

THERMODYNAMIC GRAPH-REWRITING

VINCENT DANOS^a, RUSS HARMER^b, AND RICARDO HONORATO-ZIMMER^c

^{a,c} School of Informatics, University of Edinburgh
e-mail address: vdanos@inf.ed.ac.uk, r.honorato@sms.ed.ac.uk

^b LIP, CNRS & ENS Lyon
e-mail address: russell.harmer@ens-lyon.fr

ABSTRACT. We develop a new thermodynamic approach to stochastic graph-rewriting. The ingredients are a finite set of reversible graph-rewriting rules \mathcal{G} (called generating rules), a finite set of connected graphs \mathcal{P} (called energy patterns), and an energy cost function ϵ which associates real values to each of these energy patterns. The idea is that \mathcal{G} defines the qualitative dynamics, by showing which transformations are possible, while \mathcal{P} and ϵ allow one to attach an energy to the reachable graphs and, thereby, describe their long-term probability distribution π . Given \mathcal{G} and \mathcal{P} , we construct a finite set of rules $\mathcal{G}_{\mathcal{P}}$ which (i) has the same qualitative transition system as \mathcal{G} ; and (ii) when equipped with rates according to ϵ , defines a continuous-time Markov chain of which π is the unique fixed point. The construction relies on the use of site graphs and a technique of ‘growth policy’ for quantitative rule refinement which is of independent interest.

This division of labour between the qualitative and long-term quantitative aspects of the dynamics leads to intuitive and concise descriptions for realistic models (see §4). It also guarantees thermodynamical consistency (*aka* detailed balance), otherwise known to be undecidable, which is important for some applications. Finally, it leads to parsimonious parameterizations of models, again an important point in some applications.

1. INTRODUCTION

Along with Petri nets, communicating finite state machines, and process algebras, models of concurrent systems based on graphs and graph transformations (GTS) have long been investigated as means to describe, verify and synthesize distributed systems [14]. Beyond their visual aspect, which is often useful in modelling situations, there is a lot to like about GTSs: there are category-theoretic frameworks to express them and encapsulate their syntax; and the existence of a strong meta-theory [21] is a reassurance that methodologies developed in specific cases can be ‘ported’ to other variants.

2012 ACM CCS: [**Software and its engineering**]: Software notations and tools—Formal language definitions—Syntax / Semantics; [**Theory of Computation**]: Logic; Formal languages and automata theory—Formalisms—Rewrite systems.

Key words and phrases: graph rewriting, thermodynamics, statistical mechanics.

Graph-rewriting rules are convenient for writing compact models and modifying them [8], and lend themselves naturally to probabilistic extensions [18, 20]. However, for all their flexibility, even rules can only do so much. We ask in this paper “what if we did not have to write the rules?”. This is where we take a page from the book of classical statistical mechanics. In such models, which often involve graph-like structures, as in the Ising model, the dynamics is not described upfront. Instead, the system of interest is equipped with an ‘energy landscape’ which specifies its long run behaviour, be it deterministic as in classical mechanics, or probabilistic in statistical physics. The dynamics just follows from the energy data. In the eye of a computer scientist, this use of energy looks like a latent syntax. (This is especially true in the application of these ideas to molecular dynamics.)

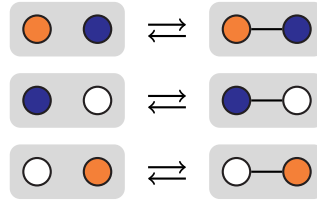
The broad aim of this paper is to make this syntax explicit by introducing energy patterns and costs from which the total energy of a state of the system can be computed; and to define a procedure whereby, indeed, the dynamics described as probabilistic graph-rewriting rules can be derived from these energy data. Descriptively, this takes us to an entirely new level of conciseness (as in the example shown in §4). It also guarantees thermodynamical consistency by construction, a property known otherwise to be undecidable [11]. This property provides a predictable equilibrium state for the system which can be used as a guide when writing models by giving an intuitive understanding of favoured states and the (possibly overlapping) subgraphs within them. But perhaps the nicest byproduct of this approach is the fact that the methodology leads to parsimonious parameterizations. The parameter space which usually scales as the number of rules (which in turn has at best a logarithmic impact on the cost of a simulation event [5]), will now scale as the number of energy patterns provided in the specification.

The particular GTSs we consider form a reversible subset of the Kappa site-graph stochastic rewriting language. Kappa is used for the simulation and analysis of combinatorial dynamical systems as typically found in cellular signalling networks [22, 26] and has been predicted to “become one of the mainstream modelling tools of systems biology within the coming decade” [1]. Similar graph formalisms where nodes have a controlled valence/degree have been considered *e.g.* the BNG language [15, 19], Kissinger and Dixon’s quantum proof language [13], and Kirchner *et al.* chemical calculi [3]. Site-graph rewriting has found recently a ‘home’ both in the single-pushout GTS tradition [9] and the double-pushout one [6, 17]. This makes one hopeful that the thermodynamic methodology we propose can cross over to other fields where quantitative GTSs can be used, *e.g.* in the modelling of adaptive networks [16]. While our scalable energy-based parameterization is particularly important in biological applications where parameters often need to be inferred, one can imagine it to be useful in other modelling situations with uncertainty.

Outline: We start by introducing a running example that will help us illustrate the concepts presented throughout the paper. Then we proceed with the definition and relevant properties of the specific GTS we use, namely a simple reversible fragment of Kappa. Next, we introduce growth policies (adapted from Ref. [25]), a tool which allows one to replace a rule with an orthogonal set of refined rules while preserving the quantitative semantics. We use this tool with a specific policy which refines a rule into finitely many rules, each of which has a definite energy balance with respect to a given set of energy patterns. This leads to our main theorem which guarantees that the stochastic dynamics of the obtained refined rule set converges to an equilibrium distribution parametrized by the cost of each energy pattern.

Throughout, the presentation is set in category-theoretic terms and mostly self-contained. A substantial example concludes the paper.

1.1. Running example: Assembling triangles. The following example will be used to show the intricacies of our rule generation mechanism. We consider graphs with three types of nodes where each can only bind nodes of a different type, and at most one thereof. As generators, we consider simple binding and unbinding rules subject to this constraint (see details in §2.4):



These rules, given an unbounded supply of the three types of nodes, can create chains (of any length) and closed cycles (of length some multiple of three) as shown in Fig. 1. Given the simplicity of the graphs that can be formed, we can describe them using a linear textual notation where numbers are used to represent the three types of nodes, then chains are simply written as words (*e.g.* 2312) while for cycles we indicate half-edges at both ends, *e.g.* ·123· is the triangle and ·123123· is the hexagon. We will use that shorthand notation in the sequel.

Our goal here is to be able to favour the formation at equilibrium of certain structures like the triangle by assigning a significantly negative energy to them (the convention is that a lower energy means a higher probability).

2. SITE GRAPH REWRITING

2.1. Site graphs and homomorphisms. A *site graph* G consists of finite sets of *agents* (nodes) and *sites* (connection slots), \mathcal{A}_G and \mathcal{S}_G , a partial function $\sigma_G : \mathcal{S}_G \rightarrow \mathcal{A}_G$, and a symmetric *edge* relation \mathcal{E}_G on \mathcal{S}_G . The pair $\mathcal{S}_G, \mathcal{E}_G$ form an undirected graph; sites not in the domain of \mathcal{E}_G are said to be *free*. The role of the additional map σ_G is to assign sites to agents; sites not in the domain of σ_G are said to be *dangling*, and will be used to represent half-edges (see below). Usually one also endows agents and/or sites with states (as we do in the example treated in §4); our main construction in §3 carries over readily to these.

One says G is *realizable* iff (i) sites have at most one incident edge; (ii) no dangling site is free; and, (iii) edges have at most one dangling site. Note that point (i) implies that no site has an edge to itself.

A *homomorphism* $h : G \rightarrow G'$ of site graphs is a pair of functions, $h_S : \mathcal{S}_G \rightarrow \mathcal{S}_{G'}$ and $h_A : \mathcal{A}_G \rightarrow \mathcal{A}_{G'}$, such that (i) whenever $h_A(\sigma_G(s))$ is defined, so is $\sigma_{G'}(h_S(s))$ and they are equal; and (ii) if $s \mathcal{E}_G s'$ then $h_S(s) \mathcal{E}_{G'} h_S(s')$.

A homomorphism $h : G \rightarrow G'$ is an *embedding* iff (i) h_A and h_S are injective; and (ii) if s is free in G , so is $h_S(s)$ in G' . If $h : G \rightarrow G'$ is an embedding and G' is realizable then G is also realizable.

$$\begin{array}{ccc}
 \mathcal{S}_G & \xrightarrow{h_S} & \mathcal{S}_{G'} \\
 \sigma_G \downarrow & \leq & \downarrow \sigma_{G'} \\
 \mathcal{A}_G & \xrightarrow{h_A} & \mathcal{A}_{G'}
 \end{array}$$

Site graphs and homomorphisms form a category \mathbf{SG} with the natural ‘tiered’ composition; embeddings form a subcategory; if in addition, we restrict objects to be realizable, we get the subcategory \mathbf{rSGe} of realizable site graphs and embeddings.

Returning to our chains-and-cycles example, we can use sites as a means to enforce on nodes our earlier constraint where a node can connect to at most one node of each other type. Here we present a possible encoding of a simple chain of 3 nodes:

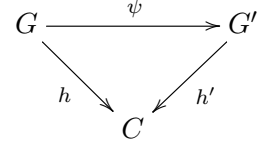
$$G = \begin{array}{c} \text{---} \text{---} \text{---} \\ \text{---} \end{array} \begin{array}{ccc} \textcircled{x} & \textcircled{y} & \textcircled{z} \\ \text{---} & \text{---} & \text{---} \\ \text{---} & \text{---} & \text{---} \end{array} \begin{array}{ccc} r_x & r_y & r_z \\ l_x & l_y & l_z \end{array}$$

$$\begin{aligned} \mathcal{A}_G &= \{x, y, z\}, & \sigma_G &= \{s_a \mapsto a : s \in \{l, r\}, a \in \mathcal{A}_G\}, \\ \mathcal{S}_G &= \{l_x, r_x, l_y, r_y, l_z, r_z\}, & \mathcal{E}_G &= \{(r_x, l_y), (l_y, r_x), (r_y, l_z), (l_z, r_y)\}. \end{aligned}$$

Note that this site graph is realizable and the l_x and r_z sites are free. To complete the encoding of our example we need types for nodes which we now introduce.

2.2. The category of site graphs over C . A homomorphism $h : G \rightarrow C$ is a *contact map* over C iff (i) G is realizable, (ii) σ_C is total and (iii) whenever $h_S(s_1) = h_S(s_2)$ and $\sigma_G(s_1) = \sigma_G(s_2)$, then $s_1 = s_2$. The third condition of local injectivity means that every agent of G has at most one copy of each site of its corresponding agent in C ; C is called the *contact graph*.

Hereafter, we work in the (comma) category \mathbf{rSGe}_C whose objects are contact maps over C , and arrows are embeddings such that the associated triangle commutes in \mathbf{SG} . We write $\Upsilon_C(h, h')$ for the set of such embeddings between h, h' contact maps over C ; we also write $|-|$ for the domain functor from \mathbf{rSGe}_C to \mathbf{rSGe} which forgets types. In particular, if $h : G \rightarrow C$ is a contact map, we write $|h|$ for its domain G .

