Math. Struct. in Comp. Science (2009), *vol.* 19, *pp.* 1141–1160.
doi:10.1017/S0960129509990156

(Tissue) P systems with cell polarity

DANIELA BESOZZI[†], NADIA BUSI, PAOLO CAZZANIGA[‡], CLAUDIO FERRETTI[‡], ALBERTO LEPORATI[‡], GIANCARLO MAURI[‡], DARIO PESCINI[‡] and CLAUDIO ZANDRON[‡§]

[†]*Università degli Studi di Milano, Dipartimento di Informatica e Comunicazione, Via Comelico 39, 20135 Milano, Italy*
Email: besozzi@dico.unimi.it
[‡]*Università degli Studi di Milano-Bicocca, Dipartimento di Informatica, Sistemistica e Comunicazione, Viale Sarca 336, 20126 Milano, Italy*
Email: {cazzaniga;ferretti;leporati;mauri;pescini;zandron}@disco.unimib.it

Received 20 Mar 2008; revised 3 February 2009

In memory of Nadia Busi

We consider the structure of the intestinal epithelial tissue and of cell–cell junctions as the biological model inspiring a new class of P systems. First we define the concept of cell polarity, a formal property derived from epithelial cells, which present morphologically and functionally distinct regions of the plasma membrane. Then we show two preliminary results for this new model of computation: on the theoretical side, we show that P systems with cell polarity are computationally (Turing) complete; on the modelling side, we show that the transepithelial movement of glucose from the intestinal lumen into the blood can be described by such a formal system. Finally, we define tissue P systems with cell polarity, where each cell has fixed connections to the neighbouring cells and to the environment, according to both the cell polarity and specific cell–cell junctions.

1. Introduction

The living cell can be seen as a sophisticated information processing device, and in recent years a number of computation models inspired by the structure and function of living cells have been proposed. Among them, we will consider membrane systems, also known as P systems, which were introduced in Păun (2000). The basic model consists of a hierarchical structure composed of several membranes, which are embedded into a main membrane called the *skin*. Membranes divide the Euclidean space into *regions*, which contain some *objects* (represented by symbols over an alphabet) and *evolution rules*. Using these rules, the objects may evolve and/or move from one region to a neighbouring one.

[§] This work was supported by 'Models of natural computing and applications', FIAR 2007, University of Milano-Bicocca, Italy

The rules are applied in a non-deterministic and maximally parallel way: all the objects that may evolve are forced to evolve. A *computation* starts from an initial configuration of the system and halts when no evolution rule can be applied. The result of a computation is the multiset of objects contained within an *output membrane* or sent outside the skin, to the outer environment. In the rest of this paper we assume a familiarity with the basic notions and terminology underlying P systems. For a systematic introduction to P systems, see Păun (2002), and for the latest information, see the P Systems Web Page: `http://ppage.psystems.eu`.

In this paper we define new classes of P systems by taking inspiration from the structure of epithelial cells and the intestinal epithelial tissue. In particular, we introduce the novel concept of *cell polarity*, which is a structural property that many living cells present and consists of a peculiar intracellular organisation with morphologically and functionally distinct regions of the plasma membrane. We stress the fact that we do not use the term 'polarity' here to mean the presence of electrical charges on the membrane, that is, the 'membrane polarisation' as usually found in the P systems literature, but, rather, the *orientation* that each cell presents with respect to the surrounding cells and the external environment. We then extend the structure of a single-cell polarised P system to introduce *tissue P systems with cell polarity*, where each cell has fixed connections – by means of specific cell–cell junctions – to the neighbouring cells in the tissue and to the surrounding environment, according to its polarity. In our models, we will consider specific types of communication and evolution rules, which are needed to import (respectively, export) objects from (respectively, to) the external environment, to move objects within a single cell, and to exchange objects between adjacent cells. For this reason, we briefly recall some notions and computational aspects of symport/antiport rules in P systems, as well as tissue P systems, which will also be considered in the rest of our work.

With regard to the communication of objects, the biochemical idea of transporting 'pairs of molecules' was first considered in Păun (2002) by introducing the notion of P systems with symport/antiport rules. In such systems, multisets of objects – here denoted by x, y (which can be of any size, but not empty) – are moved across the membranes by means of rules of the form (x, in) and (x, out) (*symport rules*, where multiset x is moved across the membrane in one direction, either inward or outward, respectively), and $(x, out; y, in)$ (*antiport rules*, where multiset x is moved outward, and multiset y is simultaneously moved inward). Computing by communication turns out to be computationally complete: Păun *et al.* (2002) showed that we can generate all Turing computable sets of numbers using just symport and antiport rules. However, in order to reach Turing completeness, some 'infinity' must be present, and in P systems with symport/antiport this is provided in the form of an infinite supply of objects taken from an external environment (in our definition of (tissue) P systems with cell polarity, we will also need to consider the occurrence of infinite copies of symbols in the surroundings of the cells). Several subsequent papers have been devoted to improving this result with respect to both the number of membranes and the size of the symport/antiport rules: see Rogozhin *et al.* (2006) for a survey.

In an attempt to go from cell-like to tissue-like architectures, Martín-Vide *et al.* (2002) defined *tissue P systems*, where cells are placed in the nodes of a (directed) graph, and

https://doi.org/10.1017/S0960129509990156 Published online by Cambridge University Press

objects are communicated along the edges of the graph. This model has been further elaborated, for example in Freund *et al.* (2005) and Păun *et al.* (2002), with recent results covering both theoretical properties (Alhazov *et al.* 2006) and applications (Oswald 2007). The few variants of tissue P systems considered in the literature essentially differ in the mechanisms used to communicate objects between cells. For instance, particular sets of communication rules can be assigned to the edges of the graph that defines the structure of the tissue in order to model the existence of communication channels between the cells (Păun *et al.* 2002; Freund *et al.* 2005). There are also evolution–communication tissue P systems (adopting the terminology introduced in Cavaliere (2003)), where the objects produced by particular transformations occurring within the cells are non-deterministically propagated from one place to another (Martín-Vide *et al.* 2003; Bernardini and Gheorghe 2005).

The power of communication has also been studied in tissue P systems. In particular, purely communicative tissue P systems with symport/antiport were investigated in Păun (2002) by showing completeness results for systems using rules with different sizes and different structures for the underlying graph. More recently, Alhazov *et al.* (2005) proved that tissue P systems with symport/antiport rules of a minimal size (that is, rules of the form (a, in), (a, out) or $(a, out; b, in)$, with a, b objects from a given alphabet) are computationally complete, and that two cells suffice. Verlan *et al.* (2006) considered tissue P systems with *conditional uniport*, meaning that every application of a communication rule moves one object in a certain direction by, possibly, using another one as an activator, which is left unchanged in the place where it is. Verlan *et al.* (2006) showed that this purely communicative model of computation is also computationally complete as it is able to simulate deterministic register machines using 24 cells. This result can be compared in an interesting way with Bernardini and Gheorghe (2005), where an evolution–communication model was considered. This means that, besides conditional uniport, multiset rewriting rules can be used to modify the objects placed inside the cells. The result given in Bernardini and Gheorghe (2005) shows that this model of computation achieves computational completeness through the use of only two cells and non-cooperative multiset rewriting rules.

The structure of the tissue P systems we are proposing here differs from all other membrane tissue structures, since in this case the cells can be linked together by means of specific connections, or communication channels, that are characterised by dynamical states – which can change from open to closed and *vice versa*. In this way, any pair of connected cells within the tissue can either exchange objects, or, otherwise, has to wait for a communication channel to (possibly) open. This is achieved by considering appropriate 'control multisets' for each connection between adjacent cells.

