

# From Processes to ODEs by Chemistry

Luca Cardelli

Microsoft Research luca@microsoft.com

**Abstract.** We investigate the collective behavior of processes in terms of differential equations, using chemistry as a stepping stone. Chemical reactions can be converted to ordinary differential equations, and also to processes in a stochastic process algebra. Conversely, certain stochastic processes (in *Chemical Parametric Form*, or CPF) can be converted to chemical reactions. CPF is a subset of  $\pi$ -calculus, but is already more powerful than what is strictly needed to represent chemistry: it supports also parameterization and compositional reuse of models. The mapping of CPF to chemistry thus induces a parametric and compositional mapping of CPF to differential equations; the indirect mapping through chemistry is easier to define and understand than a direct mapping. As an example, we derive a quantitative interleaving law from the differential equations.

## 1 Introduction

In Systems Biology, biochemical systems are routinely described as large state transition diagrams with rates on transition [20], with emphasis on the graphical and database-oriented representation of the models. Graphical representation has advantages in terms of readability and sharing of information, but has obvious disadvantages in terms of precision, scalability (compositionality), and analyzability. Given that such models are presented already as state transition systems, it is natural to interpret them as term-rewriting systems or process algebras. These interpretations have a better chance than diagrams of satisfying scalability requirements, and can be mapped to increasingly promising analysis tools and techniques [23,17], including ones such as Petri Nets that already have a long tradition in other areas. This is not to say that current modeling approaches lack formality: biochemical systems, including the diagrammatic ones just discussed, are usually interpreted as systems of chemical reactions, and ultimately as systems of ordinary differential equations. A question then arises: what is the relationship between transition-based models and differential equation models, and even more fundamentally with chemistry?

Our starting point is stochastic process algebra, which provides us with a quantitative compositional semantics, and with simulation and analysis techniques [15,25]. Our goal is to relate process algebra to ordinary differential equations (ODEs), so that one can hopefully use techniques and tools from both camps [16]. We cannot carry out this program yet for general process algebras (e.g.  $\pi$ -calculus [21]) where some features go “beyond ordinary chemistry”, but we can do it for interesting fragments that include unbounded-state systems, and that provide rich model parameterization. In this paper we establish a mapping from processes to ODEs via a detour through

chemical reactions, primarily to obtain an easy two-step translation, but also to build a connection with chemistry. The foundations of this work are covered in more detail elsewhere [6]; here we emphasize the intuitive connection to chemical reactions by a number of examples, and we additionally handle parametric models.

The basic chemistry of well-mixed solutions can be described, at the molecular level, in terms of random molecular collisions and subsequent reactions. From this *microscopic* point of view, it can be modeled by Markov chains in continuous time (with real-valued reaction rates), with a discrete unbounded state space (an unbounded number of molecules that may flow in or be produced), and with a bounded number of chemical species (kinds of molecules). Alternatively, *macroscopically*, the fundamental law of interaction in chemistry is the *law of mass action*, which quantitatively determines the speed of reactions based on the continuous state space of concentrations of chemical species. The relationships between the discrete-space and the continuous-space views are subtle [10, 30, 6], and must be considered when relating discrete-state stochastic process algebra to continuous-state differential equations.

This paper is organized as follows. In Section 2 we introduce the notation of chemical reactions and its standard interpretation in terms of ordinary differential equations, relating changes of concentration of chemical species over time [18]. We also provide an interpretation of chemical reactions as a stochastic process algebra (CGF) that is a fragment of well-known ones. Stochastic processes can in turn be mapped to continuous-time Markov chains [12], which provide another standard interpretation of chemical reactions [11]. In Section 3 we translate stochastic processes (CGF) back to chemical reactions. We also consider a more general process algebra (CPF) that supports various kinds of parameterization. We show how to translate CPF down to CGF, and we provide an incremental algorithm for doing so. We thus obtain a systematic way of translating CPF parametric stochastic processes, through chemical systems, to differential equations. In section 4 we give various examples of the translations, including non-chemical ODE systems such as Kermack-McKendrick epidemics and Lotka-Volterra predation. The mapping to ODEs can be used also as a semantics of processes, and we show how to derive a quantitative interleaving law from it.

## 2 From Chemistry to ODEs and Processes

### 2.1 Chemical Reactions

We consider three basic kinds of chemical reactions. First, in *unary reactions*, a chemical species  $A$  may spontaneously degrade into other species; the rate of such a reaction is given by the *exponential decay law*: the rate is proportional to the concentration of the species  $A$ . Second, in *hetero binary reactions*, two chemical species  $A_1$  and  $A_2$  may react and produce other species; the rate of such a reaction is given by the *law of mass action*: the rate is proportional to the product of the concentrations of  $A_1$  and  $A_2$ . Third, in *homeo binary reactions*,  $A_1$  and  $A_2$  are the same species  $A$ , and the rate is then pro-

portional to the square of the concentration of A. We write  $[A]$  for the concentration of A in moles per liter as a function of continuous time, and  $[A]^\bullet$  for its time derivative.

$A$	$\xrightarrow{k} B_1 + \dots + B_n$	Unary	$k : s^{-1}$	$[A]^\bullet = -k[A]$
$A_1 + A_2$	$\xrightarrow{k} B_1 + \dots + B_n$	Hetero Binary	$k : M^{-1}s^{-1}$	$[A_i]^\bullet = -k[A_1][A_2]$
$A + A$	$\xrightarrow{k} B_1 + \dots + B_n$	Homeo Binary	$k : M^{-1}s^{-1}$	$[A]^\bullet = -2k[A]^2$ (assuming $A \neq B_i \neq A_j$ for all $i, j$ )

**Table 1** The Three Kinds of Chemical Reactions

Chemical reactions and the law of mass action can be presented in a more general form, with any number of molecules on the left-hand side. Still, the only chemical reactions of interest to us are unary and binary, in view of the molecular interpretation of interactions between chemical species. For example, we can ignore unlikely reactions that require three molecules to collide at the same time: “Genuinely *trimolecular* reactions do not physically occur in dilute fluids with any appreciable frequency. *Apparently* trimolecular reactions in a fluid are usually the combined result of two bimolecular reactions and one monomolecular reaction, and involve an additional short-lived species.”[11]

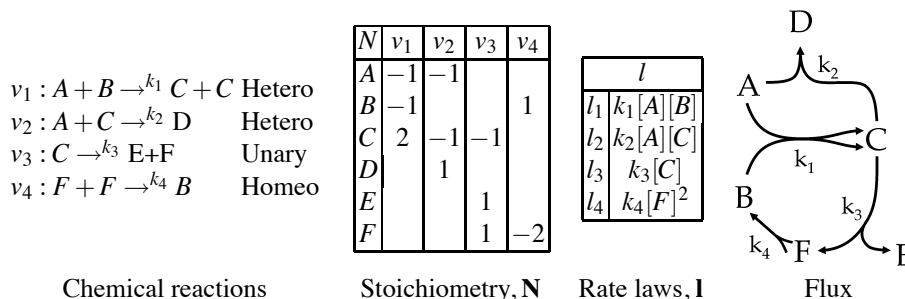
A *system of chemical reactions* is a finite set of reactions between a finite set of chemical species<sup>1</sup>. We assume, as is common, that our reactions happen in a *well-stirred solution*, that is, that the dynamics of chemical reactions depends only on concentrations of the species (and on other factors, such as temperature, that are assumed fixed), and not on the positions of the molecules. Each reaction,  $\xrightarrow{k}$ , has a (*base*) *rate*,  $k$ , which is a proportionality constant used in the corresponding rate law, with bigger base rates meaning faster reactions. The *initial conditions* of the system, that is, the initial concentrations of the chemical species, are specified separately from the reactions.

## 2.2 From Chemistry to ODEs

A system of ordinary differential equations can be extracted from any system of chemical reactions, to describe the rate of change in concentration of chemical species over time. The ODEs provides the *kinetics* of the chemical system, that is, they completely describe the dynamic time evolution of the various quantities.

The procedure for extracting ODEs is standard [18]. Consider, as an example, the following system of 4 chemical reactions  $\nu_1, \nu_2, \nu_3, \nu_4$  with corresponding reaction rates  $k_1, k_2, k_3, k_4$ , between 6 chemical species  $A, B, C, D, E, F$ .

<sup>1</sup> More generally, it could be a collection of reactions and chemical species indexed by an infinite set; this is necessary, e.g., to describe polymerization. A corresponding effect can be obtained within -calculus [26], but here we consider only finite systems of reactions.



