

Molecules as Automata

Representing Biochemical Systems as
Collectives of Interacting Automata

Luca Cardelli

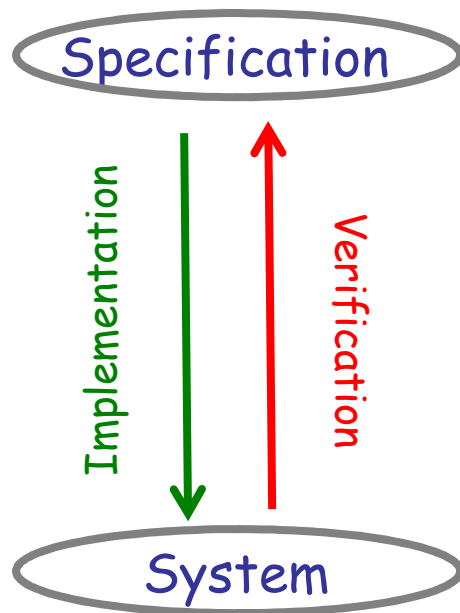
Microsoft Research

WADT, Pisa, 2008-06-14

<http://LucaCardelli.name>

Formal Methods vs. Scientific Method

Direct
Engineering

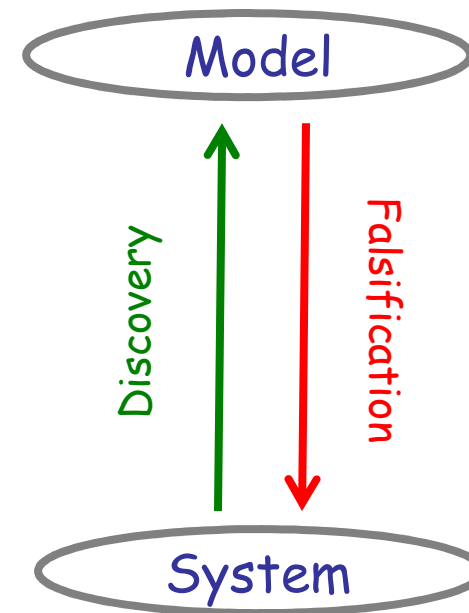


Formal Methods

<- Abstract ->

<- Concrete ->

Reverse
Engineering



Scientific Method

Formal Methods vs. Scientific Method

Direct Engineering

Reverse Engineering

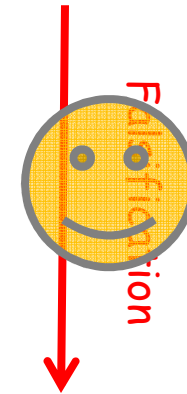
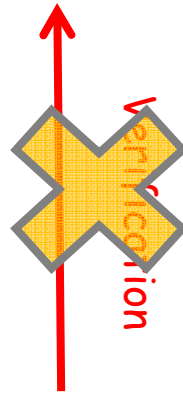
"Truth"

"Debugging"

Specification

<- Abstract ->

Model



System

<- Concrete ->

System

"Debugging"

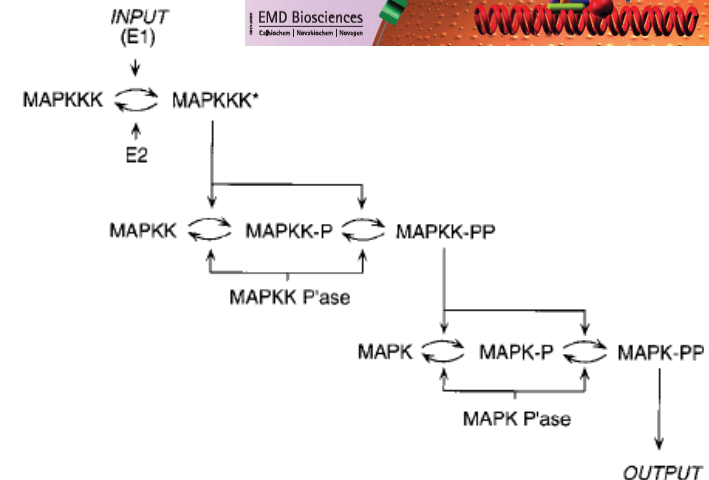
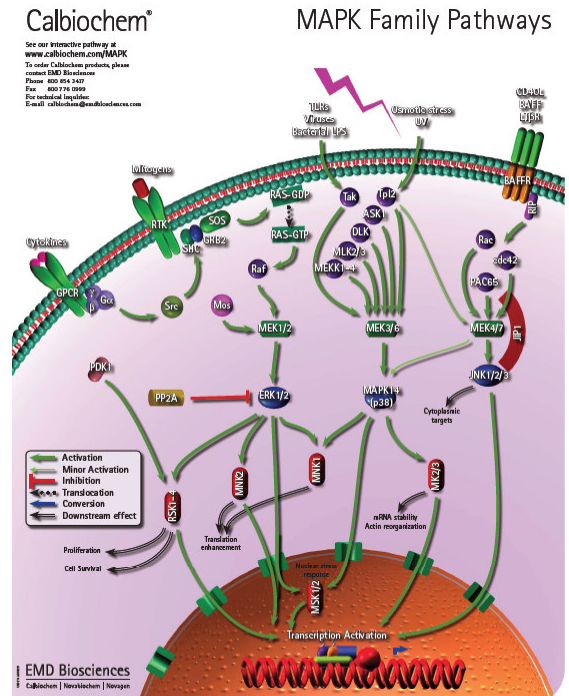
"Truth"

Formal Methods

Scientific Method

Motivation: Cells Compute

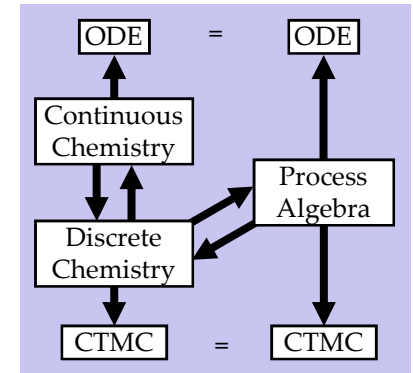
- No survival without computation!
 - Finding food
 - Avoiding predators
- How do they compute?
 - Unusual computational paradigms.
 - Proteins: do they work like electronic circuits?
 - Genes: what kind of software is that?
- Signaling networks
 - Clearly "information processing"
 - They are "just chemistry": 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.
- General models of biological computation
 - What are the appropriate ones?



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.

Aims

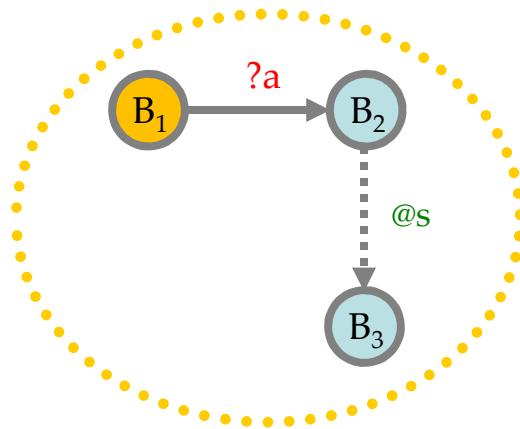
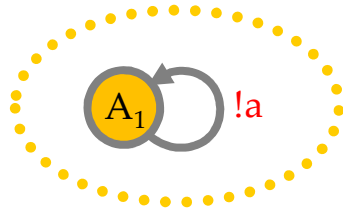
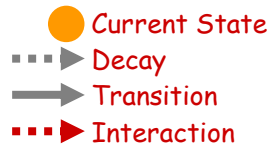
- **Connections between modeling approaches**
 - Connecting the **discrete/concurrent/stochastic/molecular** approach
 - to the **continuous/sequential/deterministic/population** approach
- **Connecting syntax with semantics**
 - **Syntax** = model presentation (equations/programs/diagrams/blobs etc.)
 - **Semantics** = state space (generated by the syntax)
- **Ultimately, connections between analysis techniques**
 - We need (and sometimes have) good semantic techniques to analyze state spaces (e.g. calculus, but also increasingly modelchecking)
 - But we need equally good syntactic techniques to structure complex models (e.g. compositionality) and analyze them (e.g. process algebra)



(Macro-) Molecules as (Interacting) Automata

- Concurrent (math is based on processes, not functions)
 - Asynchronous (no global clock)
 - Stochastic (or nondeterministic)
 - Stateful (e.g. phosphorylation state)
 - Discrete (transitions between states)
 - Interacting (an "interaction" can be pretty much anything you want that changes molecular state)
-
- Based on work on process algebra and biological modeling; see references in related papers.

Interacting Automata



A_1 is a *state*

a is a *channel* i.e. a named *interaction interface* (e.g. a surface patch)

$?a, !a$ indicate any *complementarity* of interaction (e.g. charge)

$?a, !a$ indicate *complementary actions*,

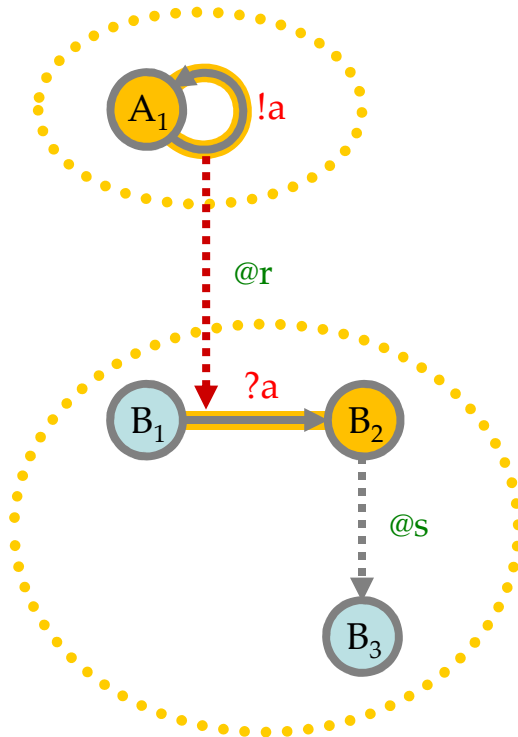
$@r, @s$ are rates

Interacting Automata

- Current State
- ⋯→ Decay
- Transition
- ⋯▶ Interaction

Kinetic laws:

Two complementary actions may result in an interaction.



- A_1 is a *state*
- a is a *channel* i.e. a named *interaction interface* (e.g. a surface patch)
- $?,!$ indicate any *complementarity* of interaction (e.g. charge)
- $?a, !a$ indicate *complementary actions*, joined by an interaction arrow ⋯▶
- $@r, @s$ are rates

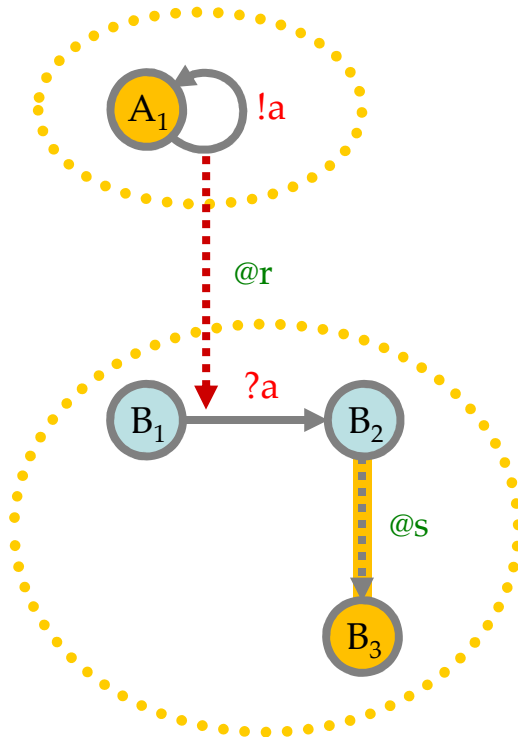
Interacting Automata



Kinetic laws:

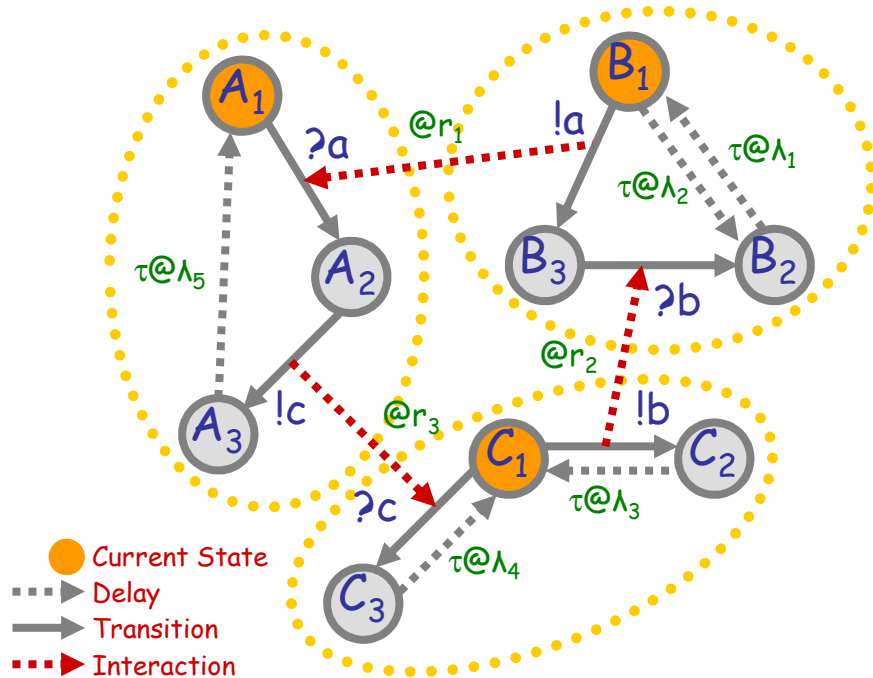
Two complementary actions may result in an interaction.

A decay may happen spontaneously.



- A_1 is a *state*
- a is a *channel* i.e. a named *interaction interface* (e.g. a surface patch)
- $?,!$ indicate any *complementarity* of interaction (e.g. charge)
- $?a, !a$ indicate *complementary actions*, joined by an interaction arrow - - - ->
- $@r, @s$ are rates

Interacting Automata



Interactions have rates. Actions DO NOT have rates.

