Parameter synthesis in dynamical systems by model checking

David Šafránek

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

keynote talk
SynCoP + PV 2017





Masaryk University
Czech Republic

Outline

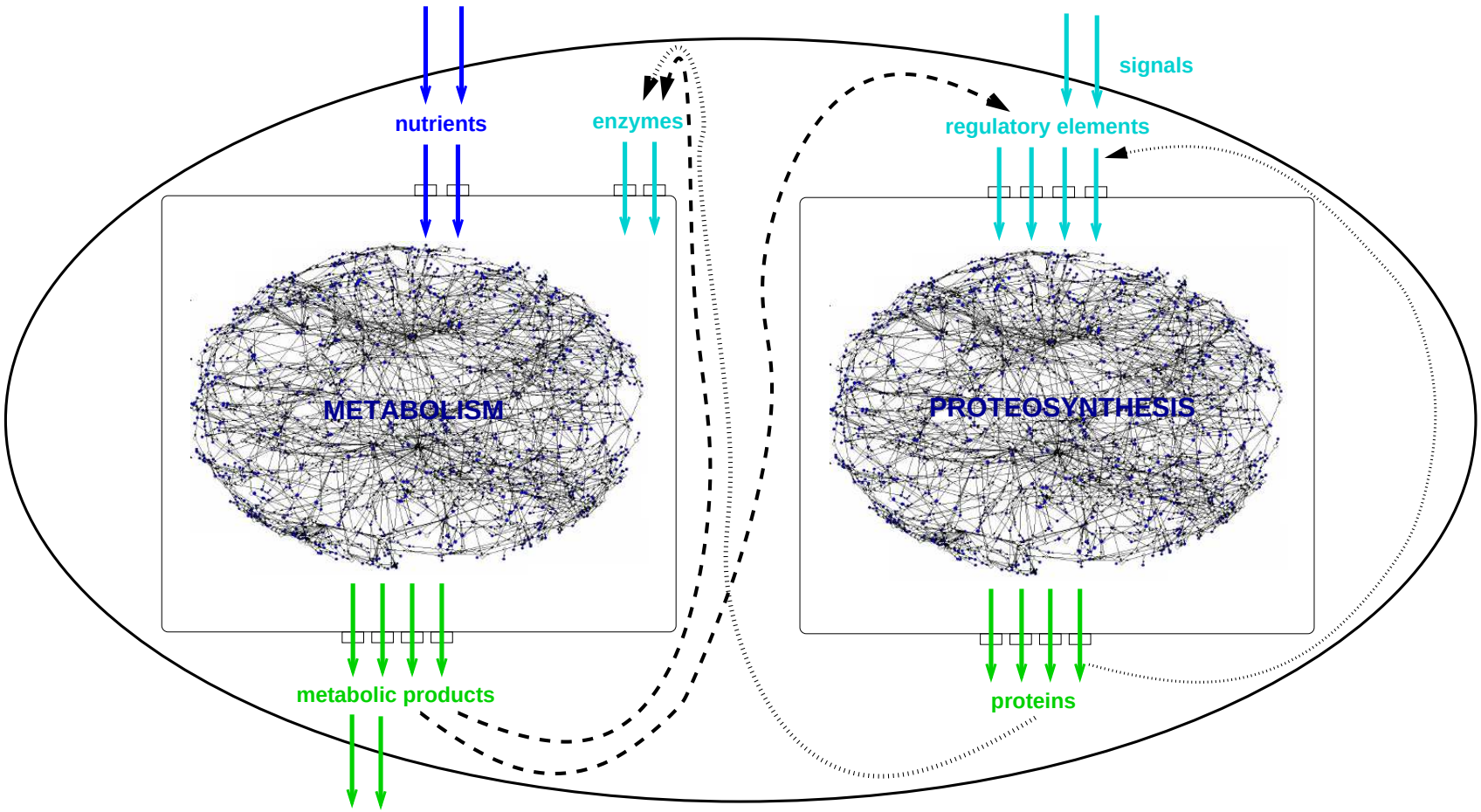
1. Motivation
2. Parameter Synthesis by Coloured Model Checking
3. Discrete Bifurcation Analysis
4. Case Studies
   - Case Study using Parameter Synthesis
   - Case Study using Discrete Bifurcation Analysis
5. Discussion

Outline

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

Motivation: Systems View of Processes Driving the Cell

To Mechanistically Understand Natural/Physical Systems



Motivation: Models of Complex Dynamical Systems
Understanding Role of Parameters

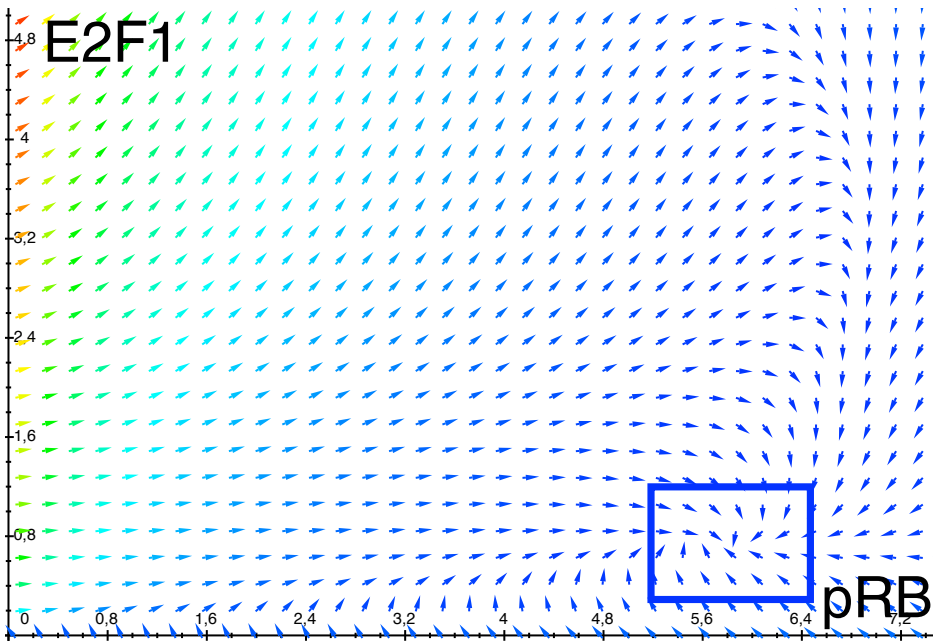
- deterministic models of dynamical systems:

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

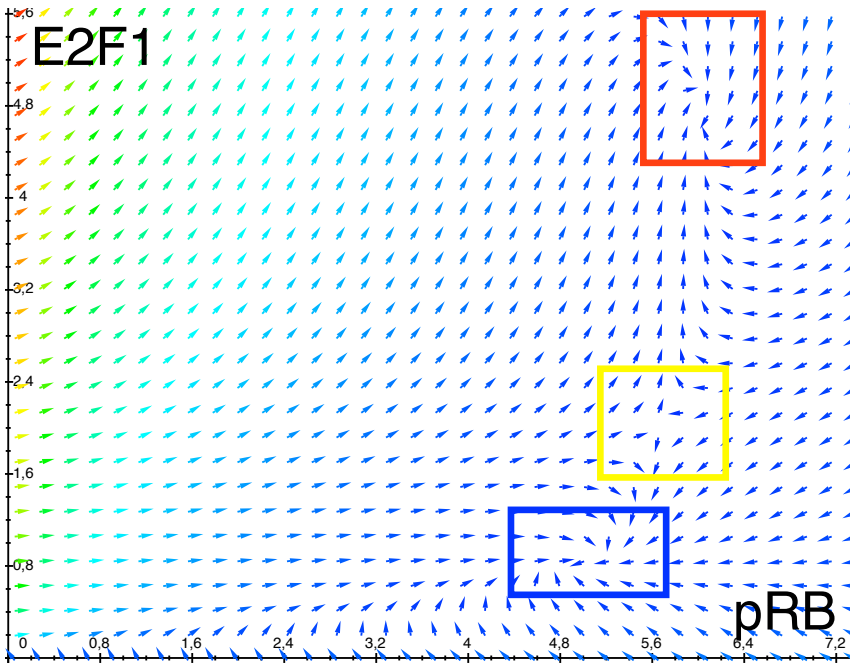
f ... phase space (vector field), $f : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$

x ... state vector ($\mathbb{R}^n$)

p ... parameter vector ($\mathbb{R}^m$)

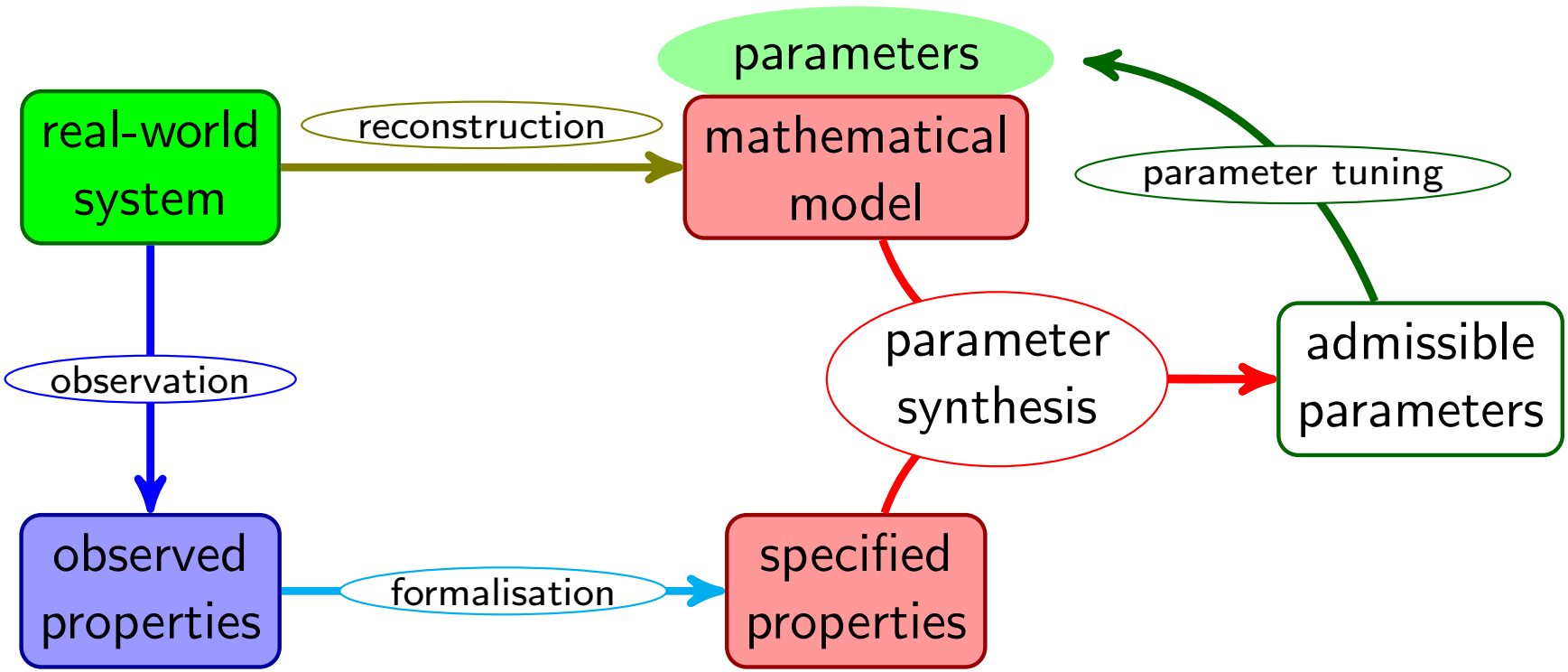


$p = 0.006$

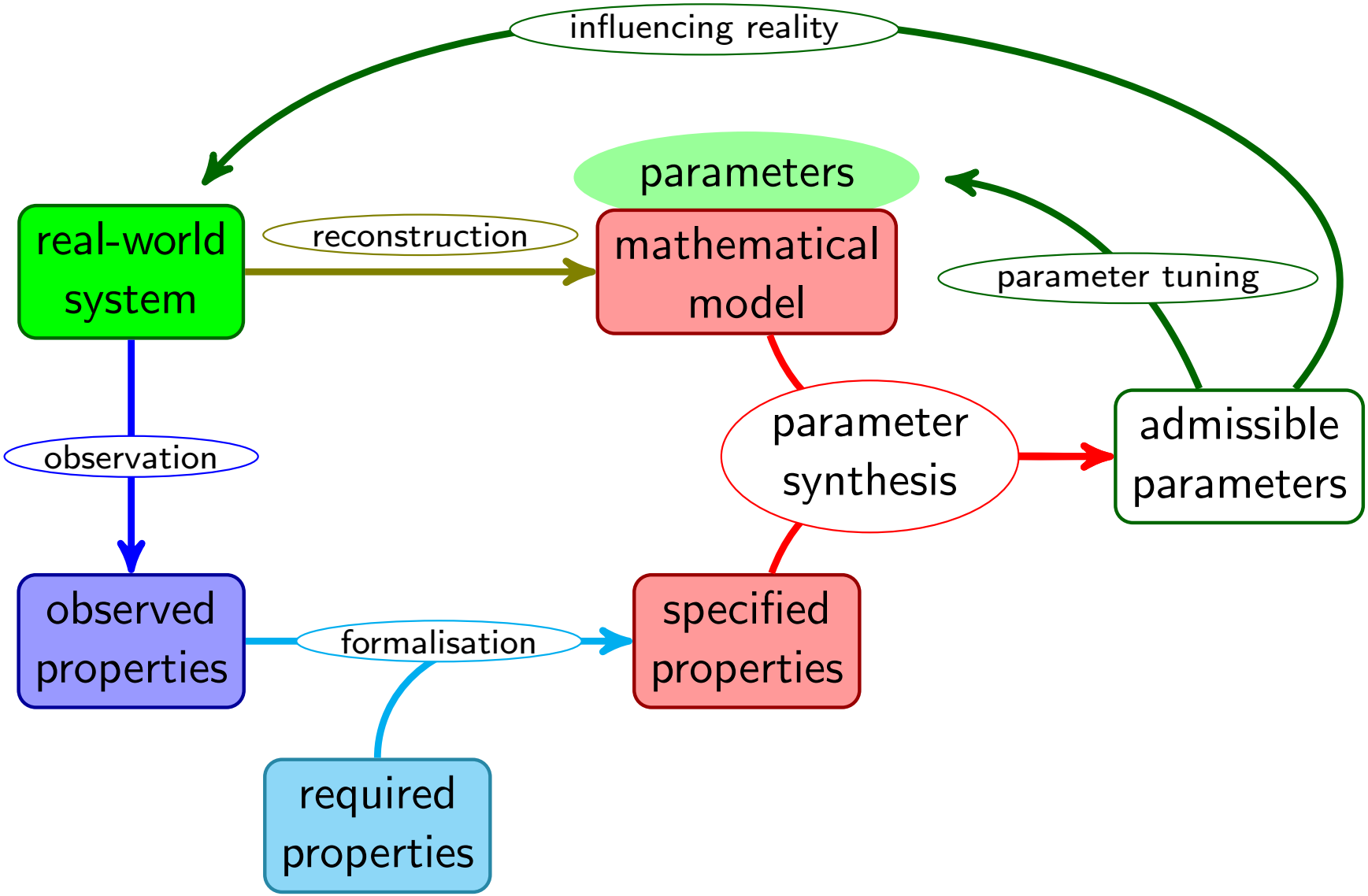


$p = 0.012$

Motivation: Dynamical Systems with Parameters



Motivation: Dynamical Systems with Parameters



- *(bio)physics*: often use **parameterised** non-linear ODEs, supplied with **local** methods (e.g., 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