We first build the *stoichiometric matrix*,  $N$ , which has one row for each species and one column for each reaction. Each cell  $\langle S, v \rangle$  in the matrix contains a positive number  $n$  if  $n$  molecules of species  $S$  are produced (overall) in reaction  $v$ ; it contains a negative number  $-n$  if  $n$  molecules of species  $S$  are removed (overall) in reaction  $v$ , and otherwise it contains 0.

Then, we build the *vector of rate laws*,  $l$ : for each reaction it specifies the rate law for that reaction. In our case,  $v_1, v_2$  have the hetero rate law,  $v_3$  has the unary rate law, and  $v_4$  has the homeo rate law. (In general, other rate laws may include steady-state approximations, such as the Michaelis-Menten law for enzymatic reactions, or empirical rate laws).

Finally let  $\mathbf{X}$  be the vector of chemical species ( $A, B, C, D, E, F$ ). The system of ODEs is then, in general, given by the following *rate equation*:

$$[\mathbf{X}]^\bullet = \mathbf{N} \cdot \mathbf{l}$$

**Table 2** From Chemical Reactions to Ordinary Differential Equations

Expanding for our set of reactions we obtain:

$$\begin{aligned} [A]^\bullet &= -l_1 - l_2 = -k_1[A][B] - k_2[A][C] & [D]^\bullet &= l_2 = k_2[A][C] \\ [B]^\bullet &= -l_1 + l_4 = -k_1[A][B] + k_4[F]^2 & [E]^\bullet &= l_3 = k_3[C] \\ [C]^\bullet &= 2l_1 - l_2 - l_3 = 2k_1[A][B] - k_2[A][C] - k_3[C] & [F]^\bullet &= l_3 - 2l_4 = k_3[C] - 2k_4[F]^2 \end{aligned}$$

The rate law  $l_4$  for the homeo reactions is  $k_4[F]^2$ , but the contribution of  $v_4$  to  $[F]^\bullet$  is  $-2k_4[F]^2$  because two  $F$  are consumed in that reaction (hence the rate law shown in table 2). Compare this with the contribution of  $v_4$  to  $[B]^\bullet$ , which is  $k_4[F]^2$ .

### 2.3 Processes in Chemical Ground Form

We introduce a subset of  $\pi$ -calculus (and of CCS) [21]: the *Chemical Ground Form* (CGF) [6], which is sufficient, in stochastic version, for translating chemical reactions to processes. See [25, 23, 3, 4, 12, 13, 14, 15] for the semantics of stochastic  $\pi$ -calculus and other stochastic process algebras.

$E ::= 0 \dot{:} X = M, E$	Reagents	(empty, or a reagent $X=M$ and Reagents)
$M ::= 0 \dot{:} \pi; P \oplus M$	Molecule	(empty, or an interaction $\pi; P$ and Molecule)
$P ::= 0 \dot{:} X   P$	Solution	(empty, or a variable $X$ and Solution)
$\pi ::= \tau_{(r)} \dot{:} ?n_{(r)} \dot{:} !n_{(r)}$	Interaction prefix	(delay, input, output)
$CGF ::= E, P$	Chemical Ground Form	(Reagents with initial Solution)

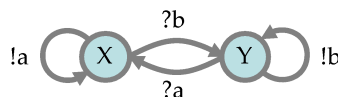
**Table 3** Chemical Ground Form (CGF)

A chemical ground form CGF has a finite set  $E$  of *reagents*  $X_i = M_i$  (named molecules) for distinct *variables*  $X_i$  naming chemical species, and *molecules*  $M_i$  describing the interaction capabilities of the corresponding species. The possible process *interactions*  $\pi$  are: *delay*  $\tau_{(r)}$  at *rate*  $r$  (where  $r$  is a positive real), *input*  $?n_{(r)}$  on *channel*  $n$  at *rate*  $r$ , and *output*  $!n_{(r)}$  on *channel*  $n$  at *rate*  $r$  (each channel always has the same rate). In the syntax of molecules, each interaction  $\pi$  leads to releasing a *solution*  $P$  (a multiset of variables). We use  $\oplus$  for *choice*,  $|$  for *parallel composition*, and  $0$  for the empty reagent, the empty molecule, and the empty solution. Trailing  $0$ 's are usually left implicit, and we use  $|$  also as an operator over the syntax: if  $P$  and  $P'$  are  $0$ -terminated lists of variables, according to the syntax above, then  $P|P'$  means appending the two lists into a single  $0$ -terminated list. Therefore, if  $P$  is a solution, then  $0|P$ ,  $P|0$ , and  $P$  are syntactically equal.

A CGF  $(E, P)$  is a set of reagents  $E$  together with *initial conditions*, which are a solution  $P$ . If a variable  $X$  occurs in some  $M_i$  or initial conditions  $P$ , but  $X$  is not defined in  $E$ , we can assume the existence of an additional reagent  $X = 0$ . The meaning of a CGF can be given by directly extracting a continuous time Markov chain from it [6].

Some CGFs can be drawn conveniently as stochastic interacting automata; for example here is a two-state automaton, with states  $X, Y$ , which interacts with copies of itself over the channels  $a_{(r)}, b_{(s)}$ :

$$\begin{aligned}
 X &= !a_{(r)}; X \oplus ?b_{(s)}; Y \\
 Y &= !b_{(s)}; Y \oplus ?a_{(r)}; X \\
 X &| X | X | Y | Y \quad (\text{initial conditions})
 \end{aligned}$$



In general, however, a CGF can “split” by parallel composition after an interaction, and then some less standard graphical notation (similar to Petri Net transitions) must be used to represent such splitting.

## 2.4 From Chemistry to Processes

We now discuss how to produce CGF processes from systems of chemical reactions. We need, in particular, to convert concentrations of chemical species, of dimension  $M$  (*molarity*), to discrete numbers of molecules/processes, for which we need a conversion factor  $\gamma$  of dimension  $M^{-1}$ . In chemistry,  $\gamma = N_A V$ , where  $N_A$  is Avogadro’s number, and  $V$  is the volume of the solution; if we take  $\gamma = 1.0$ , for example, it means that we are considering a chemical solution of volume  $1/N_A$ .

The factor  $\gamma$  has other uses too. The mass action rates “ $k$ ” have dimension  $s^{-1}$  for unary reactions and  $M^1 s^{-1}$  for binary reactions. Stochastic processes, instead, operate on molecule counts, and the stochastic rates “ $r$ ” always have dimension  $s^{-1}$ . Therefore, an appropriate  $M^{-1}$  conversion factor is needed for the rates of binary reactions. In particular, the conversions between stochastic “ $r$ ” and mass action “ $k$ ” rates are:  $r = k$  for unary reactions,  $r = k/\gamma$  for hetero reactions, and  $r = 2k/\gamma$  for homeo reactions [10, 30]. There is, however, an additional twist for homeo reactions. The natural encoding of homeo reactions, as processes that offer both an input and an output on the same channel, produces an artificial doubling of interactions; see for example  $F$  below, where 2 copies of  $F$  have 2 interactions on channel  $v_4$ , instead of 1 “collision” [25]. We can compensate for this doubling by halving the stochastic rate of the interaction channel, with the net effect that homeo *channels* too end up with  $r = k/\gamma$ . We keep the two contributions to the rate of homeo reactions separate in Table 4 for emphasis.

To convert chemical reactions to process reagents, we first prepare a separate channel  $v_{(r)}$  of rate  $r$  for each binary reaction  $v$  of rate  $k$ , setting  $r = k/\gamma$  as discussed. The unary reactions do not need channels, and use a  $\tau$  delay with  $r = k$ . Setting up such channels is similar to setting up the vector of rate laws in Section 2, but fixing the base rates is sufficient here because the semantics of the intended process algebra already incorporates the decay law and the mass action law [23].

