Model Checking in Systems Biology

Luboš Brim and Milan Češka and David Šafránek

Masaryk University
Czech Republic

Outline

1. Introduction
2. LTL Model Checking
3. Parallel LTL Model Checking
4. Discrete Abstraction of ODE Models
5. Case Studies
   - Model Checking of E. Coli Ammonium Transport
   - Parameter Synthesis by Model Checking
   - Parameter Synthesis and Classification for Boolean Networks

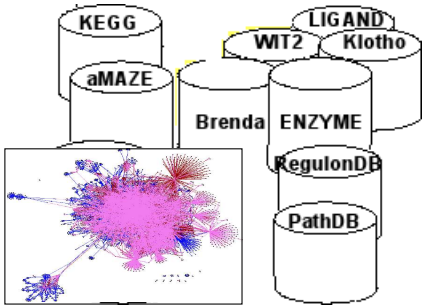
Outline

1. Introduction
2. LTL Model Checking
3. Parallel LTL Model Checking
4. Discrete Abstraction of ODE Models
5. Case Studies
   - Model Checking of E. Coli Ammonium Transport
   - Parameter Synthesis by Model Checking
   - Parameter Synthesis and Classification for Boolean Networks

Motivation
Model Analysis in Systems Biology

network reconstruction

biological knowledge databases

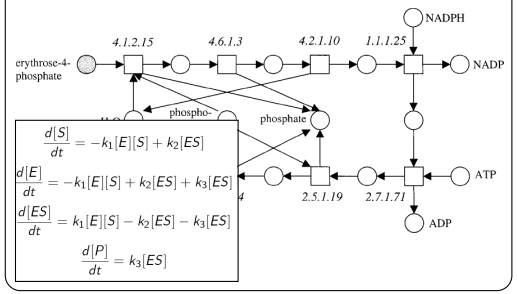


biological network

hypothesis

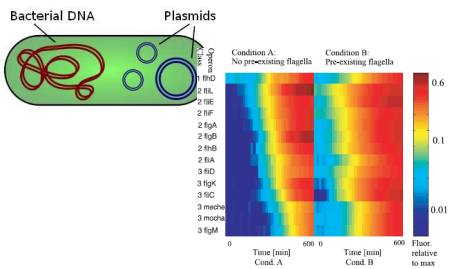
model specification

SBML, diferenciální rovnice,
boolovská sít, Petriho sít, ...



model validation

gene reporters, DNA microarray,
mass spectrometry, ...

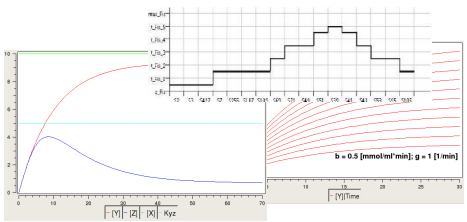


emergent properties

model questions

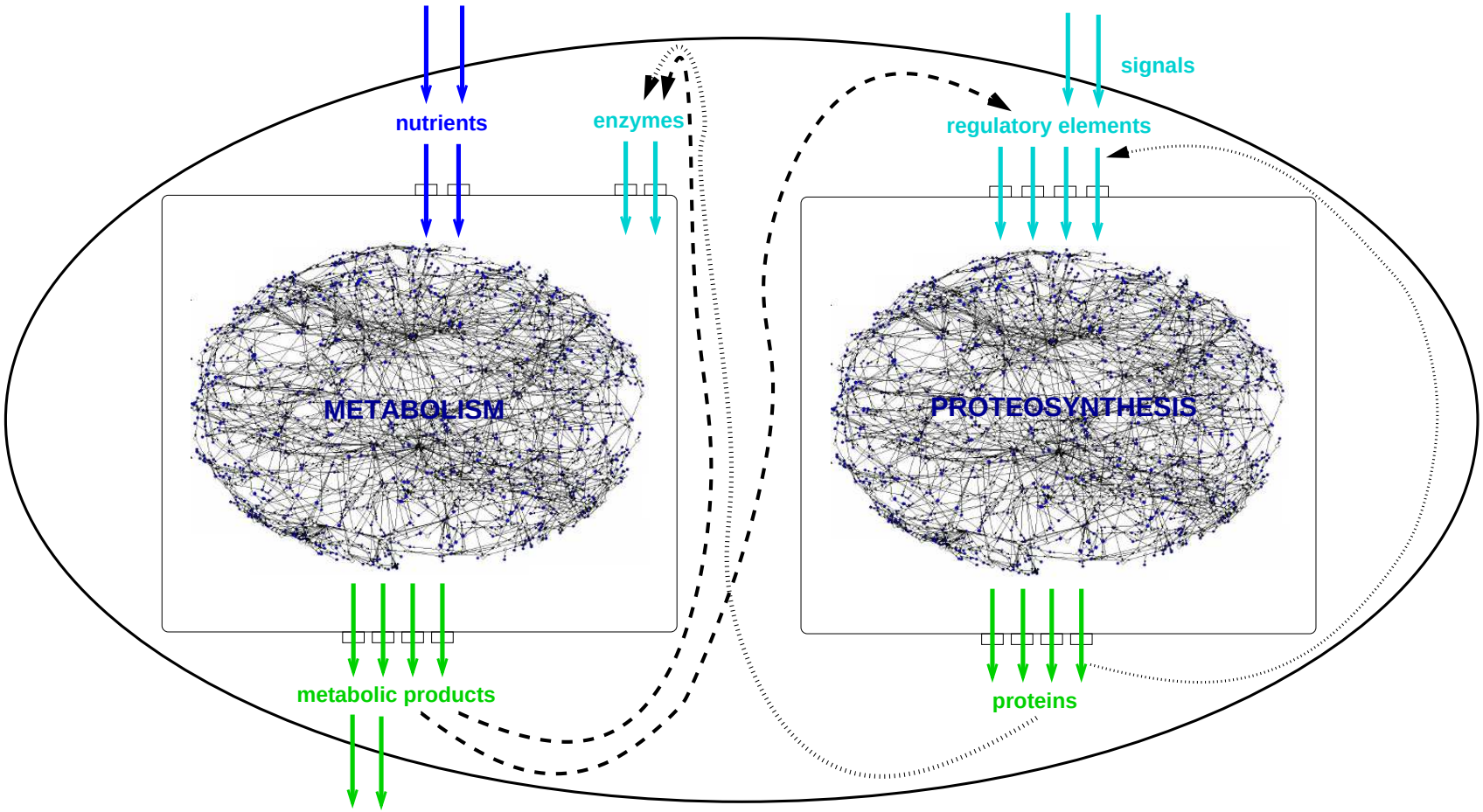
model analysis

static analysis, numerical simulation,
analytical methods, model checking

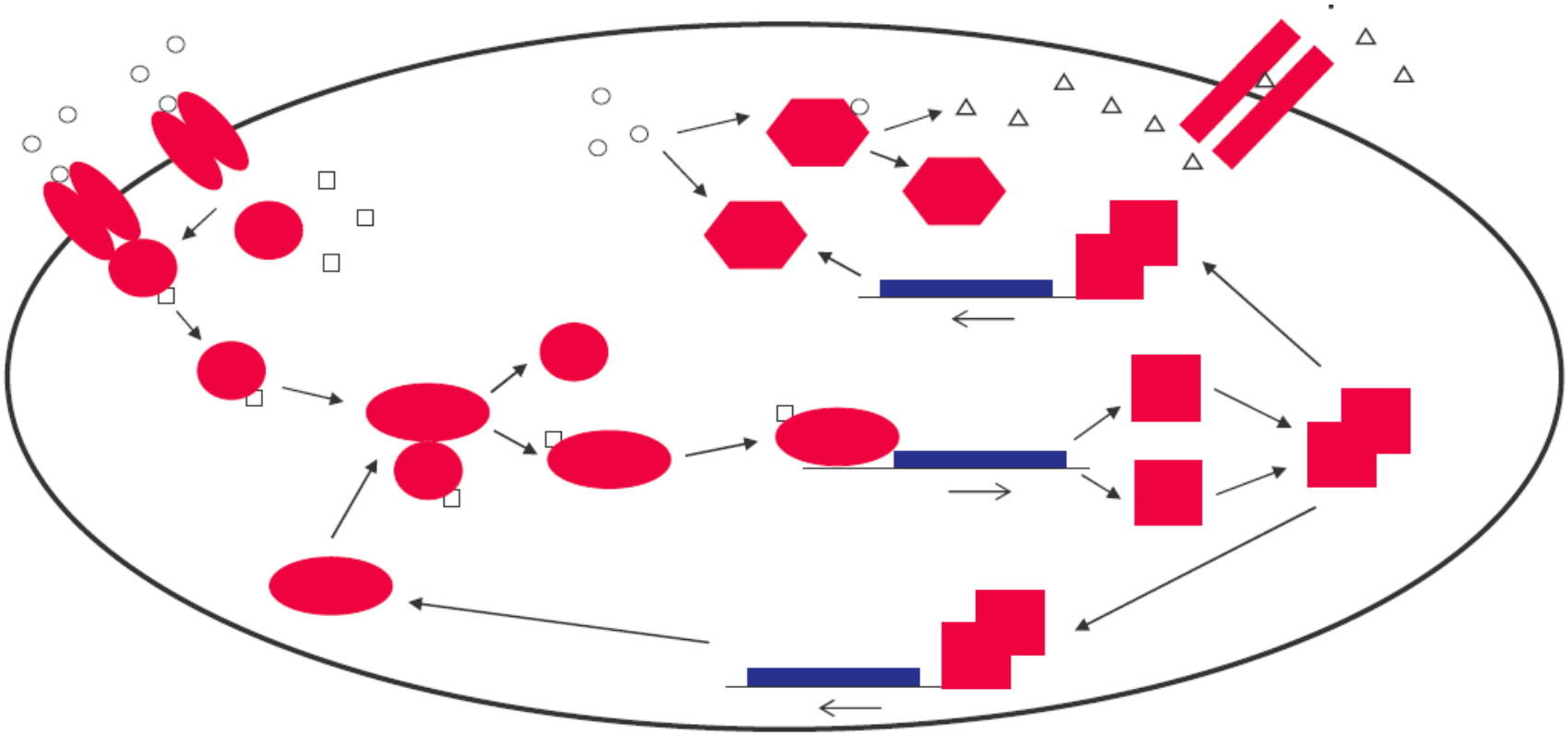


hypothesis testing, property detection,
new hypothesis inference

Systems View of Processes Driving the Cell

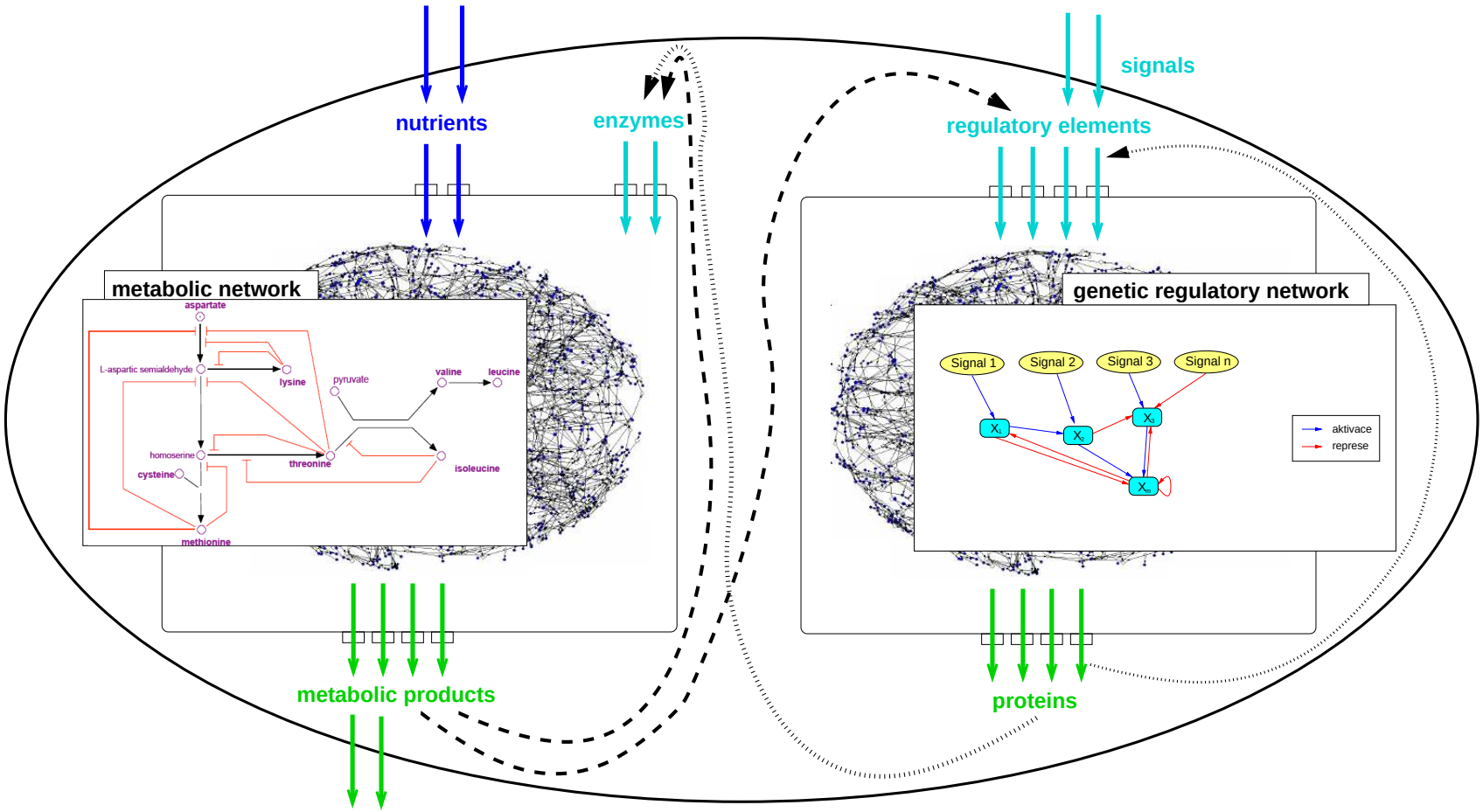


Systems View of a Cell: Biological Networks

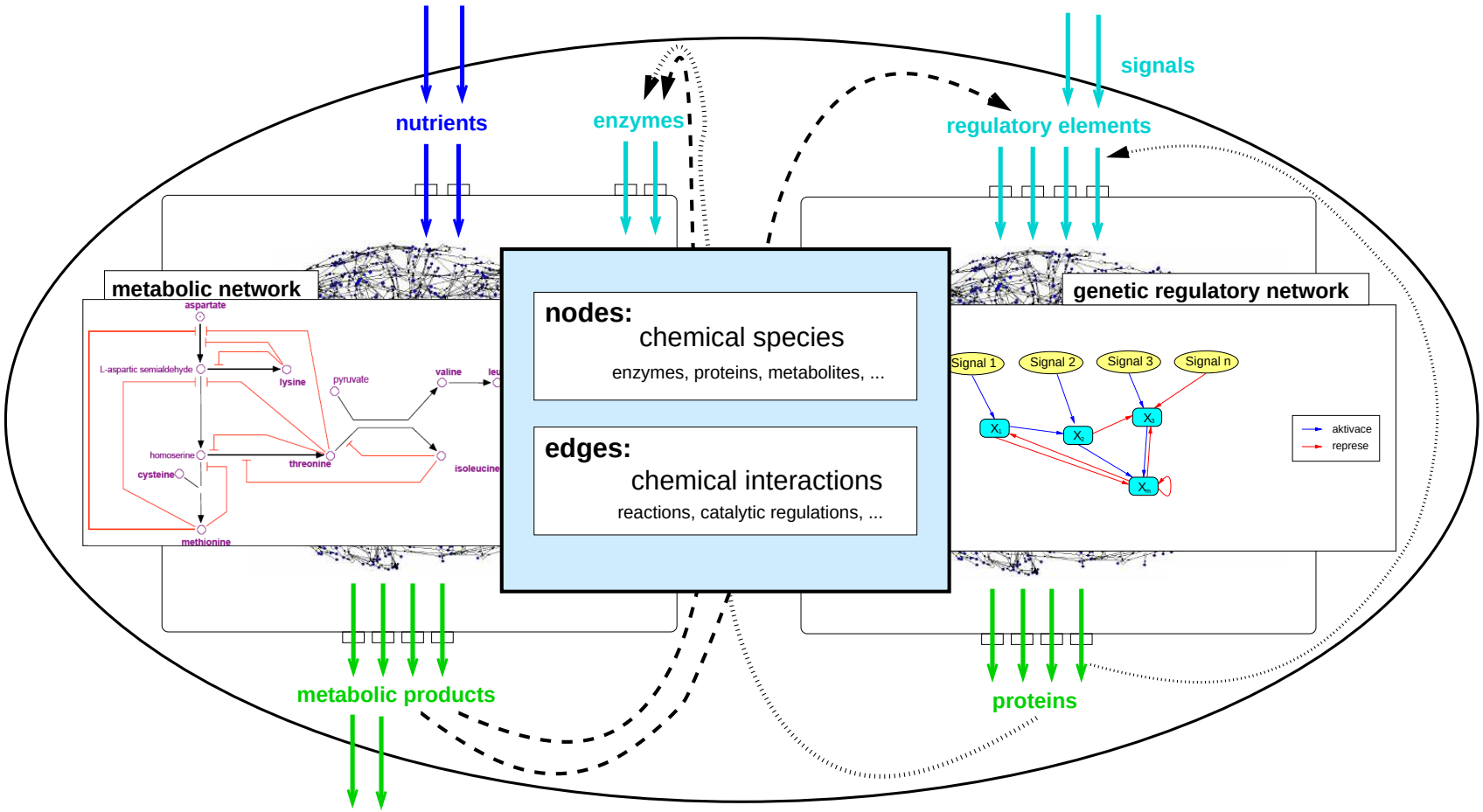


- identify substances (macromolecules, ligands, proteins, genes, . . .)
- identify interactions ((de)complexation, (de)phosphorylation, . . .)

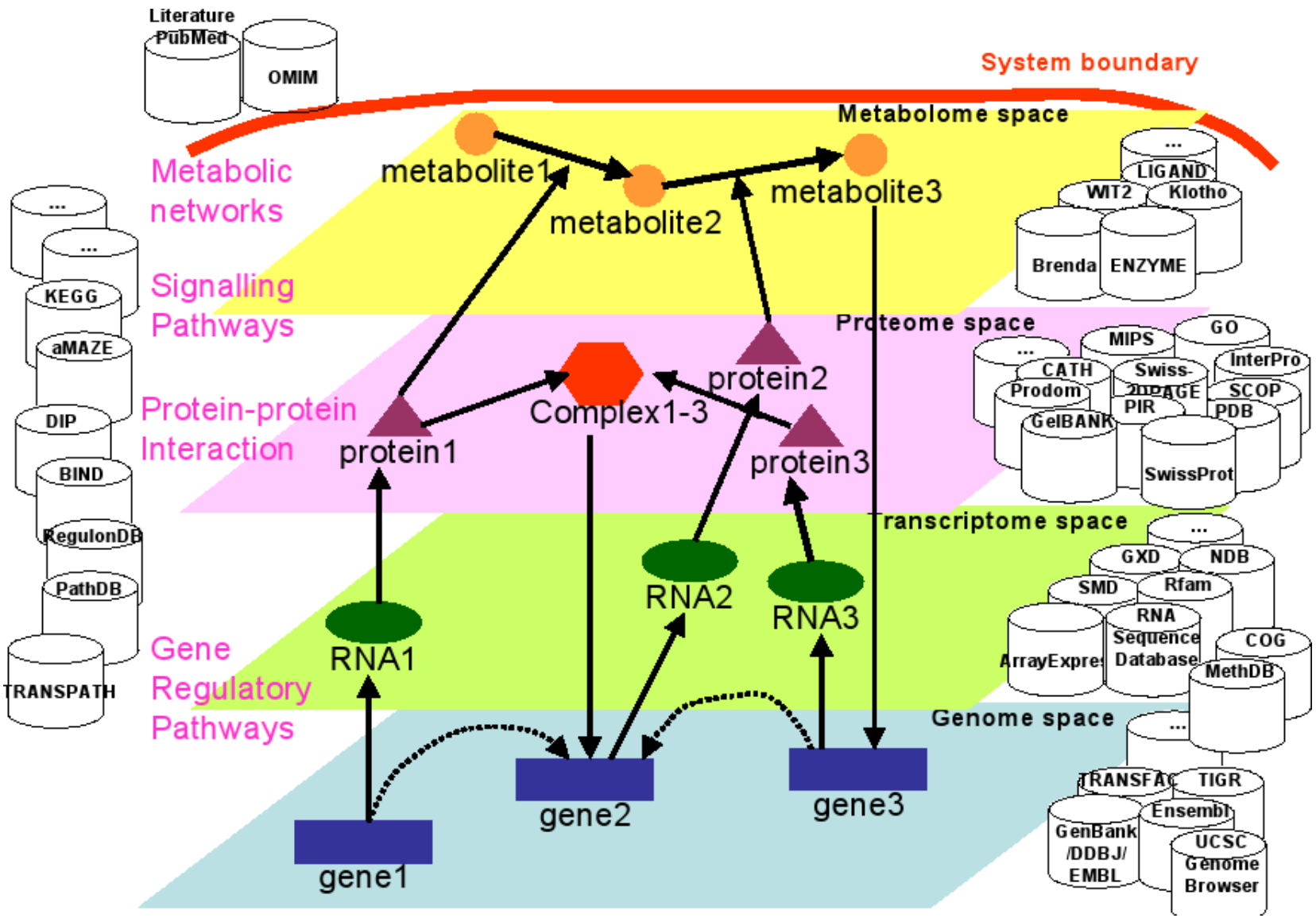
Systemic View of the Cell: Biological Networks



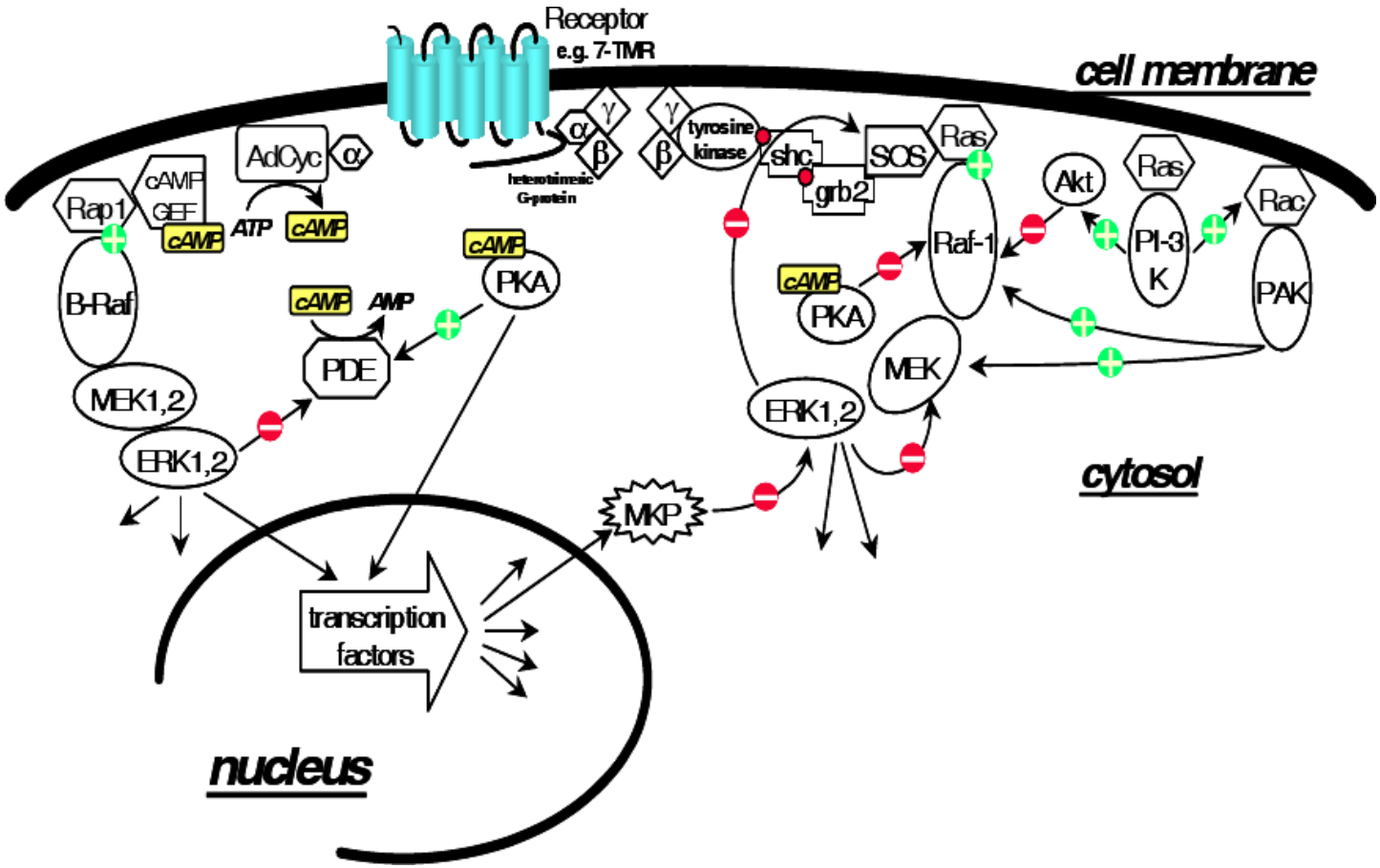
Systemic View of the Cell: Biological Networks



Systems Biology of a Cell

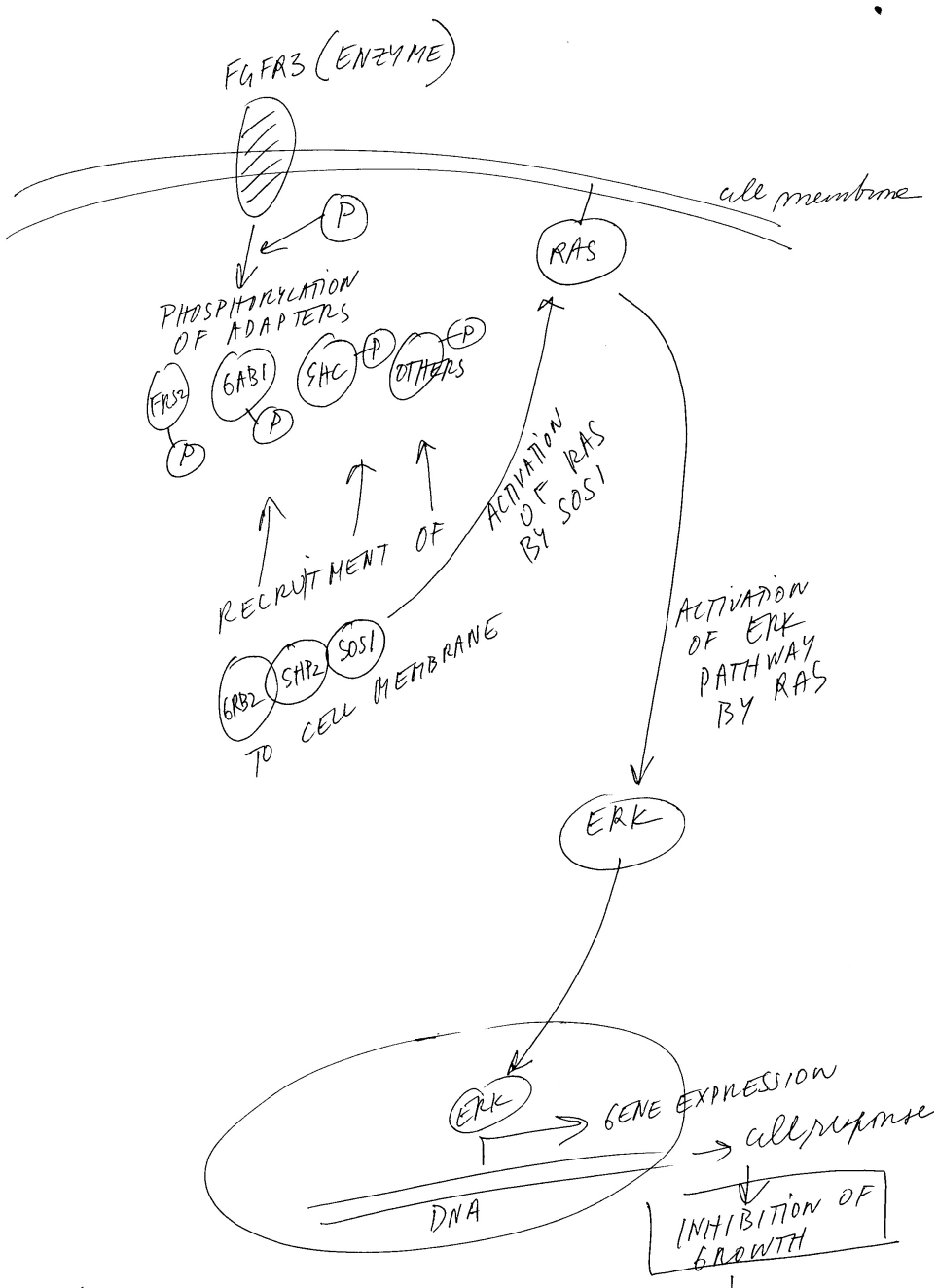


Biological Networks and Pathways



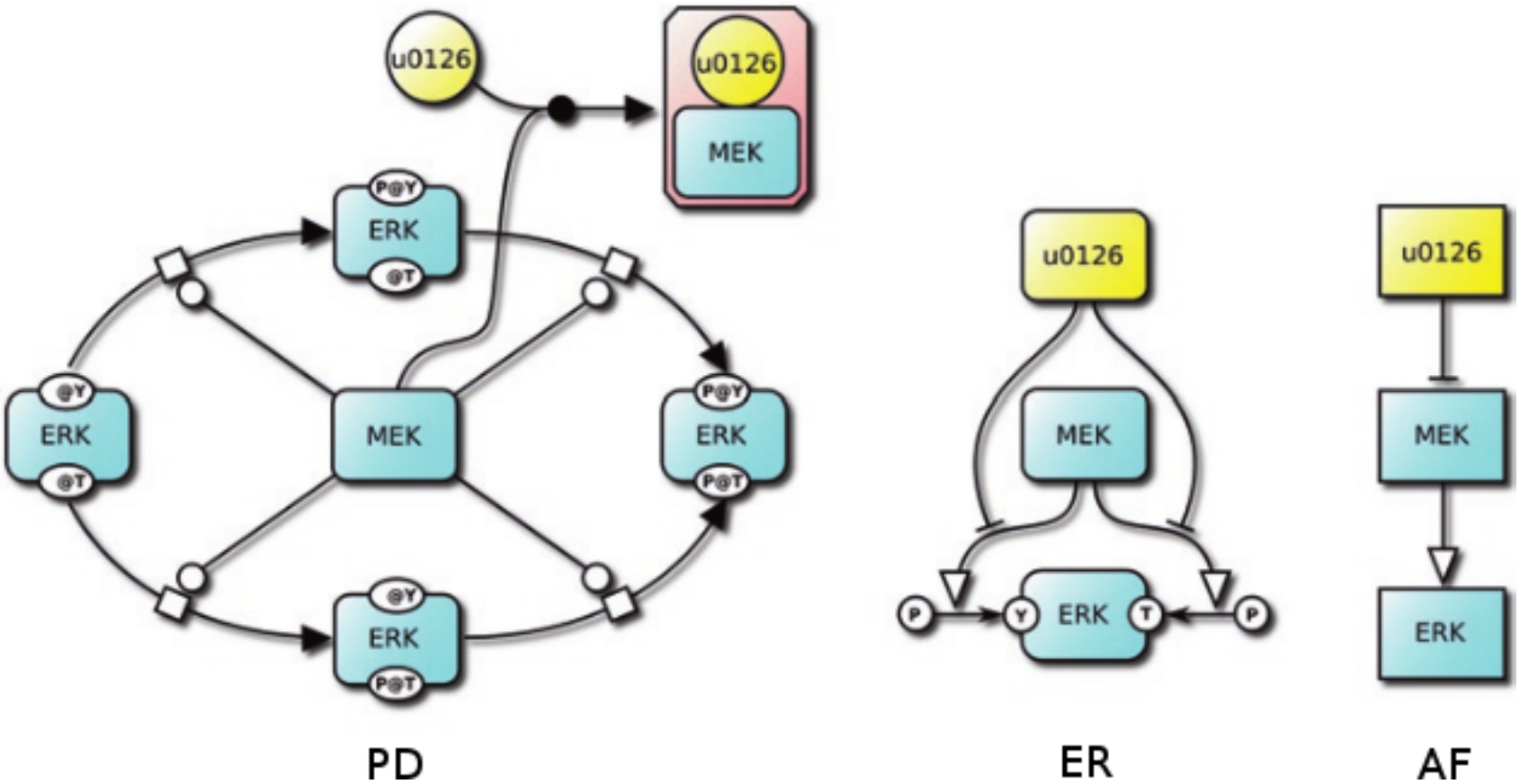
- what is the "right" meaning?
- in order to *analyse* we need to *formalise*

Biological Networks and Pathways



Graphical Specification in SBGN

Systems Biology Graphical Notation



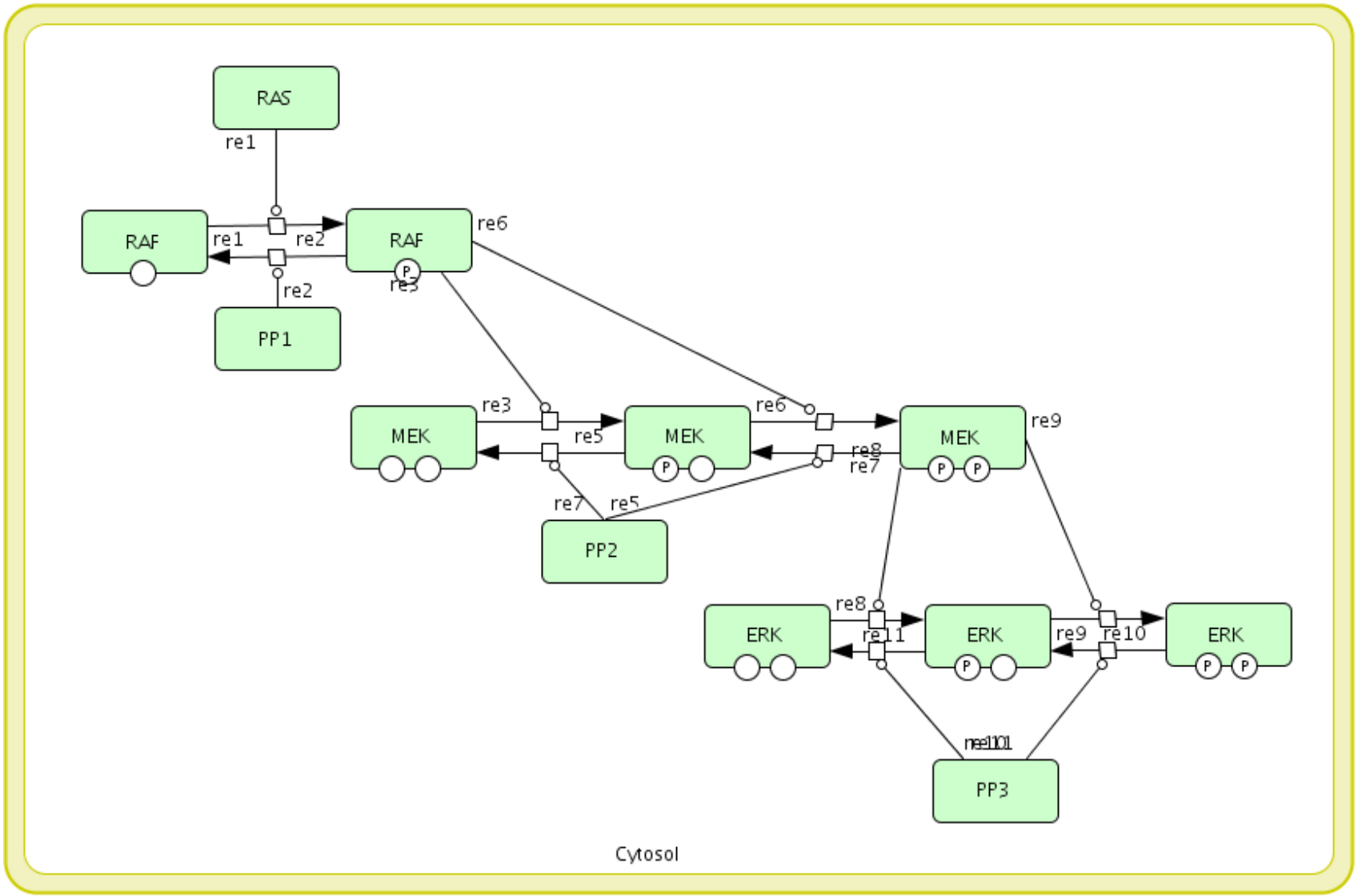
- PD: biochemical interaction level (the most concrete)
- ER: relations among components and interactions
- AF: abstraction to mutual interaction among activities

Graphical Specification in SBGN

Systems Biology Graphical Notation

- SBGN.org iniciative (from 2008)
- standard notation for biological processes
- `http://sbgn.org`
- Nature Biotechnology (doi:10.1038/nbt.1558, 08/2009)
- three sub-languages:
  - SBGN PD (Process Description) (doi:10.1038/npre.2009.3721.1)
  - SBGN ER (Entity Relationship) (doi:10.1038/npre.2009.3719.1)
  - SBGN AF (Activity Flow) (doi:10.1038/npre.2009.3724.1)
- tool support:
  - SBGN PD supported by CellDesigner
  - SBGN-ED (`http://www.sbgn-ed.org`)

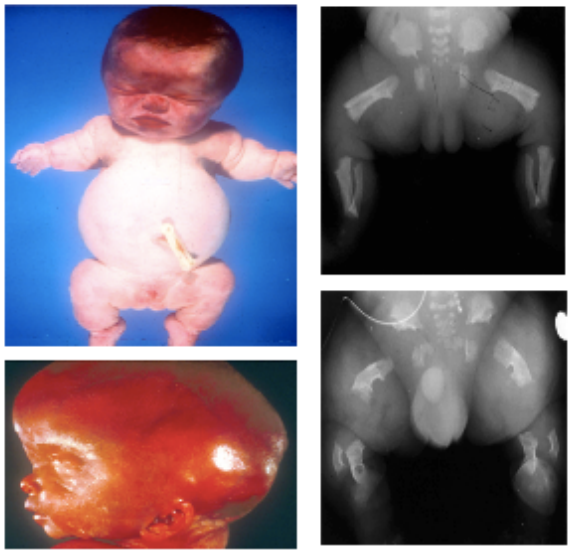
Kinase Cascade in CellDesigner (SBGN)



Why to model?

Achondroplasia

Thanatophoric Dysplasia

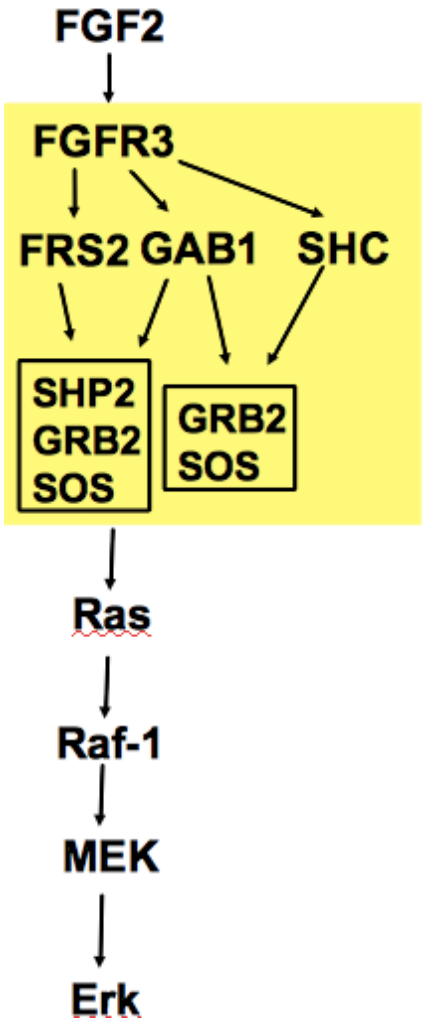


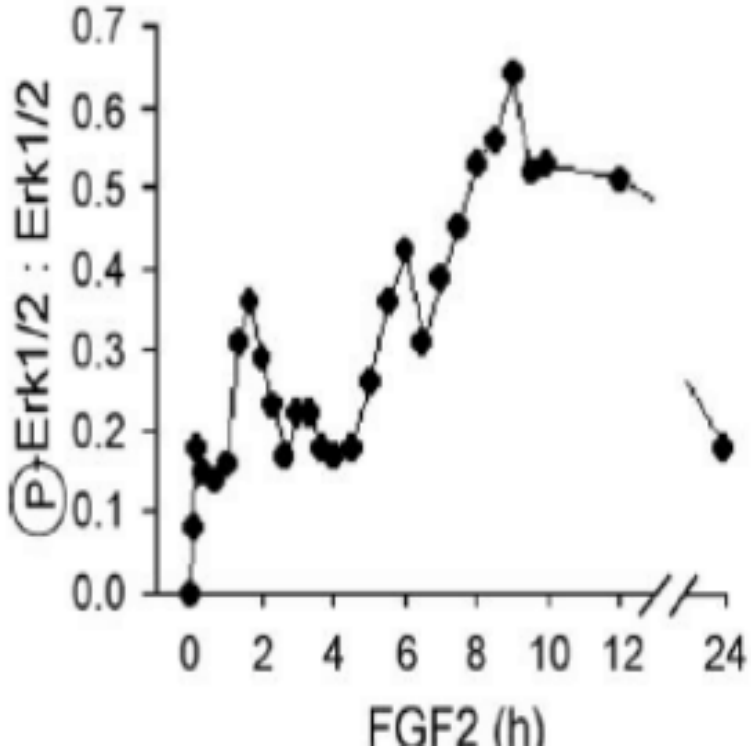
- short long bones
- brachydactyly
- macrocephaly
- low nasal bridge
- spinal stenosis
- temporal lobe malformations

Nat Genet 1995, 9:321-8.

e.g., FGFR3-related skeletal dysplasia

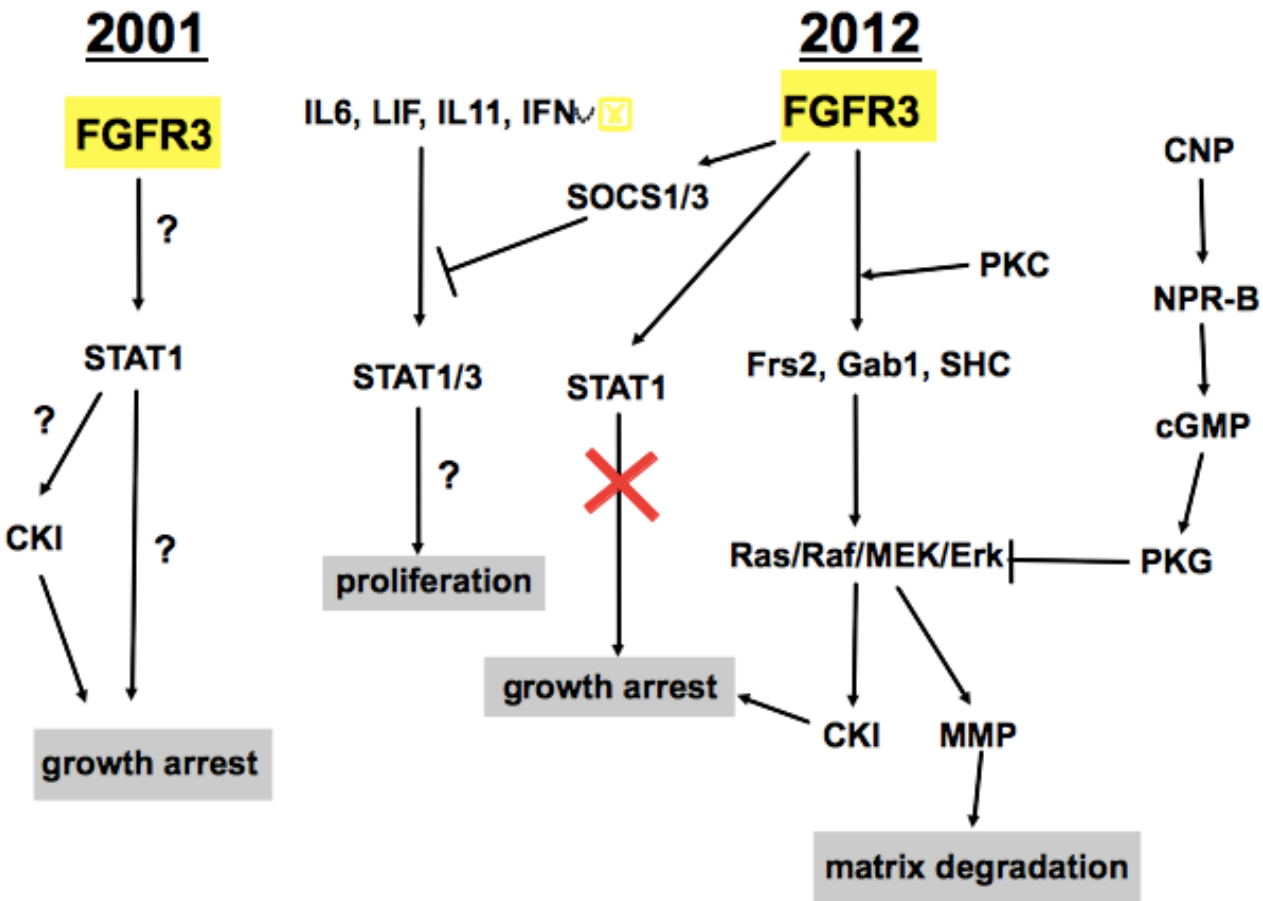
Why to model?
Need to explain...





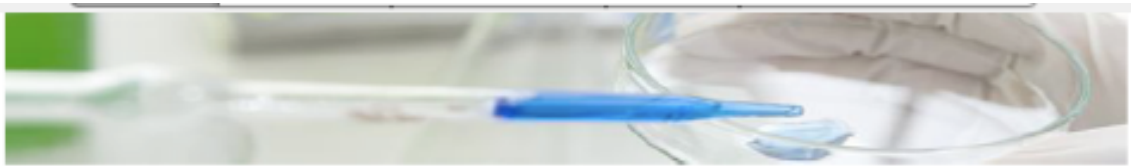
P. Krejčí, Masaryk University

Why to model?
Knowledge is increasing...



Why to model?

Investors



Home Investors RSS Content

- Overview
- Financial Information
- SEC Filings
- Corporate Governance
- Investors' Kit
- Calendar of Events

RSS Content

BioMarin Announces Program for BMN-111 for the Treatment of Achondroplasia

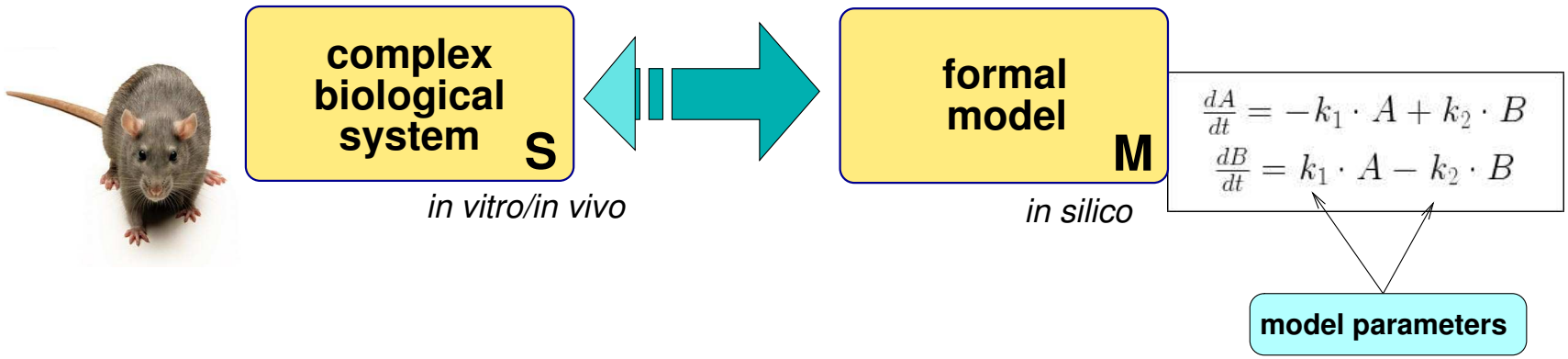
NOVATO, Calif., Oct 19, 2010 /PRNewswire via COMTEX/ --

BioMarin Pharmaceutical Inc. (Nasdaq: BMRN) today announced its program for BMN-111, a peptide therapeutic for the treatment of achondroplasia. BioMarin plans to file an IND in the fourth quarter of 2011 and to initiate a Phase 1 clinical trial by the first quarter of 2012.

BMN-111, for the treatment of achondroplasia, is an analog of C-type Natriuretic Peptide (CNP), a small cyclic peptide that is a positive regulator of bone growth. It is produced and has a receptor in the growth plate, and along with the fibroblast growth factor receptor 3 (FGFR3), regulates normal bone growth. In addition to short stature, there are complications in achondroplasia that are related to bone compression (e.g. foramen magnum narrowing, spinal stenosis, upper respiratory narrowing) of nervous tissues or other tissues.

