# MPRI

# Abstract interpretation of protein-protein interactions networks

## Jérôme Feret

DI - ÉNS

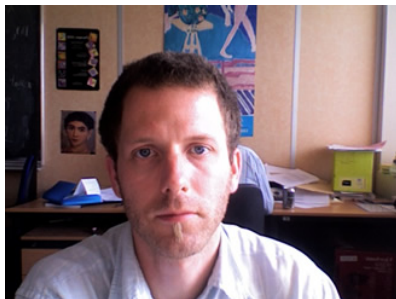Wednesday, the 4th of November, 2015

# Joint-work with...



Walter Fontana
Harvard Medical School

Vincent Danos
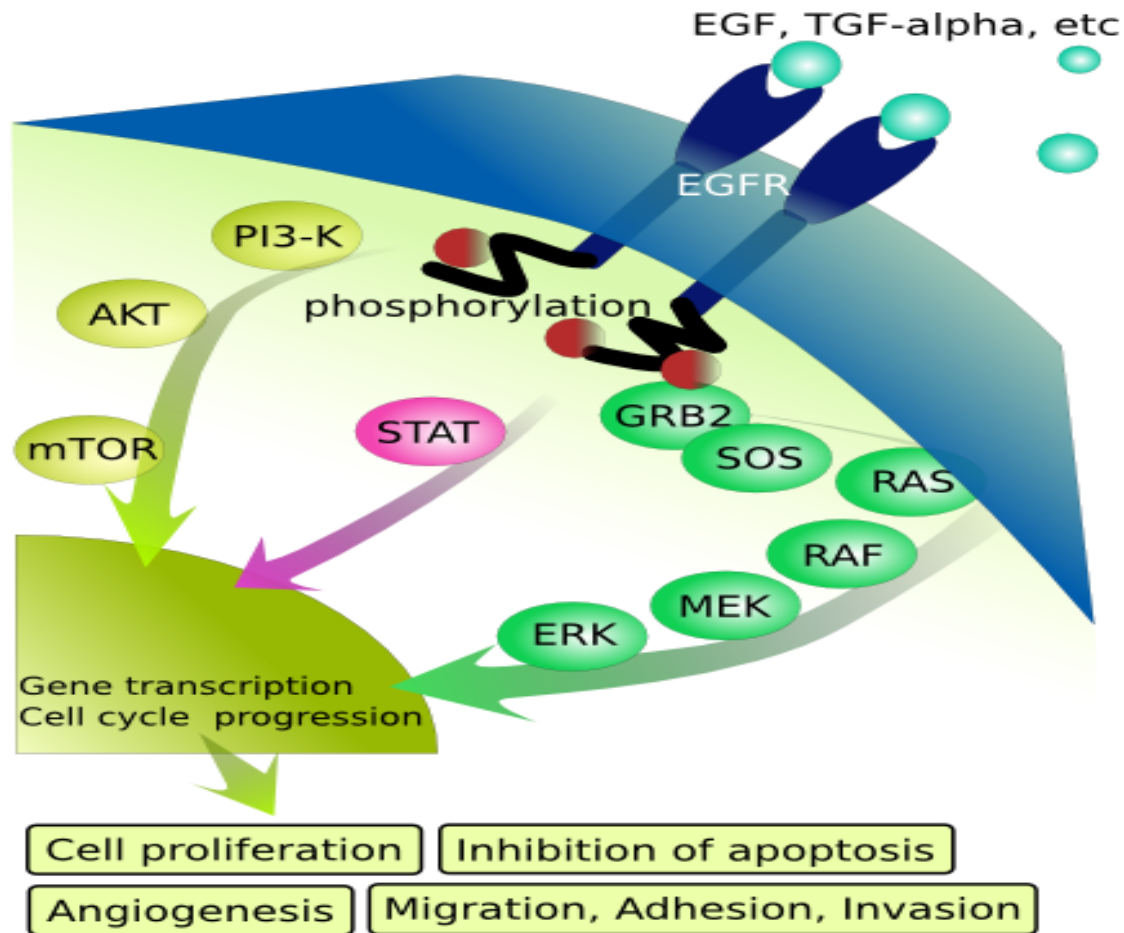ÉNS

Ferdinanda Camporesi
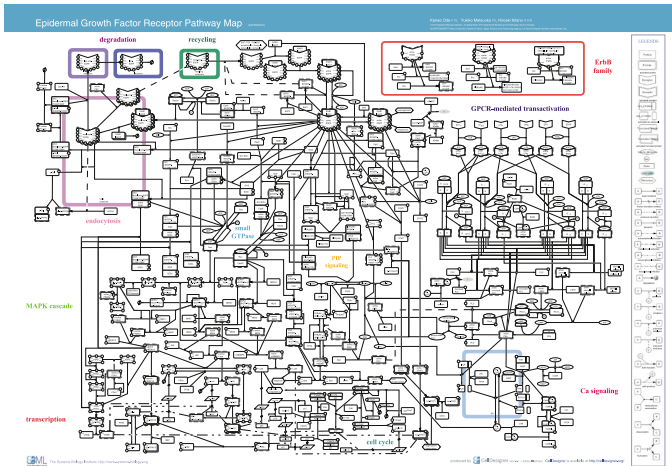Bologna / ÉNS

Russ Harmer
ÉNS Lyon

Jean Krivine
Paris VII

# Signalling Pathways



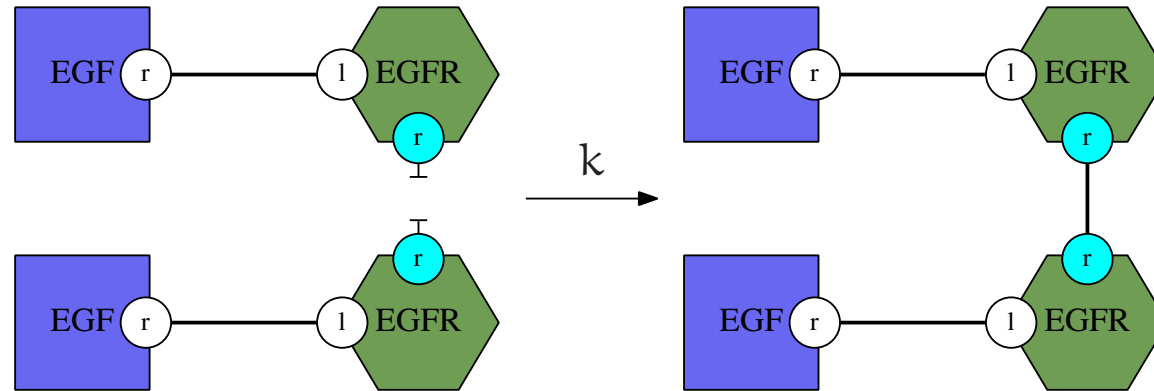Eikuch, 2007

# Bridging the gap between...



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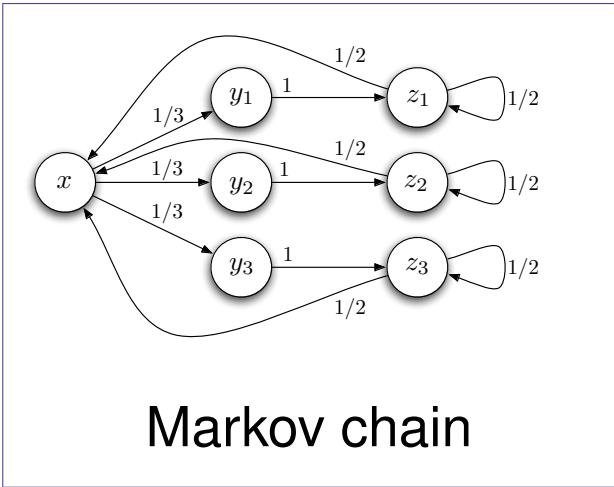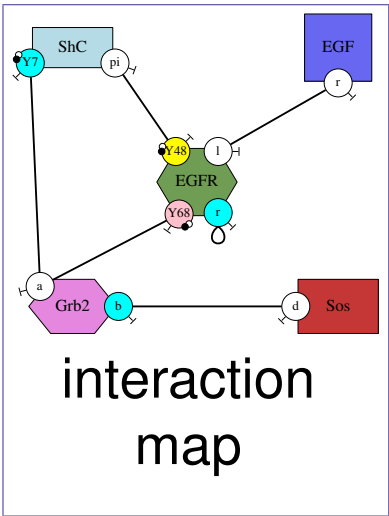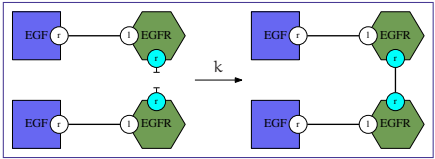$$

## 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

ordinary differential equations

# Complexity walls
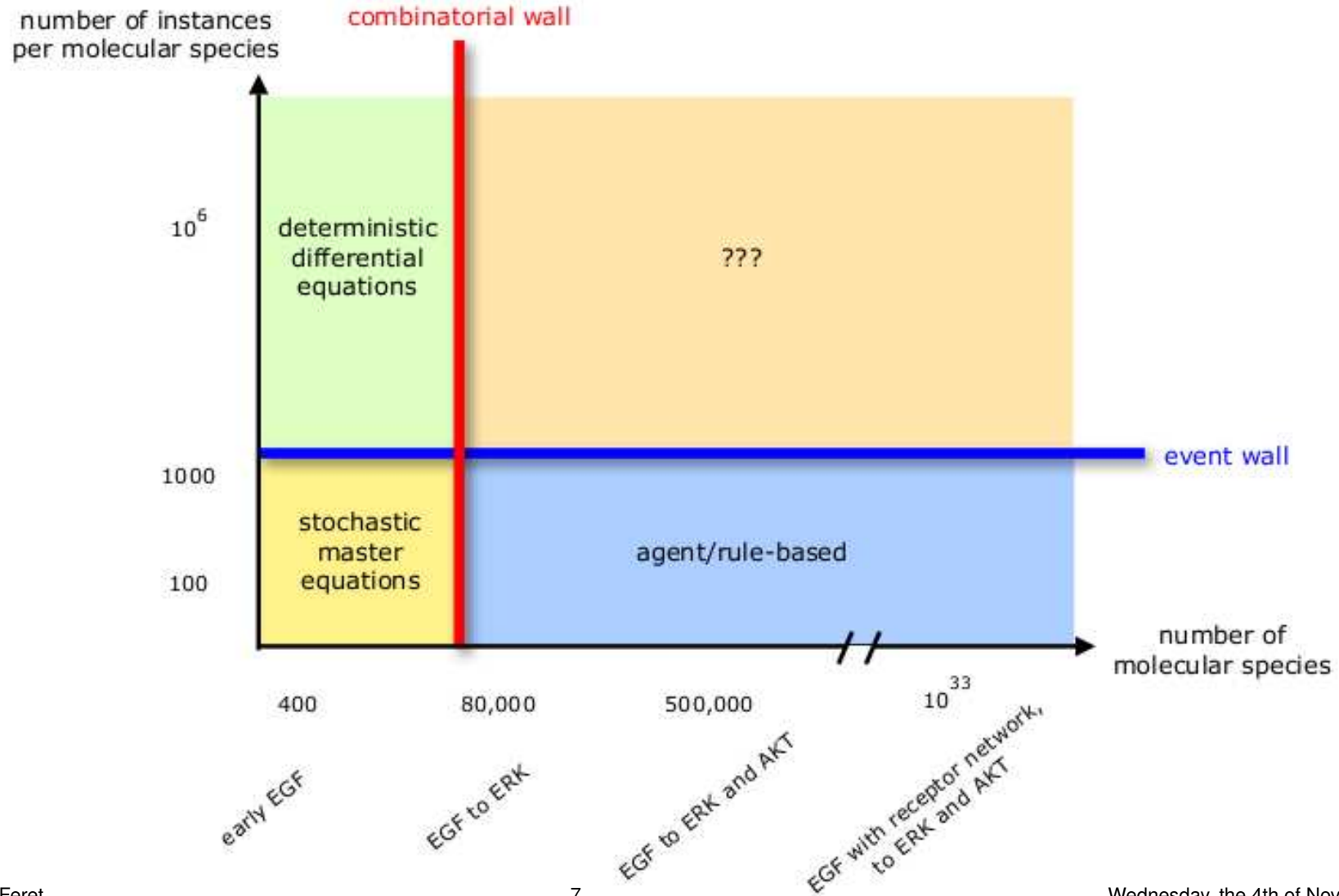
# Abstractions offer different perspectives on models
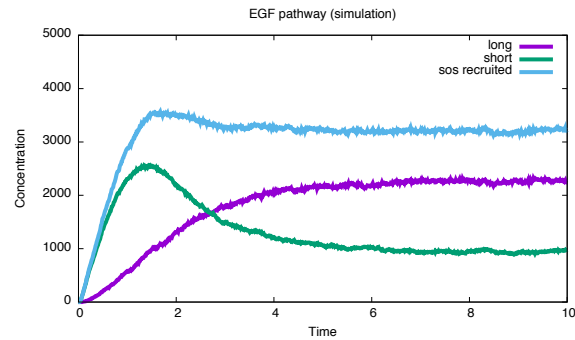


concrete semantics
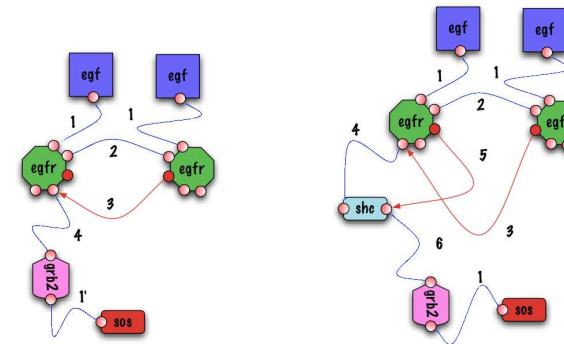


causal traces
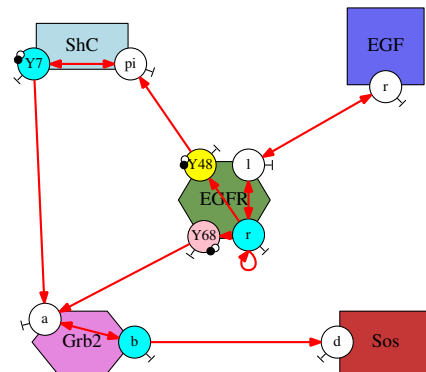


information flow
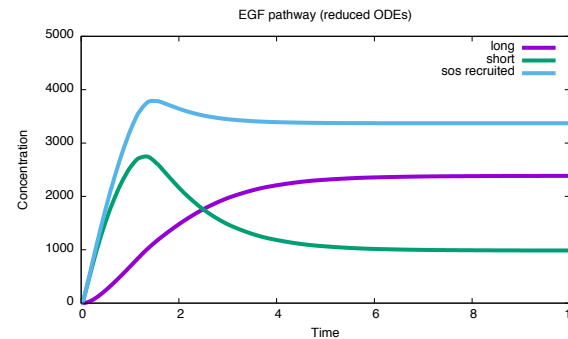


exact projection
of the ODE semantics

# Static analysis of reachable species (I/II)

We capture the relationships between the states of the sites of each agent.
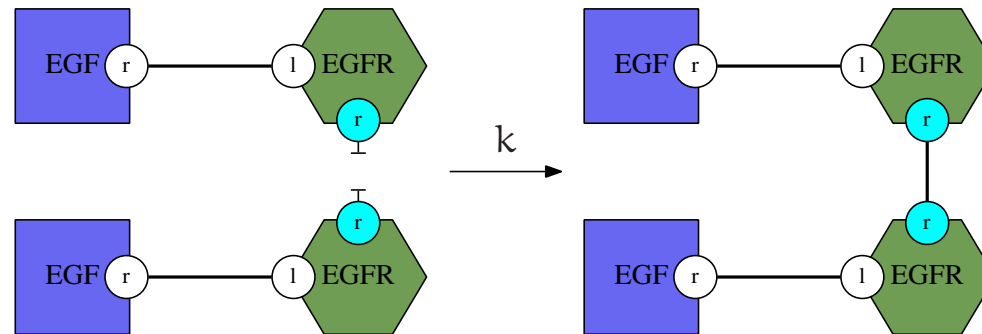
# Static analysis of reachable species (I/II)

We capture the relationships between the states of the sites of each agent.
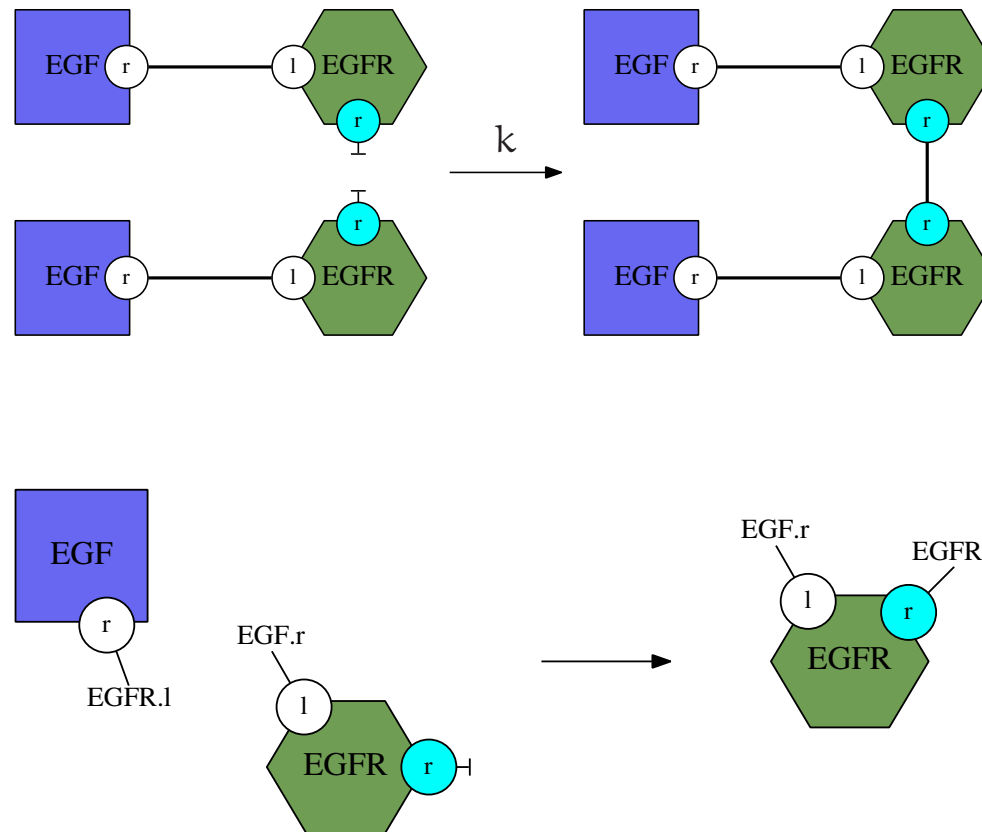
# Static analysis of reachable species (II/II)

Applications:

1. check the consistency of a model [ICCMSE'07]

2. compute the properties to allow fast simulation [APLAS'07]

3. simplify models,

4. compute independent fragments of chemical species [PNAS'09, LICS'10,Chaos'10]

The analysis is complete (no false positif) for a significatif kernel of Kappa [VMCAI'08].

