# Artificial Biochemistry

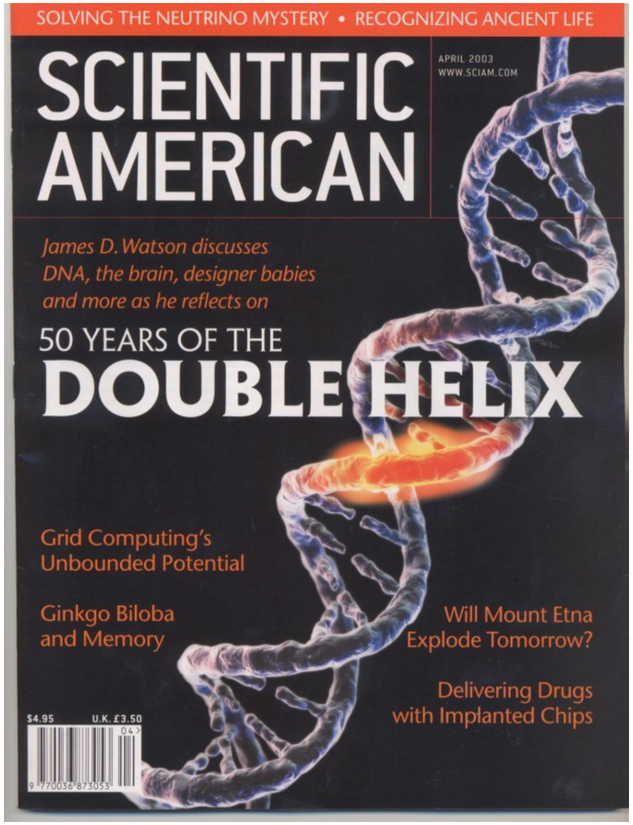
Luca Cardelli

Microsoft Research

MSRC 10th Anniversary
Cambridge, 2007-07-09

# 50 Years of Molecular Cell Biology

- The genome (human: 3 GBases = 750MB) is made of DNA
  - Stores digital information as sequences of 4 different nucleotides
  - Directs protein assembly through RNA and the Genetic Code
- Proteins (~1M coded from ~30K genes) are made of amino acids strings
  - Catalyze all biochemical reactions
  - Control metabolism (energy & materials)
  - Process signals, activate genes
- Bootstrapping still a mystery
  - DNA, RNA, proteins, membranes are today interdependent. Not clear who came first
  - Not understood, not essential for us


SOLVING THE NEUTRINO MYSTERY • RECOGNIZING ANCIENT LIFE
SCIENTIFIC AMERICAN
APRIL 2003
WWW.SCIAM.COM
James D. Watson discusses DNA, the brain, designer babies and more as he reflects on
50 YEARS OF THE DOUBLE HELIX
Grid Computing's Unbounded Potential
Ginkgo Biloba and Memory
Will Mount Etna Explode Tomorrow?
Delivering Drugs with Implanted Chips
$4.95
U.K. £3.50

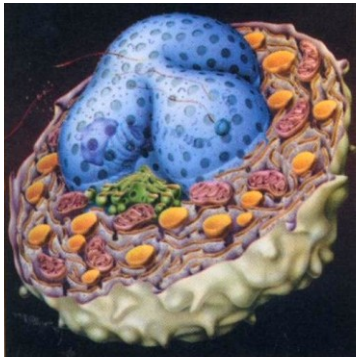
# Cells Compute

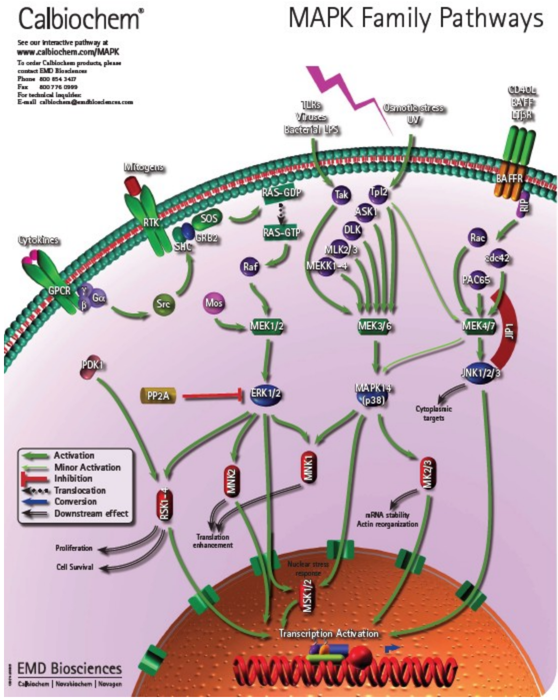
- No survival without computation!
  - Finding food
  - Avoiding predators

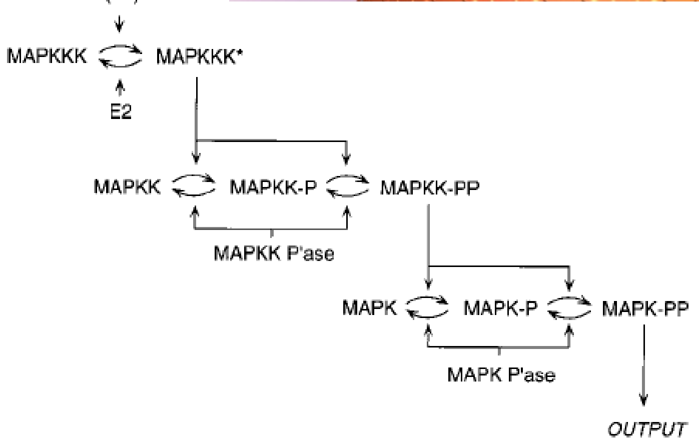
- How do they compute?
  - Unusual computational paradigms.
  - Proteins: do they work like electronic circuits? or process algebra?
  - Genes: what kind of software is that?

- Signaling networks
  - Clearly “information processing”
  - They are “just chemistry”: molecule interactions
  - But what are their principles and algorithms?

- Complex, higher-order interactions
  - MAPKKK = MAP Kinase Kinase Kinase: that which operates on that which operates on that which operates on protein.

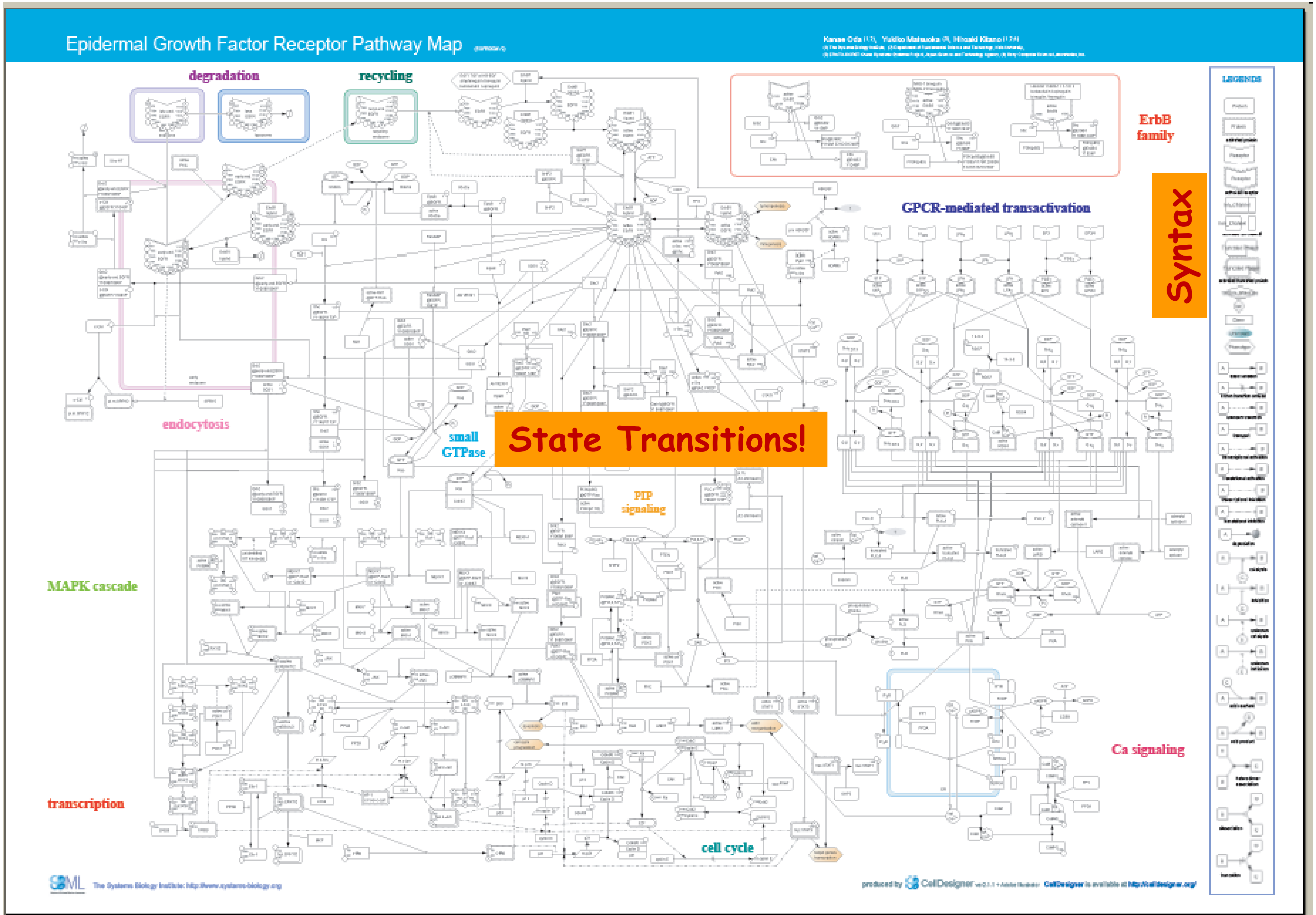






Ultrasensitivity in the mitogen-activated protein **cascade**, Chi-Ying F. Huang and James E. Ferrell, Jr., 1996, *Proc. Natl. Acad. Sci. USA*, 93, 10078-10083.

