5th Workshop on Logic and Systems Biology

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

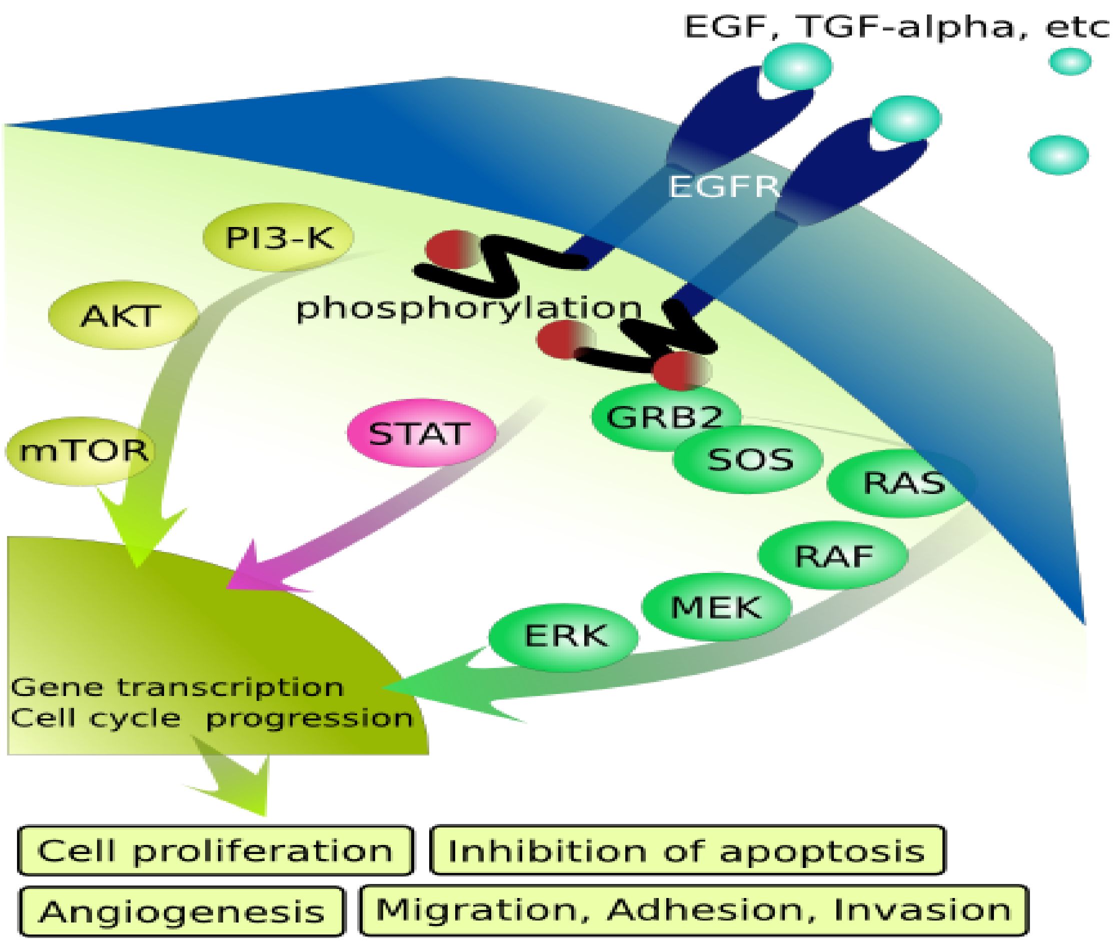
Jérôme Feret
DI - ÉNS

2014, July 13

Overview

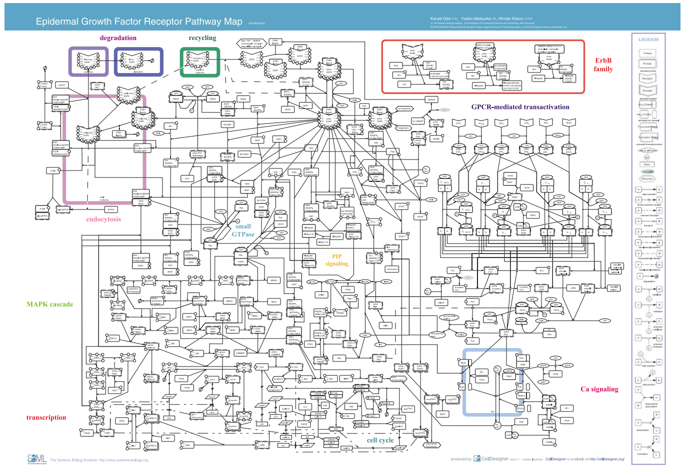
1. Context and motivations
2. Case study
3. Kappa semantics
4. Symmetries in site-graphs
5. Symmetric models
6. Conclusion

Signalling Pathways



Eikuch, 2007

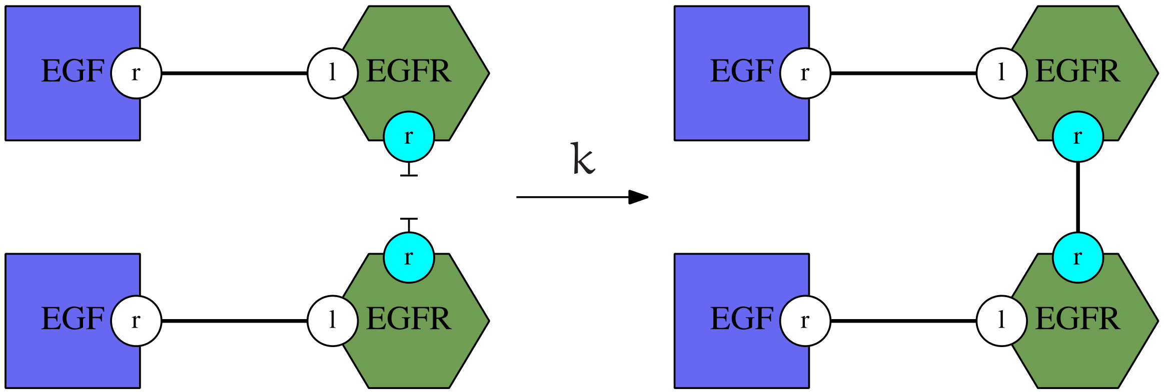
Bridging the gap between...



$$\begin{cases} \frac{dx_1}{dt} = -k_1 \cdot x_1 \cdot x_2 + k_{-1} \cdot x_3 \\ \frac{dx_2}{dt} = -k_1 \cdot x_1 \cdot x_2 + k_{-1} \cdot x_3 \\ \frac{dx_3}{dt} = k_1 \cdot x_1 \cdot x_2 - k_{-1} \cdot x_3 + 2 \cdot k_2 \cdot x_3 \cdot x_3 - k_{-2} \cdot x_4 \\ \frac{dx_4}{dt} = k_2 \cdot x_3^2 - k_2 \cdot x_4 + \frac{v_4 \cdot x_5}{p_4 + x_5} - k_3 \cdot x_4 - k_{-3} \cdot x_5 \\ \frac{dx_5}{dt} = \cdots \\ \vdots \\ \frac{dx_n}{dt} = -k_1 \cdot x_1 \cdot c_2 + k_{-1} \cdot x_3 \end{cases}$$

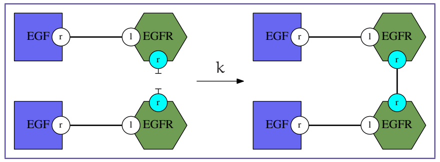
knowledge representation **and** **models of the behaviour of systems**

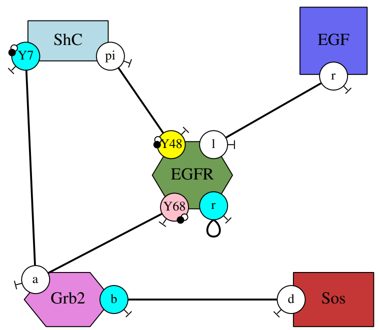
Site-graphs rewriting



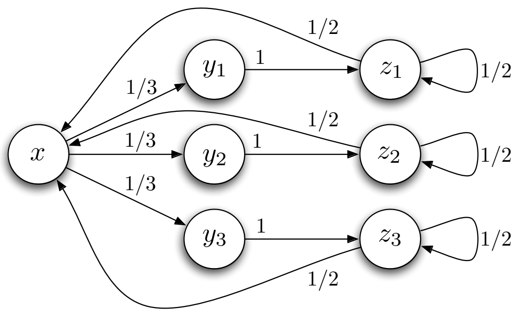
- a language close to knowledge representation;
- rules are easy to update;
- a compact description of models.

Choices of semantics





interaction map

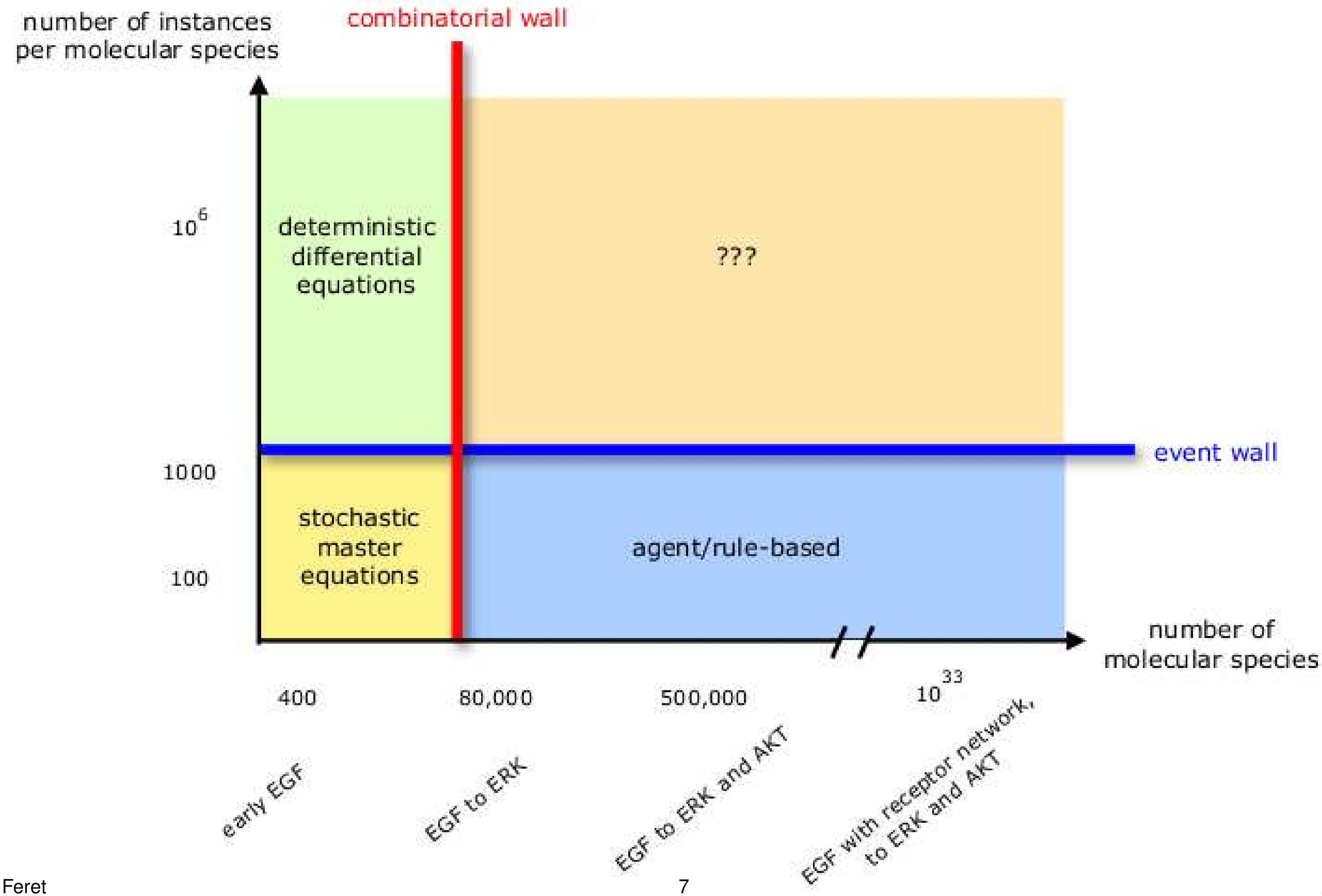


Markov chain

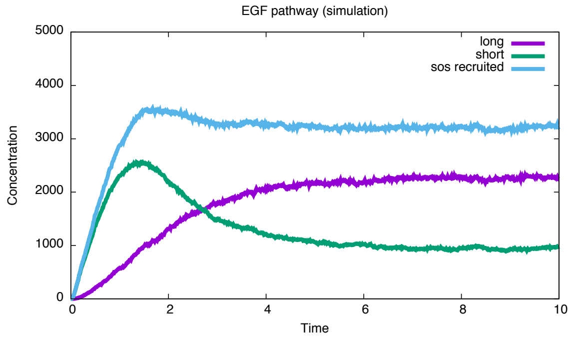
$$\begin{cases} \frac{dx_1}{dt} = -k_1 \cdot x_1 \cdot x_2 + k_{-1} \cdot x_3 \\ \frac{dx_2}{dt} = -k_1 \cdot x_1 \cdot x_2 + k_{-1} \cdot x_3 \\ \frac{dx_3}{dt} = k_1 \cdot x_1 \cdot x_2 - k_{-1} \cdot x_3 + 2 \cdot k_2 \cdot x_3 \cdot x_3 - k_{-2} \cdot x_4 \\ \frac{dx_4}{dt} = k_2 \cdot x_3^2 - k_2 \cdot x_4 + \frac{v_4 \cdot x_5}{p_4 + x_5} - k_3 \cdot x_4 - k_{-3} \cdot x_5 \\ \frac{dx_5}{dt} = \cdots \\ \vdots \\ \frac{dx_n}{dt} = -k_1 \cdot x_1 \cdot c_2 + k_{-1} \cdot x_3 \end{cases}$$

ordinary differential equations

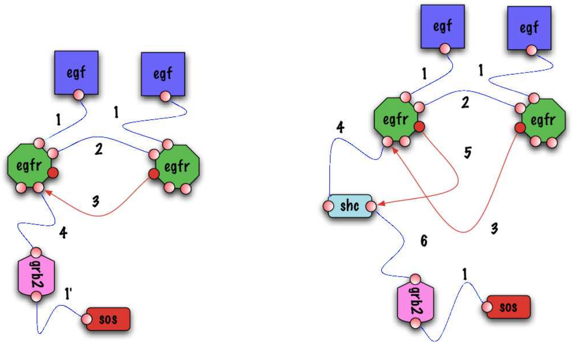
Complexity walls



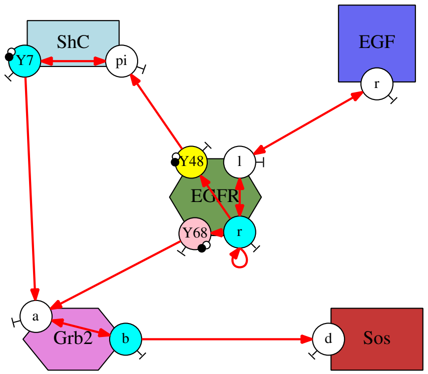
Abstractions offer different perspectives on models



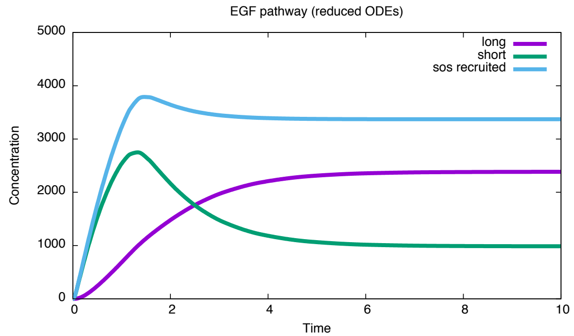
concrete semantics



causal traces



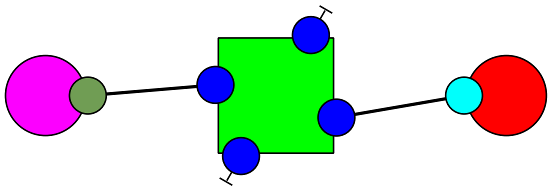
information flow



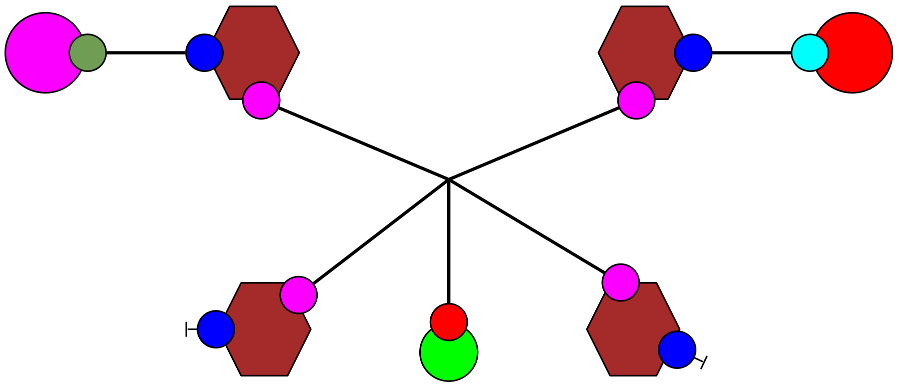
exact projection
of the ODE semantics

Symmetric sites

- in BNGL or MetaKappa (multiple-occurrences of sites):

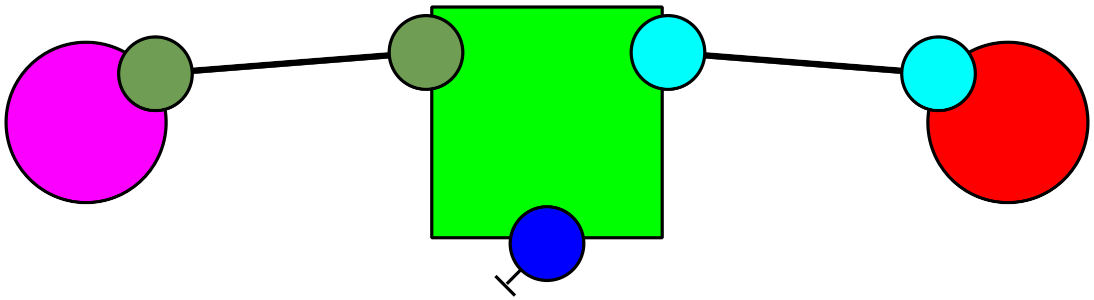


- in Formal Cellular Machinery or React(C) (hyper-edges):

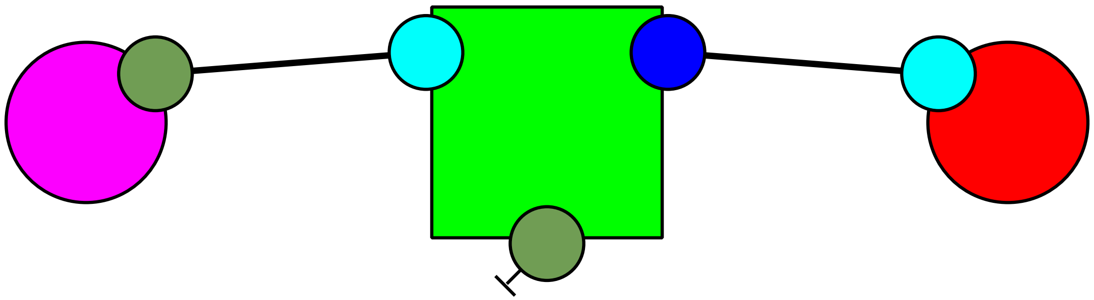


Blinov et al., BioNetGen: software for rule-based modeling of signal transduction based on the interactions of molecular domains, Bioinformatics 2004
Danos et al., Rule-Based Modelling and Model Perturbation, TCSB 2009
Damgaard et al., Formal cellular machinery, Damgaard et al., SASB 2011
John et al., Biochemical Reaction Rules with Constraints, ESOP 2011

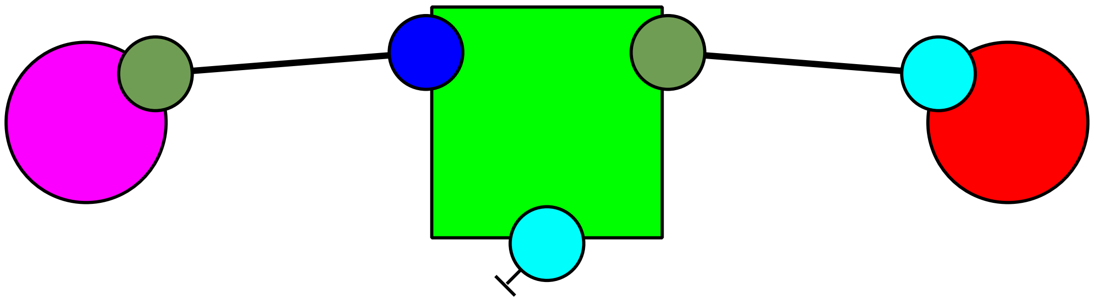
Other kinds of symmetries:
Circular permutations



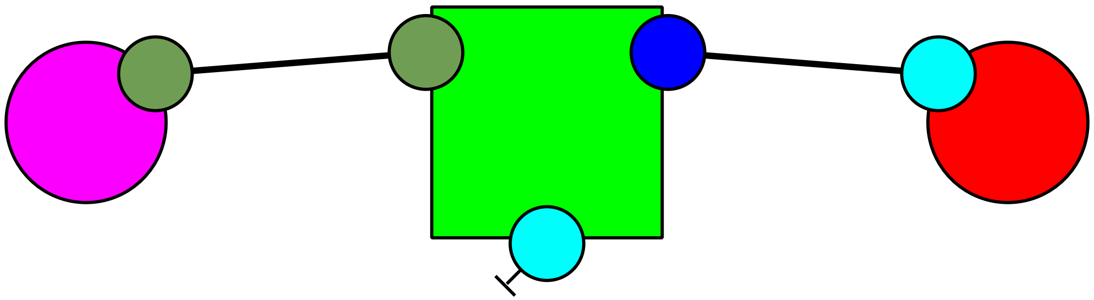
Other kinds of symmetries:
Circular permutations



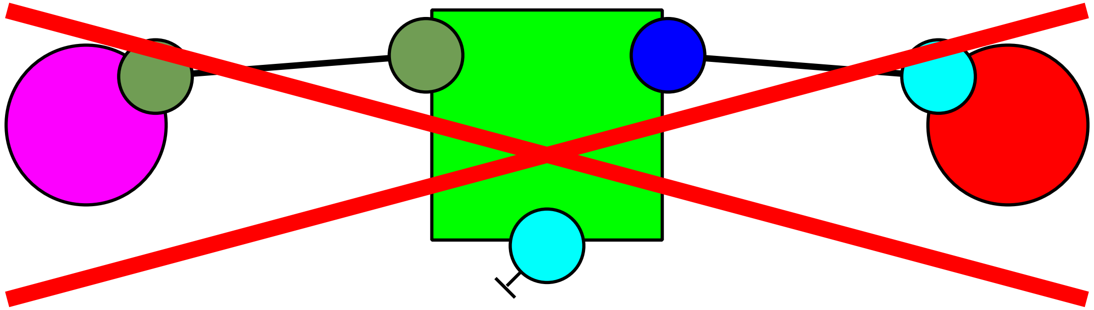
Other kinds of symmetries: Circular permutations



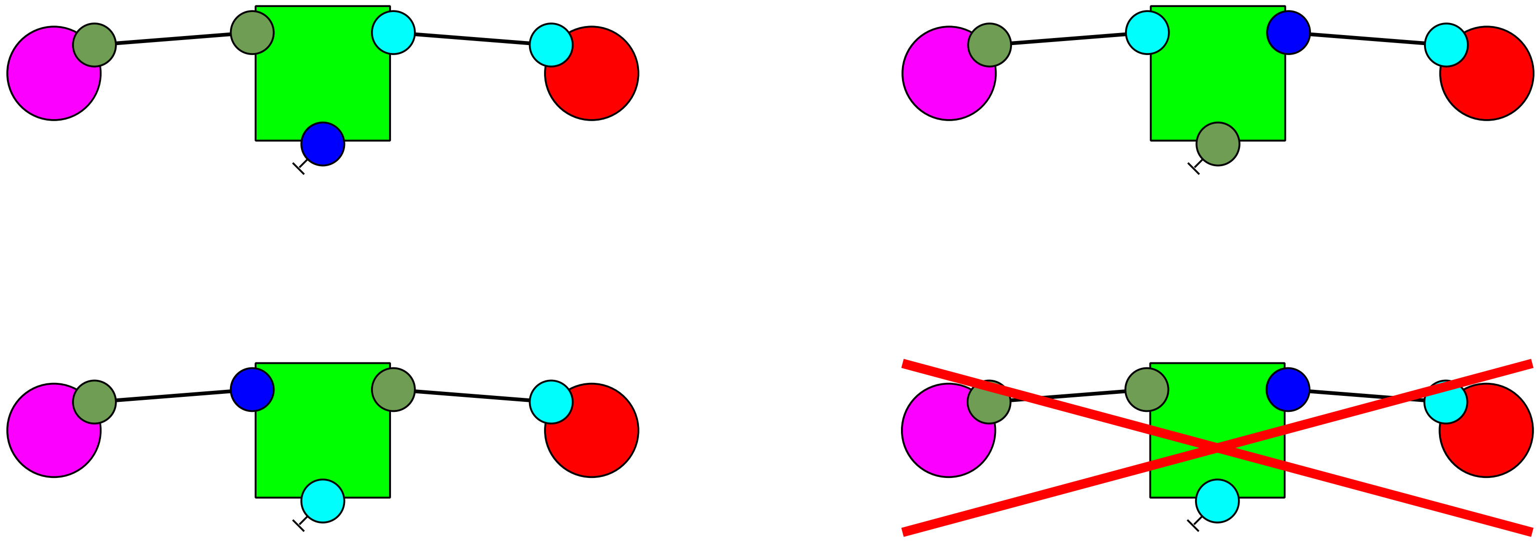
Other kinds of symmetries:
Circular permutations



Other kinds of symmetries:
Circular permutations

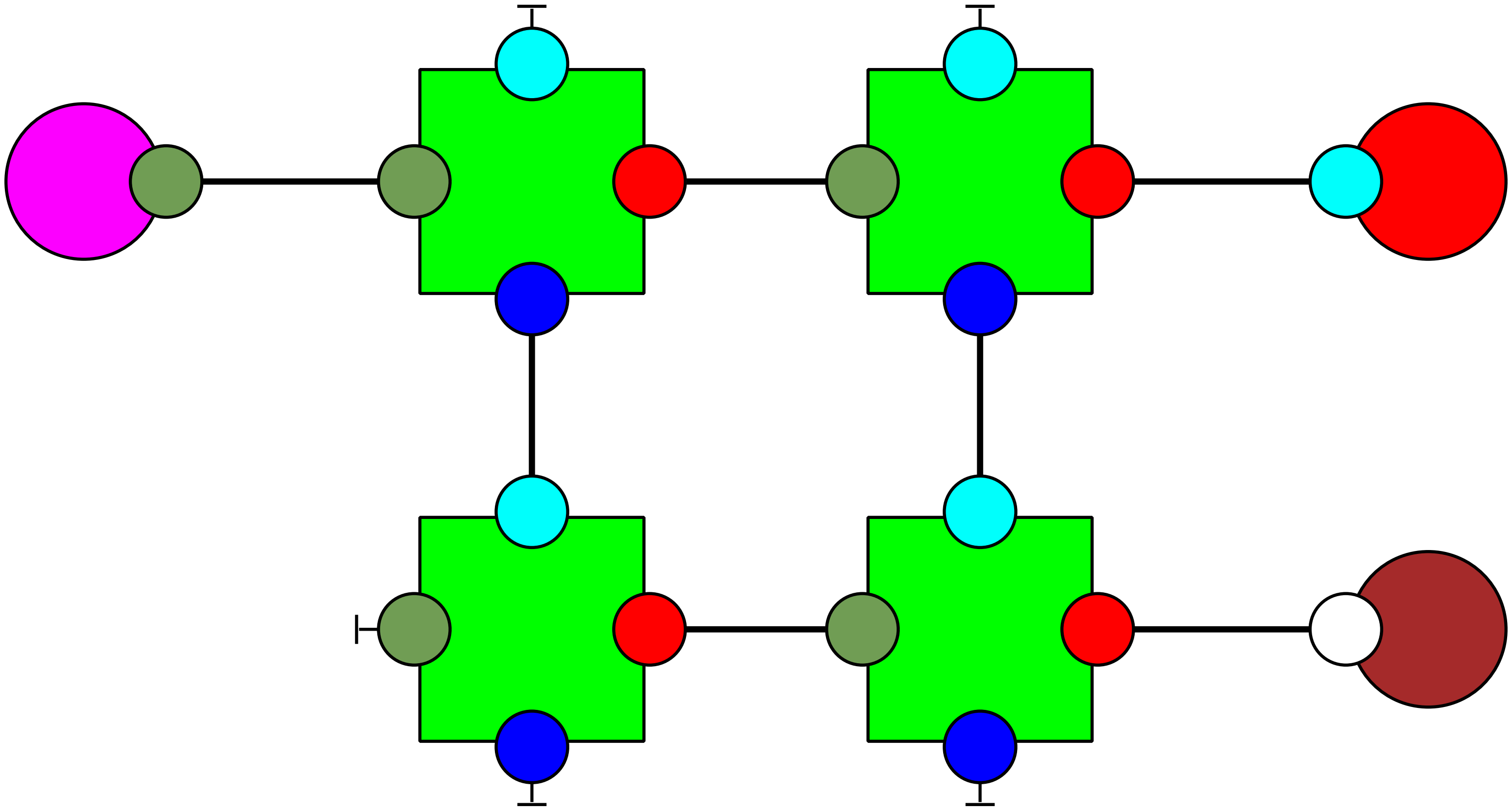


Other kinds of symmetries:
Circular permutations



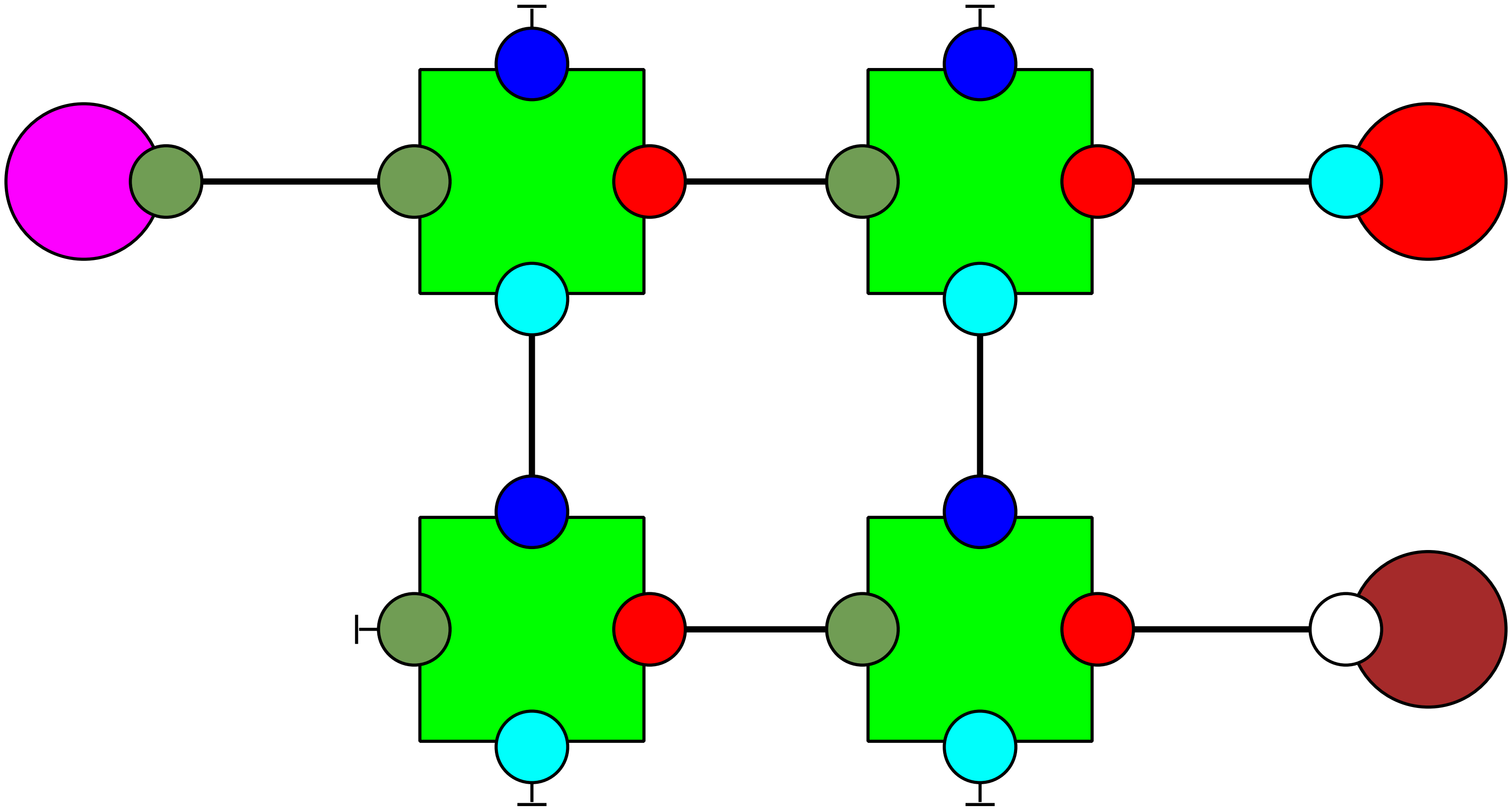
Other kinds of symmetries: Homogeneous symmetries

We can compute a horizontal reflection.



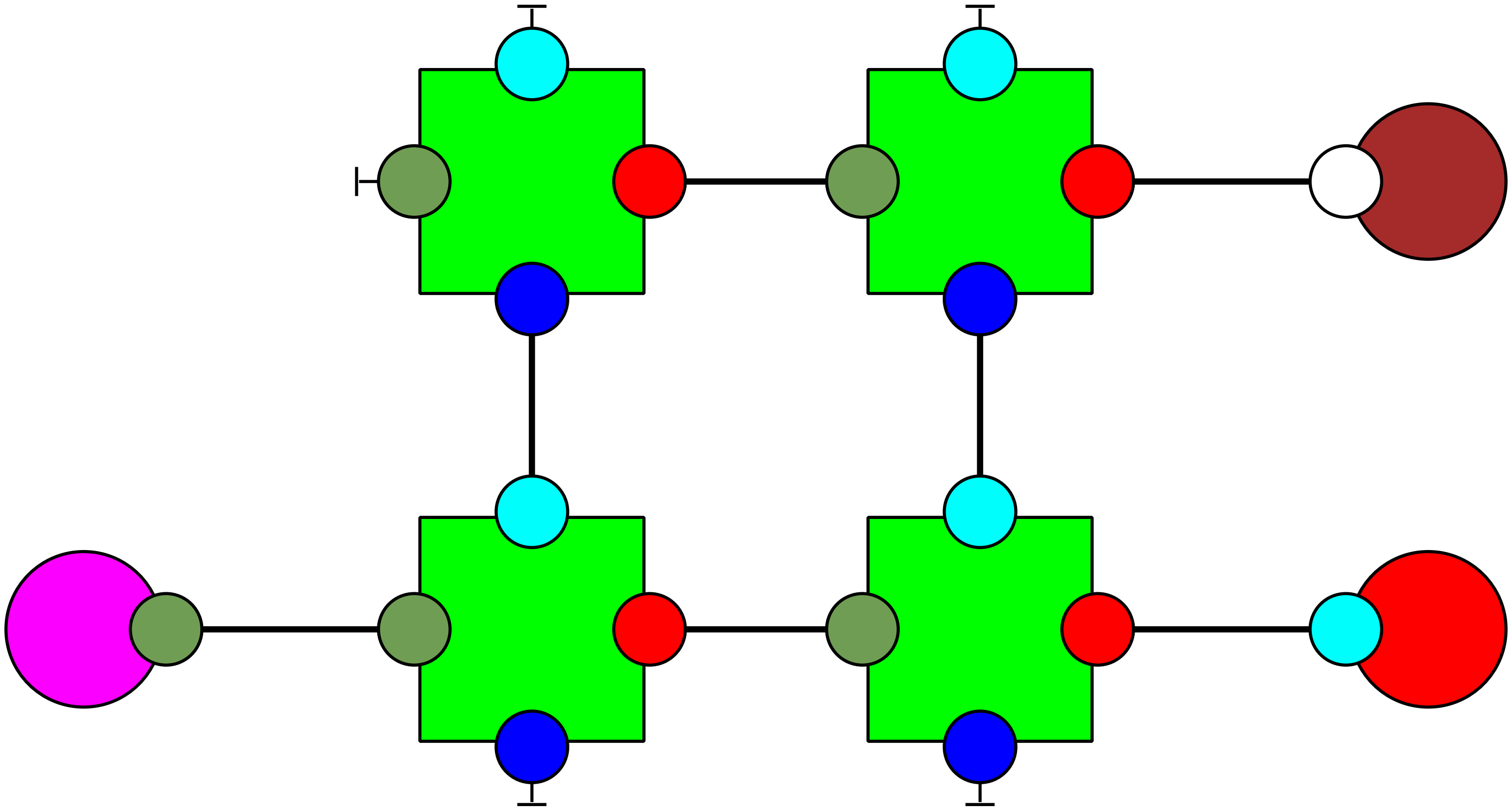
Other kinds of symmetries: Homogeneous symmetries

We can compute a horizontal reflection.