With these channels, we can produce the CGF reagents for the reactions from Section 2:

$$\begin{array}{ll}
 A = ?v_1(k_1/\gamma); (C|C) \oplus ?v_2(k_2/\gamma); D & \oplus ?cA_{(0)}; 0 \\
 B = !v_1(k_1/\gamma); 0 & \oplus ?cB_{(0)}; 0 \\
 C = !v_2(k_2/\gamma); 0 \oplus \tau_{(k_3)}; (E|F) & \oplus ?cC_{(0)}; 0 \\
 D = 0 & \oplus ?cD_{(0)}; 0 \\
 E = 0 & \oplus ?cE_{(0)}; 0 \\
 F = ?v_4(k_4/\gamma); B \oplus !v_4(k_4/\gamma); 0 & \oplus ?cF_{(0)}; 0
 \end{array}$$

That is done as follows. For each species  $X$  we produce an initially empty reagent,  $X = 0$ . Then we scan each chemical reaction to gradually populate the reagents with summands. For a degradation reaction  $v : X \rightarrow^k P$  we add a summand  $\tau_{(r)};P$  with  $r = k$  to the reagent  $X$ . For a hetero reaction  $v : X + Y \rightarrow^k P$  we add a summand  $?v_{(r)};P$  with  $r = k/\gamma$  to the reagent  $X$  and a summand  $!v_{(r)};0$  to the reagent  $Y$ , using the reaction names as the channel names. For a homeo reaction  $v : X + X \rightarrow^k P$  we add two summands  $?v_{(r/2)};P$  and  $!v_{(r/2)};0$  to the reagent  $X$ , with  $r = 2k/\gamma$ . (We also change all chemical “+” to process “|”.)

We may optionally add an extra summand  $?cX_{(0)};0$  to the definition of each  $X$ , where  $cX$  is a channel where no interaction ever happens. This is useful if we want to *observe* the system (e.g., counting the number of  $X$  for plotting), by observing how many  $?cX$  are being generated [23].

The formal procedure for obtaining processes  $Pi_\gamma(C)$  from a chemical system  $C$  is finally given in Table 4, assuming that the reactions in  $C$  are uniquely named. The initial conditions of a chemical system consist of a vector  $V$  of concentrations  $V_{X_i} : M = [X_i]$  for the various species  $X_i$ ; these can be converted to CGF initial conditions  $P$  with  $\#X_i(P) = \lceil \gamma[X_i] \rceil$ , with a rounding error.  $Pi_\gamma(C)$  has the same dynamics as  $C$  [6].

$ \begin{aligned} Pi_\gamma(C) = \{ & (X = \oplus((v : X \rightarrow^k P) \in C) \text{ of } (\tau_{(r)};P) && \oplus \text{ with } r = k \\ & \oplus((v : X + Y \rightarrow^k P) \in C \text{ and } Y \neq X) \text{ of } (?v_{(r)};P) && \oplus \text{ with } r = k/\gamma \\ & \oplus((v : Y + X \rightarrow^k P) \in C \text{ and } Y \neq X) \text{ of } (!v_{(r)};0) && \oplus \text{ with } r = k/\gamma \\ & \oplus((v : X + X \rightarrow^k P) \in C) \text{ of } (?v_{(r/2)};P \oplus !v_{(r/2)};0) && \oplus \text{ with } r = 2k/\gamma \\ & \text{s.t. } X \text{ is a species in } C \} \\ Pi_\gamma(C, V) = E, P & \text{ where } E = Pi_\gamma(C) \text{ and } \#X(P) = \lceil \gamma V^X \rceil \text{ for all } X \in E \end{aligned} $
---

**Table 4** From a Chemical Reaction System  $C$  to a Chemical Ground Form  $Pi_\gamma(C)$

$(C, V)$	$Pi_\gamma(C, V)$	where	initially	ODEs
$n : X \rightarrow^k 0,$	$V \quad X = \tau_{(r)};0, \quad P$	$r = k$	$\#X(P) = \lceil \gamma V_X \rceil$	$[X]^\bullet = -k[X] = -r[X]$
$n : X + Y \rightarrow^k 0,$	$V \quad X = ?n_{(r)};0, \quad P$ $Y = !n_{(r)};0,$	$r = k/\gamma$	$\#X(P) = \lceil \gamma V_X \rceil$ $\#Y(P) = \lceil \gamma V_Y \rceil$	$[X]^\bullet = -k[X][Y] = -r\gamma[X][Y]$ $[Y]^\bullet = -k[X][Y] = -r\gamma[X][Y]$
$n : X + X \rightarrow^k 0,$	$V \quad X = ?n_{(r/2)};0 \oplus$ $!n_{(r/2)};0, \quad P$	$r = 2k/\gamma$	$\#X(P) = \lceil \gamma V_X \rceil$	$[X]^\bullet = -2k[X]^2 = -r\gamma[X]^2$

**Table 5** Examples: from  $(C, V)$  to  $Pi_\gamma(C, V)$

### 3 From Processes to Chemistry and ODEs

We have seen that we can convert chemical reactions to ODEs. Therefore, a mapping from a stochastic process algebra to chemical reactions, which we study in this sec-

tion, induces in two steps a mapping from that process algebra to ODEs. A two-step approach is desirable, because a direct mapping from a process algebra to ODEs, although possible and intuitively understandable, is more challenging [16, 6]. The first step, from process algebra to chemical reactions, has the effect of identifying the transitions that the system performs, and the second step, from chemical reactions to ODEs, identifies the rate of change of populations of processes.

### 3.1 Processes in Chemical Parametric Form

We begin by defining a more general subset of  $\pi$ -calculus, the *Chemical Parametric Form* (CPF), which extends the CGF with parameterization and communication. CPF is not technically a subset of CCS, since it allows channel passing, but the subsequent translation of CPF to CGF essentially amounts to a translation of CPF to CCS. The reason for these subsets is that, in general, it is not possible to translate an arbitrary  $\pi$ -calculus process to a system of chemical reactions with a finite set of chemical species, because in full  $\pi$ -calculus, via name generation, we can generate unboundedly many species. Therefore, the CPF incorporates limitations that, as we shall see, are sufficient to enable the translation to chemistry. The limitations are not that our systems be *finite state* (since it is convenient to abstract from detailed accounting of energy and to express chemical systems that produce unbounded quantities of product), nor that they be *finite control* [8] (since parallel composition within recursive definitions models chemical reactions that generate multiple products). However, it is essential that our systems have a *finite number of species*.

$E ::= 0 \mid X(\mathbf{p}) = M, E$	Reagents (empty, or a parametric reagent $X(\mathbf{p}) = M$ and Reagents)
$M ::= 0 \mid \pi; P \oplus M$	Molecule (empty, or an interaction $\pi; P$ and Molecule)
$P ::= 0 \mid X(\mathbf{p}) \mid P$	Solution (empty, or an instanced variable $X(\mathbf{p})$ and Solution)
$\pi ::= \tau_{(r)} \mid ?n(\mathbf{p}) \mid !n(\mathbf{p})$	Interaction prefix (delay, parametric input, instanced output)
$CPF ::= E, P$	Chemical Parametric Form (Reagents with initial Solution)

**Table 6** Chemical Parametric Form (CPF)

The syntax of the CPF is the same as that of the CGF, but with additional parameter lists  $\mathbf{p}$ . There is a finite set  $E$  of parametric reagents  $X_i(\mathbf{p}_i) = M_i$  for distinct variables  $X_i$ . Each  $\mathbf{p}$  is a vector of distinct channel names, and  $\#\mathbf{p}$ , the length of  $\mathbf{p}$ , is the arity of the corresponding  $X$ , which must be used consistently through  $E, P$ . If  $X(\mathbf{p})$  occurs in some  $M_i$  or initial conditions  $P$ , but  $X$  is not defined in  $E$ , we can assume the existence of an additional reagent of the form  $X(\mathbf{q}) = 0$ .



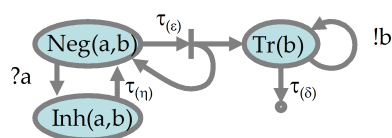
A name  $n$  in  $E$  is *free* if it occurs in some  $M_i$  but is not bound by the corresponding  $\mathbf{p}_i$  or any enclosing  $?n(\mathbf{p})$ ; we then say that  $n \in \text{fn}(E)$ . Moreover, any name  $n$  occurring in the initial conditions  $P$  is free ( $n \in \text{fn}(P)$ ). Each free name  $n$  is uniformly associated with a fixed rate  $n_{(r)}$ ; we may also keep track of this information separately by saying that  $\rho_{(E,P)}(n) = r$  for  $n \in \text{fn}(E,P)$ . The non-free (*parametric*) names are not annotated with a rate, and simply acquire the rate of the free names that they must be replaced with before any interaction can happen. Therefore, the possible process interactions are: stochastic delay  $\tau_{(r)}$  at rate  $r$ , input  $?n_{(r)}(\mathbf{p})$  of names  $\mathbf{p}$  (parametric and distinct) on channel  $n$  at rate  $r$ , and output  $!n_{(r)}(\mathbf{p})$  of names  $\mathbf{p}$  (free at the time of interaction) on channel  $n$  at rate  $r$ .

