Molecules as Automata

Representing Biochemical Systems as Collectives of Interacting Automata

Luca Cardelli

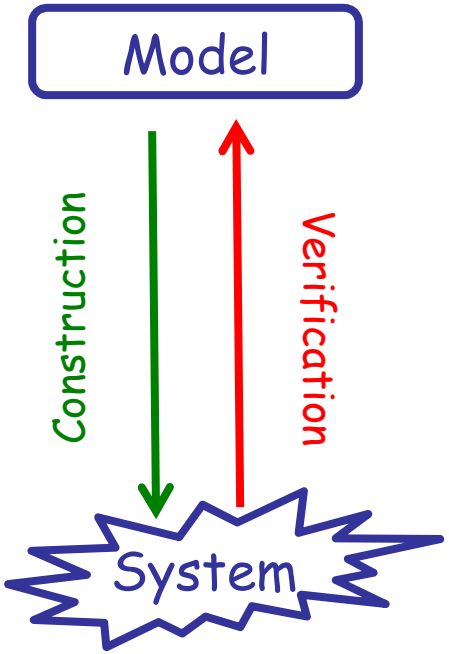
Microsoft Research

Sheffield, 2008-10-24

http://LucaCardelli.name

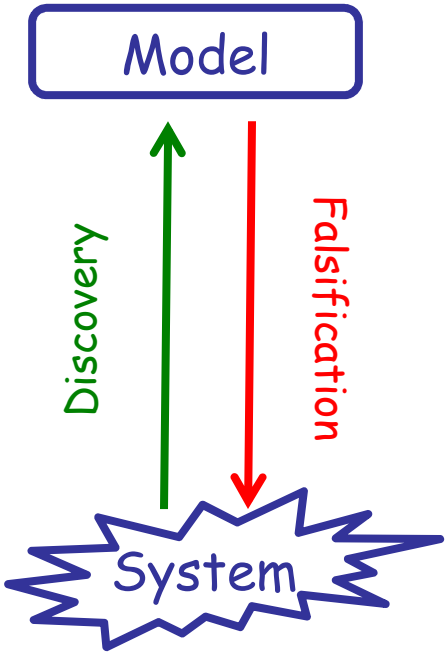
Scientific Method vs. Engineering Method

Engineering Method



Direct Engineering
(Synthetic Biology)

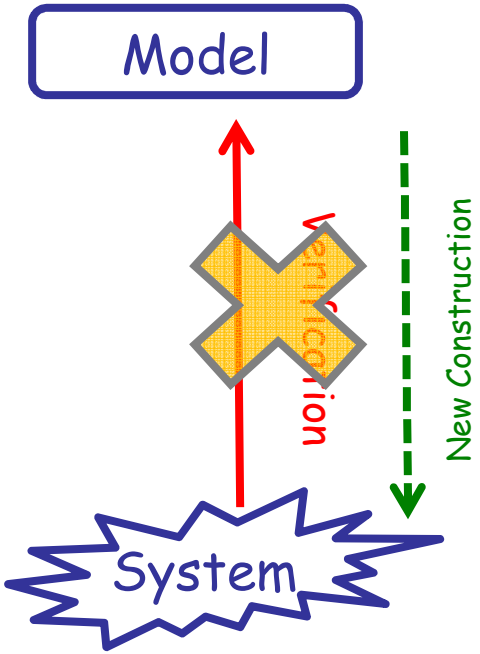
Scientific Method



Reverse Engineering
(Systems Biology)

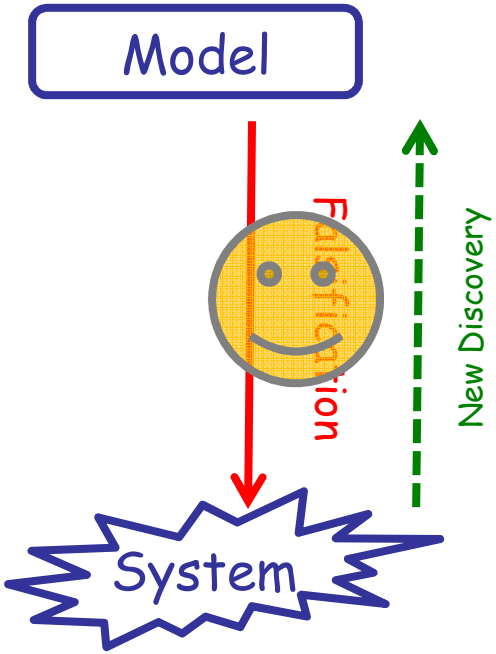
Scientific Method vs. Engineering Method

Engineering Method



Direct Engineering

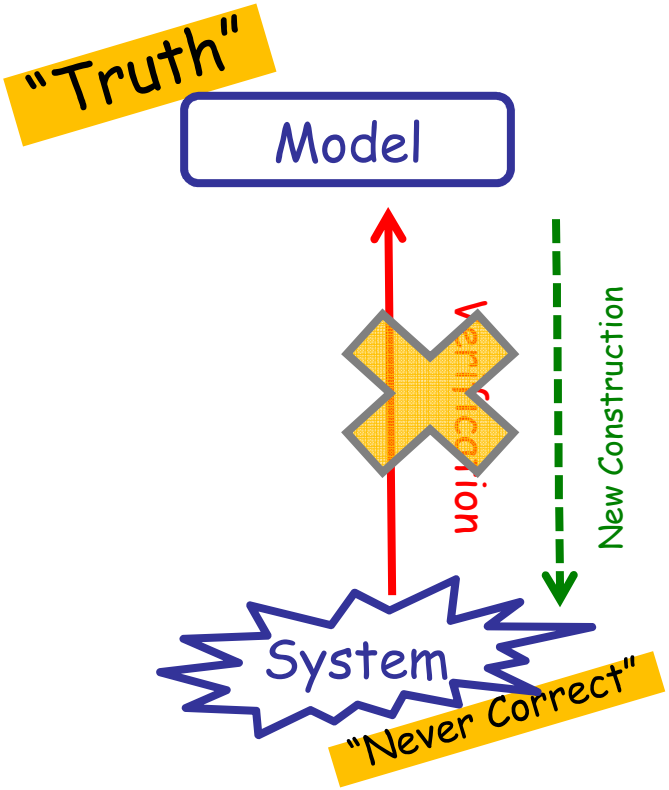
Scientific Method



Reverse Engineering

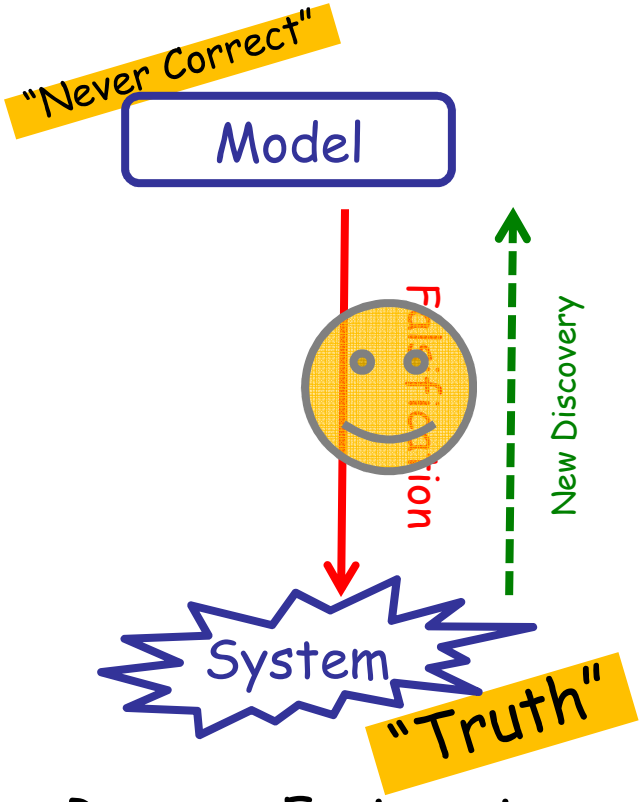
Scientific Method vs. Engineering Method

Engineering Method



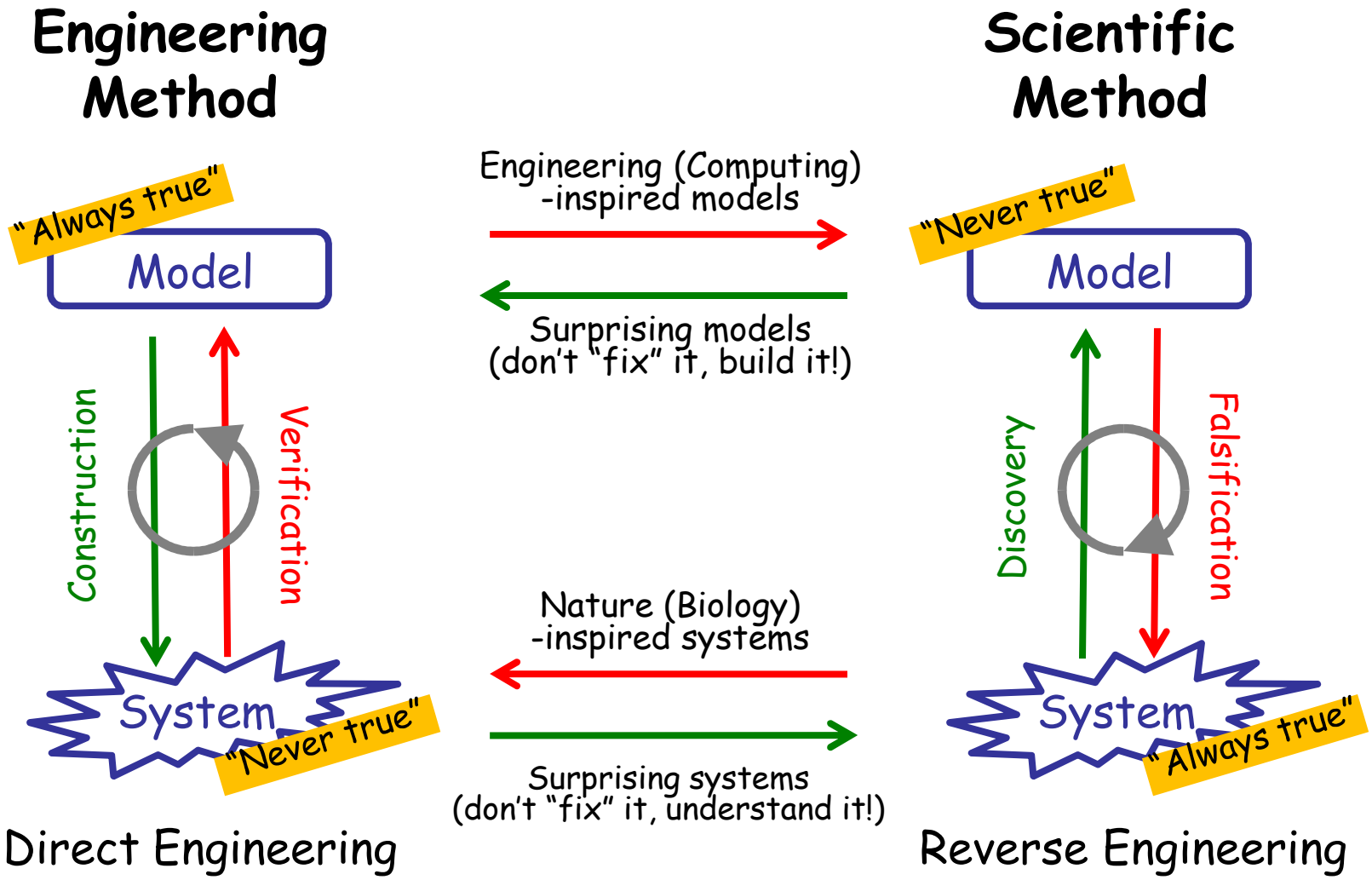
Direct Engineering

Scientific Method



Reverse Engineering

Scientific Method vs. Engineering Method

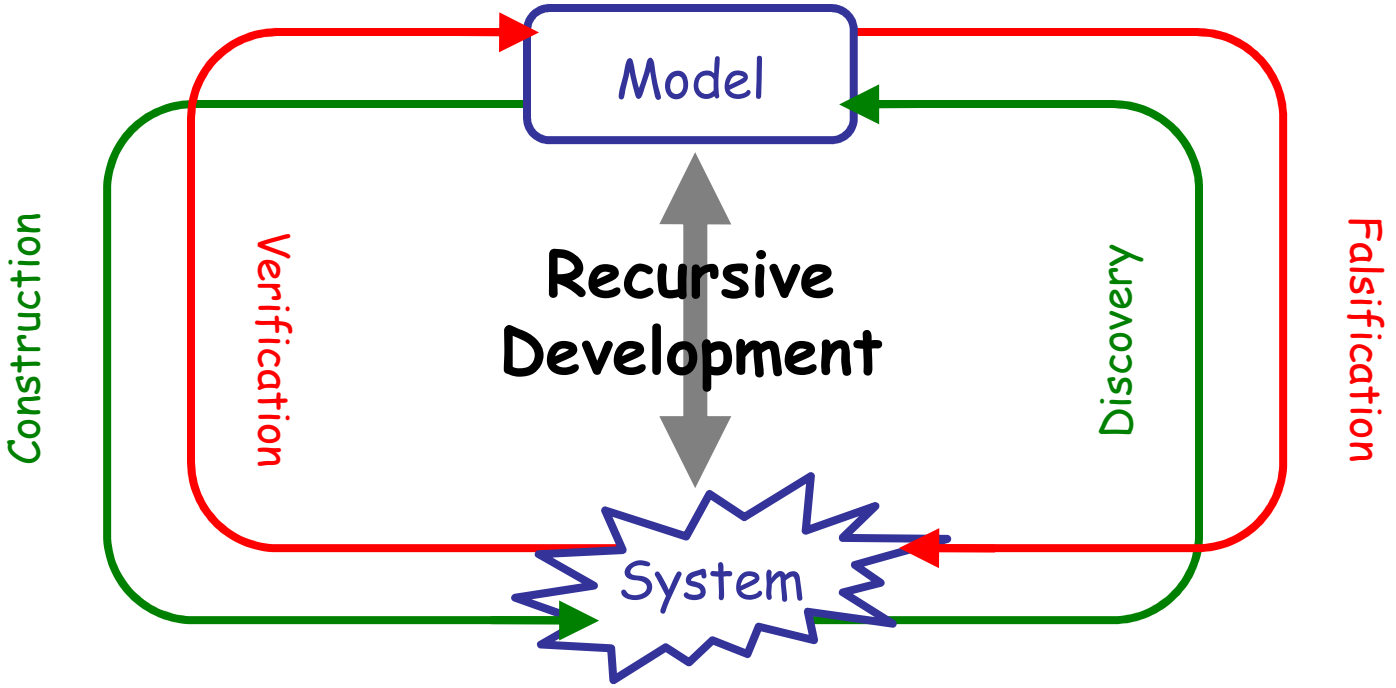


Scientific Method vs. Engineering Method

When the models and the systems are *both* too complex to *either* be the full Truth

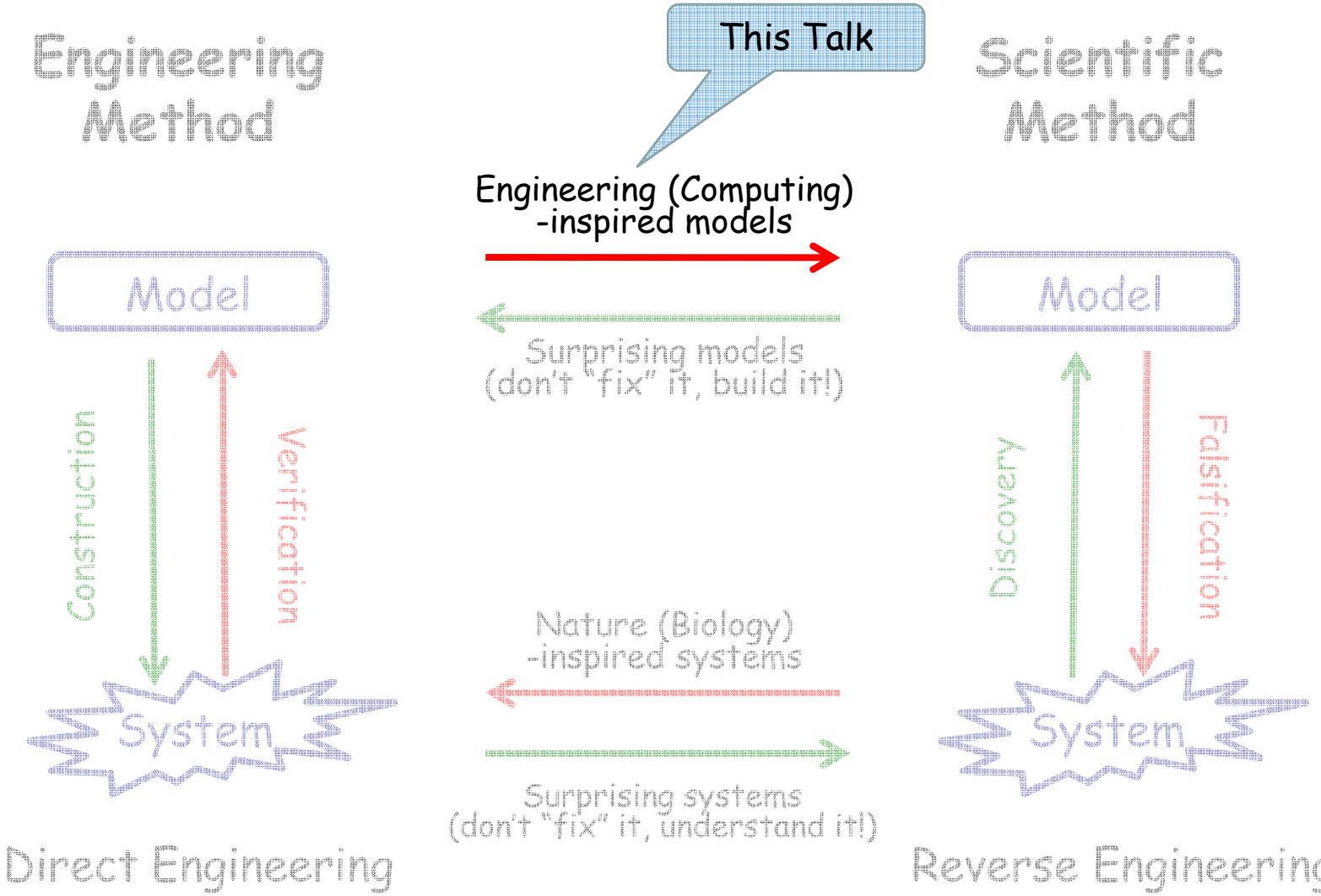
Combined Method

The models that we discover should be suitable for construction



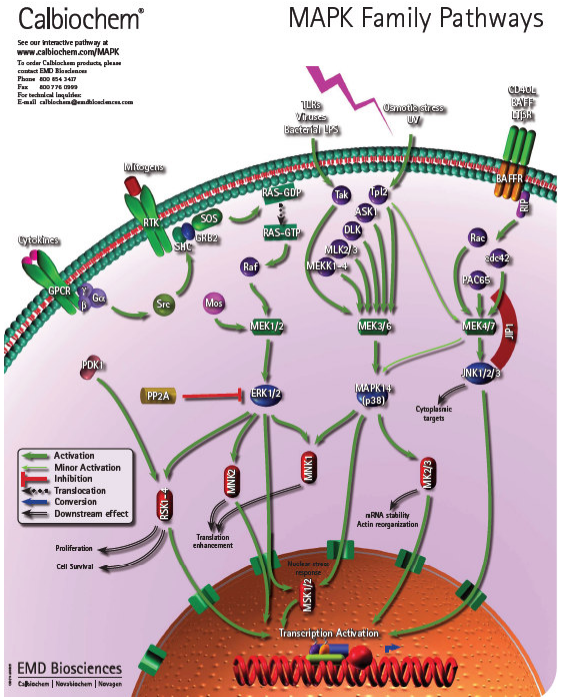
The systems that we build should be suitable for discovery

