

Reachability Analysis of Rule-based Models

[ICCMSE'07,VMCAI'08]

Jérôme Feret

DI - ÉNS

Inria
informatics mathematics



PSL
RESEARCH
UNIVERSITY

Tuesday, the 30th of October, 2018

In this talk...

We illustrate the following concepts:

- Galois connections:
 - the upper closure operator $\gamma \circ \alpha$,
 - the lower closure operator $\alpha \circ \gamma$;
- soundness:
 - the abstraction forgets no behavior;
- completeness:
 - sufficient conditions that ensure the absence of false positive;

on an abstraction of the reachable connected components in a site-graph rewriting language.

Joint-work with...



Walter Fontana
Harvard Medical School



Vincent Danos
ÉNS



Russ Harmer
ÉNS Lyon

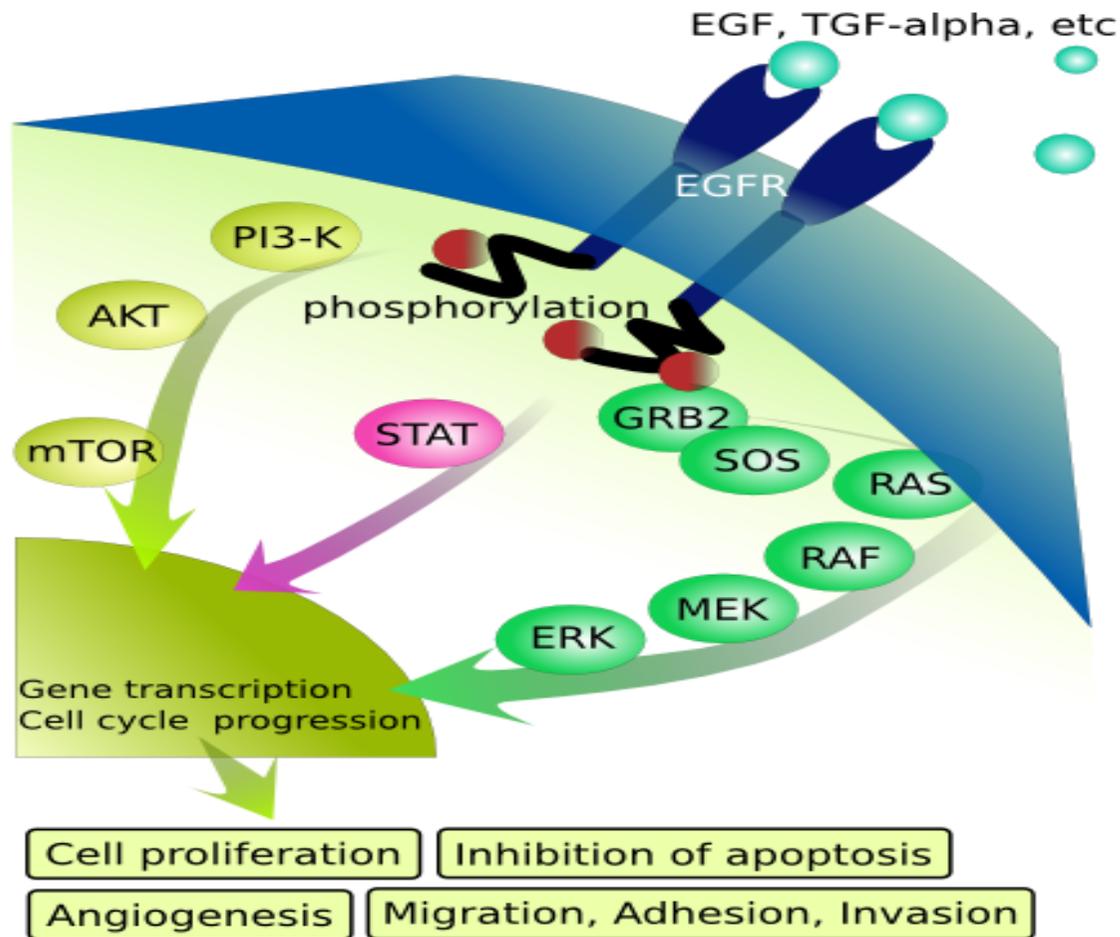


Jean Krivine
Paris VII

Overview

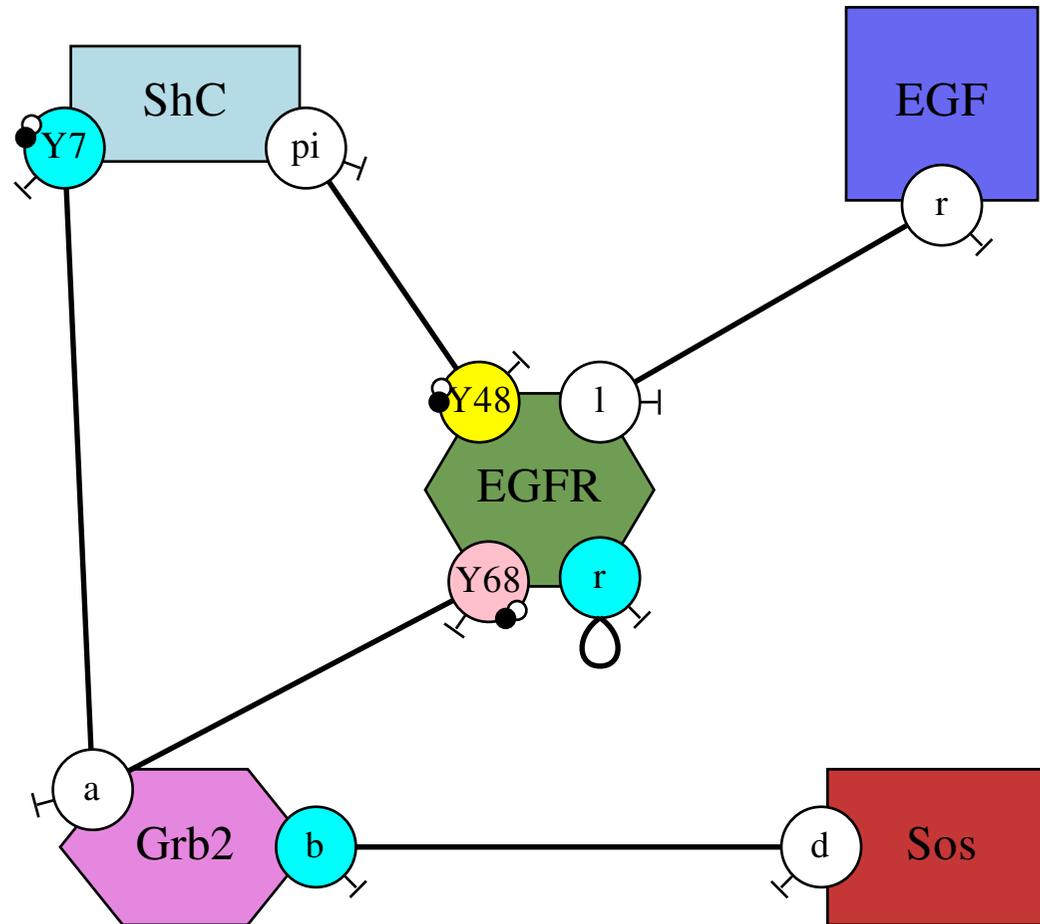
1. Introduction
2. Language: Kappa
3. Abstraction: Local views
4. Completeness: false positives?
5. Local fragment of Kappa
6. Decontextualization
7. Conclusion