The paper is structured as follows. We begin in the next section by providing an overview of biological tissues (in particular, epithelia), presenting the concept of cell polarity, and giving a description of intercellular junctions. Taking inspiration from the biology of epithelial cells, in Section 3 we define P systems with cell polarity. Then, we report two preliminary results for single-cell P systems with cell polarity: we prove that they are computationally (Turing) complete, since they are able to simulate (albeit in a non-deterministic way) Random Access Machines (Section 4); and we show that they

https://doi.org/10.1017/S0960129509990156 Published online by Cambridge University Press

can be properly used to model a cellular process, *viz*. the transcellular glucose transport in intestinal cells (Section 5). In Section 6 we define tissue P systems with cell polarity by extending the definition of single-cell polarised P systems to include (dynamically changing) communication channels between adjacent cells. Finally, in Section 7 we provide some concluding remarks and suggest some directions for future research.

2. Epithelial tissues and cell polarity

In any organism, specific and differentiated types of cells are assembled together to form structural complexes, called *tissues*, which differ in both morphology and functionality. In vertebrates, the main types are the epithelial, connective, muscular, lymphoid and nervous tissues. Among these, the first two represent the extreme examples of tissue organisation. Connective tissue is characterised by a plentiful *extracellular matrix* (a complex network of macromolecules having a supporting function for cells and other tissues), within which cells are sparsely distributed. By contrast, in epithelial tissue the extracellular matrix is sparse, and cells are tightly bound to each other by cell–cell adhesions, forming sheets called epithelia.

Epithelia can be made up of one or more layers of cells, whose major functions are: to form the lining of internal cavities (for example, lungs, intestine, stomach) and of the free surfaces (skin) of the organism; to protect against mechanical, chemical and physical damages; to receive stimuli; and so on. Epithelial cells have differentiated functions, corresponding to their position in the organism. For instance, the absorption of the products of digestion (such as glucose and amino acids) is mediated by the epithelial cells that line the intestinal lumen, while the acidification of the stomach lumen is carried out by the parietal cells in the gastric lining.

Within each layer of an epithelial tissue, the epithelial cells are connected to one another by means of specialised regions of the plasma membrane, called *cell junctions*. There are several types of junctions, whose major functions are: to form barriers (to prevent the passage of water and solutes across the opposite sides of the tissue); to supply strength and rigidity to the tissue; to anchor the epithelium to the underlying connective tissue; and so on.

In order to accomplish all of these functions, epithelial cells have a peculiar characteristic, *viz*. a morpho-functional orientation, called *polarity*, which allows one to distinguish between the two opposite sides of the cell from both the structural and molecular points of view. The region of the cell facing the lumen is called the *apical* face, while the opposite region is called the *basal* face (Fig. 1); the other sides of the cell, which are connected to the neighbouring cells, are called the *lateral* faces. For each of these regions, the plasma membrane presents several corresponding functional specialisations, which are described in the following paragraphs. Moreover, the polarity can also be reflected within the cell in the placement of organelles (the nucleus, Golgi apparatus, and so on) towards the apical or basal face.

The specialisations of the apical face consist of extraflexions or invaginations of the plasma membrane, such as microvilli, cilia or flagella, which can enhance the functionality of the cell. For instance, in the (single-layered) epithelium of the small intestine, each cell

https://doi.org/10.1017/S0960129509990156 Published online by Cambridge University Press

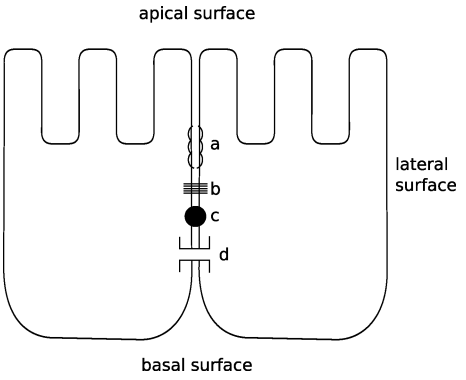

Fig. 1. Polarity and specialisations of the apical and basolateral faces of an epithelial cell: (a) tight junction, (b) adherens junction, (c) desmosome, (d) gap junction.

has thousands of membrane projections (called *microvilli*, see Figure 1), which increase the surface area of the plasma membrane – and the number of transport proteins it can contain – thus raising the absorptive capacity of the whole epithelium.

The specialisations of the basolateral face of epithelial cells consist of junctional systems, which are used: to connect neighbouring cells to one another; to coordinate their functioning; to anchor the tissue to the underlying extracellular matrix (also called the *basal lamina*); and to control the movement of ions and small molecules between adjacent cells. Cell–cell junctions can be classified as *adherent* or *communicating junctions* (see Figure 1). Adherent junctions form a mechanical connection between adjacent cells, and can be further divided into occluding and anchoring junctions, while communicating junctions allow the passage of small molecules between connected cells. We will now give a brief description of the principal types of cell junction.

Tight junctions are occluding junctions that are located just below the apical surface and seal the plasma membranes of neighbouring cells, thus creating an impermeable barrier against the diffusion of water-soluble substances between the opposite faces of the epithelium. For instance, tight junctions prevent digestion products contained within the intestinal lumen passing through the interstices between epithelial cells and directly entering the bloodstream. Tight junctions completely encircle each epithelial cell, and constitute a sort of 'belt' around it and thus create a very close connection between the cell and all its neighbours. Moreover, these barriers prevent the migration of membrane proteins and lipids between the apical and basolateral faces of the epithelial cell: the restriction of specific proteins to well-defined regions of the plasma membrane determines a non-homogeneous distribution of these molecules and is essential for the proper functioning of the cell.

Adherent anchoring junctions are found immediately below tight junctions. They serve to attach each cell mechanically to its neighbours (at their lateral faces), and also to the basal lamina (at the basal face). Anchoring junctions occur in different forms: *adherens junctions* and *desmosomes* are, respectively, belt-like and button-like structures of contact between cells that confer mechanical strength to the tissue.

https://doi.org/10.1017/S0960129509990156 Published online by Cambridge University Press

Finally, *gap junctions* are communicating channels, which are found below adherent junctions and allow the movement of small molecules directly between the cytosolic spaces of neighbouring cells. In this way, the cells connected by gap junctions can share their metabolites, and can thus be coordinated by this chemical and electrical coupling (for example, for the synchronised contractions of heart muscle cells or for gene regulation, in differentiation and embryogenesis, and so on). Gap junctions are not always open, but flip between the open and closed states in response to specific changes occurring within the cell. Moreover, they are not permanent structures, but can be synthesised by the cell whenever there is a need for functional communication between neighbouring cells.

3. P systems with cell polarity

In this section we provide a formalisation of a single-cell P system with polarity. In Section 6 we will extend this definition to consider tissue systems made up of polarised cells.

In the following, we assume that each polarised cell (either as a separate computing unit, or as part of a tissue) reflects the biological orientation of epithelial cells, as described in Section 2, in two main ways: on the one hand, the polarised cell looks 'externally' onto two distinct and non-contiguous regions of the environment, called the *apical* and *basal environments*; on the other hand, the 'internal' structure of the cell is built in such a way that only specified elements (which we will generically call *nodes*) can have a direct interaction with either the apical or basal environments, but not with both at the same time. We will also consider another type of internal element, whose role is to create communication between the apical and basal elements. The rationale behind this is that the apical and basal elements can be seen as those portions of the plasma membrane, or other intracellular structures, that keep an epithelial cell in contact with the extracellular (luminal and basal) regions, while the internal elements informally represent the cytoplasmic structures that constitute the intermediate (intracellular) space between the two opposite poles of the cell. Accordingly, we will not consider the possibility of any direct 'communication' between the nodes residing at the apical and basal sides of the cell.

