Coloured Stochastic Multilevel Multiset Rewriting

Nicolas Oury
LFCS, School of Informatics
University of Edinburgh
nicolas.oury@gmail.com

Gordon D. Plotkin
LFCS, School of Informatics
University of Edinburgh
gdp@inf.ed.ac.uk

ABSTRACT

From the phosphorylation state of a molecule to the volume of a cell, parameters are ubiquitous in systems biology. At the same time, most models involve static or dynamic compartments, for example to separate cells from their environment. We introduce coloured stochastic multilevel multiset rewriting, an expressive formalism for modelling systems with parameters and complex dynamic, multilevel compartment structures, and an extension of both multilevel multiset rewriting and coloured Petri nets. While being very expressive, it allows the direct and natural expression of biological ideas. We give some illustrative examples, demonstrating the use of parameters to handle cell states, position and volume, and variable rates. We further demonstrate the use of rules with complex right-hand sides for reproduction. We regard these examples as paving the way for more biologically relevant models.

Categories and Subject Descriptors

J.3 [**Life and Medical Sciences**]: Biology and genetics; D.3.2 [**Language Classifications**]: Specialized application languages; G.3 [**Probability and Statistics**]: Stochastic processes

Keywords

Term rewriting, stochastic, multilevel

1. INTRODUCTION

In previous work we introduced a rule-based stochastic formalism, *stochastic multilevel multiset rewriting (SMMR)* [OP11]. This can be viewed as providing a system of terms for Milner's place graphs (a half of his bigraphs [Mil09, KMT08]) and it is also very close to the Calculus of Wrapped Compartments (CWC) of Coppo et al [CD10a, CD10b].

The terms of the rules are normal forms for the theory of a commutative monoid, with sets of constants and unary function symbols; this presents the formalism algebraically, as stochastic associative-commutative (AC) term rewriting. As shown below, one obtains a very natural rendition of various biological situations, using constants to model species and unary function symbols to model agents (their arguments model the agents' contents). However, as will also be seen, there is a need for parameters (a.k.a. colours) and that is the main concern of this paper: we present *coloured stochastic multilevel multiset rewriting (CSMMR)*.

SMMR includes the usual multiset rendition of chemical reactions, such as

$$\text{Glc}, \text{ATP} \xrightarrow{k} \text{Glc-6-P}, \text{ADP}$$

where Glc, for example, is a species constant (modelling glucose), and k is a stochastic rate. SMMR also permits hierarchical multisets, with rules such as the following for modelling viral infection:

$$\text{Virus}, \text{Cell}(x) \xrightarrow{k} \text{ICell}(x, \text{gen}, 80\text{V}_1, 40\text{V}_2)$$

where Cell and ICell are agent function symbols. When applying the rule, the variable "x" is matched against the current population of the matched cell, so that its contents are also present in the newly infected cell.

Multisets can be at the top level or within agents such as Cell. The levels can be continued down as far as one wishes using agents within agents, and rules can be applied at all levels. SMMR allows general forms of rules for transport across agent boundaries, for agent creation, and for agent destruction:

Transport

$$M, A(N, x) \xrightarrow{k} M', A(N', x)$$

Creation

$$M \xrightarrow{k} M', A(N', x)$$

Destruction

$$M, A(N, x) \xrightarrow{k} M', N'$$

where M, N, etc, are multisets of species.

That said, SMMR has several limitations. It deals only awkwardly with cell type, with cell position, or with population splitting in cell reproduction [OP11]. It does not allow rate functions depending, for example, on cell volume. It further does not allow rules applicable only if some condition is satisfied, for example, a condition on cell type. (The

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

formalism also has no provision for complexes, other than as atomic species; but no such provision is made here either.)

To deal with these limitations we extend SMMR, principally by allowing species and agents to be parameterised. For example if we have two cell types "infected" and "uninfected" we might instead write the above rule as:

$$\mathrm{Virus}, \mathrm{Cell}_{\mathrm{type:uninf}}(x) \xrightarrow{k} \mathrm{Cell}_{\mathrm{type:inf}}(x, \mathrm{gen}, 80\mathrm{V}_1, 40\mathrm{V}_2)$$

Note that parameters are written using labelled subscripts.

Here is an example where the rate of a reaction inside a cell depends on its volume:

$$\begin{array}{c}\mathrm{Cell}_{\mathrm{vol:}v}(x, \mathrm{Glucose}, \mathrm{ATP}) \xrightarrow{kv^{-1}} \\ \mathrm{Cell}_{\mathrm{vol:}v}(x, \mathrm{Gluc\text{-}6\text{-}P}, \mathrm{ADP}) \quad (v > 0)\end{array}$$

Note the *precondition* $v > 0$, which avoids division by zero.

One can also write natural rules for movement, such as

$$\mathrm{LIGAND}_{\mathrm{x\text{-}cell:}x,\mathrm{y\text{-}cell:}y} \xrightarrow{k} \mathrm{LIGAND}_{\mathrm{x\text{-}cell:}x,\mathrm{y\text{-}cell:}y+1}$$

where space has been divided up into cells, given by x and y coordinates.

Parameters also help with species as they can be used to give site modifications. Here is an example taken from a MAPK cascade:

$$\mathrm{RAF}_{p_1:\mathrm{true}}, \mathrm{MEK}_{p_1:x,p_2:\mathrm{false}} \xrightarrow{k} \mathrm{RAF}_{p_1:\mathrm{true}}, \mathrm{MEK}_{p_1:\mathrm{true},p_2:x}$$

Note that the use of a boolean parameter enables the condensation of two rules into one.

As chemical reactions correspond to P/T nets (taking places to correspond with species, and transitions with reactions), parameters correspond to the colours of coloured Petri nets [Jen92, Jen94]. Coloured Petri nets have been used to model biological processes [Run04, TM06, LH10]; this paper can be viewed as extending the idea of colours to all levels.

Section 2 presents an extended example modelling evolutionary competition between different types of cells with competing strategies for survival in their common environment.[1] The particular model chosen is intended to illustrate some of the uses of parameters in coloured stochastic multilevel multiset rewriting and to show a construct for population-splitting. This construct could well equally have been added to SMMR: colouring is an orthogonal matter to population splitting.

CSMMR is presented in detail in Section 4, following a brief section on technical preliminaries. Section 5 gives a brief discussion of algorithmic aspects of our implementation of CSMMR. Some more extended examples, including a simplified bacterial chemotaxis model are presented in Section 6. Finally, in Section 7, we discuss possible further work and make some general remarks on adding parameterisation to rule-based systems: we regard the work presented here as illustrative of a general methodology for such additions.

As regards comparison with other formalisms for dynamic compartments, ours is rule-based and offers a system of terms that is uncommitted yet, we feel, enables a direct natural expression of cells and their sub-compartments; the closest formalism is CWC, whose term system, although essentially equivalent (see [OP11]), is, rather, oriented to the expression of membrane systems. As far as we know, coloured Petri nets provide the only comparable formalism with a general means of handling parameters; they are equivalent to the special case of ours with species, but no agents. They can handle static systems of cells and compartments by using colours for the different levels; however they seem less adapted to handle dynamically changing systems.

[1] We are grateful to Peter Swain and Andrea Weiße who suggested this idea (personal communication).

2. AN EXAMPLE: EVOLUTIONARY COMPETITION

We now sketch a CSMMR model of evolutionary competition between two cell types of competing genetic lineages. Multilevel modelling, particularly two-level modelling, seems the minimum needed to model evolutionary competition: one needs different organisms, with their different genomes, competing within a common environment.

The two cell types we consider have different strategies for consuming the sugar glucose from the environment: "fast" and "slow," resulting in different energy levels, as measured by ATP, and so different survival rates. Fast cells eat (meaning consume glucose) faster than slow cells, but have to pay a higher energetic price to start eating.

The eating strategies of our model are not biologically relevant, although we have been inspired by the biological literature, for example [PSB01]. Our aim here is rather to illustrate the use of CSMMR while demonstrating the possibility of modelling evolutionary competition.

We introduce an agent Cell to represent cells. The expressivity of CSMMR arises from the ability to store parameters in each species or agent. Each Cell is parameterised by

- the Cell type : a boolean with 0 and 1, denoted by slow and fast, representing *slow-* and *fast-eating* cells;
- the Cell state: a boolean indicating whether the cell has been recently eating or not; it prevents the cell restarting an eating process too quickly; and
- the Cell ATP level: a natural number giving the ATP population of the cell.

The first two parameters are phenomenological rather than mechanistic: the type would correspond to some genes, and the state would correspond to the population level of certain proteins in a feedback mechanism involving these genes. Modelling ATP as a state rather than as a species inside cells lets us illustrate some CSMMR parameter-handling features.

While we generally take a quite abstract view of the cell, we model the absorption of sugar in more detail; we hope thereby to illustrate the ability of CSMMR to prototype systems, progressively supplying greater mechanistic detail. We use the species Glc to represent glucose, and summarise the proteins involved in the absorption of glucose by a fictitious single generic glucose metabolic pathway enzyme GlcP.

