Artificial Biochemistry

Luca Cardelli

Microsoft Research

Santa Fe Institute 2006-11-03

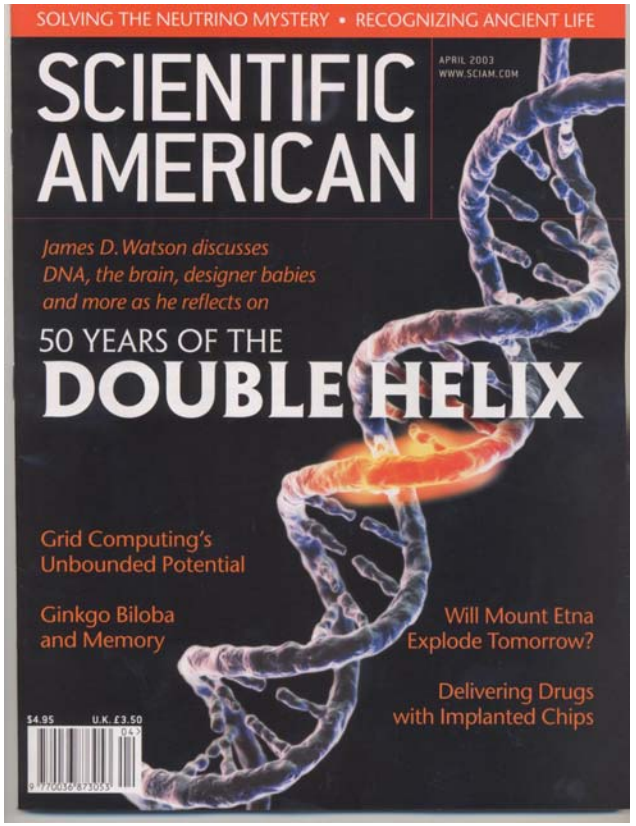
http://LucaCardelli.name

50 Years of Molecular Cell Biology

- Genes are made of DNA
  - Store digital information as sequences of 4 different nucleotides
  - Direct protein assembly through RNA and the Genetic Code

- Proteins (>10000) are made of amino acids
  - Process signals
  - Activate genes
  - Move materials
  - Catalyze reactions to produce substances
  - Control energy production and consumption

- Bootstrapping still a mystery
  - DNA, RNA, proteins, membranes are today interdependent. Not clear who came first
  - Separation of tasks happened a long time ago
  - Not understood, not essential



Towards Systems Biology

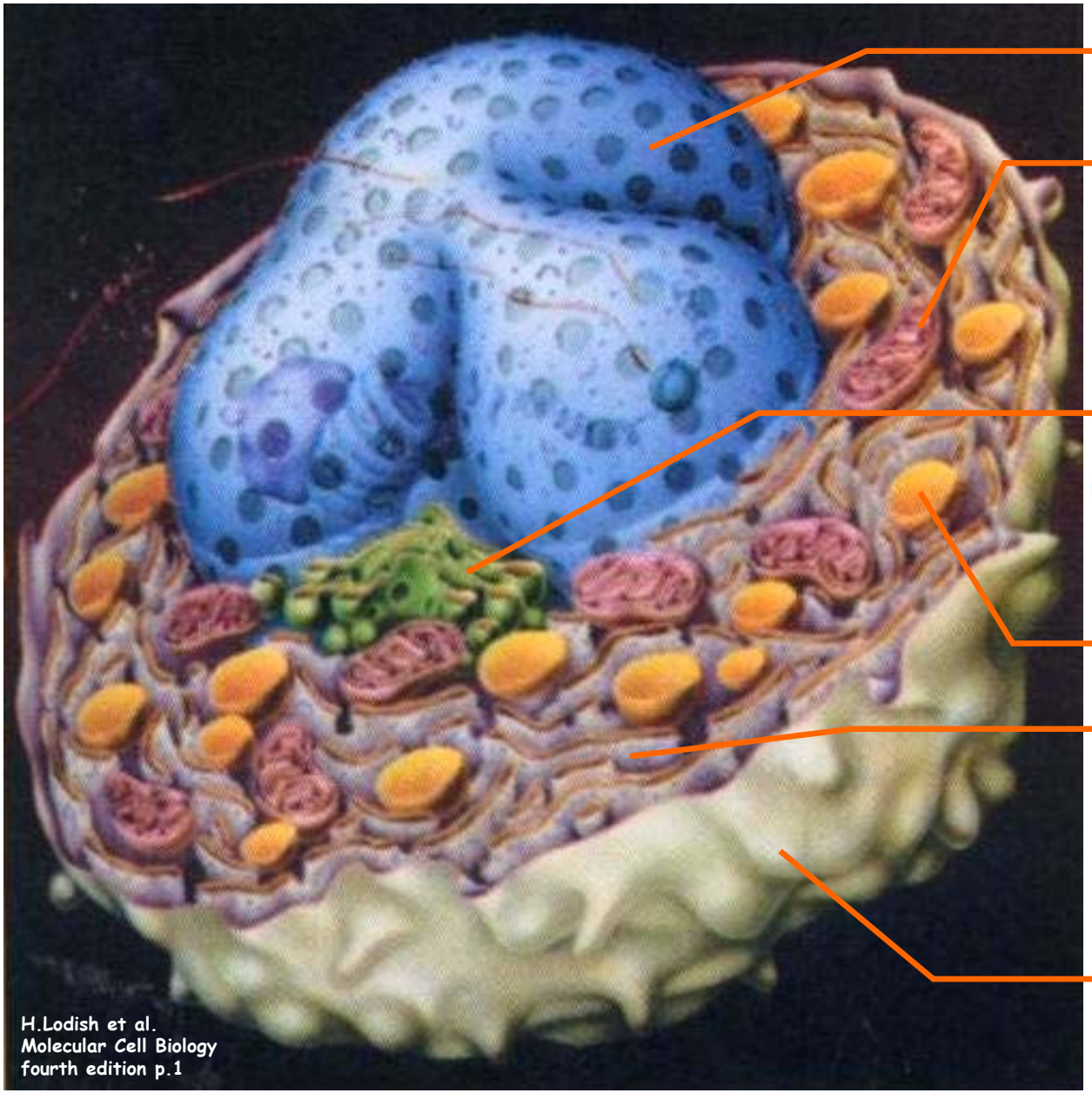
- Biologists now understand many of the cellular components
  - A whole team of biologists will typically study a single protein for years
  - Reductionism: understand the components in order to understand the system
- But this has not led to understand how "the system" works
  - Behavior comes from complex patterns of interactions between components
  - Predictive biology and pharmacology still rare
  - Synthetic biology still unreliable
- New approach: try to understand "the system"
  - Experimentally: massive data gathering and data mining (e.g. Genome projects)
  - Conceptually: modeling and analyzing networks (i.e. interactions) of components
- What kind of a system?
  - Just beyond the basic chemistry of energy and materials processing…
  - Built right out of digital information (DNA)
  - Based on information processing for both survival and evolution
  - *Highly* concurrent
- Can we fix it when it breaks?
  - Really becomes: How is information structured and processed?

Reverse-Engineer This!

Eukaryotic Cell

(10~100 trillion in human body)

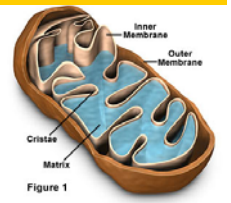
Membranes everywhere



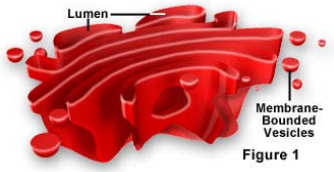
H.Lodish et al.
Molecular Cell Biology
fourth edition p.1

Nuclear membrane

Mitochondria



Golgi



Vesicles

E.R.

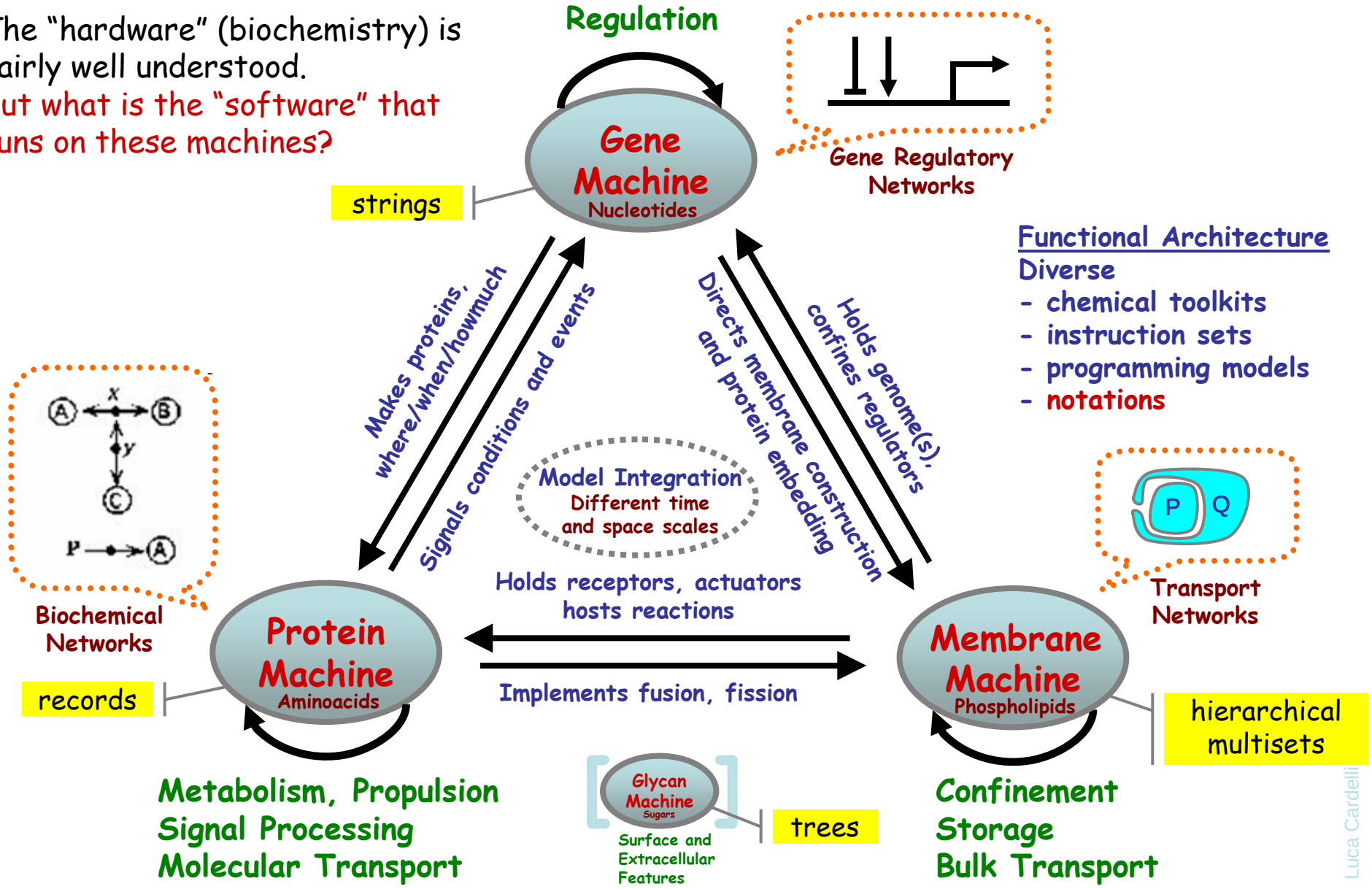
Plasma membrane (<10% of all membranes)

…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?

Abstract Machines of Systems Biology

The "hardware" (biochemistry) is fairly well understood.
But what is the "software" that runs on these machines?



Storing *Processes*

- Today we represent, store, search, and analyze:
  - Gene sequence data
  - Protein structure data
  - Metabolic network data
  - Signaling pathway data
  - …

Cellular Abstractions: Cells as Computation
Regev&Shapiro **NATURE** vol 419, 2002-09-26, 343

- How can we represent, store, and analyze *biological processes*?
  - Scalable, precise, dynamic, highly structured, maintainable representations for *systems biology*.
  - Not just huge lists of chemical reactions or differential equations.

- In computing…
  - There are well-established scalable representations of dynamic reactive processes.
  - They look more or less like little, mathematically based, programming languages.

Methods

- Model Construction (writing things down precisely)
  - Formalizing the notations used in systems biology.
  - Formulating description languages.
  - Studying their kinetics (semantics).

- Model Validation (using models for postdiction and prediction)
  - Simulation from compositional descriptions
    - Stochastic: quantitative concurrent semantics.
    - Hybrid: discrete transitions between continuously evolving states.
  - "Program" Analysis
    - Control flow analysis
    - Causality analysis
  - Modelchecking
    - Standard, Quantitative, Probabilistic

Modeling as Mapping

- Building a model is like building a map of the territory
  - It has to be accurate (it shows you where to go)
  - and manageable (you can take it with you).
- What is the best map?
  - 1:10,000,000 very manageable, but can't even see Santa Fe.
  - 1:1 very accurate, but can't fold it in your pocket.
  - 1:10,000 good for Santa Fe, and unfolds on your lap.
- Animating maps
  - More than static geography (should include traffic patterns, weather, etc.)
  - Perturbation (effects of roadwork) and Prediction (what's the traffic like tomorrow morning).
- Correlating maps
  - Correlating features between maps of different scales, as well as between maps and the territory. (Needs a theory of correlating models.)
- Composing maps (compositionality)
  - A single map of the whole earth, at almost any scale, is unmanageable.
  - Should have separate maps of different areas (including e.g. their local traffic patterns), then "compose" them on need when crossing areas.

