Computational Methods in Systems and Synthetic Biology

François Fages
Constraint Programming Group

INRIA Paris-Rocquencourt

mailto:Francois.Fages@inria.fr

http://contraintes.inria.fr

Overview of the Lectures

1. Formal molecules and reaction models in BIOCHAM
2. Kinetics
3. Qualitative properties formalized in temporal logic CTL
4. Quantitative properties formalized in LTL(R) and pLTL(R)
5. Reaction hypergraphs and influence graphs
6. Hierarchy of semantics and typing for systems biology by abstract interpretation
   - Theory of abstract interpretation
   - Domain of reaction rule models
   - Domains of stochastic, discrete and boolean traces
   - Type checking/inference of protein functions, influence graphs, compartments topology
7. ...

Abstract Interpretation for Systems Biology

Part I: Hierarchy of Semantics

1. Theory of Abstract Interpretation
2. Syntactical Domain of SBML Reaction Rules
3. Stochastic Semantics Domain
4. Discrete Semantics Domain
5. Boolean Semantics Domain

Abstraction in Systems Biology

Models are built in Systems Biology with two contradictory perspectives :

Abstraction 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 ...

Abstraction 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 ...

Abstraction 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 into hierarchies of abstractions.

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

The Theory of Abstract Interpretation

In this setting [Cousot Cousot 77], a domain is a lattice $\mathcal{D}(\sqsubseteq, \perp, \top, \sqcup, \sqcap)$ where $\sqsubseteq$ is the "information loss" ordering.

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

The Theory of Abstract Interpretation

In this setting [Cousot Cousot 77], a domain is a lattice $\mathcal{D}(\sqsubseteq, \bot, \top, \sqcup, \sqcap)$ where $\sqsubseteq$ is the "information loss" ordering.

A Galois connection $\mathcal{C} \rightarrow_\alpha \mathcal{A}$ between two lattices $\mathcal{C}$ and $\mathcal{A}$ is defined by two abstraction and concretization functions $\alpha : \mathcal{C} \rightarrow \mathcal{A}$ and $\gamma : \mathcal{A} \rightarrow \mathcal{C}$ that are monotonic:

- $\forall\ x, y \in \mathcal{C}\ x \sqsubseteq_\mathcal{C} y\ \Rightarrow\ \alpha(x) \sqsubseteq_\mathcal{A} \alpha(y)$,
- $\forall\ x, y \in \mathcal{A}\ x \sqsubseteq_\mathcal{A} y\ \Rightarrow\ \gamma(x) \sqsubseteq_\mathcal{C} \gamma(y)$,

and are adjoint:

- $\forall c \in \mathcal{C}, \forall y \in \mathcal{A} : x \sqsubseteq_\mathcal{C} \gamma(y) \Leftrightarrow \alpha(x) \sqsubseteq_\mathcal{A} y$.

The Theory of Abstract Interpretation

In this setting [Cousot Cousot 77], a domain is a lattice $\mathcal{D}(\sqsubseteq, \bot, \top, \sqcup, \sqcap)$ where $\sqsubseteq$ is the "information loss" ordering.

A Galois connection $\mathcal{C} \underset{\gamma}{\overset{\alpha}{\rightleftarrows}} \mathcal{A}$ between two lattices $\mathcal{C}$ and $\mathcal{A}$ is defined by two abstraction and concretization functions $\alpha : \mathcal{C} \to \mathcal{A}$ and $\gamma : \mathcal{A} \to \mathcal{C}$ that are monotonic:

- $\forall\ x, y \in \mathcal{C}\ x \sqsubseteq_{\mathcal{C}} y \ \Rightarrow\ \alpha(x) \sqsubseteq_{\mathcal{A}} \alpha(y)$,
- $\forall\ x, y \in \mathcal{A}\ x \sqsubseteq_{\mathcal{A}} y \ \Rightarrow\ \gamma(x) \sqsubseteq_{\mathcal{C}} \gamma(y)$,

and are adjoint:

- $\forall c \in \mathcal{C}, \forall y \in \mathcal{A} : x \sqsubseteq_{\mathcal{C}} \gamma(y) \Leftrightarrow \alpha(x) \sqsubseteq_{\mathcal{A}} y$.

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

1. $\gamma \circ \alpha$ is extensive (i.e. $x \sqsubseteq_{\mathcal{C}} \gamma \circ \alpha(x)$) and represents the information lost by the abstraction;

2. $\alpha \circ \gamma$ is contracting (i.e. $\alpha \circ \gamma(y) \sqsubseteq_{\mathcal{A}} y$);

3. $\gamma \circ \alpha$ is the identity *iff* γ is onto *iff* α is one-to-one.

4. α preserves $\sqcup$, and γ preserves $\sqcap$;

5. $\gamma(a) = max\ \alpha^{-1}(\downarrow a) = \sqcup \alpha^{-1}(\downarrow a)$

6. $\alpha(c) = min\ \gamma^{-1}(\uparrow c) = \sqcap \gamma^{-1}(\uparrow c)$

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

It is equivalent in the definition of Galois connections to replace the condition of adjointness by conditions 1 and 2,

or by condition 5 which also entails the monotonicity of γ.

