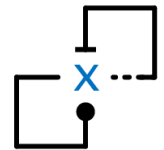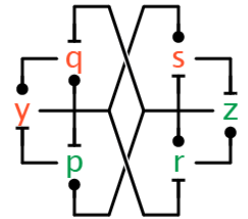# The Cell Cycle Switch Computes Approximate Majority

Luca Cardelli, Microsoft Research & Oxford University

Joint work with Attila Csikász-Nagy, Fondazione Edmund Mach & King's College London

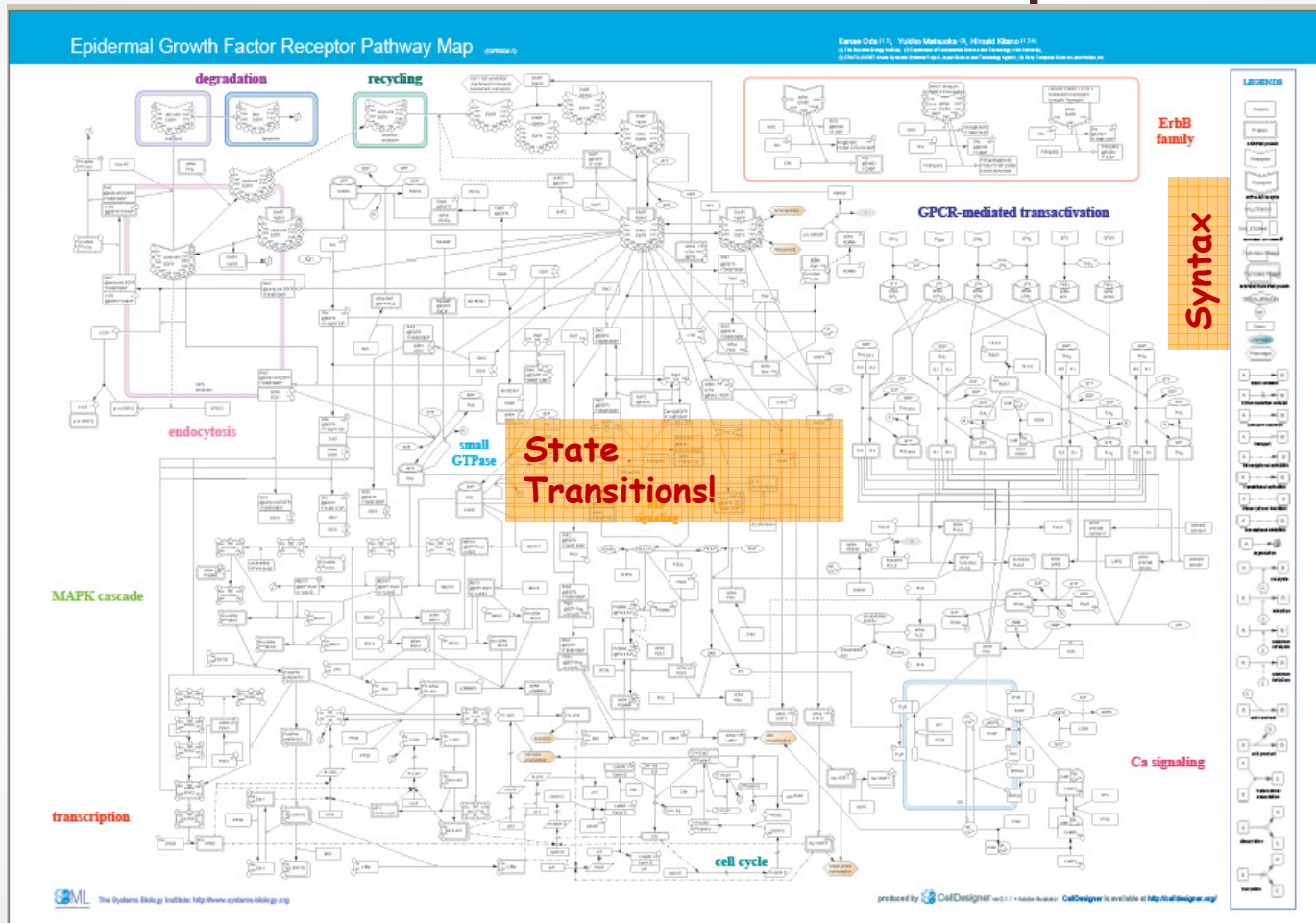St Andrews Distinguished Lectures Series, 2014-11-25

Cells Compute

- ## No survival without computation!
  - Finding food
  - Avoiding predators

- ## How do they compute?
  - *Clearly* doing "information processing"
  - But can we actually catch nature running an (optimal) *algorithm*?

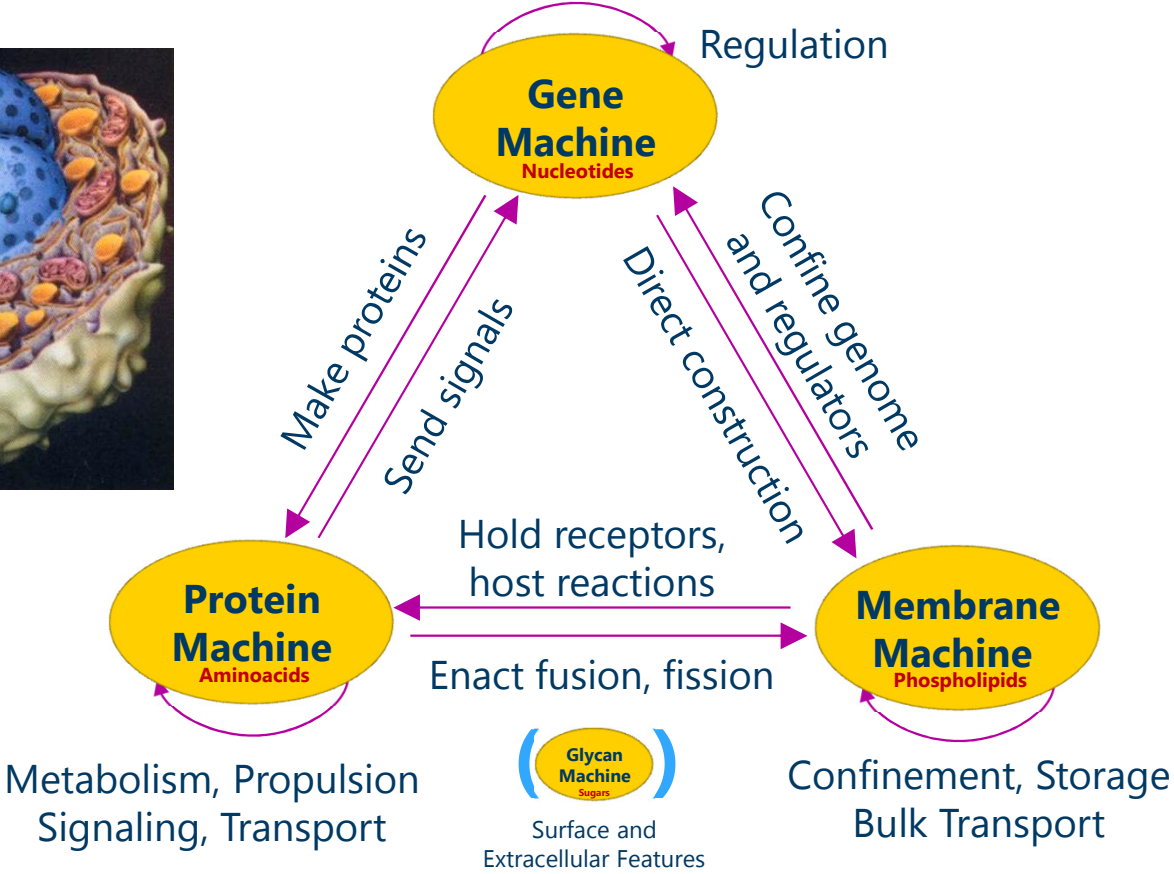- MAPKKK = MAP Kinase Kinase Kinase = *that which operates on that which operates on that which operates on protein.*
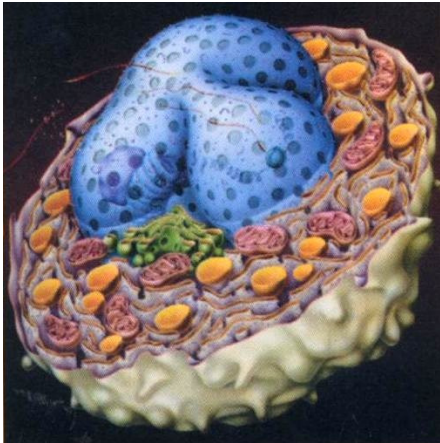

