A Model Checking Approach to Dynamical Systems Analysis

David Šafránek

with Nikola Beneš, Luboš Brim, Martin Demko, Samuel Pastva

sybila systems biology laboratory

Masaryk University
Czech Republic

Outline

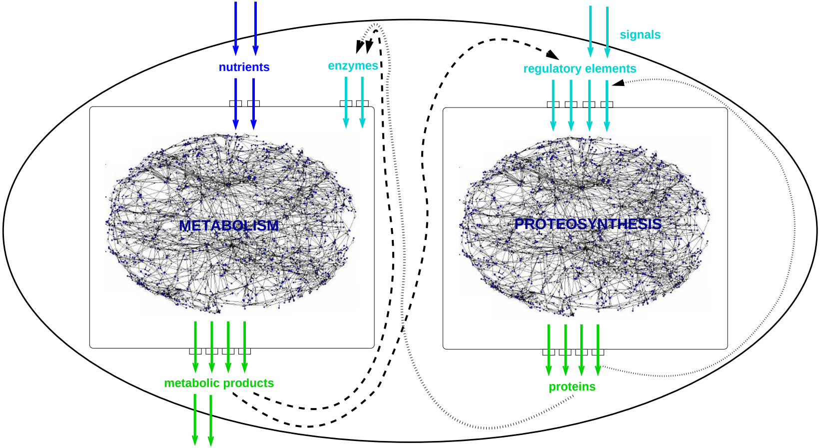
Outline

1. Motivation

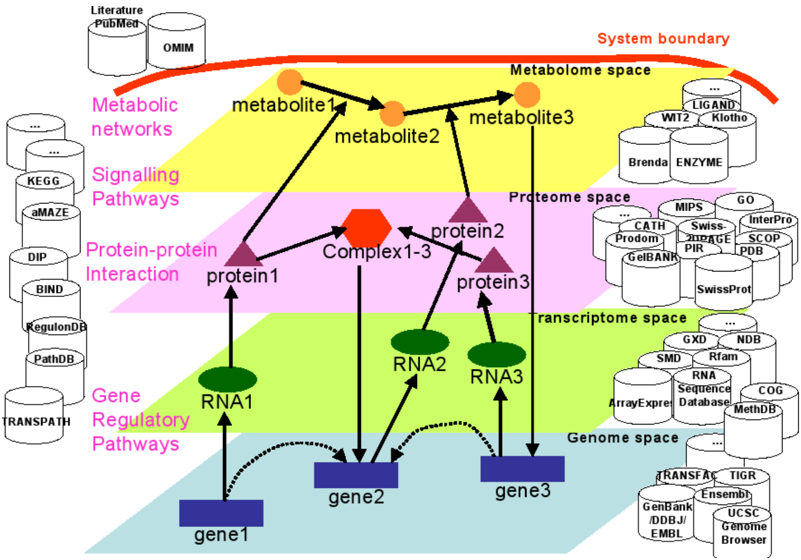
2. Parameter Synthesis by Coloured Model Checking
   - Case Study using Parameter Synthesis

3. Discrete Bifurcation Analysis
   - Case Study using Discrete Bifurcation Analysis

Motivation: Complex Real-World Systems
Systems View of Processes Driving the Cell



Motivation: Complex Dynamics of a Cell



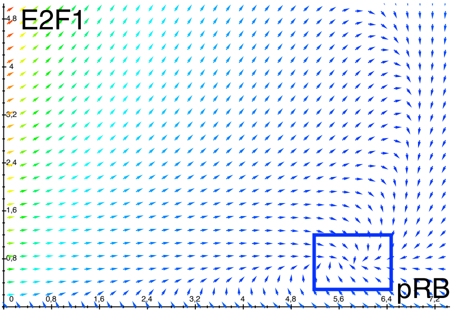
- continuous-time models of dynamical systems:

$$\dot{x} = f(x(t), p)$$

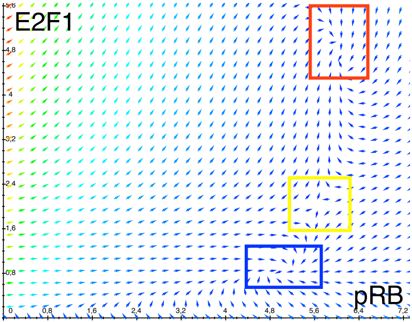
f … phase space (vector field), $f : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$

x … state vector ($\mathbb{R}^n$)

p … parameter vector ($\mathbb{R}^m$)



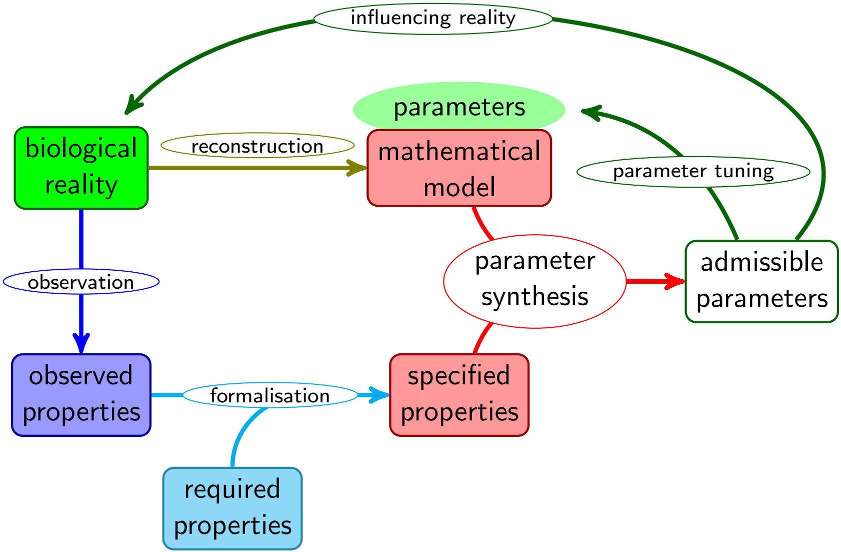
$p = 0.006$



$p = 0.012$

- *biophysics*: often use **parameterised** continuous-time models (ODEs), typically analysed by **local** methods (simulation)
- *biology*: observations in the form of **time-series data**
- literature provides further **constraints** on systems dynamics
- *computer science*: turn all known facts into **formal specification** and find admissible model **parameters**
- a suitable formal language is provided by **temporal logics**
- if the model is given as a state-transition system we can employ **model checking**
  $\Rightarrow$ **exhaustive** – global view wrt parameters and initial conditions, different than simulation