The equivalent process algebra model

```

new a@r1
new b@r2
new c@r3
} Communication channels

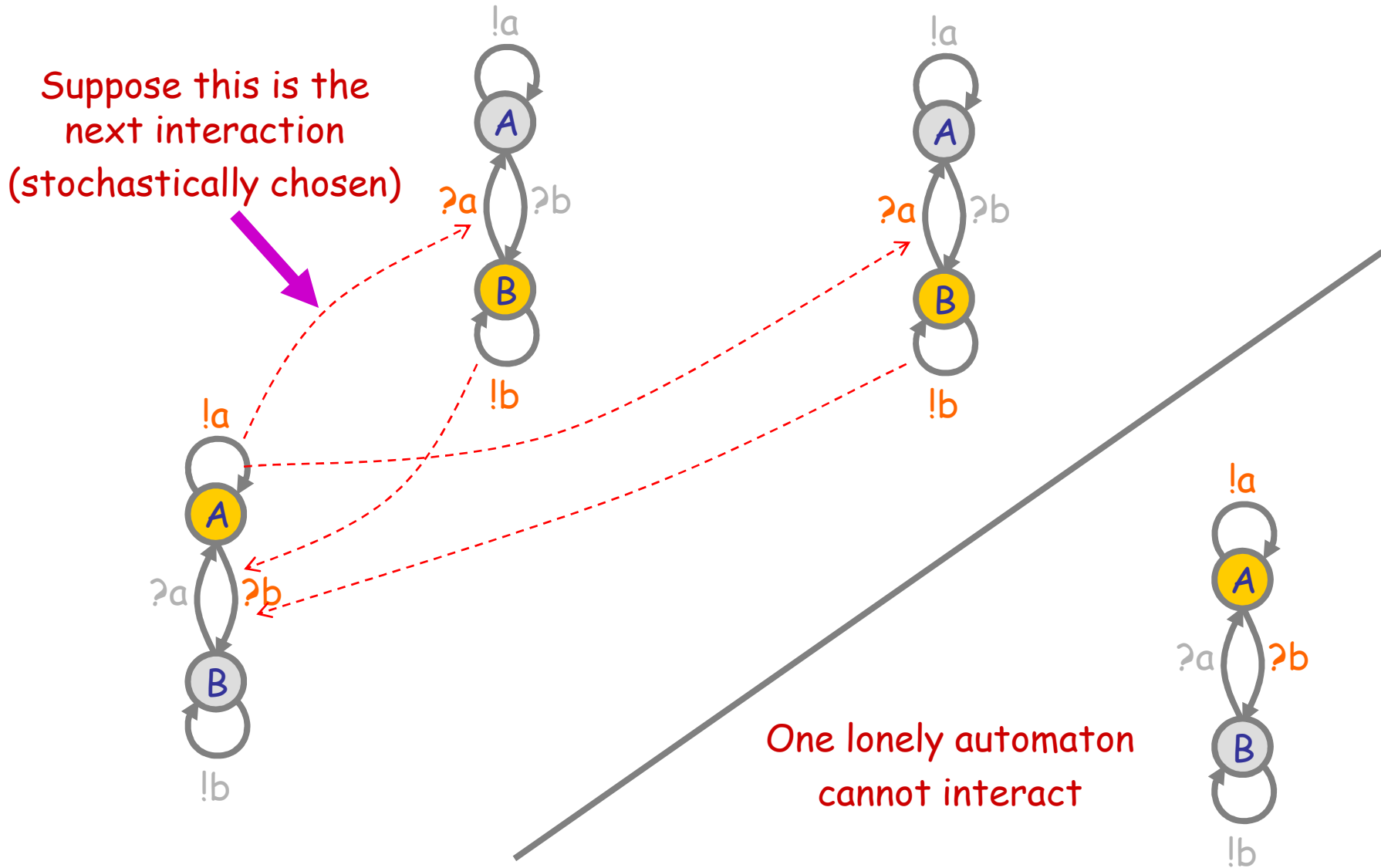
A1 = ?a; A2
A2 = !c; A3
A3 = τ@λ5; A1
} Automata

B1 = τ@λ2; B2 + !a; B3
B2 = τ@λ1; B1
B3 = ?b; B2
}

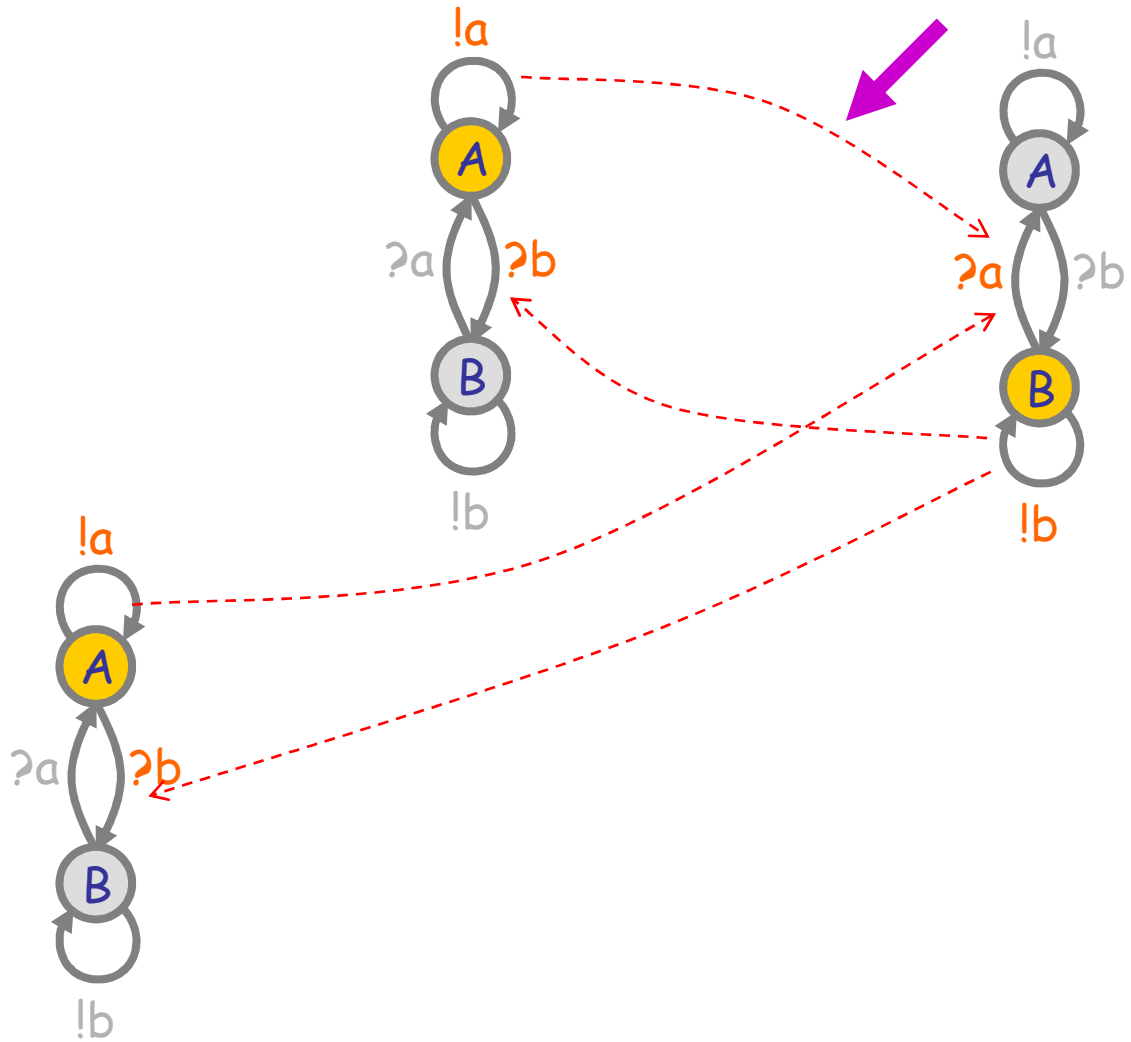
C1 = !b; C2 + ?c; C3
C2 = τ@λ3; C1
C3 = τ@λ4; C2
}

A1 | B1 | C1 } The system and initial state
    
```

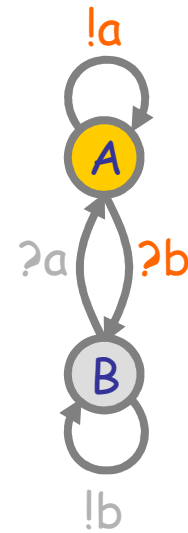
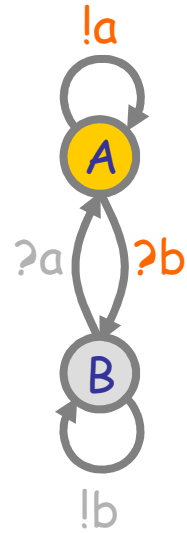
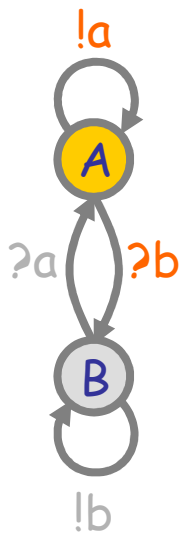
Interactions in a Population



Interactions in a Population

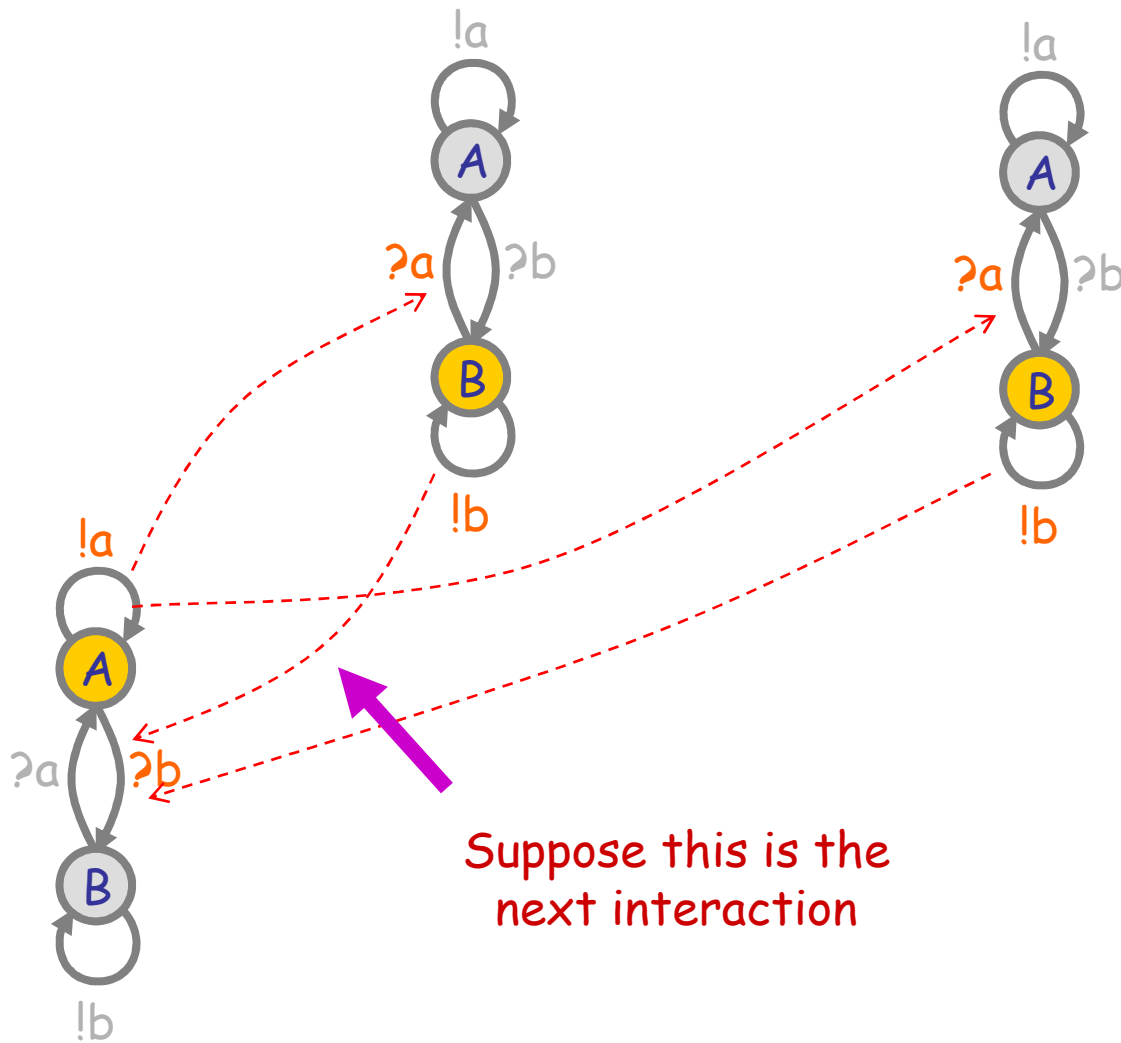


Interactions in a Population

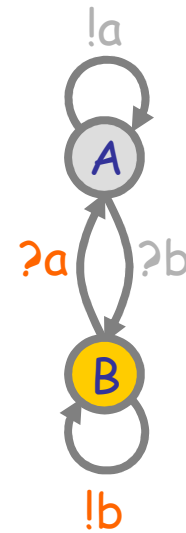
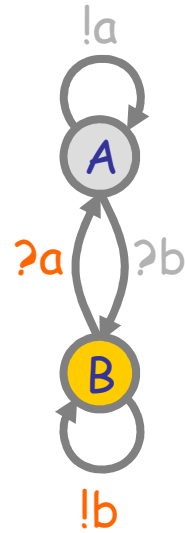
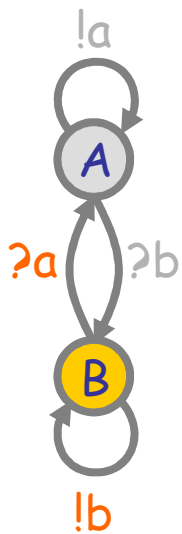


All-A stable
population

Interactions in a Population (2)



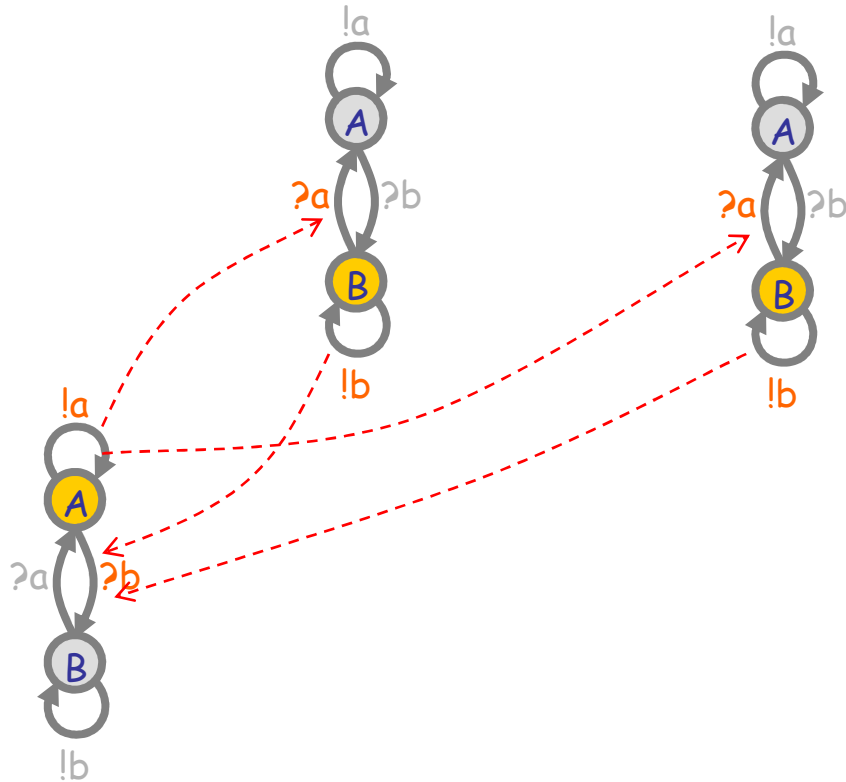
Interactions in a Population (2)



All-B stable population

Nondeterministic population behavior ("multistability")

CTMC Semantics



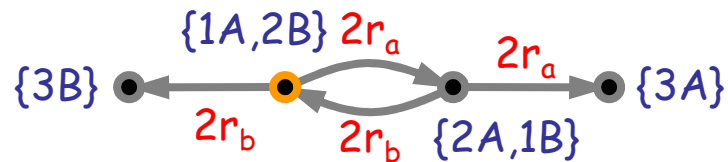
CTMC
(homogeneous) Continuous Time Markov Chain

- directed graph with no self loops
- nodes are system states
- arcs have transition rates

Probability of holding in state A:

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

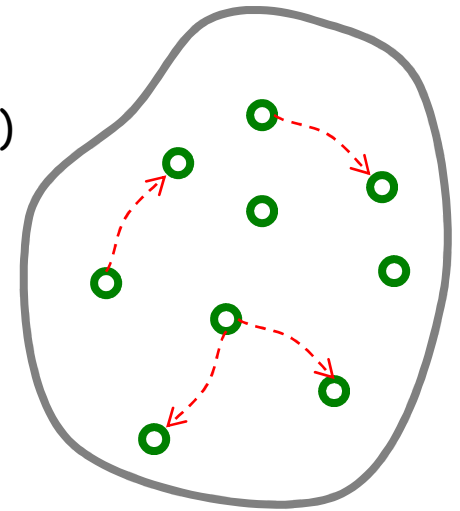
in general, $\Pr(H_A > t) = e^{-Rt}$ where R is the sum of all the exit rates from A



CTMC

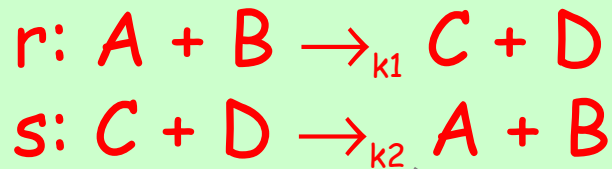
Stochastic Automata Collectives

- "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].

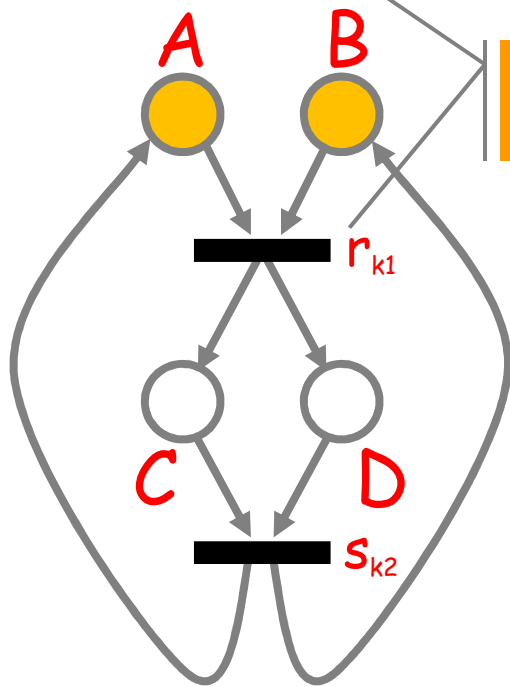


Chemistry vs. Automata

A process algebra (chemistry)



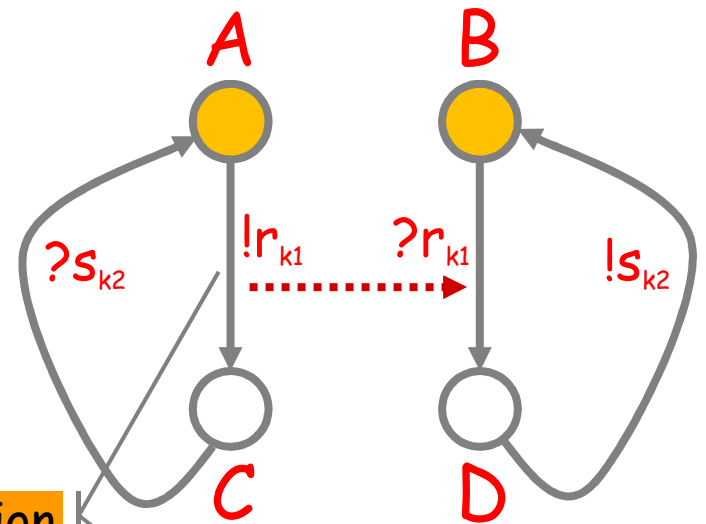
Does A become C or D?