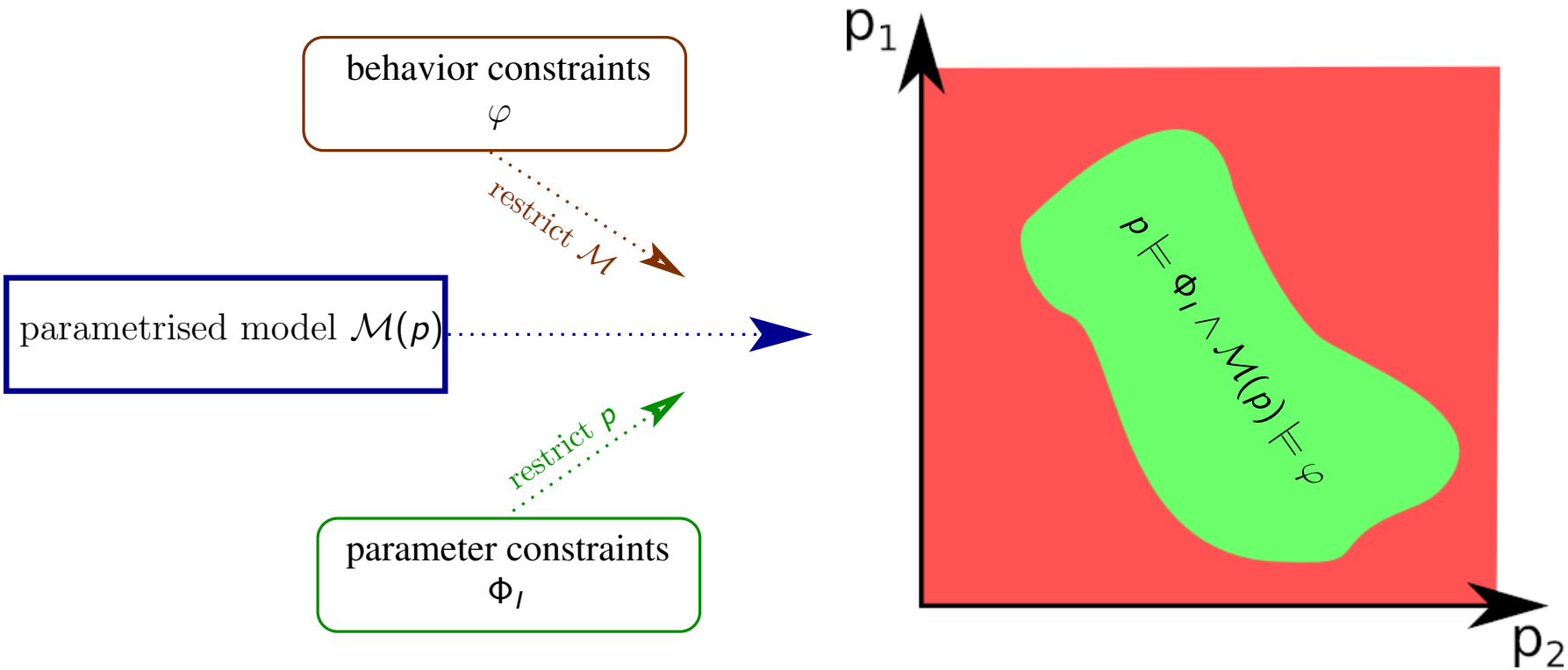


Outline

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

Problem Formulation

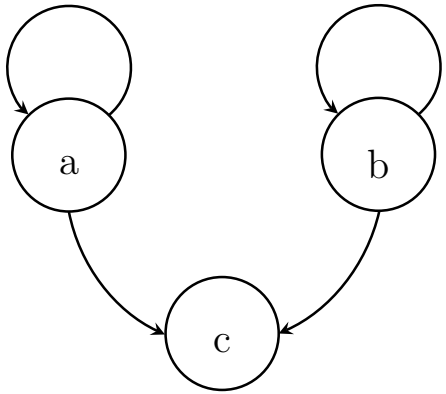
Parameter Synthesis for Dynamical Systems



Parameter Synthesis Problem

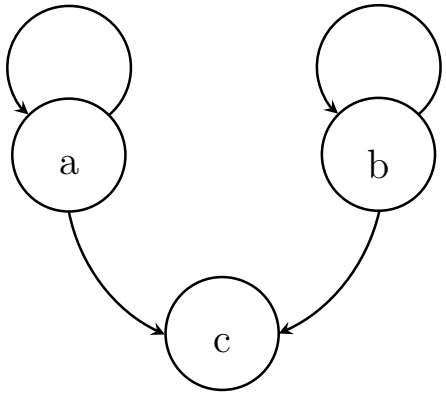
Assume $\mathcal{P}$ is the admissible **parameter space**. Given a *behaviour constraint* φ, *parameter constraint* Φ_I, and a *parameterised model* $\mathcal{M}$, **find the maximal set** $P \subseteq \mathcal{P}$ **of parameterisations** such that $p \models \Phi_I$ and $\mathcal{M}(p) \models \varphi$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

Considered Parameterised Models of Dynamical Systems



- dynamical systems as regulatory networks
- discrete models (e.g., Boolean networks)
  - parameters determine the logic of interactions
  - deterministic (synchronous) vs. non-deterministic (asynchronous)
  - finite-state, finite parameters, parameter explosion dominates
  - TCCB 2010, CMSB 2012, SASB 2016
  - Henzinger et al. TACAS 2015; Ballarini et al. 2016

Considered Parameterised Models of Dynamical Systems



- dynamical systems as regulatory networks
- discrete models (e.g., Boolean networks)
  - parameters determine the logic of interactions
  - deterministic (synchronous) vs. non-deterministic (asynchronous)
  - finite-state, finite parameters, parameter explosion dominates
  - TCCB 2010, CMSB 2012, SASB 2016
  - Henzinger et al. TACAS 2015; Ballarini et al. 2016
- non-linear ODE models
  - continuous parameters determine the vector field
  - undecidable
  - semi-decidable by employing approximations/abstractions
    - piece-wise affine and piece-wise multi-affine
    - overapproximation by (parametrised) finite automata
    - state explosion and parameter explosion

Work Chronology

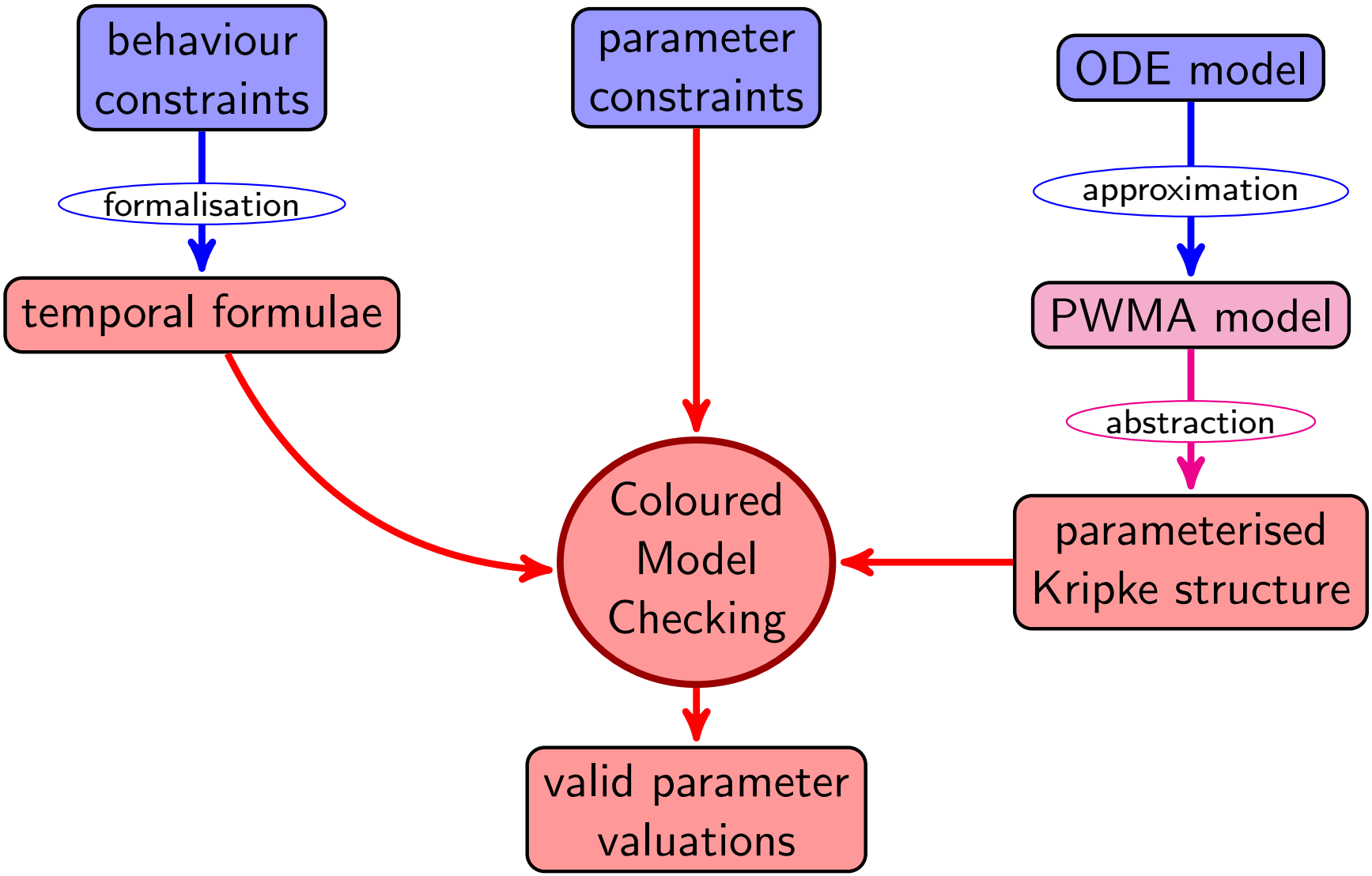
Related Work

- Batt et al. 2007: *RoverGene*, BDD/Polytopes-based approach
- de Jong 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

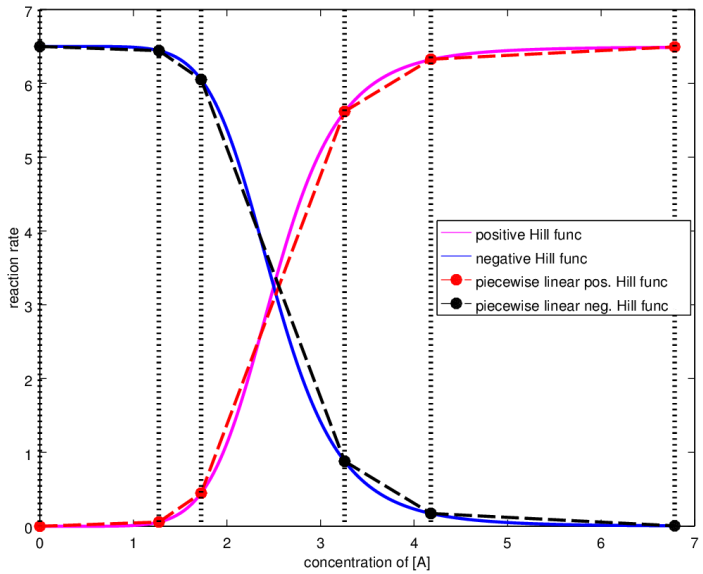
- HIBI 2010, TCCB 2012: coloured LTL model checking, piecewise multi-affine, parallel algorithm
- CMSB 2015: coloured CTL model checking, piecewise multi-affine, parallel algorithm, independent parameters
- HSB 2015, FM 2016: discrete bifurcation analysis by coloured HUCTL model checking
- ATVA 2016, CMSB 2016, CAV 2017: SMT employed, **interdependent parameters**, `Pithya` tool

Workflow


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

Step 1: Approximation
Discretisable Continuous (ODE) Models

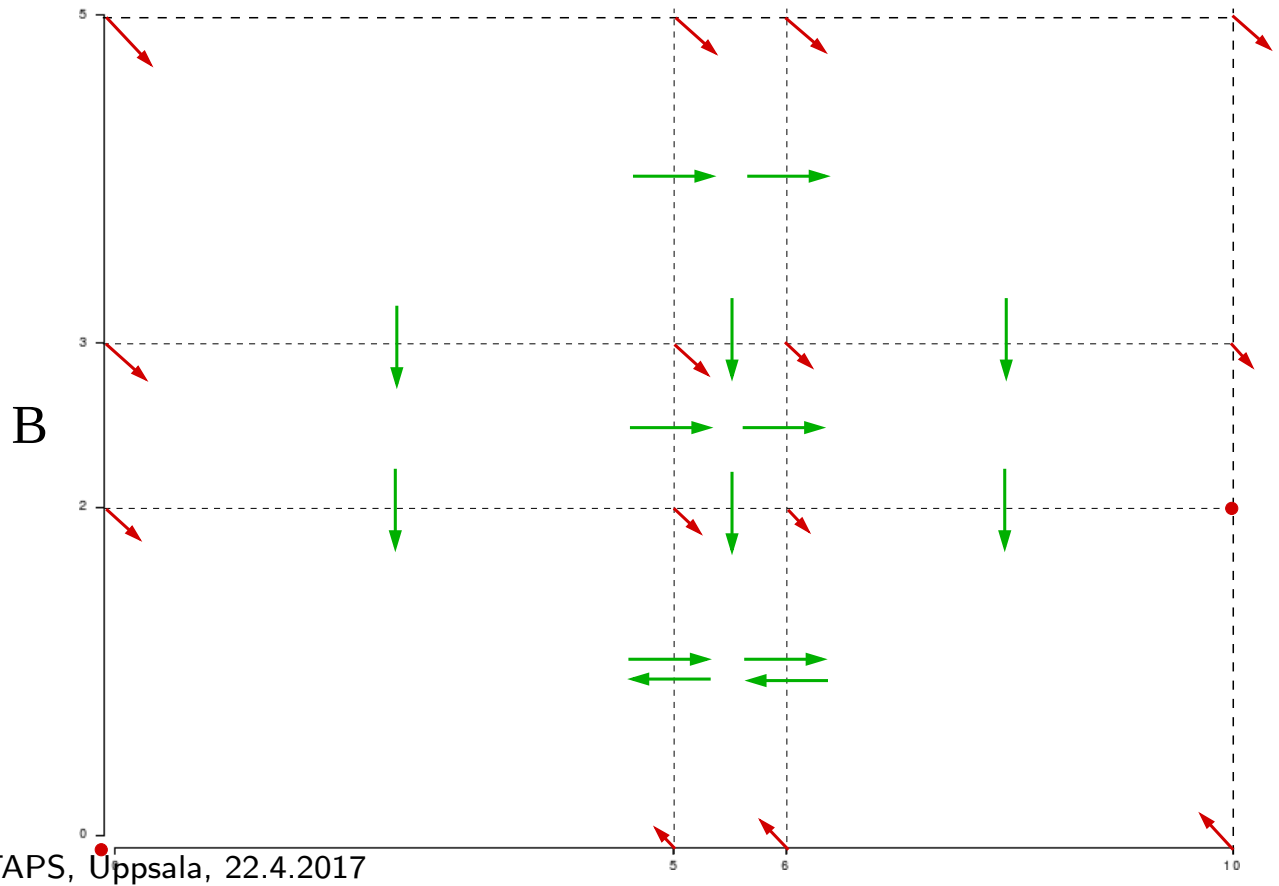
- (bio)physical mechanisms modeled by sigmoidal kinetics (e.g., signalling pathways, gene regulatory circuits, ...)
- optimal approximation of sigmoid functions by piece-wise affine functions (ramps) [Grosu et al. CAV 2011]



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 2006]
- continuous phase-space is partitioned into (hypher)rectangles
- **no diagonal transitions**, overapproximation
- **self-transitions**, in PWMA only necessary condition for exiting a box in finite time $\Longrightarrow$ overapproximation



Step 3: Parameter Synthesis – Main Idea

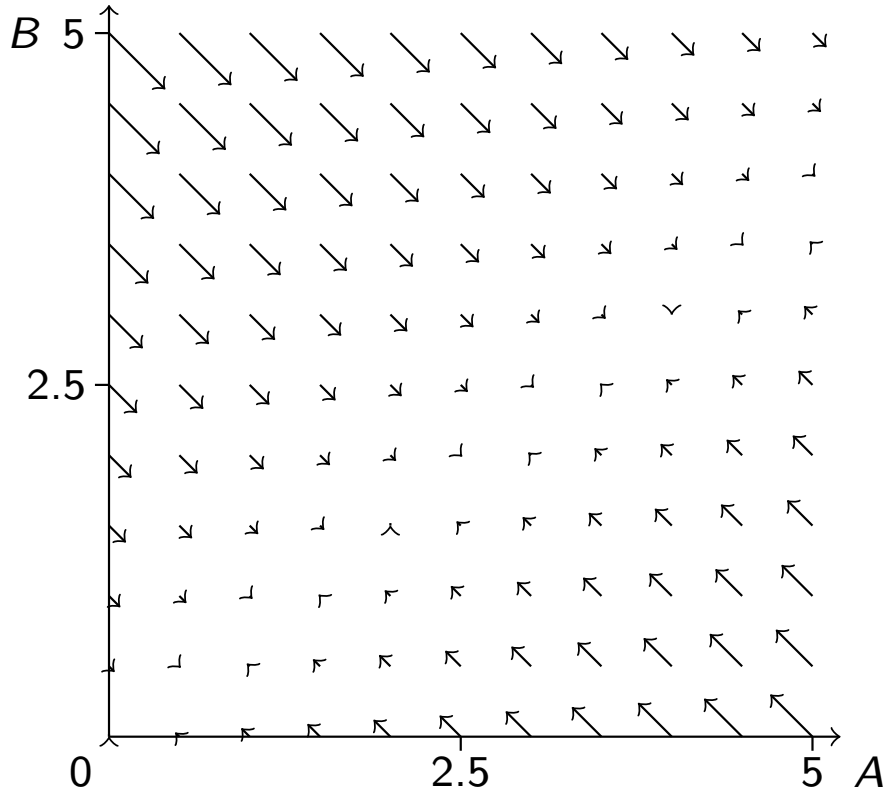
Phase Space Discretisation Leads to Parameter Space Discretisation