Motivation: Dynamical Systems with Parameters

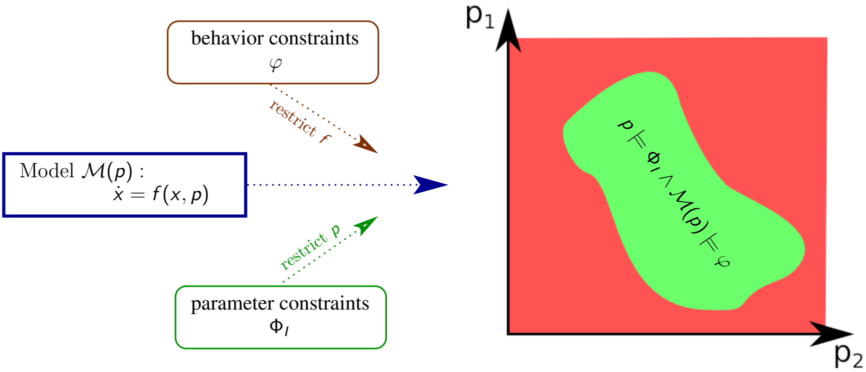


Outline

1. Motivation

2. Parameter Synthesis by Coloured Model Checking
   - Case Study using Parameter Synthesis

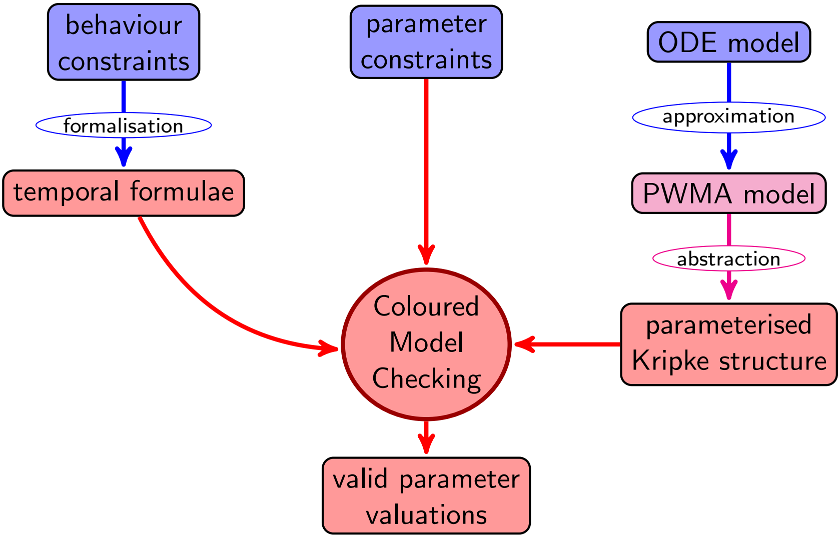
3. Discrete Bifurcation Analysis
   - Case Study using Discrete Bifurcation Analysis



Parameter Synthesis Problem

Assume $\mathcal{P}$ is the admissible **parameter space**. Given a *behaviour constraint* φ, *parameter constraint* Φ_I, and a *parameterised model* $\mathcal{M}$, **find the maximal set** $P \subseteq \mathcal{P}$ **of parameterisations** such that $p \models \Phi_I$ and $\mathcal{M}(p) \models \varphi$ for all $p \in P$.

Workflow


behaviour constraints
formalisation
temporal formulae
parameter constraints
ODE model
approximation
PWMA model
abstraction
parameterised Kripke structure
Coloured Model Checking
valid parameter valuations

Work Chronology

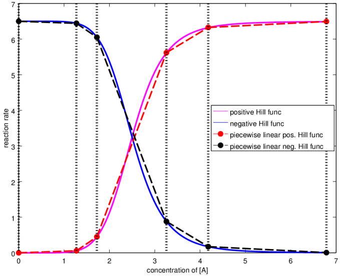
Related Work

- Batt et al. 2007: *RoverGene*, BDD/Polytopes-based approach
- Batt et al. 2010: *GNA*, symbolic approach, piecewise affine
- Grosu et al. 2011: *RoverGene* revisited, approximation improved
- Bogomolov et al. 2015, *SpaceEx*, multi-affine hybrid automata

Our Contribution

- HIBI 2010, TCCB 2012: coloured LTL model checking, piecewise multi-affine, parallel algorithm
- CMSB 2015: coloured CTL model checking, piecewise multi-affine, parallel algorithm
  - parameters represented as intervals
  - limitation: **independent parameters** only
- ATVA 2016, CMSB 2016: parameters represented in first order logic, SMT solver employed, **interdependent parameters**
- HSB 2015, FM 2016: discrete bifurcation analysis by coloured CTL model checking

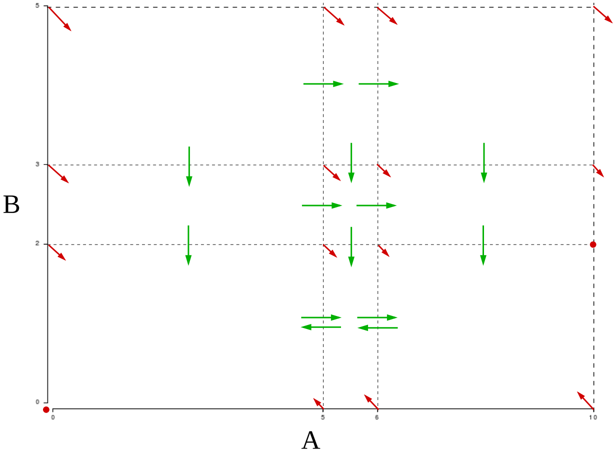
- a large class of molecular mechanisms modeled at activity-flow level (e.g., signalling pathways, gene regulatory circuits, ...)
- optimal approximation of sigmoid functions by piece-wise affine functions (ramps) [Grosu et al. CAV 2011]



model	abstraction	kinetics
piece-wise multi-affine	transient over-approximated steady state over-approximated	sigmoidal kinetics mass action
piece-wise affine	transient over-approximated steady state exact	first-order sigmoidal kinetics

Step 2: Rectangular Abstraction

- approach originates in [Batt, Belta, Habets, van Schuppen]
- continuous phase-space is partitioned into (hypher)rectangles
- **no diagonal transitions**, overapproximation

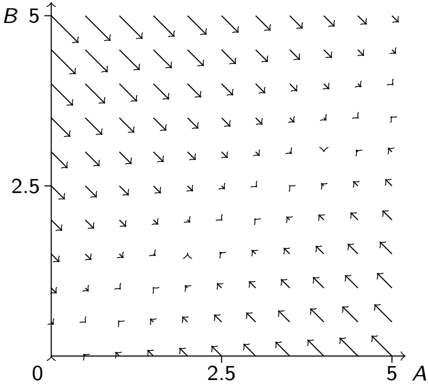


$$\frac{dA}{dt} = -k_1 \cdot A + k_2 \cdot B$$

$$\frac{dB}{dt} = k_1 \cdot A - k_2 \cdot B$$

$$k_2 = 0.8$$

$$k_1 = 0.6$$



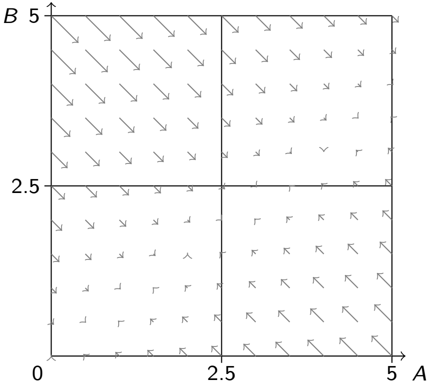
Phase Space Discretisation Leads to Parameter Space Discretisation

