Artificial Biochemistry

Luca Cardelli

Microsoft Research

Understanding Adaptive
Multi-Component Systems
Rostock 2006-11-10

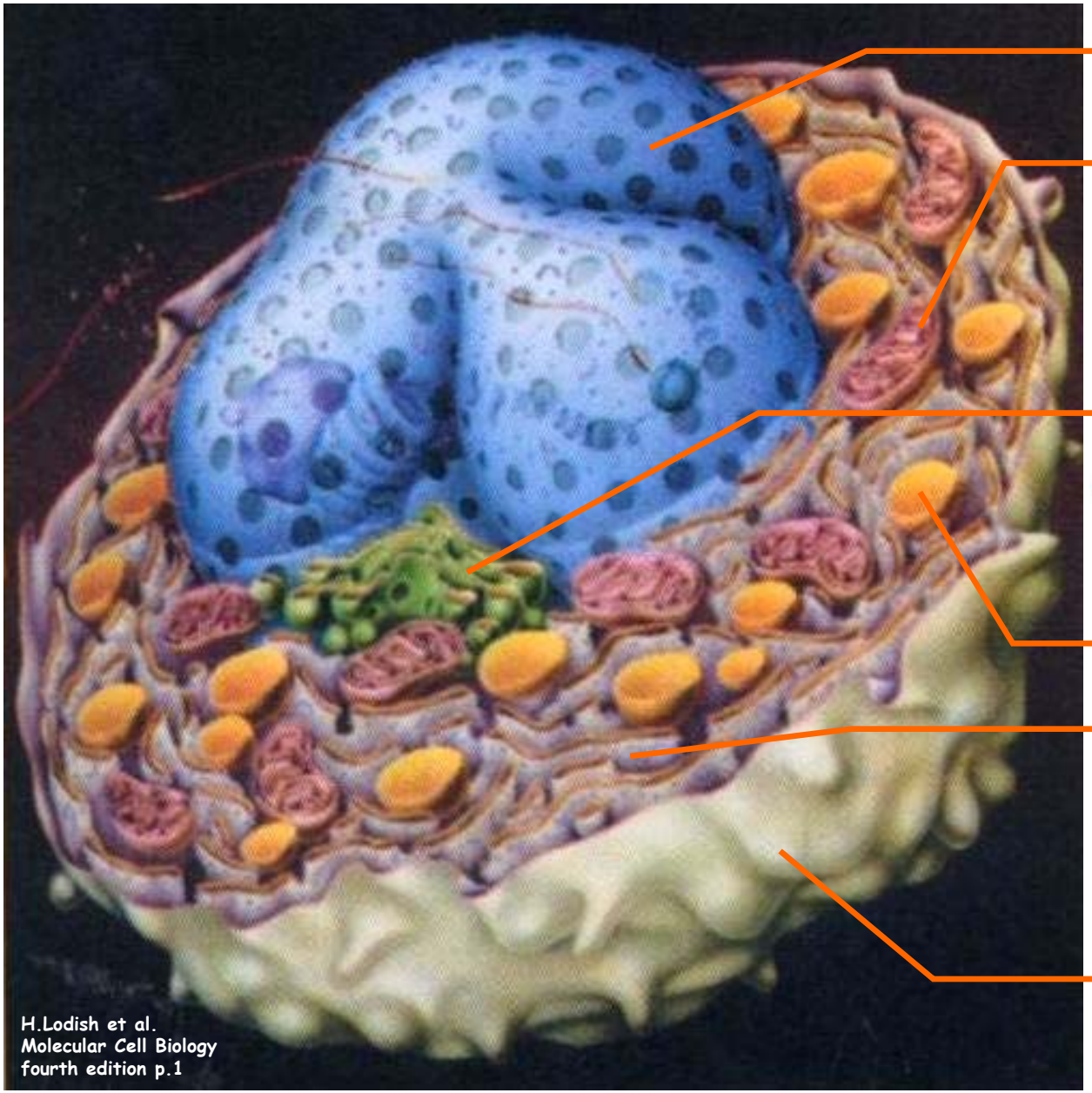
http://LucaCardelli.name

Reverse-Engineer This!

Eukaryotic Cell

(10~100 trillion in human body)

Membranes everywhere



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

dna inside

…and Model It

- Even if we understood it, how would we model it?
  - Millions of differential equations? Hmmm..
- And we will have to model it in order to understand it.
- What's different about modeling these systems?

Stochastic Collectives

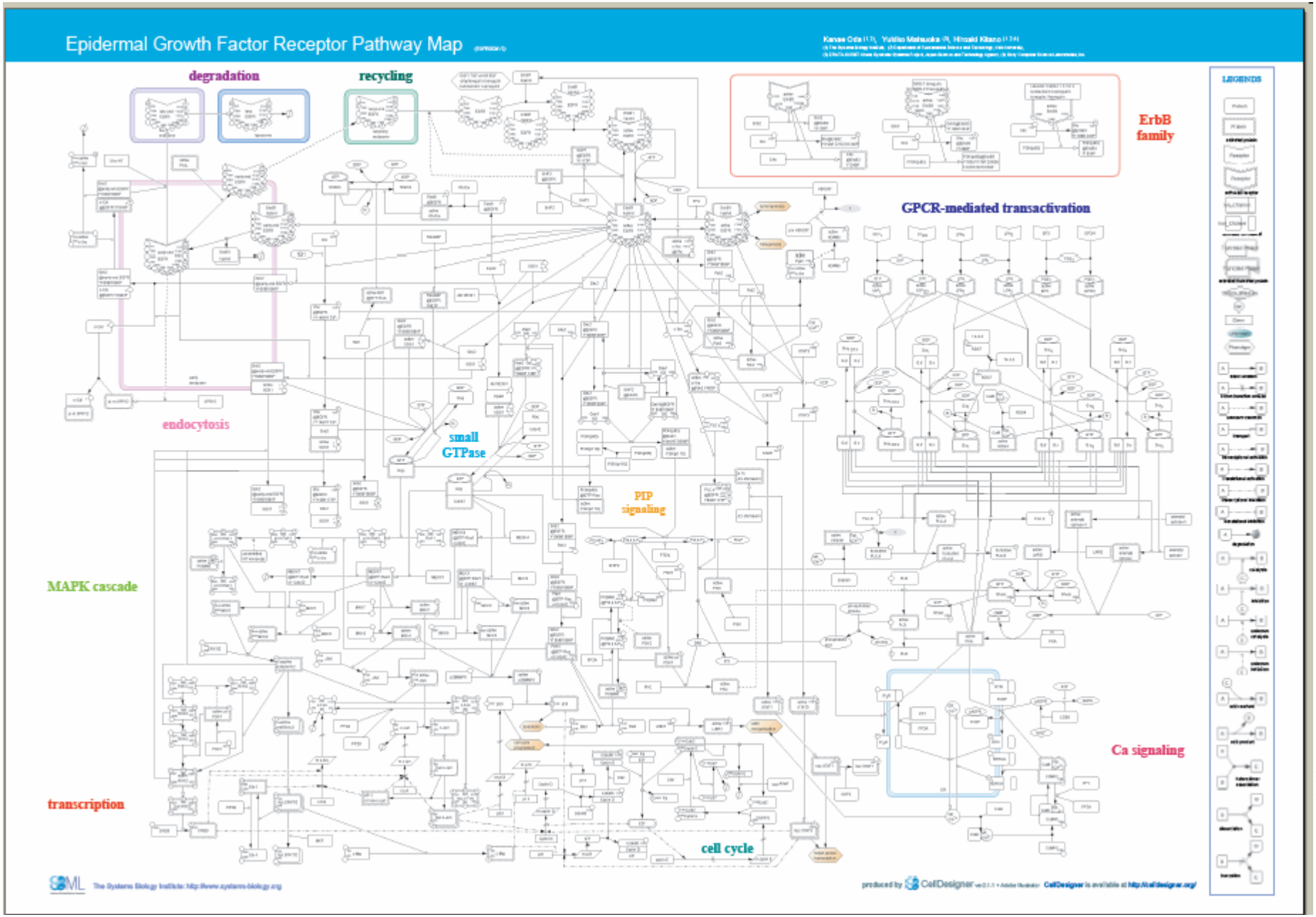
Stochastic Collectives

- "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"

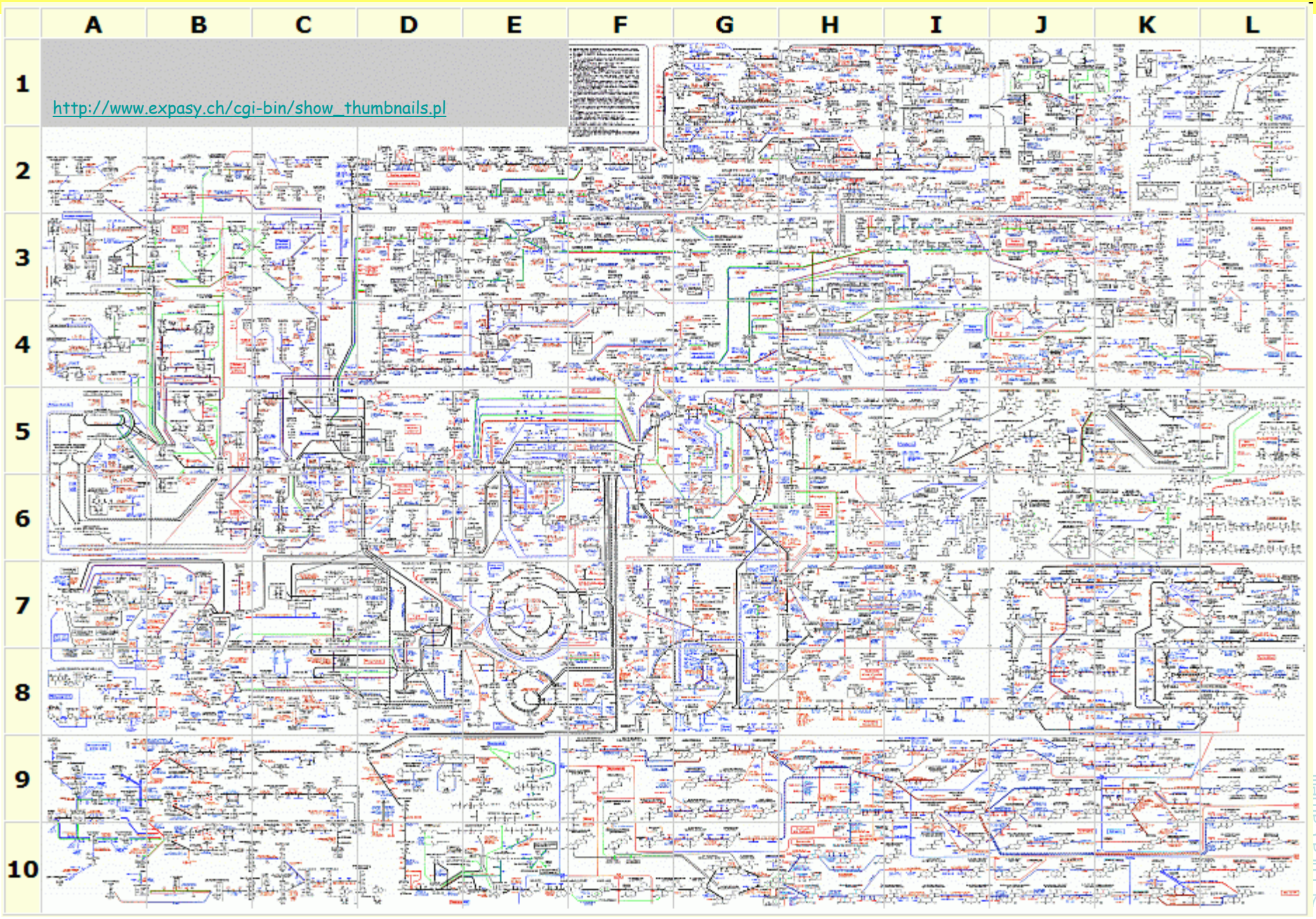
- "Stochastic":
  - Interactions have *rates*
    - Not quite discrete (hundreds or thousands of components)
    - Not quite continuous (non-trivial stochastic effects)
    - Not quite hybrid (no "switching" between regimes)

- 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].

