

Replace this file with `prentcsmacro.sty` for your meeting,  
or with `entcsmacro.sty` for your meeting. Both can be  
found at the [ENTCS Macro Home Page](#).

# Containment in Rule-Based Models

C. D. Thompson-Walsh, J. Hayman<sup>1</sup>, G. Winskel<sup>1</sup>

*Computer Laboratory, University of Cambridge, United Kingdom*

---

## Abstract

Recently, there has been substantial interest in using rule-based modelling approaches, such as the Kappa modelling language, to attack the combinatorial intractability of many biochemical systems. These approaches have allowed several novel static analyses to be developed, which motivates broadening their expressivity. In this paper, we build upon prior work giving Kappa an SPO-rewriting semantics to add containment structure, to model the various ways in which biological mixtures are partitioned and enclosed by membranes.

*Keywords:* biochemical networks, Kappa, regions, rule-based modelling, SPO-rewriting

---

## 1 Introduction

There has recently been a great deal of interest in and investigation of the use of *rule-based* modelling frameworks for the simulation and static analysis of biochemical systems, in particular well-mixed chemical solutions such as the cell cytoplasm. Rule-based approaches are attractive because modern approaches to biochemistry emphasise how proteins interact via different ‘functional domains’. This lends itself naturally to a modelling proteins as collections of interacting functional domains or sites; rule-based approaches help to manage the severe combinatorial explosion in numbers of different species of protein complex possible in even fairly simple biological systems [9].

Two examples of the rule-based approach are Kappa and the BNG modelling systems [11,2]. There has been substantial recent work done on applying techniques based on Kappa to managing the combinatorial complexity of biological systems [7,10]. This motivates expanding the expressivity and

---

<sup>1</sup> JH and GW gratefully acknowledge the support of the ERC through the Advanced Grant ECSYM.

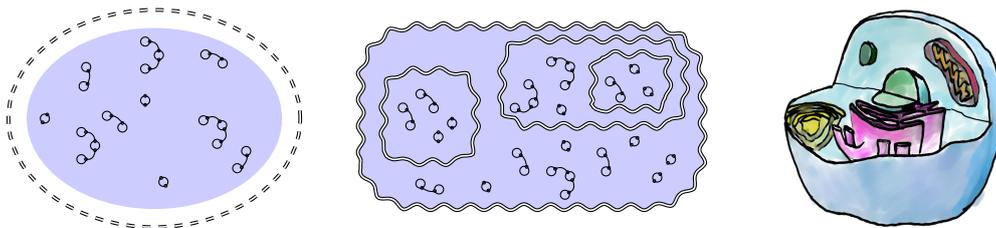


Fig. 1. A spectrum of models of a cell, at different levels of detail. At left, a representation of the approximation of a cell as a “cytoplasm in a bag”, with the outer membrane shown dashed, circles representing proteins and arcs showing complexification as in standard representations of Kappa; at right, an informal sketch of a eukaryotic cell; centre, the model we examine in this paper.

applicability of the rule-based approach, to bring it to more richly-structured systems.

One particular area in which Kappa is currently weak in modelling expressiveness is in the description of different “regions” or areas in which reactions can take place. Real cells, particularly eukaryotic cells, are complex and highly structured spaces. Far from being simple well-mixed cytoplasms, cells have a huge diversity of bilipid membranes and other structures *within* the outer cell membrane, all of them reducing the degree to which the contents of a cell can really be considered well-mixed at all. A typical “cut-away” depiction of a eukaryotic cell gives a sense of the complexity (see the rightmost part of Fig. 1)

This partition matters because different compartments can exhibit strongly divergent concentrations, and hence divergent chemical behaviours; and because frequently, the cell will exploit concentration gradients across membranes. *Where* a reaction happens turns out to matter to how the system evolves. Nor are the structures static: new compartments can form, compartments can be destroyed, and compartments can merge or bud (hive off from existing compartments).

Any simple formalisation of the complexity of actual eukaryotic cells necessarily approximates; the existing approximation in Kappa modelling is to describe biochemical networks as operating in a single well-mixed solution (*i.e.* one in which there is no structure statistically significant to the probabilities of particular particles interacting.)

Our contribution in this paper is to describe a feasible compromise between the full complexity of (for example) a cell and this existing approximation, as shown in figure 1. Our approach is based around adding containment structure to Kappa; it is related to other efforts (such as [13]), but differs in the emphasis placed on being able to model the dynamics of these structures.

The model we give will be based around the idea of *containment* of protein complexes and regions within other regions; hence we adopt the name containment structures. We will build this model as additional structure to be carried into a semantics for Kappa developed in previous work [6].

