An algebraic approach for inferring and using symmetries in rule-based models

Jérôme Feret[1]

DIENS (INRIA/ÉNS/CNRS), Paris, France

Abstract

Symmetries arise naturally in rule-based models, and under various forms. Besides automorphisms between site graphs, which are usually built within the semantics, symmetries can take the form of pairs of sites having the same capabilities of interactions, of some protein variants behaving exactly the same way, or of some linear, planar, or 3D molecular complexes which could be seen modulo permutations of their axis and/or mirror-image symmetries.
In this paper, we propose a unifying handling of symmetries in Kappa. We follow an algebraic approach, that is based on the single pushout semantics of Kappa. We model classes of symmetries as finite groups of transformations between site graphs, which are compatible with the notion of embedding (that is to say that it is always possible to restrict a symmetry that is applied with the image of an embedding to the domain of this embedding) and we provide some assumptions that ensure that symmetries are compatible with pushouts. Then, we characterise when a set of rules is symmetric with respect to a group of symmetries and, in such a case, we give sufficient conditions so that this group of symmetries induces a forward bisimulation and/or a backward bisimulation over the population semantics.

Keywords: Rule-based models, symmetries, category theory, group actions, bisimulations

1 Introduction

Symmetries play an important role in rule-based modelling. One simple example is the case of symmetric sites, that is to say when two sites in a protein have exactly the same capabilities of interaction. Symmetric sites can be handled with in various ways according to the choice of modelling language. Some languages propose syntactic solutions to describe them explicitly. In BNGL [1] a site can occur several times in the interface of an agent. In the Formal Cell Machinery Language [7] and in React(C) [20], one can use hyper-edges to connect several sites of the same kind with a given agent. In Kappa [13], there is no direct means to specify that two sites are symmetric in the core language, but it is possible in higher level front ends

[1] This material is based upon some works sponsored by the "Agence Nationale de la Recherche" (ANR) under the Chair of Excellence AbstractCell and upon some 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 "Agence Nationale de la Recherche", the Defense Advanced Research Projects Agency, or the Department of Defense.

[2] feret@ens.fr

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

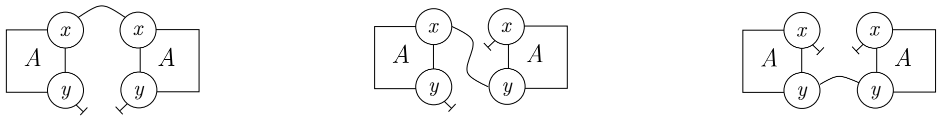
such as Meta-Kappa [10]. Models in Meta-Kappa can then be macro-processed into proper Kappa models, by enumerating the concrete instances of symmetric rules. Another approach consists in detecting which sites are symmetric by inspecting the rules, and in lumping [2,16] the states of the model accordingly [6], by the means of back and forth (both forward and backward) bisimulations [3]. As a consequence, symmetric models enjoy nice statistical properties: (i) if for each pair (q, q') of symmetric states, the system has the same probability (we assume for the sake of simplicity that q and q' have no non trivial automorphism) to be in the state q as in the state q' at a given time t, then, it is also the case for any time t' such that $t' \geq t$. (ii) otherwise, if there exists a state that can be reached by each other state in the system, any two symmetric states q and q' will have the same probability to occur at the limit when time goes to the infinity. Moreover, the bisimulations that are induced by symmetries are particular in the sense that they are induced by a partitioning of the variables (of the systems) and can also be used to lump the ODE semantics of Kappa [4].

Symmetries between sites are not the only kind of symmetries: there are many other kinds. For instance, we can consider the sites of an agent as a list of ordered loci on a ring, seen modulo circular permutations. Another example is the case of rigid structures. In such structures, the sites of agents should not be permuted independently: we should only consider as symmetries the transformations in which the sites of each pair of agents that have the same type and that are in the same connected component are reordered by the same permutation. This former kind of symmetry is especially useful in the case of macro-molecules, self-assembly models, and diffusion models. For instance, in the population migration model that is described in [19], ants are moving on a landscape that is encoded as a grid of agents linked by some specified sites named 'north', 'west', 'south', and 'east'. These sites encode the orientation of the landscape: a rotation of the landscape can be modelled by applying the same circular permutation to the four direction sites of all the agents of the grid. To the best of our knowledge, there is no tool to describe such symmetries in any rule-based language.

In this paper, we propose an algebraic framework for describing symmetries among site-graphs. This framework is abstract enough so that all the kinds of symmetries that we have mentioned so far can be dealt with in a uniform setting. Indeed, circular permutations and homogeneous symmetries (in which the same permutation is applied with the sites of all the agents of a given connected component) can easily be defined as subgroups of another (simpler) group of symmetries. Then, we show that our notion of symmetry is compatible with the single pushout construction, which we use as a foundation of the operational semantics of Kappa [8]. We give sufficient conditions for a model to be symmetric with respect to a given group of symmetries, and give extra-assumptions so that these symmetries induce forward bisimulations or even back and forth bisimulations over the Markov chain that is induced by this model.

2 Kappa

In this section, we describe the single-pushout (SPO) semantics of Kappa.



(a) Two agents of type A bound via their site x.

(b) Two agents of type A bound respectively via their site x and y.

(c) Two agents of type A bound via their site y.

Fig. 1. Three site graphs.

2.1 Site graphs

Firstly we define the signature of a model.

Definition 2.1 A *signature* is a tuple $\Sigma = (\Sigma_{ag}, \Sigma_{st}, \Sigma_{ag\text{-}st})$ where Σ_{ag} is a finite set of agent types, Σ_{st} is a finite set of site identifiers, $\Sigma_{ag\text{-}st} : \Sigma_{ag} \to \wp(\Sigma_{st})$ is a site map.

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. Each agent type A is associated with a set of sites which can be linked $\Sigma_{ag\text{-}st}(A)$ (we omit the use of internal states so as to simplify the presentation).

Example 2.2 We consider only one type of agent, A, having two sites x and y. This is encoded by the following signature: $\Sigma \triangleq (\{A\}, \{x, y\}, [A \mapsto \{x, y\}])$.

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, the nodes of which are typed agents with some sites which can bear a linking state.

Definition 2.3 A *site-graph* is a tuple $G = (\mathcal{A}, \mathit{type}, \mathcal{S}, \mathcal{L})$ where $\mathcal{A}$ is a set of agents, $\mathit{type} : \mathcal{A} \to \Sigma_{ag}$ is a function mapping each agent to its type, $\mathcal{S}$ is a set of sites such that $\mathcal{S} \subseteq \{(n, i) \mid n \in \mathcal{A}, i \in \Sigma_{ag\text{-}st}(\mathit{type}(n))\}$, $\mathcal{L}$ is a symmetric relation such that $\mathcal{L} \subseteq (\mathcal{S} \cup \{\dashv, -\})^2 \setminus \{\dashv, -\}^2$; such that: (i) the set $\mathcal{A}$ is finite; (ii) its link relation $\mathcal{L}$ is irreflexive; (iii) for any binding site $(n, i) \in \mathcal{S}$, $((n, i), x) \in \mathcal{L}$ and $((n, i), y) \in \mathcal{L}$ implies $x = y$.

Whenever $((n, i), \dashv) \in \mathcal{L}$, the binding site (n, i) is free. Various levels of information can be given about the sites that are bound. Whenever $((n, i), -) \in \mathcal{L}$, then the binding site (n, i) is bound to some other site which is not specified. Whenever $((n, i), s) \in \mathcal{L}$ with $s \in \mathcal{S}$ then the binding site (n, i) is bound to the binding site s. We introduce a sub-typing relation $\leq_G$ over binding states, that is defined as the least reflexive relation such that $- \leq_G s$ for any $s \in \mathcal{S}$.

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, and $\mathcal{L}_G$ its set of links.

Example 2.4 Examples of site graphs (for the signature given in Exa. 2.2) are given in Fig. 1.

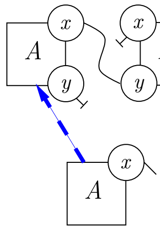


Fig. 2. an embedding between two site graphs.

2.2 Embeddings

Two site graphs can be related by structure-preserving injective functions, which are called embeddings, as illustrated in Fig. 2.

