Reachability analysis via orthogonal sets of patterns

Jérôme Feret, Kim Quyen Ly

Reachability Analysis via Orthogonal Sets of Patterns

Jérôme Feret[1]

DIENS (INRIA/ÉNS/CNRS)
PSL Research University, F-75005 Paris, France

Kim Quyên Lý[2]

DIENS (INRIA/ÉNS/CNRS)
PSL Research University, F-75005 Paris, France

Abstract

Rule-based modelling languages, such as Kappa, allow for the description of very detailed mechanistic models. Yet, as the rules become more and more numerous, there is a need for formal methods to enhance the level of confidence in the models that are described with these languages.
We develop abstract interpretation tools to capture invariants about the biochemical structure of the biomolecular species that may occur in a given model. In previous works, we have focused on the relationships between the states of the sites that belong to the same instance of a protein. This comes down to detect for a specific set of patterns, which ones may be reachable during the execution of the model.
In this paper, we generalise this approach to a broader family of abstract domains that we call orthogonal sets of patterns. More precisely, an orthogonal set of patterns is obtained by refining recursively the information about some patterns containing a given protein, so as to partition the set of occurrences of this protein in any mixture.
We show that orthogonal sets of patterns offer a convenient choice to design scalable and accurate static analyses. As an example, we use them to infer properties in models with transport of molecules (more precisely, we show that each pair of proteins that are connected, always belong to the same compartment), and models involving double bindings (we show that whenever a protein of type A is bound twice to proteins of type B, then the protein A is necessarily bound twice to the same instance of the protein B).

Keywords: Rule-based models, Abstract interpretation, Reachability analysis

1 Introduction

Mechanistic models of signalling pathways suffer from a large combinatorial complexity that is due to potential bindings between proteins and post-translational

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

[1] Email: feret@ens.fr

[2] Email: quyen@di.ens.fr

This paper is electronically published in
Electronic Notes in Theoretical Computer Science
URL: www.elsevier.com/locate/entcs

transformations of proteins. Rule-based modelling languages, such as Kappa [11] or BNGL [2], allow for the description of very detailed mechanistic models, thanks to context free rewrite rules. Yet, as the rules become more and more numerous, there is a need for formal methods to enhance the level of confidence in the models that are described with these languages.

Our classical pipeline to check the consistency of models consists in combining static analysis [9,13,15] and causality analysis [10,6]. Static analysis provides automatically and without executing the model, an over-approximation of its potential behaviour. When static analysis warns about an unexpected behaviour, causal analysis can be used to check whether this is a true alarm and, if so, to understand its origin. Then, the modeller can update the model accordingly.

In previous works, we have used abstract interpretation [4,5] to formalise static analyses that can capture the relationships between the potential states of the sites that belong to a same instance of a protein [9,13]. In this paper, we generalise this approach to a broader family of abstract domains that we call orthogonal sets of patterns. Orthogonal sets of patterns can be obtained by refining recursively the information about a given pattern so as to partition the set of occurrences of a given protein in any mixture. We show that orthogonal sets of patterns offer a convenient choice to design scalable and accurate static analyses. As an example, we use them to infer properties in models with transport of molecules (more precisely, we show that each pair of proteins that are connected, always belong to the same compartment), and models involving double bindings (we show that whenever a protein of type A is bound twice to proteins of type B, then the protein A is necessarily bound twice to the same instance of the protein B).

We have integrated our framework within an open source static analyser for Kappa models [3], and tested it against models in development. The computation time for these analyses demonstrates the scalability of our approach.

The paper is organised as follows. In Sect. 2, we provide some case studies to illustrate the goal of our analysis. In Sect. 3, we recall the semantics of Kappa. In Sect. 4, we describe a generic abstraction of bio-molecular mixtures by the means of sets of patterns and we introduce the notion of orthogonal sets of patterns. In Sect. 5, we use orthogonal sets of patterns to abstract the set of reachable mixtures in Kappa models.

2 Case studies

We introduce three toy models, so as to illustrate the kinds of properties concerning the biochemical structure of molecular species we are interested in.

2.1 A model with relationships among the sites of a protein

Firstly, we introduce a model in order to sketch how relationships among the sites of a given protein may emerge from the description of the mechanistic interactions between proteins. In this model, we consider three kinds of protein. Let us call them A, B, and C. Proteins of kind A have a single binding site that we call r. Proteins of kind B have three identified sites: two binding sites that we call l and

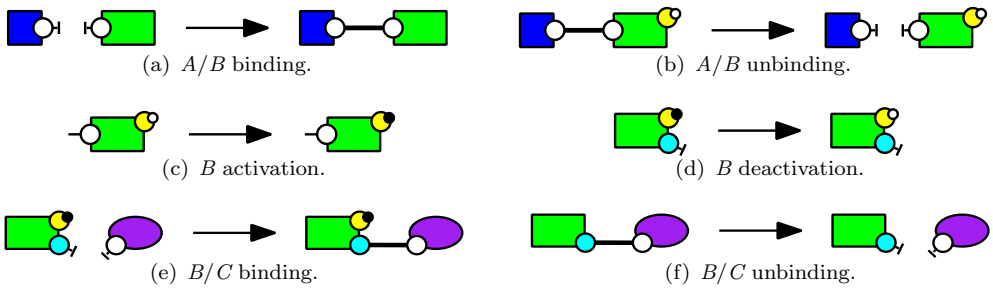


Fig. 1. Rules for a model with relationships among the sites of proteins.

r, and a phosphorylatable site that we call x. Proteins of kind C have a single binding site that we call l.

We describe the instances of these proteins graphically. Each protein of kind A is drawn as a square with its site r on the right. Each protein of kind B is drawn as a rectangle, with its site l on the left, its site r on the bottom right, and its site x on the top right. Each protein of kind C is drawn as an oval shape with its site l on the left. When the site x of a protein B is phosphorylated, we annotate it with a black smaller disk, otherwise, we annotate it with a white disk. Pairs of binding sites that are linked together are depicted thanks to an edge between these sites, whereas free sites are annotated with the symbol $\dashv$. In a rule, it is also possible to specify that a site is bound without specifying its partner site, in such a case, we annotate the site with a dandling edge.

The model is specified as a set of rewrite rules (see Fig. 1) that describe the potential biochemical interactions between the sites of the proteins. We consider the following interactions. A protein A and a protein B may bind together provided that their respective sites r and l are free (see Fig. 1(a)). When the site l of a protein B is bound, this protein may get phosphorylated (see Fig. 1(c)). Lastly, when the protein B is phosphorylated, this protein may bind with a protein C, provided that the site r of the protein B and the site l of the protein C are both free (see Fig. 1(e)). These interactions are reversible under specific conditions. We assume that a protein A and a protein B may dissociate from each other, only if the protein B is not phosphorylated (see Fig. 1(b)). In other words, the phosphorylation of a protein B by a protein A stabilises the binding between these two proteins. Moreover, we assume that a protein B may be dephosphorylated only if it is not bound to a protein C (see Fig. 1(d)). That is to say that the binding between a protein B and a protein C blocks the phosphate ion that is docked to this protein B. Lastly, a binding between a protein B and a protein C may be released unconditionally (see Fig. 1(f)).

In this model, because the phosphorylation of a protein B blocks its potential dissociation from a protein A, any protein B that is phosphorylated is necessarily bound to a protein A. In the same way, because the binding between a protein B and a protein C blocks the potential dephosphorylation of this protein B, any protein B that is bound to a protein C is necessarily phosphorylated (and hence bound to a protein A). These are the properties of interest we would like to compute automatically from the model. We notice that views analysis [13,9] can capture these properties already. Views analysis is indeed a particular case of the framework that

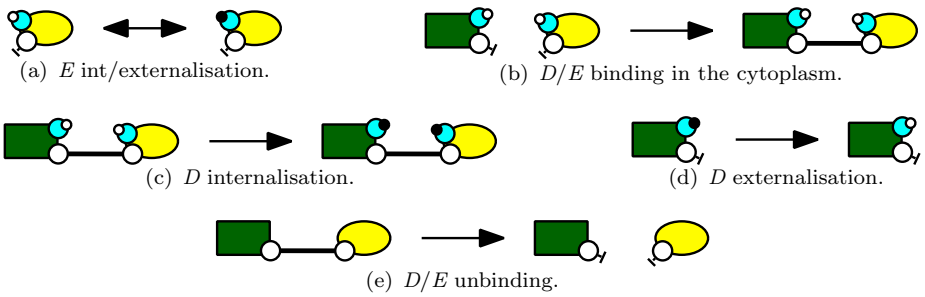


Fig. 2. Rules for a model with molecules transportation.

we are describing in this paper.

