Abstracting the differential semantics of rule-based models: exact and automated model reduction

Vincent Danos, Jérôme Feret, Walter Fontana, Russell Harmer, Jean Krivine

Abstracting the differential semantics of rule-based models: exact and automated model reduction (revised version)

Vincent Danos[1,4], Jérôme Feret[2], Walter Fontana[3], Russell Harmer[3,4], and Jean Krivine[4]

[1] *University of Edinburgh*
[2] *LIENS, INRIA-ENS-CNRS*
[3] *Harvard Medical School*
[4] *CNRS, Université Paris-Diderot*

July 26, 2010

Abstract

Rule-based approaches (as in our own Kappa [18, 22], or the BNG language [26], or many other propositions allowing the consideration of "reaction classes") offer new and more powerful ways to capture the combinatorial interactions that are typical of molecular biological systems. They afford relatively compact and faithful descriptions of cellular interaction networks despite the combination of two broad types of interaction: the formation of complexes (a biological term for the ubiquitous non-covalent binding of bio-molecules), and the chemical modifications of macromolecules (aka post-translational modifications).

However, all is not perfect. This same combinatorial explosion that pervades biological systems also seems to prevent the simulation of molecular networks using systems of differential equations. In all but the simplest cases the generation (and even more the integration) of the explicit system of differential equations which is canonically associated to a rule set is unfeasible (eg, see Ref. [19, 43] for examples). So there seems to be a price to pay for this increase in clarity and precision of the description, namely that one can only execute such rule-based systems using their stochastic semantics as continuous time Markov chains, which means a slower if more accurate simulation.

In this paper, we take a fresh look at this question, and, using techniques from the abstract interpretation framework [17], we construct a reduction method which generates (typically) far smaller systems of differential equations than the concrete/canonical one. We show that the abstract/reduced differential system has solutions which are linear combinations of the canonical ones. Importantly, our method: 1) does not require the concrete system to be explicitly computed, so it is intensional, 2) nor does it rely on the choice of a specific set of rate constants for the system to be reduced, so it is symbolic, and 3) achieves good compression when tested on rule-based models of significant size, so it is also realistic.

1 Introduction

One of the major conceptual shift in modern biology is the gradual realization that proteins often decompose into clearly identifiable independent substructures called *domains* (among which specific recognition sites), which when suitably combined into a protein will determine a particular set of molecular functions [54]. There is a hitherto hidden alphabet of relatively few domain families that combine into the amazing variety of proteins and functions thereof (and incidentally, the archeology of domains has confirmed that innovation is by and large obtained by recombination of existing domains). As the ways in which domains interact within and between these modular proteins become better known, rudimentary forms of engineering of protein networks become possible (eg see Refs. [2, 55, 61] where *in vivo* rewirings of protein networks based on domain recombinations are investigated). Clearly this new ontology of the universe of biomolecules has fundamental consequences on the way protein networks should be formally described and modelled, and perhaps more importantly on the fashion in which they should be conceptualized as a computing medium. In particular, in order to express interactions in terms of domains, one needs another and more flexible mechanism than traditional reactions (aka Petri nets, or multiset rewriting). Because domain interactions will define whole classes of reactions at once, and because one wants to make these classes first class objects, one needs to be able to stipulate interactions at a refined level of granularity. That is the common idea at the heart of the various attempts at rule-based modelling languages.

In our own particular approach [23] (or that of the BNG language [4]), one has *agents* with *sites* that can be used to hold internal states and/or bind other sites - modelling both binding domains and chemically modifiable ones. *Species*, which within this paper will be taken to mean any complex structure that one can obtain by binding basic agents, now have an explicit internal structure. As a result, elements of the dynamics can be defined by a modeller without having to spell out a complete description of the species it applies to. To do so she/he writes rules specifying conditions under which elementary bindings and modifications happen. It must be noted that several similar agent-based languages have been proposed, some of them derived from the tradition of process algebras [12, 56, 58, 59]. In practice using process-centric approaches requires some ingenuity and results in a measure of encoding which makes further static analyses difficult (another potential limiting factor is that process calculi allow for the expression of many artefactual systems that have no meaning in terms of biochemical interactions). From that point of view, Kappa and BNGL strike a nice trade-off offering both process-algebraic conciseness and a straightforward expression of the domain combinatorics, that ultimately facilitates the analysis of networks [33]. It is also worth spelling out that Kappa is but an idealization of the relevant phenomenology of biomolecular networks as it leaves aside lower-scale effects such a steric hindrance, conformational changes etc, or phenomena relating to spatial organization such as diffusion, transport and membrane-related interactions.

1.1 The problem with rules

Now to the point of the paper. Systems of ordinary differential equations (ODEs) are widely used to probe the dynamics of biomolecular networks. As might be expected from the above explanations, the design of ODE models (usually based on systems of reactions) is a time-consuming process, as combinatorial complexity is avoided by deploying *ad hoc* successive approximations such as neglecting certain species or quotienting others [43]. This makes the resulting models hard to document, as this necessary complexity avoidance manoeuvering can only be based on intuitions and is therefore not an integral part of the description of the model - and in addition one has no means of controlling its impact on the dynamics. (In fact, one wonders what the quality of an intuition that one must have is!) By the same token, such models are even harder to modify, since any modification propagates throughout the model. Besides, the very process ties the choice of parameters to the preliminary approximation step, which is tantamount to foregoing a mechanistic interpretation of the said parameters. Yet, there is also an upside to their simplicity, as they can be simulated efficiently. Not so for rule-based models, as in all but the simplest cases, the explicitation of the underlying differential semantics is completely unfeasible (either because the differential system is infinite, or finite but too large to ever be written). It seems one has to pay a price for the use of rules, and forego their manipulation and study via a deterministic dynamics.

The object of this paper is to prove that this does not need to be the case, as it describes a method to generate a reduced differential semantics of any rule-based model. This reduced differential semantics is based on a linear change of variables, the computation of which is driven by the set of rules constituting the model (which is to say, it is sensitive to the presentation of the implicit dynamics as usual in static analysis). To find our reduction, we develop an analysis to detect which parts of our agents can influence the behaviour of other parts in the context of a given rule set. This mode of reasoning is reminiscent of dependency detection techniques used to prove that sensitive data does not depend on less sensitive data [1]. This analysis is then used to break down *species* (of which there are too many) into (typically) far fewer *fragments*, the behaviour of which can be self-consistently described by a reduced differential system. The derivation of the reduced system is proved to be sound by abstract interpretation: trajectories of the reduced systems are the projection of the trajectories of the original one (which is never explicitly computed). There is no approximation involved, and in particular the actual rate constants of the rules in the rule set are treated symbolically.

The algorithm that finds a set of fragments for a rule set is implemented (and downloadable together other Kappa related tools at `kappalanguage.org`). While we prove the correctness of our method, it comes with no guarantee that the reduced system will be smaller than the original one, and it is actually easy to construct examples that will defeat the analysis and result in no compression at all. So the question of the value of the method is a delicate one that cannot be decided by a theorem. However there is evidence that on actual models, the method performs well. Specifically, we have tried many examples among which the largest is a pilot model of a large and detailed section of the EGF system involving about 10^{19} species. The analysis produces about 10^5 fragments

- few enough to yield a differential semantics of the model which is executable. The fragmentation process itself lasts about $15'$ in this example, and since it is independent of the rate constants of the rules, it is enough to run it only once per rule set. Another consequence of the reduction being symbolic is that one can complement the exact reduction we propose with traditional approximate ones and obtain far better reductions when rate constants are known or posited. Further examples and explanations about the wider relevance of the present work to modelling in a biological context can be found in Ref. [31] (and its supplementary information).

1.2 Related work

Abstract interpretation has been widely used to approximate qualitative properties such as the potential configurations of biological systems [21, 28, 30, 39, 53] or their temporal properties [40]. In this paper, however, we consider *quantitative* properties as we abstract the system of differential equations/dynamical systems attached to such systems. In particular, the analysis we present here is different from our own reachability analysis in Ref. [21] which focuses on qualitative properties, and unlike the present one, can deal with dependencies only within a single agent.

Having said that, many quantitative static analysis methods already exist. Typically one tries to over-approximate the ranges of variables in continuous or discrete differential systems. In Refs. [7, 37, 38], discrete integration methods are proposed. In the field of reactive systems, suitable abstractions deal with ranges of variables: in Refs. [27, 29], abstract domains which discover and prove inductive invariants on output ranges of linear recursive filters are introduced; in Ref. [8] a framework allows to handle accurately the differential specification of numerical inputs of reactive systems; and static analyses are proposed in Ref. [11] to bound the error due to numerical integration. Our abstraction differs from all of the above as it does not attempt to abstract numerical values, but to find a reduced dynamical system.

The field of biomathematics has seen several propositions with the same ambitions, building on the actual particular structures of dynamical systems generated by reactions. In Ref. [6, 16], invertible changes of variables are used to block-diagonalize ODEs. Yet it is clearly preferable to avoid computing the starting ODEs at all, and our method indeed computes the reduced ODEs directly from the set of rules. The approach proposed in Ref. [15] does not presuppose an explicit ODE system, however, it is not automated and suffers from a combinatorial blow-up in the (rather common) case of chains of agents and site modifications that propagate through bindings [14, p. 82–83]. One can solve this problem by neglecting certain species, but then the dynamics of the system is not preserved. The framework in Ref. [5], for automatically reducing the ODEs of protein networks, suffers from the same lack of soundness when site modifications propagate through bindings. Our approach deals efficiently with this case.

Finally, abstract interpretation has also been used to analyse stochastic semantics. In [49, 50], a generic framework has been introduced in order to lift numerical domains to probabilistic semantics featuring non-determinism. This framework has inspired work for the abstraction of Markov chains generated by systems of ground reactions [13].

Our generic framework has stronger similarities with Ref. [24] which aims at abstracting stochastic distributions. However, our framework deals with differential semantics and, more to the point, is applied to a particular range of models that are of direct relevance.

1.3 Outline

We first discuss a system reduction performed on a simple set of reactions (§2). This example is the occasion to introduce the main intuitions and some of the terminology used further in the paper. Then, we introduce a notion of reduction for a differential system (§3) using concepts familiar from the framework of abstract interpretation. Once this is in place, we define Kappa formally -which means we have to deal with the unavoidable syntax of rules and rule application (§4), after which we can turn to the definition of its ground/concrete differential semantics (§5). After that we turn to the key construction, namely the dependency analysis on which the reduction method relies (§6) and which returns a set of new variables or fragments. From there we can define our abstract/fragment-based differential semantics (§7), and prove its soundness (not an easy proof!). Some numerical experimental results are reported (§8) - while these are not strictly necessary, we felt that they added to the end-to-end quality of the work we present here, which goes from a rather theoretical premise to the concrete results of an implementation.

An earlier version of our results appeared in Ref. [31]. The present paper however is of a more technical nature, and is the first which has a complete proof of the soundness of the reduction method. The proof and its setup are rather dense, and perhaps not as perspicuous as they could be, but hopefully the few ideas behind the construction will be clear to the reader.

2 Prologue

We describe in this section a simple reaction system and explain how the induced differential system can be decomposed into smaller ones. In the remainder of the paper, we will generalize in a non trivial way this kind of compression to differential systems induced by sets of Kappa rules, and show how one can derive a general algorithm to discover and perform such compressions.

Consider an agent B able to bind reversibly and simultaneously agents A and C. This means we have 6 possible species, A, B, C, and AB, BC, ABC. Let us also suppose that B is introduced into the system at some rate σ. This gives us the following set of reactions (rate constants are indicated next to each reaction; for reversible binding rules the first constant is the association one, the second is the dissociation one):

$$\begin{array}{lcll} & \rightarrow & B & \sigma \\ A, B & \leftrightarrow & AB & k_A, k'_A \\ A, BC & \leftrightarrow & ABC & k_A, k'_A \\ B, C & \leftrightarrow & BC & k_C, k'_C \\ AB, C & \leftrightarrow & ABC & k_C, k'_C \end{array}$$

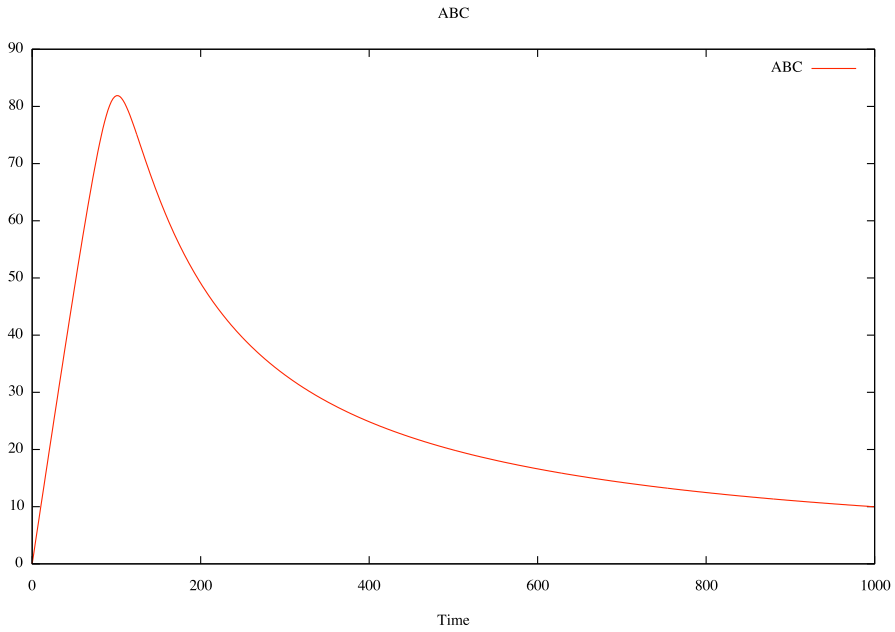


Figure 1: *The time course of* $[ABC]$*; one sees that the system eventually reaches larger concentrations of* B *where* ABC*s are diluted away. (The initial state consists of 100* A*s and* C*s, and zero* B*s; all rate constants are set to 1, as well as the volume correction; the plots are obtained using Maple.).*

The *mass action principle* states that a reaction's rate is given by its rate constant times the concentrations of its reactants. For instance the rate at which A binds B is $k_A[A][B]$, where $[A]$ is the traditional notation for the concentration of A (the number of As per unit volume). This gives us the associated differential system describing the time derivative of each of the 6 possible species:

$$
\begin{aligned}
[A]' &= k'_A([AB] + [ABC]) - k_A[A]([B] + [BC]) \\
[C]' &= k'_C([BC] + [ABC]) - k_C[C]([B] + [AB]) \\
[B]' &= \sigma + k'_A[AB] + k'_C[BC] - [B](k_A[A] + k_C[C])
\end{aligned}
$$

$$
\begin{aligned}
[AB]' &= k_A[A][B] + k'_C[ABC] - [AB](k'_A + k_C[C]) \\
[BC]' &= k_C[B][C] + k'_A[ABC] - [BC](k'_C + k_A[A]) \\
[ABC]' &= k_A[A][BC] + k_C[AB][C] - [ABC](k'_A + k'_C)
\end{aligned}
$$

Note that the differential system is *autonomous*, meaning that the time derivatives of variables do not explicitly depend on time. Fixing some values for the rate constants, and the initial state of the system, we can integrate it. Fig. 1 gives an illustration of the non-monotonic time course of $[ABC]$. We have assumed that the association and dissociation rate constants of A and B, k_A and k'_A, are the same whether or not B is bound to C, and similarly for the association and dissociation rate constants k_C, k'_C of B and C. We can exploit this symmetry and split the system into an A and a C subsystem

by introducing the following new variables:

$$[AB?] := [AB] + [ABC]$$
$$[B?] := [B] + [BC]$$

and we get the following differential system for the A subsystem:

$$[A]' = k'_A[AB?] - k_A[A][B?]$$
$$[AB?]' = [AB]' + [ABC]' = k_A[A][B?] - k'_A[AB?]$$
$$[B?]' = [B]' + [BC]' = \sigma + k'_A[AB?] - k_A[A][B?]$$

The point of this (linear) change of variables is that it produces a system where the derivatives of $[A]$, $[AB?]$, $[B?]$ can be expressed as functions of themselves. We say that our new set of variables is *self-consistent*, and we call these new variables *fragments* (a name which is supposed to remind the reader that fragments are partial species). In a similar fashion one can write a self-consistent differential system for the C subsystem using the variables $[C]$, $[?BC] := [BC] + [ABC]$, and $[?B] := [B] + [AB]$. Now, if we are interested in the time course of any of the above fragments/new variables, all we need to know is the three equations subsystem the fragment belongs to. Thus, we have achieved an exact reduction of the original system.