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

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

$$k_2 = 0.8$$

$$k_1 = 0.6$$



Step 3: Parameter Synthesis – Main Idea

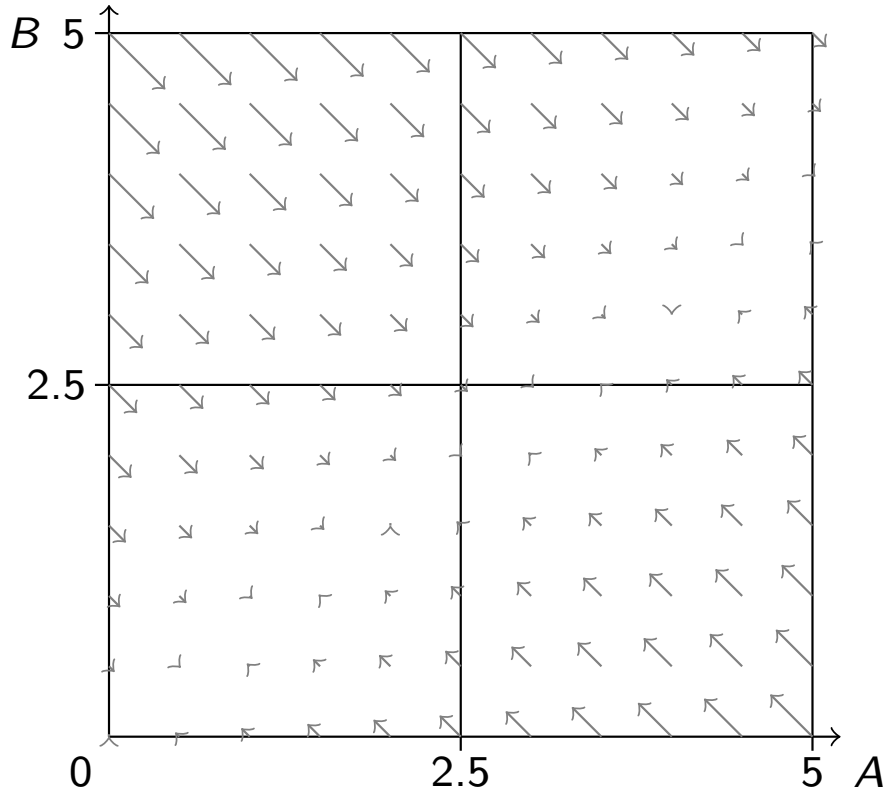
Phase Space Discretisation Leads to Parameter Space Discretisation

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

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

$$k_2 = 0.8$$

$$k_1 = 0.6$$



Step 3: Parameter Synthesis – Main Idea

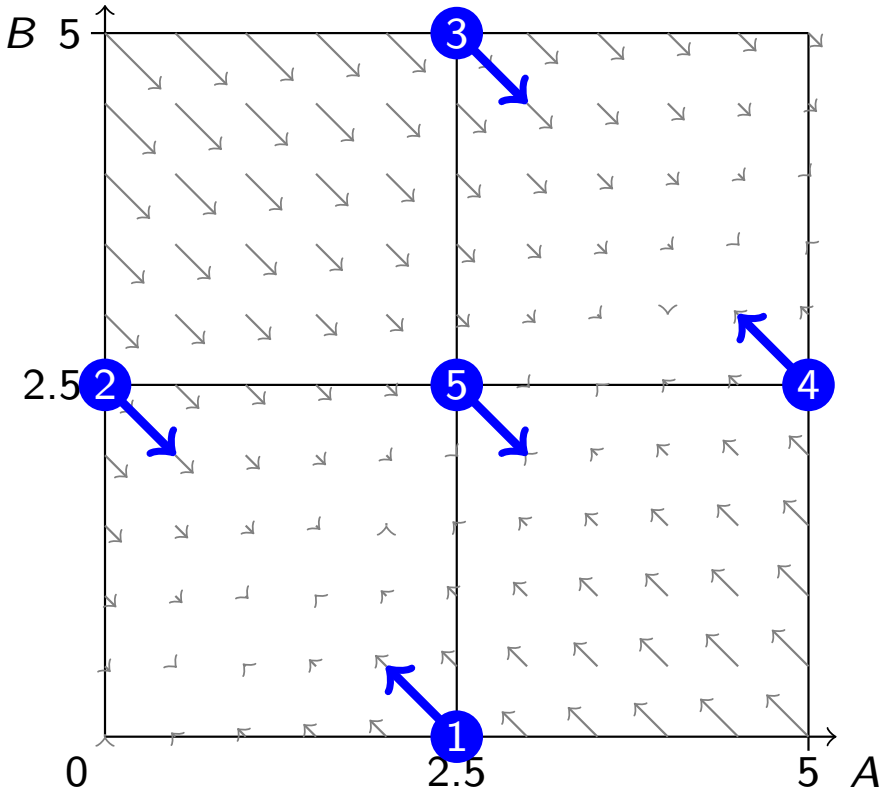
Phase Space Discretisation Leads to Parameter Space Discretisation

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

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

$$k_2 = 0.8$$

$$k_1 = 0.6$$



Step 3: Parameter Synthesis – Main Idea

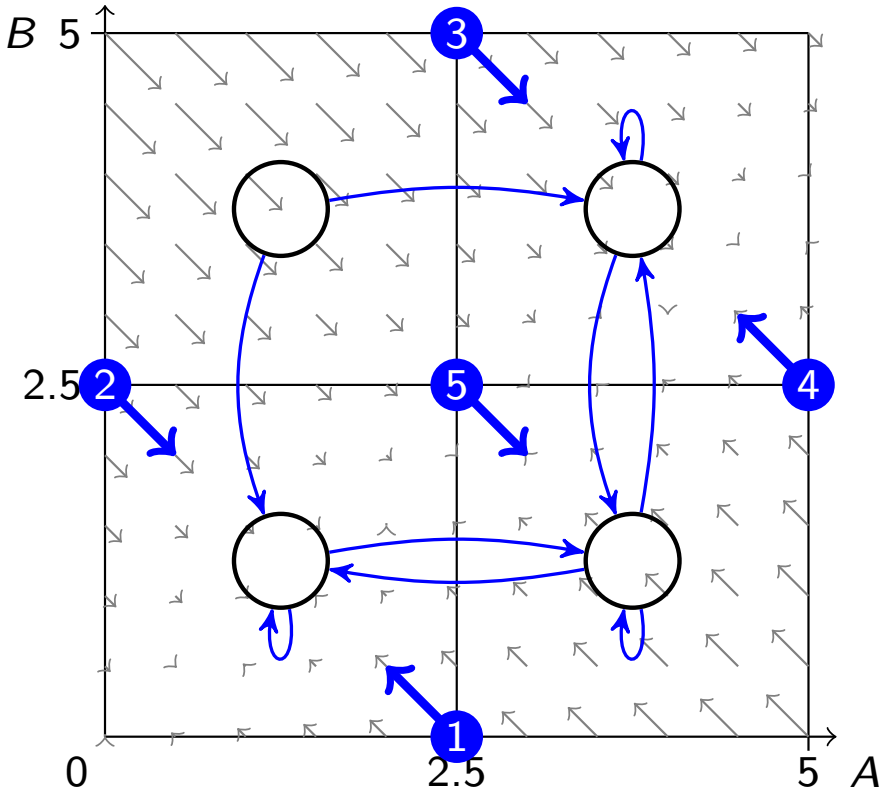
Phase Space Discretisation Leads to Parameter Space Discretisation

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

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

$$k_2 = 0.8$$

$$k_1 = 0.6$$



Step 3: Parameter Synthesis – Main Idea

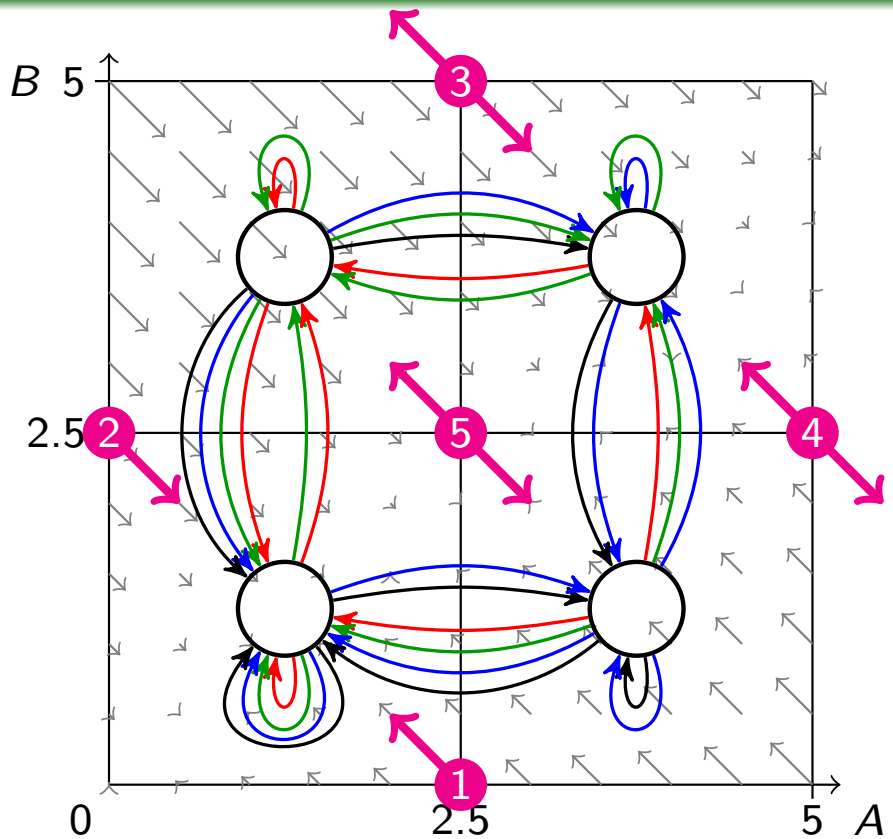
Phase Space Discretisation Leads to Parameter Space Discretisation

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

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

$$k_2 = 0.8$$

$$k_1 = ?$$



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

Step 3: Parameter Synthesis – Main Idea

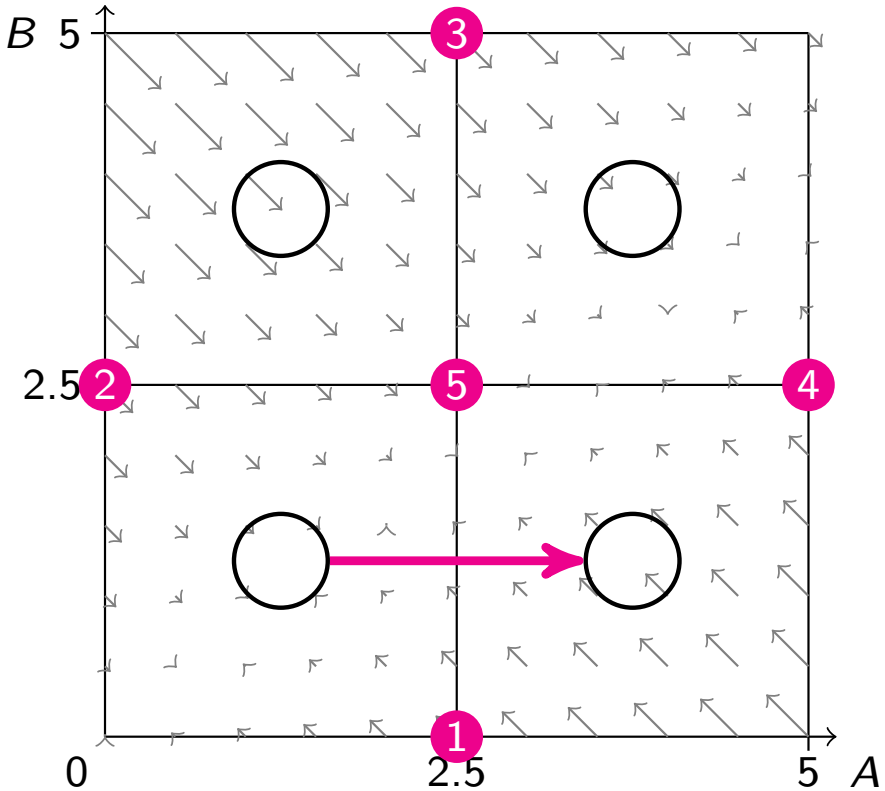
Phase Space Discretisation Leads to Parameter Space Discretisation

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

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

$$k_2 = 0.8$$

$$k_1 = ?$$



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

Step 3: Parameter Synthesis – Main Idea

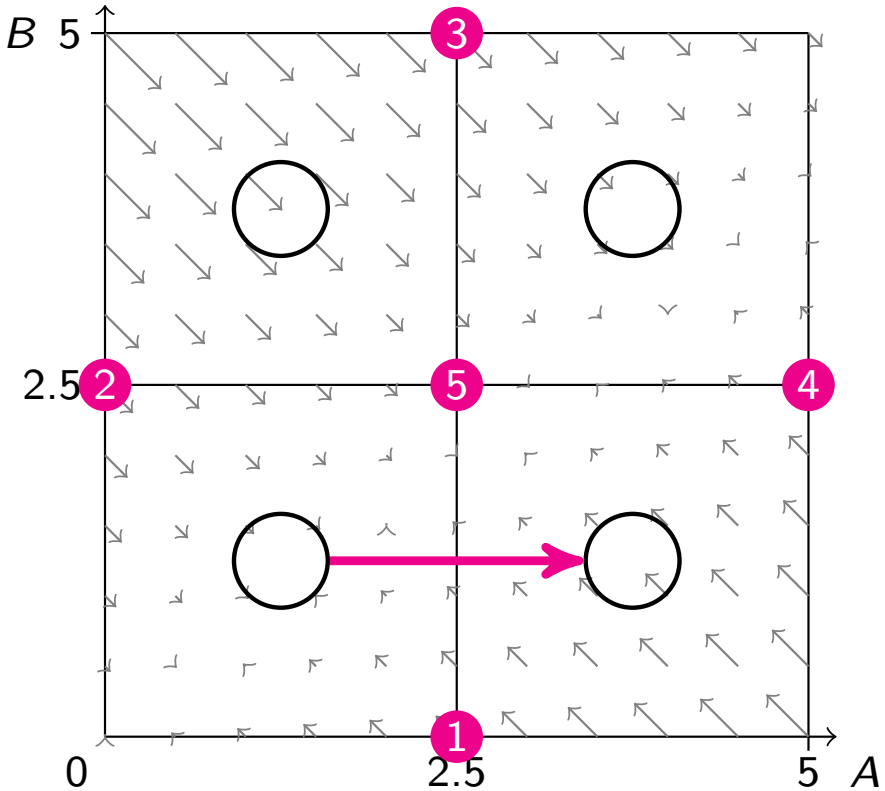
Phase Space Discretisation Leads to Parameter Space Discretisation