The sugar absorption rule is

$$\begin{array}{c}\mathrm{Glc}, \mathrm{Cell}_{\mathrm{type:}l,\mathrm{state:}s,\mathrm{ATP:}a}(\mathrm{GlcP}, x) \\ \xrightarrow{k_{abs}} \mathrm{Cell}_{\mathrm{type:}l,\mathrm{state:}s,\mathrm{ATP:}a+20}(\mathrm{GlcP}, x)\end{array}$$

It expresses that, with the help of GlcP, cells can convert glucose into energy.

GlcP is produced by cells when they detect Glc in the environment. The quantity produced depends on their strategy:

- *fast-eating* cells produce 6 molecules of GlcP, at a cost of 45 molecules of ATP; but

- *slow-eating* cells produce 2 molecules of GlcP, at a cost of 10 molecules of ATP.

This is expressed by the following two "GlcP rules":

$$\mathrm{Glc}, \mathrm{Cell}_{\mathrm{type:fast,state:0,ATP:}a}(x) \xrightarrow{k_{start}} \mathrm{Glc}, \mathrm{Cell}_{\mathrm{type:fast,state:1,ATP:}a \,\dot{-}\, 45}(6\mathrm{GlcP}, x) \quad (a > 45)$$

$$\mathrm{Glc}, \mathrm{Cell}_{\mathrm{type:slow,state:0,ATP:}a}(x) \xrightarrow{k_{start}} \mathrm{Glc}, \mathrm{Cell}_{\mathrm{type:slow,state:1,ATP:}a \,\dot{-}\, 10}(2\mathrm{GlcP}, x) \quad (a > 10)$$

Note the use of the preconditions to ensure that the rules cannot be applied unless the cell has sufficient energy. They also cannot be applied unless the state is 0; they then change the state to 1 so that they cannot be reapplied.

After some time, cells can start eating again:

$$\mathrm{Cell}_{\mathrm{type:}l,\mathrm{state:1,ATP:}a}(x) \xrightarrow{k_{reinit}} \mathrm{Cell}_{\mathrm{type:}l,\mathrm{state:0,ATP:}a}(x)$$

This is obviously not realistic. It is a very simple abstraction of a negative feedback loop.

Cells consumes energy and GlcP degrades:

$$\mathrm{Cell}_{\mathrm{type:}l,\mathrm{state:}s,\mathrm{ATP:}a}(x) \xrightarrow{k_{e^-}} \mathrm{Cell}_{\mathrm{type:}l,\mathrm{state:}s,\mathrm{ATP:}a \,\dot{-}\, 1}(x) \quad (a \geq 1)$$

$$\mathrm{Cell}_{\mathrm{type:}l,\mathrm{state:}s,\mathrm{ATP:}a}(\mathrm{GlcP}, x) \xrightarrow{k_{deg}} \mathrm{Cell}_{\mathrm{type:}l,\mathrm{state:}s,\mathrm{ATP:}a}(x)$$

When cells have too little energy, they die:

$$\mathrm{Cell}_{\mathrm{type:}l,\mathrm{state:}s,\mathrm{ATP:}a}(x) \xrightarrow{k_{death}} \qquad (a = 0)$$

When cells have stocked enough energy, they divide in two; each of their two child cells share their parent's energy evenly and the remaining GlcPs randomly:

$$\mathrm{Cell}_{\mathrm{type:}l,\mathrm{state:}s,\mathrm{ATP:}a}(x) \xrightarrow{k_{\mathrm{div}}(a)} \mathit{let}\ a_{1/2} = \mathrm{halfEnergy}(\mathrm{a} \,\dot{-}\, 100)\ \mathit{in}\ \mathit{let}\ (y, z) = x\ \mathit{in}\ \mathrm{Cell}_{\mathrm{type:}l,\mathrm{state:0,ATP:}a_{1/2}}(y), \mathrm{Cell}_{\mathrm{type:}l,\mathrm{state:0,ATP:}a \,\dot{-}\, a_{1/2}}(z) \quad (a \geq 100)$$

This rule introduces three new constructs:

- $k_{\mathrm{div}} : \mathrm{nat} \to \mathrm{real}$ is a *rate function*.
- *let* $x = t$ *in* ... is a binding construct. The variable x is bound in ... to an element chosen randomly from a probability distribution, given by t.
- *let* $(y, z) = x$ *in* ... is a binding construct. The contents of the variable x are split randomly between the variables y and z of ....

The rate function k_{div} enables the division rate of a cell to depend on its energy level; it is chosen to be a step function, 0 below 300, 5 thereafter. The term halfEnergy(a) denotes the distribution on natural numbers taking value $b(k; a, 1/2)$ at k, where $b(k; n, p)$ is the *binomial probability function*, which gives the probability of getting exactly k successes in n trials, where each trial has probability p of success.

We have finished our model of cells, but still need to model the environment. We use an agent Env parameterised by a boolean state, where:

- in state 1, sugar is regularly introduced into the system; and
- in state 0, no sugar is introduced into the system.

$$\mathrm{Env}_{\mathrm{state:1}}(x) \xrightarrow{k_{sug}} \mathrm{Env}_{\mathrm{state:1}}(50\ \mathrm{Glc}, x)$$

From time to time, the state of the environment changes:

$$\mathrm{Env}_{\mathrm{state:}s}(x) \xrightarrow{k_{env}} \mathrm{Env}_{\mathrm{state:}\neg s}(x)$$

We have chosen to give a very simple model to illustrate the features of CSMMR. We could make the model more precise by adding more species (or even agents to represent the nucleus or different vesicles). We could also choose to represent the spatiality of the competition, attaching the cells, and the sugar to different locations. Section 6 shows how to use colours to represent geometry, space and diffusion.

Figure 1 shows a sample run of a simulation of the model described here, starting with one fast-eating cell and one slow-eating cell. System states are modelled by multilevel multisets, with the initial one being:

$$\mathrm{Env}_{\mathrm{state:0}}(\mathrm{Cell}_{\mathrm{type:fast,state:0,ATP:100}}(), \mathrm{Cell}_{\mathrm{type:slow,state:0,ATP:100}}())$$

We used the following rates:

k_{abs}	2.5	k_{start}	100
k_{reinit}	0.03	k_{e^-}	0.17
k_{deg}	0.2	k_{death}	1000000
k_{sug}	0.1	k_{env}	0.01

The figure shows that when the environment is favourable, with sugar abundance and a low population, the fast-eating cells do very well. However their behaviour is too expensive when there is sugar scarcity and a high population.

While the current model is not biologically realistic, it does show the possibility of using CSMMR to model evolutionary competition.

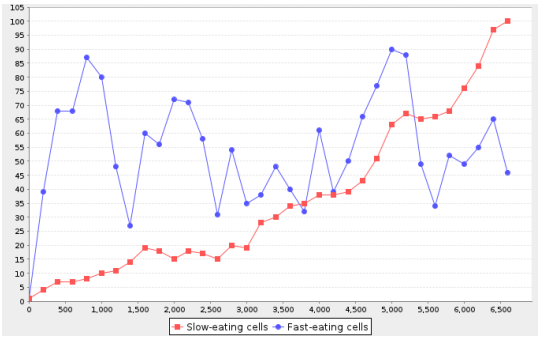

Figure 1: An evolutionary competition simulation

3. TECHNICAL PRELIMINARIES

We write $[b_1/a_1, \ldots, b_n/a_n]$ for finite functions sending a_i to b_i (for $i = 1, n$); we write $\mathrm{dom}(f)$ for the domain of a function. We write $f \frown g$ for the function with domain $\mathrm{dom}(f) \cup \mathrm{dom}(g)$, equal to g on $\mathrm{dom}(g)$ and to f elsewhere.

Multisets over a set X are considered to be functions from X to the natural numbers; with this representation, multiset union is pointwise addition, and we also use the pointwise versions of other operations and of $\leq$. We write $|m|$ for the *support* of a multiset m, i.e., the set of its elements (those $a \in X$ with $m(a) \neq 0$). We only consider finite multisets,

i.e., those with finite support. We identify elements of X with their corresponding singleton multisets; we sometimes present multisets as a list, for example $a_1, \ldots, a_n$; and we sum over finite multisets, writing $\sum_{a \in a_1,\ldots,a_n, P(a)} f(a)$ for $\sum_{i=1}^{n} b_i$, where b_i is $f(a_i)$ if $P(a_i)$ holds and 0, otherwise (if omitted $P(a)$ is taken to be the universally true predicate).

4. COLOURED STOCHASTIC MULTI-LEVEL MULTISET REWRITING

We provide a formal definition of CSMMR and its stochastic semantics. CSMMR is parameterised on the datatypes available for parameterisation. We model these by a choice of first-order multisorted signature Σ (the *parameter signature*) and interpretation $\mathcal{A}$ (the *parameter interpretation*). By way of orientation, the interpretations of sorts correspond to the colour sets of coloured Petri nets.

