# Stochastic modelling of cellular growth and division by means of the $\pi$@ calculus

Cristian Versari

Università di Bologna, Dipartimento di Scienze dell'Informazione
Mura Anteo Zamboni 7, 40127 Bologna, Italy
versari (at) cs.unibo.it

**Abstract**

The application of Concurrency Theory to Systems Biology is in its earliest stage of progress. The metaphor of cells as computing systems by Regev and Shapiro [14] opened the employment of concurrent languages for the modelling of biological systems. Their peculiar characteristics led to the design of many bio-inspired formalisms which achieve higher faithfulness and specificity.

In this paper we discuss the application to the biological modelling of $\pi$@, a core calculus for the representation of biological systems. The $\pi$@ language represents a keystone in this respect, thanks to its expressiveness capabilities which allow the modelling of a wide variety of phenomena (e.g. simple chemical reactions, but also formation of molecular or protein complexes, organisation of complex system in dynamical compartment hierarchies) despite of its simplicity and conservativeness. Here we analyse a biological case study involving cellular growth and division, modelled in the stochastic variant of $\pi$@: the case study is formalised and stochastically simulated according to a multi-compartment extension of Gillespie's stochastic simulation algorithm. The results underline the usefulness of the modelling approach adopted in $\pi$@ for the correct handling of systems with variable volume.

## 1 Introduction

The study of complex interactions at molecular and cellular level constitute the main focus of Systems Biology, a recent discipline aiming at the deep understanding of the behaviour of complex biological systems. The distinctive approach of this discipline consist in the assumption that the emerging behaviour of a complex system cannot be simply characterised by the exhaustive knowledge of its elements: it can only be captured by considering the system as a whole in the full complexity of interaction of its subparts. The implications of such approach are difficult to pursue as well as challenging: advances in the detailed knowledge of cellular dynamics are going to produce a deep impact on



Dagstuhl Seminar Proceedings 09091
Formal Methods in Molecular Biology
http://drops.dagstuhl.de/opus/volltexte/2009/1990

several research areas of primary importance – from the related fields of Bioinformatics to farther ones like Nanotechnology, all having important effects on many branches of medical research, e.g. Pharmacology, Preventive Medicine – and also affect the development of new disciplines such as Predictive and Personalised Medicine, Synthetic Biology, Natural Computing. On the other hand, such advances are bound to the application of cutting edge technologies or even to the development of newer ones for the collection of huge amounts of data, while new tools, techniques and computational power are needed for their analysis.

The assertion of cells as computing systems by Regev et al. [14] constituted an important step for the development of such techniques and tools for the analysis of complex biological systems, by proposing abstract computer languages as optimal candidates for the representation of biomolecular systems. The crucial idea behind this proposal was the metaphor of chemical/biological elements as processes, characterised by internal states and capable of interacting with the neighbour elements through biochemical reactions, so that complex biological systems could be thought, to some extent, as distributed computing devices.

A wide variety of languages (e.g. [15, 13, 4, 2, 12, 3]) has been designed and proposed ever since for the representation of biological systems, by focusing on peculiar biological phenomena and adapting or introducing new ad-hoc primitives and structures in the languages in order to achieve more faithful representations of the addressed systems. The stochastic variants of such calculi introduce in the models quantitative information (in particular *reaction rates*) which allow quantitative analyses to be applied such as stochastic simulation and ordinary differential equations.

In contrast with the current trend which leads to more faithful languages but also increasingly complex and specialised, the $\pi@$ calculus has been designed in order to be as minimal as possible and to provide, at the same time, general applicability and extreme flexibility. This language is strongly based on the $\pi$-calculus, which represents an optimal starting point for its broad applicability. *Mobility* constitutes the peculiar feature of this calculus, expressed by the capability of establishing new communication links between processes thanks to the transmission of new channel names over the existing communication channels.

The addition of polyadic synchronisation and priority to the $\pi$-calculus allows the expression of biological structures and the achievement of atomicity in concurrent settings, so that complex operations such as those modelled by bio-inspired formalisms can be straightforwardly encoded. In this respect, the expressiveness of $\pi@$ has been already demonstrated as capability of encoding several other bio-inspired formalisms [16, 17, 19].

The direct application of $\pi@$ to the biological modelling has been accomplished by the formulation of a stochastic variant of the calculus, S$\pi@$ [18], whose definition is associated with an extension of one of the most exploited simulation algorithms – Gillespie's [6] – modified so that compartments with varying volumes can be properly taken into account within a purely stochastic approach.

In this paper we consider the formalisation in S$\pi@$ of a biological model

where such possibility of considering compartments with variable volume plays a crucial role for the correct analysis of the system. The model describes the behaviour of a *constitutive promoter* during the process of cellular growth and division. This model was firstly analysed in [7] under the hypothesis of constant volume, while in [8] a hybrid variant of Gillespie's stochastic simulation algorithm was formulated in order to take into account the variation of volume of the cell during the process of growth and subsequent division. The analysis presented here, whose results are in perfect agreement with those presented in [8], constitute a proof of concept of how complex phenomena like cellular growth and division can be straightforwardly formalised in a pure stochastic framework by means of the $\pi@$ language.

