Appears in *Proceedings of CMSB 2008, Rostock, Germany*,
LNCS 5307, pp.103–122. Springer-Verlag, 2008

The Continuous π-Calculus: A Process Algebra for Biochemical Modelling

Marek Kwiatkowski and Ian Stark

Laboratory for Foundations of Computer Science
School of Informatics, The University of Edinburgh, Scotland
{M.Kwiatkowski,Ian.Stark}@ed.ac.uk

Abstract. We introduce the *continuous π-calculus*, a process algebra for modelling behaviour and variation in molecular systems. Key features of the language are: its expressive succinctness; support for diverse interaction between agents via a flexible network of molecular affinities; and operational semantics for a continuous space of processes. This compositional semantics also gives a modular way to generate conventional differential equations for system behaviour over time. We illustrate these features with a model of an existing biological system, a simple oscillatory pathway in cyanobacteria. We then discuss future research directions, in particular routes to applying the calculus in the study of evolutionary properties of biochemical pathways.

1 Introduction

This research aims to develop computational methods for studying the Darwinian evolution of biochemical pathways. We work in the framework introduced by Regev et al. [1,2,3], who identified the *π-calculus* process algebra as a promising formalism for biological modelling. We modify it in a way that allows us to mention quantitative parameters explicitly, and makes the interaction network of the agents more flexible (see §1.1 below). To take advantage of this quantitative information, we develop a novel operational semantics through a compositional description of continuous system behaviour in terms of real vector spaces (§2.2). We illustrate these concepts with an example of a concrete biological system, a simple oscillatory pathway in cyanobacteria (§7). Finally, we discuss the possibilities of answering questions related to the evolution of pathways using process-algebraic techniques such as model checking and behavioural equivalences (§4).

Reliable models and simulations of evolution on the molecular level would have wide applications, from pure evolutionary theory to drug design. Our particular interest is in the ubiquitous phenomenon of mutational robustness, which has recently received attention as a cross-level organisational principle of biological systems [4,5]. Its understanding, especially on the level of gene regulation and cellular signalling, is an important challenge; and, moreover, one where a computational approach may give essential assistance in tackling the complexity of the systems involved.

We motivate the use of process algebras in this context as follows: firstly, they have already been successfully used to model biochemical networks (see §1.2 below).

Secondly, much of the genetic variation results in qualitative or quantitative changes in the interaction network; process-algebraic descriptions of networks enable us to express this variability easily, in syntactic terms. Finally, to model evolution we need means to express fitness and related concepts such as neutrality [6,5]; to this end we plan to use well-developed process-algebraic techniques like model checking and behavioural equivalences (§4).

The main contribution of this paper is the *continuous π-calculus* ($c\pi$), a process algebra designed specifically to model biomolecular systems in an evolutionary context, with an original semantics in terms of real vector spaces. It also offers a fully modular and compositional method of generating a set of ordinary differential equations (ODEs) governing a given system. Moreover, we give a process-algebraic model of a biomolecular system of considerable interest — a primitive bacterial circadian clock recently recreated *in vitro* [7].

In the remainder of this section we introduce $c\pi$ by means of small examples (§1.1) and then very briefly recall related work in the fields of computational and theoretical biology (§1.2). Section 2 is a formal presentation of the calculus, while §3 contains the $c\pi$ model of the KaiC circadian clock [8], with graphs showing its oscillatory behaviour. In the final section we present and discuss the future directions of our research.

1.1 Key Features

The classic π-calculus is well-described in existing texts [9,10,11]. Here we focus on the key distinguishing features of our continuous π, and follow with two small examples to illustrate them.

1. Every $c\pi$ process is a parallel ($\|$) combination of *species*. Species are very similar to classic π-processes. Every species in a given process is equipped with a real number, to be thought of as the concentration of the substance described by the species. Thus, every $c\pi$ process represents a complete molecular system at a certain point in time.
2. As usual in the π-calculus, communication is through named channels. However, in contrast to most π derivatives, there are no co-names. Instead, any name can in principle communicate with any other — potentially more than one — and for any two names it is specified whether they can communicate and at what rate. Biologically, names are intended to model distinctive reaction sites, with the communication rate between two names corresponding to the rate constant of the biochemical reaction between sites. The relevant technical device to manage this information is an *affinity network*. One consequence of this approach is that every new name must come with the information about its communication potential. Also, whenever we consider a particular process P, we assume some given affinity network on the free names of P.
   The reason for this approach is two-fold. First, it makes sense in an evolutionary context to abandon the strict correspondence of sites and co-sites, and hence the symmetry of names and co-names. Second, the affinity networks give a convenient collection of parameters for varying the model: in particular, those important for questions of evolvability and robustness.

3. All $c\pi$ reactions have at most two substrates and are governed by the Law of Mass Action (following, for example, the observations of [12, p.298]). It is worth stressing, however, that more complex kinetic behaviour can emerge as we build up larger processes from smaller ones. Nonetheless, the system dynamics remains purely deterministic: for every process P we derive a term $\frac{dP}{dt}$, denoting the speed and direction of the temporal evolution of P.
4. To model spontaneous monomolecular reactions, such as degradation, or conformation changes, we use silent actions labelled with real numbers denoting reaction rate constants.
5. Molecular complexes are represented by parallel components within the shared scope of one or more private names, following Regev [2]. As usual for the π-calculus, communication between shared private names within the complex gives rise to silent actions; and these in turn model spontaneous actions like complex dissociation.

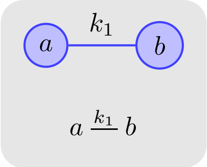


Fig. 1: A very simple affinity network and its textual rendering.

As a first simple example, consider two molecules, A and B, that can bind to each other (at rate k_1) and subsequently unbind (at rate k_2). As noted above, we model complexation as scope extrusion and decomplexation as interaction on a shared private channel. This gives rise to the following $c\pi$ definition of species A and B:

$$A \stackrel{\text{df}}{=} (\boldsymbol{\nu}\; u \overset{k_2}{-\!\!-} v)(a\langle u\rangle.v.A) \tag{1}$$

$$B \stackrel{\text{df}}{=} b(x).x.B \tag{2}$$

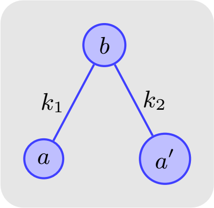
with the global affinity network of Fig. 1.

Here the public names a and b represent protein interaction sites, and a communication event between these two names models binding of these sites. The prefix $(\boldsymbol{\nu}\; u \overset{k_2}{-\!\!-} v)$ declares a new affinity network consisting of two private names, u and v, that can communicate with each other at the rate k_2. Species A and B can react by a communication event on the public $a \overset{k_1}{-\!\!-} b$ channel, with the private name u sent via a, received on b, and then substituted for x in $x.B$. This extends the scope of the network to encompass the remaining parts of A and B and so form complex C:

$$A \mid B \quad \xrightarrow{\tau\langle a,b\rangle} \quad C \stackrel{\text{df}}{=} (\boldsymbol{\nu}\; u \overset{k_2}{-\!\!-} v)(v.A \mid u.B) \tag{3}$$

If we mix species A and B together in concentrations c_1 and c_2, then we obtain a $c\pi$ process $(c_1 \cdot A \parallel c_2 \cdot B)$. The formal semantics of this process reflect mass-action dynamics: the complex C is produced in proportion to the product of substrate concentrations, while the substrates themselves (A and B) deplete similarly. If we compute the semantics (an appropriate $\frac{d\cdot}{dt}$ term) as described in §2.2, we see that this is indeed the case:

$$\frac{d(c_1 \cdot A \parallel c_2 \cdot B)}{dt} = k_1 c_1 c_2 \cdot C - k_1 c_1 c_2 \cdot A - k_1 c_1 c_2 \cdot B\,. \tag{4}$$



(a) Affinity network

$$
\begin{aligned}
A &\overset{\text{df}}{=} a.A^{+} \\
A^{+} &\overset{\text{df}}{=} a'.A^{++} + \tau@k_3.A \\
K &\overset{\text{df}}{=} b.K \\[1em]
P &= c_1 \cdot A \;\parallel\; c_2 \cdot K
\end{aligned}
$$

(b) Species and process definitions

Fig. 2: A simple $c\pi$ system with a non-trivial affinity network modelling discriminative binding of the kinase K (via site b) to the molecules A and A^{+} (at sites a and a', respectively). The definition of the inactive A^{++} species is omitted.

