

Formal Methods in Systems and Synthetic Biology

François Fages
Constraint Programming Group
INRIA Paris-Rocquencourt

`mailto:Francois.Fages@inria.fr`

`http://contraintes.inria.fr`

Need for Abstractions in Systems Biology

Models are built in Systems Biology with two contradictory perspectives :

1) **Models for representing knowledge** : **the more concrete the better**

detailed mechanistic reaction models (SBML), gene ontologies, protein functions, protein interactions, structures ...

2) **Models for making predictions** : **the more abstract the better.**

schematic reaction models (SBML), variable elimination, approximations, stationary states, influence graph ...

These perspectives can be reconciled by organizing **models and formalisms** in **abstraction hierarchies**.

“To understand a system is not to know everything about it but to know abstraction levels that are sufficient for answering questions about it”

Overview of the Lectures

1. Introduction
2. Rule-based modeling in Biocham
3. Temporal logic constraints in Biocham
4. Conclusion
5. *Killing lecture*: Abstract interpretation in systems biology
 - Theory of abstract interpretation
 - Domain of reaction rule models
 - Hierarchy of semantics: stochastic, discrete and boolean traces
 - Analyses by type checking/type inference: dimensions, protein functions, influence graphs, compartment topology

F. F., S. Soliman, Abstract Interpretation in Systems Biology, *Theoretical Computer Science* 2008.

Theory of Abstract Interpretation I: Domains

Simple algebraic theory of abstraction introduced by [Cousot Cousot 77] to reason about programs.

In this setting, a (computation) **domain** is a lattice $\mathcal{D}(\sqsubseteq, \perp, \top, \sqcup, \sqcap)$ where \sqsubseteq is the “less coarse” information ordering.

Often just a power-set $\mathcal{P}(\mathcal{S})(\subseteq, \emptyset, \mathcal{S}, \cup, \cap)$ ordered by set inclusion.

Theory of Abstract Interpretation I: Domains

Simple algebraic theory of abstraction introduced by [Cousot Cousot 77] to reason about programs.

In this setting, a (computation) **domain** is a lattice $\mathcal{D}(\sqsubseteq, \perp, \top, \sqcup, \sqcap)$ where \sqsubseteq is the “less coarse” information ordering.

Often just a power-set $\mathcal{P}(\mathcal{S})(\subseteq, \emptyset, \mathcal{S}, \cup, \cap)$ ordered by set inclusion.

Given a finite set \mathcal{M} of molecule names, the universe of reaction rules is the set

$$\mathcal{R} = \{e \text{ for } S \Rightarrow S' \mid \begin{array}{l} e \text{ is a kinetic expression,} \\ \text{and } S \text{ and } S' \text{ are multisets of molecules in } \mathcal{M}. \end{array}\}$$

Def. 1 *The domain of Biocham reaction models is $\mathcal{C}_{\mathcal{R}} = (\mathcal{P}(\mathcal{R}), \subseteq)$.*

$\perp = \emptyset$ is the empty model, $\top = \{\mathcal{R}\}$ is the universal model.

Theory of Abstract Interpretation II: Abstractions

Def. 2 A *Galois connection* $\mathcal{C} \rightarrow_{\alpha} \mathcal{A}$ between two lattices \mathcal{C} and \mathcal{A} is defined by an abstraction function $\alpha : \mathcal{C} \rightarrow \mathcal{A}$ and a concretization function $\gamma : \mathcal{A} \rightarrow \mathcal{C}$ which are *monotonic*:

- $\forall c, d \in \mathcal{C} \ c \sqsubseteq_{\mathcal{C}} d \Rightarrow \alpha(c) \sqsubseteq_{\mathcal{A}} \alpha(d),$
- $\forall a, b \in \mathcal{A} \ a \sqsubseteq_{\mathcal{A}} b \Rightarrow \gamma(a) \sqsubseteq_{\mathcal{C}} \gamma(b),$

$\gamma \circ \alpha$ is extensive and represents the information lost by the abstraction:

- $\forall c \in \mathcal{C} \ c \sqsubseteq_{\mathcal{C}} \gamma \circ \alpha(c),$

$\alpha \circ \gamma$ is contracting:

- $\forall a \in \mathcal{A} \ \alpha \circ \gamma(a) \sqsubseteq_{\mathcal{A}} a.$

If $\gamma \circ \alpha$ is the identity, the abstraction α loses no information, and \mathcal{C} and \mathcal{A} are isomorphic from the information standpoint (although α may be not onto and γ not one-to-one).

Properties of Galois Connections

Let $\downarrow a = \{b \mid b \sqsubseteq a\}$ and $\uparrow a = \{b \mid a \sqsubseteq b\}$.

1. α, γ are adjoint functors: $\forall c \in \mathcal{C}, \forall a \in \mathcal{A} : c \sqsubseteq_{\mathcal{C}} \gamma(a) \Leftrightarrow \alpha(c) \sqsubseteq_{\mathcal{A}} a$.
2. $\gamma(a) = \max \alpha^{-1}(\downarrow a) = \sqcup \alpha^{-1}(\downarrow a)$
3. $\alpha(c) = \min \gamma^{-1}(\uparrow c) = \sqcap \gamma^{-1}(\uparrow c)$ item $\gamma \circ \alpha$ is the identity *iff* γ is onto *iff* α is one-to-one.
4. α preserves \sqcup , and γ preserves \sqcap ;

It is equivalent in the definition of Galois connections to replace the conditions of extensivity and contraction by adjointness 1, or by condition 2 which also entails the monotonicity of γ .

Pointwise Galois Connections between Powersets

Lemma 3 *Let \mathcal{C} and \mathcal{A} be two sets, and $\alpha : \mathcal{P}(\mathcal{C}) \longrightarrow \mathcal{P}(\mathcal{A})$ be a function such that $\alpha(c) = \bigcup_{e \in c} \alpha(\{e\})$.*

Then the function $\gamma(a) = \bigcup \alpha^{-1}(\downarrow a)$ forms a Galois connection $\mathcal{P}(\mathcal{C}) \xrightleftharpoons[\gamma]{\alpha} \mathcal{P}(\mathcal{A})$ between $(\mathcal{P}(\mathcal{C}), \subseteq)$ and $(\mathcal{P}(\mathcal{A}), \subseteq)$.

Pointwise Galois Connections between Powersets

Lemma 3 *Let \mathcal{C} and \mathcal{A} be two sets, and $\alpha : \mathcal{P}(\mathcal{C}) \longrightarrow \mathcal{P}(\mathcal{A})$ be a function such that $\alpha(c) = \bigcup_{e \in c} \alpha(\{e\})$.*

Then the function $\gamma(a) = \bigcup \alpha^{-1}(\downarrow a)$ forms a Galois connection $\mathcal{P}(\mathcal{C}) \xrightleftharpoons[\gamma]{\alpha} \mathcal{P}(\mathcal{A})$ between $(\mathcal{P}(\mathcal{C}), \subseteq)$ and $(\mathcal{P}(\mathcal{A}), \subseteq)$.

