Rigid Geometric Constraints for Kappa Models
Vincent Danos, Ricardo Honorato-Zimmer, Sebastián Jaramillo-Riveri, Sandro Stucki

To cite this version:

HAL Id: hal-01976400
https://hal.archives-ouvertes.fr/hal-01976400
Submitted on 10 Jan 2019

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.
Rigid Geometric Constraints for Kappa Models

Vincent Danos\textsuperscript{a,1}  Ricardo Honorato-Zimmer\textsuperscript{b,2}  Sebastián Jaramillo-Riveri\textsuperscript{c,3}  Sandro Stucki\textsuperscript{d,4}

\textsuperscript{a} School of Informatics  
University of Edinburgh  
Edinburgh, United Kingdom
\textsuperscript{b} Informatics Life-Sciences Institute  
University of Edinburgh  
Edinburgh, United Kingdom
\textsuperscript{c} SynthSys  
University of Edinburgh  
Edinburgh, United Kingdom
\textsuperscript{d} PPS Laboratory  
Université Paris Diderot – Paris 7  
Paris, France

Abstract

Rule-based modeling languages such as Kappa \cite{11,7} and BNGL \cite{3,2} allow for a concise description of combinatorially complex biochemical processes as well as efficient simulations of the resulting models \cite{8,7}. A key aspect of the rule-based modeling approach is to partially expose the structure of the chemical species involved. However, the above-mentioned languages do not provide means to directly express the three-dimensional geometry of chemical species. As a consequence models typically capture only the network-topological structure of the species involved. For certain biochemical processes, such as the assembly of molecular complexes, in which steric constraints play a key role, it would seem natural to also model the geometric structure of species. We propose an extension to the Kappa modeling language allowing the annotation of the structure of chemical species with three-dimensional geometric information. This naturally introduces rigidity constraints on the species and reduces the state space of the resulting model by excluding species that are not geometrically sound. We show that models extended in this way can still be simulated efficiently, albeit at the cost of a greater number of null-events occurring during the simulation. The geometric constraints introduced by the extension are inherently non-local in that they may entangle the position and orientation of sub-structures at arbitrary distances in large species such as polymers. We give a formal definition of the notion of locality based on the intuition that local modifications should only affect sub-structures within a finite radius around the point where the modification occurred. We show that there are indeed geometrically enhanced Kappa models that are non-local, and conversely, that every local model can be simulated accurately using a finite classical Kappa model at the expense of a possible combinatorial explosion of its rule set. We also give some sufficient conditions for the locality of a model but show that locality is undecidable in general.

Keywords: rule-based modeling, Kappa, geometric constraints, molecular structure, three-dimensional, rigid body
1 Introduction

Rule-based modeling languages such as Kappa [11,7], or BNGL [3,2], allow for a concise description and the efficient simulation of combinatorially complex biochemical processes, such as cellular signaling pathways. In contrast to reaction-based approaches, rule-based modeling frameworks do not rely on the enumeration of all chemical species involved in a process. The combinatorial complexity of a process can easily make this enumeration intractable or even impossible, for example in the case of polymerization reactions, where the number of possible species involved is infinite in principle.

Due to their stability, molecules such as nucleic acids and proteins can be represented as elementary molecular entities, or agents with an interface of sites through which they interact. In this context, sites may represent functional groups within molecules that are responsible for the interactions they participate in. Classic examples are binding sites, phosphorylation sites, methylation sites, etc. In particular, agents may form weak inter-molecular bonds between sites. These bonds are also called links and together with agents they define site graphs representing the network-topological structure of molecular complexes.

In rule-based modeling, reactions are replaced by rules. Unlike reactions, rules operate on patterns in the sub-structure of chemical species rather than on the species as a whole. A single rule can thereby capture a set of multiple chemical reactions, potentially even infinitely many. However, the above-mentioned rule-based languages do not offer direct support for describing the geometric structure of agents and their link structure in three-dimensional space. Hence, steric effects that constrain molecular interactions are not easily translated into rule-based languages, and it is not clear if it is always possible to do so.

In this paper we present geometrically enhanced Kappa, a minimalistic extension to the Kappa language. As in Kappa, chemical species are represented by site graphs and modified through site graph rewriting rules. In addition, our extension allows for the annotation of individual agents, sites and links, with radii, positions and orientations, respectively, which introduces implicit rigidity constraints on the corresponding site graphs. Rewrite rules may include updates to the geometric annotations of a site graph in order to capture conformational changes of a molecular complex.

Our extension is only concerned with geometric realizability or soundness of molecular complexes and not with modeling the evolution of their physical embedding in three-dimensional euclidean space. In particular, we do not track the absolute positions and orientations of agents and therefore can not handle diffusion or collision processes explicitly. Instead we retain the assumption of a well-stirred mixture from classical, geometry-less Kappa. In fact, our framework for geometri-
cally enhanced site graph rewriting contains geometry-less Kappa as a particular subset, and we show that for our restricted set of geometric annotations, simulations remain tractable, and indeed can be implemented using a slightly modified version of the known efficient Kappa simulation algorithm [8].

Geometric constraints can, in principle, relate and propagate the positions and orientations of agents over arbitrary distances, for example in macro-molecular complexes such as polymer chains. Our extension thereby violates a core principle of classical Kappa, namely that rules can only make local changes to the state of a system. To make this idea more precise, we give a formal definition of locality for Kappa models, which captures the intuition that the effects of a locally acting rule on the geometric constraints of a site graph should not extend beyond a fixed, finite radius $\delta$ around the substructure modified by the rule. We show that there are indeed geometrically enhanced Kappa models that are non-local, meaning that no such $\delta$ exists for these models. We also show that, for purely local models, it is always possible to translate a geometrically enhanced model into a geometry-less model, albeit at the cost of increasing the number of rules of the model exponentially. This further exemplifies the importance of non-locality for our extension. Finally, we show that, in general, it is not decidable whether a model is local, and hence whether the said reduction from a geometrically enhanced model into a geometry-less model is possible.

The principal motivation for developing this extension was to find a basic framework to describe processes such as the self-assembly of protein complexes [12,16,18,15], where agents may be thought of as having a fixed-size volume, sites with fixed sets of possible positions with respect to the center of mass of the agent, and links with fixed orientations relative to the interacting agents. This is the case, for instance, in the assembly of the proteasome, whose structure consists of four stacked heptameric rings (see Figure 1). In this protein complex, interactions between the monomers have a particular geometry that we will revisit later.

Finally, we also expect this work to serve as a formal starting point for the development and implementation of more advanced extensions for modeling spatial aspects such as more generalized steric constraints, diffusion and compartmentalization, or coarse-grained molecular dynamics in rule-based languages, to be explored in future work.

1.1 Related work

In [14] Gruenert et al. introduce an approach to simulating combinatorially complex chemical systems by combining a coarse-grained, particle-based spatial simulation with a rule-based language for triggering association and dissociation events between particles. Unlike in our approach, the simulation not only handles steric constraints, but tracks and updates the positions of individual particles according to a combination of force field and diffusion terms that simulate the interactions between particles and their Brownian motion, respectively.

Another particle-based simulator for biological systems is BioShape [4]. BioShape is a scale-independent simulation environment based on a process algebra
for systems biology, the Shape Calculus \[1\], in which every process is assigned a position, velocity and mass. Processes move according to physical laws and can collide with and bind to other processes depending on their communication on certain channels.

A different approach to describing spatial phenomena using a processes algebra is Cardelli and Gardner’s 3\(\pi\) \[5\]. 3\(\pi\) is a geometric process algebra that combines the interaction primitives of the \(\pi\)-calculus with geometric transformations. Processes in 3\(\pi\) are localized and move explicitly through frame shift operations. 3\(\pi\) is a powerful process algebra but does not offer any direct support for biological modeling: geometric constraints between processes or laws of motion have to be implemented explicitly. There is also no notion of time-flow, although one could easily be adapted, for example from the stochastic \(\pi\)-calculus.

The rest of this paper is structured as follows: in section 2 we define geometrically enhanced Kappa and its stochastic semantics and outline a basic simulation algorithm. In section 3 we introduce the notion of locality, give examples of local and non-local Kappa models, show that locality is undecidable in general, and finally show how to reduce local models to geometry-less models. Section 4 contains some concluding remarks and outlines future work.