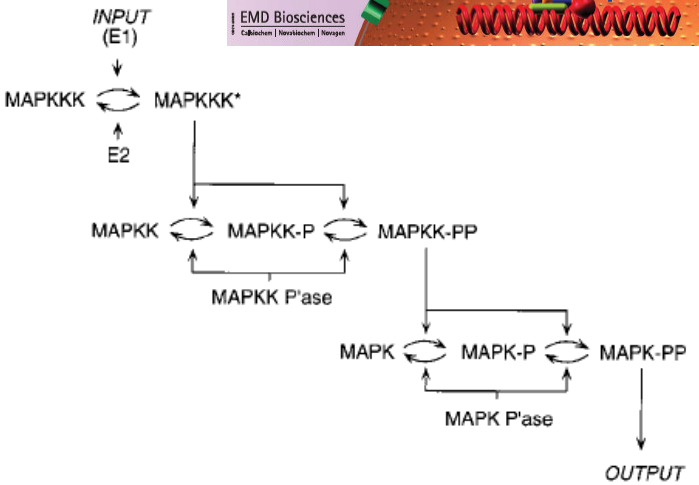
Scientific Method vs. Engineering Method



Motivation: Cells Compute

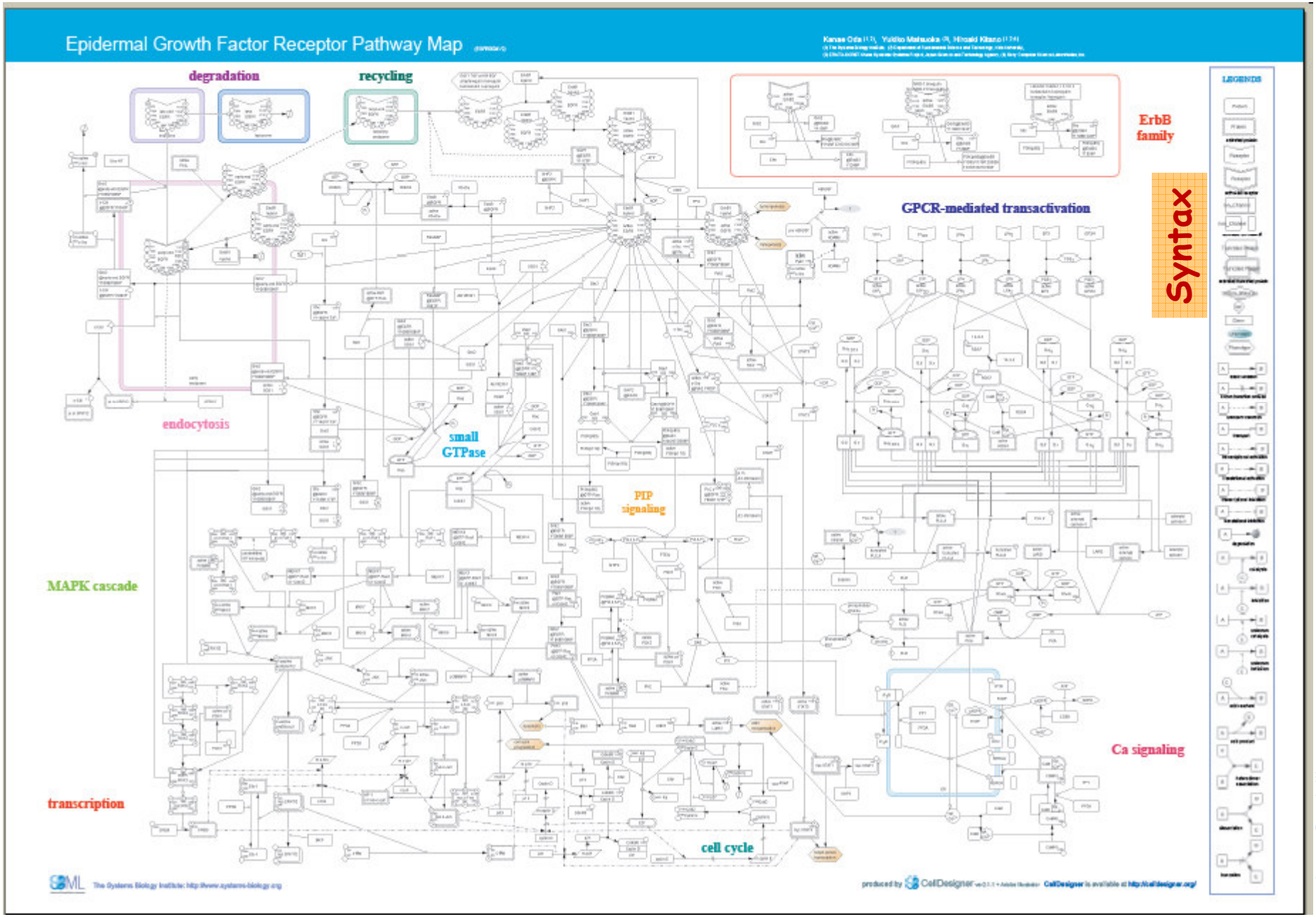
- No survival without computation!
  - Finding food
  - Avoiding predators
- *How* do they compute?
  - Unusual computational paradigms.
  - Proteins: do they work like electronic circuits?
  - Genes: what kind of software is that?
- Signaling networks
  - Clearly "information processing"
  - They are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

Discrete State Transitions



Modeling Approach

- We believe that {petri nets, process algebra, term rewriting, multiagent systems} are {better, complementary} for modeling biological systems than {SBML, Kohn charts, chemical reactions, ODEs}.

- We take a paper from the literature (usually ODEs or chemical reactions) and "code it up" in e.g. Petri nets.

- How do we know that's the "same system" ? How do we convince mathematical biologists that we are doing the "right thing"?

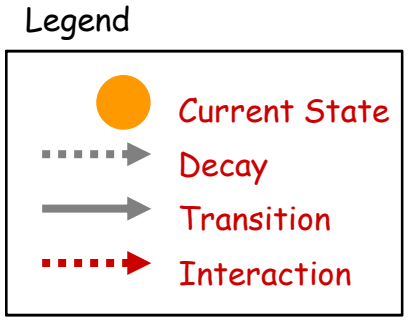
(Macro-) Molecules as (Interacting) Automata

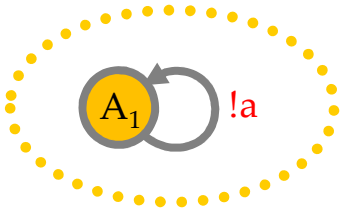
Process Algebra

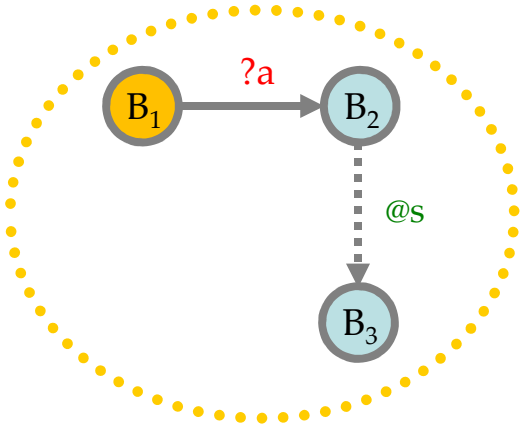
[Hoare, Milner, Pnueli, etc.]

- Reactive systems (living organisms, computer networks, operating systems, …)
  - Math is based on *entities that react/interact with their environment ("processes")*, not on functions from domains to codomains.
- Concurrent
  - Events (reactions/interactions) happen concurrently and asynchronously, not sequentially like in function composition.
- Stochastic
  - Or probabilistic, or nondeterministic, but is never about deterministic system evolution.
- Stateful
  - Each concurrent activity ("process") maintains its own local state, as opposed to stateless functions from inputs to outputs.
- Discrete
  - Evolution through discrete transitions between discrete states, not incremental changes of continuous quantities.
- Kinetics of interaction
  - An "interaction" is anything that moves a system from one state to another.

Interacting Automata



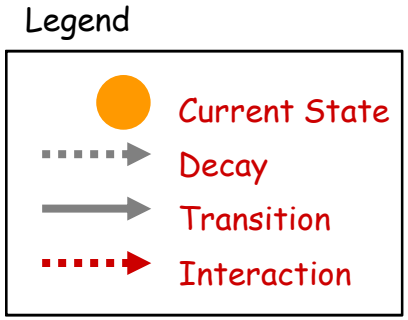


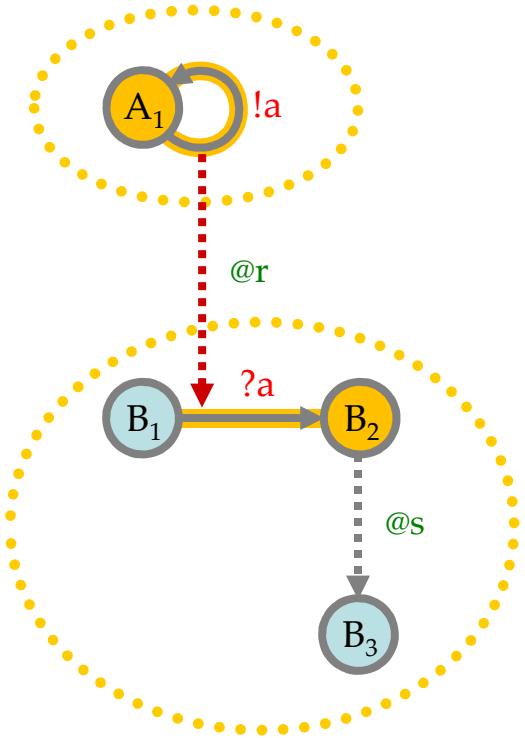


A_1	is a *state*
a	is a *channel* i.e. a named *interaction interface* (e.g. a surface patch)
?,!	indicate any *complementarity* of interaction (e.g. charge)
?a, !a	indicate *complementary actions*,
@r, @s	are rates

Kinetic laws:

Interacting Automata



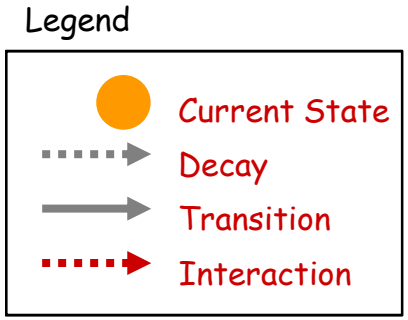


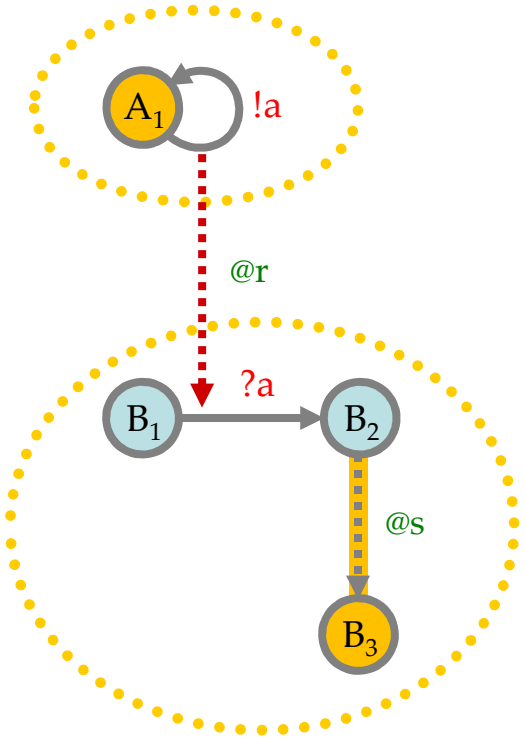
A_1	is a *state*
a	is a *channel* i.e. a named *interaction interface* (e.g. a surface patch)
?,!	indicate any *complementarity* of interaction (e.g. charge, shape)
?a, !a	indicate *complementary actions*, joined by an interaction arrow ⋯▶
@r, @s	are rates

Kinetic laws:

Two complementary actions may result in an interaction.

Interacting Automata





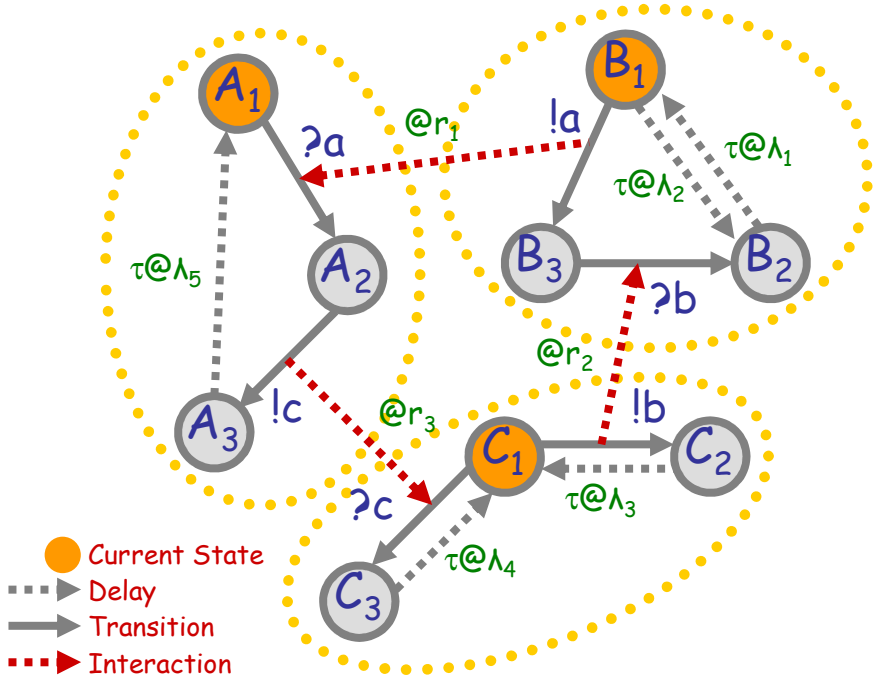
A_1	is a *state*
a	is a *channel* i.e. a named *interaction interface* (e.g. a surface patch)
?,!	indicate any *complementarity* of interaction (e.g. charge)
?a, !a	indicate *complementary actions*, joined by an interaction arrow ⋯▸
@r, @s	are rates

Kinetic laws:

Two complementary actions may result in an interaction.

A decay may happen spontaneously.

Interacting Automata



Interactions have rates. Actions DO NOT have rates.

