A Process Algebra Framework for Multi-Scale Modelling of Biological Systems

A. Degasperi[a,*], M. Calder[b]

[a] *Systems Biology Ireland, Conway Institute, University College Dublin, Belfield, Dublin, Ireland*
[b] *School of Computing Science, University of Glasgow, Sir Alwyn Williams Building, G12 8QQ, United Kingdom*

Abstract

We introduce a novel process algebra for modelling biological systems at multiple scales, called process algebra with hooks (PAH). Processes represent biological entities, such as molecules, cells and tissues, while two algebraic operators, both symmetric, define composition of processes within and between scales. Composed actions allow for biological events to interact within and between scales at the same time. The algebra has a stochastic semantics based on functional rates of reactions. Two bisimulations are defined on PAH processes. The first bisimulation is used to aid model development by checking that two biological scales can interact correctly. The second bisimulation is a congruence that relates models, or part of models, that can perform the same timed events at a specified scale. Finally, we provide a PAH model of pattern formation in a tissue and illustrate reasoning about its behaviour using the PAH framework.

Keywords: Process algebra, multi-scale, biological systems, functional rates

[☆] This paper is the extension of published work [1] and has been adapted from the first author's Ph.D. thesis [2].

[*] Corresponding author

Email addresses: andrea.degasperi@ucd.ie (A. Degasperi), Muffy.Calder@glasgow.ac.uk (M. Calder)

1. Introduction

Systems Biology [3] is an emerging discipline that aims to improve our understanding of the dynamics of biological processes with the aid of mathematical models. As our knowledge about the mechanics and the complexity of biological phenomena increases, predictive models become necessary to validate understanding and generate new hypotheses.

The level of detail at which biological processes are most commonly modelled is biochemical reactions, using mathematical approaches such as ordinary differential equations (ODE) and stochastic processes [4]. Other approaches are employed to represent diffusion of molecules, using partial differential equations (PDE) [5], or higher order structures such as cells or tissue, using cellular automata (CA) [6] or other agent based techniques. Descriptive languages, e.g. SBML [7], and graphical notations, e.g. Kitano Map [8], have been developed to help writing, maintaining and sharing models. In addition, formalisms from the field of computer science have been proposed not only to provide an unambiguous definition of biological phenomena, but also to improve the overall modelling approach. Some of the most successful formalisms are process algebras and other calculi [9, 10, 11, 12], rewriting rules [13, 14] and programming languages [15, 16].

Process algebras are a family of calculi developed to represent and analyse formally the behaviour of concurrent systems, such as programs on a computer or computers in a network [17, 18]. They have been shown to be one of the most promising approaches to the formalisation of biological systems, because of the deep analogies that exist between concurrent agent interactions and biochemical reactions [19].

The development and application of process algebra for biology has mostly been aimed at modelling biochemical reactions and compartments [20, 10, 9, 11, 21]. More recently there has been a growing interest in combining different levels of detail of biological phenomena into single multi-scale models that represent both biochemical details and higher order structures. This is a necessary step to achieve a complete understanding of the emerging behaviour in a complex biological phenomenon. Model construction follows mainly two approaches: *bottom-up* and *top-down*. The former begins from identifying elementary parts, such as molecules, and aims at explaining more complex phenomena as the emergent behaviour of its components. The latter begins instead from reproducing observed phenomena and then adds internal details, attempting to recreate governing mechanisms. Different mathemati-

cal approaches are often considered for different scales and integrated into a multi-scale model tailored to a specific biological problem [22, 23, 24]. As a consequence, composition and comparison of two multi-scale models is often very difficult.

It has been proposed [25] that new, more flexible modelling techniques should allow for a *middle-out* approach. This means that one begins studying, and so modelling, a biological phenomenon from any level of detail or spatial scale and, in a second stage, extending its study and so its model either up scale, integrating with other components, or down scale, adding more internal details. To our knowledge, we were among the first to addresses the problem of integrating multiple scales under the same mathematical framework with the flexibility of treating different scales as the same formal objects [1, 26].

In this paper we propose that a process algebra framework is a perfect candidate as a middle-out approach for multi-scale modelling. In particular, its natural support of compositionality and its abstraction mechanisms can provide the required flexibility that **writing**, **composing** and **comparing** multi-scale models require.

In these paper we show the following advantages of using a process algebra framework in place of traditional modelling approaches such as in [27, 28]:

- (writing) different biological scales are represented by the same mathematical objects under a unified framework. A modeller can begin writing a multi-scale model from any scale and continue up-scale or down-scale without changing the mathematical approach;

- (composing) composition of models is facilitated by operators for composition within scale (e.g. two tissues next to each other) or between scales (e.g. cells that constitute a tissue);

- (comparing) most importantly, the unified framework allows for *automated reasoning* between entities in a way that is not accessible by traditional modelling approaches. In particular, process algebra has a well established theory of relations based on behaviour. Here we show how we can *aid model development* by defining a relation between scales such that the relation holds true only when two scales interact correctly. Moreover, we provide a relation that allows the behaviour of systems to be *compared at a specified scale* or part of a system to be abstracted with other parts that are behaviourally equivalent.

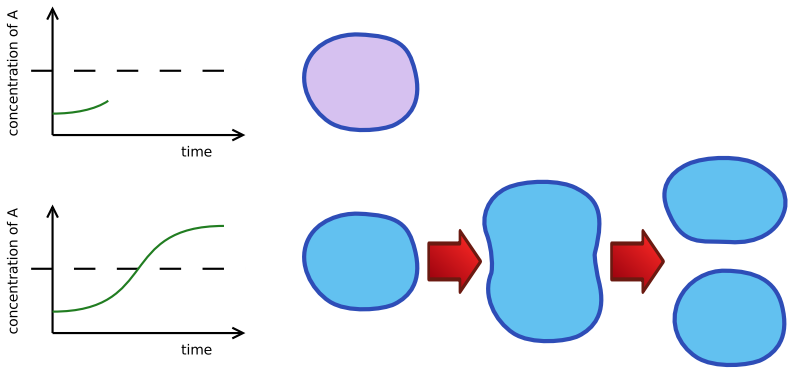


Figure 1: Interactions between scales. Only if the concentration of a certain molecule (molecular scale) is high, then a cell can duplicate (cellular scale).

Biological scales. We focus on the definition of scales and of interactions within and between scales. We illustrate the meaning of "interactions between scales" with the following examples. The first example is illustrated in Figure 1. A dependency is defined between the molecular scale (on the left of the figure) and the cellular scale (on the right of the figure): cellular duplication is possible if and only if the concentration of molecule **A** is above a certain threshold. In other words, high concentration of molecule **A** activates the ability of the cell to duplicate. The second example is illustrated in Figure 2. If a cell dies (top of figure) this implies the concentration of the molecules inside it is dispersed. If a cell **C** duplicates (bottom of figure), independent concentrations of molecules originally in **C** will be present in both the resulting cells **C**' and **C**''.

The main contributions of this paper are:

- the definition of *process algebra with hooks*, a process algebra designed for multi-scale modelling of biological systems. Its main features are: explicit modelling of scales and interactions within and between scales; use of *composed* actions in a multi-way synchronisation setting; a *vertical* cooperation operator in addition to the standard cooperation operator for composition of processes. The vertical cooperation is symmetric, i.e. events at higher scales can influence the behaviour of lower scales and vice versa;

- the definition of a functional rate semantics for process algebras based

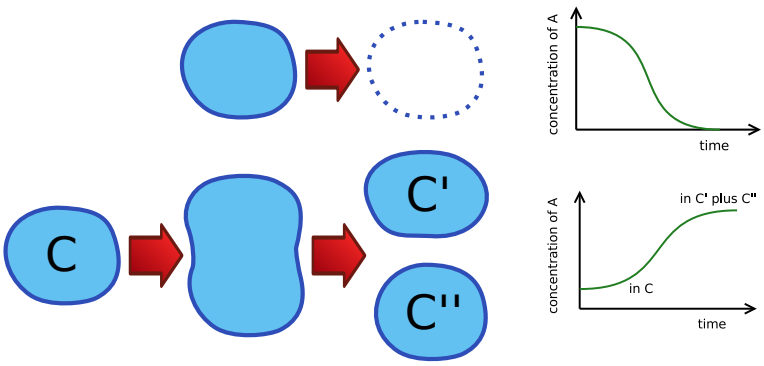


Figure 2: Dependencies between scales. Cellular events such as death and duplication (cellular scale), imply changes in concentration of molecules (molecular scale) inside the cells.

on biological principles, where actions can be rated only if *closed*, i.e. only if all the expected participants to that action synchronise;

- the definition of a relation between scales that aims to aid model development by detecting when two scales do not interact as intended. This relation is the compatibility $\mathcal{L}$-bisimulation (Section 5);
- the definition of a congruence relation to relate and substitute process algebra with hooks processes. This relates processes by their temporal behaviour at a specified scale (Markovian $(\mathcal{T}, \Gamma)$-bisimulation) (Section 6). The proof of congruence for Markovian $(\mathcal{T}, \Gamma)$-bisimulation is possible because of the concept of closed actions introduced with our definition of functional rates;
- a detailed illustration of use of process algebra with hooks to model, simulate and relate a multi-scale model of a well known problem of pattern formation in a tissue (Section 8).

2. Key Concepts in Process Algebra

In this section we give an overview of key concepts in process algebra. The fundamental elements in a process algebra model are autonomous agents called *processes*. Each process is characterised by its behaviour, expressed in terms of *actions* it can perform. For example, if a process P can perform a sequence of three a actions, we denote it by:

$$P \triangleq a.a.a.nil$$

where "." is called the prefix operator and *nil* is defined as the terminated process, i.e. the process that cannot perform any action. A *labelled transition* provides semantics. For example, process P can perform action a and become process P', where $P' \triangleq a.a.nil$. This is denoted by:

$$P \xrightarrow{a} P'$$

Process P' is called *one step derivative of* P, while if a process P'' can be reach after one or more transitions then P'' is called simply a derivative of P. The collection of process P, the set of derivatives of P and all possible labelled transitions between the derivatives is called the *derivation graph of* P.

A process may choose non-deterministically between multiple available actions. This is denoted using the *choice* operator "+", for example if

$$Q \triangleq a.nil + b.nil + c.d.nil$$

then there are three labelled transitions:

$$Q \xrightarrow{a} nil \qquad Q \xrightarrow{b} nil \qquad Q \xrightarrow{c} d.nil$$

Most importantly, processes can synchronise on actions. Synchronisation can be *binary* between two actions with complementary names, in the style of calculus of communicating systems (CCS) [17], or *multi-way* between any number of actions sharing the same name, in the style of communicating sequential processes (CSP) [18]. We follow the latter approach, as this allows us to model biochemical reactions that involve any number of substrates and products with a single transition (as presented in [9]). Multi-way synchronisation is possible using the *cooperation* operator $\underset{\mathcal{L}}{\bowtie}$, using the notation of performance evaluation process algebra (PEPA [29]). The set of actions $\mathcal{L}$, or cooperation set, indicates which actions are used for synchronisation. For example, given the processes:

$$R \triangleq a.nil + b.nil \qquad S \triangleq a.nil + b.nil + c.nil$$

and the overall model defined as $R \underset{a,c}{\bowtie} S$, the following transitions are possible:

$$R \underset{a,c}{\bowtie} S \xrightarrow{a} nil \underset{a,c}{\bowtie} nil \qquad R \underset{a,c}{\bowtie} S \xrightarrow{b} nil \underset{a,c}{\bowtie} S \qquad R \underset{a,c}{\bowtie} S \xrightarrow{b} R \underset{a,c}{\bowtie} nil$$

Because action a is in the cooperation set, R and S can synchronise on a, but cannot perform a individually. On the contrary, b is not in the cooperation set, so R and S cannot synchronise on b, though they can perform b individually. Finally, c cannot be performed by S, because it is present in the cooperation set, which would require that also R had the possibility of performing c.

Another key feature of process algebra is the possibility for actions to become *hidden*. This is usually expressed by replacing the name of an action with the *unknown action type* τ. This substitution may happen in an *implicit* way, as in CCS, or in an *explicit* way, as in CSP. In CCS, as a result of a binary synchronisation, the name of the two complementary actions that synchronise is replaced by τ. In contrast, in CSP it is the responsibility of the modeller to place *hiding* ($\setminus$) operators appropriately in the system. For example, $R \underset{a,c}{\bowtie} S \setminus \{b\}$ denotes that if $R \underset{a,c}{\bowtie} S$ can perform action b, it will be replaced with τ. This results in the following labelled transitions:

$$(R \underset{a,c}{\bowtie} S) \setminus \{b\} \xrightarrow{a} (nil \underset{a,c}{\bowtie} nil) \setminus \{b\} \qquad (R \underset{a,c}{\bowtie} S) \setminus \{b\} \xrightarrow{\tau} (nil \underset{a,c}{\bowtie} S) \setminus \{b\}$$
$$(R \underset{a,c}{\bowtie} S) \setminus \{b\} \xrightarrow{\tau} (R \underset{a,c}{\bowtie} nil) \setminus \{b\}$$

Finally, *relations* are defined between processes, to express similar behaviour. In particular, fundamental to every process algebra are notions of *equivalences*, such as *bisimulation*, and whether such equivalences are also *congruences* or not. In general, a congruence relation allows the substitution of a process with a behaviourally equivalent, and possibly less complex, other process.

3. Process Algebra with Hooks by Examples

In this section we introduce PAH by examples, such as modelling simple cell behaviour, biochemistry and interactions between scales. Rating of actions and relations between processes are also discussed.

3.1. Simple Model of Cell Behaviour

Example 1. In PAH we represent biological entities as processes and biological events as actions. For example, assume we want to represent the

behaviour of a cell, which we denote as **Cell**, that can either move or absorb nutrients, yet not both at the same time. Biological entity **Cell** can be represented by the following two processes $Cell_0$ and $Cell_1$:

$$Cell_0 \triangleq x.Cell_1 + move.Cell_0 \quad Cell_1 \triangleq y.Cell_0 + absorb.Cell_1$$

Processes $Cell_0$ and $Cell_1$ represent the two possible states of **Cell**, one in which the cell can only move, represented by the action *move*, while the other where the cell can only absorb nutrients, represented by the action *absorb*. In general, the number of states a biological entity can assume is not restricted to two and is determined by how many processes are associated with such entity.

Transitions between the two states are denoted by the actions x and y, which may represent biological events, either internal or external to the cell, that influence the behaviour of the cell. In PAH the rate r_a of an action a is computed evaluating a *functional rate* f_a that may depend on the state of the biological entity involved in the biological event represented by a. The rate r_a is the parameter of the exponential distribution of the time required to perform action a. Each process is associated with a *variable*, in this case **Cell**, and with a *value*, here we chose arbitrarily zero or one depending on the state. We use partial functions *Var* and *Val* to define this association:

$$Var(Cell_0) = \mathbf{Cell} \quad Var(Cell_1) = \mathbf{Cell} \quad Val(Cell_0) = 0 \quad Val(Cell_1) = 1$$

In this basic example, the functional rates for all the actions are constants:

$$f_{move} = k_{move} \quad f_{absorb} = k_{absorb} \quad f_x = k_x \quad f_y = k_y$$

In addition to a functional rate f_a, an action is associated with a *set of participants* p_a, which we will discuss in the next section. In general, a set of participants indicates which biological entities (and so which processes) are expected to participate to a biological event (and so action). Assuming a model defined by the single $Cell_0$ process, a possible transition is given by:

$$Cell_0 \xrightarrow{(move, k_{move})} Cell_0$$

In the next section we show how variables, values, functional rates and sets of participants are used when modelling biochemical reactions in PAH.

3.2. Modelling the Biochemical Scale

The concentration of each biochemical species can be modelled with discrete levels of concentration, using the *process as level of concentration* abstraction [30]. In this approach, for each species, each level is represented by a process, while biochemical reactions are represented by actions that produce discrete changes of concentration levels. Each level corresponds to a discrete concentration, denoted by h, which is the ratio between a maximum concentration M and the number of levels N, i.e. $h = M/N$. In this case the variables associated to the processes correspond to species names, while the values correspond to concentration levels. An action may be associated with a functional rate and with a *set of participants* of the biological event. A set of participants indicates which biological entities are expected to participate to a biological event, and so which processes are expected to synchronise on a given action. In the following example of biochemical reactions, reaction velocities are used to formulate functional rates in the style of [30].

Example 2.

$$R_a : \mathbf{A} + \mathbf{B} \rightarrow_{v_a} \mathbf{C}, \quad v_a = k_a[\mathbf{A}][\mathbf{B}]$$

In this example we use PAH to model biochemical reaction R_a, which has velocity v_a, measured in concentration per second. The above notation means that in R_a, molecules **A** and **B** bind together to yield molecule **C**. Moreover, the notation $[\cdot]$ means concentration.

$$\begin{array}{lll} A_L \triangleq nil & B_L \triangleq nil & C_L \triangleq a.C_H \\ A_H \triangleq a.A_L & B_H \triangleq a.B_L & C_H \triangleq nil \end{array}$$

$$\begin{array}{lll} Var(A_L) = \mathbf{A} & Var(B_L) = \mathbf{B} & Var(C_L) = \mathbf{C} \\ Var(A_H) = \mathbf{A} & Var(B_H) = \mathbf{B} & Var(C_H) = \mathbf{C} \end{array}$$

$$\begin{array}{llll} Val(A_L) = 0 & Val(B_L) = 0 & Val(C_L) = 0 \\ Val(A_H) = 1 & Val(B_H) = 1 & Val(C_H) = 1 \end{array}$$

$$p_a = \{\mathbf{A},\mathbf{B},\mathbf{C}\}, \quad f_a = (k_a \cdot \mathbf{A} \cdot h \cdot \mathbf{B} \cdot h)/h$$

In this case, biochemical concentration is the biological entity we are modelling. Process A_L represents the concentration of species **A** at low level, that is we use L for "low" and H for "high". We use two states (high and low) to represent the concentration just for illustration purposes. In general, any finite number of states, and so of concentration levels, can be used. Notice that

the subscript is not strictly part of the syntax. We could have written AL or $Alow$ instead. What is important is that at any moment there is only one process for each species, indicating its concentration level. We use functions $Var(\cdot)$ and $Val(\cdot)$ to associate processes with variables and values. In the above example, p_a is the set of participants of reaction R_a, associated with action a. Function f_a is the functional rate of a, where $\mathbf{A}$ and $\mathbf{B}$ are species names, that is the *variables* that at the evaluation of the functional rate will be substituted by the appropriate *value*, here the concentration levels.

The model of Example 2 is defined by the following process[1]:

$$A_H \underset{\{\!|a|\!\}}{\bowtie} (B_H \underset{\{\!|a|\!\}}{\bowtie} C_L)$$

Processes A_H, B_H and C_L can perform a and synchronise via the cooperation operator $\underset{\{\!|a|\!\}}{\bowtie}$. This results in the transition:

$$A_H \underset{\{\!|a|\!\}}{\bowtie} (B_H \underset{\{\!|a|\!\}}{\bowtie} C_L) \xrightarrow{(a,\Delta)} A_L \underset{\{\!|a|\!\}}{\bowtie} (B_L \underset{\{\!|a|\!\}}{\bowtie} C_H)$$

This indicates that most of the concentration of $\mathbf{A}$ and $\mathbf{B}$ has been converted into concentration of $\mathbf{C}$. The partial function Δ is constructed using variables and values associated with the processes A_H and B_H and it is used to evaluate f_a. In this case $\Delta = \{(Var(A_H), Val(A_H)), (Var(B_H), Val(B_H))\} = \{(\mathbf{A}, 1), (\mathbf{B}, 1)\}$.

Because the synchronisation involves all the participants in p_a, the resulting transition is defined as *closed* and the reaction rate r_a can be computed using the functional rate f_a.

$$A_H \underset{\{\!|a|\!\}}{\bowtie} (B_H \underset{\{\!|a|\!\}}{\bowtie} C_L) \xrightarrow{(a,r_a)} A_L \underset{\{\!|a|\!\}}{\bowtie} (B_L \underset{\{\!|a|\!\}}{\bowtie} C_H)$$

If the model consisted only of $A_H \underset{\{\!|a|\!\}}{\bowtie} B_H$, then a transition would produce $A_L \underset{\{\!|a|\!\}}{\bowtie} B_L$, but the biochemical reaction associated with action a would be missing a participant, that is $\mathbf{C}$. We consider this an incomplete or *open* transition because the effects on $\mathbf{C}$ are not included. We have chosen to forbid rating of open transitions, to permit the definition of the congruence

[1]brackets {| and |} delimit multi-sets. Although multi-sets are not necessary in this example, we use them for consistency with our language definition. See Example 6 in Section 3.4 to see an example that motivates the use multi-sets.

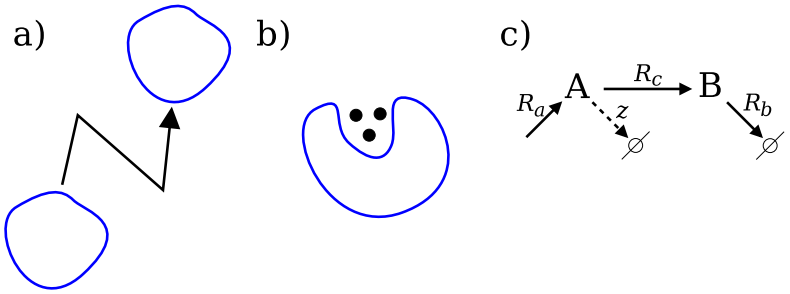


Figure 3: Illustration of Example 3. a) Only if the concentration of **B** in **Cell** is low, then **Cell** can move. b) Only if the concentration of **B** in **Cell** is high, then **Cell** can absorb nutrients. c) Biochemical scale of the system: straight lines are biochemical reactions, while the segmented line labelled z represents the reduction of concentration of **A** caused by the absorption of nutrients by **Cell**.

relation Markovian $(\mathcal{T}, \Gamma)$-bisimulation in PAH (Definition 36). Moreover, we can consider $A_H \underset{\{|a|\}}{\bowtie} B_H$ and C_L as two parts that need each other to express behaviour that is biologically meaningful, that is the execution of reaction R_a.