Pointwise Galois Connections between Powersets

Lemma 1 *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 $\gamma(a) = \cup\alpha^{-1}(\downarrow a)$ forms a Galois connection $\mathcal{P}(\mathcal{C}) \underset{\gamma}{\overset{\alpha}{\rightleftarrows}} \mathcal{P}(\mathcal{A})$ between $(\mathcal{P}(\mathcal{C}), \subseteq)$ and $(\mathcal{P}(\mathcal{A}), \subseteq)$.

Pointwise Galois Connections between Powersets

Lemma 2 *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 $\gamma(a) = \cup \alpha^{-1}(\downarrow a)$ forms a Galois connection $\mathcal{P}(\mathcal{C}) \underset{\gamma}{\overset{\alpha}{\rightleftarrows}} \mathcal{P}(\mathcal{A})$ 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 $\gamma(a) = \cup \alpha^{-1}(\downarrow a)$ forms a Galois connection $\mathcal{P}(\mathcal{C}) \overset{\alpha}{\underset{\gamma}{\rightleftarrows}} \mathcal{P}(\mathcal{A})$ 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 $c = \gamma(a) = \cup \alpha^{-1}(\downarrow a)$, we need to prove that $c \in \alpha^{-1}(\downarrow a)$, i.e. $\alpha(c) \in \downarrow a$.

... □

Pointwise Galois Connections between Powersets

Lemma 4 *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 $\gamma(a) = \cup\alpha^{-1}(\downarrow a)$ forms a Galois connection $\mathcal{P}(\mathcal{C}) \overset{\alpha}{\underset{\gamma}{\rightleftarrows}} \mathcal{P}(\mathcal{A})$ 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 $c = \gamma(a) = \cup\alpha^{-1}(\downarrow a)$, we need to prove that $c \in \alpha^{-1}(\downarrow a)$, i.e. $\alpha(c) \in \downarrow a$. We know that $\alpha(c) = \bigcup_{e \in c} \alpha(\{e\}) = \bigcup_{e \in \cup\alpha^{-1}(\downarrow a)} \alpha(\{e\})$. For each e in $\cup\alpha^{-1}(\downarrow a)$ 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 5 *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 $\gamma(a) = \cup\alpha^{-1}(\downarrow a)$ forms a Galois connection $\mathcal{P}(\mathcal{C}) \overset{\alpha}{\underset{\gamma}{\rightleftarrows}} \mathcal{P}(\mathcal{A})$ 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 $c = \gamma(a) = \cup\alpha^{-1}(\downarrow a)$, we need to prove that $c \in \alpha^{-1}(\downarrow a)$, i.e. $\alpha(c) \in \downarrow a$. We know that $\alpha(c) = \bigcup_{e \in c} \alpha(\{e\}) = \bigcup_{e \in \cup\alpha^{-1}(\downarrow a)} \alpha(\{e\})$. For each e in $\cup\alpha^{-1}(\downarrow a)$ there exists $d \in \mathcal{P}(\mathcal{C})$ such that $e \in d$ and $\alpha(d) \subseteq a$, therefore $\alpha(\{e\}) \subseteq a$.

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

Systems Biology Markup Language SBML Models

Formally, the concrete domain of reaction models is the powerset of all possible reaction rules ordered by set inclusion :

Def. -3 *Given a finite set $\mathcal{M}$ of molecule names, the universe of reactions is the set of rules*

$$\begin{aligned}\mathcal{R} = \{e \texttt{ for } S\texttt{=>}S' \mid \quad & \textit{e is a kinetic expression,} \\ & \textit{and } S \textit{ and } S' \textit{ are solutions of molecules in } \mathcal{M}\}.\end{aligned}$$

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

Systems Biology Markup Language SBML Models

Formally, the concrete domain of reaction models is the powerset of all possible reaction rules ordered by set inclusion :

Def. -2 *Given a finite set $\mathcal{M}$ of molecule names, the universe of reactions is the set of rules*

$$\mathcal{R} = \{e \texttt{ for } S\texttt{=>}S' \mid \quad e \textit{ is a kinetic expression,} \\ \textit{and } S \textit{ and } S' \textit{ are solutions of molecules in } \mathcal{M}\}.$$

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

In the SBML exchange format, no semantics are defined.

In BIOCHAM, three semantics are considered:

1. boolean : non-deterministic asynchronous transition system
2. differential : ODE (or hybrid system)
3. stochastic : continuous time Markov chain.

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, instead of concentrations, 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. -1 *Let a discrete state be a vector of integers of dimension* $|\mathcal{M}|$. *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 $\mathcal{D}_\mathcal{S} = (\mathcal{P}(\mathcal{S}), \subseteq)$.

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

Galois Connection Syntactical → Stochastic Domain

Proposition 0 *Let $\alpha_{\mathcal{RS}} : \mathcal{C}_{\mathcal{R}} \rightarrow \mathcal{D}_{\mathcal{S}}$ be the function associating to a reaction model the state transition graph labelled with thte $\tau_{i,j}$'s. Let $\gamma_{\mathcal{RS}}(s) = \cup {\alpha_{\mathcal{RS}}}^{-1}(\downarrow s)$. $\mathcal{C}_{\mathcal{R}} \underset{\gamma_{\mathcal{RS}}}{\overset{\alpha_{\mathcal{RS}}}{\rightleftarrows}} \mathcal{D}_{\mathcal{S}}$ is a Galois connection.*

Galois Connection Syntactical → Stochastic Domain

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

PROOF: Simply note that $\alpha_{\mathcal{RS}}$ is defined by its union on each rule of the concrete model and apply Lemma 1. □

Stochastic Semantics Domain

$\alpha_{\mathcal{RS}}$ is not one-to-one.

