Master AIV

Reachability Analysis of Rule-based Models

[ICCMSE'07,VMCAI'08]

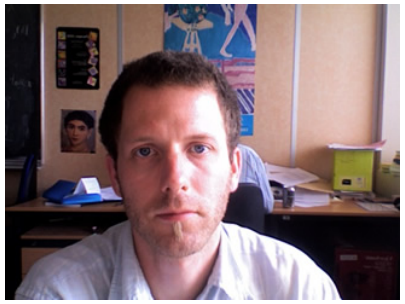
Jérôme Feret
Laboratoire d'Informatique de l'École Normale Supérieure
INRIA, ÉNS, CNRS, PSL⋆
http://www.di.ens.fr/~feret

2014, December the 12th

Joint-work with...

Walter Fontana
Harvard Medical School

Vincent Danos
ÉNS



Russ Harmer
ÉNS Lyon

Jean Krivine
Paris VII

Overview

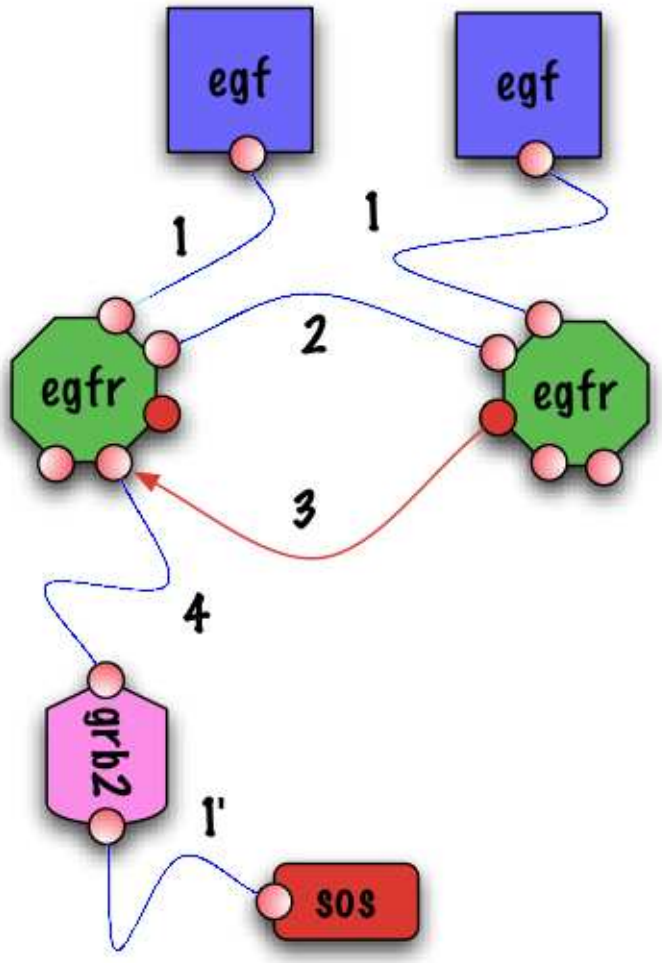
1. Introduction
2. Kappa language
3. Local views
4. Local set of chemical species
5. Local rule systems
6. Decontextualization
7. Conclusion

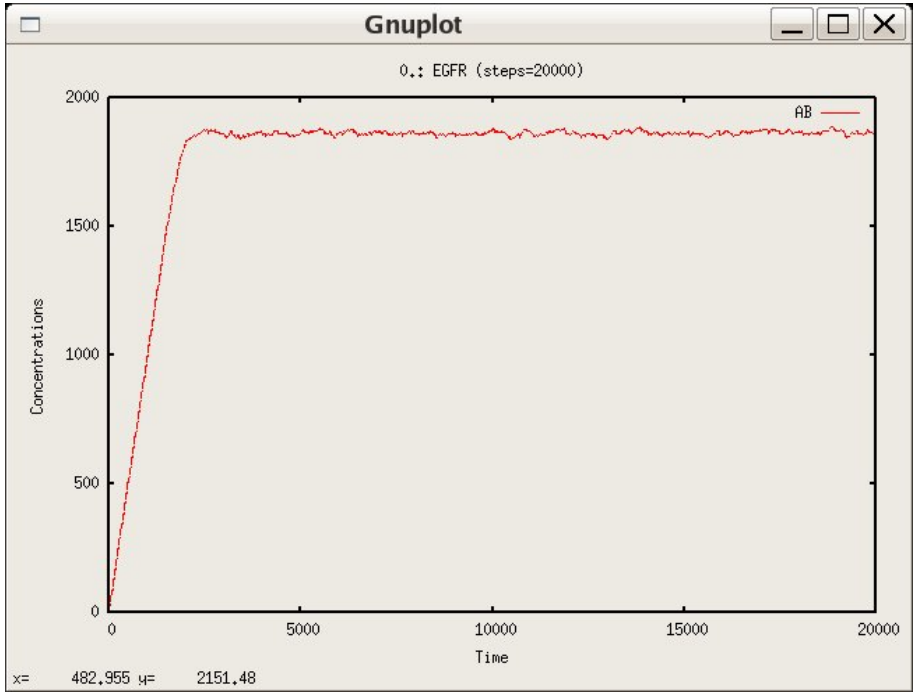
Modeling signaling pathway

- Signaling pathway:
  - A cell measures (i.e. checks thresholds, integrates, compares) the concentration of some proteins in order to make decisions.
  - Many proteins (enzymes, receptors, transport molecules) are involved.
  - They interact by binding with each other and activating each other.
- rule-based models:
  - A site graph-based rewrite language.
  - Description level matches with biologists' observation and manipulation level.
- Static analysis:

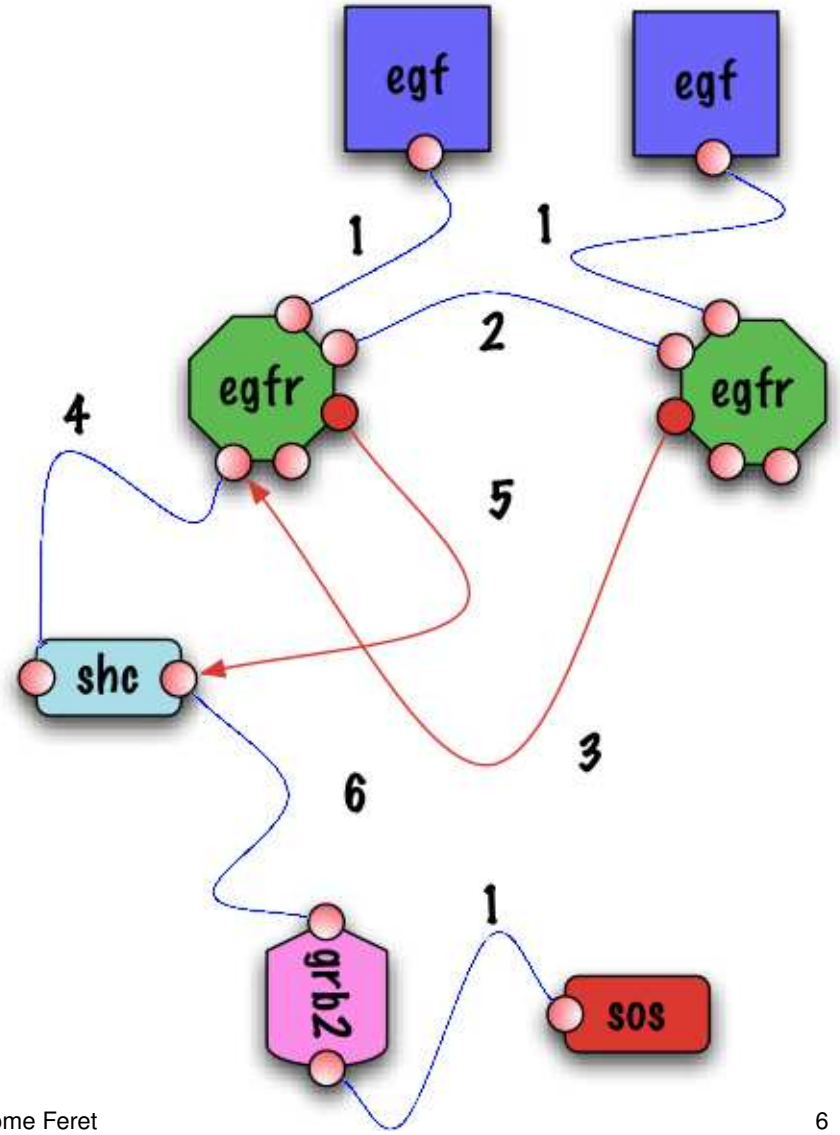
  We propose some static analysis tools in order to:
  - help the design of rule-based models;
  - compute (*abstract*) the properties of rule-based models.