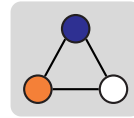


Note that in \mathbf{rSGe} , an embedding $h : G \rightarrow G'$ may map a dangling site s of G to any site s' of G' (provided h_S is injective). In particular, s' does not have itself to be dangling. This means that dangling sites can be used as an ‘any site’ wild card when matching G . In the typed case, *i.e.* in \mathbf{rSGe}_C , the contact map $h : G \rightarrow C$ tells us which agent in C the dangling site belongs to because σ_C is total, and this must be respected by h . Dangling sites can now be used as ‘binding types’ to express the property of being bound to the site s of an agent of a given type.

The contact graph C is fixed and plays the role of a *type*: it specifies the kinds of agents that exist, the sites that they may possess, and which of the $|\mathcal{S}_C|(|\mathcal{S}_C| + 1)/2$ possible edge types are actually valid. It also gives canonical names to the types of agents and their sites.

In our running example we use the triangle $C = \cdot 123 \cdot$ as contact graph:

$$\begin{aligned} \mathcal{A}_C &= \{1, 2, 3\}, & \mathcal{S}_C &= \{l_1, r_1, l_2, r_2, l_3, r_3\}, \\ \sigma_C &= \{s_a \mapsto a : s \in \{l, r\}, a \in \mathcal{A}_C\}, \\ \mathcal{E}_C &= \{(r_1, l_2), (l_2, r_1), (r_2, l_3), (l_3, r_2), (r_3, l_1), (l_1, r_3)\} \end{aligned}$$



This contact graph constrains agents to bind only agents of a different type. The local injectivity condition on contact maps restricts the number of sites available for each node to two. Our simple chain of three nodes from the previous section (§2.1) can thus be typed using a contact map $h : G \rightarrow C$ with $h_A = \{x \mapsto 1, y \mapsto 2, z \mapsto 3\}$ and $h_S = \{s_x \mapsto s_{h_A(x)} : s \in \{l, r\}, x \in \mathcal{A}_G\}$.

We now extend the shorthand notation introduced in §1.1 to cover the case when h_S is not locally surjective. Specifically, when a node does not mention its two allowed sites, we write ‘?’ for the missing site. A missing site can be introduced by an embedding and can be free or bound. Sites do not explicitly show in this notation but they are implicitly positioned at the left (l) and right (r) of each agent. Binding types are denoted by an exponent. Thus, for instance, in ${}^312?$ we have two agents u, v of type 1, 2, respectively, where u is bound to a dangling site of type r_3 on its l_u site and bound to v 's l_v site on its r_u site. Similarly, v is bound to u on its l_v site and does not have any r site. Using this extended notation we can write down our three rules in text: $?1 + 2? \rightleftharpoons ?12?$, $?2 + 3? \rightleftharpoons ?23?$, and $?3 + 1? \rightleftharpoons ?31?$.

Recall that the goal of this example is to control the type of cycles which one finds at equilibrium. To this effect, we introduce a set of energy patterns \mathcal{P} consisting of one contact map for each edge type (*i.e.* $?12?$, $?23?$, and $?31?$), and one for the triangle $\cdot 123 \cdot$. As we will see, a careful choice of energy costs for each pattern will indeed lead to a system producing almost only triangles in the long run (see Fig. 4 in Appendix A).

2.3. Multi-sums in \mathbf{rSGe}_C . The category \mathbf{SG} has all pull-backs, constructed from those in \mathbf{Set} ; it is easy to see that they restrict to \mathbf{rSGe}_C . \mathbf{SG} also has all push-outs and all sums, but these do not in general restrict to \mathbf{rSGe}_C . (Just as push-outs and sums in \mathbf{Set} do not restrict to the subcategory of injective functions.)

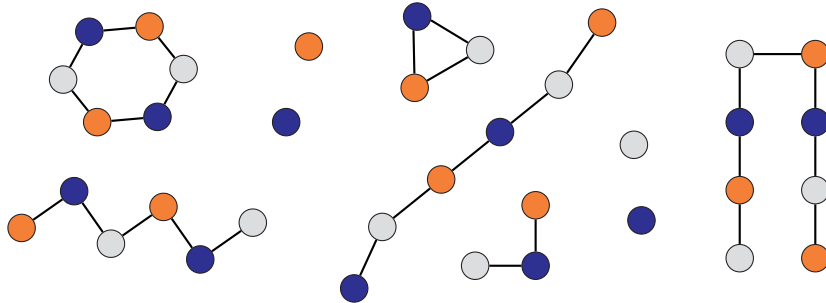
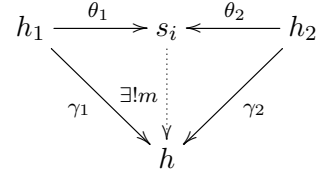


Figure 1: An example of a mixture of chains and $3n$ -gons which our three rules can reach starting with 12 agents of each type.

However, \mathbf{rSGe}_C has *multi-sums* meaning that, for all pairs of site graphs of type C , $h_1 : G_1 \rightarrow C$ and $h_2 : G_2 \rightarrow C$, there exists a family of co-spans $\theta_1^i : h_1 \rightarrow s_i \leftarrow h_2$, such that any co-span $\gamma_1 : h_1 \rightarrow h \leftarrow h_2 : \gamma_2$ factors through *exactly one* of the family and does so *uniquely*. In outline, given a co-span $\gamma_1 : h_1 \rightarrow h \leftarrow h_2 : \gamma_2$, we first take its pull-back (which is guaranteed to remain within \mathbf{rSGe}_C) and then the push-out of that: the fact that the original co-span exists implies that this push-out also remains within \mathbf{rSGe}_C . The multi-sum is then a choice of push-out for each (isomorphism class of) pull-back.

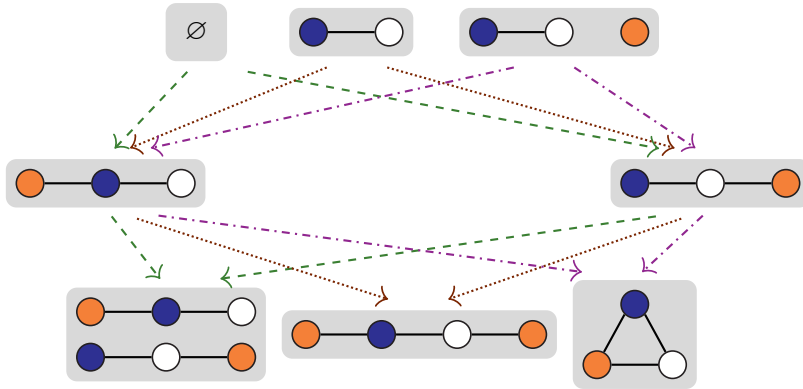


The idea is that the pairs θ_1^i, θ_2^i enumerate all minimal ways in which one can glue h_1 and h_2 , that is to say all the minimal gluings of G_1 and G_2 that respect C . There are *finitely* many, all of which factor through the standard sum in the larger slice category \mathbf{SG}_C .

The notion of multi-sum dates back to Ref. [12]; we will call them *minimal gluings* in \mathbf{rSGe}_C according to their intuition in this concrete context, and use them in §3.3 to construct balanced rules.

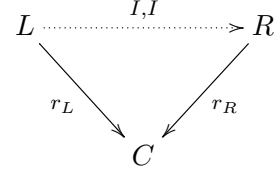
To illustrate this idea, let us consider, in the context of our example, the minimal gluings of $D_1 = ?123?$ and $D_2 = ?231?$. Computing them is a matter of computing all pull-backs. One such pull-back is $D_1 \leftarrow ?23? \rightarrow D_2$, which gives us the minimal gluing $?1231?$. On the other hand, the span $D_1 \leftarrow ?3? \rightarrow D_2$ is not a pull-back since for all cospans that can close the square, the span factors through $D_1 \leftarrow ?23? \rightarrow D_2$. This is a consequence of $?3?$ not being a maximal overlap of D_1 and D_2 : whenever 3 is contained in a possible overlap of D_1 and D_2 , 2 will also be there.

All minimal gluings of D_1 and D_2 are displayed in the following diagram with their respective pullbacks. The left square has the empty graph as pullback and $?123? + ?231?$ as pushout, the central square has $?23?$ and $?1231?$, and the right square has $?23? + ?1?$ and $?123?$. Each square uses arrows of different colour and style.



2.4. Rules. A rule r over C is a pair of contact maps $r_L : L \rightarrow C, r_R : R \rightarrow C$ which differ only in their edge structures, *i.e.* $\mathcal{A}_L = \mathcal{A}_R, \mathcal{S}_L = \mathcal{S}_R, \sigma_L = \sigma_R, r_{L,\mathcal{A}} = r_{R,\mathcal{A}}$ and $r_{L,\mathcal{S}} = r_{R,\mathcal{S}}$.

For example, the unbinding rule $g_{12} = ?12? \rightarrow ?1 + ?2$ may be represented as:



$$\begin{aligned} \mathcal{A}_L &= \mathcal{A}_R &= \{u, v\} \\ \mathcal{S}_L &= \mathcal{S}_R &= \{r_u, l_v\} \\ \sigma_L &= \sigma_R &= \{r_u \mapsto u, l_v \mapsto v\} \\ r_{L,\mathcal{A}} &= r_{R,\mathcal{A}} &= \{u \mapsto 1, v \mapsto 2\} \\ r_{L,\mathcal{S}} &= r_{R,\mathcal{S}} &= \{r_u \mapsto r_1, l_v \mapsto l_2\} \end{aligned}$$

with only the edge structures $\mathcal{E}_L = \{(r_u, l_v), (l_v, r_u)\}$ and $\mathcal{E}_R = \emptyset$ differing.

A contact map $h : G \rightarrow C$ is a *mixture* iff σ_G is total (no dangling edge) and h_S is locally surjective, *i.e.* for all $a \in \mathcal{A}_G, h_S(\sigma_G^{-1}(a)) = \sigma_C^{-1}(h_A(a))$. In words, a mixture is a *fully-specified* site graph with respect to the type C .

An embedding $\psi : r_L \rightarrow h$ induces a *rewrite* of the mixture h by modifying the edge structure of the image of ψ from that of r_L to that of r_R . The result of rewriting is a new mixture h^* , where $|h^*|$ has the same agents and sites as $|h|$, and an embedding $\psi^* : r_R \rightarrow h^*$.

$$\begin{array}{ccc} r_L & \xrightarrow{\quad\quad\quad} & r_R & (2.1) \\ \psi \downarrow & & \downarrow \psi^* & \\ h & \xrightarrow{\quad\quad\quad} & h^* & \end{array}$$

This type of rewriting can be formalized using double push-out rewriting [6], but with the simple rules considered here, there is no need. The inverse of r , defined as $r^* := (r_R, r_L)$ is also a valid rule; by rewriting h^* with r^* via ψ^* , we recover h and ψ .

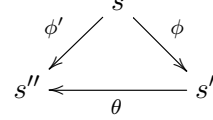
Given a finite set of rules \mathcal{G} over C , we define a labelled transition system $\mathcal{L}_{\mathcal{G}}$ on mixtures over C : a transition from a mixture h is a rewriting step determined and labelled by an *event* (r, ψ) , as in diagram (2.1), with r in \mathcal{G} and ψ in $\mathcal{Y}_C(r, h)$. We suppose hereafter that \mathcal{G} is closed under rule inversion, *i.e.* $\mathcal{G} = \mathcal{G}^*$. Hence, every (r, ψ) -transition has an inverse (r^*, ψ^*) , and $\mathcal{L}_{\mathcal{G}}$ is symmetric.