3.3. Linking Scales with Hook Actions

We show now how two scales can be defined and merged into a single multi-scale model. Communication between scales is achieved via *hook actions*. These are so-called because of their role in the algebra: they are attached to other actions, written $a[x]$, where x is a hook action. Action $a[x]$, a *composed action*, is almost equivalent to a, the difference is that the additional x can be observed by another process, using the vertical cooperation operator $\underset{\mathcal{L}}{\boxtimes}$. In analogy with the horizontal cooperation operator $\underset{\mathcal{L}}{\bowtie}$ (Section 2), composed actions that include actions in the cooperation set $\mathcal{L}$ are not allowed to be executed asynchronously. The new operator explicitly separates scales and it is symmetric: synchronisations are possible bottom-up and top-down.

Example 3. In a cell **Cell**, there are two molecules **A** and **B**. Molecule **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:

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

The cell **Cell** can be in one of two states, Cell_0 and Cell_1, which depend on the concentration of **B**. When **Cell** is in Cell_0, it moves, that is it performs cell action *move*. When it is in Cell_1 it absorbs nutrients, that is it performs cell action *absorb*. As a bottom-up interaction, when the concentration of **B** in the cell is high, then the cell is in state Cell_1, Cell_0 otherwise. As a top-down interaction, when **Cell** absorbs nutrients the concentration of **A** is lowered, reducing the production of **B** and in turn making **Cell** less likely to stop again to absorb new nutrients (Figure 3). We can model this scenario using PAH in the following way:

$$\begin{array}{lll} A_L \triangleq a.A_M + z.A_L & A_M \triangleq a.A_H + c.A_L + z.A_L & A_H \triangleq c.A_M + z.A_M \\ B_L \triangleq c.B_M & B_M \triangleq c[x].B_H + b.B_L & B_H \triangleq b[y].B_M \\ \multicolumn{3}{l}{Cell_0 \triangleq x.Cell_1 + move.Cell_0 \qquad Cell_1 \triangleq y.Cell_0 + absorb[z].Cell_1} \end{array}$$

The concentration of **A** and **B** is represented by three processes for each molecule, indicating a concentration level, low (L), medium (M) and high (H). The state of the cell is represented by two processes $Cell_0$ and $Cell_1$. We use hook actions x and y to indicate that the concentration of **B** has passed a threshold and that the state of **Cell** has to change at the same time. We use hook action z to indicate that the concentration of **A** is lowered by the absorption of nutrients. Consider the model defined by the following process:

$$(A_H \underset{\{\!|c|\!\}}{\bowtie} B_M) \underset{\{\!|x,y,z|\!\}}{\bowtie} Cell_0$$

The vertical synchronisation operator $\underset{\{\!|x,y,z|\!\}}{\bowtie}$ clearly separates the molecular scale from the cellular scale, while indicating that actions x, y and z are actions that operate between scales. Additionally, this operator prevents x, y and z from being executed unless a synchronisation with a composed action that presents either x, y or z as hook actions is possible. Assuming a functional rate f_c is defined and can be evaluated to rate r_c, an example of a valid transition is:

$$(A_H \underset{\{\!|c|\!\}}{\bowtie} B_M) \underset{\{\!|x,y,z|\!\}}{\bowtie} Cell_0 \xrightarrow{(\{\!|c,x|\!\},\emptyset,r_c)} (A_M \underset{\{\!|c|\!\}}{\bowtie} B_H) \underset{\{\!|x,y,z|\!\}}{\bowtie} Cell_1$$

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 **Cell** changed its state from Cell_0 to Cell_1. The empty set indicates that no hook has been left unused.

If the model consisted only of $A_H \underset{\{\!|c|\!\}}{\bowtie} B_M$, and assuming $p_c = \{\mathbf{A}, \mathbf{B}\}$, an analogous transition would still be possible:

$$A_H \underset{\{\!|c|\!\}}{\bowtie} B_M \xrightarrow{(\{\!|c|\!\},\{\!|x|\!\},r_c)} A_M \underset{\{\!|c|\!\}}{\bowtie} B_H$$

The facts that c and x are in different multi-sets and that x is in the second multi-set indicates that hook x has been left unused.

Assuming functional rate f_{absorb} is defined and can be evaluated to r_{absorb}, another example of a valid transition is:

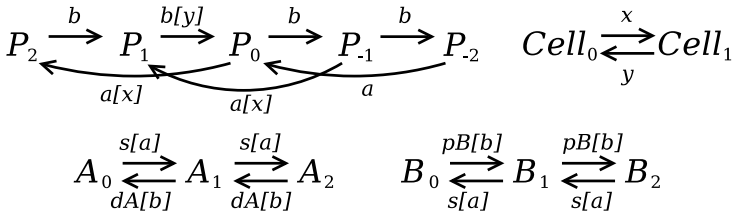
$$(A_M \underset{\{\!|c|\!\}}{\bowtie} B_H) \underset{\{\!|x,y,z|\!\}}{\bowtie} Cell_1 \xrightarrow{(\{\!|absorb,z|\!\},\emptyset,r_{absorb})} (A_L \underset{\{\!|c|\!\}}{\bowtie} B_H) \underset{\{\!|x,y,z|\!\}}{\bowtie} Cell_1$$

In the above transition, an action at the cellular scale, the absorption of nutrients, influences the biochemical scale, the concentration of **A**, in a top-down way.

3.4. Examples of interactions between scales

In this section we focus on examples of interactions between scales and we ignore for the moment the computation of rates. We also use a graphical representation of processes, where if a process is defined as $P \triangleq a.P' + b.P''$ then there is a directed edge labelled a from node P to P' and a directed edge labelled b from node P to P''.

Example 4. In this example, a change of behaviour of a cell is triggered when the concentration of molecule **A** exceeds the concentration of molecule **B**. Processes P_i, $i \in \{-2, \ldots, 2\}$, can be used to count the difference between the concentration levels of **A** and **B**. The graphical representation of the processes is given by:



The initial state is:

$$((A_1 \underset{\{\!|s|\!\}}{\bowtie} B_2) \underset{\{\!|a,b|\!\}}{\bowtie} P_{-1}) \underset{\{\!|x,y|\!\}}{\bowtie} Cell_0$$

Species **A** can degrade (dA), **B** can be produced (pB), while both **A** and **B** can synchronise (on s) so that a level of **B** is converted into a level of **A**. Processes P_i, $i \in \{-2, \ldots, 2\}$, represent the difference between the current level of **A** and of **B**, while a and b actions represent events that make this difference increase by 2 and decrease by 1 respectively. An example transition is:

$$((A_1 \underset{\{\!|s|\!\}}{\bowtie} B_2) \underset{\{\!|a,b|\!\}}{\bowtie} P_{-1}) \underset{\{\!|x,y|\!\}}{\bowtie} Cell_0 \xrightarrow{\{\!|s,a,x|\!\}[\emptyset]} ((A_2 \underset{\{\!|s|\!\}}{\bowtie} B_1) \underset{\{\!|a,b|\!\}}{\bowtie} P_1) \underset{\{\!|x,y|\!\}}{\bowtie} Cell_1$$

Example 5. If a scale triggers more than one hook action, these hook actions can be observed individually by multiple observers or together by a single observer. Consider the following processes:

$$P_0 \xrightarrow{x} P_1 \qquad Q_0 \xrightarrow{y} Q_1$$

$$A_0 \underset{s[x]}{\overset{a[x]}{\rightrightarrows}} A_1 \qquad B_0 \underset{s[y]}{\overset{b[y]}{\leftleftarrows}} B_1 \qquad R_0 \underset{x}{\overset{\{x,y\}}{\rightrightarrows}} R_1$$

Processes A_0 and B_1 can produce the following transition:

$$A_0 \underset{\{\!|s|\!\}}{\bowtie} B_1 \xrightarrow{\{\!|s|\!\}[\{\!|x,y|\!\}]} A_1 \underset{\{\!|s|\!\}}{\bowtie} B_0$$

In this case, two different instances of s synchronise, their set of hook actions merge in the resulting activity. Now consider the addition of processes P_0, Q_0 and R_0. Two possible examples of transition are:

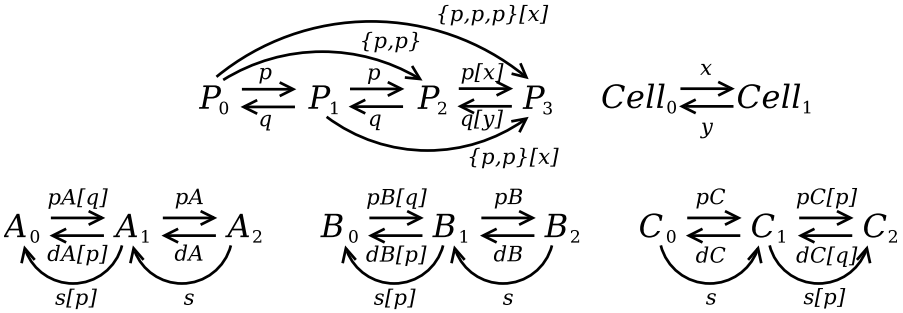
$$(A_0 \underset{\{\!|x|\!\}}{\bowtie} P_0) \underset{\{\!|s|\!\}}{\bowtie} (B_1 \underset{\{\!|y|\!\}}{\bowtie} Q_0) \xrightarrow{\{\!|s,x,y|\!\}[\emptyset]} (A_1 \underset{\{\!|x|\!\}}{\bowtie} P_1) \underset{\{\!|s|\!\}}{\bowtie} (B_0 \underset{\{\!|y|\!\}}{\bowtie} Q_1)$$

$$(A_0 \underset{\{\!|s|\!\}}{\bowtie} B_1) \underset{\{\!|x,y|\!\}}{\bowtie} R_0 \xrightarrow{\{\!|s,x,y|\!\}[\emptyset]} (A_1 \underset{\{\!|s|\!\}}{\bowtie} B_0) \underset{\{\!|x,y|\!\}}{\bowtie} R_1$$

In the first transition, hook actions x and y are observed individually by processes P_0 and Q_0. If only hook action x were present, it would still be observed by P_0 and the same for y with Q_0. In the second transition, hook actions x and y are observed at the same time by process R_0. Most importantly, we impose that R_0 can synchronise *only* using the transition that includes the largest number of hooks available. In other words, transition $\xrightarrow{\{\!|x|\!\}}$ from R_0 cannot synchronised with $\xrightarrow{\{\!|s|\!\}[\{\!|x,y|\!\}]}$, because R_0 can perform

$\xrightarrow{\{|x,y|\}}$. This mechanism allows the modeller to design specific responses to events that trigger multiple multi-scale events at the same time.

Example 6. In some cases, a scale may trigger a multi-set of hook actions. Consider the following example. A cell contains three biochemical species **A**, **B** and **C**. The cell changes its behaviour if the biochemical scale reaches a specific configuration, that is when the concentrations of **A** and **B** are low and when the concentration of **C** is high. Species **A**, **B** and **C** are produced (actions pA, pB and pC) and degrade (actions dA, dB and dC) in the cell. Finally, species **A**, **B** and **C** are involved in the biochemical reaction $R_s : \mathbf{A} + \mathbf{B} \rightarrow \mathbf{C}$. The model is defined as:



The initial state is:

$$((A_1 \underset{\{|s|\}}{\bowtie} B_1 \underset{\{|s|\}}{\bowtie} C_1) \underset{\{|p,p,p,q|\}}{\boxtimes} P_0) \underset{\{|x,y|\}}{\boxtimes} Cell_0$$

Again we use processes P_i, $i \in \{0, \ldots, 3\}$, to count the concentration required for change at the cellular scale. In this particular case, a single transition can involve more than one identical hook action (i.e. p). The transition is the following:

$$\begin{array}{c}((A_1 \underset{\{|s|\}}{\bowtie} B_1 \underset{\{|s|\}}{\bowtie} C_1) \underset{\{|p,p,p,q|\}}{\boxtimes} P_0) \underset{\{|x,y|\}}{\boxtimes} Cell_0 \xrightarrow{\{|s,p,p,p,x|\}[\emptyset]} \\ ((A_0 \underset{\{|s|\}}{\bowtie} B_0 \underset{\{|s|\}}{\bowtie} C_2) \underset{\{|p,p,p,q|\}}{\boxtimes} P_3) \underset{\{|x,y|\}}{\boxtimes} Cell_1\end{array}$$

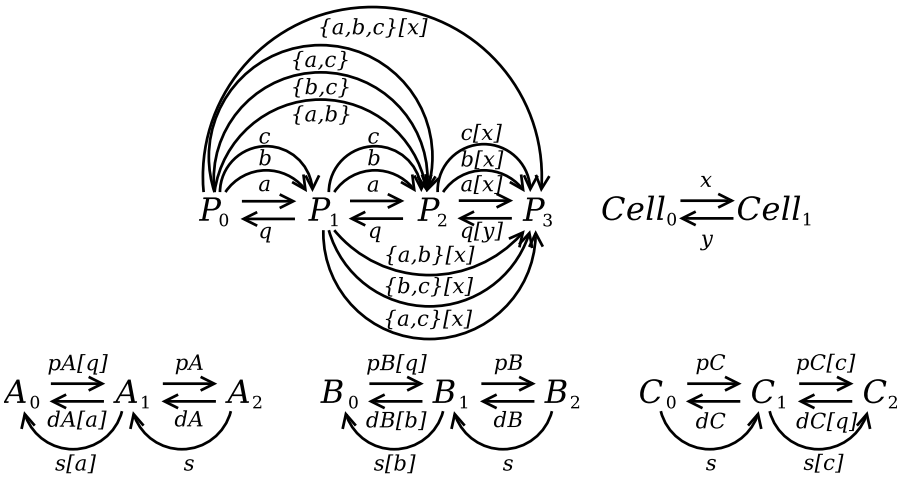
Another transition of the derivation graph generated by this model is:

$$\begin{array}{c}((A_2 \underset{\{|s|\}}{\bowtie} B_1 \underset{\{|s|\}}{\bowtie} C_1) \underset{\{|p,p,p,q|\}}{\boxtimes} P_1) \underset{\{|x,y|\}}{\boxtimes} Cell_0 \xrightarrow{\{|s,p,p,x|\}[\emptyset]} \\ ((A_1 \underset{\{|s|\}}{\bowtie} B_0 \underset{\{|s|\}}{\bowtie} C_2) \underset{\{|p,p,p,q|\}}{\boxtimes} P_3) \underset{\{|x,y|\}}{\boxtimes} Cell_1\end{array}$$

The multiplicity of the vertical cooperation set $\{\!|p,p,p,q|\!\}$ is necessary. In the above transition, the biochemical scale performs transition $\xrightarrow{\{\!|s|\!\}[\{\!|p,p|\!\}]}$, while P_1 can either perform transition $\xrightarrow{\{\!|p|\!\}}$ or $\xrightarrow{\{\!|p,p|\!\}[x]}$. In order to choose the correct transition, as seen in Example 5, we require that the multi-set of actions from P_1 is included in both the multi-set of hooks and the cooperation set, that is $\{\!|p,p|\!\} \subseteq \{\!|p,p|\!\} \cap \{\!|p,p,p,q|\!\}$.

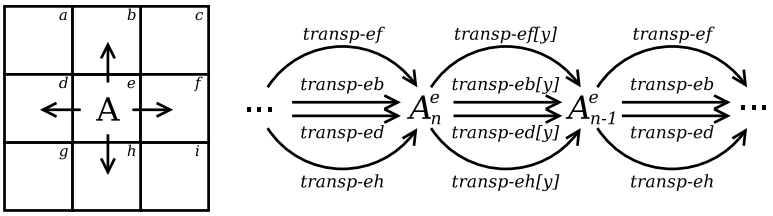
The use of multi-sets allows for a compact definition of this model, where we use the same action p to indicate that either **A**, **B** or **C** have reached the concentration required for a change in behaviour of the cell.

Without multi-sets, an alternative definition of the same model is as follows:



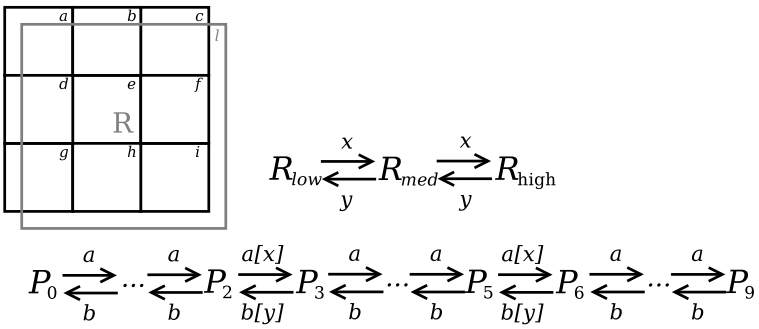
The above version of the model uses actions a, b and c in place of p creating a combinatorial problem. As a result, the derivation graph of process P_0 presents 19 transitions instead of nine.

Example 7. The positioning of hook actions on actions at the biochemical scale is particularly useful when geometrical space is considered. Let A^e_n denote the process representing a concentration level n of species **A** in region R_e. Concentration can migrate to and from region R_e and many different transport actions will have the same effect of lowering or increasing the concentration of **A** in one region, as shown in the following diagram (only outgoing transport shown):



The concentration of **A** is decreased, from A^e_n to A^e_{n-1}, through a transport action of the form *transp-es*, $s \in \{b, d, f, h\}$. Correspondingly, at region R_s, the concentration of **A** increases, from A^s_m to A^s_{m+1}. If we want to denote that a threshold is crossed when passing from level n to $n-1$ of **A** at R_e, we can add a hook action to the four transport actions, *transp-es*, obtaining *transp-es*$[y]$.

Example 8. In this example we show how to abstract multiple regions to a single region, with respect to a specific property. Consider an area R_l of 3×3 regions, labelled from R_a to R_i. We ignore detail of the biochemical reactions, but assume that at some point each of these locations can become *infected*, exposing hook action a, or they can *recover*, exposing hook action b. We are not interested in which region has changed its status, only the number infected in R_l. A scale that represents the degree of infection of area R_l is defined by three processes, R_{low}, R_{med} and R_{high}.



Processes P_i, $i \in \{0, \ldots, 9\}$ are used to count the number of infected regions. If the number of infected regions is between 0 and 2, the degree of infection is low; between 3 and 5 it is medium; larger than 5 it is high. Hooks x and y identify transitions between stages of infection.

3.5. Examples of Equivalences

Example 9. Consider the following processes:

Figure 4: Filtered transition graphs of the example in this section. If $r_d = r_a + r_e = r$ the transition systems are Markovian $\mathcal{T}$-bisimilar.

$$\begin{array}{llll} A_0 \triangleq a[b].A_1 + e[b].A_1 & D_0 \triangleq b[c].D_1 & A_1 \triangleq nil & D_1 \triangleq nil \\ B_0 \triangleq b[c].B_1 & E_0 \triangleq d.E_1 & B_1 \triangleq nil & E_1 \triangleq nil \\ C_0 \triangleq d[b].C_1 & & C_1 \triangleq nil & \end{array}$$

Consider two PAH processes: $A_0 \underset{\{\!|b|\!\}}{\bowtie} B_0$ and $(C_0 \underset{\{\!|b|\!\}}{\bowtie} D_0) \underset{\{\!|d|\!\}}{\bowtie} E_0$. The following transitions are possible:

$$A_0 \underset{\{\!|b|\!\}}{\bowtie} B_0 \xrightarrow{(\{\!|a,b|\!\}[c],\Delta)} A_1 \underset{\{\!|b|\!\}}{\bowtie} B_1$$
$$A_0 \underset{\{\!|b|\!\}}{\bowtie} B_0 \xrightarrow{(\{\!|e,b|\!\}[c],\Delta'')} A_1 \underset{\{\!|b|\!\}}{\bowtie} B_1$$
$$(C_0 \underset{\{\!|b|\!\}}{\bowtie} D_0) \underset{\{\!|d|\!\}}{\bowtie} E_0 \xrightarrow{(\{\!|d,b|\!\}[c],\Delta')} (C_1 \underset{\{\!|b|\!\}}{\bowtie} D_1) \underset{\{\!|d|\!\}}{\bowtie} E_1$$

These are the only possible transitions. We cannot consider the two processes equivalent in the sense that they generate isomorphic transition graphs, so $A_0 \underset{\{\!|b|\!\}}{\bowtie} B_0 \not\equiv (C_0 \underset{\{\!|b|\!\}}{\bowtie} D_0) \underset{\{\!|d|\!\}}{\bowtie} E_0$. However, if we decide to select only actions in the multi-set $\mathcal{T} = \{\!|b|\!\}$, and rating yields rates r_a, r_e and r_d, we obtain transitions:

$$A_0 \underset{\{\!|b|\!\}}{\bowtie} B_0 \xrightarrow{(\{\!|b|\!\},\{\!|c|\!\},r_a)}_{\mathcal{T}} A_1 \underset{\{\!|b|\!\}}{\bowtie} B_1$$
$$A_0 \underset{\{\!|b|\!\}}{\bowtie} B_0 \xrightarrow{(\{\!|b|\!\},\{\!|c|\!\},r_e)}_{\mathcal{T}} A_1 \underset{\{\!|b|\!\}}{\bowtie} B_1$$
$$(C_0 \underset{\{\!|b|\!\}}{\bowtie} D_0) \underset{\{\!|d|\!\}}{\bowtie} E_0 \xrightarrow{(\{\!|b|\!\},\{\!|c|\!\},r_d)}_{\mathcal{T}} (C_1 \underset{\{\!|b|\!\}}{\bowtie} D_1) \underset{\{\!|d|\!\}}{\bowtie} E_1$$

We call *filtering* the operation of selecting actions on labels and *filtered transitions* the resulting transitions. If $r_d = r_a + r_e = r$, both $A_0 \underset{b}{\bowtie} B_0$ and $(C_0 \underset{\{\!|b|\!\}}{\bowtie} D_0) \underset{\{\!|d|\!\}}{\bowtie} E_0$ can move to a terminal state with filtered set of layer actions $\{\!|b|\!\}$ and set of unused hook actions $\{\!|c|\!\}$ with a total rate of r. In other words, the pair $(\{\!|b|\!\}, \{\!|c|\!\})$ appears on an activity at the same time with the same probability, implying the two model processes are Markovian $\mathcal{T}$-bisimilar, written $A_0 \underset{\{\!|b|\!\}}{\bowtie} B_0 \simeq_{\mathcal{T}} (C_0 \underset{\{\!|b|\!\}}{\bowtie} D_0) \underset{\{\!|d|\!\}}{\bowtie} E_0$ (Figure 4). We will demonstrate $\simeq_{\mathcal{T}}$ is a congruence for process algebra with hooks processes.