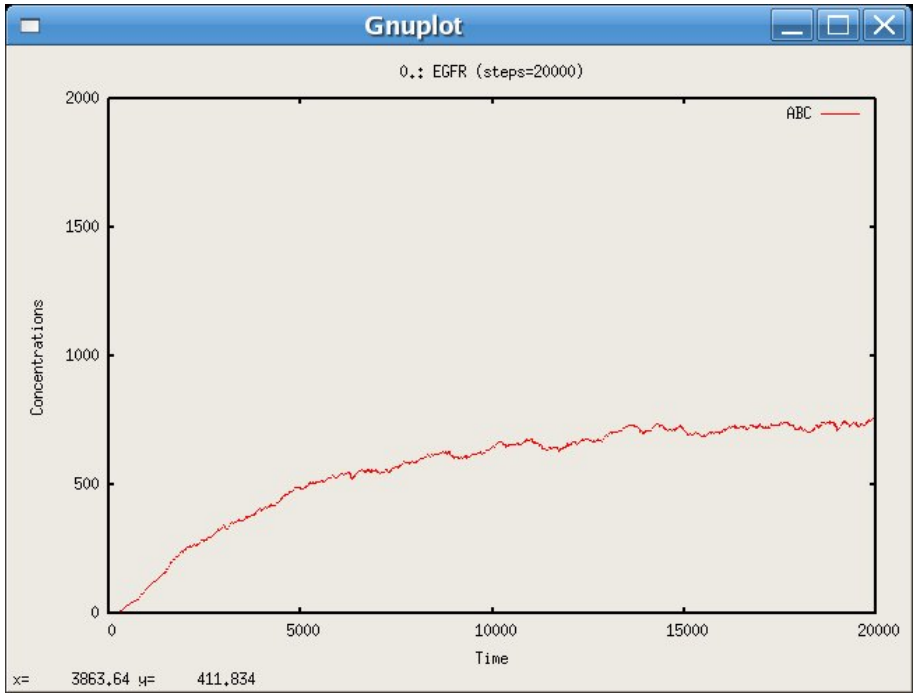
A single story





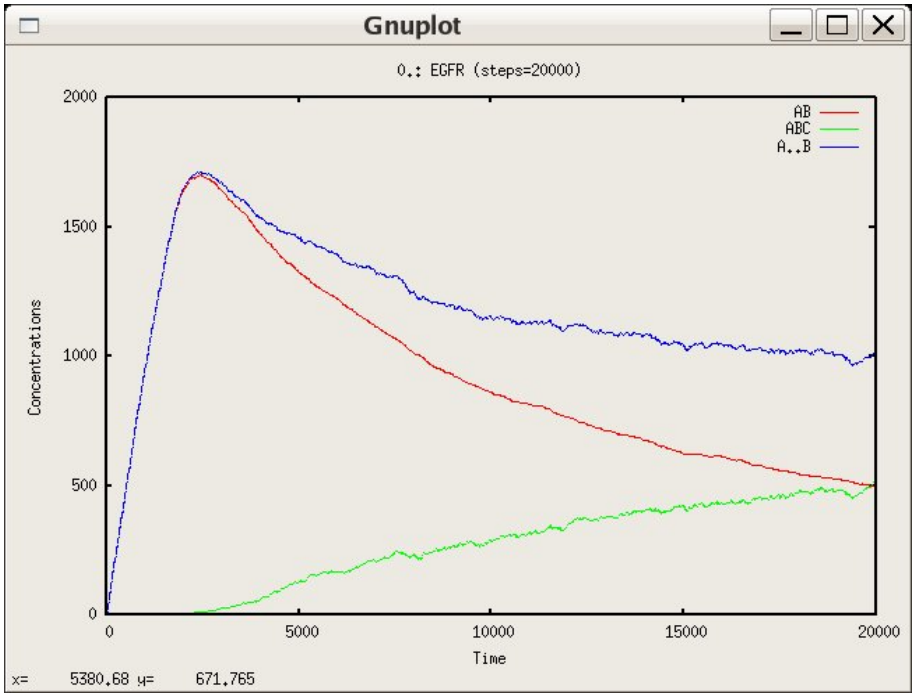
A concurrent story





Overshoot

When we combine the two stories. . .

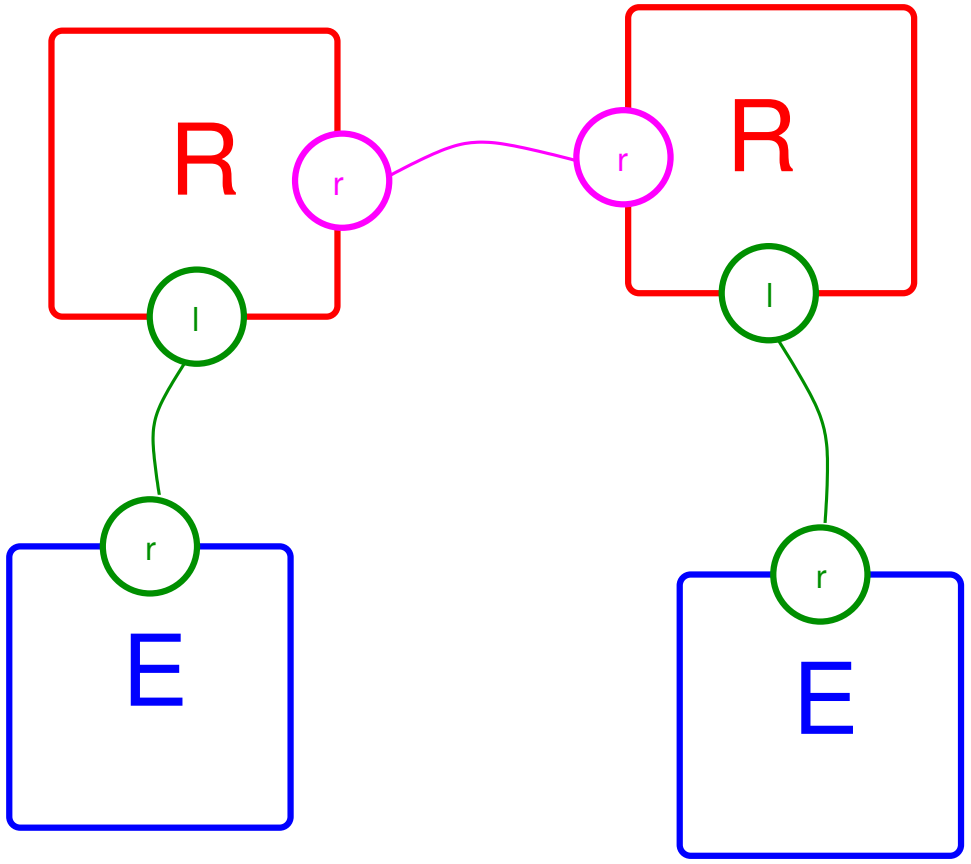


. . . we get an overshoot.