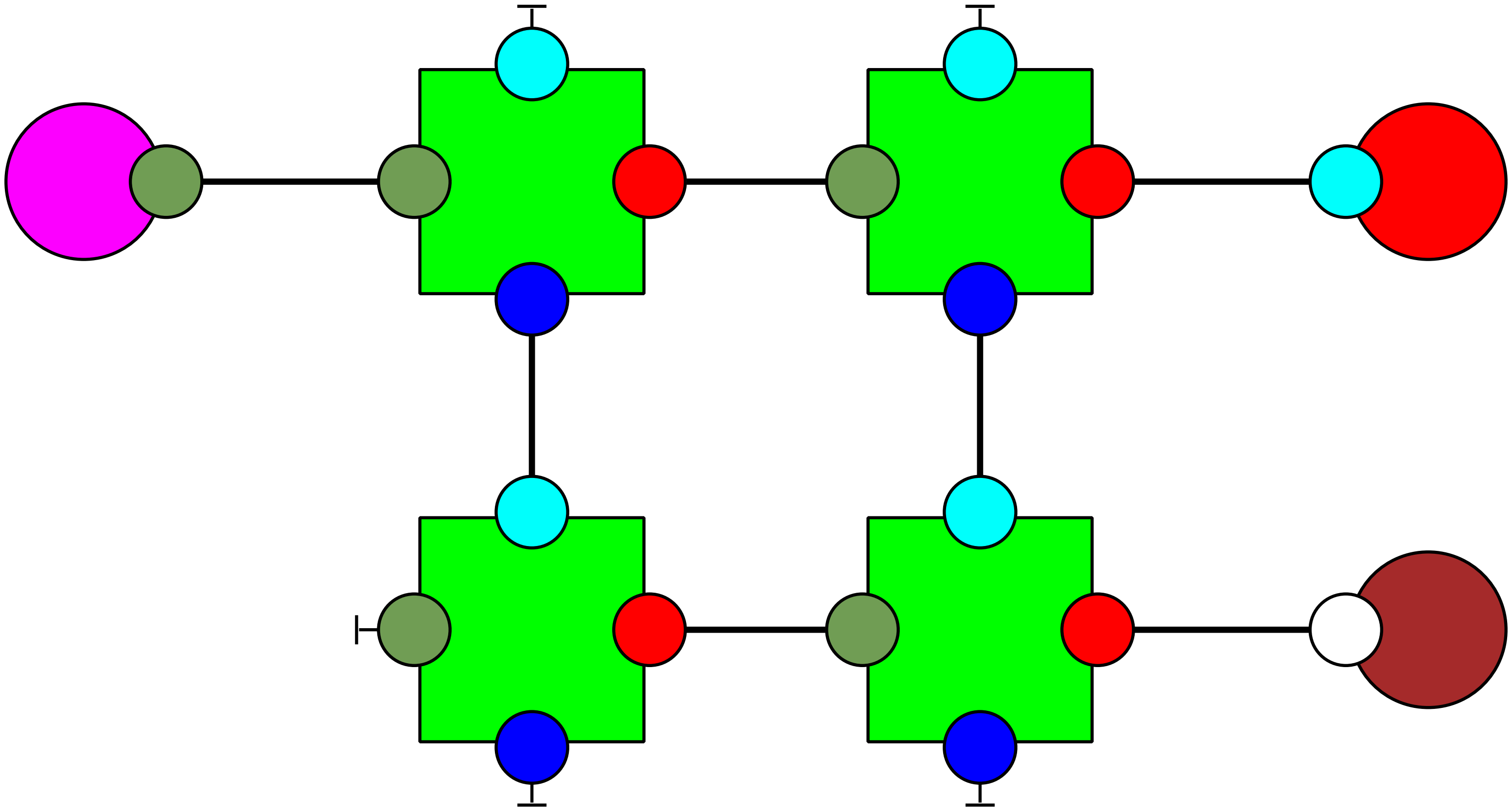
Other kinds of symmetries:
Homogeneous symmetries

We can compute a horizontal reflection.



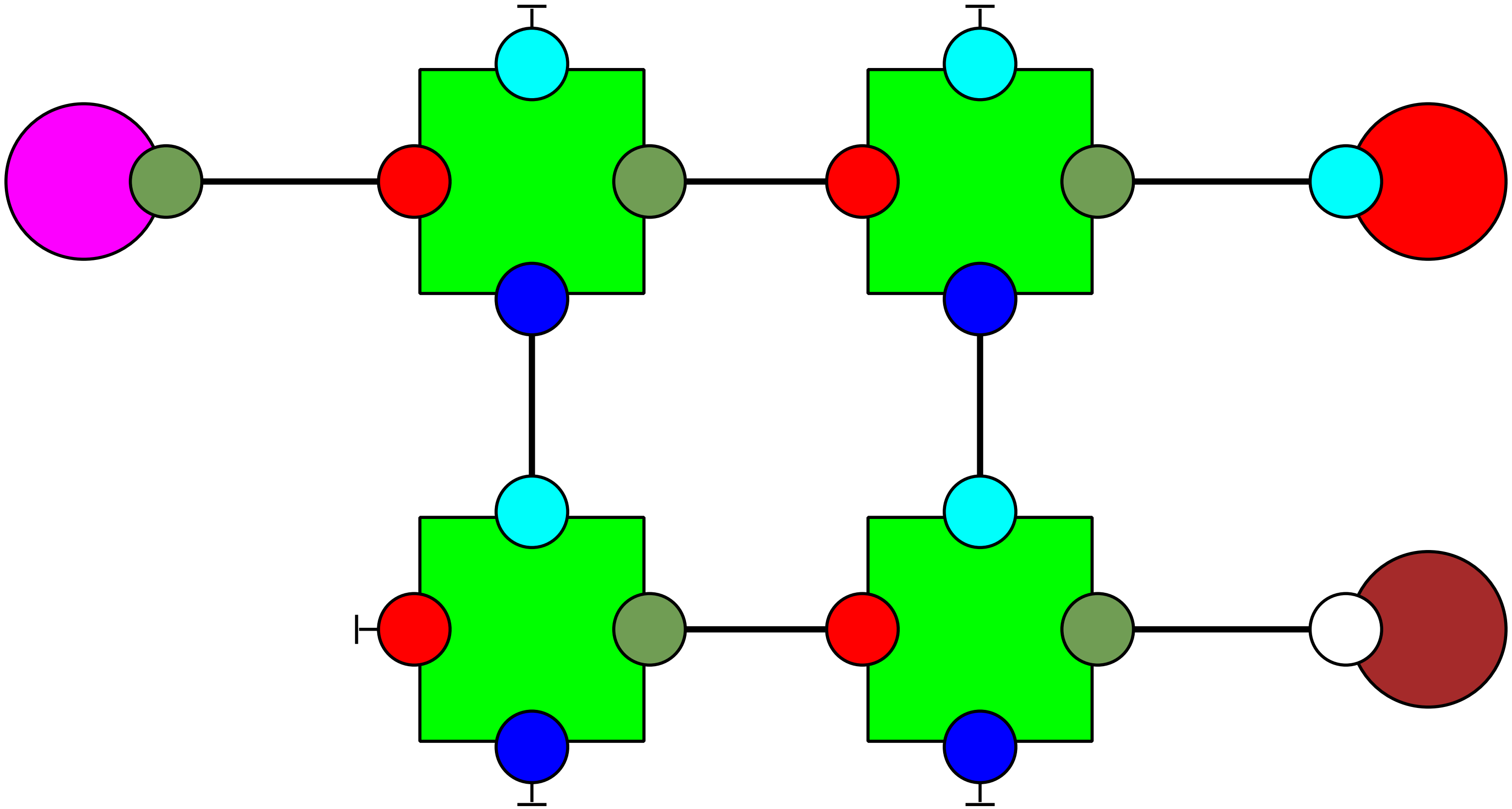
Other kinds of symmetries: Homogeneous symmetries

We can compute a vertical reflection.



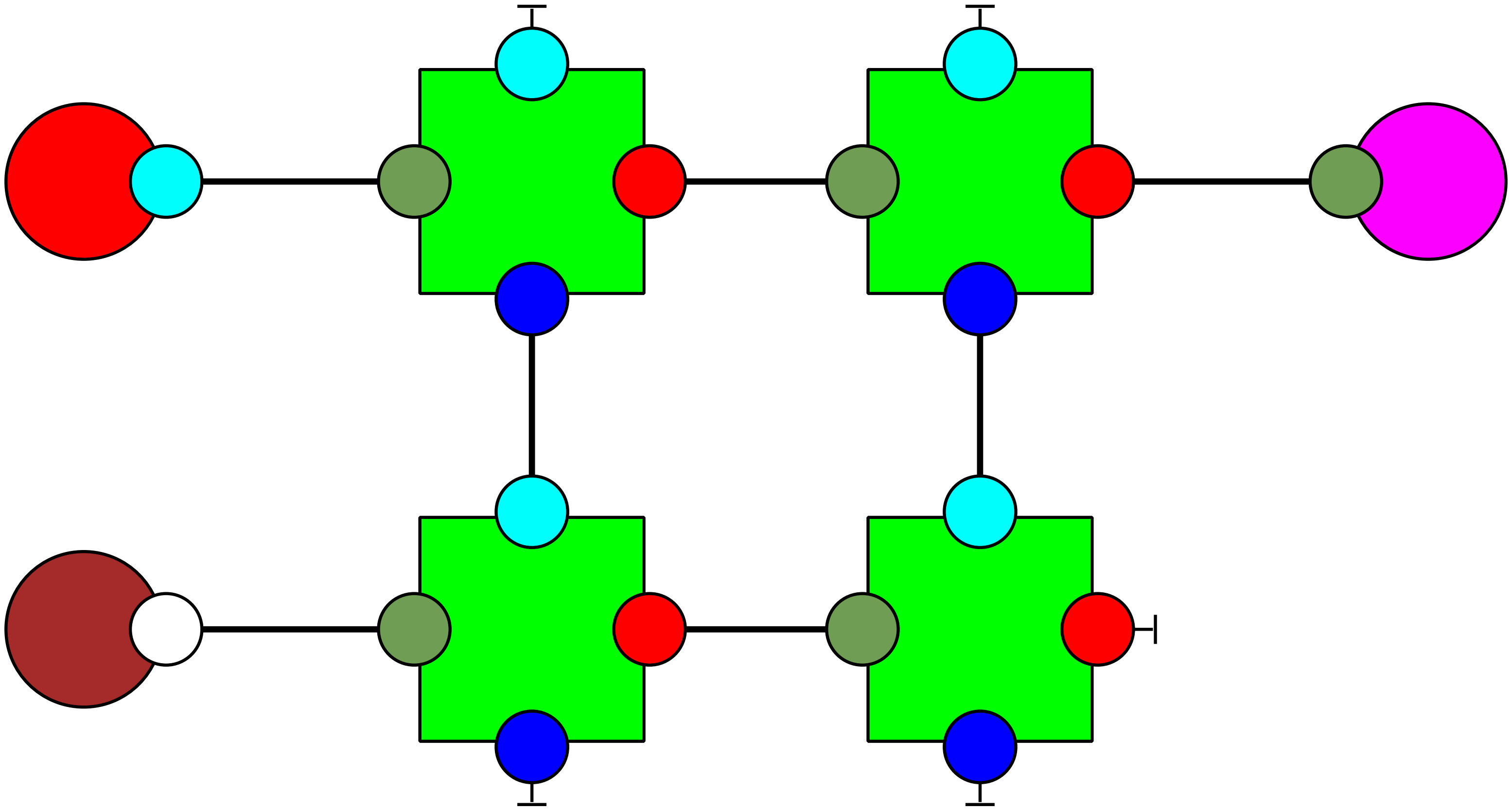
Other kinds of symmetries:
Homogeneous symmetries

We can compute a vertical reflection.



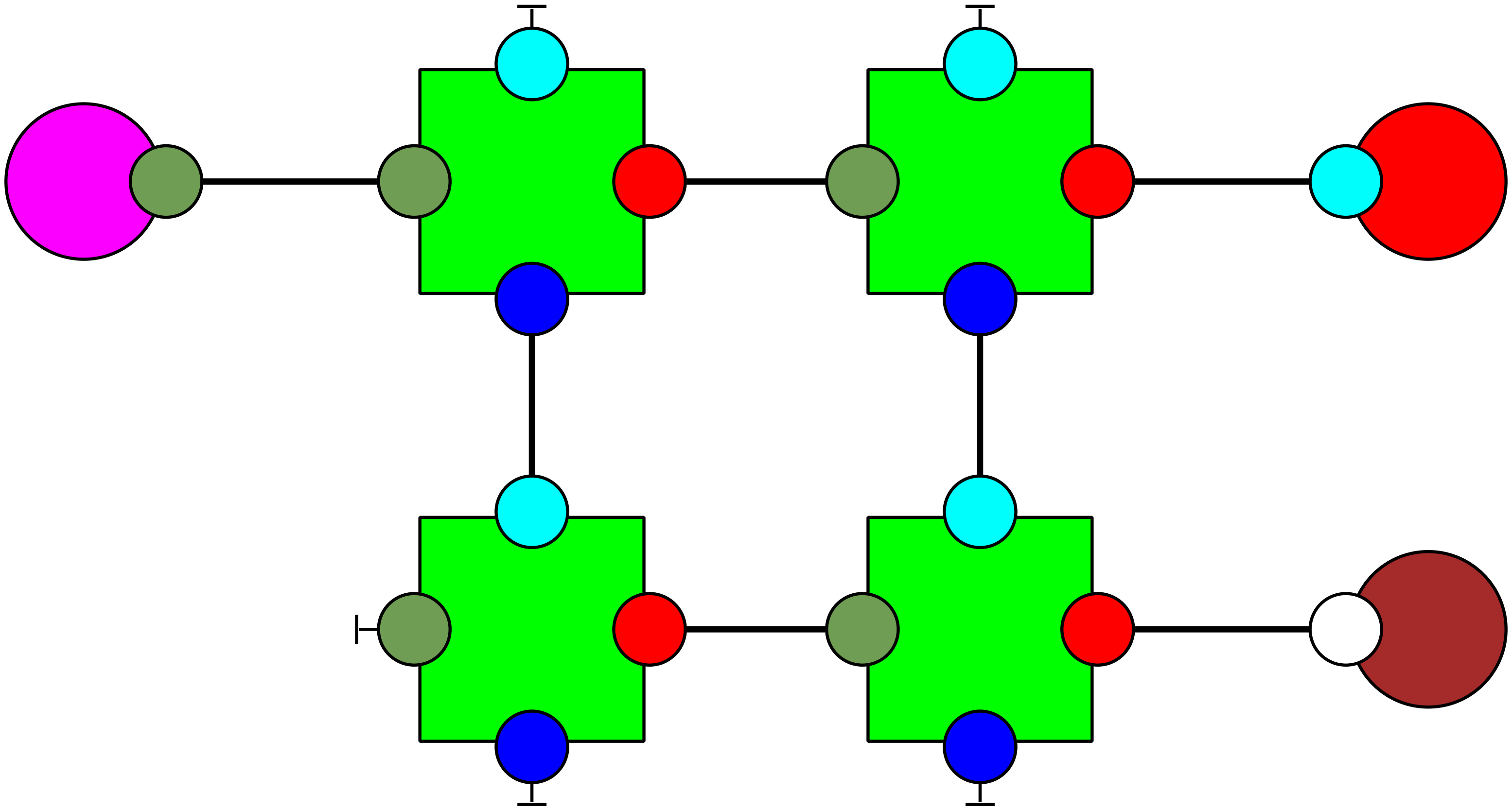
Other kinds of symmetries:
Homogeneous symmetries

We can compute a vertical reflection.



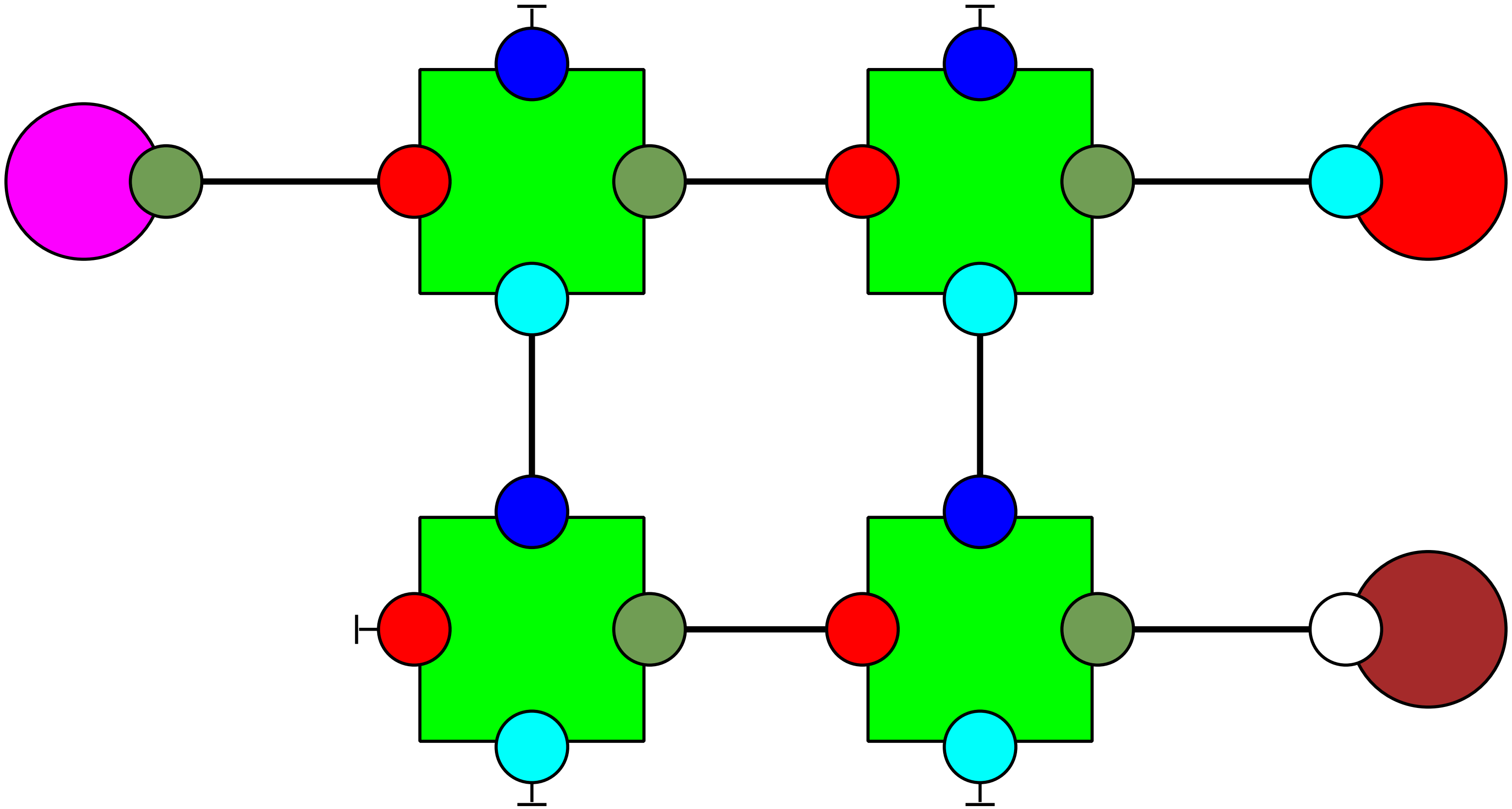
Other kinds of symmetries:
Homogeneous symmetries

We can compute both reflections.



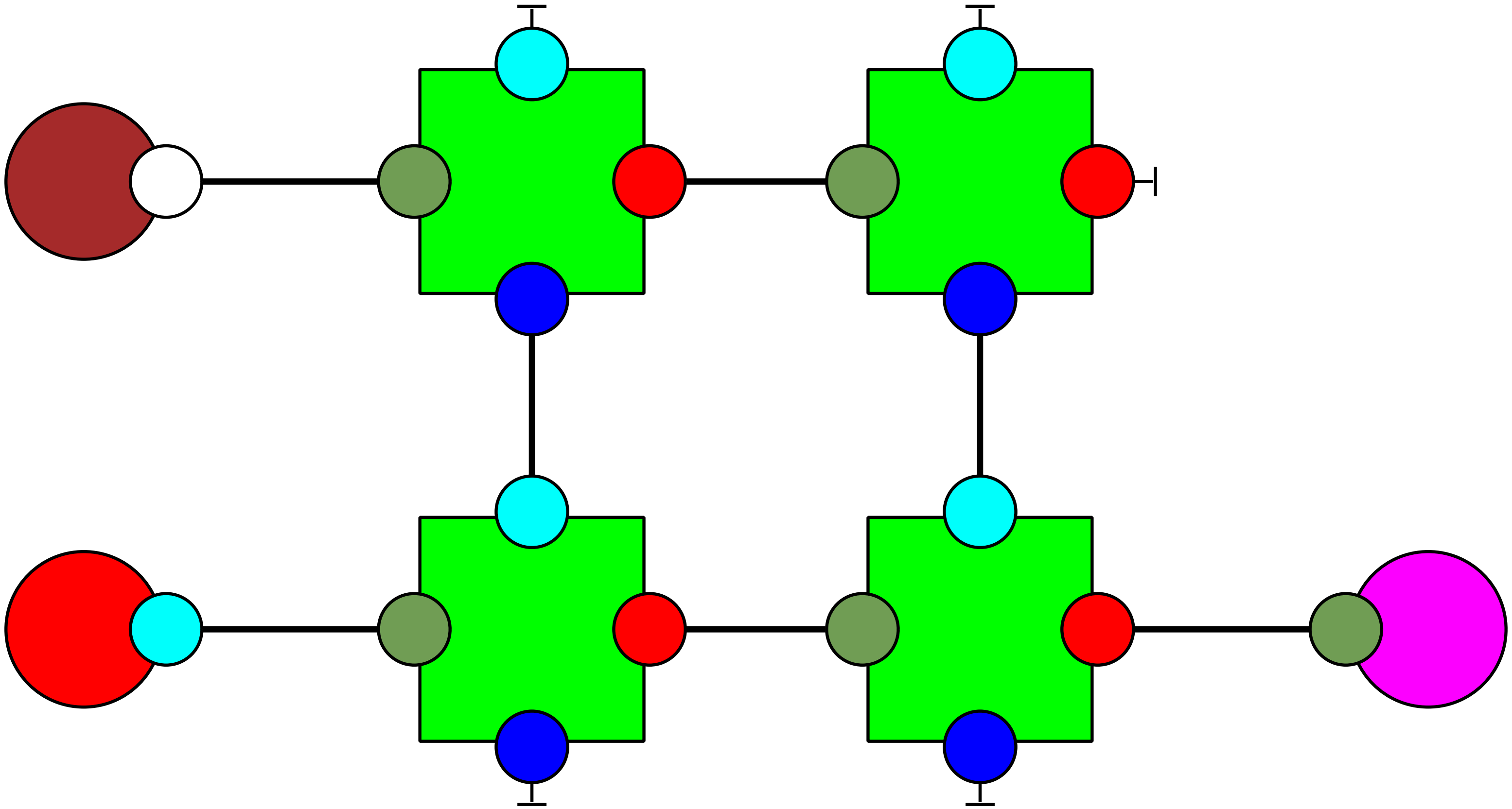
Other kinds of symmetries: Homogeneous symmetries

We can compute both reflections.



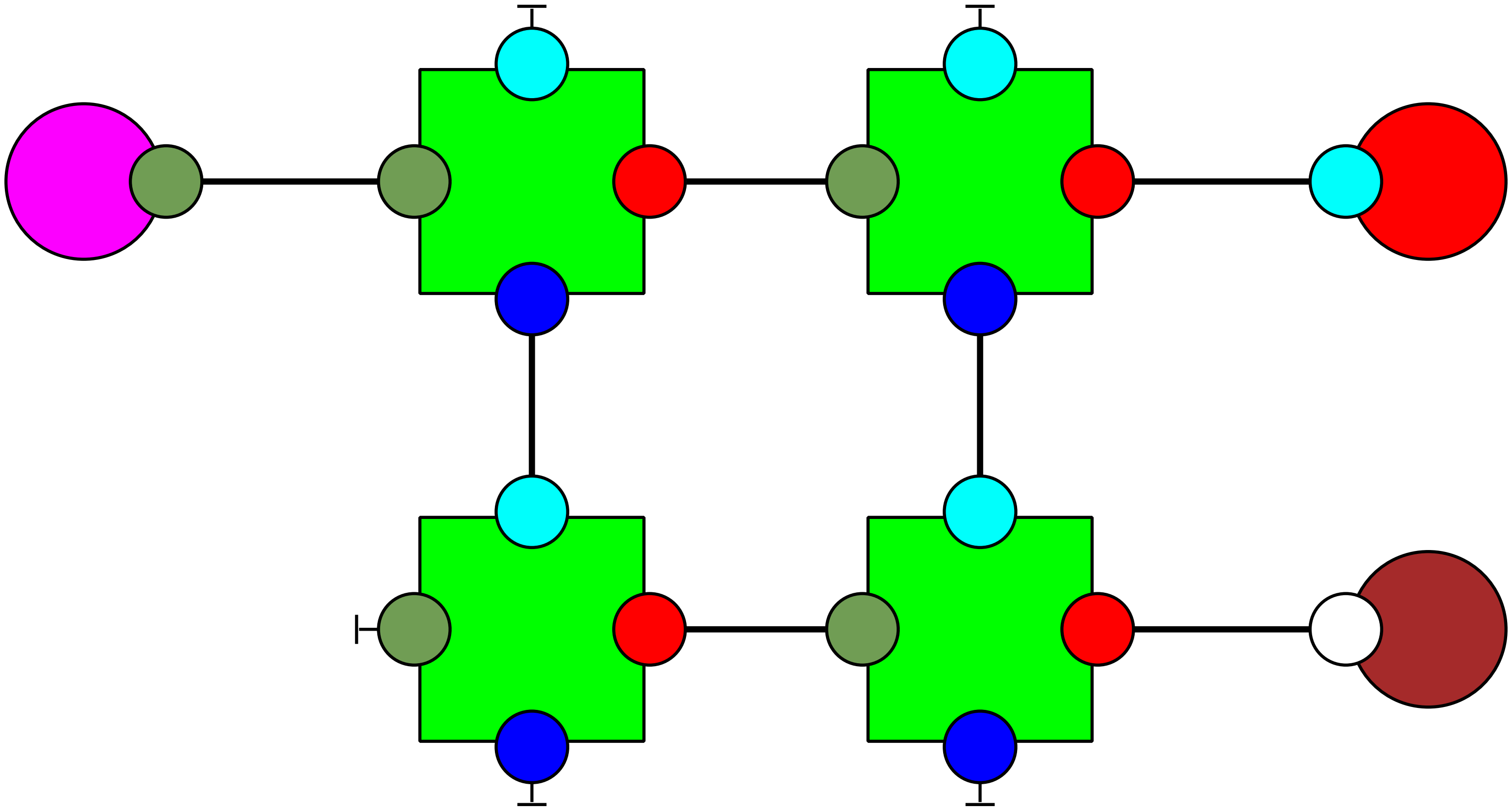
Other kinds of symmetries: Homogeneous symmetries

We can compute both reflections.



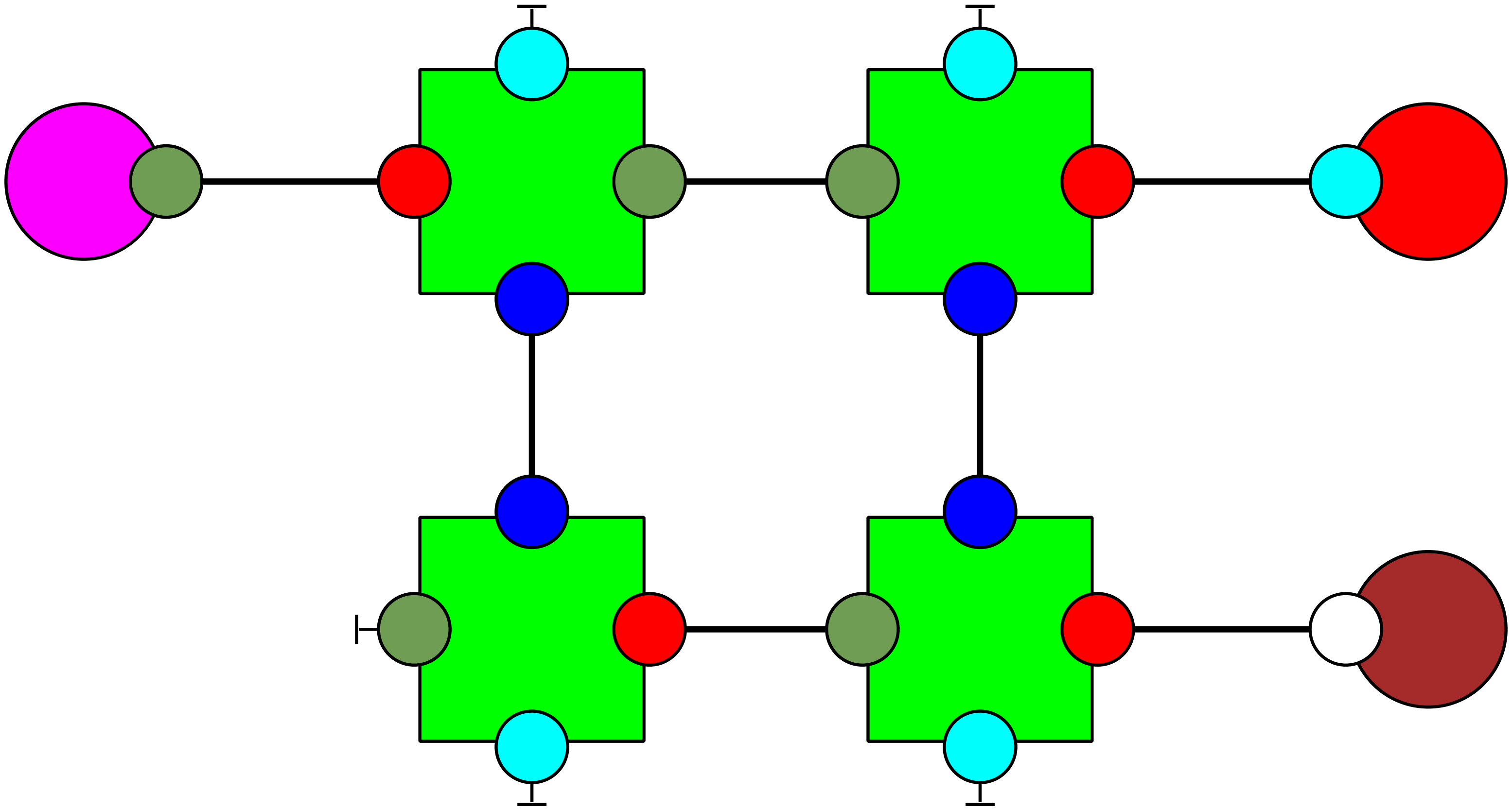
Other kinds of symmetries:
Homogeneous symmetries

But we cannot apply different permutations!!!.



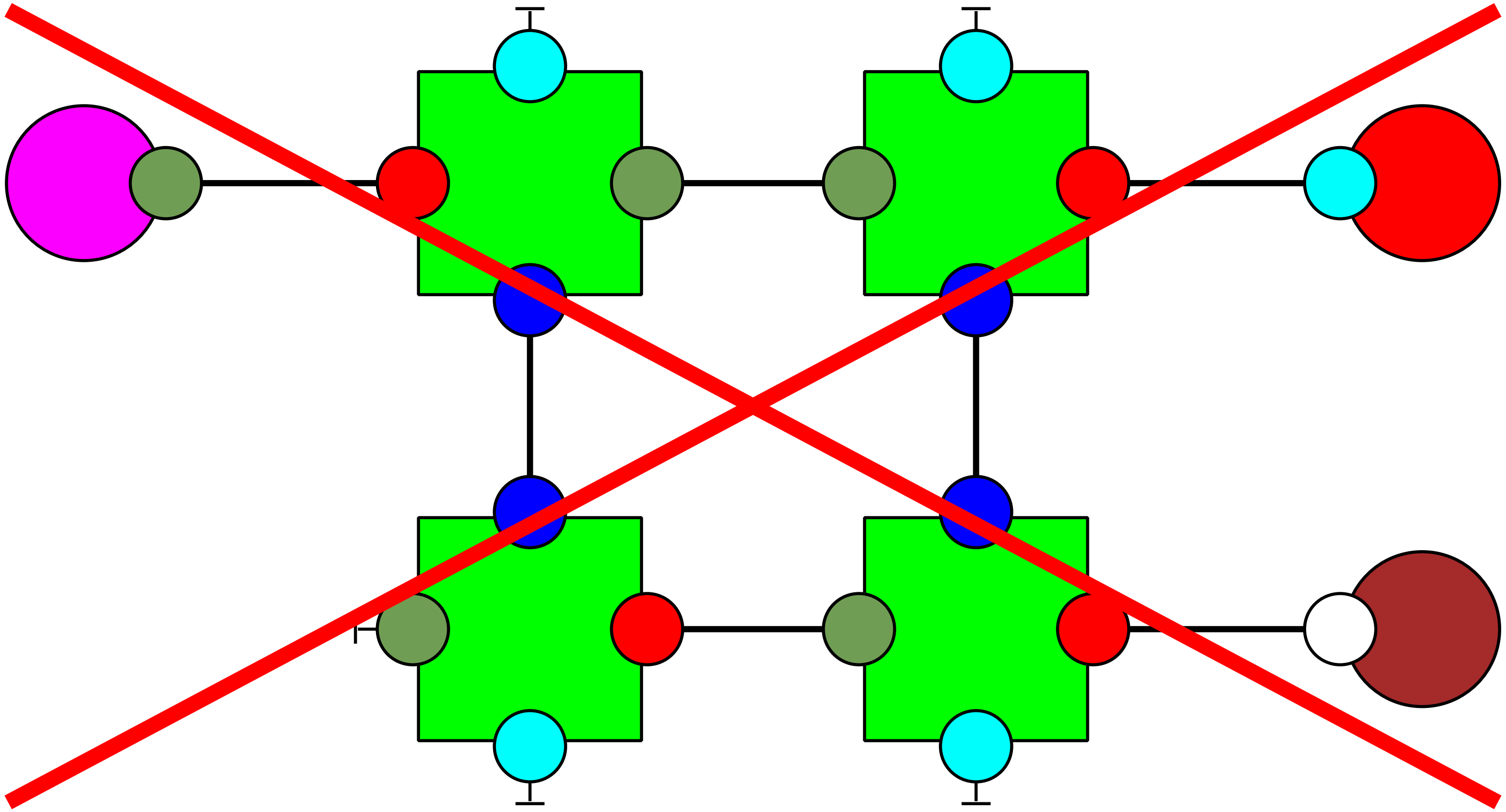
Other kinds of symmetries:
Homogeneous symmetries

But we cannot apply different permutations!!!.