Definition 2.5 An *embedding* $h : G \hookrightarrow H$ between two site graphs G and H is a function of agents $h : \mathcal{A}_G \to \mathcal{A}_H$ satisfying: (i) $m = n$ for all m,$n \in \mathcal{A}_G$ such that $h(n) = h(m)$; (ii) $type_G(n) = type_H(h(n))$ for all $n \in \mathcal{A}_G$; (iii) $(h(n), i) \in \mathcal{S}_H$ for all $(n, i) \in \mathcal{S}_G$; (iv) $((h(n), i),(h(n'), i')) \in \mathcal{L}_H$ for all $((n, i),(n', i')) \in \mathcal{L}_G \cap \mathcal{S}_G^2$; (v) there exists $y \in \mathcal{S}_H \cup \{\dashv, -\}$ such that $((h(n), i), y) \in \mathcal{L}_H$ and $x \leq_H y$ for all $((n, i), x) \in \mathcal{L}_G$ such that $x \in \{\dashv, -\}$.

Whenever f is an embedding between two site graphs E and F, the site graph E (resp. F) is called the domain of (resp. the image of) f and is denoted as $dom(f)$ (resp. as $im(f)$). The number of embeddings between two site graphs G and H is denoted as $[G, H]$. Whenever $G = H$, f is called an automorphism. We notice that the identity function always induces an automorphism. The identity automorphism of a site graph E is denoted as i_E. Two embeddings f and g such that $im(f) = dom(g)$ compose in the usual way (and form an embedding between the domain of f and the image of g). Moreover, whenever two embeddings f and g satisfy (i) $im(f) = dom(g)$, (ii) $im(g) = dom(f)$ and (iii) $gf = i_{im(g)}$, the composition fg is equal to $i_{im(f)}$. In such a case, f and g are called isomorphisms, f is said to be the inverse of g, and G and H are said to be isomorphic which is written $G \approx H$. All the constructions in this paper are defined up to isomorphism.

Lastly, two embeddings f and g are said to be isomorphic, which is written $f \approx g$, if and only if there exists an isomorphism ϕ such that $f = \phi g$.

2.3 Weak embeddings

We stress out on the fact 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 can be embedded only into a site that is free. It is sometimes convenient to relax the definition of embedding so as to allow sites that are free to be mapped to arbitrary sites, which gives the notion of weak embedding.

Definition 2.6 A weak embedding between two site-graphs G and H is an embedding between the site graph $\hat{G}$ and the site graph H, where the site graph $\hat{G}$ is defined by: $\mathcal{A}_{\hat{G}} \triangleq \mathcal{A}_G$, $\mathcal{S}_{\hat{G}} \triangleq \mathcal{S}_G$, $\mathcal{L}_{\hat{G}} \triangleq \{(x, y) \in \mathcal{L}_G \mid x \neq\dashv \text{ and } y \neq\dashv\}$.

A weak embedding between two site graphs G and H is denoted as $G \rightarrowtail H$.

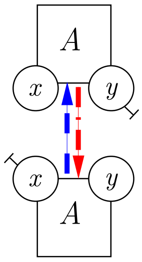


Fig. 3. A pair of weak embeddings between two non isomorphic site graphs.

We notice that any embedding is also a weak embedding (but the converse is wrong in general). Moreover, there may exist a weak embedding $f : G \rightarrowtail H$ between two site graphs G and H and a weak embedding $g : H \rightarrowtail G$ between the two site graphs H and G, even if the site graphs G and H are not isomorphic (see Fig. 3 for an example). Thus, weak embedding must be handled with very carefully.

2.4 Partial embeddings

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 in 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 pair of embeddings with the same domain. For the sake of generality, we define firstly the notion of weak partial embeddings.

Definition 2.7 A *weak partial embedding* $\phi : L \leftarrowtail D \rightarrowtail R$ between two site graphs L and R with domain D, is a pair (h_L, h_R) made of a weak embedding h_L between the site graphs D and L and a weak embedding h_R between the site graphs D and R

In a weak partial embedding $\phi : L \leftarrowtail D \rightarrowtail R$, 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 between a site graph and 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 can be composed thanks to the pullback construction.

Definition 2.8 Let $\phi : L \xleftarrow{h_L} D_1 \xrightarrow{h_R} R$ and $\phi' : R \xleftarrow{h'_L} D_2 \xrightarrow{h'_R} R'$ be two weak partial embeddings such that $\mathrm{rhs}(\phi) = \mathrm{lhs}(\phi')$. There necessarily exist a site graph D_3 and a partial embedding $\phi'' : D_1 \xleftarrow{h_{D_1}} D_3 \xrightarrow{h_{D_2}} D_2$ between the site graph D_1 and 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 $\phi''' \ : \ D_1 \xleftarrow{h'_{D_1}} D_4 \xhookrightarrow{h'_{D_2}} D_2$ such that $h_R h'_{D_1} = h'_L h'_{D_2}$, there exists a unique embedding h between D_4 and D_3 such that $h'_{D_1} = h_{D_1} h$ and $h'_{D_2} = h_{D_2} h$.

$$\begin{tikzcd}
L & & R & & R' \\
& D_1 \arrow[ul, "h_L"'] \arrow[ur, "h_R"] & & D_2 \arrow[ul, "h'_L"'] \arrow[ur, "h'_R"'] & \\
& & D_3 \arrow[ul, hook, "h_{D_1}"'] \arrow[ur, hook', "h_{D_2}"] & & \\
& & D_4 \arrow[uul, hook, "h'_{D_1}"] \arrow[uur, hook', "h'_{D_2}"'] \arrow[u, dotted, "h"', "!"] & &
\end{tikzcd}$$

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 can be replaced with any isomorphic one.

A weak embedding h between two site graphs L and R can be seen as a 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.

2.5 Rules

Transformations between site graphs are described by rules. Some examples are given in Fig. 4.

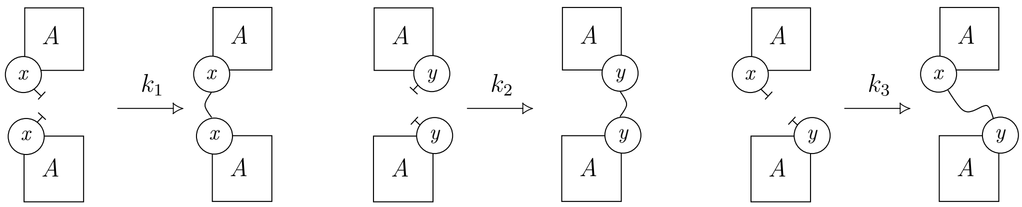
A rule is a transformation between two site graphs, a left hand side (*lhs*) L and a right hand side (*rhs*) R. In a rule, some agents and some sites are preserved. This is specified by a site graph D which is embedded both into L and into R and which describes anything that is preserved. Not all transformations 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 2.9 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 $((n, i), -) \in \mathcal{L}_R$ then there exists $m \in \mathcal{A}_D$ such that $n = h_R(m)$, $(m, i) \in \mathcal{S}_D$, and $((m, i), -) \in \mathcal{L}_D$;

(iii) if $m \in \mathcal{A}_D$, then for any $i \in \Sigma_{\mathit{ag\text{-}st}}(\mathit{type}_D(m))$, $(m, i) \in \mathcal{S}_D$ if and only if $(h_L(m), i) \in \mathcal{S}_L$ if and only if $((h_R(m), i)) \in \mathcal{S}_R$; and, in such a case, there exists $y \in \mathcal{S}_L \cup \{\dashv, -\}$ such that $((h_L(m), i), y) \in \mathcal{L}_L$ if and only if there exists $y \in \mathcal{S}_R \cup \{\dashv, -\}$ such that $((h_R(m), i), y) \in \mathcal{L}_R$;



(a) r_1 binds two sites x.

(b) r_2 binds two sites y.

(c) r_3 binds the site x and the site y of two different agents.

Fig. 4. Under which conditions are the sites x and y symmetric ?

(iv) if $m \in \mathcal{A}_R$ and $m \notin im(h_R)$, then, for any $i \in \Sigma_{ag\text{-}st}(type_R(m))$, $(h_R(m), i) \in \mathcal{S}_R$ and there exists $y \in \mathcal{S}_R \cup \{\dashv\}$ such that $((h_R(m), i), y) \in \mathcal{L}_R$.