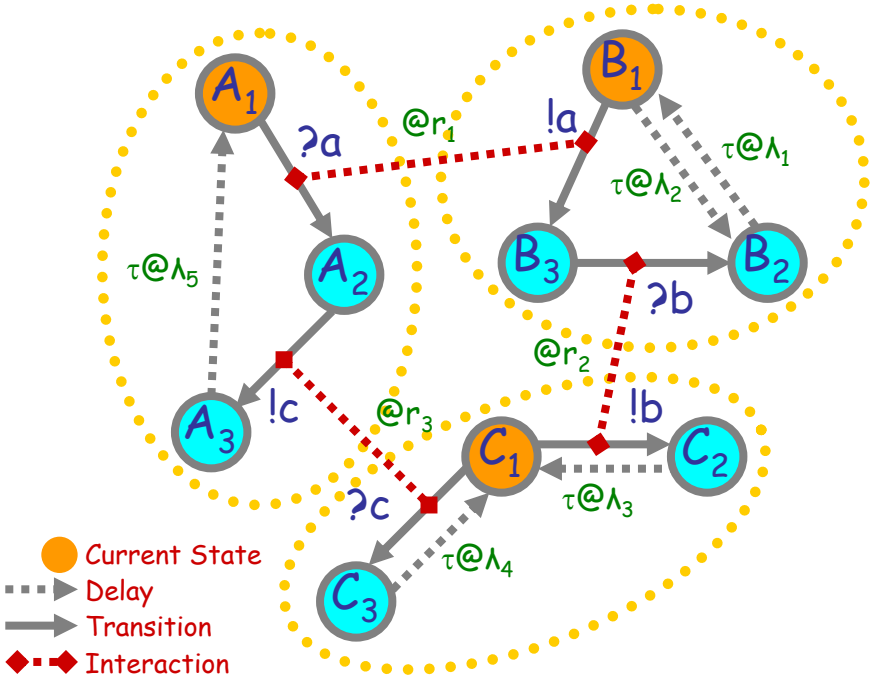
State Transitions



Even More State Transitions



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.

$$\begin{aligned}
&\text{new } a@r_1 \\
&\text{new } b@r_2 \\
&\text{new } c@r_3
\end{aligned}\Big\}\ \text{Communication channels}$$

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

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

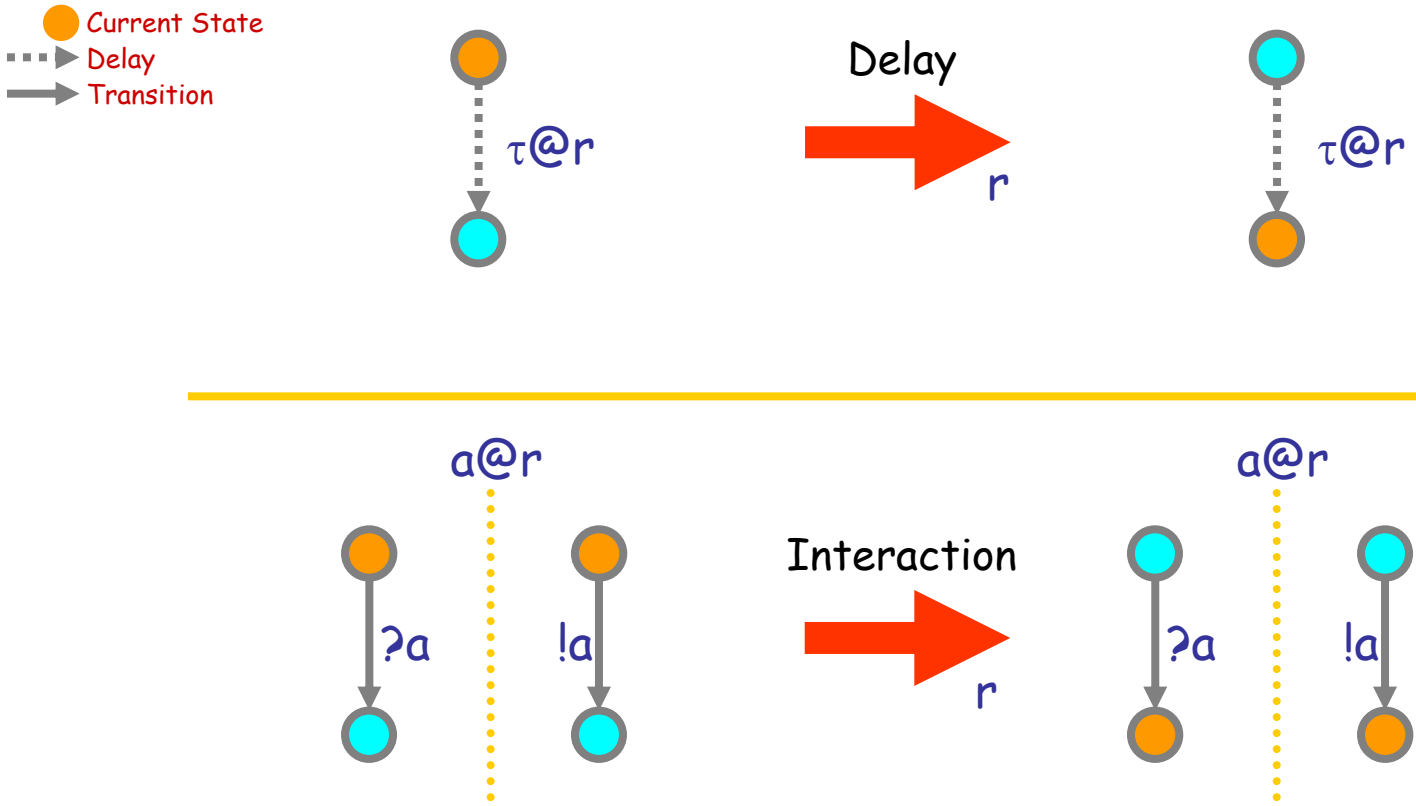
$$\begin{aligned}
C_1 &= !b;\ C_2 + ?c;\ C_3 \\
C_2 &= \tau@\lambda_3;\ C_1 \\
C_3 &= \tau@\lambda_4;\ C_2
\end{aligned}\Big\}\ \text{Automata}$$

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

"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]