## 2 Geometrically enhanced Kappa

We will now introduce our extension to Kappa as a simple categorical language. The definitions given below are, for the most part, direct extensions of those given in \[17\] and \[10\]. The reader that is somewhat familiar with the Kappa language will recognize many of the concepts introduced here, such as contact graphs, site graphs and embeddings, as variations of those used in classic, geometry-less Kappa.
2.1 Notation

Given a family of sets \( \{A_i, i \in I\} \), we write \( \bigcup_{i \in I} A_i \) for their disjoint union, that is, \( \bigcup_{i \in I} \{x, i : x \in A_i\} \). We write \( A - B \) for the set difference of two sets \( A \) and \( B \) or the relative complement of \( B \) in \( A \), \( \mathcal{P}(A) \) for the power set of \( A \), that is, the set of subsets of \( A \), and \( A_* \) for the disjoint union \( A + \{\star\} \) that adds a "stub" \( \star \) to \( A \).

If \( f : A \to B \) is a map from \( A \) to \( B \), we write \( f(A) \subseteq B \) for its image (or range). If \( f \) is partial, we write \( \text{dom}(f) \subseteq A \) for its domain of definition. If \( A' \subseteq A \) is a subset of the domain of \( f \), we write \( f|_{A'} \) for the restriction of \( f \) to \( A' \), that is, \( f|_{A'} : A' \to B, x \mapsto f(x) \). We write \( 1_A \) for the identity map from \( A \) to itself.

If \( R \subseteq A \times B \) is a binary relation between two sets \( A \) and \( B \), we sometimes treat it as a partial map \( R : A \to \mathcal{P}(B) \) and write \( \text{dom}(R) \subseteq A \) for its domain of definition, that is, \( \text{dom}(R) = \{a \in A : R(a) \neq \emptyset\} \), and \( R(A) \) for its image. Similarly, we often treat partial maps \( f : A \to \mathcal{P}(B) \) and \( g : A \to B \) as relations \( f \subseteq A \times B \) and functional relations \( g \subseteq A \times B \), respectively.

**Definition 2.1** A partial pairing on a set \( A \) is an irreflexive, symmetric and one-to-one binary relation on \( A \), that is, every element of \( A \) is in relation with at most one other element in \( A \) and no element in \( A \) is in relation with itself.

2.2 Contact graphs and site graphs

**Definition 2.2** A contact graph is a tuple \( C = (\mathcal{A}, \mathcal{S}, \mathcal{I}, \Sigma, \Lambda, I, P, S, \Omega) \), where

- \( \mathcal{A}, \mathcal{S} \) and \( \mathcal{I} \) are finite sets of agent types, site names and internal state names, respectively,
- \( \Sigma : \mathcal{A} \to \mathcal{P} (\mathcal{S}) \) assigns sets of site names to agent types,
- \( \Lambda \subseteq (\Sigma_*)^2 \setminus \{\{\star, \star\}\} \) is a symmetric binary relation that represents possible links between sites,
- \( I : \Sigma \to \mathcal{P} (\mathcal{I}) \) assigns sets of internal state names to sites,
- \( P : \mathcal{A} \to \mathcal{P} (\mathbb{R}) \) assigns finite sets of radii to agent types,
- \( S : \mathcal{S} \to \mathcal{P} (\mathbb{R}^3) \) assigns finite sets of positions to sites, and
- \( \Omega : (\Lambda \cap \Sigma^2) \to \mathcal{P} (\text{SO}(3)) \) assigns finite sets of orientations to links \((A, a, B, b)\), such that \( O \in \Omega(A, a, B, b) \Leftrightarrow O^{-1} \in \Omega(B, b, A, a) \).

We refer to \( \Sigma(A) \) as the signature of an agent type \( A \in \mathcal{A} \), and to \( \Sigma \) and \( \Lambda \) as the collective signature and link structure of the contact graph \( C \), respectively. We often treat the collective signature \( \Sigma \) and the link structure \( \Lambda \) of a contact graph as sets of pairs and quadruples, respectively (that is, as the graphs of the corresponding relations) as in the definitions of \( \Lambda, I, S \) and \( \Omega \) above. We call \( P, S \) and \( \Omega \) the agent, site and link geometry, respectively, of \( C \) and the combined map \((P, S, \Omega)\) the geometry of \( C \).

**Definition 2.3** A site graph is a tuple \( s = (V, C, \Gamma, \lambda, \iota, \rho, s, \omega) \), where

- \( V \) is a finite set of agents,
Figure 2. Geometry of a site graph. The circles represent agents $u \in V$ and $v \in V$, the gray arrows represent their frame of reference (or orientation). The sites $a \in \sigma(u)$ and $b \in \sigma(v)$ (and their positions $s(u,a), s(v,b)$) are represented by colored arrows. The presence of a link $(u,a,v,b)$ forces the absolute position of the sites $(u,a)$ and $(v,b)$ to coincide.

- $C$ is a contact graph,
- $\Gamma : V \to A$ is map assigning types to agents,
- $\sigma = \Sigma \circ \Gamma$ assigns sets of site names to agents,
- $\lambda \subseteq (\sigma_*)^2 - \{(\star, \star)\}$ is a partial pairing that represents links between sites,
- $\iota : \sigma \to I$ is a partial map that assigns internal states names to sites,
- $\rho : V \to \mathbb{R}$ is a partial map that assigns radii to agents,
- $s : \sigma \to \mathbb{R}^3$ is a partial map that assigns positions to sites,
- $\omega : (\lambda \cap \sigma^2) \to SO(3)$ is a partial map that assigns orientations $\omega(u,a,v,b)$ to links $(u,a,v,b)$, such that
  $$(u,a,v,b) \in \text{dom}(\omega) \Rightarrow \omega(v,b,u,a) = \omega(u,a,v,b)^{-1}.$$ We call $\Gamma$ the contact map of $s$. A site graph $s$ with contact map $\Gamma$ is said to be well-defined or well-typed if

(i) $\Gamma$ preserves links: $(u,a) \lambda (v,b) \Rightarrow (\Gamma(u),a) \Lambda (\Gamma(v),b),$

(ii) $\Gamma$ preserves stubs: $(u,a) \lambda \star \Rightarrow (\Gamma(u),a) \Lambda \star,$

(iii) $\Gamma$ preserves internal states: $(u,a) \in \text{dom}(\iota) \Rightarrow \iota(u,a) \in I(\Gamma(u),a),$

(iv) $\Gamma$ preserves agent geometry: $\rho \subseteq P \circ \Gamma,$

(v) $\Gamma$ preserves site geometry: $(u,a) \in \text{dom}(s) \Rightarrow s(u,a) \in S(\Gamma(u),a),$ and

(vi) $\Gamma$ preserves link geometry:

$$(u,a,v,b) \in \text{dom}(\omega) \Rightarrow \omega(u,a,v,b) \in \Omega(\Gamma(u),a,\Gamma(v),b).$$

We will only consider well-defined site graphs hereafter.

We refer to $\sigma$ and $\lambda$ as the collective interface and link structure of $s$, respectively, and we call $\sigma(u)$ the interface of an agent $u$. For an agent $u$ and a site $a \in \sigma(u)$, we say $(u,a)$ is bound if $\exists (v,b) : (u,a) \lambda (v,b)$. Conversely, we say that $(u,a)$ is free or has a stub if $(u,a) \lambda \star$. We call $\rho$, $s$ and $\omega$ the agent, site and link geometry, respectively, of $s$ and the combined map $(\rho,s,\omega)$ the geometry of $s$. The position $s(u,a)$ of a site $(u,a)$ is specified relative to the position and orientation of the agent $u$. If the orientation $\omega(u,a,v,b)$ of a link $(u,a,v,b) \in \lambda$ is defined, the sites $(u,a)$
and \((v, b)\) are considered aligned, and \(\omega(u, a, v, b)\) corresponds to the orientation of \(v\) relative to \(u\). Figure 2 illustrates the various components of the geometry of a site graph.

Two agents \(u\) and \(v\) of a site graph \(s\) are said to be connected if there is a path \((u, a)\lambda(w_1, c_1), (w_1, c'_1)\lambda(w_2, c_2), \ldots, (w_N, c'_N)\lambda(v, b)\) in \(s\) connecting \(u\) and \(v\). A site graph \(s\) is connected if every pair of agents \((u, v)\in V^2\) in \(s\) is connected.