PROOF: We show that α is monotonic and $\gamma(a) = \max \alpha^{-1}(\downarrow a)$.

The monotonicity of α is immediate since if $c \subseteq c'$ we have

$$\bigcup_{c_i \in c} \alpha(\{c_i\}) \subseteq \bigcup_{c_i \in c'} \alpha(\{c_i\}).$$

...



Pointwise Galois Connections between Powersets

Lemma 3 *Let \mathcal{C} and \mathcal{A} be two sets, and $\alpha : \mathcal{P}(\mathcal{C}) \longrightarrow \mathcal{P}(\mathcal{A})$ be a function such that $\alpha(c) = \bigcup_{e \in c} \alpha(\{e\})$.*

Then the function $\gamma(a) = \bigcup \alpha^{-1}(\downarrow a)$ forms a Galois connection $\mathcal{P}(\mathcal{C}) \xrightleftharpoons[\gamma]{\alpha} \mathcal{P}(\mathcal{A})$ between $(\mathcal{P}(\mathcal{C}), \subseteq)$ and $(\mathcal{P}(\mathcal{A}), \subseteq)$.

PROOF: We show that α is monotonic and $\gamma(a) = \max \alpha^{-1}(\downarrow a)$.

The monotonicity of α is immediate since if $c \subseteq c'$ we have

$$\bigcup_{c_i \in c} \alpha(\{c_i\}) \subseteq \bigcup_{c_i \in c'} \alpha(\{c_i\}).$$

Now, let us consider $c = \gamma(a) = \bigcup \alpha^{-1}(\downarrow a)$, we need to prove that $c \in \alpha^{-1}(\downarrow a)$, i.e. $\alpha(c) \in \downarrow a$.

...

□

Pointwise Galois Connections between Powersets

Lemma 3 *Let \mathcal{C} and \mathcal{A} be two sets, and $\alpha : \mathcal{P}(\mathcal{C}) \longrightarrow \mathcal{P}(\mathcal{A})$ be a function such that $\alpha(c) = \bigcup_{e \in c} \alpha(\{e\})$.*

Then the function $\gamma(a) = \bigcup \alpha^{-1}(\downarrow a)$ forms a Galois connection $\mathcal{P}(\mathcal{C}) \xrightleftharpoons[\gamma]{\alpha} \mathcal{P}(\mathcal{A})$ between $(\mathcal{P}(\mathcal{C}), \subseteq)$ and $(\mathcal{P}(\mathcal{A}), \subseteq)$.

PROOF: We show that α is monotonic and $\gamma(a) = \max \alpha^{-1}(\downarrow a)$.

The monotonicity of α is immediate since if $c \subseteq c'$ we have

$$\bigcup_{c_i \in c} \alpha(\{c_i\}) \subseteq \bigcup_{c_i \in c'} \alpha(\{c_i\}).$$

Now, let us consider $c = \gamma(a) = \bigcup \alpha^{-1}(\downarrow a)$, we need to prove that

$c \in \alpha^{-1}(\downarrow a)$, i.e. $\alpha(c) \in \downarrow a$. We know that

$\alpha(c) = \bigcup_{e \in c} \alpha(\{e\}) = \bigcup_{e \in \bigcup \alpha^{-1}(\downarrow a)} \alpha(\{e\})$. For each e in $\bigcup \alpha^{-1}(\downarrow a)$ there exists $d \in \mathcal{P}(\mathcal{C})$ such that $e \in d$ and $\alpha(d) \subseteq a$, therefore $\alpha(\{e\}) \subseteq a$.

...

□

Pointwise Galois Connections between Powersets

Lemma 3 *Let \mathcal{C} and \mathcal{A} be two sets, and $\alpha : \mathcal{P}(\mathcal{C}) \longrightarrow \mathcal{P}(\mathcal{A})$ be a function such that $\alpha(c) = \bigcup_{e \in c} \alpha(\{e\})$.*

Then the function $\gamma(a) = \bigcup \alpha^{-1}(\downarrow a)$ forms a Galois connection $\mathcal{P}(\mathcal{C}) \xrightleftharpoons[\gamma]{\alpha} \mathcal{P}(\mathcal{A})$ between $(\mathcal{P}(\mathcal{C}), \subseteq)$ and $(\mathcal{P}(\mathcal{A}), \subseteq)$.

PROOF: We show that α is monotonic and $\gamma(a) = \max \alpha^{-1}(\downarrow a)$.

The monotonicity of α is immediate since if $c \subseteq c'$ we have

$$\bigcup_{c_i \in c} \alpha(\{c_i\}) \subseteq \bigcup_{c_i \in c'} \alpha(\{c_i\}).$$

Now, let us consider $c = \gamma(a) = \bigcup \alpha^{-1}(\downarrow a)$, we need to prove that

$c \in \alpha^{-1}(\downarrow a)$, i.e. $\alpha(c) \in \downarrow a$. We know that

$\alpha(c) = \bigcup_{e \in c} \alpha(\{e\}) = \bigcup_{e \in \bigcup \alpha^{-1}(\downarrow a)} \alpha(\{e\})$. For each e in $\bigcup \alpha^{-1}(\downarrow a)$ there exists $d \in \mathcal{P}(\mathcal{C})$ such that $e \in d$ and $\alpha(d) \subseteq a$, therefore $\alpha(\{e\}) \subseteq a$.

Hence $\bigcup_{e \in \bigcup \alpha^{-1}(\downarrow a)} \alpha(\{e\}) \subseteq a$ and thus $\alpha(c) \subseteq a$. □

1. Stochastic Semantics

For a given volume V_k of the location where the compound x_k resides, a concentration C_k for a molecule is translated into a number of molecules $N_k = \lfloor C_k \times V_k \times N_A \rfloor$, where N_A is Avogadro's number.

The **kinetic expression** e_i for each reaction i evaluates on numbers of molecules for each compound in a (positive) **reaction weight** τ_i .

An element s of the domain precisely defines a Markov chain, where the probability p_{ij} of transition from state S_i to S_j is obtained by normalizing the reaction rate $\tau_{i,j} = \sum_{(S_i, S_j, \tau) \in s} \tau$ in

$$p_{ij} = \frac{\tau_{ij}}{\sum_{(S_i, S_k, \tau_{ik}) \in s} \tau_{ik}}$$

Stochastic Semantics Domain

Def. 4 Let a *discrete state* be a vector of integers of dimension $|\mathcal{M}|$.

Def. 5 The universe \mathcal{S} of *stochastic transitions* is the set of triplets (S_i, S_j, τ_{ij}) where S_i and S_j are discrete states and $\tau_{ij} \in \mathbb{R}^+$.

The domain of stochastic transitions is $\mathcal{D}_{\mathcal{S}} = (\mathcal{P}(\mathcal{S}), \subseteq)$.