The constraint i ensures that D is a local greatest upper bound. The constraint ii ensures that when a site gets bound, we know which site it is bound to. 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 documented.

A rule $L \hookleftarrow D \hookrightarrow R$ is usually denoted as $L \twoheadrightarrow R$ (leaving the two embeddings and the common region implicit).

Now we give the definition of a model.

Definition 2.10 A *model* $\mathcal{M}$, is a function from the set of rules into the set of non-negative real numbers $\mathbb{R}^+$, such that only a finite number of rules has a non zero rate.

In a model $\mathcal{M}$, the rate of the rule r is usually denoted as $k_{\mathcal{M}}(r)$. We also introduce the corrected rate $\gamma_{\mathcal{M}}(r)$ of the rule r as the quotient between $k_{\mathcal{M}}(r)$ and the number $[\text{lhs}(r), \text{lhs}(r)]$ of automorphisms in the left hand side of the rule r.

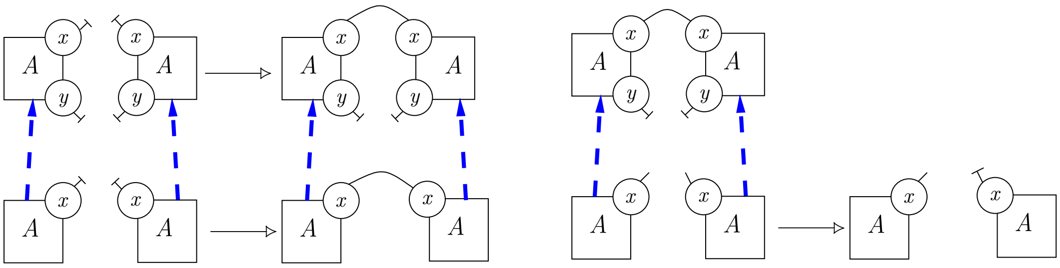
2.6 Refinements

Rules can be more or less refined [9,21], by adding more or less information about the context in which they can be applied.

Definition 2.11 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 sites graphs L' and R', $h_{L'}$ is an embedding between the site graphs L and L', and $h_{R'}$ is an embedding between the site graphs R and R' such that: (i) $h_{R'}r = r'h_{L'}$; (ii) and for any rule r'' between the site graph L' and a site graph R'', and any embedding $h_{R''}$ between the site graphs R and R'', such that $h_{R''}r = r''h_{L'}$, there exists a unique weak embedding h between R' and R'' such that $r'' = hr'$ and $h_{R''} = hh_{R'}$.

$$\begin{tikzcd}
 & & R'' \\
L' \arrow[r, "r'"] \arrow[urr, "r''"] & R' \arrow[ur, dotted, "h_!"] & \\
L \arrow[r, "r"] \arrow[u, hook, "h_{L'}"] & R \arrow[u, hook, "h_{R'}"'] \arrow[uur, hook, "h_{R''}"'] &
\end{tikzcd}$$

We also say that the pair $(r', h_{R'})$ is a pushout of the pair $(h_{L'}, r)$.



(a) An embedding between the left hand side of a rule and a fully specified site graph that induces a refinement.

(b) An embedding between the left hand side of a rule and a fully specified site graph that cannot induce a refinement.

Fig. 5. In (a), applying the rule consists in binding the sites x of both agents. In (b), applying the rule would require to keep the site of the first agent bound, and to release the binding of the site of the second agent. This is not possible since these two sites are bound together in the site graph on the top.

Example 2.12 We give in Fig. 5 two examples of embedding between the left hand side of a rule and a site graph. In 5(a), the embedding induces a refinement. This refinement is obtained by binding the sites x of both agents. In 5(b), no refinement can be formed. Indeed, the rule requires to keep one site x bound and to release the other one and the embedding states that the two sites are bound together. Thus it is not possible to release one, without releasing the other.

2.7 Semantics

Now we are ready to define the stochastic semantics of Kappa, as a weighted transition system [17], that is made of a set of states, related by labelled transitions weighted by non-negative real numbers.

The states of the system are the isomorphism classes of fully specified site graphs, where a fully specified site graph is a site graph that cannot be embedded in any non-isomorphic site graph without adding a new connected component.

The fact that an embedding does not add a connected component is captured by the notion of epimorphism [11, Cor IV.6] which is defined as follows:

Definition 2.13 An embedding f between two site graphs E and F is called an epimorphism, if and only if, for any site graph G and any two embeddings g_1 and g_2 between the site graphs F and G such that $g_1 f = g_2 f$, we have $g_1 = g_2$.

Now we can define fully specified site graphs.

Definition 2.14 We say that a site graph E is *fully specified* if and only for any epimorphism f such that $dom(f) = E$, the site graphs E and $im(f)$ are isomorphic.

Intuitively, in a fully specified site graph G, each agent documents all its sites (*i.e.* $\mathcal{S}_G = \{(n,i) \mid n \in \mathcal{A}_G, i \in \Sigma_{ag\text{-}st}(type_G(n))\}$) and each site in $\mathcal{S}_G$ is either free, or bound explicitly to another site in $\mathcal{S}_G$.

We notice, that, in a rule r, if the left hand side is a fully specified site graph, then the right hand side is a fully specified site graph as well.

Now we are ready to define of the semantics of Kappa.

Definition 2.15 The *set of states* $\mathcal{Q}$ is the set of $\approx$-equivalence class of fully specified site graphs.

Definition 2.16 The *set of transition labels* $\mathcal{L}$ is the set of the pairs (r, C) where r is a rule and C is an $\approx$-equivalence class of embeddings $h_{L'}$ such that $dom(h_{L'}) = \text{lhs}(r)$.

Definition 2.17 Given two states $q, q' \in \mathcal{Q}$ and a label $(r, C) \in \mathcal{L}$, there is a *transition* between the state q and the state q' with label (r, C) if and only if there exists a refinement $(r', r'', h_{L'}, h_{R'})$ such that: (i) $r = r'$; (ii) $\text{lhs}(r'') \in q$; (iii) $\text{rhs}(r'') \in q'$; (iv) and $h_{L'} \in C$.

In such a case, we write: $q \xrightarrow{(r,C)} q'$.

Such a transition is given a rate that is denoted as $\text{RATE}(r, C)$, and that is defined as follows:

$$\text{RATE}(r, C) \stackrel{\Delta}{=} \gamma_{\mathcal{M}}(r) Card(\{\phi h_{L'} \mid \phi \textit{ automorphism of } im(h_{L'})\}).$$

In Def. 2.17, we observe that the quantity $\text{RATE}(r, C)$ does not depend on the choice of the embedding $h_{L'}$ in C.

The stochastic semantics of a model can be easily defined, as the distribution of traces, or as a continuous-time Markov chain that is induced by this weighted labelled transition system [17].

3 Groups of symmetries over site graphs

3.1 Elements of group theory

Group theory offers convenient ways to formalise symmetries among mathematical structures.

Definition 3.1 A *group* is a pair $(\mathbb{G}, \circ)$ where $\mathbb{G}$ is a set of elements and $\circ$ is an infix associative binary operator over the set $\mathbb{G}$, such that: (i) (identity element) there exists an element $\varepsilon_{\mathbb{G}} \in \mathbb{G}$ which satisfies, $a \circ \varepsilon_{\mathbb{G}} = a$ and $\varepsilon_{\mathbb{G}} \circ a = a$; (ii) (inverse) for any element $a \in \mathbb{G}$, there exists an element $a^{-1} \in \mathbb{G}$ such that $a \circ a^{-1} = \varepsilon_{\mathbb{G}}$ and $a^{-1} \circ a = \varepsilon_{\mathbb{G}}$.

In a group $\mathbb{G}$, there is only one identity element that we denote by $\varepsilon_{\mathbb{G}}$.

Groups are not necessarily commutative. Intuitively, an element of the group can be seen as a sequence of atomic elements, modulo the fact that an element and its inverse cancel each other when they have a consecutive position in the sequence. The identity element can be then seen as the empty sequence.

The action of symmetries over a set of elements can be described by the means of a group action.

