Morphisms of Reaction Networks

Luca Cardelli, Microsoft Research & Oxford University

related work: Attila Csikász-Nagy, Neil Dalchau, Mirco Tribastone, Max Tschaikowski, Andrea Vandin

HSB, Madrid 2015-09-04
Cellular Computation

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

- How do cells compute?
  - *Clearly* doing "information processing"
  - What are their computational principles?

More concretely

• Give substance to the claim that “cells compute”
  • Yes, but *what* do they compute?

• Catch nature red-handed in the act of running a computational task
  • Something that a computer scientist would recognize as an *algorithm*
Chemical Algorithms
Can *Chemistry* Compute?

- If we believe that biology can do computation...
  - It must be somehow based on chemistry

- So, can chemistry compute, and how?
  - That is in itself a very interesting question with non-trivial answers
Chemical Programming Examples

**specification**

\[
Y := \min(X1, X2)
\]

**program**

\[
X1 + X2 \rightarrow Y
\]

\[
Y := \max(X1, X2)
\]

\[
X1 \rightarrow L1 + Y
\]

\[
X2 \rightarrow L2 + Y
\]

\[
L1 + L2 \rightarrow K
\]

\[
Y + K \rightarrow 0
\]

\[
\text{max}(X1,X2) = (X1+X2) - \min(X1,X2)
\]

(but is not computed “sequentially”: it is a form of concurrent computation)

*(bio-*)chemical reaction network
Biochemical Networks

Across species: Ortholog genes

Within species: Paralog genes

“same function”

“new function”
How do we know networks exist?

- If you can break it, it must exist
  - Genome sequencing identifies genes (their “coding” regions)
  - Sequence comparison identifies orthologs and paralogs
  - Gene-produced proteins are isolated or synthetically produced in vitro or in vivo (all difficult)
  - Their qual/quant interactions are studied (often only in vitro)
  - Their 3D structure is determined (may take decades)
  - Networks are hypothesized, often qualitatively
  - Models are build, quantitative function is inferred
  - Further experiments (such as gene knockouts) are performed to break the network.

- Genes and networks are compared across and within species
  - High-value activity: 2001 Nobel prize in Physiology for the discovery of “Key regulators of the cell cycle ... they have identified key molecules that regulate the cell cycle in all eukaryotic organisms, including yeast, plants, animals, and human.” These are actually not (currently) the same molecules, but it is (still) “the same network” in all of them.
Simplified example

- Genes for x, s, r identified
- Say protein x exists in high quantity
  - Knock gene-x out: one protein goes missing, that must be x’s protein
- Say proteins s exists in “undetectable” quantities
  - Maybe 10~100 copies per cell on average: it cannot be found
  - Knock gene-s out: nothing seems to go missing, but the network’s function stops
    - Then we know protein-s must be in the network, although we don’t know “where”
- Heterogeneous system
  - It is indeed the case (in this cell-cycle-switch example) that x is “deterministic” (high copy count), while s,r, are “stochastic” (very low copy count) and yet s,r control x.
Consensus Networks
A Consensus Problem

- Population Consensus
  - Given two populations of $x$ and $y$ “agents”
  - We want them to “reach consensus”
  - By converting all agents to $x$ or to $y$
    depending on which population was in majority initially

- Population Protocols Model
  - Finite-state identity-free agents (molecules) interact in randomly chosen pairs ($\Rightarrow$ stochastic symmetry breaking)
  - Each interaction (collision) can result in state changes
  - Complete connectivity, no centralized control (well-mixed solution)

*specification*

\[
X, Y := X + Y, 0 \quad \text{if} \quad X_0 \geq Y_0 \\
X, Y := 0, X + Y \quad \text{if} \quad Y_0 \geq X_0
\]
A Consensus Algorithm

- Approximate Majority (AM) Algorithm
  - Uses a third “undecided” population $b$
  - Disagreements cause agents to become undecided
  - Undecided agents agree with any non-undecided agent

\[
\begin{align*}
    x + y &\rightarrow_r y + b \\
    y + x &\rightarrow_r x + b \\
    b + x &\rightarrow_r x + x \\
    b + y &\rightarrow_r y + y
\end{align*}
\]
A Biological Implementation

Approximate Majority (AM)

1) Bistable
   Even when initially x=y (stochastically)

