Artificial Biochemistry

Biological Systems as Reactive Systems

Luca Cardelli

Microsoft Research

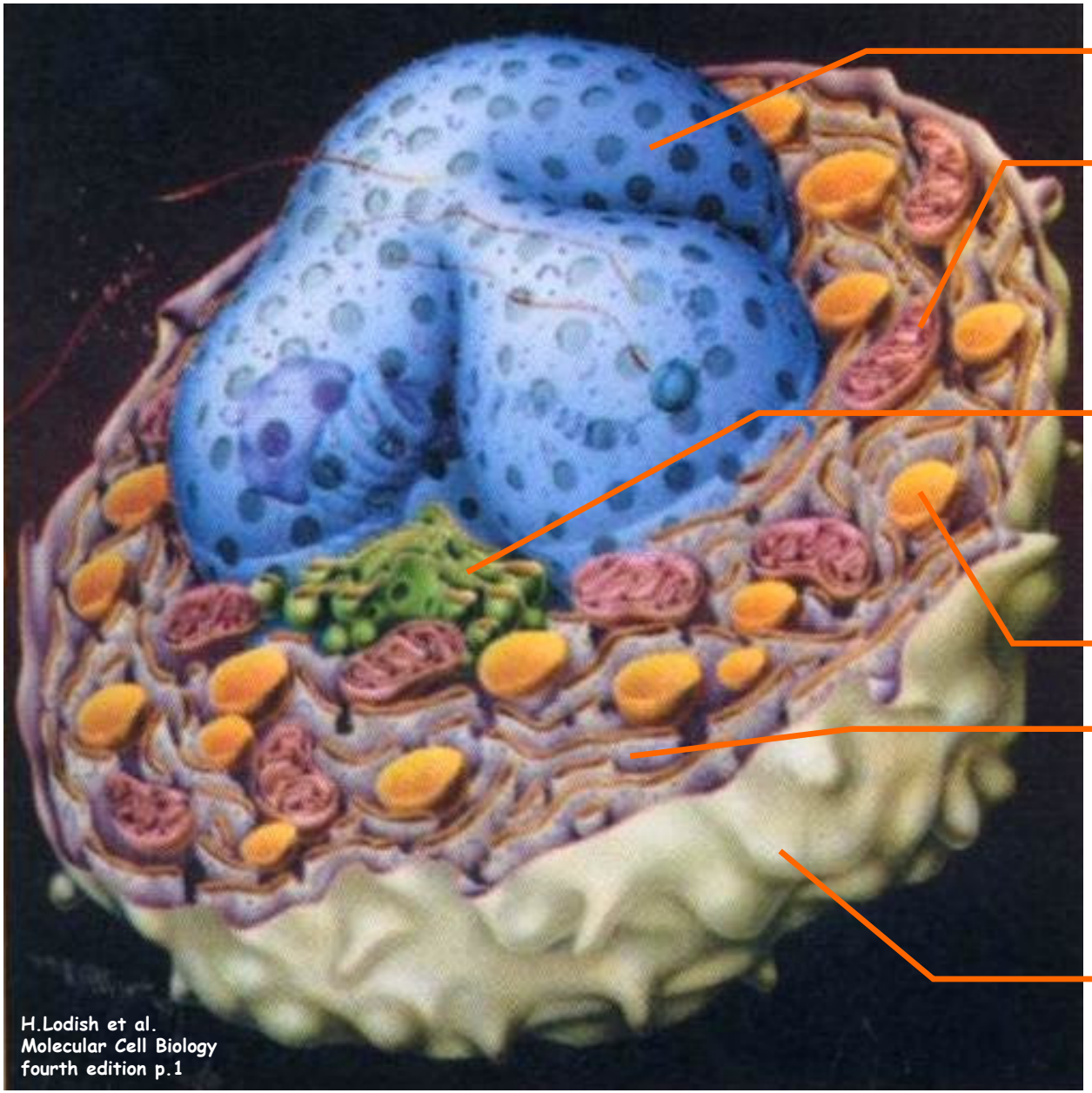
Computability in Europe, Swansea
2006-07-05

Structural Architecture

Eukaryotic Cell

(10~100 trillion in human body)

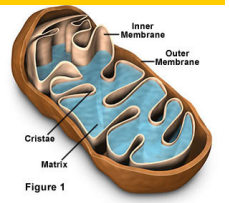
Membranes everywhere



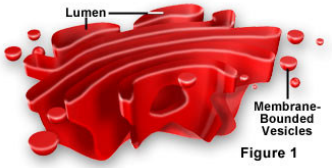
H.Lodish et al.
Molecular Cell Biology
fourth edition p.1

Nuclear membrane

Mitochondria

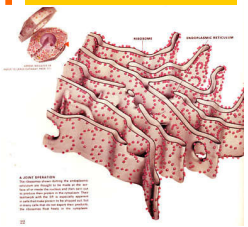


Golgi



Vesicles

E.R.



Plasma membrane (<10% of all membranes)

dna inside

© Luca Cardelli

Stochastic Collectives

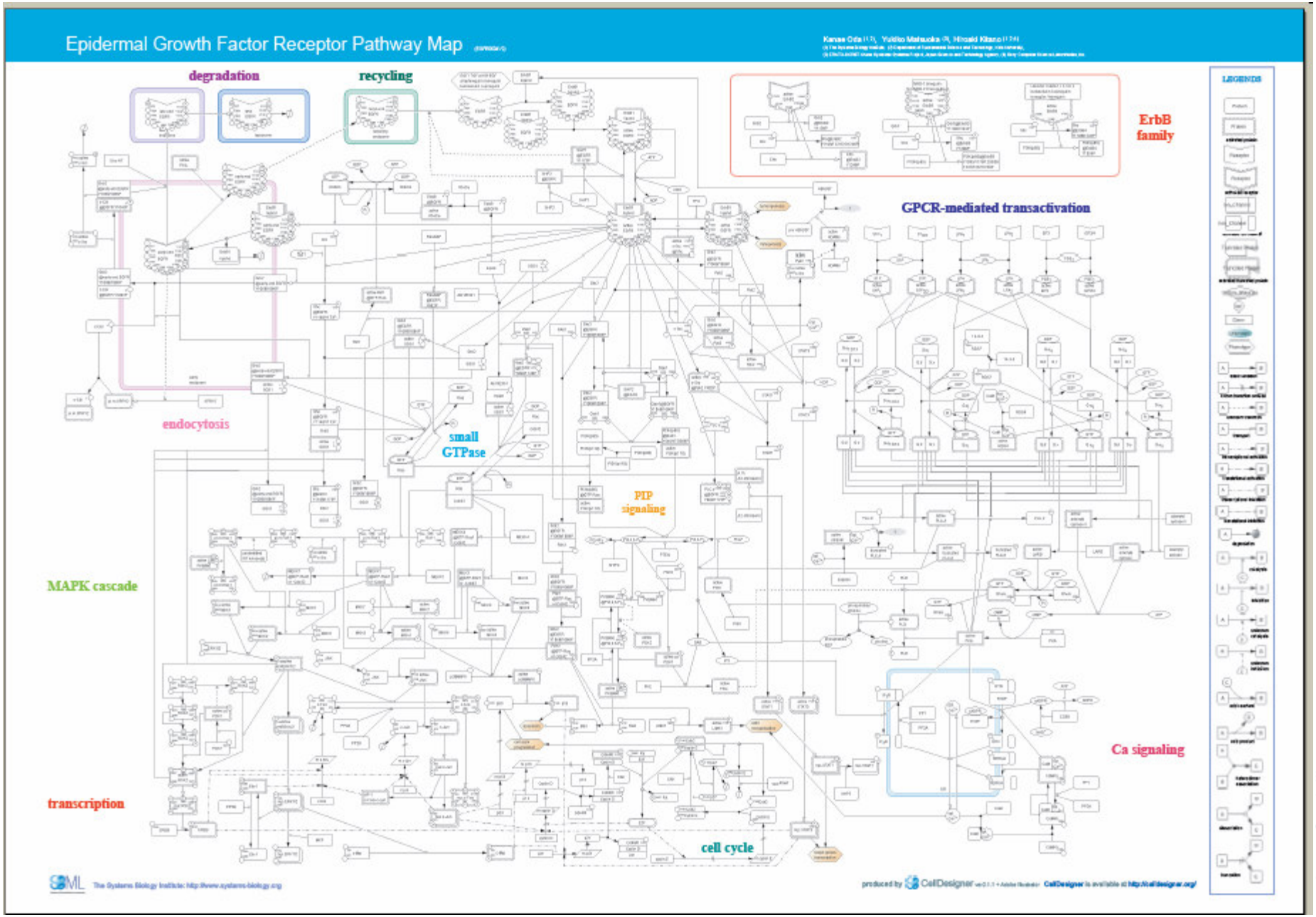
Stochastic Collectives

- "Collective":
  - A large set of interacting finite state automata:
    - Not quite language automata ("large set")
    - Not quite cellular automata ("interacting" but not on a grid)
    - Not quite process algebra ("finite state" and "collective")
    - Cf. "multi-agent systems" and "swarm intelligence"

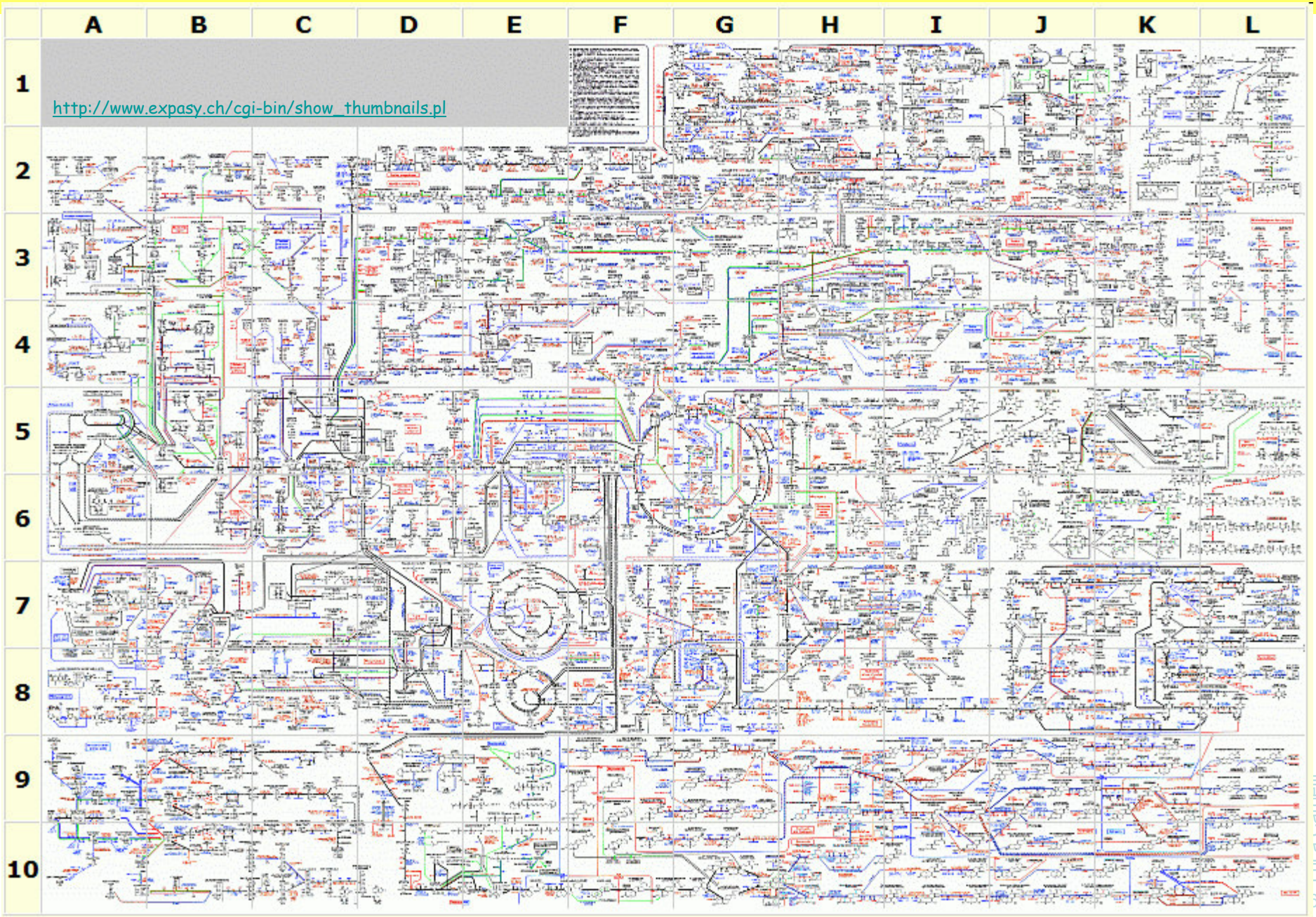
- "Stochastic":
  - Interactions have *rates*
    - Not quite discrete (hundreds or thousands of components)
    - Not quite continuous (non-trivial stochastic effects)
    - Not quite hybrid (no "switching" between regimes)

- Very much like biochemistry
  - Which is a large set of stochastically interacting molecules/proteins
  - Are proteins finite state and subject to automata-like transitions?
    - Let's say they are, at least because:
    - Much of the knowledge being accumulated in Systems Biology is described as state transition diagrams [Kitano].