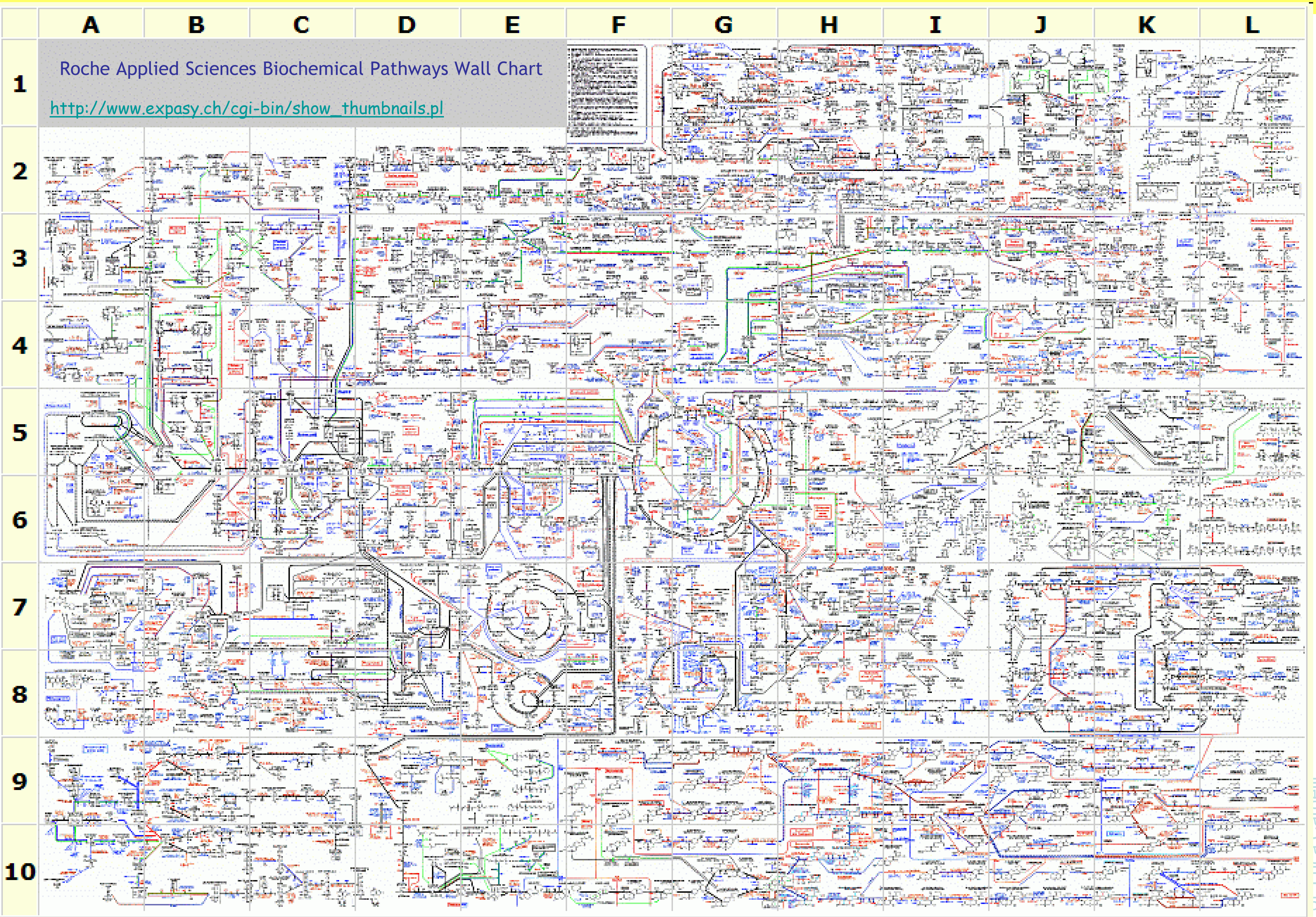
# Towards Systems Biology


Epidermal Growth Factor Receptor Pathway Map
Kanae Oda, Yukiko Matsuoka, Hiroaki Kitano
degradation
recycling
ErbB family
GPCR-mediated transactivation
LEGENDS
endocytosis
small GTPase
PIP signaling
MAPK cascade
Ca signaling
transcription
cell cycle
SBML The Systems Biology Institute: http://www.systems-biology.org
produced by CellDesigner CellDesigner is available at http://celldesigner.org/

Syntax

State Transitions!

# Compositionality (NOT!)



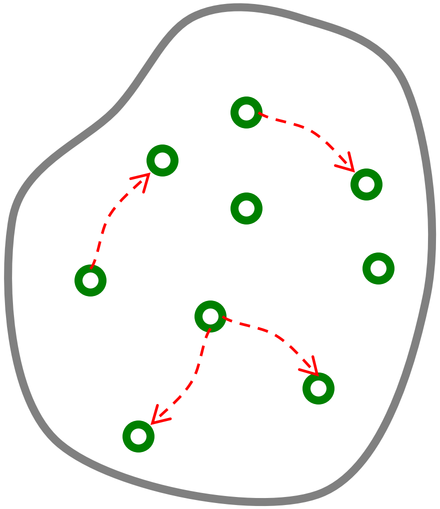
Roche Applied Sciences Biochemical Pathways Wall Chart

http://www.expasy.ch/cgi-bin/show_thumbnails.pl

# Stochastic Collectives

# Stochastic Collectives

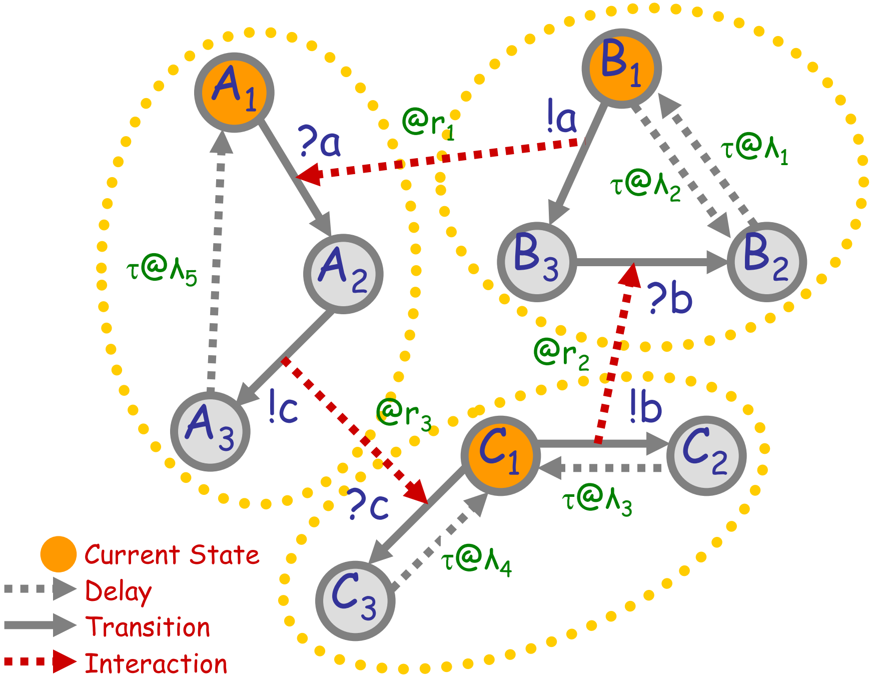
- "Collective":
  - A large set of interacting finite state automata:
    - Not quite language automata ("large set")
    - Not quite cellular automata ("interacting" but not on a grid)
    - Not quite process algebra ("collective behavior")
    - Cf. multi-agent systems and swarm intelligence



- "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].