One could ask conversely if the information contained in the A and C subsystems is enough to reconstruct the original system. Specifically, is there a way to express (any of) $[ABC]$, $[AB]$ or $[BC]$, as a function of the new variables. It is easy to see that the linear mapping of the six old variables to the 6 new ones is of rank 5, and so not invertible. Nevertheless, one could think of reconstructing non-linearly the old variables by exploiting the idea that whether a B is bound to an A is independent of whether it is bound to a C. Suppose we set $[?B?] := [?BC] + [?B] = [AB?] + [B?]$ for the total concentration of B, we can then express the fraction of Bs with:
- both an A and a C attached as $[ABC]/[?B?]$,
- an A attached as $[AB?]/[?B?]$,
- and with a C attached as $[?BC]/[?B?]$.

If bindings are independent, the first expression is the product of the two remaining ones. Equivalently, one has $[ABC][?B?] = [AB?][?BC]$, which means $[ABC]$ (and hence all the other old variables) can be expressed in terms of the new variables. But Fig. 2 shows (for some arbitrary values of the various needed parameters) that the reconstruction is wrong - ie, there is a correlation between the B bindings.

We can look further into this simple example and define $\chi := [ABC][?B?]-[AB?][?BC]$ as a measure of non-independence. When $\chi > 0$, the presence of A and C correlate positively. A bit of symbol-pushing gives a closed formula for the time derivative of χ:

$$\chi' = \sigma[ABC] - \chi(k_A[A] + k_C[C] + k'_A + k'_C)$$

Clearly, and unless $\sigma = 0$, χ will not be everywhere zero. Although there is no apparent constraint between B's bindings, knowing if B is bound to an A does give information about when that B was created, which affects how likely it is that it is also bound to a

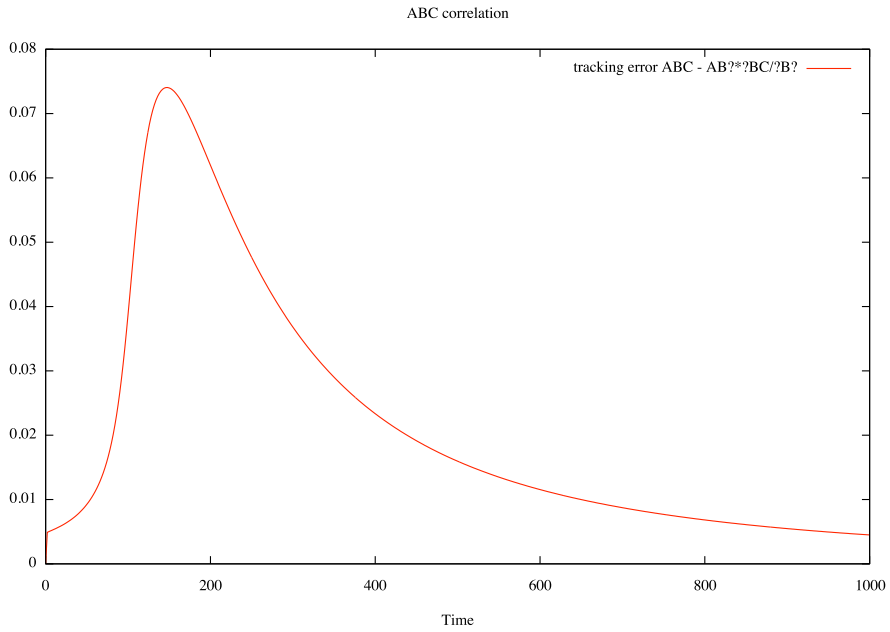


Figure 2: *The time course of a variant of the occupancy correlation on B, defined as $[ABC] - [AB?][?BC]/[?B?]$. If B's bindings were independent, it should be identically zero, and we do see a tiny deviation from zero which lags behind the production of ABC and eventually washes away for large times.*

C. One could say, by analogy with problems of non-interference, that there is an implicit information flow [1] which induces a correlation - here measured by χ. Yet, the set of reactions defining the differential system never observes that correlation.

This example, to which we will return, teaches us two things. Firstly, and most importantly, one can exploit structural features of a given differential system to identify sets of fragments, ie specific linear combinations of the system variables that have a self-consistent dynamics - seeing how this can be done in the case where the dynamics is described by rules, and not just reactions, is the main goal of this paper. As we will see, working with rules is an advantage here, as one can read directly the reduction from the rules without having to ever explicitly consider the ground set of reactions the rules correspond to. Secondly, even though a set of fragments is independently solvable, in general its behaviour (in our example the time courses of the subsystem variables $[A]$, $[AB?]$, $[B?]$, etc.) is not enough to recover that of the original system - perhaps unsurprisingly, some information can be lost in the reduction.

3 Exact reduction of differential systems

Before we turn to the syntax of rules, we need to define the specific notion of linear reduction of an autonomous differential system that we will use in the remainder of

the text. It is inspired by the methodology of abstract interpretation (AI), a general framework for approximating the semantics of programs [17]. Hence, it might be worth pointing out the correspondence with traditional AI concepts: our notion of reduction (defined below) can be seen as an *abstraction* map, that is to say a transformation going from a *concrete state space* (here the concentrations of all species) to an *abstract* one (here the concentrations of all fragments). The reduced dynamics corresponds to that of a *backward complete counterpart* [35, 36] to the concrete dynamics.

Let $\mathcal{V}$ be a finite set.

Maps from $\mathcal{V}$ to $\mathbb{R}$ form a normed vector space with norm:

$$\|\rho\| := \max_{X\in\mathcal{V}} |\rho(X)|$$

where $|\_|$ denotes the absolute value. For U a subset of $\mathcal{V} \to \mathbb{R}^+$, define $\|U\| = \sup_{\rho\in U} \|\rho\| \leq +\infty$.

If $\mathcal{V}$ is a set of species, and $\rho(X)$ the concentration of X, then $\|\rho\|$ controls the total number (per unit volume) of species in the system.

A ρ such that for all $X \in \mathcal{V}$, $\rho(X) \geq 0$ is called a *state*, and we write simply $\rho \geq 0$. We write $\rho[X \to r]$ for the state that maps X to $r \geq 0$ and otherwise is as ρ.

Consider another finite set $\mathcal{V}'$, and a map ψ from $\mathcal{V} \to \mathbb{R}$ to $\mathcal{V}' \to \mathbb{R}$, we say ψ is:
- *positive* if for all $\rho \geq 0$, $\psi(\rho) \geq 0$;
- *expansive* if for all subset U of $\mathcal{V} \to \mathbb{R}^+$, $\|\psi(U)\| < \infty$ implies $\|U\| < \infty$.

Definition 3.1. *A (positive autonomous) differential system over $\mathcal{V}$ is a map $\mathbb{F}$ from $\mathcal{V} \to \mathbb{R}$ to $\mathcal{V} \to \mathbb{R}$:*
- which is continuously differentiable, and for which
- there exists $\epsilon > 0$, and a family of positive and continuous maps G_X from $\mathcal{V} \to \mathbb{R}$ to $\mathcal{V} \to \mathbb{R}$ such that, for all $\rho \geq 0$, and X in $\mathcal{V}$, if $\rho(X) < \epsilon$ then:

$$-\rho(X) \cdot G_X(\rho[X \to 1]) \leq \mathbb{F}(\rho)(X)$$

By the Cauchy-Lipschitz theorem [42], $\mathbb{F}$ defines for any initial state ρ_0, a unique maximal differentiable $f : [0, T) \to \mathcal{V} \to \mathbb{R}$ such that $f(0) = \rho_0$, and $f' = \mathbb{F} \circ f$, with $T \leq +\infty$. This unique f is called *a solution* of $\mathbb{F}$, sometimes written f_{ρ_0} to make the dependency in the initial state explicit.

Note that it may be that $T < +\infty$: consider reaction $2A \to 3A$, one has $f'(t) = k[A]^2$ and the maximal solution $f(t) = 1/(A_0 - kt)$ is only defined on $[0, A_0/k)$.

The G_X family ensures that, starting from a positive initial state, if $\rho(X)$ gets close enough to zero, its derivative $\mathbb{F}(X)(\rho)$ becomes bounded below, so can't get too negative, so that $\rho(X)$ never reaches negative values.

In our application, $\mathcal{V}$ is the set of species generated by a molecular network, and $\mathbb{F}$ is given by mass action (as in §2). The repelling property is clearly satisfied since any negative contribution to $\mathbb{F}(\rho)(X)$ comes from a reaction that consumes X, and therefore must be of the form $-k \cdot \rho(X) \cdot \rho(Y_1) \dots \rho(Y_n)$.

Definition 3.2. *A reduction of a differential system $\mathbb{F}$ over $\mathcal{V}$ is a commuting square:*

$$\begin{array}{ccc} \mathbb{R}^{\mathcal{V}} & \xrightarrow{\mathbb{F}} & \mathbb{R}^{\mathcal{V}} \\ {\scriptstyle\psi}\downarrow & & \downarrow{\scriptstyle\psi} \\ \mathbb{R}^{\mathcal{V}^\sharp} & \xrightarrow{\mathbb{F}^\sharp} & \mathbb{R}^{\mathcal{V}^\sharp} \end{array}$$

where $\mathcal{V}^\sharp$ is a finite set of (abstract) variables, ψ is a positive, expansive and linear map from $\mathcal{V} \to \mathbb{R}$ to $\mathcal{V}^\sharp \to \mathbb{R}$, and $\mathbb{F}^\sharp$ is a differential system over $\mathcal{V}^\sharp$.

Note that since ψ is positive, by definition, it maps states to states. Nevertheless, we need values in $\mathbb{R}$ and not just $\mathbb{R}^+$ so that we can apply the abstraction function ψ also to vectors of derivatives $\mathbb{F}(\rho)$ which are not positive.

We can infer a strong form of soundness for our notion of reduction:

Theorem 3.3 (soundness)**.** *$T = T^\sharp$ and $f^\sharp_{\psi(\rho_0)} = \psi \circ f_{\rho_0}$.*

Proof. For $t < T$, one has:

$$\begin{array}{rcl} (\psi \circ f_{\rho_0})'(t) & = & \psi(f'_{\rho_0}(t)) \\ & = & \psi(\mathbb{F}(f_{\rho_0}(t))) \\ & = & \mathbb{F}^\sharp((\psi \circ f_{\rho_0})(t)) \end{array}$$

because ψ is linear (first equation), f_{ρ_0} is a solution of $\mathbb{F}$ (second equation), and $\psi \circ \mathbb{F} = \mathbb{F}^\sharp \circ \psi$ by assumption. Hence $\psi \circ f_{\rho_0}$ is differentiable on $[0, T)$, and it is a (unique) solution of $\mathbb{F}^\sharp$ for the initial condition $\psi(f_{\rho_0}(0)) = \psi(\rho_0)$ on $[0, T)$. In other words, on $[0, T)$, we have:

$$\psi \circ f_{\rho_0} = f^\sharp_{\psi(\rho_0)}$$

It follows that $T \leq T^\sharp$. But, in fact, $T = T^\sharp$. To see this, suppose $T < \infty$, then $\|f_{\rho_0}(t)\|$ diverges as t tends to T, and ψ being expansive, so does $\|\psi(f_{\rho_0}(t))\|$. □

A simple consequence is that for $t \in [0, T) = [0, T^\sharp)$:

$$0 \leq f^\sharp_{\psi(\rho_0)}(t)$$

Thus our reduction guarantees that: 1) trajectories of abstract variables can be computed directly in the abstract domain without loss of information; 2) positivity is preserved; 3) the life-time of the system is also preserved. (This is in contrast with syntactic program slicing that may not preserve non-termination - see eg [10, 34]).

3.1 Back to the example of §2

We can return to the example of §2, to illustrate the above definitions. The set $\mathcal{V}$ of concrete variables is A, B, C, AB, BC, and ABC, while the differential system $\mathbb{F}$ is given in new notation as:

$$\mathbb{F}(\rho)(A) \quad = \quad -k_A \cdot \rho(A)(\rho(B) + \rho(BC)) + k'_A(\rho(AB) + \rho(ABC))$$

and similarly for the other terms.

The set $\mathcal{V}^\sharp$ of abstract variables is A, $AB?$, $B?$, C, $?B$, and $?BC$, and the linear map ψ is given by:

$$\begin{aligned}\psi(\rho)(A) &= \rho(A)\\ \psi(\rho)(AB?) &= \rho(AB) + \rho(ABC)\\ \psi(\rho)(B?) &= \rho(B) + \rho(BC)\end{aligned}$$

and similarly for other terms. The abstract counterpart to $\mathbb{F}$, written $\mathbb{F}^\sharp$, is given by (we write $\rho^\sharp$ for an abstract state, that is to say a map from $\mathcal{V}^\sharp$ to $\mathbb{R}^+$):

$$\mathbb{F}^\sharp(\rho^\sharp)(A) \quad = \quad k'_A \cdot \rho^\sharp(AB?) - k_A \cdot \rho^\sharp(A) \cdot \rho^\sharp(B?)$$

and similarly for the other terms. And one can check the commutativity condition which expresses the self-consistency of ψ. If we verify it for A, we get:

$$\begin{aligned}\psi(\mathbb{F}(\rho))(A) &= \mathbb{F}(\rho)(A)\\ &= -k_A \cdot \rho(A) \cdot (\rho(B) + \rho(BC)) + k'_A \cdot (\rho(AB) + \rho(ABC))\\ &= -k_A \cdot \psi(\rho)(A) \cdot \psi(\rho)(B?) + k'_A \cdot \psi(\rho)(AB?)\\ &= \mathbb{F}^\sharp(\psi(\rho))(A)\end{aligned}$$

4 Kappa

We now introduce Kappa, which, in essence, is a certain type of graph rewriting system. We are going to introduce both a process-algebra notation (as in Ref. [23]), and a straight graphical notation. The former simplifies the presentation of the qualitative operational semantics, especially regarding finer notions of matching (using wildcards and binding types, see below) and the notion of inverting a rule - both of which we will need. It is also closer to our actual implementation. However, as for any process notation, and despite the fact that Kappa is a rather simple formalism, it can become cumbersome when it comes to the quantitative semantics where combinatorics and counting come to the fore. This is best done with a plain geometric/graphical view. Thus, we also introduce an equivalent graph-theoretical/graphical notation that is best suited to define the (concrete) differential semantics, as it allows an easier manipulation of the central notion of embedding (see definition below), and other constructions of a more geometric nature such as pushouts. We also present a notion of rule refinement which we will need in the construction of our reduced/abstract differential semantics (§7). We note in passing that a purely categorical presentation of the central tenets of Kappa was developed to handle the refinement problem [51]. In the longer term, it might be interesting to work in the framework of general graph transformation systems [25] and/or adhesive categories [46] - but exactly how much of the current theory and algorithmics [20] of Kappa can be extended to a more general setting remains to be seen.