### 1.1 Structure of the paper

The paper is structured as follows. In the next section, the stochastic variant of the $\pi@$ calculus, S$\pi@$, is formalised. In order to introduce the reader to the $\pi@$ language and the related biological modelling, a short explanation is given about the basic principles exploited for the modelling by means of the $\pi$-calculus, on which $\pi@$ is strongly based. Then polyadic synchronisation, the extension to the stochastic $\pi$-calculus through which S$\pi@$ is obtained, is described. In Sect. 3 the formalisation and stochastic analysis by means of S$\pi@$ of a generic constitutive promoter in the presence of cellular growth and division is presented, while in Sect. 4 some final remarks are reported.

## 2 The stochastic $\pi@$ calculus

In this section we present the stochastic $\pi@$ language — a simple extension to the stochastic $\pi$-calculus, obtained with the addition of polyadic synchronisation, which allows for the modelling of compartments in a natural way, but still in the classic message-passing flavour typical of the $\pi$-calculus. Priority, which in the standard $\pi@$ language is extremely useful for implementing transactional mechanisms that are essential when dealing with complex operations, is recovered by means of *immediate reactions*, that is reactions denoted by *infinite rate*. For an exhaustive introduction to the $\pi@$ language and its variants we refer to [19, 18].

### 2.1 Biochemical modelling in the $\pi$-calculus

The $\pi$-calculus [9, 10, 13] is a simple calculus of concurrent processes which interact through synchronisation over named channels, with the capability of receiving new channels and subsequently using them for the interaction with other processes, in order to model mobility.

Names constitute the basic entities of the calculus. Each name represents a channel which can be used for synchronisation by parallel processes. For

example, the system

$$a(x).P \mid \overline{a}\langle z\rangle.Q \tag{1}$$

represents two parallel processes $a(x).P$ and $\overline{a}\langle z\rangle.Q$, the first one ready to receive some datum (whose local name is $x$) over the channel $a$, the second one ready to send some datum $z$ over the same channel $a$. The datum $z$ represents in turn another channel, which can be used by the first process for subsequent communications.

If $\overline{a}\langle z\rangle.Q$ sends $z$ to $a(x).P$, then the subsequent behaviour of the two processes is specified by the expressions $Q$ and $P$ respectively. More precisely, we write that the system of Expr. (1) may evolve in the following way:

$$a(x).P \mid \overline{a}\langle z\rangle.Q \quad \rightarrow \quad P\{z/x\} \mid Q \tag{2}$$

where $P\{z/x\}$ represents the process $P$ where all the occurrences of the placeholder $x$ have been replaced by $z$. Here, $x$ is said to be a *bound name*, in opposition to $a$ which is *free*.

The *transition* of the system $a(x).P \mid \overline{a}\langle z\rangle.Q$ to the system $P\{z/x\} \mid Q$ is governed by to *reduction relation* "$\rightarrow$", which states that two processes may exchange data if they are ready to perform input/output respectively over the same channel.

The *nondeterministic choice* between two (or more) possible transitions is denoted by the *choice operator "+"*. For example, in the system

$$a(x).P' + b(y).P'' \mid \overline{a}\langle z\rangle.Q \mid \overline{b}\langle w\rangle.R$$

the first process may undergo two different, equally possible transitions, caused by a synchronisation with the second process or the third one, respectively. The first transition can be written as

$$a(x).P' + b(y).P'' \mid \overline{a}\langle z\rangle.Q \mid \overline{b}\langle w\rangle.R \quad \rightarrow \quad P'\{z/x\} \mid Q \mid \overline{b}\langle w\rangle.R$$

while the second as

$$a(x).P' + b(y).P'' \mid \overline{a}\langle z\rangle.Q \mid \overline{b}\langle w\rangle.R \quad \rightarrow \quad P''\{w/y\} \mid \overline{a}\langle z\rangle.Q \mid R$$

Depending on the occurring transition, the future behaviour of the first process is denoted by $P'\{z/x\}$ or $P''\{w/y\}$ respectively.

Since the order used for enumerating the possible choices is meaningless, i.e. the choice operator is commutative (and associative), we write that

$$a(x).P' + b(y).P'' \quad \equiv \quad b(y).P'' + a(x).P'$$

where "$\equiv$" represents a *congruence relation* between processes that are meant to be characterised by the same behaviour.

In order to model recursive behaviour, an operator of *replication* is introduced in the language. A process $P$ preceded by "!" is thought as being replicated an unlimited number of times. That is

$$!\, P \equiv P \mid P \mid \cdots$$

The key idea behind the modelling of biological systems by means of the $\pi$-calculus is that biochemical elements can be seen as parallel processes, and their interaction as communication. In particular, each *molecule* of the system can be represented by a process and its *reaction* with other molecules can be modelled as a communication over a fixed channel. For example, the chemical reaction

$$R: \quad R_1 + R_2 \;\rightarrow\; P_1 + P_2$$

where the molecules $R_1$ and $R_2$ react according to reaction $R$, and release $P_1$ and $P_2$ as products of the reaction, can be modelled in $\pi$-calculus as

$$R_1 \triangleq r.P_1 \qquad R_2 \triangleq \overline{r}.P_2 \qquad\qquad R_1 \mid R_2 \quad\rightarrow\quad P_1 \mid P_2$$

where each process is named as the corresponding molecule, and reaction $R$ is associated with channel $r$.