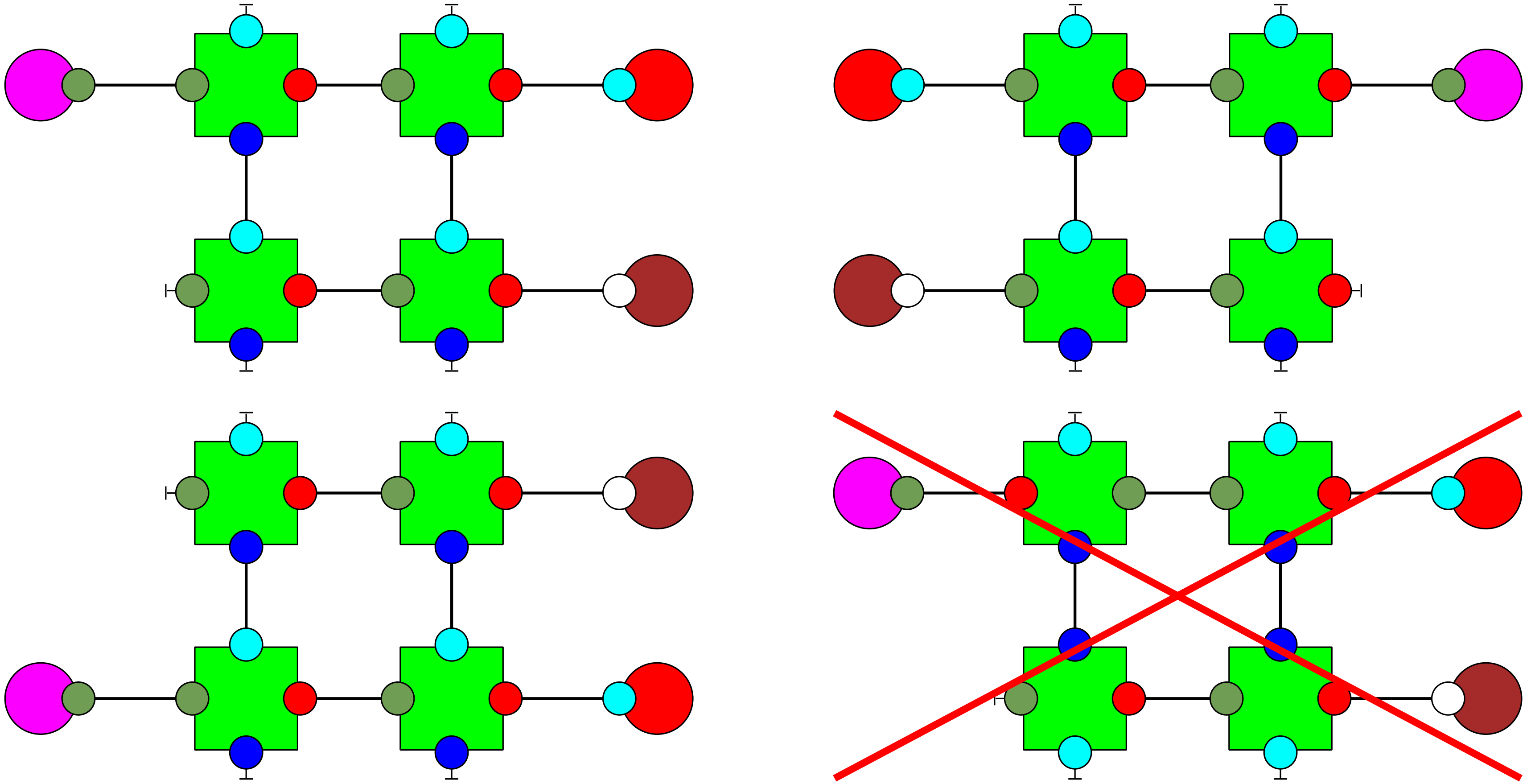


Other kinds of symmetries:
Homogeneous symmetries

But we cannot apply different permutations!!!.



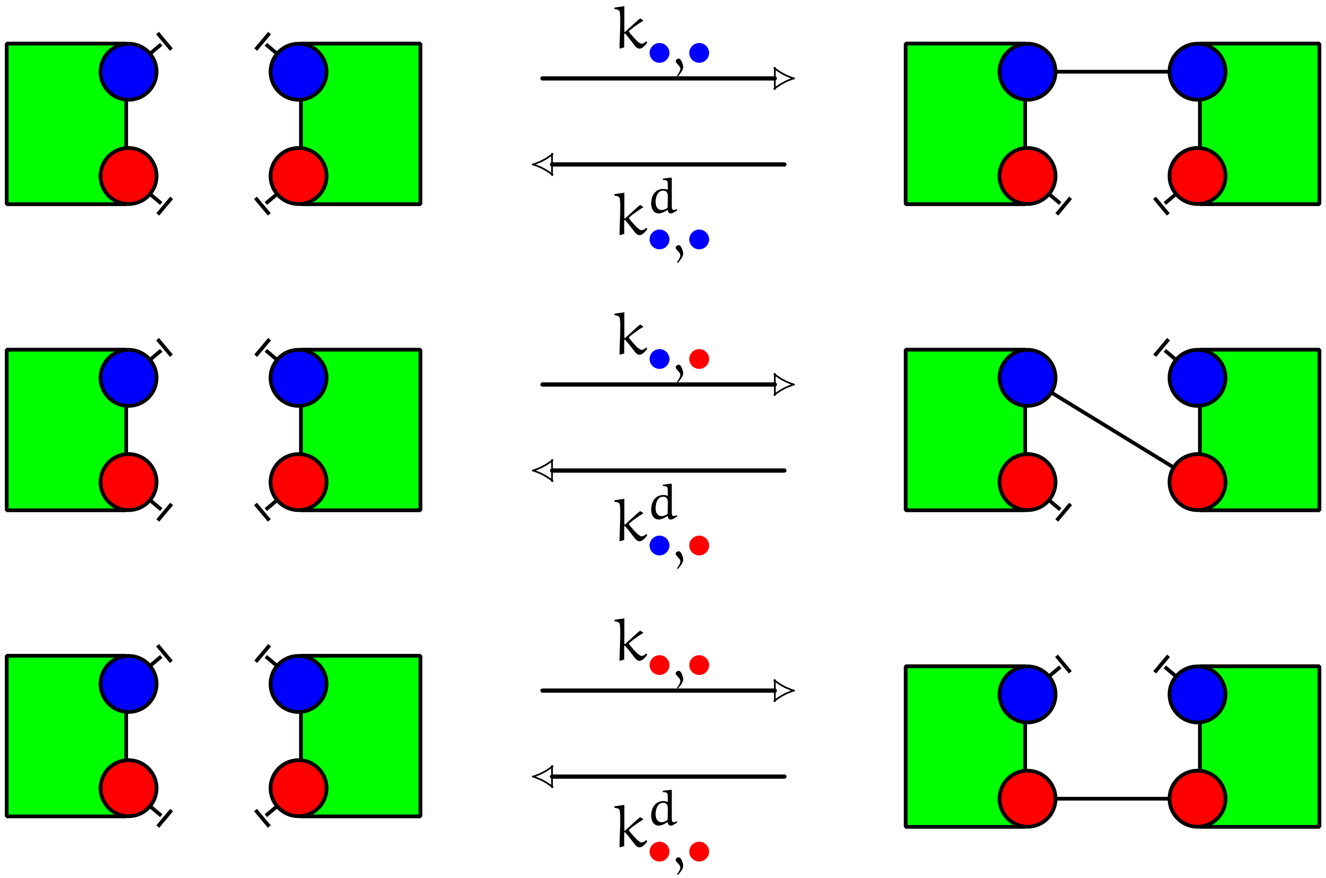
Other kinds of symmetries: Homogeneous symmetries



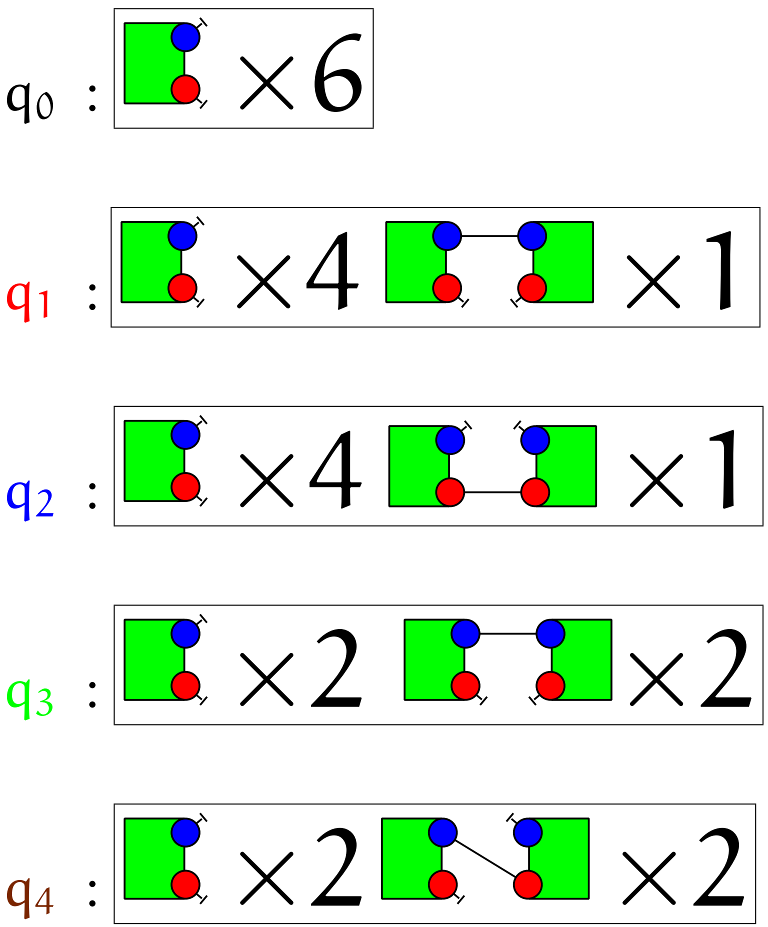
Overview

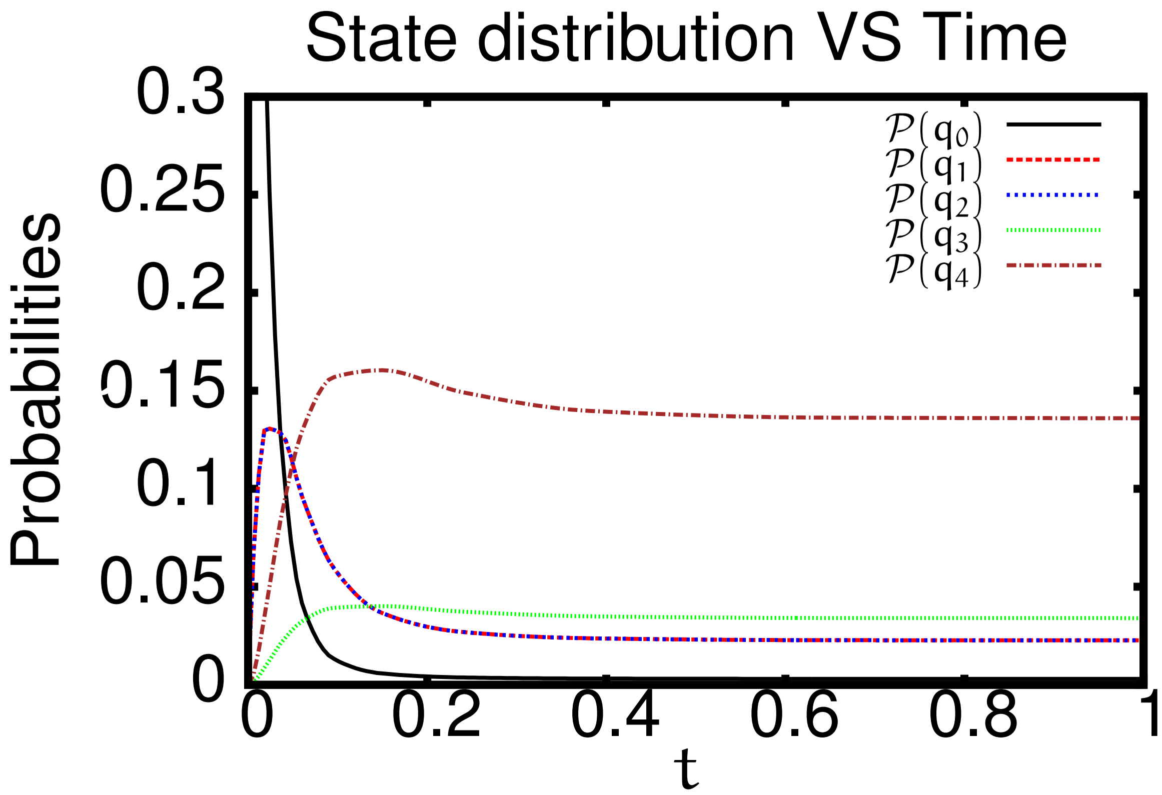
1. Context and motivations
2. Case study
   (a) Symetric model with symmetric initial state
   (b) Symmetric model with non-symmetric initial state
   (c) Non-symmetric model
3. Kappa semantics
4. Symmetries in site-graphs
5. Symmetric models
6. Conclusion

Case study



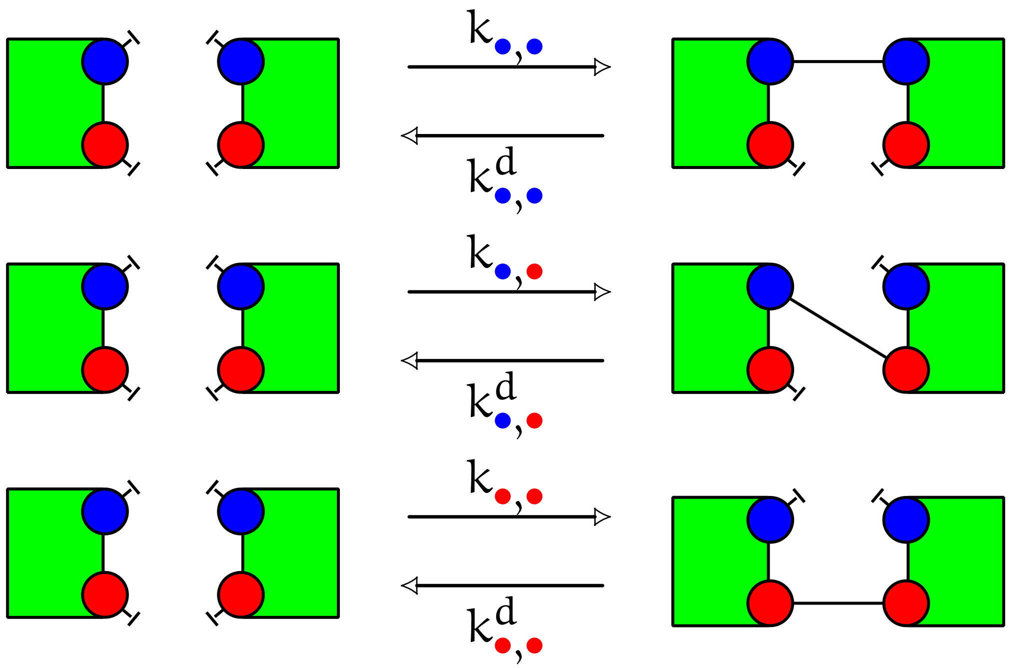
State distribution





with: $$\begin{cases} k_{\bullet,\bullet} = k_{\bullet,\bullet} = 1 \\ k_{\bullet,\bullet} = k^{d}_{\bullet,\bullet} = k^{d}_{\bullet,\bullet} = k^{d}_{\bullet,\bullet} = 2 \\ P(q_0 \mid t = 0) = 1 \end{cases}$$

Lumpability

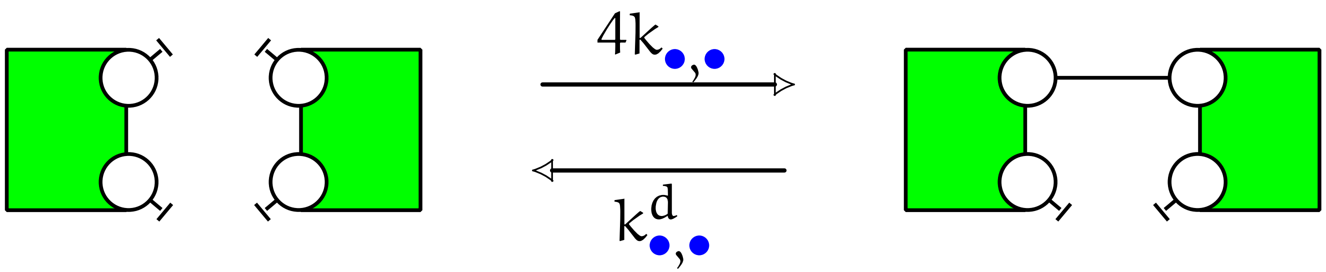


Whenever:

$$\begin{cases} 2k_{\color{blue}{\bullet},\color{blue}{\bullet}} = 2k_{\color{red}{\bullet},\color{red}{\bullet}} = k_{\color{blue}{\bullet},\color{red}{\bullet}} \\ k^{d}_{\color{blue}{\bullet},\color{blue}{\bullet}} = k^{d}_{\color{blue}{\bullet},\color{red}{\bullet}} = k^{d}_{\color{red}{\bullet},\color{red}{\bullet}} \end{cases}$$

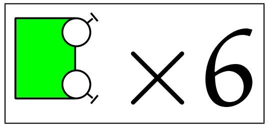
We can lump the system.

Lumped system

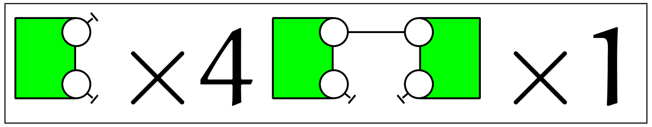


Macrostate distribution

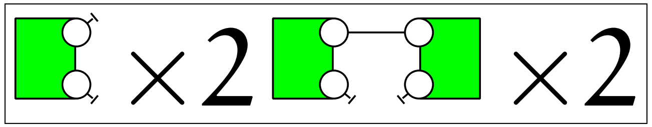
Q_0 :



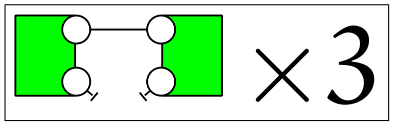
Q_1 :

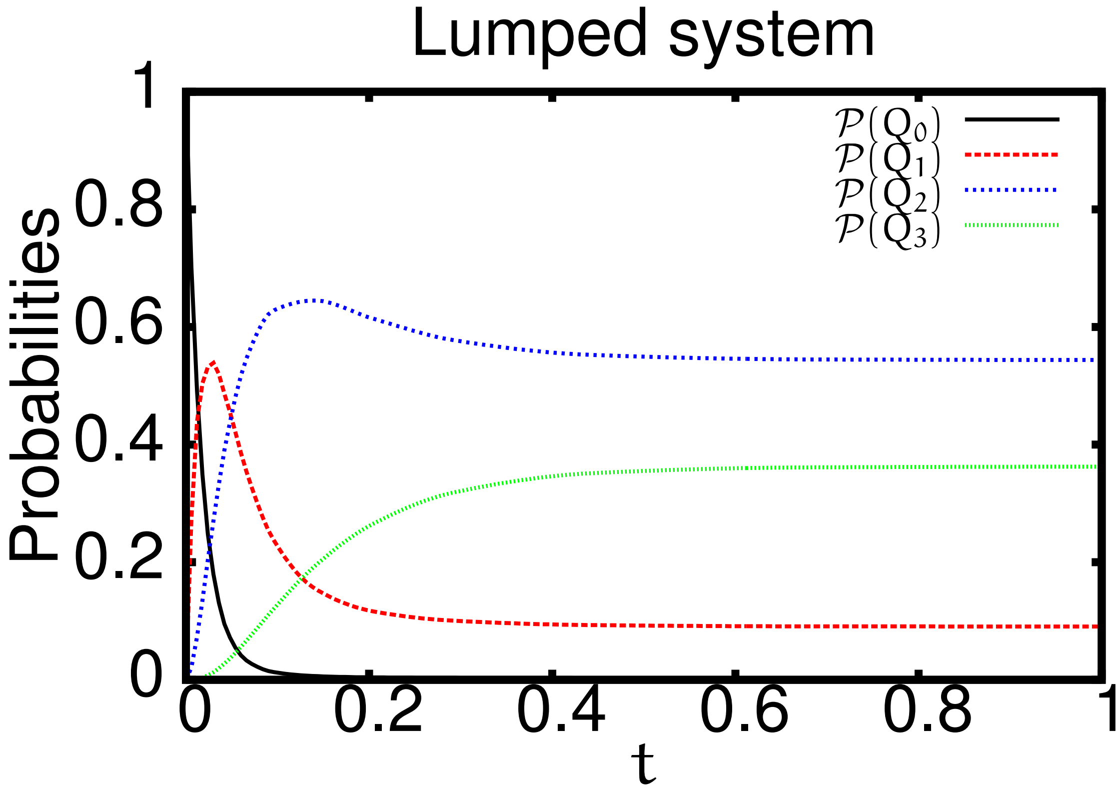


Q_2 :



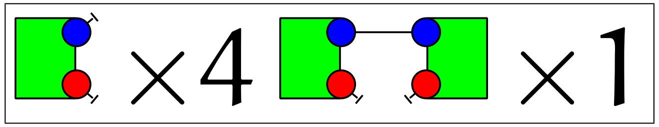
Q_3 :

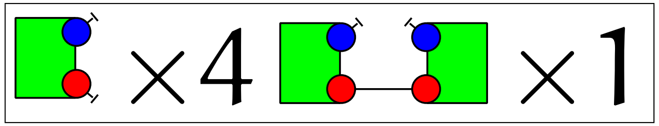


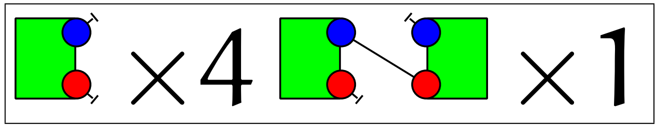


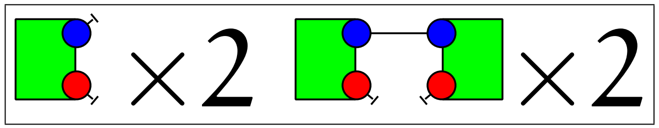
with: $\begin{cases} k_{\bullet,\bullet} = 1 \\ k^{d}_{\bullet,\bullet} = 2 \\ P(q_0 \mid t = 0) = 1 \end{cases}$

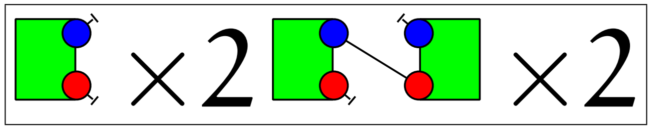
Probability ratios

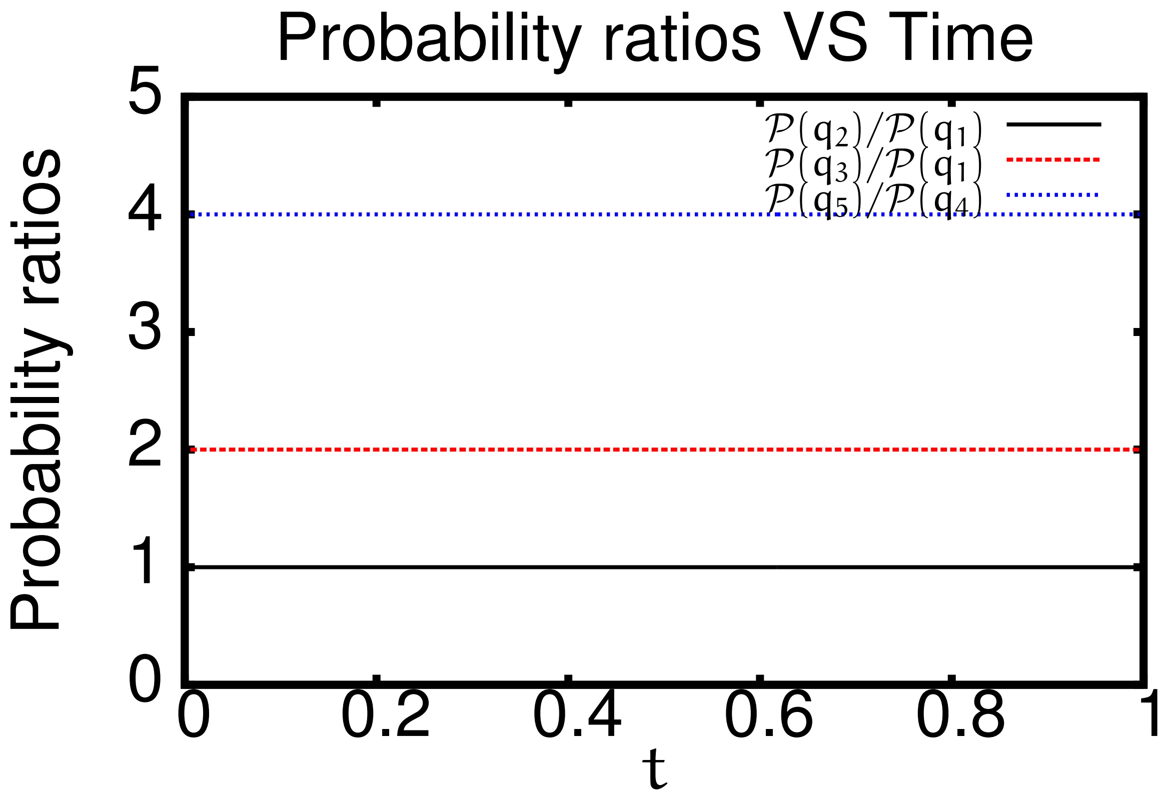
q_1 : 

q_2 : 

q_3 : 

q_4 : 

q_5 : 

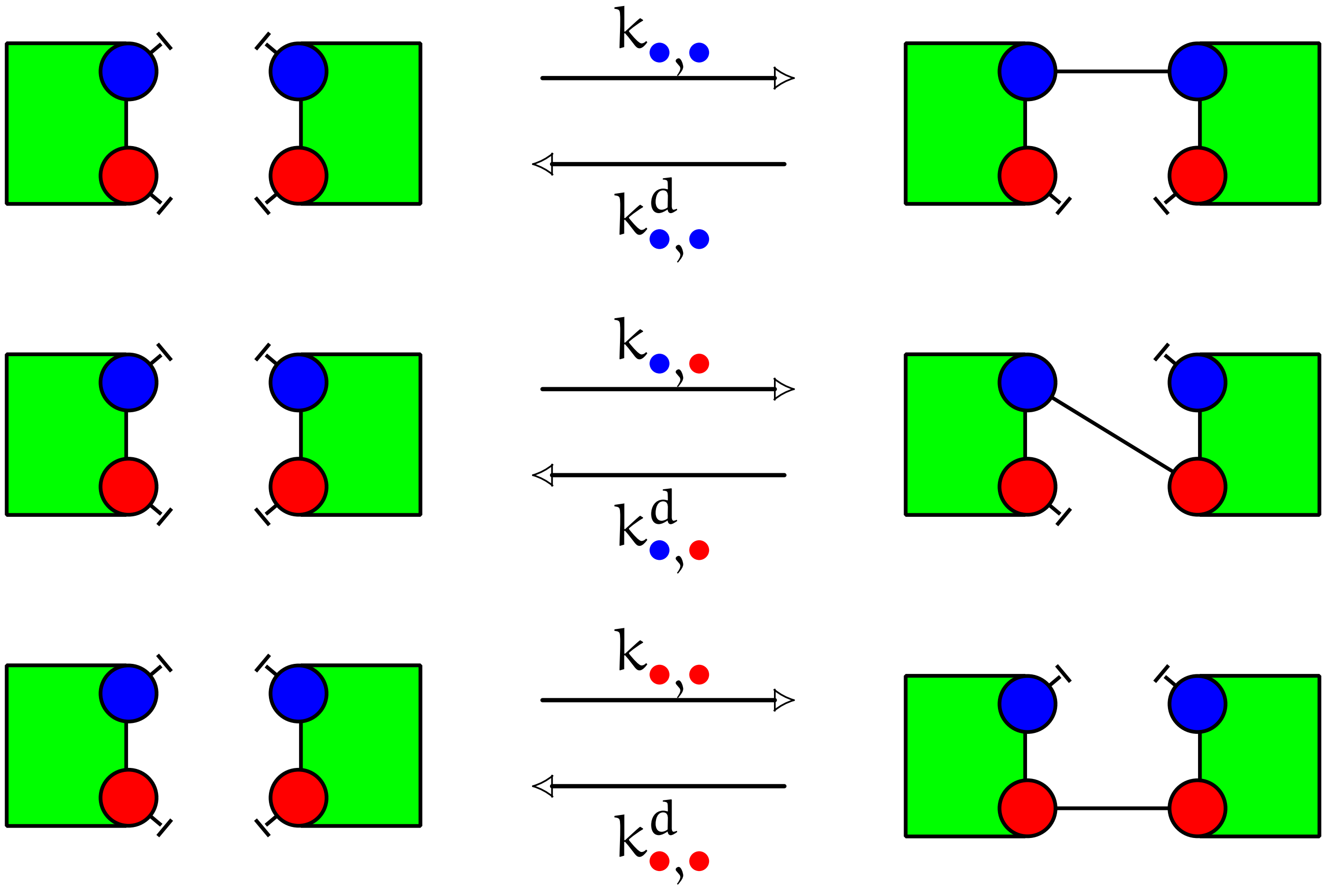


with: $\begin{cases} k_{\textcolor{blue}{\bullet},\textcolor{blue}{\bullet}} = k_{\textcolor{red}{\bullet},\textcolor{red}{\bullet}} = 1 \\ k_{\textcolor{blue}{\bullet},\textcolor{red}{\bullet}} = k^{d}_{\textcolor{blue}{\bullet},\textcolor{blue}{\bullet}} = k^{d}_{\textcolor{red}{\bullet},\textcolor{red}{\bullet}} = k^{d}_{\textcolor{blue}{\bullet},\textcolor{red}{\bullet}} = 2 \\ P(q_0 \mid t = 0) = 1 \end{cases}$

Overview

1. Context and motivations
2. Case study
   (a) Symetric model with symmetric initial state
   (b) Symmetric model with non-symmetric initial state
   (c) Non-symmetric model
3. Kappa semantics
4. Symmetries in site-graphs
5. Symmetric models
6. Conclusion

Model



State distribution

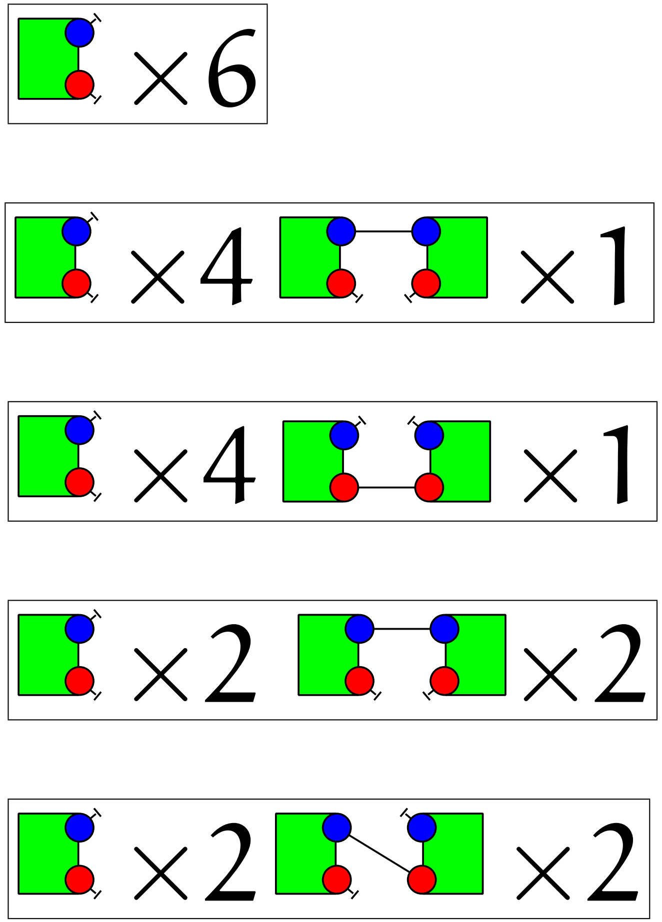
q_0 :

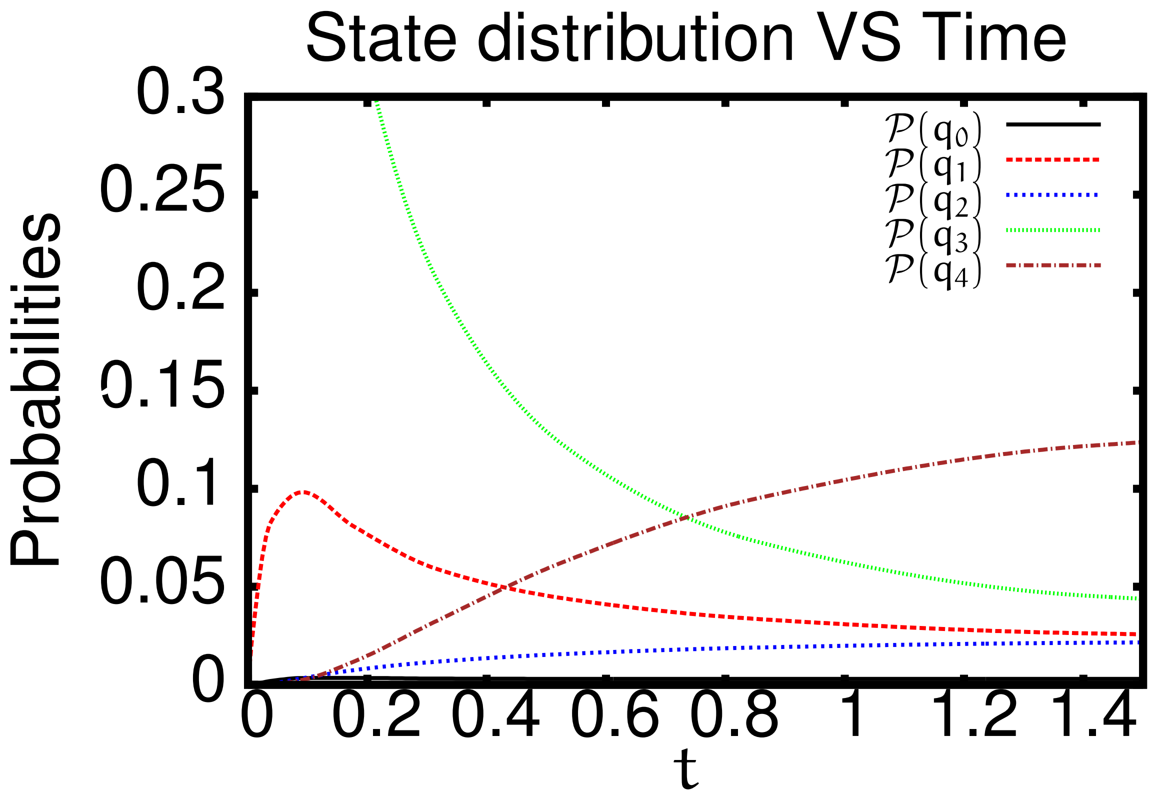
q_1 :

q_2 :

q_3 :