2.5. CTMC semantics. It is not difficult to see that for any rule $r, |\mathcal{Y}_C(r_L, h)| \leq |\mathcal{A}_{|h|}|^{d(r)}$ where $d(r)$ is the number of connected components in r_L . Hence, $\mathcal{L}_{\mathcal{G}}$ has finite out-degree, bounded by $|\mathcal{G}| \cdot |\mathcal{A}_{|h|}|^d$ for some d . Also, as agents are preserved by rules, the (strongly) connected components of $\mathcal{L}_{\mathcal{G}}$ are finite. Given a *rate map* k from \mathcal{G} to \mathbb{R}_+ , we can therefore equip $\mathcal{L}_{\mathcal{G}}$ with the structure of a time-homogeneous irreducible continuous-time Markov chain (CTMC), simply by assigning rate $k(r)$ to an event of the form (r, ψ) . We write $\mathcal{L}_{\mathcal{G}}^k$ for the CTMC thus obtained.

A finite time-homogeneous CTMC \mathcal{M} has *detailed balance* for a probability distribution π on \mathcal{M} 's state space iff, for all states x and $y, \pi(x) \cdot q(x, y) = \pi(y) \cdot q(y, x)$ where $q(x, y)$ is \mathcal{M} 's transition rate from x to y . This implies, assuming \mathcal{M} is irreducible, that π is the unique fixed point of the action of \mathcal{M} to which the probabilistic state of \mathcal{M} converges, regardless of the initial state.

In our case, the probability $\pi(x)$ will be proportional to $e^{-E(x)}$, where $E(x)$ is the energy of the system as defined by a set of patterns and their associated costs.

2.6. Extensions and rule refinement. We say an embedding $\phi : s \rightarrow s'$ is a *prefix* of $\phi' : s \rightarrow s''$ if there is some embedding $\theta : s' \rightarrow s''$ such that $\theta\phi = \phi'$ and write $\psi \leq \phi$ for this. We refer to a prefix of an epi $\phi : s \rightarrow s'$ as an *extension* of s . In the category of extensions of s , a morphism between objects $\phi : s \rightarrow s'$ and $\phi' : s \rightarrow s''$ is an embedding $\theta : s' \rightarrow s''$ such that the triangle commutes. If θ is an iso we write $\phi \cong_s \phi'$.



Epis of \mathbf{rSGe}_C are characterized as follows [25]: suppose $s : G \rightarrow C$ and $s' : G' \rightarrow C$ are contact maps then $\phi : s \rightarrow s'$ of \mathbf{rSGe}_C is an epi iff every connected component of G' contains at least one agent in the image of ϕ_A . Rule application preserves epis and in fact also prefixes of epis:

Lemma 2.1. *Let $r = (r_L, r_R)$ be a rule and $\phi : r_L \rightarrow x$ be an embedding with r_L, r_R , and x contact maps in \mathbf{rSGe}_C . The embedding $\phi^* : r_R \rightarrow x^*$ that results from applying r to ϕ is a prefix of an epi iff ϕ is.*

Proof. This amounts to proving that $\psi^* \geq \phi^*$ is an epi if $\psi \geq \phi$ is. For this it is enough to consider the case where the rule adds or deletes exactly one edge. Rules that modify more than one edge at a time can be decomposed as sequences of deletions and insertions of edges. Given that each deletion and insertion preserves the property, the sequence will preserve it as well. The case of adding an edge is easy, as the image of ψ^* has fewer connected components to “touch”. The case where r deletes an edge can introduce new connected components, however in this case both ends u, v of the deleted edge must be in r_L , so whether the deletion disconnects or not the codomain of ψ , the components of $\psi^*(u)$ and $\psi^*(v)$ will have a pre-image, namely u and v . \square

It follows that the category of extensions of r_L and r_R are isomorphic. Indeed, because we have chosen to restrict to reversible rules, the full categories under r_L and r_R are evidently isomorphic, and by the lemma above this correspondence preserves being prefix-of-an-epi; if ϕ is an extension of r_L , we will write ϕ^* for the corresponding extension of r_R .

A family of epis $\phi_i : s \rightarrow t_i$ *uniquely decomposes* s , or is a *refinement* of s , if, for all mixtures h and embeddings $\psi : s \rightarrow h$, there exists a unique i and ψ' such that $\psi = \psi'\phi_i$. This is the basic requirement for a reasonable notion of rule refinement: it guarantees that the LHS s of a given rule splits into a non-overlapping and exhaustive collection of more specific cases t_i .

In the next section, we will be constructing specific such decompositions in order to produce families of subrules that are compatible with energy patterns, *i.e.* each subrule should produce and consume the same number of energy patterns in each application, regardless of the particular embedding $\psi \in \mathcal{Y}_C(r_L, h)$ that the subrule is applied to.

To find such unique decompositions we first recall the growth policy method [25], which works by detailing which agents and sites should be added to s . Specifically, a *growth policy* Γ for s is a family of functions Γ_ϕ , indexed by all extensions $\phi : s \rightarrow t$, where Γ_ϕ maps $u \in \mathcal{A}_{|t|}$ to a subset $\Gamma_\phi(u)$ of $\sigma_C^{-1}(t_A(u))$, *i.e.* each agent in $|t|$ is allocated a subset of the set of sites belonging to the agent $t_A(u)$ it is mapped to in C . An agent in $|t|$ may cover some, or all, of these sites or even completely extraneous sites: if the former, *i.e.* for all u in $\mathcal{A}_{|t|}$, $t_S(\sigma_{|t|}^{-1}(u)) \subseteq \Gamma_\phi(u)$, we say that ϕ is *immature*; if for all u , the inclusion is an equality and ϕ is also an epi, we say that ϕ is *mature*; otherwise ϕ is said to be *overgrown*. The functions Γ_ϕ must satisfy, for all extensions ϕ and $\phi' \geq \phi$, the *faithfulness* property,

$\Gamma_\phi = \Gamma_{\phi'} \circ \psi_{\mathcal{A}}$ with ψ such that $\psi\phi = \phi'$; so a site requested by ϕ must be requested by any further extension. Additionally, this property forces Γ to eagerly ask for all sites that will be eventually requested at any given agent in the codomain of ϕ . If ϕ is not overgrown then no $\phi' \leq \phi$ is overgrown either. Also, note that the *union* of two growth policies is itself a growth policy.

Given an s and a growth policy Γ for s , we define $\Gamma(s)$ by choosing one representative per \cong_s -isomorphism class of the set of all extensions of s which are mature according to Γ .

The following theorem guarantees that factorizations through $\Gamma(s)$ are unique when they exist, but *not* that they necessarily do exist. In the next section, we will construct a specific growth policy of interest for which this property of exhaustivity of the decomposition can be proved by hand. As such, it fulfils our desired criteria of providing an exhaustive collection of mutually exclusive sub-cases.

Theorem 2.2. *Let s and x be contact maps and Γ a growth policy for s . If an embedding $\psi : s \rightarrow x$ in $\mathcal{T}_C(s, x)$ can be decomposed in two ways as $\gamma_1\phi_1$ and $\gamma_2\phi_2$ with $\phi_i : s \rightarrow t_i$ in $\Gamma(s)$ and $\gamma_i : t_i \rightarrow x$ then $\phi_1 = \phi_2$ and $\gamma_1 = \gamma_2$.*

Proof. Suppose that $\gamma_1\phi_1 = \gamma_2\phi_2$, where ϕ_1 and ϕ_2 are mature extensions of s according to Γ and $\phi_1 \neq \phi_2$. As shown in diagram (2.2), we have an inner square formed by the pull-back π_1, π_2 , and the minimal gluing θ_1, θ_2 of γ_1 and γ_2 . Also θ_1 and θ_2 are epis, as every connected component of m has a pre-image in t_1 or t_2 and so also in s , since the ϕ_i s are epis, and so also in the other of t_2 and t_1 .

$$\begin{array}{ccc}
 s & \xrightarrow{\phi_1} & t_1 \\
 \phi \searrow & & \nearrow \pi_1 \\
 & p & \\
 \pi_2 \swarrow & & \searrow \theta_1 \\
 & m & \\
 \theta_2 \swarrow & & \searrow \\
 t_2 & \xrightarrow{\gamma_2} & x \\
 \phi_2 \swarrow & & \searrow \gamma_1
 \end{array} \tag{2.2}$$

If θ_1, θ_2 are not both isomorphisms then there must be a pair u, z , consisting of a node in m with pre-images u_1, u_2 in t_1, t_2 and a site z of u , such that z has no pre-image in exactly one of θ_1, θ_2 . Say it is θ_2 . Since ϕ_1 is not overgrown, $z \in \Gamma_{\phi_1}(u_1)$ and, by faithfulness, $z \in \Gamma_\phi((u_1, u_2))$, where (u_1, u_2) is the pull-back pre-image of u_1 and u_2 . So again, by faithfulness, $z \in \Gamma_{\phi_2}(u_2)$ which contradicts our original assumption. So θ_1 and θ_2 are isos. It follows that $\phi_1 = \phi_2$ as there is only one representative per \cong_s -isomorphism class in $\Gamma(s)$. Finally, $\gamma_1 = \gamma_2$ because ϕ_1 is an epi. \square

Given a rule r and an extension $\phi : r_L \rightarrow t$ of r_L , we write r_ϕ for the refined rule associated to ϕ ; that is, r_ϕ is the pair (t, t^*) with t^* the codomain of the extension ϕ^* corresponding to ϕ . Given Γ a growth policy for r_L , we write $\Gamma(r)$ for the family of rules obtained by refining r according to Γ ; that is $\Gamma(r)$ is the family of rules r_ϕ , for ϕ ranging in $\Gamma(r_L)$.

An example of growth policy is the *ground* policy which assigns all possible sites to all agents. In this case, $\Gamma(s)$ is simply the set, possibly infinite, of all epis of s into mixtures, considered up to \cong_s ; and $\Gamma(r)$, the ground refinement of r , contains all refinements of r along those epis, which therefore directly manipulate mixtures. It is easy to see that the ground refinement for our running example is infinite, since each of the three rules can trigger the extension of a chain of any length.

3. RULE GENERATION

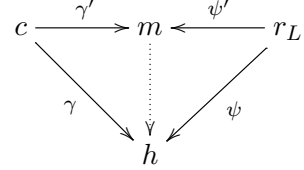
We fix a finite set \mathcal{G} of *generator* rules; and a finite set \mathcal{P} of connected contact maps in \mathbf{rSGe}_C ; these are our *energy patterns*.

A canonical set of generator rules can be derived from the contact graph C by constructing a pair of binding/unbinding rules (and a full set of state changes if we were to use site graphs with states as we do in applications) for each edge in \mathcal{E}_C . This set is maximally general in that each generator rule asks for the least possible context to trigger a binding or unbinding event. This is the set which we have chosen to use in our running example. In the next example which we present in §4, we will consider a strict subset of this canonical generating set. But, in general, there is no prescription regarding which rules one decides to incorporate to \mathcal{G} .