Similarly, when we consider a solution of complexes in concentration c_3 and derive the semantics of the process $(c_3 \cdot C)$, we observe that the complex dissolves to give back substrates A and B at the expected rate:

$$\frac{d(c_3 \cdot C)}{dt} \;=\; -k_2c_3 \cdot C + k_2c_3 \cdot A + k_2c_3 \cdot B\ . \tag{5}$$

As another example, consider a molecule A that can exists in three states: unphosphorylated, phosphorylated and doubly phosphorylated; denoted A, A^{+} and A^{++} respectively. Furthermore, suppose that kinase K (the phosphorylating agent) is more effective at the initial phosphorylation step $A \longrightarrow A^{+}$ than at the subsequent one $A^{+} \longrightarrow A^{++}$, having reaction rate constants $k_1 > k_2$. Finally, assume that A^{+} (only) can spontaneously relax back to A at a rate k_3. Figure 2 shows a $c\pi$ model for this system, and in particular a process P representing an initial state where only A and K are present, at concentrations c_1 and c_2 respectively.

The affinity network in Fig. 2(a) indicates that site b can react with either site a or site a': this will capture the double action of the kinase. Figure 2(b) gives the definitions of the species involved in the system. The first equation states that species A can be transformed into A^{+} on interaction at the site a. The second states that A^{+} can either interact on a' and be transformed into A^{++} or convert back to A at rate k_3 without any external agent — here "+" models the mutually exclusive choice of alternatives. In the third equation, kinase K can interact at site b and then return to its initial state; recall that according to the affinity network, this interaction might be with site a (on A) or a' (on A^{+}). The final line defines the initial state of the system, with A and K present at the specified concentrations.

This is a dynamic model, with P evolving in a continuous fashion. At any time instant we can, using the methods of §2.2, formally derive the vector $\frac{dP}{dt}$ specifying the gradient of this temporal evolution. In particular, in the initial state we have:

$$\frac{dP}{dt} \;=\; k_1c_1c_2 \cdot A^{+} - k_1c_1c_2 \cdot A\ . \tag{6}$$

1.2 Related Work

The application of the π-calculus to biology is due to Regev and Shapiro [3], who identified the fundamental correspondence between cellular processes and concurrent computations [13]. They proposed modelling molecules as π-processes, and the use of parallel composition to express the fact that such molecules act independently. In this framework, names denote molecular interaction sites and communication models interaction. A further important abstraction was the use of private names to model molecular complexes and compartments.

Further refinements of this framework addressed the introduction of quantitative information into the model and on modelling compartments more directly. This led, for example, to the development of the biochemical stochastic π-calculus [1] and BioAmbients [14]. Other process algebras, such as PEPA [15] have also been applied to model biological systems [16]. Although seen mainly as simulation engines, these formalisms have also been used to perform static analysis of the model [17].

Using process algebras as a high-level descriptive language, Calder et al. [18] have shown how PEPA models can generate both discrete (stochastic) and continuous (ODE) behavioural specifications for the same system. Unlike raw ODEs then, a process algebra model is not itself the behaviour, but can be used to generate it. We believe that this abstraction step is important in modelling variation, to identify how emergent behaviour depends on changes in a process or its parameters.

Meanwhile, the interest in mutational robustness has been growing amongst biologists for the past 15 years. A recent monograph on the subject [5] identifies the explosion of high-throughput techniques as an important factor for this interest; the other is the importance of this concept in the context of systems biology [19]. The methods applied to study this phenomenon range from pure mathematics [20] to exhaustive computations [21].

2 The Continuous π-Calculus

In this section we set out a formal syntax and mathematical semantics for the continuous π-calculus. Both syntax and semantics have two “layers”: of *species* corresponding to individual molecules, and *processes* to populations of these. It is important to keep in mind, however, that none of these terms should invoke associations with their meaning in the context of evolutionary theory.

2.1 Syntax

Definition 1. Take $\mathcal{N}$ a fixed, countably infinite set of *names*, denoted by lower-case letters a, b, x, y, ... Vectors of names are denoted by $\vec{a}$, $\vec{x}$ etc.; these may be of zero length.

Definition 2. A *prefix* is a syntactic expression of the form $a(\vec{b}; \vec{y})$ (a *communication prefix*) or $\tau@k$ (*spontaneous* or *silent prefix*), where all elements of $\vec{y}$ are distinct, $\tau \notin \mathcal{N}$ is a fixed symbol and $k \in \mathbb{R}_{\geq 0}$. We use symbols like π, π', etc. to denote prefixes.

A communication prefix models the ability of one molecule to engage in an interaction with another, at site a. Details of the interaction are modelled as name-passing, where $\vec{b}$ is the vector of names to be sent and $\vec{y}$ is a vector of placeholders for names to be received (and so binds subsequent occurrences of the $\vec{y}$). This symmetry and synchrony of communication is mildly novel, and introduced to reflect the fact that molecular interaction is a synchronous and (usually) symmetric event. For readability we abbreviate when possible: $a(\vec{y})$ for $a(\,;\vec{y})$, $a\langle\vec{b}\rangle$ for $a(\vec{b};\,)$, and a for $a(\,;\,)$.

The silent prefix $\tau@k$ models the case where a molecule may undergo spontaneous change, without any interaction with the external environment; or, at least, no observed interaction at the level of abstraction being modelled (cf. the dephosphorylation of species A^+ in Fig. 2(b)). The rate of this transformation is recorded directly in the prefix as k.

Definition 3. An *affinity network* is a pair $\langle M, f\rangle$, where $M \subseteq_{\text{fin}} \mathcal{N}$ is a carrier set and f is a symmetric partial function $f\colon M \times M \rightharpoonup \mathbb{R}_{\geq 0}$. We often blur the distinction between a network and its carrier: for a network M we write $x \in M$ to indicate that x lies in the carrier of M, and similarly for other set-theoretic predicates. The expression $M(a,b)$ denotes the value the network assigns to a pair of names (a,b) if this is defined; we write $M(a,b)\downarrow$ when this is the case.

From here on we assume a distinguished *global affinity network* Aff, which must be a total relation on its carrier. We shall also use the notation $X \# Y$, read X *fresh for* Y, to state that $X \cap Y = \varnothing$ for name sets X and Y; and then overload this when X or Y are terms to refer to their free name sets.

