Molecules as Automata

Representing Biochemical Systems as Collectives of Interacting Automata

Luca Cardelli

Microsoft Research

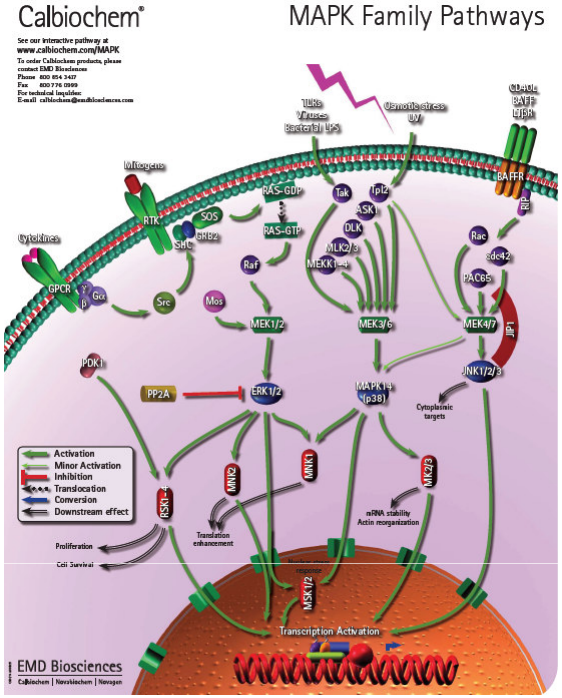
Open Lectures for PhD Students in Computer Science
Warsaw 2009-03-12..13

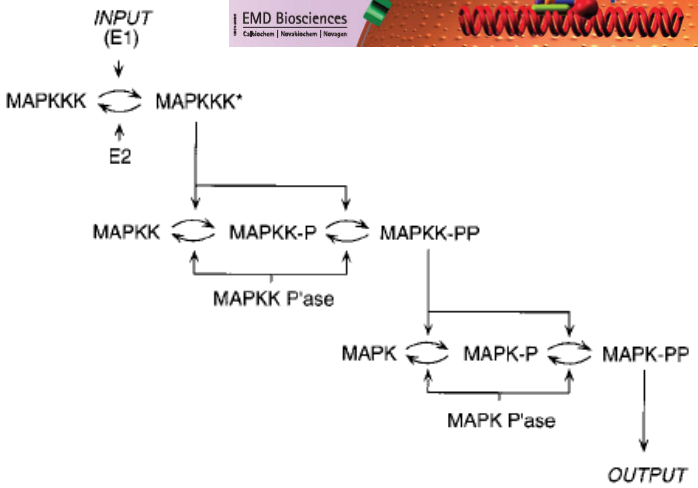
http://lucacardelli.name

Macro-Molecules as Interacting Automata

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?

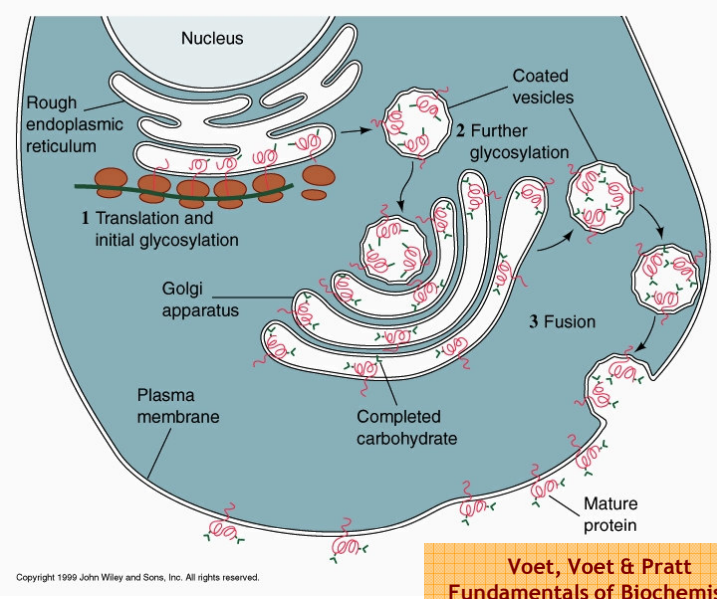




Ultrasensitivity in the mitogen-activated protein cascade, Chi-Ying F. Huang and James E. Ferrell, Jr., 1996, *Proc. Natl. Acad. Sci. USA*, 93, 10078-10083.

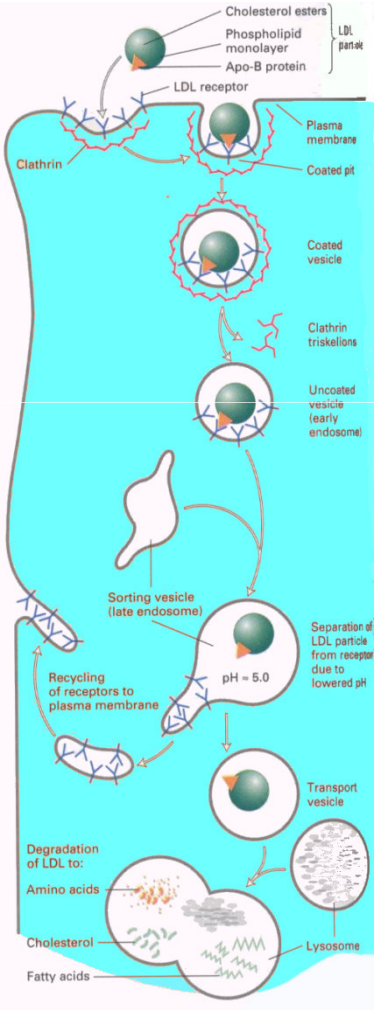
Biological "Algorithms"

Protein Production and Secretion



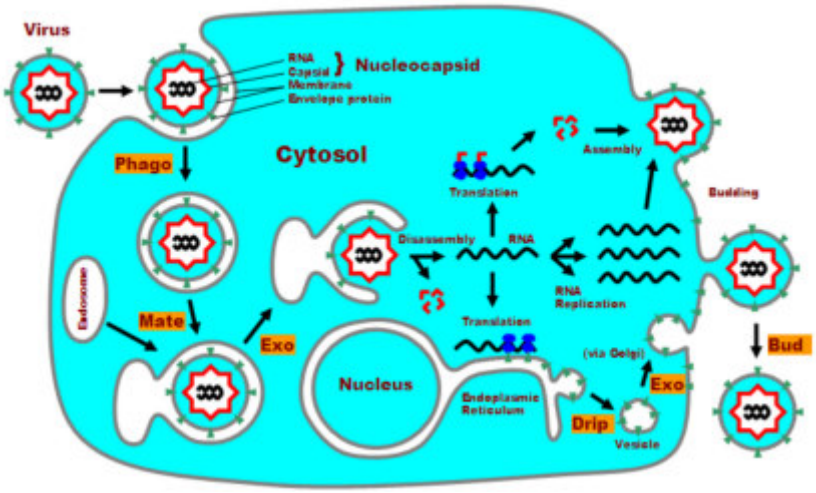
Voet, Voet & Pratt
Fundamentals of Biochemistry
Wiley 1999. Ch10 Fig 10-22.

