

Homological Groups, Spanning Forests and Membrane Computing

Ainhoa Berciano^{1,2}, Daniel Díaz-Pernil¹

¹Research Group on Computational Topology and Applied Mathematics
Department of Applied Mathematics
University of Seville

²Departamento de Didáctica de la Matemática y de las Ciencias Experimentales
University of the Basque Country
`ainhoa.berciano@ehu.es, sbdani@us.es`

Abstract. In this paper we present a new way to determine the geometrical objects associated to the Homology groups of a 2D- digital binary image. In fact, using Membrane Computing with techniques of spanning forests, we are able to define a parallel algorithm which calculates non only the Betti numbers, but also the geometric objects which generate these holes.

1 Introduction

Homology theory is a branch of algebraic topology that attempts to distinguish between spaces by constructing algebraic invariants that reflect the connectivity properties of the space. The field has its origins in the work of Poincaré. Homology groups (related to the *different* n -dimensional holes, connected components, tunnels, cavities, etc.) are invariants from Algebraic Topology which are frequently used in Digital Image Analysis and Structural Pattern Recognition. In some sense, it reflects the topological nature of the object in terms of the number and features of its holes.

In this paper we introduce a solution of the problem of the identification of geometric objects associated to generators of the homology groups of a 2D-binary image using spanning tree forest techniques in the framework of *Membrane Computing*¹. This new computational paradigm is based on the assumption that the processes taking place within the compartmental structure of a living cell can be interpreted as computations. The computational devices in Membrane Computing are called *P systems*. Roughly speaking, a tissue-like P system is a distributed and parallel computing device structured like a tissue, consisting of a non-hierarchical arrangement of membranes which delimit compartments where various chemicals (we call them *objects*) evolve according to interchange rules.

¹ We refer to [3] for basic information in this area, to [4] for a comprehensive presentation and the web site [5] for the up-to-date information.

2 Calculating Homology Groups

In this section, firstly we will establish some required concepts and later on we will introduce the problem we want to study. For more details see [1, 2].

We use tissue-like P systems (cells are nodes in a general graph) and the application of the rules are regulated by *promoters* and *priorities*. These promoters have a clear biological inspiration. The rule is applied if the reactants are present, but it is also necessary the presence of the promoter in the corresponding cell. The promoter is not *consumed* nor *produced* by the application of the rule, but if it is not present in the cell, the rule cannot be applied.

Definition 1. *A tissue-like P system with promoters and priorities of degree $q \geq 1$ is a tuple of the form $\Pi = (\Gamma, \Sigma, \mathcal{E}, w_1, \dots, w_q, \mathcal{R}, Pri, i_{in}, i_0)$ where*

1. Γ is a finite alphabet, whose symbols will be called objects, $\Sigma \subseteq \Gamma$ is the input alphabet, $\mathcal{E} \subseteq \Gamma$ is a finite alphabet representing the set of the objects in the environment available in an arbitrary large amount of copies;
2. w_1, \dots, w_q are strings over Γ representing the multisets of objects associated with the cells in the initial configuration;
3. \mathcal{R} is a finite set of rules of the following form:

$$(pro | i, u/v, j), \text{ for } 0 \leq i \neq j \leq q, pro, u, v \in \Gamma^*$$

4. Pri is a finite set of relations $R_i > R_j$, where R_i and R_j are rules from \mathcal{R} . It means that if R_i and R_j can be applied, then the application of R_i has priority on R_j .
5. $i_{in}, i_0 \in \{0, 1, 2, \dots, q\}$ ($i_0 \neq 0$) denotes the input and output region, respectively.

The rule $(pro | i, u/v, j)$ can be applied over two cells (or a cell and the environment) i and j such that u (contained in cell i) is traded against v (contained in cell j). The rule is applied if the objects of the promoter pro are present in i . The promoter is not modified by the application of the rule. If the promoter is empty, we will write $(i, u/v, j)$ instead of $(\emptyset | i, u/v, j)$.

Rules are used as usual in the framework of membrane computing, that is, in a maximally parallel way (a universal clock is considered).

Next, we give some notions about Digital Image.

An image on $P = \{1, \dots, n\} \times \{1, \dots, n\}$ with colors in the finite set \mathcal{C} is a mapping $I : P \rightarrow \mathcal{C}$. As usual, such image can be written as a set of pairs $((i, j), a)$ where $i, j \in \{1, \dots, n\}$ and $a = I(i, j)$. In this paper we deal with black and white pixels, so for the sake of simplicity we will write b_{ij} for a black pixel at position (i, j) and w_{ij} for a white one. In the solution described below, we will also use a more complex object. We associate one flag (k, l) to each colored pixel a_{ij} . The pair pixel-flag will be written as $(a_{ij}, (k, l))$. Finally, let us remark that in this paper we will use the 4-adjacency for black pixels and the 8-adjacency for white pixels. The rationale for this decision is based on technical reasons due to the Membrane Computing model used.

The *Homology Groups of Binary 2D Image (HGB2I) Problem* can be settled as follows: Given a binary 2D digital image, calculate the number of black connected components and the representative curves of the holes of these components.