Stochastic Collectives

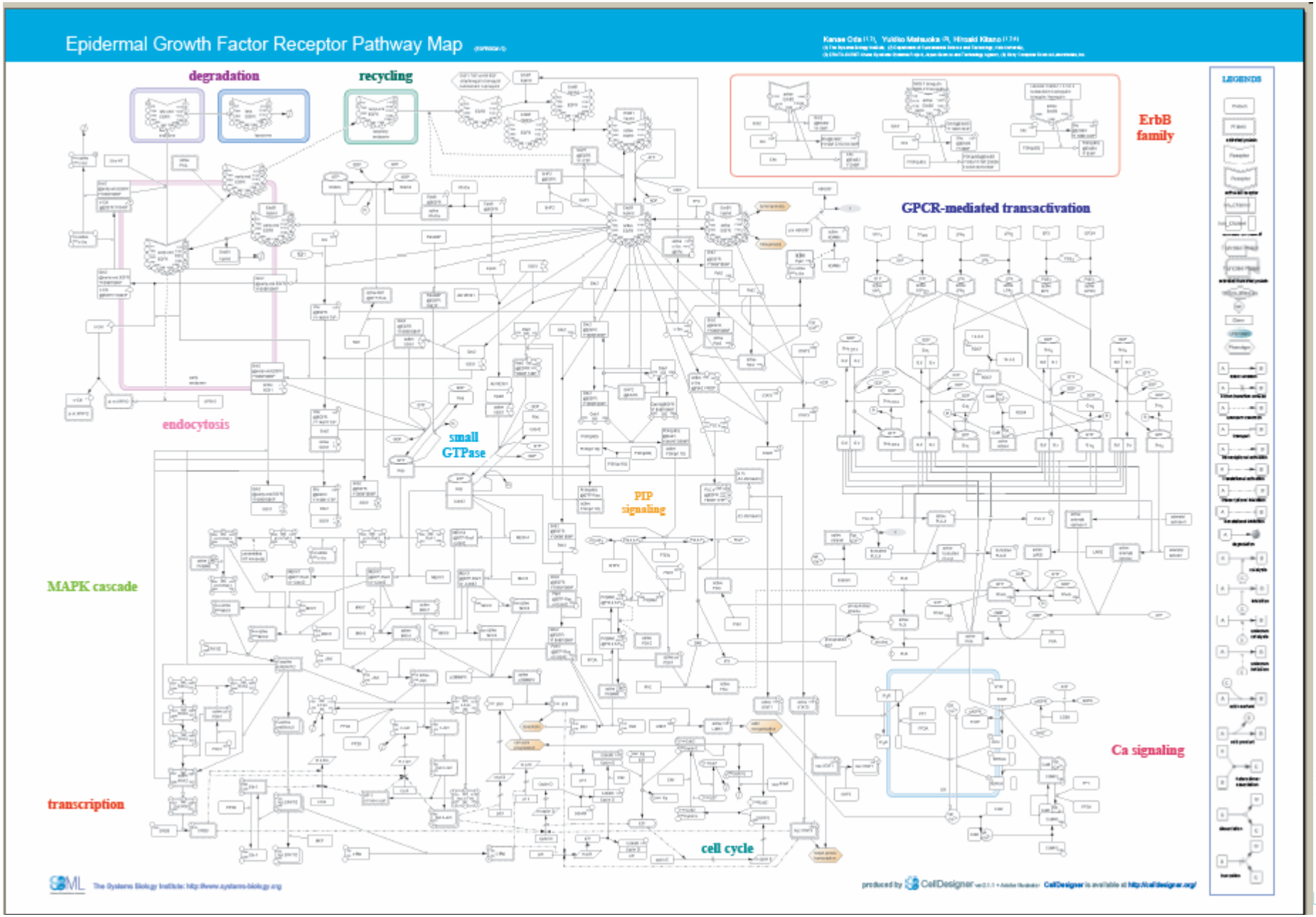
Stochastic Collectives

- "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, and heterogeneous)
    - Not quite process algebra ("finite state" and "collective" emphasis)

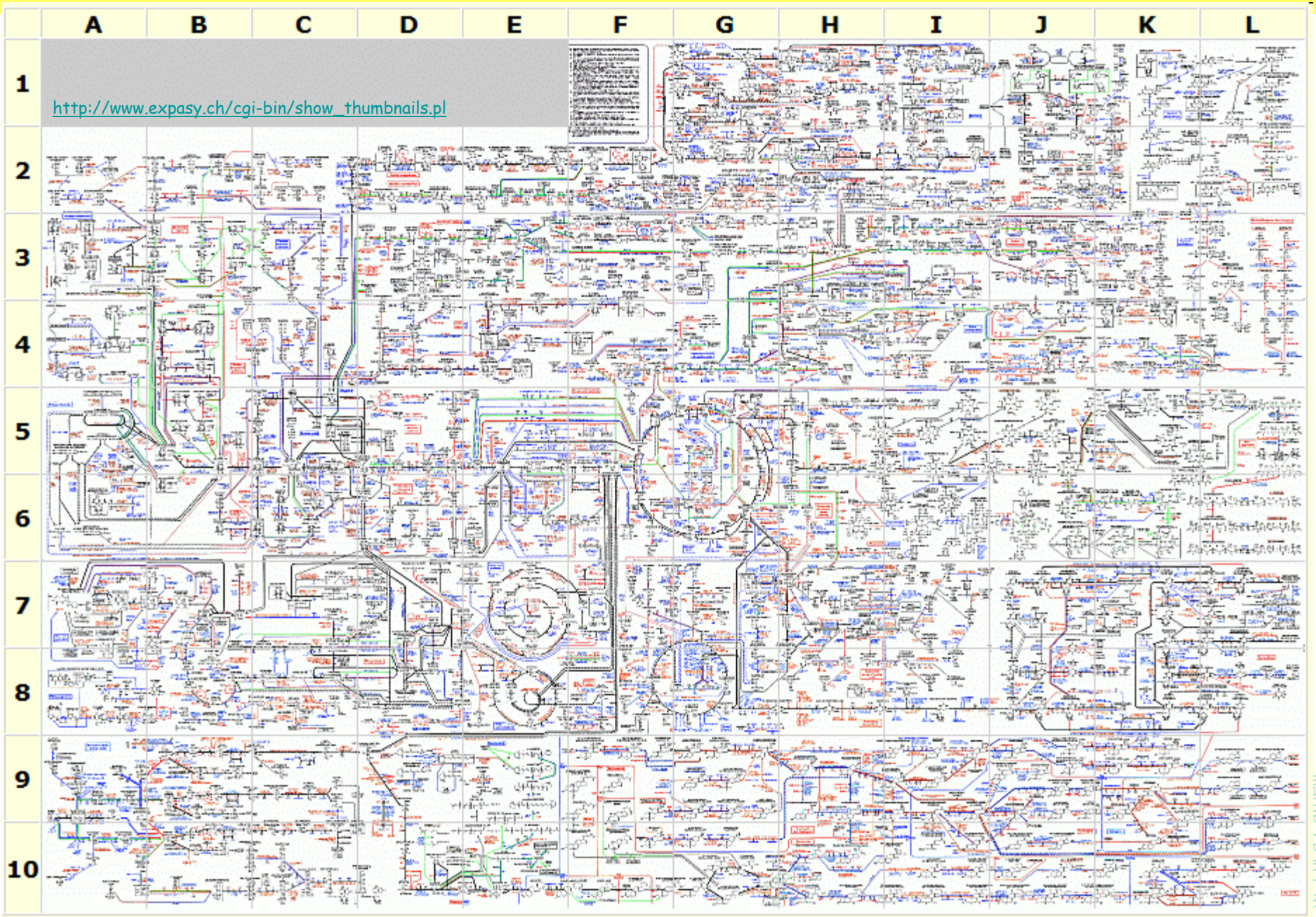
- "Stochastic":
  - Interactions have *rates*
    - Not quite discrete (hundreds or thousands of components)
    - Not quite continuous (non-trivial stochastic effects)
    - Not quite hybrid (no "switching" between regimes)

- 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].

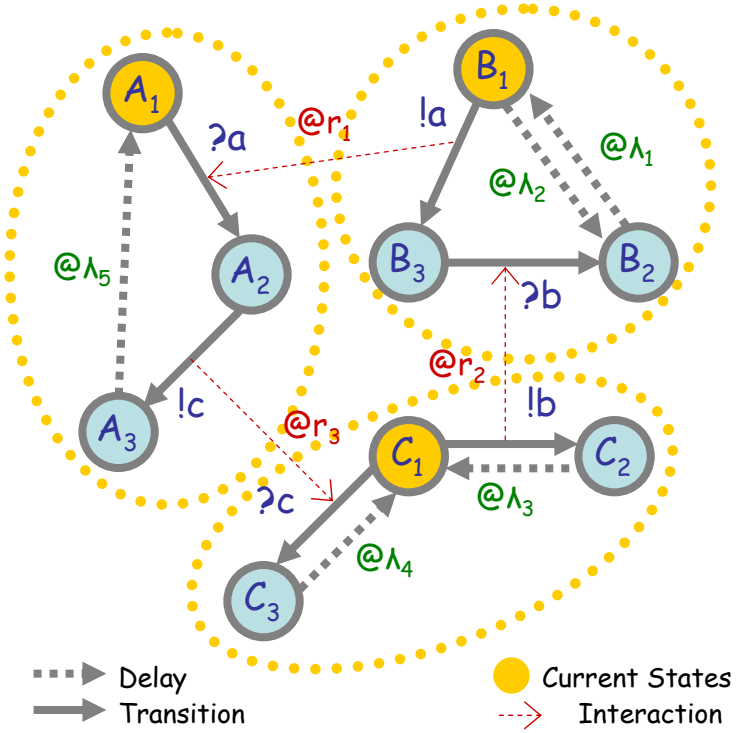
State Transitions



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.

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

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

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

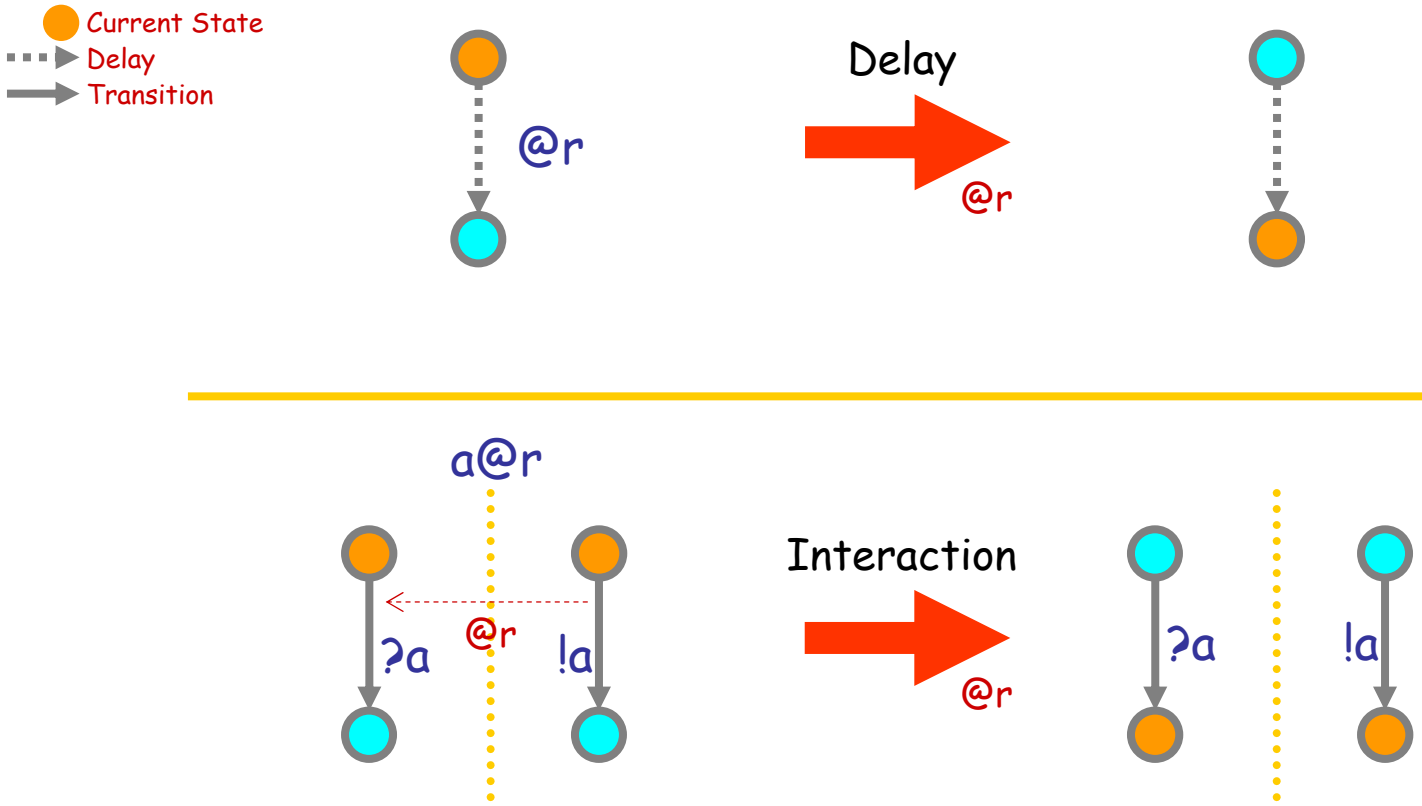
$$\begin{array}{l}C_1 = !b;\ C_2 + ?c;\ C_3\\ C_2 = \tau@\lambda_3;\ C_1\\ C_3 = \tau@\lambda_4;\ C_2\end{array}\Big\}\ \text{Automata}$$