An example of a CPF system (with no initial conditions, no free names, and no input and output parameters), is the following gene gate  $\text{Neg}(a,b)$  [1]. This is a process that at stochastic intervals produces copies of  $\text{Tr}(b)$ , unless it is inhibited (for some time) on channel  $a$ .  $\text{Tr}(b)$  can in turn inhibit other gates that accept input on channel  $b$ , or decay.

$$\text{Neg}(a,b) = ?a(); \text{Inh}(a,b) \oplus \tau_{(\epsilon)}; (\text{Tr}(b)|\text{Neg}(a,b))$$

$$\text{Inh}(a,b) = \tau_{(\eta)}; \text{Neg}(a,b)$$

$$\text{Tr}(b) = !b(); \text{Tr}(b) \oplus \tau_{(\delta)}; 0$$



This description is parametric in that it defines the behavior of a gate in a network without specifying its connectivity; the connectivity of the network is then given in the initial conditions. Initial conditions for this CPF system could be given by  $\text{Neg}(x_{(r)}, x_{(r)})$ : a single gate with a self loop (with free name  $x_{(r)}$ ), or by  $\text{Neg}(x_{(r)}, y_{(s)}) | \text{Neg}(y_{(s)}, z_{(t)}) | \text{Neg}(z_{(t)}, x_{(r)})$ : a network of three gates (with free names  $\{x_{(r)}, y_{(s)}, z_{(t)}\}$ ) which can function as a stochastic oscillator.

A more general normal form, that can represent any  $\pi$ -calculus process, can be obtained by allowing  $\pi$ -calculus restriction in reagents:  $X(\mathbf{p}) = (\nu \mathbf{q})M$ . This way we can express complexation and polymerization by channel passing [25], but ODE translations are not currently known.

### 3.2 From CGF to Chemistry

The chemical ground form, CGF, from Section 2.3 is a restricted version of the CPF, where there are zero parameters in definitions, inputs and outputs. Empty parameters,  $()$ , are omitted.

We first consider the problem of converting a CGF  $(E,P)$  to a system of chemical reactions  $Ch_{\gamma}(E,P)$  (the resulting  $Ch_{\gamma}(E)$  has the same dynamics as  $E$  [6]). This is achieved by producing a degradation reaction for each  $\tau_{(r)}$  delay in  $E$ , a hetero reaction for each pair  $?a, !a$  of interactions in different molecules of  $E$ , and a rate-doubled homeo reaction for each pair of interactions  $?a, !a$  in the same molecule of  $E$ . Several

examples are shown in section 4. The mass action rate for homeo reactions is  $r\gamma$ , but we keep the factors contributing to it ( $r\gamma = 2(r\gamma/2)$ ) separate in Table 7 for emphasis.

$Ch_\gamma(E) =$	
$\{\{X.i\}X \rightarrow^k P \text{ s.t. } E(X).i = \tau_{(r)}; P\} \cup$	with $k = r$
$\{\{X.i.Y.j\}X + Y \rightarrow^k P + Q \text{ s.t. } X \neq Y \text{ and } E(X).i = ?n_{(r)}; P \text{ and } E(Y).j = !n_{(r)}; Q\} \cup$	with $k = r\gamma$
$\{\{X.i.X.j\}X + X \rightarrow^{2k} P + Q \text{ s.t. } E(X).i = ?n_{(r)}; P \text{ and } E(X).j = !n_{(r)}; Q\}$	with $k = r\gamma/2$
$Ch_\gamma(E, P) = C, V$ where $C = Ch_\gamma(E)$ and $V_X = \#X(P)/\gamma$ for all $X \in E$	

**Table 7** From a Chemical Ground Form  $E$  to a Chemical Reaction System  $Ch_\gamma(E)$

When inserting a  $P$  into a chemical reaction, we change all process “|” to chemical “+”. The initial conditions of the chemical system can be obtained from the initial conditions  $P$  of the CGF by setting  $[X_i] = \#X_i(P)/\gamma$  for each species  $X_i$ . Note how we have tagged the resulting reactions (by  $\{\dots\}$ ): here  $M.i$  is the  $i$ -th summand in molecule  $M$ , and  $X.i$  refers to the molecule summand  $E(X).i$ . This tagging allows us to easily account for multiplicity of reactions. Applying this procedure to the process reagents  $A \dots F$  in Section 2.4 (without the observer channels), reproduces the system of reactions from Section 2.

$(E, P)$	$Ch_\gamma(E, P)$	where	initially	ODEs
$X = \tau_{(r)}; 0, P$	$\{X.1\} X \rightarrow^k 0, V$	$k = r$	$V_X = \#X(P)/\gamma$	$[X]^\bullet = -k[X] = -r[X]$
$X = ?n_{(r)}; 0, Y = !n_{(r)}; 0, P$	$\{X.1.Y.1\} X + Y \rightarrow^k 0, V$	$k = r\gamma$	$V_X = \#X(P)/\gamma$ $V_Y = \#Y(P)/\gamma$	$[X]^\bullet = -k[X][Y] = -r\gamma[X][Y]$ $[Y]^\bullet = -k[X][Y] = -r\gamma[X][Y]$
$X = ?n_{(r)}; 0 \oplus !n_{(r)}; 0, P$	$\{X.1.X.2\} X + X \rightarrow^{2k} 0, V$	$k = r\gamma/2$	$V_X = \#X(P)/\gamma$	$[X]^\bullet = -4k[X]^2 = -2r\gamma[X]^2$

**Table 8** Examples: from  $(E, P)$  to  $Ch_\gamma(E, P)$

### 3.3 From CGF to ODEs directly

We have seen how to convert CGF to chemistry (Section 3.2) and how to convert chemistry to ODEs (Section 2). We can obviously compose the two conversions to go from CGF to ODEs, but we can also do it more directly via a stoichiometric matrix technique. In comparison to the chemical technique, the role of the set of chemical reactions is replaced by the following set:

$$\begin{aligned} \mathcal{I} = & \{ \{X.i\} \text{ s.t. } E.X.i = \tau_{(r)}.Q \} \\ & \cup \{ \{X.i.Y.j\} \text{ s.t. } E.X.i = ?n_{(r)}.Q \text{ and } E.Y.j = !n_{(r)}.R \} \quad (\text{for any } r, n, Q, R) \end{aligned}$$

$\mathcal{I}$  is the finite set of *possible interactions* arising from a set of reagents  $E$ , where  $X.i$  is an ordered pair identifying a molecule summand in  $E$ , and  $E.X.i$  is a molecule summand as previously defined.

The stoichiometric matrix used in the conversion has as many rows as species, and as many columns as interactions  $\mathcal{I}$ : each column contains coefficients for the reagents that are gained or lost in that interaction. The corresponding vector of rate laws has as many rows as interactions, and contains the rate laws for the interactions. For example, in volume  $\gamma : M^{-1} = N_A V$  where  $N_A : mol^{-1}$  and  $V : L$ , and  $r, t, u : s^{-1}$ , we have:

$$\begin{array}{l} X = \tau_{(t)}.Y \oplus ?a_{(r)}.X \\ Y = !a_{(r)}.(Y|Y) \oplus \tau_{(u)}.Y \end{array} \quad \{\{X.1\}, \{X.2, Y.1\}, \{Y.2\}\}$$

Processes Interactions  $\mathcal{I}$

$N$	$\{X.1\}$	$\{X.2, Y.1\}$	$\{Y.2\}$
$X$	-1	0	0
$Y$	+1	+1	-1

Stoichiometry,  $\mathbf{N}$ 

$\mathbf{I}$	
$\{X.1\}$	$t[X] : M \cdot s^{-1}$
$\{X.2, Y.1\}$	$r\gamma[X][Y] : M \cdot s^{-1}$
$\{Y.2\}$	$u[Y] : M \dots^{-1}$

Rate laws,  $\mathbf{I}$ 

The ODEs are then obtained, as in the chemical technique, as  $\mathbf{N} \cdot \mathbf{I}$ :

$$\begin{aligned} [X]^\bullet : M \cdot s^{-1} &= -t[X] \\ [Y]^\bullet : M \cdot s^{-1} &= t[X] + r\gamma[X][Y] - u[Y] \end{aligned}$$

### 3.4 From CPF to CGF

The procedure in Section 3.2 allows us to obtain chemical systems from ground forms. But we can use it also for the more general parametric forms, if we can first convert a CPF to a CGF. To that end, *grounding* ( $/_N$ ) is a process that converts molecules and solutions of a CPF to those of a CGF. It eliminates parameters on the basis of a set of free names  $N$  (covering all free names), which is initially chosen to be that of the CPF. Here  $n/\mathbf{p}$  denote (single) channel names in bijection with pairs  $\langle n, \mathbf{p} \rangle$ , and  $X/\mathbf{p}$  denote species names in bijection with pairs  $\langle X, \mathbf{p} \rangle$  (any rate annotations in  $\mathbf{p}$  are ignored). Each  $X/\mathbf{p}$  has the role of a separate chemical species for the parameter instantiation given by  $\mathbf{p}$ .