Reaction oriented

1 line per reaction

A different process algebra (automata)



Interaction oriented

1 line per component

$$A = !r_{k_1}; C$$

$$C = ?s_{k_2}; A$$

$$B = ?r_{k_1}; D$$

$$D = !s_{k_2}; B$$

A becomes C not D!

The same "model"

Maps to a CTMC

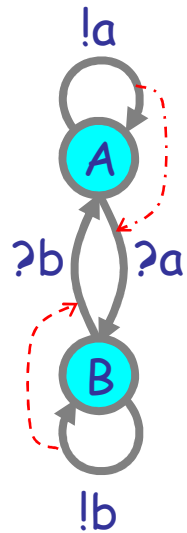
Maps to a CTMC

A Petri-Net-like representation. Precise and dynamic, but not modular, scalable, or maintainable.

A compositional graphical representation (precise, dynamic *and* modular) and the corresponding calculus.

Groupies and Celebrities

Groupies and Celebrities



Celebrity

(does not want to be like somebody else)

```
directive sample 1.0 1000
directive plot A(); B()

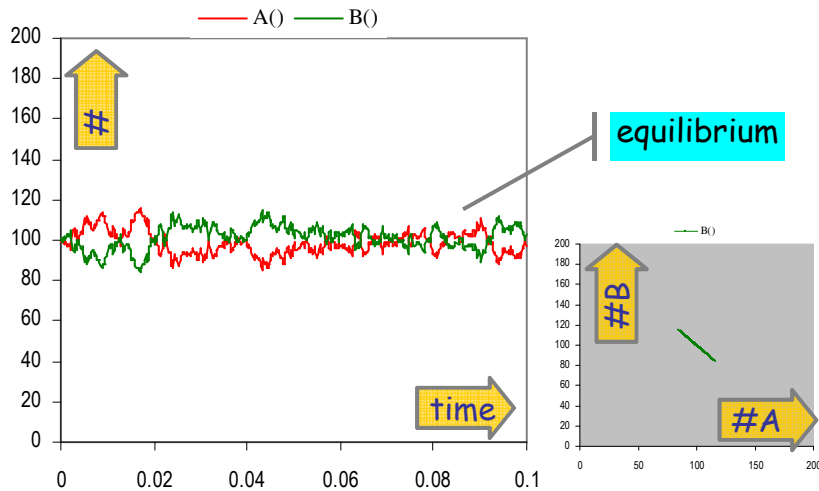
new a@1.0:chan()
new b@1.0:chan()

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

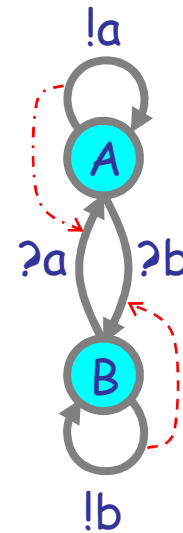
run 100 of (A() | B())
```

a@1.0
b@1.0

A stochastic collective of celebrities:



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 1.0 1000
directive plot A(); B()

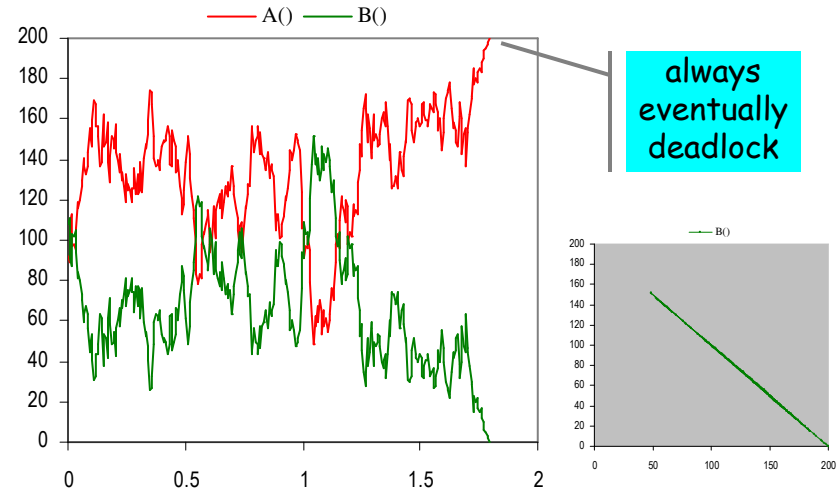
new a@1.0:chan()
new b@1.0:chan()

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

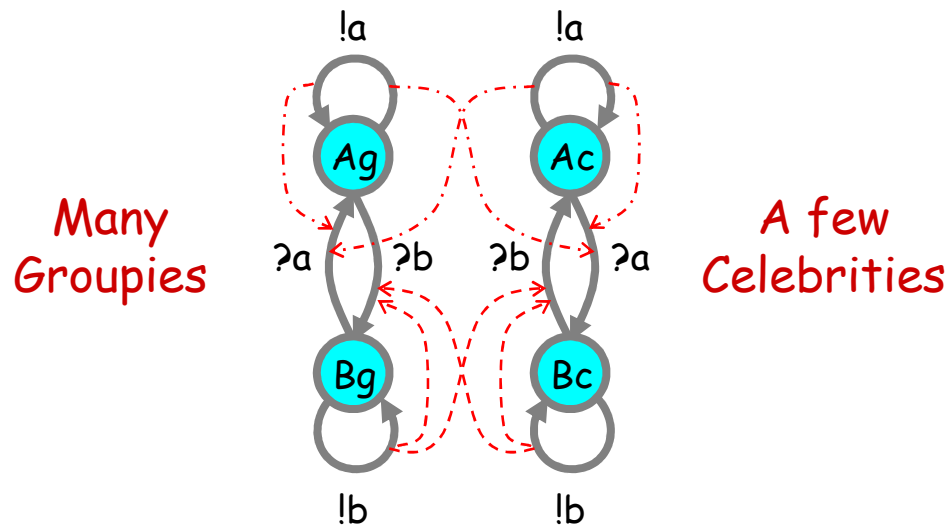
A stochastic collective of groupies:



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



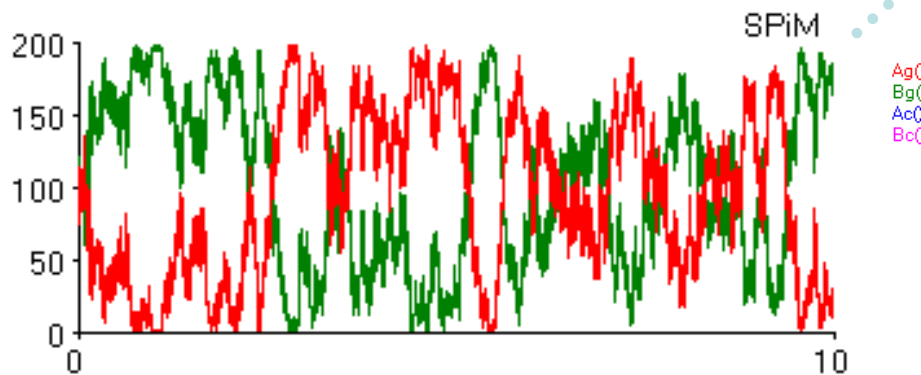
```
directive sample 10.0
directive plot Ag(); Bg(); Ac(); Bc()

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())
```



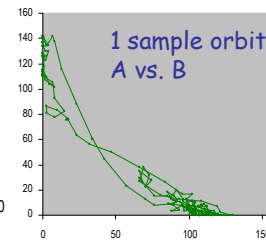
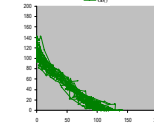
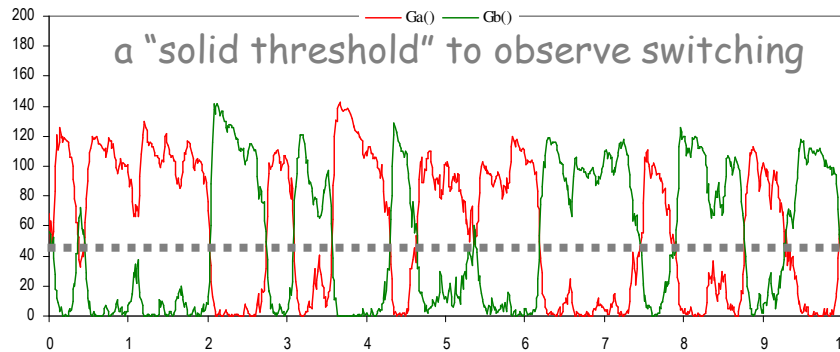
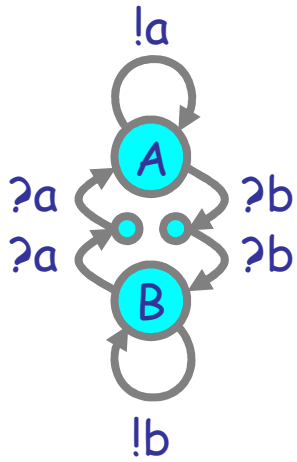
never
deadlock

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.



```
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()

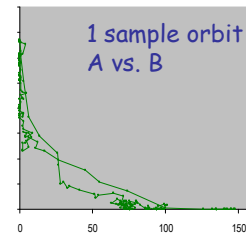
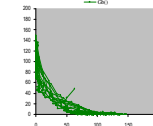
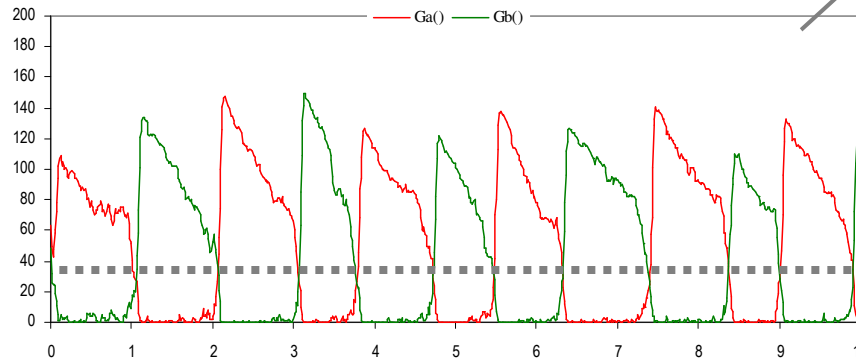
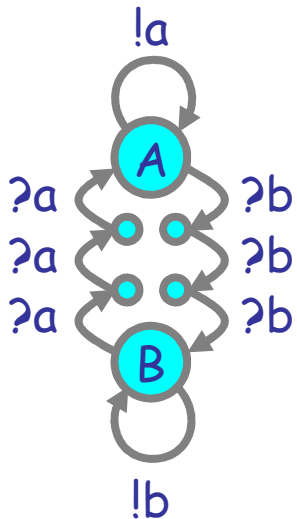
run 100 of (Ga() | Gb())
run 1 of (Da() | Db())
```



(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()

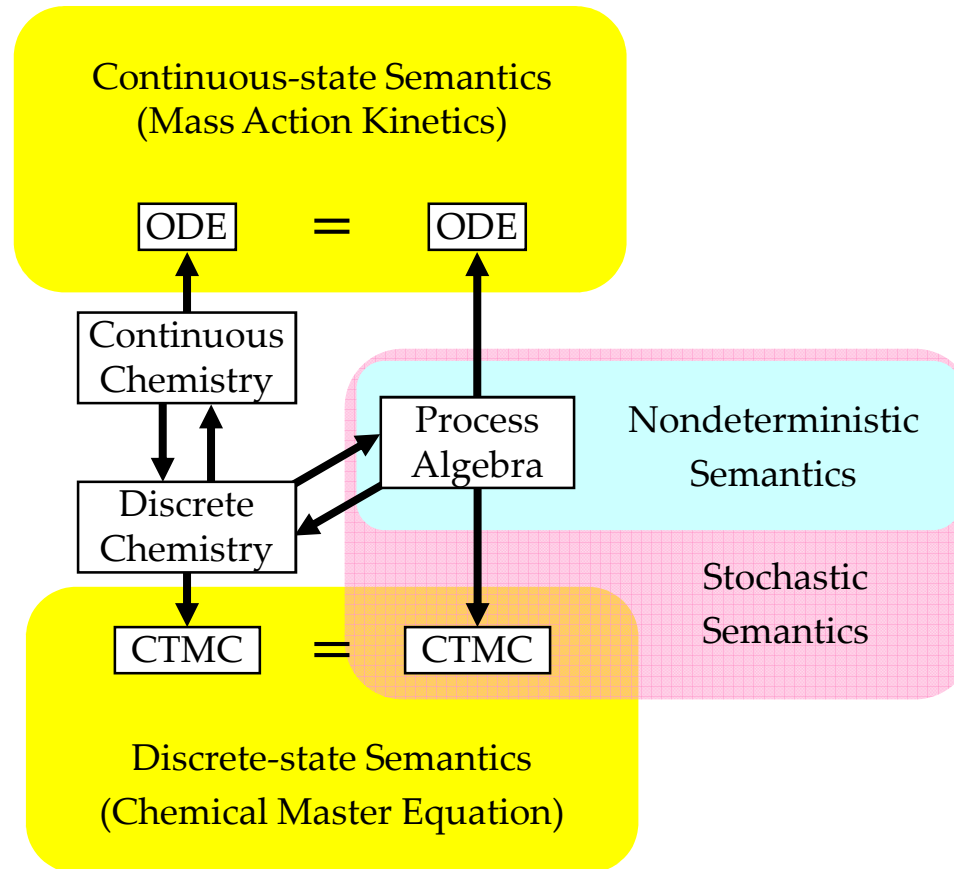
let Ga() = do !a; Ga() or ?b; ?b; ?b; Gb()
and Gb() = do !b; Gb() or ?a; ?a; ?a; Ga()

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

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

Semantics of Collective Behavior

The Two Semantic Sides of Chemistry

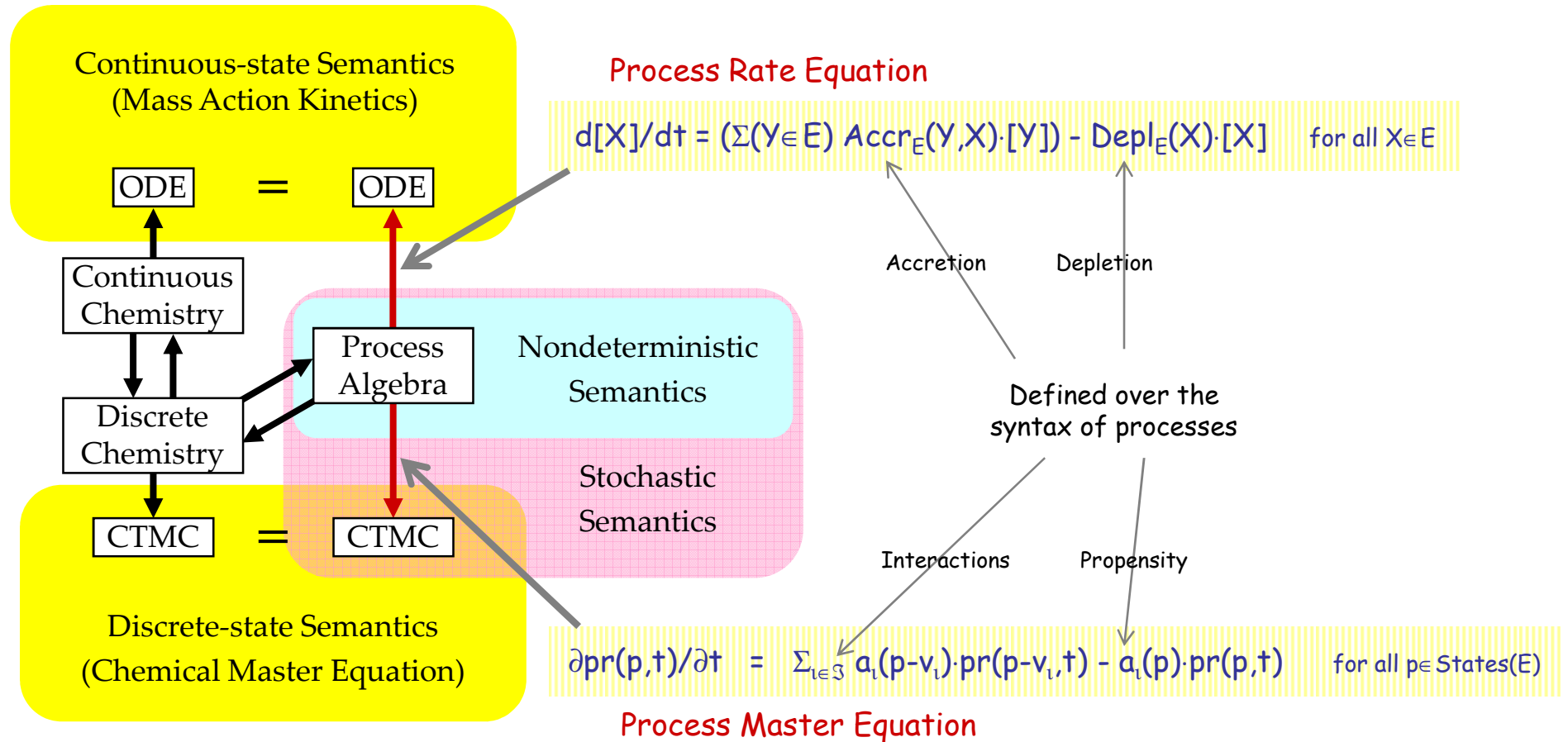


These diagrams commute via appropriate maps.