Definition 3.2 A *group action* is a triple $((\mathbb{G}, \circ), X, .)$ where $(\mathbb{G}, \circ)$ is a group, X is a set, and $\cdot$ is an infix operator from $\mathbb{G} \times X$ into X such that, for any element $x \in X$ and any elements $a, b \in \mathbb{G}$: (i) $\varepsilon_{\mathbb{G}}.x = x$; (ii) $(a \circ b).x = a.(b.x)$.

3.2 Groups of symmetries

Now we introduce our framework to describe sets of symmetries over site graphs. Our goal is to provide a definition of symmetries which enables to lift the action of symmetries over the basic elements of the semantics, namely fully specified site graphs, weak embeddings, embeddings, rules, and refinements. To achieve this goal, we require that each symmetry which can be applied with the image $im(h)$ of a weak embedding h can be restricted to a symmetry which can be applied to the domain $dom(h)$ of the weak embedding h, hence providing a weak embedding between the symmetric of the domain $dom(h)$ and the symmetric of the image $im(h)$ of the embedding h.

Definition 3.3 A *set of symmetries* $\mathbb{G}$ over site graphs is defined as a quadruple $((\mathbb{G}_E, \circ_E), .\,, .', .'')$, made of a family of finite groups $(\mathbb{G}_E, \circ_E)$ indexed by site graphs, and three functions $.$, $.'$, and $.''$, where:

- the function $.$ maps each pair (σ_E, E) such that E is a site graph and σ_E is an element of $\mathbb{G}_E$ to a site graph, that is written $\sigma_E.E$;
- the function $.'$ maps each pair (f, σ_F) such that f is a weak embedding between two site graphs E and F, and σ_F is an element of $\mathbb{G}_F$ to an element in $\mathbb{G}_E$, that is written $f.'\sigma_F$;
- the function $.''$ maps each pair (σ_F, f) such that f is a weak embedding between two site graphs E and F, and σ_F is an element of $\mathbb{G}_F$ to a weak embedding $\sigma_F.''f$ between the site graph $(f.'\sigma_F).E$ and the site graph $\sigma_F.F$;

$$\begin{tikzcd} E \arrow[r, tail] \arrow[d, squiggly, "f.'\sigma_F"'] & F \arrow[d, squiggly, "\sigma_F"] \\ (f.'\sigma_F).E \arrow[r, tail, "\sigma_F.''f"] & \sigma_F.F \end{tikzcd}$$

such that, for any site graph E and any element $\sigma_E \in \mathbb{G}_E$, the following properties are satisfied:

(i) the groups $(\mathbb{G}_E, \circ_E)$ and $(\mathbb{G}_{\sigma_E.E}, \circ_{\sigma_E.E})$ are the same;

(ii) the function mapping each pair $(\sigma'_E, E') \in \mathbb{G}_E \times \{\sigma.E \mid \sigma \in \mathbb{G}_E\}$ to the site graph $\sigma'_E.E'$, is a group action;

Let us consider $\mathbb{G}$ a set of symmetries. We say that two site graphs G and G' are symmetric, and we write $G \approx_{\mathbb{G}} G'$ if and only if there exists a symmetry $\sigma \in \mathbb{G}_G$ such that $G' = \sigma.G$. This is an equivalence relation. Moreover, by Def. 3.3.i, the same symmetries can be applied with a pair of symmetric site graphs.

Given a weak embedding f and a symmetry $\sigma \in \mathbb{G}_{im(f)}$. The symmetry σ can be applied to the image of the weak embedding f and the result is written as $\sigma.(im(f))$; the permutation σ can also be restricted to the domain of f, this restriction is written $f.'\sigma$. Moreover, the weak embedding $\sigma.''f$ is a weak embedding between the site graph $(f.'\sigma).(dom(f))$ and the site graph $\sigma.(im(f))$.

There are several ways to restrict a symmetry to the domain of a weak embedding, because a symmetry can be described as the product of two others or because the weak embedding can be described as the composition of two others. In such a case, it is important to get the same result. Thus, we need to strengthen Def. 3.3, which is the purpose of the two following definitions.

Definition 3.4 Let $\mathbb{G}$ be a set of symmetries. We say that the *symmetries* in $\mathbb{G}$ *distribute over the composition of weak embeddings*, if and only if, for any two weak embeddings f and g that compose and any symmetry $\sigma \in \mathbb{G}_{im(g)}$, the following constraints hold: (i) $i_{im(g)}.'\sigma = \sigma$; (ii) $\sigma.''i_{im(g)} = i_{\sigma.(im(g))}$; (iii) $(gf).'\sigma = f.'(g.'\sigma)$; (iv) and $\sigma.''(gf) = (\sigma.''g)((g.'\sigma).''f)$.

Definition 3.5 Let $\mathbb{G}$ be a set of symmetries. We say that *restriction of the symmetries* in $\mathbb{G}$ *to the domain of weak embeddings distributes over the product between symmetries*, if and only if, for any weak embedding f and any elements $\sigma \in \mathbb{G}_{im(f)}$ and $\sigma' \in \mathbb{G}_{im(f)}$, the following constraints are satisfied: (i) the function mapping each pair (σ'', f'') such that $\sigma'' \in \mathbb{G}_{im(f)}$ and f'' is a weak embedding such that $im(f'') \approx_{\mathbb{G}} im(f)$, to the weak embedding $\sigma''.''f''$, is a group action; (ii) $f.'\varepsilon_{im(f)} = \varepsilon_{dom(f)}$, where $\varepsilon_{im(f)}$ (resp. $\varepsilon_{dom(f)}$) is the identity element of the group $\mathbb{G}_{im(f)}$ (resp. $\mathbb{G}_{dom(f)}$); (iii) $f.'(\sigma' \circ_{im(f)} \sigma) = ((\sigma.''f).'\sigma') \circ_{dom(f)} (f.'\sigma)$.

Let us explain Def. 3.4 in more details. The first constraint ensures that restricting a given symmetry to the domain of an identity embedding does not modify the symmetry. The second constraint ensures that the symmetric of an identity embedding over a given site graph is the identity function over the symmetric of the same site graph. The rest of Def. 3.4 deals with the composition of weak embeddings. There are two ways to restrict a symmetry σ to the domain of a composition gf of two weak embeddings g and f. In one step, we get the symmetry $(gf).'\sigma$ and the image of the weak embedding gf by the symmetry σ is the weak embedding $\sigma.''(gf)$. In two steps, firstly we restrict the symmetry σ to the domain of g, we get the symmetry $g.'\sigma$ and the image of the weak embedding g is defined as $\sigma.''g$. Then, we restrict the symmetry $g.'\sigma$ to the domain of f, and get the symmetry $f.'(g.'\sigma)$ and the image of the weak embedding f is given by $(g.'\sigma).''f$. Thus, whenever the symmetries distributes over the composition of weak embeddings, we get the same result with both ways. Def. 3.5 works the same way. It ensures that: (i) identity symmetries act neutrally; (ii) and whenever the restriction of the symmetries in $\mathbb{G}$ to the domain of weak embeddings distributes over the product between symmetries, one can compute the restriction of the composition of symmetries to the domain of a weak embedding, in one or two steps, and get the same result.

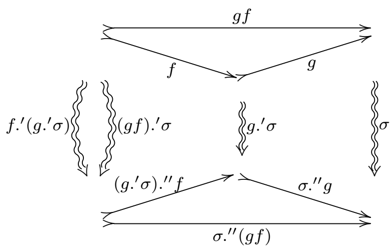


Fig. 6: The symmetric of a composition of embeddings

Since fully specified site graphs are key elements in the definition of the states of the semantics. We have to assume that the symmetric of a fully specified site graph is a fully specified site graph which is the goal of the following definition.

Definition 3.6 Let $\mathbb{G}$ be a set of symmetries. We say that *the symmetries* in $\mathbb{G}$ *preserve fully specified site graphs*, if and only if, for any fully specified site graph E and any symmetry $E \in \mathbb{G}_E$, the site graph $\sigma_E.E$ is fully specified as well.

Embeddings are crucial in the definition of computation steps, and we have to assume that the symmetric of an embedding is an embedding as well.