$$\frac{dA}{dt} = -k_1 \cdot A + k_2 \cdot B$$

$$\frac{dB}{dt} = k_1 \cdot A - k_2 \cdot B$$

$$k_2 = 0.8$$

$$k_1 = 0.6$$

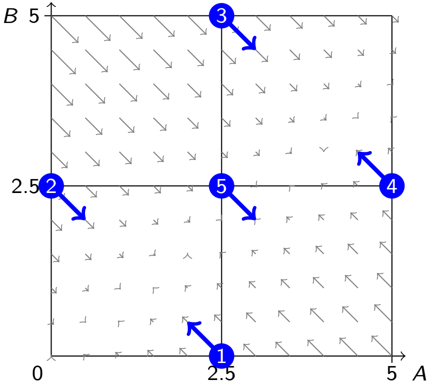


$$\frac{dA}{dt} = -k_1 \cdot A + k_2 \cdot B$$

$$\frac{dB}{dt} = k_1 \cdot A - k_2 \cdot B$$

$$k_2 = 0.8$$

$$k_1 = 0.6$$

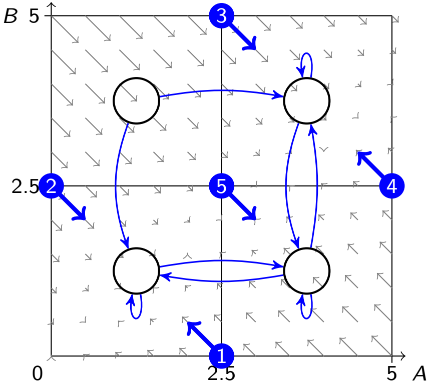


$$\frac{dA}{dt} = -k_1 \cdot A + k_2 \cdot B$$

$$\frac{dB}{dt} = k_1 \cdot A - k_2 \cdot B$$

$$k_2 = 0.8$$

$$k_1 = 0.6$$



Step 3: Parameter Synthesis

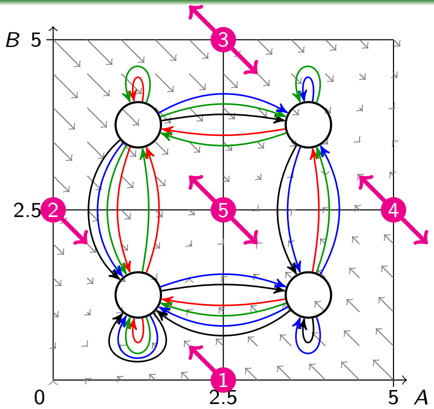
Phase Space Discretisation Leads to Parameter Space Discretisation

$$\frac{dA}{dt} = -k_1 \cdot A + k_2 \cdot B$$

$$\frac{dB}{dt} = k_1 \cdot A - k_2 \cdot B$$

$$k_2 = 0.8$$

$$k_1 = ?$$



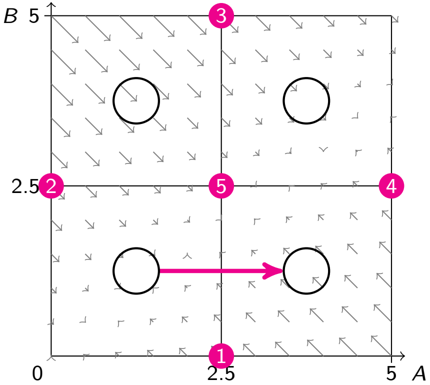
	(0,0.4)	(0.4,0.8)	(0.8,1.6)	(1.6,max)
1	↖	↖	↖	↖
2	↘	↘	↘	↘
3	↘	↘	↘	↖
4	↘	↖	↖	↖
5	↘	↘	↖	↖

$$\frac{dA}{dt} = -k_1 \cdot A + k_2 \cdot B$$

$$\frac{dB}{dt} = k_1 \cdot A - k_2 \cdot B$$

$$k_2 = 0.8$$

$$k_1 = ?$$



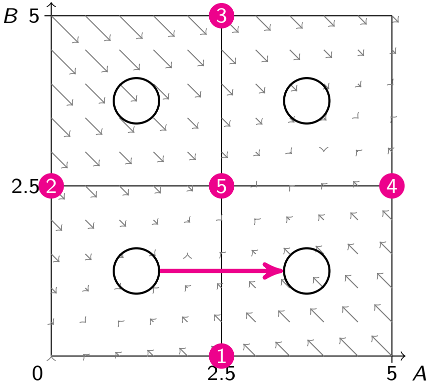
	(0,0.4)	(0.4,0.8)	(0.8,1.6)	(1.6,max)	
1	↖	↖	↖	↖	$-2.5 \cdot k_1 > 0$
2	↘	↘	↘	↘	
3	↘	↘	↘	↖	
4	↘	↖	↖	↖	
5	↘	↘	↖	↖	$-2.5 \cdot k_1 + 2.5 \cdot k_2 > 0$

$$\frac{dA}{dt} = -k_1 \cdot A + k_2 \cdot B$$

$$\frac{dB}{dt} = k_1 \cdot A - k_2 \cdot B$$

$$k_2 = 0.8$$

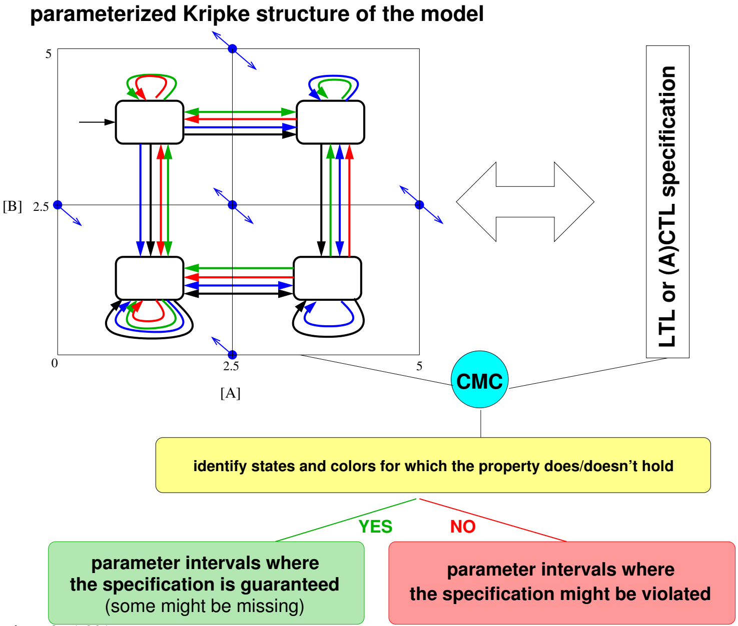
$$k_1 = ?$$



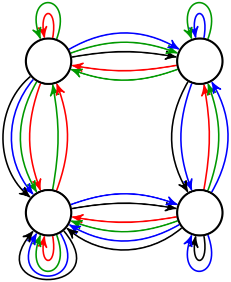
$$\Phi_{\text{state00}\to\text{state10}} := -2.5 \cdot k_1 > 0 \lor -2.5 \cdot k_1 + 2.5 \cdot k_2 > 0$$

The transition exists if and only if the formula is **satisfiable**.
Local parameter constraints are **predicates over reals**.