Furthermore, the communication of restricted names between $\pi$-calculus processes can be exploited for the modelling of local bounds between molecules. If $M_1$ and $M_2$ represent two molecules ready to bind, the corresponding expression in $\pi$-calculus is

$$M_1 \triangleq (\nu\, b)(\overline{bind}\langle b\rangle.M_1') \qquad M_2 \triangleq bind(x).M_2'$$
$$M_1 \mid M_2 \quad\rightarrow\quad (\nu\, b)(M_1' \mid M_2'\{b/x\}) \qquad\qquad (b \notin \mathrm{fn}(M_2'))$$

where $M_1'$ and $M_2'\{b/x\}$ (and no other process) share the name $b$ after their reaction.

## 2.2 Polyadic Synchronisation

In the $\pi$-calculus previously introduced, channels and transmitted names are usually synonyms. Polyadic synchronisation [1] consists in giving *structure* to channels: each channel is composed of one or more names and identified by all of them in relation to the exact sequence of their occurrence. For example, an email address is usually written in the form *username*@*domain*, where *username* and *domain* are two strings – two names – both necessary to identify the given email address. Moreover, their order is crucial since *domain*@*username* specifies another, likely nonexistent, address. Similarly, polyadic synchronisation (in its simplest form) provides the capability of writing channels as $name_1@name_2$. In other words, a channel is indicated by a vector of two names $(name_1, name_2)$ and communication between two processes may happen only if they are pursuing a synchronisation along channels denoted by the same names.

Apart from this, communication happens in the same way as in the $\pi$-calculus. For example, the transition

$$\overline{polyadic@comm}\langle d\rangle.P \mid polyadic@comm(x).Q \quad\rightarrow\quad P \mid Q\{d/x\}$$

produces the same renaming effect of a $\pi$-calculus transition, but with one difference: in the $\pi$-calculus, the transmission of a name always stands for the transmission of a channel, while in the above example the transmitted name constitutes only one component of it.

## 2.3 S$\pi$@ syntax and semantics

We formalise now the S$\pi$@ language. The terms of the language are specified through a simple grammar based on two sets of names $\mathcal{N}, \mathcal{C}$ representing reaction channels and compartments respectively.

**Definition 1** *Let $\mathcal{N}, \mathcal{C}$ be distinct sets of names on a finite alphabet, with $m, n$ ranging over $\mathcal{N}$, $a, b$ over $\mathcal{C}$ and $x, y$ over $\mathcal{X} = \mathcal{N} \cup \mathcal{C}$. Also let $v$ range over $\mathbb{R}$ within the interval $[0, +\infty[$. The syntax of the S$\pi$@ language is defined as*

$$\begin{array}{lcl} P & ::= & \mathbf{0} \;\Big|\; \displaystyle\sum_{i \in I} \pi_i.P_i \;\Big|\; P \mid Q \;\Big|\; !\,\pi.P \;\Big|\; (\nu\, x)P \\ \pi & ::= & \tau_r \;\Big|\; n@a : v(\mathbf{x}) \;\Big|\; \overline{n}@a : v\langle \mathbf{x} \rangle \end{array}$$

*where $\mathbf{x}$ represents zero or more names $x_1, \ldots, x_i$ ranging over $\mathcal{X}$.*

The meaning of the above syntax closely follows that of the standard $\pi$-calculus:

- $\mathbf{0}$ is the null process, capable of doing nothing;
- $\sum_{i \in I} \pi_i.P_i$, written also $\pi_1.P_1 + \pi_2.P_2$ in the case $|I| = 2$, represents the guarded choice between different actions;
- $P \mid Q$ means that $P$ and $Q$ are two processes executing in parallel;
- $!\,\pi.P$ represents guarded replication, which allows the expression of recursive behaviour in $\pi$-like calculi;
- $(\nu\, x)P$ allows the scope restriction of the name $x$: the restriction of compartment names allows the creation of new compartments, while the restriction of reaction names is used in several ways, such as for representing bindings between different elements;
- $\tau_r$ represents an internal transition (silent action) characterised by exponential rate $r \in \mathbb{R} \cup \{\infty\}$.

The expressions $n@a : v(\mathbf{x})$ and $\overline{n}@a : v\langle \mathbf{x} \rangle$ represent respectively the polyadic input and output capabilities of a process, where

- $n$ is the kind of reaction the process is ready to perform;
- $a$ is the compartment where the reaction may take place;
- $v$ represents the micro-volume occupied inside compartment $a$ by the process ready to perform the input or output action.

The micro-volumes of the elements inside each compartments determine the total volume of the compartment itself, so that the variation of the volume can

be expressed as linear function of the (multiplicity and type of the) enclosed elements and treated accordingly in a purely stochastic manner.