Any implementation of CSMMR would have to provide a means of defining the signature and interpretation; this is typically done via some formalism for enumeration, when the colour sets are finite, and/or providing types for the sorts and programs for the needed functions and relations.

A multi-sorted signature Σ consists of: a set Sort of *sorts* ranged over by s; a set of *function symbols* ranged over by f, each with an arity (u, s') where $u \in \mathrm{Sort}^*, s' \in \mathrm{Sort}$; and a set of *relation symbols* ranged over by P each with an arity u, where $u \in \mathrm{Sort}^*$ (we always include the equality relation symbols). We write $f : u \to s$ and $P : u$ to show the arity of function and relation symbols. *Constants* are function symbols of sort $c : \varepsilon \to s'$. We assume a distinguished sort real (to be used for rate functions).

Given such a signature, we have available standard notions of terms t of sort s (written $t : s$) and first-order formulas φ, where an infinite supply of (parameter) variables x^s of sort s is assumed available, for each sort s. We may call these terms *parameter* terms.

An *interpretation* $\mathcal{A}$ of such a signature consists of: a set $s_{\mathcal{A}}$ for each $s \in \mathrm{Sort}$; a function $f_{\mathcal{A}} : u_{\mathcal{A}} \to s_{\mathcal{A}}$, for each function symbol $f : u \to s$ (where $(s_1 \ldots s_n)_{\mathcal{A}} =_{\mathrm{def}} \prod_{i=1}^{n} (s_i)_{\mathcal{A}}$); and a subset $P_{\mathcal{A}} \subseteq u_{\mathcal{A}}$, for each relation symbol $P : u$. We assume $\mathrm{real}_{\mathcal{A}} \subseteq \mathbb{R}_+$, the set of non-negative reals (not insisting on equality allows computationally useful possibilities).

Given such an interpretation $\mathcal{A}$, we have available the standard interpretation $t^{\rho}_{\mathcal{A}} \in s_{\mathcal{A}}$ of terms $t : s$ (respectively, notion of satisfaction $\mathcal{A} \models^{\rho} \varphi$), for any *assignment* ρ, i.e., a finite map on parameter variables assigning elements of $s_{\mathcal{A}}$ to variables of sort s, whose domain includes the variables occurring in t (respectively, occurring freely in φ). We assume available a constant $c_a : \varepsilon \to s$ for each $s \in S, a \in s_{\mathcal{A}}$ such that $a = c_a()^{[\,]}_{\mathcal{A}}$, and identify a and $c_a()$.

The evolutionary competition model provides a running example. The sorts are bool, nat, and real, interpreted by $\mathbb{B}$, $\mathbb{N}$, and, say, $\mathbb{Q}_+$, the non-negative rationals. Constants include $0, 1, \ldots$ and fast and slow; function symbols include the unary k_{div} and the binary $+$, $\dot{-}$; and relation symbols include the binary $>, =$. They all have evident interpretations.

The terms of our calculus need a further *multiset signature* for the species and agents. Given a set Lab of *labels* l, a signature consists of sets Spec of species S, and Agent of agents A, each with an arity (L, λ) where $L \subseteq_{\mathrm{fin}} \mathrm{Lab}$ and $\lambda : L \to \mathrm{Sort}$ (we just write $S : \lambda$ and $A : \lambda$). In the example, the species are $\mathrm{Glc}, \mathrm{GlcP} : [\,]$, and the agents are $\mathrm{Cell} : [\mathrm{bool/type}, \mathrm{bool/state}, \mathrm{nat/ATP}]$ and $\mathrm{Env} : [\mathrm{bool/state}]$.

We can now define the terms of our calculus. We inductively define *multilevel multiset terms* and *atomic multilevel multiset terms* as follows, where we further assume available an infinite set of *multiset variables* x:

- every finite multiset $at_0, \ldots, at_{n-1}$ $(n \geq 0)$ of atomic multilevel multiset terms is a multilevel multiset term;
- for every species $S : [s_1/l_1, \ldots, s_n/l_n]$ in Spec and parameter terms $t_i : s_i$ $(i = 1, \ldots, n)$, S_α is an atomic multilevel multiset term, where $\alpha = [t_1/l_1, \ldots, t_n/l_n]$;
- for every agent $A : [s_1/l_1, \ldots, s_n/l_n]$ in Agent, parameter terms $t_i : s_i$ $(i = 1, \ldots, n)$, and multilevel multiset term t, $A_\alpha(t)$ is an atomic multilevel multiset term, where $\alpha = [t_1/l_1, \ldots, t_n/l_n]$; and
- every multiset variable x is an atomic multilevel multiset term.

Instead of (atomic) multiset multilevel term, we may say (atomic) multiset term, (or even, when there is no confusion, just (atomic) term). We write $S_{l_1:t_1,\ldots,l_n:t_n}$ instead of $S_{[t_1/l_1,\ldots,t_n/l_n]}$, and similarly for agents.

As with [OP11], the terms are normal forms for the theory of a commutative monoid with sets of constants and unary operators. There is a constant $S_{[a_1/l_1,\ldots,a_n/l_n]}$ for every species $S : [s_1/l_1, \ldots, s_n/l_n]$ in Spec, and $a_i \in (s_i)_{\mathcal{A}}$ $(i = 1, n)$, and there is a unary operator $A_{[a_1/l_1,\ldots,a_n/l_n]}$, for every agent $A : [s_1/l_1, \ldots, s_n/l_n]$ in Agent and $a_i \in (s_i)_{\mathcal{A}}$ $(i = 1, n)$. The correspondence with universal algebra can be made even closer by considering parametric equational logic, as sketched in [Plo06]; one would then have a 1-1 correspondence between species and agents, and parameterised constants and unary function symbols.

A multiset term is *ground* if it contains no parameter or multiset variables; it is *simple* if every term it contains is an element of some $s_{\mathcal{A}}$. Simple ground terms are used to model populations. In the example such a term could be $t_{\mathrm{pop}} =_{\mathrm{def}} \mathrm{Env}_{\mathrm{state:0}}(t_{\mathrm{cont}})$ where t_{cont} is

$$2\mathrm{Glc},\ \mathrm{Cell}_{\mathrm{type:fast,state:0,ATP:60}}(), \mathrm{Cell}_{\mathrm{type:fast,state:0,ATP:40}}(\mathrm{GlcP}),\ \mathrm{Cell}_{\mathrm{type:slow,state:0,ATP:15}}()$$

SMMR terms are the subcase where the signature has no sorts. Another choice for parameter arguments than by label is by position: one assigns arities $S : u, A : u$ and writes $S_{t_1,\ldots,t_n}$ and $A_{t_1,\ldots,t_n}(t)$. The choice has no theoretical importance but labels seems more appealing in practice, see, for example [DL03, CFS06].

Rules, ranged over by R, are pairs l, r of multiset terms, the *left-* and *right-hand sides*, together with a *stochastic rate term*, i.e., a parameter term k : real, and a *condition*, a quantifier-free formula φ. Rules are written:

$$l \xrightarrow{k} r \ (\varphi)$$

If omitted, the condition is taken to be $\top$.

We impose several constraints on allowable rules. First, every parameter or multiset variable occurring in r, k or φ is required also to occur in l; this is standard in term rewriting, where one matches the left-hand side against the current state, and uses the resulting match to bind the variables in the rest of the rule.

Next come four constraints, discussed further in [OP11]. The *wide subterm* relation is defined to be the least reflexive relation between multiset terms such that if t is a wide multiset subterm of t' then it is a wide subterm of $A_{t_1,\ldots,t_n}(t')$.

The first three constraints are that any multiset variable occurs at most once in l, that every wide subterm of l contains a multiset variable, and that any multiset variable occurs at most once in r: these avoid biologically unfeasible abilities to test for equality or emptiness, or to duplicate populations exactly. (When writing rules, as in all our examples, we conventionally omit "top-level" multiset variables writing $l \xrightarrow{k} r\ (\varphi)$ rather than $l, x \xrightarrow{k} r, x\ (\varphi)$.) The next is that l has no wide subterms with occurrences of distinct multiset variables. This avoids a matching ambiguity and a consequent uncertainty in how to assign rates; the complex right-hand sides introduced below provide an alternative way to split a match without that disadvantage.

The last constraints concern the parameters. We wish the left-hand side parameter terms to be as general as possible, with the idea that any constraints will be picked up by the condition. We therefore impose the constraint that all parameter terms in l are variables, and that no parameter variable occurs twice in l. This constraint makes the exposition simpler, and loses no generality.

In practice, reaction rates would be found as usual, from a combination of biological knowledge and estimation. In some cases, e.g., dependence on volume, the general form of rate functions will be known. The difficulties will surely vary greatly according to the application; as an example, [Mur02, Chapter 11] discusses reaction-diffusion systems, with many references to models, and [DK10] discusses parameter estimation for stochastic such models.