In this paper, we first describe a version of Kappa and its associated Single-Pushout (SPO) semantics, simplified to take account of the fact that issues in providing an expressive model for containment are largely orthogonal to those in providing an expressive model for Kappa. This is a simplification of work to appear in [6], which provides a specialised graph-based approach to describe Kappa rewriting, comparable to graph-based perspectives on other systems for modelling signalling pathways [1,3]. We then describe, in an analogous fashion, the statics and dynamics of our containment structures, and give examples of how various biological processes can be modelled. Finally, we describe how our containment structure dynamics can be dovetailed onto Kappa; the usual stochastics can then be applied to give a model with both stochastic chemical and structural expressiveness. We conclude by giving a brief sketch of present and future work based on this new model.

## 2 Simplified Kappa

There are a variety of papers on Kappa; a good recent introduction can be found in [8]. The essential approach in Kappa is to model the state of a well-mixed solution in terms of *complexes* of *agents*. The intention is that agents model individual proteins or molecules; each agent has a collection of functional *sites* associated with it, which may enter into binary *links* with other sites. The dynamics are given in terms of rules, which may be thought of as *locally* describing how the solution is affected by describing its effect on any collection of agents matching a given test pattern.

*Complexes* are connected components of these agents in the graph defined by link structure. Complexes model aggregations of molecules: they can model aggregates of proteins that are chemically bonded, or else associated by any one of a number of “weak” bonding processes.

The overall pattern of development for our semantics is first to establish a category of structures representing Kappa states and rule patterns, which has as morphisms structure-preserving maps. This is used to introduce a notion of partial map (using a standard span construction) and is followed by distinguishing classes of *matchings* and *action* morphisms. We can then model rules using action morphisms, and rule applications as matchings from the left-hand side of the action morphism into an object representing the mixture. We define the result of applying a rule to be the pushout of the action morphism against this matching. Finally, we prove that this pushout always exists. (When we come to define our compartment-modelling structures, we will follow a similar pattern.)

We begin by defining  $\Sigma$ -graphs, a kind of generalised Kappa state:

**Definition 2.1** *Assume that we have the ambient sets  $Type$ , a set of agent types, and  $SiteId$  a set of site identifiers; and a map  $\Sigma : Type \rightarrow \mathcal{P}(SiteId)$*

which gives the set of sites on an agent of given type.

Then, a  $\Sigma$ -graph is a tuple  $(Ag, type, Sts, L)$  where

- $Ag$  is the set of agents.
- $type : Ag \rightarrow Type$  is a total typing function, giving what “kind” of molecule or protein each agent models.
- $Sts \subseteq \{(A, i) \mid A \in Ag \text{ and } i \in \Sigma \circ type(A)\}$  is the set of sites specified in the  $\Sigma$  graph.
- $L \subseteq Sts \times Sts$  is the set of links, a symmetric irreflexive relation on  $Sts$ .

Note the ability to leave a site’s link state unspecified. Later we will see that an object with unspecified link state at a site may be match structures which have that site specified as either linked or unlinked.

We identify a class of “site graph” objects which represent the physically meaningful rule patterns:

**Definition 2.2** A *Kappa site graph* is a  $\Sigma$ -graph  $(Ag, type, Sts, L)$  where the link relation satisfies the partial function condition:  $(s, s') \in L$  and  $(s, s'') \in L \Rightarrow s' = s''$ .

In a site graph, each site participates in at most one link.

We can also pick out those objects which represent complete system states, without any underspecification — we call these *mixtures*, in reference to the biochemical setting of Kappa:

**Definition 2.3** A *mixture* is a *Kappa site graph*  $(Ag, type, Sts, L)$  in which  $Sts = \{(A, i) \mid A \in Ag \text{ and } i \in \Sigma \circ type(A)\}$ .

In a mixture, every agent has to have every site specified allowed by the signature.

For any  $\Sigma$ -graph  $X$ , it is convenient to adopt the convention of writing  $Ag_X$  for its set of agents,  $Sts_X$  for its set of sites,  $L_X$  for its link relation and  $type_X$  for its typing function.

**Definition 2.4** A  $\Sigma$ -homomorphism  $f : X \rightarrow Y$  is a (total) function  $f : Ag_X \rightarrow Ag_Y$  such that:

- $f$  preserves agent types:  $type_X(C) = type_Y(f(C))$ .
- $f$  preserves the presence of sites:  $\{(f(A), i) \mid (A, i) \in Sts_X\} \subseteq Sts_Y$ .
- Link structure is respected: if  $((C, i), (D, j)) \in L_X$  then  $((f(C), i), (f(D), j)) \in L_Y$ .

Note that homomorphisms only preserve the *presence* of links from a site; they do not preserve the *absence* of links from a given site. We shall see that this generality is important when we come to consider action morphisms.