State Transitions

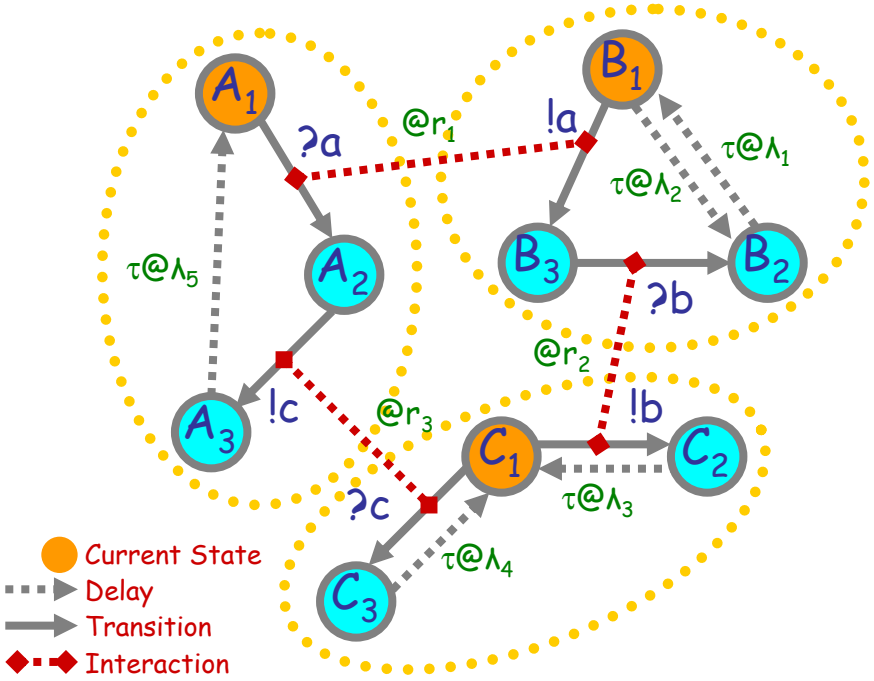


Even More State Transitions



© Luca Cardelli

Interacting Automata



Communicating automata: a graphical FSA-like notation for "finite state restriction-free π-calculus processes". *Interacting* automata do not even exchange values on communication.

The stochastic version has *rates* on communications, and delays.

$$\begin{array}{l}
\text{new } a@r_1 \\
\text{new } b@r_2 \\
\text{new } c@r_3
\end{array}$$ } Communication channels

$$\begin{array}{l}
A_1 = ?a;\ A_2 \\
A_2 = !c;\ A_3 \\
A_3 = \tau@\lambda_5;\ A_1
\end{array}$$

$$\begin{array}{l}
B_1 = \tau@\lambda_2;\ B_2 + !a;\ B_3 \\
B_2 = \tau@\lambda_1;\ B_1 \\
B_3 = ?b;\ B_2
\end{array}$$

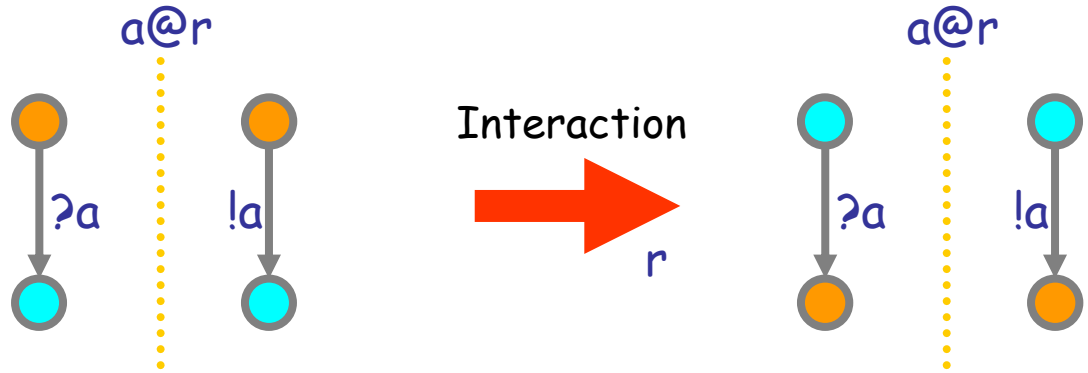
$$\begin{array}{l}
C_1 = !b;\ C_2 + ?c;\ C_3 \\
C_2 = \tau@\lambda_3;\ C_1 \\
C_3 = \tau@\lambda_4;\ C_2
\end{array}$$ } Automata

$$A_1 \mid B_1 \mid C_1$$ } The system and initial state

"Finite state" means: no composition or restriction inside recursion.

Analyzable by standard Markovian techniques, by first computing the "product automaton" to obtain the underlying finite Markov transition system. [Buchholz]

Interacting Automata Transition Rules

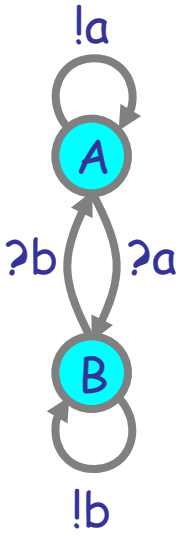


Q: What kind of mass behavior can this produce?

(We need to understand that if want to understand biochemical systems.)

∞ Luca Cardelli

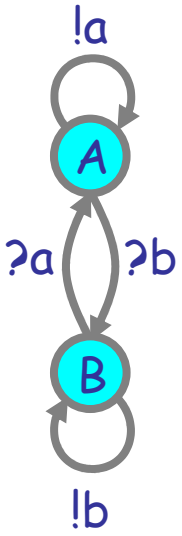
Groupies and Celebrities



Celebrity

(does not want to be like somebody else)

```
directive sample 0.1 200
directive plot A(); B()

new a@1.0:chan()
new b@1.0:chan()

let A() = do !a; A() or ?a; B()
and B() = do !b; B() or ?b; A()

run 100 of (A() | B())
```

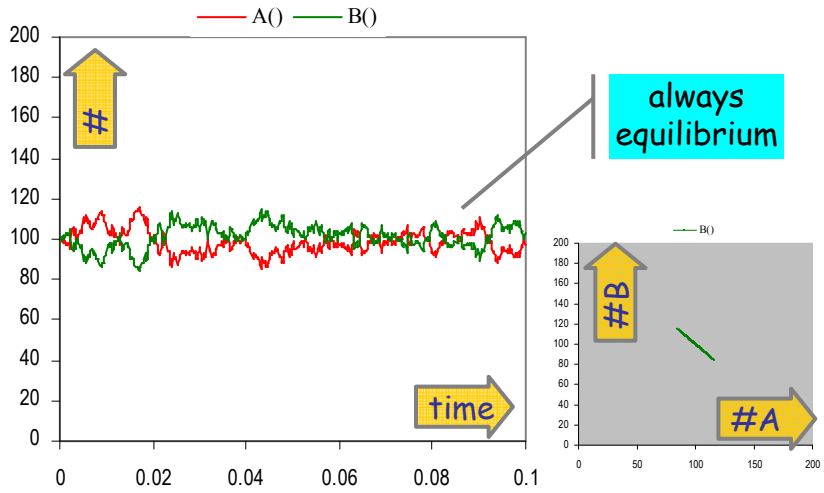


Groupie

(wants to be like somebody different)

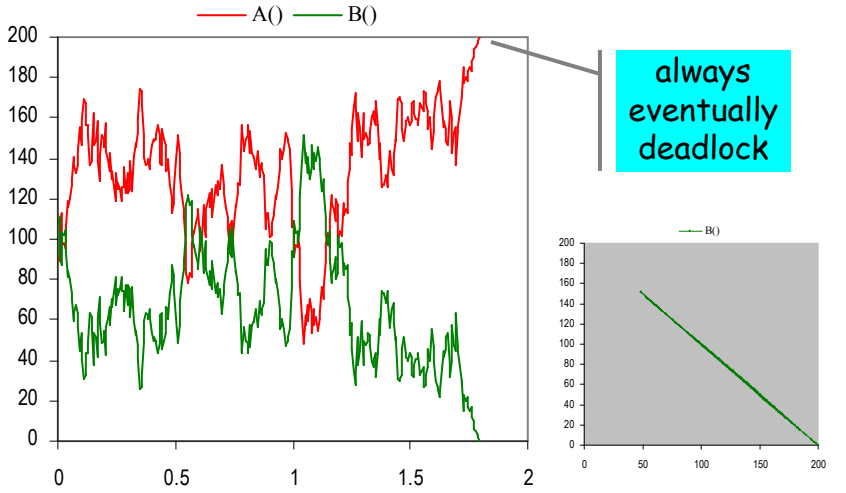
```
directive sample 0.1 200
directive plot A(); B()

new a@1.0:chan()
new b@1.0:chan()

let A() = do !a; A() or ?b; B()
and B() = do !b; B() or ?a; A()

run 100 of (A() | B())
```

A stochastic collective of celebrities:



Stable because as soon as a A finds itself in the majority, it is more likely to find somebody in the same state, and hence change, so the majority is weakened.

A stochastic collective of groupies:



Unstable because within an A majority, an A has difficulty finding a B to emulate, but the few B's have plenty of A's to emulate, so the majority may switch to B. Leads to deadlock when everybody is in the same state and there is nobody different to emulate.

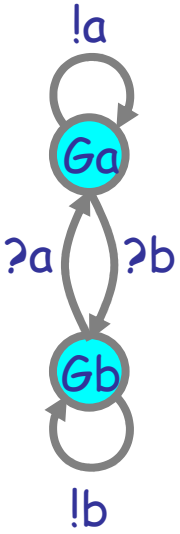
© Luca Cardelli

A tiny bit of "noise" can make a huge difference

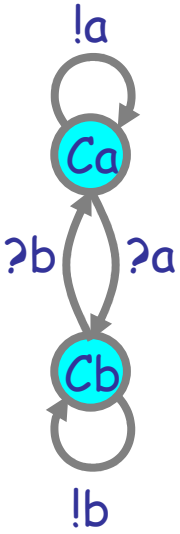
Both Together

A way to break the deadlocks: Groupies with just a few Celebrities

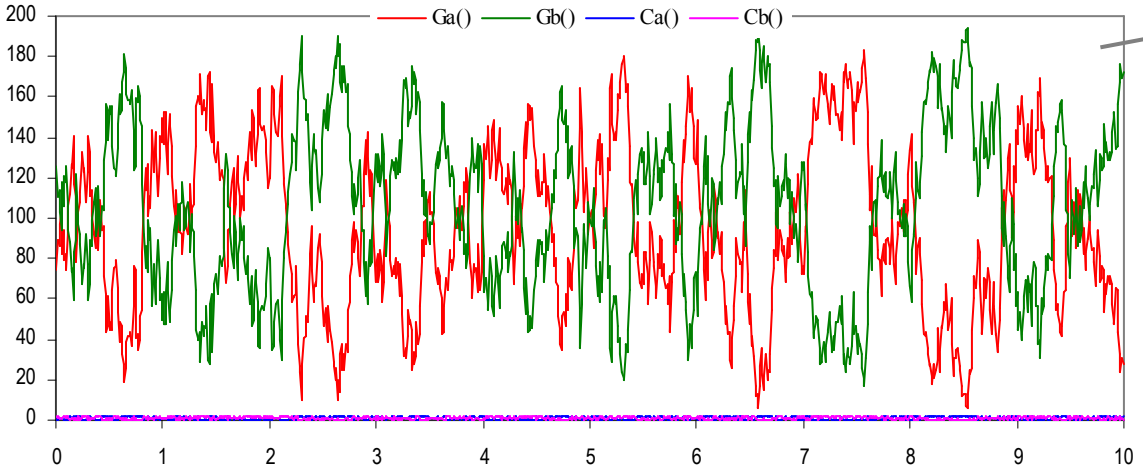
Many Groupies



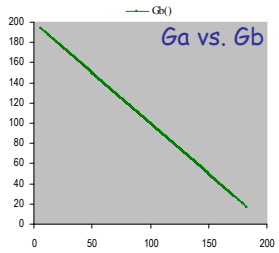
A few (2) Celebrities




```
directive sample 10.0 1000
directive plot Ga(); Gb(); Ca(); Cb()

new a@1.0:chan()
new b@1.0:chan()

let Ca() = do !a; Ca() or ?a; Cb()
and Cb() = do !b; Cb() or ?b; Ca()

let Ga() = do !a; Ga() or ?b; Gb()
and Gb() = do !b; Gb() or ?a; Ga()

run    1 of (Ca() | Cb())
run 100 of (Ga() | Gb())
```



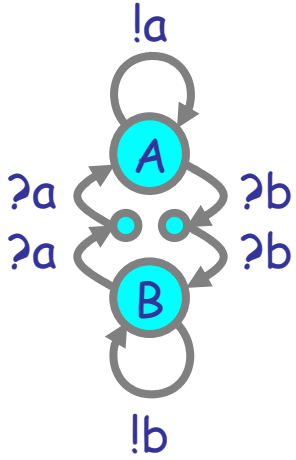
never deadlock

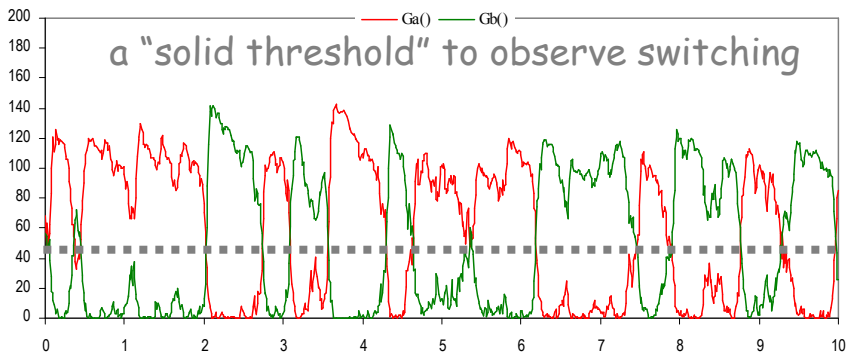


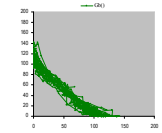
Regularity can arise not far from chaos

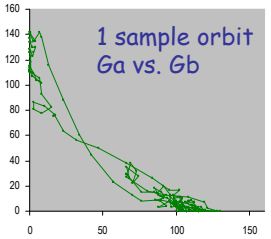
Hysteric Groupies

We can get more regular behavior from groupies if they "need more convincing", or "hysteresis" (history-dependence), to switch states.