# Model reduction

The ground differential system uses one variable per chemical species;
We directly compute its exact projection over independent fragments of chemical species.
With a small model, 356 chemical species are reduced into 38 fragments:
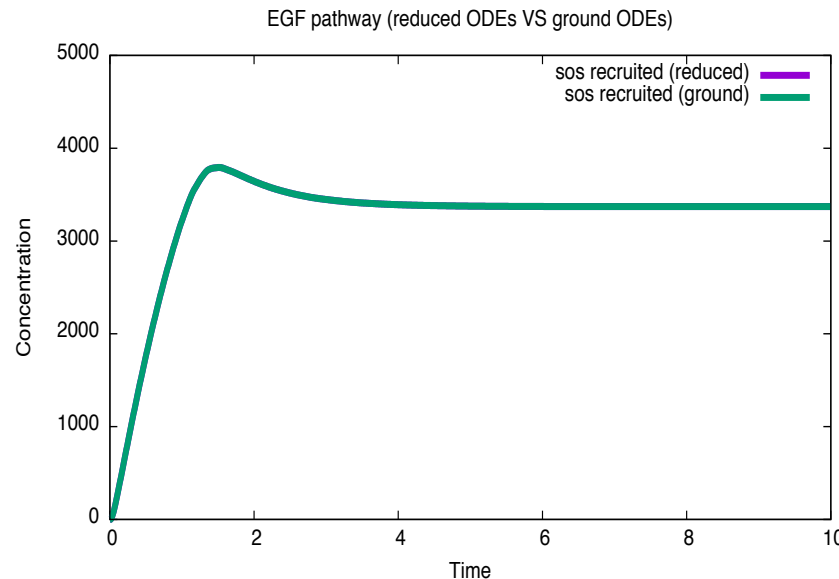


On a bigger model, $10^{19}$ chemical species are reduced into 180 000 fragments. [PNAS'09,LICS'10,Chaos'10]

MPRI

# Reachability Analysis of Rule-based Models

**Jérôme Feret**

DI - ÉNS

# 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
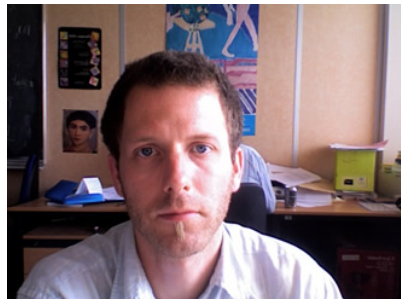


Russ Harmer
ÉNS Lyon



Jean Krivine
Paris VII

# **Overview**

# Signaling Pathways
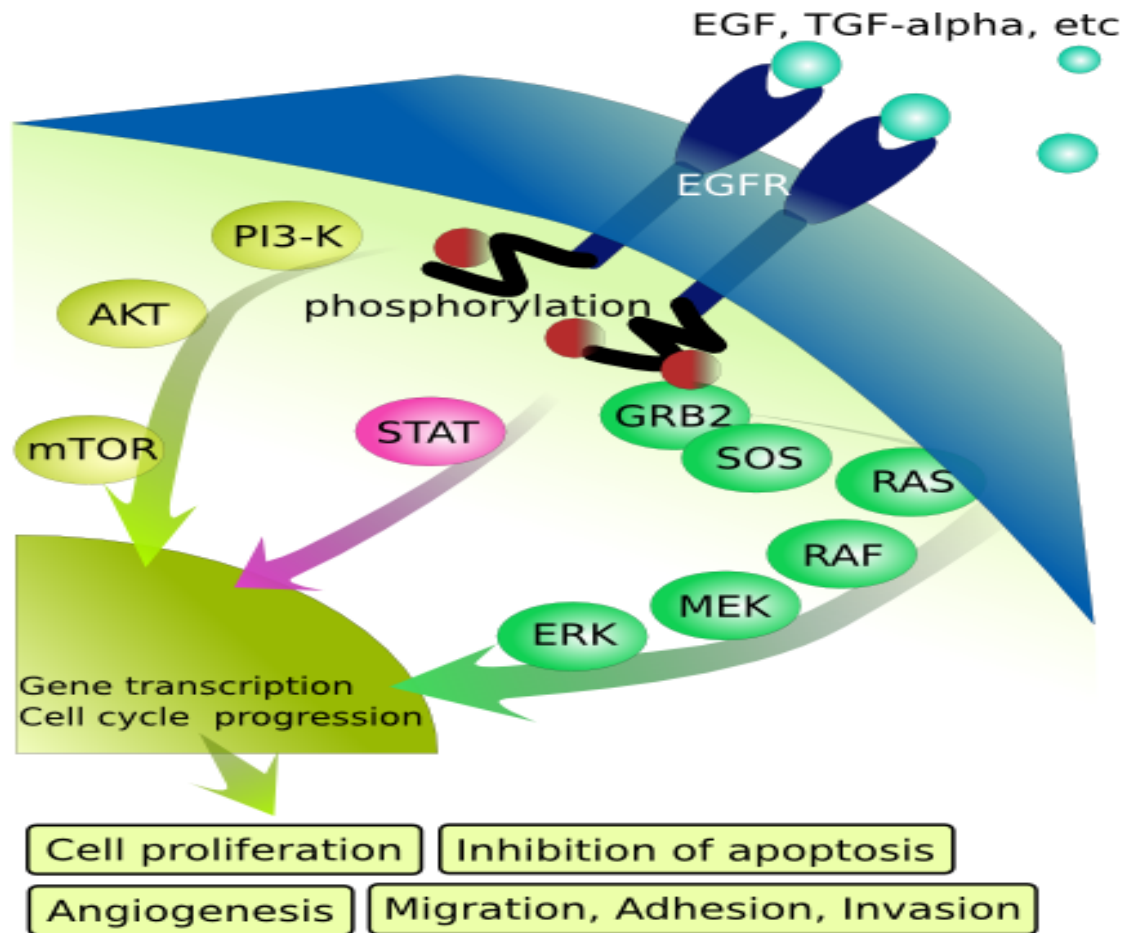


Eikuch, 2007

# Contact map
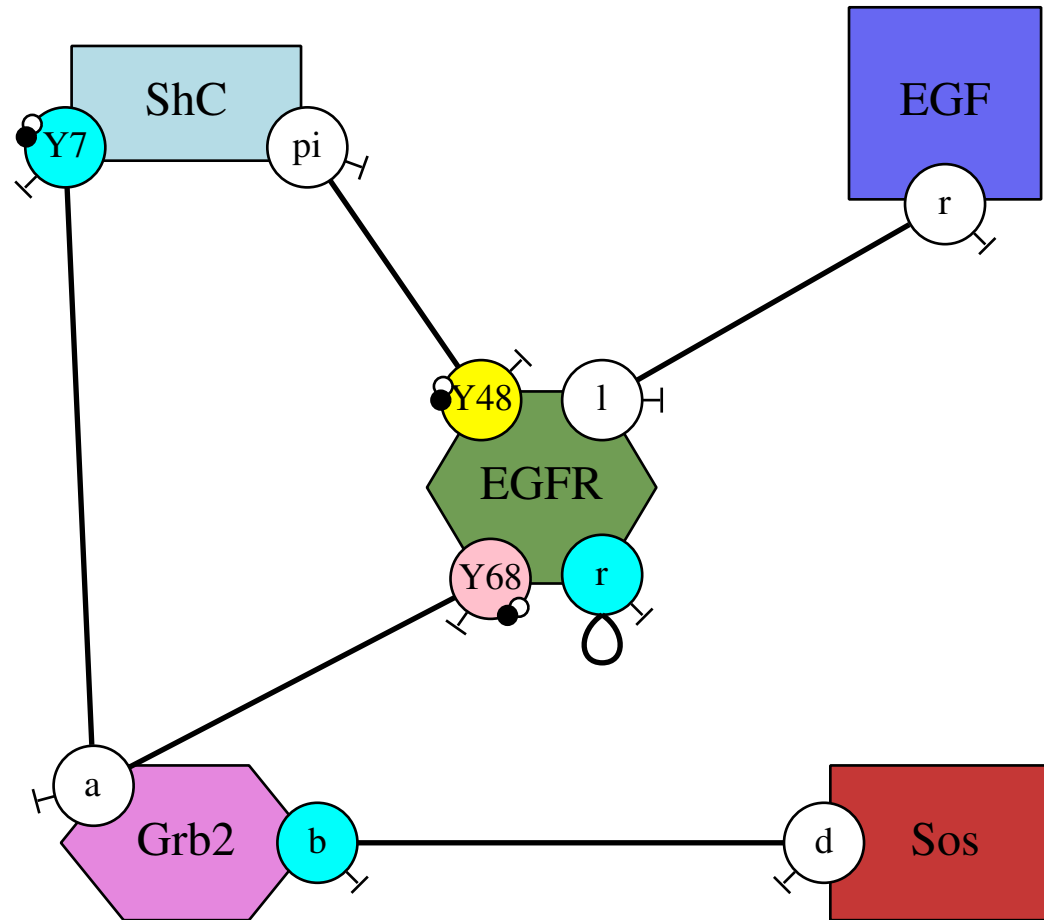
# Causal traces

# ODE semantics

EGF pathway (reduced ODEs)
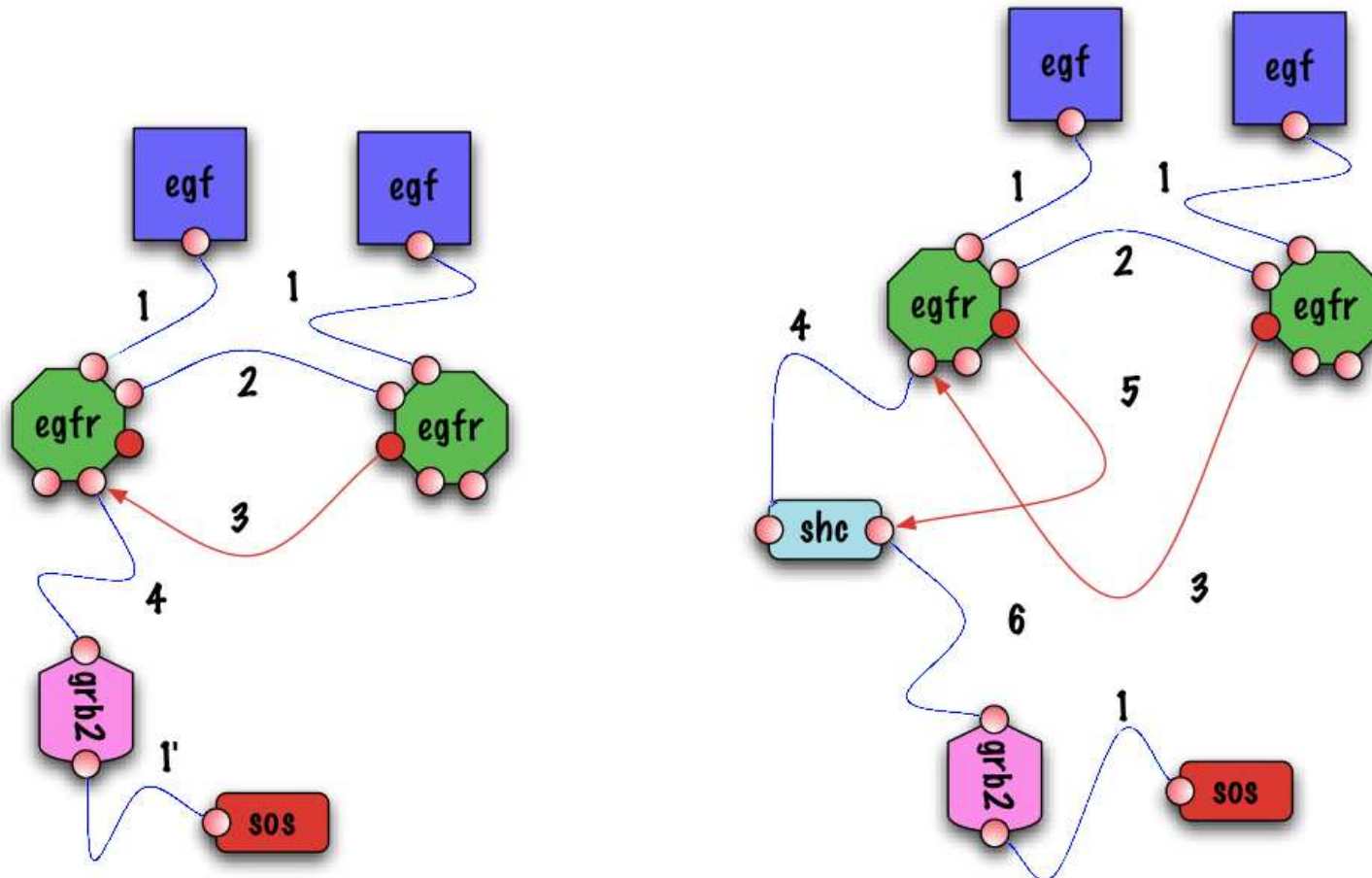
# ODE semantics
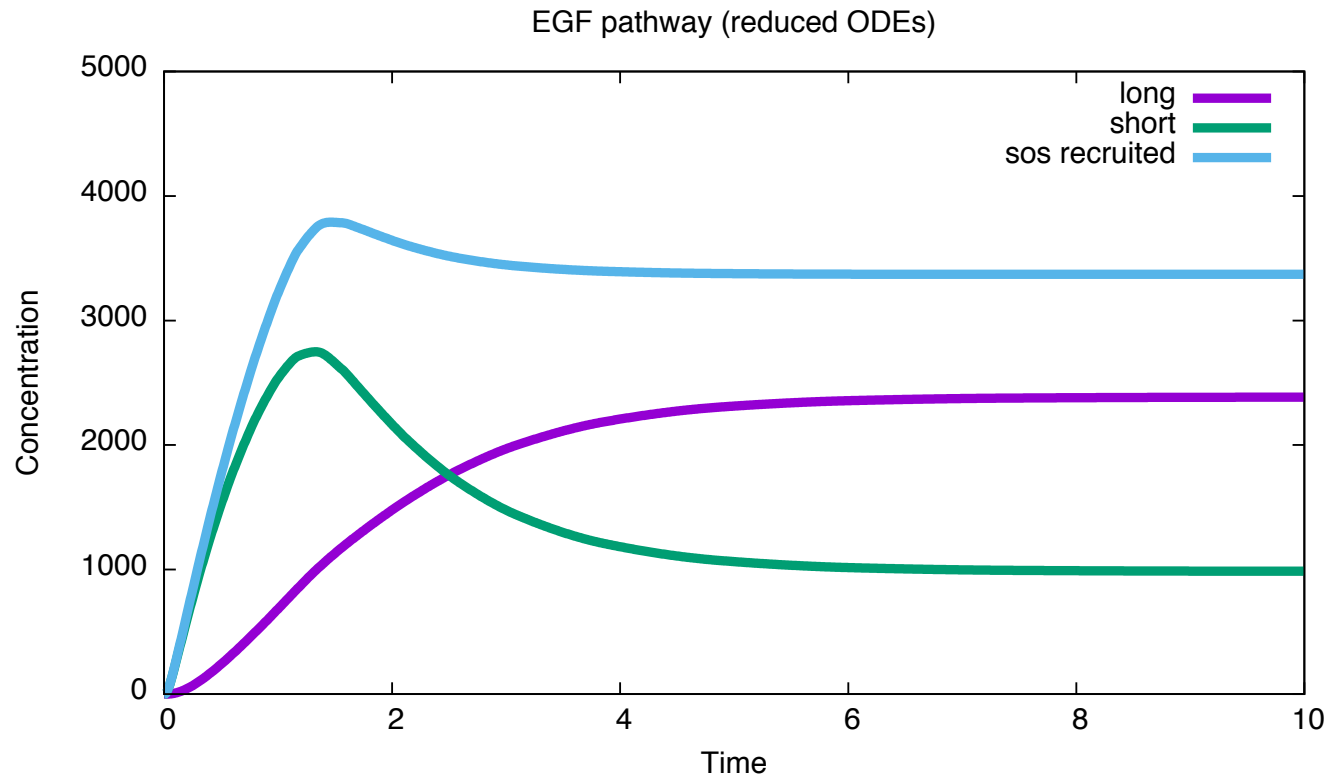
EGF pathway (reduced ODEs)
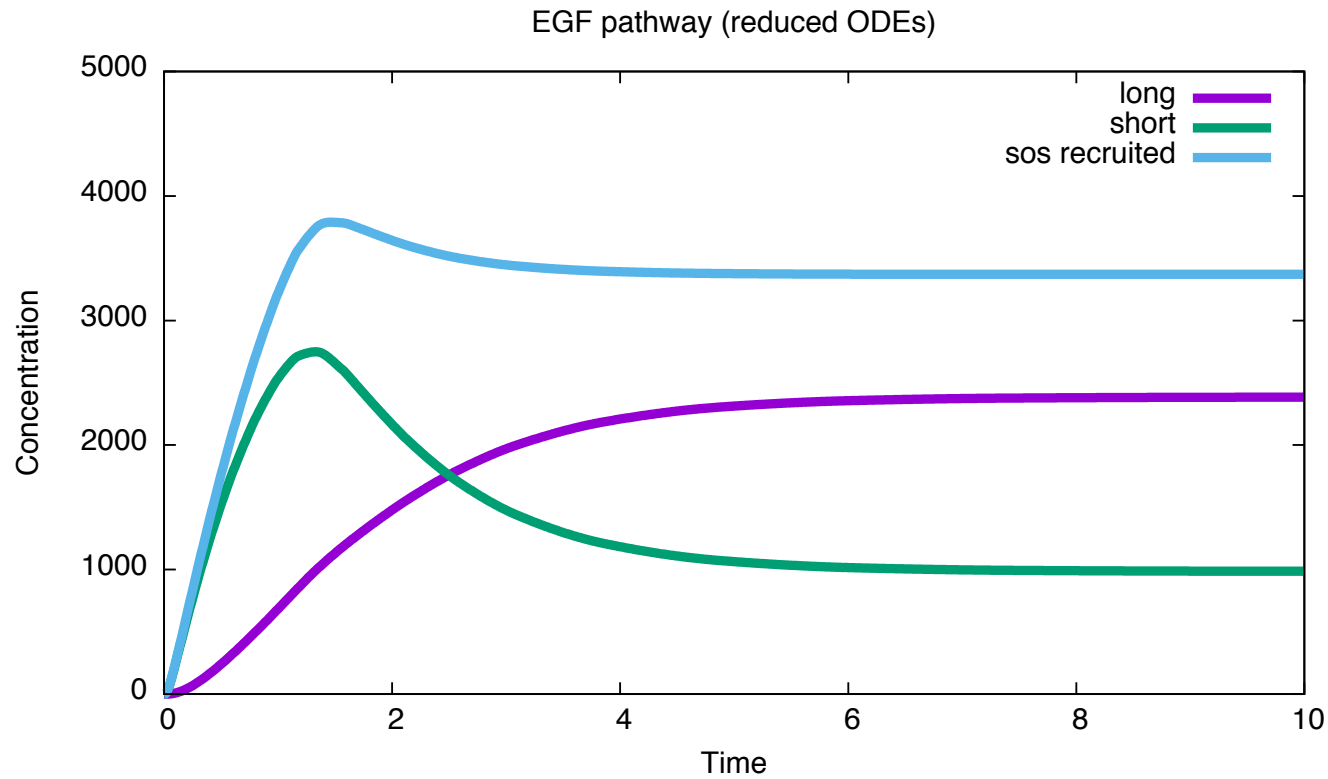