**Definition 2.4** A site graph \((V, C, \Gamma, \lambda, t, \rho, s, \omega)\) is said to be complete if its collective interface, link structure, state map and geometry are all maximal with respect to its contact graph, that is,

- \((\Gamma(u), a)\in\text{dom}(\Lambda)\Rightarrow(u, a)\in\text{dom}(\lambda)\),
- \((\Gamma(u), a)\in\text{dom}(I)\Rightarrow(u, a)\in\text{dom}(I)\),
- \(\text{dom}(\rho) = \text{dom}(P\circ\Gamma)\),
- \((\Gamma(u), a)\in\text{dom}(S)\Rightarrow(u, a)\in\text{dom}(S)\),
- \((\Gamma(u), a, \Gamma(v), b)\in\text{dom}(\Omega)\Rightarrow(u, a, v, b)\in\text{dom}(\omega)\).

A site graph that is not complete is said to be partial.

Note that the link state \(\lambda(u, a)\) or internal state \(t(u, a)\) of a site \((u, a)\) may be left undefined in a complete site graph as long as the set of possible link states \(\Lambda(\Gamma(u), a)\) or internal states \(I(\Gamma(u), a)\) of the corresponding site \((\Gamma(u), a)\) in the contact graph are empty. Similarly, a complete geometrically enhanced site graph need not have all its geometry defined as long as the corresponding piece of geometry is undefined in the contact graph. This allows us to treat ordinary, geometry-less site graphs such as those defined in [8] as just a special case of geometrically enhanced site graphs. It also allows us to model semi-rigid structures, that is, structures where only certain substructures are subject to rigid geometric constraints.

Let us now have a look at some examples of contact and site graphs. Our first example introduces a set of simple, small graphs that we will build upon throughout this section to illustrate the various concepts form the theory of geometrically enhanced site graphs.

**Example 2.5** Consider the following definitions of a contact graph \(C\) and two site graphs \(t\) and \(s\):

\[
C = \begin{pmatrix}
A = \{A\}, & S = \{l, r\}, & I = \emptyset, & \Sigma = \{A\mapsto l, A\mapsto r\}, \\
\Lambda = \{(A, l, A, r), (A, r, A, l)\} \cup \Sigma \times \{\star\} \cup \{\star\} \times \Sigma, & I = \emptyset, \\
P = \{A\mapsto 1\}, & S = \{(A, l)\mapsto (-2, 0, 0), (A, r)\mapsto (2, 0, 0)\}, \\
\Omega = \{\{(A, l, A, r)\mapsto \{1, R_z(\frac{\pi}{2})\}\}, (A, r, A, l)\mapsto \{1, R_z(-\frac{\pi}{2})\}\}. 
\end{pmatrix}
\]

\[
t = \begin{pmatrix}
V_t = \{u\}, & C, & \Gamma_t = \{u\mapsto A\}, & \sigma_t = \Sigma \circ \Gamma_t, \\
\lambda_t = \sigma_t \times \{\star\} \cup \{\star\} \times \sigma_t, & \iota_t = \emptyset, & \rho_t = \{u\mapsto 1\}, \\
\sigma_t = \{(u, l)\mapsto (-2, 0, 0), (u, r)\mapsto (2, 0, 0)\}, & \omega_t = \emptyset, 
\end{pmatrix}
\]

\[
s = \begin{pmatrix}
V_s = \{u, v\}, & C, & \Gamma_s = \{u\mapsto A, v\mapsto A\}, & \sigma_s = \Sigma \circ \Gamma_s, \\
\lambda_s = \{(u, l, \star), (v, r, \star), (\star, u, l), (\star, v, r)\}, & \iota_s = \emptyset, & \rho_s = \{u\mapsto 1, v\mapsto 1\}, \\
\sigma_s = \{(u, l)\mapsto (-2, 0, 0), (v, r)\mapsto (2, 0, 0)\}, & \omega_s = \emptyset, 
\end{pmatrix}
\]
Adding the corresponding geometries as follows: would be an obvious choice. We extend this contact graph by the example 2.6 inner rings of the proteasome. A possible site graph representation of this structure could be linked to form a chain of length two. Note that the link state of sites is not complete, while represents a single, unconnected agent, while and are examples of a linear extension and a turn, respectively. is complete, while and are not connected, while is not. is a single, unconnected agent, while and are examples of a linear extension and a turn, respectively. , and are all connected, while is not. is complete, while and are not.

Now, let us have a look at a somewhat more involved system, namely the two inner rings of the proteasome. A possible site graph representation of this structure is shown in figure 3b.

**Example 2.6** In a model of the assembly of the two inner heptameric rings formed by the subunits of the proteasome, a geometry-less contact graph like the one shown in figure 3c would be an obvious choice. We extend this contact graph by adding the corresponding geometries as follows:

\[
C = \begin{pmatrix}
A = \{\beta\}, \ S = \{l_t, r_t, l_b, r_b\}, \ I = \emptyset, \ S = \{\beta \mapsto l_t, \beta \mapsto r_t, \beta \mapsto l_b, \beta \mapsto r_b\}, \\
\Lambda = \{\beta, l_t, \beta, r_t, (\beta, r_b, \beta, l_t), (\beta, l_b, \beta, l_b), (\beta, r_b, \beta, r_b)\} \cup \Sigma \times \{*\} \cup \{*\} \times \Sigma, \\
I = \emptyset, \ P = \{\beta \mapsto 1\}, \\
S = \{(\beta, l_t) \mapsto (-\cos \frac{\pi}{4}, \sin \frac{\pi}{4}, 0), \ (\beta, r_t) \mapsto (\cos \frac{\pi}{4}, \sin \frac{\pi}{4}, 0), \\
(\beta, l_b) \mapsto (-\frac{1}{2}, \frac{1}{2} \tan \frac{\pi}{4}, -1), \ (\beta, r_b) \mapsto (\frac{1}{2}, \frac{1}{2} \tan \frac{\pi}{4}, -1)\}, \\
\Omega = \{(\beta, l_t, \beta, r_t) \mapsto R_z (\frac{2\pi}{3}), \ (\beta, r_t, \beta, l_t) \mapsto R_z (\frac{2\pi}{3}), \\
(\beta, l_b, \beta, l_b) \mapsto R_z (-\frac{\pi}{3}) R_y (\pi), \ (\beta, r_b, \beta, r_b) \mapsto R_z (\frac{2\pi}{3}) R_y (\pi)\}
\end{pmatrix}
\]

The contact graph allows us to build site graphs like the one shown in figure 3b. Indeed, the site graph of figure 3b is the maximum sound site graph that has C as its contact graph. We will define what it means for a site graph to be sound in section 2.6.
2.3 Embeddings

**Definition 2.7** An **embedding** from a site graph \( x \) to another site graph \( y \) is a map \( f : V_x \to V_y \) such that

(i) \( f \) is injective.

(ii) \( f \) preserves types: \( \Gamma_x = \Gamma_y \circ f \),

(iii) \( f \) preserves sites: \( \sigma_x \subseteq \sigma_y \circ f \),

(iv) \( f \) preserves links: \( (u, a) \lambda_x (v, b) \Rightarrow (f(u), a) \lambda_y (f(v), b) \),

(v) \( f \) preserves stubs: \( (u, a) \lambda_x \star \Rightarrow (f(u), a) \lambda_y \star \),

(vi) \( f \) preserves states: \( (u, a) \in \text{dom}(\iota_x) \Rightarrow \iota_x(u, a) = \iota_y(f(u), a) \),

(vii) \( f \) preserves agent geometry: \( \rho_x \subseteq \rho_y \circ f \),

(viii) \( f \) preserves site geometry: \( (u, a) \in \text{dom}(s_x) \Rightarrow s_x(u, a) = s_y(f(u), a) \),

(ix) \( f \) preserves link geometry:

\[
(u, a, v, b) \in \text{dom}(\omega_x) \Rightarrow \omega_x(u, a, v, b) = \omega_y(f(u), a, f(v), b).
\]

We write \([x; y]\) for the set of embeddings from \( x \) to \( y \). When \( f \in [x; y] \), we write \( x \underrightarrow{f} y \) and say that \( y \) embeds \( x \) or that \( x \) is embedded in \( y \) via \( f \). We write \( f(x) \) for \( y \), the target site graph of \( f \in [x; y] \), not to be confused with the image of \( f \), \( f(V_x) \subseteq V_y \).

Together, site graphs and their embeddings form the category **GS** of geometrically enhanced site graphs, with site graphs as objects and embeddings as arrows. In the rest of this document, we will use the terms object and site graph interchangeably, as well as the terms embedding, morphism and arrow.