Partial morphisms on  $\Sigma$ -graphs are now constructed as particular kinds

of *spans* of homomorphisms. Essentially, this amounts to describing a partial morphism in two stages, firstly defining the domain of definition of the partial map and then defining the effect of the partial morphism as a homomorphism from the domain of definition into the target. The domain of definition is just a subset of elements in the domain:

**Definition 2.5** *A  $\Sigma$ -sub-graph  $D$  of a  $\Sigma$ -graph  $S$  is a  $\Sigma$ -graph where  $Ag_D \subseteq Ag_S$ ,  $L_D \subseteq L_S$ ,  $Sts_D \subseteq Sts_S$  and  $type_D$  is the restriction of  $type_S$  to  $Ag_D$ . A  $\Sigma$ -sub-graph inclusion is the obvious identity-like morphism from  $D$  to  $S$ .*

We can now define a partial morphism from  $S$  to  $T$  to be a span consisting of an inclusion  $i : D \hookrightarrow S$ , where  $D$  is a  $\Sigma$ -subgraph of  $S$ , together with a homomorphism  $h : D \rightarrow T$ . Since the inclusion  $i$  is uniquely determined by the choice of  $D$ , we write  $(D, h)$  for such spans.

For any partial morphism  $f : S \rightarrow T$ , we adopt the convention of writing  $D_f$  for the subgraph  $D$  above and  $f_0$  for the homomorphism  $h$  above, so  $f = (D_f, f_0)$ . For any agent  $A$  in  $D_f$ , we allow ourselves to write  $f(A)$  for the result of applying the homomorphism  $f_0$  to  $A$ .

Suppose we have partial morphisms  $f = (D_f, f_0) : S \rightarrow T$  and  $g = (D_g, g_0) : T \rightarrow U$ . Their composition is defined in the obvious way, with the domain of definition of the composition formed as the ‘inverse image’ of  $f_0$  applied to the domain of definition of  $g$ . In full detail:

- For any agent  $A \in Ag_S$ , we have  $A \in Ag_{D_{g \circ f}}$  iff  $A \in Ag_{D_f}$  and  $f(A) \in Ag_{D_g}$ .
- For any site  $(A, i) \in Sts_S$ , we have  $(A, i) \in Sts_{D_{g \circ f}}$  iff  $(A, i) \in Sts_{D_f}$  and  $(f(A), i) \in Sts_{D_g}$ .
- For any link  $((A, i), (A', i')) \in L_S$ , we have  $((A, i), (A', i')) \in L_{D_{g \circ f}}$  iff  $((A, i), (A', i')) \in L_{D_f}$  and  $((f(A), i), (f(A'), i')) \in L_{D_g}$ .

It can be seen [6] that this construction yields a pullback of the homomorphism  $f_0 : D_f \rightarrow T$  against the inclusion  $D_g \hookrightarrow T$  in the category of  $\Sigma$ -graphs with homomorphisms between them, so it corresponds to the standard definition of composition of spans by taking a pullback.

With this composition between partial homomorphisms, we can form a category of  $\Sigma$ -graphs with partial homomorphisms. Clearly, any homomorphism can be regarded as a partial morphism by taking the inclusion to be the identity.

With all the categorical underpinnings in place, we can now describe the two distinguished classes of partial morphisms, matchings and actions, which are used to model matches and rules.

For rewriting rules, the idea is to model an action as a partial morphism  $\alpha : X \rightarrow Y$  between site graphs  $X$  and  $Y$ . Site graphs play the rôle of patterns: an action can be applied to a mixture if its left-hand side  $X$  matches some part of the mixture. Matchings, which we define next, model these matches;

they are injective homomorphisms that additionally preserve the absence of links on determined sites. They can be thought of as picking out an exact image of the left-hand side of a rule in a state.

**Definition 2.6** An matching from a site graph  $X$  into a site graph  $Y$  is a homomorphism  $f : X \rightarrow Y$  which satisfies the following axioms:

- $f$  is injective.
- $f$  reflects link information: For any  $(C, i) \in Sts_X$ , if there exists some  $(D, j) \in Sts_Y$  such that  $((f(C), i), (D, j)) \in L_Y$ , then there must exist some  $C'$  such that  $((C, i), (C', j)) \in L_X$ . (The agent  $C'$  is necessarily the unique agent such that  $f(C') = D$ ).

We now define action morphisms. Viewing an action  $\alpha$  as a span, the domain of definition  $D_\alpha$  represents the part of  $X$  that is tested by the rule. The elements of  $X$  not in  $D_\alpha$  are to be destroyed by the rule, and the elements of  $Y$  outside the image of  $\alpha_0$  are created by the rule.

