Multi-Scale Modelling of Biological Systems in Process Algebra with Multi-Way Synchronisation

Andrea Degasperi
Sir Alwyn Williams Building
School of Computing Science
University of Glasgow, Glasgow, U.K.
Andrea.Degasperi@glasgow.ac.uk

Muffy Calder
Sir Alwyn Williams Building
School of Computing Science
University of Glasgow, Glasgow, U.K.
Muffy.Calder@glasgow.ac.uk

ABSTRACT

We present a stochastic process algebra designed for modelling biological systems at multiple scales, called process algebra with hooks (PAH). In PAH, processes represent different scales, e.g. biochemistry, cells or tissue. There are two synchronisation operators, distinguishing interactions within and between scales; composed actions represent events that occur at multiple scales. A stochastic semantics is provided, based on functional rates derived from kinetic laws. A parametric version of the algebra ensures that a model description is compact. An example illustrates how the algebra can be used to model and simulate multi-scale tissue growth, where growth and death of tissue depend on the local concentration of chemicals.

Categories and Subject Descriptors

F.4.3 [**Mathematical Logic and Formal Languages**]: Formal Languages—*Algebraic Language Theory*; G.3 [**Probability and Statistics**]: Stochastic Processes; J.3 [**Life and Medical Sciences**]: Computational Biology

Keywords

Process algebra, multi-scale, functional rates, tissue growth

1. INTRODUCTION

Stochastic process algebra (SPA) has shown to be well suited for quantitative modelling and analysis of biological systems [2, 5, 11]. The main focus has been on biochemical reactions and interactions within and between compartments. Here, we consider how SPA can be used for modelling and quantitative analysis of biochemistry and location, as well as higher order structures such as cells and tissue: this requires an approach that allows modelling across multiple *scales*.

By *scale* we mean a level of abstraction. For example, at the biochemical scale, molecules are entities and reactions are events; at the cellular scale, cells are entities and movement, proliferation, death are events. Usually, entities and events at a scale can be described in more detail using the entities and events of other scales. For example, cells are made of molecules and cell movement is the result of a large number of molecular reactions. Our main interest is models of tissue growth and pattern formation, for which the current standard practice consists of partial differential equations (PDEs) and two-dimensional cellular automata (CA) [10]. However, both PDEs and CA do not provide a formal framework for the comparison and substitution of parts of models, such as congruences, nor do they support compositionality, concepts intrinsic to process algebra.

In this paper we focus on formalising the concept of scale, with particular attention to modelling actions within a scale and between scales. We propose an algebra called process algebra with hooks (PAH), based on an early version published in [6], to formalise multi-scale models. In our previous work we followed a *bottom-up* approach where the biochemical scale determines the rates and other scales are abstractions of lower scales. Here we follow a *middle-out* [9] approach where one can begin modelling at any scale, and then relate to higher or lower scales. In particular, PAH provides:

- the concept of processes at a given scale, events that occur within that scale and events that occur between scales;
- a stochastic semantics based on functional rates;
- two multi-way synchronisation composition operators, $\underset{\mathcal{L}}{\bowtie}$, which expresses cooperation between processes at the same scale, and $\underset{\mathcal{L}}{\rotatebox[origin=c]{90}{$\bowtie$}}$, which expresses cooperation between processes at different scales. The second operator, $\underset{\mathcal{L}}{\rotatebox[origin=c]{90}{$\bowtie$}}$, is an improved definition of the listen operator $\underset{\mathcal{L}}{\lhd}$ introduced in [6];
- relations that allow comparison between models according to specified scales and substitution of parts that are behaviourally congruent (not discussed here).

After a gentle introduction to the concepts and the design choices of PAH in the next section, we move to the theoretical foundations of the language, presenting syntax, semantics, isomorphism and functional rate evaluation. This is followed by a parametric form of the syntax, to make specifications more compact. There follows an application of the approach, focussing on tissue growth in a two dimensional space. To summarise, the contributions of this paper are:

- the definition of the new operator $\underset{\mathcal{L}}{\rotatebox[origin=c]{90}{$\bowtie$}}$;

CMSB '11, September 21-23, 2011 Paris, FR

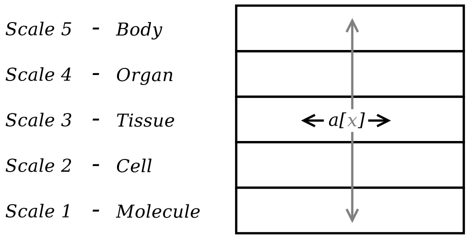

Figure 1: In process algebra with hooks composed actions present layer actions that operate within the current scale and hook actions (here a) that operate between scales (here x).

- details of a stochastic semantics for PAH based on functional rates and their evaluation;
- the definition of isomorphism on PAH processes;
- the definition of a parametric version of PAH, which reduces the length of model definitions;
- the illustration of use of PAH with a multi-scale model of tissue growth.

The paper is organised as follows. In Section 2 process algebra with hooks is introduced. In Section 3 syntax, semantics, isomorphism and functional rate evaluation are defined. A parametric version of PAH is presented in Section 4 and used to define a multi scale model of tissue growth in Section 5. This is followed by discussion of related work in Section 6 and conclusions and future work in Section 7.

2. CONCEPTS OF PROCESS ALGEBRA WITH HOOKS

In process algebra, autonomous agents called *processes* are used to represent behaviour. The atomic behaviour is represented by *actions* and a process P can perform a sequence of actions (P defined as $a.b.c.P'$), can choose between actions (P defined as $a.P' + b.P''$) or can perform actions concurrently (P defined as $a.P' \underset{\emptyset}{\bowtie} b.P''$). Additionally, two processes can synchronise on one action. Usually two styles of synchronisation are considered: binary, CCS style [8], and multi-way, CSP style [7]. In binary synchronisation, only two processes can synchronise at a time using complementary action names. In multi-way synchronisation, any number of processes can synchronise on a single action name. In this paper we use the second type of synchronisation, which is performed using the cooperation operator $\underset{\mathcal{L}}{\bowtie}$, where $\mathcal{L}$ is a set indicating on which actions the synchronisation is possible. Our choice of multi-way over binary synchronisation is motivated by the fact that the former allows one to model biochemical reactions with any number of reactants with a single action, and so atomically [5]. Moreover, a rate based on one of a variety of kinetic laws [12] can be associated to that action.

The main differences between PAH and a traditional process algebra with multi-way synchronisation, such as CSP [7], are the substitution of simple a actions with more complex composed actions $\mathcal{L}'[\mathcal{L}'']$ and the addition of the vertical cooperation operator $\underset{\mathcal{L}}{\mathbin{\rotatebox[origin=c]{90}{$\bowtie$}}}$. The interpretation of these new features is as follows:

- $\mathcal{L}'[\mathcal{L}'']$ is interpreted as "on this scale perform the actions in set $\mathcal{L}'$ alltogether, while broadcasting actions in set $\mathcal{L}''$ to the other scales". This mechanism is depicted in Figure 1. We will refer to actions performed by a process as *layer* actions if they belong to set $\mathcal{L}'$ or *hook* actions if they belong to set $\mathcal{L}''$. Hook actions synchronise with, and so "hook", layer actions in other scales;

- $\underset{\mathcal{L}}{\mathbin{\rotatebox[origin=c]{90}{$\bowtie$}}}$ synchronises layer actions on one side of the operator with hook actions on the other side via actions present in $\mathcal{L}$. This implies that the process on the left is on a different scale from the process on the right. No hook with hook or layer with layer action synchronisations are allowed with this operator.

The modeller should bear in mind that the set of actions $\mathcal{A}$ in a composed action $\mathcal{A}[\mathcal{E}]$ has been introduced with the sole intent of capturing multiple hook actions coming from other scales at the same time. For examples of how composed actions are used see [6].

2.1 Modelling Entities and Events

Consider the following example. In a cell C, there are two molecules A and B. A can increase its concentration via biochemical reaction R_a, B can decrease its concentration via R_b, while A can turn into B via R_c as follows:

$$R_a : \rightarrow \mathrm{A} \quad R_b : \mathrm{B} \rightarrow \quad R_c : \mathrm{A} \rightarrow \mathrm{B}$$

The cell C can have two states, C_0 and C_1, which depend on the concentration of B. When C is in C_0 it performs cell action m, while when it is in C_1 it performs cell action n. Moreover, when the concentration of B in the cell is high, then the cell is in state C_1, C_0 otherwise. We can model this scenario using PAH in the following way:

$$\begin{array}{lll} A_L \triangleq a.A_M & A_M \triangleq a.A_H + c.A_L & A_H \triangleq c.A_M \\ B_L \triangleq c.B_M & B_M \triangleq c[x].B_H + b.B_L & B_H \triangleq b[y].B_M \\ C_0 \triangleq x.C_1 + m.C_0 & C_1 \triangleq y.C_0 + n.C_1 & \end{array}$$

The concentration of A and B is represented by three processes for each molecule, indicating a concentration level, from low (L) to high (H). The state of the cell is represented by two processes C_0 and C_1. We use hook actions x and y to indicate that the concentration of B has passed a threshold and that the state of C has to change at the same time. The initial state is:

$$(A_H \underset{\{c\}}{\bowtie} B_M) \underset{\{x,y\}}{\mathbin{\rotatebox[origin=c]{90}{$\bowtie$}}} C_0$$

The vertical synchronisation operator $\underset{\{x,y\}}{\mathbin{\rotatebox[origin=c]{90}{$\bowtie$}}}$ clearly separates the molecular scale from the cellular scale, while indicating that actions x and y are actions that operate between scales. An example of a valid transition is:

$$(A_H \underset{\{c\}}{\bowtie} B_M) \underset{\{x,y\}}{\mathbin{\rotatebox[origin=c]{90}{$\bowtie$}}} C_0 \xrightarrow{\{c,x\}} (A_M \underset{\{c\}}{\bowtie} B_H) \underset{\{x,y\}}{\mathbin{\rotatebox[origin=c]{90}{$\bowtie$}}} C_1$$

On the label of the transition we have both c, which indicates that biochemical reaction R_c took place and x which indicates that a threshold of concentration of B has been crossed and that cell C changed its state from C_0 to C_1.

2.2 Rating Transitions

In order to determine the rate at which the above transition takes place, i.e. the parameter of the exponential distribution of the time necessary for the transition to happen, we employ functional rates. The motivation for functional rates comes from the fact that rates of biological events often depend on the current state of the system. In terms of process algebra, this means that an action is associated with a set of processes and that these processes are associated with variables and values that are used to evaluate functional rates. Processes represent the concentration of species (the variables) and the current concentration level (the values). Functional rates based on kinetic laws are associated with actions. When processes synchronise via a specific action, the corresponding functional rate is evaluated according to the information associated to the processes. We use functions $Var(\cdot)$ and $Val(\cdot)$ to associate variables and values with processes. In the example above have:

$$\begin{array}{lll} Var(A_L) = A & Var(A_M) = A & Var(A_H) = A \\ Val(A_L) = 0 & Val(A_M) = 1 & Val(A_H) = 2 \\ Var(B_L) = B & Var(B_M) = B & Var(B_H) = B \\ Val(B_L) = 0 & Val(B_M) = 1 & Val(B_H) = 2 \end{array}$$

A suitable velocity, in terms of concentration (M, molar) per second (s), for reaction R_c is given by $v_c = k_c[A]$, where k_c is a kinetic constant and $[A]$ indicates the concentration of molecule A. From v_c we can derive a rate for the above transition, in analogy with the *continuous time Markov chains with levels of concentration* approach [4]. A functional rate is defined as:

$$f_c = (k_c \cdot A \cdot h)/h$$

where A is the variable that indicates the current concentration level for molecule A, while h indicates how much concentration is represented by a single concentration level. For details of how this rate is formulated see Appendix A. At this point we can construct an environment Γ using variables and values associated with the processes, and additional constant definitions. From the initial state we obtain an environment $\Gamma' = \{(Var(A_H), Val(A_H)), (Var(B_M), Val(B_M))\}$. We add the environment to a transition. Thus:

$$(A_H \underset{\{c\}}{\bowtie} B_M) \underset{\{x,y\}}{\mathbin{\rotatebox[origin=c]{90}{$\bowtie$}}} C_0 \xrightarrow{(\{c,x\},\Gamma')} (A_M \underset{\{c\}}{\bowtie} B_H) \underset{\{x,y\}}{\mathbin{\rotatebox[origin=c]{90}{$\bowtie$}}} C_1$$

The pair $(\{c,x\},\Gamma')$ is called an *activity.* Combining Γ' with the environment $\Gamma'' = \{(k_c, 1), (h, 1)\}$ we construct environment $\Gamma = \Gamma' \cup \Gamma''$ which is used to evaluate functional rate f_c. The result is the *rated* transition

$$(A_H \underset{\{c\}}{\bowtie} B_M) \underset{\{x,y\}}{\mathbin{\rotatebox[origin=c]{90}{$\bowtie$}}} C_0 \xrightarrow{(\{c,x\},2)} (A_M \underset{\{c\}}{\bowtie} B_H) \underset{\{x,y\}}{\mathbin{\rotatebox[origin=c]{90}{$\bowtie$}}} C_1$$

because f_c evaluated with environment Γ is equal to 2.