Overview

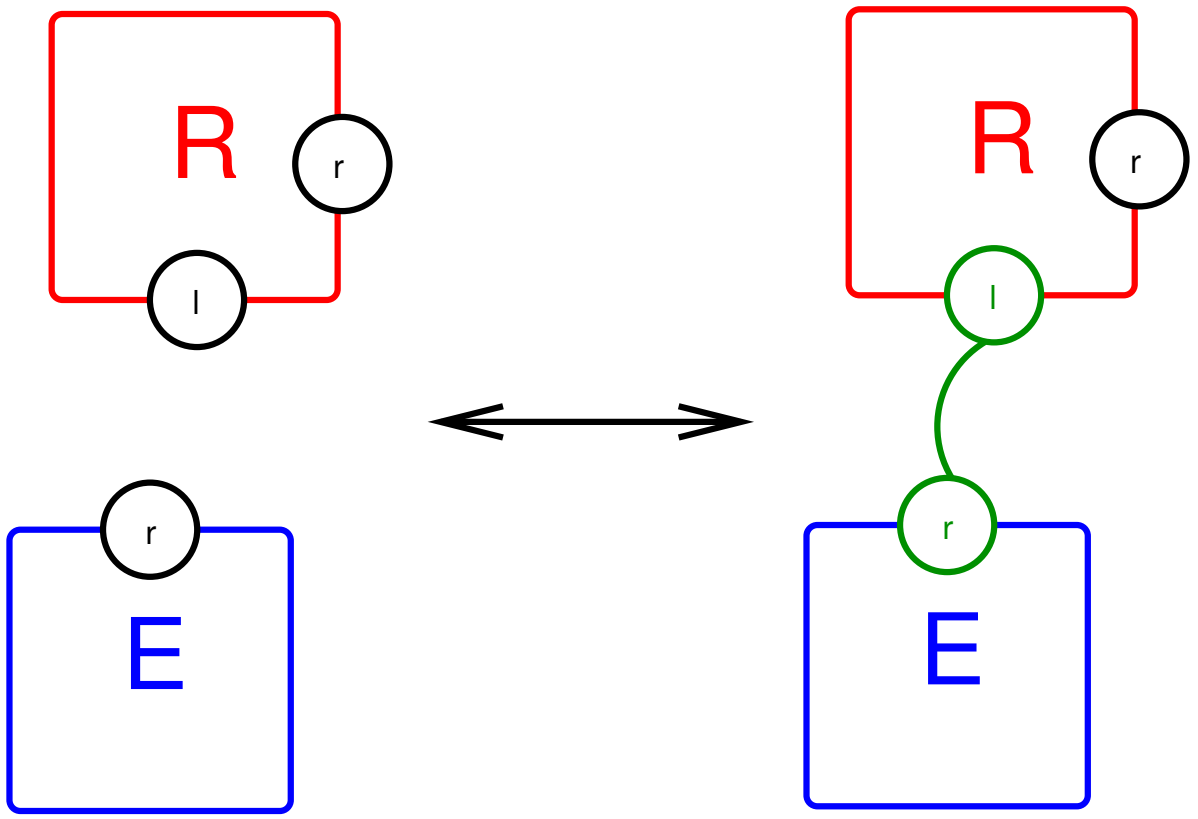
1. Introduction
2. Kappa language
3. Local views
4. Local set of chemical species
5. Local rule systems
6. Decontextualization
7. Conclusion

A chemical species



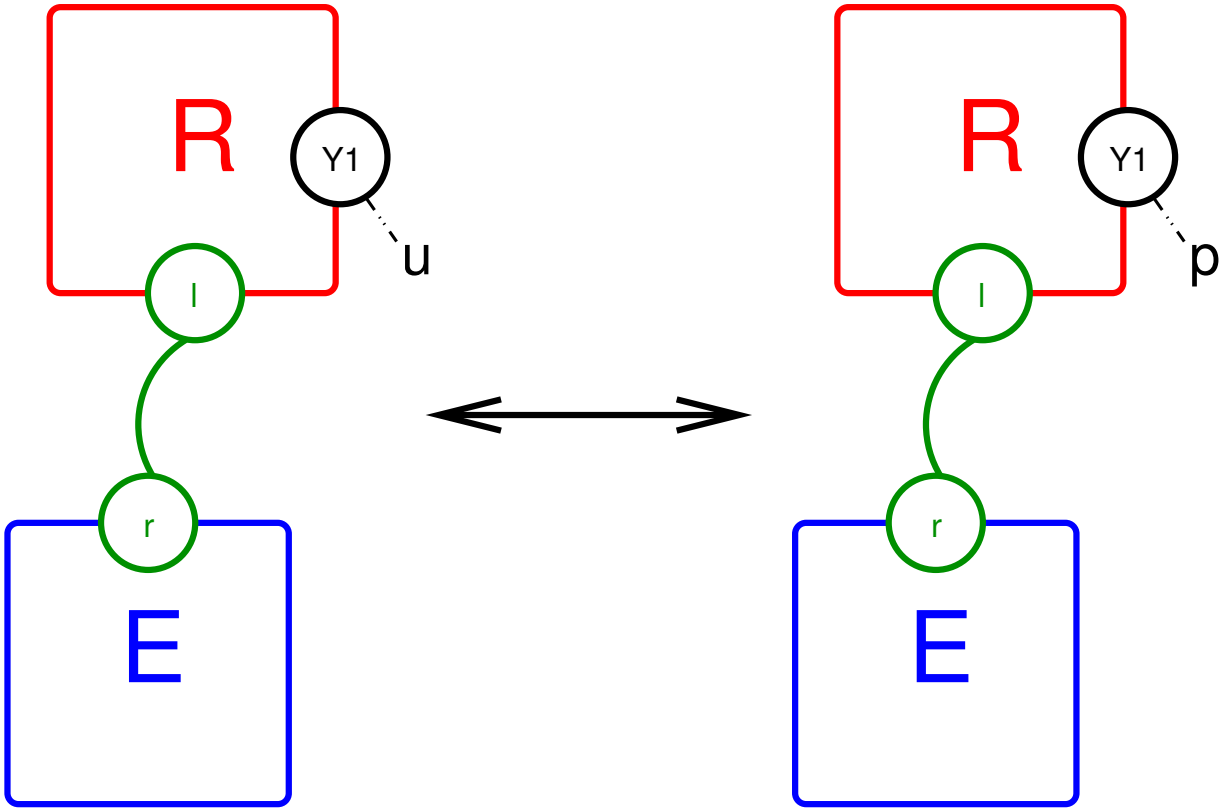
E(r!1), R(l!1,r!2), R(r!2,l!3), E(r!3)