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

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

$$k_2 = 0.8$$

$$k_1 = ?$$



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

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

- necessary condition for escaping a state (rectangle) s in finite time is

$$\mathbf{0} \notin \mathit{hull}\{f(v,p) \mid v \in \mathit{vertices}(s)\}$$

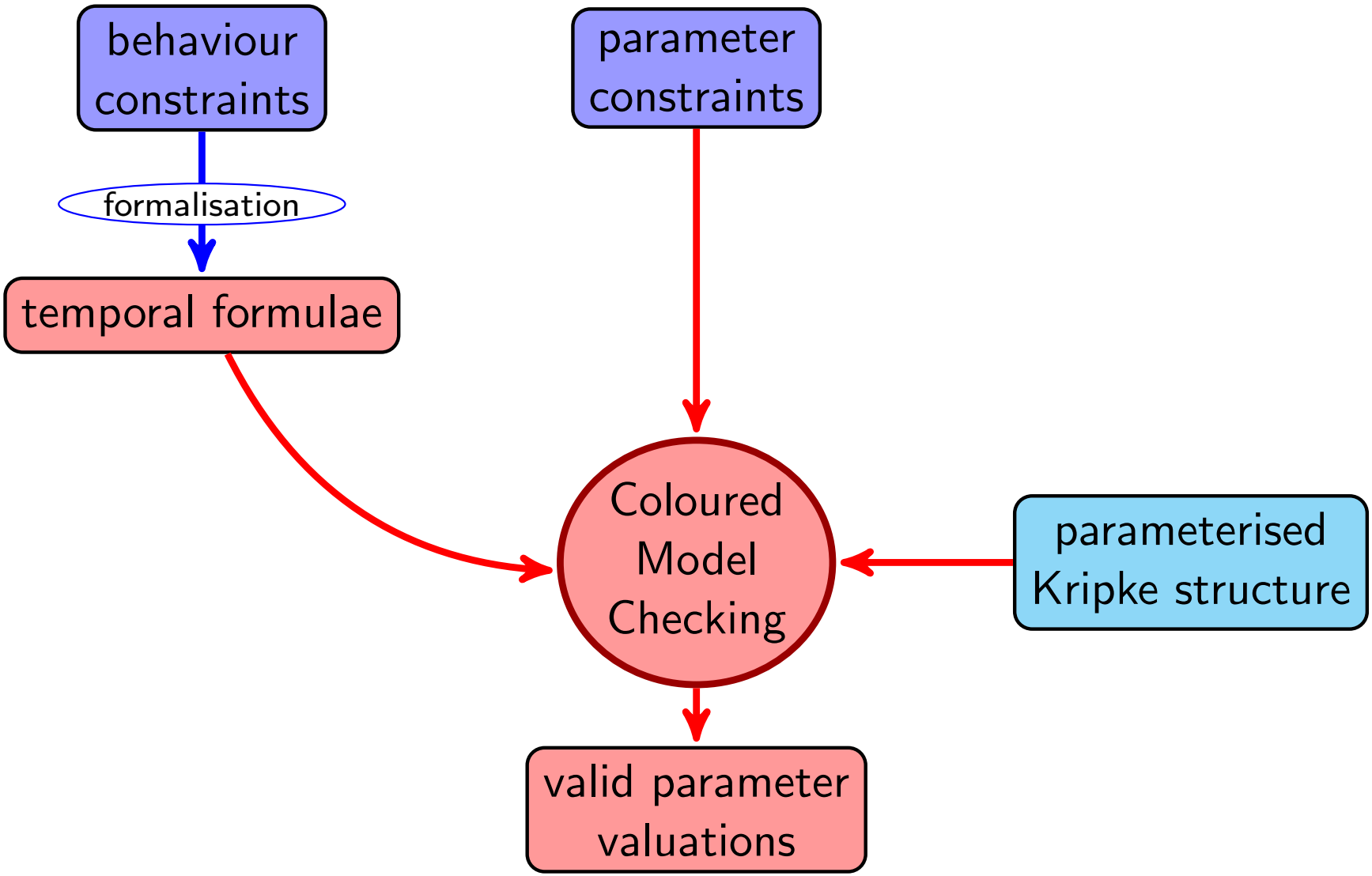
- for all $p \in \mathcal{P}$ not satisfying the necessary condition above a self-transition is added to s:

$$\Phi_{s,s} := \exists c_1, \ldots, c_k : \left(\bigwedge_{i=1}^{k} c_i \geq 0\right) \wedge \left(\sum_{i=1}^{k} c_i = 1\right) \wedge \left(\sum_{i=1}^{k} c_i \cdot f(v_i,p) = 0\right)$$

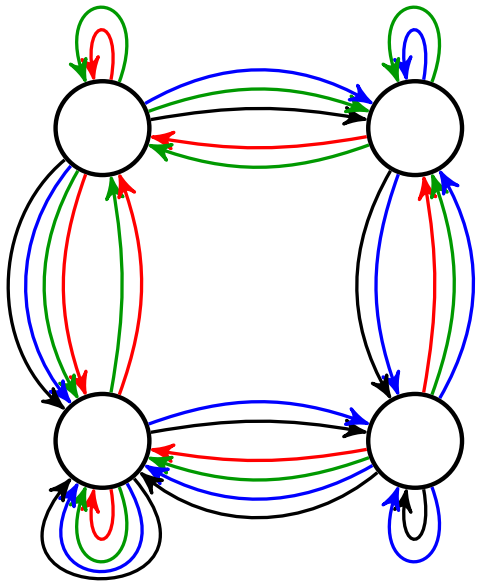
- quantifier-free formulae giving coarser overapproximation of self-transitions can be used alternatively

Workflow

Parameterised Kripke Structures and Model Checking

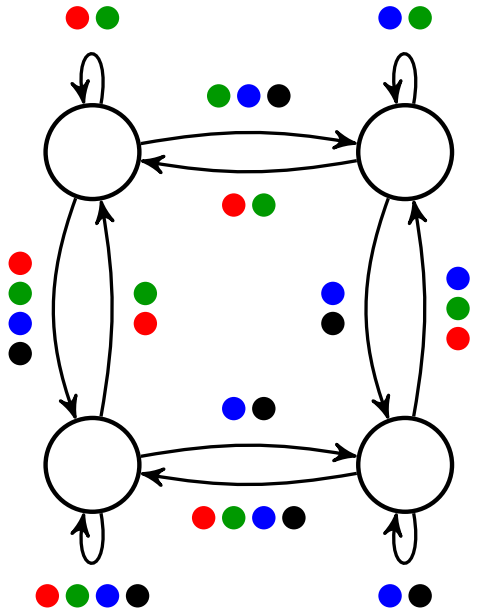


Transitions with Parameters (coloured transitions)



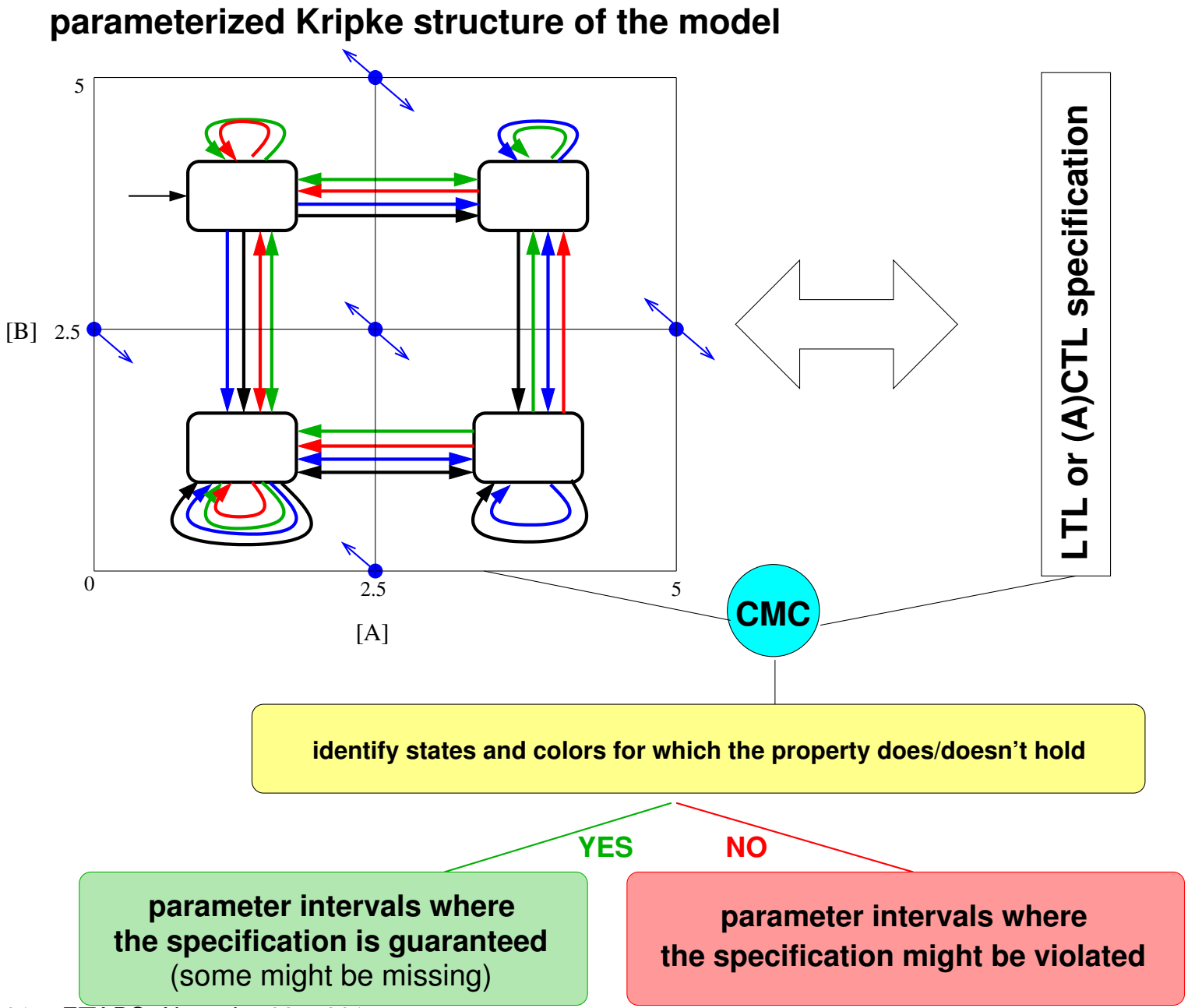
- each parameter valuation represents one Kripke structure
- shared state space, different transition space

Transitions with Parameters (coloured transitions)



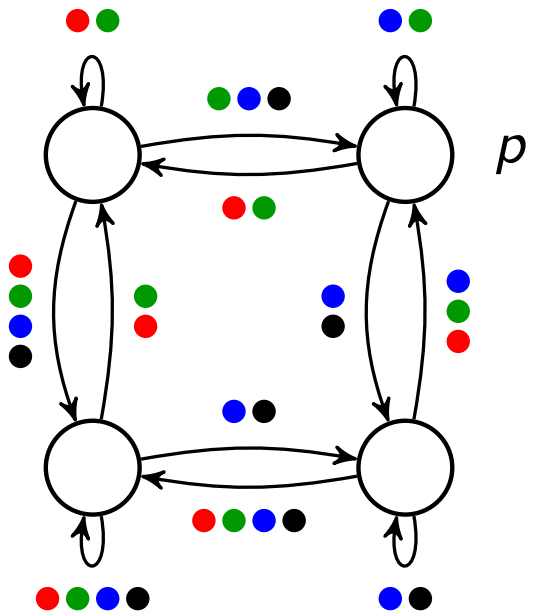
- each parameter valuation represents one Kripke structure
- shared state space, different transition space
- symbolic representation of parameters
- symbolic PKS: every transition is associated with a formula

Parameter Synthesis by Coloured Model Checking



- exloiting experiences with enumerative MC
- many parameter values typically shared on a transition
  - coloured MC is an effective heuristics wrt naïve approach
- parallel automata-based LTL MC [`BioDiVinE` – TCBB 2010]
  - LTL nicely preserved by the abstractions
  - abstractions are non-deterministic
  - need to analyse decision states (neighbourhood around unstable equillibria)
  - we went more to monitoring/robustness analysis STL, STL* (extending work of Donze et al.)
- towards coloured CTL model checking [`Pithya` – CAV 2017]

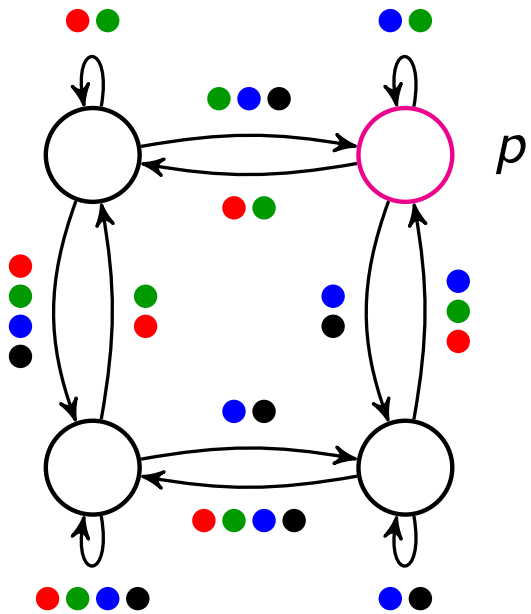
- enumerative approach
- CTL model checking: dynamic programming, back propagation
- coloured CTL model checking: back propagation of colours



CTL formula: **EF** p

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

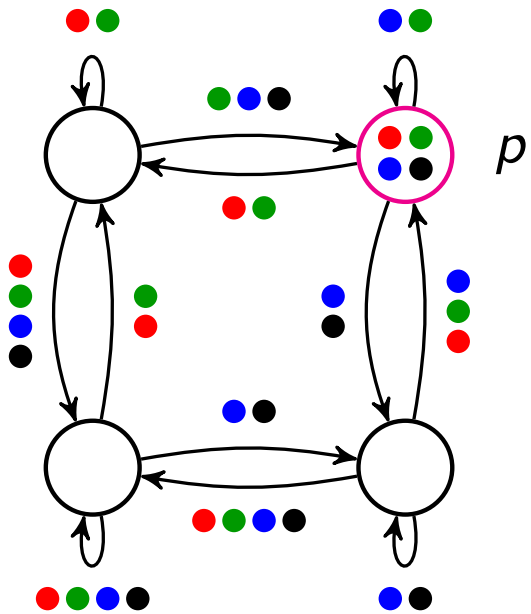
- enumerative approach
- CTL model checking: dynamic programming, back propagation
- coloured CTL model checking: back propagation of colours



CTL formula: **EF** p

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

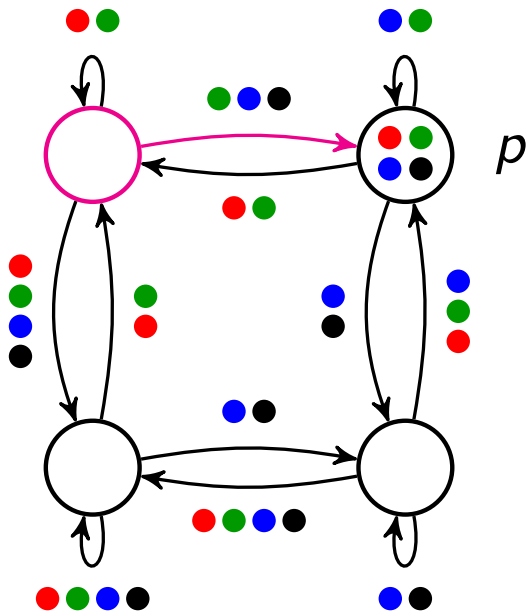
- enumerative approach
- CTL model checking: dynamic programming, back propagation
- coloured CTL model checking: back propagation of colours



CTL formula: **EF** p

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

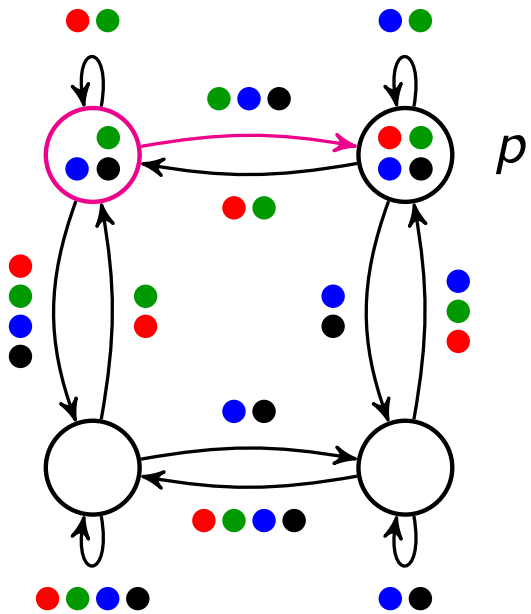
- enumerative approach
- CTL model checking: dynamic programming, back propagation
- coloured CTL model checking: back propagation of colours



CTL formula: **EF** p

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

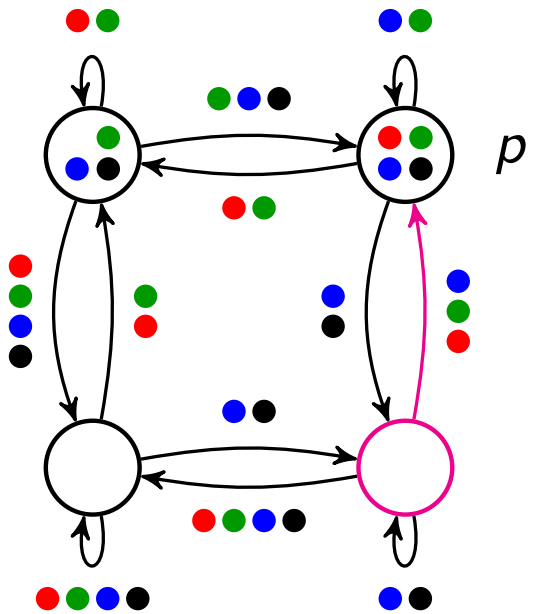
- enumerative approach
- CTL model checking: dynamic programming, back propagation
- coloured CTL model checking: back propagation of colours



CTL formula: **EF** p

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

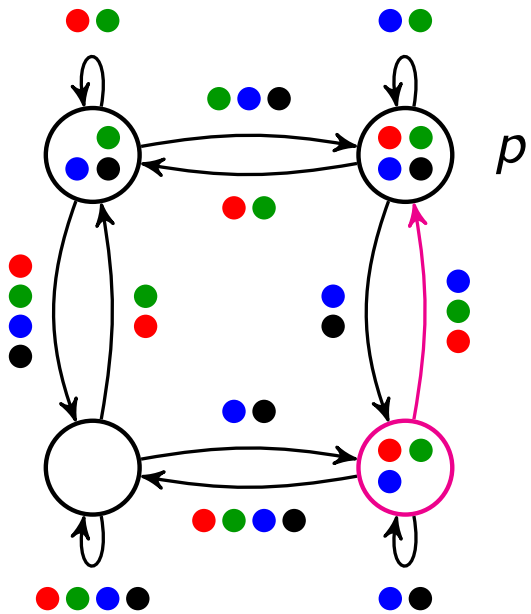
- enumerative approach
- CTL model checking: dynamic programming, back propagation
- coloured CTL model checking: back propagation of colours