LDL-Cholesterol Degradation



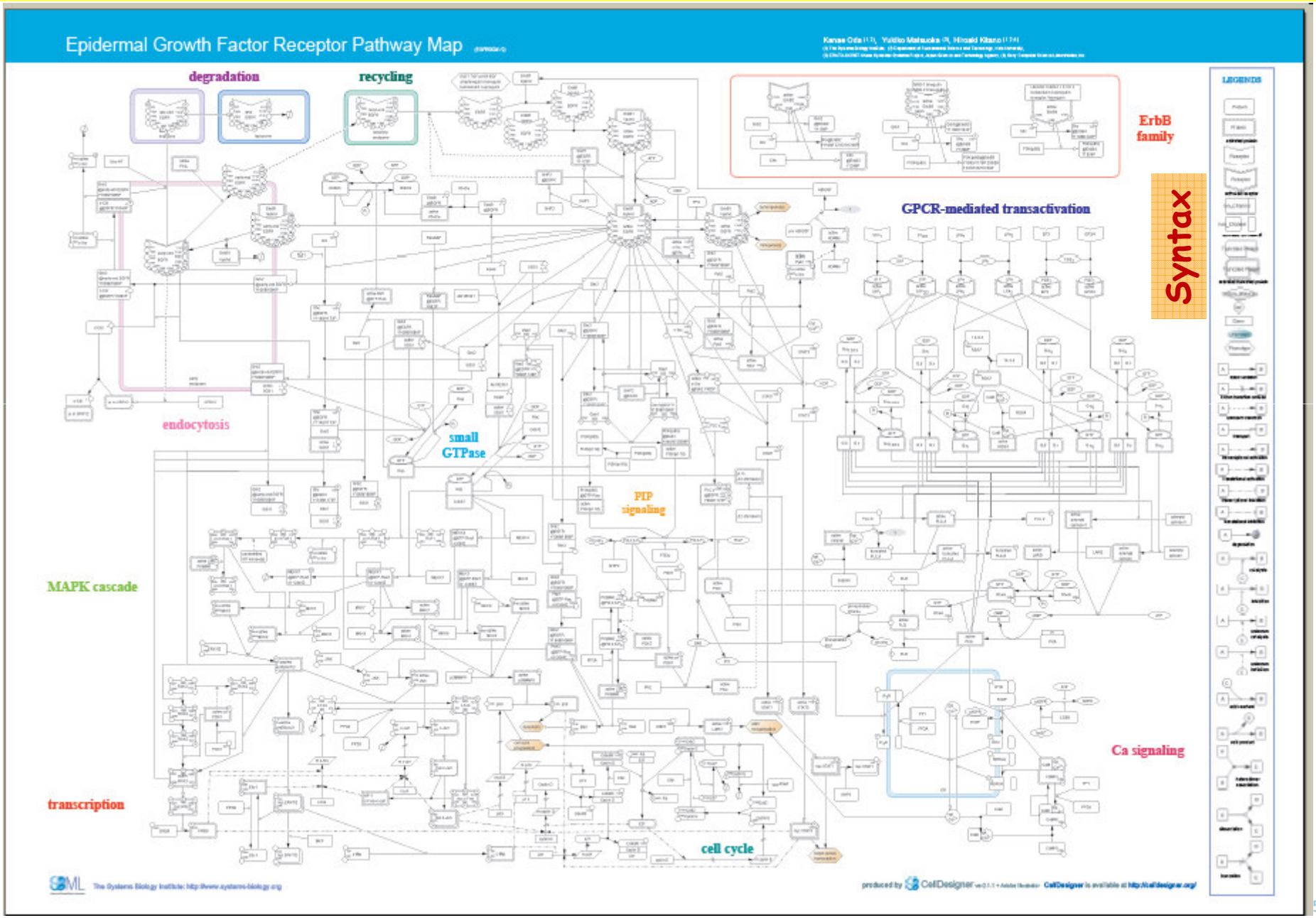
H.Lodish et al.
Molecular Cell Biology.
fourth Edition p.730.

Viral Replication

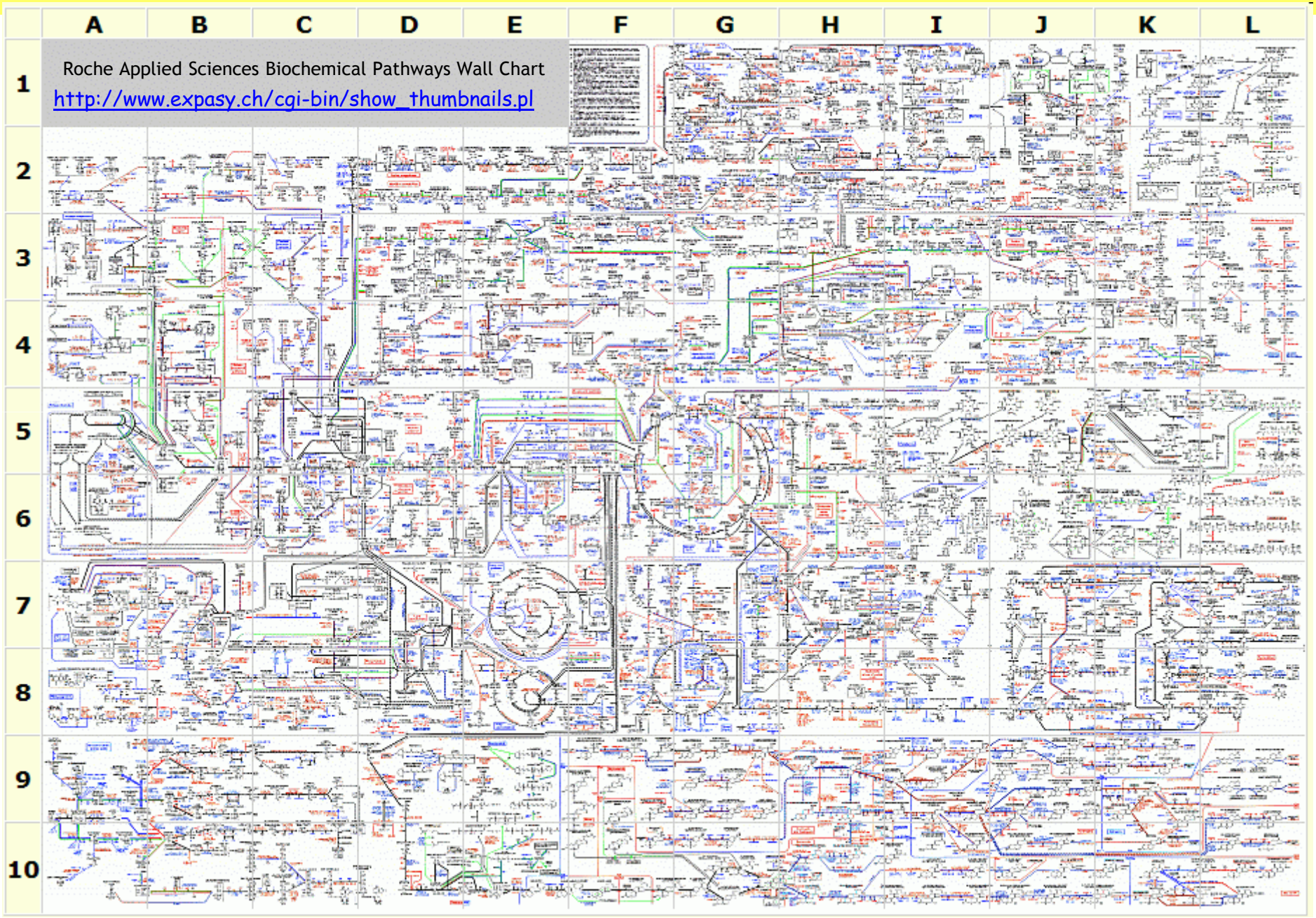


Adapted from: B.Alberts et al.
Molecular Biology of the Cell
third edition p.279.

Discrete State Transitions



Compositionality (NOT!)

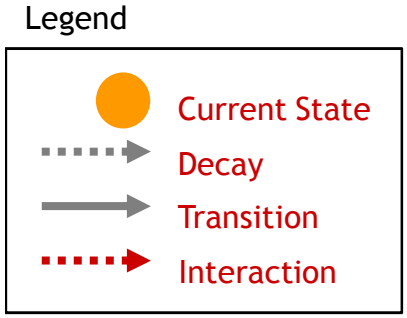


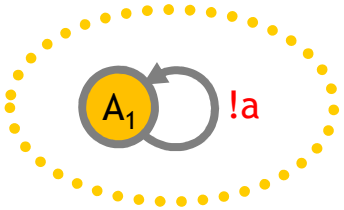
Process Algebra

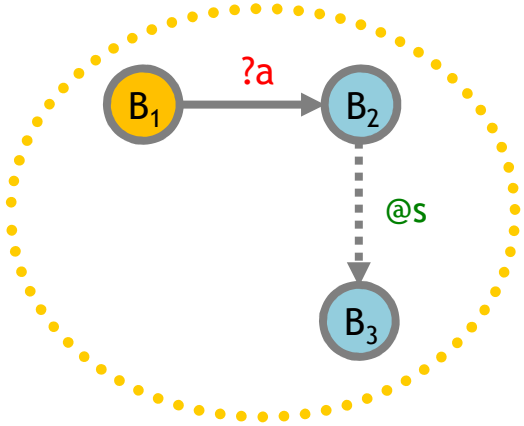
[Hoare, Milner, Pnueli, etc.]