In Table 9,  $n/\mathbf{p}_{(r)}$  means that the (single) name  $n/\mathbf{p}$  is annotated with  $r$ . The notation  $\{\mathbf{p} \leftarrow \mathbf{q}\}$  is the simultaneous substitution of name vectors, where  $\mathbf{q} \in N^{\#\mathbf{p}}$  are vectors of free names (of the same size as  $\mathbf{p}$ ) and hence annotated with rates. As an invariant of the definition, the names in channel position and in output must be annotated with rates; this is maintained by  $\{\mathbf{p} \leftarrow \mathbf{q}\}$ .

Then, a process of *parametric explosion* converts a parametric form  $E$ , to a ground form  $E_G$ , by instantiating all possible parameter lists with respect to the set  $N$  of free names of  $E$ . Grounding is used in such a process. The initial conditions are simply grounded once.

$$\begin{aligned}
/N(\tau_{(r)};P) &= \tau_{(r)}/N(P) \\
/N(!n_{(r)}(\mathbf{p});P) &= !n/\mathbf{p}_{(r)}/N(P) \\
/N(?n_{(r)}(\mathbf{p});P) &= \oplus(\mathbf{q} \in N^{\#\mathbf{p}} \text{ of } ?n/\mathbf{q}_{(r)}/N(P\{\mathbf{p} \leftarrow \mathbf{q}\}) \\
/N(\pi_1;P_1 \oplus \dots \oplus \pi_n;P_n) &= /N(\pi_1;P_1) \oplus \dots \oplus /N(\pi_n;P_n) \\
/N(X_1(\mathbf{p}_1) | \dots | X_n(\mathbf{p}_n)) &= X_1/\mathbf{p}_1 | \dots | X_n/\mathbf{p}_n
\end{aligned}$$

**Table 9** Grounding

$$\begin{aligned}
E_G &= \{(X/\mathbf{q} = /N(M\{\mathbf{p} \leftarrow \mathbf{q}\})) \text{ s.t. } (X(\mathbf{p}) = M) \in E \text{ and } N = \text{fn}(E, P) \text{ and } \mathbf{q} \in N^{\#\mathbf{p}}\} \\
P_G &= /N(P) \quad \text{where } N = \text{fn}(E, P)
\end{aligned}$$

**Table 10** Parametric Explosion: From a CPF  $(E, P)$  to a CGF  $(E_G, P_G)$ 

Finally, we can convert a CPF to chemical reactions simply by first exploding it into a CGF, and then applying the previous  $Ch_\gamma$  procedure. See section 4.5 for an example.

$$Cp_\gamma(E, P) = Ch_\gamma(E_G, P_G)$$

**Table 11** From a Chemical Parametric Form  $(E, P)$  to a Chemical Reaction System  $Cp_\gamma(E, P)$ 

### 3.5 Iterative CPF to CGF Algorithm

A system  $Cp_\gamma(E, P)$  computed from  $E_G, P_G$  is highly redundant because it includes all the parameter permutation symmetries, many of which are not needed for any given set of initial conditions. The following iterative algorithm for the CPF case, combining definitions 7 and 10, computes a subset of  $E_G$  from the initial conditions of  $P_G$ . It produces a (usually) much smaller although not necessarily minimal set  $C$ . Again, see section 4.5 for an example.

The algorithm terminates:  $E_C$  never shrinks and is always a subset of  $E_G$ , which is finite.

<p><i>initialization</i> for a CPF <math>(E, P)</math></p> <p><math>N = \text{fn}(E, P)</math></p> <p><math>E_C := \{X/\mathbf{q} = /_N(M\{\mathbf{p} \leftarrow \mathbf{q}\})\}</math> s.t. <math>X(\mathbf{q})</math> occurs in <math>P</math> and <math>(X(\mathbf{p}) = M) \in E</math> (initial conditions)</p> <p><i>iteration</i></p> <p><math>C := \text{Ch}_\gamma(E_C)</math></p> <p><math>E'_C := E_C \cup X/\mathbf{q} = /_N(M\{\mathbf{p} \leftarrow \mathbf{q}\})</math> s.t. <math>X/\mathbf{q}</math> occurs in <math>C</math> and <math>(X(\mathbf{p}) = M) \in E</math></p> <p><i>termination</i></p> <p>if <math>E'_C = E_C</math> then stop and return <math>(C, /_N(P))</math>, else <math>E_C := E'_C</math> and iterate.</p>
---

**Table 12** Algorithm: Chemical Reaction System from CPF Initial Conditions

## 4 Examples

We illustrate the translations  $Pi_\gamma(-)$  from 4,  $Ch_\gamma(-)$  from 7 and  $Cp_\gamma(-)$  from 11, 12. There are natural issues about correctness of these translations, which are investigated in detail in [6]; the examples are provided to give some appreciation of the expected properties of the translations.

### 4.1 Unary Reactions

We begin with a degradation reaction that is not finite-control (because parallel splitting occurs) and is not finite-state (because the cardinality of  $X$  grows over time). However, the set of species is fixed ( $\{X\}$ ), so we can still carry out translations between processes and reactions.

Chemistry ( $C$ )	to Processes ( $Pi_\gamma(C)$ )	to Chemistry ( $Ch_\gamma(Pi_\gamma(C))$ )
$v : X \xrightarrow{r} X + X$	$X = \tau_{(r)} ; (X X)$	$v : X \xrightarrow{r} X + X$

Next is a similar unbounded-state system, but its size may grow or shrink depending on the rates  $r, s$ .

Chemistry ( $C$ )	to Processes ( $Pi_\gamma(C)$ )	to Chemistry ( $Ch_\gamma(Pi_\gamma(C))$ )
$v : X \xrightarrow{r} X + X$ $d : X \xrightarrow{s} 0$	$X = \tau_{(r)} ; (X X) \oplus \tau_{(s)} ; 0$	$v : X \xrightarrow{r} X + X$ $d : X \xrightarrow{s} 0$

### 4.2 Hetero Binary Reactions

The translation of reversible ionization reactions between  $Na$  and  $Cl$  is shown below. Note that a more natural version of  $Pi_\gamma(C)$  would map  $Na$  to  $Na^+$  and  $Cl$  to  $Cl^-$ , but that is not what the default translation produces. The back translation ( $Ch_\gamma(Pi_\gamma(C))$ ) yields the initial reactions (once retagged).

$C$	$Pi_\gamma(C)$	$Ch_\gamma(Pi_\gamma(C))$
$i: Na + Cl \xrightarrow{k} Na^+ + Cl^-$ $d: Na^+ + Cl^- \xrightarrow{v} Na + Cl$	$Na = ?i_{(k/\gamma)}; (Na^+   Cl^-)$ $Cl = !i_{(k/\gamma)}; 0$ $Na^+ = ?d_{(v/\gamma)}; (Na   Cl)$ $Cl^- = !d_{(v/\gamma)}; 0$	$i: Na + Cl \xrightarrow{k} Na^+ + Cl^-$ $d: Na^+ + Cl^- \xrightarrow{v} Na + Cl$

The next example starts from a two state process (from Section 2.3), and translates it to chemistry and back; the result is an equivalent but not identical process.

$E$	$Ch_\gamma(E)$ ( $= Ch_\gamma(Pi_\gamma(Ch_\gamma(E)))$ )	$Pi_\gamma(Ch_\gamma(E))$
$X = ?a_{(r)}; Y \oplus !b_{(s)}; X$ $Y = ?b_{(s)}; X \oplus !a_{(r)}; Y$	$a: X + Y \xrightarrow{r\gamma} Y + Y$ $b: Y + X \xrightarrow{s\gamma} X + X$	$X = ?a_{(r)}; (Y   Y) \oplus !b_{(s)}; 0$ $Y = !a_{(r)}; 0 \oplus ?b_{(s)}; (X   X)$

### 4.3 Homeo Binary Reactions

The inverse translation of a homeo chemical reaction gives back in the original reaction, and in particular it reproduces the original rate  $k = 2((k/\gamma)\gamma/2)$ .