4.1 Qualitative semantics

4.1.1 Expressions

We fix a finite set of agent types $\mathcal{A}$, a finite set of sites $\mathcal{S}$, and a *signature* map Σ from $\mathcal{A}$ to finite subsets of $\mathcal{S}$ assigning a set of sites to each agent type. As said in the introduction, in Kappa, sites can also hold modifiable internal states. This is convenient in practice but adds no difficulty to the theoretical side of affairs, so we will leave these aside (the implementation does consider them). With this simplification, the syntax of agents and expressions is given below:

$$\begin{array}{rcll} E & ::= & \varepsilon \mid a, E & \text{(expression)} \\ a & ::= & \emptyset \mid A(\sigma) & \text{(agent)} \\ \sigma & ::= & \varepsilon \mid s,\sigma & \text{(interface)} \\ s & ::= & x^{\lambda} & \text{(site)} \\ \lambda & ::= & \epsilon \mid i \mid A@x \mid - & \text{(binding state)} \end{array}$$

with $A \in \mathcal{A}$, $x \in \mathcal{S}$, and $i \in \mathbb{N}$ a natural number.

An *expression* is a (possibly empty) sequence of agents. An agent is either a *proper agent* or a *ghost agent* $\emptyset$; a proper agent is a name in $\mathcal{A}$ and an interface. An *interface* is a (possibly empty) sequence of sites with binding states; one writes x^{λ} for a site x with binding state λ. When the binding state of x is ϵ, we say x is *free*; otherwise x is *bound*. In examples, we generally omit the ϵ indicating a free site. (Be careful not to confuse ε the empty expression, or interface, and ϵ which denotes a free site.)

Note that the syntax distinguishes three types of bound sites. First, we have *binding labels*, $i \in \mathbb{N}$, when we know the binding partner (which is also bearing the same i); second, we have *binding types*, $A@x$, when we know the partner is the site x of some agent of type A; and last, we have *wildcards* '$-$' when we only know that a site is bound but have no further information about its partner. One can think of wildcards as semi-edges. (In practice, binding types and wildcards are key for obtaining more efficient compressions, using rule simplification techniques developed in Ref. [21]; more about this in the last section.)

A *structural equivalence*, which we use to tidy up an expression and match it against another one, is defined as the smallest equivalence relation on expressions such that:

$$\begin{array}{rcl} E, A(\sigma,s,s',\sigma'), E' & \equiv & E, A(\sigma,s',s,\sigma'), E' \\ E, a, a', E' & \equiv & E, a', a, E' \\ E & \equiv & E, \emptyset \\ i,j \in \mathbb{N} \wedge i \text{ does not occur in } E & \Rightarrow & E[i/j] \equiv E \\ i \in \mathbb{N} \wedge i \text{ occurs once in } E & \Rightarrow & E[\epsilon/i] \equiv E \end{array}$$

This equivalence says that: the order of sites in interfaces and of agents in expressions does not matter; ghost agents can be erased, binding labels can be injectively renamed and *dangling bonds* (ie binding labels that occur once) removed.

We now define useful classes of expressions.

Definition 4.1. *A pattern is an expression E such that:*
- (i) a site x occurs at most once in any agent $A(\sigma)$ in E;
- (ii) if x occurs in $A(\sigma)$ then $x \in \Sigma(A)$;
- (iii) each binding label i in E occurs exactly twice (there are no dangling bonds).

A pattern E is said to be: *proper* if it has only proper agents; *disconnected* if $E \equiv E', E''$ for some non-empty proper patterns E', E'' (in which case E' and E'' share no binding labels by condition *(iii)*). A *pattern component* is just a connected pattern. A *mixture* E is a non-empty proper pattern that is fully specified, ie each agent occurrence A in E documents its full interface $\Sigma(A)$, and sites can only be free or bear a binding label $i \in \mathbb{N}$. Finally, a *species* is a fully specified non-empty pattern component, or, equivalently, a connected mixture.

4.1.2 Rules

A *rule* is an ordered pair of patterns E_ℓ, E_r, sometimes written $E_\ell \to E_r$ (mostly in examples), with additional constraints (explained below). The left hand side (lhs) E_ℓ of a rule describes the agents taking part in it and various conditions on their binding states for the rule to apply. The right hand side (rhs) describes what the rule does. Ghost agents are used for agent creation in the lhs and agent removal in the rhs. A rule where both sides E_ℓ, E_r are fully specified proper patterns is a *reaction*. The key additional flexibility offered by rules, is that one *does not have to* use fully specified patterns, ie they can be left partial (see below for an illustration).

Definition 4.2 (constraints on rules)**.** *In a rule E_ℓ, E_r, the pattern E_r must be obtainable from the pattern E_ℓ in the following stepwise fashion (the order matters):*
- some wildcards and pairs of binding labels are removed (edge deletion);
- some ghost agents in E_ℓ are replaced by agents with full free interface (as specified by Σ) (agent creation);
- some agents with only free sites are replaced with ghost agents (agent deletion);
- some free sites are bound using fresh labels in $\mathbb{N}$ (edge creation).

It follows from the above constraint that both sides $E_\ell = a_1, \ldots, a_n$, and $E_r = a'_1, \ldots, a'_n$ must have the same number $n(r) \geq 0$ of agents (which is the reason for having have ghost agents in the syntax). Thus, there is a canonical bijective correspondence between agent occurrences in E_ℓ and E_r. Any two proper agents in correspondence have the same name and their interfaces must exhibit the same sites (possibly with different binding states). Another consequence of the above, is that rules are invertible except for the agent deletion steps. In this case, unless the deleted agent has a full free interface, we cannot recover in the inverse rule its exact binding context at the time of deletion.

Definition 4.3. *Let r be a rule with left hand side $E_\ell = a_1, \ldots, a_n$. A site x is said to be tested by r at position i, or (r, i)-tested, if x occurs in the interface of a_i in E_ℓ. A site x is said to be modified by r at position i, or (r, i)-modified, if one of the following holds:*
- $a_i = \emptyset$, $a'_i = A(\sigma')$, $x \in \sigma'$ (x is created)

- $a_i = A(\sigma)$, $a'_i = \emptyset$, $x \in \sigma$ *(x is deleted)*
- $a_i = A(\sigma)$, $a'_i = A(\sigma')$, $x \in \sigma$, $x \in \sigma'$, *and x has a different binding state in* σ *and* σ'.

Note that according to the constraint above (Def. 4.2), binding types can only be tested, not modified (although we will use this slight extension sometimes in examples).

A *system* is an initial mixture and a finite set of rules $\mathcal{R}$.

4.1.3 Prologue (continued)

As an illustration, we can refactor in Kappa the §2 example by introducing one site in B for each binding partner A, C, so that they can bind concurrently to B. Specifically, we set $\mathcal{A} = \{A, B, C\}$, $\mathcal{S} = \{a, b, c\}$, $\Sigma(A) = \{b\}$, $\Sigma(B) = \{a, c\}$, and $\Sigma(C) = \{b\}$. The species ABC is now written $A(b^1), B(a^1, c^2), C(b^2)$, and the rules emulating the earlier reactions are (rate constants are added to the right as in the case of reactions):

$$\begin{array}{ll} \emptyset \to B(a, c) & \sigma \\ A(b), B(a) \leftrightarrow A(b^1), B(a^1) & k_A, k'_A \\ B(c), C(b) \leftrightarrow B(c^1), C(b^1) & k_C, k'_C \end{array}$$

If one compares the last two rules with the corresponding four reactions, one sees that the rules are making the rate independence assumption explicit by, eg, not mentioning B's binding site c (resp. a) in the A (resp. C) to B binding rule.

4.1.4 Qualitative Semantics

We have now to explain how to apply a rule $r = E_\ell, E_r$ to a mixture E.

The first step is to "align" E with E_ℓ, ie, to use structural congruence to bring the participating agents to the front of E with their sites ordered as in E_ℓ, renaming binding labels and introducing ghost agents as necessary (for agents created by r). This yields an equivalent expression $E' \equiv E$ that *matches* the rule lhs, written $E' \models E_\ell$ (definition below). Note that, in so doing, one only uses structural equivalence on E, not on E_ℓ.

The actual notion of matching we use is straighforward. If E' and E_ℓ were plain graphs, to say that E matches E_ℓ would mean that E_ℓ is an induced subgraph of E (an NP-complete problem in general, but not in our case, that of *site graphs*, because of Prop. 4.4 below). The only slightly subtle point is the matching of a binding type by a binding label. For a binding label i in an agent $A(x^i)$ belonging to E' to match a binding type $A(x^{B@y})$ in E_ℓ, the unique other binding label i in E' must be of the form $B(y^i)$ (if there is no other binding label i, the match fails). To this effect, we use a partial look-up function π_E which given (A, x, i) returns the binding type of i's other occurrence in E if any. That is to say, $\pi_E(A, x, i) = B@y$ if y in B is the (unique) site in E with label i distinct from site x in A.

To define matching, set $E \models E_\ell := E \models_E E_\ell$ (auxiliary arguments are needed for the

inductive definition), and then define inductively $E' \models_E E_\ell$ as follows.

$$\begin{array}{rcl} x^\lambda & \models_E^A & x^\lambda \\ x^i & \models_E^A & x^- \\ x^i & \models_E^A & x^{\pi_E(A,x,i)} \end{array}$$

$$\begin{array}{rcl} \sigma & \models_E^A & \varepsilon \\ s \models_E^A s_\ell \wedge \sigma \models_E^A \sigma_\ell & \Rightarrow & s, \sigma \models_E^A s_\ell, \sigma_\ell \\ \sigma \models_E^A \sigma_\ell & \Rightarrow & A(\sigma) \models_E A(\sigma_\ell) \end{array}$$

$$\begin{array}{rcl} \emptyset & \models_E & \emptyset \\ E & \models_E & \varepsilon \\ a \models_E a_\ell \wedge E' \models_E E_\ell & \Rightarrow & a, E' \models_E a_\ell, E_\ell \end{array}$$

Note that matching can succeed in at most one way - all the non-determinism being handled by the 'alignment' phase. (Recall that mixtures do not use binding types or wildcards, so we do not need to consider such cases in the inductive definition above.)

The second step, once a match is realized, is to replace in E' the lhs E_ℓ by the rhs E_r. We write $E'[E_r]$ for the result of this substitution. This may produce dangling bonds (if r unbinds a wildcard bond or deletes an agent on one side of a bond, there is a "side-effect" as the other side of the bond needs to be erased as well) and/or ghost agents (if r deletes agents), which one can clean up using $\equiv$ afterwards.

Substitution is defined inductively as below.

$$\begin{array}{rcl} \lambda_r = i, \epsilon & \Rightarrow & x^\lambda[x^{\lambda_r}] = x^{\lambda_r} \\ \lambda_r = A@x, - & \Rightarrow & x^\lambda[x^{\lambda_r}] = x^\lambda \end{array}$$

$$\begin{array}{rcl} \sigma[\varepsilon] & = & \sigma \\ (s,\sigma)[s_r, \sigma_r] & = & s[s_r], \sigma[\sigma_r] \\ A(\sigma)[A(\sigma_r)] & = & A(\sigma[\sigma_r]) \end{array}$$

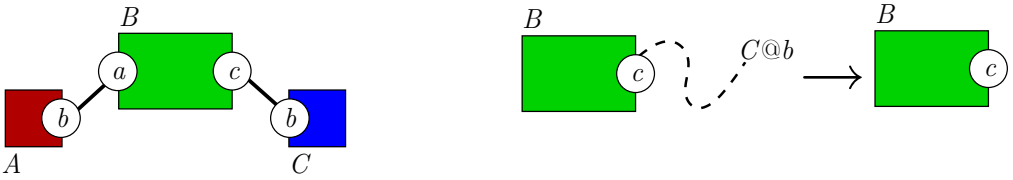
$$\begin{array}{rcl} \emptyset[a_r] & = & a_r \\ a[\emptyset] & = & \emptyset \\ E[\varepsilon] & = & E \\ (a,E)[a_r, E_r] & = & a[a_r], E[E_r] \end{array}$$

Finally, we can define the transition system generated by a set of rules $\mathcal{R}$. Suppose E_0, E_1 are mixtures, $r = E_\ell \to E_r$ is a rule in $\mathcal{R}$, $E_0 \equiv E_0'$, $E_0' \models E_\ell$, and $E_0'[E_r] \equiv E_1$, then we write $E_0 \to_r E_1$, and say that E_0 can be rewritten into E_1. Transitions are labelled by the rule they use.

4.1.5 An example

Here is a simple example:

$$\begin{array}{l} r := B(c^{C@b}) \to B(c) \\ E := A(b^1), B(a^1, c^2), C(b^2) \end{array}$$



(a) $A(b^1), B(a^1, c^2), C(b^2)$ (b) $B(c^{C@b}) \to B(c)$

Figure 3: *Graphical notation for a species (left) and for a rule (right).*

In order to apply the rule r to the expression E, we rewrite E to the equivalent form $E' = B(c^2, a^1), A(b^1), C(b^2)$; then we check $E' \models B(c^{C@b})$ which is true since $\pi_{E'}(B, c, 2) = C@b$; so we can proceed to the second step:

$$\begin{aligned} E'[B(c)] &= B(c^2, a^1)[B(c)], A(b^1), C(b^2) \\ &= B(c^2[c], a^1), A(b^1), C(b^2) \\ &= B(c, a^1), A(b^1), C(b^2) \\ &\equiv B(c, a^1), A(b^1), C(b) \end{aligned}$$

This particular rule has the side-effect of half-erasing an edge, and hence creates a dangling bond, which we can get rid of using $\equiv$ to recover a mixture.

4.2 Site graphs

As said, patterns (and mixtures) can be presented as site graphs, that is to say undirected graphs where typed nodes have sites, and edges form a partial matching on sites (meaning a site can be used in one edge only). Fig. 3 shows an example of the site graph corresponding to the ABC species from §2, as well as the graphical version of the rule we have examined in the example above. With this change of representation, a pattern component is simply a connected site graph, a mixture is a site graph where every node shows a full interface (according to the global signature Σ) and no binding type or wildcard occurs, and a species is a connected mixture.