Remark: Discrete states and solutions in reaction rules have the same mathematical structure (multisets) and can both be represented by $|\mathcal{M}|$ -dimensional vectors of integers.

Rules Domain \rightarrow Stochastic Domain

Proposition 6 *Let $\alpha_{\mathcal{R}\mathcal{S}} : \mathcal{C}_{\mathcal{R}} \rightarrow \mathcal{D}_{\mathcal{S}}$ be the function associating to a reaction model the state transition graph labelled with the $\tau_{i,j}$'s. Let $\gamma_{\mathcal{R}\mathcal{S}}(s) = \cup \alpha_{\mathcal{R}\mathcal{S}}^{-1}(\downarrow s)$.*

$\mathcal{C}_{\mathcal{R}} \xrightarrow{\alpha_{\mathcal{R}\mathcal{S}}} \mathcal{D}_{\mathcal{S}} \xleftarrow{\gamma_{\mathcal{R}\mathcal{S}}} \mathcal{C}_{\mathcal{R}}$ is a Galois connection.

Remarks: $\alpha_{\mathcal{R}\mathcal{S}}$ is not one-to-one.

For instance, the reaction models $m1 = \{ \text{e for } A \Rightarrow B \}$ and $m2 = m1 \cup \{ \text{e for } 2*A \Rightarrow A+B \}$ have the same set of stochastic transitions. $\gamma \circ \alpha$ is thus not the identity, the information lost by the stochastic abstraction is the elimination of redundant rules in the reaction model.

$\alpha_{\mathcal{R}\mathcal{S}}$ is neither onto

2. Discrete Semantics

Def. 7 *The universe \mathcal{D} of **discrete transitions** is the set of pairs of discrete states. The domain of discrete transitions is $\mathcal{D}_{\mathcal{D}} = (\mathcal{P}(\mathcal{D}), \subseteq)$.*

Remark: The discrete semantics is the classical Petri net semantics of reaction models [RML93ismb,SHK06bmcbi,Chaouiya07bi,GHL07cmsb].

Classical Petri net analysis tools can be used for the analysis of reaction models at this abstraction level.

For instance, the elementary mode analysis of metabolic networks [SPM02bi] has been shown in [ZS03insilicobio] to be equivalent to the classical analysis of Petri nets by T-invariants.

Discrete Semantics

Proposition 8 *Let $\alpha_{\mathcal{S}\mathcal{D}} : \mathcal{D}_{\mathcal{S}} \rightarrow \mathcal{D}_{\mathcal{D}}$ be the function associating to a set of stochastic transitions the discrete transitions obtained by projection on the two first components, and $\gamma_{\mathcal{S}\mathcal{D}}(d) = \cup \alpha_{\mathcal{S}\mathcal{D}}^{-1}(\downarrow d)$.*

$\mathcal{D}_{\mathcal{S}} \xrightleftharpoons[\gamma_{\mathcal{S}\mathcal{D}}]{\alpha_{\mathcal{S}\mathcal{D}}} \mathcal{D}_{\mathcal{D}}$ is a Galois connection.

Remarks: $\alpha_{\mathcal{S}\mathcal{D}}$ is onto

but **not one-to-one** as the transition rates are simply forgotten.

3. Boolean Semantics

Let a **boolean state** be a vector of booleans of dimension $|\mathcal{M}|$ indicating the presence of each molecule in the state.

Def. 9 *The universe \mathcal{B} of **boolean transitions** is the set of pairs of boolean states.*

The domain of boolean transitions is $\mathcal{D}_{\mathcal{B}} = (\mathcal{P}(\mathcal{B}), \subseteq)$.

Boolean Semantics

Let a **boolean state** be a vector of booleans of dimension $|\mathcal{M}|$ indicating the presence of each molecule in the state.

Def. 9 *The universe \mathcal{B} of **boolean transitions** is the set of pairs of boolean states.*

The domain of boolean transitions is $\mathcal{D}_{\mathcal{B}} = (\mathcal{P}(\mathcal{B}), \subseteq)$.

Let $\alpha_{\mathcal{N}\mathcal{B}} : \mathbb{N}^{|\mathcal{M}|} \rightarrow \mathbb{B}^{|\mathcal{M}|}$ be the **zero/non-zero abstraction** from the integers to the booleans, and its pointwise extension from discrete states to boolean states.

Proposition 10 *Let $\alpha_{\mathcal{D}\mathcal{B}} : \mathcal{D}_{\mathcal{D}} \rightarrow \mathcal{D}_{\mathcal{B}}$ be the set extension of $\alpha_{\mathcal{N}\mathcal{B}}$. Let $\gamma_{\mathcal{D}\mathcal{B}}(b) = \cup \alpha_{\mathcal{D}\mathcal{B}}^{-1}(\downarrow b)$.*

$\mathcal{D}_{\mathcal{D}} \xrightleftharpoons[\gamma_{\mathcal{D}\mathcal{B}}]{\alpha_{\mathcal{D}\mathcal{B}}} \mathcal{D}_{\mathcal{B}}$ *is a Galois connection.*

Remark: $\alpha_{\mathcal{D}\mathcal{B}}$ is onto but not one-to-one

4. Biocham Boolean Semantics

Given a reaction model R , let us denote by S_{BB} the set of boolean transitions obtained by considering all possible consumption of reactants.

For instance, the rule $A+B \Rightarrow C+D$ gives rise to four boolean transitions:

- $A \wedge B \longrightarrow A \wedge B \wedge C \wedge D$
- $A \wedge B \longrightarrow \neg A \wedge B \wedge C \wedge D$
- $A \wedge B \longrightarrow A \wedge \neg B \wedge C \wedge D$
- $A \wedge B \longrightarrow \neg A \wedge \neg B \wedge C \wedge D$

Remark: Biocham Boolean semantics differs from Boolean Petri nets, Pathway Logic, etc. where complete consumption of the reactants is always assumed.

Over-approximation Theorem of Biocham Boolean Semantics

Theorem 11 *For any reaction model R , $\alpha_{DB}(\alpha_{SD}(\alpha_{RS}(R))) \subseteq S_{BB}$.*

Over-approximation Theorem of Biocham Boolean Semantics

Theorem 11 *For any reaction model R , $\alpha_{\mathcal{DB}}(\alpha_{\mathcal{SD}}(\alpha_{\mathcal{RS}}(R))) \subseteq S_{BB}$.*

PROOF: Since all our abstractions are defined pointwise, it is enough to prove it for only one rule in R . Let us consider e for $S \Rightarrow S'$.

$$\alpha_{\mathcal{RS}}(R) = \{(S_i, S_j, e) \mid S_i \geq S, S_j = S_i - S + S'\},$$

□

Over-approximation Theorem of Biocham Boolean Semantics

Theorem 11 *For any reaction model R , $\alpha_{\mathcal{DB}}(\alpha_{\mathcal{SD}}(\alpha_{\mathcal{RS}}(R))) \subseteq S_{BB}$.*

PROOF: Since all our abstractions are defined pointwise, it is enough to prove it for only one rule in R . Let us consider e for $S \Rightarrow S'$.