Exp Cell Res. 2004;297:152-64.
J Cell Sci. 2005; 118: 5089-100.
J Biol Chem. 2007 ;282:2929-36.
Pediatr Res. 2007; 61(3):267-72.
Invest New Drugs 2007; 25:391-95.
PLoS One 2008; 3:e3961.
J Cell Sci 2008; 121:272-81.
Cell Signal 2009; 21:151-60.
Hum Mol Genet. 2009; 18:227-40.
J Biol Chem 2010; 285:20644-53.
Bone 2010; 47:102-10.
Leukemia 2011; 25:538-50.
Human Mutation 2012; 33:29-41

Model of a Biological System



- model is an approximation of the biological system
  - built on first-principles and known hypotheses
    $\Rightarrow$ e.g., elemental reactions, experimental observations, ...
- model is parametrized
  - parameters provide a space for refinement
    $\Rightarrow$ typically quantitative information (e.g., reaction rate)

Systems View of the Cell

- syntax of the systems model is a network:
  - components (nodes) – e.g. chemical substances
  - interactions (edges) – e.g. chemical reactions
- each component is assigned some quantity
  - discrete: number of molecules
  - continuous: molecule concentration in a compartment (solution)
  - can be visualized by color intensity of a node
- dynamics drives the change of node colour intensity in time
  - driven by global rules (e.g., mass-action reactions)

Biological Model Formal Definition

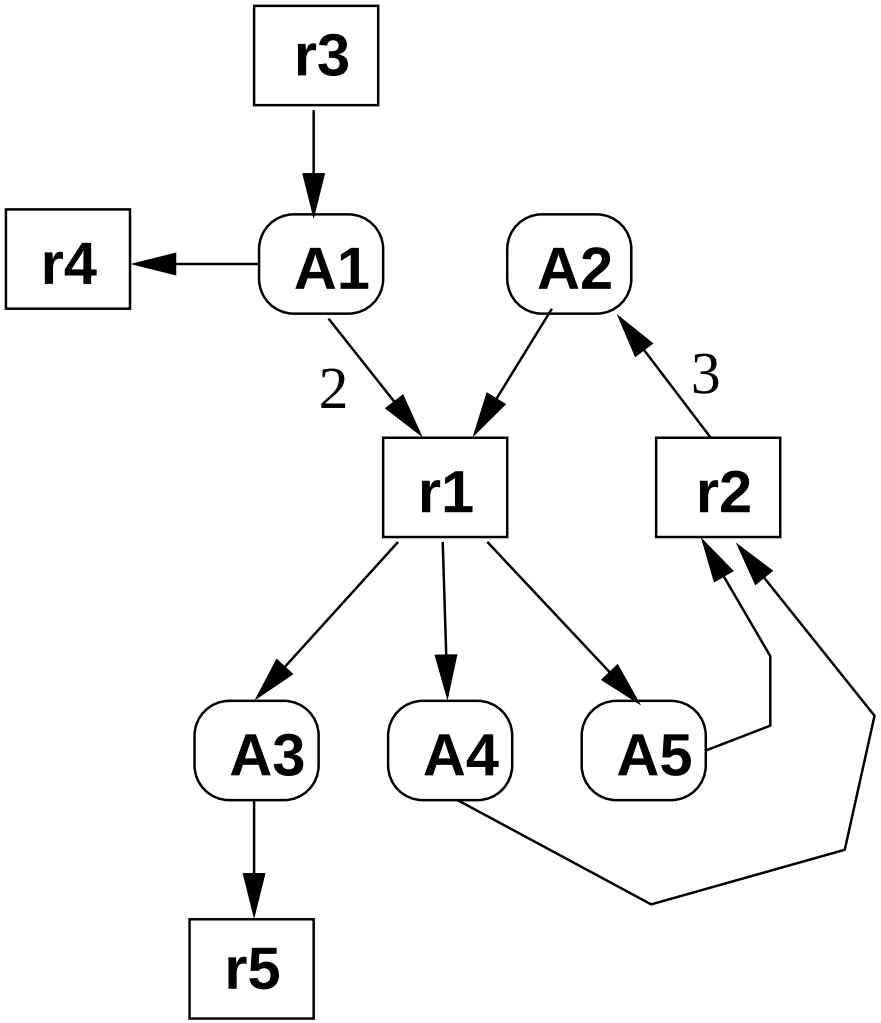
Denote $\mathbb{S}_t = \mathbb{Z}$ domain of stoichiometric coefficients.

Biological model is a tuple $(S, R, \texttt{reanet}, \mathrm{regnet}, \mathrm{map})$, where:

- $S \subset \mathbb{N}$ ... (finite) *species* index set
- $R \subset \mathbb{N}$ ... (finite) *reactions* index set
- $\mathrm{reanet} \subseteq S \times R$ ... *reaction network*
- $\mathrm{regnet} \subseteq S \times R \times \{\texttt{inh}, \texttt{act}\}$ ... *regulatory network*
- $\mathrm{map} : \mathrm{reanet} \to \mathbb{S}_t$ ... *stoichiometric map*

Members of S are denoted: $s_1, s_2, \ldots$.
Members of R are denoted: $r_1, r_2, \ldots$.

Biological Model – Example

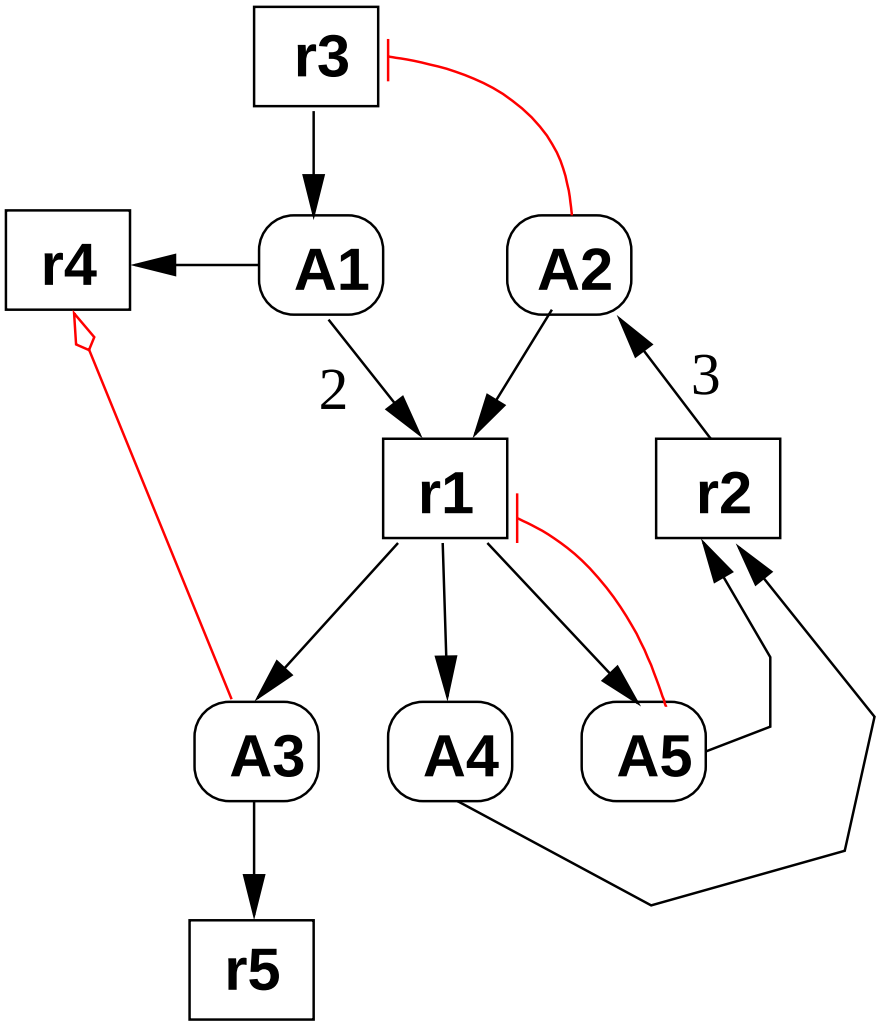


Biological Model – Example

- $S = \{A_1, A_2, A_3, A_4, A_5\}$
- $R = \{r_1, r_2, r_3, r_4, r_5\}$
- $\mathrm{reanet}, \mathrm{map}$:

  (r_1) $2A_1 + A_2 \to A_3 + A_4 + A_5$
  (r_2) $A_4 + A_5 \to 3A_2$
  (r_3) $\to A_1$
  (r_4) $A_1 \to$
  (r_5) $A_3 \to$

Biological Model – Example



Biological Model – Example

- $S = \{A_1, A_2, A_3, A_4, A_5\}$
- $R = \{r_1, r_2, r_3, r_4, r_5\}$
- $\mathrm{reanet}, \mathrm{map}$:

  (r_1) $2A_1 + A_2 \to A_3 + A_4 + A_5$
  (r_2) $A_4 + A_5 \to 3A_2$
  (r_3) $\to A_1$
  (r_4) $A_1 \to$
  (r_5) $A_3 \to$

- regnet : A_2 inhibits r_3, A_3 activates r_4, A_5 inhibits r_1

$$A + B \rightarrow AB$$

- state configuration captures number of molecules:

$$\langle \#[AB], \#[A], \#[B] \rangle \in \mathbb{N}_0^3$$

- global rule:
  - one molecule AB is added to the solution
  - one molecule A is removed from the solution
  - one molecule B is removed from the solution

$$\begin{aligned} \#[AB](t+1) &= \#[AB](t) + 1 \\ \#[A](t+1) &= \#[A](t) - 1 \\ \#[B](t+1) &= \#[B](t) - 1 \end{aligned}$$

Consider three reactions:

$$A \to B$$
$$A + B \to AB$$
$$AB \to A + B$$

- state configuration has the form $\langle \#A, \#B, \#AB \rangle \in \mathbb{N}_0^3$
- consider, e.g., configuration $\langle 2, 2, 1 \rangle$
  $\Rightarrow$ what is the next configuration?
- reactions run in parallel ...

$$A + B \rightarrow AB$$

- continuous state captures concentration of molecules in a certain volume (the solution):

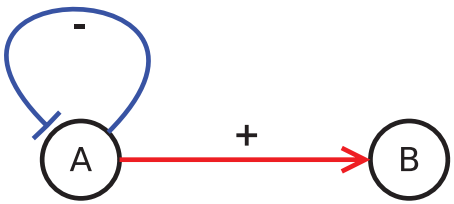
$$\langle [AB], [A], [B] \rangle \in \mathbb{R}^3_+$$

- global rule:
  - a mass of AB outflows from the solution
  - a mass of A inflows to the solution
  - a mass of B inflows to the solution

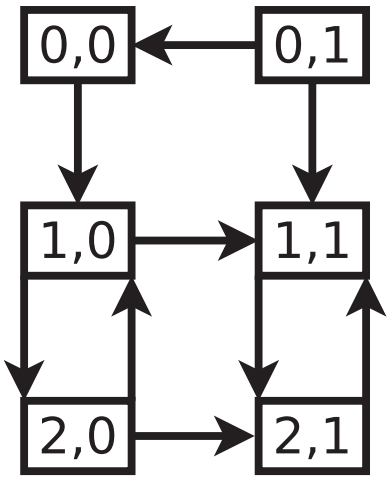
$$\frac{d[AB]}{dt} = v$$
$$\frac{d[A]}{dt} = \frac{d[B]}{dt} = -v$$

where $v = k[A][B]$, k is the *reaction rate constant*.

The law of mass action.



$A \in \{0, 1, 2\},\ B \in \{0, 1\}$
$t_{AA} = 2,\ t_{AB} = 1$
$K_{A,\emptyset} = 2$
$K_{A,\{A\}} = 0$
$K_{B,\emptyset} = 0$
$K_{B,\{A\}} = 1$



- introduced by René Thomas [1973]
- refined by Chaouiya et al. [2003]

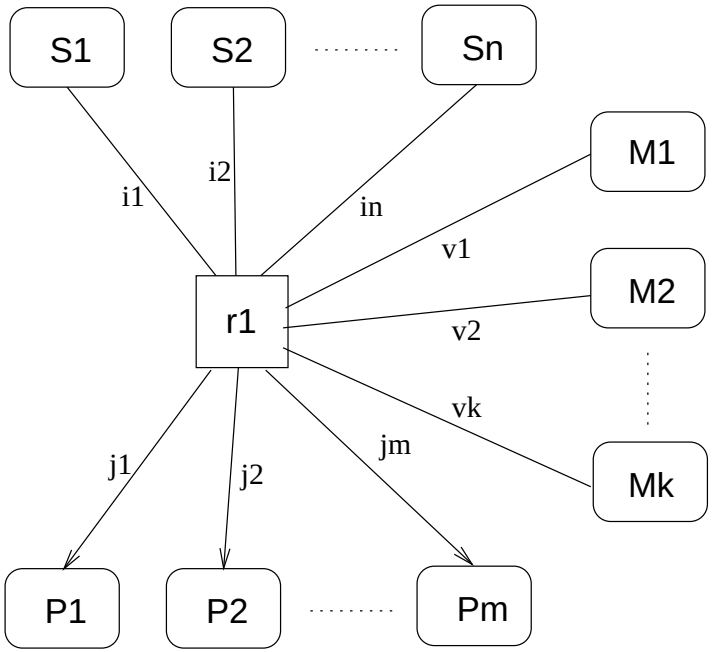
Model Semantics

- species S are interpreted as model **variables**
  - boolean models: $val(S_i) \in \{\text{present}, \text{absent}\}$
  - discrete-value models: $val(S_i) \in \mathbb{N}_0$
  - continuous-value models: $val(S_i) \in \mathbb{R}_0^+$
- current values of all model variables make the **state**
- reaction is interpreted as a *rule* that affects (changes) the state

Note

Variables are always considered bounded (maximal values can be given by physical limits, e.g., the cell volume).

Model Semantics



- $v_i \in \{inh, act\}$
- $\mathrm{S}_i \in S$ – *reactants*, $\mathrm{P}_i \in S$ – *products*, $\mathrm{M}_i \in S$ – *modifiers*
- reaction *rule* (locally) affects the variable values according to the stoichiometric map:
  $\Rightarrow \mathrm{S}_l$ decreased by i_l, P_l increased by j_l, M_l not affected

Rule Interpretation

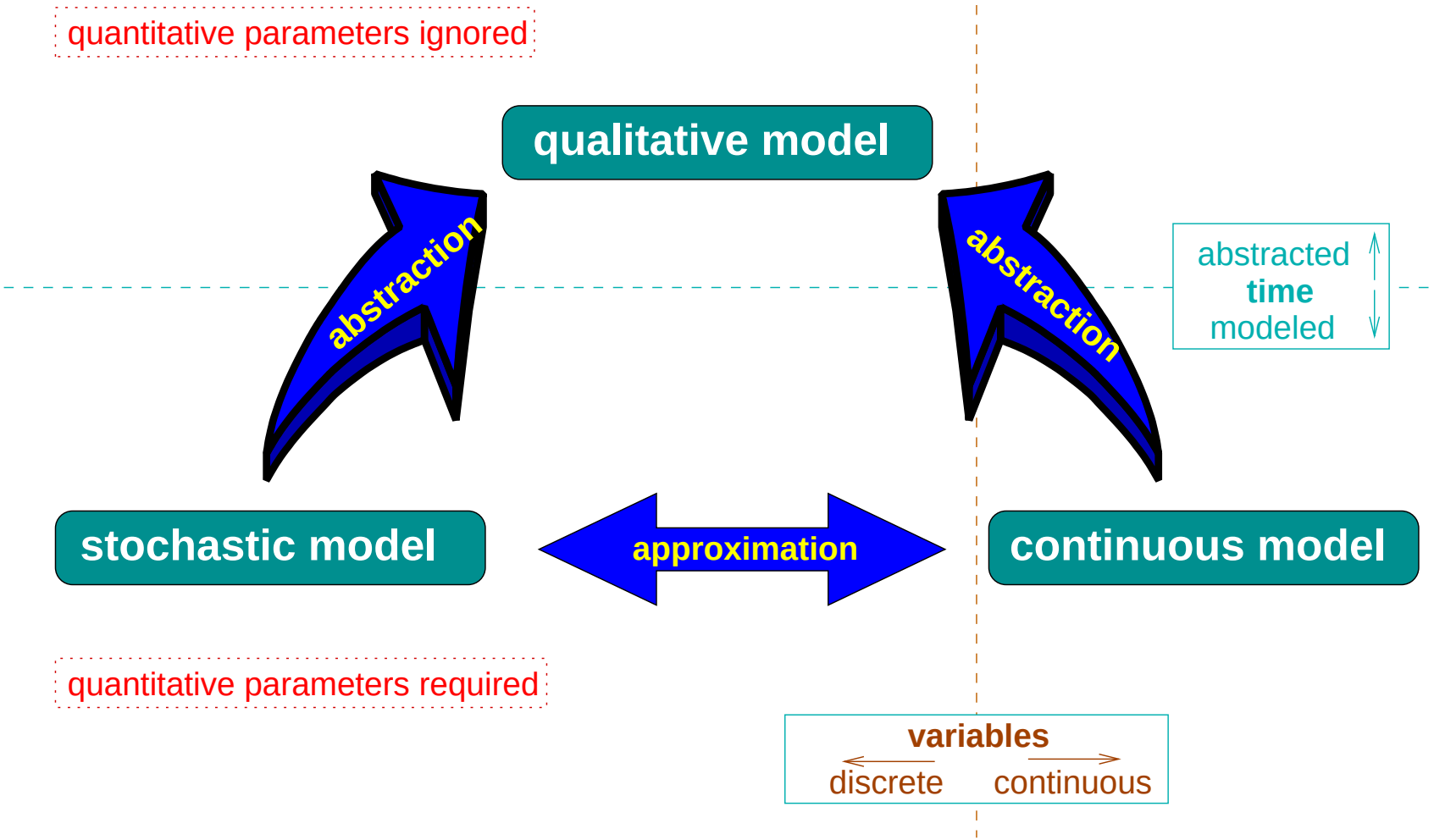
Modelling of time

- exact time of reaction occurence
  $\Rightarrow$ continuous-time models
- time of reaction occurence abstracted
  $\Rightarrow$ discrete-time models (ticked or untimed)

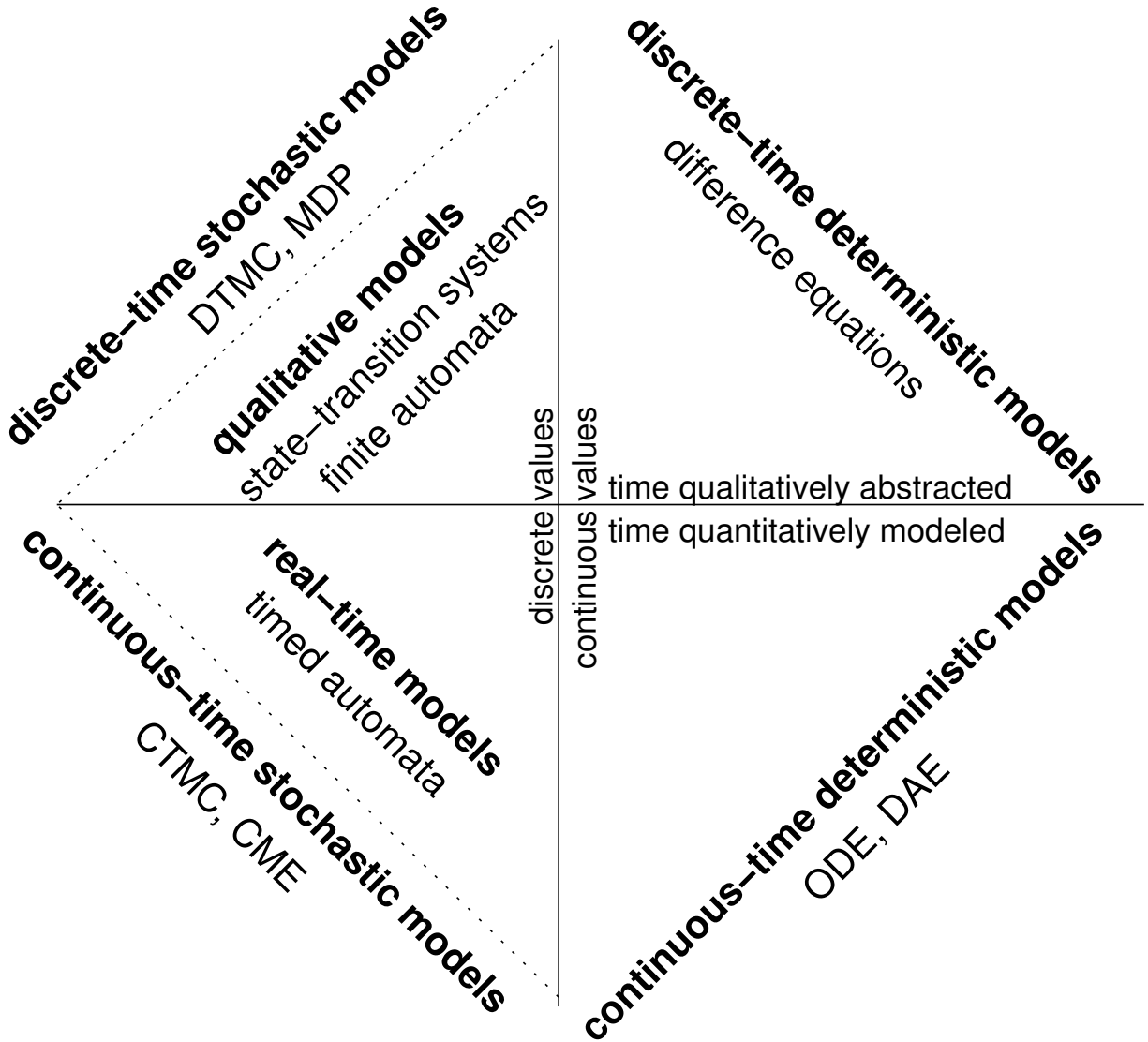
Modelling of noise

- deterministic rules – noise absent (large populations)
  $\Rightarrow$ always one possible execution under the same conditions
- stochastic rules – noise present (small populations)
  $\Rightarrow$ many different executions possible under the same conditions

Model Semantics Spectrum – Brief



Model Semantics Spectrum – Detailed

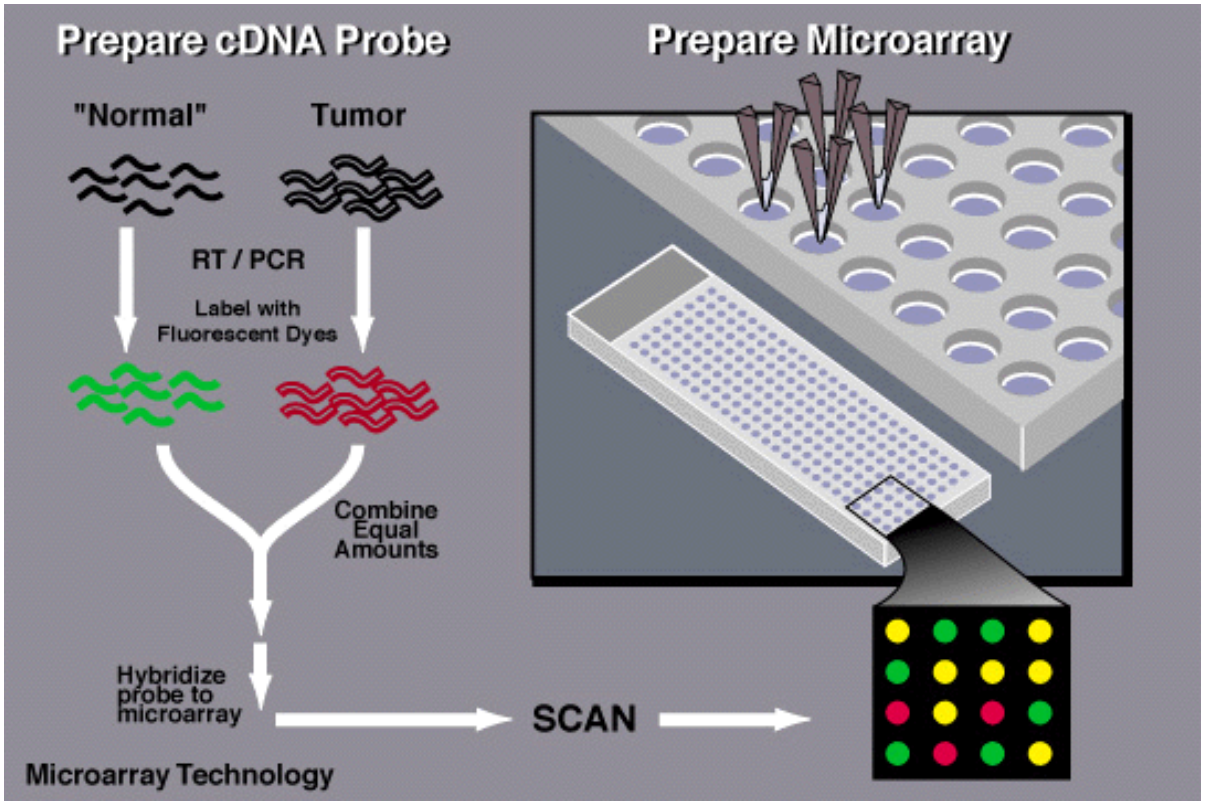


Biological Hypotheses as Temporal Properties

- wet-lab measurements $\Rightarrow$ time-series data
  - low resolution – e.g., microarray data, series of western blots
  - high resolution – fluorescence measurements (e.g., gene reporters)
  - most typically population-level measurements (average behaviour)
- literature provides other constraints on system dynamics
  - e.g., multiple steady states, species concentration correlation, ...
- all can be formally captured by means of temporal logics

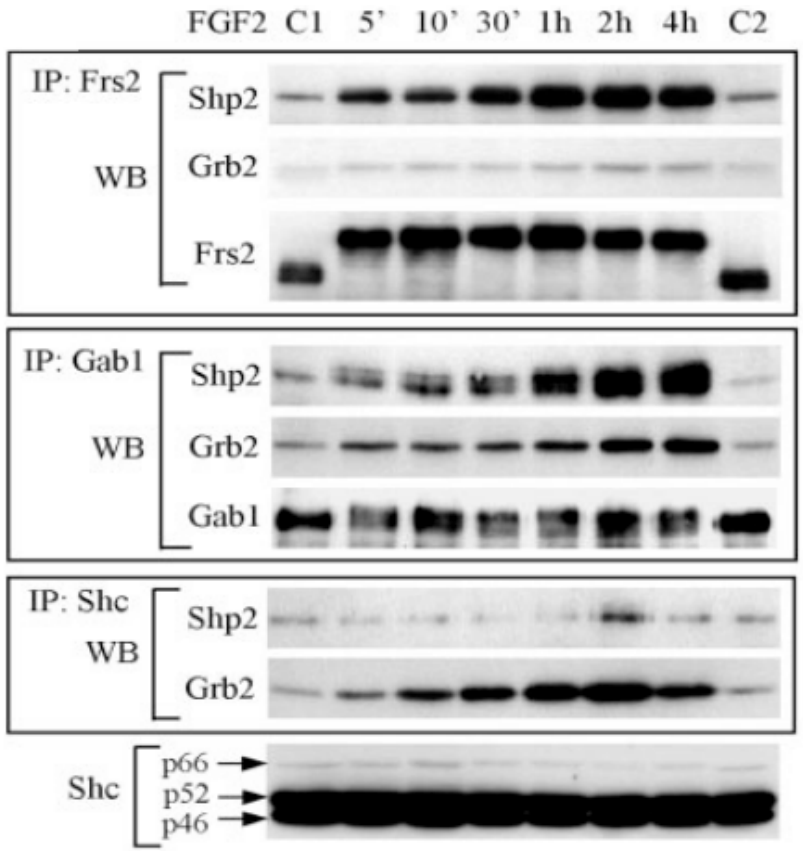
Experimental Measurements of Regulatory Dynamics

- systems measurements of transcriptome (mRNA concentration)
- very imprecise!



Experimental Measurements of Regulatory Dynamics

- western blots
- measurements of protein binding (presence of certain proteins)



Experimental Measurements of Regulatory Dynamics

1. Model is built on first-principles
   $\Rightarrow$ purely qualitative (network topology)
2. To build the model we need to find all possible constraints that can be formulated.
   $\Rightarrow$ static and dynamic constraints (properties)
3. Fitting is not enough, some data are too imprecise to be fittable.

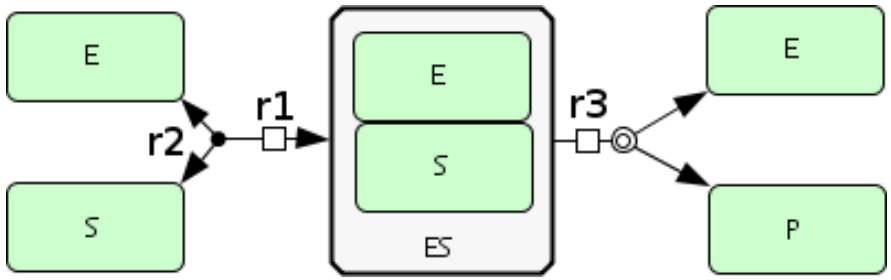
Qualitative vs. quantitative temporal properties

- qualitative properties (LTL, CTL)
  - modalities (possibilities/necessities in future behaviour)
  - reachability of particular (sets of) states
  - temporal ordering of events, monotonicity
  - temporal correlations of model variables
  - stability (attractors, basins of attraction)
- quantitative properties
  - deterministic (MTL, MITL, STL)
    - enhance modalities with (dense) time information
    - exact timing of events, time-bounds
  - stochastic (PLTL, PCTL, CSL)
    - probability of property satisfaction
    - stochasticity combined with time

Qualitative vs. quantitative temporal properties

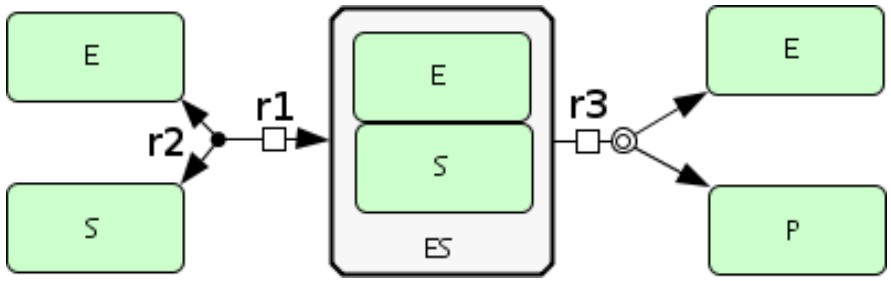
- qualitative properties (LTL, CTL)
  - modalities (possibilities/necessities in future behaviour)
  - reachability of particular (sets of) states
  - temporal ordering of events, monotonicity – time-series
  - temporal correlations of model variables – time-series
  - stability (attractors, basins of attraction)
- quantitative properties
  - deterministic (MTL, MITL, STL)
    - enhance modalities with (dense) time information
    - exact timing of events, time-bounds
  - stochastic (PLTL, PCTL, CSL)
    - probability of property satisfaction
    - stochasticity combined with time

Temporal Property Examples
Qualitative properties



- enzyme E is never permanently exhausted
  $\mathbf{GF}(E > 0)$
- all molecules of the substrate S are finally transfered to the product P provided that the final state is stable
  $S == 5 \Rightarrow \mathbf{FG}(P == 5 \wedge S == 0)$
- enzyme E is used and finally returned back
  $(E \geq 2)\ \mathbf{U}\ [(0 < E < 2)\ \mathbf{U}\ (E \geq 2)]$

Temporal Property Examples
Quantitative properties

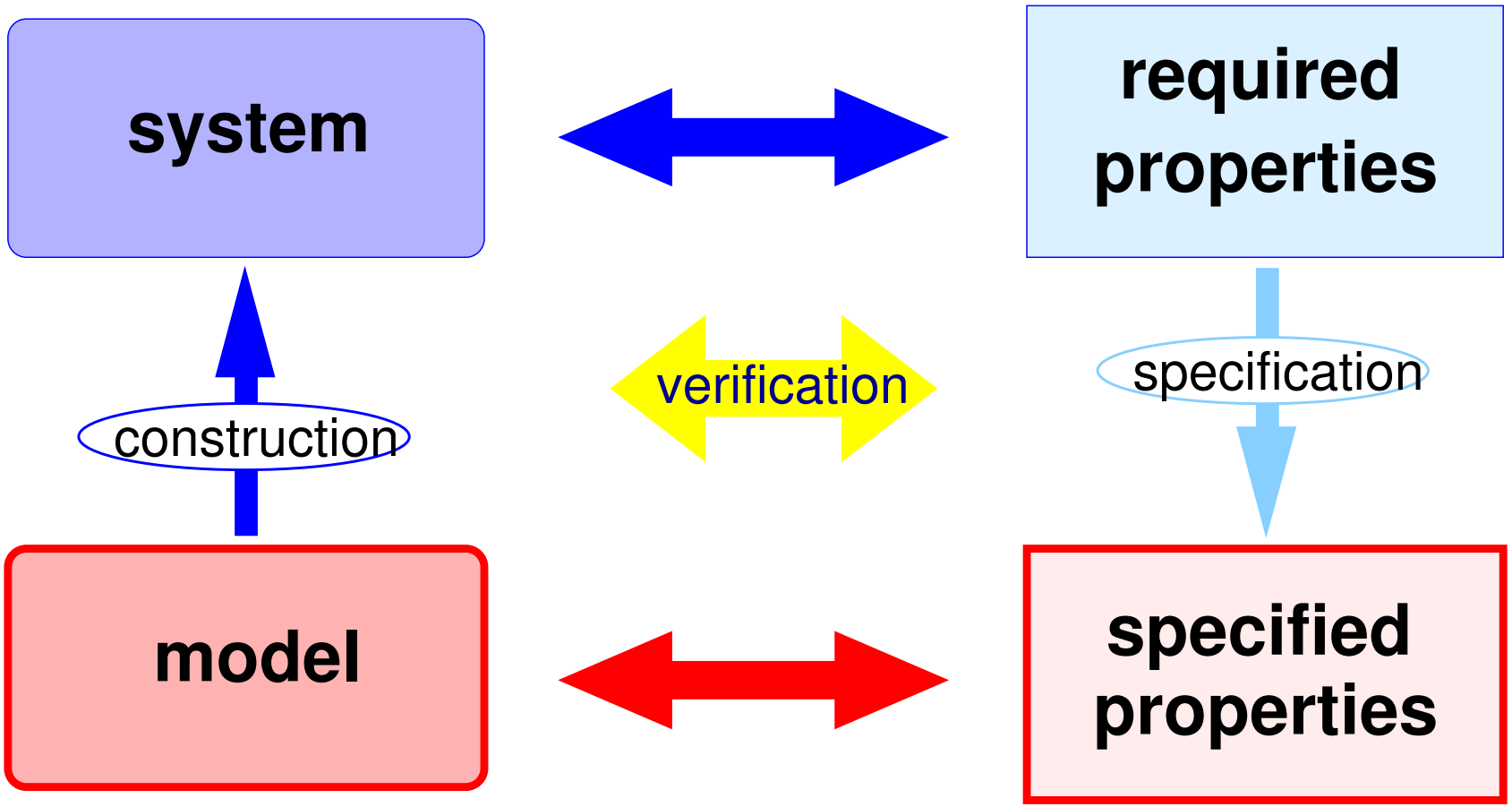


- in the first 10 time units, enzyme E cannot permanently exhausted
  $\mathbf{G}^{[0,10]}\mathbf{F}(E > 0)$
- all molecules of the substrate S are transfered to the product P minimally in 2 and maximally in 5 time units provided that the final state is stable
  $S == 5 \Rightarrow \mathbf{F}^{[2,5]}\mathbf{G}(P == 5 \wedge S == 0)$
- enzyme E is used and finally returned back within the given time intervals
  $(E \geq 2)\ \mathbf{U}^{[1,2]}\ [(0 < E < 2)\ \mathbf{U}^{[1,2]}\ (E \geq 2)]$

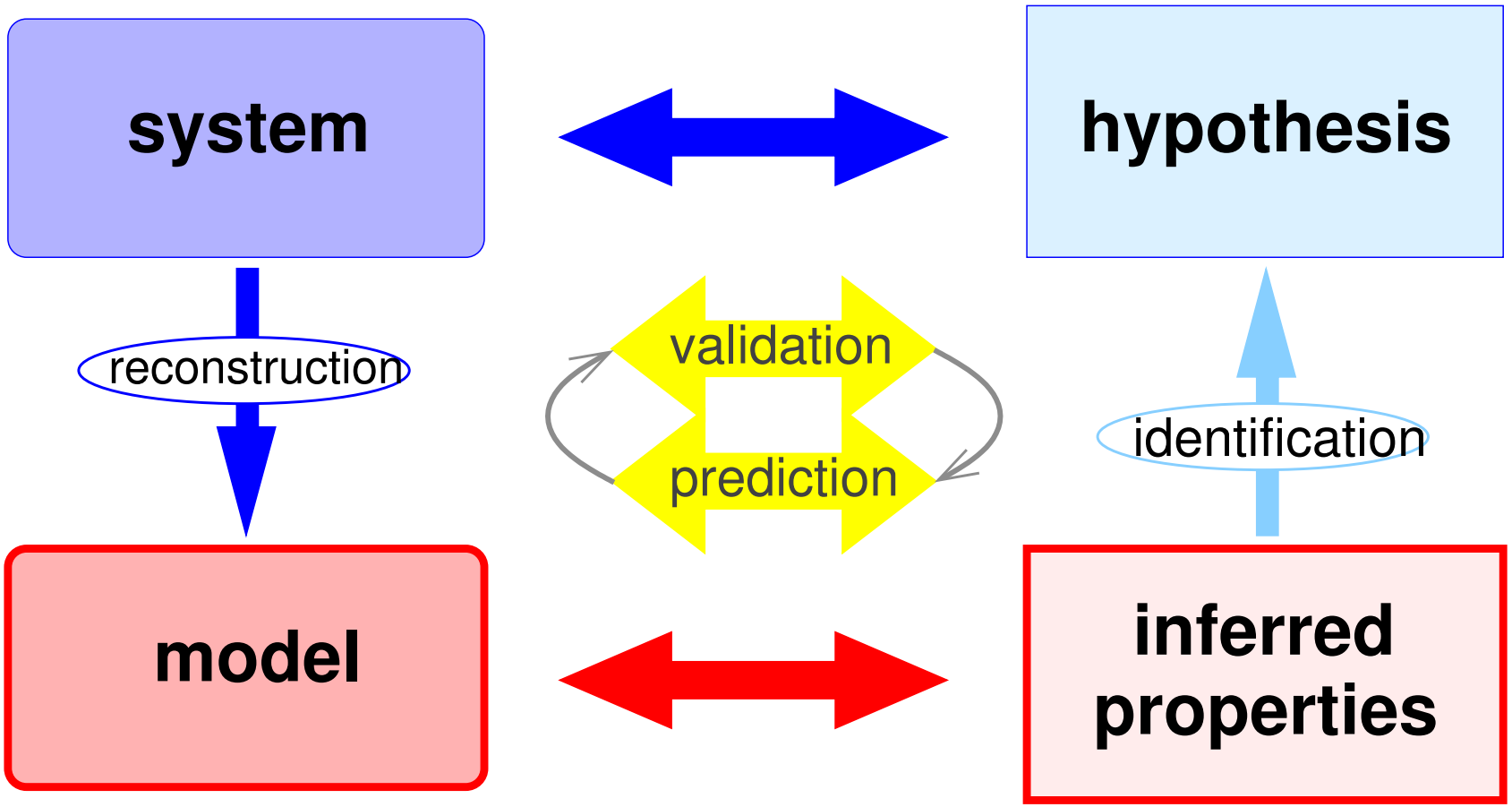
Temporal Property Examples

- oscillation
  LTL: $(\mathbf{G}[(A \leq 3) \Rightarrow \mathbf{F}(A > 3)]) \wedge (\mathbf{G}[(A > 3) \Rightarrow \mathbf{F}(A \leq 3)])$
- bistability
  CTL: $\mathbf{EFAG}(A \leq 5) \wedge \mathbf{EFAG}(A > 5)$
- probabilistic modality
  PCTL: $\mathbf{P}_{\geq 0.9}[\mathbf{F}^{\leq 5}(A = 3)]$
- probabilistic modality with time
  CSL: $\mathbf{P}_{\geq 0.9}[\mathbf{F}^{[1,2]}(A = 3)]$

System Construction and Formal Methods



Knowledge Discovery and Formal Methods



Outline

1. Introduction
2. LTL Model Checking
3. Parallel LTL Model Checking
4. Discrete Abstraction of ODE Models
5. Case Studies
   - Model Checking of E. Coli Ammonium Transport
   - Parameter Synthesis by Model Checking
   - Parameter Synthesis and Classification for Boolean Networks

Kripke Structure

Definition

Let AP be the set of *atomic propositions* (logical expressions over model variables, typical inequalities). *Kripke structure* is the quadruple $K = \langle S, S_0, T, L \rangle$ where:

- S is the finite set of states
- $S_0 \subseteq S$ is the set of inititial states
- $T \subseteq S \times S$ such that $\forall s \in S, \exists s' \in S : \langle s, s' \rangle \in T$
- L is the labeling $L : S \to 2^{AP}$

Kripke structure – properties

- for a state $s \in S$, $L(s)$ represents the set of all atomic propositions satisfied in s
- unfolding of the Kripke structure from any initial state is always an infinite-depth tree
  - maximal paths in the unfolding represent individual (infinite) executions of the Kripke structure

Linear-time Temporal Logic – syntax

Let AP be the set of atomic propositions. Formula φ is *linear temporal logic (LTL) formula* iff the following holds:

- $\varphi = p$ for any $p \in AP$
- If φ_1 and φ_2 LTL formulae then:
  - $\neg\varphi_1$, $\varphi_1 \wedge \varphi_2$ and $\varphi_1 \vee \varphi_2$ are LTL formulae
  - $\mathbf{X}\varphi_1$, $\mathbf{F}\varphi_1$ a $\mathbf{G}\varphi_1$ are LTL formulae
  - $\varphi_1 \mathbf{U} \varphi_2$ a $\varphi_1 \mathbf{R} \varphi_2$ are LTL formulae

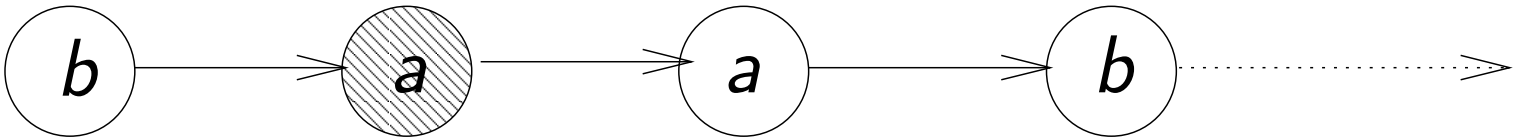
Linear Temporal Logic – semantics

Let $\pi = s_0, s_1, ..., s_i, ...$ be an infinite sequence of states (a path) in a Kripke structure K. For $j > 0$ we denote π^j the suffix $s_j, s_{j+1}, ..., s_i, ...$ Satisfiability relation $\models$ is defined by induction:

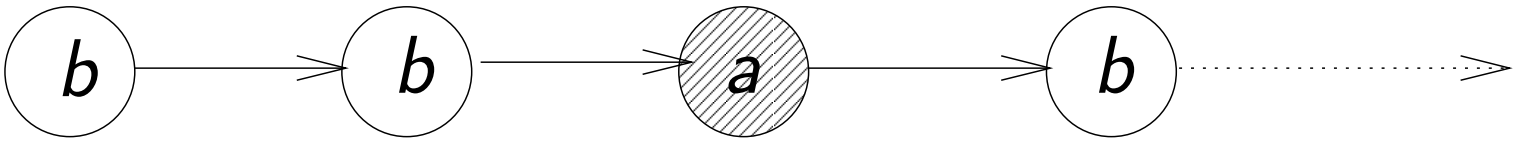
- $\pi \models p$ iff $p \in L(s_0)$
- $\pi \models \neg\varphi$ iff $\pi \not\models \varphi$
- $\pi \models \varphi_1 \wedge \varphi_2$ iff $\pi \models \varphi_1$ and $\pi \models \varphi_2$
- $\pi \models \varphi_1 \vee \varphi_2$ iff $\pi \models \varphi_1$ or $\pi \models \varphi_2$
- $\pi \models \mathbf{X}\varphi$ iff $\pi^1 \models \varphi$
- $\pi \models \mathbf{F}\varphi$ iff $\exists i \geq 0. \pi^i \models \varphi$
- $\pi \models \mathbf{G}\varphi$ iff $\forall i \geq 0. \pi^i \models \varphi$
- $\pi \models \varphi_1 \mathbf{U} \varphi_2$ iff $\exists j \geq 0. \pi^j \models \varphi_2$ and $\forall i < j. \pi^i \models \varphi_1$
- $\pi \models \varphi_1 \mathbf{R} \varphi_2$ iff $\forall j \geq 0, \forall 0 \leq i < j. \pi^i \not\models \varphi_1 \Rightarrow \pi^j \models \varphi_2$.

Linear Temporal Logic – semantics

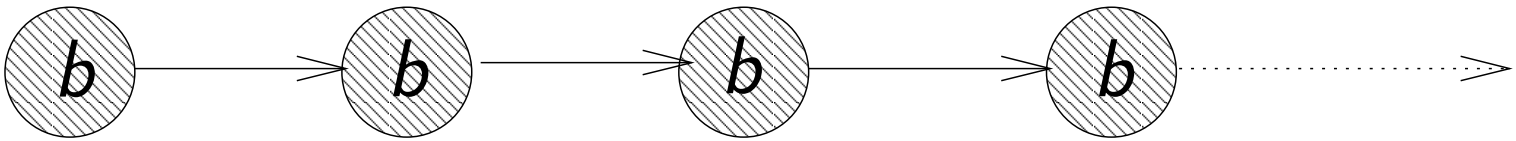
X*a*



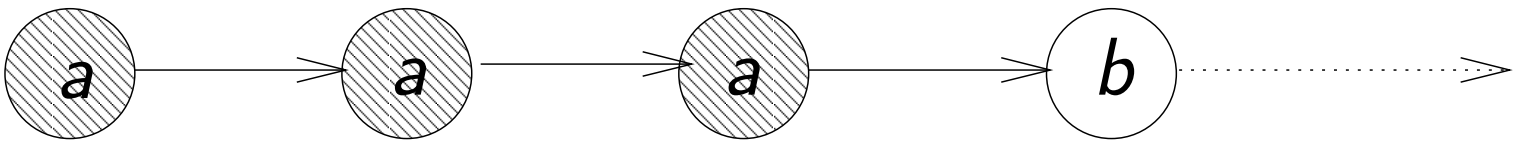
F*a*



G*b*



*a***U***b*



Linear Temporal Logic – semantics

For any formulae φ_1, φ_2 the following holds:
$\neg \mathbf{F}\varphi \equiv \mathbf{G}\neg\varphi$
$\neg(\varphi_1 \mathbf{U} \varphi_2) \equiv \neg\varphi_1 \mathbf{R} \neg\varphi_2$

The full expressiveness is achieved by using just the operators $\neg, \wedge, \mathbf{X}, \mathbf{U}$.

Linear Temporal Logic – semantics

For any formulae φ_1, φ_2 the following holds:
$\neg \mathbf{F}\varphi \equiv \mathbf{G}\neg\varphi$
$\neg(\varphi_1 \mathbf{U} \varphi_2) \equiv \neg\varphi_1 \mathbf{R} \neg\varphi_2$

The full expressiveness is achieved by using just the operators $\neg, \wedge, \mathbf{X}, \mathbf{U}$.

LTL formulae are most typically interpreted universally over Kripke structure paths:

Linear Temporal Logic – semantics

For any formulae φ_1, φ_2 the following holds:
$\neg \mathbf{F}\varphi \equiv \mathbf{G}\neg\varphi$
$\neg(\varphi_1 \mathbf{U} \varphi_2) \equiv \neg\varphi_1 \mathbf{R} \neg\varphi_2$

The full expressiveness is achieved by using just the operators $\neg, \wedge, \mathbf{X}, \mathbf{U}$.

LTL formulae are most typically interpreted universally over Kripke structure paths:

Kripke structure as a model for a formula

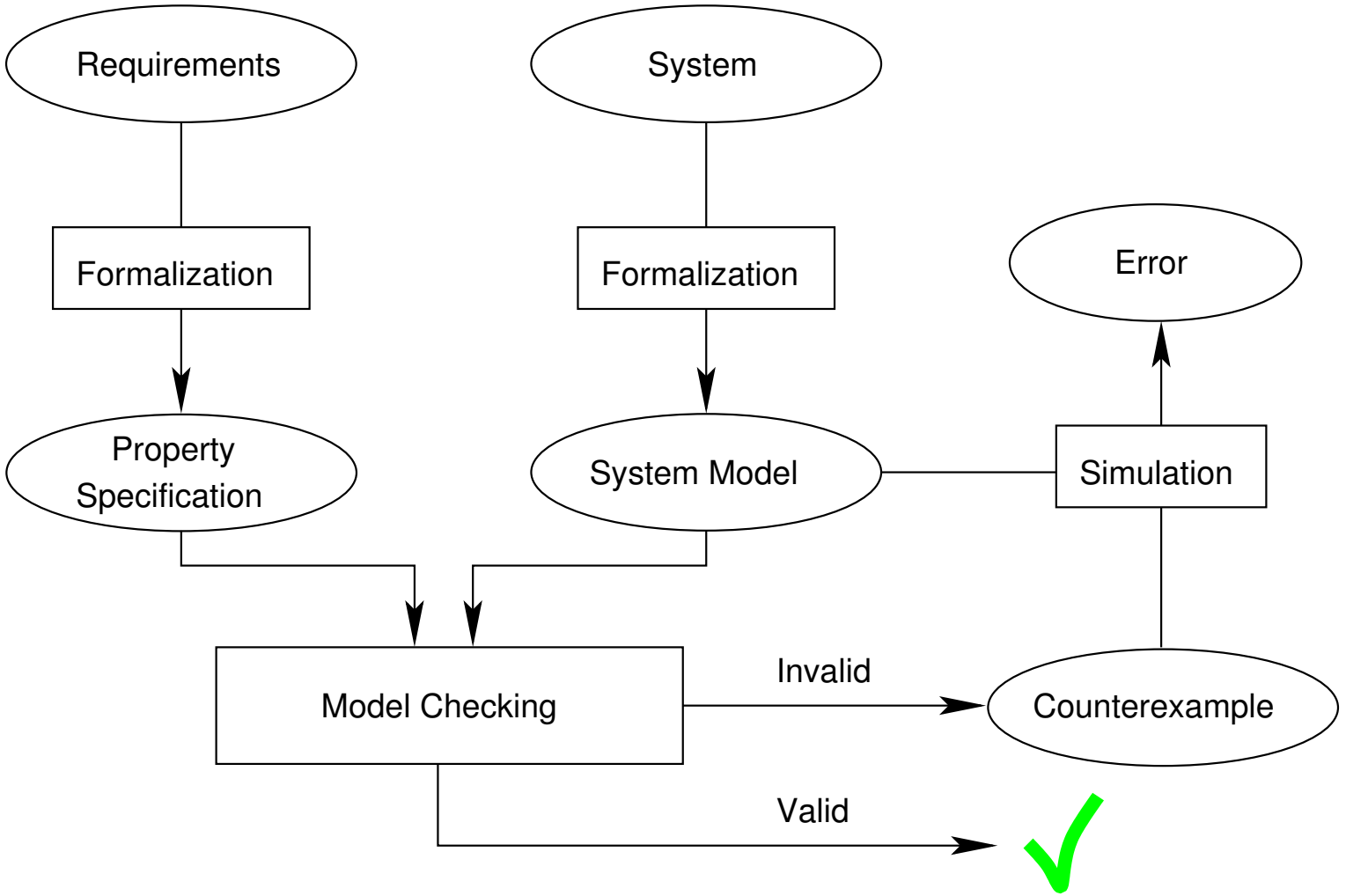
Let K be a Kripke structure. A formula φ is satisfied by K, $K \models \varphi$ iff for each execution $\pi = s_0, \ldots$ such that $s_0 \in S_0$ it holds $\pi \models \varphi$.