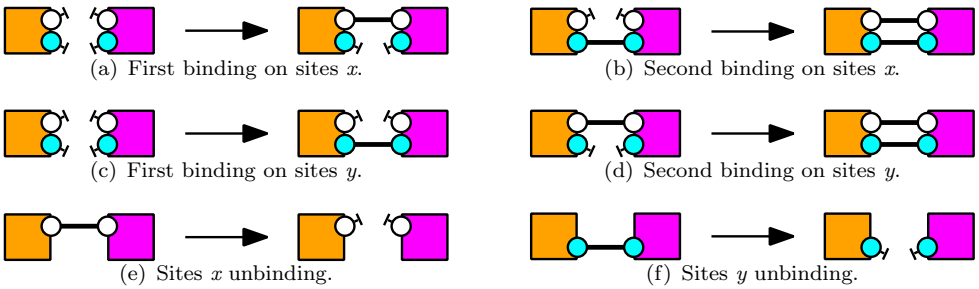
2.2 A model with transportation

Now we introduce another model to illustrate the kind of properties that may arise when proteins may be transported within a finite set of compartments. In this model, we consider two compartments: the cytoplasm and the nucleus. We also consider two kinds of protein D and E. The protein E is assumed to be a transport molecule. Its purpose is to grab a protein D in the cytoplasm, so as to take it into the nucleus. Each protein has two sites l and b. The site l is used to encode the location of the protein, whereas the site b is a binding site.

Each protein D is drawn as a rectangle. Its site l is depicted at the top right of the protein and its site b at the bottom right. Each protein E is drawn as an oval shape. Its site l is depicted at the top left of the protein and its site b at the bottom left. Furthermore, we annotate the site l of a protein with a white disk whenever this protein is in the cytoplasm, and with a black disk whenever this protein is in the nucleus.

Now we describe the rewrite rules that define the potential mechanistic interactions between the proteins of our models. These rules are depicted in Fig. 2. A protein E that is not bound to a protein D may freely move from the cytoplasm into the nucleus, and conversely (see Fig. 2(a)). Such a move is encoded as a change of the state of the site l of this protein. A protein D and a protein E may bind to each other when they are both located in the cytoplasm (see Fig. 2(b)). Then, when bound together, a protein E may take the protein D inside the nucleus (see Fig. 2(c)). We notice that, in this rule, the state of the site l of D and the state of the site l of E are replaced by a black disk simultaneously, which encodes the move of the proteins. Two proteins that are bound may dissociate from each other without any further conditions (see Fig. 2(e)). Moreover, we assume that a protein D that is no longer attached to a protein E may return into the cytoplasm (see Fig. 2(d)). We also assume that a dimer formed of a protein D and a protein E may not move from the nucleus to the cytoplasm.

In this model, the main property of interest is that when two proteins are bound together, they are necessarily located in the same compartment. This kind of properties cannot be captured by views analysis [13,9] because it concerns the state of two sites that do not belong to the same protein. The framework that we propose in this paper can capture automatically these properties, hence extending the



(a) First binding on sites x. (b) Second binding on sites x.

(c) First binding on sites y. (d) Second binding on sites y.

(e) Sites x unbinding. (f) Sites y unbinding.

Fig. 3. Rules for a model with double binding.

framework of views analysis.

2.3 A model with double binding

Our last case study involves two kinds of protein which may form a double binding between each other. Double bindings between proteins are important because they make dimers more stable.

In this model, we consider two kinds of protein F and G. Each protein has two binding sites x and y. Each protein F is drawn as a square with its site x at the top right and its site y at the bottom right. Each protein G is drawn as a square with its site x at the top left and its site y at the bottom left.

The rules for this model are given in Fig. 3. Two proteins F and G may bind their respective sites x providing that either all their sites are free (see Fig. 3(a)), or that they are already connected together via their sites y (see Fig. 3(b)). In the same way, two proteins F and G may bind their sites y providing that either all their sites are free (see Fig. 3(c)), or that they are already connected together via their sites x (see Fig. 3(d)). Moreover, any bond may be released without any condition (see Figs. 3(e) and 3(f)).

We notice that whenever both sites of a protein are bound, then they are necessarily bound to the same instance of a protein. This is the property of interest for this model. Our framework is able to capture it automatically.

3 Kappa

In this section, we recall the operational semantics of Kappa.

3.1 Site-graphs

We start by introducing site-graphs which are used to describe bio-molecular species and patterns.

Firstly we define the signature of a model.

Definition 3.1 [signature] A *signature* Σ is a tuple $(\Sigma_{ag}, \Sigma_{st}, \Sigma_{int}, \Sigma^{int}_{ag\text{-}st}, \Sigma^{lnk}_{ag\text{-}st})$ where:

(i) Σ_{ag} is a finite set of agent types;
(ii) Σ_{st} is a finite set of site identifiers;
(iii) Σ_{int} is a finite set of internal state identifiers;

(iv) $\Sigma^{int}_{ag\text{-}st} : \Sigma_{ag} \to \wp(\Sigma_{st})$ and $\Sigma^{lnk}_{ag\text{-}st} : \Sigma_{ag} \to \wp(\Sigma_{st})$ are site maps.

Agent types in Σ_{ag} denote agents of interest, as kinds of proteins for instance. A site identifier in Σ_{st} represents an identified locus for capability of interactions between agents. Internal state identifiers in Σ_{int} are special attributes which encode potential state configurations, as the phosphorylation state, the ubiquitination state, or the methylation state. Each agent type A is associated with a set of sites which may bear an internal state $\Sigma^{int}_{ag\text{-}st}(A)$ and a set of sites which may be linked $\Sigma^{lnk}_{ag\text{-}st}(A)$. We assume without any loss of generality that $\Sigma^{lnk}_{ag\text{-}st}(A) \cap \Sigma^{int}_{ag\text{-}st}(A) = \emptyset$, for every $A \in \Sigma_{ag}$ and we write $\Sigma_{ag\text{-}st}(A)$ for the set of sites $\Sigma^{lnk}_{ag\text{-}st}(A) \uplus \Sigma^{int}_{ag\text{-}st}(A)$.

Example 3.2 We give the signatures for our three case studies (*e.g.* see Sect. 2).

(i) In the first case study, the signature is the following one:
  (a) $\Sigma_{ag} \triangleq \{A, B, C\}$;
  (b) $\Sigma_{st} \triangleq \{l, r, x\}$;
  (c) $\Sigma_{int} \triangleq \{\mathrm{p}, \mathrm{u}\}$;
  (d) $\Sigma^{int}_{ag\text{-}st} \triangleq [A \mapsto \emptyset, B \mapsto \{x\}, C \mapsto \emptyset]$;
  (e) $\Sigma^{lnk}_{ag\text{-}st} \triangleq [A \mapsto \{r\}, B \mapsto \{l, r\}, C \mapsto \{l\}]$.

(ii) In the model with transportation of proteins, the signature is the following one:
  (a) $\Sigma_{ag} \triangleq \{D, E\}$;
  (b) $\Sigma_{st} \triangleq \{b, l\}$;
  (c) $\Sigma_{int} \triangleq \{\mathrm{cytoplasm}, \mathrm{nucleus}\}$;
  (d) $\Sigma^{int}_{ag\text{-}st} \triangleq [D \mapsto \{l\}, E \mapsto \{l\}]$;
  (e) $\Sigma^{lnk}_{ag\text{-}st} \triangleq [D \mapsto \{b\}, E \mapsto \{b\}]$.

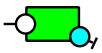
(iii) In the model with double bonds, the signature is the following one:
  (a) $\Sigma_{ag} \triangleq \{F, G\}$;
  (b) $\Sigma_{st} \triangleq \{x, y\}$;
  (c) $\Sigma_{int} \triangleq \emptyset$;
  (d) $\Sigma^{int}_{ag\text{-}st} \triangleq [F \mapsto \emptyset, G \mapsto \emptyset]$;
  (e) $\Sigma^{lnk}_{ag\text{-}st} \triangleq [F \mapsto \{x, y\}, G \mapsto \{x, y\}]$.

For the rest of the paper, we assume that we are given a signature Σ.

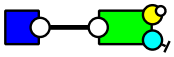
In Kappa, both the state of the system and the patterns which are used to describe transformation rules are defined as site-graphs.