Definition 4. The set of *species* is generated by the following grammar:

$$A, B ::= \mathbf{0} \;\mid\; D(\vec{a}) \;\mid\; \Sigma_{i=0}^{n}\pi_i.A_i \;\mid\; A \mid B \;\mid\; (\boldsymbol{\nu} M)A \qquad (7)$$

where $M \# \mathit{Aff}$. These are in turn the inactive species $\mathbf{0}$, invocation of a species definition, guarded choice, parallel composition, and local name declaration. This last is also *restriction*: names declared in the affinity network M are available in A, but not elsewhere until explicitly passed out in a communication from A. Small instances of guarded choice Σ are usually written with $+$; as in Fig. 2(b). For every invocation $D(\vec{a})$ we assume a corresponding definition $D(\vec{y}) \overset{\text{df}}{=} A$, such that every free name of A appears in either $\vec{y}$ or Aff. Furthermore, these definitions must be *productive* in that any recursive cycle includes a prefix guard. We identify α-equivalent species: the binding operations are prefix π and restriction $(\boldsymbol{\nu} M)$. Finally, there is a *structural congruence* $\equiv$ on species, generated by the axioms in Fig. 3. We use $\mathcal{S}$ to denote the set of species modulo $\equiv$, and write $[A]$ for the equivalence class of species A up to structural congruence.

Definition 5. The set of *processes* is generated by the following grammar:

$$P, Q ::= c \cdot A \;\mid\; P \parallel Q \qquad (8)$$

where A is a species, $c \in \mathbb{R}_{\geq 0}$ and all free names appear in Aff. Figure 3 gives a structural congruence $\equiv$ on processes.

$$\begin{aligned}
A \mid \mathbf{0} &\equiv A \\
A \mid B &\equiv B \mid A \\
(A \mid B) \mid C &\equiv A \mid (B \mid C) \\
\Sigma_{i=0}^{n} \pi_i.A_i &\equiv \Sigma_{i=0}^{n} \pi_{\sigma_i}.A_{\sigma_i} && \text{perm. } \sigma \\
(\boldsymbol{\nu} M)(A \mid B) &\equiv A \mid (\boldsymbol{\nu} M)B && M \# A \\
(\boldsymbol{\nu} M)A &\equiv A && M \# A \\
(\boldsymbol{\nu} M)(\boldsymbol{\nu} N)A &\equiv (\boldsymbol{\nu} N)(\boldsymbol{\nu} M)A && M \# N
\end{aligned}$$

$$\begin{aligned}
P \parallel (c \cdot \mathbf{0}) &\equiv P \\
P \parallel Q &\equiv Q \parallel P \\
P \parallel (Q \parallel R) &\equiv (P \parallel Q) \parallel R \\
(c \cdot A) \parallel (d \cdot A) &\equiv (c + d) \cdot A \\
c \cdot A &\equiv c \cdot B && A \equiv B
\end{aligned}$$

Fig. 3: Axioms generating structural congruence on species (l) and processes (r).

In what follows we maintain a careful distinction between a species and its $\equiv$-equivalence class, and the same for processes, in order to precisely state the correctness results for $c\pi$ semantics.

2.2 Semantics

The two-layered nature of syntax (a “layer” of species, then another of processes) is reflected in the semantics: to obtain the semantics of a process, we first examine the species involved. Formally, we do this by defining a multi-transition system on species and then using this information to build the continuous semantics for processes. Both levels maintain compositionality.

The transition system for species A *multi-transition system* is a labelled transition system that allows multiple transitions with the same source, target and label. This extension is necessary to keep track of quantitative aspects of behaviour in $c\pi$, as done for example in PEPA [15].

We present the multi-transition system for species in an *abstraction-concretion* style, following Milner [9] (or see [11] for a shorter explanation). In continuous π the use of symmetric prefixes eliminates the distinction between abstractions and concretions, and hence we use “concretion” to refer to both constructions.

Definition 6. A *concretion* is a term generated by the following grammar:

$$F, G ::= (\vec{b};\vec{y})A \;\mid\; F \,|\, A \;\mid\; A \,|\, F \;\mid\; (\boldsymbol{\nu} M)F \tag{9}$$

where all elements of $\vec{y}$ are distinct and binding. As with species, we identify α-equivalent concretions. We build a structural congruence $\equiv$ from the axioms in Fig. 4, writing $[F]$ for the $\equiv$-equivalence class of F, and take $\mathcal{C}$ as the set of concretions modulo structural congruence.

A concretion can be seen as a species that has committed to take part in a specific binary interaction. When it encounters another compatible concretion that interaction may take place. We formalize this with a notion of “pseudo-application”.

$$\begin{array}{rcrr}
F \mid \mathbf{0} \equiv F & & & \\
F \mid A \equiv A \mid F & & (\boldsymbol{\nu} M)(A \mid F) \equiv A \mid (\boldsymbol{\nu} M)F & M \# A \\
(F \mid A) \mid B \equiv F \mid (A \mid B) & & (\boldsymbol{\nu} M)(F \mid A) \equiv F \mid (\boldsymbol{\nu} M)A & M \# F \\
(A \mid F) \mid B \equiv A \mid (F \mid B) & & (\boldsymbol{\nu} M)F \equiv F & M \# F \\
F \mid A \equiv F \mid B & A \equiv B & (\boldsymbol{\nu} M)(\boldsymbol{\nu} N)F \equiv (\boldsymbol{\nu} N)(\boldsymbol{\nu} M)F & M \# N \\
(\vec{b}; \vec{y})A \equiv (\vec{b}; \vec{y})B & A \equiv B & (\vec{b}; \vec{y})(A \mid B) \equiv A \mid (\vec{b}; \vec{y})B & \vec{y} \# A
\end{array}$$

Fig. 4: Axioms generating structural congruence for concretions.

Definition 7. The operation of *pseudo-application* is a binary partial function $\circ$ on concretions, defined by structural induction over its arguments. For the base case, $(\vec{a}; \vec{x})A \circ (\vec{b}; \vec{y})B$ is defined if and only if $|\vec{a}| = |\vec{y}|$ and $|\vec{b}| = |\vec{x}|$, in which case the result is $A\{\vec{b}/\vec{x}\} \mid B\{\vec{a}/\vec{y}\}$. The inductive clauses are as follows:

$$\begin{array}{ll}
(\vec{a}; \vec{x})A \circ (F \mid B) \overset{\text{df}}{=} ((\vec{a}; \vec{x})A \circ F) \mid B & (A \mid F) \circ F' \overset{\text{df}}{=} A \mid (F \circ F') \\
(\vec{a}; \vec{x})A \circ (B \mid F) \overset{\text{df}}{=} B \mid ((\vec{a}; \vec{x})A \circ F) & (F \mid A) \circ F' \overset{\text{df}}{=} (F \circ F') \mid A \\
(\vec{a}; \vec{x})A \circ (\boldsymbol{\nu} M)F \overset{\text{df}}{=} (\boldsymbol{\nu} M)((\vec{a}; \vec{x})A \circ F) & (\boldsymbol{\nu} M)(F) \circ F' \overset{\text{df}}{=} (\boldsymbol{\nu} M)(F \circ F') \, .
\end{array}$$

For the two clauses in the bottom line we assume that M is fresh for the other concretion involved. Observe that in the presence of α-equivalence this condition can always be met, and hence the only reason for a pseudo-application to be undefined is the arity mismatch of the concretions in the base case.

Where pseudo-application is defined we write $F \circ G \downarrow$ and say that F and G are *compatible*. It is possible to define a type system for sites to ensure compatibility, but this complicates the calculus and we shall not do so here.