As pointed out in the description of the choice operator in the $\pi$-calculus, the following congruence relations allows the definition of processes which are syntactically different, but whose desired behaviour is the same. It states, for example, that the parallel operator is commutative and associative, so that $P \mid Q \mid R$ is not different from $R \mid P \mid Q$.

**Definition 2** *The congruence relation $\equiv$ is defined as the least congruence satisfying alpha conversion, the commutative monoidal laws with respect to both ($\mid$ ,$\mathbf{0}$) and ($+$,$\mathbf{0}$) and the following axioms:*

$$\begin{aligned} (\nu\, x)P \mid Q &\equiv (\nu\, x)(P \mid Q) && \text{if } x \notin fn(Q) \\ (\nu\, x)P &\equiv P && \text{if } x \notin fn(P) \\ !\,\pi.P &\equiv \pi.(!\,\pi.P \mid P) && \text{if } \mathrm{fn}(\pi) \cap \mathrm{bn}(\pi) = \emptyset \end{aligned}$$

*where the function* fn *is defined as*

$$\begin{aligned} \mathrm{fn}(n@a\!:\!v(\mathbf{x})) &\triangleq \{n, a\} & \mathrm{fn}(\overline{n}@a\!:\!v\langle\mathbf{x}\rangle) &\triangleq \{n, a, \mathbf{x}\} \\ \mathrm{fn}(\mathbf{0}) = \mathrm{fn}(\tau_r) &\triangleq \emptyset & \mathrm{fn}((\nu\, x)P) &\triangleq \mathrm{fn}(P) \setminus \{x\} \\ \mathrm{fn}(\pi.P) &\triangleq \mathrm{fn}(\pi) \cup \mathrm{fn}(P) \setminus \mathrm{bn}(\pi) & \mathrm{fn}(\sum_{i\in I} \pi_i.P_i) &\triangleq \bigcup_i \mathrm{fn}(\pi_i.P_i) \\ \mathrm{fn}(P \mid Q) &\triangleq \mathrm{fn}(P) \cup \mathrm{fn}(Q) & \mathrm{fn}(!\,\pi.P) &\triangleq \mathrm{fn}(\pi.P) \end{aligned}$$

*with* $\mathrm{bn}(\pi) \triangleq \{\mathbf{x}\}$ *if* $\pi = n@a\!:\!v(\mathbf{x})$ *or* $\mathrm{bn}(\pi) \triangleq \emptyset$ *otherwise.*

Finally, the operational semantics of S$\pi$@ is specified through a relation between processes consisting in few reduction rules.

**Definition 3** *S$\pi$@ semantics is given in terms of the following reduction system:*

$$(S)\ \frac{r = \infty \ \vee \ M \xrightarrow{\infty} M'}{\tau_r.P + M \xrightarrow{r} P}$$

$$(C)\ \frac{\mathrm{rate}(n) = \infty \ \vee \ M \mid N \xrightarrow{\infty} S}{(n@a\!:\!v_1(\mathbf{x}).P + M) \mid (\overline{n}@a\!:\!v_2\langle\mathbf{y}\rangle.Q + N) \xrightarrow{\mathrm{rate}(n)} P\{\mathbf{y}/\mathbf{x}\} \mid Q}$$

$$(R)\ \frac{P \xrightarrow{r} P'}{(\nu\, x)P \xrightarrow{r} (\nu\, x)P'} \qquad (P)\ \frac{P \xrightarrow{r} P' \qquad r = \infty \ \vee \ P \mid Q \xrightarrow{\infty} S}{P \mid Q \xrightarrow{r} P' \mid Q}$$

$$(E)\ \frac{P \equiv Q \quad P \xrightarrow{r} P' \quad P' \equiv Q'}{Q \xrightarrow{r} Q'}$$

The rule $(S)$ models the internal, silent transition of a process while the rule $(C)$ allows the communication of the names $\mathbf{x}$ from process $P$ to $Q$, where they are properly substituted to names $\mathbf{y}$. The function

$$\text{rate} : \mathcal{N} \to (\mathbb{R} \cup +\infty) \tag{3}$$

is an external function which permits us to associate the correct rate with each reaction. Rules $(R), (P), (E)$ allow the transition of processes in the presence of restriction and of parallel operator, or by exploiting structural equivalence.

**Definition 4** *A Sπ@ system $S$ is said to be in* standard form *if*

$$S = (\nu\, \mathbf{x})\big(P_1 \mid \cdots \mid P_j \mid !\, P_{j+1} \mid \cdots \mid !\, P_k\big)$$

*and each $P_i$ is a non-empty sum.*

The standard form constitutes a more readable way to write (and an easier way to handle) systems in $\pi$-like calculi: restricted names are all collected on the left and replicated processes are listed after the non-replicated ones.

**Proposition 5** *For every Sπ@ system $S$, there exists a system $S'$ such that $S \equiv S'$ and $S'$ is in standard form.*

The function Act which permits us to know the number of possible combinations of inputs and outputs on a reaction channel inside a given compartment or the number of silent actions of a given rate.