Formally, a *P system with cell polarity* is a structure of the form

$$\Pi = (\Sigma, A, B, C, E_A, E_B, conn)$$

where

— Σ is a finite alphabet of objects.

— $A = \{A_1, A_2, \ldots, A_{k_1}\}$, for some $k_1 \in \mathbb{N}$, is the set of *apical nodes*. Each apical node is defined as

$$A_i = (\omega_{A_i}, R_{A_i}), \ \ 1 \leqslant i \leqslant k_1,$$

where ω_{A_i} is the initial multiset over Σ contained inside node A_i, and R_{A_i} is the set of rules of node A_i.

— $B = \{B_1, B_2, \ldots, B_{k_2}\}$, for some $k_2 \in \mathbb{N}$, is the set of *basal nodes*. Each basal node is defined as

$$B_i = (\omega_{B_i}, R_{B_i}), \ \ 1 \leqslant i \leqslant k_2,$$

https://doi.org/10.1017/S0960129509990156 Published online by Cambridge University Press

where ω_{B_i} is the initial multiset over Σ contained inside node B_i, and R_{B_i} is the set of rules of node B_i.

— $C = \{C_1, C_2, \ldots, C_{k_3}\}$, for some $k_3 \in \mathbb{N}$, is the set of the *internal nodes*. Each internal node is defined as

$$C_i = (\omega_{C_i}, R_{C_i}), \ 1 \leqslant i \leqslant k_3,$$

where ω_{C_i} is the initial multiset over Σ contained inside node C_i, and R_{C_i} is the set of rules of node C_i.

— E_A, called the *apical environment*, is a region containing an (up to) infinite number of objects from Σ. The apical environment can only be accessed through the nodes in the set A.

— E_B, called the *basal environment*, is a region containing an (up to) infinite number of objects from Σ. The basal environment can only be accessed through the nodes in the set B.

— $conn \subseteq C \times (A \cup B \cup C)$ is the set of all existing *connections* between any internal node of the cell and any other node of the cell, either apical, basal or internal.

The rules in R_{A_i} (respectively, R_{B_i}) can be of the following four types:

1 Symport rule: (u, out) or (u, in), for some (non-empty) multiset u over Σ. When a rule of the form (u, out) is applied, the multiset of objects u is sent to the apical (respectively, basal) environment. On the other hand, when a rule of the form (u, in) is applied, the multiset of objects u is taken from the apical (respectively, basal) environment. If $|u| = 1$, then the rule is said to be a uniport rule.

2 Antiport rule: $(u, out;\ v, in)$, for some u, v multisets over Σ. When a rule of this type is applied, the multiset of objects u is sent from the node to the apical (respectively, basal) environment while, at the same time, the multiset of objects v is taken from the apical (respectively, basal) environment and placed into the node.

3 Communication rule: (u, go), for some u multiset over Σ. When a rule of this type is applied, the multiset of objects u is sent from the node to an internal node in the set C, chosen non-deterministically from amongst all internal nodes connected to A_i (respectively, B_i) according to the connections defined in the set *conn*.

4 Evolution rule: $(a \to b)$, for some $a, b \in \Sigma$. When a rule of this type is applied, the single object a is transformed into the (not necessarily distinct) single object b, which remains within the node. Evolution rules are non-cooperative and conservative, meaning that the number of objects in the system cannot be increased or decreased by means of such rules; objects can only be modified by them.

Since we assume that internal nodes cannot communicate directly with the external (apical or basal) environments, the rules appearing in the sets R_{C_i} can only be of types 3 and 4. In particular, if a communication rule (u, go) is applied within an internal node C_i, then the multiset u is sent from C_i to another node (internal, apical or basal), chosen non-deterministically from amongst all nodes connected to C_i by means of a connection in the set *conn*.

Note that, following the standard notation used for writing biochemical reactions, we choose to denote the multisets appearing in any (symport, antiport, or communication)

https://doi.org/10.1017/S0960129509990156 Published online by Cambridge University Press

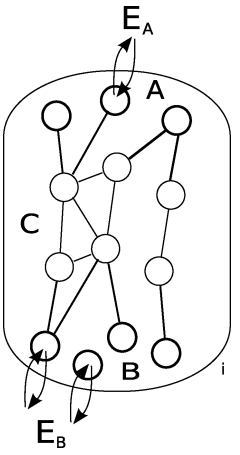

Fig. 2. Structure of a P system with cell polarity.

rule as $u = \alpha_1 a_1, \alpha_2 a_2, \ldots, \alpha_n a_n$, where $a_1, a_2, \ldots, a_n$ are distinct symbols from Σ, and $\alpha_1, \ldots, \alpha_n \in \mathbb{N}$ are their respective multiplicities in u.

Figure 2 shows the structure of a polarised cell with apical, basal and internal nodes, their interconnections, and the apical and basal environments. We tacitly assume that all nodes are contained within a unique membrane, which is not explicitly included in the formal definition of Π, since it is uniquely determined and has no explicit role in computations. By convention, we say that the *orientation* of a polarised cell Π is directed from the apical to the basal environment, and this is reflected in the internal structure of the cell as the (ordered) series of the apical, internal and basal nodes. According to this interpretation, we denote the orientation of Π as $or(\Pi) = (E_A \rightsquigarrow E_B)$.

The apical and basal environments, which surround the opposite faces of the cell, are in general assumed to contain an infinite number of objects over the same alphabet. When investigating the computational power of P systems with cell polarity, anyway, we will assume that the basal environment is initially empty. In fact, we can imagine that, again reflecting the chosen orientation of the cell, the apical environment corresponds to the 'input region', while the basal environment corresponds to the 'output region' (where we look for objects at the end of the computation); hence, we need to avoid the situation in which any objects appear in E_B at the beginning of the computation, which might falsify its result.

In a P system with cell polarity, the communication of objects occurs as follows. Objects can be exchanged between the environments and the cell only by means of the apical and basal nodes, respectively. In particular, environmental objects can only enter an apical (basal) node when a symport or antiport rule is applied within that node. On the other hand, communication rules in the apical (basal) nodes can only be used to exchange objects between an apical (basal) node and an internal node, whenever the two of them are linked through a connection belonging to the set *conn*.

Communication rules also differ from symport rules as they imply a non-deterministic choice of the target node to which the objects will be sent. In other words, when a communication rule is applied within an apical (or basal) node, the objects specified by

https://doi.org/10.1017/S0960129509990156 Published online by Cambridge University Press

the rule can be sent to *any* other internal node that shares a connection with the apical (or basal) node where the rule is applied. If a communication rule is applied within an internal node, the objects can be sent to *any* apical, basal or internal node that shares a connection with the internal node where the rule is applied. In particular, if an internal node is connected to itself, that is, if there exists a connection $(C_i, C_i) \in conn$, for some $i = 1, \ldots, k_3$, then the objects specified by the communication rule can (according to a non-deterministic choice) remain within the same node.

As we will see in Section 6, the same type of communication rules used for a P system with cell polarity can also be used to communicate objects between two adjacent cells in a tissue connected by means of 'gap junction' connections.

A *configuration* $\mathscr{C}_t$ of a P system with cell polarity Π at step t, $t \geqslant 0$, is defined as

$$\mathscr{C}_t = \{\omega_{A_1}, \ldots, \omega_{A_{k_1}}, \omega_{B_1}, \ldots, \omega_{B_{k_2}}, \omega_{C_1}, \ldots, \omega_{C_{k_3}}\},$$

where $\omega_{A_p}, \omega_{B_q}, \omega_{C_r}$, $1 \leqslant p \leqslant k_1, 1 \leqslant q \leqslant k_2, 1 \leqslant r \leqslant k_3$, are the multisets over Σ contained at step t in the corresponding apical, basal or internal node of Π, respectively.