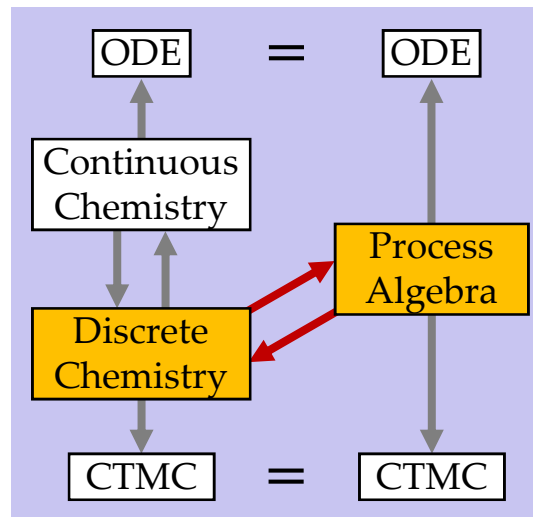
L. Cardelli: "On Process Rate Semantics" (TCS)

L. Cardelli: "A Process Algebra Master Equation" (QEST'07)

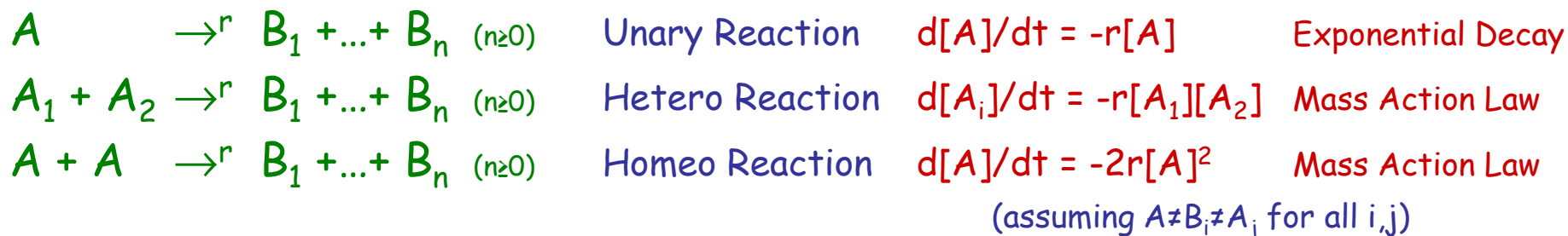
Quantitative Process Semantics



Stochastic Processes & Discrete Chemistry



Chemical Reactions



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:



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



Enzymatic reactions:



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



Reactions have rates. Molecules do not have rates.

Chemical Ground Form (CGF)

$E ::= O : X=M, E$

Reagents

$M ::= O : \pi; P \oplus M$

Molecules

$P ::= O : X | P$

Solutions

$\pi ::= \tau_{(r)} : ?a_{(r)} : !a_{(r)}$

Actions (delay, input, output)

$CGF ::= E, P$

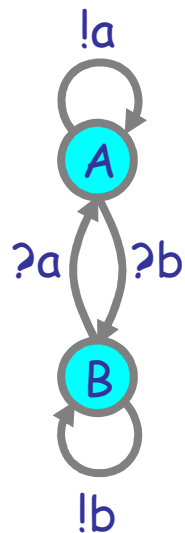
Reagents plus Initial Conditions

A stochastic subset of CCS
(no values, no restriction)

Interacting Automata
+ dynamic forking

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

\oplus is stochastic choice (vs. + for chemical reactions)
O is the null solution ($P|O = O|P = P$)
and null molecule ($M \oplus O = O \oplus M = M$)
Each X in E is a distinct *species*
Each name a is assigned a fixed rate r: $\alpha_{(r)}$



Ex: Interacting Automata

(= finite-control CGFs: they use "|" only in initial conditions):

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

Automaton in state A


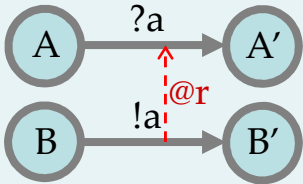
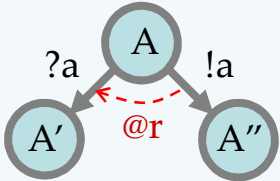
$B = !b; B \oplus ?a; A$

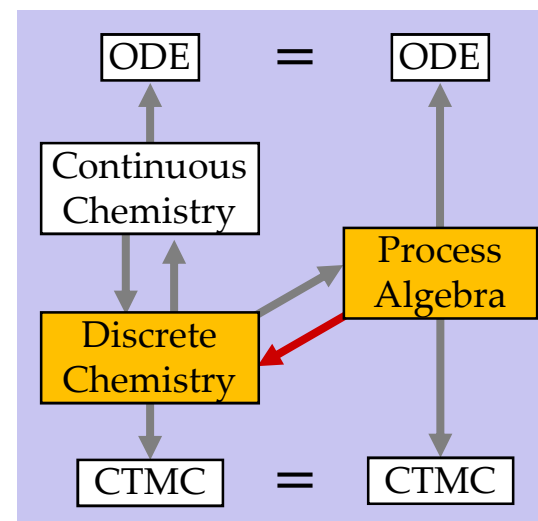
Automaton in state B

$A|A|B|B$

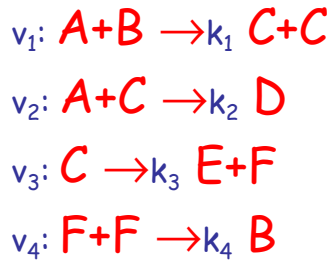
Initial conditions:
2A and 2B

From Reagents to Reactions (by example)

Interacting Automata	Discrete Chemistry
initial states $A \mid A \mid \dots \mid A$	initial quantities $\#A_0$
	$A \xrightarrow{r} A'$
	$A+B \xrightarrow{r} A'+B'$
	$A+A \xrightarrow{2r} A'+A''$



From Reactions to Reagents (by example)

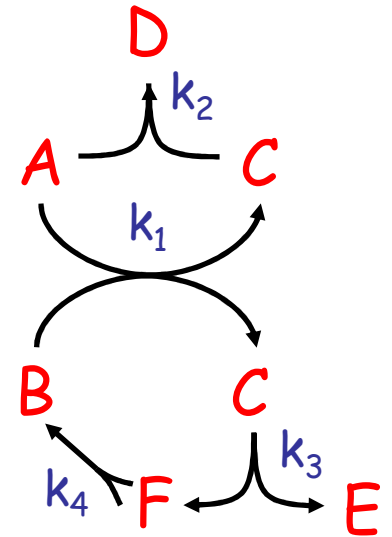


Interaction Matrix

channels and rates
(1 per reaction)

Half-rate for homeo reactions

	$v_1(k_1)$	$v_2(k_2)$	$v_3(k_3)$	$v_4(k_4/2)$
A	?:(C C)	?;D		
B	!;0			
C		!;0	$\tau:(E F)$	
D				
E				
F				?;B !;0



1: Fill the matrix by columns:

Degradation reaction $v_i: X \rightarrow_{k_i} P_i$
add $\tau;P_i$ to $\langle X, v_i \rangle$.

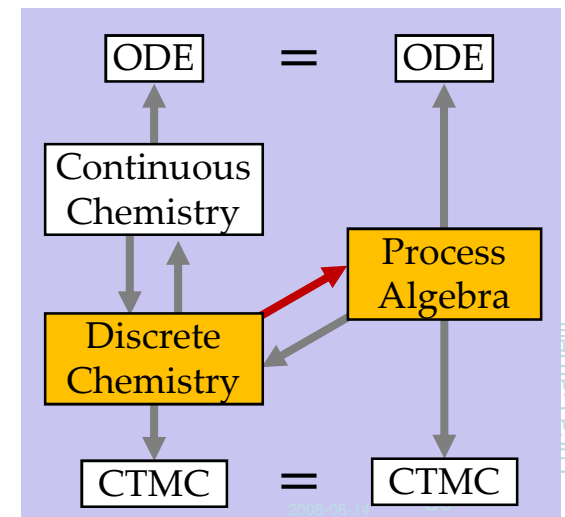
Hetero reaction $v_i: X+Y \rightarrow_{k_i} P_i$
add $?;P_i$ to $\langle X, v_i \rangle$ and $!;0$ to $\langle Y, v_i \rangle$

Homeo reaction $v_i: X+X \rightarrow_{k_i} P_i$
add $?;P_i$ and $!;0$ to $\langle X, v_i \rangle$

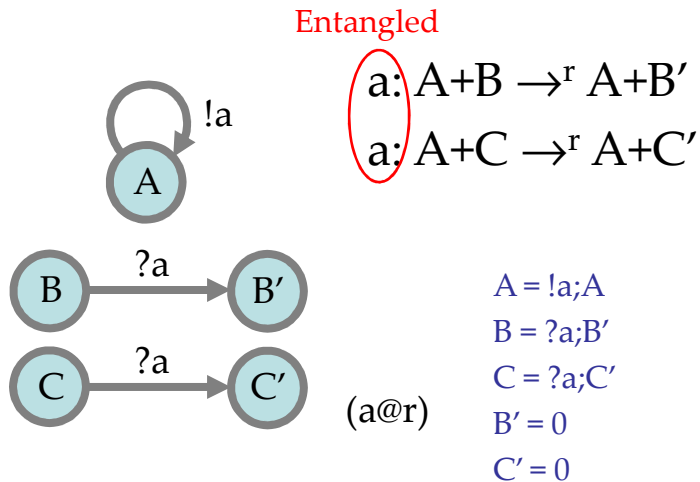
definitions
(1 per species)

2: Read the result by rows:

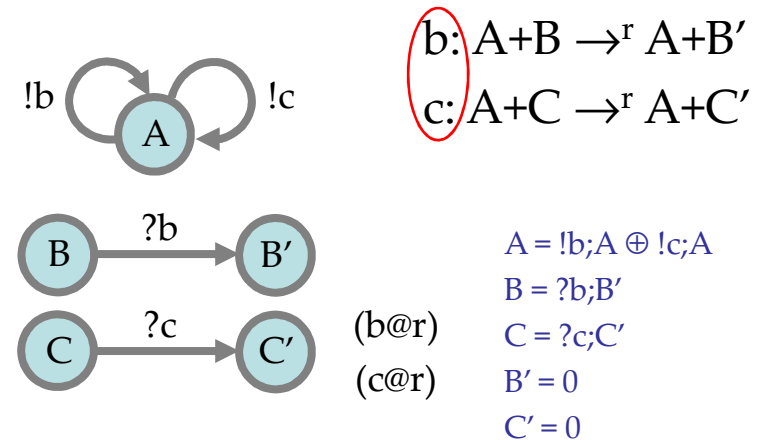
$$\begin{aligned}
 A &= ?v_{1(k_1)}:(C|C) \oplus ?v_{2(k_2)};D \\
 B &= !v_{1(k_1)};0 \\
 C &= !v_{2(k_2)};0 \oplus \tau_{k_3}:(E|F) \\
 D &= 0 \\
 E &= 0 \\
 F &= ?v_{4(k_4/2)};B \oplus !v_{4(k_4/2)};0
 \end{aligned}$$



Entangled vs Detangled



Entangled: Two reactions on one channel



Detangled: Two reactions on two separate channels

We need a semantics of automata that identifies automata that have the "same chemistry".

No process algebra equivalence is like this!

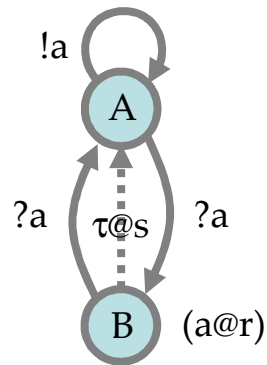
Entangled automata lead to more compact models than in chemistry.

Detangled automata are in simple correspondence with chemistry.

Same Semantics

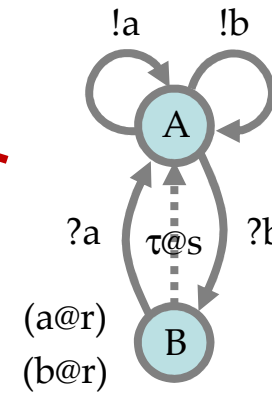
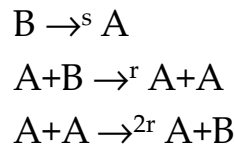
Could chemistry itself be that semantics?

No: different sets of reactions can have the same behavior!



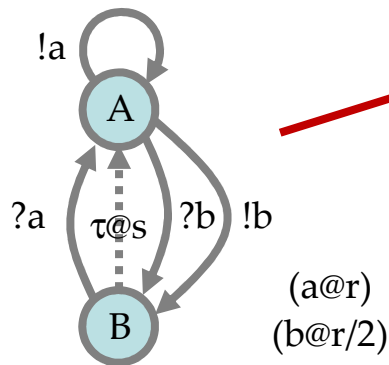
$$A = !a;A \oplus ?a;B$$

$$B = ?a;A \oplus \tau_{(s)};A$$



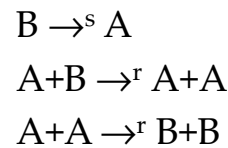
$$A = !a;A \oplus !b;A \oplus ?a;B$$

$$B = ?a;A \oplus \tau_{(s)};A$$



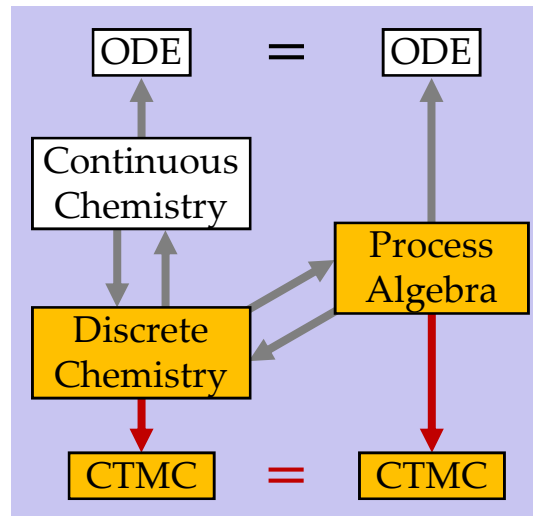
$$A = !a;A \oplus !b;B \oplus ?b;B$$

$$B = ?a;A \oplus \tau_{(s)};A$$



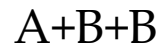
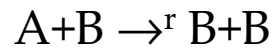
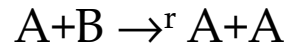
Different reactions,
but they induce the
same ODEs