What will happen if more Shc(s) is put in the system?

# ODE semantics
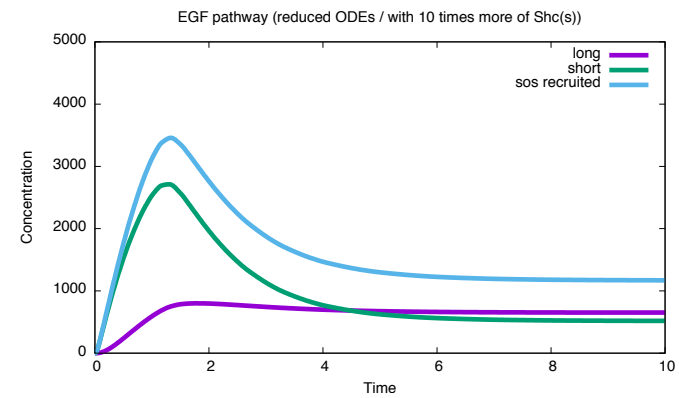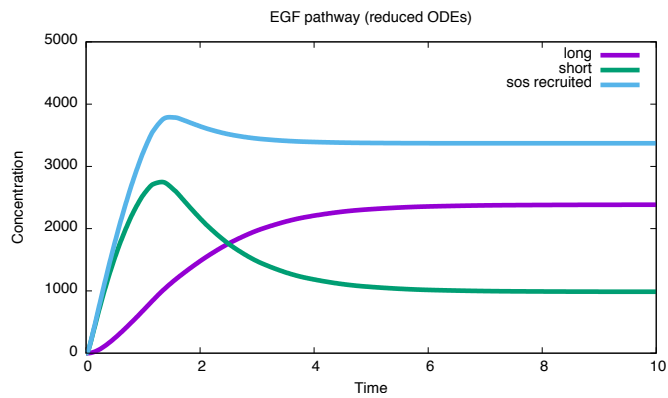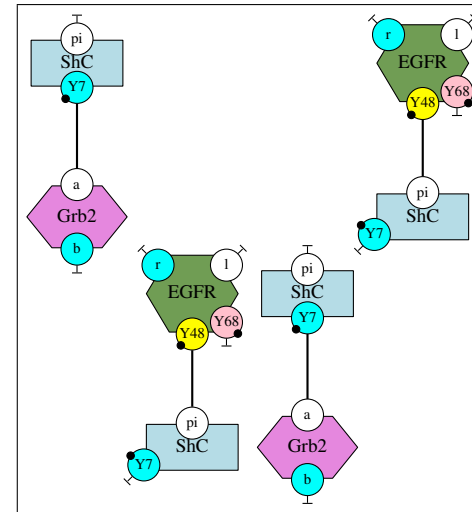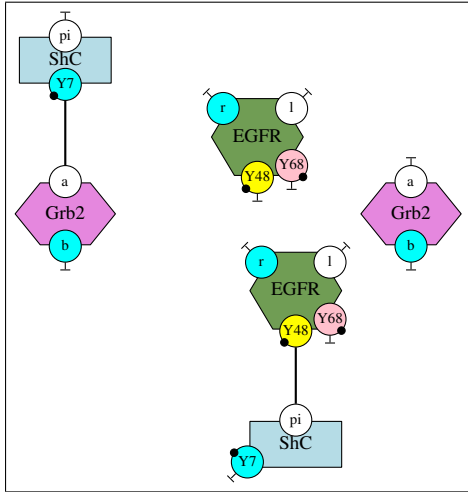


EGF pathway (reduced ODEs)



EGF pathway (reduced ODEs / with 10 times more of Shc(s))

# 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
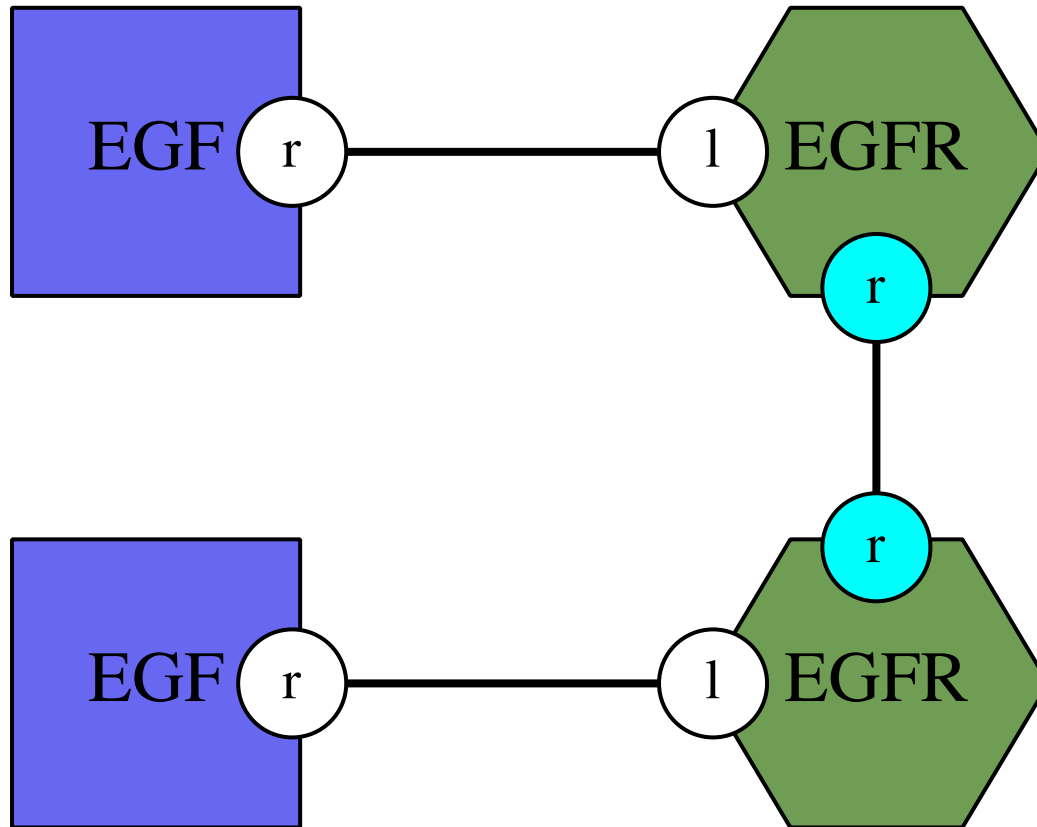


EGF(r!1), EGFR(l!1,r!2), EGFR(r!2,l!3), EGF(r!3)

# A Unbinding/Binding Rule



$$EGF(r),\ EGFR(l,r) \longleftrightarrow EGF(r!1),\ EGFR(l!1,r)$$

# Internal state



EGFR(Y48~u?,l!1), EGF(r!1) ⟷ EGFR(Y48~p?,l!1), EGF(r!1)
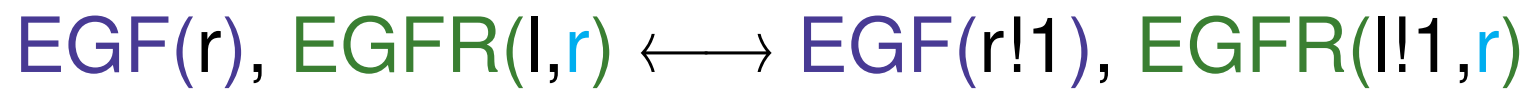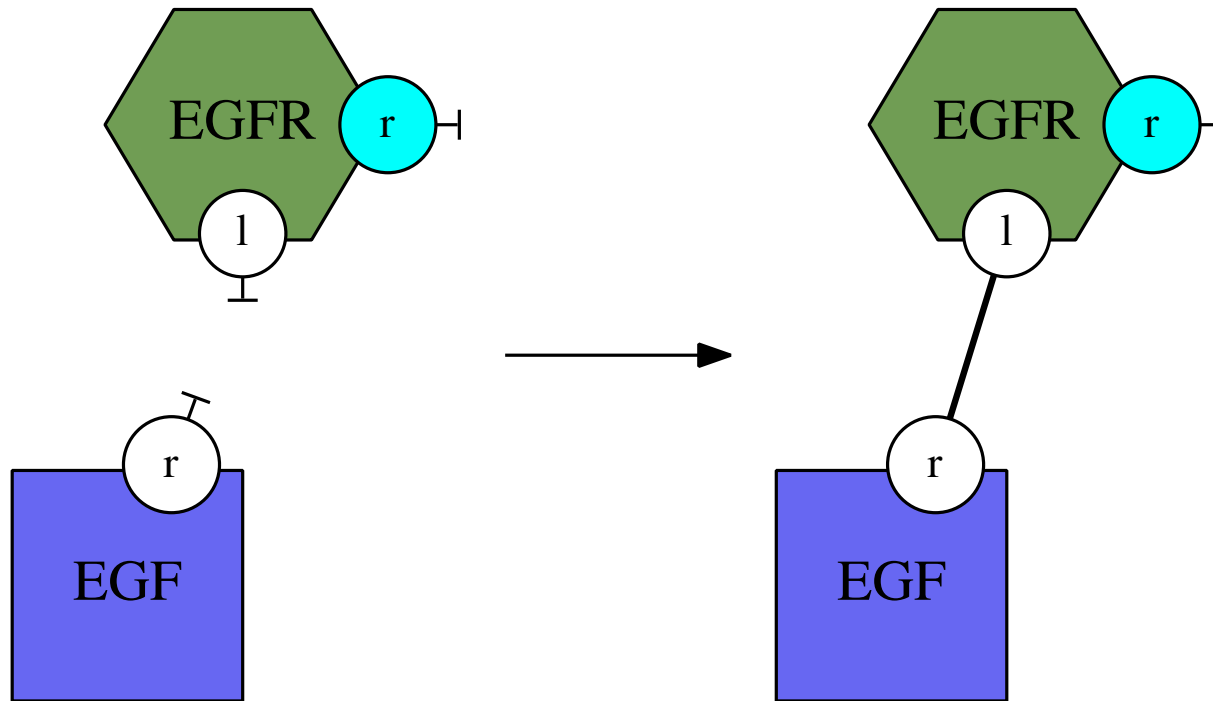
# Don't care, Don't write

# A contextual rule



EGFR(Y48~u,r!_) $\rightarrow$ EGFR(Y48~p,r)