$C$	$Pi_\gamma(C)$	$Ch_\gamma(Pi_\gamma(C))$
$v: X + X \xrightarrow{k} Y$	$X = ?v_{(k/\gamma)}; Y \oplus !v_{(k/\gamma)}; 0$ $Y = 0$	$v: X + X \xrightarrow{k} Y$

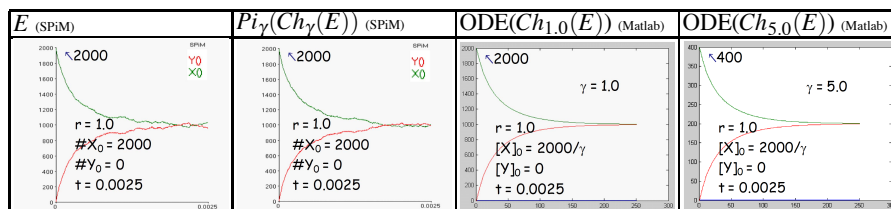
Conversely, starting from processes that self-interact, we produce homeo reactions, and then we go back again to equivalent but not identical processes.

$E$	$Ch_\gamma(E)$ (and $Ch_\gamma(Pi_\gamma(Ch_\gamma(E)))$ )	$Pi_\gamma(Ch_\gamma(E))$
$X = ?a_{(r)}; Y \oplus !a_{(r)}; X$ $Y = ?b_{(s)}; X \oplus !b_{(s)}; Y$	$a: X + X \xrightarrow{r\gamma} Y + X$ $b: Y + Y \xrightarrow{s\gamma} X + Y$	$X = ?a_{(r)}; (Y   X) \oplus !a_{(r)}; 0$ $Y = ?b_{(s)}; (X   Y) \oplus !b_{(s)}; 0$

### 4.4 Hetero and Homeo Reactions on a Shared Channel

This example involves both homeo and hetero reactions. We start with processes  $E$  and we obtain  $Ch_\gamma(E)$ , assigning unique reaction names  $b, c$  (this is a precondition for applying  $Pi_\gamma(-)$ ). The translation back,  $Pi_\gamma(Ch_\gamma(E))$ , produces different-looking processes, but both in  $E$  and in  $Pi_\gamma(Ch_\gamma(E))$ , the interaction of  $X$  with  $Y$  produces  $X|X$ , and the interaction of  $X$  with  $X$  produces  $Y|X$ . We show stochastic simulations of  $E$  and  $Pi_\gamma(Ch_\gamma(E))$ , and Matlab simulations of  $ODE(Ch_\gamma(E))$  for two values of  $\gamma$ .

$E$	$Ch_\gamma(E)$ (and $Ch_\gamma(Pi_\gamma(Ch_\gamma(E)))$ )	$Pi_\gamma(Ch_\gamma(E))$
$X = !a_{(r)}; X \oplus ?a_{(r)}; Y$ $Y = ?a_{(r)}; X$	$b: Y + X \xrightarrow{r\gamma} X + X$ $c: X + X \xrightarrow{r\gamma} Y + X$	$X = !b_{(r)}; 0 \oplus ?c_{(r)}; (Y   X) \oplus !c_{(r)}; 0$ $Y = ?b_{(r)}; (X   X)$



#### 4.5 A Parametric Example: Gene Networks

We compute the reactions for the parametric gate of Section 3.1, with initial conditions  $\text{Neg}(x, x)$ . Since this system  $E, P$  has a single free name,  $x$ , the parametric explosion does not actually increase its size, and we can easily show its expansion  $E_G, P_G$ . We could then compute  $Ch_\gamma(E_G)$  directly. For illustration, though, we convert  $E$  to chemical reactions  $E_C$  using the iterative algorithm of Section 3.5. With initial conditions  $\text{Tr}(x)|\text{Neg}(x, y)$  (not shown), with two free variables, the parametric explosion is larger, but the algorithm terminates in just two iterations with an output that is much smaller than  $E_G$ .

$E, P$ (Input) $\text{Neg}(a, b) = ?a(); \text{Inh}(a, b) \oplus \tau_\epsilon; (\text{Tr}(b) \text{Neg}(a, b))$ $\text{Inh}(a, b) = \tau_\eta; \text{Neg}(a, b)$ $\text{Tr}(b) = !b(); \text{Tr}(b) \oplus \tau_\delta; 0$ $\text{Neg}(x_{(r)}, x_{(r)}) \quad N = \text{fn}(E, P) = \{x_{(r)}\}$	$E_G, P_G$ (directly obtained, for comparison) $\text{Neg}/x, x = ?x/(r); \text{Inh}/x, x \oplus \tau_\epsilon; (\text{Tr}/x \text{Neg}/x, x)$ $\text{Inh}/x, x = \tau_\eta; \text{Neg}/x, x$ $\text{Tr}/x = !x/(r); \text{Tr}/x \oplus \tau_\delta; 0$ $\text{Neg}/x, x$
Iterative Algorithm for $E, P$	Initialization: $E_C$ $\text{Neg}/x, x = ?x/(r); \text{Inh}/x, x \oplus \tau_\epsilon; (\text{Tr}/x \text{Neg}/x, x)$
Iteration 1: $C$ $\text{Neg}/x, x \xrightarrow{\epsilon} \text{Tr}/x + \text{Neg}/x, x$	Iteration 1: $E_C$ $\text{Neg}/x, x = ?x/(r); \text{Inh}/x, x \oplus \tau_\epsilon; (\text{Tr}/x \text{Neg}/x, x)$ $\text{Tr}/x = !x/(r); \text{Tr}/x \oplus \tau_\delta; 0$
Iteration 2: $C$ $\text{Neg}/x, x \xrightarrow{\epsilon} \text{Tr}/x + \text{Neg}/x, x$ $\text{Tr}/x \xrightarrow{\delta} 0$ $\text{Tr}/x + \text{Neg}/x, x \xrightarrow{r\gamma} \text{Tr}/x + \text{Inh}/x, x$	Iteration 2: $E_C$ $\text{Neg}/x, x = ?x/(r); \text{Inh}/x, x \oplus \tau_\epsilon; (\text{Tr}/x \text{Neg}/x, x)$ $\text{Tr}/x = !x/(r); \text{Tr}/x \oplus \tau_\delta; 0$ $\text{Inh}/x, x = \tau_\eta; \text{Neg}/x, x$
Iteration 3: $C$ (Result) $\text{Neg}/x, x \xrightarrow{\epsilon} \text{Tr}/x + \text{Neg}/x, x$ $\text{Tr}/x \xrightarrow{\delta} 0$ $\text{Tr}/x + \text{Neg}/x, x \xrightarrow{r\gamma} \text{Tr}/x + \text{Inh}/x, x$ $\text{Inh}/x, x \xrightarrow{\eta} \text{Neg}/x, x$	Iteration 3: $E_C$ (Termination) no change Resulting initial conditions: $\text{Neg}/x, x$

In the context of this example we can discuss a basic question: if we can translate freely between processes and chemistry, why are processes better than chemistry? One answer, which is at the core of managing large models, is: consistent parameterization and modularization. With processes, we can describe a gate  $\text{Neg}(a, b)$  as a module with input  $a$  and output  $b$ . We can later connect several of these modules, and

even connect them in loops like  $\text{Neg}(x, x)$  without problems. In the case of chemical reactions, instead, we would have a “chemical module”  $\text{Neg}(A, B)$  consisting of the reactions involving the chemicals  $A, B$  and the gate. In particular, the module would contain degradation reactions for both species:  $A \rightarrow^\delta 0, B \rightarrow^\delta 0$ , because one possible system is simply  $\text{Neg}(A, B)$ . If we then want to connect several gates, we have to make copies of the module by appropriately instantiating the species  $A$  and  $B$ . But even  $\text{Neg}(A, A)$  goes wrong, because this creates two copies of the reaction  $A \rightarrow^\delta 0$ , resulting in a doubled degradation rate for  $A$ . This problem does not occur with the processes as defined above (and does not even require any planning), because each molecule is a process that knows how to degrade. Parameterized chemical reactions are not as good as parameterized processes.

#### 4.6 Processes to ODEs: The Kermack-McKendrick Model of Epidemics

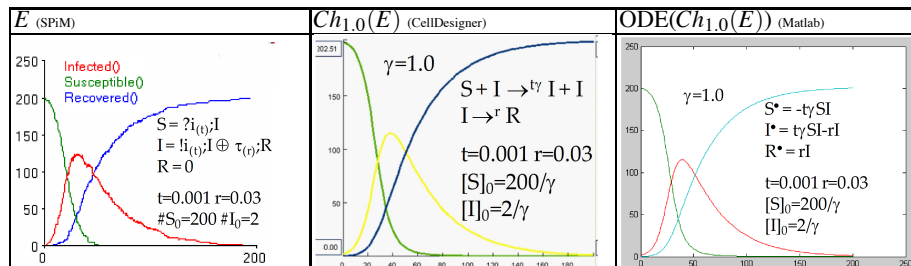
This example, and the next one, are not about chemistry, but they describe interactions governed by laws equivalent to the law of mass action (“chance of collision”), and use reactions to derive the ODEs.