Definition 3.3 [site-graph] A *site-graph* is a tuple $G = (\mathcal{A}, \mathit{type}, \mathcal{S}, \mathcal{L}, p)$ where:

(i) $\mathcal{A} \subseteq \mathbb{N}$ is a finite set of agents,
(ii) $\mathit{type} : \mathcal{A} \to \Sigma_{ag}$ is a function mapping each agent to its type,
(iii) $\mathcal{S}$ is a subset of the set $\{(n, i) \mid n \in \mathcal{A}, i \in \Sigma_{ag\text{-}st}(\mathit{type}(n))\}$,
(iv) $\mathcal{L}$ is a function from the set $\{(n, i) \in \mathcal{S} \mid i \in \Sigma^{lnk}_{ag\text{-}st}(\mathit{type}(n))\}$ to the set $\{(n, i) \in \mathcal{S} \mid i \in \Sigma^{lnk}_{ag\text{-}st}(\mathit{type}(n))\} \cup \{\dashv, -\}$, such that for every two sites $(n, i), (n', i') \in \mathcal{S}$, we have $(n', i') = \mathcal{L}(n, i)$ if and only if $(n, i) = \mathcal{L}(n', i')$;
(v) and p is a function from the set $\{(n, i) \in \mathcal{S} \mid i \in \Sigma^{int}_{ag\text{-}st}(\mathit{type}(n))\}$ to the set Σ_{int}.



(a) An agent B with the site l bound and the site r free.



(b) An agent B with the site r free and the site l bound to the site r of an agent A.

Fig. 4. Two site-graphs.

A site $(n,i) \in \mathcal{S}$ such that $i \in \Sigma^{int}_{ag\text{-}st}(type(n))$ is called a property site, whereas a site $(n,i) \in \mathcal{S}$ such that $i \in \Sigma^{lnk}_{ag\text{-}st}(type(n))$ is called a binding site. Whenever $\mathcal{L}(n,i) = \dashv$, the binding site (n,i) is free. Various levels of information may be given about the sites that are bound. Whenever $\mathcal{L}(n,i) = -$, the binding site (n,i) is bound to an unspecified site. Whenever $\mathcal{L}(n,i) = (n',i')$ (and hence $\mathcal{L}(n',i') = (n,i)$), the sites (n,i) and (n',i') are bound together.

For a site-graph G, we write as $\mathcal{A}_G$ its set of agents, $type_G$ its typing function, $\mathcal{S}_G$ its set of sites, $\mathcal{L}_G$ its set of links, and p_G the valuation of its property sites.

Example 3.4 The tuples $(\mathcal{A}_G, type_G, \mathcal{S}_G, \mathcal{L}_G, p_G)$ and $(\mathcal{A}_H, type_H, \mathcal{S}_H, \mathcal{L}_H, p_H)$, defined as follows:

(i) (a) $\mathcal{A}_G \triangleq \{1\}$;
(b) $type_G \triangleq [2 \mapsto B]$;
(c) $\mathcal{S}_G \triangleq \{(2,l),(2,r),\}$;
(d) $\mathcal{L}_G \triangleq [(1,l) \mapsto -, (1,r) \mapsto \dashv]$;
(e) $p_G \triangleq \emptyset$.

(ii) (a) $\mathcal{A}_H \triangleq \{2,3\}$;
(b) $type_H \triangleq [2 \mapsto A, 3 \mapsto B]$;
(c) $\mathcal{S}_H \triangleq \{(2,r),(3,l),(3,r),(3,x)\}$;
(d) $\mathcal{L}_H \triangleq [(2,r) \mapsto (3,l),(3,l) \mapsto (2,r),(3,r) \mapsto \dashv]$;
(e) $p_H \triangleq [(3,x) \mapsto u]$.

are two site-graphs for the signature of the first case study (*e.g.* see Fig. 2.1).

The site-graph G is depicted in Fig. 4(a) and the site-graph H is drawn in Fig. 4(b).

In Exa. 3.4, it is worth noticing that in the site-graph H, the state of every site in every protein is fully defined. The site-graph is called a chemical mixture. This is not the case for the site-graph G, since not only the state of the site l of the protein B only partially specified, but also the site x is missing. The site-graph G is called a pattern.

3.2 Relations among site-graphs

Site-graphs can be more or less specific. We introduce some materials in order to compare site-graphs according to different levels of refinement.

Two site-graphs may be related by structure-preserving functions, which are called homomorphisms.

Definition 3.5 [homomorphisms] A *homomorphism* $h : G \twoheadrightarrow H$ from a site-graph G to a site-graph H is a function of agents $h : \mathcal{A}_G \to \mathcal{A}_H$ satisfying the following

conditions:
(i) $type_G(n) = type_H(h(n))$;
(ii) if $(n, i) \in \mathcal{S}_G$, then $(h(n), i) \in \mathcal{S}_H$;
(iii) if $\mathcal{L}_G(n, i) = (n', i')$, then $\mathcal{L}_H(h(n), i) = (h(n'), i')$;
(iv) if $\mathcal{L}_G(n, i) = \dashv$, then $\mathcal{L}_H(h(n), i) = \dashv$;
(v) if $\mathcal{L}_G(n, i) = -$, then $\mathcal{L}_H(h(n), i) \in \{-\} \cup \mathcal{S}_H$;
(vi) if $p_G(n, i) = \iota$, then $p_H(h(n), i) = \iota$.

We notice that a homomorphism from a site-graph G to a site-graph H, and a homomorphism from the site-graph H to another site-graph I compose. The result of this composition is a homomorphism from the site-graph G to the site-graph I.

An embedding is a homomorphism that is induced by an injective function, defined as follows:

Definition 3.6 [embeddings] An *embedding* $h : G \rightarrow H$ from a site-graph G to a site-graph H is a function of agents $h : \mathcal{A}_G \rightarrow \mathcal{A}_H$ satisfying:
(i) h is a homomorphism from the site-graph G to the site-graph H;
(ii) $m = n$ for all m,$n \in \mathcal{A}_G$ such that $h(m) = h(n)$.

An embedding f from a site-graph G to a site-graph H, is usually denoted as $f : G \hookrightarrow H$. We also use the notation $G \hookrightarrow H$ to express the fact that there exists an embedding from the site-graphs G to H, without specifying the embedding.

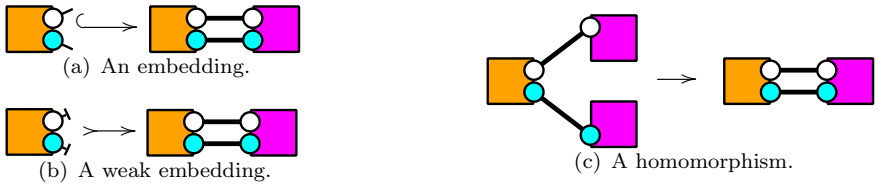
We notice that, if it is defined, the composition between two embeddings is also an embedding.

It is worth noticing that the notion of embedding between site-graphs is not the same as the notion of embedding between graphs. The major difference is that in a site-graph we have to specify explicitly when a site is free. As a consequence, a site that is free may be embedded only into a site that is free as well. It is sometimes convenient to relax the definition of embedding so as to allow sites that are free to be mapped to sites with arbitrary binding state. We introduce the notion of weak embedding for this purpose.

Definition 3.7 [weak embedding] A *weak embedding* from a site-graph G to a site-graph H is an embedding from the site-graph $\hat{G}$ to the site-graph H, where the site-graph $\hat{G}$ is defined by:
(i) $\mathcal{A}_{\hat{G}} \triangleq \mathcal{A}_G$;
(ii) $\mathcal{S}_{\hat{G}} \triangleq \{(n, i) \in \mathcal{S}_G \mid i \in \Sigma^{int}_{ag\text{-}st}(type_G(n)) \text{ or } \mathcal{L}_G(n, i) \neq \dashv\}$;
(iii) $\mathcal{L}_{\hat{G}} \triangleq [(n, i) \in \mathcal{S}'_{\hat{G}} \mapsto \mathcal{L}(n, i)]$;
(iv) $p_{\hat{G}} = p_G$, where $\mathcal{S}'_{\hat{G}} \triangleq \{\mathcal{S}_{\hat{G}} \mid i \in \Sigma^{lnk}_{ag\text{-}st}(type_G(n))\}$.
A weak embedding from a site-graph G to a site-graph H is denoted as $G \rightarrowtail H$.

We notice that every embedding is also a weak embedding.

Example 3.8 We give in Fig. 5, an example of an embedding from a site-graph to another one, an example of a weak embedding (that is not an embedding), and an example of a homomorphism (that is not an embedding).



(a) An embedding.

(b) A weak embedding.

(c) A homomorphism.

Fig. 5. An embedding, a weak embedding (which is not an embedding), and a homomorphism (which is not an embedding).

3.3 Gluing patterns

Thanks to embeddings, we can glue some patterns together along a shared region. We explain this construction as follows.