4. Process Algebra with Hooks

A preliminary version of process algebra with hooks (PAH) has been published in [31]. In this early work we followed a *bottom-up* approach where the biochemical scale determines the rates and other scales are abstractions of lower scales. The current syntax and almost identical semantics were first introduced in [1], where we followed a *middle-out* [25] approach and where one can begin modelling at any scale, and then relate to higher or lower scales.

The syntax of PAH is:

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

$$M ::= A \mid M \underset{\mathcal{L}}{\bowtie} M \mid M \underset{\mathcal{L}}{\Subset} M$$

where:

- 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. $\mathbb{P}_d \cup \mathbb{P}_m = \mathbb{P}$ and $\mathbb{P}_d \cap \mathbb{P}_m = \emptyset$, with $\mathbb{P}$ the set of processes;
- Agents are defined as $A \triangleq D$, that is we use definition processes to define the behaviour of agents. This definition has to be unique for each agent;
- a model is defined by a model process M, which in turn is either an agent A, a horizontal cooperation between model processes $M \underset{\mathcal{L}}{\bowtie} M$ or a vertical cooperation between model processes $M \underset{\mathcal{L}}{\Subset} M$;
- action execution $\mathcal{A}[\mathcal{E}].A$ 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;
- 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;
- $\mathcal{L}$, $\mathcal{A}$ and $\mathcal{E}$ are multi-sets of actions, with $\mathcal{L} = (\mathcal{L}', m_{\mathcal{L}})$, $\mathcal{A} = (\mathcal{A}', m_{\mathcal{A}})$ and $\mathcal{E} = (\mathcal{E}', m_{\mathcal{E}})$. Definitions of multi-sets and operations on multi-sets are given in Appendix A. Moreover, $\mathcal{L}' \subseteq Actions$, $\mathcal{A}' \subseteq Actions \wedge \mathcal{A} \neq \emptyset$, $\mathcal{E}' \subseteq Actions \wedge |\mathcal{E}'| \leq 1$, with $Actions$ the set of actions;

- $\mathcal{A}[\mathcal{E}]$ is a composed action. Actions in $\mathcal{A}$ are called *layer* actions, while actions in $\mathcal{E}$ are called *hook* actions;
- *nil* is the terminated process;
- $\mathcal{A}[\mathcal{E}].A$ expresses the fact that the composed action $\mathcal{A}[\mathcal{E}]$ has to be performed in order to change process $\mathcal{A}[\mathcal{E}].A$ into the new process A;
- $D + D$ expresses the non deterministic choice between two processes. Once one is chosen, the other is discarded;
- $M \underset{\mathcal{L}}{\bowtie} M$ expresses the horizontal cooperation between two independent processes on *the same* scale via the cooperation multi-set $\mathcal{L}$. It is a symmetrical operator;
- $M \underset{\mathcal{L}}{\Bowtie} M$ expresses the vertical cooperation between two independent processes on *different* scales via the cooperation multi-set $\mathcal{L}$. It is a symmetrical operator;

Conventions for the notation of actions are as follows. Given a composed action $\mathcal{A}[\mathcal{E}]$, if $|\mathcal{A}| = 1$ or $|\mathcal{E}| = 1$, 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 $\mathcal{A}[\emptyset]$ can be written $\mathcal{A}$.

The termination process *nil* is used in the definition of agents that cannot perform any action. These agents are still associated with a variable and a value. For example, a biological entity may represent the differentiation of a cell and such differentiation may be uncertain until a final, irreversible decision is made, represented by reaching an agent defined by *nil*.

The semantics of PAH is defined by the derivation rules in Figure 5. Rule **Prefix** is an axiom that expresses that process $\mathcal{A}[\mathcal{E}].A$ can become agent process A via the execution of composed action $\mathcal{A}[\mathcal{E}]$. Although we restrict the set $\mathcal{E}$ in the syntax to be either empty or a singleton, this set can merge with others upon the application of rules **Layer Synchronisation**, **Vertical Synchronisation Left** and **Vertical Synchronisation Right**, producing a multi-set of hooks. We use multi-sets to allow a more general and flexible composition both within and between scales. Example 6 in Section 3 provides an example of these compositions.

Rules **Choice Left** and **Choice Right** express choice between the execution of composed actions.

Prefix

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

Agent

$$\frac{D \xrightarrow{\mathcal{A}[\mathcal{E}]} A' \quad A \triangleq D}{A \xrightarrow{(\mathcal{A}[\mathcal{E}],\Delta)} A'} \ \wedge \Delta = \{(Var(A), Val(A))\}$$

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

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

Asynchronous Right

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

Layer Synchronisation

$$\frac{M_1 \xrightarrow{(\mathcal{A}[\mathcal{E}],\Delta_1)} M_1' \quad M_2 \xrightarrow{(\mathcal{B}[\mathcal{F}],\Delta_2)} M_2'}{M_1 \underset{\mathcal{L}}{\bowtie} M_2 \xrightarrow{(\mathcal{A}\cup\mathcal{B}[\mathcal{E}\uplus\mathcal{F}],\Delta_1\cup\Delta_2)} M_1' \underset{\mathcal{L}}{\bowtie} M_2'} \ \mathcal{A} \cap \mathcal{B} \cap \mathcal{L} \neq \emptyset$$

Vertical Asynchronous Left

$$\frac{M_1 \xrightarrow{(\mathcal{A}[\mathcal{E}],\Delta)} M_1'}{M_1 \underset{\mathcal{L}}{\boxtimes} M_2 \xrightarrow{(\mathcal{A}[\mathcal{E}],\Delta)} M_1' \underset{\mathcal{L}}{\boxtimes} M_2} \ \mathcal{A} \cap \mathcal{L} = \emptyset \ \wedge \mathcal{E} \cap \mathcal{L} = \emptyset$$

Vertical Asynchronous Right

$$\frac{M_2 \xrightarrow{(\mathcal{B}[\mathcal{F}],\Delta)} M_2'}{M_1 \underset{\mathcal{L}}{\boxtimes} M_2 \xrightarrow{(\mathcal{B}[\mathcal{F}],\Delta)} M_1 \underset{\mathcal{L}}{\boxtimes} M_2'} \ \mathcal{B} \cap \mathcal{L} = \emptyset \ \wedge \mathcal{F} \cap \mathcal{L} = \emptyset$$

Vertical Synchronisation Left

$$\frac{M_1 \xrightarrow{(\mathcal{A}[\mathcal{E}],\Delta_1)} M_1' \quad M_2 \xrightarrow{(\mathcal{B}[\mathcal{F}],\Delta_2)} M_2'}{M_1 \underset{\mathcal{L}}{\boxtimes} M_2 \xrightarrow{(\mathcal{A}\cup\mathcal{B}[(\mathcal{E}\setminus\mathcal{B})\uplus\mathcal{F}],\Delta_1\cup\Delta_2)} M_1' \underset{\mathcal{L}}{\boxtimes} M_2'} \ \begin{array}{l}\mathcal{B} \subseteq \mathcal{E} \cap \mathcal{L} \ \wedge \ \neg(\exists M_2 \xrightarrow{(\mathcal{B}'[\mathcal{F}'],\Delta_2')} M_2''. \\ (\mathcal{B}' \subseteq \mathcal{E} \cap \mathcal{L}) \wedge (|\mathcal{B}'| > |\mathcal{B}|))\end{array}$$

Vertical Synchronisation Right

$$\frac{M_1 \xrightarrow{(\mathcal{A}[\mathcal{E}],\Delta_1)} M_1' \quad M_2 \xrightarrow{(\mathcal{B}[\mathcal{F}],\Delta_2)} M_2'}{M_1 \underset{\mathcal{L}}{\boxtimes} M_2 \xrightarrow{(\mathcal{A}\cup\mathcal{B}[(\mathcal{F}\setminus\mathcal{A})\uplus\mathcal{E}],\Delta_1\cup\Delta_2)} M_1' \underset{\mathcal{L}}{\boxtimes} M_2'} \ \begin{array}{l}\mathcal{A} \subseteq \mathcal{F} \cap \mathcal{L} \ \wedge \ \neg(\exists M_1 \xrightarrow{(\mathcal{A}'[\mathcal{E}'],\Delta_1')} M_1''. \\ (\mathcal{A}' \subseteq \mathcal{F} \cap \mathcal{L}) \wedge (|\mathcal{A}'| > |\mathcal{A}|))\end{array}$$

Figure 5: Stochastic semantics of process algebra with hooks. Union of multi-sets is denoted by $\cup$, while sum of multi-sets is denoted by $\uplus$.

Rule **Agent** replaces definition processes with agent processes and constructs environment Δ using variables and values associated with agents.

Rule **Layer Synchronisation** is a weaker version of typical multi-way synchronisation in process algebra. In fact, this rule can be applied even if the labels of the transitions from processes M_1 and M_2 are not identical: it is only required that multi-sets $\mathcal{A}$ and $\mathcal{B}$ share at least a name and that this name is also in $\mathcal{L}$. The resulting transition presents the multi-set union of multi-sets of layer actions $\mathcal{A}$ and $\mathcal{B}$, to represent the result of the synchronisation. Conversely, multi-set sum of multi-sets of hooks $\mathcal{E}$ and $\mathcal{F}$ is used to represent the collection of hooks summing the multiplicity of the hook actions. Moreover, partial functions Δ_1 and Δ_2 are merged assuming no clashing of variable names. In rules **Asynchronous Left** and **Asynchronous Right**, processes in cooperation can proceed asynchronously only if action set $\mathcal{A}$ does not share actions with cooperation set $\mathcal{L}$.

The behaviour induced by the $\underset{\mathcal{L}}{\boxtimes}$ operator is regulated by the rules **Vertical Synchronisation Left**, **Vertical Synchronisation Right**, **Vertical Asynchronous Left** and **Vertical Asynchronous Right**. Rules differing in the name only by **Left** and **Right** are symmetric, so we explain only one of them. In **Vertical Synchronisation Left**, the synchronisation is between the multi-set of hook actions on the left hand side ($\mathcal{E}$) and the multi-set of layer actions on the right hand side ($\mathcal{B}$), via actions in the cooperation multi-set $\mathcal{L}$. More specifically, some inter-scale actions in $\mathcal{E}$ are interpreted by another scale via $\mathcal{B}$. For this to happen we impose in the side rule that $\mathcal{B}$ must be included in both $\mathcal{E}$ and $\mathcal{L}$. The resulting transition presents the multi-set union of multi-sets $\mathcal{A}$ and $\mathcal{B}$, while $\mathcal{B}$ is subtracted from $\mathcal{E}$ to represent the fact that some of the hook actions of the left hand side have been used. Multi-set sum is used between $\mathcal{E} \setminus \mathcal{B}$ and $\mathcal{F}$ to collect the remaining hook actions. It may be that more than one transition from M_2 presents a multi-set of suitable layer actions $\mathcal{B}$. In this case, we consider the largest $\mathcal{B}$ multi-sets, imposed by the side condition, which states that there is no other transition from M_2 which presents a multi-set of layer actions $\mathcal{B}'$ included in both $\mathcal{E}$ and $\mathcal{L}$ that is larger than $\mathcal{B}$. This gives the possibility to the modeller to choose how the model should behave when multiple hook actions are offered in a single transition (see Examples 5 and 6 in Section 3). Finally, in **Vertical Synchronisation Left** we collect environments Δ_1 and Δ_2.

Consider now the inference rule **Vertical Asynchronous Left**. In this case, we allow a single process to transition asynchronously only if there are

no actions in the multi-set $\mathcal{A}$ which are also contained in $\mathcal{L}$ and no actions in the hook multi-set $\mathcal{E}$ contained in $\mathcal{L}$. This is because the actions in the vertical cooperation set $\mathcal{L}$ are hooks and if $\mathcal{A} \cap \mathcal{L} \neq \emptyset$ (or $\mathcal{E} \cap \mathcal{L} \neq \emptyset$) then $\mathcal{A}$ (or $\mathcal{E}$) contains hooks and the transition is not intended to be used asynchronously, but only with a suitable transition from M_2.

We introduce now definitions necessary to define the derivation graph for PAH processes.

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

Definition 2. *One step derivative.* Given $P \in \mathbb{P}$, if $P \xrightarrow{a} P'$ then P' is a one step derivative of P. We denote the set of one step derivatives of P as $osds(P) = \{P' \mid P \xrightarrow{a} P'\}$.

Definition 3. *Derivative.* Given $M_i \in \mathbb{P}_m$, If $M_i \xrightarrow{a} \ldots \xrightarrow{a'} M_j$ then M_j is a derivative of M_i.

Definition 4. *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 $M_i \xrightarrow{(\mathcal{A}[\mathcal{E}],\Delta)} M_j$ then $M_j \in ds(M)$.

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

- $(a, P') \in Moves(P)$ iff $P \xrightarrow{a} P'$, with the same multiplicity as the number of derivation trees that can derive $P \xrightarrow{a} P'$ using the derivation rules in Figure 5.

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:

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

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 $M' \in ds(M)$, $(M', \mathcal{A}[\mathcal{E}], \Delta, M'') \in \rightarrow$ with the same multiplicity as $((\mathcal{A}[\mathcal{E}], \Delta), M'')$ in $Moves(M')$.

4.1. Functional Rates

Functional rates are arithmetical expressions used to define rates of biological events that are parametric with respect to the current state of the system. In order to do so, functional rates contain parameter names, the value of which depend on the environment Δ in an activity $(\mathcal{A}[\mathcal{E}], \Delta)$ and on an additional environment Γ of constant model parameters. Functional rates are associated with actions. Because we use sets of actions, in Section 4.2 we introduce constraints that ensure that at most one functional rate is associated with each transition. The syntax of functional rates is given by:

$$f ::= k \mid i \mid f \; binop \; f \mid unop(f)$$

$$binop ::= + \mid - \mid * \mid / \mid \wedge \qquad unop ::= exp \mid log \mid sin \mid cos$$

- $k \in \mathbb{R}$ and $i \in Names$, i.e. i is a parameter name;
- f is a functional rate, $f \in \mathbb{F}$;
- exp is the base e exponential operator;
- $\wedge$ is the binary exponential operator.

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 semantics of arithmetical expressions. Given an environment $\Delta' \subseteq Names \times \mathbb{R}$, and a functional rate f, f evaluates to $k \in \mathbb{R}$ iff $\Delta' \vdash f \to k$ is valid. In practice, the environment Δ' is the partial function obtained from the union $\Delta' = \Delta \cup \Gamma$ of the environment of constant model parameters Γ and the environment Δ in the activity to be rated.

4.2. PAH Model and Well-Formed PAH Model

We proceed now to the definition of a PAH model and well-formed PAH model.

Definition 9. PAH *model*. A PAH model is a tuple:

$$(AgentDef, M, Actions, Names, \mathbb{F}, \Gamma, Part, Var, Val)$$

where:

- *AgentDef* is the finite set of agent definitions $\{A_1 \triangleq D_1, A_2 \triangleq D_2, \dots\}$;
- M is the initial state of the model, with $M \in \mathbb{P}_m$;
- *Actions* is the finite set of actions;
- $Names$ is the finite set of parameter names;
- $\mathbb{F}$ is the finite set of functional rates;
- Γ is a partial function that associates parameters names with their values, with $\Gamma \subseteq Names \times \mathbb{R}$. This partial function contains constant model parameters;
- $Part$ is the finite set of sets of participants;
- *Var* and *Val* are the functions associating agents with variables (i.e. parameter names) and values, with $Var : \mathbb{P}_m \to Names$ and $Val : \mathbb{P}_m \to \mathbb{R}$.

In order to ensure a correct and unambiguous rate evaluation (Section 4.3) and to guarantee that congruence relations (Section 6.1) can be defined on PAH processes, we consider only well-formed PAH models, which are characterised as follows.

Definition 10. *Well formed* PAH *model.* A PAH model is well formed if and only if:

1. each functional rate $f_a \in \mathbb{F}$ is associated with a set of participants $p_a \subseteq Names$:

$$\forall a \in Actions,\ f_a \in \mathbb{F} \Leftrightarrow p_a \in Part$$

2. at any time only one agent can be associated with a certain variable. 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;

3. whenever an agent A performs an action (application of derivation rule **Agent**), the resulting agent A' will be associated with the same variable A is associated with. Given a definition of an agent A as a choice of prefix actions of the form

$$A \triangleq \sum_i a_i.A_i$$

then $\forall A_i\ Var(A) = Var(A_i)$;

4. if an agent A can perform activity $(\mathcal{A}[\mathcal{E}], \Delta)$ and $\mathcal{A}$ contains an action a associated with a functional rate f_a, then $\mathcal{A} = \{\!|a|\!\}$. Moreover, if an action is used as hook it cannot be associated with a functional rate. $\forall A$ agents defined as

$$A \triangleq \sum_i \mathcal{A}_i[\mathcal{H}_i].A_i$$

$\forall a$ s.t. $f_a \in \mathbb{F}$, if $a \in \mathcal{A}_i$ then $\mathcal{A}_i = \{\!|a|\!\}$ and $\forall a$ s.t. $f_a \in \mathbb{F},\ a \notin \mathcal{H}_i$.

5. an agent A or one of its derivatives can perform action a if and only if A is associated with a variable in p_a. $\forall a$ s.t. $f_a \in \mathbb{F}$, $\forall A$ agents

$$\exists \mathcal{E}, \Delta \text{ s.t. } (a[\mathcal{E}], \Delta) \in \overrightarrow{Activities}(A) \Leftrightarrow Var(A) \in p_a$$

6. whenever $M_1 \underset{\mathcal{L}}{\rtimes\!\ltimes} M_2$ then M_1 and M_2 do not contain $\underset{\mathcal{L}}{\bowtie}$.

A well-formed PAH model implies the following propositions.

Proposition 11. Given a well-formed PAH model, at any time only one agent is associated with a certain variable name, i.e. given $M \in \mathbb{P}_m$ initial state of the PAH model, $\forall M' \in ds(M)$, point 2 of Definition 10 holds.

Proof. This is follows from points 2 and 3 in Definition 10 and by the syntax and semantics of PAH that ensure that given an initial state, no transition can increase or decrease the number of processes.

Proposition 12. Given a well-formed PAH model, environments Δ_1 and Δ_2 in Layer Synchronisation, **Vertical Synchronisation Left** and **Vertical Synchronisation Right** rules, always contain disjoint parameter names, ensuring no clashes of names in the union $\Delta_1 \cup \Delta_2$, i.e. $\forall i, j \in Names$, with $(i, k_1) \in \Delta_1$ and $(j, k_2) \in \Delta_2$, then $i \neq j$.

Proof. This follows from Proposition 11.

Proposition 13. Given a well-formed PAH model with initial state M, in each transition $M' \xrightarrow{(\mathcal{A}[\mathcal{E}],\Delta)} M''$ with $M' \in ds(M)$, $\mathcal{A}$ contains at most one action associated with a functional rate.

Proof. This follows from points 4 and 6 in Definition 10. Given a composed action $\mathcal{A}[\mathcal{E}]$, Point 4 in Definition 10, ensures that before any synchronisation is applied, if $\mathcal{A}$ contains a such that $f_a \in \mathbb{F}$, then $\mathcal{A} = \{\!|a|\!\}$, while $\mathcal{E}$ cannot contain actions associated with functional rates. The only way to add other actions to $\mathcal{A}$ is via vertical synchronisation, because horizontal synchronisation can happen only with other $\{\!|a|\!\}$ multi-sets. The application of a vertical synchronisation can only add to $\mathcal{A}$ actions that were previously in a hook multi-set, and, by point 4 in Definition 10, are not associated with functional rates. After a vertical synchronisation, the only way to merge $\mathcal{A}$ with a different multi-set containing an action associated with a functional rate, would be to apply a horizontal synchronisation. However, this is impossible because of point 6 in Definition 10.

Proposition 14. Given a well-formed PAH model, no more than $|p_a|$ agents can synchronise via action a.

Proof. This follows from point 5 in Definition 10 and Proposition 11.

4.3. Rating PAH Activities

In this section we formalise the concepts of open and closed activities that we introduced in Section 3.2 as well as rating of PAH activities.