**Definition 6** *The activity* Act *of an action $\pi$ is defined as*

$$\text{Act}_\pi(S) = (\text{In}_{n@a}(S) \cdot \text{Out}_{n@a}(S)) - \text{Mix}_{n@a}(S)$$

*if $\pi = n@a$, corresponding to channel $n$ inside compartment $a$ in the system $S$, and*

$$\text{Act}_\pi(S) = \text{Num}_{\tau_r}(S)$$

*if $\pi = \tau_r$. $S$ is in standard form, $\text{In}_{n@a}(S)$ and $\text{Out}_{n@a}(S)$ are the number of unguarded inputs and outputs on channel $n$ inside compartment $a$, and $\text{Mix}_{n@a}(S)$ is the sum of $\text{In}_{n@a}(\sum_i) \cdot \text{Out}_{n@a}(\sum_i)$ for each summation $\sum_i$ in $S$. $\text{Num}_{\tau_r}(S)$ is the number of silent transitions of rate $r$ in $S$.*

The function chan returns all the active channels inside each compartment in a given system $S$.

**Definition 7** *Given a Sπ@ system $S$ in standard form*

$$S = (\nu\, \mathbf{x})\big(P_1 \mid \cdots \mid P_j \mid !\, P_{j+1} \mid \cdots \mid !\, P_k\big)$$

*the function* chan *is defined recursively as follows:*

$$\begin{aligned}
\mathrm{chan}(S) &= \bigcup_{i=1}^{k} \mathrm{chan}(P_i) \\
\mathrm{chan}(\sum_{i\in I} \pi_i.P_i) &= \bigcup_{i\in I} \mathrm{chan}(\pi_i) \\
\mathrm{chan}(n@a : v(\mathbf{x})) &= \{n@a\} \\
\mathrm{chan}(\overline{n}@a : v\langle\mathbf{x}\rangle) &= \{n@a\} \\
\mathrm{chan}(\tau_r) &= \{\tau_r\}
\end{aligned}$$

The function Vol calculates the volume of each compartment as the sum of the micro-volumes of the elements inside it.

**Definition 8** *Given a S$\pi$@ system $S$ in standard form*

$$S = (\nu\ \mathbf{x})\big(P_1 \mid \cdots \mid P_j \mid !\, P_{j+1} \mid \cdots \mid !\, P_k\big)$$

*the volume* $\mathrm{Vol}_a$ *of the compartment $a$ in the system $S$ is calculated as follows:*

$$\begin{aligned}
\mathrm{Vol}_a(S) &= \sum_{i=1}^{k} \mathrm{Vol}_a(P_i) \\
\mathrm{Vol}_a(\sum_{i\in I} \pi_i.P_i) &= \sum_{i\in I} \mathrm{Vol}_a(\pi_i) \\
\mathrm{Vol}_a(\tau_r) &= 0 \\
\mathrm{Vol}_a(\overline{n}@a : v\langle\mathbf{x}\rangle) &= \mathrm{Vol}_a(n@a : v(\mathbf{x})) \\
\mathrm{Vol}_a(n@b : v(\mathbf{x})) &= \begin{cases} v & a = b \\ 0 & \text{otherwise} \end{cases}
\end{aligned}$$

*If* $\mathrm{Vol}_a(S) = 0$*, then $a$ is given the default volume value* 1*.*

Each molecule is represented by a choice $\sum_{i\in I} \pi_i.P_i$, which may occupy volume in more than one compartment: $Vol_a$ considers only that part of the molecule falling inside compartment $a$.

The following algorithm determines the stochastic choice of the next reaction to be executed (and the elapsed time before its execution), in dependence of the activity of each reaction channel.

**Algorithm 1** *Given a S$\pi$@ system $S$ in standard form, the selection of the next reaction* $\mathrm{Next}(S)$ *and of the delay* $\mathrm{Delay}(S)$ *relative to the MSSA are described by the following algorithm:*

1. *For each channel $c_i$ in* $\mathrm{chan}(S)$*, with* $\mathrm{chan}(S) = \{c_1, \ldots, c_j\}$*, calculate*

$$a_i = \mathrm{Act}_{n@b}(S) * \mathrm{rate}(n) / \mathrm{Vol}_b(S)$$

| | Reaction | | | Propensity function |
|---|---|---|---|---|
| $R_1$ : | $S_0 + S_{Act}$ | $\xrightarrow{k_1}$ | $S_1$ | $k_1 \cdot [S_0] \cdot [S_{Act}]$ |
| $R_2$ : | $S_1$ | $\xrightarrow{k_{-1}}$ | $S_0 + S_{Act}$ | $k_{-1} \cdot [S_1]$ |
| $R_3$ : | $S_0$ | $\xrightarrow{\alpha_0}$ | $S_0 + X$ | $\alpha_0 \cdot [S_0]$ |
| $R_4$ : | $S_1$ | $\xrightarrow{\alpha_1}$ | $S_1 + X$ | $\alpha_1 \cdot [S_1]$ |
| $R_5$ : | $X$ | $\xrightarrow{k_x}$ | | $k_x \cdot [X]$ |

Table 1: Biochemical reactions for a simple system describing a constitutive promoter $S$.