A Unbinding/Binding Rule



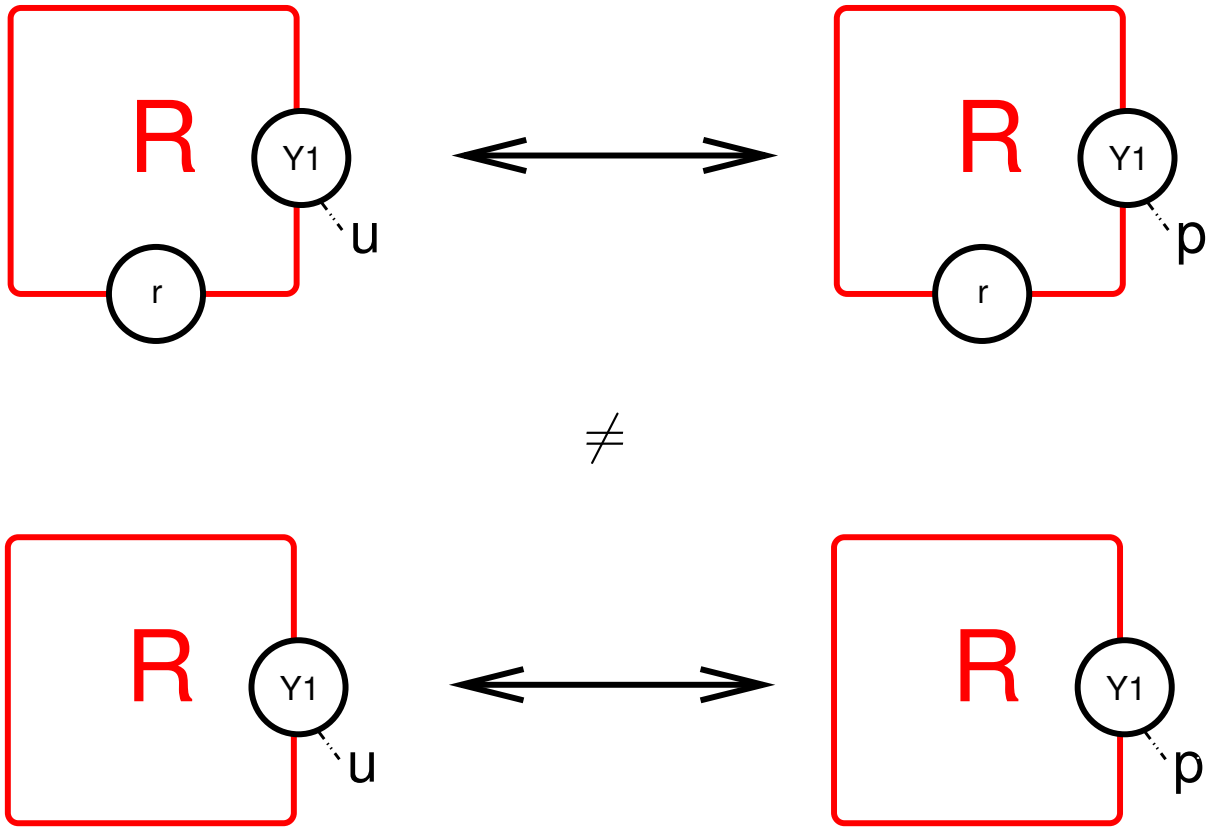
$$E(r),\ R(l,r) \longleftrightarrow E(r!1),\ R(l!1,r)$$

Internal state

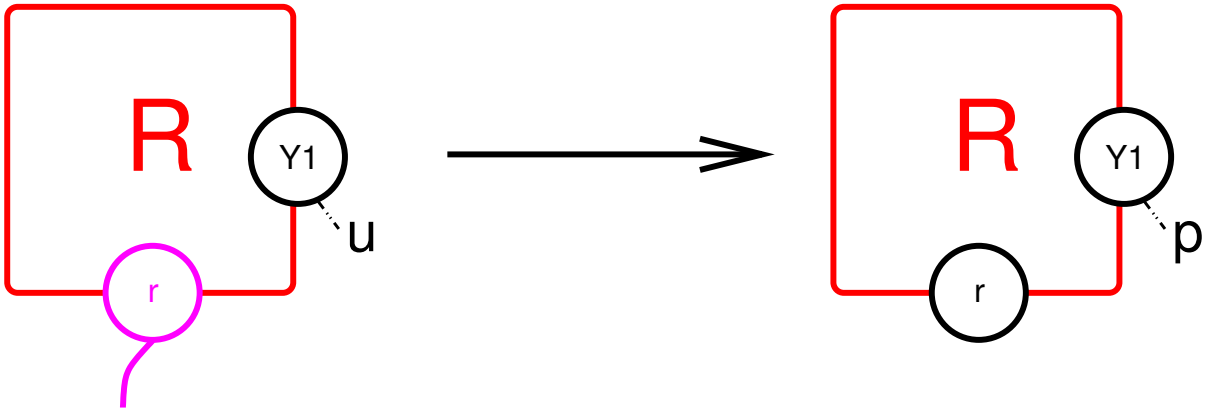


R(Y1~u,l!1), E(r!1) ⟷ R(Y1~p,l!1), E(r!1)

Don’t care, Don’t write

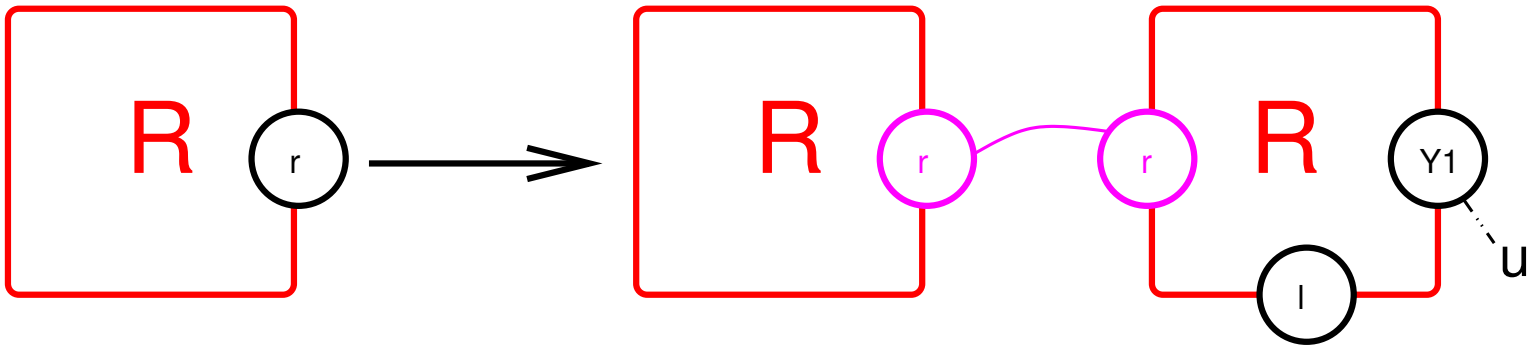


A contextual rule

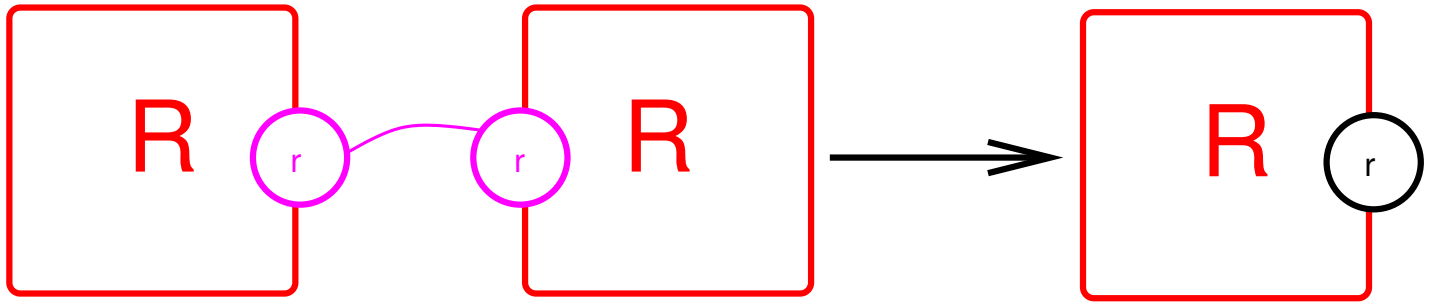


$$R(Y1{\sim}u,r!\_) \rightarrow R(Y1{\sim}p,r)$$

Creation/Suppression



$R(r) \rightarrow R(r!1), R(r!1,l,Y1)$



$R(r!1), R(r!1) \rightarrow R(r)$

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, π:=PTB/SH2

- Ligand-receptor binding, receptor dimerisation, rtk x-phosph, & de-phosph
  - 01: R(l,r), E(r) <-> R(l¹,r), E(r¹)
  - 02: R(l¹,r), R(l²,r) <-> R(l¹,r³), R(l²,r³)
  - 03: R(r¹,Y68) -> R(r¹,Y68ᵖ)
  - R(Y68ᵖ) -> R(Y68)
  - 04: R(r¹,Y48) -> R(r¹,Y48ᵖ)
  - R(Y48ᵖ) -> R(Y48)

