Network Transformations of Switches and Oscillators

Luca Cardelli
Microsoft Research

Cambridge 2011-08-02
http://lucacardelli.name
Motivation

- Building synthetic (DNA) oscillators

DSD simulation
The Trammel of Archimedes

- A device to draw ellipses
  - Two interconnected switches.
  - Note that amplitude is kept constant by mechanical constraints.
  - 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).

en.wikipedia.org/wiki/Trammel_of_Archimedes
The Shishi Odoshi

• A Japanese scarecrow (scare-deer)
  ○ Used by Bela Novak to illustrate the cell cycle switch.

empty + tap → tap + full
up + full → full + dn
full + dn → dn + empty
dn + empty → empty + up

To make it into a full trammel (dotted line), we could make the up position mechanically open the tap (i.e. take up = tap)
The Cell Cycle

• Feedback speeds
  o fast (post-translational)
  o slow (transcriptional)

• Some feedbacks may be missing

• Switches are asymmetric
  o One switch is usually simpler than that, just causing a negative feedback
  o One switch is usually more sophisticated than that, because of biochemical constraints
Outline

• Questions that nature has answered
  o Building ‘good’ bistable systems
  o Building ‘switches’ (switchable bistable system)
  o Building switches with hysteresys (needed for good oscillators)
  o Building limit-cycle oscillators
  o Building robust oscillators that resist parameter variations

• Engineering solutions to the same problems
  o Are they related?
  o In nature there are chemical constraints
    • Not all reactions can be easily implemented
    • Not all molecules can perform all functions we want them to

• From the point of view of network structure
  o Transforming a network and preserve some function
  o “Program transformations”
Switches
The Cell Cycle Switch

Why this network structure?

- Double positive feedback on x
- Double negative feedback on x
- No feedback on y

Why on earth .... ??
A Bad Algorithm

• Direct $x$–$y$ competition
  o $x$ catalyzes the transformation of $y$ into $x$
  o $y$ catalyzes the transformation of $x$ into $y$

  $x + y \rightarrow x + x$

  $y + x \rightarrow y + y$

• This system is bistable, but
  o Convergence to a stable state is slow (a random walk).
  o *Any* perturbation of a stable state can initiate a random walk to the other stable state.
  o With 100 molecules of $x$ and $y$, convergence is quick, but with 10000 molecules, even at the same concentration, you will wait for a long time.
A Very Good Algorithm

• Approximate Majority
  o Decide which of two populations is in majority

• A fundamental ‘population protocol’
  o Agents in a population start in state x or state y.
  o A pair of agents is chosen randomly at each step, they interact ("collide") and change state.
  o The whole population must eventually agree on a majority value (all x or all y) with probability 1.

---

A Simple Population Protocol for Fast Robust Approximate Majority

We analyze the behavior of the following population protocol with states $Q = \{b, x, y\}$. The state $b$ is the blank state. Row labels give the initiator’s state and column labels the responder’s state.

<table>
<thead>
<tr>
<th></th>
<th>x</th>
<th>b</th>
<th>y</th>
</tr>
</thead>
<tbody>
<tr>
<td>x</td>
<td>(x, x)</td>
<td>(x, x)</td>
<td>(x, b)</td>
</tr>
<tr>
<td>b</td>
<td>(b, x)</td>
<td>(b, b)</td>
<td>(b, y)</td>
</tr>
<tr>
<td>y</td>
<td>(y, b)</td>
<td>(y, y)</td>
<td>(y, y)</td>
</tr>
</tbody>
</table>
Properties

- Using martingales, we show that with high probability, the number of state changes before converging is $O(n \log n)$.
- The total number of interactions before converging is $O(n \log n)$.
- The final outcome is correct if the initial disparity is $\omega(\sqrt{n \log n})$.
- This algorithm is the fastest possible.
- Must wait $\Omega(n \log n)$ steps in expectation for all agents to interact.

[Angluin et al.]

“Parallel time” is the number of steps divided by the number of agents. Hence the algorithm terminates with high probability in $O(\log n)$ steps per agent.