4.2.1 Embeddings

Our notion of matching can be reformulated as a notion of site graph embedding. Suppose Z, Z', Z_ℓ are proper patterns such that $Z \equiv Z' \models Z_\ell$. Decompose Z, Z', Z_ℓ as sequences of (proper) agents:

$$\begin{aligned} Z &= A_1, \ldots, A_m, \\ Z' &= A'_1, \ldots, A'_m \\ Z_\ell &= B_1, \ldots, B_n \end{aligned}$$

where necessarily $0 < n \leq m$. In the absence of ghost agents the derivation of $Z \equiv Z'$ preserves the number of agents, and hence, it defines a unique permutation π mapping A'_i to $A_{\pi(i)}$ for $0 < i \leq m$. The restriction ϕ of π to $0 < i \leq n$ is called an *embedding* of Z_ℓ into Z. There may be several embeddings between Z_ℓ and Z - we write $[Z_\ell, Z]$ for the set of such embeddings.

With the notations just above, $\iota \in [Z_\ell, Z']$ with ι the canonical injection of $\{1, \ldots, n\}$ into $\{1, \ldots, m\}$, $\pi \in [Z', Z]$, and $\pi\iota \in [Z_\ell, Z]$. Thus, matches are special embeddings, corresponding to canonical injections, and the alignement procedure to produce a match can be seen as the factorization of an embedding ϕ as $\phi = \pi\iota$. Working directly with embeddings results in more perspicuous arguments as we will see.

One can extend the notion of embedding to patterns with ghost agents by defining $\phi \in [Z_1, Z_2]$ if $\phi \in [\hat{Z}_1, \hat{Z}_2]$, where $\hat{Z} \equiv Z$ is Z where all ghost agents have been removed. Patterns and embeddings then form a category (analogous to the category of plain graphs, where morphisms are embeddings as induced subgraphs). One says that ϕ is an *isomorphism* (iso) between Z_1 and Z_2 if ϕ has an inverse. It is easy to see that every ϕ in $[Z, Z]$ is an iso, usually called an automorphism (aka a symmetry) of Z. We write $|[Z, Z]|$ for the number of such automorphisms.

The graphical representation carries over nicely to rules. One can now apply a rule $r = E_\ell, E_r$ directly to a Z given $\phi \in [E_\ell, Z]$. The result of applying r to Z according to ϕ is always defined and yields a unique result, say Z', as well as a unique embedding ϕ' from E_r into Z'. Of course, there are, in general, many embeddings and the result will depend, again in general, on the particular choice made (in sharp contrast with reactions!).

For example, the pattern $B(c^{C@b})$ embeds twice in the mixture

$$A(b^1), B(a^1, c^2), C(b^2), B(a, c^3), C(b^3)$$

So we can apply the rule $B(c^{C@b}) \to \emptyset$ to get either of:

$$A(b^1), B(a^1, c^2), C(b^2), C(b)$$
$$A(b), C(b), B(a, c^3), C(b^3)$$

4.2.2 Epimorphisms

We recall an easy result which is a consequence of the strong requirements on an embedding, and the fact that all sites of an agent are distinguishable (ie form a set, not a multiset).

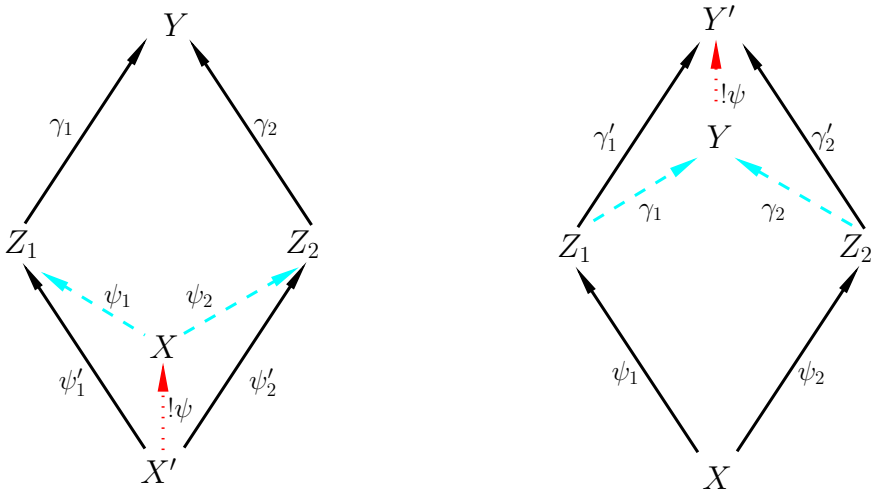
Lemma 4.4 (rigidity [20])**.** *An embedding of a pattern component C into a pattern Z is fully defined by the image of one agent. That is to say, whenever there are i, ϕ, and ϕ' such that $\phi \in [C, Z]$, $\phi' \in [C, Z]$ and $\phi(i) = \phi'(i)$, then $\phi = \phi'$.*

In general, an epimorphism (epi) is a $\psi \in [Z, X]$, such that for all $\phi \in [X, Z']$, $\phi' \in [X, Z']$, $\phi\psi = \phi'\psi$ implies $\phi = \phi'$. We can describe epis neatly.

Lemma 4.5. *Non-empty embeddings into a pattern component are epis. That is to say, if C is a pattern component, $\psi \in [Z, C]$ is an epi iff $\hat{Z}$ is not empty.*

It follows that epis enjoy a much weaker property than being surjective on nodes:

Corollary 4.6. *$\phi \in [Z_1, Z_2]$ is an epi iff the image of Z_1 intersects each component of Z_2.*



(a) Pullback.

(b) Idem pushout.

Figure 4: *Overlap: a commutative square of embeddings which is both a pullback and an idem pushout.*

One says that $\phi_1 \in [X, Z_1]$, $\phi_2 \in [X, Z_2]$ are isomorphic, if $\phi_1 = \phi\phi_2$, with ϕ an isomorphism in $[Z_1, Z_2]$. If both ϕ_is are epis, then ϕ is unique.

One says that $\phi \in [Z_1, Z_2]$ is a *straight epi* if $\hat{Z}_1$, and $\hat{Z}_2$ can be decomposed in two sequences of pattern components of equal length (possibly zero):

$$\begin{aligned} \hat{Z}_1 &= C_1, \ldots, C_n \\ \hat{Z}_2 &= D_1, \ldots, D_n \end{aligned}$$

and ϕ can be written as a (possibly empty) disjoint sum of $\phi_i \in [C_i, D_i]$.

Clearly, straight epis are epis. Any epi in $[Z_1, Z_2]$ must preserve the number of components, and is therefore canonically isomorphic to a straight one (just permute the components D_i). That isomorphism is unique because of the remark above.

4.2.3 Overlaps

We would like now to define the notion of *overlap* between patterns which will be central to our soudness argument. Since patterns might overlap in more than one way, to define an overlap unambiguously one has to provide additional data that one can think of as explicit glueing instructions.

Definition 4.7 (overlap)**.** *An overlap between patterns Z_1 and Z_2 is a commuting square $X, \psi_1, \psi_2, \gamma_1, \gamma_2, Y$ with $\psi_i \in [X, Z_i]$, $\gamma_i \in [Z_i, Y]$, which is both a pullback and an idem pushout (See Fig. 4).*

The triple X, ψ_1, ψ_2 is called a *span* and indicates a region common to Z_1 and Z_2, whereas the triple γ_1, γ_2, Y is called a *co-span* and indicates a way to glue Z_1 and Z_2.

The commuting square condition namely $\gamma_1\psi_1 = \gamma_2\psi_2$ (see Fig. 4) expresses the fact that the common region defined by the span is identified by the glueing defined by the co-span.

Given a co-span γ_1, γ_2, Y, there is always a span X, ψ_1, ψ_2 which makes the diagram commute. In fact there is always a universal such, called the *pullback* of γ_1, γ_2, Y. (It is universal in the sense that for any other solution X', ψ_1', ψ_2' there is a unique embedding $\psi \in [X', X]$ such that $\psi_1' = \psi_1\psi$, $\psi_2' = \psi_2\psi$. This implies that the pullback is unique up to unique isomorphism.)

Conversely, given a span X, ψ_1, ψ_2 there might be no co-span that 'closes the span' (ie, forms a commutative square), since away from the common region so defined, the patterns Z_1 and Z_2 might make incompatible choices. However, given any such a closing co-span, say γ_1', γ_2', Y', there is a universal compatible closing co-span say γ_1, γ_2, Y. In the particular case of a cospan which closes the span and is its own minimal compatible closing cospan (meaning that γ_1', γ_2', Y' is isomorphic to γ_1, γ_2, Y), one says the obtained square is an *idem pushout* [47].

We say the overlap is *non trivial* if X is not empty. We only consider non trivial overlaps in the rest of the paper, and for counting purposes, we fix a representative in each isomorphism class.

4.2.4 Overlaps (examples)

We consider first an example of two patterns Z_1 and Z_2 that can be glued in two (non-trivial) ways:

$$Z_1 = R(r^1, s), R(r^1),$$
$$Z_2 = R(a)$$

Depending on which agents R one chooses to identify we obtain two glueings. The first glueing is obtained using span $R(), \phi_1, \phi_2$ and co-span $\phi_3, \phi_4, R(r^1, s, a), R(r^1)$ where ϕ_1, ϕ_2, ϕ_3, ϕ_4 are identical maps. The other glueing is obtained using span $R(), \phi_1', \phi_2'$ and co-span $\phi_3', \phi_4', R(r^1, s), R(r^1, a)$, where ϕ_1' and ϕ_4' map 1 to 2, and ϕ_2' and ϕ_3' are identical maps. In both cases the obtained square is an overlap. This tells us, concretely, that we definitely need glueing instructions (ie a span) to know what to do.

Here is another, more subtle, example of two idem pushouts (and overlaps) on the same initial span (maps are uniquely definable so we omit them):

$$Z_1 = A(x^1), B(x^1) \leftarrow A() \rightarrow A(y^1), B(y^1) = Z_2$$
$$Z_1 \rightarrow A(x^1, y^2), B(y^2), B(x^1) \leftarrow Z_2$$
$$Z_1 \rightarrow A(x^1, y^2), B(y^2, x^1) \leftarrow Z_2$$

Concretely, this means that there are essentially different ways to execute glueing instructions specified by a span - so in effect, we need not just a span but a complete square. In the sequel, such constructions will always be made in a context where a commuting square is given (as in Fig. 6, 7), so it will not be a problem.

5 Concrete differential semantics

We turn now to the definition of the concrete differential semantics of a rule set - which eventually will serve as our reference semantics for the soundness of our reduction method. That is to say, we need to define a differential system $\mathbb{F}$ as in Def. 3.1. As we have seen in the prologue, this is a simple operation if one starts from a set of reactions. In the case of rules, one needs a first step to map the rule set at hand into a set of reactions. To do this effectively, we introduce a key technical notion, that of rule refinement (studied at length in Ref. [52]).

5.1 Rule refinements

Fix a rule $r = E_\ell, E_r$. Suppose Z is a pattern such that $Z \models E_\ell$, we define the *left refinement* of r by Z, written $Z\{r\}$, as the rule $Z, Z[E_r]$. Similarly, suppose Z is such that $Z \models E_r$, we define the *right refinement* of r by Z, written $\{r\}Z$, as the rule $[E_\ell; E_r]Z, Z$ using inverse substitution as defined inductively below.

$$
\begin{aligned}
&[\epsilon;\epsilon]E = E\\
&[a_\ell, E_\ell; a_r, E_r](a, E) = [a_\ell; a_r]a, [E_\ell; E_r]E
\end{aligned}
$$

$$
\begin{aligned}
&[a_\ell;\emptyset]\emptyset = a_\ell\\
&[\emptyset; a_r]a = \emptyset\\
&[A(\sigma_\ell); A(\sigma_r)]A(\sigma) = A([\sigma_\ell; \sigma_r]\sigma)\\
&[\epsilon;\epsilon]\sigma = \sigma\\
&[s_\ell, \sigma_\ell; s_r, \sigma_r]s, \sigma = [s_\ell; s_r]s, [\sigma_\ell; \sigma_r]\sigma
\end{aligned}
$$

$$
\begin{aligned}
&[x^{\lambda_\ell}; x^{\lambda_r}]x^\lambda = x^{\lambda_\ell} \text{ if } \lambda_\ell = \epsilon, i \in \mathbb{N}\\
&[x^{A@x}; x^{A@x}]x^\lambda = x^\lambda\\
&[x^{\text{-}}; x^{\text{-}}]x^\lambda = x^\lambda\\
&[x^{\text{-}}; x^{\lambda_r}]x^\lambda = x^{\text{-}} \text{ if } \lambda_r \neq -
\end{aligned}
$$

It is easy to see that $[E_\ell; E_r]E_r = E_\ell$ as it should. Note that inverse substitution depends not only on E_ℓ, but also on E_r. In fact, we need E_r only in the last two equations below to test whether the wildcard $x^{\text{-}}$ is deleted by r or not. This is different from substitution (defined earlier in §4.1.4) which only depends on E_r.

It is perhaps useful to give an example of left/right refinement where this is used:

$$
\begin{aligned}
&r = A(x^{\text{-}}), \emptyset \to A(x^1), B(x^1)\\
&Z_r = A(x^1, y), B(x^1)\\
&Z_\ell = A(x^{\text{-}}, y)\\
&\{r\}Z_r = A(x^{\text{-}}, y), \emptyset \to A(x^1, y), B(x^1) = Z_\ell\{r\}
\end{aligned}
$$

Refinements extend in a straightforward way (by factoring embeddings via matchings) to the more general case where we have an embedding of Z (instead of just a matching) into the lhs or the rhs of the rule of interest.

We write:
- $(Z, \phi)\{r\}$ for the left refinement of r along $\phi \in [E_\ell, Z]$
- $\{r\}(Z, \psi)$ for the right refinement of r along $\psi \in [E_r, Z]$

If either ϕ or ψ is a canonical injection ι, we get the earlier notion, ie $(Z, \iota)\{r\} = Z\{r\}$, and $\{r\}(Z, \iota) = \{r\}Z$. We say two refinements (left or right) are isomorphic if their defining embeddings are.

Definition 5.1. *Let r be a rule, and ϕ be a straight epi in $[E_\ell, M]$ where M is a mixture. The pair ϕ, r_ϕ where r_ϕ is the left refinement*

$$r_\phi := (M, \phi)\{r\}$$

is called the ground refinement of r along ϕ.

In the above definition, it is important to keep track of the embedding that generates the refinement r_ϕ, since there can be many ϕ that produce the same r_ϕ. Note also that the requirement that ϕ is a straight epi implies that E_ℓ and M have the same number of components (whereas an epi could, in general, see the number of components decrease).

If E_ℓ consists only of empty agents, then there is only one refinement, the empty map (which is a straight epi!), and E_r is necessarily a sequence of species (by Def. 4.2).