Proposition 8. *The following hold for any compatible concretions F and G.*

(i) *Application $F \circ G$ is a species.*
(ii) *Application $G \circ F$ is defined and $G \circ F \equiv F \circ G$.*
(iii) *If $F' \equiv F$ and $G' \equiv G$ then $F' \circ G'$ exists and is congruent to $F \circ G$.*

Proof. Induction over the derivation of $F \circ G$ shows (i) and (ii), while (iii) uses induction over the derivations of $F' \equiv F$ and $G' \equiv G$. □

Finally, Fig. 5 sets out the rules for generating the multi-transition system on species, as a structural operational semantics [22]. For a species A, we write $\mathit{Trans}(A)$ for the associated transition multiset. These transitions are of three kinds:

1. From a species to a concretion, labelled by a name. This represents a potential for interaction; more precisely, a transition $A \xrightarrow{a} (\vec{b}; \vec{y})B$ means that the species A can interact with another by sending $\vec{b}$ on the channel a and then evolve to B, with $\vec{y}$ replaced by data received.

$$\frac{0 \le j \le n \quad \pi_j = a_j(\vec{b}_j;\vec{y}_j)}{\Sigma_{i=0}^{n}\pi_i.A_i \xrightarrow{a_j} (\vec{b}_j;\vec{y}_j)A_j}\ \textsc{Choice-1} \qquad \frac{A \xrightarrow{\alpha} E}{A \mid B \xrightarrow{\alpha} E \mid B}\ \textsc{Par-Left}$$

$$\frac{0 \le j \le n \quad \pi_j = \tau@k}{\Sigma_{i=0}^{n}\pi_i.A_i \xrightarrow{\tau@k} A_j}\ \textsc{Choice-2} \qquad \frac{B \xrightarrow{\alpha} E}{A \mid B \xrightarrow{\alpha} A \mid E}\ \textsc{Par-Right}$$

$$\frac{A \xrightarrow{a} F \quad B \xrightarrow{b} G \quad F \circ G{\downarrow}}{A \mid B \xrightarrow{\tau\langle a,b\rangle} F \circ G}\ \textsc{Com-1} \qquad \frac{A \xrightarrow{\alpha} E \quad \alpha \notin M}{(\boldsymbol{\nu} M)A \xrightarrow{\alpha} (\boldsymbol{\nu} M)E}\ \textsc{Res-1}$$

$$\frac{A \xrightarrow{\tau\langle a,b\rangle} B \quad a,b \in M \quad M(a,b){\downarrow}}{(\boldsymbol{\nu} M)A \xrightarrow{\tau@M(a,b)} (\boldsymbol{\nu} M)B}\ \textsc{Com-2} \qquad \frac{A \xrightarrow{\tau\langle a,b\rangle} E \quad a,b \notin M}{(\boldsymbol{\nu} M)A \xrightarrow{\tau\langle a,b\rangle} (\boldsymbol{\nu} M)E}\ \textsc{Res-2}$$

$$\frac{B \xrightarrow{\alpha} E \quad D(\vec{y}) \overset{\text{df}}{=} B}{D(\vec{b}) \xrightarrow{\alpha} E\{\vec{b}/\vec{y}\}}\ \textsc{Defn}$$

Fig. 5: Transition rules for species. Here α ranges over all kinds of transition labels and E ranges over both species and concretions.

2. From a species to another species, labelled by $\tau@k$ where k is a real number, such as $A \xrightarrow{\tau@1.5} B$. This denotes the ability of species A to evolve into B without other interaction, at a basal rate of k (here 1.5). Examples include degradation, where B is $\mathbf{0}$, or complex dissociation, with B of the form $B' \mid B''$.
3. From one species to another, labelled by a term $\tau\langle a,b\rangle$ with names a and b, for example $A \xrightarrow{\tau\langle a,b\rangle} B$. This also denotes a potential for evolution of A into B, but now the basal rate of evolution is the affinity between a and b. This affinity will be determined by either the global affinity network Aff or some local network M to be introduced by restriction $(\boldsymbol{\nu} M)$. Thus local interaction at private names becomes visible externally as a spontaneous action, maintaining the compositionality of the semantics.

The following result states that the structural congruence of species is indeed a behavioural equivalence.

Theorem 9. *Let $A \equiv B$. There exists a bijection ϕ: $\mathit{Trans}(A) \to \mathit{Trans}(B)$ such that if $\phi(A \xrightarrow{\alpha} E) = B \xrightarrow{\alpha'} E'$ then $\alpha = \alpha'$ and $E \equiv E'$.*

Proof (sketch). We proceed by induction on the derivation of $A \equiv B$: for every transition in $\mathit{Trans}(A)$ we exhibit a corresponding one in $\mathit{Trans}(B)$ via a case analysis of the transition derivation tree, and then show that this association is a bijection. □

The behaviour of processes We give a compositional semantics to $c\pi$ processes in terms of real vector spaces $\mathbb{P}$ and $\mathbb{D}$, capturing respectively the immediate actions $\frac{dP}{dt}$ and the potential interactions ∂P of a process P. First, however, we need several preliminary definitions. Recall that $\mathcal{S}$ and $\mathcal{C}$ are the sets of species and concretions, respectively, modulo structural congruence.

Definition 10. A non-zero $[A] \in \mathcal{S}$ is *prime* if it is not a parallel composition of non-trivial species, i.e. if $A \equiv (B \mid C)$ implies either $B \equiv \mathbf{0}$ or $C \equiv \mathbf{0}$. We write $\mathcal{S}^{\#}$ for the set of prime species, with $\mathcal{S}^{\#} \subsetneq \mathcal{S}$.

Theorem 11. *For any nonzero species A there exists a unique finite multiset $\{\![[A_1],\dots,[A_n]]\!\} \subset \mathcal{S}^{\#}$ such that $A \equiv A_1 \mid \cdots \mid A_n$. Call this* prime decomposition *of A.*

Proof (sketch). We assign normal forms to species using a normalising and confluent term rewriting system respecting $\equiv$, and take the prime decomposition as the multiset of $|$-components of the normal form. □

The decomposition theorem allows us to represent any $c\pi$ process as a collection of prime species weighted by real numbers.

Definition 12. The *process space* $\mathbb{P}$ is the (infinite dimensional) vector space $\mathbb{R}^{(\mathcal{S}^{\#})}$.

There is a natural mapping from species into process space, with the concentration of participating prime species matching their multiplicity in the prime decomposition.

Definition 13. Define $\langle\cdot\rangle : \mathcal{S} \to \mathbb{P}$ inductively over the structure of its argument:

$$\langle A \rangle \stackrel{\text{df}}{=} \begin{cases} 0 \text{ at every position} & \text{if } [A] = [\mathbf{0}] \\ 1 \text{ at } [A], 0 \text{ elsewhere} & \text{if } [A] \text{ prime} \\ \langle B \rangle + \langle C \rangle & \text{if } [A] = [B \mid C] \text{ for } B, C \not\equiv \mathbf{0}. \end{cases} \tag{10}$$

It follows from Thm. 11 that $\langle\cdot\rangle$ is well-defined and constant inside every equivalence class of species.

In due course we shall use space $\mathbb{P}$ to capture immediate process behaviour. Although this behaviour is what we are most interested in, it is impossible to define it compositionally without further information on the possible ways a process may interact with others. We therefore define a space of interaction potentials $\mathbb{D}$ and an interaction tensor $\oplus\!: \mathbb{D} \times \mathbb{D} \to \mathbb{P}$ that combines two compatible potentials into an immediate behaviour.

Definition 14. The *interaction space* $\mathbb{D}$ is the (infinite dimensional) vector space $\mathbb{R}^{\mathcal{S}\times\mathcal{C}\times\mathcal{N}}$ of ternary real functions with pointwise addition and scalar (real) multiplication. Note that it has a basis consisting of functions of the form $\mathbf{1}_{[A],[F],a}$ which take the value 1 for the indicated arguments and 0 for any other set of inputs.