# Interacting Automata



*Interactions* have rates. Actions DO NOT have rates.

new $a@r_1$
new $b@r_2$
new $c@r_3$

Communication channels

$A_1$ = ?a; $A_2$
$A_2$ = !c; $A_3$
$A_3$ = $\tau@\lambda_5$; $A_1$

$B_1$ = $\tau@\lambda_2$; $B_2$ + !a; $B_3$
$B_2$ = $\tau@\lambda_1$; $B_1$
$B_3$ = ?b; $B_2$

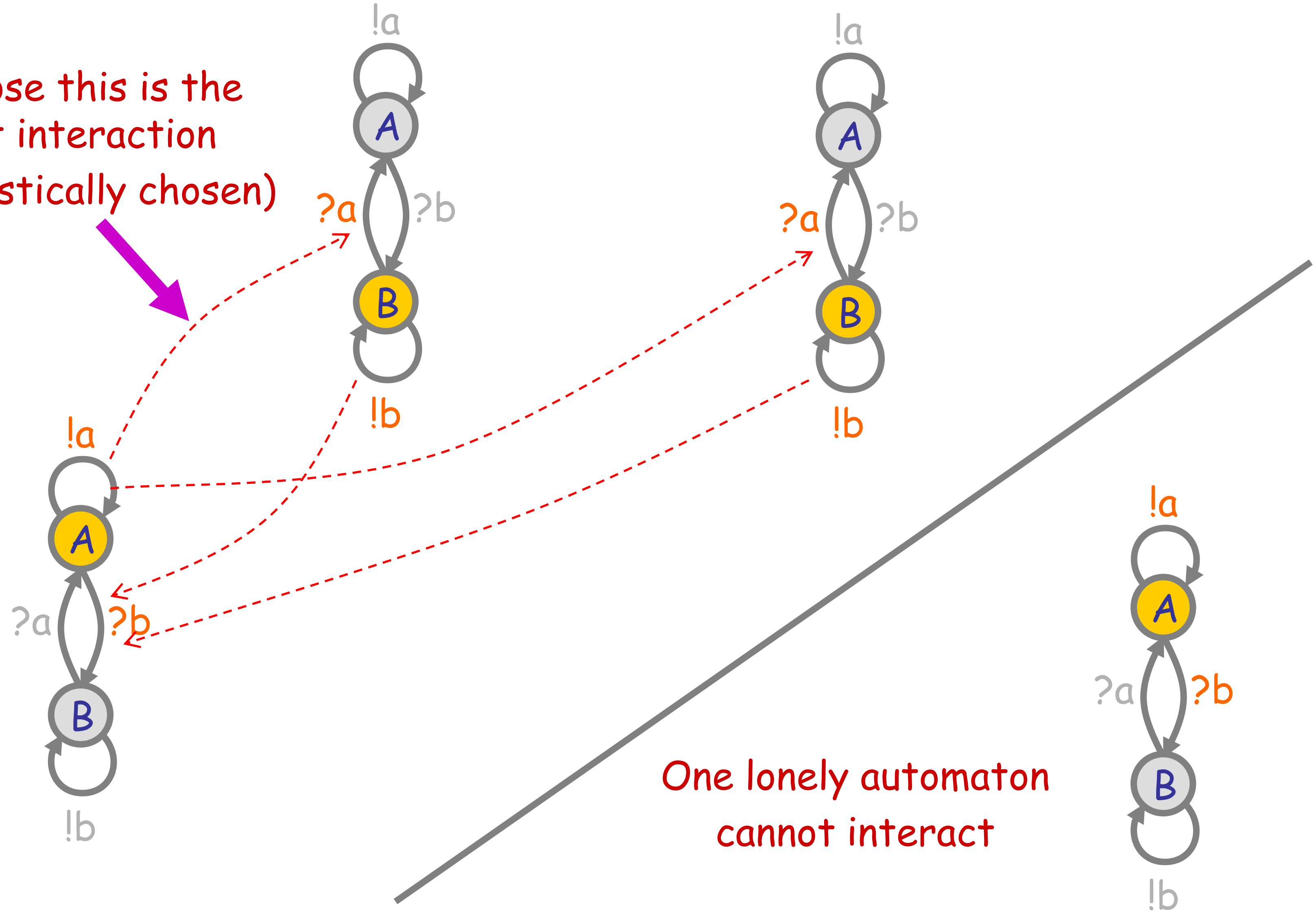
$C_1$ = !b; $C_2$ + ?c; $C_3$
$C_2$ = $\tau@\lambda_3$; $C_1$
$C_3$ = $\tau@\lambda_4$; $C_2$

Automata

$A_1$ | $B_1$ | $C_1$

The system and initial state

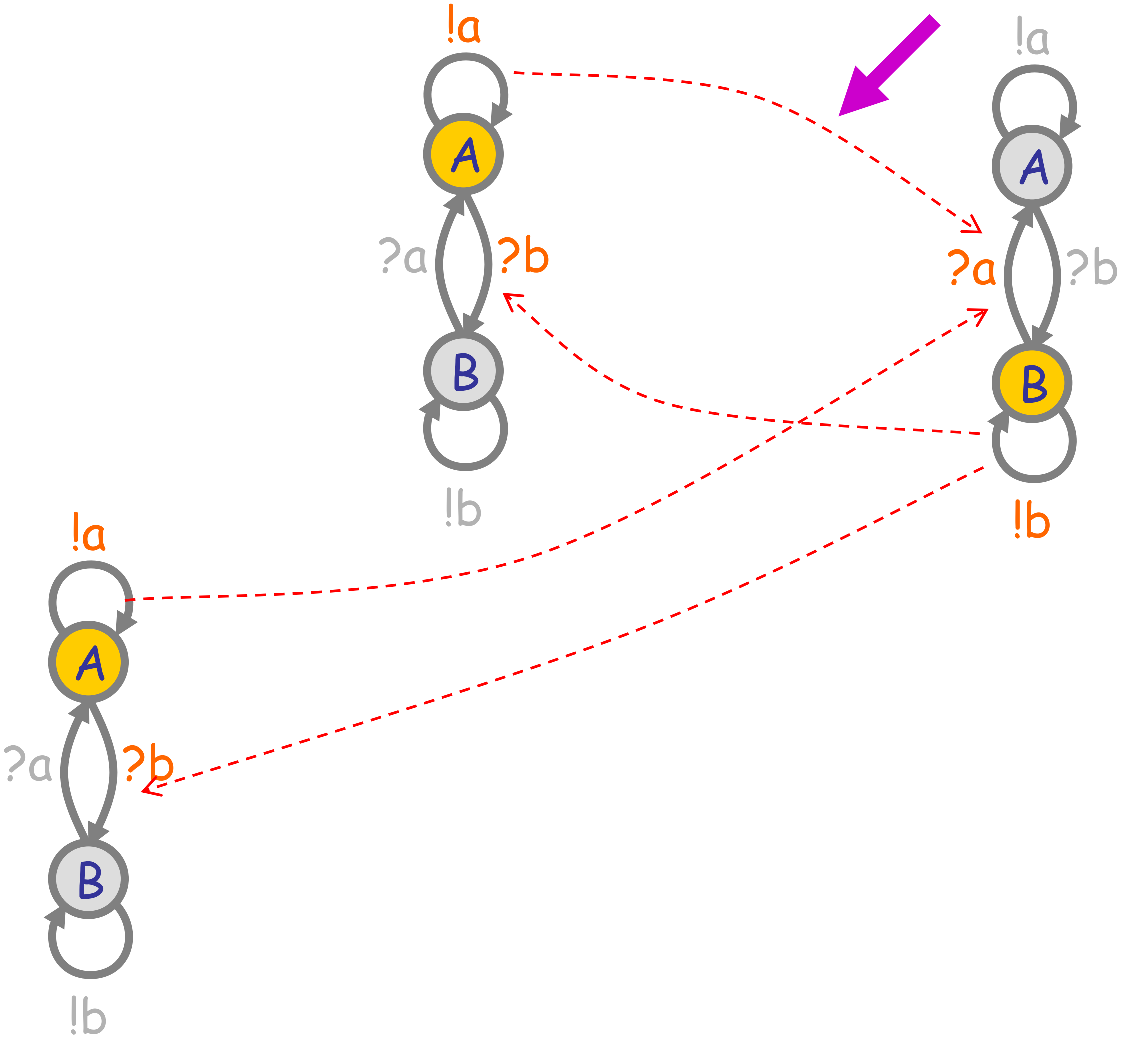
# Interactions in a Population



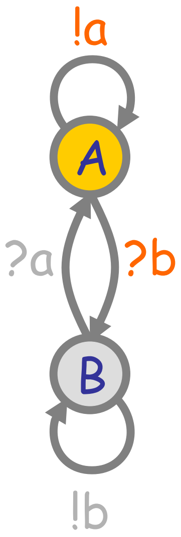
Suppose this is the next interaction (stochastically chosen)