Proposition 5.2. *Consider $\{r\}(Z, \gamma) =: E'_\ell, Z$ the right refinement of a rule $r = E_\ell, E_r$ along $\gamma \in [E_r, Z]$. Suppose that γ is an epi, then the induced $\gamma' \in [E_\ell, E'_\ell]$ is also an epi, and if γ' preserves the number of components, every ground refinement of $\{r\}(Z, \gamma)$ determines injectively a ground refinement of r.*

Proof. In order to extend E_r with an epi γ, one must extend the interface of agents in E_r (Cor. 4.6). Such 'extended' agents cannot be created by r, since created agents get a maximal interface (Def. 4.2). It follows that every interface extension will transfer by inverse subsitution to the left hand side, hence γ' is an epi. If, in addition, E'_ℓ has the same number of components as E_ℓ, then any straight epi in $[E'_\ell, M]$, ie any ground refinement of $\{r\}(Z, \gamma)$, gives rise to an embedding $\psi\gamma' \in [E_\ell, M]$ which also preserves the number of components, and is therefore canonically isomorphic to a ground refinement of r. Because γ' is an epi, this correspondence is 1-1. □

We are not saying that γ' must always preserve the number of components. Here is an example:

$$A(x), A(x) \to A(x^1), A(x^1)$$
$$\gamma : A(x^1), A(x^1) \to A(x^1, y^2), A(x^1, y^2)$$
$$\gamma' : A(x), A(x) \to A(x, y^2), A(x, y^2)$$

5.2 Concrete differential semantics

5.2.1 The concrete domain

To obtain the concrete differential semantics of a rule set $\mathcal{R}$ we need to choose a *finite* set of species $\mathcal{V}$ closed under the rules in $\mathcal{R}$, which contains all species present in the

system's initial state, and has at most one representative per species isomorphism class. (By closed under r, we mean that any application of r to a sequence of species in $\mathcal{V}$ produces a sequence of species in $\mathcal{V}$.)

How do we choose $\mathcal{V}$ in practice? We cannot always take all species as defined by the signature Σ, as there might be countably many. A better choice is to use the efficient symbolic description of a $\mathcal{V}$ closed under $\mathcal{R}$ and containing any given set of initial species obtained in Ref. [21,30]. Having said that, in this theoretical development, we just assume we have a proper $\mathcal{V}$. (The finiteness assumption is a limitation discussed again in the conclusion.)

Following §3, a state ρ will be a map from $\mathcal{V}$ to $\mathbb{R}^+$, mapping each $S \in \mathcal{V}$ to its concentration $\rho(S) \geq 0$.

5.2.2 The concrete differential system

Suppose now each rule r in $\mathcal{R}$ is equipped with a *rate constant* $k(r)$ (a positive real number).

We construct the differential system $\mathbb{F}$ in a piecewise fashion, by defining for each rule r in $\mathcal{R}$, each ground refinement ϕ of that r (up to iso, and with all components in $\mathcal{V}$), and each species S in $\mathcal{V}$, the positive and negative contributions of the pair r, ϕ to $\mathbb{F}(\rho)(S)$ in a state ρ.

Pick a rule $r = E_\ell, E_r$ in $\mathcal{R}$, and a straight epi ϕ from E_ℓ into some mixture $M = R_1, \ldots, R_n$ over $\mathcal{V}$. Decompose the ground refinement ϕ, r_ϕ into components:

$$r_\phi := R_1, \ldots, R_n \to P_1, \ldots, P_m$$

where the 'products' P_j are the species in $\mathcal{V}$ produced by the application of ϕ, r_ϕ to $R_1, \ldots, R_n$. The P_js are in $\mathcal{V}$ because the 'reactants' R_i are also in $\mathcal{V}$, and $\mathcal{V}$ is closed under rules, by assumption.

Define (following §3) the activity of r_ϕ (aka flux, rate, velocity, etc.) in a state ρ as

$$\gamma(r) \prod_i \rho(R_i)$$

with $\gamma(r) = k(r)/|[E_\ell, E_\ell]|$. Recall $|[E_\ell, E_\ell]|$ stands for the number of automorphisms of E_ℓ. This predivision of the rate constant of the rule by its number of automorphisms is the usual convention (see the discussion below).

The activity of each r_ϕ contributes to the consumption and production of species in $\mathcal{V}$ as follows:

$$\begin{array}{lcll} \mathbb{F}(\rho)(R_i) & \stackrel{+}{=} & -\gamma(r)\rho(R_1)\cdots\rho(R_n) & \text{for } 1 \leq i \leq n \\ \mathbb{F}(\rho)(P_j) & \stackrel{+}{=} & \gamma(r)\rho(R_1)\cdots\rho(R_n) & \text{for } 1 \leq j \leq m \end{array}$$

Note that monomials can accumulate for different values of j or k. Eg for the ground rule:

$$r_\phi = R, R \to P$$

we find that $\mathbb{F}(\rho)(R) \stackrel{+}{=} -2\gamma(r)\rho(R)^2$, that is to say the rule contributes twice to the consumption of R.

Since $\mathcal{V}$ and $\mathcal{R}$ are finite, so is the number of ground refinements of rules in $\mathcal{R}$ (up to iso), and hence the total number of contributions is finite as well. So $\mathbb{F}(\rho)(S)$ is a well-defined finite sum of monomials of the above form for each $S \in \mathcal{V}$.

This $\mathbb{F}$ constitutes the *concrete differential system* or semantics of $\mathcal{R}$, and will be the reference for proving the soundness of reductions.

5.2.3 Discussion

It might be useful to point at the relationship of the above definition with the usual notion of activity from chemical kinetics. Suppose r is already a reaction, that is to say r's lhs can be written $S_1, \ldots, S_n$ with $S_i \in \mathcal{V}$. Then r has $\prod_i |[S_i, S_i]|$ ground refinements, which are all identical.

By our definition above the *joint* activity of r's ground refinements is:

$$\prod_i |[S_i, S_i]| \cdot k(r)/|[E_\ell, E_\ell]| \cdot \rho(S_1) \cdots \rho(S_n)$$

Now if the S_is are considered as pure names as in chemical kinetics (aka Petri nets, multiset rewriting), the activity of the corresponding reaction is:

$$k(r)/\tau \cdot \rho(S_1) \cdots \rho(S_n)$$

where τ is the number of *multiset* automorphisms of $S_1, \ldots, S_n$.

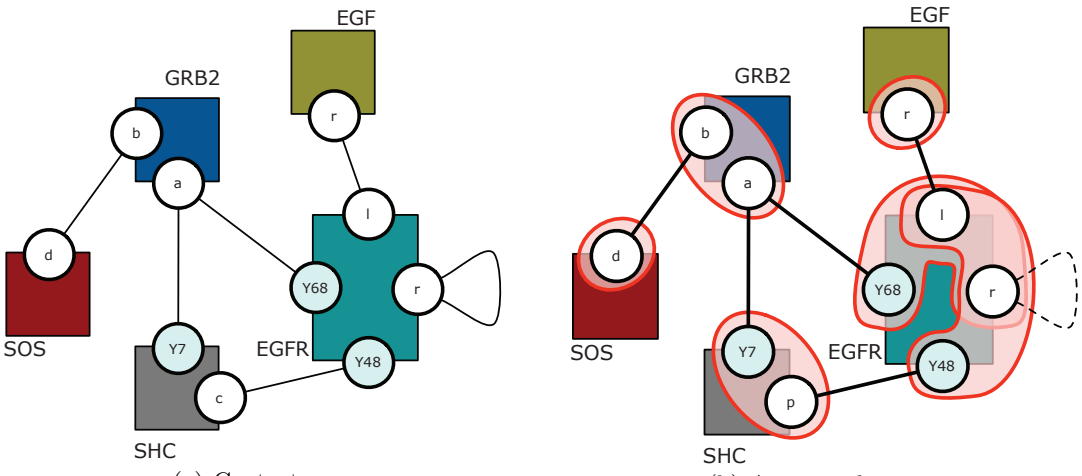
Clearly $\tau = |[E_\ell, E_\ell]|/\prod_i |[S_i, S_i]|$, since the internal structure is hidden, which is to say that the joint activity of the ground refinements of r is the same as its usual activity as a flat reaction.

A point worth noticing is that the differential semantics we have just defined ignores rule applications where two pattern components of E_ℓ embed into two distinct areas of the *same* species - since we have required ground refinements ϕ to preserve the number of connected components. In the stochastic semantics, one can allows for such rule instances. It does not matter in the sense that the differential semantics is the expected behaviour of the stochastic one when both the initial mixture size and the volume diverge with a constant ratio [44,45] - in which case, ground refinements with lower arities, corresponding to epis which do not preserve the number of components, are negligible anyway.

6 Fragments: the abstract domain

We now turn to the construction of the abstract/reduced semantics. The first step is to define a family of suitable pattern components called *fragments*, that will be the basis of our abstract domain and enable the definition of a (backward complete) counterpart $\mathbb{F}^\sharp$ to $\mathbb{F}$ (next section). To define our fragments, we will use an annotated contact map (defined below) which over-approximates the correlations that can be observed by the rules.

Throughout this section and the next we suppose fixed a rule set $\mathcal{R}$, an initial state, and a finite superset of reachable species $\mathcal{V}$, forming the basis of our concrete domain - as discussed in §5.2.1.



(a) Contact map.

(b) Annotated contact map.

Figure 5: *Maps for the early EGF model.*

The contact map associated to $\mathcal{V}$ is a summary of the bindings found in the species of $\mathcal{V}$. Specifically, the nodes of the contact map are the agent types occurring in $\mathcal{V}$ with their full set of sites according to the signature Σ, with an edge between two sites iff these two sites form a bond in *some* species in $\mathcal{V}$. Therefore, any species in $\mathcal{V}$ projects uniquely to the contact map.

To lighten the notations we will suppose that Σ maps different agent types to disjoint set of sites - or in other words, that a site can only belong to one type of agent. An example of a contact map is given in Fig. 5(a). As one can see, sites in the contact map may be connected to several sites, which implies a competition between two binding states; indeed, an agent can even be connected to itself (via the same, or different, sites). (This means that the contact map is not a site graph in general - one can rather think of it as a type for a set of site graphs.)

A *parsimonious covering* of a set X is a set $\mathcal{C}$ of subsets of X such that $\cup\mathcal{C} = X$ and, for no $X_1, X_2 \in \mathcal{C}$, $X_1 \subset X_2$ (strict inclusion); the elements of $\mathcal{C}$ are referred to as *classes* of $\mathcal{C}$. Hence a covering is not necessarily a partition, and we will use this flexibility. One can define a partial order on coverings by setting $\mathcal{C}_1 \sqsubseteq \mathcal{C}_2$ if for any $X_1 \in \mathcal{C}_1$, there exists $X_2 \in \mathcal{C}_2$ such that $X_1 \subseteq X_2$.

Definition 6.1. *An annotated contact map (aCM) is a contact map where in addition:*
- (i) each agent A has a parsimonious covering $\mathcal{C}_A$ of $\Sigma(A)$;
- (ii) a subset of edges is distinguished.

Distinguished edges are called *soft* (represented with dashed lines in Fig. 5(b)), the others are called *solid*. The idea is that a class in the covering of an agent denotes a relationship between sites that has to be tracked in order to define the abstract dynamics of the system. Solid edges indicate bonds that also need tracking.

6.1 Dependency analysis

A rule is *trivial* if it deletes a bond without testing or modifiying anything else, ie it has one of the following forms:

$$A(a^1), B(b^1) \to A(a), B(b)$$
$$A(a^-) \to A(a)$$

Let r be a rule with left hand side E_ℓ. A site x is said to be a *docking site* for r at position i, or an (r,i)-docking site, if x occurs at position i, and there is a path from x leading to a modified agent. To be precise, this means that there is a finite sequence of edges (x_k, y_k), $k \leq n$, belonging to E_ℓ such that: $x = x_0$; y_k, x_{k+1} are distinct sites of the same agent in E_ℓ for $k < n$; and y_n belongs to an agent which has a site modified by r (possibly y_n itself).

Definition 6.2. *An aCM is valid with respect to a rule set $\mathcal{R}$ if satisfies the following constraints.*

For every rule r in $\mathcal{R}$, and every $i \in \mathbb{N}$:
- (1.i) if x is an (r,i)-docking site or an (r,i)-modified one, and y is (r,i)-tested, every covering class which contains x also contains y;
- (1.ii) the set of (r,i)-tested sites is included in a covering class.

For every non-trivial rule r in $\mathcal{R}$, any edge in the aCM which (2.i) either occurs in the lhs of r, or (2.ii) can be deleted by r must be solid. (Due to side-effects, both ends of the edge need not actually occur in r.)

Finally, (2.iii) if a cycle in a species in $\mathcal{V}$ has only one soft edge, then no (trivial) rule in $\mathcal{R}$ can delete it.

Note that trivial rules do not constrain the aCM, as they automatically verify *(1.i)*, *(1.ii)*. Clause *(2.iii)* ensures that trivial rules don't generate ambiguous production terms on fragments (see Prop. 6.8).

The idea behind the above definition is that when no correlations are observable between (not necessarily disjoint) subparts of a species, one can safely fragment this species into its subparts (which is why we call them fragments!). Each valid annotated contact map tells us how to obtain fragments. Soft edges specify where we can cut species (using binding types), and coverings specify which sites must be kept together in interfaces.

Definition 6.3. *Given an aCM, a fragment for that aCM is a proper pattern component F such that:*
- F has no wildcard,
- F embeds in some S of $\mathcal{V}$,
- (i) each agent interface in F projects to a covering class;
- (ii) each binding label in F projects to a solid edge;
- (iii) each binding type in F projects to a soft edge.

Note that clause *(i)* makes no obligation to choose the same class for different oc-

currences of the same agent type in F. This is key for the flexibility of fragments.

There are two particular valid aCMs (which may coincide). The *trivial* aCM arises by taking for all agent types A the trivial covering $\{\Sigma(A)\}$, and taking all edges to be solid. Its set of fragments is the set of all species in $\mathcal{V}$. The *minimal* aCM is obtained by choosing edges soft whenever possible, and a $\sqsubseteq$-minimal covering for every agent type. (It is easy to see that there is such a minimal aCM.)

We suppose hereafter that we have fixed a valid aCM and we write $\mathcal{V}^\sharp$ for the finite set of fragments it generates according to the definition above. By definition, every fragment can be embedded in some species in $\mathcal{V}$, so this set is finite. Typically it is much smaller than $\mathcal{V}$.

6.2 Discussion

We see that the fewer non-trivial rules one has in $\mathcal{R}$, and the smaller their components, the fewer fragments are generated by the minimal aCM. Since the efficiency of the reduction is eventually measured by the number of fragments that are generated (the smaller the better) one would like to minimise that number as much as possible. The separate treatment of trivial rules obviously helps as it allows more soft edges. Another complementary way to improve reduction is to remove redundant tests in a rule (because of clauses *(1.i)*, *(1.ii)* above). This is one application of the qualitative static analysis proposed in Ref. [21], and we do use it in real examples. (More about efficiency matters in the application section.) In passing, these are the reasons that have prompted us to introduce binding types in the syntax of Kappa.

Another point worth of notice is that intermediate granularities can be useful. One can refine the aCM if there is need to express the concentration of a pattern component C of interest. This amounts to considering a fictitious rule $C \to C$, which may incur larger covering classes and fewer soft edges, and consequently less of a reduction.