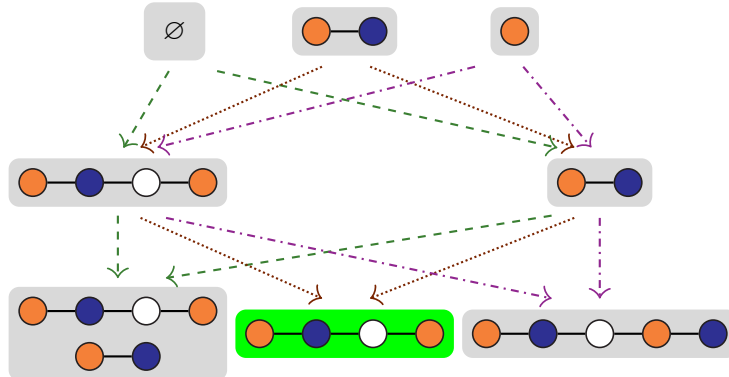
The goal is now to refine \mathcal{G} into a new rule set $\mathcal{G}_{\mathcal{P}}$ where each refined rule is \mathcal{P} -*balanced*, which means that, however applied, it consumes or produces a fixed amount of each c in \mathcal{P} . The construction proceeds in two steps. Firstly, we characterize balanced refinements; secondly, we define a specific growth policy with balanced mature extensions. Using Th. 2.2, we show that these mature extensions obtain a proper finite refinement of \mathcal{G} .

Note that ground extensions of g are trivially balanced but, in general, the ground refinement is impractically large or even infinite; ours will always be finite.

3.1. Consumption and production of patterns. Consider c in \mathcal{P} and a rule r . For an r -event ψ to *consume* an instance γ of c in a mixture h , γ_S and ψ_S must have images which intersect on at least one site which is modified by r (by adding an edge if it was free, or by deleting or changing an edge it was part of). This is the case iff the associated minimal gluing (γ', ψ') (obtained by restricting the co-span to the union of its images in h) has the same property. Likewise, for an r -event to *produce* an instance of c , the associated minimal gluing between c and r_R must have a modified intersection. We call such minimal gluings *relevant*; they are the ones which underlie events that can affect the instances of c .



This notion can be illustrated by looking at the minimal gluings of the left-hand side of the unbinding rule $?12? \rightarrow ?1 + 2?$ and energy pattern $?1231?$ (supposing for a moment that we had $?1231?$ in \mathcal{P}):



Of these, only the central square is a relevant minimal gluing since the image of the edge consumed by the rule is contained in the image of the energy pattern in the minimal gluing.

3.2. \mathcal{P} -balanced extensions. Given g in \mathcal{G} with LHS g_L and $\phi : g_L \rightarrow t$ an extension of g_L , we say that ϕ is *\mathcal{P} -left-balanced* iff, for all relevant minimal gluings $\gamma : c \rightarrow m \leftarrow t : \theta$ with $c \in \mathcal{P}$, θ is an isomorphism. This means that the image of c under γ is contained in t . Symmetrically, one says that ϕ is *\mathcal{P} -right-balanced* iff ϕ^* is a \mathcal{P} -left-balanced extension of r^* . An extension ϕ is *\mathcal{P} -balanced* iff it is \mathcal{P} -left- and \mathcal{P} -right-balanced. We say that a balanced extension ϕ is *prime* iff it is minimally so, *i.e.* any prefix of ϕ that is \mathcal{P} -balanced is isomorphic to ϕ (as an extension). Prime extensions are epis since erasing an ‘untouched’ connected component in the codomain preserves balance.

$$\begin{array}{ccc}
 & & g_L \\
 & & \downarrow \phi \\
 c & & t \\
 \searrow \gamma & \simeq & \swarrow \theta \\
 & m &
 \end{array} \quad (3.1)$$

If ϕ is a \mathcal{P} -balanced extension of g , the refined rule g_ϕ has a *balance vector* in $\mathbb{Z}^{\mathcal{P}}$, written $\Delta\phi$, where, for each $c \in \mathcal{P}$, $\Delta\phi(c)$ is the number of copies of c produced by *any* g_ϕ -event, which is also the difference between the number of embeddings of c in the RHS and the LHS of g_ϕ . In other words, for any mixture h , balance of ϕ guarantees $\Delta\phi(c) = |\mathcal{Y}_C(c, h^*)| - |\mathcal{Y}_C(c, h)| = |\mathcal{Y}_C(c, g_{\phi,R})| - |\mathcal{Y}_C(c, g_{\phi,L})|$. Conversely, if c in \mathcal{P} violates the condition of diagram (3.1), well-chosen applications of g_ϕ will result in different and context-dependent $\Delta\phi(c)$. Thus, the notion of balanced extension characterizes the property that we want.

To illustrate these definitions consider the unbinding rule $g_{12} = ?12? \rightarrow ?1 + 2?$ with the trivial extension $\phi = \mathbf{1}_{r_L}$ (the identity arrow of the left-hand side of rule g_{12}). Define $\mathcal{P}_1 = \{?12?, ?23?, ?31?\}$ and $\mathcal{P}_2 = \{.123.\}$. On the one hand, ϕ is \mathcal{P}_1 -balanced. Of all possible gluings of patterns in \mathcal{P}_1 with ϕ 's codomain, only the trivial one of $?12?$ with itself is relevant to g_{12} . On the other hand, ϕ is not \mathcal{P}_2 -balanced, because there is a gluing between $.123.$ and $?12?$ mapping the 12 edge to itself in the triangle. This gluing is relevant since applying g_{12} breaks the triangle; and clearly $\theta : ?12? \rightarrow .123.$ is not an iso. Failure to be balanced reflects in the fact that different applications of g_{12} incur different values of $\Delta\phi(.123.)$: namely 1 or 0 depending on whether the broken edge belongs to a triangle.

The ideas introduced now are discussed in a detailed example in §3.5.

3.3. Absorb or Avoid. We have just seen how the property of being \mathcal{P} -balanced characterizes, among all extensions ϕ of g_L , those with an unambiguous energy balance. The next step is to define a growth policy with a set of mature extensions which are balanced and form a unique decomposition of g_L . If we return to diagram (3.1) in the case where θ is not an isomorphism, the natural idea is to request the addition to the codomain of ϕ of these sites which are in m and belong to nodes in the image of $\theta_{\mathcal{A}}$. In this way, we force ϕ to grow further and make a decision about whether it wants to absorb the image of c in m , or would rather avoid this by growing in a way that is incompatible with the γ, θ gluing.

Some care is needed to ensure faithfulness, *i.e.* $\Gamma_\phi = \Gamma_{\phi'\phi} \phi'_{\mathcal{A}}$, since relevant minimal gluings on ϕ can disappear along a further extension ϕ' so that a site that was requested at ϕ may no longer be so after at $\phi'\phi$. To address this, we add site requests from all relevant minimal gluings in the *past* of an extension.

$\Gamma^{\mathcal{P}}$ since, by construction, it has all the sites requested by ϕ_y and so, by faithfulness, all those requested by ϕ_z .

Non-empty: Clause (i) guarantees that we request at least the sites in g which implies that g is not overgrown.

Balanced: If $\phi \in \Gamma^{\mathcal{P}}(g)$ is not balanced then there must be some relevant minimal gluing inducing a further site request, hence ϕ cannot be mature.

Finite: A request for a site a at some node in an extension $\phi : g_L \rightarrow t$, or $\phi^* : g_R \rightarrow t$, originates from a relevant minimal gluing of some c in \mathcal{P} with a prefix ϕ_1 of ϕ . Because this gluing is relevant, it must be that a is at a distance from the image of g_L in the codomain of ϕ_1 which is at most $\delta(c)$, the diameter of c (else c would not intersect the image of g_L). The same bound holds in the codomain of ϕ , as distances can only contract by further extension. Therefore any site requested in t has a distance to the image $\phi(g_L)$ which is bounded by $\max_{c \in \mathcal{P}} \delta(c)$. If ϕ is not overgrown, this sets a bound on the diameter of t . Hence there are finitely many mature extensions. \square

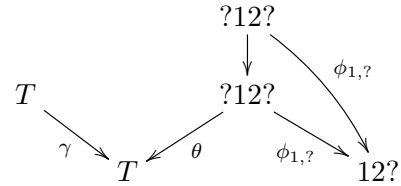
Therefore, given \mathcal{G} and \mathcal{P} , we obtain a finite \mathcal{P} -balanced rule set which refines \mathcal{G} exhaustively, by setting $\mathcal{G}_{\mathcal{P}} := \dot{\cup}_{g \in \mathcal{G}} \Gamma^{\mathcal{P}}(g)$ (disjoint sum). To every refinement g_{ϕ} , corresponds an inverse refinement g_{ϕ^*} ; hence, $\mathcal{G}_{\mathcal{P}} = \mathcal{G}_{\mathcal{P}}^*$ is closed under inversion like \mathcal{G} .

3.4. Other options considered. As said earlier, there are other ways to generate a balanced rule set. Beyond the ground extensions, one idea is to use *primes*, *i.e.* minimal balanced extensions. Clearly, prime extensions factorize all balanced ones -including the ones we generate by using the absorb-or-avoid growth policy presented above. The problem is that this factorization is not unique: prime extensions do not define in general a valid refinement as distinct sub-cases may overlap.

Another idea is to obtain balanced rules by gluing \mathcal{P} directly onto a generator rule g , in all possible maximal combinations, rather than on all extensions of g as in the above growth policy. This always reveals enough of the context in which g is applied so as to get \mathcal{P} -balanced refinements because, by definition, no additional form can be glued further. The problem with this approach is the opposite to that with primes: it does not in general build *enough* rules; so the refinement is valid but not exhaustive.

3.5. Triangles in detail. We now discuss the concepts introduced in this section using the example of the unbinding rule $g_{12} = ?12? \rightarrow ?1 + 2?$. Of the four patterns in \mathcal{P} , we need only consider the triangle $T = \cdot 123 \cdot$ as the others all trivially glue or trivially don't.

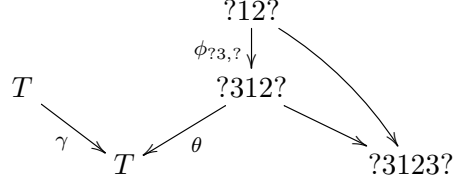
We apply our growth policy to the extension $\phi_{1,?} : ?12? \rightarrow 12?$ whereby we reveal a new free site in 1. The relevant minimal gluing $T \rightarrow T \leftarrow ?12?$ disappears in this extension, as the new site is bound in T . However, by rewinding back to $?12?$, our growth policy requires us to reveal additionally the second site of 2; as such, $\phi_{1,?}$ is not mature despite being balanced (indeed minimally so).



Intuitively, the growth policy forces us to reveal additional sites of $?12?$ because an instance of g_{12} may or may not consume a triangle and so has an ambiguous energy balance.

The definition of the growth policy makes it explicit that we have to reveal both of the hidden sites of $?12?$ and so there are four sub-cases to consider: 12 , 312 , 12^3 and ${}^312^3$ (using the notation introduced earlier for binding types). The first three cases all avoid the triangle, thereby having definite energy balance, and are moreover mature with respect to the growth policy; however, the case of ${}^312^3$ requires further analysis as it covers both the case of a triangle and of any chain of length at least 4.

Consider then the case of the extension to $?3123?$ where the two binding types of ${}^312^3$ have been embodied in two *distinct* 3 agents. Any rewind of interest has to go back far enough to glue T ; there are two maximal such extensions: $\phi_{?3,?} : ?12? \rightarrow ?312?$ and $\phi_{?,3?} : ?12? \rightarrow ?123?$. The former requests the hidden site of the left-hand 3 agent of $?3123?$; the latter (diagram not shown) requests the hidden site of the right-hand 3 agent. As such, $?3123?$ is not mature with respect to the growth policy and we must expand it into its four sub-cases: 3123 , 23123 , 3123^1 , ${}^23123^1$.