$$\alpha_{\mathcal{RS}}(R) = \{(S_i, S_j, e) \mid S_i \geq S, S_j = S_i - S + S'\},$$

$$\alpha_{\mathcal{SD}}(\alpha_{\mathcal{RS}}(R)) = \{(S_i, S_j) \mid S_i \geq S, S_j = S_i - S + S'\},$$

□

Over-approximation Theorem of Biocham Boolean Semantics

Theorem 11 *For any reaction model R , $\alpha_{DB}(\alpha_{SD}(\alpha_{RS}(R))) \subseteq S_{BB}$.*

PROOF: Since all our abstractions are defined pointwise, it is enough to prove it for only one rule in R . Let us consider e for $S \Rightarrow S'$.

$$\alpha_{RS}(R) = \{(S_i, S_j, e) \mid S_i \geq S, S_j = S_i - S + S'\},$$

$$\alpha_{SD}(\alpha_{RS}(R)) = \{(S_i, S_j) \mid S_i \geq S, S_j = S_i - S + S'\},$$

$$\alpha_{DB}(\alpha_{SD}(\alpha_{RS}(R))) = \{(S'_i, S'_j) \mid S_i \geq S, S_j = S_i - S + S', S'_i = \alpha_{NB}(S_i), S'_j = \alpha_{NB}(S_j)\}.$$

$$S_{BB} = \{(T, T') \mid T \geq \alpha_{NB}(S), \alpha_{NB}(S') \vee (T \wedge \neg \alpha_{NB}(S)) \leq T' \leq \alpha_{NB}(T) \vee \alpha_{NB}(S')\}$$

□

Over-approximation Theorem of Biocham Boolean Semantics

Theorem 11 For any reaction model R , $\alpha_{\mathcal{DB}}(\alpha_{\mathcal{SD}}(\alpha_{\mathcal{RS}}(R))) \subseteq S_{BB}$.

PROOF: Since all our abstractions are defined pointwise, it is enough to prove it for only one rule in R . Let us consider e for $S \Rightarrow S'$.

$$\alpha_{\mathcal{RS}}(R) = \{(S_i, S_j, e) \mid S_i \geq S, S_j = S_i - S + S'\}$$

$$\alpha_{\mathcal{SD}}(\alpha_{\mathcal{RS}}(R)) = \{(S_i, S_j) \mid S_i \geq S, S_j = S_i - S + S'\}$$

$$\alpha_{\mathcal{DB}}(\alpha_{\mathcal{SD}}(\alpha_{\mathcal{RS}}(R))) = \{(S'_i, S'_j) \mid S_i \geq S, S_j = S_i - S + S', S'_i = \alpha_{\mathcal{NB}}(S_i), S'_j = \alpha_{\mathcal{NB}}(S_j)\}.$$

$$S_{BB} = \{(T, T') \mid T \geq \alpha_{\mathcal{NB}}(S), \alpha_{\mathcal{NB}}(S') \vee (T \wedge \neg \alpha_{\mathcal{NB}}(S)) \leq T' \leq \alpha_{\mathcal{NB}}(T) \vee \alpha_{\mathcal{NB}}(S')\}$$

Since $S_i \geq S$ we have $S'_i \geq \alpha_{\mathcal{NB}}(S)$ by monotonicity of $\alpha_{\mathcal{NB}}$

We have $S_j = S_i - S + S'$ hence $S_j \leq S_i + S'$, and

$$\alpha_{\mathcal{NB}}(S_j) = \alpha_{\mathcal{NB}}(S') \vee (\alpha_{\mathcal{NB}}(S_i) \wedge \neg \alpha_{\mathcal{NB}}(S)) \leq S'_j$$

$$S'_j \leq \alpha_{\mathcal{NB}}(S_i + S') = \alpha_{\mathcal{NB}}(S_i) \vee \alpha_{\mathcal{NB}}(S')$$

□

Abstract Interpretation for Systems Biology

Part II: Type Checking and Type Inference

1. Type Checking and Type Inference
2. Domain of Protein Functions
3. Domain of Protein Influences
4. Domain of Compartment Neighborhoods

Type Checking/Inference by Abstract Interpretation

A **type system** \mathcal{A} for a concrete domain \mathcal{C} is a Galois connection $\mathcal{C} \rightarrow_{\alpha} \mathcal{A}$.

Type Checking/Inference by Abstract Interpretation

A **type system** \mathcal{A} for a concrete domain \mathcal{C} is a Galois connection $\mathcal{C} \rightarrow_{\alpha} \mathcal{A}$.

The **type inference** problem is

INPUT a concrete element $x \in \mathcal{C}$ (e.g. a reaction model)

OUTPUT its typing $\alpha(x)$ (e.g. the protein functions of the model).

Type Checking/Inference by Abstract Interpretation

A **type system** A for a concrete domain C is a Galois connection $C \rightarrow_{\alpha} \mathcal{A}$.

The **type inference** problem is

INPUT a concrete element $x \in C$ (e.g. a reaction model)

OUTPUT its typing $\alpha(x)$ (e.g. the protein functions of the model).

The **type checking** problem is,

INPUT $x \in C$ (e.g. a reaction model)

and a typing $y \in \mathcal{A}$ (e.g. a set of protein functions),

OUTPUT determine whether $x \sqsubseteq_C \gamma(y)$

(i.e. whether the reactions are compatible with the protein functions)

or equivalently $\alpha(x) \sqsubseteq_{\mathcal{A}} y$ (the typing contains the inferred types)

Type Checking/Inference by Abstract Interpretation

A **type system** A for a concrete domain C is a Galois connection $C \rightarrow_{\alpha} \mathcal{A}$.

The **type inference** problem is

INPUT a concrete element $x \in C$ (e.g. a reaction model)

OUTPUT its typing $\alpha(x)$ (e.g. the protein functions of the model).

The **type checking** problem is,

INPUT $x \in C$ (e.g. a reaction model)

and a typing $y \in \mathcal{A}$ (e.g. a set of protein functions),

OUTPUT determine whether $x \sqsubseteq_C \gamma(y)$

(i.e. whether the reactions are compatible with the protein functions)

or equivalently $\alpha(x) \sqsubseteq_{\mathcal{A}} y$ (the typing contains the inferred types)

Algorithms in $O(n)$ if the abstractions can be computed rule per rule.