Discrete-State Semantics

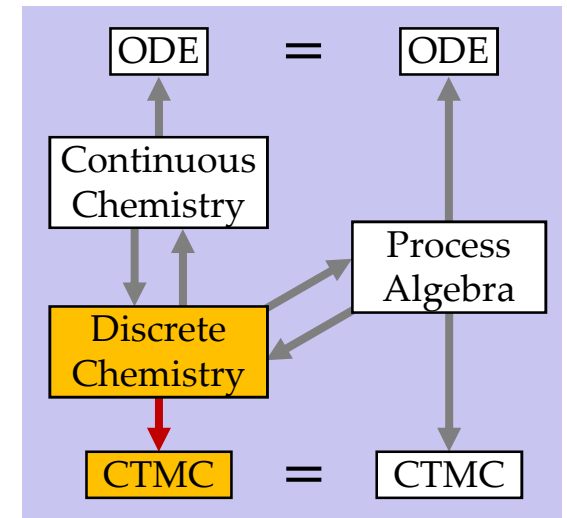
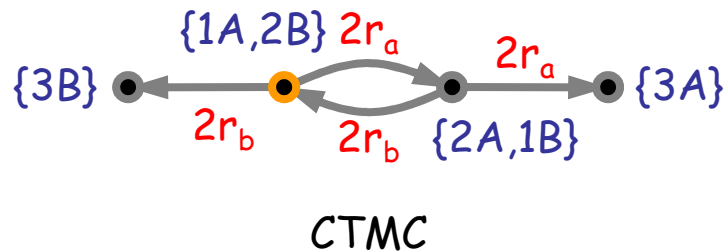


Discrete Semantics of Reactions

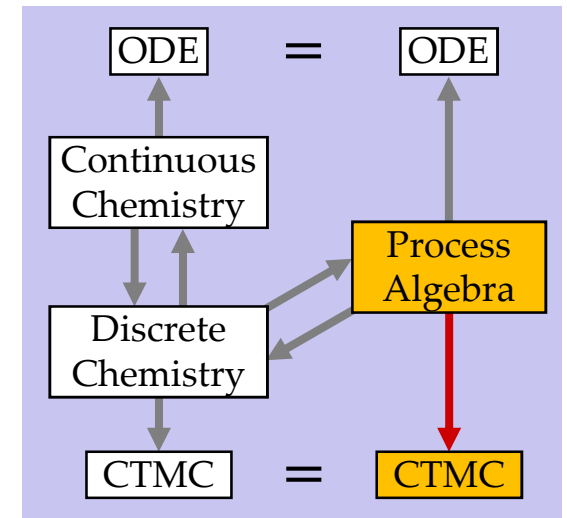
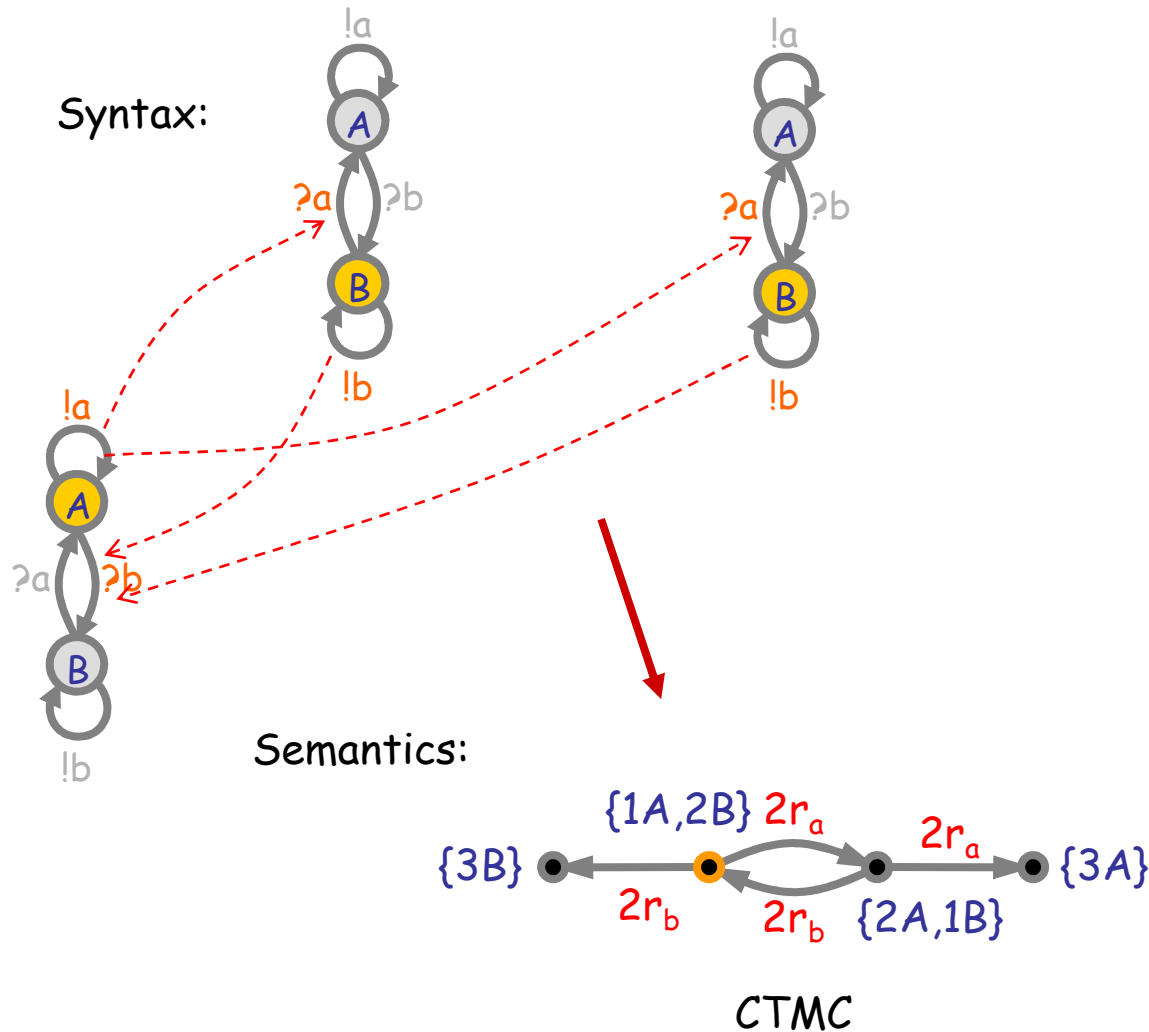
Syntax:



Semantics:



Discrete Semantics of Reagents

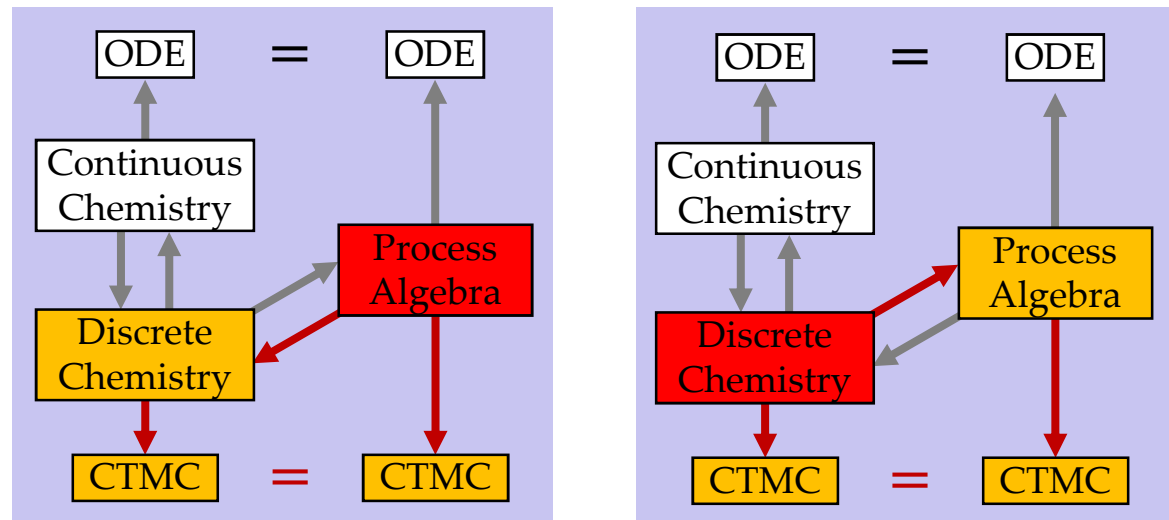


Discrete State Equivalence

- Def: \approx is equivalent CTMC's (isomorphic graphs with same rates).

- Thm: $E \approx \text{Ch}(E)$

- Thm: $C \approx \text{Pi}(C)$



- For each E there is an $E' \approx E$ that is detangled ($E' = \text{Pi}(\text{Ch}(E))$)

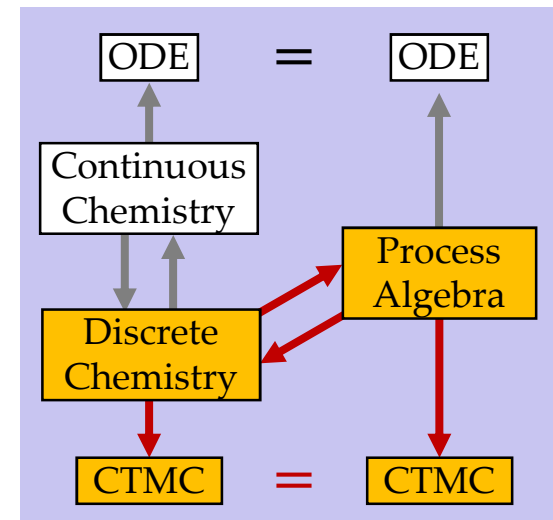
- For each E in automata form there is an $E' \approx E$ that is detangled and in automata form ($E' = \text{Detangle}(E)$).

Process Algebra = Discrete Chemistry

This is enough to establish that the process algebra is really faithful to the chemistry.

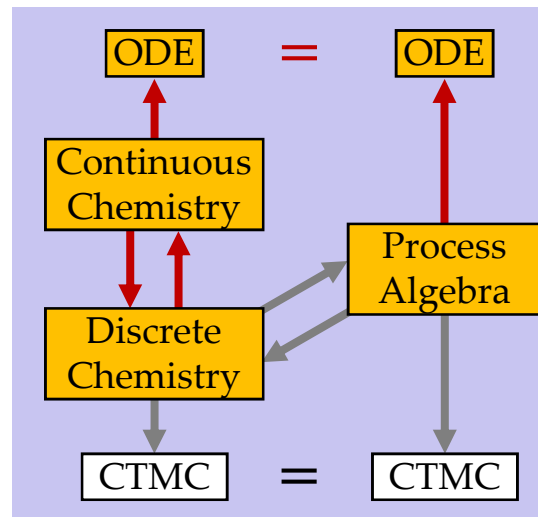
But CTMC are not the “ultimate semantics” because there are still questions of when two different CTMCs are actually equivalent (e.g. “lumping”).

The “ultimate semantics” of chemistry is the *Chemical Master Equation* (derivable from the Chapman-Kolmogorov equation of the CTMC).



Continuous-State Semantics

(short version)



The Gillespie(?) Conversion

Discrete Chemistry	Continuous Chemistry	$\gamma = N_A V$	$:M^{-1}$
initial quantities $\#A_0$	initial concentrations $[A]_0$	with $[A]_0 = \#A_0/\gamma$	
$A \xrightarrow{r} A'$	$A \xrightarrow{k} A'$	with $k = r$	$:s^{-1}$
$A+B \xrightarrow{r} A'+B'$	$A+B \xrightarrow{k} A'+B'$	with $k = r\gamma$	$:M^{-1}s^{-1}$
$A+A \xrightarrow{r} A'+A''$	$A+A \xrightarrow{k} A'+A''$	with $k = r\gamma/2$	$:M^{-1}s^{-1}$

V = interaction volume

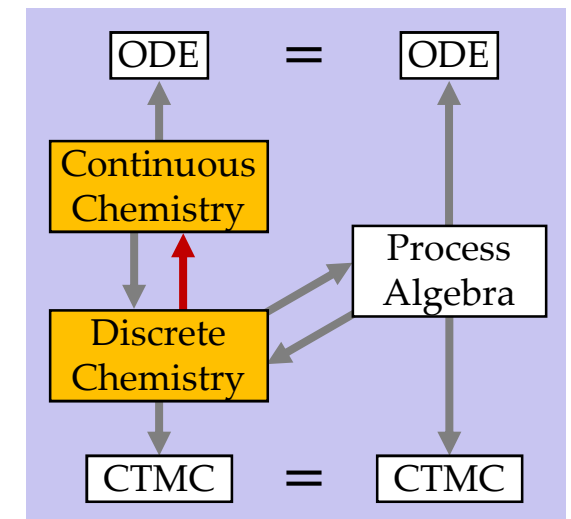
N_A = Avogadro's number

Think $\gamma = 1$

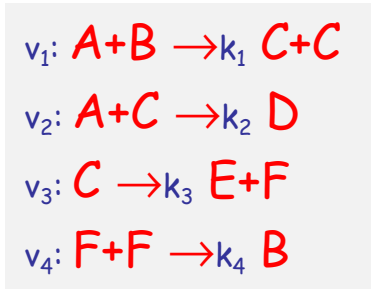
i.e. $V = 1/N_A$

$M = mol \cdot L^{-1}$

molarity (concentration)



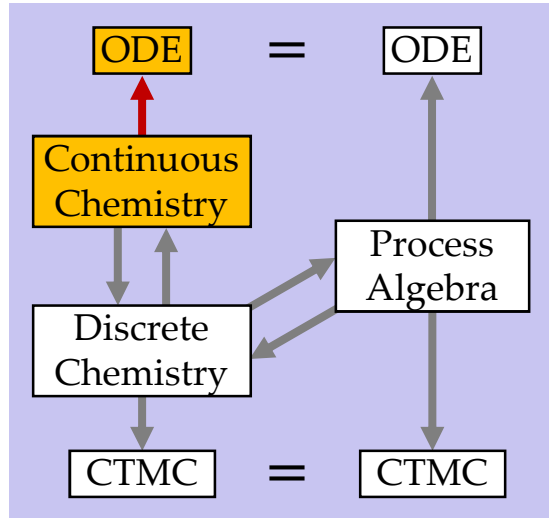
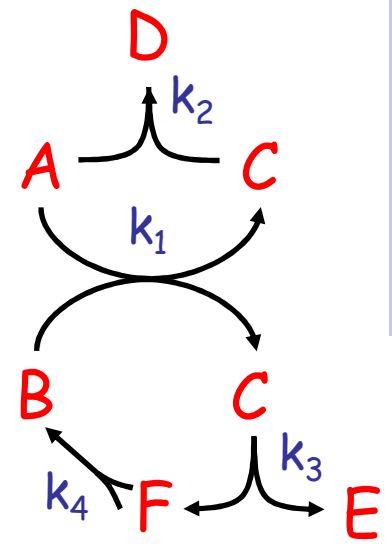
From Reactions to ODEs (Law of Mass Action)



Write the coefficients by columns

Stoichiometric Matrix

		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



Quantity changes

Stoichiometric matrix

Rate laws

$$d[X]/dt = N \cdot I$$

$d[A]/dt = -I_1 - I_2$
 $d[B]/dt = -I_1 + I_4$
 $d[C]/dt = 2I_1 - I_2 - I_3$
 $d[D]/dt = I_2$
 $d[E]/dt = I_3$
 $d[F]/dt = I_3 - 2I_4$

Read the concentration changes from the rows

E.g. $d[A]/dt = -k_1[A][B] - k_2[A][C]$

Set a rate law for each reaction (Degradation/Hetero/Homeo)

I	
I ₁	$k_1[A][B]$
I ₂	$k_2[A][C]$
I ₃	$k_3[C]$
I ₄	$k_4[F]^2$

X: chemical species
[-]: quantity of molecules
I: rate laws
k: kinetic parameters
N: stoichiometric matrix

Processes Rate Equation

Process Rate Equation for Reagents E in volume γ

$$d[X]/dt = (\sum(Y \in E) \text{Accr}_E(Y, X) \cdot [Y]) - \text{Depl}_E(X) \cdot [X]$$

for all $X \in E$

"The change in process concentration (!!) for X at time t is:
 the sum over all possible (kinds of) processes Y of:
 the concentration at time t of Y
 times the accretion from Y to X
 minus the concentration at time t of X
 times the depletion of X to some other Y"

$\text{Depl}_E(X) =$

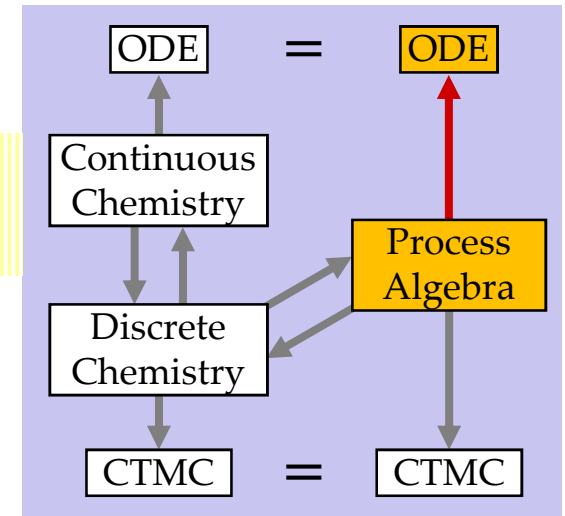
$$\begin{aligned} & \sum(i: E.X.i = \tau_{(r)}; P) r + \\ & \sum(i: E.X.i = ?a_{(r)}; P) r\gamma \cdot \text{OutsOn}_E(a) + \\ & \sum(i: E.X.i = !a_{(r)}; P) r\gamma \cdot \text{InsOn}_E(a) \end{aligned}$$