N.B. this bound holds even if the $x, y$ populations are initially of equal size!
Chemical Implementation

\[ x + y \rightarrow y + b \]
\[ y + x \rightarrow x + b \]
\[ b + x \rightarrow x + x \]
\[ b + y \rightarrow y + y \]

Alternatives:

This too is a bistable system, but:
• It converges slowly, by a random walk, hence \( O(n^2) \).
• It is unstable: any random fluctuation from an all-x or all-y state can send it (by a random walk) to the other state.

This one gives no significant improvement over the above.
Majority of $x>y$

Gillespie simulation of the chemical reactions in SPiM.

$2000k$ molecules
$1100k$ $x$
$900k$ $y$

All rates are equal.

$x + y \rightarrow y + b$

$y + x \rightarrow x + b$

$b + x \rightarrow x + x$

$b + y \rightarrow y + y$

Eventually:
all $x$
no $y$
no $b$
Majority of $x=y$ (!!)

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

Eventually either:
all $x$ all $y$
no $y$ no $x$
no $b$ no $b$

2000k molecules
Gillespie simulation of the chemical reactions in SPiM.

All rates are equal.

The final majority is robust (insensitive to possible noise) because a significant majority always stays a majority:
The final outcome is correct if the initial disparity is
$$\omega(\sqrt{n \log n})$$

N.B. a deterministic (ODE) simulation with $x=y$ would not converge ever!
A Digression about Other Switches

• The AM network is an ‘optimal’ switch in a computational sense. How does it compare with other switches?

• Let us first compare the ‘kernel’ of AM without feedbacks (i.e. ‘double phosphorylation’) with the Goldbeter–Koshland switch

• And then compare the full AM network with GK plus the same feedbacks as AM
Double-Phosphorylation Switch

Ultrasensitive (but no hysteresis)

\[ x + E \rightarrow E + b \]
\[ b + E \rightarrow E + y \]
\[ y + F \rightarrow F + b \]
\[ b + F \rightarrow F + x \]

AM without feedbacks

Initially 10000 x, no y, 100 F, no E. E growing from 0 (t=100) to 3000 (t=400) then back to 0 (t=800)
The Goldbeter–Koshland Switch

Ultrasensitive (but no hysteresis)

\[ S + E \xrightarrow{a} SE \xrightarrow{k} P + E \]
\[ P + F \xrightarrow{a} PF \xrightarrow{k} S + F \]

Initially 10000 S, no P, 1000 F, no E.

E growing from 0 (t=100) to 2000 (t=300) then back to 0 (t=500)

The first switch happens at t=200, the second at t=400.

E/F ratio can be lower: GK is a ‘better’ more sensitive switch.
Can GK do majority switching?

GK in "AM configuration"

\[ S + P \xleftrightarrow{d}^a PS \rightarrow_k^a P + P \]
\[ P + S \xleftrightarrow{d}^a SP \rightarrow_k^a S + S \]

GK in "AM configuration" does not compute a majority.
- The initial minority goes down to 0
- The initial majority goes down to \( \text{maj}_{t=0} - \text{min}_{t=0} \)
- When \( \text{maj}_{t=0} \sim \text{min}_{t=0} \) the system cannot decide.
• Problem may be that the feedbacks put GK outside of zero–order regime.
• Hence, should check to see if GK works in the case of

\[
\begin{align*}
x + w & \xleftrightarrow{a} xw \rightarrow_k y + w \\
y + r & \xleftrightarrow{a} yr \rightarrow_k x + r \\
p + x & \xleftrightarrow{a} px \rightarrow_k r + x \\
r + t & \xleftrightarrow{a} rt \rightarrow_k p + t \\
w + s & \xleftrightarrow{a} ws \rightarrow_k z + s \\
z + y & \xleftrightarrow{a} zy \rightarrow_k w + y
\end{align*}
\]
‘Double phosphorylation’ motif is key

It is not just a non-linearity of the x–y transition mechanism that matters: it is the 'double phosphorylation' network structure of AM, with a common 'undecided' state.
Chemical Constraints