Definition 15. The *interaction tensor* $\oplus\!: \mathbb{D} \times \mathbb{D} \to \mathbb{P}$ is the bilinear function defined by the following action on basis values:

$$\mathbf{1}_{[A],[F],a} \oplus \mathbf{1}_{[B],[G],b} \stackrel{\text{df}}{=} \begin{cases} \mathit{Aff}(a,b)(\langle F \circ G \rangle - \langle A \rangle - \langle B \rangle) & \text{if } F \circ G\!\downarrow \\ 0 & \text{otherwise .} \end{cases} \tag{11}$$

Each element $\xi \in \mathbb{D}$ associates with every triple $([A],[F],a)$ a real number. When ξ describes the interaction capabilities of a process, this real number denotes the sum of concentrations of all species present in the process which can make the transition

$A \xrightarrow{a} F$, or structural equivalent. The interaction tensor $\obar$ combines two such interaction potentials into an actual process behaviour. Combination of basis elements, which can be seen as the two transitions $A \xrightarrow{a} F$ and $B \xrightarrow{b} G$, gives a new species resulting from the interaction $F \circ G$, the substrate species A and B are lost, and the whole expression is weighted by the interaction rate of the a — b channel in Aff. Combination of more complex $\xi, \xi' \in \mathbb{D}$ is computed by bilinear extension of this, which ensures that all combinations of interaction potentials in ξ and ξ' are considered. Moreover, bilinearity means that the result is scaled in proportion to the "amounts" of interaction potential, and so reflects the Law of Mass Action.

With the above definitions at hand, we are now in a position to define the formal semantics of $c\pi$.

Definition 16. The *complete behaviour* of a process P is a pair $(\frac{dP}{dt}, \partial P) \in \mathbb{P} \times \mathbb{D}$ of its *immediate actions* and *potential interactions* defined inductively on the structure of P as follows:

$$\partial(c \cdot A)([B],[F],a) \overset{\text{df}}{=} c \cdot \text{card}\{\!|\, C \xrightarrow{a} G \in \mathit{Trans}(A) \mid C \in [B] \wedge G \in [F] \,|\!\} \tag{12}$$

$$\begin{aligned} \frac{d(c \cdot A)}{dt} &\overset{\text{df}}{=} c \cdot \Sigma_{B \xrightarrow{\tau @ k} C \in \mathit{Trans}(A)} k \cdot (\langle C \rangle - \langle B \rangle) \\ &+ c \cdot \Sigma_{B \xrightarrow{\tau\langle a,b\rangle} C \in \mathit{Trans}(A)} \mathit{Aff}(a,b) \cdot (\langle C \rangle - \langle B \rangle) \\ &+ \frac{1}{2}(\partial(c \cdot A) \obar \partial(c \cdot A)) \end{aligned} \tag{13}$$

$$\partial(P \parallel Q) \overset{\text{df}}{=} \partial P + \partial Q \tag{14}$$

$$\frac{d(P \parallel Q)}{dt} \overset{\text{df}}{=} \frac{dP}{dt} + \frac{dQ}{dt} + \partial P \obar \partial Q\ . \tag{15}$$

We explain briefly the intuitions behind these definitions. Because every process can be identified with a point in $\mathbb{P}$, we can view the immediate behaviour $\frac{dP}{dt}$ as a vector field over $\mathbb{P}$, associating with each process the gradient of its temporal evolution. The equations (13) and (15) reflect this interpretation. Thus in (13) we take into account the effect of all τ-transitions of species A, weighted with the interaction rates (given by the transition labels or the global affinity network) and the initial concentration c, and then add the behaviour arising from interactions between pairs of A molecules. In (15), the immediate behaviour of a composition of two processes is the sum of immediate behaviours of the components plus the behaviour that emerges from their interaction.

Computing the interaction potential ∂P for a process is more straightforward: equation (12) lifts all the appropriate transitions from the multi-transition system and multiplies them by the concentration c; while (14) reflects the fact that the interaction potential of a composition of processes is simply the sum of the interaction potentials of the components, with no cancellation or further emergent interaction.

The following theorems demonstrate that structural congruence of processes is a behavioural equivalence, and that further identification of $|$ and $\|$ only slightly weakens this.

Theorem 17. *Let $P \equiv Q$. Then $\partial P = \partial Q$ in $\mathbb{D}$ and $\frac{dP}{dt} = \frac{dQ}{dt}$ in $\mathbb{P}$.*

Proof. Straightforward induction on derivation of $P \equiv Q$. □

Theorem 18. *Let $\equiv^+$ be the smallest congruence on processes containing $\equiv$ and satisfying the additional rule*

$$c \cdot (A \mid B) \ \equiv^+ \ (c \cdot A) \ \| \ (c \cdot B) \tag{16}$$

and let $P \equiv^+ Q$. Then $\frac{dP}{dt} = \frac{dQ}{dt}$ and for any $\xi \in \mathbb{D}$, $\partial P \obar \xi = \partial Q \obar \xi$.

Proof. By induction on derivation of $P \equiv^+ Q$. □

In general we may have $P \equiv^+ Q$ but $\partial P \neq \partial Q$, because the transitions $A \xrightarrow{x} F$ and $A \mid B \xrightarrow{x} F \mid B$ give rise to different points of $\mathbb{D}$ via (12), despite being essentially equivalent as interaction potentials. We discuss this further in §4.4, with a possible remedy. Notice, though, that the property we do have of equality under $- \obar \xi$ for all ξ is a form of *observational equivalence*: there is no way from within process space to observe any difference between ∂P and ∂Q.

3 Example

In this section we give a $c\pi$ model for a simple biomolecular system, the KaiC circadian clock. Our reference for this system is the work of van Zon et al. [8].

3.1 The system

Introduction Circadian clocks are molecular systems that exhibit oscillatory behaviour synchronized with the 24-hour day cycle. They play an important role in many organisms by helping to regulate their cellular behaviour according to the circadian rhythm.

The system we model is a primitive circadian clock found in the cyanobacterium *Synechococcus elongatus* [23,24]. It consists of three kinds of protein: KaiA, KaiB and KaiC. In particular, KaiC forms hexamers with 6 phosphorylation sites which are phosphorylated and dephosphorylated in a cyclic manner, thus dictating the circadian rhythm.

The KaiC circadian clock has two features that make it of particular interest to the biological community. The first is its simplicity — it requires only 3 kinds of molecules to function. The other is that it does not rely on either intracellular compartments or gene regulation, which sets it apart from other circadian clocks and (remarkably) makes it possible to reproduce its self-sustaining cycle relatively easily *in vitro* [7].

The allosteric model Although there are extensive experimental results on the activity of the various components of the KaiC system, its precise mechanism is not yet understood. In order to explain the observed behaviour, the authors of [8] propose an elegant model based on two assumptions. The first assumption is that every KaiC protein is *allosteric*, i.e. it can adopt two distinct 3D shapes (conformations), denoted active and inactive. This gives every KaiC hexamer a propensity to spontaneously undergo a phosphorylation-dephosphorylation cycle, as shown in Fig. 6. The cycles of

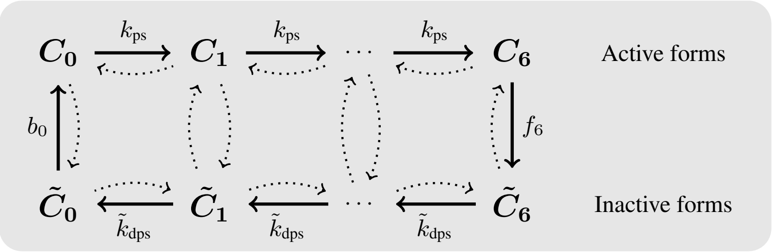


Fig. 6: The phosphorylation cycle of a single KaiC molecule. The two allosteric forms have opposite (de-)phosphorylation tendencies: phosphorylation proceeds from left to right, dephosphorylation from right to left. The potential to flip between the conformations closes the cycle. Non-dominant reactions are indicated with dotted arrows. Adapted from [8, Fig. 1(B)].

individual KaiC are then synchronized with each other thanks to the other assumption: that the phosphorylating agent KaiA binds more strongly to weakly phosphorylated KaiC molecules. This mechanism is called *differential affinity*. The role of KaiB in this model is to stabilise the inactive form of KaiC and to increase the competition for free KaiA molecules between the differently phosphorylated active forms.