Stochastic Semantics Domain

$\alpha_{\mathcal{RS}}$ is not one-to-one.

For instance, the reaction models $m1 = \{$ `e for A => B`$\}$ and $m2 = m1 \cup \{$ `e for 2*A => 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.

Stochastic Semantics Domain

$\alpha_{\mathcal{RS}}$ is not one-to-one.

For instance, the reaction models $m1 = \{$ `e for A => B`$\}$ and $m2 = m1 \cup \{$ `e for 2*A => 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.

$\alpha_{\mathcal{RS}}$ is neither onto

Stochastic Semantics Domain

$\alpha_{\mathcal{RS}}$ is not one-to-one.

For instance, the reaction models $m1 = \{$ `e for A => B`$\}$ and $m2 = m1 \cup \{$ `e for 2*A => 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.

$\alpha_{\mathcal{RS}}$ is neither onto as the stochastic transitions obtained from a reaction model enjoy some particular properties, such as for instance the following stability property w.r.t. the number of molecules in the states:

Proposition 2 *If two states S_1, S_2 are such that $S_1 \leq S_2$ pointwise, then for any model m and all transitions S_i, τ_i such that $(S_1, S_i, \tau_i) \in \alpha_{\mathcal{RS}}(m)$, there exist states $S_j = S_i + S_2 - S_1$ (pointwise) such that $(S_2, S_j, \tau_i) \in \alpha_{\mathcal{RS}}(m)$, i.e. all rules that apply in S_1 apply in S_2 with the same changes.*

Proof: By definition of $\alpha_{\mathcal{RS}}$. □

Discrete Semantics

Def. 3 *The universe $\mathcal{D}$ of discrete transitions is the set of pairs of discrete states. The domain of discrete transitions is $\mathcal{D}_{\mathcal{D}} = (\mathcal{P}(\mathcal{D}), \subseteq)$.*

The discrete semantics is the classical Petri net semantics of reaction models [RML93ismb,SHK06bmcbi,Chaouiya07bioinfo,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 [SPM02bioinfo] has been shown in [ZS03insilicobio] to be equivalent to the classical analysis of Petri nets by T-invariants.

Discrete Semantics

Proposition 4 *Let* $\alpha_{\mathcal{SD}} : \mathcal{D}_{\mathcal{S}} \rightarrow \mathcal{D}_{\mathcal{D}}$ *be the function associating to a set of stochastic transitions the discrete transitions obtained by projection on the two first components, and* $\gamma_{\mathcal{SD}}(d) = \cup {\alpha_{\mathcal{SD}}}^{-1}(\downarrow d)$. $\mathcal{D}_{\mathcal{S}} \underset{\gamma_{\mathcal{SD}}}{\overset{\alpha_{\mathcal{SD}}}{\rightleftarrows}} \mathcal{D}_{\mathcal{D}}$ *is a Galois connection.*

Discrete Semantics

Proposition 5 *Let $\alpha_{\mathcal{SD}} : \mathcal{D}_{\mathcal{S}} \rightarrow \mathcal{D}_{\mathcal{D}}$ be the function associating to a set of stochastic transitions the discrete transitions obtained by projection on the two first components, and $\gamma_{\mathcal{SD}}(d) = \cup {\alpha_{\mathcal{SD}}}^{-1}(\downarrow d)$. $\mathcal{D}_{\mathcal{S}} \underset{\gamma_{\mathcal{SD}}}{\overset{\alpha_{\mathcal{SD}}}{\rightleftarrows}} \mathcal{D}_{\mathcal{D}}$ is a Galois connection.*

PROOF: Here again it suffices to note that $\alpha_{\mathcal{SD}}$ is defined by its union on each single stochastic transition of the concrete model and to apply Lemma 1. □

Remark that $\alpha_{\mathcal{SD}}$ is onto, but not one-to-one as the transition rates are simply forgotten.

Boolean Semantics

Def. 6 *Let a boolean state be a vector of booleans of dimension* $|\mathcal{M}|$ *indicating the presence of each molecule in the state. 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

Def. 7 *Let a boolean state be a vector of booleans of dimension $|\mathcal{M}|$ indicating the presence of each molecule in the state. 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 $\alpha_{\mathcal{NB}} : \mathbb{N}^{|\mathcal{M}|} \rightarrow \mathbb{B}^{|\mathcal{M}|}$ be the zero/non-zero abstraction (or threshold abstraction) from the integers to the booleans, and its pointwise extension from discrete states to boolean states.

Proposition 8 *Let $\alpha_{\mathcal{DB}} : \mathcal{D}_{\mathcal{D}} \rightarrow \mathcal{D}_{\mathcal{B}}$ be the set extension of $\alpha_{\mathcal{NB}}$. Let $\gamma_{\mathcal{DB}}(b) = \cup {\alpha_{\mathcal{DB}}}^{-1}(\downarrow b)$. $\mathcal{D}_{\mathcal{D}} \underset{\gamma_{\mathcal{DB}}}{\overset{\alpha_{\mathcal{DB}}}{\rightleftarrows}} \mathcal{D}_{\mathcal{B}}$ is a Galois connection.*

Proof: As before, note that $\alpha_{\mathcal{DB}}$ is defined by its union on each transition of the concrete model and apply Lemma 1. □

BIOCHAM Boolean Semantics

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

For instance, a rule like `A+B=>C+D` is interpreted by four boolean transition rules :

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

Note that in Boolean Petri nets, or in Pathway Logic, complete consumption is always assumed.