A *computation* of a P system with cell polarity is a sequence of configurations $(\mathscr{C}_0, \mathscr{C}_1, \mathscr{C}_2, \ldots)$ starting from the initial configuration of Π. A *transition* $\mathscr{C}_t \rightarrow \mathscr{C}_{t+1}$ between two consecutive configurations, at time steps $t, t+1$, $t \geqslant 0$, is determined as follows. All apical, internal and basal nodes are processed in parallel. Within each node, all applicable communication and evolution rules are applied in a maximally parallel and non-deterministic way, while every applicable symport and antiport rule can be applied only once during each computation step (in this case, if the same multiset can be assigned to competing rules, then one of them is chosen non-deterministically). The rationale behind this sequential application of symport and antiport rules is that they mimic the presence of specific transport proteins occurring on the apical and basal faces of the cell: each protein carrying out a symport or antiport action can only move a limited and precise number of molecules at a time, therefore we do not consider the possibility of applying each symport or antiport rule an unlimited number of times in each step of the computation. Note that in this way we also avoid the possibility of generating illegal configurations, where infinite copies of objects (imported from the apical and basal environments through the maximally parallel application of symport and antiport rules) would occur within the cell nodes. On the other hand, communication rules between internal and apical (basal) nodes – and, similarly, evolution rules – can be applied in parallel, since they do not formally represent any specific molecular structures within the cell, but just mimic the free diffusion processes of molecules within the cytoplasm.

The computation halts when no further rule can be applied within any node of the system. In this case, the output of the system is given by the multiset of symbols from (a specified subset of) Σ found in the basal environment in the halting configuration.

4. Computational power of P systems with cell polarity

In this section we show that (single-cell) P systems with cell polarity are Turing complete. In particular, we show how to simulate Random Access Machines (RAMs) (Shepherdson and Sturgis 1963), which are a well-known Turing powerful formalism.

https://doi.org/10.1017/S0960129509990156 Published online by Cambridge University Press

4.1. *Random Access Machines*

RAMs are a computational model based on finite programs acting on a finite set of registers. More precisely, a RAM R is composed of the registers $r_1, \ldots, r_n$, which can hold arbitrary large natural numbers, together with a sequence of indexed instructions $(1 : I_1), \ldots, (m : I_m)$. Minsky (1967) showed that the following two instructions are sufficient to compute any partially recursive function:

— $(i : Inc(r_j))$: add 1 to the contents of register r_j and go to the next instruction;
— $(i : DecJump(r_j, s))$: if the contents of the register r_j is not zero, then decrease it by 1 and go to the next instruction, otherwise jump to the instruction s.

The computation starts from the first instruction, with all registers set to zero, and it continues by executing the instructions in sequence, unless a jump instruction is encountered. The execution stops when an instruction, whose number is higher than the length of the program, is reached. For the sake of brevity, we consider RAMs that satisfy the following constraint: if the RAM has m instructions, then all the jumps to addresses higher than m are jumps to the address $m + 1$. This constraint is not restrictive, because for any RAM not satisfying the constraint it is possible to construct an equivalent RAM (that is, a RAM computing the same function) that satisfies it.

A state of a RAM is modelled by $(i, c_1, \ldots, c_n)$, where i is the program counter indicating the next instruction to be executed, and $c_1, \ldots, c_n$ are the current contents of the registers $r_1, \ldots, r_n$, respectively. We use the notation $(i, c_1, \ldots, c_n) \rightarrow_R (i', c'_1, \ldots, c'_n)$ to denote the fact that the state of the RAM R changes from $(i, c_1, \ldots, c_n)$ to $(i', c'_1, \ldots, c'_n)$ as a consequence of the execution of the ith instruction. A state $(i, c_1, \ldots, c_n)$ is *final* if the program counter i is strictly greater than the number of instructions m. We say that a RAM R *halts* if its computation reaches a final state. The output of a halting computation is the contents of register r_1 in the final state. Non-halting computations produce no output.

4.2. *Simulating RAMs by means of P systems with cell polarity*

We are now going to show how to simulate RAMs using P systems with cell polarity. The basic idea is to represent the contents of the registers by means of sets of copies of objects in the basal environment (which is initially empty, corresponding to the values of all registers being equal to zero): to be precise, if register r_i contains value c_i, then c_i copies of object r_i will be present in the basal environment. The instructions are represented as follows. If the program counter contains the value i (that is, the instruction to be executed is the ith one), then an object $Instr_i$ will occur within the system. At the beginning of the computation, we start with the object $Instr_1$ inside an apical node.

The simulation of an instruction proceeds as follows. The symbol $Instr_i$ is sent out of the system and, at the same time, the symbols needed to simulate the corresponding instruction, and to prepare the execution of the next instruction, are brought into the system. In particular, if the ith instruction is an increment instruction, then a copy of object r_j is taken from the apical environment (together with an auxiliary object) and sent to the basal environment. At the same time, the object $Instr_{i+1}$ used to start the simulation of the next instruction enters into the system.

https://doi.org/10.1017/S0960129509990156 Published online by Cambridge University Press

If the ith instruction is an instruction of type $(i : DecJump(r_j, s))$, then two different behaviours are possible. If at least one copy of object r_j is present in the basal environment, then one of these copies is moved (through the system) from the basal to the apical environment, thus decreasing by one the value of register r_j, and the object $Instr_{i+1}$ is brought into the system. If, on the other hand, no copies of object r_j occur in the basal environment, then a jump instruction is simulated by bringing into the system the object $Instr_s$.

The simulation we propose is non-deterministic. In particular, when a 'wrong' branch of the simulation is executed, a particular object *loop* is introduced, which is then rewritten forever to induce a non-halting computation.

If the computation eventually halts, the output of the RAM encoding is the number of occurrences of object r_1 in the basal environment, just as the output of the RAM is the contents of register r_1 in the final state of the computation.

Theorem 4.1. P systems with cell polarity are Turing complete.

Proof. We show how to simulate a RAM machine M with m instructions and n registers by means of a (single-cell) P system with cell polarity.

Consider the system

$$\Pi = (\Sigma, A, B, C, E_A, E_B, conn)$$

where:

— $\Sigma = \{Instr_i \mid 1 \leqslant i \leqslant m+1\} \cup \{r_j \mid 1 \leqslant j \leqslant n\} \cup$
$\{Dec_{i+1,s}, Dec'_{i+1,s}, Dec''_{i+1,s}, D_j, D'_j \mid$
$(i : DecJump(r_j, s))$ is a decrement instruction of $M\} \cup$
$\{Inc, Inc', loop\}$;
— $A = \{A_1 = (\{Instr_1\}, R_{A_1})\}$;
— $B = \{B_1 = (\varnothing, R_{B_1})\}$;
— $C = \{C_1 = (\varnothing, R_{C_1})\}$;
— E_A is the apical environment, which initially contains an infinite number of each symbol from Σ;
— E_B is the basal environment, which is initially empty;
— $conn = \{(C_1, A_1), (C_1, B_1)\}$.

Various rules are defined in each node in order to simulate each instruction of the RAM. Moreover, each node A_1, B_1, and C_1 contains the evolution rule $(loop \to loop)$, which is used to induce infinite computations, as explained above.