The equivalent process algebra model

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

Communication channels

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

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

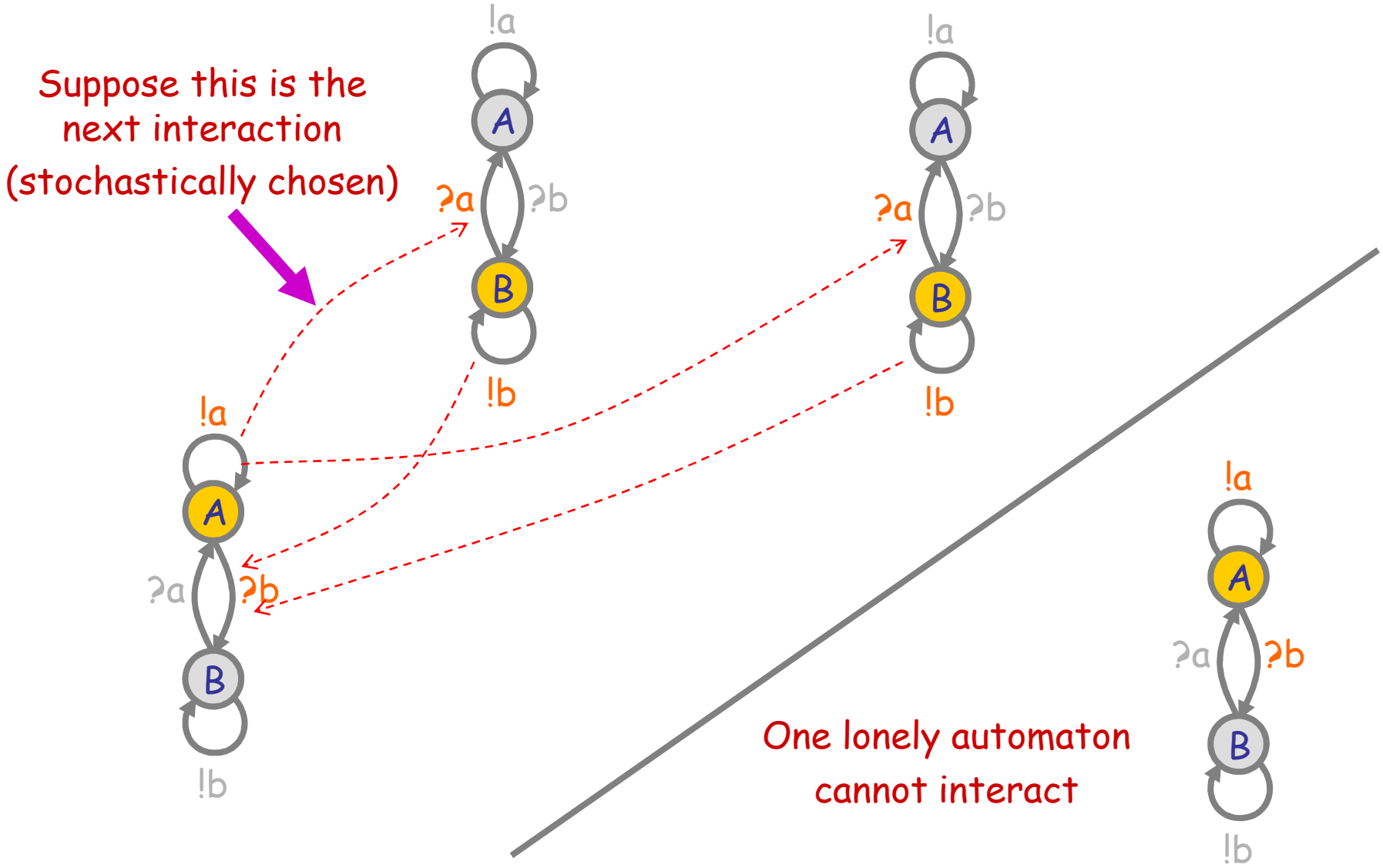
$C_1 = !b;\ C_2 + ?c;\ C_3$
$C_2 = \tau@\lambda_3;\ C_1$
$C_3 = \tau@\lambda_4;\ C_2$

Automata

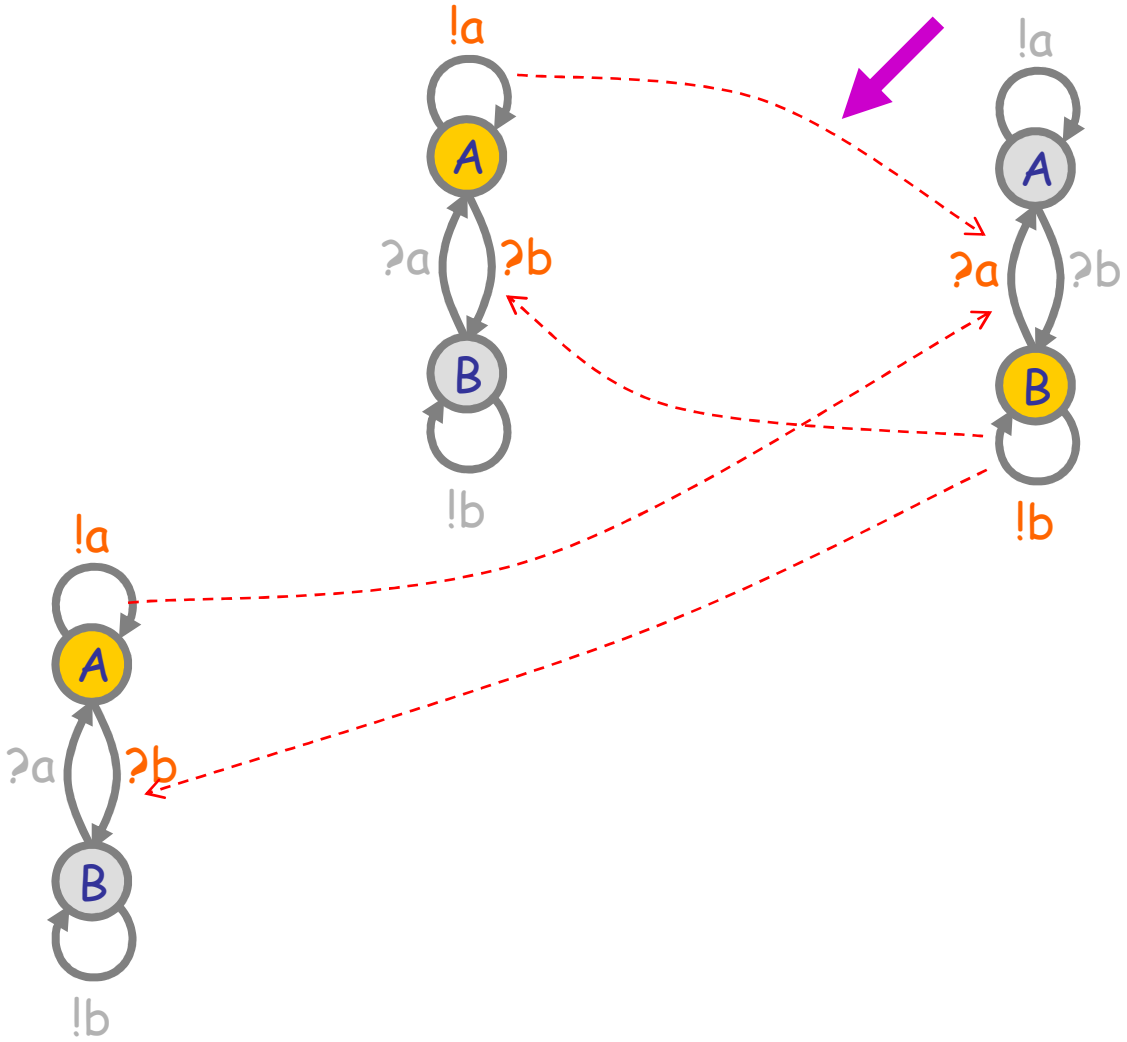
$A_1 \mid B_1 \mid C_1$

The system and initial state

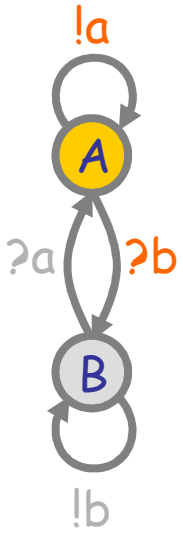
Interactions in a Population

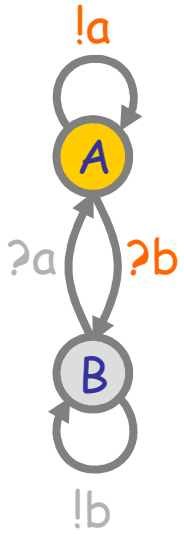


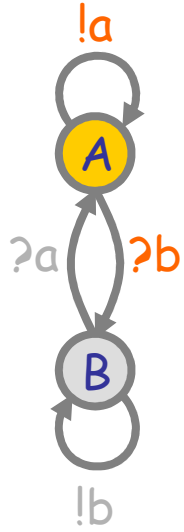
Interactions in a Population



Interactions in a Population

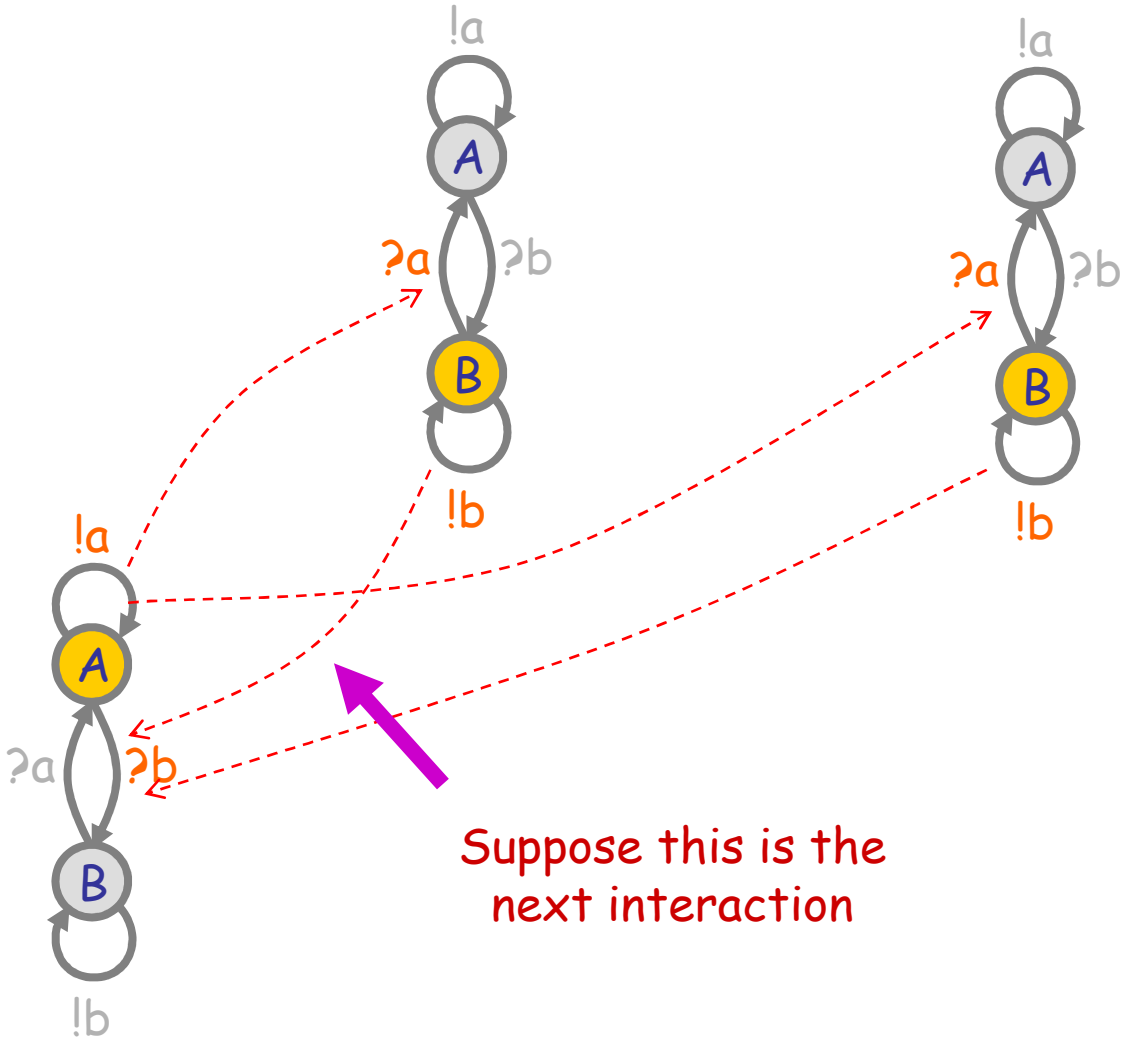




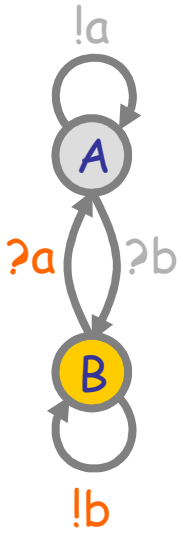


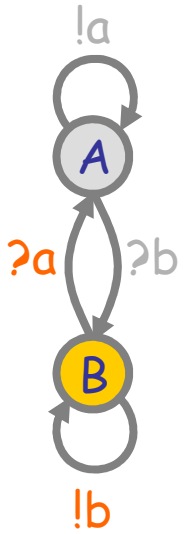
All-A stable population

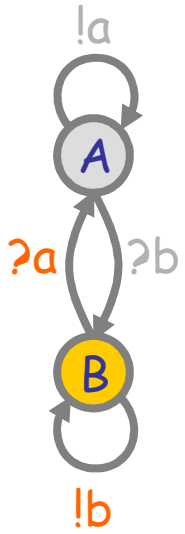
Interactions in a Population (2)



Interactions in a Population (2)



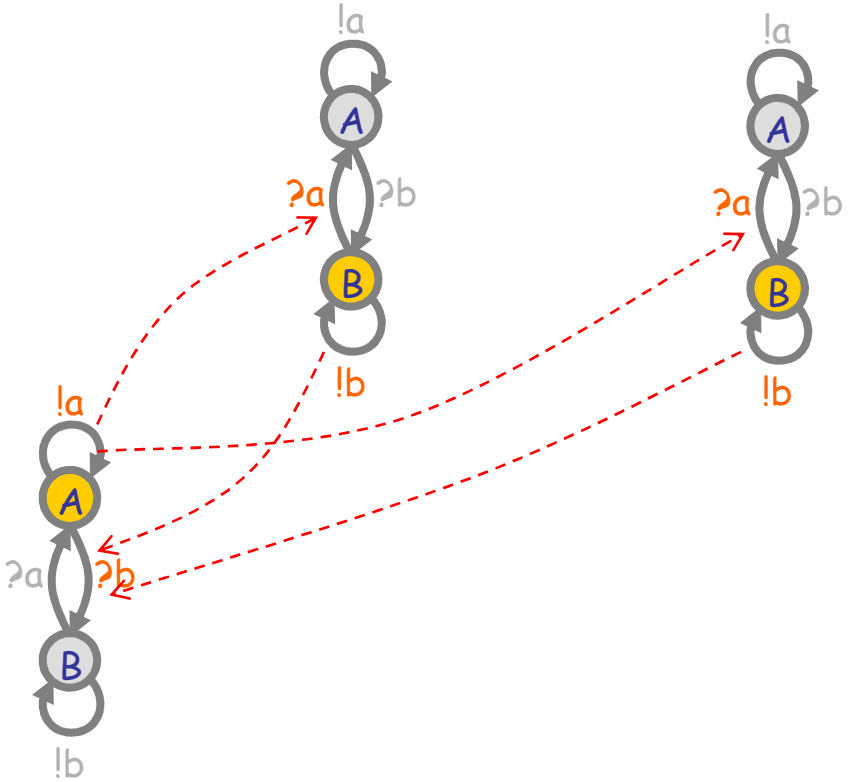


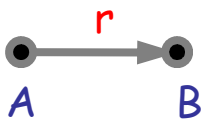


All-B stable population

Nondeterministic population behavior ("multistability")

CTMC Semantics



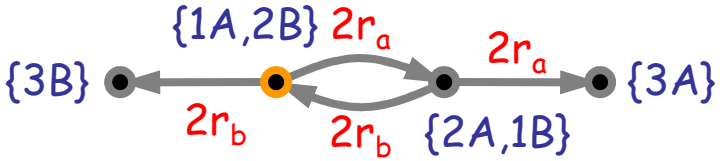


CTMC
(homogeneous) Continuous Time Markov Chain
- directed graph with no self loops
- nodes are system states
- arcs have transition rates

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

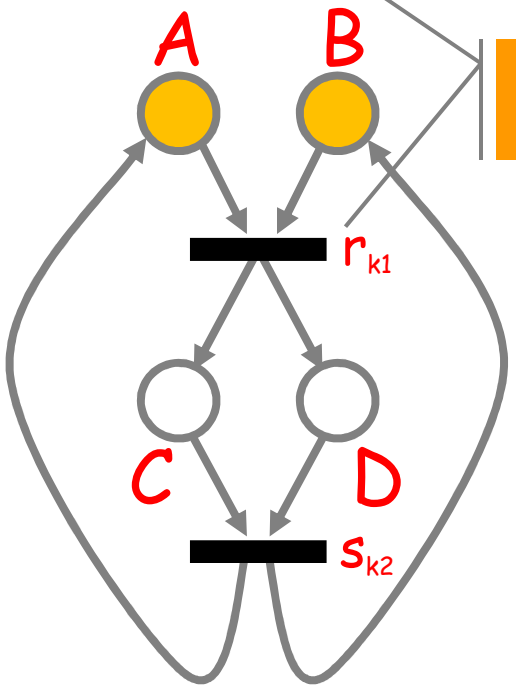
Reactions vs. Components

Says what "A" *does*.

$r: A + B \rightarrow_{k1} C + D$

$s: C + D \rightarrow_{k2} A + B$

Does A become C or D?

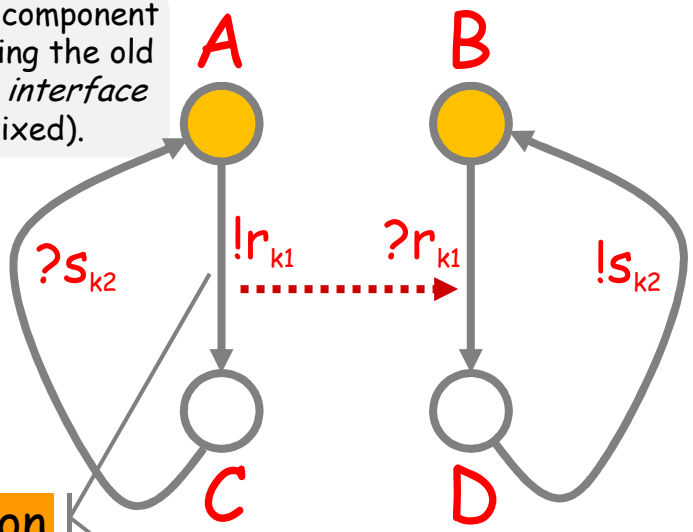


Reaction oriented

1 line per reaction

Says what "A" *is*.

Can add a new component without changing the old ones (if their *interface* remains fixed).



Interaction oriented

1 line per component

$A = !r_{k1}; C$

$C = ?s_{k2}; A$

$B = ?r_{k1}; D$

$D = !s_{k2}; B$

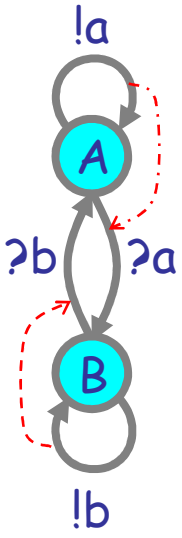
A becomes C not D!

