

Local traces: an over-approximation of the behaviour of the proteins in rule-based models

Jérôme Feret, Kim Quyen Ly

► **To cite this version:**

Jérôme Feret, Kim Quyen Ly. Local traces: an over-approximation of the behaviour of the proteins in rule-based models. E. Bartocci; P. Lio'; N. Paoletti. CMSB 2016 - Fourteenth Conference on Computational Method in Systems Biology, Sep 2016, Cambridge, United Kingdom. Springer, LNCS/LNBI, 9859, pp.116-131, 2016, Computational Methods in Systems Biology. <<https://www.cl.cam.ac.uk/events/cmsb2016>>. <10.1007/978-3-319-45177-0_8>. <hal-01379897>

HAL Id: hal-01379897

<https://hal.inria.fr/hal-01379897>

Submitted on 14 Oct 2016

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Local traces: an over-approximation of the behaviour of the proteins in rule-based models^{*}

Jérôme Feret¹ and Kim Lý Quyên¹

DI-ENS (INRIA/ÉNS/CNRS/PSL^{*}), Paris, France
feret@ens.fr, quyen@di.ens.fr

Abstract. Thanks to rule-based modelling languages, we can assemble large sets of mechanistic protein-protein interactions within integrated models. Our goal would be to understand how the behaviour of these systems emerges from these low-level interactions. Yet this is a quite long term challenge and it is desirable to offer intermediary levels of abstraction, so as to get a better understanding of the models and to increase our confidence within our mechanistic assumptions.

In this paper, we propose an abstract interpretation of the behaviour of each protein, in isolation. Given a model written in Kappa, this abstraction computes for each kind of protein a transition system that describes which conformations this protein can take and how a protein can pass from one conformation to another one. Then, we use simplicial complexes to abstract away the interleaving order of the transformations between conformations that commute. As a result, we get a compact summary of the potential behaviour of each protein of the model.

1 Introduction

Thanks to rule-based modelling languages, as Kappa, one can model accurately the biochemical interactions between proteins involved for instance in signalling pathways, without abstracting away *a priori*, when they are available, the mechanistic details about these interactions. For example, one can describe faithfully the formation of dimmers, scaffold proteins, and the phosphorylation of proteins on multiple sites, in a very compact way. Yet, understanding how the behaviour of the systems may emerge from these interactions remains a challenge. Moreover, when models become large, no matter they have been humanly written, or automatically assembled from the literature, as suggested in [14], it becomes crucial to get some automatic tools to understand the content of the models and to check that what is modelled matches with what the modeller has in mind.

We use the abstract interpretation framework [3,4] to systematically derive automatic static analyses for Kappa models. Applications range from model

^{*} This material is based upon works partially sponsored by the Defense Advanced Research Projects Agency (DARPA) and the U. S. Army Research Office under grant number W911NF-14-1-0367, and by the ITMO Plan Cancer 2014. 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 DARPA, the U. S. Department of Defense, or ITMO.

debugging, to the abstraction of complex properties offering new insights to investigate the system overall behaviour. In this paper, we propose to study the behaviour of each protein in isolation. Starting from a formal definition of the trace semantics, we collect the behaviour of each kind of protein independently, and summarise the potential steps to reach these conformations within a transition system. When proteins have too many interaction sites, it is crucial to take benefit of the potential independence between some conformation changes in some protein states. Taking inspiration from simplicial complexes [8], we introduce the notion of macrotransition systems, in which the behaviour of different subsets of sites can be described independently, abstracting away the potential interleaving between their behaviour. The result is a scalable and convenient way to visualise both the different conformations that each protein may take and the causal relations among the different conformation changes.

Related works. A qualitative analysis is proposed in [9,6]. This abstraction captures all the conformations an agent may take in a Kappa model. In the present paper, we go further and compute, for each agent, a transition system that describes the causal relationships among its potential conformational changes.

Causality plays an important role in the understanding and the verification of concurrent systems, as found in Systems Biology. Several frameworks are available to study and understand causality, and to reduce the combinatorial complexity of the models, by exploiting pair of commutative transitions. Partial order reduction is broadly used in model checking [10]. It consists in restricting the transitions of a concurrent system so as to force its computation to follow a canonical order for the interleaving of commutative transitions. Event structures [13] focus on the causal relations between events in a concurrent system. In [5], they provide a compact description of trace samples, in which the events which are not necessary, are discarded. Yet, it is worth noting that these discarded events may have a kinetics impact. An application of event structures in static analysis can be found in [2]. Since they focus on accumulating the effect of causally related transformations, event structures somehow obfuscate the notion of states. Our notion of macrotransition systems is inspired from simplicial complexes. Simplicial complexes can be used for describing concurrent systems up to the interleaving order of commutative transitions [8]. They describe the state of the system as a point moving along a geometrical object, in which commutative transitions are denoted by higher dimension faces. Our formalism offers a convenient compact abstraction of all the potential conformation changes of a protein, without discarding any transition.

Outline. In Sect. 2, we introduce two case studies to motivate our framework. In Sect. 3, we describe Kappa. In Sect. 4, we define its finite trace semantics, that we abstract in Sect. 5, by over-approximating the behaviour of each kind of agent thanks to *local* transition systems. Lastly in Sect. 6, we explain how to abstract away the interleaving order of the transitions that commute in these local transition systems.

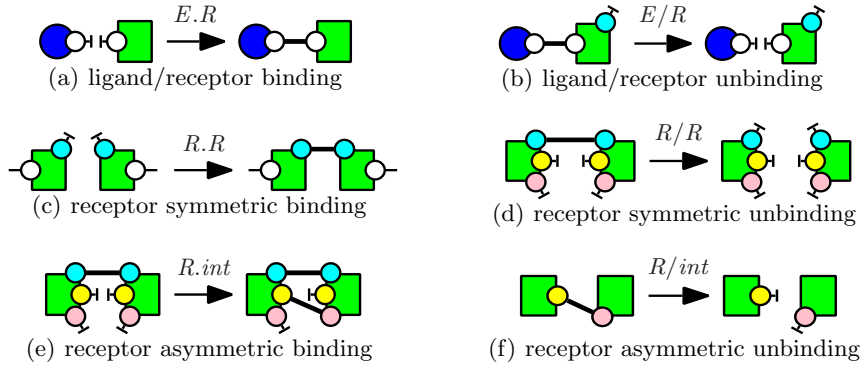


Fig. 1. Rules for dimer formation.

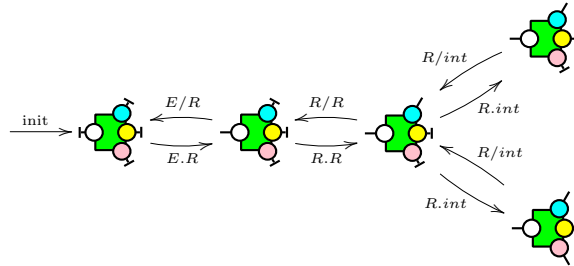


Fig. 2. The local transition system of membrane receptors.

2 Case studies

So as to motivate our goal, we introduce two models as case studies.

The first model describes the formation of some dimmers. Two kinds of proteins are involved: ligands and membrane receptors. When activated by ligands, receptors can form stable dimmers, as described by the means of the interaction rules in Fig. 1. We are interested in one particular binding site in ligand proteins, and in four sites in receptor proteins. Ligand proteins are depicted as circles, whereas receptor proteins are depicted as rectangles. Their binding sites are drawn as smaller circles. Some sites are connected pair-wisely. For the others, we use the symbol '–' to specify a free site and the symbol '–' to specify a site that is bound to an unspecified site. By convention, the site alone on its side in a receptor protein is the one that can bind to a ligand protein; the three sites on the other side can form bonds with other receptors (their order matters).