- 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





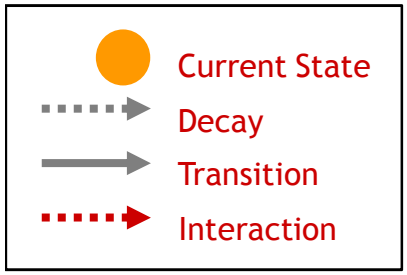


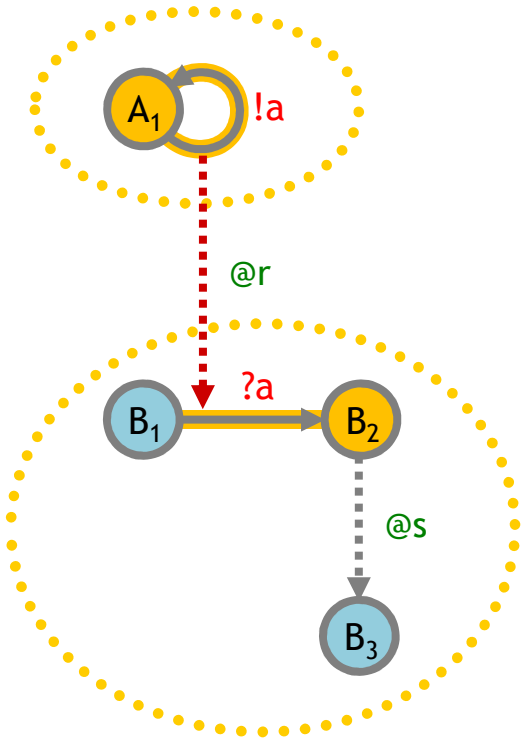
A_1	is a *state*
a	is a *channel* i.e. a named *interaction interface* (e.g. a surface patch)
?,!	indicate any *complementarity* of interaction (e.g. charge)
?a, !a	indicate *complementary actions*,
@r, @s	are rates

Kinetic laws:

Interacting Automata

Legend



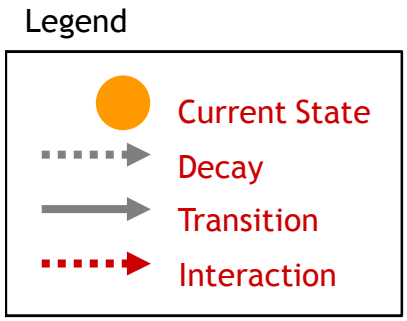


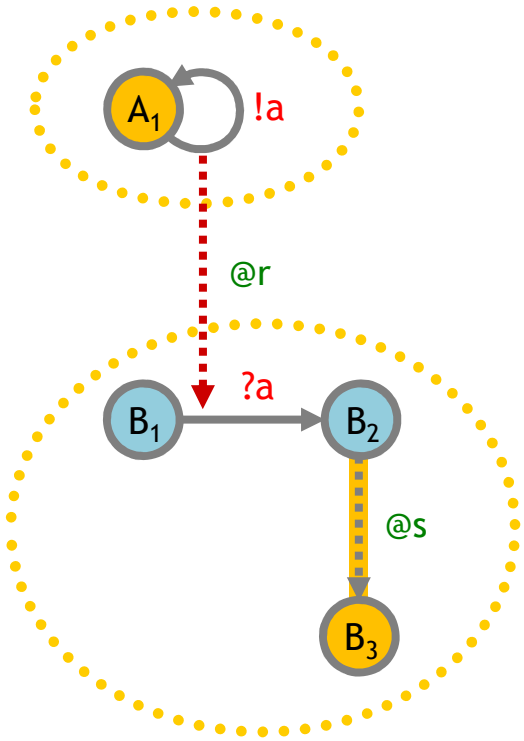
A_1	is a *state*
a	is a *channel* i.e. a named *interaction interface* (e.g. a surface patch)
?,!	indicate any *complementarity* of interaction (e.g. charge, shape)
?a, !a	indicate *complementary actions*, joined by an interaction arrow ⋯▸
@r, @s	are rates

Kinetic laws:

Two complementary actions may result in an interaction.

Interacting Automata





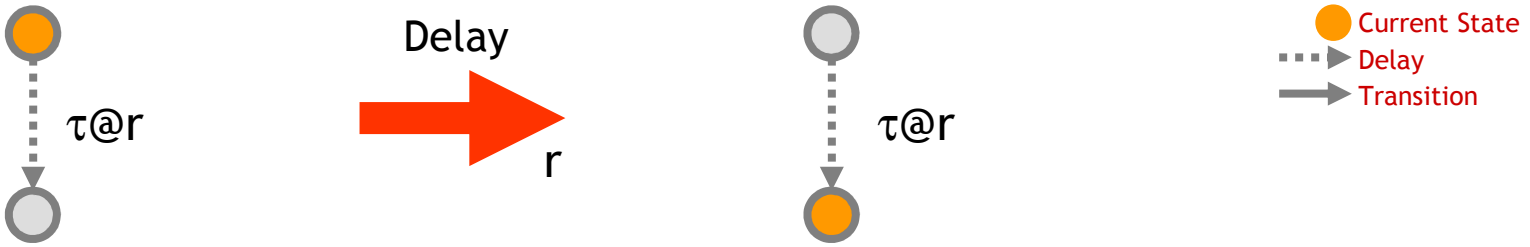
A_1	is a *state*
a	is a *channel* i.e. a named *interaction interface* (e.g. a surface patch)
?,!	indicate any *complementarity* of interaction (e.g. charge)
?a, !a	indicate *complementary actions*, joined by an interaction arrow ⋯▸
@r, @s	are rates

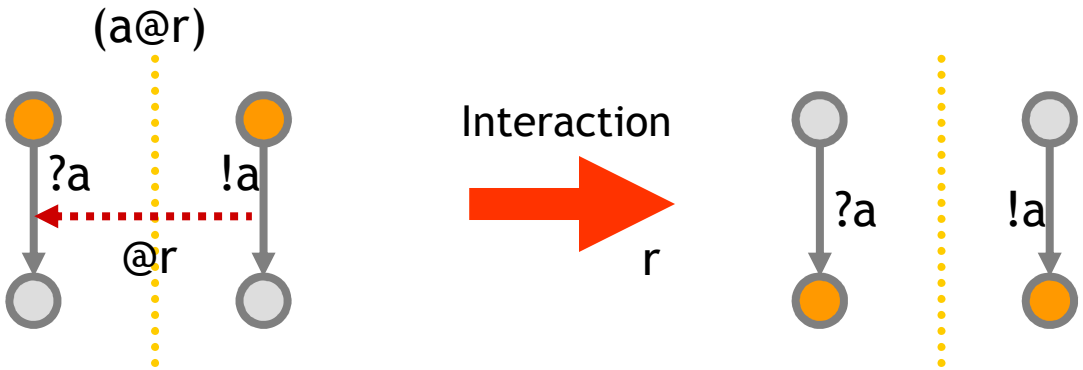
Kinetic laws:

Two complementary actions may result in an interaction.

A decay may happen spontaneously.