The same "state space"

CTMC

Groupies and Celebrities

Groupies and Celebrities



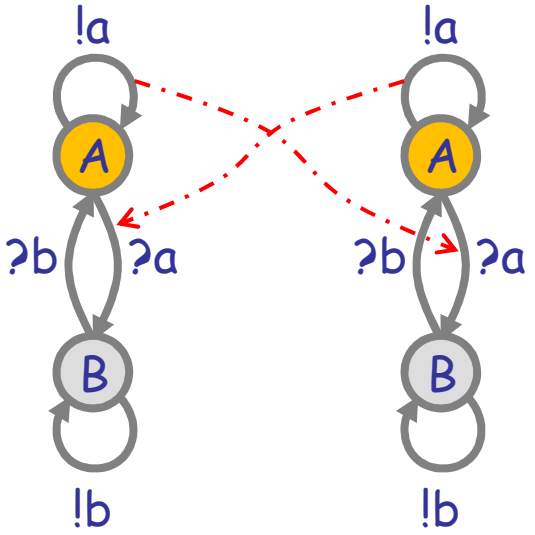
Celebrity

(does not want to be like somebody else)

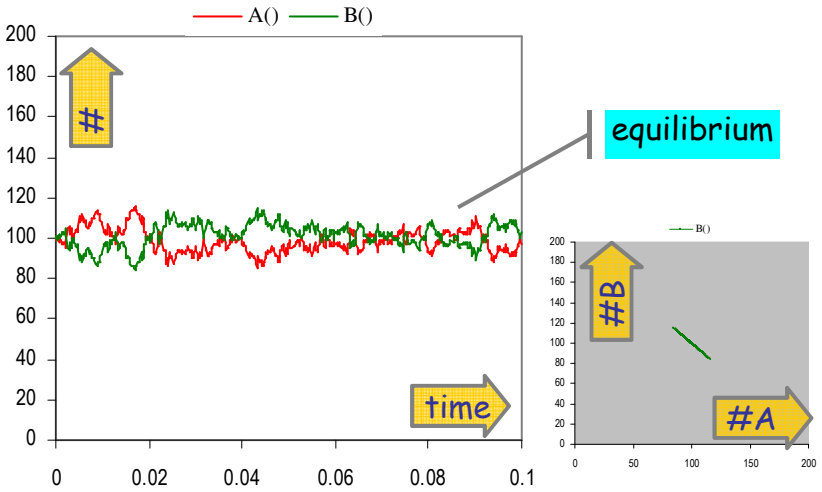
```
directive sample 1.0 1000
directive plot A(); B()

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

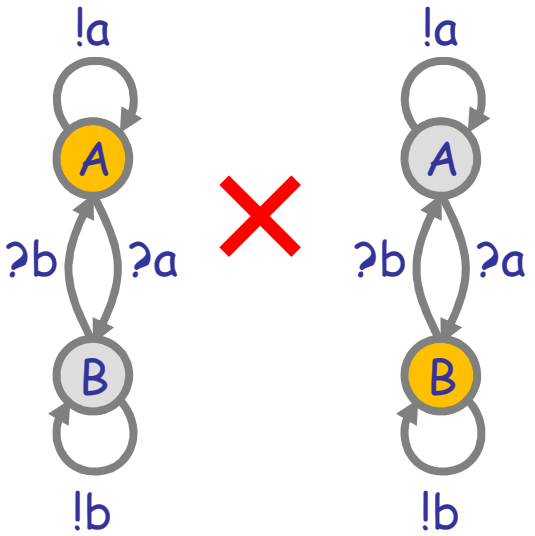
b@1.0



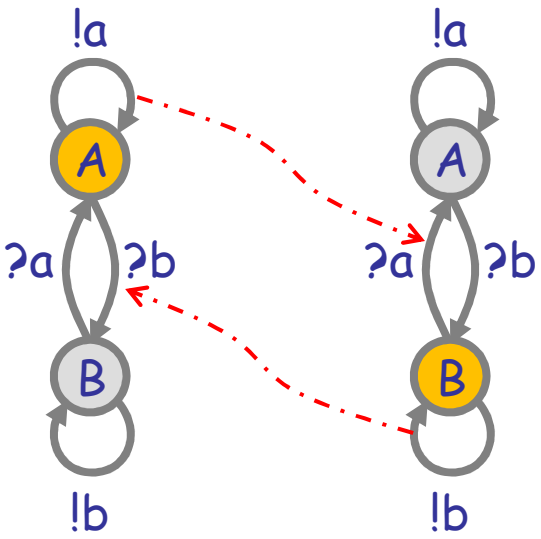
A stochastic collective of celebrities:

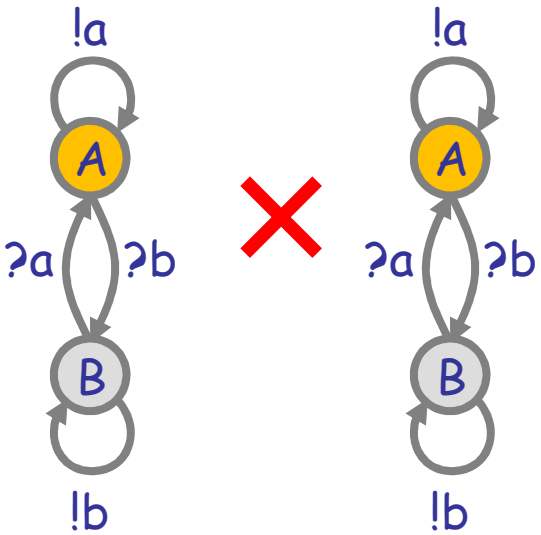


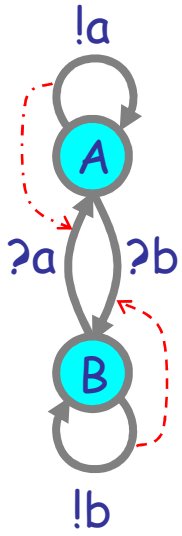
Stable because as soon as a A finds itself in the majority, it is more likely to find somebody in the same state, and hence change, so the majority is weakened.



Groupies and Celebrities







Groupie

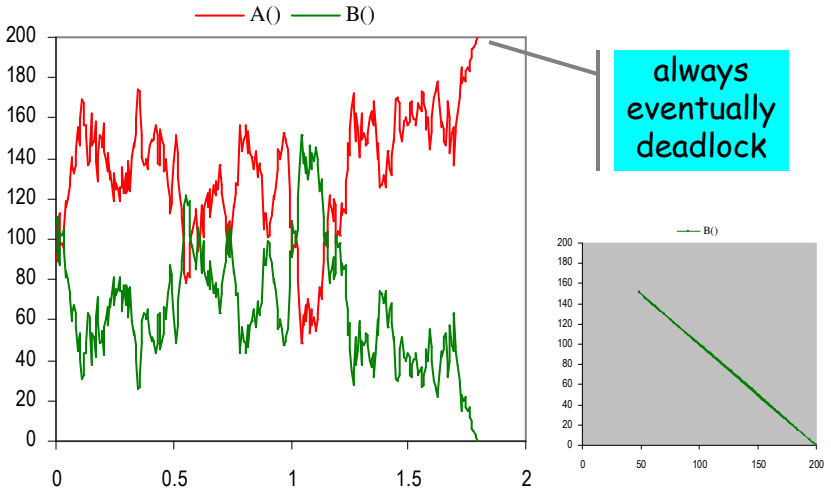
(wants to be like somebody different)

```
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:

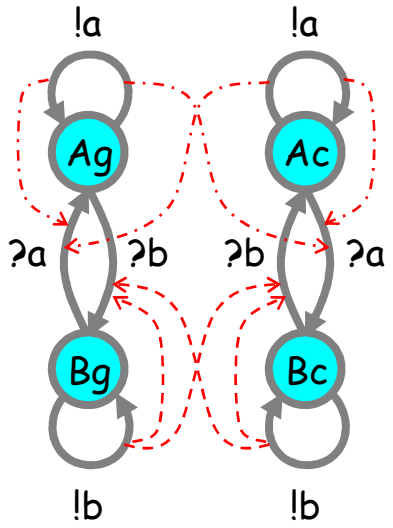


Unstable because within an A majority, an A has difficulty finding a B to emulate, but the few B's have plenty of A's to emulate, so the majority may switch to B. Leads to deadlock when everybody is in the same state and there is nobody different to emulate.

Both Together

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

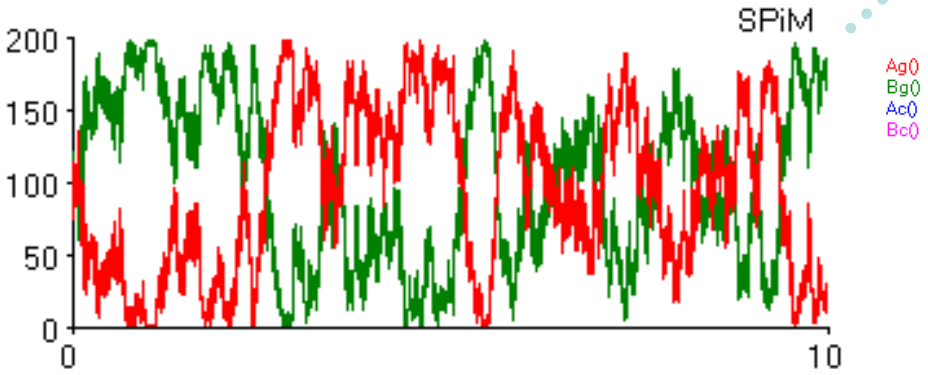
Many Groupies



A few Celebrities

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

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

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

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

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

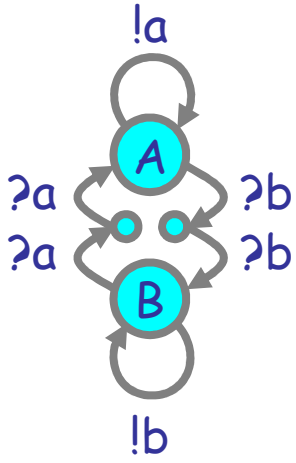
never deadlock

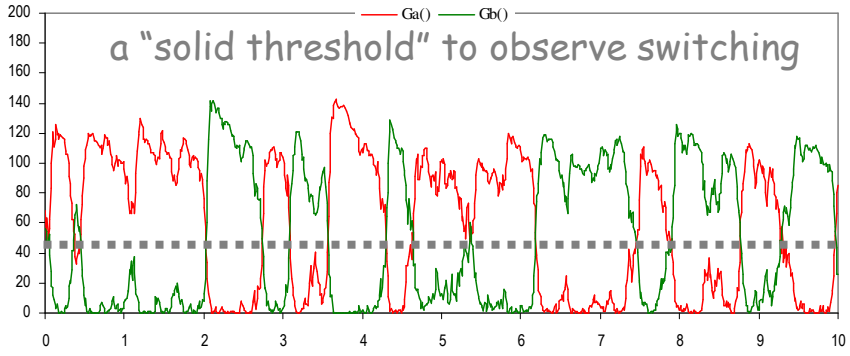


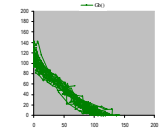
A tiny bit of "noise" can make a huge difference

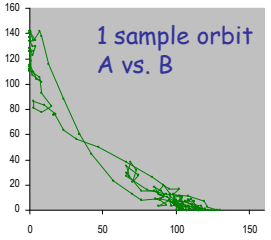
Hysteric Groupies

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

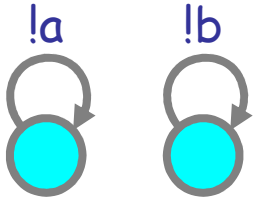








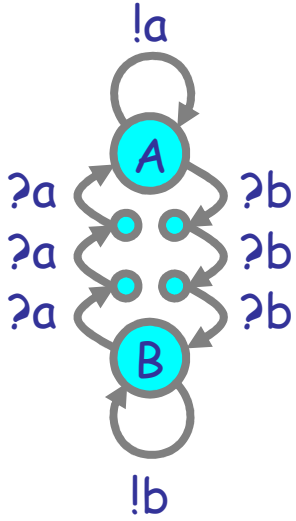

```
directive sample 10.0 1000
directive plot Ga(); Gb()

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())
```

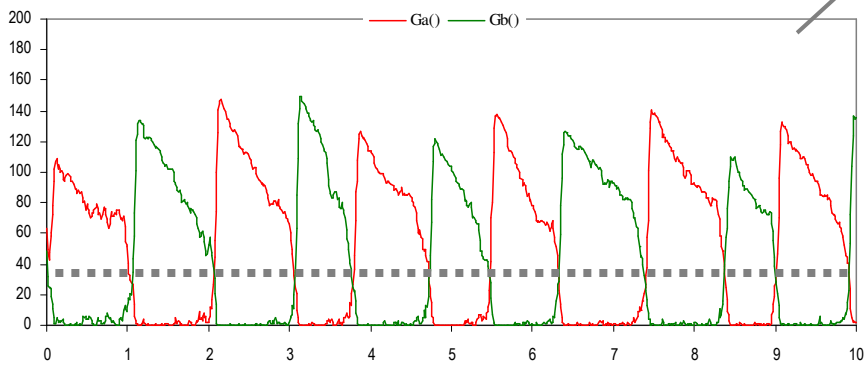


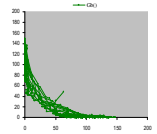
(With doping to break deadlocks)

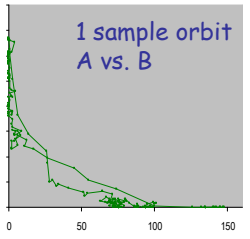
N.B.: It will not oscillate without doping (noise)

"regular" oscillation





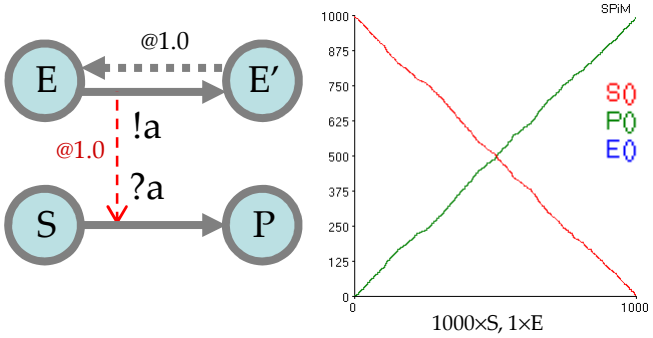




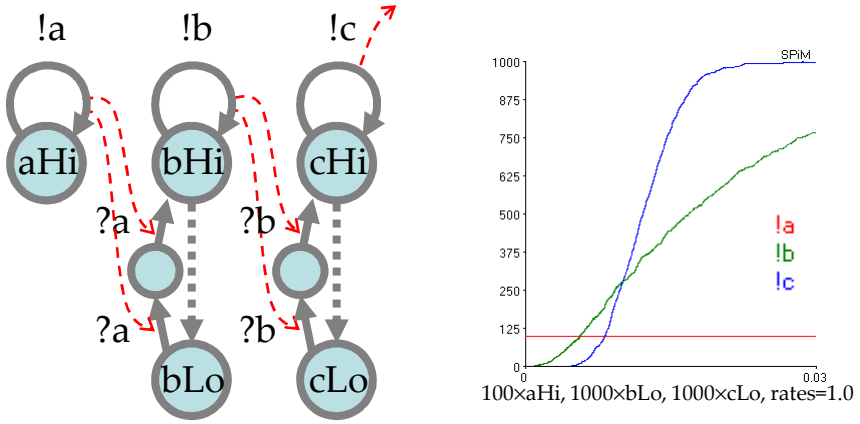

```
directive sample 10.0 1000
directive plot Ga(); Gb()

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())
```

Some Devices

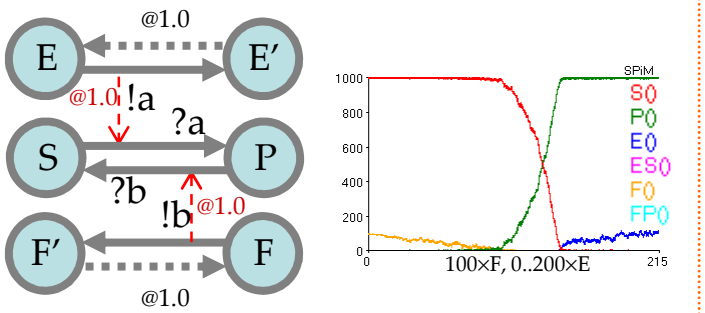
Linear Pump



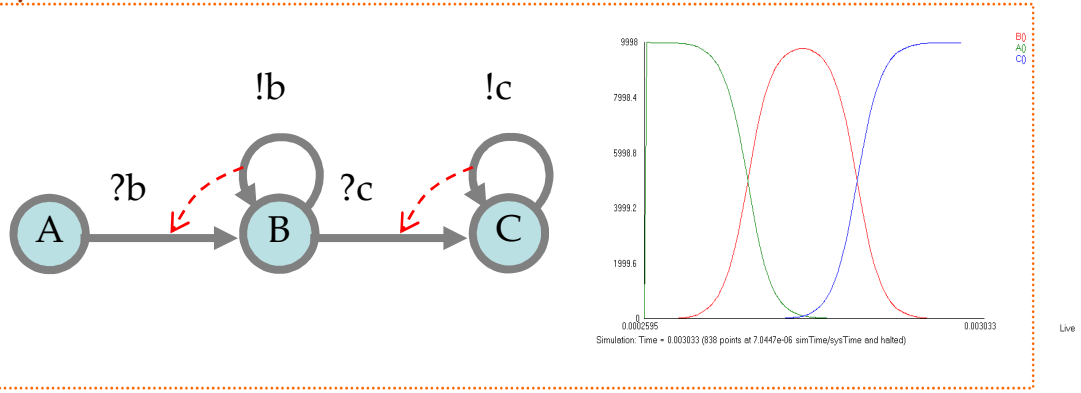
Cascade Amplifier



Ultrasensitive Switch

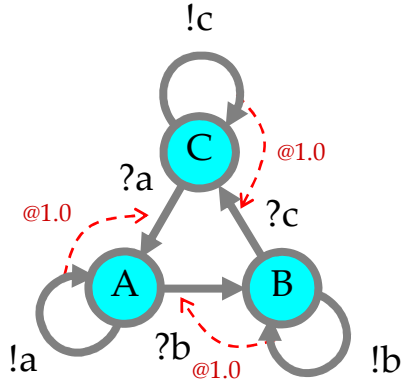


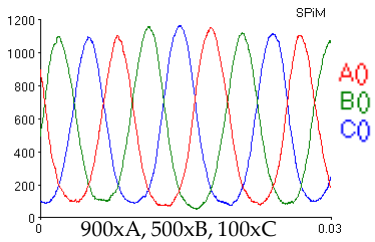
Symmetric Wave Generator



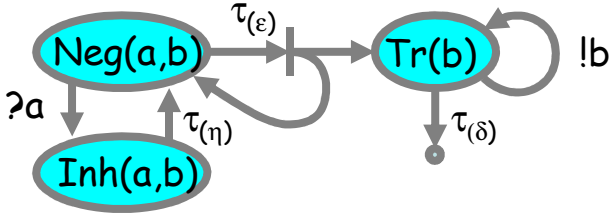
More Devices

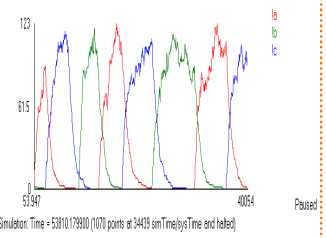
Oscillator



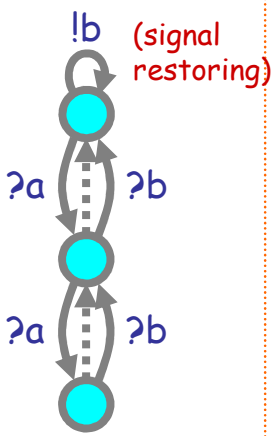


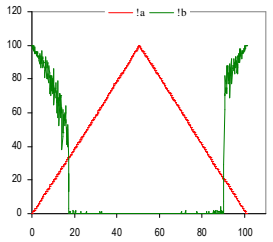
Repressilator (1 of 3 similar gates)