Let us now give more details about the interactions between these proteins. A ligand protein and a receptor protein may bind to each other provided that the sites that are dedicated to this binding are both free (e. g. see Fig. 1(a)), or detach from each other, provided that the receptor protein is not yet involved in a dimer (e. g. see Fig. 1(b)). Two activated receptor proteins can form a symmetric bond by connecting their respective top-most site (e. g. see Fig. 1(c)),

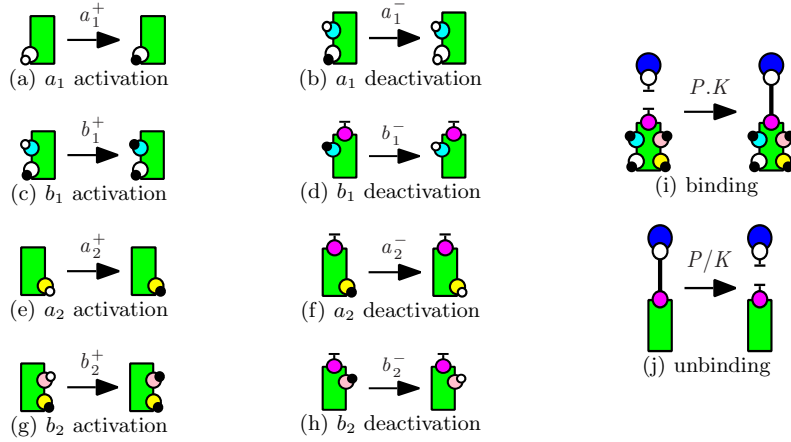


Fig. 3. Rules for the protein with four phosphorylation sites.

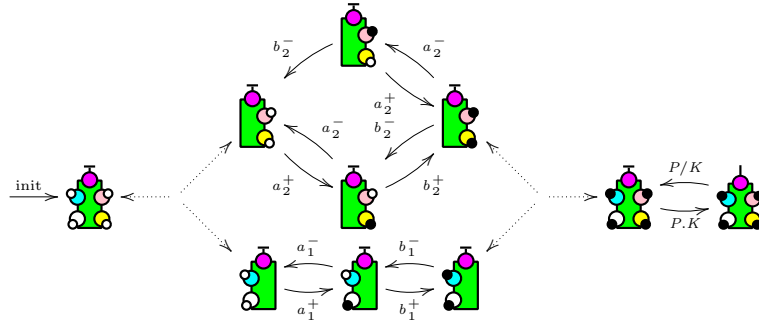


Fig. 4. Local transition system for the protein with four phosphorylation sites.

or break this bond unless an asymmetric bond has been formed already (e. g. see Fig. 1(d)). To gain stability, a dimer with a symmetric link can form an asymmetric one by connecting one of its free site in the first receptor protein to the free site in the other kind in the second receptor protein (e. g. see Fig. 1(e)), or break this connection (e. g. see Fig. 1(f)).

Writing interaction rules can be error prone. Especially, which amount of information should be put in rules, is often not so clear. So as to gain confidence in our modelling process, we propose to compute, for each kind of protein, a *local* transition system. The goal is to abstract the different conformations that each protein may take, and how a given protein may pass from one conformation to another one. As an example, the local transition system for receptor proteins is given in Fig. 2 (there are two transitions for the rule R/Int , since it operates differently on the first and on the second receptort of its left hand side; the same remark holds for the rule $R.Int$). We claim that it provides a helpful summary of the effect of the rules on the behaviour of each protein instance.

When proteins have too many interaction sites, we can no longer describe extensively their sets of potential conformations. Our second model deals with a protein with four phosphorylation sites and a single binding site. The lower left (resp. lower right) site can be phosphorylated without any condition (e. g. see Figs. 3(a) and 3(e)). The upper left (resp. upper right) site can get phosphorylated, if the lower left (resp. lower right) site is still phosphorylated (e. g. see Fig. 3(c) and 3(g)). When the four sites are all phosphorylated, the conformation of the protein changes which reveals the binding site. Then the protein can bind to another kind of protein (e. g. see Fig. 3(i)). This bond can be released with no condition (e. g. see Fig. 3(j)). Phosphorylated sites can be dephosphorylated under the following conditions: as long as a protein is bound, none of its site can be dephosphorylated; as long as the upper left site is phosphorylated, the lower left site cannot be dephosphorylated (e. g. see Figs. 3(b), 3(d), 3(f), and 3(h)).

We notice that, in a protein instance, the potential transformations of the states of both sites on the left *commute* with the potential transformations of those of both sites on the right. Thanks to this, we can describe the transition system between the different conformations of the protein in a more compact way (e. g. see Fig. 4). In this transition system, the behaviour of the pair of sites on the left and of the pair of sites on the right is described as two independent subprocesses. This description is inspired by simplicial complexes [8]. It describes independent processes modulo the interleaving order of their execution.

3 Kappa

In this section, we describe Kappa and its single push-out (SPO) semantics.

Firstly we define the signature of a model.

Definition 1. *A signature is a tuple $\Sigma = (\Sigma_{ag}, \Sigma_{site}, \Sigma_{int}, \Sigma_{ag-st}^{int}, \Sigma_{ag-st}^{lnk})$ where:*

1. Σ_{ag} is a finite set of agent types,
2. Σ_{site} is a finite set of site identifiers,
3. Σ_{int} is a finite set of internal state identifiers,
4. and $\Sigma_{ag-st}^{lnk} : \Sigma_{ag} \rightarrow \wp(\Sigma_{site})$ and $\Sigma_{ag-st}^{int} : \Sigma_{ag} \rightarrow \wp(\Sigma_{site})$ are site maps.

Agent types in Σ_{ag} denote agents of interest, as kinds of proteins for instance. A site identifier in Σ_{site} represents an identified locus for capability of interactions. Each agent type $A \in \Sigma_{ag}$ is associated with a set of sites which can bear an internal state $\Sigma_{ag-st}^{int}(A)$ and a set of sites which can be linked $\Sigma_{ag-st}^{lnk}(A)$. We assume without any loss of generality that $\Sigma_{ag-st}^{lnk}(A) \cap \Sigma_{ag-st}^{int}(A) = \emptyset$, for any $A \in \Sigma_{ag}$ and we write $\Sigma_{ag-st}(A)$ for the set of sites $\Sigma_{ag-st}^{lnk}(A) \uplus \Sigma_{ag-st}^{int}(A)$.

Example 1. We define the signature for the model in the second case study as $\Sigma := (\Sigma_{ag}, \Sigma_{site}, \Sigma_{int}, \Sigma_{ag-st}^{int}, \Sigma_{ag-st}^{lnk})$ where: $\Sigma_{ag} := \{P, K\}$; $\Sigma_{site} := \{a_1, a_2, b_1, b_2, x\}$; $\Sigma_{int} := \{\circ, \bullet\}$; $\Sigma_{ag-st}^{int} := [P \mapsto \{a_1, a_2, b_1, b_2\}, K \mapsto \emptyset]$; $\Sigma_{ag-st}^{lnk} := [P \mapsto \{x\}, K \mapsto \{x\}]$. The agent type P denotes the first kind of proteins and K the second one; the site identifier x denotes the binding site (both in P and K), and the site identifiers a_1, a_2, b_1, b_2 denote respectively the lower left, upper left, lower right, and upper right sites in the protein P .

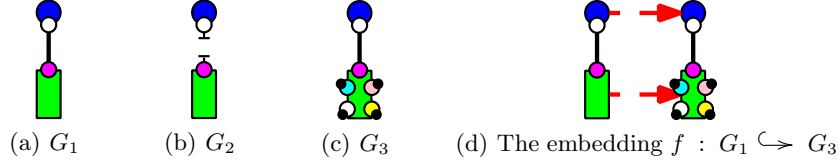


Fig. 5. Three site-graphs G_1 , G_2 , and G_3 , and an embedding f .

Site-graphs describe both patterns and chemical mixtures. Their nodes are typed agents with some sites which can bear internal states and binding states.