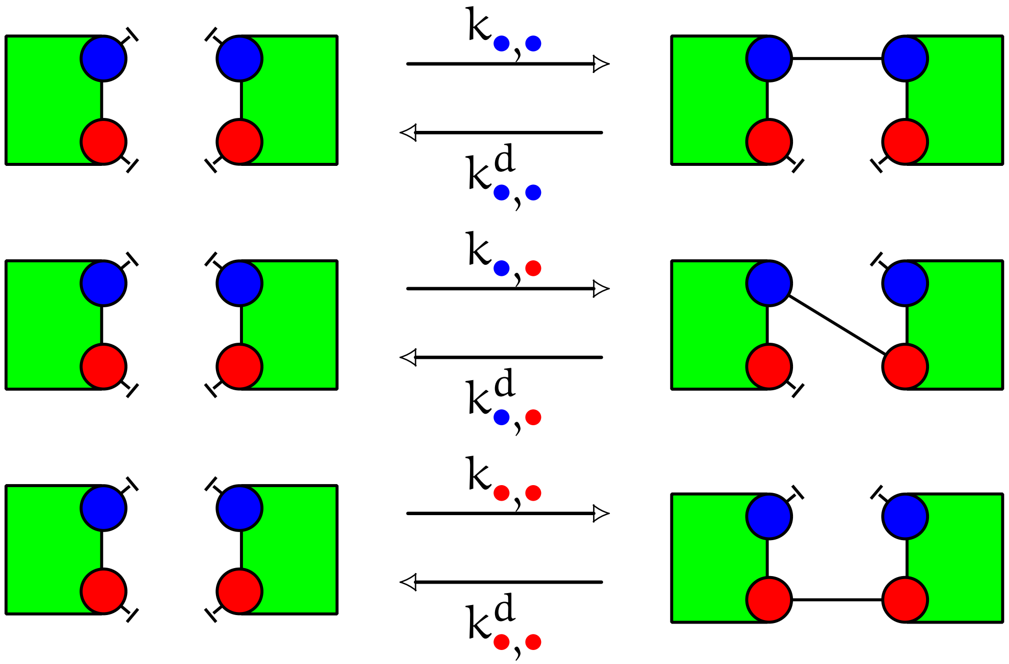
q_4 :





with: $$\begin{cases} k_{{\color{blue}\bullet},{\color{blue}\bullet}} = k_{{\color{red}\bullet},{\color{red}\bullet}} = 1 \\ k_{{\color{blue}\bullet},{\color{red}\bullet}} = k^d_{{\color{blue}\bullet},{\color{blue}\bullet}} = k^d_{{\color{red}\bullet},{\color{red}\bullet}} = k^d_{{\color{blue}\bullet},{\color{red}\bullet}} = 2 \\ P(q_3 \mid t = 0) = 1 \end{cases}$$

Lumpability

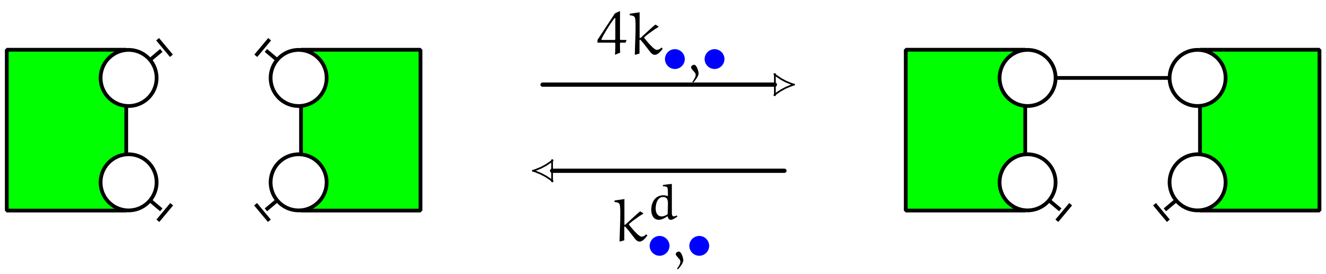


Whenever:

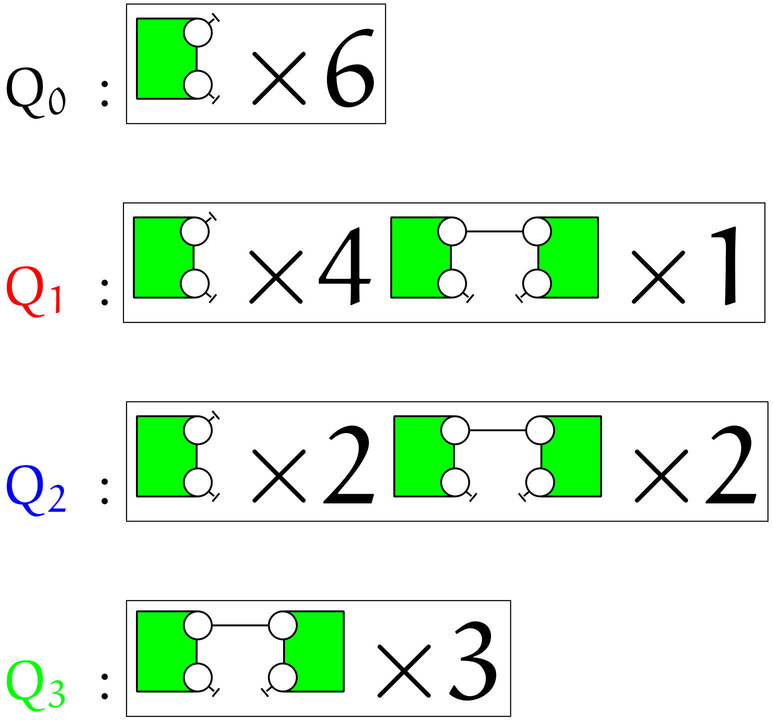
$$\begin{cases} 2k_{{\color{blue}\bullet},{\color{blue}\bullet}} = 2k_{{\color{red}\bullet},{\color{red}\bullet}} = k_{{\color{blue}\bullet},{\color{red}\bullet}} \\ k^{d}_{{\color{blue}\bullet},{\color{blue}\bullet}} = k^{d}_{{\color{blue}\bullet},{\color{red}\bullet}} = k^{d}_{{\color{red}\bullet},{\color{red}\bullet}} \end{cases}$$

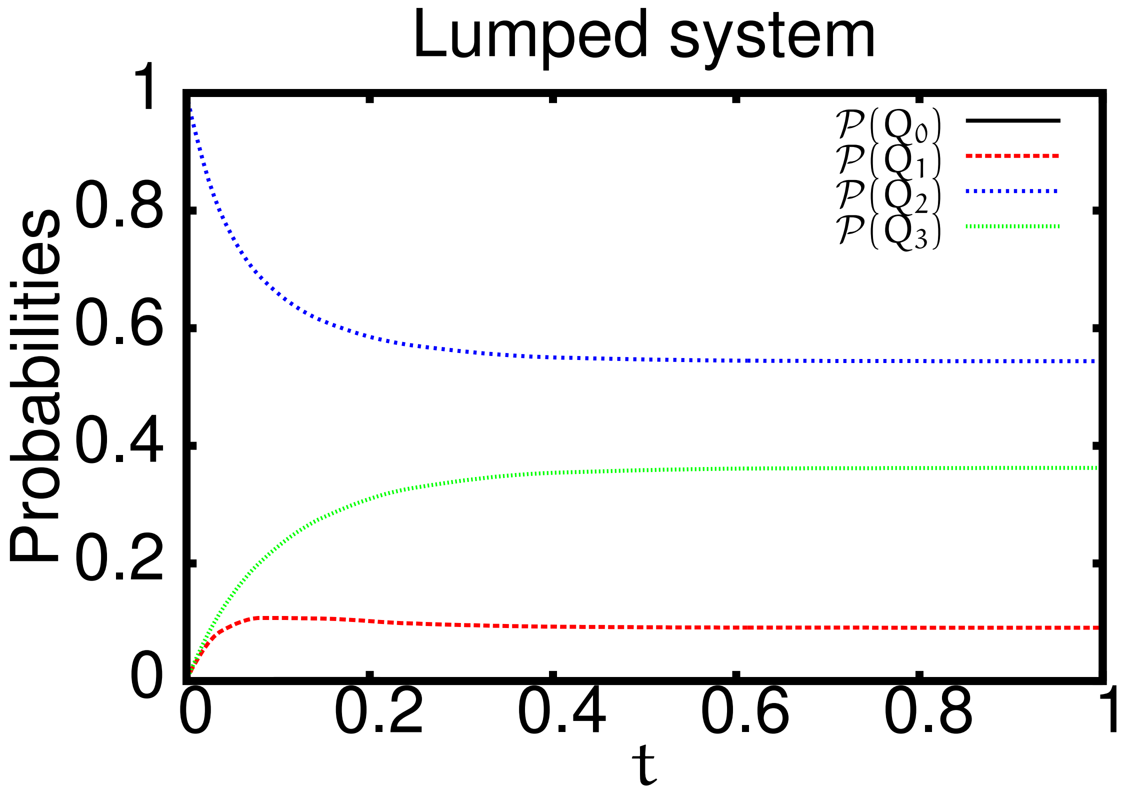
We can lump the system.

Lumped system

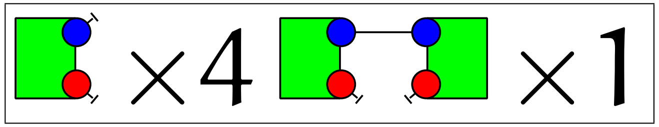


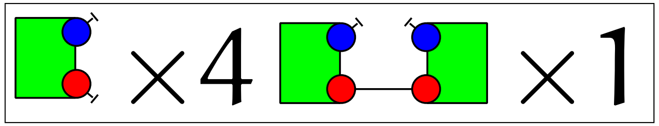
Macrostate distribution

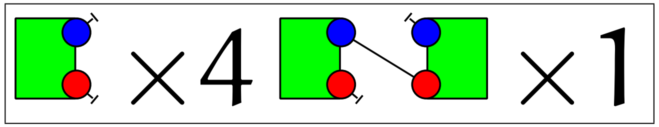


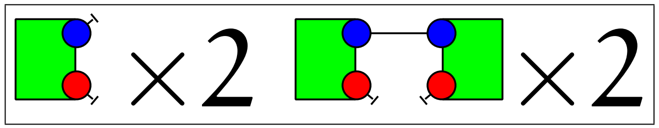


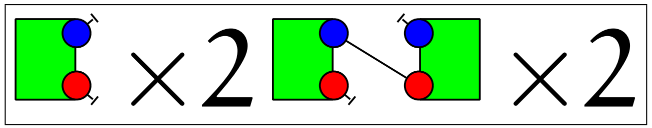
Probability ratios (wrong initial condition)

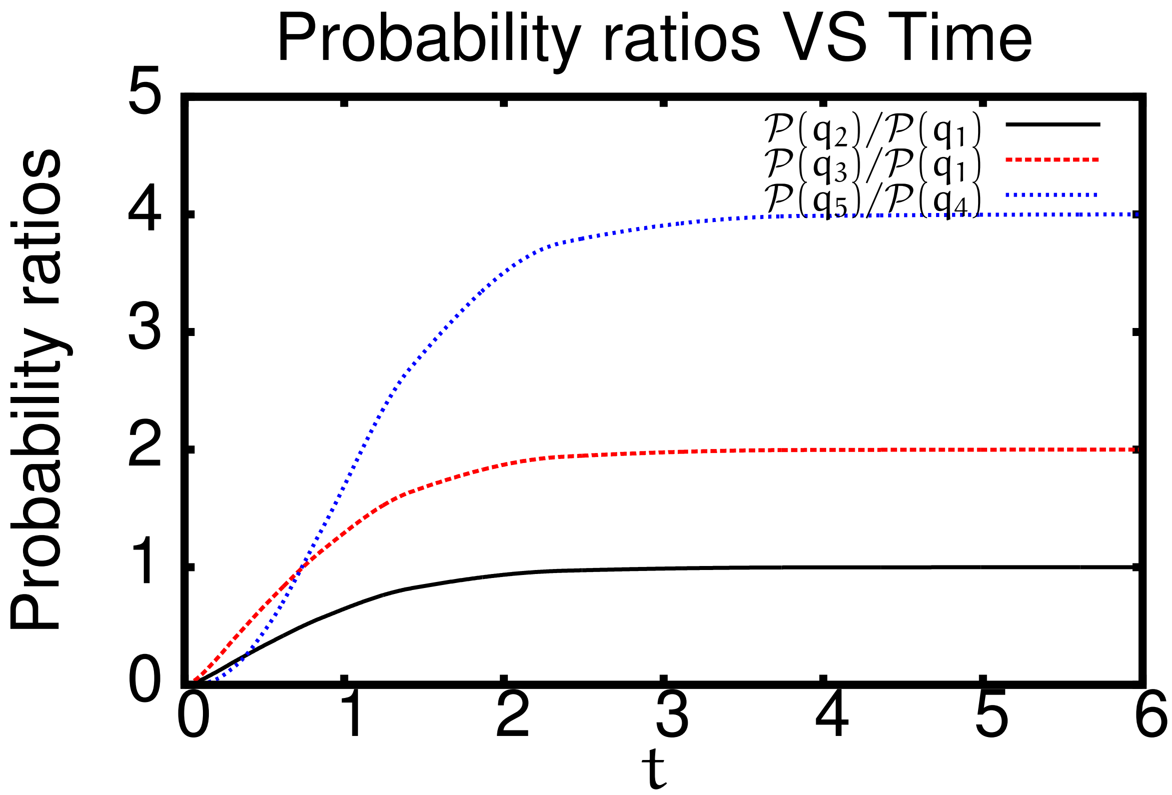
q_1 : 

q_2 : 

q_3 : 

q_4 : 

q_5 : 

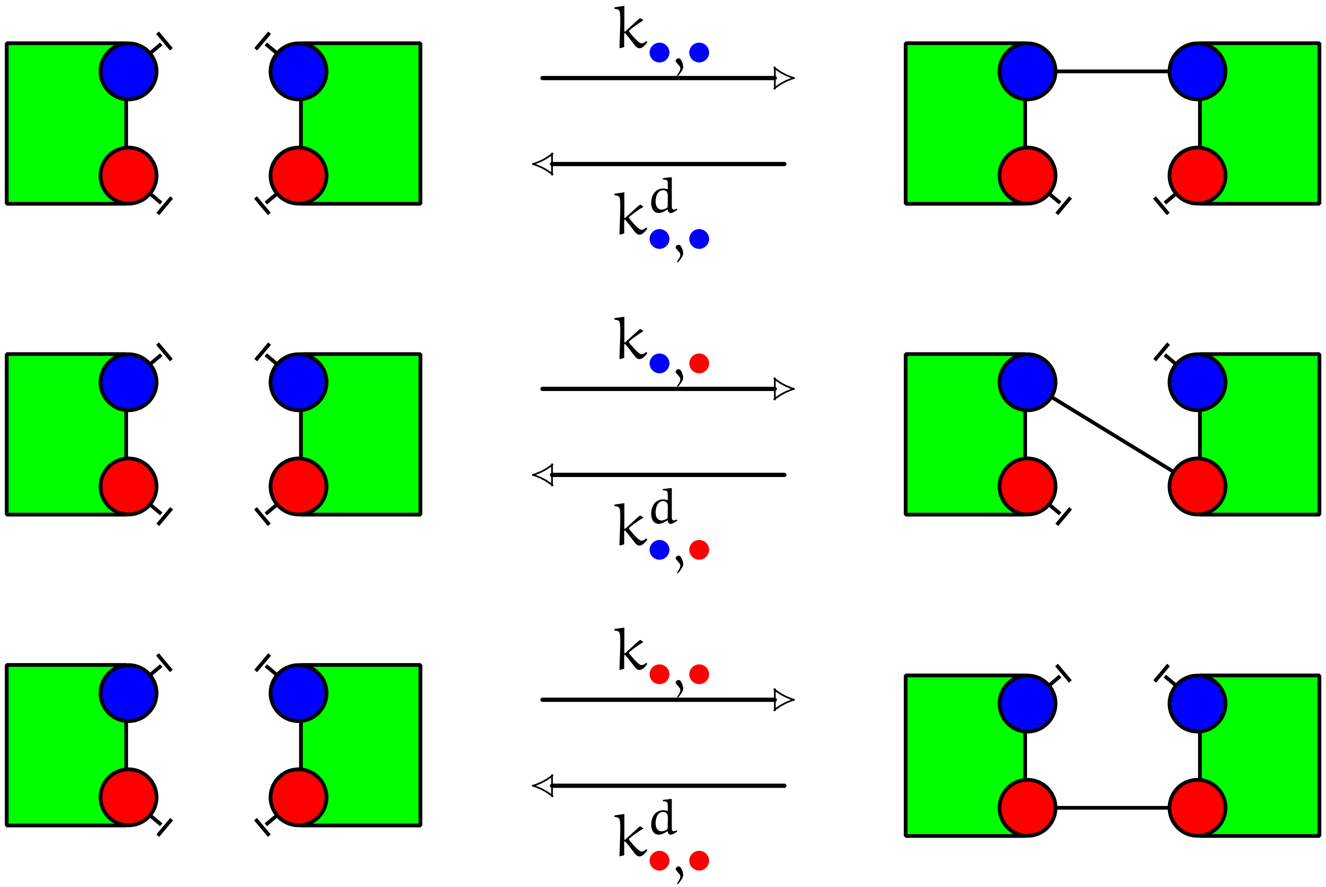


with: $$\begin{cases} k_{\bullet,\bullet} = k_{\bullet,\bullet} = 1 \\ k_{\bullet,\bullet} = k^{d}_{\bullet,\bullet} = k^{d}_{\bullet,\bullet} = k^{d}_{\bullet,\bullet} = 2 \\ P(q_4 \mid t = 0) = 1 \end{cases}$$

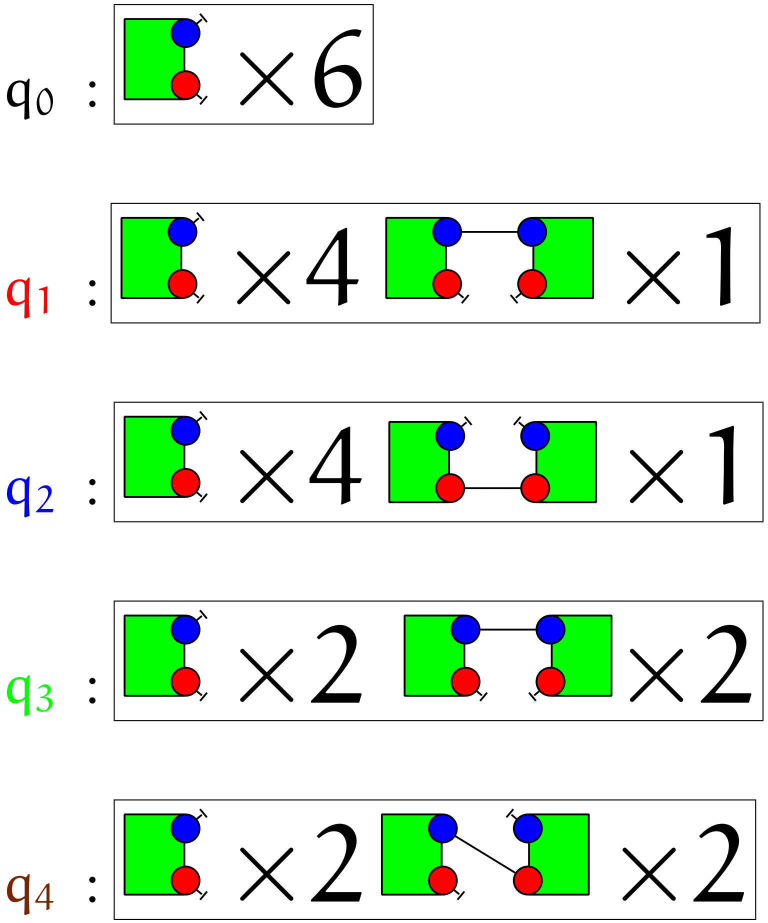
Overview

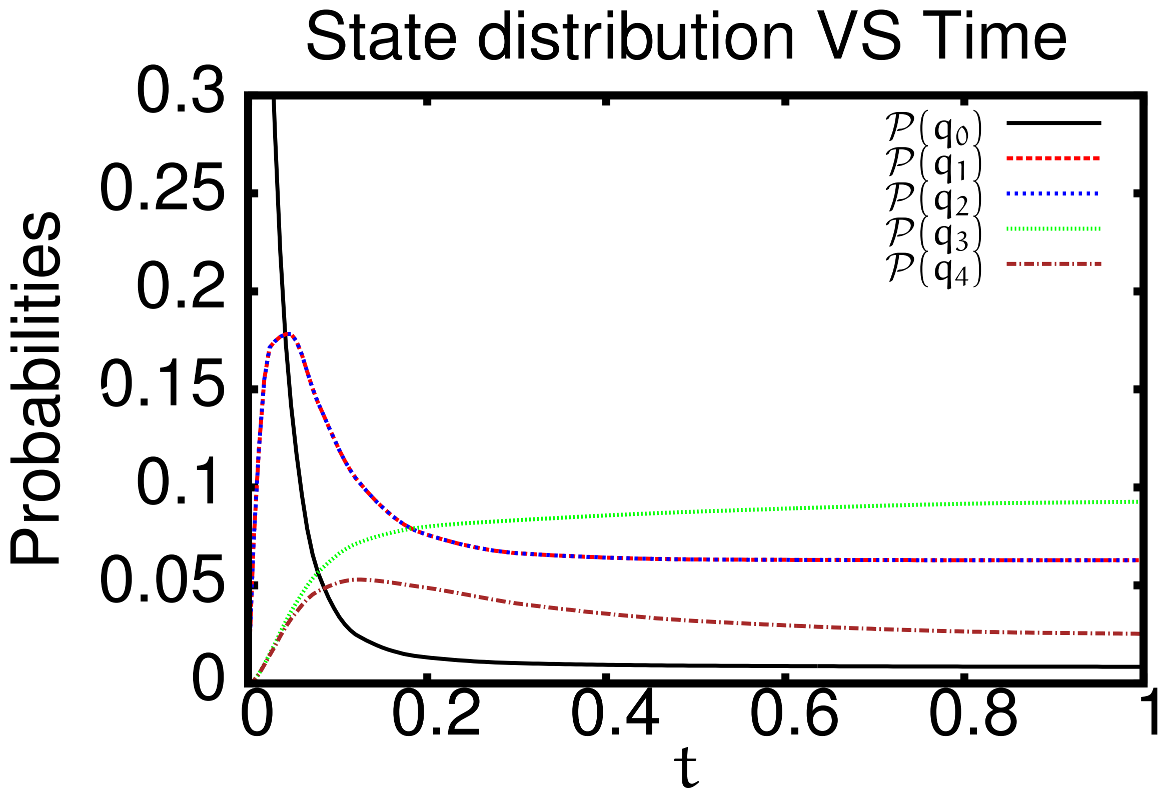
1. Context and motivations
2. Case study
    (a) Symetric model with symmetric initial state
    (b) Symmetric model with non-symmetric initial state
    (c) Non-symmetric model
3. Kappa semantics
4. Symmetries in site-graphs
5. Symmetric models
6. Conclusion

Model



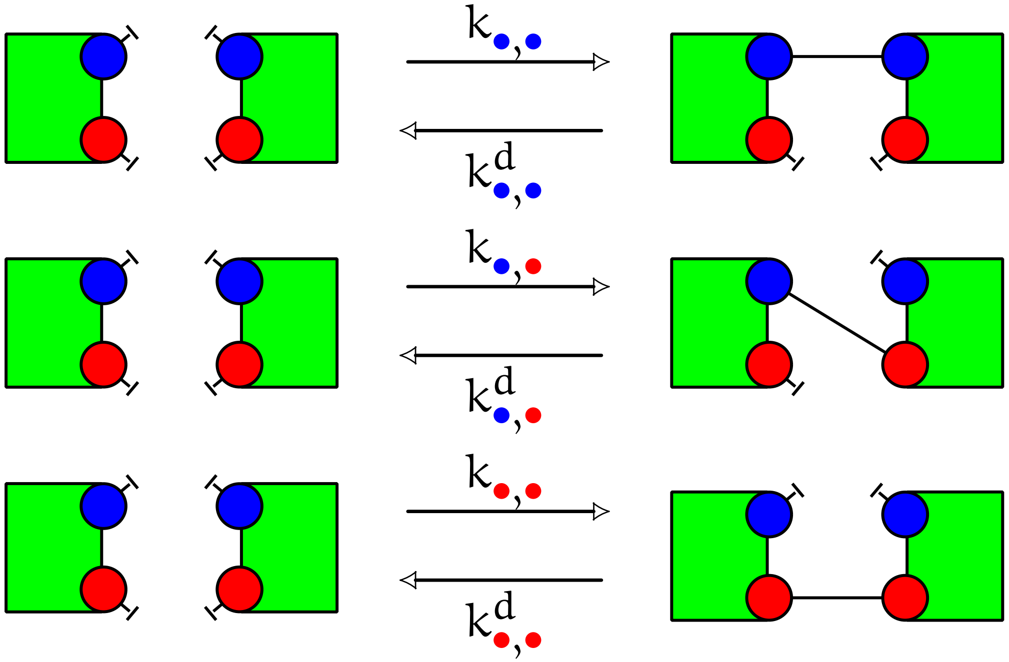
State distribution





with:
$$\begin{cases} k_{{\color{blue}\bullet},{\color{blue}\bullet}} = k_{{\color{red}\bullet},{\color{red}\bullet}} = k_{{\color{blue}\bullet},{\color{red}\bullet}} = 1 \\ k^{d}_{{\color{blue}\bullet},{\color{blue}\bullet}} = k^{d}_{{\color{red}\bullet},{\color{red}\bullet}} = 2 \\ k^{d}_{{\color{blue}\bullet},{\color{red}\bullet}} = 4 \\ P(q_0 \mid t = 0)) = 1 \end{cases}$$

Lumpability

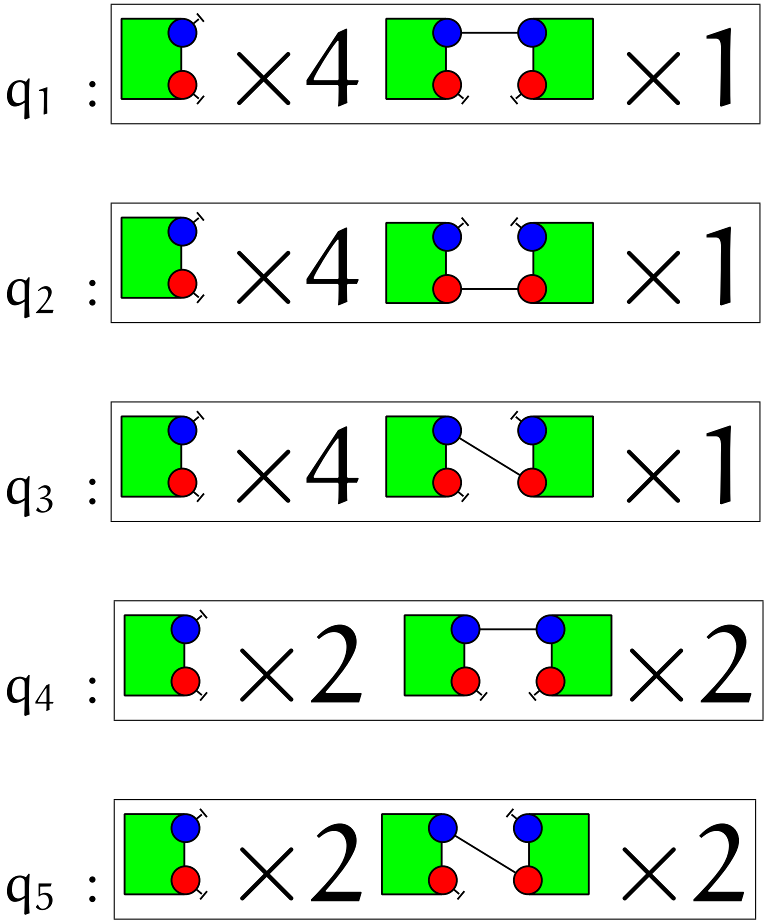


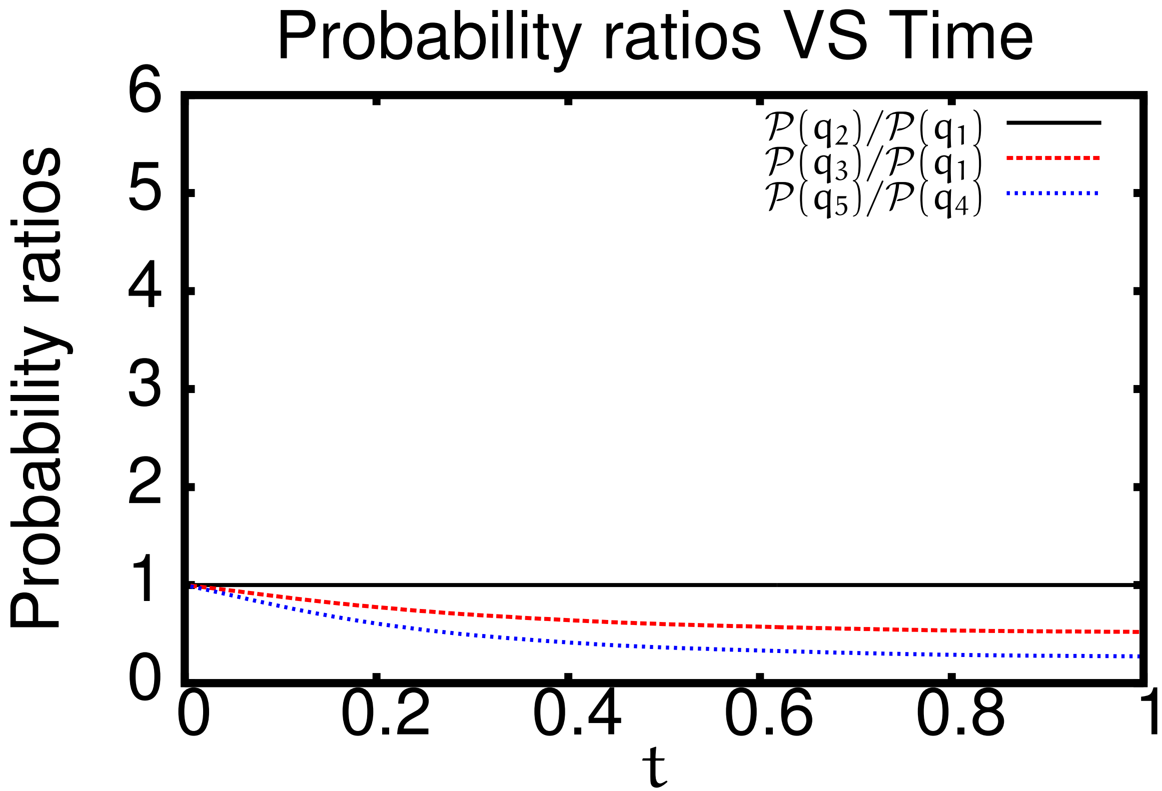
In general, when the following system:

$$\begin{cases} 2k_{\color{blue}\bullet,\color{blue}\bullet} = 2k_{\color{red}\bullet,\color{red}\bullet} = k_{\color{blue}\bullet,\color{red}\bullet} \\ k^d_{\color{blue}\bullet,\color{blue}\bullet} = k^d_{\color{blue}\bullet,\color{red}\bullet} = k^d_{\color{red}\bullet,\color{red}\bullet} \end{cases}$$

is not satisfied, we cannot lump the system.

Probability ratios (wrong coefficients)





with: $$\begin{cases} k_{\textcolor{blue}{\bullet},\textcolor{blue}{\bullet}} = k_{\textcolor{red}{\bullet},\textcolor{red}{\bullet}} = k_{\textcolor{blue}{\bullet},\textcolor{red}{\bullet}} = 1 \\ k^d_{\textcolor{blue}{\bullet},\textcolor{blue}{\bullet}} = k^d_{\textcolor{red}{\bullet},\textcolor{red}{\bullet}} = 2 \\ k^d_{\textcolor{blue}{\bullet},\textcolor{red}{\bullet}} = 4 \\ P(q_0 \mid t = 0)) = 1 \end{cases}$$

In this talk

An algebraic notion of symmetries over site graphs:

- compatible with the SPO (Single Push-Out) semantics of Kappa;
- with a notion of subgroups of symmetries;
- with a notion of symmetric models.

Some conditions so that symmetries over a model induce

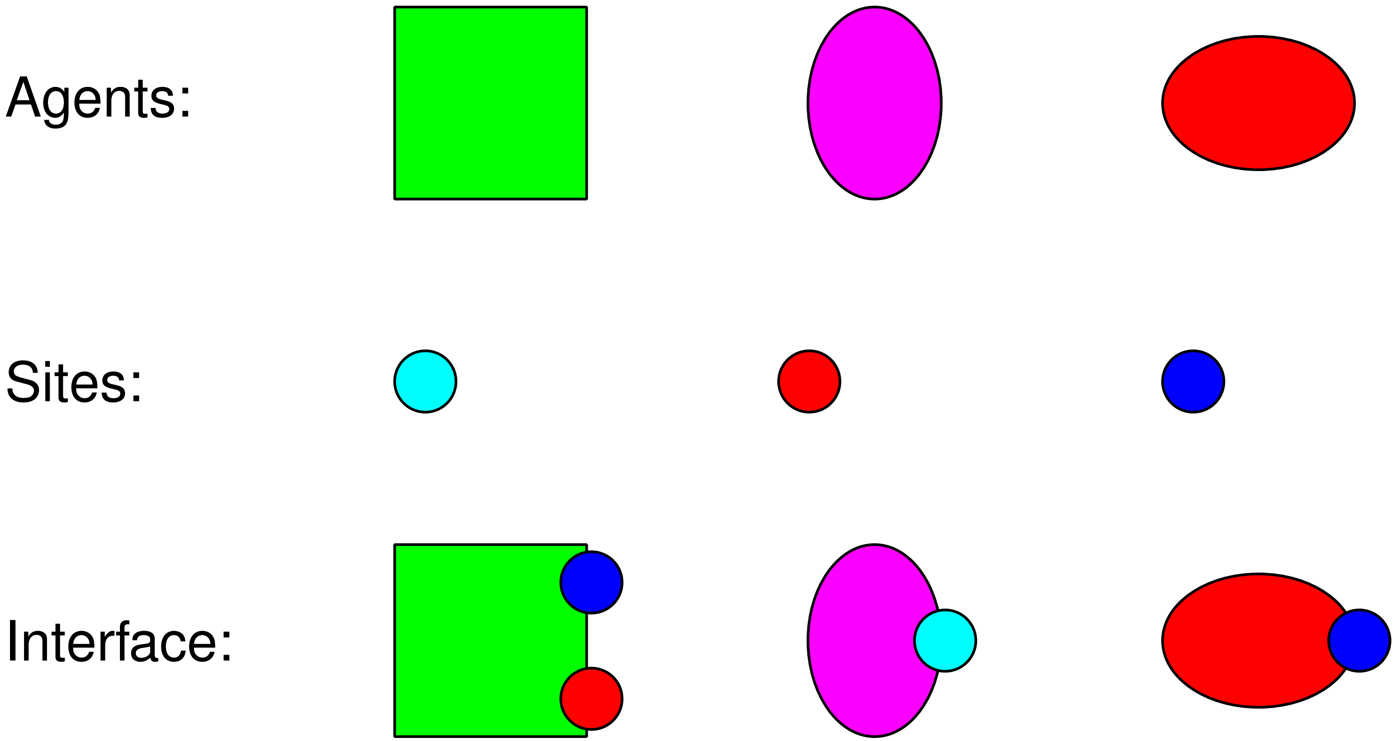
- a forward bisimulation;
- a backward bisimulation.