3.2 The $c\pi$ model

Figure 7 displays our $c\pi$ model of the KaiC system proposed in [8]. It combines species declarations, a global affinity network Aff and local affinity networks M_i.

Species In the first few lines (17)–(22) we define the 14 distinct species of KaiC: active and inactive forms, each in one of 7 phosphorylation states. Lines (23)–(28) then define the complexes formed by inactive KaiC with KaiB and KaiA, while (29) defines species KaiA and KaiB themselves. Finally, process (30) describes an initial state of the model. Note that several intermediate species, not explicitly defined here, naturally emerge from subsequent interactions. For example, the A-C_0 complex

$$(\boldsymbol{\nu} M_0)((u_0.C_0 + r_0.C_1) \mid act_0.A) \tag{31}$$

arising from an interaction between A and C_0 on the $a \overset{k_0^{\mathrm{Af}}}{\text{—}} a_0$ channel, communicating local names u_0 and r_0. This complex can then dissociate, triggered by the interaction between local site act_0 and either u_0 or r_0, corresponding respectively to simple unbinding of C_0 or catalysed phosphorylation to C_1.

The definitions of the species are based entirely on their interaction capabilities as postulated in [8]. For example, the active and unphosphorylated KaiC molecule (species C_0 in (17)) can either flip to the inactive state at rate f_0 (the $\tau@f_0.\tilde{C}_0$ component), spontaneously phosphorylate at rate k_{ps} (the $\tau@k_{\mathrm{ps}}.C_1$ component), or bind a KaiA molecule to form the complex of (31) above (the $a_0\langle act_0\rangle.(u.C_0 + r.C_1)$ component).

We have one minor deviation from [8], concerning the binding of KaiB and KaiA to inactivated KaiC in lines (23)–(28). Both of these bind in multiples to KaiC: based on

$$C_0 \overset{\text{df}}{=} (\boldsymbol{\nu} M_0)(\tau@f_0.\tilde{C}_0 + \tau@k_{\text{ps}}.C_1 + a_0\langle act_0\rangle.(u_0.C_0 + r_0.C_1)) \tag{17}$$

$$\begin{aligned} C_i \overset{\text{df}}{=} {} & (\boldsymbol{\nu} M_i)(\tau@f_i.\tilde{C}_i + \tau@k_{\text{ps}}.C_{i+1} \\ & + \tau@k_{\text{dps}}.C_{i-1} + a_i\langle act_i\rangle.(u_i.C_i + r_i.C_{i+1})) \end{aligned} \tag{18}$$

$$C_6 \overset{\text{df}}{=} \tau@f_6.\tilde{C}_6 + \tau@k_{\text{dps}}.C_5 \tag{19}$$

$$\tilde{C}_0 \overset{\text{df}}{=} \tau@b_0.C_0 + \tau@\tilde{k}_{\text{ps}}.\tilde{C}_1 + b_0.b'.B\tilde{C}_0 \tag{20}$$

$$\tilde{C}_i \overset{\text{df}}{=} \tau@b_i.C_i + \tau@\tilde{k}_{\text{ps}}.\tilde{C}_{i+1} + \tau@\tilde{k}_{\text{dps}}.\tilde{C}_{i-1} + b_i.b'.B\tilde{C}_i \tag{21}$$

$$\tilde{C}_6 \overset{\text{df}}{=} \tau@b_6.C_6 + \tau@\tilde{k}_{\text{dps}}.\tilde{C}_5 + b_6.b'.B\tilde{C}_6 \tag{22}$$

$$B\tilde{C}_0 \overset{\text{df}}{=} \tau@k_0^{\text{Bb}}.(\tilde{C}_0 \mid B \mid B) + \tau@\tilde{k}_{\text{ps}}.B\tilde{C}_1 + \tilde{a}_0.\tilde{a}'.AB\tilde{C}_0 \tag{23}$$

$$\begin{aligned} B\tilde{C}_i \overset{\text{df}}{=} {} & \tau@k_i^{\text{Bb}}.(\tilde{C}_i \mid B \mid B) + \tau@\tilde{k}_{\text{ps}}.B\tilde{C}_{i+1} \\ & + \tau@\tilde{k}_{\text{dps}}.B\tilde{C}_{i-1} + \tilde{a}_i.\tilde{a}'.AB\tilde{C}_i \end{aligned} \tag{24}$$

$$B\tilde{C}_6 \overset{\text{df}}{=} \tau@k_6^{\text{Bb}}.(\tilde{C}_6 \mid B \mid B) + \tau@\tilde{k}_{\text{dps}}.B\tilde{C}_5 + \tilde{a}_6.\tilde{a}'.AB\tilde{C}_6 \tag{25}$$

$$AB\tilde{C}_0 \overset{\text{df}}{=} \tau@\tilde{k}_0^{\text{Ab}}.(B\tilde{C}_0 \mid A \mid A) + \tau@\tilde{k}_{\text{ps}}.AB\tilde{C}_1 \tag{26}$$

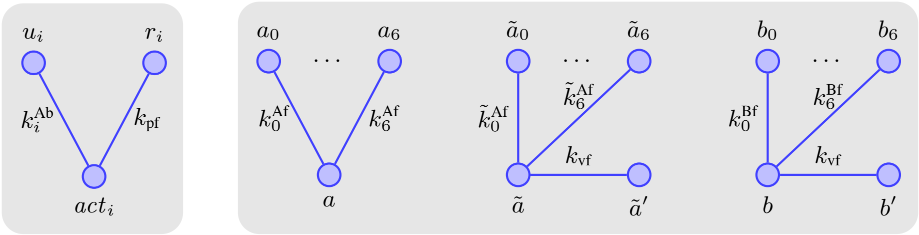
$$AB\tilde{C}_i \overset{\text{df}}{=} \tau@\tilde{k}_i^{\text{Ab}}.(B\tilde{C}_i \mid A \mid A) + \tau@\tilde{k}_{\text{ps}}.AB\tilde{C}_{i+1} + \tau@\tilde{k}_{\text{dps}}.AB\tilde{C}_{i-1} \tag{27}$$

$$AB\tilde{C}_6 \overset{\text{df}}{=} \tau@\tilde{k}_6^{\text{Ab}}.(B\tilde{C}_6 \mid A \mid A) + \tau@\tilde{k}_{\text{dps}}.AB\tilde{C}_5 \tag{28}$$

$$A \overset{\text{df}}{=} a(x).x.A + \tilde{a}.\mathbf{0} \qquad B \overset{\text{df}}{=} b.\mathbf{0} \tag{29}$$

$$P \overset{\text{df}}{=} c_A \cdot A \parallel c_B \cdot B \parallel c_C \cdot C_0 \tag{30}$$

(a) Species and process definitions



(b) Local affinity networks M_i

(c) The global affinity network *Aff*

Fig. 7: The $c\pi$ model of the KaiC circadian clock. Parameter i takes values $1\dots5$ in species definitions and $0\dots6$ in the affinity networks.

size measurements, van Zon et al. assume that each KaiC binds two KaiB and then two KaiA. They model these with a 3-substrate reaction, which cannot be expressed directly in $c\pi$. Instead, we model the binding by two consecutive binary interactions, where the rate of the second (k_{vf}) is much greater than that of the first (k_i^{Bf} or k_i^{Af}).