CTL formula: **EF** p

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

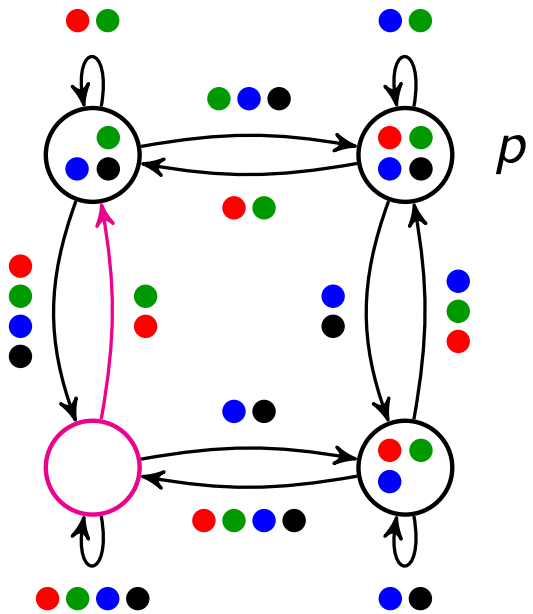
- enumerative approach
- CTL model checking: dynamic programming, back propagation
- coloured CTL model checking: back propagation of colours



CTL formula: **EF** p

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

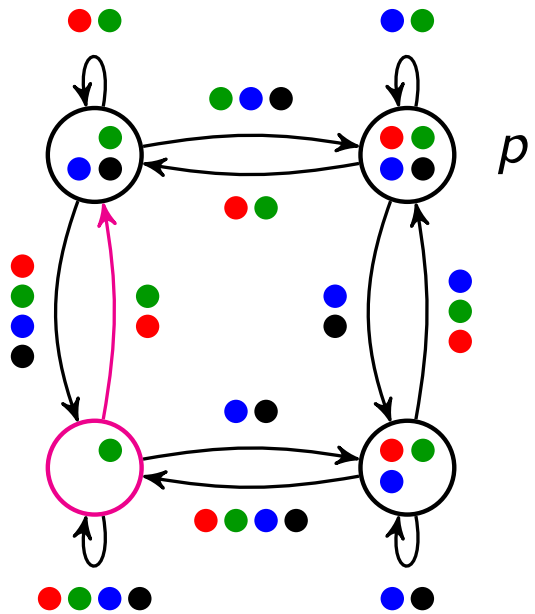
- enumerative approach
- CTL model checking: dynamic programming, back propagation
- coloured CTL model checking: back propagation of colours



CTL formula: **EF** p

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

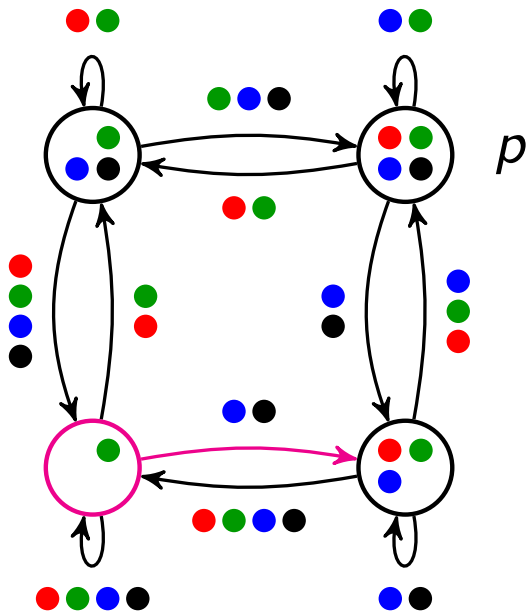
- enumerative approach
- CTL model checking: dynamic programming, back propagation
- coloured CTL model checking: back propagation of colours



CTL formula: **EF** p

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

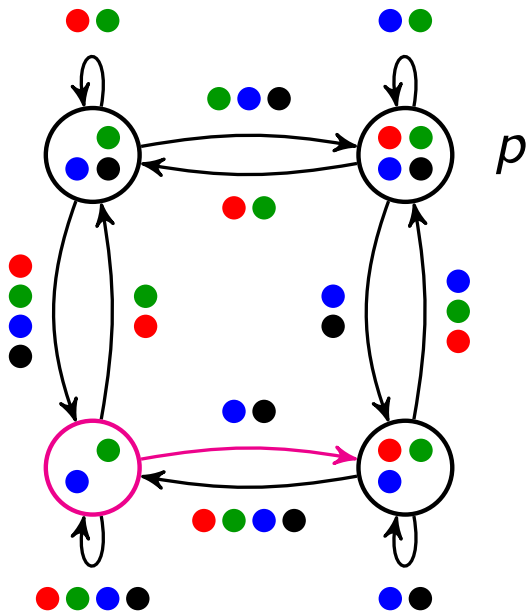
- enumerative approach
- CTL model checking: dynamic programming, back propagation
- coloured CTL model checking: back propagation of colours



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

Parameter Synthesis wrt CTL Specification

Let $\mathcal{K} = (\mathcal{P}, S, I, \rightarrow, L)$ be a **parameterised Kripke structure** over AP. Let Φ_I be an **initial parameter constraint**. Let further φ be a **behaviour constraint** defined as a CTL formula over AP. The *parameter synthesis problem* is to find the function $\mathcal{F} : S \rightarrow 2^{\mathcal{P}}$:

$$\mathcal{F}(s) = \{p \in \mathcal{P} \mid p \models \Phi_I, s \models_{\mathcal{K}_p} \varphi\}$$

where $\mathcal{K}_p$ is a concrete (un-parametrised) Kripke structure for a parameter value $p \in \mathcal{P}$.

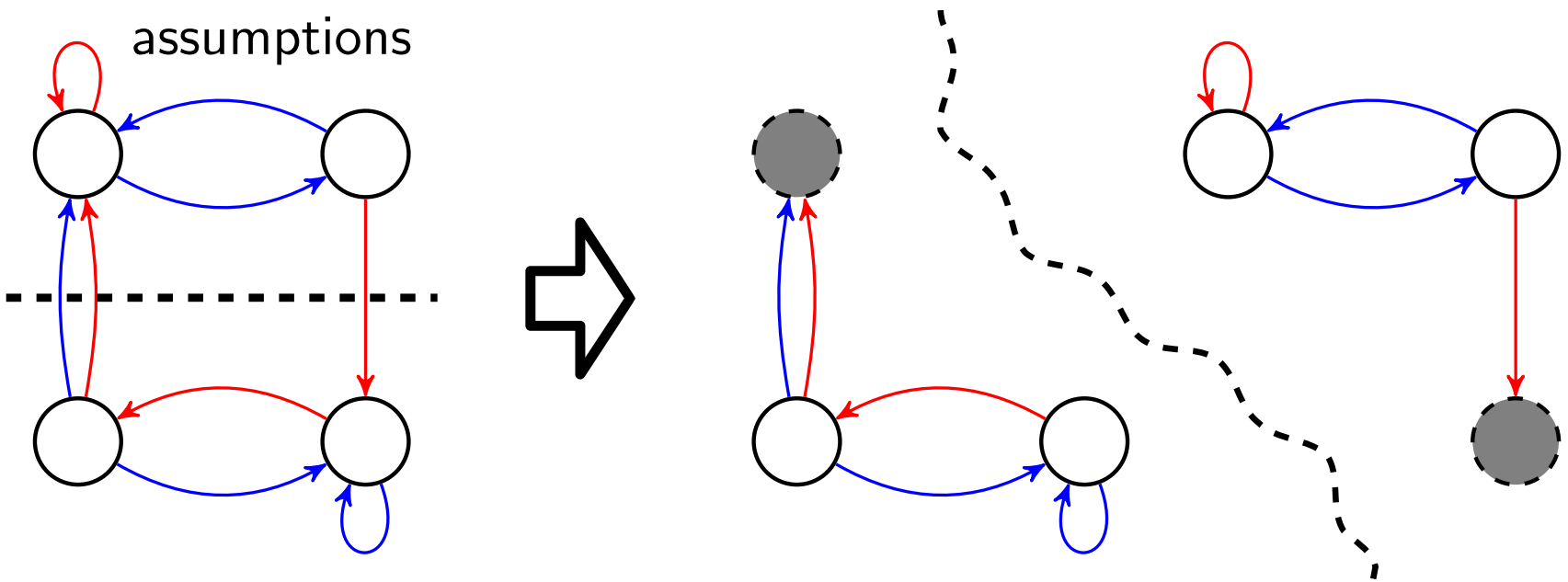
Encoding

- 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
- simplification: if parameters are independent we can work directly with intervals (symbolic sets) instead of SMT

Expressiveness: anything an SMT solver can handle (we employ linear arithmetics over reals)

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



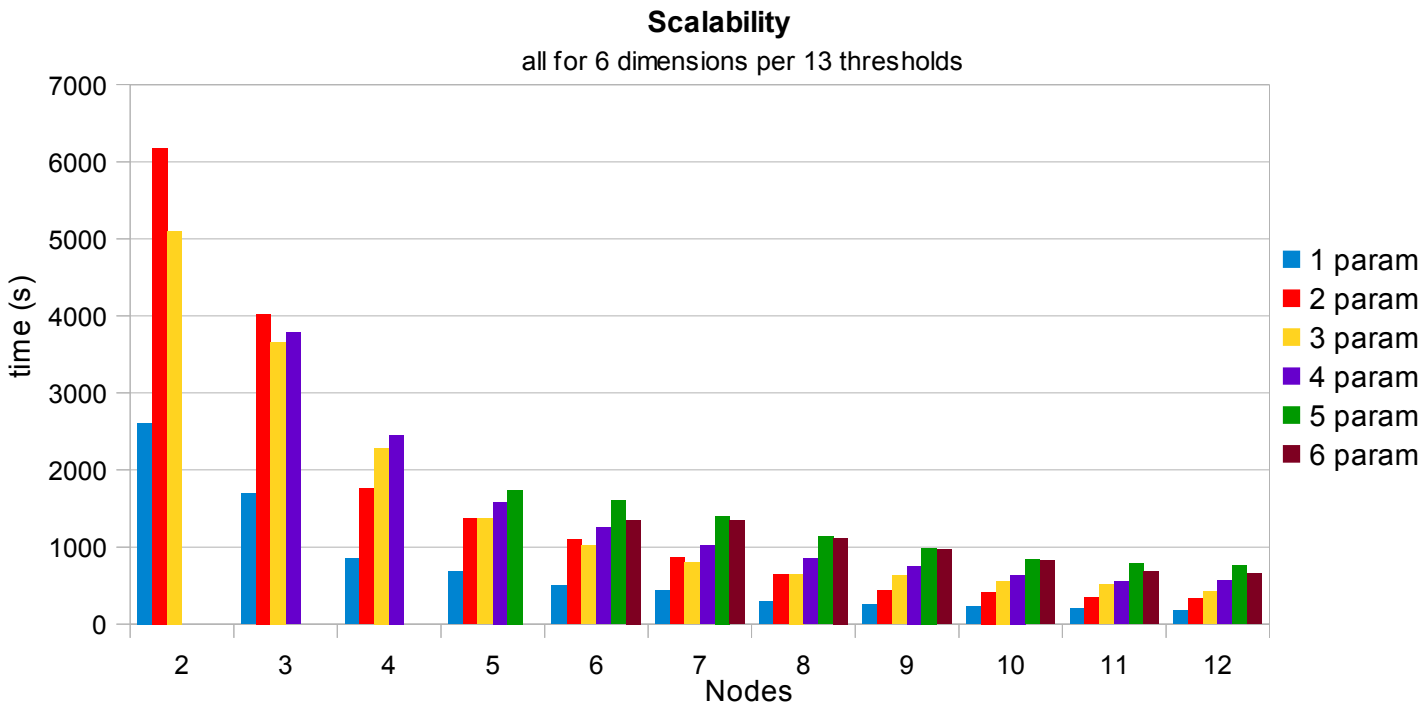
Idea based on:

Brim, Yorav, Žídková 2005: parallel CTL model checking

Performance and Scalability
Enzymatic Chain ODE as a Benchmark

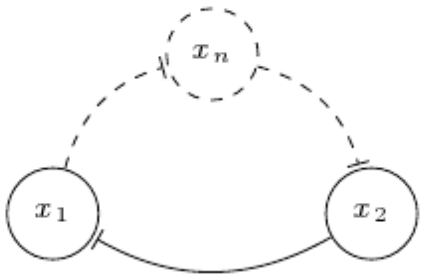
$$S + E \rightleftharpoons ES_1 \rightleftharpoons \cdots \rightleftharpoons ES_k \rightleftharpoons P + E$$

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



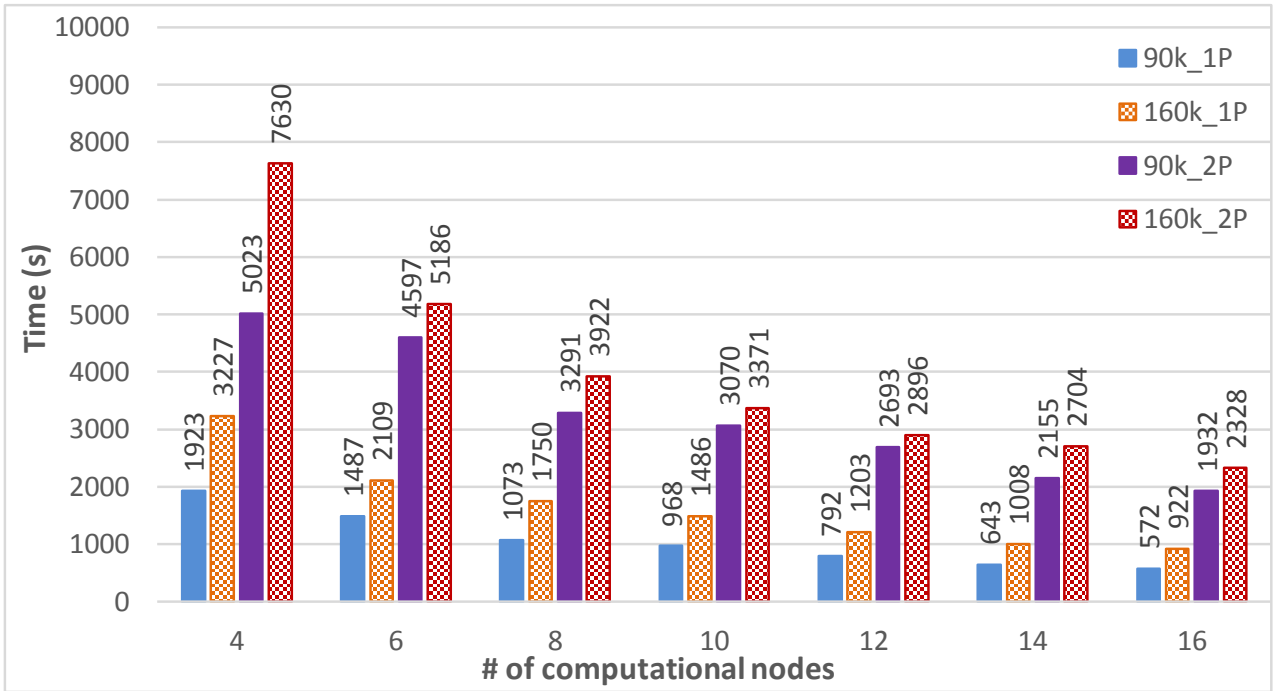
quad-core 2GHz Intel Xeon CPUs, 1 core used per node, without SMT

Performance and Scalability
Repressilator ODE



$$\frac{d[x_1]}{dt} = k_1 \frac{K_1^5}{K_1^5 + [x_2]^5} - \phi_{x_1}[x_1]$$
$$\frac{d[x_2]}{dt} = k_2 \frac{K_2^5}{K_2^5 + [x_n]^5} - \phi_{x_2}[x_2]$$
$$\frac{d[x_n]}{dt} = k_n \frac{K_n^5}{K_n^5 + [x_1]^5} - \phi_{x_n}[x_n]$$

$k_i = 1$, $K_i = 5$ for all $i \in \{1, \ldots, n\}$, $n = 6$
$\phi_{x_j}, \phi_{x_1} \in (0, 1)$ considered unknown



quad-core 2GHz Intel Xeon CPUs, 1 core used per node, with SMT

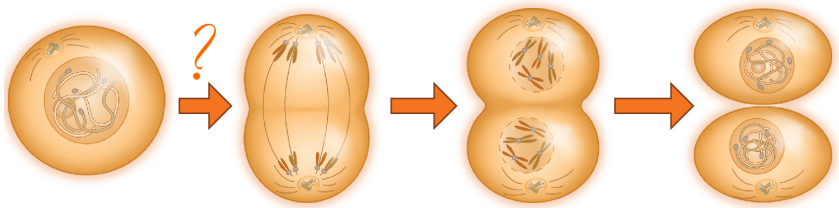
Outline

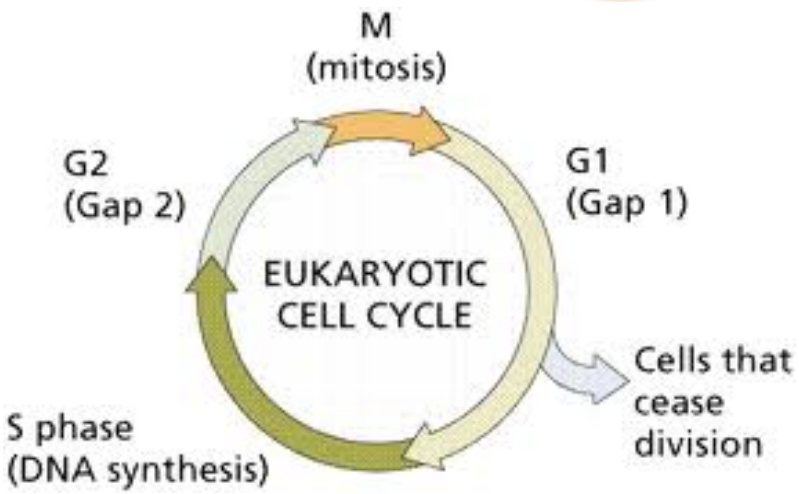
1. Motivation
2. Parameter Synthesis by Coloured Model Checking
3. Discrete Bifurcation Analysis
4. Case Studies
   - Case Study using Parameter Synthesis
   - Case Study using Discrete Bifurcation Analysis
5. Discussion

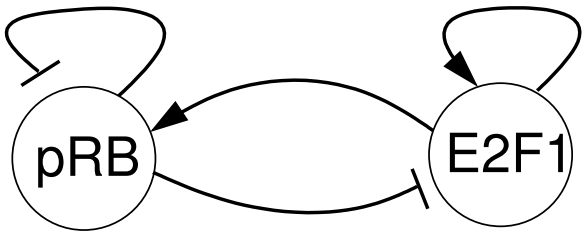
Motivation: How Parameters Influence Systems Dynamics?