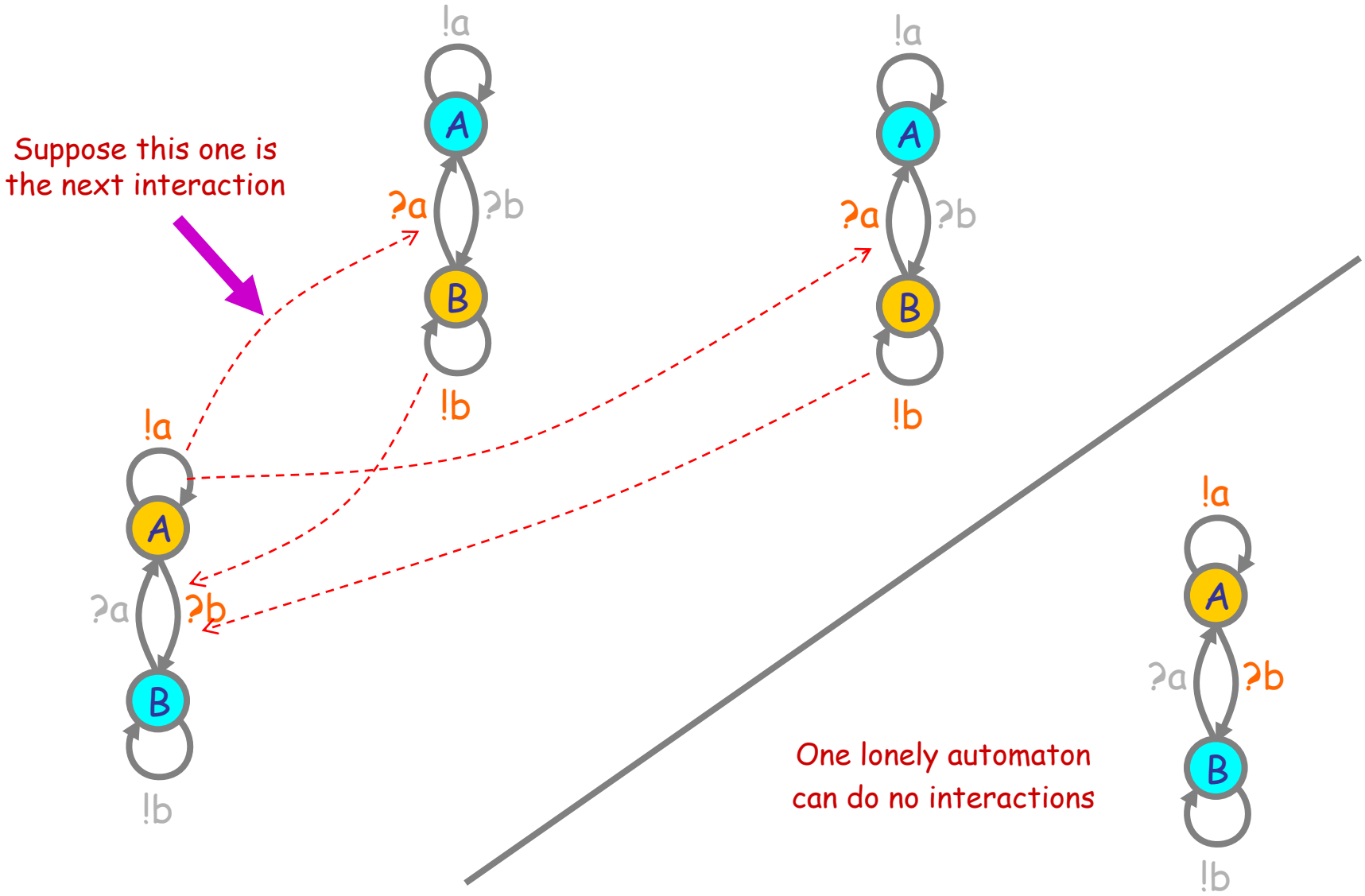
Interacting Automata Transition Rules



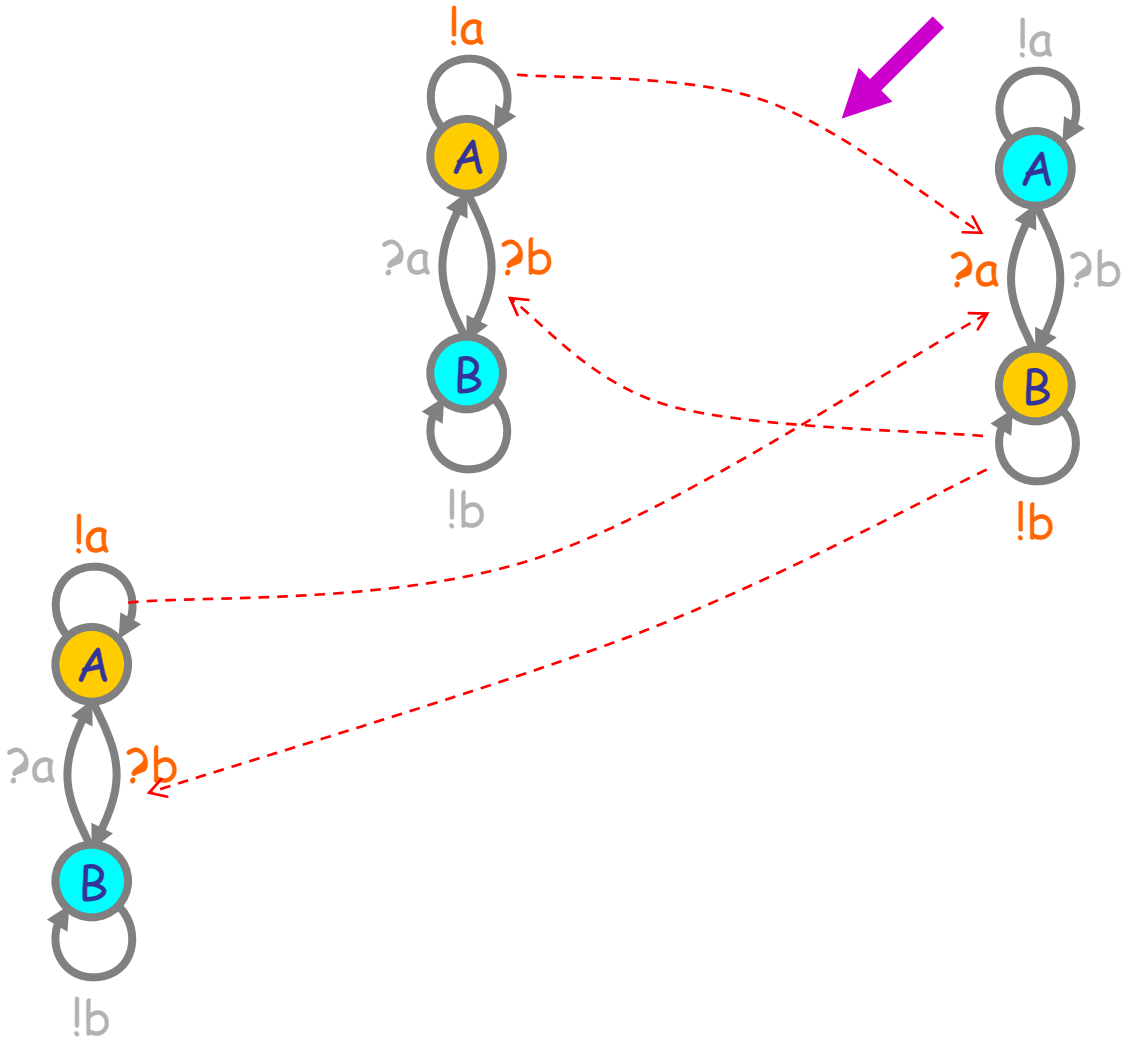
Q: What kind of mass behavior can this produce?

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

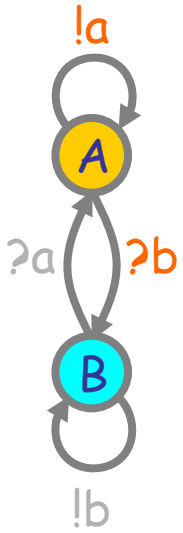
Interactions in a Population

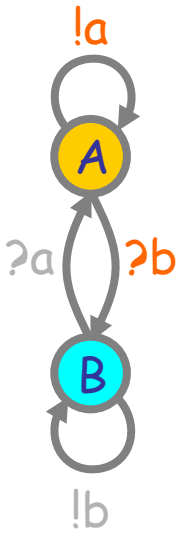


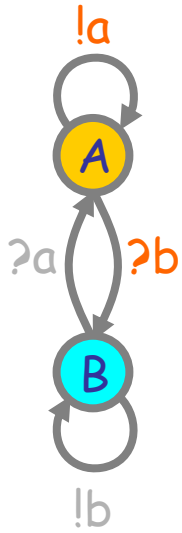
Interactions in a Population



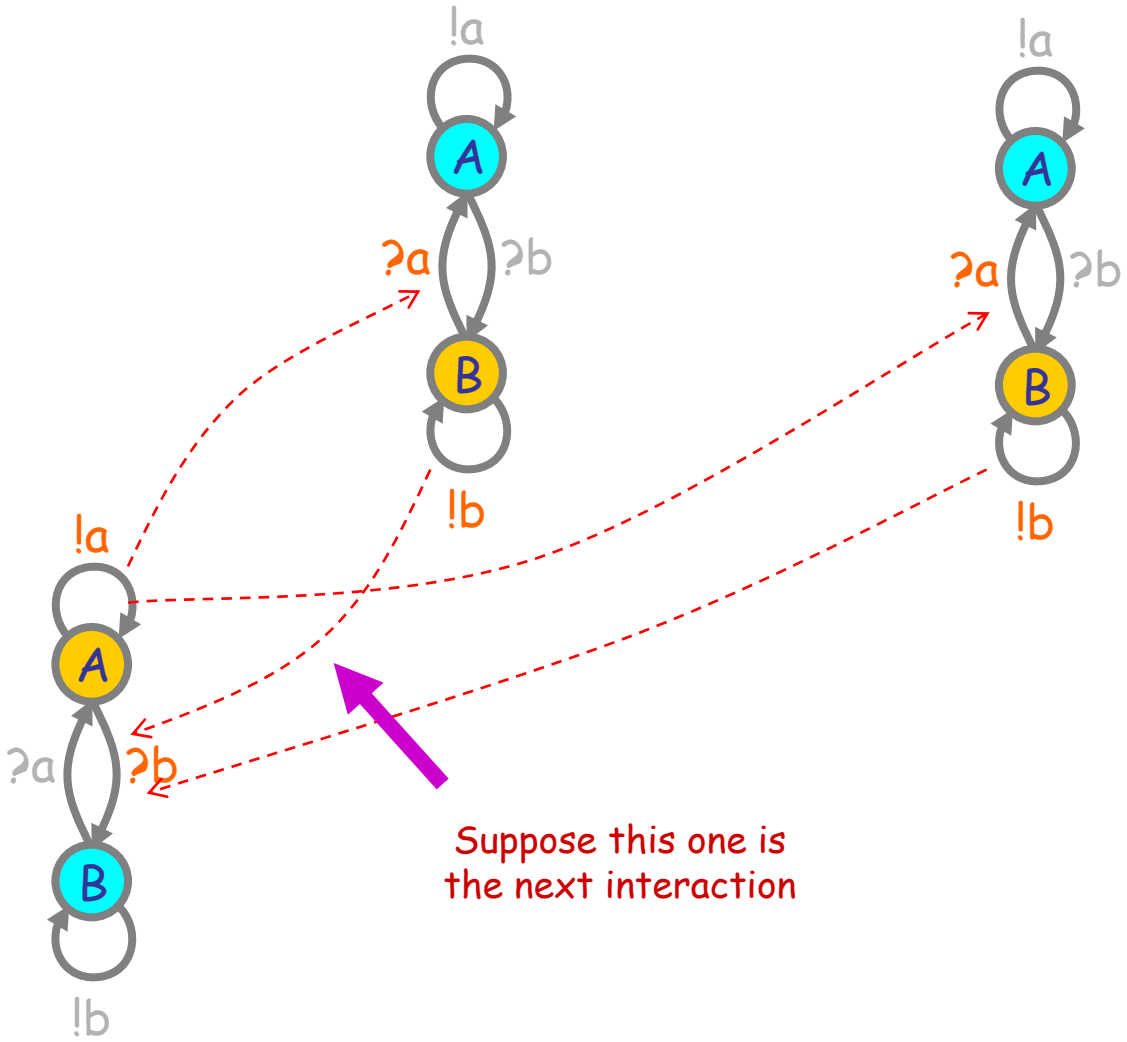
Interactions in a Population



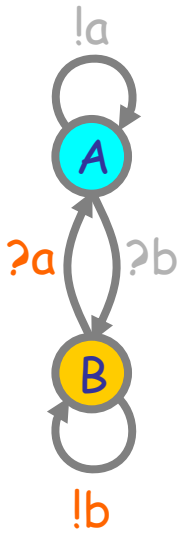


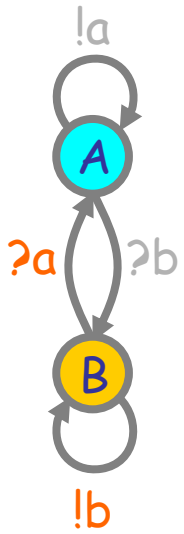


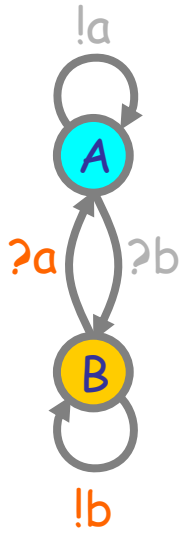
Interactions in a Population (2)



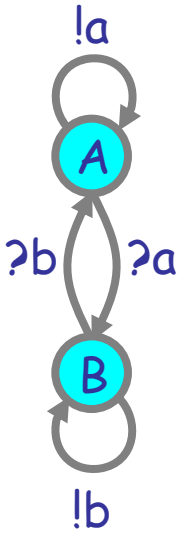
Interactions in a Population (2)