# Creation/Suppression



$R(r) \rightarrow R(r!1), R(r!1,l,Y48{\sim}u)$



$R(r!1), R(r!1) \rightarrow R(r)$
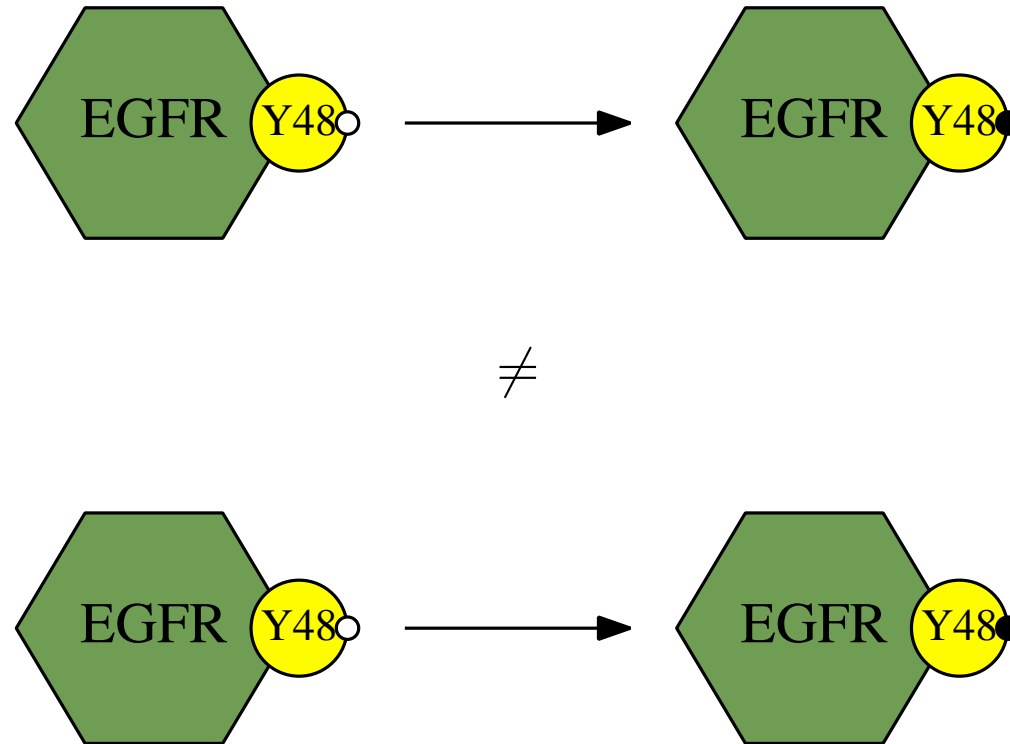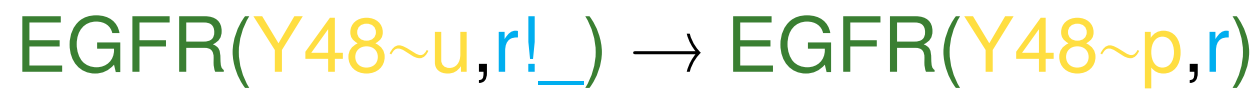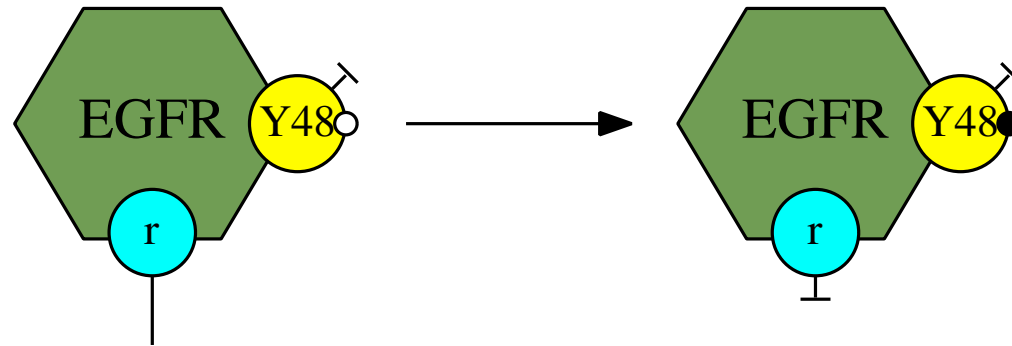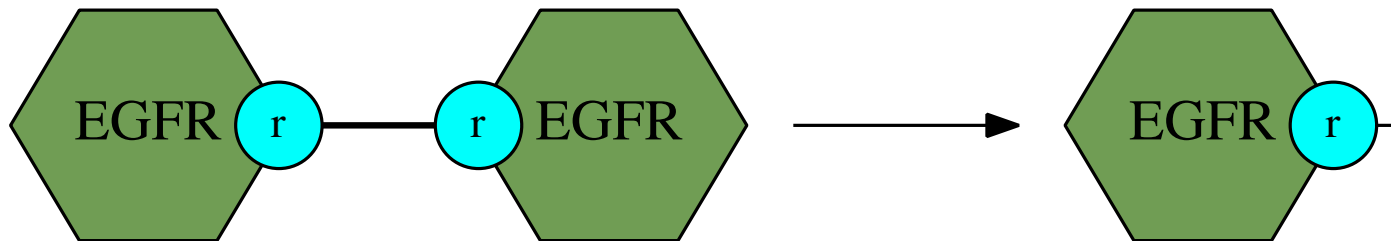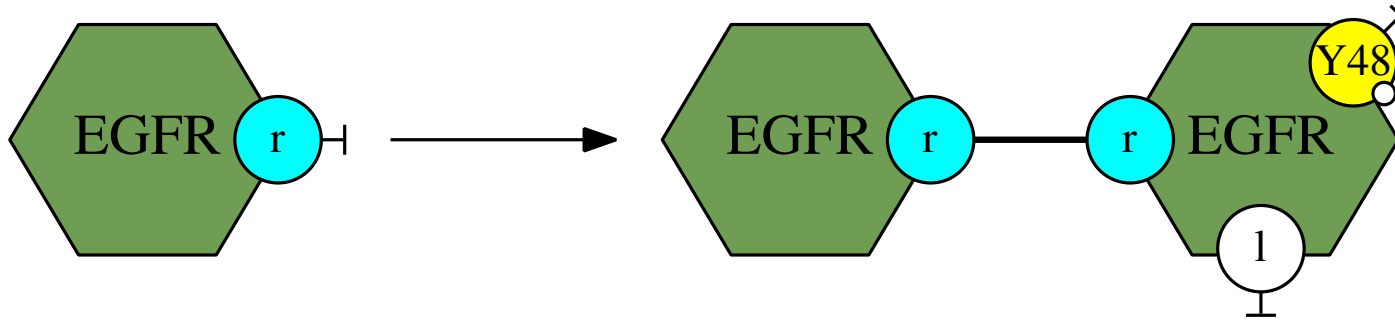
# Early EGF example

## 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, $\pi$:=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)<->So(d$^1$)+G(b$^1$)

interface note: highlight
the interacting parts

- **G-So binding**
  - 10: R(Y68$^{p1}$), G(a$^1$,b), So(d) <-> R(Y68$^{p1}$), G(a$^1$,b$^2$), So(d$^2$)
  - 12: G(a,b), So(d) <-> G(a,b$^1$), So(d$^1$)
  - 22: Sh(π,Y7$^{p2}$), G(a$^2$,b), So(d) <-> Sh(π,Y7$^{p2}$), G(a$^2$,b$^1$), S(d$^1$)
  - 19: Sh(π$^1$,Y7$^{p2}$), G(a$^2$,b), So(d) <-> Sh(π$^1$,Y7$^{p2}$), G(a$^2$,b$^1$), S(d$^1$)

- **Y48-Sh binding**
  - 13: R(Y48$^p$), Sh(π,Y7) <-> R(Y48$^{p1}$), Sh(π$^1$,Y7)
  - 15: R(Y48$^p$), Sh(π,Y7$^p$) <-> R(Y48$^{p1}$), Sh(π$^1$,Y7$^p$)
  - 18: R(Y48$^p$), Sh(π,Y7$^{p1}$), G(a$^1$,b) <-> R(Y48$^{p2}$), Sh(π$^2$,Y7$^{p1}$), G(a$^1$,b)
  - 20: R(Y48$^p$), Sh(π,Y7$^{p1}$), G(a$^1$,b$^3$), S(d$^3$) <-> R(Y48$^{p2}$), Sh(π$^2$,Y7$^{p1}$), G(a$^1$,b$^3$), S(d$^3$)

refined from
R(Y48$^p$)+Sh(π)<->R(Y48$^{p1}$)+Sh(π$^1$)

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

- **Sh-G binding**
  - 17: R(Y48$^{p1}$), Sh(π$^1$,Y7$^p$), G(a,b) <-> R(Y48$^{p1}$), Sh(π$^1$,Y7$^{p2}$), G(a$^2$,b)
  - 21: Sh(π,Y7$^p$), G(a,b) <-> Sh(π,Y7$^{p1}$), G(a$^1$,b)
  - 23: Sh(π,Y7$^p$), G(a,b$^2$) <-> Sh(π,Y7$^{p1}$), G(a$^1$,b$^2$)
  - 24: R(Y48$^{p1}$), Sh(π$^1$,Y7$^p$), G(a,b$^3$), S(d$^3$) <-> R(Y48$^{p1}$), Sh(π$^1$,Y7$^{p2}$), G(a$^2$, b$^3$), S(d$^3$)

refined from
Sh(π), G(a)<->Sh(π$^1$), 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
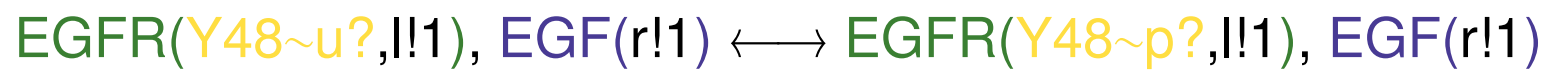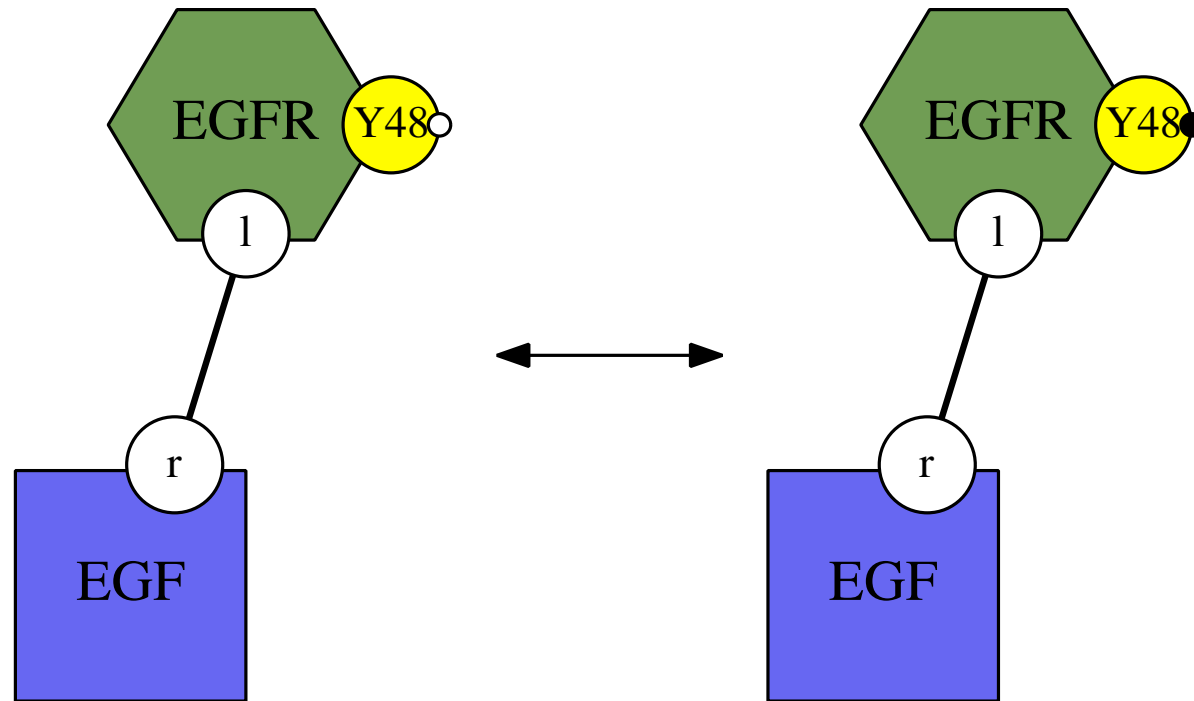
A rule is a symbolic representation of a multi-set of reactions.

For instance, the rule:
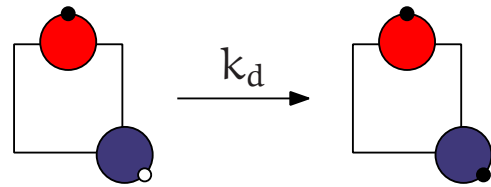


within a model with the following signature:
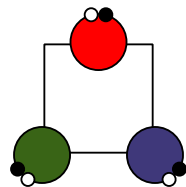


denotes the following two rules:
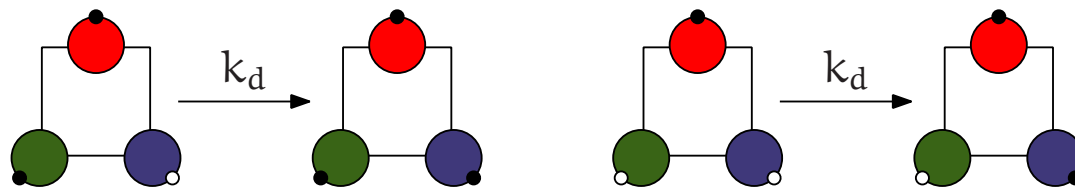
# Set of reachable chemical species

Let $\mathcal{R} = \{R_i\}$ be a set of rules.
Let *Species* be the set of all chemical species ($C, c_1, c_1', \ldots, c_k, c_k', \ldots \in$ *Species*).
Let $Species_0$ be the set of initial .

We are interested in $Species_\omega$ 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:

$$\mathbb{F} : \begin{cases} \wp(Species) & \rightarrow \wp(Species) \\ X & \mapsto X \cup \left\{ c_j' \middle| \begin{array}{l} \exists R_k \in \mathcal{R}, c_1, \ldots, c_m \in X, \\ c_1, \ldots, c_m \rightarrow_{R_k} c_1', \ldots, c_n' \end{array} \right\} \end{cases}.$$

The set $\wp(Species)$ is a complete lattice.
The mapping $\mathbb{F}$ is an extensive $\cup$-complete morphism.

We define the set of reachable chemical species as follows:

$$Species_\omega = \bigcup \left\{ \mathbb{F}^n(Species_0) \middle| n \in \mathbb{N} \right\}.$$

# Local views
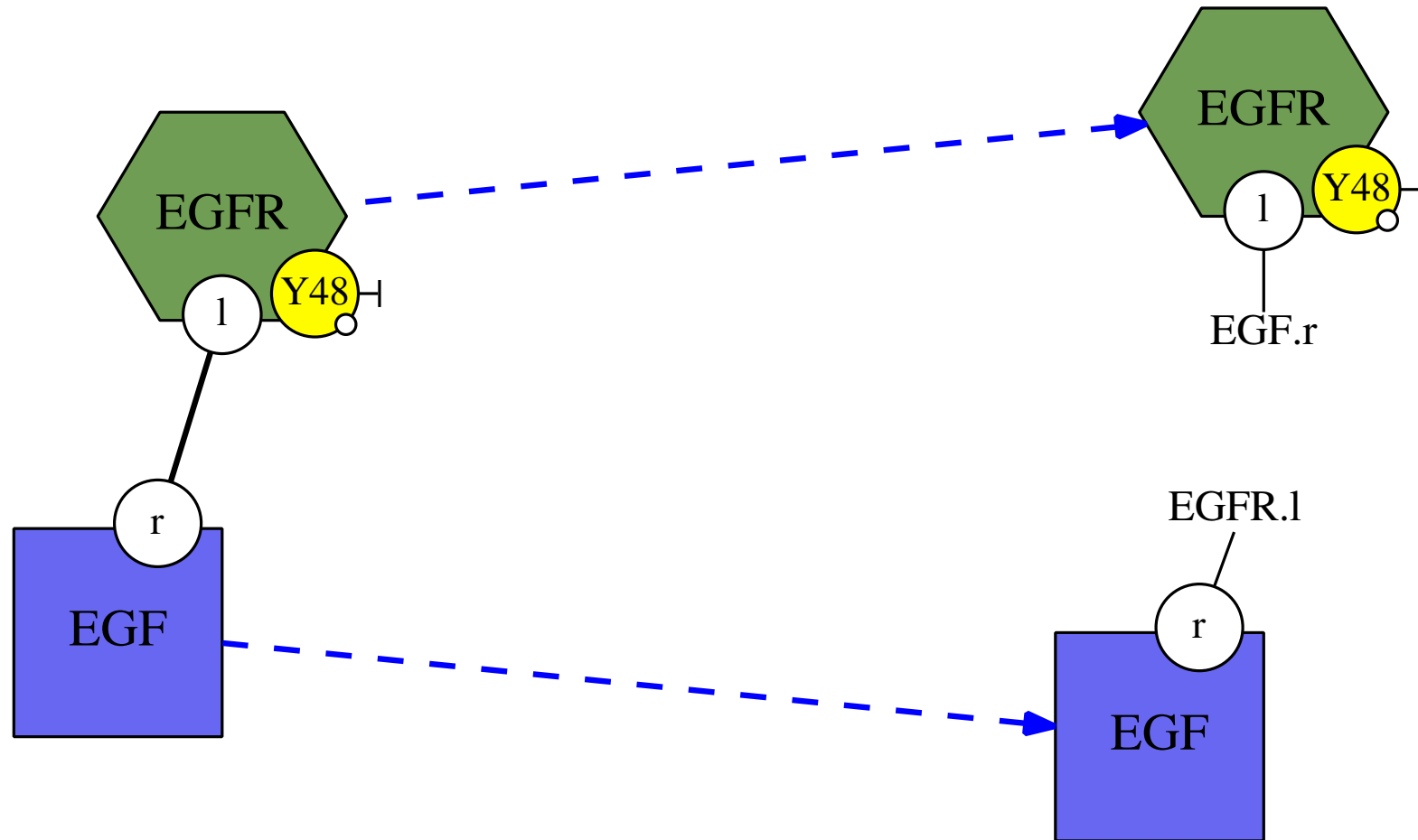


$$\alpha(\{R(Y1{\sim}u,l!1),\ E(r!1)\}) = \{R(Y1{\sim}u,l!r.E);\ E(r!l.R)\}.$$

# 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 $\alpha$ is a $\cup$-complete morphism.

Thus, it defines a Galois connection:

$$\wp(Species) \xleftrightarrow[\alpha]{\gamma} \wp(Local\_view).$$

(The function $\gamma$ 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 ?
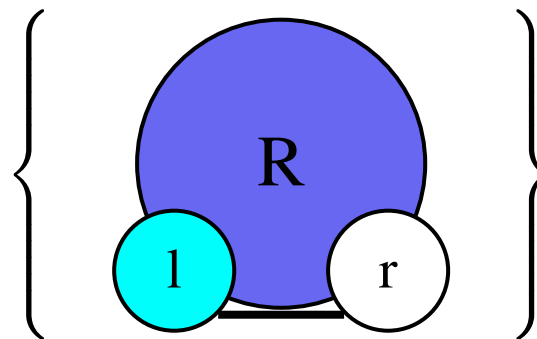
# α ∘ γ