Model checking

Model Checking Problem

Model checking problem is to deside for a given Kripke structure K and a temporal property Φ the problem $K \models \Phi$.
If the result is negative, a path π such that $\pi \not\models \Phi$ is returned (a so-called *counterexample*).

Model-Checking Overview



Büchi Automaton

Definition

Büchi automaton is the tuple $A = (S, \Sigma, S_0, \delta, F)$ where

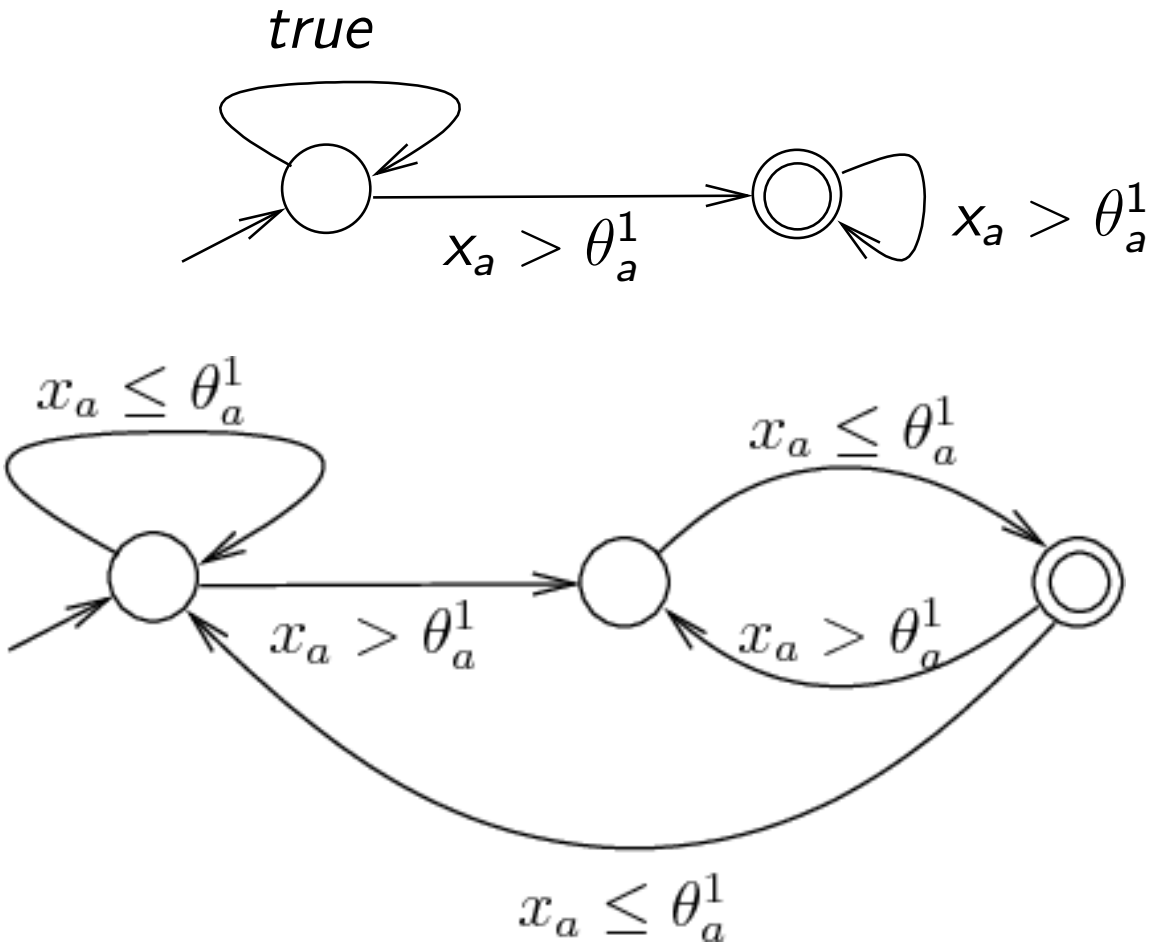
- Σ is the finite set of symbols,
- S is the finite set of states,
- $S_0 \subseteq S$ is the set of initial states,
- $\delta : S \times \Sigma \to 2^S$ is the transition relation,
- $F \subseteq S$ is the set of final (accepting) states.

Büchi Automaton

Language accepted by a Büchi automaton

- (infinite) *run* of an automaton A over an infinite word $w = a_1 a_2 ...$ is the sequence of states $\rho = s_0, s_1, ...$ such that $\forall i : s_i \in \delta(s_{i-1}, a_i)$
- $inf(\rho)$ – the set of states that occur infinitely often in ρ,
- a run ρ is accepting iff $inf(\rho) \cap F \neq \emptyset$
- $\mathcal{L}(A)$ denotes the so-called ω-regular language accepted by A, the set of all (infinite) words for which there exist a corresponding accepting run of A,
- ω-regular languages are closed under complementation.

Büchi automata examples



LTL Model Checking

LTL Model Checking

- Specification formalized as LTL formula

Automata-based approach to LTL model checking

- Employs Büchi automata to express
  - all paths of the Kripke structure under consideration
  - all paths violating the specification
- Model satisfies the specification if the intersection of the sets is empty, i.e., if the synchronized Büchi automaton accepts empty language.

LTL model checking problem is reduced to the detection of accepting cycles in the graph of a Büchi automaton.

Model Checking as a language inclusion problem

Interpretation of a path $\pi = s_0, s_1, \ldots$ in a Kripke structure K is a sequence of sets of APs:

$$L(\pi) = L(s_0), L(s_1), \ldots$$

Problem

For a given Kripke structure $K = (S, S_0, T, L)$ and a given LTL formula φ decide $K \models \varphi$.

Reformulation

Let $\Sigma = 2^{AP}$. Consider two languages of infinite words:

1. $\mathcal{L}_K = \{L(\pi) \mid \pi \text{ is a path in } K\}$
2. $\mathcal{L}_\varphi = \{L(\pi) \mid \pi \models \varphi\}$

Then $K \models \varphi$ iff $\mathcal{L}_K \subseteq \mathcal{L}_\varphi$.

Kripke structure as a Büchi automaton

Claim

For each Kripke structure $K = (S, S_0, T, L)$ we can construct a Büchi automaton A_K such that $\mathcal{L}_K = \mathcal{L}(A_K)$:

- $A_K = (S, 2^{AP}, S_0, \delta, S)$
  where $q \in \delta(p, a) \Leftrightarrow (p, q) \in T \wedge L(p) = a$.

Observation

Note that $F = S$ (the set of final states coincides with the state space).

LTL formula as a Büchi automaton

Theorem [Vardi, Wolper 1986]

For each LTL formula φ there exists (and can be effectively constructed) a Büchi automaton A_φ such that $\mathcal{L}_\varphi = \mathcal{L}(A_\varphi)$.

Construction goes through a generalized BA (extended in the acceptance condition – a system of accepting states sets, requirement to infinitely often visit all accepting sets). Complexity is $2^{\mathcal{O}(n)}$ where n is the size of the formula. There exist many algorithms – check, e.g., `http://spot.lip6.fr/wiki/`.

Note

LTL is less expressive then BAs.

Synchronous Product

Claim

Let $A = (S_A, \Sigma, S_{0_A}, \delta_A, S_A)$, $B = (S_B, \Sigma, S_{0_B}, \delta_B, F_B)$ be Büchi automata, and $F_A = S_A$. Then a Büchi automaton $A \times B$ that accepts the language $L(A \times B) = L(A) \cap L(B)$ can be constructed in the following way:

- $A \times B = (S_A \times S_B, \Sigma, S_{0_A} \times S_{0_B}, \delta_{A \times B}, S_A \times F_B)$,
- $(p', q') \in \delta_{A \times B}((p, q), a)$ for all $p' \in \delta_A(p, a)$ and $q' \in \delta_B(q, a)$.

Model Checking reduced to language emptyness problem

Claim

For each LTL formula φ it holds that $co\text{-}\mathcal{L}(A_\varphi) = \mathcal{L}(A_{\neg\varphi})$.

- $K \models \varphi \Leftrightarrow \mathcal{L}_K \subseteq \mathcal{L}_\varphi$
- $K \models \varphi \Leftrightarrow L(A_K) \subseteq L(A_\varphi)$
- $K \models \varphi \Leftrightarrow L(A_K) \cap co\text{-}\mathcal{L}(A_\varphi) = \emptyset$
- $K \models \varphi \Leftrightarrow L(A_K) \cap L(A_{\neg\varphi}) = \emptyset$
- $K \models \varphi \Leftrightarrow (L(A_K) \times L(A_{\neg\varphi})) = \emptyset$

Model Checking as an accepting cycle detection problem

Claim

- A Büchi automaton $A = (S, \Sigma, S_0, \delta, F)$ accepts a nonempty language iff there exist states $s \in F$, $s_0 \in S_0$, and the words $w_1, w_2 \in \Sigma^*$ such that $s \in \hat{\delta}(s_0, w_1)$ and $s \in \hat{\delta}(s, w_2)$.
- In other words, the graph of the automaton contains a reachable accepting cycle.

Model Checking Procedure

1. construct $(A_K \times A_{\neg\varphi})$
2. detect if there is any accepting cycle
3. If accepting cycle found then $K \not\models \varphi$.
4. If accepting cycle not found then $K \models \varphi$.

Accepting cycle detection

Input

- Product automaton represented by three functions:
  - $init()$ – returns the initial states
  - $succs(s)$ – returns the direct successors of $s \in S$
  - $accept(s)$ – decides whether $s \in S$ is accepting

Output

- The answer YES/NO.
- A counterexample if the answer is NO.

$$\pi = \pi_1 \cdot (\pi_2)^\omega$$

where

- $\pi_1 = s_0, s_1, ..., s_k$
- $\pi_2 = s_{k+1}, s_{k+2}, ..., s_{k+n}$ where $s_k \equiv s_{k+n}$
  $\Rightarrow$ a so-called lasso shape.

Accepting cycle detection

Nested DFS algorithm

- Performs two depth-first searches on the graph:
  - 1st identifies reachable accepting states,
  - 2nd test each accepting state for self-reachability.
- Search procedures must interleave in a particular way.
- 2nd (nested) procedure is started from an accepting state, when the 1st procedure backtracks from it (DFS postorder).

Outline

1. Introduction
2. LTL Model Checking
3. Parallel LTL Model Checking
4. Discrete Abstraction of ODE Models
5. Case Studies
   - Model Checking of E. Coli Ammonium Transport
   - Parameter Synthesis by Model Checking
   - Parameter Synthesis and Classification for Boolean Networks

Parallel Model Checking

Observation

- The complexity of biological models is continuously growing with the grand challenge of systems biology to integrate the partial models.
- A solution is to employ all the power of suitable contemporary HW platforms — parallelization.

Problem

- Computing DFS-postorder is inherently sequential problem.
- Optimal parallel and scalable algorithm for computing DFS-postorder is unknown (and unlikely to exist).
- **Nested DFS cannot efficiently use parallel hardware.**

New Algorithms for LTL Model Checking

Nested DFS algorithm

- Optimal, but unusable for parallel HW architectures.

Other optimal algorithms

- Variants of Tarjan's SCC decomposition
- Suffer from the same DFS-postorder problem.

Desired algorithms

- Must be independent of DFS-postorder exploration.
- Must outperform DFS-postorder algorithms on new HW.
- But need not exhibit optimal complexity in general.

Algorithm OWCTY

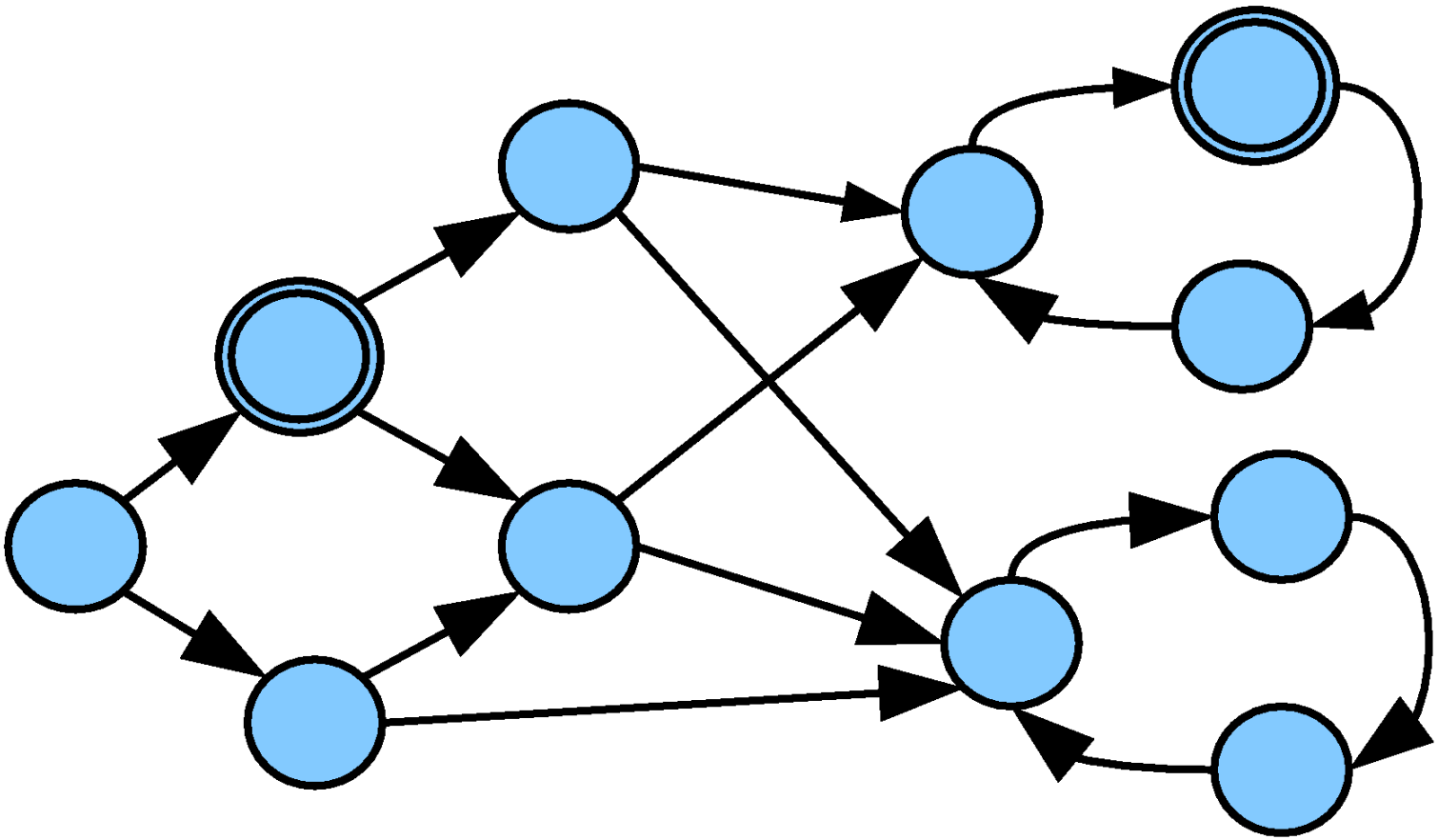
Idea

- Remove states that cannot lie on an accepting cycle.
- A state cannot be part of an accepting cycle if
  - it is unreachable from an accepting state,
  - it has no immediate predecessor.

Realization

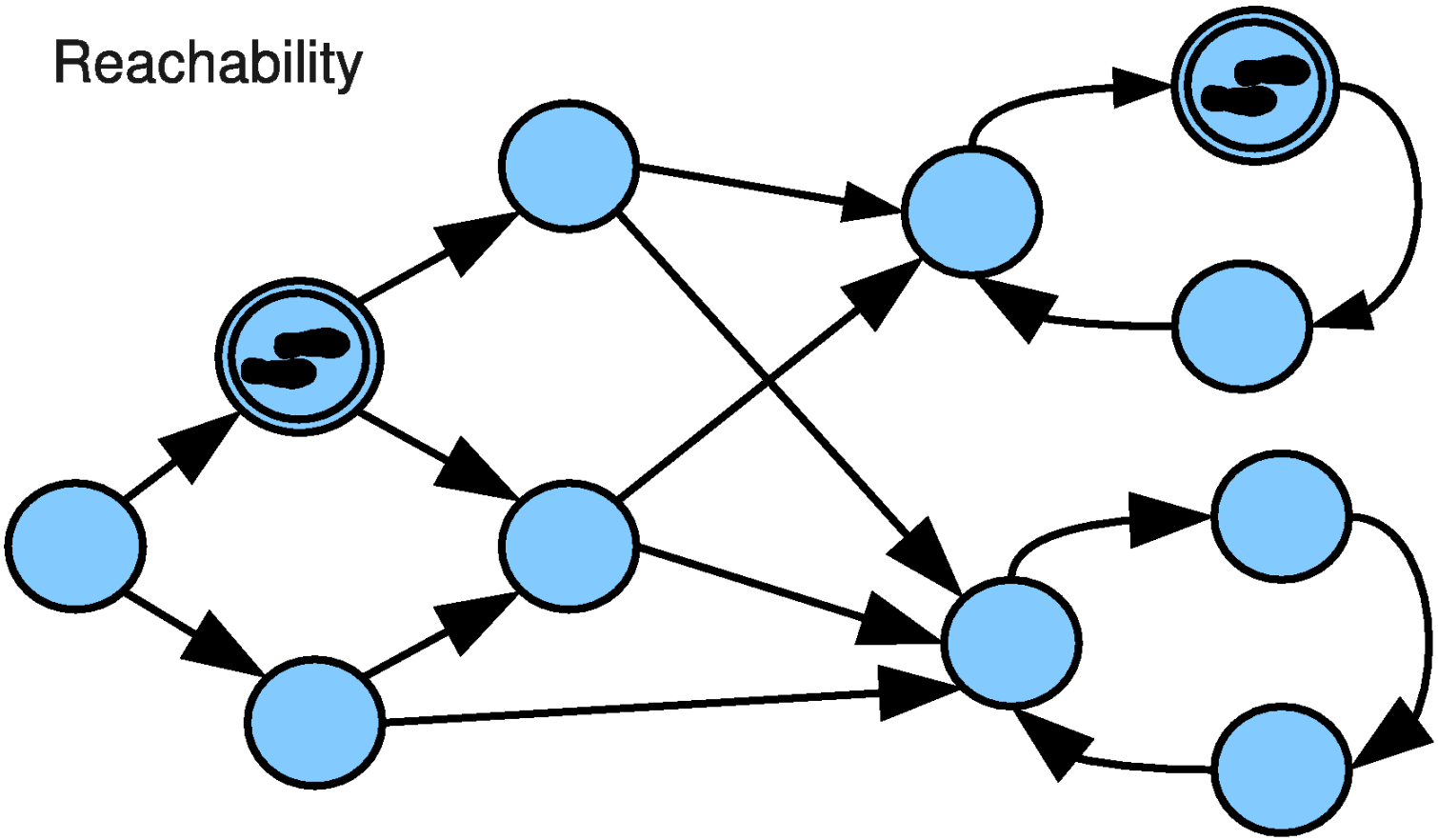
- **Parallel** removal procedures
  - Reachability
  - Elimination
- Repeated application of removal procedures until no state can be removed (fix-point).
- Non-empty graph indicates presence of accepting cycle.