The notion of an embeddings gives us a straight-forward way to define **subgraphs** in the category **GS**:

**Definition 2.8** A site graph \( t \) is a **site subgraph** of a site graph \( s \), denoted \( t \subseteq s \), if \( V_t \subseteq V_s \) and \( t \underrightarrow{\text{V}} s \). A site subgraph is said to be **minimal** for some property \( P \) if, for every other site subgraph \( t' \subseteq s \) for which \( P \) holds, \( t' \subseteq t \) implies \( t' = t \), and **maximal** for \( P \) if, \( t \subseteq t' \) implies \( t = t' \). A site subgraph \( s[W] \subseteq s \) is said to be **induced** if it is maximal given its agent set \( V_s[W] = W \). A site graph \( t \subseteq s \) is a **connected component** of \( s \) if it is a maximal connected site subgraph of \( s \).

At this point it seems appropriate to note that, although many graph-theoretical concepts, such as subgraphs, connected components, paths, etc. extend in an intuitive manner to site graphs, the category of site graphs also has a few somewhat surprising properties due to the restriction of one link per site and the injectivity of morphisms. One such property is **rigidity**:

**Lemma 2.9 (rigidity)** Let \( x \) and \( y \) be site graphs and suppose \( x \) is connected, then any non-empty partial injection \( f \) from \( V_x \) to \( V_y \) extends to at most one embedding in \([x; y]\).
This is a known result from the theory of geometry-less site graphs, which extends unchanged to geometrically enhanced site graphs. We refer the interested reader to [17,10] for the proof. One consequence of lemma 2.9, is that there is a polynomial-time algorithm for computing the number of embeddings $|s;x|$ of a connected site graphs $s$ in a site graph $x$, that is, to count the number of site subgraphs in $x$ that are isomorphic to $s$. This is at the heart of the efficient simulation algorithm for Kappa models presented in [8].

We denote by $\text{GS}_C$ the subcategory of site graphs with contact graph $C$. Most of the definitions in the rest of our presentation of geometrically enhanced Kappa will be formulated with respect to a given $\text{GS}_C$ rather than $\text{GS}$. This restriction reflects the intuition that, in the context of some biochemical process, the active sites of a given type of elementary molecule (represented by the interface $\Sigma(A)$ of some agent type $A \in \mathcal{A}_C$), its set of potential configurations ($I(A)$, $P(A)$, $S(A)$), as well as its potential interactions with other types of molecules ($\Lambda(A)$, $\Omega(A)$) are all fixed, irrespective of where and how the particular instances of $A$ are embedded in some molecular complex.

2.4 Actions and rules

In order to define rewriting rules on site graphs, we need to be able to relate the site graphs before and after the application of a given rule. One option would be to extend one of the classic semantic approaches from graph rewriting, such as double or single pushout rewriting, which have been defined for geometry-less Kappa [6,10]. Instead, we will adopt the more syntactic concept of an action script as a sequence of atomic rewrite instructions, which has been used more extensively in the Kappa literature [8,17].

**Definition 2.10** An atomic action $\alpha$ on a site graph $s$ in $\text{GS}_C$ is one of the following:

- a subgraph addition $+t$ of a complete site graph $t$ in $\text{GS}_C$,
- a subgraph deletion $-t$ of a complete site graph $t$ in $\text{GS}_C$,
- a link addition $+(u,a,v,b,Q)$ with orientation $Q \in \Omega(u,a,v,b)$,
- a link deletion $-(u,a,v,b)$,
- a state change $\iota(u,a) \leftarrow p$ to $p \in I(u,a)$,
- a change of radius $\rho(u) \leftarrow r$ to $r \in P(u)$,
- a change of site position $s(u,a) \leftarrow p$ to $p \in S(u,a)$, or
- a change of link orientation $\omega(u,a,v,b) \leftarrow O$ to $O \in \Omega(u,a,v,b)$,

with $u \in V_s$, $v \in V_s$, $a \in \sigma_s(u)$ and $b \in \sigma_s(v)$.

An action $\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_N)$ is a sequence of atomic actions $\alpha_i$ on $(\alpha_{i-1} \cdot \alpha_{i-2} \cdot \cdots \cdot \alpha_1 \cdot s)$.

Link additions and deletions are symmetric. We denote by $\alpha \cdot s$ the result of an atomic action $\alpha$ on $s$ in $\text{GS}_C$. An action $\alpha$ on $s$ is well-defined if its result $\alpha \cdot s$ is
a well-defined site graph. The parameter $Q$ of a well-defined link addition may be $Q = \star$ iff $\Omega(u, a, v, b) = \emptyset$, in which case $\omega_{\alpha \cdot s}(u, a, v, b)$ remains undefined. Link deletion are only well-defined for complete links. We only consider well-defined actions hereafter. A formal definition of the result of an action can be found in appendix A.1.

Given an embedding $f \in [s; x]$ between two site graphs $s$ and $x$ in $\text{GS}_C$, we define the transport $f(\alpha)$ of an atomic action $\alpha$ along $f$ as the sequence of atomic actions $f(\alpha_i)$, such that every $f(\alpha_i)$ performs the analogue of the action $\alpha_i$ on the corresponding site subgraph of the image of $s$ in $x$ (see appendix A.2 for a formal definition).

Figure 4 illustrates the various objects, arrows and actions defined on $s$, $\alpha$, $x$ and $f$.

Using the above definitions, we can now define the rules of our site graph rewriting system.

**Example 2.11** Let $t$, $s$, $x$ and $y$ be as defined in example 2.5, then we can define two actions $\alpha_1$ and $\alpha_2$ as follows:

$$\alpha_1 = +(u, l, v, r, 1), \quad \alpha_2 = \left(+(u, l, v, r, R_2 \left(-\frac{\pi}{2}\right)), +t\right).$$

Then $\alpha_1$ is an atomic action defined on $s$ that adds a straight link to $s$, that is, $\alpha_1 \cdot s = x$. $\alpha_2$ is a non-atomic action defined on $s$ that first adds a link (a $90^\circ$ turn) to $s$ and then adds the subgraph $t$, that is, $\alpha_1 \cdot s = +t \cdot y$.

**Definition 2.12** A rule is a triple $r = (s, \alpha, k)$, where $s$ is a site graph, $\alpha$ is an action defined on $s$ and $k \in \mathbb{R}_0^+$ is a rate.

**Example 2.13** Let $t$ and $s$ be defined as in example 2.5, and $\alpha_1$, $\alpha_2$ as defined in example 2.11, then we can define two rules $r_1$ and $r_2$ as follows:

$$r_1 = (s, (\alpha_1, +t), k_1), \quad r_2 = (s, \alpha_2, k_2).$$
Finally, we are ready to define our notion of a geometrically enhanced Kappa model.

**Definition 2.14** A Kappa model is a pair $K = (x_0, R)$ with $x_0$ a complete site graph in $\text{GS}_{C_K}$ called the initial state of $K$, and $R = \{r_1, r_2, \ldots, r_N\}$ a finite set of rules $r_i = (s_i, \alpha_i, k_i)$ called the rule set of $K$, with left-hand sides $s_i$ in $\text{GS}_{C_K}$.

For simplicity, we will write $\text{GS}_K$ instead of $\text{GS}_{C_K}$ hereafter.

**Example 2.15** Let $C$, $t$, $s$, $r_1$ and $r_2$ be defined as in example 2.13. Then we can define the Kappa model $K = (x_0, R)$ with initial state $x_0 = +t \cdot t$ and rule set $R = \{r_1, r_2\}$.

### 2.5 Events and probabilities

Every Kappa model $K$ has an associated Weighted Labeled Transition System (WLTS) [13] on complete site graphs:

**Definition 2.16** Let $r = (s, \alpha, k)$ be a rule, $R$ be a set of rules and $x$ be a complete site graph. We define

- the set of events in $x$ associated with $r$ as $\mathcal{E}(x, r) = [s; x]$,
- the set of events in $x$ associated with $R$ as $\mathcal{E}_R(x) = \sum_{r \in R} \mathcal{E}(x, r)$,
- the activity of $r$ in $x$ as $a(x, r) = k \cdot [s; x]$, and
- the activity of $R$ in $x$ as $a_R(x) = \sum_{r \in R} a(x, r)$.