Signaling Pathways

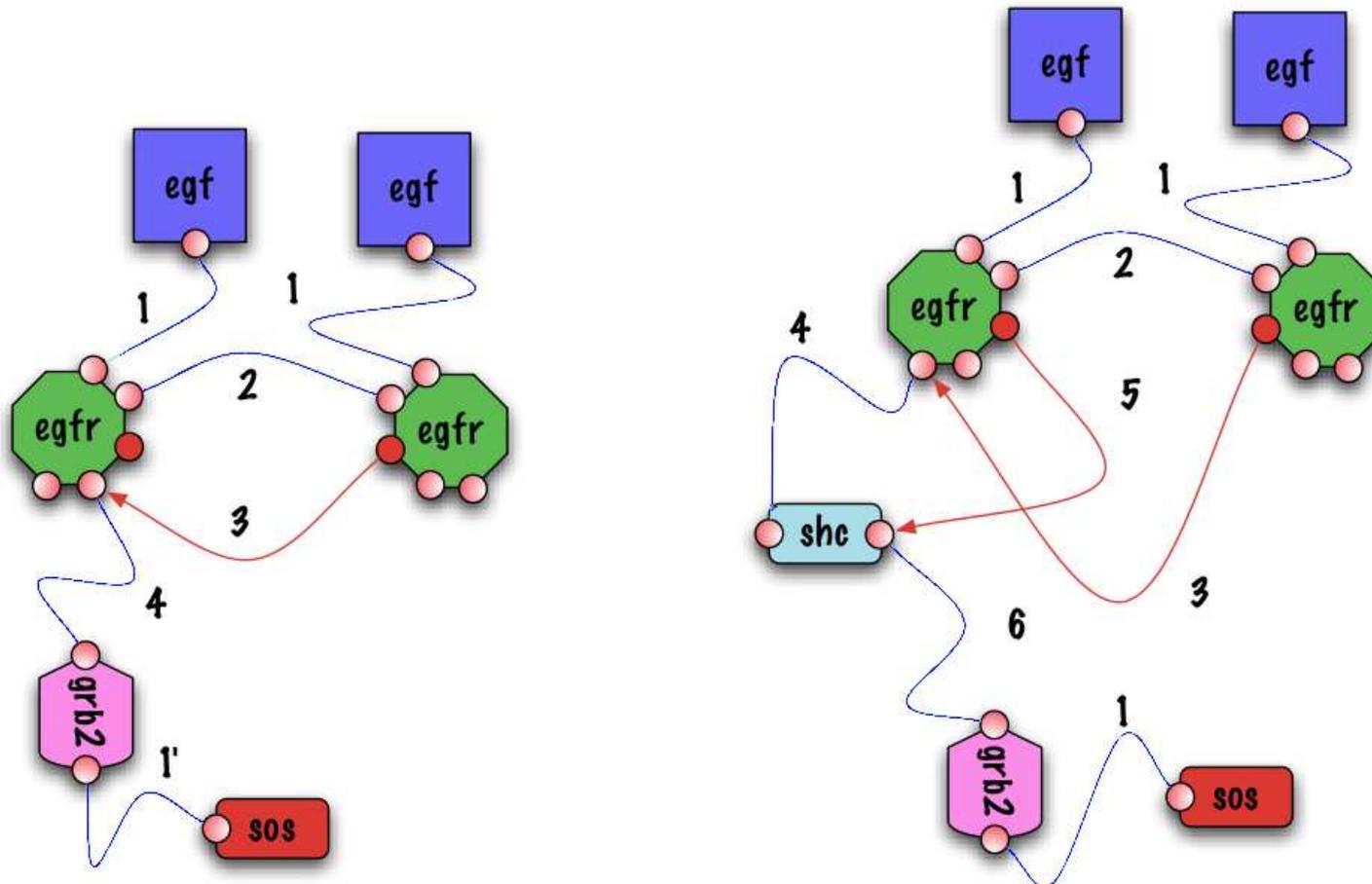


Eikuch, 2007

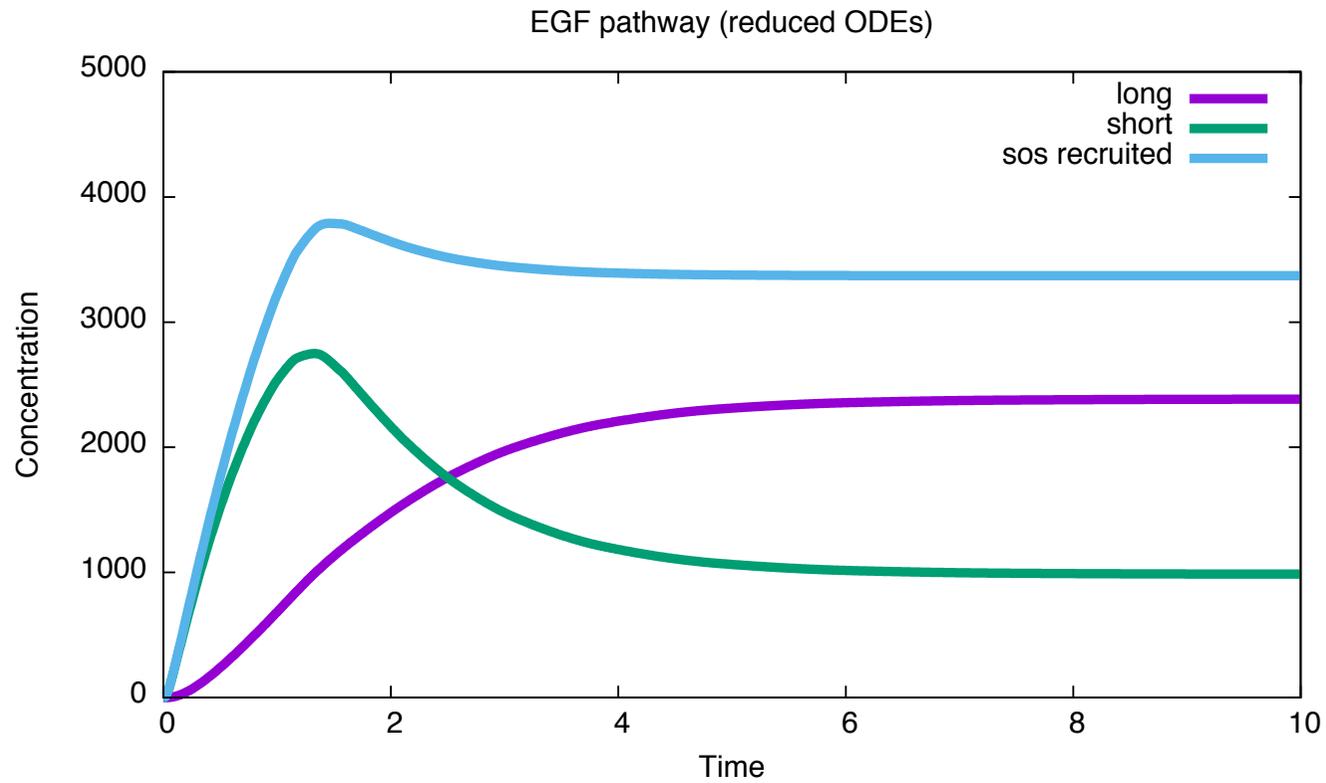
Contact map



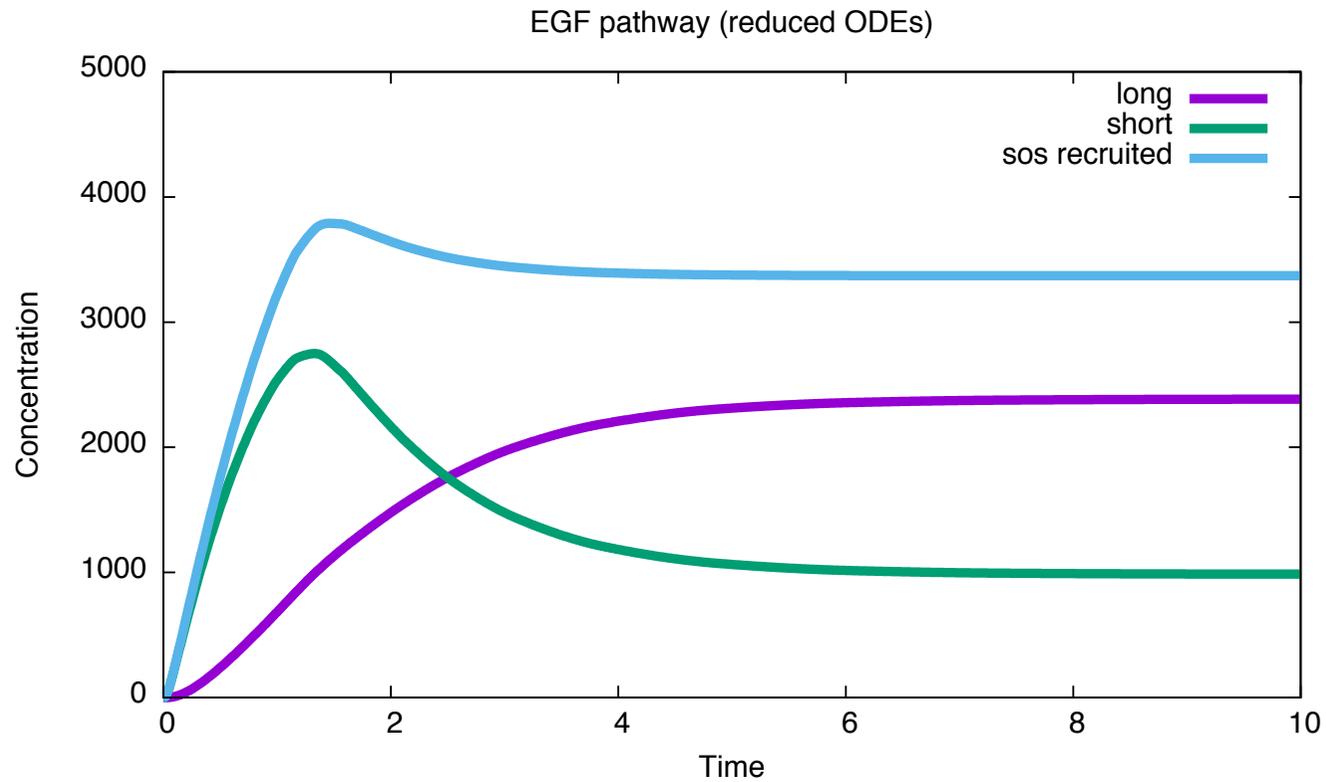
Causal traces



ODE semantics

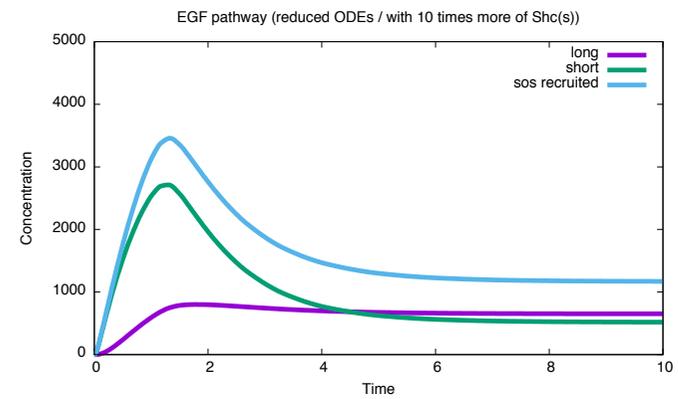
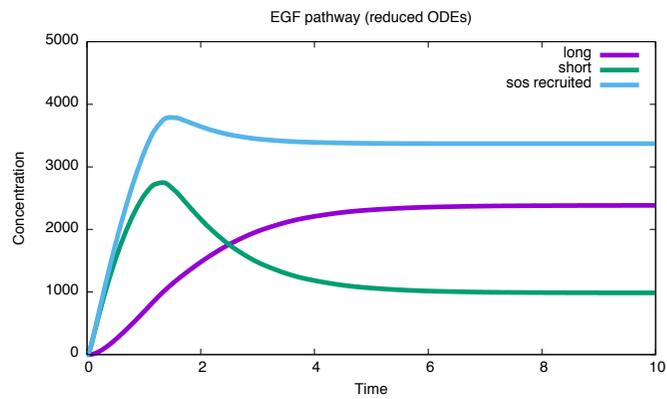


ODE semantics

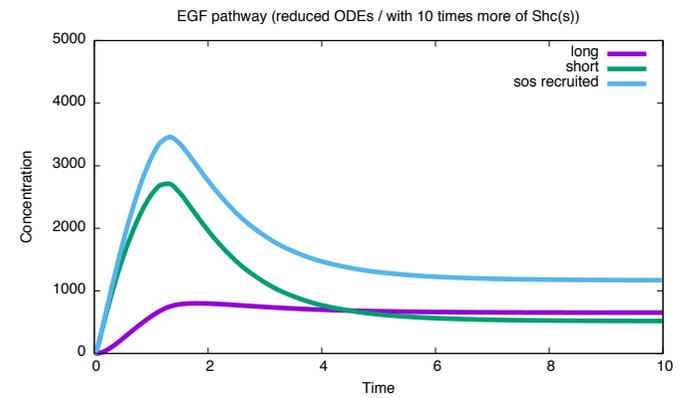
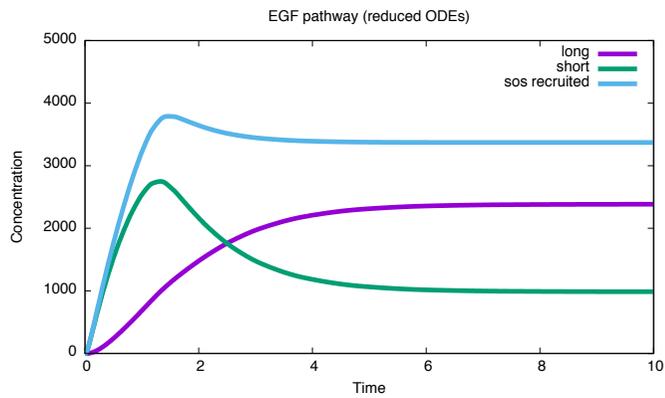
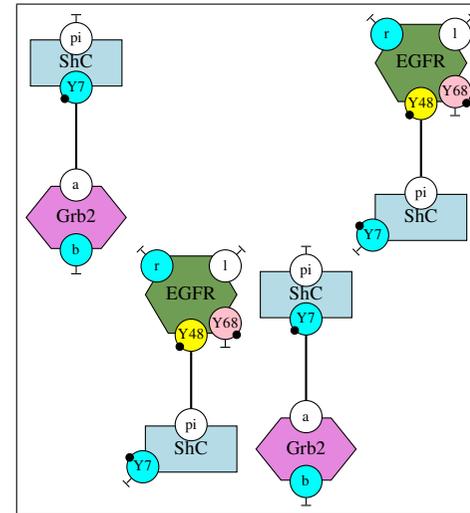
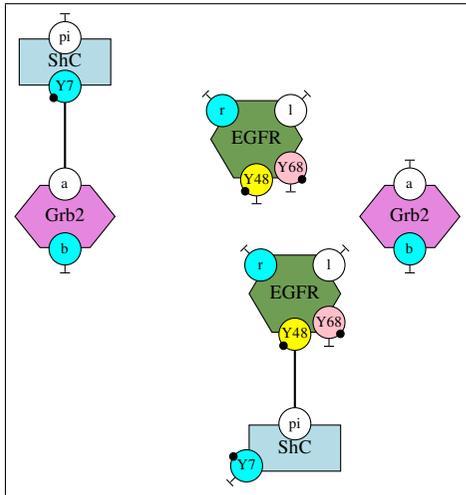


What will happen if more Shc(s) is put in the system?

ODE semantics



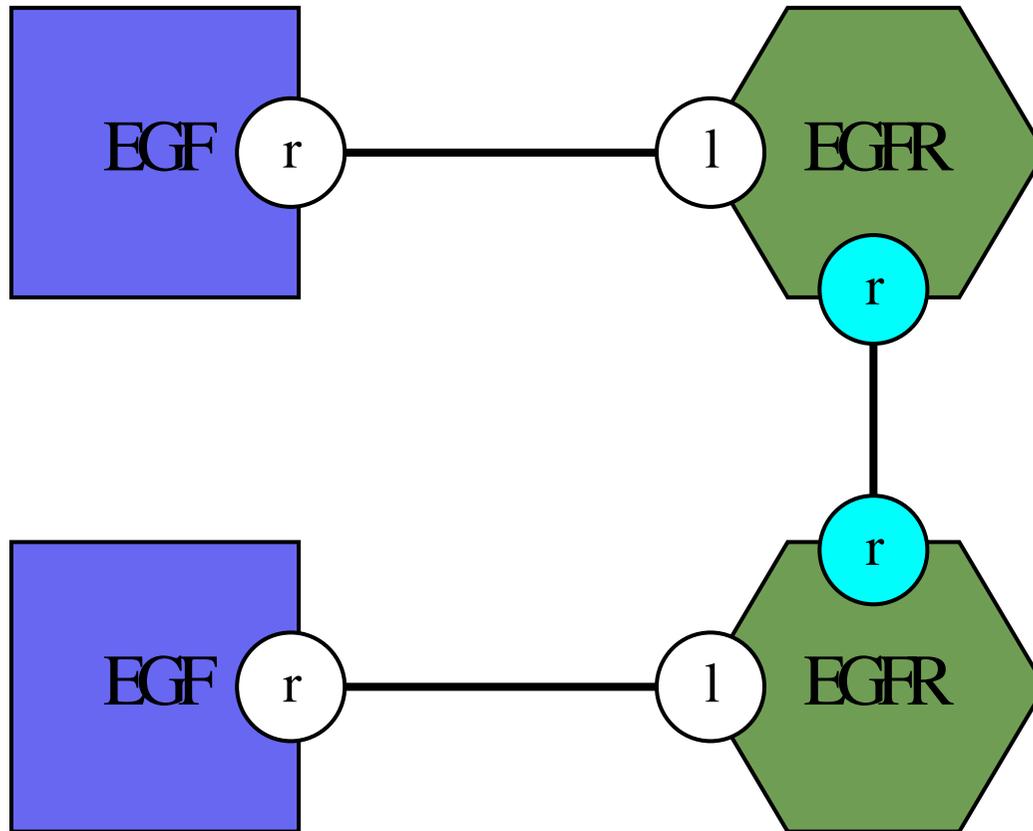
Crowding effect



Overview

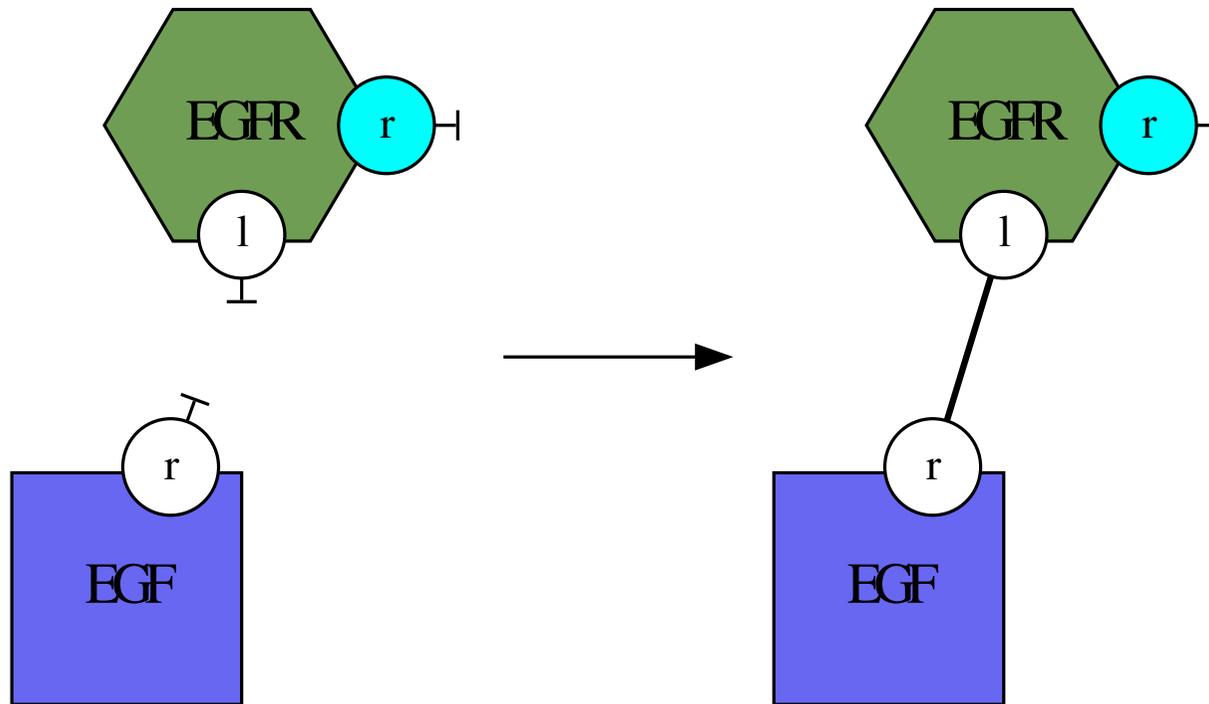
1. Introduction
2. Language: Kappa
3. Abstraction: Local views
4. Completeness: false positives?
5. Local fragment of Kappa
6. Decontextualization
7. Conclusion

A chemical species

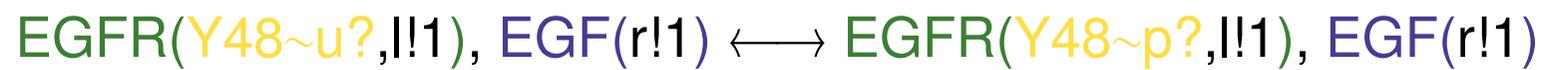
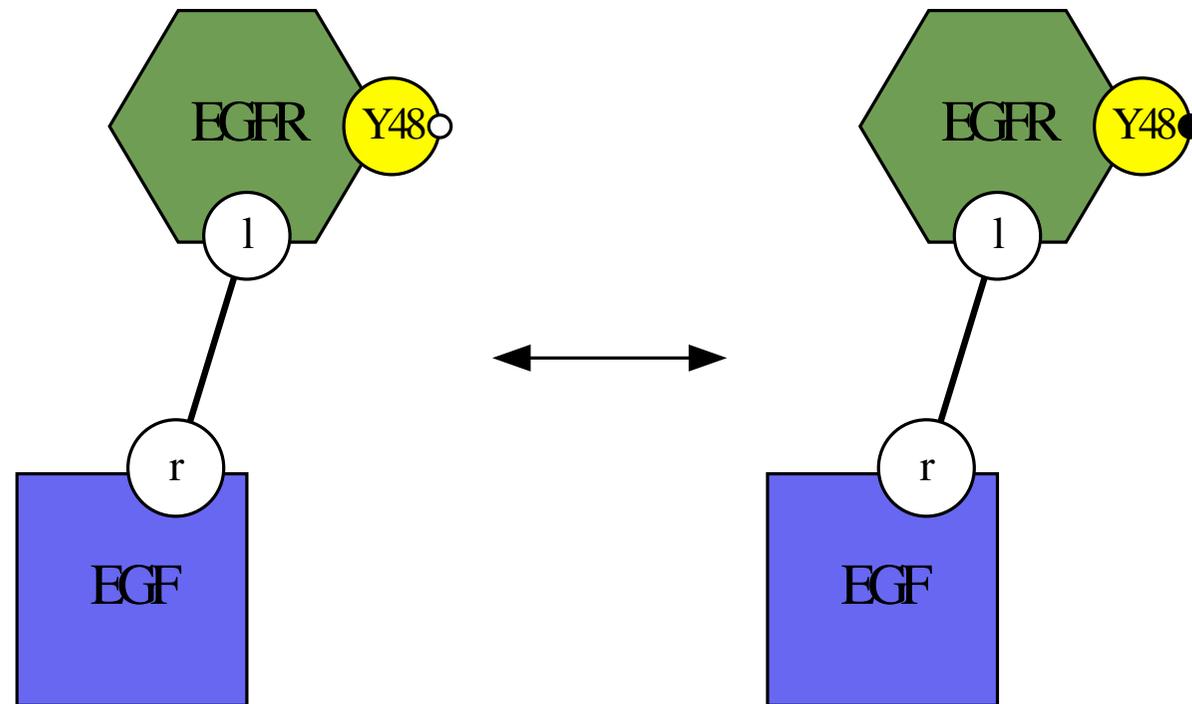