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

- Sh x-phosph & de-phosph
  - 14: R(r²,Y48ᵖ¹), Sh(π¹,Y7) -> R(r²,Y48ᵖ¹), Sh(π¹,Y7ᵖ)
  - ??: Sh(π¹,Y7ᵖ) -> Sh(π¹,Y7)
  - 16: Sh(π,Y7ᵖ) -> Sh(π,Y7)

refined from Sh(Y7ᵖ)-> Sh(Y7)

- Y68-G binding
  - 09: R(Y68ᵖ), G(a,b) <-> R(Y68ᵖ¹)+G(a¹,b)
  - 11: R(Y68ᵖ), G(a,b²) <-> R(Y68ᵖ¹)+G(a¹,b²)

refined from R(Y68ᵖ)+G(a)<->R(Y68ᵖ¹)+G(a¹)

Early EGF example

egf rules 2

refined from
So(d)+G(b)<->So(d¹)+G(b¹)

interface note: highlight the interacting parts

- G-So binding
  - 10: R(Y68^{p1}), G(a¹,b), So(d) <-> R(Y68^{p1}), G(a¹,b²), So(d²)
  - 12: G(a,b), So(d) <-> G(a,b¹), So(d¹)
  - 22: Sh(π,Y7^{p2}), G(a²,b), So(d) <-> Sh(π,Y7^{p2}), G(a²,b¹), S(d¹)
  - 19: Sh(π¹,Y7^{p2}), G(a²,b), So(d) <-> Sh(π¹,Y7^{p2}), G(a²,b¹), S(d¹)
- Y48-Sh binding
  - 13: R(Y48^{p}), Sh(π,Y7) <-> R(Y48^{p1}), Sh(π¹,Y7)
  - 15: R(Y48^{p}), Sh(π,Y7^{p}) <-> R(Y48^{p1}), Sh(π¹,Y7^{p})
  - 18: R(Y48^{p}), Sh(π,Y7^{p1}), G(a¹,b) <-> R(Y48^{p2}), Sh(π²,Y7^{p1}), G(a¹,b)
  - 20: R(Y48^{p}), Sh(π,Y7^{p1}), G(a¹,b³), S(d³) <-> R(Y48^{p2}), Sh(π²,Y7^{p1}), G(a¹,b³), S(d³)

refined from
R(Y48^{p})+Sh(π)<->R(Y48^{p1})+Sh(π¹)

why not simply G(b³)??

- Sh-G binding
  - 17: R(Y48^{p1}), Sh(π¹,Y7^{p}), G(a,b) <-> R(Y48^{p1}), Sh(π¹,Y7^{p2}), G(a²,b)
  - 21: Sh(π,Y7^{p}), G(a,b) <-> Sh(π,Y7^{p1}), G(a¹,b)
  - 23: Sh(π,Y7^{p}), G(a,b²) <-> Sh(π,Y7^{p1}), G(a¹,b²)
  - 24: R(Y48^{p1}), Sh(π¹,Y7^{p}), G(a,b³), S(d³) <-> R(Y48^{p1}), Sh(π¹,Y7^{p2}), G(a², b³), S(d³)

refined from
Sh(π), G(a)<->Sh(π¹), G(a¹)

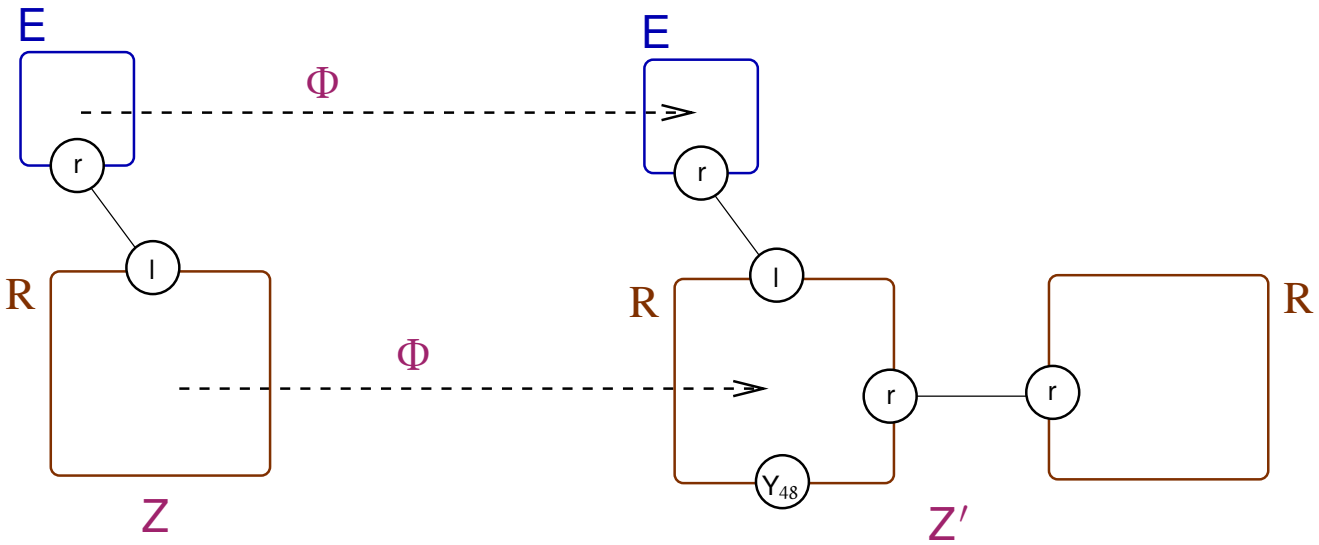
Properties of interest

1. Show the absence of modelling 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. Kappa language
3. Local views
4. Local set of chemical species
5. Local rule systems
6. Decontextualization
7. Conclusion

Embedding



We write $Z \lhd_\Phi Z'$ iff:

- Φ is a site-graph morphism:
  - i is less specific than $\Phi(i)$,
  - if there is a link between (i, s) and (i', s'), then there is a link between $(\Phi(i), s)$ and $(\Phi(i'), s')$.
- Φ is an into map (injective):
  - $\Phi(i) = \Phi(i')$ implies that $i = i'$.

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 write:

$$c_1, \ldots, c_m \rightarrow_{R_k} c'_1, \ldots, c'_n$$

whenever:

1. there is an embedding of the lhs of R_k in the solution $c_1, \ldots, c_m$;
2. the (embedding/rule) produces the solution $c'_1, \ldots, c'_n$.

We are interested in $Species_\omega$ the set of all chemical species that can be constructed in one or several applications of 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

