Semantical and Algorithmic Aspects of the Living

François Fages
Constraint Programming Group
INRIA Paris-Rocquencourt
http://contraintes.inria.fr/

To tackle the complexity of biological systems, investigate:

- Programming Theory Concepts
- Formal Methods of Circuit and Program Verification
- Automated Reasoning Tools

Prototype implementation in the Biochemical Abstract Machine BIOCHAM v3.2

- modeling environment for Systems Biology

Systems Biology ?

"*Systems Biology aims at systems-level understanding [of biological processes] which requires a set of principles and methodologies that links the behaviors of molecules to systems characteristics and functions.*"

H. Kitano, ICSB 2000

Analyze post-genomic RNA and protein interaction data produced with high-throughput technologies available in databases like GO, KEGG, BioCyc, etc.

Integrate heterogeneous data about a specific problem

Understand and predict the behaviors of large networks of genes and proteins

→ Systems Biology Markup Language (SBML): model exchange format
→ Model repositories: e.g. `biomodels.net` 261 curated models of cell processes
→ Simulation tools

Issue of Abstraction in Systems Biology

Models are built in Systems Biology with two contradictory perspectives :

Issue of Abstraction in Systems Biology

Models are built in Systems Biology with two contradictory perspectives :

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

Issue of Abstraction in Systems Biology

Models are built in Systems Biology with two contradictory perspectives :

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

2) Models for making predictions : the more abstract the better !

Issue of Abstraction in Systems Biology

Models are built in Systems Biology with two contradictory perspectives :

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

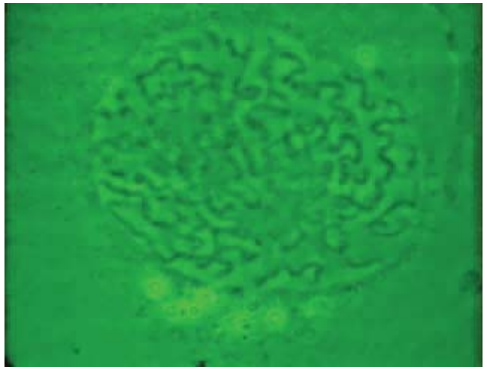
2) Models for making predictions : the more abstract the better !

These perspectives can be reconciled by organizing models and formalisms in hierarchies of abstractions.

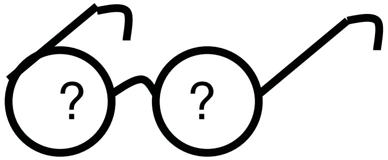
To understand a system is not to know everything about it but to know abstraction levels that are sufficient for answering questions about it

Semantics of Living Processes ?

Formally, the semantics of a system depend on *our choice* of observables.



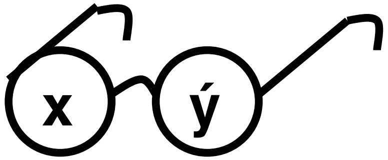
Mitosis movie [Lodish et al. 03]

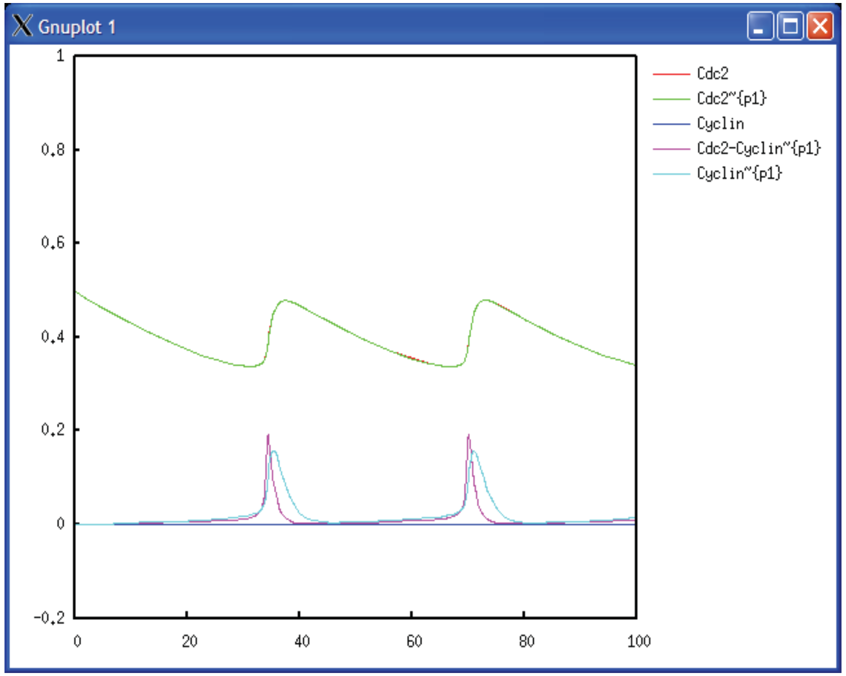


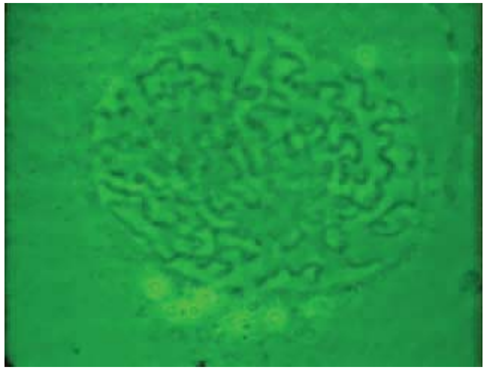
Continuous Differential Semantics

Formally, the semantics of a system depend on *our choice* of observables.

- Concentrations of molecules
- Reaction rates
- Ordinary Differential Equation (ODE) model





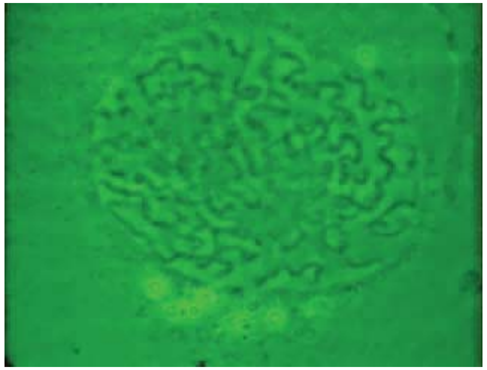


Mitosis movie [Lodish et al. 03]

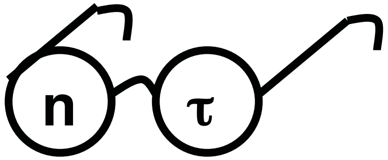
Stochastic Semantics

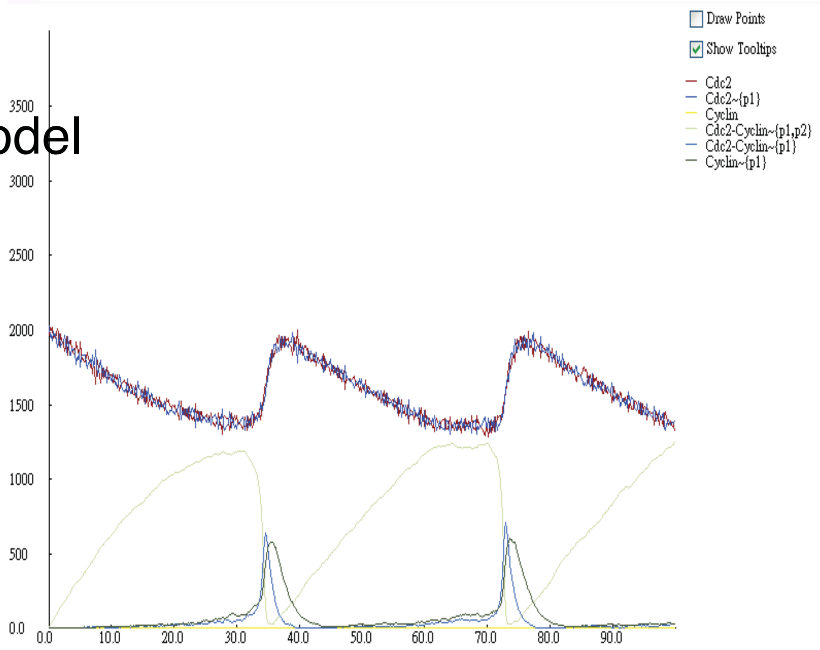
Formally, the semantics of a system depend on *our choice* of observables.

- (Small) numbers of molecules
- Probabilities of reaction
- Continuous Time Markov Chain (CTMC) model



Mitosis movie [Lodish et al. 03]

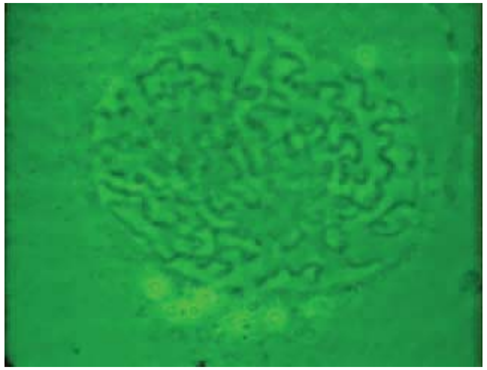




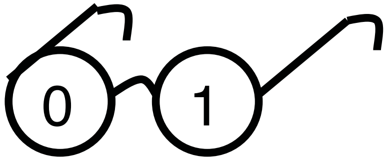
Boolean Semantics

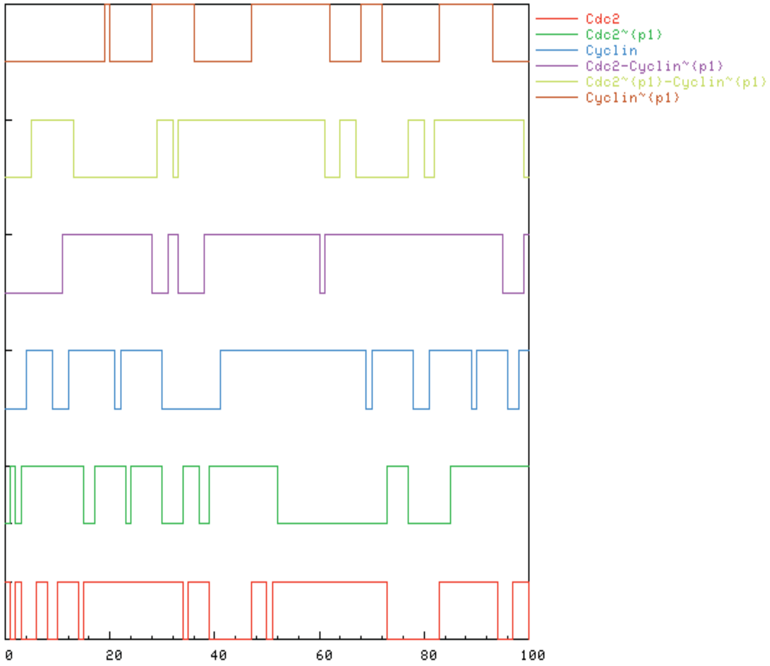
Formally, the semantics of a system depend on *our choice* of observables.

- Presence/absence of molecules
- Boolean transition model



Mitosis movie [Lodish et al. 03]

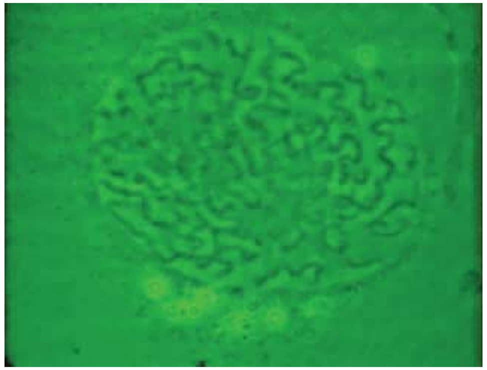




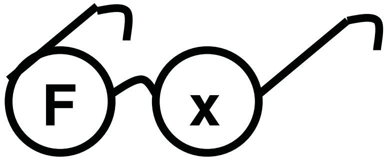
Propositional Temporal Logic

Formally, the semantics of a system depend on *our choice* of observables.

- Presence/absence of molecules
- Temporal logic formulas on Boolean traces



Mitosis movie [Lodish et al. 03]



$\mathbf{F}\ x$

$\mathbf{F}\ (x \wedge \mathbf{F}\ (\neg\ x \wedge y))$

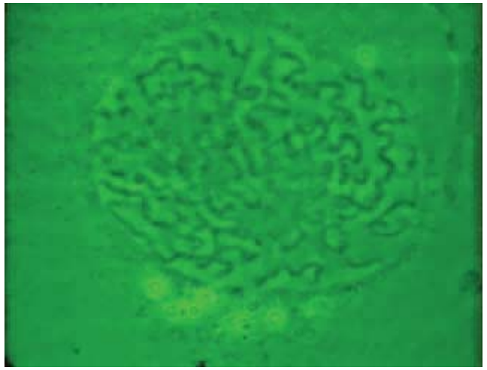
$\mathbf{FG}\ (x \vee y)$

…

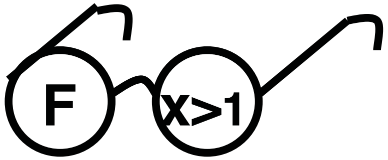
Constraint Temporal Logic LTL(R)

Formally, the semantics of a system depend on *our choice* of observables.

- Concentrations of molecules
- Temporal logic with constraints over R on quantitative traces



Mitosis movie [Lodish et al. 03]



F (x >0.2)

F (x >0.2 ^ **F** (x<0.1 ^ y>0.2))

FG (x>0.2 v y>0.2)

…

A Logical Paradigm for Systems Biology

Biological process model = (Quantitative) Transition System

Biological property = Temporal Logic Formula

Biological validation = Model-checking

Model inference = TL Constraint Solving

[Eker, Lincoln et al. PSB'02] [Chabrier Fages CMSB'03] [Bernot et al. TCS'04] …

A Logical Paradigm for Systems Biology

Biological process model = (Quantitative) Transition System

Biological property = Temporal Logic Formula

Biological validation = Model-checking

Model inference = TL Constraint Solving

[Eker, Lincoln et al. PSB’02] [Chabrier Fages CMSB’03] [Bernot et al. TCS’04] …

Model: ⟷

- Boolean
- Differential
- Stochastic

BIOCHAM ⟷

- simulation
- query evaluation
- rule learning
- parameter learning
- robustness measure

Biological Properties Specification:

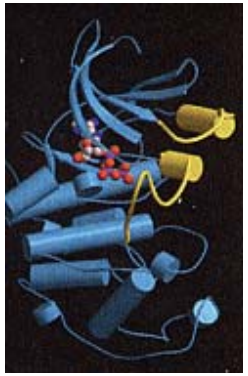
- Temporal logic CTL
- LTL(R), QFLTL(R) constraints
- CSL

Overview of the Talk

1. *Rule-based Modeling* of Biochemical Systems
    1. Syntax of molecules, compartments and reactions
    2. Hierarchy of semantics: boolean, differential, stochastic
    3. Cell cycle control models
2. *Temporal Logic Language* for formalizing Biological Properties
    1. CTL for the boolean semantics
    2. LTL(R) for the differential semantics
    3. Parameter inference from LTL(R) properties
    4. Robustness w.r.t. LTL(R) properties
3. *Conclusion* and collaborations

- F. Fages, S. Soliman. Formal Cell Biology in Biocham. Springer *LNBI* 5016, 2008
- A. Rizk, G. Batt, F. Fages, S. Soliman. *Bioinformatics* 25:169-178, 2009
- S. Gay, S. Soliman, F. Fages*. Bioinfomatics* 26:575-581, 2010

Formal Proteins: Syntax



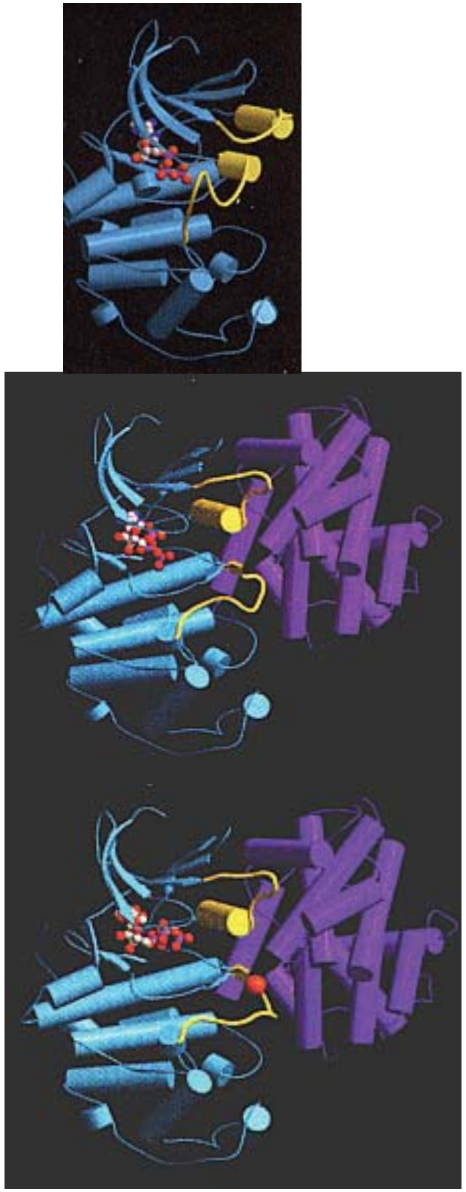
- Cyclin dependent kinase 1 `Cdk1`
  (free, inactive)

Formal Proteins: Syntax

- Cyclin dependent kinase 1 (free, inactive) `Cdk1`

- Complex Cdk1-Cyclin B (low activity) `Cdk1-CycB`

Formal Proteins: Syntax



- Cyclin dependent kinase 1 `Cdk1`
  (free, inactive)

- Complex Cdk1-Cyclin B `Cdk1-CycB`
  (low activity)

- Phosphorylated form `Cdk1~{thr161}-CycB`
  at site threonine 161
  (high activity)
  "Mitosis-Promoting Factor"
  phosphorylates actin in microtubules → nuclear division

Formal Reaction Rules

Complexation: A + B => A-B.