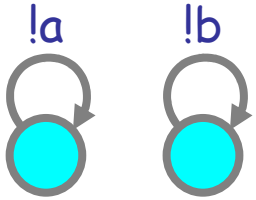








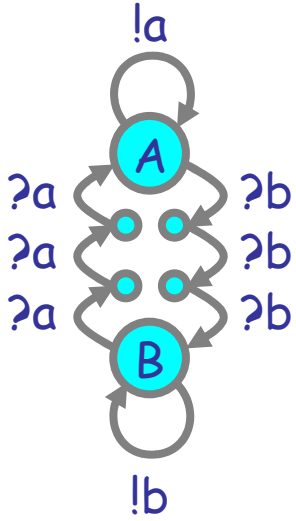

```
directive sample 10.0 1000
directive plot Ga(); Gb()

new a@1.0:chan()
new b@1.0:chan()

let Ga() = do !a; Ga() or ?b; ?b; Gb()
and Gb() = do !b; Gb() or ?a; ?a; Ga()

let Da() = !a; Da()
and Db() = !b; Db()

run 100 of (Ga() | Gb())
run   1 of (Da() | Db())
```

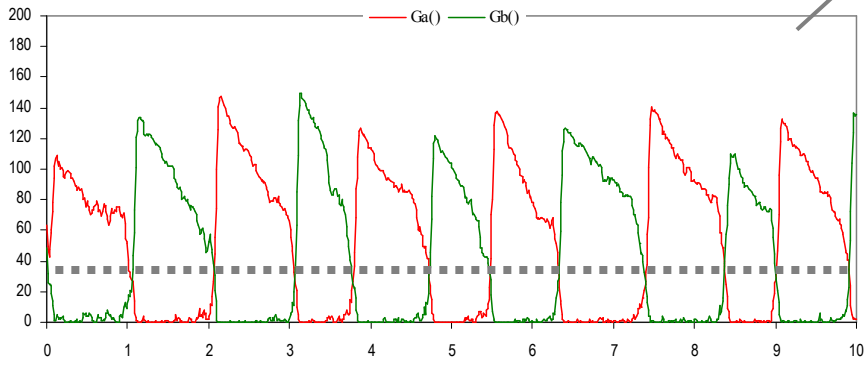


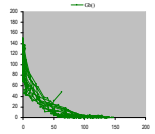
(With doping to break deadlocks)

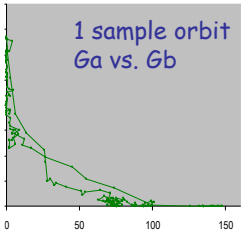
N.B.: It will not oscillate without doping (noise)

"regular" oscillation



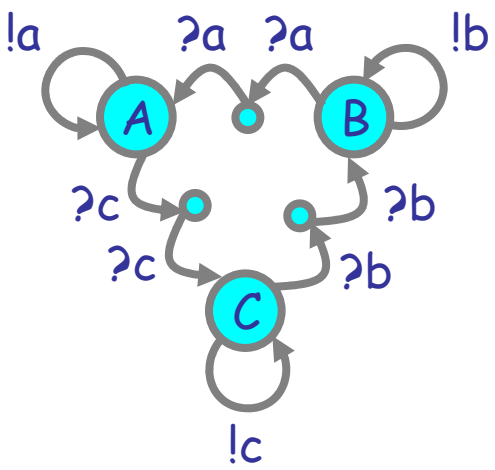


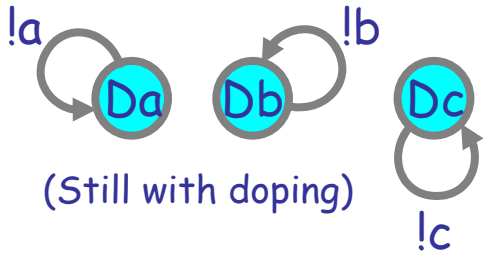





```
directive sample 10.0 1000
directive plot Ga(); Gb()

new a@1.0:chan()
new b@1.0:chan()

let Ga() = do !a; Ga() or ?b; ?b; ?b; Gb()
and Gb() = do !b; Gb() or ?a; ?a; ?a; Ga()

let Da() = !a; Da()
and Db() = !b; Db()

run 100 of (Ga() | Gb())
run   1 of (Da() | Db())
```

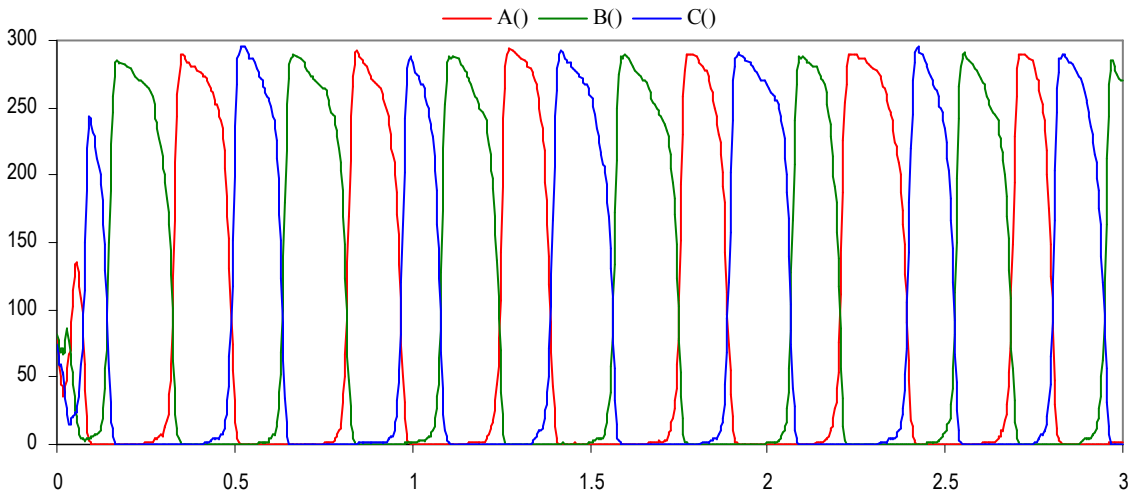
Hysteric 3-Way Groupies

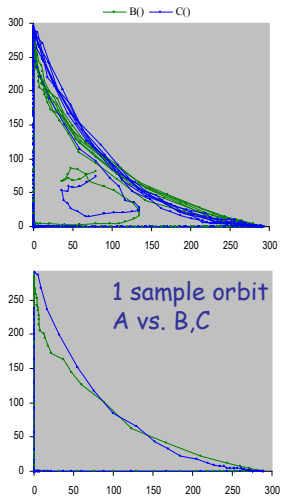




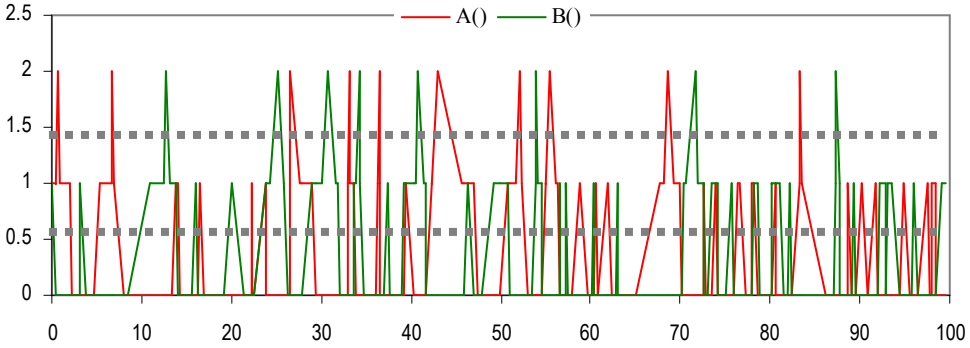

```
directive sample 3.0 1000
directive plot A(); B(); C()

new a@1.0:chan()
new b@1.0:chan()
new c@1.0:chan()

let A() = do !a; A() or ?c; ?c; C()
and B() = do !b; B() or ?a; ?a; A()
and C() = do !c; C() or ?b; ?b; B()

let Da() = !a; Da()
and Db() = !b; Db()
and Dc() = !c; Dc()

run 100 of (A() | B() | C())
run 1 of (Da() | Db() | Dc())
```

N.B.: It will not oscillate without doping (noise)



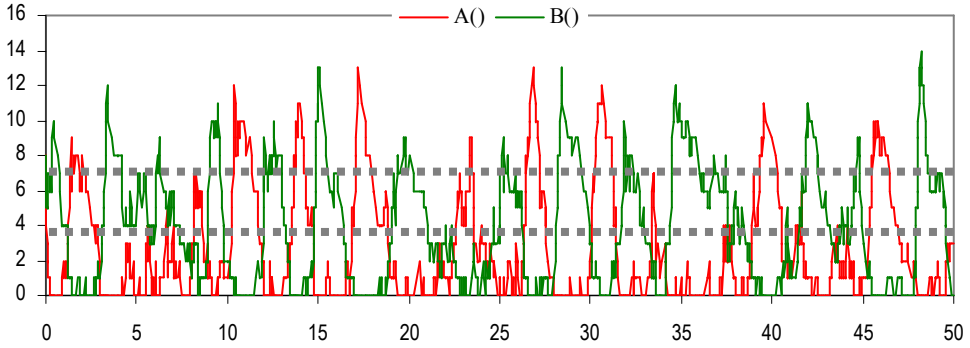


Oscillation as Emergence



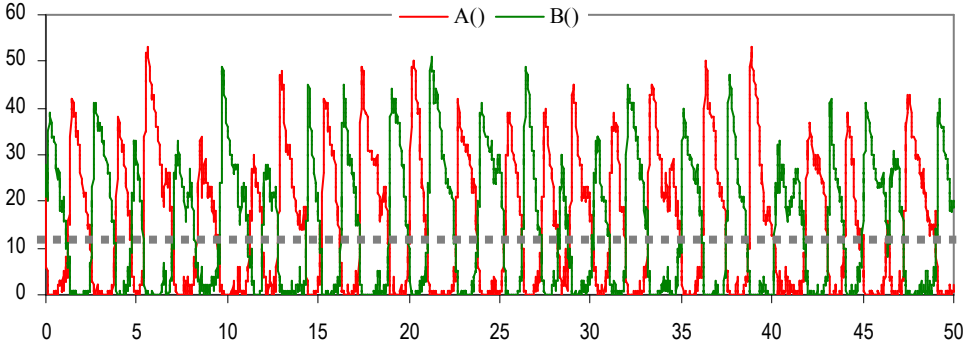
Just 2 of the hysteric groupies do not oscillate regularly at all!

Without changing the components, interesting properties emerge with a critical size of the population.



Nor 16…

Dotted lines indicate cross sections where one may look for evidence of alternation.



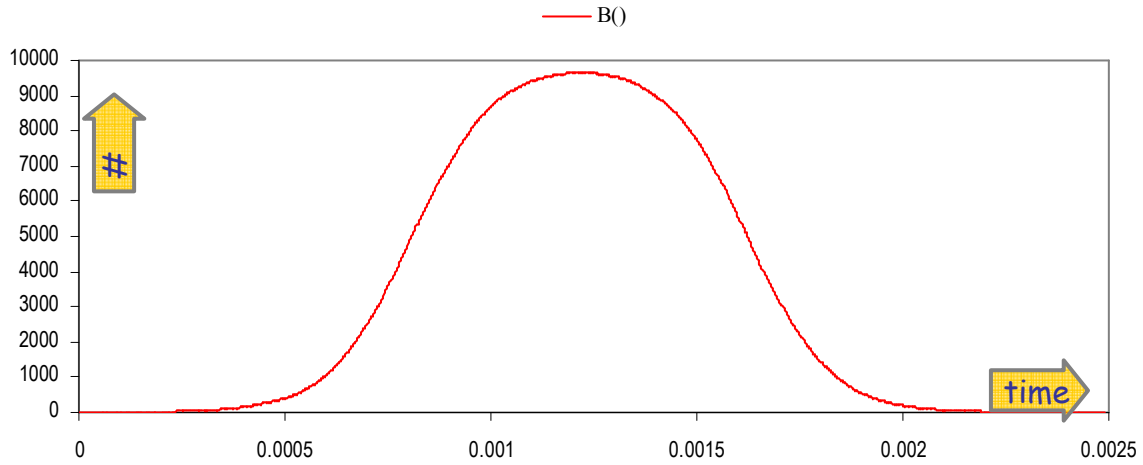
Pretty good with 64…

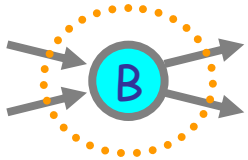
```
new a@1.0:chan()
new b@1.0:chan()

let A() = do !a; A() or ?b; ?b; ?b; B()
and B() = do !b; B() or ?a; ?a; ?a; A()

let As() = !a; As()
and Bs() = !b; Bs()

run 64 of (A() | B())
run  1 of (As() | Bs())
```

Distributions can be Programmed

Exercise (hard):

Build a *small* automaton where one state has an occupation distribution like this:





Or, more specifically, build a 3-state, A-B-C, automaton such that:

$$[B]^{\bullet} = [B]([A]-[C])$$

Semantics of Collective Behavior

"Micromodels": Continuous Time Markov Chains

- The underlying semantics of stochastic π-calculus (and stochastic interacting automata). Well established in many ways.
  - Automata with rates on transitions.
- "The" correct semantics for chemistry, executable.
  - Gillespie stochastic simulation algorithm
- Lots of advantages
  - Compositional, compact, mechanistic, etc.
- But do not give a good sense of "collective" properties.
  - Yes one can do simulation.
  - Yes one can do program analysis.
  - Yes one can do modelchecking.
  - But somewhat lacking in "analytical properties" and "predictive power".