In this talk, we consider only a side-effect free fragment of Kappa.
The full language is handled with in, the paper.

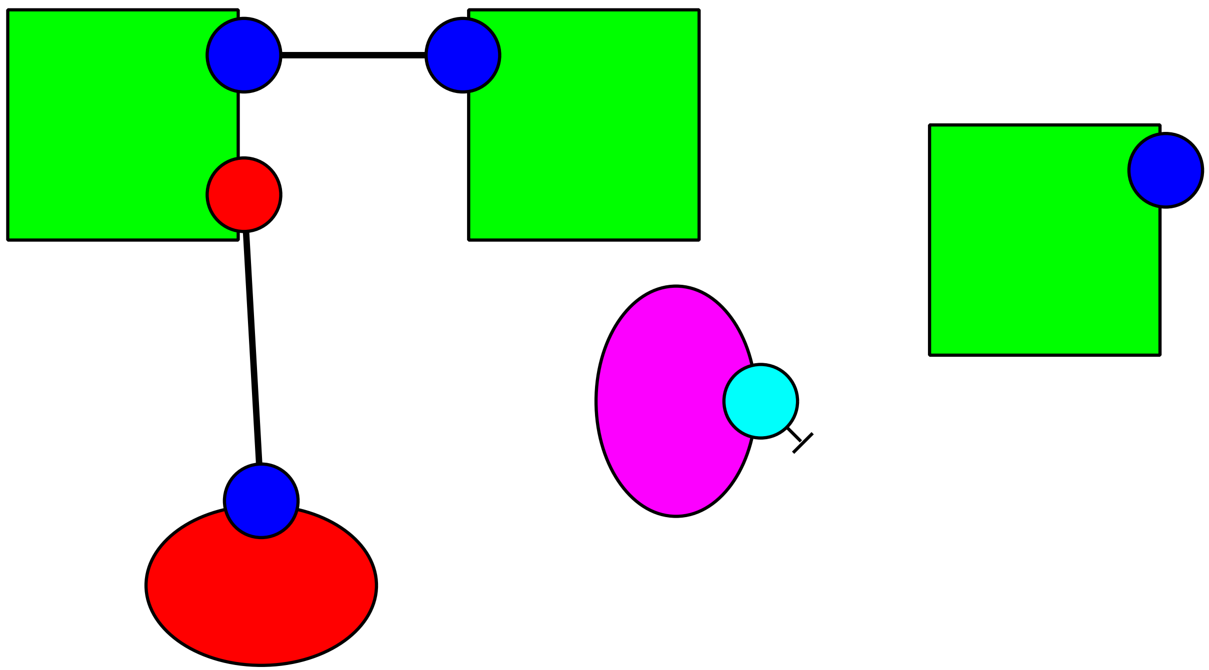
Overview

1. Context and motivations
2. Case study
3. Kappa semantics
4. Symmetries in site-graphs
5. Symmetric models
6. Conclusion

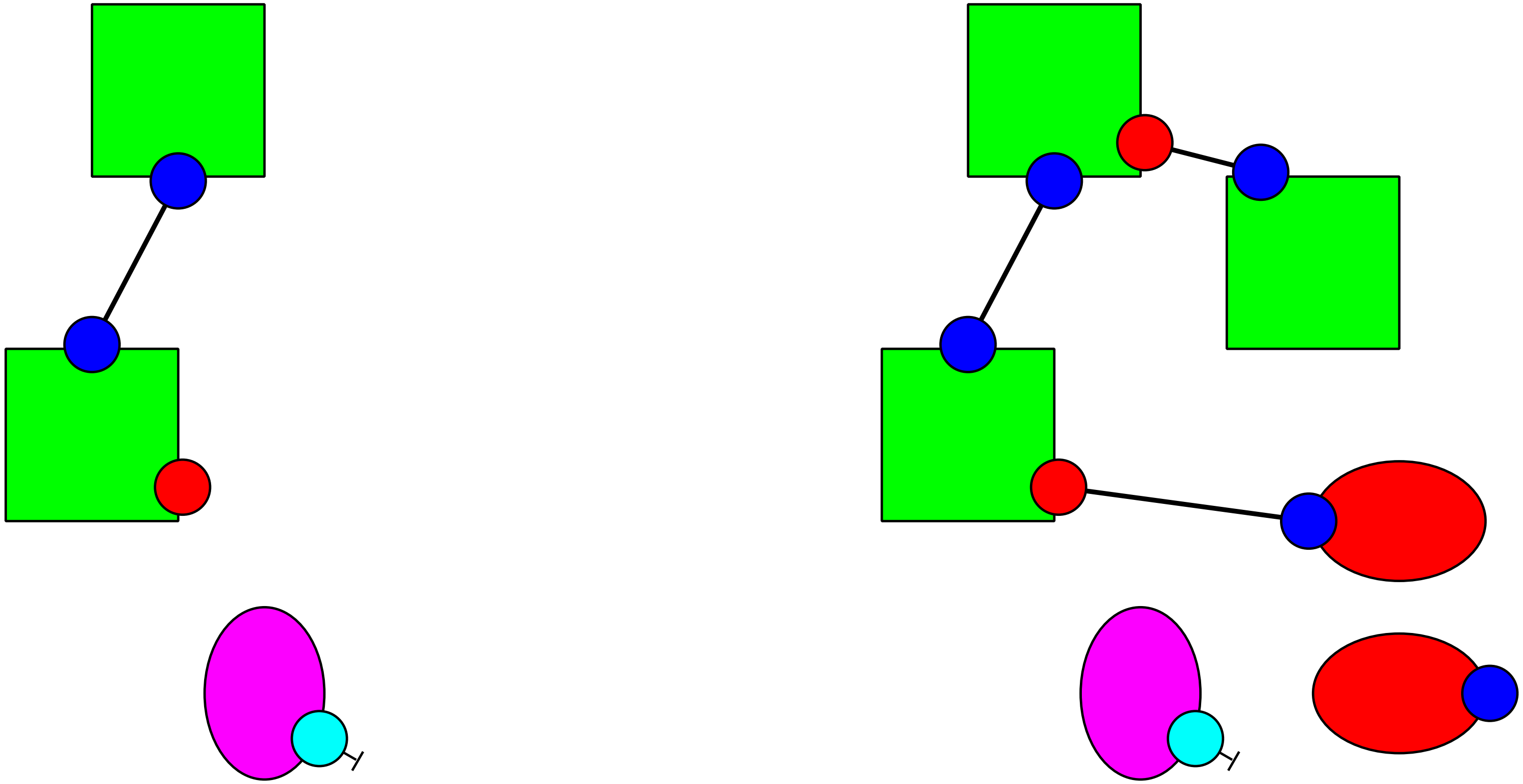
Signature



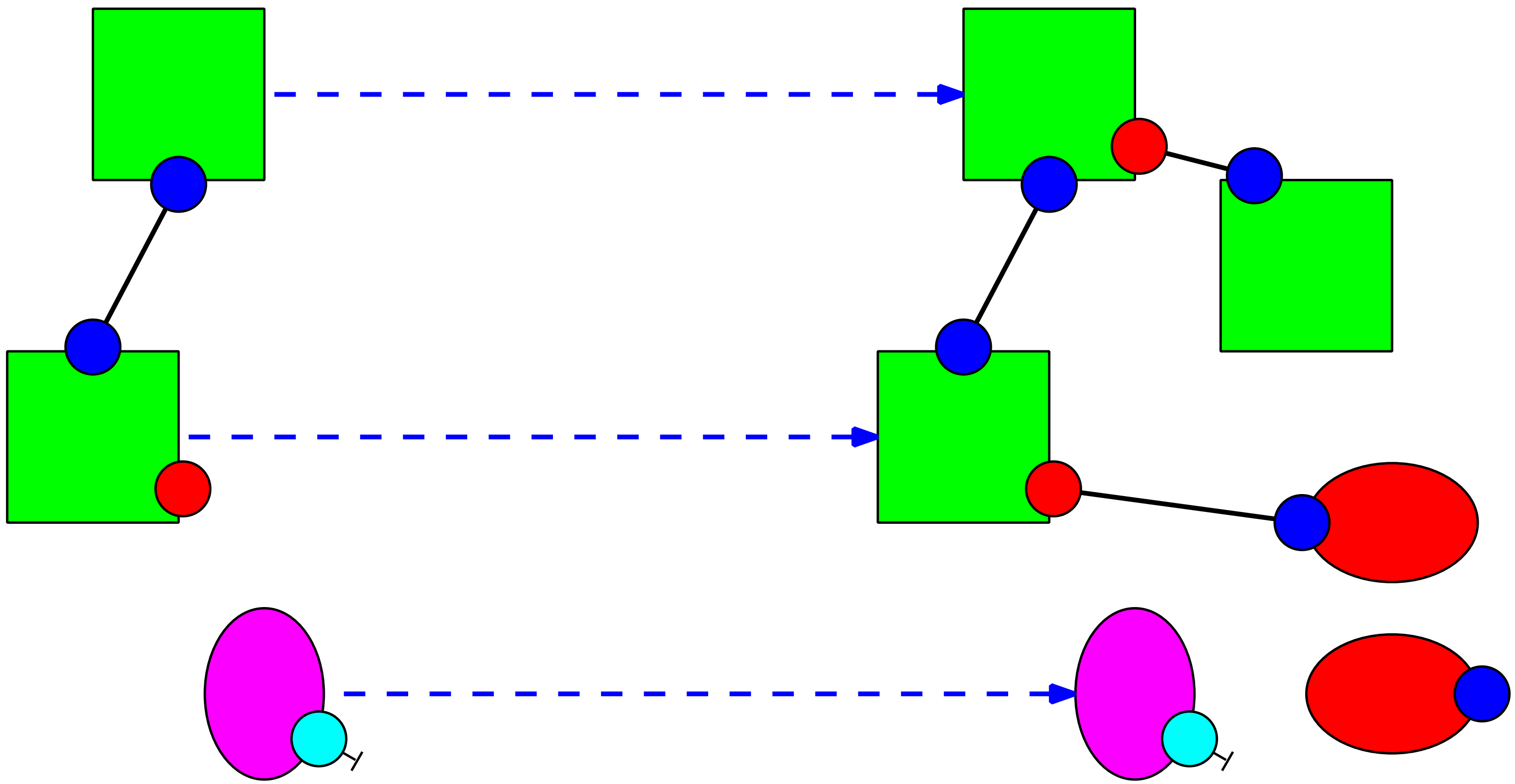
Site graphs



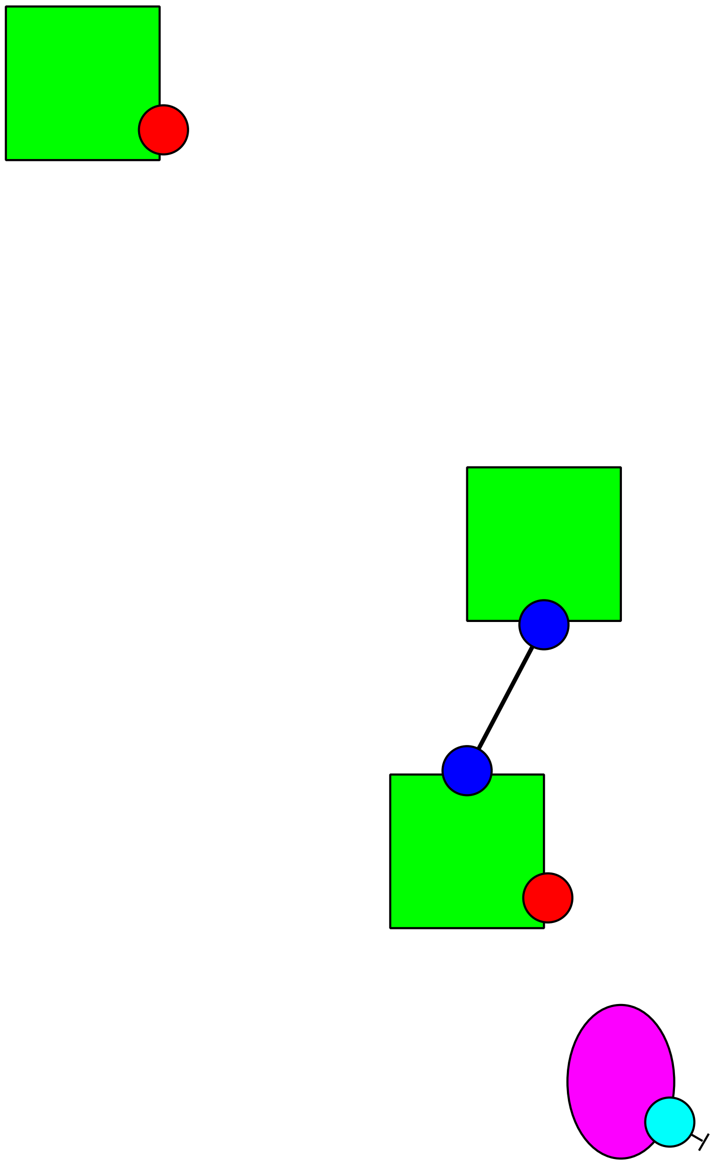
Embeddings

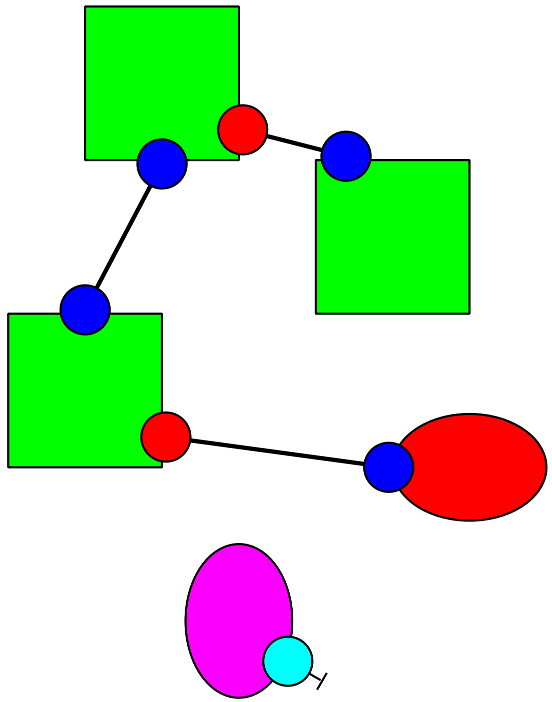


Embeddings

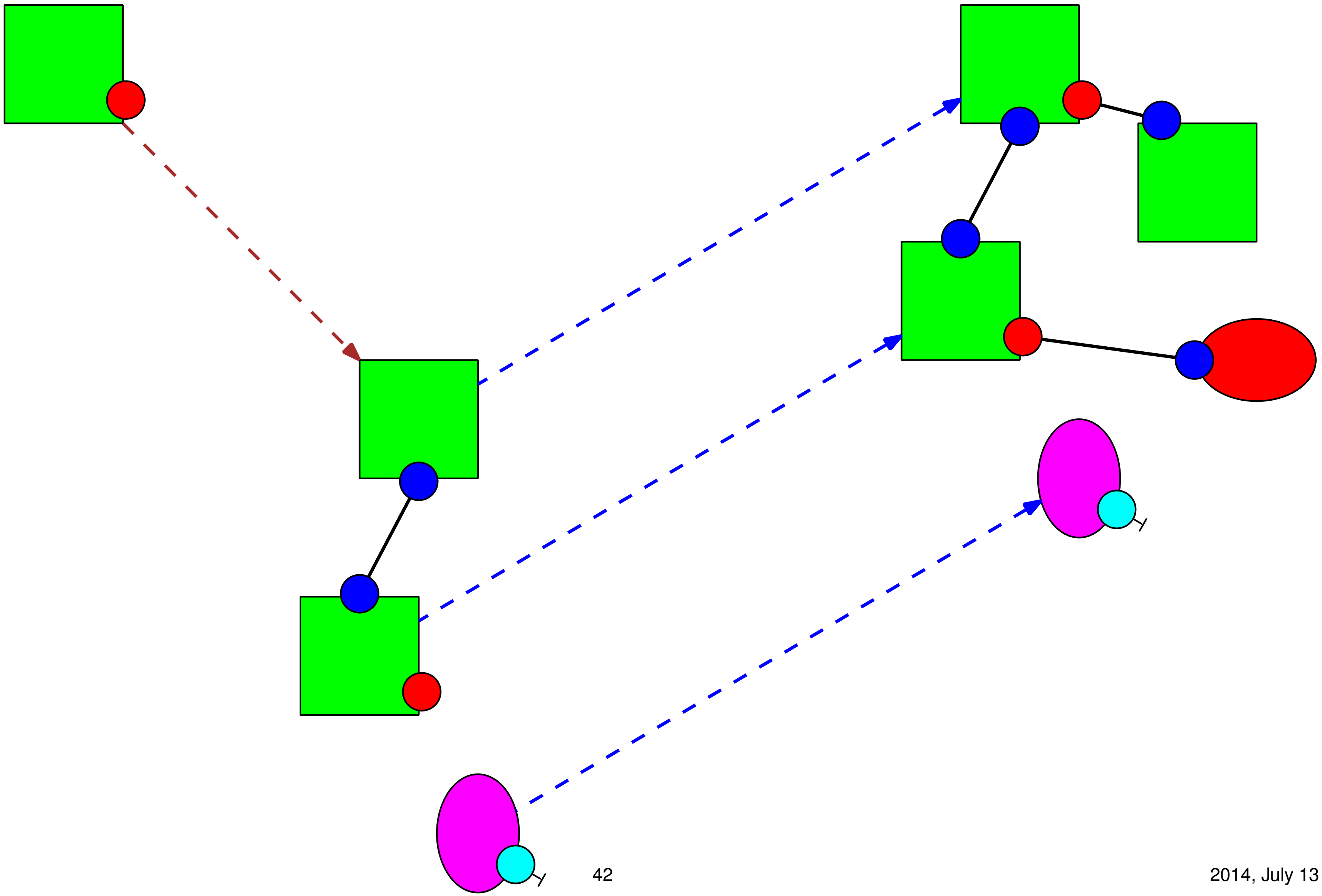


Composition of embeddings

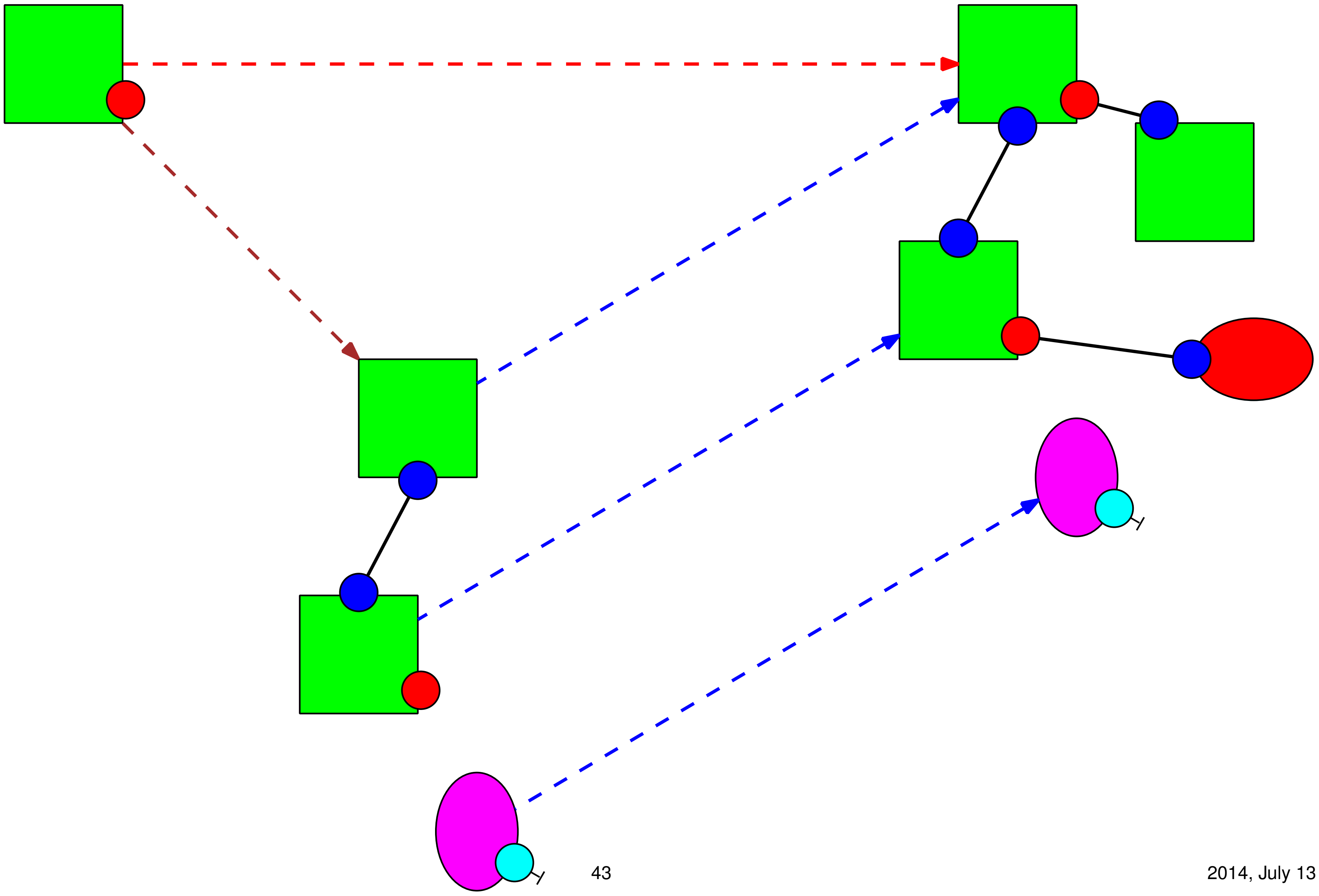




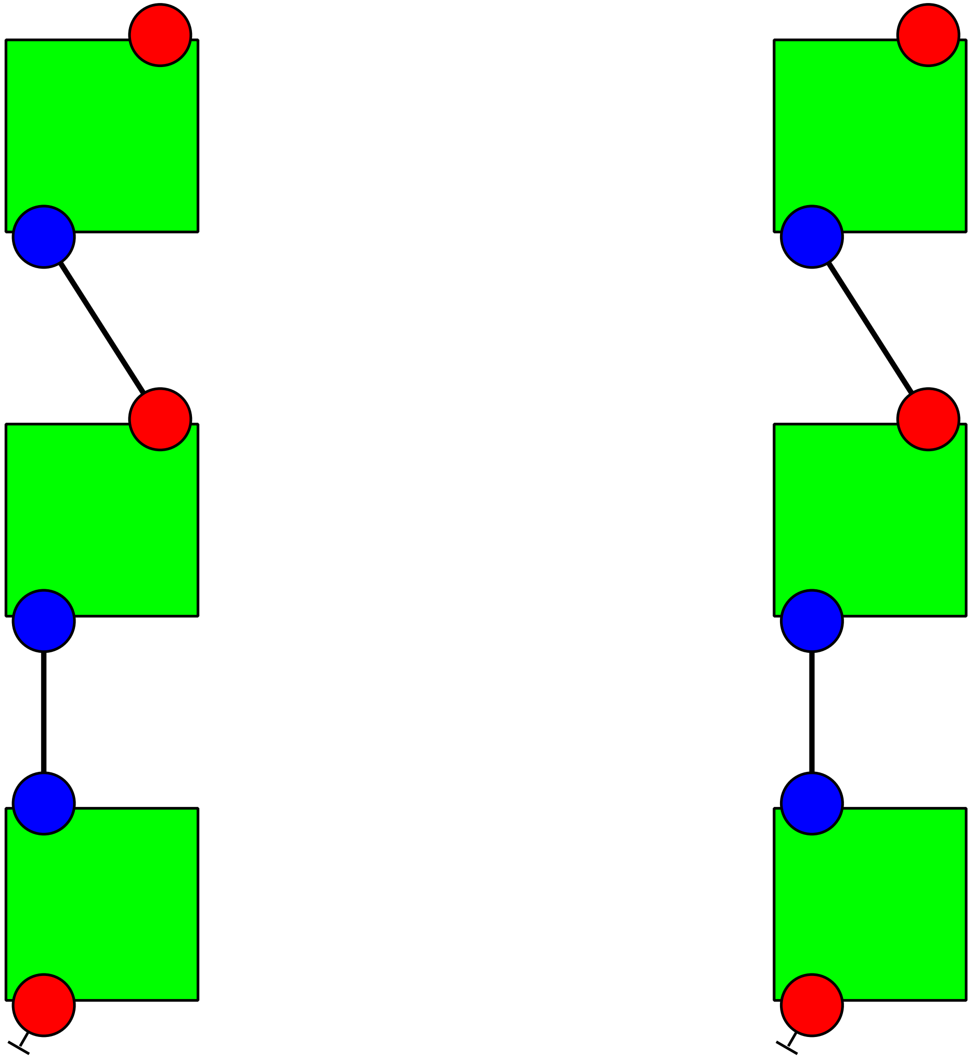
Composition of embeddings



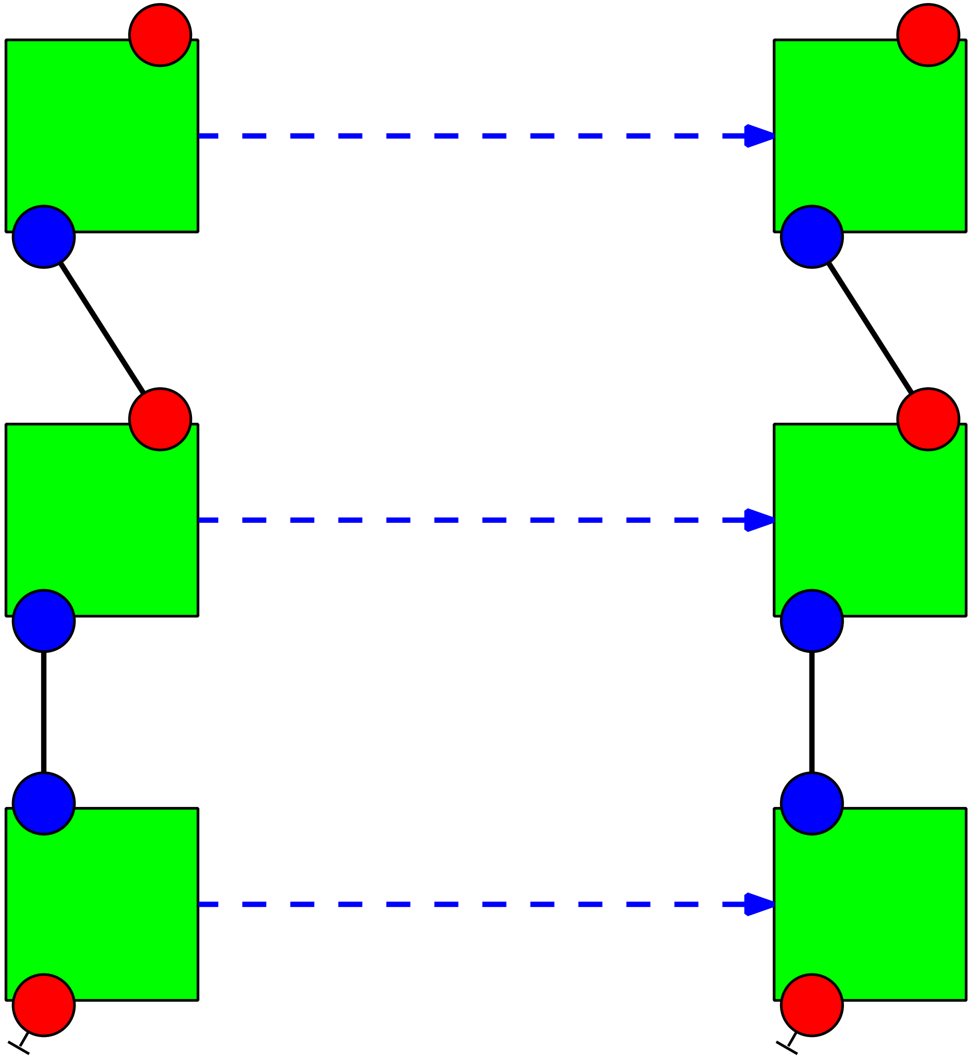
Composition of embeddings



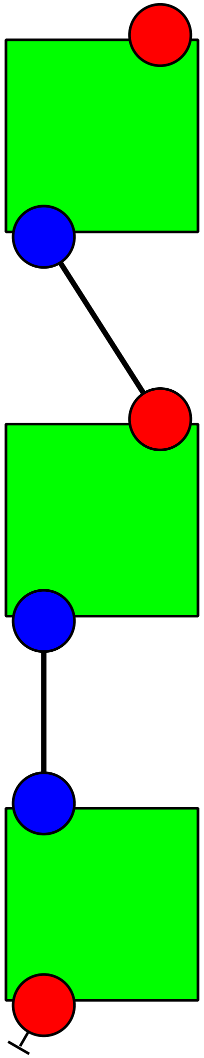
Identity embeddings

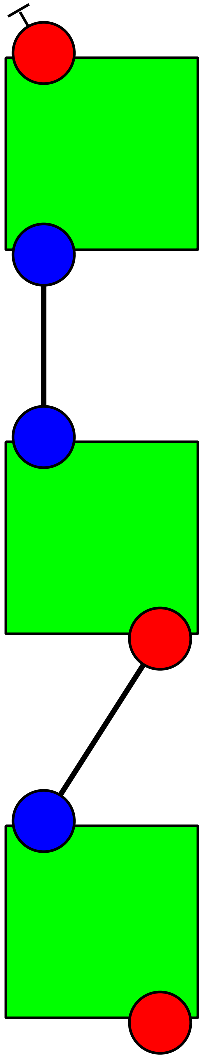


Identity embeddings

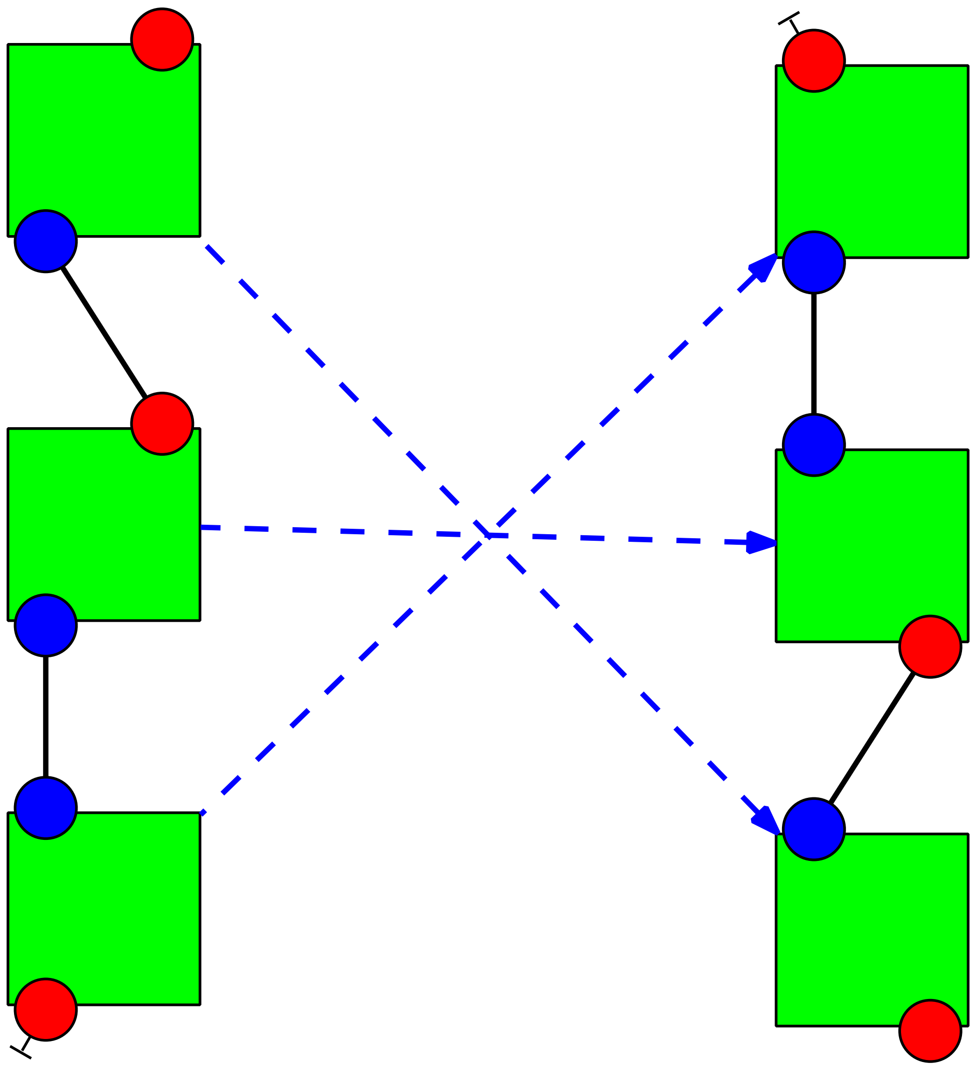


Isomorphisms

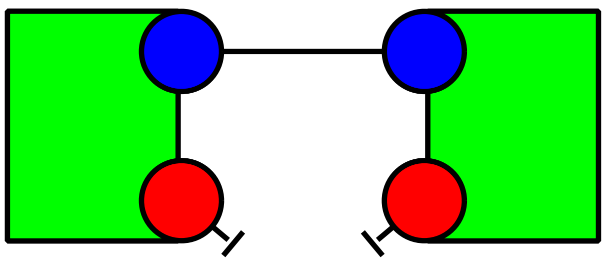




Isomorphisms



Fully specified site graphs



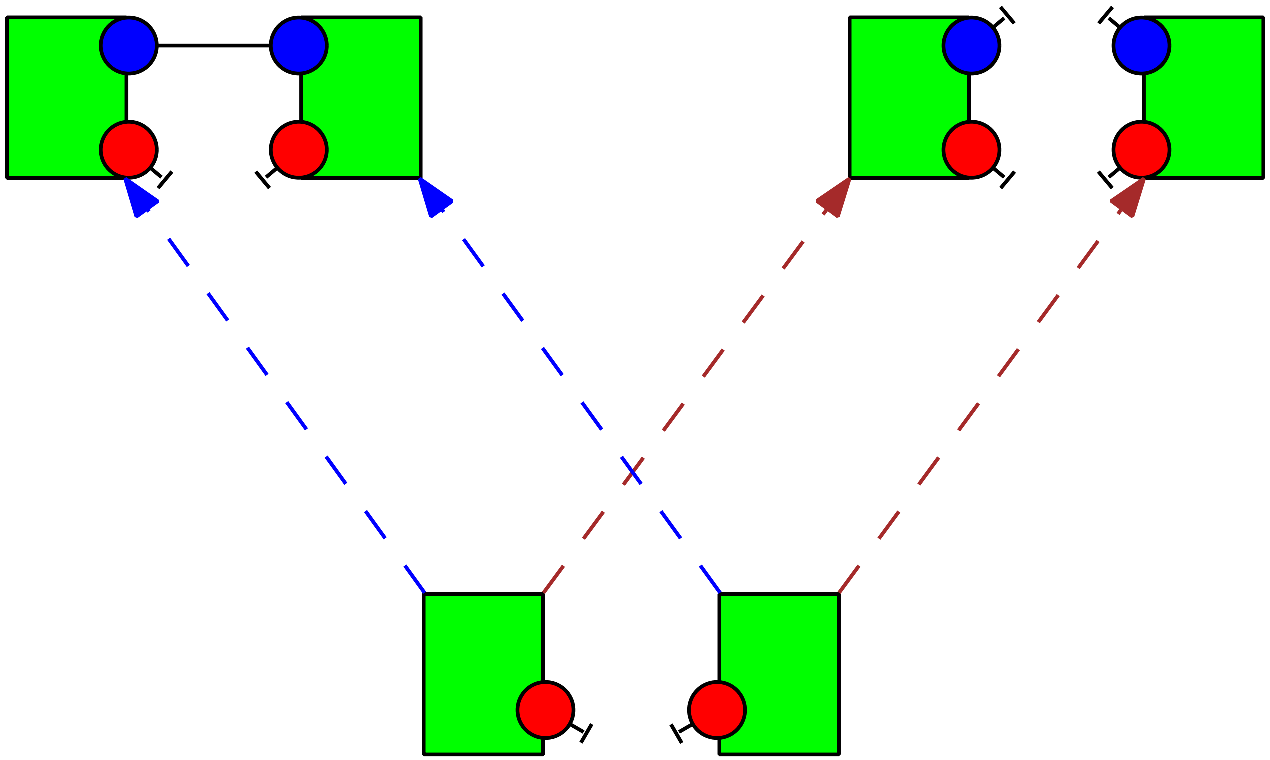
Isomorphic embeddings

When the following diagram:

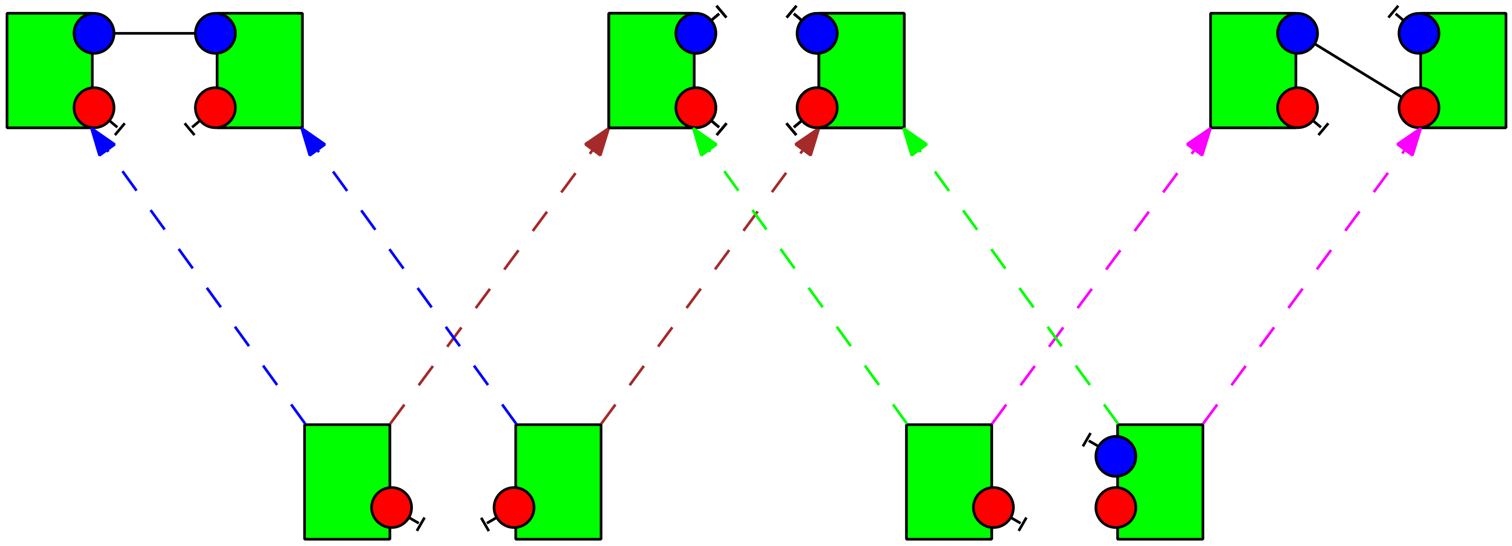
$$\begin{tikzcd} {} \arrow[rr, hook, "f"] \arrow[dr, hook, "g"'] & & {} \\ & {} \arrow[ur, "\approx"'] & \end{tikzcd}$$

commutes, we say that the embeddings f and g are isomorphic, and we write $f \approx g$.