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

Guess the image of the following set of local views ?
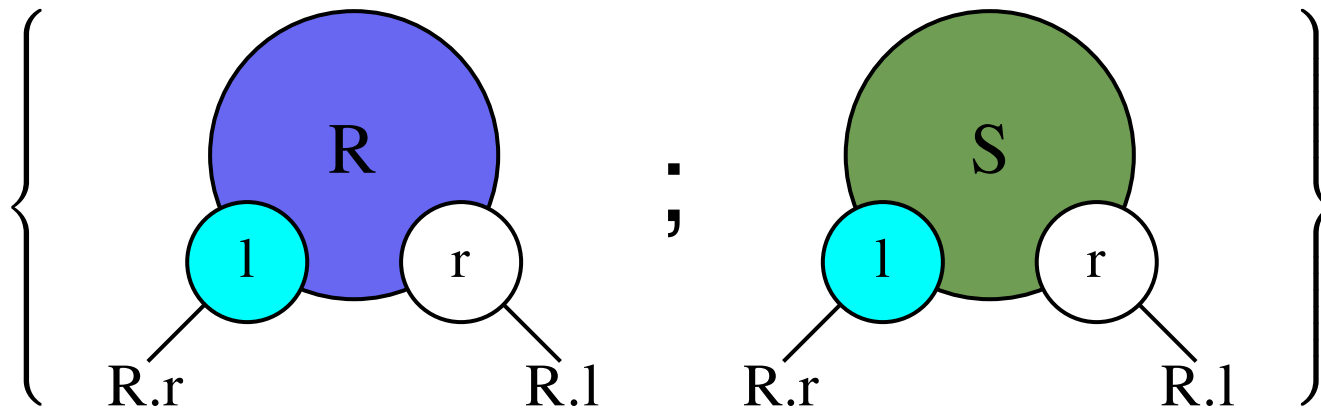
# One more question

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

Guess the image of the following set of local views ?
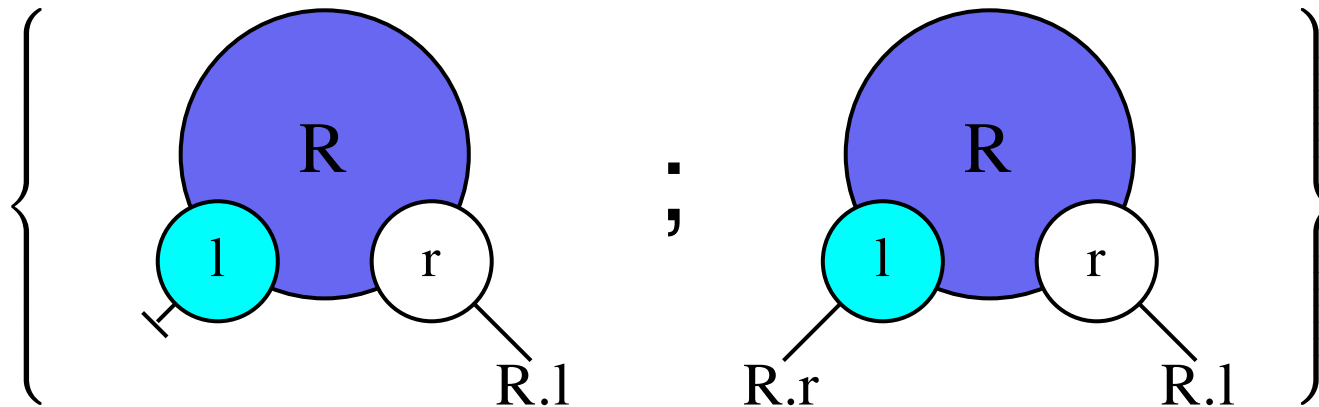
# Abstract reactions
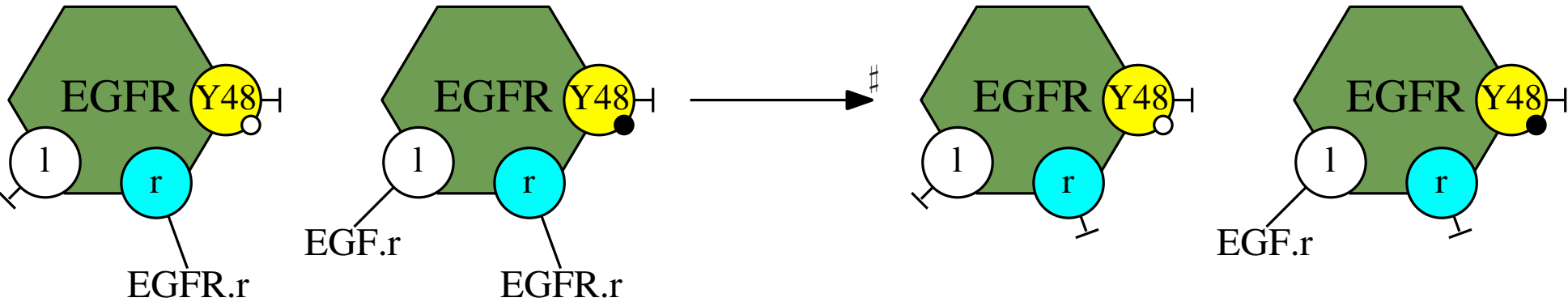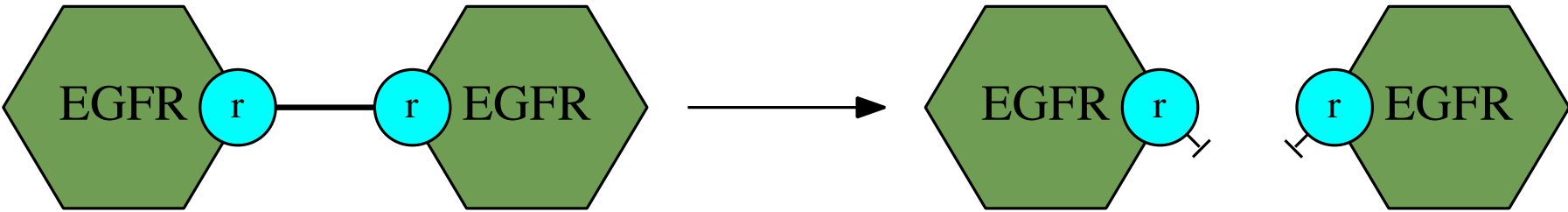
# Abstract counterpart to $\mathbb{F}$

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

$$\mathbb{F}^\sharp : \begin{cases} \wp(\textit{Local\_view}) & \to \wp(\textit{Local\_view}) \\ Y & \mapsto Y \cup \left\{ lv'_j \;\middle|\; \begin{array}{l} \exists R_k \in \mathcal{R}, lv_1, \ldots, lv_m \in Y, \\ lv_1, \ldots, lv_m \to^\sharp_{R_k} lv'_1, \ldots, lv'_n \end{array} \right\} \end{cases}.$$

We have:

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

# **Soundness**

**Theorem 1** Let:

1. $(D, \subseteq, \cup)$ and $(D^\sharp, \sqsubseteq, \sqcup)$ be chain-complete partial orders;

2. $D \xleftarrow[\alpha]{\gamma} D^\sharp$ be a Galois connection;

3. $\mathbb{F} \in D \to D$ and $\mathbb{F}^\sharp \in D^\sharp \to D^\sharp$ be monotonic mappings such that: $\mathbb{F} \circ \gamma \mathbin{\dot{\subseteq}} \gamma \circ \mathbb{F}^\sharp$;

4. $X_0 \in D$ be an element such that: $X_0 \subseteq \mathbb{F}(X_0)$;

Then:

1. both $\mathit{lfp}_{X_0} \mathbb{F}$ and $\mathit{lfp}_{\alpha(X_0)} \mathbb{F}^\sharp$ exist,

2. $\mathit{lfp}_{X_0} \mathbb{F} \subseteq \gamma(\mathit{lfp}_{\alpha(X_0)} \mathbb{F}^\sharp)$.

# **Overview**

1. Introduction

2. Language: Kappa

3. Abstraction: Local views

4. <span style="color:red">Completeness: false positives?</span>

5. Local fragment of Kappa

6. Decontextualization

7. Conclusion

# From views to species

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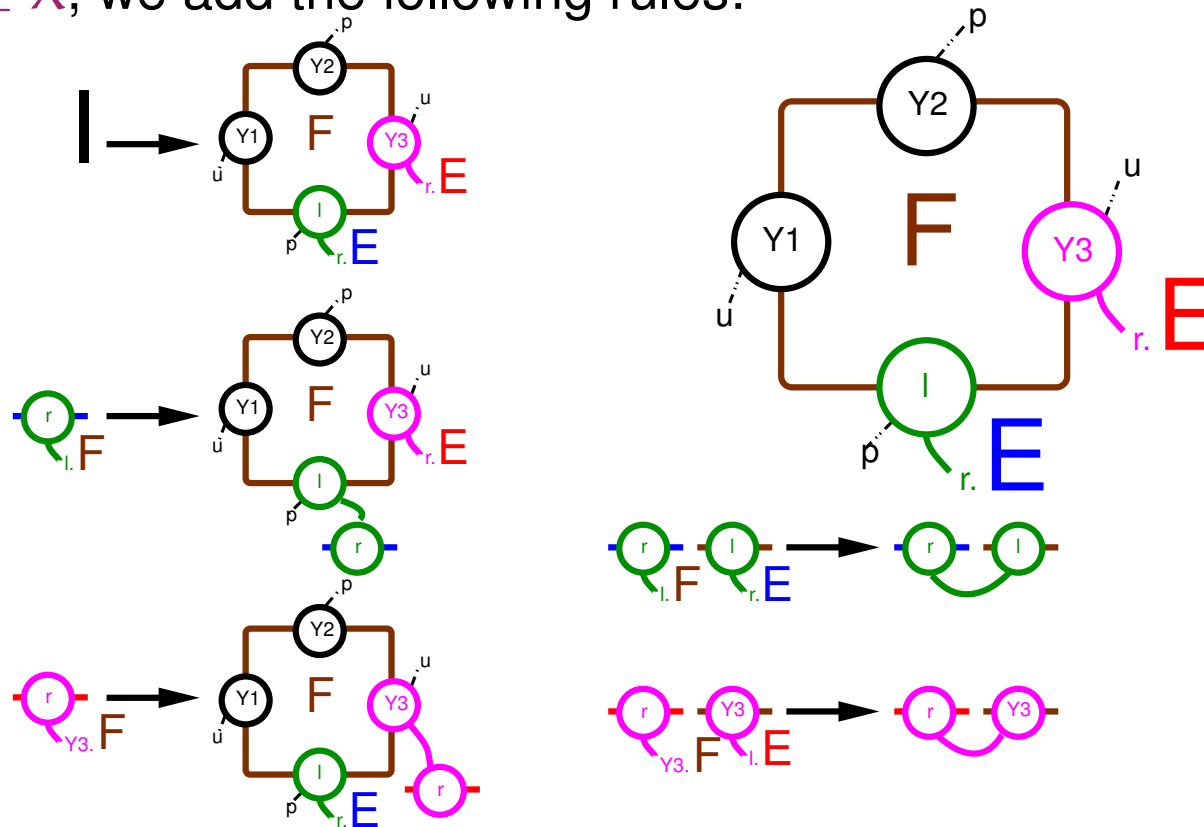


I and semi-links are non-terminal.
I is the initial symbol.

# 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:
     $$\mathsf{R}.\mathsf{l}\text{-}\mathsf{r}.\mathsf{E} \ldots \mathsf{R}.\mathsf{l}\text{-}\mathsf{r}.\mathsf{E}$$

     can be built.

- We only enumerate chemical species that are reached through an acyclic rewriting computation.

- It turns out that: if $X \in \alpha(\wp(\mathit{Species}))$ 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. $(D^\sharp, \sqsubseteq, \sqcup)$ be chain-complete partial order;

3. $D \xleftarrow[\alpha]{\gamma} D^\sharp$ be a Galois connection;

4. $\mathbb{F} \in D \to D$ and $\mathbb{F}^\sharp \in D^\sharp \to D^\sharp$ are monotonic;

5. $\mathbb{F} \circ \gamma \mathrel{\dot\subseteq} \gamma \circ \mathbb{F}^\sharp$;

6. $X_0, inv \in D$ such that:

   - $X_0 \subseteq \mathbb{F}(X_0) \subseteq \mathbb{F}(inv) \subseteq inv$,
   - $inv = \gamma(\alpha(inv))$,
   - and $\alpha(\mathbb{F}(inv)) = \mathbb{F}^\sharp(\alpha(inv))$;
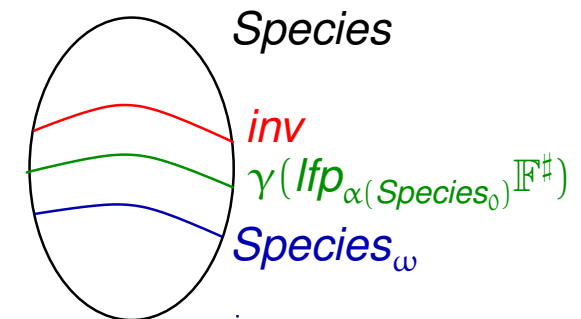
Then, $lfp_{\alpha(X_0)} \mathbb{F}^\sharp$ exists and $\gamma(lfp_{\alpha(X_0)} \mathbb{F}^\sharp) \subseteq inv$.



*Species*

*inv*

$\gamma(lfp_{\alpha(Species_0)} \mathbb{F}^\sharp)$

$Species_\omega$

# **Proof I/III**

We have already seen (previous lectures) that:

1. $\textit{lfp}_{\alpha(X_0)} \mathbb{F}^{\sharp}$ exists;

2. there exists an ordinal $\delta$ such that $\textit{lfp}_{\alpha(X_0)} \mathbb{F}^{\sharp} = \mathbb{F}^{\sharp\delta}(\alpha(X_0))$.

# **Proof II/III**

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

Let us prove instead by induction over $\delta$ that $\mathbb{F}^{\sharp\delta}(\alpha(X_0)) \sqsubseteq \alpha(\mathit{inv})$.

- If $Y \in D^\sharp$ is an element such that $Y \sqsubseteq \alpha(\mathit{inv})$,
  $$\mathbb{F}^\sharp(Y) \sqsubseteq \mathbb{F}^\sharp(\alpha(\mathit{inv})) \qquad (\mathbb{F}^\sharp \text{ is mon})$$
  $$\mathbb{F}^\sharp(\alpha(\mathit{inv})) = \alpha(\mathbb{F}(\mathit{inv})) \quad (\text{assumption})$$
  $$\alpha(\mathbb{F}(\mathit{inv})) \sqsubseteq \alpha(\mathit{inv}). \qquad (\alpha \text{ is mon and } \mathit{inv} \text{ is a post})$$

  Thus: $\mathbb{F}^\sharp(Y) \sqsubseteq \alpha(\mathit{inv})$

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

So: $\mathbb{F}^{\sharp\delta}(\alpha(X_0)) \sqsubseteq \alpha(\mathit{inv})$.

# **Proof III/III**

We have:
$$\mathbb{F}^{\sharp\delta}(\alpha(X_0)) \sqsubseteq \alpha(inv).$$

Since $\gamma$ is monotonic:

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

But, by assumption, $\gamma(\alpha(inv)) = inv$.
Thus,
$$\gamma(\mathbb{F}^{\sharp\delta}(\alpha(X_0))) \subseteq inv.$$

# When is there no false positive ?

**Theorem 3** We suppose that:

1. $(D, \subseteq, \cup)$ and $(D^\sharp, \sqsubseteq, \sqcup)$ are chain-complete partial orders;

2. $(D, \subseteq) \xleftrightarrow[\alpha]{\gamma} (D^\sharp, \sqsubseteq)$ is a Galois connection;

3. $\mathbb{F} : D \to D$ is a monotonic map;

4. $X_0$ is a concrete element such that $X_0 \subseteq \mathbb{F}(X_0)$;

5. $\mathbb{F} \circ \gamma \mathrel{\dot{\subseteq}} \gamma \circ \mathbb{F}^\sharp$;

6. $\mathbb{F}^\sharp \circ \alpha = \alpha \circ \mathbb{F} \circ \gamma \circ \alpha$.

Then:

- $lfp_{X_0} \mathbb{F}$ and $lfp_{\alpha(X_0)} \mathbb{F}^\sharp$ exist;

- $lfp_{X_0} \mathbb{F} = \gamma(\alpha(lfp_{X_0} \mathbb{F})) \iff lfp_{X_0} \mathbb{F} = \gamma(lfp_{\alpha(X_0)} \mathbb{F}^\sharp)$.

We need to understand under which assumptions $lfp_{X_0} \mathbb{F} = \gamma(\alpha(lfp_{X_0} \mathbb{F}))$.
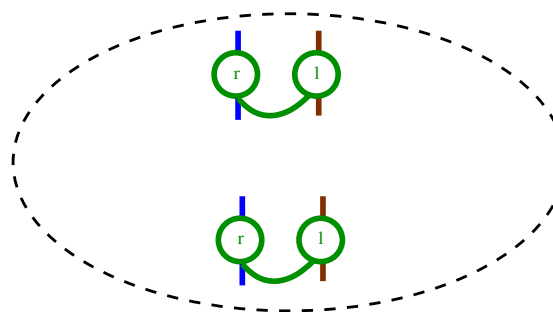
# Local set of chemical species

**Definition 1** We say that a set $X \in \wp(Species)$ of chemical species is local if and only if $X \in \gamma(\wp(Local\_view))$.