$EGF(r!1)$, $EGFR(l!1, r!2)$, $EGFR(r!2, l!3)$, $EGF(r!3)$

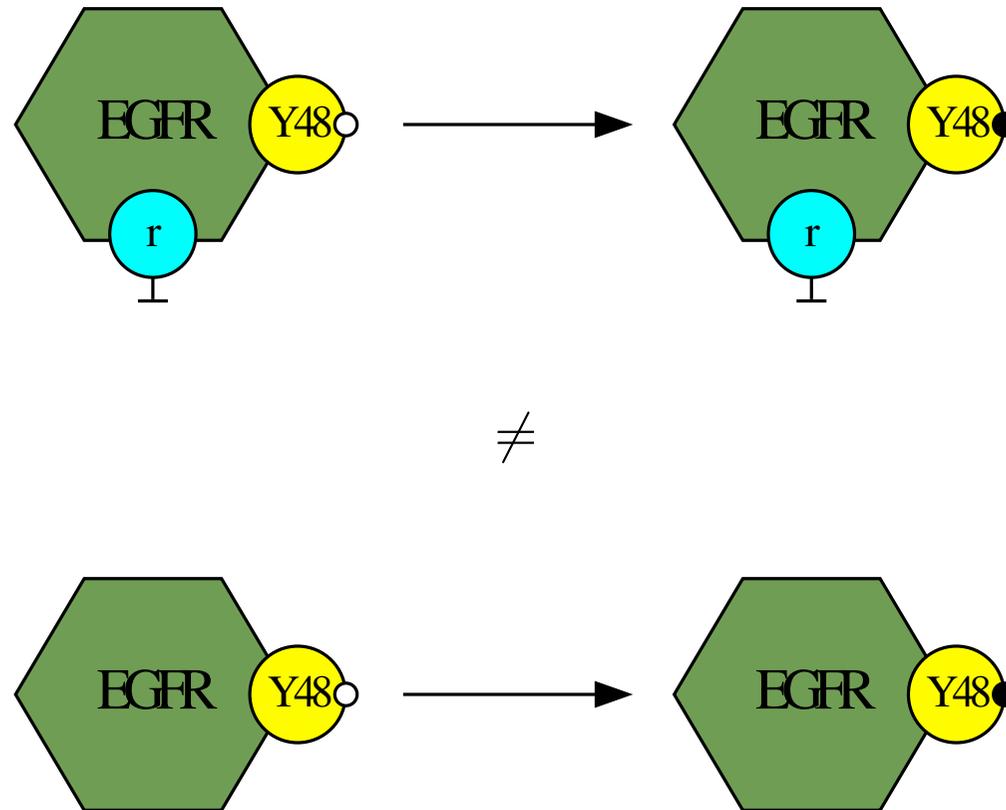
A Unbinding/Binding Rule



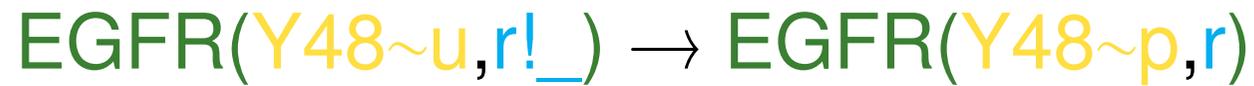
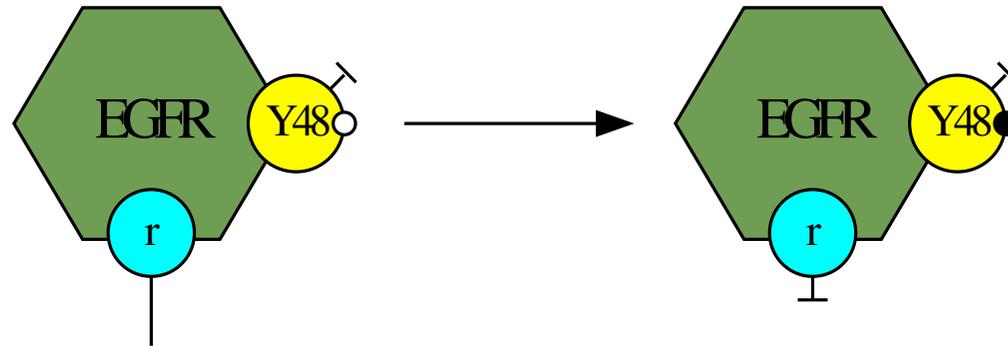
Internal state



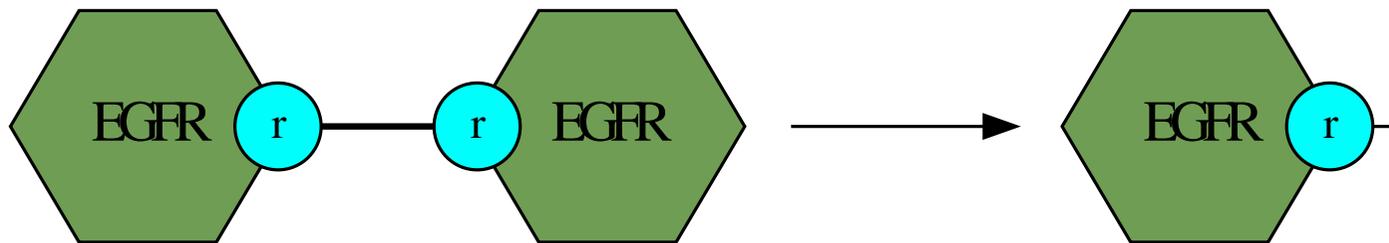
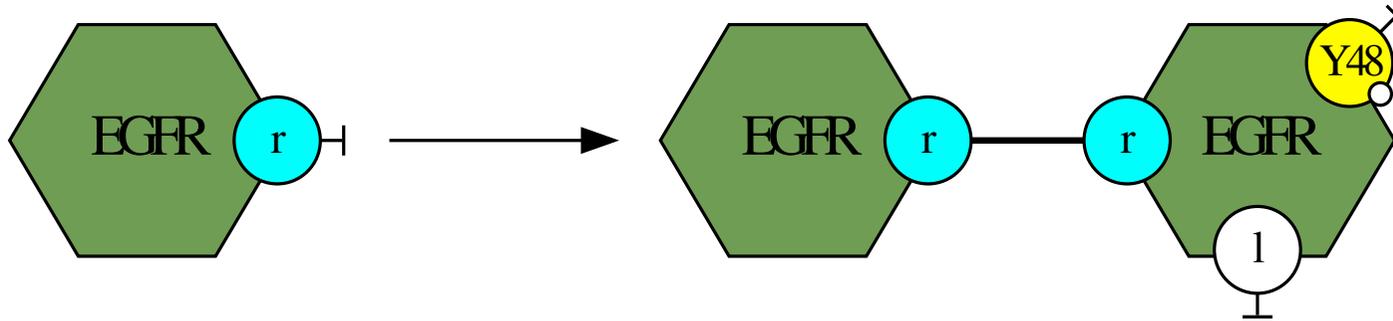
Don't care, Don't write



A contextual rule



Creation/Suppression



Early EGF example

egf rules 1

protein shorthands: E:=egf, R:=egfr, So:=Sos, Sh:=Sh, G:=grb2
site abbreviations & fusions: Y68:=Y1068, Y48:=Y1148/73, Y7:=Y317, π :=PTB/SH2

- Ligand-receptor binding, receptor dimerisation, rtk x-phosph, & de-phosph

- 01: $R(l,r), E(r) \leftrightarrow R(l^1,r), E(r^1)$
- 02: $R(l^1,r), R(l^2,r) \leftrightarrow R(l^1,r^3), R(l^2,r^3)$
- 03: $R(r^1,Y68) \rightarrow R(r^1,Y68^p)$
 $R(Y68^p) \rightarrow R(Y68)$
- 04: $R(r^1,Y48) \rightarrow R(r^1,Y48^p)$
 $R(Y48^p) \rightarrow R(Y48)$

receptor type: $R(l,r,Y68,Y48)$

- Sh x-phosph & de-phosph

- 14: $R(r^2,Y48^{p1}), Sh(\pi^1,Y7) \rightarrow R(r^2,Y48^{p1}), Sh(\pi^1,Y7^p)$
- ??: $Sh(\pi^1,Y7^p) \rightarrow Sh(\pi^1,Y7)$
- 16: $Sh(\pi,Y7^p) \rightarrow Sh(\pi,Y7)$

refined from
 $Sh(Y7^p) \rightarrow Sh(Y7)$

- Y68-G binding

- 09: $R(Y68^p), G(a,b) \leftrightarrow R(Y68^{p1})+G(a^1,b)$
- 11: $R(Y68^p), G(a,b^2) \leftrightarrow R(Y68^{p1})+G(a^1,b^2)$

refined from
 $R(Y68^p)+G(a) \leftrightarrow R(Y68^{p1})+G(a^1)$

Early EGF example

egf rules 2

refined from
 $So(d)+G(b) \leftrightarrow Sold^1)+G(b^1)$

interface note: highlight
 the interacting parts

- **G-So binding**

- 10: $R(Y68^{p1}), G(a^1, b), So(d) \leftrightarrow R(Y68^{p1}), G(a^1, b^2), Sold^2)$
- 12: $G(a, b), So(d) \leftrightarrow G(a, b^1), Sold^1)$
- 22: $Sh(\pi, Y7^{p2}), G(a^2, b), So(d) \leftrightarrow Sh(\pi, Y7^{p2}), G(a^2, b^1), S(d^1)$
- 19: $Sh(\pi^1, Y7^{p2}), G(a^2, b), So(d) \leftrightarrow Sh(\pi^1, Y7^{p2}), G(a^2, b^1), S(d^1)$

- **Y48-Sh binding**

- 13: $R(Y48^p), Sh(\pi, Y7) \leftrightarrow R(Y48^{p1}), Sh(\pi^1, Y7)$
- 15: $R(Y48^p), Sh(\pi, Y7^p) \leftrightarrow R(Y48^{p1}), Sh(\pi^1, Y7^p)$
- 18: $R(Y48^p), Sh(\pi, Y7^{p1}), G(a^1, b) \leftrightarrow R(Y48^{p2}), Sh(\pi^2, Y7^{p1}), G(a^1, b)$
- 20: $R(Y48^p), Sh(\pi, Y7^{p1}), G(a^1, b^3), S(d^3) \leftrightarrow R(Y48^{p2}), Sh(\pi^2, Y7^{p1}), G(a^1, b^3), S(d^3)$

refined from
 $R(Y48^p)+Sh(\pi) \leftrightarrow R(Y48^{p1})+Sh(\pi^1)$

why not simply $G(b^3)??$

- **Sh-G binding**

- 17: $R(Y48^{p1}), Sh(\pi^1, Y7^p), G(a, b) \leftrightarrow R(Y48^{p1}), Sh(\pi^1, Y7^{p2}), G(a^2, b)$
- 21: $Sh(\pi, Y7^p), G(a, b) \leftrightarrow Sh(\pi, Y7^{p1}), G(a^1, b)$
- 23: $Sh(\pi, Y7^p), G(a, b^2) \leftrightarrow Sh(\pi, Y7^{p1}), G(a^1, b^2)$
- 24: $R(Y48^{p1}), Sh(\pi^1, Y7^p), G(a, b^3), S(d^3) \leftrightarrow R(Y48^{p1}), Sh(\pi^1, Y7^{p2}), G(a^2, b^3), S(d^3)$

refined from
 $Sh(\pi), G(a) \leftrightarrow Sh(\pi^1), G(a^1)$

Properties of interest

1. Show the absence of modeling errors:

- detect dead rules;
- detect overlapping rules;
- detect non exhaustive interactions;
- detect rules with ambiguous molecularity.

2. Get idiomatic description of the networks:

- capture causality;
- capture potential interactions;
- capture relationships between site states;
- simplify rules.

3. Allow fast simulation:

- capture accurate approximation of the wake-up relation.

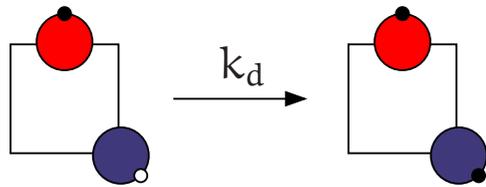
Overview

1. Introduction
2. Language: Kappa
3. **Abstraction: Local views**
4. Completeness: false positives?
5. Local fragment of Kappa
6. Decontextualization
7. Conclusion

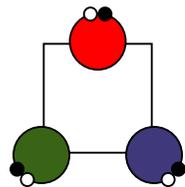
Concrete semantics

A rule is a symbolic representation of a multi-set of reactions.

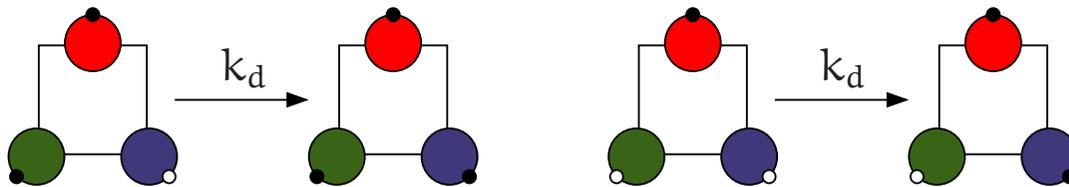
For instance, the rule:



within a model with the following signature:



denotes the following two reactions:



Set of reachable chemical species

Let $\mathcal{R} = \{R_i\}$ be a set of rules.

Let *Species* be the set of all chemical species ($C, c_1, c'_1, \dots, c_k, c'_k, \dots \in \textit{Species}$).

Let *Species*₀ be the set of initial .

We are interested in *Species*_ω the set of all chemical species that can be constructed in one or several applications of the reactions induced by the rules in \mathcal{R} , starting from the set *Species*₀ of initial chemical species.

(We do not care about the number of occurrences of each chemical species).

Inductive definition

We define the mapping \mathbb{F} as follows:

$$\mathbb{F} : \begin{cases} \wp(\mathit{Species}) & \rightarrow \wp(\mathit{Species}) \\ X & \mapsto X \cup \left\{ c'_j \mid \begin{array}{l} \exists R_k \in \mathcal{R}, c_1, \dots, c_m \in X, \\ c_1, \dots, c_m \xrightarrow{R_k} c'_1, \dots, c'_n \end{array} \right\}. \end{cases}$$

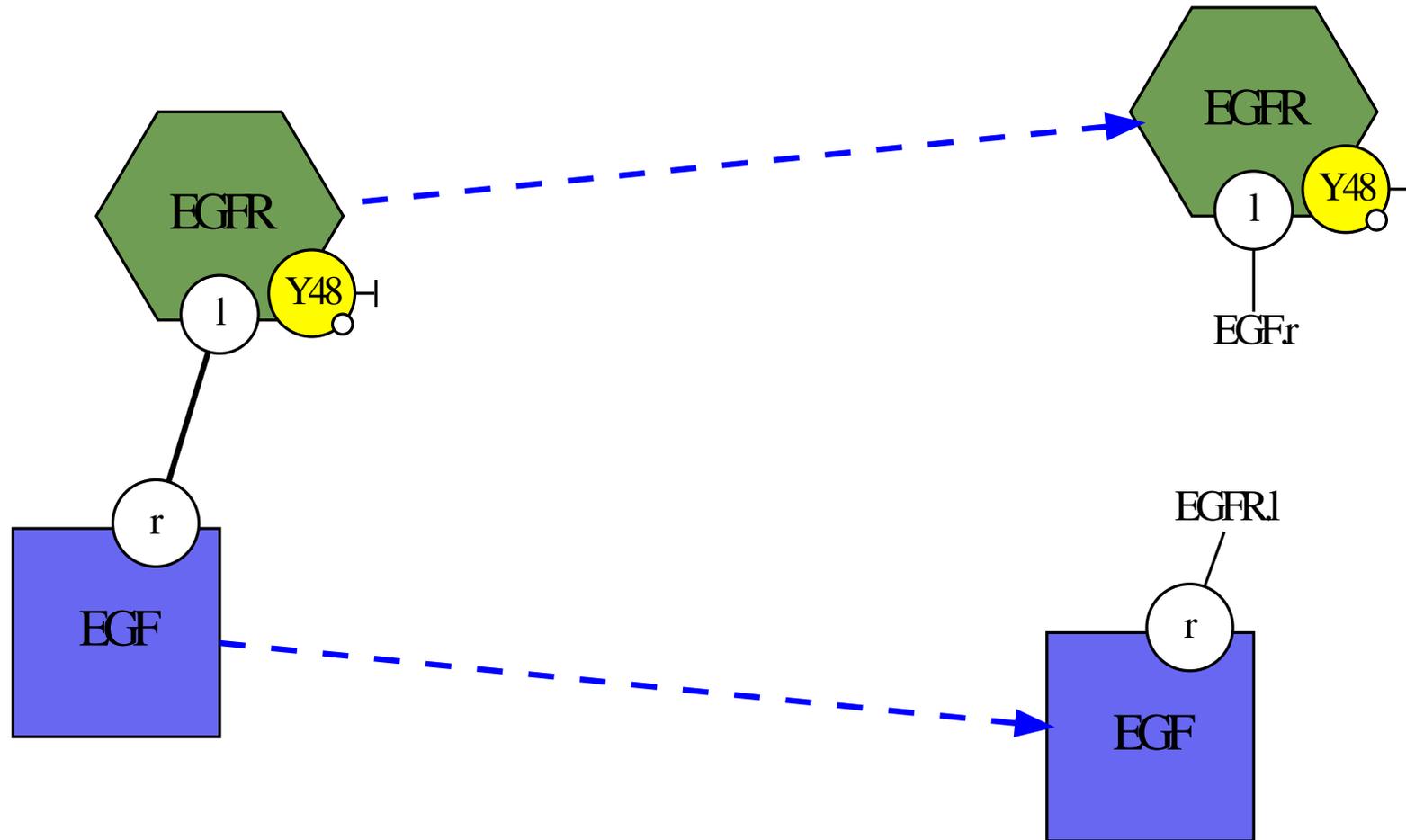
The set $\wp(\mathit{Species})$ is a complete lattice.

The mapping \mathbb{F} is an extensive \cup -complete morphism.

We define the set of reachable chemical species as follows:

$$\mathit{Species}_\omega = \bigcup \{ \mathbb{F}^n(\mathit{Species}_0) \mid n \in \mathbb{N} \}.$$

Local views



$$\alpha(\{R(Y1 \sim u, !1), E(r!1)\}) = \{R(Y1 \sim u, !r.E); E(r!.R)\}.$$

Galois connection

Let $Local_view$ be the set of all local views.