`cdk1+cycB => cdk1-cycB`

Decomplexation A-B => A + B.

Formal Reaction Rules

Complexation: A + B => A-B.

Decomplexation A-B => A + B.

```
cdk1+cycB => cdk1-cycB
```

Phosphorylation: A =[K]=> A~{p}.

Dephosphorylation A~{p} =[P]=> A.

```
Cdk1-CycB =[Myt1]=> Cdk1~{thr161}-CycB
Cdk1~{thr14,tyr15}-CycB =[Cdc25~{Nterm}]=> Cdk1-CycB
```

Formal Reaction Rules

Complexation: A + B => A-B.

```
cdk1+cycB => cdk1-cycB
```

Decomplexation A-B => A + B.

Phosphorylation: A =[K]=> A~{p}.

Dephosphorylation A~{p} =[P]=> A.

```
Cdk1-CycB =[Myt1]=> Cdk1~{thr161}-CycB
Cdk1~{thr14,tyr15}-CycB =[Cdc25~{Nterm}]=> Cdk1-CycB
```

Synthesis: _ =[G]=> A.

```
_ =[#E2-E2f13-Dp12]=> cycA
```

Degradation: A =[C]=> _.

```
cycE =[UbiPro]=> _
```

(not for `cycE-cdk2` which is stable)

Formal Reaction Rules

Complexation: A + B => A-B.

`cdk1+cycB => cdk1-cycB`

Decomplexation A-B => A + B.

Phosphorylation: A =[K]=> A~{p}.

`Cdk1-CycB =[Myt1]=> Cdk1~{thr161}-CycB`

`Cdk1~{thr14,tyr15}-CycB =[Cdc25~{Nterm}]=> Cdk1-CycB`

Dephosphorylation A~{p} =[P]=> A.

Synthesis: _ =[G]=> A.

`_ =[#E2-E2f13-Dp12]=> cycA`

Degradation: A =[C]=> _.

`cycE =[UbiPro]=> _`

(not for `cycE-cdk2` which is stable)

Transport: A::L1 => A::L2.

`Cdk1~{p}-CycB::cytoplasm => Cdk1~{p}-CycB::nucleus`

Reaction Rule Models $\{ e_i \texttt{ for } l_i \texttt{ => } r_i \}_{i \in I}$

SBML reaction rules with kinetics: **`k*[A]*[B] for A + B => C`**

Reaction Rule Models $\{ e_i$ `for` l_i => $r_i \}_{i \in I}$

SBML reaction rules with kinetics: **`k*[A]*[B] for A + B => C`**

Interpretations at four abstraction levels:

1. Differential Semantics: concentrations
   - Ordinary Differential Equations
   - Det. continuous hybrid automata

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

Reaction Rule Models $\{ e_i \texttt{ for } l_i \Rightarrow r_i \}_{i \in I}$

SBML reaction rules with kinetics: **`k*[A]*[B] for A + B => C`**

Interpretations at four abstraction levels:

1. Differential Semantics: concentrations
   - Ordinary Differential Equations
   - Det. continuous hybrid automata

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

2. Stochastic Semantics: numbers of molecules
   - Continuous time Markov chain

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

Reaction Rule Models $\{ e_i \ \texttt{for} \ l_i \Rightarrow r_i \}_{i \in I}$

SBML reaction rules with kinetics: **`k*[A]*[B] for A + B => C`**

Interpretations at four abstraction levels:

1. Differential Semantics: concentrations
    - Ordinary Differential Equations
    - Det. continuous hybrid automata

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

2. Stochastic Semantics: numbers of molecules
    - Continuous time Markov chain

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

3. Discrete Semantics: numbers of molecules
    - Multiset rewriting, Petri net
    - CHAM [Berry Boudol 90] [Banatre Le Metayer 86]

A , B → C++ A- - B- -

Reaction Rule Models $\{ e_i \; \texttt{for} \; l_i \Rightarrow r_i \}_{i \in I}$

SBML reaction rules with kinetics: **`k*[A]*[B] for A + B => C`**

Interpretations at four abstraction levels:

1. Differential Semantics: concentrations
   - Ordinary Differential Equations
   - Det. continuous hybrid automata

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

2. Stochastic Semantics: numbers of molecules
   - Continuous time Markov chain

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

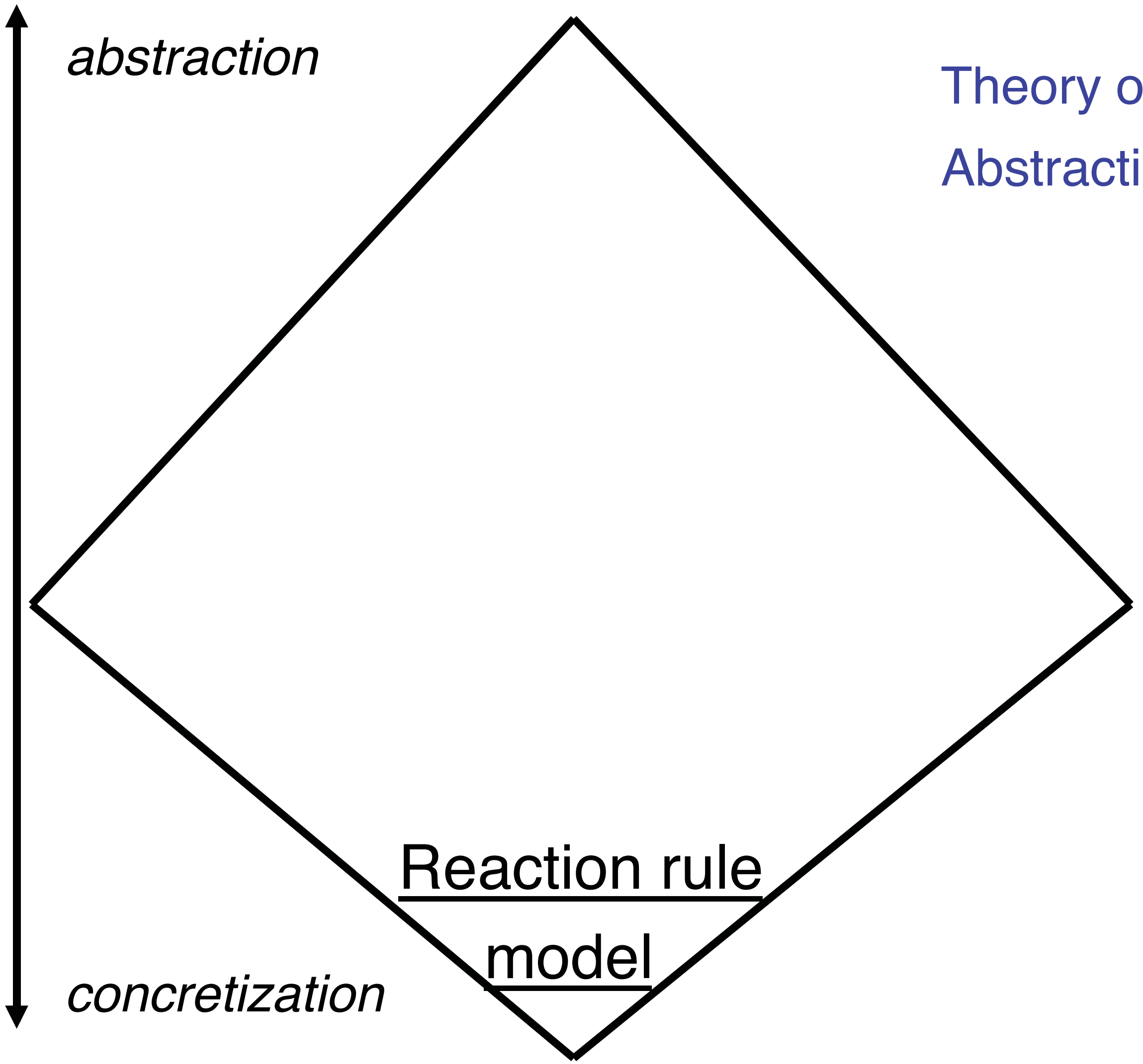
3. Discrete Semantics: numbers of molecules
   - Multiset rewriting, Petri net
   - CHAM [Berry Boudol 90] [Banatre Le Metayer 86]

A , B → C++ A- - B- -

4. Boolean Semantics: presence-absence of molecules
   - Asynchronous Transition System

$A \wedge B \rightarrow C \wedge A/\neg A \wedge B/\neg B$

Hierarchy of Semantics

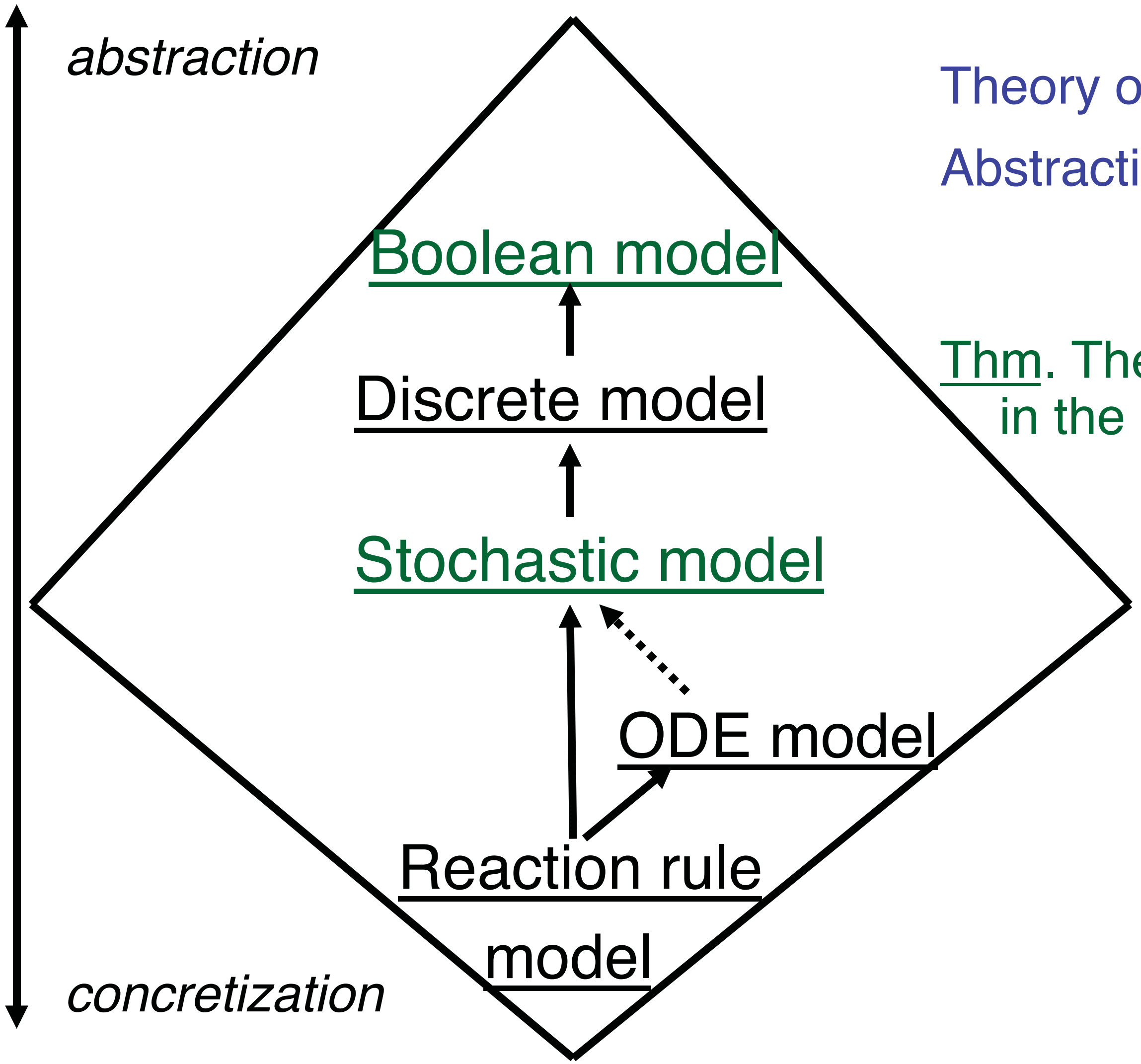


Theory of abstract Interpretation

Abstractions as Galois connections

[Cousot Cousot POPL'77]

Hierarchy of Semantics



Theory of abstract Interpretation

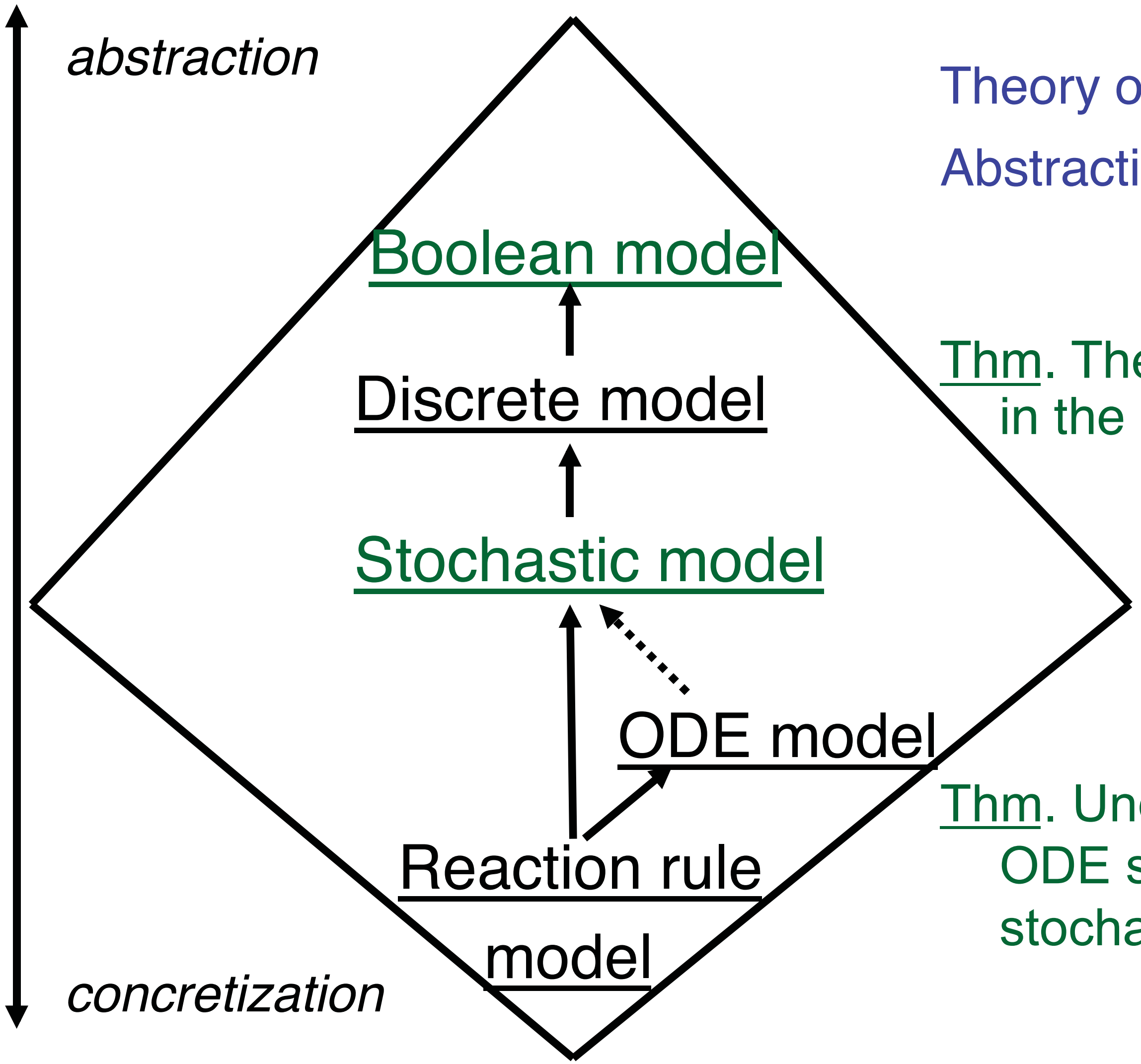
Abstractions as Galois connections

[Cousot Cousot POPL'77]

Thm. The stochastic behaviors are visible in the boolean semantics

[Fages Soliman CMSB'06,TCS'08]