OWCTY Demonstration



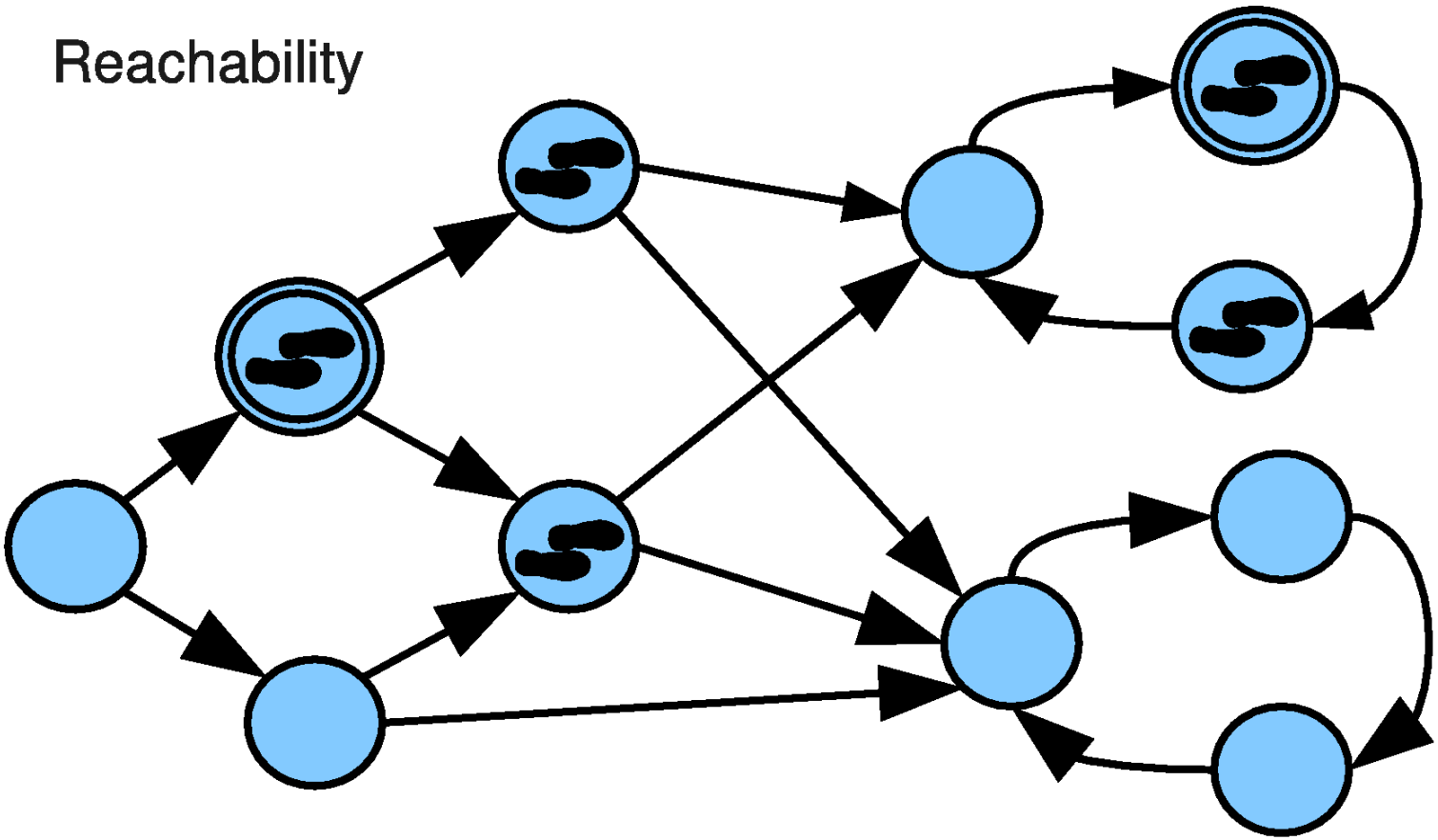
OWCTY Demonstration

1st iteration



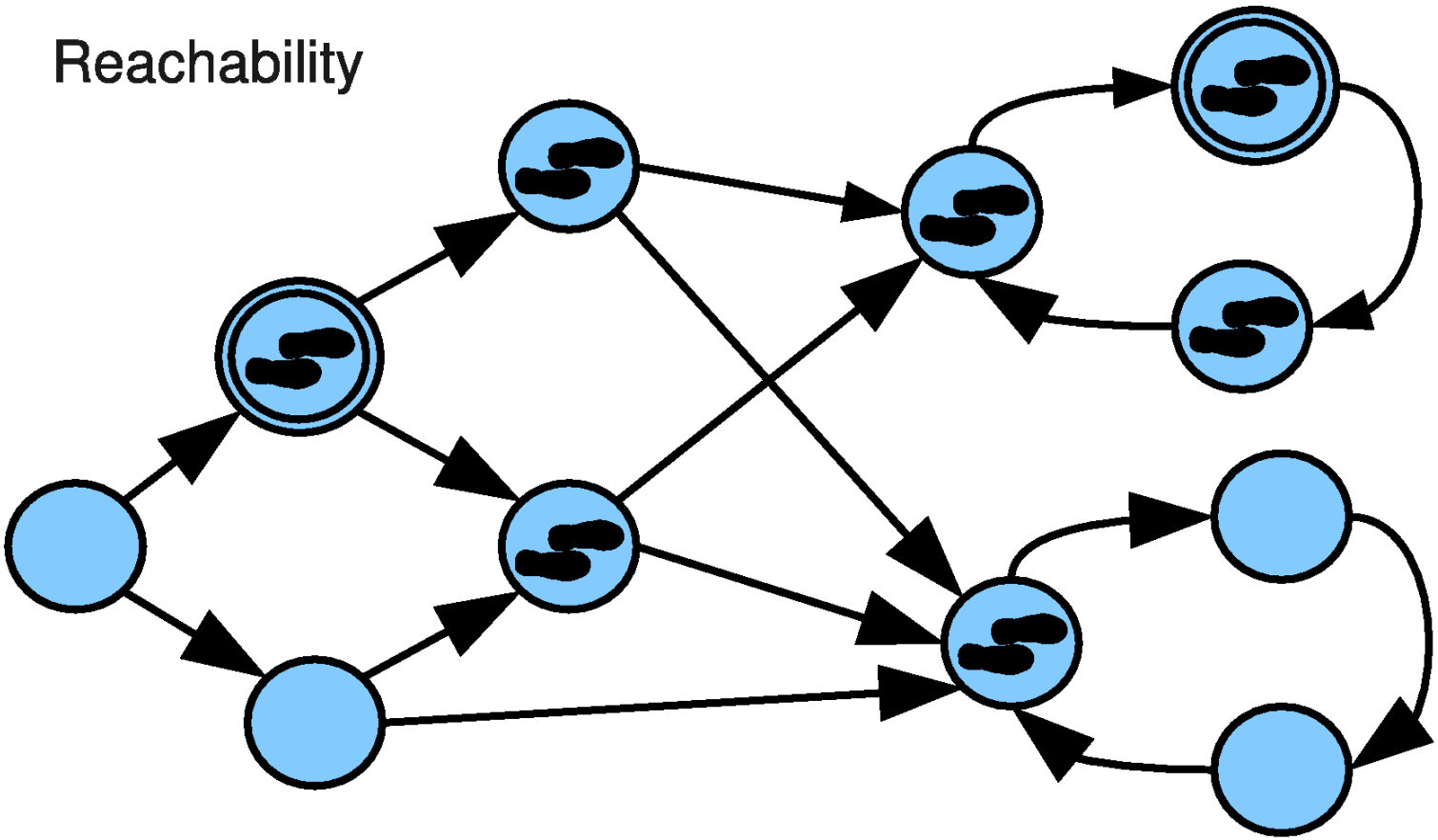
OWCTY Demonstration

1st iteration



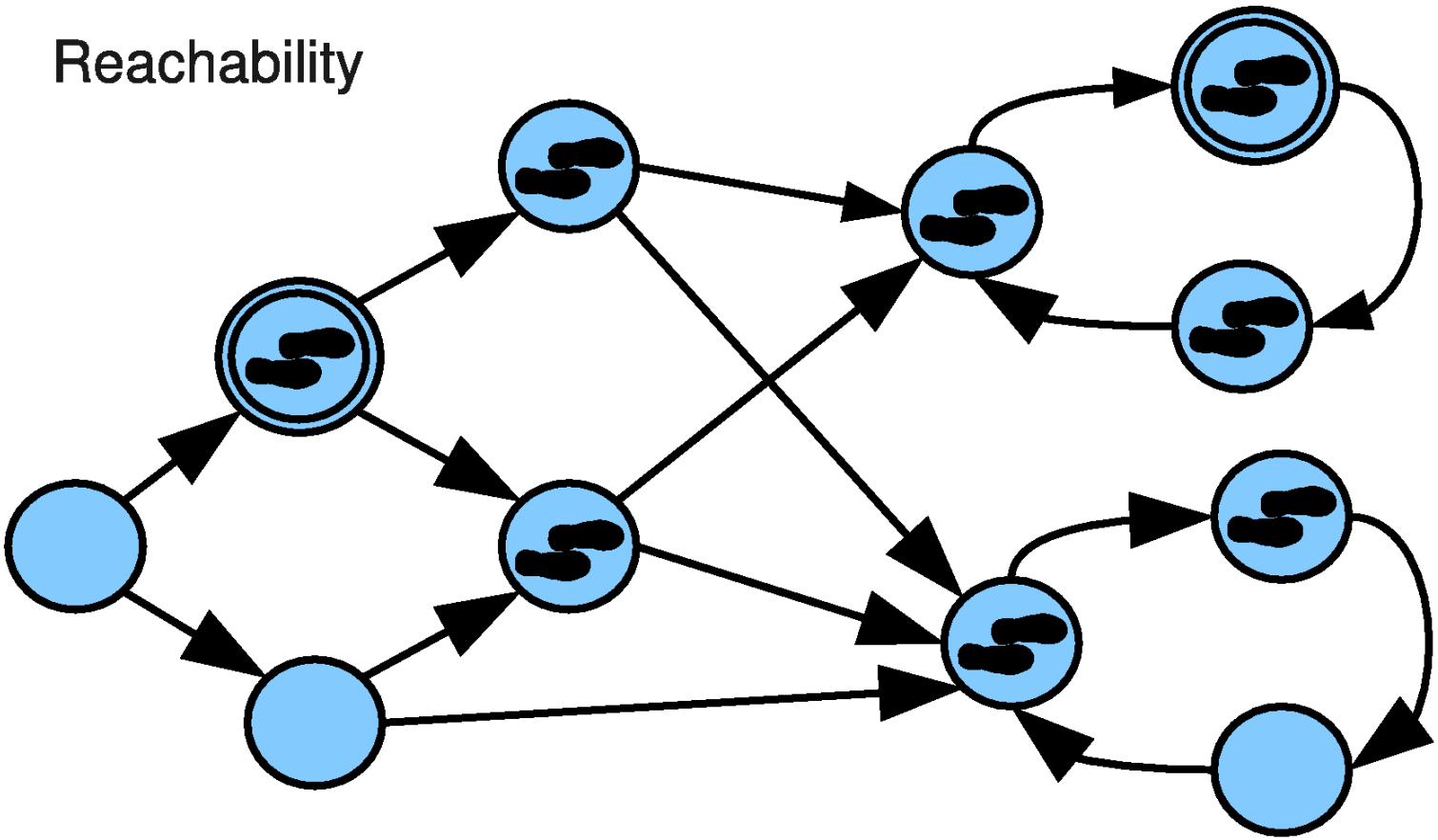
OWCTY Demonstration

1st iteration



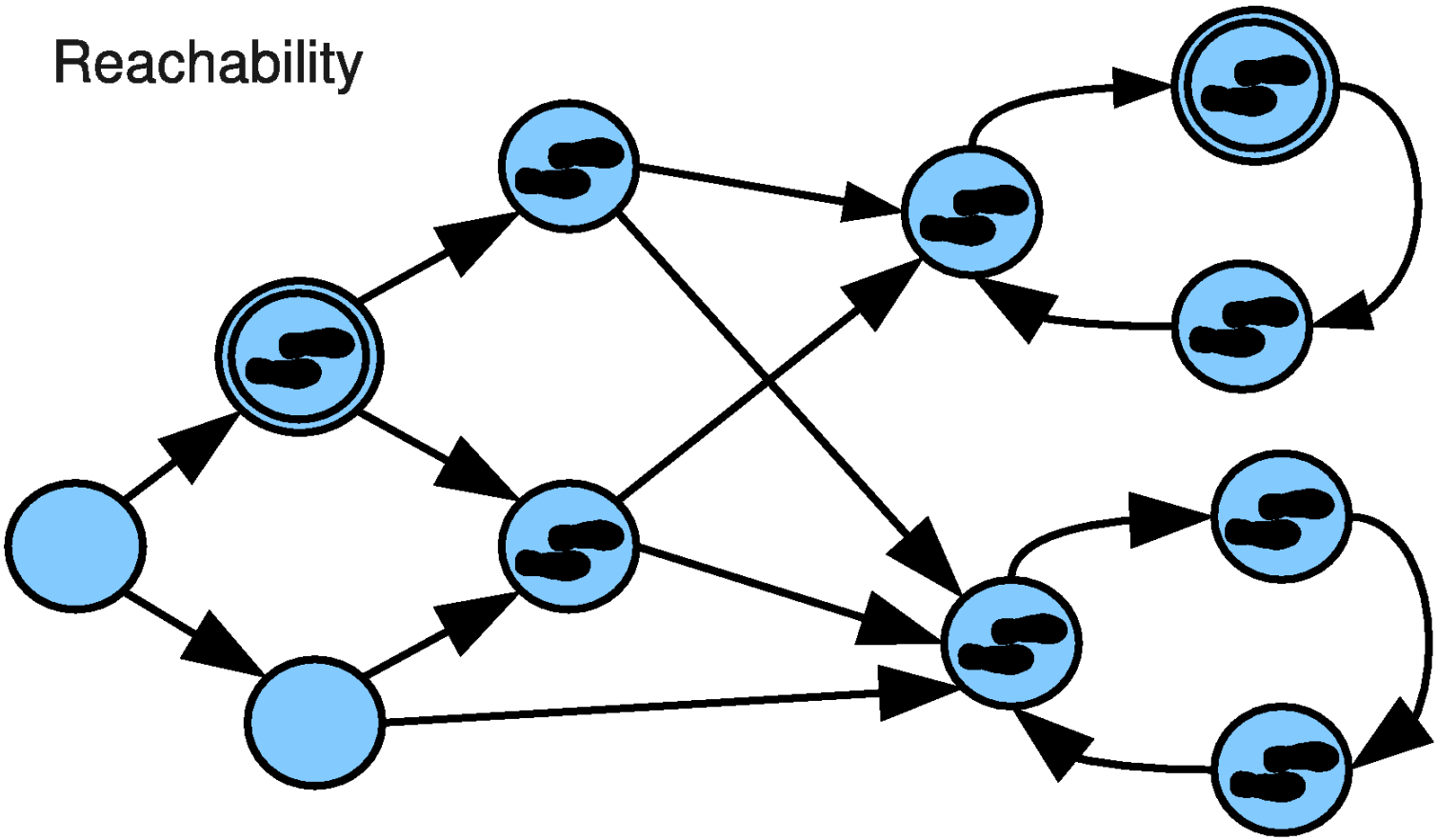
OWCTY Demonstration

1st iteration



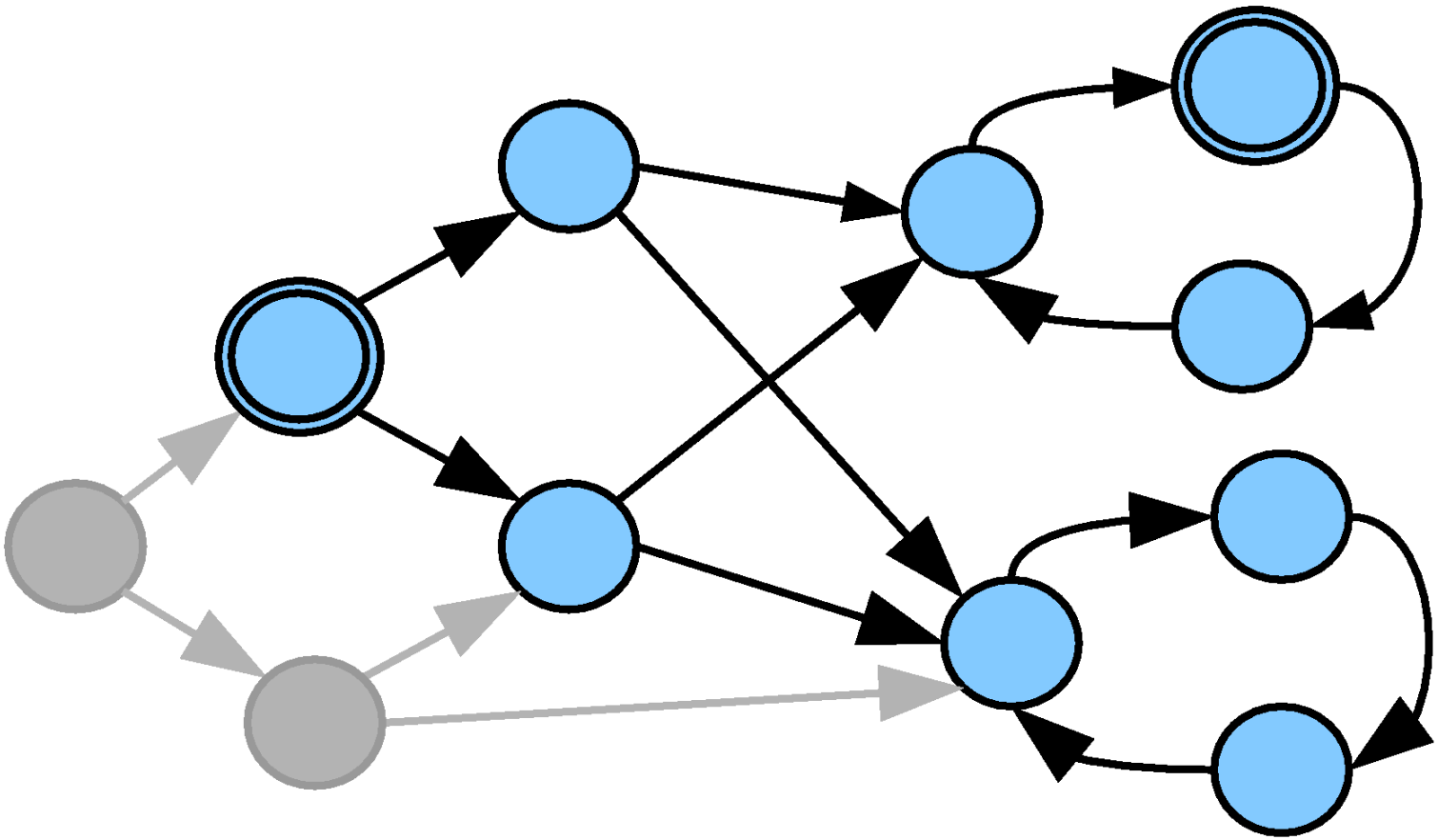
OWCTY Demonstration

1st iteration



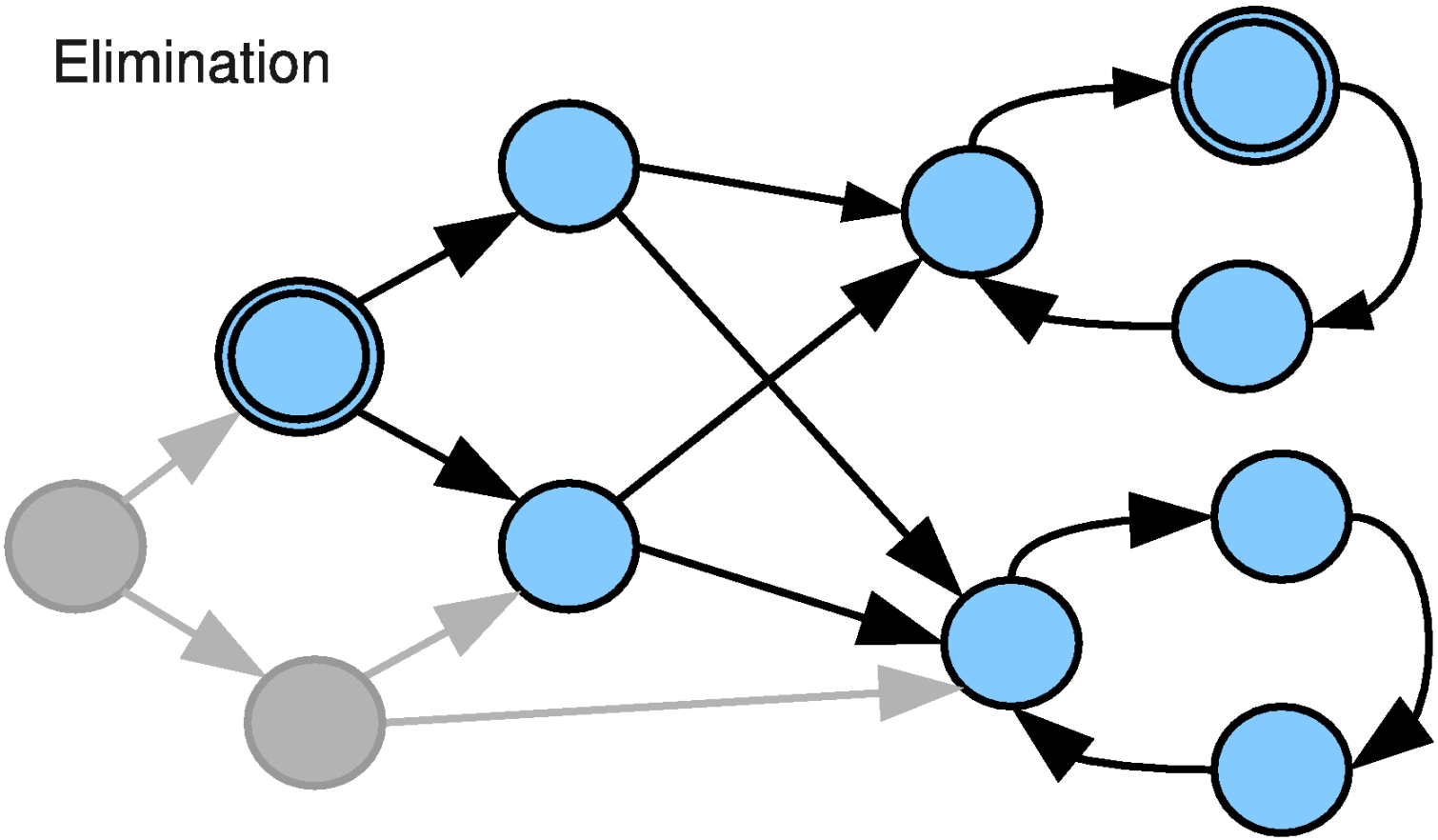
OWCTY Demonstration

1st iteration



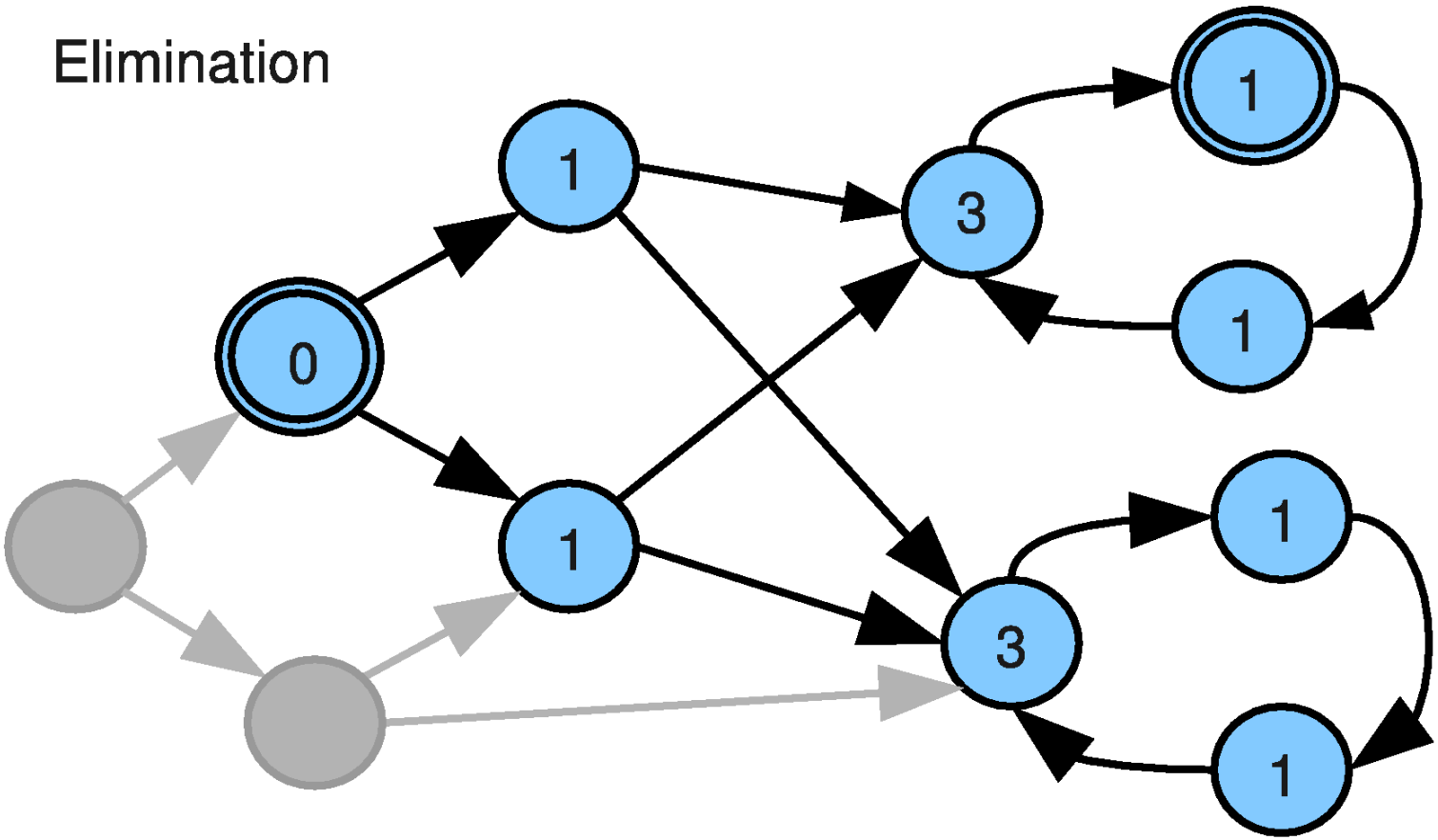
OWCTY Demonstration

1st iteration



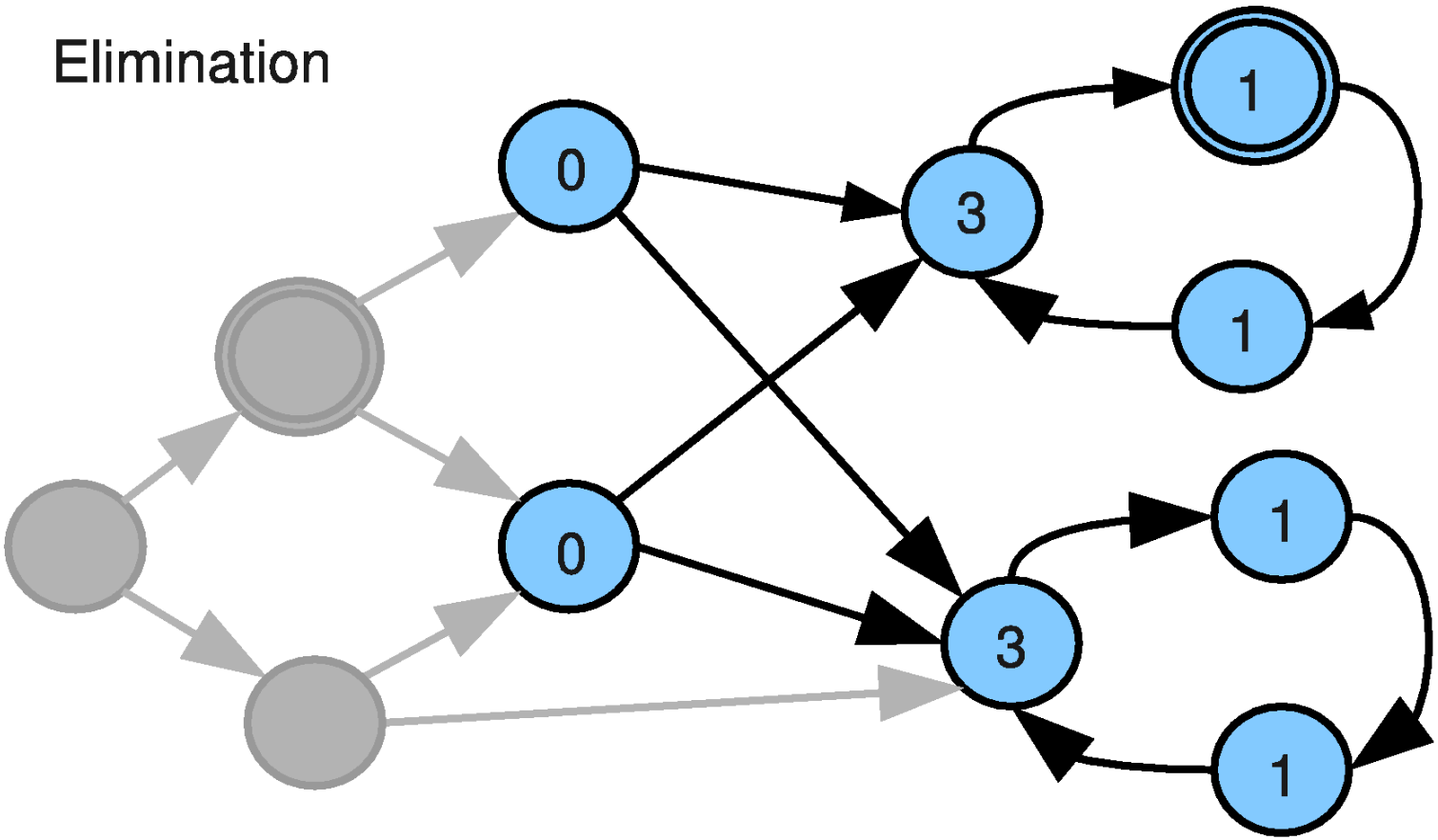
OWCTY Demonstration

1st iteration



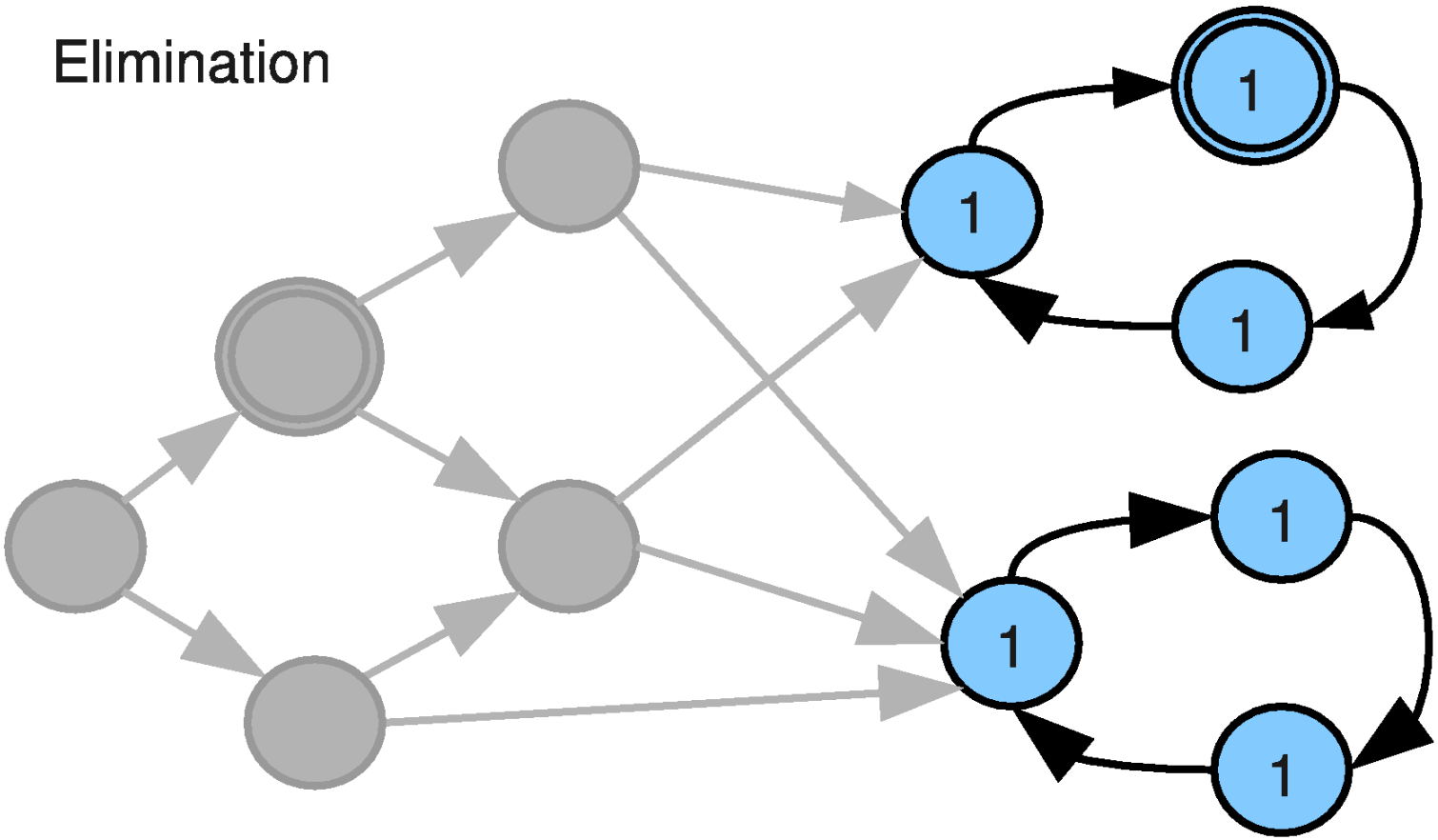
OWCTY Demonstration

1st iteration



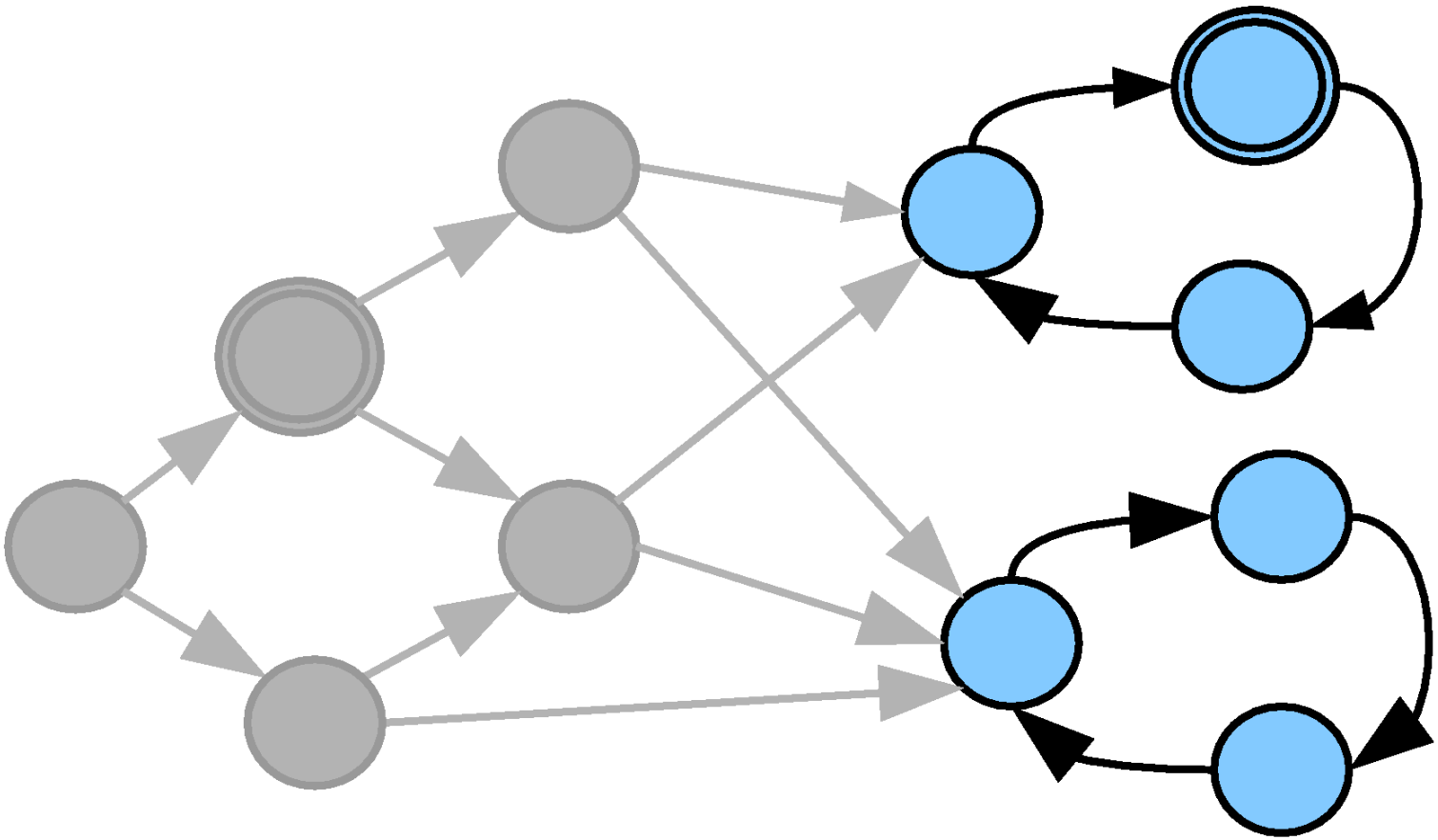
OWCTY Demonstration

1st iteration



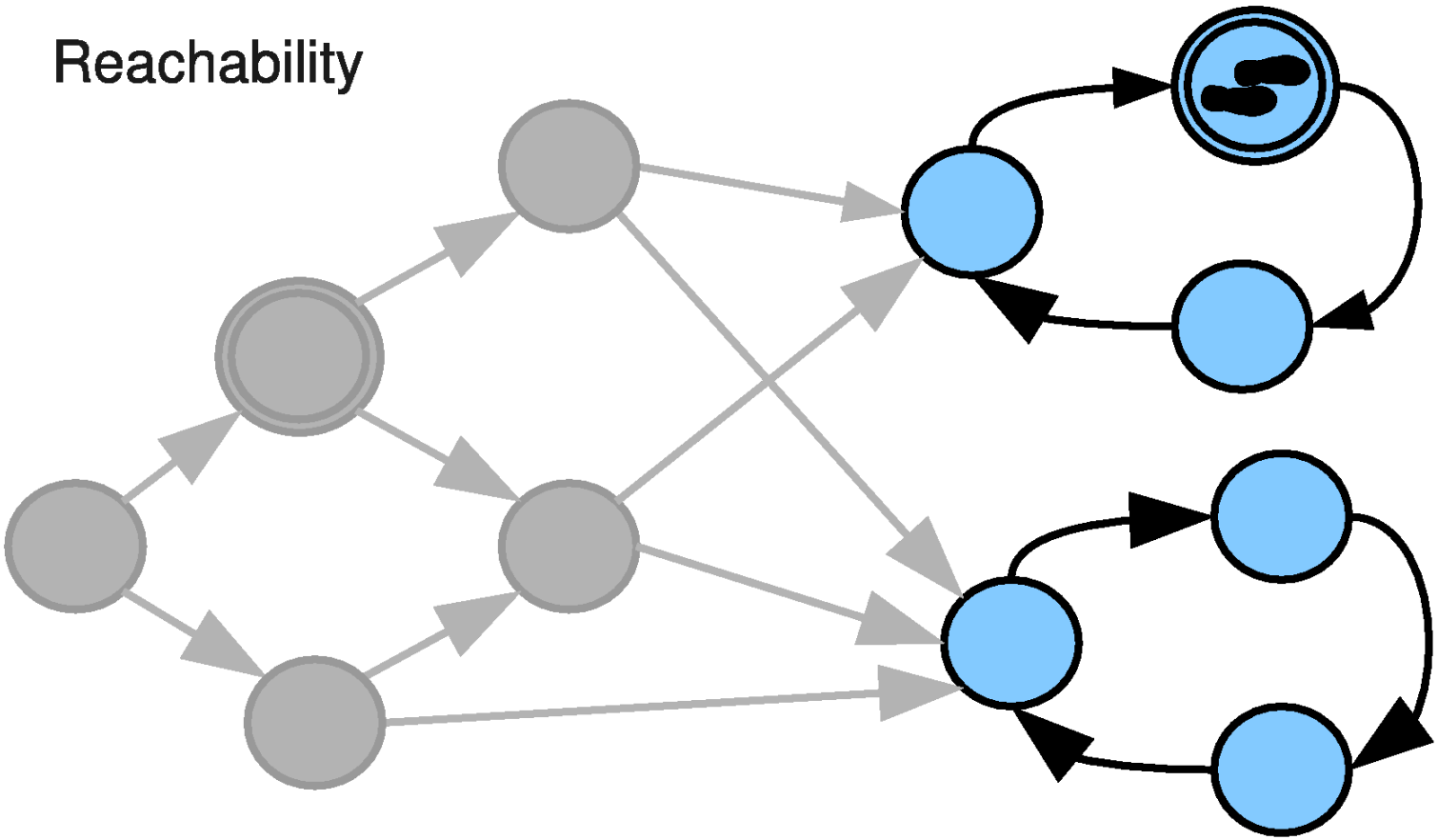
OWCTY Demonstration

1st iteration



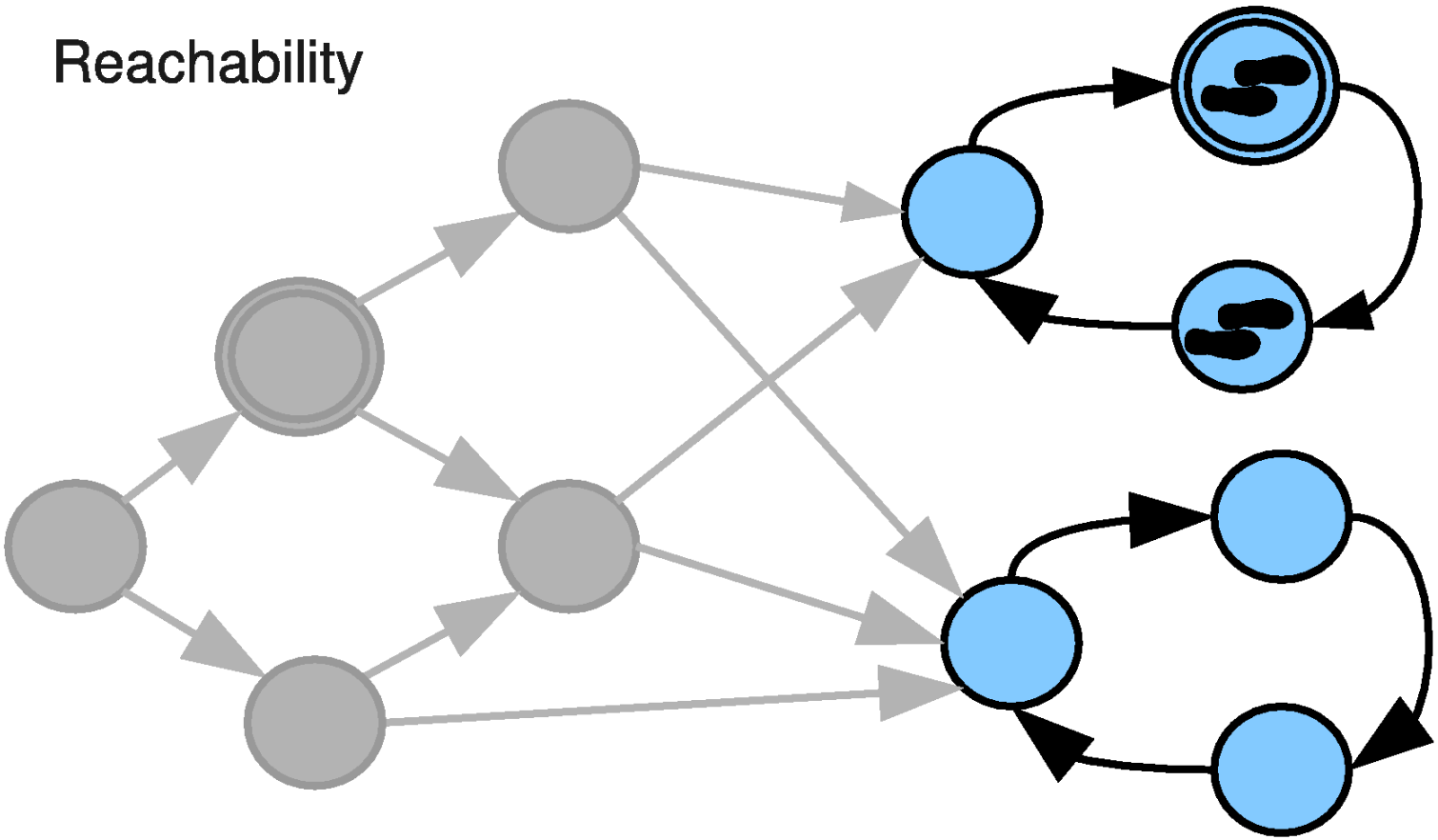
OWCTY Demonstration

2nd iteration



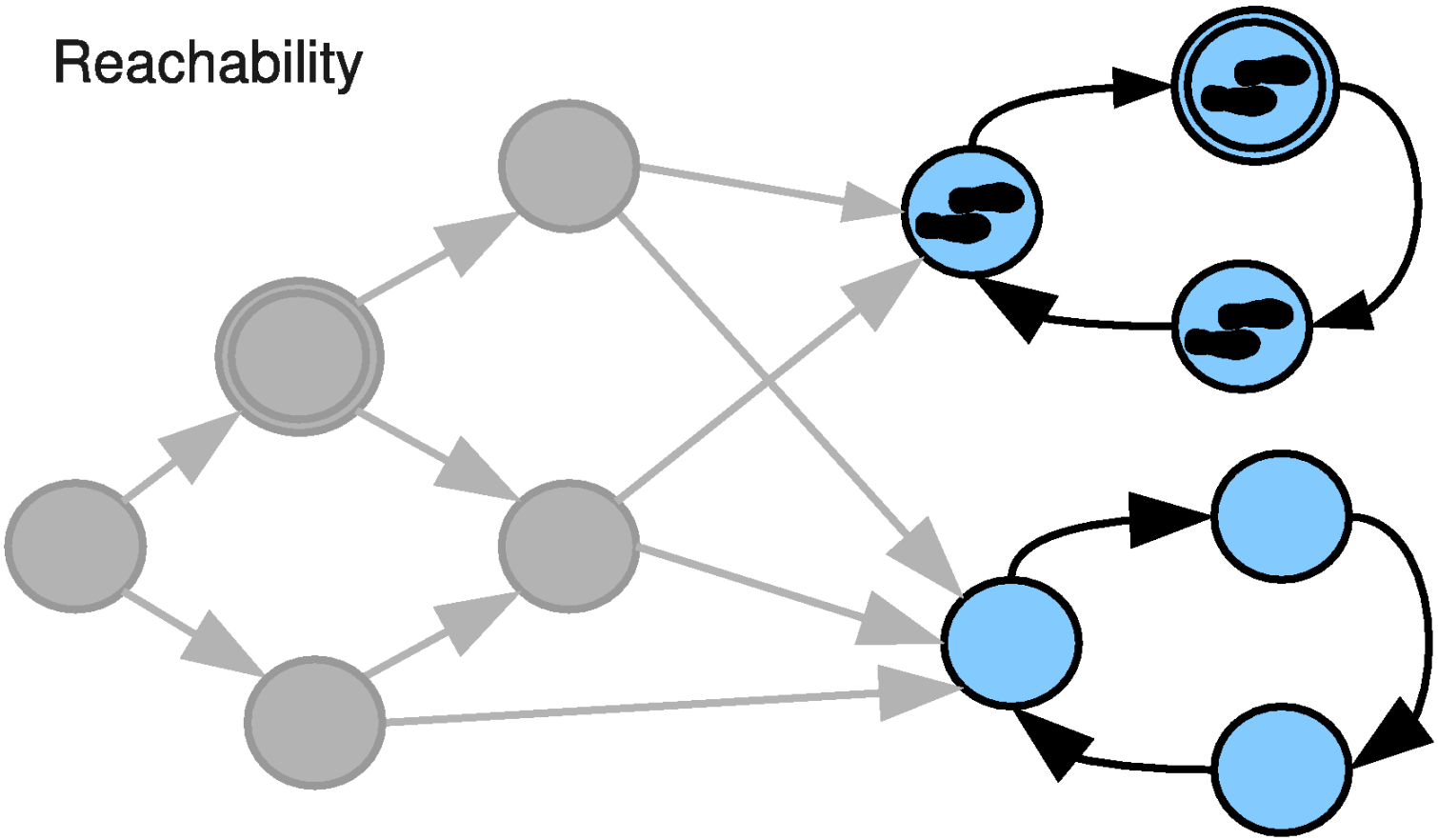
OWCTY Demonstration

2nd iteration



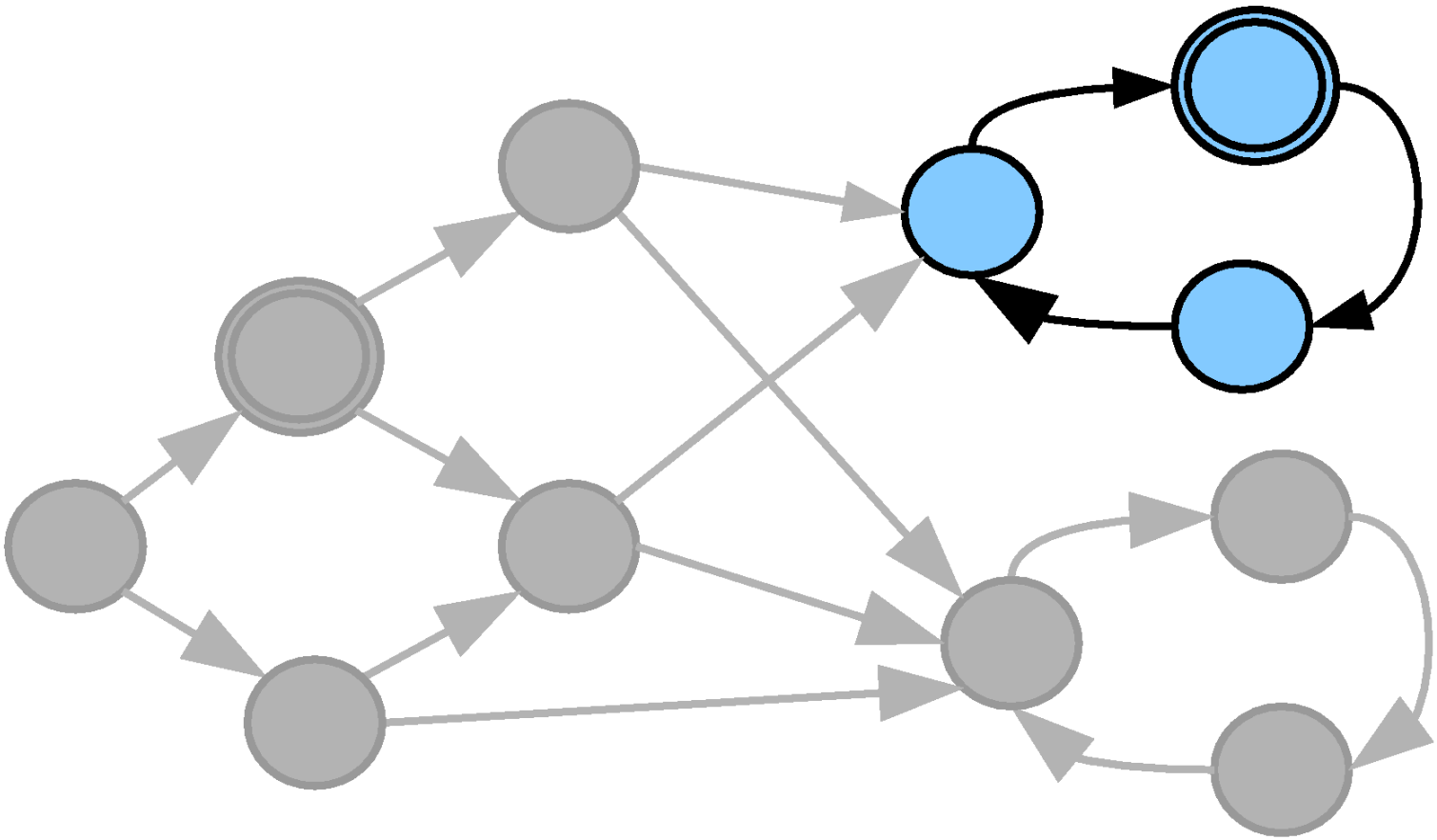
OWCTY Demonstration

2nd iteration



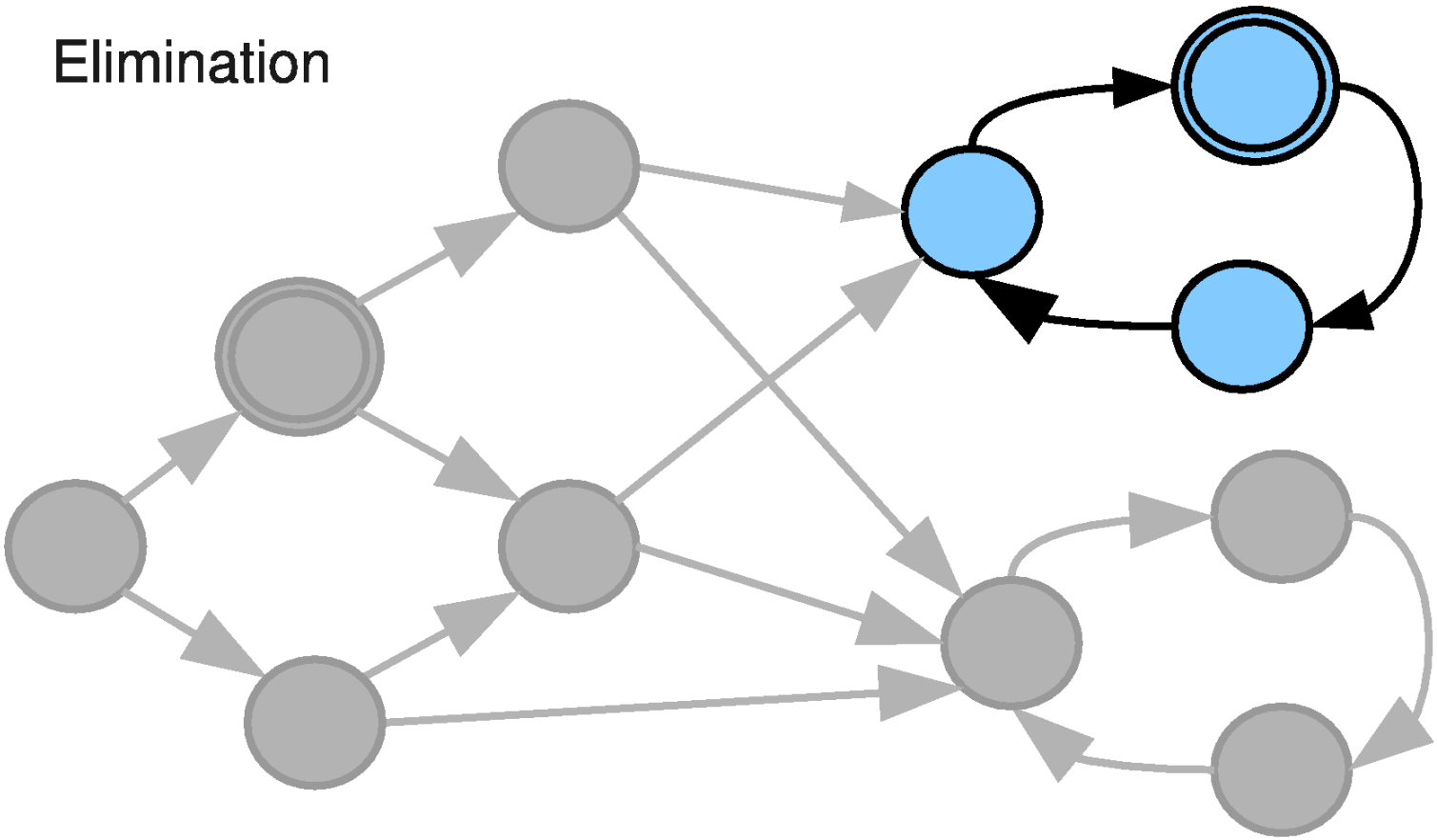
OWCTY Demonstration

2nd iteration

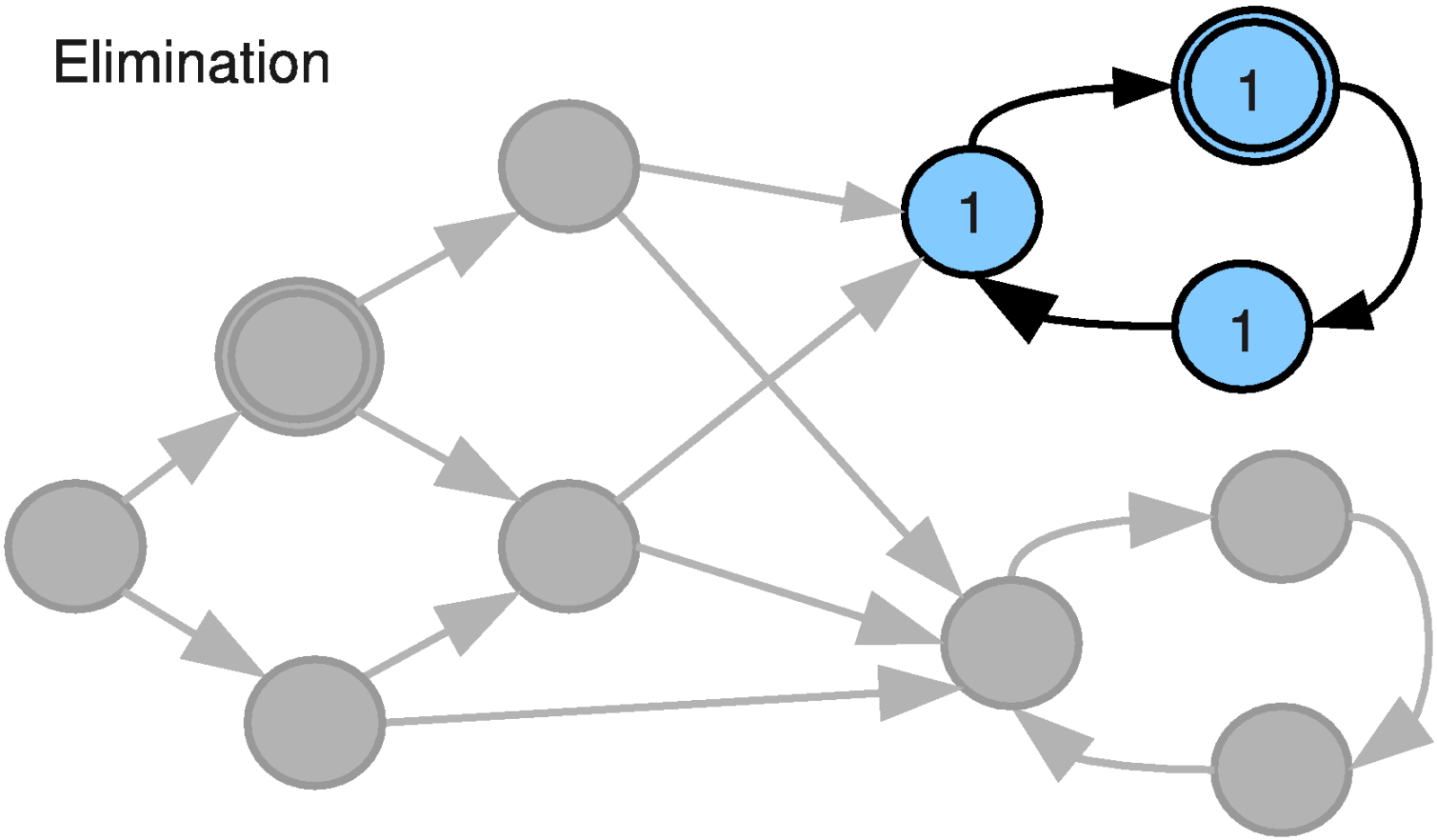


OWCTY Demonstration

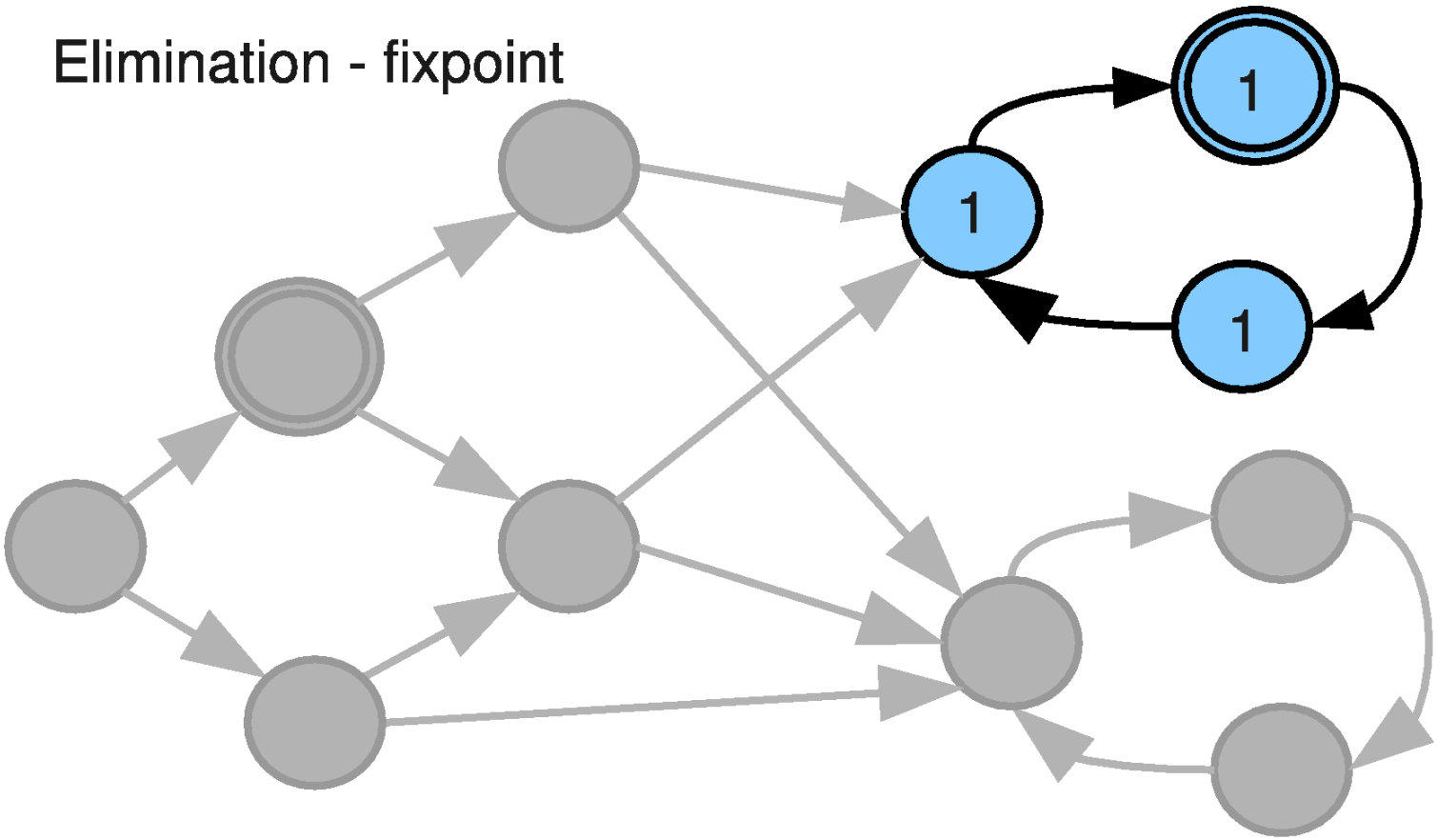
2nd iteration



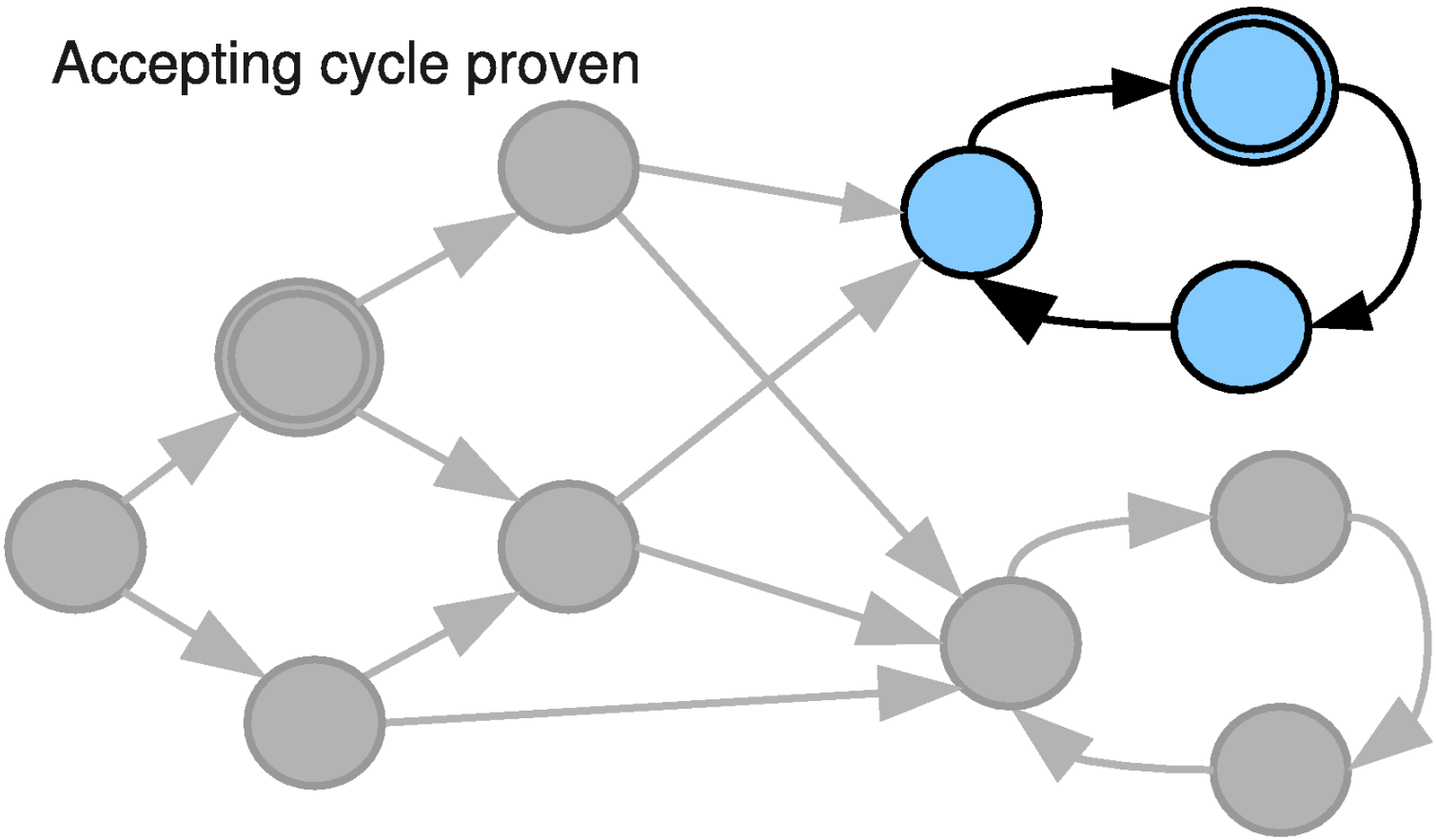
2nd iteration



2nd iteration



OWCTY Demonstration



Algorithm MAP

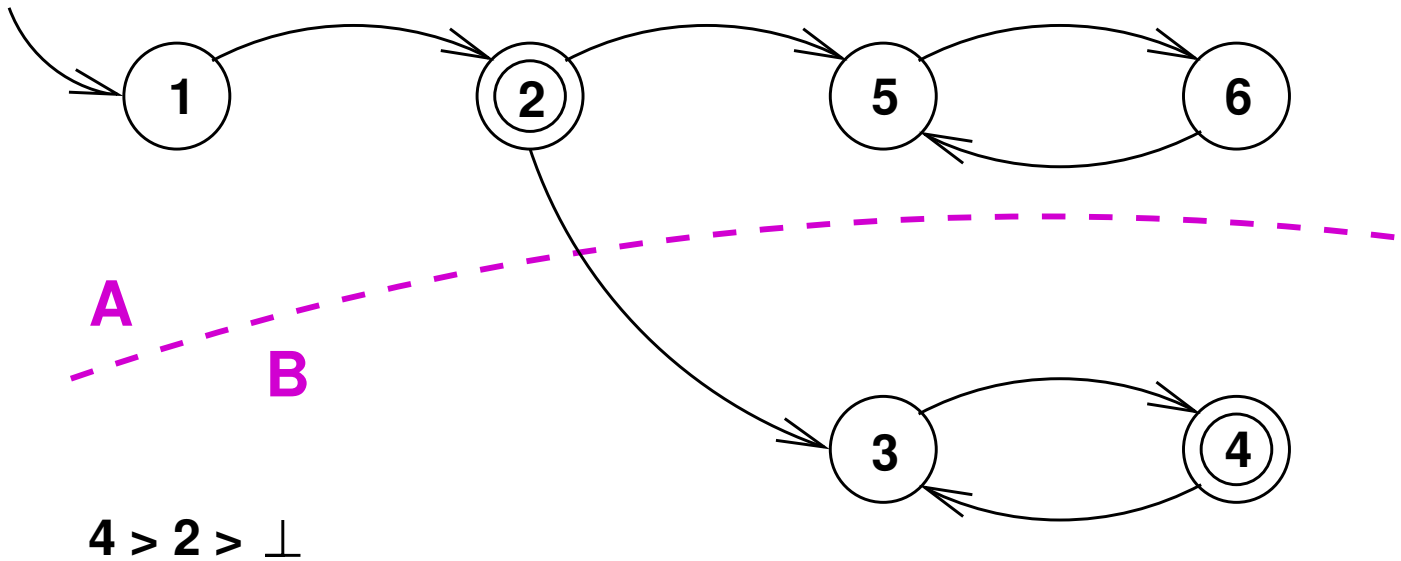
Idea

- Eliminate accepting states that are outside an accepting cycle.
- An accepting state does not lie on a cycle if it is not a predecessor of itself.

Realization

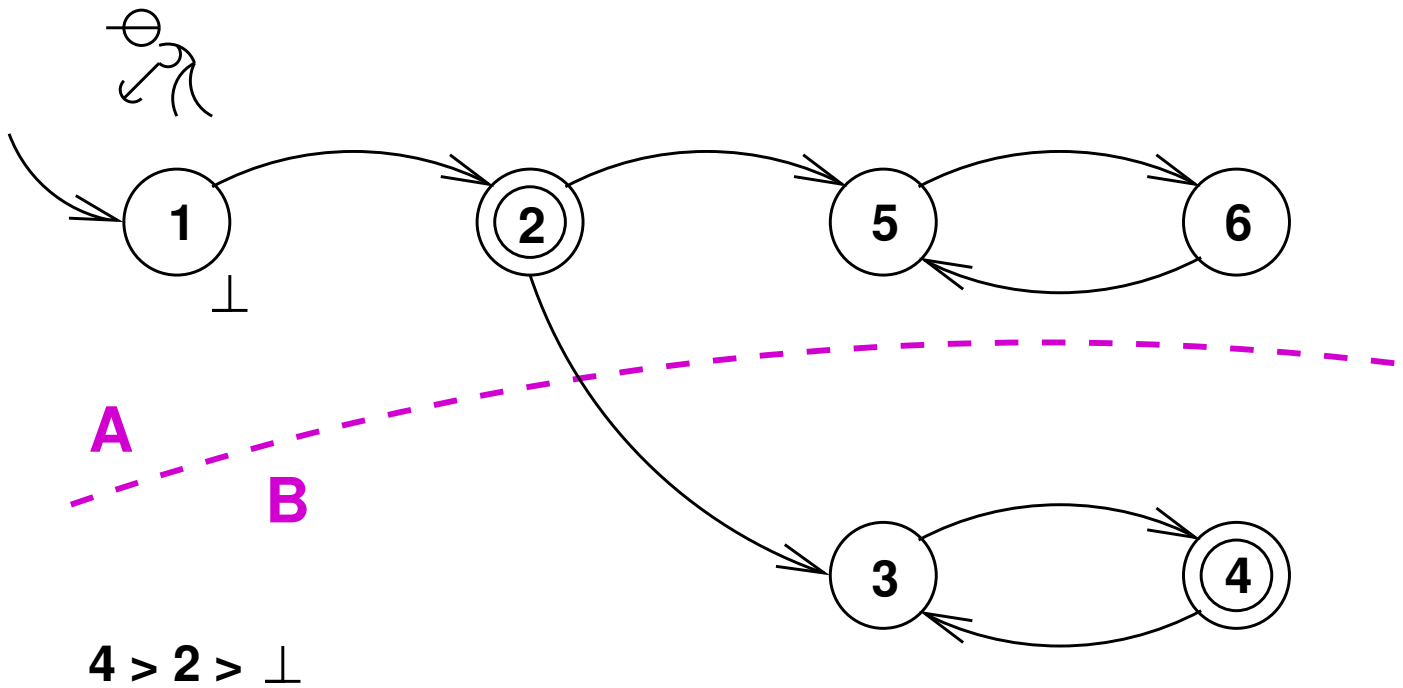
- Assume an ordering on accepting states.
- Propagate maximal accepting states.
- If a state is propagated into itself, accepting cycle is found.
- Remove maximal accepting states that are outside a cycle, and repeat until there are some accepting states left.
- **Propagation of accepting states can be done in parallel.**

Computing maximal accepting predecessors (MAPs)



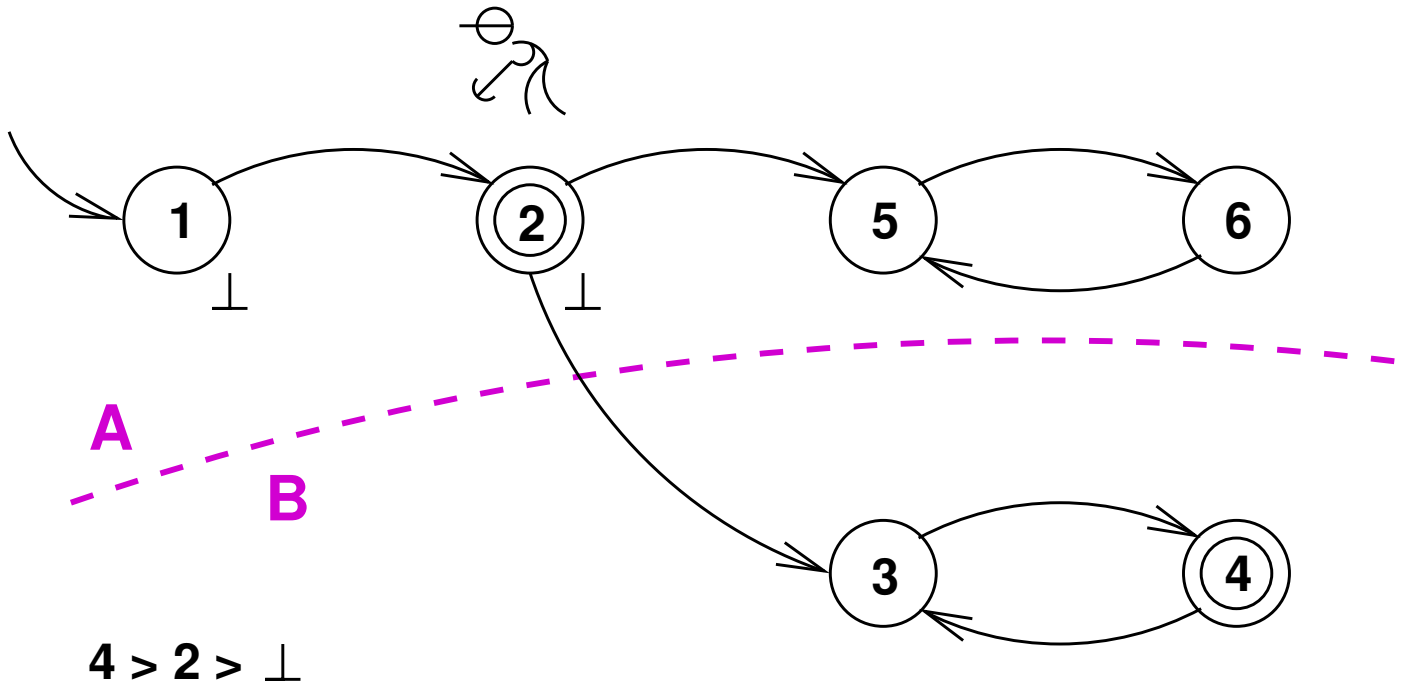
Two workers A and B.

Computing maximal accepting predecessors (MAPs)



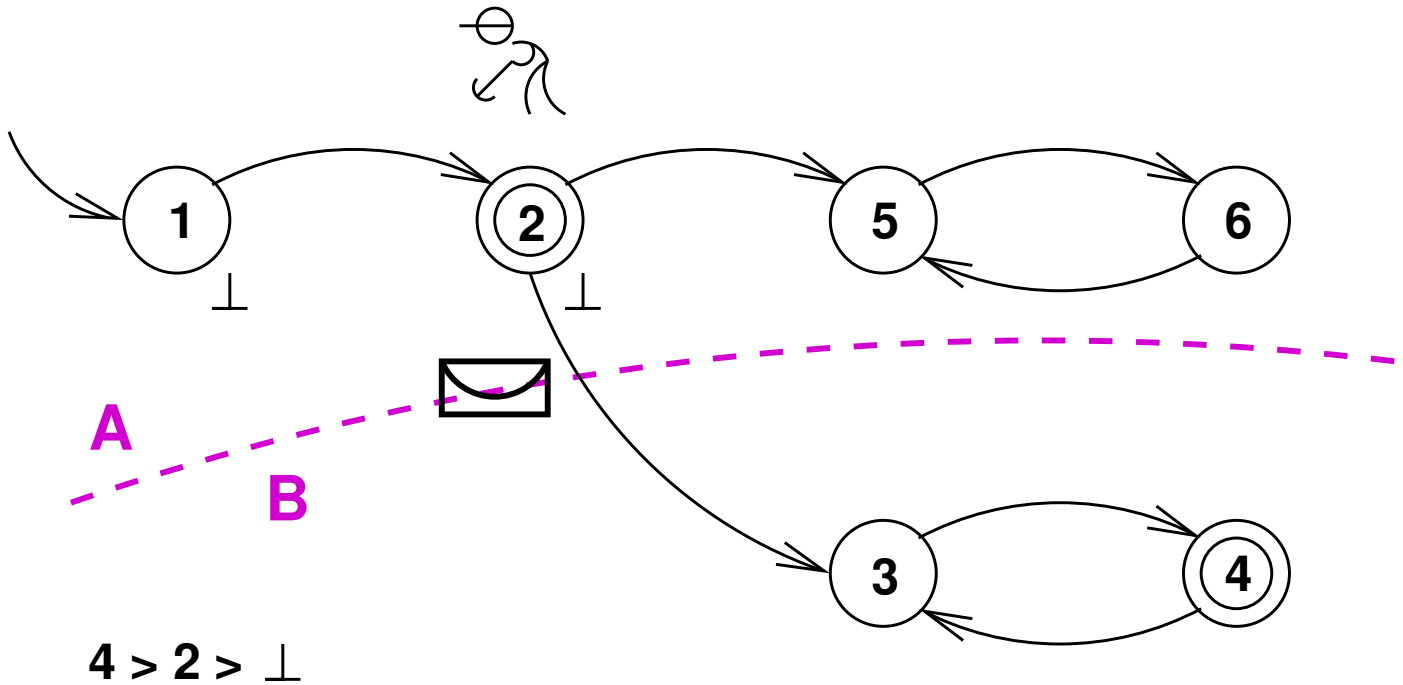
Each worker processes its own states.