Let $\alpha \in \wp(Species) \rightarrow \wp(Local_view)$ be the function that maps any set of chemical species into the set of their local views.

The set $\wp(Local_view)$ is a complete lattice.

The function α is a \cup -complete morphism.

Thus, it defines a Galois connection:

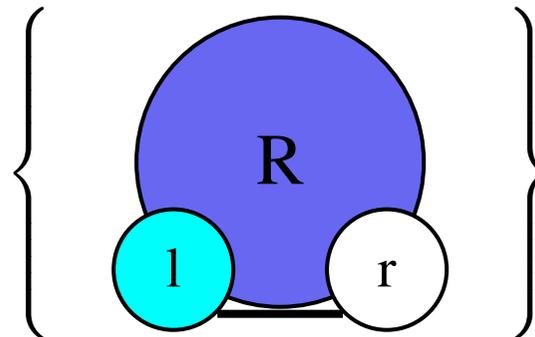
$$\wp(Species) \begin{array}{c} \xleftarrow{\gamma} \\ \xrightarrow{\alpha} \end{array} \wp(Local_view).$$

(The function γ maps a set of local views into the set of complexes that can be built with these local views).

$$\gamma \circ \alpha$$

$\gamma \circ \alpha$ is an upper closure operator: it abstracts away some information.

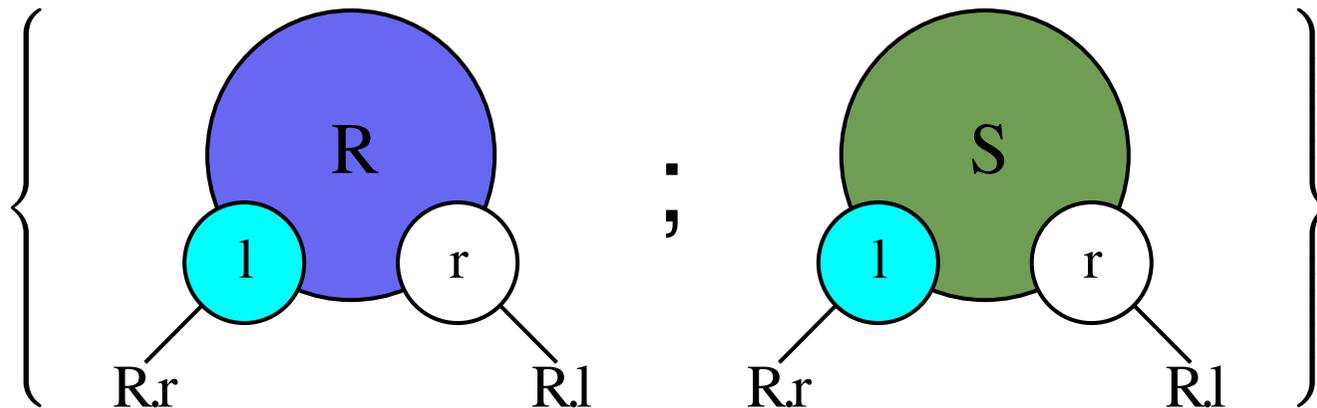
Guess the image of the following set of chemical species ?



$$\alpha \circ \gamma$$

$\alpha \circ \gamma$ is a lower closure operator: it simplifies (or reduces) constraints.

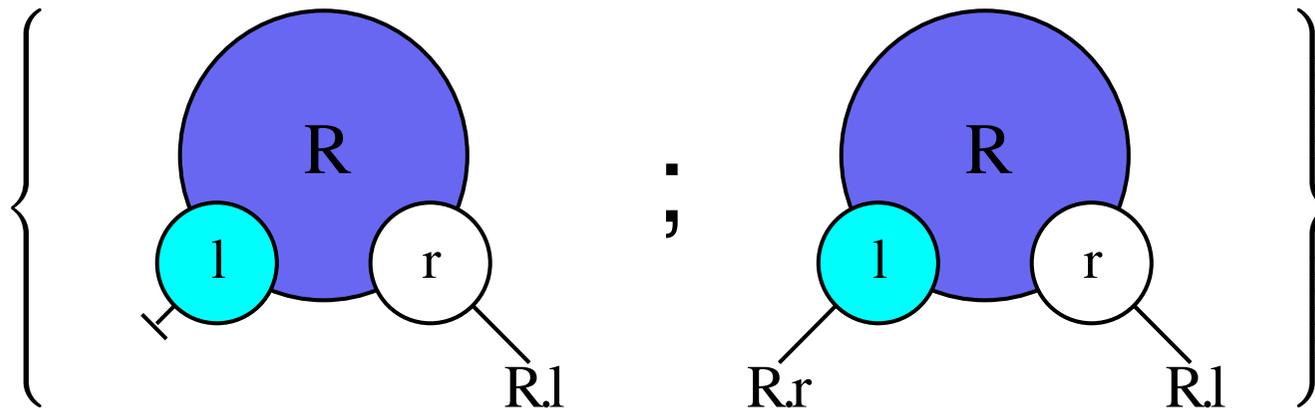
Guess the image of the following set of local views ?



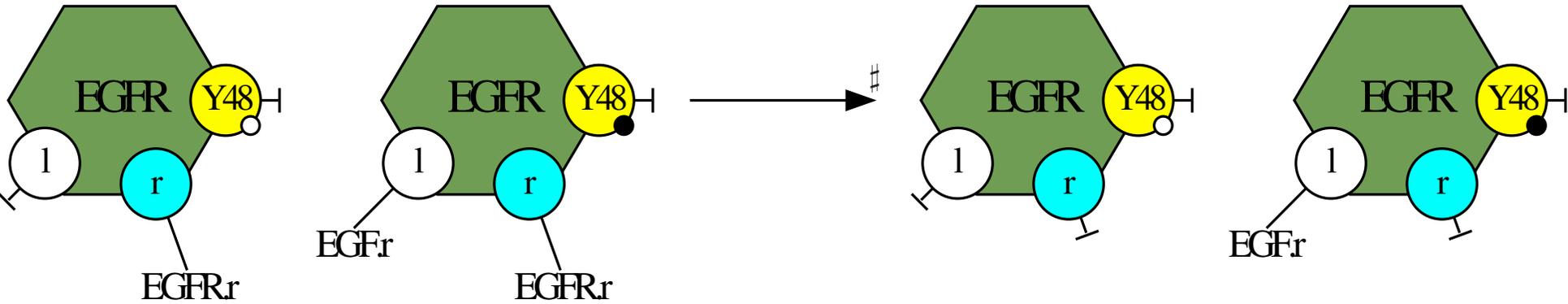
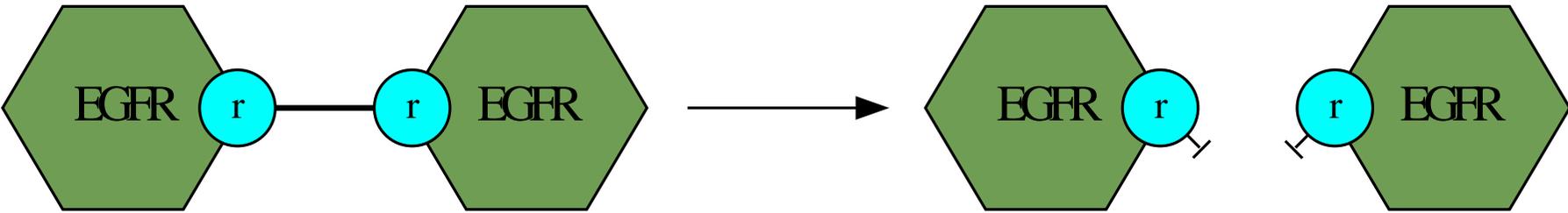
One more question

$\alpha \circ \gamma$ is a lower closure operator: it simplifies (or reduces) constraints.

Guess the image of the following set of local views ?



Abstract reactions



Abstract counterpart to \mathbb{F}

We define $\mathbb{F}^\#$ as:

$$\mathbb{F}^\# : \begin{cases} \wp(\text{Local_view}) & \rightarrow \wp(\text{Local_view}) \\ Y & \mapsto Y \cup \left\{ IV'_j \mid \begin{array}{l} \exists R_k \in \mathcal{R}, IV_1, \dots, IV_m \in Y, \\ IV_1, \dots, IV_m \xrightarrow{R_k^\#} IV'_1, \dots, IV'_n \end{array} \right\}. \end{cases}$$

We have:

- $\mathbb{F}^\#$ is extensive;
- $\mathbb{F}^\#$ is monotonic;
- $\mathbb{F} \circ \gamma \subseteq \gamma \circ \mathbb{F}^\#$;
- $\mathbb{F}^\# \circ \alpha = \alpha \circ \mathbb{F} \circ \gamma \circ \alpha$ (we will see later why).

Soundness

Theorem 1 Let:

1. (D, \subseteq, \cup) and $(D^\#, \sqsubseteq, \sqcup)$ be chain-complete partial orders;
2. $D \xrightleftharpoons[\alpha]{\gamma} D^\#$ be a Galois connection;
3. $F \in D \rightarrow D$ and $F^\# \in D^\# \rightarrow D^\#$ be monotonic mappings such that:
 $F \circ \gamma \subseteq \gamma \circ F^\#$;
4. $X_0 \in D$ be an element such that: $X_0 \subseteq F(X_0)$;

Then:

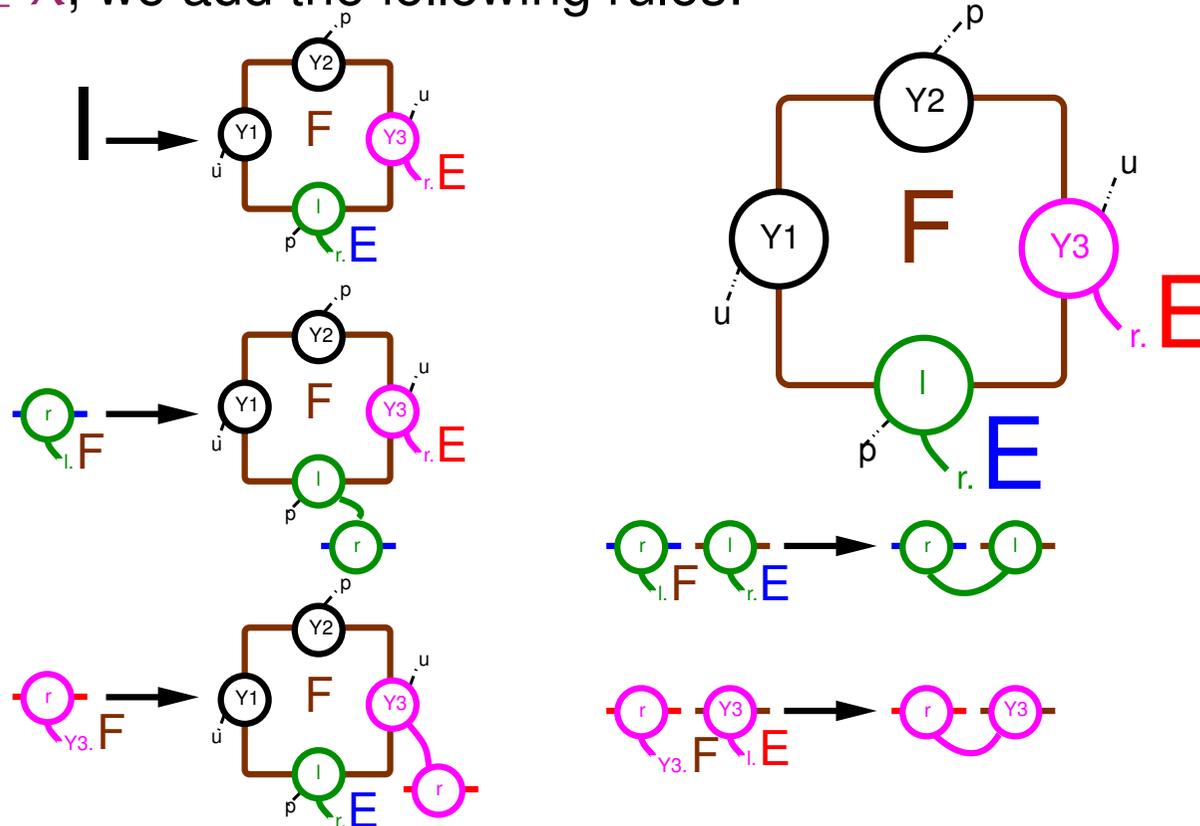
1. both $lfp_{X_0} F$ and $lfp_{\alpha(X_0)} F^\#$ exist,
2. $lfp_{X_0} F \subseteq \gamma(lfp_{\alpha(X_0)} F^\#)$.

Overview

1. Introduction
2. Language: Kappa
3. Abstraction: Local views
4. **Completeness: false positives?**
5. Local fragment of Kappa
6. Decontextualization
7. Conclusion

From views to species

For any $X \in \wp(\text{Local_view})$, $\gamma(X)$ is given by a rewrite system:
 For any $lv \in X$, we add the following rules:



l and semi-links are non-terminal.
 l is the initial symbol.

Pumping lemma

- We use this rewrite system to enumerate the chemical species of $\gamma(X)$.
- There are two cases:
 1. either there is a finite number of rewrite sequences;
 2. or we encounter cyclic derivations
i.e. an open chemical species with a cycle of the following form:

$R.l-r.E \dots R.l-r.E$

can be built.

- We only enumerate chemical species that are reached through an acyclic rewriting computation.
- It turns out that: if $X \in \alpha(\wp(\text{Species}))$ then each rewrite sequence is the prefix of a terminating rewrite sequence.
(So there is an unbounded number of species if, and only if, there is an unbounded number of rewrite sequences.)

Examples

1. Make the demo for egf
2. Make the demo for fgf
3. Make the demo for Global invariants

Counting chemical species

Given a set of local views X , we can easily count the number of species in $\gamma(X)$ by using the following lemmas:

Lemma 1 (rigidity) An embedding between two connected components is fully characterized by the image of one agent.

Lemma 2 (automorphism) If $\gamma(X)$ is finite, then for any $C \in \gamma(X)$:

- C has at most two automorphisms;
- if C has two automorphisms, then C has a bond of the form $R.r - r.R$. Moreover one automorphism swaps the two R of this bond.

Lemma 3 (Euler) If a chemical species has no cycle, then it has an agent with only one site.

sketch the algorithm

Which information is abstracted away ?

Our analysis is exact (no false positive):

- for EGF cascade (356 chemical species);
- for FGF cascade (79080 chemical species);
- for SBF cascade (around 10^{19} chemical species).

We know how to build systems with false positives...

...but they seem to be biologically meaningless.

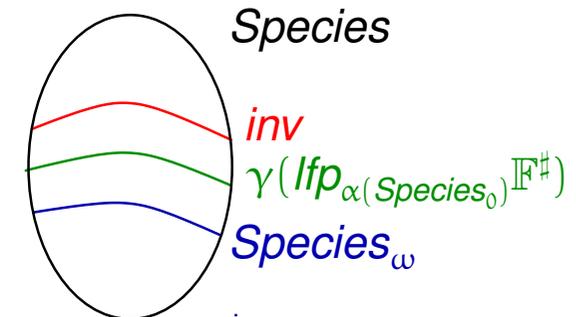
This raises the following issues:

- Can we characterize which information is abstracted away ?
- Which is the form of the systems, for which we have no false positive ?
- Do we learn something about the biological systems that we describe ?

Which information is abstracted away ?

Theorem 2 We suppose that:

1. (D, \subseteq) be a partial order;
2. $(D^\#, \subseteq, \sqcup)$ be chain-complete partial order;
3. $D \xrightleftharpoons[\alpha]{\gamma} D^\#$ be a Galois connection;
4. $F \in D \rightarrow D$ and $F^\# \in D^\# \rightarrow D^\#$ are monotonic;
5. $F \circ \gamma \subseteq \gamma \circ F^\#$;
6. $X_0, inv \in D$ such that:
 - $X_0 \subseteq F(X_0) \subseteq F(inv) \subseteq inv$,
 - $inv = \gamma(\alpha(inv))$,
 - and $\alpha(F(inv)) = F^\#(\alpha(inv))$;



Then, $lfp_{\alpha(X_0)}F^\#$ exists and $\gamma(lfp_{\alpha(X_0)}F^\#) \subseteq inv$.

Proof I/III

We have already seen (previous lectures) that:

1. $\text{lfp}_{\alpha(X_0)} \mathbb{F}^\#$ exists;
2. there exists an ordinal δ such that $\text{lfp}_{\alpha(X_0)} \mathbb{F}^\# = \mathbb{F}^{\#\delta}(\alpha(X_0))$.

Proof II/III

Let us show that $\gamma(\text{lfp}_{\alpha(X_0)} \mathbb{F}^\#) \subseteq \text{inv}$.