Definition 2. A site-graph is a tuple $G = (\mathcal{A}, \text{type}, \mathcal{S}, \mathcal{L}, p\kappa)$ where: 1. $\mathcal{A} \subseteq \mathbb{N}$ is a finite set of agents, 2. $\text{type} : \mathcal{A} \rightarrow \Sigma_{ag}$ is a function mapping each agent to its type, 3. \mathcal{S} is a set of sites such that $\mathcal{S} \subseteq \{(n, i) \mid n \in \mathcal{A}, i \in \Sigma_{ag-st}(\text{type}(n))\}$, 4. \mathcal{L} is a function between the sets $\{(n, i) \in \mathcal{S} \mid i \in \Sigma_{ag-st}^{lnk}(\text{type}(n))\}$ and $\{(n, i) \in \mathcal{S} \mid i \in \Sigma_{ag-st}^{int}(\text{type}(n))\} \cup \{-, -\}$, such that for any two sites $(n, i), (n', i') \in \mathcal{S}$, we have $(n', i') = \mathcal{L}(n, i)$ if and only if $(n, i) = \mathcal{L}(n', i')$; 5. and $p\kappa$ is a function between the sets $\{(n, i) \in \mathcal{S} \mid i \in \Sigma_{ag-st}^{int}(\text{type}(n))\}$ and Σ_{int} .

A site $(n, i) \in \mathcal{S}$ such that $i \in \Sigma_{ag-st}^{int}(\text{type}(n))$ is called a property site, whereas a site $(n, i) \in \mathcal{S}$ such that $i \in \Sigma_{ag-st}^{lnk}(\text{type}(n))$ is called a binding site. Whenever $\mathcal{L}(n, i) = \neg$, the binding site (n, i) is free. Various levels of information can 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 $p\kappa_G$ its set of the internal states.

A mixture is a site-graph in which the state of each site in each agent is documented. Formally, a site-graph G is a chemical mixture, if and only if, $\mathcal{S}_G = \{(n, i) \mid n \in \mathcal{A}_G, i \in \Sigma_{ag-st}(\text{type}_G(n))\}$.

Example 2. Three site-graphs G_1 , G_2 , and G_3 are drawn in Figs. 5(a), 5(b), and 5(c). For the sake of brevity, we only give the explicit definition of the first one: 1. $\mathcal{A}_{G_1} = \{1, 2\}$, 2. $\text{type}_{G_1} = [1 \mapsto P, 2 \mapsto K]$, 3. $\mathcal{S}_{G_1} = \{(1, x), (2, x)\}$, 4. $\mathcal{L}_{G_1} = [(1, x) \mapsto (2, x), (2, x) \mapsto (1, x)]$, 5. $p\kappa_{G_1} = []$; Among these three site-graphs, we notice that only G_3 is a chemical mixture.

Two site-graphs can be related by structure-preserving injective functions, which are called embeddings. the notion of embedding is defined as follows:

Definition 3. An embedding $h : G \hookrightarrow H$ between two site-graphs G and H is a function of agents $h : \mathcal{A}_G \rightarrow \mathcal{A}_H$ satisfying, for all agent identifiers $m, n \in \mathcal{A}_G$, for all site identifiers $i \in \Sigma_{ag-st}(\text{type}_G(n))$, $i' \in \Sigma_{ag-st}(\text{type}_G(n'))$, and for all internal state identifier $\iota \in \Sigma_{int}$: 1. if $m \neq n$, then $h(m) \neq h(n)$; 2. $\text{type}_G(n) = \text{type}_H(h(n))$; 3. if $(n, i) \in \mathcal{S}_G$, then $(h(n), i) \in \mathcal{S}_H$; 4. if $\mathcal{L}(n, i) = (n', i')$, then $\mathcal{L}(h(n), i) = (h(n'), i')$; 5. if $\mathcal{L}(n, i) = \neg$, then $\mathcal{L}(h(n), i) = \neg$; 6. if $\mathcal{L}(n, i) = -$, then $\mathcal{L}(h(n), i) \in \{-\} \cup \mathcal{S}_H$; 7. if $p\kappa(n, i) = \iota$, then $p\kappa(h(n), i) = \iota$.

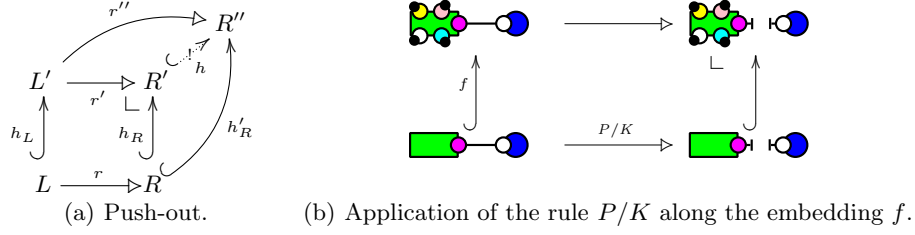


Fig. 6. Rule application.

Example 3. An embedding between G_1 and G_3 , is shown in Fig. 5(d).

Two embeddings respectively between two site-graphs E and F , and between the site-graph F and a site-graph G , compose in the usual way (and form an embedding between the site-graphs E and G). Moreover, two site-graphs E and F , such that there exists an embedding between E and F , and an embedding between F and E , are said isomorphic. An embedding between two isomorphic site-graphs, is called an isomorphism. Given three site-graphs L , R , and D , a couple of embeddings respectively between the site-graphs D and L , and between the site-graphs D and R , is called a span between L and R . Besides, a couple of embeddings respectively between the site-graphs L and D , and between the site-graphs R and D is called a cospan between L and R .

Transformations between site-graphs are described by rules (e. g. see Figs. 1 and 3). For the sake of simplicity, we assume that rules can break and create bonds between sites, and can change the internal states of sites, but we consider neither agent degradation, nor agent creation.

These requirements are formalised in the following definition:

Definition 4. A rule is a span of embeddings $L \xleftarrow{h_L} D \xrightarrow{h_R} R$ such that:
 1. $\mathcal{A}_D = \mathcal{A}_L$ and $\mathcal{A}_D = \mathcal{A}_R$; 2. for all agents $n \in \mathcal{A}_D$, $h_L(n) = n$ and $h_R(n) = n$;
 3. $\mathcal{S}_D = \mathcal{S}_L$ and $\mathcal{S}_D = \mathcal{S}_R$; 4. for all sites $(n, i) \in \mathcal{S}_D$, if $\mathcal{L}_R(n, i) = -$, then $\mathcal{L}_L(n, i) = -$.

Since we do consider neither agent creation, nor agent degradation, we can assume that the agents in the left hand side and in the right hand side of a rule are the same (constraint 1) and that both embeddings preserve agent identifiers (constraint 2). The constraint 3 ensures that, in a rule, sites cannot be removed or added. Lastly, the constraint 4 ensures that, when the binding state of a site is modified, it is not replaced with the state $-$.

A rule $L \xleftarrow{h_L} D \xrightarrow{h_R} R$ is usually denoted as $L \rightarrow R$.

Rules can be applied to site-graphs via an embedding, by the means of a push-out construction.

Definition 5 ([5]). Let r be a rule $L \rightarrow R$, L' be a site-graph, and h_L be an embedding between the site-graphs L and L' . Then, there exists a rule r' between

the site-graph L' and a site-graph R' and an embedding h_R between the site-graphs R and R' such that both following properties are satisfied: 1. $h_{Rr} = r'h_L$; 2. for all rule r'' between the site-graph L' and a site-graph R'' and all embedding h'_R between R and R'' such that: $h'_{Rr} = r''h_L$, there exists a unique embedding h between R' and R'' such that $r'' = hr'$ and $h_{R''} = hh_{R'}$. With these notations, we say that there is a transition from the state L' into 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.

In Def. 5, the compositions between rules and embeddings are defined by the means of a pull-back construction.