2) Fast (asymptotically optimal)
   $O(\log n)$ convergence time

3) Robust to perturbation
   above a threshold, initial majority wins whp

Epigenetic Switch

Dana Angluin · James Aspnes · David Eisenstat
A Simple Population Protocol for Fast Robust Approximate Majority 2007

Figure 1. Basic Ingredients of the Model

Cell
Theoretical Analysis of Epigenetic Cell Memory by Nucleosome Modification

2007
Here We Got Lucky

• We can claim that the epigenetic switch is a *direct* biological implementation of an algorithm
  • Although we may have to qualify that with some notion of approximation of the (enzymatic) kinetics

• In most cases the biological implementation seems more *indirect* or *obfuscated*
  • “*Nature is subtle but not malicious – Einstein*” Ha! think again!
  • Other implementations of Approximate Majority seem more convoluted and approximate
How to model "Influence"

"True" molecular interactions.

"Equivalent" influence interactions.

Chemical Reaction Network

Influence Network

Evolving a Primitive Eukaryotic Cell Cycle Model

Malte Lücken, Jotun Hein, Bela Novak

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 Triplet Model of Influence

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

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:

For example:

Approximate Majority
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, G₂/M transition:
    - 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

- 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
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

- 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)
A Good Algorithm

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

- **With high probability, for $n$ agents**
  - The total number of interactions before converging is $O(n \log n)$
    $\implies$ fast (optimal)
  - The final outcome is correct if the initial disparity is $\omega(\sqrt{n} \log n)$
    $\implies$ 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 $O(\log n)$

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

**Activation inhibition**

Dana Angluin - James Aspnes - David Eisenstat

A Simple Population Protocol for Fast Robust Approximate Majority
An “Ugly” Algorithm: Cell Cycle Switch

- Is it a good algorithm? Is it bad?
- Is it optimal or suboptimal?
Convergence Analysis – CONSENSUS

• Switches as computational systems

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

Start symmetrical
\((x_0=x_1=x_2\text{ etc.})\)

Black lines: several stochastic simulation traces
Color: full probability distribution of small-size system
Steady State Analysis - SWITCH

- Switches as dynamical systems

Black lines: deterministic ODE bifurcation diagrams
Red lines: noisy stochastic simulations
Color: full probability distribution of small-size system
The Cell Cycle Switch Computes AM

- 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’
Antagonistic Networks

1 vs. 1
Mutual Inhibition & Self Activation

1 vs. 1
Mutual Inhibition & Mutual Anti-activation

2 vs. 2
low Notch $\rightarrow$ high Delta

3 vs. 3
high Delta $\rightarrow$ high Notch

Cell cycle transitions

Polarity establishment

Gene networks

Construction of a genetic toggle switch in Escherichia coli

The “new” cell cycle switch

Phosphorylation network dynamics in the control of cell cycle transitions

Dynamics of Drosophila asymmetric establishment

Patterned expression of Notch and Delta in the zebrafish lateral line

Lateral inhibition through Delta-Notch signaling: A review

SOURCE: [Origin of the Antagonistic Networks](#)
Network Morphisms

When does a (complex) network implement a (simpler) algorithm?
Comparing networks

- How can we compare different networks?
  - Different number of species
  - Different number of reactions
  - Apparently unrelated connectivity

- So that we can compare their function?
  - Does antagonism (in network structure) guarantee bistability (in function)?

- We do it by *mapping* networks onto one another so that they *emulate* each other
  - Deterministic semantics version of “simulation” of systems
  - (Stochastic semantics was the starting point, but too difficult/demanding for typical biological networks.)
Mapping one network into another

- Notion is strangely missing from the literature
  - Seen in Biology: single-network analysis (e.g. structure of feedback loops) and network reduction (e.g. while preserving steady states). Study of common or frequent subnetworks.
  - Seen in C.S.: comparing network behaviors (e.g. morphisms of event structures).
  - Nothing much resembling (bi)simulation "on the syntax" (structure) of whole biochemical networks.

- Model reduction is unavoidable and pervasive, but
  - Often criticized/ignored by biologists when it leads to quantities that are “not biologically meaningful”. E.g. a fusion or change a variables in the ODEs where the new variables do not correspond to biological parts. The reduced model should “inform” the original one.