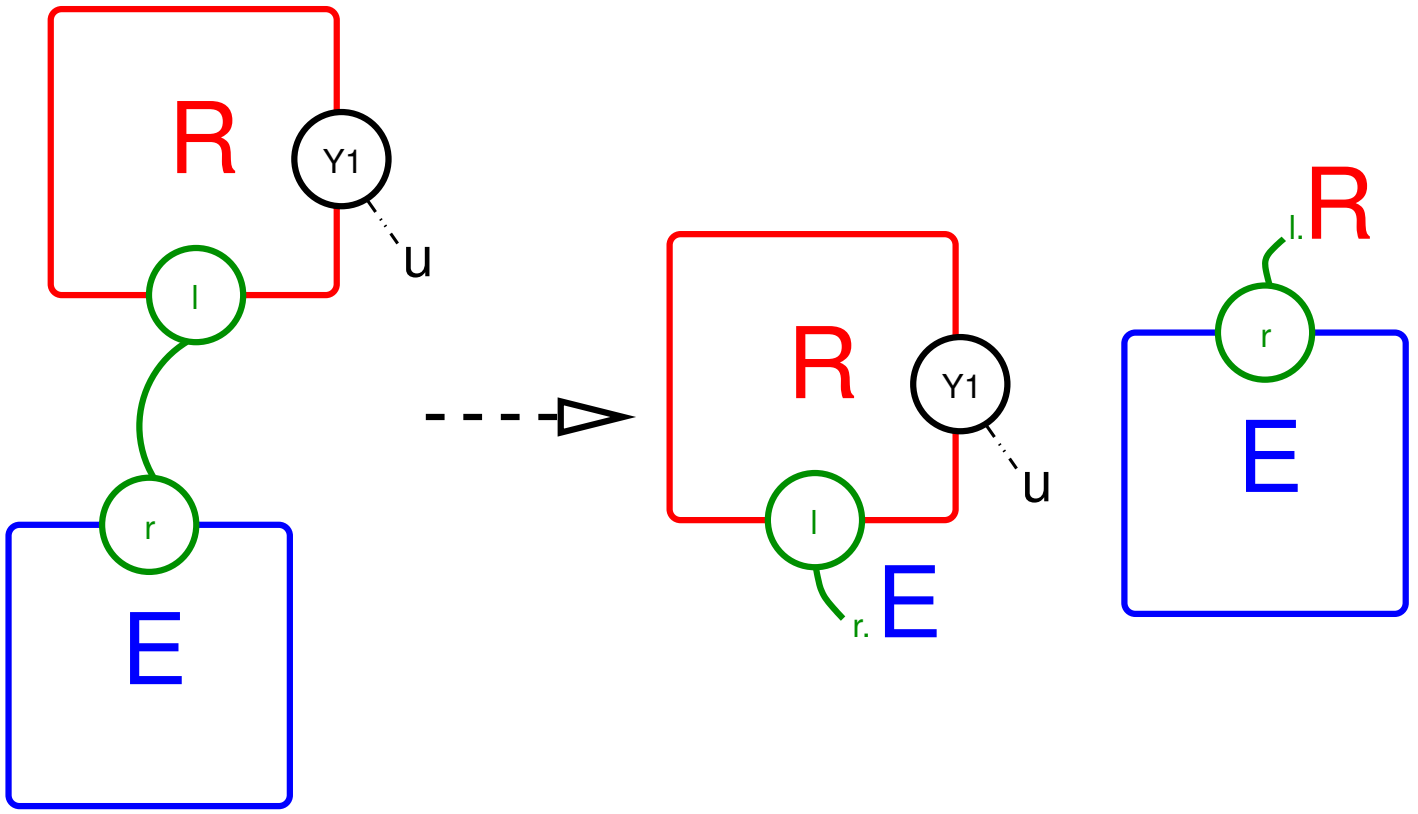
We define the mapping $\mathbb{F}$ as follows:

$$\mathbb{F} : \begin{cases} \wp(\mathit{Species}) & \to \wp(\mathit{Species}) \\ X & \mapsto X \cup \left\{ c'_j \;\middle|\; \begin{array}{l} \exists R_k \in \mathcal{R}, c_1, \ldots, c_m \in X, \\ c_1, \ldots, c_m \to_{R_k} c'_1, \ldots, c'_n \end{array} \right\}. \end{cases}$$

We define the set of reachable chemical species as follows:

$$\mathit{Species}_\omega = \bigcup \left\{ \mathbb{F}^n(\mathit{Species}_0) \;\middle|\; n \in \mathbb{N} \right\}.$$

Local views



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

Galois connexion

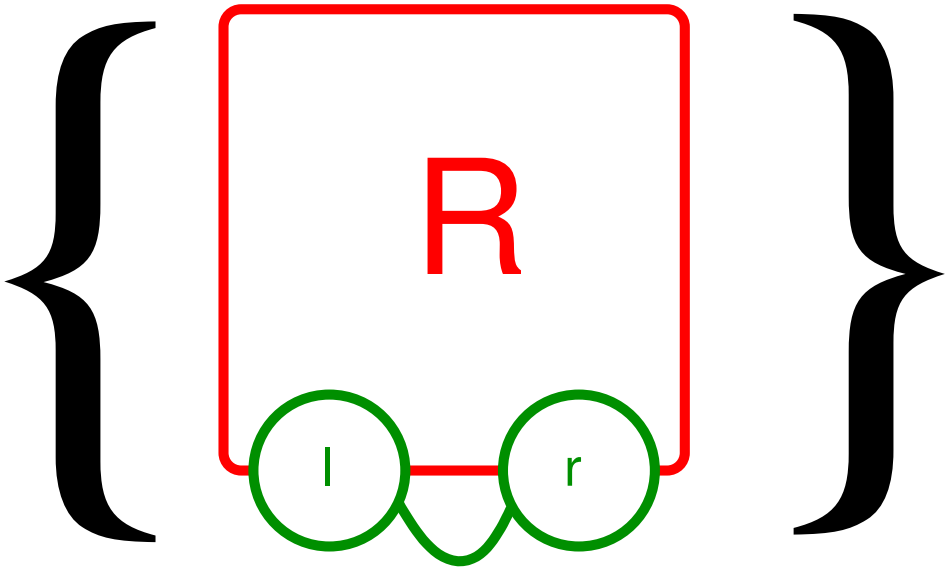
- Let $\mathit{Local\_view}$ be the set of all local views.
- Let $\alpha \in \wp(\mathit{Species}) \rightarrow \wp(\mathit{Local\_view})$ be the function that maps any set of chemical species into the set of their local views.
- Let $\gamma \in \wp(\mathit{Local\_view}) \rightarrow \wp(\mathit{Species})$ be the function that maps any set of local views into the set of chemical species that can be built with these local views.
- The pair (α, γ) forms a Galois connexion:

$$\wp(\mathit{Species}) \underset{\alpha}{\overset{\gamma}{\leftrightarrows}} \wp(\mathit{Local\_view}).$$

$\gamma \circ \alpha$

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

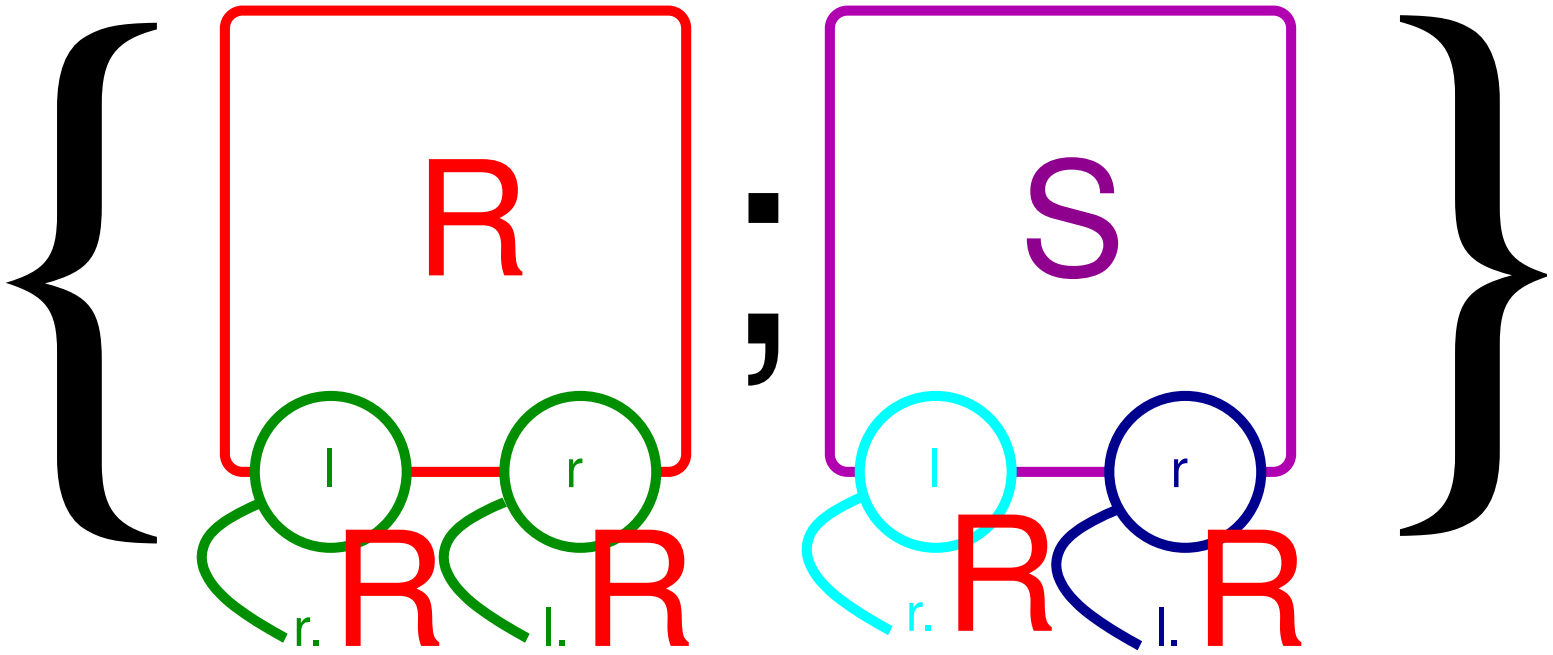
Guess the image of the following set of chemical species ?