Hierarchy of Semantics



Theory of abstract Interpretation

Abstractions as Galois connections

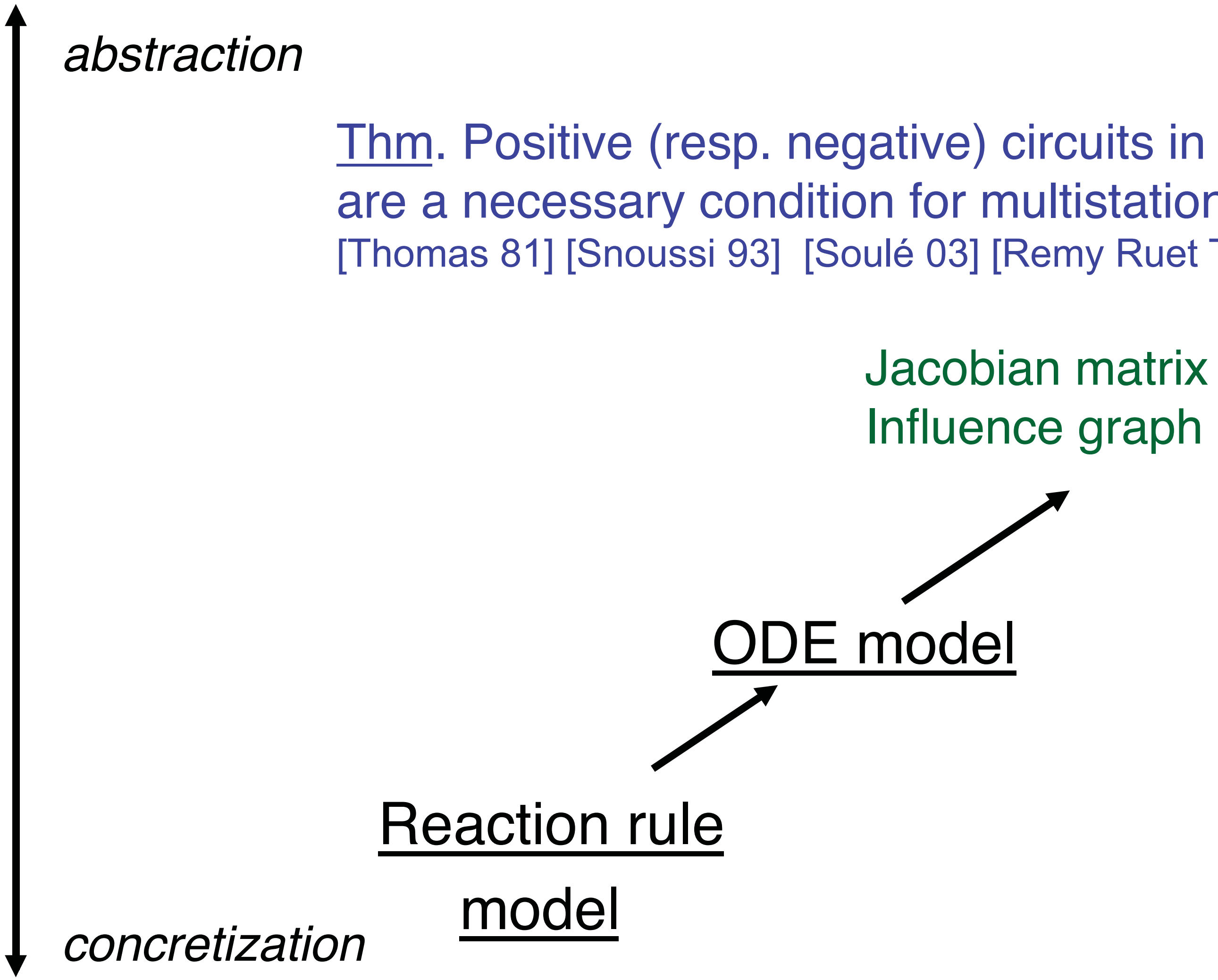
[Cousot Cousot POPL'77]

Thm. The stochastic behaviors are visible in the boolean semantics

[Fages Soliman CMSB'06,TCS'08]

Thm. Under appropriate conditions the ODE semantics approximates the mean stochastic behavior [Gillespie 71]

Influence Graph Abstraction



abstraction

Thm. Positive (resp. negative) circuits in the influence graph are a necessary condition for multistationarity (resp. oscillations)
[Thomas 81] [Snoussi 93] [Soulé 03] [Remy Ruet Thieffry 05] [Richard 07]

Jacobian matrix
Influence graph

ODE model

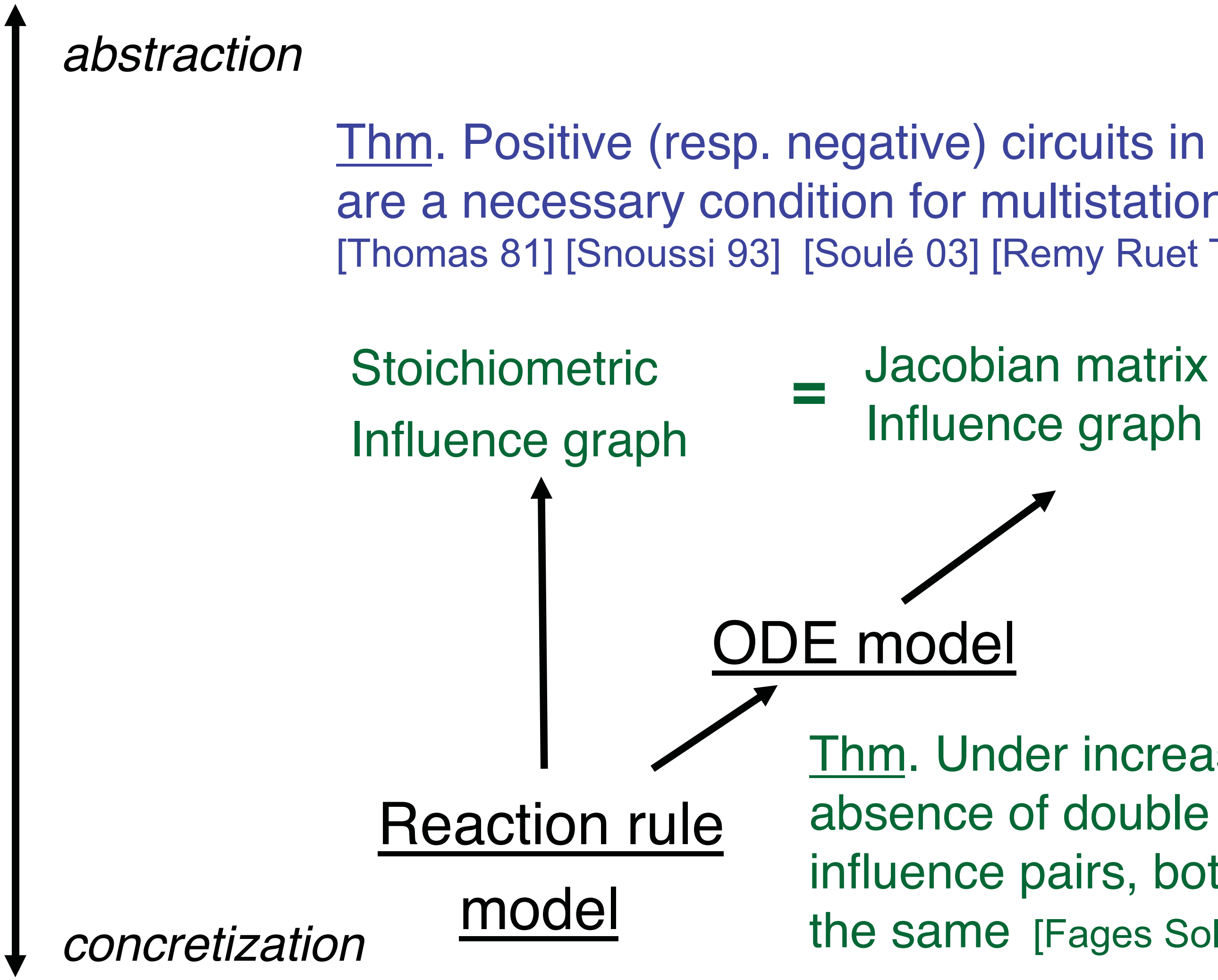
Reaction rule model

concretization

Influence Graph Abstraction

abstraction

Thm. Positive (resp. negative) circuits in the influence graph are a necessary condition for multistationarity (resp. oscillations) [Thomas 81] [Snoussi 93] [Soulé 03] [Remy Ruet Thieffry 05] [Richard 07]



Stoichiometric Influence graph

=

Jacobian matrix Influence graph

ODE model

Reaction rule model

concretization

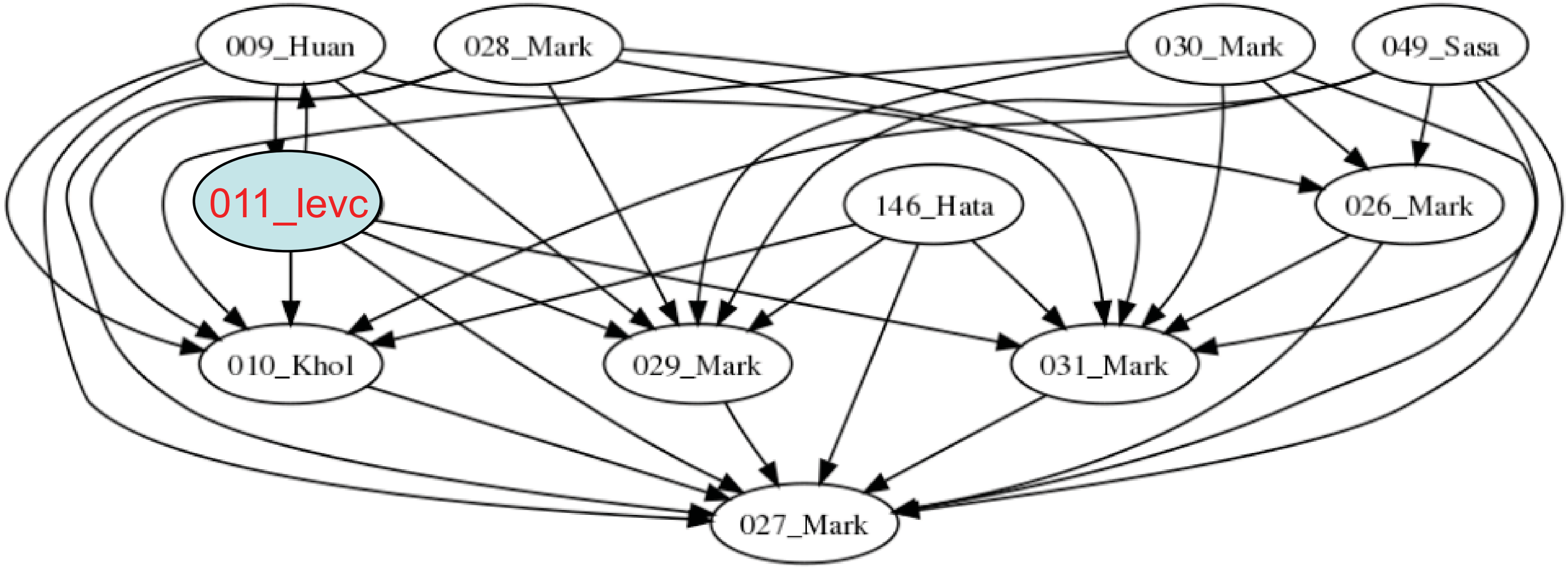
Thm. Under increasing kinetics and in absence of double positive negative influence pairs, both influence graphs are the same [Fages Soliman FMSB 08]

Hierarchies of Models

Model reductions as subgraph epimorphisms (constraint program)

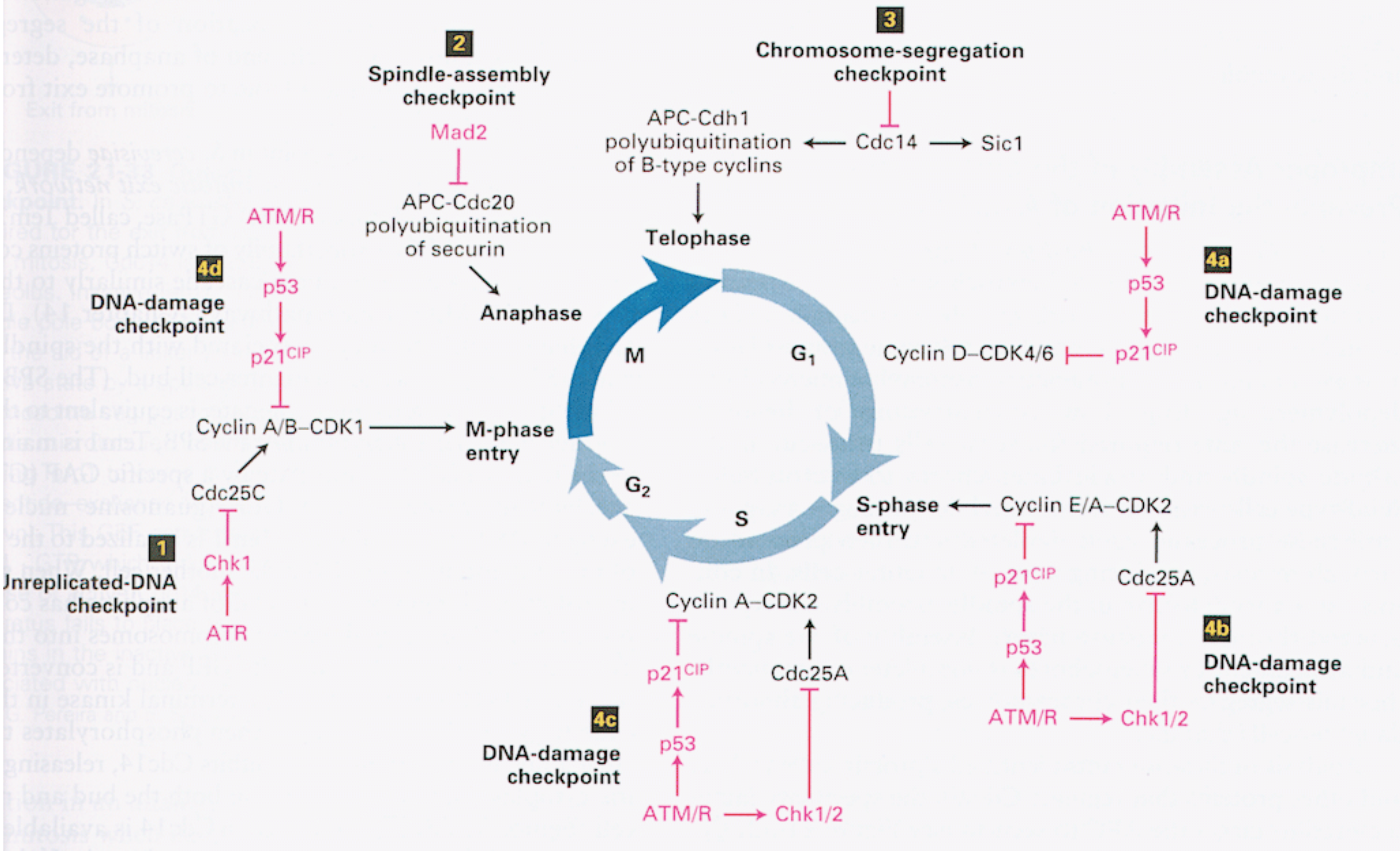
- 4 operations on reaction graphs: delete/merge species/reactions
- Query language for model repositories

[Gay Fages Soliman, A graph matching method for reducing and relating models 2010 biionfo]

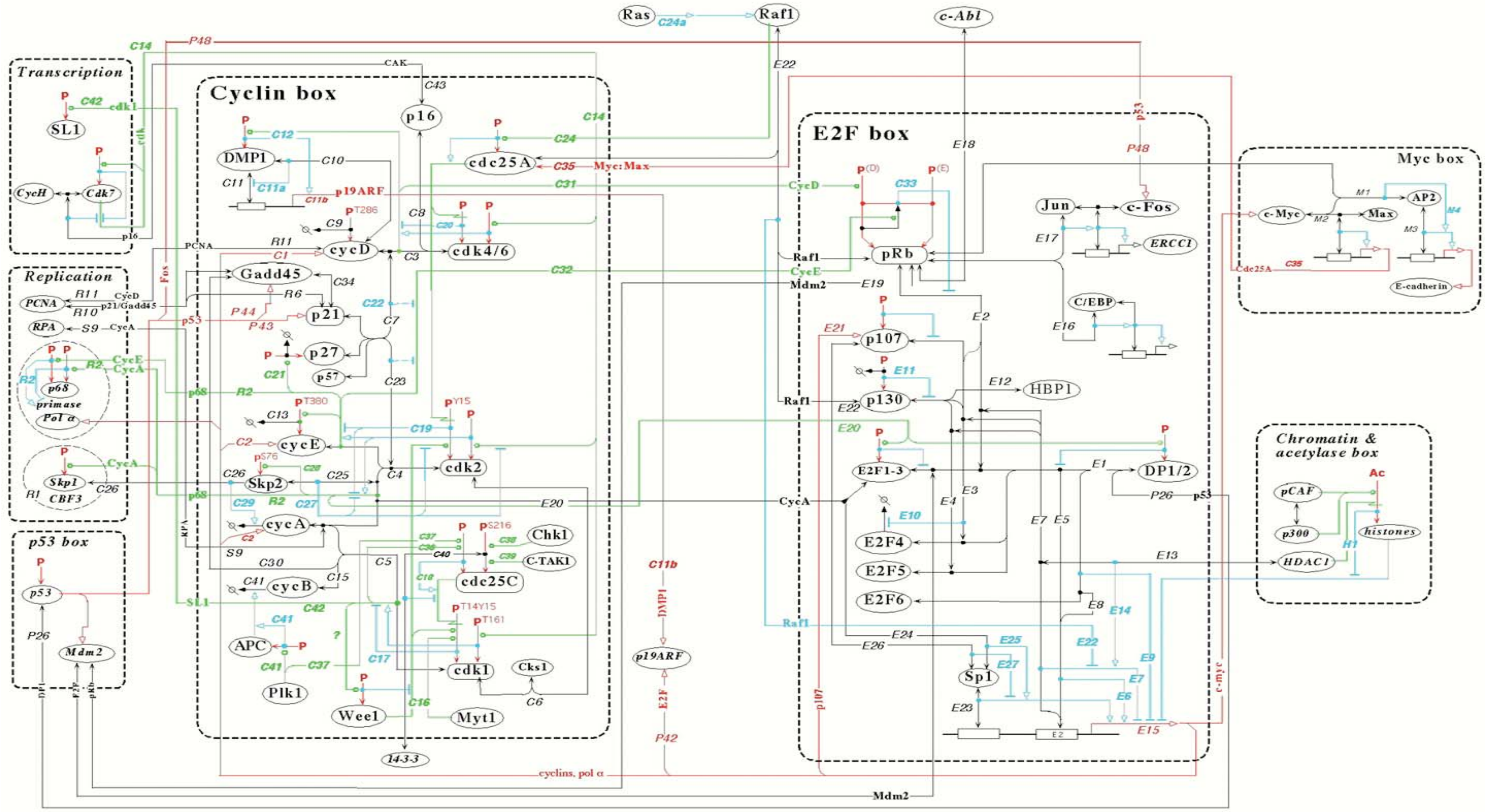


MAPK models from SBML model repository http://www.biomodels.net

Cell Cycle Control

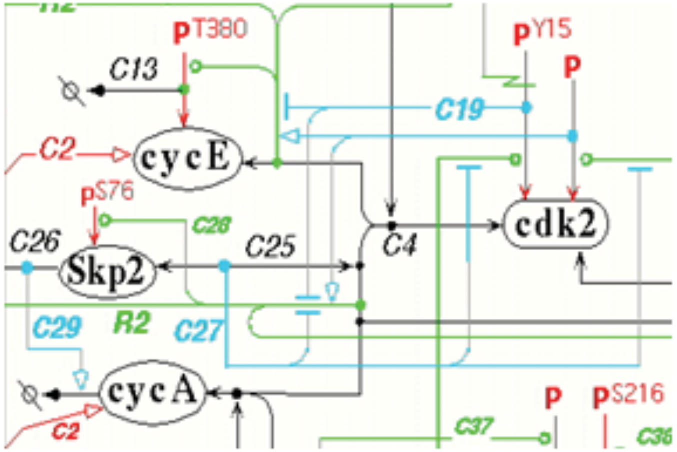


Mammalian Cell Cycle Control Map [Kohn 99]



Kohn’s map detail for Cdk2

Complexations with CycA and CycE



```
cdk2~$P + cycA-$C => cdk2~$P-cycA-$C
    where $C in {_,cks1} .
cdk2~$P + cycE~$Q-$C => cdk2~$P-cycE~$Q-$C
    where $C in {_,cks1} .
p57 + cdk2~$P-cycA-$C => p57-cdk2~$P-cycA-$C
    where $C in {_, cks1}.
cycE-$C =[cdk2~{p2}-cycE-$S]=> cycE~{T380}-$C
    where $S in {_, cks1} and $C in {_, cdk2~?, cdk2~?-cks1}
```

147 rule patterns → 2733 expanded rules [Chabrier Chiaverini Danos Fages Schachter 04]

How to query the possible dynamical properties of such a system ?
reachability, checkpoints, steady states, oscillations ?