Example 4. The embedding f between the site-graph G_1 and the left hand side of the rule P/K induces a computation step as described in Fig. 6(b).

In Kappa, a model \mathcal{M} (over a given signature Σ) is defined as a pair (G_0, \mathcal{R}) where: 1. G_0 is a chemical mixture; 2. and \mathcal{R} is a set of rules. The chemical mixture G_0 denotes the initial state. Since we focus only on qualitative properties, we do not associate rules with kinetic rates.

4 Trace semantics

In this section, we define the semantics of a model (written in Kappa) as the set of the traces that is induced by the underlying transition system.

We assume that we are given \mathbb{Q} a set of *states* and \mathbb{L} a set of *labels*. We call a *transition* any triple (q, λ, q') in the set $\mathbb{Q} \times \mathbb{L} \times \mathbb{Q}$. In Kappa, states are chemical mixtures whereas transition labels are pairs composed of a rule and an embedding between the left hand side of this rule and a chemical mixture.

A transition system is given by a set of initial states and a set of transitions, as formalised in the following definition:

Definition 6 (Transition system). A transition system is a pair (\mathcal{Q}_0, T) where: 1. $\mathcal{Q}_0 \subseteq \mathbb{Q}$; 2. $T \subseteq \mathbb{Q} \times \mathbb{L} \times \mathbb{Q}$.

We denote as $\mathbb{T}_{\mathbb{Q}, \mathbb{L}}$ the set of all the transition systems over \mathbb{Q} and \mathbb{L} . Transition systems can be ordered by the relation \sqsubseteq that is defined as $(Q_0, T) \sqsubseteq (Q'_0, T')$ if and only if 1. $Q_0 \subseteq Q'_0$, 2. and $T \subseteq T'$. The pair $(\mathbb{T}_{\mathbb{Q}, \mathbb{L}}, \sqsubseteq)$ is indeed a complete lattice. This means that any family $(T_i)_{i \in I}$ of transition systems has a least upper bound, that we denote by $\sqcup\{T_i \mid i \in I\}$ (this way, 1. for each element $i \in I$, $T_i \sqsubseteq \sqcup\{T_i \mid i \in I\}$, 2. and for any transition system $Y \in \mathbb{T}_{\mathbb{Q}, \mathbb{L}}$ such that $T_i \sqsubseteq Y$ for each element $i \in I$, we have $\sqcup\{T_i \mid i \in I\} \sqsubseteq Y$).

Each model $\mathcal{M} := (G_0, \mathcal{R})$ in Kappa is associated with the transition system (\mathcal{Q}_0, T) where 1. $\mathcal{Q}_0 = \{G_0\}$; 2. and T is the set of the transitions $(L', (r, h_L), R')$ such that $L' \xrightarrow{(r, h_L)} R'$ as defined in Def. 5.

Each transition system induces a set of traces. In this paper, we focus on finite traces, that are made of an initial state followed by a (potentially empty)

finite sequence of transitions, each of them starting from the state the previous transition had ended in. This is formalised in the following definition:

Definition 7 (Finite traces). *A finite trace is a pair $\tau = (q'_0, (q_i, \lambda_i, q'_i)_{1 \leq i \leq p})$, where q'_0 is a state (in \mathbb{Q}), and $(q_i, \lambda_i, q'_i)_{1 \leq i \leq p}$ is a family of transitions (in $\mathbb{Q} \times \mathbb{L} \times \mathbb{Q}$), such that $q'_{i-1} = q_i$ for each integer i between 1 and p .*

We denote as $\mathcal{T}_{\mathbb{Q}, \mathbb{L}}^*$ the set of all the traces over the sets \mathbb{Q} and \mathbb{L} .

With the notations of Def. 7, we call the state q'_0 (resp. q'_p) the initial (resp. the final) state of the trace τ and we denote it as $\text{fst}(\tau)$, (resp. as $\text{last}(\tau)$). When a trace is made of a single state, we write it as q instead of $(q, ())$. Besides, any transition $t := (q_{p+1}, \lambda_{p+1}, q'_{p+1})$ such that the state q_{p+1} is equal to the final state q'_p of the trace τ , can be concatenated to the trace τ . In such a case, we write $\tau \frown (q_{p+1}, \lambda_{p+1}, q'_{p+1})$ for the finite trace $(q'_0, (q_i, \lambda_i, q'_i)_{1 \leq i \leq p+1})$.

A transition system (\mathcal{Q}_0, T) induces a set of traces $\gamma_{\mathbb{Q}, \mathbb{L}}(\mathcal{Q}_0, T)$, that is defined as the set of the traces $(q'_0, (q_i, \lambda_i, q'_i)_{1 \leq i \leq p})$ such that: 1. the state q'_0 belongs to the set \mathcal{Q}_0 ; 2. and for each integer i between 1 and p , the transition (q_i, λ_i, q'_i) belongs to the set T . The set of traces $\gamma_{\mathbb{Q}, \mathbb{L}}(\mathcal{Q}_0, T)$ can also be defined as the least fix-point of the operator $\mathbb{F}_{\mathcal{Q}_0, T}$ that maps any set of traces $X \in \wp(\mathcal{T}_{\mathbb{Q}, \mathbb{L}}^*)$ into the set of traces $\{q \mid q \in \mathcal{Q}_0\} \cup \{\tau \frown (q, \lambda, q') \mid \tau \in X \wedge \text{last}(\tau) = q \wedge (q, \lambda, q') \in T\}$. The operator $\mathbb{F}_{\mathcal{Q}_0, T}$ is monotonic (that is to say that, for any two sets of traces $X, Y \in \wp(\mathcal{T}_{\mathbb{Q}, \mathbb{L}}^*)$, if $X \subseteq Y$, then $\mathbb{F}_{\mathcal{Q}_0, T}(X) \subseteq \mathbb{F}_{\mathcal{Q}_0, T}(Y)$). By [12], it follows that $\mathbb{F}_{\mathcal{Q}_0, T}$ has a fix-point that is included in any other fix-point. We denote this fix-point $\text{lfp } \mathbb{F}_{\mathcal{Q}_0, T}$ (thus, we have 1. $\text{lfp } \mathbb{F}_{\mathcal{Q}_0, T} = \mathbb{F}_{\mathcal{Q}_0, T}(\text{lfp } \mathbb{F}_{\mathcal{Q}_0, T})$, 2. and for each set $X' \subseteq \wp(\mathcal{T}_{\mathbb{Q}, \mathbb{L}}^*)$ such that $X' = \mathbb{F}_{\mathcal{Q}_0, T}(X')$, we have: $\text{lfp } \mathbb{F}_{\mathcal{Q}_0, T} \subseteq X'$).

Conversely, a set $X \subseteq \mathcal{T}_{\mathbb{Q}, \mathbb{L}}^*$ of finite traces can be abstracted by the transition system $\alpha_{\mathbb{Q}, \mathbb{L}}(X)$ that is defined as the pair (\mathcal{Q}_0, T) with: 1. $\mathcal{Q}_0 = \{\text{fst}(\tau) \mid \tau \in X\}$; 2. $T = \{t_i \mid \exists (q_0, (t_i)_{1 \leq i \leq p}) \in X, i \in \llbracket 1, p \rrbracket\}$. The pair $(\alpha_{\mathbb{Q}, \mathbb{L}}, \gamma_{\mathbb{Q}, \mathbb{L}})$ forms a Galois connection between the complete lattices $(\wp(\mathcal{T}_{\mathbb{Q}, \mathbb{L}}^*), \subseteq)$ and $(\mathbb{T}_{\mathbb{Q}, \mathbb{L}}, \sqsubseteq)$. This means that for any transition system (\mathcal{Q}'_0, T') , we have $\alpha_{\mathbb{Q}, \mathbb{L}}(X) \sqsubseteq (\mathcal{Q}'_0, T')$, if and only if, $X \subseteq \gamma_{\mathbb{Q}, \mathbb{L}}(\mathcal{Q}'_0, T')$. It follows (e. g. see [4]), that: 1. functions $\alpha_{\mathbb{Q}, \mathbb{L}}$ and $\gamma_{\mathbb{Q}, \mathbb{L}}$ are both monotonic; 2. the function $\alpha_{\mathbb{Q}, \mathbb{L}}$ maps each set of finite traces to the smallest (for \sqsubseteq) transition system which induces this set of traces.