Parameter Synthesis by Coloured Model Checking

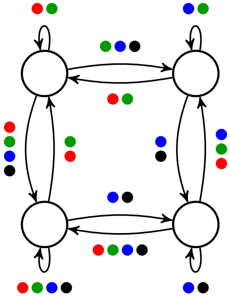


Transitions with Parameters (coloured transitions)



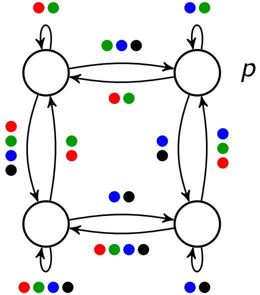
- each parameter valuation represents one Kripke structure
- shared state space, different transition space

Transitions with Parameters (coloured transitions)



- each parameter valuation represents one Kripke structure
- shared state space, different transition space
- we assume symbolic representation of parameters
- symbolic PKS: every transition is associated with a formula

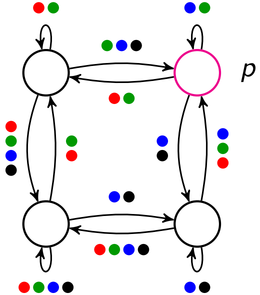
- enumerative approach
- CTL model checking: dynamic programming, back propagation
- coloured CTL model checking: back propagation of colours



CTL formula: **EF** p

- **intersection** of colour sets: state + transition
- **union** of colour sets: inside state

- enumerative approach
- CTL model checking: dynamic programming, back propagation
- coloured CTL model checking: back propagation of colours



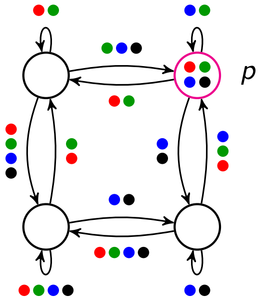
CTL formula: **EF** p

- **intersection** of colour sets: state + transition
- **union** of colour sets: inside state

Coloured CTL Model Checking

Basic Idea

- enumerative approach
- CTL model checking: dynamic programming, back propagation
- coloured CTL model checking: back propagation of colours

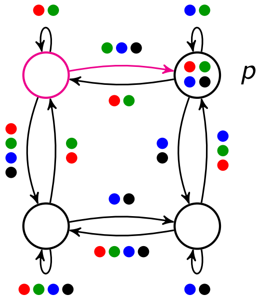


CTL formula: **EF** p

- **intersection** of colour sets: state + transition
- **union** of colour sets: inside state

Coloured CTL Model Checking
Basic Idea

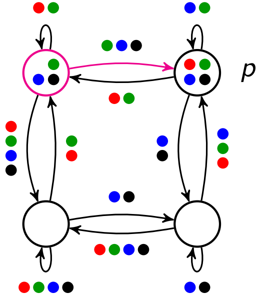
- enumerative approach
- CTL model checking: dynamic programming, back propagation
- coloured CTL model checking: back propagation of colours



CTL formula: **EF** p

- **intersection** of colour sets: state + transition
- **union** of colour sets: inside state

- enumerative approach
- CTL model checking: dynamic programming, back propagation
- coloured CTL model checking: back propagation of colours

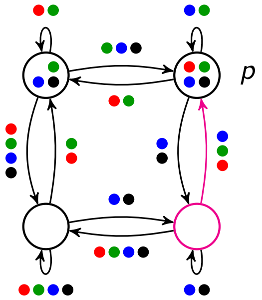


CTL formula: **EF** p

- **intersection** of colour sets: state + transition
- **union** of colour sets: inside state

Coloured CTL Model Checking
Basic Idea

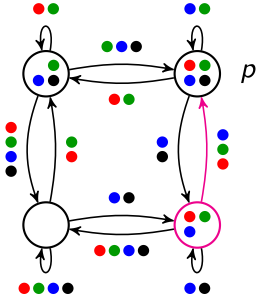
- enumerative approach
- CTL model checking: dynamic programming, back propagation
- coloured CTL model checking: back propagation of colours



CTL formula: **EF** p

- **intersection** of colour sets: state + transition
- **union** of colour sets: inside state

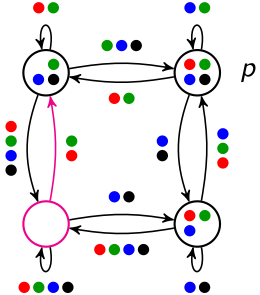
- enumerative approach
- CTL model checking: dynamic programming, back propagation
- coloured CTL model checking: back propagation of colours



CTL formula: **EF** p

- **intersection** of colour sets: state + transition
- **union** of colour sets: inside state

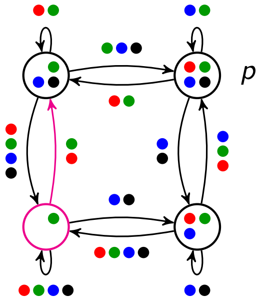
- enumerative approach
- CTL model checking: dynamic programming, back propagation
- coloured CTL model checking: back propagation of colours



CTL formula: **EF** p

- **intersection** of colour sets: state + transition
- **union** of colour sets: inside state

- enumerative approach
- CTL model checking: dynamic programming, back propagation
- coloured CTL model checking: back propagation of colours

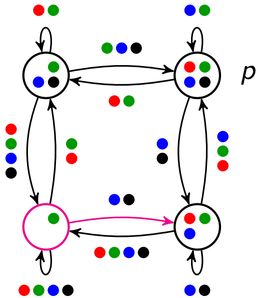


CTL formula: **EF** p

- **intersection** of colour sets: state + transition
- **union** of colour sets: inside state

Coloured CTL Model Checking
Basic Idea

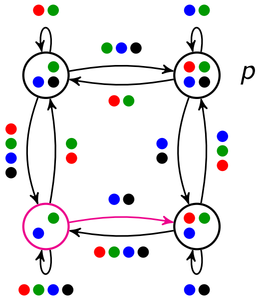
- enumerative approach
- CTL model checking: dynamic programming, back propagation
- coloured CTL model checking: back propagation of colours



CTL formula: **EF** p

- **intersection** of colour sets: state + transition
- **union** of colour sets: inside state

- enumerative approach
- CTL model checking: dynamic programming, back propagation
- coloured CTL model checking: back propagation of colours

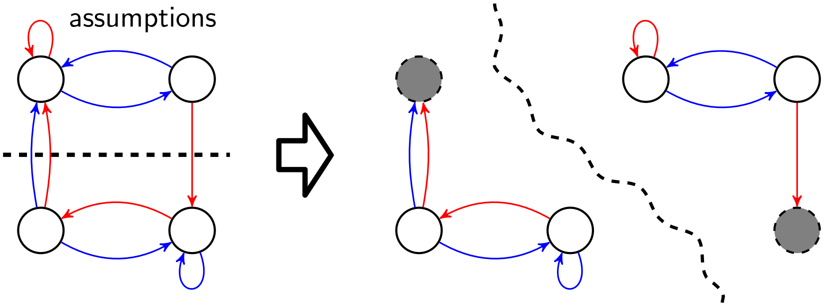


CTL formula: **EF** p

- **intersection** of colour sets: state + transition
- **union** of colour sets: inside state

Kripke Fragments

- each worker owns a part of the whole state space
- extended with *border states*
- assumption-based approach, three-valued (*true*/*false*/*unknown*)
- after everything is computed locally, exchange border state assumptions



Idea based on:
Brim, Yorav, Žídková 2005: parallel CTL model checking

Encoding

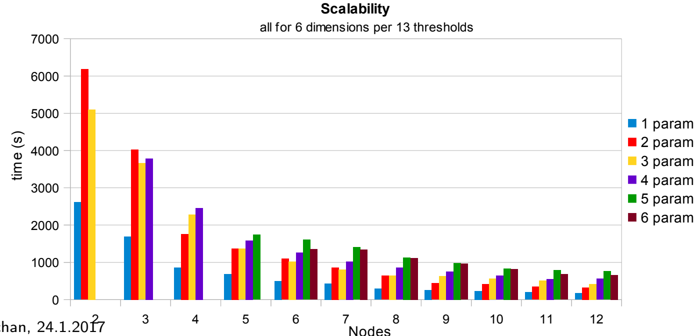
- every set of parameters (on transitions, inside states) represented by a formula with free variables: satisfying assignments are set elements
- union is disjunction, intersection is conjunction
- call SMT solver to check whether a formula is satisifiable (i.e. whether the set is nonempty)
- call SMT solver to check whether two formulae are equivalent (i.e. whether the set has changed)
- optimisation: delay SMT solver calls, cache SMT results