Type Checking/Inference of Protein Functions

Abstract domain $\mathcal{A}_{\mathcal{F}} = \mathcal{P}(\{\text{kinase}(A) \mid A \in \mathcal{M}\} \cup \{\text{phosphatase}(A) \mid A \in \mathcal{M}\})$

The [typing of reactions by protein functions](#) is defined by the abstraction :

$\alpha_{\mathcal{F}}(A = [B] \Rightarrow C) = \{\text{kinase}(B)\}$ if C is strictly more phosphorylated than A

$\alpha_{\mathcal{F}}(A = [B] \Rightarrow C) = \{\text{phosphatase}(B)\}$ if C is strictly less phosphorylated

$\alpha_{\mathcal{F}}(A + B \Rightarrow A-B, A-B \Rightarrow C + B) = \{\text{kinase}(B)\}$

if C is strictly more phosphorylated than A

$\alpha_{\mathcal{F}}(A + B \Rightarrow A-B, A-B \Rightarrow C + B) = \{\text{phosphatase}(B)\}$

if C is strictly less phosphorylated than A

Type Checking/Inference of Protein Functions

Abstract domain $\mathcal{A}_{\mathcal{F}} = \mathcal{P}(\{\text{kinase}(A) \mid A \in \mathcal{M}\} \cup \{\text{phosphatase}(A) \mid A \in \mathcal{M}\})$

The **typing of reactions by protein functions** is defined by the abstraction :

$\alpha_{\mathcal{F}}(A \xrightarrow{[B]} C) = \{\text{kinase}(B)\}$ if C is strictly more phosphorylated than A

$\alpha_{\mathcal{F}}(A \xrightarrow{[B]} C) = \{\text{phosphatase}(B)\}$ if C is strictly less phosphorylated

$\alpha_{\mathcal{F}}(A + B \Rightarrow A-B, A-B \Rightarrow C + B) = \{\text{kinase}(B)\}$

if C is strictly more phosphorylated than A

$\alpha_{\mathcal{F}}(A + B \Rightarrow A-B, A-B \Rightarrow C + B) = \{\text{phosphatase}(B)\}$

if C is strictly less phosphorylated than A

Proposition 12 $\alpha_{\mathcal{F}}$ can be computed in $O(n^2)$ time where n is the number of rules.

Proposition 13 Let $\gamma_{\mathcal{F}}(f) = \cup \alpha_{\mathcal{F}}^{-1}(\downarrow f)$, $\mathcal{C}_{\mathcal{R}} \xrightleftharpoons[\gamma_{\mathcal{F}}]{\alpha_{\mathcal{F}}} \mathcal{A}_{\mathcal{F}}$ is a Galois connection.

More Precise Protein Function Typing

In SBML : no typing possible as there is no syntax for phosphorylation

In BIOCHAM : typing is possible but the syntax does not distinguish between phosphorylation, acetylation etc.

More precise protein function types:

$$\tau ::= kinase|phosphatase|kinase(\tau)|phosphatase(\tau)|T$$

where T denotes some basic types of proteins, with the [subtyping relations](#) $kinase(\tau) \preceq kinase$ and $phosphatase(\tau) \preceq phosphatase$.

Evaluation Results in BIOCHAM

- MAPK model [Levchenko et al. 00]
the kinase function of RAFK, RAF~{p1} and MEK~{p1,p2} is inferred;
the phosphatase function of RAFPH, MEKPH and MAPKPH is inferred;
the kinase function of MAPK~{p1,p2} is not visible and not inferred.

Evaluation Results in BIOCHAM

- MAPK model [Levchenko et al. 00]
the kinase function of RAFK, $RAF\sim\{p1\}$ and $MEK\sim\{p1,p2\}$ is inferred;
the phosphatase function of RAFPH, MEKPH and MAPKPH is inferred;
the kinase function of $MAPK\sim\{p1,p2\}$ is not visible and not inferred.
- Model of the mammalian cell cycle control after [Kohn 99] 165 proteins and genes, 500 variables and 800 rules. **Type inference in < 1sec CPU :**
 - No compound is both a kinase and a phosphatase;
 - `cdc25A` and `cdc25C` are the only phosphatases found together with the deacetylase `HDAC1`.
 - The `cdk` are inferred to be kinases only in complexes with cyclins;
 - the acetylases `pCAF`, `p300` are identified to kinases.

Type Checking/Inference of Location Neighborhood

Abstract domain $\mathcal{A}_{\mathcal{N}} = \mathcal{P}(\{\text{neighbors}(A, B) \mid A, B \in \mathcal{M}\})$.

$\alpha_{\mathcal{RN}}(\mathbf{e} \text{ for } \mathbf{A}_1 + \dots + \mathbf{A}_m \Rightarrow \mathbf{A}_{m+1} + \dots + \mathbf{A}_n) =$

$\{\text{neighbors}(A_i, A_j) \mid 1 \leq i, j \leq n\} \cup \{\text{neighbors}(A_i, C) \mid 1 \leq i \leq n, C \in \mathbf{e}\}$.

Proposition 14 $\alpha_{\mathcal{RN}}$ can be computed in $O(n)$ time where n is the number of reaction rules.

Type Checking/Inference of Location Neighborhood

SBML models <http://www.biomodels.net>

13 over 50 models have compartments and 7 use the *outside* attribute

BIOMD39.xml: `neighbor(cytosol,reticulum)`, `neighbor(cytosol,mitochondria)` inferred and checked consistent with the outside attributes.

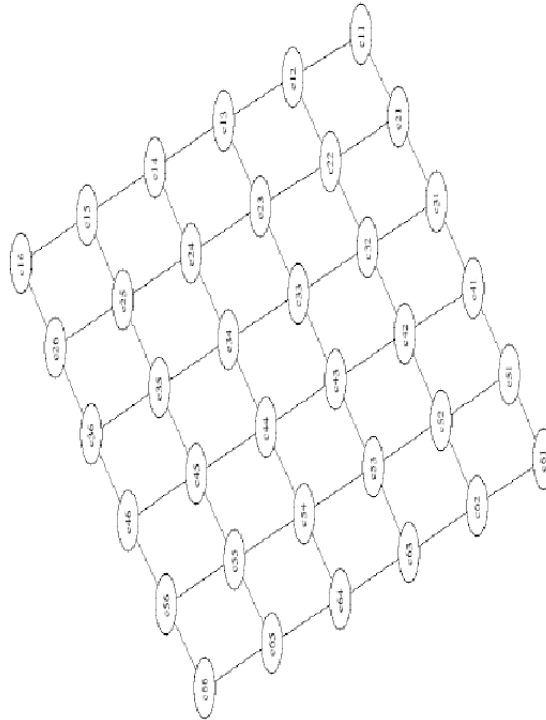
BIOMD45.xml: `neighbor(cytosol,extracellular)`, `neighbor(cytosol,vesicula1)`, `neighbor(cytosol, vesicula1)` inferred and checked consistent with the outside attributes.