This defines a WLTS $\mathcal{W}_K = (\mathcal{K}_K, \mathcal{E}_R, w, \pi_0)$ for every Kappa model $K = (x_0, R)$, with

- **state space** $\mathcal{K}_K \subseteq \text{GS}_K$, the class of complete site graphs in $\text{GS}_K$,
- **labels** $\mathcal{E}_R = \sum_{x \in \mathcal{K}_K} \mathcal{E}_R(x)$, the set of events in $\mathcal{K}_K$ associated with $R$,
- **weight function** $w : \mathcal{E}_R \times \mathcal{K}_K \to \mathbb{R}_0^+$ defined as
  $$ w(x, e, x') = \begin{cases} 
  k & \text{if } e = ((s, \alpha, k), f) \in \mathcal{E}_R(x) \text{ and } x' = f(\alpha) \cdot x, \\
  0 & \text{otherwise},
  \end{cases} $$
- **initial probability distribution** $\pi_0 = 1_{\{x = x_0\}}$. 
A transitions \( x \xrightarrow{e} x' \), in \( W_K \) has probability
\[
P\{ x \xrightarrow{e} x' \} = \frac{w(x,e,x')}{a_R(x)}
\]
if \( a_R(x) > 0 \), and the subsequent time advance is an exponential random variable \( \delta t(x) \) such that
\[
P\{ \delta t(x) > t \} = e^{-a_R(x)t}.
\]
We write \( x_1 \xrightarrow{*} x_N \) if there is a sequence of transitions
\[
x_1 \xrightarrow{e_1} x_2 \xrightarrow{e_2} x_3 \xrightarrow{e_3} \cdots \xrightarrow{e_{N-1}} x_N
\]
in \( W_K \) with \( w(x_i,e_i,x_{i+1}) > 0 \), and denote by \( \text{reach}(K) \subseteq K \) the set of reachable states of \( K \): \( \text{reach}(K) = \{ x : x_0 \xrightarrow{*} x \} \).

### 2.6 Realizations and constraints

**Definition 2.17** A realization \( \pi : V \rightarrow \mathbb{R}^3 \times SO(3) \) of a geometrically enhanced site graph \( s \) is an embedding of \( s \) in three-dimensional euclidean space, that is, a map \( u \mapsto (r(u), R(u)) \) that assigns positions \( r(u) \) and orientations \( R(u) \) to agents \( u \in V \).

Given a realization \( \pi = (r, R) \) of a site graph \( s \), we call \( r \) and \( R \) the position map and orientation map of \( s \), respectively. We can now define geometric soundness constraints for site graphs and Kappa models:

**Definition 2.18** A site graph \( s \) with associated geometry \((\rho, s, \omega)\) is said to be sound if and only if there is a realization \( \pi = (r, R) \) of \( s \), such that

- for every link \((u,a,v,b) \in \lambda \) in the link structure of \( s \), the geometry of \((u,a),(v,b)\) and \((u,a,v,b)\) is consistent, that is,
  \[
  \forall (u,a,v,b) \in (\text{dom}(\omega) \cap \text{dom}(s)^2) : \quad \begin{align*}
  R(v) & = \omega(u,a,v,b)R(u) \\
  r(v) + R(v)s(v,b) & = r(u) + R(u)s(u,a)
  \end{align*}
  \]
  and

- there are no collisions in \( s \), that is,
  \[
  \forall (u,v) \in \text{dom}(\rho)^2 : u \neq v \Rightarrow \|r(u) - r(v)\| \geq \rho(u) + \rho(v).
  \]

We denote by \( \mathbb{G} \) the set of site graphs that are geometrically sound and write \( s \in \mathbb{G} \) if the site graph \( s \) is sound. We say that a Kappa model \( K = (x_0,R) \) is geometrically sound, if its initial state \( x_0 \) is geometrically sound.

We say that a pair of agents \((u,v)\) in a site graph \( s \) are geometrically connected if \( u \) and \( v \) are connected by a path composed of links \((w,a,w',b)\) with completely specified geometries: \((w,a,w',b) \in (\text{dom}(\omega) \cap \text{dom}(s)^2)\). We say \( s \) is geometrically connected if every pair \((u,v) \in V^2 \) in \( s \) is geometrically connected. Let \( s \) be a site graph with geometrically connected components \( \{c_1, c_2, \ldots, c_N\} \), then we have
\[
s \in \mathbb{G} \iff \forall i \leq N : c_i \in \mathbb{G},
\]
because any agents that are not geometrically connected may be located arbitrarily far apart without violating any of the soundness constraints. It is easy to see that the system of equations in definition 2.18 becomes over-constrained if one fixes the position \( r(u_i) \) and orientation \( R(u_i) \) of an arbitrary agent \( u_i \) in every geometrically connected component \( c_i \) of \( s \). This suggests a simple \( O(n^2) \) algorithm (with \( n = |\lambda_i| \)) for computing the soundness \( s \in \mathcal{G} \) of a site graph \( s \): for every \( c_i \subseteq s \\
(i) set \( r(u_i) = 0 \) and \( R(u_i) = 1 \) for an arbitrary agent \( u_i \in V_i \), \n(ii) solve and check the soundness equations for the rest of \((r, R)|_{V_i}\) by traversing \( c_i \) starting from \( u_i \) (in \( O(|\lambda_i|) \) time), \n(iii) check for potential collisions in \( c_i \) (in \( O(|V_i|^2) \) time).

See algorithm B.1 in appendix B for details.

**Definition 2.19** Let \( \mathcal{W}_K = (\mathcal{K}_K, \mathcal{E}_R, w, \pi_0) \) be the WLTS associated with the geometrically sound Kappa model \( K = (x_0, R) \), then the geometrically sound transition system \( \mathcal{W}_K \) is defined as the WLTS \( \mathcal{W}_K = (\mathcal{K}_K, \mathcal{E}_R, w', \pi_0) \) with weight function \( w' : \mathcal{E}_R \times \mathcal{K}_K \rightarrow \mathbb{R}_0^+ \) defined as

\[
  w(x, e, x') = \begin{cases} 
    k & \text{if } e = ((s, \alpha, k), f) \in \mathcal{E}_R(x) \text{ and } x' = f(\alpha) \cdot x \text{ and } x' \in \mathcal{G}, \\
    0 & \text{otherwise.}
  \end{cases}
\]

In other words, \( \mathcal{W}_K \) is the WLTS with transitions \( x \xrightarrow{(r,f)} f(\alpha_r) \cdot x \) restricted to those transitions in \( \mathcal{W}_K \) for which \( f(\alpha_r) \cdot x \) is sound.

Provided we start out from a geometrically sound initial state \( x_0 \), a geometrically sound transition system \( \mathcal{W}_K \) will only evolve along trajectories whose states \( x_0 \rightarrow^* x \) are also geometrically sound.

Using algorithm B.1 we can modify the well known simulation algorithm for Kappa systems [8] to efficiently simulate the \( \mathcal{W}_K \) of any geometrically sound model \( K \) by generating a null-event (that is, time advance without a transition) whenever algorithm B.1 returns \( f(\alpha_r) \cdot x \notin \mathcal{G} \) for some event \((r, f)\) in state \( x \).

### 3 Locality

In this section we will introduce the notion of *locality* of a Kappa model. The core idea behind locality is the following: a Kappa model \( K = (x_0, R) \) is said to be *local* if it is possible to decide whether or not any event \(((s, \alpha, k), f) \in \mathcal{E}_R(x)\) will produce a sound site graph \( f(\alpha) \cdot x \) by inspecting only a finite amount of context around the image of the right-hand side \( \alpha \cdot s \) in \( f(\alpha) \cdot x \). Note that the distinction of local vs. non-local models only makes sense for geometrically enhanced Kappa, as all geometry-less site graphs are geometrically sound by definition, and hence events in geometry-less Kappa models can never produce geometrically unsound graphs. Indeed, the existence of non-local Kappa models is one of the fundamental differences between geometrically enhanced and geometry-less Kappa.
In order to make the concept of locality more precise, we first need to define what exactly we mean by context. For this purpose, we will use the concept of a closed ball around a subset of agents in a site graph.

**Definition 3.1** Given a pair of connected agents \((u, v) \in V^2\) of a site graph \(s\), the distance \(d(u, v)\) between \(u\) and \(v\) is defined as the length, that is, the minimum number of links, of the shortest path between \(u\) and \(v\).