Our final refinement, according to the growth policy, consists in the following extensions 12 , 312 , 12^3 , $.123.$, 3123 , 23123 , 3123^1 , ${}^23123^1$.

If were to glue only on $?12?$, not on all extensions thereof, we would only generate the rule $T \rightarrow 231$, *i.e.* the case where the bond happens to belong to a triangle. Clearly, this is a valid refinement but is far from covering all cases. Note also that the extension $\phi_{1,2} : ?12? \rightarrow 12$ is not a minimal balanced extension as it factors through either $\phi_{1,?} : ?12? \rightarrow 12?$ or $\phi_{?,2} : ?12? \rightarrow ?12$ which are both minimal. This illustrates the problem with prime extensions discussed earlier: the candidate refinement $\{12?, ?12\}$ is not valid precisely because it leads to an ambiguous decomposition of the case of $\phi_{1,2}$ and will therefore generate rates that are different for instances of g_{12} with the same balance vector $\Delta\phi$.

On the other hand, the subrules 3123 , 23123 , 3123^1 , and ${}^23123^1$ all share the same balance vector. In fact, they can be summarized by a single prime extension $\phi_{?3,3?} : ?12? \rightarrow ?3123?$ which, together with $\phi_T : ?12? \rightarrow T$, mutually exclusively and exhaustively refines $\phi_{3123} : ?12? \rightarrow {}^312^3$. Our growth policy is too local to see this optimisation, and has to refine $\phi_{?3,3?}$ into the above four explicit subcases.

That said, the output of the growth policy could be either optimized by hand or indeed subjected to the procedure of rule compression which would automatically perform this optimization for us. A working Kappa model where the generators of this example have been fully expanded according to the growth policy and then manually compressed can be found in Appendix A.

3.6. Rates. To equip $\mathcal{G}_{\mathcal{P}}$ with rates, we suppose given a \mathcal{P} -indexed real-valued vector of *energy costs* ϵ , and a rate map $k : \mathcal{G}_{\mathcal{P}} \rightarrow \mathbb{R}_+$ such that, for all g_{ϕ} in $\mathcal{G}_{\mathcal{P}}$:

$$\log k(g_{\phi^*}) - \log k(g_{\phi}) = \epsilon \cdot \Delta\phi \quad (3.3)$$

with $\Delta\phi$ in $\mathbb{Z}^{\mathcal{P}}$, the balance vector of the refined rule g_{ϕ} with respect to \mathcal{P} , a well-defined quantity by Th. 3.1.

We write $\mathcal{P}(x)$ for the \mathcal{P} -indexed vector which maps c to $|\mathcal{T}_C(c, x)|$, and define the *energy* $E(x)$ of x as $\epsilon \cdot \mathcal{P}(x)$, *i.e.* the sum over all $c \in \mathcal{P}$ of the product of the number of

instances of c in x and the energy cost of c . We also write $\mathcal{L}_{\mathcal{G}}(x)$ for the finite (strongly) connected component of x in $\mathcal{L}_{\mathcal{G}}$, and define a probability distribution (in Boltzmann format) π_x on $\mathcal{L}_{\mathcal{G}}(x)$ by:

$$\pi_x(y) := \frac{e^{-\epsilon \cdot \mathcal{P}(y)}}{\sum_{z \in \mathcal{L}_{\mathcal{G}}(x)} e^{-\epsilon \cdot \mathcal{P}(z)}} \quad (3.4)$$

We can now prove our main theorem.

Theorem 3.2. *Let \mathcal{G} , \mathcal{P} , $\mathcal{G}_{\mathcal{P}}$, k , and π_x be defined as above; then (i) $\mathcal{L}_{\mathcal{G}_{\mathcal{P}}}$ and $\mathcal{L}_{\mathcal{G}}$ are isomorphic as symmetric LTSs; and (ii) for any mixture x , the irreducible time-homogeneous continuous-time Markov chain $\mathcal{L}_{\mathcal{G}_{\mathcal{P}}}^k$ has detailed balance for, and converges to, π_x on $\mathcal{L}_{\mathcal{G}_{\mathcal{P}}}(x)$.*

Proof. Both $\mathcal{L}_{\mathcal{G}}$ and $\mathcal{L}_{\mathcal{G}_{\mathcal{P}}}$ offer transitions from a mixture x : the former are labelled by pairs (g, ψ) with g in \mathcal{G} , ψ in $\mathcal{Y}_C(g_L, x)$; the latter by pairs (g_ϕ, γ) with g_ϕ the refinement of g along a mature extension $\phi : g_L \rightarrow t$, and γ in $\mathcal{Y}_C(t, x)$. Steps in the latter can be mapped to steps in the former by transforming labels as follows: $(g_\phi, \gamma) \mapsto (g, \gamma\phi)$. As $\mathcal{G}_{\mathcal{P}}$ refines \mathcal{G} exhaustively (Th. 3.1), this correspondence is a bijection, which establishes the first claim.

(Pedantically, there is a full and faithful functor between the two corresponding free categories which is the identity on objects —incidentally, this bijection is readily seen to respect the symmetries on labels.)

Since we have multiple rules in $\mathcal{L}_{\mathcal{G}_{\mathcal{P}}}$, each of which can be applied in several ways, there can be more than one transition from x to the same y —each uniquely described by a (g_ϕ, γ) label. Each such (g_ϕ, γ) has an inverse, (g_{ϕ^*}, γ^*) , where: g^* is the rule inverse to g ; ϕ^* corresponds to ϕ in the isomorphism between the categories of extensions of x and y , with $\phi_{\mathcal{A}} = \phi_{\mathcal{A}}^*$; and γ^* is the embedding corresponding to γ , also with $\gamma_{\mathcal{A}} = \gamma_{\mathcal{A}}^*$. One can easily verify that ϕ^* is an epi, and that ϕ^* is also mature. Hence (g_{ϕ^*}, γ^*) determines a valid transition in $\mathcal{L}_{\mathcal{G}_{\mathcal{P}}}$ which is inverse to (g_ϕ, γ) , and we have a bijective correspondence between transitions from x to y and those from y to x .

Consider a pair e, e^* of such corresponding events due to g_ϕ and g_{ϕ^*} ; because e is a transition from x to y , and ϕ is \mathcal{P} -balanced (Th. 3.1), we have $\mathcal{P}(y) = \mathcal{P}(x) + \Delta\phi$, and hence $\epsilon \cdot \Delta\phi = \epsilon \cdot (\mathcal{P}(y) - \mathcal{P}(x))$; so, by (3.3), the rates of e, e^* are such that:

$$k(e^*) e^{-\epsilon \cdot \mathcal{P}(y)} = k(e) e^{-\epsilon \cdot \mathcal{P}(x)}$$

and by summing this equation over all pairs, we obtain detailed balance for the probability local to the component $\mathcal{L}_{\mathcal{G}_{\mathcal{P}}}(x) = \mathcal{L}_{\mathcal{G}_{\mathcal{P}}}(y)$, defined above as $\pi_x = \pi_y$, since:

$$q(y, x) e^{-\epsilon \cdot \mathcal{P}(y)} = q(x, y) e^{-\epsilon \cdot \mathcal{P}(x)}$$

The convergence statement follows by standard results on finite CTMCs. \square

Note that the subset of the state space which is reachable from x in $\mathcal{L}_{\mathcal{G}}$, namely $\mathcal{L}_{\mathcal{G}}(x)$, is finite; hence, the *partition function* $Z(x) := \sum_{z \in \mathcal{L}_{\mathcal{G}}(x)} e^{-E(z)}$ which figures in the denominator of π_x is also finite. In the presence of rules which increase the number of agents, the components $\mathcal{L}_{\mathcal{G}}(x)$ can be infinite and $Z(x)$ may diverge. For (mass action stochastic) Petri nets, convergence is guaranteed if detailed balance holds, but it is not true in general for Kappa [7, 11].

Another point worth making is that the result holds symbolically —regardless of the energy costs ϵ . So ϵ can be seen as a set of parameters, an ideal support for machine learning techniques if one were contemplating fitting a model to data.

3.7. A linear kinetic model. We now ask: what of the *actual* rates of $\mathcal{L}_{\mathcal{G}\mathcal{P}}^k$? Among all possible choices which accord with (3.3), we delineate a tractable subset whose size grows quadratically in $|\mathcal{P}|$. This is a useful log-linear heuristic, which is common in machine learning, but has no particular claim to validity.

Suppose we have, for every generating rule g in \mathcal{G} , a constant $c_g \in \mathbb{R}$, and a matrix A_g of dimension $|\mathcal{P}| \times |\mathcal{P}|$. Subject to the constraints that $c_{g^*} = c_g$, and $A_{g^*} + A_g = I$, we can define a log-affine rate map which satisfies (3.3) by:

$$\log(k(g_\phi)) := c_g - A_g(\epsilon) \cdot \Delta\phi \quad (3.5)$$

The kinetic model expressed in (3.5) requires of the order of $|\mathcal{P}|^2 \times |\mathcal{G}|$ parameters. In practice, one needs even fewer parameters, as only those energy patterns that are relevant to a given g , *i.e.* have non-zero balance for at least one rule in $\Gamma^{\mathcal{P}}(g)$, need to be considered when building A_g . Typically, for larger models, this will be a far smaller number than $|\mathcal{P}|$. This relative parsimony is compounded by the fact that the number of *independent* parameters will be often lower, because the $\Delta\phi$ family often has low rank. It is to be compared with the total number of choices which is far greater as it is of the order of the number of refinements, that is to say $\sum_{g \in \mathcal{G}} |\Gamma^{\mathcal{P}}(g)|$.

If we set $c_{g^*} = c_g = 0$, $A_{g^*} = 0$, $A_g = I$, we get: $k(g_\phi) = e^{-\epsilon \cdot \Delta\phi}$, $k(g_{\phi^*}) = 1$. As $\epsilon \cdot \Delta\phi$ is the difference of energy between the target and source in any application g_ϕ , this choice amounts to being exponentially reluctant to climb up the energy gradient. This is a continuous-time version of the celebrated Metropolis algorithm [24].

Another particular case, completely symmetric and which can be a reasonable choice to begin with, is obtained for $A_{g^*} = A_g = I/2$:

$$\begin{aligned} k_g(\phi) &= C_g e^{-\epsilon \cdot \Delta\phi/2} \\ k_{g^*}(\phi^*) &= C_g e^{\epsilon \cdot \Delta\phi/2} \end{aligned}$$

with $C_g = e^{c_g}$. As $\Delta\phi^* = -\Delta\phi$, this is indeed a symmetric definition.

Finally, it is fun to draw a comparison between the ascription given in (3.5) and the Arrhenius rate law. This law posits a dependency of the rate constant k of a reaction of the form $\log k = c - E_a/kT$, where c is a constant (defining the basic time scale of the reaction), E_a is the so-called *activation energy* of the reaction and T is the temperature. In our case, we are not concerned with the effect of T on the (logarithm of the) rate but with the effect of consuming and producing various energy patterns in \mathcal{P} at the locus of the instance of the generator rule g . In this view of things, (3.5) posits that the ‘activation energy’ of ϕ depends linearly on the cost of the various patterns and the balance of ϕ .