In order to glue two patterns G and H together, we need a common region $G_\cap$ that embeds inside both patterns G and H, via two embeddings that we call respectively ψ_G and ψ_H. The pair (ψ_G, ψ_H) is called a span of embeddings. Intuitively, it identifies pairs of agents which we are going to fuse in the gluing. More precisely, given $n \in \mathcal{A}_{G_\cap}$, the agent $\psi_G(n)$ in G and the agent $\psi_H(n)$ in H are going to be merged. Not all spans define a gluing: we have to check if the context of the agents that are fused together is compatible. This is achieved by requiring the existence of a site-graph $G_\cup$ such that both G and H embed into $G_\cup$, via two embeddings that we call respectively ϕ_G and ϕ_H, and that shall satisfy the constraint: $\phi_G\psi_G = \phi_H\psi_H$. In such a case, the pair (ϕ_G, ϕ_H) is called a cospan of embeddings. The site-graph $G_\cup$ is a good candidate to define the gluing of the site-graphs G and H according to the span (ψ_G, ψ_H), however we have to check that on the first hand, any site in $G_\cup$ either comes from the site-graph G, or from the site-graph H, and on the second hand that we have fused as few pairs of sites as it was imposed by the span. Roughly speaking, the cospan (ψ_G, ψ_H) should be minimal.

We give the formal definition of a gluing between two patterns, as follows:

Definition 3.9 We call a *gluing* between two site-graphs G and H, any tuple $(G_\cap, \psi_G, \psi_H, \phi_G, \phi_H, G_\cup)$ such that:

(i) $G_\cap$ and $G_\cup$ are two site-graphs;
(ii) ψ_G is an embedding from the site-graph $G_\cap$ to the site-graph G;
(iii) ψ_H is an embedding from the site-graph $G_\cap$ to the site-graph H;
(iv) ϕ_G is an embedding from the site-graph G to the site-graph $G_\cup$;
(v) ϕ_H is an embedding from the site-graph H to the site-graph $G_\cup$;
(vi) $\phi_G\psi_G = \phi_H\psi_H$;
(vii) for every triple $(\phi'_G, \phi'_H, G'_\cup)$ such that the following three conditions:
  (a) ϕ'_G is an embedding from the site-graph G to the site-graph $G'_\cup$,
  (b) ϕ'_H is an embedding from the site-graph H to the site-graph $G'_\cup$,
  (c) $\phi'_G\psi_G = \phi'_H\psi_H$,
  are satisfied, there exists a unique homomorphism h such that the following constraints:

$$\begin{cases} \phi'_G = h\phi_G, \\ \phi'_H = h\phi_H \end{cases}$$

  are both satisfied as well.

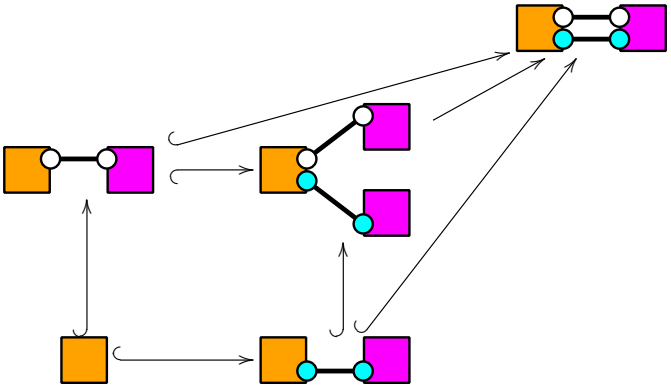
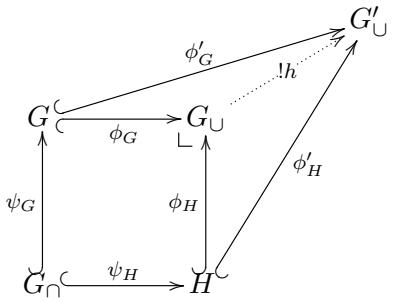


Fig. 6. An example of gluing between two site-graphs.



In Def. 3.9, the use of a homomorphism, may sound surprising. It is used not only to ensure that any information in $G_\cup$ comes either from G and H, but also to ensure that we only fuse as few pairs of agents as necessary.

Example 3.10 We give an example of a gluing between two site-graphs in Fig. 6. More precisely, we glue together two patterns, the first one is made of a protein F and a protein G bound together by their respective site x whereas the second one is made of a protein F and a protein G bound together by their respective site y, by fusing their respective proteins F. As a result, we get a site-graph made of one protein F and two proteins G, the site x of the protein F being bound to the site x of one of the protein G and the site y of the protein G being bound to the site y of the other protein G.

It could have been possible to also fuse the two instances of the protein G, but it would have led to a more specific gluing, as showed by the existence of a homomorphism from the less specific gluing to the more specific one, making the diagram commute.

3.4 Transformation between site-graphs

Rules are symbolic representation of sets of reactions between bio-molecular species. We formalise the notion of rules, and we explain how to refine rules in order to tune their level of specificity.

When a site-graph G is transformed into another site-graph H, it is important to identify which agents of G correspond to which agents of H. Since some agents of G may disappear and some agents of H may be created during the transformation, we need to formalise a partial matching between the agents of the site-graphs G

and H. This partial matching is described by the means of a span of embeddings. For the sake of generality, we define firstly the notion of weak partial embeddings.

Definition 3.11 [(weak) partial embedding] A *weak partial embedding* from a site-graph L to a site-graph R with domain D, is a pair (h_L, h_R) made of a weak embedding h_L from the site-graph D to the site-graph L and a weak embedding h_R from the site-graph D to the site-graph R.

A weak partial embedding from a site-graph L to a site-graph R with domain D is usually denoted as $\phi : L \leftarrow\!\!\prec D \rightarrowtail R$.

In a weak partial embedding $\phi : L \leftarrow\!\!\prec D \rightarrowtail R$, the site-graph L (resp. R) is called the left hand side of (resp. the right hand side of) ϕ and is written $\mathrm{lhs}(\phi)$ (resp. $\mathrm{rhs}(\phi)$). The domain D denotes a region that is shared between the site-graphs L and R. The choice of the domain can be made modulo isomorphism. That is to say that a weak partial embedding $\phi = (h_L, h_R)$ and a weak partial embedding $(h_L h, h_R h)$, where h is an isomorphism from a site-graph to the domain of ϕ are considered to be equivalent.

A weak partial embedding is called a partial embedding when each of its two weak embeddings is an embedding.

Weak partial embeddings may be composed thanks to the pullback construction.

Definition 3.12 [composition] Let ϕ and ϕ' be two weak partial embeddings such that $\mathrm{rhs}(\phi) = \mathrm{lhs}(\phi')$. We write:

$$\phi : L \xleftarrow{h_L}\!\!\prec D_1 \xrightarrow{h_R}\!\!\succ R$$

and:

$$\phi' : R \xleftarrow{h'_L}\!\!\prec D_2 \xrightarrow{h'_R}\!\!\succ R'.$$

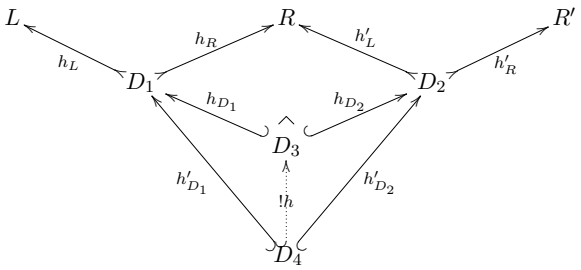
There necessarily exist a site-graph D_3 and a partial embedding:

$$\phi'' : D_1 \xleftarrow{h_{D_1}}\!\!\supset D_3 \xhookrightarrow{h_{D_2}} D_2$$

from the site-graph D_1 to the site-graph D_2, such that:

(i) $h_R h_{D_1} = h'_L h_{D_2}$;

(ii) and for any other site-graph D_4 and any partial embedding $\phi''' : D_1 \xleftarrow{h'_{D_1}}\!\!\supset D_4 \xhookrightarrow{h'_{D_2}} D_2$ such that $h_R h'_{D_1} = h'_L h'_{D_2}$, there exists a unique embedding h from D_4 to D_3 such that $h'_{D_1} = h_{D_1} h$ and $h'_{D_2} = h_{D_2} h$.



With these notations, the weak partial embedding $(h_L h_{D_1}, h'_R h'_{D_2})$ is called the *composition* of the weak partial embeddings ϕ and ϕ' and is written as $\phi'\phi$.

The composition of two weak partial embeddings is uniquely defined modulo the fact that the domain may be replaced by any isomorphic one.

A weak embedding h from a site-graph L to a site-graph R may be seen as the weak partial embedding (i_L, h). Thus, we can compose a weak partial embedding and a weak embedding (provided that the right hand side of the weak partial embedding is equal to the domain of the weak embedding). We can also compose a weak embedding and a weak partial embedding (provided that the codomain of the embedding equal to the left hand side of the weak partial embedding). We notice that the composition of two partial embeddings is also a partial embedding.