Given an input image $I : P \rightarrow \mathcal{C}$, the input data for this problem is given by the objects $\{A_{ij} : (i, j) \in P \wedge a = I(i, j)\}$, so we should construct a family of tissue-like P systems with promoters and priorities $\Pi = \{\Pi(n) : n \in \mathbb{N}\}$ to give an efficient solution to HGB2I problem. Given an input data I^* with size n^2 , our problem will be solved by system $\Pi(n)$ according with the following stages:

1. *Input Stage*: When the input objects, A_{ij} (with $A = B \vee W$), arrive to cell 1 we change these objects by objects a_{ij} and a'_{ij} . In the next step, we send objects a_{ij} to cell 2 and objects a'_{ij} to cell 3. The process occurs in cell 2, and it is dedicated to obtain a representative object of each black connected components that appear in our image.
2. *Finding Roots Process*: It begins when objects a_{ij} arrive to cell 2.
 - (a) *Label Allocation Stage*: Cell 2 trades objects b_{ij} against others with the form $(b_{ij}, (i, j))$ with the environment. White objects are not transformed.
 - (b) *Label Conversion Stage*: We can compare the black adjacent pixels by using promoter, and we trade the label of the greatest pixel against the label of the other pixel, as we can see in the following rules:

$$(i, (b_{ij}, (i', j')))(b_{kl}, (k', l')) / (b_{ij}, (i', j'))(b_{kl}, (i', j')) A_{i'j'k'l', j}$$

where (i, j) and (k, l) are adjacent pixels. Moreover, we can see a new object ($A_{i'j'k'l'}$ in this case) arriving to cell i . It is a promoter and it is used to codify two labels have been compared and, they are connected. So, one of them is changed by the other one.

- (c) *Identifying Roots (of connected component) Stage*: In the step $n + 2$, the object z_{n+2} arrives to the cell 1 due to the counter. It is used by the system as a promoter, and the objects with the form $(b_{ij}, (i, j))$ are marked as 0_{ij} representing the root of a black connected component. The P system has used $n + 2$ steps to obtain the roots of black connected components of an n^2 image.
3. *Construction (0 – 1 spanning tree) Stage*: In this step, for each root (which the P system identified in the previous stage), new edges are added to each spanning tree of vertices and edges. We will eliminate the cycles inside of the black connected components and order the nodes of the generated trees. This order gives us a opposite direction to contract the tree below. To identify the edges e_{ij} of our spanning trees we change the object e by natural numbers. So, we will obtain an order in our trees. Therefore, we mark the edges incident with leaves of the trees. When we finish, (using a counter) we send all objects to the cell 4.
4. *Construction (1 – 2 Spanning tree) Stage*: In this step, we consider a new graph (in fact, it is a forest), taking the faces as vertices and objects g as

edges. We add leaves of our forest objects w_{ij} with i, j even. We consider this objects like empty faces. The faces adjacent to empty faces add a number 1 $((c, 1))$ and eliminate the object g incident with it and the empty faces. Faces adjacent (by g) to objects (c, k) with $k \in \mathbb{N}$ pass to be $(c, k + 1)$.

5. *Contraction Stage:* We send elements of the $0 - 1$ spanning tree to cell 5 and the elements of the $1 - 2$ spanning tree to cell 6. In each one of this cell we work or the same manner. We eliminate the leaves of the trees together with the edge incident with them. In the $0 - 1$ spanning trees we marked the leaves, and in the second type of trees the leaves are the objects $(c, 1)$.
6. *Output Stage:* We send the objects b that remain in cell 5 at the end of the previous stage and they are sent to the empty cell. We do the same with the objects w in cell 6. The number of objects b and w give us the number of black connected component and the holes, respectively.

Conclusions

Even though the computation of the homology groups of a 2D-binary image has been studied in depth by several authors in the last decades, we have showed how is possible to compute Betti Numbers and geometrical properties of an image in another completely different framework. Furthermore, this new approach opens the door to continue investigating in the computation of higher algebraic structures associated to 2D-image using Membrane Computing.

Acknowledgement

DDP acknowledges the support of the projects TIN2008-04487-E and TIN-2009-13192 of the Ministerio de Ciencia e Innovación of Spain and the support of the Project of Excellence with *Investigador de Reconocida Valía* of the Junta de Andalucía, grant P08-TIC-04200. AB acknowledges the support of the research projects MTM2006-03722 and EHU09/04.

References

1. Christinal, H.A., Díaz-Pernil, D., Real, P.: P systems and computational algebraic topology. *Mathematical and Computer Modelling* **52**(11-12), 1982 – 1996 (2010).
2. Díaz-Pernil, D., Gutiérrez-Naranjo, M.A., Real, P., Sánchez-Canales, V.: Computing homology groups in binary 2D imagery by tissue-like P systems. *Romanian Journal of Information Science and Technology (ROMJIST)* **13**(2), 141–152 (2010)
3. Păun, Gh.: *Membrane Computing. An Introduction*. Springer-Verlag, Berlin, Germany (2002)
4. Păun, Gh., Rozenberg, G., Salomaa, A. (eds.): *The Oxford Handbook of Membrane Computing*. Oxford University Press (2010).
5. P system web page. <http://ppage.psystems.eu>