3.8. Energy functions do not need to be linear. We now return to a key assumption made in the preceding section and consider a more general situation where the energy function E is no longer asked to be linear. For reasons to become clear shortly, we still assume the much weaker property that E can be factored as $v \circ \mathcal{P}(-)$ for some finite set of patterns \mathcal{P} .

$$\mathbf{rSGe}_C \xrightarrow{\mathcal{P}(-)} \mathbf{mSet}(\mathcal{P}) \xrightarrow{v} \mathbb{R} \quad (3.6)$$

Note that if we see multi-sets as equipped with the usual point-wise partial order, $\mathcal{P}(-)$ is evidently functorial.

As an example, consider a divalent agent with contact graph $X(a^1, b^1)$ —where the shared superscript symbolizes an edge between sites a and b — with which one can form only chains and cycles, and a single generator g which can create/delete the unique edge type. Write c_3 for a cycle of length 3 (a triangle), and t_2 for an open chain of length 2. We define the quadratic energy function $E(x) = |\Upsilon_C(c_3, x)|^2$, *i.e.* $v(n) = n^2$. Applying g forward to a chain t_2 in a site graph of the form $x = t_2 + x'$ will create a new copy of c_3 , and give the following energy balance:

$$\Delta E = (|\Upsilon_C(c_3, x)| + 1)^2 - (|\Upsilon_C(c_3, x)|)^2 = 2|\Upsilon_C(c_3, x)| + 1$$

Therefore, detailed balance forces the log-ratio \hat{K}_g of the backward/forward rates assigned to an edge creation to depend on x . This is unlike the case of linear potentials where this ratio is independent of x . However, this extension g_ϕ of g (where it is applied to a chain t_2) is balanced with respect to \mathcal{P} . This means, as we have seen, that the stoichiometric \mathcal{P} -vector $\Delta\phi$ associated to g_ϕ —where each component $\Delta\phi(c)$ is defined as the difference of $|\Upsilon_C(c, y)| - |\Upsilon_C(c, x)|$ for a g -transition from x to y — is the same for all x, y . In the example $\Delta\phi$ has only one component, which is constant $\Delta\phi(c_3) = 1$ because the binding of the two *free* extremes of an open chain t_2 can only produce one triangle, regardless of the context in which the refined rule g_ϕ is applied.

In general, one can visualize the situation as follows.

$$\begin{array}{ccc} x & \xrightarrow{g} & y \\ \mathcal{P} \downarrow & & \downarrow \mathcal{P} \\ \mathcal{P}(x) & \xrightarrow{+\Delta\phi} & \mathcal{P}(y) \\ v \downarrow & & \downarrow v \\ \mathbb{R} & \xrightarrow{+\Delta E} & \mathbb{R} \end{array}$$

and detailed balance amounts to asking for $\hat{K}_g = v(\mathcal{P}(x) + \Delta\phi(x)) - v(\mathcal{P}(x))$. If v happens to be linear then this is the usual condition $\hat{K}_g = v(\Delta\phi(x))$. If v is not linear, the constraint does not seem very helpful as *a priori* one has to know x to compute the right hand side. But by the assumption (3.6) opening the paragraph, $\Delta\phi$ factors through \mathcal{P} , hence:

$$\hat{K}_g = v(\mathcal{P}(x) + \sigma_g(\mathcal{P}(x))) - v(\mathcal{P}(x)) =: \psi_g \circ \mathcal{P}(x)$$

where the second equation defining ψ_g uniquely as a real-valued function from \mathcal{P} -multisets. With this rewriting, it is plain that the constraint depends not on full knowledge of x , but only on $\mathcal{P}(x)$. Equivalently, we see that \hat{K}_g factors through $\mathcal{P}(\cdot)$ just like E . In the example, $\hat{K}_g = 2|\Upsilon_C(c_3, x)| + 1$, and $\psi_g(n) = 2n + 1$.

This is good enough to define rates for a balanced g_ϕ . For example, by analogy with the earlier linear kinetic model, we can choose log-rates (seen as real-valued multi-set functions) as follows:

$$\hat{k}_g = \alpha_g - \beta_g \psi_g \tag{3.7}$$

with α_g, β_g real-valued functions on \mathcal{P} -multisets such that $\alpha_{g^*} = \alpha_g$ and $\beta_{g^*} + \beta_g = 1$. This assignment solves the above constraint as, clearly, $\psi_{g^*} + \psi_g = 0$.

From the *simulation* point of view, this added generality requires two things: (i) that rates can be made to depend explicitly on observables; (ii) that the internal state of the simulation be extended to incorporate $\mathcal{P}(x)$. Both possibilities are already generically available in the current version of the main Kappa simulator KaSim [4]. A slight modification of the engine (not implemented) could obtain direct updates to $\mathcal{P}(x)$ as, by assumption, applying g_ϕ leads to a constant $+\Delta\phi$ update; and the same holds for propagating these updates to the rates of the rules which depend on them, *e.g.* as in (3.7). Thus, the complexity properties of the simulation algorithm are preserved [10].

4. ALLOSTERIC RING

We now put our methodology to use on a realistic example of a bacterial flagellar engine. In this section, we will use the traditional syntax of Kappa to denote site graphs: subscripts for states and shared superscripts for edges between sites, *e.g.* $A(x_0^1), B(y^1)$. Unlike the mathematical definitions of §2, agent and site types are indicated as explicit labels. Again, we use KaSim (the standard Kappa engine) for the simulation shown below.

4.1. The model. The engine can rotate clockwise or anti-clockwise at high angular velocities, and this will decide whether the bacterium tumbles or swims forward. One can build a simple model of the switch between the two modes [2]. The engine is seen as a ring of n identical components, P , with two possible conformations, 0 and 1. (In reality, each of the $n = 34$ component protomers is itself a tiny complex made of different subcomponents, but the model ignores this.) A ring homogeneously in state 0 (1) rotates (anti-) clockwise and induces tumbling (straight motion). Importantly, neighbouring P s on the ring prefer to have matching conformations. States of the ring with many mismatches thus incur high penalties. In the absence of any Y molecule binding a P , its favoured conformation is 0; conversely, in the presence of a Y , P favours 1. (Y stands for a small diffusible protein named *CheY*.) To bind, Y has to be activated by an external signal. Hence the switch can be triggered by a sudden activation of Y which then binds the ring and induces a change of regime. The sharper the transition between the two regimes the better.

As each of the P s can be in four states, the ring has on the order of 10^{20} non-isomorphic configurations which precludes any reaction-based (*e.g.* Petri nets) approach to the dynamics where each global state is considered as one chemical species. At this stage, we could apply the rule-based approach, or, better, we can obtain the rules *indirectly* by applying the methodology of §3. This is what we now do informally.

First, we define our contact graph with two agent types: $P(x, y, f_{0,1}, s)$ with domains x, y to form the ring, s to bind its signal Y , and f a placeholder for P 's conformation; $Y(s_{u,p})$ with two internal states to represent activity.

Second, we capture the informal statements in the discussion above by defining the energy patterns and associated costs. Note that the various motifs overlap. Following §3, we associate to each ring configuration x the occurrence vector $\mathcal{P}(x)$ and total energy $\epsilon \cdot \mathcal{P}(x)$. For example, a ring of size n uniformly in state 0 and with no bound Y s has

<i>Motif</i>	<i>Cost</i>
$P(f_i, x^1), P(y^1, f_j)$	ϵ_{ij}^{PP}
$P(f_i)$	ϵ_i^P
$P(f_i, s^1), Y(s^1)$	ϵ_i^{PY}

total energy $n(\epsilon_{00}^{PP} + \epsilon_0^P)$. This, in turn, defines the equilibrium distribution of the ring, namely x has probability proportional to $\exp(-\epsilon \cdot \mathcal{P}(x))$. (The convention is that the lower the energy, the likelier the state.)

In order to complete our energy landscape, we need to pick energy costs which reward or penalize local configurations as discussed above: the role of (4.1) is to align the internal states of neighbours on the ring—an Ising penalty term for mismatching neighbours which will “spread conformation”; (4.2) makes 0 the favoured state, while (4.3), which kicks in only in the presence of Y , makes 1 the favoured state.

$$\epsilon_{00}^{PP}, \epsilon_{11}^{PP} < \epsilon_{10}^{PP}, \epsilon_{01}^{PP} \quad (4.1)$$

$$\epsilon_0^P < \epsilon_1^P \quad (4.2)$$

$$\epsilon_0^{PY} > \epsilon_1^{PY} \quad (4.3)$$

The next step is to create the dynamics. The naive rule b for PY binding:

$$b := P(s), Y(s_p) \leftrightarrow P(s^1), Y(s_p^1)$$

has a ΔE which is ambiguous as it will be either ϵ_0^{PY} or ϵ_1^{PY} depending on its instances; hence, we have no hope of assigning rates to this rule that satisfy detailed balance—unless $\epsilon_0^{PY} = \epsilon_1^{PY}$, which contradicts (4.3). To get a definite balance, one needs to refine this rule:

$$\begin{aligned} b_0 &:= P(f_0, s), Y(s_p) \leftrightarrow P(f_0, s^1), Y(s_p^1) \\ b_1 &:= P(f_1, s), Y(s_p) \leftrightarrow P(f_1, s^1), Y(s_p^1) \end{aligned}$$

Now each rule b_i specifies enough of the context in which it applies to have a definite energy balance ϵ_i^{PY} . Following the same intuition of revealing (just) enough context, we obtain a balanced rule set for conformational changes:

$$\begin{aligned} f_{ij} &:= P(f_i, y^1), P(x^1, f_0, y^2, s), P(x^2, f_j) \leftrightarrow P(f_i, y^1), P(x^1, f_1, y^2, s), P(x^2, f_j) \\ f'_{ij} &:= P(f_i, y^1), P(x^1, f_0, y^2, s^-), P(x^2, f_j) \leftrightarrow P(f_i, y^1), P(x^1, f_1, y^2, s^-), P(x^2, f_j) \end{aligned}$$

The first (second) group of rules represents the changes in the absence (presence) of a Y bound to the middle P undergoing a change of conformation. (The fact that P 's site s is bound is indicated by the underscore exponent.)

These f -rules have respective balance:

$$\begin{aligned} \epsilon_{i1}^{PP} + \epsilon_{1j}^{PP} - \epsilon_{i0}^{PP} - \epsilon_{0j}^{PP} + \epsilon_1^P - \epsilon_0^P \\ \epsilon_{i1}^{PP} + \epsilon_{1j}^{PP} - \epsilon_{i0}^{PP} - \epsilon_{0j}^{PP} + \epsilon_1^P - \epsilon_0^P + \epsilon_1^{PY} - \epsilon_0^{PY} \end{aligned}$$

As we have ten reversible rules, and only eight energy patterns, there must be linear dependencies between the various balances. Indeed, in this case, it is easy to see that the family of vector balances has rank six. Thermodynamic consistency induces relationships between rates; a well-established fact in the case of reaction networks (*e.g.* see Ref. [7]).

With the rules in place, the final step is to choose rates which satisfy detailed balance. This guarantees that the obtained rule set converges to the equilibrium specified by the choice of the energy cost vector. Convergence will happen whatever ϵ is, *i.e.* symbolically. If, in addition, ϵ follows (4.1–4.3), one can see in Fig. 2 that the ring (i) undergoes sharp transitions when active Y is stepped up and down again; and (ii) has at all times very few mismatches.