$\text{Accr}_E(Y, X) =$

$$\begin{aligned} & \sum(i: E.Y.i = \tau_{(r)}; P) \#X(P) \cdot r + \\ & \sum(i: E.Y.i = ?a_{(r)}; P) \#X(P) \cdot r\gamma \cdot \text{OutsOn}_E(a) + \\ & \sum(i: E.Y.i = !a_{(r)}; P) \#X(P) \cdot r\gamma \cdot \text{InsOn}_E(a) \end{aligned}$$

$\text{InsOn}_E(a) = \sum(Y \in E) \#\{Y.i \mid E.Y.i = ?a_{(r)}; P\} \cdot [Y]$

$\text{OutsOn}_E(a) = \sum(Y \in E) \#\{Y.i \mid E.Y.i = !a_{(r)}; P\} \cdot [Y]$



$$X = \tau_{(r)}; 0 \quad \rightarrow \quad d[X]/dt = -r[X]$$

$$\begin{aligned} X = ?a_{(r)}; 0 \\ Y = !a_{(r)}; 0 \end{aligned} \quad \rightarrow \quad \begin{aligned} d[X]/dt &= -r\gamma[X][Y] \\ d[Y]/dt &= -r\gamma[X][Y] \end{aligned}$$

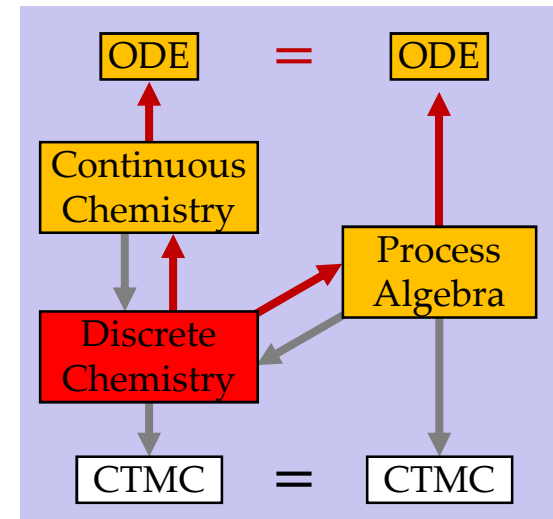
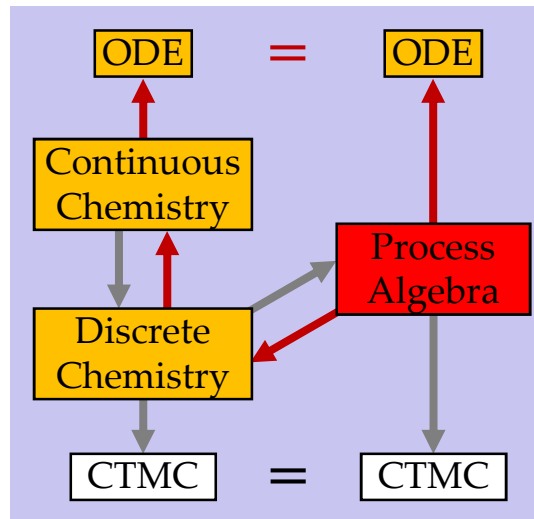
$$\begin{aligned} X = ?a_{(r)}; 0 \\ \oplus !a_{(r)}; 0 \end{aligned} \quad \rightarrow \quad d[X]/dt = -2r\gamma[X]^2$$

Continuous State Equivalence

- Def: \approx is equivalence of polynomials over the field of reals.

- Thm: $E \approx \text{Cont}(\text{Ch}(E))$

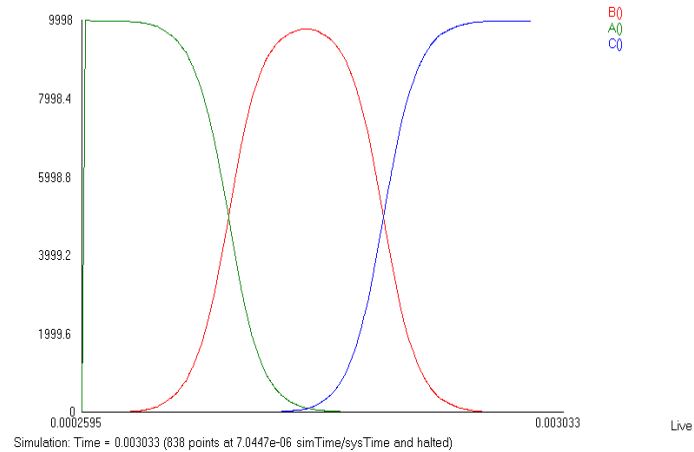
- Thm: $\text{Cont}(C) \approx \text{Pi}(C)$



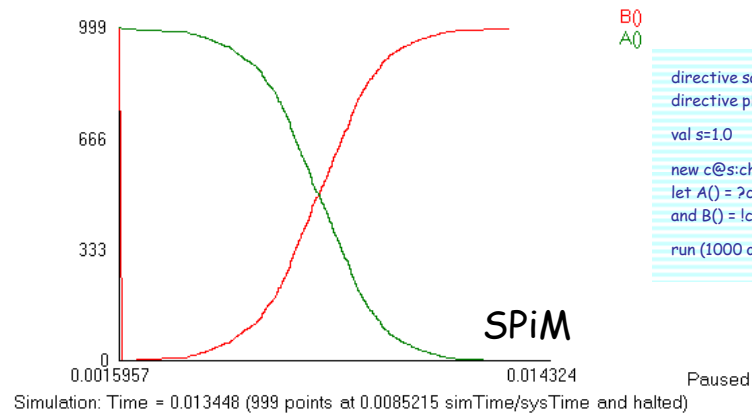
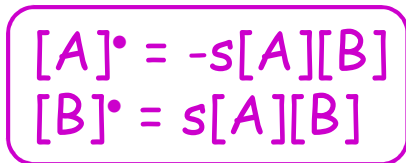
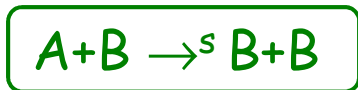
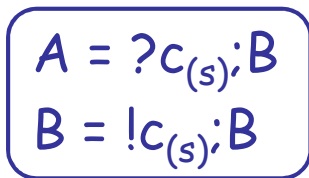
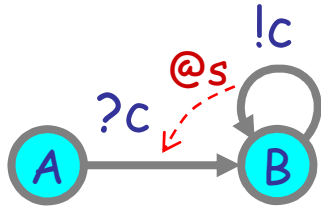
- For each E there is an $E' \approx E$ that is detangled ($E' = \text{Pi}(\text{Ch}(E))$)
- For each E in automata form there is an $E' \approx E$ that is detangled and in automata form ($E' = \text{Detangle}(E)$).

Exercise: Making Waves

Build me a population like this:



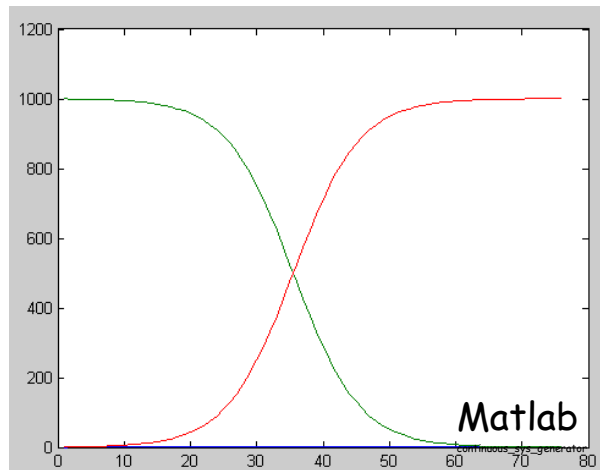
Nonlinear Transition (NLT)



```

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())
    
```

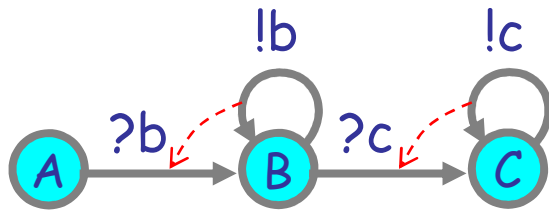
N.B.: needs at least 1 B to "get started".



```

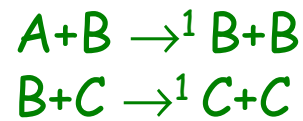
interval/step [0:0.001:0.0]
(A) dx1/dt = - x1*x2 1000.0
(B) dx2/dt = x1*x2 1.0
    
```

Two NLTs: Bell Shape



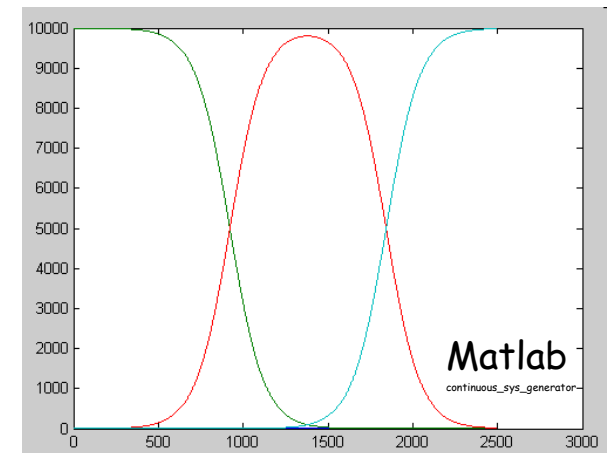
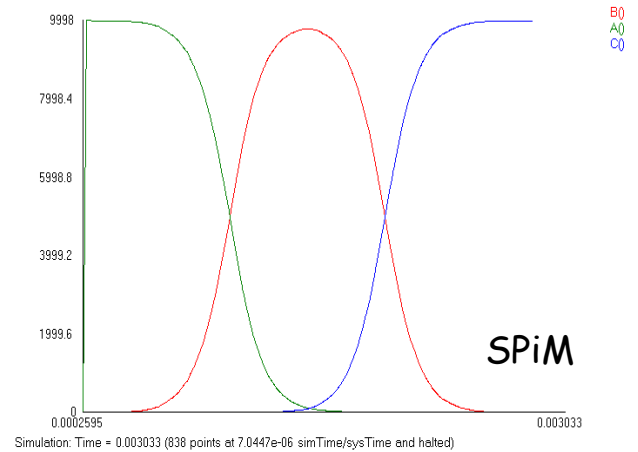
$$[B]^{\bullet} = [B]([A] - [C])$$

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



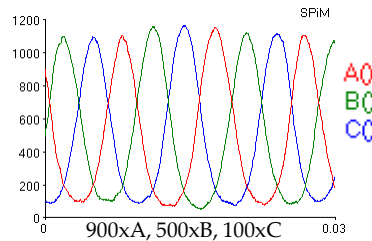
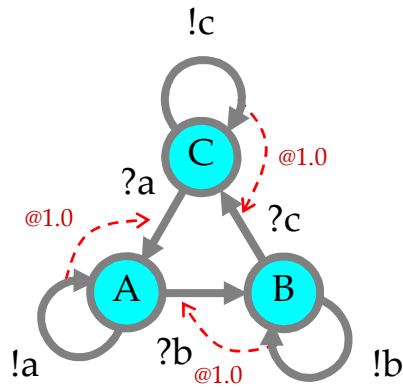
$$\begin{aligned} [A]^{\bullet} &= -[A][B] \\ [B]^{\bullet} &= [A][B] - [B][C] \\ [C]^{\bullet} &= [B][C] \end{aligned}$$

```
directive sample 0.0025 1000
directive plot B(); A(); C()
new b@1.0:chan new c@1.0:chan
let A() = ?b; B()
and B() = do !b;B() or ?c; C()
and C() = !c;C()
run ((10000 of A()) | B() | C())
```



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

NLT in a Cycle: Oscillator (unstable)



```
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()
```

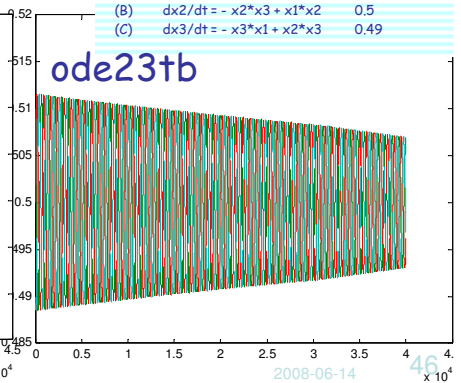
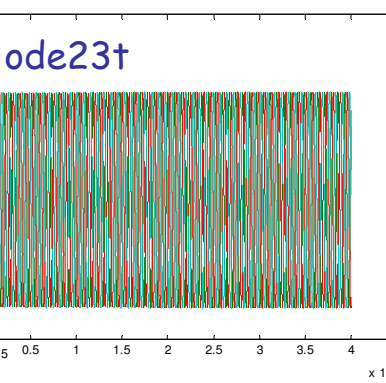
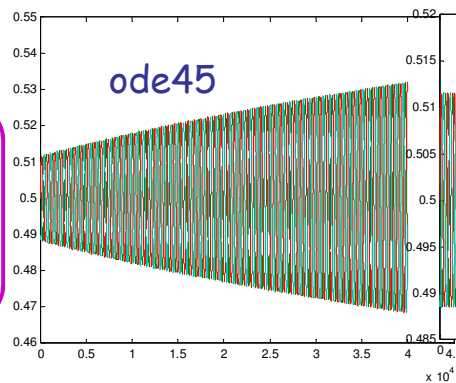
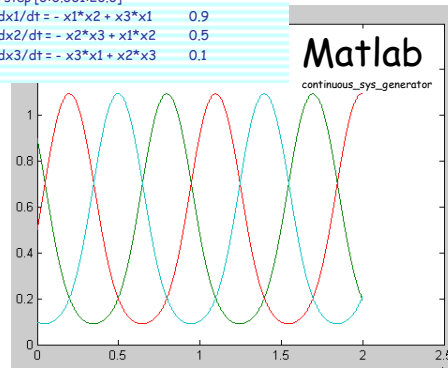
```
run (900 of A() | 500 of B() | 100 of C())
```

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

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

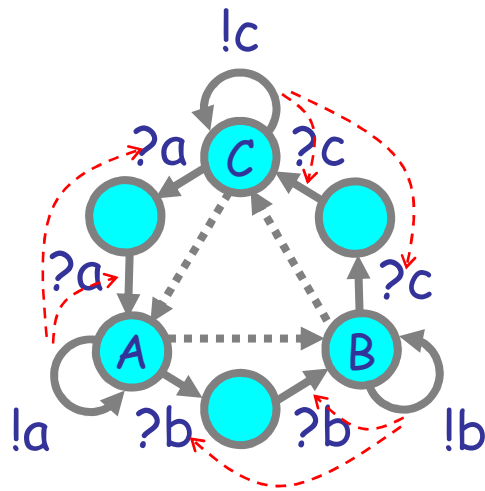
$[A]^{\bullet} = -s[A][B] + s[C][A]$
 $[B]^{\bullet} = -s[B][C] + s[A][B]$
 $[C]^{\bullet} = -s[C][A] + s[B][C]$

```
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
```

Oscillator (stable)