Quantitative Model of Cell Cycle Control [Tyson 91]

k1 for _ => Cyclin.

k2*[Cyclin] for Cyclin => _.

k3*[Cyclin]*[Cdc2~{p1}] for
    Cyclin + Cdc2~{p1} => Cdc2~{p1}-Cyclin~{p1}.

k6*[Cdc2-Cyclin~{p1}] for
    Cdc2-Cyclin~{p1} => Cdc2 + Cyclin~{p1}.

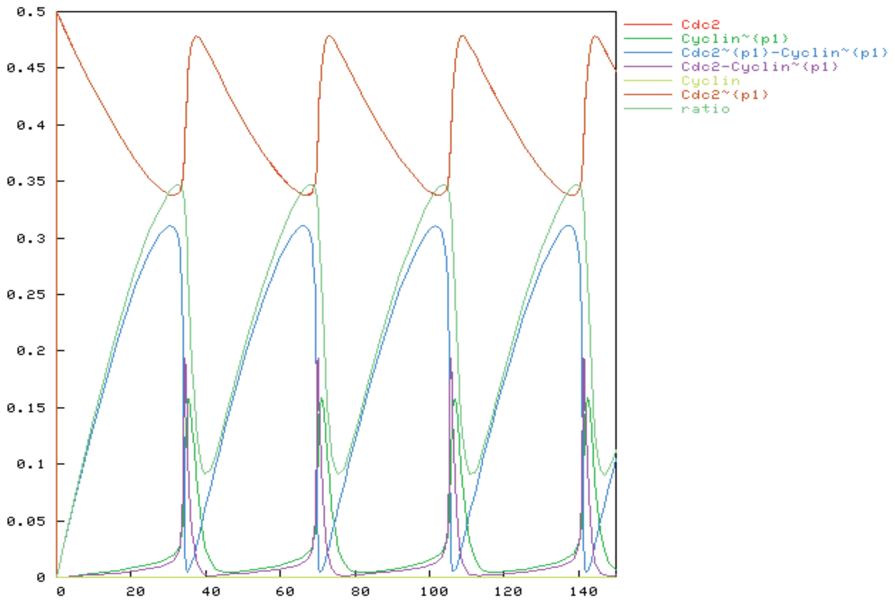
k7*[Cyclin~{p1}] for Cyclin~{p1} => _.

k8*[Cdc2] for Cdc2 => Cdc2~{p1}.

k9*[Cdc2~{p1}] for Cdc2~{p1} => Cdc2.

k4p*[Cdc2~{p1}-Cyclin~{p1}] for
    Cdc2~{p1}-Cyclin~{p1} => Cdc2-Cyclin~{p1}.

k4*[Cdc2-Cyclin~{p1}]^2*[Cdc2~{p1}-Cyclin~{p1}] for
    Cdc2~{p1}-Cyclin~{p1} =[Cdc2-Cyclin~{p1}]=> Cdc2-Cyclin~{p1}.



Overview of the Talk

1. *Rule-based Modeling* of Biochemical Systems
    1. Syntax of molecules, compartments and reactions
    2. Hierarchy of semantics: boolean, differential, stochastic
    3. Cell cycle control models
2. *Temporal Logic Language* for formalizing Biological Properties
    1. CTL for the boolean semantics
    2. LTL(R) for the differential semantics
    3. Parameter inference from LTL(R) properties
    4. Robustness w.r.t. LTL(R) properties
3. *Conclusion* and collaborations

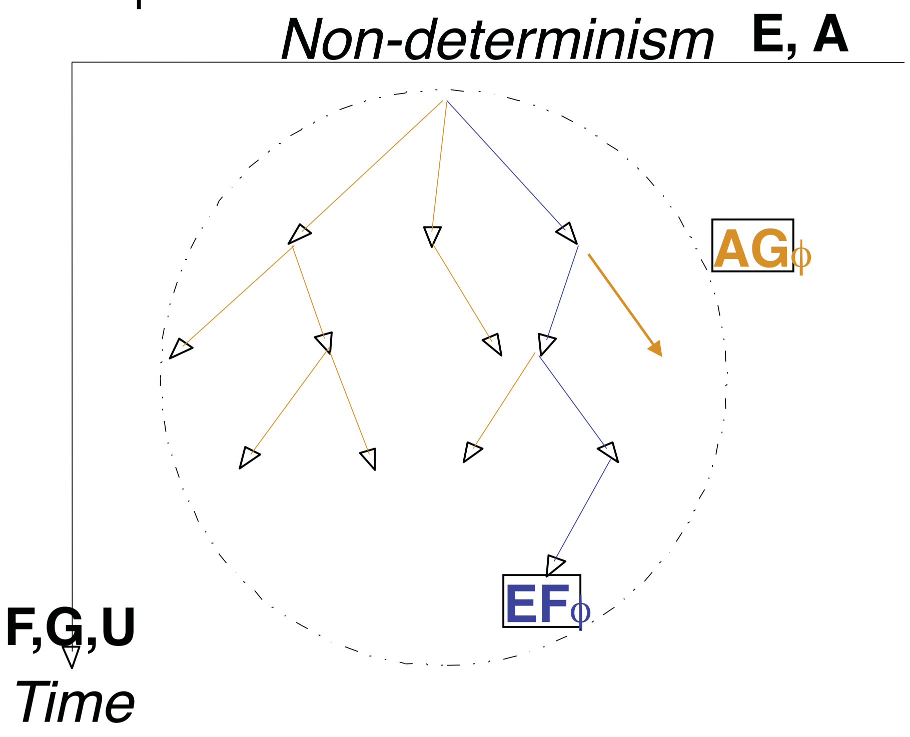
- F. Fages, S. Soliman. Formal Cell Biology in Biocham. *Springer LNBI* 5016, 2008
- A. Rizk, G. Batt, F. Fages, S. Soliman. *Bioinformatics* 25:169-178, 2009
- S. Gay, S. Soliman, F. Fages. *Bioinfomatics* 26:575-581, 2010

Temporal Logic Queries

Temporal logics introduced for program verification by [Pnueli 77]

Computation Tree Logic CTL [Emerson Clarke 80]

Non-det. Time	**E** exists	**A** always
X next time	**EX**(φ)	**AX**(φ)
F finally	**EF**(φ) *¬**AG**(¬ φ)*	**AF**(φ) *liveness*
G globally	**EG**(φ) *¬ **AF**(¬ φ)*	**AG**() *safety*
U until	**E** (φ1 **U** φ2)	**A** (φ1 **U** φ2)



Cell Cycle Model-Checking (with NuSMV Binary Decision Diagrams)

```
biocham: check(reachable(cdk46~{p1,p2}-cycD~{p1})).
  Ei(EF(cdk46~{p1,p2}-cycD~{p1})) is true
biocham: check(checkpoint(cdc25C~{p1,p2}, cdk1~{p1,p3}-cycB)).
  Ai(!(E(!(cdc25C~{p1,p2}) U cdk1~{p1,p3}-cycB))) is true
biocham: nusmv(Ai(AG(!(cdk1~{p1,p2,p3}-cycB) -> checkpoint(Wee1, cdk1~{p1,p2,p3}-cycB))))).
  Ai(AG(!(cdk1~{p1,p2,p3}-cycB)->!(E(!(Wee1) U cdk1~{p1,p2,p3}-cycB)))) is false
biocham: why.
-- Loop starts here
   cycB-cdk1~{p1,p2,p3} is present
   cdk7 is present
   cycH is present
   cdk1 is present
   Myt1 is present
   cdc25C~{p1} is present
rule_114 cycB-cdk1~{p1,p2,p3}=[cdc25C~{p1}]=>cycB-cdk1~{p2,p3}.
   cycB-cdk1~{p2,p3} is present
   cycB-cdk1~{p1,p2,p3} is absent
rule_74 cycB-cdk1~{p2,p3}=[Myt1]=>cycB-cdk1~{p1,p2,p3}.
   cycB-cdk1~{p2,p3} is absent
   cycB-cdk1~{p1,p2,p3} is present
```

Kohn’s Map Model-Checking

147-2733 rules, 165 proteins and genes, 500 variables, 2^{500} states.

Biocham NuSMV model-checker time in seconds [Chabrier Fages CMSB 03]

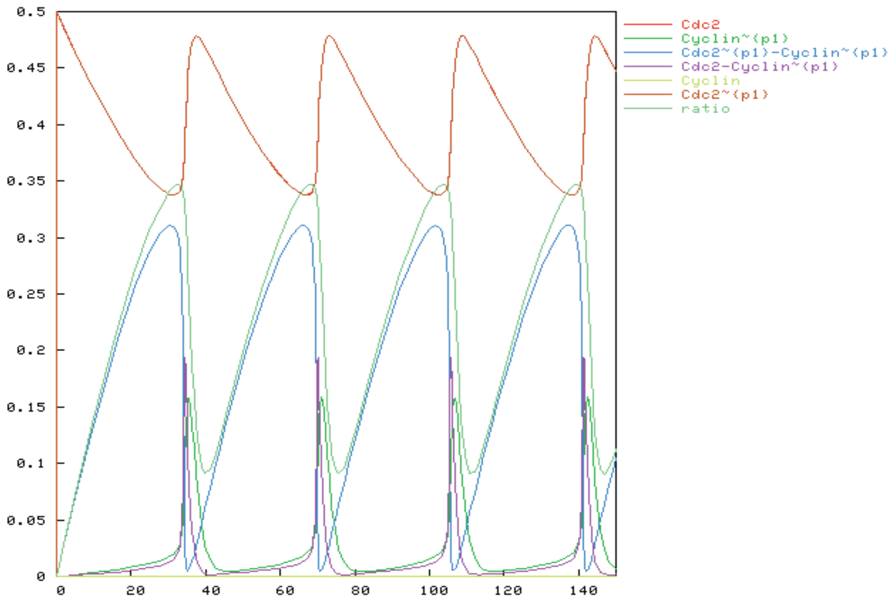
Initial state G2	Query:	Time:
	compiling	29
Reachability G1	**EF** CycE	2
Reachability G1	**EF** CycD	1.9
Reachability G1	**EF** PCNA-CycD	1.7
Checkpoint for mitosis complex	¬ **EF (**¬ Cdc25~{Nterm} **U** Cdk1~{Thr161}-CycB**)**	2.2
Oscillations CycA	**EG ((EF** ¬ CycA**)** ∧ **(EF** CycA**))**	31.8
Osciallations CycB	**EG ((EF** ¬ CycB**)** ∧ **(EF** CycB**))** false !	6

Constraint Linear Time Logic LTL(R)

- **F**([A]>10) : the concentration of A eventually gets above 10.
- **FG**([A]>10) : the concentration of A eventually reaches and remains above value 10.
- **F**(Time=t1∧[A]=v1 ∧ **F**(.... ∧ **F**(Time=tN∧[A]=vN)...))
  Numerical data time series (e.g. experimental curves)
- **G**([A]+[B]<[C]) : the concentration of C is always greater than the sum of the concentrations of A and B.
- $\mathbf{F}([M] < V \wedge \mathbf{X}([M] = V) \wedge \mathbf{X}([M] < V)))$: local maximum V
- oscillations, period constraints, etc.

Quantitative Model of Cell Cycle Control [Tyson 91]

```
k1 for  _ => Cyclin.
k2*[Cyclin] for  Cyclin => _.
k3*[Cyclin]*[Cdc2~{p1}] for
     Cyclin + Cdc2~{p1} => Cdc2~{p1}-Cyclin~{p1}.
k6*[Cdc2-Cyclin~{p1}] for
     Cdc2-Cyclin~{p1} => Cdc2 + Cyclin~{p1}.
k7*[Cyclin~{p1}] for Cyclin~{p1} => _.
k8*[Cdc2] for  Cdc2 => Cdc2~{p1}.
k9*[Cdc2~{p1}] for Cdc2~{p1} => Cdc2.
k4p*[Cdc2~{p1}-Cyclin~{p1}] for
     Cdc2~{p1}-Cyclin~{p1} => Cdc2-Cyclin~{p1}.
k4*[Cdc2-Cyclin~{p1}]^2*[Cdc2~{p1}-Cyclin~{p1}] for
      Cdc2~{p1}-Cyclin~{p1}  =[Cdc2-Cyclin~{p1}]=>  Cdc2-Cyclin~{p1}.
```



Numerical Integration of ODE Models

$dX/dt = f(X)$.

Initial conditions X_0

Idea: discretize time t_0, $t_1=t_0+\Delta t_0$, $t_2=t_1+\Delta t_1$, …
and compute a numerical trace

(t_0,X_0), (t_1,X_1), …, (t_n,X_n)…

- *Euler's method*: $t_{i+1}=t_i+\Delta t$ $X_{i+1}=X_i+f(X_i)*\Delta t$
  error estimation $E(X_{i+1})=|f(X_i)-f(X_{i+1})|*\Delta t$

- *Runge-Kutta's method*: intermediate computations at $\Delta t/2$
- *Adaptive step method*: $\Delta t_{i+1}= \Delta t_i/2$ while E>Emax, otherwise $\Delta t_{i+1}= 2*\Delta t_i$

- *Rosenbrock's stiff method*: solve $X_{i+1}=X_i+f(X_{i+1})*\Delta t$ by formal differentiation

Parameter Search from Temporal Properties

```
biocham: learn_parameter([k3,k4],[(0,200),(0,200)],20,
                          oscil(Cdc2-Cyclin~{p1},3),150).
```

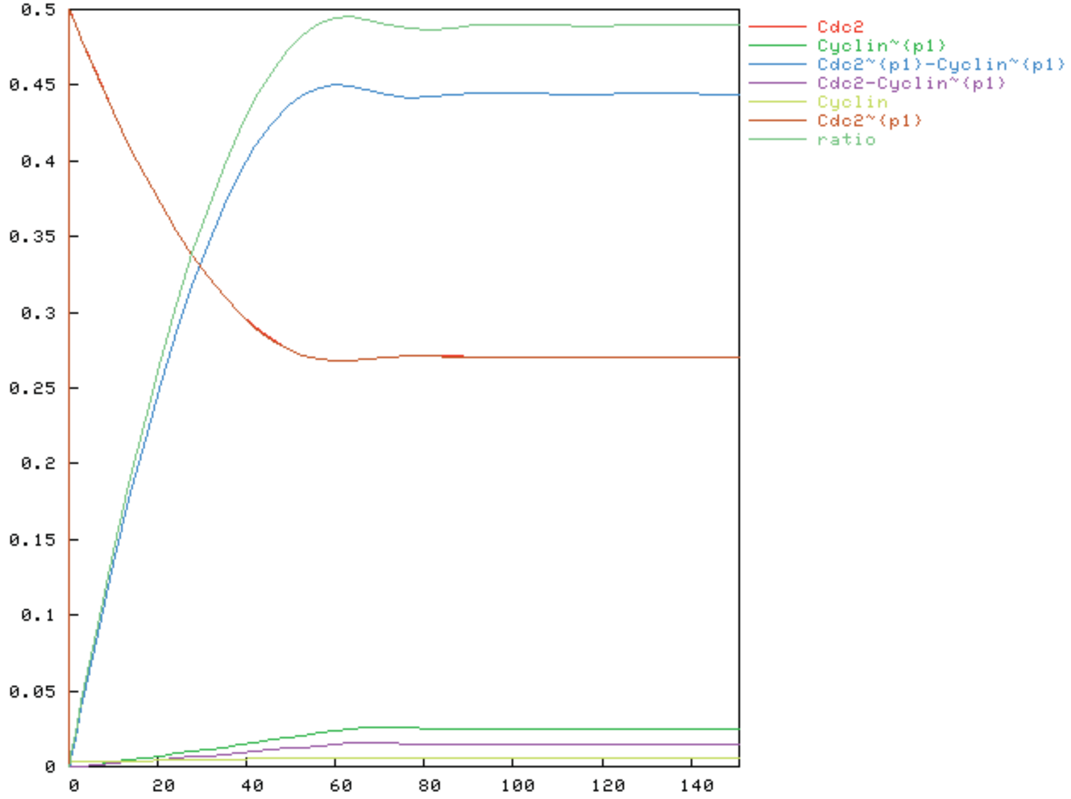
Parameter Search from Temporal Properties

biocham: learn_parameter([k3,k4],[(0,200),(0,200)],20,
oscil(Cdc2-Cyclin~{p1},3),150).