Example: decision making in living cells

— *to divide or not to divide?*





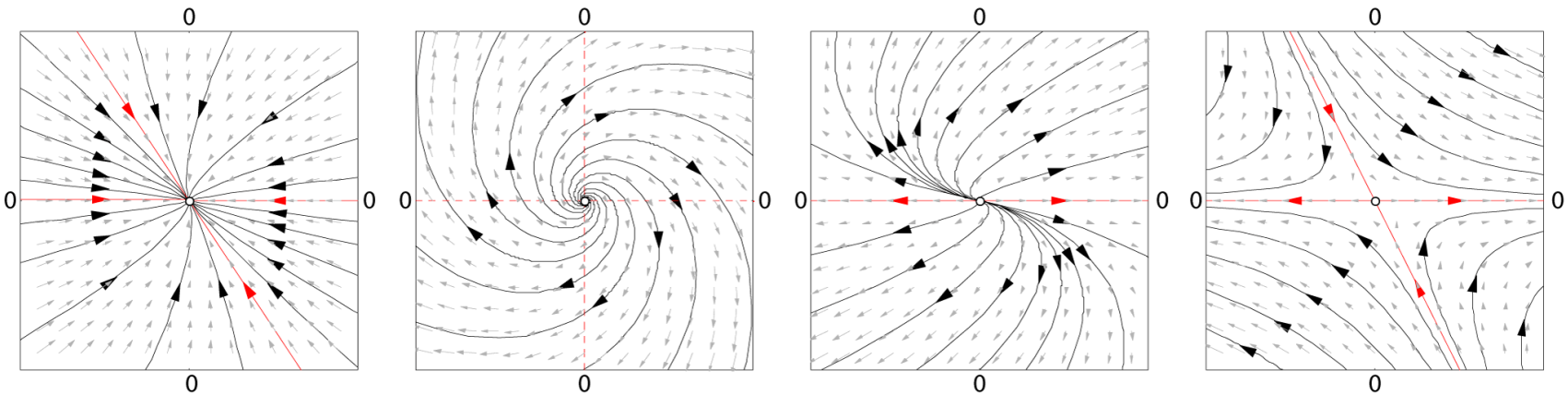


[Swat et al. 2004]

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

Motivation: How Parameters Influence Systems Dynamics?
Bifurcation Analysis of Dynamical Systems

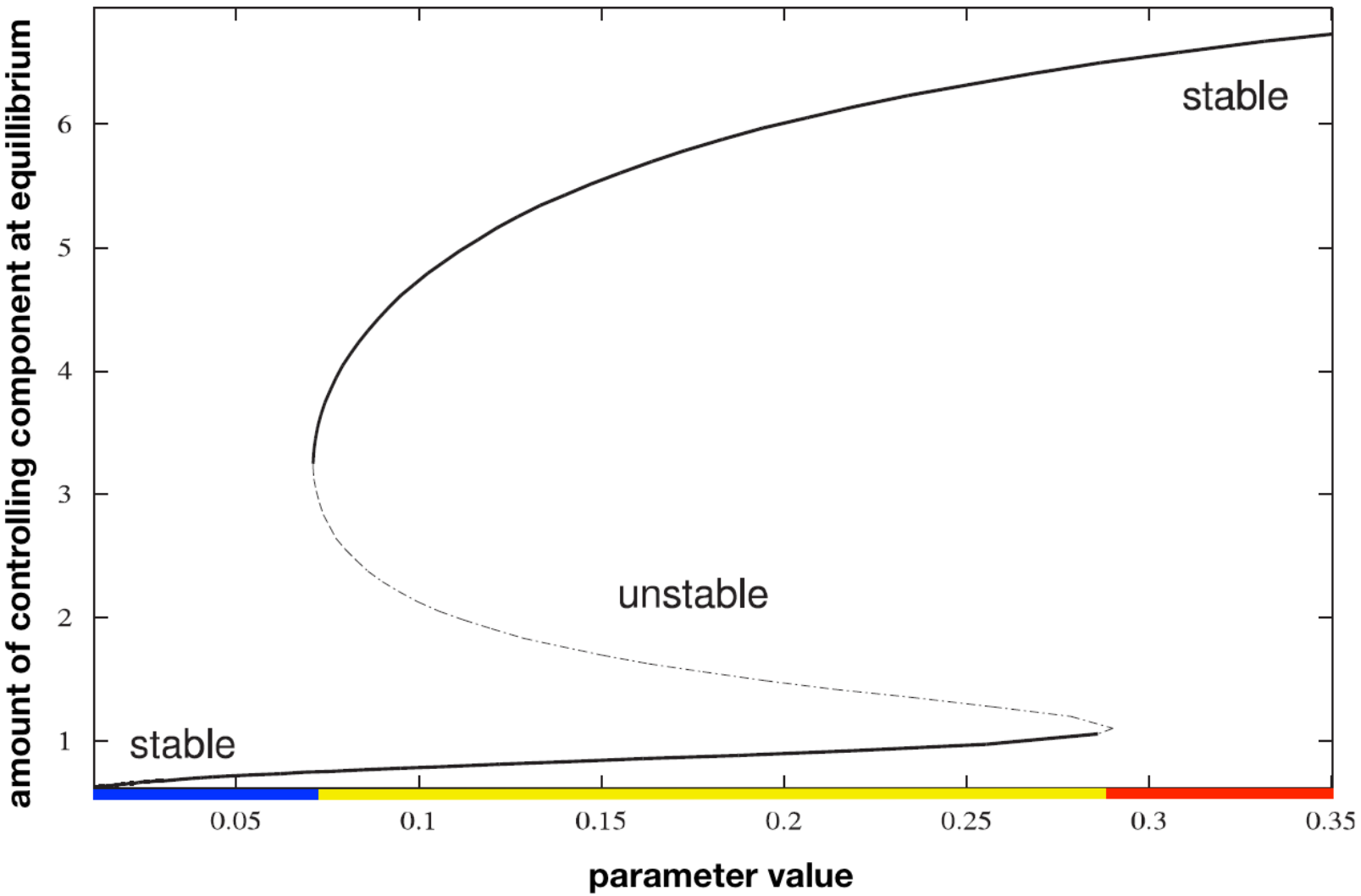
typical **phase portraits** around equilibria:



- 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**

Motivation: How Parameters Influence Systems Dynamics?

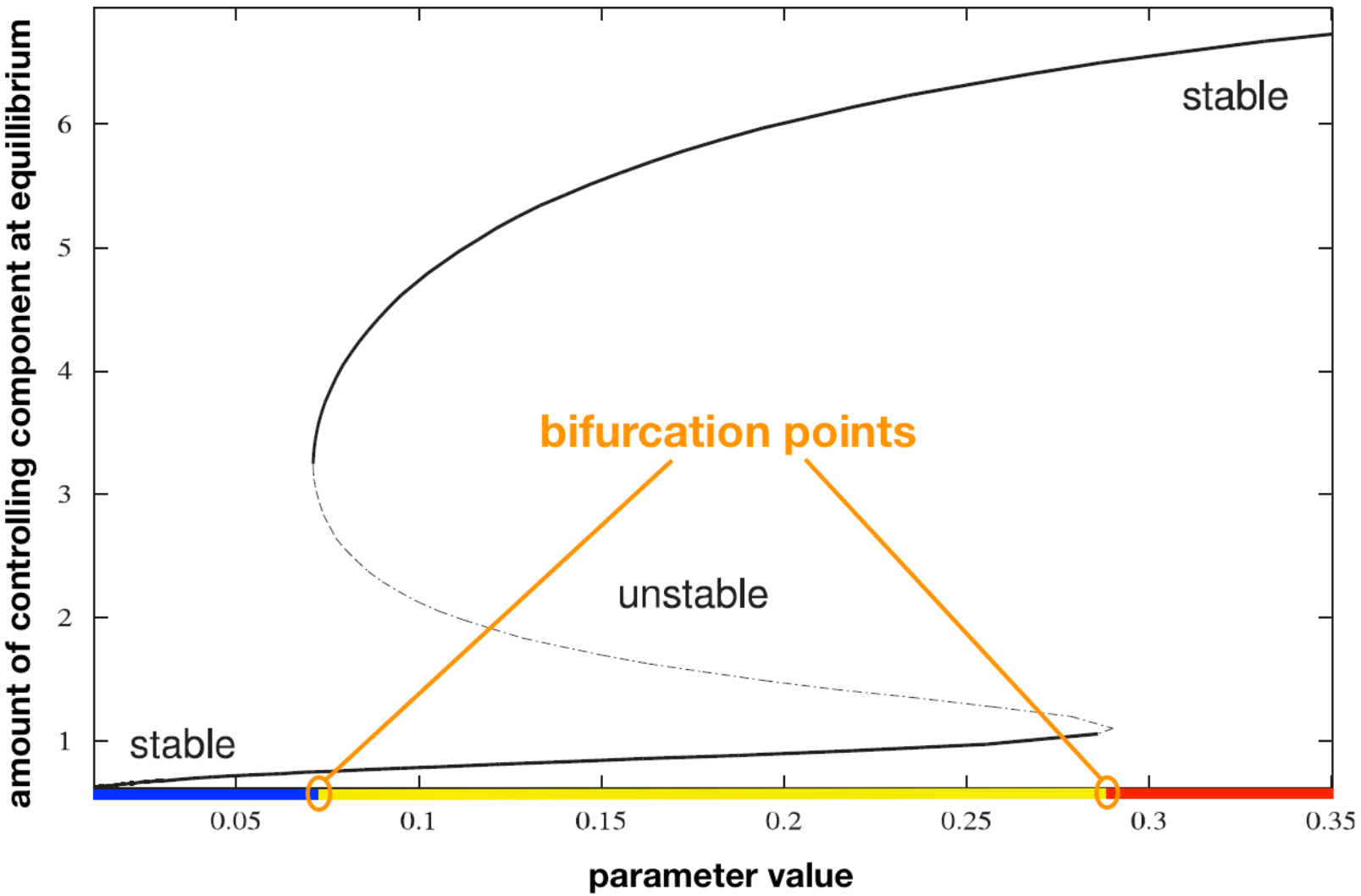
Bifurcation Analysis in Systems Theory



cannot divide can decide for division must divide

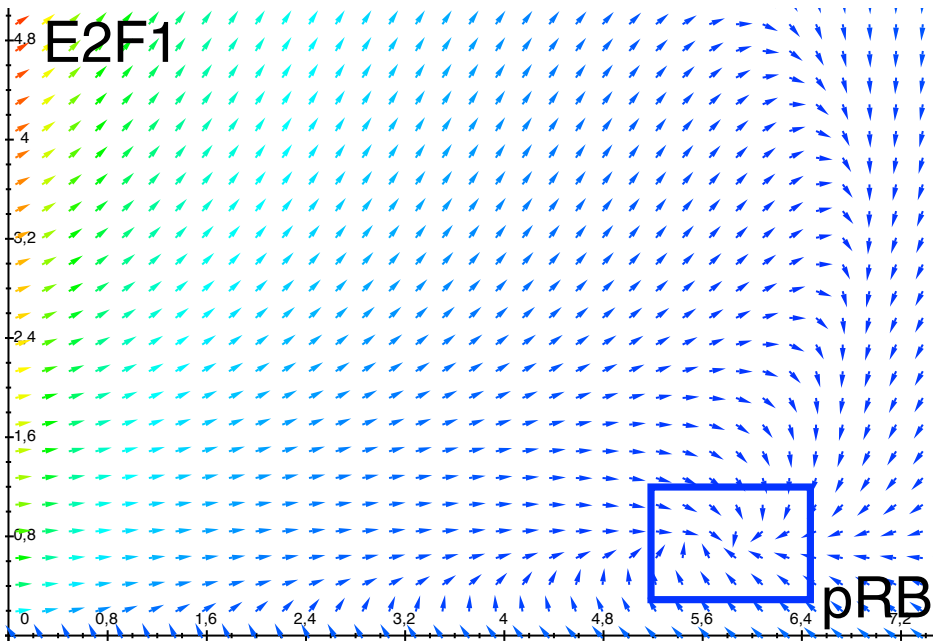
Motivation: How Parameters Influence Systems Dynamics?