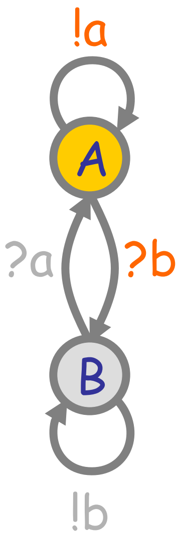
One lonely automaton cannot interact

# Interactions in a Population

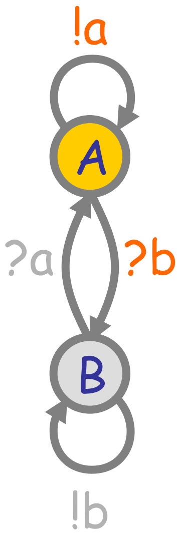


# Interactions in a Population



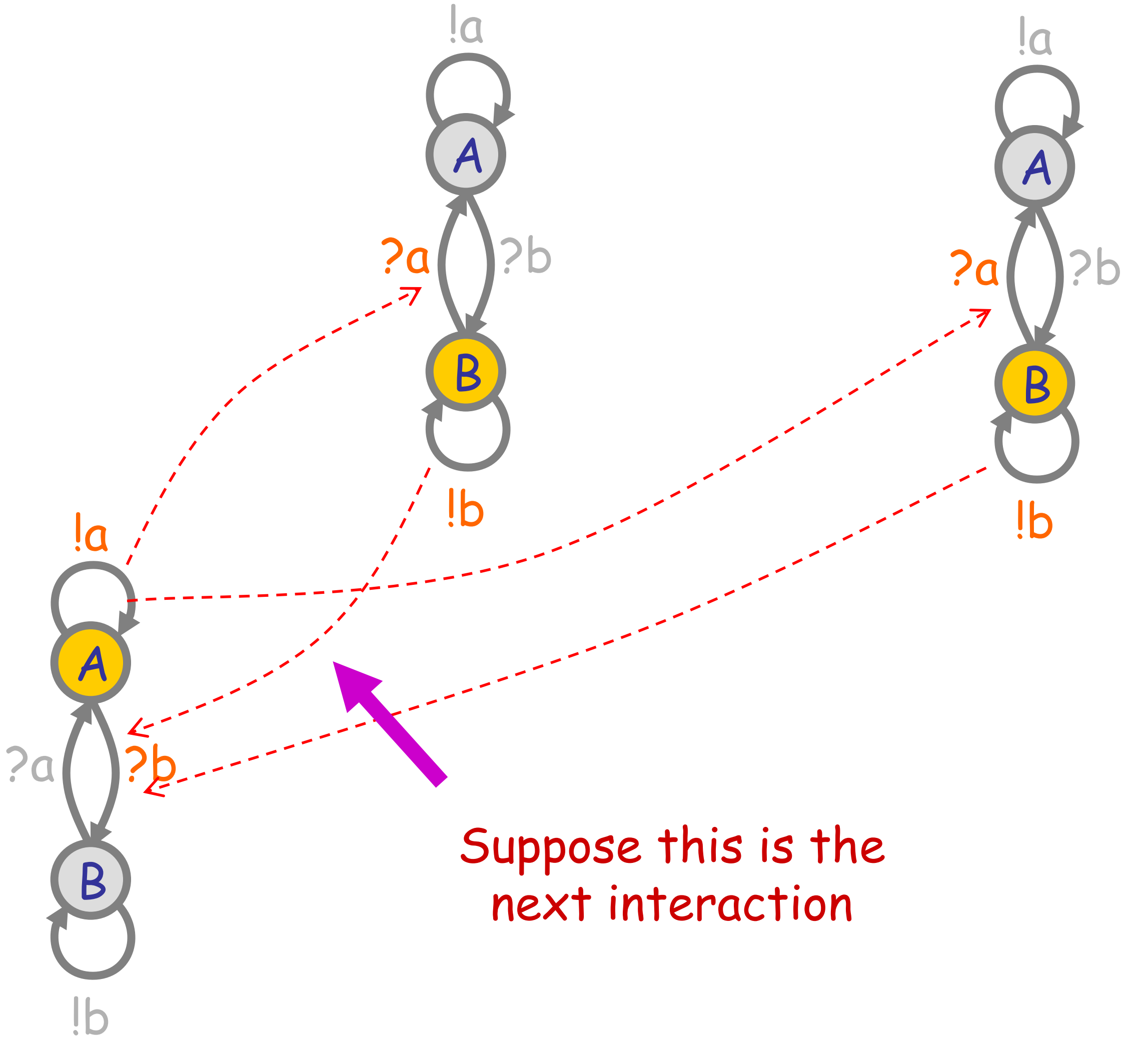


✗



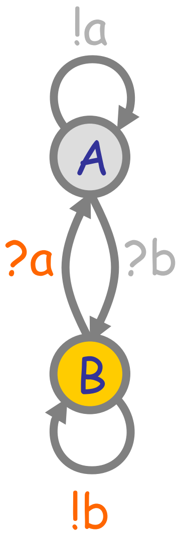
All-A stable
population

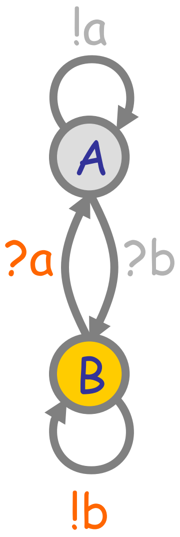
# Interactions in a Population (2)

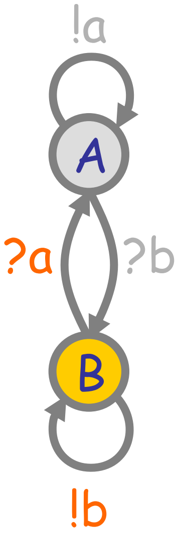


Suppose this is the next interaction

# Interactions in a Population (2)



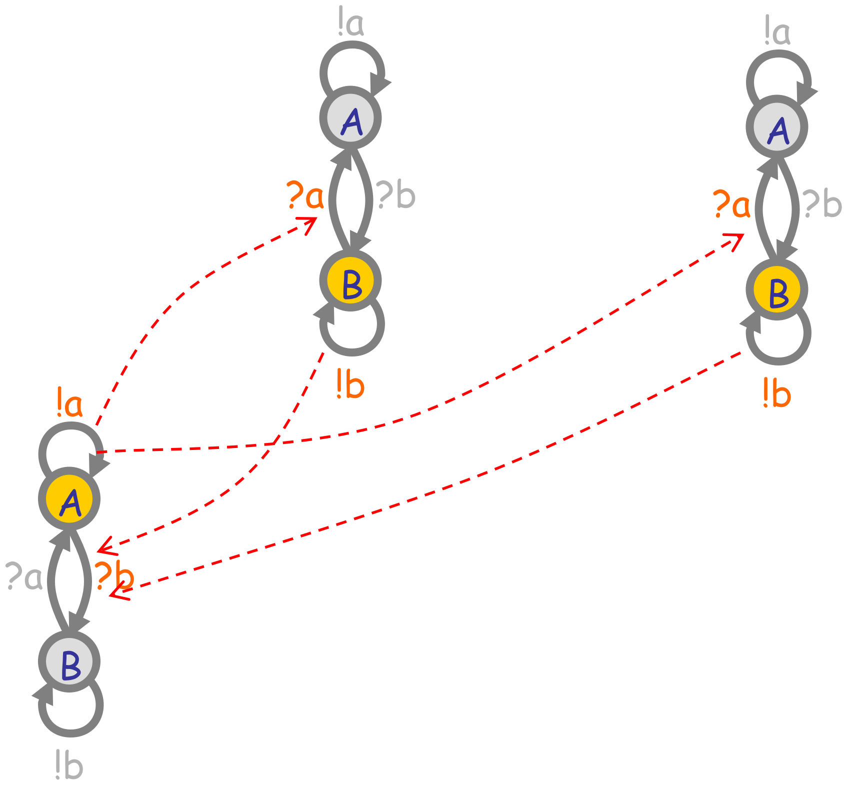


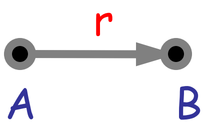


All-B stable
population

Nondeterministic
population behavior
("multistability")

# CTMC Semantics



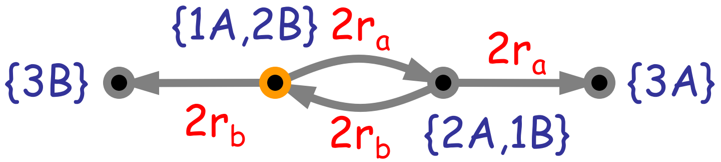


CTMC
(homogeneous) Continuous Time Markov Chain
- directed graph with no self loops
- nodes are system states
- arcs have transition rates

Probability of holding in state A:

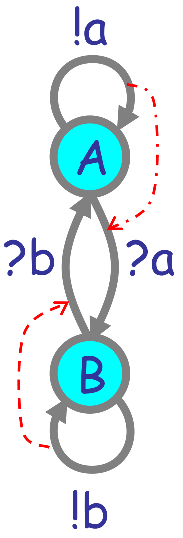
$Pr(H_A>t) = e^{-rt}$

in general, $Pr(H_A>t) = e^{-Rt}$ where R is the sum of all the exit rates from A



CTMC

# Groupies and Celebrities



## Celebrity

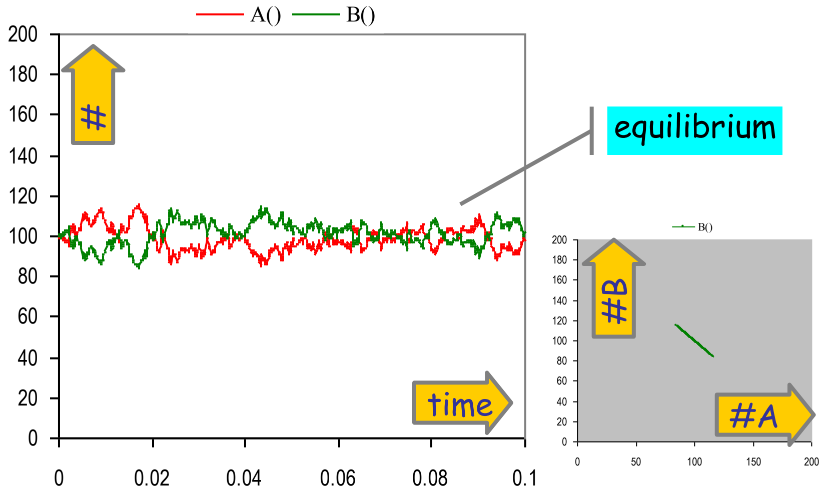
(does not want to be like somebody else)

```
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())
```

a@1.0

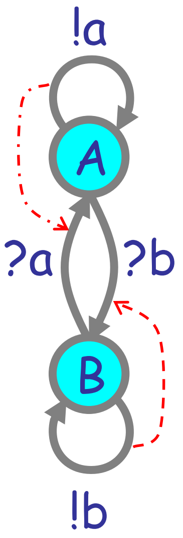
b@1.0

A stochastic collective of celebrities:



equilibrium

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.



## Groupie

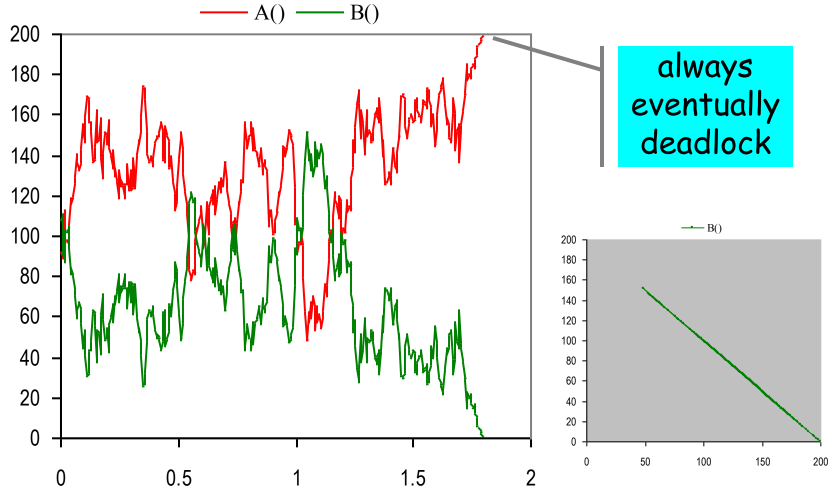
(wants to be like somebody different)

```
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@1.0

b@1.0

A stochastic collective of groupies:



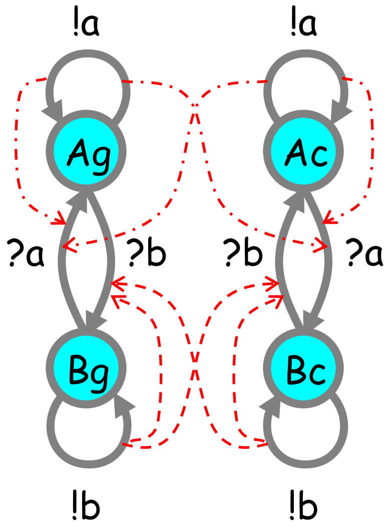
always eventually deadlock

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.

# Both Together

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

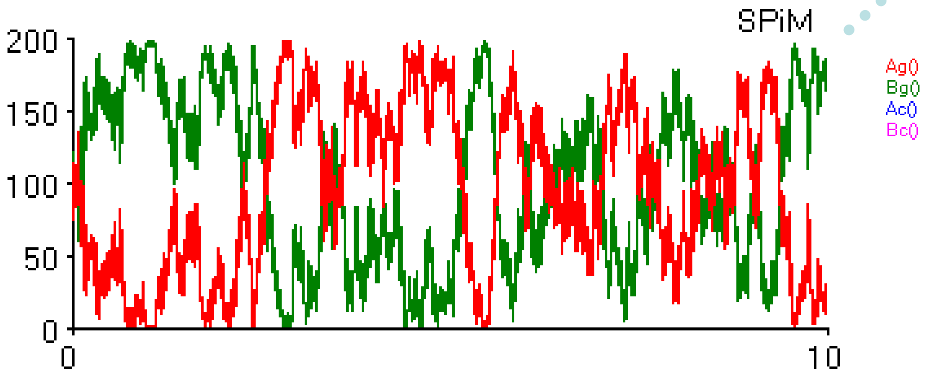
Many Groupies



A few Celebrities

```
directive sample 10.0
directive plot Ag(); Bg(); Ac(); Bc()

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

let Ac() = do !a; Ac() or ?a; Bc()
and Bc() = do !b; Bc() or ?b; Ac()

let Ag() = do !a; Ag() or ?b; Bg()
and Bg() = do !b; Bg() or ?a; Ag()

run 1 of Ac()
run 100 of (Ag() | Bg())
```

never deadlock

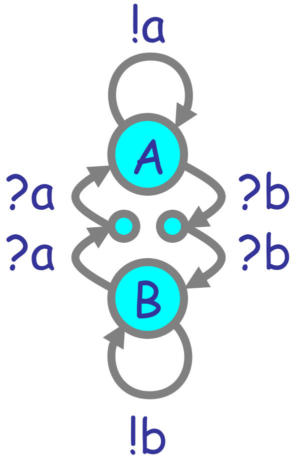


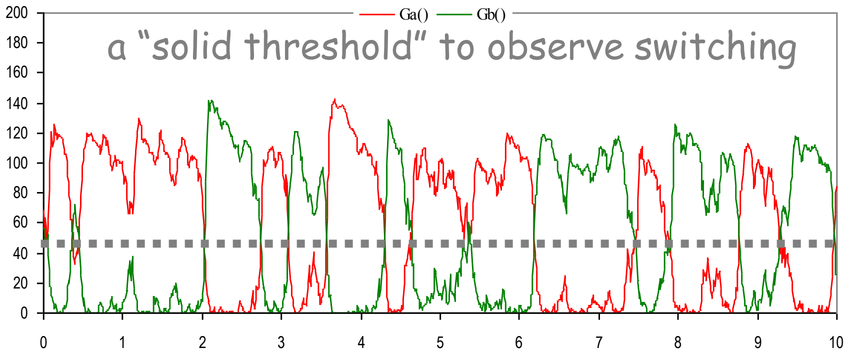
A tiny bit of "noise" can make a huge difference

Regularity can arise not far from chaos

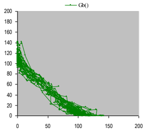
# Hysteric Groupies

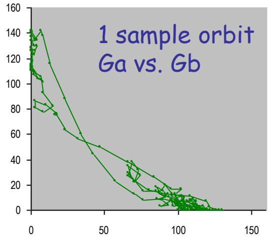
We can get more regular behavior from groupies if they “need more convincing”, or “hysteresis” (history-dependence), to switch states.





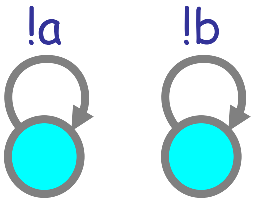
a “solid threshold” to observe switching





1 sample orbit
Ga vs. Gb

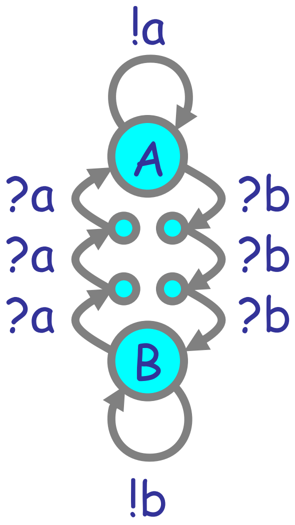
```
directive sample 10.0 1000
directive plot Ga(); Gb()