This occurs, for example, in the KaiB-KaiC complex $B\tilde{C}_0$ of (23). This can either spontaneously dissociate, at rate k_0^{Bb}, into an inactivated KaiC and two KaiB, phosphorylate at rate $\tilde{k}_{\mathrm{ps}}$, or bind successively to the $\tilde{a}$ site on two KaiA molecules to form a KaiA-KaiB-KaiC complex.

The model also shows other kinetics in action, for example in the binding of KaiA to active KaiC modelled by scope extrusion shown above (31). When the extrusion is reversible, as it is here, the $c\pi$ semantics of the combined reaction generates a Michaelis-Menten kinetics. Other binding events are modelled as simple communication, which gives rise to Mass Action kinetics.

Affinity networks We model the differential affinity mechanism with fan-like affinity networks, where a single site can interact with several others at different rates (Fig. 7(c)). For the sake of symmetry and for the ease of potential perturbation analysis of the model, we retain the fan shape even where [8] assumes no differential affinity. The differential affinity of the KaiA-KaiC binding is modelled by differing interaction rates k_i^{Ab} in the collection of local affinity networks M_i (Fig. 7(b)).

Results There is a close correspondence between the dynamical behaviour of the $c\pi$ model, as generated by the semantics of §2.2, and that of our reference paper [8]. Moreover, we can extract from the species and network declarations of Fig. 7 a set of ODEs that matches those reported in [25, p.14] (up to minor differences due to our alternative modelling of multiple binding).

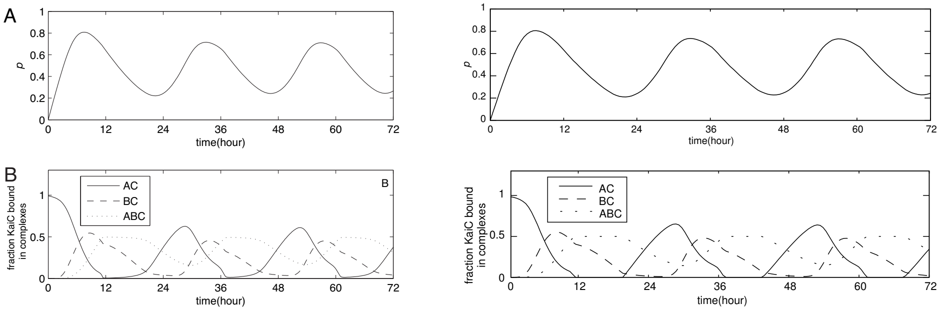
We have a prototype tool that takes textual descriptions of $c\pi$ systems and applies the semantics of §2.2: exploring the transition state space of species, and then combining these to compute the potential and immediate behaviour of processes. The tool is written in Haskell [26,27], and generates ODEs in a format suitable for numerical analysis by Octave [28,29].

Figure 8 shows the result of this tool applied to the $c\pi$ model of Figure 7. We take values for the 65 or so system parameters from those proposed by van Zon et al. in [25], with some corrections from [30]. Our tool generates a set of 50 ODEs covering all derived species, which are readily solved by Octave. For comparison, we have replicated the graphs presented in [8] to show sustained oscillations in the model: the original graphs are on the left, ours on the right.

4 Discussion

4.1 Alternative behavioural semantics

Process algebras offer a distinct level of abstraction compared to more widespread dynamical models of biochemical systems such as ordinary differential equations or Markov chains. A process algebraic description of a system may be translated to more than one such formalism (see e.g. [31]) and so a modeller may choose the most



(a) Graphs from the model of van Zon et al. [8, Fig.3]

(b) Graphs generated from the continuous π-calculus system in Fig. 7

Fig. 8: Graphs comparing oscillatory behaviour of the models defined by van Zon et al. [8] and the $c\pi$ terms of §3. The upper graphs show mean phosphorylation level of KaiC over three circadian cycles. The lower ones show the relative amounts of the complexes KaiA-KaiC (active); KaiB-KaiC (inactive); and KaiA-KaiB-KaiC (sequestering KaiA and so inhibiting the phosphorylation of active KaiC).

appropriate dynamic paradigm for a given situation. This is true of $c\pi$: aside from the "native" ODE semantics, it is relatively easy to generate Markov chains (by introducing integer quantities of processes instead of real-valued concentrations) and it may also be possible to map the $c\pi$ syntax to other behavioural models (e.g. Petri Nets as in [32]). We see this flexibility as a strength of process algebras like $c\pi$.

4.2 Modelling evolution

The target application of continuous π is the investigation of Darwinian evolution on the molecular level. At present, we are able to identify two promising concrete applications of $c\pi$ in this context. The first is direct simulation of evolution; the other is analysis of evolutionary robustness.

Evolutionary trajectories In order to simulate molecular evolution by natural selection, we must be able to express variability, populations and fitness in our process-algebraic framework. While populations of individuals can be modelled simply as collections of processes, the other two concepts use the process-algebraic nature of the model in an essential way.

Variability We propose addressing qualitative and quantitative variation of pathway topology (connectivity) in two ways. The first is by considering small variations in affinity networks, in order to model changes in the interaction capability of existing active sites. The other is by altering the structure of the species and thus modelling evolution of new sites, domain duplication and similar higher-level discrete events. It is also possible to consider variation in the initial concentrations of processes and interpret this – particularly in simpler models – as variation in gene expression.

It remains to be investigated whether it is biologically sensible to include this type of variation in a model that is otherwise focused on the evolution of network topologies.

Fitness It is clear that any notion of fitness is problem-dependent and must be defined externally by the modeller. We plan to use a form of quantitative model checking to compute the fitness value. This requires a modeller to formulate a fitness measure in an appropriate logic.

A recently published study [33] uses the process algebra of *Beta-binders* [34] in a similar programme of simulating molecular evolution, which provides some validation of this approach.

Robustness A further intended application of the calculus is the study of neutrality of biochemical pathways and related concepts such as robustness and evolvability. Since by definition two pathways are neutral with respect to each other if they have the same fitness, the question of determining neutrality can be reduced to that of assessing fitness (see above). We plan to treat it separately, however, as it may be possible to characterize or approximate neutrality without actually computing fitness. For example, we might deem two pathways neutral if they satisfy the same subset of a well-chosen set Φ of sentences in an appropriate logic. Even more desirable would be to characterize neutrality via a suitable behavioural equivalence, requiring no input from the modeller at all. In either case we expected the method to be applicable only to a restricted class of pathways (actual biological entities), whose identification is a challenge in itself.

4.3 Hybrid modelling

A dynamical system is *hybrid* if its dynamics have both discrete and continuous characteristics. This is a common situation in models for cell biology: for example, consider gene regulation by simple direct negative feedback. A protein is produced at a constant rate (continuous dynamics) until transcription is switched off due to a protein molecule binding to the DNA (discrete event). Continuous π as it is now cannot model this situation because we require every molecular species to be present in some concentration, while the DNA is present as a single copy only. In general, whenever it is impossible (or undesirable) to abstract over gene expression (or at least transcription), we are faced with the need to model a genuinely hybrid situation. In addition, even when the use of a purely continuous approach is conceptually possible, a model may fail to produce characteristic behaviour that depends crucially on stochastic effects. See e.g. [35] for a comparison of the continuous and stochastic approaches.