"Macromodels": Ordinary Differential Equations

- They always ask:
  - "Yes, but how does you automata model relate to the 75 ODE models in the literature?"

- Going from processes/automata to ODEs directly:
  - *In principle*: just write down the Rate Equation: [Calder, Hillston]
    - Determine the set of all possible *states* S of each process.
    - Determine the rates of the transitions between such states.
    - Let [S] be the "number of processes in state S" as a function of time.
    - Define for each state S:

      $[S]^{\bullet}$ = (rate of change of the number of processes in state S)

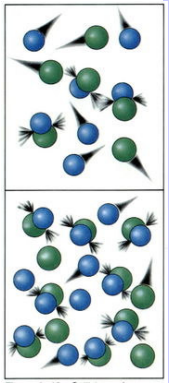
      Cumulative rate of transitions from any state S' to state S, times [S'],
      minus cumulative rate of transitions from S to any state S", times [S].
  - Intuitive (rate = inflow minus outflow), but often clumsy to write down precisely.

- But why go to the trouble?
  - If we first convert processes to chemical reactions, then we can convert to ODEs by standard means!

Macromodel of Interaction

Law of Mass Interaction

The speed of interaction† is proportional to the number of *possible interactions*.



Chemical Law of Mass Action

http://en.wikipedia.org/wiki/Chemical_kinetics

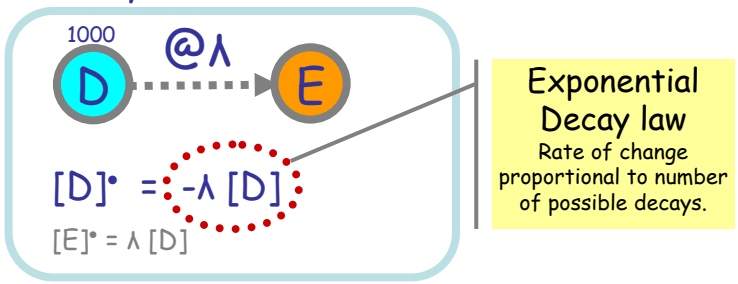
The **speed** of a chemical reaction is proportional to the **activity** of the reacting substances.

Activity = concentration, for well-stirred aqueous medium

Concentration = number of moles per liter of solution

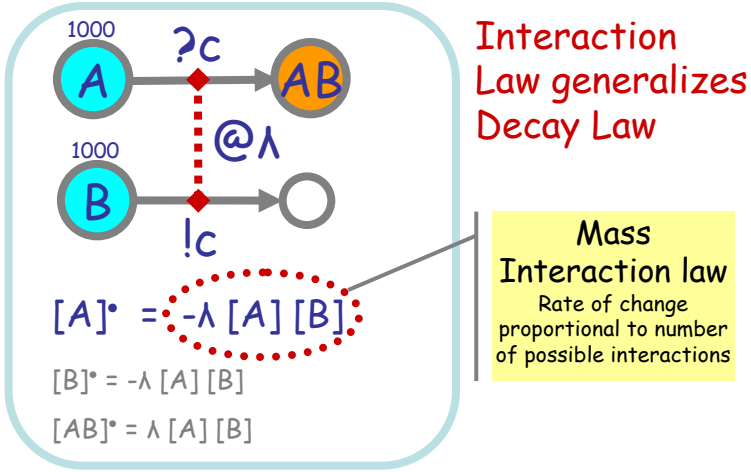
Mole = 6.022141×10^{23} particles

Decay



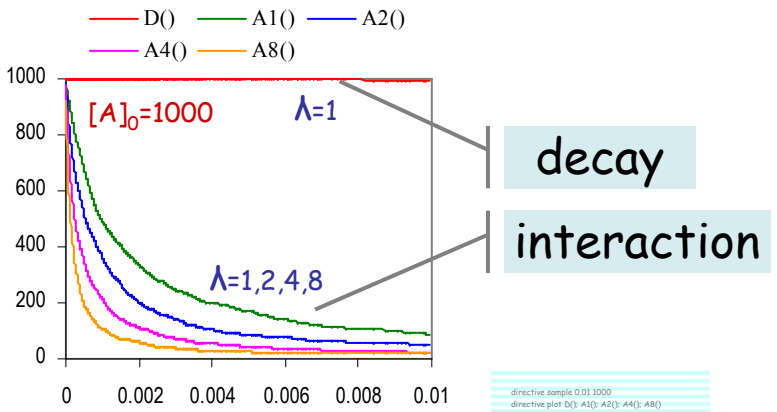
Exponential Decay law
Rate of change proportional to number of possible decays.

Mass interaction



Interaction Law generalizes Decay Law

Mass Interaction law
Rate of change proportional to number of possible interactions




```
directive sample 0.01 1000
directive plot D(); A1(); A2(); A4(); A8()

new c1@1.0: chan()  new c2@2.0: chan()
new c4@4.0: chan()  new c8@8.0: chan()

let D() = delay@1.0
let A1() = ?c1 and B1() = !c1
let A2() = ?c2 and B2() = !c2
let A4() = ?c4 and B4() = !c4
let A8() = ?c8 and B8() = !c8

run 1000 of (D() | A1() | B1() | A2()
| B2() | A4() | B4() | A8() | B8())
```

† speed of interaction (formally definable)
= number of interactions over time
not proportional to the number of interacting processes!

[P] is the number of processes P (this is informal; it is only meaningful for a set of processes offering a given action, but a set of such processes can be counted and plotted)

From Chemistry to ODEs

Chemical Reactions

$A \rightarrow^r B_1 + \ldots + B_n$	Degradation	$[A]^\bullet = -r[A]$	Exponential Decay
$A_1 + A_2 \rightarrow^r B_1 + \ldots + B_n$	Asymmetric Collision	$[A_i]^\bullet = -r[A_1][A_2]$	Mass Action Law
$A + A \rightarrow^r B_1 + \ldots + B_n$	Symmetric Collision	$[A]^\bullet = -r[A]([A]-1)$	Mass Action Law

(assuming $A \neq B_i \neq A_j$ for all i,j)

No other reactions!

JOURNAL OF CHEMICAL PHYSICS VOLUME 113, NUMBER 1

The chemical Langevin equation

Daniel T. Gillespie[a)]
Research Department, Code 4T4100D, Naval Air Warfare Center, China Lake, California 93555

Genuinely *trimolecular* reactions do not physically occur in dilute fluids with any appreciable frequency. *Apparently* trimolecular reactions in a fluid are usually the combined result of two bimolecular reactions and one monomolecular reaction, and involve an additional short-lived species.

Chapter IV: Chemical Kinetics
[David A. Reckhow , CEE 572 Course]

... reactions may be either elementary or non-elementary. Elementary reactions are those reactions that occur exactly as they are written, without any intermediate steps. These reactions almost always involve just one or two reactants. ... Non-elementary reactions involve a series of two or more elementary reactions. Many complex environmental reactions are non-elementary. In general, reactions with an overall reaction order greater than two, or reactions with some non-integer reaction order are non-elementary.

THE COLLISION THEORY OF REACTION RATES
www.chemguide.co.uk

The chances of all this happening if your reaction needed a collision involving more than 2 particles are remote. All three (or more) particles would have to arrive at exactly the same point in space at the same time, with everything lined up exactly right, and having enough energy to react. That's not likely to happen very often!

Trimolecular reactions:

$A + B + C \rightarrow^r D$

the measured "r" is an (imperfect) aggregate of e.g.:

$A + B \leftrightarrow AB$

$AB + C \rightarrow D$

Enzymatic reactions:

$S \xrightarrow{E} {}^r P$

the "r" is given by Michaelis-Menten (approximated steady-state) laws:

$E + S \leftrightarrow ES$

$ES \rightarrow P + E$

From Reactions to ODEs

CAVEAT: A deterministic *approximation* of a stochastic system (i.e. possibly *misleading*)

v_1: $A+B \rightarrow_{k_1} C+C$

v_2: $A+C \rightarrow_{k_2} D$

v_3: $C \rightarrow_{k_3} E+F$

v_4: $F+F \rightarrow_{k_4} B$

Write the coefficients by columns

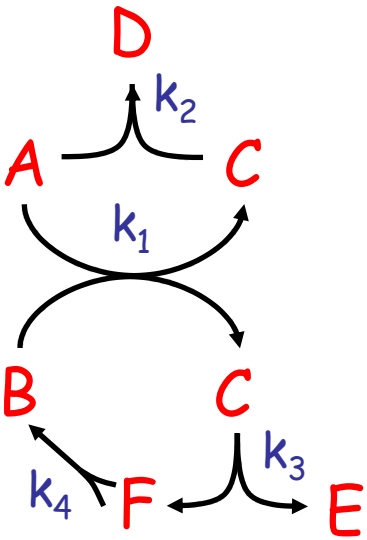
reactions

species

N	v_1	v_2	v_3	v_4
A	-1	-1		
B	-1			1
C	2	-1	-1	
D		1		
E			1	
F			1	-2

X

Stoichiometric Matrix



Quantity changes

Stoichiometric matrix

Rate laws

$$[X]^{\bullet} = N \cdot I$$

$[A]^{\bullet} = -I_1 - I_2$

$[B]^{\bullet} = -I_1 + I_4$

$[C]^{\bullet} = 2I_1 - I_2 - I_3$

$[D]^{\bullet} = I_2$

$[E]^{\bullet} = I_3$

$[F]^{\bullet} = I_3 - 2I_4$

Read the concentration changes from the rows

E.g. $[A]^{\bullet} = -k_1[A][B] - k_2[A][C]$

Set a rate law for each reaction (Degradation/Asymmetric/Symmetric)

I	
I_1	$k_1[A][B]$
I_2	$k_2[A][C]$
I_3	$k_3[C]$
I_4	$k_4[F]([F]-1)/2$

X: chemical species

[-]: quantity of molecules

I: rate laws

k: kinetic parameters

N: stoichiometric matrix

From Processes to Chemistry

Chemical Ground Form (CGF)

$E ::= X_1=M_1, \ldots, X_n=M_n$ Definitions $(n \geq 0)$

$M ::= \pi_1;P_1 \oplus \ldots \oplus \pi_n;P_n$ Molecules $(n \geq 0)$

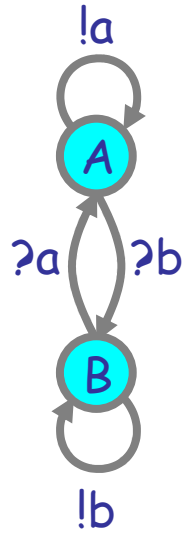
$P ::= X_1 \mid \ldots \mid X_n$ Solutions $(n \geq 0)$

$\pi ::= \tau_r \;\; ?n_{(r)} \;\; !n_{(r)}$ Interactions (delay, input, output)

$CGF ::= E,P$ Definitions with Initial Conditions

(To translate chemistry back to processes we need a bit more than simple automata: we may have "+" on the right of $\rightarrow$, that is we may need "|" after π.)

$\oplus$ is stochastic choice (vs. + for chemical reactions)
0 is the null solution ($P|0 = 0|P = P$)
and null molecule ($M \oplus 0 = 0 \oplus M = M$) ($\tau_0;P = 0$)
X_i are distinct in E
Each name n is assigned a fixed rate r: $n_{(r)}$



Ex: interacting automata
(which are CGFs using "|" only in initial conditions):

$A = !a;A \oplus ?b;B$ — Automaton in state A

$B = !b;B \oplus ?a;A$ — Automaton in state B

$A|A|B|B$ — Initial conditions: 2A and 2B

CGF Semantics

Reduction

$E, (X_1 \mid P) \rightarrow^r E, (P_1 \mid P)$ if $E \equiv X_1 = \tau_r; P_1 \oplus M_1, E'$

$E, (X_1 \mid X_2 \mid P) \rightarrow^r E, (P_1 \mid P_2 \mid P)$ if $E \equiv X_1 = ?n_{(r)}; P_1 \oplus M_1, E_1 \equiv X_2 = !n_{(r)}; P_2 \oplus M_2, E_2$

$E,P \rightarrow^r E'',P''$ if $E,P \equiv E',P_1 \wedge E',P_1 \rightarrow^r E',P_2 \wedge E',P_2 \equiv E'',P'' \Rightarrow$

Structural Congruence

$\equiv$ is an equivalence relation

$E, E' \equiv E', E$

$M \oplus M' \equiv M' \oplus M$

$P \mid P' \equiv P' \mid P$

$E \equiv E' \wedge P \equiv P' \Rightarrow E,P \equiv E',P'$

$E \equiv E' \wedge M \equiv M' \Rightarrow X = M, E \equiv X = M', E'$

$M \equiv M' \wedge P \equiv P' \Rightarrow p;P \oplus M \equiv p;P' \oplus M'$

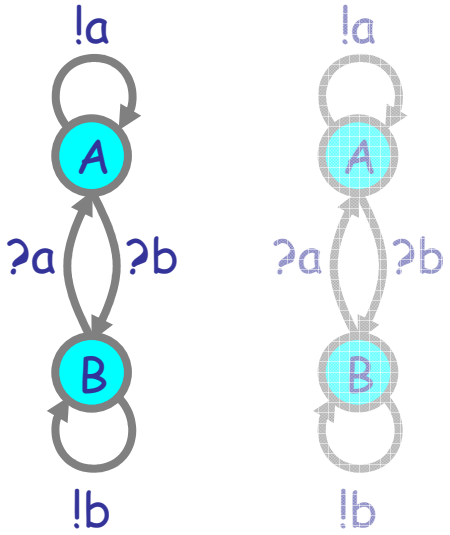
$P \equiv P' \Rightarrow X \mid P \equiv X \mid P'$

$E = (A = !a;A \oplus ?b;B$

$\quad B = !b;B \oplus ?a;A)$

$E, (A|B|B) \rightarrow^{r(a)} E, (A|A|B) \rightarrow^{r(b)} E, (A|B|B) \rightarrow^{r(b)} E, (B|B|B)$

Automata to Chemistry



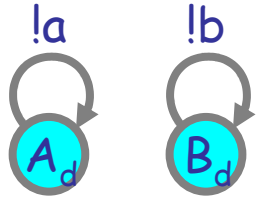
$A+B \rightarrow B+B$

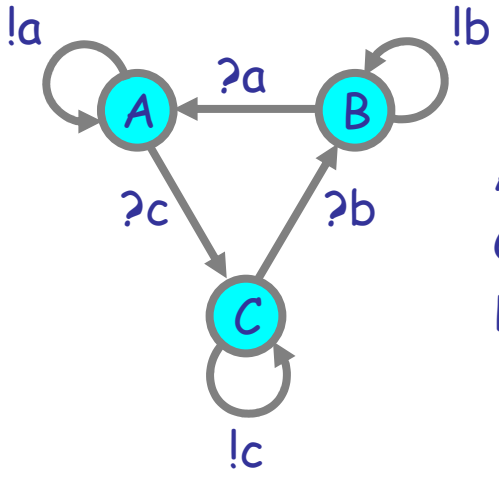
$B+A \rightarrow A+A$

$A+B_d \rightarrow B+B_d$

$B+A_d \rightarrow A+A_d$

Doping





$A+C \rightarrow C+C$

$C+B \rightarrow B+B$

$B+A \rightarrow A+A$

$A+C_d \rightarrow C+C_d$

$C+B_d \rightarrow B+B_d$

$B+A_d \rightarrow A+A_d$

Doping

Three Main Cases

Unary reactions. These are not *finite state systems*, but *finite species systems* are ok!