5 Local transition systems

In Sect. 4, we have associated each model with a transition system, that describes the set of the finite traces of this model. However, such transition systems are usually too large to be computed, or even if they could, they are too complex to help understanding the behaviour of the models. We propose to simplify these transition systems by focusing on the behaviour of each protein independently, abstracting away which proteins are bound together. This abstraction had already been applied in [9,6], to infer the relationships among the state of sites in protein instances. Here we extend this static analysis to traces.

Firstly we explain how to track the behaviour of each protein independently while forgetting about the bonds between pairs of binding sites. For any agent

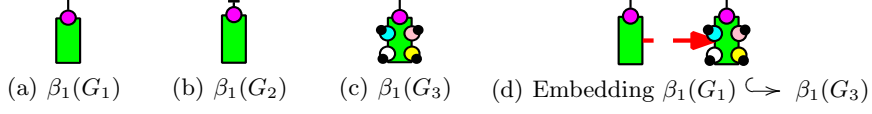


Fig. 7. Abstraction of the site-graphs G_1 , G_2 , G_3 , and of the embedding f .

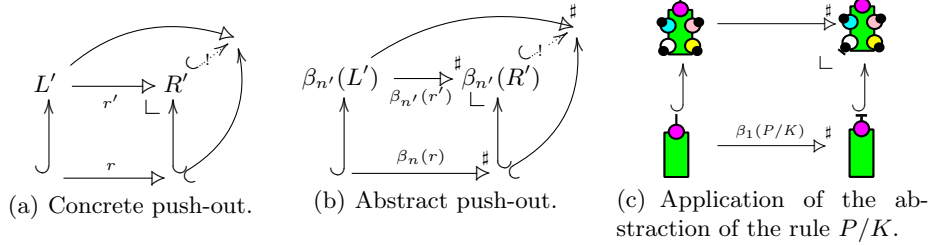


Fig. 8. The abstraction of a push-out (in the concrete) is a push-out (in the abstract), and application of rule (in the abstract).

identifier $n \in \mathbb{N}$, we denote by β_n the function that, when applied to a site-graph G containing an agent with identifier n : 1. replaces any bond between two sites by two occurrences of the symbol ‘-’; 2. restricts the site-graph G to the agent with identifier n ; 3. renames the identifier n with 1. More formally, the site-graph $\beta_n(G)$ is defined by: 1. $\mathcal{A}_{\beta_n(G)} := \{1\}$; 2. $type_{\beta_n(G)} := [1 \mapsto type_G(n)]$; 3. $\mathcal{S}_{\beta_n(G)} := \{(1, i) \mid (n, i) \in \mathcal{S}_G\}$; 4. the function $\mathcal{L}_{\beta_n(G)}$ maps each site $(1, i)$ such that $i \in \Sigma_{ag-st}^{lnk}(type_G(n))$ and $(n, i) \in \mathcal{S}_G$, to the symbol ‘-’ whenever $\mathcal{L}_G(n, i) = -$, and to the symbol ‘-’ otherwise; 5. the function $p\kappa_{\beta_n(G)}$ maps each site $(1, i)$ such that $i \in \Sigma_{ag-st}^{int}(type_G(n))$ and $(n, i) \in \mathcal{S}_G$, to $p\kappa_G(n, i)$.

Example 5. The restriction of the site-graphs G_1 , G_2 , and G_3 to their agent with identifier 1, are depicted in Figs. 7(a), 7(b), and 7(c). For instance, the site-graph $\beta_1(G_1)$ is equal to the tuple $(\{1\}, [1 \mapsto P], \{(1, x)\}, [(1, x) \mapsto -, []])$. Moreover, the function $[1 \mapsto 1]$ induces an embedding between the site-graph $\beta_1(G_1)$ and the site-graph $\beta_1(G_3)$, as depicted in Fig. 7(d).

For each agent type A , we define the set of the local views \mathbb{Q}_A , for the agents of type A , as the set of the site-graphs G with only one agent of type A and identifier 1, that documents the state of all the sites in $\Sigma_{ag-st}(A)$ (that is to say such that $\mathcal{S}_G = \{(1, i) \mid i \in \Sigma_{ag-st}(A)\}$). Besides, we define the set of the local transition labels \mathbb{L}_A as the set of pairs (r, n) where r is a rule, and n is an agent identifier. Intuitively, the local transition label (r, n) denotes the fact that a rule r is applied along an embedding that matches the agent with identifier n in the left hand side of the rule r , to the local view to which we want to apply the rule. Given a rule $r : L \leftarrow D \hookrightarrow R$ and an identifier $n \in \mathcal{A}_L$ of an agent in the left hand side L of the rule r , we define the abstraction $\beta_n(r)$ of the rule r as the span $\beta_n(L) \leftarrow \beta_n(D) \hookrightarrow \beta_n(R)$. In this span, both embeddings are induced by the function $[1 \mapsto 1]$. The copspan $\beta_n(r)$ is not a rule in general, yet

it can be applied along any embedding between its left hand side $\beta_n(L)$ and a site-graph L' thanks to a push-out construction (e. g. see Fig. 8). In such a case, we write $L' \xrightarrow{(r,n)}^\# R'$ for the corresponding abstract computation step. Moreover, we notice that in an abstract computation step $q^\# \xrightarrow{(r,n)}^\# q^{\#'}$ if $q^\#$ is a local view in \mathbb{Q}_A , then $q^{\#'}$ is a local view in \mathbb{Q}_A as well.

Example 6. The application of the abstraction $\beta_1(P/K)$ of the rule P/K can be applied to the site-graph $\beta_1(G_1)$, as drawn in Fig. 8(c).

The application of the function β_n can be lifted to traces. Given a trace $\tau := (q'_0, (q_i, (r_i, f_i), q'_i)_{1 \leq i \leq p}) \in \mathcal{T}_{\mathbb{Q}, \mathbb{L}}^*$ and an agent identifier $n \in \mathcal{A}_{q'_0}$, we define the local trace $\beta_n(\tau)$ as the trace: $(\beta_n(q'_0), (\beta_n(q_{\sigma(i)}), (r_{\sigma(i)}, n_i), \beta_n(q'_{\sigma(i)}))_{1 \leq i \leq p'})$ where 1. $\sigma_1, \dots, \sigma_{p'}$ is the sequence (in increasing order) of the integers i between 1 and p such that $\beta_n(q_i) \neq \beta_n(q_{i+1})$; 2. and for each integer i between 1 and p' , the integer n_i is the unique identifier such that the embedding f_{σ_i} maps the agent with identifier n_i in the left hand side of the rule r_{σ_i} into the agent with identifier n in the chemical mixture q_{σ_i} (there always exists such an integer, otherwise the abstract state would not have been modified).

Now we combine our abstractions to over-approximate the behaviour of each agent as an independent *local* transition system. The abstraction $\alpha_\pi(X)$ of a set of traces $X \subseteq \mathcal{T}_{\mathbb{Q}, \mathbb{L}}^*$ is defined as the function mapping each agent type $A \in \Sigma_{ag}$ into the local transition system $\alpha_{\mathbb{Q}_A, \mathbb{L}_A}(\{\beta_n(\tau) \mid \tau \in X, n \in \mathcal{A}_{\text{fst}(\tau)}\})$ (i. e. the best over-approximation, as a transition system, of the set of the local traces that can be associated with an agent of type A). Conversely, the concretization $\gamma_\pi(Y)$ of a function Y between agent types and local transition systems, is defined as the set of the traces $\tau \in \mathcal{T}_{\mathbb{Q}, \mathbb{L}}^*$ such that for each identifier n of an agent in the initial state $\text{fst}(\tau)$ of the trace τ , the local trace $\beta_n(\tau)$ belongs to the set $\gamma_{\mathbb{Q}_A, \mathbb{L}_A}(Y(\text{type}_{\text{fst}(\tau)}(n)))$.