Definition 3.7 Let $\mathbb{G}$ be a set of symmetries. We say that *the symmetries* in $\mathbb{G}$ *preserve embeddings*, if and only if, for any embedding f and any symmetry $\sigma_{im(f)} \in \mathbb{G}_{im(f)}$, the weak embedding $\sigma_{im(f)}.''f$ is an embedding as well.

We can simultaneously apply a symmetry to the left hand side and a symmetry to the right of side of a partial embedding ϕ, providing that these symmetries have the same restriction over the domain of the partial embedding. Moreover, we need the set of pairs that can be applied to a partial embedding to form a group. That is why we assume in the following definition, that the set of pairs of symmetries that can be applied to a given partial embedding is stable upon pairwise product.

Definition 3.8 Let $\mathbb{G}$ be a set of symmetries. We say that *the pairs of symmetries* in $\mathbb{G}$ *over partial embeddings compose pairwise*, if and only if, for any partial embedding $L \xleftarrow{h_L} D \xrightarrow{h_R} R$, any symmetries $\sigma_L \in \mathbb{G}_L$, $\sigma'_L \in \mathbb{G}_L$, $\sigma_R \in \mathbb{G}_R$, and $\sigma'_R \in \mathbb{G}_R$ such that $h_L.'\sigma_L = h_R.'\sigma_R$ and $h_L.'\sigma'_L = h_R.'\sigma'_R$, we have $h_L.'(\sigma'_L \circ_L \sigma_L) = h_R.'(\sigma'_R \circ_R \sigma_R)$.

Lastly, we have to assume that the symmetric of a rule is a rule and that symmetries preserve the existence of refinements. This is the purpose of the next two definitions.

Definition 3.9 Let $\mathbb{G}$ be a set of symmetries. We say that *the symmetries* in $\mathbb{G}$ *preserve rules*, if and only if, for any rule $L \xleftarrow{h_L} D \xrightarrow{h_R} R$ and any pair of symmetries $(\sigma_L, \sigma_R) \in \mathbb{G}_L \times \mathbb{G}_R$ such that $h_L.\sigma_L = h_R.\sigma_R$, the weak partial embedding $\sigma_L.L \xleftarrow{\sigma_L.h_L} (h_R.\sigma_R).D \xrightarrow{\sigma_R.h_R} \sigma_R.R$ is also a rule.

Definition 3.10 Let $\mathbb{G}$ be a set of symmetries that preserve rules, we say that *the symmetries* in $\mathbb{G}$ *preserve the existence of refinements* as well, if and only if, for any rule $L \xleftarrow{h_L} D \xrightarrow{h_R} R$ (that we denote by r), for any embedding $h_{L'}$ between the site graphs L and a site graph L', and for any pair of symmetries $(\sigma_{L'}, \sigma_R) \in \mathbb{G}^{L'} \times \mathbb{G}^R$ such that: (i) $(h_{L'}h_L).\sigma_{L'} = h_R.\sigma_R$, (ii) there exist a rule r' and an embedding $h_{R'}$ such that the tuple $(r, r', h_{L'}, h_{R'})$ is a refinement, there exist a rule r'_σ and an embedding $h_{R'_\sigma}$ such that the tuple $(r_\sigma, r'_\sigma, \sigma_{L'}.h_{L'}, h_{R'_\sigma})$ is a refinement as well, where the r_σ is defined as the rule $(h_{L'}.\sigma_{L'}).L \xleftarrow{(h_{L'}.\sigma_{L'}).h_L} (h_R.\sigma_R).D \xrightarrow{\sigma_R.h_R} \sigma_R.R$.

We are now ready to define the valid sets of symmetries as those which satisfy the additional assumptions that we have given in Definitions 3.4–3.10.

Definition 3.11 We say that a set of symmetries $\mathbb{G}$ is *valid*, if and only if, (i) the symmetries in $\mathbb{G}$ distribute over the composition of weak embeddings; (ii) the restriction of the symmetries in $\mathbb{G}$ to the domain of weak embeddings distributes over the product between symmetries; (iii) the symmetries in $\mathbb{G}$ preserve fully specified site graphs; (iv) the symmetries in $\mathbb{G}$ preserve embeddings; (v) the pairs of symmetries in $\mathbb{G}$ over partial embeddings compose pairwise; (vi) the symmetries in $\mathbb{G}$ preserve rules; (vii) the symmetries in $\mathbb{G}$ preserve the existence of refinements.

Now we give examples of valid sets of symmetries over site graphs.

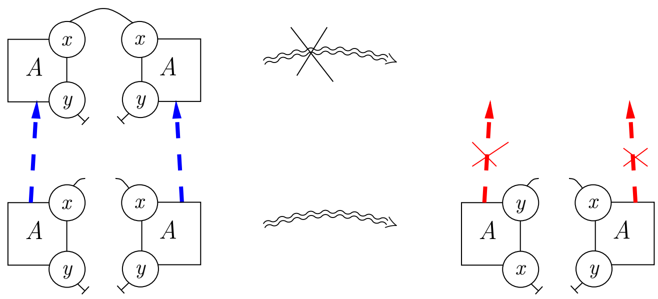


Fig. 7. A symmetry which cannot be extended to the image of an embedding (we consider as symmetries the ones of Exa. 3.13 that consist in applying the same site permutations to each agent with the same type in a same connected component).

Example 3.12 We consider an example in which symmetries consist of permuting the sites of each agent independently.

We assume that we are given a group of bijections $S(A)$ of $\Sigma_{ag\text{-}st}(A)$ for any agent type $A \in \Sigma_{ag}$. We define a set of symmetries $\mathbb{G}$ as follows: for any site graph F, let $\mathbb{G}_F$ be the set of the functions which map each agent $n \in \mathcal{A}_F$ into a bijection in $S(type_F(n))$ (product is defined component-wise). The symmetric $\sigma.F$ of a site graph F by a symmetry $\sigma \in \mathbb{G}_F$ is defined by injectively renaming the sites (n,i) of each agent n in F into the sites $(n,(\sigma(n)(i)))$. Now we consider a weak embedding h between two site graphs E and F, that is induced by a function between the agents $\mathcal{A}_E$ of E and the agents $\mathcal{A}_F$ of F. The restriction $h.'\sigma$ of a symmetry $\sigma \in \mathbb{G}_F$ of the site graph F to the site graph E, is defined as the function mapping each agent $n \in \mathcal{A}_E$ to the bijection $\sigma(h(n))$. The function h also induces a weak embedding between the site graph $(h.'\sigma).E$ and the site graph $\sigma.F$, the symmetric of h with respect to σ is defined as this weak embedding.

In our running example, we can define $S(A)$ as the set of two elements $[x \mapsto x, y \mapsto y]$ and $[x \mapsto y, y \mapsto x]$. The three site graphs in Fig. 1, are the symmetric of one another with respect to this set of symmetries.

One can define new valid sets of symmetries as subsets of other ones. Let us consider $((\mathbb{G}_E, \circ_E), ., .', .'')$ a valid set of symmetries. We assume that for any site graph G, we are given $\mathbb{G}^{'}_G$ a subset of $\mathbb{G}_G$ such that for any weak embedding f between E and F, and any symmetry $\sigma, \sigma' \in \mathbb{G}^{'}_F$: (i) $\mathbb{G}^{'}_F = \mathbb{G}^{'}_{\sigma.F}$; (ii) $\sigma \circ \sigma' \in \mathbb{G}^{'}_F$; (iii) $f.'\sigma \in \mathbb{G}^{'}_E$. Under these assumptions the tuple $((\mathbb{G}_E, \tilde{\circ_E}), \tilde{.}, \tilde{.'}, \tilde{.''})$ is a valid set of symmetries as well (where for any operator $\Box$, $\tilde{\Box}$ denotes the restriction of the operator $\Box$ to the correct domain).

Example 3.13 We can restrict the set of symmetries of Exa. 3.12 so that, we cannot apply different site permutations to two agents of same type in a same connected component. With this subset of symmetries, only the first and the third site graphs in Fig. 1 are the symmetric of one another. The second site graph cannot be obtained from the first one, since it would require to swap the site of the first agent, but not the ones of the second agent.

Example 3.14 It is also possible to consider only the symmetries in which we cannot apply different permutations of sites with two agents of same type in the whole site graph.