This work: linear arithmetic over reals
Generally: anything an SMT solver can handle

$$S + E \rightleftharpoons ES_1 \rightleftharpoons \cdots \rightleftharpoons ES_k \rightleftharpoons P + E$$

$$\dot{S} = 0.1 \cdot ES_1 - p_1 \cdot E \cdot S$$
$$\dot{E} = 0.1 \cdot ES_1 - p_2 \cdot E \cdot S + 0.1 \cdot ES_k - p_2 \cdot E \cdot P$$
$$\dot{ES_1} = 0.01 \cdot E \cdot S - p_3 \cdot ES_1 + 0.05 \cdot ES_2$$
$$\vdots$$
$$\dot{ES_k} = 0.1 \cdot ES_{k-1} - p_k \cdot ES_k + 0.01 \cdot E \cdot P$$
$$\dot{P} = 0.1 \cdot ES_k - p_{k+1} \cdot E \cdot P - 0.1 \cdot P$$

$$p_1 = 0.01, p_2 = 0.01, p_3 = 0.2,$$
$$p_k = 0.15, p_{k+1} = 0.01$$

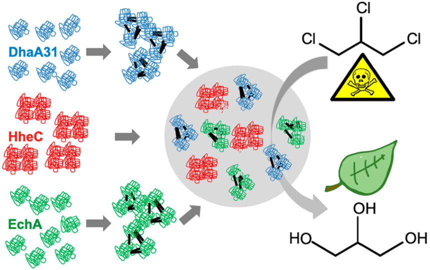


Outline

1. Motivation
2. Parameter Synthesis by Coloured Model Checking
   - Case Study using Parameter Synthesis
3. Discrete Bifurcation Analysis
   - Case Study using Discrete Bifurcation Analysis

Case study: Biodegradation of Trichloropropane in *E. coli*

$$\textbf{TCP} \xrightarrow{\text{DhaA}} \textbf{DCP} \xrightarrow{\text{HheC}} \textbf{ECH} \xrightarrow{\text{EchA}} \textbf{CPD} \xrightarrow{\text{HheC}} \textbf{GDL} \xrightarrow{\text{EchA}} \textbf{GLY}$$



$$\frac{d[TCP]}{dt} = -\frac{k_1 \cdot DhaA \cdot [TCP]}{K_{m,1} + [TCP]}$$

$$\frac{d[DCP]}{dt} = \frac{k_1 \cdot DhaA \cdot \cdot [TCP]}{K_{m,1} + [TCP]} - \frac{k_2 \cdot HheC \cdot [DCP]}{K_{m,2} + [DCP]}$$

$$\frac{d[ECH]}{dt} = \frac{k_2 \cdot HheC \cdot [DCP]}{K_{m,2} + [DCP]} - \frac{k_3 \cdot EchA \cdot [ECH]}{K_{m,3} + [ECH]}$$

$$\frac{d[CPD]}{dt} = \frac{k_3 \cdot EchA \cdot [ECH]}{K_{m,3} + [ECH]} - \frac{k_4 \cdot HheC \cdot [CPD]}{K_{m,4} + [CPD]}$$

$$\frac{d[GDL]}{dt} = \frac{k_4 \cdot HheC \cdot [CPD]}{K_{m,4} + [CPD]} - \frac{k_5 \cdot HheC \cdot [GDL]}{K_{m,5} + [GDL]}$$

$$\frac{d[GLY]}{dt} = \frac{k_5 \cdot HheC \cdot [GDL]}{K_{m,5} + [GDL]}$$

- biodegradation of toxic substrate and intermediates
- synthetic pathway utilising enzymes from two other bacteria

  Rhodococcus rhodochrous NCIMB 13064; *Agrobacterium radiobacter* AD1
- find optimal enzymes concentration balancing *metabolic burden* and *toxicity*

Case study: Biodegradation of Trichloropropane in *E. coli*

Desired behaviour:

"*TCP is finally completely degraded and the concentration of intermediates does not exceed given bounds*"

Formally:

$$\varphi_1 = (\mathbf{A}([TCP] > x)\mathbf{U}(\mathbf{AF}(\mathbf{AG}\ [TCP] < y))),$$

$$\varphi_2 = (\mathbf{A}([GLY] < y)\mathbf{U}(\mathbf{AF}(\mathbf{AG}\ [GLY] > x))),$$

$$\varphi_3 = (\mathbf{AG}\ [DCP] < v) \wedge (\mathbf{AG}\ [GDL] < w),$$

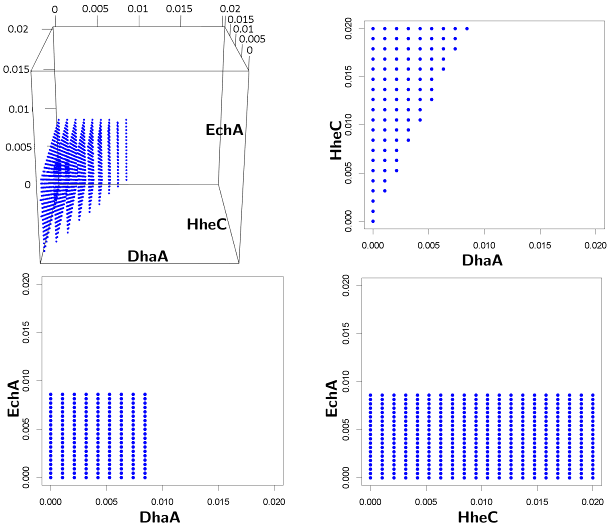
$$\varphi = (\varphi_1 \wedge \varphi_2 \wedge \varphi_3),$$

where x, y, v and w are estimated values making an instance of this property:

- $x = 1.9$ (according to authors[1] using the value 2 mM),
- $y = 0.01$ (obviously, cannot be zero),
- $v \in \{0.5, 0.3, 0.1\}$ (variations based on experimental data observation)
- $w \in \{0.5, 0.25, 0.1\}$ (variations based on experimental data observation)

[1] Kurumbang et al., ACS Synthetic Biology, 2013

Case study: Biodegradation of Trichloropropane in *E. coli*

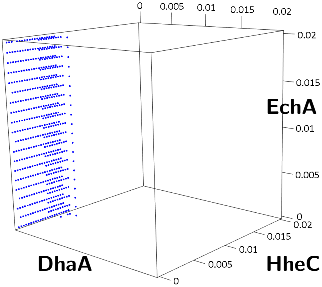


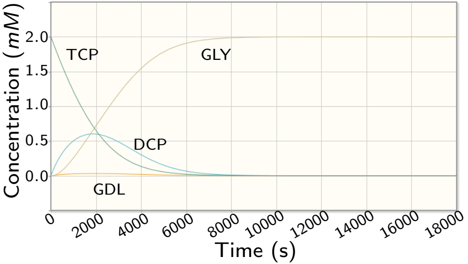
A sample of the resulting parameter space for a particular initial state:
TCP $\in [1.9, 1.9586]$, DCP $\in [0.448898, 0.5]$, GDL $\in [0.0, 0.0669138]$, GLY $\in [0.0, 0.01]$

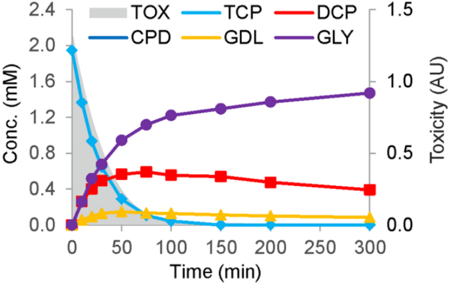
Dotted area corresponds to φ ($v = 0.5$, $w = 0.25$).