Representing all possible consumptions is necessary for getting an over-approximation result.

BIOCHAM Boolean Semantics in the hierarchy of semantics

Proposition 9 *For any reaction model R, $\alpha_{\mathcal{DB}}(\alpha_{\mathcal{SD}}(\alpha_{\mathcal{RS}}(R))) \subseteq S_{BB}$.*

BIOCHAM Boolean Semantics in the hierarchy of semantics

Proposition 10 *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=>S'. By abuse of notation we will denote by S and S' the discrete states corresponding to solutions of same name. We have

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

... □

BIOCHAM Boolean Semantics in the hierarchy of semantics

Proposition 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=>S'. By abuse of notation we will denote by S and S' the discrete states corresponding to solutions of same name. We have
$\alpha_{\mathcal{RS}}(R) = \{(S_i, S_j, e) | S_i \geq S,\ S_j = S_i - S + S'\}$ and thus
$\alpha_{\mathcal{SD}}(\alpha_{\mathcal{RS}}(R)) = \{(S_i, S_j) | S_i \geq S,\ S_j = S_i - S + S'\}$,

... □

BIOCHAM Boolean Semantics in the hierarchy of semantics

Proposition 12 *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=>S'. By abuse of notation we will denote by S and S' the discrete states corresponding to solutions of same name. We have
$\alpha_{\mathcal{RS}}(R) = \{(S_i, S_j, e) | S_i \geq S,\ S_j = S_i - S + S'\}$ and thus
$\alpha_{\mathcal{SD}}(\alpha_{\mathcal{RS}}(R)) = \{(S_i, S_j) | S_i \geq S,\ S_j = S_i - S + S'\}$, which leads to
$\alpha_{\mathcal{DB}}(\alpha_{\mathcal{SD}}(\alpha_{\mathcal{RS}}(R))) = \{(S'_i, S'_j) | S_i \geq S,\ S_j = S_i - S + S',\ S'_i = \alpha_{\mathcal{NB}}(S_i),\ S'_j = \alpha_{\mathcal{NB}}(S_j)\}$.

... □

BIOCHAM Boolean Semantics in the hierarchy of semantics

Proposition 13 *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=>S'. By abuse of notation we will denote by S and S' the discrete states corresponding to solutions of same name. We have
$\alpha_{\mathcal{RS}}(R) = \{(S_i, S_j, e) | S_i \geq S,\ S_j = S_i - S + S'\}$ and thus
$\alpha_{\mathcal{SD}}(\alpha_{\mathcal{RS}}(R)) = \{(S_i, S_j) | S_i \geq S,\ S_j = S_i - S + S'\}$, which leads to
$\alpha_{\mathcal{DB}}(\alpha_{\mathcal{SD}}(\alpha_{\mathcal{RS}}(R))) = \{(S'_i, S'_j) | S_i \geq S,\ S_j = S_i - S + S',\ S'_i = \alpha_{\mathcal{NB}}(S_i),\ S'_j = \alpha_{\mathcal{NB}}(S_j)\}$. Since $S_{BB} = \{(T, T') | T \geq \alpha_{\mathcal{NB}}(S),\ \alpha_{\mathcal{NB}}(S') \vee (T \wedge \neg\alpha_{\mathcal{NB}}(S)) \leq T' \leq \alpha_{\mathcal{NB}}(T) \vee \alpha_{\mathcal{NB}}(S')\}$ the property holds as $S_i \geq S$ implies $S'_i \geq \alpha_{\mathcal{NB}}(S)$, and since $S_i \geq S$ we have
$S_j = S_i - S + S' \Rightarrow S_i - S + S' \leq S_j \leq S_i + S' \Rightarrow \alpha_{\mathcal{NB}}(S_i - S + S') = \alpha_{\mathcal{NB}}(S') \vee (\alpha_{\mathcal{NB}}(S_i) \wedge \neg\alpha_{\mathcal{NB}}(S)) \leq S'_j \leq \alpha_{\mathcal{NB}}(S_i + S') = \alpha_{\mathcal{NB}}(S_i) \vee \alpha_{\mathcal{NB}}(S')$
□

Differential Semantics ?

The differential semantics of reaction models interprets a set of reaction rules $\{e_i \texttt{ for } S_i{=}{>}S'_i\}_{i=1,\dots,n}$ over molecular concentration variables $\{x_1, ..., x_m\}$, by the following system of Ordinary Differential Equations (ODE):

$$dx_k/dt = \sum_{i=1}^{n} r_i(x_k) * e_i - \sum_{j=1}^{n} l_j(x_k) * e_j$$

where we recall that $r_i(x_k)$ (resp. l_i) is the stoichiometric coefficient of x_k in the right (resp. left) member of rule i.

- synchronous semantics (evolution of variables in parallel)
- deterministic semantics (average behavior)
- not compatible with the rule set inclusion ordering
- infinite number of molecules
- infinitesimal time steps

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 C is a Galois connection $\mathcal{C} \rightarrow_{\alpha} \mathcal{A}$.

