\mathcal{C}_1, \mathcal{F}) = d((a,s), (a, outside)) = 1.$$

- To compute $D(\mathcal{C}_3, \mathcal{F})$, the case is analogous to the previous one, we only have to consider $\mathcal{K}_{(a,outside)} = \{\mathcal{M}_1\}$ with, in this case, $\mathcal{M}_1 = \{(b,e)\}$, i.e., $s_{(a,outside)} = 1$ and $l_1 = 1$, therefore

$$D(\mathcal{C}_1, \mathcal{F}) = d((b,e), (a, outside)) = 3.$$

- The most interesting case is $D(\mathcal{C}_3, \mathcal{F})$, where $\mathcal{K}_{(a,outside)} = \{\mathcal{M}_1\}$ with $\mathcal{M}_1 = \{(b,e),(a,s)\}$ and

$$D(\mathcal{C}_2, \mathcal{F}) = \max\{d((a,s),(a,outside)), d((b,e),(a,outside)) = 3.$$

# 7 Final Remarks

P systems are computational devices typically non–deterministic. Usually, the evolution of a P system generates a tree of computation which is potentially infinite. Even in the case of finite trees, the amount of information is too big to be handled efficiently with present-day computers. If we are dealing with a confluent recognizer P system, i.e., a P system where every computation halts and all the computations output the same answer (*yes* or *no*), then it is not important, from the correctness point of view, which computation is actually chosen. In every choice point one branch can be chosen randomly and at the end of the evolution we will know if the P system accepts or rejects the input.

But from a computational point of view, the situation is quite different. Even if the answer is the same, the computational cost can be pretty different from a branch to another. In order to obtain efficiently an answer, (evolution) rules need to be complemented by another component, usually called *strategy* or *search plan*, which is responsible for the *control* of the rules.

In this framework, it would be very useful to have a tool to help us in the decision of choosing a *short* branch in the computation tree. In order to reach this target, we have presented several approaches to quantify the closeness between configurations via mappings such that they verify totally or partially Fréchet's conditions.

The ideal situation would be to have a mapping $h^*$ such that it associates to every node a number which indicates the length of the shortest path from this node to a leaf, and of course, with a tractable computational cost when it is used as input of the algorithm $A^*$.

In a more realistic situation, we look for a mapping $h$ (*heuristic function*) being an *estimation* of $h^*$, and such that it associates to every node $e$ of the tree a number $h(e)$ verifying $h(e) \leq h^*(e)$ with a low computational cost. Then $h(e)$ must be a lower bound of the number of transition steps needed to reach a halting configuration from $e$. Such a mapping $h$ will guide the search, using the algorithm $A^*$. The additional cost of this algorithm depends on the *kindness of the estimation* of the heuristic function $h$.

We hope to return to these topics in a forthcoming paper.

**Acknowledgement.** Support this research through the project TIC2002-04220-C03-01 of the Ministerio de Ciencia y Tecnología of Spain, cofinanced by FEDER funds, is gratefully acknowledged.

# References

[1] Fréchet, M.: Sur quelques points du calcul fonctionnel. *Rendiconti del Circolo Matematico di Palermo*, **22** (1906).

[2] Păun, G.: Computing with membranes, *Journal of Computer and System Sciences*, **61**, 1) (2000), 108–143.

[3] Pérez-Jiménez, M.J. and Păun, G.: Recent computing models inspired from biology: DNA and membrane computing, *Theoria*, **18**, 46 (2003), 72–84.

[4] Pérez-Jiménez, M.J., Romero-Jiménez, A., and Sancho-Caparrini, F.: Complexity classes in cellular computing with membranes, *Natural Computing*, **2**, 3 (2000), 265–285.

[5] `http://psystems.disco.unimib.it/`