2.3 Compositionality and Relations

A key concept in PAH is the idea of relating processes with respect to a specified scale, hiding as much as possible of other scales. This involves *filtering* action sets on a transition. For example, consider the rated transition $M \xrightarrow{(\{x,c\},2)} M'$. If one wants to focus on the biochemical scale (actions a, b, c) the result of filtering is filtered transition $M \xrightarrow{(\{c\},2)} M'$. If one wants to focus on the cellular scale (actions n, m) the result of filtering is filtered transition $M \xrightarrow{(\{\},2)} M'$. Notice that, while there is no action name, the transition cannot be hidden completely, because the rate, and so the delay of the transition, is still necessary to represent the correct timing of this and other events in the system. The operation of *filtering* hides information about actions and one can define relations based on filtered behaviour. For example, one could determine whether two models with different actions at the biochemical scale present the same actions with the same timing at the cellular scale. Because of the limited space, filtering and relations between PAH processes are not discussed further in this paper.

3. PROCESS ALGEBRA WITH HOOKS

3.1 Syntax and Semantics

$$D ::= nil \mid \mathcal{L}'[\mathcal{L}''].A \mid D + D$$

$$M ::= A \mid M \underset{\mathcal{L}}{\bowtie} M \mid M \underset{\mathcal{L}}{\mathbin{\rotatebox[origin=c]{90}{$\bowtie$}}} M$$

- D is a *definition* process, $D \in \mathbb{P}_d$, while M is a *model* process, $M \in \mathbb{P}_m$. Definition and model processes are disjoint and are both processes, i.e. $\mathbb{P}_d \cup \mathbb{P}_m = \mathbb{P}$;
- a is an action, $a \in Actions$, with $Actions$ the set of actions;
- $\mathcal{L} \subseteq Actions$, $\mathcal{L}' \subseteq Actions \wedge \mathcal{L}' \neq \emptyset$, $\mathcal{L}'' \subseteq Actions \wedge |\mathcal{L}''| \leq 1$;
- $\mathcal{L}'[\mathcal{L}'']$ is a composed action;
- nil is the deadlock process;
- A is an agent, used to define processes, via the agent definition $A \triangleq D$;
- sequential action execution $\mathcal{L}'[\mathcal{L}''].A$ is always followed by an agent A. This ensures that at any time the state of a model will be constituted of cooperations of agents;
- $D + D$ expresses the non deterministic choice between two processes;
- $M \underset{\mathcal{L}}{\bowtie} M$ expresses the horizontal cooperation between two independent processes on *the same* scale via the cooperation set $\mathcal{L}$, with $\mathcal{L} \subseteq Actions$;
- $M \underset{\mathcal{L}}{\mathbin{\rotatebox[origin=c]{90}{$\bowtie$}}} M$ expresses the vertical cooperation between two independent processes on *different* scales via the cooperation set $\mathcal{L}$, with $\mathcal{L} \subseteq Actions$;
- functions $Var(A)$ and $Val(A)$ must be defined for each agent A, with $Var(A) \in Names$, $Val(A) \in \mathbb{R}$ and $Names$ the set of parameter names.

Conventions for the notation of actions are as follows. Given a composed action $\mathcal{A}[\mathcal{E}]$, if $|\mathcal{A}| = 1$ or if $|\mathcal{E}| = 1$ then set delimiters can be omitted, e.g. if $\mathcal{A} = \{a\}$ then it can be written a. If $\mathcal{E} = \emptyset$ then the hook part of the composed action can be omitted completely, that is $\mathcal{A}[\mathcal{E}]$ can be written $\mathcal{A}$.

The semantics for process algebra with hooks is presented in Figure 2 as inference rules for the derivation of valid transitions. In particular, the $\underset{\mathcal{L}}{\mathbin{\rotatebox[origin=c]{90}{$\bowtie$}}}$ operator works as follows. Consider the inference rule **Vertical Synchronisation Left**.

Prefix

$$\frac{}{\mathcal{A}[\mathcal{E}].A \xrightarrow{\mathcal{A}[\mathcal{E}]} A}$$

Choice Left

$$\frac{D_1 \xrightarrow{\mathcal{A}[\mathcal{E}]} A}{D_1 + D_2 \xrightarrow{\mathcal{A}[\mathcal{E}]} A}$$

Choice Right

$$\frac{D_2 \xrightarrow{\mathcal{A}[\mathcal{E}]} A}{D_1 + D_2 \xrightarrow{\mathcal{A}[\mathcal{E}]} A}$$

Asynchronous Left

$$\frac{M_1 \xrightarrow{(\mathcal{A}[\mathcal{E}],\Gamma)} M_1'}{M_1 \underset{\mathcal{L}}{\bowtie} M_2 \xrightarrow{(\mathcal{A}[\mathcal{E}],\Gamma)} M_1' \underset{\mathcal{L}}{\bowtie} M_2} \text{ if } \mathcal{A} \cap \mathcal{L} = \emptyset$$

Asynchronous Right

$$\frac{M_2 \xrightarrow{(\mathcal{A}[\mathcal{E}],\Gamma)} M_2'}{M_1 \underset{\mathcal{L}}{\bowtie} M_2 \xrightarrow{(\mathcal{A}[\mathcal{E}],\Gamma)} M_1 \underset{\mathcal{L}}{\bowtie} M_2'} \text{ if } \mathcal{A} \cap \mathcal{L} = \emptyset$$

Layer Synchronisation

$$\frac{M_1 \xrightarrow{(\mathcal{A}[\mathcal{E}],\Gamma_1)} M_1' \quad M_2 \xrightarrow{(\mathcal{B}[\mathcal{F}],\Gamma_2)} M_2'}{M_1 \underset{\mathcal{L}}{\bowtie} M_2 \xrightarrow{(\mathcal{A}\cup\mathcal{B}[\mathcal{E}\cup\mathcal{F}],\Gamma_1\cup\Gamma_2)} M_1' \underset{\mathcal{L}}{\bowtie} M_2'} \text{ if } \mathcal{A} \cap \mathcal{B} \cap \mathcal{L} \neq \emptyset$$

Agent

$$\frac{D \xrightarrow{\mathcal{A}[\mathcal{E}]} A'}{A \xrightarrow{(\mathcal{A}[\mathcal{E}],\Gamma)} A'} \begin{array}{l} \text{if } A \triangleq D \\ \wedge\, \Gamma = \{(Var(A), Val(A))\} \end{array}$$

Vertical Asynchronous Left

$$\frac{M_1 \xrightarrow{(\mathcal{A}[\mathcal{E}],\Gamma)} M_1'}{M_1 \underset{\mathcal{L}}{\rotatebox[origin=c]{90}{$\bowtie$}} M_2 \xrightarrow{(\mathcal{A}[\mathcal{E}],\Gamma)} M_1' \underset{\mathcal{L}}{\rotatebox[origin=c]{90}{$\bowtie$}} M_2} \begin{array}{l} \text{if } \neg(M_2 \xrightarrow{(\mathcal{B}[\mathcal{F}],\Gamma')} M_2' \\ \text{with } \mathcal{B} \subseteq \mathcal{E} \cap \mathcal{L}) \\ \wedge\, \mathcal{A} \cap \mathcal{L} = \emptyset \end{array}$$

Vertical Asynchronous Right

$$\frac{M_2 \xrightarrow{(\mathcal{B}[\mathcal{F}],\Gamma)} M_2'}{M_1 \underset{\mathcal{L}}{\rotatebox[origin=c]{90}{$\bowtie$}} M_2 \xrightarrow{(\mathcal{B}[\mathcal{F}],\Gamma)} M_1 \underset{\mathcal{L}}{\rotatebox[origin=c]{90}{$\bowtie$}} M_2'} \begin{array}{l} \text{if } \neg(M_1 \xrightarrow{(\mathcal{A}[\mathcal{E}],\Gamma')} M_1' \\ \text{with } \mathcal{A} \subseteq \mathcal{F} \cap \mathcal{L}) \\ \wedge\, \mathcal{B} \cap \mathcal{L} = \emptyset \end{array}$$

Vertical Synchronisation Left

$$\frac{M_1 \xrightarrow{(\mathcal{A}[\mathcal{E}],\Gamma_1)} M_1' \quad M_2 \xrightarrow{(\mathcal{B}[\mathcal{F}],\Gamma_2)} M_2'}{M_1 \underset{\mathcal{L}}{\rotatebox[origin=c]{90}{$\bowtie$}} M_2 \xrightarrow{(\mathcal{A}\cup\mathcal{B}[(\mathcal{E}\setminus\mathcal{B})\cup\mathcal{F}],\Gamma_1)} M_1' \underset{\mathcal{L}}{\rotatebox[origin=c]{90}{$\bowtie$}} M_2'} \begin{array}{l} \text{if } \mathcal{B} \subseteq \mathcal{E} \cap \mathcal{L} \ \wedge\ \neg(M_2 \xrightarrow{(\mathcal{B}'[\mathcal{F}'],\Gamma_2')} M_2'' \\ \text{with } \mathcal{B}' \subseteq \mathcal{E} \cap \mathcal{L} \wedge |\mathcal{B}'| > |\mathcal{B}|) \end{array}$$

Vertical Synchronisation Right

$$\frac{M_1 \xrightarrow{(\mathcal{A}[\mathcal{E}],\Gamma_1)} M_1' \quad M_2 \xrightarrow{(\mathcal{B}[\mathcal{F}],\Gamma_2)} M_2'}{M_1 \underset{\mathcal{L}}{\rotatebox[origin=c]{90}{$\bowtie$}} M_2 \xrightarrow{(\mathcal{A}\cup\mathcal{B}[(\mathcal{F}\setminus\mathcal{A})\cup\mathcal{E}],\Gamma_2)} M_1' \underset{\mathcal{L}}{\rotatebox[origin=c]{90}{$\bowtie$}} M_2'} \begin{array}{l} \text{if } \mathcal{A} \subseteq \mathcal{F} \cap \mathcal{L} \ \wedge\ \neg(M_1 \xrightarrow{(\mathcal{A}'[\mathcal{E}'],\Gamma_1')} M_1'' \\ \text{with } \mathcal{A}' \subseteq \mathcal{F} \cap \mathcal{L} \wedge |\mathcal{A}'| > |\mathcal{A}|) \end{array}$$

Figure 2: Semantics of process algebra with hooks.