Computing maximal accepting predecessors (MAPs)



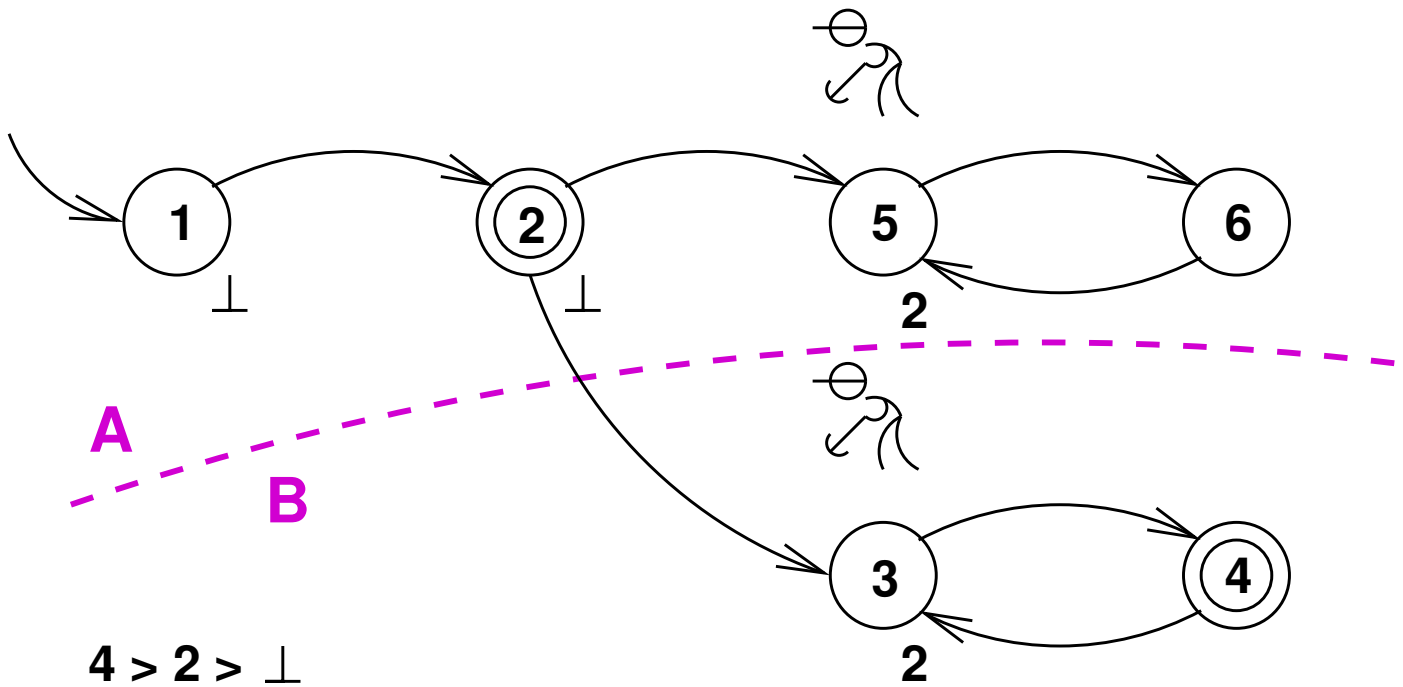
Each worker processes its own states.

Computing maximal accepting predecessors (MAPs)



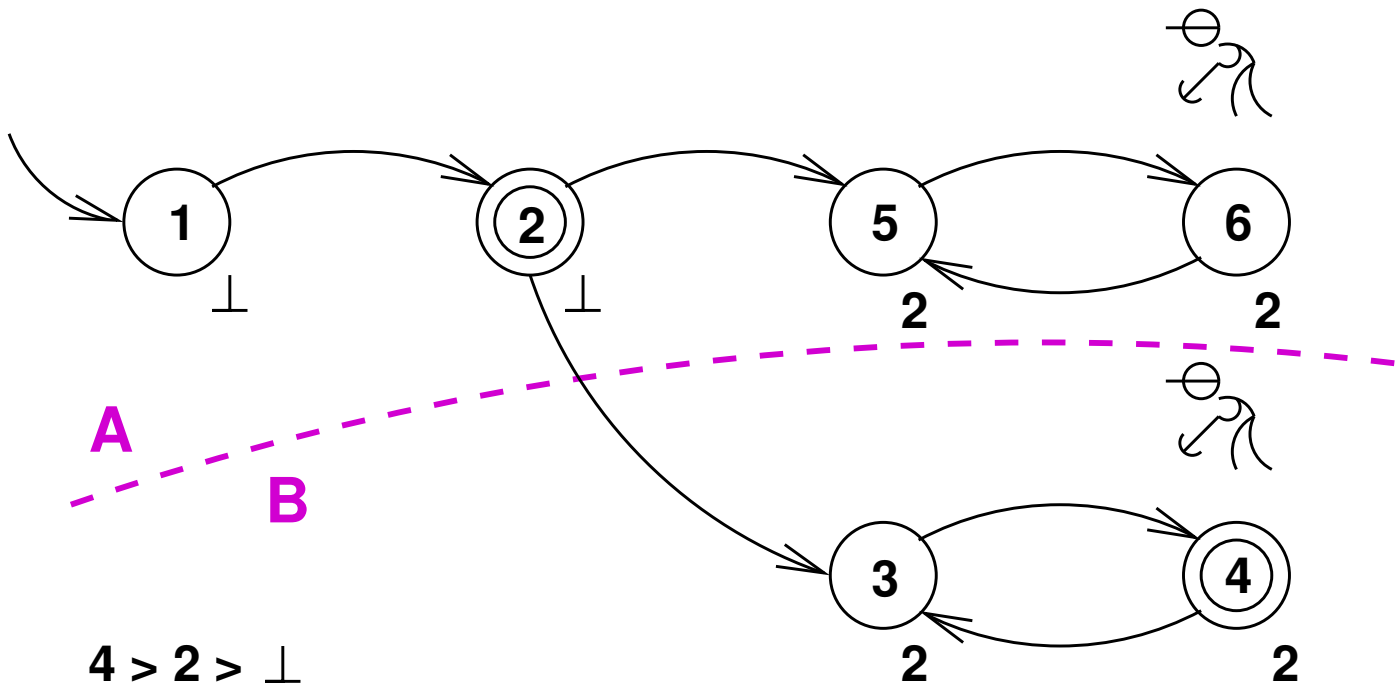
Non local states are sent over.

Computing maximal accepting predecessors (MAPs)



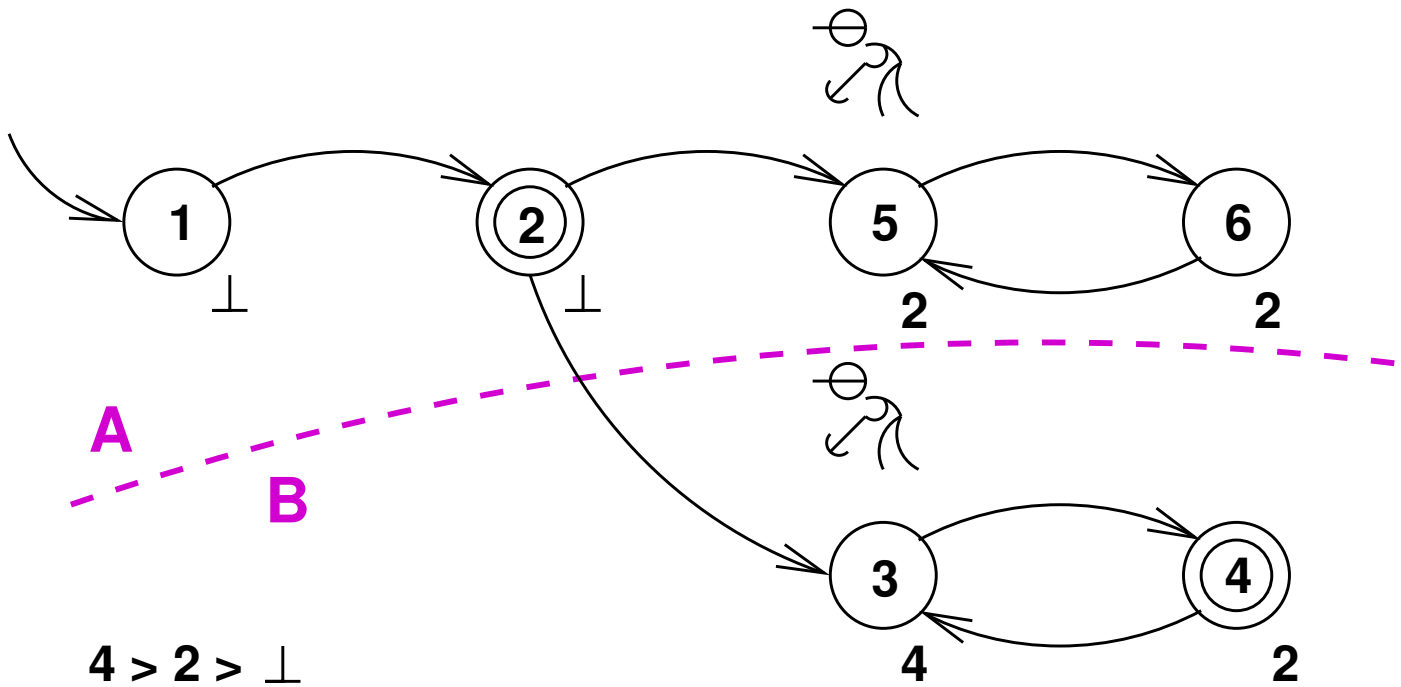
States are processed in parallel.

Computing maximal accepting predecessors (MAPs)



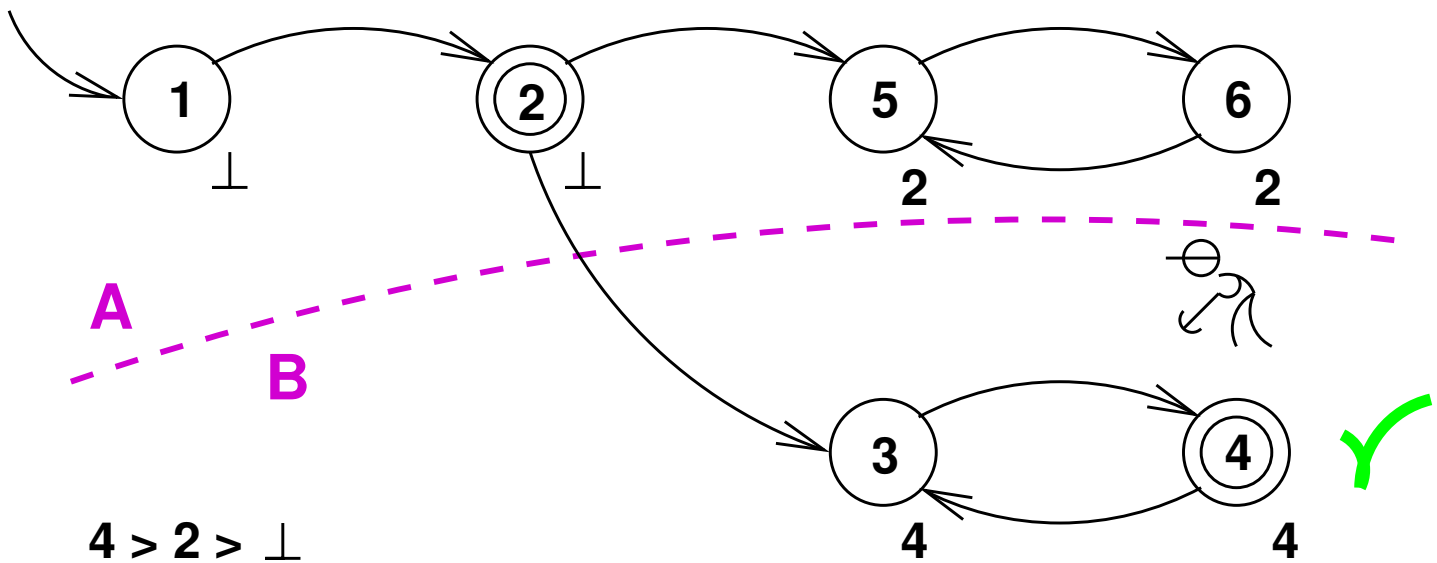
States are processed in parallel.

Computing maximal accepting predecessors (MAPs)



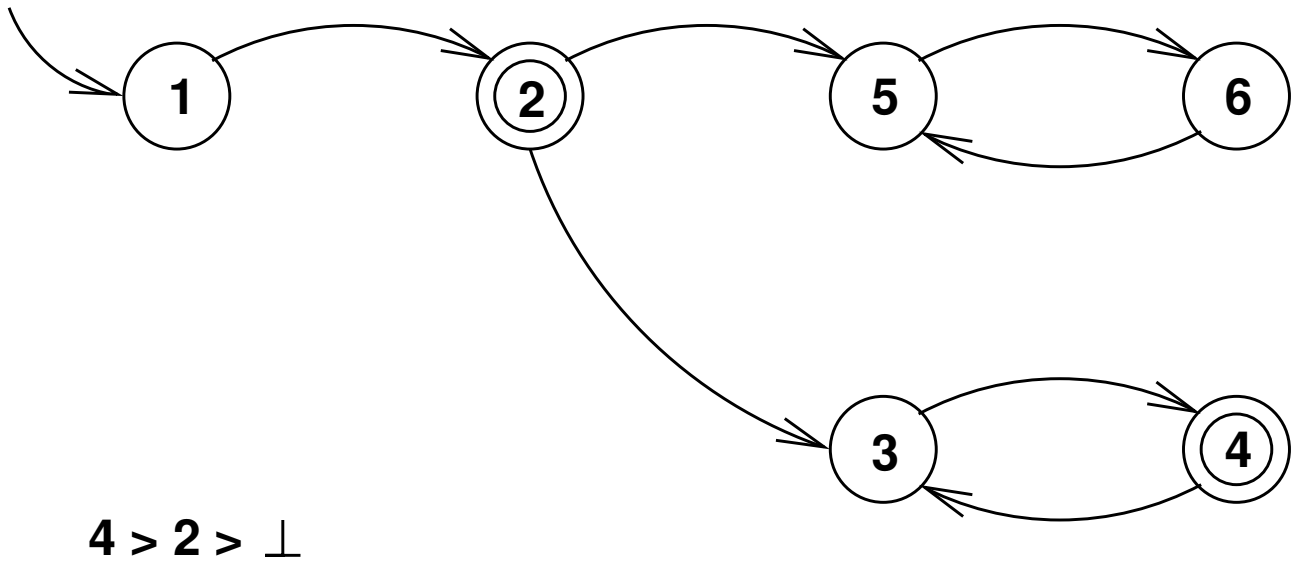
States are processed in parallel.

Computing maximal accepting predecessors (MAPs)



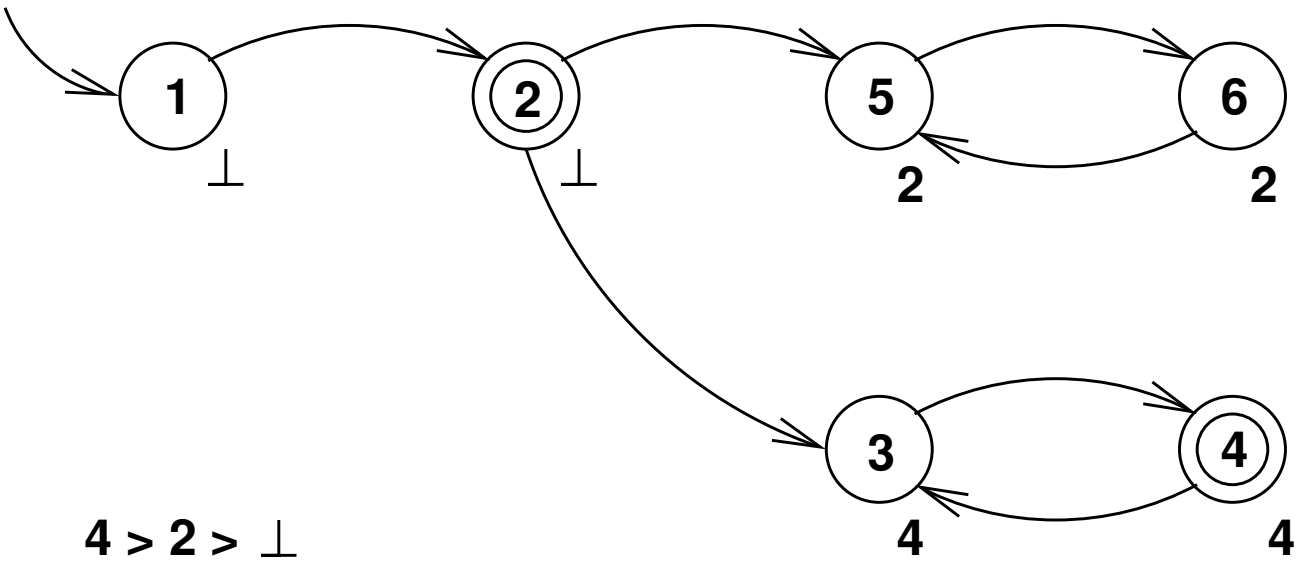
States are processed in parallel.

Algorithm MAP



State ordering.

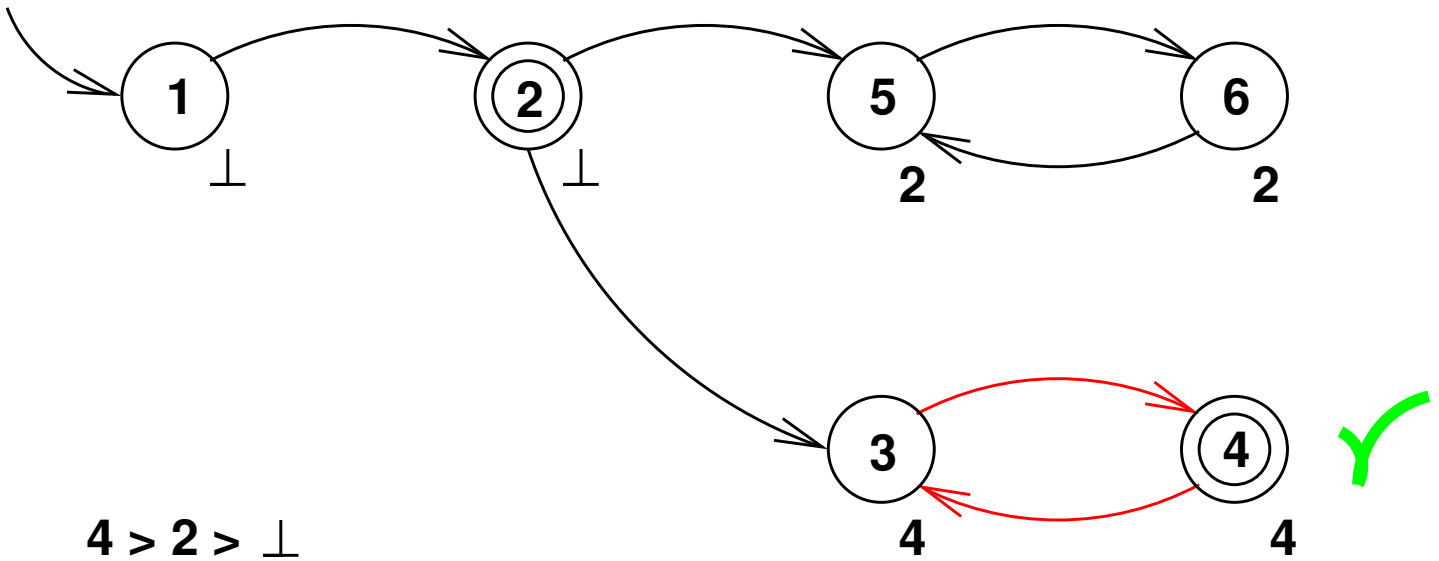
Algorithm MAP



Maximal Accepting Predecessor (MAP)

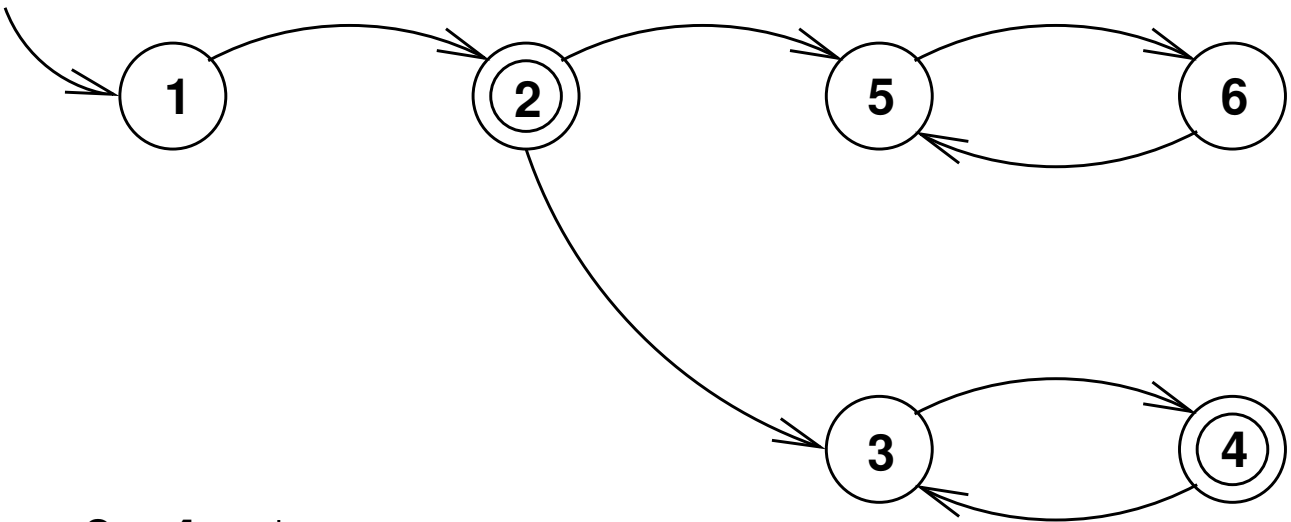
$$map(v) = \max\{\bot, u \mid (u, v) \in E^+ \wedge u \in F\}$$

Algorithm MAP



$$map(v) = v \quad \Longrightarrow \quad \text{accepting cycle}$$

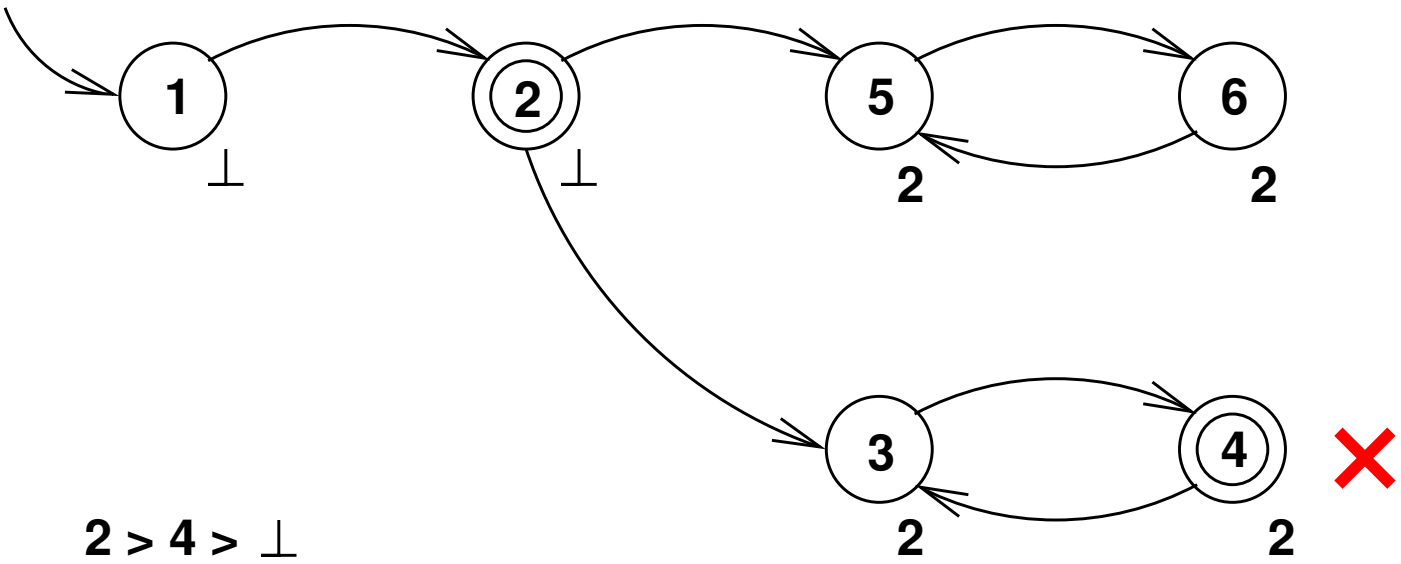
Algorithm MAP



2 > 4 > $\perp$

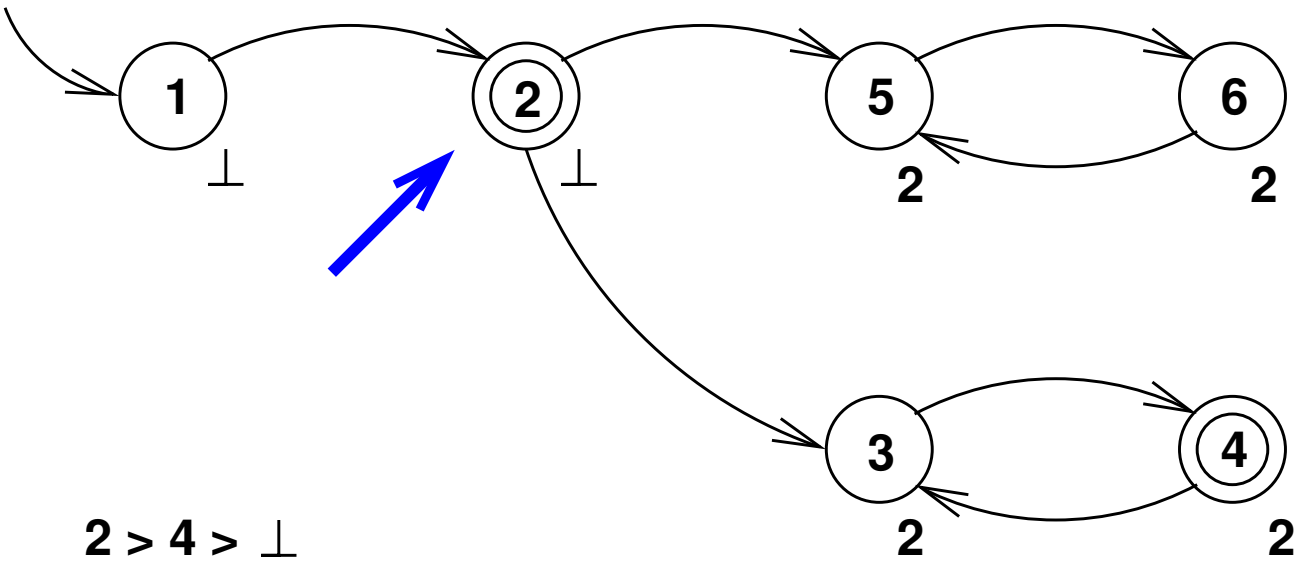
What if $2 > 4$?

Algorithm MAP



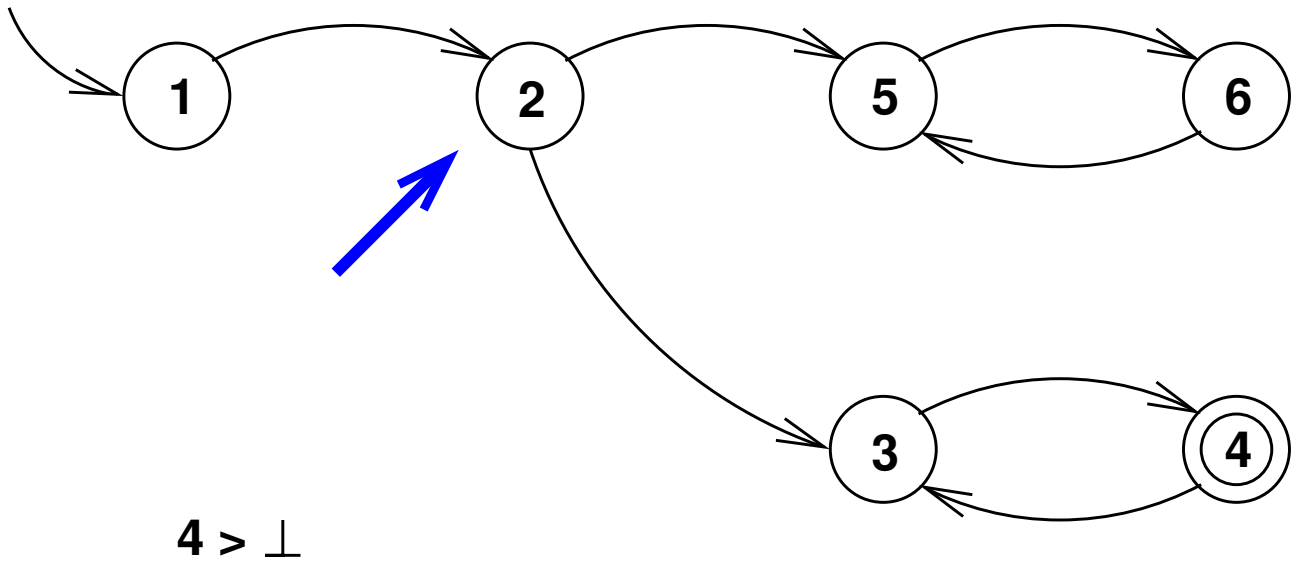
Accepting cycle undetected.

Algorithm MAP



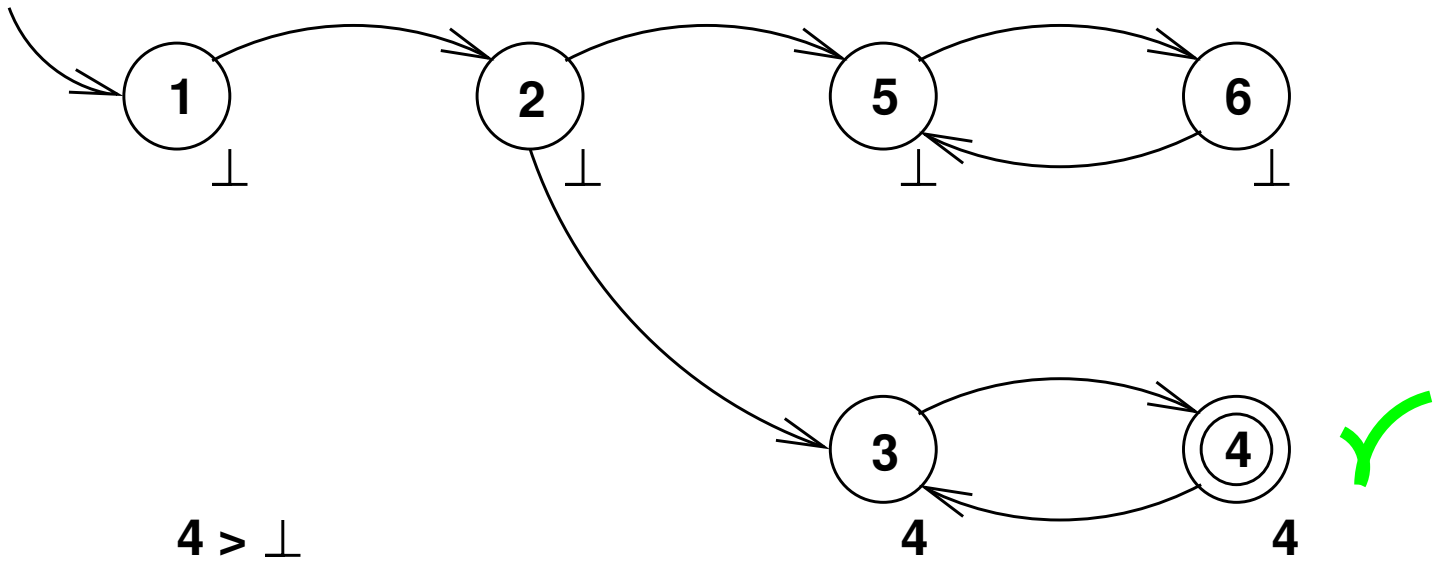
If no accepting cycle is found, then maximal accepting vertices cannot be part of an accepting cycle.

Algorithm MAP



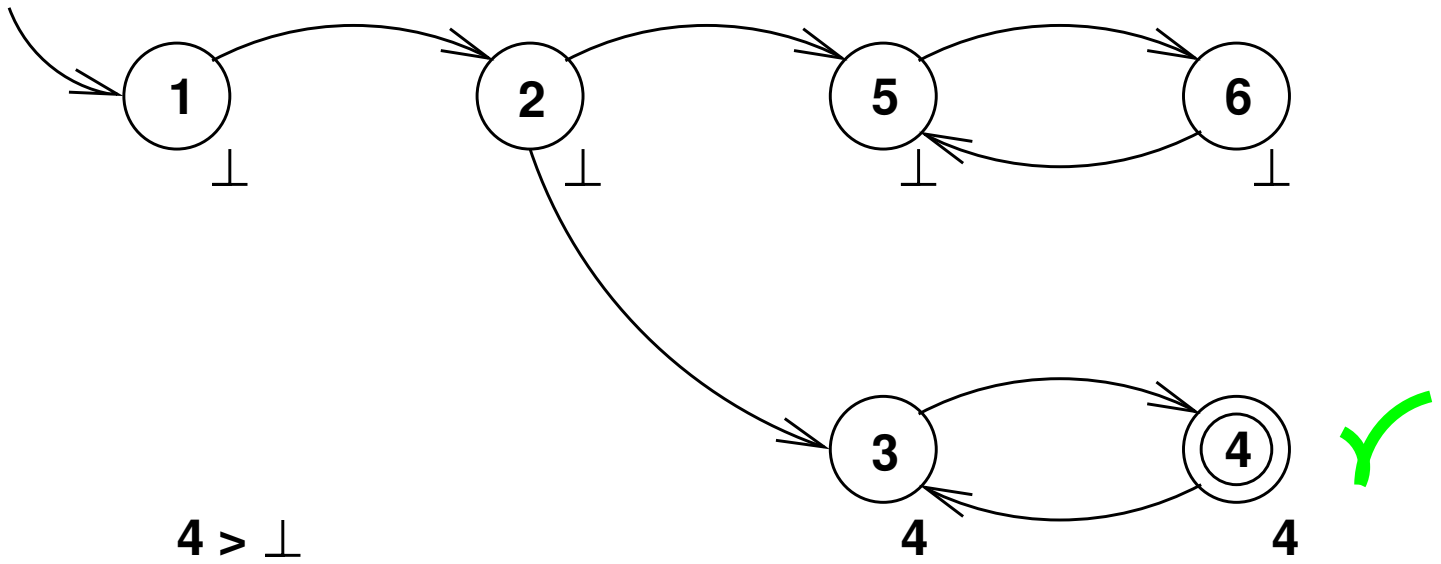
Maximal accepting vertices marked as non-accepting.

Algorithm MAP



Repeat until accepting cycle is found or
there are no accepting vertices.

Algorithm MAP



Succeeding iterations localized to subgraphs
with the same value of MAP.

New algorithms – Conclusion

Algorithm comparison

	Complexity	Optimality	On-The-Fly
Nested DFS	O(N+M)	Yes	Yes
OWCTY Algorithm			
general LTL properties	O(N.(N+M))	No	No
weak LTL properties	O(N+M)	Yes	No
MAP Algorithm	O(N.N.(N+M))	No	Yes

N – number of states
M – number of transitions

Possible first impression: **The new algorithms are useless.**

Experimental Validation: Shared-Memory Systems

Used hardware

- 16 core server (8x AMD dual core)
- 64 GB RAM

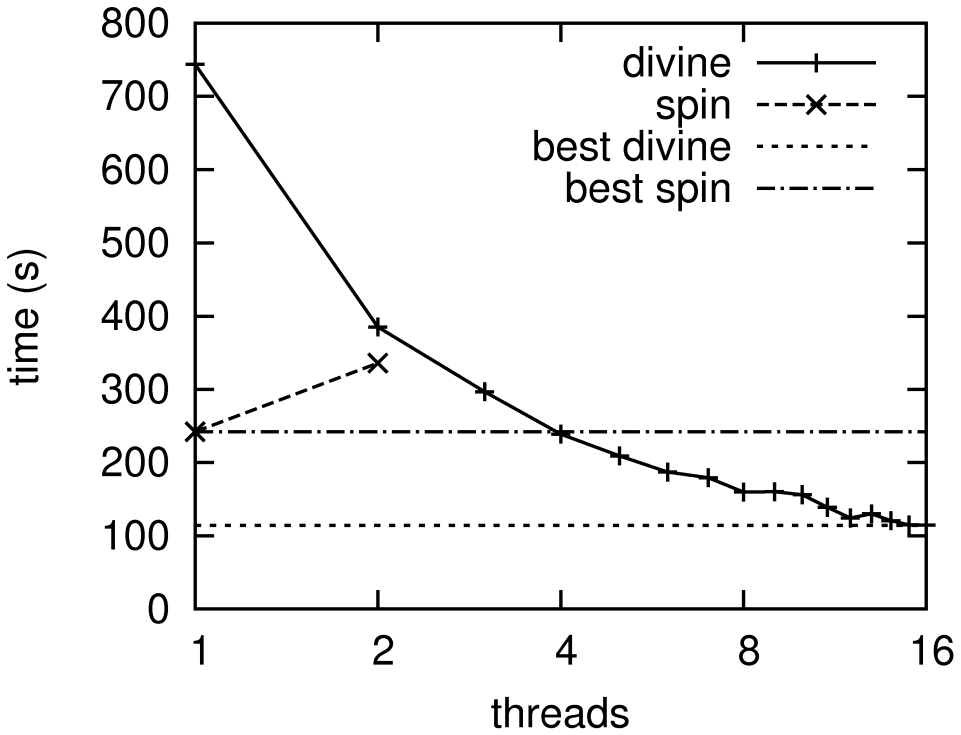
LTL Model Checking tools

- SPIN
  - Nested DFS algorithm
  - Implementation is optimized continuously for almost 20 years
- DiVinE Multi-Core
  - OWCTY algorithm
  - Still space for implementation improvements

The Question

- Can new algorithms outperform the optimal ones?

Shared-Memory Systems – OWCTY Scalability



SPIN: 1 Core: 238 sec
2 Cores: 343 sec

DiVinE Multi-Core:

Cores	Runtime (sec)	Efficiency
1	738	100%
2	392	94%
4	235	78%
8	170	54%
16	106	43%

GPU-Accelerated Systems

Used hardware

- NVIDIA CUDA technology
- MSI N280GTX-T2D1G-SUPER OC (1 GB DDR3,FAN)

DiVinE-CUDA

- Builds CSR representation of the underlying graph.
- Stores it in the memory of the GPU.
- MAP algorithm reformulated as a matrix-vector product.

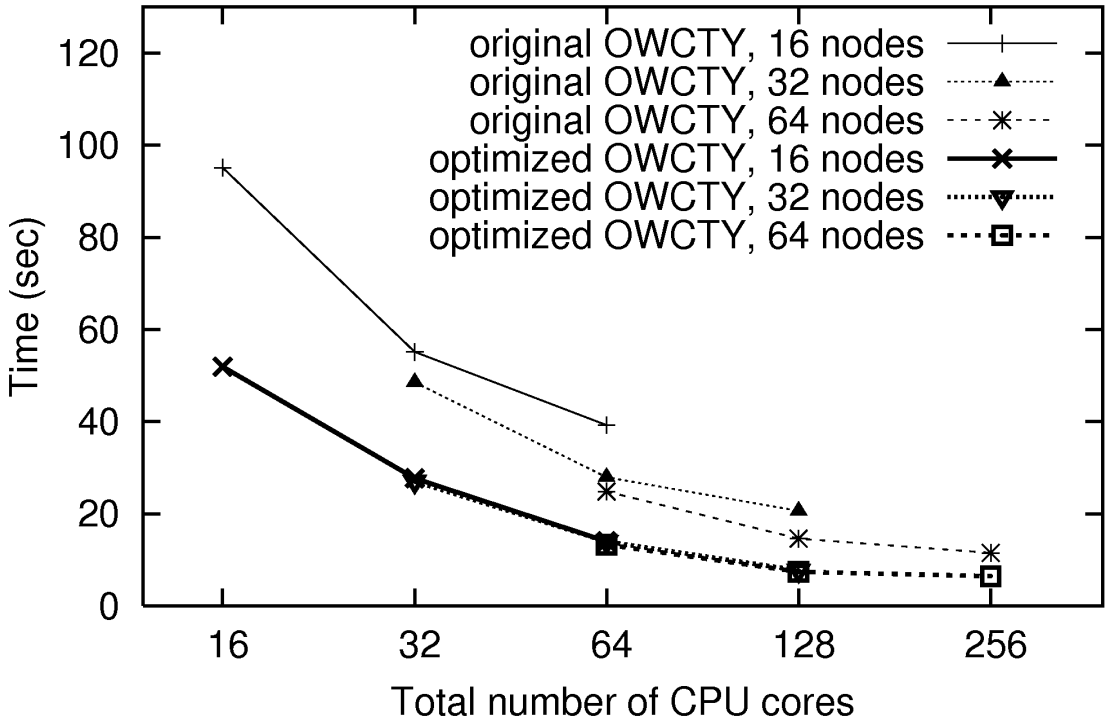
The Question

- Can one hundred cores in modern GPU accelerate LTL model checking procedure one hundred times?

GPU-Accelerated Systems vs. Single-Core Systems

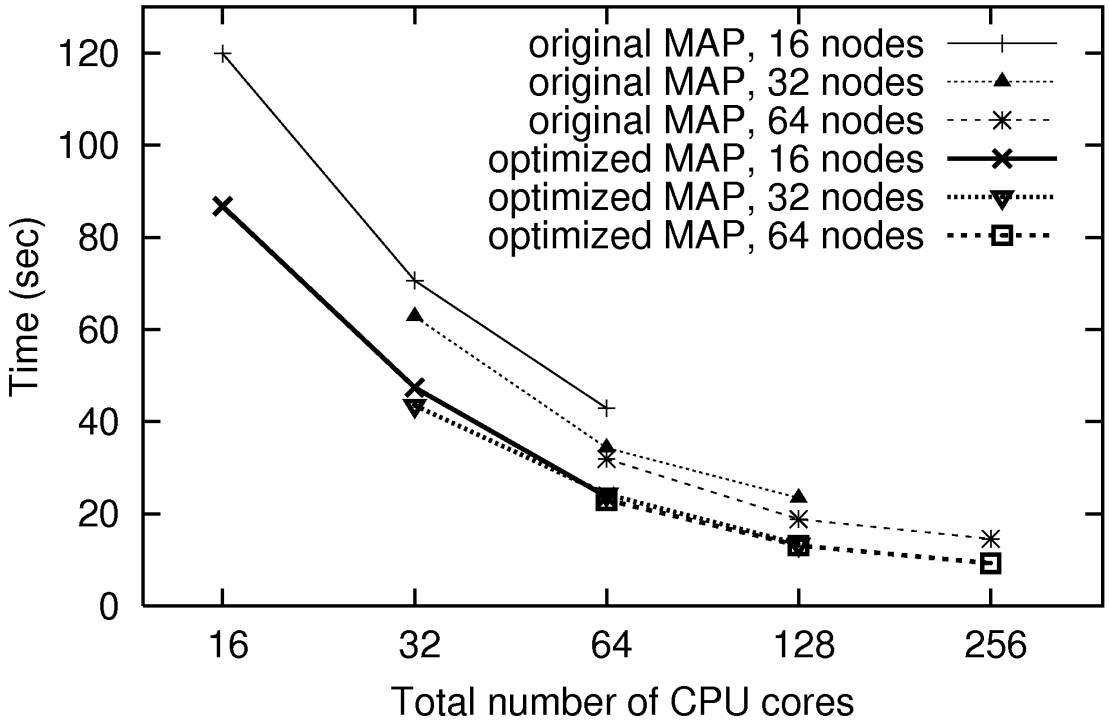
Model	acc. cycle	CUDA MAP			CPU MAP				CPU OWCTY	
		CSR time	*CUDA time*	*total time*	*1st iter. time*	*other iter. time*	*total time*	*# iter.*	*reach. time*	*total time*
elevator 1	N	27	7	34	44	56	100	16	24	41
leader	N	88	1	90	97	600	697	17	90	297
peterson 1	N	107	6	113	175	270	445	16	110	188
anderson	N	32	7	39	64	51	115	5	33	113
elevator 2	Y	34	1	35	50	–	50	1	41	177
phils	Y	47	1	47	295	102	397	5	180	576
peterson 2	Y	26	5	31	173	–	173	1	114	404
bakery	Y	25	1	26	240	–	240	1	219	907
Total time:		386 + 29 = 415			Total time:		2 173		Total time:	2 730
					Speedup:		5.24		Speedup:	6.51

Distributed-Memory Systems – OWCTY Scalability



Cores	Runtime (sec)	Efficiency
1	631.7	100%
64	13.3	74%
128	7.4	67%
256	5.0	49%

Distributed-Memory Systems – MAP Scalability



Cores	Runtime (sec)	Efficiency
1	1052.5	100%
64	23.0	72%
128	13.1	63%
256	8.9	46%

Experimental Validation – Conclusions

Changes in hardware design made some algorithms obsolete.

All tool producers should reconsider the algorithms their tools are using and possibly replace them with parallel ones.

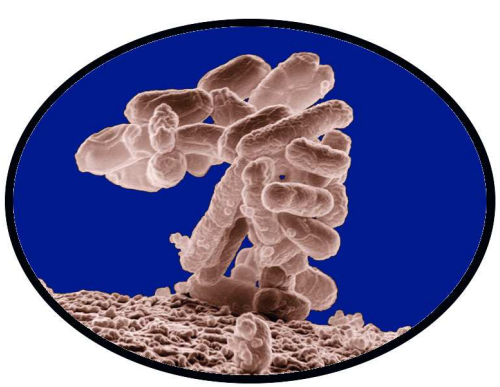
With new algorithms, we can squeeze all juice out of our HW in order to analyze very large systems.