BIOCHAM model p53-Mdm2 : `neighbor(cytosol,nucleus)` inferred

volume ratio not systematically used in the published model [Ciliberto 05]

Cell Grid Inferred in a Square 6x6 Delta-Notch Model

```
(if [D::c21]+[D::c23]+[D::c12]+[D::c32] < 0.2 then 0 else ka,MA(kd)) for _ <=> N::c22.  
(if [D::c22]+[D::c24]+[D::c13]+[D::c33] < 0.2 then 0 else ka,MA(kd)) for _ <=> N::c23.  
(if [D::c23]+[D::c25]+[D::c14]+[D::c34] < 0.2 then 0 else ka,MA(kd)) for _ <=> N::c24.  
(if [D::c24]+[D::c26]+[D::c15]+[D::c35] < 0.2 then 0 else ka,MA(kd)) for _ <=> N::c25.  
...
```



Type Checking/Inference of Influence Graphs

$$\mathcal{A}_I = \mathcal{P}(\{A \xrightarrow{+} B \mid A, B \in \mathcal{M}\} \cup \{A \xrightarrow{-} B \mid A, B \in \mathcal{M}\}).$$

The **influence graph of a reaction model** is defined by $\alpha_{\mathcal{R}I} : \mathcal{C}_{\mathcal{R}} \rightarrow \mathcal{A}_I$

$$\begin{aligned} \alpha_{\mathcal{R}I}(x) = & \{A \xrightarrow{-} B \mid \exists(e_i \text{ for } S_i \Rightarrow S'_i) \in x, \\ & l_i(A) > 0 \text{ and } r_i(B) - l_i(B) < 0\} \\ & \cup \{A \xrightarrow{+} B \mid \exists(e_i \text{ for } S_i \Rightarrow S'_i) \in x, \\ & l_i(A) > 0 \text{ and } r_i(B) - l_i(B) > 0\} \end{aligned}$$

Type Checking/Inference of Influence Graphs

$$\mathcal{A}_I = \mathcal{P}(\{A \xrightarrow{+} B \mid A, B \in \mathcal{M}\} \cup \{A \xrightarrow{-} B \mid A, B \in \mathcal{M}\}).$$

The **influence graph of a reaction model** is defined by $\alpha_{\mathcal{R}I} : \mathcal{C}_{\mathcal{R}} \rightarrow \mathcal{A}_I$

$$\begin{aligned} \alpha_{\mathcal{R}I}(x) = & \{A \xrightarrow{-} B \mid \exists(e_i \text{ for } S_i \Rightarrow S'_i) \in x, \\ & l_i(A) > 0 \text{ and } r_i(B) - l_i(B) < 0\} \\ & \cup \{A \xrightarrow{+} B \mid \exists(e_i \text{ for } S_i \Rightarrow S'_i) \in x, \\ & l_i(A) > 0 \text{ and } r_i(B) - l_i(B) > 0\} \end{aligned}$$

$$\alpha_{\mathcal{R}I}(\{A + B \Rightarrow C\}) = \{ \quad A \xrightarrow{-} B, A \xrightarrow{-} A, B \xrightarrow{-} A, \\ \quad B \xrightarrow{-} B, A \xrightarrow{+} C, B \xrightarrow{+} C\}$$

$$\alpha_{\mathcal{R}I}(\{A = [C] \Rightarrow B\}) = \{ \quad C \xrightarrow{-} A, A \xrightarrow{-} A, A \xrightarrow{+} B, C \xrightarrow{+} B\}$$

$$\alpha_{\mathcal{R}I}(\{A = [B] \Rightarrow -\}) = \{ \quad B \xrightarrow{-} A, A \xrightarrow{-} A\}$$

$$\alpha_{\mathcal{R}I}(\{- = [B] \Rightarrow A\}) = \{ \quad B \xrightarrow{+} A\}$$

Type Checking/Inference of Influence Graphs

$$\mathcal{A}_I = \mathcal{P}(\{A \xrightarrow{+} B \mid A, B \in \mathcal{M}\} \cup \{A \xrightarrow{-} B \mid A, B \in \mathcal{M}\}).$$

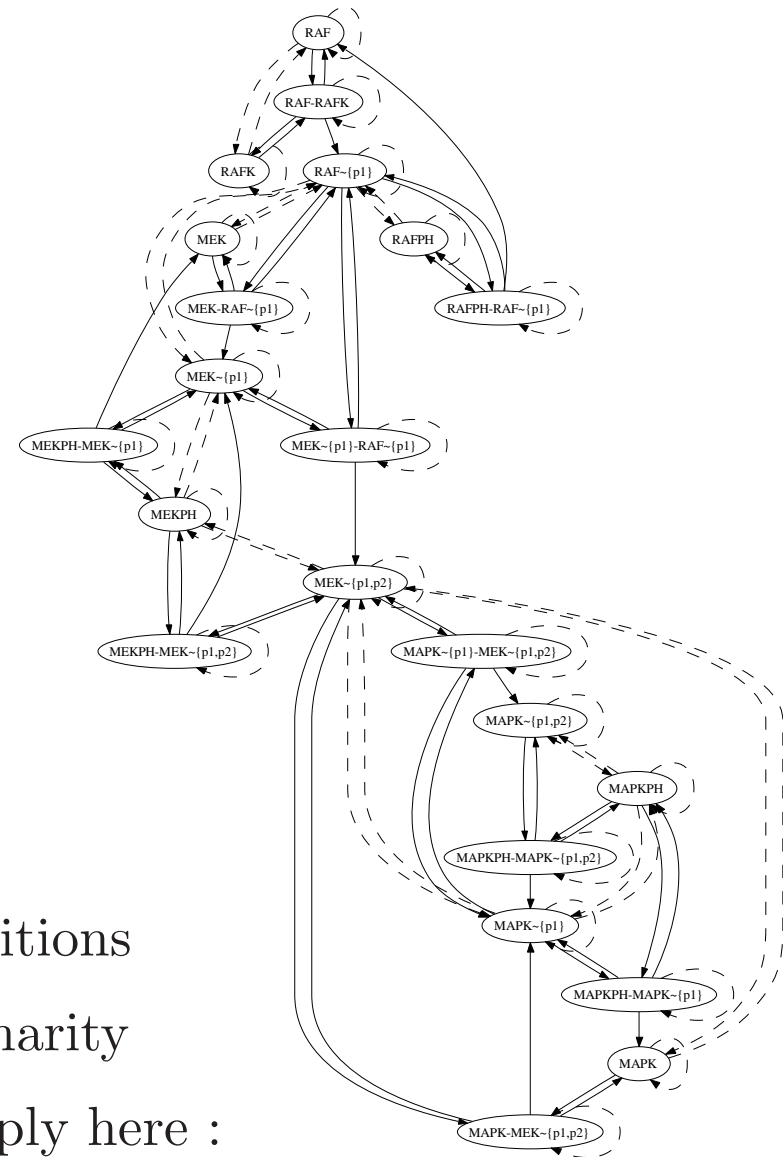
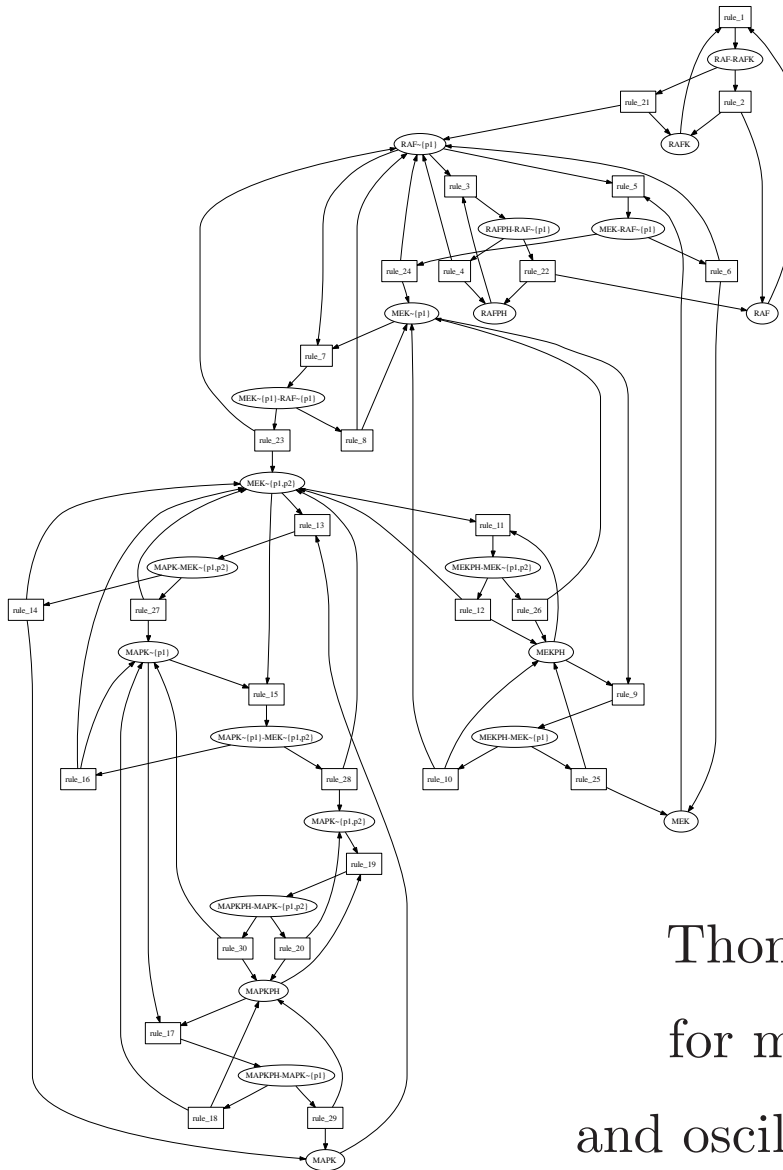
The **influence graph of a reaction model** is defined by $\alpha_{\mathcal{R}I} : \mathcal{C}_{\mathcal{R}} \rightarrow \mathcal{A}_I$