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

"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]

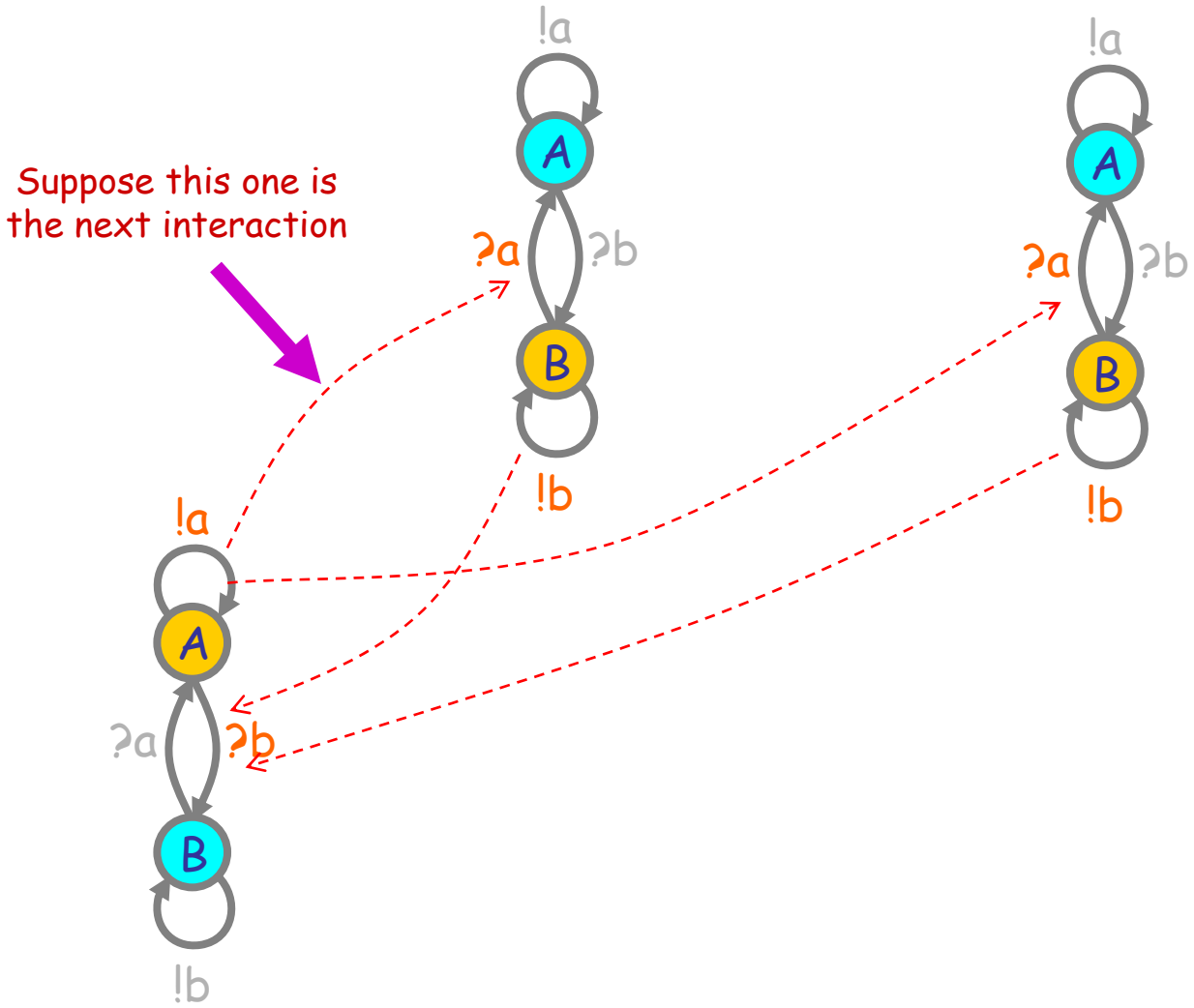
Interacting Automata Transition Rules



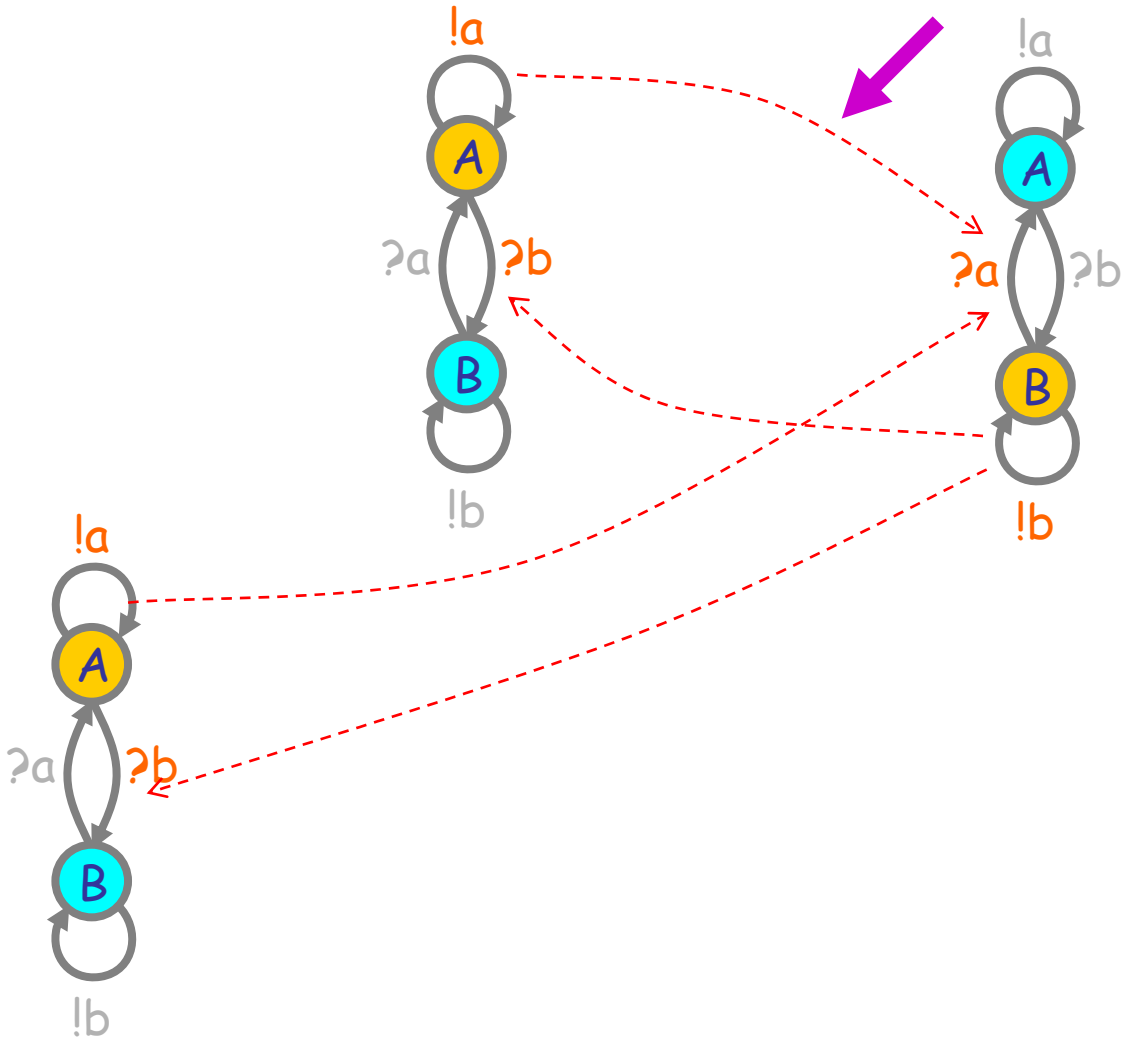
Q: What kind of mass behavior can this produce?

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

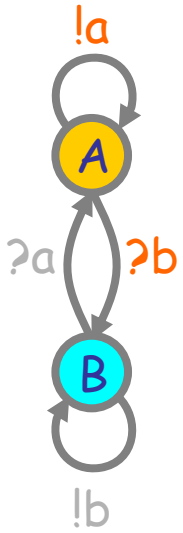
Interactions in a Population

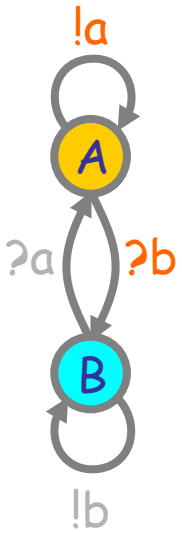


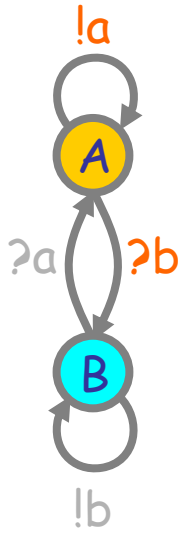
Interactions in a Population



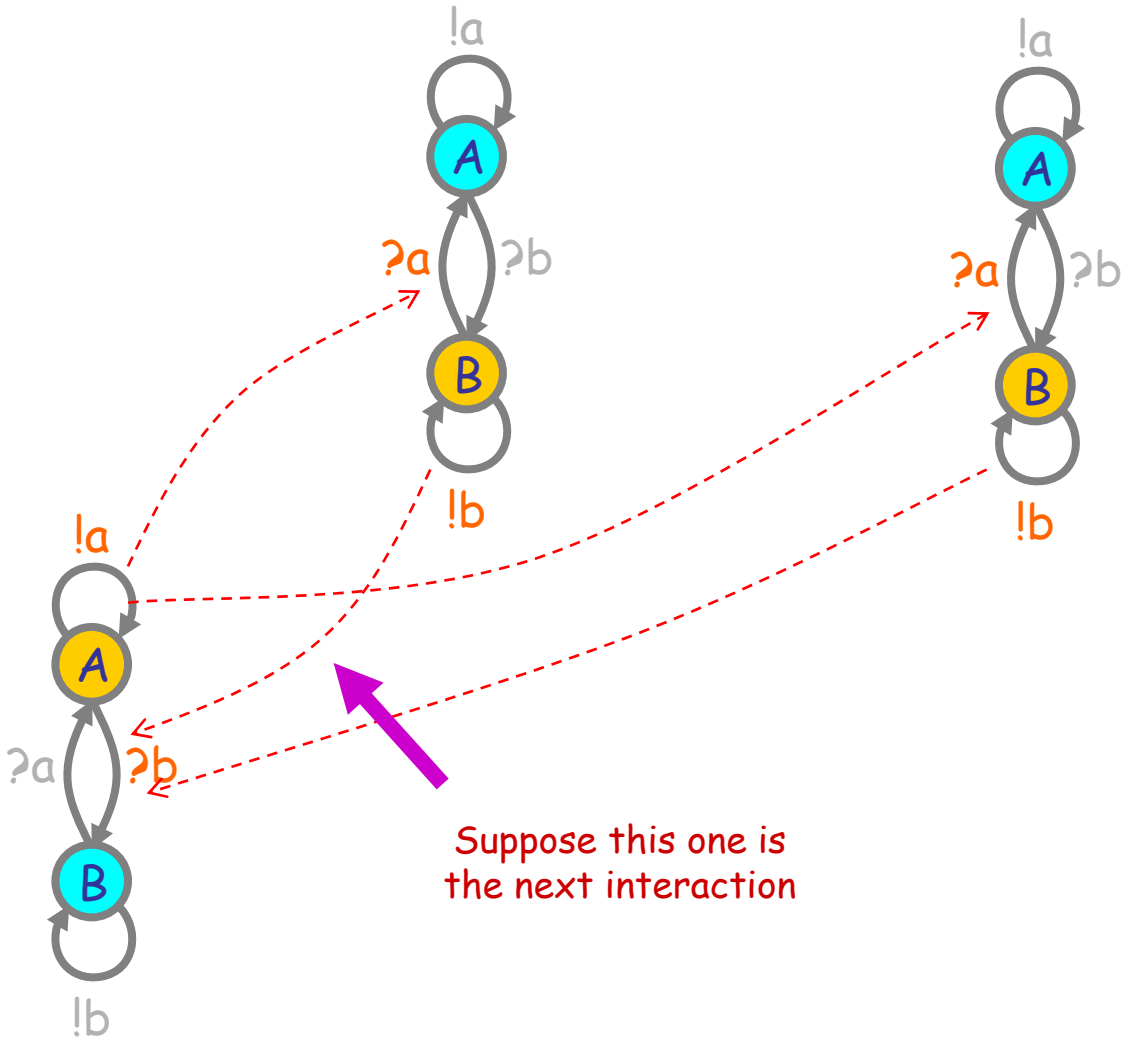
Interactions in a Population



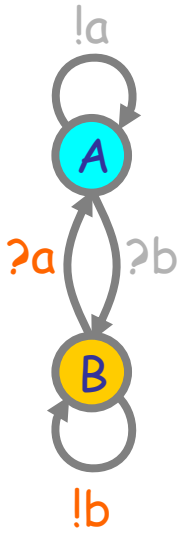


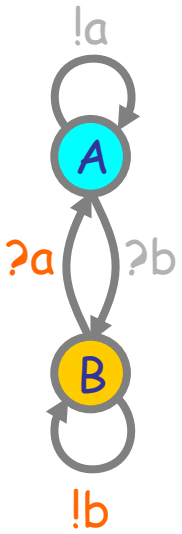


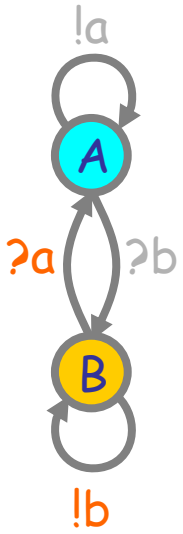
Interactions in a Population (2)