E:

$X = \tau_r;(X \mid X)$

C(E):

$X \rightarrow^{r} X + X$

Unbounded state, but only 1 species. No problem!

Binary reactions.

The same interaction can occur multiple times and must be taken into account:

E:

$A = ?n;B \oplus ?n;B$

$C = !n;D$

C(E):

$A + C \rightarrow^{\rho(n)} B + D$

$A + C \rightarrow^{\rho(n)} B + D$

That is:

$A + C \rightarrow^{2\rho(n)} B + D$

Symmetric reactions:

E:

$X = !a;0 \oplus ?a;Y$

C(E):

$X + X \rightarrow^{2\rho(a)} Y$

The rate of a was pre-halved and must be restored.

CGF to Chemistry

$E ::= X_1=M_1, \ldots, X_n=M_n$ Definitions $(n\geq 0)$

$M ::= \pi_1;P_1 \oplus \ldots \oplus \pi_n;P_n$ Molecules $(n\geq 0)$

$P ::= X_1 \mid \ldots \mid X_n$ Solutions $(n\geq 0)$

$\pi ::= \tau_r \;\; ?n_{(r)} \;\; !n_{(r)}$ Interactions (delay, input, output)

$CGF ::= E,P$ Definitions with Initial Conditions

Each X in E is seen as a separate *species*.

Chemical reactions for E: (N.B.: $\{\ldots\}^m$ is a multiset, and P is P with all the | changed to +)

$$Ch_G(E) := \{(X \to^r P) \text{ s.t. } (X \equiv \tau_r;P \oplus \ldots) \in E\}^m$$
$$\cup^m \{(X + Y \to^r P + Q) \text{ s.t. } X\neq Y, \langle (X \equiv ?n_{(r)};P \oplus \ldots),(Y \equiv !n_{(r)};Q \oplus \ldots)\rangle \in E^2\}^m$$
$$\cup^m \{(X + X \to^{2r} P + Q) \text{ s.t. } (X \equiv ?n_{(r)};P \oplus \ldots \equiv !n_{(r)};Q \oplus \ldots)\rangle \in E\}^m$$

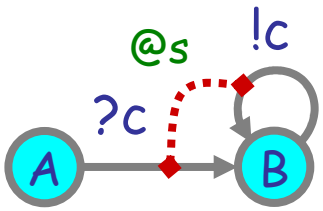
Initial conditions for P:

$$Ch_G(P) := P$$

From Processes to ODEs

Nonlinear Transitions

Basic Nonlinear Transition



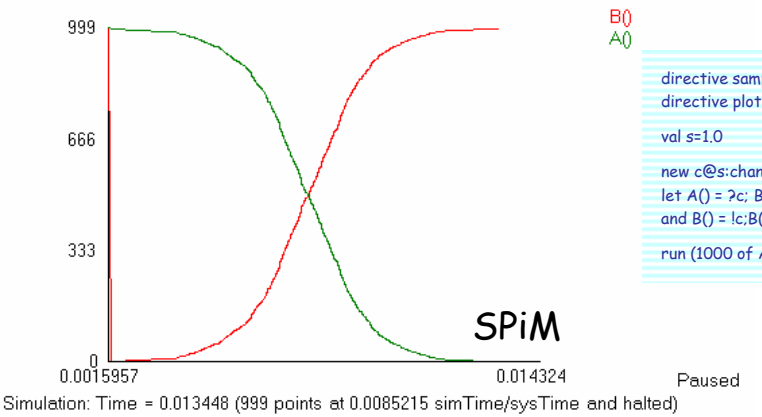
$A = ?c_{(s)};B$

$B = !c_{(s)};B$

$A+B \rightarrow^{s} B+B$

$[A]^{\bullet} = -s[A][B]$

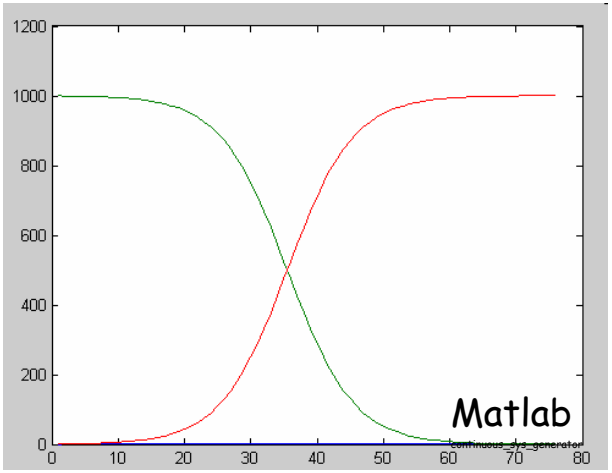
$[B]^{\bullet} = s[A][B]$



Simulation: Time = 0.013448 (999 points at 0.0085215 simTime/sysTime and halted)

```
directive sample 0.02 1000
directive plot B(); A()

val s=1.0

new c@s:chan
let A() = ?c; B()
and B() = !c;B()

run (1000 of A() | 1 of B())
```

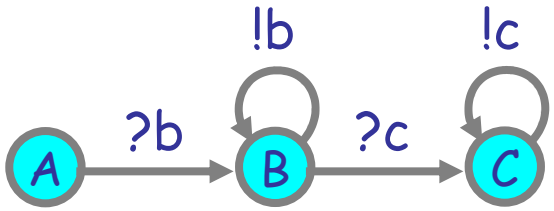
N.B.: needs at least 1 B to "get started".




```
interval/step [0:0.001:0.0]
(A)     dx1/dt = - x1*x2        1000.0
(B)     dx2/dt = x1*x2          1.0
```

Bell Exercise

Build a *small* network where one node has a distribution like B():



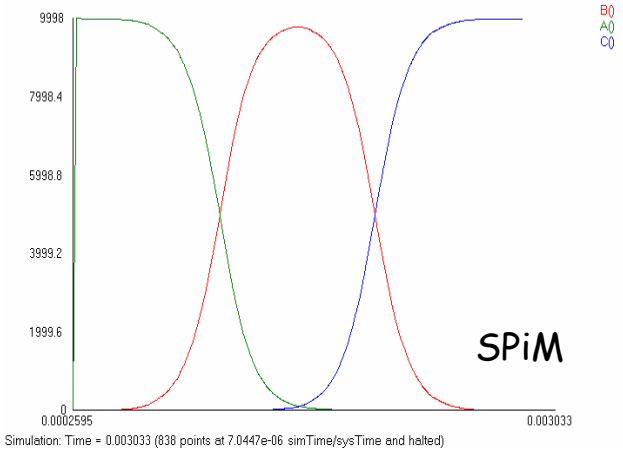
$[B]^\bullet = [B]([A]-[C])$

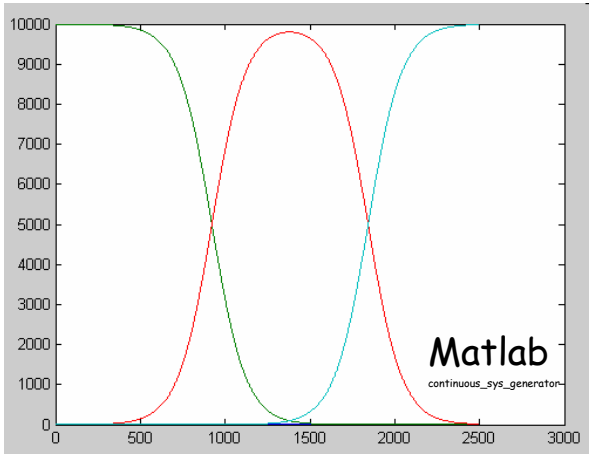
$A = ?b_{(1)};B$
$B = !b_{(1)};B \oplus ?c_{(1)};C$
$C = !c_{(1)};C$

$A+B \rightarrow^1 B+B$
$B+C \rightarrow^1 C+C$

$[A]^\bullet = -[A][B]$
$[B]^\bullet = [A][B]-[B][C]$
$[C]^\bullet = [B][C]$

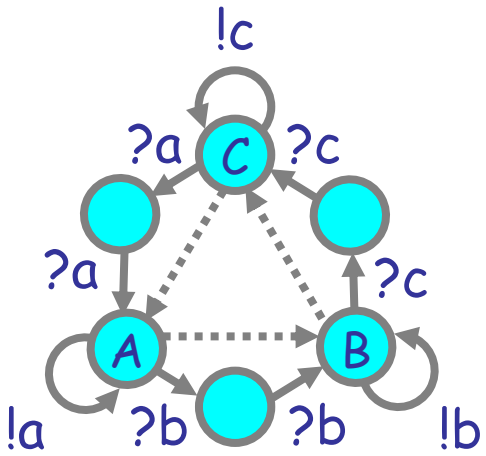
```
directive sample 0.0025 1000
directive plot B(); A(); C()

new b@1.0:chan new c@1.0:chan

let A() = ?b; B()
and B() = do !b;B() or ?c; C()
and C() = !c;C()

run ((10000 of A()) | B() | C())
```






```
interval/step [0:0.000001:0.0025]
(A)   dx1/dt = -x1*x2          10000.0
(B)   dx2/dt = x1*x2 - x2*x3   1.0
(C)   dx3/dt = x2*x3           1.0
```

Oscillator



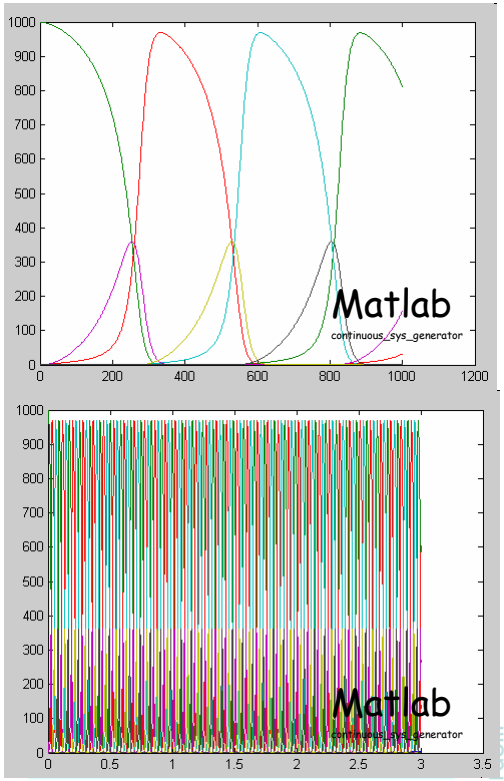

```
directive sample 0.1 1000
directive plot A1(); A2(); A3()

val r=1.0 val s=1.0

new a1@s:chan new a2@s:chan new a3@s:chan
let A1() = do !a1;A1() or delay@r;A2() or ?a2; ?a2; A2()
and A2() = do !a2;A2() or delay@r;A3() or ?a3; ?a3; A3()
and A3() = do !a3;A3() or delay@r;A1() or ?a1; ?a1; A1()

run 1000 of A1()
```

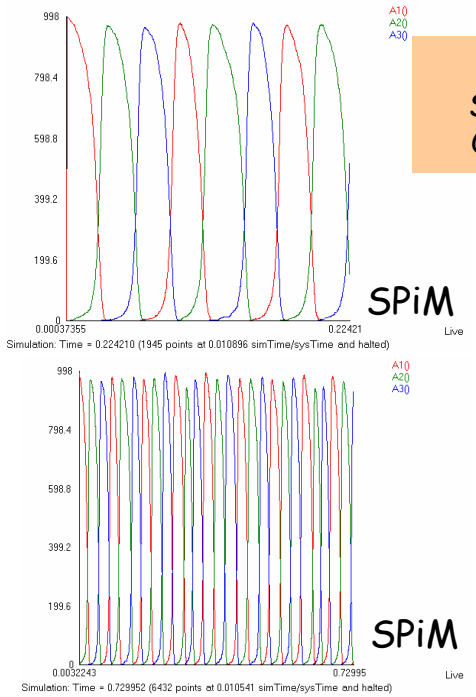
N.B. this does not deadlock!