Interacting Automata Transition Rules



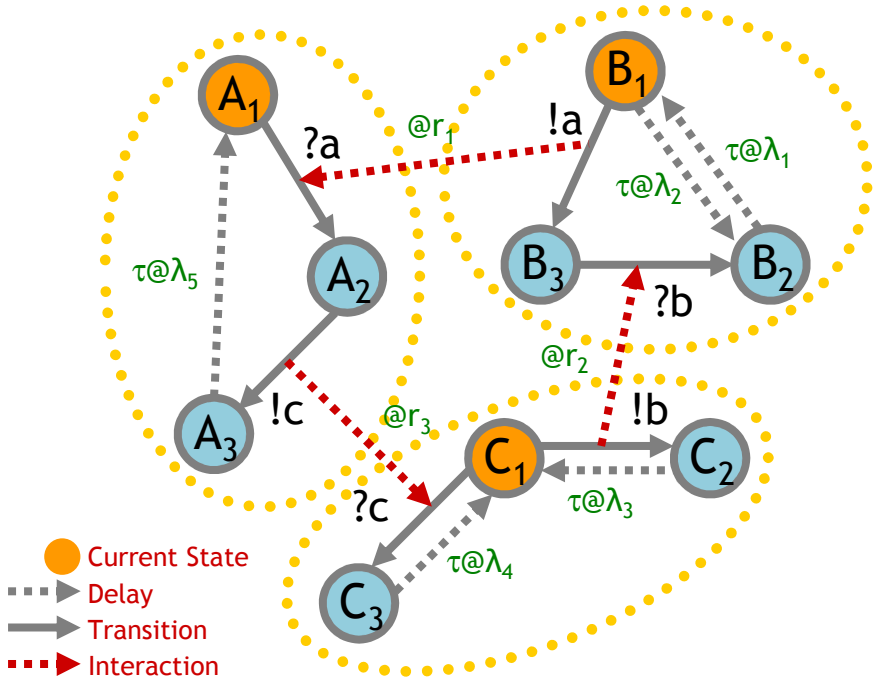


Interactions have rates. Actions DO NOT have rates.

Q: What kind of mass behavior can this produce?

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

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

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

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

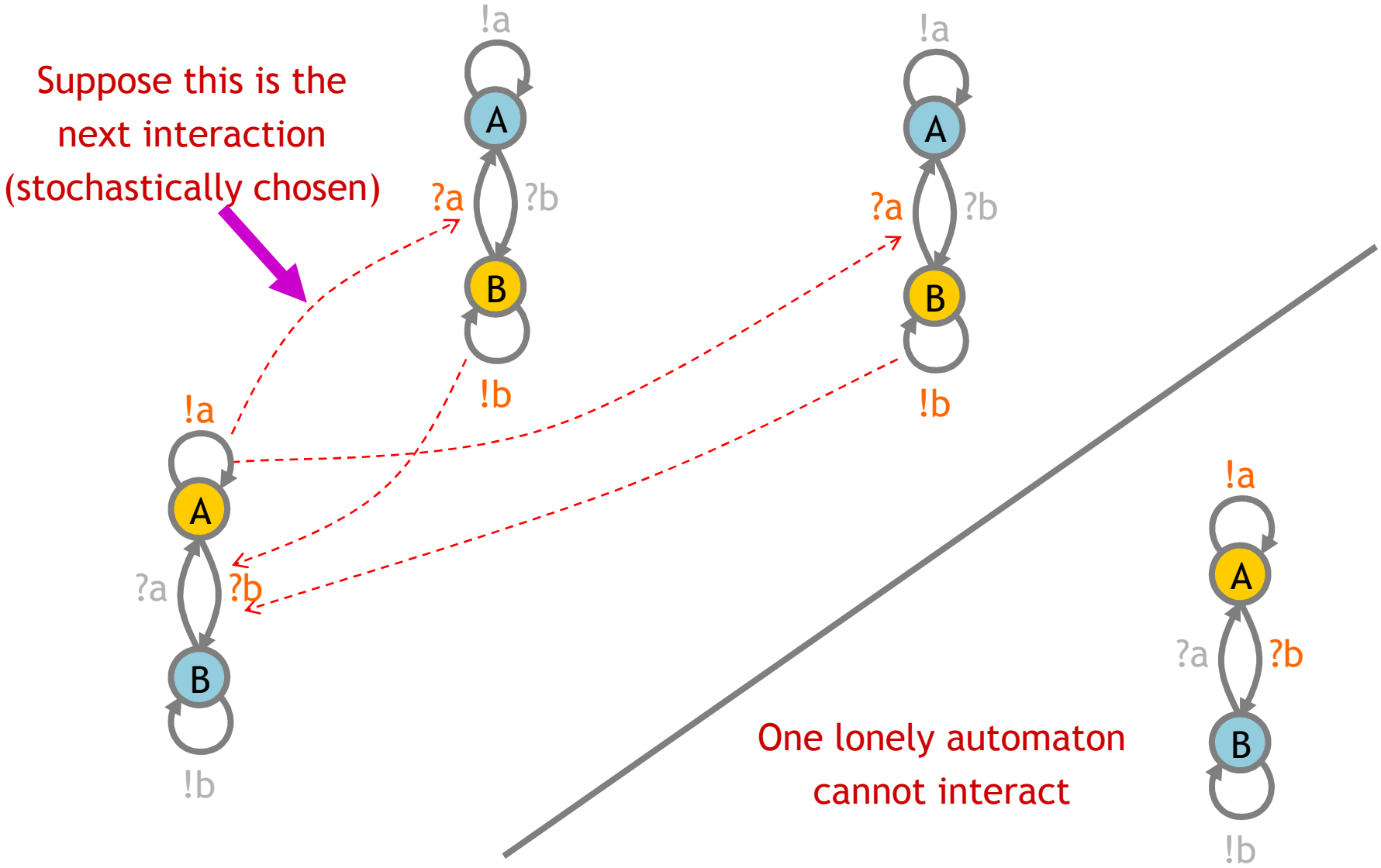
$C_1 = !b; C_2 + ?c; C_3$
$C_2 = \tau@\lambda_3; C_1$
$C_3 = \tau@\lambda_4; C_2$