$\alpha \circ \gamma$

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

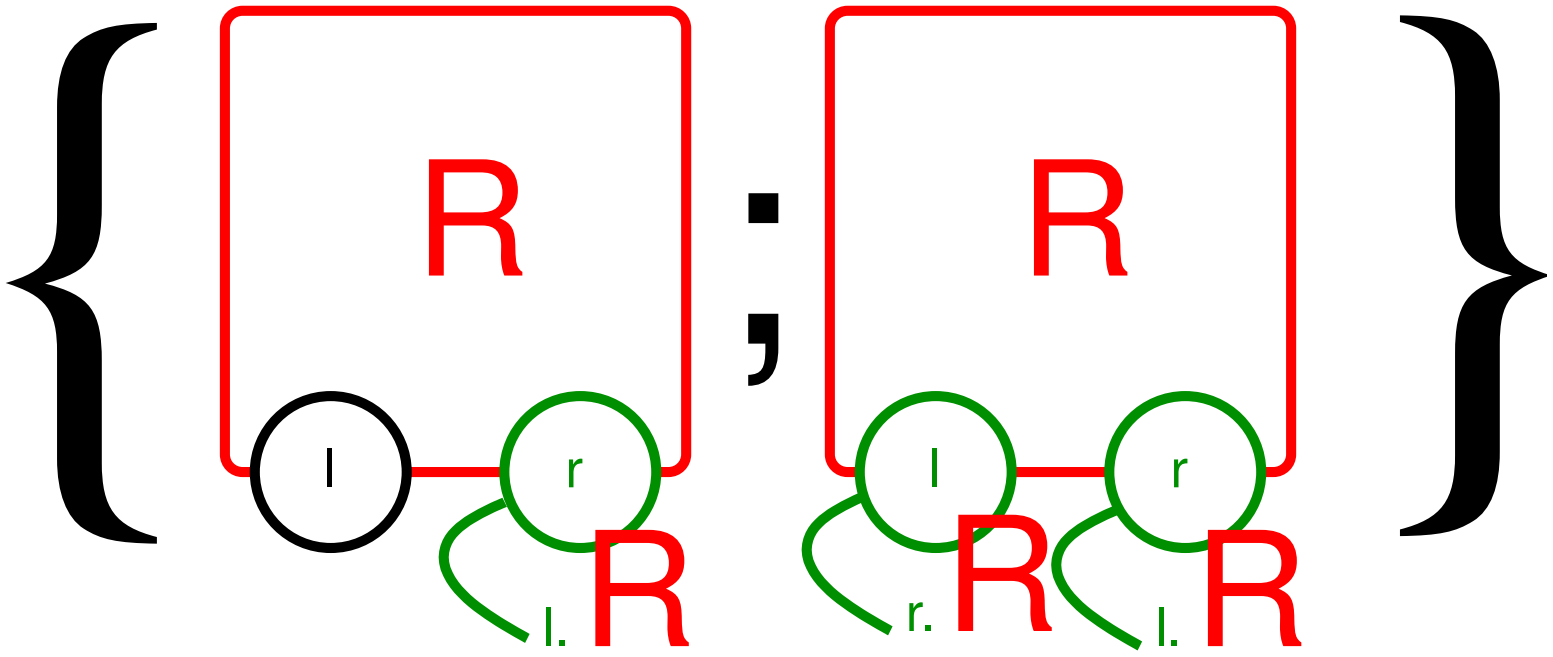
Guess the image of the following set of local views ?



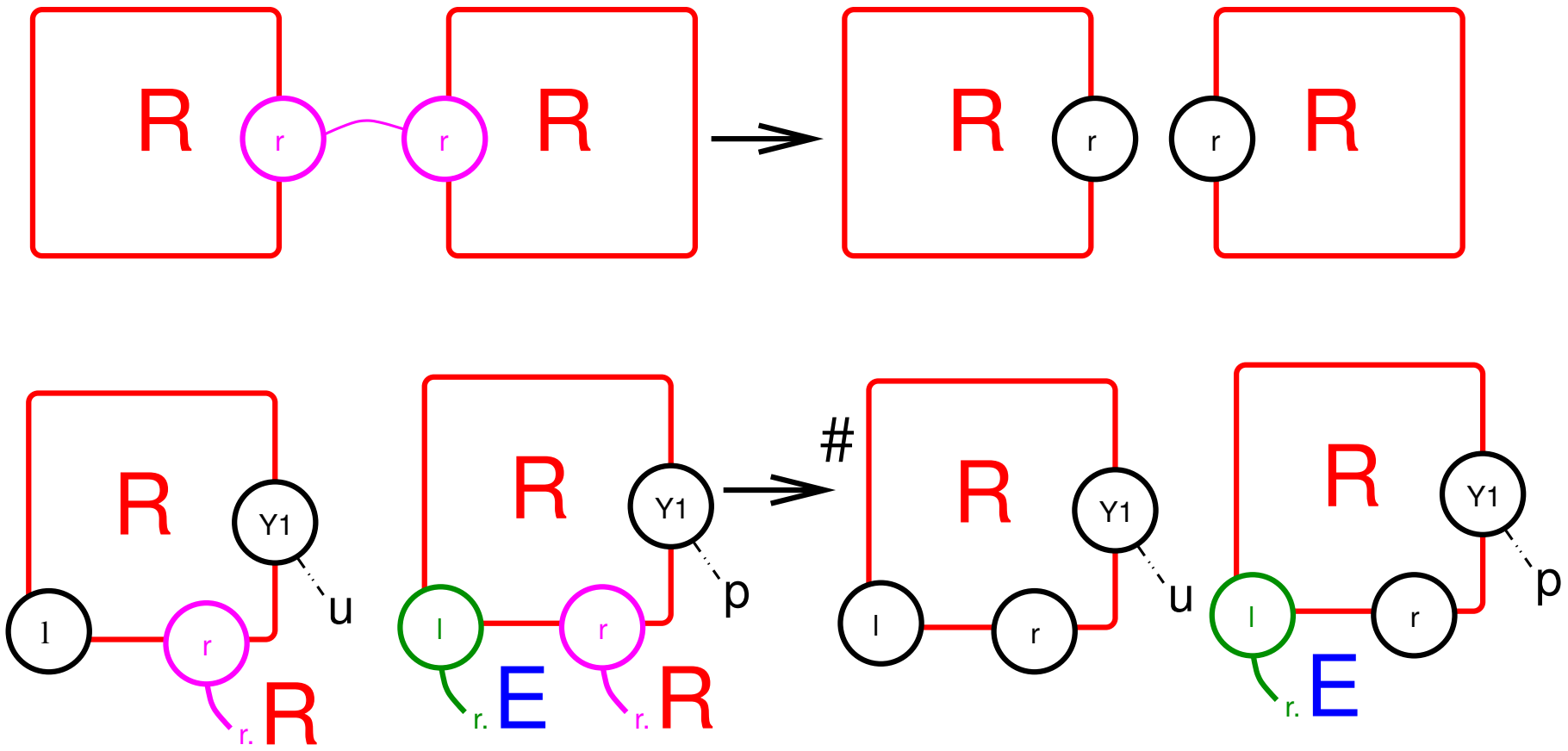
One more question

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

Guess the image of the following set of local views ?