MAPK Family Pathways

Calbiochem®
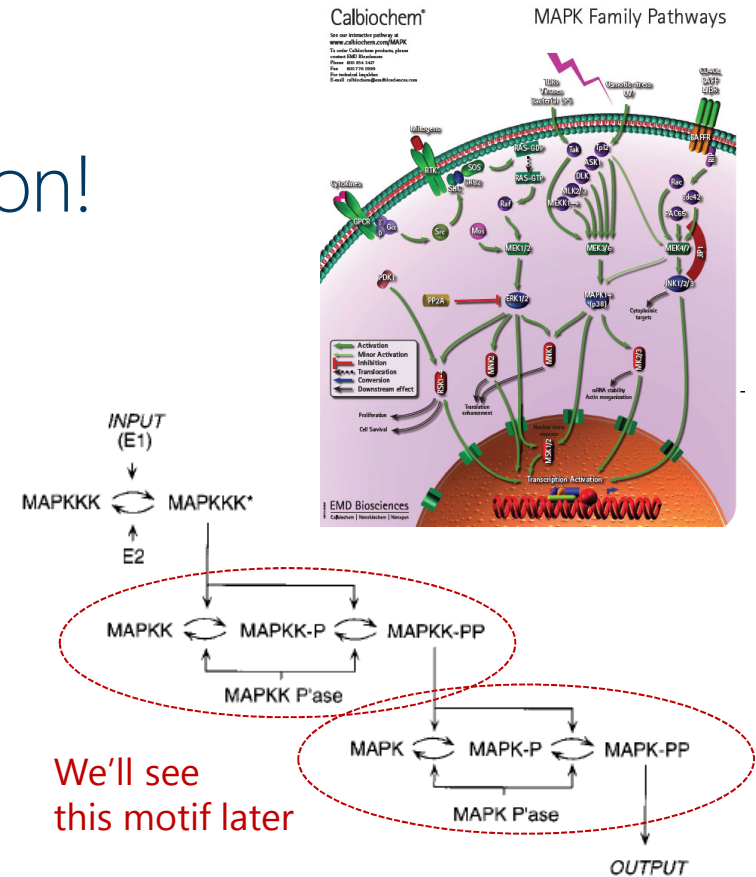
We'll see
this motif later

Ultrasensitivity in the mitogen-activated protein **cascade**, Chi-Ying F. Huang and James E. Ferrell, Jr., 1996, *Proc. Natl. Acad. Sci. USA*, 93, 10078-10083.
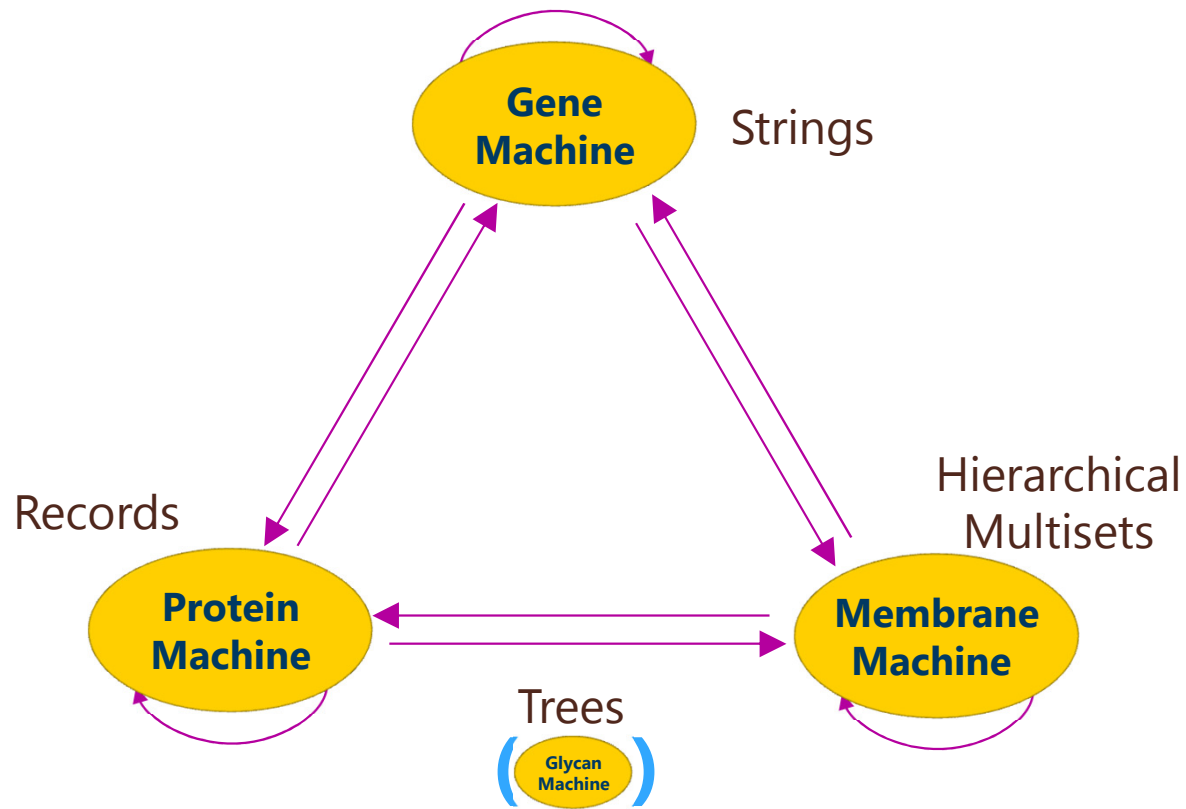
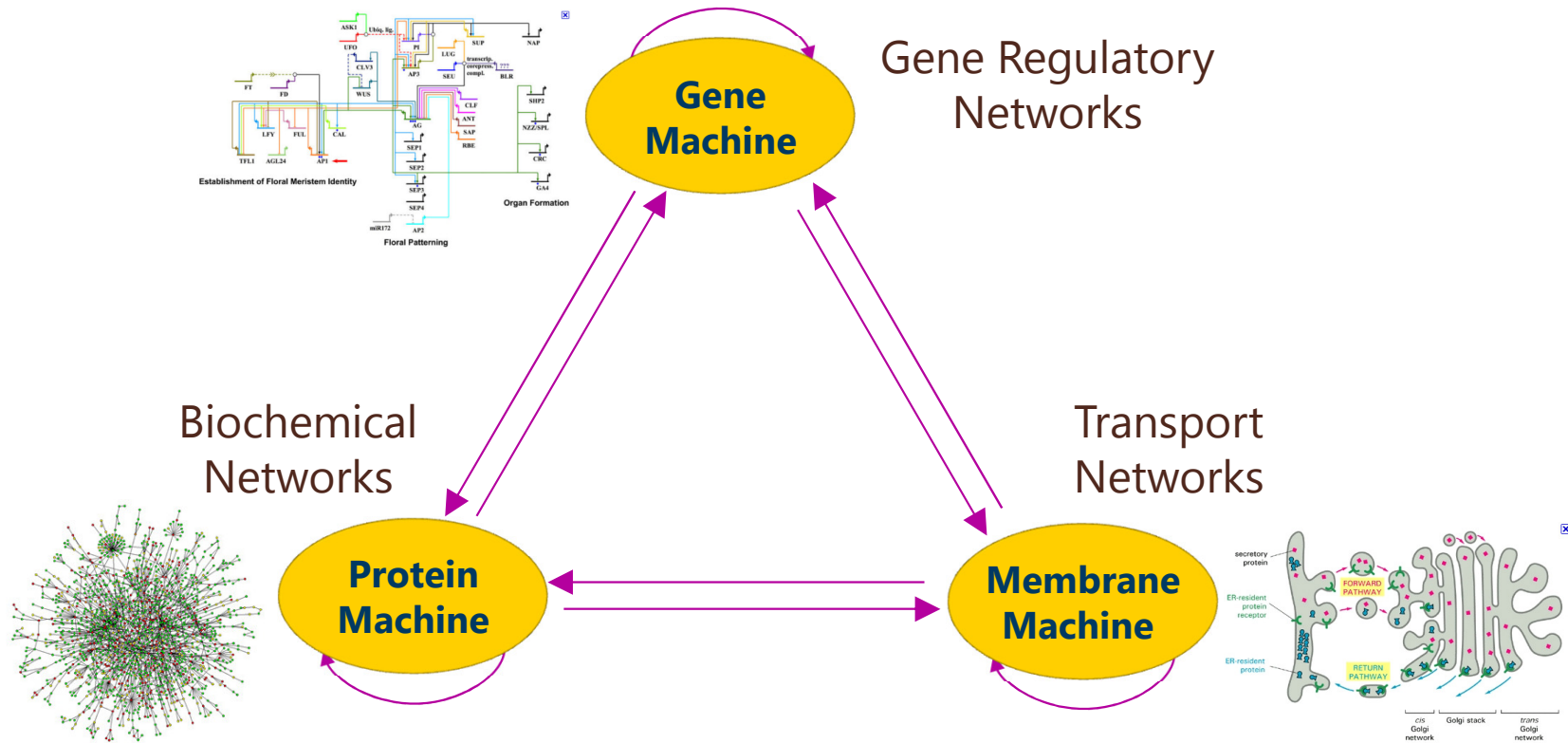# Molecular Interaction Maps (Kohn/Kitano)

Abstract Machines of Biochemistry

Gene Machine — Nucleotides

Regulation

Protein Machine — Aminoacids

Membrane Machine — Phospholipids

Glycan Machine — Sugars

Make proteins

Send signals

Confine genome and regulators

Direct construction

Hold receptors, host reactions

Enact fusion, fission

Metabolism, Propulsion
Signaling, Transport

Surface and
Extracellular Features

Confinement, Storage
Bulk Transport

Bioinformatics View (Data Structures)

Gene Machine — Strings

Protein Machine — Records

Membrane Machine — Hierarchical Multisets

Glycan Machine — Trees

Systems Biology View (Networks)

Gene Regulatory Networks

Gene Machine

Biochemical Networks

Protein Machine

Transport Networks

Membrane Machine

Algorithmic View (Languages)

Molecular Interaction Maps
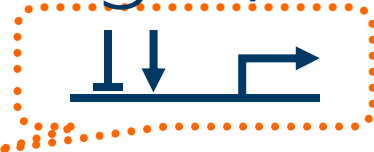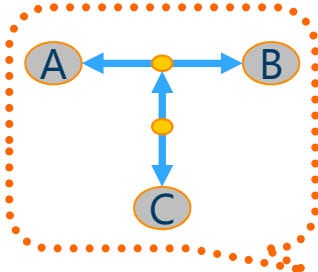
Gene Machine

Gene Networks

Transport Networks

Protein Machine

Membrane Machine
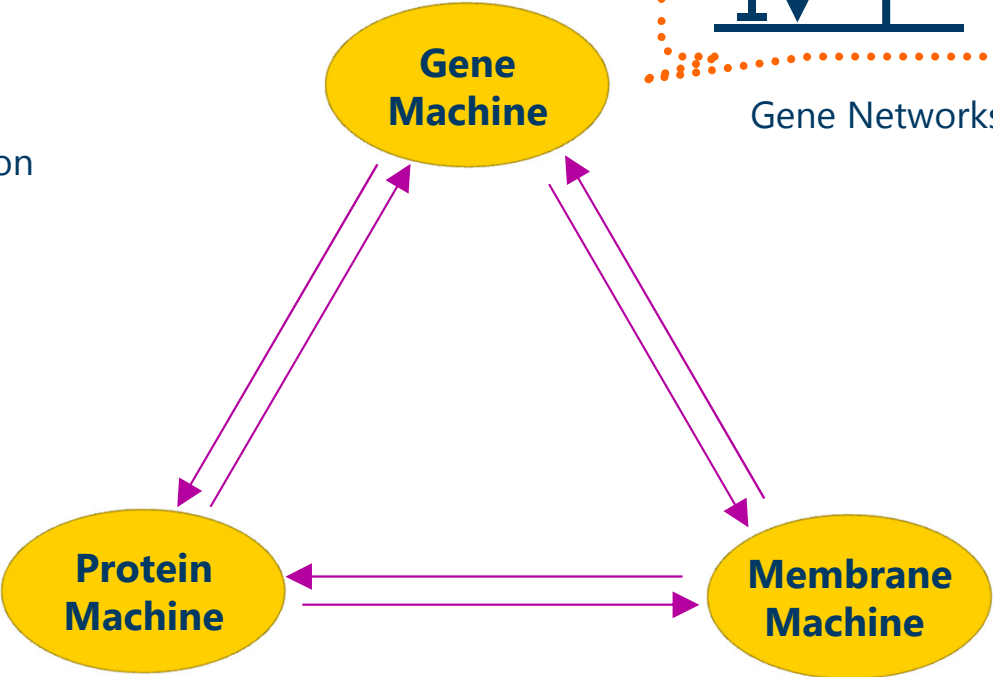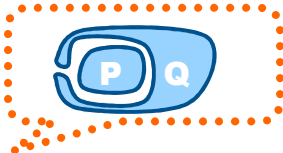
**These 3 machines are Turing powerful!**

Outline

- ## Analyzing biomolecular networks
  - Try do understand the function of a network
  - But also try to understand its *structure*, and what determines it

- ## The Cell-Cycle Switches
  - Some of the best studied molecular networks
  - Important because of their fundamental function (cell division) and the stability of the network across evolution

- ## We ask:
  - What does the cell cycles switch compute?
  - How does it compute it?