Let us prove instead by induction over δ that $\mathbb{F}^{\#\delta}(\alpha(X_0)) \subseteq \alpha(\text{inv})$.

- If $Y \in D^\#$ is an element such that $Y \subseteq \alpha(\text{inv})$,
 $\mathbb{F}^\#(Y) \subseteq \mathbb{F}^\#(\alpha(\text{inv}))$ ($\mathbb{F}^\#$ is mon)
 $\mathbb{F}^\#(\alpha(\text{inv})) = \alpha(\mathbb{F}(\text{inv}))$ (assumption)
 $\alpha(\mathbb{F}(\text{inv})) \subseteq \alpha(\text{inv})$. (α is mon and inv is a post)

Thus: $\mathbb{F}^\#(Y) \subseteq \alpha(\text{inv})$

- If $Y_i \in D^{\#I}$ is a chain of elements such that $Y_i \subseteq \alpha(\text{inv})$ for any $i \in I$,
then, $\sqcup Y_i \subseteq \alpha(\text{inv})$ (lub).

So: $\mathbb{F}^{\#\delta}(\alpha(X_0)) \subseteq \alpha(\text{inv})$.

Proof II/III

We have:

$$\mathbb{F}^{\#\delta}(\alpha(X_0)) \subseteq \alpha(inv).$$

Since γ is monotonic:

$$\gamma(\mathbb{F}^{\#\delta}(\alpha(X_0))) \subseteq \gamma(\alpha(inv)).$$

But, by assumption, $\gamma(\alpha(inv)) = inv$.

Thus,

$$\gamma(\mathbb{F}^{\#\delta}(\alpha(X_0))) \subseteq inv.$$

When is there no false positive ?

Theorem 3 We suppose that:

1. (D, \subseteq, \cup) and $(D^\#, \sqsubseteq, \sqcup)$ are chain-complete partial orders;
2. $(D, \subseteq) \xrightleftharpoons[\alpha]{\gamma} (D^\#, \sqsubseteq)$ is a Galois connection;
3. $F : D \rightarrow D$ is a monotonic map;
4. X_0 is a concrete element such that $X_0 \subseteq F(X_0)$;
5. $F \circ \gamma \subseteq \gamma \circ F^\#$;
6. $F^\# \circ \alpha = \alpha \circ F \circ \gamma \circ \alpha$.

Then:

- $lfp_{X_0} F$ and $lfp_{\alpha(X_0)} F^\#$ exist;
- $lfp_{X_0} F = \gamma(\alpha(lfp_{X_0} F)) \iff lfp_{X_0} F = \gamma(lfp_{\alpha(X_0)} F^\#)$.

We need to understand under which assumptions $lfp_{X_0} F = \gamma(\alpha(lfp_{X_0} F))$.

Local set of chemical species

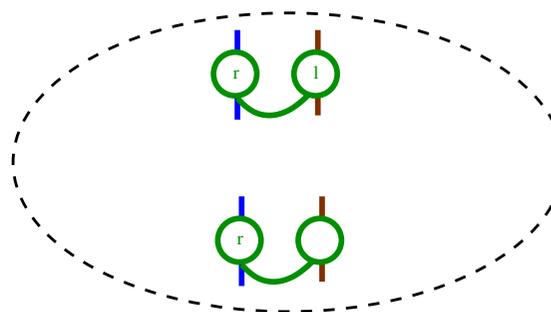
Definition 1 We say that a set $X \in \wp(\textit{Species})$ of chemical species is local if and only if $X \in \gamma(\wp(\textit{Local_view}))$.

(ie. a set X is local if and only if X is exactly the set of all the species that are generated by a given set of local views.)

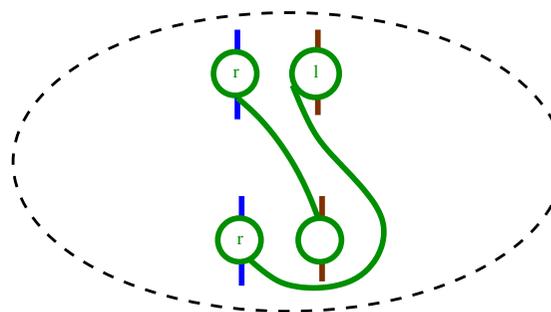
Swapping relation

We define the binary relation \sim^{SWAP} among tuples *Species** of chemical species.
We say that $(C_1, \dots, C_m) \sim^{\text{SWAP}} (D_1, \dots, D_n)$ if and only if:

(C_1, \dots, C_m) matches with



while (D_1, \dots, D_n) matches with



Swapping closure

Theorem 4 Let $X \in \wp(\textit{Species})$ be a set of chemical species.

The two following assertions are equivalent:

1. $X = \gamma(\alpha(X))$;
2. for any tuples $(C_i), (D_j) \in \textit{Species}^*$ such that:
 - $(C_i) \in X^*$,
 - and $(C_i) \stackrel{\text{SWAP}}{\sim} (D_j)$;we have $(D_j) \in X^*$.

Proof (easier implication way)

If:

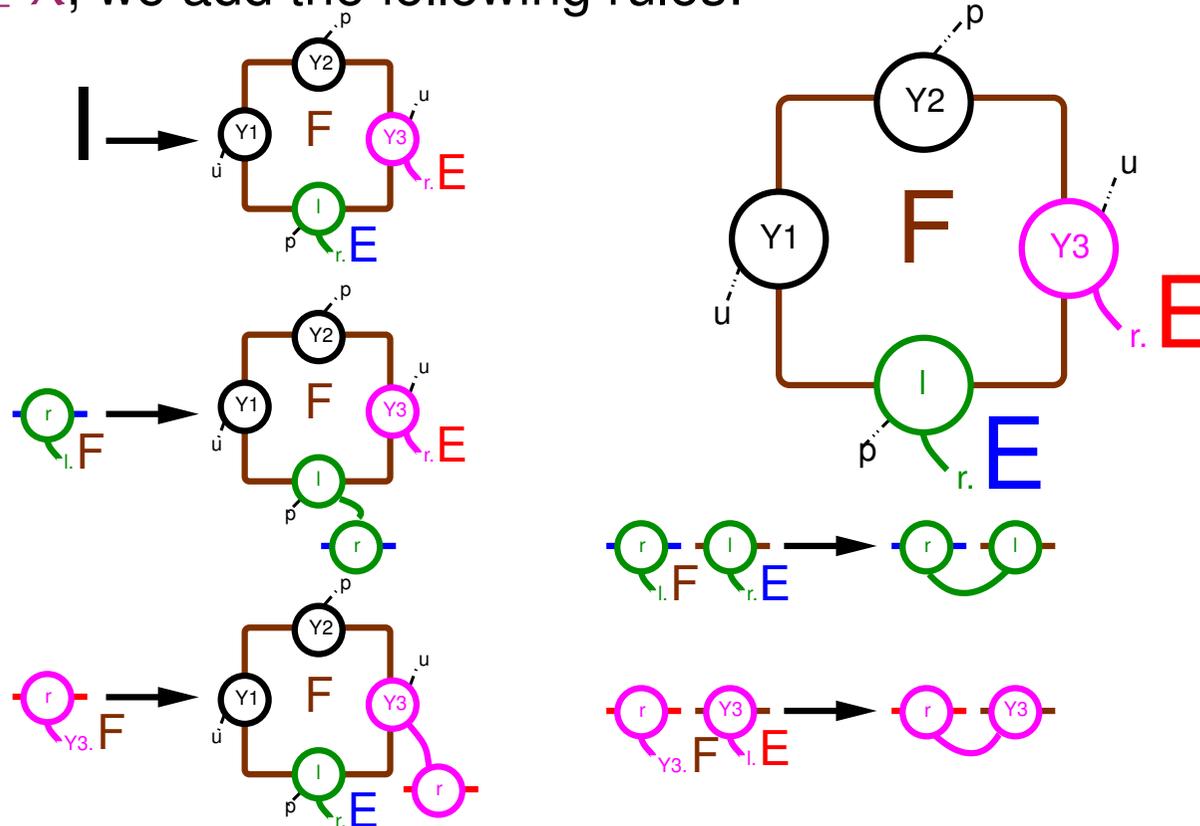
- $X = \gamma(\alpha(X))$,
- $(C_i)_{i \in I} \in X^*$,
- and $(C_i)_{i \in I} \stackrel{\text{SWAP}}{\sim} (D_j)_{j \in J}$;

Then:

we have $\alpha(\{C_i \mid i \in I\}) = \alpha(\{D_j \mid j \in J\})$ (because $(C_i) \stackrel{\text{SWAP}}{\sim} (D_j)$)
and $\alpha(\{C_i \mid i \in I\}) \subseteq \alpha(X)$ (because $(C_i) \in X^*$ and α mon);
so $\alpha(\{D_j \mid j \in J\}) \subseteq \alpha(X)$;
so $\{D_j \mid j \in J\} \subseteq \gamma(\alpha(X))$ (by def. of Galois connections);
so $\{D_j \mid j \in J\} \subseteq X$ (since $X = \gamma(\alpha(X))$);
so $(D_j)_{j \in J} \in X^*$.

Proof: more difficult implication way

For any $X \in \wp(\text{Local_view})$, $\gamma(X)$ is given by a rewrite system:
 For any $lv \in X$, we add the following rules:



I and semi-links are non-terminal.
 I is the initial symbol.

Proof (more difficult implication way)

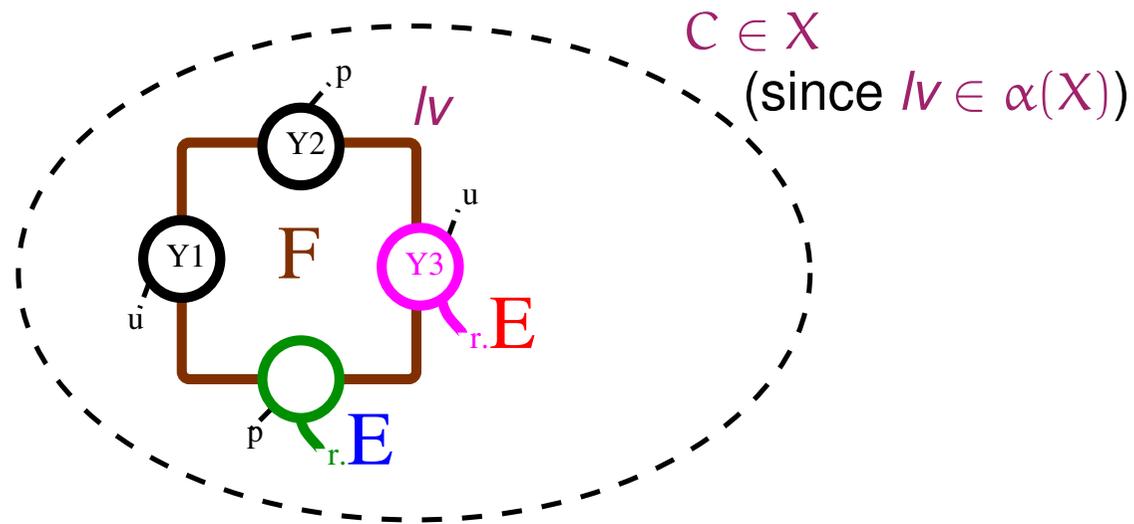
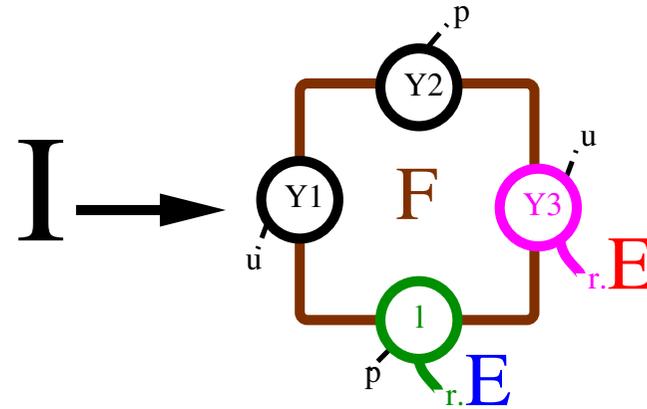
We suppose that X is close with respect to $\overset{\text{SWAP}}{\sim}$.

We want to prove that $\gamma(\alpha(X)) \subseteq X$.

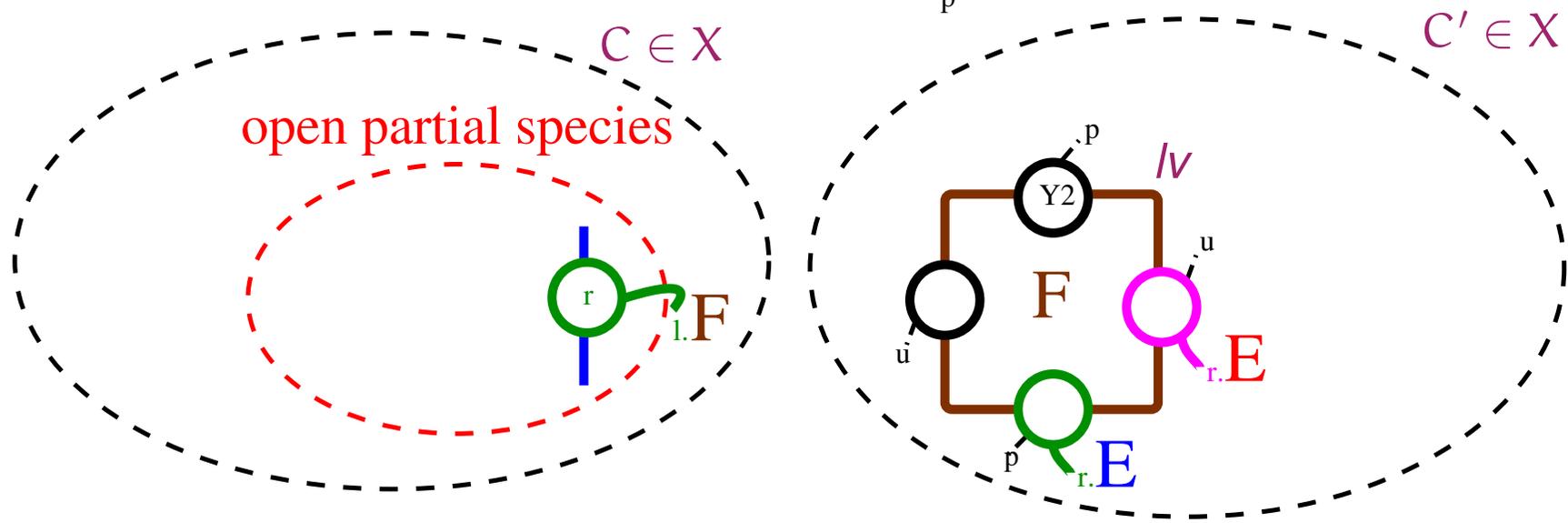
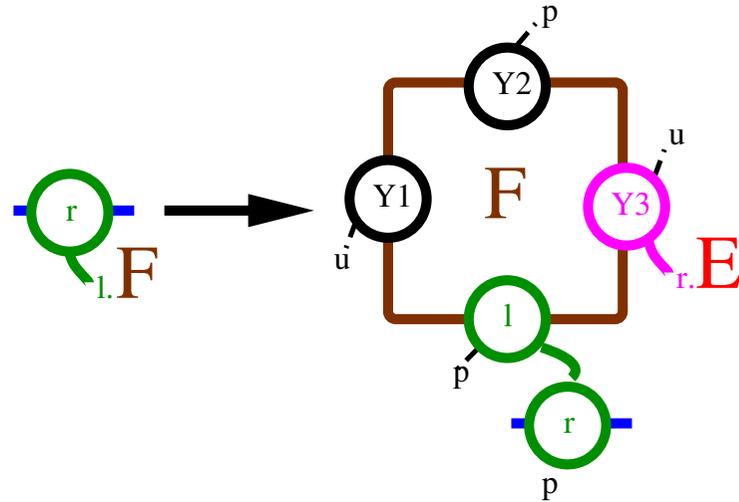
We prove, by induction, that any open complex that can be built by gathering the views of $\alpha(X)$, can be embedded in a complex in X :

- By def. of α , this is satisfied for any local view in $\alpha(X)$;
- This remains satisfied after unfolding a semi-link with a local view;
- This remains satisfied after binding two semi-links.