We regard coloured rules as abbreviations for their simple *instances*, which we now define. To each parameter signature, interpretation $\mathcal{A}$, and multiset signature, as above, we associate its *instance* parameter signature, interpretation and multiset signature. This consists of the empty parameter signature, the empty interpretation, and the multiset signature with, for every species $S : [s_1/l_1, \ldots, s_n/l_n]$ and $a_i \in (s_i)_{\mathcal{A}}$ $(i = 1, n)$, a species $S_{[a_1/l_1,\ldots,a_n/l_n]}$, and with, for every agent $A : [s_1/l_1, \ldots, s_n/l_n]$ and $a_1, \ldots, a_n$ in $(s_1)_{\mathcal{A}}, \ldots, (s_n)_{\mathcal{A}}$, an agent $A_{[a_1/l_1,\ldots,a_n/l_n]}$. Multiset terms in the instance signature are in an evident bijective correspondence with simple ground terms; it is convenient to identify them.

With that, given an assignment ρ whose domain includes the parameter variables of a multiset term t (atomic multiset term at) we define a corresponding multiset term t^ρ (respectively, atomic multiset term at^ρ) of the instance signature as follows:

$$\begin{array}{lcl}
(at_0, \ldots, at_{n-1})^\rho & = & at_0^\rho, \ldots, at_{n-1}^\rho \\
(S_{l_1:t_1,\ldots,l_n:t_n})^\rho & = & S_{[(t_1)^\rho_{\mathcal{A}}/l_1,\ldots,(t_n)^\rho_{\mathcal{A}}/l_n]} \\
(A_{l_1:t_1,\ldots,l_n:t_n}(t))^\rho & = & A_{[(t_1)^\rho_{\mathcal{A}}/l_1,\ldots,(t_n)^\rho_{\mathcal{A}}/l_n]}(t^\rho) \\
x^\rho & = & x
\end{array}$$

We can then associate to every rule $R = l \xrightarrow{k} r\ (\varphi)$ a set of rules in the instance signature by:

$$\mathcal{I}(R) = \{l^\rho \xrightarrow{k^\rho} r^\rho \mid \mathrm{dom}(\rho) = \mathrm{PVar}(l), \mathcal{A} \models^\rho \varphi\}$$

where $\mathrm{PVar}(t)$ is the set of parameter variables of a multiset term t.

We next show how to ascribe a quantitative semantics—a stochastic matrix—to a rule. The correctness of the definition is then shown by relating this semantics to that of all the instances of the rule. First we need to calculate the multiset of matches of the left-hand side l against a simple ground term t. A *substitution* is a finite function σ from multiset variables to multiset terms. There are evident actions $t\sigma$, $(at)\sigma$ of such a substitution on a term t and an atomic term at, assuming its domain includes the multiset variables of t, or at, as needed.

A *match* of a multiset term u against a simple ground term t is a pair (ρ, σ) such that $(u^\rho)\sigma = t$, where the domains of ρ and σ are, respectively the parameter and multiset variables of u. Matches of atomic multiset terms against atomic simple ground multiset terms are defined similarly, and disjoint union is extended pairwise to matches. We now define a finite multiset $m(u; t)$ of such matches for multiset terms u (and $m(au; at)$ for atomic multiset terms au) obeying the constraints imposed above on the left-hand side of rules:

$$m(at_0, \ldots, at_{m-1}, x; at'_0, \ldots, at'_{n-1}) =$$

$$\textstyle\sum_{f:[m]\hookrightarrow[n]} \bigcup_{i=0}^{m-1}\ m(at_i; at'_{f(i)}) \uplus ([\,], [\sum_{j\notin f([m])} at_j/x])$$

and:

$$m(S_{l_1:x_1,\ldots,l_n:x_n}; at) =$$

$$\begin{cases} ([\alpha^{-1}(l_1)/x_1, \ldots, \alpha^{-1}(l_n)/x_n], [\,]) & (at = S_\alpha) \\ 0 & (\text{otherwise}) \end{cases}$$

$$m(A_{l_1:x_1,\ldots,l_n:x_n}(u); at) =$$

$$\begin{cases} ([\alpha^{-1}(l_1)/x_1, \ldots, \alpha^{-1}(l_n)/x_n], [\,]) \uplus m(u; t) & (at = A_\alpha(t)) \\ 0 & (\text{otherwise}) \end{cases}$$

using the natural extension of disjoint union of matches to multisets of matches defined by:

$$m \uplus m' = \sum_{(\alpha,\sigma)\in m}\ \sum_{(\alpha',\sigma')\in m'} (\alpha, \sigma) \uplus (\alpha', \sigma')$$

Continuing the example,

$$m(l_{\mathrm{fast}}, y\,; t_{\mathrm{cont}}) = 2([60/a], \sigma_1) + 2([40/a], \sigma_2)$$

where l_{fast} is

$$\mathrm{Glc}, \mathrm{Cell}_{\mathrm{type:fast,state:0,ATP:}a}(x)$$

the left-hand side of R_{fast}, the first GlcP rule (and y is taken to be the variable conventionally omitted), and where

$$\sigma_1 =_{\mathrm{def}} [0/x, \mathrm{Glc}, \mathrm{Cell}_{\mathrm{type:fast,state:0,ATP:40}}(\mathrm{GlcP}), \mathrm{Cell}_{\mathrm{type:slow,state:0,ATP:15}}()/y]$$

$$\sigma_2 =_{\mathrm{def}} [\mathrm{GlcP}/x, \mathrm{Glc}, \mathrm{Cell}_{\mathrm{type:fast,state:0,ATP:60}}(), \mathrm{Cell}_{\mathrm{type:slow,state:0,ATP:15}}()/y]$$

Contexts $C[\,]$ are multiset terms with a hole in them. In order to avoid double counting, we need a narrower class, the *wide contexts* $\mathrm{W}[\,]$. They are defined inductively, taking $[\,]$ to be a wide context, and the multiset $A_{l_1:t_1,\ldots,l_n:t_n}(\mathrm{W}[\,]), t$ to be a wide context if $\mathrm{W}[\,]$ is. Every context can be written in the form $\mathrm{W}[[\,], t]$, and a context $\mathrm{C}[\,]$ is wide if, and only if, every multiset term t is a wide subterm of $\mathrm{C}[t]$.

We next need a count $\mathrm{occ}_t(\mathrm{W}[\,], u)$ of the number of ways in which a simple ground multiset term t can have the form

$\mathrm{W}[u]$, for a given wide context $W[\,]$ and term u:

$$\mathrm{occ}_t([\,], u) \;=\; \begin{cases} 1 & (t = u) \\ 0 & (t \neq u) \end{cases}$$

$$\mathrm{occ}_t((A_\alpha(\mathrm{W}[\,]), v), u) =$$

$$\begin{cases} t(A_\alpha(t'))\,\mathrm{occ}_{t'}(W[\,], u) & (t = A_\alpha(t') + v) \\ 0 & (\text{otherwise}) \end{cases}$$

Note: there is at most one t' such that $t = A_\alpha(t') + v$, and $t(A_\alpha(t'))$ is the number of occurrences of $A_\alpha(t')$ in t. In the example, we have $\mathrm{occ}_{t_{\mathrm{pop}}}(\mathrm{Env}_{\mathrm{state:0}}([\,]), t_{\mathrm{cont}}) = 1$.

We can now define the stochastic matrix Q_R associated to a rule $R = l \xrightarrow{k} r\ (\varphi)$. This is a function from pairs of simple ground terms to non-negative reals, where, for two distinct such ground terms t, t':

$$Q_R(t, t') = \sum_{t = \mathrm{W}[u]} \mathrm{occ}_t(\mathrm{W}[\,], u) \sum_{\substack{(\rho,\sigma) \in m(l;u) \\ \mathcal{A} \models^\rho \varphi,\ t' = \mathrm{W}[(r^\rho)\sigma]}} k^\rho$$

with the diagonal entries being, as usual, one minus the sum of the off-diagonal entries (one easily sees that almost all the off-diagonal entries are 0).

The idea behind this formula is to find all the ways, with their associated multiplicities, that a rule can send a given ground term t to another t', and use this information to determine the total rate of the transition. Such a way is given by a wide context $W[\,]$, a ground term u, and a substitution pair (ρ, σ) such that $t = W[u]$, $t' = \mathrm{W}[(r^\rho)\sigma]$, and $\mathcal{A} \models^\rho \varphi$; and it occurs with multiplicity $\mathrm{occ}_t(\mathrm{W}[\,], u)(m(l, u)(\rho, \sigma))$ at rate k^ρ. Note that the formula makes use of the convention introduced above for summing over elements of a multiset.

Wide contexts are needed to avoid a possibility of double-counting when defining stochastic rates. For example, a rule with left-hand side S, x applies to the term S, S'. However, absent wide contexts, that can be shown in two ways, using either $[\,]$ or $[\,], S'$, but only the first of these is wide.