(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.)
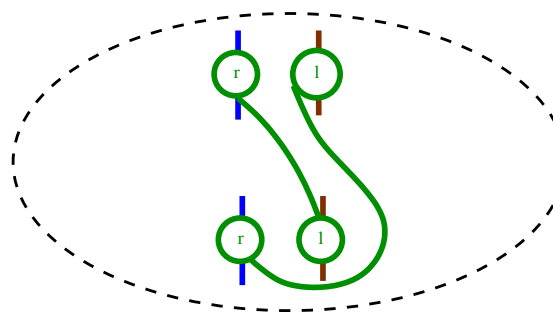
# Swapping relation

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

$(C_1, \ldots, C_m)$ matches with



while $(D_1, \ldots, D_n)$ 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 $(C_i) \overset{\text{SWAP}}{\sim} (D_j)$;
   
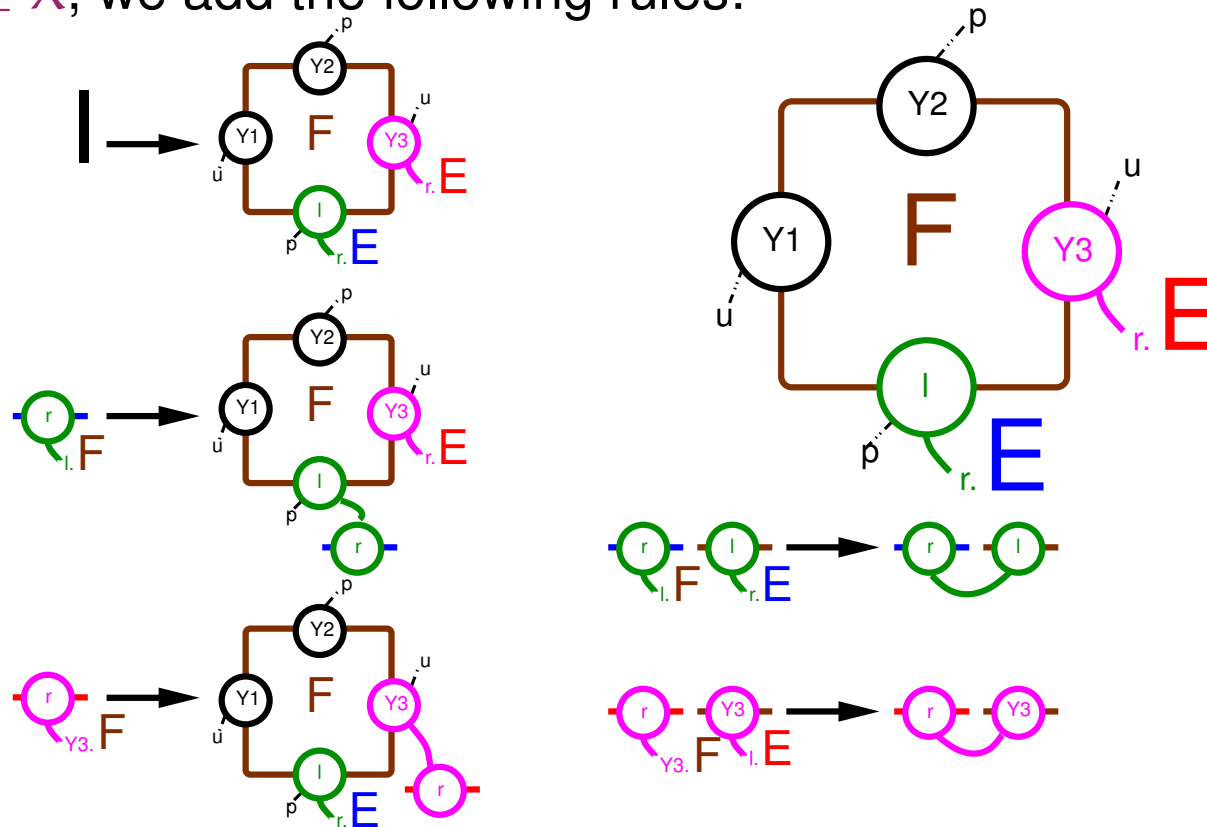   we have $(D_j) \in X^*$.

# Proof (easier implication way)

If:

- $X = \gamma(\alpha(X))$,

- $(C_i)_{i \in I} \in X^*$,

- and $(C_i)_{i \in I} \overset{\text{SWAP}}{\sim} (D_j)_{j \in J}$;

Then:

we have $\alpha(\{C_i \mid i \in I\}) = \alpha(\{D_j \mid j \in J\})$   (because $(C_i) \overset{\text{SWAP}}{\sim} (D_j)$)

and $\alpha(\{C_i \mid i \in I\}) \subseteq \alpha(X)$           (because $(C_i) \in X^*$ and $\alpha$ 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 $X \in \wp(Local\_view)$, $\gamma(X)$ 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 $\overset{\text{SWAP}}{\sim}$.
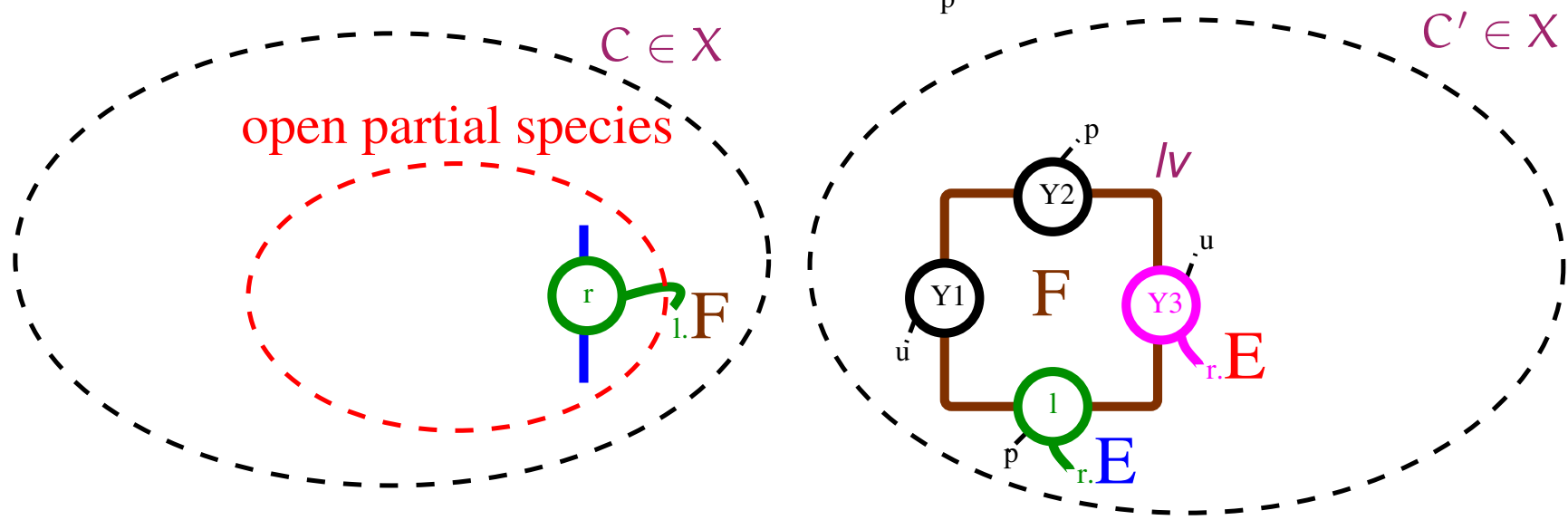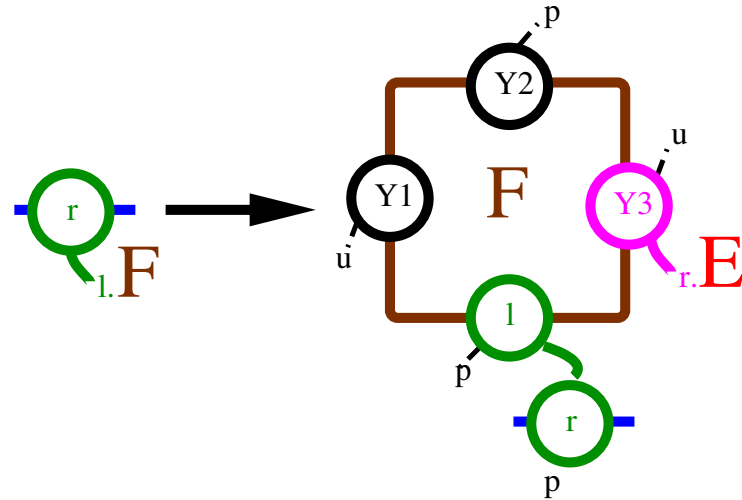We want to prove that $\gamma(\alpha(X)) \subseteq X$.

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 $\alpha$, this is satisfied for any local view in $\alpha(X)$;

- This remains satisfied after unfolding a semi-link with a local view;

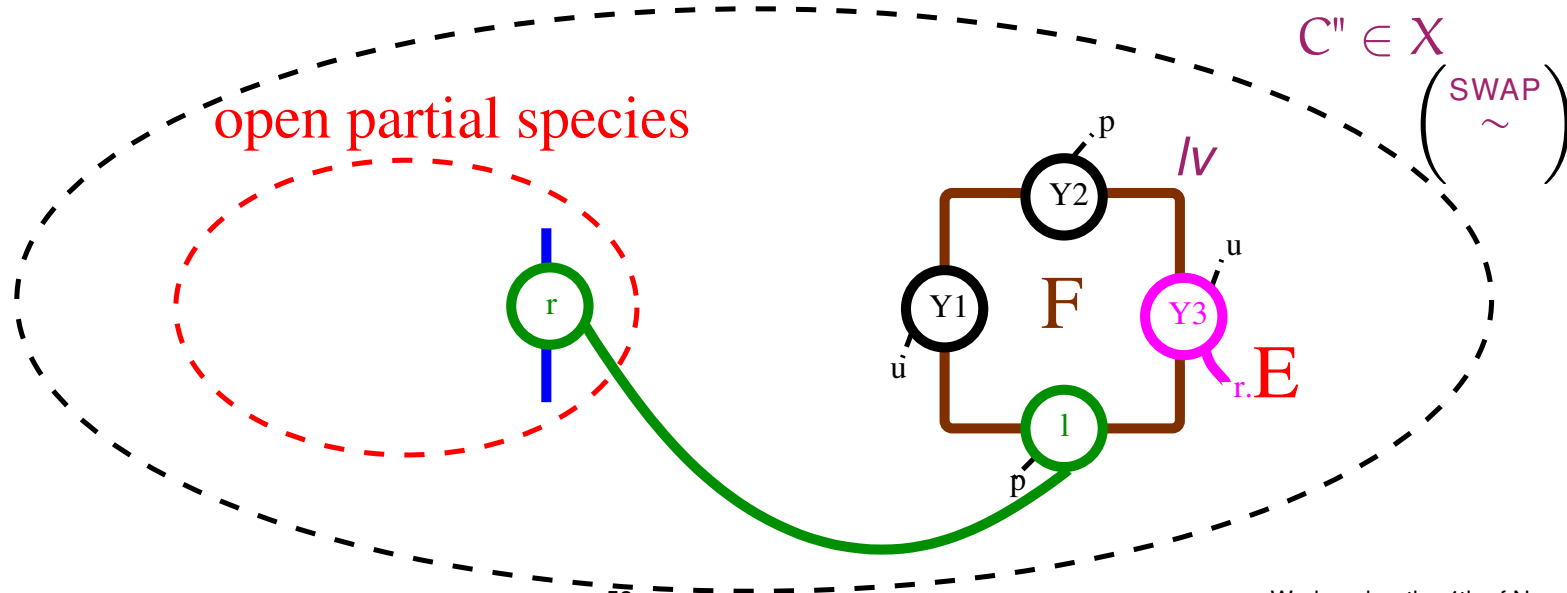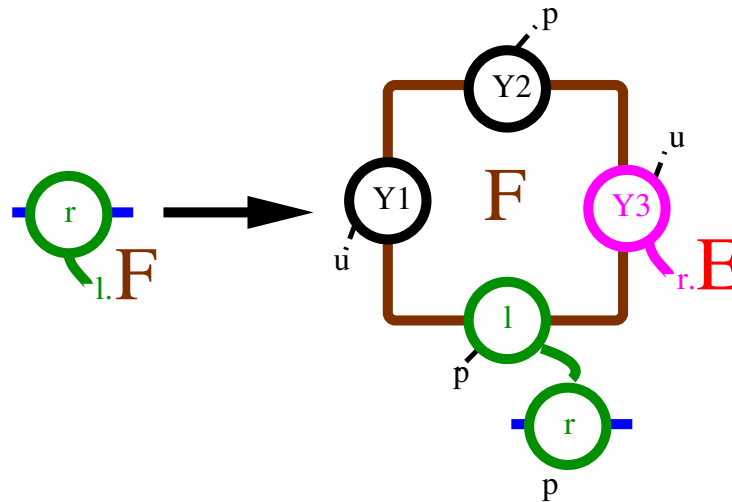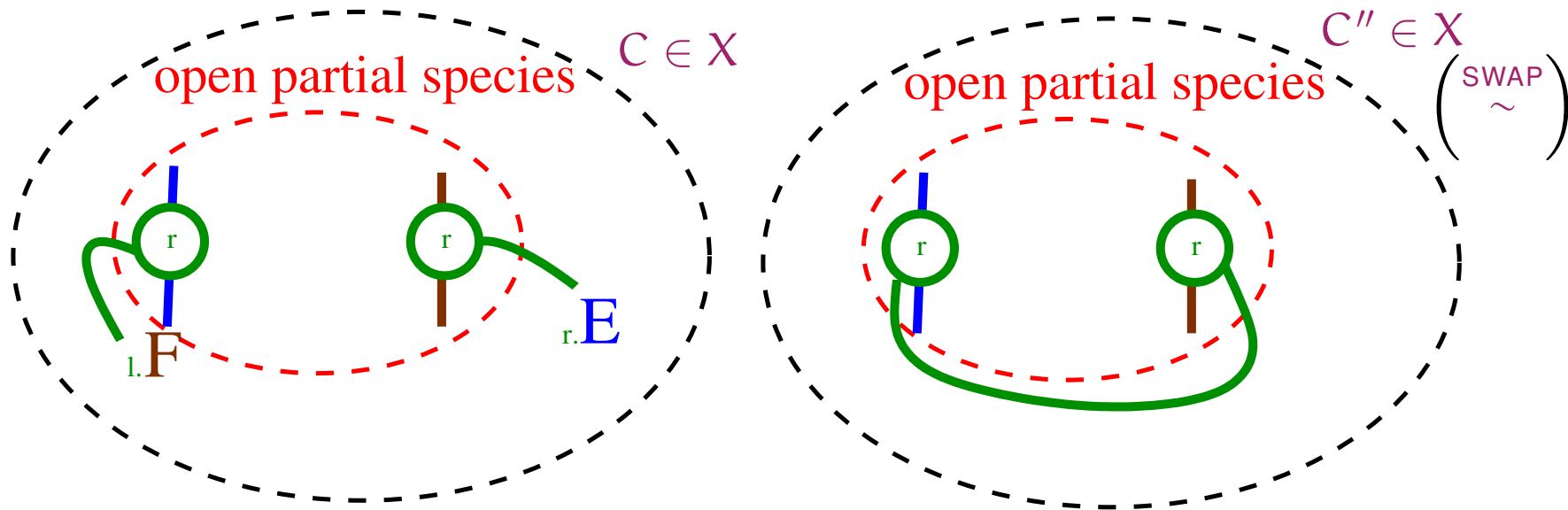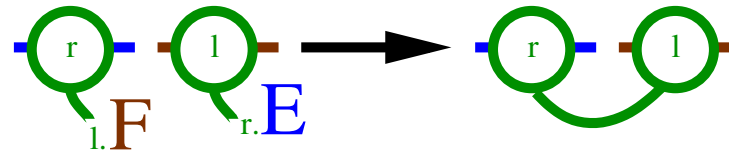- This remains satisfied after binding two semi-links.