**Definition 2.7** An *action* is a partial morphism  $\alpha : X \rightarrow Y$  from a site graph  $X$  to a site graph  $Y$  that obeys the following axioms:

- $\alpha$  is partial injective:  
for any  $A, A' \in Ag_{D_\alpha}$ , if  $\alpha(A) = \alpha(A')$  then  $A = A'$
- the only way to delete a site is to delete its agent:  
if  $(A, i) \in Sts_X$  and  $A \in Ag_{D_\alpha}$  then  $(A, i) \in Sts_{D_\alpha}$ .
- no sites are introduced to existing agents:  
if  $A \in Ag_{D_\alpha}$  and  $(\alpha(A), i) \in Sts_Y$  then  $(A, i) \in Sts_{D_\alpha}$ .
- all newly-created agents are created with all possible sites:  
if  $A \in Ag_Y$  and there is no  $A_0 \in Ag_{D_\alpha}$  such that  $\alpha(A_0) = A$  then  $(A, i) \in Sts_Y$  for all  $i \in \Sigma(type_Y(A))$ .

We can see how rules apply to states by taking the pushout of the action against the matching, in the category of  $\Sigma$ -graphs with partial morphisms:

**Proposition 2.8** *Given any span  $\alpha : X \rightarrow Y, i : X \rightarrow S$  in the category of partial  $\Sigma$ -morphisms, with  $\alpha$  an action and  $i$  an matching, and  $S$  a mixture, there is a pushout*

$$\begin{array}{ccc} X & \xrightarrow{\alpha} & Y \\ i \downarrow & \lrcorner & \downarrow i' \\ S & \xrightarrow{\alpha'} & T \end{array}$$

where  $\alpha'$  is an action and  $i'$  is an matching, and  $T$  is also a mixture.

It is shown in [6] how this corresponds to the usual interpretation of Kappa actions as describing how agents and links are added to and removed from some

part of the mixture initially matching  $S$ .

### 3 Regions

Real biological systems exhibit remarkable complexity in structure and arrangement. A common, simplifying pattern is wanted to be the basis of our model. In the simplest, and most common case, one can think of a membrane that encloses a well-mixed region of space, and whose precise shape is unimportant.

We therefore adopt a perspective that views the cell (or subsystem of interest) as a set of regions, each of which may recursively contain other regions, and which assumes the contents of each region to be well-mixed, *i.e.* to obey standard mass-action dynamics. Each region models one compartment; we model one compartment held inside another through a notion of *containment* on regions.

In this section, we study the definition of systems composed exclusively of regions. In subsequent sections, we will extend our Kappa semantics from above to carry this region structure, resulting in a recursive scheme of regions containing well-mixed solutions of agents and other regions. The development of this section parallels that of Section 2.

#### 3.1 Containment Structures

We represent region containment in essence as a finite-depth forest. We have a set of regions  $R$  and a reflexive relation  $\preceq^1$  over them; we read  $x \preceq^1 y$  as “the region  $x$  is immediately contained in or equal to the region  $y$ ”.

**Definition 3.1** *A containment structure  $X$  is a pair of a set of regions  $R_X$  and a containment relation  $\preceq^1_X$ , where  $\preceq^1_X \subseteq R_X \times R_X$ , such that the  $\preceq^1_X$  relation satisfies the following axioms:*

- $\preceq^1_X$  has finite depth: writing  $\preceq_X$  for the transitive closure of  $\preceq^1_X$ , we insist that for any  $x$ ,  $\{y \mid x \preceq_X y\}$  is finite.
- For any  $x, y, z$  with  $z \neq x \neq y$ , if  $x \preceq^1_X y$  and  $x \preceq^1_X z$  then  $y = z$ .
- For all  $x$ , we have  $x \preceq^1_X x$ .

*If the order  $\preceq_X$  has a maximum element, we call the containment structure  $X$  connected; we call the maximum element the top region.*

A region models a region of space. We think of the cytoplasm (or other solution) as being partitioned into regions by membranes, so each region implies the existence of an enclosing membrane. Containment structures are easily represented as a forest, as in Fig. 2.

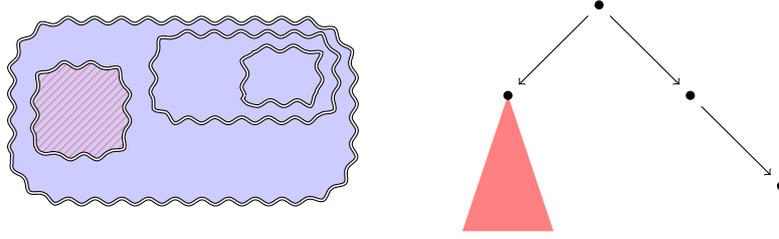


Fig. 2. Representing containment structures as trees. The informally depicted situation on the left is represented as the tree shown at right (region identities are omitted for clarity.) It will be convenient to be able to use filled triangles to represent subtrees whose precise members we have not specified; in this case, the triangle represents all of the contents of the shaded membrane.