The synchronisation is between the set of hook actions on the left hand side ($\mathcal{E}$) and the set of layer actions on the right hand side ($\mathcal{B}$), via actions in the cooperation set $\mathcal{L}$. This means that some inter-scale actions in $\mathcal{E}$ communicate events to another scale. It may be that more than one transition from M_2 presents a set of layer actions $\mathcal{B}$ suitable. In this case, as explained more in details in [6], we give priority to transitions with largest $\mathcal{B}$ sets. Consider now the inference rule **Vertical Asynchronous Left**. In this case, we allow a single process to transition asynchronously only if no transitions from M_2 present a set of layer actions $\mathcal{B}$ suitable for synchronisation. Moreover, the rule blocks possible transitions if the set of layer actions $\mathcal{A}$ in such transitions contains actions in $\mathcal{L}$. Additional definitions are:

Definition 1. Activity. The pair $(\mathcal{A}[\mathcal{E}], \Gamma)$ such that $\mathcal{A}, \mathcal{E} \subseteq$ *Actions* and environment $\Gamma \subseteq Names \times \mathbb{R}$ is called an activity.

Definition 2. One step derivative. If $M \xrightarrow{(\mathcal{A}[\mathcal{E}],\Gamma)} M'$ then M is a one step derivative of M.

Definition 3. Derivative Set. The derivative set of a model process $M \in \mathbb{P}_m$ is denoted by $ds(M)$ and is defined as the smallest set of model processes such that:

- $M \in ds(M)$;
- if $M_i \in ds(M)$ and $M_i \xrightarrow{(\mathcal{A}[\mathcal{E}],\Gamma)} M_j$ then $M_j \in ds(M)$.

In the following, we define activities that a process can perform as *multi sets*, since their multiplicity may be significant to determine a correct rate for transitions (see section 3.3 and normalised transition rates).

Definition 4. Current moves of a definition process. The multi set of moves that $D \in \mathbb{P}_d$ can perform is denoted by $Moves(D)$ and is defined as:

- $Moves(nil) = \{\!|\,|\!\}$;
- $Moves(\mathcal{A}[\mathcal{E}].A) = \{\!|(\mathcal{A}[\mathcal{E}], A)|\!\}$;
- $Moves(D_1 + D_2) = Moves(D_1) \uplus Moves(D_2)$.

with $\{\!|\,|\!\}$ delimiting a multi set and $\uplus$ the union of multi sets.

Definition 5. Current moves of a model process. The multi set of moves that $M \in \mathbb{P}_m$ can perform is denoted by $Moves(M)$ and is defined as:

- $Moves(A) = \{\!|((\mathcal{A}[\mathcal{E}], \Gamma), A') \mid (\mathcal{A}[\mathcal{E}], A') \in Moves(D)$ $\wedge A \triangleq D \wedge \Gamma = \{(Var(A), Val(A))\}|\!\}$;
- $Moves(M_1 \underset{\mathcal{L}}{\bowtie} M_2) = \{\!|((\mathcal{A}[\mathcal{E}], \Gamma), M_1' \underset{\mathcal{L}}{\bowtie} M_2) \mid ((\mathcal{A}[\mathcal{E}], \Gamma), M_1') \in Moves(M_1) \wedge \mathcal{A} \cap \mathcal{L} = \emptyset|\!\}$ $\uplus\{\!|((\mathcal{A}[\mathcal{E}], \Gamma), M_1 \underset{\mathcal{L}}{\bowtie} M_2') \mid ((\mathcal{A}[\mathcal{E}], \Gamma), M_2') \in Moves(M_2)$ $\wedge \mathcal{A} \cap \mathcal{L} = \emptyset|\!\}$ $\uplus\{\!|((\mathcal{A}\cup\mathcal{B}[\mathcal{E}\cup\mathcal{F}], \Gamma_1\cup\Gamma_2), M_1' \underset{\mathcal{L}}{\bowtie} M_2') \mid ((\mathcal{A}[\mathcal{E}], \Gamma_1), M_1')$ $\in Moves(M_1) \wedge ((\mathcal{B}[\mathcal{F}], \Gamma_2), M_2') \in Moves(M_2) \wedge \mathcal{A} \cap \mathcal{B} \cap \mathcal{L} \neq \emptyset|\!\}$;
- $Moves(M_1 \underset{\mathcal{L}}{\rotatebox[origin=c]{90}{$\bowtie$}} M_2) = \{\!|((\mathcal{A}[\mathcal{E}], \Gamma), M_1' \underset{\mathcal{L}}{\rotatebox[origin=c]{90}{$\bowtie$}} M_2) \mid ((\mathcal{A}[\mathcal{E}], \Gamma), M_1') \in Moves(M_1) \wedge$ $\neg(\exists \mathcal{B}, \mathcal{F}, \Gamma', M_2'$ s.t. $((\mathcal{B}[\mathcal{F}], \Gamma'), M_2') \in Moves(M_2)$ $\wedge \mathcal{B} \subseteq \mathcal{E} \cap \mathcal{L}) \wedge\ \mathcal{A} \cap \mathcal{L} = \emptyset|\!\}$ $\uplus\{\!|((\mathcal{B}[\mathcal{F}], \Gamma), M_1 \underset{\mathcal{L}}{\rotatebox[origin=c]{90}{$\bowtie$}} M_2') \mid ((\mathcal{B}[\mathcal{F}], \Gamma), M_2') \in Moves(M_2)$

$\wedge\neg(\exists\mathcal{A},\mathcal{E},\Gamma',M_1'$ s.t. $((\mathcal{A}[\mathcal{E}],\Gamma'),M_1') \in Moves(M_1)$ $\wedge\mathcal{A}\subseteq\mathcal{F}\cap\mathcal{L}) \wedge\ \mathcal{B}\cap\mathcal{L}=\emptyset\}\!\!\}$

$\uplus\{\!\!\{((\mathcal{A}\cup\mathcal{B}[(\mathcal{E}\backslash\mathcal{B})\cup\mathcal{F}],\Gamma_1), M_1' \underset{\mathcal{L}}{\boxtimes} M_2') \mid ((\mathcal{A}[\mathcal{E}],\Gamma_1),M_1')$ $\in Moves(M_1) \wedge ((\mathcal{B}[\mathcal{F}],\Gamma_2),M_2') \in Moves(M_2) \wedge \mathcal{B} \subseteq$ $\mathcal{E} \cap \mathcal{L} \wedge \neg(\exists\mathcal{B}',\mathcal{F}',\Gamma_2',M_2''$ s.t. $((\mathcal{B}'[\mathcal{F}'],\Gamma_2'),M_2'') \in$ $Moves(M_2) \wedge \mathcal{B}' \subseteq \mathcal{E}\cap\mathcal{L} \wedge |\mathcal{B}'| > |\mathcal{B}|)\}\!\!\}$

$\uplus\{\!\!\{((\mathcal{A}\cup\mathcal{B}[(\mathcal{F}\backslash\mathcal{A})\cup\mathcal{E}],\Gamma_2), M_1' \underset{\mathcal{L}}{\boxtimes} M_2') \mid ((\mathcal{A}[\mathcal{E}],\Gamma_1),M_1')$ $\in Moves(M_1) \wedge ((\mathcal{B}[\mathcal{F}],\Gamma_2),M_2') \in Moves(M_2) \wedge \mathcal{A} \subseteq$ $\mathcal{F} \cap \mathcal{L} \wedge \neg(\exists\mathcal{A}',\mathcal{E}',\Gamma_1',M_1''$ s.t. $((\mathcal{A}'[\mathcal{E}'],\Gamma_1'),M_1'') \in$ $Moves(M_1) \wedge \mathcal{A}' \subseteq \mathcal{F}\cap\mathcal{L} \wedge |\mathcal{A}'| > |\mathcal{A}|)\}\!\!\}$

Definition 6. Current activities for model Processes. The multi set of activities that $M \in \mathbb{P}_m$ can perform is denoted by $Activities(M)$ and is defined as:

$$Activities(M) = \{\!\!\{(\mathcal{A}[\mathcal{E}],\Gamma) \mid ((\mathcal{A}[\mathcal{E}],\Gamma),M') \in Moves(M)\}\!\!\}$$

Definition 7. Activity set. The multi set of activities that a model process $M \in \mathbb{P}_m$ and its derivatives can perform is given by:

$$\overrightarrow{Activities}(M) = \biguplus_{M_i \in ds(M)} Activities(M_i)$$

Definition 8. Derivation graph. Given a model component $M \in \mathbb{P}_m$, the derivation graph $\mathcal{D}(M)$ is the labelled directed graph with:

- set of nodes $ds(M)$;
- multi set of transition labels $\overrightarrow{Activities}(M)$;
- multi set of labelled transitions $\rightarrow\subseteq ds(M)\times \overrightarrow{Activities}(M) \times ds(M)$. Given $M' \in ds(M)$, $(M',\mathcal{A}[\mathcal{E}],\Gamma,M'') \in\rightarrow$ iff $M' \xrightarrow{(\mathcal{A}[\mathcal{E}],\Gamma)} M''$, with the same multiplicity of $((\mathcal{A}[\mathcal{E}],\Gamma),M'')$ in $Moves(M')$.

3.2 Isomorphism

In this section we define isomorphism on PAH processes. Informally, two processes are isomorphic if they generate isomorphic derivation graphs. With this definition we propose equational laws for isomorphic PAH processes and show that isomorphism is a congruence.

Definition 9. Function apply. Given a function $f : \mathbb{P} \rightarrow \mathbb{P}$ and a multi set of filtered moves $MSet$, function *apply* applies f to $MSet$ in the following way:

$$apply(f)(MSet) = \{\!\!\{(a, f(P)) \mid (a,P) \in MSet\}\!\!\}$$

Definition 10. Model process isomorphism. A function $\mathcal{F}$: $ds(M_1) \rightarrow ds(M_2)$ is a model process isomorphism between M_1 and M_2 ($M_1, M_2 \in \mathbb{P}_m$), if $\mathcal{F}$ is injective and $\forall M_1' \in ds(M_1)$,

$$Moves(\mathcal{F}(M_1')) = apply(\mathcal{F})(Moves(M_1'))$$

Definition 11. Isomorphic model processes. Two model processes $M_1, M_2 \in \mathbb{P}_m$ are isomorphic, written $M_1 \equiv M_2$, if there is a model process isomorphism $\mathcal{F}$ between them such that $\mathcal{D}(\mathcal{F}(M_1)) = \mathcal{D}(M_2)$.

Definition 12. Isomorphic definition processes. Two definition processes $D_1, D_2 \in \mathbb{P}_d$ are isomorphic ($D_1 \equiv D_2$) iff there exists an injective function $\mathcal{F} : ds(D_1) \rightarrow ds(D_2)$ such that $\forall A \in ds(D_1)$, $A \equiv \mathcal{F}(A)$ and

$$Moves(D_2) = apply(\mathcal{F})(Moves(D_1))$$

Proposition 1. Equational laws for isomorphic PAH *processes.* The following laws can be proved using the stochastic operational semantics and the definition of model and definition process isomorphisms:

1. $D_1 + D_2 \equiv D_2 + D_1$;
2. $(D_1 + D_2) + D_3 \equiv D_1 + (D_2 + D_3)$;
3. $M_1 \underset{\mathcal{L}}{\bowtie} M_2 \equiv M_2 \underset{\mathcal{L}}{\bowtie} M_1$;
4. $(M_1 \underset{\mathcal{L}}{\bowtie} M_2) \underset{\mathcal{L}}{\bowtie} M_3 \equiv M_1 \underset{\mathcal{L}}{\bowtie} (M_2 \underset{\mathcal{L}}{\bowtie} M_3)$;
5. $M_1 \underset{\mathcal{L}}{\boxtimes} M_2 \equiv M_2 \underset{\mathcal{L}}{\boxtimes} M_1$;
6. $(M_1 \underset{\mathcal{L}}{\bowtie} M_2) \underset{\mathcal{K}}{\bowtie} M_3 \equiv M_1 \underset{\mathcal{L}}{\bowtie} (M_2 \underset{\mathcal{K}}{\bowtie} M_3)$, if $\forall(\mathcal{A}[\mathcal{E}],\Gamma) \in \overrightarrow{Activities}(M_1)$, $\forall(\mathcal{N}[\mathcal{H}],\Gamma'') \in \overrightarrow{Activities}(M_3)$, $\mathcal{N} \cap (\mathcal{L}\setminus\mathcal{K}) = \emptyset \wedge \mathcal{A} \cap (\mathcal{K}\setminus\mathcal{L}) = \emptyset$;

Proof. Appendix B. □

Proposition 2. Isomorphism as a Congruence. If $P_1, P_2 \in \mathbb{P}$ such that $P_1 \equiv P_2$, then

1. $\mathcal{A}[\mathcal{E}].P_1 \equiv \mathcal{A}[\mathcal{E}].P_2$, with P_1, P_2 agents
2. $P_1 + Q \equiv P_2 + Q$, with $P_1, P_2, Q \in \mathbb{P}_d$
3. $P_1 \underset{\mathcal{L}}{\bowtie} Q \equiv P_2 \underset{\mathcal{L}}{\bowtie} Q$, with $P_1, P_2, Q \in \mathbb{P}_m$
4. $P_1 \underset{\mathcal{L}}{\boxtimes} Q \equiv P_2 \underset{\mathcal{L}}{\boxtimes} Q$, with $P_1, P_2, Q \in \mathbb{P}_m$

Proof. Appendix B. □

3.3 Functional Rates and Rate Evaluation

Functional rates are arithmetical expression which allow the modeller to use existing kinetic laws used in biology to define the quantitative time evolution of the system. In order to do so, functional rates contain parameter names, the value of which depend on the current state of the system, or more precisely on the environment Γ in an activity $(\mathcal{A}[\mathcal{E}],\Gamma)$. Functional rates are associated with actions, so we introduce constraints that ensure that at most one functional rate is associated with a transition. The syntax of functional rates is given by:

$$f ::= k \mid i \mid f\ op_1\ f \mid op_2(f) \mid f^f$$
$$op_1 ::= + \mid - \mid * \mid / \qquad op_2 ::= exp \mid log \mid sin \mid cos$$

- $k \in \mathbb{R}$ and $i \in Names$, i.e. i is a parameter name;
- f is a functional rate, $f \in \mathbb{F}$.