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())
```

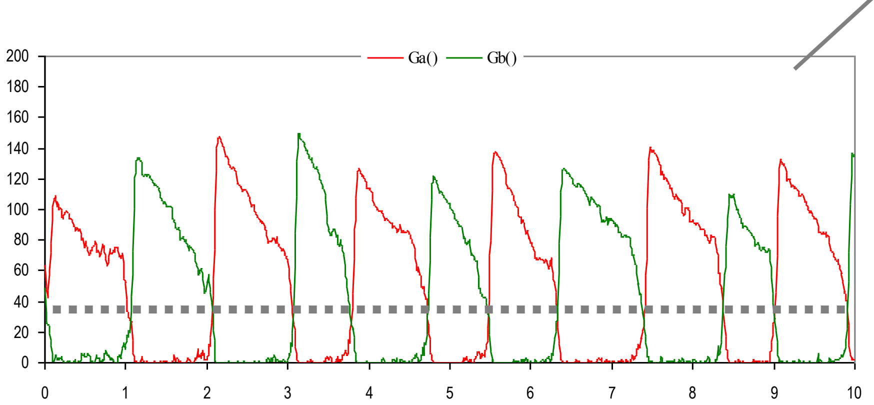


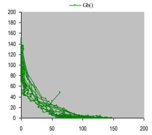
(With doping to break deadlocks)

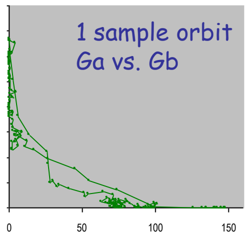
N.B.: It will not oscillate without doping (noise)

“regular” oscillation









1 sample orbit
Ga vs. Gb

```
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())
```

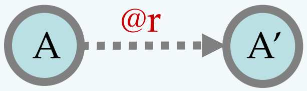
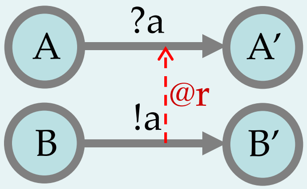
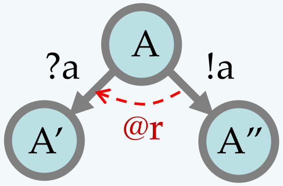
# Semantics of Collective Behavior

The collective behavior of even very simple automata is difficult to predict.

# Automata to Chemistry

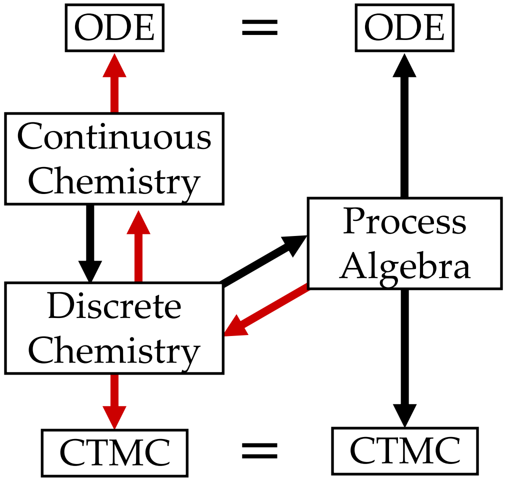
V = interaction volume

$N_A$ = Avogadro's number

| Automata → | Discrete Chemistry<br>(molecule counts) → | Continuous Chemistry<br>(concentrations) $\gamma = N_A V$ |
|---|---|---|
| initial states<br>A \| A \| … \| A | initial quantities<br>$\#A_0$ | initial concentrations<br>$[A]_0$ with $[A]_0 = \#A_0/\gamma$ |
|  | $A \overset{r}{\dashrightarrow} A'$ | $A \rightarrow^{k} A'$ with $k = r$ |
|  | $A+B \overset{r}{\dashrightarrow} A'+B'$ | $A+B \rightarrow^{k} A'+B'$ with $k = r\gamma$ |
|  | $A+A \overset{2r}{\dashrightarrow} A'+A''$ | $A+A \rightarrow^{2k} A'+A''$ with $k = r\gamma/2$ |
|  | ↓<br>CTMC | ↓<br>ODE ($[A]^{\bullet} \equiv d[A]/dt$<br>change of concentration over time) |

Think $\gamma = 1$

i.e. $V = 1/N_A$



ODE = ODE

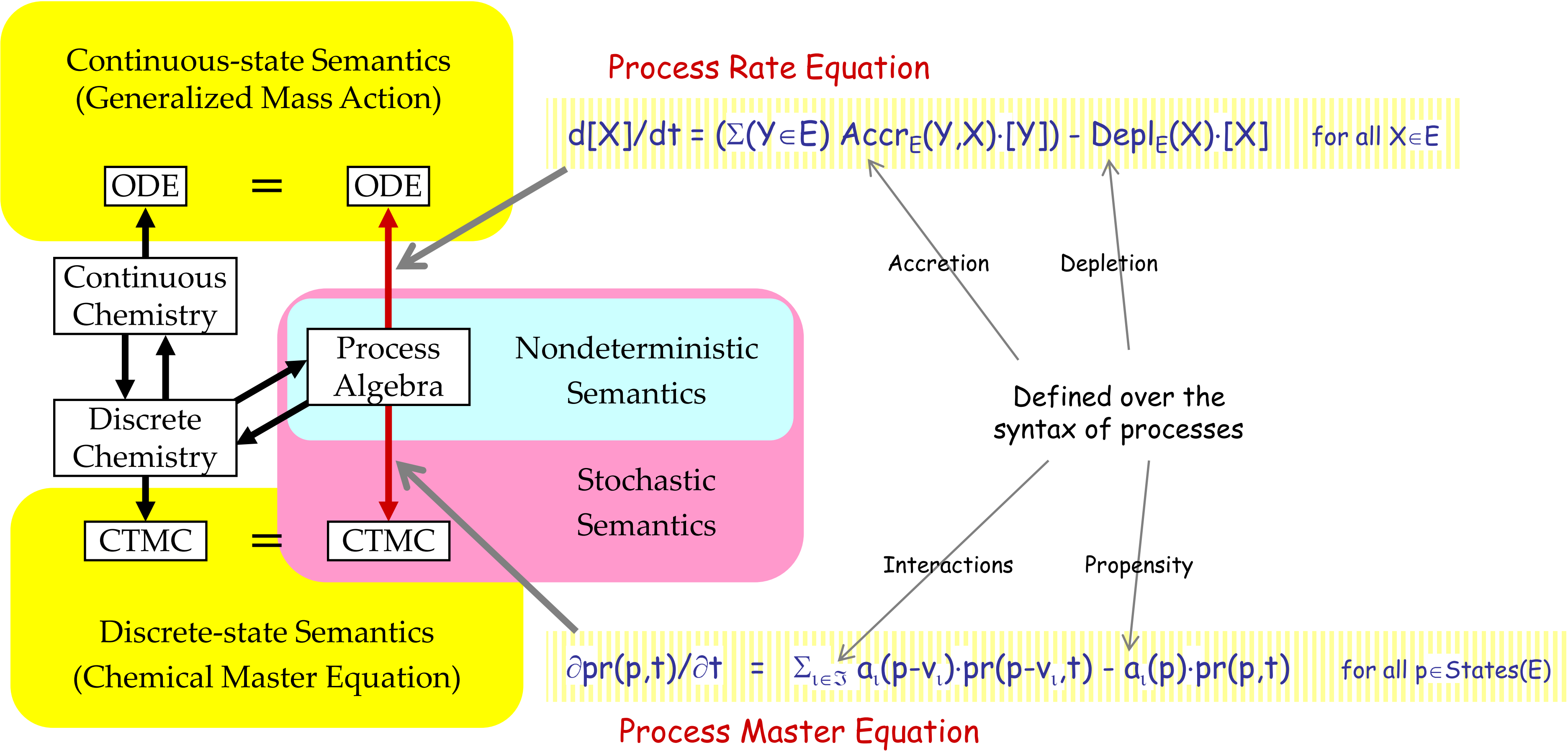
Continuous Chemistry

Process Algebra

Discrete Chemistry

CTMC = CTMC

# Quantitative Process Semantics



Continuous-state Semantics (Generalized Mass Action)

ODE = ODE

Continuous Chemistry

Discrete Chemistry

Process Algebra

Nondeterministic Semantics

Stochastic Semantics

CTMC = CTMC

Discrete-state Semantics (Chemical Master Equation)

Process Rate Equation

$d[X]/dt = (\Sigma(Y\in E)\ Accr_E(Y,X)\cdot[Y]) - Depl_E(X)\cdot[X] \quad \text{for all } X\in E$

Accretion

Depletion

Defined over the syntax of processes

Interactions

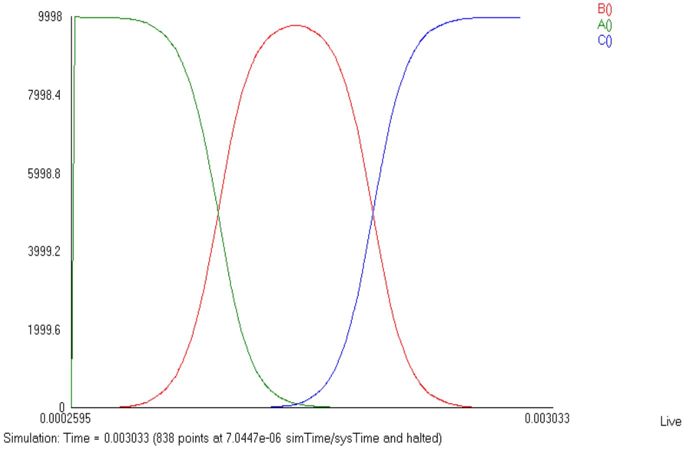
Propensity

$\partial pr(p,t)/\partial t \ = \ \Sigma_{\iota\in\mathfrak{I}}\ a_\iota(p-v_\iota)\cdot pr(p-v_\iota,t) - a_\iota(p)\cdot pr(p,t) \quad \text{for all } p\in States(E)$

Process Master Equation

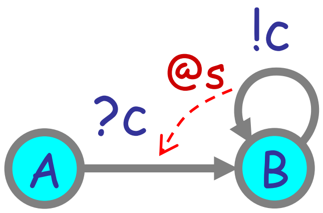
# Waves

A programming exercise:
build me a population like this:



Simulation: Time = 0.003033 (838 points at 7.0447e-06 simTime/sysTime and halted)

# Nonlinear Transition (NLT)



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

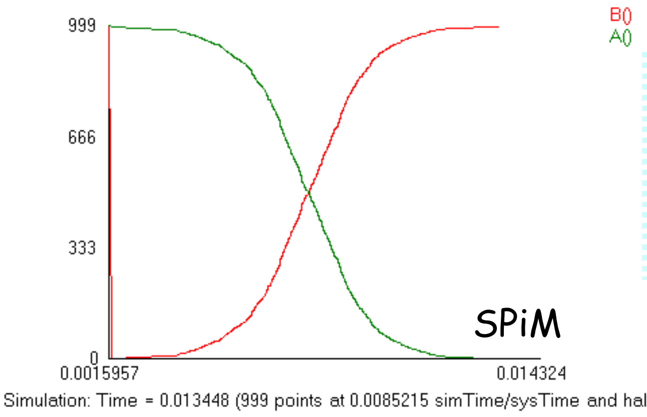
↓

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

↓

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

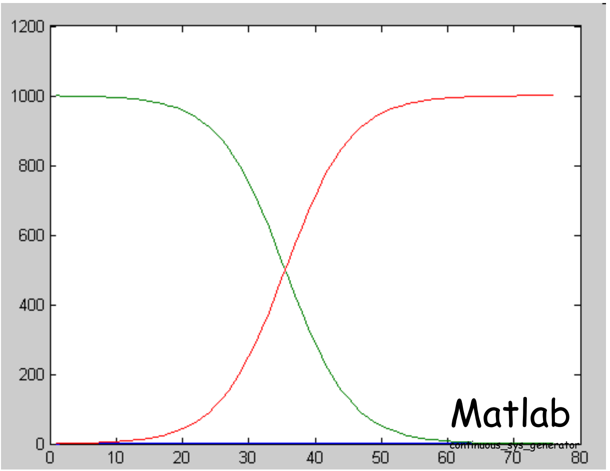
($[A]^{\bullet} \equiv d[A]/dt$
change of concentration
over time)



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

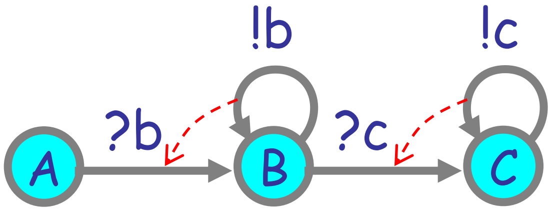
```
directive sample 0.02 1000
directive plot B(); A()