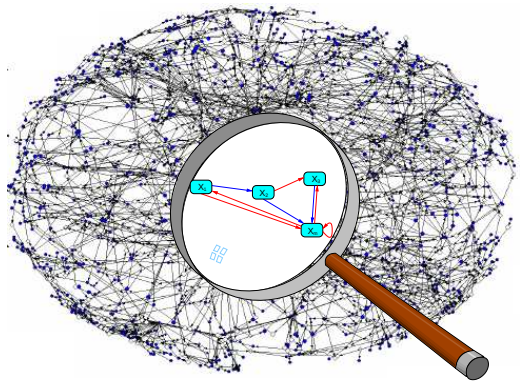
Outline

1. Introduction
2. LTL Model Checking
3. Parallel LTL Model Checking
4. Discrete Abstraction of ODE Models
5. Case Studies
   - Model Checking of E. Coli Ammonium Transport
   - Parameter Synthesis by Model Checking
   - Parameter Synthesis and Classification for Boolean Networks

Rectangular Abstraction: The Big Picture

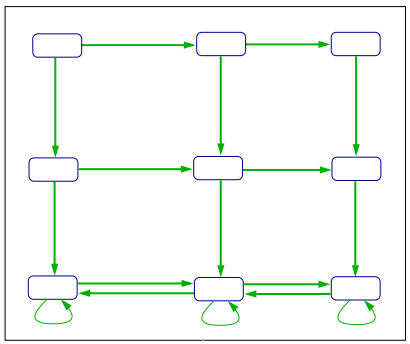
From a Continuous System to a Discrete Finite Quotient

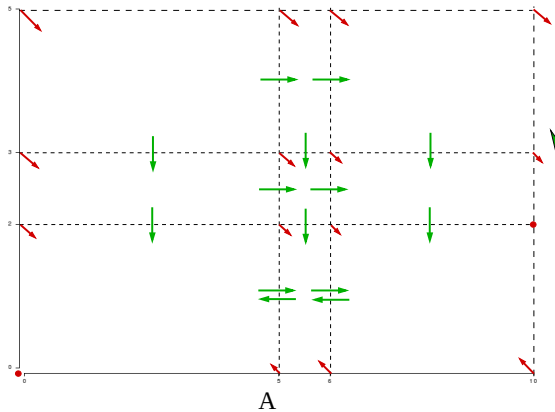


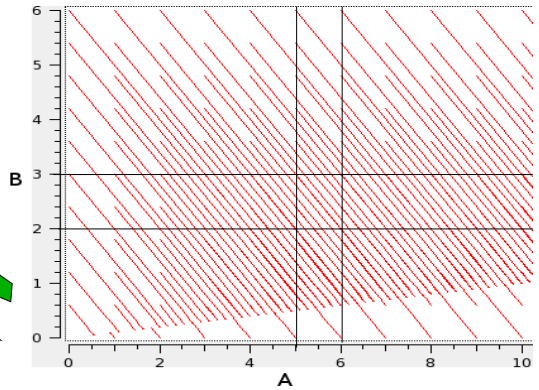


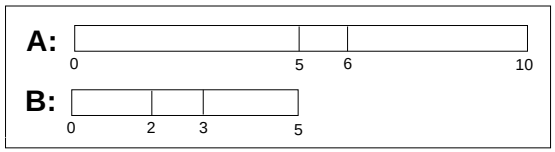
system of ODEs

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



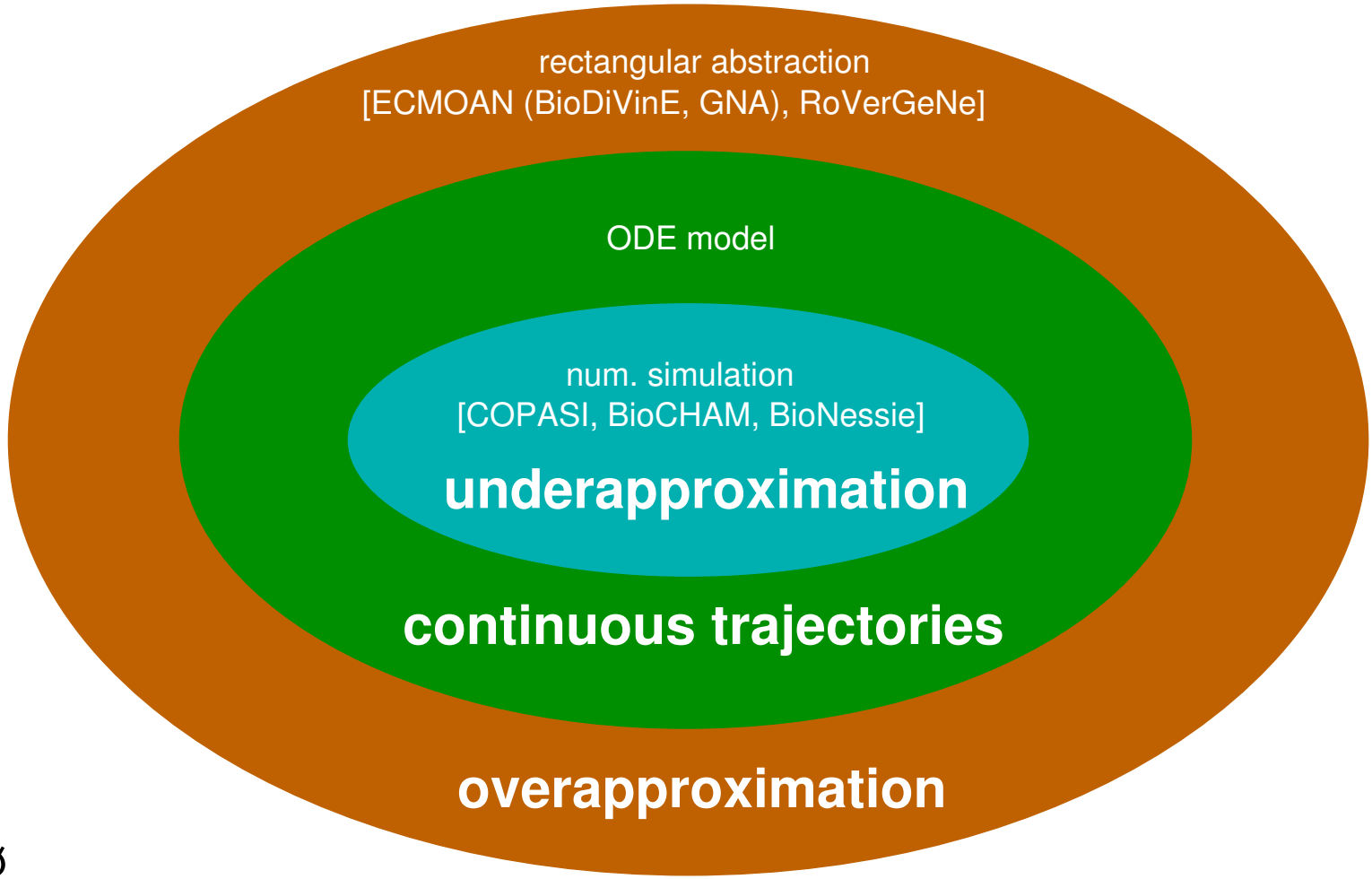






P. Collins, L. Habets, J.H. van Schuppen, I. Černá, J. Fabriková, and D. Šafránek. Abstraction of Biochemical Reaction Systems on Polytopes. In Proceedings of 18th IFAC World Congress, 2011.

Rectangular Abstraction: The Big Picture



$$NumSims(\mathcal{S}) \sqsubset Trajects(\mathcal{S}) \sqsubset QuotientPaths(\mathcal{S})$$

Rectangular Abstractions for Kinetic Models

Rectangular Abstraction of Reaction Kinetics

[Belta, Habets, Schuppen]

$$S + E \underset{k_2}{\overset{k_1}{\rightleftharpoons}} ES \xrightarrow{k_3} P + E$$

mass action kinetics

with limitation of 1 molecule per each reactant species

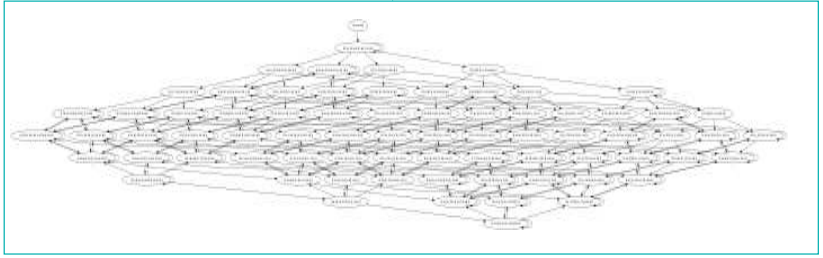
multi−affine ODEs

$$\frac{d[S]}{dt} = -k_1[E][S] + k_2[ES]$$
$$\frac{d[E]}{dt} = -k_1[E][S] + k_2[ES] + k_3[ES]$$
$$\frac{d[ES]}{dt} = k_1[E][S] - k_2[ES] - k_3[ES]$$
$$\frac{d[P]}{dt} = k_3[ES]$$

set discrete value domains per each variable

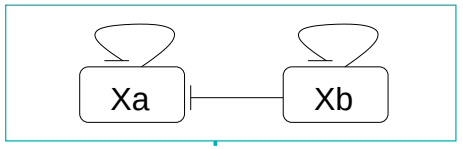
overapproximation by RATS

(Rectangular Abstraction Transition System)



Rectangular Abstraction of Regulatory Kinetics

[de Jong, Batt]



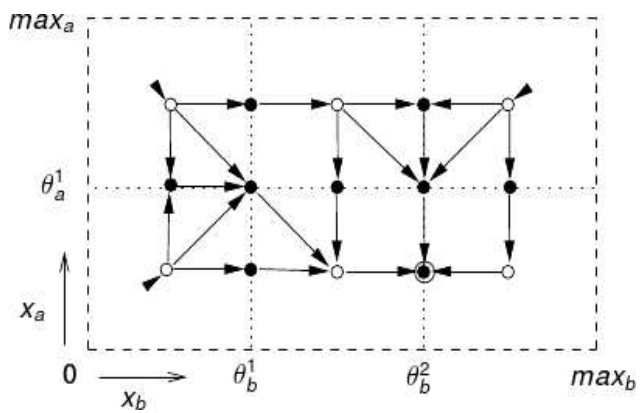
Hill kinetics

piece−wise affine ODEs

$$\frac{dx_a}{dt} = \kappa_a\, s^-(x_a, \theta_a^1)\, s^-(x_b, \theta_b^1) - \gamma_a\, x_a,$$
$$\frac{dx_b}{dt} = \kappa_b\, s^-(x_b, \theta_b^2) - \gamma_b\, x_b$$

overapproximation by RATS

(Rectangular Abstraction Transition System)



Rectangular Abstractions for Kinetic Models

Rectangular Abstraction of Reaction Kinetics

[Belta, Habets, Schuppen]

$$S + E \underset{k_2}{\overset{k_1}{\rightleftharpoons}} ES \xrightarrow{k_3} P + E$$

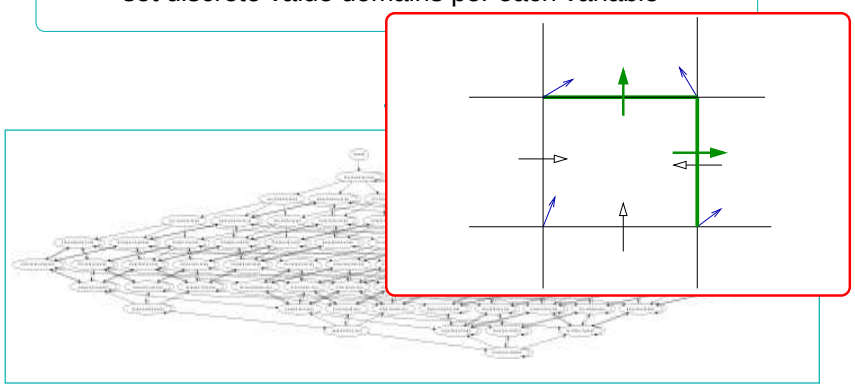
mass action kinetics

with limitation of 1 molecule per each reactant species

multi−affine ODEs

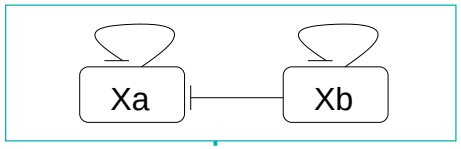
$$\frac{d[S]}{dt} = -k_1[E][S] + k_2[ES]$$
$$\frac{d[E]}{dt} = -k_1[E][S] + k_2[ES] + k_3[ES]$$
$$\frac{d[ES]}{dt} = k_1[E][S] - k_2[ES] - k_3[ES]$$
$$\frac{d[P]}{dt} = k_3[ES]$$

set discrete value domains per each variable



Rectangular Abstraction of Regulatory Kinetics

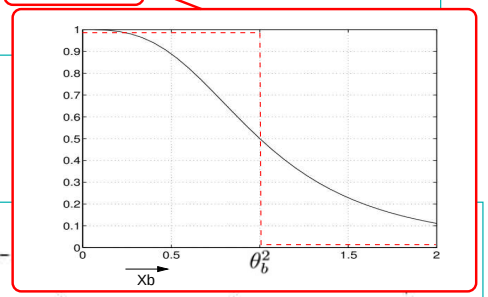
[de Jong, Batt]

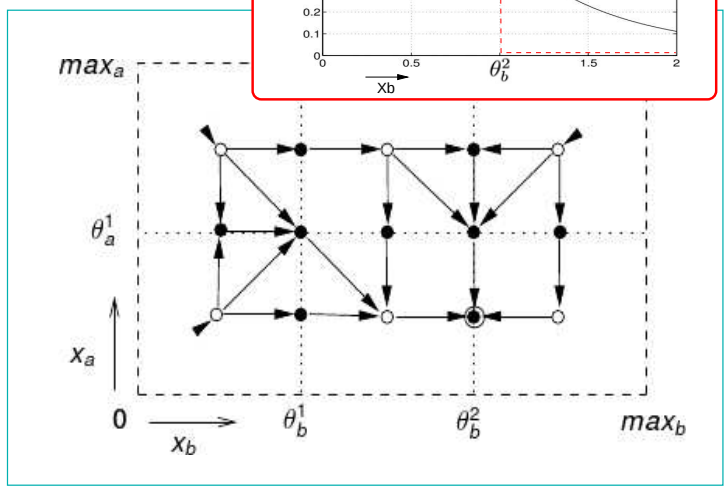


Hill kinetics

piece−wise affine ODEs

$$\frac{dx_a}{dt} = \kappa_a\, s^-(x_a, \theta_a^1)\, s^-(x_b, \theta_b^1) - \gamma_a\, x_a,$$
$$\frac{dx_b}{dt} = \kappa_b\, s^-(x_b, \theta_b^2) - \gamma_b\, x_b$$





Reaction Kinetics

- format of chemical reactions:

$$\gamma_1 X_1 + \cdots + \gamma_m X_m \to \delta_1 Y_1 + \cdots + \delta_n Y_n, \quad \gamma_i \in \{0,1\}, \delta_i \in \mathbb{N}$$

  note we expect $\{X_1, ..., X_m\} \cap \{Y_1, ..., Y_n\} = \emptyset$

- subclass of general **mass action kinetics**:

$$\forall i \in \{1, ..., n\}. \frac{dY_i}{dt} = g(X_1, ..., X_m) = \delta_i k X_1^{\gamma_1} X_2^{\gamma_2} \cdots X_m^{\gamma_m}$$

$$\forall i \in \{1, ..., m\}. \frac{dX_i}{dt} = g(X_1, ..., X_m) = -\gamma_i k X_1^{\gamma_1} X_2^{\gamma_2} \cdots X_m^{\gamma_m}$$

- corresponds to the class of multi-affine autonomous systems
- limitation: homodimerization $A + A \to AA$

Reaction Kinetics

- format of chemical reactions:

$$\gamma_1 X_1 + \cdots + \gamma_m X_m \to \delta_1 Y_1 + \cdots + \delta_n Y_n, \quad \gamma_i \in \{0,1\}, \delta_i \in \mathbb{N}$$

  note we expect $\{X_1, ..., X_m\} \cap \{Y_1, ..., Y_n\} = \emptyset$

- subclass of general **mass action kinetics**:

$$\forall i \in \{1, ..., n\}. \frac{dY_i}{dt} = g(X_1, ..., X_m) = \delta_i k X_1^{\gamma_1} X_2^{\gamma_2} \cdots X_m^{\gamma_m}$$

$$\forall i \in \{1, ..., m\}. \frac{dX_i}{dt} = g(X_1, ..., X_m) = -\gamma_i k X_1^{\gamma_1} X_2^{\gamma_2} \cdots X_m^{\gamma_m}$$

- corresponds to the class of multi-affine autonomous systems
- limitation: homodimerization $A + A \to AA$
- reactions of the form $X \to \delta_1 Y_1 + \cdots + \delta_n Y_n, \delta_i \in \mathbb{N}$ result in affine systems

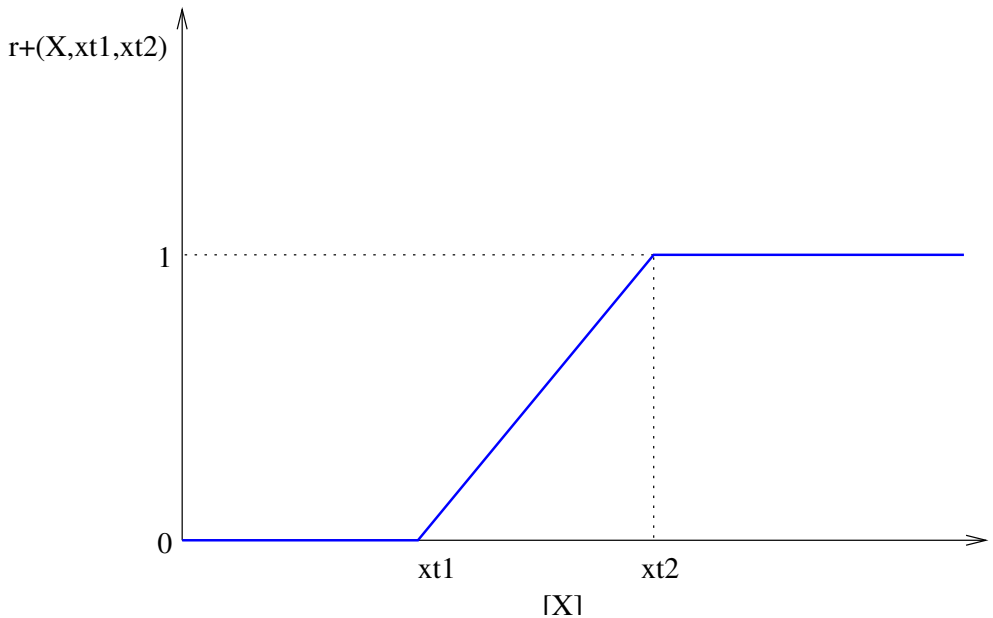
Regulatory Kinetics

- protein dynamics driven by protein-regulated transcription
- Hill kinetics approximated in terms of *ramp functions*

$$X \longrightarrow^{+} Y$$

$$\frac{dY}{dt} = kr^{+}(X, xt_1, xt_2)$$

- $k \in \mathbb{R}^{+}$ is *kinetic parameter*



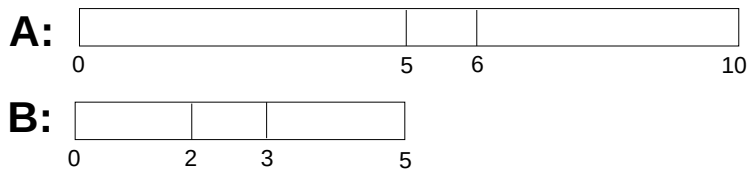
- ramp functions can describe cooperative regulations by means of summation and multiplication

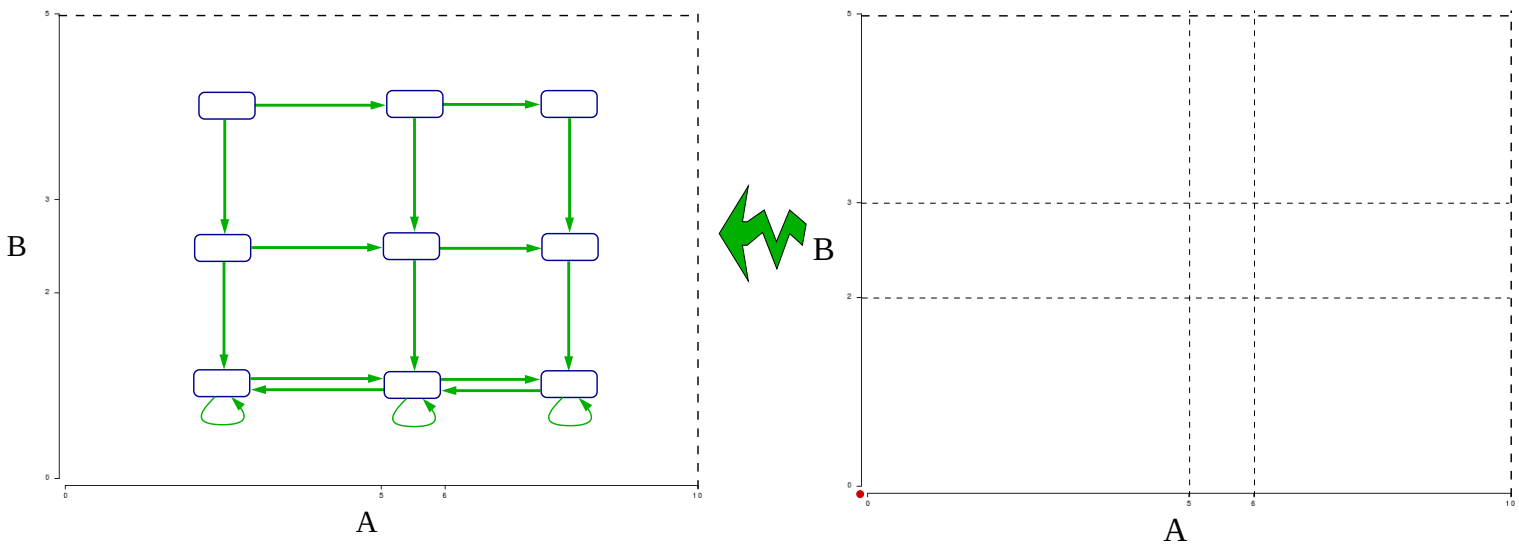
General Kinetic Models

- both kinetics combined
- right-hand side of any ODE is a mapping $g(\mathbf{x}, \mathbf{p})$ where $\mathbf{p}$ is a vector of unknown parameters
  - (piece-wise) multi-affine in $\mathbf{x}$
  - affine in $\mathbf{p}$
- these properties enable us to (are necessary to):
  - make a discrete finite overapproximation of the system dynamics
  - discretize the *parameter space* – possible values of $\mathbf{p}$
    $\Rightarrow$ synthesis of unknown parameters

$$A \overset{k_1}{\underset{k_2}{\leftrightarrows}} B$$

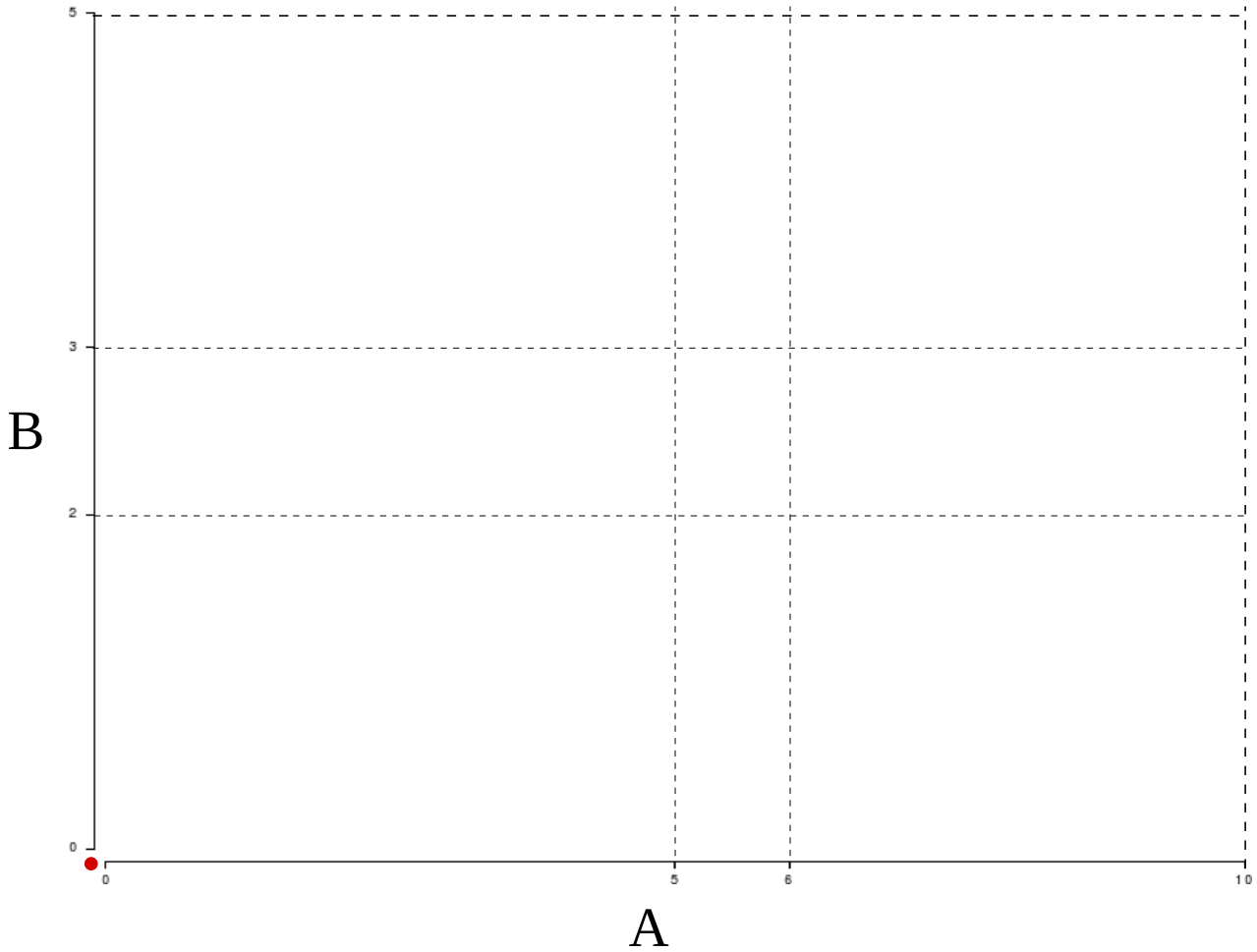
$$\frac{dA}{dt} = -k_1 \cdot A + k_2 \cdot B$$
$$\frac{dB}{dt} = k_1 \cdot A - k_2 \cdot B$$





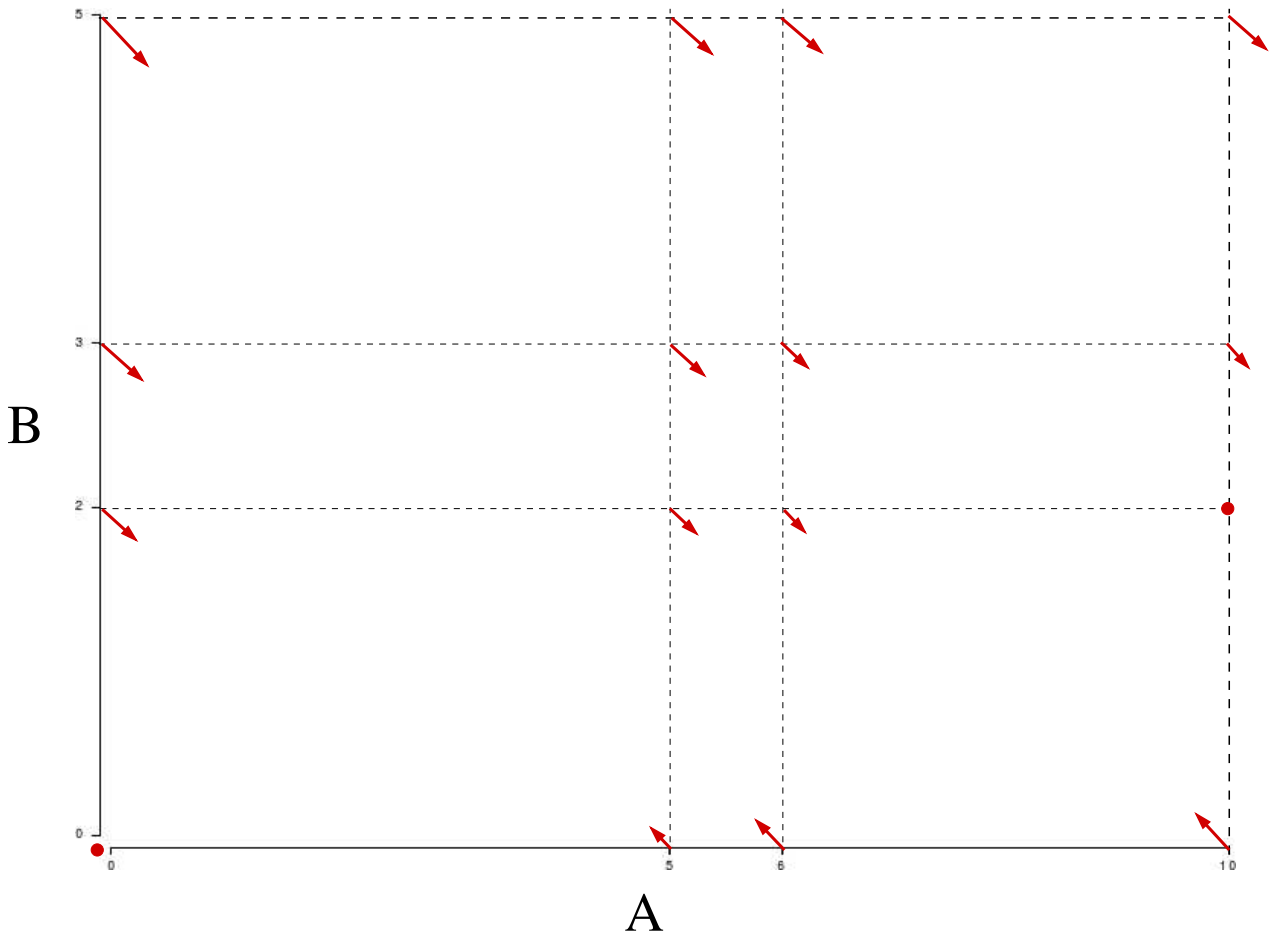
Rectangular Abstraction for Kinetic Models

- approach of [Belta, Habets, van Schuppen]
- continuous phase-space is bounded and abstracted by a non-deterministic automaton



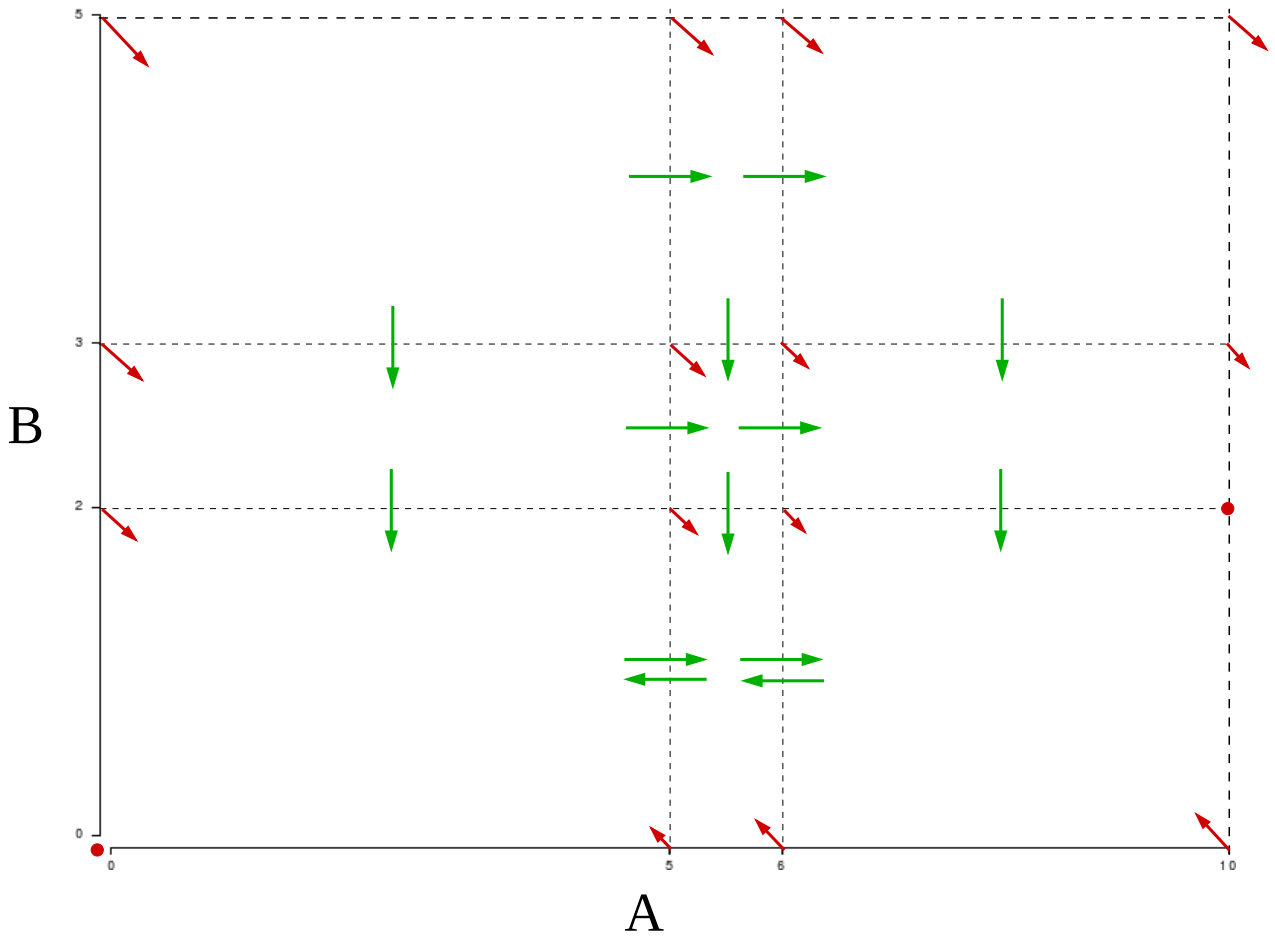
Rectangular Abstraction for Kinetic Models

- approach of [Belta, Habets, van Schuppen]
- continuous phase-space is bounded and abstracted by a non-deterministic automaton



Rectangular Abstraction for Kinetic Models

- approach of [Belta, Habets, van Schuppen]
- continuous phase-space is bounded and abstracted by a non-deterministic automaton



Definition

Let $X \subset \mathbb{R}^n$ be a closed full-dimensional polytope. A partitioning $X_{\text{part}}(X) = \{X_i \mid i = 1, \ldots, m\}$ of X is called *admissible* if

1. for all $i = 1, \ldots, m$: X_i is a closed full-dimensional polytope in $\mathbb{R}^n$,
2. $\cup_{i=1}^{m} X_i = X$,
3. for all $i, j = 1, \ldots, m$, $i \neq j$, the intersection $X_i \cap X_j$ is either empty, or a common face of X_i and X_j.

If both polytope X and all subpolytopes X_i, $(i = 1, \ldots, m)$ are n-dimensional rectangles, then an admissible partitioning $X_{\text{part}}(X)$ is called *rectangular*.

Definition

A mapping $g : X \to \mathbb{R}^n$ is called *piecewise-affine* on $X_{\text{part}}(X)$ if the following two conditions hold:

1. g is continuous on X
2. for all $i = 1, \ldots, m$ there exist $A_i \in \mathbb{R}^{n \times n}$ and $a_i \in \mathbb{R}^n$ such that for all $x \in X_i$: $g(x) = A_i x + a_i$, i.e. $g\,|_{X_i}$ is an affine mapping.

A mapping $g : X \to \mathbb{R}^n$ is called *multi-affine* on $X_{\text{part}}(X)$ if the following two conditions hold:

1. g is continuous on X
2. for all $i = 1, \ldots, m$, $g\,|_{X_i}$ is multi-affine, i.e. $g\,|_{X_i}$ is affine w.r.t. every of its variables, while keeping all other variables constant.

Definition

A *piecewise-affine system on a polytope* is a tuple

$$\chi = (X, X_{\text{part}}(X), x_0, t_0, g),$$

where state set X is a full-dimensional polytope in $\mathbb{R}^n$, $X_{\text{part}}(X)$ is an admissible partitioning of X, $x_0 \in X$ is the initial continuous state, $t_0 \in \mathbb{R}$ is the initial time, and $g : X \to \mathbb{R}^n$ is a piecewise-affine function on $X_{\text{part}}(X)$. A trajectory $x : [t_0, t_1] \to X$ of system χ is a solution of the differential equation

$$\dot{x}(t) = g(x(t)), \qquad x(t_0) = x_0, \tag{1}$$

where t_1 is either the time instant that the trajectory leaves state polytope X, or $t_1 = \infty$, if trajectory $x(t)$ remains in X forever.

Example

Consider the affine system Σ on rectangle $[0,2] \times [0,2]$ given by

$$\dot{x}(t) = \begin{pmatrix} -4 & 0 \\ 0 & -5 \end{pmatrix} x(t) + \begin{pmatrix} 6.8 \\ 6.5 \end{pmatrix}, \; x(t_0) = x_0.$$

Obviously, $(1.7, 1.3)^T$ is the unique steady state of this system. We partition the state set X into four squares:

$$\begin{array}{lllllll} X_{(0,0)} & = & [0,1] \times [0,1], & X_{(1,0)} & = & [1,2] \times [0,1], \\ X_{(0,1)} & = & [0,1] \times [1,2], & X_{(1,1)} & = & [1,2] \times [1,2]. \end{array}$$

Rectangular Abstraction for Kinetic Models

Note

One may distinguish systems with the same dynamics on all polytopes in the partitioning, and systems with different dynamics on each subpolytope. In the second case, the dynamics on the boundary of two polytopes is still assumed to be continuous.

Rectangular Abstraction for Kinetic Models

Note

One may distinguish systems with the same dynamics on all polytopes in the partitioning, and systems with different dynamics on each subpolytope. In the second case, the dynamics on the boundary of two polytopes is still assumed to be continuous.

Definition

If X is an n-dimensional rectangle, $X_{\mathrm{part}}(X)$ is a rectangular partioning of X, and $g : X \to \mathbb{R}^n$ is multi-affine on $X_{\mathrm{part}}(X)$, then $\chi = (X, X_{\mathrm{part}}(X), x_0, t_0, g)$ is called a *multi-affine system on rectangles*.

Exit Facet

Let $\chi = (X, X_{\text{part}}(X), x_0,\ t_0, g)$ be a piecewise-affine system on a polytope X. A facet F of subpolytope X_i is called an *exit facet* if there exists a trajectory of system Σ, starting in X_i, that attempts to leave X_i in finite time by crossing facet F.

Exit Facet

Let $\chi = (X, X_{\text{part}}(X), x_0,\ t_0, g)$ be a piecewise-affine system on a polytope X. A facet F of subpolytope X_i is called an *exit facet* if there exists a trajectory of system Σ, starting in X_i, that attempts to leave X_i in finite time by crossing facet F.

Observation

Let n_F denote the normal vector of F, pointing out of subpolytope X_i, and let the affine dynamics on X_i be described by $\dot{x} = A_i x + a_i$. Then F is an exit facet if and only if there exists $\hat{x} \in F$ such that

$$n_F^T (A_i \hat{x} + a_i) > 0. \tag{2}$$

Since the dynamics $\dot{x} = A_i x + a_i$ is affine, it suffices to check condition (2) on $\mathcal{V}(F)$, i.e. on the set of all vertices of facet F.

Problem

On which facet the trajectory exits a polytope X_i?

- if the trajectory leaves X_i through a point in the relative interior of a facet F, then it continues to an adjacent polytope X_j such that $X_i \cap X_j = F$,
- if it leaves through a point on a lower-dimensional face, a problem arises since the face can be shared by more than two polytopes
  $\Rightarrow$ this possibility is excluded and considered as singular (it is replaced by a sequence of several adjacent transitions executed in the same time instant)

Problem

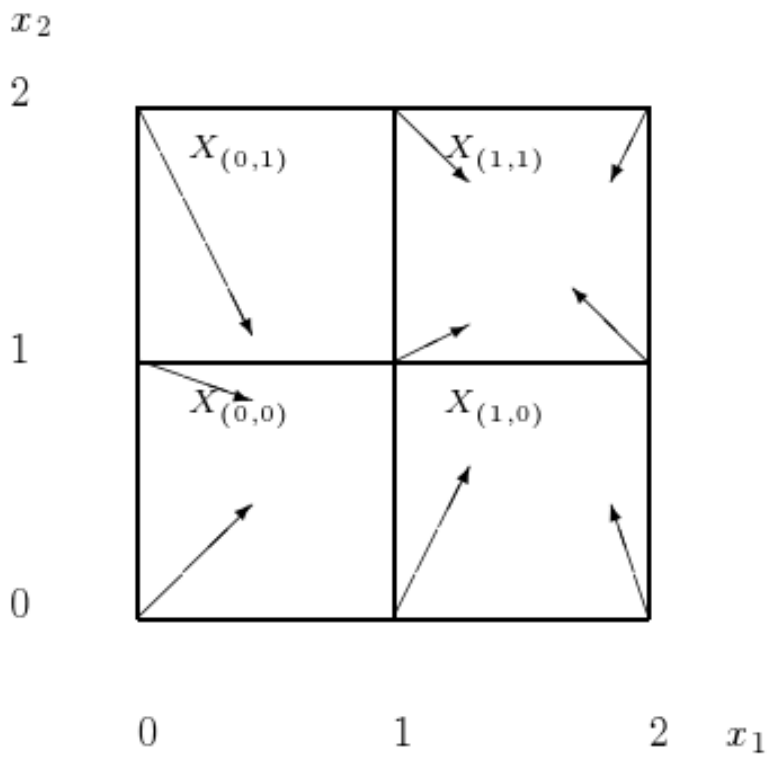
On which facet the trajectory exits a polytope X_i?

- if the trajectory leaves X_i through a point in the relative interior of a facet F, then it continues to an adjacent polytope X_j such that $X_i \cap X_j = F$,
- if it leaves through a point on a lower-dimensional face, a problem arises since the face can be shared by more than two polytopes
  $\Rightarrow$ this possibility is excluded and considered as singular (it is replaced by a sequence of several adjacent transitions executed in the same time instant)

Note

The rectangular abstraction abstracts from time.

Example



Problem

Does the trajectory leave a polytope X_i in finite time?

Theorem [Habets, Collins, Schuppen 2006]

Consider an affine system $\dot{x}(t) = A_i x(t) + a_i$ on a closed full-dimensional subpolytope $X_i \subset \mathbb{R}^n$. There exists an initial state $x_0 \in X_i$ such that for all times $t \in T = [t_0, \infty)$ the state trajectory belongs to the subpolytope, i.e. $x(t; t_0, x_0) \in X_i$ if and only if there exists a fixed state in subpolytope X_i.

Problem

Does the trajectory leave a polytope X_i in finite time?

Lemma

Consider an affine system $\dot{x}(t) = A_i x(t) + a_i$ on a closed full-dimensional subpolytope $X_i \subset \mathbb{R}^n$. There exists an $\hat{x} \in X_i$ such that $A_i \hat{x} + a_i = 0$ if and only if

$$0 \in \mathrm{ConvexHull}(\{A_i v + a_i \mid v \in \mathcal{V}(X_i)\}), \tag{3}$$

i.e. if and only if the zero vector is a convex combination of the direction vectors at the vertices.

- alternatively numerical approaches can be used

1. Subpolytope X_i contains a fixed point, and at all vertices of X_i, the direction vector of the differential equation is pointing inward. In this case all trajectories that enter subpolytope X_i will remain in X_i forever.
2. Subpolytope X_i does not contain a fixed point. Then all trajectories that enter X_i leave X_i in finite time.
3. Subpolytope X_i contains a fixed point, and there exists a vertex of X_i where the direction vector of the differential equation is pointing out of X_i.
   I.e., there exist trajectories that leave X_i and also trajectories that do not

Let $\chi = (X, X_{\text{part}}(X), x_0, t_0, g)$ a piecewise-affine system, $N = |X_{\text{part}}(X)|$. We construct a Kripke structure $K_\chi = (S, S_0, T, L)$ representing the *rectangular abstraction* of χ:

- $S = \{s_1, ..., s_N\}$ and we define a bijective map $\Pi : X_{\text{part}}(X) \to S$ such that $\Pi(X_i) = s_i$,
- $S_0 = \{s_i\}$ such that $x_0 \in \Pi^{-1}(s_i)$ and $x(t; t_0, x_0) \in \Pi^{-1}(s_i)$ for all $t \in (t_0, t_0 + \epsilon)$ for some $\epsilon > 0$
- $(s_i, s_i) \in T$ if there exists $\hat{x}$ such that $g(\hat{x}) = 0$
- for every facet $F = X_i \cap X_j$ for that there exists a vertex $v \in \mathcal{V}$ satisfying $n_F^T g(v) > 0$, $(\Pi(X_i), \Pi(X_j)) \in T$

Rectangular abstraction can be employed also for (piecewise) multi-affine systems (proved only for rectangular polytopes).

C. Belta, L.C.G.J.M. Habets, and V. Kumar. "Control of multi-affine systems on rectangles with applications to hybrid biomolecular networks." In Proc. 41th IEEE Conf. on Decision and Control, pages 534–539, New York, 2002. IEEE Press.

Problem

A sufficient and necessary condition for exiting a rectangle in finite time is not known.

Theorem

Let $\dot{x}(t) = g(x(t))$ be a multi-affine system on an n-dimensional rectangle $R_i \subset \mathbb{R}^n$. If there exists a vector $w \in \mathbb{R}^n$ such that for all vertices $v \in \mathcal{V}(R_i)$ we have $w^T g(v) > 0$, then all state trajectories of this system leave rectangle R_i in finite time.

Rectangular Abstraction for Kinetic Models

Let $\chi = (X, X_{\text{part}}(X), x_0, t_0, g)$ a piecewise-affine (or piecewise multi-affine) system and K_χ its rectangular abstraction.

Global necessity

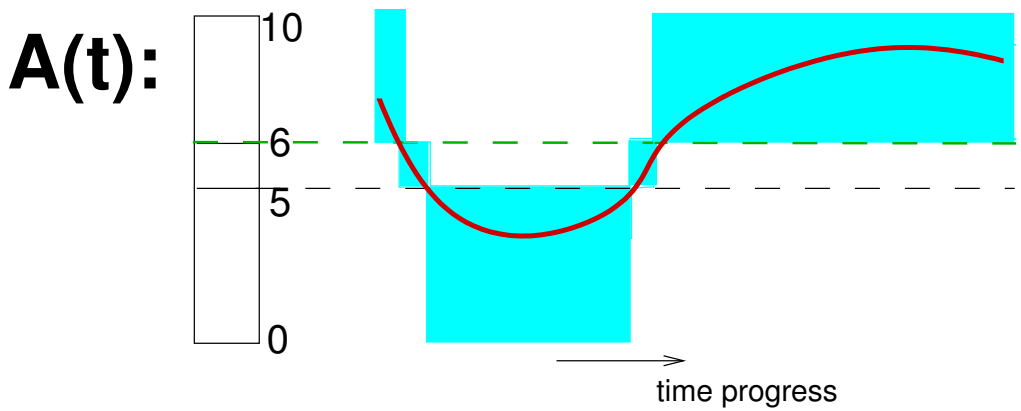
If for every path $\pi = s_0, \ldots$ in K_χ, $s_0 \in S_0$, there exists an initial point $x_0 \in \Pi(s_i)$ such that the trajectory $x(t; t_0, x_0)$ of χ corresponds to π, i.e., $x \subseteq \bigcup_{s_j \in \pi}(\Pi^{-1}(s_j))$.

Global sufficiency

If for every trajectory $x = x(t; t_0, x_0)$ of χ there exists a path $\pi = s_0, \ldots$ for some $s_0 \in S_0$ such that $x_0 \in \Pi(s_0)$ and x corresponds to π.

Temporal Properties for the Abstraction Kripke Structure

- reachability
  - *global*: regardless the initial state, B eventually falls below 2
  - *local*: if B initally below 2 then B does not exceed 2
- temporal properties
  - there is no initial state from which A falls below 6 before A exceeds 6



- properties defined by ω-regular languages
- many useful properties can be formulated in LTL
- some properties may require branching time (e.g., reachability of multiple steady state)

Rectangular Abstraction and Model Checking

Let K_χ be a rectangular automaton for a system χ that is either (piecewise) affine or (piecewise) multi-affine. Let φ be an ω-regular property.

- global sufficiency holds
  - $K_\chi \models \varphi \implies \chi$ preserves φ
- global necessity does not hold
  - $K_\chi \not\models \varphi$ does not necessarily imply "χ does not preserve φ"
  - there might exist paths in K_χ for which there is no trajectory in S, the reasons are of two kinds:
    1. the abstraction combines behaviour of different trajectories
       $\Rightarrow$ in piecewise-affine and multi-affine systems
    2. known condition for exiting a rectangle in finite time is not sufficient
       $\Rightarrow$ in multi-affine systems

P. Collins, L. Habets, J.H. van Schuppen, I. Cerna, J. Fabrikova, and D. Šafránek. "Abstraction of Biochemical Reaction Systems on Polytopes", In Proceedings of the 18th IFAC World Congress. IFAC, 2011. pages 14869-14875

Other Approach

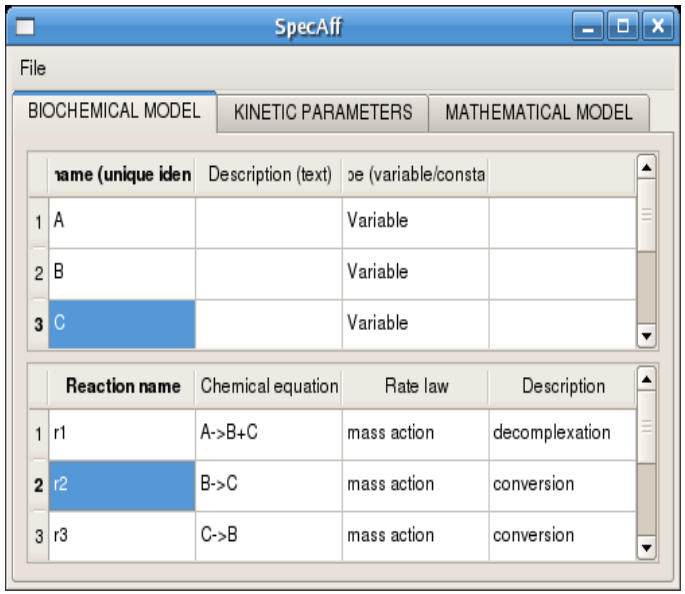
- regulatory kinetics abstracted by step functions
- results in a piecewise-affine abstraction with different dynamics on individual rectalngles
- gives a qualitative abstraction that is an overapproximation of original system
- faces must be also included in the abstraction, trajectories are not continuous on faces
  $\Rightarrow$ large state spaces, more expensive successor function
- good representation of regulatory logic (the extent of overapproximation is reasonable)

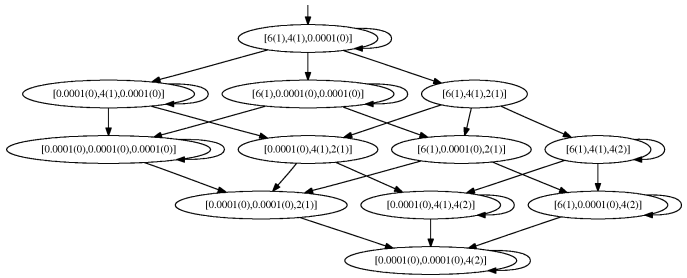
H. de Jong, J.-L. Gouzé, C. Hernandez, M. Page, T. Sari, J. Geiselmann (2004), Qualitative simulation of genetic regulatory networks using piecewise-linear models, Bulletin of Mathematical Biology, 66(2):301-340.