Bifurcation Analysis in Systems Theory

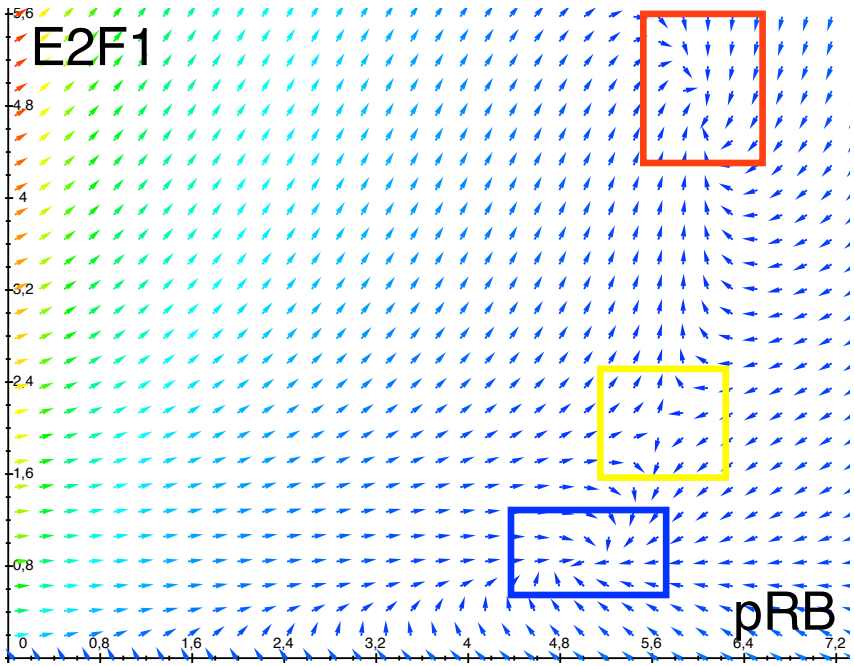


Motivation: Models of Complex Dynamical Systems
Understanding Role of Parameters

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



$p = 0.006$

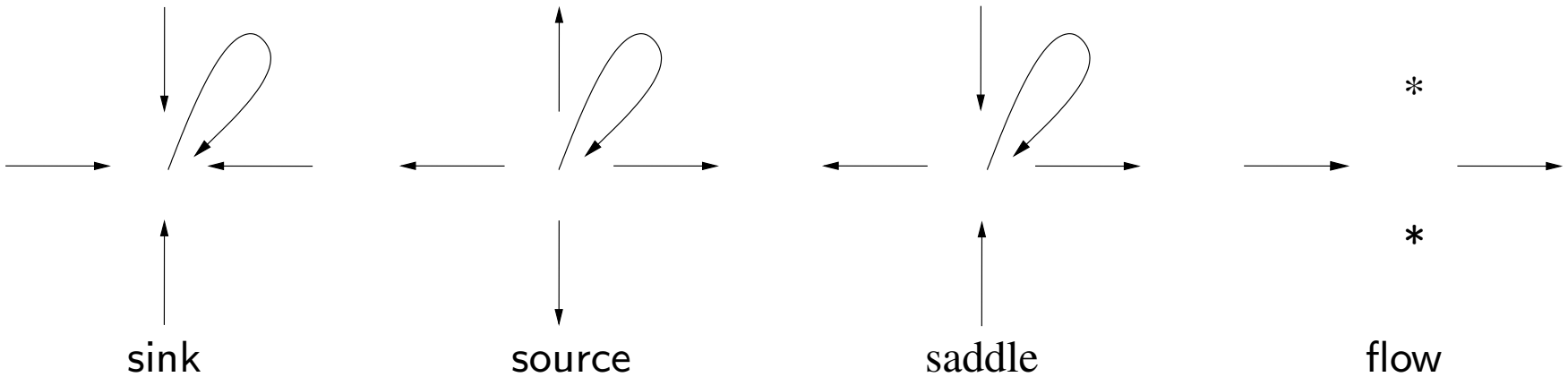


$p = 0.012$

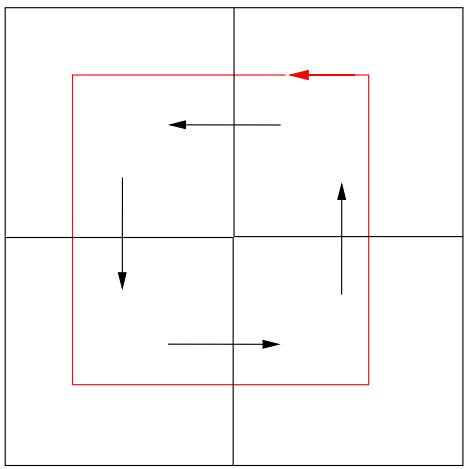
Phase Portrait Specification

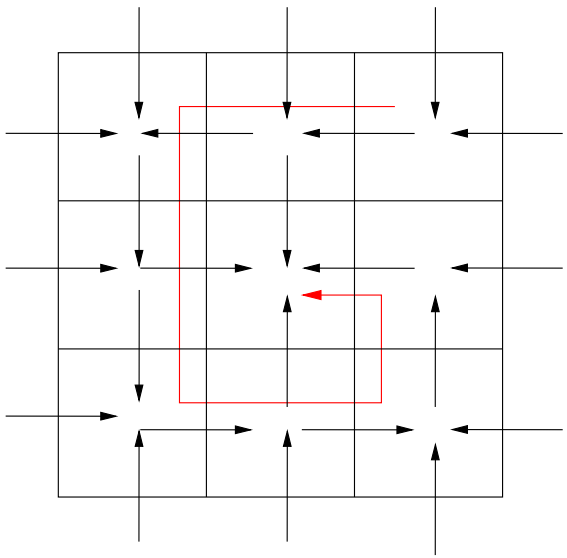
Elementary Patterns

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], ...

to combine all of these we introduce $\mathrm{HUCTL_P}$

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$_{\mathsf{P}}$

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:
$\exists s. \exists t. (@s. \neg t \wedge \mathbf{AX}\, s) \wedge (@t.\, \mathbf{AX}\, t)$

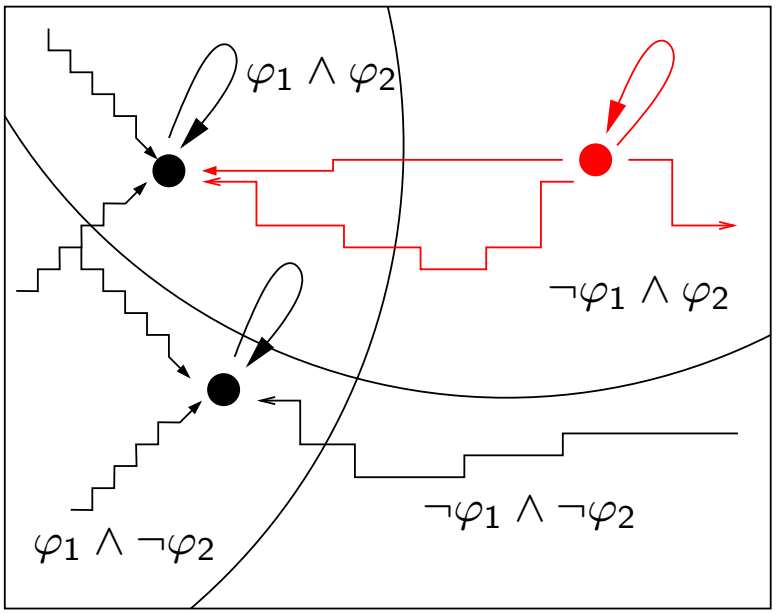
Phase Portrait Specification

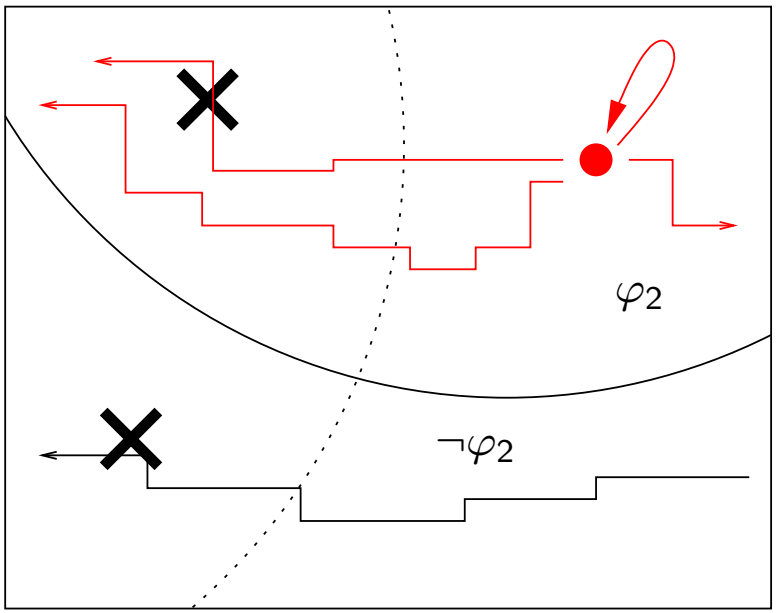
From Phase Portrait Specification to Parametric Phase Portrait

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





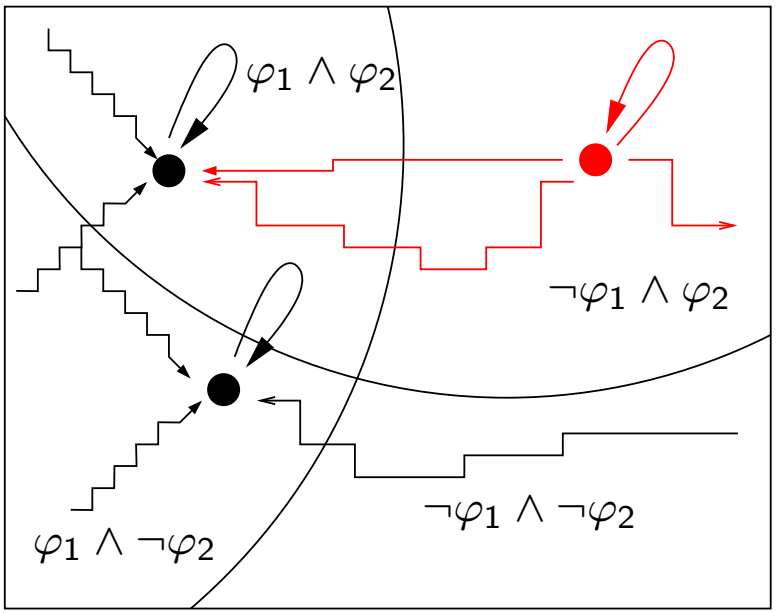
Phase Portrait Specification

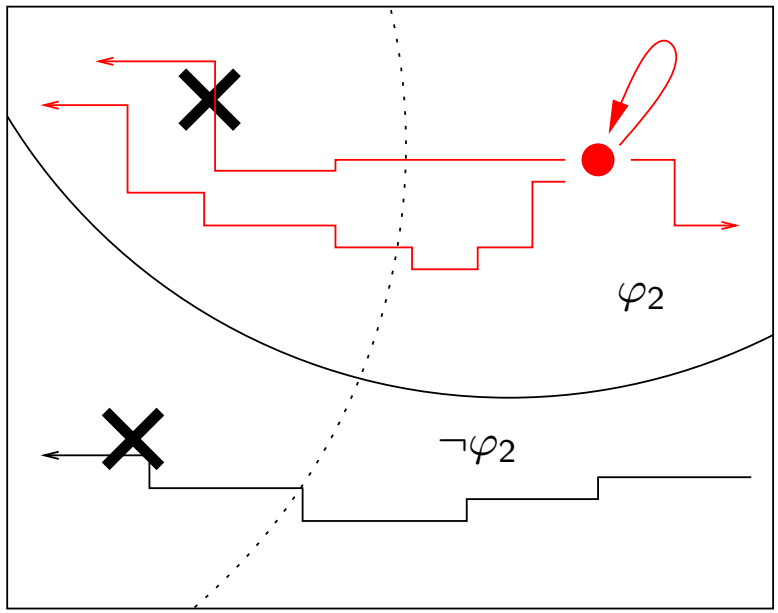
From Phase Portrait Specification to Parametric Phase Portrait

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





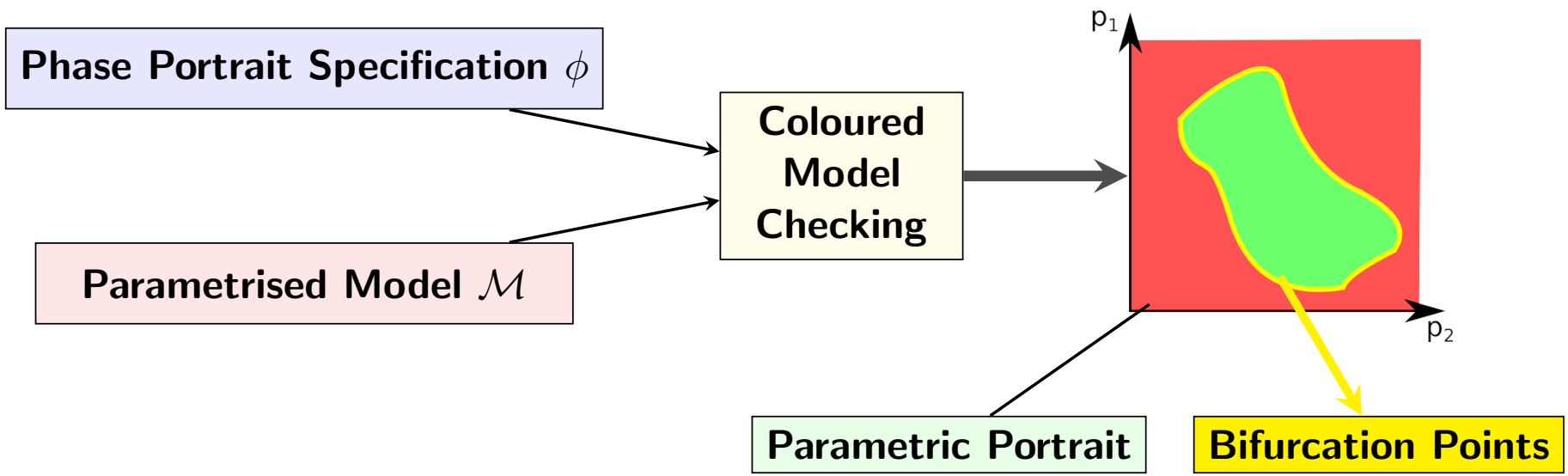
Parametric Phase Portrait

- 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

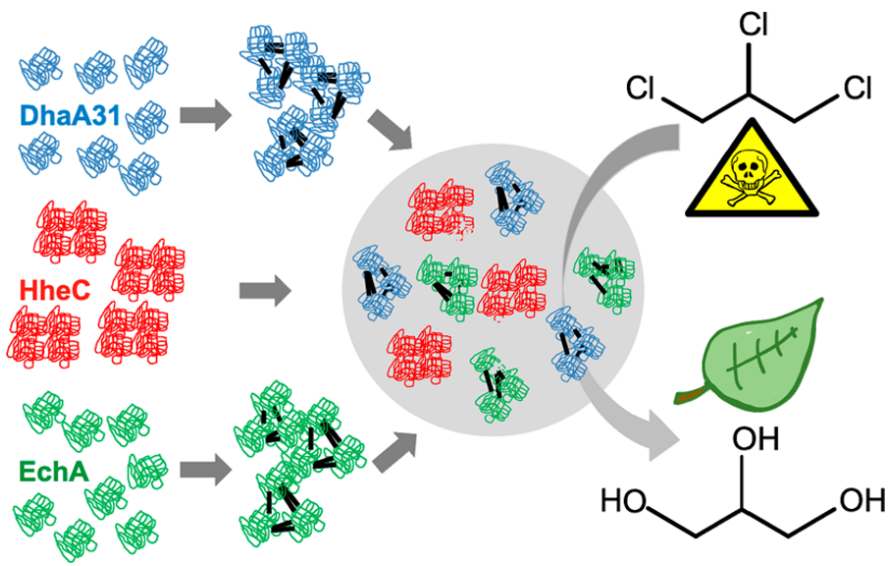
1. Motivation
2. Parameter Synthesis by Coloured Model Checking
3. Discrete Bifurcation Analysis
4. Case Studies
   - Case Study using Parameter Synthesis
   - Case Study using Discrete Bifurcation Analysis
5. Discussion

Outline

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

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





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

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

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

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

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

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

- 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:

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

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

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

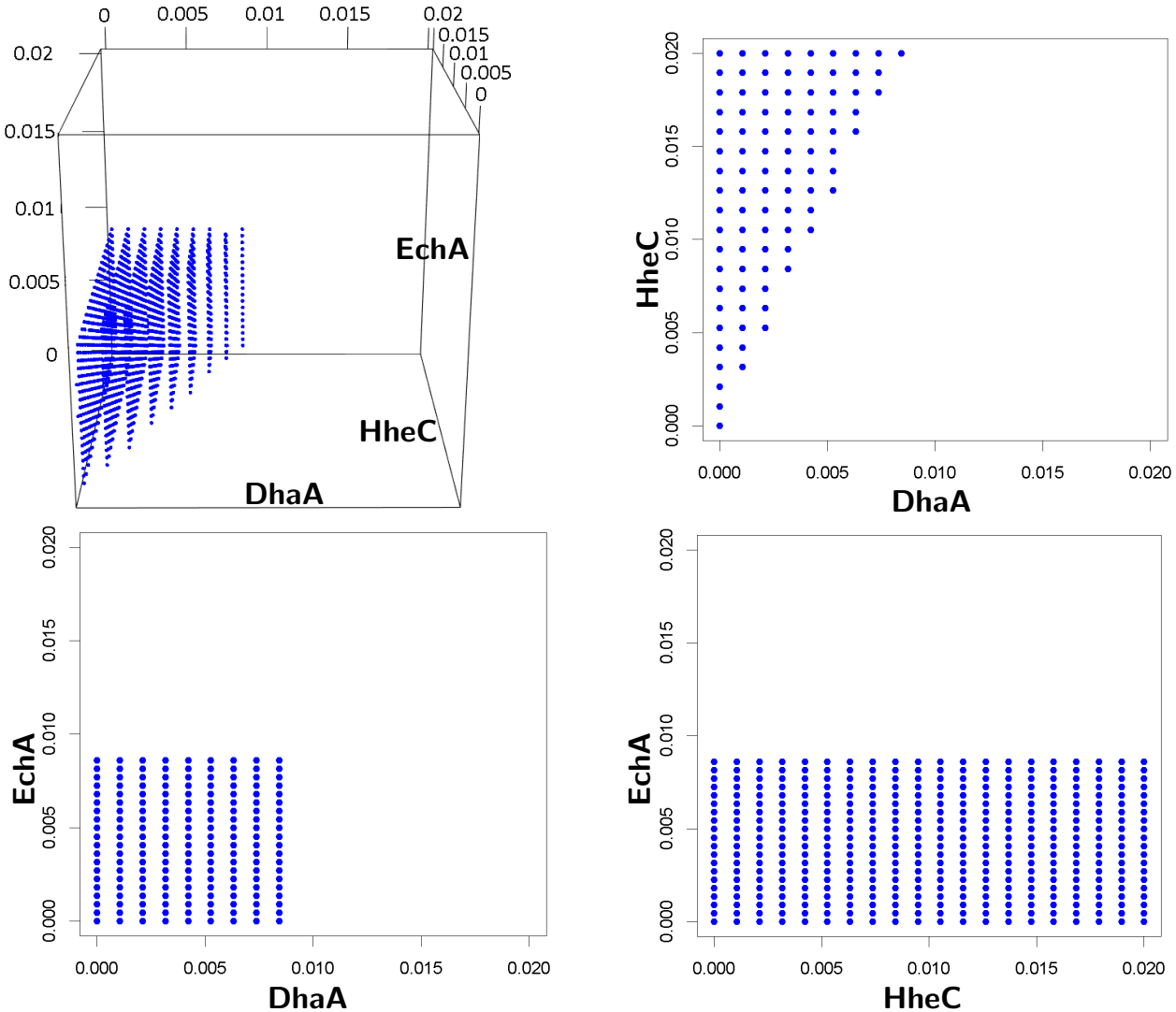
$$\varphi = (\varphi_1 \wedge \varphi_2 \wedge \varphi_3),$$