To simulate an instruction $(i : Inc(r_j))$, $1 \leqslant i \leqslant m$, $1 \leqslant j \leqslant n$, we define the following sets of rules:

— inside R_{A_1}:
$(Instr_i, out;\ Instr_{i+1}, Inc, r_j, in)$
(Inc, r_j, go)
$(Inc' \to loop)$
— inside R_{B_1}:
(Inc', r_j, out)

https://doi.org/10.1017/S0960129509990156 Published online by Cambridge University Press

— inside R_{C_1}:
$(Inc \to Inc')$
(Inc', r_j, go)

Assume we have an object $Instr_i$ in the unique apical node corresponding to an instruction $(i : Inc(r_j))$. The simulation then proceeds as follows.

Using an antiport rule, the object $Instr_i$ is sent to the apical environment, while the objects Inc and r_j are brought into the node A_1 together with the object $Instr_{i+1}$, which can immediately start the simulation of the next instruction. Meanwhile, the objects Inc and r_j are sent together to node C_1, using the rule (Inc, r_j, go). Here, Inc is primed, and the objects are then sent out together. If they come back to node A_1, then Inc' is changed into the symbol *loop*, and the computation never halts, and no output is produced. If, on the other hand, they are sent to node B_1, they are moved to the basal environment by means of the symport rule (Inc', r_j, out), thus correctly adding an object r_j within it.

To simulate an instruction $(i : DecJump(r_j, s))$, we need to define the following sets of rules:

— inside R_{A_1}:
$(Instr_i, out;\ Dec_{i+1,s}, D_j, in)$
$(Dec_{i+1,s}, D_j, go)$
$(D'_j \to loop)$
$(Dec''_{i+1,s} \to loop)$
$(Dec''_{i+1,s}, r_j, out;\ Instr_{i+1}, in)$
$(Dec''_{i+1,s}, D'_j, out;\ Instr_s, in)$
— inside R_{B_1}:
$(D'_j, out;\ r_j, in)$
$(Dec''_{i+1,s}, r_j, go)$
$(Dec''_{i+1,s}, D'_j, go)$
$(Dec''_{i+1,s} \to loop)$
— inside R_{C_1}:
$(Dec_{i+1,s} \to Dec'_{i+1,s})$
$(D_j \to D'_j)$
$(Dec'_{i+1,s} \to Dec''_{i+1,s})$
(D'_j, go)
$(Dec''_{i+1,s}, go)$
$(Dec''_{i+1,s}, r_j, go)$
$(Dec''_{i+1,s}, D'_j, go)$

Assume we have an object $Instr_i$ in the unique apical node corresponding to an instruction $(i : DecJump(r_j, s))$. The simulation then proceeds as follows.

Using an antiport rule, the object $Instr_i$ is sent to the apical environment, while the objects $Dec_{i+1,s}$ and D_j are brought into the node A_1. The objects are then sent together to node C_1, using the rule $(Dec_{i+1,s}, D_j, go)$. Here they are both primed, and $Dec'_{i+1,s}$ is then changed to $Dec''_{i+1,s}$, while D'_j is sent out of the node. If D'_j comes back to node A_1, the symbol *loop* is produced at the next step, and the computation never halts. Otherwise, if it reaches node B_1, the simulation proceeds in one of two possible ways. If at least

https://doi.org/10.1017/S0960129509990156 Published online by Cambridge University Press

one object r_j is present in the basal environment, then the antiport rule $(D'_j, out;\ r_j, in)$ can be applied, bringing an object r_j into node B_1. If no object r_j is present in the basal environment, then no rule can be applied at this moment. Hence, a decrement is performed if and only if at least one object r_j occurs in E_B.

Meanwhile, object $Dec''_{i+1,s}$ is ejected from node C_1. If it reaches node A_1, then the only possible rule that can be applied at the next step is $(Dec''_{i+1,s} \to loop)$, which induces an infinite loop. If, instead, $Dec''_{i+1,s}$ is sent to cell B_1, then the computation can continue in different ways. We could apply the rule $(Dec''_{i+1,s} \to loop)$, thus inducing an infinite loop, or one of the rules $(Dec''_{i+1,s}, r_j, go)$, $(Dec''_{i+1,s}, D'_j, go)$, depending on the other symbol present in B_1:

(i) If we have the symbol r_j, we can apply the rule $(Dec''_{i+1,s}, r_j, go)$, and both objects $Dec''_{i+1,s}$ and r_j are sent to C_1. Here, we can apply $(Dec''_{i+1,s}, go)$, which sends $Dec''_{i+1,s}$ alone to either A_1 or B_1. In both cases, an infinite loop is induced by the rule $(Dec''_{i+1,s} \to loop)$. The other applicable rule is $(Dec''_{i+1,s}, r_j, go)$, which sends both objects to either A_1 or B_1. In this case, we obtain exactly the same conditions as we had two computation steps earlier, and the computation can proceed in the same way. Thus, after some time spent moving between nodes B_1 and C_1, both objects eventually reach node A_1. Here, we can apply the rule $(Dec''_{i+1,s} \to loop)$, again producing no output, or the rule $(Dec''_{i+1,s}, r_j, out; Instr_{i+1}, in)$, which correctly concludes the decrement operation, and brings into node A_1 the object corresponding to the next instruction (the $(i+1)$th one) to be simulated.

(ii) If B_1 contains the symbol D'_j, we can apply the rule $(Dec''_{i+1,s}, D'_j, go)$, whereby both objects $Dec''_{i+1,s}$ and D'_j are sent to C_1. Here we can apply the rule $(Dec''_{i+1,s}, D'_j, go)$, which sends out both objects, to either A_1 or B_1. As previously explained, after some cyclic movements between nodes B_1 and C_1, both objects can eventually reach node A_1. Here we can apply the rule $(Dec''_{i+1,s} \to loop)$, again producing no output, or the rule $(Dec''_{i+1,s}, D'_j, out; Instr_s, in)$, which correctly concludes the operation by jumping to instruction s.

As in the previous case, in C_1 we can also apply $(Dec''_{i+1,s}, go)$, which sends $Dec''_{i+1,s}$, alone, to A_1 or B_1; unlike the previous case, we can also apply the rule (D'_j, go), which sends D'_j, alone, to A_1 or B_1. If both objects are sent separately to the same node, the effect is the same as in the application of the rule $(Dec''_{i+1,s}, go)$. If they are sent to different nodes, they generate the *loop* object, inducing a non-halting computation.

Finally, if the object $Instr_{m+1}$ enters the system, no further rule can be applied, so the computation stops and the output is the number of symbols r_1 appearing in the basal environment. □

5. A formal description of the transcellular transport of glucose in intestinal cells

In this section we provide a formal description of the transcellular transport of glucose in intestinal epithelial cells by exploiting the concept of orientation and the communication rules of a P system with cell polarity. In particular, we describe this process by focusing on the (symport, antiport, uniport) communication of the solutes (potassium and sodium

https://doi.org/10.1017/S0960129509990156 Published online by Cambridge University Press

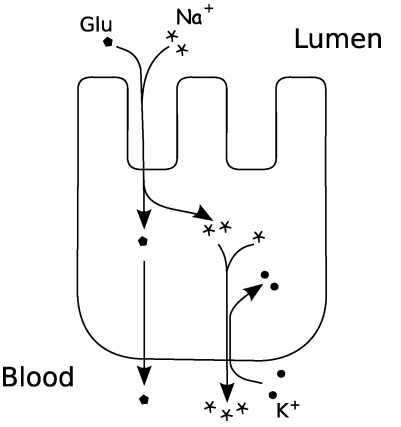

Fig. 3. Glucose transport in intestinal epithelial cells.

ion and glucose) carried out by specific transmembrane transport proteins placed at the apical and basal faces of the membrane (see Figure 3). This biological process can occur efficiently in intestinal cells thanks to their polarised structure and to the adherent junctions, which prevent the transported substances from diffusing back into the intestinal lumen. For further details about this biological process, see Alberts *et al.* (2002), Lodish *et al.* (2000) and Nelson and Cox (2004).