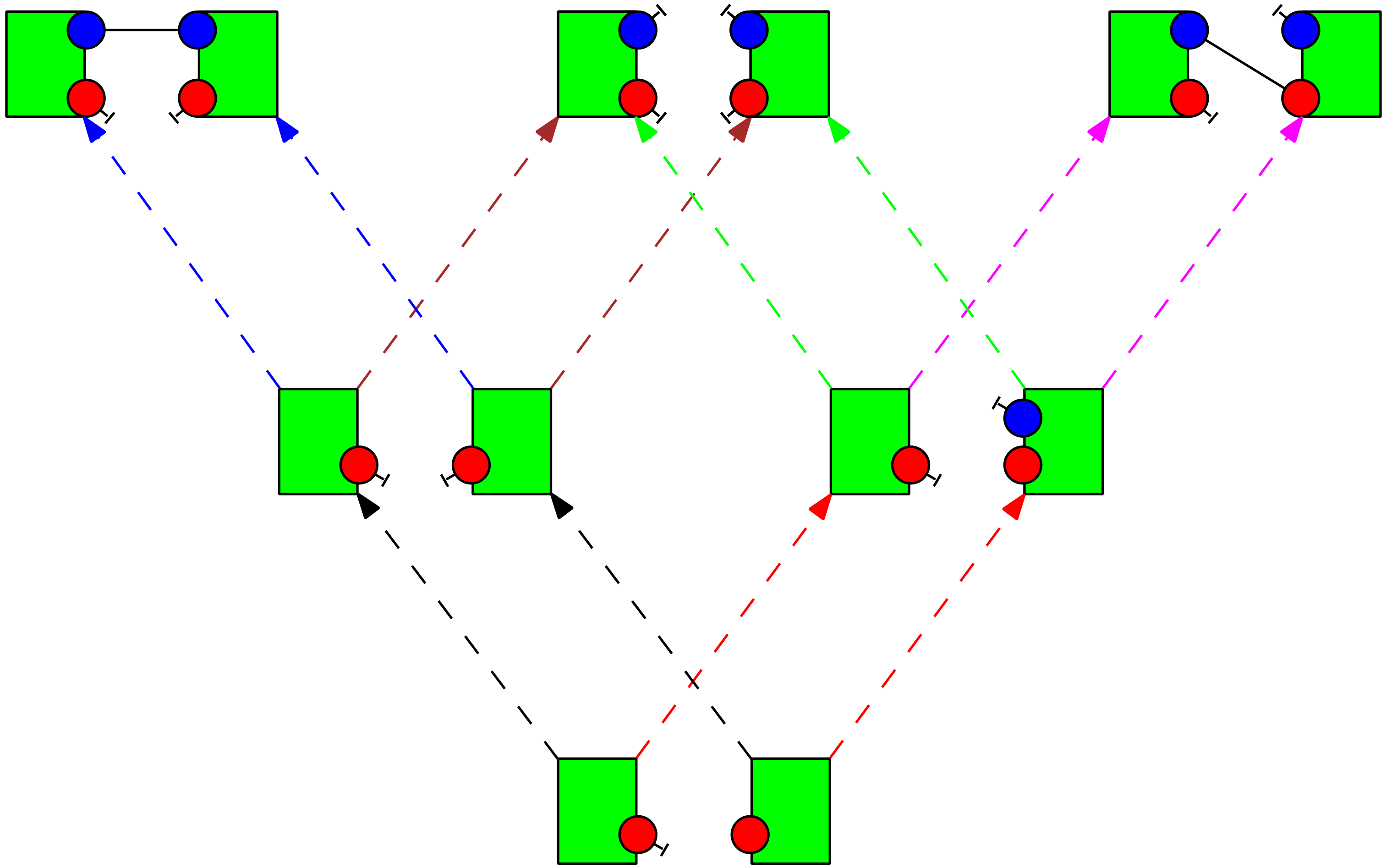
Partial embeddings



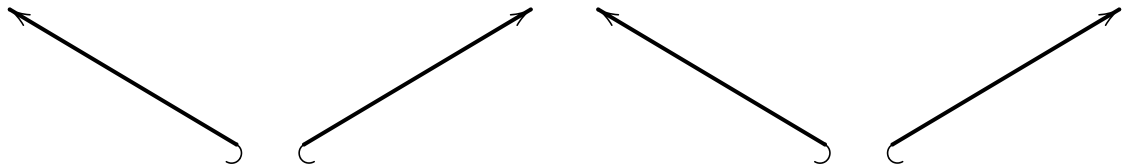
Composition of partial embeddings



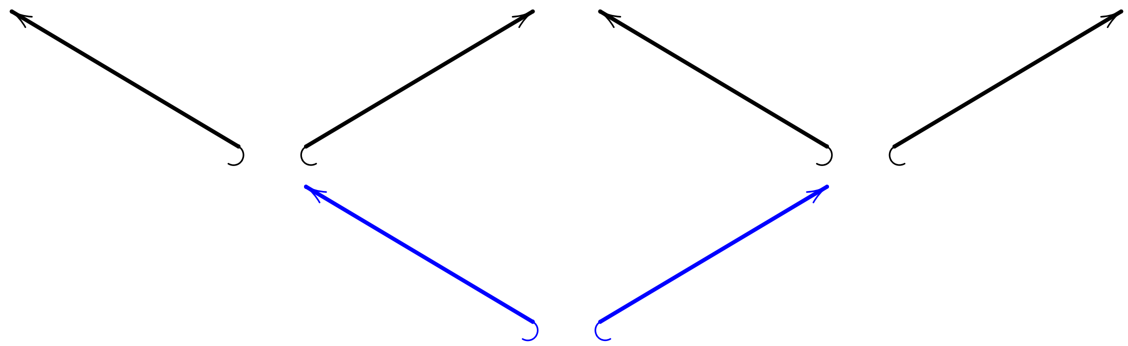
Composition of partial embeddings



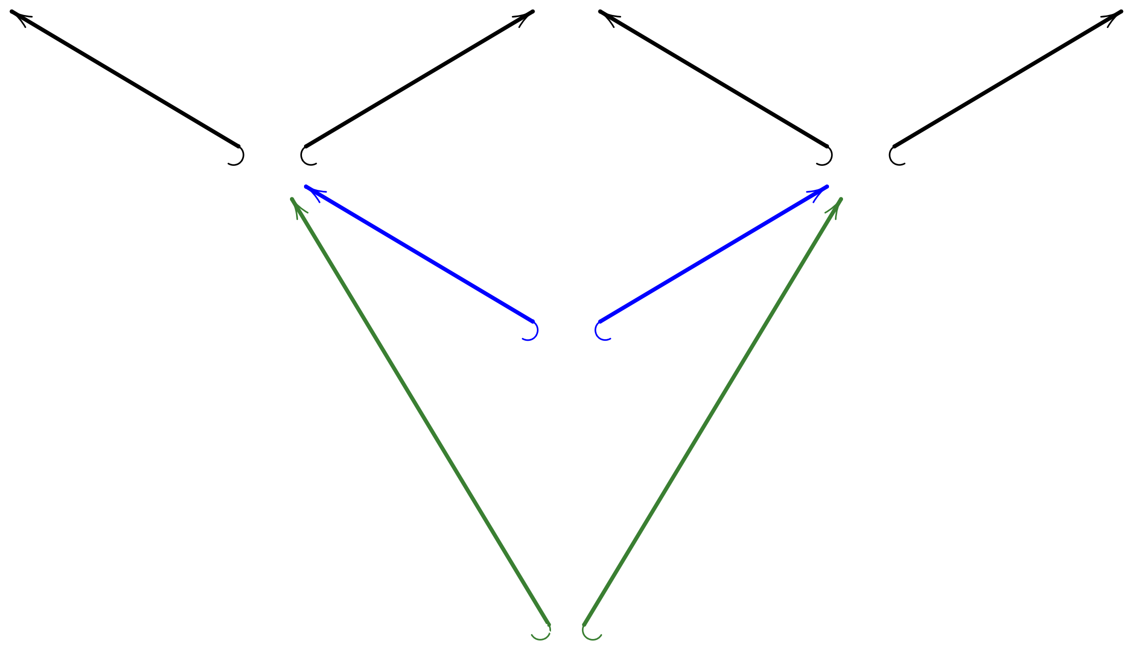
Composition of partial embeddings



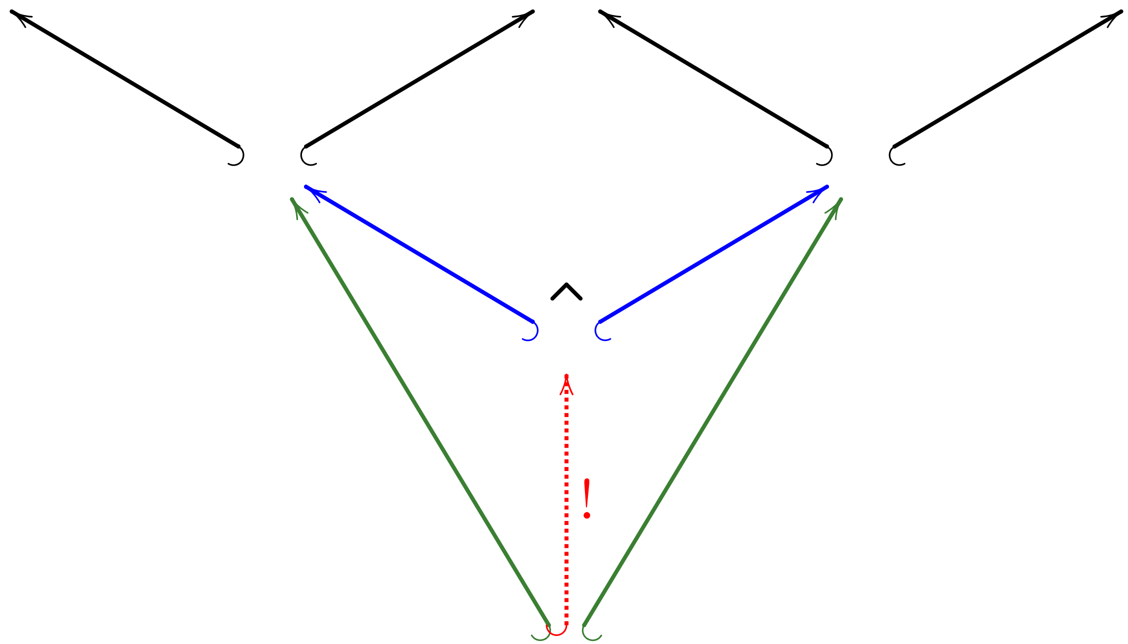
Composition of partial embeddings



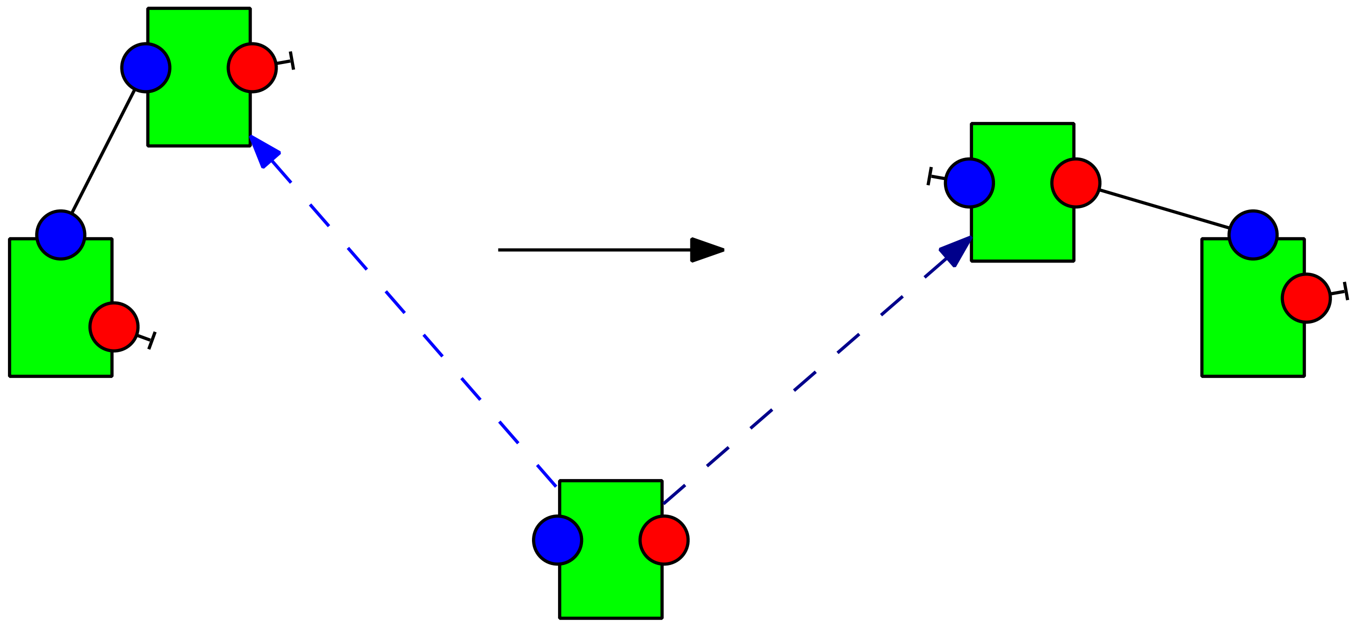
Composition of partial embeddings



Composition of partial embeddings



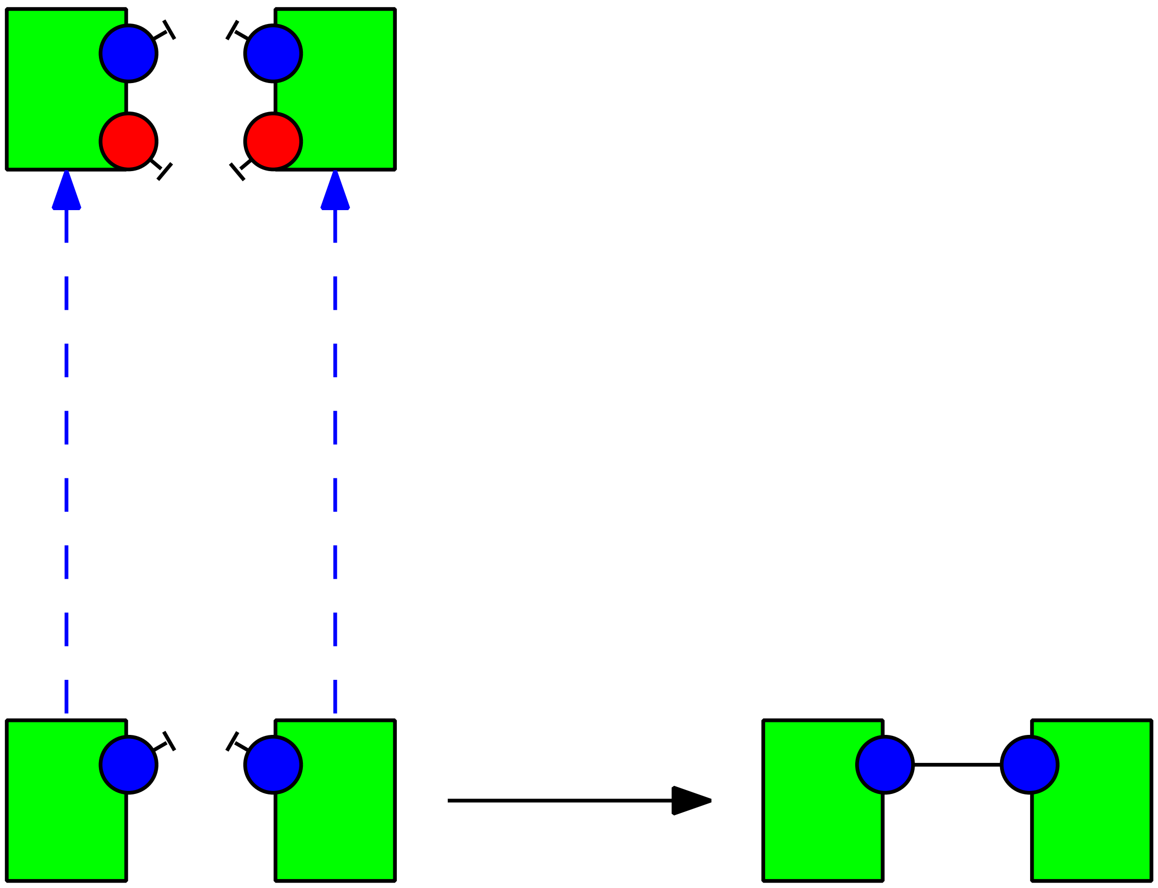
Rules



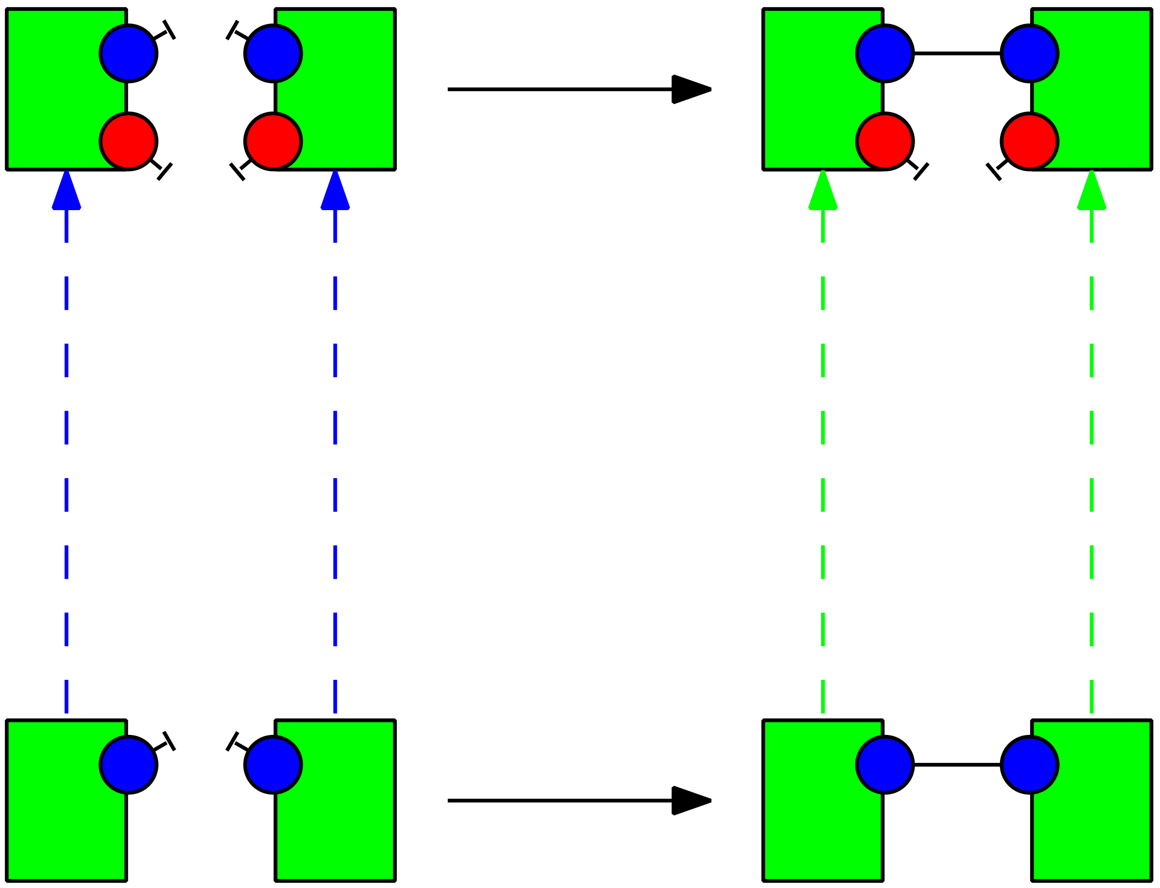
A rule is a partial embedding such that:

- the domain (D) is maximal;
- some constraints that we omit here are satisfied.

Rule application

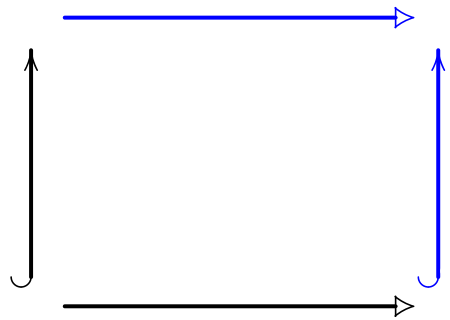


Rule applications

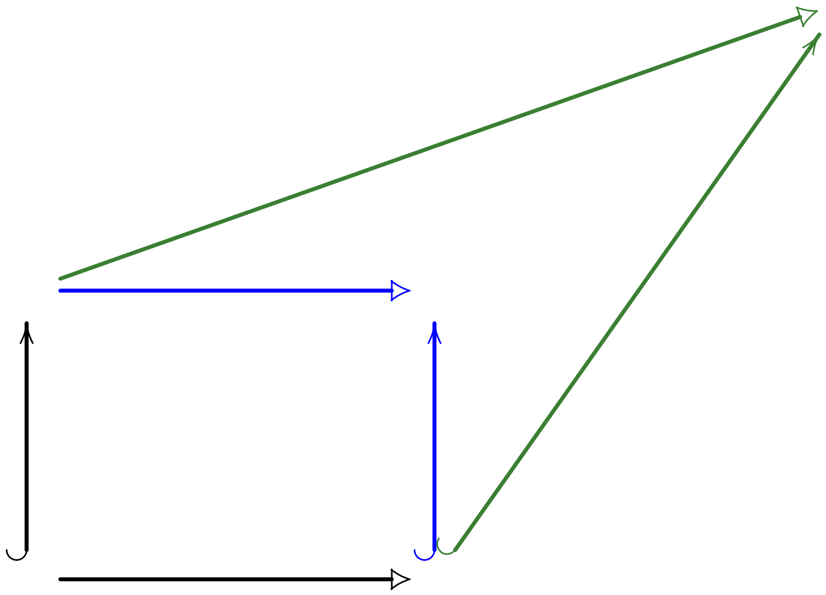


Refinement

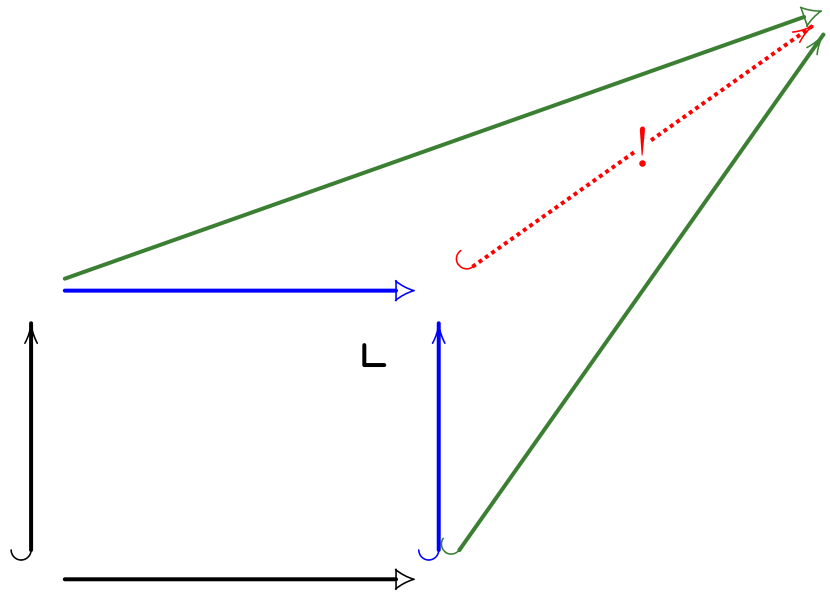
Refinement



Refinement



Refinement



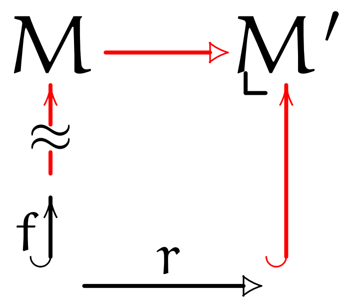
Semantics

1. A model is a map k from rules to non negative real numbers;
2. $\mathcal{Q} \stackrel{\Delta}{=} \{[G]_{\approx} \mid G$ fully specified site graph$\}$;
3. $\mathcal{L} \stackrel{\Delta}{=} \left\{ (r, [f]_{\approx}) \;\middle|\; \begin{array}{l} r \text{ a rule }, f \text{ an embedding from } \textit{lhs}(r) \\ \text{to a fully specified site graph} \end{array} \right\}$;
4. $[M]_{\approx} \xrightarrow{(r,[\phi]_{\approx})} [M']_{\approx}$ if and only if:

$$\begin{tikzcd} M & M' \\ {} \arrow[u, hook, "f"] \arrow[r, "r"] & {} \end{tikzcd}$$

Semantics

1. A model is a map k from rules to non negative real numbers;
2. $\mathcal{Q} \stackrel{\Delta}{=} \{[G]_{\approx} \mid G \text{ fully specified site graph}\}$;
3. $\mathcal{L} \stackrel{\Delta}{=} \left\{ (r, [f]_{\approx}) \;\middle|\; \begin{array}{l} r \text{ a rule }, f \text{ an embedding from } \textit{lhs}(r) \\ \text{to a fully specified site graph} \end{array} \right\}$;
4. $[M]_{\approx} \xrightarrow{(r,[f]_{\approx})} [M']_{\approx}$ if and only if:



The rate of such a transition is defined as:

$$\frac{\gamma(r)\textit{card}(\{\phi f \mid \phi \in \textit{Aut}(\textit{im}(f))\})}{\textit{card}(\textit{Aut}(\textit{lhs}(r)))}.$$

Applying transformations over push-outs

We would like to make pairs of transformations act over push-outs,

$$\begin{tikzcd}
q \arrow[r] & q' \\
L \arrow[u, hook] \arrow[r, "r"] & R \arrow[u, hook]
\end{tikzcd}
\qquad\qquad
\begin{tikzcd}
\sigma_q.q \arrow[rr] & & \sigma_{q'}.q' \\
& \text{agree} & \\
\sigma'_q.L \arrow[uu, hook] \arrow[rr, "(\sigma'_q,\sigma'_{q'}).r"] & & \sigma'_{q'}.R \arrow[uu, hook]
\end{tikzcd}$$

whenever they act the same way on preserved agents.

Overview

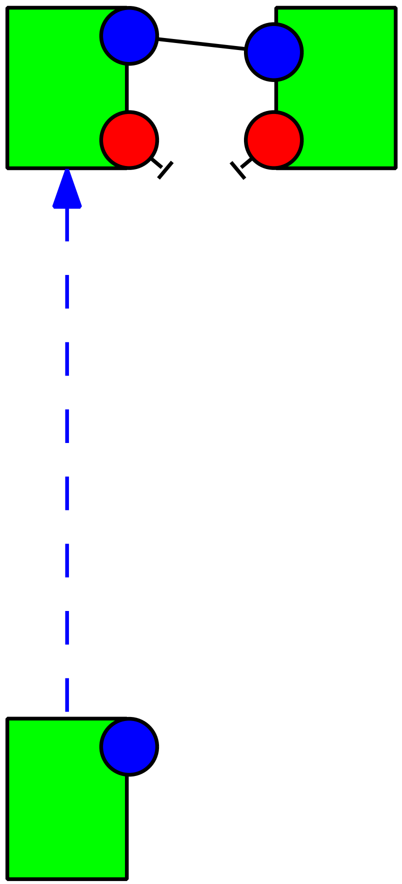
1. Context and motivations
2. Case study
3. Kappa semantics
4. Symmetries in site-graphs
   (a) Groups of transformations
   (b) Action of the transformations
5. Symmetric models
6. Conclusion

Transformations over site graphs

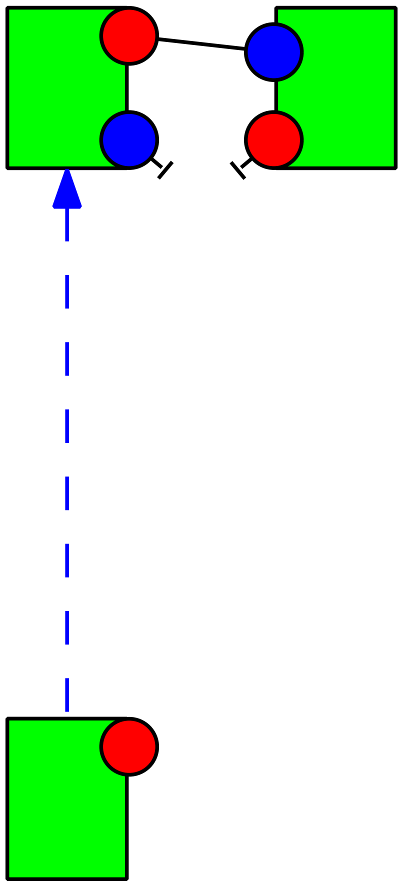
- For any site graph G, we introduce a finite group of transformations $\mathbb{G}_{\mathrm{G}}$.

- For any site graph G and any transformation $\sigma \in \mathbb{G}_{\mathrm{G}}$, we introduce the site graph $\sigma.\mathrm{G}$ and we call it the image of G by σ.
- We assume that $\mathbb{G}_{\mathrm{G}}$ and $\mathbb{G}_{(\sigma.\mathrm{G})}$ are the same group.