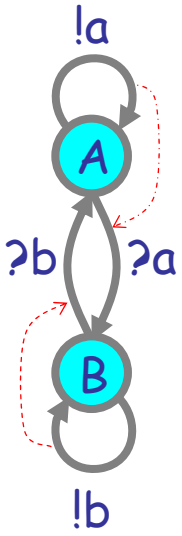
Interactions in a Population (2)







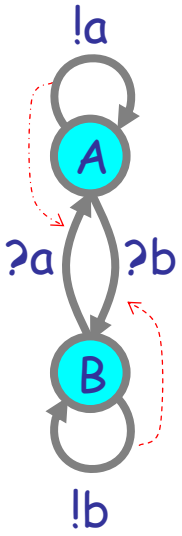
Groupies and Celebrities



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())
```

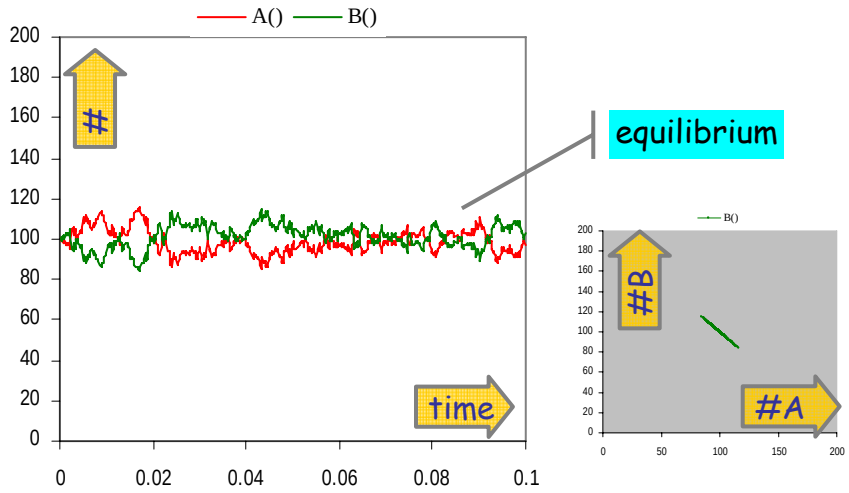


Groupie

(wants to be like somebody different)

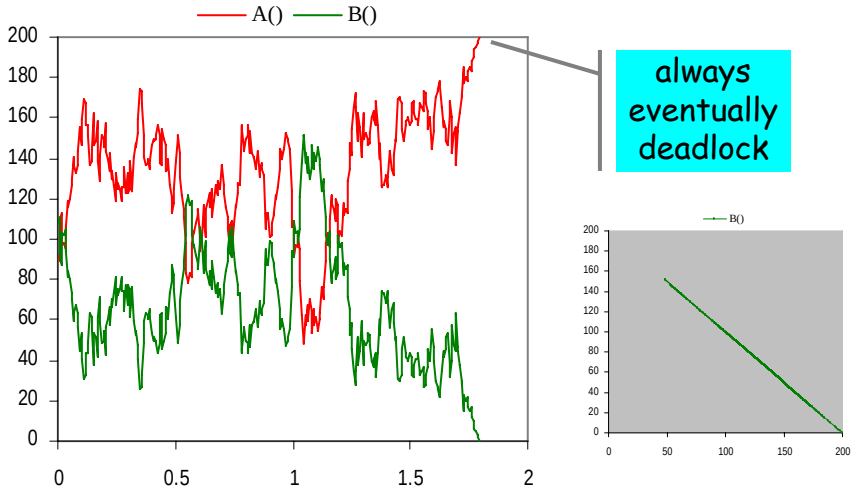
```
directive sample 0.1 200
directive plot A(); B()

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:



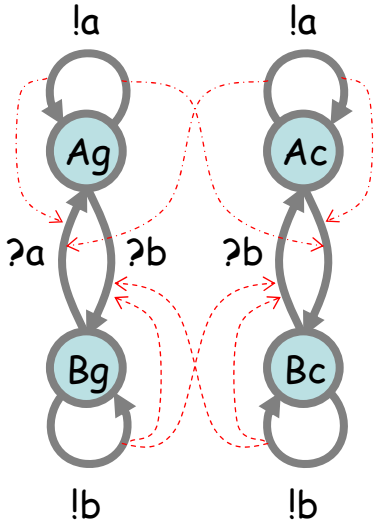
Unstable because within an A majority, an A has difficulty finding a B to emulate, but the few B's have plenty of A's to emulate, so the majority may switch to B. Leads to deadlock when everybody is in the same state and there is nobody different to emulate.

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

Both Together

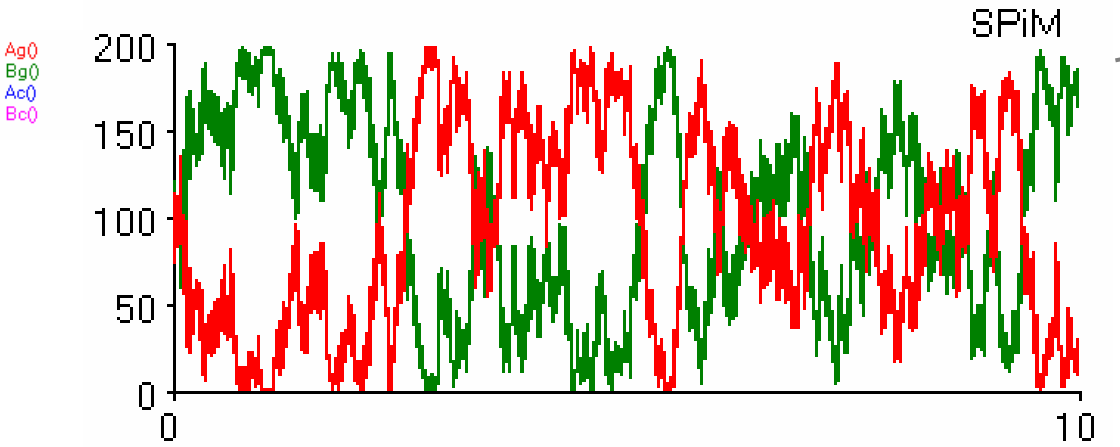
A way to break the deadlocks: Groupies with just a few Celebrities

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())
```

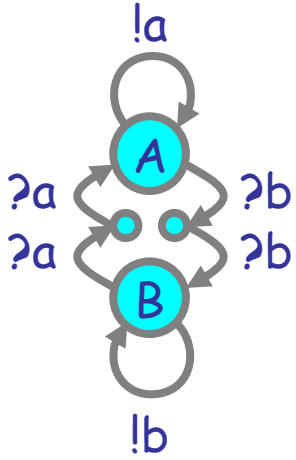


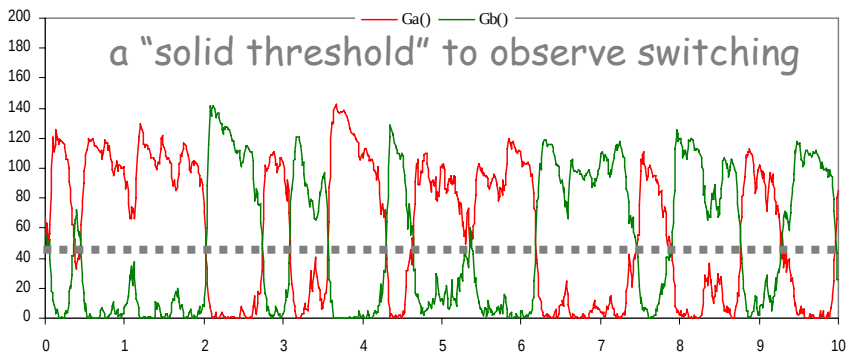
never deadlock

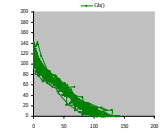
Regularity can arise not far from chaos

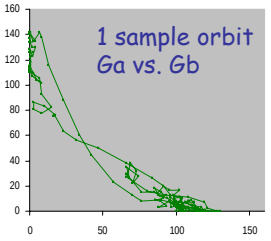
Hysteric Groupies

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

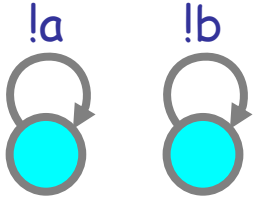








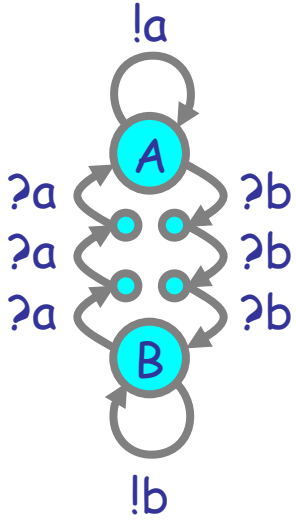

```
directive sample 10.0 1000
directive plot Ga(); Gb()

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())
```

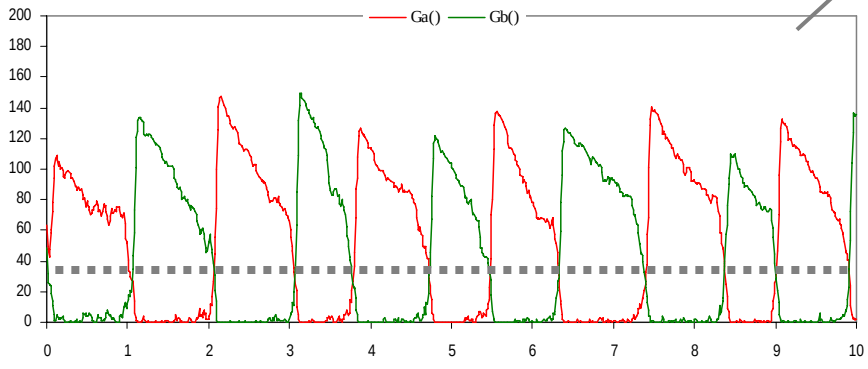


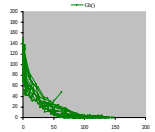
(With doping to break deadlocks)

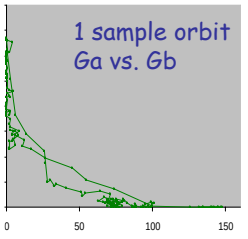
N.B.: It will not oscillate without doping (noise)

"regular" oscillation



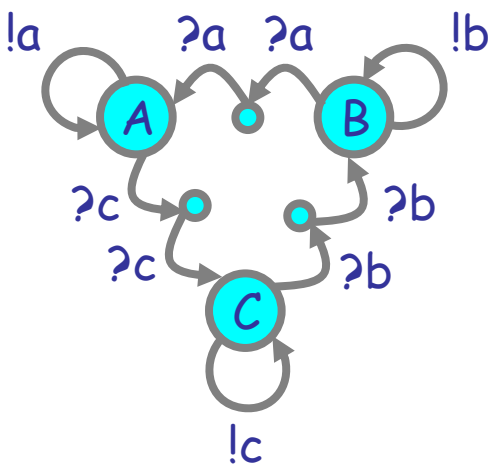


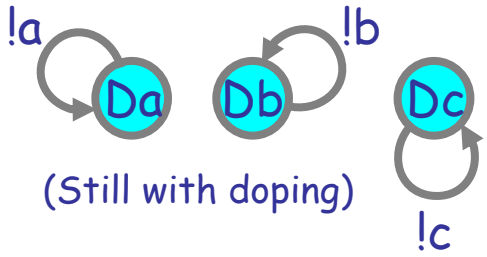





```
directive sample 10.0 1000
directive plot Ga(); Gb()