Case study: Biodegradation of Trichloropropane in *E. coli*

Preliminary Biological Validation





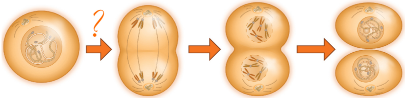


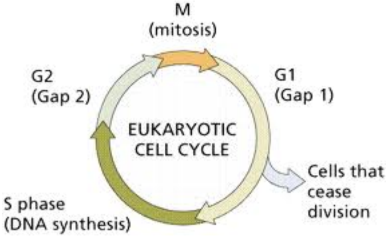
Outline

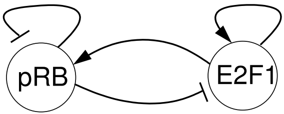
Motivation: How Parameters Influence Systems Dynamics?

Example: decision making in living cells

— *to divide or not to divide?*



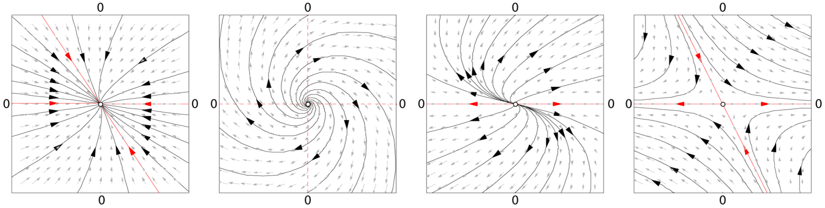




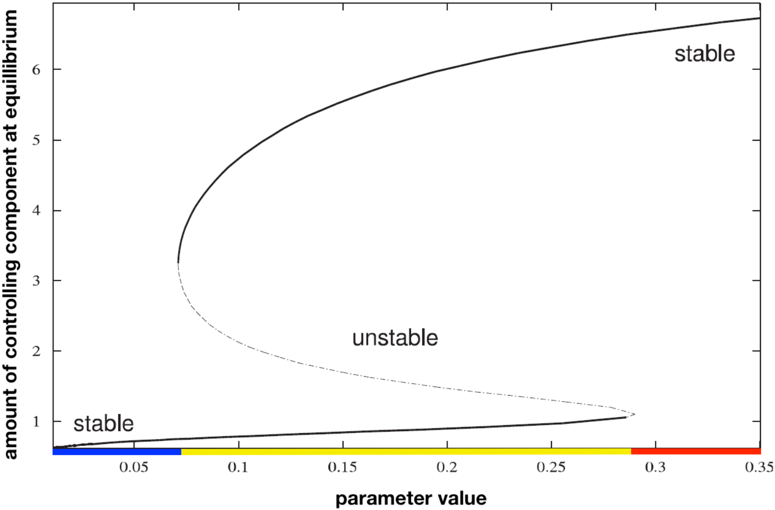
[Swat et al. 2004]

decisions implemented by circuits of positive and negative interactions
modelling of cell cycle since 1970 [Goldbetter et al.]

typical **phase portraits** around equilibria:



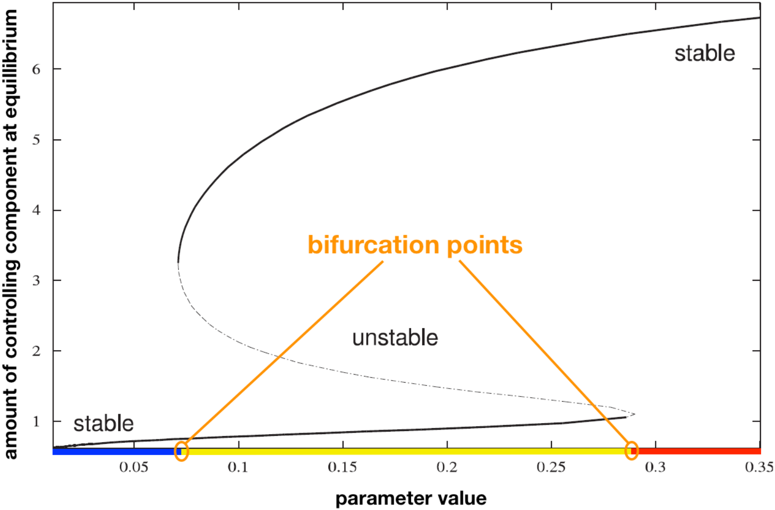
- bifurcation is defined as a topological change in phase space
  - small change in parameter $\Longrightarrow$ qualitative change in dynamics
  - the goal of bifurcation analysis is to **identify bifurcation points**



cannot divide can decide for division must divide

Motivation: How Parameters Influence Systems Dynamics?

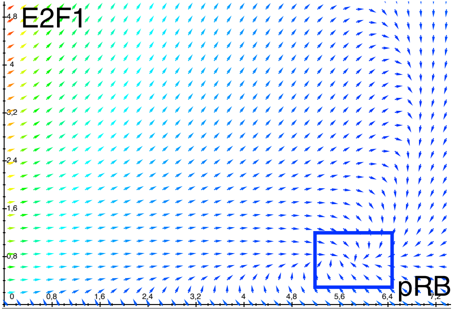
Bifurcation Analysis in Systems Theory



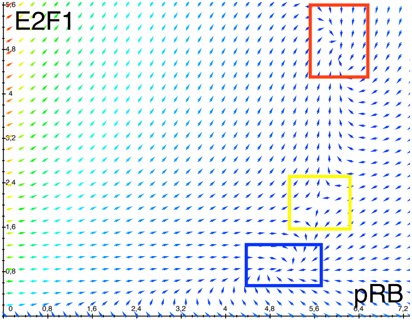
cannot divide | can decide for division | must divide

Motivation: Models of Complex Dynamical Systems
Understanding Role of Parameters

- in the vector field the equilbria have certain patterns
- the patterns change with parameters (appear, disappear, change shape)

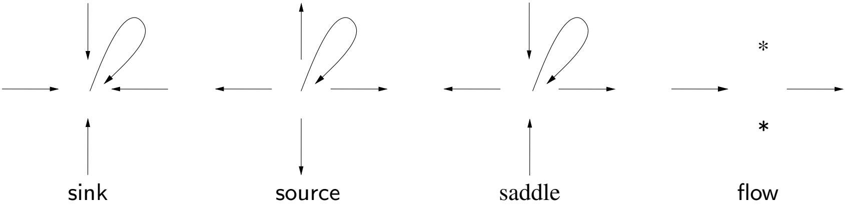


$p = 0.006$

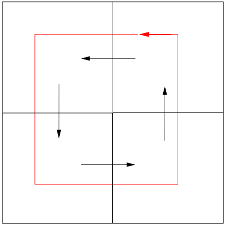


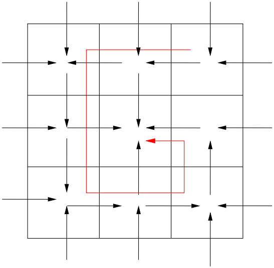
$p = 0.012$

Single-state patterns



Multi-state patterns





Phase Portrait Specification

- elementary patterns describe temporal behaviour in states
  - (in)stability, stabilisation, flow direction, ...
  - in non-deterministic system: possibility or inevitability
- employ **temporal logics** to formalise the patterns
- need to express branching over labeled transitions, future and past, state variables
  - stability (sink)
    there is a state with no outgoing transition (only a self-loop)
  - instability (source)
    there is a state with no incoming transition (only a self-loop)
  - increasing flow
    only transitions in particular direction
- some work done – UCTL [ter Beek et al.], hybrid logics [Arellano et al. 2011], ...