Formally, we define the P system with cell polarity for transepithelial glucose transport as $\Pi_{gluc} = (\Sigma, A, B, C, E_A, E_B, conn)$ where:

— $\Sigma = \{gluc, K^+, Na^+\}$ is the alphabet denoting the solutes glucose, potassium and sodium, respectively;
— $A = \{A_1, \ldots, A_{k_1}\}$, $k_1 \in \mathbb{N}$, is the set of apical nodes, where $\omega_{A_i} = \emptyset$ and $R_{A_i} = \{(2Na^+, gluc, in), (gluc, go), (Na^+, go)\}$, for all $1 \leqslant i \leqslant k_1$;
— $B = \{B_1, \ldots, B_{k_2}\}$, $k_2 \in \mathbb{N}$, is the set of basal nodes, where $\omega_{B_i} = \emptyset$ and $R_{B_i} = \{(3Na^+, out; 2K^+, in), (gluc, out)\}$, for all $1 \leqslant i \leqslant k_2$;
— $C = \{C_1, \ldots, C_{k_3}\}$, $k_3 \in \mathbb{N}$, is the set of internal nodes, where $\omega_{C_i} = \emptyset$ and $R_{C_i} = \{(gluc, go), (Na^+, go)\}$, for all $1 \leqslant i \leqslant k_3$;
— E_A is the *apical environment*, initially containing an (up to) infinite number of objects from Σ;
— E_B is the *basal environment*, initially containing an (up to) infinite number of objects from Σ;
— $conn = C \times (A \cup B \cup C)$.

The transepithelial transport of glucose from the intestinal lumen (apical face) into the bloodstream (basal face) is a cellular phenomenon involving two main phases.

In the first phase, glucose is imported from the lumen into the cell by means of transmembrane transport proteins placed (on the microvilli) at the apical face of the intestinal cells. Since the inflow of glucose occurs against its concentration gradient (the intracellular concentration of glucose being higher than its luminal concentration), its energetically unfavorable inward passage is coupled to the energetically favorable inward

https://doi.org/10.1017/S0960129509990156 Published online by Cambridge University Press

passage of two sodium ions. This movement is described by means of the symport rule $(2Na^+, gluc, in)$ in apical nodes. The number k_1 of apical nodes can here be imagined as the copy number of symporters (that is, transport proteins that allow the symport of molecules) occurring at the apical membrane. Glucose and sodium ions can then diffuse into the cytoplasm, where they can either take part in other cellular processes (for example, the metabolism of glucose), or reach the basal side of the cell. This step is modelled by means of communication rules of type $(gluc, go), (Na^+, go)$, which move the objects from the apical to the internal nodes. The connections between apical (or basal) nodes and internal nodes represent, in fact, all the possible ways of diffusing within the cellular cytosol.

The glucose and sodium objects can then move from internal nodes to basal nodes by the rules $(gluc, go), (Na^+, go)$. Note that since the application of communication rules is assumed to be non-deterministic, these objects can even go back to apical nodes from where, however, they can only return to internal nodes (by means of the similar rules $(gluc, go), (Na^+, go)$ defined inside the apical nodes).

The second phase of the process consists of the outflow of glucose and sodium ions from the cells into the extracellular ambient, which occurs at the basal face of intestinal cells by means of different transport proteins. In order to maintain low (physiological) concentrations within the cell, sodium has to be exported from the cell. The sodium outflow is carried out by transport proteins called ATPases, or pumps, which exploit the energy released by the hydrolysis of ATP molecules (see also Besozzi and Ciobanu (2005)). Here, we will only look at the communication process and will not consider the role of ATP: we will just describe the movement of ions using the antiport rule $(3Na^+, out; 2K^+, in)$, which denotes the fact that three sodium ions are moved to the basal environment while, at the same time, two potassium ions are taken into the cell. On the other side, the glucose outflow at the basal face of the cell occurs by means of uniporter proteins, which catalyse the passage of glucose down its concentration gradient (the extracellular basal concentration of glucose being lower than its intracellular concentration). This movement is described through the uniport rule $(gluc, out)$, which is defined in basal nodes. The number k_2 of basal nodes can be seen as the sum of the copy numbers of both ATPases and uniporters occurring at the basal membrane.

A more detailed model of the transepithelial movement of glucose would require us also to consider the extracellular and intracellular concentrations of glucose and ions, as well as the fact that these solutes move according to (that is, against or down) their concentration gradients. Since we have not defined the application of rules based on some 'control condition' for P systems with cell polarity, we cannot include these biological aspects directly in the formal description presented in this section, but a further extension of P systems with cell polarity could include this. See Besozzi and Rozenberg (2006) for more detailed formalisations of the functioning of transmembrane transport proteins.

As described in Section 2, epithelial cells also have several basolateral junctions, which are needed to keep adjacent cells in tight contact with each other, and to allow a direct link (via gap junctions) between the cytoplasmic regions of the coupled cells. Many biological phenomena occurring in tissues depend on the existence of these communication junctions (Alberts *et al.* 2002; Lodish *et al.* 2000). For instance, the electrical coupling

https://doi.org/10.1017/S0960129509990156 Published online by Cambridge University Press

through gap junctions allows the contractive synchronisation of heart muscle cells; also, some nerve cells are coupled electrically and can spread the action potential very rapidly from cell to cell, thus avoiding the transmission delay at synapses. Gap junctions also permit the exchange of small metabolites, ions or intracellular secondary messengers, thus providing the coordination of linked cells (for example, during embryogenesis, or for hormonal stimulation in the liver and pancreas). Intercellular processes like these cannot be modelled by single-cell polarised P systems, but require the definition of tissue-like systems, where the cells are linked together through appropriate communication channels. We propose a formalisation of tissue systems like this in the following section.

6. Tissue P systems with cell polarity

In this section we introduce tissue P systems with cell polarity. Informally, this type of system is composed of several polarised cells, which are all oriented in the same direction (and look on common apical and basal environments) and are linked together by means of particular connections, called *gap junctions*. Gap junctions are 'communication channels' between pairs of internal nodes, each belonging to two adjacent polarised cells. In particular, note that we do not consider the existence of gap junctions between apical (or basal) nodes and internal nodes. In this way, we reflect the biological presence of adherent junctions between adjacent cells in the epithelial tissue, which ensures that the apical and basal regions of the membranes are well separated.

Moreover, since cellular gap junctions between adjacent cells are not always open, we consider that, during the computation, each gap connection can switch between two states: *open*, meaning that objects can be exchanged among the adjacent cells through the gap connection, or *closed*, when no object can be communicated through it. The state of a gap junction is determined, step by step, by means of two 'control' multisets, one for each node involved in the connection. The gap junction is open only if both control multisets are not contained in the current multisets of the nodes, otherwise the connection remains closed.

Hence, we build a system structure where each cell can be seen as a computing unit *per se* – with its own apical, basal and internal nodes, connected in a specified way – but also where different cells are coupled and can communicate by means of appropriate intercellular links.