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())
```

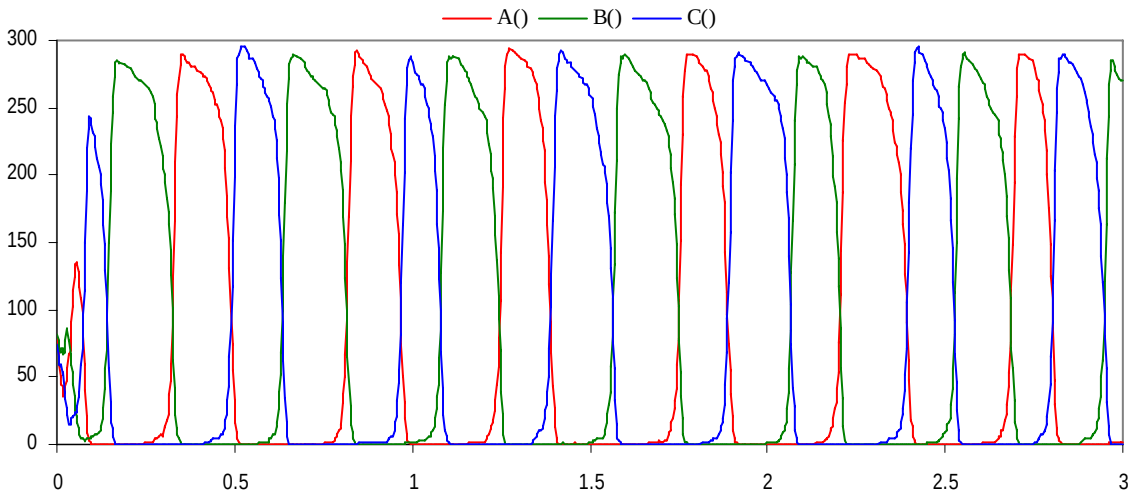
Hysteric 3-Way Groupies

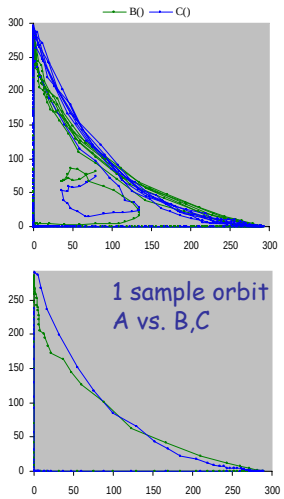





```
directive sample 3.0 1000
directive plot A(); B(); C()

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 Automata 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:

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

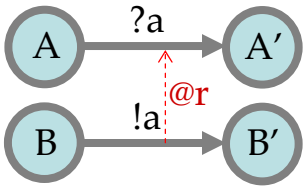
the "r" is given by Michaelis-Menten (approximated steady-state) laws:

$E + S \leftrightarrow ES$

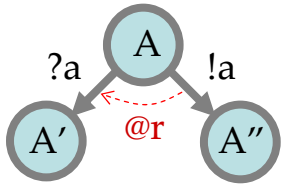
$ES \rightarrow P + E$

Automata to Chemistry

$A \rightarrow^{r} A'$

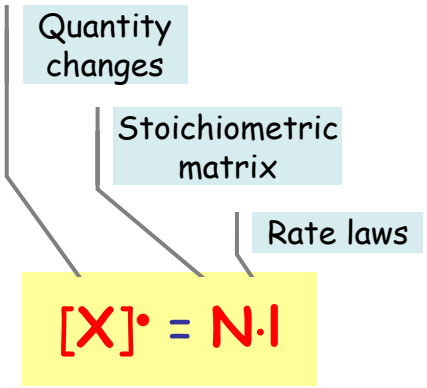


$A+B \rightarrow^{r} A'+B'$

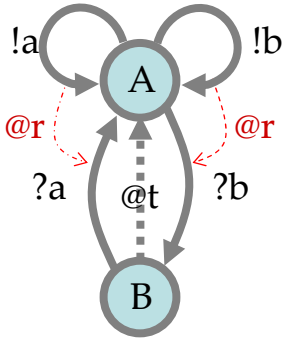


$A+A \rightarrow^{2r} A'+A''$

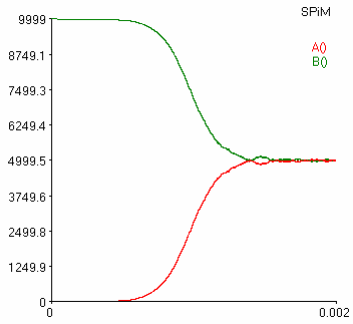
From Reactions to ODEs



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()
```

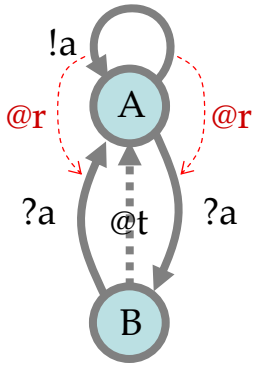


$B \rightarrow^{t} A$

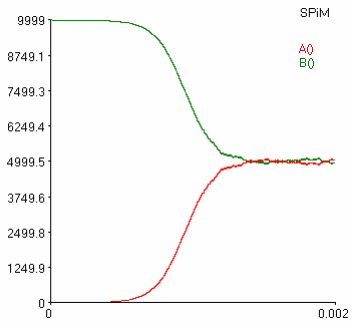
$A+B \rightarrow^{r} A+A$

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

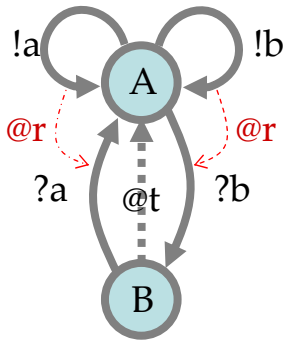
Same chemistry, hence equivalent automata




```
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

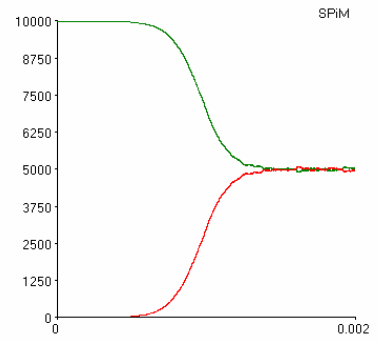


τ: B $\rightarrow^{t}$ A

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

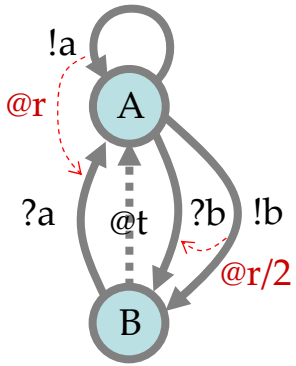
b: A+A $\rightarrow^{2r}$ A+B

```
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()
```



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

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

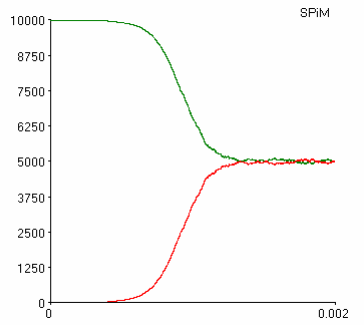


τ: B $\rightarrow^{t}$ A

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

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

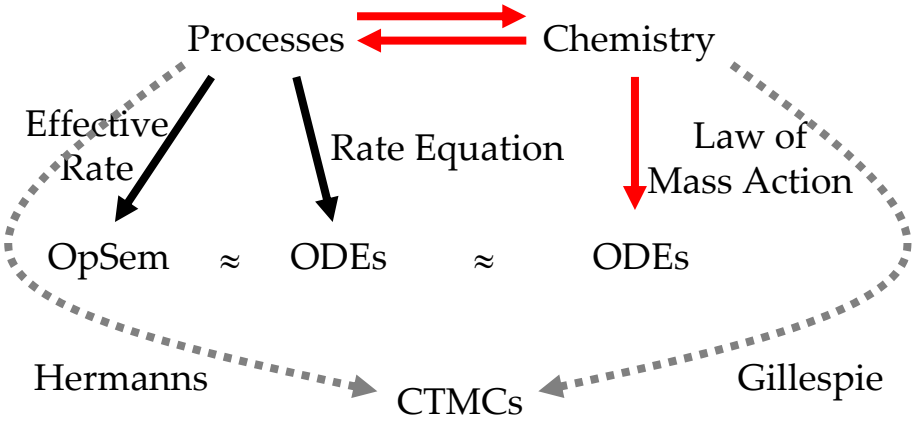
```
directive sample 0.002 10000
directive plot A(); B()

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()
```



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

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

Relationships

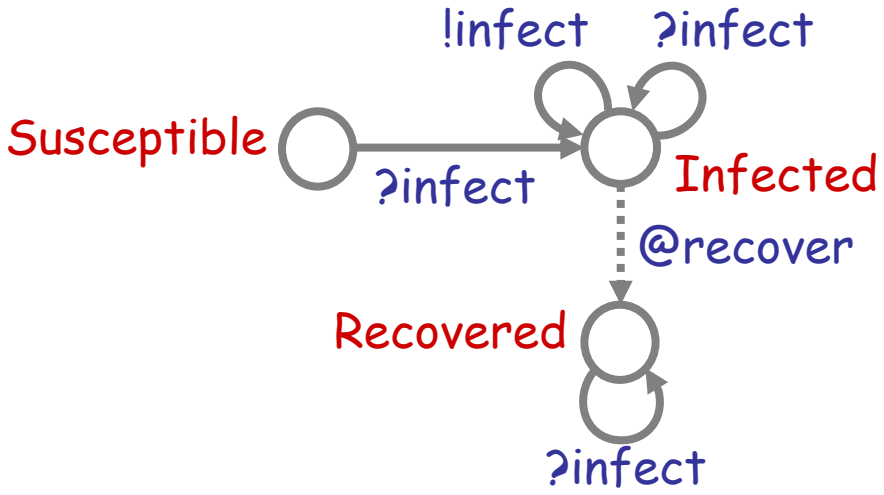


Epidemics

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



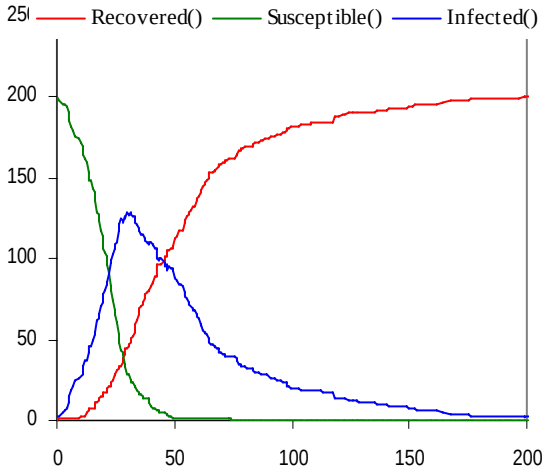

```
directive sample 500.0 1000
directive plot Recovered(); Susceptible(); Infected()

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

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

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

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

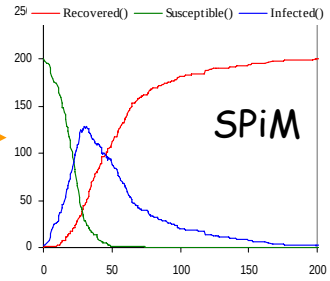
Automata match the standard ODE model!

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

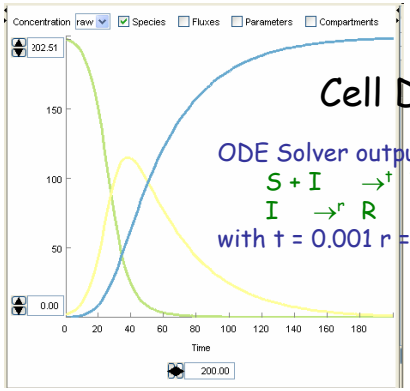
$$\frac{dI}{dt} = aIS - bI$$

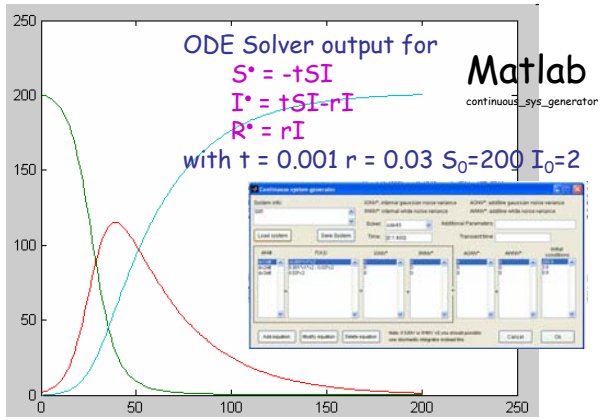
$$\frac{dR}{dt} = bI$$

(the Kermack-McKendrick, or SIR model)

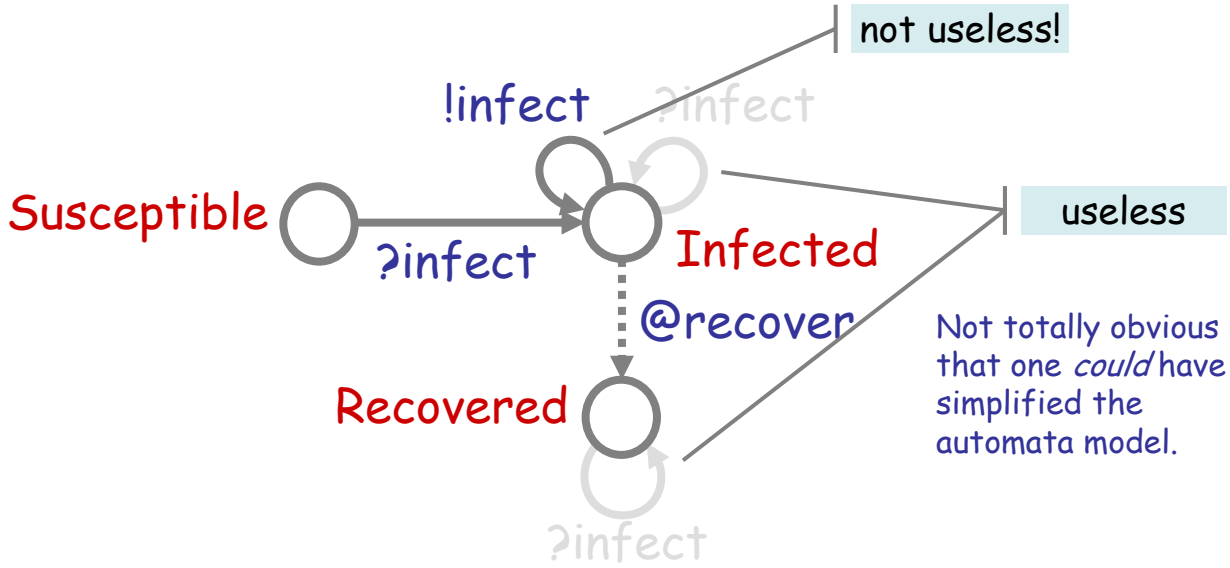



```
directive sample 500.0 1000
directive plot Recovered(); Susceptible(); Infected()

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.