we combine all of these – introduce HUCTL_P

Phase Portrait Specification: HUCTL_P

- HUCTL_P — hybrid UCTL with past
- in addition to AP there are **direction formulae**:

$$\chi ::= \mathit{true} \mid d \mid \neg\chi \mid \chi \wedge \chi \text{ where } d \in \mathit{Dir}$$

- state formulae

$$\varphi ::= \mathit{true} \mid p \mid \neg\varphi \mid \varphi \wedge \varphi \mid \mathbf{E}\,\psi \mid \mathbf{A}\,\psi \mid \hat{\mathbf{E}}\,\psi \mid \hat{\mathbf{A}}\,\psi \mid x \mid \downarrow x.\varphi \mid @x.\varphi \mid \exists x.\varphi$$

- path formulae

$$\psi ::= \mathbf{X}_\chi\,\varphi \mid \varphi\,{}_\chi\mathbf{U}\,\varphi \mid \varphi\,{}_\chi\mathbf{U}_\chi\,\varphi \mid \varphi\,{}_\chi\mathbf{W}\,\varphi \mid \varphi\,{}_\chi\mathbf{W}_\chi\,\varphi$$

Phase Portrait Specification: HUCTL$_P$

Single-state patterns

- sink (stable steady state): $\downarrow s.\, \mathbf{AX}\, s$
- source (only self-loops, no other incoming): $\downarrow s.\, \hat{\mathbf{A}}\mathbf{X}\, s$
- 2d-saddle (nort-south outgoing, west-east incoming):

$$\mathbf{AX}_{N\vee S}\, \mathit{true} \wedge \mathbf{EX}_N\, \mathit{true} \wedge \mathbf{EX}_S\, \mathit{true} \wedge$$
$$\hat{\mathbf{A}}\mathbf{X}_{E\vee W}\, \mathit{true} \wedge \hat{\mathbf{E}}\mathbf{X}_E\, \mathit{true} \wedge \hat{\mathbf{E}}\mathbf{X}_W\, \mathit{true}$$

Multi-state patterns

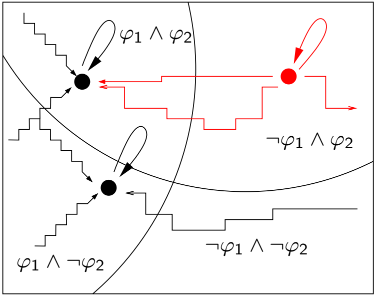
- state in a nontrivial SCC: $\downarrow s.\, \mathbf{EX}\, \mathbf{EF}\, s$
- state in a final SCC (generalised sink): $\downarrow s.\, \mathbf{AG}\, \mathbf{EF}\, s$

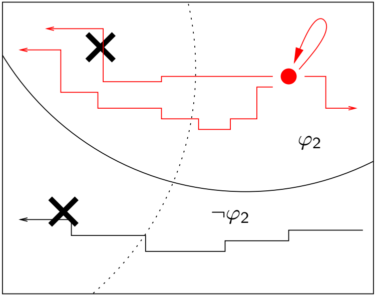
Relations among patterns

- at least two sinks in the whole system:
  $\exists s. \exists t. (@s. \neg t \wedge \mathbf{AX}\, s) \wedge (@t.\, \mathbf{AX}\, t)$

phase portrait specification $\phi = \{\varphi_1, \varphi_2\}$
phase portrait pattern $X_\phi = \{\varphi_1 \wedge \varphi_2, \varphi_1 \wedge \neg\varphi_2, \neg\varphi_1 \wedge \varphi_2, \neg\varphi_1 \wedge \neg\varphi_2\}$

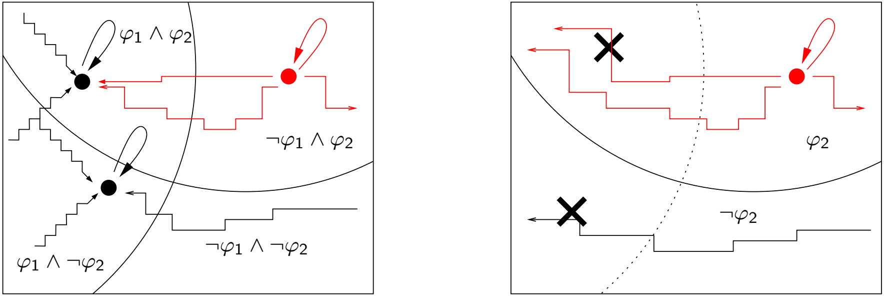
subpatterns, e.g., $X' = \{\neg\varphi_1 \wedge \varphi_2, \neg\varphi_1 \wedge \neg\varphi_2\} \subset X_\phi$





phase portrait specification $\phi = \{\varphi_1, \varphi_2\}$
phase portrait pattern $X_\phi = \{\varphi_1 \wedge \varphi_2, \varphi_1 \wedge \neg\varphi_2, \neg\varphi_1 \wedge \varphi_2, \neg\varphi_1 \wedge \neg\varphi_2\}$

subpatterns, e.g., $X' = \{\neg\varphi_1 \wedge \varphi_2, \neg\varphi_1 \wedge \neg\varphi_2\} \subset X_\phi$



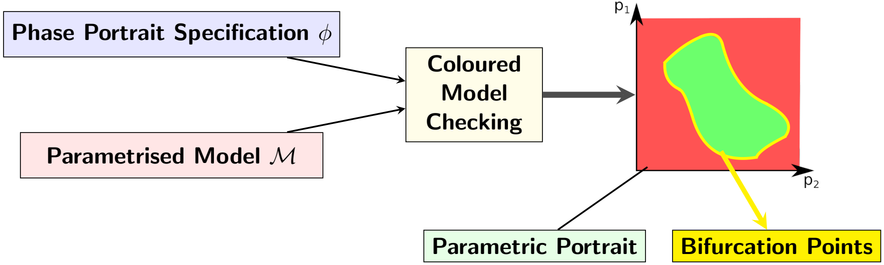
Parametric Phase Portrait

- parameter space $\mathcal{P}$
- $\mathcal{P}(\varphi) \subseteq \mathcal{P}$ all parameters for which there is a state satisfying φ
- **stratum**: $\Gamma_{X_\phi} = \bigcap_{\varphi \in X_\phi} \mathcal{P}(\varphi)$, $\Gamma_{X'} = \bigcap_{\varphi \in X'} \mathcal{P}(\varphi)$, ...
- $p \in \mathcal{P}$ is a **bifurcation point** if it is a *boundary point of some stratum*

Discrete Bifurcation Analysis Problem

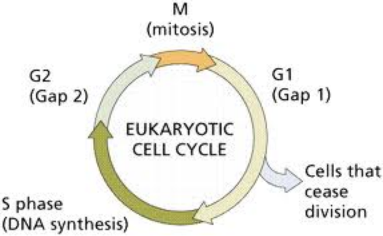
Problem Definition

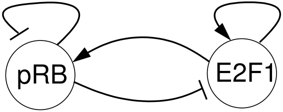
Assume $\mathcal{P}$ is a finite partially ordered domain representing the *m*-dimensional **parameter space**. Given a *parameterised model* $\mathcal{M}$ and a *phase portrait specification* ϕ, **compute the parametric portrait** of $\mathcal{M}$ wrt ϕ and **identify all bifurcation points** in $\mathcal{P}$ wrt $\mathcal{M}$ and ϕ.