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*

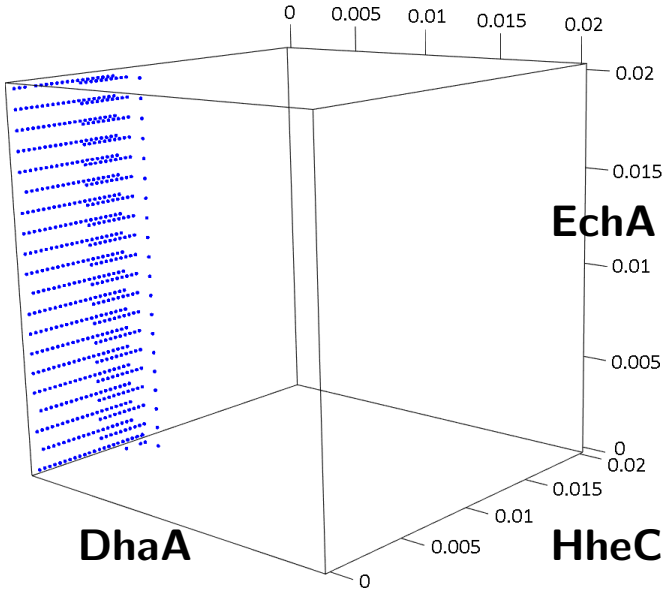


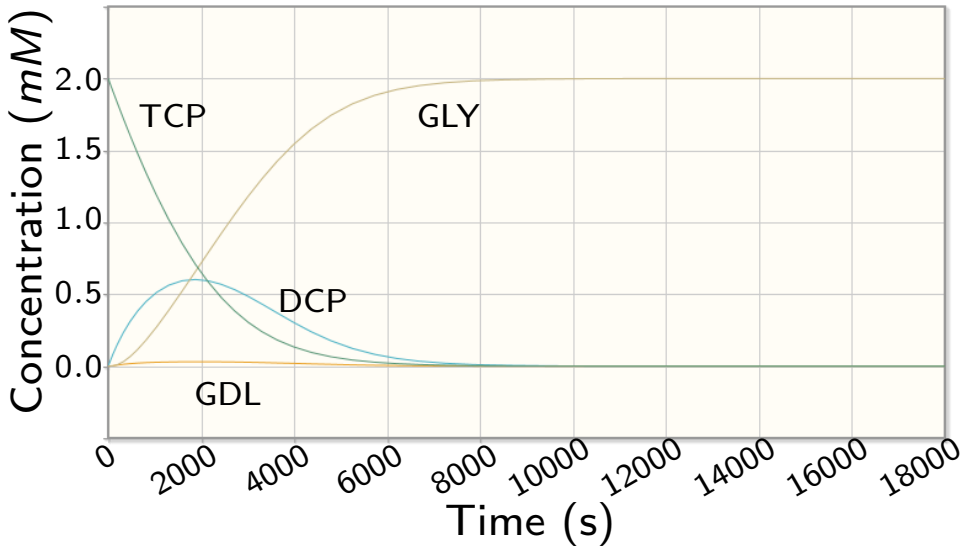
A sample of the resulting parameter space for a particular initial state:
$\mathrm{TCP} \in [1.9, 1.9586]$, $\mathrm{DCP} \in [0.448898, 0.5]$, $\mathrm{GDL} \in [0.0, 0.0669138]$, $\mathrm{GLY} \in [0.0, 0.01]$

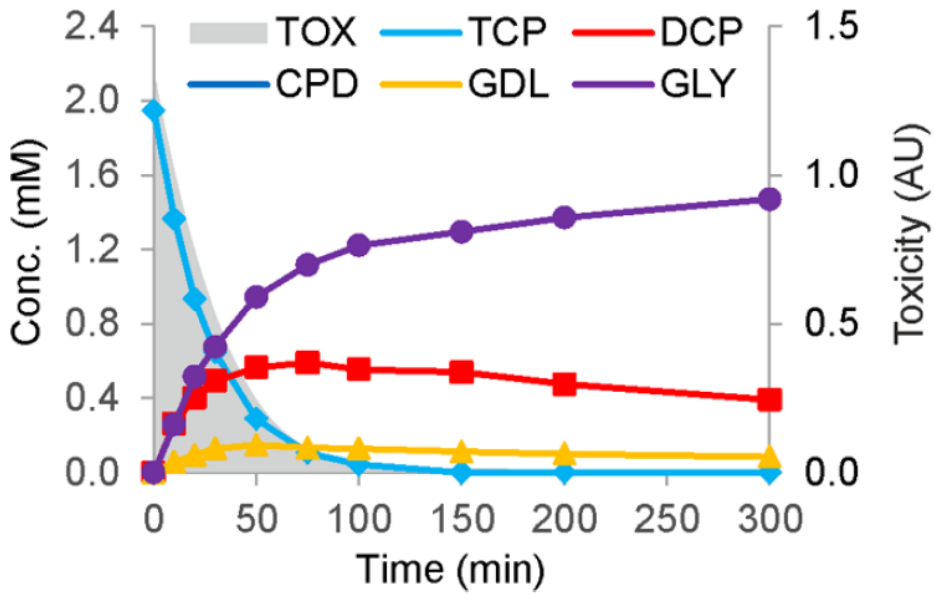
Dotted area corresponds to φ ($v = 0.5$, $w = 0.25$).

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

Preliminary Biological Validation



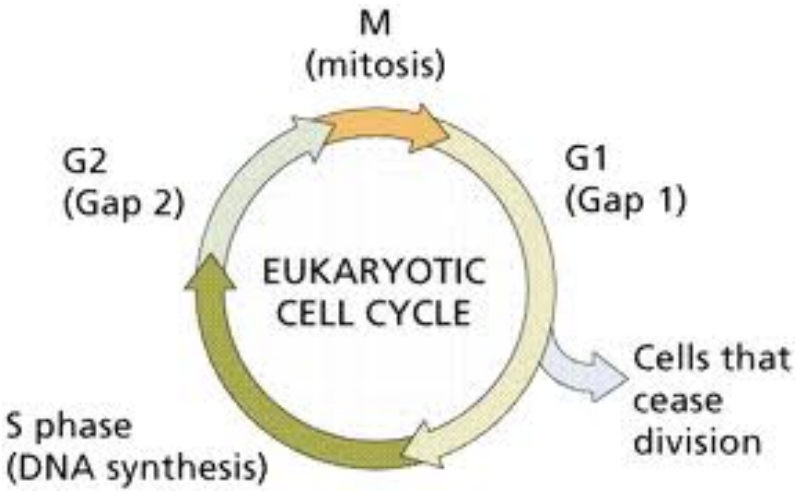


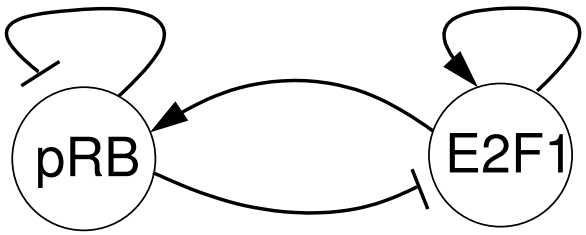


Outline

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

Case Study: Regulation of G_1/S Cell Cycle Transition





[Swat et al. 2004]

$$\frac{d[pRB]}{dt} = k_1 \frac{[E2F1]}{K_{m1}+[E2F1]} \frac{J_{11}}{J_{11}+[pRB]} - \phi_{pRB}[pRB]$$

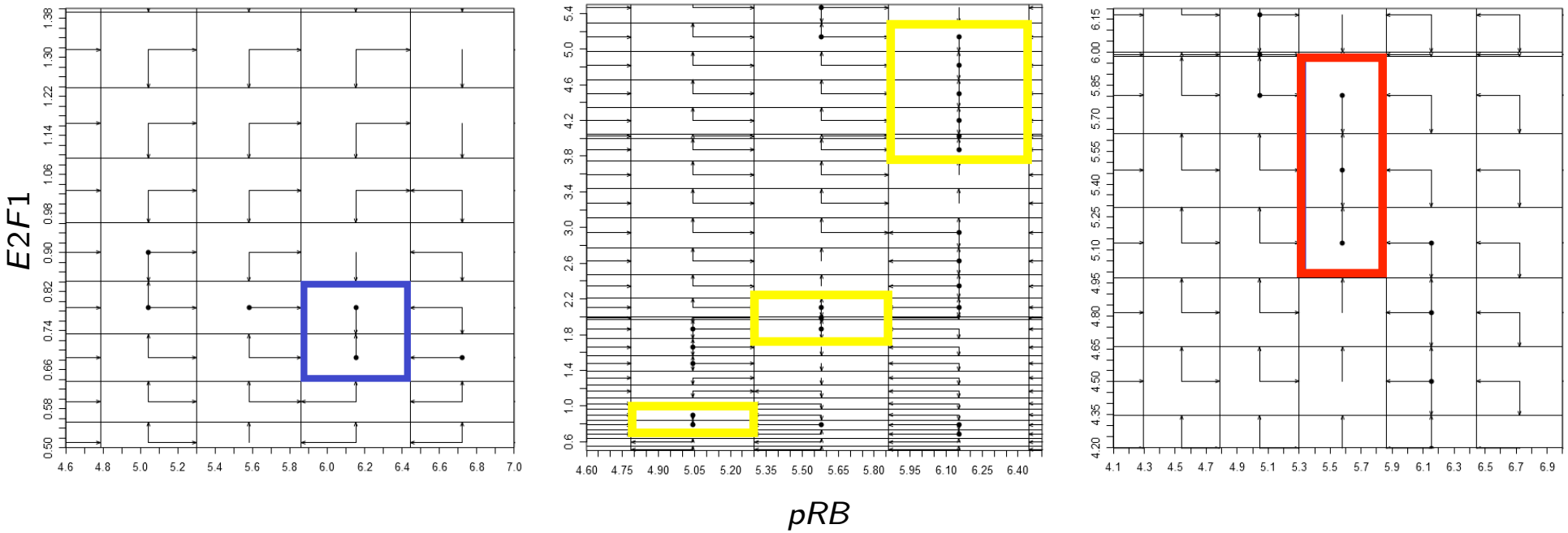
$$\frac{d[E2F1]}{dt} = k_p + k_2 \frac{a^2+[E2F1]^2}{K_{m2}^2+[E2F1]^2} \frac{J_{12}}{J_{12}+[pRB]} - \phi_{E2F1}[E2F1]$$

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: Results



φ_2

$\phi_{pRB} = 0.0075$

$[0.002, 0.011]$

φ_1

$\phi_{pRB} = 0.0115$

$[0.011, 0.0136]$

φ_3

$\phi_{pRB} = 0.014$

$[0.0136, 0.5]$

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

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

Outline

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

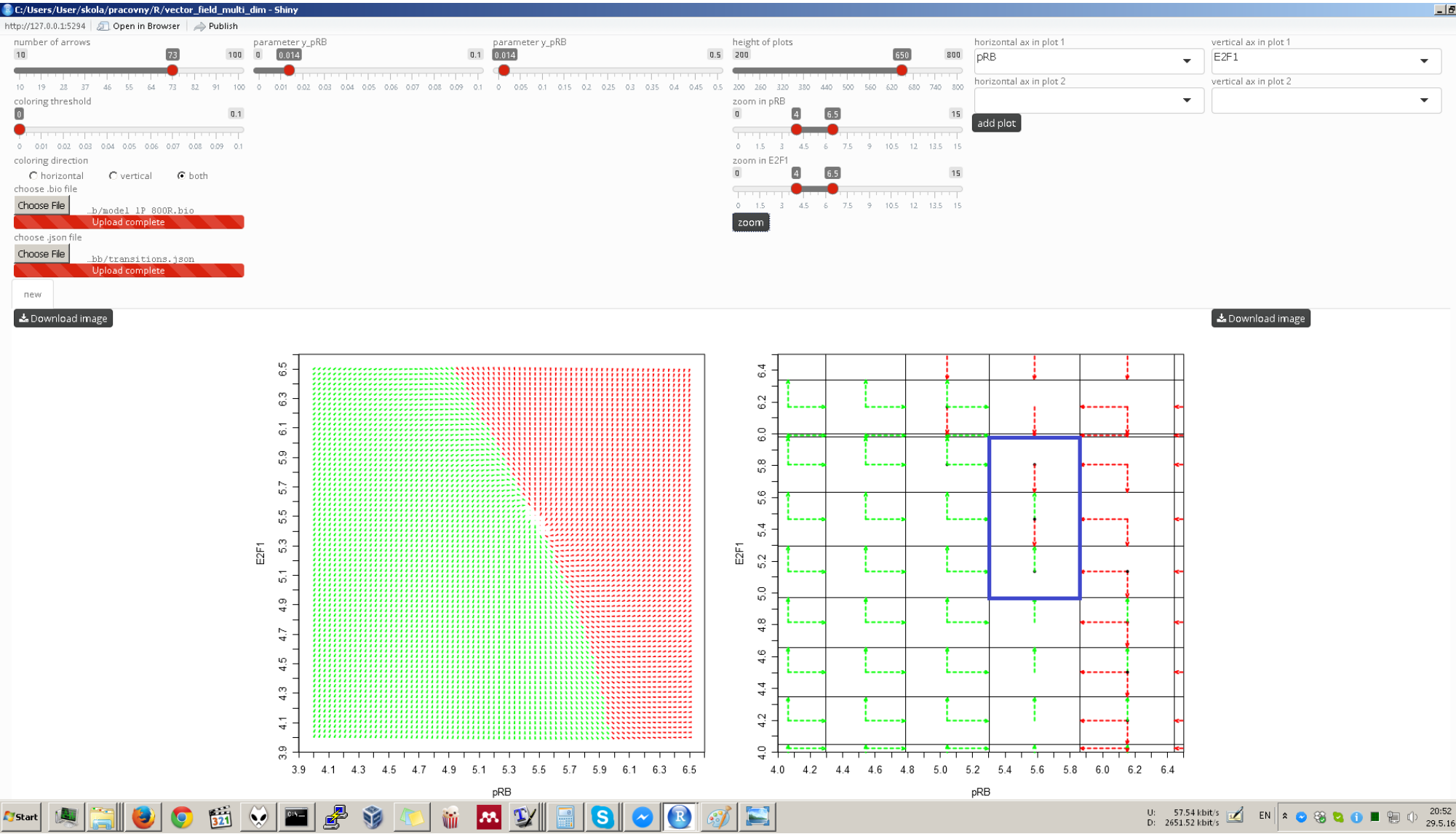
Summary and Conclusions

- exploiting dynamical systems under parameter uncertainty by enumerative model checking with symbolic representation of parameters
- good scalability achieved for interval-based representation of parameters
- scalability of SMT-based version depends on model structure
- remaining challenges:
  - approximation: explore errors, what can be guaranteed?
  - abstraction: narrow the extent of overapproximation – can Darboux polynomials or barrier certificates help?
  - improve scalability
  - is it possible to combine model checking with simulation? [TCSB XIV, 2012]

Discussion

- relations to parametric model checking
  - action synthesis [Knapik et al. 2014]
  - entirely symbolic approaches
  - in our case action parameters depend on states
- bifurcation of programs/protocols
  - a form of robustness analysis
- abstract interpretation wrt parameters
- parameter space compaction

Check Out Pithya



https://github.com/sybila/pithya-gui

Digital Systems Biology Laboratory

sybila
systems biology laboratory
http://sybila.fi.muni.cz



- since 2009
- originated in model checking community
- involved in systems biology since 2007 (EC-MOAN FP6)
  - parallel model checking algorithms development
- adapting formal methods to applications in systems biology
  - formal specification and modelling of biological systems
  - parameter synthesis and bifurcation analysis
- contributions to systems biology and bioinformatics
  - integration of formal methods with bionformatics and experimental biology
  - case studies with biological groups

Credits

Computer Science

Luboš Brim, Marta Kwiatkowska, Thomas Henzinger, Loïc Paulevé, Ezio Bartocci, Luca Bortolussi, Jérôme Feret, Andrzej Mizera, Alessandro Abate, Jan Van Schuppen, Milan Češka, Nikola Beneš, Stefan Haar, Heike Siebert, Hidde de Jong

Biology

Ralf Steuer, Louis Mahadevan, Jan Červený, Dušan Lazár, Pavel Krejčí, Stefanie Hertel, Christoff Flamm

Funding

- Grant Agency of Czech Republic
- C4SYS – National Infrastructure for Systems Biology