$$A = !a_{(s)};A \oplus \tau_r;B \oplus ?b_{(s)};A'$$
$$A' = ?b_{(s)};B$$
$$B = !b_{(s)};B \oplus \tau_r;C \oplus ?c_{(s)};B'$$
$$B' = ?c_{(s)};C$$
$$C = !c_{(s)};C \oplus \tau_r;A \oplus ?a_{(s)};C'$$
$$C' = ?a_{(s)};A$$

Sustained Deterministic Oscillation

$$A \rightarrow^r B$$
$$A+B \rightarrow^s A'+B$$
$$A'+B \rightarrow^s B+B$$
$$B \rightarrow^r C$$
$$B+C \rightarrow^s B'+C$$
$$B'+C \rightarrow^s C+C$$
$$C \rightarrow^r A$$
$$C+A \rightarrow^s C'+A$$
$$C'+A \rightarrow^s A+A$$





Robust Stochastic Oscillation

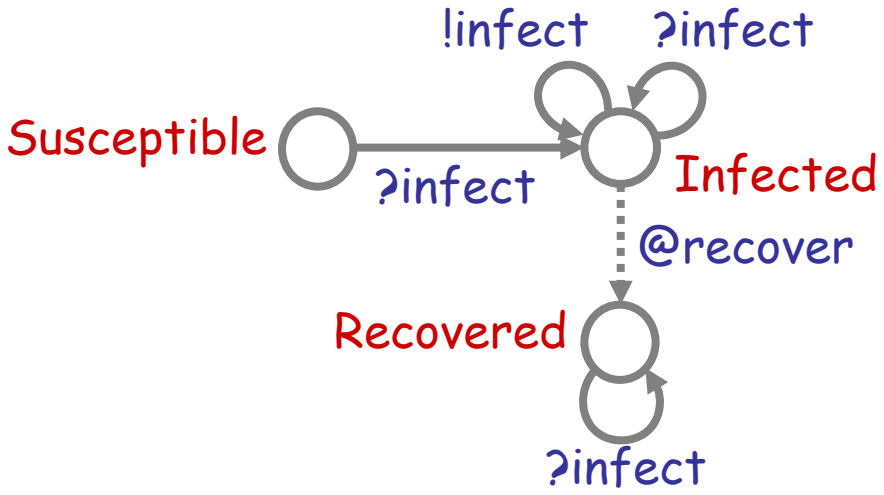
$$[A]^\bullet = -r[A]-s[A][B]+r[C]+s[C'][A]$$
$$[B]^\bullet = -r[B]-s[B][C]+r[A]+s[A'][B]$$
$$[C]^\bullet = -r[C]-s[C][A]+r[B]+s[B'][C]$$
$$[A']^\bullet = -s[A'][B] + s[A][B]$$
$$[B']^\bullet = -s[B'][C] + s[B][C]$$
$$[C']^\bullet = -s[C'][A] + s[C][A]$$

Epidemics

Kermack, W. O. and McKendrick, A. G. "**A Contribution to the Mathematical Theory of Epidemics.**" *Proc. Roy. Soc. Lond. A* **115**, 700-721, 1927.

http://mathworld.wolfram.com/Kermack-McKendrickModel.html

Epidemics



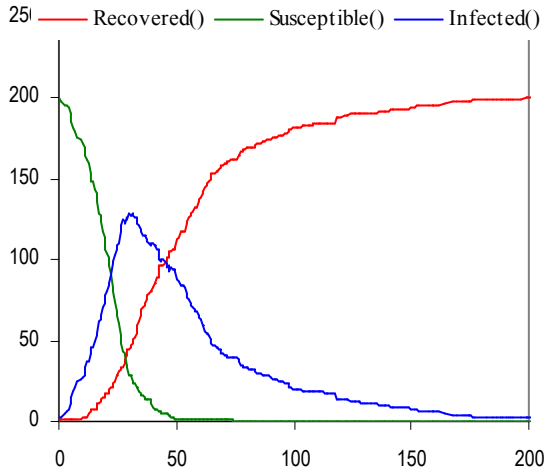

```
directive sample 500.0 1000
directive plot Recovered(); Susceptible(); Infected()

new infect @0.001:chan()
val recover = 0.03

let Recovered() =
  ?infect; Recovered()

and Susceptible() =
  ?infect; Infected()

and Infected() =
  do !infect; Infected()
  or ?infect; Infected()
  or delay@recover; Recovered()

run (200 of Susceptible() | 2 of Infected())
```

Developing the Use of Process Algebra in the Derivation and Analysis of Mathematical Models of Infectious Disease

R. Norman and C. Shankland

Department of Computing Science and Mathematics, University of Stirling, UK.
{ces,ran}@cs.stir.ac.uk

Abstract. We introduce a series of descriptions of disease spread using the process algebra WSCCS and compare the derived mean field equations with the traditional ordinary differential equation model. Even the preliminary work presented here brings to light interesting theoretical questions about the "best" way to defined the model.



ODE

$S = ?i_{(t)}.I$

$I = !i_{(t)}.I \oplus ?i_{(t)}.I \oplus \tau_r.R$

$R = ?i_{(t)}.R$

$S + I \rightarrow^{t} I + I$

$I + I \rightarrow^{t} I + I$

$I \rightarrow^{r} R$

$R + I \rightarrow^{t} R + I$

"useless" reactions

$[S]^{\bullet} = -t[S][I]$

$[I]^{\bullet} = t[S][I]-r[I]$

$[R]^{\bullet} = r[I]$

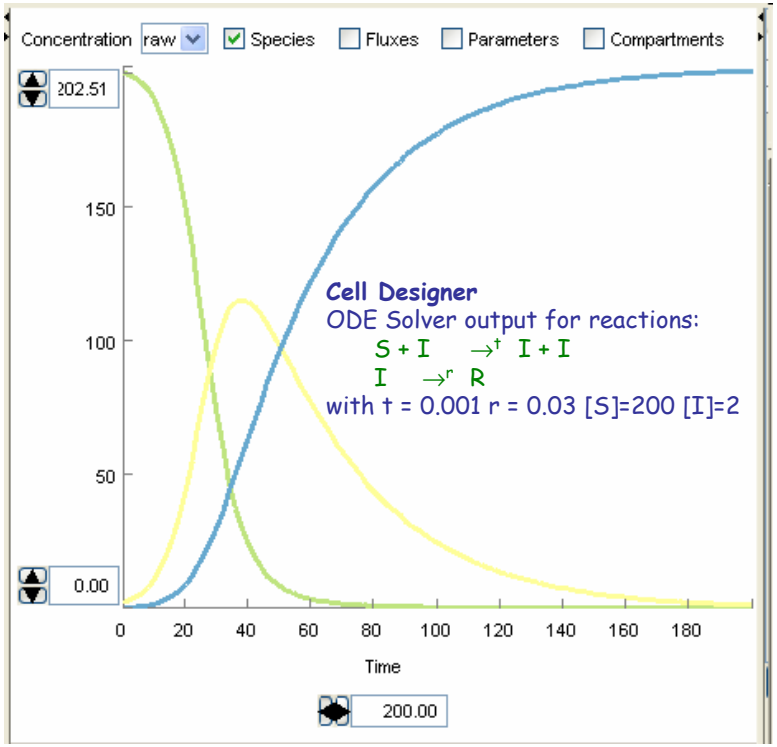
Automata match the standard ODE model!

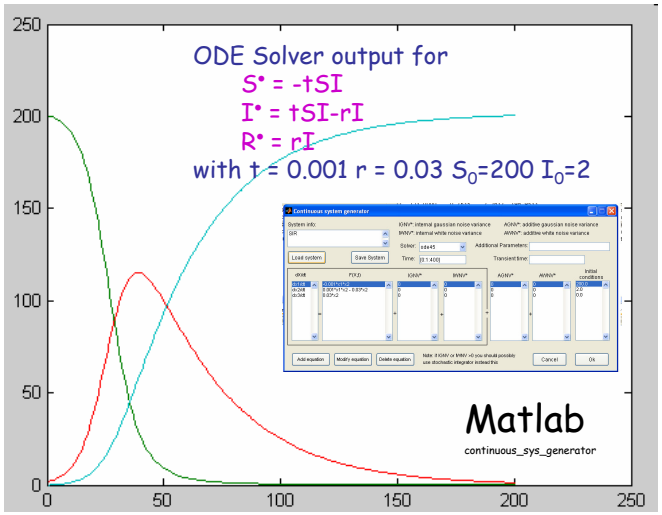
$$\frac{dS}{dt} = -aIS$$

$$\frac{dI}{dt} = aIS - bI$$

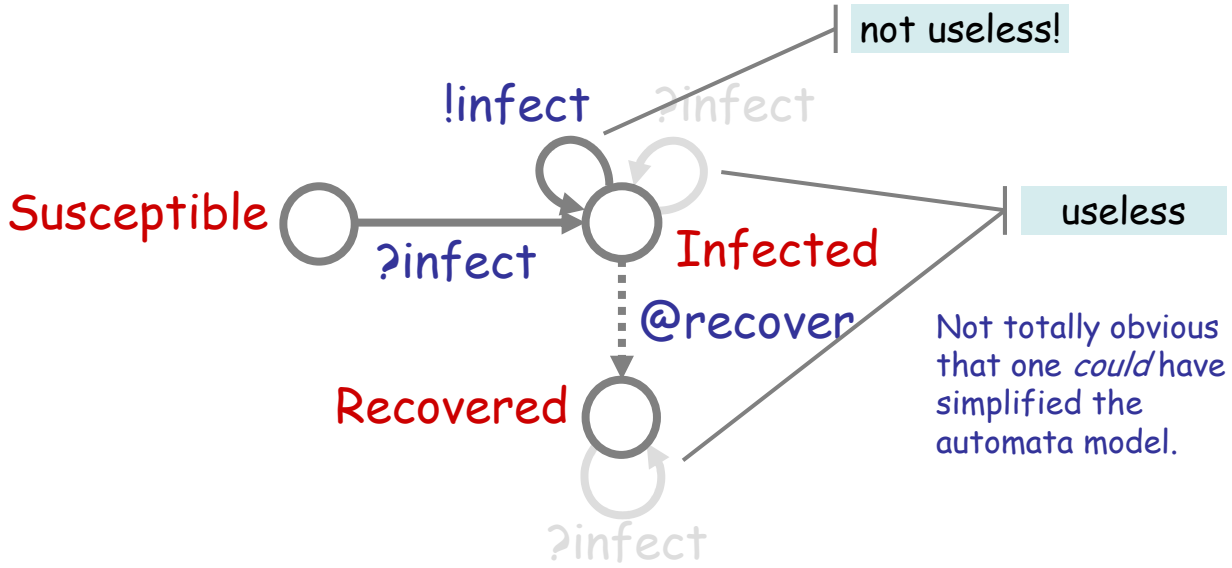
$$\frac{dR}{dt} = bI$$

(the Kermack-McKendrick, or SIR model)[





Simplified Model



Not totally obvious that one *could* have simplified the automata model.

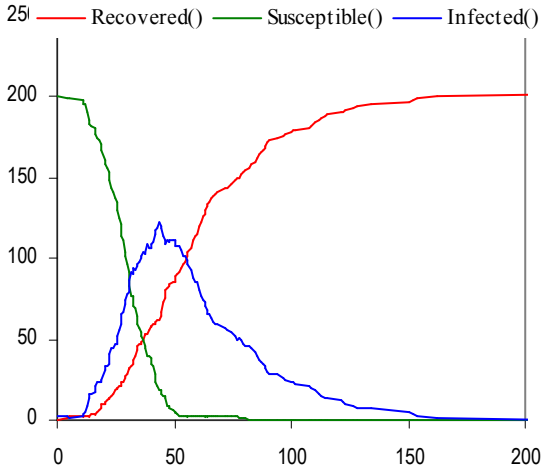
$$S = ?i_{(t)};I$$
$$I = !i_{(t)};I \oplus \tau_r;R$$
$$R = 0$$

$$S + I \rightarrow^{t} I + I$$
$$I \rightarrow^{r} R$$

$$[S]^{\bullet} = -t[S][I]$$
$$[I]^{\bullet} = t[S][I]-r[I]$$
$$[R]^{\bullet} = r[I]$$

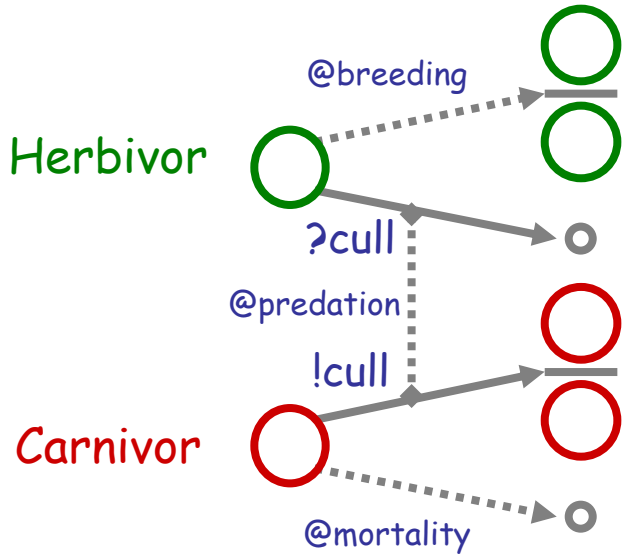
Same ODE, hence equivalent automata models.

```
directive sample 500.0 1000
directive plot Recovered(); Susceptible(); Infected()

new infect @0.001:chan()
val recover = 0.03

let Recovered() =
  ()

and Susceptible() =
  ?infect; Infected()

and Infected() =
  do !infect; Infected()
  or delay@recover; Recovered()

run (200 of Susceptible() | 2 of Infected())
```

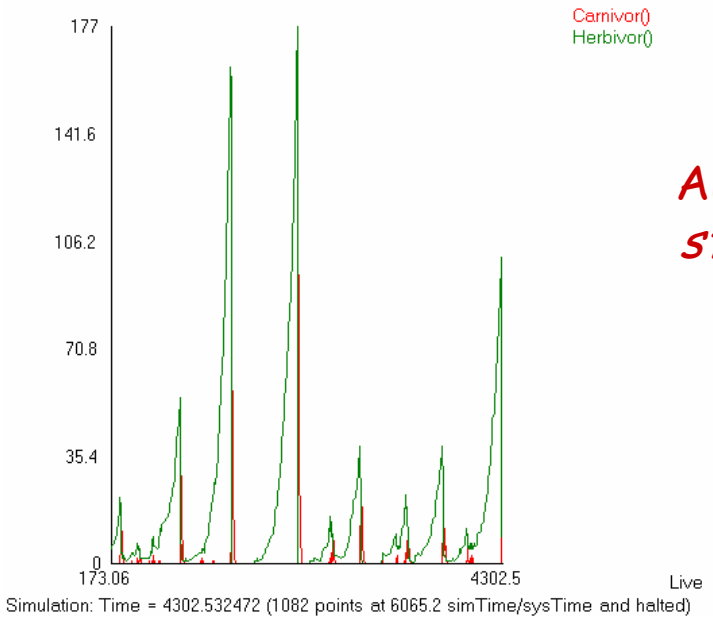


Lotka-Volterra

Predator-Prey




```
directive sample 5000.0 1000
directive plot Carnivor(); Herbivor()

val mortality = 0.01
val breeding = 0.01
val predation = 0.01
new cull @predation:chan()

let Herbivor() =
  do delay@breeding; (Herbivor() | Herbivor())
  or ?cull; ()

and Carnivor() =
  do delay@mortality; ()
  or !cull; (Carnivor() | Carnivor())

run replicate delay@0.01; (Herbivor() | Carnivor())
```



An unbounded state system!