In our framework, it is always possible to restrict a symmetry that can be applied to the image of an embedding to its domain. Yet the opposite is not always true. We call $\mathbb{G}$-forward compatible the embeddings f such that any symmetry in $\mathbb{G}_{dom(f)}$ that can be applied to the domain of the embedding f, can be extended to a symmetry which can be applied to the image of f. This notion is formalised in the following definition.

Definition 3.15 Let $\mathbb{G}$ be a set of symmetries over site graphs. Let h be an embedding. We say that h is $\mathbb{G}$*-forward compatible* if and only if for any symmetry $\sigma \in \mathbb{G}_{dom(h)}$, there exists a symmetry $\sigma' \in \mathbb{G}_{im(h)}$ such that $\sigma = h.'\sigma'$.

We notice that any embedding is forward compatible with the set of symmetries that is defined in Exa. 3.12. But in general, if an embedding h is $\mathbb{G}$-forward compatible for a given set of symmetries $\mathbb{G}$ over site graphs, h is not necessarily $\mathbb{G}'$-forward compatible for a subset $\mathbb{G}'$ of $\mathbb{G}$.

Example 3.16 We apply the set of symmetries of Exa. 3.13 in our running example. Thus, for each connected component, we can either swap the sites of each agent, or do nothing. We give in Fig. 7 an example of a symmetry over the domain of an embedding which cannot be extended to the image of this embedding. The point is that there are two connected components in the domain of the embedding which become connected in its image. We can apply different permutations to the two agents of the domain of the embedding, since they are not connected, but we cannot do it in the image, which prevents us from extending this symmetry to the image of the embedding.

3.3 Induced group actions

A valid set of symmetries induces group actions over the basic elements of the semantics of Kappa. We are going to give the list of them in this subsection. We introduce a valid set of symmetries $\mathbb{G}$ that we will use all along the subsection.

Def. 3.6 ensures that the image of a fully specified site graph E by a symmetry in $\mathbb{G}_E$ is a fully specified site graph. Thus, the group action of symmetries over site graphs induces another group action over fully specified site graphs, as written in the following proposition.

Proposition 3.17 (. . . over fully specified site graphs) *For a given fully specified site graph E, the function mapping each pair $(\sigma'_E, E') \in \mathbb{G}_E \times \{\sigma.E \mid \sigma \in \mathbb{G}_E\}$ to the fully specified site graph $\sigma'_E.E'$, is a group action.*

Def. 3.7 ensures that the image of an embedding f by a symmetry in $\mathbb{G}_{im(f)}$ is an embedding. Thus, the group action of symmetries over weak embeddings can be restricted to embeddings, as written in the the following proposition.

Proposition 3.18 (. . . over embeddings) *For a given embedding h, the function mapping each pair $(\sigma'_{im(f)}, f') \in \mathbb{G}_{im(f)} \times \{\sigma.''f \mid \sigma \in \mathbb{G}_{im(f)}\}$ to the embedding $\sigma'_{im(f)}.f'$, is a group action.*

The image of an isomorphism ϕ by a symmetry σ is also an isomorphism (by Def. 3.4, the inverse of the embedding $\sigma.''\phi$ is the embedding $(\phi.'\sigma).''\phi^{-1}$). Thus, we can restrict the action of symmetries over embeddings to isomorphisms, as expressed in the following proposition.

Proposition 3.19 (... over isomorphisms) *For a given isomorphism ϕ, the function mapping each pair $(\sigma'_{im(\phi)}, \phi') \in \mathbb{G}_{im(\phi)} \times \{\sigma.''\phi \mid \sigma \in \mathbb{G}_{im(\phi)}\}$ to the isomorphism $\sigma'_{im(\phi)}.''\phi'$, is a group action.*

Applying symmetries to a rule consists in applying symmetries simultaneously to its left hand side and symmetries to its right hand side, provided that these symmetries agree on the symmetries to be applied to the domain on the rule. We define the set of pairs of symmetries that can be applied to a rule as follows.

Definition 3.20 Given a rule $r : L \xleftarrow{h_L} D \xrightarrow{h_R} R$, we define *the set* $\mathbb{G}_r$ *of the symmetries over the rule* r as the set of pairs (σ_L, σ_R) in $\mathbb{G}_{\mathrm{lhs}(r)} \times \mathbb{G}_{\mathrm{rhs}(r)}$ such that $h_L.'\sigma_L = h_R.'\sigma_R$.

For any rule r, it follows from Def. 3.8 that the set $\mathbb{G}_r$ is stable upon pairwise product. Since, moreover, both groups $\mathbb{G}_{\mathrm{lhs}}$ and $\mathbb{G}_{\mathrm{rhs}}$ are finite groups, the set $\mathbb{G}_r$ is a finite group as well.

Definition 3.21 We define *the symmetric of a rule* $r : L \xleftarrow{h_L} D \xrightarrow{h_R} R$ by a pair of symmetries $(\sigma_L, \sigma_R) \in \mathbb{G}_r$ as the following partial embedding:

$$\sigma_L.L \xleftarrow{\sigma_L.''h_L} (h_L.'\sigma_L).D \xrightarrow{\sigma_R.''h_R} \sigma_R.R$$

and we denote it as $(\sigma_L, \sigma_R).r$.

It follows from Def. 3.9 that the symmetric of a rule is indeed a rule. We can then construct a group action over rules, as a restriction of the pairwise group action over their left and right embeddings.

Proposition 3.22 (... over rules) *For a given rule r, the function mapping each pair (σ', r') such that $\sigma' \in \mathbb{G}_r$ and r' is a rule in the set $\{\sigma.r \mid \sigma \in \mathbb{G}_r\}$, to the rule $\sigma'.r'$, is a group action.*

By Def. 3.4 and Def. 3.20, for any refinement $(r, r', h_{L'}, h_{R'})$ and any two symmetries $\sigma_{L'} \in \mathbb{G}_{\mathrm{lhs}(r')}$ and $\sigma_{R'} \in \mathbb{G}_{\mathrm{rhs}(r')}$ such that $(\sigma_{L'}, \sigma_{R'}) \in \mathbb{G}_{r'}$, the pair of symmetries $(h_{L'}.'\sigma_{L'}, h_{R'}.'\sigma_{R'})$ belongs to the group $\mathbb{G}_r$ (this result is not direct and more details about the proof can be found in [14]). Thanks to this property, we can safely apply a pair of symmetries to a refinement, as formalised in the following definition:

Definition 3.23 Given a refinement $(r, r', h_{L'}, h_{R'})$ and $\sigma_{r'} \triangleq (\sigma_{L'}, \sigma_{R'})$ a symmetry in $\mathbb{G}_{r'}$, we call *the symmetric of the refinement* $(r, r', h_{L'}, h_{R'})$ by the symmetry $\sigma_{r'}$ the tuple $(\sigma_r.r, \sigma_{r'}.r', \sigma_{L'}.''h_{L'}, \sigma_{R'}.''h_{R'})$, where the symmetry $\sigma_r \in \mathbb{G}_r$ over the rule r is defined as the pair $(h_{L'}.'\sigma_{L'}, h_{R'}.'\sigma_{h_{R'}})$.

We denote this tuple as $\sigma_{r'}.(r, r', h_{L'}, h_{R'})$.

Importantly, the symmetric of a refinement is a refinement as well, and applying symmetries to refinements is a group action. These results are formalised in the following theorems.

Theorem 3.24 (refinements preservation) *Given a refinement* $(r, r', h_{L'}, h_{R'})$ *and* $\sigma_{r'}$ *a symmetry in* $\mathbb{G}_{r'}$*, the symmetric* $\sigma_{r'}.(r, r', h_{L'}, h_{R'})$ *of the refinement* $(r, r', h_{L'}, h_{R'})$ *by the symmetry* $\sigma_{r'}$*, is a refinement as well.*

Theorem 3.25 (induced group action over refinements) *For a given rule* r*, the function mapping each pair* $(\sigma', (r', r'', h_{L''}, h_{R''}))$ *such that* $\sigma' \in \mathbb{G}_r$*,* r'' *is a rule in the set* $\{\sigma.r \mid \sigma \in \mathbb{G}_r\}$*, and the tuple* $(r', r'', h_{L''}, h_{R''})$ *is a refinement, to the refinement* $\sigma'.(r', r'', h_{L'}, h_{R'})$*, is a group action.*

The proofs of Thm. 3.24 and Thm. 3.25 can be found in [14]. Intuitively, these theorems state that the symmetries of a valid set of symmetries are compatible with the operational semantics of Kappa.