The set $\mathbb{F}$ contains the functional rates defined in a PAH model, indexed by action names. For example, if $f_a \in \mathbb{F}$ then f_a is the functional rate associated with action a. The evaluation of functional rates follows the standard operational semantics of arithmetical expressions. Given an environment $\Gamma \subseteq Names \times \mathbb{R}$ and a functional rate f, f evaluates to $k \in \mathbb{R}$ iff $\Gamma \vdash f \rightarrow k$ is valid.

In order to ensure correct and unambiguous rate evaluation and to guarantee that congruence relations can be defined on PAH processes, we employ the following additional constraints:

- each functional rate $f_a \in \mathbb{F}$ is associated with a set of participants $p_a \subseteq Names$;
- an activity $(\mathcal{A}[\mathcal{E}], \Gamma)$ can be rated only if Γ contains exactly the variables in p_a and $\mathcal{A}$ contains exactly one action name a such that $f_a \in \mathbb{F}$. Such activity is called *closed*. An activity that is not closed is called *open*;
- only agents associated with variables in p_a can perform action a. This is to prevent an additional synchronisation via $\underset{\mathcal{L}}{\bowtie}$ changing a closed activity into an open activity, due to an increase of the size of Γ. Moreover, at any time only one agent can be associated with each variable;
- actions used as hook actions must not be associated with functional rates;
- if more than one transition from a certain state is associated with the same functional rate, the evaluated rate has to be normalised, i.e. it has to be divided by the number of such transitions.

The above constraints are formalised by the following definitions. With these we can define a *rated derivation graph.*

Definition 13. Well formed process algebra with hooks model. A process algebra with hooks model is well formed if and only if:

1. Given a model process as a cooperation of agents of the form
$$A_1 \circ A_2 \circ \cdots \circ A_n$$
then $\forall A_i, A_j$ if $i \neq j$ then $Var(A_j) \neq Var(A_j)$, where $\circ$ is either a vertical or horizontal cooperation;
2. Given the definition of an agent A as a choice of sequential actions of the form
$$A \triangleq \sum_i a_i.A_i$$
then $\forall A_i \; Var(A) = Var(A_i)$;
3. $\forall a$ s.t. $f_a \in \mathbb{F}$, $\forall A$ agents
$$\exists(\mathcal{A}[\mathcal{E}], \Gamma) \in \overrightarrow{Activities}(A) \text{ s.t. } a \in \mathcal{A} \cup \mathcal{E} \Leftrightarrow Var(A) \in p_a$$
Moreover, whenever $M_1 \underset{\mathcal{L}}{\bowtie} M_2$ then $\forall a$ s.t. $f_a \in \mathbb{F}$
$$a \in \mathcal{L} \Leftrightarrow \exists(\mathcal{A}[\mathcal{E}], \Gamma) \in \overrightarrow{Activities}(M_1),$$
$$(\mathcal{B}[\mathcal{F}], \Gamma') \in \overrightarrow{Activities}(M_2) \text{ s.t. } a \in \mathcal{A} \wedge a \in \mathcal{B}$$
4. hook actions are not associated with functional rates:
$$\forall \text{ agents } A \forall(\mathcal{A}[\mathcal{E}], \Gamma) \in \overrightarrow{Activities}(A),$$
$$\forall a \text{ s.t. } f_a \in \mathbb{F},\; a \notin \mathcal{E}$$
5. For all agents A defined as
$$A \triangleq \sum_i \mathcal{A}_i[\mathcal{H}_i].A_i$$
for all a s.t. $f_a \in \mathbb{F}$, if $a \in \mathcal{A}_i$ then $\mathcal{A}_i = \{a\}$.

Definition 13 ensures that whenever $M \xrightarrow{(\mathcal{A}[\mathcal{H}], \Gamma)} M'$ then either $\forall a$ s.t. $f_a \in \mathbb{F}$, $a \notin \mathcal{A}$ or $\exists! a$ s.t. $f_a \in \mathbb{F}$ and $a \in \mathcal{A}$. In other words, for every valid transition, the set of layer actions contains at most one action associated with a functional rate.

Definition 14. Function envVar. The function *envVar* extracts the set of variables in an environment $\Gamma \subseteq Names \times \mathbb{R}$:

$$envVar(\Gamma) = \{i \mid (i, k) \in \Gamma\}$$

Definition 15. Function activeActions. The function *activeActions* selects actions a such that a functional rate in the set $\mathbb{F}$ is associated with a, i.e. $f_a \in \mathbb{F}$, from an action set $\mathcal{A} \subseteq Actions$:

$$activeActions(\mathcal{A})_{\mathbb{F}} = \{a \mid a \in A \wedge f_a \in \mathbb{F}\}$$

Definition 16. Open activity. An open activity is an activity $(\mathcal{A}[\mathcal{E}], \Gamma)$ where at least one of the following conditions are true:

- the number of active actions in $\mathcal{A}$ is not 1, i.e. $|activeActions(\mathcal{A})_{\mathbb{F}}| \neq 1$;
- if $|activeActions(\mathcal{A})_{\mathbb{F}}| = 1$ and $a \in activeActions(\mathcal{A})_{\mathbb{F}}$, Γ does not contain the *exact* variables present in the participant set p_a, i.e. $p_a \neq envVar(\Gamma)$.

Definition 17. Function openActivities. The function *openActivities* selects open activities from a set of activities $A \subseteq 2^{Actions} \times 2^{Actions} \times 2^{Names \times \mathbb{R}}$:

$$openActivities(A) = \{\!|(\mathcal{A}[\mathcal{E}], \Gamma) \mid (\mathcal{A}[\mathcal{E}], \Gamma) \in A \wedge$$
$$(|activeActions(\mathcal{A})_{\mathbb{F}}| \neq 1 \vee (activeActions(\mathcal{A})_{\mathbb{F}} = \{a\} \wedge$$
$$p_a \neq envVar(\Gamma)))|\!\}$$

Definition 18. Current open activities. Given a model process $M \in \mathbb{P}_m$, the multi set of open activities that P can perform is defined as:

$$OpenAct(M) = openActivities(Activities(M))$$

Definition 19. Open activity set. The multi set of all open activities that a model process $M \in \mathbb{P}_m$ can perform is given by:

$$\overrightarrow{OpenAct}(M) = openActivities(\overrightarrow{Activities}(M))$$

Definition 20. Closed activity. A closed activity is an activity $(\mathcal{A}[\mathcal{E}],\Gamma)$ where:

- $|activeActions(\mathcal{A})_{\mathbb{F}}| = 1$, $a \in activeActions(\mathcal{A})_{\mathbb{F}}$ and Γ contains the exact variables present in the participant set p_a, i.e. $p_a = envVar(\Gamma)$.

Definition 21. Function closedActivities*.* The function *closedActivities* selects closed activities from a set of activities $A \subseteq 2^{Actions} \times 2^{Actions} \times 2^{Names\times\mathbb{R}}$:

$$closedActivities(A) = (A \setminus openActivities(A))$$

Definition 22. Current closed activities. Given a model process $M \in \mathbb{P}_m$, the multi set of closed activities that M can perform is defined as:

$$ClosedAct(M) = closedActivities(Activities(M))$$

Definition 23. Closed activity set. The multi set of all closed activities that a model process $M \in \mathbb{P}_m$ can perform is given by:

$$\overrightarrow{ClosedAct}(M) = closedActivities(\overrightarrow{Activities}(M))$$

Definition 24. Rated activity. The pair $(\mathcal{A}[\mathcal{E}], r)$ such that $\mathcal{A},\mathcal{E} \subseteq Actions$ and $r \in \mathbb{R}_{>0}$ is called a rated activity.

Definition 25. Function rateActivities*.* Given an environment $\Gamma \subseteq Names \times \mathbb{R}$, *rateActivities* converts a set of activities $A \subseteq 2^{Actions} \times 2^{Actions} \times 2^{Names\times\mathbb{R}}$ into a set of rated activities $B \subseteq 2^{Actions} \times 2^{Actions} \times \mathbb{R}$:

$$\begin{gathered} rateActivities(\Gamma)(A) = \{\!|(\mathcal{A}[\mathcal{E}], r) \mid (\mathcal{A}[\mathcal{E}],\Gamma') \in A \wedge \\ \{a\} = activeActions(\mathcal{A})_{\mathbb{F}} \wedge\ \Gamma \cup \Gamma' \vdash f_a \to k \wedge \\ r_a = k/\pi(A,(a,\Gamma')) \wedge\ f_a \in \mathbb{F}|\!\} \end{gathered}$$

where $\pi(A,(a,\Gamma'))$ returns the number of occurrences of $(\mathcal{A}[\mathcal{E}],\Gamma')$ in the multi set A such that $activeActions(\mathcal{A})_{\mathbb{F}} = \{a\}$.

Definition 26. Current rated activities. Given a model process $M \in \mathbb{P}_m$ and an environment $\Gamma \subseteq Names \times \mathbb{R}$, the multi set of rated activities that M can perform is defined as:

$$RatedAct(M)_\Gamma = rateActivities(\Gamma)(ClosedAct(M))$$

Definition 27. Rated activity set. Given an environment $\Gamma \subseteq Names \times \mathbb{R}$, the multi set of rated activities that a model process $M \in \mathbb{P}_m$ and its derivatives can perform is given by:

$$\overrightarrow{RatedAct}(M)_\Gamma = rateActivities(\Gamma)(\overrightarrow{ClosedAct}(M))$$

Definition 28. Rated moves of a model process. Given a model process $M \in \mathbb{P}_m$ and an environment $\Gamma \subseteq Names\times\mathbb{R}$, the multi set of rated moves of M, denoted $RatedMoves(M)_\Gamma$, is defined as:

$$\begin{gathered} RatedMoves(M)_\Gamma = \{\!|((\mathcal{A}[\mathcal{E}], r), M') \mid ((\mathcal{A}[\mathcal{E}],\Gamma'), M') \in \\ Moves(M') \wedge (\mathcal{A}[\mathcal{E}],\Gamma') \in ClosedAct(M) \wedge \{\!|((\mathcal{A}[\mathcal{E}],k))|\!\} \\ = rateActivities(\Gamma)(\{\!|(\mathcal{A}[\mathcal{E}],\Gamma')|\!\}) \wedge activeActions(\mathcal{A})_{\mathbb{F}} \\ = a \wedge r = k/\pi(ClosedAct(M),(a,\Gamma'))|\!\} \end{gathered}$$

Definition 29. Rated transitions. Given $M \in \mathbb{P}_m$ and $\Gamma \subseteq Names \times \mathbb{R}$, $M \xrightarrow{(\mathcal{A}[\mathcal{E}],r)}_{\Gamma} M'$ is a valid rated transition iff $M \xrightarrow{(\mathcal{A}[\mathcal{E}],\Gamma')} M'$, $(\mathcal{A}[\mathcal{E}],\Gamma') \in ClosedAct(M)$, $\{\!|((\mathcal{A}[\mathcal{E}],k))|\!\} = rateActivities(\Gamma)(\{\!|(\mathcal{A}[\mathcal{E}],\Gamma')|\!\})$, $activeActions(\mathcal{A})_{\mathbb{F}} = \{a\}$ and $r = k/\pi(ClosedAct(M),(a,\Gamma'))$.