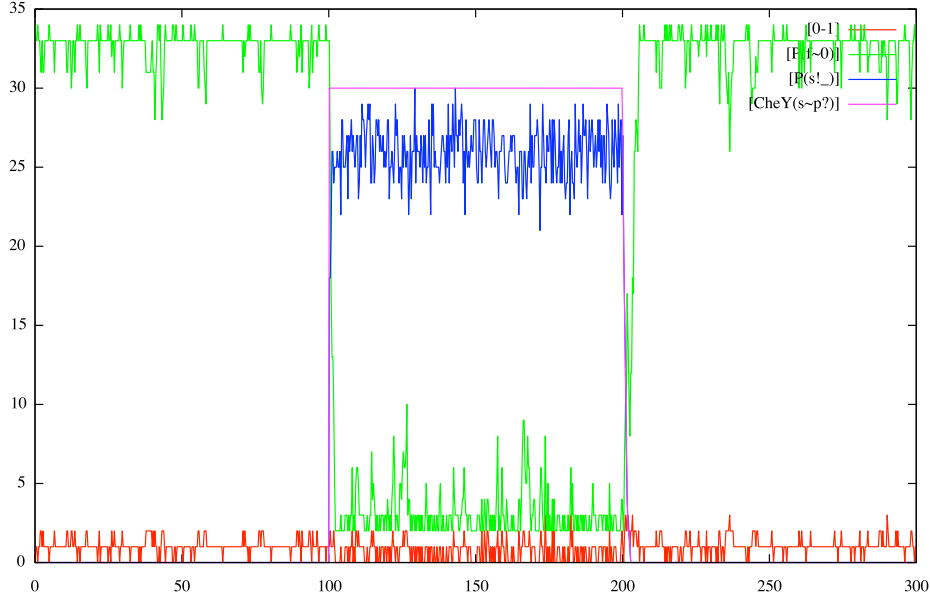


Figure 2: The simulation steps up the amount of active Y at $t = 100$, and down again at $t = 200$; this sends the entire ring into an homogeneously 1 conformation, and back to 0. The number of mismatches (lowest curve) stays low, even during transitions.

4.2. How to generate the rules. Our set of balanced rules was based on two generators, b for binding, f for flipping:

$$\begin{aligned} b &:= P(s), Y(s_p) \leftrightarrow P(s^1), Y(s_p^1) \\ f &:= P(f_0) \leftrightarrow P(f_1) \end{aligned}$$

Note that there is a design choice here. In effect, we are saying that we are not interested in forming/breaking the bonds between the P s in the ring. If we wanted to incorporate also the ring assembly in the model, we would have to add $P(x), P(y) \leftrightarrow P(x^1), P(y^1)$ among our generator set \mathcal{G} . This would generate many more refined rules, as we will see. Recall that our patterns fall in three subgroups: $P(f_i, x^1), P(y^1, f_j)$; $P(f_i)$; and $P(f_i, s^1), Y(s^1)$.

Consider the extensions of b : clearly only the last pattern can glue relevantly on it; the corresponding (unique) site request is for P to reveal f and its internal state. This gives the first two rules b_0, b_1 .

Consider now the more interesting extensions of f : the second pattern type glues relevantly but does not generate any site request; the third one asks P to reveal its site s , resulting in two possible extensions (s^- means that s is bound):

$$\begin{aligned} P(f_0, s) &\leftrightarrow P(f_1, s) \\ P(f_0, s^-) &\leftrightarrow P(f_1, s^-) \end{aligned}$$

These extensions are *not* mature yet, as one can glue relevantly patterns of the first type on both sides of P , inducing a further request for revealing P 's sites x and y .

If we are in the component of an initial state where P s are arranged in a ring, then we know that the neighbours on both sides exist and are P s; this gives the final refinement of the above into the rules f_{ij}, f'_{ij} described earlier. If, on the other hand, we do not know that,

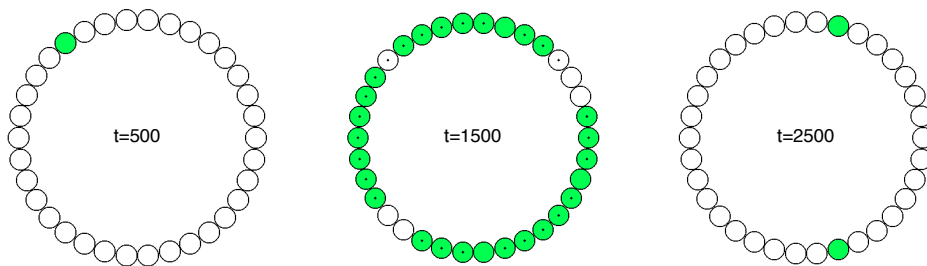


Figure 3: Snapshots of the ring configuration are taken at time 500, 1500, and 2500. Solid (green) circles indicate conformation 1, hollow ones conformation 0; a dot in the centre indicates a bound (hence active) Y . At times 500, 2500, no Y is bound (because they are all inactive) and the ring is globally in state 0, up to tiny fluctuations; at time 1500, it is globally in state 1 as a consequence of the binding of Y s.

we also have to add several rules where one or both of x , y are free, corresponding to open P -chains. This demonstrates the sensitivity of the obtained rule set to the initial choice of generators.

Hence, the rules we generated by hand are indeed the ones we would generate using our general refinement strategy of §3.

We can visualize the obtained simulations by extracting snapshots before, after and during the injection of active Y s, as in Fig. 3. Again we see few mismatches in both regime because of the Ising interaction expressed by the ϵ^{PP} energy costs. The choice of rates made in Ref. [2] for the f -generator is the symmetric version of (3.5).

5. CONCLUSION

We have presented a new ‘energy-oriented’ methodology for the development of site graph rewriting models based on a set \mathcal{P} of energy patterns; these patterns use a graphical syntax which allows us to specify the energy landscape. Rewrite rules are implicitly defined by \mathcal{P} and generator rules \mathcal{G} . The resulting rule set $\mathcal{G}_{\mathcal{P}}$ is guaranteed to be thermodynamically correct and to converge to the probability distribution described by the energy landscape, given suitable rates. The construction is entirely parametric in the energy costs ϵ , and modular in \mathcal{G} . This means that in a modelling context, one can sweep over various values for ϵ without having to rebuild the model, and compositionally add new rule components to \mathcal{G} . Both features are clearly useful. We expect our construction to provide a broad and uniform language to describe and analyse models of interacting biomolecules and similar systems of a quantitative fine-grained and distributed nature.

There are no specific conditions bearing on this construction other than that energy patterns should be local. It would be interesting to investigate whether suitable constraints on patterns and generator rules can obtain optimized generated rule sets. Another interesting extension would be to deal with non-local forms of energies expressing long-range interactions, where the metric is read off the graph itself. In practice, there will be many more rules generated, and beyond the descriptive aspects, simulations will need new ideas to be feasible.

A ray of hope comes from the log-affine kinetic model presented in §3, as rules can be partitioned by energy balances for faster selection.

Finally, as said in the introduction, there is a growing body of literature which turns a theoretical eye to site graph rewriting [17, 13, 18, 6], and it is tempting to ask whether our derivation can be replayed in more abstract settings; in particular, it would be very interesting to investigate its integration with the abstract framework for rule-based modelling developed in [23].

Acknowledgements. We would like to thank Peter Swain, Andrea Weisse, Julien Ollivier, Nicolas Oury, Finlo Boyde and Eric Deeds for many useful conversations concerning the subject of this paper. VD acknowledges the support of the ERC 320823 RULE project.

This work was sponsored by the Defense Advanced Research Projects Agency (DARPA) and the U. S. Army Research Office under grant number W911NF-14-1-0367. The views, opinions, and/or findings contained in this report are those of the authors and should not be interpreted as representing the official views or policies, either expressed or implied, of the Defense Advanced Research Projects Agency or the Department of Defense.

REFERENCES

- [1] John A Bachman and Peter Sorger. New approaches to modeling complex biochemistry. *Nature methods*, 8(2):130, 2011.
- [2] F. Bai, R.W. Branch, D.V. Nicolau Jr, T. Pilizota, B.C. Steel, P.K. Maini, and R.M. Berry. Conformational spread as a mechanism for cooperativity in the bacterial flagellar switch. *Science*, 327(5966):685–689, 2010.
- [3] Olivier Bournez, Guy-Marie Côme, Valérie Conraud, Hélène Kirchner, and Liliana Ibanescu. A rule-based approach for automated generation of kinetic chemical mechanisms. In Robert Nieuwenhuis, editor, *RTA*, volume 2706 of *Lecture Notes in Computer Science*, pages 30–45. Springer, 2003.
- [4] Pierre Boutillier, Jérôme Feret, Jean Krivine, and Lý Kim Quyên. *KaSim and KaSa Reference Manual*, 2014.
- [5] V. Danos, J. Feret, W. Fontana, and J. Krivine. Scalable simulation of cellular signaling networks. *Asian Symposium on Programming Languages and Systems*, pages 139–157, 2007.
- [6] V. Danos, R. Harmer, and G. Winskel. Constraining rule-based dynamics with types. *Mathematical Structures in Computer Science*, 23(2):272–289, 2013.
- [7] V. Danos and N. Oury. Equilibrium and termination II: the case of Petri Nets. *Mathematical Structures in Computer Science*, 23(2):290–307, 2013.
- [8] Vincent Danos. Agile modelling of cellular signalling. *SOS’08 Invited paper, Electronic Notes in Theoretical Computer Science*, 229(4):3–10, 2009.
- [9] Vincent Danos, Jérôme Feret, Walter Fontana, Russell Harmer, Jonathan Hayman, Jean Krivine, Christopher D. Thompson-Walsh, and Glynn Winskel. Graphs, rewriting and pathway reconstruction for rule-based models. In Deepak D’Souza, Telikepalli Kavitha, and Jaikumar Radhakrishnan, editors, *Foundations of Software Technology and Theoretical Computer Science, FSTTCS 2012*, pages 276–288, 2012.
- [10] Vincent Danos, Jérôme Feret, Walter Fontana, and Jean Krivine. Scalable simulation of cellular signaling networks. In Z. Shao, editor, *Proceedings of APLAS 2007*, volume 4807 of *LNCS*, pages 139–157, 2007.
- [11] Vincent Danos and Nicolas Oury. Equilibrium and termination. In S. Barry Cooper, Prakash Panangaden, and Elham Kashefi, editors, *Proceedings Sixth Workshop on Developments in Computational Models: Causality, Computation, and Physics*, volume 26 of *EPTCS*, pages 75–84, 2010.
- [12] Y. Diers. Familles universelles de morphismes. *Tech. report, Université des Sciences et Techniques de Lille I*, 1978.
- [13] L. Dixon and A. Kissinger. Open graphs and monoidal theories. *arXiv:1011.4114*, 2010.
- [14] Hartmut Ehrig. *Handbook of graph grammars and computing by graph transformation: Applications, Languages and Tools*, volume 2. World Scientific Publishing Company, 1999.