Formally, a *tissue P system with cell polarity* (TPCP) of degree k, $k \geqslant 2$, is a structure

$$\Theta = (\Pi_1, \Pi_2, \ldots, \Pi_k, GJ)$$

where:

— Π_i is a polarised cell such that $or(\Pi_i) = (E_A \rightsquigarrow E_B)$, for all $1 \leqslant i \leqslant k$;
— $GJ = \{\langle gap_{i,j}, \widehat{M}_{i,j}\rangle \mid 1 \leqslant i, j \leqslant k\}$ is a set of *gap junctions* between two polarised cells Π_i, Π_j, where:
  – $gap_{i,j} \subseteq \{\{C_{i,1}, C_{i,2}, \ldots, C_{i,n}\} \times \{C_{j,1}, C_{j,2}, \ldots, C_{j,m}\} \mid C_{i,h}, 1 \leqslant h \leqslant n, C_{j,l}, 1 \leqslant l \leqslant m,$ are internal nodes of Π_i and Π_j, respectively$\}$ is the set of *gap connections* between Π_i and Π_j, $1 \leqslant i, j \leqslant k, i \neq j$;

https://doi.org/10.1017/S0960129509990156 Published online by Cambridge University Press

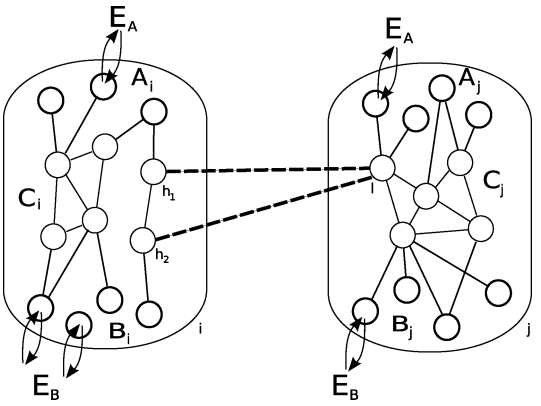

Fig. 4. Gap junctions between two polarised cells in a tissue.

– $\widehat{M}_{i,j} = \{(\widehat{\omega}_{i,h}, \widehat{\omega}_{j,l}) \mid \widehat{\omega}_{i,h}, \widehat{\omega}_{j,l}$ are multisets over Σ, for all $1 \leqslant h \leqslant n, 1 \leqslant l \leqslant m$, such that $(C_{i,h}, C_{j,l}) \in gap_{i,j}\}$, is the set of *control multisets* of the gap connections $gap_{i,j}$, $1 \leqslant i, j \leqslant k, i \neq j$.

As an example, Figure 4 shows the structure of a TPCP of degree 2, with a representation of two gap connections, $(C_{i,h_1}, C_{j,l}), (C_{i,h_2}, C_{j,l})$, between the internal nodes C_{i,h_1}, C_{i,h_2} in cell Π_i, and the internal node $C_{j,l}$ in cell Π_j. Note also that both cells face the same apical and basal environments, and have the same orientation.

The set of gap junctions defines how many communication channels (that is, gap connections) there are between the internal nodes of two adjacent cells. These connections are exactly the means by which adjacent cells in a tissue can exchange objects. Consider, for instance, an internal node $C_{i,h}$ in cell Π_i, which contains a communication rule of the type (u, go): the application of such a rule will determine the non-deterministic choice of the target node – chosen from amongst all the nodes connected to $C_{i,h}$ – to which the multiset u will be sent. These nodes can be apical, basal or internal nodes in the cell Π_i (connected to $C_{i,h}$ by means of connections in the set *conn*), or the internal nodes of another cell Π_j (connected to $C_{i,h}$ by means of gap connections in the set $gap_{i,j}$).

The multiset u can be sent non-deterministically from $C_{i,h}$ to other nodes in Π_i without any restriction, as defined in Section 3 for a single-cell P system with polarity. On the other hand, u can be communicated to an internal node of a different cell *only* if the gap connection between these two nodes is in an appropriate 'state'. The state of the gap connection is determined throughout the computation, step by step, by looking at both the current multisets and the control multisets of the connected nodes. To be precise, we define the *state function* of the gap connections, $S_t : gap_{i,j} \rightarrow \{open, closed\}$, at time step $t, t \geqslant 0$, as follows. For each $(C_{i,h}, C_{j,l}) \in gap_{i,j}$, $1 \leqslant i, j \leqslant k, i \neq j$, we have:

$$S_t(C_{i,h}, C_{j,l}) = \begin{cases} open & \text{if } \widehat{\omega}_{i,h} \not\subseteq \omega_{i,h} \text{ and } \widehat{\omega}_{j,l} \not\subseteq \omega_{j,l} \\ closed & \text{otherwise} \end{cases}$$

https://doi.org/10.1017/S0960129509990156 Published online by Cambridge University Press

where $\widehat{\omega}_{i,h}, \widehat{\omega}_{j,l}$ are the control multisets over Σ of the gap connection $(C_{i,h}, C_{j,l})$, and $\omega_{i,h}, \omega_{j,l}$ are the multisets over Σ occurring at time t in the internal nodes $C_{i,h}$ and $C_{j,l}$, respectively.

In other words, at any step of the computation, a gap connection can be in either the open or closed state. The gap connection is open only if the control multisets of *both* connected internal nodes are not contained within the multisets currently occurring within these nodes. On the other hand, if at least one control multiset appears inside one node, then the gap connection is closed (in both communication directions). Note that if an internal node, say $C_{j,l}$ in cell Π_j in Figure 4, is connected to two or more internal nodes of another cell, say C_{i,h_1}, C_{i,h_2} in cell Π_i in Figure 4, then the gap connection $(C_{i,h_1}, C_{j,l})$ may be open even if the other connection $(C_{i,h_2}, C_{j,l})$ is closed (this happens when C_{i,h_2} contains its control multiset within its current multiset). Obviously, both connections are closed if $C_{j,l}$ contains its control multiset. The state of each gap connection can change from step t to step $t+1$ because of the application of communication and evolution rules that involve the multisets contained in the internal nodes.

A *configuration* $\mathscr{C}_t$ of a TPCP at step t, $t \geqslant 0$, is determined by the multisets occurring at time t within each (apical, basal and internal) node of every polarised cell. Formally, we have $\mathscr{C}_t = (\mathscr{C}_1(t), \mathscr{C}_2(t), \ldots, \mathscr{C}_k(t))$, where $\mathscr{C}_i(t) = \{\omega_{A_1}, \ldots, \omega_{A_{k_1}}, \omega_{B_1}, \ldots, \omega_{B_{k_2}}, \omega_{C_1}, \ldots, \omega_{C_{k_3}}\}$ is the configuration of the cell Π_i at step t, and $\omega_{A_p}, \omega_{B_q}, \omega_{C_r}$, $1 \leqslant p \leqslant k_1, 1 \leqslant q \leqslant k_2, 1 \leqslant r \leqslant k_3$, are the multisets occurring at step t in the corresponding apical, basal or internal nodes, respectively, of cell $\Pi_i, 1 \leqslant i \leqslant k$.

By looking at the configuration of all polarised cells in the TPCP, and by checking the conditions determined by such configurations for all gap connections, we are therefore able to determine which internal nodes of two linked (adjacent) cells can exchange objects, at any step t of the computation.

Formally, a *computation* of a TPCP Θ is a sequence of configurations $(\mathscr{C}_0, \mathscr{C}_1, \mathscr{C}_2, \ldots)$ starting from the initial configuration of Θ. A *transition* $\mathscr{C}_t \rightarrow \mathscr{C}_{t+1}$ between two consecutive configurations, at time steps $t, t+1$, $t \geqslant 0$, is determined as follows. All cells appearing in Θ are processed in parallel and, within each cell, the prescriptions given in Section 3 for symport, antiport, evolution and communication rules are followed.