By [4], the function $\alpha_\pi \circ \mathbb{F}_{\mathbb{Q}_0, T} \circ \gamma_\pi$ is the best abstract counterpart to the function $\mathbb{F}_{\mathbb{Q}_0, T}$. The function $\alpha_\pi \circ \mathbb{F}_{\mathbb{Q}_0, T} \circ \gamma_\pi$ is monotonic and its least fix-point satisfies the inclusion: $\text{lfp } \mathbb{F}_{\mathbb{Q}_0, T} \subseteq \gamma_\pi(\text{lfp } \alpha_\pi \circ \mathbb{F}_{\mathbb{Q}_0, T} \circ \gamma_\pi)$. The least fix-point in the right hand side of the inclusion can be computed in a finite number of iterations, since the domain of $\alpha_\pi \circ \mathbb{F}_{\mathbb{Q}_0, T} \circ \gamma_\pi$ is finite. However computing these iterations is quite cumbersome, because it intertwines the computation of the local transition systems that are associated to each agent type. We propose to desynchronise these computations. To do this, we introduce, for each agent type $A \in \Sigma_{ag}$, the function $\mathbb{F}_{\mathbb{Q}_0, T, A}^\#$ that maps any local transition system $(\mathcal{Q}_0^\#, T^\#) \in \mathbb{T}_{\mathbb{Q}_A, \mathbb{L}_A}$ to the local transition system $(\mathcal{Q}_0^{\#'}; T^{\#'}) \in \mathbb{T}_{\mathbb{Q}_A, \mathbb{L}_A}$ where: 1. $\mathcal{Q}_0^{\#'} = \mathcal{Q}_0^\# \cup \{\beta_n(q_0) \mid q_0 \in \mathcal{Q}_0, n \in \mathcal{A}_{q_0}, \text{type}_{q_0}(n) = A\}$; 2. and $T^{\#'}$ is the union between the set $T^\#$ and the set of the transitions $(q^\#, (r, n), q^{\#'})$ such that the local view $q^\#$ is reachable in the transition system $(\mathcal{Q}_0^\#, T^\#)$ and such that $q^\# \xrightarrow{(r,n)}^\# q^{\#'}$.

For any function Y mapping each agent type $A \in \Sigma_{ag}$ to a local transition system over the states \mathbb{Q}_A and the transition labels \mathbb{L}_A , we have $[\alpha_\pi \circ \mathbb{F}_{\mathbb{Q}_0, T} \circ \gamma_\pi](Y) \sqsubseteq \mathbb{F}_{\mathbb{Q}_0, T, A}^\#(Y)$. Moreover, for each agent type $A \in \Sigma_{ag}$, the function $\mathbb{F}_{\mathbb{Q}_0, T, A}^\#$ is monotonic. By [12], for each agent type $A \in \Sigma_{ag}$, the function



Fig. 9. Pairs of commutative transitions.

$\mathbb{F}_{\mathcal{Q}_0, T, A}^\sharp$ has a least fix-point. By [4], the inclusion $lfp \mathbb{F}_{\mathcal{Q}_0, T} \subseteq \gamma_\pi([A \in \Sigma_{ag} \mapsto lfp \mathbb{F}_{\mathcal{Q}_0, T, A}^\sharp(A)])$ is satisfied, which ensures the soundness of our approach.

Thus, we have derived an abstraction of the finite trace semantics, as a family of local transition systems, that describes the behaviour of each particular type of agent, in isolation. Each such local transition system can be computed independently iteratively. We can apply our framework to abstract the local transition system associated to the membrane receptors in the model of dimmer formation (e. g. see Fig. 1). As expected, the result is the abstract transition system that is given in Fig. 2.

6 Macrotransition systems

The analysis described in Sect. 5 can also be applied to our second case study (e. g. see Fig. 3). But there are too many conformations for the result to be visualisable in practice. In this section, we propose to identify which transitions commute and provide a more compact symbolic representation of local transition systems, which abstracts the interleaving order of commutative transitions.

We firstly define the notion of pairs of commutative transitions in a set of traces. Given a set $X \subseteq \mathcal{T}_{\mathbb{Q}, \mathbb{L}}^*$ of traces, We denote by T_X the set of the transitions that occur in X , i. e. $T_X := \{(q_i, \lambda_i, q'_i) \mid (q_0, (q_i, \lambda_i, q'_i))_{1 \leq i \leq p}, 1 \leq i \leq p\}$. For the sake of simplicity, and since this is the case in Kappa, we assume that any transition is fully defined by the state it is starting from and the label of the transition. Formally, we assume that for any two transitions $(q_1, \lambda_1, q'_1), (q_2, \lambda_2, q'_2) \in T_X$, the states q'_1 and q'_2 are equal whenever the pairs (q_1, λ_1) and (q_2, λ_2) are equal.

Definition 8. *We say that transitions with labels $\lambda_a \in \mathbb{L}$ and $\lambda_b \in \mathbb{L}$ commute in the set of traces $X \subseteq \mathcal{T}_{\mathbb{Q}, \mathbb{L}}^*$, if and only if both following properties are satisfied: 1. for each two traces $(q_0, (t_i)_{1 \leq i \leq p})$ and $(q'_0, (t'_i)_{1 \leq i \leq p'})$ in the set X and an integer j such that: (a) $1 \leq j \leq p$; (b) $j \leq p'$; (c) $q_0 = q'_0$; (d) $t_i = t'_i$, for every integer i such that $1 \leq i < j$; (e) the label of the transition t_j is equal to λ_a ; (f) the label of the transition t'_j is equal to λ_b (we denote $t_j = (q_a, \lambda_a, q'_a)$ and $t'_j = (q_b, \lambda_b, q'_b)$); there exists a trace $(q''_0, (t''_i)_{1 \leq i \leq p''})$ in the set X such that: (a) $j + 1 \leq p''$; (b) $q_0 = q''_0$; (c) $t_i = t''_i$, for every integer i such that $1 \leq i \leq j$; (d) the label of the transition t_{j+1} is equal to λ_b ; 2. and for each trace $(q_0, (t_i)_{1 \leq i \leq p})$ in X , such that there exists three states $q_a, q'_a, q'_b \in \mathbb{Q}$ and an integer j between 1 and $p - 1$ satisfying both $t_j = (q_a, \lambda_a, q'_a)$ and $t_{j+1} = (q'_a, \lambda_b, q'_b)$, there exists a state $q'' \in \mathbb{Q}$ such that the trace $(q_0, (t'_i)_{1 \leq i \leq p})$, where*

t'_i is defined as (q_a, λ_b, q'') whenever $i = j$, as (q'', λ_a, q'_b) whenever $i = j + 1$, and as t_i otherwise, is well defined and belongs to the set X .

In Def. 8, the first property entails that whenever after a given prefix of trace, two transitions that commute are enable (they necessarily starts from the same state), each transition can be followed by the other one modulo the fact that the latter transition should now start from the ending state of the former one (e. g. see Fig. 9(a)), whereas the second property entails that two consecutive transitions that commute can always be performed in the reverse order, modulo the fact that the intermediary state has to be modified (e. g. see Fig. 9(b)).