**Definition 3.2** The closed ball \(B_\delta(u, x)\) of radius \(\delta\) around an agent \(u \in V_x\) of a site graph \(x\) is defined as the set of agents \(B_\delta(u, x) = \{v \in V_x : d(u, v) \leq \delta\}\) that lie within distance \(\delta\) of \(u\) in \(x\).

The closed ball \(B_\delta(f)\) of radius \(\delta\) around an embedding \(f \in [s; x]\) is the induced site subgraph

\[
B_\delta(f) = x \left[ \bigcup_{u \in V_s} B_\delta(f(u), x) \right].
\]

**Definition 3.3** A Kappa model \(K = (x_0, R)\) with initial state \(x_0\) and rule set \(R\) is said to be \(\delta\)-local for a finite radius \(\delta \in \mathbb{N}\), if for every state \(x \in \text{reach}(K)\) reachable in \(K\), rule \((s, \alpha, k) \in R\) and embedding \(f \in [s; x]\), the following holds

\[
B_\delta(\alpha \cdot f) \in \mathbb{G} \Rightarrow f(\alpha) \cdot x \in \mathbb{G}.
\]

\(K\) is called *local* if there exists a finite radius \(\delta \in \mathbb{N}\) such that \(K\) is \(\delta\)-local.

We denote by \(\mathbb{L}_\delta\) the set of \(\delta\)-local Kappa models and by \(\mathbb{L} = \bigcup_i \mathbb{L}_i\) the set of local Kappa models.

**Example 3.4** To see that there are indeed non-local Kappa models \(K \notin \mathbb{L}\), consider the Kappa model \(K = (x_0, R)\), as defined in example 2.15:

After \(n\) events, a state \(x_n \in \text{reach}(K)\) of this model consists of two or more connected components that are either free agents \(t\) of type \(A\), open chains \(c_m\), \(m \leq n\) of \(m + 1\) agents \(w_i\) of type \(A\) with adjacent agents \(w_i\) and \(w_{i+1}\) linked through \(m\) links \((w_i, l, w_{i+1}, r)\), or loops, that is chains that have been closed. A given state \(x_n\) can contain many loops, but at most two free agents \(t\), or a single free agent \(t\) and a single open chain \(c_m\). Most events will connect the single free
agent $t$ to the existing chain $c_m$ producing a new chain $c_{m+1}$. Events involving rule $r_1$ create a straight extension at one of the ends of the existing chain $c_m$, while rule $r_2$ creates a left or right turn at the $l$-end or $r$-end, respectively, of $c_m$. Both $r_1$ and $r_2$ also create a new free agent $t$ for the further extension of $c_{m+1}$. Occasionally, the model will produce chains $c_{2N+2M-1}$, such as shown in figure 6b, that can be closed into a rectangular loop $r_{N,M}$ of dimensions $N \times M$ by the event $(r_2, f = \{u \mapsto w_{2N+2M-1}, v \mapsto w_0\})$.

Given any $\delta \in \mathbb{N}$, let $x_{2\delta+1} = +t \cdot c_{2\delta+1}$ be a site graph where $c_{2\delta+1}$ is a chain as shown in figure 6a with $N = \delta$ and $M = 1$, and let $f_{2\delta+1} = \{u \mapsto w_{2\delta+1}, v \mapsto w_0\}$ with $w_{2\delta+1}$ and $w_0$ the agents at the $l$-end and $r$-end of $c_{2\delta+1}$, respectively. Clearly, $x_{2\delta+1} \in \text{reach}(K)$ and $f_{2\delta+1} \in \mathcal{E}(x_{2\delta+1}, r_2)$. Assume we apply $f_{2\delta+1}( \alpha_2 )$ to $x_{2\delta+1}$, that is, we attempt to close the chain $c_{2\delta+1}$. Then the resulting site graph $f_{2\delta+1}( \alpha_2 ) \cdot x_{2\delta+1}$ must be unsound, as the chain $c_{2\delta+1}$ has too few turns to be closed (only chains with three or more turns, such as the chain in figure 6b, can be closed, ). However, the closed ball $B_\delta(\alpha_2 \cdot f_{2\delta+1})$ is sound, as it is not closed (the agent $w_\delta$ is missing from $B_\delta(\alpha_2 \cdot f_{2\delta+1})$). We conclude that

$$\forall \delta \in \mathbb{N} : B_\delta(\alpha_2 \cdot f_{2\delta+1}) \in \mathbb{G} \land f(\alpha_2) \cdot x_{2\delta+1} \notin \mathbb{G},$$

and therefore $K \notin \mathbb{L}$.

### 3.1 Sufficient conditions for locality

The following are sufficient conditions for the locality of a model $K$:

(i) A model $K$ is local if its contact graph $C_K$ is acyclic. As we have seen in section 2.6, a site graph $x$ is sound if all its connected components $c \subseteq x$ are sound. If $C_K$ is acyclic, then no connected component $c$ can have more agents than there are agent types in $C_K$, that is, $|V_c| \leq |A_K|$. Hence no rule of $(s, \alpha, k) \in R_K$ can affect any agent at a distance greater than $\delta = |A_K|$ around its left-hand side $s$, and it follows that $K$ is $|A_K|$-local.

(ii) $K$ is local if

$$\exists \delta \in \mathbb{N} \forall x \in \mathcal{K}_K \forall (s, \alpha, k) \in R \forall f \in \{s; x\} : B_\delta(\alpha \cdot f) \in \mathbb{G} \Rightarrow f(\alpha) \cdot x \in \mathbb{G}.$$  

Since $\text{reach}(K) \subseteq \mathcal{K}_K$, the above statement is simply a stronger version of locality. We say that $K$ is local with respect to its contact graph if it fulfills the above condition. An example of a model that is local with respect to its contact graph is a modified version $K'$ of the model $K$ from example 3.4, with the rule $r_1$ and the straight extension in $\Omega_K$ removed. The possible connected components of any complete site graphs $x \in \mathcal{K}_{K'}$ are free agents $t$, chains $c_1$, $c_2$, $c_3$ with one to three turns, respectively, and rectangles $r_{1,1}$. Hence $K' \in \mathbb{L}_4$.

An example of a model that is not local with respect to its contact graph but still in $\mathbb{L}$ would be another modified version $K''$ of the model from example 3.4, where we replace the initial state $x_0$ by $x_1 = +t \cdot c_1$, with $c_1$ a chain of two agents
connected by a straight link and \( t \) a single free agent, and we refine the left-hand side of the rules \( r_1 \) and \( r_2 \), such that the new left-hand sides \( s_1 \) and \( s_2 \) match exactly the two agents at the free end of a chain that are connected by a turn and a straight extension, respectively. The possible connected components of any state \( x \in \text{reach}(K'') \) are free agents, chains \( c_1 \) to \( c_7 \), with straight extensions and turns interleaved, and rectangles \( r_{2,2} \). Thus \( K'' \in \mathbb{L}_8 \). However, the class \( \mathcal{K}_{K''} = \mathcal{K}_K \) is identical to that of the model \( K \) from example 3.4, and hence \( K'' \) cannot be local with respect to its contact graph.

3.2 Undecidability of locality

**Proposition 3.5 (undecidability)** The set of local Kappa models \( \mathbb{L} \) is undecidable.

We will not give a full proof of proposition 3.5 here, but rather a proof sketch.

**Proof sketch** The proof is by many-one reduction from the halting problem. We will assume that we can fix a suitable Gödel numbering for Kappa models such that they can be manipulated by Turing machines.\(^5\) Conversely, when given a Turing machine \( M \), it is easy to construct a geometry-less Kappa model \( (x_0(M,n), R(M)) \) that simulates \( M \) for input \( n \), with the rule set \( R(M) \) implementing the state transition function of \( M \), and \( x_0(M,n) \) representing the initial state of \( M \) for input \( n \), that is, a site graph representing \( M \) with the contents of its tape set to \( n \) and the head of \( M \) pointing to the first cell of its tape. Figure 7 illustrates this idea.