### 3.2 Maps and Dynamics

We now describe containment structure homomorphisms. These may be thought of as describing how one structure can be mapped over into another.

**Definition 3.2** *A containment structure homomorphism  $f : X \rightarrow Y$  is a total function  $f : R_X \rightarrow R_Y$  such that  $f$  preserves  $\preceq^1_X$  — formally, for all  $x, y$  we have that  $x \preceq^1_X y$  implies  $f(x) \preceq^1_Y f(y)$ .*

Containment structures with these homomorphisms form the category **Reg**. The next step is to introduce partiality, as we did before, as a span consisting of an inclusion and a homomorphism.

**Definition 3.3** *A containment structure inclusion is a morphism  $f : X \rightarrow Y$  where  $f$  is total and identity-like (so  $R_X \subseteq R_Y$ ), and  $\preceq^1_X$  is contained in  $\preceq^1_Y$*

**Definition 3.4** *A partial containment structure morphism  $f : X \rightarrow Y$  is a span of an inclusion with a homomorphism:  $D_f \hookrightarrow X$  and  $f_0 : D_f \rightarrow Y$ .  $D_f$  is the “domain of definition”; it identifies that part of  $X$  on which  $f$  is defined.*

It can be shown that these spans compose associatively in the way expected, and that there is an identity; we therefore have the category of containment structures with partial morphisms, called **PReg**. Moreover, homomorphisms are included into the category by simply using the identity inclusion to form the inclusion leg of the span; we thus have a class of “total” morphisms in **PReg**.

We can now establish our class of action morphisms, used to model rules and updates, and our class of matchings, used to model rule applications.

We utilise a constrained form of partiality for our actions, where maps must be total on all regions. Maps need not, however, preserve the *containments* between regions. This choice of partiality, taken together with the rest of the defined dynamics, ensures that the effect of a rule is confined to the region of the containment structure where it applies.

**Definition 3.5** *A containment structure matching is a total morphism  $i : X \rightarrow S$  such that  $i$  is injective.*

**Definition 3.6** A *containment structure action morphism* is a morphism  $\alpha : X \rightarrow Y$ , in **PReg**, such that both  $X$  and  $Y$  are connected,  $\alpha$  takes the top region of  $X$  to the top region of  $Y$  and  $\alpha$  is total on regions.

This is all that is needed for the following result:

**Proposition 3.7** Given any span in **PReg**  $\alpha : X \rightarrow Y, i : X \rightarrow S$ , with  $\alpha$  an action and  $i$  an matching, and  $S$  a connected containment structure, there is a pushout

$$\begin{array}{ccc} X & \xrightarrow{\alpha} & Y \\ i \downarrow & \lrcorner & \downarrow i' \\ S & \xrightarrow{\alpha'} & T \end{array}$$

where  $\alpha'$  is an action and  $i'$  is an matching, and  $T$  is also connected.

### 3.3 Expressiveness

This dynamics enable us to model several biologically important processes, some of which are briefly sketched here and in Figure 3. We also indicate certain processes that are not describable in these semantics.

**Merging** In merging, the object is to combine two membranes into a new, larger membrane with the union of the contents of the original membrane. We can merge membranes in the natural way, by constructing a morphism which maps the two to the same target.

Suppose we have two sibling membranes,  $b$  and  $c$ , which we wish to merge. Since they are siblings, they are both immediately contained in some region  $a$ . Our containment structure morphism will consist of a map from  $a, b, c$  to  $a, d$ , forming a morphism between the containment structures  $X$  and  $Y$ :

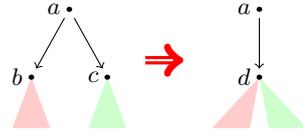
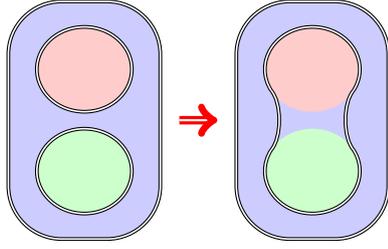
$$R_X = \{a, b, c\} \quad \alpha \triangleq \begin{cases} a \mapsto a & R_Y = \{a, d\} \\ b \mapsto d & \preceq^1_Y = \{(d, a), (a, a), \\ c \mapsto d & (d, d)\} \end{cases} \\ \preceq^1_X = \{(b, a), (c, a)\} \cup Id_{R_X}$$

(Since the map is total, the domain of definition  $D_\alpha$  is simply  $X$ .)