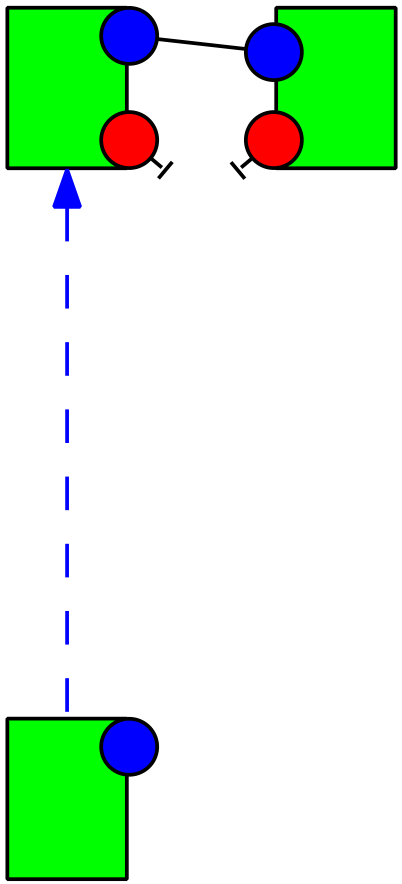
Restricting a transformation to the domain of an embedding



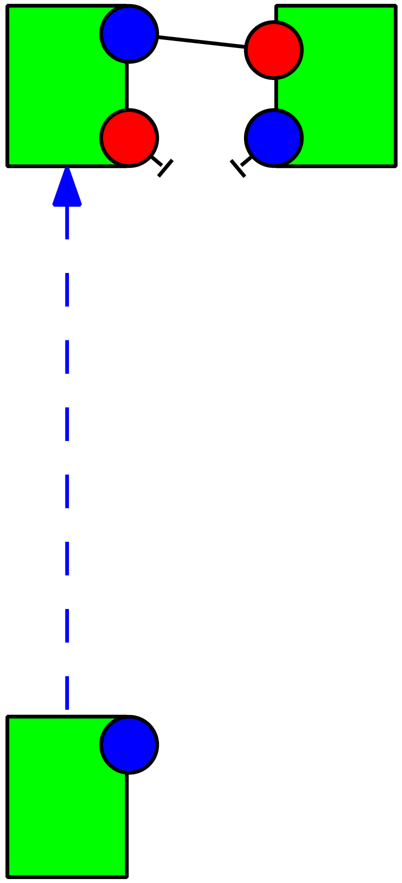
Restricting a transformation to the domain of an embedding



Restricting a transformation to the domain of an embedding



Restricting a transformation to the domain of an embedding



Restriction of symmetry to the domain of an embedding

$$\begin{tikzcd} G \arrow[r, hook, "f"] & H \arrow[d, squiggly, "\sigma"] \\ & \sigma.H \end{tikzcd}$$

Restriction of symmetry to the domain of an embedding

$$\begin{tikzcd}
G \arrow[r, hook, "f"] \arrow[d, squiggly, "f.\sigma"'] & H \arrow[d, squiggly, "\sigma"] \\
(f.\sigma).G \arrow[r, hook, "\sigma.f"] & \sigma.H
\end{tikzcd}$$

Identity function

$$\begin{tikzcd} E \arrow[r, hook, "i_E"] & E \arrow[d, squiggly, "\sigma"] \\ & \end{tikzcd}$$

Identity function

$$
\begin{tikzcd}
E \arrow[r, hook, "i_E"] & E \arrow[d, squiggly, "\sigma"] \\
 & \sigma.E
\end{tikzcd}
$$

Identity function

$$
\begin{tikzcd}
E \arrow[r, hook, "i_E"] \arrow[d, squiggly, "i_E.\sigma"'] & E \arrow[d, squiggly, "\sigma"] \\
(i_E.\sigma).E \arrow[r, hook, "\sigma.i_E"'] & \sigma.E
\end{tikzcd}
$$

Identity function

$$\begin{tikzcd}
E \arrow[r, hook, "i_E"] \arrow[d, squiggly, shift right, "i_E.\sigma"'] \arrow[d, squiggly, shift left, "\sigma"] & E \arrow[d, squiggly, "\sigma"] \\
(i_E.\sigma).E \arrow[r, hook, shift left, "i_{(\sigma.E)}"] \arrow[r, hook, shift right, "\sigma.i_E"'] & \sigma.E
\end{tikzcd}$$

Identity function

$$
\begin{tikzcd}
E \arrow[r, hook, "i_E"] \arrow[d, squiggly, shift right, "i_E.\sigma"'] \arrow[d, squiggly, shift left, "\sigma"] & E \arrow[d, squiggly, "\sigma"] \\
(i_E.\sigma).E \arrow[r, hook, shift left, "i_{(\sigma.E)}"] \arrow[r, hook, shift right, "\sigma.i_E"'] & \sigma.E
\end{tikzcd}
$$

We assume that:

- $i_E.\sigma = \sigma$
- $\sigma.i_E = i_{(\sigma.E)}$

Identity symmetry

$$\begin{tikzcd}
E \arrow[r, hook, "f"] & F \arrow[d, Rightsquigarrow, "\varepsilon_F"] \\
& {}
\end{tikzcd}$$

Identity symmetry

$$\begin{tikzcd} E \arrow[r, hook, "f"] & F \arrow[d, squiggly, "\varepsilon_F"] \\ & \varepsilon_F.F \end{tikzcd}$$

Identity symmetry

$$\begin{tikzcd}[column sep=huge, row sep=huge]
E \arrow[r, hook, "f"] \arrow[d, squiggly, "f.\varepsilon_F"'] & F \arrow[d, squiggly, "\varepsilon_F"] \\
(f.\varepsilon_F).E \arrow[r, hook, "\varepsilon_F.f"'] & \varepsilon_F.F
\end{tikzcd}$$

Identity symmetry

$$\begin{tikzcd}[column sep=huge, row sep=huge]
E \arrow[r, hook, "f"] \arrow[d, squiggly, shift right, "f.\varepsilon_F"'] \arrow[d, squiggly, shift left, "\varepsilon_E"] & F \arrow[d, squiggly, "\varepsilon_F"] \\
E = (f.\varepsilon_F).E \arrow[r, hook, shift left, "f"] \arrow[r, hook, shift right, "\varepsilon_F.f"'] & \varepsilon_F.F = F
\end{tikzcd}$$

Identity symmetry

$$\begin{tikzcd}
E \arrow[r, hook, "f"] \arrow[d, squiggly, shift right, "f.\varepsilon_F"'] \arrow[d, squiggly, shift left, "\varepsilon_E"] & F \arrow[d, squiggly, "\varepsilon_F"] \\
E = (f.\varepsilon_F).E \arrow[r, hook, shift left, "f"] \arrow[r, hook, shift right, "\varepsilon_F.f"'] & \varepsilon_F.F = F
\end{tikzcd}$$

We assume that:

- $\varepsilon_F.F = F$
- $f.\varepsilon_F = \varepsilon_E$
- $\varepsilon_F.f = f$

Composition of embeddings

$$\begin{tikzcd} E \arrow[dr, hook, "f"'] & & G \arrow[d, rightsquigarrow, Rightarrow, "\sigma"] \\ & F \arrow[ur, hook, "g"'] & \end{tikzcd}$$

Composition of embeddings

$$\begin{tikzcd}
E \arrow[rr, hook, "gf"] \arrow[dr, hook, "f"'] & & G \arrow[d, Rightarrow, squiggly, "\sigma"] \\
& F \arrow[ur, hook, "g"'] & {}
\end{tikzcd}$$

Composition of embeddings

$$
\begin{tikzcd}
E \arrow[rr, hook, "gf"] \arrow[dr, hook, "f"'] \arrow[dd, rightsquigarrow, "(gf).\sigma"'] & & G \arrow[dd, rightsquigarrow, "\sigma"] \\
& F \arrow[ur, hook, "g"'] & \\
((gf).\sigma).E \arrow[rr, hook, "\sigma.(gf)"'] & & \sigma.G
\end{tikzcd}
$$

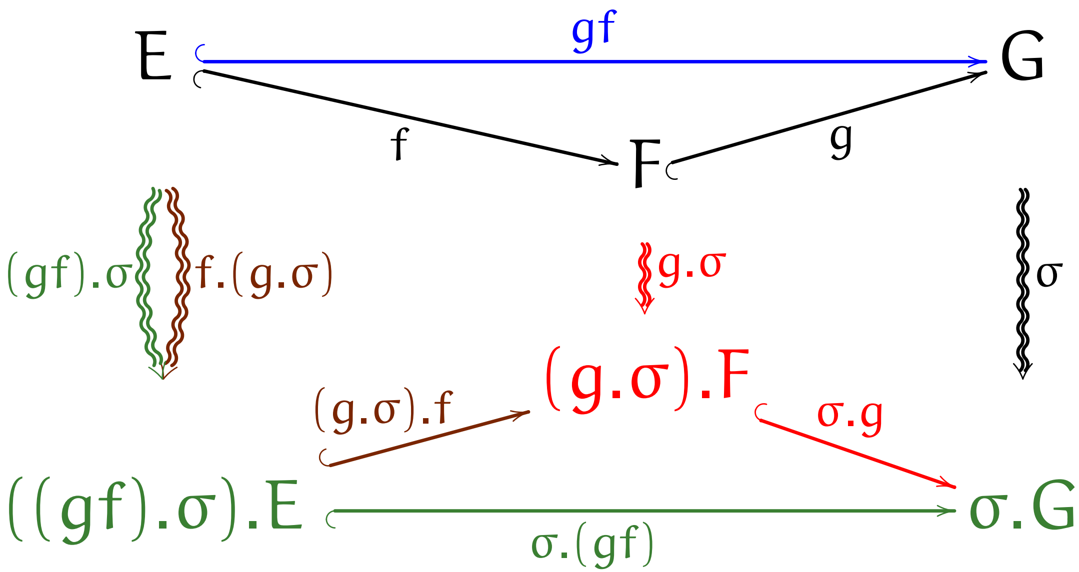
Composition of embeddings

$$
\begin{tikzcd}
E \arrow[rr, hook, "gf"] \arrow[dr, hook, "f"'] \arrow[ddd, rightsquigarrow, "(gf).\sigma"'] & & G \arrow[ddd, rightsquigarrow, "\sigma"] \\
& F \arrow[ur, hook, "g"'] \arrow[d, rightsquigarrow, "g.\sigma"] & \\
& (g.\sigma).F \arrow[dr, hook, "\sigma.g"] & \\
((gf).\sigma).E \arrow[rr, hook, "\sigma.(gf)"'] & & \sigma.G
\end{tikzcd}
$$

Composition of embeddings

$$\begin{tikzcd}
E \arrow[rr, hook, "gf"] \arrow[dr, hook, "f"'] \arrow[ddd, squiggly, shift right, "(gf).\sigma"'] \arrow[ddd, squiggly, shift left, "f.(g.\sigma)"] & & G \arrow[ddd, squiggly, "\sigma"] \\
& F \arrow[ur, hook, "g"'] \arrow[d, squiggly, "g.\sigma"] & \\
& (g.\sigma).F \arrow[dr, hook, "\sigma.g"] & \\
((gf).\sigma).E \arrow[ur, hook, "(g.\sigma).f"] \arrow[rr, hook, "\sigma.(gf)"'] & & \sigma.G
\end{tikzcd}$$

Composition of embeddings



We assume that:

- $(gf).\sigma = f.(g.\sigma)$
- $\sigma.(gf) = (\sigma.g)((g.\sigma).f)$

Product of transformations

$$\begin{tikzcd} E \arrow[r, hook, "f"] & F \arrow[d, rightsquigarrow, "\sigma' \circ \sigma"] \\ & \end{tikzcd}$$

Product of transformations

$$
\begin{tikzcd}
E \arrow[r, hook, "f"] \arrow[d, rightsquigarrow, "f.(\sigma'\circ\sigma)"'] & F \arrow[d, rightsquigarrow, "\sigma'\circ\sigma"] \\
(f.(\sigma' \circ \sigma)).E \arrow[r, hook, "(\sigma'\circ\sigma).f"] & (\sigma' \circ \sigma).F
\end{tikzcd}
$$

Product of transformations

$$\begin{tikzcd}
E \arrow[rrr, hook, "f"] \arrow[dr, squiggly, "f.\sigma"] \arrow[dd, squiggly, "f.(\sigma'\circ\sigma)"'] & & & F \arrow[dl, squiggly, "\sigma"'] \arrow[dd, squiggly, "\sigma'\circ\sigma"] \\
& (f.\sigma).E \arrow[r, hook, "\sigma.f"] & \sigma.F & \\
(f.(\sigma'\circ\sigma)).E \arrow[rrr, hook, "(\sigma'\circ\sigma).f"] & & & (\sigma'\circ\sigma).F
\end{tikzcd}$$

Product of transformations

$$
\begin{tikzcd}
E \arrow[rrr, hook, "f"] \arrow[dr, squiggly, "f.\sigma"] \arrow[dd, squiggly, "f.(\sigma'\circ\sigma)"'] & & & F \arrow[dl, squiggly, "\sigma"'] \arrow[dd, squiggly, "\sigma'\circ\sigma"] \\
& (f.\sigma).E \arrow[r, hook, "\sigma.f"] \arrow[dl, squiggly, "(\sigma.f).\sigma'"] & \sigma.F \arrow[dr, squiggly, "\sigma'"'] & \\
(f.(\sigma'\circ\sigma)).E \arrow[rrr, hook, "(\sigma'\circ\sigma).f"] & & & (\sigma'\circ\sigma).F
\end{tikzcd}
$$

Product of transformations

$$
\begin{tikzcd}
E \arrow[rrr, hook, "f"] \arrow[dr, squiggly, "f.\sigma"] \arrow[dd, squiggly, "f.(\sigma'\circ\sigma)"'] & & & F \arrow[dl, squiggly, "\sigma"'] \arrow[dd, squiggly, "\sigma'\circ\sigma"] \\
& (f.\sigma).E \arrow[r, hook, "\sigma.f"] \arrow[dl, squiggly, "(\sigma.f).\sigma'"] & \sigma.F \arrow[dr, squiggly, "\sigma'"] & \\
(f.(\sigma'\circ\sigma)).E \arrow[rrr, hook, "(\sigma'\circ\sigma).f"] & & & (\sigma'\circ\sigma).F
\end{tikzcd}
$$

We assume that:

- $(\sigma' \circ \sigma).F = \sigma'.(\sigma.F)$
- $f.(\sigma' \circ \sigma) = ((f.\sigma).\sigma') \circ (f.\sigma)$
- $(\sigma' \circ \sigma).f = \sigma'.(\sigma.f)$

Images of fully specified site graphs

We assume that for any site graph G and any transformation $\sigma \in \mathbb{G}_{\mathrm{G}}$ the two following assertions are equivalent:

1. G is fully specified;
2. $\sigma.\mathrm{G}$ is fully specified.

Images of partial embeddings

For any partial embedding $\phi \;:\; L \overset{f}{\hookleftarrow} D \overset{g}{\hookrightarrow} R$,
We assume that:

- if

$$\begin{cases} f.\sigma_L = g.\sigma_R \\ f.\sigma'_L = g.\sigma'_R \end{cases}$$

- then

$$f.(\sigma_L \circ \sigma'_L) = g.(\sigma_R \circ \sigma'_R),$$

for any $\sigma_L, \sigma'_L \in \mathbb{G}_L$, $\sigma_R, \sigma'_R \in \mathbb{G}_R$,

We consider:

$$\mathbb{G}_\phi \overset{\Delta}{=} \{(\sigma_L, \sigma_R) \in \mathbb{G}_L \times \mathbb{G}_R \mid f.\sigma_L = g.\sigma_R\}.$$

Images of rules

We assume that for any partial embedding $\phi : L \overset{f}{\hookleftarrow} D \overset{g}{\hookrightarrow} R$ and any (pair of) transformation(s) $(\sigma_L, \sigma_R) \in \mathbb{G}_\phi$ the two following assertions are equivalent:

1. ϕ is a rule;
2. $\sigma_L.L \overset{\sigma_L.f}{\hookleftarrow} (f.\sigma_L).D \overset{\sigma_R.g}{\hookrightarrow} \sigma_R.R$ is a rule.

Images of push-outs

Theorem 1 Let r be a rule, and $(\sigma_L, \sigma_R) \in \mathbb{G}_r$ be a pair of transformations. If the following diagram:

$$
\begin{tikzcd}
L' \arrow[r, "r"] & R' \arrow[dl, phantom, "\llcorner", very near start] \\
L \arrow[u, hook, "h_L"] \arrow[r, "r'"'] & R \arrow[u, hook', "h_R"']
\end{tikzcd}
$$

is a push-out, then the following diagram:

$$
\begin{tikzcd}
\sigma_L.L' \arrow[r, "(\sigma_L{,}\sigma_R).r"] & \sigma_R.R' \arrow[dl, phantom, "\llcorner", very near start] \\
(h_L.\sigma_L).L \arrow[u, hook, "\sigma_L.h_L"] \arrow[r, "(h_L.\sigma_L{,}h_R.\sigma_R).r'"'] & (h_R.\sigma_R).R \arrow[u, hook', "\sigma_R.h_R"']
\end{tikzcd}
$$

is a push-out as well.

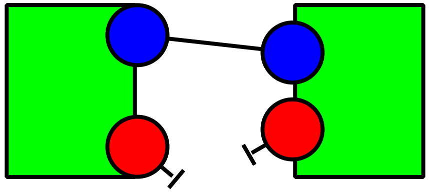
Subgroups of transformations

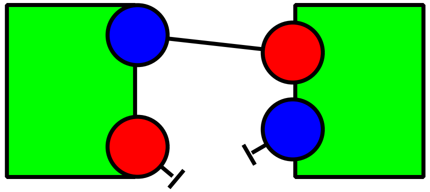
Theorem 2
If, for any embedding h between two site graphs G and H:

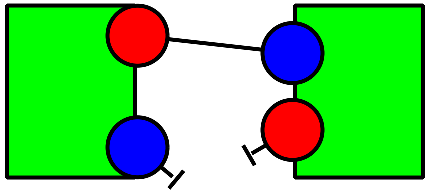
- we have a subset $\mathbb{G}'_G$ of $\mathbb{G}_G$;
- for any transformation $\sigma \in \mathbb{G}'_G$, $\mathbb{G}'_G = \mathbb{G}'_{(\sigma.G)}$;
- for any two σ, σ' transformations in $\mathbb{G}'_G$, $\sigma \circ \sigma' \in \mathbb{G}'_G$;
- for any transformation $\sigma \in \mathbb{G}'_H$, $h.\sigma \in \mathbb{G}'_G$;

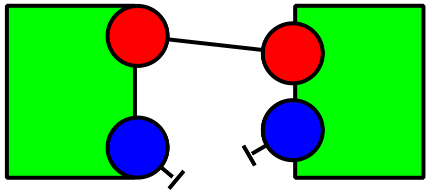
then the groups $(\mathbb{G}'_G)$ define a set of transformations.

Example:
Heterogeneous site permutations

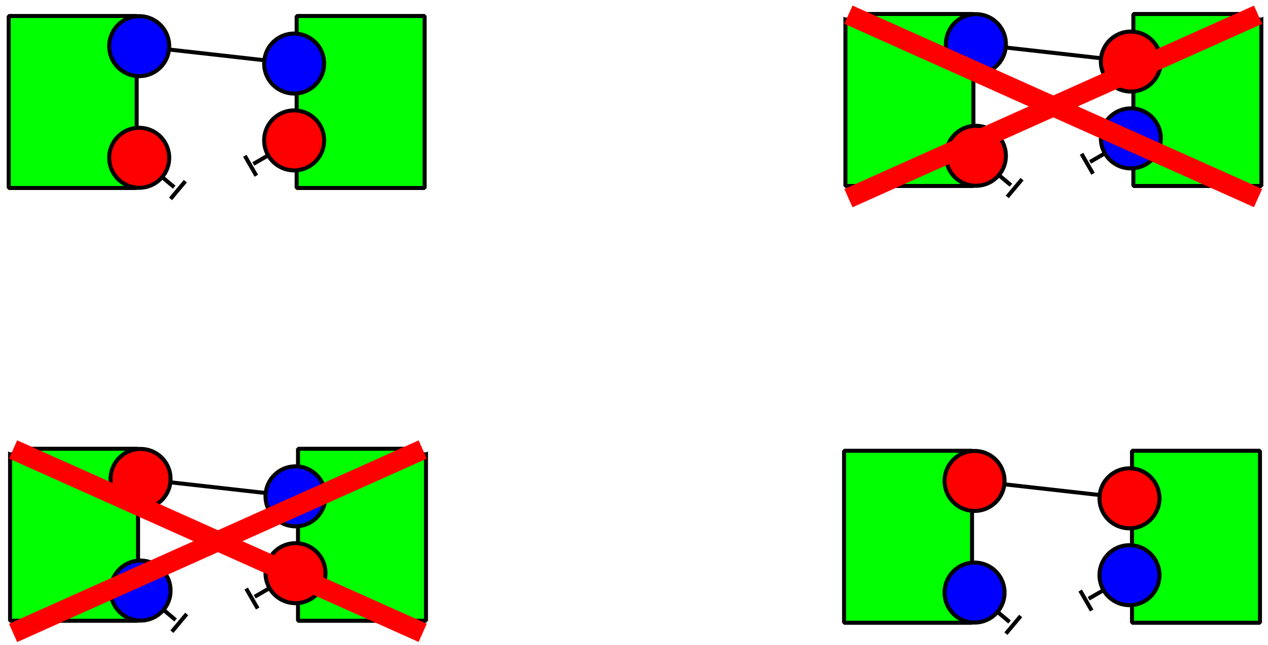








Example:
Homogeneous site permutations



Overview

1. Context and motivations
2. Case study
3. Kappa semantics
4. Symmetries in site-graphs
    (a) Groups of transformations
    (b) Action of the transformations
5. Symmetric models
6. Conclusion

Group actions over site graphs

Let G, G' be two site graphs.

We write $G \approx_{\mathbb{G}} G'$ if and only if there exists $\sigma \in \mathbb{G}_G$ such that $G' = \sigma.G$.

The function:

$$\begin{cases} \mathbb{G}_G \times [G]_{\approx_{\mathbb{G}}} & \rightarrow & [G]_{\approx_{\mathbb{G}}} \\ (\sigma, G) & \mapsto & \sigma.G \end{cases}$$

is a group action.

That is to say:

- $\varepsilon.G = G$;
- $\sigma'.(\sigma.G) = (\sigma' \circ \sigma).G$.

Group actions over embeddings

Let f, f' be two embeddings.

We write $f \approx_{\mathbb{G}} f'$ if and only if there exists $\sigma \in \mathbb{G}_{\mathsf{IM}(f)}$ such that $f' = \sigma.f$.

The function:

$$\begin{cases} \mathbb{G}_{\mathsf{IM}(f)} \times [f]_{\approx_{\mathbb{G}}} & \rightarrow & [f]_{\approx_{\mathbb{G}}} \\ (\sigma, f) & \mapsto & \sigma.f \end{cases}$$

is a group action.

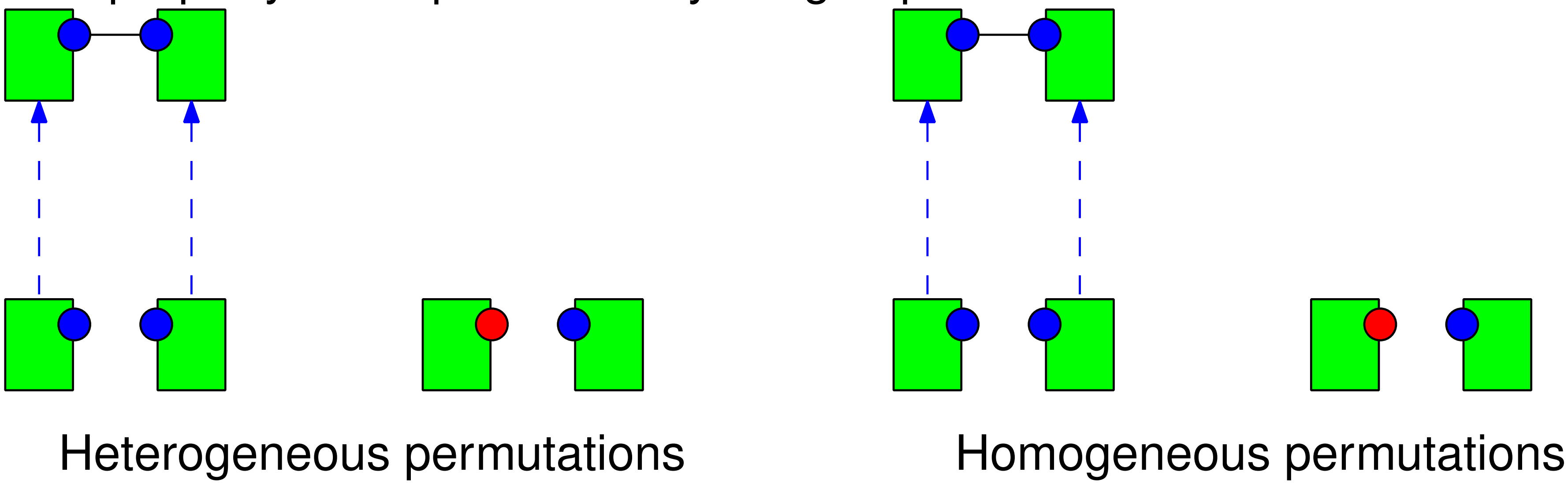
Compatible embeddings

An embedding f between two site graphs G and H is said compatible if and only if:

$$\mathbb{G}_G = \{f.\sigma \mid \sigma \in \mathbb{G}_H\}$$

(that is to say that any transformation that can be applied to the domain of f can be extended to the image of f).

This property is not preserved by subgroups of transformations:



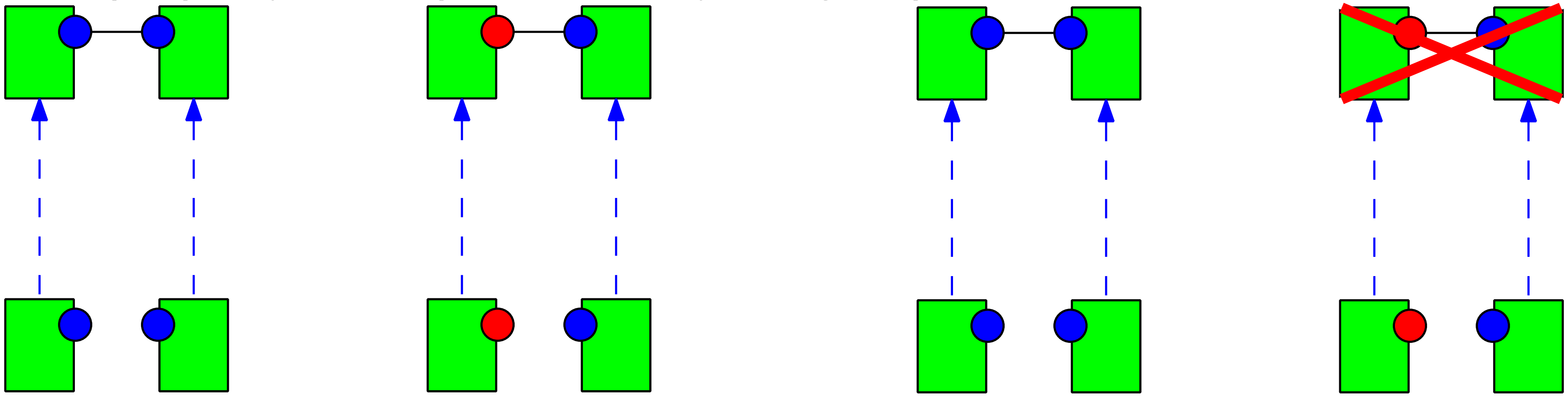
Compatible embeddings

An embedding f between two site graphs G and H is said compatible if and only if:

$$\mathbb{G}_G = \{f.\sigma \mid \sigma \in \mathbb{G}_H\}$$

(that is to say that any transformation that can be applied to the domain of f can be extended to the image of f).

This property is not preserved by subgroups of transformations:



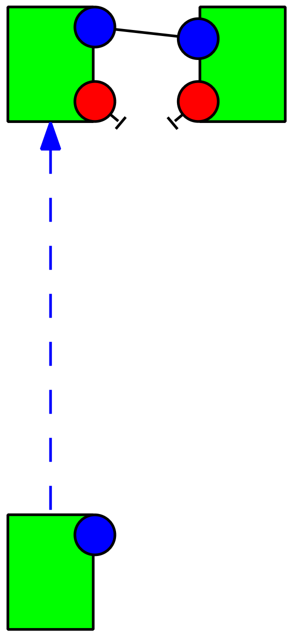
Heterogeneous permutations

Homogeneous permutations

Decomposition of transformations along an embedding

When f is an embedding between two site graphs G and H, we have:

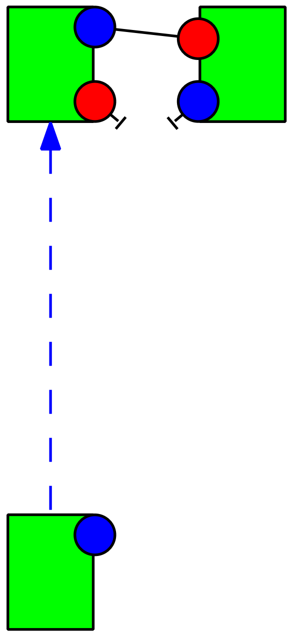
$$\mathbb{G}_H \approx \{\sigma \in \mathbb{G}_H \mid f.\sigma = \varepsilon_G\} \times \{h.\sigma \mid \sigma \in \mathbb{G}_H\}.$$



Decomposition of transformations along an embedding

When f is an embedding between two site graphs G and H, we have:

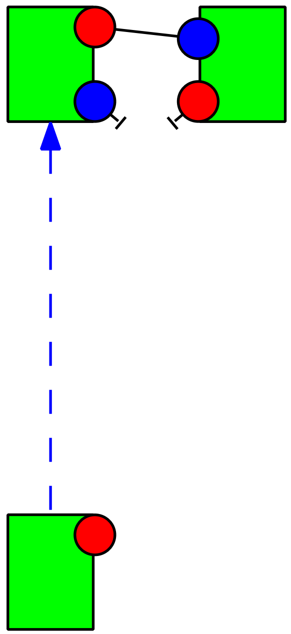
$$\mathbb{G}_H \approx \{\sigma \in \mathbb{G}_H \mid f.\sigma = \varepsilon_G\} \times \{h.\sigma \mid \sigma \in \mathbb{G}_H\}.$$



Decomposition of transformations along an embedding

When f is an embedding between two site graphs G and H, we have:

$$\mathbb{G}_H \approx \{\sigma \in \mathbb{G}_H \mid f.\sigma = \varepsilon_G\} \times \{h.\sigma \mid \sigma \in \mathbb{G}_H\}.$$



Images of isomorphisms

The image of an isomorphism is an isomorphism.

$$
\begin{tikzcd}
\sigma_F.F \arrow[rr, hook, "i_{\sigma_F.F}"] \arrow[dr, hook, "(f.\sigma_F).(f^{-1})"'] & & \sigma_F.F \\
& (f.\sigma_F).E \arrow[ur, hook, "\sigma_F.f"'] &
\end{tikzcd}
$$

The image of an automorphism may be not an automorphism.

Yet, for any site graph G, we have:

$$\mathit{Card}(G) = \mathit{Card}(\{\phi \mid \phi \in \mathit{Aut}(G)\}) \times \mathit{Card}(\{G' \mid G' \approx G \text{ and } G' \approx_{\mathbb{G}} G\}).$$

Group actions over rules

Let $r : L \overset{f}{\hookleftarrow} D \overset{g}{\hookrightarrow} R$ be a rule.

We define the symmetric of r by a symmetry $(\sigma_L, \sigma_R) \in \mathbb{G}_r$ as follows:

$$(\sigma_L, \sigma_R).r \stackrel{\Delta}{=} \sigma_L.L \overset{\sigma_L.f}{\hookleftarrow} (f.\sigma_L).D \overset{\sigma_R.g}{\hookrightarrow} \sigma_R.R$$

We write $r \approx_{\mathbb{G}} r'$ if and only if there exists $\sigma \in \mathbb{G}_r$ such that $r' = \sigma.r$.

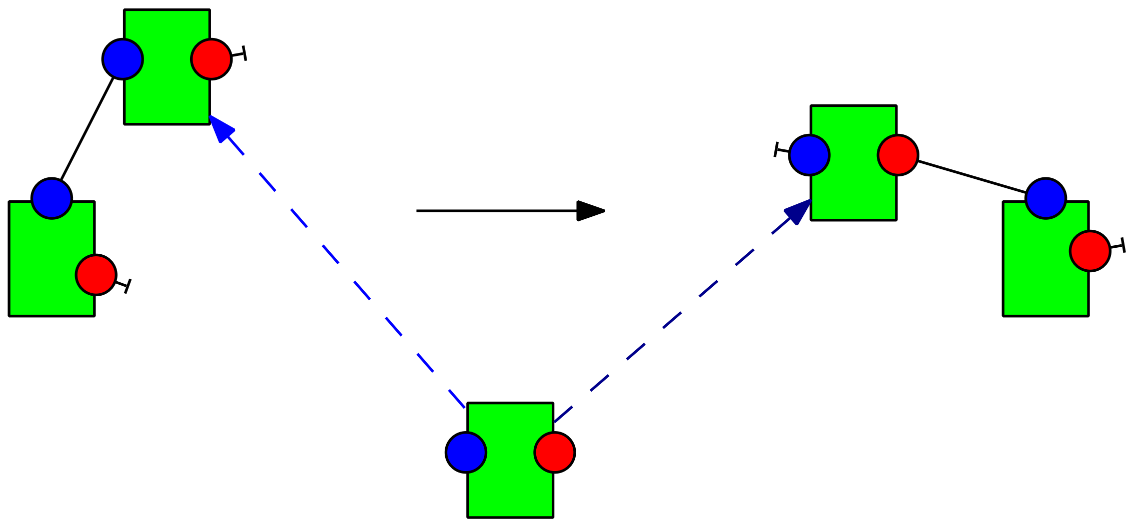
Then:

- $\mathbb{G}_r$ is a group.
- the groups $\mathbb{G}_r$ and $\mathbb{G}_{\sigma.r}$ are the same, for any symmetry $\sigma \in \mathbb{G}_r$.
- The function:

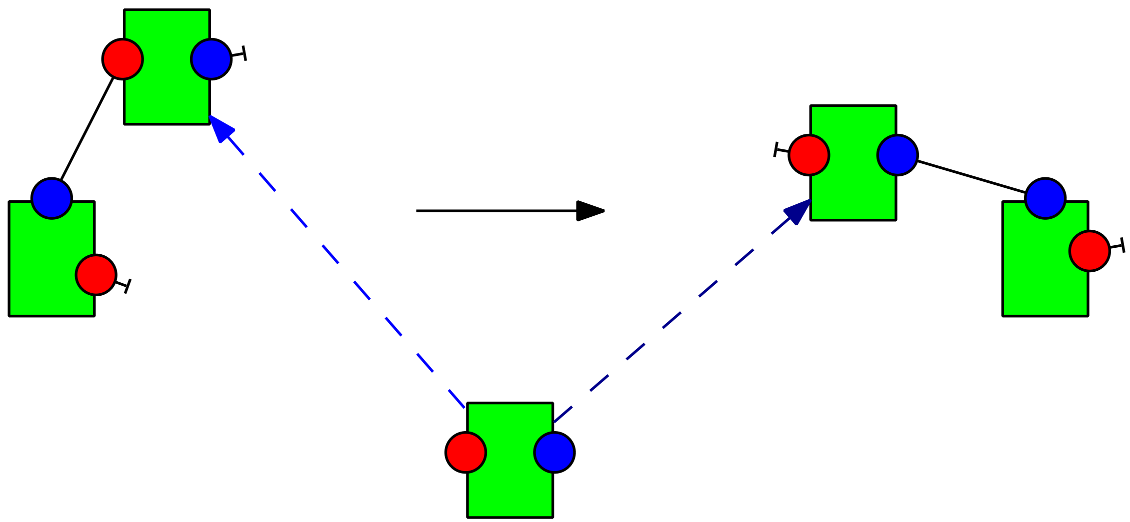
$$\begin{cases} \mathbb{G}_r \times [r]_{\approx_{\mathbb{G}}} & \rightarrow & [r]_{\approx_{\mathbb{G}}} \\ (\sigma, r) & \mapsto & \sigma.r. \end{cases}$$

  is a group action.