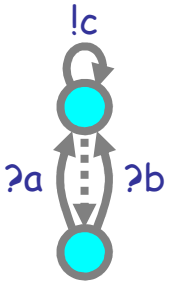


b = not a

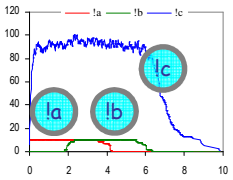




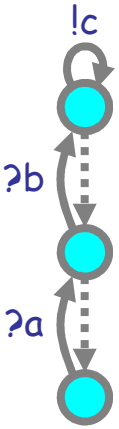
c = a or b

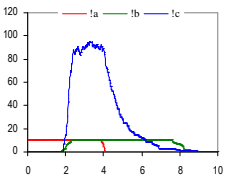


Inputs:
10 !a for 4t
2t; 10 !b for 4t

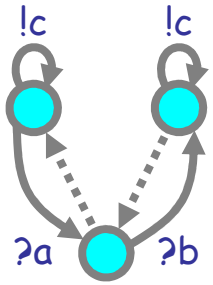


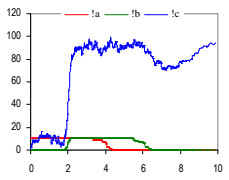
c = a and b



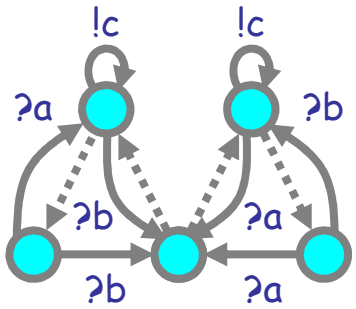


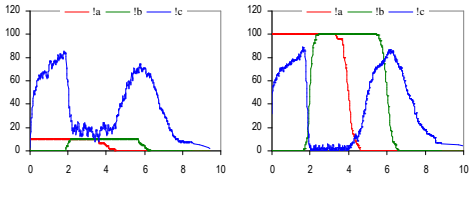
c = a imply b





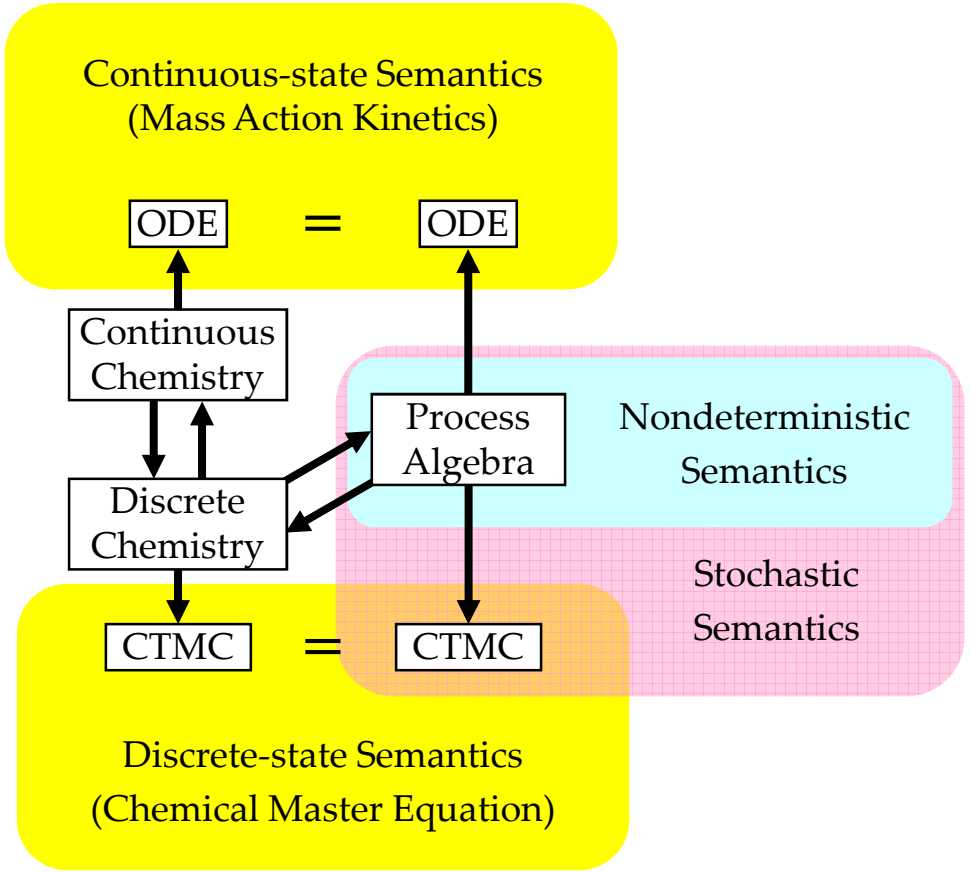
c = a xor b





Semantics of Collective Behavior

The Two Semantic Sides of Chemistry

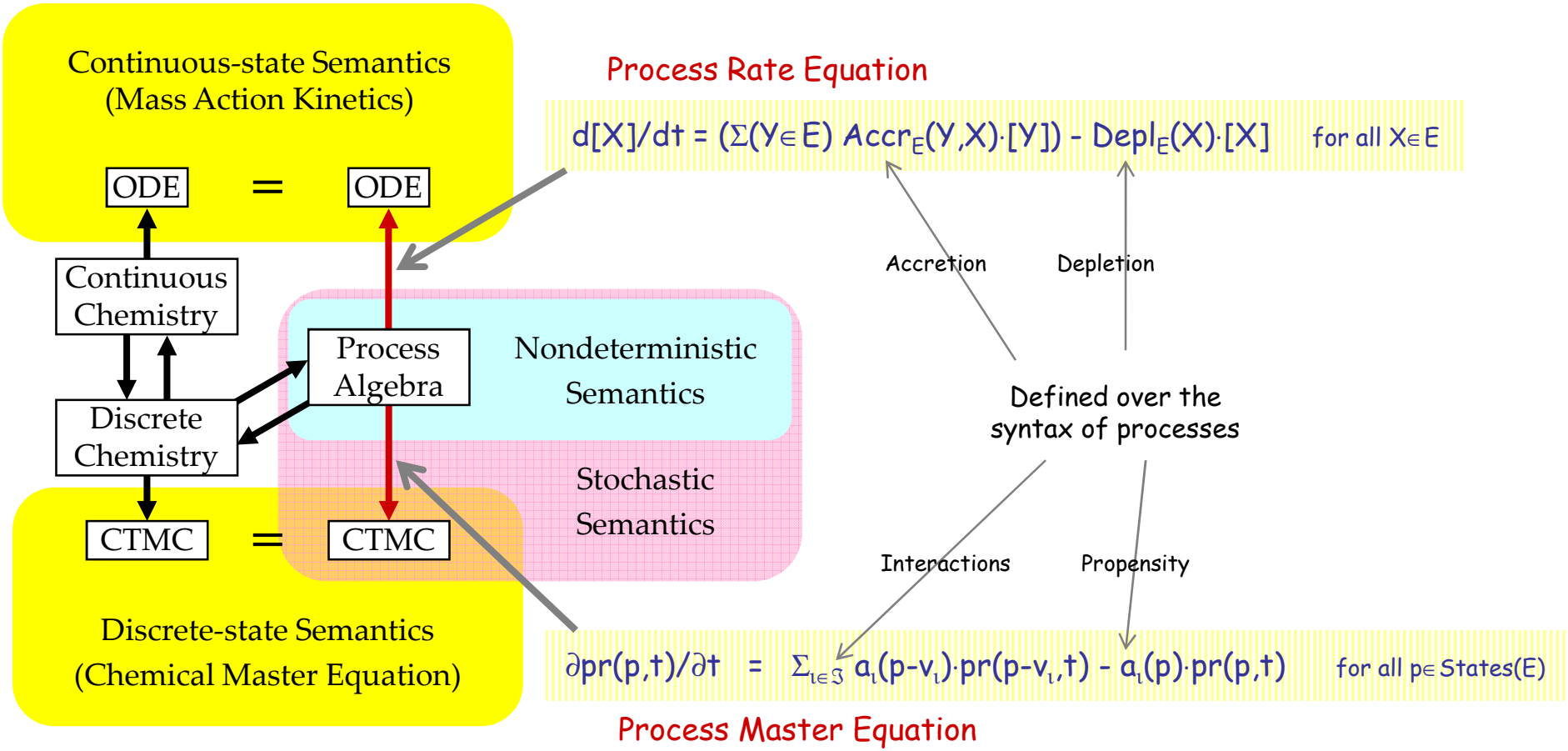


These diagrams commute via appropriate maps.

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

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

Quantitative Process Semantics



From CGF to Chemistry

Chemical Reactions

$A \rightarrow^{r} B_1 + \ldots + B_n$ $(n \geq 0)$	Unary Reaction	$d[A]/dt = -r[A]$	Exponential Decay
$A_1 + A_2 \rightarrow^{r} B_1 + \ldots + B_n$ $(n \geq 0)$	Hetero Reaction	$d[A_i]/dt = -r[A_1][A_2]$	Mass Action Law
$A + A \rightarrow^{r} B_1 + \ldots + B_n$ $(n \geq 0)$	Homeo Reaction	$d[A]/dt = -2r[A]^2$	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:

$E + S \leftrightarrow ES$

$ES \rightarrow P + E$

Chemical Ground Form (CGF)

$E ::= 0 \mid X=M, E$ Reagents

$M ::= 0 \mid \pi;P \oplus M$ Molecules

$P ::= 0 \mid X \mid P$ Solutions

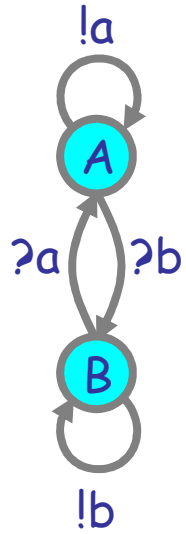
$\pi ::= \tau_{(r)} \mid ?a_{(r)} \mid !a_{(r)}$ Actions (delay, input, output)

$CGF ::= E,P$ Reagents plus Initial Conditions

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

(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)
0 is the null solution ($P|0 = 0|P = P$)
and null molecule ($M \oplus 0 = 0 \oplus M = M$)
Each X in E is a distinct *species*
Each name a is assigned a fixed rate r: $a_{(r)}$



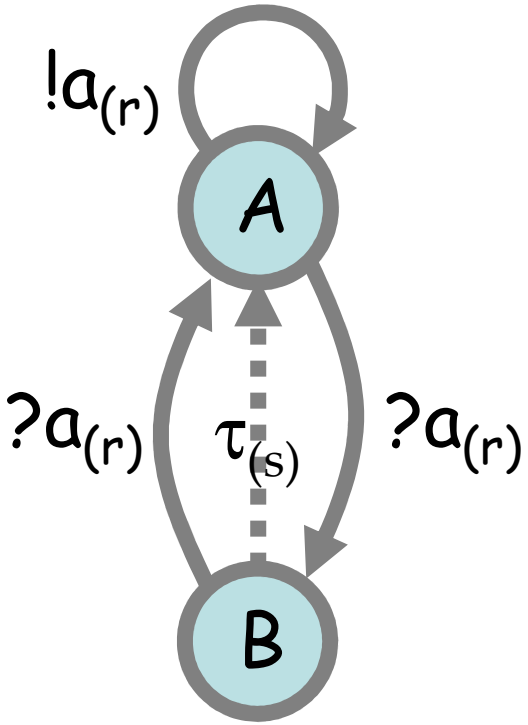
Ex: Interacting Automata
(= finite-control CGFs: they use "|" only in initial conditions):

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

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

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

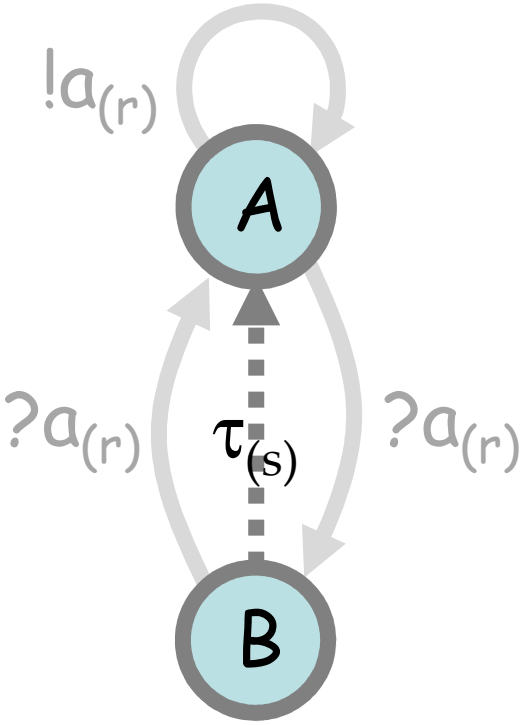
From CGF to Chemistry (by example)

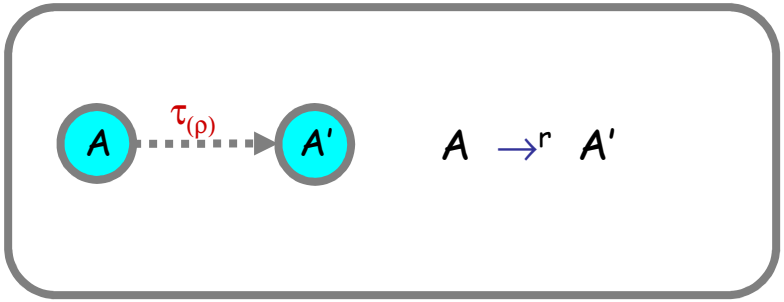


$A = !a_{(r)};A \oplus ?a_{(r)};B$

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

From CGF to Chemistry (by example)





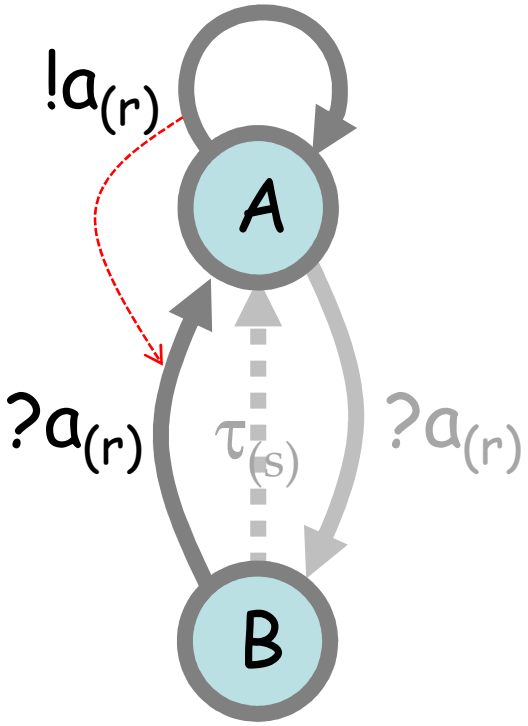
B →s A

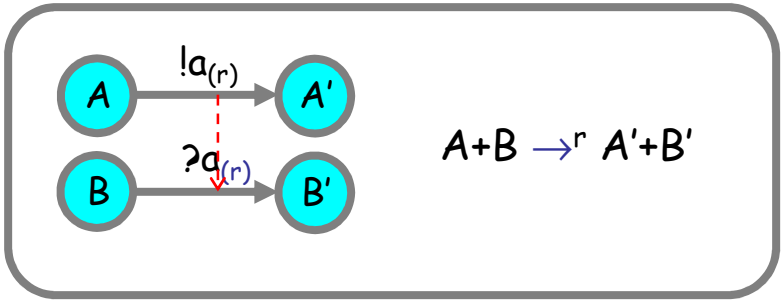
A = !a;A ⊕ ?a;B

B = ?a;A ⊕

τ(s);A

From CGF to Chemistry (by example)





$B \rightarrow^{s} A$

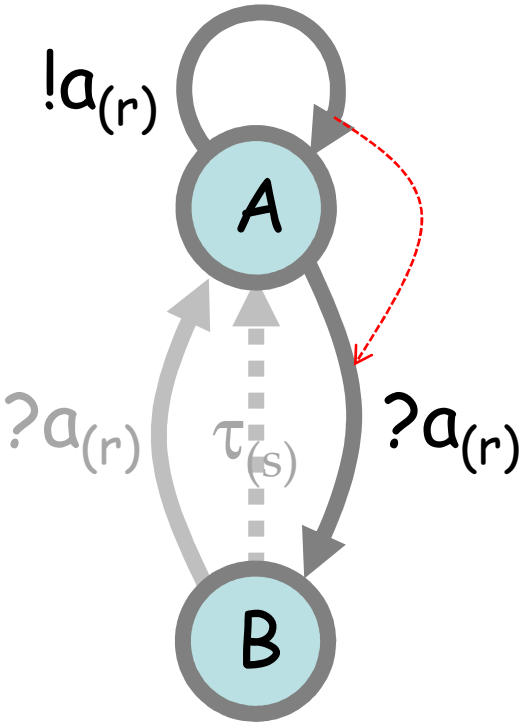
$A+B \rightarrow^{r} A+A$

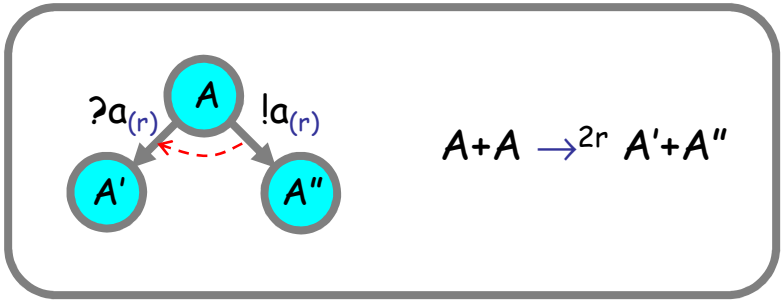
$A = !a;A \oplus ?a;B$

$B = ?a;A \oplus$

$\tau_{(s)};A$

From CGF to Chemistry (by example)