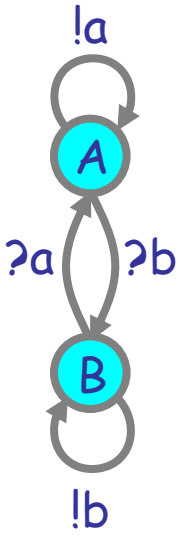
Groupies and Celebrities



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())
```

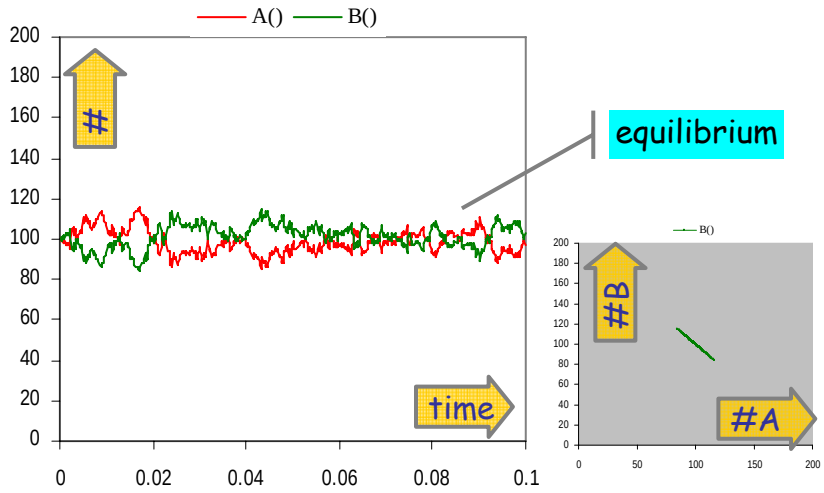


Groupie

(wants to be like somebody different)

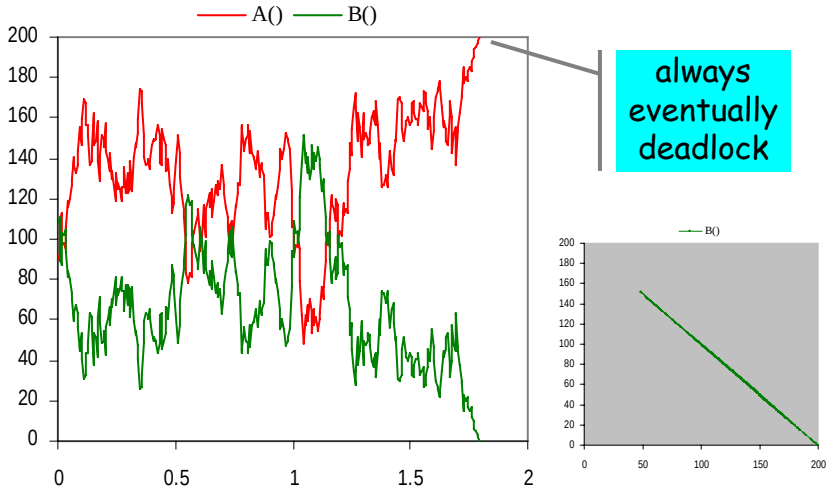
```
directive sample 0.1 200
directive plot A(); B()

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:



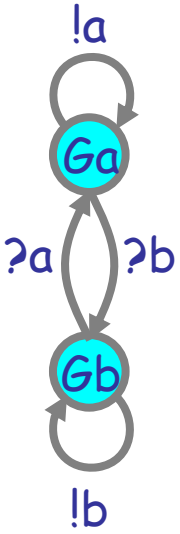
Unstable because within an A majority, an A has difficulty finding a B to emulate, but the few B's have plenty of A's to emulate, so the majority may switch to B. Leads to deadlock when everybody is in the same state and there is nobody different to emulate.

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

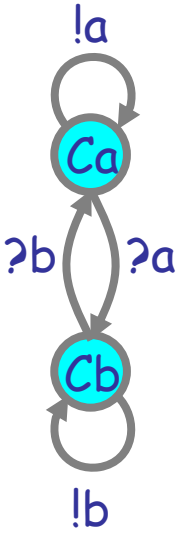
Both Together

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

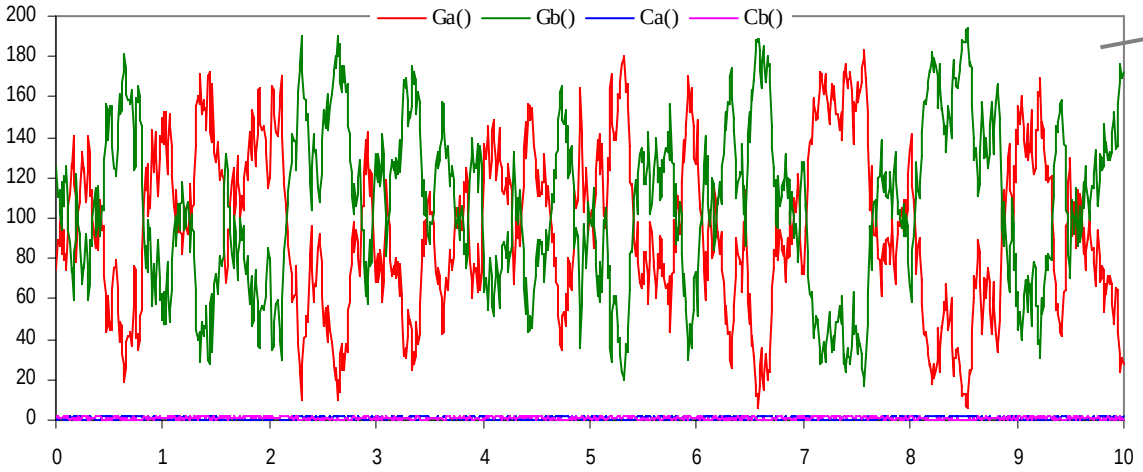
Many Groupies



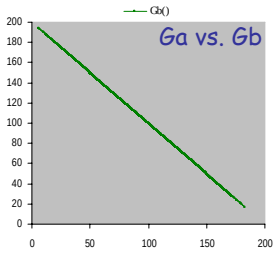
A few Celebrities




```
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())
```



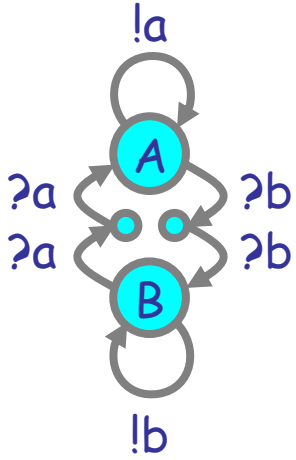
never deadlock

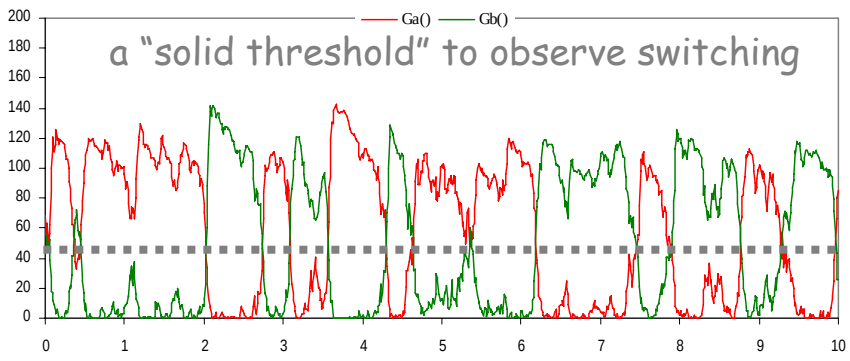


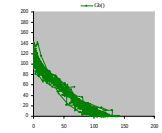
Regularity can arise not far from chaos

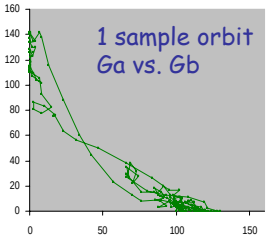
Hysteric Groupies

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

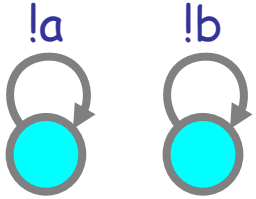








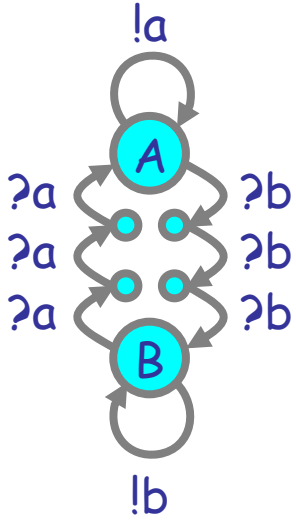

```
directive sample 10.0 1000
directive plot Ga(); Gb()

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())
```

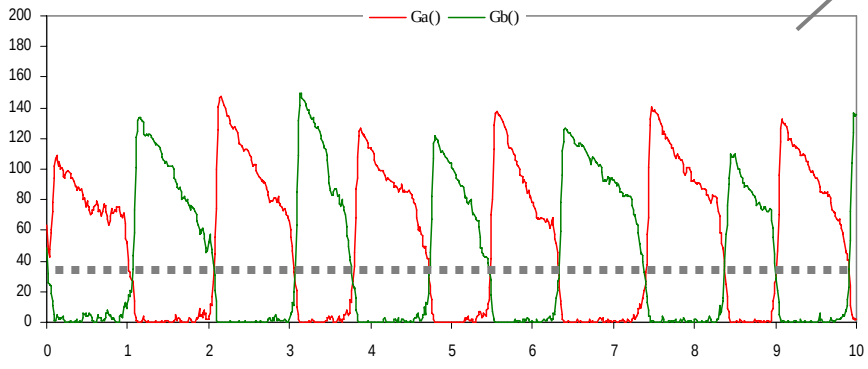


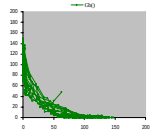
(With doping to break deadlocks)

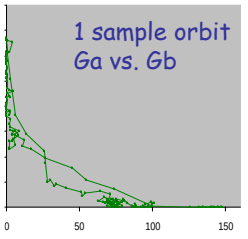
N.B.: It will not oscillate without doping (noise)

"regular" oscillation



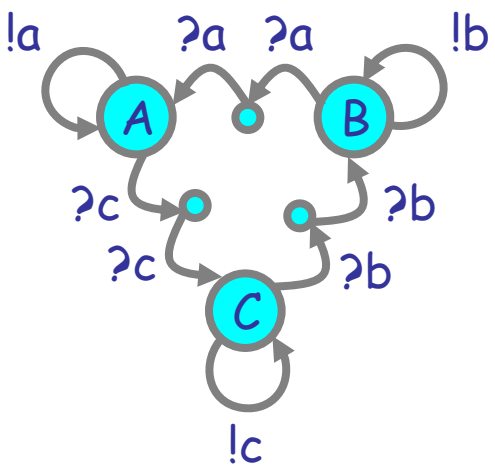


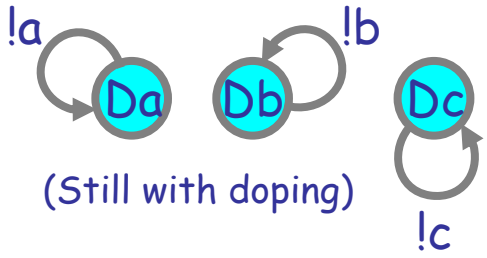





```
directive sample 10.0 1000
directive plot Ga(); Gb()

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())
```