6.3 Example

An example aCM obtained from a simple model of the early events in the *EGF* pathway [3] is given in Fig. 5. Let us examine one of the rules, a non-trivial dissociation:

$$EGFR(Y48^1), SHC(p^1, Y7^2), GRB2(a^2, b) \to \\ EGFR(Y48), SHC(p, Y7^2), GRB2(a^2, b)$$

Since it is non-trivial, and it contains modified sites, the rule does generate constraints: by *(2.i)* both the $(Y48, p)$ and $(a, Y7)$ edges must be solid, by *(1.i)* any class that contains p (modified at position 2) must also contain $Y7$ (tested at position 2); again by *(1.i)* any class that contains a (a docking site at position 3) must also contain b (tested at position 3). (To see that a is indeed a docking site, we follow its edge to position 2 and we find p modified at position 2.)

If one looks at the other rules of the model (not shown here), one sees that the sites $Y48$ and $Y68$ are independent, but can both only be activated if the site r is bound, a binding which only happens if the site l is bound. This determines two covering classes

$\{l, r, Y48\}$ and $\{l, r, Y68\}$ for *EGFR*. The edge from r to itself can be kept soft because, in the same model, the state of one receptor in a dimer does not affect the behaviour of the other. This is how the aCM example is derived.

6.4 Abstraction function

Having now defined our set of abstract variables $\mathcal{V}^\sharp$, the next step is to define the abstraction function ψ from $\mathcal{V} \to \mathbb{R}$ into $\mathcal{V}^\sharp \to \mathbb{R}$. We first need a couple of auxiliary definitions related to the counting of pattern components.

Given a concrete state $\rho \in \mathcal{V} \to \mathbb{R}^+$, we define the (real positive) number of embeddings $\overline{\rho}(C)$ of a pattern component C in to ρ as:

$$\overline{\rho}(C) = \textstyle\sum_{S \in \mathcal{V}} \rho(S) \cdot |[C, S]|$$

It is also convenient to define a version of $\bar{\rho}$ which counts instances or concentrations, that is to say embeddings up to automorphisms:

$$\widetilde{\rho}(C) := \frac{\overline{\rho}(C)}{|[C, C]|}$$

Clearly, $[S, S']$ is empty unless, $S = S'$ (recall that we have picked one representative per iso class in $\mathcal{V}$), so $\overline{\rho}(S) = \rho(S) \cdot |[S, S]|$, and $\widetilde{\rho}(S) = \rho(S)$, hence $\widetilde{\rho}(S)$ is an extension of ρ.

By convention we set $\overline{\rho}(\emptyset) := 1 = \widetilde{\rho}(\emptyset)$.

Finally, for any fragment F, we define:

$$\psi(\rho)(F) = \widetilde{\rho}(F)$$

Clearly the function ψ is a linear mapping with positive coefficients.

We can check that it is expansive (as required in §3). Suppose one has an unbounded subset U of $\mathcal{V} \to \mathbb{R}^+$, then there must be an $S \in \mathcal{V}$ such that $\sup_{\rho \in U} \rho(S) = +\infty$. Pick such an S, and a fragment F_S that embeds into S (clearly there is always one). One has:

$$\begin{array}{rcl} \psi(\rho)(F_S) & = & \textstyle\sum_{S' \in \mathcal{V}} \rho(S') \cdot |[F_S, S']| / |[F_S, F_S]| \\ & \geq & \rho(S) \cdot |[F_S, S]| / |[F_S, F_S]| \end{array}$$

so $\sup_{\rho \in \psi(U)} \psi(\rho)(S) = +\infty$ as well.

6.5 Fragment properties

We identify in the following the key properties of our set of fragments. These will be sufficient for the derivation and the proof of correctness of the abstract counterpart $\mathbb{F}^\sharp$ to $\mathbb{F}$ in the next section. Some of the proofs are only sketched.

We define a *subfragment* as a pattern component that can be embedded in a fragment.

Proposition 6.4 (growth)**.** *Let C be a subfragment, its concentration $\widetilde{\rho}(C)$ can be expressed as a linear combination of concentrations of fragments.*

Proof. The idea is to compute $\overline{\rho}(C)$ recursively. At each step one picks a place where to grow C, and one does it in all possible ways compatible with $\mathcal{V}$. If C does not embed into a species in $\mathcal{V}$, we set $\overline{\rho}(C) = 0$. Else:
- either we pick a solid binding type $B@b$ in C which we replace in all the following ways: 1) with an edge to any dual binding type in C, 2) with an edge to a new agent of type B;
- or we add a site x to an agent in C the interface of which is included in a covering class which contains x, both free and bearing a wildcard;
- or we pick a wildcard which we substitute with all the binding types compatible with the CM. The recursion stops when all terms in the sum are fragments. □

The growth procedure is not unique, as we can see in the example below. As an example, consider the first step of the recursion for the derivation of the number of embeddings of the pattern component $C = R(l^-, r^-)$. Since the set $\{l, r\}$ is a subset of two classes $\{l, r, Y48\}$, and $\{l, r, Y68\}$ one can grow C in two ways. If one grows C along $Y48$, one can express $\overline{\rho}(R(l^-, r^-))$ as the sum:

$$\overline{\rho}(R(l^-, r^-, Y48)) + \overline{\rho}(R(l^-, r^-, Y48^{SHC@p}))$$

Then the binding type $SHC@p$ needs to be expanded, because the edge is solid in the aCM, and so on. The non-uniqueness of the decomposition comes partly from the fact that coverings are not partitions.

Proposition 6.5 (subfragment)**.** *Any pattern component that occurs in the lhs of a non-trivial rule is a subfragment.*

Proof. Let C be a pattern component occurring in the lhs of a non-trivial rule. By *(1.ii)*, each $A(\sigma)$ in C has its sites contained in a class in $\mathcal{C}(A)$; and by *(2.i)*, any bond in C is solid; so by Def. 6.3, C embeds in a fragment. □

The combination of Prop. 6.4 and 6.5 ensures that the concentrations of the lhs components of all rules in $\mathcal{R}$, and hence all the rule activities, can be expressed as various functions of the concentrations of fragments.

Proposition 6.6 (left intersection)**.** *Let F be a fragment, $r = E_\ell, E_r$ be a non-trivial rule, and C be a pattern component of E_ℓ. F cannot overlap C on a site that is modified by r.*

More precisely: 1) if $X, \psi_1, \psi_2, \gamma_1, \gamma_2, Y$ is an overlap between C and F, and the image of X along ψ_1 is modified by r, then ψ_1 is an iso; and 2) if F contains a (bound) site x that can be freed by a side-effect of r (either a wildcard or agent deletion), then F also contains the site x is bound to.

Proof. 2) is a direct consequence of *(2.ii)*. Let us prove 1). By assumption, F contains an agent A with a site that is modified. By *(1.i)*, this agent A contains in F all the sites that are tested by the rule, and which therefore also occur in C. Since the overlap is a pull-back, all these sites also feature in X, and since the square of the overlap commutes,

these sites must have compatible states in F and C. By *(2.i)*, all edges in C are solid, so F must contain all the bonds emanating from A that are present in C. If we follow one, it leads us to a new agent B via a docking site x from where we can repeat our reasoning with B, using again *(1.i)*. Hence F contains a copy of C, so X is isomorphic to C. □

The combination of Prop. 6.4 and 6.6 ensures that one can express the concentration of fragments that are *consumed* by a rule as a function of the concentration of other fragments. For instance, this prevents situations such as the rule $A(x^-, y) \to \emptyset$ with $F = A(x^{B@b}, z)$, which is a good thing, since in such a case, one cannot express the rate at which r consumes F without knowing the exact correlation between the binding states of y, and z.

Proposition 6.7 (right intersection)**.** *Let F be a fragment, $r = E_\ell, E_r$ be a non-trivial rule, and $X, \psi_1, \psi_2, \gamma_1, \gamma_2, Y$ an overlap between F and E_r, where X is modified by r.*

Consider the right refinement $r' = \{r\}(Y, \gamma_2) = Y', Y$

$$\begin{array}{ccccc} X' & \xrightarrow{\psi_2'} & E_\ell & \dashrightarrow & E_r \xleftarrow{\psi_2} X \\ {\scriptstyle\psi_1'}\downarrow & & \downarrow{\scriptstyle\gamma_2'} & & {\scriptstyle\gamma_2}\downarrow \quad \downarrow{\scriptstyle\psi_1} \\ F' & \xrightarrow[\gamma_1']{} & Y' & & Y \xleftarrow[\gamma_1]{} F \end{array}$$

If E_ℓ and Y' have the same number of components, then any component in Y' is a subfragment.

Proof. Suppose F', the (possibly not connected) antecedent of F in Y', intersects some component C in E_ℓ, then it must intersect C on a modified site. To see why, call C' a component of F' which intersects C. If F' is disconnected, then C' must be modified by the rule, else F would not be connected (which it is, being a fragment); if F' is connected, then $C' = F'$, and again must be modified, else $F = F'$ and X is not modified by the rule (contrary to what we assume). In both cases, C' is modified, which means it must intersect E_ℓ on a modified site. That site must belong to C, else C' is connecting C with another component of E_ℓ in Y', which contradicts γ_2' preserving the number of components.

Edges in Y' either come from E_ℓ or from F, so by *(2.i)* and by definition of fragments, they are solid (recall we have assumed that r is non-trivial). Agents in Y' come either from E_ℓ, or F, or both (if they are merged in Y). So their sites form subsets of classes of the aCM: by *(1.ii)* in the first case; by definition of a fragment in the second case; and by *(1.i)* in the third case, since the agent in F contains a docking or a modified site (by the opening observation). It follows from the definition of fragments, that any component of Y' can be embedded in a fragment. □

Props. 6.4 and 6.7 ensure that we can express the concentrations of the fragments that are *produced* in terms of the concentrations of the other fragments.

Proposition 6.8 (cycles)**.** *If a fragment F contains two distinct and compatible binding types $A(a^{B@b})$, and $B(b^{A@a})$, then no rule can delete an a, b bond.*

Prop. 6.8 is a reformulation of *(2.iii)* which avoids a situation where by applying a rule deleting an a, b bond to an F, one will free a and b in F *simultaneously* if a, b are bound together in some concrete state. In this case, to compute the concentration of fragments produced, one would need to know the proportion of F where a, b are bound to themselves, an information which one cannot derive from the abstract state.

7 Abstract differential semantics

Using Prop. 6.4-6.8 we can now get to our main and final construction, that of our abstract/reduced semantics $\mathbb{F}^\sharp$. Following §3, we want to express, for any fragment F, $\psi(\mathbb{F}(\rho))(F)$ as a function $\mathbb{F}^\sharp$ of the $\psi(\rho)(F_i)$ where F_i are also fragments. The existence of such a function is what we have called the self-consistency of ψ in §2.

From §6.4, $\psi(\rho)(F_i) = \widetilde{\rho}(F_i)$, and by the "growth" Prop. 6.5, we see that to construct $\mathbb{F}^\sharp$, it is enough to express $\psi(\mathbb{F}(\rho))(F)$ as a function of the concentration $\widetilde{\rho}(C)$ of *subfragments* C.

Consider a rule $r = E_\ell, E_r$, with $\hat{E}_\ell = C_1, \ldots, C_n$, and C_i are components. In §5 we have computed the contributions of r to $\mathbb{F}$, by enumerating its ground refinements. Let us review quickly this construction.

7.0.1 Reformulation of the goal

We consider the set $\hat{r}$ of all triples:

$$(R_i, \phi_{R_i} : 1 \leq i \leq n), (P_j : 1 \leq j \leq m), \phi_P$$

with ϕ_i in $[C_i, R_i]$, each R_i a species in $\mathcal{V}$, $P_1, \ldots, P_m$ the sequence of species produced by the application of r along $\sum_i \phi_{R_i}$, and ϕ_P the corresponding embedding between E_r and $P_1, \ldots, P_m$.

The *negative* and *positive* contributions of r to the concentration of $S \in \mathcal{V}$ are then obtained as the respective sums:

$$\delta^-(r)(S) \quad = \quad \gamma(r) \cdot \textstyle\sum_{(R_i,\phi_{R_i}),(P_j),\phi_P \in \hat{r}} \sum_{\{k|S=R_k\}} \prod_i \rho(R_i)$$

$$\delta^+(r)(S) \quad = \quad \gamma(r) \cdot \textstyle\sum_{(R_i,\phi_{R_i}),(P_j),\phi_P \in \hat{r}} \sum_{\{k|S=P_k\}} \prod_i \rho(R_i)$$

Similarly, we consider the ways in which a fragment $F \in \mathcal{V}^\sharp$ can be embedded in a species occurring (on either side) of a ground refinement of r. Thus, we introduce the following set of 5-tuples $Neg(r, F)$ (resp. $Pos(r, F)$):

$$\begin{array}{l}(R_i, \phi_{R_i} : 1 \leq i \leq n), (P_j : 1 \leq j \leq m), \phi_P \in \hat{r}, \\ 1 \leq k \leq n \text{ (resp. } 1 \leq k \leq m), \\ \phi \in [F, R_k] \text{ (resp. } \phi \in [F, P_k])\end{array}$$

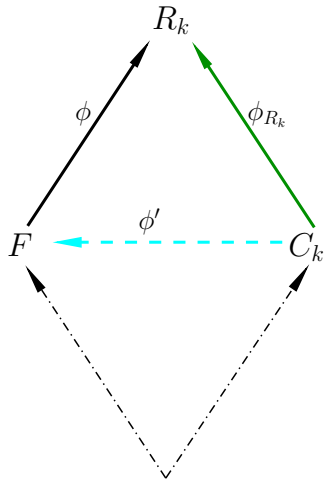


Figure 6: *Consumption and trivial rules.*

From the definition of ψ, it follows that the negative and positive contributions of r to the concentration of F are given as the respective sums:

$$\psi(\delta^-(r))(F) = \frac{\gamma(r)}{|[F,F]|} \cdot \sum_{(R_i,\phi_{R_i}),(P_j),\phi_P \in \hat{r}} \sum_{k,\phi \in [F,R_k]} \prod_i \rho(R_i)$$

$$\psi(\delta^+(r))(F) = \frac{\gamma(r)}{|[F,F]|} \cdot \sum_{(R_i,\phi_{R_i}),(P_j),\phi_P \in \hat{r}} \sum_{k,\phi \in [F,P_k]} \prod_i \rho(R_i)$$

So we can rephrase our goal as that of expressing, for each F, and each r, the difference $\psi(\delta^+(r))(F) - \psi(\delta^-(r))(F)$ in terms of (sub-) fragments.

7.0.2 Mute contributions

Pick $t = (R_i, \phi_{R_i}), (P_j), \phi_P, k, \phi$ in $Neg(r, F)$. We have a co-span ϕ_k, ϕ, R_k (see Fig. 6). If the image of F by ϕ is not modified by r, we say that t is mute.

Pick $t = (R_i, \phi_{R_i}), (P_j), \phi_P, k, \phi$ in $Pos(r, F)$. We have a co-span $\phi_{P_k}\phi$, ϕ_P, $P_1, \ldots, P_m$ (See Fig. 7), with ϕ_{P_k} the canonical inclusion of P_k into $P_1, \ldots, P_m$. If the image of F by $\phi_{P_k}\phi$ is not modified by r, we say that t is mute.