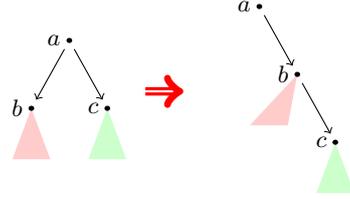
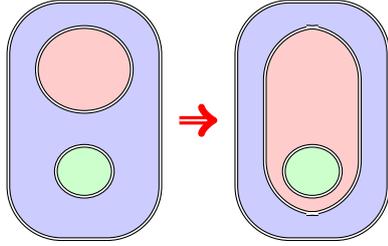
**Motion** The object of motion is to take one membrane and move it, along with its contents, into another. This is modelled with a morphism mapping membranes from their starting configuration into their ending configuration. Perhaps surprisingly, however, motion exploits *partiality*: we use undefinedness to “break” immediate containments which no longer hold in the target.

For example, suppose that we have membranes  $b$  and  $c$  inside a top-level membrane  $a$ . We wish to move the membrane  $c$  and its contents so that they are inside  $b$ . The immediate containment of  $c$  inside  $a$  will not be preserved, so it will not be present in the domain of definition. If  $X$  is the left-hand

## • Merging



## • Motion



## • Dissolution

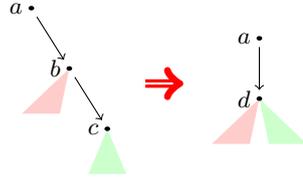
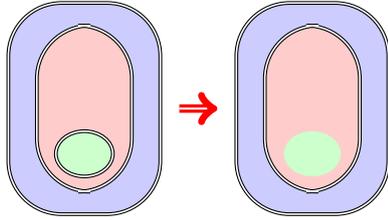


Fig. 3. Merging, motion and dissolution. In each case, we illustrate at left a sketch of the structure of the modelled membranes before and after, respectively, merging, motion and dissolution. At right, we show how this evolution is represented in a tree-like depiction of a containment structure. As above, the shaded triangles serve to indicate where the subtrees (which we do not draw in detail) of each region lie. For clarity, we neglect other descendants of the top region.

side of the rule and  $Y$  the right-hand side of the rule, we can model this example of motion as follows:

$$\begin{array}{l}
 R_X = \{a, b, c\} \\
 \preceq^1_X = \{(c, a), (b, a)\} \cup Id_{R_X} \\
 R_D = \{a, b, c\} \\
 \preceq^1_D = \{(b, a)\} \cup Id_{R_D}
 \end{array}
 \quad
 \alpha \triangleq
 \begin{cases}
 a \mapsto a & R_Y = \{a, b, c\} \\
 b \mapsto b & \preceq^1_Y = \{(b, a), (c, b)\} \\
 c \mapsto c & \cup Id_{R_Y}
 \end{cases}$$

**Dissolution of membranes** The object in dissolution is to model a rule which removes a membrane, placing all of its contents in the next enclosing membrane. This is possible using a non-injective mapping on membranes: given a membrane  $c$ , which we wish to delete, and an enclosing membrane  $b$ , itself living inside  $a$ , we construct a morphism which maps both  $b$  and  $c$  to a region  $d$ .

$$\begin{array}{l}
 R_X = \{a, b, c\} \\
 \preceq^1_X = \{(c, b), (b, a)\} \\
 \cup Id_{R_X}
 \end{array}
 \quad
 \alpha \triangleq
 \begin{cases}
 a \mapsto a \\
 b \mapsto d \\
 c \mapsto d
 \end{cases}
 \quad
 \begin{array}{l}
 R_Y = \{a, d\} \\
 \preceq^1_X = \{(d, a), (a, a), (d, d)\}
 \end{array}$$

**Creation of new empty regions** is straightforward, as these may be added freely to the codomain of an action.

The general flavour of our semantics is of a relatively rigid matching and rewriting of parts of the region structures. There are some situations which do not sit well in this model. Many natural processes require a kind of “universal rendezvous” — that is, the rule-application will require the systematic modification of an unbounded number of regions without direct pre-images in the rule LHS. In general, our quite rigid matchings do not support these. Two physically motivated examples not present in our semantics are:

**Destruction** It is sometimes desirable to model the total annihilation of a region, together with all of its contents. This is not possible in the present semantics: we in fact insist that regions are never destroyed. Even if we could destroy regions, we would need to match on “everything” inside a region, which is a form of universal rendezvous not possible with our rigid matchings.

**Budding** Many processes that form membranes work by “budding”: partitioning an existing region. In budding then, the original region therefore need to be exhaustively and stochastically split between the two new daughter regions. Although we can model “small” budding events using techniques such as those described in [14], we cannot express the general case: there is no way to get a “handle”, via matchings, on “all contents” of a region. In addition, there is no way of describing stochastic partitioning of these contents.

## 4 Augmenting Kappa with Containment

In this section, we give an account of how to equip our simplified Kappa semantics with containment structure.

**Definition 4.1** *We define a  $\Sigma$ -region-graph to be a triple of:*

- A  $\Sigma$ -graph  $G = (Ag, type, Sts, L)$ .
- An containment structure  $S = (R, \preceq^1)$ .
- A placing function  $p : Ag \rightarrow R$ .