- Science’s ethos
  - The “truth” is the big network, not the small one!
    If you depart from the truth in any way, you have to explain how you can get back to it.
  - The point is not to reduce the size of the network (although that’s neat), but to understand aspects of the big network by reference to a smaller one.
  - The mapping is more important than either networks.

Norbert Wiener
Pioneer of stochastic processes and inventor of Cybernetics.

“The best material model of a cat is another, or preferably the same, cat”
Network Emulation  E.g.: MI emulates AM

- For *any* rates and initial conditions of AM, we can find *some* rates and initial conditions of MI such that the (6) trajectories of MI retrace those (3) of AM:

  - How do we find these matching parameters? By a network morphism!
CRN Morphisms

A CRN morphism from \((S, R)\) to \((\hat{S}, \hat{R})\) written \(m \in (S, R) \rightarrow (\hat{S}, \hat{R})\)
is a pair of maps \(m = (m_S, m_R)\)
  - a species map \(m_S \in S \rightarrow \hat{S}\)
  - a reaction map \(m_R \in R \rightarrow \hat{R}\)

extended to a complex map \(m_S \in \mathbb{N}^S \rightarrow \mathbb{N}^{\hat{S}}\)
linearly: \(m_S(\rho)_{\hat{S}} = \Sigma_{S \in m_S^{-1}(\hat{S})} \rho_S\)
How to check emulations

• How do we check a potential emulation morphism for all possible initial conditions of the target?
  • Statically! Check conditions on the joint stoichiometric matrices of the two networks under the mapping.

• How do we check a potential emulation morphism for all possible rates of the target?
  • Can’t; but if one emulation is found, then the rates of the target network can be changed arbitrarily and a related emulation will again exist.
Network Emulation: MI emulates AM

A mapping of species and reactions

A homomorphic mapping

\[ z \rightarrow x \]
\[ \sim y \rightarrow x \]

**initial conditions:**

\[ z_0 = y_2 = x_0 \]
\[ z_1 = y_1 = x_1 \]
\[ z_2 = y_0 = x_2 \]

It need not preserve the out-degree of a node!
Network Emulation: SI emulates AM

A mapping of species and reactions

*homomorphic* mapping

\[ z \rightarrow x \]
\[ \sim y \rightarrow x \]
Network Emulation: NCC emulates MI

- For any rates and initial conditions of MI we can find some rates and initial conditions of NCC such that the (18) trajectories of NCC retrace those (6) of MI.

\[
\begin{align*}
\text{NCC: } & \quad \langle z, r, p \rangle \rightarrow \langle z \rangle \\
\text{MI: } & \quad \langle y, q, s \rangle \rightarrow \langle y \rangle \\
\text{(3 species each)}
\end{align*}
\]
The (18) trajectories NCC can *always* retrace those (3) of AM.

The new cell cycle switch can emulate AM *exactly*. For *any* initial conditions of AM.

And for *any* rates of AM.
Emulations are Modular
Static Test for Emulation

Emulation Theorem: If $m \in (S, R) \rightarrow (\hat{S}, \hat{R})$ is a CRN reactant morphism and stoichiomorphism then it is a CRN emulation.

Reactant morphism: $m_S^T \cdot \rho = \hat{\rho} \cdot m_R^T$

Stoichiomorphism: $\varphi \cdot m_R = m_S \cdot \hat{\varphi}$

Emulation: $\forall \hat{\mathbf{v}}. \ F(\hat{\mathbf{v}} \circ m_S) = \hat{F}(\hat{\mathbf{v}}) \circ m_S$

$F$ is the differential system of $(S, R)$, given by the law of mass action, $\hat{\mathbf{v}}$ is a state of $(\hat{S}, \hat{R})$, $\varphi$ is the stoichiometric matrix and $\rho$ is the related reactant matrix. $m_S$ and $m_R$ are the characteristic 0-1 matrices of the morphism maps $m_S$ (on species) and $m_R$ (on reactions). $-^T$ is transpose.

Stoichiomorphisms condition is sufficient for “networks of interest” and actually “close” to a necessary condition.
Emulation is (Backward) Bisimulation

Definition 13 (Cumulative flux rate). Let \((S, R)\) be a CRN, \(X \in S\), \(\rho \in \mathcal{M}S(S)\), and \(\mathcal{M} \subseteq \mathcal{M}S(S)\). Then, we define

\[
fr(X, \rho) := \sum_{\rho \xrightarrow{\alpha} \pi \in R} (\pi(X) - \rho(X)) \cdot \alpha, \quad fr[X, \mathcal{M}] := \sum_{\rho \in \mathcal{M}} fr(X, \rho).
\]

We call \(fr(X, \rho)\) and \(fr[X, \mathcal{M}]\) \(\rho\)-flux rate and cumulative \(\mathcal{M}\)-flux rate of \(X\), respectively.