The formula is the natural extension of that for SMMR, extends the usual stochastic interpretation of chemical reactions, and is similar to that used in Danos and Laneve's κ [DL03], another stochastic rule-based system.

Continuing the example, we find that $Q_{R_{\mathrm{fast}}}(t_{\mathrm{pop}}, t'_{\mathrm{pop}})$ is $2k_{start}$, where $t'_{\mathrm{pop}} = \mathrm{Env}_{\mathrm{state:0}}(t'_{\mathrm{cont}})$, and t'_{cont} is

$$2\mathrm{Glc}, \mathrm{Cell}_{\mathrm{type:fast,state:0,ATP:15}}(6\mathrm{GlcP}),$$
$$\mathrm{Cell}_{\mathrm{type:fast,state:0,ATP:40}}(\mathrm{GlcP}), \mathrm{Cell}_{\mathrm{type:slow,state:0,ATP:15}}()$$

This is because there is just one way to get from t_{pop} to t'_{pop}, viz using $([60/a], \sigma_1)$ and it occurs with multiplicity 2 (note that the required condition is fulfilled in this way, as $60 > 45$).

We obtain a stochastic matrix for a finite set of rules $\mathcal{R}$:

$$Q_{\mathcal{R}}(t, t') =_{\mathrm{def}} \sum_{R \in \mathcal{R}} Q_R(t, t')$$

As detailed above every parametric rule is really an abbreviation for a set of non-parametric rules. So the stochastic matrix for it, as just defined, should be the same as that given by its instances:

Proposition 4.1. *For any rule R and simple ground terms t,t' we have*

$$Q_R(t, t') = \sum_{S \in \mathcal{I}(R)} Q_S(t, t')$$

Proof. One shows by induction on u that

$$m(u; t)(\rho, \sigma) = m(u^\rho; t)([], \sigma)$$

if $\mathrm{PVar}(u) \subseteq \mathrm{dom}(\rho)$, and the rest is a calculation. □

We next describe how to simulate the CTMC given by this stochastic rate matrix in terms of choosing and applying rules from a finite set of rules $\mathcal{R}$. The *activity* of a rule $R = l \xrightarrow{k} r\ (\varphi)$ on a simple ground term t is the total transition rate from t, i.e.,

$$\mathrm{Act}(R, t) = \sum_{t' \neq t} Q_R(t, t')$$

Explicitly we have:

$$\mathrm{Act}(R, t) = \sum_{t = \mathrm{W}[u]} \mathrm{occ}_t(\mathrm{W}[\,], u) \sum_{\substack{(\rho,\sigma) \in m(l;u) \\ \mathcal{A} \models^\rho \varphi}} k^\rho \qquad (**)$$

Note that $\mathrm{Act}(R_{\mathrm{fast}}, t_{\mathrm{pop}})$ is $2k_{start}$ and not $4k_{start}$, as there is no way to get from t_{pop} using $([40/a], \sigma_2)$, since the rule condition is not then fulfilled.

The simulation has a current time, initialised to 0, and a current state, a simple ground term t. It proceeds by cycling through the following sequence, for as long as required:

- If $\lambda =_{\mathrm{def}} \sum_{R \in \mathcal{R}} \mathrm{Act}(R, t)$ is zero, stop the simulation.
- Choose τ from the exponential distribution with rate parameter λ, and add it to the current time.
- Choose a rule $R = l \xrightarrow{k} r\ (\varphi)$ from $\mathcal{R}$ with probability $\mathrm{Act}(R, t)\lambda^{-1}$.
- Choose a wide context $\mathrm{W}[\,]$, a u, and a (ρ, σ) such that $t = \mathrm{W}[u]$, $(l^\rho)\sigma = u$, and $\mathcal{A} \models^\rho \varphi$ with probability
$$\frac{\mathrm{occ}_t(\mathrm{W}[\,], u)(m(l; u)(\rho, \sigma))k^\rho}{\mathrm{Act}(R, t)}$$
- Update t to $\mathrm{W}[(r^\rho)\sigma]$.

Note that in the fourth step, a way of applying a rule is chosen with probability its fraction of the total activity of the rule.

Normally in a simulation one graphs species populations against time. We generalise this to *observation patterns* which we take to be pairs (l, φ) of multiset terms and quantifier-free formulas. The activity of such a pattern (l, φ) in a simple ground term t is:

$$\mathrm{Act}((l, \varphi), t) =_{\mathrm{def}} \sum_{\substack{(\rho,\sigma) \in m(l;t) \\ \mathcal{A} \models^\rho \varphi}} 1$$

So far, the right-hand sides of rules are simple in the sense that they only involve multiset terms. We next introduce some complexity on the right-hand side in order to allow for random choice of parameters and for splitting populations.

First we allow an additional signature of *probabilistic function symbols* p with arity (u, s'), with the restriction that $s'_{\mathcal{A}}$ be countable, to ensure a countable distribution (for sampling reasons). Parameter terms are now formed using both kinds of function symbols and the *probabilistic parameter terms* are those containing a probabilistic function symbol. We then assume that the interpretation $\mathcal{A}$ additionally

assigns to each probabilistic function symbol p with arity (u, s') a function $p_{\mathcal{A}} : u_{\mathcal{A}} \to \mathcal{P}(s_{\mathcal{A}})$ where, for any countable set X, $\mathcal{P}(X)$ is the set of probability distributions on X. We now have an interpretation $t^{\rho}_{\mathcal{A}} \in \mathcal{P}(s_{\mathcal{A}})$ of terms $t : s$.

Complex right-hand sides, r_c, have the following form:

$$\begin{array}{l} let\ x_1 = t_1, \ldots, x_m = t_m\ in \\ let\ (y_1, z_1) = w_1, \ldots, (y_n, z_n) = w_n\ in\ r \end{array}$$

where: the parameter variables x_i are all different; each x_i has the same sort as t_i; the multiset variables w_j, y_j, z_j are all different; and r is a multiset term not containing any probabilistic parameter terms. (The restriction on where probabilistic terms can appear on the right-hand side loses no generality and simplifies the exposition.) The *free variables* of r_c are the variables of the t_i, the w_j, and any variable of r which is not one of the x_i, y_j or z_j. When m or n is zero, one writes the evident simpler forms.

Rules with complex right-hand sides have the form:

$$R = l \xrightarrow{k} r_c\ (\varphi)$$

where r_c has the above form, neither l nor k are probabilistic, and φ contains no probabilistic terms. The constraints on rules are the same for l, k and φ as before; we also require that all the free variables of r_c occur in l and that no multiset variable has two occurrences in r.

The idea of such rules is that each x_i is assigned a random element of the distribution given by t_i, and that the population assigned to w_j is divided in two, with each element being assigned to y_j or z_j with equal probability. (The construct could be generalised to, e.g., unequal probabilities, but we do not know any use for the greater generality.)

Rather than repeat the above theory, but for complex right-hand sides, we simply explain how to modify the simulation algorithm, where now $\mathcal{R}$ is a finite set of rules of the new form. The simulation cycle proceeds as in the first three steps. The fifth step is replaced by this one:

- Choose a_i from the probability distribution $(t_i)^{\rho}$, for $i = 1, m$, and choose $t'_j \leq w_j\sigma$ $(j = 1, n)$ with probability

$$\prod_{j=1}^{n} \prod_{at' \in |w_j\sigma|} b(t'_j(at'); w_j\sigma(at'), 1/2)$$

and update t to $W[(r^{\rho'})\sigma']$ where ρ' is

$$\rho \frown [a_1/x_1, \ldots, a_m/x_m]$$

and σ' is

$$\sigma \frown \bigcup_{j=1}^{n} [t'_j/y_j, (w_j\sigma \dot{-} t'_j)/z_j]$$

The use of the binary distribution ensures that the elements of each $w_j\sigma$ are assigned with equal probability to y_j or z_j.

5. IMPLEMENTATION

We have implemented a prototype stochastic simulator for CSMMR with certain restrictions, explained below, and with some inessential syntactic differences. We explain the main implementation ideas.

Multisets of species or agents are represented by *hash tries* [Bag01], as are vectors of colours.

We implement the simulation algorithm of Section 4. There are three important steps in this algorithm (with or without complex right-hand sides):

- the computation of the total activity of the system;
- the random drawing of a rule according to its activity; and
- the random drawing of an instance of the rule according to its activity.

These steps are implemented by computing the distribution of rule instances. To implement finite distributions we use a data structure of *mass-extended hash tries*; it allows:

- the efficient insertion and deletion of elements;
- the efficient computation of the total mass of the distribution; and
- the efficient drawing of an element according to its mass.

Mass-extended hash tries extend hash tries with a partial mass at every node of the hash trie, which represents the mass of the elements in the sub-trie rooted at that node. While having the same efficiency as hash tries for insertion and deletion, this also allows us to compute the total mass of the trie in constant time, and to draw an element in logarithmic time. This extension of hash tries seems novel.