Given $\mathcal{C} \in \wp(\mathbb{L}^2)$ a set of pairs of transition labels, we define the operator $\rho_{\mathcal{C}}$ which maps each set of traces $X \subseteq \mathcal{T}_{\mathbb{Q}, \mathbb{L}}^*$, into the smallest set of traces $\rho_{\mathcal{C}}(X) \subseteq \mathcal{T}_{\mathbb{Q}, \mathbb{L}}^*$ that contains the set X and in which each pair of transitions $((q_a, \lambda_a, q'_a), (q_b, \lambda_b, q'_b)) \in T_X^2$ such that $(\lambda_a, \lambda_b) \in \mathcal{C}$, commutes. The function $\rho_{\mathcal{C}}$ is an upper closure operator, that is to say that: 1. $\rho_{\mathcal{C}}$ is monotonic; 2. $\rho_{\mathcal{C}}$ is idempotent (i. e. $\rho_{\mathcal{C}} \circ \rho_{\mathcal{C}} = \rho_{\mathcal{C}}$); 3. and $\rho_{\mathcal{C}}$ is extensive (i. e. $X \subseteq \rho_{\mathcal{C}}(X)$, $\forall X \subseteq \mathcal{T}_{\mathbb{Q}, \mathbb{L}}^*$). We notice that the fix-points of the upper closure $\rho_{\mathcal{C}}$ are the set of traces $X \subseteq \mathcal{T}_{\mathbb{Q}, \mathbb{L}}^*$ such that each pair of transitions $((q_a, \lambda_a, q'_a), (q_b, \lambda_b, q'_b)) \in T_X^2$ such that $(\lambda_a, \lambda_b) \in \mathcal{C}$, commutes. We can use the upper closure $\rho_{\mathcal{C}}$ to accelerate the computation of the set of traces $\gamma_{\mathbb{Q}, \mathbb{L}}(\mathcal{Q}_0, T)$ that is induced by a given transition system (\mathcal{Q}_0, T) provided that each pair of transitions $((q_a, \lambda_a, q'_a), (q_b, \lambda_b, q'_b)) \in T_{\gamma_{\mathbb{Q}, \mathbb{L}}(\mathcal{Q}_0, T)}^2$ such that $(\lambda_a, \lambda_b) \in \mathcal{C}$, commutes in the set of traces $\gamma_{\mathbb{Q}, \mathbb{L}}(\mathcal{Q}_0, T)$. The function $\rho_{\mathcal{C}} \circ \mathbb{F}_{\mathcal{Q}_0, T}$ is monotonic and satisfies $\mathbb{F}_{\mathcal{Q}_0, T}(X) \subseteq [\rho_{\mathcal{C}} \circ \mathbb{F}_{\mathcal{Q}_0, T}](X)$, for any set of traces $X \subseteq \mathcal{T}_{\mathbb{Q}, \mathbb{L}}^*$. So, by [12], the function $\rho_{\mathcal{C}} \circ \mathbb{F}_{\mathcal{Q}_0, T}$ has a least fix-point, and by [4], $\text{lfp } \mathbb{F}_{\mathcal{Q}_0, T} \subseteq \text{lfp } [\rho_{\mathcal{C}} \circ \mathbb{F}_{\mathcal{Q}_0, T}]$. If additionally, each pair of transitions $((q_a, \lambda_a, q'_a), (q_b, \lambda_b, q'_b)) \in T_{\gamma_{\mathbb{Q}, \mathbb{L}}(\mathcal{Q}_0, T)}$ such that $(\lambda_a, \lambda_b) \in \mathcal{C}$, commutes, we get that $\rho_{\mathcal{C}}(\text{lfp } \mathbb{F}_{\mathcal{Q}_0, T}) = \text{lfp } \mathbb{F}_{\mathcal{Q}_0, T}$, and thus that $\text{lfp } \mathbb{F}_{\mathcal{Q}_0, T} = \text{lfp } [\rho_{\mathcal{C}} \circ \mathbb{F}_{\mathcal{Q}_0, T}]$.

In Kappa, pairs of commutative local transitions can be identified syntactically. Let us consider an agent type $A \in \Sigma_{ag}$. Given the label $(r, n) \in \mathbb{L}_A$ of a local transition, we denote as $\text{TEST}(r, n)$ the set of the site identifiers i such that the site (n, i) occurs in the domain of the rule r , and as $\text{MOD}(r, n)$ the set of the site identifiers i such that the site (n, i) occurs in the domain of the rule r without occurring in its left hand side. Then, for any two labels $\lambda_1, \lambda_2 \in \mathbb{L}_A$ of local transitions such that both $\text{MOD}(\lambda_1) \cap \text{TEST}(\lambda_2) = \emptyset$ and $\text{MOD}(\lambda_2) \cap \text{TEST}(\lambda_1) = \emptyset$, any pair of local transitions with the labels λ_1 and λ_2 commutes [11].

Example 7. In the rules Fig. 3, we have: We have: $\text{TEST}(b_1^+, 1) = \{a_1, b_1\}$; $\text{MOD}(b_1^+, 1) = \{b_1\}$; $\text{TEST}(b_2^+, 1) = \{a_2, b_2\}$; $\text{MOD}(b_2^+, 1) = \{b_2\}$. As a consequence any two local transitions with respective labels $(b_1^+, 1)$ and $(b_2^+, 1)$ commute.

Given a local view $v \in \mathbb{Q}_A$, we consider the set $\Lambda(v)$ as the set of all the transitions labels λ such that there exists a local transition starting from the view v and with the label λ . We define the site-graph $\text{FRAME}_{\lambda}(v)$ as the site-graph that is obtained by removing from the site-graph v any site that belongs to the set $\bigcup \{\text{MOD}(\lambda') \mid \lambda' \in \Lambda(v) \setminus \{\lambda\}\}$. Intuitively, $\text{FRAME}_{\lambda}(v)$ is the restriction of the local view v to the sites that cannot be modified by local transitions that commute with a local transition with the label λ .

We are left to provide a compact data-structure to represent transition systems with pairs of commutative transitions. We propose to use macrostates and macrotransitions. A *macrostate* is a symbolic representation of a set of (micro)states (that behave similarly), and *macrotransitions* are transitions between macrostates, that denote some transitions between the corresponding microstates. Let us assume that we are given a set of macrostates \mathbb{Q}^\sharp . Each macrostate $q^\sharp \in \mathbb{Q}^\sharp$ denotes a set of microstates $\Gamma_{\mathbb{Q}, \mathbb{Q}^\sharp}(q^\sharp) \subseteq \mathbb{Q}$.

A macrotransition system is defined as follows:

Definition 9. A macrotransition system is a pair $(\mathcal{Q}_0, T^\sharp)$ where: 1. $\mathcal{Q}_0 \subseteq \mathbb{Q}$; 2. $T^\sharp \subseteq \mathbb{Q}^\sharp \times \mathbb{L} \times \mathbb{Q}^\sharp$.

A macrotransition system is made of a set of initial microstates, and a set of labelled transitions between macrostates. Each macrotransition is an implicit representation of a set of transitions between microstates. Formally, each macrotransition $(q^\sharp, \lambda, q'^\sharp)$ denotes the set of the transitions $(q, \lambda, q') \in \mathbb{Q} \times \mathbb{L} \times \mathbb{Q}$ for which there exists a set of macrostates $X \subseteq \mathbb{Q}^\sharp$ satisfying: $q = \bigcap \{ \Gamma_{\mathbb{Q}, \mathbb{Q}^\sharp}(x) \mid x \in X \cup \{q^\sharp\} \}$ and $q' = \bigcap \{ \Gamma_{\mathbb{Q}, \mathbb{Q}^\sharp}(x) \mid x \in X \cup \{q'^\sharp\} \}$. We denote as $\Gamma'_{\mathbb{Q}, \mathbb{Q}^\sharp}(q^\sharp, \lambda, q'^\sharp)$ the set of the transitions denoted by the macrotransition $(q^\sharp, \lambda, q'^\sharp)$. A macrotransition system $(\mathcal{Q}_0, T^\sharp)$ is an abstraction of the transition system $(\mathcal{Q}_0, \cup \{ \Gamma'_{\mathbb{Q}, \mathbb{Q}^\sharp}(t) \mid t \in T^\sharp \})$, that we denote as $\Gamma_{\mathcal{Q}_0, T, \mathbb{Q}^\sharp}(\mathcal{Q}_0, T^\sharp)$.