The type inference problem is

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

TPUT 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 $\mathcal{C} \rightarrow_{\alpha} \mathcal{A}$.

The type inference problem is

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

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

The type checking problem is,

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

TPUT determine whether $x \sqsubseteq_{\mathcal{C}} \gamma(y)$
(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 $\mathcal{C} \rightarrow_\alpha \mathcal{A}$.

The type inference problem is

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

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

The type checking problem is,

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

TPUT determine whether $x \sqsubseteq_{\mathcal{C}} \gamma(y)$
(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 $\mathcal{A}_{\mathcal{F}} = \mathcal{P}(\{\text{kinase}(A) | A \in \mathcal{M}\} \cup \{\text{phosphatase}(A) | A \in \mathcal{M}\})$

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

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

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

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

if `C` is strictly more phosphorylated than `A`

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

if `C` is strictly less phosphorylated than `A`

Type Checking/Inference of Protein Functions

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

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

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

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

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

if `C` is strictly more phosphorylated than `A`

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

if `C` is strictly less phosphorylated than `A`

Proposition 14 *$\alpha_{\mathcal{F}}$ can be computed in $O(n^2)$ time where n is the number of rules.*

Proposition 15 *Let $\gamma_{\mathcal{F}}(f) = \cup {\alpha_{\mathcal{F}}}^{-1}(\downarrow f)$, $\mathcal{C}_{\mathcal{R}} \underset{\gamma_{\mathcal{F}}}{\overset{\alpha_{\mathcal{F}}}{\rightleftarrows}} \mathcal{A}_{\mathcal{F}}$ 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 $phosphatase(\tau) \preceq phosphotase$.

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~{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.

- 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.

Use of Protein Functions Types

- Check the consistency of reaction models.
- Restrict the search space for reaction rules in model revision or network inference.
- Build modules according to protein functions

Type Checking/Inference of Influence Graphs

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

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

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

Type Checking/Inference of Influence Graphs

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

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

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

$$\begin{aligned}
\alpha_{\mathcal{RI}}(\{\mathtt{A} + \mathtt{B} \mathtt{=>} \mathtt{C}\}) = \{ \quad & \mathrm{A} \xrightarrow{-} \mathrm{B},\ \mathrm{A} \xrightarrow{-} \mathrm{A},\ \mathrm{B} \xrightarrow{-} \mathrm{A}, \\
& \mathrm{B} \xrightarrow{-} \mathrm{B},\ \mathrm{A} \xrightarrow{+} \mathrm{C},\ \mathrm{B} \xrightarrow{+} \mathrm{C}\} \\
\alpha_{\mathcal{RI}}(\{\mathtt{A} = [\mathtt{C}] \mathtt{=>} \mathtt{B}\}) = \{ \quad & \mathrm{C} \xrightarrow{-} \mathrm{A},\ \mathrm{A} \xrightarrow{-} \mathrm{A},\ \mathrm{A} \xrightarrow{+} \mathrm{B},\ \mathrm{C} \xrightarrow{+} \mathrm{B}\} \\
\alpha_{\mathcal{RI}}(\{\mathtt{A} = [\mathtt{B}] \mathtt{=>} \_\}) = \{ \quad & \mathrm{B} \xrightarrow{-} \mathrm{A},\ \mathrm{A} \xrightarrow{-} \mathrm{A}\} \\
\alpha_{\mathcal{RI}}(\{\_ = [\mathtt{B}] \mathtt{=>} \mathtt{A}\}) = \{ \quad & \mathrm{B} \xrightarrow{+} \mathrm{A}\}
\end{aligned}$$