Negative and positive mute ts are in bijection, thus their contributions cancel pairwise. So, we can restrict the sums $\psi(\delta^-(r))(F)$ and $\psi(\delta^+(r))(F)$ to proper, ie non-mute, tuples in $Pos(r, F)$ and consumption ones in $Neg(r, F)$. We write $Neg'(r, F)$ and $Pos'(r, F)$ for the remaining proper contributions.

It remains to express each of these terms as functions of subfragment concentrations. Firstly, we deal with soft edges and the two forms of trivial rules, and then with non-trivial rules.

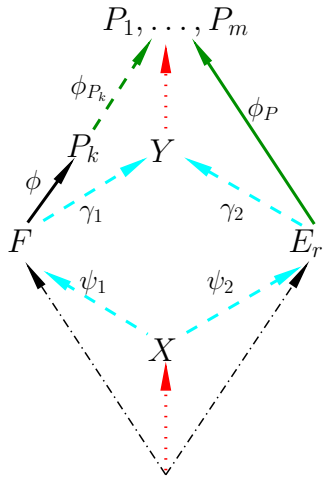


Figure 7: *Production.*

7.0.3 Trivial rules

Consider the first form of trivial dissociation (possibly $(A, a) = (B, b)$):

$$r = A(a^1), B(b^1) \to A(a), B(b)$$

and suppose the edge between a and b is soft.

Pick a term $t = (R_1, \phi_{R_1}), (P_j)_j, \phi_P, 1, \phi$ in $Neg'(r, F)$. Because t is not mute, and by Prop. 6.8, F cannot overlap with both A and B, and there is a unique embedding ϕ' between either $A(a^{B@b})$ or $B(b^{A@a})$ and F, with $\phi\phi' = \phi_{R_1}$. Conversely, if ϕ' is such an embedding, then $\phi\phi'$ is in $[C_1, R_1]$. Thus, one has a bijection between $[C_1, R_1]$ and $[A(a^{B@b}), F] \cup [B(b^{A@a}), F]$. Hence, the proper consumption of F is the sum of the following (constant) terms:

$$\frac{\gamma(r)}{|[F, F]|} \cdot \textstyle\sum_{\phi \in [F, R_1]} \rho(R_1) = \gamma(r) \cdot \widetilde{\rho}(F)$$

over ϕ' in $[A(a^{B@b}), F] \cup [B(b^{A@a}), F]$, and therefore, $\psi(\delta^-(r))(F)$ can indeed be expressed as a function of fragment concentrations (here F's concentration is enough).

Pick now a t in $Pos'(r, F)$. Similarly, by Prop. 6.8, there is a unique embedding ϕ' between $A(a)$ or $B(b)$ and F. If we denote by $F'_{\phi'}$ the antecedent of F, that is to say the fragment obtained by setting the binding state of site a to $B@b$ (in the first case) or that of site b to $A@a$ (in the second case) in the unique agent in the range of ϕ', then the proper production of F is a sum of the following terms:

$$\frac{\gamma(r)}{|[F, F]|} \cdot \overline{\rho}(F'_{\phi'})$$

over ϕ' between $[A(a), F] \cup [B(b), F]$, and therefore, $\psi(\delta^+(r))(F)$ can also be expressed as a function of fragment concentrations.

If r is the second form of trivial rule, $A(a^-) \to A(a)$, one can refine r by finding in the CM all potential bindings partners $B(b)$. Depending on whether the refining bond is soft or not, one ends up with a refinement as above, or below where we consider general rules.

7.0.4 Non-trivial rules (consumption)

Let us split first the sum that expresses the proper consumption of F by r:

$$\frac{\gamma(r)}{|[F,F]|} \cdot \sum_{(R_i,\phi_{R_i}),(P_j),\phi_P \in \hat{r}} \sum_{k,\phi \in [F,R_k]} \prod_i \rho(R_i)$$

according to k, the index of the reactant into which F embeds. Then, we can factor each summand by noticing that the set $\hat{r}$ is in bijection with the Cartesian product:

$$\prod_i \{(R_i, \phi_{R_i}) \mid \phi_{R_i} \in [C_i, R_i]\}$$

and also that for a tuple $t = (R_i, \phi_{R_i}), (P_j), \phi_P, k, \phi$, whether t is mute, only depends on k, R_k, ϕ_{R_k}, and, of course, the embedding ϕ in $[F, R_k]$ (Fig. 6).

This yields an equivalent expression for the proper consumption of the form:

$$\frac{\gamma(r)}{|[F,F]|} \cdot \sum_k \prod_i \Theta(i,k)$$

For any $i \neq k$, $\Theta(i,k)$ is the sum of $\rho(R_i)$ for each species $R_i \in \mathcal{V}$ and each embedding ϕ_i in $[C_i, R_i]$. This sum is equal to $\overline{\rho}(C_i)$. By Prop. 6.5, C_i can be embedded into a fragment, thus, by Prop. 6.4, $\overline{\rho}(C_i)$ can be expressed as a linear combination of fragment concentrations.

There remains $\Theta(k,k)$. If C_k is not modified by r, we have $\Theta(k,k) = 0$. Otherwise, $\Theta(k,k)$ is the sum of the terms $\rho(R_k)$, over co-spans ϕ_{R_k}, ϕ, R_k.

As in the first case of trivial rule, we build a bijection between the ϕ_{R_k}s such that $\phi_{R_k} \in [C_k, R_k]$ and the ϕ's such that $\phi' \in [C_k, F]$ (Fig. 6). By definition of $Neg'(r, F)$, there exists i, i' such that $\phi(i) = \phi_{R_k}(i')$ (with a site in the agent $\phi(i)$ modified by r): this defines an overlap between F and C_k, and by Prop. 6.6, there exists an embedding $\phi' \in [C_k, F]$ with $\phi_{R_k} = \phi'\phi$.

By Cor. 4.5, ϕ' is uniquely defined by R_k, F, ϕ and ϕ_{R_k}. Conversely, given ϕ and ϕ' such that $\phi \in [F, R_k]$ and $\phi' \in [C_k, F]$, we have $\phi\phi' \in [C_k, R_k]$. Thus we have the expected bijection between the ϕ_{R_k}s such that $\phi_{R_k} \in [C_k, R_k]$ and the ϕ's such that $\phi' \in [C_k, F]$.

As a consequence, $\Theta(k,k)$ is equal to the sum of the terms $\rho(R_k)$ for any R_k, ϕ and ϕ' such that $\phi \in [F, R_k]$ and $\phi' \in [C_k, F]$. Hence:

$$\begin{array}{rcl} \Theta(k,k) & = & \sum_{\phi' \in [C_k,F]} \sum_{R_k, \phi \in [F,R_k]} \rho(R_k) \\ & = & \sum_{\phi' \in [C_k,F]} \overline{\rho}(F) \end{array}$$

where the second equation comes by definition of $\overline{\rho}$.

Putting everything together, and using $|[F,F]| \cdot \widetilde{\rho}(F) = \overline{\rho}(F)$, we get that the proper consumption of F by r is the sum of the following terms:

$$\gamma(r) \cdot \widetilde{\rho}(F) \prod_{i \neq k} \overline{\rho}(C_i)$$

over k's such that the kth component C_k of r is modified by r, and over ϕ' in $[C_k, F]$. Every of these terms can be expressed as a function of fragment concentrations by Prop. 6.5.

7.0.5 Non-trivial rules (production)

Pick a production tuple:

$$t = (R_i, \phi_{R_i}), (P_j), \phi_P, k, \phi \in Pos'(r, F)$$

together with the family the canonical injection ϕ_{P_k} from P_k to $P_1, \ldots, P_m$, and the co-span $\phi_{P_k}\phi, \phi_P, P_1, \ldots, P_m$.

Since $t \in Pos'(r, F)$, there exists i, i' such that $\phi_{P_k}\phi(i) = \phi_P(i')$. Thus we have a unique overlap (up to isomorphism) $\omega(t) = X, \psi_1, \psi_2, \gamma_1, \gamma_2, Y$ between F and E_r (see Fig. 7).

We can partition $Pos'(r, F)$ and split the proper production of F according to the overlap ω between F and E_r (we shall recall that, for counting purposes, we have fixed a representative in each isomorphism class of overlaps).

This allows us to rewrite the proper production term as a sum over the overlaps ω between F and E_r of the following terms:

$$\Gamma(\omega) = \frac{\gamma(r)}{|[F,F]|} \sum_{\{t \in Pos'(r,F) | \omega(t) = \omega\}} \prod_i \rho(R_i)$$

Let us fix the overlap $\omega = X, \psi_1, \psi_2, \gamma_1, \gamma_2, Y$, and write $r' = E'_\ell, E'_r$ for the right refinement $\{r\}(Y, \gamma_2)$ of r along γ_2 (§5.1). We also write $\hat{E}'_\ell = C'_1, \ldots, C_{n'}$ as a sequence of pattern components.

If the number of non-empty connected patterns in E_ℓ and in E'_ℓ differ (ie $n \neq n'$), there is no corresponding production triple, so $\Gamma(\omega) = 0$. Likewise, if F does not overlap with E_r on a modified site, $\Gamma(\omega) = 0$.

Otherwise, by Prop. 5.2, the expression $\Gamma(\omega)$ is equal to the sum of the $\prod_i \rho(R'_i)$ for any tuple $(R'_i, \phi_{R'_i} : 1 \leq i \leq n)$ where for all i, $\phi_{R'_i} \in [C'_i, R'_i]$.

Clearly, the set of such tuples is in bijection with the Cartesian product

$$\prod_i \{(R'_i, \phi_{R'_i}) \mid \phi_{R'_i} \in [C'_i, R'_i]\}$$

so one has:

$$\begin{aligned} \Gamma(\omega) &= \frac{\gamma(r)}{|[F,F]|} \cdot \prod_{1 \leq i \leq n} \sum_{\phi_{R'_i} \in [C'_i, R'_i]} \rho(R'_i) \\ &= \frac{\gamma(r)}{|[F,F]|} \cdot \prod_{1 \leq i \leq n} \overline{\rho}(C'_i) \end{aligned}$$

Putting everything together, we get that the proper production of the fragment F is given by the sum of the expressions:

$$\frac{\gamma(r)}{|[F,F]|}\prod_i \overline{\rho}(C'_i)$$

for any overlap $X, \psi_1, \psi_2, \gamma_1, \gamma_2, Y$ between F and E_r (on a modified site), and where C'_i is the ith non-empty pattern component of the lhs of the right refinement $\{r\}(Y, \gamma_2)$. By Prop. 6.7, the pattern component C'_i can be embedded into a fragment, so by Prop. 6.4, all terms above can be expressed as linear combinations of fragment concentrations.

7.0.6 Conclusion of the construction

We have successfully expressed $\psi(\mathbb{F}(\rho))(F)$ as the sum over r of the difference $\psi(\delta^+(r))(F) - \psi(\delta^-(r))(F)$, in the sense that in all cases we could write all non-mute contributions in these terms as polynomial functions of the concentrations of fragments. Thus, we have obtained a polynomial endoapplication $\mathbb{F}^\sharp$ on the set of abstract states $\mathcal{V}^\sharp \to \mathbb{R}^+$, which is clearly continuously differentiable, and defines a differential system.

Theorem 7.1 (Fragmentation)**.** *The abstraction function ψ (defined in §6.4) and abstract counterpart $\mathbb{F}^\sharp$ (defined above) form a reduction (as defined in Th. 3.3) of the differential system $\mathbb{F}$.*

Proof. By construction, one has $\psi \circ \mathbb{F} = \mathbb{F}^\sharp \circ \psi$. Inspecting the polynomial form obtained for $\mathbb{F}^\sharp(\psi(\rho))(F)$, one sees that production terms are polynomials with positive coefficients, while consumption ones are opposite of polynomials with positive coefficients, where in addition one can always factor $\overline{\rho}(F)$. This implies the existence of the repelling functions as required in Def. 3.1. □

It is easy to verify that in the particular case where one chooses species as fragments (what we called earlier the trivial aCM), the above derivation gives exactly the concrete differential semantics.

8 Application

We have implemented a prototype of our framework in Objective Caml [48] (7,000 lines of code excluding the front-end and rule simplification). We have tested this prototype on several examples: a model of the early EGF pathway [3], two models of cross-talk between the EGF and insulin receptors (the first model, INS1, is taken from [15, table 7] whereas the second, INS2, is obtained by removing certain tests in the unbinding rule for EGF receptors), and a version of our pilot study on a larger section of the EGF pathway [3, 9, 18, 60].

We give, in Fig. 8, the number of rules, the computation time for automatic rule simplification [21], the exact number of dimensions and computation time of the concrete semantics, and the number of dimensions and computation time of the abstract semantics (which, we recall, is computed directly without precomputing the concrete one).

model	EGF	INS1	INS2	SFB
number of rules	39	76	74	69
rule simplication	0.28	0.75	0.78	0.56
concrete semantics				
number of species	356	2899	2899	$\approx 2.10^{19}$
ODE computation	2.85	27	27	$*$
abstract semantics				
number of fragments	38	208	88	$\approx 2.10^{5}$
ODE computation	0.13	0.72	0.28	871

Figure 8: *Size and computation time (in seconds) of the concrete and abstract semantics.*

Computation time also includes output generation (both for Latex and Octave [57]) that takes roughly half of the computation time. These results have been obtained on an Intel Centrino Duo, 2G RAM, 2GHz PC and show that our framework can scale to interesting pathways.

An important factor of reduction comes from the dissection of dimers. In Fig. 5(b), there are two classes $\{r, l, \mathit{Y48}\}$ $\{r, l, \mathit{Y68}\}$ for the sites of *EGFR*, and the bond between the site r and itself is soft. If we assume that p species can connect to the site *Y48* of *EGFR*, and q species can connect to the site *Y68*, there will be roughly $\frac{1}{2}((p+1)(q+1))^2$ potential dimers, which are abstracted by only $(p+q+2)$ fragments. In the model INS1, the dimerization bond is solid which leads to a less efficient reduction, since one has roughly $\frac{1}{2}(p+q)^2$ fragments for dimers.

In Fig. 9, we show the superposition of the behaviours of the EGF model in one stochastic simulation [20] and during integration of the abstract semantics. We have chosen as observables the number of proteins *SOS* that are attached to a receptor *EGFR*. The protein *SOS* can be attached to the receptor by two ways called the *short arm* and the *long arm*. The two semantics match, although only the correspondence between the concrete differential semantics and the reduced differential semantics have been explored in this paper.

9 Conclusion

We have shown a new application for abstract interpretation by using it to reduce the dimension of the (ordinary) differential semantics of rule-based models and prove that the trajectories in the reduced system are projections of the trajectories in the concrete system. In realistic examples this can make a real difference, as models with an inherently intractable concrete semantics get a much smaller abstract semantics. This means one can study, eg calibrate those models, using ODE integration which is faster than stochastic simulation. Combined with numerical approximation, our technique should extend significantly the reach of modelling in the context of large networks, where it is the most needed. Note also that the abstract/reduced semantics is likely to be more accurately related to the stochastic one, than the concrete one, as it deals with larger populations of

(smaller) objects. This prompts the remark that one should be able to extend the scope of the method to encompass infinite-dimensional differential semantics [41], for which the compressed version is nevertheless finite.