First values found :

parameter(k3,10).

parameter(k4,70).

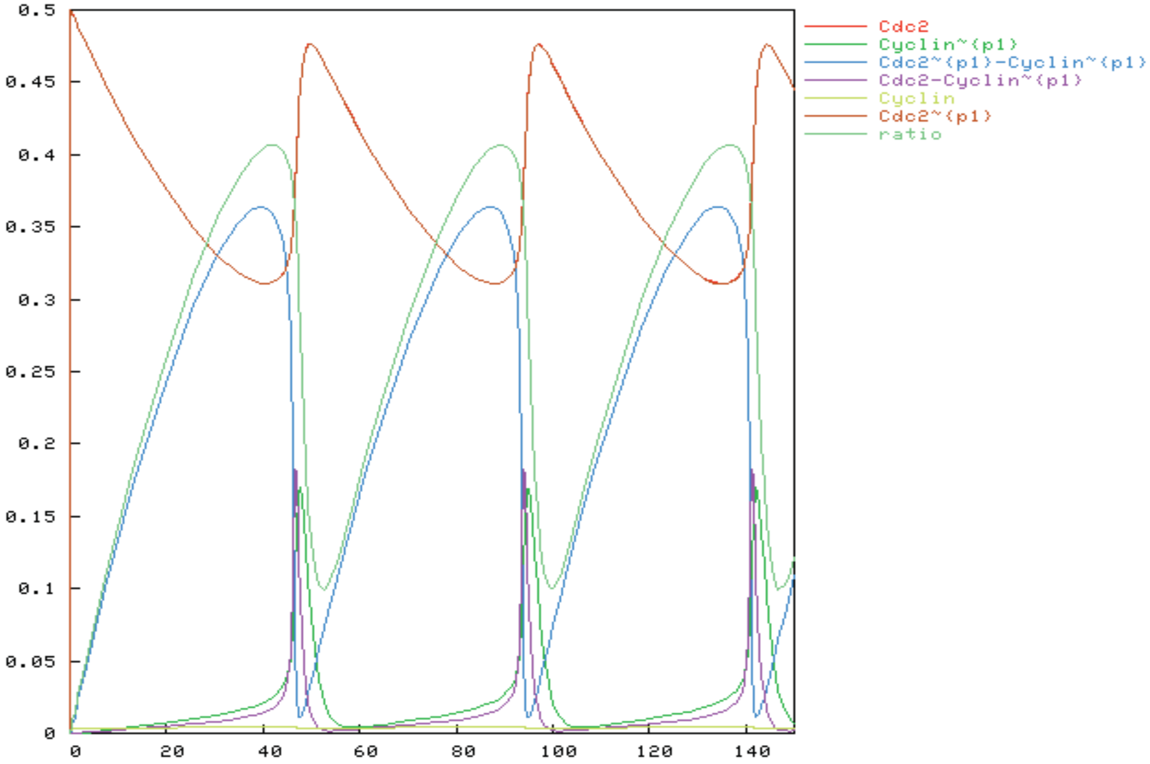


Parameter Search from Temporal Properties

```
biocham: learn_parameter([k3,k4],[(0,200),(0,200)],20,
            oscil(Cdc2-Cyclin~{p1},3) & F([Cdc2-Cyclin~{p1}]>0.15), 150).
```

First values found :

```
parameter(k3,10).
parameter(k4,120).
```



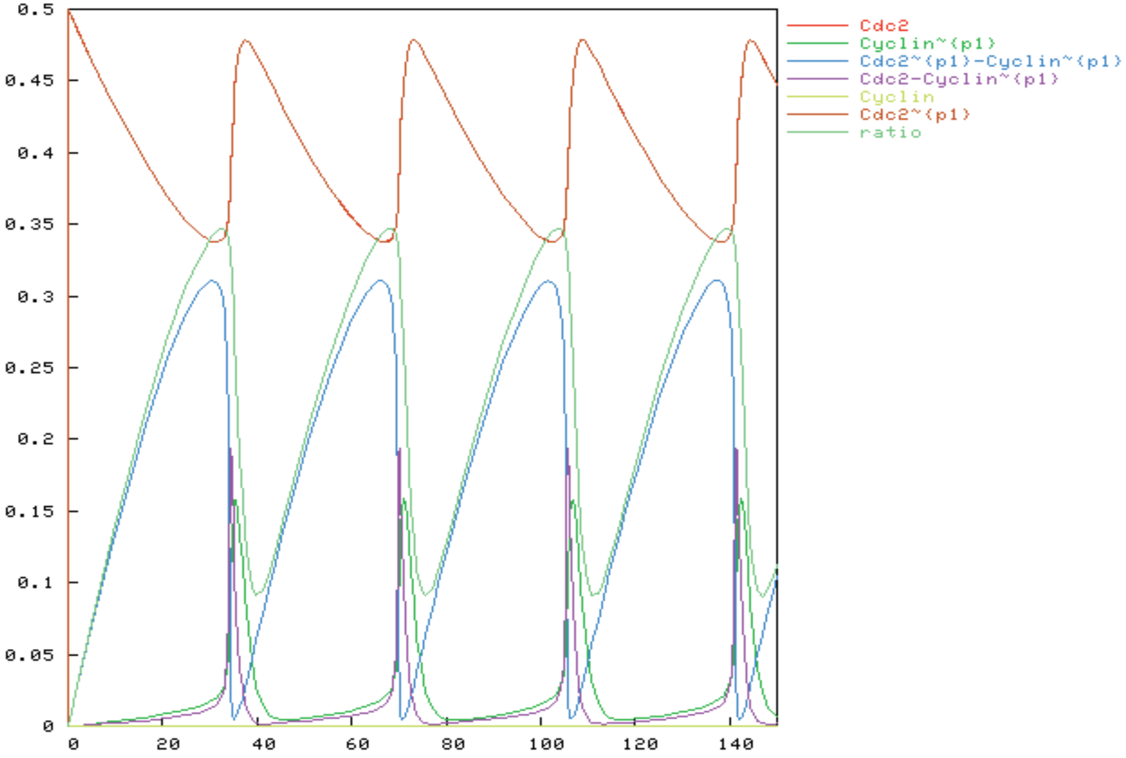
Parameter Search from Temporal Properties

biocham: learn_parameter([k3,k4],[(0,200),(0,200)],20,
period(Cdc2-Cyclin~{p1},35), 150).

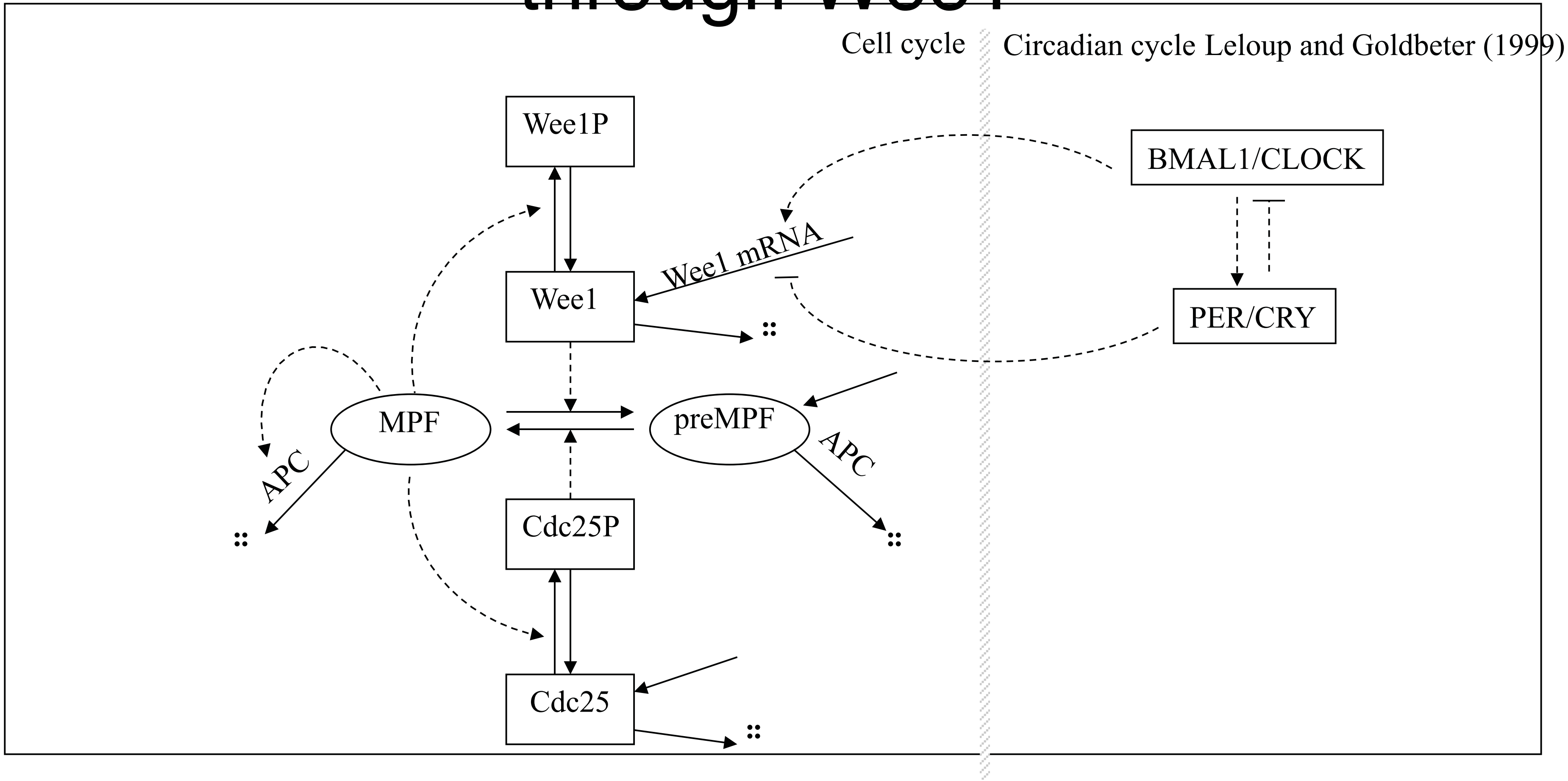
First values found:

parameter(k3,10).

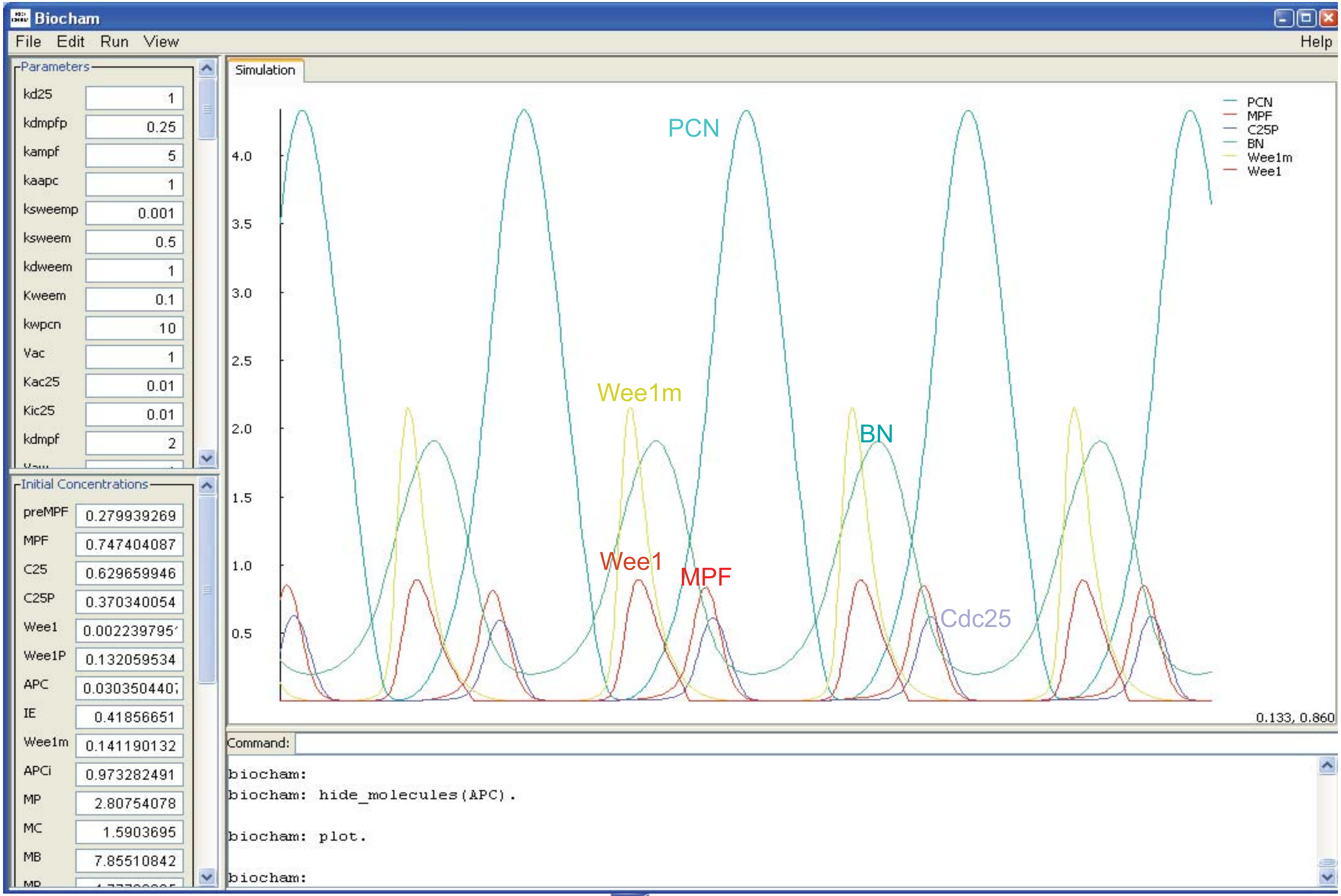
parameter(k4,280).



Linking the Cell and Circadian Cycles through Wee1



[L. Calzone, S. Soliman 2006]



Parameters

kd25	1
kdmpfp	0.25
kampf	5
kaapc	1
ksweemp	0.001
ksweem	0.5
kdweem	1
Kweem	0.1
kwpcn	10
Vac	1
Kac25	0.01
Kic25	0.01
kdmpf	2

Initial Concentrations

preMPF	0.279939269
MPF	0.747404087
C25	0.629659946
C25P	0.370340054
Wee1	0.002239795
Wee1P	0.132059534
APC	0.030350440
IE	0.41856651
Wee1m	0.141190132
APCi	0.973282491
MP	2.80754078
MC	1.5903695
MB	7.85510842

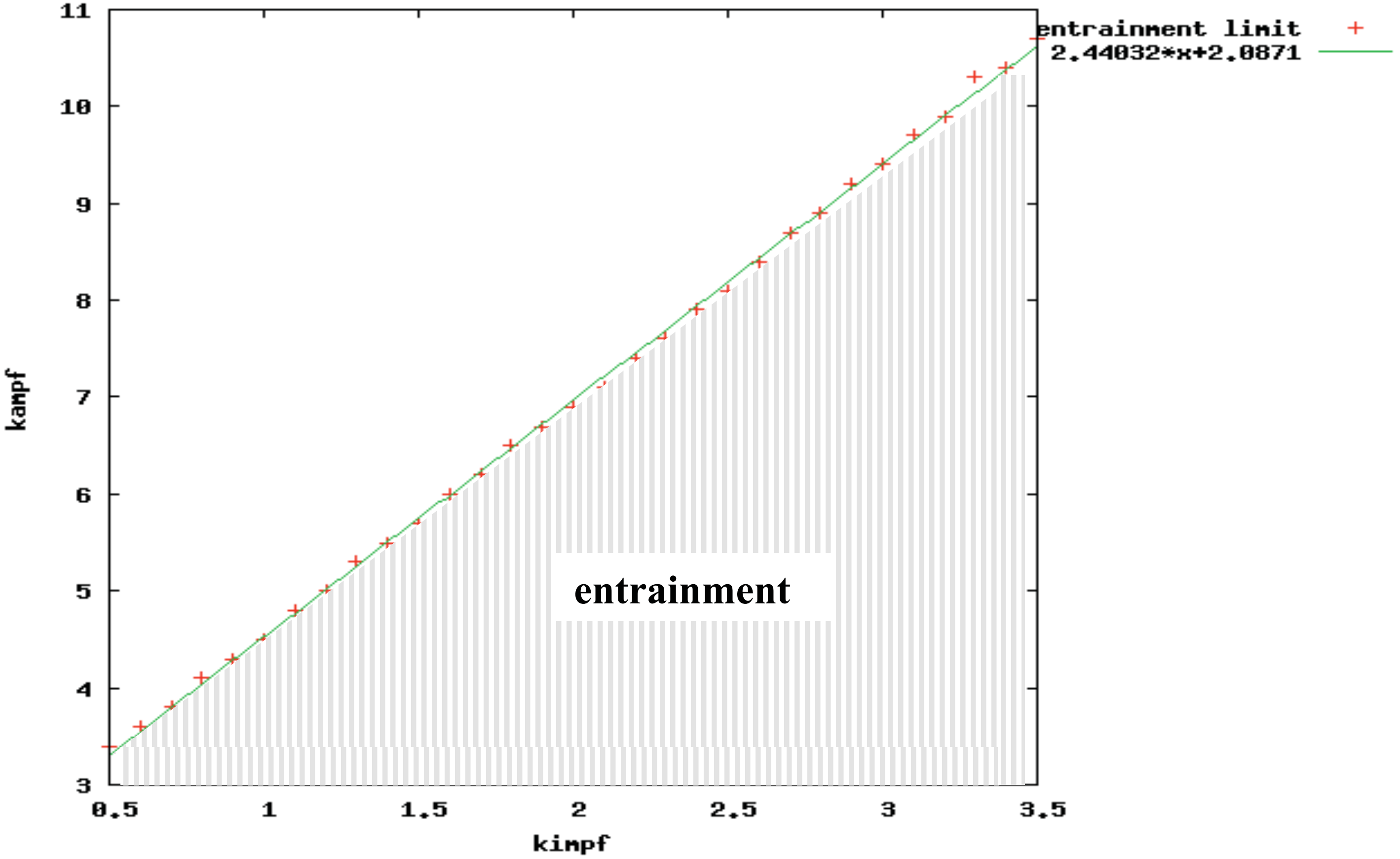
Command:

```
biocham:
biocham: hide_molecules(APC).

biocham: plot.

biocham:
```

Condition on Wee1/Cdc25 for the Entrainment in Period



Entrainment in period constraint expressed in LTL with the period formula

True/False valuation of temporal logic formulae

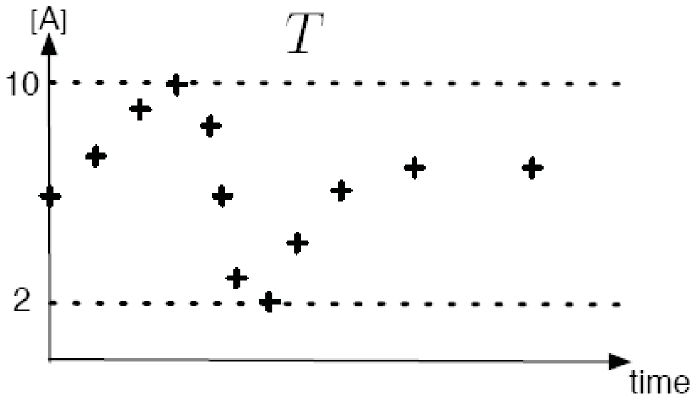
The **True/False** valuation of temporal logic formulae is **not well adapted** to several problems :

- parameter search, optimization and control of continuous models
- quantitative estimation of robustness
- sensitivity analyses

→ need for a continuous degree of satisfaction of temporal logic formulae

How far is the system from verifying the specification ?