A rule is a transformation between two site-graphs, a left hand side L and a right hand side R. In a rule, some agents and some sites are preserved. This is specified by a site-graph D which is embedded both into L and into R and which describes everything that is preserved. Not all transformations are allowed: one can remove and add agents, create links between free sites, and free pairs of sites that are connected. The agents that are created have to fully define the state of their sites. Our requirements are formalised in the following definition:

Definition 3.13 A *rule* is a partial embedding $L \xleftarrow{h_L} D \xhookrightarrow{h_R} R$ such that:

(i) for any partial embedding $L \xleftarrow{h'_L} D' \xhookrightarrow{h'_R} R$ and any embedding $D \xhookrightarrow{h} D'$ such that $h_L = h'_L h$ and $h_R = h'_R h$, then h is an isomorphism;
(ii) for any site $(n, i) \in \mathcal{S}_R$, if $\mathcal{L}_R(n, i) = -$, then there exists $m \in \mathcal{A}_D$ such that $n = h_R(m)$, $(m, i) \in \mathcal{S}_D$, and $\mathcal{L}_D(m, i) = -$;
(iii) if $m \in \mathcal{A}_D$, then for any $i \in \Sigma^{lnk}_{ag\text{-}st}(type_D(m))$, $(m, i) \in \mathcal{S}_D$ if and only if $(h_L(m), i) \in \mathcal{S}_L$ if and only if $((h_R(m), i)) \in \mathcal{S}_R$;
(iv) if $m \in \mathcal{A}_R$ and $m \notin im(h_R)$, then for every site identifier $i \in \Sigma_{ag\text{-}st}(type_R(m))$, we have: $(h_R(m), i) \in \mathcal{S}_R$ and for every site identifier $i \in \Sigma^{lnk}_{ag\text{-}st}(type_R(m))$, we have: $\mathcal{L}_R((h_R(m), i), y) \in \mathcal{S}_R \cup \{\dashv\}$.

The constraint (i) ensures that D is a local greatest upper bound. The constraint (ii) ensures that when a site gets bound, we know to which site it is bound. The constraint (iii) ensures that the sites which occur both in the left hand side and in the right hand side of a rule, have a binding state in the left hand side of this rule if and only if they have a binding state in the right hand side of this rule. The constraint (iv) ensures that when an agent is created, the state of all its sites is defined.

A rule $L \hookleftarrow D \hookrightarrow R$ is usually denoted as $L \rightarrowtriangle R$ (leaving the two embeddings and the common region implicit). Moreover, we usually denote the left hand side L (resp. the right hand side R) of a rule $r : L \rightarrowtriangle R$ as $\mathrm{lhs}(r)$ (resp. as $\mathrm{rhs}(r)$).

Example 3.14 Examples of rules can be found in Figs. 1, 2, and 3.

Rules may be more or less refined [7,17], by adding more or less information about the context in which they may be applied.

Definition 3.15 [refinements] A *refinement* $(r, r', h_{L'}, h_{R'})$ is a tuple where r is a rule between two site-graphs L and R, r' is a rule between two site-graphs L' and

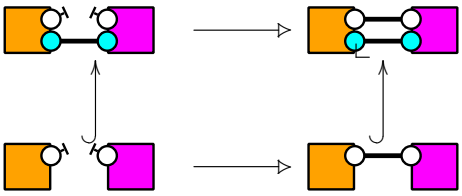
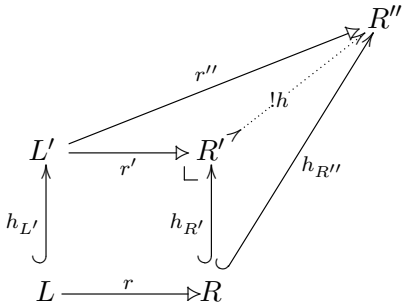


Fig. 7. A rule refinement.

R', $h_{L'}$ is an embedding from the site-graph L to the site-graph L', and $h_{R'}$ is an embedding from the site-graph R to the site-graph R' such that:

(i) $h_{R'}r = r'h_{L'}$;

(ii) and for every rule r'' from the site-graph L' to a site-graph R'', and every embedding $h_{R''}$ between the site-graph R to the site-graph R'', such that the condition $h_{R''}r = r''h_{L'}$ is satisfied, there exists a unique weak embedding h from the site-graph R' to the site-graph R'' such that both following conditions:
   (a) $r'' = hr'$;
   (b) $h_{R''} = hh_{R'}$;
   are satisfied.



The use of a homomorphism in Def. 3.15 is required in case of side effects (that is to say if, in rules, some agents are degraded without specifying the state of all their binding sites, or if some instances of the symbol − is removed) (*e.g.* see [12]).

With the notations of Def. 3.15, we say that there is a transition from the state L' to the state R' via a computation step with the label (r, h_L), and we write $L' \xrightarrow{(r,h_L)} R'$. Moreover, with the same notations, whenever the site-graph L' is a chemical mixture, the site-graph R' is a chemical mixture as well [6].

Example 3.16 We give an example of a rule refinement in Fig. 7.

Refinements can also be used to specialise a rule, for the production of a given pattern [8,12]. Given a rule $L \rightarrowtriangle R$ and an embedding f from the right hand side R of the rule r to a site-graph R', we denote by $\mathit{right\_ref}(L \rightarrowtriangle R, f)$ the set of the refinements $(r, r', h_{L'}, h_{R'})$ such that both $r = L \rightarrowtriangle R$ and $h_{R'} = f$. If the rule $L \rightarrowtriangle R$ induces no side-effects, then the set $\mathit{right\_ref}(L \rightarrowtriangle R, f)$ is singleton. Otherwise, it may contains several elements, according to which sites of the site-graph R' might have been released by side-effects (*e.g.* see [12], for an operational definition of $\mathit{right\_ref}(L \rightarrowtriangle R, f)$).

4 Orthogonal sets of patterns

Now that we have explained the operational semantics of Kappa. We introduce orthogonal sets of patterns as a means to abstract sets of chemical mixtures.

We denote as $\mathcal{M}$ the set of all the chemical mixtures that can be written with the signature Σ. We assume that we are given $\mathcal{P}$ a set of patterns of interest. The goal of our analysis is to detect which patterns of interest may occur in a mixture that is reachable from a given set of initial mixtures, after having applied zero, one, or several transition steps. The main idea is to abstract a set of chemical mixtures $X \subseteq \mathcal{M}$ by the subset Y of the patterns in $\mathcal{P}$ that may occur in at least one of the chemical mixtures in the set X. Thus, we define an abstraction function $\alpha_{\mathcal{P}}$ from the set $\wp(\mathcal{M})$ to the set $\wp(\mathcal{P})$, mapping each set $X \subseteq \mathcal{M}$ to the set $\{P \in \mathcal{P} \mid \exists M \in X, P \hookrightarrow M\}$. Intuitively, a subset $Y \subseteq \mathcal{P}$ denotes the set $X \subseteq \mathcal{M}$ of the mixtures such that any pattern in $\mathcal{P}$ that occurs in a mixture $M \in X$, belongs to the set Y. This relation may be formalised by the means of a concretisation function: we define the concretisation function $\gamma_{\mathcal{P}}$ from the set $\wp(\mathcal{P})$ to the set $\wp(\mathcal{M})$, that maps any subset $Y \subseteq \mathcal{P}$ to the set $\{M \in \mathcal{M} \mid \forall P \in \mathcal{P},\ P \hookrightarrow M \Rightarrow P \in Y\}$.

The functions $\alpha_{\mathcal{P}}$ and $\gamma_{\mathcal{P}}$ satisfy the following property: for any set $X \subseteq \mathcal{M}$ and any set $Y \subseteq \mathcal{P}$, $\alpha_{\mathcal{P}}(X) \subseteq Y$ if and only if $X \subseteq \gamma_{\mathcal{P}}(Y)$. Thus, the pair of functions $(\alpha_{\mathcal{P}}, \gamma_{\mathcal{P}})$ forms a Galois connexion [4,5] between the complete lattice $\wp(\mathcal{M})$ and the complete lattice $\wp(\mathcal{P})$.

In order to apply a rule in a specific context, we have to check whether it is possible to reach a mixture that contains an occurrence of the left hand side of the refinement of this rule, that specialises this rule to this particular context. This comes down to check whether a given pattern occurs in a reachable mixture. Thus, we propose to use our abstraction to answer to this question approximately.

This is the purpose of the following definition.