4 Symmetries in rule-base models

In this section, we define symmetric models as those which are invariant by a set of symmetries and investigate their quantitative properties.

4.1 Symmetries among a set of rules

So as to define symmetric models properly, we have to be careful with rule isomorphisms. Indeed, two isomorphic rules induce the same transformations over site graphs. Thus, we do not want to distinguish them. In order to solve this issue, we consider rules in models modulo isomorphism.

Firstly, we give one definition (there are several equivalent ones) of isomorphism between rules.

Definition 4.1 Two rules r and r' are *isomorphic* if and only if there exist four embeddings h_L, h_R, $h_{L'}$, and $h_{R'}$ such that both tuples $(r, r', h_{L'}, h_{R'})$ and (r', r, h_L, h_R) are refinements.

In such a case, we write $r \approx r'$.

We notice that the relation $\approx$ is an equivalence relation over rules. Moreover, in Def. 4.1, when the rule r and r' are isomorphic, then each of the four embeddings is an isomorphism between site graphs.

Now we can define the equivalence among two models: two models are equivalent if they have the same overall rate for each isomorphism class of rules.

Definition 4.2 Two models $\mathcal{M}$ and $\mathcal{M}'$ are said $\approx$*-equivalent* if and only if

$$\sum\nolimits_{r' \in [r]_{\approx}} k_{\mathcal{M}}(r') = \sum\nolimits_{r' \in [r]_{\approx}} k_{\mathcal{M}'}(r'),$$

for any rule r.

Then, we introduce an idempotent operator over models which symmetrises each models by replacing each rule with the set of its symmetrics, correcting the rate according to the loss/gain of symmetries and to the number of symmetric rules.

Definition 4.3 Let $\mathbb{G}$ be a valid set of symmetries over site graphs. We introduce the *symmetrisation* operator $sym_{\mathbb{G}}$, which maps each model $\mathcal{M}$ to the model

$\Rsh$	r_1	r_2	r_3
r_1	$\frac{k_1}{4}$		
r_2		$\frac{k_2}{4}$	
r_3			$\frac{k_3}{4}$

(a) No site permutation.

$\Rsh$	r_1	r_2	r_3
r_1			$\frac{k_1}{8}$
r_2			$\frac{k_2}{8}$
r_3		$\frac{k_3}{2}$	

(b) Permutation of the sites of the agent on the top.

$\Rsh$	r_1	r_2	r_3
r_1			$\frac{k_1}{8}$
r_2			$\frac{k_2}{8}$
r_3	$\frac{k_3}{2}$		

(c) Permutation of the sites of the agent on the bottom.

$\Rsh$	r_1	r_2	r_3
r_1		$\frac{k_1}{4}$	
r_2	$\frac{k_2}{4}$		
r_3			$\frac{k_3}{4}$

(d) Permutation of the sites of both agents.

Fig. 8. Action of the four symmetries over the rules in Fig. 4.

$sym_{\mathbb{G}}(\mathcal{M})$ that is defined by the following equation:

$$k_{sym_{\mathbb{G}}(\mathcal{M})}(r) \triangleq \frac{[\mathrm{lhs}(r), \mathrm{lhs}(r)]}{Card(\mathbb{G}_r)} \sum_{\sigma \in \mathbb{G}_r} \frac{k_{\mathcal{M}}(\sigma.r)}{[\mathrm{lhs}(\sigma.r), \mathrm{lhs}(\sigma.r)]},$$

for any rule r.

Symmetric models are those which are equivalent to their symmetrisation, as defined as follows.

Definition 4.4 We say that a model is *symmetric* with respect to a valid set of symmetries $\mathbb{G}$ if and only if the model $\mathcal{M}$ and the model $sym_{\mathbb{G}}(\mathcal{M})$ are $\approx$-equivalent.

Example 4.5 We wonder whether or not, the model that we had given in Fig. 4 is symmetric with respect to the set of symmetries that we had defined in Exa. 3.12. Following Def. 4.4, we apply each pair of permutations to the rules. There are only four pair of permutations, according to whether or not, we swap the sites x and y of the agent on the top, and whether of nor, we swap the sites x and y of the agent on the bottom. The action of these symmetries to the rules are summarised in Fig. 8. We have divided each rate by 4 and taken into account the gain/loss of symmetries in the left hand sides of rules. It follows that the sites x and y are symmetric in the agents of type A if and only if the following system of equations is satisfied:

$$\left\{ k_1 = \frac{k_1}{4} + \frac{k_3}{2} + \frac{k_2}{4},\ k_2 = \frac{k_2}{4} + \frac{k_3}{2} + \frac{k_1}{4},\ k_3 = \frac{k_3}{4} + \frac{k_1}{8} + \frac{k_2}{8} + \frac{k_1}{8} + \frac{k_2}{8} + \frac{k_3}{4}. \right.$$

That is to say that the sites x and y are symmetric in the agents of type A if and only if the rates k_1, k_2, and k_3 are equal.

4.2 Bisimulations induced by symmetries

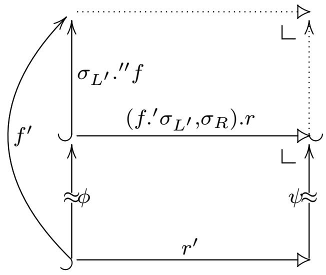
Under appropriate assumptions, symmetries among models can be used to lump the set of states of the population semantics.

Firstly, we define the notion of states and transition labels modulo symmetries.

Definition 4.6 Let $\mathbb{G}$ be a valid set of symmetries over site graphs. Let q, q' be two states in $\mathcal{Q}$. We say that the states q and q' are *symmetric* with respect to the symmetries in $\mathbb{G}$ and we write $q \approx_{\mathbb{G}} q'$ if and only if there exist two site graphs $M \in q$ and $M' \in q'$, respectively in the $\approx$-equivalence class q and in the $\approx$-equivalence class q' such that $M \approx_{\mathbb{G}} M'$.

The relation $\approx_{\mathbb{G}}$ is an equivalence relation over the states of the population semantics.

Definition 4.7 Let $\mathbb{G}$ be a valid set of symmetries over site graphs. Let (r, C) and (r', C') be two transition labels in $\mathcal{L}$. We say that the labels (r, C) and (r', C') are *symmetric* with respect to the symmetries in $\mathbb{G}$ and we write $(r, C) \approx_{\mathbb{G}} (r, C')$, if and only if, there exist an embedding f in C, an embedding f' in C', a pair of symmetries $(\sigma_{L'}, \sigma_R)$ in the set $\mathbb{G}_{im(f)} \times \mathbb{G}_{\mathrm{rhs}(r)}$ such that $(f.'\sigma_{L'}, \sigma_R) \in \mathbb{G}_r$ and two isomorphisms ϕ and ψ such that: (i) $(r', (f.'\sigma_{L'}, \sigma_R).r, \phi, \psi)$ is a refinement; (ii) $f' = (\sigma_{L'}.''f)\phi$.



The relation $\approx_{\mathbb{G}}$ is an equivalence relation over the transition labels of the population semantics.

We notice that transition labels are quotiented according to two criteria. On the first hand, labels are seen modulo symmetries. On the second hand, since we cannot detect whether a model is symmetric without gathering the rules in isomorphic classes (see Def. 4.3), we consider rules modulo isomorphism.

Now we can quotient the population semantics by symmetries and see whether this quotient induces bisimulations, or not. For this we introduce the notion of flow between set of states in the following definition.

Definition 4.8 We define the *flow* $\textsc{flow}_\omega(X, Y, X')$ from a set of states $X \subseteq \mathcal{Q}$ into a set of states $X' \subseteq \mathcal{Q}$ via the transitions with labels in the set $Y \subseteq \mathcal{L}$, weighted by a reward function ω between the sets $\mathcal{Q}$ and $\mathbb{R}^+$, as follows:

$$\textsc{flow}_\omega\left(X, Y, X'\right) \triangleq \sum\nolimits_{q\in X, q'\in X', \lambda\in Y \textit{ such that } q \xrightarrow{\lambda} q'} \omega(q)\mathrm{RATE}(\lambda).$$

We now state the third theorem of the paper, as follows.