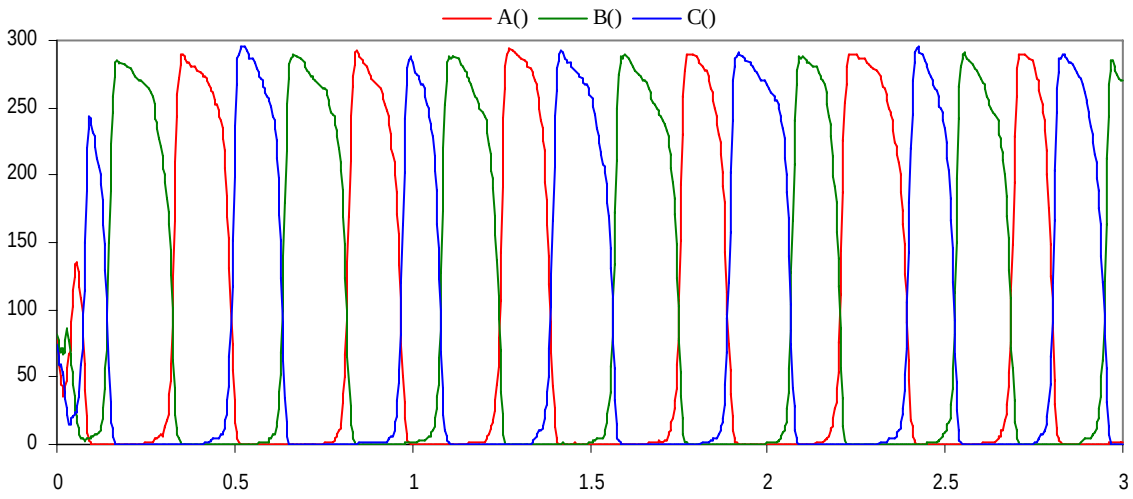
Hysteric 3-Way Groupies

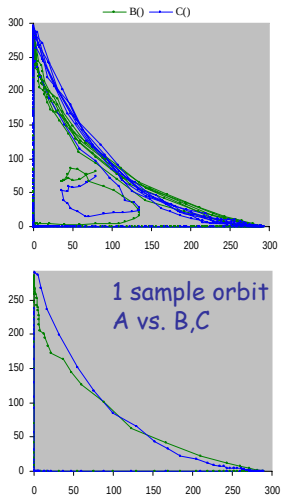





```
directive sample 3.0 1000
directive plot A(); B(); C()

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)





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!

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

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

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

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

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

Write the coefficients by columns

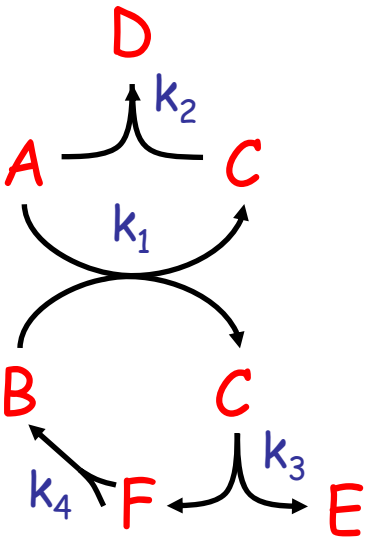
reactions

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

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

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

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

Read the concentration changes from the rows

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

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

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

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

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

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

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

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 $(n \geq 0)$

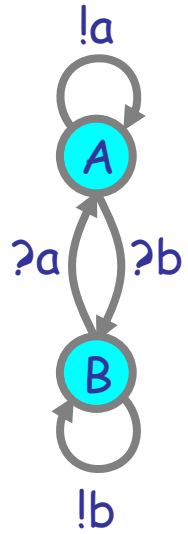
$P ::= X_1 \mid \ldots \mid X_n$ Solutions $(n \geq 0)$

$\pi ::= \tau_r \;\; ?n_{(r)} \;\; !n_{(r)}$ 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):

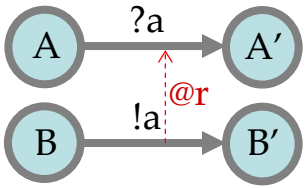
$A = !a;A \oplus ?b;B$ — Automaton in state A

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

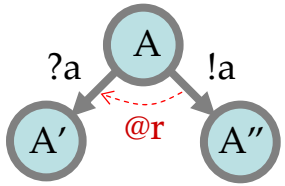
$A|A|B|B$ — Initial conditions: 2A and 2B

Automata to Chemistry

$A \rightarrow^{r} A'$



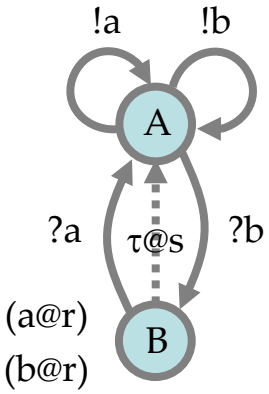
$A+B \rightarrow^{r} A'+B'$



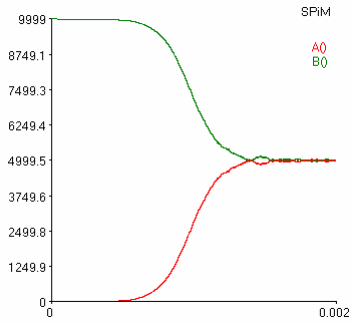
$A+A \rightarrow^{2r} A'+A''$

Process Rate Semantics

Same Chemistry




```
directive sample 0.002 10000
directive plot A(); B()

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

let A() = do !a; A() or !b; A() or ?b; B()
and B() = do delay@1.0; A() or ?a; A()

run 10000 of  B()
```

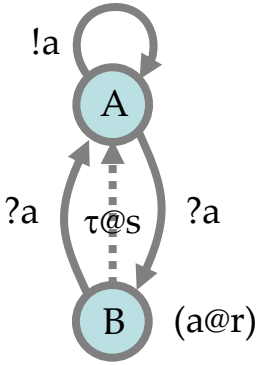


$B \rightarrow^{s} A$

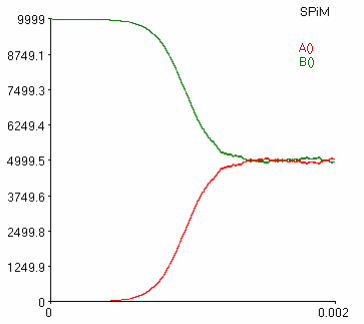
$A+B \rightarrow^{r} A+A$

$A+A \rightarrow^{2r} A+B$

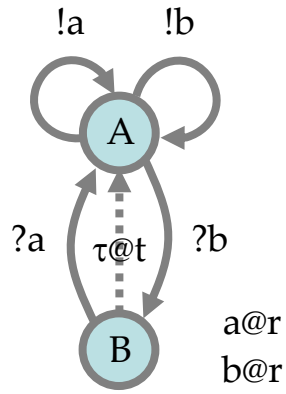
Same chemistry, hence equivalent automata




```
directive sample 0.002 10000
directive plot A(); B()

new a@1.0:chan()

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

run 10000 of  B()
```



Same ODEs

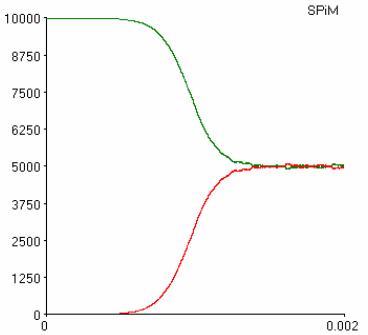


τ: B $\rightarrow^{t}$ A

a: A+B $\rightarrow^{r}$ A+A

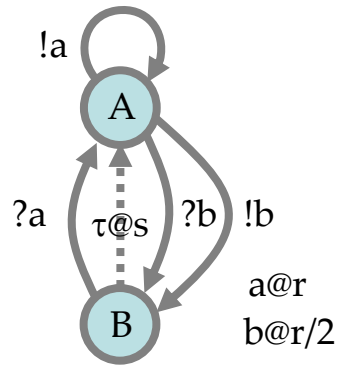
b: A+A $\rightarrow^{2r}$ A+B

```
directive sample 0.002 10000
directive plot A(); B()

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

let A() = do !a; A() or !b; A() or ?b; B()
and B() = do delay@1.0; A() or ?a; A()

run 10000 of  B()
```



$[A]^{\bullet} = t[B] + r[A][B] - r[A]([A]-1)$

$[B]^{\bullet} = -t[B] - r[A][B] + r[A]([A]-1)$

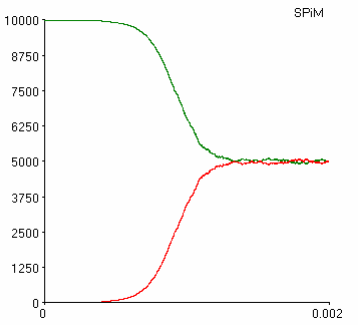


τ: B $\rightarrow^{s}$ A

a: A+B $\rightarrow^{r}$ A+A

b: A+A $\rightarrow^{r}$ B+B

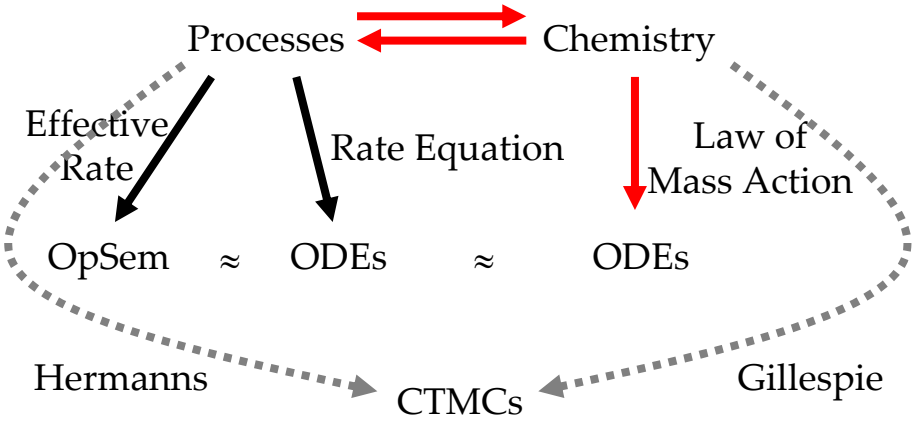
```
directive sample 0.002 10000
directive plot A(); B()

new a@1.0:chan()
new b@0.5:chan()

let A() = do !a; A() or !b; B() or ?b; B()
and B() = do delay@1.0; A() or ?a; A()