$B \rightarrow^{s} A$

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

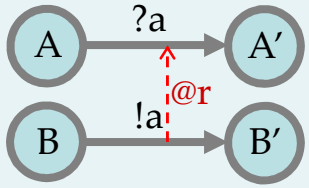
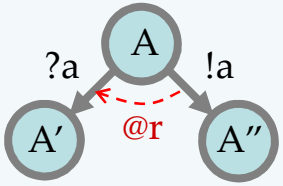
$A+A \rightarrow^{2r} A+B$

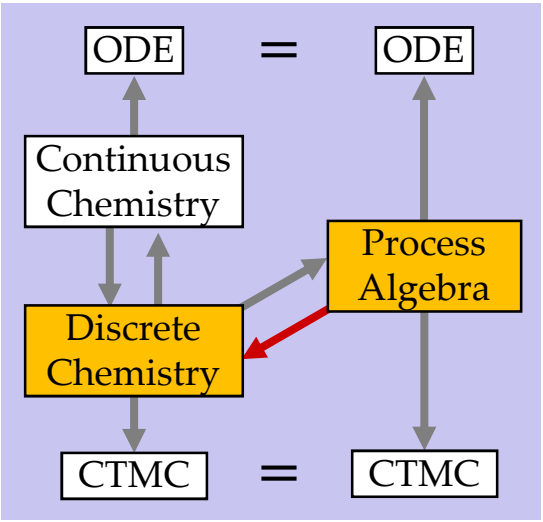
Double rate for homeo reactions

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

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

From CGF to Chemistry (by example)

Interacting Automata →	Discrete Chemistry
initial states A | A | ... | A	initial quantities $\#A_0$
	$A \dashrightarrow^{r} A'$
	$A+B \dashrightarrow^{r} A'+B'$
	$A+A \dashrightarrow^{2r} A'+A''$



From CGF to Chemistry: Ch(E)

$E ::= 0 \mid X=M, E$	Reagents
$M ::= 0 \mid \pi;P \oplus M$	Molecules
$P ::= 0 \mid X \mid P$	Solutions
$\pi ::= \tau_{(r)} \mid ?a_{(r)} \mid !a_{(r)}$	Interactions (delay, input, output)
$CGF ::= E,P$	Reagents plus Initial Conditions

$E.X.i \stackrel{def}{=}$ the i-th $\oplus$-summand of the molecule M associated with the X reagent of E

Chemical reactions for E,P: (N.B.: <...> are reaction tags to obtain multiplicity of reactions, and P is P with all the | changed to +)

$$
\begin{aligned}
&Ch(E) := \\
&\quad \{(\langle X.i \rangle : X \rightarrow^{r} P) \ s.t. \ E.X.i = \tau_{(r)};P\} \cup \\
&\quad \{(\langle X.i,Y.j \rangle : X + Y \rightarrow^{r} P + Q) \ s.t. \ X \neq Y, \ E.X.i = ?a_{(r)};P, \ E.Y.j = !a_{(r)};Q\} \cup \\
&\quad \{(\langle X.i,X.j \rangle : X + X \rightarrow^{2r} P + Q) \ s.t. \ E.X.i = ?a_{(r)};P, \ E.X.j = !a_{(r)};Q) \rangle \in E\}
\end{aligned}
$$

Initial conditions for P:

$$Ch(P) := P$$

From Chemistry to CGF

From Chemistry to CGF (by example)

x: $B \rightarrow^{s} A$

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

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

Unique reaction names

	$x_{(s)}$	$b_{(r)}$	$c_{(r)}$
A			
B			

Reactions names

Half-rate for homeo reactions

Species

From Chemistry to CGF (by example)

x: $B \rightarrow^{s} A$

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

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

	$x_{(s)}$	$b_{(r)}$	$c_{(r)}$
A			
B	$\tau;A$		

1: Fill the matrix by columns:

Degradation reaction v_i: $X \rightarrow^{k_i} P_i$
add $t;P_i$ to $\langle X, v_i \rangle$.

From FSRN to CGF (by example)

x: $B \rightarrow^{s} A$

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

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

	$x_{(s)}$	$b_{(r)}$	$c_{(r)}$
A		?;A | A	
B	τ;A	!;0	

1: Fill the matrix by columns:

Degradation reaction v_i: $X \rightarrow^{k_i} P_i$
add t;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 $\langle Y, v_i \rangle$

From FSRN to CGF (by example)

x: $B \rightarrow^{s} A$

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

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

	$x_{(s)}$	$b_{(r)}$	$c_{(r)}$
A		?;A|A	?;A|B !;0
B	τ;A	!;0	

1: Fill the matrix by columns:

Degradation reaction v_i: $X \rightarrow k_i P_i$
add t;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 $\langle X, v_i \rangle$

From FSRN to CGF (by example)

x: $B \rightarrow^{s} A$

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

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

	$x_{(s)}$	$b_{(r)}$	$c_{(r)}$
A		?;A|A	?;A|B !;0
B	τ;A	!;0	

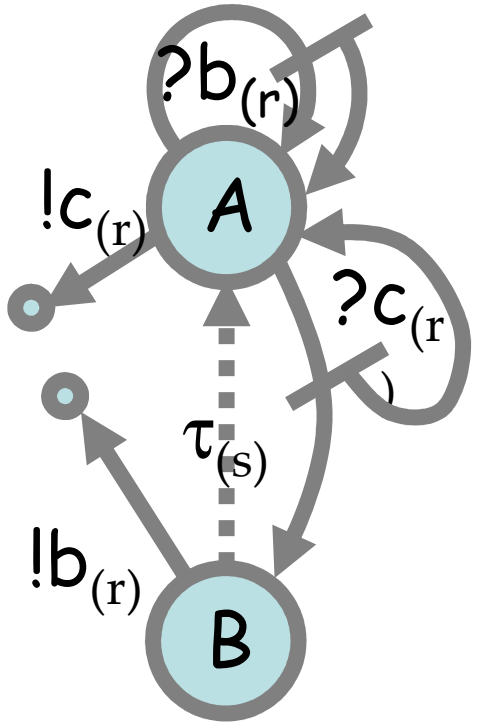
1: Fill the matrix by columns:

Degradation reaction v_i: $X \rightarrow k_i P_i$
add t;P_i to <X,v_i>.
Hetero reaction v_i: $X+Y \rightarrow k_i P_i$
add ?;P_i to <X,v_i> and !;0 to <Y,v_i>
Homeo reaction v_i: $X+X \rightarrow k_i P_i$
add ?;P_i and !;0 to <X,v_i>

2: Read the result by rows:

$A = ?b_{(r)};(A|A) \oplus ?c_{(r)};(A|B) \oplus !c_{(r)};0$

$B = \tau_{(s)};A \oplus !b_{(r)};0$



From FSRN to CGF (by example)

x: $B \rightarrow^{s} A$

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

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

	$x_{(s)}$	$b_{(r)}$	$c_{(r)}$
A		?;A	?;A|B !;0
B	τ;A	!;A	

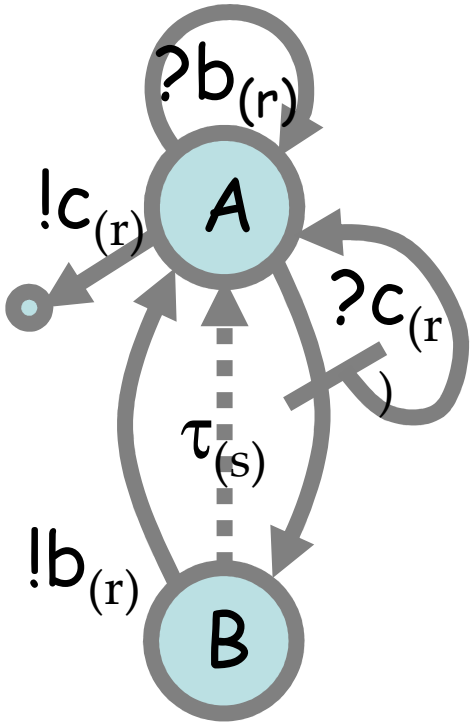
1: Fill the matrix by columns:

Degradation reaction v_i: $X \rightarrow k_i P_i$
add t;P_i to <X,v_i>.
Hetero reaction v_i: $X+Y \rightarrow k_i P_i$
add ?;P_i to <X,v_i> and !;0 to <Y,v_i>
Homeo reaction v_i: $X+X \rightarrow k_i P_i$
add ?;P_i and !;0 to <X,v_i>

2: Read the result by rows:

$A = ?b_{(r)};A \oplus ?c_{(r)};(A|B) \oplus !c_{(r)};0$

$B = t_{(s)};A \oplus !b_{(r)};A$



From FSRN to CGF (by example)

x: $B \rightarrow^{s} A$

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

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

	$x_{(s)}$	$b_{(r)}$	$c_{(r)}$
A		?;A	?;B !;A
B	τ;A	!;A	

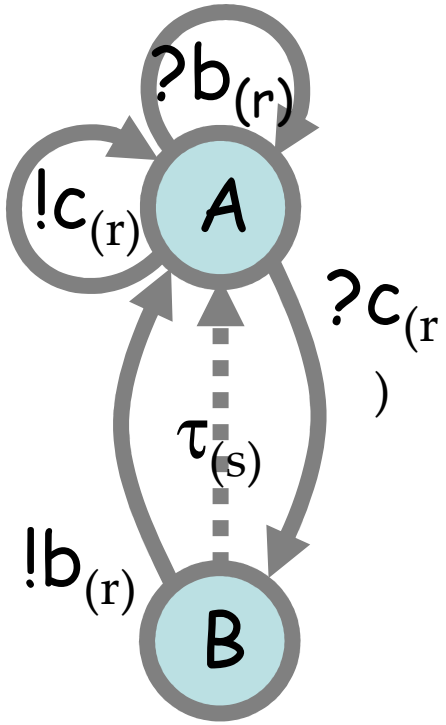
1: Fill the matrix by columns:

Degradation reaction v_i: $X \rightarrow k_i P_i$
add t;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 $\langle X, v_i \rangle$

2: Read the result by rows:

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

$B = t_{(s)};A \oplus !b_{(r)};A$



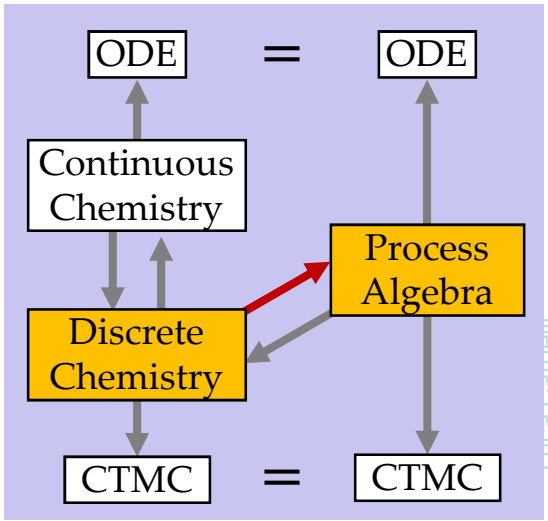
From Chemistry to CGF: Pi(C)

v: $X \rightarrow^{r} Y_1 + \ldots + Y_n + 0$ Unary Reaction

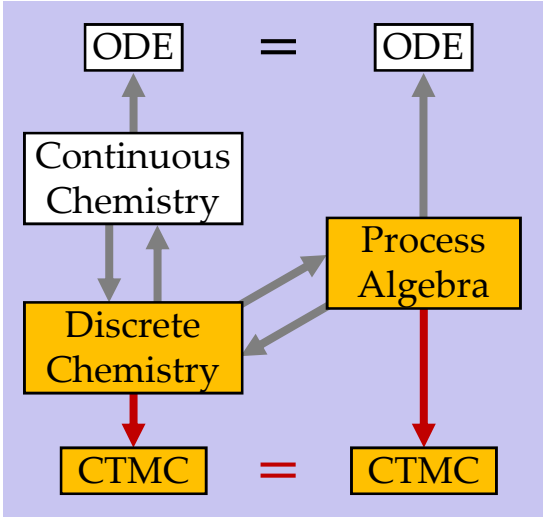
v: $X_1 + X_2 \rightarrow^{r} Y_1 + \ldots + Y_n + 0$ Binary Reaction

From uniquely-labeled (v:) chemical reactions *C* to a CGF Pi(*C*):

$$
\begin{aligned}
Pi(C) = \{(X = \; & \oplus((v: X \rightarrow^{k} P) \in C) \; of \; (\tau_{(k)}; P) \; \oplus \\
& \oplus((v: X+Y \rightarrow^{k} P) \in C \text{ and } Y \neq X) \; of \; (?v_{(k)}; P) \; \oplus \\
& \oplus((v: Y+X \rightarrow^{k} P) \in C \text{ and } Y \neq X) \; of \; (!v_{(k)}; 0) \; \oplus \\
& \oplus((v: X+X \rightarrow^{k} P) \in C) \; of \; (?v_{(k/2)}; P \oplus !v_{(k/2)}; 0) \;) \\
& \textit{s.t.} \; X \text{ is a species in } C\}
\end{aligned}
$$

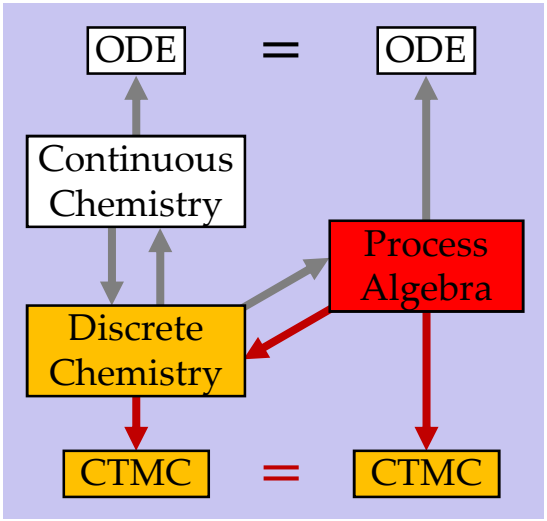


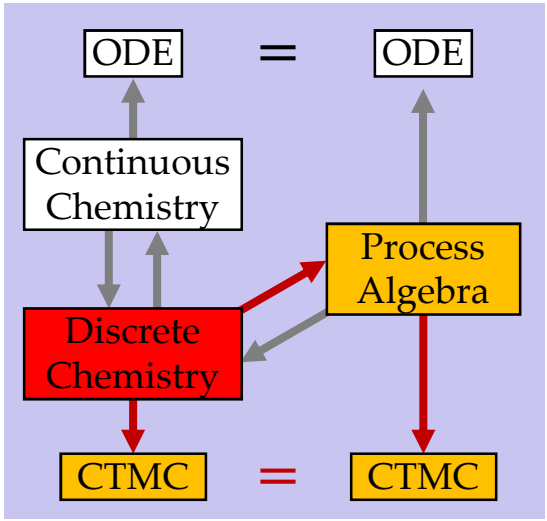
Discrete-State Semantics



Discrete State Equivalence

- Def: ≋ is equivalent CTMC's (isomorphic graphs with same rates).
- Thm: E ≋ Ch(E)
- Thm: C ≋ Pi(C)





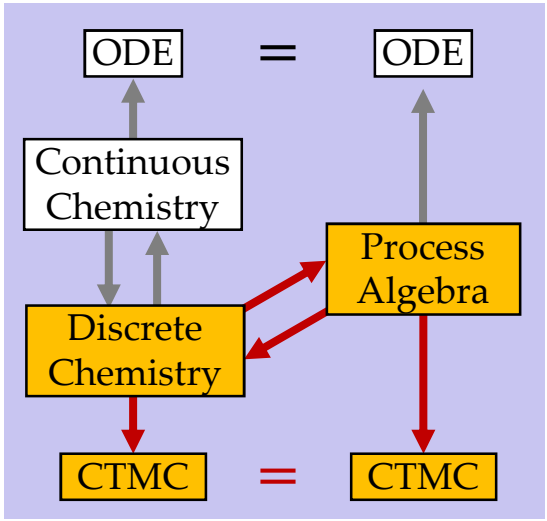
- For each E there is an E' ≋ E that is detangled (E' = Pi(Ch(E)))
- For each E in automata form there is an an E' ≋ E that is detangled and in automata form (E' = Detangle(E)).

Interacting Automata = 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).



From Discrete to Continuous Chemistry

Luca Cardelli

The Gillespie Conversion

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

V = interaction volume

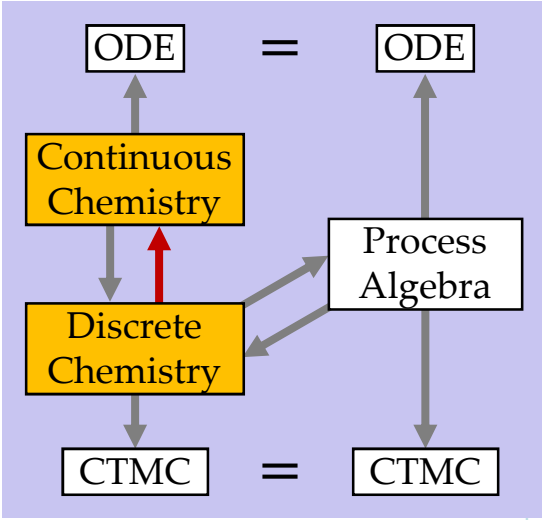
N_A = Avogadro's number

Think $\gamma = 1$

i.e. $V = 1/N_A$

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

molarity (concentration)



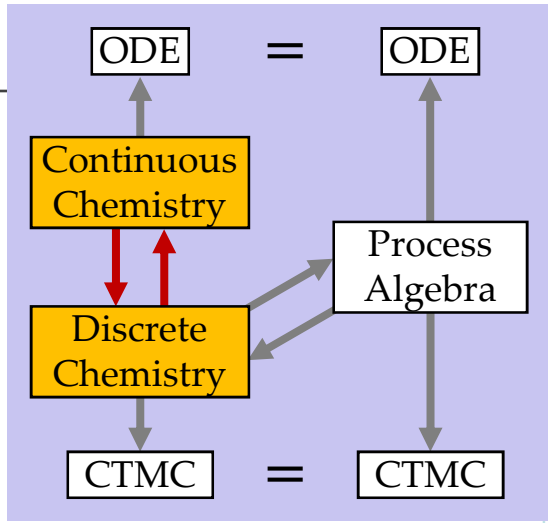
$Cont_\gamma$ and $Disc_\gamma$

4.2–3 Definition: $Cont_\gamma$ and $Disc_\gamma$

For a volumetric factor $\gamma{:}M^{-1}$, we define a translation $Cont_\gamma$ from a discrete chemical systems (C,P), with species X and initial molecule count $\#X_0 = \#X(P)$, to a continuous chemical systems (C,V) with initial concentration $[X]_0 = V_X$. The translation $Disc_\gamma$ is its inverse, up to a rounding error $\lceil \gamma[X]_0 \rceil$ in converting concentrations to molecule counts. Since γ is a global conversion constant, we later usually omit it as a subscript.