In Kappa, a macrostate is a site-graph that can be embedded in a local view. Each macrostate intentionally denotes the set of the local views it can be embedded in. Now we mimick the computation of $\mathbb{F}_{\mathcal{Q}_0, T, A}^\sharp$ in macrotransition systems. We define the function $\mathbb{G}_{\mathcal{Q}_0, T^\sharp}^\sharp$ mapping each macrotransition system $(\mathcal{Q}_0, T^\sharp)$ to the macrotransition system where the set of initial states is defined as the union between \mathcal{Q}_0 and the set $\{ \beta_n(q_0) \mid n \in \mathcal{A}(q_0), \text{type}_{q_0}(n) = A \}$; and where the set of macrotransitions is obtained 1. by adding in the set T^\sharp any transition of the form $(\text{FRAME}_\lambda(v), \lambda, v')$ for any local view $v \in \mathbb{Q}_A$ such that: (a) the local view v is reachable in the transition system $\gamma_{\mathbb{Q}_A, \mathbb{L}_A}(\Gamma_{\mathcal{Q}_0, T, \mathbb{Q}^\sharp}(\mathcal{Q}_0, T^\sharp))$; (b) and $\text{FRAME}_\lambda(v) \xrightarrow{\lambda}^\sharp v'$, 2. before removing any macrotransition t such that there exists a macrotransition t' satisfying $\Gamma'_{\mathbb{Q}, \mathbb{Q}^\sharp}(t) \subset \Gamma'_{\mathbb{Q}, \mathbb{Q}^\sharp}(t')$.

For any macrotransition system X^\sharp , both following inclusions are satisfied:

1. $\gamma_{\mathbb{Q}_A, \mathbb{L}_A}(\mathbb{F}_{\mathcal{Q}_0, T, A}^\sharp(\Gamma_{\mathcal{Q}_0, T, \mathbb{Q}^\sharp}(X^\sharp))) \subseteq \gamma_{\mathbb{Q}_A, \mathbb{L}_A}(\Gamma_{\mathcal{Q}_0, T, \mathbb{Q}^\sharp}(\mathbb{G}_{\mathcal{Q}_0, T^\sharp}^\sharp(X^\sharp)))$ (soundness);
2. $\gamma_{\mathbb{Q}_A, \mathbb{L}_A}(\Gamma_{\mathcal{Q}_0, T, \mathbb{Q}^\sharp}(\mathbb{G}_{\mathcal{Q}_0, T^\sharp}^\sharp(X^\sharp))) \subseteq \rho\mathcal{C}(\gamma_{\mathbb{Q}_A, \mathbb{L}_A}(\mathbb{F}_{\mathcal{Q}_0, T, A}^\sharp(\Gamma_{\mathcal{Q}_0, T, \mathbb{Q}^\sharp}(X^\sharp))))$ (relative completeness). It follows (since $\Gamma_{\mathcal{Q}_0, T, \mathbb{Q}^\sharp}(\emptyset, \emptyset) = (\emptyset, \emptyset)$) that, when $k \in \mathbb{N}$ increases, the sequence of the images by $[\gamma_{\mathbb{Q}_A, \mathbb{L}_A} \circ \Gamma_{\mathcal{Q}_0, T, \mathbb{Q}^\sharp}]$ of the iterates $\mathbb{G}_{\mathcal{Q}_0, T^\sharp}^{\sharp(k)}(\emptyset, \emptyset)$ of the function $\mathbb{G}_{\mathcal{Q}_0, T^\sharp}^\sharp$, starting from the macrotransition system (\emptyset, \emptyset) , stations ultimately at the value $\gamma_{\mathbb{Q}_A, \mathbb{L}_A}(\text{lfp } \mathbb{F}_{\mathcal{Q}_0, T, A}^\sharp)$. The result of our analysis is defined as the macrotransition system $\mathbb{G}_{\mathcal{Q}_0, T^\sharp}^{\sharp(l)}(\emptyset, \emptyset)$, where $l \in \mathbb{N}$ is the rank at which this limit is reached.

Our analysis is integrated within an open-source static analyser [1]. It relies on binary decision diagrams, to describe implicitly which microtransitions are not yet covered by any macrotransition (as required both in the function $\mathbb{G}_{\mathcal{Q}_0, T^\sharp}^\sharp$

and to detect when to stop the iterations). We obtain, for the second case study in Fig. 3, the macrotransition system that is described in Fig. 4. We have also applied to a similar model with 5 pairs of phosphorylation sites and obtained the corresponding macrotransition system in around one second.

7 Conclusion

Kappa [7] allows for the description of highly combinatorial systems of interactions between proteins. But it is not always obvious to check the consistency of the models that are written in Kappa or to get an overview of how these models work. To cope with this issue, we have proposed an abstract interpretation framework to automatically over-approximate the behaviour of each agent of a model, independently, by the means of a *local* transition system.

Since these local transition systems may remain too combinatorial when proteins have many interaction sites, we have designed a coarser description of the local transition systems, inspired by simplicial systems. This latter description abstracts away the interleaving order between commutative transitions, by the means of transitions between macrostates, that denotes symbolically sets of transitions between microstates. Our tool computes these macrotransitions directly.

Acknowledgement. This work has been motivated by models written by Héctor Medina, and by Nathalie Th eret and Jean Cocquet. We deeply thank them, as well as Pierre Boutillier, Ioana Cristescu, Vincent Danos, Walter Fontana, Russ Harmer, Jean Krivine, Jonathan Laurent, and Jean Yang, for fruitful discussions.

References

1. Boutillier, P., Feret, J., Krivine, J., Q., Kim L y.: Kasim development homepage, <http://kappalanguage.org>
2. Chatain, T., Haar, S., Jezequel, L., Paulev e, L., Schwoon, S.: Characterization of reachable attractors using Petri net unfoldings. In: Computational Methods in Systems Biology. LNCS, vol. 8859 (2014)
3. Cousot, P., Cousot, R.: Abstract interpretation: A unified lattice model for static analysis of programs by construction or approximation of fixpoints. In: Proc. POPL'77 (1977)
4. Cousot, P., Cousot, R.: Systematic design of program analysis framework. In: Proc. POPL'79 (1979)
5. Danos, V., Feret, J., Fontana, W., Harmer, R., Hayman, J., Krivine, J., Thompson-Walsh, C.D., Winskel, G.: Graphs, rewriting and pathway reconstruction for rule-based models. In: Proc. FSTTCS '12. LIPIcs, vol. 18 (2012)
6. Danos, V., Feret, J., Fontana, W., Krivine, J.: Abstract interpretation of cellular signalling networks. In: Proc. VMCAI'2008. LNCS, vol. 4905. Springer (2008)
7. Danos, V., Laneve, C.: Formal molecular biology. TCS 325(1) (2004)
8. Fajstrup, L., Goubault, E., Rau fen, M.: Detecting deadlocks in concurrent systems. In: Proc. CONCUR '98. LNCS, vol. 1466 (1998)
9. Feret, J.: Reachability analysis of biological signalling pathways by abstract interpretation. In: Proc. ICCMSE'2007. vol. 963.(2). AIP (2007)

10. Godefroid, P.: Partial-Order Methods for the Verification of Concurrent Systems - An Approach to the State-Explosion Problem, LNCS, vol. 1032 (1996)
11. Laurent, J.: Causal analysis of rule-based models of signaling pathways (2015)
12. Tarski, A.: A lattice-theoretical fixpoint theorem and its applications. *Pacific J. Math.* 5(2) (1955)
13. Winskel, G.: Event structures. In: *Advances in Petri Nets*. LNCS, vol. 255 (1987)
14. You, J.: Darpa sets out to automate research. *Science* 347 (2015)