On Switches and Oscillators
Program Equivalence in Biology?

Luca Cardelli
Microsoft Research

Turin 2011-05-05
http://lucacardelli.name
Outline

• Some questions that nature has (apparently) 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 that do not dampen or diverge
  o Building robust oscillators that resist parameter variations
Outline

• Subject to ‘chemical constraints’
  o Not all reactions can be easily implemented
  o Not all molecules can perform all functions

• The ‘logical’ solutions
  o Need to be adapted due to chemical constraints
  o Can we then still recognize them (if they exist)?
Switches
The Cell Cycle Switch

Numerical analysis of a comprehensive model of M-phase control in Xenopus oocyte extracts and intact embryos

Bela Novak* and John J. Tyson†
Department of Biology, Virginia Polytechnic Institute and State University, Blacksburg, Virginia 24060-0406, USA
*Permanent address: Department of Agricultural Chemical Technology, Technical University of Budapest, 1521 Budapest Gelett Ter 4, Hungary
†Author for correspondence

(dimers is left off the diagram to keep it simple.) (B) Positive feedback loops. Active MPF stimulates its own production from tyrosine-phosphorylated dimers by activating Cdc25 and inhibiting Wee1. We suspect that these signals are indirect, but intermediary enzymes are unknown and we ignore them in this paper. The signals from active MPF to Wee1 and Cdc25 generate an autocatalytic instability in the control system. We indicate also an ‘external’ signal from unreplicated DNA to Wee1 and Cdc25, which can be used to control the efficacy of the positive feedback loops. The letters a, b, e and f are used to label the rate constants for these reactions in Fig. 2. (C) Negative feedback loop. Active
Direct Competition

- $x$ catalyzes the transformation of $y$ into $x$
- $y$ catalyzes the transformation of $x$ into $y$

This system is bistable, but
- Convergence to a stable state is slow (a random walk).
- *Any* perturbation of a stable state can initiate a random walk to the other stable state.
- With 100 molecules of $x$ and $y$, convergence is quick, but with 10000 molecules, even at the same concentration (adjusting the rate) you will wait for a long time.
Approximate 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 and change state.
  o The whole population must eventually agree on a majority value (x or y) with probability 1.
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

[N.B. this holds even if the x,y populations are initially of equal size!]

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

[Angluin et al.]
vs. Stochastic Chemistry

- Chemical systems as distributed systems
  - What can they compute?
  - What can we say about their dynamics?
- Replace “agent” with “molecule” $\Rightarrow$ Chemical Master Equation (modulo details)
- Objection: in real life, some pairs of agents are more likely to interact than others
  - Agents in the same state are interchangeable
  - In a well-stirred chemical mixture, reaction types occur with the right probabilities (Gillespie, Physica A 1992)

[Angluin et al.]
Chemical Implementation

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

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 $x=y$ (CRN)

Gillespie simulation of the chemical reactions in SPiM.

- 10x more molecules at same concentration (i.e. lower rate) converge in ‘comparable’ time.
- Not shown: 10x more molecules in same volume, (i.e. higher concentration) converge 10x faster.

**N.B. a deterministic (ODE) simulation with $x=y$ would not converge at all!**

20k molecules, rate 10.0/s

200k molecules, rate 1.0/s (same concentration)

2000k molecules, rate 0.1/s (same concentration)
Bistable Element

- 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: it’s a switch.

A + B → B + C
B + A → A + C
C + A → A + A
C + B → B + B
As a Flip-Flop

Outputs \(x,y\):
\[
\begin{align*}
x + y &\rightarrow y + b \\
y + x &\rightarrow x + b \\
b + x &\rightarrow x + x \\
b + y &\rightarrow y + y
\end{align*}
\]

Inputs \(X,Y\):
\[
\begin{align*}
x + Y &\rightarrow Y + b \\
b + Y &\rightarrow Y + y \\
y + X &\rightarrow X + b \\
b + X &\rightarrow X + y
\end{align*}
\]

Can be switched by external (catalytic) signals that are only 20% of the \(x,y\) levels.