Formally, an activity $(\mathcal{A}[\mathcal{E}], \Delta)$ can be rated only if $\mathcal{A}$ contains exactly one action name a such that $f_a \in \mathbb{F}$ and Δ contains the variables in p_a. Such activity is called *closed*. An activity that is not closed is called *open*.

Additionally, when multiple transitions from a certain state are associated with the same functional rate we impose that the evaluated rate has to be divided by the number of such transitions. This situation can arise as a result of non-deterministic vertical synchronisations.

Definition 15. *Current closed activities of a model process.* Given a set of functional rates $\mathbb{F}$ and a set of sets of participants $Part$, activity $(\mathcal{A}[\mathcal{E}], \Delta)$ is *closed* iff the following points are both true:

- there exists unique a s.t. $a \in \mathcal{A}$ and $f_a \in \mathbb{F}$;
- the corresponding $p_a \in Part$ is included in the variables in Δ.

Given a model process $M \in \mathbb{P}_m$, the set of closed activities of M is denoted by $ClosedAct(M)$ and is defines as:

$$ClosedAct(M) = \left\{ \left| (\mathcal{A}[\mathcal{E}], \Delta) \;\middle|\; \begin{array}{c} (\mathcal{A}[\mathcal{E}], \Delta) \in Activities(M) \wedge \exists! a \; s.t. \; (a \in \mathcal{A} \\ \wedge f_a \in \mathbb{F}) \wedge (p_a \in Part \wedge p_a \subseteq \{i | (i,k) \in \Delta\}) \end{array} \right| \right\}$$

Definition 16. *Current open activities of a model process.* Given a model process $M \in \mathbb{P}_m$, the multi-set of closed activities that M can perform is defined as:

$$OpenAct(M) = Activities(M) \setminus ClosedAct(M)$$

Definition 17. *Closed and open activity sets.* The multi-sets of all closed and open activities that a model process $M \in \mathbb{P}_m$ can perform are defined as:

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

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

Definition 18. *Open moves of a model process.* Given a model process $M \in \mathbb{P}_m$, the multi-set of open moves of M, denoted $OpenMoves(M)$, is defined as:

$$OpenMoves(M) = \{\!|(a, M') \mid (a, M') \in Moves(M) \wedge a \in OpenAct(M)|\!\}$$

Definition 19. *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 20. *Current rated activities of a model process.* Given an environment of constant model parameters $\Gamma \subseteq Names \times \mathbb{R}$ and a model process $M \in \mathbb{P}_m$, the current set of rated activities of M is defined as:

$$RatedAct(M)_\Gamma = \left\{\!\!\left| (\mathcal{A}[\mathcal{E}], r) \;\middle|\; \begin{array}{l} (\mathcal{A}[\mathcal{E}], \Delta) \in ClosedAct(M) \wedge f_a \in \mathbb{F} \wedge a \in \mathcal{A} \\ \wedge\, \Gamma \cup \Delta \vdash f_a \rightarrow k \wedge r = k/\pi(ClosedAct(M), a) \end{array} \right|\!\!\right\}$$

where $\pi(A, a)$ returns the number of occurrences of $(\mathcal{B}[\mathcal{F}], \Delta')$ in the multi-set A such that $a \in \mathcal{B}$.

Definition 21. *Rated activity set.* Given an environment of constant model parameters $\Gamma \subseteq Names \times \mathbb{R}$, the multi-set of rated activities that a model process $M \in \mathbb{P}_m$ and its derivatives can perform is given by:

$$\overrightarrow{RatedAct}(M)_\Gamma = \biguplus_{M_i \in ds(M)} RatedAct(M_i)$$

$\overrightarrow{RatedAct}(M)_\Gamma$ can be written $\overrightarrow{RatedAct}(M)$ if Γ is clear from the context.

Definition 22. *Rated moves of a model process.* Given a model process $M \in \mathbb{P}_m$ and an environment of constant model parameters $\Gamma \subseteq Names \times \mathbb{R}$, the multi-set of rated moves of M, denoted $RatedMoves(M)_\Gamma$, is defined as:

$$RatedMoves(M)_\Gamma = \left\{\!\!\left| ((\mathcal{A}[\mathcal{E}], r), M') \;\middle|\; \begin{array}{l} ((\mathcal{A}[\mathcal{E}], \Delta), M') \in Moves(M) \\ \wedge (\mathcal{A}[\mathcal{E}], r) \in RatedAct(M)_\Gamma \end{array} \right|\!\!\right\}$$

Definition 23. *Rated derivation graph.* Given a model process $M \in \mathbb{P}_m$ and an environment of constant model parameters $\Gamma \subseteq Names \times \mathbb{R}$, 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 $\rightarrow_\Gamma \subseteq ds(M) \times \overrightarrow{RatedAct}(M)_\Gamma \times ds(M)$. Given $M' \in ds(M)$, $(M', \mathcal{A}[\mathcal{E}], r, M'') \in \rightarrow_\Gamma$ with the same multiplicity as $((\mathcal{A}[\mathcal{E}], r), M'')$ in $RatedMoves(M')_\Gamma$.
- multi-set of labelled transitions $\rightarrow_o \subseteq ds(M) \times \overrightarrow{OpenAct}(M) \times ds(M)$. Given $M' \in ds(M)$, $(M', \mathcal{A}[\mathcal{E}], \Delta, M'') \in \rightarrow_o$ with the same multiplicity as $((\mathcal{A}[\mathcal{E}], \Delta), M'')$ in $OpenMoves(M')$.

$\mathcal{D}_r(M)_\Gamma$ can be written $\mathcal{D}_r(M)$ if Γ is clear from the context.

5. Relating Biological Systems to Aid Model Development

The development of PAH models can become overwhelming when an increasing number of interactions between scales is considered. Ensuring consistency of interactions between scales is an issue that affects multi-scale models in general, often requires *manual* checking and is specific to the model at hand. But, in a process algebraic framework, there is an elegant solution in the form of a relation that describes compatibility between scales and can be applied to any PAH model.

In this section we introduce the *compatibility $\mathcal{L}$-bisimulation* ($\asymp_\mathcal{L}$), that ensures that two model processes can interact correctly via a vertical synchronisation with cooperation set $\mathcal{L}$ (i.e. $\underset{\mathcal{L}}{\bowtie}$). Two processes interact correctly when whenever one process can perform a composed action $\mathcal{A}[\mathcal{E}]$ with hooks $\mathcal{E}$ that are present in the cooperation set $\mathcal{L}$, then the other process is able to synchronise with that action. If the condition is not satisfied, then the two scales are not compatibility $\mathcal{L}$-bisimilar. There is an error in the definition of the interactions between scales that the modeller needs to rectify.

In order to define this relation as bisimulation, we need to ensure that every transition the first process can perform is matched by transitions of the second and vice versa.

Before we can give a formal definition of compatibility $\mathcal{L}$-bisimulation, we need to define:

- a new transition symbol $\Rightarrow_\mathcal{L}$, which stands for zero or more transitions that do not contain any action in $\mathcal{L}$. We use this to ignore transitions

that are not involved with the vertical syncronisation we want to check. This is analogous to the approach used in weak bisimulation to ignore τ actions [17];

- a notion of transitions *safely blocked*, that cannot synchronise via vertical cooperation and that can be ignored. When two model processes M_1 and M_2 are put in parallel with $\underset{\mathcal{L}}{\bowtie}$, some of the transitions $\xrightarrow{(\mathcal{A}[\mathcal{E}],\Delta)}$ that M_1 or M_2 alone can perform become blocked. There are two reasons why this occurs. The former is that $\mathcal{A} \cap \mathcal{L} \neq \emptyset$ and synchronisation cannot take place because the other process cannot immediately produce the necessary hooks. These are the *safely blocked* transitions. The latter is that $\mathcal{E} \cap \mathcal{L} \neq \emptyset$ and synchronisation cannot take place because the other process cannot "respond" to the hooks present in $\mathcal{E} \cap \mathcal{L}$. If this is the case, then $M_1 \not\asymp_{\mathcal{L}} M_2$, so we exclude this case from the definition of compatibility $\mathcal{L}$-bisimulation.

Formal definitions are given below. Because in this section we are not interested in environments and rates, we use $M_1 \xrightarrow{\mathcal{A}[\mathcal{E}]} M_2$ to imply $\exists \Delta.\ M_1 \xrightarrow{(\mathcal{A}[\mathcal{E}],\Delta)} M_2$.

Definition 24. *Transition* $\Rightarrow_{\mathcal{L}}$. Given $M_1, M_2 \in \mathbb{P}_m$, $M_1 \Rightarrow_{\mathcal{L}} M_2$ iff:

- $M_1 = M_2$ or
- $M_1 \xrightarrow{\mathcal{A}_1[\mathcal{E}_1]} ... \xrightarrow{\mathcal{A}_i[\mathcal{E}_i]} ... \xrightarrow{\mathcal{A}_n[\mathcal{E}_n]} M_2$ and $\forall i = 1, ..., n,\ \mathcal{A}_i \cap \mathcal{L} = \emptyset$ and $\mathcal{E}_i \cap \mathcal{L} = \emptyset$

Definition 25. *Transition safely blocked by a model process via a cooperation set.* Given model processes $M_1, M_2 \in \mathbb{P}_m$ and a cooperation set $\mathcal{L} \subseteq \mathit{Actions}$, a transition $M_1 \xrightarrow{\mathcal{A}[\mathcal{E}]} M_1'$ is *safely blocked* by M_2 via $\mathcal{L}$ iff $\mathcal{A} \cap \mathcal{L} \neq \emptyset$ and:

- $\neg(\exists M_2'.\ M_2 \xrightarrow{\mathcal{B}[\mathcal{F}]} M_2'$ with $\mathcal{A} \subseteq \mathcal{F} \cap \mathcal{L})$ or
- $\exists M_2'.\ M_2 \xrightarrow{\mathcal{B}[\mathcal{F}]} M_2'$ with $\mathcal{A} \subseteq \mathcal{F} \cap \mathcal{L}$ and $\exists M_1''.\ M_1 \xrightarrow{\mathcal{A}'[\mathcal{E}']} M_1''$ with $\mathcal{A}' \subseteq \mathcal{F} \cap \mathcal{L}$ and $|\mathcal{A}'| \geq |\mathcal{A}|$

Definition 26. *Compatibility $\mathcal{L}$-bisimulation.* A relation $\mathcal{R} \subseteq \mathbb{P}_m \times \mathbb{P}_m$ is a compatibility $\mathcal{L}$-bisimulation iff whenever $(M_1, M_2) \in \mathcal{R}$ then for all $M_1 \xrightarrow{\mathcal{A}[\mathcal{E}]} M_1'$ that are not safely blocked by M_2 via $\mathcal{L}$ we have one of the following:

- if $\mathcal{A} \cap \mathcal{L} = \emptyset \wedge \mathcal{E} \cap \mathcal{L} = \emptyset$ then $\exists M_2'.\ M_2 \Rightarrow_{\mathcal{L}} M_2'$ and $(M_1', M_2') \in \mathcal{R}$;
- if $\mathcal{A} \cap \mathcal{L} \neq \emptyset$ then $\exists M_2'.\ M_2 \xrightarrow{\mathcal{B}[\mathcal{F}]} M_2'$ with $\mathcal{A} \subseteq \mathcal{F} \cap \mathcal{L}$) and $\neg(\exists M_1''.\ M_1 \xrightarrow{\mathcal{A}'[\mathcal{E}']} M_1''$ with $\mathcal{A}' \subseteq \mathcal{F}' \cap \mathcal{L}$ and $|\mathcal{A}'| \geq |\mathcal{A}|$) and $(M_1', M_2') \in \mathcal{R}$;
- if $\mathcal{E} \cap \mathcal{L} \neq \emptyset$ then $\exists M_2'.\ M_2 \xrightarrow{\mathcal{B}[\mathcal{F}]} M_2'$ with $\mathcal{B} \subseteq \mathcal{E} \cap \mathcal{L}$ and $\neg(\exists M_2''.\ M_2 \xrightarrow{\mathcal{B}'[\mathcal{F}']} M_2''$ with $|\mathcal{B}'| \geq |\mathcal{B}|$) and $(M_1', M_2') \in \mathcal{R}$.

In addition, the same must be true for all $M_2 \xrightarrow{\mathcal{B}[\mathcal{F}]} M_2'$ that are not safely blocked by M_1 via $\mathcal{L}$.

Model processes M_1 and M_2 are compatibility $\mathcal{L}$-bisimilar, denoted $M_1 \asymp_{\mathcal{L}} M_2$, if $(M_1, M_2) \in \mathcal{R}$ for a compatibility $\mathcal{L}$-bisimulation $\mathcal{R}$.

5.1. Example of Use of Compatibility $\mathcal{L}$-Bisimulation

In this section we propose a simple model of tissue growth. This model is based on a more complex model of tissue growth that we presented in our previous work [1].

At the tissue scale we consider an area divided into regions of the same size and shape. Without loss of generality, we consider only two regions, *R1* and *R2* (Figure 6). Each region can be empty (agents beginning with E) or contain tissue. There are two types of tissue: active and inactive. Active tissue (agents beginning with *Ton*) is mitosis enabled (can perform actions beginning with *mito*), that is its cells can duplicate and produce tissue that will occupy a neighbouring empty region. Action *mito12* represents, for example, growth from region *R1* to region *R2*. If no adjacent region is empty, mitosis is inhibited. Inactive tissue (agents beginning with *Toff*) cannot replicate. Both types of tissue can become empty space through apoptosis (actions beginning with *apo*), that is cell death.

The biochemical scale consists only of a biochemical species **A**, present in the two regions. The concentration of **A** (agents beginning with A) can assume three possible values: low, medium and high (agents with subscript L, M and H). Concentration level can change because of transport between

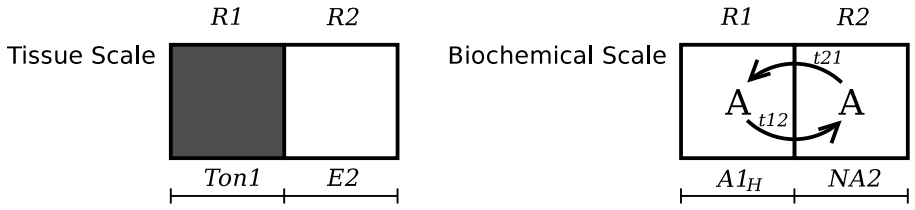


Figure 6: A theoretical multi-scale model of tissue growth.

regions (actions beginning with t, such as *t12* for transport from region *R1* to region *R2*).

The following constraints, which require communication *between* scales, must hold:

- tissue is active in a region if and only if the concentration of **A** in the same region is high. Hook actions beginning with *mitoon* and *mitooff* should ensure this is the case;
- a region is empty if and only if there is no biochemistry. To represent the absence of biochemistry we use processes beginning with *NA*. Hook actions beginning with *bioon* and *biooff* should ensure this is the case.

Agent definitions are given below, with a graphical representation illustrated in Figure 7:

$$
\begin{aligned}
&NA1 \triangleq bioon1.A1_L \\
&A1_L \triangleq biooff1.NA1 + t21.A1_M \\
&A1_M \triangleq biooff1.NA1 \\
&\qquad +t21[mitoon1].A1_H + t12.A1_L \\
&A1_H \triangleq biooff1.NA1 \\
&\qquad +t12[mitooff1].A1_M
\end{aligned}
\qquad
\begin{aligned}
&E1 \triangleq mito21[bioon1].Toff1 \\
&Toff1 \triangleq apo1[biooff1].E1 \\
&\qquad +mitoon1.Ton1 \\
&Ton1 \triangleq apo1[biooff1].E1 \\
&\qquad +mito12.Ton1 \\
&\qquad +mitooff1.Toff1
\end{aligned}
$$

$$
\begin{aligned}
&NA2 \triangleq bioon2.A2_L \\
&A2_L \triangleq biooff2.NA2 + t12.A2_M \\
&A2_M \triangleq biooff2.NA2 \\
&\qquad +t12[mitoon2].A2_H + t21.A2_L \\
&A2_H \triangleq biooff2.NA2 \\
&\qquad +t21[mitooff2].A2_M
\end{aligned}
\qquad
\begin{aligned}
&E2 \triangleq mito12[bioon2].Toff2 \\
&Toff2 \triangleq apo2[biooff2].E2 \\
&\qquad +mitoon2.Ton2 \\
&Ton2 \triangleq apo2[biooff2].E2 \\
&\qquad +mito21.Ton2 \\
&\qquad +mitooff2.Toff2
\end{aligned}
$$

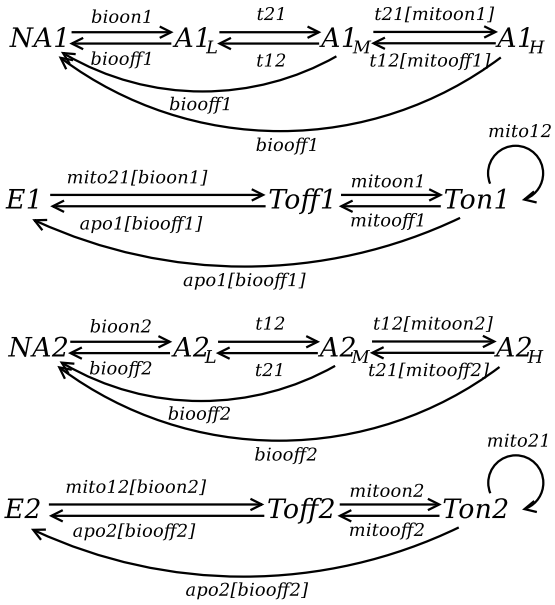


Figure 7: Graphical representation of the processes defined in the tissue growth example.

The initial state of this model is $(A1_H \underset{\mathcal{L}}{\bowtie} NA2) \underset{\mathcal{L}'}{\bowtie} (Ton1 \underset{\mathcal{L}''}{\bowtie} E2)$, where $\mathcal{L} = \{t12, t21\}$, $\mathcal{L}' = \{bioon1,\ bioon2,\ biooff1,\ biooff2,\ mitoon1,\ mitoon2,\ mitooff1,\ mitooff2\}$ and $\mathcal{L}'' = \{mito12,\ mito21\}$.

An example of transition is:

$$(A1_H \underset{\mathcal{L}}{\bowtie} NA2) \underset{\mathcal{L}'}{\bowtie} (Ton1 \underset{\mathcal{L}''}{\bowtie} E2) \xrightarrow{\{mito12,bioon2\}[\emptyset]} (A1_H \underset{\mathcal{L}}{\bowtie} A2_L) \underset{\mathcal{L}'}{\bowtie} (Ton1 \underset{\mathcal{L}''}{\bowtie} Toff2)$$

In the above transition, action *mito12* represents growth of tissue from region *R1* to region *R2*, while vertical synchronisation with action *bioon2* ensures that the biochemical scale in region *R2* is enabled. It can be shown that $A1_H \underset{\mathcal{L}}{\bowtie} NA2 \asymp_{\mathcal{L}'} Ton1 \underset{\mathcal{L}''}{\bowtie} E2$.

An example of a safely blocked transition is:

$$Ton1 \underset{\mathcal{L}''}{\bowtie} E2 \xrightarrow{mitooff1} Toff1 \underset{\mathcal{L}''}{\bowtie} E2$$

The above transition is safely blocked by $A1_H \underset{\mathcal{L}}{\bowtie} NA2$ via $\mathcal{L}'$. This is because in this model the only way to reduce the concentration of **A** is by transport from *R1* to *R2*, while such transport is not possible because *NA2*

represents the absence of biochemistry and cannot produce any transition that can synchronise with $A1_H \xrightarrow{t12[mitooff1]} A1_M$.

Detect incorrect model definitions. We give now an example of a mistake in the model definition that leads to non compatibility $\mathcal{L}$-bisimilar processes. We show that by replacing process definition $E2 \triangleq mito12[bioon2].Toff2$ with $E2 \triangleq mito12.Toff2$, that is "forgetting" to place hook *bioon2*, then $A1_H \underset{\mathcal{L}}{\bowtie} NA2 \not\sim_{\mathcal{L}'} Ton1 \underset{\mathcal{L}''}{\bowtie} E2$.

After the replacement, following the definition of compatibility $\mathcal{L}$-bisimulation, transition

$$Ton1 \underset{\mathcal{L}''}{\bowtie} E2 \xrightarrow{mito12} Ton1 \underset{\mathcal{L}''}{\bowtie} Toff2$$

is matched by

$$A1_H \underset{\mathcal{L}}{\bowtie} NA2 \Rightarrow_{\mathcal{L}'} A1_H \underset{\mathcal{L}}{\bowtie} NA2$$

However, transition

$$Ton1 \underset{\mathcal{L}''}{\bowtie} Toff2 \xrightarrow{apo2[biooff2]} Ton1 \underset{\mathcal{L}''}{\bowtie} E2$$

cannot be matched by any transition from $A1_H \underset{\mathcal{L}}{\bowtie} NA2$. This implies $A1_H \underset{\mathcal{L}}{\bowtie} NA2 \not\sim_{\mathcal{L}} Ton1 \underset{\mathcal{L}''}{\bowtie} Toff2$, which in turn implies $A1_H \underset{\mathcal{L}}{\bowtie} NA2 \not\sim_{\mathcal{L}'} Ton1 \underset{\mathcal{L}''}{\bowtie} E2$.

6. Relating Biological Systems at Specified Scales

In this section we define a relation that will allow us to:

- compare the behaviour of two different process algebra models with respect to a specified scale;
- substitute parts of a model with behaviourally equivalent and less complex ones.