The Cell Cycle Switch

- This basic network is universal in Eukaryotes [P. Nurse]
  - The *switching function* and the *basic network* is *the same* from yeast to us.
  - In particular detail, in frog eggs:
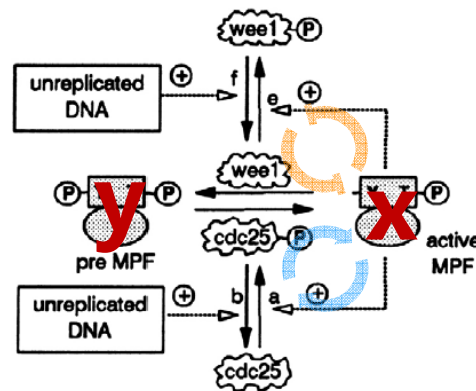
Double positive feedback on x
Double negative feedback on x
No feedback on y
Why ???

- The function is very well-studied. But why this network structure?
- That is, why this peculiar algorithm?

How to Build a Good Switch

- ## What is a "good" switch?
  - We need first a bistable system: one that has two *distinct* and *stable* states.
    I.e., given any initial state the system must settle into one of two states
  - The settling must be fast (not get stuck in the middle for too long)
    and robust (must not spontaneously switch back)
  - Finally, we need to be able to flip the switch by external inputs

- ## "Population protocol" switches
  - Identical agents ('molecules') in a population start in some state, say x or y
  - A pair of agents is chosen randomly at each step,
    they interact ('collide') and change state
  - The whole population must eventually agree on a
   majority value (all-x or all-y) with probability 1

A Bad Algorithm

- ## Direct Competition
  - x catalyzes the transformation of y into x
  - y catalyzes the transformation of x into y
  - when all-x or all-y, it stops

$$y + x \rightarrow x + x$$
$$x + y \rightarrow y + y$$

- ## This system has two end states, but
  - Convergence to an end state is slow (a random walk)
  - Any perturbation of an end state can start a random walk to the other end state (hence not really *bistable*)


11
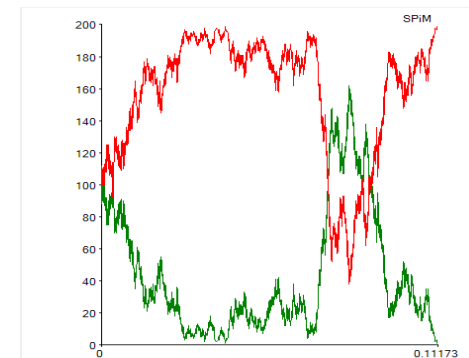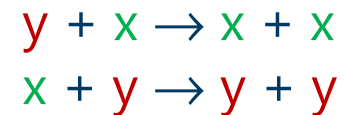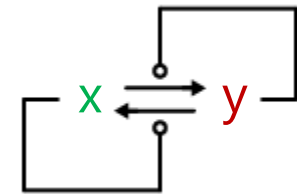
A Very Good Algorithm

catalysis

$x \rightleftarrows b \rightleftarrows y$

- Approximate Majority (AM)
  - Third, undecided, state b
  - Disagreements cause agents to become undecided
  - Undecided agents believe any non-undecided agent

$$x + y \rightarrow y + b$$
$$y + x \rightarrow x + b$$
$$b + x \rightarrow x + x$$
$$b + y \rightarrow y + y$$

- With high probability, for *n* agents
  - The total number of interactions before converging is *O(n log n)*
    $\Rightarrow$ fast (optimal)
  - The final outcome is correct if the initial disparity is *ω(sqrt(n) log n)*
    $\Rightarrow$ solution states are robust to perturbations

- Logarithmic time bound in parallel time
  - *Parallel time* is the number of steps divided by the number of agents
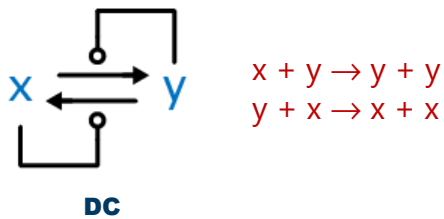 - In parallel time the algorithm converges with high probability in *O(log n)*
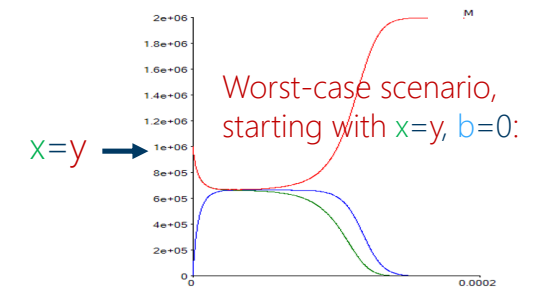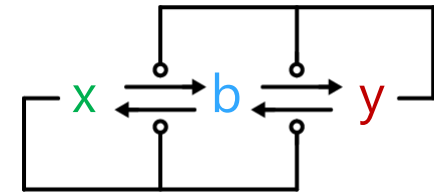
x=y →

Worst-case scenario, starting with x=y, b=0:

Dana Angluin · James Aspnes · David Eisenstat

A Simple Population Protocol for Fast Robust Approximate Majority
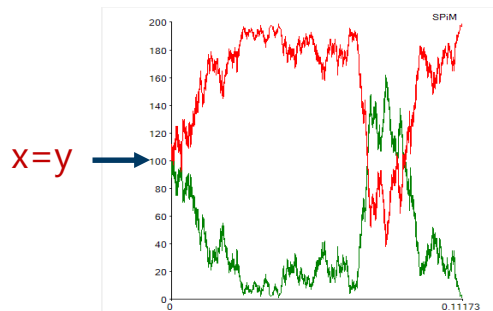
12

Consensus Algorithms

Direct Competition
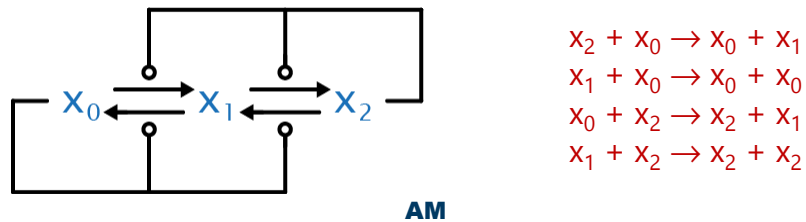
$$x + y \rightarrow y + y$$
$$y + x \rightarrow x + x$$

DC

Approximate Majority

$$x_2 + x_0 \rightarrow x_0 + x_1$$
$$x_1 + x_0 \rightarrow x_0 + x_0$$
$$x_0 + x_2 \rightarrow x_2 + x_1$$
$$x_1 + x_2 \rightarrow x_2 + x_2$$

AM

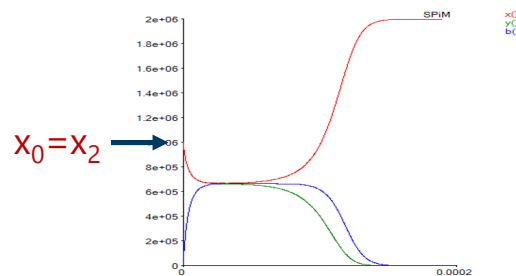### Bad: O(n)

$x=y$

Dana Angluin · James Aspnes · David Eisenstat

A Simple Population Protocol for Fast Robust Approximate Majority
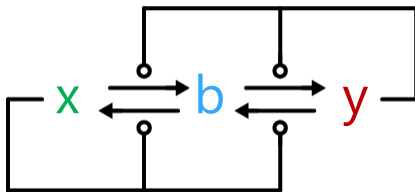
Good: O(log n)

$x_0=x_2$

Worse-case scenario example, starting with $x_0=x_2$, $x_1=0$:

13

A Biological Implementation

Approximate Majority (AM)

Bistable
Even when x=y (stochastically)

Fast
$O(log\ n)$ convergence time

Robust to perturbation
above a threshold, initial majority wins *whp*

Dana Angluin · James Aspnes · David Eisenstat
A Simple Population Protocol for Fast Robust
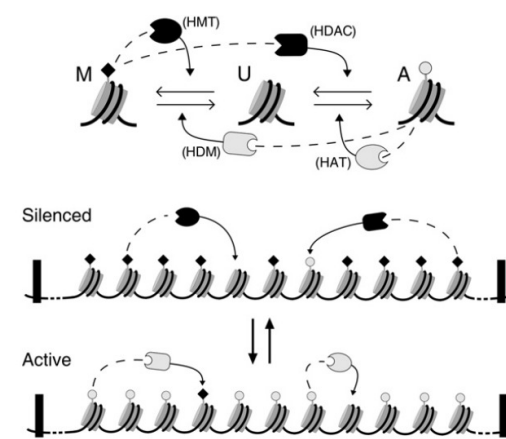Approximate Majority

Epigenetic Switch

Figure 1. Basic Ingredients of the Model

Back to Biology

- The AM algorithm has ideal properties for settling a population into one of two states

- Seems like this would be useful in Biology
  - Can we find biological implementations of this algorithm?
  - Could it be related to the cell cycle switch?

How to model "Influence"
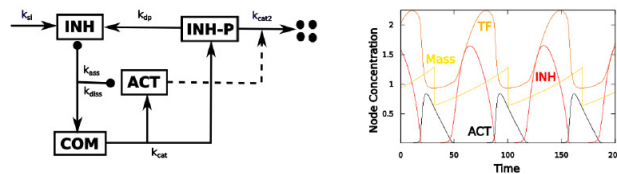
"True" molecular interactions.

Figure 3: a) Schematic diagram of a simplified SIMM model [17]. The activa-

Evolving a Primitive Eukaryotic Cell Cycle Model
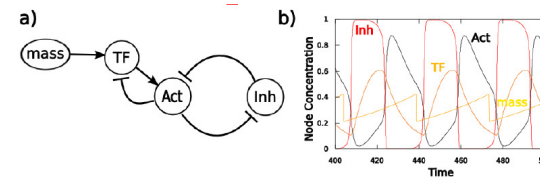
Malte Lücken, Jotun Hein, Bela Novak

"Equivalent" influence interactions.

Figure 4: a) Schematic diagram of a primitive cell cycle in the reinitz framework.