It is used to implement sums of distributions, following formula (**) of Section 4 giving the activity of a rule. However, one then needs to sum k^{ρ} over all matches (ρ, σ). This may be too expensive when applied to a non-unary left-hand side. For example, consider a rule of the form

$$A_{l:a}(x), B_{m:b}(y) \xrightarrow{k(a,b)} \ldots$$

where A and B are different. To sum k^{ρ} over all (ρ, σ), we need to find all instances of $A_{l:a}(x), B_{m:b}(y)$ and if there are n_A instances of $A_{l:a}(x)$ and n_B instances of $B_{m:b}(y)$, there are $n_A.n_B$ instances of the pair $A_{l:a}(x), B_{m:b}(y)$. Worse, if, during a step, we change one of the $A_{l:a}(x)$ instances, we have to recompute k^{ρ} for the n_B pairs in which it occurred. This becomes prohibitively slow for large systems.

There are, however, cases where we can use a far better algorithm. Suppose k can be written as $k_A \cdot k_B$, where b does not occur in k_A and a does not occur in k_B. We can write each ρ as $\rho_A \cup \rho_B$, where the domains of ρ_A and ρ_B are, respectively, $\{a\}$ and $\{b\}$, and we can write each σ as $\sigma_A \cup \sigma_B$ analogously: this is because, by the linearity restriction, no variable can occur in both $A_{l:a}(x)$ and $B_{m:b}(y)$. Then:

$$\sum_{(\rho,\sigma)} k^{\rho} = \sum_{\rho_A,\rho_B,\sigma_A,\sigma_B} k_A^{\rho_A}.k_B^{\rho_B} = \left(\sum_{(\rho_A,\sigma_A)} k_A^{\rho_A}\right) . \left(\sum_{(\rho_B,\sigma_B)} k_B^{\rho_B}\right)$$

So we need only compute only a distribution of the instances of $A_{l:a}(x)$ and a distribution of the instances of $B_{m:b}(y)$, which can be done in time $O(n_A + n_B)$. Then a distribution of the pair instances can be obtained in constant time, as

- the activity of the pair is simply the product of the activities; and
- to draw a pair instance randomly according to its mass, one can draw independently from the distribution of instances of $A_{l:a}(x)$ and the distribution of instances of $B_{m:b}(y)$.

Further, when an instance of $A_{l:a}(x)$ is modified, only one k_A and the product has to be recomputed, and this can be done in constant time.

There is a complication if we change the form of the rule to

$$A_{l:a}(x), A_{l:a'}(y) \xrightarrow{k} \ldots$$

The distributions of $A_{l:a}(x)$ and $A_{l:a'}(y)$ will have a non-empty intersection, if they are non-empty. So when computing activities we would over-approximate and when drawing independently from these distributions, it is possible to draw the same element. We handle this in step 4 of the stochastic simulation, by testing whether the instance drawn is conflicting. If there is a conflict, we *reject* (i.e., we do not apply the rule) and continue with the next simulation step.

The decomposition of rates into an independent product given above can be generalised to *hierarchically independent (h.i.)* rate functions. These are defined as follows:

- k is h.i. for $at_1, \ldots, at_n$ if it can be written as $\prod_{i=1}^{n} k_i$ where k_i is h.i. for at_i;
- k is h.i. for $S_{[t_1/l_1,\ldots,t_n/l_n]}$ if its variables are included in those of $t_1, \ldots, t_n$;
- k is h.i. for $A_{[t_1/l_1,\ldots,t_n/l_n]}(t)$ if it can be written as $k_1 k_2$ where the variables of k_1 are included in those of $t_1, \ldots, t_n$, and k_2 is h.i. for t; and
- k is h.i. for any multiset variable.

We can define *hierarchically independent (h.i.)* rule conditions similarly, using conjunctions in place of products. Our implementation only allows independent rates and conditions which are conjunctions of a h.i. condition and another. We use the above ideas to handle the h.i. parts; the other part of the condition is not considered when computing activities, but only in the fourth simulation step where instances not satisfying it are rejected, and the simulation continues with the next step.

As explained in [DFFK07], similar ideas are used in the κ language simulator. Indeed, again for efficiency purposes, that simulator also decomposes instances of pattern pairs into pairs of instances, over-approximates acitvities, and rejects conflicts. Our adaption to multilevel terms, coloured systems, and independent rates and conditions is novel.

6. FURTHER EXAMPLES

6.1 Volume

We start with a very simple system illustrating the use of colours to represent the volume of a growing cell. The central feature of such a system is that when the volume of the cell changes, the rates of the reactions inside it change too. This is because the probability of two or more molecules meeting decreases as the volume increases.

First we represent the volume of a cell as a positive real colour attached to the cell agent: Cell : [real/vol]. We then have to model the growth of the cell. This could be related to various other colours, representing the state of the cell, or to its contents. Here, for the sake of simplicity, we choose a constant rate of growth with factor $\alpha =_{\text{def}} 0.01$:

$$\text{Cell}_{\text{vol}:v}(x) \xrightarrow{k_{growth}} \text{Cell}_{\text{vol}:\alpha \times v}(x)$$

We have assumed a function symbol $\times$: real, real $\to$ real, interpreted by multiplication. It is easy to modify this rule to stop the growth at a maximum size, to slow down the growth as the cell grows, or to model linear growth.

We introduce two cellular species A : [ ] and B : [ ] and three rules:

- the constant creation of A's within a cell:

$$\text{Cell}_{\text{vol}:v}(x) \xrightarrow{k_A} \text{Cell}_{\text{vol}:v}(A, x)$$

- the interaction of two A's to create a B, with the rate of interaction decreasing with volume:

$$\text{Cell}_{\text{vol}:v}(2A, x) \xrightarrow{k_B/v} \text{Cell}_{\text{vol}:v}(B, x) \qquad (v > 0)$$

  where / is interpreted by real division (except at 0).

- the degradation of B's:

$$B \xrightarrow{k_{B^-}}$$

Figures 2 and 3 give a simulation of this system, with the following rates:

k_{growth}	7
k_A	1
k_B	1
k_{B^-}	0.0001

and the following initial population:

$$\text{Cell}_{\text{vol}:1.0}()$$

As expected, the equilibrium between A's and B's shifts from a high concentration of B's to a higher concentration of A's as the volume increases. Indeed, the bigger the cell, the less likely are two A's to meet and form a B.

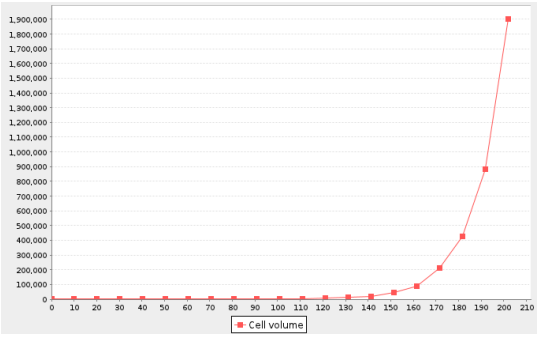

Figure 2: Cell volume

6.2 Diffusion

In this section, we illustrate the use of colours to model simple geometry and diffusion. In the example, molecules of a species diffuse, and are activated when they reach a position-dependent heat source.

The species is A : [pos/pos, real/mass, col/col] where pos and col are sorts of positions and colours; we take the positions to be the first 10 natural numbers, and the colours to be {blue, red}, corresponding to two states of activation: blue for inactive and red for active.

We also assume that:

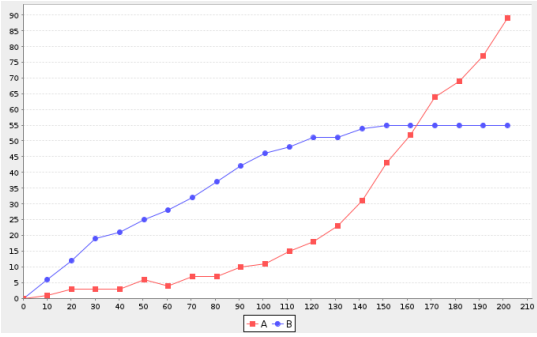

Figure 3: The population of B's initially grows faster, but as the cell grows, the population of A's increases.

- rN : pos → pos, a "randomNeighbour" probabilistic function symbol used to choose the direction of diffusion (we make a simple choice, taking $\mathrm{rN}(p)$ to be $l(p)$ or $r(p)$ with equal probability, where $l(p) =_{\mathrm{def}} p \dot{-} 1$ and r is defined symmetrically); and
- δ : real → real, a mass-dependent rate function computing the species diffusion rate (we again make a simple choice, taking it to be constantly 1).