*if* $c_i = n@b$ *for some* $n \in \mathcal{N}, b \in \mathcal{C}$ *or*

$$a_i = \mathrm{Act}_{\tau_r}(S) * r$$

*if* $c_i = \tau_r$.

2. *Calculate* $a_0 = \sum_{i=1}^{j} a_i$

3. *Generate two random numbers* $z_1, z_2 \in [0, 1]$ *and calculate* $\tau, \lambda$ *such that*

$$\tau = (1/a_0)\ln(1/z_1) \qquad \sum_{i=1}^{\lambda-1} a_i < z_2 a_0 \leq \sum_{i=1}^{\lambda} a_i$$

4. $\mathrm{Next}(S) = c_\lambda$ *and* $\mathrm{Delay}(S) = \tau$.

The value $c_\lambda = \tau_r$ for some $r$ or $c_\lambda = n@b$ for some $n, b$ denotes the rate of the silent action or the reaction channel $n$ and the compartment $b$ of the next reaction happening after $\tau$ time. The process performing the silent transition or the two processes performing the synchronisation step on $c_\lambda$ are then randomly chosen as for SPiM [11].

# 3 Cell growth and division

In this section we consider the effect of the variation of volume as a consequence of cellular growth and division on the system in Table 1, which was firstly analysed in [7] under the hypothesis of constant volume, while in [8] a hybrid variant of Gillespie's stochastic simulation algorithm was formulated in order to take into account the variation of volume of the cell during the process of growth and subsequent division.

The system involves a single gene which fluctuates between two states $S_0$ and $S_1$. The transition $S_0 \to S_1$ occurs when one regulator protein $S_{Act}$ binds to the gene's promoter, while the reverse transition $S_1 \to S_0$ is supposed to

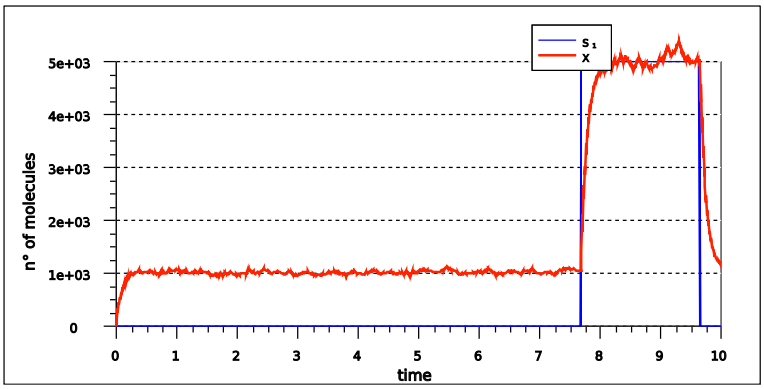


Figure 1: Simulation of the system specified in table 1 in the case of a single molecule of the promoter $S$ (in the graph its level is multiplied by $5 \cdot 10^3$) and constant volume $Vol = 1$, with $k_1 = k_{-1} = 0.1$, $\alpha_0 = 1 \cdot 10^4$, $\alpha_1 = 5 \cdot 10^4$, $k_x = 10$.

occur autonomously. The transcription of gene $S$ leads to the production of the protein $X$ at rate $\alpha_0$ when its promoter is in the state $S_0$, and at rate $\alpha_1$ when it is in the state $S_1$ (with $\alpha_0 < \alpha_1$). The protein $X$ spontaneously degrades at rate $k_x$.

Under standard conditions, the concentration of the protein $X$ follows a bistable condition which depends on the state of the gene $S$. In the state $S_0$, the concentration level of $X$ quickly reaches an equilibrium at $\alpha_0/k_x$, while in $S_1$ at $\alpha_1/k_x$, as shown in Fig. 1.

We shall now consider the same system in the case of volume variation as a consequence of the growth and division of the cellular compartment. In [8], such growth obeys a deterministic (exponential) function and the division occurs at fixed time steps. Here we are following an alternative approach, which totally sticks to the stochastic spirit of Gillespie's simulation algorithm.

The exponential growth of the *cell* compartment can be easily reproduced by means of the introduction of a new chemical species represented by the process $BH2O$ with micro-volume $v$ which may represent the amount of water (and other substances) inside the cell itself. The following $\pi$@ expression

$$\begin{aligned} BH2O &\equiv H2O \mid !\, w@cell.(H2O \mid H2O) \\ H2O &\equiv \overline{w}@cell : v + killw@cell \end{aligned}$$

seamlessly introduces an exponential growth of cellular volume (whose rate is a function of the rate associated with the channel $w$), supposing that the volume occupied by all the other elements is negligible.