# Initialization
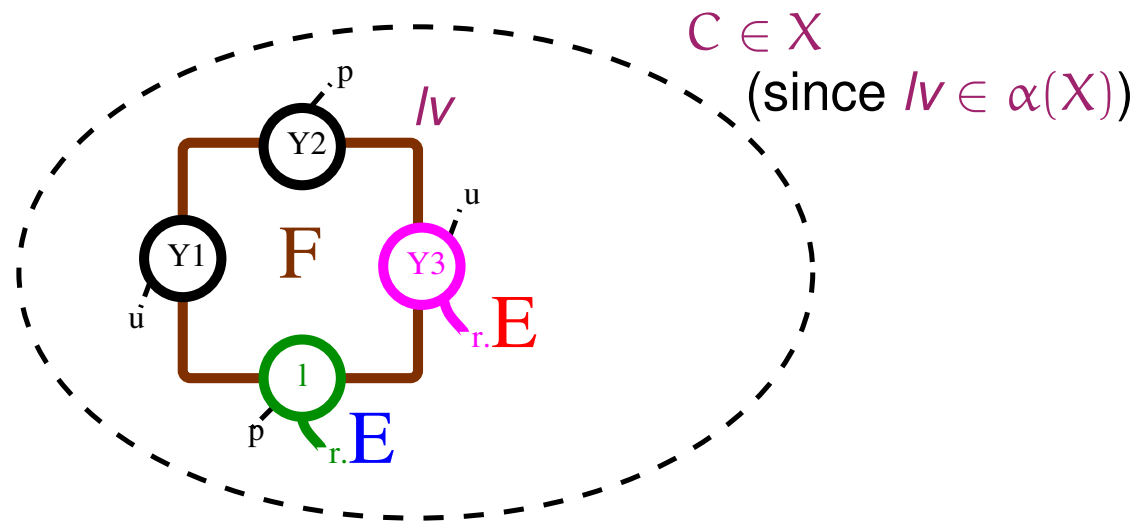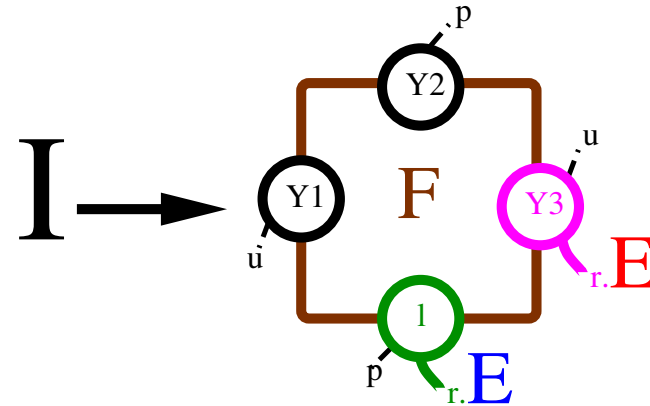
# Unfolding a semi-link

# Unfolding a semi-link

# Binding two semi-links

# Consequences

Let $Y \in \wp(\mathit{Local\_view}))$ 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 $\mathbb{F}^\sharp \circ \alpha = \alpha \circ \mathbb{F} \circ \gamma \circ \alpha$.

# **Overview**

# Outline

We have proved that:

- if the set $Species_\omega$ of reachable chemical species is close with respect to swapping $\overset{\text{SWAP}}{\sim}$,

- then the reachability analysis is exact (i.e. $Species_\omega = \gamma(lfp_{\alpha(Species_0)}\mathbb{F}^\sharp)$).

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~m,S),B(b~n,T) $\rightarrow$ A(a~m!1,S),B(b~n!1,T)
   - and A(a~m',S'),B(b~n',T') $\rightarrow$ A(a~m'!1,S'),B(b~n'!1,T'),

   then:

   - A(a~m,S),B(b~n',T') $\rightarrow$ A(a~m!1,S),B(b~n'!1,T');

5. chemical species in $\gamma(\alpha(Species_\omega))$ 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)
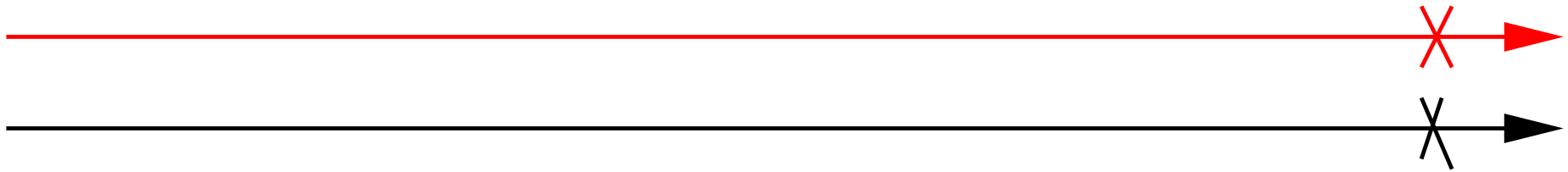
$\mathbb{C} \in Species_{\omega}$

$\mathbb{C}' \in Species_{\omega}$
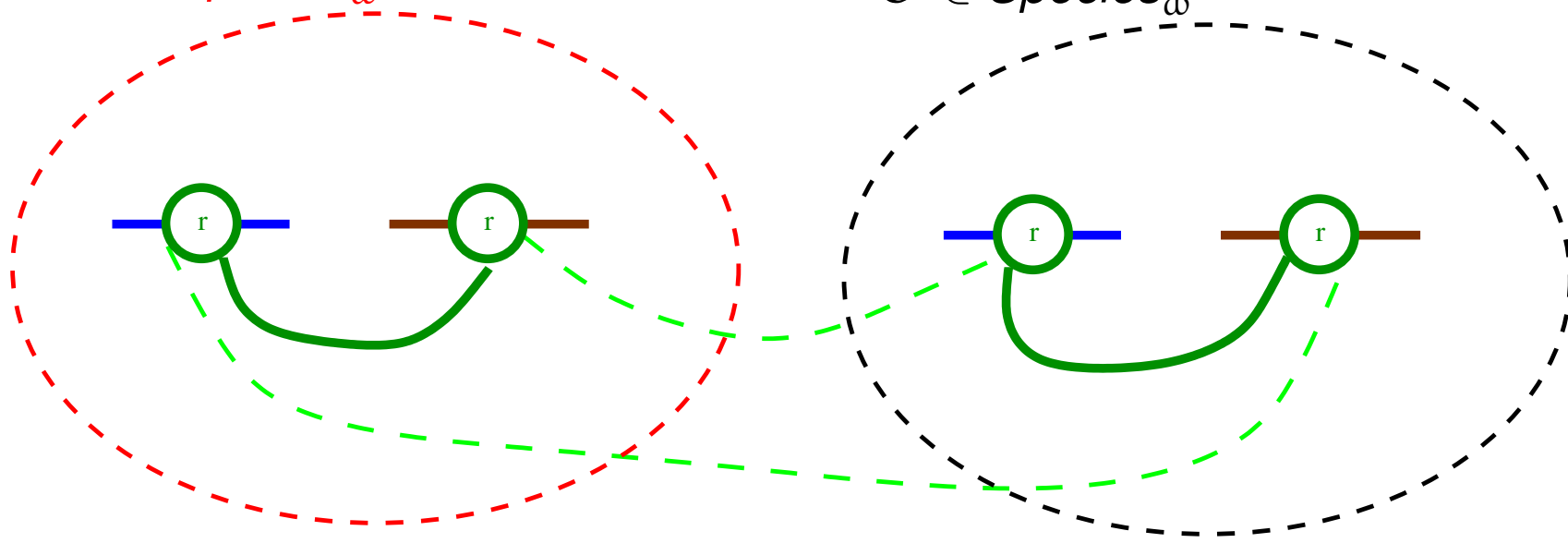
# First case (II/V)

just before the links are made



$C \in Species^*_\omega$
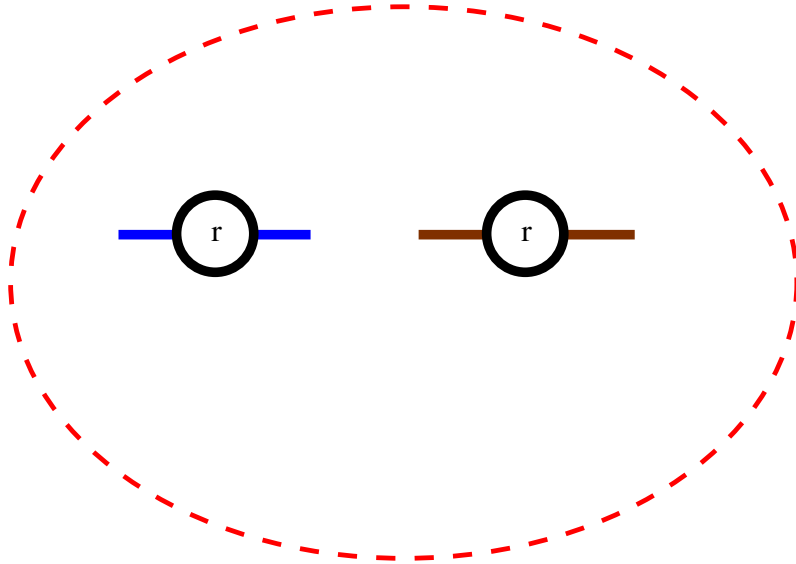
$C' \in Species^*_\omega$
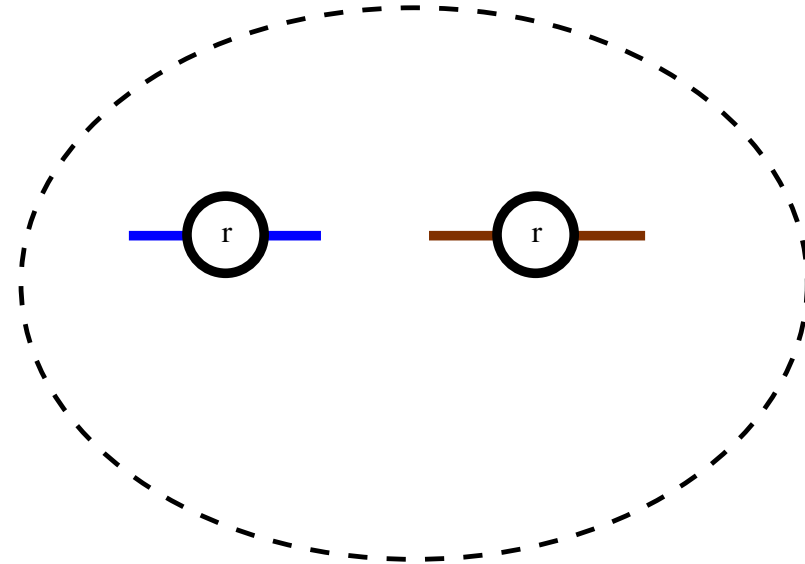
# First case (III/V)

we suppose we can swap the links
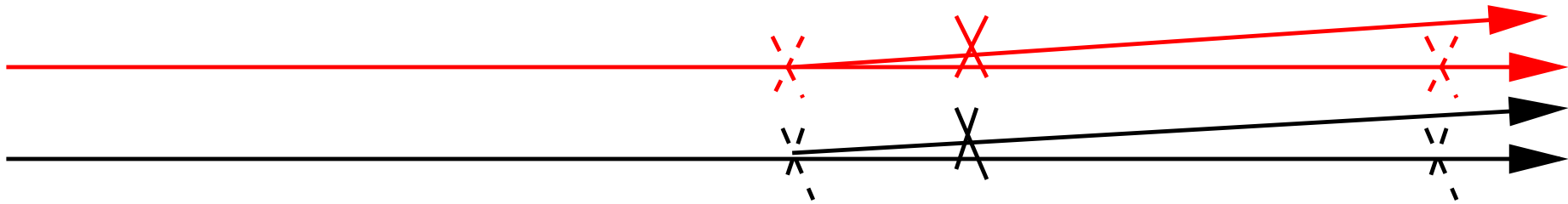


$C \in Species_\omega^*$
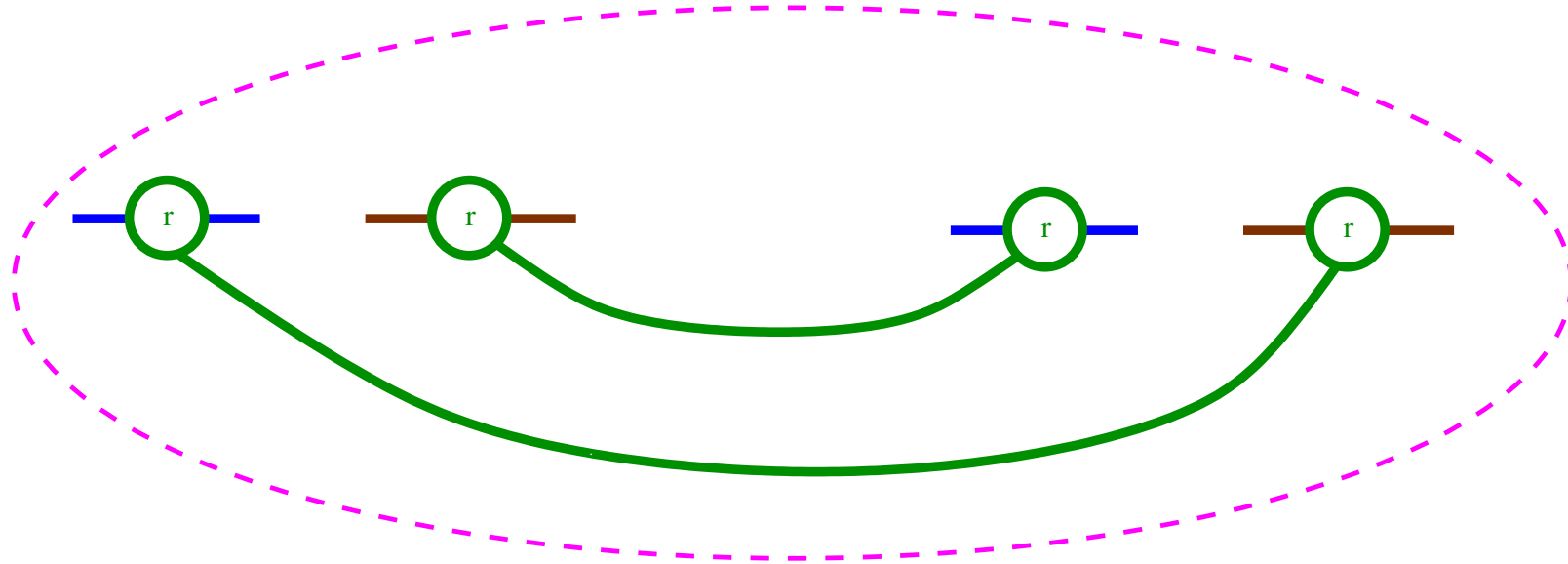
# First case (IV/V)

Then, we ensure that further computation steps:
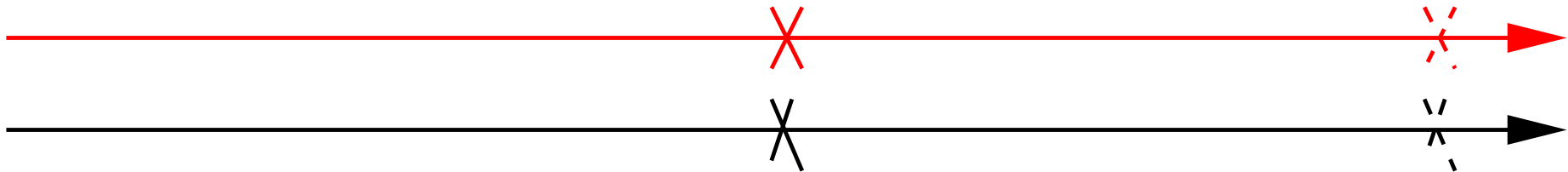
- are always possible;

- have the same effect on local views;

- commute with the swapping relation $\overset{\text{SWAP}}{\sim}$.