Automata

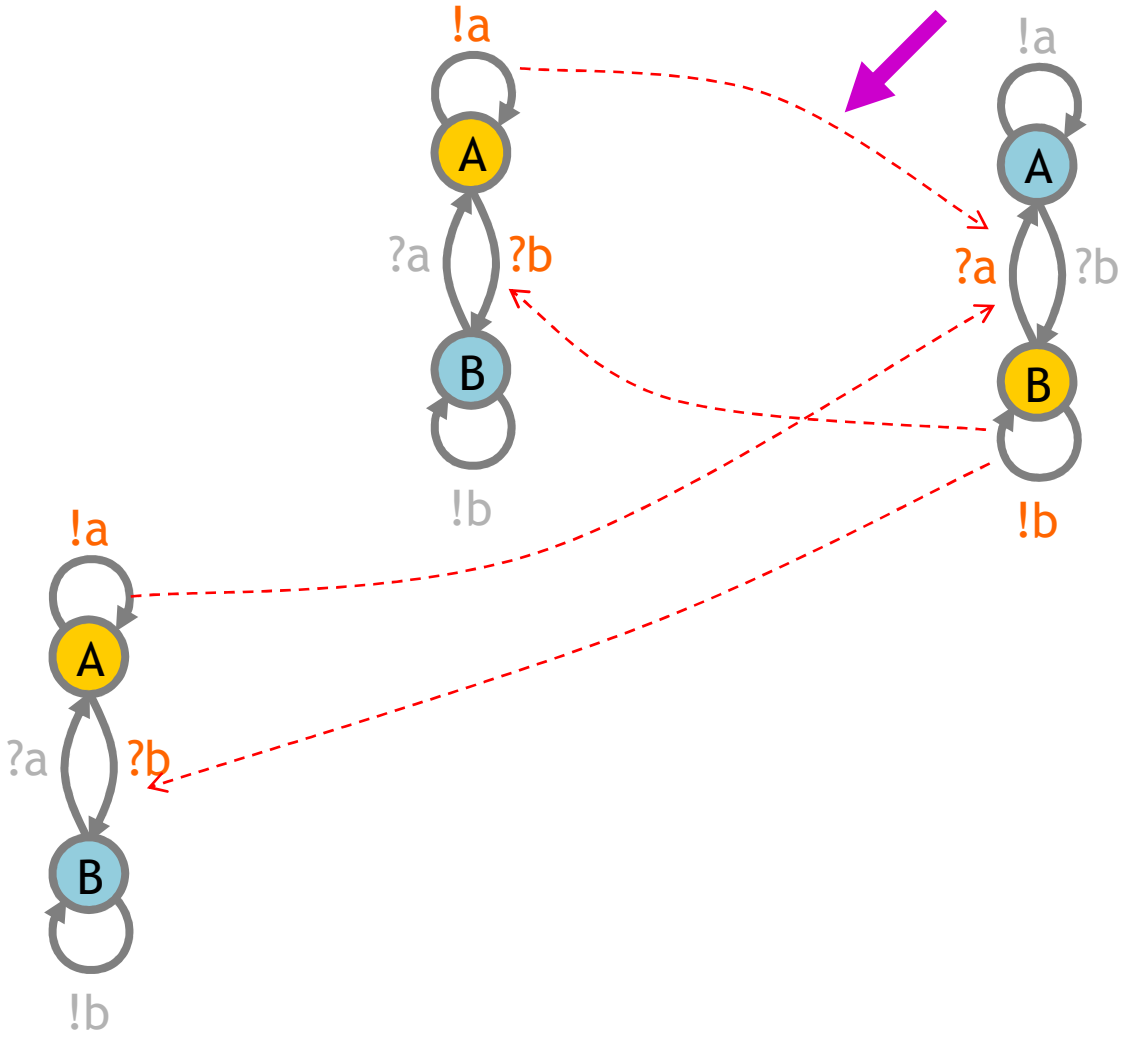
$A_1 \mid B_1 \mid C_1$

The system and initial state

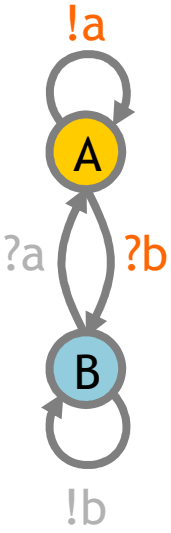
Interactions in a Population

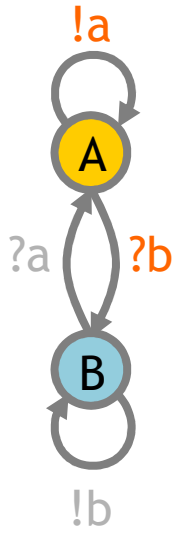


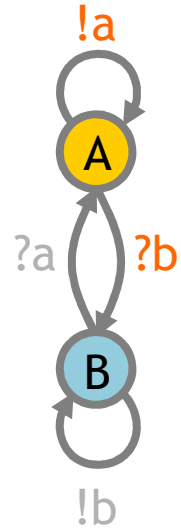
Interactions in a Population



Interactions in a Population

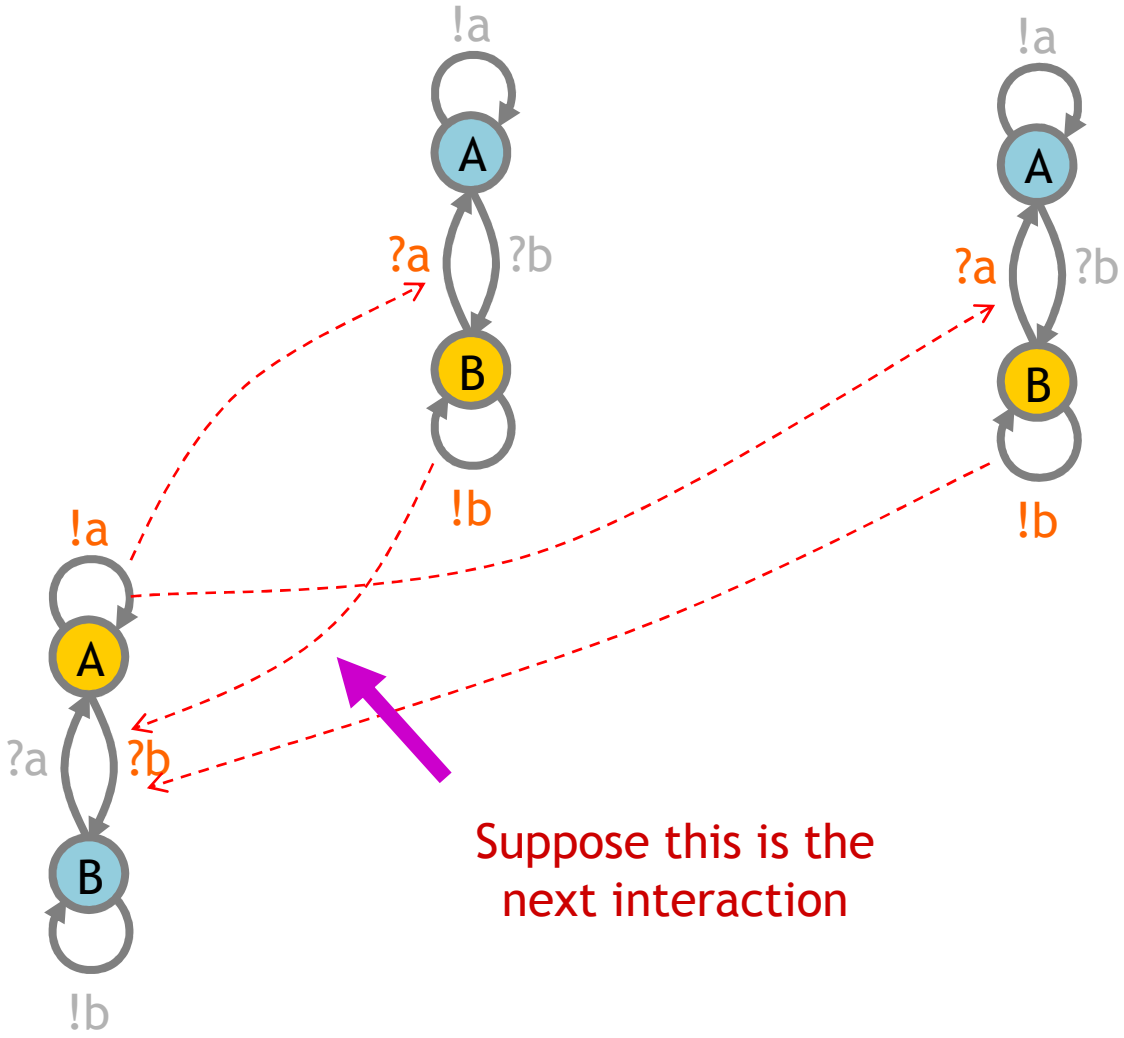




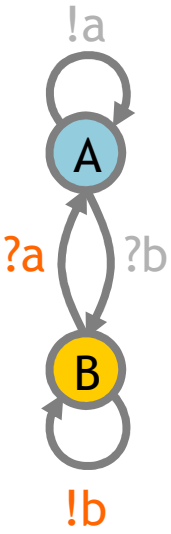


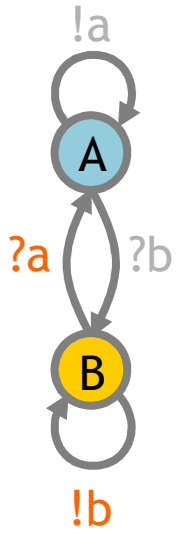
All-A stable population

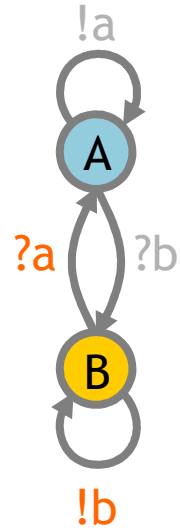
Interactions in a Population (2)



Interactions in a Population (2)