There is also scope to design more efficient approximations. To this effect, we could detect and use symmetries between sites and potentially relax certain hypotheses on fragments so as to obtain smaller ones. Another interesting avenue for further investigation is that of the relationship between the stochastic and differential semantics in agent-based models, as the ODE compression of a rule-based system could be shown directly to approximate its natural stochastic semantics.

The abstraction of the stochastic semantics cannot work directly with our approach (which was not intended for this) because, in the case of fragments with common sites, reactions that are applied to these fragments are coupled by the correlation between the states of sites in these fragments - which is exactly the information that our abstract does not detect (as discussed in §2). This issue is addressed in Ref. [32], by detecting a notion of stochastic fragment, different than the one we have used here in the deterministic case, on the states of which reactions cannot enforce correlations.

Acknowledgment

Jérôme Feret's contribution was partially supported by the AbstractCell ANR-Chair of Excellence.

References

[1] M. Abadi. Secrecy by typing in security protocols. *Journal of the ACM*, 46:611–638, 1999.

[2] C. Bashor, N. Helman, S. Yan, and W. Lim. Using engineered scaffold interactions to reshape map kinase pathway signaling dynamics. *Science*, 319(5869):1539, 2008.

[3] M. L. Blinov, J. R. Faeder, B. Goldstein, and W. S. Hlavacek. A network model of early events in epidermal growth factor receptor signaling that accounts for combinatorial complexity. *BioSystems*, 83:136–151, Jan. 2006.

[4] M. L. Blinov, J. R. Faeder, and W. S. Hlavacek. BioNetGen: software for rule-based modeling of signal transduction based on the interactions of molecular domains. *Bioinformatics*, 20:3289–3292, 2004.

[5] N. M. Borisov, A. S. Chistopolsky, J. R. Faeder, and B. N. Kholodenko. Domain-oriented reduction of rule-based network models. *IET Syst. Biol.*, 2:342–351, 2008.

[6] N. M. Borisov, N. I. Markevich, B. N. Kholodenko, and E. D. Gilles. Signaling through receptors and scaffolds: Independent interactions reduce combinatorial complexity. *Biophysical Journal*, 89:951–966, 2005.

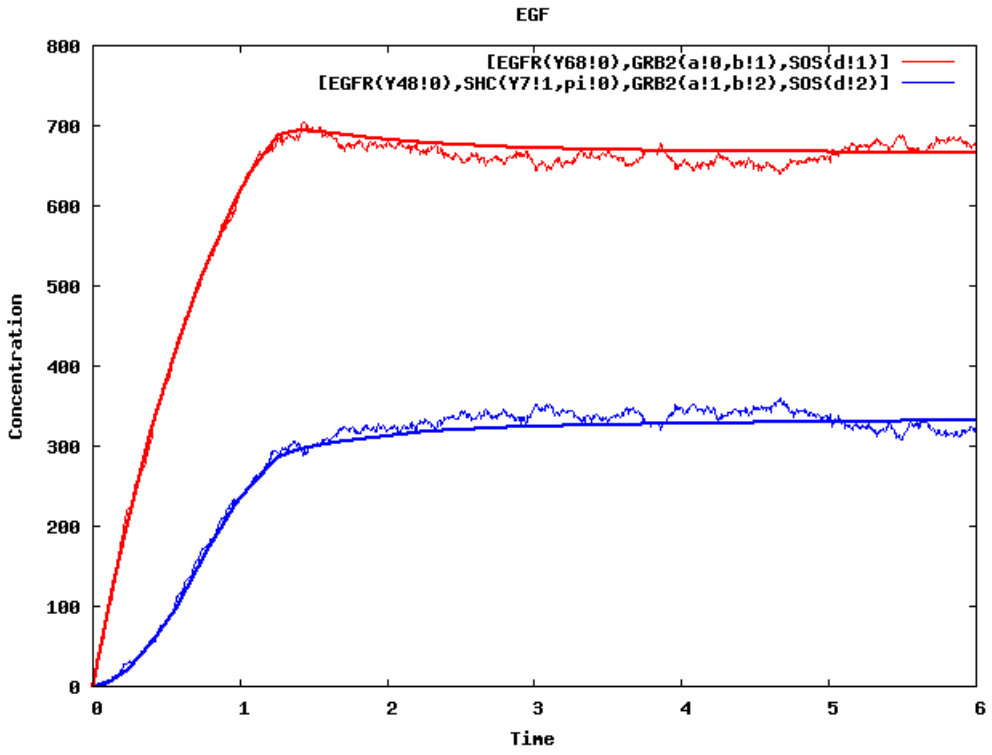


Figure 9: *Concentration of proteins SOS attached to the membrane in a stochastic simulation (wiggly curves) and in the (abstract) differential semantics, via the short arm (upper curve) and the long arm (lower curve). Units (time, concentration) and rule rate constants are arbitrary.*

[7] O. Bouissou and M. Martel. Grklib: a guaranteed Runge Kutta library. In *Proc. of SCAN '06*, page 8. IEEE Computer Society, 2006.

[8] O. Bouissou and M. Martel. Abstract interpretation of the physical inputs of embedded programs. In *Proc. of VMCAI'08*, volume 4905 of *LNCS*, pages 37–51. Springer, 2008.

[9] F. A. Brightman and D. A. Fell. Differential feedback regulation of the MAPK cascade underlies the quantitative differences in EGF and NGF signalling in PC12 cells. *FEBS Letters*, 482(3):169–174, October 2000.

[10] R. Cartwright and M. Felleisen. The semantics of program dependence. In *Proc. of PLDI'89*, pages 13–27, 1989.

[11] A. Chapoutot. *Simulation abstraite : une analyse statique de modèles Simulink*. PhD thesis, École Polytechnique, December 2008.

[12] F. Ciocchetta and J. Hillston. Bio-PEPA: A framework for the modelling and analysis of biological systems. *TCS*, 410(33-34):3065–3084, 2009.

[13] A. Coletta, R. Gori, and F. Levi. Approximating probabilistic behaviors of biological systems using abstract interpretation. *ENTCS*, 229(1):165–182, 2009.

[14] H. Conzelmann. *Mathematical Modeling of Cellular Signal Transduction Pathways — A Domain-Oriented Approach to Reduce Combinatorial Complexity*. PhD thesis, Institut für Systemdynamik des Universität Stuttgart, 2008.

[15] H. Conzelmann, D. Fey, and E. D. Gilles. Exact model reduction of combinatorial reaction networks. *BMC Systems Biology*, 2:78, 2008.

[16] H. Conzelmann, J. Saez-Rodriguez, T. Sauter, B. N. Kholodenko, and E. D. Gilles. A domain-oriented approach to the reduction of combinatorial complexity in signal transduction networks. *BMC Bioinformatics*, 7:34, 2006.

[17] P. Cousot and R. Cousot. Abstract interpretation: a unified lattice model for static analysis of programs by construction or approximation of fixpoints. In *Proc. of POPL'77*, pages 238–252. ACM Press, 1977.

[18] V. Danos, J. Feret, W. Fontana, R. Harmer, and J. Krivine. Rule-based modelling of cellular signalling. In *Proc. of CONCUR'07*, volume 4703 of *LNCS*, pages 17–41. Springer, 2007.

[19] V. Danos, J. Feret, W. Fontana, R. Harmer, and J. Krivine. Rule-based modelling of cellular signalling. *CONCUR 2007*, pages 17–41, 2007.

[20] V. Danos, J. Feret, W. Fontana, and J. Krivine. Scalable simulation of cellular signaling networks. In *Proc. of APLAS'07*, volume 4807 of *LNCS*, pages 139–157. Springer, 2007.

[21] V. Danos, J. Feret, W. Fontana, and J. Krivine. Abstract interpretation of biological signalling networks. In *Proc. of VMCAI'08*, volume 4905 of *LNCS*, pages 42–58. Springer, 2008.

[22] V. Danos and C. Laneve. Formal molecular biology. *Theoretical Computer Science*, 325(1):69–110, 2004.

[23] V. Danos and C. Laneve. Formal molecular biology. *Theoretical Computer Science*, 325(1):69–110, Sept. 2004.

[24] A. Di Pierro and H. Wiklicky. Probabilistic abstract interpretation and statistical testing. In *Proc. of PAPM-PROBMIV'02*, pages 211–212. Springer, 2002.

[25] H. Ehrig and G. Rozenberg. *Handbook of graph grammars and computing by graph transformation: Applications, languages and tools*. World Scientific Pub Co Inc, 1999.

[26] J. Faeder, M. Blinov, and W. Hlavacek. Rule-based modeling of biochemical systems with BioNetGen. *Methods Mol. Biol*, 500:113–167, 2009.

[27] J. Feret. Static analysis of digital filters. In *Proc. of ESOP'04*, volume 2986 of *LNCS*. Springer, 2004.

[28] J. Feret. *Analysis of mobile systems by abstract interpretation*. PhD thesis, École Polytechnique, 2005.

[29] J. Feret. Numerical abstract domains for digital filters, 2005. NSAD'05.

[30] J. Feret. Reachability analysis of biological signalling pathways by abstract interpretation. In *Proc. of ICCMSE'07*. American Institute of Physics conference proceedings, 2007.

[31] J. Feret, V. Danos, J. Krivine, R. Harmer, and W. Fontana. Internal coarse-graining of molecular systems. *Proc. of the National Academy of Sciences*, 106(16):6453–6458, 2009.

[32] J. Feret, H. Koeppl, and T. Petrov. Stochastic fragments: A framework for the exact reduction of the stochastic semantics of rule-based models. *International Journal of Software and Informatics*, 2010. To appear.

[33] W. Fontana. Systems biology, models, and concurrency. In *Proc. of POPL'08*, pages 1–2. ACM, 2008.

[34] R. Giacobazzi and I. Mastroeni. Non-standard semantics for program slicing. In *Special issue on Partial Evalution and Semantics-Based Program Manipulation*, pages 297–339, 2003.

[35] R. Giacobazzi and E. Quintarelli. Incompleteness, counterexamples and refinements in abstract model-checking. In *Proc. of SAS'01*, volume 2126 of *LNCS*, pages 356–373. Springer, 2001.

[36] R. Giacobazzi, F. Ranzato, and F. Scozzari. Making abstract interpretations complete. *Journal of the ACM*, 47(2):361–416, 2000.

[37] A. Girard and C. Le Guernic. Efficient reachability analysis for linear systems using support functions. In *Proc. of IFAC'08*. IFAC, 2008.

[38] A. Girard and C. Le Guernic. Zonotope/hyperplane intersection for hybrid systems reachability analysis. In *Proc. of HSCC'08*, volume 4981 of *LNCS*, pages 215–228, 2008.

[39] R. Gori and F. Levi. A new occurrence counting analysis for bioambients. In *Proc. of APLAS'05*, volume 3780 of *LNCS*, pages 381–400. Springer, 2005.

[40] R. Gori and F. Levi. An analysis for proving temporal properties of biological systems. In *Proc. of APLAS'06*, volume 4279 of *LNCS*, pages 234–252. Springer, 2006.

[41] W. L. Hart. The Cauchy-Lipschitz method for infinite systems of differential equations. *American Journal of Mathematics*, 43(4):226–231, 1921.

[42] E. L. Ince. *Ordinary Differential Equations*. Dover Publications, 1956.

[43] C. Kühn, K. Prasad, E. Klipp, and P. Gennemark. Formal Representation of the High Osmolarity Glycerol Pathway in Yeast. *Genome Informatics*, pages 22–83, 2010.

[44] T. G. Kurtz. Solutions of ordinary differential equations as limits of pure jump Markov processes. *Journal of Applied Probability*, 7:49–58, 1970.

[45] T. G. Kurtz. Limit theorems for sequences of jump Markov processes approximating ordinary differential processes. *Journal of Applied Probability*, 8:244–356, 1971.

[46] S. Lack and P. Sobocinski. Adhesive categories. In *Foundations of Software Science and Computation Structures*, pages 273–288. Springer, 2004.

[47] J. Leifer and R. Milner. Deriving Bisimulation Congruences for Reactive Systems. In *Proceedings of the 11th International Conference on Concurrency Theory*, pages 243–258. Springer-Verlag, 2000.

[48] X. Leroy, D. Doligez, J. Garrigue, D. Rémy, and J. Vouillon. The Objective Caml system, documentation and user's manual (release 3.06). Technical report, INRIA, Rocquencourt, France, 19 Aug.. 2002.

[49] D. Monniaux. Abstract interpretation of probabilistic semantics. In *Proc. of SAS'00*, volume 1824 of *LNCS*, pages 322–339. Springer Verlag, 2000.

[50] D. Monniaux. An abstract Monte-Carlo method for the analysis of probabilistic programs (extended abstract). In *Proc. of POPL'01*, pages 93–101. ACM, 2001.

[51] E. Murphy, V. Danos, J. Feret, R. Harmer, and J. Krivine. Rule based modelling and model refinement. In H. Lodhi and S. Muggleton, editors, *Elements of Computational Systems Biology*. Wiley Book Series on Bioinformatics, 2009.

[52] E. Murphy, V. Danos, J. Feret, R. Harmer, and J. Krivine. Rule based modelling and model refinement. *Elements of Computational Systems Biology. Wiley Book Series on Bioinformatics*, 2009.

[53] H. R. Nielson, F. Nielson, and H. Pilegaard. Spatial analysis of bioambients. In *Proc. SAS'04*, volume 3148 of *LNCS*, pages 69–83, 2004.

[54] T. Pawson and P. Nash. Assembly of cell regulatory systems through protein interaction domains. *Science*, 300(5618):445–52, Apr 2003.

[55] S. G. Peisajovich, J. E. Garbarino, P. Wei, and W. A. Lim. Rapid diversification of cell signaling phenotypes by modular domain recombination. *Science*, 328(5976):368–372, Apr 2010.

[56] C. Priami and P. Quaglia. Beta binders for biological interactions. *Proc. of CMSB'04*, 3082:20–33, 2004.

[57] J. B. Rawlings and J. G. Ekerd. GNU Octave. `www.octave.org`.

[58] A. Regev, E. M. Panina, W. Silverman, L. Cardelli, and E. Shapiro. Bioambients: An abstraction for biological compartments. *Theoretical Computer Science*, 2003.

[59] A. Regev, W. Silverman, and E. Shapiro. Representation and simulation of biochemical processes using the π-calculus process algebra. In *Proc. of the Pacific Symposium of Biocomputing*, pages 6:459–470, 2001.

[60] B. Schoeberl, C. Eichler-Jonsson, E. D. Gilles, and G. Müller. Computational modeling of the dynamics of the MAP kinase cascade activated by surface and internalized EGF receptors. *Nat Biotechnol*, 20(4):370–375, April 2002.

[61] B. Yeh, R. Rutigliano, A. Deb, D. Bar-Sagi, and W. Lim. Rewiring cellular morphology pathways with synthetic guanine nucleotide exchange factors. *Nature*, 447(7144):596–600, 2007.