There are a variety of computational approaches to hybrid modelling, such as hybrid I/O automata [36] and hybrid process algebras [37]. We plan to build on this tradition and extend $c\pi$ with discrete features, in the form of species present as single individuals. Interactions with these species will act as discrete control events on top of the existing continuous semantics, giving truly hybrid process behaviour. We take the *lac* operon molecular system [38] as a suitable target to validate this approach. This is a regulatory network involving protein-DNA interactions which modify the transcription process of several genes; what is more, it is relatively well understood and well known among biologists.

4.4 Refinement of semantics

At present the potential behaviour ∂P of a process includes a description of every reaction in which it can engage. While this is necessary for compositionality, we believe there is room for improvement in the encoding of this information. Specifically, we need not record for every offered communication precisely what is consumed and what produced; a "net result" is enough (cf. formulae (11) and (13)). Consider for example the process

$$P = c \cdot A, \qquad \text{where } A \stackrel{\text{df}}{=} a.(A \mid B)\ . \tag{32}$$

In the semantics of §2.2 this communication potential is recorded as $\partial P(A, (;)(A \mid B), a) \neq 0$: in a single communication event over a one A is lost while another is produced along with B. It would be enough just to record that this reaction results in a production of a single new B. It remains to formalize this and extend it to arbitrary concretions.

Finally, we conjecture that $\mathbb{R}^\omega$ is unnecessarily large to serve as $\mathbb{P}$ and that for any process P, its immediate behaviour $\frac{dP}{dt}$ is an element of an ℓ^p space for some fixed p (the space of infinite real sequences with a finite p-norm). Moving from $\mathbb{R}^\omega$ to ℓ^p would allow us to use the rich theory of Banach (and Hilbert, if $p = 2$) spaces to study the properties of the calculus and to approach biological questions about, for example, system trajectories.

References

1. Priami, C., Regev, A., Shapiro, E., Silverman, W.: Application of a stochastic name-passing calculus to representation and simulation of molecular processes. Inf. Proc. Lett. **80** (2001)
2. Regev, A.: Computational Systems Biology: A Calculus for Biochemical Knowledge. PhD thesis, Tel Aviv University (2002)
3. Regev, A., Silverman, W., Shapiro, E.: Representation and simulation of biochemical processes using the pi-calculus process algebra. Pacific Symposium on Biocomputing (2001)
4. Kitano, H.: Biological robustness. Nature **5** (2004) 826–837
5. Wagner, A.: Robustness and Evolvability in Living Systems. Princeton University Press (2005)
6. Schuster, P., Fontana, W., Stadler, P., Hofacker, I.: From sequences to shapes and back: A case-study in RNA secondary structures. Proc. Royal Soc. Ser. B **255** (1994)
7. Tomita, J., Nakajima, M., Kondo, T., Iwasaki, H.: No transcription-translation feedback in circadian rhythm of KaiC phosphorylation. Science **307**(5707) (2005) 251–254
8. van Zon, J.S., Lubensky, D.K., Altena, P.R.H., ten Wolde, P.R.: An allosteric model of circadian KaiC phosphorylation. PNAS **104**(18) (2007) 7420–7425
9. Milner, R.: The polyadic π-calculus: A tutorial. Technical Report ECS-LFCS-91-180, LFCS, University of Edinburgh (1991)
10. Milner, R.: Communicating and Mobile Systems: The π Calculus. Cambridge University Press (1999)
11. Parrow, J.: An introduction to the π-calculus. In: Handbook of Process Algebra. Elsevier (2001) 479–543
12. Gillespie, D.T.: The chemical Langevin equation. J. Chem. Phys. **113**(1) (2000) 297–306
13. Regev, A., Shapiro, E.: Cellular abstractions: Cells as computations. Nature **419** (2002)

14. Regev, A., Panina, E.M., Silverman, W., Cardelli, L., Shapiro, E.: Bioambients: An abstraction for biological compartments. Theor. Comput. Sci. **325** (2004)
15. Hillston, J.: A Compositional Approach to Performance Modelling. Cambridge University Press (1996)
16. Calder, M., Gilmore, S., Hillston, J.: Modelling the influence of RKIP on the ERK signalling pathway using the stochastic process algebra PEPA. In: Proc. BioConcur. (2004)
17. Heath, J., Kwiatkowska, M., Norman, G., Parker, D., Tymchyshyn, O.: Probabilistic model checking of complex biological pathways. Theor. Comput. Sci. (2007)
18. Calder, M., Duguid, A., Gilmore, S., Hillston, J.: Stronger computational modelling of signalling pathways using both continuous and discrete-state methods. Lecture Notes in Bioinformatics **4210** (2006)
19. Kitano, H.: Towards system-level understanding of biological systems. In: Foundations of Systems Biology. MIT Press (2001)
20. Stadler, B.M.R., Stadler, P.F., Wagner, G., Fontana, W.: The topology of the possible: Formal spaces underlying patterns of evolutionary change. J. Theor. Biol. **213** (2001)
21. Soyer, O., Salathe, M., Bonhoeffer, S.: Signal transduction networks: Topology, response, and biochemical reactions. J. Theor. Biol. **238** (2006)
22. Plotkin, G.D.: A structural approach to operational semantics. J. Log. Algeb. Progr. **60–61** (2004) 17–139
23. Ishiura, M., Kutsuna, S., Aoki, S., Iwasaki, H., Andersson, C.R., Tanabe, A., Golden, S.S., Johnson, C.H., Kondo, T.: Expression of a gene cluster kaiABC as a circadian feedback process in cyanobacteria. Science **281**(5382) (1998) 1519–1523
24. Golden, S.S., Johnson, C.H., Kondo, T.: The cyanobacterial circadian system: A clock apart. Current Opinion in Microbiology **1**(6) (1998) 669–673
25. van Zon, J.S., Lubensky, D.K., Altena, P.R.H., ten Wolde, P.R.: An allosteric model of circadian KaiC phosphorylation: Supporting information. Available at `http://www.pnas.org/cgi/content/full/0608665104/DC1` (2007)
26. Haskell: `http://www.haskell.org/`
27. Peyton Jones, S., ed.: Haskell 98 Language and Libraries: The Revised Report. Cambridge University Press (April 2003)
28. Octave: `http://www.gnu.org/software/octave/`
29. Eaton, J.W.: GNU Octave Manual. Network Theory (2002)
30. van Zon, J.S.: A detail of the KaiABC model. Personal communication (2008)
31. Calder, M., Gilmore, S., Hillston, J.: Automatically deriving ODEs from process algebra models of signalling pathways. In: Proc. CMSB. (2005)
32. Meyer, R., Khomenko, V., Strazny, T.: A practical approach to verification of mobile systems using net unfoldings. In: Application and Theory of Petri Nets: Proc. ATPN. (2008) to appear.
33. Demate, L., Priami, C., Romanel, A., Soyer, O.: A formal and integrated framework to simulate evolution of biological pathways. LNBI **4695** (2007)
34. Priami, C., Quaglia, P.: Beta binders for biological interactions. Lecture Notes in Computer Science **3082** (2004)
35. Bortolussi, L., Policriti, A.: Connecting process algebras and differential equations for systems biology. Process Algebra and Stochastically Timed Activities: Proc. 6th PASTA workshop (2006)
36. Segala, R., Vaandrager, F., Lynch, N.: Hybrid I/O automata. Inf. & Comput. **185**(1) (2003)
37. Bergstra, J.A., Middleburg, C.A.: Process algebra for hybrid systems. Theor. Comput. Sci. **335** (2005)
38. Vilar, J.M.G., Guet, C.G., Leibler, S.: Modeling network dynamics: The lac operon, a case study. J. Cell Biol. **161** (2003)