The computation of a TPCP halts when no further rule can be applied within any node of any cell of the tissue P system. The output of a TPCP is defined by the multiset of symbols from (a specified subset of) Σ that occurs in the basal environment in the halting configuration.

7. Conclusions

Many variants of P systems inspired by various biological phenomena have been defined in the literature. Following in this tradition, in this paper we have introduced P systems with cell polarity as a synchronous, parallel and distributed model of computation inspired by the structure and functioning of the cells that compose the mono-layered intestinal epithelial tissue. In these systems, polarity is intended to represent the mutual positions of the computing elements that occur in each cell. These computing elements are placed in the nodes of a directed graph, and are of three types, apical, basal and internal, all living

https://doi.org/10.1017/S0960129509990156 Published online by Cambridge University Press

in the same region (the skin membrane). Communication of objects between the nodes (as well as with the apical and the basal environments) is performed through symport, antiport, uniport and communication rules. While they are inside the system, single objects can also be modified by non-cooperative and conservative evolution rules.

We have shown two preliminary results for this new model of computation. From a theoretical point of view, we have shown that P systems with cell polarity achieve Turing completeness since they are able to simulate RAMs non-deterministically. From a modelling point of view, we have shown that the transepithelial movement of glucose from the intestinal lumen into the blood can be described by such a formal system. We have subsequently defined tissue P systems with cell polarity as a model of tissue P systems in which cells are connected by gap junctions. These are communication channels that can move objects between different cells during computations, depending on their state, which can be open or closed. Such states are controlled by the presence of specific multisets of objects in the cells connected by the junction.

Two directions for future research are of clear interest. First, determining whether a deterministic simulation of RAMs can be performed using a single cell remains an open problem. It would also be interesting to assess the computational power of purely communicative systems, where evolution rules are not admitted. A natural question to ask for these systems is whether we can, by exploiting the controlled states of gap junctions of polarised tissue P systems, achieve the computational power of Turing machines, considering the various restrictions on the rules used.

References

Alberts, B., Johnson, A., Lewis, J., Raff, M., Roberts, K. and Walter, P. (2002) *Molecular Biology of the Cell*, 4th edition, Garland Science, New York.

Alhazov, A., Freund, R. and Oswald, M. (2006) Cell/symbol complexity of tissue P systems with symport/antiport rules. *Int. Journal of Foundations of Computer Science* **17** (1) 3–26.

Alhazov, A., Rogozhin, Y. and Verlan, S. (2005) Symport/antiport tissue P systems with minimal cooperation. In: *Proc. of the ESF Exploratory Workshop on Cellular Computing (Complexity Aspects)*, Sevilla, Spain 37–52.

Bernardini, F. and Gheorghe, M. (2005) Cell communication in tissue P systems: universality results. *Soft Computing* **9** (9) 640–649.

Besozzi, D. and Ciobanu, G. (2005) A P system description of the sodium-potassium pump. In: Mauri, G., Păun, Gh., Pérez-Jiménez, M. J., Rozenberg, G. and Salomaa, A. (eds.) Proc. of 5th Membrane Computing International Workshop (WMC04). *Springer-Verlag Lecture Notes in Computer Science* **3365** 210–223.

Besozzi, D. and Rozenberg, G. (2006) Formalizing spherical membrane structures and membrane proteins populations. In: Hoogeboom, H. J., Păun, Gh., Rozenberg, G. and Salomaa, A. (eds.) Membrane Computing, 7th International Workshop, WMC 2006, Leiden. *Springer-Verlag Lecture Notes in Computer Science* **4361** 18–41.

Cavaliere, M. (2003) Evolution–communication P systems. In: Păun, Gh., Rozenberg, G., Salomaa, A. and Zandron, C. (eds.) Membrane Computing. International Workshop, WMC–CdeA 02. *Springer-Verlag Lecture Notes in Computer Science* **2597** 134–145.

Freund, R. and Oswald, M. (2003) P systems with activated/prohibited membrane channels. In: Păun, Gh., Rozenberg, G., Salomaa, A. and Zandron, C. (eds.) Membrane Computing.

International Workshop, WMC–CdeA 02. *Springer-Verlag Lecture Notes in Computer Science* **2597** 261–269.

Freund, R., Păun, Gh. and Pérez-Jiménez, M. J. (2005) Tissue-like P systems with channel states. *Theoretical Computer Science* **330** (1) 101–116.

Lodish, H., Berk, A., Zipursky, S. L., Matsudaira, P., Baltimore, D. and Darnell, J. E. (2000) *Molecular Cell Biology*, 4th Edition, W. H. Freeman and Co.

Martín-Vide, C., Păun, Gh., Pazos, J. and Rodríguez-Patón, A. (2003) Tissue P systems. *Theoretical Computer Science* **296** 295–326.

Martín-Vide, C., Pazos, J., Păun, Gh. and Rodríguez-Patón, A. (2002) A new class of symbolic abstract neural nets: tissue P systems. In: Proc. of COCOON 2002, Singapore. *Springer-Verlag Lecture Notes in Computer Science* **2387** 290–299.

Minsky, M. L. (1967) *Computation: Finite and Infinite Machines*, Prentice-Hall.

Nelson, D. L. and Cox, M. M. (2004) *Lehninger Principles of Biochemistry*, 4th Edition, W. H. Freeman and Co

Oswald, M. (2007) Independent agents in a globalized world modelled by tissue P systems. *Artificial Life and Robotics* **11** (2) 171–174.

Păun, A. and Păun, Gh. (2002) The power of communication: P systems with symport/antiport. *New Generation Computing* **20** 295–305.

Păun, A., Păun, Gh. and Rozenberg, G. (2002) Computing by communication in networks of membranes. *Int. Journal of Foundations of Computer Science* **13** 779–798.

Păun, Gh. (1998) Computing with membranes. *Journal of Computer and System Sciences* **61** (1) 108–143. (See also Turku Centre for Computer Science – TUCS Report No. 208, 1998. Available at: `http://www.tucs.fi/Publications/techreports/TR208.php`.)

Păun, Gh. (1999) Computing with membranes. An introduction. *Bulletin of the EATCS* **67** 139–152.

Păun, Gh. (2002) *Membrane Computing: An Introduction*, Springer-Verlag.

Păun, Gh. and Rozenberg, G. (2002) A guide to Membrane Computing. *Theoretical Computer Science* **287** (1) 73–100.

Păun, Gh., Sakakibara, Y. and Yokomori, T. (2002) P systems on graphs of restricted forms. *Publicationes Mathematicae Debrecen* **60** 635–660.

Rogozhin, Y., Alhazov, A. and Freund, R. (2006) Computational power of symport/antiport: history, advances and open problems. In: Freund, R., Păun, Gh., Rozenberg, G. and Salomaa, A. (eds.) Membrane Computing, 6th International Workshop, WMC 2005. *Springer-Verlag Lecture Notes in Computer Science* **3850** 1–31.

Ross, M. R. and Pawlina, W. (2006) *Histology: A Text and Atlas with Correlated Cell and Molecular Biology*, 5th edition, Lippincott Williams and Wilkins.

Shepherdson, J. C. and Sturgis, J. E. (1963) Computability of recursive functions. *Journal of the ACM* **10** 217–255.

Verlan, S., Bernardini, F., Gheorghe, M. and Margenstern, M. (2006) Computational completeness of tissue P systems with conditional uniport. In: Hoogeboom, H. J., Păun, Gh., Rozenberg, G. and Salomaa, A. (eds.) Membrane Computing, 7th International Workshop, WMC 2006, Leiden. *Springer-Verlag Lecture Notes in Computer Science* **4361** 521–535.