Type Checking/Inference of Influence Graphs

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

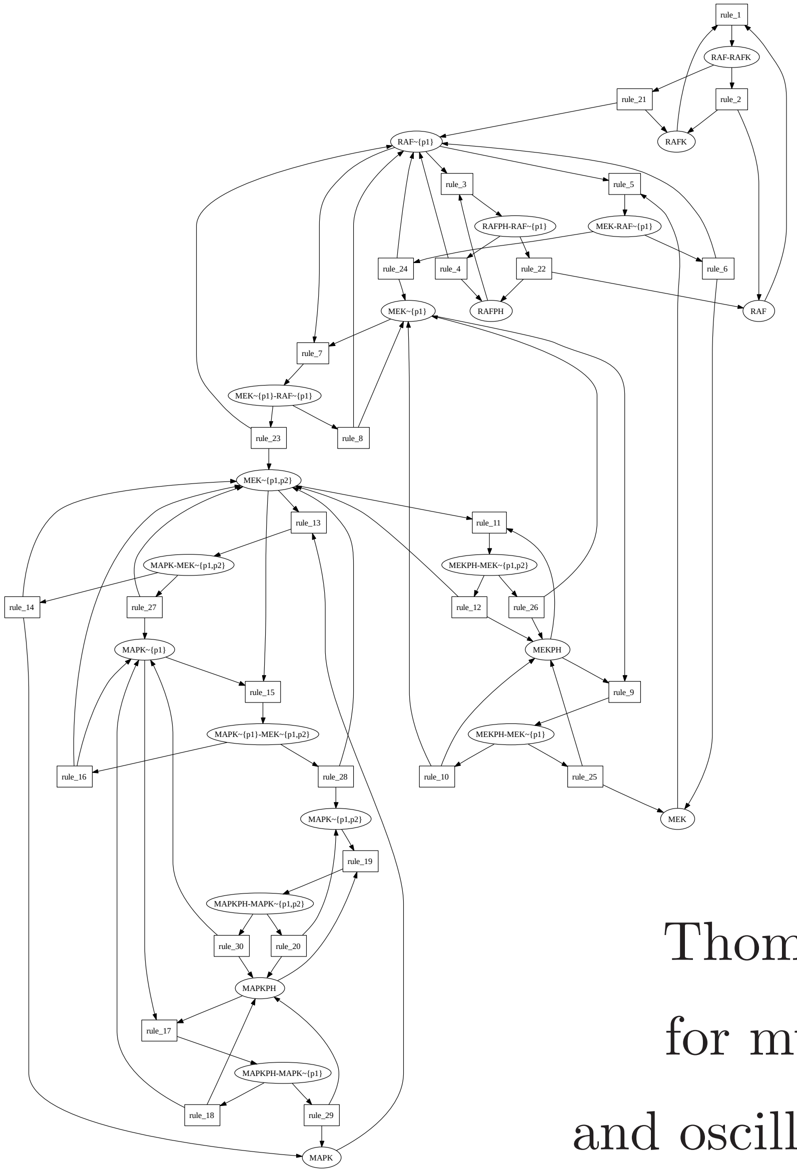
The influence graph of a reaction model is defined by $\alpha_{\mathcal{RI}} : \mathcal{C}_{\mathcal{R}} \rightarrow \mathcal{A}_{\mathcal{I}}$

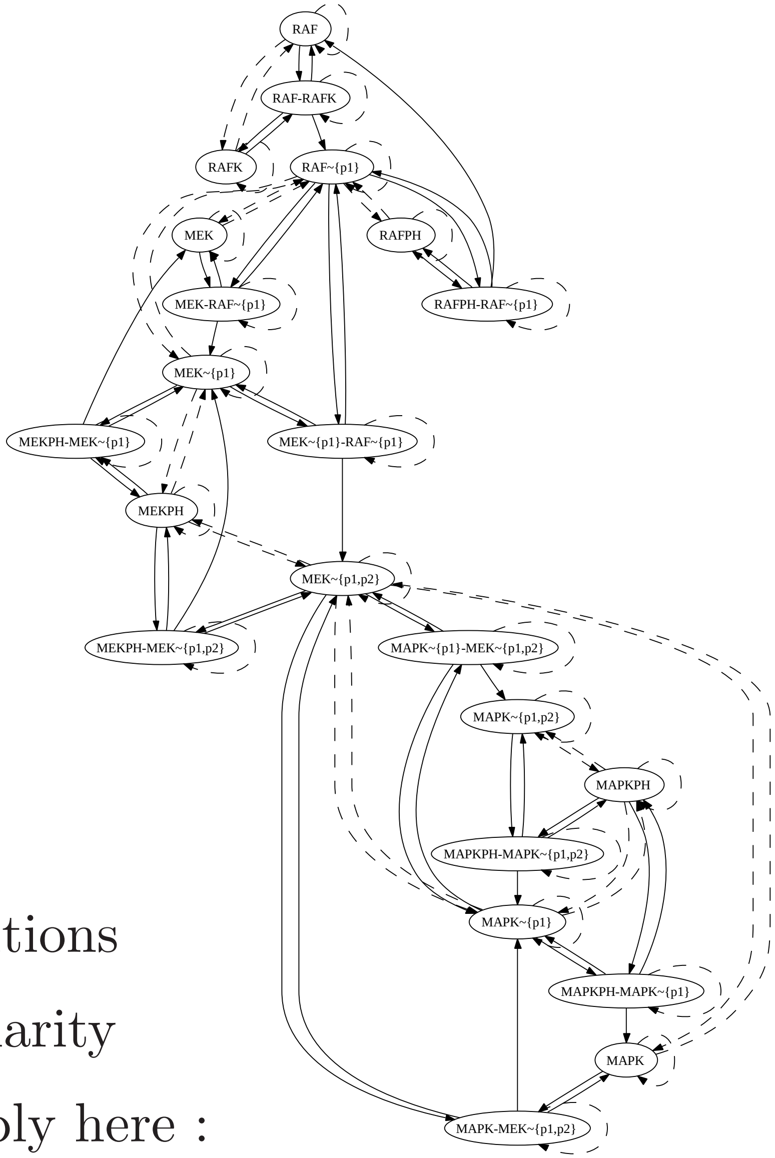
$$\begin{aligned}
\alpha_{\mathcal{RI}}(x) = \quad & \{A \xrightarrow{-} B && \mid \exists (e_i \text{ for } S_i \Rightarrow S_i') \in x, \\
& && l_i(A) > 0 \text{ and } r_i(B) - l_i(B) < 0\} \\
& \cup\{A \xrightarrow{+} B && \mid \exists (e_i \text{ for } S_i \Rightarrow S_i') \in x, \\
& && l_i(A)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

Proposition 16 *$\alpha_{\mathcal{RI}}$ can be computed in $O(n)$ time where n is the number of rules.*