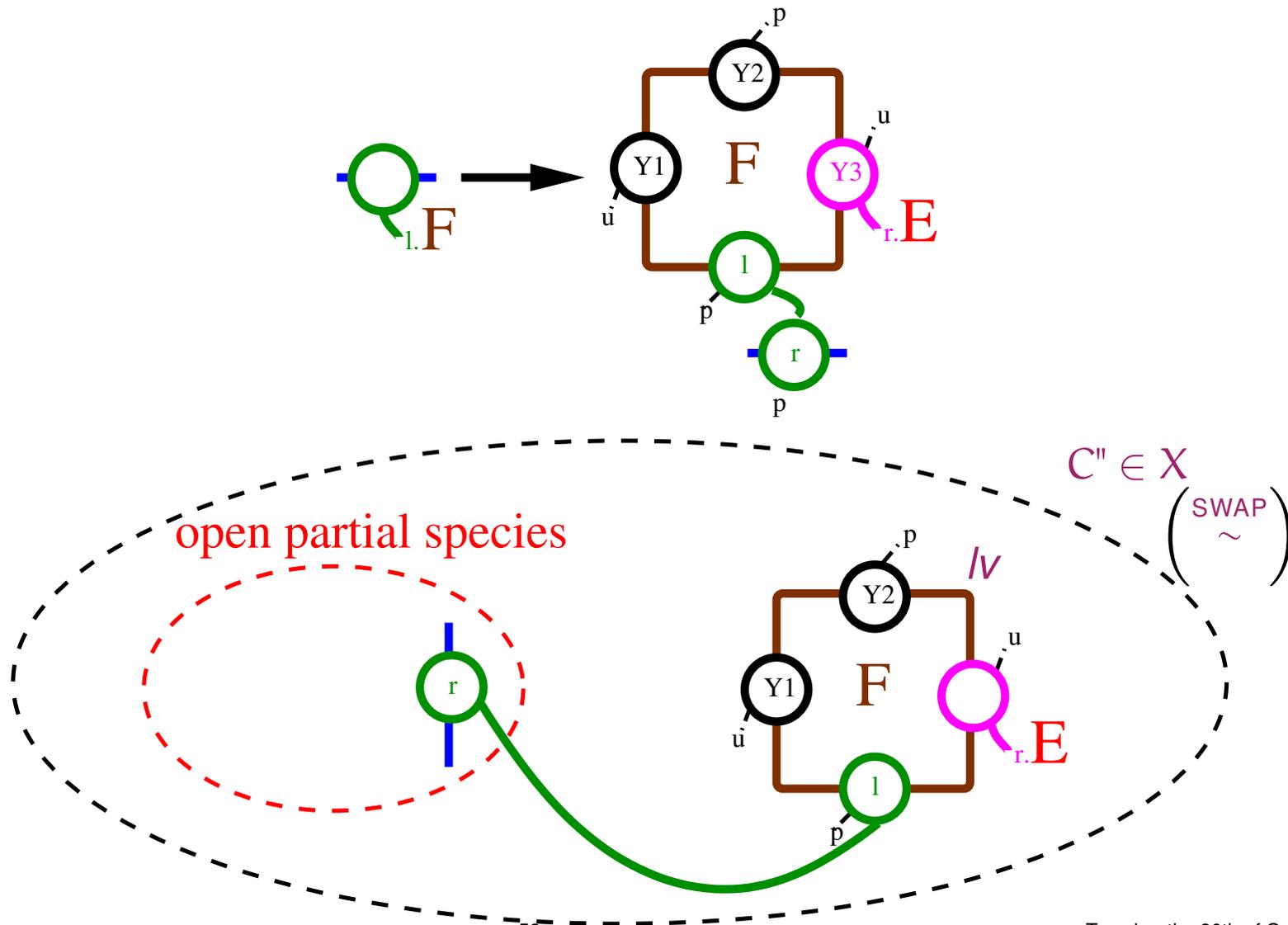
Initialization



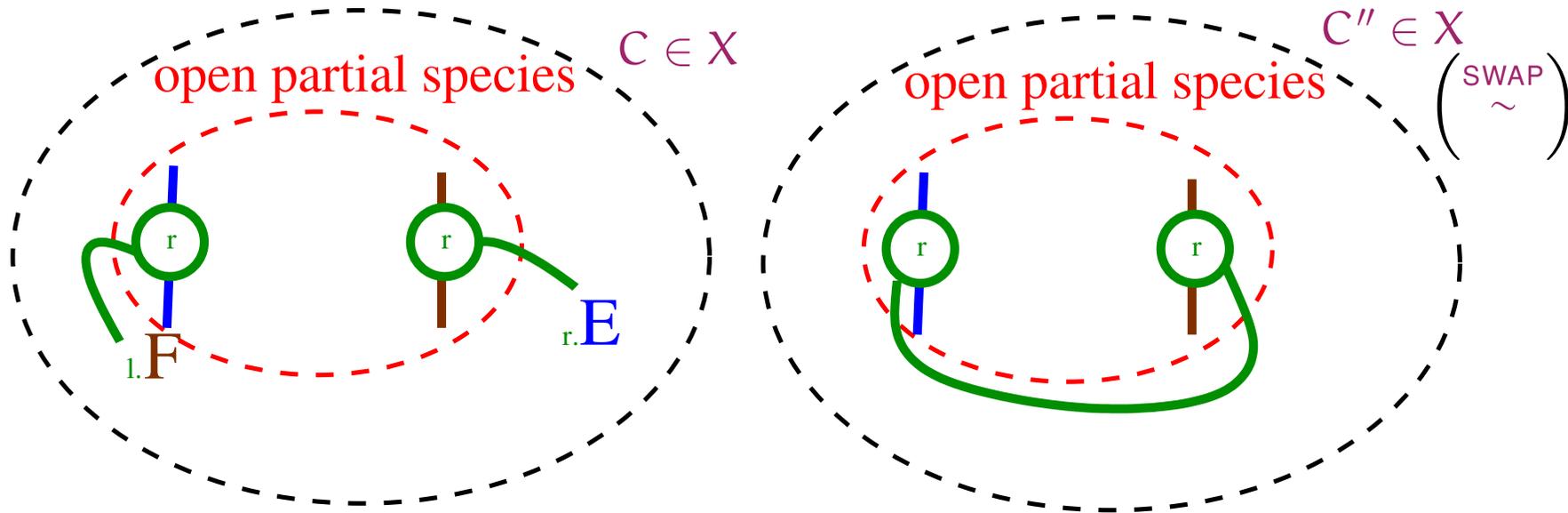
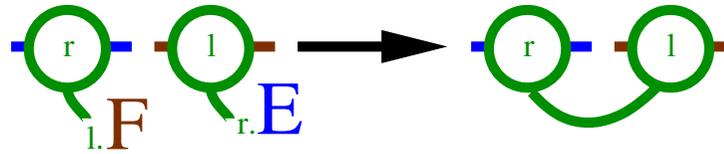
Unfolding a semi-link



Unfolding a semi-link



Binding two semi-links



Consequences

Let $Y \in \wp(\text{Local_view})$ be a set of local views such that $\alpha(\gamma(Y)) = Y$.

1. Each open complex C built with the local views in Y is a sub-complex of a close complex C' in $\gamma(Y)$.
2. When considering the rewrite system that computes $\gamma(Y)$, **any partial rewriting sequence can be completed in a successful one.**

Thus:

- (a) $\gamma(Y)$ is finite if and only if the grammar has a finite set of prefixes (and the latter is decidable);
- (b) We have $\mathbb{F}^\# \circ \alpha = \alpha \circ \mathbb{F} \circ \gamma \circ \alpha$.

Overview

1. Introduction
2. Language: Kappa
3. Abstraction: Local views
4. Completeness: false positives?
5. **Local fragment of Kappa**
6. Decontextualization
7. Conclusion

Outline

We have proved that:

- if the set $Species_\omega$ of reachable chemical species is close with respect to swapping $\overset{SWAP}{\sim}$,
swapping $\overset{SWAP}{\sim}$,
- then the reachability analysis is exact (i.e. $Species_\omega = \gamma(lfp_{\alpha(Species_0)} \mathbb{F}^\#)$).

Now we give some sufficient conditions that ensure this property.

Sufficient conditions

Whenever the following assumptions:

1. initial agents are not bound;
2. rules are atomic;
3. rules are local:
 - only agents that interact are tested,
 - no cyclic patterns (neither in lhs, nor in rhs);
4. binding rules do not interfere i.e. if both:
 - $A(a_{\sim m}, S), B(b_{\sim n}, T) \rightarrow A(a_{\sim m}!1, S), B(b_{\sim n}!1, T)$
 - and $A(a_{\sim m}', S'), B(b_{\sim n}', T') \rightarrow A(a_{\sim m}'!1, S'), B(b_{\sim n}'!1, T')$,

then:

- $A(a_{\sim m}, S), B(b_{\sim n}', T') \rightarrow A(a_{\sim m}!1, S), B(b_{\sim n}'!1, T')$;

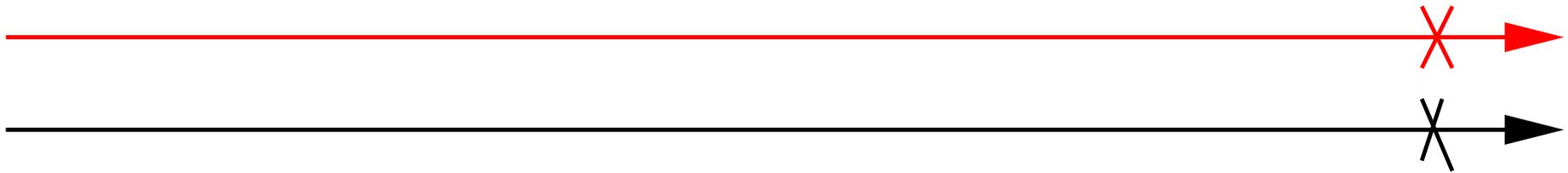
5. chemical species in $\gamma(\alpha(\textit{Species}_\omega))$ are acyclic,
are satisfied, the set of reachable chemical species is local.

Proof outline

We sketch a proof in order to discover sufficient conditions that ensure this property:

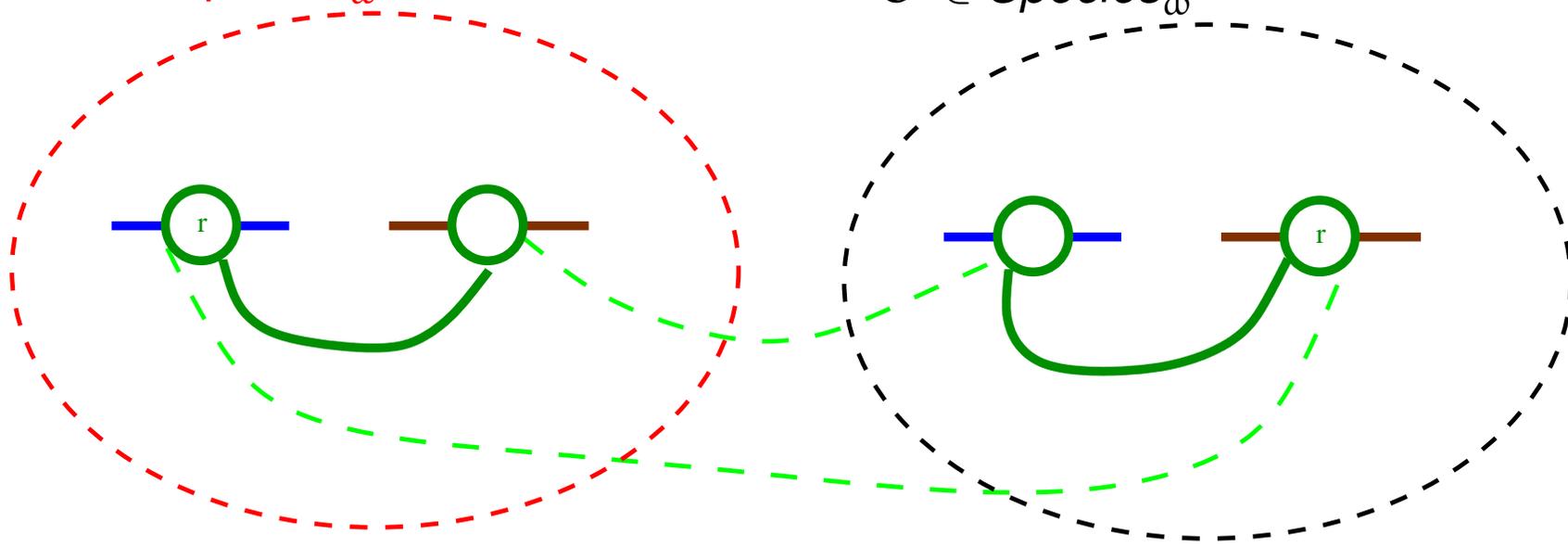
- We consider tuples of complexes in which the same kind of links occur twice.
- We want to swap these links.
- We introduce the history of their computation.
- There are several cases. . .

First case (I/V)



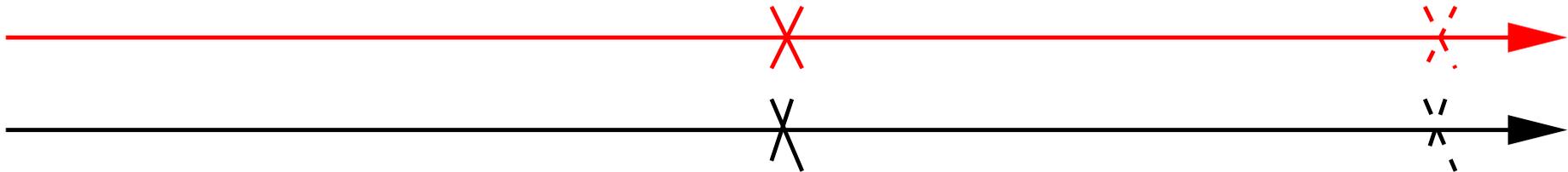
$C \in \text{Species}_\omega$

$C' \in \text{Species}_\omega$

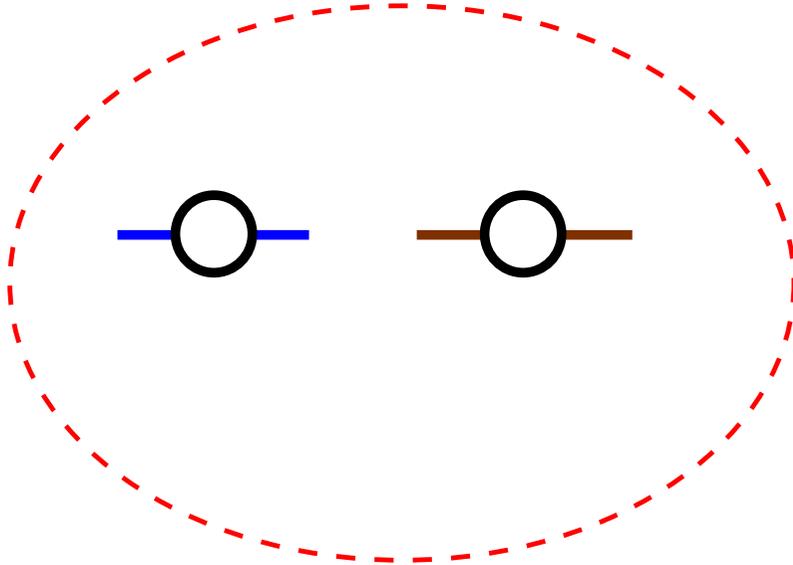


First case (II/V)