Model-Checking Generalized to Constraint Solving



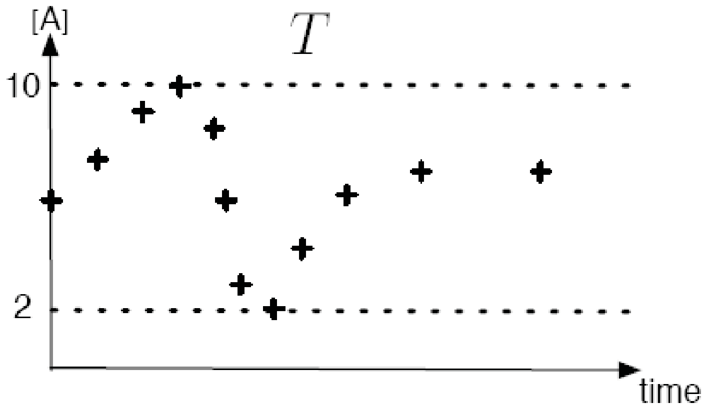
$LTL(\mathbb{R})$

$\Phi = F([A] \geq 7$
$\wedge F([A] \leq 0))$

Model-checking

the formula is false

Model-Checking Generalized to Constraint Solving



$LTL(\mathbb{R})$

$\Phi = F([A] \geq 7$
$\wedge F([A] \leq 0))$

Model-checking

the formula is false

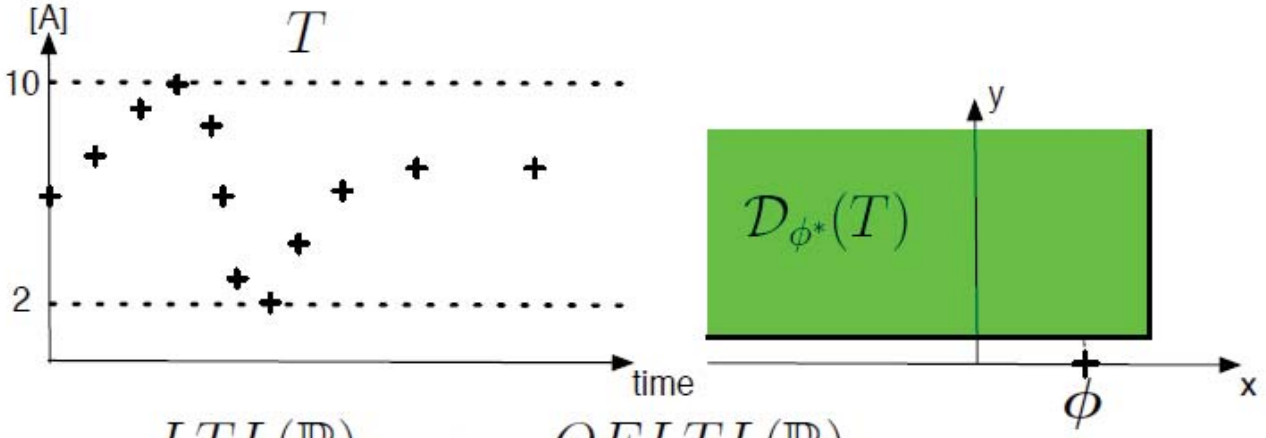
$QFLTL(\mathbb{R})$

$\Phi^* = F([A] \geq x$
$\wedge F([A] \leq y))$

Constraint solving

the formula is true for any
$x \leq 10 \wedge y \geq 2$

Model-Checking Generalized to Constraint Solving



$LTL(\mathbb{R})$

Φ=F([A]≥7
∧F([A]≤0))

Model-checking

the formula is false vd=2 **sd=1/3**

$QFLTL(\mathbb{R})$

Φ*=F([A]≥x
∧F([A]≤y))

Constraint solving

the formula is true for any
x≤10 ∧ y≥2

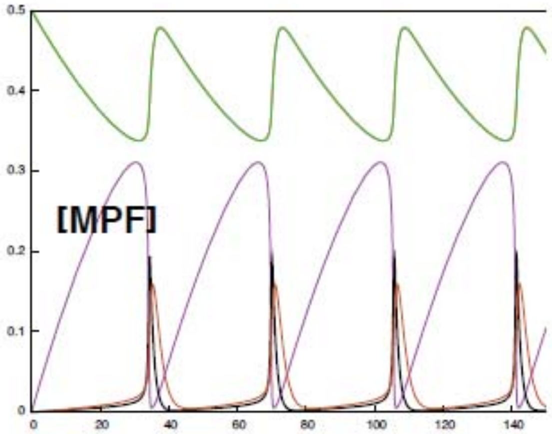
Validity domain $\mathcal{D}_{\phi^*}(T)$ for the **free variables** in ϕ^* [Fages Rizk CMSB'07]

Violation degree $vd(T,\phi) = \text{distance}(val(\phi), D_{\phi^*}(T))$

Satisfaction degree $sd(T,\phi) = \frac{1}{1+vd(T,\phi)} \in [0,1]$

Learning kinetic parameter values from LTL specifications

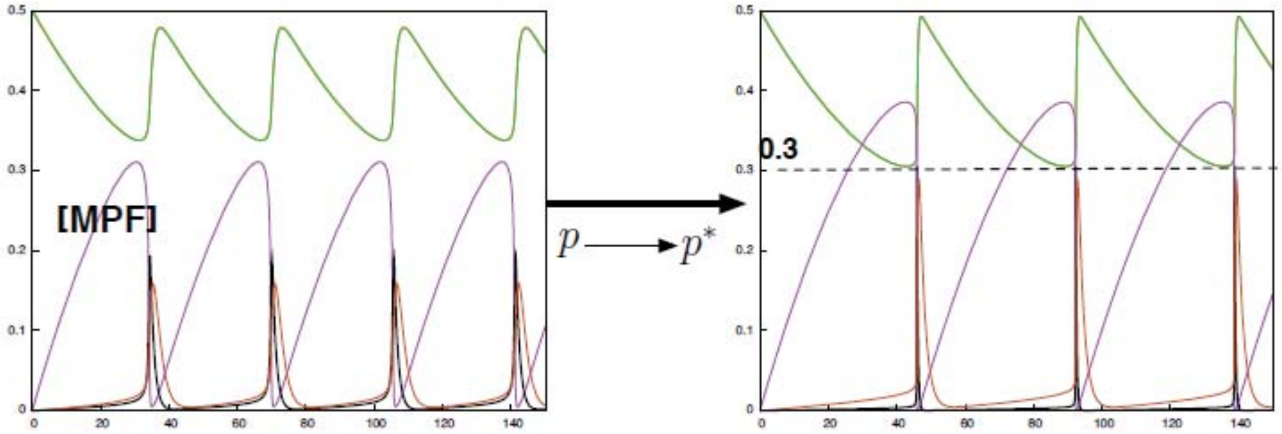
- simple model of the yeast cell cycle from [Tyson PNAS 91]
- models Cdc2 and Cyclin interactions (6 variables, 8 kinetic parameters)



- Pb : find values of 8 parameters such that amplitude is ≥ 0.3
  ϕ^*: **F**([A]>x ∧ **F**([A]<y))
  amplitude z=x-y
  goal : z = 0.3

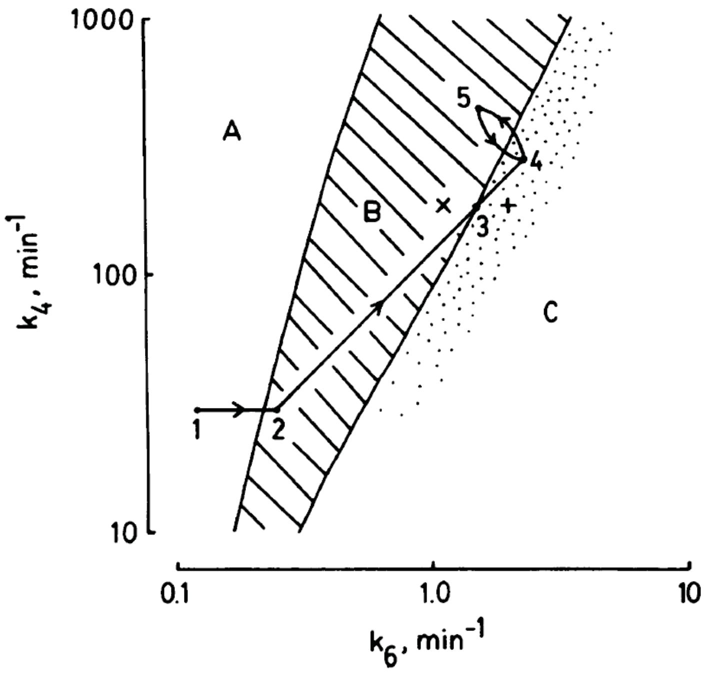
Learning kinetic parameter values from LTL specifications

- simple model of the yeast cell cycle from [Tyson PNAS 91]
- models Cdc2 and Cyclin interactions (6 variables, 8 kinetic parameters)



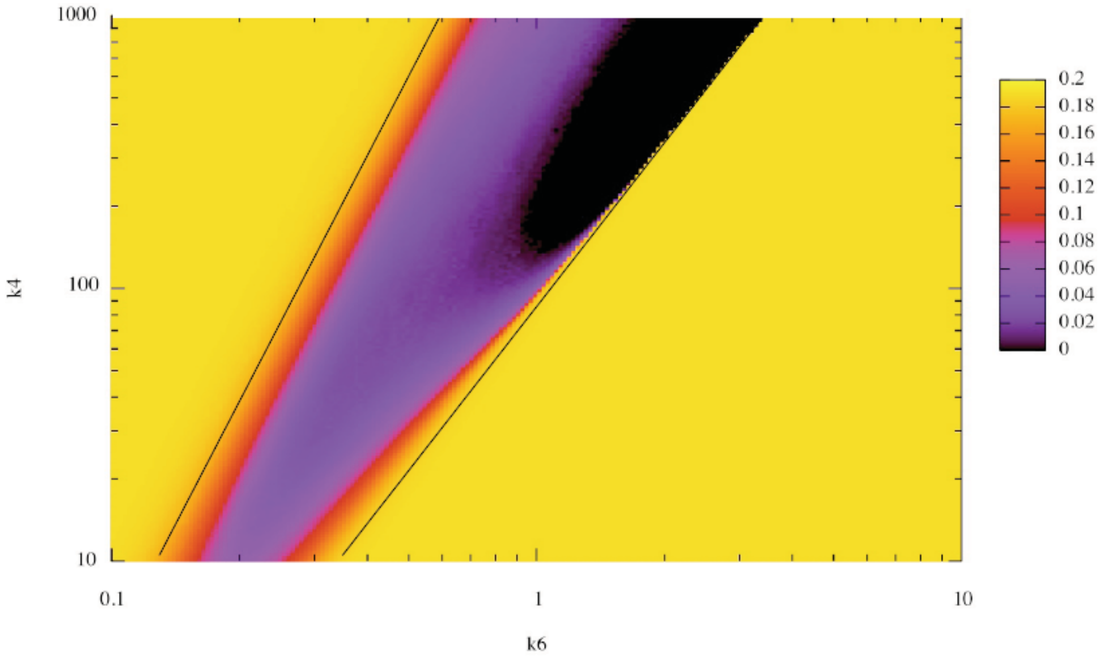
- Pb : find values of 8 parameters such that amplitude is ≥ 0.3
  ϕ^*: $\mathbf{F}([A]>x \wedge \mathbf{F}([A]<y))$
  amplitude z=x-y
  goal : z = 0.3
- → solution found after 30s (100 calls to the fitness function)

LTL(R) Satisfaction Degree and Bifurcation Diagram



Bifurcation diagram on k4, k6

[Tyson 91]

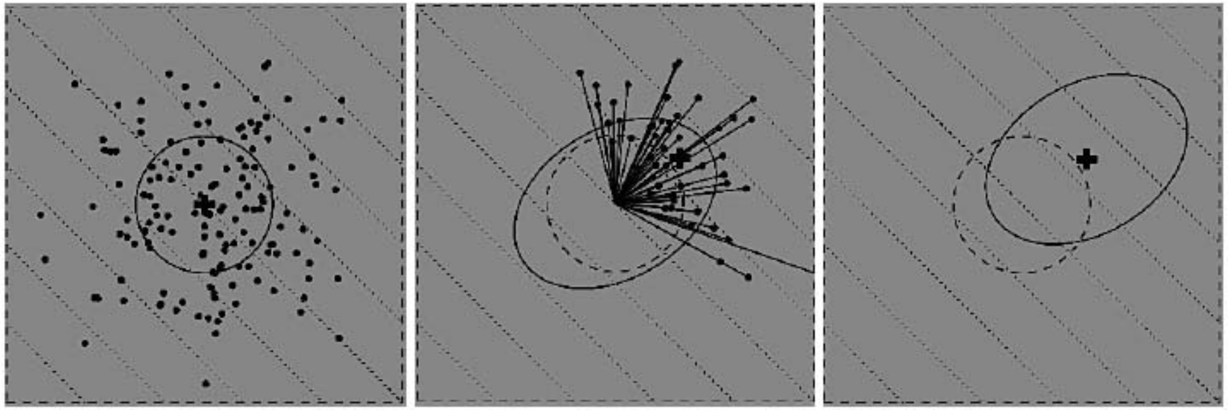


Continuous satisfaction degree in [0,1]
of an LTL(R) formula for oscillation
with amplitude constraint

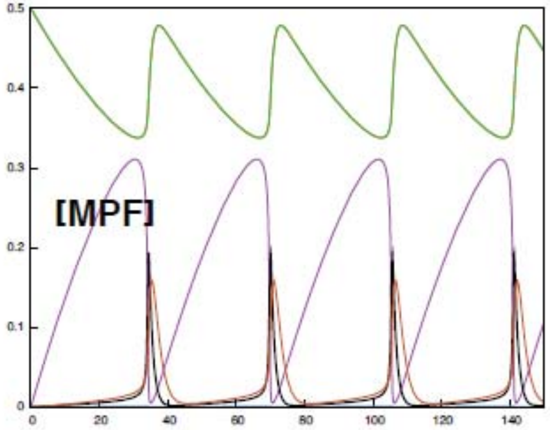
[Rizk Batt Fages Soliman CMSB 08]

Black-box Randomized Non-linear Optimization Method

- Use existing non-linear optimization toolbox for kinetic parameter search using satisfaction degree as fitness function
- We use the state-of-the-art Covariance Matrix Adaptation Evolution Strategy (CMA-ES) [Hansen Osermeier 01, Hansen 08]
- CMA-ES maximizes an objective function in continuous domain in a black box scenario
- CMA-ES uses a probabilistic neighborhood and updates information in covariance matrix at each move