Proposition 17 *Let $\gamma_{\mathcal{RI}}(f) = \cup{\alpha_{\mathcal{RI}}}^{-1}(\downarrow f)$, $\mathcal{C}_{\mathcal{R}} \underset{\gamma_{\mathcal{RI}}}{\overset{\alpha_{\mathcal{RI}}}{\rightleftarrows}} \mathcal{A}_{\mathcal{I}}$ is a Galois connection.*

MAPK model: Reaction Graph $\rightarrow_{\alpha}$ Influence Graph



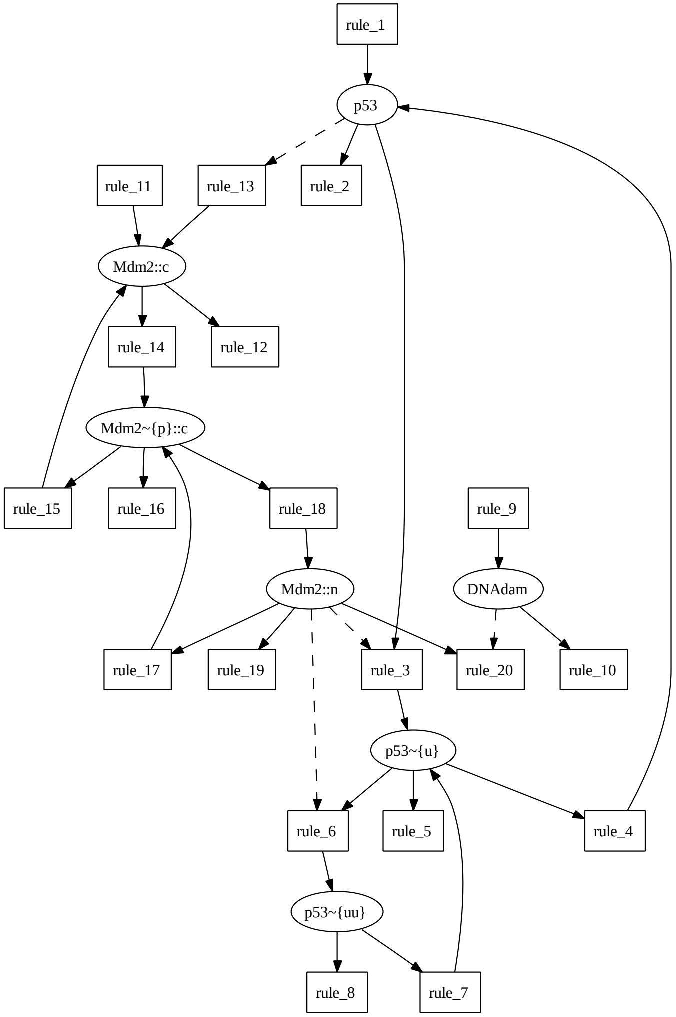


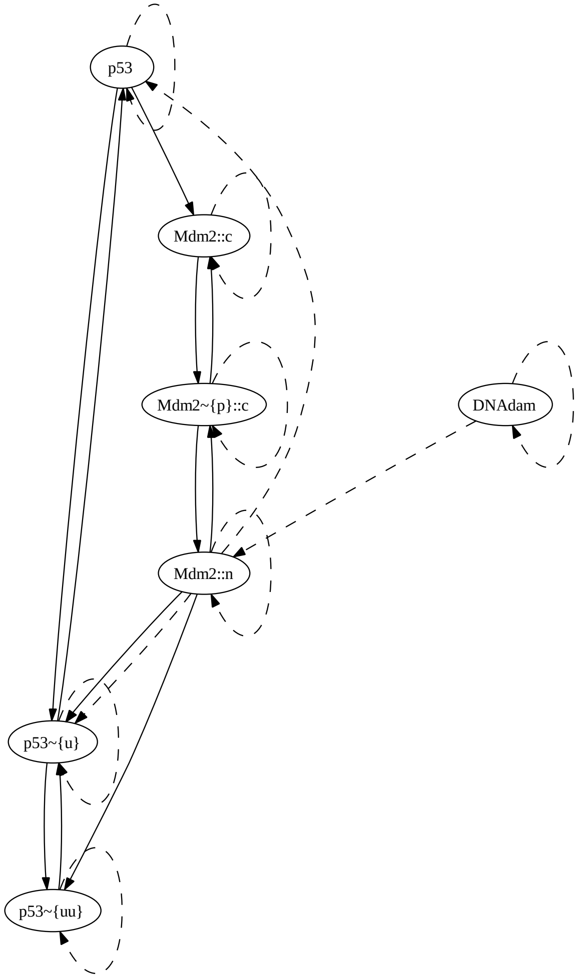
Thomas's conditions

for multistationarity

and oscillations apply here :

P53-Mdm2: Reaction Graph $\rightarrow_\alpha$ Influence Graph





Inhitions hidden in the kinetic expressions are missed

Use of Influence Types

- Check the consistency of reaction models
- Analyze the dynamics of the reaction model (multistationarity, oscillations, ...)
- Restrict the search space for reaction rules in model revision or network inference
- Build modules according to the influence graph

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

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

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

Increasing Kinetics

Def. 19 *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, $e_i = k * \Pi {x_i}^{l_i}$, are increasing

Hill's kinetics (and Michaelis-Menten kinetics when $n = 1$) $e_i = V_m * {x_s}^n / (K_m + {x_s}^n)$ where $V_m = k * (x_e + x_e * x_s / K_m)$ for an enzymatic reaction $x_s = [x_e] => x_p$, are also increasing.