The diffusion of A's is modelled by:

$$A_{\mathrm{pos}:p,\mathrm{mass}:m,\mathrm{col}:k} \xrightarrow{\delta(m)} \mathit{let}\ p' = \mathrm{rN}(p)\ \mathit{in}\ A_{\mathrm{pos}:p',\mathrm{mass}:m,\mathrm{col}:k}$$

To model the heat-induced activation of A's we add:

$$A_{\mathrm{pos}:p,\mathrm{mass}:m,\mathrm{col}:\mathrm{blue}} \xrightarrow{\mathrm{heat}(p)} A_{\mathrm{pos}:p,\mathrm{mass}:m,\mathrm{col}:\mathrm{red}}$$

where we further assume:

- heat : pos → real, a position-dependent activation rate function, measuring the heat at a given position. (We take it to be 0 except at a given position p_0 where it is 100; and we take $p_0 = 0$.)

A's can cool down slowly:

$$A_{\mathrm{pos}:p,\mathrm{mass}:m,\mathrm{col}:\mathrm{red}} \xrightarrow{k_{cool}} A_{\mathrm{pos}:p,\mathrm{mass}:m,\mathrm{col}:\mathrm{blue}}$$

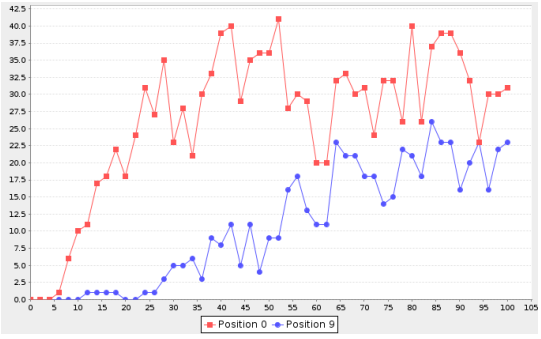

Figure 4: Activated A's at positions 0 and 9.

Figure 4 shows a run of a simulation of this system, taking $k_{cool} = 0.005$, The figure shows the activated A's at positions 0 and 9. Initially 300 inactive A's are at 9, and there are none elsewhere. We see a first wave of activation of A's at 0 as inactive molecules diffuse from 9 to 0, the only heat source, where they are activated. We then see a second wave of activated A's at 9, as the activated A's diffuse from 0 to 9.

This simple model has a limitation. If we were to introduce a rule (*) that can be applied when two A's meet:

$$A_{\mathrm{pos}:p_1}, A_{\mathrm{pos}:p_2} \xrightarrow{k} \ldots \quad (p_1 = p_2)$$

we would have to use the non-hierarchical condition $p_1 = p_2$. As explained in Section 5, this would be quadratic in the number of molecules. That could be improved by representing each position p by an agent Pos : [pos/pos] containing all the species in that position. However one would then need a non-hierarchical condition for the diffusion rule:

$$\begin{aligned}&\mathrm{Pos}_{\mathrm{pos}:p_1}(A,x), \mathrm{Pos}_{\mathrm{pos}:p_2}(y) \xrightarrow{k'}\\ &\quad \mathrm{Pos}_{\mathrm{pos}:p_1}(x), \mathrm{Pos}_{\mathrm{pos}:p_2}(A,y) \quad (|p_1 - p_2| = 1)\end{aligned}$$

and this rule is quadratic in the number of positions. This is an improvement as the number of positions is typically lower than the number of molecules. In Section 7, we sketch a possible more efficient solution to this problem.

6.3 A toy chemotaxis model

Our final example is more substantial; it is a simple version of bacterial *chemotaxis*, inspired by [KLH09]. Chemotaxis allows bacteria to direct their movement in response to certain chemicals in the environment. For example, while it is beneficial for a bacterium to move towards glucose, it may be too small to sense the environmental glucose gradient. To solve this problem, some bacteria use a system of biased random movement, alternating between two phases:

- straight motion; and
- random change of direction

with the first phase lasting longer if glucose concentration increases. See [WA04] for a more complete description.

Our model is simplified in multiple ways:

- there are just two chemotactic proteins;
- there are two flagella (E. Coli has five to eight); and
- bacteria only move along a line, either *left* or *right*.

We use an agent Bact : [real/pos, sign/dir] to model bacteria positioned along the real line with direction +1 (facing right) or −1 (facing left).

Our model bacteria have two flagellal *motors*, each either turning clockwise (CW) or counterclockwise(CCW). We model motors by a species Motor : [rot/dir], where each motor can vary between CW and CCW:

$$\mathrm{Motor}_{\mathrm{dir}:d} \xrightarrow{k_{dir}} \mathrm{Motor}_{\mathrm{dir}:\mathrm{op}(d)}$$

with the evident interpretation of op : dir → dir.

The motors determine the direction of the motion of the bacterium. First, if both are turning counterclockwise, then the bacterium moves forwards:

$$\begin{aligned}&\mathrm{Bact}_{\mathrm{pos}:p,\mathrm{dir}:d}(2\mathrm{Motor}_{\mathrm{dir}:\mathrm{CCW}}, x)\\ &\quad \xrightarrow{k_{step}} \mathrm{Bact}_{\mathrm{pos}:p+\delta d,\mathrm{dir}:d}(2\mathrm{Motor}_{\mathrm{dir}:\mathrm{CCW}}, x)\end{aligned}$$

This rule uses a step size $\delta =_{\mathrm{def}} 0.02$. It also uses the fact that the direction of the bacterium is represented by either

$+1$ or -1. Next, if either motor is turning clockwise, the bacterium changes direction:

$$\mathrm{Bact}_{\mathrm{pos}:p,\mathrm{dir}:d}(\mathrm{Motor}_{\mathrm{dir}:\mathrm{CW}}, x) \xrightarrow{k_{turn}} \mathrm{Bact}_{\mathrm{pos}:p,\mathrm{dir}:\mathrm{op}(d)}(\mathrm{Motor}_{\mathrm{dir}:\mathrm{CW}}, x)$$

The bacterium can sense food in the environment, using a probe, the Tar complex. This is modelled by a species Tar : [bool/active]. A Tar complex can either be active (1) or inactive (0). The level of active Tar complexes is used as a proxy for the quality of the environment: the better the environment, the smaller the number of active Tar's. This allows the bacterium to capture the gradient of the sugar along the direction of its movement. In more detail, sugar absorption is modelled by the following rule:

$$\mathrm{Bact}_{\mathrm{pos}:p,\mathrm{dir}:d}(\mathrm{Tar}_{\mathrm{active}:1}, x) \xrightarrow{\mathrm{gluc}(p)} \mathrm{Bact}_{\mathrm{pos}:p,\mathrm{dir}:d}(\mathrm{Tar}_{\mathrm{active}:1}, \mathrm{Glc}, x)$$

where sugar is modelled by the species Glc : [ ], and we assume a function symbol gluc : pos $\to$ real, associating to each position, the abundance of glucose at this point of the environment. We chose the following interpretation of gluc:

$$\mathrm{gluc}_{\mathcal{A}}(x) = \frac{10}{1 + 2|x - 5|}$$

This function is maximal at $x = 5$.

When there is sugar inside the bacterium, two species, Che^+ : [ ] and Che^- : [ ], are produced. They are abstract representations of, respectively, the chemotactic excitation network and the inhibitory feedback loop:

$$\mathrm{Glc} \xrightarrow{k_{GlcChe^+}} \mathrm{Che}^+, \mathrm{Glc} \qquad \mathrm{Glc} \xrightarrow{k_{GlcChe^-}} \mathrm{Che}^-, \mathrm{Glc}$$

Che^+ makes the motors more likely to turn CCW, which, in turn, induces longer straight motions:

$$\mathrm{Motor}_{\mathrm{dir}:\mathrm{CW}}, \mathrm{Che}^+ \xrightarrow{k^+} \mathrm{Motor}_{\mathrm{dir}:\mathrm{CCW}}, \mathrm{Che}^+$$

Che^- inhibits the sensing activity of Tar:

$$\mathrm{Tar}_{\mathrm{active}:1}, \mathrm{Che}^- \xrightarrow{k^-} \mathrm{Tar}_{\mathrm{active}:0}, \mathrm{Che}^-$$

Che^+, Che^- and Glc degrade, while inactivated Tar complexes get reactivated:

$$\mathrm{Che}^+ \xrightarrow{k_{Che^+}} \qquad \mathrm{Che}^- \xrightarrow{k_{\mathrm{Che}^-}}$$
$$\mathrm{Glc} \xrightarrow{k_{Glc}} \qquad \mathrm{Tar}_{\mathrm{active}:0} \xrightarrow{k_{\mathrm{Tar}}} \mathrm{Tar}_{\mathrm{active}:1}$$

We ran a simulation for a bacterium starting at position 7 and direction $+1$, with the following rates:

k_{dir}	3	k_{Che^+}	7
k_{GlcChe^+}	15	k_{Che^-}	1.5
k_{GlcChe^-}	10	k_{Glc}	3
k^-	0.1	k_{step}	5
k^+	5	k_{turn}	3
k_{Tar}	30		

and with the following initial population:

$$\mathrm{Bact}(16\mathrm{Tar}_{\mathrm{active}:1}, 2\mathrm{Motor}_{\mathrm{dir}:\mathrm{CCW}})$$

Figure 5 shows bacterial position as a function of simulation time: note that the bacterium tends to stay quite close to the position maximising sugar level.