Instead of modeling basic interactions, such as binding, synthesis, and degradation of molecular components, this framework models interactions simply as activation or inhibition. This approach also reduces the number of nodes necessary in the network, as e.g. the inhibitor binding tightly to the activator to form a complex, which produces phosphorylated inhibitor to be degraded under catalysis by the activator, is now simply a double negative feedback loop shown in Figure 1. This type of interaction is the basis of both aforementioned molecular model, therefore they can both be summarized in a single Reinitz model.

The Reinitz Model of Influence

- Based on early connectionist (neural network) modeling
  - Each activation/inhibition interaction is modeled as a flexible sigmoid function with 4+ parameters per node
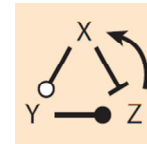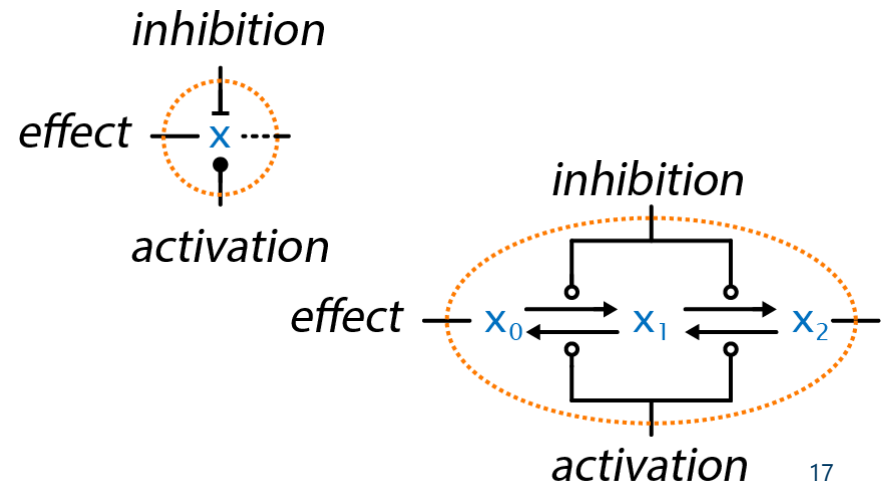

Functional Motifs in Biochemical Reaction Networks

John J. Tyson[1] and Béla Novák[2]

$$\frac{dX_i}{dt} = \gamma_i \frac{[A_i(1 - X_i) - B_i X_i]}{A_i + B_i}, \quad i = 1, \ldots, N,$$

$$A_i = \exp\left\{\sigma_i\left(\alpha_{i0} + \sum_{j=1}^{N}\alpha_{ij}X_j\right)\right\}, \quad B_i = \exp\left\{\sigma_i\left(\beta_{i0} + \sum_{j=1}^{N}\beta_{ij}X_j\right)\right\}, \tag{4}$$

- We prefer to stick to mass action kinetics
  - It will later become clear why

- We model activation/inhibition nodes by a mass action motif:
  - Using 4 rate parameters per node
  - Akin to multisite modification
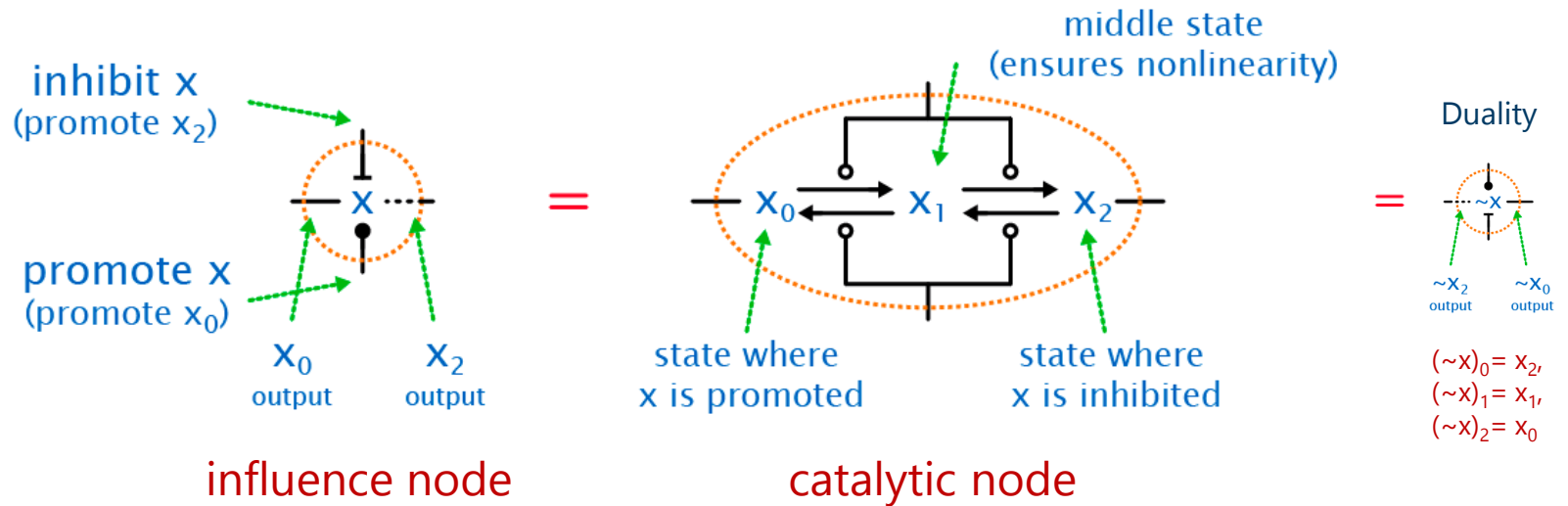
Influence Network Notation

- Catalytic reaction

$$x \xrightarrow{\;z\;} y \;=\; x \xrightarrow{\;z\;\;z\;} y$$

z is the catalyst

$$x + z \rightarrow z + y$$

- Triplet motif

inhibit x
(promote x_2)

promote x
(promote x_0)

x_0 output x_2 output

=

middle state
(ensures nonlinearity)

$$x_0 \rightleftarrows x_1 \rightleftarrows x_2$$

state where
x is promoted

state where
x is inhibited

=

Duality

$\sim x$

$\sim x_2$ output $\sim x_0$ output

$(\sim x)_0 = x_2$,
$(\sim x)_1 = x_1$,
$(\sim x)_2 = x_0$

influence node

catalytic node

18

Influence Network Duality

- Let ~x be the species such that

$$(\sim x)_0 = x_2, \quad (\sim x)_1 = x_1, \quad (\sim x)_2 = x_0$$

so that promoting x is the same as inhibiting ~x etc. Then:

The Triplet Model of Influence

inhibition

effect — x_0 ⇌ x_1 ⇌ x_2 — *(other effect)*

with k_{01}, k_{10}, k_{12}, k_{21}

activation

- Solving this mass action model at steady state with $tot = x_0 + x_1 + x_2$, obtain x_0 as a function of a and i :

$$x_0 = \frac{k_{10}k_{21}tot\,a^2}{k_{10}k_{21}a^2 + k_{01}k_{21}ai + k_{01}k_{12}i^2}$$

- Assuming $i = tot - a$ (inhibition decreases as activation increases) obtain x_0 as a function of $a \in [0..tot]$ (max stimulus = max response)

$$x_0 = \frac{k_{10}k_{21}tot\,a^2}{(k_{10}k_{21} - k_{01}k_{21} + k_{01}k_{12})a^2 + (k_{01}k_{21} - 2k_{01}k_{12})tot\,a + k_{01}k_{12}\,tot^2}$$

- By regulating the rates of flow through x_1 within 2 orders of magnitude we can obtain a range of linear, hyperbolic and sigmoid responses in the range [0..1] to linear activation $a \in [0..1]$.
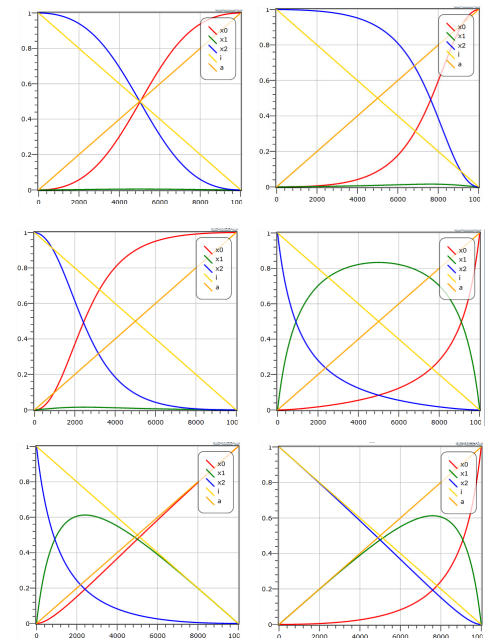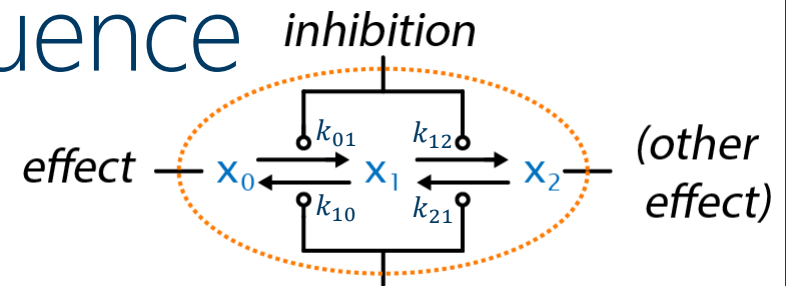
steady state transitions
from inhibited to activated
with $tot = 1$ and $a \in [0..1]$

x0
x1
x2
i
a

Influence Nodes

inhibition

high — x — low

=

x is high — $x_0 \rightleftarrows x_1 \rightleftarrows x_2$ — x is low

inhibit x

activate x

activation
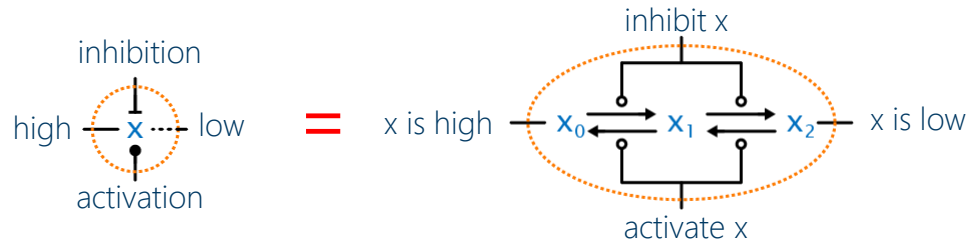
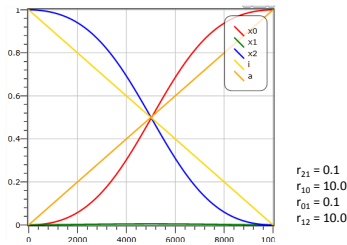Usually modeled by sigmoid (e.g. Hill or Reinitz) functions