Theorem 4.9 (induced bisimulations) *Let $\mathcal{M}$ be a model that is symmetric with respect to a valid set of symmetries $\mathbb{G}$. The following two properties hold:*

(i) (forward bisimulation)*If, for any refinement $(r, (f, g), h_{L'}, h_{R'})$ of a rule r such that $k_{\mathcal{M}}(r) > 0$, the embedding g is $\mathbb{G}$-forward compatible, then, for any three states $q, q', q'' \in \mathcal{Q}$ such that $q \approx_{\mathbb{G}} q'$ and any label $\lambda \in \mathcal{L}$, we have:*

$$\textsc{flow}_\omega\left(\{q\}, [\lambda]_{\approx_{\mathbb{G}}}, [q'']_{\approx_{\mathbb{G}}}\right) = \textsc{flow}_\omega\left(\{q'\}, [\lambda]_{\approx_{\mathbb{G}}}, [q'']_{\approx_{\mathbb{G}}}\right),$$

where ω maps each state to the value 1.

(ii) (back and forth bisimulation) *If, for any refinement $(r, (f, g), h_{L'}, h_{R'})$ of a rule r such that $k_{\mathcal{M}}(r) > 0$, both embeddings f and g are $\mathbb{G}$-forward compatible, then, for any three states $q, q', q'' \in \mathcal{Q}$ such that $q' \approx_{\mathbb{G}} q''$ and any label $\lambda \in \mathcal{L}$,* *we have:*

$$\omega(q'')\text{FLOW}_\omega\left([q]_{\approx_{\mathbb{G}}},[\lambda]_{\approx_{\mathbb{G}}},\{q'\}\right)=\omega(q')\text{FLOW}_\omega\left([q]_{\approx_{\mathbb{G}}},[\lambda]_{\approx_{\mathbb{G}}},\{q''\}\right),$$

where ω maps each state $[M]_\approx$ to the inverse of the number of automorphisms $[M,M]$ in the site graphs in the $\approx$-equivalence class $[M]_\approx$.

As discussed in [16], a forward bismulation induces a strong lumping of the states of the underlying Markov chain: whatever the initial distribution of states is, one can quotient the states of the systems modulo the equivalence relation $\approx_{\mathbb{G}}$ and the quotient is still a Markov chain. A back and forth bisimulation ensures the existence of statistical invariants: if, at a given time t, for every states q, the conditional probability of being in the state q at time t knowing that we are in its $\approx_{\mathbb{G}}$-equivalence class $[q]_{\approx_{\mathbb{G}}}$ is equal to the inverse of the sum $\sum_{q'\in[q]_{\approx_{\mathbb{G}}}}\frac{[M(q),M(q)]}{[M(q'),M(q')]}$, where $M(q)$ maps each state to an arbitrary fully specified site graph in the $\approx$-equivalence class q, then this is true at any time. If it is not the case and if additionally the set of states is directed (that is to say that there exists a state that is reachable from any other state in zero, one, or several transition steps), then, as it is explained in [22], at the limit when the time goes towards the infinity, the conditional probabilities of being in a state q, knowing that we are in its $\approx_{\mathbb{G}}$-equivalence class $[q]_{\approx_{\mathbb{G}}}$, converges towards the inverse of the sum $\sum_{q'\in[q]_{\approx_{\mathbb{G}}}}\frac{[M(q),M(q)]}{[M(q'),M(q')]}$ where $M(q)$ is defined as above.

5 Conclusion

In this paper, we have proposed an algebraic framework for detecting symmetries and lumping the population stochastic semantics accordingly. Our symmetries can also be used to lump the individual symmetries [17] and the differential semantics [11] (we omit the details about it for the sake of conciseness, but more details can be found in [14]). Our framework can also be combined with other model reduction techniques [11,15,17,5] thanks to the techniques which are described in [6].

Our framework captures not only symmetries among sites, but also homogeneous symmetries when the same transformation has to be applied to all the agents of a given type in a same connected component, or in the whole site graph. In these cases, symmetries do not always induce bisimulations. A forward bisimulation is induced whenever any symmetry that can be applied to the domain of a refinement of a rule (in the model) can be extended to the right hand side of this refinement. If additionally, one can extend the symmetries that can be applied to the domain of a refinement of a rule to the left hand side of this refinement as well, then the induced bisimulation is indeed a back and forth bisimulation.

We have not investigated yet which syntactic criteria over the rules would ensure that symmetries over the domain of their refined rules can always be extended to their left hand side and/or their right hand side. It depends on the set of symmetries that is considered. We have not investigated either whether our notion of symmetry is also compatible with the double pushout semantics [12] of Kappa.

In this paper, we have focused on exact symmetries. It would be interesting to propose an approximate notion of symmetries and use bisimulation metrics [18] to compare the distributions of traces of the models which are nearly symmetric.

Acknowledgements. The author would like to deeply thank anonymous reviewers for their insightful comments and their very helpful suggestions to improve the quality of the paper.

References

[1] 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).

[2] Buchholz, P., *Exact and ordinary lumpability of finite Markov chains*, Journal of Applied Probability **31** (1994).

[3] Buchholz, P. and I. Tarasyuk, *Net and algebraic approaches to probabilistic modeling*, in: *Joint Novosibirsk Computing Center and Institute of Informatics Systems Bulletin*, Series Computer Science, 2001.

[4] Camporesi, F. and J. Feret, *Formal reduction of rule-based models*, in: *MFPS 2011*, ENTCS **276C** (2011).

[5] Camporesi, F., J. Feret and J. Hayman, *Context-sensitive flow analyses: a hierarchy of model reductions.*, in: *CMSB 1013*, LNBI **8130** (2013).

[6] Camporesi, F., J. Feret, H. Koeppl and T. Petrov, *Combining Model Reductions*, in: *MFPS 2010*, ENTCS **265** (2010).

[7] Damgaard, T. C., E. Højsgaard and J. Krivine, *Formal cellular machinery*, in: *SASB 2011*, ENTCS **284**, 2011.

[8] Danos, V., J. Feret, W. Fontana, R. Harmer, J. Hayman, J. Krivine, C. Thompson-Walsh and G. Winskel, *Graphs, Rewriting and Pathway Reconstruction for Rule-Based Models*, in: *FSTTCS 2012*, LIPIcs **18**, 2012.

[9] Danos, V., J. Feret, W. Fontana, R. Harmer and J. Krivine, *Rule-based modelling, symmetries, refinements*, in: *FMSB 2008*, LNCS **5054** (2008).

[10] Danos, V., J. Feret, W. Fontana, R. Harmer and J. Krivine, *Rule-Based Modelling and Model Perturbation*, TCSB **5750** (2009), pp. 116–137.

[11] 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: *LICS 2010* (2010).

[12] Danos, V., R. Harmer and G. Winskel, *Constraining rule-based dynamics with types.*, MSCS (2013).

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

[14] Feret, J., *Abstract interpretations of rule-based modelling*, hDR, in preparation.

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

[16] Feret, J., T. Henzinger, H. Koeppl and T. Petrov, *Lumpability abstractions of rule-based systems*, TCS **431** (2012).

[17] Feret, J., H. Koeppl and T. Petrov, *Stochastic fragments: A framework for the exact reduction of the stochastic semantics of rule-based models*, International Journal of Software and Informatics **7** (2014).

[18] Ferns, N., P. Panangaden and D. Precup, *Bisimulation metrics for continuous markov decision processes*, SIAM J. Comput. **40** (2011).

[19] Ferreux, E. and N. Morsli, *Models of population migration in kappa*, intership report, Frontiers in Life Science Bachelor Programme, First year, 2014.

[20] John, M., C. Lhoussaine, J. Niehren and C. Versari, *Biochemical reaction rules with constraints*, in: *ESOP 2011*, LNCS **6602** (2011).

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

[22] Petrov, T., A. Ganguly and H. Koeppl, *Molecular Signaling - Model Decomposition and Stochastic Fragments*, in: *SASB 2011*, ENTCS **284**, 2011.