Learning Parameter Values from Period Constraints in LTL



- Pb : find values of 8 parameters such that period is 20

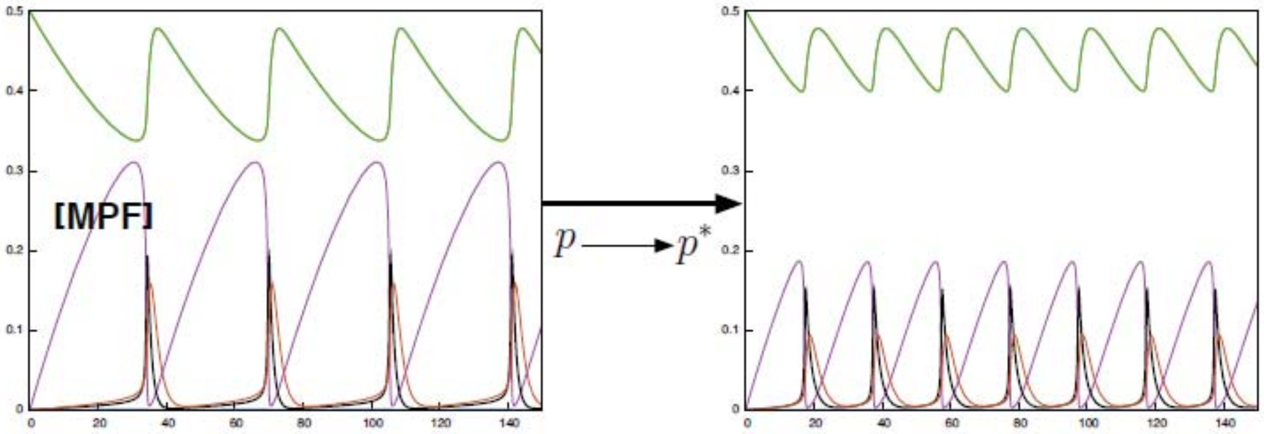
  ϕ^*:**F**(MPF$_{localmaximum}$ $\wedge$Time=t1$\wedge$ **F**(MPF$_{localmaximum}$ $\wedge$Time=t2))

  (with MPF$_{localmaximum}$: d([MPF])/dt>0 $\wedge$ **X**(d([MPF])/dt<0))

  period z=t2-t1

  goal z=20

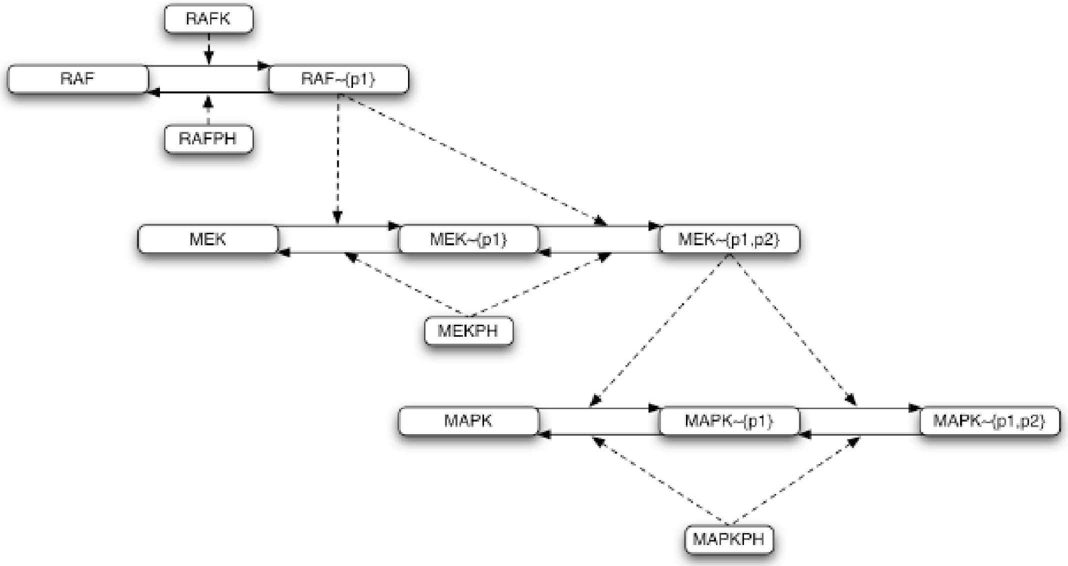
Learning Parameter Values from Period Constraints in LTL



- Pb : find values of 8 parameters such that period is 20
  ϕ^*:$\mathbf{F}(\mathrm{MPF}_{localmaximum} \wedge \mathrm{Time}=\mathrm{t1} \wedge\ \mathbf{F}(\mathrm{MPF}_{localmaximum} \wedge \mathrm{Time}=\mathrm{t2})\ )$
  (with $\mathrm{MPF}_{localmaximum}$: $d([\mathrm{MPF}])/dt>0 \wedge \mathbf{X}(d([\mathrm{MPF}])/dt<0)$)
  period z=t2-t1
  goal z=20
- → Solution found after 60s (200 calls to the fitness function)

Oscillations in MAPK signal transduction cascade

- **MAPK** signaling model [Huang Ferrel PNAS 93]



- **search for oscillations** in **37 dimensions** (30 parameters and 7 initial conditions)
  → solution found after 3 min (200 calls to the fitness function)
  Oscillations already observed by simulation [Qiao et al. 07]
- No negative feedback in the **reaction graph**, but negative circuits in the **influence graph** [Fages Soliman FMSB'08, CMSB'06]

Robustness Measure Definition

Robustness defined with respect to :

- a biological system
- a functionality property D_a
- a set P of perturbations

- General notion of robustness proposed in [Kitano MSB 07]:

$$\mathcal{R}_{a,P} = \int_{p \in P} D_a(p)\ prob(p)\ dp$$

Robustness Measure Definition

Robustness defined with respect to :

- a biological system
- a functionality property D_a
- a set P of perturbations

- General notion of robustness proposed in [Kitano MSB 07]:

$$\mathcal{R}_{a,P} = \int_{p \in P} D_a(p)\ prob(p)\ dp$$

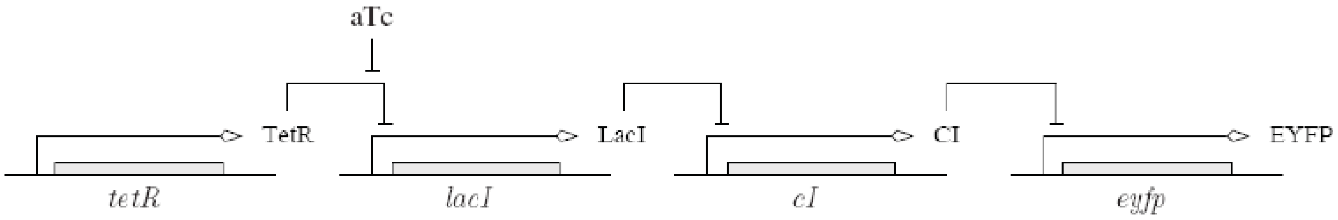
- Our computational measure of robustness w.r.t. LTL($\mathbb{R}$) spec: Given an ODE model with initial conditions, a TL formula ϕ and a set of perturbations P (on initial conditions or parameters),

$$\mathcal{R}_{\phi,P} = \int_{p \in P} sd(T(p), \phi)\ prob(p)\ dp$$

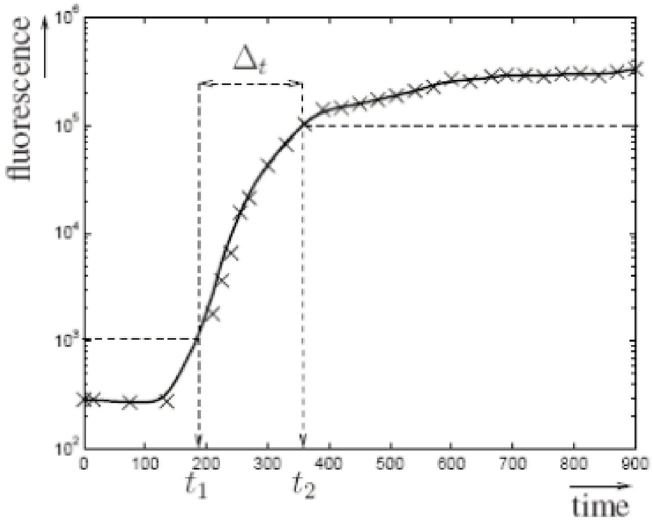
where $T(p)$ is the trace obtained by numerical integration of the ODE for perturbation p

Application to Synthetic Biology in *E. Coli*

Cascade of transcriptional inhibitions implemented in *E.coli* [Weiss et al PNAS 05]



The output protein EYFP is controlled by the small input molecule aTc



The system is well-timed if EYFP remains below 10^3 for at least 150 min., then exceeds 10^5 after at most 450 min., and switches from low to high levels in less than 150 min.

Specifying the expected behavior in LTL($\mathbb{R}$)

The timing specifications can be formalized in temporal logic as follows:

$$\begin{aligned} \phi(t_1, t_2) = & \quad \mathbf{G}(\mathit{time} < t_1 \rightarrow [\mathtt{EYFP}] < 10^3) \\ \wedge & \quad \mathbf{G}(\mathit{time} > t_2 \rightarrow [\mathtt{EYFP}] > 10^5) \\ \wedge & \quad t_1 > 150 \wedge t_2 < 450 \wedge t_2 - t_1 < 150 \end{aligned}$$

which is abstracted into

$$\begin{aligned} \phi(t_1, t_2, b_1, b_2, b_3) = & \quad \mathbf{G}(\mathit{time} < t_1 \rightarrow [\mathtt{EYFP}] < 10^3) \\ \wedge & \quad \mathbf{G}(\mathit{time} > t_2 \rightarrow [\mathtt{EYFP}] > 10^5) \\ \wedge & \quad t_1 > b1 \wedge t_2 < b_2 \wedge t_2 - t_1 < b_3 \end{aligned}$$

for computing validity domains for b_1, b_2, b_3
with the objective $b_1 = 150$, $b_2 = 450$, $b_3 = 150$ for computing the satisfaction degree in a given trace.

Improving robustness

Variance-based global sensitivity indices

$$S_i = \frac{Var(E(R|P_i))}{Var(R)} \in [0, 1]$$

S_γ	20.2 %	$S_{\kappa_{eyfp},\gamma}$	8.7 %
$S_{\kappa_{eyfp}}$	7.4 %	$S_{\kappa_{cI},\gamma}$	6.2 %
$S_{\kappa_{cI}}$	6.1 %	$S_{\kappa^0_{cI},\gamma}$	5.0 %
$S_{\kappa^0_{lacI}}$	3.3 %	$S_{\kappa^0_{cI},\kappa_{eyfp}}$	2.8 %
$S_{\kappa^0_{cI}}$	2.0 %	$S_{\kappa_{cI},\kappa_{eyfp}}$	1.8 %
$S_{\kappa_{lacI}}$	1.5 %	$S_{\kappa^0_{eyfp},\gamma}$	1.5 %
$S_{\kappa^0_{eyfp}}$	0.9 %	$S_{\kappa^0_{cI},\kappa_{cI}}$	1.1 %
$S_{u_{aTc}}$	0.4 %	$S_{\kappa^0_{cI},\kappa_{lacI}}$	0.5 %
total first order	40.7 %	total second order	31.2 %

degradation factor γ has the strongest impact on the cascade.

aTc variations have a very low impact

different importance of the basal κ^0_{eyfp} and regulated κ_{eyfp} EYFP production rates

the basal production of EYFP is due to an incomplete repression of the promoter by CI (high effect of κ_{cI}) rather than a constitutive leakage of the promoter (low effect of κ^0_{eyfp}). [Rizk Batt Fages Soliman ISMB'09 Bioinformatics]

Conclusion

- New focus in Systems Biology: formal methods from Computer Science
  - Beyond diagrammatic notations: formal semantics, abstract interpretation
  - Beyond curve fitting: high-level specifications in temporal logic
  - Automatic model-checking. Parameter optimization. Model reduction

Conclusion

- New focus in Systems Biology: formal methods from Computer Science
  - Beyond diagrammatic notations: formal semantics, abstract interpretation
  - Beyond curve fitting: high-level specifications in temporal logic
  - Automatic model-checking. Parameter optimization. Model reduction
- New focus in Programming: numerical methods
  - Beyond discrete machines: stochastic or continuous or hybrid dynamics
  - Quantitative transition systems
  - Temporal logic constraint solving, continuous satisfaction degree, optimization

Conclusion

- New focus in Systems Biology: formal methods from Computer Science
  - Beyond diagrammatic notations: formal semantics, abstract interpretation
  - Beyond curve fitting: high-level specifications in temporal logic
  - Automatic model-checking. Parameter optimization. Model reduction
- New focus in Programming: numerical methods
  - Beyond discrete machines: stochastic or continuous or hybrid dynamics
  - Quantitative transition systems
  - Temporal logic constraint solving, continuous satisfaction degree, optimization
- Synthetic Biology
  - Program the living with programming tools
  - Computational design of gene circuits

Acknowledgments

- Contraintes group at INRIA Paris-Rocquencourt on this topic:

  Grégory Batt, Sylvain Soliman, Elisabetta De Maria, Aurélien Rizk, Steven Gay, Faten Nabli, Xavier Duportet, Janis Ulhendorf, Dragana Jovanovska
- EraNet SysBio C5Sys (follow up of FP6 Tempo) on cancer chronotherapies, coord. *Francis Lévi*, INSERM; *Jean Clairambault* INRIA; …

  Coupled models of cell and circadian cycles, p53/mdm2, cytotoxic drugs.
- INRIA/INRA project Regate coord. *F. Clément* INRIA; *E. Reiter, D. Heitzler*

  Modeling of GPCR Angiotensine and FSH signaling networks
- ANR project Calamar, coord. *C. Chaouiya, D. Thieffry* , ENS, *L. Calzone,* Curie...

  Modularity and Compositionality in regulatory networks.
- OSEO Biointelligence, coord. *Dassault-Systèmes,*

  Technology transfer of Biocham concepts and tools

T7 Bacteriophage Infection Process

Retrovirus:

- Template nucleic acids (RNA): tem (initial condition of low infection)
- Genomic nucleic acids (DNA): gen
- Structural proteins: struc

```
MA(c1) for gen => tem.
MA(c2) for tem => _.
MA(c3) for tem => tem+gen.
MA(c4) for gen+struc => virus.
MA(c5) for tem => tem+struc.
MA(c6) for struc => _.
```

```
parameter(c1,0.025).
parameter(c2,0.25).
parameter(c3,1.0).
parameter(c4,0.0000075).
parameter(c5,1000).
parameter(c6,1.99).
```

T7 Bacteriophage ODE semantics

MA(c1) for gen => tem.

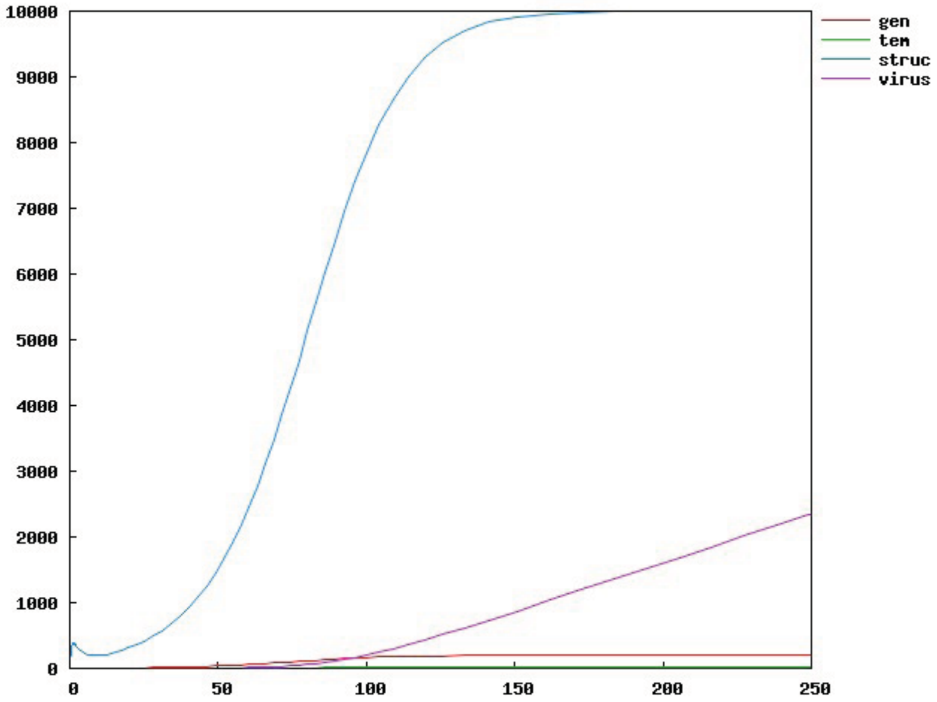
MA(c2) for tem => _.

MA(c3) for tem => tem+gen.

MA(c4) for gen+struc => virus.

MA(c5) for tem => tem+struc.

MA(c6) for struc => _.