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

val r=1.0 val s=1.0

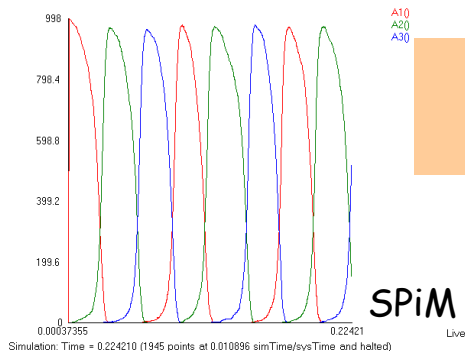
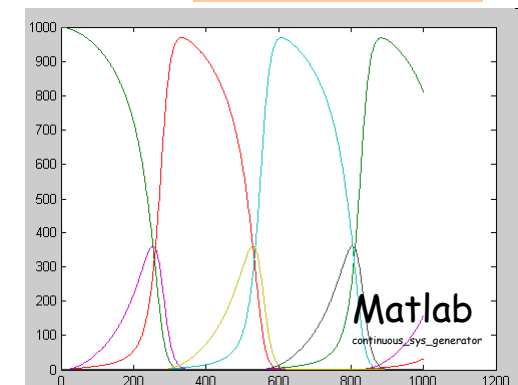
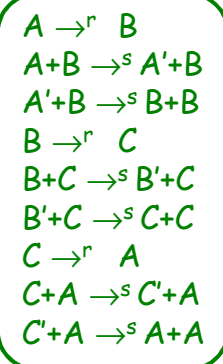
new a1@s:chan new a2@s:chan new a3@s:chan
let A1() = do !a1;A1() or delay@r;A2() or ?a2; ?a2; A2()
and A2() = do !a2;A2() or delay@r;A3() or ?a3; ?a3; A3()
and A3() = do !a3;A3() or delay@r;A1() or ?a1; ?a1; A1()

run 1000 of A1()
```

N.B. this does not deadlock!

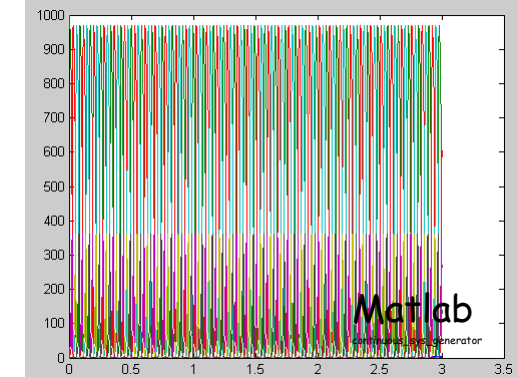
$$\begin{aligned}
 A &= !a_{(s)};A \oplus \tau_r;B \oplus ?b_{(s)};A' \\
 A' &= ?b_{(s)};B \\
 B &= !b_{(s)};B \oplus \tau_r;C \oplus ?c_{(s)};B' \\
 B' &= ?c_{(s)};C \\
 C &= !c_{(s)};C \oplus \tau_r;A \oplus ?a_{(s)};C' \\
 C' &= ?a_{(s)};A
 \end{aligned}$$

Sustained Deterministic Oscillation



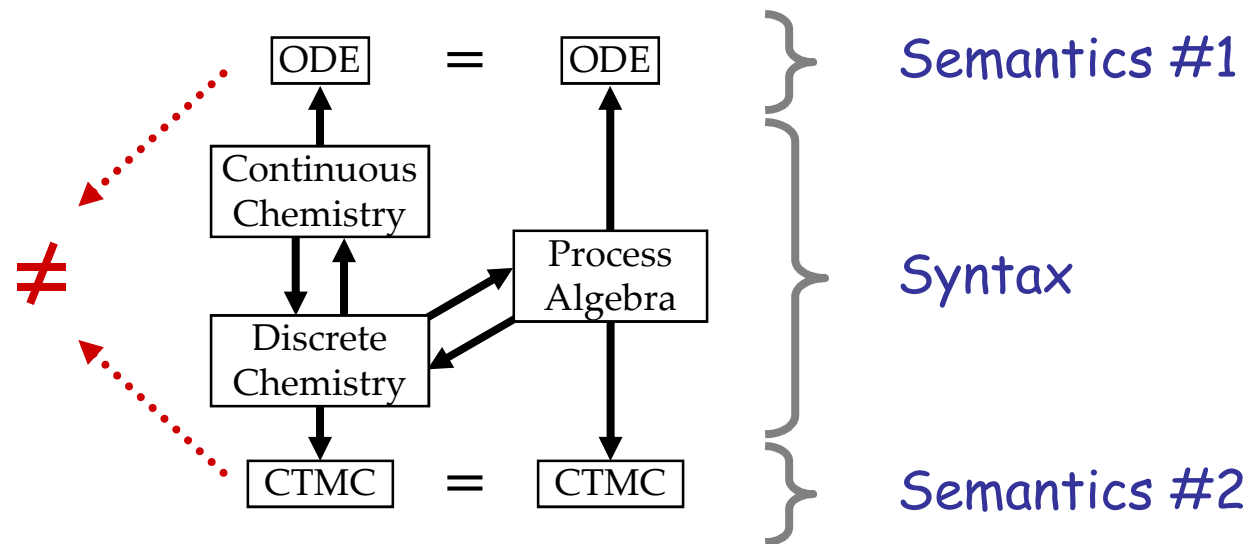
Robust Stochastic Oscillation

$$\begin{aligned}
 [A]^* &= -r[A] - s[A][B] + r[C] + s[C'][A] \\
 [B]^* &= -r[B] - s[B][C] + r[A] + s[A'][B] \\
 [C]^* &= -r[C] - s[C][A] + r[B] + s[B'][C] \\
 [A']^* &= -s[A'][B] + s[A][B] \\
 [B']^* &= -s[B'][C] + s[B][C] \\
 [C']^* &= -s[C][A] + s[C][A]
 \end{aligned}$$



```
interval/step [0:0.0001:0.1]
(A) dx1/dt = -x1 - x1^2*x2 + x3 + x6^4*x1 1000.0
(B) dx2/dt = -x2 - x2^4*x3 + x1 + x6^4*x2 0.0
(C) dx3/dt = -x3 - x3^4*x1 + x2 + x5^4*x3 0.0
(A') dx4/dt = -x4^4*x2 + x7^4*x2 0.0
(B') dx5/dt = -x5^4*x3 + x2^4*x3 0.0
(C') dx6/dt = -x6^4*x1 + x3^4*x1 0.0
```

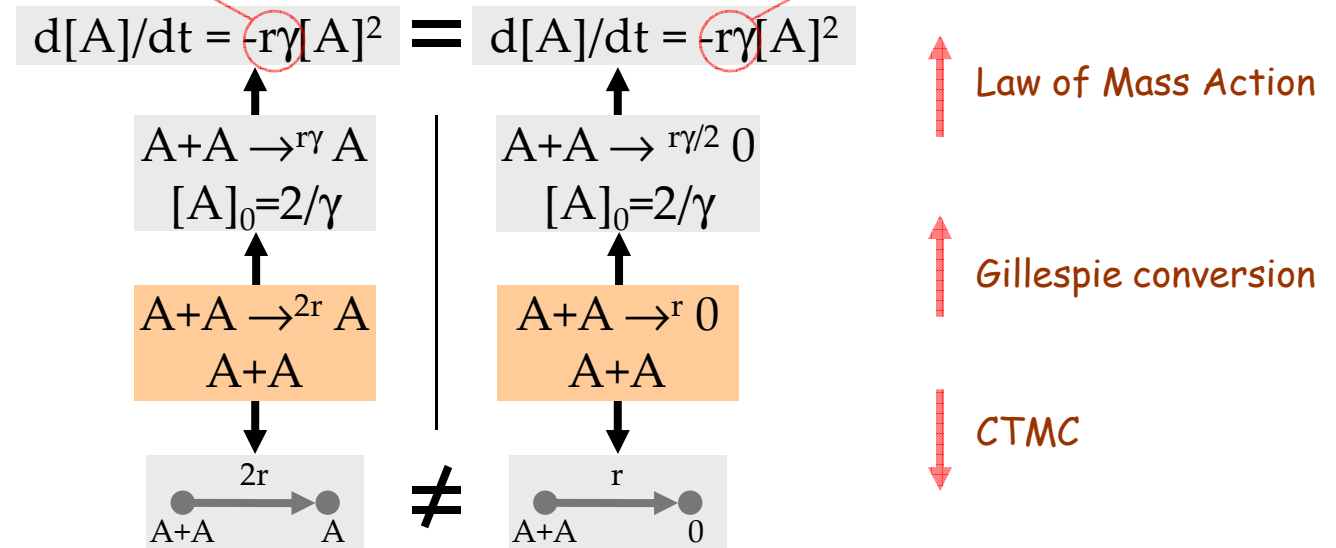
GMA \neq CME



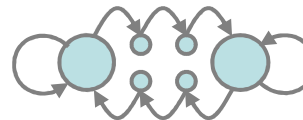
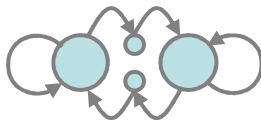
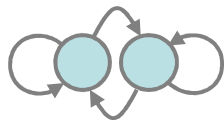


1*reaction rate $r\gamma$ because
1*A is lost in reaction.

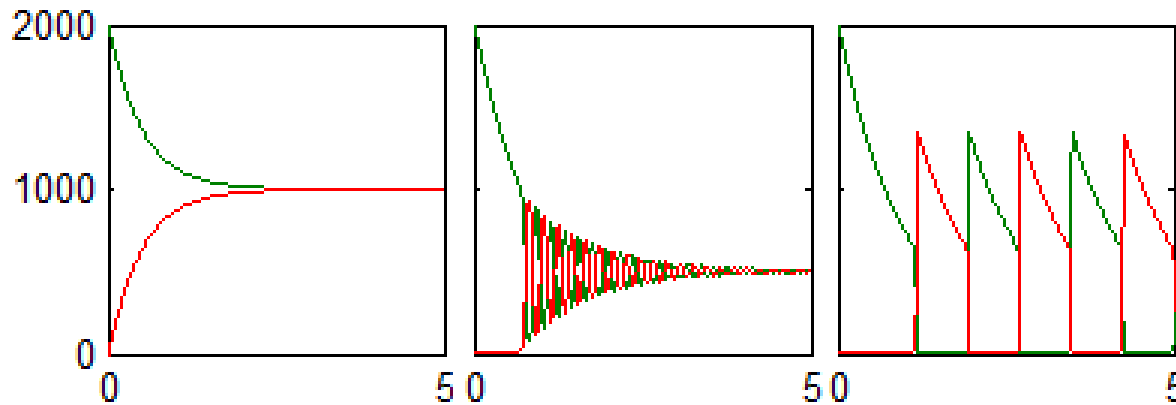
2*reaction rate $r\gamma/2$ because
2*A are lost in reaction.



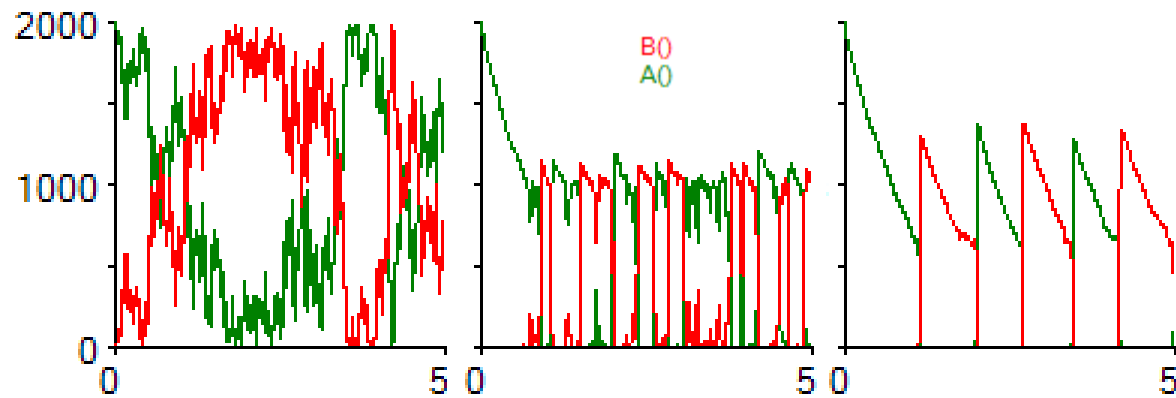
Continuous vs. Discrete Groupies



(all with doping)



Matlab



SPiM

$2000 \times A, 0 \times B, 1 \times A_d, 1 \times B_d, r = 1.0$

```
directive sample 5.0 1000
directive plot B0;A0
new a0(L:chan)
new b0(L:chan)
let A0 = do Ia; A0 or ?b; B0
and B0 = do Ib; B0 or ?a; A0
let Ad0 = Ia; Ad0
and Bd0 = Ib; Bd0
run 2000 of A0
run 1 of (Ad0 | Bd0)
```

```
directive sample 5.0 1000
directive plot B0;A0
new a0(L:chan)
new b0(L:chan)
let A0 = do Ia; A0 or ?b; B0
and B0 = do Ib; B0 or ?a; ?c; A0
let Ad0 = Ia; Ad0
and Bd0 = Ib; Bd0
run 2000 of A0
run 1 of (Ad0 | Bd0)
```

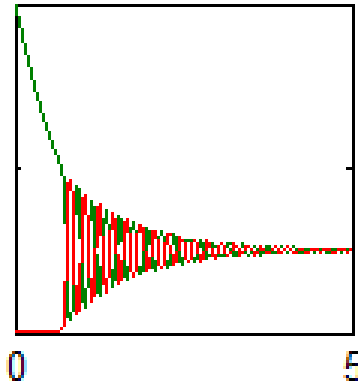
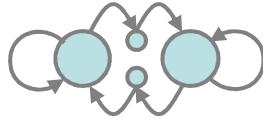
```
directive sample 5.0 1000
directive plot B0;A0
new a0(L:chan)
new b0(L:chan)
let A0 = do Ia; A0 or ?b; ?b; B0
and B0 = do Ib; B0 or Ia; ?a; ?a; A0
let Ad0 = Ia; Ad0
and Bd0 = Ib; Bd0
run 2000 of A0
run 1 of (Ad0 | Bd0)
```

```
Groupes ODEs - Groupies.mat
[0;0;0;5;0] r=1.0 k=1.0
A dx1/dt=x1^4-x3^2*x1-x1*x4, 2000.0
A' dx2/dt=x3^2*x1-x3^2*x1-x2, 0.0
B dx3/dt=x3^2*x2-x1^2*x3-x3*x2, 0.0
B' dx4/dt=x1^2*x3-x1^2*x3-x4, 0.0
B' dx4/dt=x1^2*x3-x1^2*x3-x4, 0.0
```

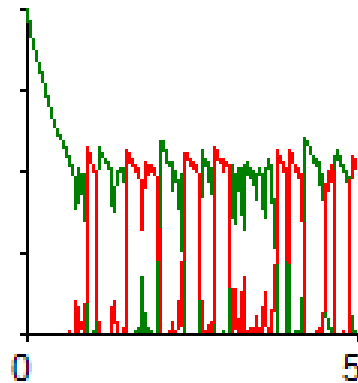
```
Groupes ODEs - Groupies Hysteric 1.mat
[0;0;0;5;0] r=1.0 k=1.0
A dx1/dt=x1^4-x3^2*x1-x1*x4, 2000.0
A' dx2/dt=x3^2*x1-x3^2*x1-x2, 0.0
B dx3/dt=x3^2*x2-x1^2*x3-x3*x2, 0.0
B' dx4/dt=x1^2*x3-x1^2*x3-x4, 0.0
B' dx4/dt=x1^2*x3-x1^2*x3-x4, 0.0
```