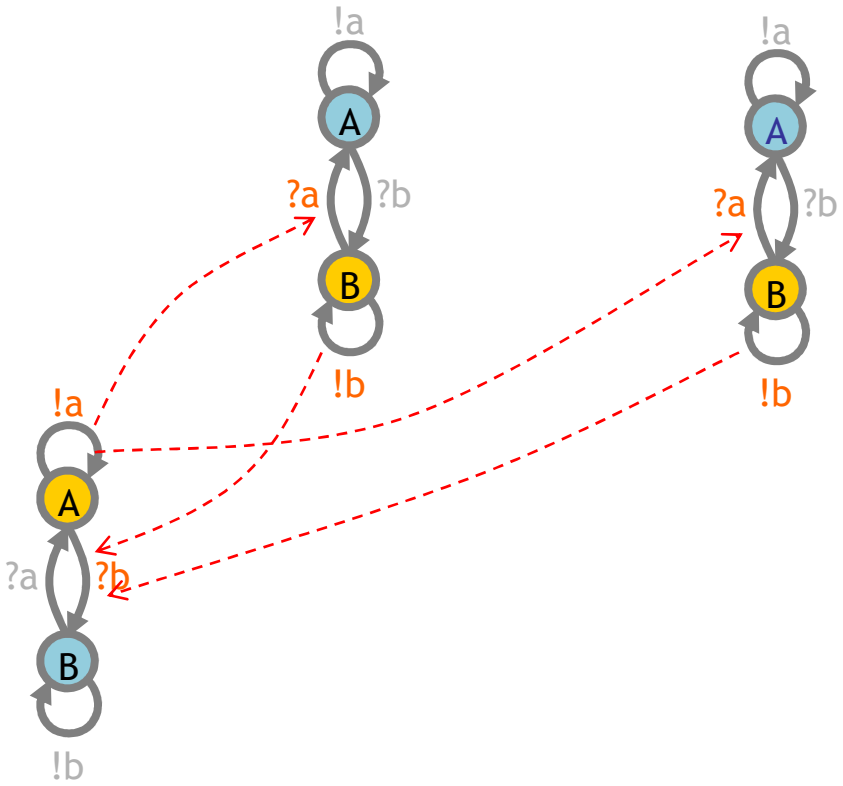


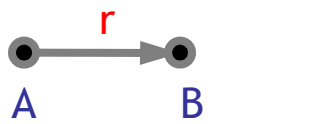


All-B stable population

Nondeterministic population behavior ("multistability")

CTMC Semantics



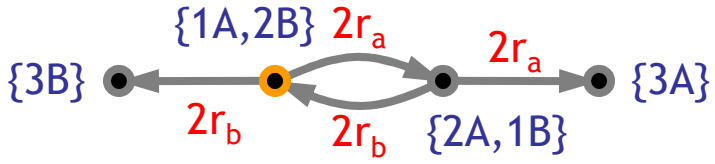


CTMC
(homogeneous) Continuous Time Markov Chain
- directed graph with no self loops
- nodes are system states
- arcs have transition rates

Probability of holding in state A:

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

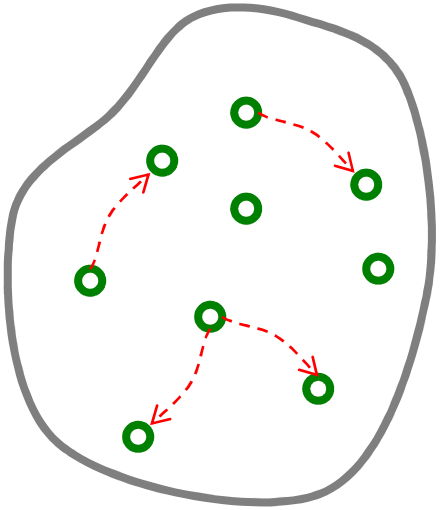
in general, $Pr(H_A>t) = e^{-Rt}$ where R is the sum of all the exit rates from A



CTMC

Stochastic Collectives

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



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

- Very much like biochemistry
  - Which is a large set of stochastically interacting molecules/proteins
  - Are proteins finite state and subject to automata-like transitions?
    - Let's say they are, at least because:
    - Much of the knowledge being accumulated in Systems Biology is described as state transition diagrams [Kitano].

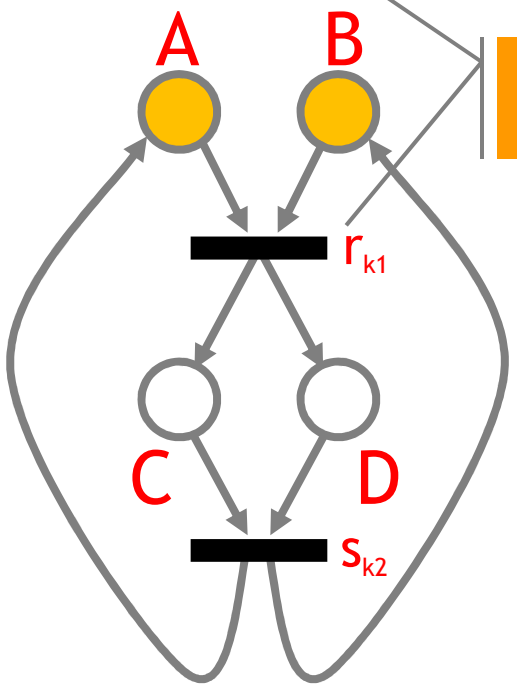
Chemistry vs. Automata

Says what "A" *does*.

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

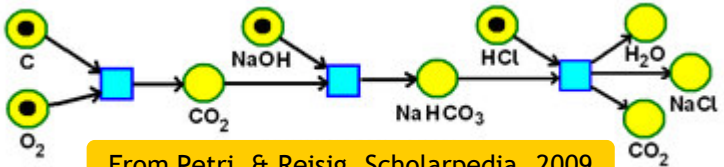
$s: C + D \rightarrow_{k2} A + B$

Does A become C or D?



Reaction oriented

1 line per reaction



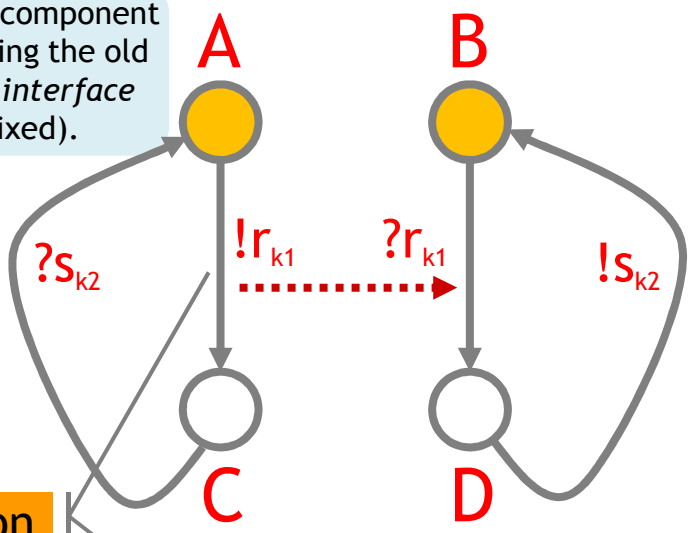
From Petri & Reisig, Scholarpedia, 2009

The same "state space"

CTMC

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

$A = !r_{k1}; C$

$C = ?s_{k2}; A$

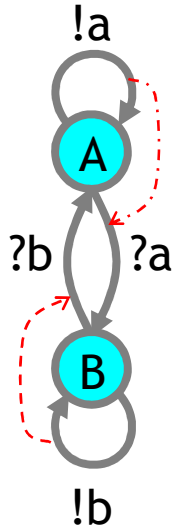
$B = ?r_{k1}; D$

$D = !s_{k2}; B$

A becomes C not D!