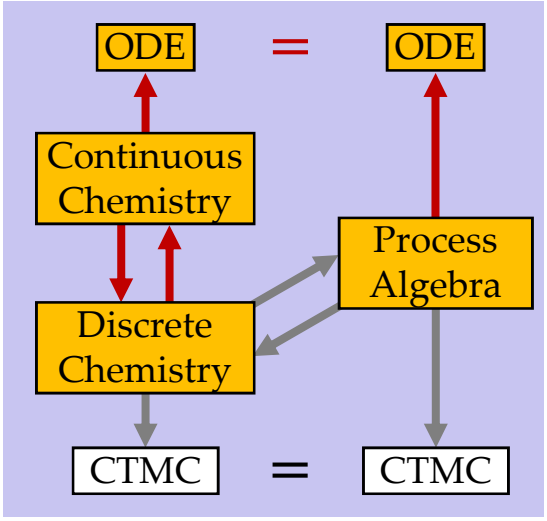
$Cont_\gamma(X \to^r P)$	$= X \to^k P$	with $k = r$,	$r{:}s^{-1}$	$k{:}s^{-1}$
$Cont_\gamma(X+Y \to^r P)$	$= X+Y \to^k P$	with $k = r\gamma$	$r{:}s^{-1}$	$k{:}M^{-1}s^{-1}$
$Cont_\gamma(X+X \to^r P)$	$= X+X \to^k P$	with $k = r\gamma/2$	$r{:}s^{-1}$	$k{:}M^{-1}s^{-1}$
$Cont_\gamma(\#X_0)$	$= [X]_0$	with $[X]_0 = \#X_0/\gamma$	$X_0{:}mol$	$[X]_0{:}M$
$Disc_\gamma(X \to^k P)$	$= X \to^r P$	with $r = k$,	$k{:}s^{-1}$	$r{:}s^{-1}$
$Disc_\gamma(X+Y \to^k P)$	$= X+Y \to^r P$	with $r = k/\gamma$	$k{:}M^{-1}s^{-1}$	$r{:}s^{-1}$
$Disc_\gamma(X+X \to^k P)$	$= X+X \to^r P$	with $r = 2k/\gamma$	$k{:}M^{-1}s^{-1}$	$r{:}s^{-1}$
$Disc_\gamma([X]_0)$	$= \#X_0$	with $\#X_0 = \lceil \gamma[X]_0 \rceil$	$[X]_0{:}M$	$X_0{:}mol$

$$Ch_\gamma := Cont_\gamma \circ Ch$$



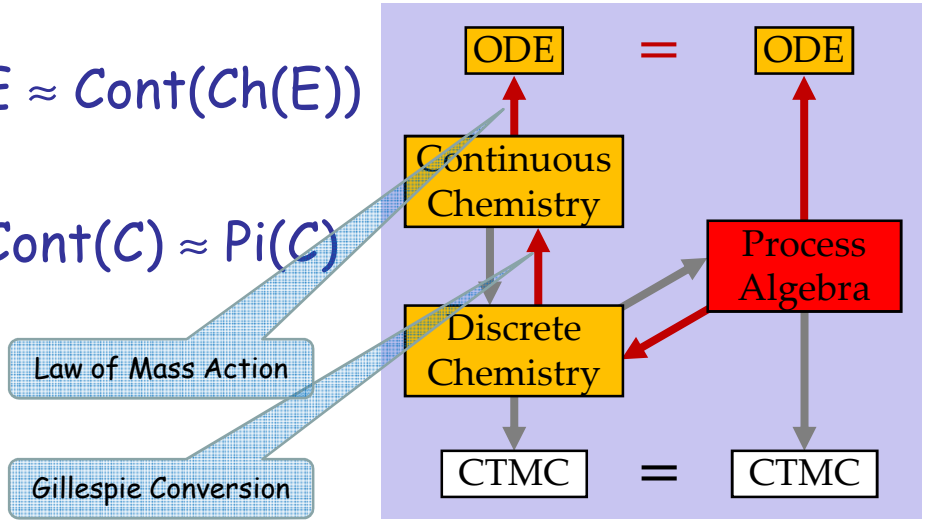
Continuous-State Semantics

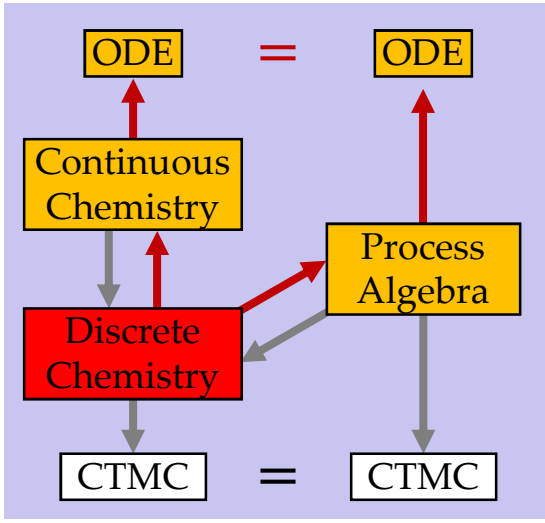
(summary)



Continuous State Equivalence

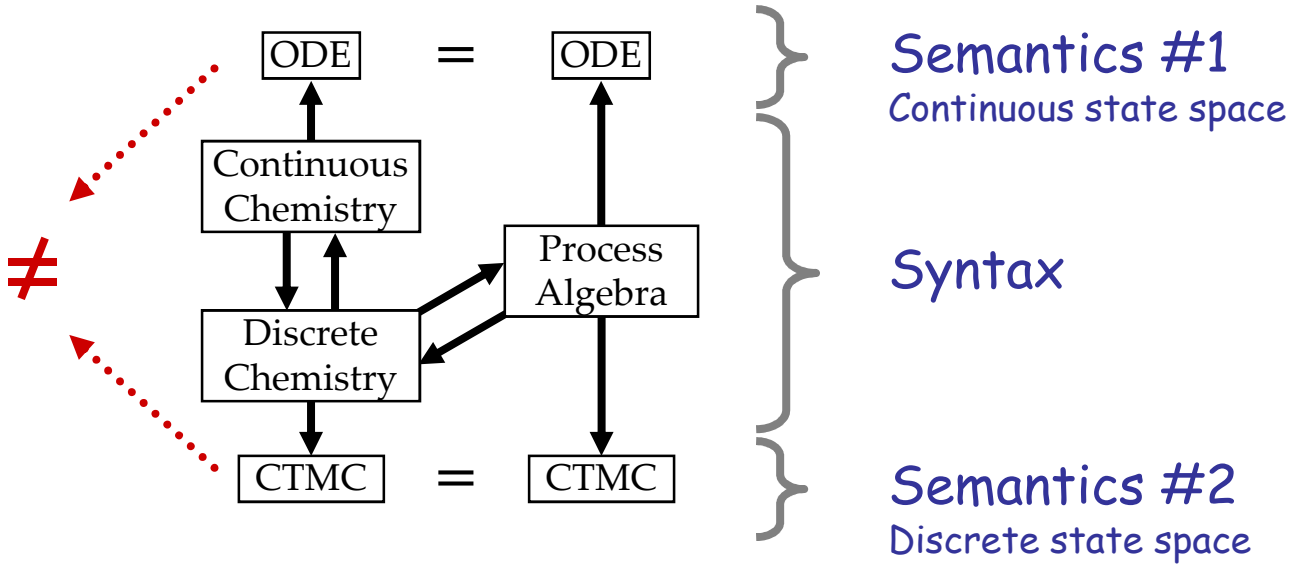
- Def: ≈ is equivalence of polynomials over the field of reals.
- Thm: E ≈ Cont(Ch(E))
- Thm: Cont(C) ≈ Pi(C)



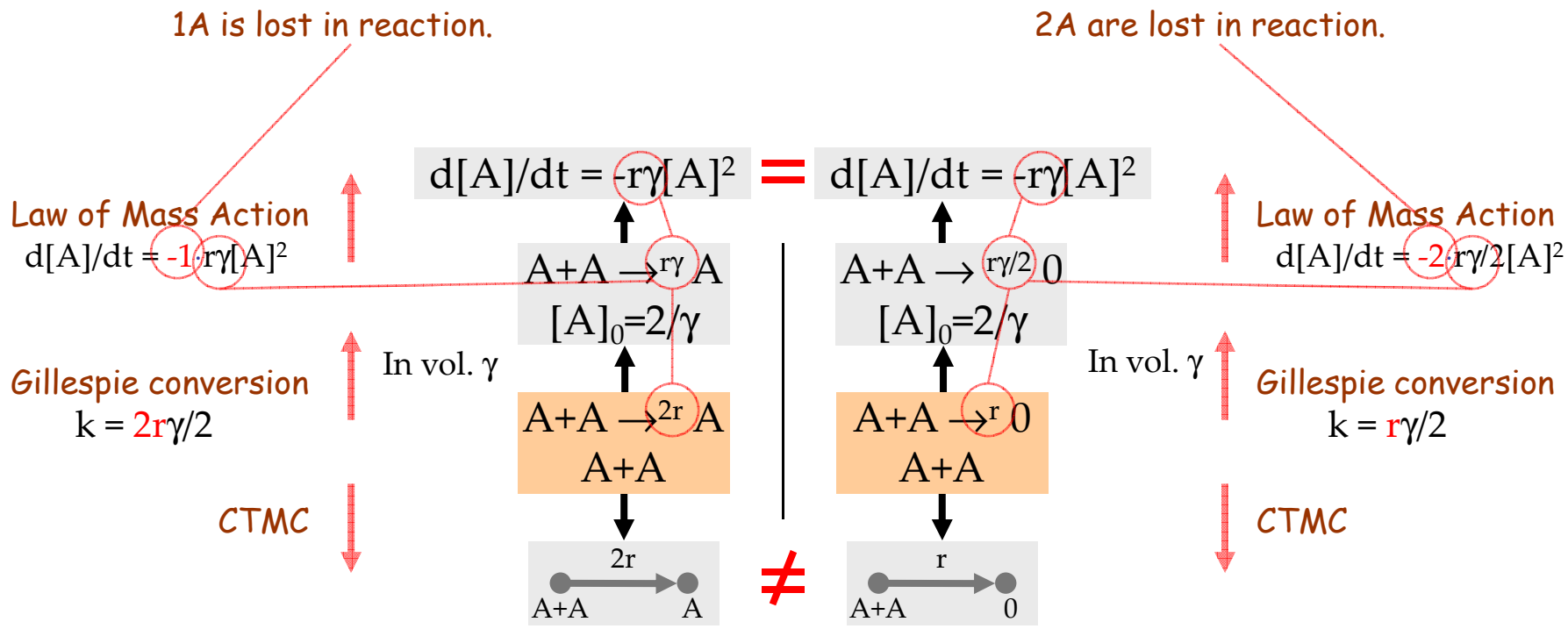


- For each E there is an E' ≈ E that is detangled (E' = Pi(Ch(E)))
- For each E in automata form there is an an E' ≈ E that is detangled and in automata form (E' = Detangle(E)).

GMA ≠ CME

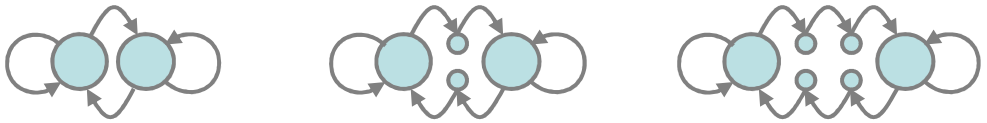

ODE
=
ODE
Continuous Chemistry
Process Algebra
Discrete Chemistry
≠
CTMC
=
CTMC
Semantics #1
Continuous state space
Syntax
Semantics #2
Discrete state space

$A+A \rightarrow^{2r} A$ =? $A+A \rightarrow^{r} 0$

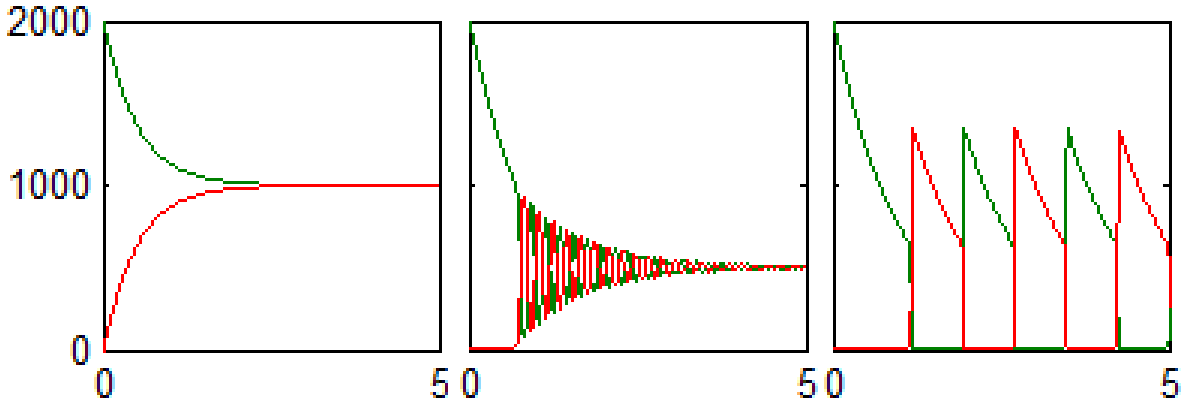


(For conservation of mass, consider instead $A+A \rightarrow^{2r} A+B$ vs. $A+A \rightarrow^{r} B+B$)

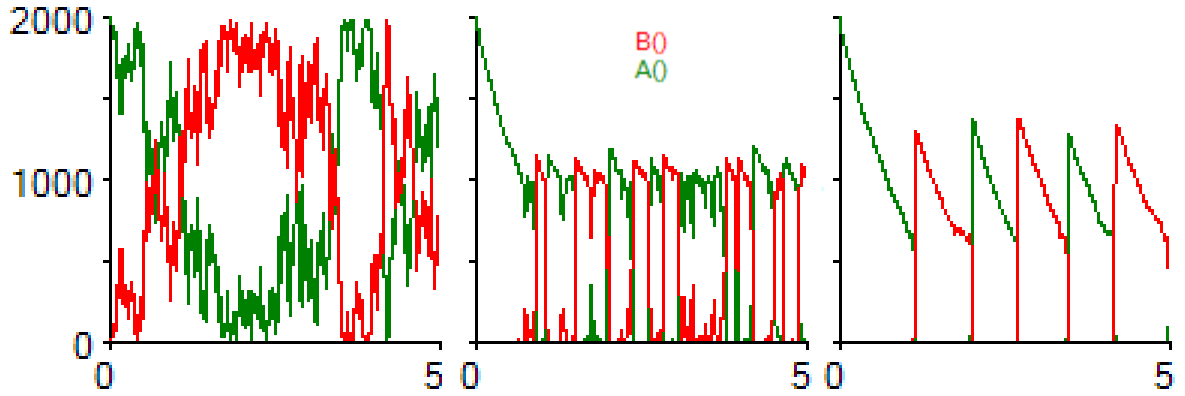
Continuous vs. Discrete Groupies



(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 B(); A()

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()

let Ad() = !a; Ad()
and Bd() = !b; Bd()

run 2000 of A()
run 1 of (Ad() | Bd())
```

```
directive sample 5.0 1000
directive plot B(); A()

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()

let Ad() = !a; Ad()
and Bd() = !b; Bd()

run 2000 of A()
run 1 of (Ad() | Bd())
```

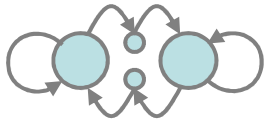
```
directive sample 5.0 1000
directive plot B(); A()

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

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

let Ad() = !a; Ad()
and Bd() = !b; Bd()

run 2000 of A()
run 1 of (Ad() | Bd())
```

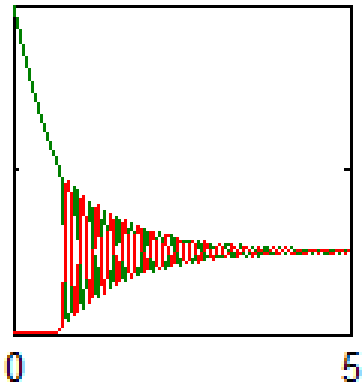
```
Groupe ODEs - Groupies.mat

[0:0.001:5.0] r=1.0 k=1.0
A dx1/dt = -(x1-x2),   2000.0
B dx2/dt = (x1-x2),    0.0
```

```
Groupe ODEs - Groupies Hysteric 1.mat

[0:0.001:5.0] r=1.0 k=1.0
A dx1/dt=x1*x4-x3*x1-x1+x4, 2000.0
A' dx2/dt=x3*x1-x3*x2+x1-x2, 0.0
A'' dx5/dt=x3*x2-x3*x5+x2-x5, 0.0
B dx3/dt=x3*x2-x1*x3-x3+x2, 0.0
B' dx4/dt=x1*x3-x1*x4+x3-x4, 0.0
```

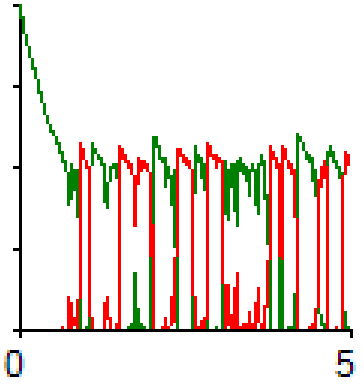
```
Groupe ODEs - Groupies Hysteric 2.mat