$$\begin{aligned} \alpha_{\mathcal{R}I}(x) = & \{A \xrightarrow{-} B \mid \exists(e_i \text{ for } S_i \Rightarrow S'_i) \in x, \\ & l_i(A) > 0 \text{ and } r_i(B) - l_i(B) < 0\} \\ & \cup \{A \xrightarrow{+} B \mid \exists(e_i \text{ for } S_i \Rightarrow S'_i) \in x, \\ & l_i(A) > 0 \text{ and } r_i(B) - l_i(B) > 0\} \end{aligned}$$

Proposition 15 $\alpha_{\mathcal{R}I}$ can be computed in $O(n)$ time where n is the number of rules.

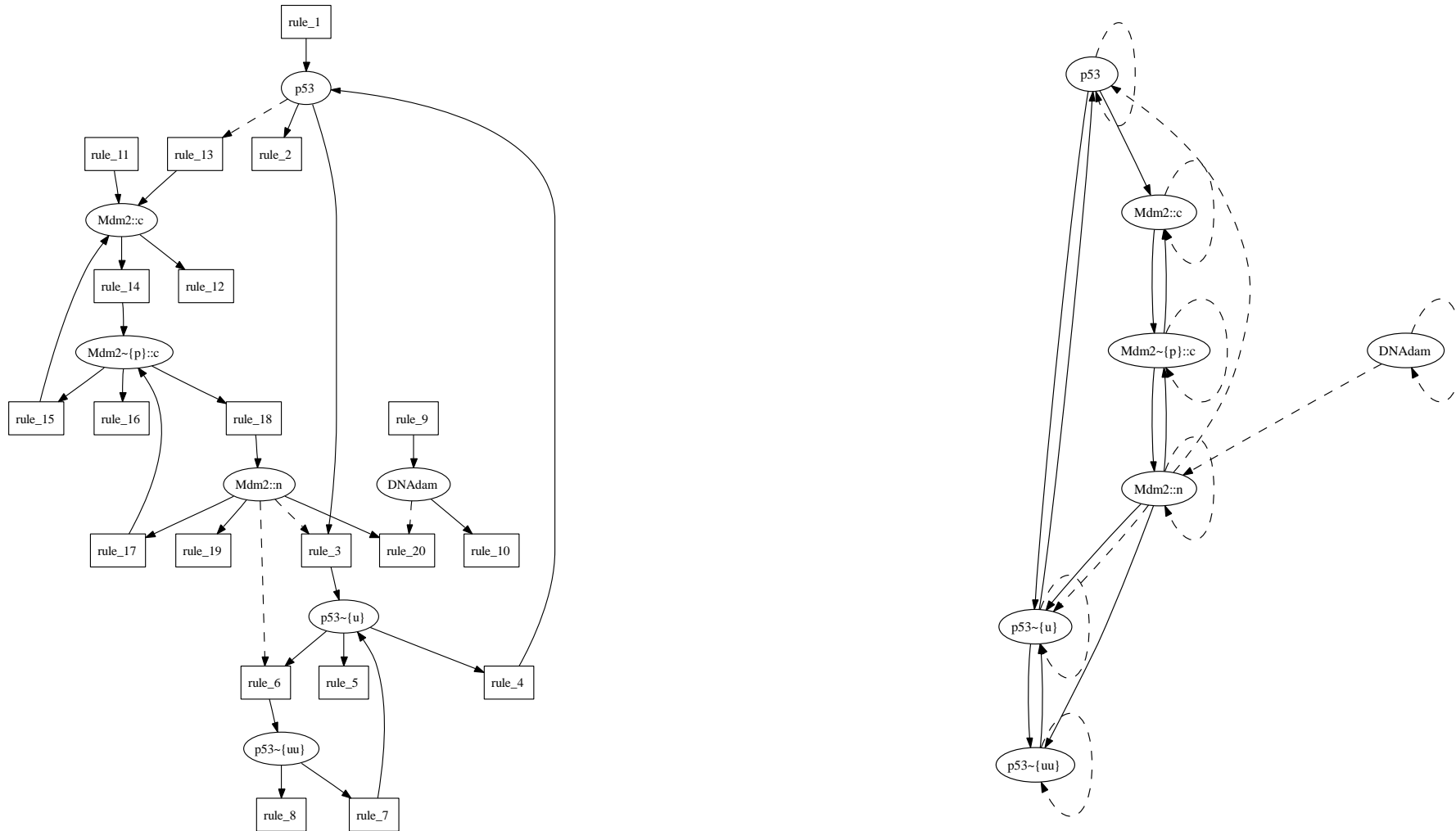
Proposition 16 Let $\gamma_{\mathcal{R}I}(f) = \cup \alpha_{\mathcal{R}I}^{-1}(\downarrow f)$, $\mathcal{C}_{\mathcal{R}} \xrightarrow{\alpha_{\mathcal{R}I}} \mathcal{A}_I \xleftarrow{\gamma_{\mathcal{R}I}} \mathcal{C}_{\mathcal{R}}$ is a Galois connection.