Groupies and Celebrities

Groupies and Celebrities



Celebrity

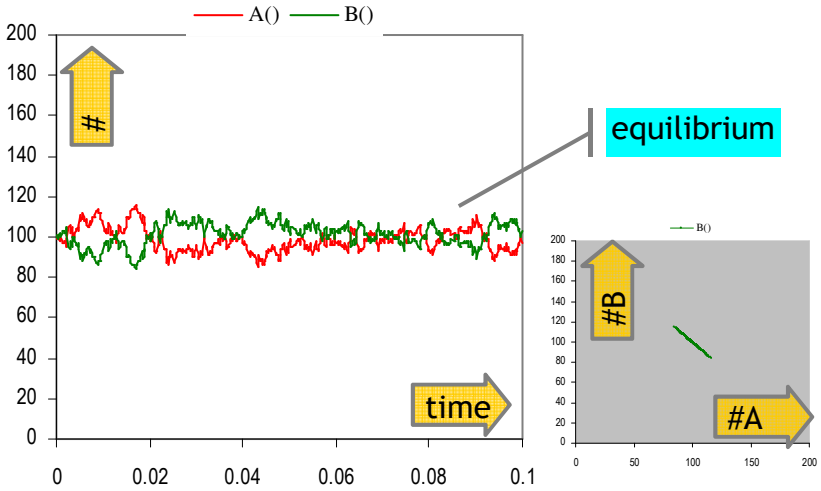
(does not want to be like somebody else)

```
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())
```

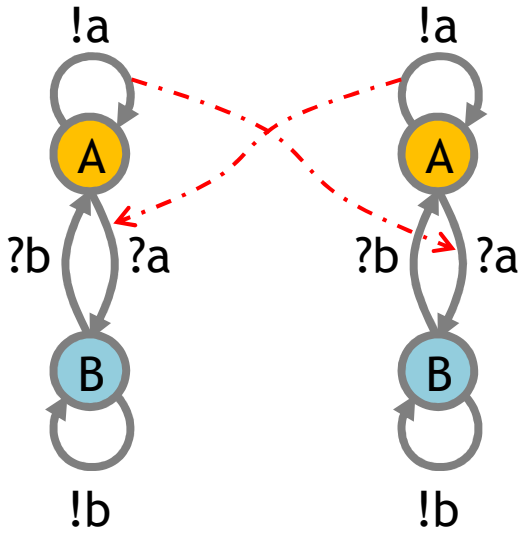
a@1.0

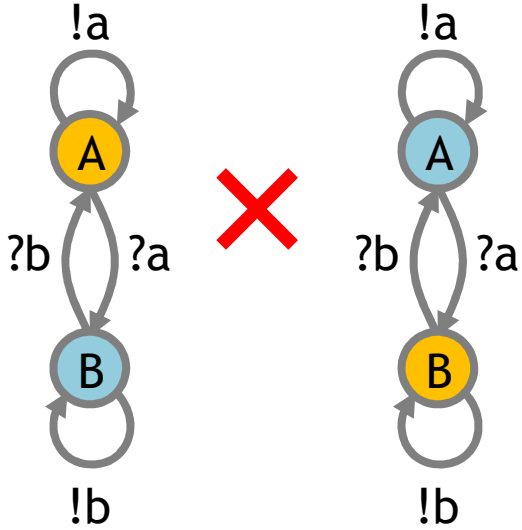
b@1.0

A stochastic collective of celebrities:

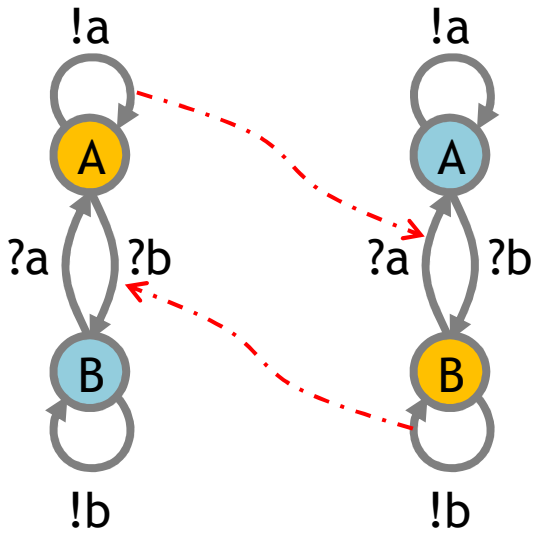


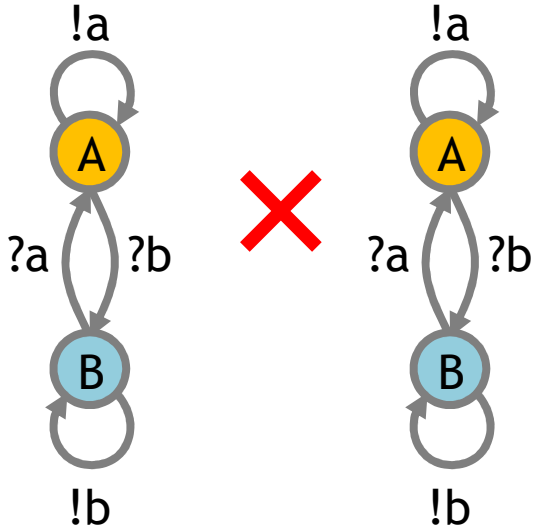
Stable because as soon as a A finds itself in the majority, it is more likely to find somebody in the same state, and hence change, so the majority is weakened.

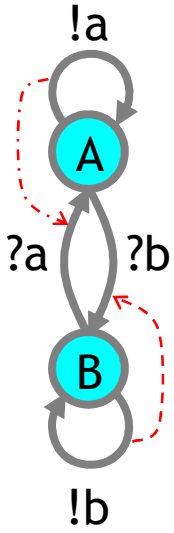




Groupies and Celebrities







Groupie

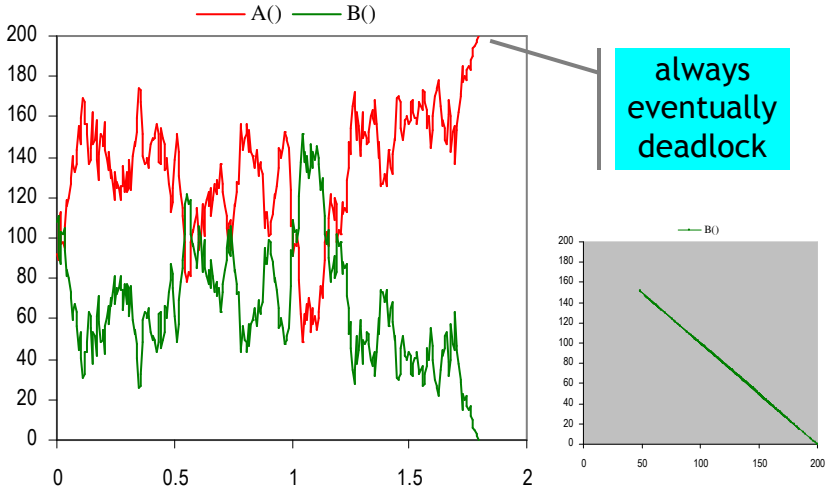
(wants to be like somebody different)

```
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:

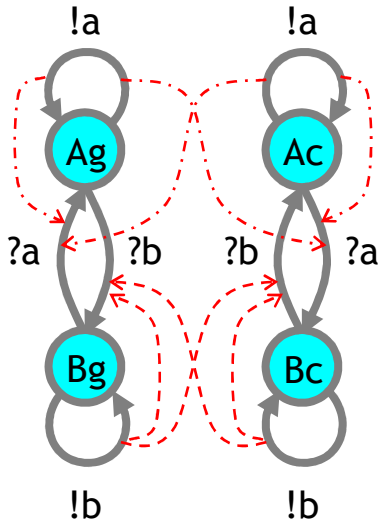


Unstable because within an A majority, an A has difficulty finding a B to emulate, but the few B's have plenty of A's to emulate, so the majority may switch to B. Leads to deadlock when everybody is in the same state and there is nobody different to emulate.

Both Together

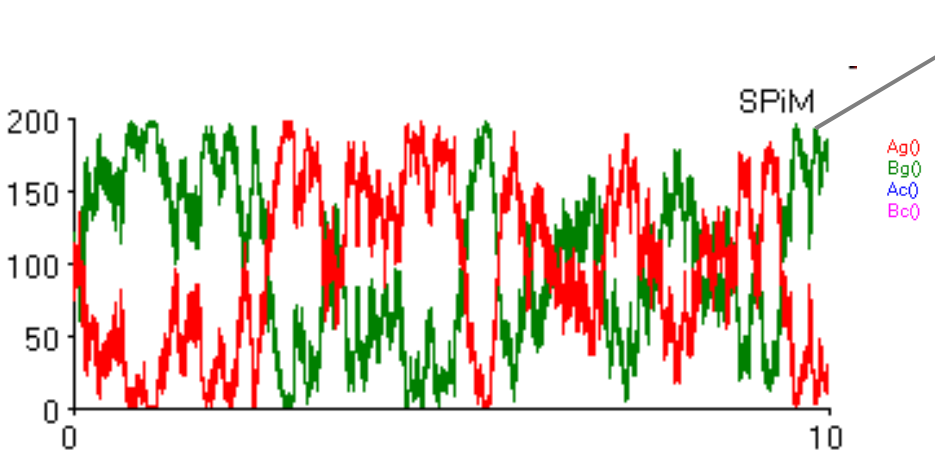
A way to break the deadlocks: Groupies with just a few Celebrities

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())
```