$E$	$Ch_\gamma(E)$	$\text{ODE}(Ch_\gamma(E))$
$S = ?i_{(t)}; I$	$v_1 : S + I \rightarrow^{t\gamma} I + I$	$[S]^\bullet = -t\gamma[S][I] \quad (v_1)$
$I = !i_{(t)}; I \oplus ?i_{(t)}; I \oplus \tau_{(r)}; R$	$v_2 : I + I \rightarrow^{t\gamma} I + I$	$[I]^\bullet = t\gamma[S][I] - r[I] \quad (v_1, v_3)$
$R = ?i_{(t)}; R$	$v_3 : I \xrightarrow{r} R$	$[R]^\bullet = r[I] \quad (v_3)$
	$v_4 : R + I \xrightarrow{t\gamma} R + I$	

In this *SIR* model, inspired by [22], we map out the behavior of individuals during an epidemic. A *Susceptible* individual may become *Infected* by interaction with an *Infected* at rate  $t$ . A *Recovered* may be infected (with no effect). An *Infected* may spontaneously become *Recovered* at rate  $r$ , or may infect a *Susceptible*, or a *Recovered*, or another *Infected* (with no effect).

Although we start with an intuitive process-oriented description of individual behavior, the resulting ODE system is exactly the Kermack-McKendrick population model [19]. Moreover, we may notice that reactions  $v_2$  and  $v_4$  do not contribute to the ODE translation. This suggests that the process model can in fact be simplified to  $S = ?i_{(t)}; I, I = !i_{(t)}; I \oplus \tau_{(r)}; R, R = 0$ , which produces only  $(v_1, v_3)$  and hence the same ODE system. Below we run the processes with SPiM [23], the chemical reactions with CellDesigner (which converts them to ODEs) [9], and the ODEs with Matlab [28].



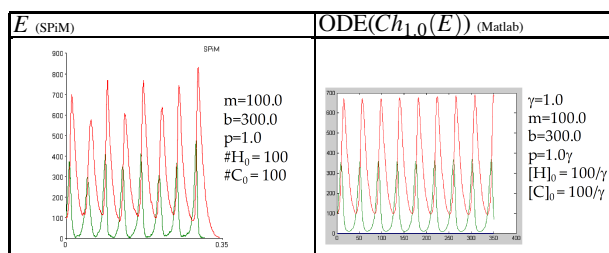


### 4.7 Unbounded Processes to ODEs: The Lotka-Volterra Model of Predation

In the Lotka-Volterra model of predation, prey ( $H$ ) can breed without bounds, but is culled by predators ( $C$ ), which can reproduce by predation, but have a regular mortality rate.

$E$	$Ch_{\gamma}(E)$	$ODE(Ch_{\gamma}(E))$
$H = \tau_{(b)}; (H H) \oplus ?c_{(p)}; 0$ $C = \tau_{(m)}; 0 \oplus !c_{(p)}; (C C)$	$v_1 : C \xrightarrow{m} 0$ $v_2 : H \xrightarrow{b} H + H$ $v_3 : H + C \xrightarrow{p\gamma} C + C$	$[H]^{\bullet} = b[H] - p\gamma[H][C] \quad (v_2, v_3)$ $[C]^{\bullet} = -m[C] + p\gamma[H][C] \quad (v_1, v_3)$

The resulting ODEs are the Lotka-Volterra equations [2], for the case where the rates at which prey decrease and predators increase are equal ( $p$ ). Different ratios can be modeled (e.g., a normal form of:  $C = \tau_{(m)}; 0 \oplus !c_{(p)}; !c_{(p)}; !c_{(p)}; (C|C)$ ), but such a model of sequential predation no longer corresponds exactly to the original Lotka-Volterra. A SPiM stochastic simulation of  $E$  is shown: it quickly leads to extinction, unlike the ODE model that with the same parameters oscillates indefinitely.



### 4.8 Process Equivalences from ODEs: The Markovian Interleaving Law

The translation of processes to ODEs can be regarded as a semantics of processes, and in particular it induces an equivalence over processes that can be used to derive alge-

braic laws. The following equivalence, an interleaving law for concurrent degradations, is derived in [12], section 4.1.2, for continuous-time Markov chains:

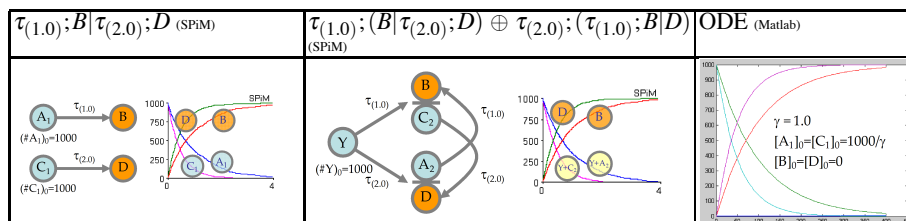
$$\tau(\lambda);B|\tau(\mu);D = \tau(\lambda);(B|\tau(\mu);D) \oplus \tau(\mu);(\tau(\lambda);B|D)$$

We now derive this law as an equivalence of ODE systems. We set up a separate CGF for the left hand side ( $X$ ) and right hand side ( $Y$ ), with initially the same number  $n$  of top-level processes  $X$  and  $Y$ , and we compute their respective chemical reactions and ODEs. The factor  $\gamma$  here does not appear (except in the initial conditions) because all the reactions are exponential decays:

Left hand side: $X = \tau(\lambda);B \tau(\mu);D$		Right hand side: $Y = \tau(\lambda);(B \tau(\mu);D) \oplus \tau(\mu);(\tau(\lambda);B D)$	
$A_1 = \tau(\lambda);B$ $C_1 = \tau(\mu);D$ $n \times A_1   n \times C_1$	The CGF $E_X, P_X$ (Sec 2.3) ← initial conditions	$Y = \tau(\lambda);(B C_2) \oplus \tau(\mu);(A_2 D)$ $C_2 = \tau(\mu);D$ $A_2 = \tau(\lambda);B$ $n \times Y$	The CGF $E_Y, P_Y$ ← initial conditions
$A_1 \xrightarrow{\lambda} B$ $C_1 \xrightarrow{\mu} D$ $[A_1]_0 = n/\gamma$ $[C_1]_0 = n/\gamma$	$Ch_\gamma(E_X, P_X)$ : (Sec 2)	$Y \xrightarrow{\lambda} B + C_2$ $Y \xrightarrow{\mu} A_2 + D$ $C_2 \xrightarrow{\mu} D$ $A_2 \xrightarrow{\lambda} B$ $[Y]_0 = n/\gamma$	$Ch_\gamma(E_Y, P_Y)$
$[A_1]^\bullet = -\lambda[A_1]$ $[B]^\bullet = \lambda[A_1]$ $[C_1]^\bullet = -\mu[C_1]$ $[D]^\bullet = \mu[C_1]$	ODE for $Ch_\gamma(E_X)$ (Sec 3.2)	$[Y]^\bullet = -\lambda[Y] - \mu[Y]$ $[A_2]^\bullet = \mu[Y] - \lambda[A_2]$ $[B]^\bullet = \lambda[Y] + \lambda[A_2]$ $[C_2]^\bullet = \lambda[Y] - \mu[C_2]$ $[D]^\bullet = \mu[Y] + \mu[C_2]$	ODE for $Ch_\gamma(E_Y)$
		$[Y + A_2]^\bullet = -\lambda[Y + A_2]$ $[B]^\bullet = \lambda[Y + A_2]$ $[Y + C_2]^\bullet = -\mu[Y + C_2]$ $[D]^\bullet = \mu[Y + C_2]$	Derived ODE

The final ODE on the right is derived from the one above it, because  $[Y + A_2]^\bullet = [Y]^\bullet + [A_2]^\bullet = (-\lambda[Y] - \mu[Y]) + (\mu[Y] - \lambda[A_2]) = -\lambda[Y] - \lambda[A_2] = -\lambda[Y + A_2]$ , and  $[B]^\bullet = \lambda[Y] + \lambda[A_2] = \lambda[Y + A_2]$ .