Init: 1000 \(x\) \(Y\) growing from 0 \((t=0)\) to 200 \((t=200)\) then back to 0 \((t=400)\) \(X\) growing from 0 \((t=400)\) to 200 \((t=600)\) then back to 0 \((t=800)\).
Chemical Constraints

- This circuit is ‘chemically demanding’
  - It requires $x$ molecules to be ‘next’ to $y$ molecules because they interact directly
  - It requires both $x$ and $y$ to be catalysts, and in fact autocatalysts, and in fact each–other’s autocatalyst!
Program 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.
Program Transformations

• Another example of relaxing constraints
  o Invent an Approximate Majority network that requires only $x$ to be a catalyst. How?

  o Enter the **Cell Cycle** switches…
Some Notation

- Catalytic reaction
  \[ x + z \rightarrow z + y \]

- Non-linear reaction
  \[ x \rightarrow y \]

The hollow circle may represent any ‘non-linear’ reaction (e.g. enzymatic, hill) in addition to this ‘double-phosphorylation’ network, which however is the standard interpretation here.
Zero–Input Switches

• ‘Zero–input switch’ = majority circuit: just working off the initial conditions.

• 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.
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.

• So that y no longer catalyzes anything.
Zero–Input Switches

• Still an AM circuit

(Although the equal-likelihood outcome here is around 4500 y vs 5500 x, and there are other paremeters)
The Cell Cycle Switch

Diagram showing the cell cycle switch with various steps and regulatory factors.
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.
One–Input Switches

• Hysteresis in AM–like switches
One-Input Switches

```
directive sample 0.02 1000
directive plot x(); y(); sx(); sy(); (* b(); *)
val rt = 10.0
val rx = 1.0
new xcat@rx:chan
new ycat@rt:chan
new sxcat@rt:chan new sxkill:chan
new syca@rt:chan new sykill:chan

let x() =
do !xcat; x() or ?ycat; b() or ?sycat; 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 sy0

let clock(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(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 < 5000 then sx() else if n < 10000 then !sxkill;() else ()
run clock(schedule,0.000001)
```

```
rx=5.0
sy=1000

rx=0.1
sy=1000

rx=0.1
sy=100
```
One–Input Switches

• Hysteresis in cell cycle switches

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
in 0
twice $s$, half $t$, and half $sy$

five times $t$
Two–input Switches

• (not really relevant here)
Oscillators
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).

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)

http://www.youtube.com/watch?v=VbvecTIfEc&NR=1&feature=fvwp
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 vs 10.0 top, and 2 vs 10, bottom.

(Simple limit-cycle oscillators in the literature have very critical rate ranges.)
Influx Oscillators

• Similar but:
  - The two-input switches are replaced by one-input switches which are reset by constant influxes.

\[
\begin{align*}
\text{r} = s &= 10, c_1 g = c_2 g = 3000 \\
\text{r} = s &= 10, c_1 g = c_2 g = 4000
\end{align*}
\]

Works best with \( s = r \).

Needs constant influx of \( c_1, c_2 \).
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).
  o It can be replaced by a one-stage switch (Ferrell’s cell cycle oscillator) but oscillation is a bit harder to obtain.

• Connection
  o Single links, as in the influx oscillator.
Novak–Tyson Oscillator

\[ w = z = 0 \]

\[ \frac{t}{2} \]

\[ p = r = 0, \text{ or } t/4, \text{ or } t*2: \text{ no oscillation} \]
Without double-positive loop

(then, block t and change amount of r)

- **Normal**
- t=0, r/2
- t=0, r/4
- t=0, r/5
- t=0, r/10
- t=0, r/12
- t=0, r/15
- t=0, r/20
Without double-negative loop

(then, with x no longer acting on z, change amount of z)
\[
\frac{(z+w)}{10},
\frac{t}{10}, \frac{r}{10},
\frac{(p+r)}{10},
\frac{t}{14}, \frac{(p+r)}{14}
\]
Conclusions
Conclusions

• A range of ‘network transformation’
  o Can explain the structure of some natural network
  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’?