Decomposition of the group of transformations over a rule

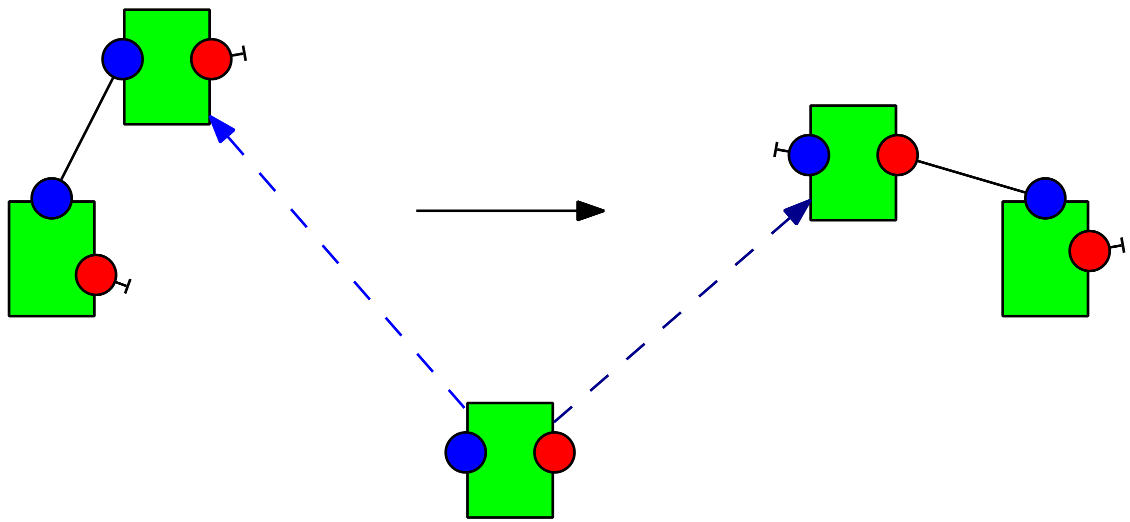


Decomposition of the group of transformations over a rule

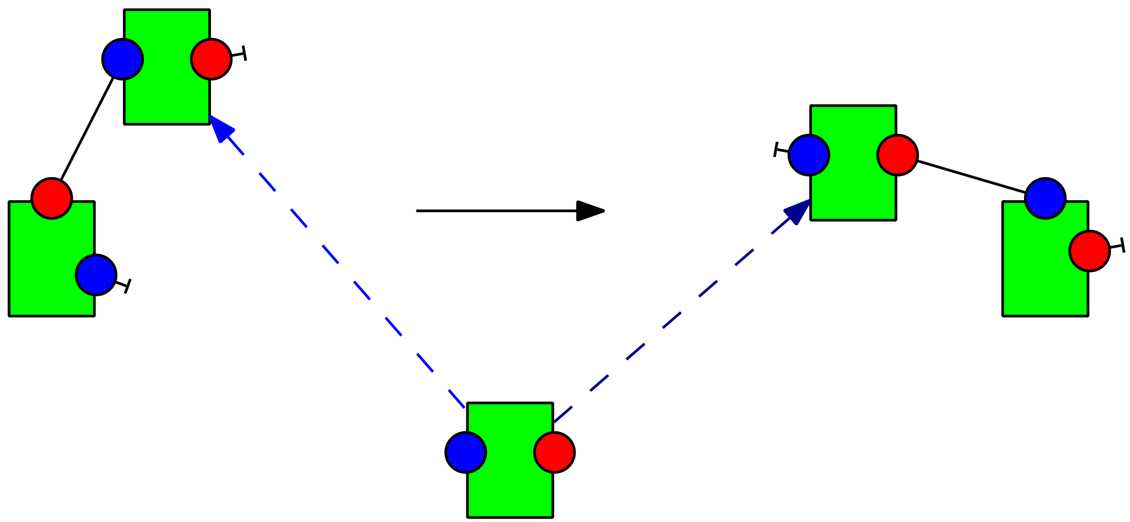


Some transformations operate on the domain of the rule.

Decomposition of the group of transformations over a rule

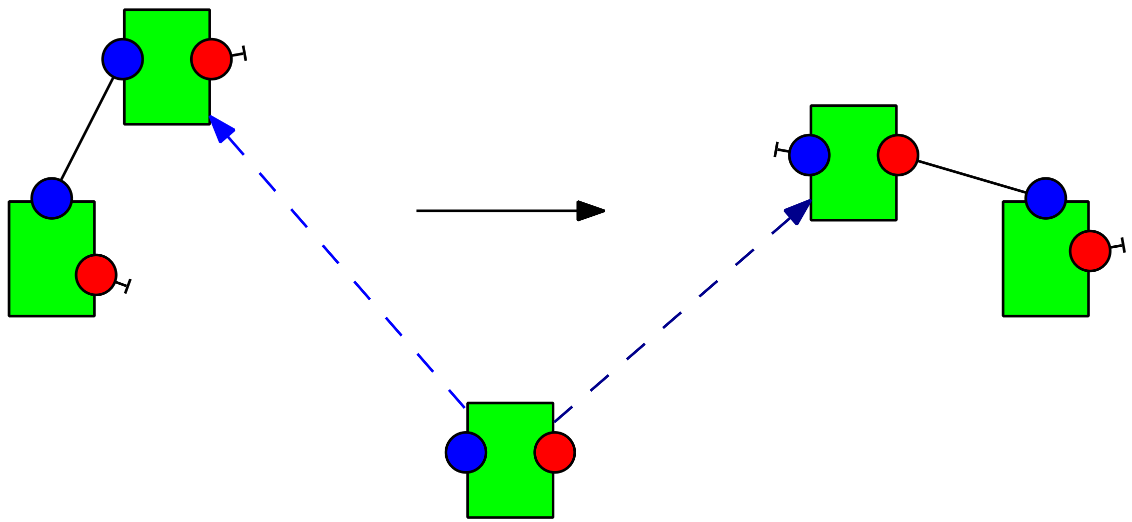


Decomposition of the group of transformations over a rule

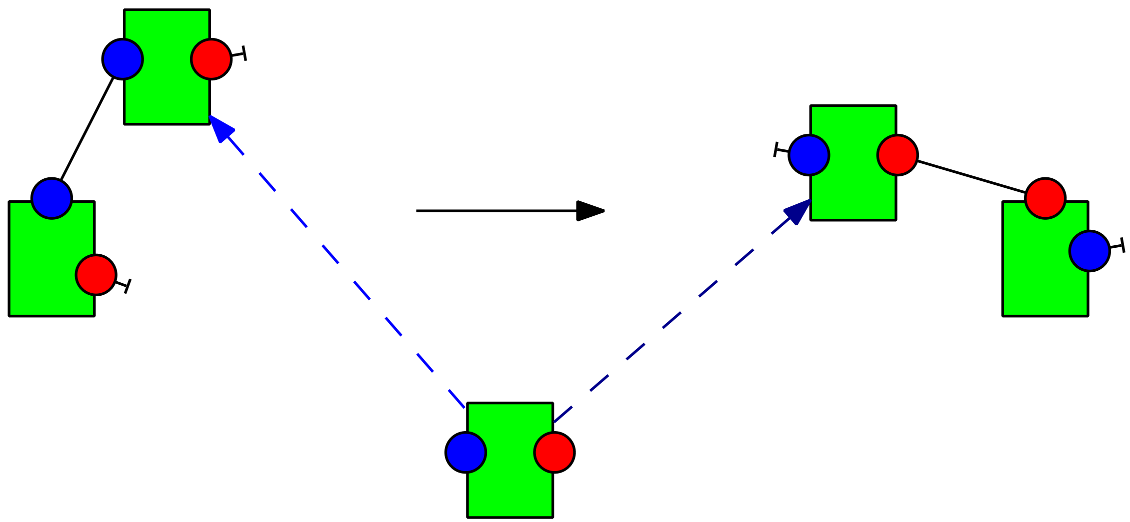


Some transformations operate on degraded agents.

Decomposition of the group of transformations over a rule



Decomposition of the group of transformations over a rule



Some transformations operate on created agents.

Decomposition of the group of transformations over a rule

When $r \ : \ L \overset{f}{\hookleftarrow} D \overset{g}{\hookrightarrow} R$ is a rule,
we have:

$$\mathbb{G}_r \approx \{\sigma \in \mathbb{G}_L \mid f.\sigma = \varepsilon_D\} \times \{\sigma \mid \exists (\sigma_L, \sigma_R) \in \mathbb{G}_r, \ \sigma = f.\sigma_L = f.\sigma_R\} \times \{\sigma \in \mathbb{G}_R \mid g.\sigma = \varepsilon_D\}.$$

Symmetries distribute over:

1. the ones on removed agents;
2. the ones on new agents;
3. the ones on the domain which are compatible with rule.

Group actions over push-out

Theorem 3 Let r be a rule. The function which maps each pair of transformations $(\sigma_L, \sigma_R) \in \mathbb{G}_r$ and each push-out of the form:

$$\begin{tikzcd}
L' \arrow[rr, "r'"] & & R' \\
L \arrow[u, hook, "h_L"] \arrow[rr, "r''"'] & & R \arrow[u, hook, "h_R"'] \arrow[ul, phantom, "\lrcorner" very near start]
\end{tikzcd}$$

with $r' \approx_{\mathbb{G}} r$, to the push-out:

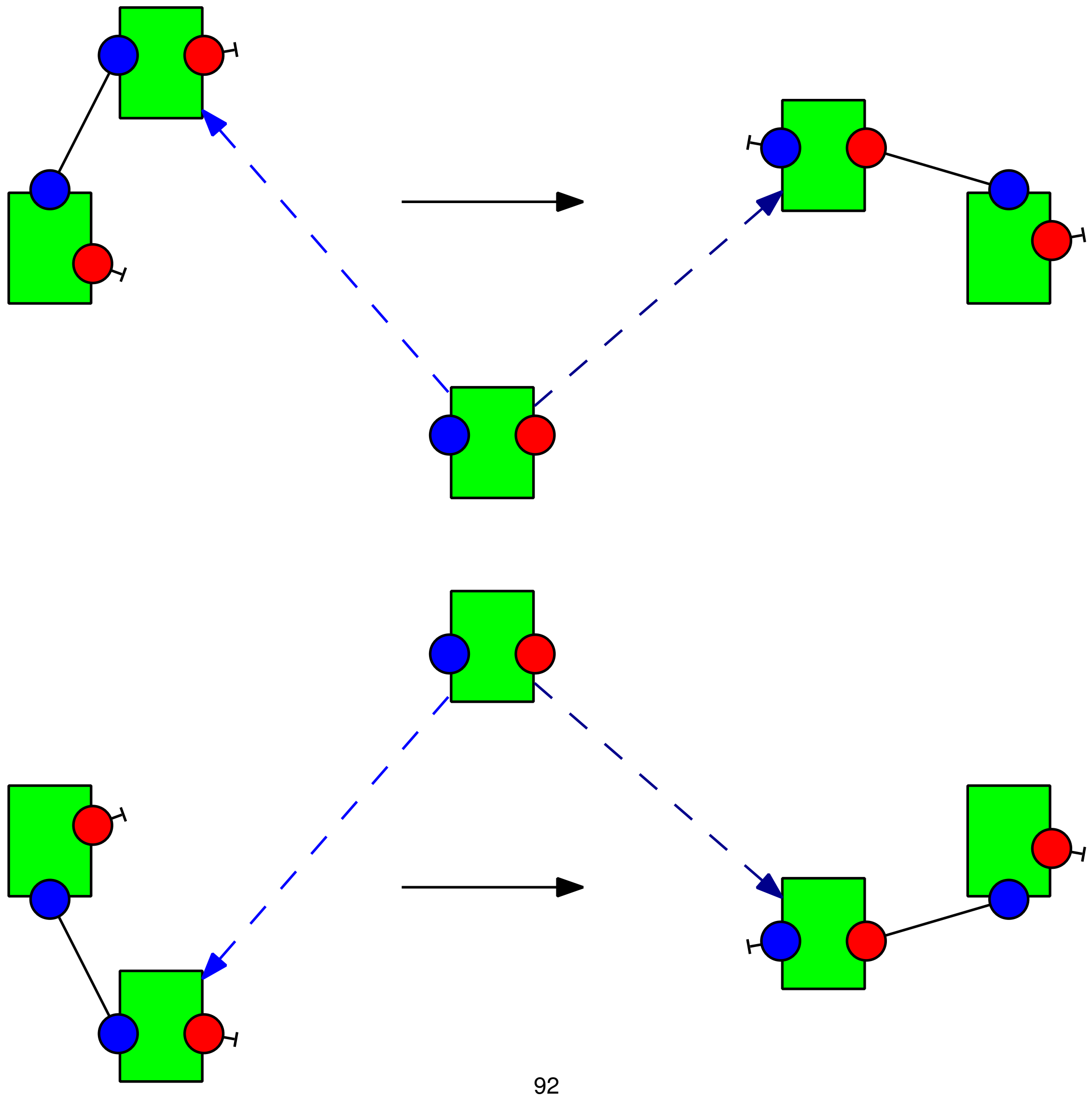
$$\begin{tikzcd}
\sigma_L.L' \arrow[rr, "(\sigma_L,\sigma_R).r'"] & & \sigma_R.R' \\
(h_L.\sigma_L).L \arrow[u, hook, "\sigma_L.h_L"] \arrow[rr, "(h_L.\sigma_L,h_R.\sigma_R).r''"'] & & (h_R.\sigma_R).R \arrow[u, hook, "\sigma_R.h_R"'] \arrow[ul, phantom, "\lrcorner" very near start]
\end{tikzcd}$$

is a group action.

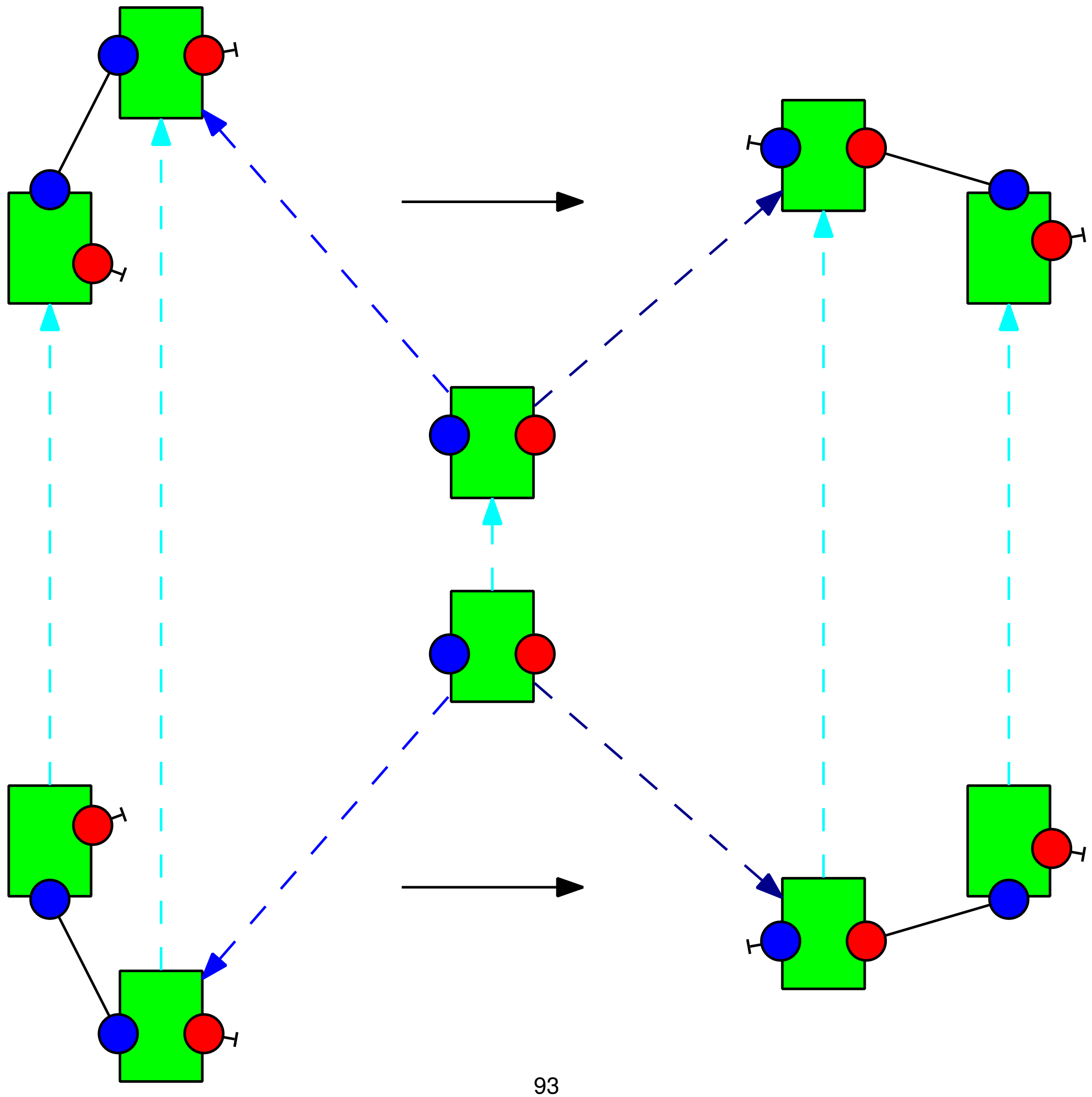
Overview

1. Context and motivations
2. Case study
3. Kappa semantics
4. Symmetries in site-graphs
5. Symmetric models
   (a) Symmetries among set of rules
   (b) Induced bisimulations
6. Conclusion

Isomorphic rules



Isomorphic rules



Symmetric model

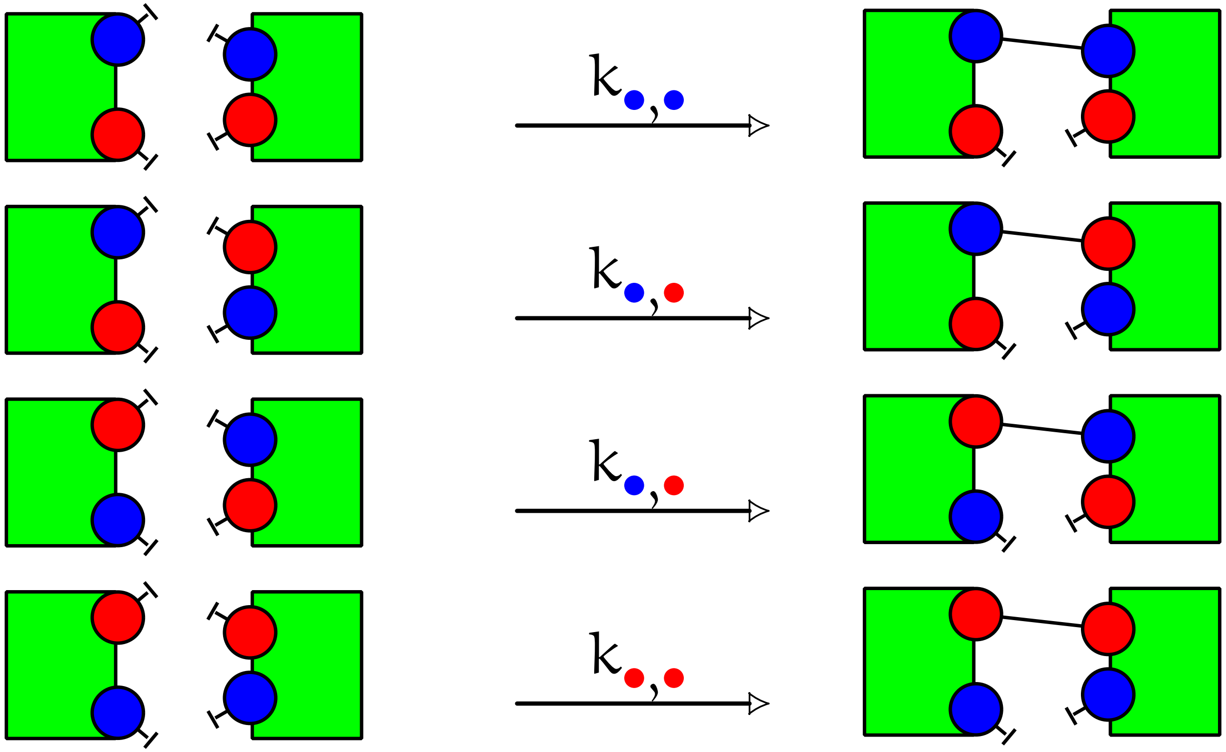
We assume that the model contains atmost one rule per isomorphism class.

A model is $\mathbb{G}$-symmetric if and only if:

- for any rule r in the model and any pair of symmetries $\sigma \in \mathbb{G}_r$, there is (unique) a rule r' in the model that is isomorphic to the rule $\sigma.r$.
- and, with the same notations, we have $g(r) = g(r')$ where:

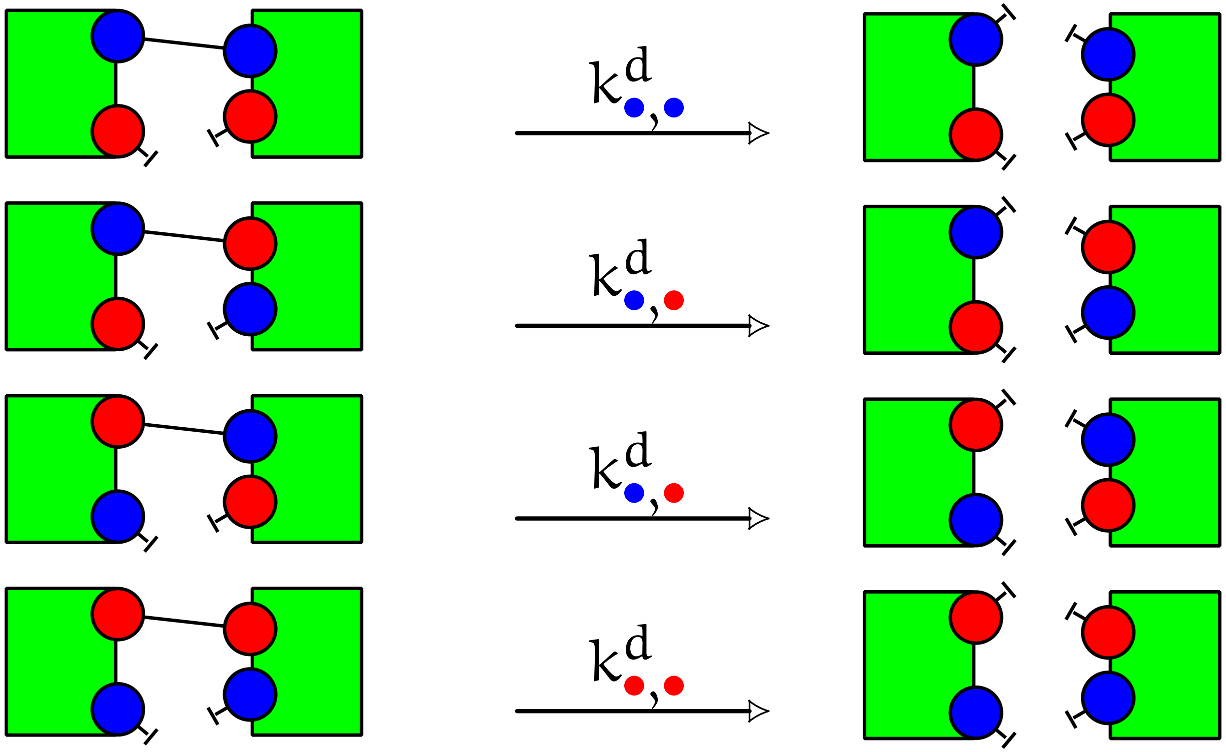
$$g(r) \stackrel{\Delta}{=} \frac{k(r)}{\mathit{card}(\{\sigma \in \mathbb{G}_r \mid \sigma.r \approx r\})\mathit{card}(\mathit{Aut}(\mathit{lhs}(r)))}.$$

Binding rules



$$\frac{k_{{\color{blue}\bullet},{\color{blue}\bullet}}}{1\cdot 2} = \frac{k_{{\color{red}\bullet},{\color{red}\bullet}}}{1\cdot 2} = \frac{k_{{\color{blue}\bullet},{\color{red}\bullet}}}{2\cdot 2}$$

Unbinding rules



$$\frac{k^{d}_{{\color{blue}\bullet},{\color{blue}\bullet}}}{1\cdot 2} = \frac{k^{d}_{{\color{red}\bullet},{\color{red}\bullet}}}{1\cdot 2} = \frac{k^{d}_{{\color{blue}\bullet},{\color{red}\bullet}}}{2\cdot 1}$$

Overview

1. Context and motivations
2. Case study
3. Kappa semantics
4. Symmetries in site-graphs
5. Symmetric models
   (a) Symmetries among set of rules
   (b) Induced bisimulations
6. Conclusion

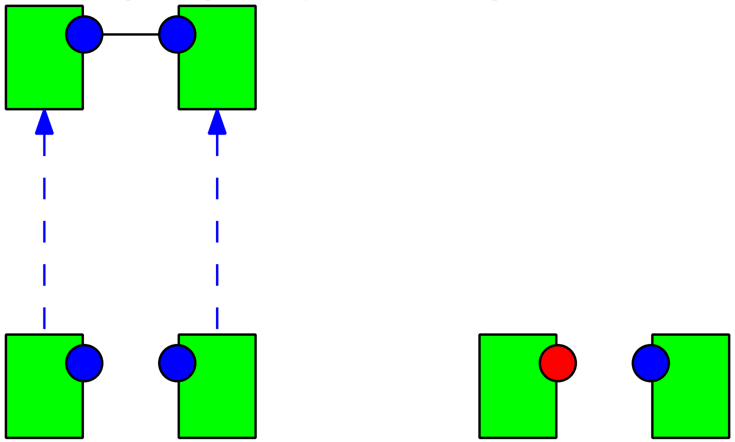
Compatible embeddings (reminders)

An embedding f between two site graphs G and H is said compatible if and only if:

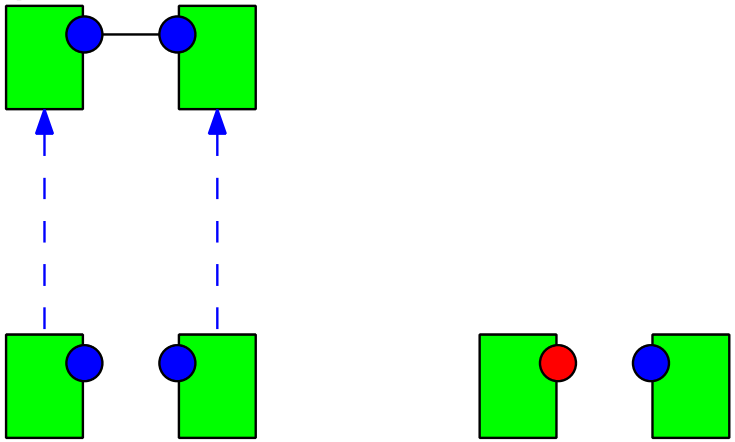
$$\mathbb{G}_G = \{f.\sigma \mid \sigma \in \mathbb{G}_H\}$$

(that is to say that any transformation that can be applied to the domain of f can be extended to the image of f).

This property is not preserved by subgroups of transformations:



Heterogeneous permutations



Homogeneous permutations

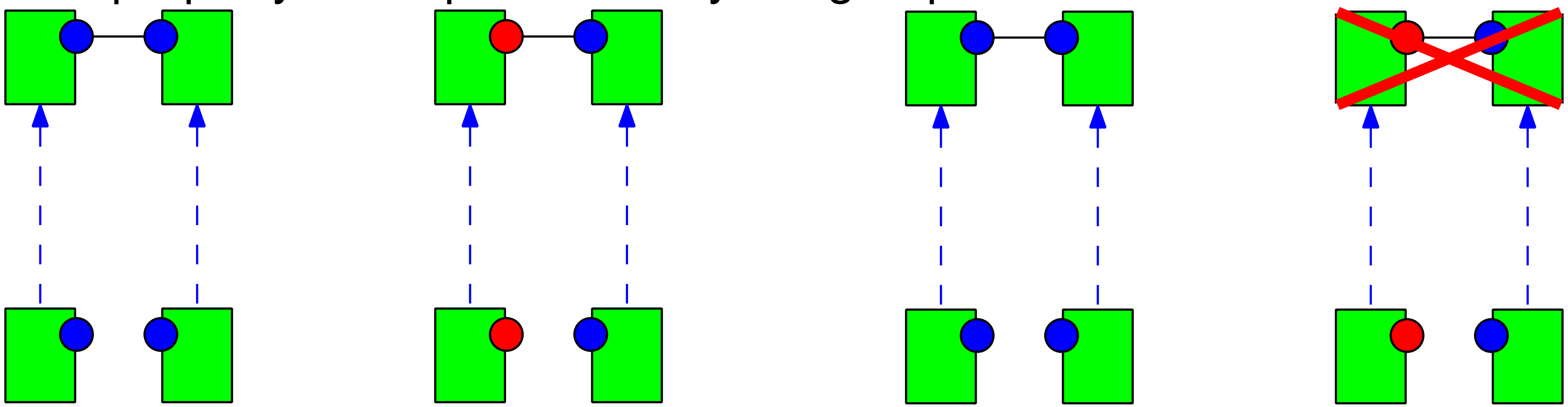
Compatible embeddings (reminders)

An embedding f between two site graphs G and H is said compatible if and only if:

$$\mathbb{G}_G = \{f.\sigma \mid \sigma \in \mathbb{G}_H\}$$

(that is to say that any transformation that can be applied to the domain of f can be extended to the image of f).

This property is not preserved by subgroups of transformations:

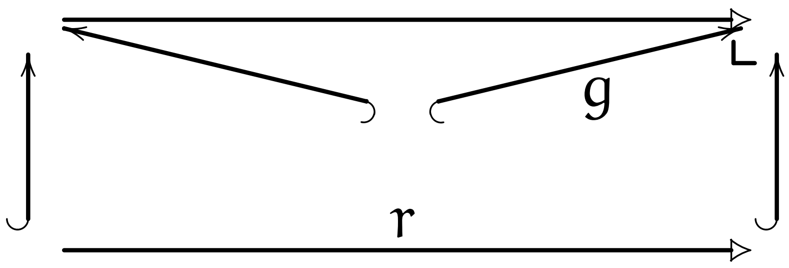


Heterogeneous permutations

Homogeneous permutations

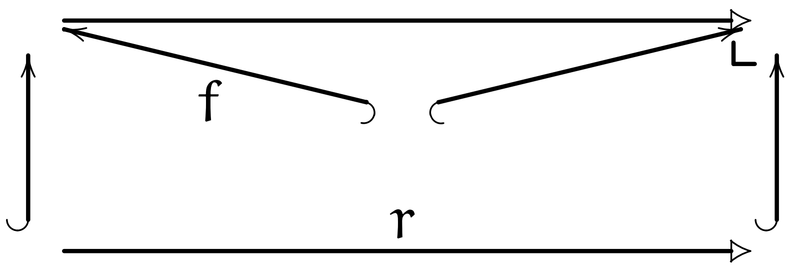
Compatible rules

We say that a rule r is forward-compatible if and only if, for any push-out of the following form:



the embedding g is compatible.

We say that a rule r is backward-compatible if and only if, for any push-out of the following form:



the embedding f is compatible.

Lumping states

We say that two states $q, q' \in \mathcal{Q}$ are isomorphic if and only if there exist $M \in q$ and $M' \in q'$ such that $M \approx_{\mathbb{G}} M'$.

In such a case, we write $q \approx_{\mathbb{G}} q'$.
$\approx_{\mathbb{G}}$ is an equivalence relation.

Lumping the transtion labels

We say that two labels $(r, C) \in \mathcal{L}$ and $(r', C') \in \mathcal{L}$ are isomorphic if and only if there exist an embedding $f \in C$, an embedding $f' \in C'$, a pair of symmetries $(\sigma_{L'}, \sigma_R) \in \mathbb{G}_{\mathsf{IM}(f)} \times \mathbb{G}_{\mathsf{rhs}(r)}$ such that $(f.'\sigma_{L'}, \sigma_R) \in \mathbb{G}_r$ and two isomorphisms ϕ and ψ such that the following diagram commutes:

$$\begin{tikzcd}
\cdot \arrow[r, dotted] & \cdot \\
\cdot \arrow[u, hook, "\sigma_{L'}.''f"'] \arrow[r, "(f.'\sigma_{L'},\sigma_R).r"] & \cdot \arrow[u, hook, dotted] \\
\cdot \arrow[u, "\approx\phi"] \arrow[r, "r'"] \arrow[uu, hook, bend left, "f'"] & \cdot \arrow[u, "\approx\psi"']
\end{tikzcd}$$

In such a case, we write $(r, C) \approx_{\mathbb{G}} (r', C')$ (this is also an equivalence relation).

Weighted flow

Let $X, X' \subseteq \mathcal{Q}$ and $Y \subseteq \mathcal{L}$.
Let ω be a function from $\mathcal{Q}$ to $\mathbb{R}^+$.

We define the flow from X to X' via Y, weighted by the reward function ω by:

$$\mathsf{FLOW}_\omega\,(X, Y, X') \stackrel{\Delta}{=} \sum_{q \in X, q' \in X', \lambda \in Y,\ q \xrightarrow{\lambda} q'} \omega(q)\mathsf{RATE}(\lambda)$$

Forward bisimulation

Theorem 4 Let $q, q', q'' \in \mathcal{Q}$ such that $q \approx_{\mathbb{G}} q'$. Let $\lambda \in \mathcal{L}$.
If the model is symmetric and if the rules of the models are forward-compatible, then the following equality holds:

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

with $\omega(q_1) = 1$ for any $q_1 \in \mathcal{Q}$.

Backward bisimulation (DTMC)

Theorem 5 Let $q, q', q'' \in \mathcal{Q}$ such that $q' \approx_{\mathbb{G}} q''$. Let $\lambda \in \mathcal{L}$.
If the model is symmetric and if the rules of the models are backward-compatible, then the following equality holds:

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

with $\omega(q_1) \stackrel{\Delta}{=} \dfrac{1}{\mathit{card}(\mathit{Aut}(q))}$, for any $q_1 \in \mathcal{Q}$.

Backward bisimulation (CTMC)

Theorem 6 Let $q, q', q'' \in \mathcal{Q}$ such that $q' \approx_{\mathbb{G}} q''$. Let $\lambda \in \mathcal{L}$.
If the model is symmetric and if the rules of the models are both forward- and backward-compatible,
then the following equalities holds:

1. $\mathsf{FLOW}_{\omega}(\{q'\}, \mathcal{Q}, \mathcal{L}) = \mathsf{FLOW}_{\omega}(\{q''\}, \mathcal{Q}, \mathcal{L})$,
   with $\omega(q_1) = 1$ for any $q_1 \in \mathcal{Q}$;
2. $\omega(q'')\mathsf{FLOW}_{\omega}\left([q]_{\approx_{\mathbb{G}}}, [\lambda]_{\approx_{\mathbb{G}}}, \{q'\}\right) = \omega(q')\mathsf{FLOW}_{\omega}\left([q]_{\approx_{\mathbb{G}}}, [\lambda]_{\approx_{\mathbb{G}}}, \{q''\}\right)$,
   with $\omega(q_1) \stackrel{\Delta}{=} \dfrac{1}{\mathit{card}(\mathit{Aut}(q))}$, for any $q_1 \in \mathcal{Q}$.

Overview

1. Context and motivations
2. Case study
3. Kappa semantics
4. Symmetries in site-graphs
5. Symmetric models
6. Conclusion

Conclusion

A fully algebraic framework to infer and use symmetries in Kappa;

- Compatible with the SPO semantics (see [FSTTCS'2012]);
- Can handle side-effects (see the paper);
- Induces forward and/or back and forth bisimulations;
- Can be applied to discover model reductions for the qualitative semantics, the ODEs semantics, and the stochastic semantics [MFPSXXVII];
- Can be combined with other exact model reductions [MFPSXXVI].

This framework is cleaner and more general that the process algebra based one [MFPSXXVII].

Camporesi et al., Combining model reductions. MFPS XXVI (2010)
Camporesi et al., Formal reduction of rule-based models, MFPS XXVII (2011)
Danos et al., Rewriting and Pathway Reconstruction for Rule-Based Models, FSTTCS 2012

Future work

- Investigate which specific classes of symmetries and which specific classes of rules ensure that rules are forward and/or backward compatible with the symmetries;
- Check the compatibility with the DPO (Double Push-Out) semantics;
- Design approximate symmetries using bisimulation metrics
(ask Norman Ferns).

“AbstractCell” (2009-2013)

“Big Mechanism” (2014-2017)
“CwC” (2015-2018)

“TGFβSysBio" (2015-2018)