In particular, we define *Markovian $\mathcal{T}$-bisimulation* ($\simeq_{\mathcal{T}}$), which ensures that two processes produce the same rated and filtered activities at the same time and with the same probability, while presenting identical open transitions.

We anticipate that such relation may be too strong for most biological applications. Nonetheless, such a relation is a fundamental relation from

which other relations can be defined. In particular, in biology one is often interested in determining whether two systems are *almost* rather than *exactly* the same, and possibly to which extent they are similar. We will discuss this again in Section 10.1.

In order to relate models at a specified scale, we define a mechanism to focus on a specified scale in PAH. This mechanism is called *filtering* and consists of removing undesired action names from valid rated transitions belonging to a rated derivation graph. The result will be a *filtered* derivation graph. Because actions are associated with functional rates, removing actions interferes with rating. As a consequence, rating of transitions should always precede filtering.

In PAH actions from every scale are collected into a unique action multi-set that labels valid transitions. If we want to focus on a specific scale, all we need to do is to keep only the actions pertaining to a given scale. For example, consider the following rated transition:

$$M \xrightarrow{(\{\!|a,h,x|\!\}[\{\!|y,z|\!\}],r)}_{\Gamma} M'$$

From this transition it is possible to infer that actions a, h and x have been performed, that hook actions y and z have not been used in any synchronisation and that the rate of the transition is r. Now assume that we are interested in only the behaviour represented by the actions in $\mathcal{T}$, with $\mathcal{T} = \{\!|x, y|\!\}$. The corresponding filtered transition should be:

$$M \xrightarrow{(\{\!|x|\!\},\{\!|y,z|\!\},r)}_{\mathcal{T},\Gamma} M'$$

Notice that the rate and the multi-set of hook actions is untouched. Moreover, a *filtered* activity is a triple, to distinguish it from a *rated* activity, which is a pair. We introduce now the formal definitions of filtered activities, filtered transitions and filtered derivation graph.

Definition 27. *Filtered activities.* The triple $(\mathcal{A}, \mathcal{E}, r)$ such that $\mathcal{A}$ and $\mathcal{E}$ are multi-sets of actions and $r \in \mathbb{R}_{>0}$ is called a filtered activity.

Definition 28. *Current filtered activities of a model process.* Given a multi-set of actions $\mathcal{T}$ and an environment $\Gamma \subseteq Names \times \mathbb{R}$, the multi-set of filtered activities that a model process $M \in \mathbb{P}_m$ can perform is denoted by $FiltAct(M)_{\mathcal{T},\Gamma}$ and is defined as:

$$FiltAct(M)_{\mathcal{T},\Gamma} = \{\!|(\mathcal{B}, \mathcal{E}, r) \mid (\mathcal{A}[\mathcal{E}], r) \in RatedAct(M)_{\Gamma} \wedge \mathcal{B} = \mathcal{A} \cap \mathcal{T}|\!\}$$

Definition 29. *Filtered activity set.* Given a multi-set of actions $\mathcal{T}$ and an environment $\Gamma \subseteq Names \times \mathbb{R}$, the multi-set of filtered activities that a model process $M \in \mathbb{P}_m$ and its derivatives can perform is denoted by $\overrightarrow{FiltAct}(M)_{\mathcal{T},\Gamma}$ and is defined as:

$$\overrightarrow{FiltAct}(M)_{\mathcal{T},\Gamma} = \biguplus_{M_i \in ds(M)} FiltAct(M_i)_{\mathcal{T},\Gamma}$$

Definition 30. *Filtered moves of a definition process.* Given a multi-set of actions $\mathcal{T}$, the multi-set of moves that a definition process $D \in \mathbb{P}_d$ can perform is denoted by $FiltMovesD_{\mathcal{T}}$ and is defined as:

$$FiltMoves(D)_{\mathcal{T}} = \{\!|((\mathcal{B},\mathcal{E}),A) \mid (\mathcal{A}[\mathcal{E}],A) \in Moves(D) \wedge \mathcal{B} = \mathcal{A} \cap \mathcal{T}|\!\}$$

Definition 31. *Filtered moves of a model process.* Given a multi-set of actions $\mathcal{T}$ and an environment $\Gamma \subseteq Names \times \mathbb{R}$, the multi-set of moves that a model process $M \in \mathbb{P}_m$ can perform is denoted by $FiltMovesM_{\mathcal{T},\Gamma}$ and is defined as:

$$FiltMoves(M)_{\mathcal{T},\Gamma} = \left\{\!\!\left| ((\mathcal{B},\mathcal{E},r),A) \;\middle|\; \begin{array}{l} ((\mathcal{A}[\mathcal{E}],r),A) \in RatedMoves(M)_\Gamma \\ \wedge \mathcal{B} = \mathcal{A} \cap \mathcal{T} \end{array} \right|\!\!\right\}$$

Definition 32. *Filtered derivation graph.* Given a model process $M \in \mathbb{P}_m$, a multi-set of actions $\mathcal{T}$ and an environment $\Gamma \subseteq Names \times \mathbb{R}$, the filtered derivation graph $\mathcal{D}_f(M)_{\mathcal{T},\Gamma}$ is the labelled directed graph with:

- set of nodes $ds(M)$;
- multi-set of transition labels $\overrightarrow{FiltAct}(M)_{\mathcal{T},\Gamma}$;
- multi-set of labelled transitions $\rightarrow_{\mathcal{T},\Gamma} \subseteq ds(M) \times \overrightarrow{FiltAct}(M)_{\mathcal{T},\Gamma} \times ds(M)$. Given $M' \in ds(M)$, $(M', \mathcal{A}, \mathcal{E}, r_a, M'') \in \rightarrow_{\mathcal{T},\Gamma}$ with the same multiplicity as $((\mathcal{A},\mathcal{E},r_a),M'')$ in $FiltMoves(M')_{\mathcal{T},\Gamma}$.
- multi-set of labelled transitions $\rightarrow_o \subseteq ds(M) \times \overrightarrow{OpenAct}(M) \times ds(M)$. Given $M' \in ds(M)$, $(M', \mathcal{A}[\mathcal{E}], \Gamma', M'') \in \rightarrow_o$ with the same multiplicity as $((\mathcal{A}[\mathcal{E}],\Gamma'),M'')$ in $OpenMoves(M')$.

$\mathcal{D}_f(M)_{\mathcal{T},\Gamma}$ can be written $\mathcal{D}_f(M)$ if $\mathcal{T}$ and Γ are clear from the context.

With the notion of filtered derivation graph we can now define relations between PAH processes based on filtering and relate PAH processes at specified scales.

6.1. Markovian $(\mathcal{T}, \Gamma)$-bisimulation

In this section we define formally Markovian $(\mathcal{T}, \Gamma)$-bisimulation ($\simeq_{\mathcal{T},\Gamma}$) on PAH processes. The definition is based on strong equivalence in PEPA [29] and integrated equivalence in EMPA [32]. Two PAH processes are considered Markovian $(\mathcal{T}, \Gamma)$-bisimilar if it is possible to group the states of their filtered derivation graphs into equivalence classes in such a way that states belonging to the same equivalence class are characterised as follows:

- the sum of the rates of rated moves presenting the same action sets from a state in an equivalence class toward the states of another equivalence class is the same for all states in the same equivalence class;
- the set of open moves from a state in an equivalence class toward the states of another equivalence class is the same for all states in the same equivalence class.

In addition we prove that $\simeq_{\mathcal{T},\Gamma}$ is an equivalence relation (Proposition 37) and a congruence (Proposition 40).

Definition 33. *Functions $\mu_{\mathcal{T},\Gamma}$ and $\nu_{\mathcal{T},\Gamma}$.* Function $\mu_{\mathcal{T},\Gamma}$ returns the rate r at which a model process M can become M' with filtered transitions labelled with $(\mathcal{A}, \mathcal{E})$. Function $\nu_{\mathcal{T},\Gamma}$ returns instead the rate at which M can move to a set of model processes $\mathcal{C}$ with filtered transitions labelled with $(\mathcal{A}, \mathcal{E})$.

$$\mu_{\mathcal{T},\Gamma}(M, \mathcal{A}, \mathcal{E}, M') = \sum_{r_i \in I} r_i$$

where $I = \{\!|r \mid ((\mathcal{A}, \mathcal{E}, r), M') \in FiltMoves(M)_{\mathcal{T},\Gamma}|\!\}$. For each rate r, the same multiplicity as $((\mathcal{A}, \mathcal{E}, r), M')$ in $FiltMoves(M)_{\mathcal{T},\Gamma}$ is used.

$$\nu_{\mathcal{T},\Gamma}(M, \mathcal{A}, \mathcal{E}, \mathcal{C}) = \sum_{M' \in \mathcal{C}} \mu_{\mathcal{T},\Gamma}(M, \mathcal{A}, \mathcal{E}, M')$$

Definition 34. *Open Activities toward a set of model processes.* The multi-set of activities toward a set of model processes $\mathcal{C} \subseteq \mathbb{P}_m$ of a model process $M \in \mathbb{P}_m$ is defined as:

$$OpenAct(M, \mathcal{C}) = \{\!|(\mathcal{A}[\mathcal{E}], \Delta) \mid ((\mathcal{A}[\mathcal{E}], \Delta), M') \in OpenMoves(M) \wedge M' \in \mathcal{C}|\!\}$$

Definition 35. *Model process Markovian* $(\mathcal{T}, \Gamma)$*-bisimulation.* Given an action multi-set $\mathcal{T}$ and an environment $\Gamma \subseteq Names \times \mathbb{R}$, an equivalence relation over model processes $\mathcal{R} \subseteq \mathbb{P}_m \times \mathbb{P}_m$ is a model process Markovian $(\mathcal{T}, \Gamma)$-bisimulation iff whenever $(M_1, M_2) \in \mathcal{R}$ then $\forall$ action multi-sets $\mathcal{A}$ and $\mathcal{E}$ and $\forall \mathcal{C} \in \mathbb{P}_m/\mathcal{R}$

$$\nu_{\mathcal{T},\Gamma}(M_1, \mathcal{A}, \mathcal{E}, \mathcal{C}) = \nu_{\mathcal{T},\Gamma}(M_2, \mathcal{A}, \mathcal{E}, \mathcal{C})$$

and

$$OpenAct(M_1, \mathcal{C}) = OpenAct(M_2, \mathcal{C})$$

Definition 36. *Model process Markovian* $(\mathcal{T}, \Gamma)$*-bisimilarity.* Model process Markovian $(\mathcal{T}, \Gamma)$-bisimilarity, denoted $\simeq_{\mathcal{T},\Gamma}$, is the union of all model process Markovian $(\mathcal{T}, \Gamma)$-bisimulations, i.e.

$$\simeq_{\mathcal{T},\Gamma} = \bigcup \{\mathcal{R} \mid \mathcal{R} \text{ is a model process Markovian } (\mathcal{T}, \Gamma)\text{-bisim.}\}$$

Two model processes $M_1, M_2 \in \mathbb{P}_m$ are Markovian $(\mathcal{T}, \Gamma)$-bisimilar, denoted $M_1 \simeq_{\mathcal{T},\Gamma} M_2$, iff there is a model process $(\mathcal{T}, \Gamma)$-bisimulation $\mathcal{R}$ between them such that $(M_1, M_2) \in \mathcal{R}$.

If Γ is clear from the context, we write $M_1 \simeq_{\mathcal{T}} M_2$ instead of $M_1 \simeq_{\mathcal{T},\Gamma} M_2$ and we say M_1 and M_2 are Markovian $\mathcal{T}$-bisimilar.

Proposition 37. Model process Markovian $(\mathcal{T}, \Gamma)$-bisimilarity $(\simeq_{\mathcal{T},\Gamma})$ is an equivalence relation and it is the largest model process Markovian $(\mathcal{T}, \Gamma)$-bisimulation.

Proof. A model process Markovian $(\mathcal{T}, \Gamma)$-bisimilarity is an equivalence relation iff it is *symmetric*, *reflexive* and *transitive*. The first two properties are trivially true. We prove *transitivity* and at the same time we conclude that $\simeq_{\mathcal{T},\Gamma}$ is the largest model process Markovian $(\mathcal{T}, \Gamma)$-bisimulation. We follow the same strategy as in [29].

We prove that given any number of bisimulations $\mathcal{R}_i$, one can always find a bisimulation that contains them all, which is given by the transitive closure of their union $\mathcal{R} = (\bigcup_i \mathcal{R}_i)^*$. Assume $\mathcal{R}_i$ is a model process Markovian $(\mathcal{T}, \Gamma)$-bisimulation for all $i \in I$. By definition, each $\mathcal{R}_i$ is an equivalence relation that partitions $\mathbb{P}_m$ into equivalence classes. By definition of transitive closure, any equivalence class $S^i_j \in \mathbb{P}_m/\mathcal{R}_i$ is included in some $T_k \in \mathbb{P}_m/\mathcal{R}$, with $T_k = \bigcup_j S^i_j$ and $j \in J^i_k$. The set J^i_k indicates how the equivalence classes

induced by $\mathcal{R}_i$ are merged into the equivalence class T_k. A bisimulation $\mathcal{R}_{i'}$ might partition T_k in a different way, but because of the transitive closure $T_k = \bigcup_j S^i_j = \bigcup_{j'} S^{i'}_{j'}$, $j' \in J^{i'}_k$. We now prove by induction on the definition of transitive closure that $\mathcal{R}$ is a model process Markovian $(\mathcal{T}, \Gamma)$-bisimulation.

Base: we define $\mathcal{R}^n = (\bigcup_i \mathcal{R}_i)^n$, that is the n-th step in the expansion of the transitive closure. Consider $\mathcal{R}^0 = \bigcup_i \mathcal{R}_i$. If $(M_1, M_2) \in \mathcal{R}^0$ then $(M_1, M_2) \in \mathcal{R}_i$ for some $i \in I$. This implies that for all action multi-sets $\mathcal{A}$ and $\mathcal{E}$ and $\forall T_k \in \mathbb{P}_m/\mathcal{R}$:

$$\begin{aligned}\nu_{\mathcal{T},\Gamma}(M_1, \mathcal{A}, \mathcal{E}, T_k) &= \nu_{\mathcal{T},\Gamma}(M_1, \mathcal{A}, \mathcal{E}, \bigcup_{j \in J^i_k} S^i_j) = \sum_{j \in J^i_k} \nu_{\mathcal{T},\Gamma}(M_1, \mathcal{A}, \mathcal{E}, S^i_j) \\ &= \sum_{j \in J^i_k} \nu_{\mathcal{T},\Gamma}(M_2, \mathcal{A}, \mathcal{E}, S^i_j) = \nu_{\mathcal{T},\Gamma}(M_2, \mathcal{A}, \mathcal{E}, T_k)\end{aligned}$$

and

$$\begin{aligned}OpenAct(M_1, T_k) &= OpenAct(M_1, \bigcup_{j \in J^i_k} S^i_j) = \biguplus_{j \in J^i_k} OpenAct(M_1, S^i_j) \\ &= \biguplus_{j \in J^i_k} OpenAct(M_2, S^i_j) = OpenAct(M_2, T_k)\end{aligned}$$

Induction: Given any $(M_1, M_2) \in \mathcal{R}^{n-1}$ we assume that $\nu_{\mathcal{T},\Gamma}(M_1, \mathcal{A}, \mathcal{E}, T_k) = \nu_{\mathcal{T},\Gamma}(M_2, \mathcal{A}, \mathcal{E}, T_k)$ and $OpenAct(M_1, T_k) = OpenAct(M_2, T_k)$. We then prove that this is the case also for any $(M'_1, M'_2) \in \mathcal{R}^n$. If $(M'_1, M'_2) \in \mathcal{R}^n$ then, by definition of transitive closure, either $(M'_1, M'_2) \in \mathcal{R}^{n-1}$ or $(M'_1, M'_2) \in \{(P, Q) \mid \exists R.\, (P, R) \in \mathcal{R}^{n-1} \wedge (R, Q) \in \mathcal{R}^{n-1}\}$. This implies $\exists R$ such that:

$$\nu_{\mathcal{T},\Gamma}(M'_1, \mathcal{A}, \mathcal{E}, T_k) = \nu_{\mathcal{T},\Gamma}(R, \mathcal{A}, \mathcal{E}, T_k) = \nu_{\mathcal{T},\Gamma}(M'_2, \mathcal{A}, \mathcal{E}, T_k)$$

and

$$OpenAct(M'_1, T_k) = OpenAct(R, T_k) = OpenAct(M'_2, T_k)$$

This proves that given any number of model process Markovian $(\mathcal{T}, \Gamma)$-bisimulations, we can always find a bisimulation that contains them. Because $\simeq_{\mathcal{T},\Gamma}$ includes all possible bisimulations, then $\simeq_{\mathcal{T},\Gamma}$ is at least as large as the largest model process Markovian $(\mathcal{T}, \Gamma)$-bisimulation.

Definition 38. *Filtered composed actions toward a set of model processes.* Given a multi-set of actions $\mathcal{T}$, the multi-set of composed actions toward a set of model processes $\mathcal{C} \subseteq \mathbb{P}_m$ of a definition process $D \in \mathbb{P}_d$ is defined as:

$$FiltCompAct(D, \mathcal{C})_{\mathcal{T}} = \{\!|(\mathcal{A}, \mathcal{E}) \mid ((\mathcal{A}, \mathcal{E}), D') \in FiltMoves(D)_{\mathcal{T}} \wedge D' \in \mathcal{C}|\!\}$$

Definition 39. *Markovian $(\mathcal{T},\Gamma)$-bisimilar definition processes.* Given an action multi-set $\mathcal{T}$ and an environment $\Gamma \subseteq Names \times \mathbb{R}$, two definition processes $D_1, D_2 \in \mathbb{P}_d$ are definition process Markovian $(\mathcal{T},\Gamma)$-bisimilar ($D_1 \simeq_{\mathcal{T},\Gamma} D_2$) iff $\forall \mathcal{C} \in \mathbb{P}_m/\simeq_{\mathcal{T},\Gamma}$

$$FiltCompAct(D_1,\mathcal{C})_{\mathcal{T}} = FiltCompAct(D_2,\mathcal{C})_{\mathcal{T}}$$

Proposition 40. Markovian $(\mathcal{T},\Gamma)$-bisimulation as a Congruence. *If $P_1, P_2 \in \mathbb{P}$ such that $P_1 \simeq_{\mathcal{T},\Gamma} P_2$, then*

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

Proof. See Appendix B.

7. Parametric Stochastic Process Algebra with Hooks

In analogy with our previous work [1], in this section we introduce the syntax of a parametric version of PAH. This version is equivalent to PAH as introduced in the previous sections while it makes models easier to write and comprehend because we can, in fact, index processes.

In general, we expect the modeller to develop his own scripts, for example in a *BASH* environment, to produce automatically models that include hundreds of processes. The definition of these scripts goes beyond the scope of this paper and, in any case, the scripts would be tailored around the specific problem modelled. For example, a script for a model of tissue growth would produce automatically a grid of processes representing discrete portions of tissue. To check that the produced model is as intended, one can define relations such as our compatibility $\mathcal{L}$-bisimulation (Section 5) and implement algorithms to check whether these relations hold.

Here we limit ourselves to augmenting PAH as defined in Section 4 with parametrised processes and actions and we obtain a parametric stochastic process algebra with hooks (psPAH). The syntax of psPAH is as follows:

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

The main differences between this and the non parametric version are:

- $k \in \mathbb{R}$ and i is a parameter name, i.e. $i \in Names$;
- actions have the form $a(exp, \ldots, exp)$, where $exp, \ldots, exp$ is a list of expressions. In particular, multi-sets $\mathcal{L} = (\mathcal{L}', m_{\mathcal{L}})$, $\mathcal{A} = (\mathcal{A}', m_{\mathcal{A}})$ and $\mathcal{E} = (\mathcal{E}', m_{\mathcal{E}})$, where $\mathcal{A}', \mathcal{E}' \subseteq Actions$ and $\mathcal{L}' \subseteq ConstActions$. $Actions = \{a(exp_{a1}, \ldots, exp_{an}), b(exp_{b1}, \ldots, exp_{bn}), \ldots\}$ and $ConstActions \subset Actions$, $ConstActions = \{a(k_{a1}, \ldots, k_{an}), b(k_{b1}, \ldots, k_{bn}), \ldots\}$;
- 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 definitions of functional rates and the variables associated with agents are also parametric.

The semantics is given in Figure 8. Given an environment Δ, the evaluation of an expression exp into a real number k is denoted by $\Delta \vdash exp \to k$, the evaluation of a boolean expression $bexp$ into $b \in \{true, false\}$ is denoted by $\Delta \vdash bexp \to b$, while the evaluation of the list of expressions of all the actions in a set $\mathcal{A}$ is denoted by $\Delta \vdash \mathcal{A} \to \mathcal{A}'$, where $\mathcal{A}'$ contains only actions with evaluated expressions.

An interpreter for psPAH has been implemented in the functional programming language OCaml. The interpreter reads as input the description of a psPAH model along with a model time threshold for the simulations. Simulations are performed on a model, producing traces of states and time delays using a standard sampling method for continuous time Markov chains.

Simulations have been performed on a laptop computer with Ubuntu Linux, two Intel Core 2 Duo 2.20 GHz CPUs and 2 GB of RAM.

The simulations given in the next section, were all computed in within at most 48 hours.