run 10000 of  B()
```



$[A]^{\bullet} = t[B] + r[A][B] - r[A]([A]-1)$

$[B]^{\bullet} = -t[B] - r[A][B] + r[A]([A]-1)$

Semantic Relationships

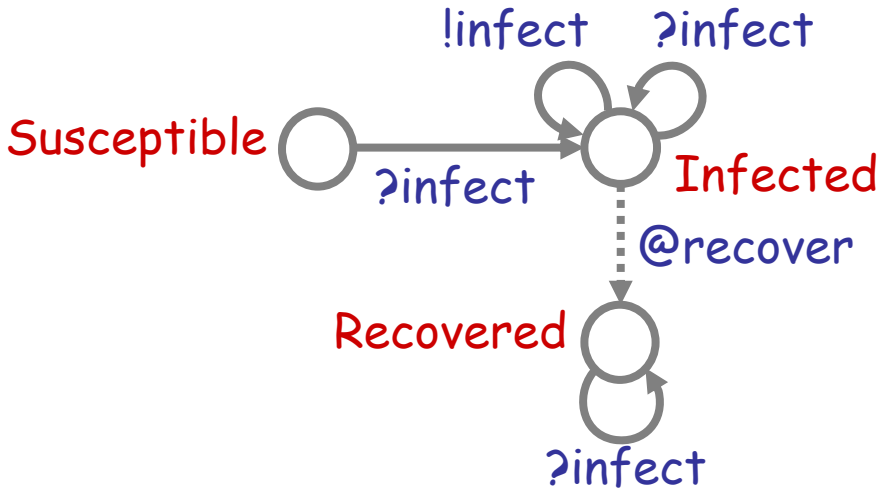


Epidemics

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



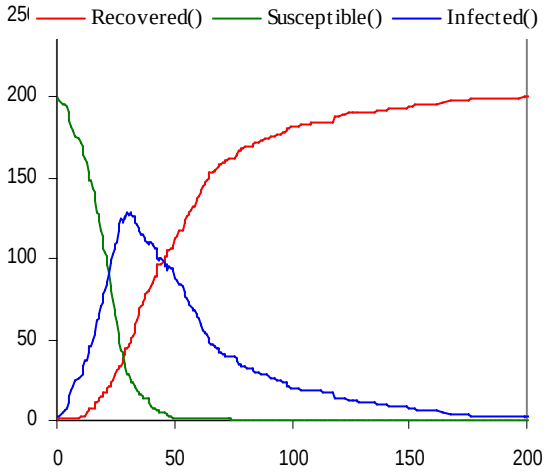

```
directive sample 500.0 1000
directive plot Recovered(); Susceptible(); Infected()

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

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

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

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

Differentiating Processes!

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

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

$I \rightarrow^{r} R$

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

"useless" reactions

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

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

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

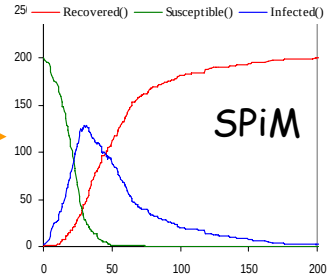
Automata match the standard ODE model!

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

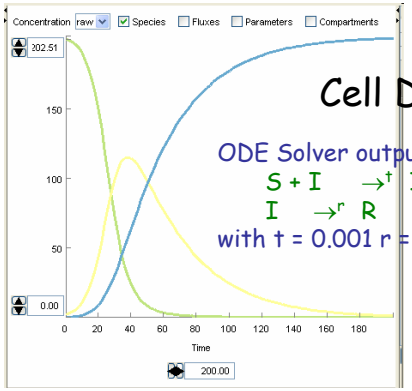
$$\frac{dI}{dt} = aIS - bI$$

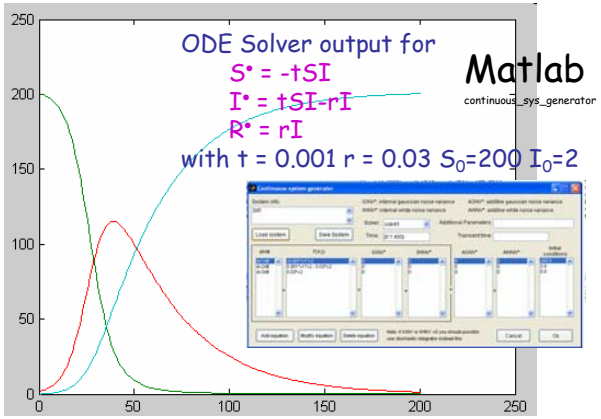
$$\frac{dR}{dt} = bI$$

(the Kermack-McKendrick, or SIR model)

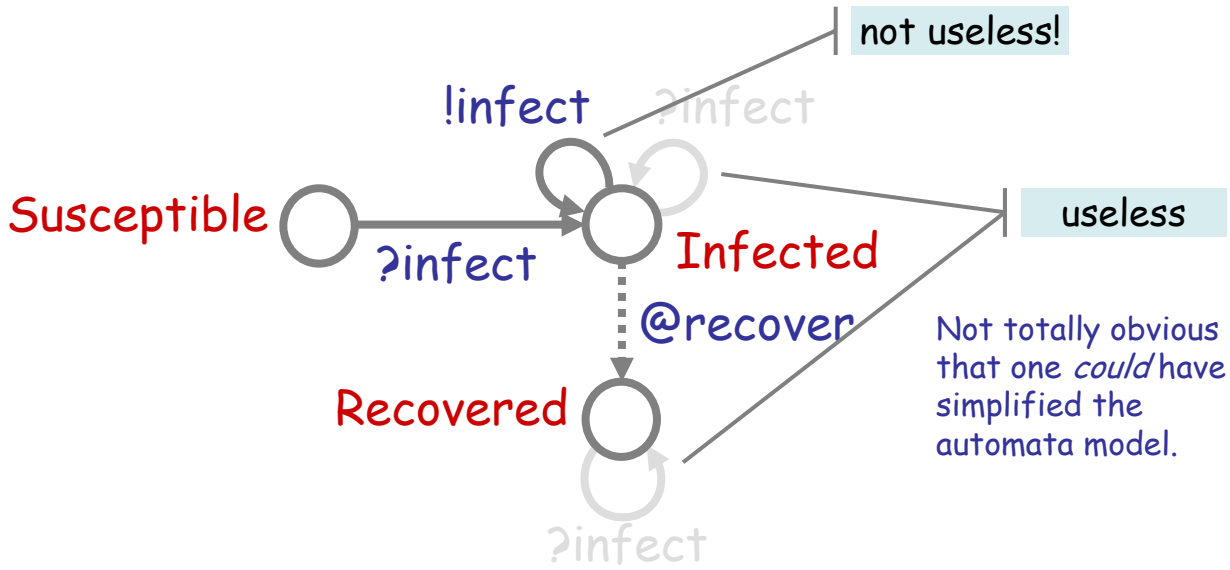



```
directive sample 500.0 1000
directive plot Recovered(); Susceptible(); Infected()

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())
```





Simplified Model



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

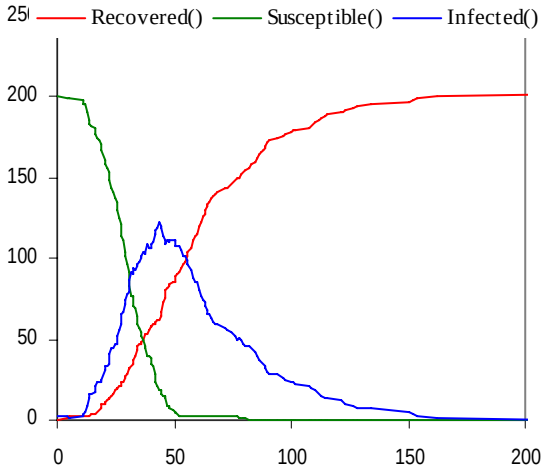
$$S = ?i_{(t)};I$$
$$I = !i_{(t)};I \oplus \tau_r;R$$
$$R = 0$$

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

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

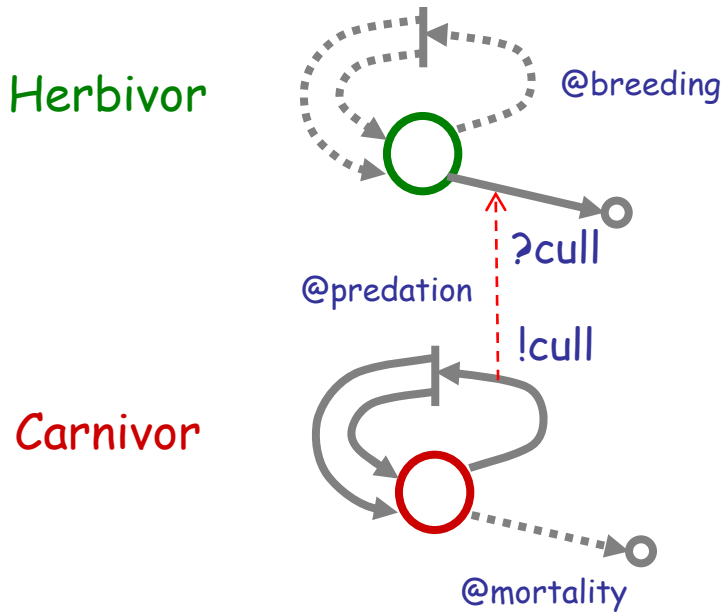
Same ODE, hence equivalent automata models.

```
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())
```

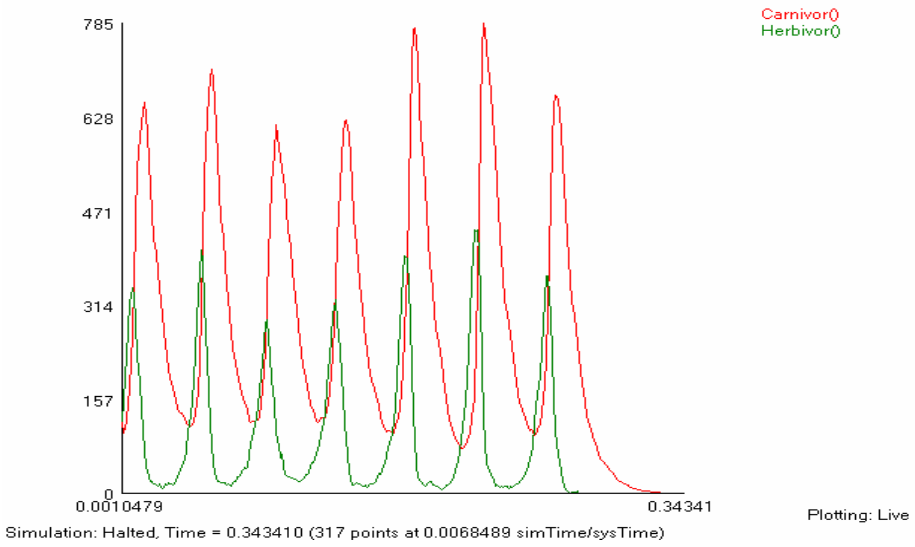


Lotka-Volterra

Predator-Prey




```
directive sample 1.0 1000
directive plot Carnivor(); Herbivor()

val mortality = 100.0
val breeding = 300.0
val predation = 1.0
new cull @predation:chan()

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

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

run 100 of Herbivor()
run 100 of Carnivor()
```



An *unbounded state* system!