Abstract rules



Abstract counterpart to $\mathbb{F}$

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

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

Theorem 1 (soundness) It follows that:

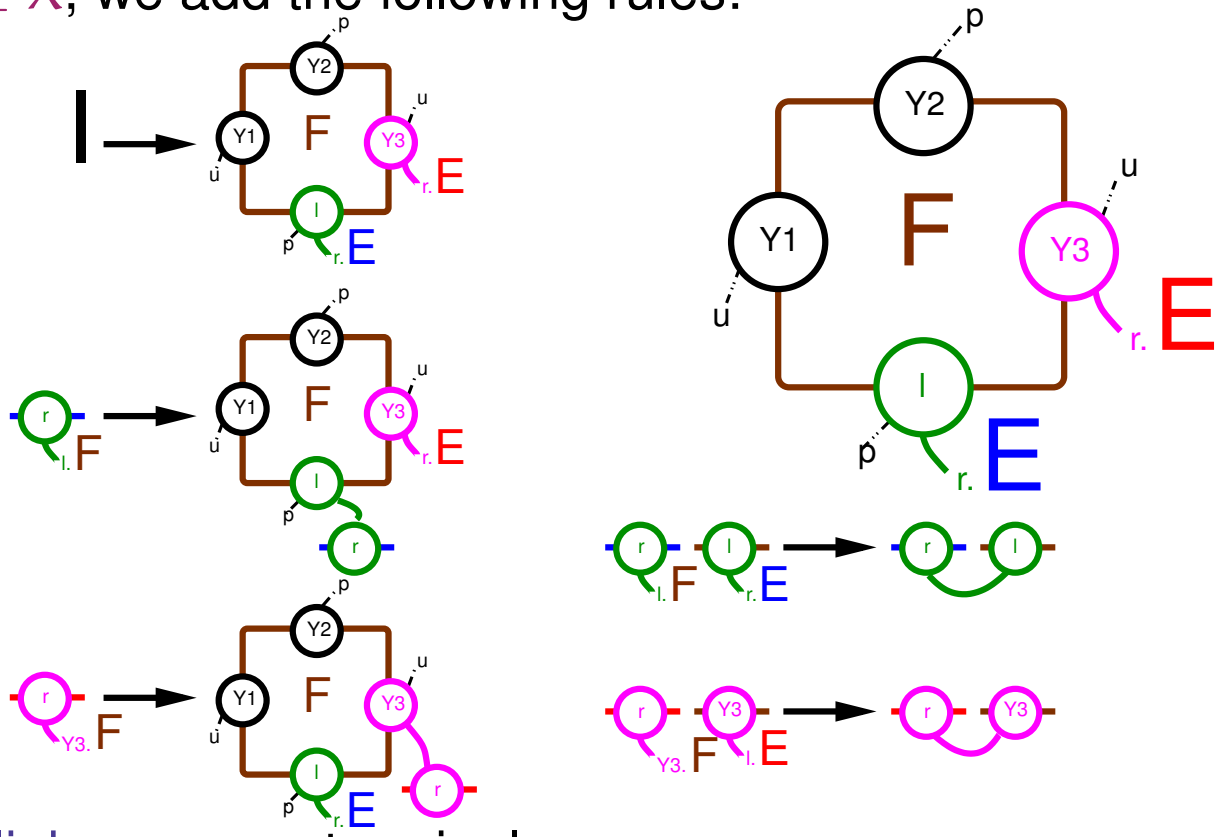
1. both $\mathit{lfp}_{x_0}\mathbb{F}$ and $\mathit{lfp}_{\alpha(x_0)}\mathbb{F}^\sharp$ exist,
2. $\mathit{lfp}_{x_0}\mathbb{F} \subseteq \gamma(\mathit{lfp}_{\alpha(x_0)}\mathbb{F}^\sharp)$.

Overview

1. Introduction
2. Kappa language
3. Local views
4. Local set of chemical species
5. Local rule systems
6. Decontextualization
7. Conclusion

Concretization

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:

     R.l-r.E ... 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 $X \in \alpha(\wp(\mathit{Species}))$ 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

Which information is abstracted away ?

Our analysis is exact (no false positive):

- for EGF cascade (356 chemical species);
- for FGF cascade (79080 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 ?

Local set of chemical species

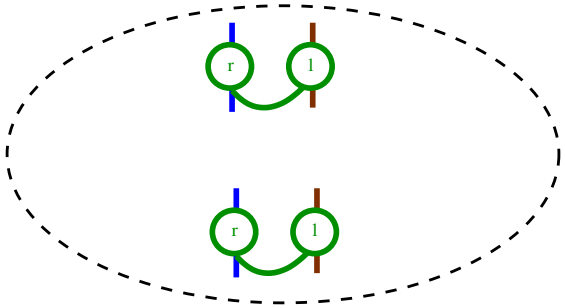
Definition 1 We say that a set $X \in \wp(Species)$ of chemical species is local if and only if $X \in \gamma(\wp(Local\_view))$.

(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.)

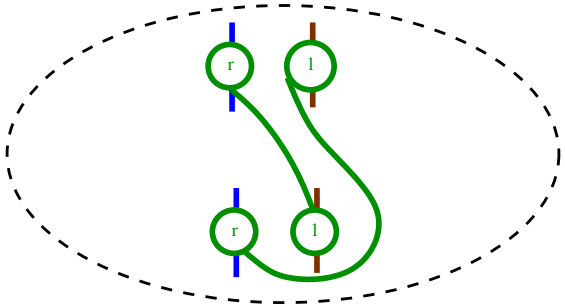
Swapping relation

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 2 Let $X \in \wp(\mathit{Species})$ be a set of chemical species.
The two following assertions are equivalent:

1. the set $X \in \wp(\mathit{Species})$ is local;
2. for any tuples $(C_i), (D_j) \in \mathit{Species}^*$ such that:
   - $(C_i) \in X^*$,
   - and $(C_i) \overset{\text{SWAP}}{\sim} (D_j)$;

   we have $(D_j) \in X^*$.

Consequences

Theorem 3 (completeness) It follows that:

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

Overview

1. Introduction
2. Kappa language
3. Local views
4. Local set of chemical species
5. Local rule systems
6. Decontextualization
7. Conclusion

Outline

We have proved that:

- if the set $\mathit{Species}_\omega$ of reachable chemical species is close with respect swapping $\overset{\text{SWAP}}{\sim}$,
- then the reachability analysis is exact (i.e. $\mathit{Species}_\omega = \gamma(\mathit{lfp}_{\alpha(\mathit{Species}_0)}\mathbb{F}^\sharp)$).

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.

Non local systems

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

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

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

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

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

Non local systems

$$\mathit{Species}_0 \triangleq \text{A(a}\sim\text{u),B(a}\sim\text{u)}$$

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

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

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

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

Non local systems

$$Species_0 \triangleq A(a\sim u)$$

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

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

Non local systems

$$\mathit{Species}_0 \triangleq \text{R(a,b)}$$
$$\mathit{Rules} \triangleq \{\ \text{R(a,b),R(a)} \rightarrow \text{R(a,b!1),R(a!1)}\}$$

R(a,b!2),R(a!2,b!1),R(a!1,b)$\in \mathit{Species}_\omega$
But R(a!1,b!1) $\notin \mathit{Species}_\omega$.

Overview

1. Introduction
2. Kappa language
3. Local views
4. Local set of chemical species
5. Local rule systems
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(l!2,r),R1(l!1,r),E2(r!1),E1(r!2) $\rightarrow$ R2(l!3,r!1),R1(l!2,r!1),E2(r!2),E1(r!3)

Decontextualized rule:

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

We can remove redundant tests.

Example

Initial rules:

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

Decontextualized rule:

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

We can remove exhaustive enumerations.

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. Kappa language
3. Local views
4. Local set of chemical species
5. Local rule systems
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)