Definition 30. Rated derivation graph. Given a model process $M \in \mathbb{P}_m$ and an environment $\Gamma \subseteq Names \times \mathbb{R}$, the rated derivation graph $\mathcal{D}_r(M)_\Gamma$ is the labelled directed graph with:

- set of nodes $ds(M)$;
- multi set of transition labels $\overrightarrow{RatedAct}(M)_\Gamma$;
- multi set of labelled transitions $\to_r \subseteq ds(M) \times \overrightarrow{RatedAct}(M)_\Gamma \times ds(M)$. Given $M' \in ds(M)$, $(M',\mathcal{A}[\mathcal{E}], r_a, M'') \in \to_r$ iff $M' \xrightarrow{(\mathcal{A}[\mathcal{E}],r_a)}_{\Gamma} M''$, with the same multiplicity as $((\mathcal{A}[\mathcal{E}], r_a), M'')$ in $RatedMoves(M')_\Gamma$;
- multi set of labelled transitions $\to_o \subseteq ds(M) \times \overrightarrow{OpenAct}(M) \times ds(M)$. Given $M' \in ds(M)$, $(M',\mathcal{A}[\mathcal{E}],\Gamma', M'') \in \to_o$ iff $M' \xrightarrow{(\mathcal{A}[\mathcal{E}],\Gamma')} M''$ and $(\mathcal{A}[\mathcal{E}],\Gamma') \in OpenAct(M)$, with the same multiplicity of $((\mathcal{A}[\mathcal{E}],\Gamma'), M'')$ in $Moves(M')$.

The definition of a rated derivation graph completes the first part of the paper, where we present the foundations of the theory of process algebra with hooks. From the following section we illustrate how the algebra can be used in practice.

4. PARAMETRIC PAH

In this section we introduce the syntax of a parametric version of PAH.

$$\begin{aligned} D ::= &\quad nil \mid \mathcal{L}'[\mathcal{L}''].A(exp,\ldots,exp) \mid D + D \mid \\ &\quad \text{if } bexp \text{ then } D \text{ else } D \\ M ::= &\quad A(k,\ldots,k) \mid M \underset{\mathcal{L}}{\bowtie} M \mid M \underset{\mathcal{L}}{\rotatebox{90}{$\bowtie$}} M \\ exp ::= &\quad k \mid i \mid exp + exp \mid exp - exp \mid exp/exp \mid exp * exp \\ bexp ::= &\quad exp = exp \mid exp < exp \mid bexp \wedge bexp \mid \\ &\quad bexp \vee bexp \mid \neg bexp \mid true \mid false \end{aligned}$$

- actions have the form $a(exp,\ldots,exp)$, where $exp,\ldots,exp$ is a list of expressions;
- $k \in \mathbb{R}$ and i is a parameter name, i.e. $i \in Names$;
- a definition process can also be an if-then-else construct: if *bexp* then D else D;
- agent definitions have now the form $A(i_1,\ldots,i_n) \triangleq D$, where $i_1,\ldots,i_n$ is a list of parameter names;
- the evaluation of the expressions is performed when inference rule **Agent** is applied;
- the definition of functional rates is also parametric.

The semantics of parametric PAH is reported in Appendix C. This syntax is used in the implementation of the interpreter of PAH models, as it allows the modeller to define agents parametric in, for example, their position on a grid or which level of concentration they represent. The current implementation derives transitions from a model process and assigns rates according to the theory presented above. Moreover, a simple simulator produces traces of stochastic simulations sampling transitions using a *race condition* between the available rated transitions. More in details, the rates are used to determine a time delay for the next transition to occur and which state is reached after the transition. In the next section, we use parametric PAH and the mentioned implementation to model and analyse a multi-scale scenario of tissue growth.

5. MODEL OF TISSUE GROWTH

In this section we propose a simple model of tissue growth. We define it as a multi-scale model because two scales are present, tissue and biochemistry, and the two scales need to interact. In particular, we show how one can define a model where growth and death of tissue depend on the local concentration of biochemical species. Here we give the idea of the model, while leaving the complete specification in Appendix D. Moreover, we present preliminary results based on simulations of the model in Figures 3 and 4.

At the tissue scale we consider an area divided into regions of the same size and shape. We consider a grid of 10 times 10 regions, each region denoted by the parameter name $R(i,j)$. Each region can be empty (agent $E(i,j)$) or contain tissue. There are four types of tissue: tissue that can neither grow nor die (agent $T(i,j)$); tissue that can grow, but not die ($Tm(i,j)$); tissue that can die but not grow ($Ta(i,j)$); tissue that can both grow and die ($Tam(i,j)$). Tissue processes change between these four agents depending on the configuration of the biochemical scale. The event of growth is represented by action $growth(i,j,i2,j2)$ which is performed by a tissue agent in region $R(i,j)$ in synchronisation with an adjacent empty space $E(i2,j2)$ in region $R(i2,j2)$. Two regions are considered adjacent if they share an edge. If no adjacent region is empty, growth is inhibited. The event of tissue death is represented by action $death(i,j)$. We assume that actions $growth(i,j,i2,j2)$ and $death(i,j)$ have constant rates k_{growth} and k_{death}.

The biochemical scale consists of biochemical species A, B and C, present in all regions. The concentration of each species varies between a concentration level of 0 and 10 (parameter $maxLevels = 10$). In particular, we use agent $A(i,j,w)$ to denote that species A in region $R(i,j)$ presents concentration level w. Analogously for species B and C. Concentration level of the three species can change because of the following local biochemical reactions (and associated velocities):

$$\begin{array}{llll} R_1 : \mathrm{A} + \mathrm{B} \rightarrow \mathrm{C} & & v_1 = k_1[A][B] & \\ R_2 : \rightarrow \mathrm{A} & v_2 = k_2 & R_3 : \rightarrow \mathrm{B} & v_3 = k_3 \\ R_4 : \mathrm{C} \rightarrow & v_4 = k_4[C] & R_5 : \mathrm{B} \rightarrow \mathrm{B'} & v_5 = k_5[B] \end{array}$$

Where R_5 is the transport of concentration of B from a compartment to an adjacent compartment. The following constraints, which require communication *between* scales, must hold:

- tissue can grow if and only if the concentration level of A in the same region is 5 or more. Actions $growthon(i,j)$ and $growthoff(i,j)$ are used as hook actions to indicate that a threshold has passed at the biochemistry. Tissue processes can synchronise with these hook actions and change accordingly;
- tissue can die if and only if the concentration level of C in the same region is 5 or more (parameter $thr = 5$). Actions $deathon(i,j)$ and $deathoff(i,j)$ are used as hook actions to indicate that a threshold has passed;
- a region is empty if and only if there is no biochemistry. To represent the absence of biochemistry we use processes $NA(i,j)$, $NB(i,j)$ and $NC(i,j)$. Actions $bioon(i,j)$ and $biooff(i,j)$ work across scales and ensure this is the case.

Consider for example the definition of agents $C(i,j,w)$ and $T(i,j)$:

```
$C(i,j,w) \triangleq biooff(i,j).NC(i,j)$
   $+($
      if $w < maxLevels$ then
         if $w == (thr-1)$ then
            $r1(i,j)[deathon(i,j)].C(i,j,w+1)$
         else $r1(i,j).C(i,j,w+1)$
      else $nil)$
   $+($
      if $w > 0$ then
         if $w == thr$ then
            $r4(i,j)[deathoff(i,j)].C(i,j,w-1)$
         else $r4(i,j).C(i,j,w-1)$
      else $nil)$
```

$$\boxed{T(i,j) \triangleq growthon(i,j).Tm(i,j) + deathon(i,j).Ta(i,j)}$$

In a region $R(i,j)$, if the concentration level of C (w) is below its maximum and A and B are available then C can participate to reaction R_1, represented by action $r1(i,j)$. In particular, if w is equal to 4 ($w == thr - 1$), then action $r1(i,j)$ carries also hook action $deathon(i,j)$, which in turn could synchronise with the tissue scale, bringing $T(i,j)$ to $Ta(i,j)$. Without the use of a parametric version of PAH, 100 definitions of $T(i,j)$ and 1000 definitions of $C(i,j,w)$ processes would have been necessary, one for each region and level of concentration to model.

The complete definition of the model along with parameter values can be found in Appendix D. The initial state is a grid composed of agents $E(i,j)$ in all regions with the exception of $R(6,6)$, where agent $Tm(6,6)$ is used. At the biochemical scale, agents $NA(i,j)$, $NB(i,j)$ and $NC(i,j)$ are used with the exception of $A(6,6,5)$, $B(6,6,0)$ and $C(6,6,0)$. In terms of model processes, the initial state consists of a vertical synchronisation between a model process representing the entire biochemical scale and the model process representing the tissue scale, $Biochem \underset{\mathcal{H}}{\bowtie} Tissue$, with cooperation set $\mathcal{H}$ containing the list of all hook actions used in the model.

Examples of simulations of the model are shown in Figure 3. The images of the grid are constructed processing the model process relative to the tissue scale resulted from the simulations. A region is white when an agent of the type $E(i,j)$ is found, black otherwise (i.e. tissue of some kind is found).

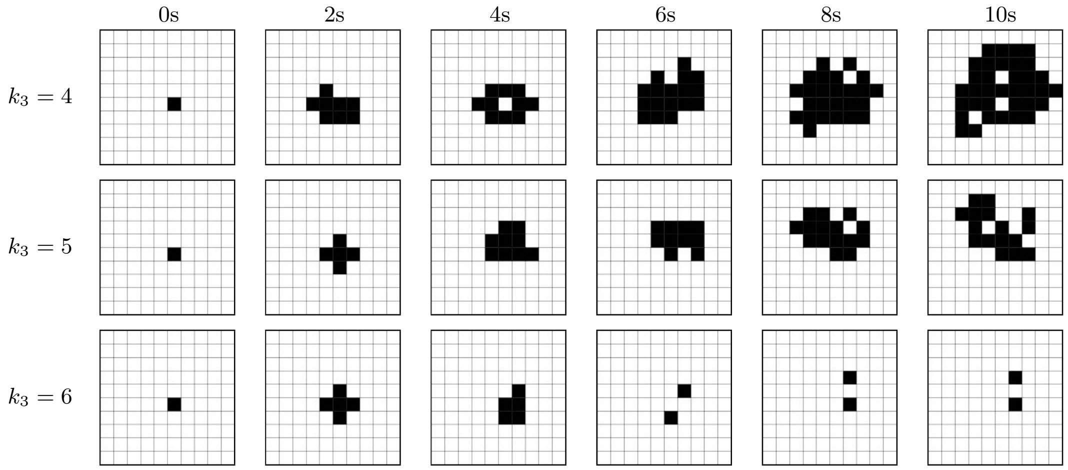

Figure 3: Three sample runs with k_3 equal to 4, 5 and 6 $Molar/s$. Black squares represent regions containing tissue.