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

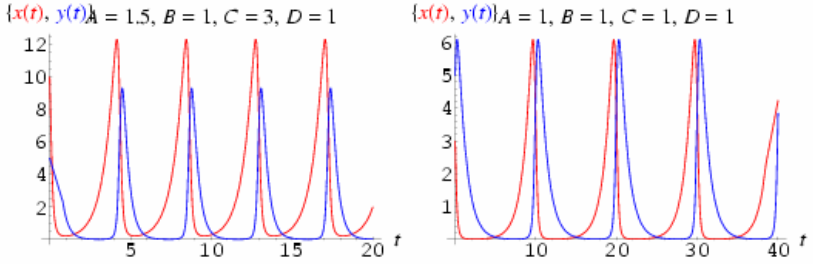
mathworld

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

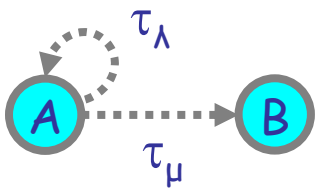
$$\frac{dx}{dt} = Ax - Bxy \quad (1)$$
$$\frac{dy}{dt} = -Cy + Dxy, \quad (2)$$

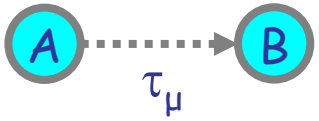
Automata match the Lotka-Volterra model (with B=D)

Laws by ODEs

Idle Delay Law by ODEs

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





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

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

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

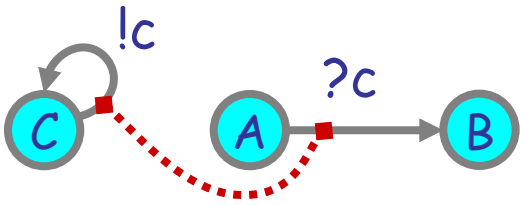
=

$A = \tau_\mu;B$

$A \rightarrow^{\mu} B$

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

Idle Interaction Law by ODEs

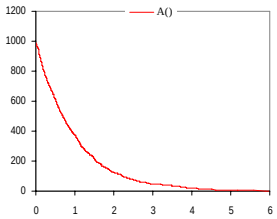


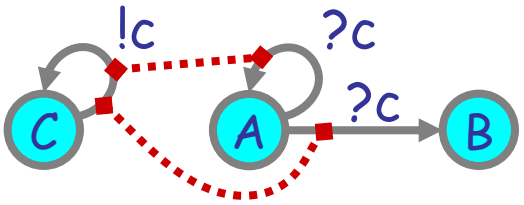
$A = ?c;B$
$C = !c;C$

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

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

```
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())
```





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

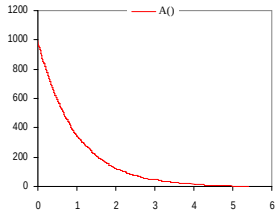
$A+C \rightarrow^{r} A+C$
$A+C \rightarrow^{r} B+C$

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

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

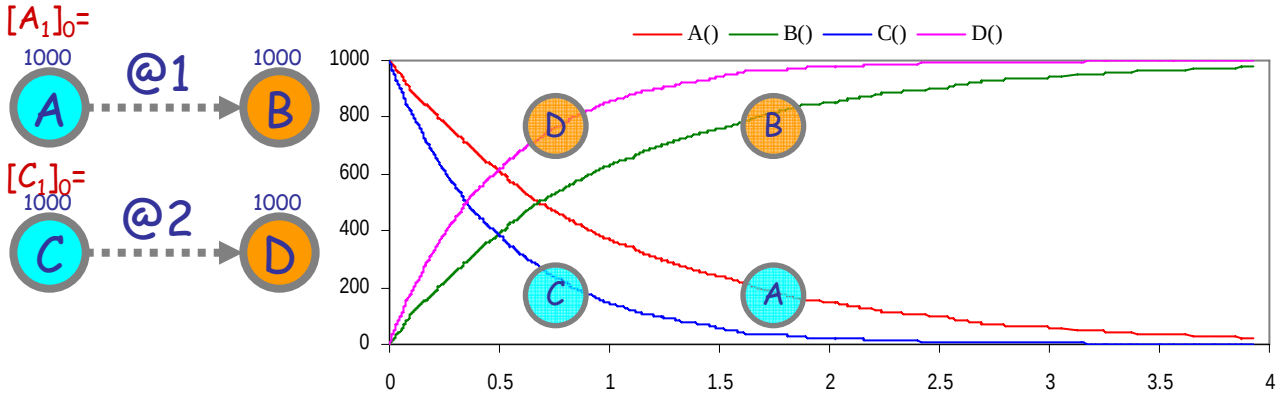
```
directive sample 6.0 1000
directive plot A()

new c@1.0:chan

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

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



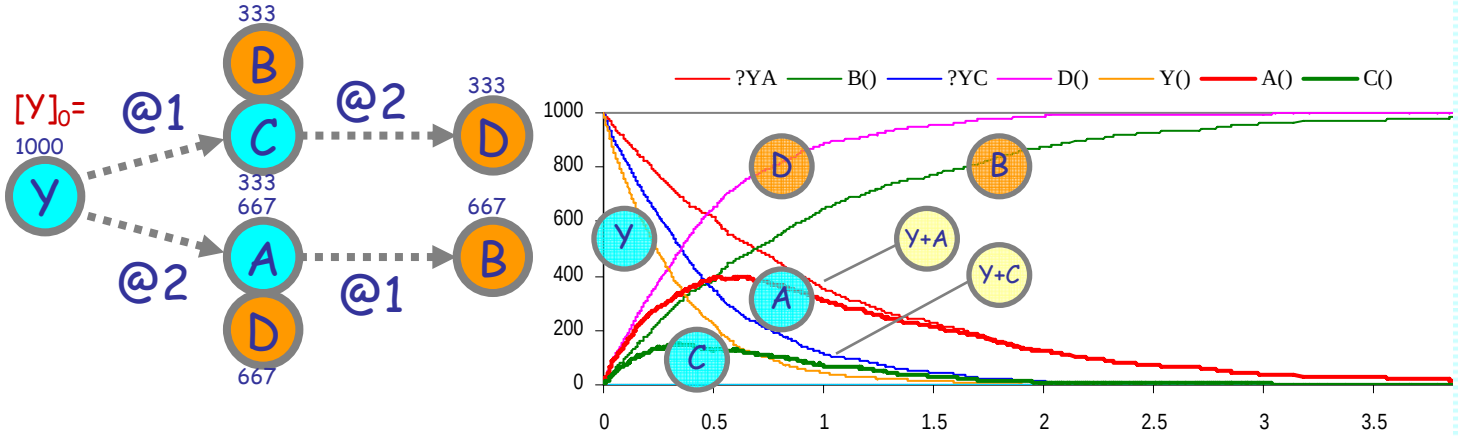
Asynchronous Interleaving

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.

Polymerization

Bidirectional Polymerization

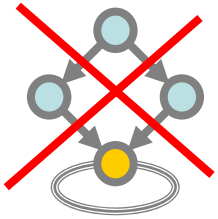
new c@μ new stop@1.0

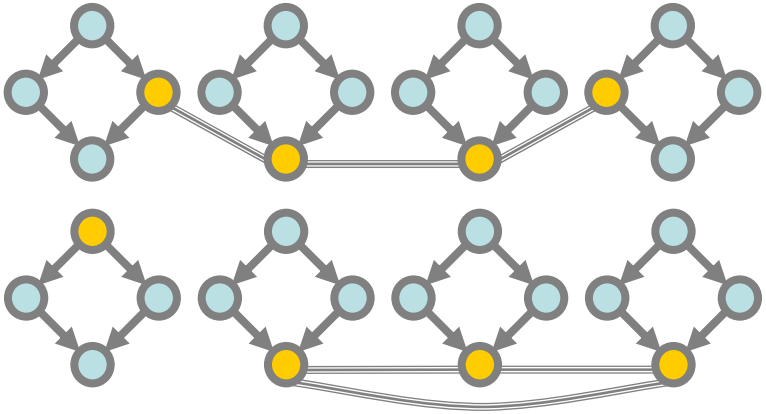
A_{free} =
  !c(ν rht$_\lambda$); A_{brht}(rht)) +
  ?c(lft); A_{blft}(lft)

A_{blft}(lft) =
  !c(ν rht$_\lambda$); A_{bound}(lft,rht))

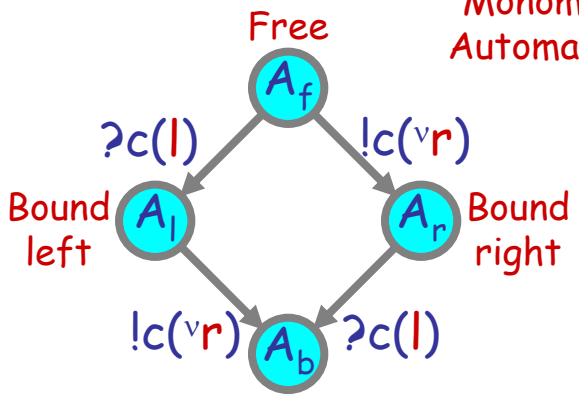
A_{brht}(rht) =
  ?c(lft); A_{bound}(lft,rht)

A_{bound}(lft,rht) = ?stop



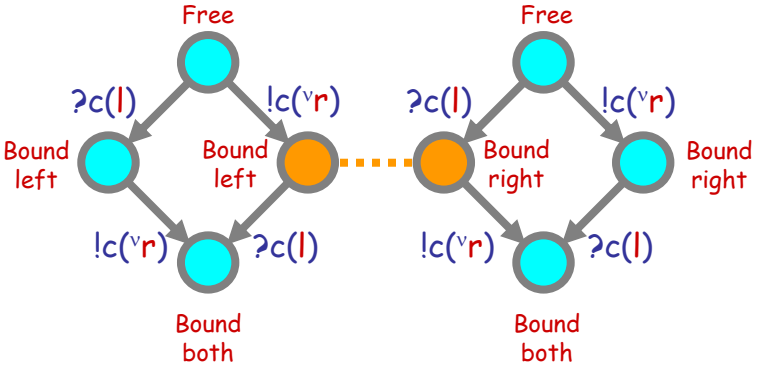


Monomer Automaton



Polymerization is iterated complexation.

Communicating Automata
Bound output !c(ν r) and input ?c(l) on automata transitions to model complexation

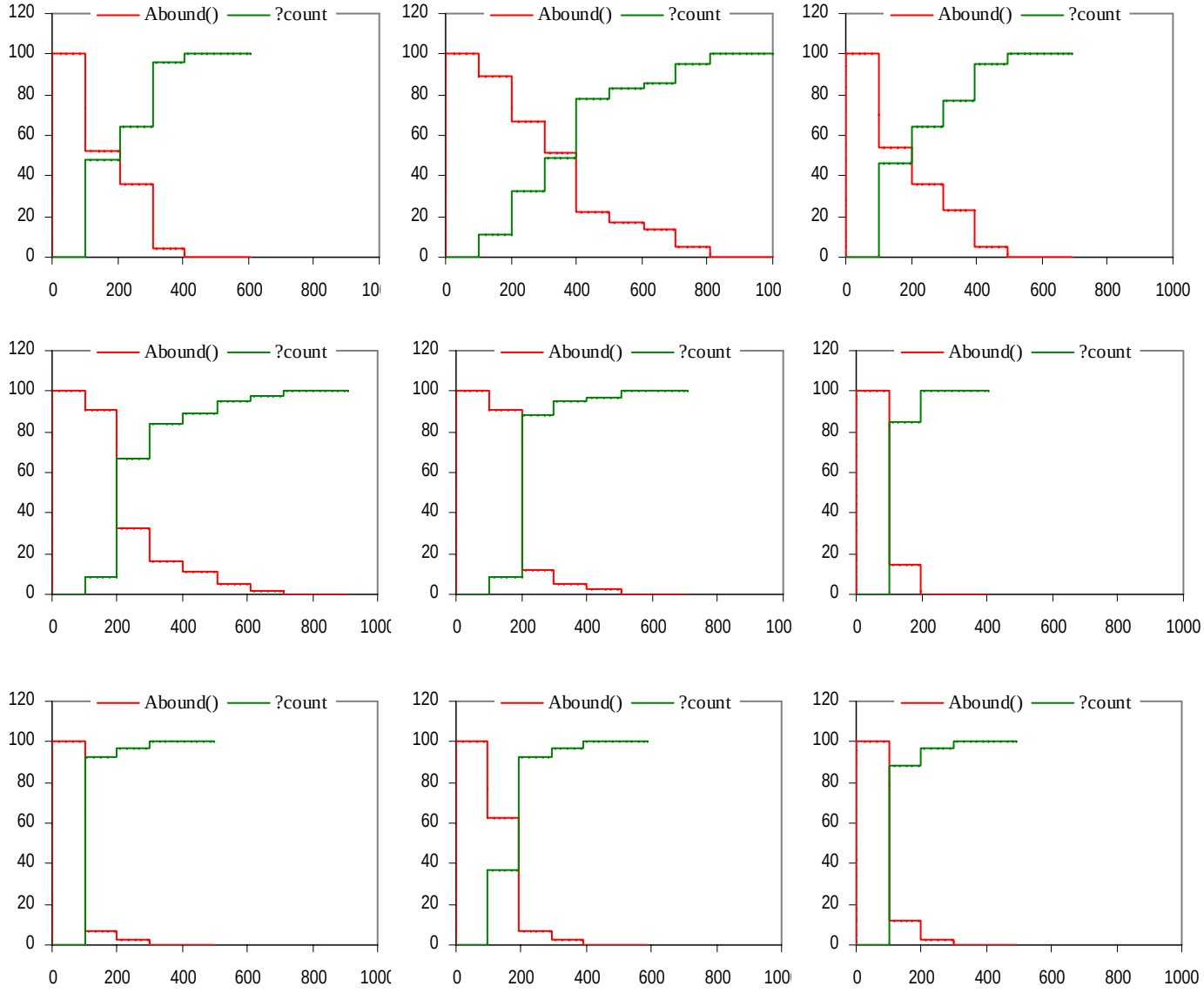