• The AM circuit is ‘chemically demanding’
  o It requires \(x\) molecules to be ‘next’ to \(y\) molecules because they interact directly
  o It requires both \(x\) and \(y\) to be catalysts, and in fact autocatalysts, and in fact each–other’s autocatalyst!
Network Transformations

• An example of relaxing those constraints
  o This circuit works just as well as the original, but it no longer requires x to be ‘next’ to y. They no longer interact directly. Instead, they interact through an additional $x_0$–$y_0$ equilibrium.

```plaintext
directive sample 0.0002 1000
directive plot x0; y0; b0
val r = 10.0
new xy0@r:chan new yx0@r:chan new x0y0@r:chan new y0x0@r:chan new bx@r:chan new by@r:chan
let x0 =
do ?xy0; b() or !bx; x() or !y0x0; x()
and y() =
do ?yx0; b() or !by; y() or !x0y0; y()
and b() =
do ?bx; x() or ?by; y()
and y0() =
do ?y0x0; x0() or !y0x0; x0()
and x0() =
do ?x0y0; y0() or !x0y0; y0()
run 5000 of x() run 5000 of x0() run 5000 of y() run 5000 of y0() run 5000 of x0() run 5000 of y0()
```

cf.
Network Transformations

• Another example of relaxing constraints
  o Build an Approximate Majority network that requires only $x$ to be a catalyst. How?
    o Enter the Cell Cycle switches…
Some Notation

• Catalytic reaction

\[ x + z \rightarrow z + y \]

\[ x \rightarrow y \]

• Double ‘kinase–phosphatase’ reactions

\[ x \rightleftharpoons y \]

\[ x \rightleftharpoons y \]
Zero–Input Switches

- ‘Zero–input switch’ = majority circuit: just working off the initial conditions, with no other inputs.

- Step 1: the original AM Network
Zero-Input Switches

- Step 2: remove auto-catalysis
  - By introducing intermediate species w, r.
  - Here w breaks the y auto-catalysis, and r breaks the x auto-catalysis, while preserving the feedbacks.
  - w and r need to ‘relax back’ (to z and t) when they are not catalyzed: s and t provide the back pressure.
... can simplify?

(it appears just slightly noisier/slower)
... no, it gets stuck!

- Equal-size initial conditions
Zero–Input Switches

• Step 3: transform a double–positive loop on $y$ into a double–negative loop on $x$.
  o Instead of $y$ (actively) activating itself through $w$, we have $z$ activating $y$ (which is passive). To counteract, now $x$ has to switch from inhibiting $y$ to inhibiting $z$.

  \[
  \begin{array}{c}
  \text{Step 1:} \\
  \text{Step 2:} \\
  \text{Step 3:} \\
  \end{array}
  \]

• So that $y$ no longer catalyzes anything
  o All species have one active and one inactive form
Zero–Input Switches

• Still an AM circuit

(The equal–likelihood outcome here is around 4500 y vs 5500 x, and can be adjusted by s/t ratio)

All rates are equal.
Equal-size initial conditions

All initial species = 10000. Probability of win seems to be x=y=50%
Note that when x wins, the system does not terminate because x has active competition from s.
AM Equal-size initial conditions

All initial species = 100000..
The Cell Cycle Switch

(Some of the bistable states can be enzymatic rather than AM.)
... can simplify?

It works better than the original?!?
... no, it gets stuck!

- Equal-size initial conditions
More Zero–Input Switches

- Other designs
  - A version with no external bias (s,t) where y is still non-catalytic and x and z are self-catalytic.
  - Both x and z have an ‘inactive’ form, y and w, although the both are double catalysts.
• Equal–size initial conditions
One–Input Switches

- Ultrasensitivity (none) and hysteresis (none) in trivial majority

```
let x() = !xcat; x() or ?ycat; y() or ?sycat; y()
let y() = !ycat; y() or ?xcat; x() or ?sxcat; x()
let sy() = !sycat; sy() or ?sykill; ()
let sx() = !sxcat; sx() or ?sxkill; ()

val rt = 10.0
val rx = 5.0
val sy = 1000

let schedule(n:int) = if n < 20000 then sx() else if n < 40000 then !sxkill;() else ()
run clock(schedule, 0.000001)
```
One–Input Switches