Functional Motifs in Biochemical Reaction Networks

John J. Tyson[1] and Béla Novák[2]

$$\frac{dX_i}{dt} = \gamma_i \frac{[A_i(1 - X_i) - B_i X_i]}{A_i + B_i}, \quad i = 1, \dots, N,$$

$$A_i = \exp\left\{\sigma_i\left(\alpha_{i0} + \sum_{j=1}^{N} \alpha_{ij} X_j\right)\right\}, \quad B_i = \exp\left\{\sigma_i\left(\beta_{i0} + \sum_{j=1}^{N} \beta_{ij} X_j\right)\right\}, \quad (4)$$

We model them by 4 mass action reactions over 3 species x_0, x_1, x_2

They actually implement a Hill function of coefficient 2:

$r_{21} = 0.1$
$r_{10} = 10.0$
$r_{01} = 0.1$
$r_{12} = 10.0$

Approximate Majority

$x_0 \rightleftarrows x_1 \rightleftarrows x_2$

=

X

AM

21
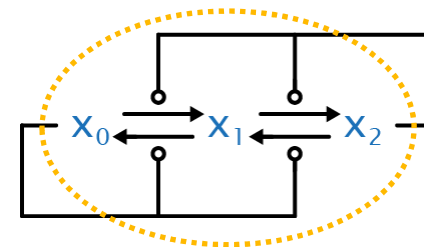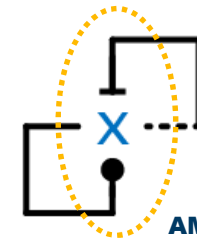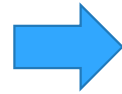
Refining the AM algorithm

- Subject to biochemical constraints

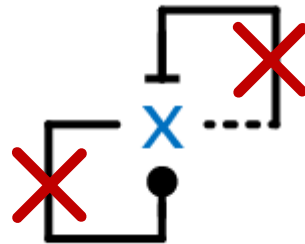# Step 1: the AM Network

(x_2 promotes x_2)

(x_0 promotes x_0)
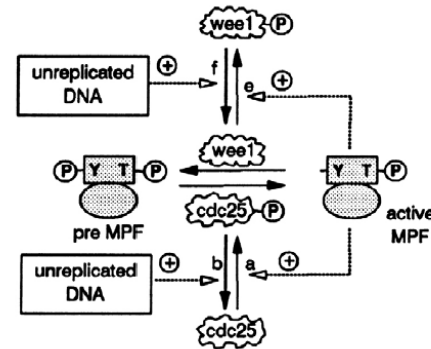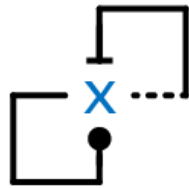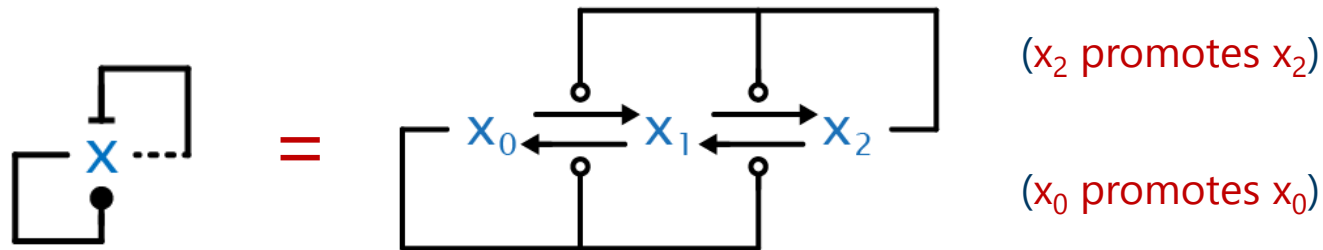
- ... not biochemically plausible

Natural Constraint #1

- Direct autocatalysis is not commonly seen in nature

$$x_1 + x_0 \rightarrow x_0 + x_0$$
$$x_1 + x_2 \rightarrow x_2 + x_2$$

Step 2: remove auto-catalysis

- Replace autocatalysis
  - By *mutual* (simple) catalysis, introducing intermediate species z and r
  - z and r need to 'relax back' when they are not being promoted:
    s and t provide the back pressure for such relaxation

(x_2 promotes z_2, promotes x_2)

(x_0 promotes r_0, promotes x_0)
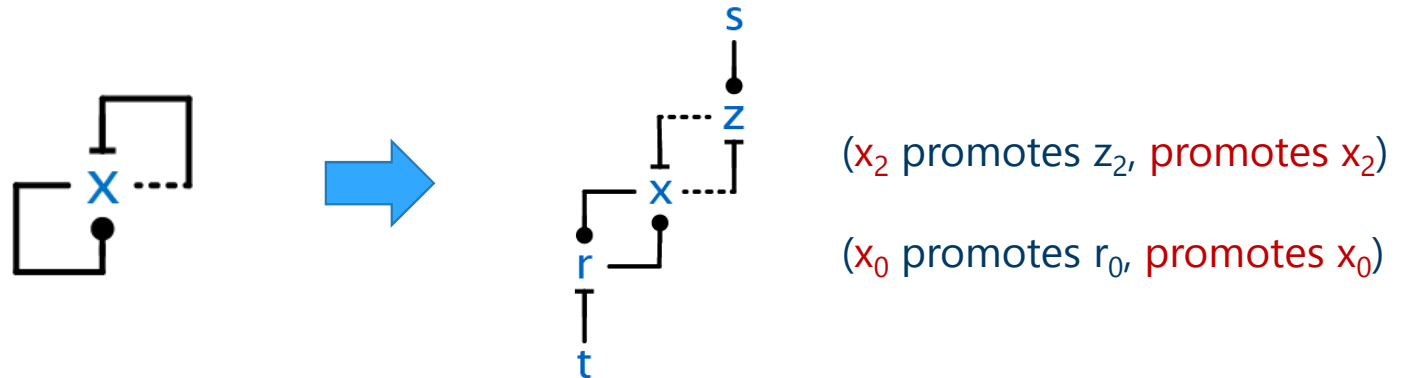
- ... still not biochemically plausible.

25

Natural Constraint #2

- x_0 and x_2 (usually two states of the same molecule) are both active catalysts in that network
- That is not commonly seen in nature

$$\perp \atop \text{—}x\,\text{---} \atop \text{|}$$ vs. $$\perp \atop \text{—}x \atop \text{|}$$ or $$\perp \atop x\,\text{---} \atop \text{|}$$
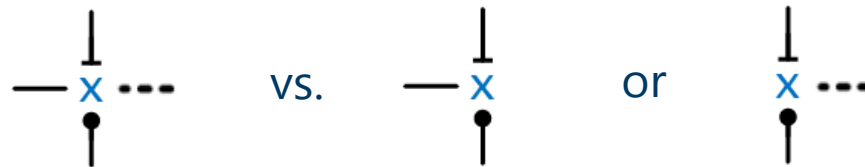
Step 3: only one active state per species

- Remove the catalytic activity of x_2
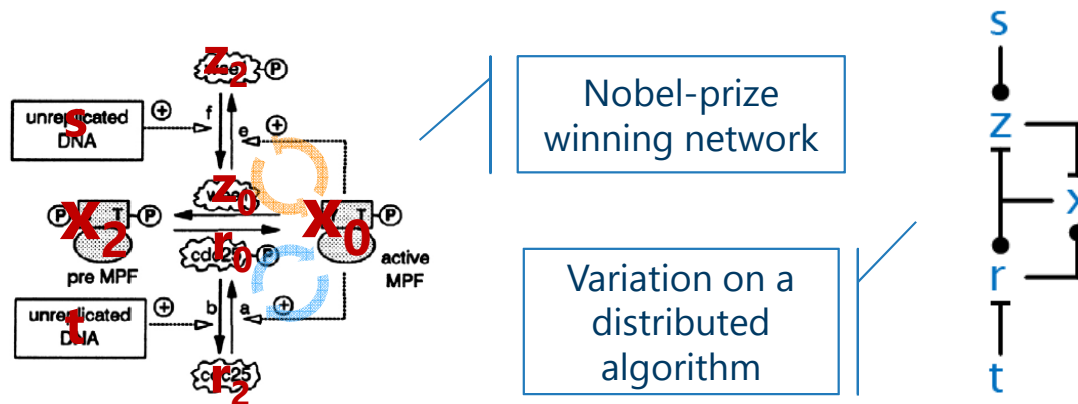 - By "flipping the z feedback to the other side"
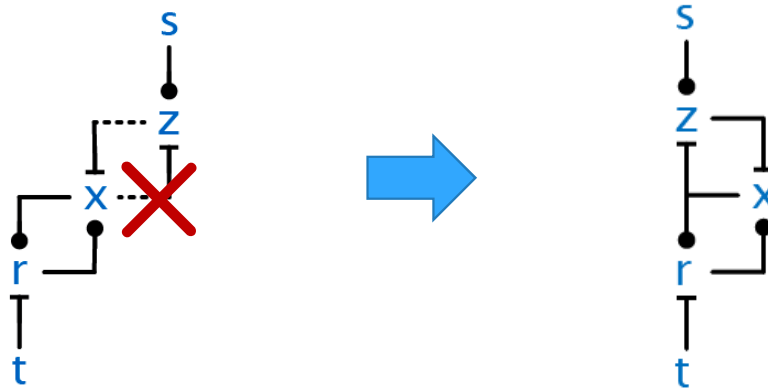
(x_2 promotes z_0 via s bias,
 z_0 promotes x_2 via inhibiting x_0)

(x_0 promotes r_0, promotes x_0)

- All species now have one active (x_0,z_0,r_0) and one inactive (x_2,z_2,r_2) form
- This is 'biochmically plausible'

Done

- … and that is the cell-cycle switch!