This model is too simple to draw any biological conclusion. However it does illustrate how CSMMR can be used to straightforwardly model complex cellular behaviour. Having a one-dimensional chemotactic model, one would next seek a model that works in two or three dimensions; this may require a more biologically realistic guidance system.

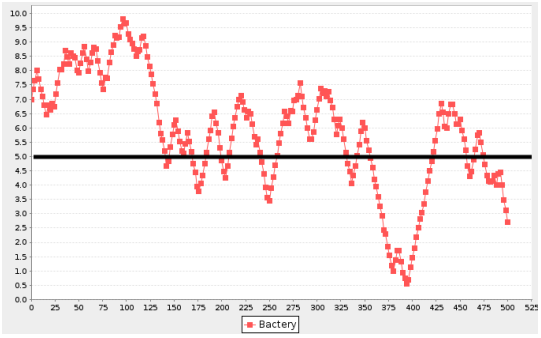

Figure 5: Bacterial position. Food is maximal at 5 (marked by a line)

7. DISCUSSION AND CONCLUSION

The immediate need for the development of coloured stochastic multilevel multiset rewriting is its application to a variety of realistic biological examples. It would also be interesting to compare our system to other general multilevel modelling systems such as Simmune [MXA06]. Such applications would provide a test of the formalism's usefulness and, doubtless, lead to improvements.

Syntactic aids for writing rules, such as abbreviations involving labels, would increase the formalism's usability (cf. Danos and Laneve's κ [DL03]). A graphical presentation of rule systems is also needed; in that connection a suitable notion of multilevel Petri net would be useful, even if only for *two-level systems*, where agents do not occur inside others in populations or rules. Finally, when there is a need for modularisation for large-scale model development, one might wish for a language along the lines of, say, LBS [PP10].

We hope that the orthogonality of the various components of CSMMR will aid its learnability. Starting from experience with the usual stochastic reaction systems, one could independently learn the use of parameters (including rate functions and conditions), and agents (to model compartments); following that, complex right-hand sides should prove natural. Further, there are standard rule patterns for common needs, such as signalling, movement, replication and so on, illustrated in the examples given above.

The formalism could be made more powerful. One could introduce complex left-hand sides to guide matching algorithms. Instead of the rule (*) of Section 6.2 one might write:

$$p : \mathrm{pos}[A_{\mathrm{pos}:p}, A_{\mathrm{pos}:p}] \longrightarrow \ldots$$

which would result in a linear search over all positions, rather than the above quadratic searches.

One could also weaken the hierarchical independence condition on rates and preconditions to one where the rate corresponding to a subterm can depend not only on the subterm's parameters, but also on the parameters of all its ancestors in the term. This would allow one to model, for example, more complex dependencies between the inside of a cell and the state of its environment, while retaining a reasonably

efficient implementation.

Parameterisation increases expressiveness: a single rule can have infinitely many instances. The *finite case* is where all the $s_{\mathcal{A}}$, ($s \neq$ real), are finite and real does not appear in the arity of any species or agent. A finite set of rules can then have at most m^n instances, where m is the maximum size of any $s_{\mathcal{A}}$, ($s \neq$ real) and n is the number of parameter variables on the left-hand side of any rule. A natural succinctness question is whether this bound is necessary, comparing rule systems up to stochastic matrix isomorphism; this is open even in the stochastic coloured Petri net case, by which we mean the one where there are no agents, and the right-hand sides of rules are not complex.

While this paper has only concerned modelling certain CTMC's, their analysis is evidently also of central importance (see [HR10] for background). Even coloured stochastic Petri nets have been little studied, and colour aggregation or symmetry methods suggest themselves [Buc07, GD97]. As regards multilevel aspects, the difficulty is that not only is the state space infinite, but the number of dimensions of the multisets involved may have no upper bound (a similar situation arises with κ); particular cases, such as two-level systems may be more tractable.

Parameterisation is not only applicable to SMMR: one can consider adding it to any rule-based formalism in which one can identify parameterisable operations. In particular, it should be straightforward to give a coloured version of bigraphs. Guided by the equivalence of SMMR with (stochastic) place graphs shown in [OP11], one would add a sort signature assigning arities (L, λ) to controls and decorating nodes with L-labelled tuples of parameter terms whose sorts are given by the sort of the control labelling the node.

It ought also to be straightforward to give a parameterised version of κ. In this regard, a coloured version of a combination of κ and multilevel mulisets could be very attractive.

Acknowledgments

Our work was supported by a joint BBSRC/EPSRC Grant, BB/D019621/1, and by a Royal Society-Wolfson Award.

8. REFERENCES

[Bag01] P. Bagwell, Ideal hash trees, Technical report, School of Computer and Communication Sciences, Swiss Institute of Technology Lausanne, 2001.

[Buc07] P. Buchholz, Iteration at Different Levels: Multi-Level Methods for Structured Markov Chains, *Web Inf. Retrieval and Linear Alg. Algorithms*, Dagstuhl Seminar Proceedings, 7071, IBFI, 2007.

[CFS06] L. Calzone, F. Fages & S. Soliman, BIOCHAM, *Bioinformatics*, 22(14), 1805–1807, 2006.

[GD97] G. Chiola, C. Dutheillet et al, A symbolic reachability graph for coloured Petri nets, *Theor. Comput. Sci.*, 176(1–2), 39–65, 1997.

[CD10a] M. Coppo, F. Damiani, et al, Stochastic calculus of wrapped compartments, *Proc. 8th. QAPL*, EPTCS, 28, 82–98, 2010.

[CD10b] M. Coppo, F. Damiani, et al, Hybrid calculus of wrapped compartments, *Proc. 4th. MeCBIC*, EPTCS, 40, 102–120, 2010.

[DL03] V. Danos & C. Laneve, Core formal molecular biology, *Proc. 12th. ESOP*, LNCS, 2618, 302–318, Springer, 2003.

[DFFK07] V. Danos, J. Feret, et al, Scalable simulation of cellular signaling networks, *Proc. 5th. APLAS*, LNCS, 4807, 139–157, 2007.

[DK10] M. A. Dewar, V. Kadirkamanathan, et al, Parameter estimation and inference for stochastic reaction-diffusion systems: application to morphogenesis in D. melanogaster, *BMC Systems Biology*, 4(2), 2010.

[HR10] M. Heiner, C. Rohr, et al, A comparative study of stochastic analysis techniques, *Proc. 8th. CMSB*, 96-106, ACM, 2010.

[Jen92] K. Jensen, *Coloured Petri Nets: Basic Concepts, Analysis Methods and Practical Use. Vol. 1, Basic Concepts*, Springer Verlag, 1992.

[Jen94] K. Jensen, An introduction to the theoretical aspects of coloured Petri nets, LNCS, 803, 230–272, 1994.

[KLH09] H. Kugler, A. Larjo, & D. Harel, Biocharts: a visual formalism for complex biological systems, *J. R. Soc. Interface*, 2009.

[KMT08] J. Krivine, R. Milner & A. Troina, Stochastic bigraphs, *ENTCS*, 218, 73–96, 2008.

[LH10] F. Liu & M. Heiner, Colored Petri nets to model and simulate biological systems, *Proc. BioPPN*, 70–84, 2010.

[Mil09] R. Milner, *The Space and Motion of Communicating Agents*, CUP, 2009.

[MXA06] M. Meier-Schellersheim, et al, Key role of local regulation in chemosensing revealed by a new molecular interaction-based modeling method, *PLoS Comput Biol.*, 2(7), e82, 2006.

[OP11] N. Oury & G. D. Plotkin, Multi-level modelling via stochastic multi-level multiset rewriting, *MSCS*, Special issue on DCM 2010, to appear.

[Mur02] J. D. Murray, *Mathematical Biology I. An Introduction*, Springer, 2002.

[PP10] M. Pedersen & G. D. Plotkin, A language for biochemical systems: design and formal specification, *T. Comp. Sys. Biology*, 12, 77–145, LNCS, 5945, 2010.

[PSB01] T. Pfeiffer, et al, Cooperation and competition in the evolution of ATP-producing pathways, *Science*, 292(5516), 504–507, 2001.

[Plo06] G. D. Plotkin, Some varieties of equational logic, *Algebra, Meaning, and Computation*, LNCS, 4060, 150–156, 2006.

[Run04] T. Runge, Application of coloured Petri nets in systems biology, *Proc. 5th. CPN*, 77–95, 2004.

[TM06] C. Taubner, et al, Modelling and simulation of the TLR4 pathway with coloured Petri nets, *Proc. 28th. Conf. Eng. in Med. and Bio. Soc.*, 2009–2012, 2006.

[WA04] G. Wadhams & J. Armitage, Making sense of it all: bacterial chemotaxis, *Nat. Rev. Mol. Cell Biol.*, 5(12), 1024–37, 2004.