The event of cell division can be signalled by the stochastic appearance of a single *killer* molecule represented by the process $BK$, which dramatically changes the configuration of the system through a series of infinite-rate reduc-

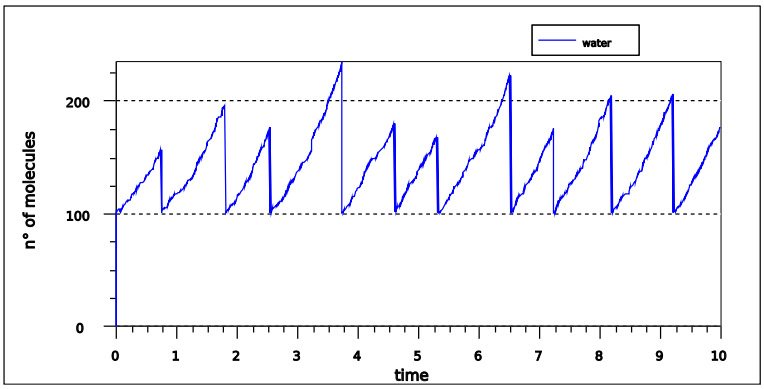


Figure 2: Simulation of stochastic volume variation in the case of growing and dividing cells. The number of water molecules exponentially grows until the division of the cell restores its initial amount.

tions. It first sets the volume of the cell to zero by causing the disappearance of all the processes representing water molecules, then it "stochastically halves" the number of proteins of the species $X$ by sending them in two different compartments (representing the two new cells):

$$\begin{aligned} BK &\equiv K \mid !\, k@cell.K \\ K &\equiv \overline{killw}@cell.\overline{k}@cell + \\ &\quad \overline{movex}@cell.(\overline{k}@cell \mid \tau_\infty.\mathbf{0} + \tau_\infty.X')) + \\ &\quad \tau_{dk}.(K_0 \mid \underbrace{W \mid \cdots \mid W}_{n}) \end{aligned}$$

where all the reactions have infinite rate, except for $\tau_{dk}$. The reactions over the channel $killw@cell$ eliminates all the water molecules, while the reactions over $movex@cell$ simulate the division of the cytosol between the two budding cells: the sequence of infinite-rate reactions replaces roughly half of the occurrences of $X$ with $X'$, which represents the protein $X$ inside one of the two new cells, while the other half is ideally moved into the other budding cell, which is not explicitly modelled.

After a negligible time, the species $K$ disappears according to the reaction $\tau_{dk}$ degradation with (very high but not infinite) rate $dk$, which also restores the initial volume of the cell $Vol = n \cdot v$ (to be precise, such volume should be "stochastically halved", but this expedient constitutes an effective way to limit its fluctuation, in the absence of the reliable control mechanisms typical of real biological cells).

A simulation of the fluctuation of the number of water molecules (and consequently of the volume of the system, according to the previous hypotheses) produced by such reaction rules is shown in Fig. 2.

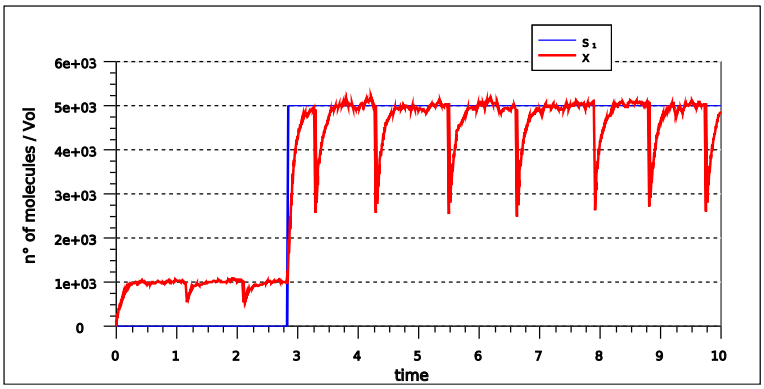


Figure 3: Simulation of cellular growth and division with complete neglect of volume variation: the concentration of protein $X$ quickly reaches the equilibrium after each cell division, which happens approximately once per time unit.

As we have previously discussed, in a standard situation the concentration of the $X$ protein in the cell reaches an equilibrium which depends on the state of the gene $S$. Upon cell division, the number of proteins is halved. If we disregard the variation of volume, the concentration of $X$ is suddenly halved after each cell division, then quickly reaches the previous equilibrium level, according to the simulation in Fig. 3.

The behaviour of the system considerably changes if, conversely, the volume of the compartment is properly considered. Fig. 4 reports the simulation of the system (in perfect agreement with the results presented in [8]) in the case that the division of the cell occurs once per time unit, after that (on average) the volume of the cell has doubled. In this situation the concentration of the protein $X$ follows a quite different law: upon cell division, its concentration does not considerably change (both the volume and the number of proteins are approximately halved), while during each growth cycle it follows an oscillatory regime whose local minima coincide with the cell division (i.e. with the local maxima of cell volume).

The different behaviour of the systems in Figures 3 and 4 demonstrates the error deriving from the neglect of volume variations which heavily influences, in this case, the activity of the protein $X$.