Outline

Case Study: Regulation of G_1/S Cell Cycle Transition





[Swat et al. 2004]

$$\frac{d[pRB]}{dt} = k_1 \frac{[E2F1]}{K_{m1}+[E2F1]} \frac{J_{11}}{J_{11}+[pRB]} - \phi_{pRB}[pRB]$$
$$\frac{d[E2F1]}{dt} = k_p + k_2 \frac{a^2+[E2F1]^2}{K_{m2}^2+[E2F1]^2} \frac{J_{12}}{J_{12}+[pRB]} - \phi_{E2F1}[E2F1]$$

bifurcation analysis wrt ϕ_{pRB}

Analysed phase portrait pattern:

- $\varphi_1 := \exists s.\exists t.(@s.\,\mathbf{AG}\,\mathbf{EF}\,s) \wedge (@t.\neg\,\mathbf{EF}\,s \wedge \mathbf{AG}\,\mathbf{EF}\,t) \wedge \mathbf{E}_{\neg\mathrm{N}}\mathbf{F}\,s \wedge \mathbf{E}_{\neg\mathrm{S}}\mathbf{F}\,t$
- $\varphi_2 := \neg\varphi_1 \wedge {\downarrow} s.\,\mathbf{AG}\,\mathbf{EF}\,s \wedge E2F1 < 4$
- $\varphi_3 := \neg\varphi_1 \wedge {\downarrow} s.\,\mathbf{AG}\,\mathbf{EF}\,s \wedge E2F1 > 4$

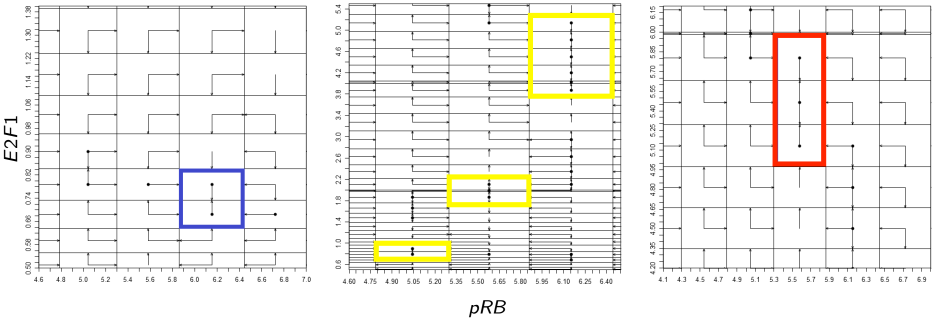
Case Study: Workflow

INPUT: phase portrait specification $\{\varphi_1, \varphi_2, \varphi_3\}$
ODE model

1. approximate the ODE model by a piece-wise multi-affine model
2. construct $\mathcal{M}$ by discretisation of the approximate model
3. run discrete bifurcation analysis procedure

OUTPUT: parametric phase portrait
bifurcation points

Case Study: Results



φ_2 $\qquad$ φ_1 $\qquad$ φ_3

$\phi_{pRB} = 0.0075$ $[0.002, 0.011]$

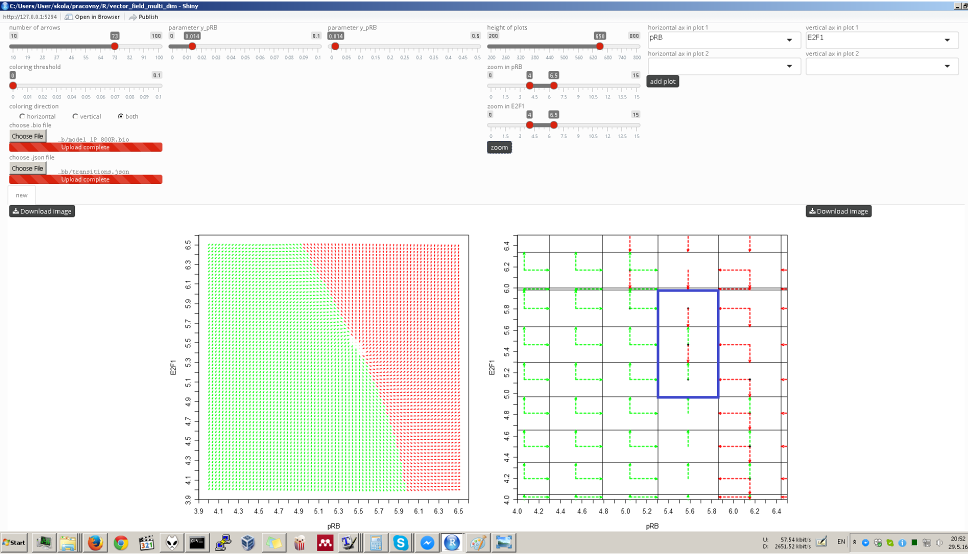
$\phi_{pRB} = 0.0115$ $[0.011, 0.0136]$

$\phi_{pRB} = 0.014$ $[0.0136, 0.5]$

bifurcation points: $\{0.011, 0.0136\}$

results agree with numerical methods up-to precision of approximation/discretisation

Coming Soon


C:/Users/User/skola/pracovny/R/vector_field_multi_dim - Shiny
http://127.0.0.1:5294
Open in Browser
Publish
number of arrows
parameter y_pRB
parameter y_pRB
height of plots
horizontal ax in plot 1
pRB
vertical ax in plot 1
E2F1
horizontal ax in plot 2
vertical ax in plot 2
add plot
coloring threshold
zoom in pRB
zoom in E2F1
zoom
coloring direction
horizontal
vertical
both
choose .bio file
Choose File
b/model_1P_800R.bio
Upload complete
choose .json file
Choose File
bb/transitions.json
Upload complete
new
Download image
Download image
E2F1
pRB
E2F1
pRB
Start

Conclusions

- exploiting dynamical systems under parameter uncertainty by model checking
- scalability achieved for number of parameters
- remaining challenges:
  - approximation: explore errors, what can be guaranteed?
  - abstraction: narrow the extent of overapproximation – can Darboux polynomials or barrier certificates help?
  - improve scalability w.r.t. systems dimensionality
  - is it possible to combine model checking with simulation? [TCSB XIV, 2012]

Other Contributions

Those worth mentioning

- specific for deterministic models:
  - STL* – value-freezing logic, monitoring, parameter exploration and robustness analysis

L. Brim, P. Dluhoš, D. Šafránek, T. Vejpustek. STL*: Extending signal temporal logic with signal-value freezing operator. Information and Computation, Volume 236, pp. 52-67, 2014

- specific for stochastic models:
  - parametric uniformisation for CTMC and CSL

L Brim, M Ceska, S Drazan, and D.Safranek. Exploring parameter space of stochastic biochemical systems using quantitative model checking. In CAV 2013. Lecture Notes in Computer Science, Volume 8044, 2013, pp 107-123.

M. Ceska, D. Safranek, S. Drazan, L. Brim: Robustness Analysis of Stochastic Biochemical Systems, (2014) Robustness Analysis of Stochastic Biochemical Systems. PLoS ONE 9(4): e94553.

- universally applicable methods:
  - formal language for biochemical space [SASB 2014, 2015]
  - parameter synthesis by coloured model checking
    $\rightarrow$ applied to boolean networks [CMSB 2012, 2013]
    $\rightarrow$ applied to qualitative abstractions of ODE models [HSB 2015, CMSB 2015, this talk]