Since predator and prey drive each other to exinction (stochastically), we restart the popolations periodically.

(This is a case where the continuous system oscillates and the stochastic one does not! We have seen examples of the opposite situation.)

ODE

$H = \tau_b.\ (H|H) \oplus ?c_{(p)}.0$
$C = \tau_m.0 \oplus !c_{(p)}.(C|C)$

$H \rightarrow^{b} H + H$
$C \rightarrow^{m} 0$
$H + C \rightarrow^{p} C + C$

$[H]^{\bullet} = b[H]-p[H][C]$
$[C]^{\bullet} = -m[C]+p[H][C]$

Lotka-Volterra Equations

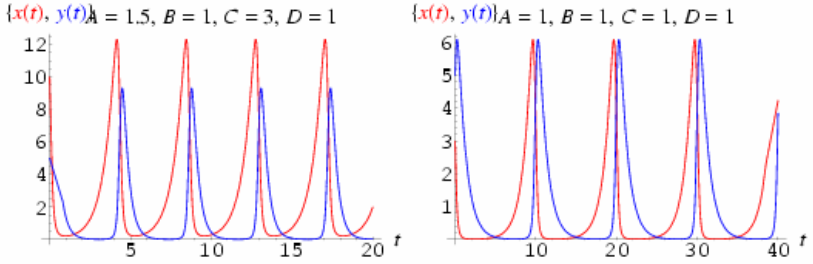
mathworld

COMMENT On this Page DOWNLOAD *Mathematica* Notebook

The Lotka-Volterra equations describe an ecological predator-prey (or parasite-host) model which assumes that, for a set of fixed positive constants A (the growth rate of prey), B (the rate at which predators destroy prey), C (the death rate of predators), and D (the rate at which predators increase by consuming prey), the following conditions hold.

1. A prey population x increases at a rate $dx = A x\,dt$ (proportional to the number of prey) but is simultaneously destroyed by predators at a rate $dx = -B x y\,dt$ (proportional to the product of the numbers of prey and predators).

2. A predator population y decreases at a rate $dy = -C y\,dt$ (proportional to the number of predators), but increases at a rate $dy = D x y\,dt$ (again proportional to the product of the numbers of prey and predators).



This gives the coupled differential equations

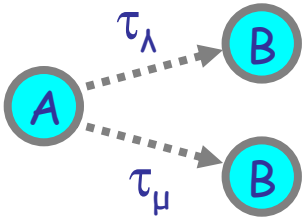
$$\frac{dx}{dt} = Ax - Bxy \quad (1)$$
$$\frac{dy}{dt} = -Cy + Dxy, \quad (2)$$

Automata match the Lotka-Volterra model (with B=D)

Laws by ODEs

Choice Law by ODEs

$$\tau_\lambda;B \oplus \tau_\mu;B = \tau_{\lambda+\mu};B$$

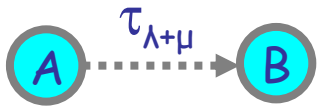


$$A = \tau_\lambda;B \oplus \tau_\mu;B$$

$$A \to^{\lambda} B \qquad A \to^{\mu} B$$

$$[A]^\bullet = -\lambda[A] - \mu[A] \qquad [B]^\bullet = \lambda[A] + \mu[A]$$

=



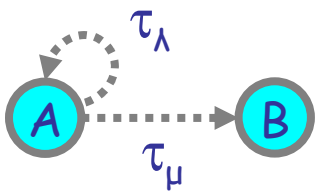
$$A = \tau_{\lambda+\mu};B$$

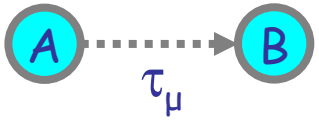
$$A \to^{\lambda+\mu} B$$

$$[A]^\bullet = -(\lambda+\mu)[A] \qquad [B]^\bullet = (\lambda+\mu)[A]$$

Idle Delay Law by ODEs

$A = \tau_\lambda;A \oplus \tau_\mu;B \quad = \quad A = \tau_\mu;B$





$A = \tau_\lambda;A \oplus \tau_\mu;B$

$A \rightarrow^{\lambda} A$
$A \rightarrow^{\mu} B$

$[A]^\bullet = -\mu[A]$
$[B]^\bullet = \mu[A]$

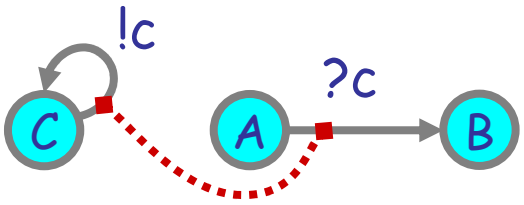
=

$A = \tau_\mu;B$

$A \rightarrow^{\mu} B$

$[A]^\bullet = -\mu[A]$
$[B]^\bullet = \mu[A]$

Idle Interaction Law by ODEs

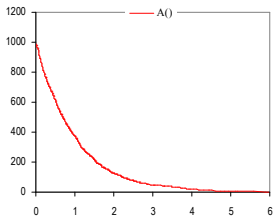


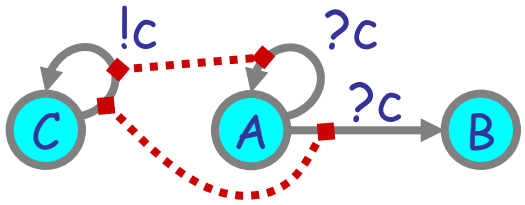
$A = ?c;B$
$C = !c;C$

$A+C \rightarrow^{r} B+C$

$[A]^{\bullet} = -r[A][C]$
$[B]^{\bullet} = r[A][C]$
$[C]^{\bullet} = 0$

```
directive sample 6.0 1000
directive plot A()

new c@1.0:chan

let A() = ?c; B()
and B() = ()
and C() = !c; C()

run (C() | 1000 of A())
```





$A = ?c;A \oplus ?c;B$
$C = !c;C$

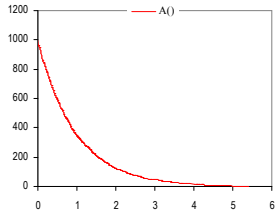
$A+C \rightarrow^{r} A+C$
$A+C \rightarrow^{r} B+C$

$[A]^{\bullet} = -r[A][C]$
$[B]^{\bullet} = r[A][C]$
$[C]^{\bullet} = 0$

It may seem like A should decrease half as fast, but NO! Two ways to explain:

- -State A is *memoryless* of any past idling.
- - Activity on c is double

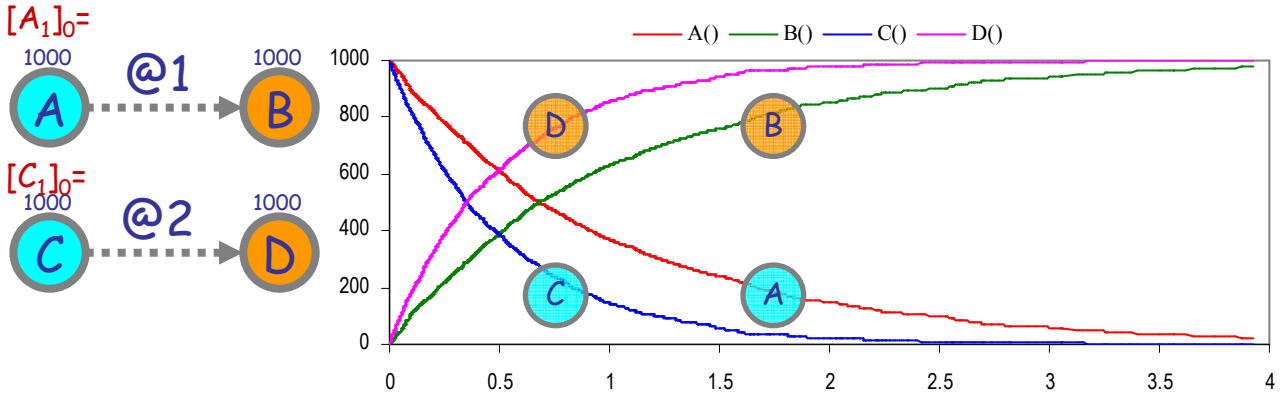
```
directive sample 6.0 1000
directive plot A()

new c@1.0:chan

let A() = do ?c; B() or ?c; A()
and B() = ()
and C() = !c; C()

run (C() | 1000 of A())
```



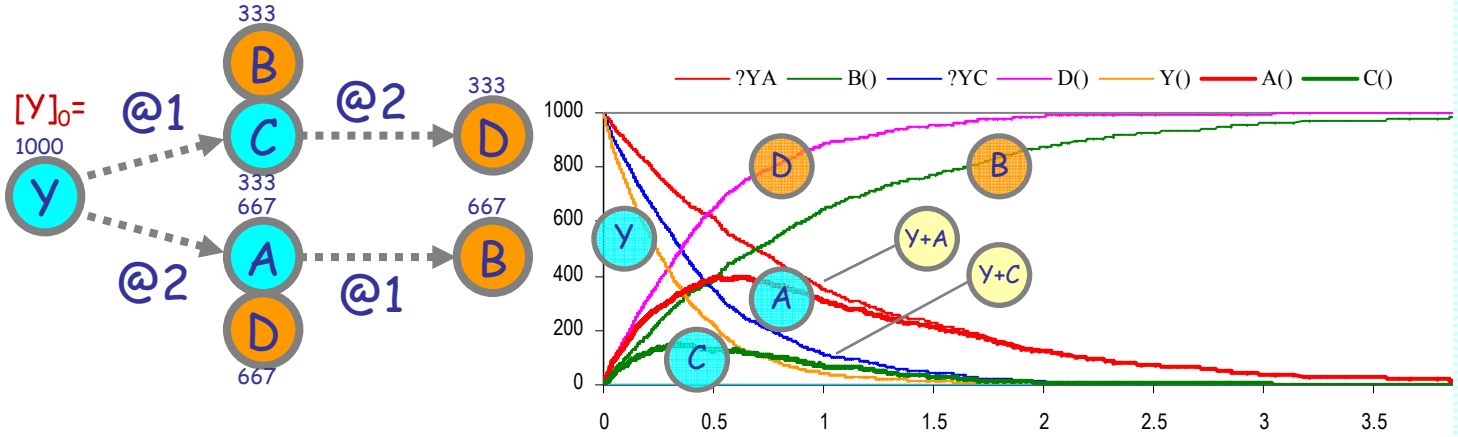
Asynchronous Interleaving

Hermanns: Interactive Markov Chains. Sec 4.1.2

$$\tau_\lambda;B \mid \tau_\mu;D = \tau_\lambda;(B \mid \tau_\mu;D) + \tau_\mu;(\tau_\lambda;B \mid D)$$




```
directive sample 4.0 10000
directive plot A(); B(); C(); D()

let A() = delay@1.0; B()
and B() = ()

let C() = delay@2.0; D()
and D() = ()

run 1000 of (A() | C())
```




```
directive sample 4.0 10000
directive plot
   ?YA; B(); ?YC; D(); Y(); A(); C()
new YA@1.0:chan new YC@1.0:chan

let A() = do delay@1.0; B() or ?YA
and B() = ()

let C() = do delay@2.0; D() or ?YC
and D() = ()

let Y() =
   do delay@1.0; (B() | C())
   or delay@2.0; (A() | D())
   or ?YA or ?YC

run 1000 of Y()
```

Amazingly, the B's and the D's from the two branches sum up to exponential distributions

Asynchronous Interleaving Law by ODEs

$$\tau_\lambda;B \mid \tau_\mu;D = \tau_\lambda;(B \mid \tau_\mu;D) + \tau_\mu;(\tau_\lambda;B \mid D)$$

Want to show that B and D on both sides have the "same behavior" (equal quantities of B and D produced at all times)

$A_1 = \tau_\lambda;B$
$C_1 = \tau_\mu;D$
$A_1 \mid C_1$

$A_1 \to^\lambda B$
$C_1 \to^\mu D$
$A_1 + C_1$

$[A_1]^\bullet = -\lambda[A_1]$
$[B]^\bullet = \lambda[A_1]$
$[C_1]^\bullet = -\mu[C_1]$
$[D]^\bullet = \mu[C_1]$

=?