The intuition of course is that this is a  $\Sigma$ -graph which carries some extra region structure; the placing relation indicates in which region each agent can be found.

We also define site graphs, mixtures and morphisms:

**Definition 4.2** We import the following definitions from containment structures and  $\Sigma$ -graphs:

- A  $\Sigma$ -region site graph is a  $\Sigma$ -region-graph  $(G, S, p)$  where  $G$  is a  $\Sigma$ -site graph,  $S$  is connected and  $p$  is total.
- A  $\Sigma$ -region mixture is a  $\Sigma$ -region site graph  $(G, S, p)$  where  $G$  is a  $\Sigma$ -mixture.

**Definition 4.3** A  $\Sigma$ -region-graph homomorphism  $f : (G_X, R_X, p_X) \rightarrow (G_Y, R_Y, p_Y)$  is a pair of maps  $(f_1, f_2)$  where  $f_1$  is a (total)  $\Sigma$ -homomorphism, and  $f_2$  is a (total) containment structure homomorphism, such that the homomorphisms respect the placing relation: for any agent  $C \in G_X$ , if  $p_X(C)$  is defined, we have that  $f_2(p_X(C)) = p_Y(f_1(C))$

**Definition 4.4** A sub- $\Sigma$ -region-graph  $D$  of  $X$  is a  $\Sigma$ -region-graph  $(G_D, S_D, p_D)$  where  $G_D$  is a  $\Sigma$ -subgraph of  $G_X$ ,  $S_D$  is a sub-containment-structure of  $S_X$ , and  $p_D$  is a restriction of  $p_X$  to a subset of the agents of  $G_D$ .

We call the obvious identity-like map from  $D$  into  $X$  the inclusion of  $D$  into  $X$ .

**Definition 4.5** A  $\Sigma$ -region-graph partial morphism is a span of an inclusion with a  $\Sigma$ -region-graph homomorphism.

Once again, one can compose by pullback, to yield a category of partial morphisms **PRk** (which includes total morphisms as a special case).

**Definition 4.6** A  $\Sigma$ -region-graph action is a  $\Sigma$ -region-graph partial morphism  $f : X \rightarrow Y$  where  $f_1$  is a  $\Sigma$ -action, and  $f_2$  is an containment structure action, and both  $X$  and  $Y$  are  $\Sigma$ -region site graphs.

**Definition 4.7** A  $\Sigma$ -region-graph matching is a total  $\Sigma$ -region-graph morphism  $(f_1, f_2)$  where the components are respectively  $\Sigma$  and containment structure matchings.

Note that the totality of matchings implies that the placing of agents found in the source of the matching must be fully respected in the target.

Finally, once again we can show our standard dynamics theorem:

**Proposition 4.8** Given any span  $\alpha : X \rightarrow Y, i : X \rightarrow S$  in **PRk** with  $\alpha$  an action and  $i$  an matching, and  $S$  a connected containment structure, there is a pushout

$$\begin{array}{ccc} X & \xrightarrow{\alpha} & Y \\ i \downarrow & & \downarrow i' \\ S & \xrightarrow{\alpha'} & T \end{array}$$

where  $\alpha'$  is an action and  $i'$  is an matching, and  $T$  is also connected.

## 5 Conclusion

We have presented a natural, conservative extension to the semantics of Kappa to include a simple dynamics of regions. It is hoped that this begins the process of enriching Kappa as a biological modelling language to represent compartments. There are several possible avenues of future research.

One important question is the complexity and efficiency of simulating Kappa with regions. A key attraction of classical Kappa is a markedly efficient simulation algorithm [9]. This algorithm relies on each rule application only enabling or disabling a small and bounded set of other applications — in other words, rule applications have only local impact. This allows efficient incremental update of a set of rule enablings.

Introducing containment structure, however, creates a kind of rule which can have decidedly non-local effect: if a container is moved or deleted, then this potentially affects (rules applying to) all of the contained Kappa agents. In general, in our relatively unconstrained full semantics, it is possible to define systems in which this results in severely worse simulation cost.

Nevertheless, it may be hoped that real systems, or perhaps sufficiently constrained and massaged real systems, might not excite this pathological behaviour. Preliminary work suggests that existing techniques can be effectively adapted to simulate constrained classes of system with reasonable efficiency; however, further work, and especially testing on real example systems of interest, is necessary to validate the new techniques.

It would be useful to establish connections with other space formalisms. In particular, representing containment structure separately from agent complexification is distinctly bigraphical in flavour, and so a connection should be attempted with [15,5].