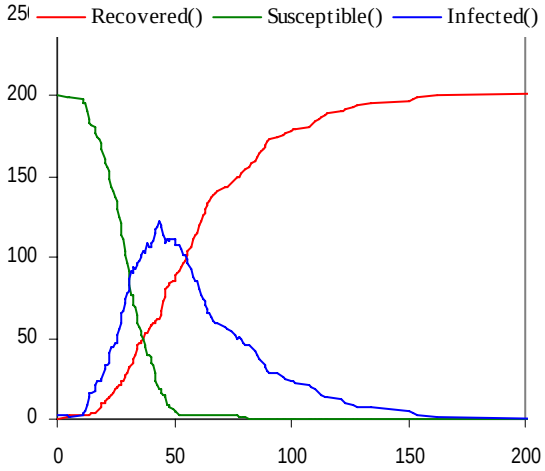
$$S = ?i_{(t)};I$$
$$I = !i_{(t)};I \oplus \tau_r;R$$
$$R = 0$$

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

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

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())
```



Polymerization

Bidirectional
Polymerization

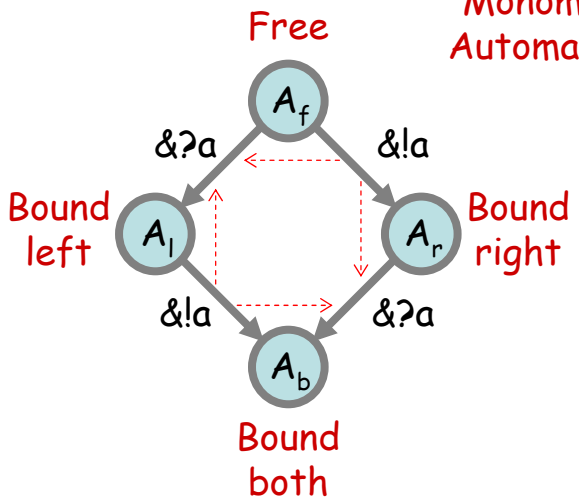
new c@μ new stop@1.0

A_{free} =
  !c($^{\nu}rht_{\lambda}$); A_{brht}(rht)) +
  ?c(lft); A_{blft}(lft)

A_{blft}(lft) =
  !c($^{\nu}rht_{\lambda}$); A_{bound}(lft,rht))

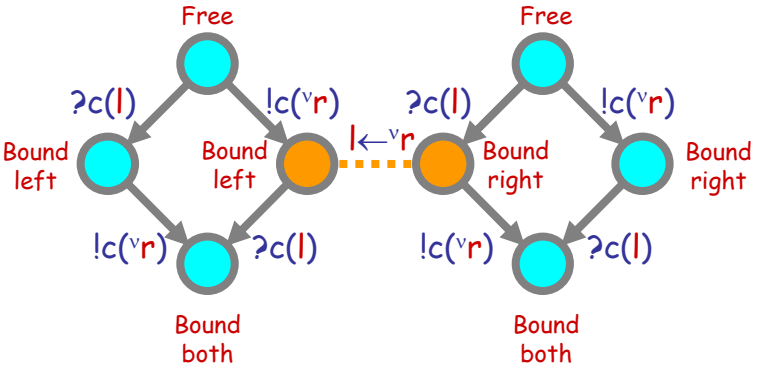
A_{brht}(rht) =
  ?c(lft); A_{bound}(lft,rht)

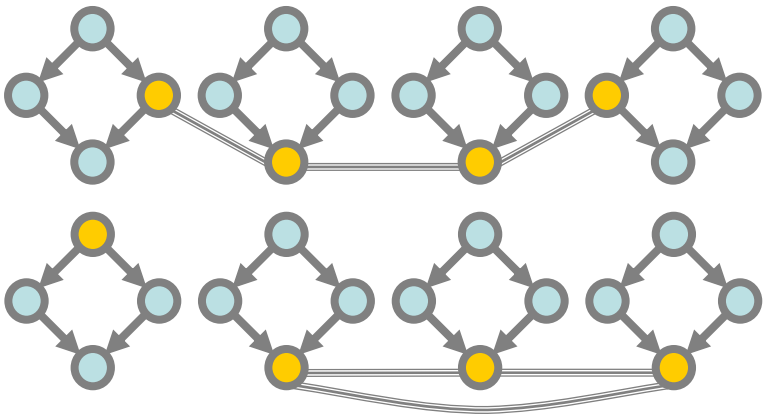
A_{bound}(lft,rht) = ?stop

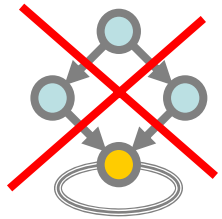


Polymerization is iterated complexation.

Communicating Automata
Bound output !c($^{\nu}$r) and input ?c(l) on automata transitions to model complexation &!c, &?c





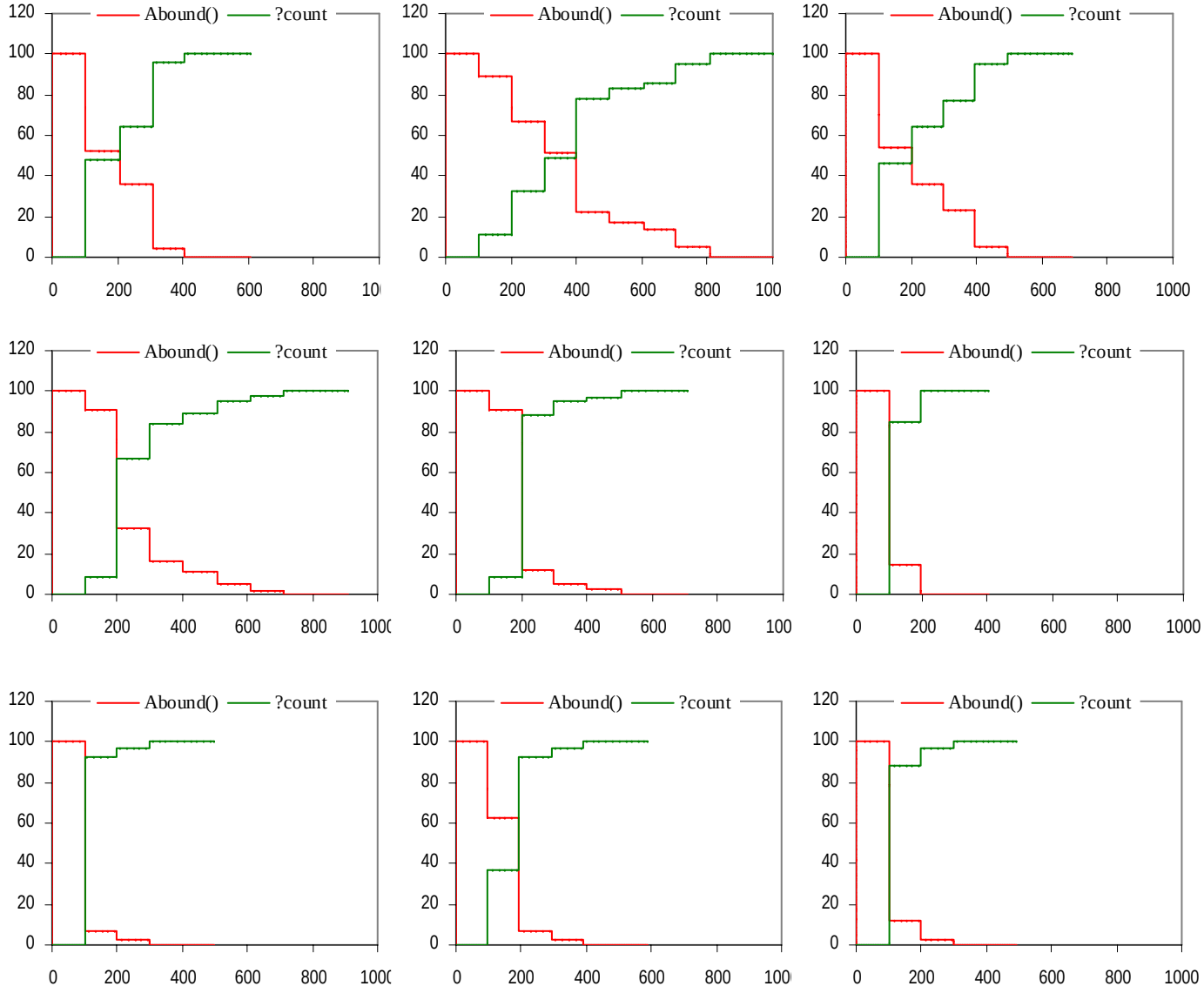


```
directive sample 10000.0
directive plot Afree(); Ablft(); Abrht(); Abound()