Comparison to the Syntactical Influence Graph

Theorem 20 (Over-approximation) *For any reaction model x with increasing kinetics, $\alpha_{\mathcal{JI}} \circ \beta(x) \subseteq \alpha_{\mathcal{RI}}(x)$.*

PROOF: If $(A \xrightarrow{+} B) \in \alpha_{\mathcal{JI}} \circ \beta(x)$ 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 $(A \xrightarrow{+} B) \in \alpha_{\mathcal{RI}}(x)$. If on the contrary $r_i(B) - l_i(B) < 0$ then $\partial e_i/\partial A < 0$, impossible.

If $(A \xrightarrow{-} B) \in \alpha_{\mathcal{JI}} \circ \beta(x)$ 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 $(A \xrightarrow{+} B) \in \alpha_{\mathcal{RI}}(x)$.

□

Comparison to the Syntactical Influence Graph

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

Comparison to the Syntactical Influence Graph

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

For instance let x be the following model :

$$\begin{array}{lll} k_1 * A & \text{for} & A => B \\ k_2 * A & \text{for} & \_ = [A] => A \end{array}$$

We have $\alpha_{\mathcal{RI}}(x) = \{\mathrm{A} \overset{+}{\rightarrow} \mathrm{B},\ \mathrm{A} \overset{+}{\rightarrow} \mathrm{A},\ \mathrm{A} \overset{-}{\rightarrow} \mathrm{A}\}$, however $\dot{A} = (k_2 - k_1) * A$, hence $\partial \dot{A} / \partial A$ can be made always positive or always negative or always null, resulting in the absence from $\alpha_{\mathcal{JI}} \circ \beta(x)$ of, respectively, $A \overset{-}{\rightarrow} A$, $A \overset{+}{\rightarrow} A$ or both.

Non-monotonicity of β

β is not monotonic since adding rules can compensate an existing rule in the differential expression and eliminate terms in the differential equations.

The differential semantics is thus not an abstraction of the reaction models ordered by set inclusion in the sense of abstract interpretation.

The above case shows that $\alpha_{\mathcal{J}\mathcal{I}} \circ \beta$ applied to the first rule contains $\mathrm{A} \xrightarrow{-} \mathrm{A}$, whereas its application to the set of two rules (greater in $\mathcal{C}_{\mathcal{R}}$) may not.

A sufficient condition for β to be monotonic is that in the model no kinetic expression can compensate another one in the Jacobian. That is :
$\forall x_i, x_j \exists ? k$ s.t. $r_k(x_i) \neq l_k(x_i)$ and $\partial e_k / \partial x_j \neq 0$.

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 $l_i(x_k) > 0$*

Note that strongly increasing implies increasing.

Proposition 22 *Mass action law, Michaelis Menten, and Hill kinetics are strongly increasing.*

Theorem 23 *If x has strongly increasing kinetics and no molecule is at the same time an activator and an inhibitor of the same target molecule, then $\alpha_{\mathcal{RI}}(x) = \alpha_{\mathcal{JI}} \circ \beta(x)$.*

Type Checking/Inference of Location Neighborhood

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

$\alpha_{\mathcal{RN}}(\texttt{e for A}_1 + \cdots + \texttt{A}_m \texttt{=>} \texttt{A}_{m+1} + \cdots + \texttt{A}_n) =$

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

Proposition 24 *$\alpha_{\mathcal{RN}}$ 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.

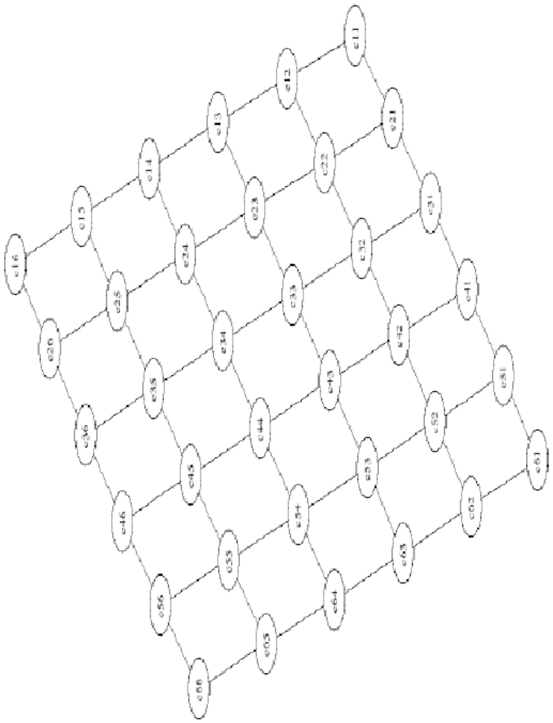
BIOMD45.xml: `neighbor(cytosol,extracellular)`, `neighbor(cytosol, vesicula1)`, `neighbor(cytosol, vesicula1)` inferred and checked consistent with the outside attributes.

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.
...
```



Use of Location Topology Types

- Check the consistency of reaction models.
- Restrict the search space for reaction rules in model revision or network inference.
- Build modules according to the locations