$Y = \tau_\lambda;(B \mid C_2) \oplus \tau_\mu;(A_2 \mid D)$
$C_2 = \tau_\mu;D$
$A_2 = \tau_\lambda;B$
Y

$Y \to^\lambda B + C_2$
$Y \to^\mu A_2 + D$
$C_2 \to^\mu D$
$A_2 \to^\lambda B$
Y

$[Y]^\bullet = -\lambda[Y]-\mu[Y]$
$[A_2]^\bullet = \mu[Y]-\lambda[A_2]$
$[B]^\bullet = \lambda[Y]+\lambda[A_2]$
$[C_2]^\bullet = \lambda[Y]-\mu[C_2]$
$[D]^\bullet = \mu[Y]+\mu[C_2]$

$[Y+A_2]^\bullet = -\lambda[Y+A_2]$
$[B]^\bullet = \lambda[Y+A_2]$
$[Y+C_2]^\bullet = -\mu[Y+C_2]$
$[D]^\bullet = \mu[Y+C_2]$

$[Y+A_2]^\bullet = [Y]^\bullet+[A_2]^\bullet$
$= -\lambda[Y]-\mu[Y]+\mu[Y]-\lambda[A_2]$
$= -\lambda[Y]-\lambda[A_2]$
$= -\lambda[Y+A_2]$

$[Y+A_2]$ decays exponentially!

[B] and [D] have equal time evolutions on the two sides provided that $[A_1]=[Y+A_2]$ and $[C_1]=[Y+C_2]$.
This imposes the constraint, in particular, that $[A_1]_0=[Y+A_2]_0$ and $[C_1]_0=[Y+C_2]_0$ (at time zero).
The initial conditions of the right hand system specify that $[A_2]_0=[C_2]_0=0$ (since only Y is present).
Therefore, we obtain that $[A_1]_0=[C_1]_0=[Y]_0$.

So, for example, if we run a stochastic simulation of the left hand side with 1000*A1 and 1000*C1, we obtain the same curves for B and D than a stochastic simulation of the right hand side with 1000*Y.

Parametric Form

Chemical Parametric Form (CPF)

$E ::= X_1(\mathbf{p}_1)=M_1, \ldots, X_n(\mathbf{p}_n)=M_n$	Definitions	$(n \geq 0)$
$M ::= \pi_1;P_1 \oplus \ldots \oplus \pi_n;P_n$	Molecules	$(n \geq 0)$
$P ::= X_1(\mathbf{p}_1) \mid \ldots \mid X_n(\mathbf{p}_n)$	Solutions	$(n \geq 0)$
$\pi ::= \tau_r \;\; ?n(\mathbf{p}) \;\; !n(\mathbf{p})$	Interactions	
$CPF ::= E,P$	with initial conditions	

$\oplus$ is stochastic choice (vs. + for chemical reactions)
0 is the null solution ($P|0 = 0|P = P$)
and null molecule ($M \oplus 0 = 0 \oplus M = M$) ($\tau_0;P = 0$)
X_i are distinct in E, **p** are vectors of names
p are vectors of distinct names when in binding position
Each free name n in E is assigned a fixed rate r:
written either $n_{(r)}$, or $\rho_{CPF}(n)=r$.

A translation from CPF to CGF exists
(expanding all possible instantiation of parameters from the initial conditions)

An incremental translation algorithm exists
(expanding on demand from initial conditions)

Repressilator ODEs

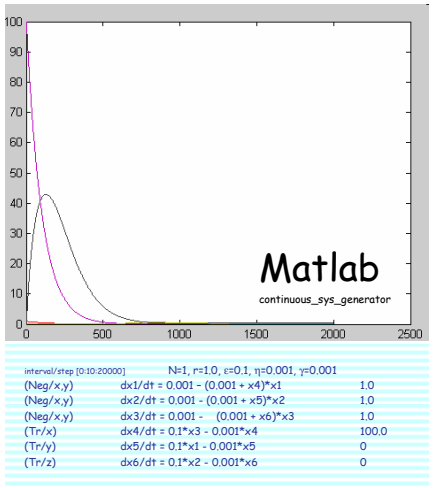
$Neg(a,b) = ?a;\ Inh(a,b) \oplus \tau_\varepsilon;\ (Tr(b) \mid Neg(a,b))$
$Inh(a,b) = \tau_\eta;\ Neg(a,b)$
$Tr(b) = !b;\ Tr(b) \oplus \tau_\gamma;\ 0$
$Neg(x_{(r)},y_{(r)}) \mid Neg(y_{(r)},z_{(r)}) \mid Neg(z_{(r)},x_{(r)})$

$Neg/x,y \to^{\varepsilon} Tr/y + Neg/x,y$
$Neg/y,z \to^{\varepsilon} Tr/z + Neg/y,z$
$Neg/z,x \to^{\varepsilon} Tr/x + Neg/z,x$
$Tr/x + Neg/x,y \to^{r} Tr/x + Inh/x,y$
$Tr/y + Neg/y,z \to^{r} Tr/y + Inh/y,z$
$Tr/z + Neg/z,x \to^{r} Tr/z + Inh/z,x$
$Inh/x,y \to^{\eta} Neg/x,y$
$Inh/y,z \to^{\eta} Neg/y,z$
$Inh/z,x \to^{\eta} Neg/z,x$
$Tr/x \to^{\gamma} 0$
$Tr/y \to^{\gamma} 0$
$Tr/z \to^{\gamma} 0$
$Neg/x,y + Neg/y,z + Neg/z,x$

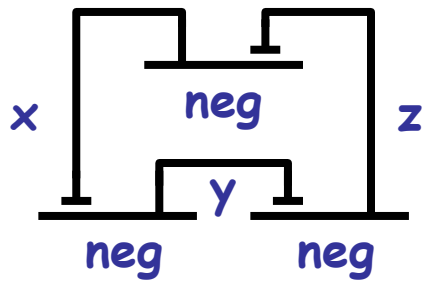
$[Neg/x,y]^\bullet = -r[Tr/x][Neg/x,y] + \eta[Inh/x,y]$
$[Neg/y,z]^\bullet = -r[Tr/y][Neg/y,z] + \eta[Inh/y,z]$
$[Neg/z,x]^\bullet = -r[Tr/z][Neg/z,x] + \eta[Inh/z,x]$
$[Inh/x,y]^\bullet = r[Tr/x][Neg/x,y] - \eta[Inh/x,y]$
$[Inh/y,z]^\bullet = r[Tr/y][Neg/y,z] - \eta[Inh/y,z]$
$[Inh/z,x]^\bullet = r[Tr/z][Neg/z,x] - \eta[Inh/z,x]$
$[Tr/x]^\bullet = \varepsilon[Neg/z,x] - \gamma[Tr/x]$
$[Tr/y]^\bullet = \varepsilon[Neg/x,y] - \gamma[Tr/y]$
$[Tr/z]^\bullet = \varepsilon[Neg/y,z] - \gamma[Tr/z]$

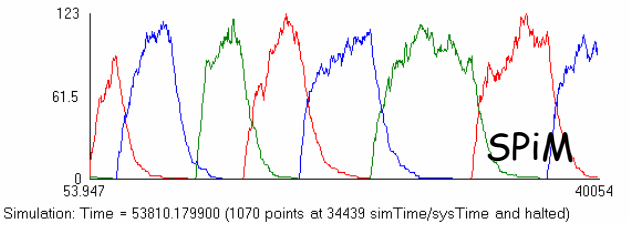
simplifying (N is the quantity of each of the 3 gates)

$[Neg/x,y]^\bullet = \eta N - (\eta + r[Tr/x])[Neg/x,y]$
$[Neg/y,z]^\bullet = \eta N - (\eta + r[Tr/y])[Neg/y,z]$
$[Neg/z,x]^\bullet = \eta N - (\eta + r[Tr/z])[Neg/z,x]$
$[Tr/x]^\bullet = \varepsilon[Neg/z,x] - \gamma[Tr/x]$
$[Tr/y]^\bullet = \varepsilon[Neg/x,y] - \gamma[Tr/y]$
$[Tr/z]^\bullet = \varepsilon[Neg/y,z] - \gamma[Tr/z]$



No sustained oscillations (with SPiM parameters). But see Elowitz&Leibler.





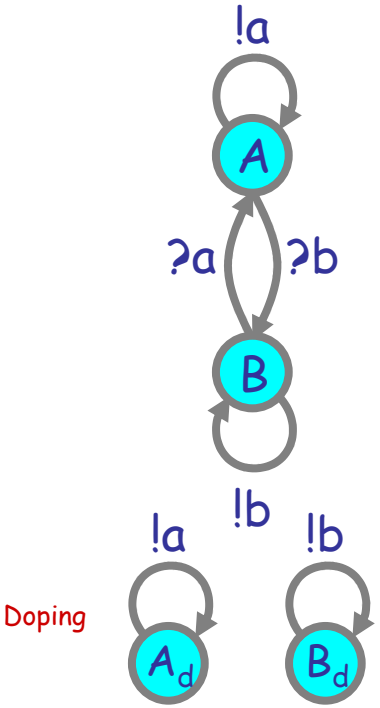

```
directive sample 50000.0 1000
directive plot !a; !b; !c

val dk = 0.001   (* Decay rate *)
val inh = 0.001  (* Inhibition rate *)
val cst = 0.1    (* Constitutive rate *)

let tr(p:chan()) = do !p; tr(p) or delay@dk

let neg(a:chan(), b:chan()) =
 do ?a; delay@inh; neg(a,b)
 or delay@cst; (tr(b) | neg(a,b))

val bnd = 1.0    (* Protein binding rate *)
new a@bnd:chan() new b@bnd:chan() new
c@bnd:chan()
run (neg(c,a) | neg(a,b) | neg(b,c))
```

Groupies ODE

Doped Groupies ODE

Relationship between stochastic and deterministic semantics is non-obvious!

Q: What does this do?



$A = !a_{(r)};A \oplus ?b_{(r)};B$ $B = !b_{(r)};B \oplus ?a_{(r)};A$	$A_d = !a_{(r)};A_d$ $B_d = !b_{(r)};B_d$

```
directive sample 10.0 1000
directive plot Ga(); Gb(); Da(); Db()

new a@1.0:chan()
new b@1.0:chan()

let Ga() = do !a; Ga() or ?b; Gb()
and Gb() = do !b; Gb() or ?a; Ga()

let Da() = !a; Da()
and Db() = !b; Db()

run  1 of (Da() | Db())
run 100 of (Ga() | Gb())
```

$A+B \rightarrow^r A+A$ $B+A \rightarrow^r B+B$	$A+B_d \rightarrow^r B+B_d$ $B+A_d \rightarrow^r A+A_d$

$[A]^\bullet = r[A][B]-r[B][A]-r[A][B_d]+r[B][A_d]$ $[B]^\bullet = r[B][A]-r[A][B]-r[B][A_d]+r[A][B_d]$	$[A_d]^\bullet = 0$ $[B_d]^\bullet = 0$

$[A_d]$,$[B_d]$ are constant; assume them both = k

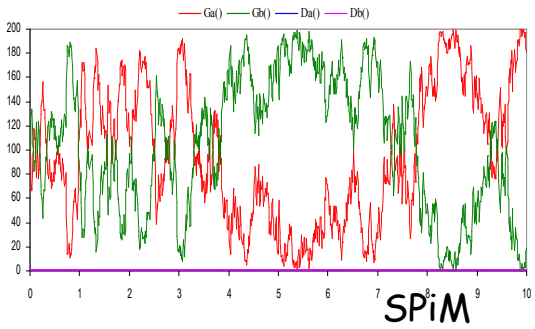
$$[A]^\bullet = -rk([A]-[B])$$
$$[B]^\bullet = rk([A]-[B])$$

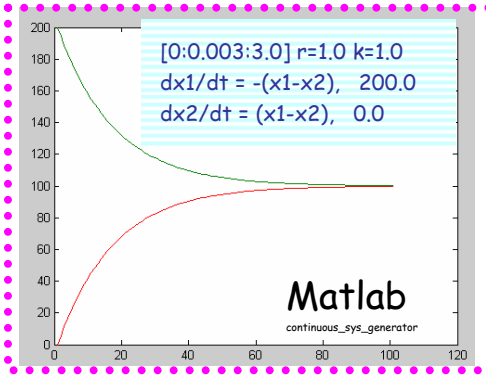
At [B]=0: $[A]^\bullet=-rk[A]$, $[B]^\bullet=rk[A]$

At $[A]\approx[B]$: $[A]^\bullet=[B]^\bullet\approx0$

At [A]=[B]: $[A]^\bullet=[B]^\bullet=0$

Stochastic Answer:
bounded random walk





Deterministic Answer:
convergence and ***stability***

ODE predicts converging stable equilibrium at [A]=[B] instead of the total chaos observed in the stochastic system!

For k=0 (no dope), predicts deadlock $[A]^\bullet=[B]^\bullet=0$ but at any value of [A], which is definitely not true in the stochastic system.

Conclusions

Conclusions

- Stochastic Collectives
  - Complex global behavior from simple components
  - Emergence of collective functionality from "non-functional" components
  - (Cf. "swarm intelligence": simple global behavior from complex components)

- Artificial Biochemistry
  - Stochastic collectives with Law of Mass Interaction kinetics
  - Connections to classical Markov theory, chemical Master Equation, and Rate Equation

- Properties of collective behavior
  - Simulation
  - Systematic translation to ODEs from parametric process "libraries"
  - Correspondence (or not) between stochastic and deterministic behavior

- Interdisciplinary connections
  - Process descriptions vs. chemical descriptions
  - Process descriptions vs. ODE descriptions