[tem] =1

d[virus]/dt=c4*[gen]*[struc]

d[tem]/dt=c1*[gen]-c2*[tem]

d[struc]/dt=c5*[tem]-c6*[struc]-c4*[gen]*[struc]

d[gen]/dt=c3*[tem]-c4*[gen]*[struc]-c1*[gen]

T7 Bacteriophage ODE semantics

2 steady states dX/dt=0 :

1) tem=gen=struc=0

2) tem=20 gen=200 struc=10000

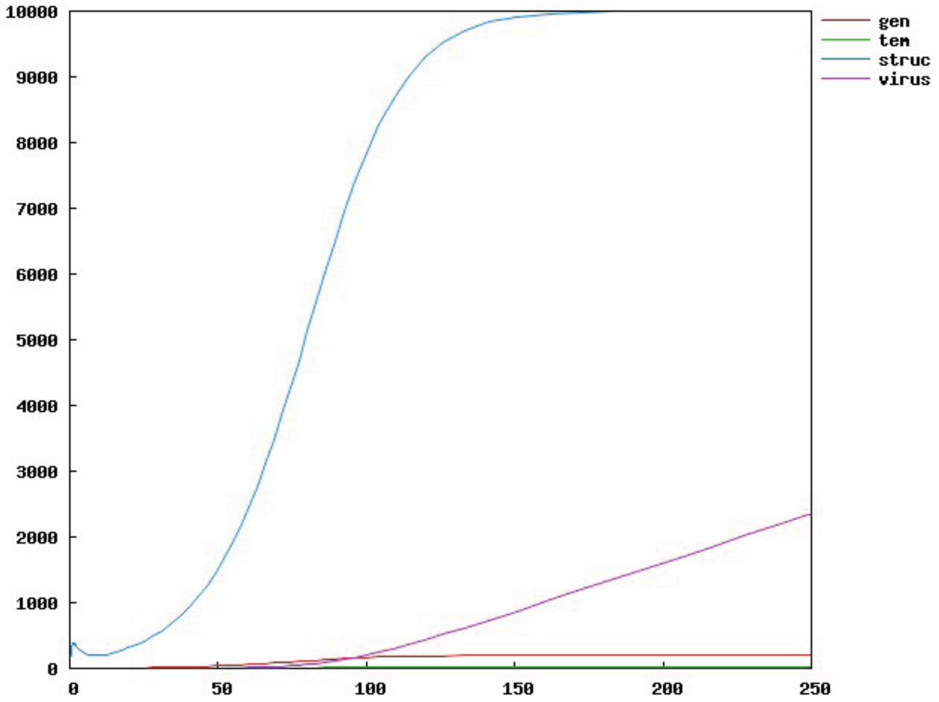
cyclic behavior

Minimal T-invariants in Petri nets

*I*V=0 (incidence stoichiometric matrix)*

Elementary Modes of flux distribution

in metabolic networks



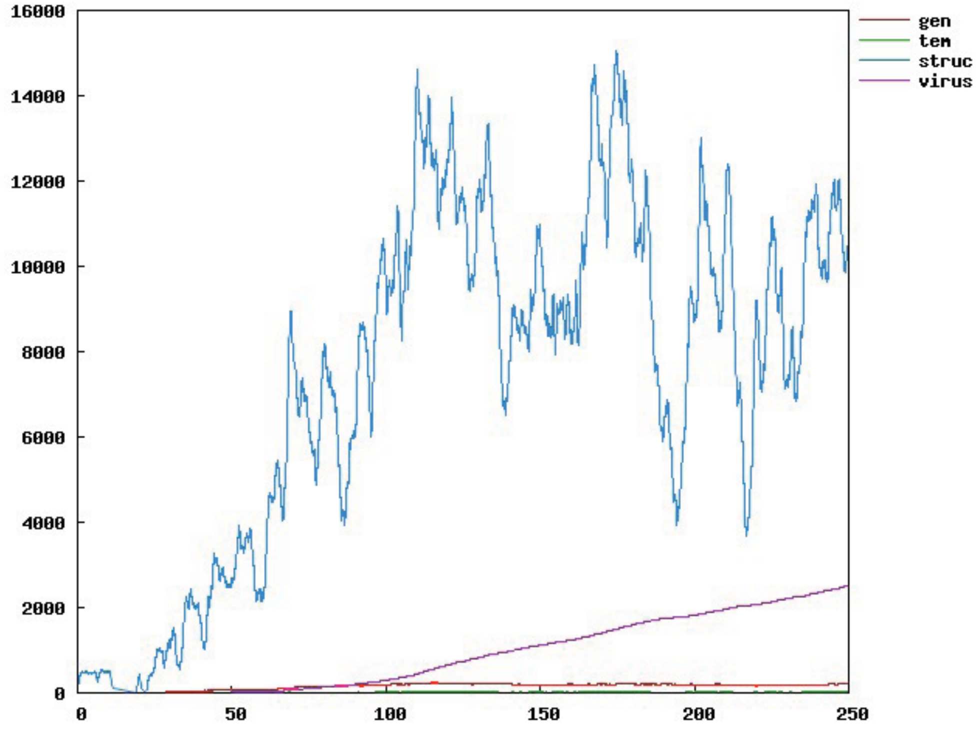
tem = - (c1*c6)/(- (c2*c5)-c1*c5+c3*c5)*(c2*c4/(- (c1*c3)+c2*c3))^ - 1

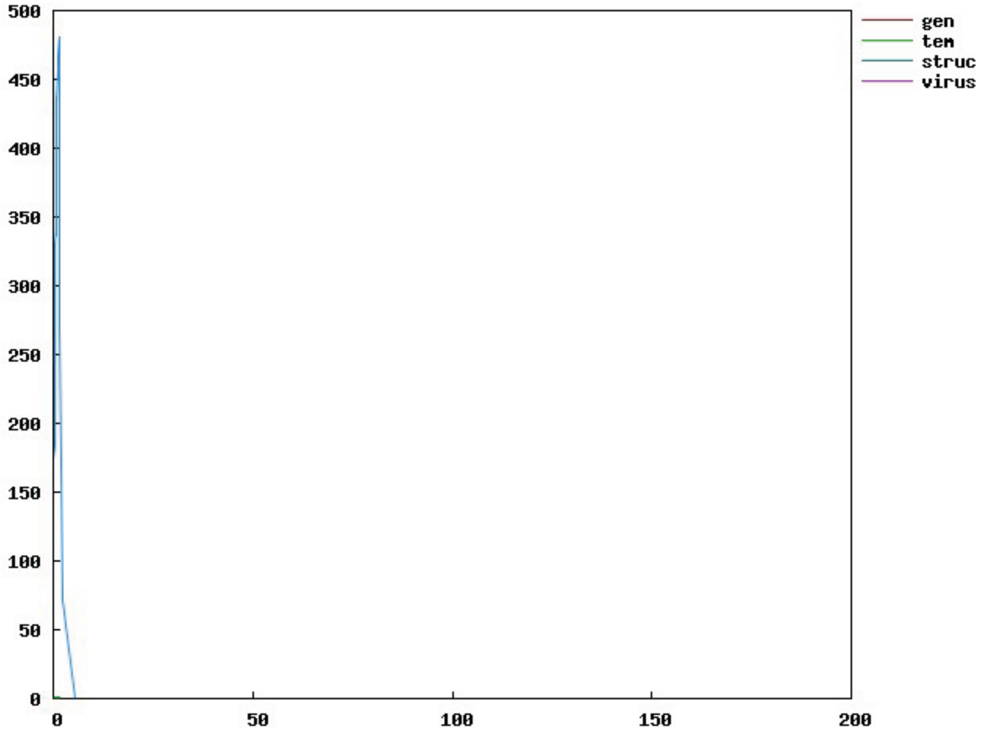
gen = ((c2*c6/(- (c3*c5)+c2*c5+c1*c5))^ - 1*(c2*c4/(- (c1*c3)+c2*c3)))^ - 1

struc = (c2*c4/(- (c1*c3)+c2*c3))^ - 1

T7 Bacteriophage Stochastic semantics

Two qualitatively different behaviors:





Similar viral explosion

Recovery

T7 Bacteriophage Boolean Semantics

Querying all possible behaviors in Computation Tree Logic
(symbolic model-checking NuSMV)

```
biocham: nusmv( EF ( !(gen) & !(tem) & !(struc) & !(virus))).
true


biocham: why.
   tem is present
2 tem => _.
   tem is absent
Query time: 0.00 s
```

T7 Bacteriophage Boolean Semantics

```
biocham: nusmv( EF ( virus & EF ( !(gen) & !(tem) & !(struc)) )).
true
biocham: why.
   tem is present
5 tem => struc+tem.
   struc is present
3 tem => gen+tem.
   gen is present
4 gen+struc => virus.
   gen is absent
   struc is absent
   virus is present
2 tem => _.
   tem is absent
```

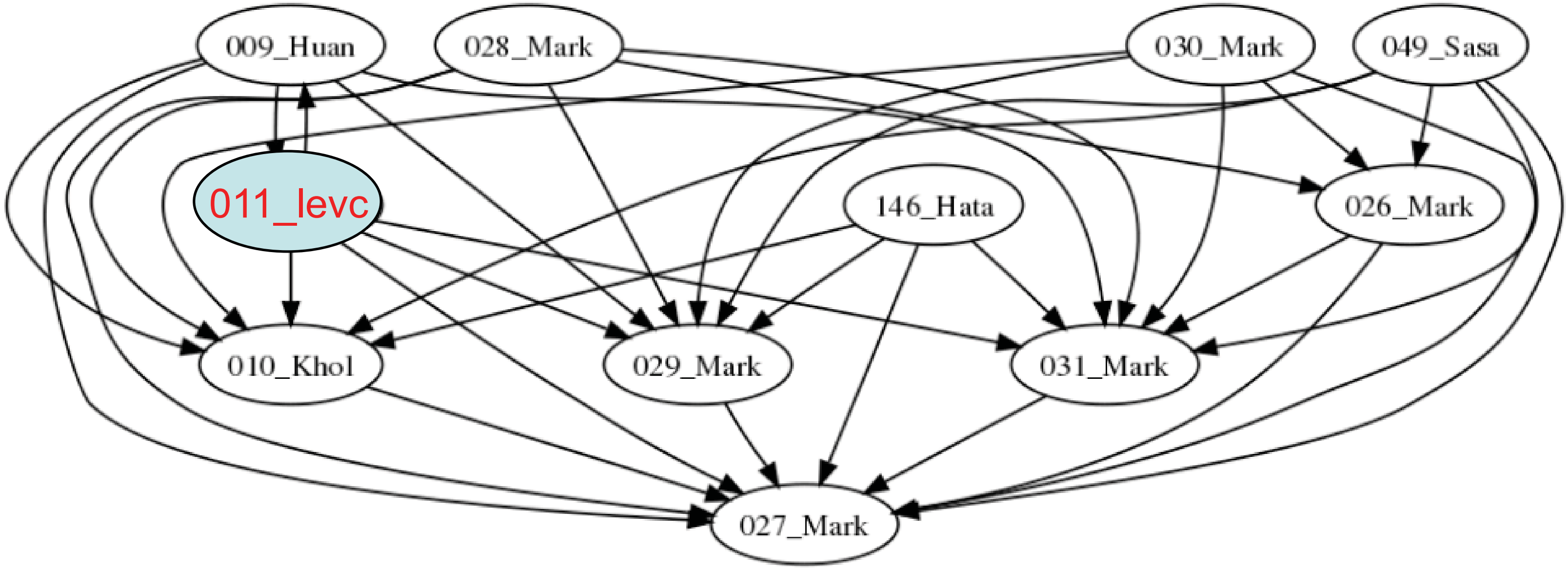
Aut. Generation of Temporal Logic Properties

Ei(reachable(gen))
Ei(reachable(!(gen)))
Ei(steady(!(gen)))
Ai(checkpoint(tem,gen)))
Ei(reachable(tem))
Ei(reachable(!(tem)))
Ei(steady(tem))
Ai(checkpoint(gen,tem)))
Ei(reachable(struc))
Ei(reachable(!(struc)))
Ei(steady(!(struc)))

Ai(checkpoint(tem,struc)))
Ei(reachable(virus))
Ei(reachable(!(virus)))
Ei(steady(!(virus)))
Ai(checkpoint(gen,virus)))
Ai(checkpoint(gen,!(virus))))
Ai(checkpoint(tem,!(virus))))
Ai(checkpoint(struc,virus)))

Time: 0.06 s

Hierarchies of Models

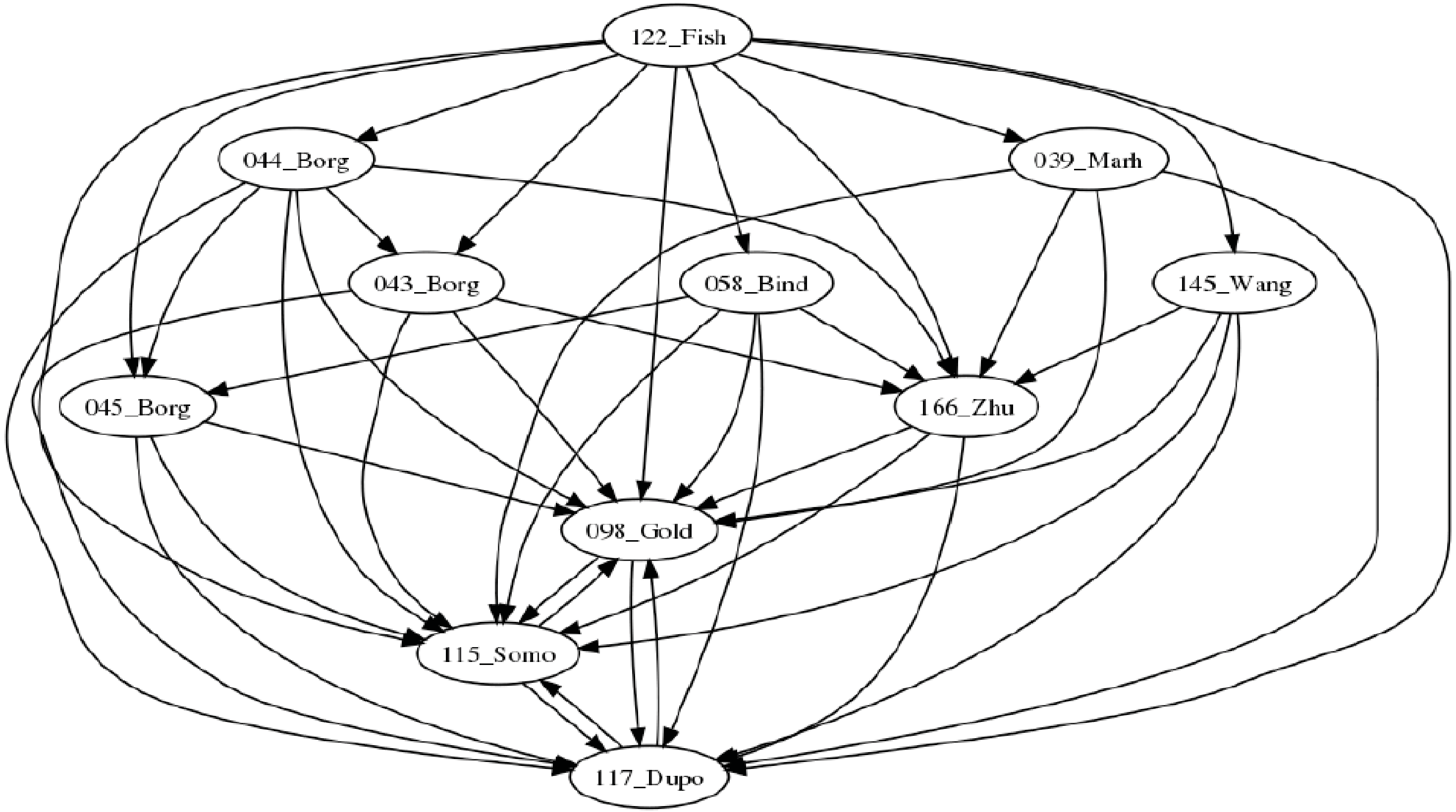


MAPK models from SBML model repository http://www.biomodels.net

[Gay Fages Soliman, A graph matching method for reducing and relating models 2010 biionfo]

- 4 graph operations: delete/merge species/reactions
- Model reductions/refinements as subgraph epimorphisms
- Query language for model repositories (constraint program)

Hierarchies of Models

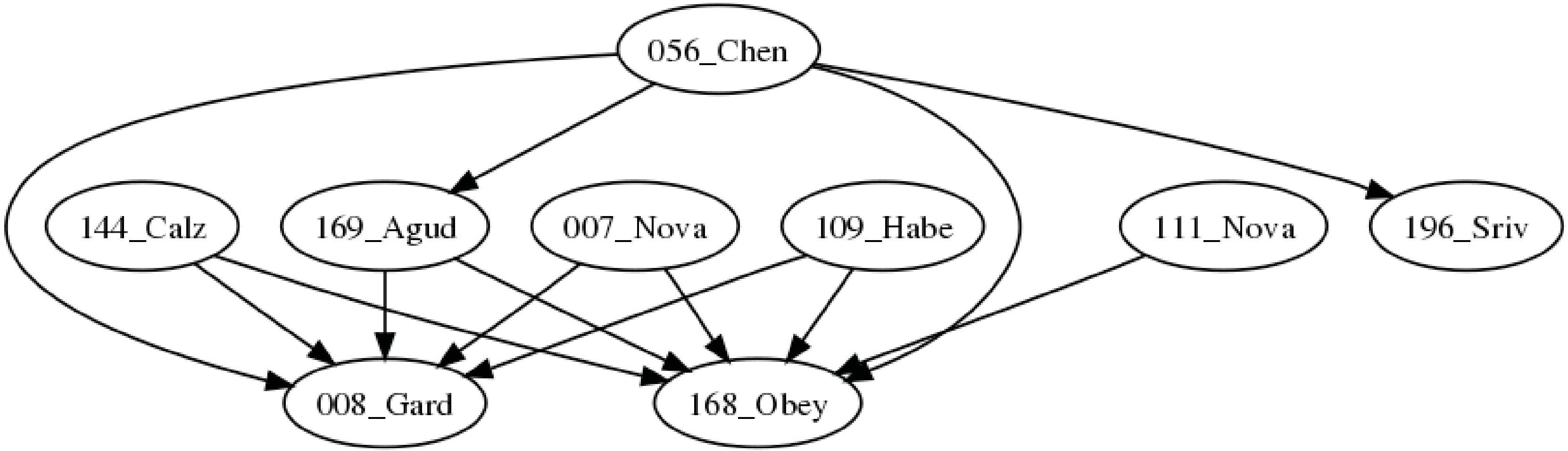


Models of circadian clock in http://www.biomodels.net

Computation of subgraph epimorphisms by constraint programming

Scales up to most of the 241^2 pairs of models in biomodels.

Hierarchies of Models



- Cell cycle models from biomodels.net
- ODE models not well transcribed in SBML (missing reactants)
- Imperfect model reductions found by graph matching

A Programmer View at Cell Computations

Size of genome

- 5 Mb for bacteria: normal size program (Biocham binary: 15Mb as yeast)
- 3 Gb for human: normal size of a video not for a program
- 140 Gb for lung fish: nature error !

Speed of interactions

- Protein interactions: enzyme-substrate collisions at 0,5 Mhz, quite slow
- Gene expression: hours ! as slow as reinstalling an operating system

Concurrent computation paradigm

- Chemical metaphor for concurrent programming [Banatre, Le Metayer 86]
- CHAM [Berry Boudol 90] to express the operational semantics of the Pi-Calculus
- Membranes for modules: just like cell compartments

Hybrid continuous+discrete computations *(energy + information)*

- Trend for future: more *physics* in *informatics*, more *informatics* in *physics*