$$
\begin{array}{ccc}
C_n & \xrightarrow{\ \overset{\text{SWAP}}{\sim},\sigma\ } & C'_n \\
\Big\downarrow{\scriptstyle R,\phi} & & \Big\downarrow{\scriptstyle R,\phi} \\
C_{n+1} & \overset{\overset{\text{SWAP}}{\sim},\sigma}{\dashrightarrow} & C'_{n+1}
\end{array}
$$
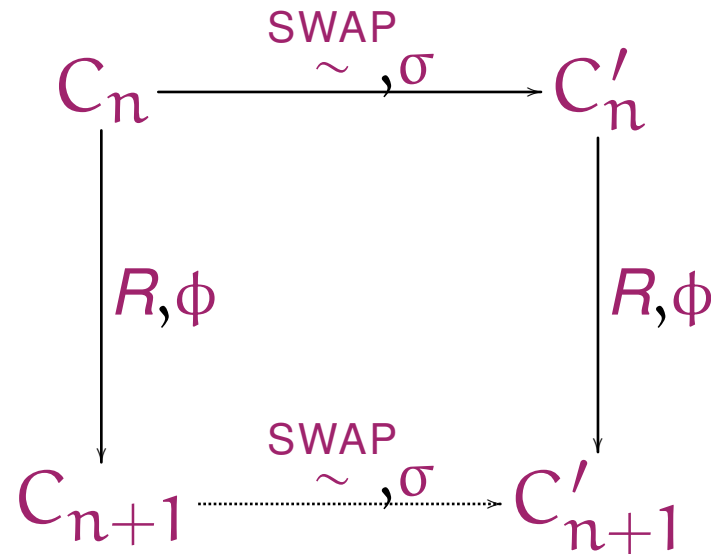
# First case (V/V)
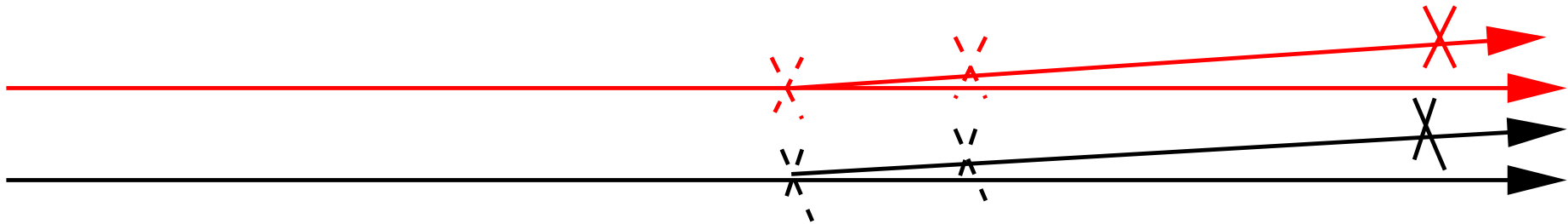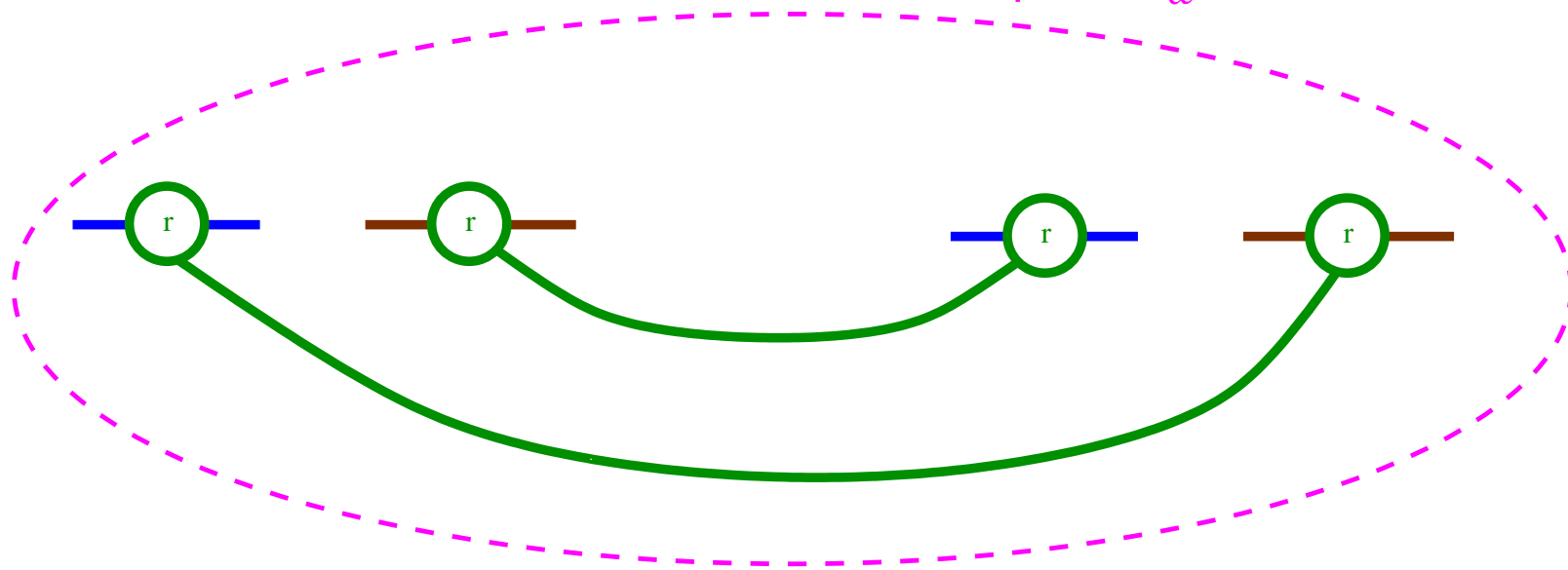


$$\mathbb{C} \in Species_\omega{}^*$$

# Second case (I/II)

$C \in Species_\omega$
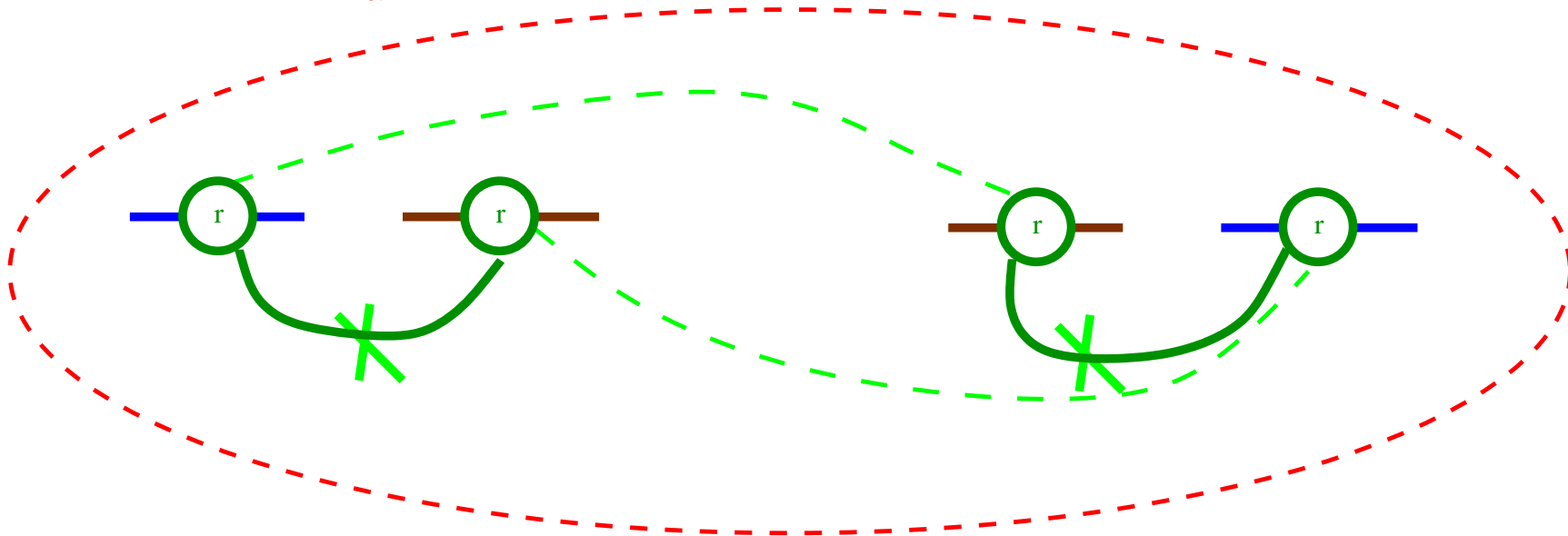


we assume that the chemical species C is acyclic

# 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~m,S),B(b~n,T) $\rightarrow$ A(a~m!1,S),B(b~n!1,T)
    - and A(a~m',S'),B(b~n',T') $\rightarrow$ A(a~m'!1,S'),B(b~n'!1,T'),

    then:

    - A(a~m,S),B(b~n',T') $\rightarrow$ A(a~m!1,S),B(b~n'!1,T');

5. chemical species in $\gamma(\alpha(\mathit{Species}_\omega))$ are acyclic,

are satisfied, the set of reachable chemical species is local.

# Third case (I/III)

$\mathbb{C} \in Species_\omega$
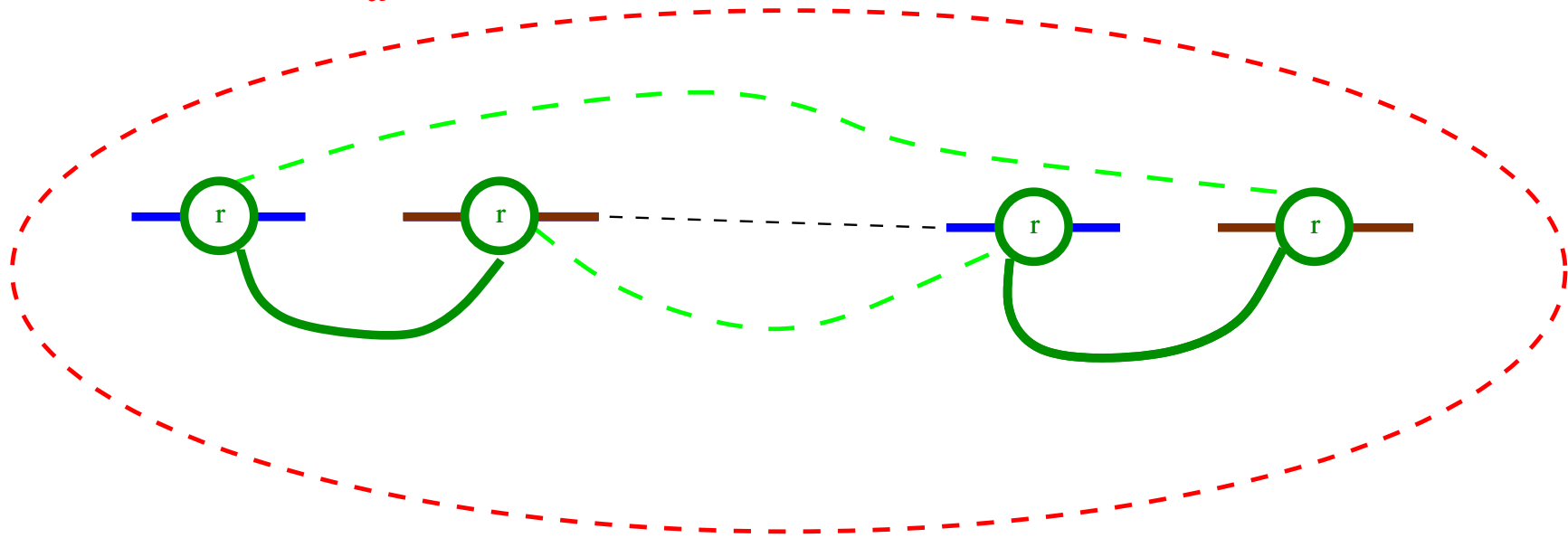
# Third case (II/III)
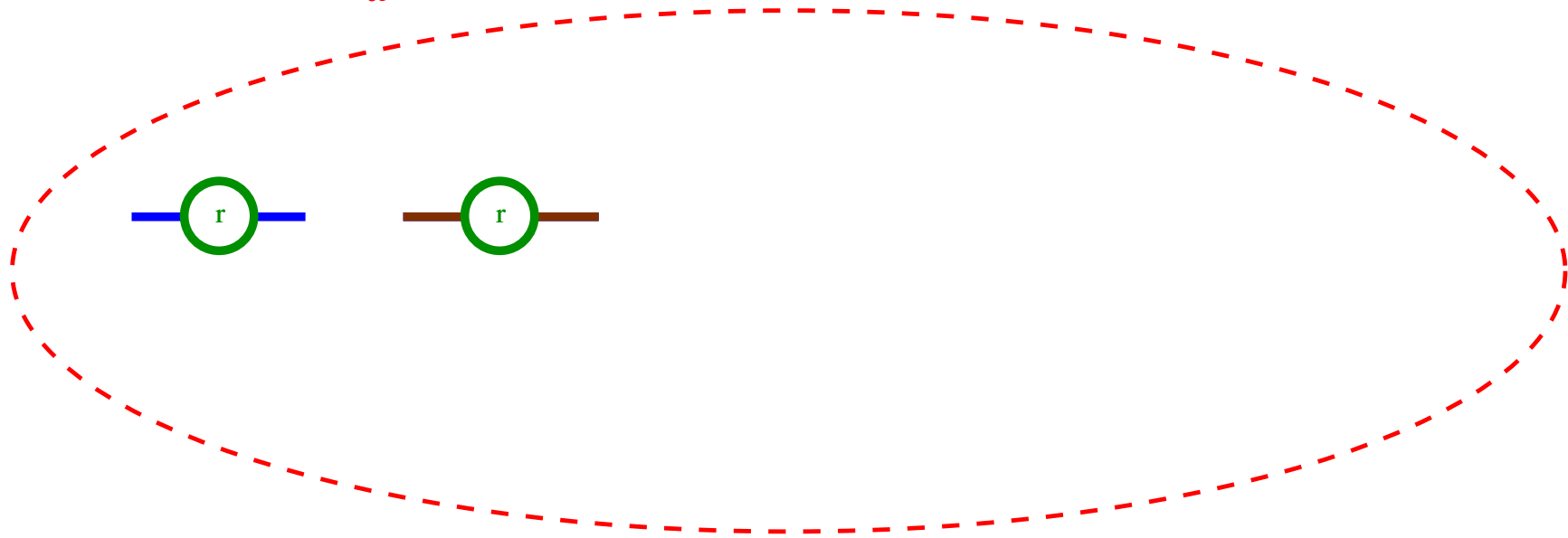
$\mathbb{C} \in \textit{Species}^*_\omega$

# Third case (II/III)
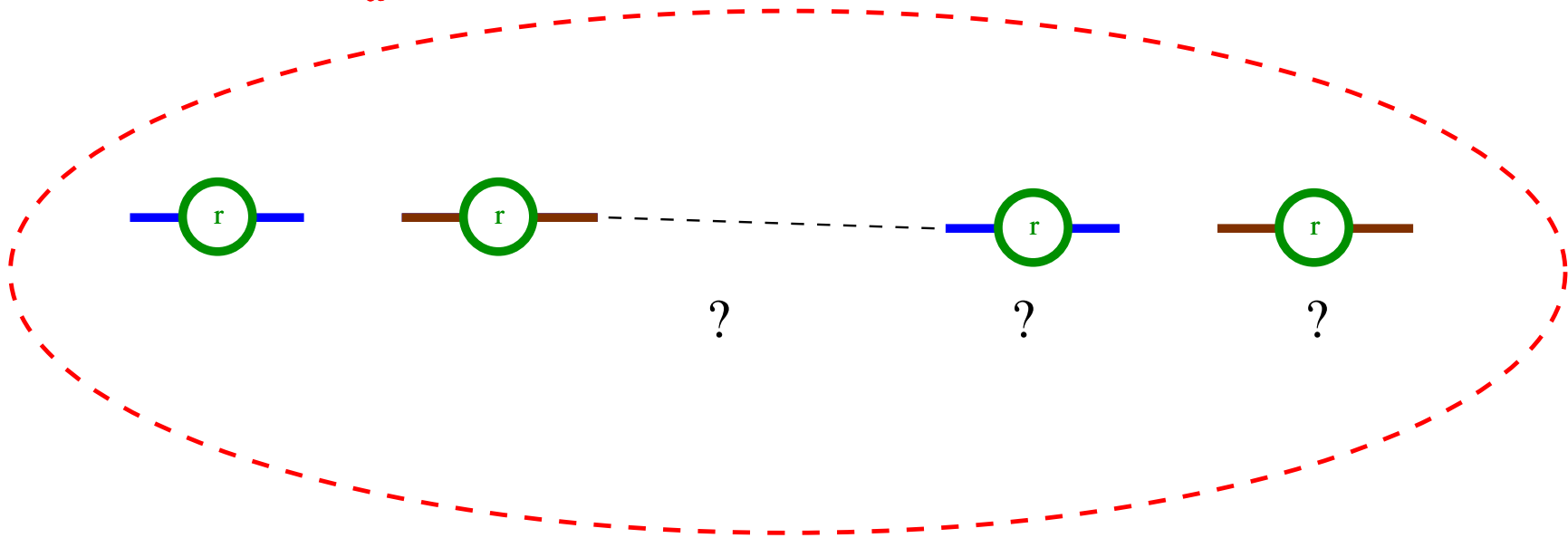


$C \in Species^*_\omega$

# Dangerous sites

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

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

$$Species_0 \triangleq R(a{\sim}u)$$

$$Rules \triangleq \left\{ \begin{array}{rcl} R(a{\sim}u) & \leftrightarrow & R(a{\sim}p) \\ R(a{\sim}u),R(a{\sim}u) & \to & R(a{\sim}u!1),R(a{\sim}u!1) \\ R(a{\sim}p),R(a{\sim}u) & \to & R(a{\sim}p!1),R(a{\sim}p!1) \\ R(a{\sim}p),R(a{\sim}p) & \to & R(a{\sim}p!1),R(a{\sim}p!1) \end{array} \right\}$$

$R(a{\sim}u!1),R(a{\sim}u!1) \in Species_\omega$

$R(a{\sim}p!1),R(a{\sim}p!1) \in Species_\omega$

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

# Non local systems

$$Species_0 \triangleq A(a{\sim}u), B(a{\sim}u)$$

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

$A(a{\sim}u!1), B(a{\sim}p!1) \in Species_\omega$
$A(a{\sim}p!1), B(a{\sim}u!1) \in Species_\omega$
But $A(a{\sim}p!1), B(a{\sim}p!1) \notin Species_\omega$.

# Non local systems

$$Species_0 \triangleq \text{A(a}\sim\text{u)}$$

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

A(a$\sim$u!1),A(a$\sim$p!1) $\in Species_\omega$
But A(a$\sim$p!1),A(a$\sim$p!1) $\notin Species_\omega$.

# 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)$\in Species_\omega$
But R(a!1,b!1) $\notin Species_\omega$.

# **Overview**

1. Introduction

2. Language: Kappa

3. Abstraction: Local views

4. Completeness: false positives?

5. Local fragment of Kappa

6. <span style="color:red">Decontextualization</span>

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(l!2,r),R1(l!1,r),E2(r!1),E1(r!2) → R2(l!3,r!1),R1(l!2,r!1),E2(r!2),E1(r!3)

Decontextualized rule:

R2(l!_,r),R1(l!_,r) → R2(l!_,r!1),R1(l!_,r!1)

We can remove redundant tests.

# Example

Initial rules:

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

Decontextualized rule:

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

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:

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

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

# An undecontextualizable rule

Initial rule:

Sh(Y7~u,pi!1),R(Y48~p!1,r!_) -> Sh(Y7~p,pi!1),R(Y48~p!1,r!_)

Decontextualized rule:

Sh(Y7~u,pi!1),R(Y48!1,r!_) -> Sh(Y7~p,pi!1),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)