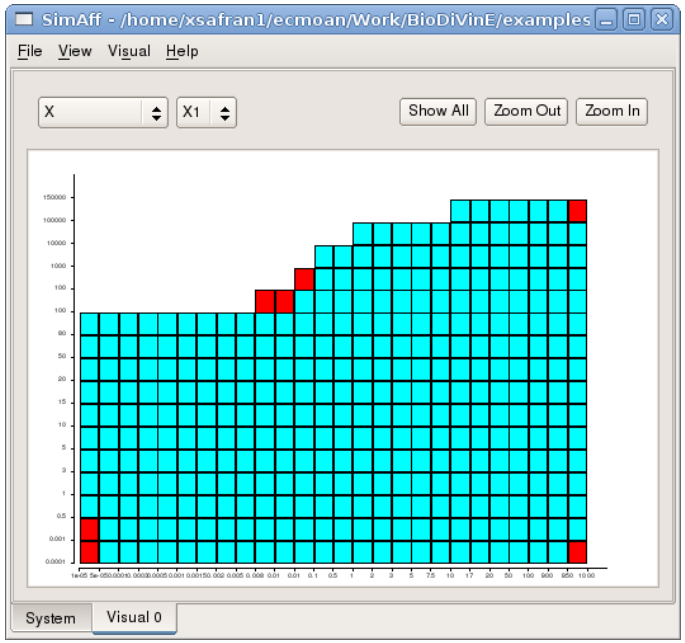
Outline

1. Introduction
2. LTL Model Checking
3. Parallel LTL Model Checking
4. Discrete Abstraction of ODE Models
5. Case Studies
   - Model Checking of E. Coli Ammonium Transport
   - Parameter Synthesis by Model Checking
   - Parameter Synthesis and Classification for Boolean Networks

BioDiVinE Toolset





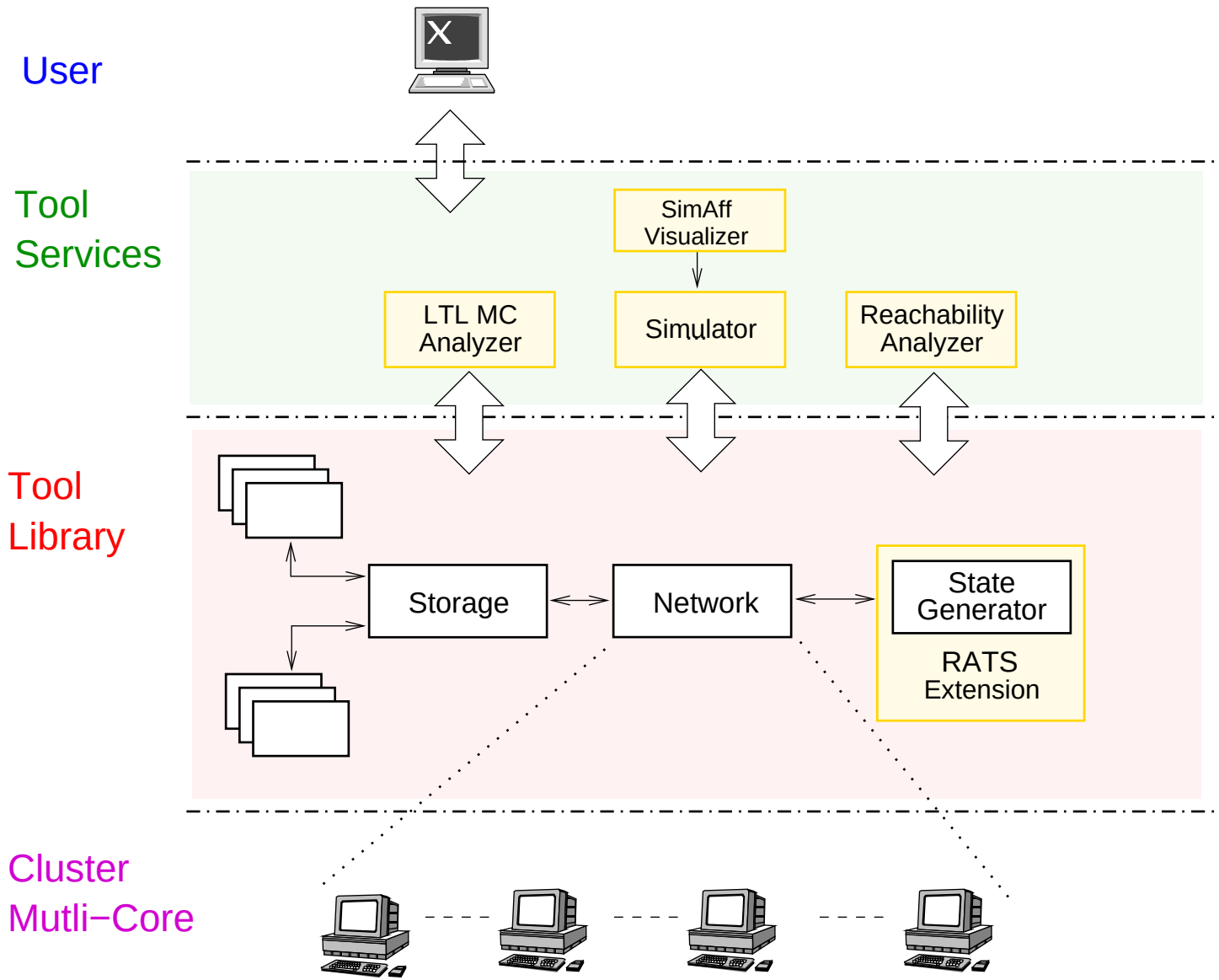


- input:
  - textual: internal .bio format
    – ODEs + LTL property
  - gui: list of chemical reactions; SBML standard
- tasks:
  - rectangular abstraction
  - parallel LTL model checking
- output:
  - model checking counterexample
  - 2D reachability visualization

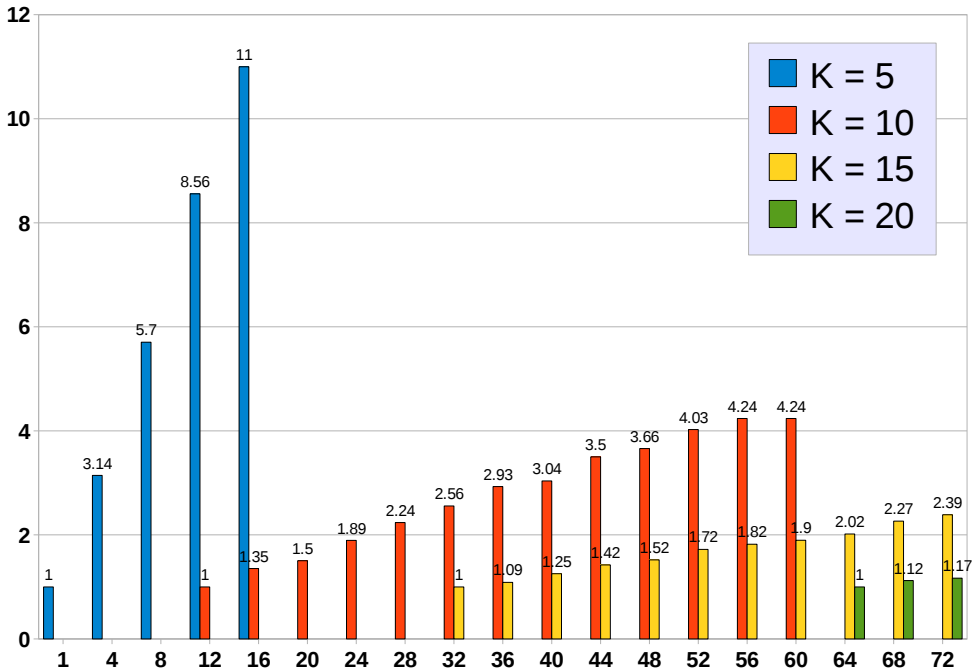
`http://anna.fi.muni.cz/~xdrazan/biodivine/`

J. Barnat, L. Brim, and D. Šafránek. "High-performance analysis of biological systems dynamics with the DiVinE model checker." Briefings in Bioinformatics 11(3):301-12 (2010)

BioDiVinE Toolset Architecture



Experimental Results
Rectangular abstraction



k	States	Trans
5	$3 \cdot 10^4$	$8.5 \cdot 10^4$
10	$9 \cdot 10^5$	$3.2 \cdot 10^6$
15	$1.6 \cdot 10^6$	$6.5 \cdot 10^6$
20	$3.2 \cdot 10^6$	$1.4 \cdot 10^7$

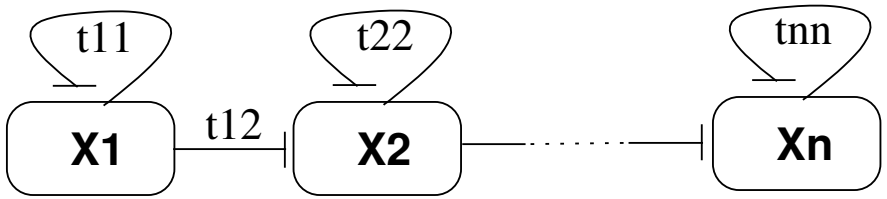
$$S + E \rightleftharpoons ES_1 \rightleftharpoons ES_2 \rightleftharpoons \cdots \rightleftharpoons ES_K \rightarrow P + E$$

$$E > 95 \wedge (E > 95\mathbf{U}(E =< 95 \wedge (E <= 95\mathbf{U}E > 95)))$$

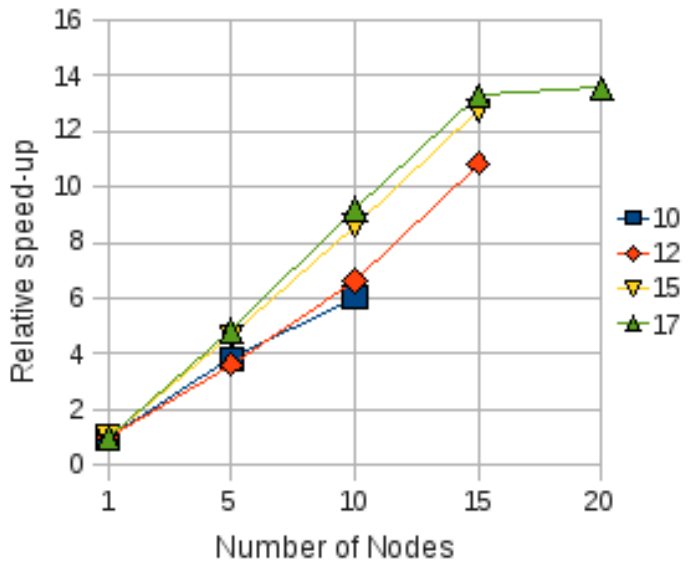
J. Barnat, L. Brim, I. Cerna, S. Drazan, J. Fabrikova, J. Lanik, H. Ma, and D. Safranek. BioDiVinE: A Framework for Parallel Analysis of Biological Models. In Proceedings of 2nd International Workshop on Computational Models for Cell Processes (COMPMOD 2009), pp. 31-45, EPTCS 6, 2009.

Experimental Results

Piece-wise linear abstraction with different dynamics on rectangles



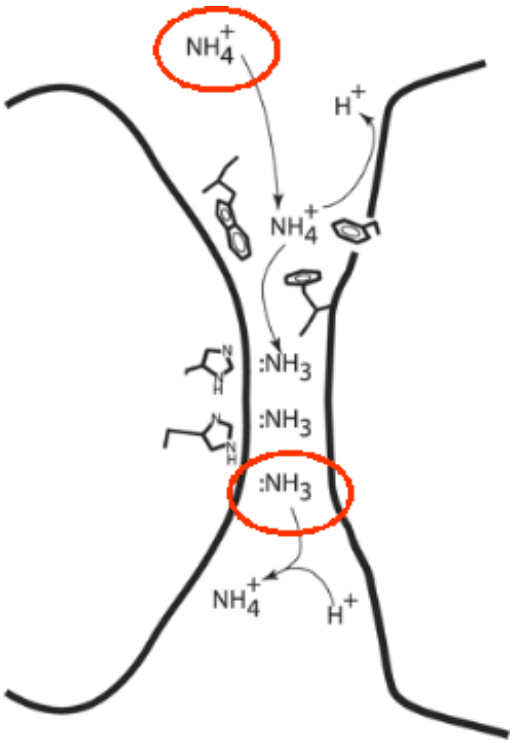
$$\neg((X_n < t_{nn} \rightarrow \mathbf{F}(X_n \geq t_{nn})) \wedge (X_n \geq t_{nn} \rightarrow \mathbf{F}(X_n < t_{nn})))$$

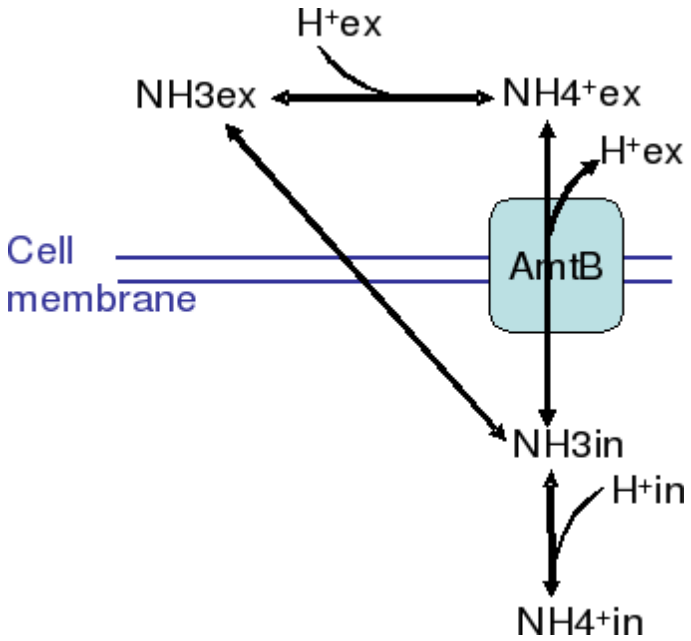


J. Barnat, L. Brim, I. Cerna, S. Drazan, J. Fabrikova, and D. Safranek. On Algorithmic Analysis of Transcriptional Regulation by LTL Model Checking. In Theoretical Computer Science 410, pp. 3128-3148, 2009.

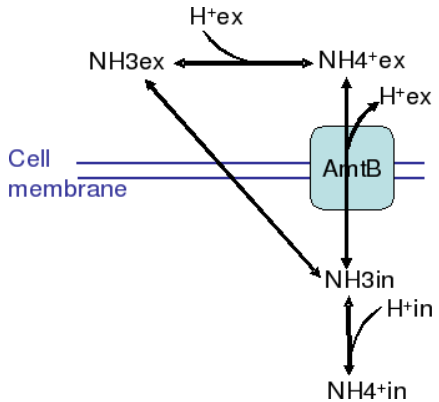
H. de Jong, J.-L. Gouzé, C. Hernandez, M. Page, T. Sari, J. Geiselmann (2004), Qualitative simulation of genetic regulatory networks using piecewise-linear models, Bulletin of Mathematical Biology, 66(2):301-340.

E. Coli Ammonium Transport Model





E. Coli Ammonium Transport Model



$$AmtB + NH_4ex \overset{k_1}{\underset{k_2}{\leftrightarrows}} AmtB : NH_4 \qquad k_1 = 5 \cdot 10^8, k_2 = 5 \cdot 10^3$$

$$AmtB : NH_4 \overset{k_3}{\rightarrow} AmtB : NH_3 + H_{ex} \qquad k_3 = 50$$

$$AmtB : NH_3 \overset{k_4}{\rightarrow} AmtB + NH_3in \qquad k_4 = 50$$

$$NH_4in \overset{k_5}{\rightarrow} \qquad k_5 = 80$$

$$NH_3in + H_{in} \overset{k_6}{\underset{k_7}{\leftrightarrows}} NH_4in \qquad k_6 = 1 \cdot 10^{15}, k_7 = 5.62 \cdot 10^5$$

$$NH_3ex \overset{k_8}{\underset{k_9}{\leftrightarrows}} NH_3in \qquad k_8 = k_9 = 1.4 \cdot 10^4$$

E. Coli Ammonium Transport: Model Settings

Settings

- mass action kinetics $\Rightarrow$ multi-affine ODE model
- kinetic parameters set w.r.t. literature
- internal and external pH conditions considered constant
- initial conditions set to intervals:

$AmtB$, $AmtB:NH_3$, $AmtB:NH_4$	NH_3in	NH_4in	NH_3ex, NH_4ex
$\langle 0, 1 \cdot 10^{-5} \rangle$	$\langle 1 \cdot 10^{-6}, 1.1 \cdot 10^{-6} \rangle$	$\langle 2 \cdot 10^{-6}, 2.1 \cdot 10^{-6} \rangle$	$\langle 0, 1 \cdot 10^{-5} \rangle$

- abstraction – number of discrete concentration levels considered:

$AmtB$	$AmtB:NH_3$	$AmtB:NH_4$	NH_3in	NH_4in
7	9	3	8	26

E. Coli Ammonium Transport Model – Results

Analyzed Properties

- what are the maximal reachable levels of NH_3in and NH_4in?
  - (A) find the lowest α satisfying $\mathbf{G}(NH_3in < \alpha)$
  - (B) find the lowest β satisfying $\mathbf{G}(NH_4^+in < \beta)$

α	$\mathbf{G}(NH_3in < \alpha)$	# states	Time (# nodes)
$1.1 \cdot 10^{-6}$	true	$1.5 \cdot 10^5$	1.9 s (18)

β	$\mathbf{G}(NH_4in < \beta)$	# states	Time (# nodes)
$1 \cdot 10^{-3}$	true	$1.6 \cdot 10^5$	2 s (18)
$5 \cdot 10^{-4}$	false	$2.7 \cdot 10^5$	3 s (18)
$6 \cdot 10^{-4}$	true	$1.5 \cdot 10^5$	1.8 s (18)
$5.4 \cdot 10^{-4}$	true	$2.1 \cdot 10^5$	4.2 s (18)
$5.3 \cdot 10^{-4}$	false	$2.7 \cdot 10^5$	2.2 s (18)

J. Barnat, L. Brim, I. Cerna, S. Drazan, J. Fabrikova, J. Lanik, H. Ma, and D. Safranek. BioDiVinE: A Framework for Parallel Analysis of Biological Models. In Proceedings of 2nd International Workshop on Computational Models for Cell Processes (COMPMOD 2009), pp. 31-45, EPTCS 6, 2009.

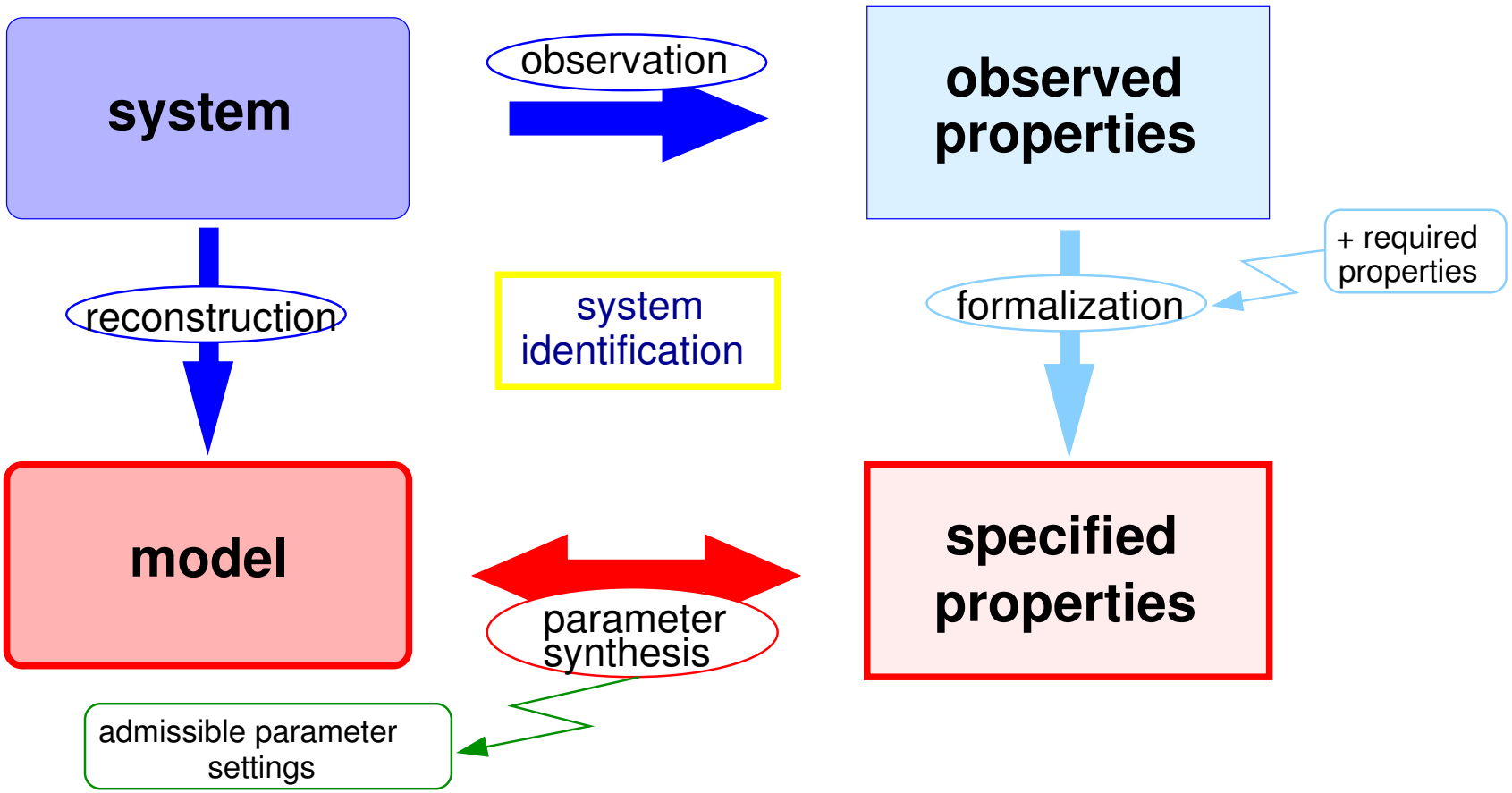
E. Coli Ammonium Transport Model – Results

Additional Property

At which level NH_3ex starts to affect NH_3in?

α	$NH_3ex < \alpha \Rightarrow \mathbf{G}(NH_3in < 1.1 \cdot 10^5)$	# states	Time (# nodes)
$19.5 \cdot 10^{-4}$	true	$1.4 \cdot 10^5$	1.9 s (36)
$19.6 \cdot 10^{-4}$	false	$3.4 \cdot 10^5$	5.9 s (36)

Parameter Synthesis



Problem Definition

Robustness

Given an LTL property φ and a parameterized model $\mathcal{M}$ check if $\mathcal{M}(p) \models \varphi$ **holds for all possible parameterizations** $p \in \mathcal{P}$ (valuations of parameters), $\mathcal{P}$ is called the **parameter space**.

Parameter Synthesis Problem

Given an LTL property φ and a parameterized model $\mathcal{M}$ **find the maximal set** $P \subseteq \mathcal{P}$ **of parameterizations** such that $\mathcal{M}(p) \models \varphi$ for all $p \in P$.

Problem Reduction

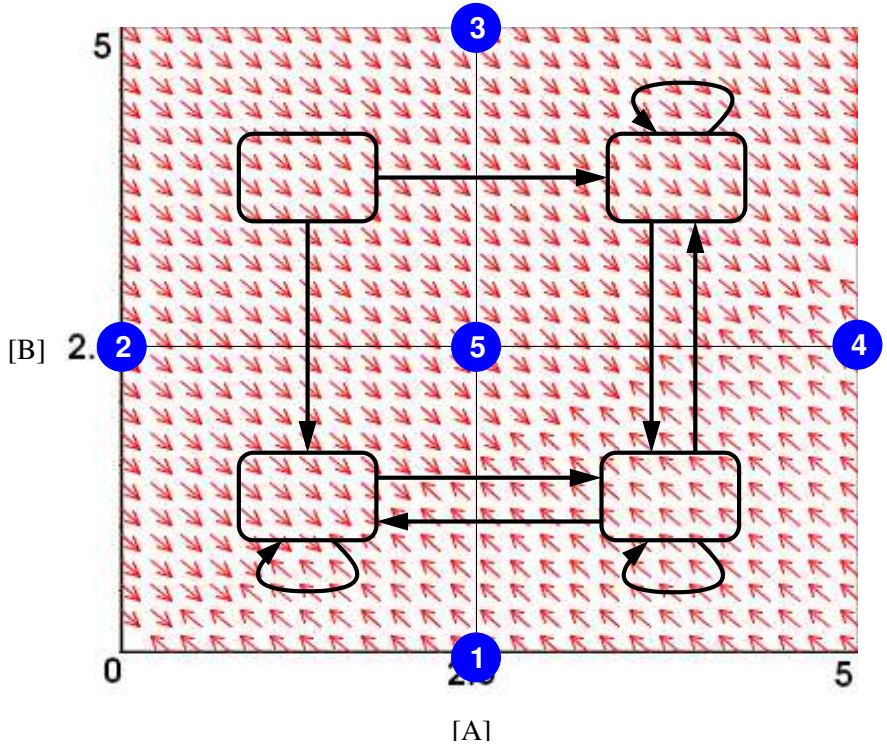
Robustness is reduced to Parameter Synthesis Problem by taking the set $\mathcal{P}$ of all possible parameterizations as P.

Computing the Parameter Space

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

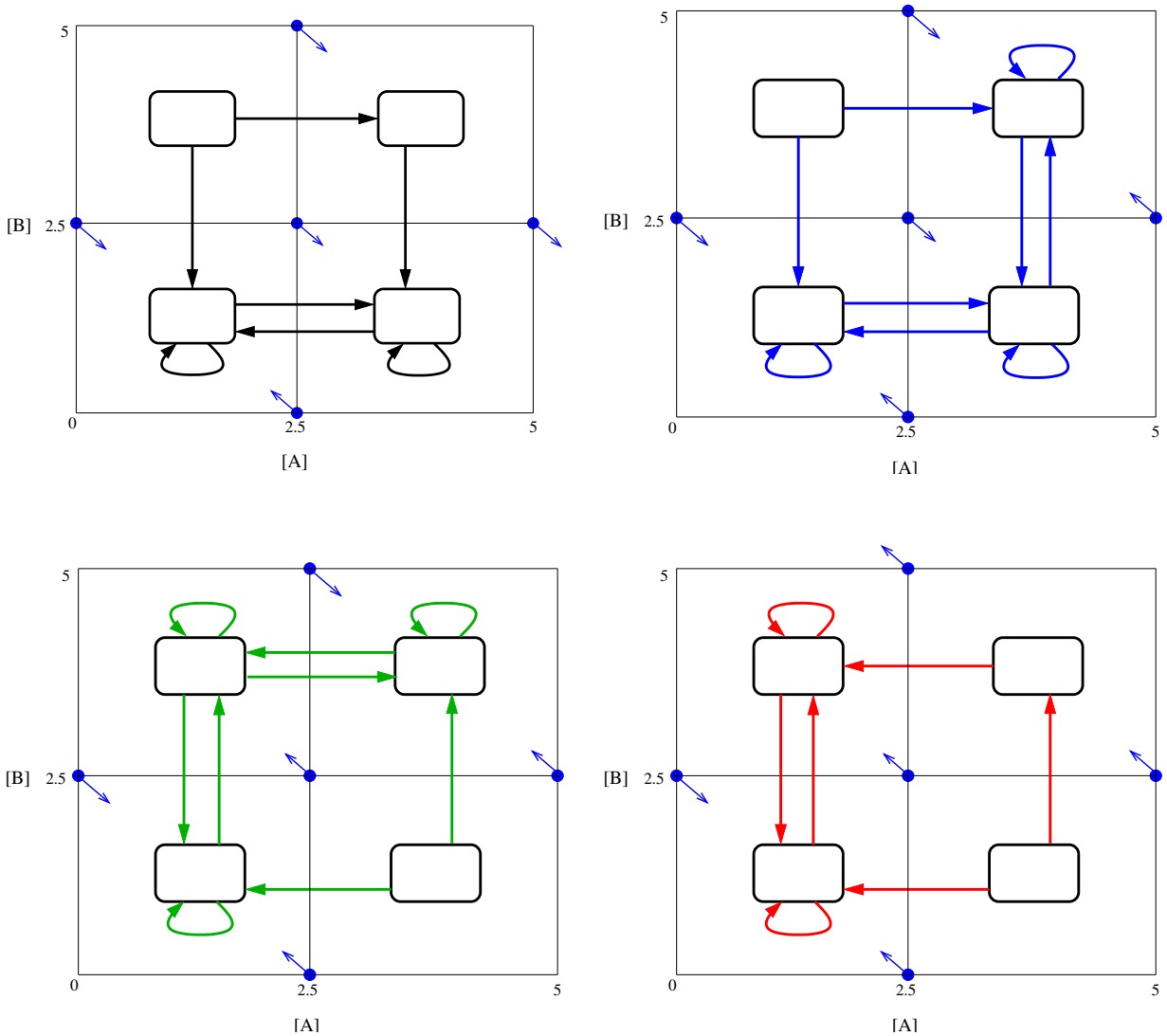
$k_2 = 0.8$

$k_1 = ?$

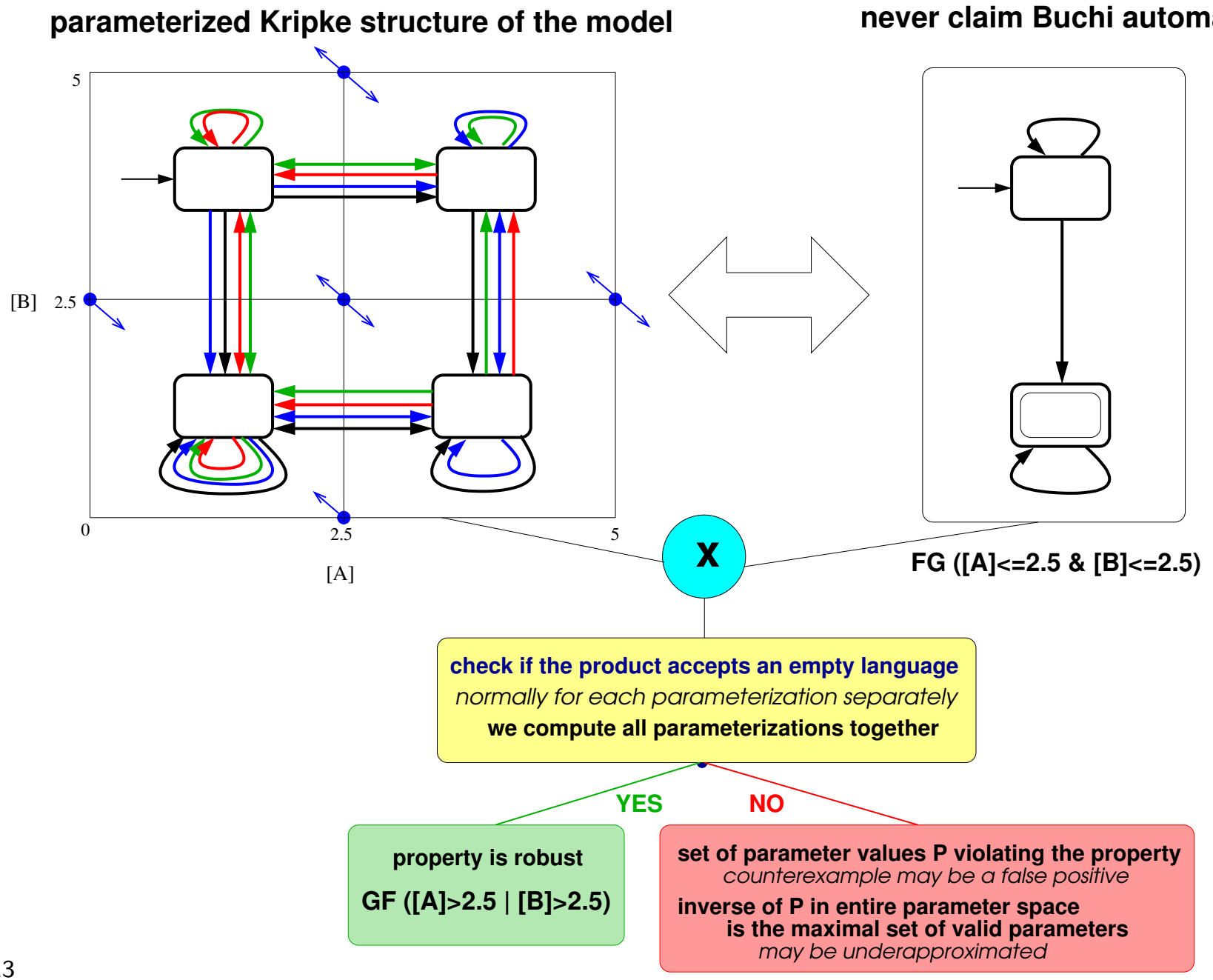


	value of k1: (0,0.4)	(0.4,0.8)	(0.8,1.6)	(1.6,max)
1	↖	↖	↖	↖
2	↘	↘	↘	↘
3	↘	↘	↘	↖
4	↘	↖	↖	↖
5	↘	↘	↖	↖

Effect of Parameters on Abstraction Automaton



Parameter Synthesis by LTL Model Checking



Model Checking of Parameterized Kripke Structures

Idea

- for a model $\mathcal{M}$ and finite *parameter space* $\mathcal{P}$ consider $K_{\mathcal{M}} = (\mathcal{P}, S, S_0, T, L)$ a *parametrized Kripke structure*
- represent each parameterization by a distinct colour $p \in \mathcal{P}$
- assume all transitions for each parameterization adequately coloured
- find accepting cycles and get colours enabling accepting runs

Procedure

1. construct the parametrized product BA of $K_{\mathcal{M}}$ and the property BA
2. compute initial mapping of colours to states (state coloring)
   $\Rightarrow$ propagate colours through the entire graph (BFS reachability)
   $\Rightarrow$ states on accepting cycles know all colours by which they are reached
3. for each reachable accepting cycle aggregate (scan) the valid colours

State Coloring

Let $\mathcal{P}$ denotes the set of all parameterizations. Further let $\mathcal{K} = (\mathcal{P}, S, \mathcal{P} \times \Sigma, S_0, \delta, F)$ a parameterized product BA and let $\alpha, \gamma \in S$, $P \subseteq \mathcal{P}$.

$$Succ(\gamma, P)(\alpha) = \{p \in P \mid \gamma \xrightarrow{p}+ \alpha\}$$

$$\forall S' \subseteq S.\, Succ(S', P) = \bigcup_{\gamma \in S'} Succ(\gamma, P)$$

Initial coloring:

$$Succ(S_0, \mathcal{P})$$

Transition-enabling colours:

$$P(\alpha, \beta) = \{p \in \mathcal{P} \mid \alpha \xrightarrow{p} \beta\}$$

Note

$\alpha \xrightarrow{p} \beta$ denotes $\beta \in \delta(\alpha, \langle p, L(\alpha)\rangle)$ where $p \in \mathcal{P}$, $L(\alpha)$ is omitted to simplify the notation.

State Coloring Computation

Compute $Succ(S', P)$ over the PKS $\mathcal{K}$:

Require: $\mathcal{K} = (\mathcal{P}, S, \mathcal{P} \times \Sigma, S_0, \delta, F)$, $P \subseteq \mathcal{P}$, $S' \subseteq S$
Ensure: $R[\alpha] = Succ(S', P)(\alpha)$

1: **for all** $\alpha \in S$ **do**
2: $\quad R[\alpha] \leftarrow \emptyset$
3: **end for**
4: $Q \leftarrow \{(\beta, \mathcal{P} \cap P(\alpha, \beta)) \mid \alpha \rightarrow \beta, \alpha \in S'\}$
5: **while** $Q \neq \emptyset$ **do**
6: $\quad$ remove (α, P) from Q
7: $\quad$ **if** $P \not\subseteq R[\alpha]$ **then**
8: $\quad\quad R[\alpha] \leftarrow R[\alpha] \cup P$
9: $\quad\quad Q \leftarrow Q \oplus \{(\beta, P \cap \mathcal{P}(\alpha, \beta)) \mid \alpha \rightarrow \beta, \beta \in S\}$
10: $\quad$ **end if**
11: **end while**

- $Q(\alpha) = \{p \in \mathcal{P} \mid \exists P \subseteq \mathcal{P}.\, p \in P \wedge (\alpha, P) \in Q\}$
- $Q \oplus Q' = \{(\alpha, P) \mid P = Q(\alpha) \cup Q'(\alpha) \wedge P \neq \emptyset\}$

Parameter Synthesis Algorithm

Require: $\mathcal{K} = (\mathcal{P}, S, \mathcal{P} \times \Sigma, S_0, \delta, F)$

Ensure: $p \in P$ iff $\alpha \xrightarrow{p}{}^{*} \gamma \xrightarrow{p}{}^{+} \gamma$ for some $\alpha \in S_0$, $\gamma \in F$

1: $P \leftarrow \emptyset$
2: $R \leftarrow Succ(S_0, \mathcal{P})$
3: **for all** $\gamma \in F$, $R[\gamma] \setminus P \neq \emptyset$ **do**
4: $\quad P \leftarrow P \cup Succ(\gamma, R[\gamma] \setminus P)(\gamma)$
5: **end for**

Complexity Issues

Parameter Synthesis Complexity

- worst case: $O(|S|^2 \cdot |E| \cdot |\mathcal{P}|)$
  $|S|$...states, E...edges, $\mathcal{P}$...colours
- in expected cases $|S|$ and $|\mathcal{P}|$ is reduced (levels of BFS)

Challenges

- number of states exponential w.r.t. number of variables
- size of the parameter space exponential w.r.t. number of unknown parameters
- many computations performed on a single graph

Parallel Implementation

- multi-core data-parallel implementation of colour mapping propagation
- states evenly distributed among threads by a hash-function
- each thread responsible for a unique partition of colour mapping
- threads communicate via a colour mapping update qeue (Q)
  - implemented as a set of lock-free qeues
  - one qeue per thread
  - threads synchronize on BFS levels

E. Coli Ammonium Transport: Model Settings

Settings

- mass action kinetics $\Rightarrow$ multi-affine ODE model
- abstraction – number of discrete concentration levels considered:

$AmtB$	$AmtB : NH_3$	$AmtB : NH_4$	NH_3in	NH_4in
7	9	3	8	26

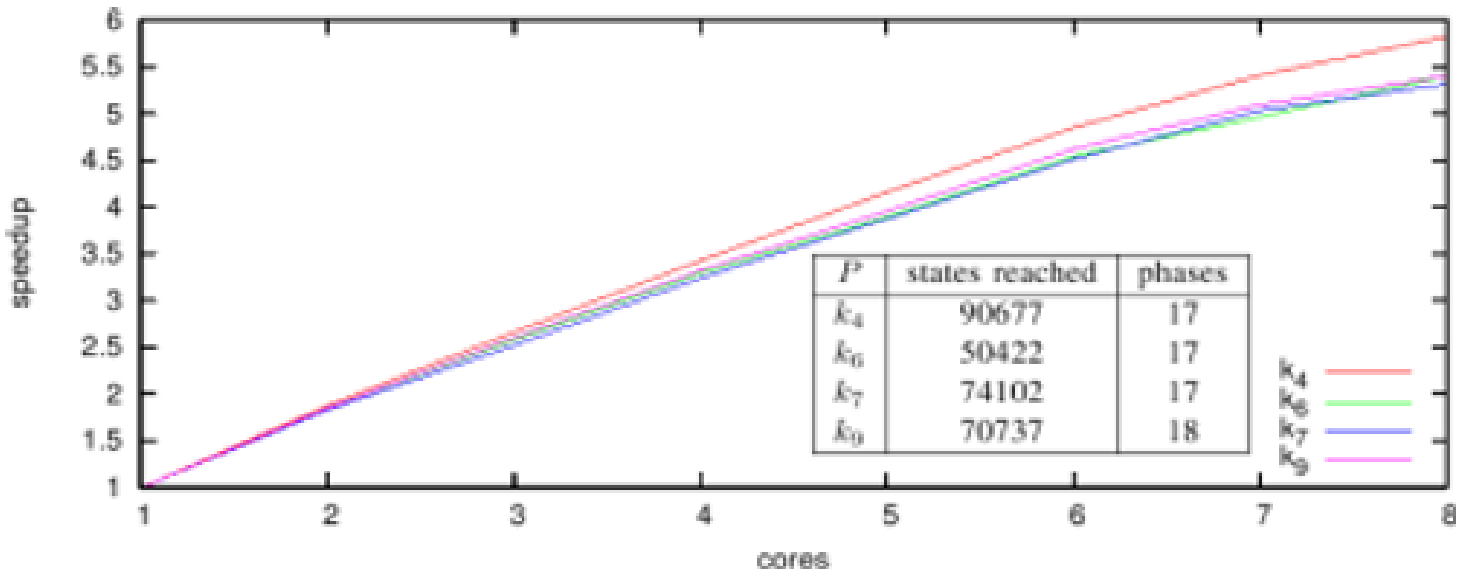
- initial conditions set to impose low external ammonium conditions

Experiments

- find the maximal set of parameter values for the given uknown parameter ensuring the maximal reachable level of internal NH_3 is $1.1 \cdot 10^6$ *mol*
- the employed LTL property: $\mathbf{G}(NH_3in < 1.1 \cdot 10^6)$

E. Coli Ammonium Transport: Experiments

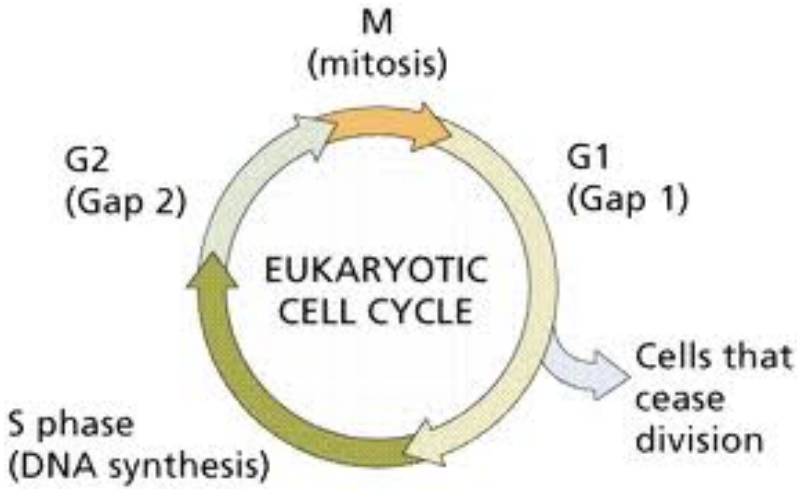
params.	intervals of validity	time
k_4	$(1 \cdot 10^{-12}, 2.7 \cdot 10^6)$	30 s
k_6	$(5.2 \cdot 10^6, 1 \cdot 10^{12})$	22 s
k_7	$(1 \cdot 10^{-12}, 3.3 \cdot 10^6)$	33 s
k_9	$(1 \cdot 10^{-12}, 2.7 \cdot 10^6)$	20 s
$k_{1,6,10}$	see the paper	19 min

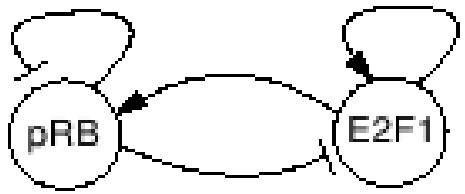


P	states reached	phases
k_4	90677	17
k_6	50422	17
k_7	74102	17
k_9	70737	18

J. Barnat, L. Brim, A. Krejci, D. Safranek, A. Streck, M. Vejnar, and T. Vejpustek. "On Parameter Synthesis by Parallel Model Checking". IEEE/ACM Transactions on Computational Biology and Bioinformatics. May-June 2012;9(3):693-705

Genetic Regulation of G_1/S Transition

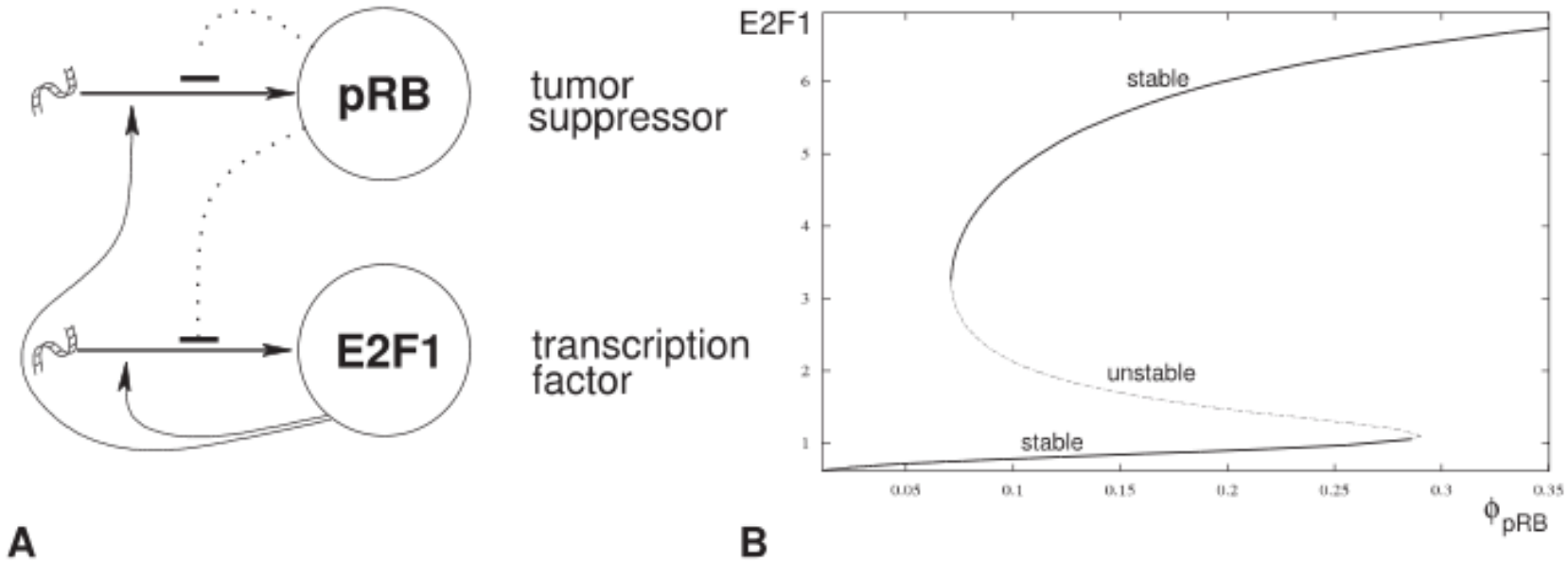




$$\frac{d[pRB]}{dt} = k_1\, \varrho_1(pRB, E2F1) - \gamma_{pRB}\,[pRB]$$
$$\frac{d[E2F1]}{dt} = k_p + k_2\, \varrho_2(pRB, E2F1) - \gamma_{E2F1}\,[E2F1]$$

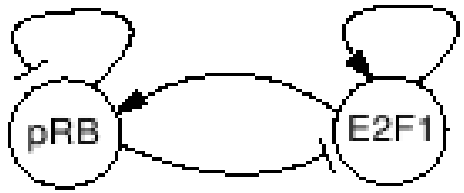
- central module controlling G_1/S transition of mammalian cells

Genetic Regulation of G_1/S Transition



M. Swat, A. Kel, and H. Herzel, "Bifurcation analysis of the regulatory modules of the mammalian G1/S transition," Bioinformatics, vol. 20, no. 10, pp. 1506–1511, 2004.

Genetic Regulation of G_1/S Transition

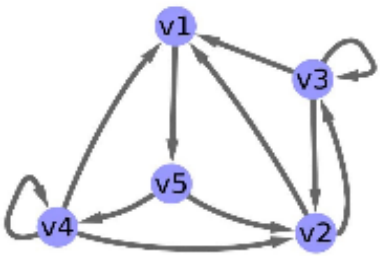


$$\frac{d[pRB]}{dt} = k_1\, \varrho_1(pRB, E2F1) - \gamma_{pRB}\,[pRB]$$
$$\frac{d[E2F1]}{dt} = k_p + k_2\, \varrho_2(pRB, E2F1) - \gamma_{E2F1}\,[E2F1]$$

- bistability w.r.t. setting of γ_{pRB} parameter in the range $[0.01, 1]$
- liveness properties $\mathbf{FG}[E2F1] > 8$ and $\mathbf{FG}[E2F1] < 3$ are employed
- many false-positive runs arise due to time-convergent behaviour introduced by abstraction
- by determining transient rectangles we were able to find acceptable results

Motivation: Learn More about Regulatory Networks

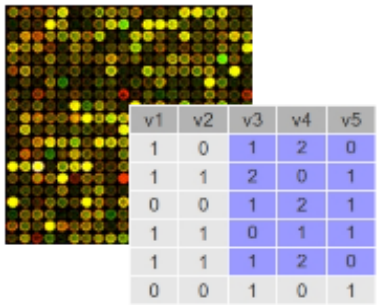
Gene Regulatory Network



Predicted structure

(databases, literature, ...)

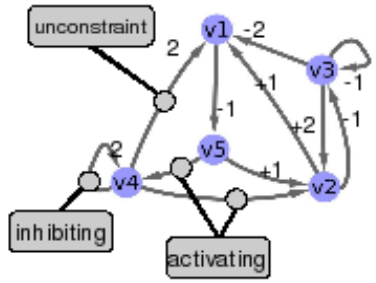
Observations



v1	v2	v3	v4	v5
1	0	1	2	0
1	1	2	0	1
0	0	1	2	1
1	1	0	1	1
1	1	1	2	0
0	0	1	0	1

Discrete time series

Hypothesis

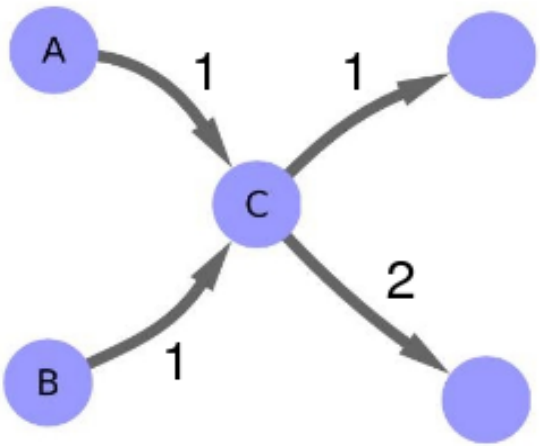


What kind of interactions?