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
```

# Two NLTs: Bell Shape



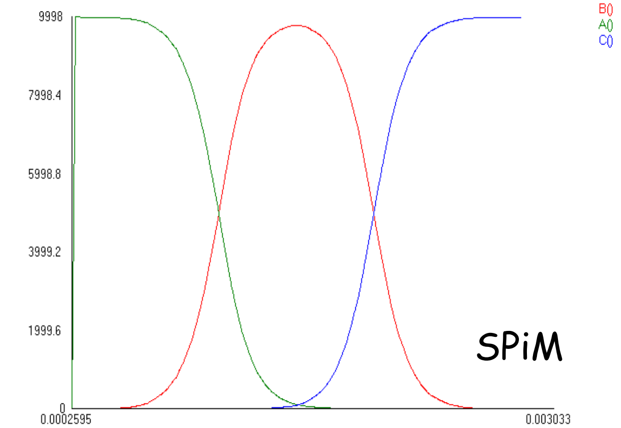
$$[B]^\bullet = [B]([A]-[C])$$

```
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())
```

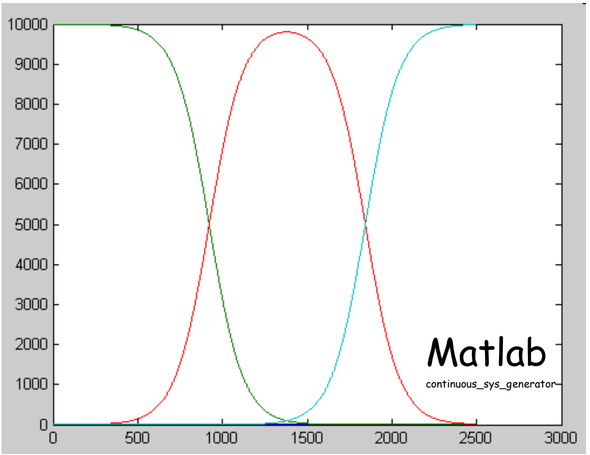
$$A = ?b_{(1)};B$$
$$B = !b_{(1)};B \oplus ?c_{(1)};C$$
$$C = !c_{(1)};C$$

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

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

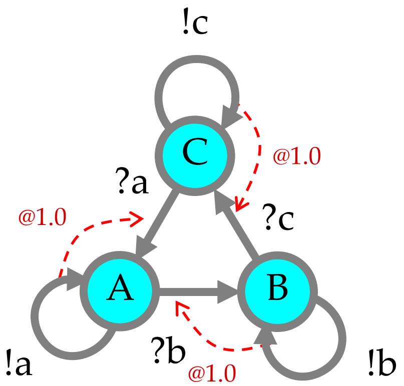


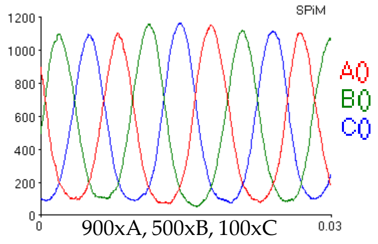
Simulation: Time = 0.003033 (838 points at 7.0447e-06 simTime/sysTime and halted)



```
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
```

# NLT in a Cycle: Oscillator





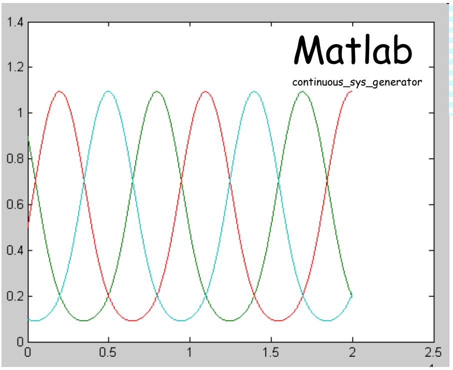
900xA, 500xB, 100xC