Prefix

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

Choice Left

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

Choice Right

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

If Then Else True

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

If Then Else False

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

Agent

$$\frac{D \xrightarrow{\mathcal{A}[\mathcal{E}],b} A'(exp_1,\ldots,exp_n)}{A(k_1,\ldots,k_n) \xrightarrow{(\mathcal{A}'[\mathcal{E}'],\Delta')} A'(k'_1,\ldots,k'_n)}\ *$$

* if $A(i_1,\ldots,i_n) \triangleq D \wedge \Delta = \{(i_1,k_1),\ldots,(i_n,k_n)\} \wedge \Delta \vdash b \to true$
$\Delta \vdash exp_1 \to k'_1 \wedge \cdots \wedge \Delta \vdash exp_n \to k'_n \wedge \Delta \vdash \mathcal{A} \to \mathcal{A}' \wedge \Delta \vdash \mathcal{E} \to \mathcal{E}'$
$\wedge\ \Delta' = \{(Var(A(k_1,\ldots,k_n)), Val(A(k_1,\ldots,k_n)))\}$

Figure 8: Semantics of parametric stochastic process algebra with hooks. Other inference rules are as in Figure 5.

8. Multi-Scale Model of Pattern Formation

We now give an example specification and analysis of a well-known problem from Systems Biology: the French Flag problem [33]. In this well-known system, a group of identical cells in a tissue are differentiated into subgroups, according to the diffusion of a morphogen.

A morphogen **M** diffuses from a source into a tissue. In the long run, the region close to the source presents a high concentration of **M**, while the further a region is from the source, the lower the concentration of **M** is in that region. The concentration of **M** in a region indicates the positional information, i.e. the position with respect to the source. Different specialisa-

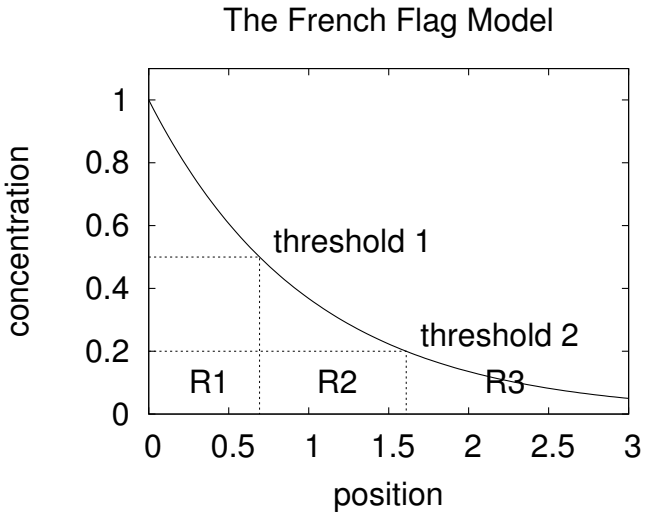


Figure 9: The French Flag Model implemented with partial differential equations. In the picture, two concentration thresholds divide the space into three regions.

tions are assigned to a region depending on the concentration of $\mathbf{M}$ in that region. Of great importance are concentration thresholds that delimit the concentration ranges associated with different specialisations.

One of the simplest models of this scenario is the following partial differential equation (PDE) [34]:

$$\frac{\partial \mathbf{M}(t,x)}{\partial t} = D\frac{\partial^2 \mathbf{M}(t,x)}{\partial x^2} - \alpha \mathbf{M}(t,x)$$

The above equation models the concentration of $\mathbf{M}$ in time and space in one-dimensional coordinates. The element $D\frac{\partial^2 \mathbf{M}(t,x)}{\partial x^2}$ is the diffusion of $\mathbf{M}$, while $\alpha\mathbf{M}(t,x)$ is its degradation. Constant D is the diffusion constant and α is the degradation constant; we assume these two constants to be equal to 1. Boundary conditions are:

1. $\mathbf{M}(t,0) = 1$ with $0 \leq t < \infty$, i.e. a constant source of $\mathbf{M}$ at position 0;
2. $\mathbf{M}(t,\infty) = 0$ with $0 \leq t < \infty$, i.e. concentration of $\mathbf{M}$ is lost in the surroundings.

Steady state solution of the PDE model is shown in Figure 9. In the figure, the steady state solution is shown (continuous line). Two concentration

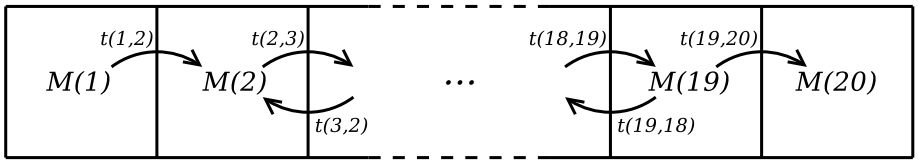


Figure 10: Discretisation of the space of the French Flag Model into 20 regions. The variable $M(i)$ indicates the concentration of $\mathbf{M}$ in region i.

thresholds (at 0.2 and 0.5, dotted lines) divide the area in three regions (R1, R2 and R3). Each of these regions are characterised by a different specialisation.

We propose now a representation of the PDE model in psPAH. Because the PDE model is continuous in the concentration of $\mathbf{M}$ and in space, while psPAH uses a discrete representation of these quantities, we provide a discretisation. We assume 20 levels for the concentration of $\mathbf{M}$ (parameter $maxLevels = 20$), with maximum concentration equal to 1 and concentration of each level equal to $h = 1/20 = 0.05$, and we assume 20 regions of space (parameter $regions = 20$), with total length equal to 3 and length of each region equal to $deltaX = 3/20 = 0,15$. The spatial discretisation is illustrated in Figure 10.

In order to represent the two boundary conditions we have:

- the left most region presents a constant concentration level of 20, which guarantees that concentration flows continuously from the left;
- the right most region presents a constant concentration level equal to 0, which implies that this region absorbs concentration levels.

We define agent $M(i, w)$ (Figure 11) to indicate that morphogen $\mathbf{M}$ in region i presents concentration level w. Actions $t(i, j)$ represent transport of $\mathbf{M}$ from region i to region j, while actions $deg(i)$ represent degradation of $\mathbf{M}$ in region i.

We model specialisation of regions explicitly as follows. Two thresholds are considered: one between 4 and 5 concentration levels and the other between 10 and 11. Whenever a concentration threshold is crossed in a region, the specialisation of the region changes. The presence of two thresholds implies that three specialisations are possible, corresponding to high, medium and low concentration ranges of $\mathbf{M}$. In addition, we consider that regions of

$M(i,w) \triangleq$
if $i == 1$ **then** //first region, the source of M
  $t(1,2).M(1,w)$
else
  if $i == regions$ **then** //last region, absorbing
    $t(regions-1, regions).M(regions, w)$
  else //any other region
    if $w > 0$ **then** //degradation of M
      if $w == (thr1+1) \vee w == (thr2+1)$ **then**
        $deg(i)[y(i)].M(i, w-1)$
      else
        $deg(i).M(i, w-1)$
    else
      nil
    $+$ //transport of M to next region
    if $i < regions \wedge w > 0$ **then**
      if $w == (thr1+1) \vee w == (thr2+1)$ **then**
        $t(i, i+1)[y(i)].M(i, w-1)$
      else
        $t(i, i+1).M(i, w-1)$
    else
      nil
    $+$ //transport of M from next region
    if $i < (regions-1) \wedge w < maxLevels$ **then**
      if $w == (thr1) \vee w == (thr2)$ **then**
        $t(i+1, i)[x(i)].M(i, w+1)$
      else
        $t(i+1, i).M(i, w+1)$
    else
      nil
    $+$ //transport of M to previous region
    if $i > 2 \wedge w > 0$ **then**
      if $w == (thr1+1) \vee w == (thr2+1)$ **then**
        $t(i, i-1)[y(i)].M(i, w-1)$
      else
        $t(i, i-1).M(i, w-1)$
    else
      nil
    $+$ //transport of M from previous region
    if $i > 1 \wedge w < maxLevels$ **then**
      if $w == (thr1) \vee w == (thr2)$ **then**
        $t(i-1, i)[x(i)].M(i, w+1)$
      else
        $t(i-1, i).M(i, w+1)$
    else
      nil

$T(i,z,w) \triangleq$
if $w < 2$ **then**
  $x(i).T(i, 0, w+1)$
else
  nil
$+$
if $w > 0$ **then**
  $(y(i).T(i, 0, w-1)+$
  if $z < 1$ **then**
    $mem(i).T(i, z+1, w)$
  else
    if $w == 2$ **then**
      $mem(i).TA(i)$
    else
      $mem(i).TB(i)$
else
  nil

$TA(i) \triangleq nil$

$TB(i) \triangleq nil$

Figure 11: Agent definitions of processes $M(i,w)$, $T(i,z,w)$, $TA(i)$ and $TB(i)$, from the multi-scale model of pattern formation.

tissue can commit permanently to a specialisation, if the concentration of **M** in those regions stays at a certain concentration range long enough. A commitment means that the cells in the committed region have memorised their positional information and further changes to the concentration of **M** will not affect the chosen specialisation. We use agent $T(i, z, w)$ (Figure 11) to represent tissue region i with specialisation w, while parameter z is part of the implementation of the commitment procedure. Parameter w is equal to 2 when the concentration of **M** is high, 1 when it is medium and 0 when it is low. Inter-scale hook actions $x(i)$ and $y(i)$ are used to synchronise processes $M(i, w)$ and $T(i, z, w)$. To obtain a commitment of a region we define a two-step memorisation:

1. when w is 1 or 2, $T(i, 0, w)$ can perform action $mem(i)$ and become $T(i, 1, w)$;
2. if $T(i, 1, w)$ performs $mem(i)$ then it becomes $TA(i)$ or $TB(i)$, depending on whether w is 2 or 1 respectively. Agents $TA(i)$ and $TB(i)$ are terminated processes that represent the commitment of tissue region i to specialisations A and B, respectively.

In addition, if $T(i, 1, w)$ changes specialisation from w to w' then the process becomes $T(i, 0, w')$. This implies that the attempt at memorising specialisation w is forgotten and a new attempt at memorising specialisation w' can begin. The agent definitions of the French Flag Model are shown in Figure 11.

The initial state of the model is given by the following model component:

$$\begin{array}{c} (TA(1) \underset{\emptyset}{\bowtie} T(2,0,0) \cdots \underset{\emptyset}{\bowtie} T(19,0,0) \underset{\emptyset}{\bowtie} T(20,0,0)) \\ \Big\Vert \\ (M(1,20) \underset{\{\!|t(1,2)|\!\}}{\bowtie} M(2,0) \cdots \underset{\{\!|t(18,19),t(19,18)|\!\}}{\bowtie} M(19,0) \underset{\{\!|t(19,20)|\!\}}{\bowtie} M(20,0)) \end{array}$$

Functional rates for actions $t(i, j)$ and $deg(i)$ are obtained from approximation of the diffusion and degradation elements in the PDE. In addition, we define the functional rate for action $mem(i)$ as the constant value 10. In particular:

$$\begin{aligned} f_{t(i,j)} &= D * M(i) * h/(deltaX * deltaX * h) \\ f_{deg(i)} &= alpha * M(i) * h/h \\ f_{mem(i)} &= 10 \end{aligned}$$

Table 1: In this table we illustrate the commitments of the 20 regions of the French Flag Model over 100 simulations and at different time points. For each region, counts over the simulations of commitments (A, B or none, i.e. not committed) are given.

Time	Comm.	1	2	3	4	5	6	7	8	9	10
0s	A	100	0	0	0	0	0	0	0	0	0
	B	0	0	0	0	0	0	0	0	0	0
	none	0	100	100	100	100	100	100	100	100	100
1.5s	A	100	100	99	98	93	81	66	35	17	8
	B	0	0	1	2	7	15	17	30	35	51
	none	0	0	0	0	0	4	17	35	48	41
3s	A	100	100	99	98	93	85	81	62	42	12
	B	0	0	1	2	7	15	17	33	42	72
	none	0	0	0	0	0	0	2	5	16	16
4.5s	A	100	100	99	98	93	85	83	65	47	17
	B	0	0	1	2	7	15	17	33	46	78
	none	0	0	0	0	0	0	0	2	7	5
6s	A	100	100	99	98	93	85	83	66	53	18
	B	0	0	1	2	7	15	17	33	46	79
	none	0	0	0	0	0	0	0	1	1	3

Time	Comm.	11	12	13	14	15	16	17	18	19	20
0s	A	0	0	0	0	0	0	0	0	0	0
	B	0	0	0	0	0	0	0	0	0	0
	none	100	100	100	100	100	100	100	100	100	100
1.5s	A	6	1	0	0	0	0	0	0	0	0
	B	34	38	28	16	10	5	0	0	0	0
	none	60	61	72	84	90	95	100	100	100	100
3s	A	12	3	0	0	0	0	0	0	0	0
	B	73	76	69	48	28	13	2	2	0	0
	none	15	21	31	52	72	87	98	98	100	100
4.5s	A	13	4	0	0	0	0	0	0	0	0
	B	82	88	84	67	48	23	8	2	0	0
	none	5	8	16	33	52	77	92	98	100	100
6s	A	14	4	0	0	0	0	0	0	0	0
	B	83	95	96	79	61	31	12	3	0	0
	none	3	1	4	21	39	69	88	97	100	100

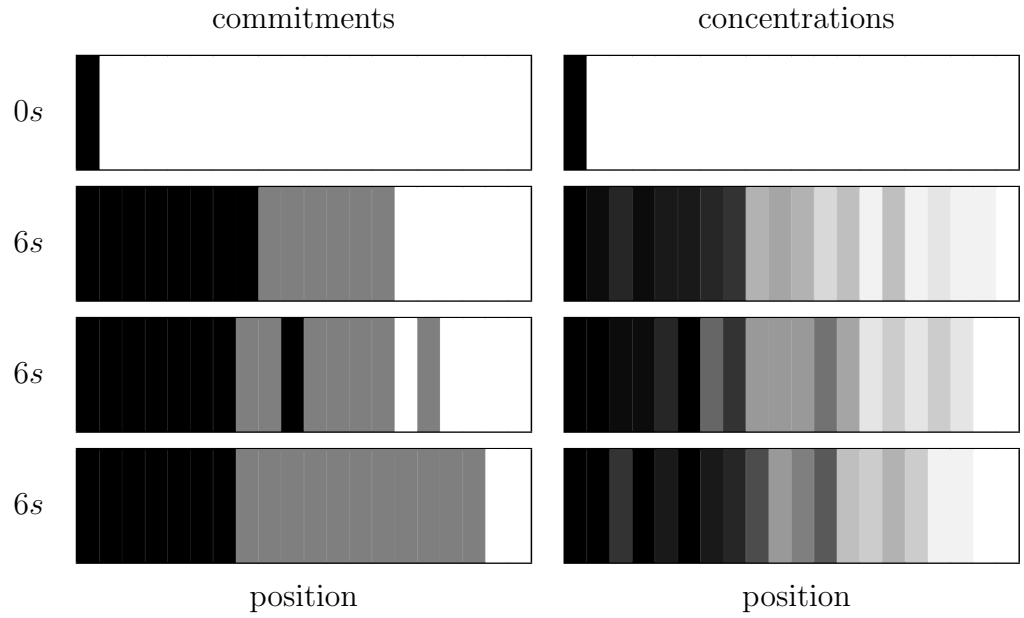


Figure 12: Example of simulations of the process algebra with hooks French Flag Model. On the left: commitments of regions to cell specialisations after 6 seconds. On the right: concentration levels after 6 seconds of the same simulation runs. Top row is the initial condition.

8.1. Analysis

100 simulations were performed on the model, recording the concentration level of **M** in the three regions, up to 6 seconds. Commitments of regions was also recorded. Figure 12 illustrates the initial condition and the typical results from single simulations at time 6 seconds. Although some variability between the runs is visible, a pattern of three distinct commitments is always visible. The images in the figure are constructed from the model component representing the current state at time 6 seconds. Each picture represents a one-dimensional space divided into 20 regions with source of morphogen **M** in the left most region. On the left, commitments to cell differentiation are shown: regions committed to A are represented by the colour black, regions committed to B by grey, while non-committed regions by white. On the right, the corresponding concentration levels are shown: each concentration level is represented by a different shade of grey, from black (maximum concentration) to white (absence of concentration). The top row shows the initial condition

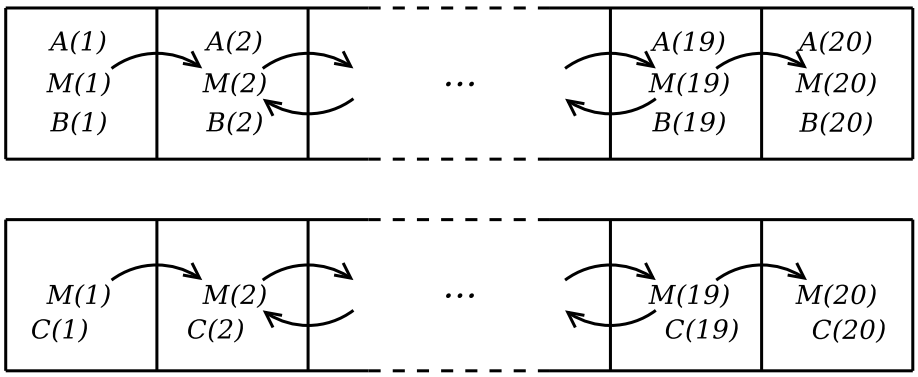


Figure 13: Two extensions of the psPAH French Flag Model. In the first extension, top, two species **A** and **B** are added. In the second extension, bottom, species **C** is added.

(time $0s$), while the other three rows show three different simulations at time $6s$.

Additional data is illustrated in Table 1. In this table commitments of regions are shown. At time 0, only region 1, the source, is committed to specialisation A, while all the other regions are not committed. As the time approaches 6 seconds, we can see the proportion of the 100 simulations in which the regions commit to a certain specialisation. For example, regions 2 to 5 present a clear preference for specialisation A, while regions 12 and 13 have a marked preference for specialisation B. Although some variability is present, a change in preference from left to right is evident.

8.2. Example of Use of Congruence

We illustrate now how the concepts of compositionality and congruence in process algebra can be used to reason about the behaviour of the French Flag Model. In particular, we prove two different extensions of the French Flag model to be Markovian $(\mathcal{T}, \Gamma)$-bisimilar by extending the equality of part of the system to the equality of the whole system. The two extensions consist of the addition of biochemical species **A** and **B** in the first case and **C** in the second. These new species do not interact with morphogen **M** and do not diffuse, but nevertheless produce their own behaviour becoming part of the system. The two extensions are illustrated in Figure 13.

Consider the following two molecular models:

1. the concentration of two molecules **A** and **B** are modelled by agent

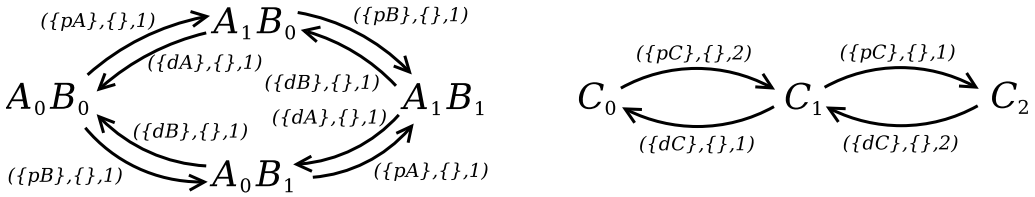


Figure 14: Rated derivation graphs of model processes $A_0(n) \underset{\emptyset}{\bowtie} B_0(n)$ and $C_0(n)$, with $n \in \mathbb{R}$. Parameter n is omitted.

processes representing their concentration as either high or low. Species **A** and **B** are produced or degraded independently according to the mass action kinetic law. Process definitions are as follows:

$$A_0(i) \triangleq pA(i).A_1(i) \quad A_1(i) \triangleq dA(i).A_0(i)$$
$$B_0(i) \triangleq pB(i).B_1(i) \quad B_1(i) \triangleq dB(i).B_0(i)$$

Reactions, functional rates and set of participants are as follows (parameters $k = 1$ and $h = 1$):

$$\begin{array}{lll} R_{pA} : \rightarrow \mathbf{A} & f_{pA(i)} = k/h & p_{pA(i)} = \{A(i)\} \\ R_{dA} : \mathbf{A} \rightarrow & f_{dA(i)} = k * A(i) * h/h & p_{dA(i)} = \{A(i)\} \\ R_{pB} : \rightarrow \mathbf{B} & f_{pB(i)} = k/h & p_{pB(i)} = \{B(i)\} \\ R_{dB} : \mathbf{B} \rightarrow & f_{dB(i)} = k * B(i) * h/h & p_{dB(i)} = \{B(i)\} \end{array}$$

The rated derivation graph $\mathcal{D}_r(A_0(n) \underset{\emptyset}{\bowtie} B_0(n))$, where $n \in \mathbb{R}$, is shown in Figure 14, on the left.

2. concentration of molecule **C** is modelled with three agents, representing high, medium and low concentration. Species **C** inhibits its own production, so the functional rate associated to the production of **C** is inversely proportional to its concentration. Species **C** can also degrade according to mass action. Process definitions are as follows:

$$C_0(i) \triangleq pC(i).C_1(i) \qquad C_1(i) \triangleq pC(i).C_2(i) + dC(i).C_0(i)$$
$$C_2(i) \triangleq dC(i).C_1(i)$$

Reactions, functional rates and set of participants are as follows (parameters $k = 1$, $k' = 0.5$ and $h = 1$):

$$\begin{array}{lll} R_{pC} : \rightarrow \mathbf{C} & f_{pC(i)} = 1/(k' * h * (1 + C(i) * h)) & p_{pC(i)} = \{C(i)\} \\ R_{dC} : \mathbf{C} \rightarrow & f_{dC(i)} = k * C(i) * h/h & p_{dC(i)} = \{C(i)\} \end{array}$$

The rated derivation graph $\mathcal{D}_r(C_0(n))$, where $n \in \mathbb{R}$, is shown in Figure 14, on the right.