Experimental design

Modeling tools: C. Chaouiya, et al. 2003, GINsim., H. de Jong et al. 2002, GNA. Data processing: I. Shmulevich, et al. 2002. Binary analysis and optimization-based normalization of gene expression data.; E. Dimitrova, et al. 2010. Discretization of time series data.

From Structure to Dynamics

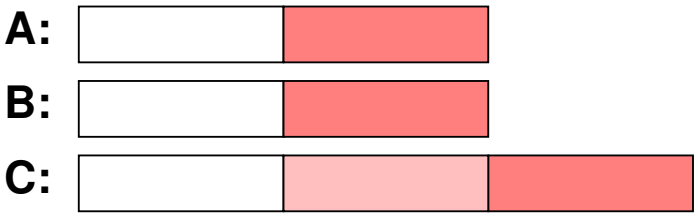


$$\emptyset \rightarrow 0$$

$$\{A\} \rightarrow 2$$

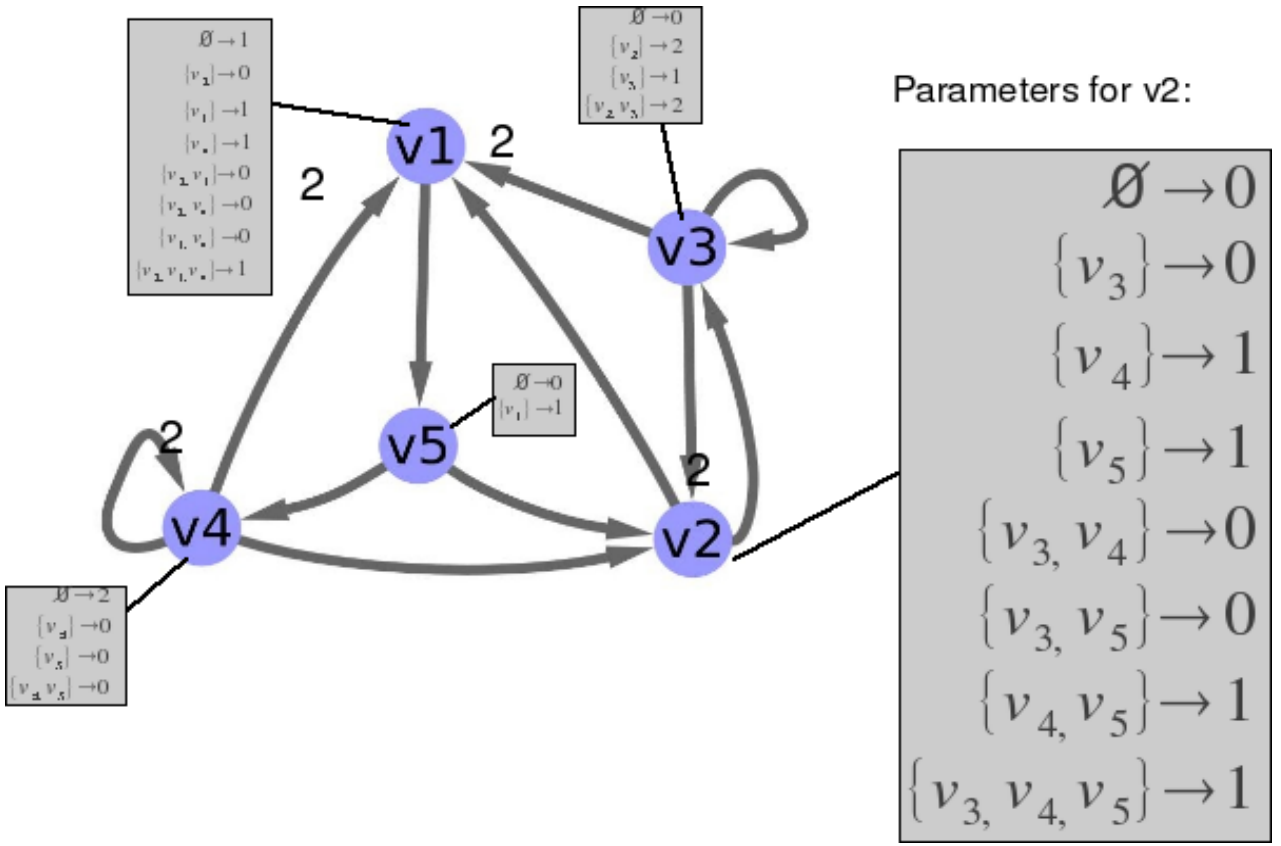
$$\{B\} \rightarrow 2$$

$$\{A, B\} \rightarrow 1$$



R. Thomas and R. d'Ari, CRC Press 1990. Biological feedback.

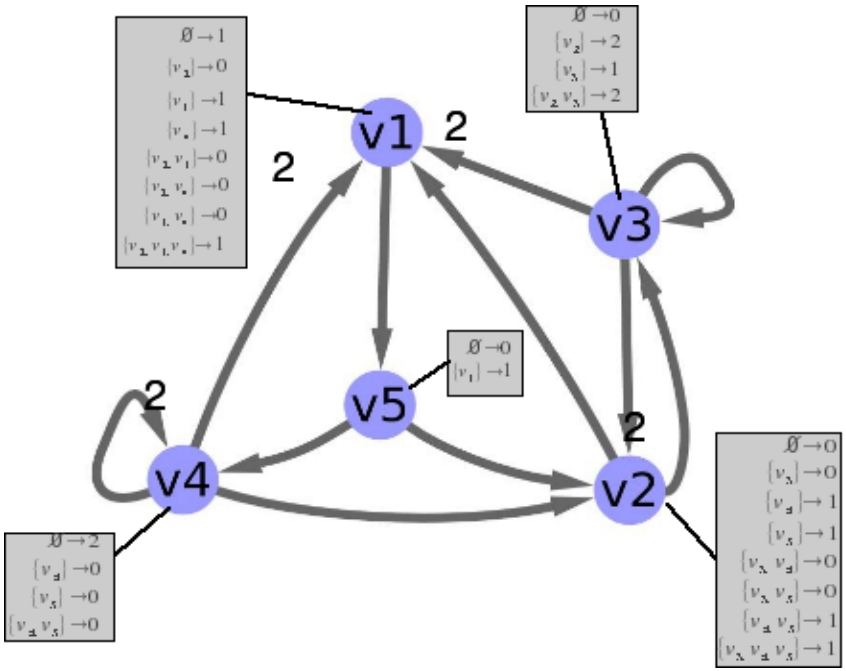
Parameterization of Regulatory Networks

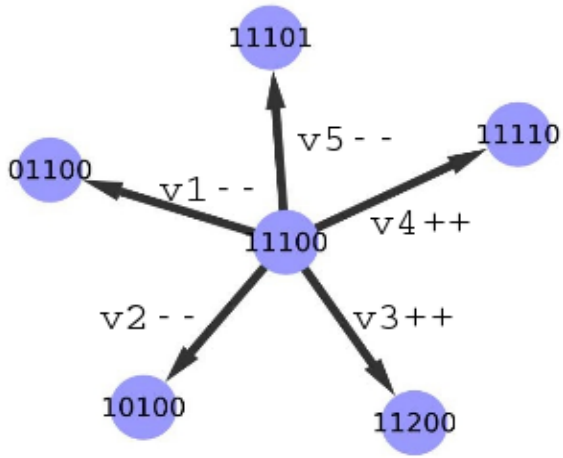


Target values assigned to regulatory contexts for all nodes make a **PARAMETER SET (parameterization)**.

R. Thomas and R. d'Ari, CRC Press 1990. Biological feedback.

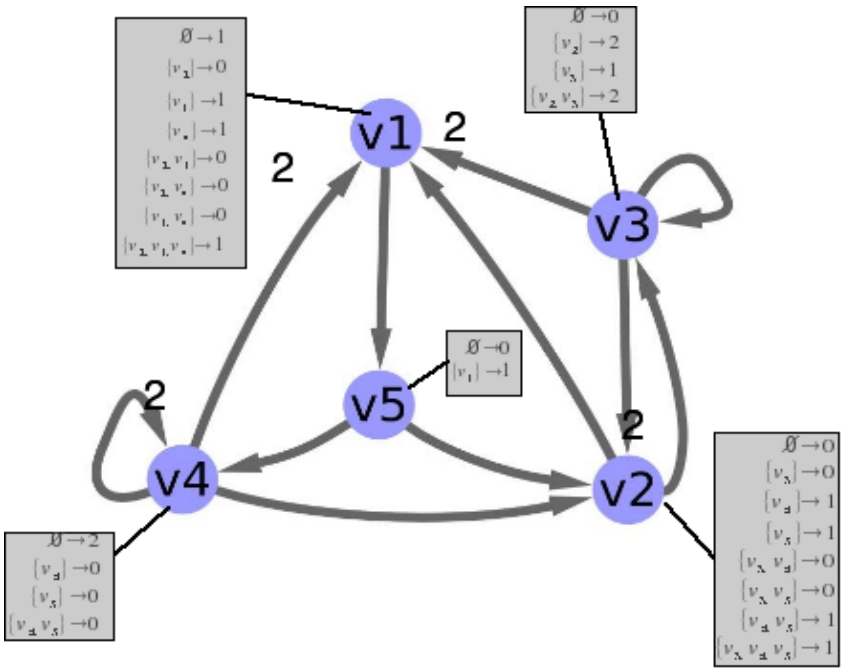
Dynamics as a State Transition Graph

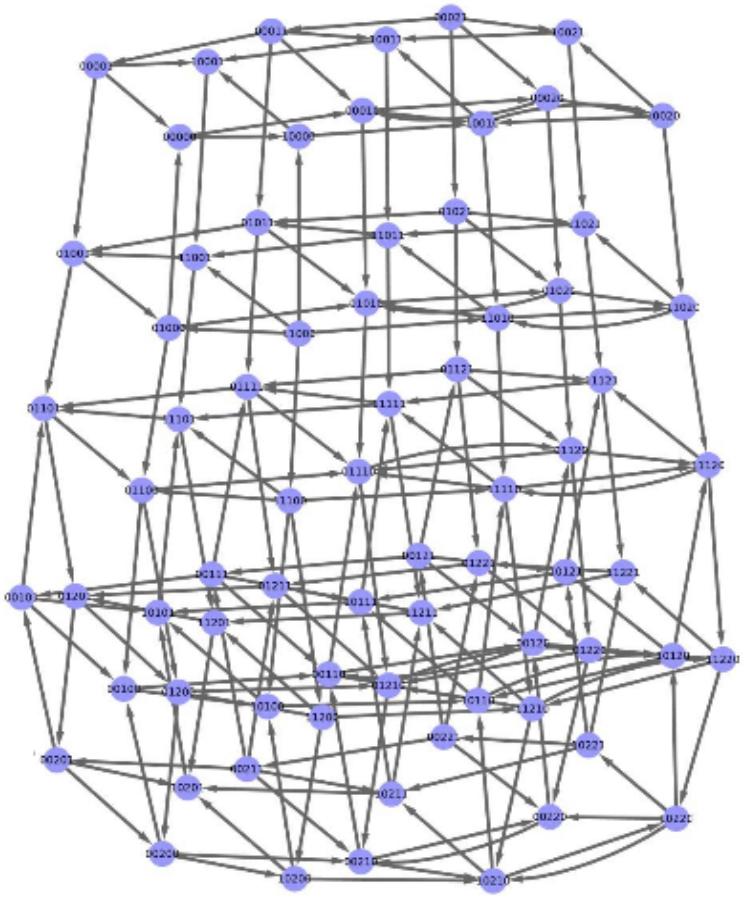




R. Thomas and R. d'Ari, CRC Press 1990. Biological feedback.

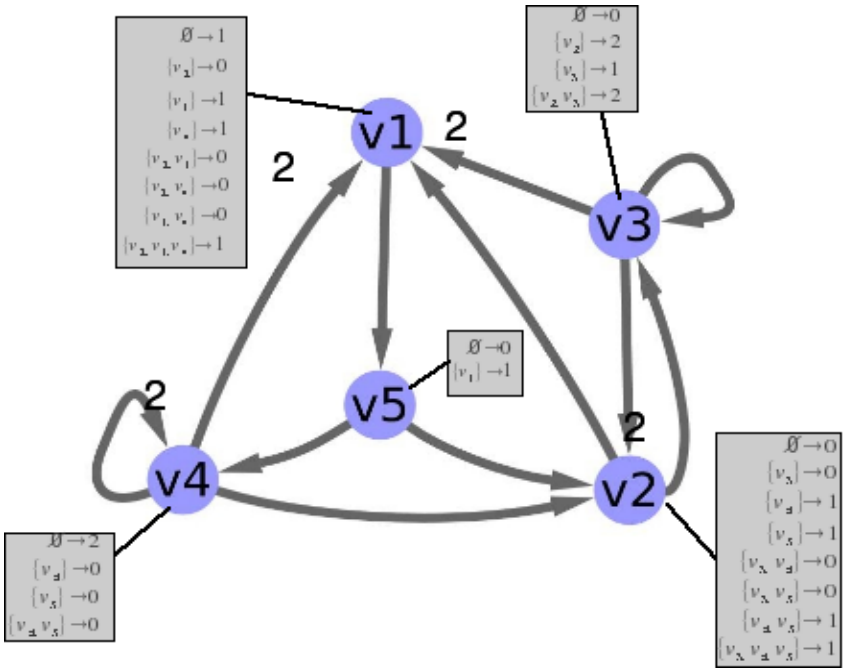
Dynamics as a State Transition Graph

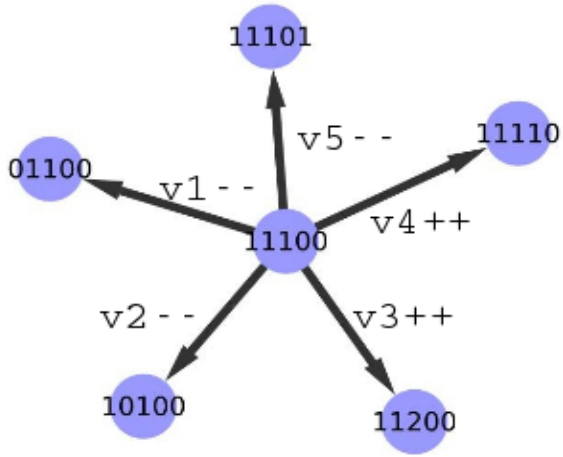




R. Thomas and R. d'Ari, CRC Press 1990. Biological feedback.

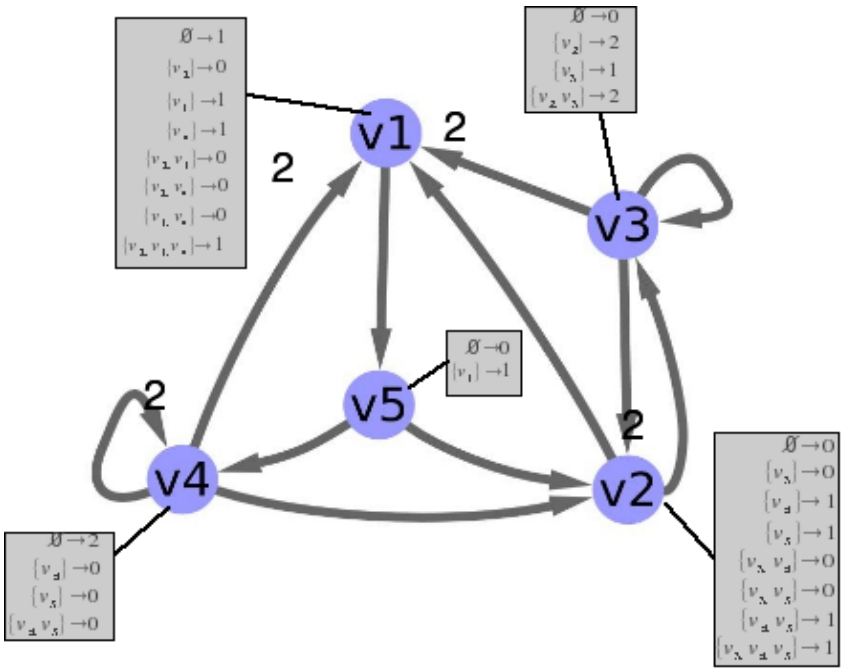
Dynamics as a State Transition Graph

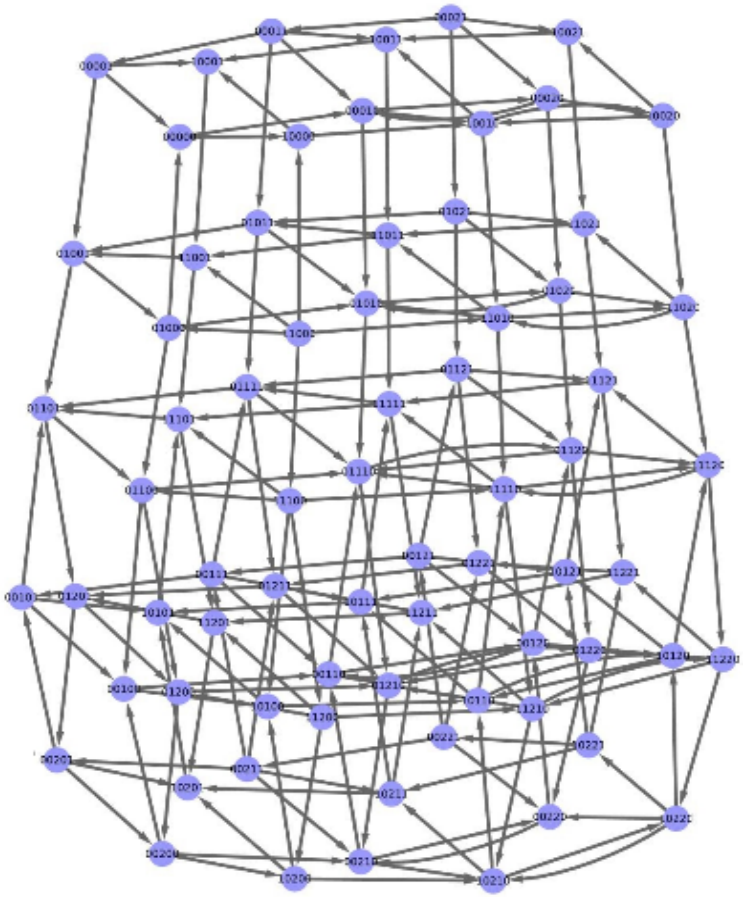




R. Thomas and R. d'Ari, CRC Press 1990. Biological feedback.

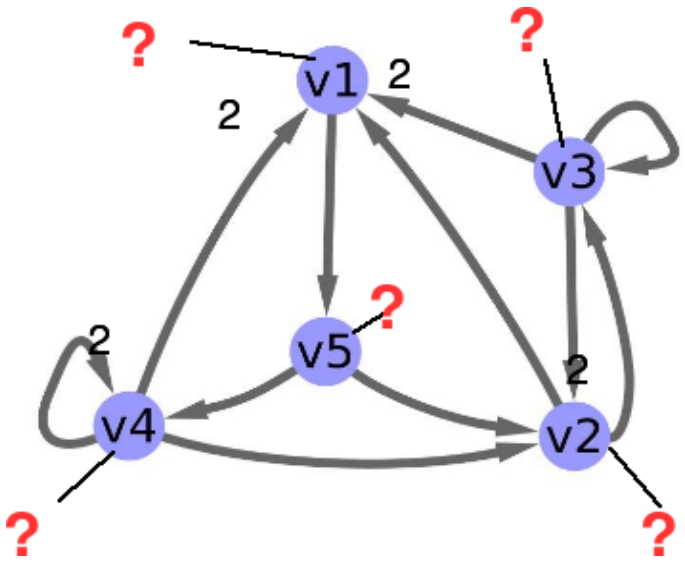
Dynamics as a State Transition Graph





R. Thomas and R. d'Ari, CRC Press 1990. Biological feedback.

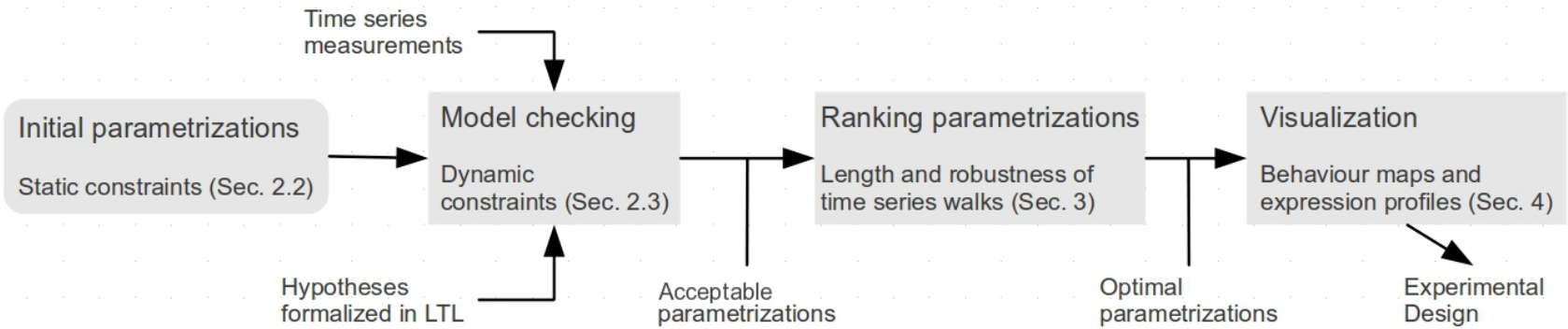
Parameter Identification Problem



Number of possible parameterizations of a single node is **exponential** w.r.t. the node's in-degree.

(more precisely w.r.t. the number of regulatory contexts)

Model Checking-based Methodology



- a prototype tool chain:

  Parsybone – `https://github.com/sybila/Parsybone.git`

  ParameterFilter – `https://github.com/sybila/ParameterFilter.git`

- distributed computation of acceptable parameterizations
- employing witnesses (counterexamples) to rank obtained parameterizations
- visualization of the results (export to Cytoscape)

Time-series Measurement as a Dynamic Constraint

Time-series measurement

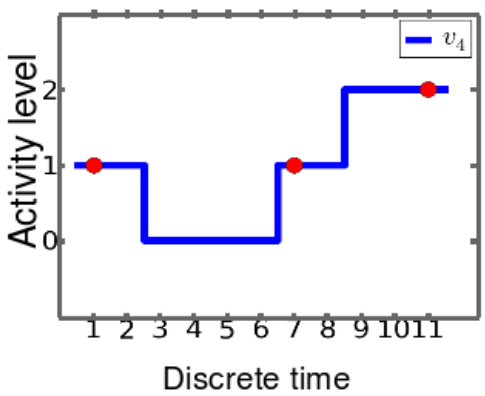
v1	v2	v3	v4	v5
1	1	1	1	1
1	0	1	1	0
1	1	2	2	1

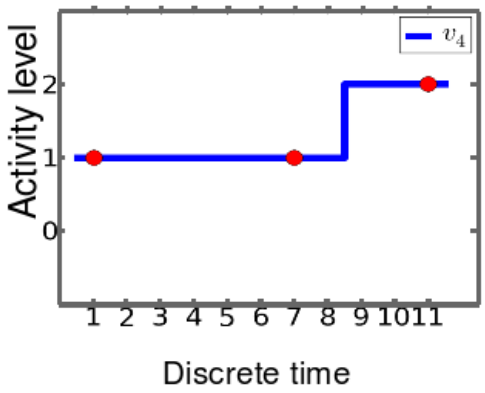
Encoded in LTL:

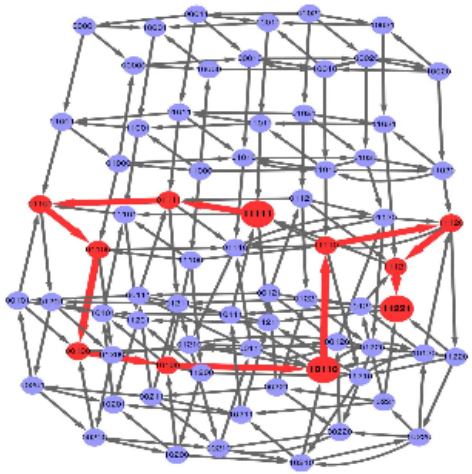
$\sigma(1) = \bigwedge_{i=1}^{5} v_i = 1$
$\sigma(2) = \bigwedge_{i\in\{1,2,4\}} v_i = 1 \wedge \bigwedge_{i\in\{2,5\}} v_i = 0$
$\sigma(3) = \bigwedge_{i\in\{1,2,5\}} v_i = 1 \wedge \bigwedge_{i\in\{3,4\}} v_i = 2$

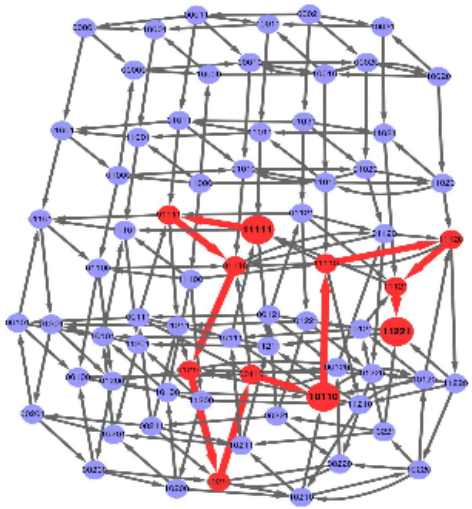
$\varphi = \sigma(1) \wedge \mathbf{F}(\sigma(2) \wedge \mathbf{F}(\sigma(3)))$

Expression of *v4* along red path









Time-series Measurement as a Dynamic Constraint

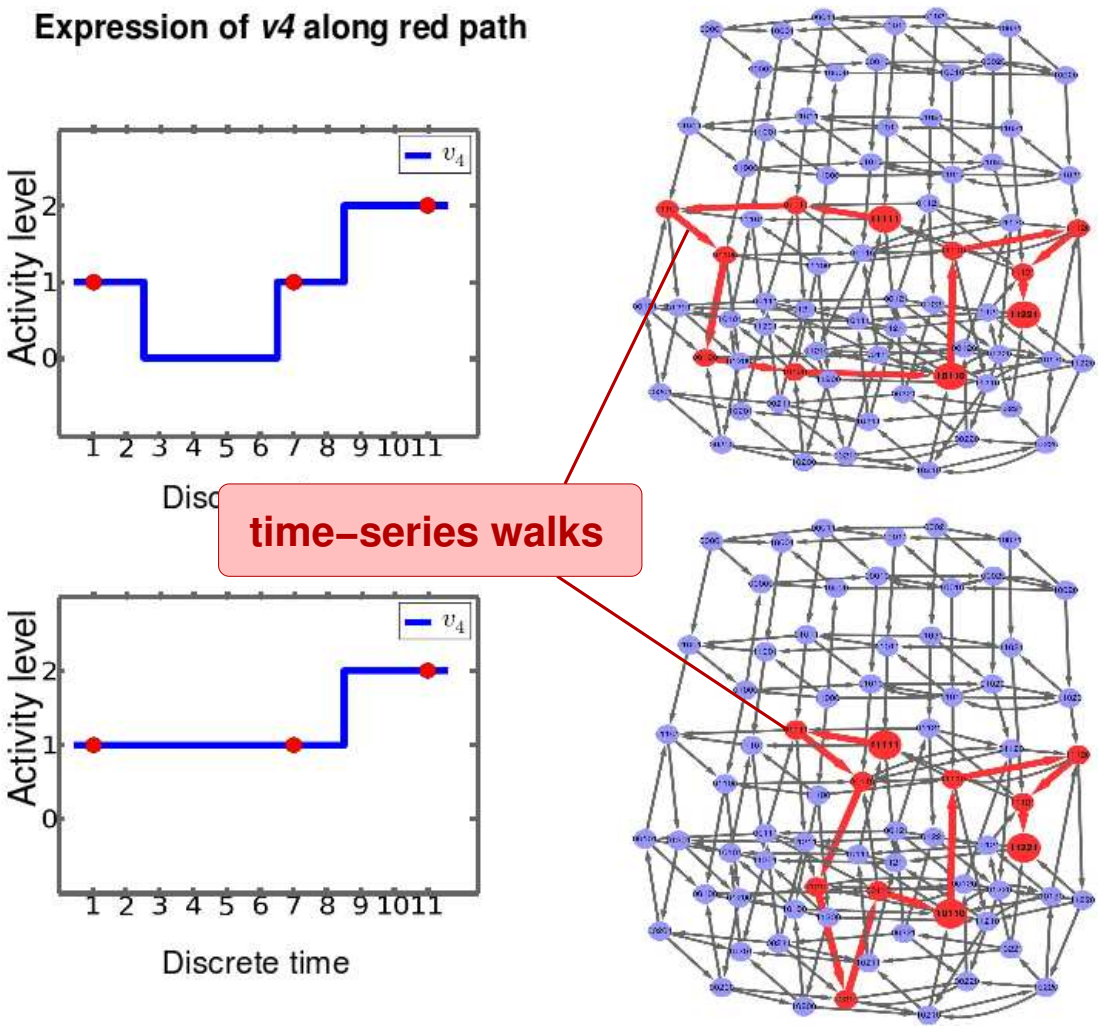
Time-series measurement

v1	v2	v3	v4	v5
1	1	1	1	1
1	0	1	1	0
1	1	2	2	1

Encoded in LTL:

$\sigma(1) = \bigwedge_{i=1}^{5} v_i = 1$
$\sigma(2) = \bigwedge_{i\in\{1,2,4\}} v_i = 1 \wedge \bigwedge_{i\in\{2,5\}} v_i = 0$
$\sigma(3) = \bigwedge_{i\in\{1,2,5\}} v_i = 1 \wedge \bigwedge_{i\in\{3,4\}} v_i = 2$

$\varphi = \sigma(1) \wedge \mathbf{F}(\sigma(2) \wedge \mathbf{F}(\sigma(3)))$



Time-series Measurement as a Dynamic Constraint

Time-series measurement

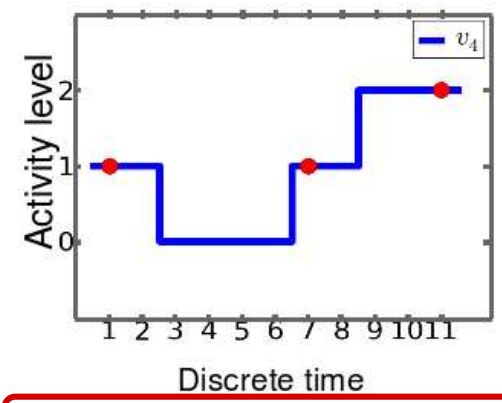
v1	v2	v3	v4	v5
1	1	1	1	1
1	0	1	1	0
1	1	2	2	1

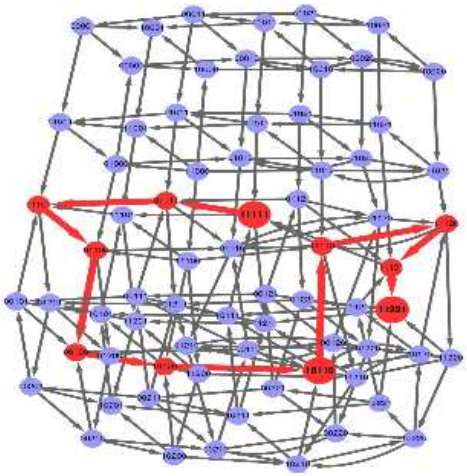
Encoded in LTL:

$\sigma(1) = \bigwedge_{i=1}^{5} v_i = 1$
$\sigma(2) = \bigwedge_{i\in\{1,2,4\}} v_i = 1 \wedge \bigwedge_{i\in\{2,5\}} v_i = 0$
$\sigma(3) = \bigwedge_{i\in\{1,2,5\}} v_i = 1 \wedge \bigwedge_{i\in\{3,4\}} v_i = 2$

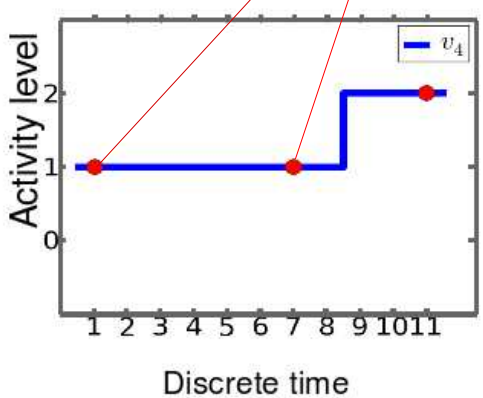
$\varphi = \sigma(1) \wedge (\sigma(1)\mathbf{U}(\sigma(2) \wedge \mathbf{F}(\sigma(3))))$

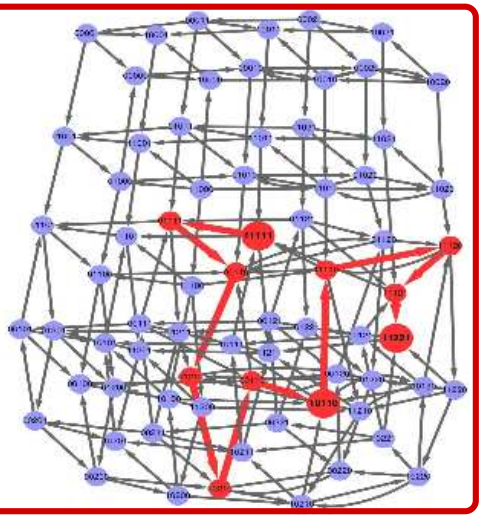
Expression of *v4* along red path





monotonicity between 1st and 2nd measurement





Time-series Measurement as a Dynamic Constraint

Time-series measurement

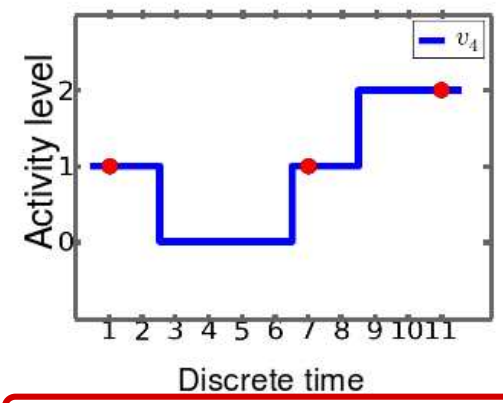
v1	v2	v3	v4	v5
?	1	1	1	**?**
1	0	1	1	0
1	1	2	2	1

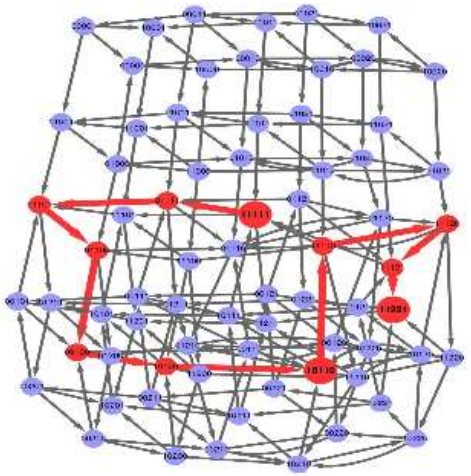
Encoded in LTL:

$\sigma(1) = \bigwedge_{i=2}^{4} v_i = 1$
$\sigma(2) = \bigwedge_{i\in\{1,2,4\}} v_i = 1 \wedge \bigwedge_{i\in\{2,5\}} v_i = 0$
$\sigma(3) = \bigwedge_{i\in\{1,2,5\}} v_i = 1 \wedge \bigwedge_{i\in\{3,4\}} v_i = 2$

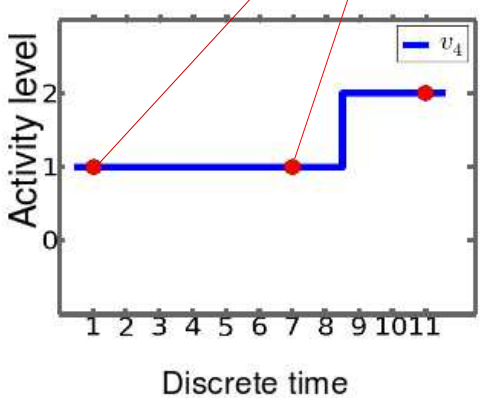
$\varphi = \sigma(1) \wedge (\sigma(1)\mathbf{U}(\sigma(2) \wedge \mathbf{F}(\sigma(3))))$

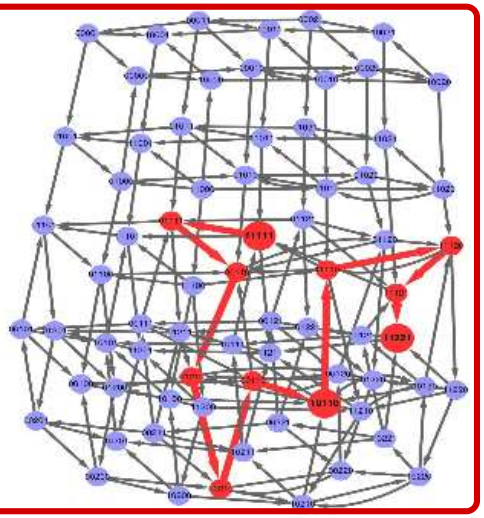
Expression of *v4* along red path





monotonicity between 1st and 2nd measurement

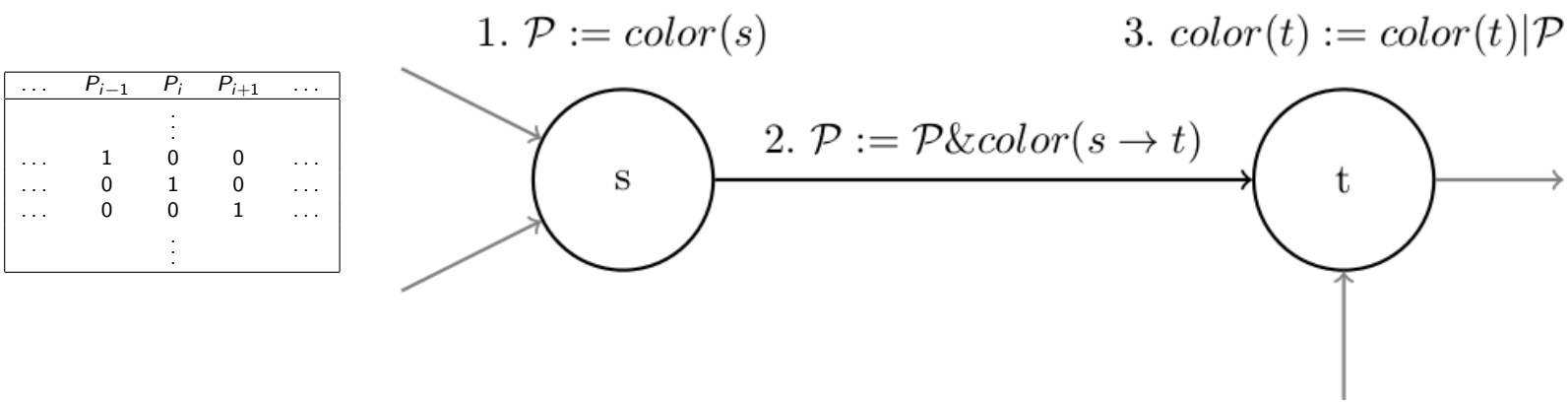




Model Checking on Coloured Graphs

Implementation

- explicit representation of indexed parameter sets (ordered bit vectors)
- parameter space split to exclusive blocks equal to size of integer type
- each block contains "close" parameter sets
- data-parallel distribution: blocks evenly distributed over the cluster



Parameterization Ranking: Length Cost

- theoretically infinitely many time-series walks
- fix a dynamic constraint and focus on compatible **shortest walks**
  - penalize unnecessarily higher energy cost
  - avoid complex model realizations of the constraint
- assign each parameterization its **length cost** – the length of a shortest time-series walk
- consider parameterizations with minimum length cost

Parameterization Ranking: Robustness

- non-deterministic dynamics caused by asynchronicity
- how can we interpret walks with less options to walk off the "optimal path" and miss the expected final state of the time-series?
- the property of the model, but...
  - another classification of parameterizations

- **local robustness**:

  property of a state – $\frac{\text{number of valid successors}}{\text{out degree}}$
- **global robustness**:

  property of a walk – product of local robustness over

  all states of the walk
- **model robustness**:

  property of a parameterization – average of global

  robustness over all time-series walks

Parameterization Ranking: Robustness

- non-deterministic dynamics caused by asynchronicity
- how can we interpret walks with less options to walk off the "optimal path" and miss the expected final state of the time-series?
- the property of the model, but...
  - another classification of parameterizations
- **local robustness – approximated**:

$$\mathrm{Prob}(x) = \frac{1}{\mathrm{out\_degree}(x)}$$

- **global robustness**:

  property of a walk – product of local robustness over

  all states of the walk
- **model robustness**:

  property of a parameterization – average of global

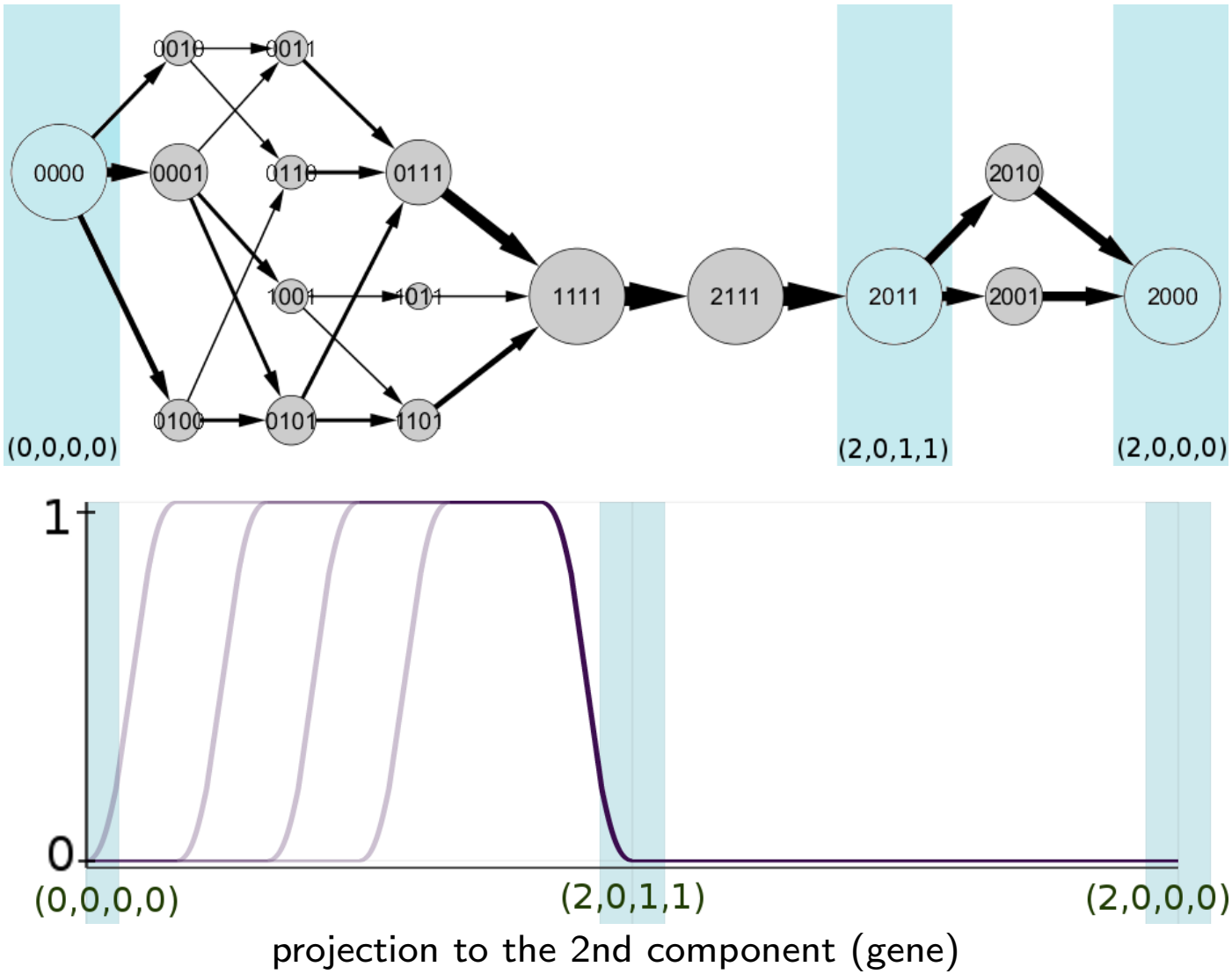
  robustness over all time-series walks

Parameterization Ranking: Overall Procedure

INPUT: regulatory network, initial parameter space, static and dynamic constraints
OUTPUT: subset of the initial parameter space containing optimal parameterizations

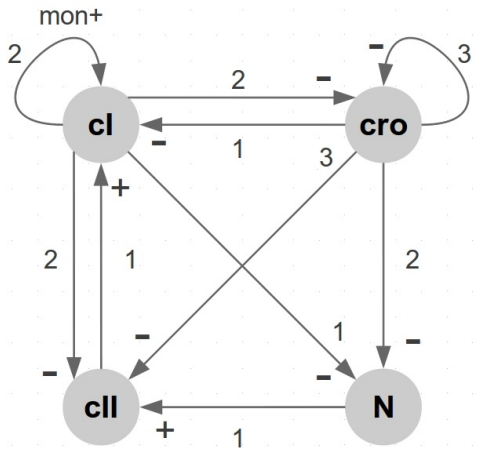
1. Remove parametrizations violating static constraints
2. Compute parameterizations acceptable by dynamic constraints
3. Select parametrizations with minimal length cost
4. Select parametrizations with maximal robustness

Visualising Results
Behaviour Maps and Expression Profiles



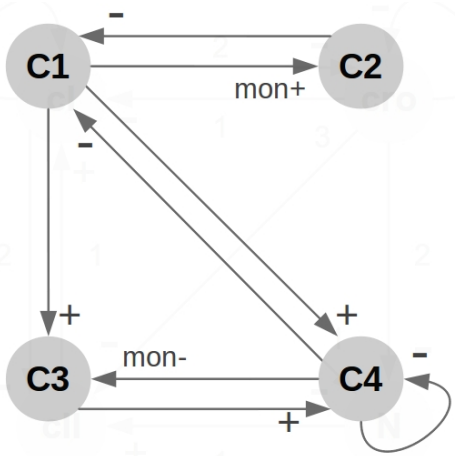
Case Studies

Bacteriophage λ[1]



[Thieffry et al. 1995]

Rat neural system[2]



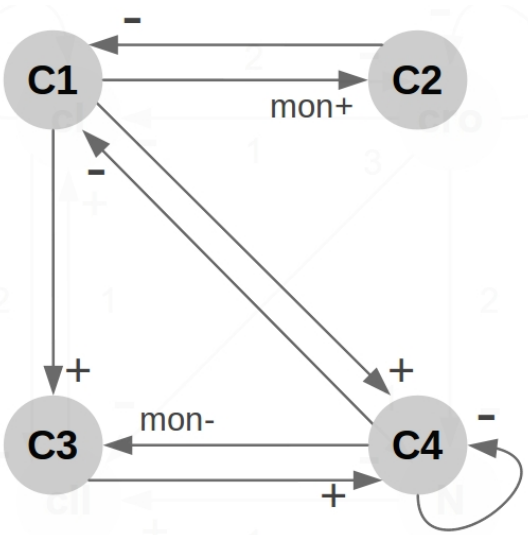
[Wahde et al. 2001]

Init. Parameter Space	$6.9 \cdot 10^9$	$2.6 \cdot 10^5$
Static Constraints	$8.2 \cdot 10^4$	162
Dynamic Constraints	537	108
Length Cost (min)	28 (length 9)	108 (length 5)
Robustness (max)	3 (9.7%)	4 (75%)

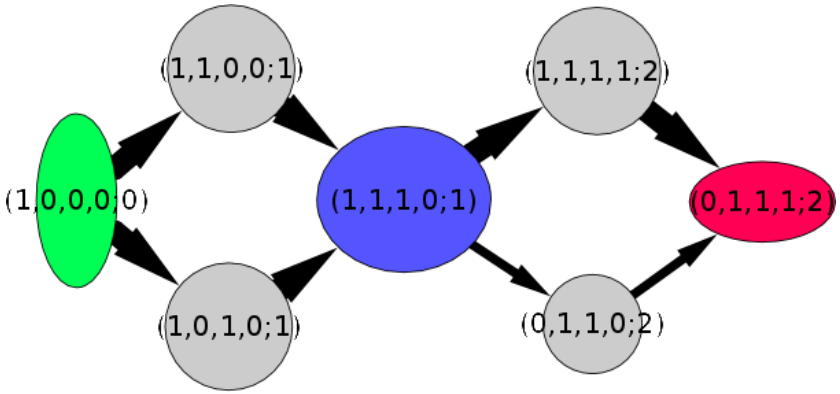
[1] CMSB 2012 Proceedings

[2] FI MU Technical Report

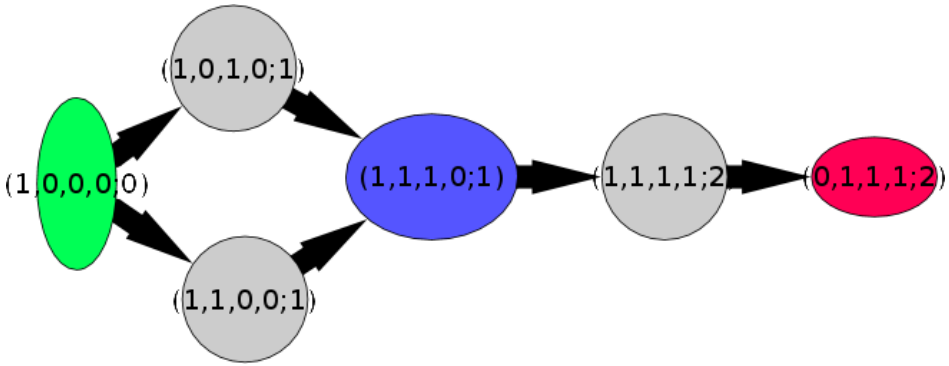
Rat Neural System: Inferring New Hypothesis
[Wan 1998, Wahde 2001]



Shortest paths



Maximally robust paths



Predicted Hypothesis

Genes in cluster 4 express before the cluster 1 expression starts to degrade.

The End

Thank You for your attention.