Definition 14 (Backward CRN bisimulation). Let \((S, R)\) be a CRN, \(\mathcal{R}\) an equivalence relation over \(S\), \(\mathcal{H} = S/\mathcal{R}\) and \(\mu\) the choice function of \(\mathcal{H}\). Then, \(\mathcal{R}\) is a backward CRN bisimulation (BB) if for any \((X, Y) \in \mathcal{R}\) it holds that

\[
fr[X, \mathcal{M}] = fr[Y, \mathcal{M}] \quad \text{for all} \quad \mathcal{M} \in \{\rho \mid \rho \xrightarrow{\alpha} \pi \in R\}/\approx_{\mathcal{H}},
\]

where any two \(\rho, \sigma \in \mathcal{M}S(S)\) satisfy \(\rho \approx_{\mathcal{H}} \sigma\) if \(\mu(\rho) = \mu(\sigma)\).

Theorem 17 (Backward bisimulation characterizes exact fluid lumpability). Let \((S, R)\) be a CRN. Then, \(\mathcal{H}\) is an exactly fluid lumpable partition of \(S\) if and only if \(\mathcal{H}\) is a BB of \(S\).

An emulation between two CRNs can be understood in terms of a backward CRN bisimulation over the species of a “union CRN” that contains all the species and reactions of the two CRNs of interest.
Applications of BB

- Model Reduction
  - Find reduced networks
  - Compute quotient CRNs
  - Find network symmetries that may be of biological interest
- Morphism Generation
  - Find morphisms between networks (e.g. all the ones for a fixed rate assignment)

Benchmarks from Sneddon et al., Nature Methods, 2011

<table>
<thead>
<tr>
<th>Model</th>
<th>Reactions</th>
<th>Species</th>
<th>FB</th>
<th>Time (s)</th>
<th>BB</th>
<th>Time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>e9</td>
<td>3538944</td>
<td>262146</td>
<td>222</td>
<td>4.61E+4</td>
<td>222</td>
<td>7.65E+4</td>
</tr>
<tr>
<td>e8</td>
<td>786432</td>
<td>65538</td>
<td>167</td>
<td>1.92E+3</td>
<td>167</td>
<td>3.68E+3</td>
</tr>
<tr>
<td>e7</td>
<td>172032</td>
<td>16386</td>
<td>122</td>
<td>8.15E+1</td>
<td>122</td>
<td>1.77E+2</td>
</tr>
<tr>
<td>e6</td>
<td>36864</td>
<td>4098</td>
<td>86</td>
<td>3.00E+0</td>
<td>86</td>
<td>7.29E+0</td>
</tr>
<tr>
<td>e5</td>
<td>7680</td>
<td>1026</td>
<td>58</td>
<td>1.54E-1</td>
<td>58</td>
<td>4.06E-1</td>
</tr>
<tr>
<td>e4</td>
<td>1536</td>
<td>258</td>
<td>37</td>
<td>9.00E-3</td>
<td>37</td>
<td>1.09E-1</td>
</tr>
<tr>
<td>e3</td>
<td>288</td>
<td>66</td>
<td>22</td>
<td>1.00E-3</td>
<td>22</td>
<td>3.00E-3</td>
</tr>
<tr>
<td>e2</td>
<td>48</td>
<td>18</td>
<td>12</td>
<td>1.00E-3</td>
<td>12</td>
<td>2.00E-3</td>
</tr>
</tbody>
</table>

Forward and Backward Bisimulations for Chemical Reaction Networks

Luca Cardelli\textsuperscript{1}, Mireo Tribastone\textsuperscript{2}, Max Tchaikowski\textsuperscript{3}, and Andrea Vandin\textsuperscript{4}

\textsuperscript{1} Microsoft Research & University of Oxford, UK
\textsuperscript{2-4} University of Southampton, UK
luca@Microsoft.com
(m.triabstone,m.tchaikowski,a.vandin)@soton.ac.uk