- [15] J.R. Faeder, M.L. Blinov, and W.S. Hlavacek. Rule-based modeling of biochemical systems with BioNetGen. *Methods Mol. Biol.*, 500:113–167, 2009.
- [16] Thilo Gross and Hiroki Sayama. *Adaptive networks*. Springer, 2009.
- [17] Jonathan Hayman and Tobias Heindel. Pattern graphs and rule-based models: The semantics of kappa. In Frank Pfenning, editor, *FoSSaCS*, volume 7794 of *Lecture Notes in Computer Science*, pages 1–16. Springer, 2013.
- [18] Reiko Heckel. DPO transformation with open maps. In Hartmut Ehrig, Gregor Engels, Hans-Jörg Kreowski, and Grzegorz Rozenberg, editors, *ICGT*, volume 7562 of *Lecture Notes in Computer Science*, pages 203–217. Springer, 2012.
- [19] W.S. Hlavacek, J.R. Faeder, M.L. Blinov, R.G. Posner, M. Hucka, and W. Fontana. Rules for modeling signal-transduction systems. *Science Signalling*, 2006(344), 2006.
- [20] Jean Krivine, Robin Milner, and Angelo Troina. Stochastic bigraphs. *Electronic Notes in Theoretical Computer Science*, 218:73–96, 2008.
- [21] Stephen Lack and Pawel Sobocinski. Adhesive categories. In Igor Walukiewicz, editor, *FoSSaCS*, volume 2987 of *Lecture Notes in Computer Science*, pages 273–288. Springer, 2004.
- [22] Carlos F Lopez, Jeremy L Muhlich, John A Bachman, and Peter K Sorger. Programming biological models in python using pysb. *Molecular systems biology*, 9(1), 2013.
- [23] J. Lynch. A logical characterization of individual-based models. *Proceedings of Logic in Computer Science*, pages 203–217, 2008.
- [24] N. Metropolis, A.W. Rosenbluth, M.N. Rosenbluth, A.H. Teller, E. Teller, et al. Equation of state calculations by fast computing machines. *The Journal of Chemical Physics*, 21(6):1087, 1953.
- [25] E. Murphy, V. Danos, J. Feret, R. Harmer, and J. Krivine. Rule-based modelling and model refinement. In H. Lohdi and S. Muggleton, editors, *Elements of Computational Systems Biology*. Wiley, 2010.
- [26] Carl-Fredrik Tiger, Falko Krause, Gunnar Cedersund, Robert Palmér, Edda Klipp, Stefan Hohmann, Hiroaki Kitano, and Marcus Krantz. A framework for mapping, visualisation and automatic model creation of signal-transduction networks. *Molecular systems biology*, 8(1), 2012.

APPENDIX A. TRIANGLES MODEL

A working model definition for our first example (introduced in §1.1) is spelled out in full here. The syntax used is for KaSim version 4. As in the example of §4, we use the symmetric linear kinetic model of §3.7.

A.1. Agents, Parameters and Initial State. We start by declaring agents together with their sites. Then we declare the energy costs for $T = \cdot 123 \cdot$ ('t'), $?12?$ ('ab'), $?23?$ ('bc'), $?31?$ ('ca') as parameters. Subsequently the initial state is defined as containing 1000 agents of each type.

```
# Agent signatures
%agent: A(l,r)
%agent: B(l,r)
%agent: C(l,r)

# Energy costs
%var: 't' -10
%var: 'ab' 0
%var: 'bc' 0
%var: 'ca' 0

# Initial state
%init: 1000 (A(), B(), C())
```

A.2. Observables. Next we declare the graphs we would like to keep track of, *i.e.* those graphs we would like to generate trajectories for plotting. Here we are interested in the fraction of agents that are used to assemble triangles.

```
%obs: 'T' |A(l!1, r!2), B(l!2, r!3), C(l!3, r!1)|
```

A.3. Rules. Finally, we write our set of rules. These have been grouped according to the qualitative generator rule they refine. Note that these rules have been manually compressed as explained in §3.5.

```
# A(r), B(l) -> A(r!1), B(l!1) refines into:
A(l,r), B(l,r) -> A(l,r!1), B(l!1,r) @ [exp] (-1/2 * 'ab')
A(l!r.C,r), B(l,r) -> A(l!r.C,r!1), B(l!1,r) @ [exp] (-1/2 * 'ab')
A(l,r), B(l,r!1.C) -> A(l,r!1), B(l!1,r!1.C) @ [exp] (-1/2 * 'ab')
A(l!1,r ), B(l ,r!3), C(l!3,r!1) -> \
A(l!1,r!2), B(l!2,r!3), C(l!3,r!1) @ [exp] (-1/2 * ('ab' + 't'))
C(r!1), A(l!1,r ), B(l ,r!3), C(l!3) -> \
C(r!1), A(l!1,r!2), B(l!2,r!3), C(l!3) @ [exp] (-1/2 * 'ab')

# A(r!1), B(l!1) -> A(r), B(l) refines into:
A(l,r!1), B(l!1,r) -> A(l,r), B(l,r) @ [exp] -(-1/2 * 'ab')
A(l!r.C,r!1), B(l!1,r) -> A(l!r.C,r), B(l,r) @ [exp] -(-1/2 * 'ab')
```



```

A(l,r!1), B(l!1,r!1.C) -> A(l,r), B(l,r!1.C) @ [exp] -(-1/2 * 'ab')
A(l!1,r!2), B(l!2,r!3), C(l!3,r!1) -> \
A(l!1,r ), B(l ,r!3), C(l!3,r!1) @ [exp] -(-1/2 * ('ab' + 't'))
C(r!1), A(l!1,r!2), B(l!2,r!3), C(l!3) -> \
C(r!1), A(l!1,r ), B(l ,r!3), C(l!3) @ [exp] -(-1/2 * 'ab')

# B(r), C(l) -> B(r!1), C(l!1) refines into:
B(l,r), C(l,r) -> B(l,r!1), C(l!1,r) @ [exp] (-1/2 * 'bc')
B(l!r.A,r), C(l,r) -> B(l!r.A,r!1), C(l!1,r) @ [exp] (-1/2 * 'bc')
B(l,r), C(l,r!1.A) -> B(l,r!1), C(l!1,r!1.A) @ [exp] (-1/2 * 'bc')
B(l!1,r ), C(l ,r!3), A(l!3,r!1) -> \
B(l!1,r!2), C(l!2,r!3), A(l!3,r!1) @ [exp] (-1/2 * ('bc' + 't'))
A(r!1), B(l!1,r ), C(l ,r!3), A(l!3) -> \
A(r!1), B(l!1,r!2), C(l!2,r!3), A(l!3) @ [exp] (-1/2 * 'bc')

# B(r!1), C(l!1) -> B(r), C(l) refines into:
B(l,r!1), C(l!1,r) -> B(l,r), C(l,r) @ [exp] -(-1/2 * 'bc')
B(l!r.A,r!1), C(l!1,r) -> B(l!r.A,r), C(l,r) @ [exp] -(-1/2 * 'bc')
B(l,r!1), C(l!1,r!1.A) -> B(l,r), C(l,r!1.A) @ [exp] -(-1/2 * 'bc')
B(l!1,r!2), C(l!2,r!3), A(l!3,r!1) -> \
B(l!1,r ), C(l ,r!3), A(l!3,r!1) @ [exp] -(-1/2 * ('bc' + 't'))
A(r!1), B(l!1,r!2), C(l!2,r!3), A(l!3) -> \
A(r!1), B(l!1,r ), C(l ,r!3), A(l!3) @ [exp] -(-1/2 * 'bc')

# C(r), A(l) -> C(r!1), A(l!1) refines into:
C(l,r), A(l,r) -> C(l,r!1), A(l!1,r) @ [exp] (-1/2 * 'ca')
C(l!r.B,r), A(l,r) -> C(l!r.B,r!1), A(l!1,r) @ [exp] (-1/2 * 'ca')
C(l,r), A(l,r!1.B) -> C(l,r!1), A(l!1,r!1.B) @ [exp] (-1/2 * 'ca')
C(l!1,r ), A(l ,r!3), B(l!3,r!1) -> \
C(l!1,r!2), A(l!2,r!3), B(l!3,r!1) @ [exp] (-1/2 * ('ca' + 't'))
B(r!1), C(l!1,r ), A(l ,r!3), B(l!3) -> \
B(r!1), C(l!1,r!2), A(l!2,r!3), B(l!3) @ [exp] (-1/2 * 'ca')

# C(r!1), A(l!1) -> C(r), A(l) refines into:
C(l,r!1), A(l!1,r) -> C(l,r), A(l,r) @ [exp] -(-1/2 * 'ca')
C(l!r.B,r!1), A(l!1,r) -> C(l!r.B,r), A(l,r) @ [exp] -(-1/2 * 'ca')
C(l,r!1), A(l!1,r!1.B) -> C(l,r), A(l,r!1.B) @ [exp] -(-1/2 * 'ca')
C(l!1,r!2), A(l!2,r!3), B(l!3,r!1) -> \
C(l!1,r ), A(l ,r!3), B(l!3,r!1) @ [exp] -(-1/2 * ('ca' + 't'))
B(r!1), C(l!1,r!2), A(l!2,r!3), B(l!3) -> \
B(r!1), C(l!1,r ), A(l ,r!3), B(l!3) @ [exp] -(-1/2 * 'ca')

```

A.4. Results. We run this model using KaSim (version 4) to obtain trajectories for the number of triangles during the simulation. From these numbers we can estimate what is the expected number of triangles at equilibrium. In Fig. 4 the results of 4 different runs with different energy costs for a unit triangle are displayed. When $\epsilon(T) = -10$, almost all

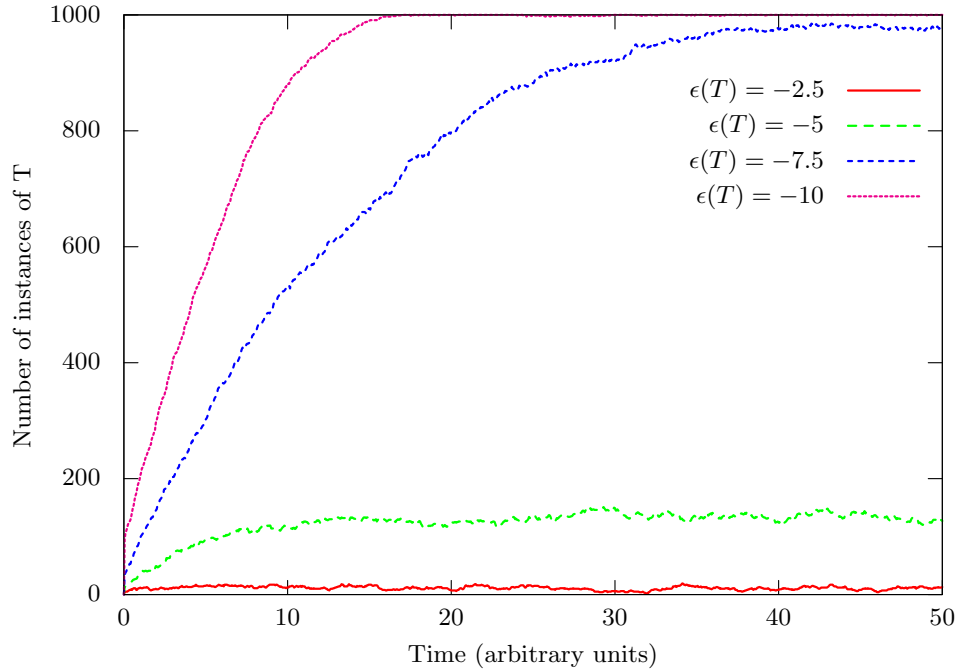


Figure 4: Trajectories for $T = .123$ when $\epsilon(T)$ varies (this value is set by changing the value of parameter ‘t’ in the KaSim file).

agents are used to build triangles. Instead, when $\epsilon(T) = -2.5$ only less than 5% is used. Interestingly, in both cases the set of states that minimize the energy function is the same, namely those states that maximize the amount of triangles. So then why is it that in the latter case there are so few triangles? The reason is entropic: although the probability of being in a state with few triangles is small, there are many such states and together they outweigh the probability of being in the few states were the energy is minimized. By further decreasing the energy of those few states we compensate for this mass effect, until at $\epsilon(T) = -10$, order wins, and the effect is not noticeable anymore.