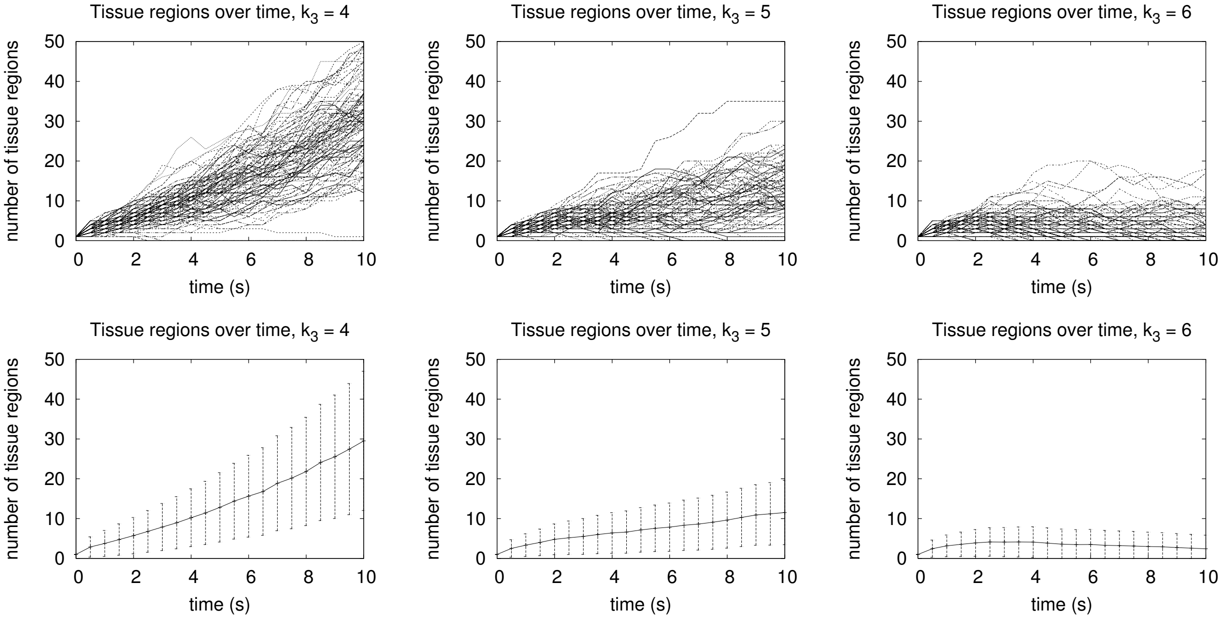

Figure 4: Number of tissue regions with parameter k_3 equal to 4, 5 and 6 $Molar/s$, with 100 simulations for each configuration. In the top row, all 100 simulations are shown, while in the bottom row average and standard deviation of the same runs.

As an example of possible analysis of the system, we focus on the role the production of species B has on tissue growth and death. Although B does not regulate tissue processes directly, it is involved along with A and C in reaction R_1. Intuitively, if the concentration of B is low, A is not consumed and growth becomes likely, while C is not produced and tissue death becomes unlikely. The parameter which regulates the production of B is k_3. Thus, we observed the behaviour of the system using three different values for k_3, 4, 5 and 6 M/s. In particular, we performed 100 simulation runs for each configuration. The results are shown in Figure 4, where one can see that increasing the production rate of B the growth/death ratio decreases.

We have seen that with PAH we have clearly separated processes on different scales. This ensures that one of the two scales, for example biochemistry, can be modified or substituted with limited or no change at the tissue scale. Moreover, the vertical synchronisation indicates which actions work across scales. Finally, we have seen that the parametric syntax reduces the redundancy at model definition.

6. RELATED WORK

Other process algebras have been defined to model biological interactions. Some of them are Bio-PEPA [5], which focusses on modelling the biochemical scale using multi-way synchronisation; BetaBinders [11], which models interactions between entities using interfaces on boxes, which in turn contain processes that define their internal behaviour; Brane Calculi [3], which models changes in hierarchical compartmental structures. In contrast with the mentioned approaches, we do not yet support the definition of compartments as hierarchical structures delimited by membranes. In this regard, PAH is more low level and focusses on the definition of hook actions and vertical cooperation, which allow the modelling of interrupt like events happening across scales. In this sense, PAH is more similar to a process algebra with priorities of actions, such as EMPA [1], with the additional ability of dealing with multiple interrupts at the same time using action sets. This ensures one can define dependencies between precesses in a way amenable to multi-scale scenarios, as we have discussed in [6]. Moreover, our approach ensures we can define equivalences on filtered PAH models. Another related approach worth mentioning is two-dimensional cellular automata [10], because of its ability to represent behaviour on a two dimensional grid. This approach has been used widely to represent biological systems such as tumour growth, or pattern formation, often with great efficiency. However, cellular automata does not provide a formal framework for the comparison and substitution of parts of models, such as congruences in process algebra. Moreover, our approach provides compositionality of biological scales.

7. CONCLUSIONS AND FUTURE WORK

In this paper we have discussed both theoretical and practical aspects of process algebra with hooks (PAH). We introduced process algebra with hooks as a process algebra designed for multi-scale modelling of biological systems. The algebra presents a new vertical cooperation operator ($\underset{\mathcal{L}}{\rotatebox{90}{$\bowtie$}}$), that separates processes on different scales, and composed actions, that allow synchronisations on multiple actions within and between scales at the same time. Moreover, quantitative analysis is possible associating actions with functional rates. Functional rates combined with the use of multi-way synchronisation ensure that standard kinetic laws can be used to define the rates of the biochemical events. On the practical side, we defined a parametric version of PAH, which allows the writing of more compact model definitions and that has been used in the implementation of the interpreter of the language. This version has been used to model a scenario of tissue growth, where growth and death of tissue depend on the local concentration of biochemical species. With this model, we have illustrated how a large process algebra model can be written in a compact and parametric way and how the two scales in the model can be separated by the vertical cooperation. Finally, we have seen examples of simulations and analysis.

Additional results, not presented in this paper, are the definition of congruences between rated and filtered PAH processes. The idea is that one could focus on a specific scale by filtering and thus removing action names from rated transitions. Only action names that represent events of a specified scale will be kept on the transition. Finally, one can formulate equivalence relations which ensure that two models have the same behaviour at a specified scale, despite other scales might be not equivalent. What we consider the logical follow up is the investigation of approximate equivalences to estimate a distance between the behaviour of two models.

8. ACKNOWLEDGMENTS

Andrea Degasperi is supported by a Lord Kelvin/Adam Smith Scholarship of the University of Glasgow and by the EPSRC funded SIGNAL project.

9. REFERENCES

[1] M. Bernardo. Extended Markovian Process Algebra. *Lecture Notes in Computer Science*, 1119:315–330, 1996.

[2] M. Calder, S. Gilmore, and J. Hillston. Modelling the influence of RKIP on the ERK signalling pathway using the stochastic process algebra PEPA. *Lecture Notes in Computer Science*, 4230:1–23, 2006.

[3] L. Cardelli. Brane calculi. *Lecture Notes in Bioinformatics*, 3082:257–278, 2005.

[4] F. Ciocchetta, A. Degasperi, J. Hillston, and M. Calder. Some investigations concerning the CTMC and the ODE model derived from Bio-PEPA. *Electronic Notes in Theoretical Computer Science*, 229(1):145–163, 2009.

[5] F. Ciocchetta and J. Hillston. Bio-PEPA: A framework for the modelling and analysis of biological systems. *Theoretical Computer Science*, 410:3065–3084, 2009.

[6] A. Degasperi and M. Calder. Process algebra with hooks for models of pattern formation. *CS2Bio 2010, Electronic Notes in Theoretical Computer Science*, 268:31–47, 2010.

[7] C. A. R. Hoare. *Communicating Sequential Processes.* Prentice-Hall, Inc., 1985.

[8] R. Milner. *Communication and Concurrency.* Prentice-Hall, Inc., 1989.

[9] D. Noble. *The Music of Life: Biology Beyond the Genome.* Oxford University Press, 2006.

[10] N. H. Packard and S. Wolfram. Two-Dimensional

Cellular Automata. *Journal of Statistical Physics*, 38:901–946, 1985.

[11] C. Priami and P. Quaglia. Beta-binders for biological interactions. *Lecture Notes in Computer Science*, 3082:20–33, 2005.

[12] I. H. Segel. *Enzyme Kinetics: Behaviour and Analysis of Rapid Equilibrium and Steady-State Enzyme Systems.* Wiley-Interscience, New York, 1993.

APPENDIX

A. FORMULATING FUNCTIONAL RATES FROM KINETIC LAWS

Consider an action a, the velocity $v(\mathbf{x})$ of the biochemical reaction R_a associated with a, S_i $(i = 1, ..., n)$ species involved in the reaction and $k_i \in \mathbb{Z}$ as their stoichiometry in the reaction R_a. The vector $\mathbf{x} \in \mathbb{R}^n$ contains the concentration of the species involved in the reaction and is used to evaluate the velocity. We use $[\mathrm{S}_i]$ to indicate the concentration of S_i, $\langle \mathrm{S}_i \rangle$ to indicate the current level of concentration of S_i, and h as the concentration represented by one level of concentration. The variation in time of $[\mathrm{S}_i]$ is given by the following differential equation:

$$\frac{\delta[\mathrm{S}_i]}{\delta t} = k_i v([S_1], ..., [S_n])$$

Consider $\Delta\langle \mathrm{S}_i \rangle = k_i$ as the change in number of levels that has to be applied to the current level of concentration of S_i when a is triggered. Moreover, consider v' Substituting $\delta[\mathrm{S}_i]$ with $\Delta\langle \mathrm{S}_i \rangle h$ and substituting the $[S_i]$ with their approximation $\langle \mathrm{S}_i \rangle h$ we obtain:

$$\begin{aligned} \frac{\delta[\mathrm{S}_i]}{\delta t} \approx \frac{\Delta\langle \mathrm{S}_i \rangle h}{\Delta t} &= k_i v(\langle \mathrm{S}_1 \rangle h, ..., \langle \mathrm{S}_n \rangle h) \\ \Longrightarrow \frac{1}{\Delta t} &= \frac{v(\langle \mathrm{S}_1 \rangle h, ..., \langle \mathrm{S}_n \rangle h)}{h} \end{aligned}$$

If we consider Δt as the average of the exponential distribution of the time necessary for a to happen, then $1/\Delta t$ can be used as the rate for such a distribution. For more details see [4].