Notice that for all $n \in \mathbb{R}$, $A_0(n) \underset{\emptyset}{\bowtie} B_0(n) \simeq_{\mathcal{T},\Gamma} C_0(n)$ for any $\mathcal{T}$ such that $\mathcal{T} \cap \{\!|pA, pB, pC, dA, dB, dC|\!\} = \emptyset$ and with the appropriate environment Γ. In other words, the two psPAH model components are equivalent if we abstract away from the specific actions they can perform, retaining only the timing and likelihood of those actions. Recall that set $\mathcal{T}$ indicates on which actions rated derivation graphs $\mathcal{D}_r(A_0(n) \underset{\emptyset}{\bowtie} B_0(n))$ and $\mathcal{D}_r(C_0(n))$ should be compared. Moreover, a suitable environment Γ is a set of parameters where constant parameters of the two models are merged without conflict of names.

Assume now that the French Flag Model is updated with the addition of chemicals **A** and **B**, which do not interact with morphogen **M**, but that are nevertheless present in the system. Assume also that we are interested in comparing the behaviour of the resulting model with the behaviour of the French Flag Model updated with the addition of **C** instead. Without the need for looking at the actual behaviour of the two new systems, we can prove that their overall behaviour is identical. We prove this simply using the fact that $A_0(n) \underset{\emptyset}{\bowtie} B_0(n)$ and $C_0(n)$ are Markovian $(\mathcal{T}, \Gamma)$-bisimilar and the fact that Markovian $(\mathcal{T}, \Gamma)$-bisimulation is a congruence (Proposition 40) and it is transitive (Proposition 37).

Because of Proposition 40 we have that:

$$(A_0(1) \underset{\emptyset}{\bowtie} B_0(1)) \underset{\emptyset}{\bowtie} M(1, 20) \simeq_{\mathcal{T},\Gamma} C_0(1) \underset{\emptyset}{\bowtie} M(1, 20) \tag{1}$$

In the same way we have:

$$(A_0(2) \underset{\emptyset}{\bowtie} B_0(2)) \underset{\emptyset}{\bowtie} M(2, 0) \simeq_{\mathcal{T},\Gamma} C_0(2) \underset{\emptyset}{\bowtie} M(2, 0) \tag{2}$$

At this point, notice that whenever $X \simeq_{\mathcal{T},\Gamma} Y$ and $W \simeq_{\mathcal{T},\Gamma} Z$, then $X \underset{\mathcal{L}}{\bowtie} W \simeq_{\mathcal{T},\Gamma} X \underset{\mathcal{L}}{\bowtie} Z \simeq_{\mathcal{T},\Gamma} Y \underset{\mathcal{L}}{\bowtie} Z$. By the transitivity of $\simeq_{\mathcal{T},\Gamma}$ (Proposition 37), we have $X \underset{\mathcal{L}}{\bowtie} W \simeq_{\mathcal{T},\Gamma} Y \underset{\mathcal{L}}{\bowtie} Z$. Using this result and Equations (1) and (2) we have:

$$
\begin{array}{c}
(A_0(1) \underset{\emptyset}{\bowtie} B_0(1)) \underset{\emptyset}{\bowtie} M(1,20) \underset{\{\!|t(1,2)|\!\}}{\bowtie} \\
(A_0(2) \underset{\emptyset}{\bowtie} B_0(2)) \underset{\emptyset}{\bowtie} M(2,0) \\
\simeq_{\mathcal{T},\Gamma} \\
C_0(1) \underset{\emptyset}{\bowtie} M(1,20) \underset{\{\!|t(1,2)|\!\}}{\bowtie} \\
C_0(2) \underset{\emptyset}{\bowtie} M(2,0)
\end{array}
$$

Continuing this demonstration with the composition of the processes representing the remaining spatial regions we obtain:

$$
\begin{array}{c}
(A_0(1) \underset{\emptyset}{\bowtie} B_0(1)) \underset{\emptyset}{\bowtie} M(1,20) \underset{\{\!|t(1,2)|\!\}}{\bowtie} \\
(A_0(2) \underset{\emptyset}{\bowtie} B_0(2)) \underset{\emptyset}{\bowtie} M(2,0) \underset{\{\!|t(2,3),t(3,2)|\!\}}{\bowtie} \\
\cdots (A_0(20) \underset{\emptyset}{\bowtie} B_0(20)) \underset{\emptyset}{\bowtie} M(20,0) \\
\simeq_{\mathcal{T},\Gamma} \\
C_0(1) \underset{\emptyset}{\bowtie} M(1,20) \underset{\{\!|t(1,2)|\!\}}{\bowtie} \\
C_0(2) \underset{\emptyset}{\bowtie} M(2,0) \underset{\{\!|t(2,3),t(3,2)|\!\}}{\bowtie} \\
\cdots C_0(20) \underset{\emptyset}{\bowtie} M(20,0)
\end{array}
$$

And finally:

$$
\begin{array}{c}
(TA(1) \underset{\emptyset}{\bowtie} T(2,0,0) \cdots \underset{\emptyset}{\bowtie} T(19,0,0) \underset{\emptyset}{\bowtie} T(20,0,0)) \\
\underset{\{\!|x(1),y(1),\cdots,x(20),y(20)|\!\}}{\bowtie} \\
((A_0(1) \underset{\emptyset}{\bowtie} B_0(1)) \underset{\emptyset}{\bowtie} M(1,20) \underset{\{\!|t(1,2)|\!\}}{\bowtie} \\
(A_0(2) \underset{\emptyset}{\bowtie} B_0(2)) \underset{\emptyset}{\bowtie} M(2,0) \underset{\{\!|t(2,3),t(3,2)|\!\}}{\bowtie} \\
\cdots (A_0(20) \underset{\emptyset}{\bowtie} B_0(20)) \underset{\emptyset}{\bowtie} M(20,0)) \\
\simeq_{\mathcal{T},\Gamma} \\
(TA(1) \underset{\emptyset}{\bowtie} T(2,0,0) \cdots \underset{\emptyset}{\bowtie} T(19,0,0) \underset{\emptyset}{\bowtie} T(20,0,0)) \\
\underset{\{\!|x(1),y(1),\cdots,x(20),y(20)|\!\}}{\bowtie} \\
(C_0(1) \underset{\emptyset}{\bowtie} M(1,20) \underset{\{\!|t(1,2)|\!\}}{\bowtie} \\
C_0(2) \underset{\emptyset}{\bowtie} M(2,0) \underset{\{\!|t(2,3),t(3,2)|\!\}}{\bowtie} \\
\cdots C_0(20) \underset{\emptyset}{\bowtie} M(20,0))
\end{array}
\tag{3}
$$

Equation (3) finally proves that the addition of molecules **A** and **B** and the addition of molecule **C** to the French Flag Model have the same effect on

its spatio-temporal behaviour. The two extended versions of the French Flag Model are Markovian $(\mathcal{T}, \Gamma)$-bisimilar for any action set $\mathcal{T}$ as long as $\mathcal{T}$ does not contain actions of molecules **A**, **B** or **C** and Γ is the union of all constant parameters of the different parts composing the model, without conflicts of names. If this is the case we can assert that, for example, the commitment of the regions to a specialisation ($\mathcal{T} = \{\!|mem(1), \ldots, mem(20)|\!\}$) happens with the same timing and with the same probability in both models.

9. Related Work

PAH bears much similarity to other stochastic process algebras, in particular the syntax and horizontal, multi-way synchronisation operator is similar to PEPA [29]. Our introduction of hook actions to represent communication between scales and a vertical operator to represent synchronisation appears to be novel. We compare our approach with other process algebras that have prioritised or preemptive actions and the use of probes.

One alternative way to implement hook synchronisation might be to use a process algebra with priority such as EMPA [32]. In this setting, one could replace composed actions $a[h]$ with two consecutive actions a and h, where h has higher priority than a. Actions representing events happening within a scale would have the lowest priority, while higher priority could be used to represent events happening between scales. There are two disadvantages with this approach. First, actions with a certain priority would interleave with actions with different priorities generating additional intermediate states that could be avoided *a priori* using hook synchronisation. Second, when multiple high priority actions are available, multiple, non-deterministic paths of execution are possible. In PAH more control is given to the modeller concerning how to model the response of a scale to events generating multiple inter-scale effects at the same time. A detailed comparison of PAH with a simple process algebra and a process algebra with priority is given in [2].

Another alternative involves preemptive actions, as provided by the only process algebra we are aware of that aims specifically to model multi-scale biological systems in process algebra (PAPC) [26]. PAPC is an extension of CCS that introduces a distinction between preemptive and conservative actions and a new summation operation. These extensions allow the representation of actions that are running and have not yet terminated. This means that a set of actions might be running at the same time and interfere with each other, mimicking interactions between biological scales.

The concept of using processes to "observe" actions in a process algebra model was first introduced in [35] for testing whether a process can perform given computations, i.e. sequences of actions. In this work an example is given in the context of CCS. More recently, a similar approach was introduced with Probes [36, 37] in PEPA. In this setting, processes (probes) are constructed by regular expressions and are used to query a model. Special start and stop labels are added to certain actions to indicate entering and leaving states that satisfy the query. Although there are analogies, our approach does not aim to query the system, but to formalise and characterise the way we can observe its behaviour from different scales. Moreover, regular expressions might in some cases not be powerful enough to construct the processes that we need to observe biochemical actions.

Finally, our work bears similarities to logical modelling approaches such as René Thomas framework [38] and the Process Hitting formalism [39]. In analogy with our framework, in these approaches a fixed number of variables is modelled by a finite number of levels. Additionally, a threshold system is used to determine the nature of the interactions between variables, such as activator or inhibitor, while we use thresholds to determine interactions between biological scales.

10. Conclusions

PAH has been designed for multi-scale modelling of biological systems, using processes to represent biological scales, actions to represent biological events and functional rates to determine the timing of events. The introduction of composed actions and vertical synchronisation allows for the explicit representation of scales and interactions within and between scales. Moreover, features such as composition and relation of PAH processes are particularly amenable to the multi-scale modelling of biological systems, where they can be used to manipulate different scales in the same formal framework.

We introduced PAH through a series of examples, then we proceeded to the definition of the syntax and semantics of PAH, before providing definitions of compatibility $\mathcal{L}$-bisimulation and Markovian $(\mathcal{T}, \Gamma)$-bisimulation between PAH processes. We illustrated how the former relation can be used to aid model development for a model of tissue growth by checking that two scales have been defined correctly and can interact as intended by the modeller. We demonstrated how the latter can be used for reasoning about a known biological example: the french flag problem. In this system a group

of identical cells in a tissue are differentiated into subgroups according to the diffusion of a morphogen from a source. Differentiation depends upon the concentration in various regions and the standard model is a PDE. Here, we showed how to represent the problem in PAH, discretising both concentrations and spatial regions. We gave simulation analysis results computed using our bespoke interpreter. We then considered two possible extensions to the model: the addition of species A and B that are produced and degraded independently of the rest of the system, and the addition of species C that is also independent of the rest of the system and inhibits its own production. We used our Markovian $(\mathcal{T}, \Gamma)$-bisimulation to show that the two additions have the *same* behaviour, thus demonstrating an advantage of an algebraic framework over (partial) differential equations. Finally, we compared our approach with alternatives based on prioritised or preemptive events.

10.1. Future Work

Possible extensions of PAH include:

- *implementation of compatibility $\mathcal{L}$-bisimulation checker.* In Section 5, we have defined a relation between PAH processes. In order to use this in practice we need to define and implement an algorithm to check whether this relation holds for any pair of given processes. A possible direction is to follow an approach analogous to the algorithm for weak bisimulation discussed in [40];

- *approximate equivalence relations.* Markovian $(\mathcal{T}, \Gamma)$-bisimulation is a fundamental equivalence relations for PAH. However, biological applications require a more qualitative interpretation of behaviour, where two systems can be considered similar, though not identical. Approximate equivalence relations and distance measures between models [41] may prove an interesting direction to explore. If we associate locations with explicit coordinates and compute distances between processes in space, these metrics could be useful in the computation of approximate equivalence relations;

- *efficient stochastic simulations.* At the moment, the simulator that samples trajectories of transitions is not optimised. At each step the complete set of rated transitions is computed, while it could be the case that only a few processes are affected by the last step and that only a few transitions need to be updated. Improvements will depend

upon determining which processes are affected by a transition. With this information, known improvements to stochastic simulations of biochemical reactions and their diffusion, e.g. [42, 43], can be adapted and integrated in our approach;

- *Development of a high level representation for non-experts.* The process algebra framework may be difficult to use by non-experts. An important step toward the dissemination of our work in the systems biology community is the development of a higher level language or a graphical representation that would make this work more accessible to non-experts.

11. Acknowledgements

This research was supported by a Lord Kelvin/ Adam Smith Scholarship of the University of Glasgow and by the EPSRC funded SIGNAL project (http://homepages.inf.ed.ac.uk/stg/research/SIGNAL/).

References

[1] A. Degasperi, M. Calder, Multi-scale modelling of biological systems in process algebra with multi-way synchronisation, in: Proceedings of CMSB 2011, ACM Press, 2011, pp. 195–208.

[2] A. Degasperi, Multi-Scale Modelling of Biological Systems in Process Algebra, Ph.D. thesis, University of Glasgow, 2011.

[3] H. Kitano, Systems Biology: A Brief Overview, Science 295 (2002) 1662–1664.

[4] E. Klipp, R. Herwig, A. Kowald, C. Wierling, H. Lehrach, Systems Biology in practice, Wiley-Vch, 2005.

[5] H. Meinhardt, Models of biological pattern formation: from elementary steps to the organization of embryonic axes, Curr. Top. Dev. Biol. 81 (2008) 1–63.

[6] G. B. Ermentrout, L. E. Keshet, Cellular Automata Approaches to Biological Modelling, Journal of Theoretical Biology 160 (1993) 97–133.

[7] M. Hucka, A. Finney, H. M. Sauro, H. Bolouri, J. C. Doyle, H. Kitano, The systems biology markup language (SBML): a medium for representation and exchange of biochemical network models, Bioinformatics 19 (2003) 524–531.

[8] H. Kitano, A graphical notation for biochemical networks, BIOSILICO 1 (2003) 169–176.

[9] M. Calder, S. Gilmore, J. Hillston, Modelling the influence of RKIP on the ERK signalling pathway using the stochastic process algebra PEPA, LNCS Trans. on Computat. Syst. Biol. VII 4230 (2006) 1–23.

[10] C. Priami, P. Quaglia, Beta-binders for biological interactions, in: Lecture Notes in Computer Science, volume 3082, Springer Verlag, 2005, pp. 20–33.

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

[12] L. Bortolussi, A. Policriti, Modeling biological systems in stochastic concurrent constraint programming, Constraints 13 (2008) 66–90.

[13] V. Danos, J. Feret, W. Fontana, R. Harmer, J. Krivine, Rule-based modelling of cellular signalling, in: Lecture Notes in Computer Science, Springer Verlag, 2007, pp. 17–41.

[14] M. L. Blinov, J. R. Faeder, B. Goldstein, W. S. Hlavacek, BioNetGen: software for rule-based modeling of signal transduction based on the interactions of molecular domains, Bioinformatics 20 (2004) 3289–3291.

[15] L. Calzone, F. Fages, S. Soliman, Biocham: An environment for modelling biological systems and formalizing experimental knowledge, Bioinformatics 22 (2006) 1805–1807.

[16] M. Pedersen, G. Plotkin, A language for biochemical systems, in: Lecture Notes in Bioinformatics, volume 5307, Springer Verlag, 2008, pp. 63–82.

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

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

[19] A. Regev, W. Silverman, E. Shapiro, Representation and simulation of biochemical processes using the pi-calculus process algebra, Pacific Symposium on Biocomputing (2001) 459–470.

[20] C. Priami, A. Regev, E. Shapiro, W. Silverman, Application of a stochastic name-passing calculus to representation and simulation of molecular processes, Inf. Process. Lett. 80 (2001) 25–31.

[21] L. Cardelli, Brane calculi, in: Lecture Notes in Bioinformatics, volume 3082, Springer Verlag, 2005, pp. 257–278.

[22] J. O. Dada, P. Mendes, Multi-scale modelling and simulation in systems biology, Integrative Biology 3 (2011) 86–96.

[23] D. C. Walker, J. Southgate, The virtual cell–a candidate co-ordinator for 'middle-out' modelling of biological systems, Briefings in Bioinformatics 10 (2009) 450–461.

[24] D. C. Walker, N. T. Georgopoulos, J. Southgate, From pathway to population - a multiscale model of juxtacrine egfr-mapk signalling, BMC Systems Biology 2 (2008) 102.

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

[26] R. Barbuti, G. Caravagna, A. Maggiolo-Schettini, P. Milazzo, S. Tini, Foundational aspects of multiscale modeling of biological systems with process algebras, Theoretical Computer Science 431 (2012) 96–116.

[27] C. Athale, Y. Mansury, T. S. Deisboeck, Simulating the impact of a molecular 'decision-process' on cellular phenotype and multicellular patterns in brain tumors, Journal of Theoretical Biology 233 (2005) 469–481.

[28] Z. Wang, L. Zhang, J. Sagotsky, T. S. Deisboeck, Simulating non-small cell lung cancer with a multiscale agent-based model, Journal of Theoretical Biology 233 (2005) 469–481.

[29] J. Hillston, A Compositional Approach to Performance Modelling, Cambridge University Press, 1996.

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

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

[32] M. Bernardo, R. Gorrieri, Extended Markovian Process Algebra, in: Lecture Notes in Computer Science, volume 1119, Springer Verlag, 1996, pp. 315–330.

[33] L. Wolpert, The French flag problem: A Contribution to the Discussion on Pattern Formation and Regulation, Towards a Theoretical Biology 1 (1968) 125–133.

[34] U. Alon, An Introduction to Systems Biology, Chapman & Hall/CRC, 2006.

[35] R. D. Nicola, M. C. B. Hennessy, Testing equivalences for processes, Theoretical Computer Science (1984) 83–133.

[36] A. Argent-Katwala, J. T. Bradley, N. J. Dingle, Expressing performance requirements using regular expressions to specify stochastic probes over process algebra models, in: Fourth International Workshop on Software and Performance, ACM Press, 2004, pp. 49–58.

[37] A. Clark, S. Gilmore, State-aware performance analysis with eXtended stochastic probes, in: Lecture Notes in Computer Science, volume 5261, Springer Verlag, 2008, pp. 125–140.

[38] R. Thomas, M. Kaufman, Multistationarity, the basis of cell differentiation and memory. ii. logical analysis of regulatory networks in terms of feedback circuits, Chaos 11 (2001) 180–195.

[39] L. Paulevé, O. R. M. Magnin, Refining dynamics of gene regulatory networks in a stochastic p-calculus framework, T. Comp. Sys. Biology 13 (2011) 171–191.

[40] H. Hermanns, M. Siegle, Bisimulation Algorithms for Stochastic Process Algebras and their BDD-based Implementation, in: Lecture Notes in Computer Science, volume 1601, Springer Verlag, 1999, pp. 244–264.

[41] S. Tini, Non-expansive epsilon-bisimulations for probabilistic processes, Theoretical Computer Science 411 (2010) 2202–2222.

[42] J. Elf, M. Ehrenberg, Spontaneous separation of bi-stable biochemical systems into spatial domains of opposite phases, Systems Biology 1 (2004) 230–236.

[43] M. A. Gibson, J. Bruck, Efficient exact stochastic simulation of chemical systems with many species and many channels, The Journal of Physical Chemistry A 104 (2000) 1876–1889.

Appendix A. Multi-Sets

We use $\{\!|$ and $|\!\}$ to delimit a multi-set. For example, $A = \{\!|5,\ 6,\ 6,\ 7,\ 7,\ 7|\!\}$ is a multi-set.

We define a multi-set M as the pair (M', m_M), where M' is a set containing the same elements of M with no repetitions and m_M is the associated multiplicity function, such that for all $x \in M'$, $m_M(x)$ is equal to the number of times x appears in M. For all $x \in M'$, $m_M(x)$ is equal to zero if x does not appear in M. For example, multi-set A is defined as the pair (A', m_A), where $A' \subset \mathbb{N}$, $A' = \{5,\ 6,\ 7\}$, and $m_A : \mathbb{N} \to \mathbb{N}$, $m_A(5) = 1$, $m_A(6) = 2$ and $m_A(7) = 3$.

Given two multi-sets A and B, defined as (A', m_A) and (B', m_B), the following operations are defined:

- multi-set union: $A \cup B = (A' \cup B', m_{A \cup B})$, where for all $x \in A' \cup B'$, $m_{A \cup B}(x) = \max(m_A(x), m_B(x))$;
- multi-set sum: $A \uplus B = (A' \cup B', m_{A \uplus B})$, where for all $x \in A' \cup B'$, $m_{A \uplus B}(x) = m_A(x) + m_B(x)$;
- multi-set intersection: $A \cap B = (A' \cap B', m_{A \cap B})$, where for all $x \in A' \cap B'$, $m_{A \cap B}(x) = \min(m_A(x), m_B(x))$;
- multi-set difference: $A \setminus B = (A', m_{A \setminus B})$, where for all $x \in A'$, $m_{A \setminus B}(x) = \min(0, m_A(x) - m_B(x))$.

Moreover we define:

- $A \subseteq B \Leftrightarrow$ for all $x \in A' \cup B'$, $m_A(x) \leq m_B(x)$;
- $|A| = \sum_{x \in A'} m_A(x)$.