[0:0.001:5.0] r=1.0 k=1.0
A  dx1/dt=x1*x6-x3*x1-x1+x6,  2000.0
A'  dx2/dt=x3*x1-x3*x2+x1-x2, 0.0
A''  dx5/dt=x3*x2-x3*x5+x2-x5, 0.0
B  dx3/dt=x3*x5-x1*x3-x3+x5, 0.0
B'  dx4/dt=x1*x3-x1*x4+x3-x4, 0.0
B''  dx6/dt=x1*x4-x1*x6+x4-x6, 0.0
```

Scientific Predictions





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

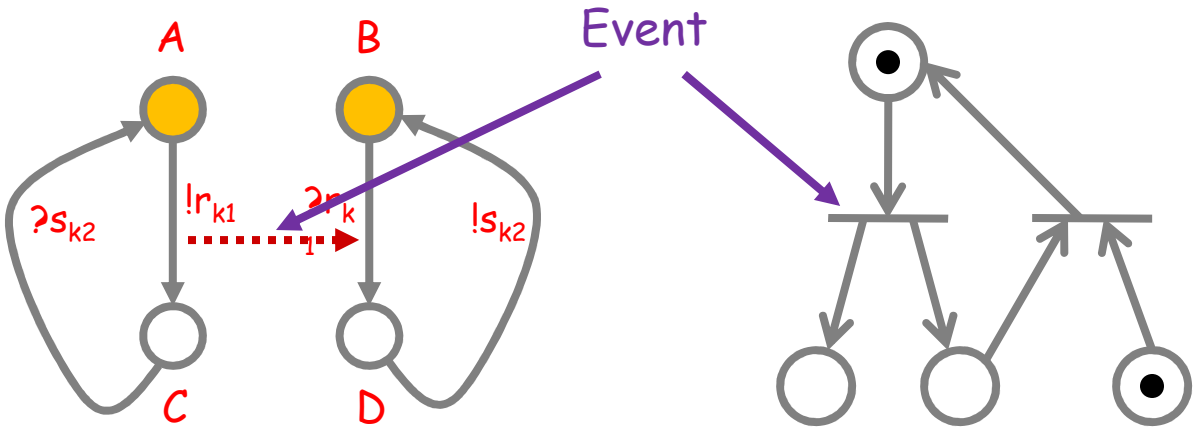


The 4 states are almost never equally occupied.

Discrete Analysis Techniques

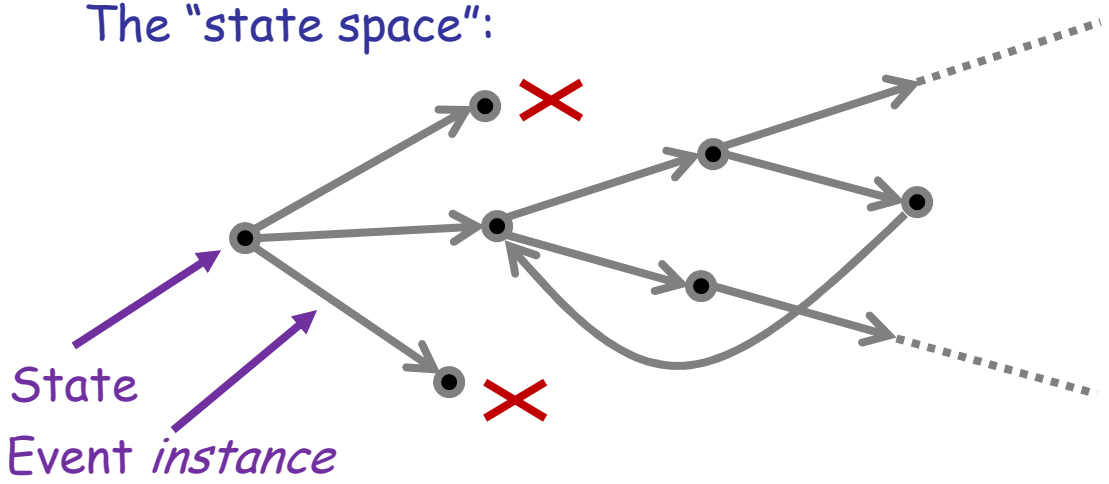
The Program vs. the State Space

The "program":



Finite

The "state space":

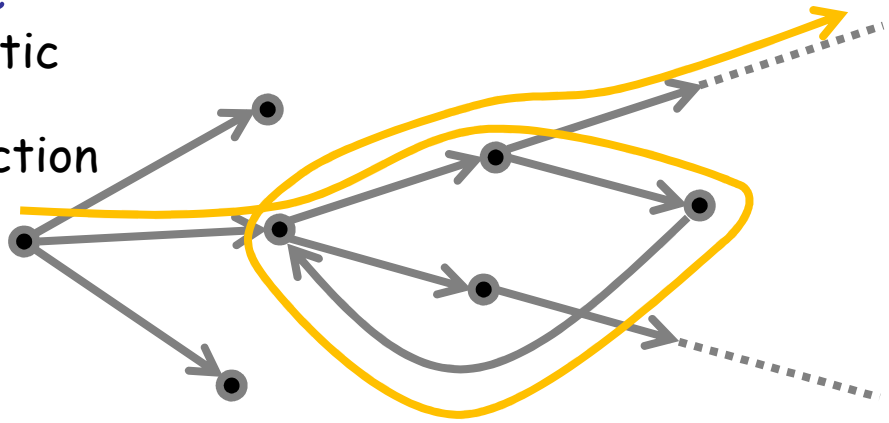


Potentially infinite

Simulation

- Run "the program" through a walk in states space.

- Basic stochastic algorithm: Gillespie
  - Exact (i.e. based on physics) stochastic simulation of chemical kinetics.
  - Can compute concentrations and reaction times for biochemical networks.



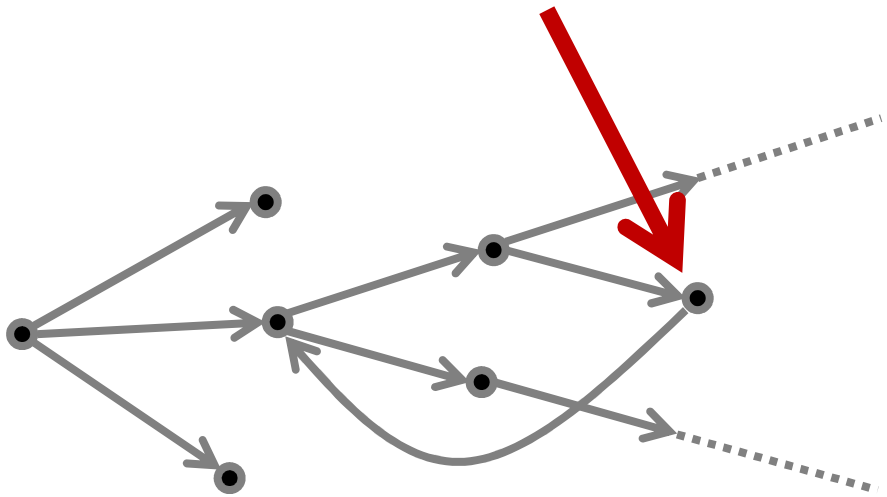
- Stochastic Process Algebras
  - Now many [BioSPi, SPiM, BioPEPA, BetaBinders, …]

- Hybrid approaches
  - Continuous + discrete/stochastic switching

Control Flow Analysis

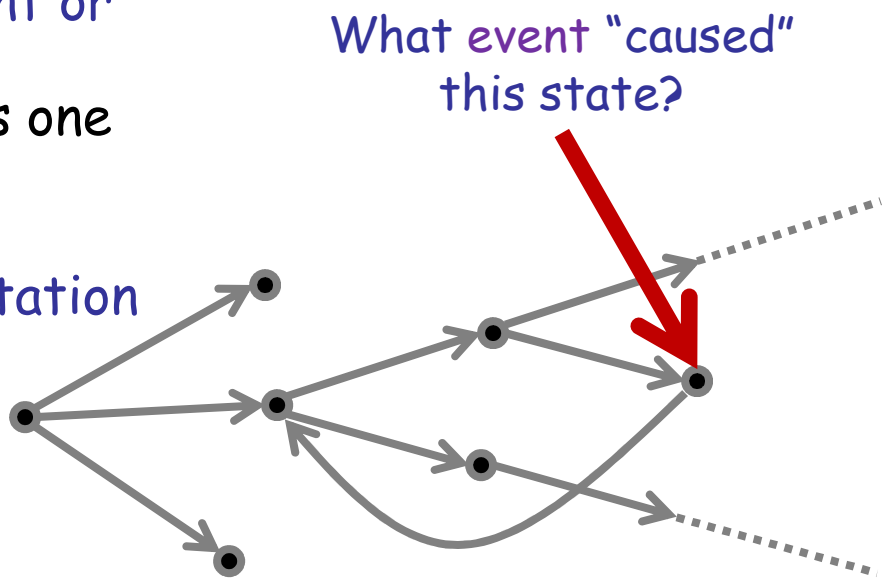
- Who may call who?
  - Overapproximation of behavior used to answer questions about what "cannot happen".

What event may (or may not) have been involved in reaching this state?

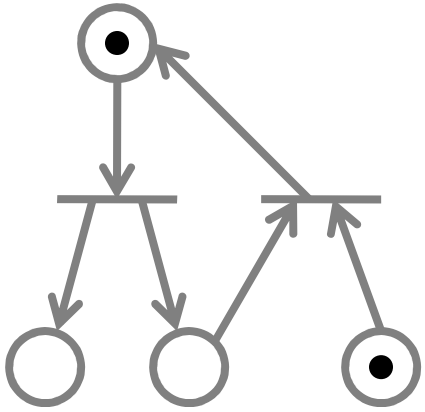


Causality Analysis

- What event caused what other event or state to happen?
  - E.g.: if in all possible executions one event always precedes another.

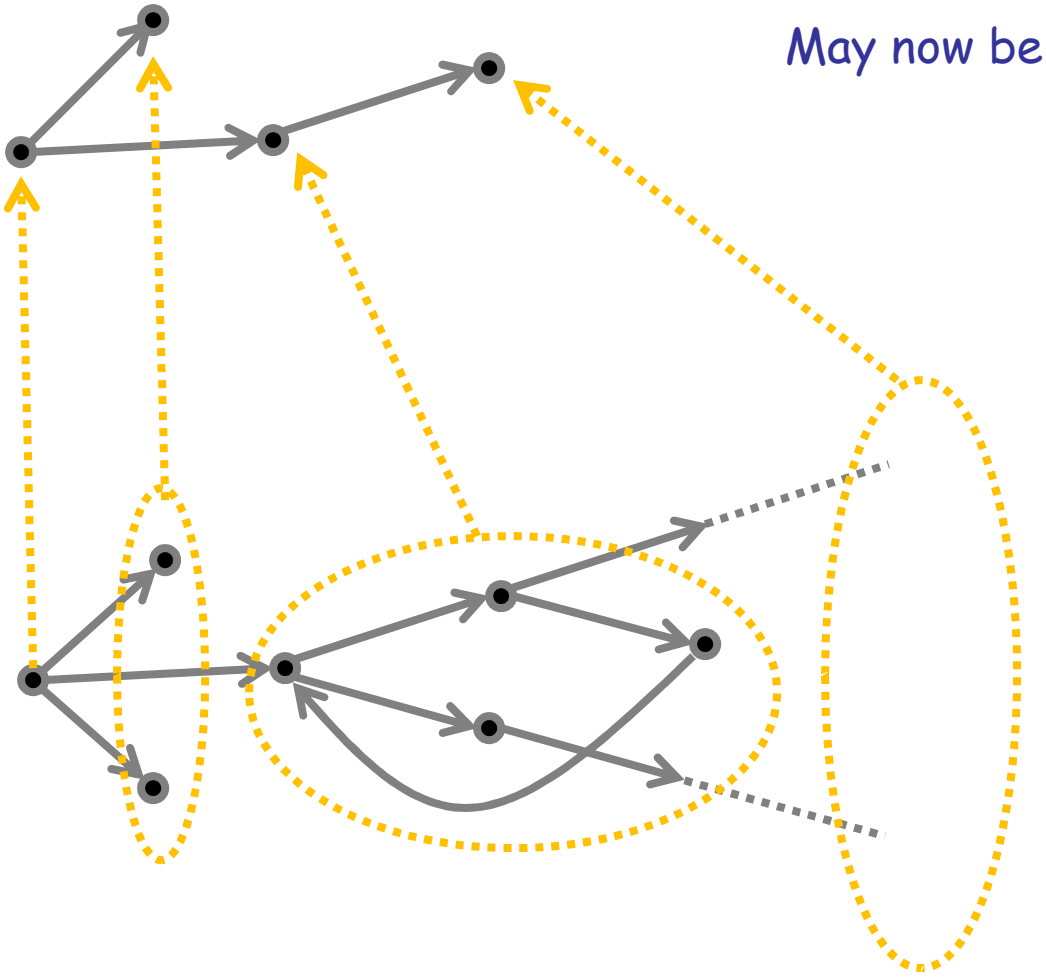


- Need a different level of representation (the "event space")
  - Petri Nets
  - Event Structures



Abstract Interpretation

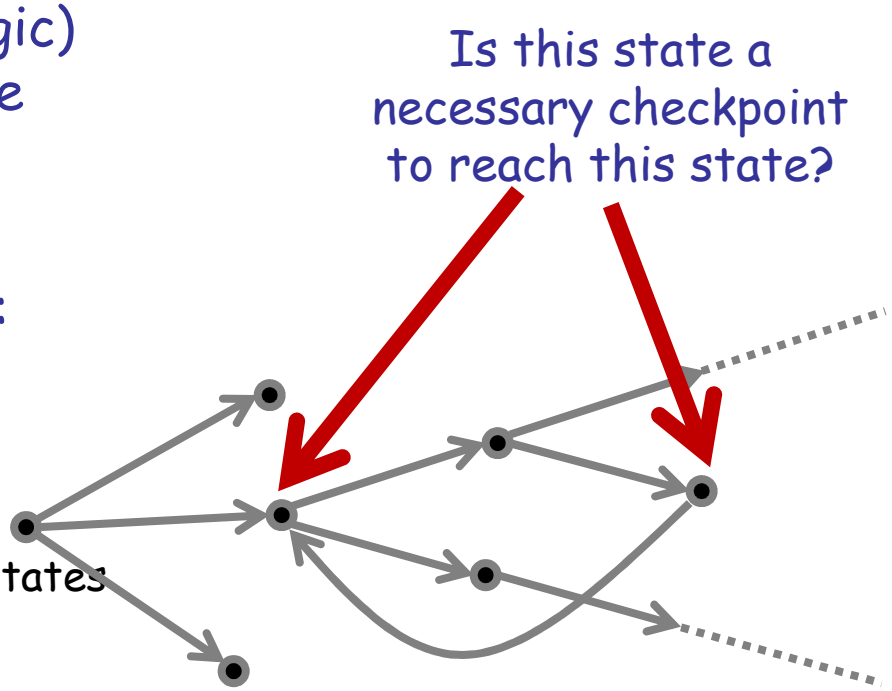
- Precisely relating abstract views to more concrete views of the system



Modelchecking

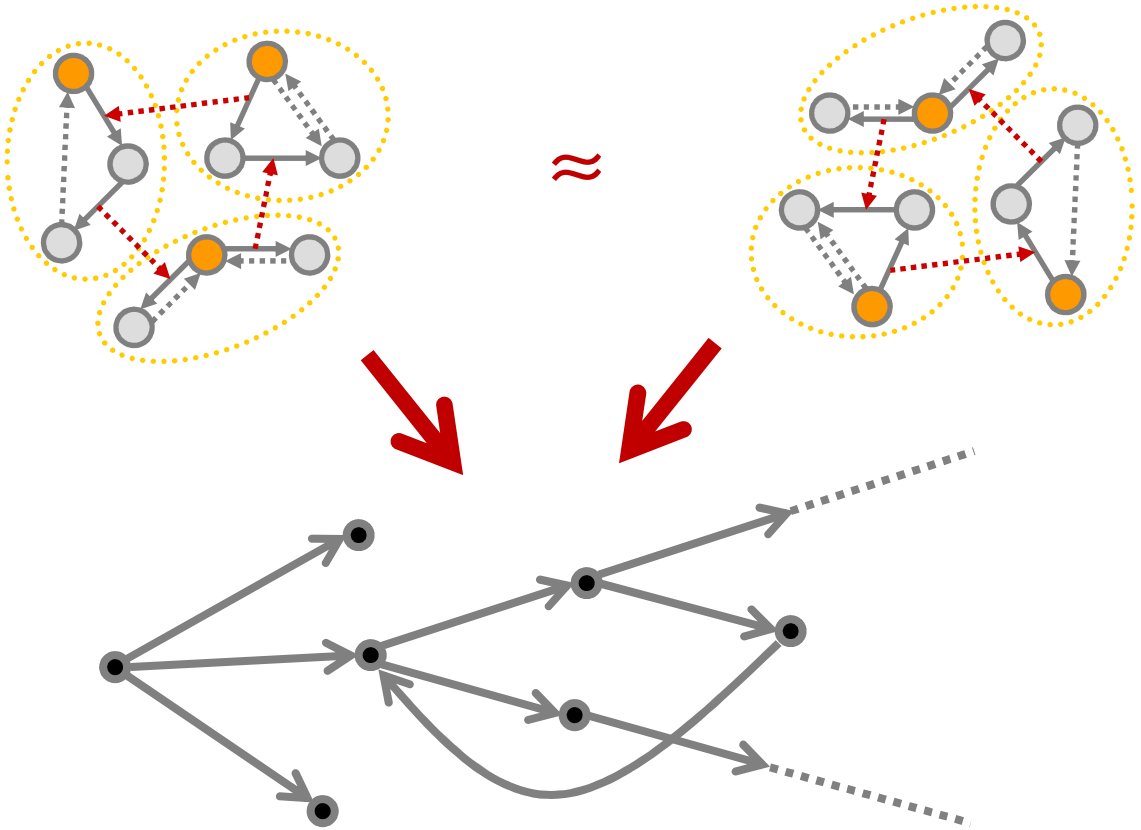
- Asking questions (in Temporal Logic) about structure of a (finite) state space.

- Various flavors of modelchecking:
  - Temporal
    - About paths through state space
  - Quantitative
    - About quantitative measures of states
  - Probabilistyc/Stochastic
    - About probabilities of reaching states.



Bisimulation

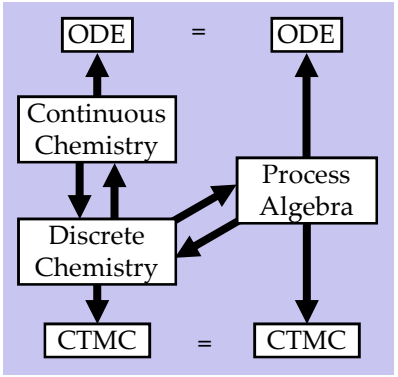
- Are two programs generating the same state space?
  - E.g.: Is a compact description of a system equivalent to a more detailed one in all possible environments?



Conclusions

Conclusions

- Process Algebra
  - An extension of automata theory to populations of interacting automata
  - Modeling the behavior of individuals in an arbitrary environment
  - Compositionality (combining models by juxtaposition)



- 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)

- A bright future for Computer Science and Logic in modern Biology
  - Biology needs good analysis techniques for discrete systems analysis (modal logics, modelchecking, causality analysis, abstract interpretation, …)