B. PROOFS

Proof. *Proof of equational laws for isomorphic* PAH *processes.* We prove each law in turn:

1. $Moves(D_1 + D_2) = Moves(D_1) \uplus Moves(D_2) = Moves(D_2 + D_1)$ with $\mathcal{F}$ the identity function $id : \mathbb{P}_m \rightarrow \mathbb{P}_m$.
2. proof analogous to 1.
3. We choose model process isomorphism $\mathcal{F}$ as

$$\begin{gathered} \forall M_1' \underset{\mathcal{L}}{\Join} M_2' \in ds(M_1 \underset{\mathcal{L}}{\Join} M_2), \\ \mathcal{F}(M_1' \underset{\mathcal{L}}{\Join} M_2') = M_2' \underset{\mathcal{L}}{\Join} M_1' \end{gathered}$$

with $M_1' \in ds(M_1)$ and $M_2' \in ds(M_2)$. Clearly, because of the symmetry of operator $\underset{\mathcal{L}}{\Join}$,

$$Moves(\mathcal{F}(M_1' \underset{\mathcal{L}}{\Join} M_2')) = apply(\mathcal{F})(Moves(M_1' \underset{\mathcal{L}}{\Join} M_2'))$$

4. proof analogous to 3.
5. We choose model process isomorphism $\mathcal{F}$ as

$$\begin{gathered} \forall M_1' \underset{\mathcal{L}}{\rotatebox{90}{$\Join$}} M_2' \in ds(M_1 \underset{\mathcal{L}}{\rotatebox{90}{$\Join$}} M_2), \\ \mathcal{F}(M_1' \underset{\mathcal{L}}{\rotatebox{90}{$\Join$}} M_2') = M_2' \underset{\mathcal{L}}{\rotatebox{90}{$\Join$}} M_1' \end{gathered}$$

with $M_1' \in ds(M_1)$ and $M_2' \in ds(M_2)$. Clearly, because of the symmetry of operator $\underset{\mathcal{L}}{\rotatebox{90}{$\Join$}}$,

$$Moves(\mathcal{F}(M_1' \underset{\mathcal{L}}{\rotatebox{90}{$\Join$}} M_2')) = apply(\mathcal{F})(Moves(M_1' \underset{\mathcal{L}}{\rotatebox{90}{$\Join$}} M_2'))$$

6. We choose model process isomorphism $\mathcal{F}$ as

$$\begin{gathered} \forall (M_1' \underset{\mathcal{L}}{\Join} M_2') \underset{\mathcal{K}}{\Join} M_3' \in ds((M_1 \underset{\mathcal{L}}{\Join} M_2) \underset{\mathcal{K}}{\Join} M_3), \\ \mathcal{F}((M_1' \underset{\mathcal{L}}{\Join} M_2') \underset{\mathcal{K}}{\Join} M_3') = M_1' \underset{\mathcal{L}}{\Join} (M_2' \underset{\mathcal{K}}{\Join} M_3') \end{gathered}$$

with $M_1' \in ds(M_1)$, $M_2' \in ds(M_2)$ and $M_3' \in ds(M_3)$. Using the additional conditions of 6. we have

$$\begin{gathered} Moves(\mathcal{F}((M_1' \underset{\mathcal{L}}{\Join} M_2') \underset{\mathcal{K}}{\Join} M_3')) \\ = apply(\mathcal{F})((Moves(M_1' \underset{\mathcal{L}}{\Join} M_2') \underset{\mathcal{K}}{\Join} M_3')) \end{gathered}$$

□

Proof. *Proof of isomorphism as congruence.* We prove each case in turn:

1. $\mathcal{A}[\mathcal{E}].P_1$ and $\mathcal{A}[\mathcal{E}].P_2$ are definition processes. Because $P_1 \equiv P_2$ there exists model process isomorphism $\mathcal{F}$ between them such that $\mathcal{F}(P_1) = P_2$ and $P_1 \equiv \mathcal{F}(P_1)$. Clearly, we have

$$Moves(\mathcal{A}[\mathcal{E}].P_2) = apply(\mathcal{F})(Moves(\mathcal{A}[\mathcal{E}].P_1))$$

2. From the assumptions we know that $\exists \mathcal{F} : ds(P_1) \rightarrow ds(P_2)$ injective such that $\mathcal{F}(P_1) = P_2$ and $\forall A \in ds(P_1)$, $A \equiv \mathcal{F}(A)$ and

$$Moves(P_2) = apply(\mathcal{F})(Moves(P_1))$$

Thus, we need $\mathcal{G} : ds(P_1 + Q) \rightarrow ds(P_2 + Q)$ injective such that $\forall A \in ds(P_1 + Q)$, $A \equiv \mathcal{F}(A)$ and

$$Moves(P_2 + Q) = apply(\mathcal{G})(Moves(P_1 + Q))$$

We define $\mathcal{G}$ as:

$$\mathcal{G}(A) = \begin{cases} A, & \text{if } Q \xrightarrow{a} A \\ \mathcal{F}(A), & \text{if } P_1 \xrightarrow{a} A \end{cases}$$

Both cases of $\mathcal{G}$ ensure that $G(A) \equiv A$. Finally:

$$\begin{gathered} Moves(P_2 + Q) = Moves(P_2) \uplus Moves(Q) \\ = apply(\mathcal{F})(Moves(P_1)) \uplus apply(id)(Moves(Q)) \\ = apply(\mathcal{G})(Moves(P_1 + Q)) \end{gathered}$$

3. We know there is a model process isomorphism $\mathcal{F}$ between P_1 and P_2. Each element of $ds(P_1 \underset{\mathcal{L}}{\Join} Q)$ has the form $P_1' \underset{\mathcal{L}}{\Join} Q'$. We define a model process isomorphism $\mathcal{G}$ as: $\forall P_1' \underset{\mathcal{L}}{\Join} Q' \in ds(P_1 \underset{\mathcal{L}}{\Join} Q)$, with $P_1' \in ds(P_1)$ and $Q' \in ds(Q)$,

$$\mathcal{G}(P_1' \underset{\mathcal{L}}{\Join} Q') = \mathcal{F}(P_1') \underset{\mathcal{L}}{\Join} Q'$$

$\mathcal{G}$ is a model process isomorphism because $\mathcal{F}$ is a model process isomorphism. In fact:

$$
\begin{aligned}
&Moves(\mathcal{G}(P_1' \underset{\mathcal{L}}{\bowtie} Q')) = Moves(\mathcal{F}(P_1') \underset{\mathcal{L}}{\bowtie} Q')\\
&= \{\!\!|((\mathcal{A}[\mathcal{E}],\Gamma), R \underset{\mathcal{L}}{\bowtie} Q') \mid ((\mathcal{A}[\mathcal{E}],\Gamma), R) \in\\
&\quad Moves(\mathcal{F}(P_1')) \wedge \mathcal{A} \cap \mathcal{L} = \emptyset|\!\!\}\\
&\quad \uplus\{\!\!|((\mathcal{A}[\mathcal{E}],\Gamma), \mathcal{F}(P_1') \underset{\mathcal{L}}{\bowtie} Q'') \mid ((\mathcal{A}[\mathcal{E}],\Gamma), Q'') \in\\
&\quad Moves(Q') \wedge \mathcal{A} \cap \mathcal{L} = \emptyset|\!\!\}\\
&\quad \uplus\{\!\!|((\mathcal{A}[\mathcal{E}],\Gamma), R \underset{\mathcal{L}}{\bowtie} Q'')) \mid ((\mathcal{A}_1[\mathcal{E}_1],\Gamma_1), R) \in\\
&\quad Moves(\mathcal{F}(P_1')) \wedge ((\mathcal{A}_2[\mathcal{E}_2],\Gamma_2), Q'') \in Moves(Q')\\
&\quad \wedge \mathcal{A}_1 \cap \mathcal{A}_2 \cap \mathcal{L} \neq \emptyset \wedge \mathcal{A} = \mathcal{A}_1 \cup \mathcal{A}_2 \wedge \mathcal{E} = \mathcal{E}_1 \cup \mathcal{E}_2 \wedge\\
&\quad \Gamma = \Gamma_1 \cup \Gamma_2|\!\!\}\\
&= \{\!\!|((\mathcal{A}[\mathcal{E}],\Gamma), \mathcal{F}(P_1'') \underset{\mathcal{L}}{\bowtie} Q') \mid ((\mathcal{A}[\mathcal{E}],\Gamma), \mathcal{F}(P_1'')) \in\\
&\quad apply(\mathcal{F})(Moves(P_1')) \wedge \mathcal{A} \cap \mathcal{L} = \emptyset|\!\!\}\\
&\quad \uplus\{\!\!|((\mathcal{A}[\mathcal{E}],\Gamma), \mathcal{F}(P_1') \underset{\mathcal{L}}{\bowtie} Q'') \mid ((\mathcal{A}[\mathcal{E}],\Gamma), Q'') \in\\
&\quad apply(id)(Moves(Q')) \wedge \mathcal{A} \cap \mathcal{L} = \emptyset|\!\!\}\\
&\quad \uplus\{\!\!|((\mathcal{A}[\mathcal{E}],\Gamma), \mathcal{F}(P_1'') \underset{\mathcal{L}}{\bowtie} Q'')) \mid\\
&\quad ((\mathcal{A}_1[\mathcal{E}_1],\Gamma_1), \mathcal{F}(P_1'')) \in apply(\mathcal{F})(Moves(P_1')) \wedge\\
&\quad ((\mathcal{A}_2[\mathcal{E}_2],\Gamma_2), Q'') \in apply(id)(Moves(Q')) \wedge\\
&\quad \mathcal{A}_1 \cap \mathcal{A}_2 \cap \mathcal{L} \neq \emptyset \wedge \mathcal{A} = \mathcal{A}_1 \cup \mathcal{A}_2 \wedge \mathcal{E} = \mathcal{E}_1 \cup \mathcal{E}_2 \wedge\\
&\quad \Gamma = \Gamma_1 \cup \Gamma_2|\!\!\}\\
&= apply(\mathcal{G})(Moves(P_1' \underset{\mathcal{L}}{\bowtie} Q')
\end{aligned}
$$

4. With the same procedure used for 3, we define a model process isomorphism $\mathcal{G}$ as: $\forall P_1' \underset{\mathcal{L}}{\boxtimes} Q' \in ds(P_1 \underset{\mathcal{L}}{\boxtimes} Q)$, with $P_1' \in ds(P_1)$ and $Q' \in ds(Q)$,

$$\mathcal{G}(P_1' \underset{\mathcal{L}}{\boxtimes} Q') = \mathcal{F}(P_1') \underset{\mathcal{L}}{\boxtimes} Q'$$

$\mathcal{G}$ is a model process isomorphism because $\mathcal{F}$ is a model process isomorphism and it can be proved in analogy with point 3.

□

C. SEMANTICS OF PARAMETRIC PAH

The semantics for parametric PAH processes is presented in Figure 5. Given an environment Γ, the evaluation of an expression *exp* into a real number k is denoted by $\Gamma \vdash exp \rightarrow k$, the evaluation of a boolean expression *bexp* into $b \in \{true, false\}$ is denoted by $\Gamma \vdash bexp \rightarrow b$, while the evaluation of the list of expressions of all the actions in a set $\mathcal{A}$ is denoted by $\Gamma \vdash \mathcal{A} \rightarrow \mathcal{A}'$, where $\mathcal{A}'$ contains only actions with evaluated expressions.

D. DETAILED DEFINITION OF THE MODEL OF TISSUE GROWTH

Constants:

$k_1 = 1/(Ms)$	$k_2 = 5\ M/s$	$k_3 = 5\ M/s$
$k_4 = 1/s$	$k_5 = 1/s$	$h = 1\ M$
$k_{death} = 1\ event/s$	$k_{growth} = 1\ event/s$	$maxLevels = 10$
$rows = 10$	$cols = 10$	$thr = 5$

Functional rates and sets of participants:

$f_{r1(i,j)} = k_1 * A(i,j) * h * B(i,j) * h/h$
$p_{r1(i,j)} = \{A(i,j), B(i,j), C(i,j)\}$
$f_{r2(i,j)} = k_2/h$
$p_{r2(i,j)} = \{A(i,j)\}$
$f_{r3(i,j)} = k_3/h$
$p_{r3(i,j)} = \{B(i,j)\}$
$f_{r4(i,j)} = k4 * C(i,j) * h/h$
$p_{r4(i,j)} = \{C(i,j)\}$
$f_{r5(i,j,i2,j2)} = k5 * B(i,j) * h/h$
$p_{r5(i,j)} = \{B(i,j), B(i2,j2)\}$
$f_{death(i,j)} = k_{death}$
$p_{death(i,j)} = \{R(i,j)\}$
$f_{growth(i,j,i2,j2)} = k_{growth}$
$p_{growth(i,j,i2,j2)} = \{R(i,j), R(i2,j2)\}$

Agent definitions:

$NA(i,j) \triangleq bioon(i,j).A(i,j,0)$

```
$A(i,j,w) \triangleq biooff(i,j).NA(i,j)$
  $+($
    if $w < maxLevels$ then
      if $w == (thr - 1)$ then
        $r2(i,j)[growthon(i,j)].A(i,j,w+1)$
      else $r2(i,j).A(i,j,w+1)$
    else $nil)$
  $+($
    if $w > 0$ then
      if $w == thr$ then
        $r1(i,j)[growthoff(i,j)].A(i,j,w-1)$
      else $r1(i,j).A(i,j,w-1)$
    else $nil)$
```