Comparing the final ODEs for  $E_X$  and  $E_Y$ , we see that the quantities  $[B]$  and  $[D]$  are identically related up to a change of variables  $[A_1] = [Y + A_2]$  and  $[C_1] = [Y + C_2]$ . That is,  $[B]$  and  $[D]$  have equal time evolutions on the two sides provided that  $[A_1] = [Y + A_2]$  and  $[C_1] = [Y + C_2]$ . Moreover we have that  $[A_1]_0 = [C_1]_0 = [Y]_0 = n/\gamma$ , and the initial conditions of the right hand system specify that  $[A_2]_0 = [C_2]_0 = 0$  (since only  $Y$  is present), so that  $[A_1]_0 = [Y + A_2]_0$  and  $[C_1]_0 = [Y + C_2]_0$ . Similarly  $[B]_0 = [D]_0 = 0$ . Therefore the final ODEs also have the same initial conditions for all variables, and hence have the same time evolution. For example, if we run a stochastic simulation of the left hand side with  $n = 1000$  and with initially  $1000 \times A_1$  and  $1000 \times C_1$ , we obtain the same curves for  $B$  and  $D$  than a simulation of the right hand side with initially  $1000 \times Y$ . The figure below illustrates the case with rates  $\lambda = 1.0$ ,  $\mu = 2.0$ .



## 5 Related Work

A direct translation from a stochastic process algebra (PEPA) to ODEs is presented in [5,16]. The rate equation and the *activity matrix* techniques presented there, are similar to the stoichiometric techniques from chemistry (section 2). Differences in apparent rate computations between PEPA and the law of mass action can be easily adapted either way.

A main difference between our CGF and PEPA, however, is our ability to represent chemical reactions of the form  $A \rightarrow^r B + C$ : these reactions require a process to “split” in two. PEPA, instead, is intentionally restricted to the composition of purely sequential processes, to enable Markov chain analysis by linear algebra. At the level of ODEs, this ability allows us, for example, to give a process model corresponding to the Lotka-Volterra equations, which are based on an unbounded growth of prey ( $H \rightarrow^b H + H$ , example 4.7). In addition, the translation of parametric processes (CPF) does not appear to have been considered before, and this is very useful for parameterizing and modularizing models of even modest size [1].

## 6 Conclusions

We have shown how to go from a subset of  $\pi$ -calculus to ODEs, and how to use chemistry as a stepping stone to simplify the translation, avoiding a direct expression of a rate equation for processes [6]. A mapping to ODEs is necessary to validate and compare models of biochemistry written in process calculi, with respect to the wider and deeper literature of chemical and ODE models.

An advantage of process algebra modeling shines through from this analysis: compositionality. Parametric models can be written in process calculi without reference to initial conditions (e.g., a library of gene gates, as in example 4.5), and then reused without change by adding specific initial conditions (e.g., by wiring gene gates to produce specific gene networks). In contrast, under translations such as the one shown here, any given network expands to a different set of chemical or differential equations. The resulting model consists of a large “flat” set of equations that unrolls the state space (as shown in [6]), and that may oddly relate in differential form entities that actually only exist in discrete quantity (the genes of example 4.5). Both the network

structure and the discrete character of the components are lost in such a translation. Still, useful analysis can often be obtained from a translation to differential form, as well as comparison with ODE models from the literature.

## 7 References

1. R. Blossey, L. Cardelli, A. Phillips: A Compositional Approach to the Stochastic Dynamics of Gene Networks. *Transactions on Computational Systems Biology IV*, LNCS 3939, 99-122. Springer 2006.
2. W.E. Boyce, R.C. DiPrima. *Elementary Differential Equations and Boundary Value Problems*, 5th ed. New York: Wiley, p. 494, 1992.
3. M. Bernardo, L. Donatiello, R. Gorrieri. *MPA: a Stochastic Process Algebra*. Technical Report UBLCS-94-10, University of Bologna, Laboratory of Computer Science. 1994
4. P. Buchholz. *Markovian Process Algebra: Composition and Equivalence*. In Proc. PAPM '94, Erlangen (Germany), 11-30, 1994.
5. M. Calder, S. Gilmore, J. Hillston: *Automatically Deriving ODEs from Process Algebra Models of Signalling Pathways*. Proc. *Computational Methods in Systems Biology 2005*, pp 204-215.
6. L. Cardelli: *On Process Rate Semantics*. *Theoretical Computer Science* 391(3) 190-215, Elsevier, 2008. DOI: <<http://dx.doi.org/10.1016/j.tcs.2007.11.012>>.
7. N. Chabrier, M. Chiaverini, V. Danos, F. Fages and V. Schchter. *Modeling and querying biomolecular interaction networks*. *Theoretical Computer Science*, 2004.
8. M. Dam: *On the Decidability of Process Equivalences for the pi-Calculus*. *Theoretical Computer Science* 183, 215-228, 1997.
9. A. Funahashi, N. Tanimura, M. Morohashi, H. Kitano. *CellDesigner: a process diagram editor for gene-regulatory and biochemical networks*, *BIOSILICO*, 1:159-162, 2003.
10. D.T. Gillespie. *Exact Stochastic Simulation of Coupled Chemical Reactions*. *Journal of Physical Chemistry* 81, 2340-2361. 1977.
11. D.T. Gillespie: *The chemical Langevin equation*. *Journal of Chemical Physics* 113(1), 297-306, 2000.
12. H. Hermanns: *Interactive Markov Chains*. Springer LNCS, vol 2428, 2002.
13. H. Hermanns, M. Rettelbach. *Syntax, Semantics, Equivalences, and Axioms for MTIPP*. Proc. of PAPM '94, Erlangen (Germany), pp 71-87. 1994.
14. N. Gtz, H. Hermanns, U. Herzog, V. Mertsotakis, M. Rettelbach. *Stochastic Process Algebras: Constructive Specification Techniques Integrating Functional, Performance and Dependability*. In Baccelli and Mitrani (eds): *Quantitative Modelling in Parallel Systems*. Chapter 1, Springer 1995.
15. J. Hillston. *A compositional approach to performance modelling*. Cambridge University Press, 1996.

16. J. Hillston: Fluid flow approximation of PEPA models. In Proceedings of the Second International Conference on the Quantitative Evaluation of Systems, 33-43. IEEE Press, 2005.
17. A. Hinton, M. Kwiatkowska, G. Norman, D. Parker. PRISM: A Tool for Automatic Verification of Probabilistic Systems. In H. Hermanns, J. Palsberg (Eds.): Proc. TACAS'06. Springer LNCS 3920, 441-444, 2006.
18. F. Horn, R. Jackson. General mass action kinetics. Arch. Rational Mech. Anal. 47, 81-116, 1972.
19. W.O. Kermack, A.G. McKendrick: A Contribution to the Mathematical Theory of Epidemics. Proc. Roy. Soc. Lond. A 115, 700-721, 1927.
20. H. Kitano: A graphical notation for biochemical networks. BioSilico 1(5): 169-76. 2003.
21. R. Milner: Communicating and Mobile Systems: The p-Calculus. Cambridge University Press, 1999.
22. R. Norman, C. Shankland. Developing the Use of Process Algebra in the Derivation and Analysis of Mathematical Models of Infectious Disease. Proc. Computer Aided Systems Theory - EUROCAST 2003. Springer LNCS 2809, 404-414, 2003.
23. A. Phillips, L. Cardelli: A Correct Abstract Machine for the Stochastic Pi-calculus. Proc. BioConcur'04.
24. C. Priami. Stochastic p-calculus. The Computer Journal, 38, 578-589, 1995.
25. C. Priami, A. Regev, E. Shapiro, W. Silverman: Application of a stochastic name-passing calculus to representation and simulation of molecular processes. Information Processing Letters 80, 25-31. 2001.
26. A. Regev. Computational systems biology: a calculus for biomolecular knowledge. Ph.D. Thesis, Tel Aviv University, 2002.
27. A. Regev, E. Shapiro. Cellular abstractions: Cells as computation. Nature 419 343. 2002.
28. The Mathworks: Matlab. <http://www.mathworks.com>.
29. J.M.G. Vilar, H.Y. Kueh, N. Barkai, S. Leibler: Mechanisms of noise-resistance in genetic oscillators. PNAS 99(9) 5988-5992. 2002.
30. O. Wolkenhauer, M. Ullah, W. Kolch, K. Cho. Modelling and simulation of intracellular dynamics: Choosing an appropriate framework. IEEE Transactions on NanoBioscience 3, 200-207. 2004.