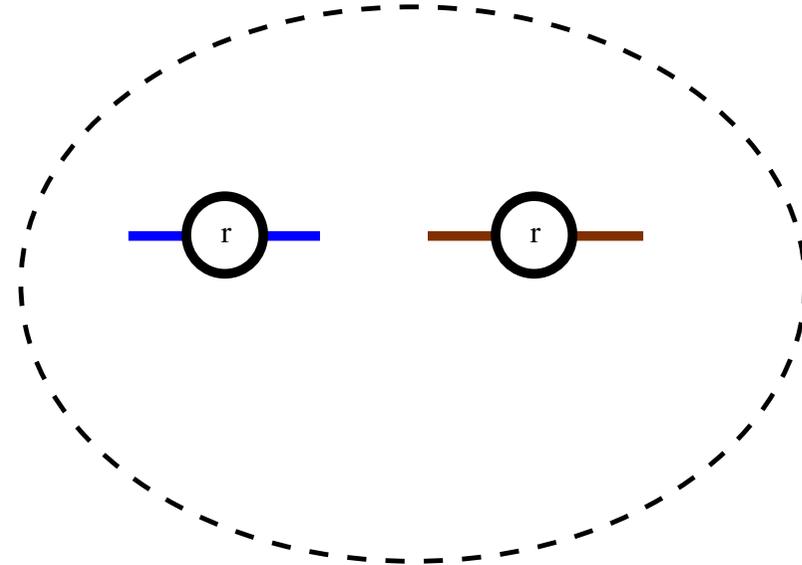
just before the links are made



$C \in \text{Species}_\omega^*$

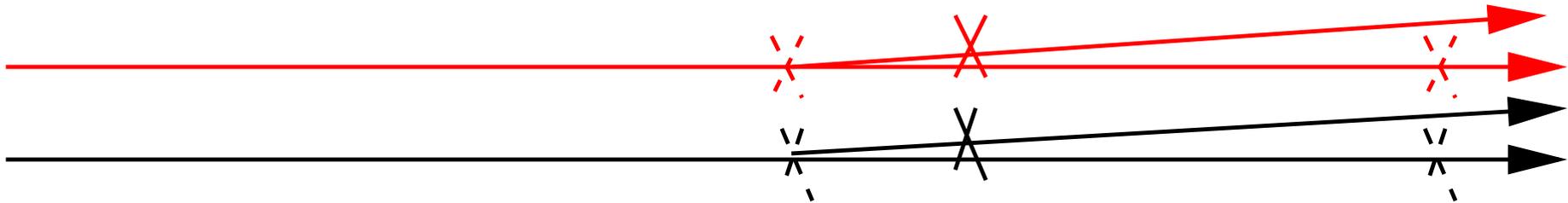


$C' \in \text{Species}_\omega^*$

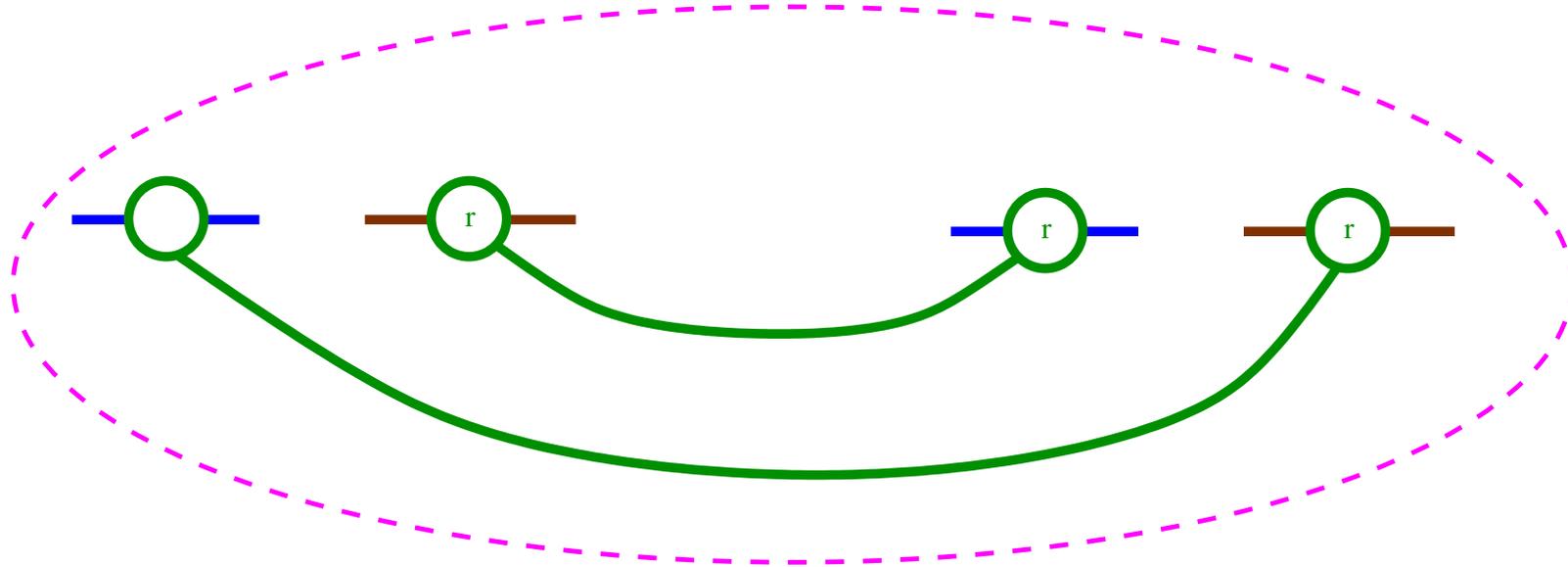


First case (III/V)

we suppose we can swap the links



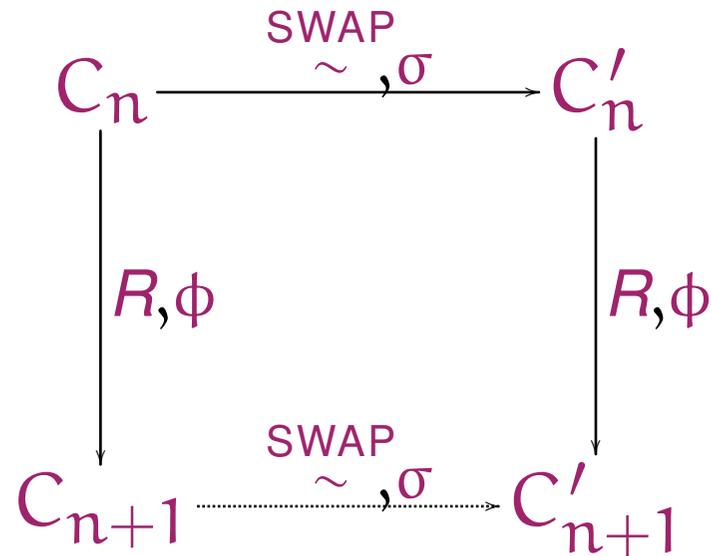
$C \in \text{Species}_\omega^*$



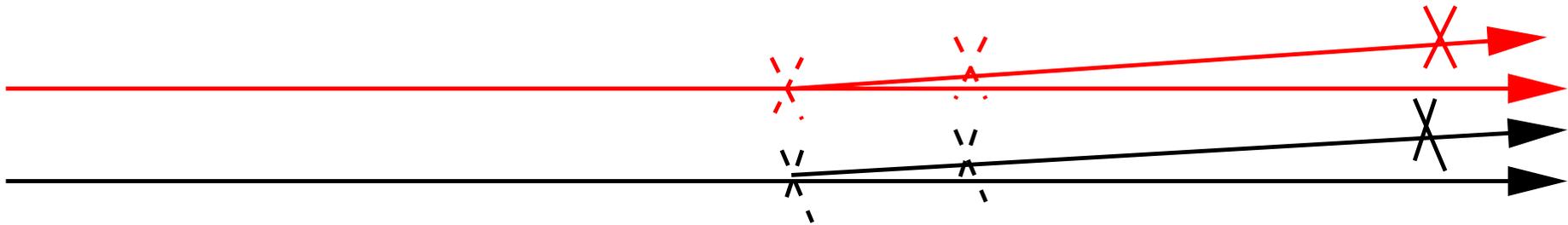
First case (IV/V)

Then, we ensure that further computation steps:

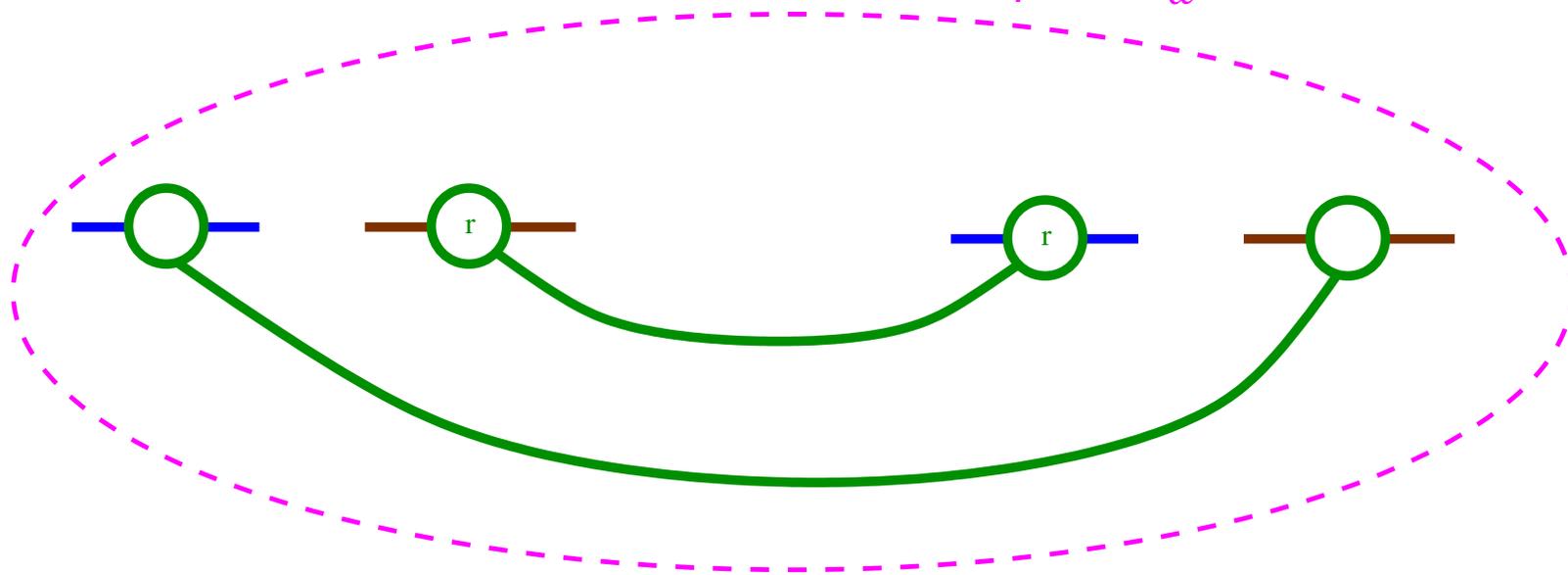
- are always possible;
- have the same effect on local views;
- commute with the swapping relation $\overset{\text{SWAP}}{\sim}$.



First case (V/V)

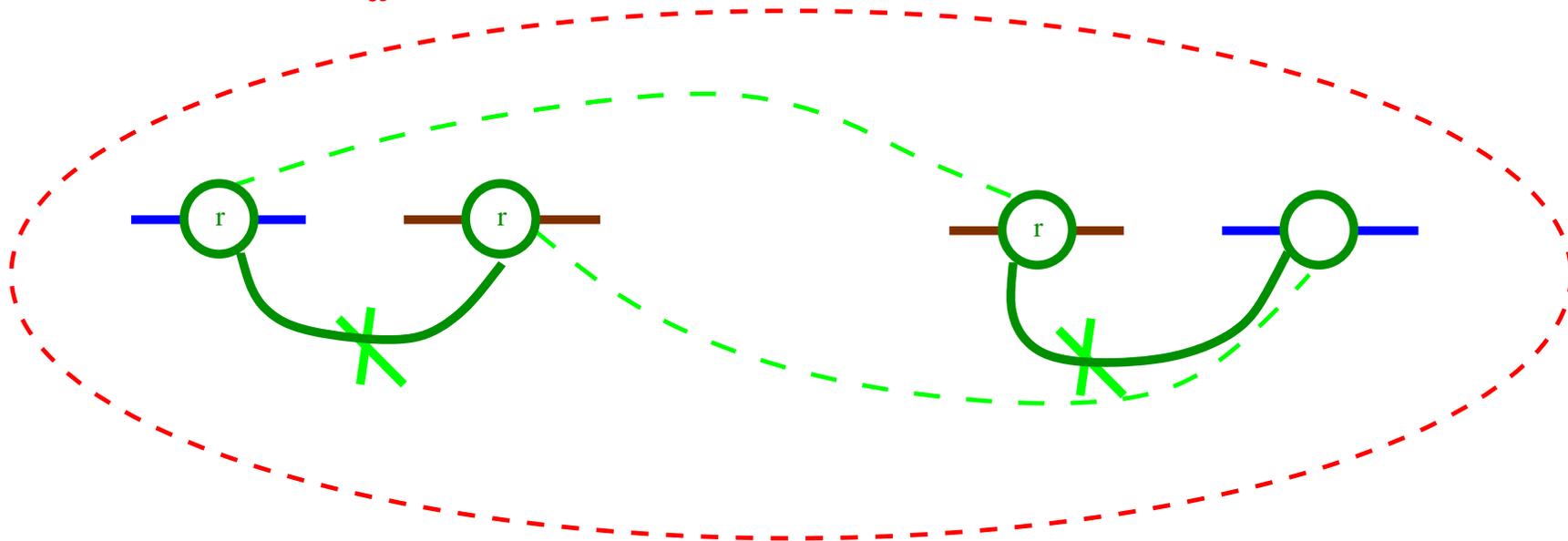


$C \in \text{Species}_\omega^*$



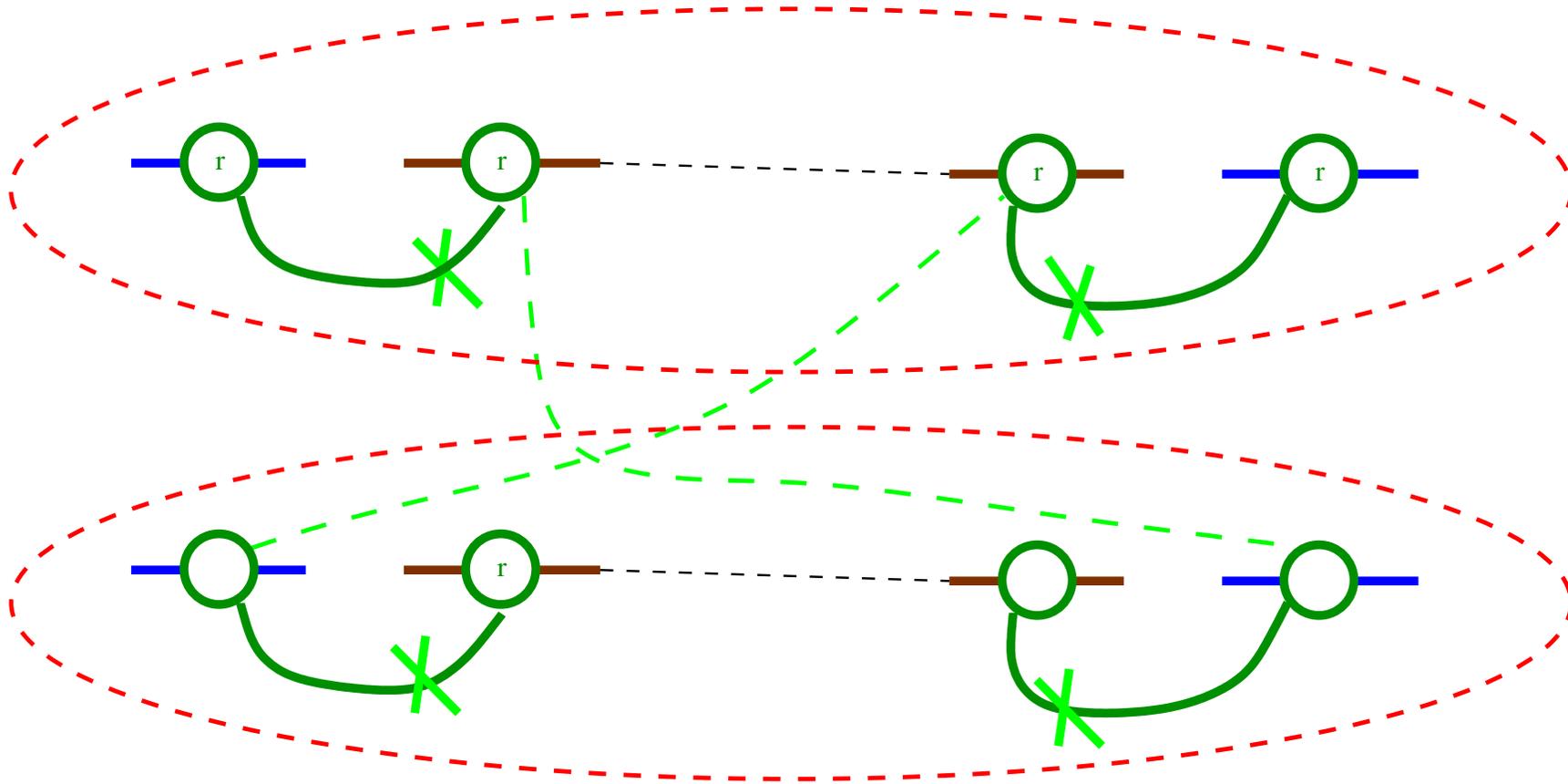
Second case (I/II)

$C \in \text{Species}_\omega$



we assume that the chemical species C is acyclic

Second case (II/II)



Sufficient conditions

Whenever the following assumptions:

1. initial agents are not bound;
2. rules are atomic;
3. rules are local:
 - only agents that interact are tested,
 - no cyclic patterns (neither in lhs, nor in rhs);
4. binding rules do not interfere i.e. if both:
 - $A(a_{\sim m}, S), B(b_{\sim n}, T) \rightarrow A(a_{\sim m}!1, S), B(b_{\sim n}!1, T)$
 - and $A(a_{\sim m}', S'), B(b_{\sim n}', T') \rightarrow A(a_{\sim m}'!1, S'), B(b_{\sim n}'!1, T')$,

then:

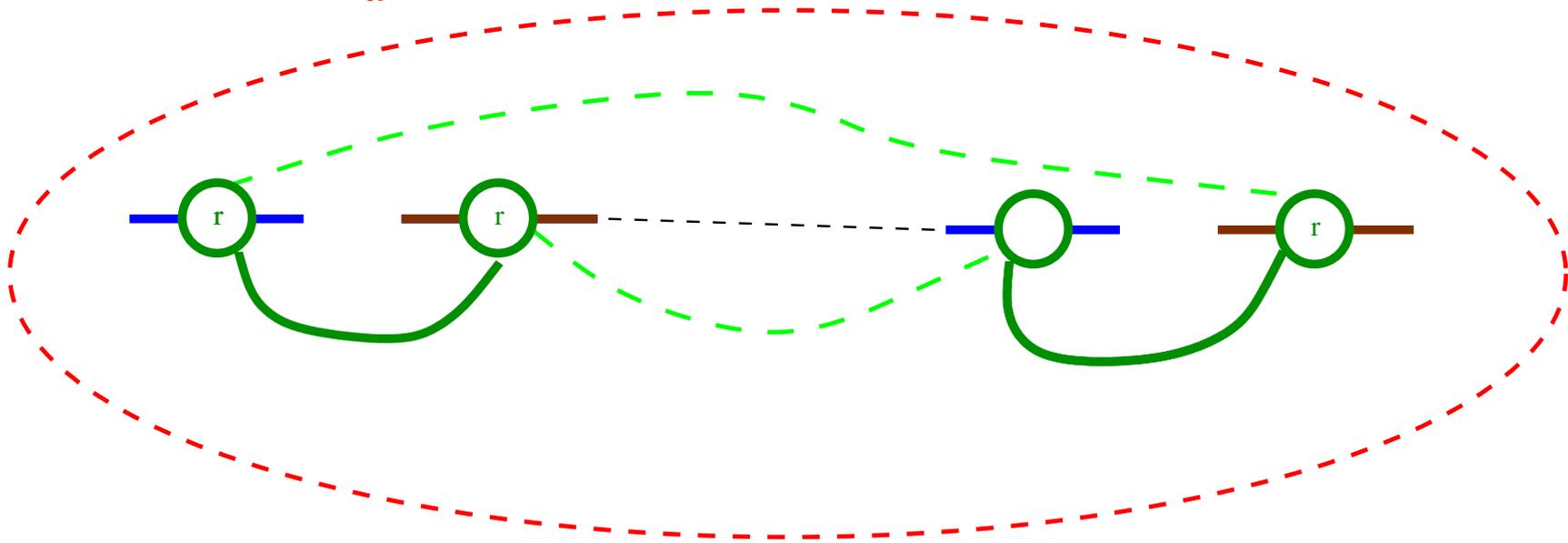
- $A(a_{\sim m}, S), B(b_{\sim n}', T') \rightarrow A(a_{\sim m}!1, S), B(b_{\sim n}'!1, T')$;

5. chemical species in $\gamma(\alpha(\textit{Species}_\omega))$ are acyclic,
are satisfied, the set of reachable chemical species is local.

Third case (I/III)



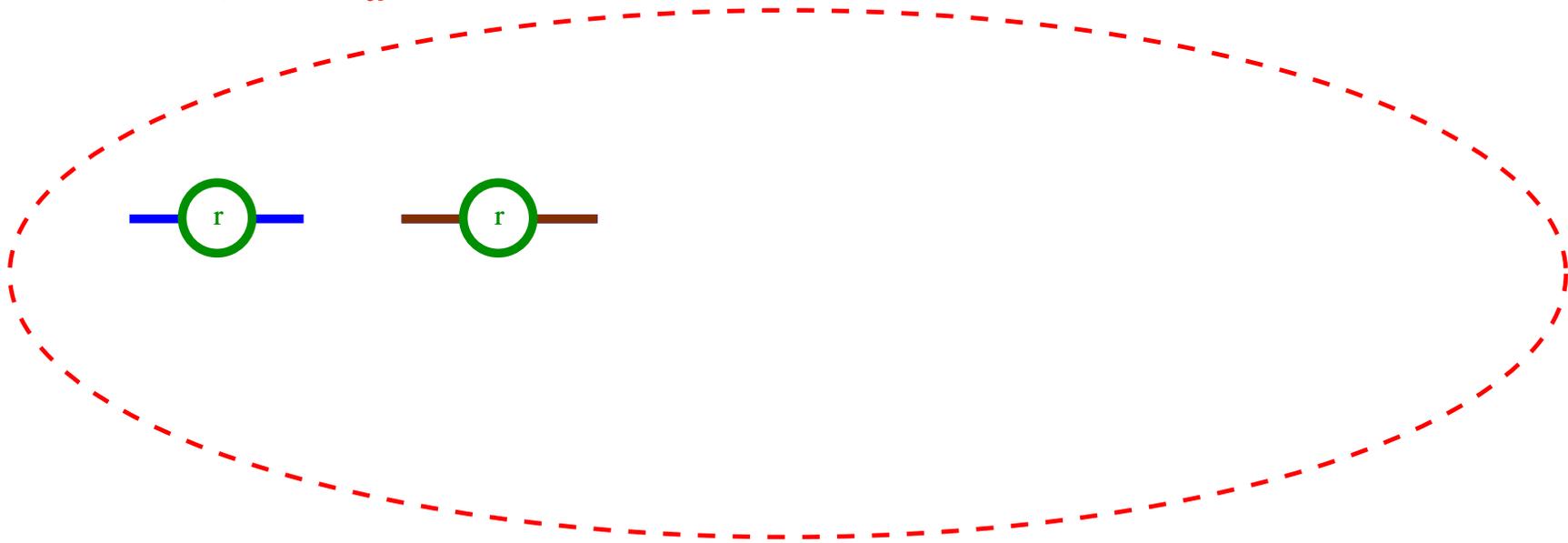
$C \in \text{Species}_\omega$



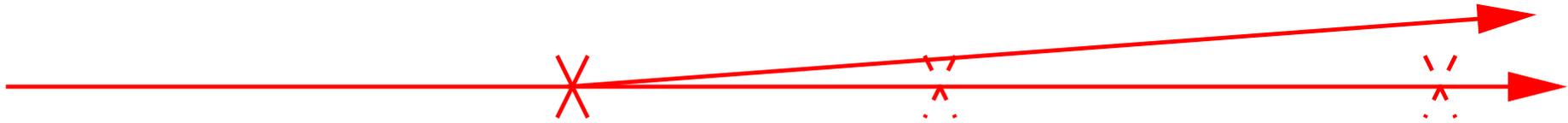
Third case (II/III)



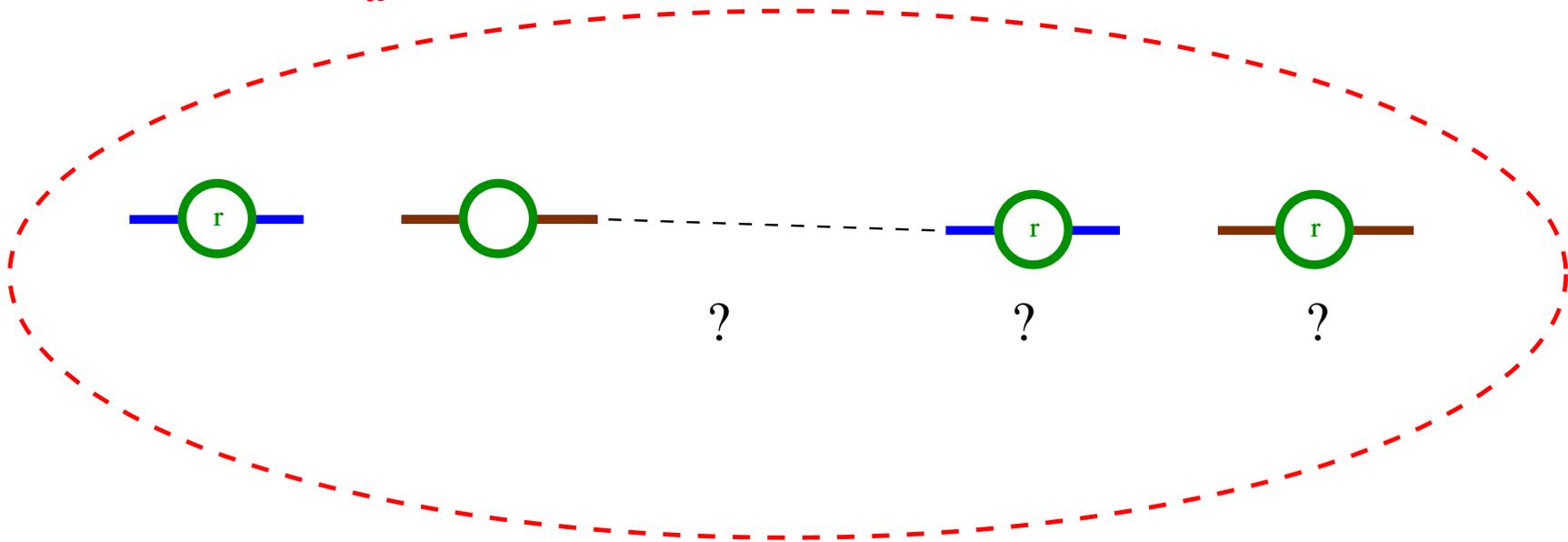
$C \in \text{Species}_\omega^*$



Third case (II/III)



$C \in \text{Species}_\omega^*$



Dangerous sites

A site is dangerous if it may occur in a cycle within a complex ($\in \gamma(\alpha(\textit{Species}_\omega))$).

We would weaken the fifth requirement into:

- The binding state of a dangerous site is never tested, unless for binding or unbinding this site.
- When we bind dangerous sites, we only test that these sites are free.

Then, we prove that:

1. we can build any complex with free dangerous sites,
2. then, we can bind them as much as we like.

Non local systems

$$\begin{array}{l}
 \textit{Species}_0 \triangleq R(a \sim u) \\
 \textit{Rules} \triangleq \left\{ \begin{array}{l}
 R(a \sim u) \leftrightarrow R(a \sim p) \\
 R(a \sim u), R(a \sim u) \rightarrow R(a \sim u!1), R(a \sim u!1) \\
 R(a \sim p), R(a \sim u) \rightarrow R(a \sim p!1), R(a \sim p!1) \\
 R(a \sim p), R(a \sim p) \rightarrow R(a \sim p!1), R(a \sim p!1)
 \end{array} \right\}
 \end{array}$$

$R(a \sim u!1), R(a \sim u!1) \in \textit{Species}_\omega$

$R(a \sim p!1), R(a \sim p!1) \in \textit{Species}_\omega$

But $R(a \sim u!1), R(a \sim p!1) \notin \textit{Species}_\omega$.

Non local systems

$$\begin{array}{l}
 \textit{Species}_0 \triangleq A(a \sim u), B(a \sim u) \\
 \textit{Rules} \triangleq \left\{ \begin{array}{l}
 A(a \sim u), B(a \sim u) \rightarrow A(a \sim u!1), B(a \sim u!1) \\
 A(a \sim u!1), B(a \sim u!1) \rightarrow A(a \sim p!1), B(a \sim u!1) \\
 A(a \sim u!1), B(a \sim u!1) \rightarrow A(a \sim u!1), B(a \sim p!1)
 \end{array} \right\}
 \end{array}$$

$A(a \sim u!1), B(a \sim p!1) \in \textit{Species}_\omega$

$A(a \sim p!1), B(a \sim u!1) \in \textit{Species}_\omega$

But $A(a \sim p!1), B(a \sim p!1) \notin \textit{Species}_\omega$.

Non local systems

$$\begin{array}{l} \textit{Species}_0 \triangleq A(a \sim u) \\ \textit{Rules} \triangleq \left\{ \begin{array}{l} A(a \sim u) \leftrightarrow A(a \sim p) \\ A(a \sim u), A(a \sim p) \rightarrow A(a \sim u!1), A(a \sim p!1) \end{array} \right\} \end{array}$$

$A(a \sim u!1), A(a \sim p!1) \in \textit{Species}_\omega$
But $A(a \sim p!1), A(a \sim p!1) \notin \textit{Species}_\omega$.

Non local systems

*Species*₀ \triangleq R(a,b)

Rules \triangleq { R(a,b), R(a) \rightarrow R(a,b!1), R(a!1) }

R(a,b!2), R(a!2,b!1), R(a!1,b) \in *Species* _{ω}

But R(a!1,b!1) \notin *Species* _{ω} .

Overview

1. Introduction
2. Language: Kappa
3. Abstraction: Local views
4. Completeness: false positives?
5. Local fragment of Kappa
6. Decontextualization
7. Conclusion

Outline

- we have a syntactic criterion in order to ensure that the set of reachable chemical species of a kappa system is local ;
- we now design program transformations to help systems satisfying this criterion ;
 1. **decontextualization**
 - is fully automatic;
 - preserves the transition system;
 - simplifies rules thanks to reachability analysis.
 2. **conjugation**
 - manual;
 - preserves the set of reachable chemical species;
 - uses backtrack to add new rules.

Example

Initial rule:

$$R2(I!2,r),R1(I!1,r),E2(r!1),E1(r!2) \rightarrow R2(I!3,r!1),R1(I!2,r!1),E2(r!2),E1(r!3)$$

Decontextualized rule:

$$R2(I!_,r),R1(I!_,r) \rightarrow R2(I!_,r!1),R1(I!_,r!1)$$

We can remove redundant tests.

Example

Initial rules:

$$\begin{aligned} \text{Sh}(Y7 \sim p!2, pi!1), G(a!2, b), R(Y48 \sim p!1) &\rightarrow \text{Sh}(Y7 \sim p, pi!1), G(a, b), R(Y48 \sim p!1) \\ \text{Sh}(Y7 \sim p!3, pi!1), G(a!3, b!2), \text{So}(d!2), R(Y48 \sim p!1) &\rightarrow \text{Sh}(Y7 \sim p, pi!1), G(a, b!2), \text{So}(d!2), R(Y48 \sim p!1) \\ \text{Sh}(Y7 \sim p!1, pi), G(a!1, b) &\rightarrow \text{Sh}(Y7 \sim p, pi), G(a, b) \\ \text{Sh}(Y7 \sim p!1, pi), G(a!1, b!_) &\rightarrow \text{Sh}(Y7 \sim p, pi), G(a, b!_) \end{aligned}$$

Decontextualized rule:

$$\text{Sh}(Y7!1), G(a!1) \rightarrow \text{Sh}(Y7), G(a)$$

We can remove exhaustive enumerations.

How does it work ?

To remove a test, we prove that:

- this test is satisfied whenever the other tests are satisfied;
- or each complex that passes all tests but this one also matches with the left hand side of another rule that performs the same action.

More formally

More formally:

- Each rule R is associated with the set $S(R)$ of open chemical species that can match its lhs;
- Rules are gathered in equivalence classes according to the actions they perform;
- For each class $[R]$, we compute:

$$\mathcal{G}([R]) = \cup\{S(R') \mid R' \in [R]\}.$$

- For each class $[R]$, $Reach([R])$ is an over approximation of the set of open chemical species that may match the lhs of a rule $R' \in [R]$.

A rule R may be decontextualized in a rule R' if:

$$S(R') \cap Reach([R]) \subseteq \mathcal{G}([R]).$$

Decontextualization is more efficient, if the reachability analysis is accurate.

An undecontextualizable rule

Initial rule:

$$\text{Sh}(Y7 \sim u, \text{pi}!1), \text{R}(Y48 \sim p!1, r!_) \rightarrow \text{Sh}(Y7 \sim p, \text{pi}!1), \text{R}(Y48 \sim p!1, r!_)$$

Decontextualized rule:

$$\text{Sh}(Y7 \sim u, \text{pi}!1), \text{R}(Y48!1, r!_) \rightarrow \text{Sh}(Y7 \sim p, \text{pi}!1), \text{R}(Y48!1, r!_)$$

Conjugation

If a rule R' is equivalent to a rule in the transitive closure of the system.

Then it may be included in the system without modifying reachable states.

To remove the context C of a rule, we try to apply it for another context C' by:

1. removing the context C' (backtrack) ;
2. building the context C ;
3. applying the initial rule ;
4. removing the context C (backtrack) ;
5. building the context C' .

This is proved manually.

Overview

1. Introduction
2. Language: Kappa
3. Abstraction: Local views
4. Completeness: false positives?
5. Local fragment of Kappa
6. Decontextualization
7. **Conclusion**

Conclusion

- A scalable static analysis to abstract the reachable chemical species.
- A class of models for which the abstraction is complete.
- Many applications:
 - idiomatic description of reachable chemical species;
 - dead rule detection;
 - rule decontextualization;
 - computer-driven kinetic refinement.
- It can also help simulation algorithms:
 - wake up/inhibition map (agent-based simulation);
 - flat rule system generation (for bounded set of chemical species);
 - on the fly flat rule generation (for large/unbounded set)