Without loss of generality, assume that every Turing machine \( M \) has exactly one dedicated final state \( q_\downarrow \). The rule set \( R(M) \) of any Kappa model \( K_M = (x_0(M,n), R(M)) \) simulating a Turing machine \( M \), can easily be extended with the rule set \( R^* \) of some canonical non-local model \( K^* = (x_0^*, R^*) \), such as the one given in example 3.4, and one additional rule \( r_\downarrow = (s_\downarrow, +x_0^*, k_\downarrow) \) with \( k_\downarrow > 0 \)

\(^5\) When we want to reason about the decidability of sets of Kappa models, we are forced to limit ourselves to the class of Kappa models that can be encoded as natural numbers or finite bit strings, that is, models with rate constants and geometries consisting of computable numbers.
some non-zero rate constant, \(x_0^*\) the initial state of \(K^*\), and \(s\) a site graph that embeds into exactly those states of \(K_M\) that represent \(M\) being in its final state \(q_\downarrow\). Since \(K_M\) is geometry-less and completely deterministic, the extended model \(K'_M = (x_0(M, n), R(M) + R^* + \{r\})\) can reach a “non-local state” if and only if there is a state \(x \in \text{reach}(K'_M)\), in which \(r\) becomes active, that is, \(E(x, r) \neq \emptyset\). But this implies that \(x\) represents \(M\) being in a final state, and therefore \(K'_M\) is a non-local model if and only if \(M\) halts. We conclude that the halting problem many-one-reduces to the decidability of \(L\). 

3.3 Reducing local models to geometry-less models

The definition of locality tells us that, for any rule \((s, \alpha, k) \in R\) of a \(\delta\)-local model \(K = (x_0, R)\), inspecting a finite amount of context (represented by the closed ball of radius \(\delta\) around the image of the right-hand side \(\alpha \cdot s\) of the rule in the result \(f(\alpha) \cdot x\) of an event \((s, \alpha, k), f\)) is sufficient to determine whether \(f(\alpha) \cdot x\) is sound or not. It therefore seems natural to think that, by extending the left-hand side \(s\) of the rule with a corresponding (finite) amount of additional context, one might be able to decide whether the result \(f'(\alpha) \cdot x\) of the extended event \((s', \alpha, k), f'\), \(f' \in [s'; x]\) is sound, simply by checking whether the right-hand side \(\alpha \cdot s'\) of the extended rule itself is sound. If so, one could statically compute the “soundness” of all such extended rules \((s', \alpha, k) \in R'\) and then flatten the model \(K'\) to obtain a geometry-less counterpart \(\overline{K}\) that “behaves” the same as \(K\). This intuition turns out to be correct.

**Proposition 3.6 (reducibility)** For every local Kappa model \(K \in L\), there is a geometry-less Kappa model \(\overline{K}\) that is strongly bisimilar to \(K\).

For lack of space, we will not give a formal proof of proposition 3.6 here, but instead sketch the reduction from a \(\delta\)-local model \(K\) to its geometry-less counterpart \(\overline{K}\). The reduction consists of three steps:

(i) **refine** the rule set \(R\) of \(K\) up to radius \(\delta\) to obtain \(K' = (x_0, R')\);

(ii) **prune** the resulting rule set \(R'\), by removing unsound rules, that is rules \((s, \alpha, k)\) for which \(\alpha \cdot s \notin G\), to obtain \(K'' = (x_0, R'')\);

(iii) **flatten** the resulting model \(K''\), that is, replace geometry with internal state in all site graphs and actions of \(K'\), to obtain \(\overline{K} = (\overline{x_0}, \overline{R})\).

Each of the Kappa models \(K', K''\) and \(\overline{K}\) resulting, respectively, from one of the above three steps can be shown to be bisimilar to its predecessor. The first step is based on the theory of rule refinement [17]. Every rule \(r \in R\) is replaced by a finite set \(B_{\delta, K}(r)\) of mutually exclusive, refined rules. Given a site graph \(s\) in \(GS_K\), one may build a set \(B_{\delta, K}(s)\) inductively, starting from the set \(B_{0, K}(s) = \{s\}\), and extending sites \((u, a) \in \sigma_i\) of site graphs \(t \in B_{i, K}(s)\) that are neither bound nor stubs according to \(\Lambda(\Gamma(u), a)\) in order to obtain \(B_{i+1, K}(s)\). This process is guaranteed to terminate for all \(i\) since \(\Lambda(A, a)\) is finite for any site \((A, a) \in \Sigma\). The theory of refinement guarantees, that every event in \((r, f) \in E_R(x)\) maps to exactly one event \((r', f') \in E_{R'}(x)\) of the refined rule set \(R' = \sum_{r \in R} B_{\delta, K}(r)\), and thereby ensures the
correctness of this step. However, for large and highly connected contact graphs $C_K$, the set $B_{\delta,K}(r)$ may suffer a combinatorial explosion in the number of rules and the size of the corresponding site graphs even for moderate values of $\delta$, making the computation of $B_{\delta,K}(r)$ intractable in practice.

The correctness of the second step follows from the definition of a geometrically sound WLTS: no transition in $\hat{W}_K'$ results from an unsound rule.

The last step can be achieved by first adding a set of dedicated sites $(A,s_{\rho})$, $(A,(s_{s},a))$ and $(A,(s_{\omega},a,B,b))$, for every agent type $A$, site $(A,a)$ and link $(A,a,B,b)$ of the contact graph $C'_K$ of $K'$, and then replacing every site graph $x$ and atomic action $\alpha$ in $K'$ by a geometry-less counterpart $\overline{x}$ and $\overline{\alpha}$, respectively, in which the geometries of every agent $u$, site $(u,a)$ and link $(u,a,v,b)$ have been replaced by appropriate internal states of $(u,s_{\rho})$, $(u,(s_{s},a))$ and $(u,(s_{\omega},a,\Gamma(v),b))$, respectively. The correctness of this step is by construction of the flattening function $\overline{\cdot}$ and locality itself: the transports of any action in $K''$ can no longer result in unsound site graphs (all left-hand sides contain the necessary context and have passed pruning), and hence their geometry has become redundant.

Figure 8 illustrates the reduction of a concrete site graph $x$ in a 1-local model $K$ to its geometry-less counterpart $\overline{x}$ in the geometry-less model $\overline{K}$.

4 Conclusion and future work

We presented geometrically enhanced Kappa, a framework for geometrically enhanced site graph rewriting and sketched an algorithm for tractable simulation.
We also showed that the geometric constraints introduced by the framework allows for the definition of non-local models and that all local models can be reduced to geometry-less Kappa models without changing the dynamics of the associated transition system, if one is willing to accept a combinatorial increase in the number of rules of the model. Finally, we showed that locality is undecidable in general.

A principal limitation of our framework is that it is restricted to rigid geometric constraints, while the chemical species of some systems, such as polymer chains, enjoy a certain amount of flexibility in their structure. A possible solution to this limitation would be to allow geometric annotations to occupy continuous intervals in the corresponding vector spaces, rather than a finite set of fixed points. A further generalization would be to allow distributions over such intervals. However, this might increase the complexity of checking the soundness of the resulting site graphs, possibly making it intractable.

Another current limitation is that, while the possible realizations of site graphs may be computed in order to check their soundness, they remain inaccessible to the modeler. Indeed, they are not even kept track of in the current framework. Extending the dynamics of the system to take the realization of its state into account might provide a way of modeling processes such as diffusion and compartmentalization, which play a key role in certain biochemical processes. Combinations of the above are of course possible, and in general there is a spectrum of possible spatial extensions to Kappa, with geometry-less Kappa at one end, and coarse-grained molecular dynamics approaches, such as [14], at the other.

Another direction for future work is to explore approaches to answer the question of whether a given Kappa model is local or not approximately. Techniques such as [9] for computing overapproximations of the set \( \text{reach}(K) \) of reachable states of a model \( K \) based on abstract interpretation could help to find subsets of \( \mathbb{L} \) if it turns out that locality is decidable with respect to the corresponding supersets of \( \text{reach}(K) \).

References

A Additional definitions

To simplify some of the definition below, we will first give a formal definition for two basic operations for composing or dividing site graphs, namely the disjoint union and the difference of two site graphs.

**Definition A.1** The **disjoint union** $x_1 + x_2$ of two site graphs $x_1$ and $x_2$ in $\mathbb{GS}_C$ is defined as the site graph in $\mathbb{GS}_C$ with agent set $V_{1+2} = V_1 + V_2$, and with $\Gamma_{1+2}$, $\lambda_{1+2}$, $\iota_{1+2}$, $\rho_{1+2}$, $s_{1+2}$ and $\omega_{1+2}$ the disjoint unions, with respect to $V_{1+2}$, of the corresponding components of $x_1$ and $x_2$.