Concur 2015
Approximate Majority Emulation Zoo

\[ \text{homomorphism and stoichiomorphism (transitive)} \]
Approximate Majority Emulation Zoo

(A homomorphism and stoichiormorphism (transitive))
Approximate Majority Emulation Zoo

Neutral paths in network space

Side jumps

(homomorphism and stoichiomorphism (transitive))
Stochasticity
The switch is noisy

- Biological conditions:
  - $x$ is abundant, $r,s$ are “undetectable”

- This situation does not emulate AM
  - Because of the extra low-count $r,s$ traces
  - BUT it emulates two separate copies of AM: one for $x$ and one (low-count) copy for $r/s$
  - Hence it is still (deterministically) a good switch in the AM family
  - In particular, the low count species can be effective regulators even though they are present in “undetectable” quantities.

- But, we can expect significant noise
  - On $r/s$ because they are in low-count
  - Likely on $x$ because it is regulated by $r/s$

CCr with $r,s$ at 1/10 of $x$, $r_0,s_0$ rates 10* the rates of $x_0,x_2$
Stochastic behavior

- We can in fact study the Chemical Master Equation

CCr with $r,s$ at $1/10$ of $x$, $r_0,s_0$ rates $10^*$ the rates of $x_0,x_2$

Mean and ±s.d. of species over time.

Mean and probability distribution of one species over time.
Trivial Example: AM vs. 2*AM

- Usually “more molecules” means “less noise”
- But not always
  - 2*AM emulates AM, hence the mean trajectories of 2*AM are the same as AM
  - The noise (s.d.) of 2*AM is also the same as AM
  - So, 2*AM has twice as many molecules, but noise is not reduced
- And not uniformly
  - MI, SI are two “intertwined” copies of AM
  - Are MI, SI less noisy than 2*AM?
  - Are MI, SI equally noisy? (They have the same number of molecules and reactions.)
Stochastic Switches

- Disentangle the contribution of complexity to stochasticity
- Compare network noise on the baseline of deterministic emulation, across networks of different size and structure

Noise Reduction in Complex Biological Switches

Luca Cardelli¹,³,⁷, *, Attila Csikász-Nagy¹,³,⁷, Neil Dalchau¹,⁷, Mirco Tribastone⁵,⁷, Max Tschaikowski⁵,⁷

(To appear.)
Stochastic Switches

- Network complexity *intrinsically* reduces noise

Noise Reduction in Complex Biological Switches

Luca Cardelli\(^{1,2,5}\), Attila Csiáksz-Nagy\(^{1,6,7}\), Neil Dalchau\(^{1,5}\), Mirco Tribastone\(^{8,9}\), Max Tschaiowski\(^{10}\)

(To appear.)

*Fig 6 – Complexity improves overall performance of the cell cycle switch.* The performance of different networks was evaluated by calculating the standard deviation of the main molecular states \(x_0\) or \(x_0\) depending on the network) over time. Standard deviations are calculated via numerical integration of the chemical master equation (CME) using the Visual GEC software, and via numerical integration of the central limit approximation (CLA) in Matlab. We investigate switching in one direction or the other by providing different initial conditions that settle (more likely) in different steady states. **(A)** In the forward direction, principal molecular states were initialised at 2 copies, and complementary molecular states were initialised at 1 copy, as shown in Fig 2C and Fig 5B. **(B)** In the reverse direction, principal molecular states were initialised at 1 copy, and complementary molecular states were initialised at 2 copies.
“Stochastic Network Morphisms”

?
Conclusions
Networks are Algorithms

- They are *methods* for achieving a function
  - We need to understand how these methods relate to each other
  - In addition to how and how well they implement function
  - Algorithms can be obfuscated, and nature can obfuscate networks (to what end?)

- Network emulation can be checked *statically*
  - By stoichiometric/reaction-rate (*structural*) properties
  - That is, no need to compare ODE (*functional*) properties
  - For *any* initial conditions and rates of (one of) the networks

- We can efficiently discover emulations
  - Automatic model reduction of large networks
Nature likes good algorithms

Approximate
“default” rates and initial conditions

Exact
any rates and initial conditions

These additional feedbacks do exist in real cell cycles (via indirections)

The cell cycle switch can exactly emulate AM
Feynman’s Blackboard

What I cannot create, I do not understand.

Know how to solve every problem that has been solved.