MAPK model: Reaction Graph \rightarrow_{α} Influence Graph



Thomas's conditions
for multistationarity
and oscillations apply here :

P53-Mdm2: Reaction Graph \rightarrow_{α} Influence Graph



Inhibitions hidden in the kinetic expressions are missed

Influence Graph Abstraction from the Differential Semantics

Let us denote by β the mapping from $\mathcal{C}_{\mathcal{R}}$ to $\mathcal{D}_{\mathcal{J}}$ that extracts \dot{x}_k and hence the Jacobian from the kinetic expressions in the reaction rules.

Def. 17 *The differential influence graph abstraction $\alpha_{\mathcal{J}\mathcal{I}} : \mathcal{D}_{\mathcal{J}} \rightarrow \mathcal{A}_{\mathcal{I}}$ is the function*

$$\alpha_{\mathcal{J}\mathcal{I}}(x) = \{A \xrightarrow{+} B \mid \partial \dot{x}_B / \partial x_A > 0 \text{ in some point of the phase space}\} \\ \cup \{A \xrightarrow{-} B \mid \partial \dot{x}_B / \partial x_A < 0 \text{ in some point of the phase space}\}$$

defined purely from the kinetic expressions... compatibility with the rules ?

Increasing Kinetics

Def. 18 A kinetic expression e_i is *increasing* w.r.t. a reaction model x iff for all molecules x_k we have

1. for all points of the phase space $\partial e_i / \partial x_k \geq 0$
2. if there exists a point in the phase space s.t. $\partial e_i / \partial x_k > 0$ then $l_i(x_k) > 0$

The model x will be said to have *increasing kinetics* if each of its reaction rules has a increasing kinetic expression.

The mass action law kinetics, $e_i = k * \prod x_i^{l_i}$, are increasing

Hill's kinetics (and Michaelis-Menten kinetics when $n = 1$)
 $e_i = V_m * x_s^n / (K_m^n + x_s^n)$ are also increasing.

Comparison to the Syntactical Influence Graph

Theorem 19 (Over-approximation) *For any reaction model x with increasing kinetics, $\alpha_{\mathcal{J}\mathcal{I}} \circ \beta(x) \subseteq \alpha_{\mathcal{R}\mathcal{I}}(x)$.*

PROOF: If $(A \xrightarrow{+} B) \in \alpha_{\mathcal{J}\mathcal{I}} \circ \beta(x)$ then $\partial \dot{B} / \partial A > 0$. Hence there exists a term $(r_i(B) - l_i(B)) * e_i$ in the ODE with $\partial e_i / \partial A$ of the same sign as $r_i(B) - l_i(B)$. Let us suppose that $r_i(B) - l_i(B) > 0$ then $\partial e_i / \partial A > 0$ and since e_i is increasing we get that $l_i(A) > 0$ and thus that $(A \xrightarrow{+} B) \in \alpha_{\mathcal{R}\mathcal{I}}(x)$. If on the contrary $r_i(B) - l_i(B) < 0$ then $\partial e_i / \partial A < 0$, impossible.

If $(A \xrightarrow{-} B) \in \alpha_{\mathcal{J}\mathcal{I}} \circ \beta(x)$ then $\partial \dot{B} / \partial A < 0$. Hence there exists a term $(r_i(B) - l_i(B)) * e_i$ with $\partial e_i / \partial A$ of sign opposite to that of $r_i(B) - l_i(B)$. Let us suppose that $r_i(B) - l_i(B) > 0$ then $\partial e_i / \partial A < 0$, impossible. If on the contrary $r_i(B) - l_i(B) < 0$ then $\partial e_i / \partial A > 0$ and since e_i is increasing we get that $l_i(A) > 0$ and thus that $(A \xrightarrow{+} B) \in \alpha_{\mathcal{R}\mathcal{I}}(x)$.

□

Comparison to the Syntactical Influence Graph

Even with mass action law kinetics, there is no equality between $\alpha_{JI} \circ \beta$ and α_{RI} .

Comparison to the Syntactical Influence Graph

Even with mass action law kinetics, there is no equality between $\alpha_{\mathcal{J}\mathcal{I}} \circ \beta$ and $\alpha_{\mathcal{R}\mathcal{I}}$.

For instance let x be the following model :



We have $\alpha_{\mathcal{R}\mathcal{I}}(x) = \{A \xrightarrow{+} B, A \xrightarrow{+} A, A \xrightarrow{-} A\}$,

however $\dot{A} = (k_2 - k_1) * A$,

hence $\partial \dot{A} / \partial A$ can be made always positive, negative or null

resulting in the absence from of $A \xrightarrow{-} A$ or $A \xrightarrow{+} A$ or both from $\alpha_{\mathcal{J}\mathcal{I}} \circ \beta(x)$.

Strongly Increasing Kinetics

Def. 20 A kinetic expression e_i is *strongly increasing* w.r.t. a reaction model x iff for all molecules x_k we have

1. for all points of the phase space $\partial e_i / \partial x_k \geq 0$
2. there exists a point in the phase space s.t. $\partial e_i / \partial x_k > 0$ iff $l_i(x_k) > 0$

Remarks: strongly increasing implies increasing. Mass action law, Michaelis Menten, and Hill kinetics are strongly increasing.

Theorem 21 If x has strongly increasing kinetics and the syntactical influence graph contains no both positive and negative pair, then $\alpha_{RI}(x) = \alpha_{JI} \circ \beta(x)$.

Strongly Increasing Kinetics

Def. 21 A kinetic expression e_i is *strongly increasing* w.r.t. a reaction model x iff for all molecules x_k we have

1. for all points of the phase space $\partial e_i / \partial x_k \geq 0$
2. there exists a point in the phase space s.t. $\partial e_i / \partial x_k > 0$ iff $l_i(x_k) > 0$

Remarks: strongly increasing implies increasing. Mass action law, Michaelis Menten, and Hill kinetics are strongly increasing.

Theorem 22 If x has strongly increasing kinetics and the syntactical influence graph contains no both positive and negative pair, then

$$\alpha_{\mathcal{RI}}(x) = \alpha_{\mathcal{JI}} \circ \beta(x).$$

Corollary 23 Under these conditions, the (global) differential influence graph of a reaction model is independent of the kinetics !

Corollary 24 The (global) differential influence graph is computable in linear time.