Finally, it may be worthwhile to explore other notions of structure, both more and less expressive. As we gain experience, it may be desirable to explore both more complex and simpler options. One could for example add “wildcard” constructs to rule patterns, and other constructs of universal rendezvous. Work here might draw on comparisons with the Brane calculus [4] and perhaps more speculatively the Join calculus [12], as well as work on multi-level multisets [16].

Alternatively, with the goal of simplification, one might consider whether systems with regions at arbitrary depth beneath other regions can be translated into systems in which regions do not contain other regions, yielding “flat” region structures. The translation should (approximately) preserve the dynamics of the original system. Such techniques might have advantages in implementation efficiency for particularly demanding systems.

## References

- [1] Oana Andrei and Hélène Kirchner. Graph rewriting and strategies for modeling biochemical networks. In *Proceedings of the Ninth International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC 2007)*, pages 407–414. IEEE Computer Society, 2007.
- [2] Michael L. Blinov, James R. Faeder, Byron Goldstein, and William S. Hlavacek. BioNetGen: software for rule-based modeling of signal transduction based on the interactions of molecular domains. *Bioinformatics*, 20(17):3289–3291, 2004.
- [3] Michael L. Blinov, Jin Yang, James R. Faeder, and William S. Hlavacek. Graph theory for rule-based modeling of biochemical networks. In *Transactions on Computational Systems Biology VII*, volume 4230 of *Lecture Notes in Computer Science (Lecture Notes in Bioinformatics)*, pages 89–106, 2006.
- [4] Luca Cardelli. Brane calculi, interactions of biological membranes. In *Proceedings of the Second International Conference on Computational Methods in Systems Biology (CMSB 2004)*, volume 3082 of *Lecture Notes in Computer Science*, pages 257–278, 2004.
- [5] Troels C. Damgaard and Jean Krivine. A generic language for biological systems based on bigraphs. Technical Report TR-2008-115, IT Univeristy of Copenhagen, 2008.
- [6] Vincent Danos, Jérôme Feret, Walter Fontana, Russell Harmer, Jonathan Hayman, Jean Krivine, Christopher D. Thompson-Walsh, and Glynn Winskel. Graphs, rewriting and causality in rule-based models. *Submitted. Available from <http://www.cl.cam.ac.uk/~jmh93/>*, 2012.
- [7] Vincent Danos, Jérôme Feret, Walter Fontana, Russell Harmer, and Jean Krivine. Rule-based modelling of cellular signalling. In *Proceedings of the 18th International Conference on Concurrency Theory (CONCUR 2007)*, volume 4703 of *Lecture Notes in Computer Science*, pages 17–41, 2007.
- [8] Vincent Danos, Jérôme Feret, Walter Fontana, Russell Harmer, and Jean Krivine. Abstracting the differential semantics of rule-based models: exact and automated model reduction. In *Proceedings of the 25th Annual IEEE Symposium on Logic in Computer Science (LICS 2010)*, pages 362–381. IEEE Computer Society, 2010.
- [9] Vincent Danos, Jérôme Feret, Walter Fontana, and Jean Krivine. Scalable simulation of cellular signaling networks. In *Proceedings of the Fifth Asian conference on Programming Languages and Systems (APLAS 2007)*, volume 4807 of *Lecture Notes in Computer Science*, pages 139–157, 2007.
- [10] Vincent Danos, Jérôme Feret, Walter Fontana, and Jean Krivine. Abstract interpretation of cellular signalling networks. In *Proceedings of the Ninth International Conference on Verification, Model Checking, and Abstract Interpretation (VMCAI 2008)*, volume 4905 of *Lecture Notes in Computer Science*, pages 83–97, 2008.
- [11] Vincent Danos and Cosimo Laneve. Formal molecular biology. *Theoretical Computer Science*, 325(1):69–110, 2004.
- [12] Cédric Fournet and Georges Gonthier. The Join calculus: a language for distributed mobile programming. In *International Summer School on Applied Semantics (APPSEM 2000)*, volume 2395 of *Lecture Notes in Computer Science*, pages 268–332, 2000.
- [13] Leonard A. Harris, Justin A. Hogg, and James R. Faeder. Compartmental rule-based modeling of biochemical systems. In *Proceedings of the 2009 Winter Simulation Conference (WSC 2009)*, pages 908–919. IEEE, 2009.
- [14] Jean Krivine, Robin Milner, and Angelo Troina. Stochastic bigraphs. In *Proceedings of the 24th Conference on the Mathematical Foundations of Programming Semantics (MFPS XXIV)*, volume 218 of *Electronic Notes in Theoretical Computer Science*, pages 73–96, 2008.
- [15] Robin Milner. *The space and motion of communicating agents*. Cambridge University Press, 2009.
- [16] Nicolas Oury and Gordon Plotkin. Multi-level modelling via stochastic multi-level multiset rewriting. *Mathematical Structures in Computer Science*, 2011. Special issue on DCM 2010, to appear.