Definition 4.1 [abstract satisfaction] Given a subset $Y \subseteq \mathcal{P}$ of patterns of interest, and P a pattern, we write $Y \models_{\mathcal{P}} P$ if and only if there exists a mixture $M \in \gamma_{\mathcal{P}}(Y)$ such that $P \hookrightarrow M$.

It is worth noticing that the choice of the set $\mathcal{P}$ of the patterns of interest is crucial. It impacts the accuracy of our abstraction, that is to say our capability to express the fact that a given pattern is reachable, or not. But it also impacts the efficiency of our analysis: according to the set of patterns of interest, it may be more or less costly to compute whether the relation $Y \models_{\mathcal{P}} P$ is satisfied for a given subset Y of patterns of interest and for a given pattern P. We address this issue by restricting the set of potential sets of patterns of interest, in order to provide them with a more convenient algebraic structure. We also approximate even further the computation of the relation $\models_{\mathcal{P}}$.

We propose to use orthogonal sets of patterns. Orthogonal sets of patterns have been introduced in [14]. They allow us to partition sets of embeddings from a given pattern to arbitrary site-graphs, according to some contextual information. We give as follows the abstract definition of an orthogonal set of patterns. As an abuse of notation, for each kind of protein $A \in \Sigma_{ag}$, we denote by A the site-graph that has only one agent of type A, and no site.

Fig. 8. Orthogonal set of patterns for the protein B.

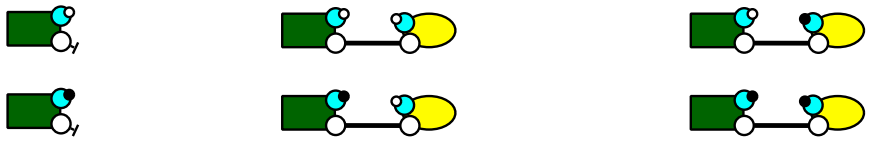


Fig. 9. Orthogonal set of patterns for the protein D.

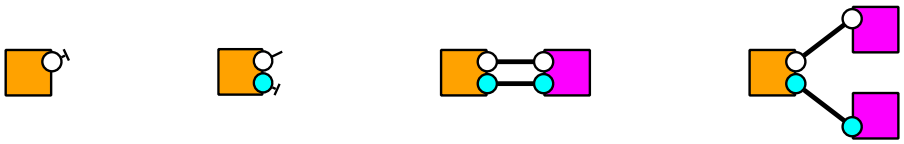


Fig. 10. Orthogonal set of patterns for the protein F.

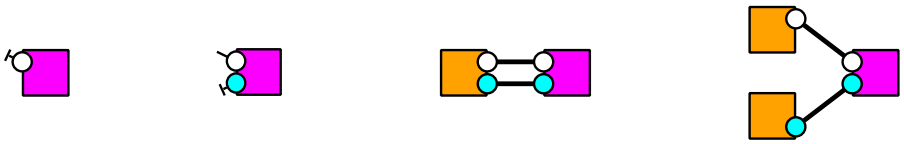


Fig. 11. Orthogonal set of patterns for the protein G.

Definition 4.2 [orthogonal set of patterns] Let A be an agent type in Σ_{ag}. An orthogonal set of patterns for the agent type A is a set $\mathcal{O}$ of pairs (P, f) where P is a site-graph and f is an embedding from the site-graph A to the site-graph P, such that for any embedding f'' from the site-graph A to a mixture $M \in \mathcal{M}$, there exists a unique pair $(P, f) \in \mathcal{O}$ and a unique embedding $f' : P \hookrightarrow M$, satisfying $f'' = f'f$.

Example 4.3 We now give examples of orthogonal sets of patterns. In these example, we keep the embeddings implicit, since each pattern contains only one instance of the protein being refined. In Fig. 8, we give an example of orthogonal set of patterns for the protein B of the first case study (see Fig. 1). In Fig. 9, we give an example of orthogonal set of patterns for the protein D of the second case study (see Fig. 2). In Figs. 10 and 11, we give examples of orthogonal set of patterns for the proteins F and G of the third case study (see Fig. 3). There is no need to provide a set of patterns for the protein E in the second case study since the fact that the proteins D and E are located in the same compartment is equivalent to the fact that the proteins E and D are located in the same compartment. This is not the case in the third case study, since, indeed, the fact that each instance of the protein F that is bound twice is bound to the same instance of the protein G does not imply that each instance of the protein G that is bound twice is bound to the same instance of the protein F.

Finite orthogonal sets of patterns for a given agent A can be defined inductively. We start with the set that contains only the pair made of the site-graph A and

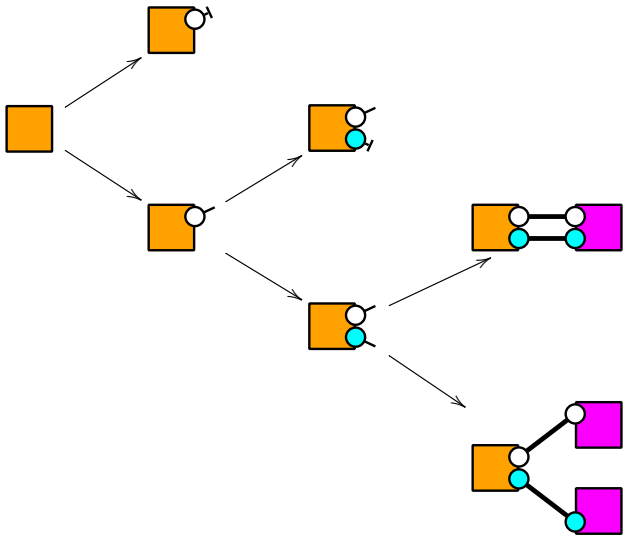


Fig. 12. Derivation of an orthogonal set of patterns for the protein F.

the unique embedding from the site-graph A to itself. Then, an induction step, consists in refining a given pattern by considering any possibility for the state of an unspecified site. More precisely, we consider the following three kinds of orthogonal refinements:

(i) one can enumerate all the potential internal states a missing property site may take;
(ii) one can consider whether a missing binding site is free, or bound (without specifying to which site it is bound to);
(iii) one can refine the binding state of a bound site and enumerate all the potential sites it may be bound to, whether these sites belong to an agent already in the pattern, or in a fresh agent.

In order to keep the size of the orthogonal sets of patterns small, we guide the refinements by the means of simple typing information about the set of states each property site may take, and the set of type of sites that may be bound together.

Example 4.4 We explain how the orthogonal set of patterns for the protein F can be obtained inductively (see Fig. 12). We start with the site-graph F. Then, we discuss about the state of the site x (top right of the protein): this site is either free, or bound. Thus we get two patterns: the first pattern of Fig. 10 and another one. We keep on by refining the second one. We discuss about the state of the site y (bottom right of the protein): this site can be either free or bound. In the case it is free, we obtain the second pattern of Fig. 10. We keep on by refining the other pattern. In this pattern, the site x is bound. Thus, we can specify that this site is bound to the site x of a fresh protein G (we have used typing information not to consider bounds to other sites). Then, we keep on refining this pattern by specifying to which kinds of site the site y is bound to. There are two possibilities: either the site y is bound to the site y of the same protein as the one that is bound to the site x, or the site y is bound to the site y of a fresh protein G.

At this stage, we get the orthogonal set of patterns that is given in Fig. 10.

Now we define our relaxed abstract satisfaction procedure in the case where the abstract domain is an orthogonal set of patterns.

Definition 4.5 [over-approximated abstract satisfaction] Let A be an agent type in Σ_{ag} and $\mathcal{O}$ be an orthogonal set of patterns for the agent type A. Let Y be a subset of $\mathcal{O}$. Let P be a pattern.

We write $Y \models^{\sharp}_{\mathcal{O}} P$, if and only if, for any agent $n \in \mathcal{A}_P$, such that $type_P(n) = A$, there exists a pair $(P', f') \in Y$ and a gluing $(G_\cap, \psi, \psi', \phi, \phi', G_\cup)$ between the site-graph P and the site-graph P' such that the following conditions are satisfied:

(i) $G_\cap = A$;
(ii) ψ maps the unique agent of the site-graph A to the agent n in the pattern P;
(iii) $\psi' = f'$.

Let us give some intuition about the procedure in Def. 4.5. Since an orthogonal set of patterns for a protein A defines a partition of the potential context for this protein, the idea is to check for each instance of the protein A in the pattern P, that its context is compatible with at least one authorised pattern $P' \in Y$. The compatibility is ensured by the existence of a gluing between the two patterns P and P'.

The relation $\models^{\sharp}$ is a conservative approximation of the relation $\models$ as formalised in Prop. 4.6. Given $\mathcal{O}$ an orthogonal set of patterns and Y a subset of $\mathcal{O}$, we denote as $\hat{Y}$ the set of patterns $\{P \mid \exists f, (P, f) \in Y\}$.