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

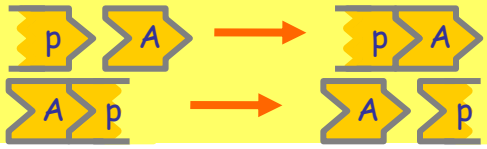
Scanning and counting the size of the circular polymers (by a cheap trick).
Polymer formation is complete within 10t; then a different polymer is scanned every 100t.




```
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

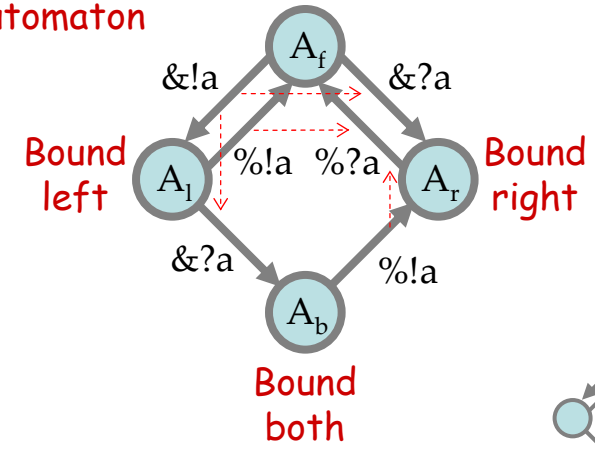
A_{free} =
  !c($^{\nu}$lft$_{\lambda}$); A_{blft}(lft)) +
  ?c(rht); A_{brht}(rht)

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

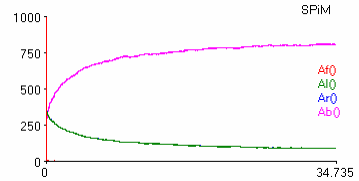
A_{brht}(rht) =
  ?rht; A_{free}

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

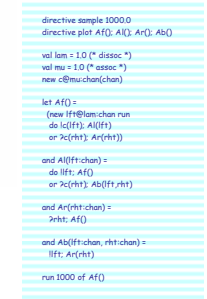
Monomer Automaton

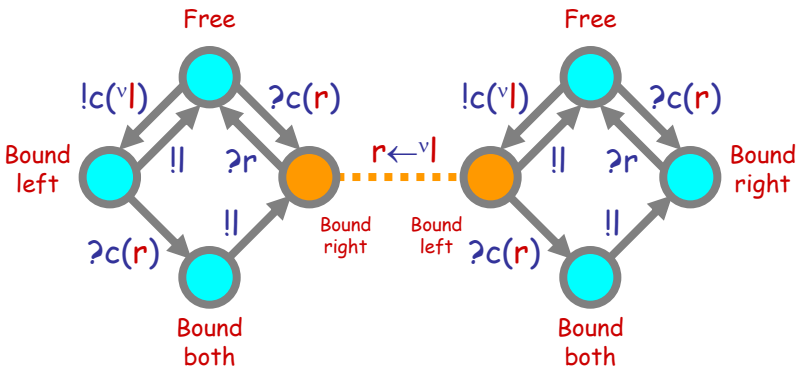


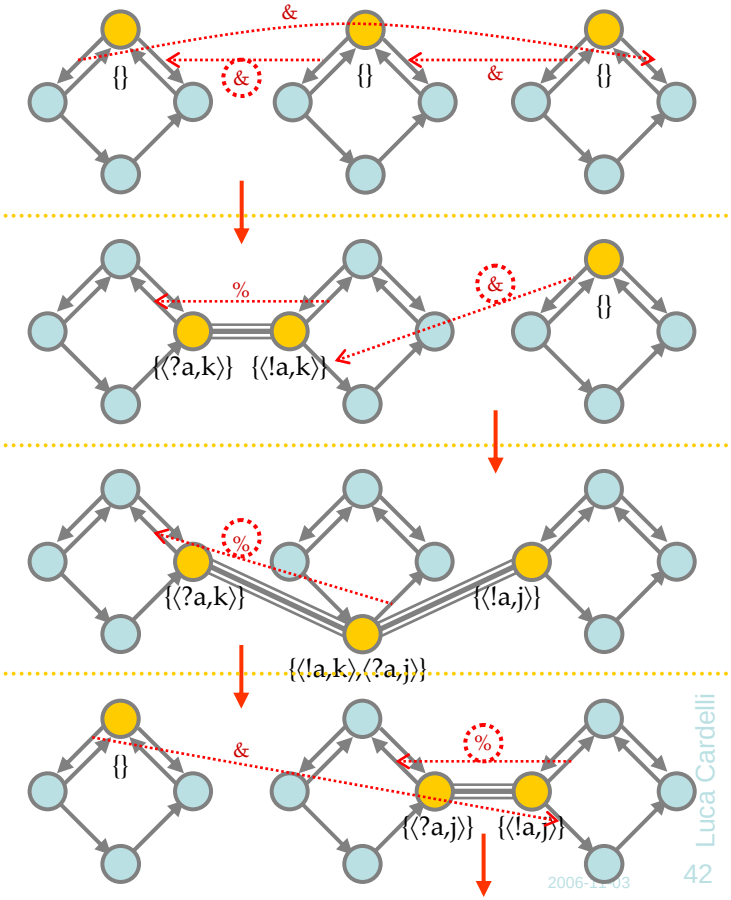
Complexation &!c, &?c
Decomplexation %!c, %?c



1000 monomers settle to ~100 polymers of size ~10

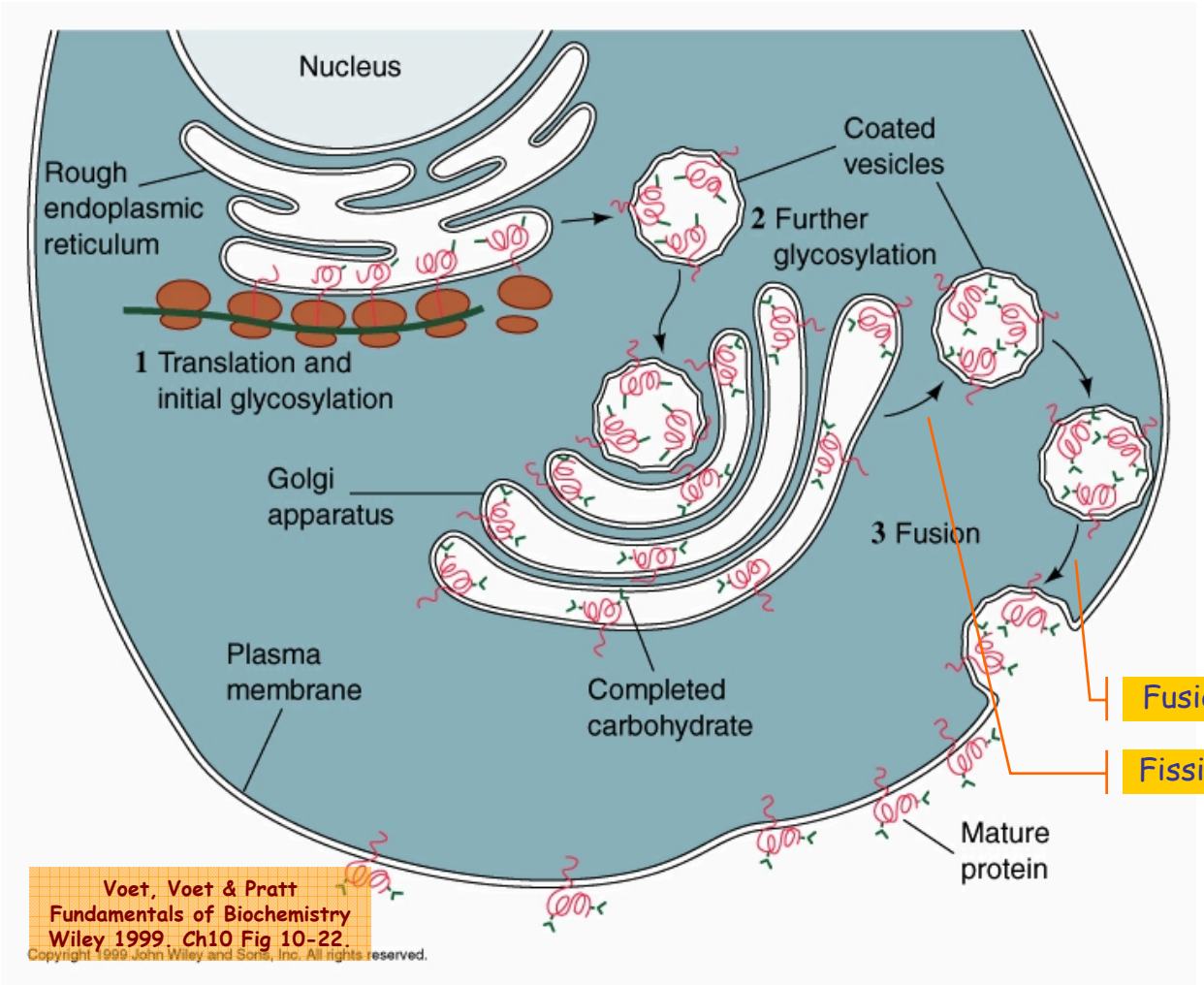






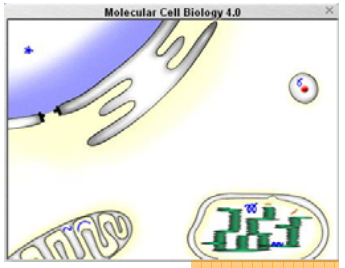
Membranes

The Membrane Machine



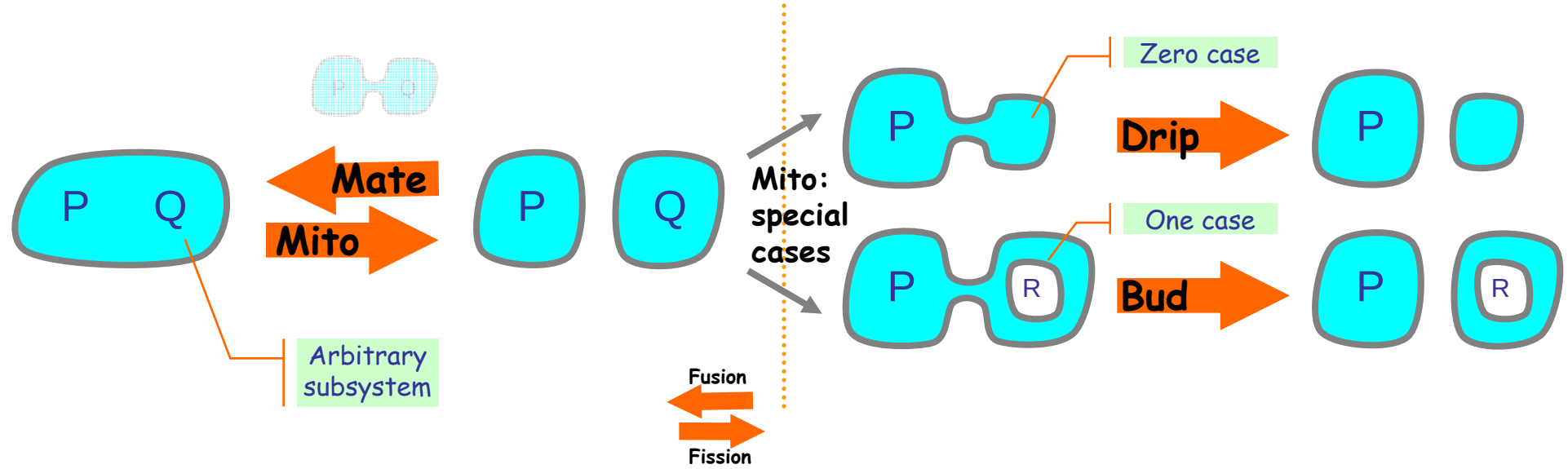
Voet, Voet & Pratt
Fundamentals of Biochemistry