Nobel-prize winning network

Variation on a distributed algorithm

- But did we preserve the AM function through our network transformations?
- Ideally: prove either that the networks are 'contextually equivalent' or that the transformations are 'correct'
- Practically: compare their 'typical' behavior

Convergence Analysis

- Switches as computational systems
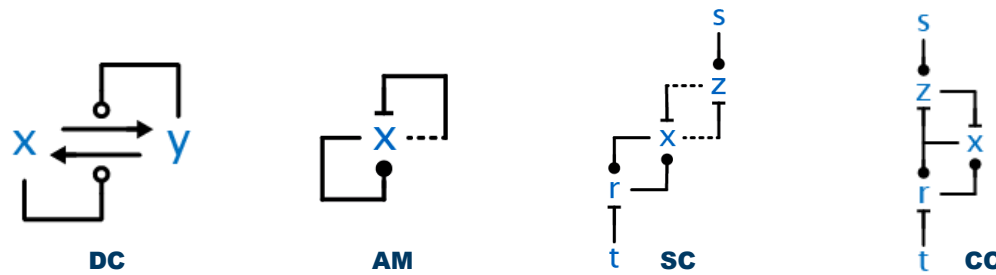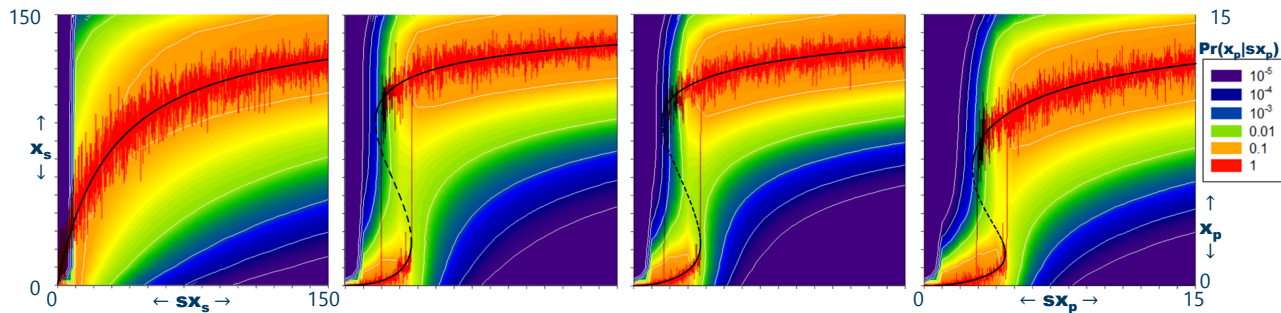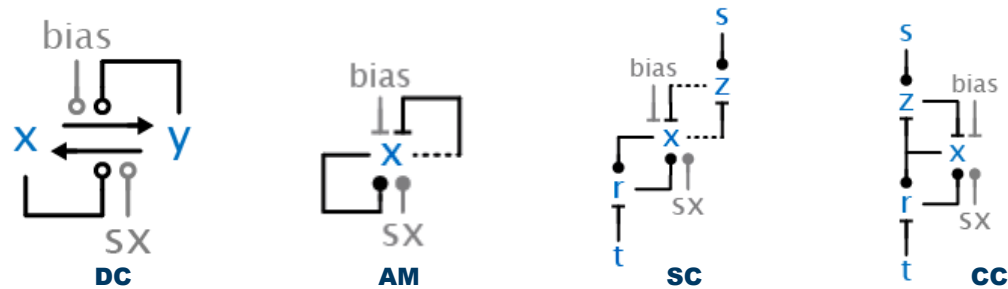

Start symmetrical
($x_0 = x_1 = x_2$ etc.)
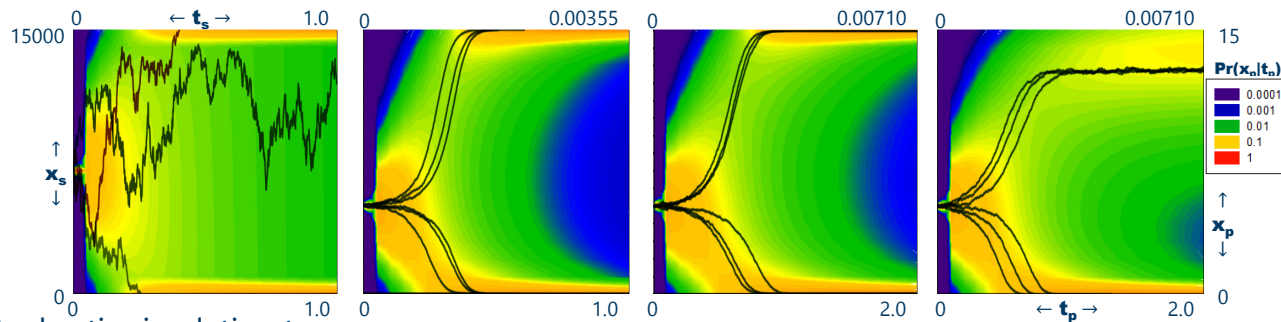
Black lines: several stochastic simulation traces
Color: full probability distribution of small-size system

NEW!
CC appears to converge in log time

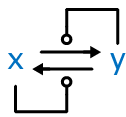# Steady State Analysis

- Switches as dynamical systems

DC   AM   SC   CC

Black lines: deterministic ODE bifurcation diagrams
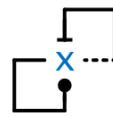Red lines: noisy stochastic simulations
Color: full probability distribution of small-size system

NEW!
AM shows hysteresis
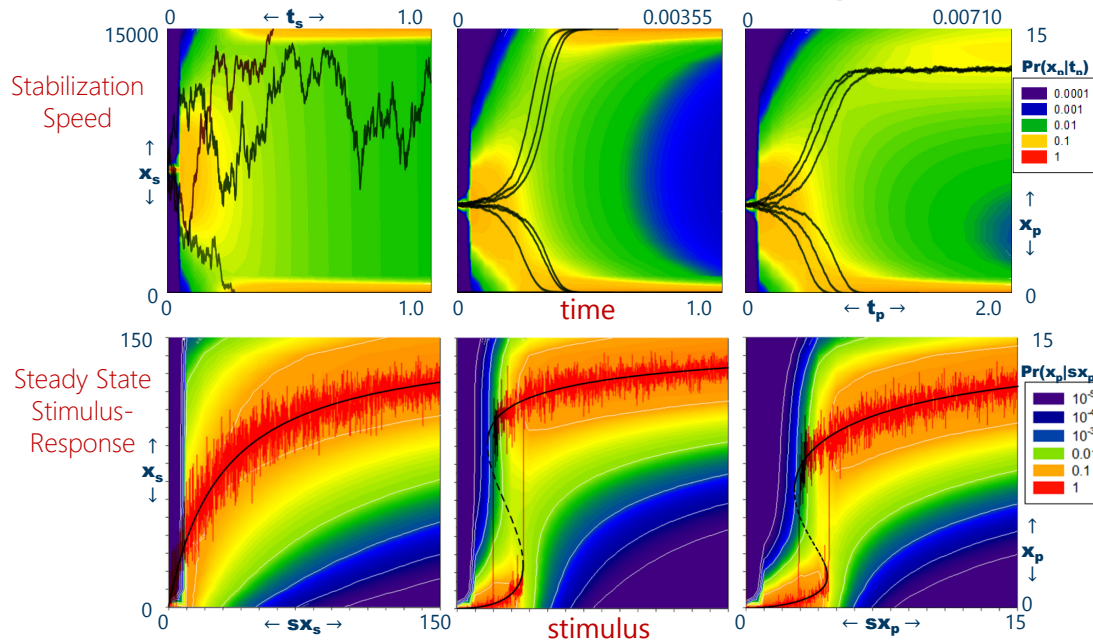
In Previous Work

activation
inhibition
catalysis

(a "bad" switch) **DC**   **AM**   **CC**

Stabilization Speed

$\uparrow$ $\mathbf{x_s}$ $\downarrow$

$\leftarrow \mathbf{t_s} \rightarrow$

time

$\leftarrow \mathbf{t_p} \rightarrow$

$\uparrow$ $\mathbf{x_p}$ $\downarrow$

Pr(x_n|t_n)
0.0001
0.001
0.01
0.1
1

Steady State Stimulus-Response

$\uparrow$ $\mathbf{x_s}$ $\downarrow$

$\leftarrow \mathbf{sx_s} \rightarrow$

stimulus

$\leftarrow \mathbf{sx_p} \rightarrow$

$\uparrow$ $\mathbf{x_p}$ $\downarrow$

Pr(x_p|sx_p)
10^{-5}
10^{-4}
10^{-3}
0.01
0.1
1
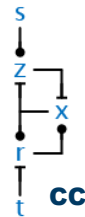
The "classical" Cell Cycle Switch CC approximates AM performance

SCIENTIFIC REPORTS

The Cell Cycle Switch Computes Approximate Majority

SUBJECT AREAS:
COMPUTATIONAL BIOLOGY

Luca Cardelli[1] & Attila Csikász-Nagy[2,3]

CC converges in O(log n) time (like AM)
(but 2x slower than AM, and does not fully switch)

Symmetrical initial conditions ($x_0=x_1=x_2$)

Black lines: high-count stochastic simulation traces
Color: full probability distribution of low-count system

Hor axis is *time*.

AM shows hysteresis (like CC)

Black lines: deterministic ODE bifurcation diagrams
Red lines: medium-count stochastic simulations
Color: full probability distribution of low-count system

Hor axis is *stimulus* pushing towards x_0 against fixed bias.

There is an *obvious* bug in CC performance!

31

# Evidence that CC is 'similar' to AM

- But there is a difference
  - The classical cell cycle switch, CC, works ok but never as well as AM
 - The output of CC does not go 'fully on':
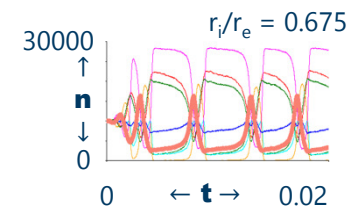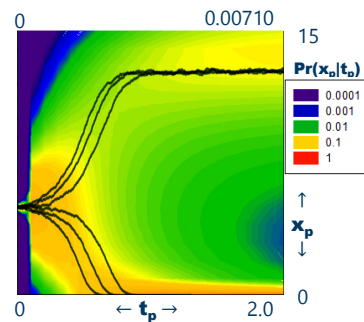
  - Because s continuously inhibits x through z, so that x cannot fully express
  - Engineering question: could *we* fix it? (Yes: let x inhibit s and t)
  - Q: Why didn't nature fix it?

Nature fixed it!

- ## There is another known feedback loop
  - By which x suppresses s "in retaliation" via the so-called Greatwall loop
  - Also, s and t happen to be the same molecule (=s)
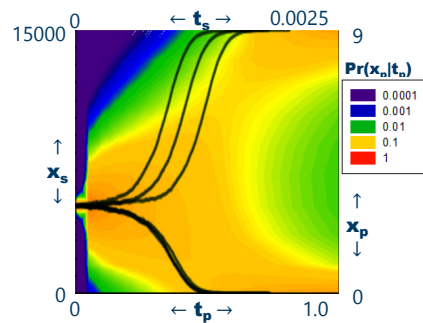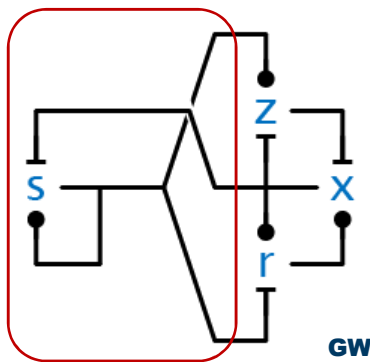
Full activation!
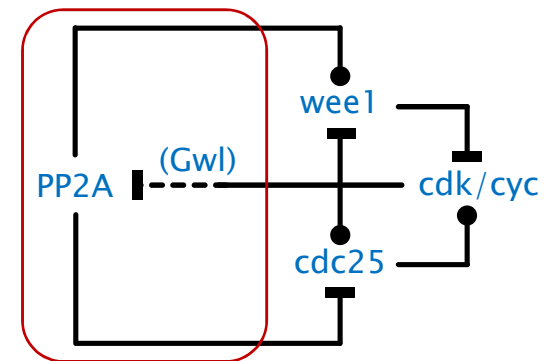
- s and x now are antagonists: they are the two halves of the switch, mutually inhibiting each other (through intermediaries).