$NB(i,j) \triangleq bioon(i,j).B(i,j,0)$

```
$B(i,j,w) \triangleq biooff(i,j).NB(i,j)$
  $+($
    if $w < maxLevels$ then
      $r3(i,j).B(i,j,w+1)$
      $+($if $i > 1$ then
        $r5(i-1,j,i,j).B(i,j,w+1)$ else $nil)$
      $+($if $i < rows$ then
        $r5(i+1,j,i,j).B(i,j,w+1)$ else $nil)$
      $+($if $j > 1$ then
        $r5(i,j-1,i,j).B(i,j,w+1)$ else $nil)$
      $+($if $j < cols$ then
        $r5(i,j+1,i,j).B(i,j,w+1)$ else $nil)$
    else $nil)$
  $+($
    if $w > 0$ then
      $r1(i,j).B(i,j,w-1)$
      $+($if $i > 1$ then
        $r5(i,j,i-1,j).B(i,j,w-1)$ else $nil)$
      $+($if $i < rows$ then
        $r5(i,j,i+1,j).B(i,j,w-1)$ else $nil)$
      $+($if $j > 1$ then
        $r5(i,j,i,j-1).B(i,j,w-1)$ else $nil)$
      $+($if $j < cols$ then
        $r5(i,j,i,j+1).B(i,j,w-1)$ else $nil)$
    else $nil)$
```

$NC(i,j) \triangleq bioon(i,j).C(i,j,0)$

Prefix

$$\frac{}{\mathcal{A}[\mathcal{E}].A(exp_1,\dots,exp_n) \xrightarrow{\mathcal{A}[\mathcal{E}],true} A(exp_1,\dots,exp_n)}$$

Choice Left

$$\frac{D_1 \xrightarrow{\mathcal{A}[\mathcal{E}],b} A(exp_1,\dots,exp_n)}{D_1 + D_2 \xrightarrow{\mathcal{A}[\mathcal{E}],b} A(exp_1,\dots,exp_n)}$$

Choice Right

$$\frac{D_2 \xrightarrow{\mathcal{A}[\mathcal{E}],b} A(exp_1,\dots,exp_n)}{D_1 + D_2 \xrightarrow{\mathcal{A}[\mathcal{E}],b} A(exp_1,\dots,exp_n)}$$

If Then Else True

$$\frac{D_1 \xrightarrow{\mathcal{A}[\mathcal{E}],b} A(exp_1,\dots,exp_n)}{\text{if } bexp \text{ then } D_1 \text{ else } D_2 \xrightarrow{\mathcal{A}[\mathcal{E}],b\wedge bexp} A(exp_1,\dots,exp_n)}$$

If Then Else False

$$\frac{D_2 \xrightarrow{\mathcal{A}[\mathcal{E}],b} A(exp_1,\dots,exp_n)}{\text{if } bexp \text{ then } D_1 \text{ else } D_2 \xrightarrow{\mathcal{A}[\mathcal{E}],b\wedge \neg bexp} A(exp_1,\dots,exp_n)}$$

Agent

$$\frac{D \xrightarrow{\mathcal{A}[\mathcal{E}],b} A'(exp_1,\dots,exp_n)}{A(k_1,\dots,k_n) \xrightarrow{(\mathcal{A}'[\mathcal{E}'],\Gamma')} A'(k'_1,\dots,k'_n)} \quad \begin{array}{l} \text{if } A(i_1,\dots,i_n) \triangleq D \wedge \Gamma = \{(i_1,k_1),\dots,(i_n,k_n)\} \wedge \Gamma \vdash b \to true \\ \Gamma \vdash exp_1 \to k'_1 \wedge \dots \wedge \Gamma \vdash exp_n \to k'_n \wedge \Gamma \vdash \mathcal{A} \to \mathcal{A}' \wedge \Gamma \vdash \mathcal{E} \to \mathcal{E}' \\ \wedge\, \Gamma' = \{(Var(A(k_1,\dots,k_n)), Val(A(k_1,\dots,k_n)))\} \end{array}$$

Figure 5: Semantics of parametric process algebra with hooks. Other inference rules are unchanged.

```
$C(i,j,w) \triangleq biooff(i,j).NC(i,j)$
    $+($
        if $w < maxLevels$ then
            if $w == (thr-1)$ then
                $r1(i,j)[deathon(i,j)].C(i,j,w+1)$
            else $r1(i,j).C(i,j,w+1)$
        else $nil)$
    $+($
        if $w > 0$ then
            if $w == thr$ then
                $r4(i,j)[deathoff(i,j)].C(i,j,w-1)$
            else $r4(i,j).C(i,j,w-1)$
        else $nil)$
```

```
$E(i,j) \triangleq$
    (if $i > 1$ then
        $growth(i-1,j,i,j)[bioon(i,j)].T(i,j)$ else $nil)$
    $+($if $i < rows$ then
        $growth(i+1,j,i,j)[bioon(i,j)].T(i,j)$ else $nil)$
    $+($if $j > 1$ then
        $growth(i,j-1,i,j)[bioon(i,j)].T(i,j)$ else $nil)$
    $+($if $j < cols$ then
        $growth(i,j+1,i,j)[bioon(i,j)].T(i,j)$ else $nil)$
```

$T(i,j) \triangleq growthon(i,j).Tm(i,j) + deathon(i,j).Ta(i,j)$

```
$Tm(i,j) \triangleq$
    (if $i > 1$ then $growth(i,j,i-1,j).Tm(i,j)$ else $nil)$
    $+($if $i < rows$ then $growth(i,j,i+1,j).Tm(i,j)$ else $nil)$
    $+($if $j > 1$ then $growth(i,j,i,j-1).Tm(i,j)$ else $nil)$
    $+($if $j < cols$ then $growth(i,j,i,j+1).Tm(i,j)$ else $nil)$
    $+growthoff(i,j).T(i,j) + deathon(i,j).Tam(i,j)$
    $+\{growthoff(i,j), deathon(i,j)\}.Ta(i,j)$
```

```
$Ta(i,j) \triangleq death(i,j)[biooff(i,j)].E(i,j)$
    $+apooff(i,j).T(i,j) + mitoon(i,j).Tam(i,j)$
    $+\{growthon(i,j), deathoff(i,j)\}.Ta(i,j)$
```

```
$Tam(i,j), [R(i,j), 2] \triangleq death(i,j)[biooff(i,j)].E(i,j)$
    $+($if $i > 1$ then $growth(i,j,i-1,j).Tam(i,j)$ else $nil)$
    $+($if $i < rows$ then $growth(i,j,i+1,j).Tam(i,j)$ else $nil)$
    $+($if $j > 1$ then $growth(i,j,i,j-1).Tam(i,j)$ else $nil)$
    $+($if $j < cols$ then $growth(i,j,i,j+1).Tam(i,j)$ else $nil)$
    $+growthoff(i,j).Ta(i,j) + deathoff(i,j).Tm(i,j)$
```

Associated variables and values:

$Var(NA(i,j)) = Var(A(i,j,w)) = A(i,j);$

$Val(NA(i,j)) = 0;\ Val(A(i,j,w)) = w;$

$Var(NB(i,j)) = Var(B(i,j,w)) = B(i,j);$

$Val(NB(i,j)) = 0;\ Val(B(i,j,w)) = w;$

$Var(NC(i,j)) = Var(C(i,j,w)) = C(i,j);$

$Val(NC(i,j)) = 0;\ Val(C(i,j,w)) = w;$

$Var(E(i,j)) = Var(T(i,j)) = Var(Ta(i,j)) = Var(Tm(i,j)) = Var(Tam(i,j)) = R(i,j);$

$Val(E(i,j)) = 0;\ Val(T(i,j)) = 1;\ Val(Ta(i,j)) = 1;\ Val(Tm(i,j)) = 1;\ Val(Tam(i,j)) = 1;$

Model process and initial state:

$((NA(1,1) \underset{\mathcal{L}_{1,1}}{\bowtie} NB(1,1) \underset{\mathcal{L}_{1,1}}{\bowtie} NC(1,1)) \underset{\mathcal{K}_{1,1}}{\bowtie}$
$\dots \underset{\mathcal{K}_{1,9}}{\bowtie} (NA(1,10) \underset{\mathcal{L}_{1,10}}{\bowtie} NB(1,10) \underset{\mathcal{L}_{1,10}}{\bowtie} NC(1,10))$
$) \underset{\mathcal{K}_{1,10}}{\bowtie} (\cdots$

$\cdots \underset{\mathcal{K}_{6,5}}{\bowtie} (A(6,6,5) \underset{\mathcal{L}_{6,6}}{\bowtie} B(6,6,0) \underset{\mathcal{L}_{6,6}}{\bowtie} C(6,6,0)) \underset{\mathcal{K}_{6,6}}{\bowtie} \cdots$

$\cdots) \underset{\mathcal{K}_{9,10}}{\bowtie} ($
$(NA(10,1) \underset{\mathcal{L}_{10,1}}{\bowtie} NB(10,1) \underset{\mathcal{L}_{10,1}}{\bowtie} NC(10,1)) \underset{\mathcal{K}_{10,1}}{\bowtie}$
$\ldots \underset{\mathcal{K}_{10,9}}{\bowtie} (NA(10,10) \underset{\mathcal{L}_{10,10}}{\bowtie} NB(10,10) \underset{\mathcal{L}_{10,10}}{\bowtie} NC(10,10)))$

$\underset{\mathcal{H}}{\rotatebox{90}{$\bowtie$}}$

$(E(1,1) \underset{\mathcal{N}_{1,1}}{\bowtie} \cdots \underset{\mathcal{N}_{1,9}}{\bowtie} E(1,10)) \underset{\mathcal{N}_{1,10}}{\bowtie} \cdots$

$\cdots (E(6,1) \underset{\mathcal{N}_{6,1}}{\bowtie} \cdots Tm(6,6) \cdots \underset{\mathcal{N}_{6,9}}{\bowtie} E(6,10)) \cdots$

$\cdots (E(10,1) \underset{\mathcal{N}_{10,1}}{\bowtie} \cdots \underset{\mathcal{N}_{10,9}}{\bowtie} E(10,10))$

$\mathcal{L}_{1,1} = \{r1(1,1), bioon(1,1), biooff(1,1)\}$
$\mathcal{K}_{1,1} = \{r5(1,1,1,2), r5(1,2,1,1)\}$
$\mathcal{K}_{1,9} = \{r5(1,9,1,10), r5(1,10,1,9)\}$
$\mathcal{L}_{1,10} = \{r1(1,10), bioon(1,10), biooff(1,10)\}$
$\mathcal{K}_{1,10} = \{r5(1,1,2,1), r5(2,1,1,1), r5(1,2,2,2), r5(2,2,1,2),$
$r5(1,3,2,3), r5(2,3,1,3), r5(1,4,2,4), r5(2,4,1,4),$
$r5(1,5,2,5), r5(2,5,1,5), r5(1,6,2,6), r5(2,6,1,6),$
$r5(1,7,2,7), r5(2,7,1,7), r5(1,8,2,8), r5(2,8,1,8),$
$r5(1,9,2,9), r5(2,9,1,9), r5(1,10,2,10), r5(2,10,1,10)\}$
$\mathcal{H} = \{bioon(1,1), biooff(1,1), deathon(1,1), deathoff(1,1),$
$growthon(1,1), growthoff(1,1), bioon(1,2), biooff(1,2),$
$deathon(1,2), \cdots, deathoff(10,9), growthon(10,9),$
$growthoff(10,9), bioon(10,10), biooff(10,10), deathon(10,10),$
$deathoff(10,10), growthon(10,10), growthoff(10,10)\}$
$\mathcal{N}_{1,9} = \{growth(1,9,1,10), growth(1,10,1,9)\}$
$\mathcal{N}_{1,10} = \{growth(1,1,2,1), growth(2,1,1,1), growth(1,2,2,2),$
$growth(2,2,1,2), growth(1,3,2,3), growth(2,3,1,3),$
$growth(1,4,2,4), growth(2,4,1,4), growth(1,5,2,5),$
$growth(2,5,1,5), growth(1,6,2,6), growth(2,6,1,6),$
$growth(1,7,2,7), growth(2,7,1,7), growth(1,8,2,8),$
$growth(2,8,1,8), growth(1,9,2,9), growth(2,9,1,9),$
$growth(1,10,2,10), growth(2,10,1,10)\}$