Given two site graphs $x_1$ and $x = (x_1 + x_2)$, we call $x - x_1 = x_2$ the **relative complement** of $x_1$ in $x$ or simply the **difference** of $x$ and $x_1$. Given a site graph $x$ we say that $\{x_1, x_2, \ldots, x_N\}$ is a **partition** of $x$ if $x = \sum_{i=1}^{N} x_i$. 
A.1 Result of an action on a site graph

The result \( \alpha \cdot s \) of an atomic action \( \alpha \) on a site graph \( s \) in \( \text{GS}_C \) is defined as

- \( +t \cdot s = s + t \),
- \( -t \cdot s = s - t \),
- \( \lambda_+(u,a,v,b) \cdot s = \lambda_s - \lambda_-(u,a,v,b) \cup \lambda_+(u,a,v,b) \),
- \( \omega_+(u,a,v,b) \cdot s = \omega_s \cup \omega_+(u,a,v,b) \),
- \( \lambda_-(u,a,v,b) \cdot s = \lambda_s - \lambda_+(u,a,v,b) \cup \lambda_-(u,a,v,b) \),
- \( \omega_-(u,a,v,b) \cdot s = \omega_s - \omega_+(u,a,v,b) \),
- \( t_-(u,a) \cdot \omega_{(u,a,v,b) \to O} \cdot s = t_s[(u,a) \mapsto t_s(u,a)/(u,a) \mapsto p] \),
- \( \rho_-(u) \cdot \omega_{(u,a,v,b) \to O} \cdot s = \rho_s[u \mapsto \rho_s(u)/u \mapsto r] \),
- \( s_{(u,a) \to p} \cdot s = s_s[(u,a) \mapsto s_s(u,a)/(u,a) \mapsto p] \),
- \( \omega_{(u,a,v,b) \to O} \cdot s = \omega_s[(u,a,v,b) \mapsto \omega_s(u,a)/(u,a,v,b) \mapsto p] \),

with

\[
A[a/b] = A - \{a\} \cup \{b\},
\]

\[
\lambda_+(u,a,v,b) = \{(u,a,v,b)\} \cup \{(v,b,u,a)\},
\]

\[
\lambda_-(u,a,v,b) = \{(u,a),(v,b)\} \times \{\star\} \cup \{\star\} \times \{(u,a),(v,b)\},
\]

\[
\omega_+(u,a,v,b) = \begin{cases} 
(u,a,v,b) \mapsto Q & \text{if } \Omega(u,a,v,b) \neq \emptyset, \\
\emptyset & \text{otherwise},
\end{cases}
\]

\[
\omega_-(u,a,v,b) = \begin{cases} 
(u,a,v,b) \mapsto \omega_s(u,a,v,b) & \text{if } \Omega(u,a,v,b) \neq \emptyset, \\
\emptyset & \text{otherwise},
\end{cases}
\]

where we only mention the components of \( \alpha \cdot s \) that differ from those in \( s \). The result of a non-atomic action \( \alpha = (\alpha_1,\alpha_2,\ldots,\alpha_N) \) is defined as

\[
\alpha \cdot s = \alpha_N \cdot \alpha_{N-1} \cdots \alpha_1 \cdot s.
\]

A.2 Transport of an action along an embedding

**Definition A.2** Given an embedding \( f \in [s;x] \) between two site graphs \( s \) and \( x \) in \( \text{GS}_C \), the transport \( f(\alpha) \) of an atomic action \( \alpha \) along \( f \) is defined as

- \( f(+t) = +t \),
- \( f(-t) = -x[f\vert_V(V_t)] \),
- \( f(+u,a,v,b,Q) = +(f(u),a,f(v),b,Q) \),
- \( f(-u,a,v,b) = -(f(u),a,f(v),b) \),
- \( f(i(u,a) \leftarrow p) = (i(f(u),a) \leftarrow p) \),
\[ f(\rho(u) \leftarrow r) = (\rho(f(u)) \leftarrow r), \]
\[ f(s(u,a) \leftarrow p) = (s(f(u), a) \leftarrow p), \] and
\[ f(\omega(u,a,v,b) \leftarrow O) = (\omega(f(u), a, f(v), b) \leftarrow O). \]

The transport \( f(\alpha) \) of a non-atomic action \( \alpha = (\alpha_1, \alpha_2, \ldots, \alpha_N) \) along \( f \) is defined as \( f(\alpha) = (f(\alpha_1), f(\alpha_2), \ldots, f(\alpha_N)) \).

### A.3 Result of an action on an embedding

**Definition A.3** Given two site graphs \( s \) and \( x \) in \( \text{GS}_C \), a well-defined action \( \alpha \) on \( s \) and \( f \in [s;x] \), there is a unique map \((\alpha \cdot f) \in [\alpha \cdot s; f(\alpha) \cdot x]\) called the result of the action \( \alpha \) on \( f \) with

- \( +t \cdot f = f + 1_{V_t} \),
- \( -t \cdot f = f|_{V_s-V_t} \),
- \( \alpha \cdot f = f \) for any other atomic action \( \alpha \), and
- \( (\alpha_1, \alpha_2, \ldots, \alpha_N) \cdot f = \alpha_N \cdot \alpha_{N-1} \cdots \alpha_1 \cdot f \) for non-atomic actions.

### B Algorithms

#### B.1 Checking soundness of a site graph

**Input:** A site graph \( s = (V, C, \lambda, \iota, \rho, s, \omega, \Gamma) \).

**Output:** `true` if \( s \in \mathbb{G} \), `false` otherwise.

1: checked \( \leftarrow \emptyset \);
2: \textbf{while} \( V - \text{checked} \neq \emptyset \) \textbf{do} \hspace{1cm} \triangleright \text{Check consistency.}
3: \quad \text{Pick an arbitrary agent } i \in (V - \text{checked});
4: \quad \text{\quad} r(i) \leftarrow 0; \text{\quad} R(i) \leftarrow 1;
5: \quad \text{\quad} \text{next} \leftarrow \{i\}; \text{\quad} \text{checked' } \leftarrow \emptyset;
6: \quad \textbf{while} \text{\quad} \text{next} \neq \emptyset \textbf{ do}
7: \quad \quad \text{Pick an arbitrary agent } u \in \text{next};
8: \quad \quad \textbf{for all } (v,b,u,a) \in (\text{dom} (\omega) \cap \text{dom} (s)^2) \textbf{ do}
9: \quad \quad \quad \text{if } v \notin (\text{checked} \cup \text{checked'}) \text{ then}
10: \quad \quad \quad \quad \text{R'} \leftarrow \omega(v,b,u,a)R(u);
11: \quad \quad \quad \quad \text{r'} \leftarrow r(u) + R(u)s(u,a) - R's(v,b);
12: \quad \quad \quad \quad \text{if } v \in \text{next then}
13: \quad \quad \quad \quad \text{\quad if } r' \neq r(v) \vee R' \neq R(v) \text{ then}
14: \quad \quad \quad \quad \quad \text{\quad return false;}
15: \quad \quad \quad \quad \end{align*}
16: \quad \quad \quad \quad \text{\quad \quad end if}
17: \quad \quad \quad \quad \text{\quad else}
18: \quad \quad \quad \quad \text{\quad \quad} r(v) \leftarrow r'; \text{\quad} R(v) \leftarrow R';
19: \quad \quad \quad \quad \text{\quad\quad next } \leftarrow \text{\quad next } \cup \{v\};
20: \quad \quad \quad \quad \end{align*}
21: \quad \quad \text{\quad end if}
22: \quad \quad \text{\quad \quad end if}
23: \quad \quad \text{\quad end for}
24: \quad \quad \text{\quad end if}
25: \quad \end{align*}
26: \quad \text{\quad end for}
27: \quad \text{\quad end while}
28: \text{\quad end if}
29: \text{\quad checked' } \leftarrow \text{\quad \text{checked} \cup \text{\quad \text{checked'}};
30: \text{\quad next } \leftarrow \text{\quad next } \cup \{v\};
31: \text{\quad end while}
32: \text{\quad end if}
33: \text{end if}
20:    end if
21: end for
22: next ← next − {u};
23: checked′ ← checked′ ∪ {u};
24: end while
25: for all (u,v) ∈ checked′ × checked′ : v ≠ u do ▷ Check for collisions.
26:     if ∥r(u) − r(v)∥ ≤ ρ(u) + ρ(v) then
27:         return false; ▷ s ∉ G
28:     end if
29: end for
30: checked ← checked ∪ checked′;
31: end while
32: return true; ▷ s ∈ G