Wiley 1999. Ch10 Fig 10-22.

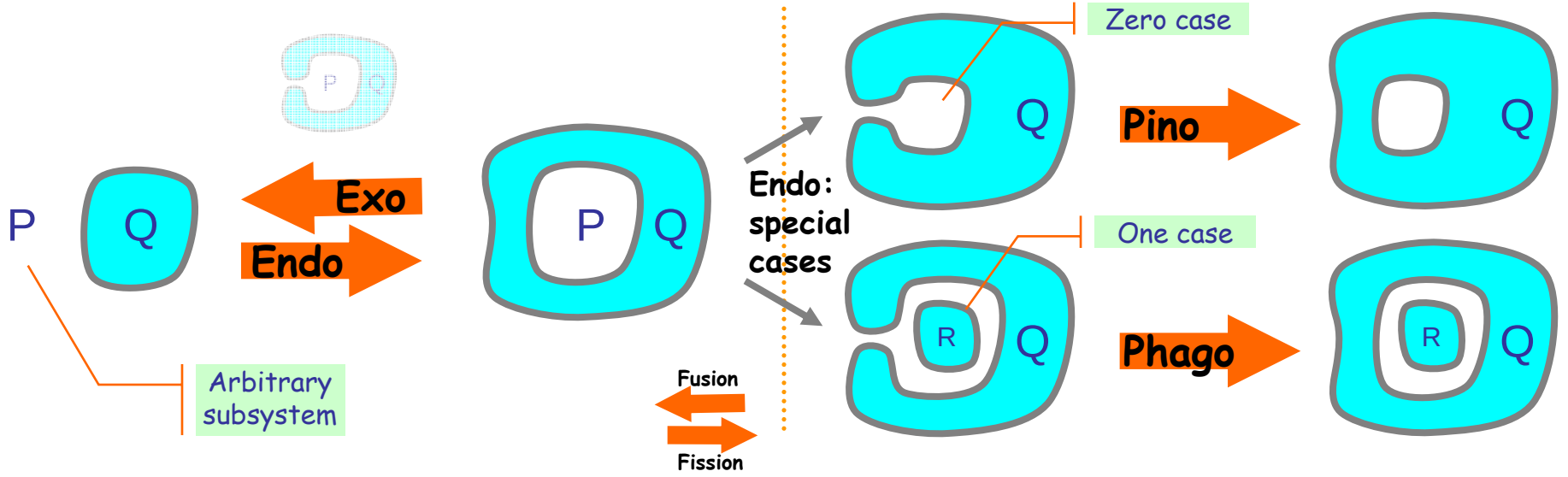
Molecular transport and transformation through dynamic compartment fusion and fission.



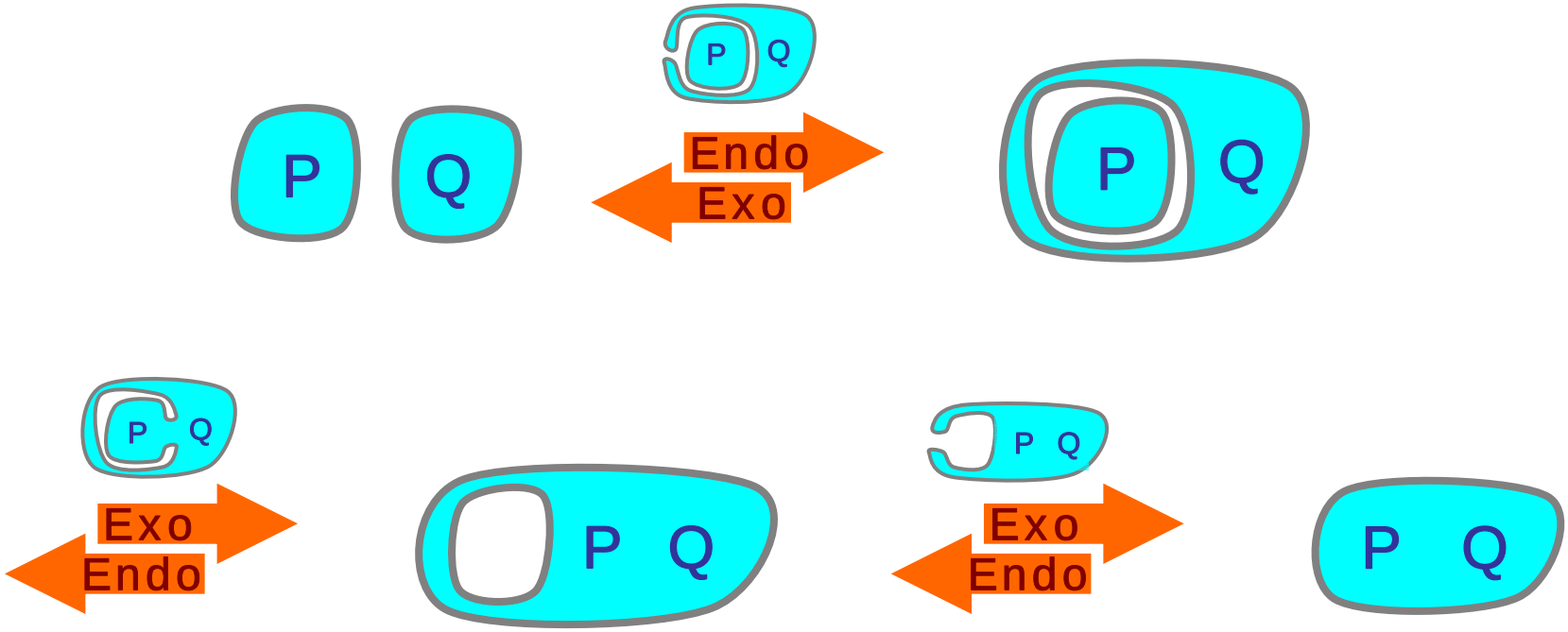
Taken from MCB CD

The Membrane Machine "Instruction Set"

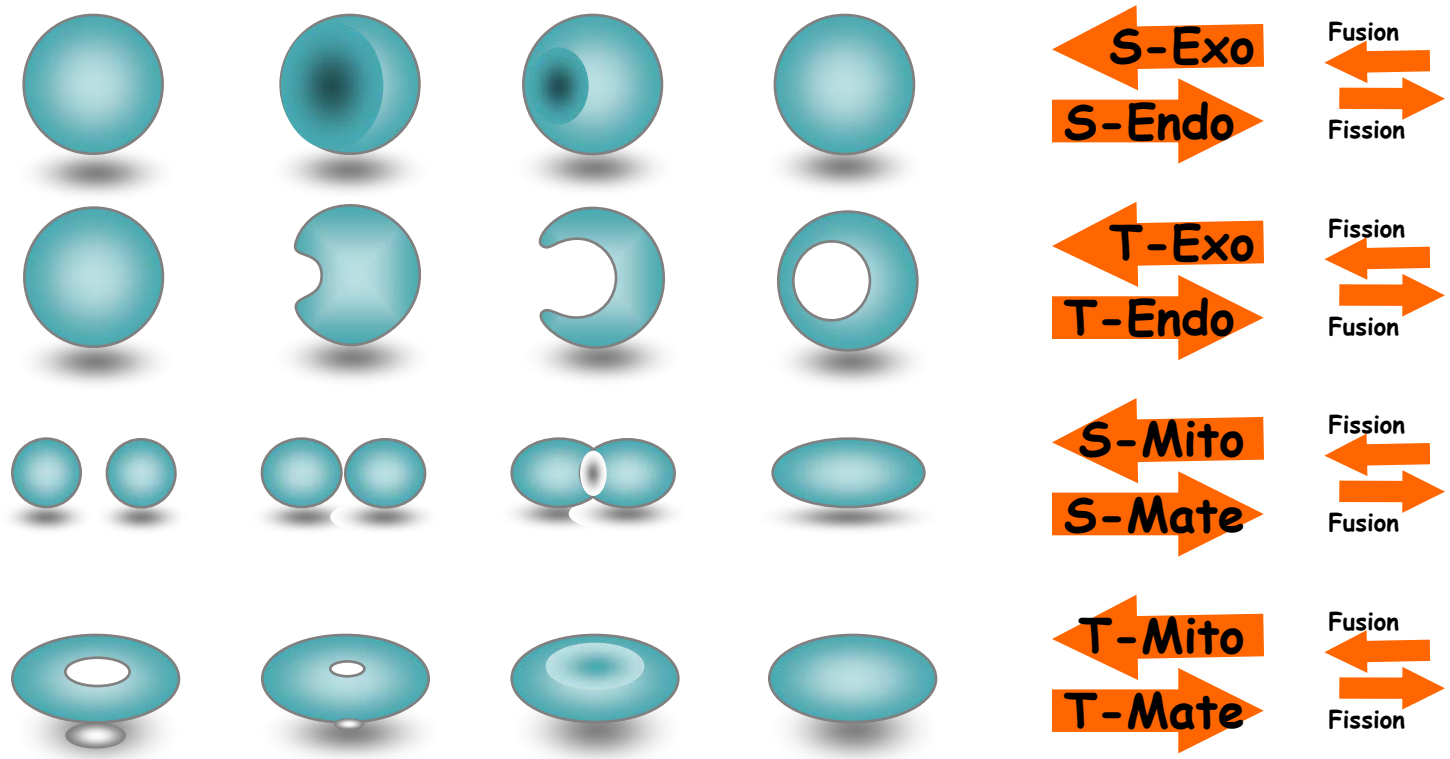




Mito/Mate by 3 Endo/Exo

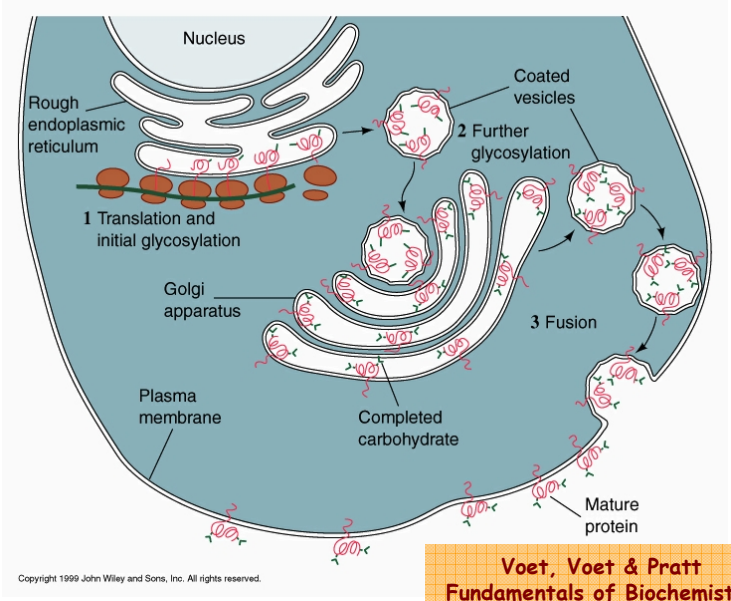


… in 3D


S-Exo
S-Endo
Fusion
Fission
T-Exo
T-Endo
Fission
Fusion
S-Mito
S-Mate
Fission
Fusion
T-Mito
T-Mate
Fusion
Fission

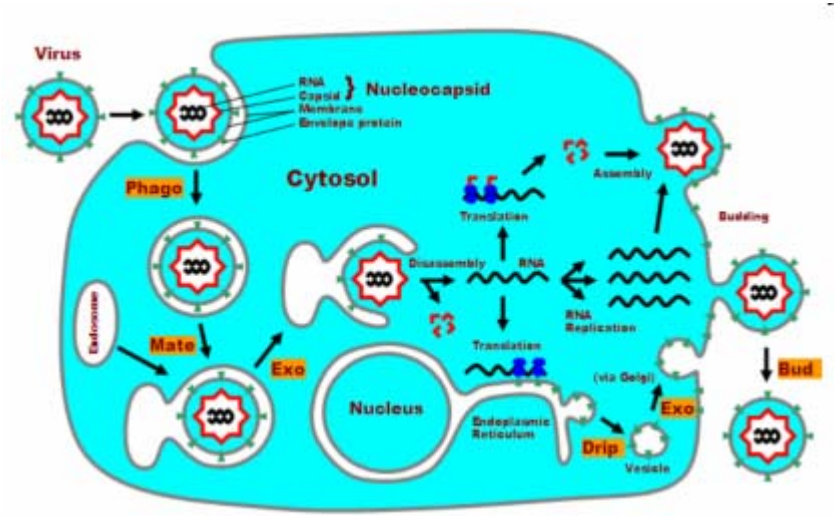
Membrane Algorithms

Protein Production and Secretion



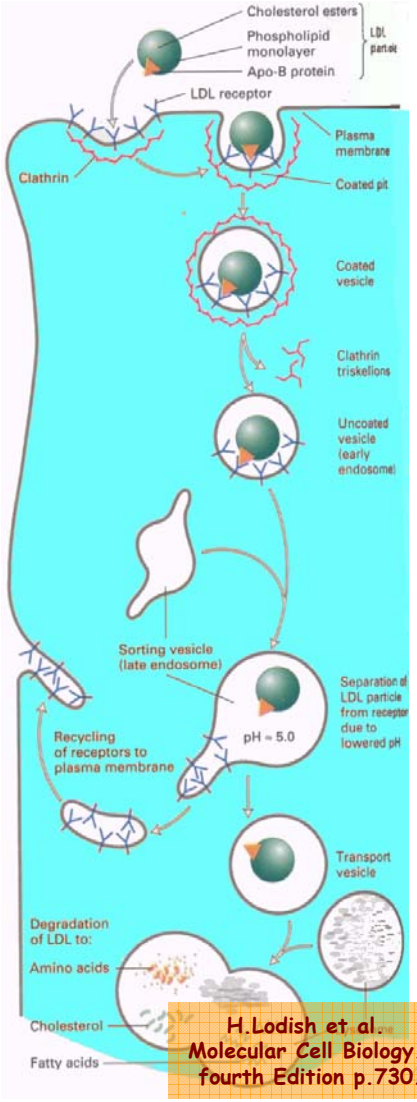
Voet, Voet & Pratt
Fundamentals of Biochemistry
Wiley 1999. Ch10 Fig 10-22.

Viral Replication



Adapted from: B.Alberts et al.
Molecular Biology of the Cell
third edition p.279.

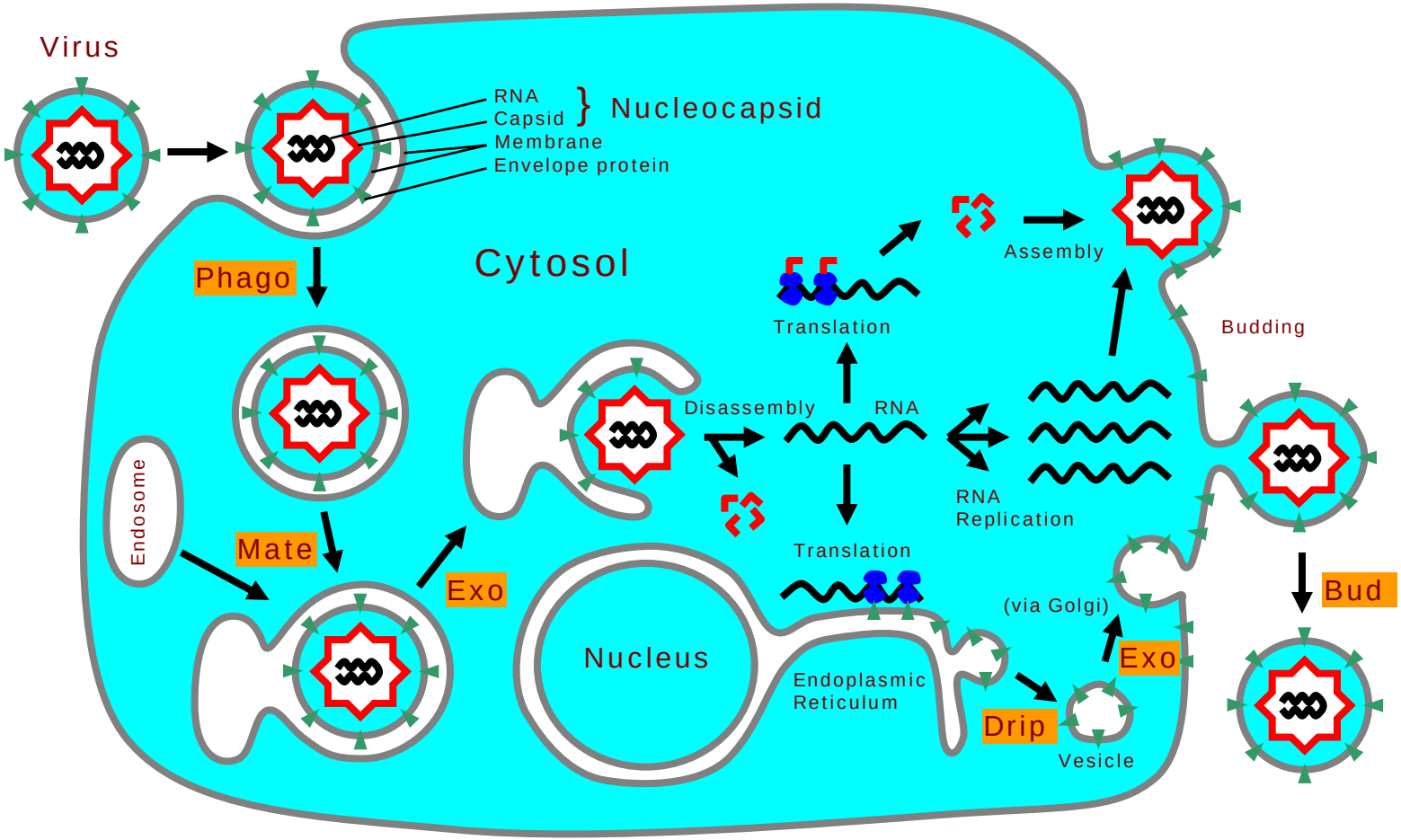
LDL-Cholesterol Degradation



H.Lodish et al.
Molecular Cell Biology.
fourth Edition p.730.

Viral Reproduction

Modeling "the whole process"



[MBC p.279]
annotated

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.