Proposition 4.6 *Let A be an agent type in Σ_{ag} and $\mathcal{O}$ be an orthogonal refinement for the agent type A. Let Y be a subset of $\mathcal{O}$. Let P be a pattern.*

The following implication is satisfied:

$$Y \not\models^{\sharp}_{\mathcal{O}} P \Rightarrow \hat{Y} \not\models_{\hat{\mathcal{O}}} P.$$

5 Reachability analysis

In this section, we use orthogonal sets of patterns to abstract the set of reachable mixtures in a given Kappa model.

We assume that we are given a set $X_0 \subseteq \mathcal{M}$ of initial mixtures and a set of rules $\mathcal{R}$. We are interested in the set of mixtures that can be obtained starting from an initial one, after zero, one, or several transition steps using the rules in the set $\mathcal{R}$. The set of reachable mixtures can also be defined as the least fix-point of the monotonic operator $\mathbb{F}$ over sets of mixtures, that is defined as follows:

$$\mathbb{F} : \begin{cases} \wp(\mathcal{M}) \to \wp(\mathcal{M}) \\ X \quad \to X_0 \cup \left\{ M' \;\middle|\; \begin{array}{l} \exists M \in X,\ r \in \mathcal{R},\ h : \mathrm{lhs}(r) \hookrightarrow M, \\ \text{such that } M \xrightarrow{(r,h)} M' \end{array} \right\}. \end{cases}$$

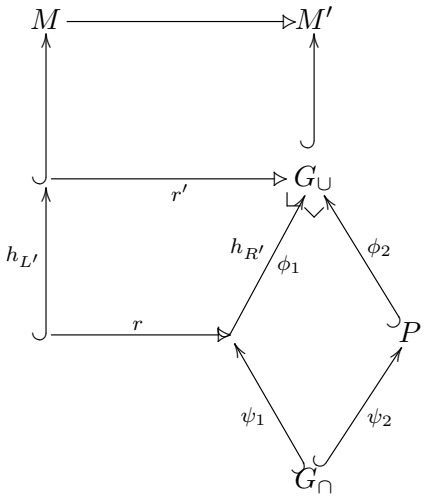
The operator $\mathbb{F}$ is monotonic map over a complete lattice. Thus, by [18], it admits a least fix-point. The least fix-point $lfp\ \mathbb{F}$ of $\mathbb{F}$ is indeed the set of reachable mixtures.

Yet, the computation of this fix-point can be costly, or could even not terminate in case of polymerisation. Thus we are going to abstract this computation, by the

means of a collection of orthogonal sets of patterns. Formally, we assume that we are given a set $\mathcal{D}$ of orthogonal sets of pattern for the agent type A. We denote as $\mathcal{P}$ the set of all the patterns that occurs in these sets, that is to say that $\mathcal{P} \triangleq \bigcup\{\hat{\mathcal{O}} \mid \mathcal{O} \in \mathcal{D}\}$. The set $\mathcal{P}$ contains the patterns of interest. We are going to abstract sets of mixtures, by subsets of the set $\mathcal{P}$. The set $\wp(\mathcal{M})$ and the set $\wp(\mathcal{P})$ are related by the Galois connexion $(\alpha_{\mathcal{P}}, \gamma_{\mathcal{P}})$ that we have introduced in Sect. 4.

The computation of the least fix-point of the function $\mathbb{F}$ can be done in the abstract, by the means of the Galois connexion $(\alpha_{\mathcal{P}}, \gamma_{\mathcal{P}})$. By [5], the function $\alpha_{\mathcal{P}} \circ \mathbb{F} \circ \gamma_{\mathcal{P}}$ is the best abstract counterpart to the function $\mathbb{F}$. In particular, the function $\alpha_{\mathcal{P}} \circ \mathbb{F} \circ \gamma_{\mathcal{P}}$ is monotonic and its least fix-point satisfies the inclusion: $\mathit{lfp}\, \mathbb{F} \subseteq \gamma_{\mathcal{P}}(\mathit{lfp}\, \alpha_{\mathcal{P}} \circ \mathbb{F} \circ \gamma_{\mathcal{P}})$.

Now we derive an explicit definition of the function $\alpha_{\mathcal{P}} \circ \mathbb{F} \circ \gamma_{\mathcal{P}}$. When applied to a subset $Y \subseteq \mathcal{P}$ of patterns, the function $\alpha_{\mathcal{P}} \circ \mathbb{F} \circ \gamma_{\mathcal{P}}$ computes the set of patterns in $\mathcal{P}$ that can occur in a mixture M' that is reachable in one step of computation from a mixture $M \in \gamma_{\mathcal{P}}(Y)$, and adds them to the set Y. For each pattern $P \in [\alpha_{\mathcal{P}} \circ \mathbb{F} \circ \gamma_{\mathcal{P}}](Y) \setminus Y$, there exists necessarily a refinement r' of a given rule r, such that the right hand side of the rule r' is a gluing between the pattern P and the right hand side of the original rule r.



It follows that the following proposition holds.

Proposition 5.1 *Let Y be a subset of the set $\mathcal{P}$. The set $[\alpha_{\mathcal{P}} \circ \mathbb{F} \circ \gamma_{\mathcal{P}}](Y)$ is exactly the set of the patterns $P \in \mathcal{P}$ such that at least one of the following conditions is satisfied:*

(i) *$P \in Y$,*

(ii) *there exist a rule $r \in \mathcal{R}$, a glueing $(G_\cap, \psi_1, \psi_2, \phi_1, \phi_2, G_\cup)$ between the right hand side $\mathit{rhs}(r)$ of the rule r and the pattern P, and a right refinement $(r, r', h_{L'}, h_{R'}) \in \mathit{right\_ref}(r, \phi_1)$ of the rule r along the embedding ϕ_1 such that: $Y \models_{\mathcal{P}} \mathit{lhs}(r')$.*

The computation of the least fix-point of the function $\alpha_{\mathcal{P}} \circ \mathbb{F} \circ \gamma_{\mathcal{P}}$ can be costly because of the cost of deciding whether the relation $\models_{\mathcal{P}}$ is satisfied, or not. We propose this decision procedure by a more abstract one, using the fact that we use a collection of orthogonal sets of patterns.

Fig. 13. Analysis result for the protein B.

We define the function $\mathbb{F}^\sharp$ as follows:

$$\mathbb{F}^\sharp : \left\{ \begin{array}{l} \wp(\mathcal{P}) \to \wp(\mathcal{P}) \\ \\ Y \quad \mapsto Y \cup \left\{ P \in \mathcal{P} \;\middle|\; \begin{array}{l} \exists r \in \mathcal{R} \\ \exists (G_\cap, \psi_1, \psi_2, \phi_1, \phi_2, G_\cup) \text{ a gluing between} \\ \qquad \mathrm{rhs}(r) \text{ and } P \\ \exists (r, r', h_{L'}, h_{R'}) \in \mathit{right\_ref}(r, \phi_1) \text{ such that:} \\ \qquad \forall \mathcal{O} \in \mathcal{D},\ \{(P, f) \in \mathcal{O} \mid P \in Y\} \models^\sharp_{\mathcal{O}} \mathrm{lhs}(r') \end{array} \right\} \end{array} \right.$$

The function $\mathbb{F}^\sharp$ is monotonic. Thus, by [18], it has a least fix-point. Moreover, the function $\mathbb{F}^\sharp$ satisfies the inclusion $\alpha_\mathcal{P} \circ \mathbb{F} \circ \gamma_\mathcal{P}(Y) \subseteq \mathbb{F}^\sharp(Y)$, for any subset $Y \subseteq \mathcal{P}$. By [4,5], it follows that $\mathit{lfp}\ \ \alpha_\mathcal{P} \circ \mathbb{F} \circ \gamma_\mathcal{P} \subseteq \mathit{lfp}\ \ \mathbb{F}^\sharp$, which ensures that $\mathit{lfp}\ \ \mathbb{F} \subseteq \gamma_\mathcal{P}(\mathit{lfp}\ \ \mathbb{F}^\sharp)$.

Example 5.2 We give in Fig. 13, the result of the fix-point iteration of the function $\mathbb{F}^\sharp$ on the model of Fig. 1, starting with an arbitrary amount of the proteins A, B, and C, with all their binding sites free, and none of their property sites phosphorylated. We have used as an abstract domain the orthogonal set of patterns that is given in Fig. 8. Our analysis has succeeded in proving that whenever the protein B is phosphorylated then its site l is bound, and that whenever its site r is bound, then it is phosphorylated. Even it our analysis is approximated, this is a definite answer: if a pattern is found by the analysis, it may be not reachable; but if a pattern does not occur in the result of the analysis, then it cannot occur in any reachable mixtures.