```
directive sample 0.03 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 ?b; B()
and B() = do !b;B() or ?c; C()
and C() = do !c;C() or ?a; A()

run (900 of A() | 500 of B() | 100 of C())
```

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

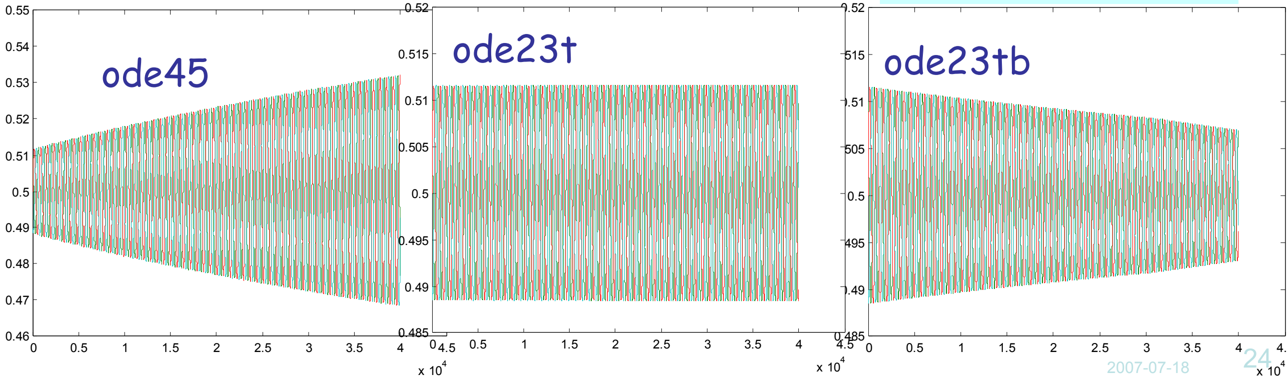
$$A+B \to^{s} B+B$$
$$B+C \to^{s} C+C$$
$$C+A \to^{s} A+A$$

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



```
interval/step [0:0.001:20.0]
(A)     dx1/dt = - x1*x2 + x3*x1      0.9
(B)     dx2/dt = - x2*x3 + x1*x2      0.5
(C)     dx3/dt = - x3*x1 + x2*x3      0.1
```

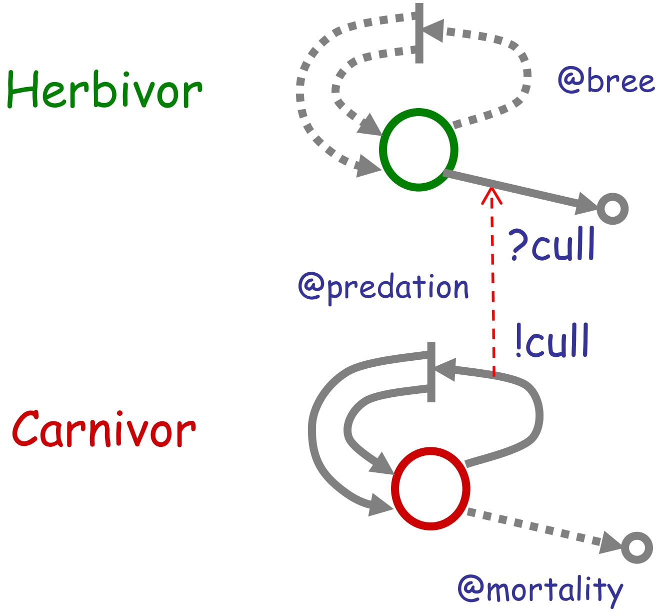
```
interval/step [0:0.01:400.0]
(A)     dx1/dt = - x1*x2 + x3*x1      0.51
(B)     dx2/dt = - x2*x3 + x1*x2      0.5
(C)     dx3/dt = - x3*x1 + x2*x3      0.49
```



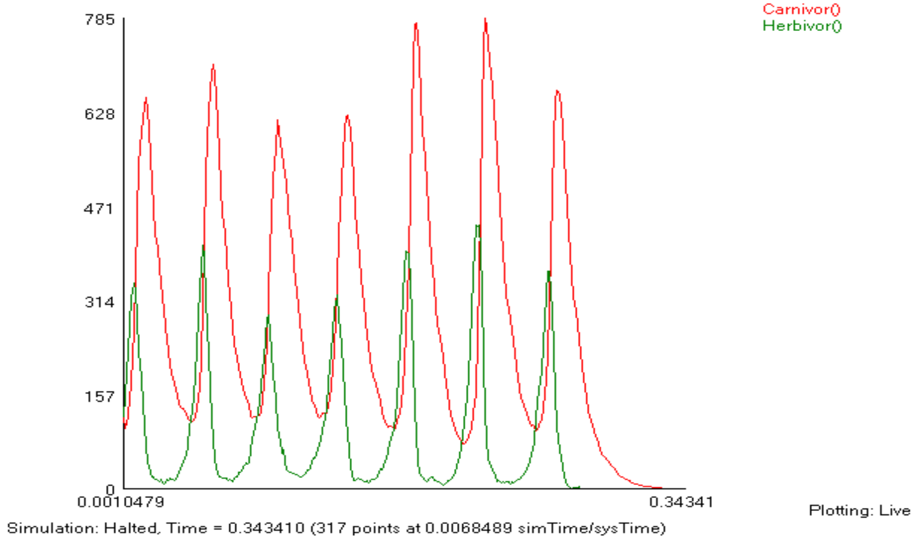
# Lotka-Volterra

Beyond Automata

# 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()
```



Simulation: Halted, Time = 0.343410 (317 points at 0.0068489 simTime/sysTime)

Plotting: Live

*An unbounded state* system!

# Lotka-Volterra in Matlab

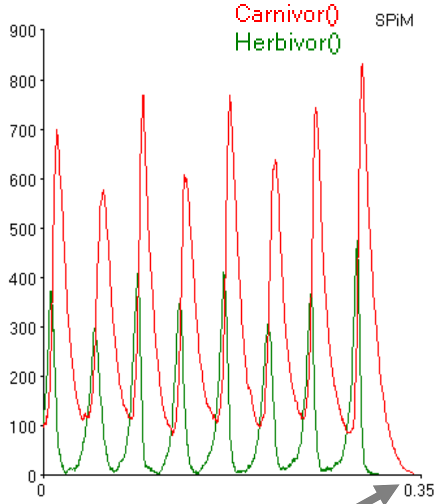
$H = \tau_b; (H|H) \oplus ?c_{(p)};0$
$C = \tau_m;0 \oplus !c_{(p)};(C|C)$
$\#H_0, \#C_0$

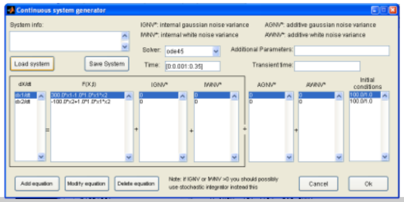
$H \rightarrow^{b} H + H$
$C \rightarrow^{m} 0$
$H + C \rightarrow^{p\gamma} C + C$
$[H]_0 = \#H_0/\gamma$
$[C]_0 = \#C_0/\gamma$

$[H]^\bullet = b[H]-p\gamma[H][C]$
$[C]^\bullet = -m[C]+p\gamma[H][C]$
$[H]_0 = \#H_0/\gamma$
$[C]_0 = \#C_0/\gamma$

$m=100.0$
$b=300.0$
$p=1.0$
$\gamma=1.0$
$\#H_0 = 100$
$\#C_0 = 100$

```
directive sample 0.35 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()
```





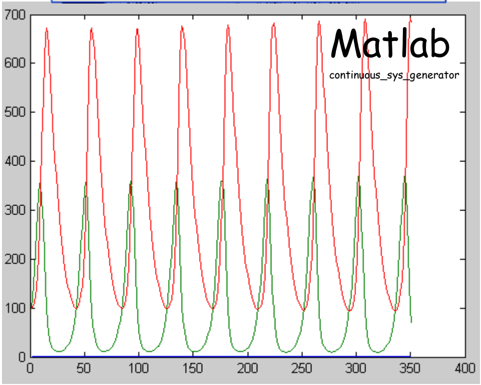
System info:

IGNV*: internal gaussian noise variance
IWNV*: internal white noise variance
AGNV*: additive gaussian noise variance
AWNV*: additive white noise variance

Solver: ode45
Time: [0:0.001:0.35]
Additional Parameters:
Transient time:

| dX/dt | | F(X,t) | | IGNV* | | IWNV* | | AGNV* | | AWNV* | Initial conditions |
|---|---|---|---|---|---|---|---|---|---|---|---|
| dx1/dt | = | 300.0*x1-1.0*1.0*x1*x2 | + | 0 | + | 0 | + | 0 | + | 0 | 100.0/1.0 |
| dx2/dt | | -100.0*x2+1.0*1.0*x1*x2 | | 0 | | 0 | | 0 | | 0 | 100.0/1.0 |

Note: if IGNV or IWNV >0 you should possibly use stochastic integrator instead this



Extinction

No extinction

Which one is the “right prediction”?

# Conclusions

# Conclusions

- Compositional models
  - Accurate (at the “appropriate” abstraction level).
  - Manageable (so we can scale them up by composition).
  - Executable (stochastic simulation).

- Analysis techniques
  - Mathematical techniques: Markov theory, Chemical Master Equation, and Rate Equation
  - Computing techniques: Abstraction and Refinement, Model Checking, Causality Analysis.

- Many “obvious” lines of extensions
  - Parametric processes for model factorization
  - *Poly*automata for ***Bio***-Chemistry: complexation and polymerization
  - Ultimately, rich process-algebra based modeling languages.

- An Artificial Biochemistry
  - A scalable mathematical and computational modeling framework.
  - To understand “real biochemistry” on a large scale.