More surprisingly

- ## Makes it faster too!
  - The extra feedback also speeds up the decision time of the switch, making it about as good as the 'optimal' AM switch:

Conclusion (in our published paper):
Nature is trying as hard as it can to implement an AM-class algorithm!

The "classical" cell cycle switch seems to be only half of the picture: the extra feedback completes it *algorithmically*.
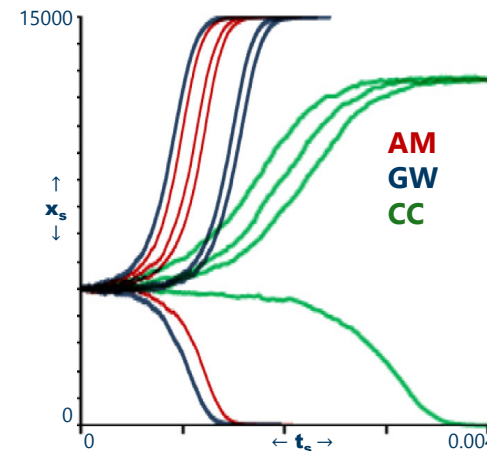
The Greatwall Kinase

- ## Our paper appeared:
  - Suggesting GW is a better switch than CC. *September 2012*

- ## Another paper that same week:
  - Showing experimentally that the Greatwall loop is a necessary component of the switch, i.e. the not-as-good-as-AM network has been 'refuted'
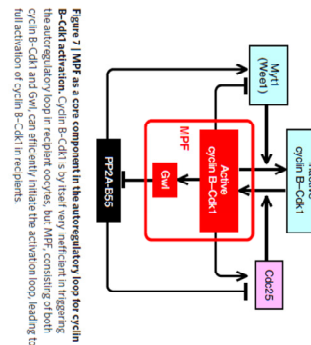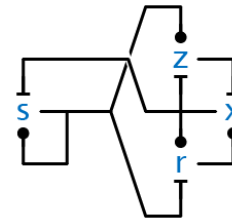
Figure 7 | MPF as a core component in the autoregulatory loop for cyclin B-Cdk1 activation. Cyclin B-Cdk1 is by itself very inefficient in triggering the autoregulatory loop in recipient oocytes, but MPF, consisting of both cyclin B-Cdk and Gwl, can efficiently initiate the activation loop, leading to full activation of cyclin B-Cdk1 in recipients.

The Argument So Far

- Relating dynamical and computational systems in isolation (as closed systems)
  - The AM algorithm (network) implements an input-driven switching function (in addition to the known majority function).
  - The CC algorithm implements a input-less majority function (in addition to the known switching function).
  - The structures of AM and CC are related, and an intermediate network shares some properties of both.

- But what about the context?
  - Will AM and CC behave similarly in any context (as open systems)?
  - That's a hard question, so we look at their intended context: implementing oscillators.
  - Also, oscillators are almost the 'worst case' contexts: very sensitive to component behavior.

Oscillators

- Basic in Physics, studied by simple *phenomenological* (not structural) ODE models.

- Non-trivial in Chemistry: it was only discovered in the 20's (Lotka) that chemical systems can (theoretically) oscillate: before, oscillation was thought impossible. Shown experimentally only in the 50's.

- Mechanics (since antiquity) and modern Electronics (as well as Chemistry) must engineer the *network structure* of oscillators.

- Biology: all natural cycles are oscillators. Here we must reverse engineer their network structure.

- Computing: how can populations of agents (read: molecules) interact (network) to achieve oscillations?

Cell Cycle Oscillator

- The cell cycle switch is part of an oscillator network
  - The cell cycle oscillation: grow-divide-grow-divide...
- The principle of the oscillator
  - Two interconnected switches yield a limit-cycle oscillator; e.g. two AM switches
  - In a Trammel of Archimedes configuration (gray rates < black rates)
  - (The biological network lacks some of these links and still oscillates)
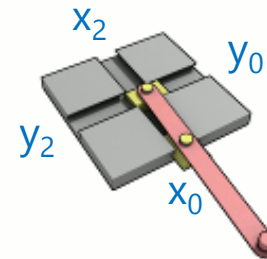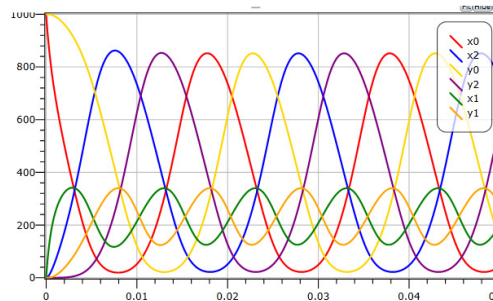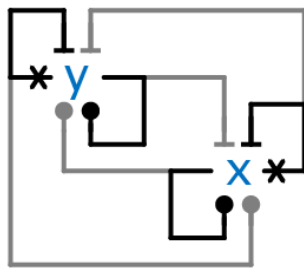
en.wikipedia.org/wiki/Trammel_of_Archimedes
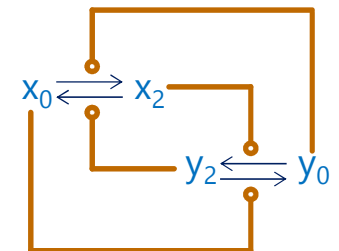
The Trammel of Archimedes

- ## A device to draw ellipses

  - Two interconnected **switches**.
  - When one switch is on (off) it flips the other switch on (off). When the other switch is on (off) it flips the first switch off (on).
  - The amplitude is kept constant by mechanical constraints.
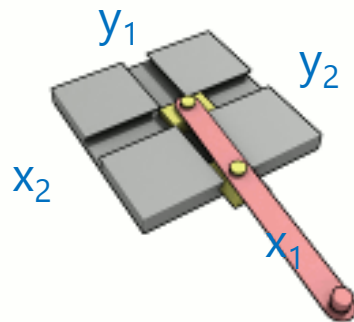
en.wikipedia.org/wiki/Trammel_of_Archimedes
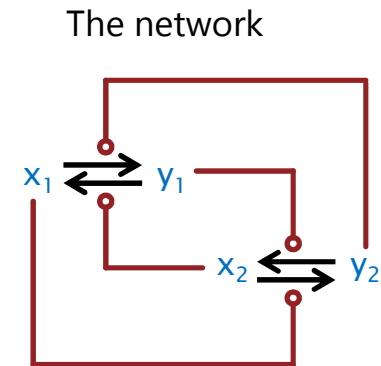
The Shishi Odoshi

- A Japanese scarecrow (*lit.* scare-deer)
  - Used by Bela Novak to illustrate the cell cycle switch.
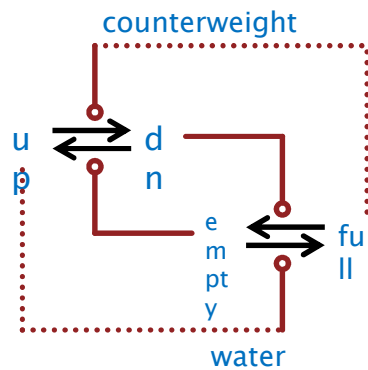
empty + up → up + full
up + full → full + dn
full + dn → dn + empty
dn + empty → empty + up

http://www.youtube.com/watch?v=VbvecTlftcE&NR=1&feature=fvwp
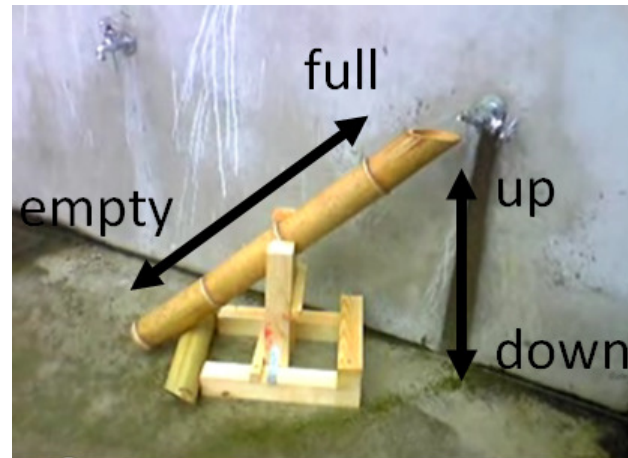
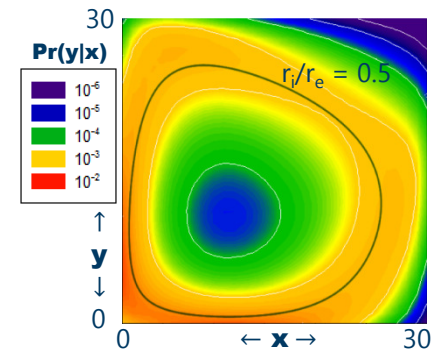Outer switched connections replaced by constant influxes: tap water and gravity.
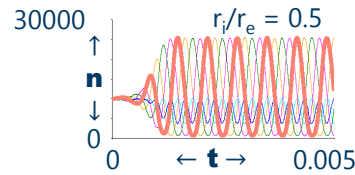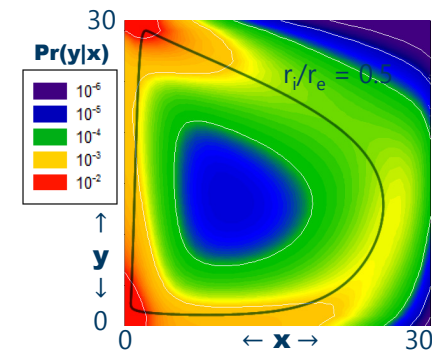
Contextual Analysis
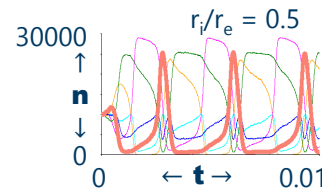
- AM switches in the context of oscillators
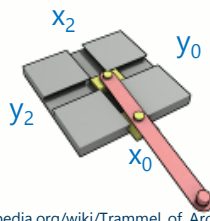
en.wikipedia.org/wiki/Trammel_of_Archimedes
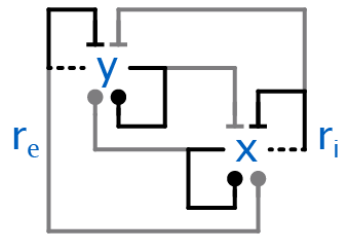
Trammel of Archimedes
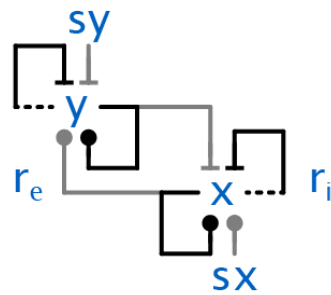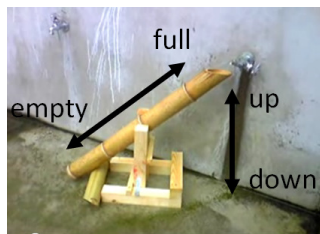
Shishi Odoshi