Appendix B. Proofs

Proof of Proposition 40. Proof of each case:

1. We know by assumption that $P_1 \simeq_{\mathcal{T},\Gamma} P_2$. This implies $P_1, P_2 \in \mathcal{C}$, with $\mathcal{C} \in \mathbb{P}_m/\simeq_{\mathcal{T},\Gamma}$. Now, $FiltMoves(\mathcal{A}[\mathcal{E}].P_1)_{\mathcal{T}} = \{\!|(((\mathcal{A}\backslash\mathcal{T}), \mathcal{E}), P_1)|\!\}$ and $FiltMoves(\mathcal{A}[\mathcal{E}].P_2)_{\mathcal{T}} = \{\!|(((\mathcal{A} \setminus \mathcal{T}), \mathcal{E}), P_2)|\!\}$. This implies $\forall \mathcal{C} \in \mathbb{P}_m/\simeq_{\mathcal{T},\Gamma}$

$$FiltCompAct(\mathcal{A}[\mathcal{E}].P_1, \mathcal{C})_{\mathcal{T}} = FiltCompAct(\mathcal{A}[\mathcal{E}].P_2, \mathcal{C})_{\mathcal{T}}$$

2. $P_1 + Q$ and $P_2 + Q$ are definition processes. Because $P_1 \simeq_{\mathcal{T},\Gamma} P_2$ we have that $\forall \mathcal{C} \in \mathbb{P}_m/\simeq_{\mathcal{T},\Gamma}$

$$\begin{gathered} FiltCompAct(P_1 + Q, \mathcal{C})_{\mathcal{T}} = \\ FiltCompAct(P_1, \mathcal{C})_{\mathcal{T}} \uplus FiltCompAct(Q, \mathcal{C})_{\mathcal{T}} = \\ FiltCompAct(P_2, \mathcal{C})_{\mathcal{T}} \uplus FiltCompAct(Q, \mathcal{C})_{\mathcal{T}} = \\ FiltCompAct(P_2 + Q, \mathcal{C})_{\mathcal{T}} \end{gathered}$$

3. We prove that the relation $\mathcal{R} = \{(P_1 \underset{\mathcal{L}}{\bowtie} Q, P_2 \underset{\mathcal{L}}{\bowtie} Q) \mid P_1 \simeq_{\mathcal{T},\Gamma} P_2\}$ is a Markovian $(\mathcal{T}, \Gamma)$-bisimulation. We consider four cases:
   (a) $\forall r \in \mathbb{R}$, $(\mathcal{A}, \mathcal{E}, r) \notin FiltAct(P_1 \underset{\mathcal{L}}{\bowtie} Q)_{\mathcal{T},\Gamma}$. This implies that $\forall r \in \mathbb{R}$, $(\mathcal{A}, \mathcal{E}, r) \notin FiltAct(P_2 \underset{\mathcal{L}}{\bowtie} Q)_{\mathcal{T},\Gamma}$, because a filtered activity for $P \underset{\mathcal{L}}{\bowtie} Q$ is either a filtered activity of P, a filtered activity of Q or it is derived from a closed activity which is the synchronisation of an open activity from P and one from Q and $P_1 \simeq_{\mathcal{T},\Gamma} P_2$ implies
      - $\forall r \in \mathbb{R}$, $(\mathcal{A}, \mathcal{E}, r) \notin FiltAct(P_1)_{\mathcal{T},\Gamma} \Leftrightarrow \forall r \in \mathbb{R}$, $(\mathcal{A}, \mathcal{E}, r) \notin FiltAct(P_2)_{\mathcal{T},\Gamma}$;
      - $\forall \mathcal{C} \in \mathbb{P}_m/\simeq_{\mathcal{T},\Gamma} OpenAct(P_1, \mathcal{C}) = OpenAct(P_2, \mathcal{C})$.

      It follows that $\forall \mathcal{C} \in \mathbb{P}_m/\simeq_{\mathcal{T},\Gamma}$

$$\nu_{\mathcal{T},\Gamma}(P_1 \underset{\mathcal{L}}{\bowtie} Q, \mathcal{A}, \mathcal{E}, \mathcal{C}) = \nu_{\mathcal{T},\Gamma}(P_2 \underset{\mathcal{L}}{\bowtie} Q, \mathcal{A}, \mathcal{E}, \mathcal{C}) = 0$$

(b) $\exists r \in \mathbb{R},\ ((\mathcal{A}, \mathcal{E}, r), P_1' \underset{\mathcal{L}}{\bowtie} Q) \in FiltMoves(P_1 \underset{\mathcal{L}}{\bowtie} Q)_{\mathcal{T},\Gamma}$. Recall that $P_1 \simeq_{\mathcal{T},\Gamma} P_2$ implies

$$\exists r \in \mathbb{R},\ ((\mathcal{A}, \mathcal{E}, r), P_1') \in FiltMoves(P_1)_{\mathcal{T},\Gamma} \Leftrightarrow$$
$$\exists r' \in \mathbb{R},\ ((\mathcal{A}, \mathcal{E}, r'), P_2') \in FiltMoves(P_2)_{\mathcal{T},\Gamma}$$

with $P_1' \simeq_{\mathcal{T},\Gamma} P_2'$, because $\nu_{\mathcal{T},\Gamma}(P_1, \mathcal{A}, \mathcal{E}, [P_1']_{\simeq_{\mathcal{T},\Gamma}}) = \nu_{\mathcal{T},\Gamma}(P_2, \mathcal{A}, \mathcal{E}, [P_1']_{\simeq_{\mathcal{T},\Gamma}})$. It follows that

$$\nu_{\mathcal{T},\Gamma}(P_1 \underset{\mathcal{L}}{\bowtie} Q, \mathcal{A}, \mathcal{E}, [P_1' \underset{\mathcal{L}}{\bowtie} Q]_{\mathcal{R}}) = \nu_{\mathcal{T},\Gamma}(P_1, \mathcal{A}, \mathcal{E}, [P_1']_{\simeq_{\mathcal{T},\Gamma}}) =$$
$$\nu_{\mathcal{T},\Gamma}(P_2, \mathcal{A}, \mathcal{E}, [P_1']_{\simeq_{\mathcal{T},\Gamma}}) = \nu_{\mathcal{T},\Gamma}(P_2 \underset{\mathcal{L}}{\bowtie} Q, \mathcal{A}, \mathcal{E}, [P_2' \underset{\mathcal{L}}{\bowtie} Q]_{\mathcal{R}})$$

It is also important to recall that if an activity is filtered then it is derived from a closed activity, which in turn means that no further synchronisation is possible via $\underset{\mathcal{L}}{\bowtie}$. This is ensured by Proposition 14 and Definition 15.

(c) $\exists r \in \mathbb{R},\ ((\mathcal{A}, \mathcal{E}, r), P_1 \underset{\mathcal{L}}{\bowtie} Q') \in FiltMoves(P_1 \underset{\mathcal{L}}{\bowtie} Q)_{\mathcal{T},\Gamma}$. It follows that

$$\nu_{\mathcal{T},\Gamma}(P_1 \underset{\mathcal{L}}{\bowtie} Q, \mathcal{A}, \mathcal{E}, [P_1 \underset{\mathcal{L}}{\bowtie} Q']_{\mathcal{R}}) = \nu_{\mathcal{T},\Gamma}(Q, \mathcal{A}, \mathcal{E}, [Q']_{\simeq_{\mathcal{T},\Gamma}}) =$$
$$= \nu_{\mathcal{T},\Gamma}(P_2 \underset{\mathcal{L}}{\bowtie} Q, \mathcal{A}, \mathcal{E}, [P_1 \underset{\mathcal{L}}{\bowtie} Q']_{\mathcal{R}})$$

Because $P_1 \simeq_{\mathcal{T},\Gamma} P_2$ and so $P_2 \underset{\mathcal{L}}{\bowtie} Q' \in [P_1 \underset{\mathcal{L}}{\bowtie} Q']_{\mathcal{R}}$.

(d) $\exists r \in \mathbb{R},\ ((\mathcal{A}, \mathcal{E}, r), P_1' \underset{\mathcal{L}}{\bowtie} Q') \in FiltMoves(P_1 \underset{\mathcal{L}}{\bowtie} Q)_{\mathcal{T},\Gamma}$. The filtered move $((\mathcal{A}, \mathcal{E}, r), P_1' \underset{\mathcal{L}}{\bowtie} Q')$ must be the result of a synchronisation between an open move from P_1 and an open move from Q. In particular, it must be that $\exists (\mathcal{A}_1[\mathcal{E}_1], \Delta_1) \in OpenAct(P_1, [P_1']_{\simeq_{\mathcal{T},\Gamma}})$ and $\exists (\mathcal{A}_2[\mathcal{E}_2], \Delta_2) \in OpenAct(Q, [Q']_{\simeq_{\mathcal{T},\Gamma}})$ such that:

- $\mathcal{B} = \mathcal{A}_1 \cup \mathcal{A}_2$, $\mathcal{E} = \mathcal{E}_1 \uplus \mathcal{E}_2$ and $\Delta = \Delta_1 \cup \Delta_2$;
- $(\mathcal{B}[\mathcal{E}], \Delta) \in ClosedAct(P_1 \underset{\mathcal{L}}{\bowtie} Q)$;
- $b \in \mathcal{B}$ and $f_b \in \mathbb{F}$ and $\Gamma \cup \Delta \vdash f_b \rightarrow k$ and $\mathcal{A} = \mathcal{B} \cap \mathcal{T}$;
- $r = k/\pi(ClosedAct(P_1 \underset{\mathcal{L}}{\bowtie} Q), b)$.

Moreover, from $P_1 \simeq_{\mathcal{T},\Gamma} P_2$ we have that

$$((\mathcal{A}_1[\mathcal{E}_1], \Delta_1), P_1') \in OpenMoves(P_1) \Leftrightarrow$$
$$((\mathcal{A}_1[\mathcal{E}_1], \Delta_1), P_2') \in OpenMoves(P_2)$$

with $P_1' \simeq_{\mathcal{T},\Gamma} P_2'$, because $OpenAct(P_1, [P_1']_{\simeq_{\mathcal{T},\Gamma}}) = OpenAct(P_2, [P_1']_{\simeq_{\mathcal{T},\Gamma}})$. It follows that

$$\nu_{\mathcal{T},\Gamma}(P_1 \underset{\mathcal{L}}{\bowtie} Q, \mathcal{A}, \mathcal{E}, [P_1' \underset{\mathcal{L}}{\bowtie} Q']_{\mathcal{R}}) = \nu_{\mathcal{T},\Gamma}(P_2 \underset{\mathcal{L}}{\bowtie} Q, \mathcal{A}, \mathcal{E}, [P_1' \underset{\mathcal{L}}{\bowtie} Q']_{\mathcal{R}})$$

where $P_2' \underset{\mathcal{L}}{\bowtie} Q' \in [P_1' \underset{\mathcal{L}}{\bowtie} Q']_{\mathcal{R}}$.

4. We prove that the relation $\mathcal{R} = \{(P_1 \underset{\mathcal{L}}{\boxtimes} Q, P_2 \underset{\mathcal{L}}{\boxtimes} Q) \mid P_1 \simeq_{\mathcal{T},\Gamma} P_2\}$ is a Markovian $(\mathcal{T}, \Gamma)$-bisimulation. We consider four cases:
   (a) $\forall r \in \mathbb{R},\ (\mathcal{A}, \mathcal{E}, r) \notin FiltAct(P_1 \underset{\mathcal{L}}{\boxtimes} Q)_{\mathcal{T},\Gamma}$. This implies that $\forall r \in \mathbb{R},\ (\mathcal{A}, \mathcal{E}, r) \notin FiltAct(P_2 \underset{\mathcal{L}}{\boxtimes} Q)_{\mathcal{T},\Gamma}$, because a filtered activity for $P \underset{\mathcal{L}}{\boxtimes} Q$ is either a filtered activity of P, a filtered activity of Q or it is derived from a synchronisation of a filtered activity from P and an open activity from Q or a filtered activity from Q and an open activity from P and $P_1 \simeq_{\mathcal{T},\Gamma} P_2$ implies
      - $\forall r \in \mathbb{R},\ (\mathcal{A}, \mathcal{E}, r) \notin FiltAct(P_1)_{\mathcal{T},\Gamma} \Leftrightarrow \forall r \in \mathbb{R},\ (\mathcal{A}, \mathcal{E}, r) \notin FiltAct(P_2)_{\mathcal{T},\Gamma}$;
      - $\forall \mathcal{C} \in \mathbb{P}_m / \simeq_{\mathcal{T},\Gamma} OpenAct(P_1, \mathcal{C}) = OpenAct(P_2, \mathcal{C})$.

      It follows that $\forall \mathcal{C} \in \mathbb{P}_m / \simeq_{\mathcal{T},\Gamma}$

$$\nu_{\mathcal{T},\Gamma}(P_1 \underset{\mathcal{L}}{\boxtimes} Q, \mathcal{A}, \mathcal{E}, \mathcal{C}) = \nu_{\mathcal{T},\Gamma}(P_2 \underset{\mathcal{L}}{\boxtimes} Q, \mathcal{A}, \mathcal{E}, \mathcal{C}) = 0$$

   (b) $\exists r \in \mathbb{R},\ ((\mathcal{A}, \mathcal{E}, r), P_1' \underset{\mathcal{L}}{\boxtimes} Q) \in FiltMoves(P_1 \underset{\mathcal{L}}{\boxtimes} Q)_{\mathcal{T},\Gamma}$. This implies that $((\mathcal{A}, \mathcal{E}, r), P_1') \in FiltMoves(P_1)_{\mathcal{T},\Gamma}$ and that $\mathcal{E} \cap \mathcal{L} = \emptyset$. Moreover,

$$\exists r \in \mathbb{R},\ ((\mathcal{A}, \mathcal{E}, r), P_1') \in FiltMoves(P_1)_{\mathcal{T},\Gamma} \Leftrightarrow$$
$$\exists r' \in \mathbb{R},\ ((\mathcal{A}, \mathcal{E}, r'), P_2') \in FiltMoves(P_2)_{\mathcal{T},\Gamma}$$

   with $P_1' \simeq_{\mathcal{T},\Gamma} P_2'$, because $\nu_{\mathcal{T},\Gamma}(P_1, \mathcal{A}, \mathcal{E}, [P_1']_{\simeq_{\mathcal{T},\Gamma}}) = \nu_{\mathcal{T},\Gamma}(P_2, \mathcal{A}, \mathcal{E}, [P_1']_{\simeq_{\mathcal{T},\Gamma}})$. It follows that

$$\nu_{\mathcal{T},\Gamma}(P_1 \underset{\mathcal{L}}{\boxtimes} Q, \mathcal{A}, \mathcal{E}, [P_1' \underset{\mathcal{L}}{\boxtimes} Q]_{\mathcal{R}}) = \nu_{\mathcal{T},\Gamma}(P_1, \mathcal{A}, \mathcal{E}, [P_1']_{\simeq_{\mathcal{T},\Gamma}}) =$$
$$\nu_{\mathcal{T},\Gamma}(P_2, \mathcal{A}, \mathcal{E}, [P_1']_{\simeq_{\mathcal{T},\Gamma}}) = \nu_{\mathcal{T},\Gamma}(P_2 \underset{\mathcal{L}}{\boxtimes} Q, \mathcal{A}, \mathcal{E}, [P_1' \underset{\mathcal{L}}{\boxtimes} Q]_{\mathcal{R}})$$

   (c) $\exists r \in \mathbb{R},\ ((\mathcal{A}, \mathcal{E}, r), P_1 \underset{\mathcal{L}}{\boxtimes} Q') \in FiltMoves(P_1 \underset{\mathcal{L}}{\boxtimes} Q)_{\mathcal{T},\Gamma}$. This implies that $((\mathcal{A}, \mathcal{E}, r), Q') \in FiltMoves(Q)_{\mathcal{T},\Gamma}$ and that $\mathcal{E} \cap \mathcal{L} = \emptyset$. Because $\forall \mathcal{C} \in \mathbb{P}_m / \simeq_{\mathcal{T},\Gamma},\ OpenAct(P_1, \mathcal{C}) = OpenAct(P_2, \mathcal{C})$ then

$$\nu_{\mathcal{T},\Gamma}(P_1 \underset{\mathcal{L}}{\boxtimes} Q, \mathcal{A}, \mathcal{E}, [P_1 \underset{\mathcal{L}}{\boxtimes} Q']_{\mathcal{R}}) = \nu_{\mathcal{T},\Gamma}(Q, \mathcal{A}, \mathcal{E}, [Q']_{\simeq_{\mathcal{T},\Gamma}}) =$$
$$\nu_{\mathcal{T},\Gamma}(P_2 \underset{\mathcal{L}}{\boxtimes} Q, \mathcal{A}, \mathcal{E}, [P_1 \underset{\mathcal{L}}{\boxtimes} Q']_{\mathcal{R}})$$

   (d) $\exists r \in \mathbb{R},\ ((\mathcal{A}, \mathcal{E}, r), P_1' \underset{\mathcal{L}}{\boxtimes} Q') \in FiltMoves(P_1 \underset{\mathcal{L}}{\boxtimes} Q)_{\mathcal{T},\Gamma}$. This implies that filtered move $((\mathcal{A}, \mathcal{E}, r), P_1' \underset{\mathcal{L}}{\boxtimes} Q')$ is the result of the

synchronisation of a filtered move of P_1 and an open move of Q or the synchronisation of a filtered move from Q and an open move from P_1.
In this case we have

$$\nu_{\mathcal{T},\Gamma}(P_1 \underset{\mathcal{L}}{\boxtimes} Q, \mathcal{A}, \mathcal{E}, [P_1' \underset{\mathcal{L}}{\boxtimes} Q']_{\mathcal{R}}) =$$
$$\sum_{i\in I} \nu_{\mathcal{T},\Gamma}(P_1, \mathcal{A}_i, \mathcal{E}_i, [P_1']_{\simeq_{\mathcal{T},\Gamma}}) + \sum_{j\in J} \nu_{\mathcal{T},\Gamma}(Q, \mathcal{A}_j, \mathcal{E}_j, [Q']_{\simeq_{\mathcal{T},\Gamma}}) =$$
$$\sum_{i\in I} \nu_{\mathcal{T},\Gamma}(P_2, \mathcal{A}_i, \mathcal{E}_i, [P_1']_{\simeq_{\mathcal{T},\Gamma}}) + \sum_{j\in J} \nu_{\mathcal{T},\Gamma}(Q, \mathcal{A}_j, \mathcal{E}_j, [Q']_{\simeq_{\mathcal{T},\Gamma}}) =$$
$$\nu_{\mathcal{T},\Gamma}(P_2 \underset{\mathcal{L}}{\boxtimes} Q, \mathcal{A}, \mathcal{E}, [P_1' \underset{\mathcal{L}}{\boxtimes} Q']_{\mathcal{R}})$$

Once again because

$$\exists r \in \mathbb{R},\ ((\mathcal{A}, \mathcal{E}, r), P_1') \in FiltMoves(P_1)_{\mathcal{T},\Gamma} \Leftrightarrow$$
$$\exists r' \in \mathbb{R},\ ((\mathcal{A}, \mathcal{E}, r'), P_2') \in FiltMoves(P_2)_{\mathcal{T},\Gamma}$$

with $P_1' \simeq_{\mathcal{T},\Gamma} P_2'$, because $\nu_{\mathcal{T},\Gamma}(P_1, \mathcal{A}, \mathcal{E}, [P_1']_{\simeq_{\mathcal{T},\Gamma}}) = \nu_{\mathcal{T},\Gamma}(P_2, \mathcal{A}, \mathcal{E}, [P_1']_{\simeq_{\mathcal{T},\Gamma}})$.
Moreover

- $i \in I$ if and only if $\exists \mathcal{B}, \mathcal{F}, \Delta, Q'$ s.t. $((\mathcal{B}[\mathcal{F}], \Delta), Q') \in Moves(Q)$ and $\mathcal{B} \subseteq \mathcal{E}_i \cap \mathcal{L}$ and $\mathcal{A} = \mathcal{A}_i \cup (\mathcal{B} \cap \mathcal{T})$ and $\mathcal{E} = (\mathcal{E}_i \setminus \mathcal{B}) \uplus \mathcal{F}$ and $\neg(\exists \mathcal{B}', \mathcal{F}', \Delta', Q''$ s.t. $((\mathcal{B}'[\mathcal{F}'], \Delta'), Q'') \in Moves(Q)$ and $\mathcal{B}' \subseteq \mathcal{E}_i \cap \mathcal{L}$ and $|\mathcal{B}'| > |\mathcal{B}|)$;
- $j \in J$ if and only if $\exists \mathcal{B}, \mathcal{F}, \Delta, P_1'$ s.t. $((\mathcal{B}[\mathcal{F}], \Delta), P_1') \in Moves(P)$ and $\mathcal{B} \subseteq \mathcal{E}_j \cap \mathcal{L}$ and $\mathcal{A} = \mathcal{A}_j \cup (\mathcal{B} \cap \mathcal{T})$ and $\mathcal{E} = (\mathcal{E}_j \setminus \mathcal{B}) \uplus \mathcal{F}$ and $\neg(\exists \mathcal{B}', \mathcal{F}', \Delta', Q''$ s.t. $((\mathcal{B}'[\mathcal{F}'], \Delta'), P_1'') \in Moves(P_1)$ and $\mathcal{B}' \subseteq \mathcal{E}_j \cap \mathcal{L}$ and $|\mathcal{B}'| > |\mathcal{B}|)$.

We also use the fact that $\forall \mathcal{C} \in \mathbb{P}_m/\ \simeq_{\mathcal{T},\Gamma},\ OpenAct(P_1, \mathcal{C}) = OpenAct(P_2, \mathcal{C})$.

Since we proved all four cases, the result holds.