# 4 Conclusion

We have discussed the formalisation into the stochastic $\pi@$ calculus of a simple biological system involving cellular growth and division. While in perfect agreement with previous analysis of this system [8] which was accomplished by exploiting a hybrid extension of Gillespie's stochastic simulation algorithm, the approach presented here constitute a proof of concept which demonstrates how

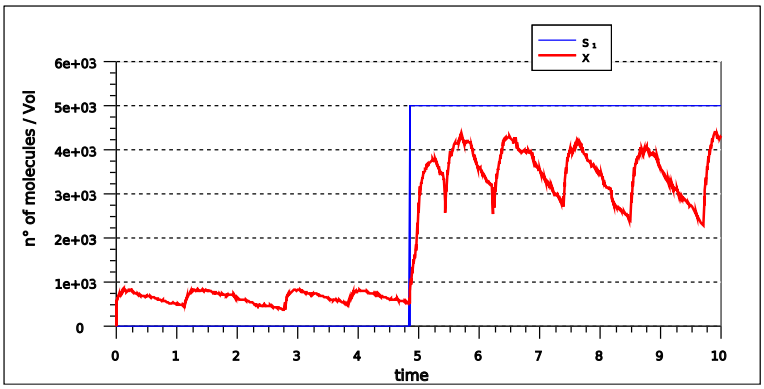


Figure 4: Level of concentration of the protein in a single cell $X$ during the process of growth and division: the division on average occurs once every one time unit, when the cell has approximately doubled in size. The concentration of protein $X$ never reaches an equilibrium, following instead an oscillatory regime.

complex biological phenomena influenced by the variation of compartment sizes can be straightforwardly modelled into this calculus in a purely stochastic manner, by relating the size of compartments to the number of molecules of each species inside them. As for the previous analysis [8], the results of stochastic simulation reported here pointed out the necessity of taking into account the variation of compartment volumes in order to provide a correct modelling of the system under examination.

# References

[1] M. Carbone and S. Maffeis. On the expressive power of polyadic synchronisation in pi-calculus. *Nord. J. Comput.*, 10(2):70–98, 2003.

[2] L. Cardelli. Brane Calculi. In Danos and Schächter [5], pages 257–278.

[3] F. Ciocchetta and J. Hillston. Bio-PEPA: a framework for the modelling and analysis of biological systems, 2008. Theoretical Computer Science.

[4] V. Danos and C. Laneve. Formal molecular biology. *Theor. Comput. Sci.*, 325(1):69–110, 2004.

[5] V. Danos and V. Schächter, editors. *Computational Methods in Systems Biology, International Conference CMSB 2004, Paris, France, May 26-28, 2004, Revised Selected Papers*, volume 3082 of *Lecture Notes in Computer Science*. Springer, 2005.

[6] D. T. Gillespie. Exact stochastic simulation of coupled chemical reactions. *J. Phys. Chem.*, 81(25):2340–2361, 1977.

[7] T. Kepler and T. Elston. Stochasticity in Transcriptional Regulation: Origins, Consequences, and Mathematical Representations. *Biophysical Journal*, 81(6):3116–3136, 2001.

[8] T. Lu, D. Volfson, L. Tsimring, and J. Hasty. Cellular growth and division in the gillespie algorithm. In *Systems Biology, IEE Proceedings*, pages 121–128, 2004.

[9] R. Milner, J. Parrow, and D. Walker. A calculus of mobile processes, I. *Inf. Comput.*, 100(1):1–40, 1992.

[10] R. Milner, J. Parrow, and D. Walker. A calculus of mobile processes, II. *Inf. Comput.*, 100(1):41–77, 1992.

[11] A. Phillips and L. Cardelli. A correct abstract machine for the stochastic pi-calculus. In *Bioconcur'04*. ENTCS, August 2004.

[12] C. Priami and P. Quaglia. Beta binders for biological interactions. In Danos and Schächter [5], pages 20–33.

[13] C. Priami, A. Regev, E. Y. Shapiro, and W. Silverman. Application of a stochastic name-passing calculus to representation and simulation of molecular processes. *Inf. Process. Lett.*, 80(1):25–31, 2001.

[14] A. Regev and E. Shapiro. Cellular abstractions: Cells as computation. *Nature*, 419(6905):343, 2002.

[15] A. Regev, W. Silverman, and E. Y. Shapiro. Representation and simulation of biochemical processes using the pi-calculus process algebra. In *Pacific Symposium on Biocomputing*, pages 459–470, 2001.

[16] C. Versari. A core calculus for a comparative analysis of bio-inspired calculi. In R. De Nicola, editor, *ESOP*, volume 4421 of *Lecture Notes in Computer Science*, pages 411–425. Springer, 2007.

[17] C. Versari. Encoding catalytic p systems in pi@. *Electr. Notes Theor. Comput. Sci.*, 171(2):171–186, 2007.

[18] C. Versari and N. Busi. Stochastic simulation of biological systems with dynamical compartment structure. In M. Calder and S. Gilmore, editors, *CMSB*, volume 4695 of *Lecture Notes in Computer Science*, pages 80–95. Springer, 2007.

[19] C. Versari and R. Gorrieri. $\pi@$: a $\pi$-based process calculus for the implementation of compartmentalised bio-inspired calculi. In M. Bernardo, P. Degano, and G. Zavattaro, editors, *SFM*, volume 5016 of *Lecture Notes in Computer Science*, pages 449–506. Springer-Verlag, 2008.