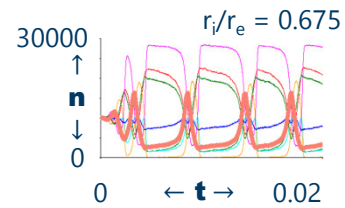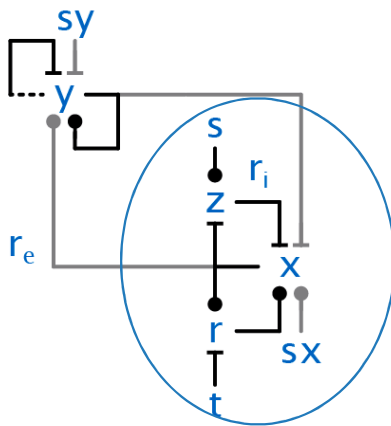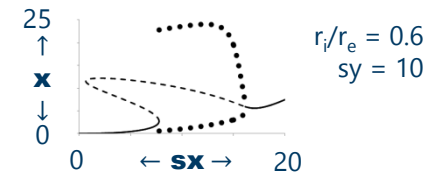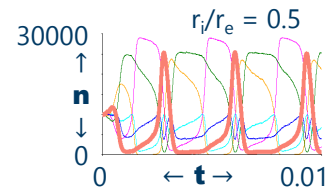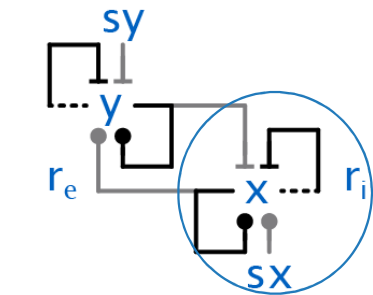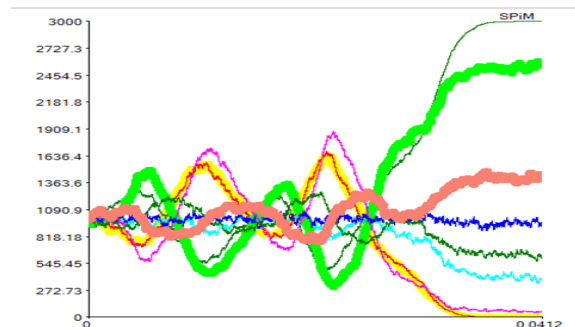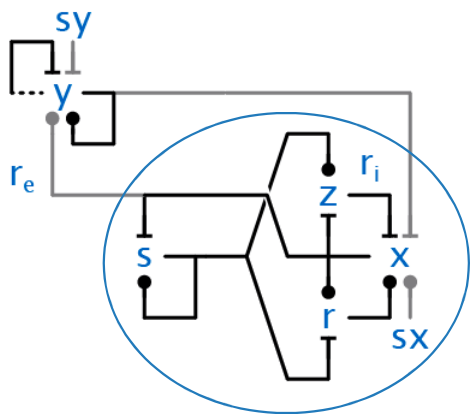# Modularity Analysis

- CC swapped in for AM
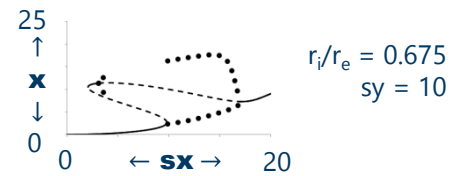
A new cell cycle switch candidate: GW

- Will it work in the normally-wired oscillator?

- Absolutely not! ☹
  - The x stable state is just too strong: a high x will shut down s completely; which means that r will be fully on, and it in turn will reinforce x fully. And y can never be strong enough to push down x when x-r are in such a strong mutual feedback. No amount of fiddling seems to give enough control on that situation.

However this will

- Put s under control of y so it can undermine x
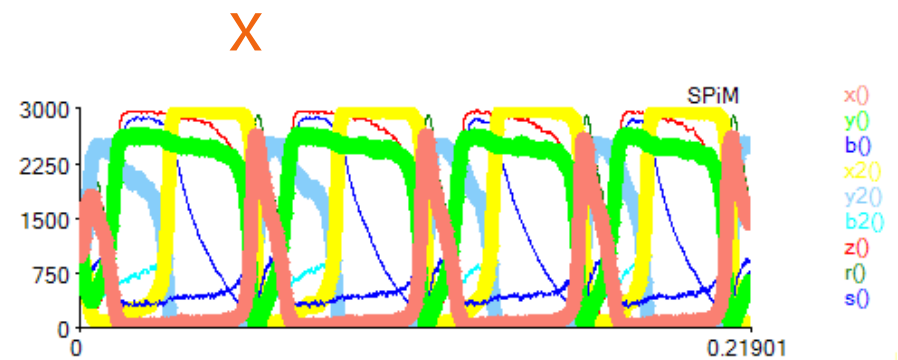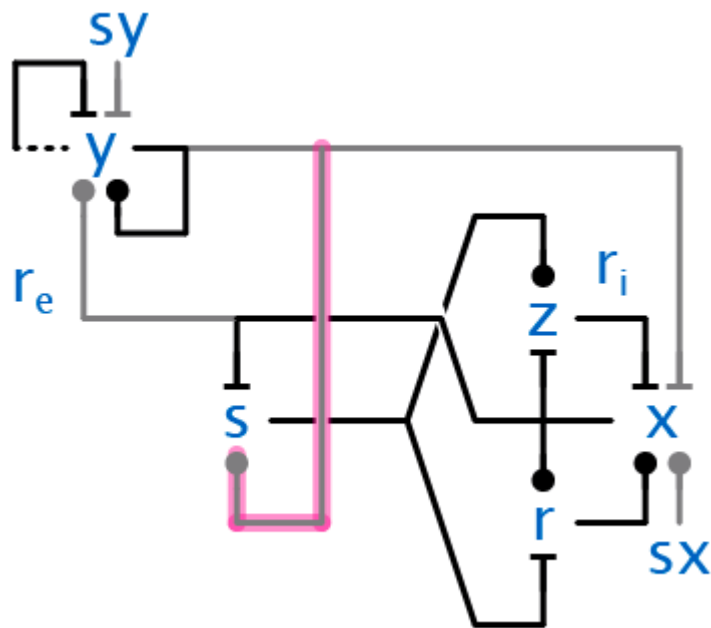
Robust full-on oscillation with all-default parameters (all black rates 1.0, all gray rates 0.5, all initial quantities equal)

Suggests a new interaction

- Either Gwl or PP2A or something along that path must be under control of cdc20.

- There are some hints in the literature that this may be the case, but no direct experimental validation.

45
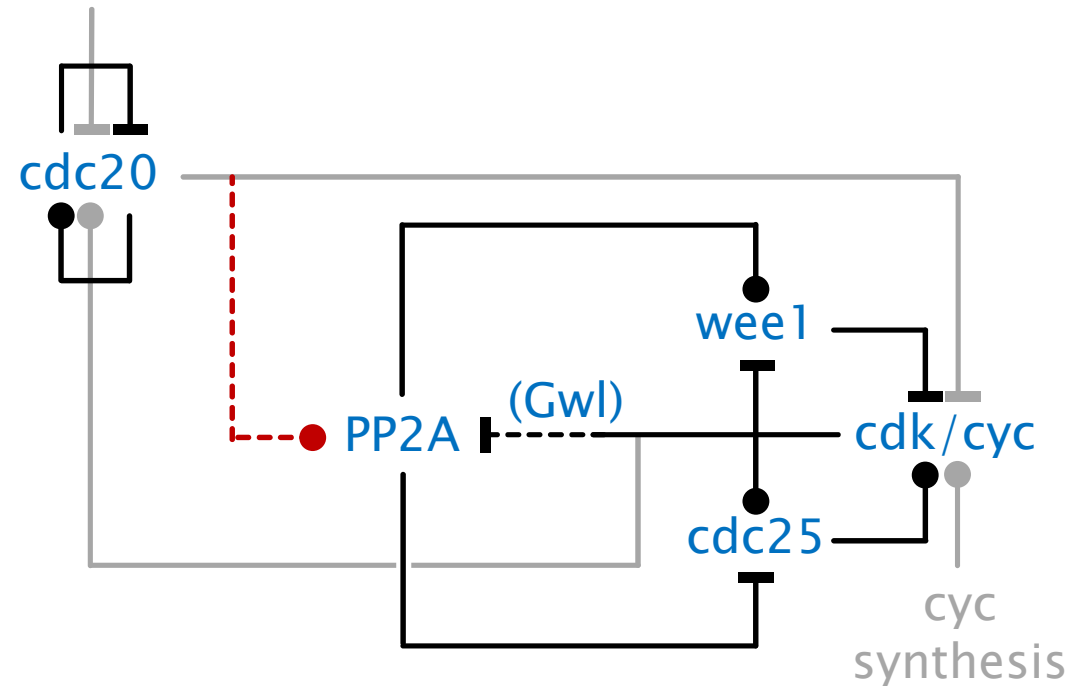
Suggests a new problem

- What is the 'proper' way to wire-in a replacement circuit?

- There is an answer (given by a study of network morphisms)

But what about network equivalence?

- Our evidence is empirical

  - Although quantitative and covering both kinetic and steady state behavior
  - Also, contextual equivalence holds in the context of oscillators (see paper)

- Analytical evidence is harder to obtain

  - The proof techniques for the AM algorithm are hard and do not generalize easily to more complex networks
  - Quantitative theories of behavioral equivalence and behavioral approximation, e.g. in process algebra, are still lacking (although rich qualitative theories exist)

47

Summary

- The structure of AM implements an input-driven switching function (in addition to the known majority function).
- The structure of CC/GW implements a input-less majority function (in addition to the known switching function).
- The structures of AM and CC/GW are related, and an intermediate network shares the properties of both.
- The behaviors of AM and CC/GW in isolation are related.
- The behaviors of AM and CC/GW in oscillator contexts are related.
- A refinement (GW) of the core CC network, known to occur in nature, improves switching performance and brings it in line with AM performance.