- Hysteresis in unbiased AM–like switches.
  - All rates are equal; constant amount of sy is sufficient for switch–back.
One–Input Switches

- Hysteresis in biased AM–like switches

```
x
  sy

<table>
<thead>
<tr>
<th>x</th>
<th>y</th>
</tr>
</thead>
<tbody>
<tr>
<td>rx</td>
<td>sx</td>
</tr>
</tbody>
</table>

let x() =
  do !xcat;
  x() or ?ycat;
  b() or ?sxcat;
  b()
  and y() =
  do !ycat;
  y() or ?xcat;
  b() or ?sxcat;
  b()
  and b() =
  do ?xcat;
  x() or ?sxcat;
  x() or ?ycat;
  y() or ?sycat;
  y() 
  and sy() =
  do !sycat;
  sy() or ?sykill;
  ()
  and sx() =
  do !sxcat;
  sx() or ?sxkill;
  ()
run 10000 of y()
run 1000 of sy()
let clock(p:proc(int), t:float) = (* Produce one p(m) every t sec with precision dt, 
  with m incremented from 0 *)
  (val dt= 100.0
  run step(p, 0, t, dt, dt))
and step(p:proc(int), m:int, t:float, n:float, dt:float) =
  if n<=0.0 then (p(m)|step(p,m+1,t,dt,dt))
  else delay@dt/t; step(p,m,t,n_1.0,dt)
let schedule(n:int) =
  if n < 10000 then sx() 
  else if n < 20000 then !sxkill;() 
  else () 
run clock(schedule,0.000001)
```
One–Input Switches

- Hysteresis vs. feedbacks in cell cycle switch

Without pos-pos feedback

Without neg-neg feedback

initial conditions:
1000 of y
1000 of z
1000 of p
1000 of t
200 of s
100 of sy

varying sx 0 to 1000 to 0

References:

[1] Reference System

[2] 2s, ½t, ½sy reference system

[3] All rates are equal.
Ferrell oscillator

Novak–Tyson oscillator
Two–input Switches

- I had rediscovered (but not analyzed so well) the same system, while looking for a memory circuit.
- The point here was not computing majority, but switching easily and quickly and stably.

\[ A + B \rightarrow B + C \]
\[ B + A \rightarrow A + C \]
\[ C + A \rightarrow A + A \]
\[ C + B \rightarrow B + B \]

---

**Artificial Biochemistry. Luca Cardelli**

In Figure 34 we show a modified version of the groupies, obtained by adding an intermediate state shared by the two state transitions. This automaton has very good memory properties. The top-left and top-center plots show that it is in fact spontaneously bistable. The bottom-left plot shows that it is stable in presence of sustained 10% fluctuations produced by doping automata. The bottom-center plot shows that, although resistant to perturbations, it can be switched from one state to another by a signal of the same magnitude as the stability level: the switching time is comparable to the stabilization time. In addition, this circuit reaches stability 10 times faster than the original groupies: the top-right plot shows the convergence times of 30 runs each of the original groupies with 2 states, the current automaton with 3 states, and a similar automaton (not shown) with 4 states that has two middle states in series. The bottom-right plot is a detailed view of the same data, showing that the automaton with 4 states is not significantly faster than the one with 3 states. Therefore, we have a stable and fast memory element.
Oscillators
The Trammel of Archimedes

• A device to draw ellipses
  o Two interconnected switches.
  o 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).

en.wikipedia.org/wiki/Trammel_of_Archimedes
The Shishi Odoshi

- A Japanese scarecrow (scare-deer)
  - Used by Bela Novak to illustrate the cell cycle switch.