```
Groupes ODEs - Groupies Hysteric 2.mat
[0;0;0;5;0] r=1.0 k=1.0
A dx1/dt=x1^4-x3^2*x1-x1*x4, 2000.0
A' dx2/dt=x3^2*x1-x3^2*x1-x2, 0.0
A' dx5/dt=x3^2*x2-x3^2*x2-x5, 0.0
B dx3/dt=x3^2*x2-x1^2*x3-x3*x2, 0.0
B' dx4/dt=x1^2*x3-x1^2*x3-x4, 0.0
B' dx4/dt=x1^2*x3-x1^2*x3-x4, 0.0
```

Scientific Predictions

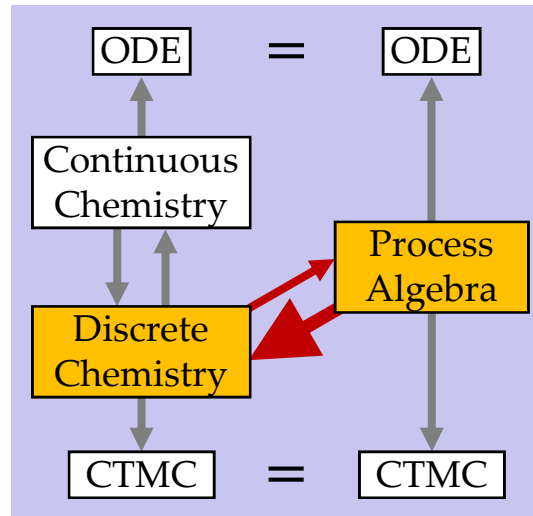


After a while, all 4 states are almost equally occupied.

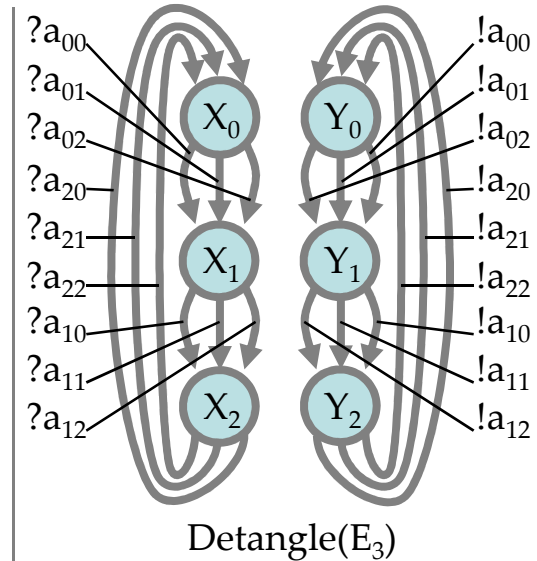
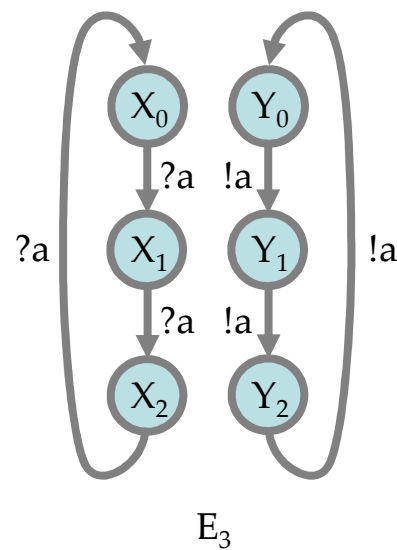


The 4 states are almost never equally occupied.

Model Compactness



Entangled vs detangled



(closely related to $\text{Pi}(\text{Ch}(E_3))$)

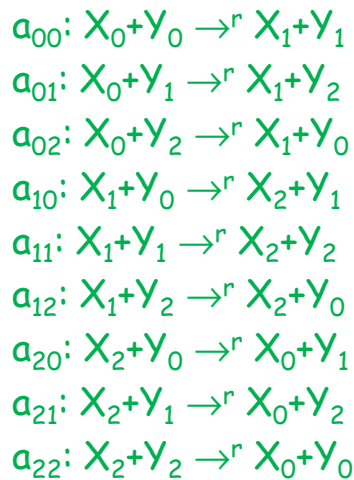
n^2 Scaling Problems

- E_n has $2n$ variables (nodes) and $2n$ terms (arcs).
- $Ch(E_n)$ has $2n$ species and n^2 reactions.
- The stoichiometric matrix has size $2n \cdot n^2 = 2n^3$.
- The ODEs have $2n$ variables and $2n(n+n) = 4n^2$ terms
(number of variables times number of accretions plus depletions when sums are distributed)

E_3

$$\begin{aligned} X_0 &= ?a_{(r)}:X_1 \\ X_1 &= ?a_{(r)}:X_2 \\ X_2 &= ?a_{(r)}:X_0 \\ Y_0 &= !a_{(r)}:Y_1 \\ Y_1 &= !a_{(r)}:Y_2 \\ Y_2 &= !a_{(r)}:Y_0 \end{aligned}$$

$Ch(E_3)$

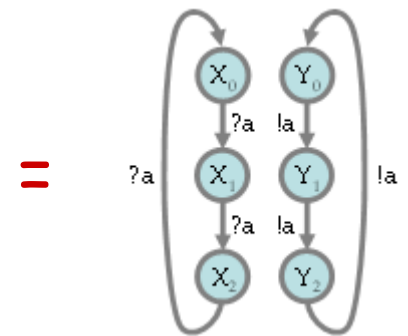


StoichiometricMatrix($Ch(E_3)$)

	a_{00}	a_{01}	a_{02}	a_{10}	a_{11}	a_{12}	a_{20}	a_{21}	a_{22}
X_0	-1	-1	-1				+1	+1	+1
X_1	+1	+1	+1	-1	-1	-1			
X_2				+1	+1	+1	-1	-1	-1
Y_0	-1		+1	-1		+1	-1		+1
Y_1	+1	-1		+1	-1		+1	-1	
Y_2		+1	-1		+1	-1		+1	-1

ODE(E_3)

$$\begin{aligned} d[X_0]/dt &= -r[X_0][Y_0] - r[X_0][Y_1] - r[X_0][Y_2] + r[X_2][Y_0] + r[X_2][Y_1] + r[X_2][Y_2] \\ d[X_1]/dt &= -r[X_1][Y_0] - r[X_1][Y_1] - r[X_1][Y_2] + r[X_0][Y_0] + r[X_0][Y_1] + r[X_0][Y_2] \\ d[X_2]/dt &= -r[X_2][Y_0] - r[X_2][Y_1] - r[X_2][Y_2] + r[X_1][Y_0] + r[X_1][Y_1] + r[X_1][Y_2] \\ d[Y_0]/dt &= -r[X_0][Y_0] - r[X_1][Y_0] - r[X_2][Y_0] + r[X_0][Y_2] + r[X_1][Y_2] + r[X_2][Y_2] \\ d[Y_1]/dt &= -r[X_0][Y_1] - r[X_1][Y_1] - r[X_2][Y_1] + r[X_0][Y_0] + r[X_1][Y_0] + r[X_2][Y_0] \\ d[Y_2]/dt &= -r[X_0][Y_2] - r[X_1][Y_2] - r[X_2][Y_2] + r[X_0][Y_1] + r[X_1][Y_1] + r[X_2][Y_1] \end{aligned}$$



On the Computational Power of Biochemistry

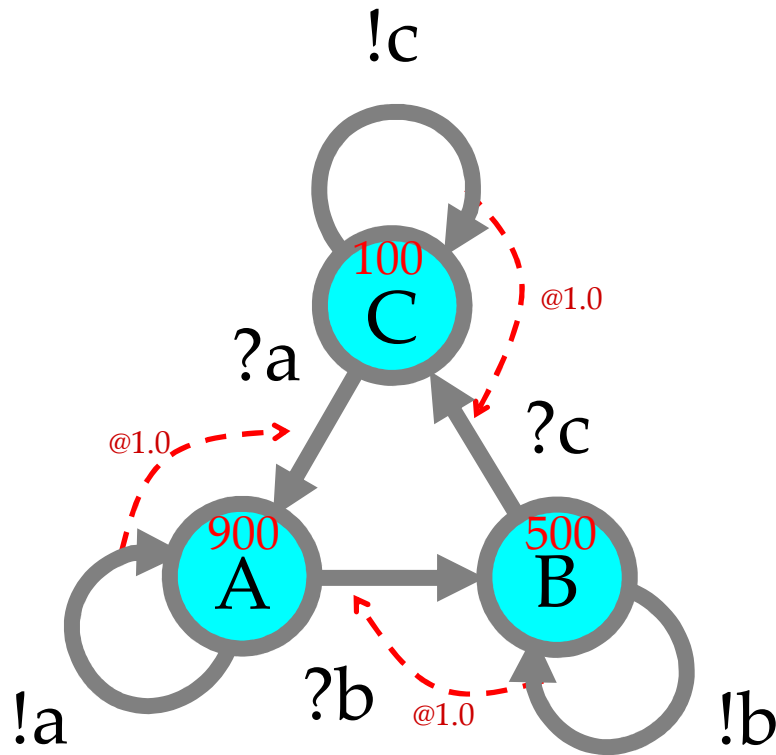
joint work with

Gianluigi Zavattaro

University of Bologna

in: Algebraic Biology '08

Can this program halt?



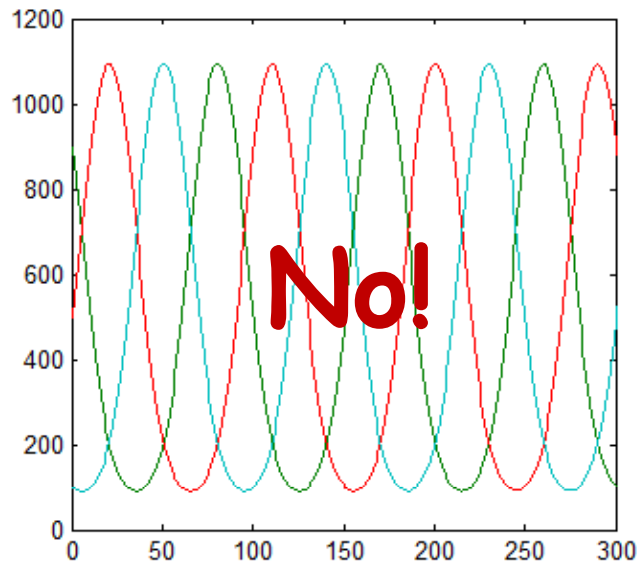
b: $A+B \rightarrow B+B$

c: $B+C \rightarrow C+C$

a: $C+A \rightarrow A+A$

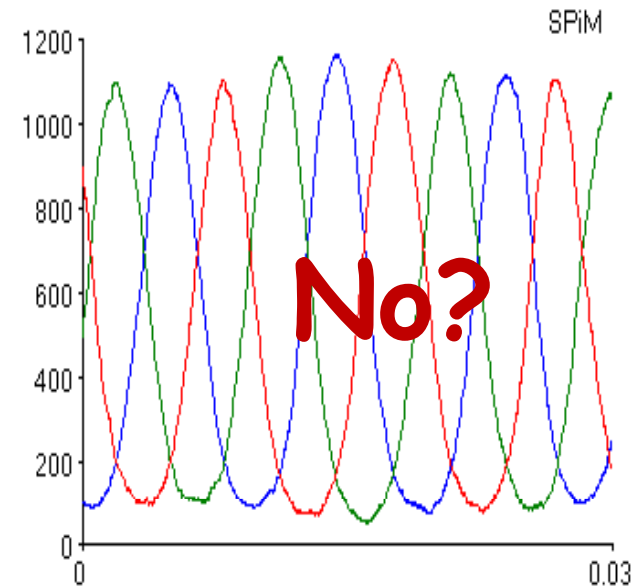
$900A + 500B + 100C$

"Experimental evidence"



Continuous-State Simulation

```
interval/step [0:0.0001:0.03]
(A) dx1/dt = - x1*x2 + x3*x1  900.0
(B) dx2/dt = - x2*x3 + x1*x2  500.0
(C) dx3/dt = - x3*x1 + x2*x3  100.0
```



Discrete-State Simulation

```
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())
```

But in a longer simulation...



Discrete-State
Simulation

$0A + 1500B + 0C$

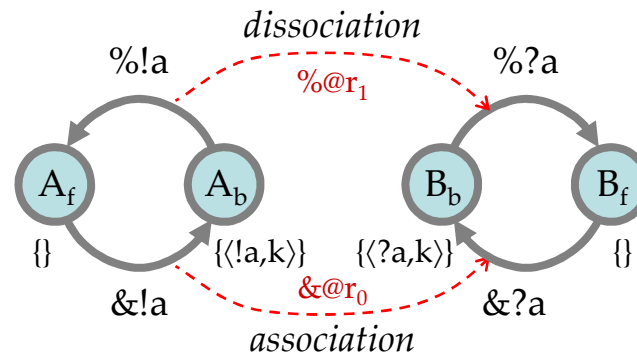
Is termination decidable in Chemistry?

- Three notations for “basic chemistry”:
 - FSRN: Finite Stochastic Reaction Networks
(*finite systems of stochastic chemical reactions*)
 - CGF: our process algebra (CTCM-equivalent to FSRN).
 - Place-Transition Petri nets.
- Answer: termination (reachability) in Chemistry is *decidable!*
 - FSRNs are not Turing-powerful (Soloveichik et al. *Computation with Finite Stochastic Chemical Reaction Networks*. In Nat. Computing. 2008).
 - Termination in CGF (whether a given molecule could be produced) can be reduced to termination in place-transition Petri Nets, where it (reachability) is decidable.
- Hence, basic chemistry is **NOT Turing-complete!**
 - (Although the full story is a bit more subtle.)

Biochemistry = Interaction + Complexation

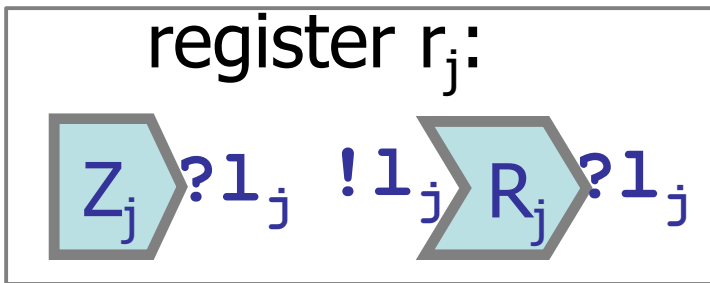
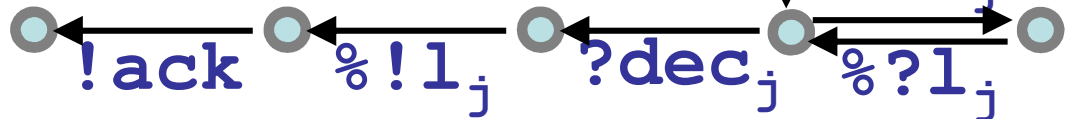
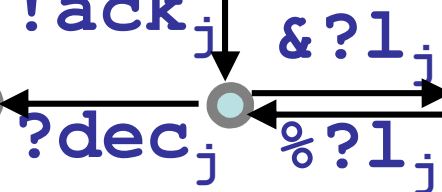
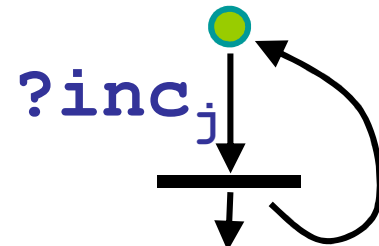
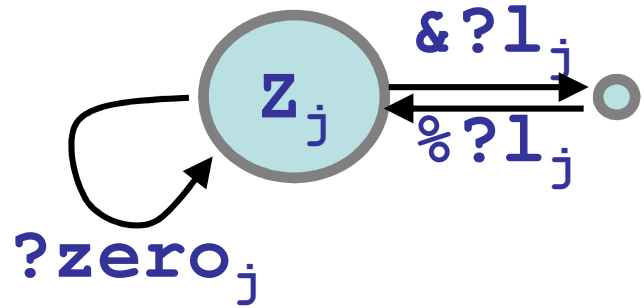
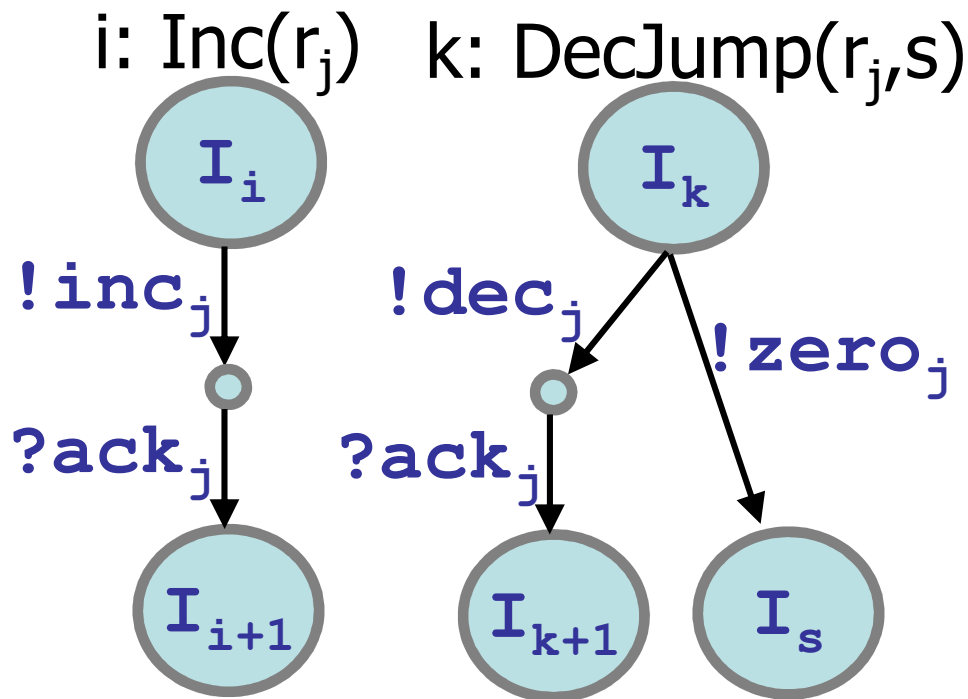


- Complexation is what proteins “do”, in contrast to simpler chemicals.



- Leading to a process algebra that we call the **Biochemical Ground Form (BGF)**.

RAM encoding in BGF



Expressiveness of Biochemistry

- Basic chemistry (FSRN, or CGF) **is not** Turing-complete
- Biochemistry (FSRN + complexation, or BGF) **is** Turing-complete.
- More powerful process algebras of course *are* Turing complete
 - They (e.g. π -calculus) include BGF, but they also have mechanisms that are not directly biologically justifiable.
 - In BGF we have in a sense the minimal biologically-inspired extension of FSRN, and it is already Turing-complete.
- Intrinsic to biochemistry (but not to simple chemistry) is at least one Turing-complete mechanism.

Conclusions

Conclusions

- **Compositional modeling languages**
 - 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 lines of extensions**
 - Parametric processes for model factorization
 - Ultimately, rich process-algebra based modeling languages.
- **Quantitative techniques**
 - Important in the "real sciences".