```
directive sample 10000.0
directive plot Afree(); Ablft(); Abrht(); Abound()

val lam = 1.0  val mu = 1.0
new c@mu:chan(chan)  new stop@1.0:chan

let Afree() =
  (new rht@lam:chan run
   do !c(rht); Abrht(rht)
   or ?c(lft); Ablft(lft))

and Ablft(lft:chan) =
  (new rht@lam:chan run
   !c(rht); Abound(lft,rht))

and Abrht(rht:chan) =
  ?c(lft); Abound(lft,rht)

and Abound(lft:chan, rht:chan) =
  ?stop

run (2 of Afree())
```

Bidirectional Polymerization

Circular Polymer Lengths

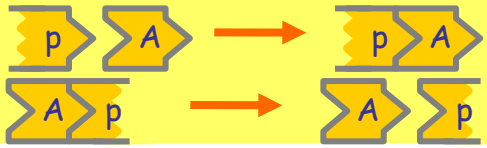
Scanning and counting the size of the circular polymers (by a cheap trick).
Polymer formation is complete within 10t; then a different polymer is scanned every 100t.




```
directive sample 1000.0
directive plot Abound(); ?count

type Link = chan(chan)
type Barb = chan

val lam = 1000.0 (* set high for better counting *)
val mu = 1.0
new c@mu:chan(Link)
new enter@lam:chan(Barb)
new count@lam:Barb

let Afree() =
  (new rht@lam:Link run
   do !c(rht); Abrht(rht)
   or ?c(lft); Ablft(lft))

and Ablft(lft:Link) =
  (new rht@lam:Link run
   !c(rht); Abound(lft,rht))

and Abrht(rht:Link) =
  ?c(lft); Abound(lft,rht)

and Abound(lft:Link, rht:Link) =
  do ?enter(barb); (?barb | !rht(barb))
  or ?lft(barb); (?barb | !rht(barb))
(* each Abound waits for a barb, exhibits it, and passes it to
   the right so we can plot number of Abound in a ring *)

let clock(t:float, tick:chan) =      (* sends a tick every t time *)
  (val ti = t/1000.0 val d = 1.0/ti
   let step(n:int) =
     if n<=0 then !tick; clock(t,tick) else delay@d; step(n-1)
   run step(1000))

new tick:chan
let Scan() = ?tick; !enter(count); Scan()

run 100 of Afree()
run (clock(100.0, tick) | Scan())
```

$100 \times A_{free}$, initially.

The height of each rising step is the size of a separate circular polymer. (Unbiased sample of nine consecutive runs.)



Actin-like
Poly/Depolymerization

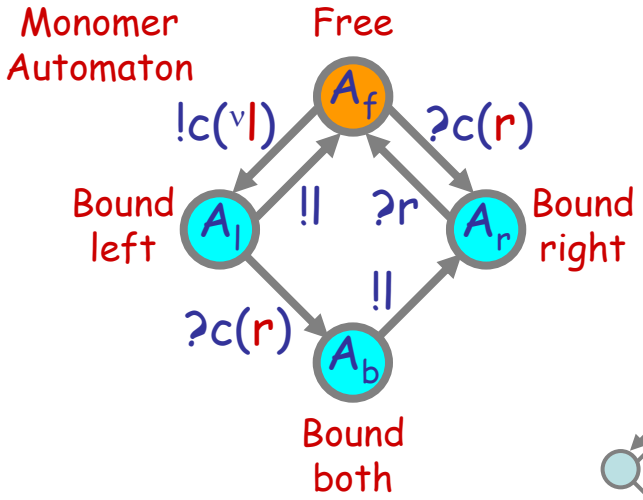
new c@μ

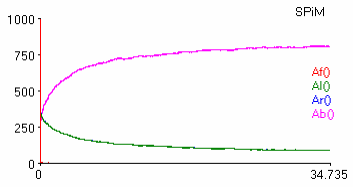
A_{free} =
  !c(ν lft$_\lambda$); A_{blft}(lft)) +
  ?c(rht); A_{brht}(rht)

A_{blft}(lft) =
  !lft; A_{free} +
  ?c(rht); A_{bound}(lft,rht)

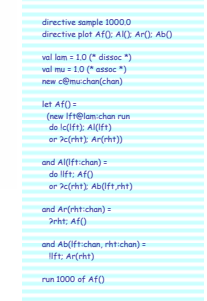
A_{brht}(rht) =
  ?rht; A_{free}

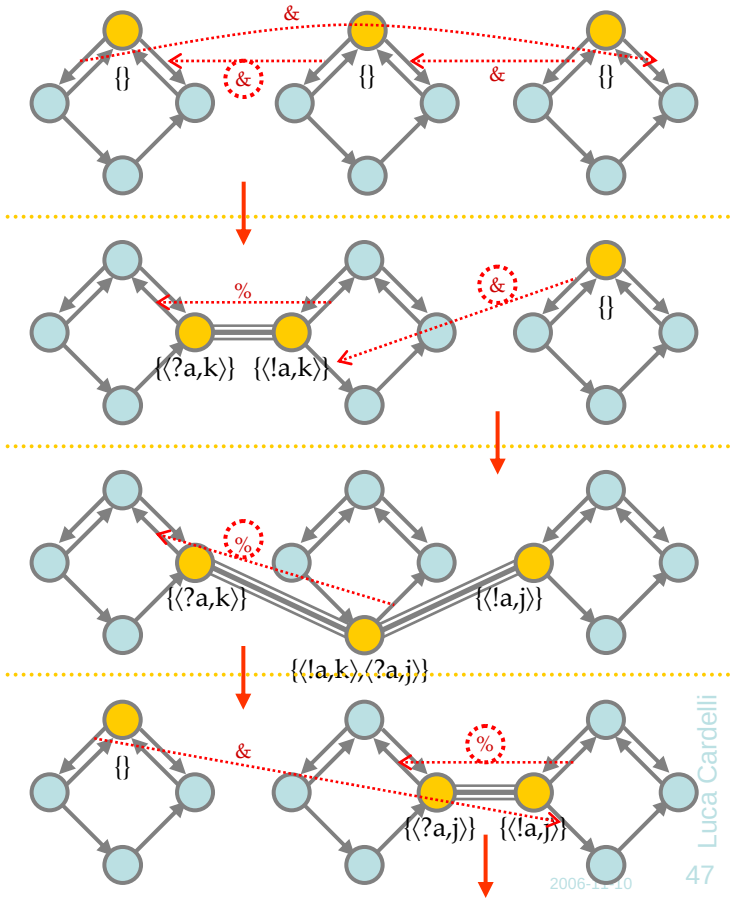
A_{bound}(lft,rht) =
  !lft; A_{brht}(rht)

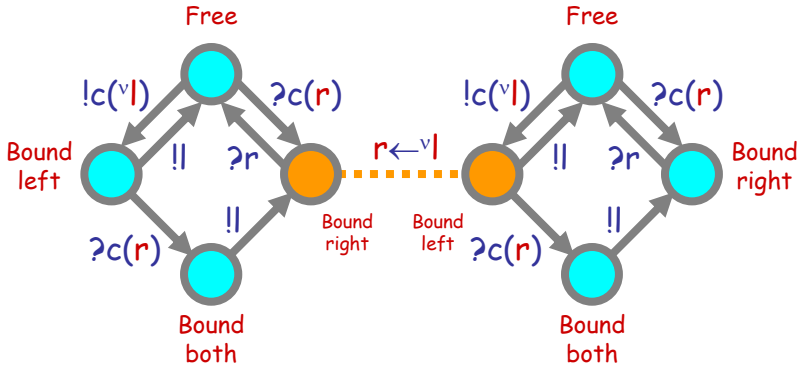




1000 monomers settle to
~100 polymers of size ~10







Conclusions

Conclusions

- Compositional Models
  - Accurate (at the "appropriate" abstraction level).
  - Manageable (so we can scale them up by composition).

- Interacting Automata
  - Complex global behavior from simple components.
  - Bridging individual and collective behavior.
  - Connections to classical Markov theory, chemical Master Equation, and Rate Equation.

- Mapping out "the whole system"
  - A bit at a time, and simultaneously at different levels.
  - For prediction and prevention.
  - Through an "artificial biochemistry" (a scalable mathematical and computational modeling framework) to investigate "real biochemistry" on a large scale.