To make it into a full trammel (dotted line), we could make the up position mechanically open the tap (i.e. take $up = tap$)
The 2AM Limit–Cycle Oscillator

• Two AM switches in a Trammel pattern

The red reactions need to be slower (even slightly) than the black reactions, but otherwise the oscillation is robust. Oscillation stops at 10 vs. 10 and 1 vs. 10. Here the rates are 8(red) vs 10(black) top, and 2 vs 10, bottom.

(Simple limit–cycle oscillators in the literature have very critical rate ranges.)
The Switch Module

This is flipped!
Replacing Switch Modules

Replace

outX \rightarrow x \rightarrow y \rightarrow outY

inXY

inYX

With

outY \rightarrow s \rightarrow z \rightarrow w \rightarrow p \rightarrow r \rightarrow t \rightarrow outX

inXY

inYX

Or

outY \rightarrow s \rightarrow z \rightarrow w \rightarrow p \rightarrow r \rightarrow t \rightarrow outX

inXY

inYX

Etc..
Modified Oscillator 1

```
directive sample 0.001 10000 directive plot x(); y(); b();
x2(); y2(); b2(); z(); r() valrt = 10.0 valst = 8.0
new ...
run 10000 of z() run 1000 of x() run 6666 of y() run 2333 of b() run 3333 of x2() run 3333 of y2() run 3333 of b2()
```

---

**Diagram Description**

- **x2cat** connects to **y2cat1**
- **x2cat1** is connected to **y2cat**
- **zcat1** is connected to **y2cat1**
- **w** and **z** are connected in a loop
- **x** and **y** are connected in a loop
- **p** and **r** are connected in a loop
- **t** is connected to **r**

**Graph**

- Plot of **x(), y(), b(), x2(), y2(), b2(), z(), r()**
- Horizontal axis: 0 to 0.0012564
- Vertical axis: 0 to 10000
- Multiple traces indicating oscillatory behavior

---

**Notes**

- Simulation parameters include **valrt = 10.0** and **valst = 8.0**
- Multiple runs for different variables and parameters
Modified Oscillator 2

```plaintext
2012_11_07 Luca Cardelli

s

w

x

y

z

p

r

outX outY inXY inYX

x2cat y2cat xcat1 y2cat1

valrt = 10.0
valst = 8.0
new...
runt 10000 of z()
runt 6666 of x()
runt 2333 of b()
runt 3333 of x2()
runt 3333 of y2()
runt 3333 of b2()
```

```
SPiM

x0
y0
b0
x20
y20
b20
z0
r0
```

Diagram of the modified oscillator with connections and variables.
Modified Oscillator 3

\[ x^2 + (y - c)^2 = r^2 \]

Diagram showing the oscillator with variables and parameters.

Graphs showing the behavior of the oscillator with different values of \( r \).
Constant–Influx Oscillator

As in the Shishi Odoshi (and the cell cycle)
Constant influx

Still working fine with the replaced switch.
The Novak–Tyson Oscillator

• First switch
  o Is the ‘transformed’ AM switch in one-input configuration (driven by constant influx of cyclin).

• Second switch
  o Is a simple two-stage switch working as a delay (the first switch is so good in terms of hysteresis that the second switch is not very critical for oscillation).

• Connection
  o The feedback from second to first switch is a bit complex, since both x and y are repressed by degrading cyclin. And there are more details still.
One of Ferrell’s Oscillators

- Second switch
  - Replaced by a one-stage switch. The oscillation still works, but is it harder to obtain (parameter tuning).
Conclusions
Conclusions

• A vast literature on cell cycle switching
  o Ferrell et.al., Novak–Tyson et.al., etc. Mostly ODE based analysis, plus noise
  o Many bistable transitions have different implementations in different cell cycle phases and organisms (phosphorylation, enzymes, synthesis/degradation, etc.)
  o We focused on a mechanism that can only be seen stochastically (quick majority switching with \(x=y\))

• A range of ‘network transformation’
  o Can explain the structure of some natural networks
  o From some non-trivial underlying algorithms
  o Discovering the transformation can elucidate the structure and function of the networks
  o But how can we say that these transformations ‘preserve (essential) behavior’?
Acknowledgements

• David Soloveichik
• Attila Csikasz-Nagy