We give in Fig. 14, the result of the fix-point iteration of the function $\mathbb{F}^\sharp$ on the model of Fig. 2, starting with an arbitrary amount of the proteins D and E, with all their binding sites free, and all located in the cytoplasm. We have used as an abstract domain the orthogonal set of patterns that is given in Fig. 9. Our analysis has succeeded in proving that whenever two proteins D and E form a dimer, then their site l takes the same value, that is to say that they are located in the same compartment.

We give in Figs. 15 and 16, the result of the fix-point iteration of the function $\mathbb{F}^\sharp$ on the model of Fig. 3, starting with an arbitrary amount of proteins F and G, with all their binding sites free. We have used as an abstract domain the orthogonal set of patterns that is given in Figs. 10 and 11. Our analysis has succeeded in proving that whenever a protein has its two binding sites occupied, then it is necessarily bound twice to the same instance of a protein.

Our framework is integrated within an open source static analyser for Kappa models [3]. A pre-analysis is used to select the set of orthogonal sets of patterns of interest. Basically, we focus on three kinds of properties:

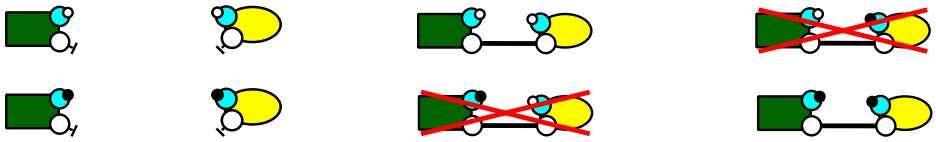


Fig. 14. Analysis result for the protein D.

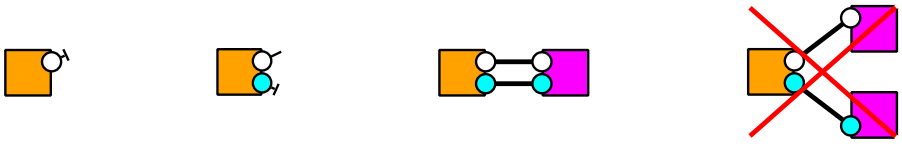


Fig. 15. Analysis result for the protein F.

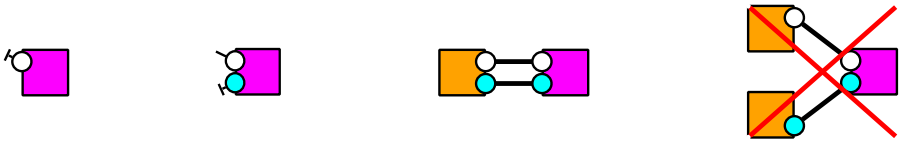


Fig. 16. Analysis result for the protein G.

model	number of rules	analysis time
EGFR	20	21 ms
TGFβ	102	660 ms
WNT	1390	12 418 ms

Fig. 17. Benchmarks: The analyses have been performed on a Dell Lattitude E6430s with 8Gb of memory with a four core Intel Core i7-3540M CPU @ 3.00GHz processor.

(i) we are interested in the relationships among the states of each set of sites that occur together in a given instance of a protein in a given rule;
(ii) we are interested in the relationships between the state of each pair of sites that occur in two instances of proteins bound together in the right hand side of a given rule;
(iii) we are interested, for each pair of proteins that can be bound together in several ways, whether one instance of one of this protein may be bound to two instances of the other one simultaneously.

We have tested our analyser on three models of various sizes. The first model describes the early events in the integration of the epidermic growth factor. It is inspired by the model that is described in [1]. The second and the third models are in development. The second model is a model of the extra-cellular matrix of the TGFβ by Nathalie Théret and Jean Cocquet, and the third one is a model of the Wnt signalling system by Héctor Francisco Medina Abarca. In the first model, the analysis has detected all the relational information about the states of sites in each protein of the model. It has also proved that a given instance of a receptor cannot be bound to two different instances of a receptor simultaneously. For the other models, the result of the analysis is used by the modellers to check the consistency of their model after each update, as well as to identify what is missing in the current version of their models.

6 Conclusion

In this paper, we have developed an abstract interpretation static analysis to capture invariants about the biochemical structure of bio-molecular species that may be reachable in a given model. Our framework extends our previous works in local views [9,13]. Our analysis not only can infer relationships among the state of the sites of a same protein, but also it can detect relationships between the sites of several proteins of a given molecular species.

We have provided a generic analysis that, given a set of patterns of interest, detects which one may not arise when executing the model. So as to ensure the scalability of our approach we have specialised our framework to the case of orthogonal sets of patterns that can be used to partition the set of the occurrences of a given agent according to some contextual information. Orthogonal sets of patterns offer a nice trade-off between accuracy, efficiency, and expressiveness. We have used them not only to reformulate our previous analysis for local views, but also to detect properties related to the transport of molecules and to the formation of double bonds between proteins. We have shown the scalability of our approach on two models in development.

For future works, we would like to use weakly relation domains [16], to infer the kind of non local properties that are involved in the formation of macro-molecules. More precisely, we would like an analysis that computes the pair of proteins conformations that may occur simultaneously in a given molecular species.

References

[1] Blinov, M., J. Faeder, B. Goldstein and W. Hlavacek, *A network model of early events in epidermal growth factor receptor signaling that accounts for combinatorial complexity*, Bio Systems **83** (2006).

[2] Blinov, M. L., J. R. Faeder, B. Goldstein and W. S. Hlavacek, *Bionetgen: software for rule-based modeling of signal transduction based on the interactions of molecular domains*, Bioinformatics **20** (2004).

[3] Boutillier, P., J. Feret, J. Krivine and Kim Lý. Q., *Kasim development homepage*.
URL http://kappalanguage.org

[4] Cousot, P. and R. Cousot, *Abstract interpretation: A unified lattice model for static analysis of programs by construction or approximation of fixpoints*, in: *Proc. POPL'77*, 1977.

[5] Cousot, P. and R. Cousot, *Systematic design of program analysis framework*, in: *Proc. POPL'79*, 1979.

[6] Danos, V., J. Feret, W. Fontana, R. Harmer, J. Hayman, J. Krivine, C. D. Thompson-Walsh and G. Winskel, *Graphs, rewriting and pathway reconstruction for rule-based models*, in: *Proc. FSTTCS '12*, LIPIcs **18**, 2012.

[7] Danos, V., J. Feret, W. Fontana, R. Harmer and J. Krivine, *Rule-based modelling, symmetries, refinements*, in: J. Fisher, editor, *Proc. FMSB'08*, LNCS (LNBI) **5054**, 2008.

[8] Danos, V., J. Feret, W. Fontana, R. Harmer and J. Krivine, *Abstracting the differential semantics of rule-based models: exact and automated model reduction*, in: *Proceedings of the Twenty-Fifth Annual IEEE Symposium on Logic in Computer Science, LICS '2010* (2010).

[9] Danos, V., J. Feret, W. Fontana and J. Krivine, *Abstract interpretation of cellular signalling networks*, in: *Proc. VMCAI'08*, LNCS **4905**, 2008.

[10] Danos, V., J. Feret, W. Fontanta, R. Harmer and J. Krivine, *Rule based modeling of biological signaling*, in: *Proc. CONCUR'07*, LNCS **4703**, 2007.

[11] Danos, V. and C. Laneve, *Formal molecular biology*, TCS **325** (2004).

[12] Feret, J., *Abstract interpretations of rule-based modelling*, Habilitation à Diriger des Recherches, in preparation.

[13] Feret, J., *Reachability analysis of biological signalling pathways by abstract interpretation*, in: *Proc. ICCMSE'07*, AIP **963**, 2007.

[14] Feret, J., V. Danos, J. Krivine, R. Harmer and W. Fontana, *Internal coarse-graining of molecular systems*, PNAS (2009).

[15] Feret, J. and Q. K. Lý, *Local traces: an over-approximation of the behaviour of the proteins in rule-based models*, in: *Proc. CMSB'16*, LNCS (2016), to appear.

[16] Miné, A., *A few graph-based relational numerical abstract domains*, in: *Proc. SAS'02*, LNCS **2477** (2002).

[17] Murphy, E., V. Danos, J. Feret, J. Krivine and R. Harmer, "Elements of Computational Systems Biology," Wiley Book Series on Bioinformatics, John Wiley & Sons, Inc., 2010 .

[18] Tarski, A., *A lattice-theoretical fixpoint theorem and its applications.*, Pacific J. Math. **5** (1955).