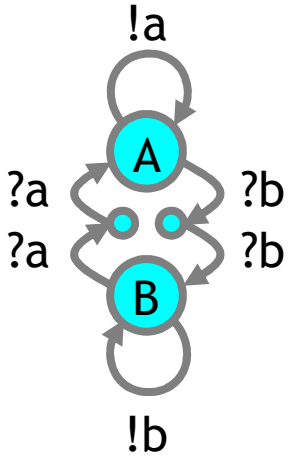


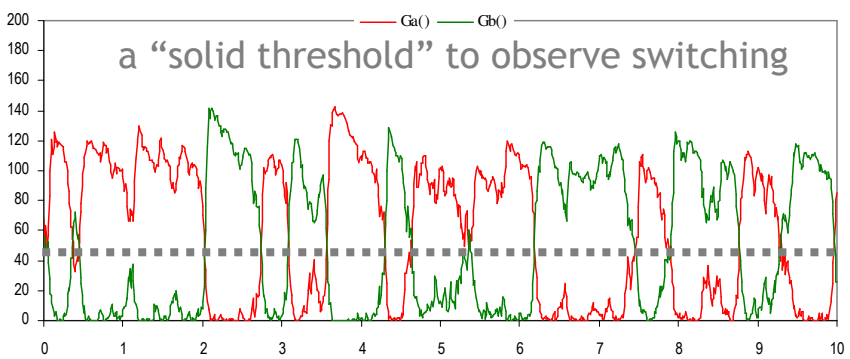
never deadlock

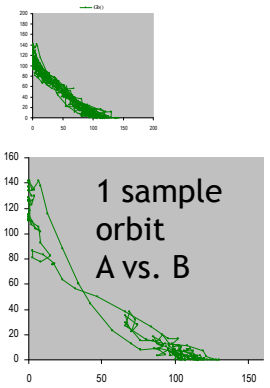
A tiny bit of "noise" can make a huge difference

Hysteric Groupies

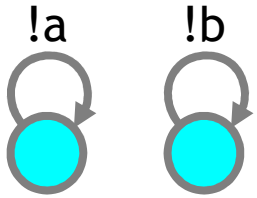
We can get more regular behavior from groupies if they "need more convincing", or "hysteresis" (history-dependence), to switch states.







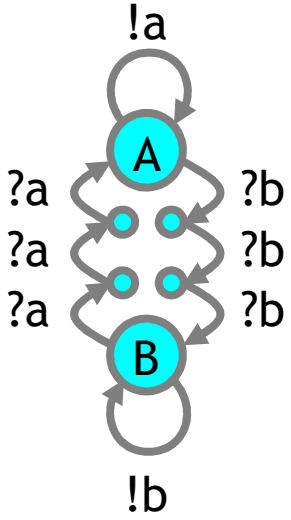

```
directive sample 10.0 1000
directive plot Ga(); Gb()

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())
```

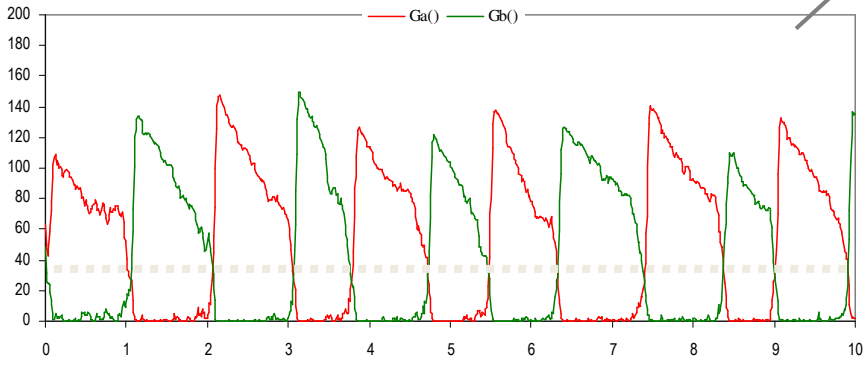


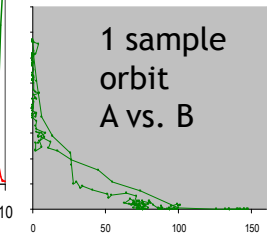
(With doping to break deadlocks)

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

"regular" oscillation





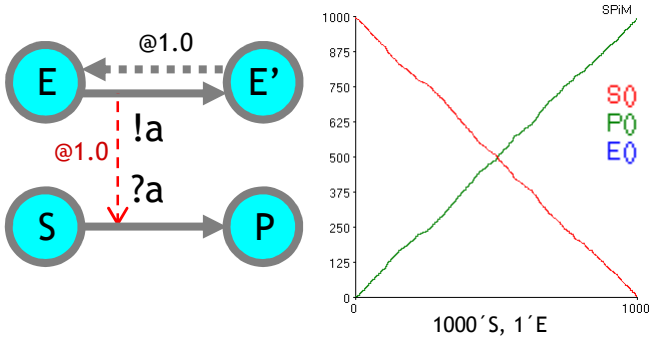



```
directive sample 10.0 1000
directive plot Ga(); Gb()

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())
```

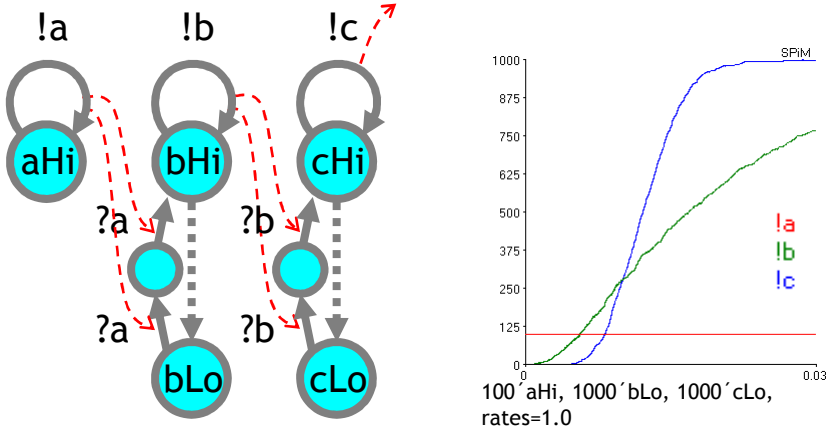
Devices

Some Devices

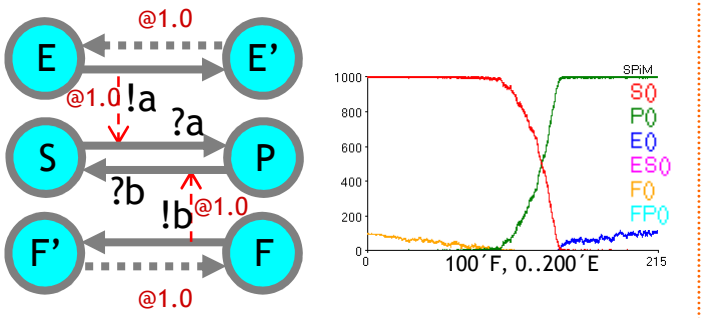
Linear Pump



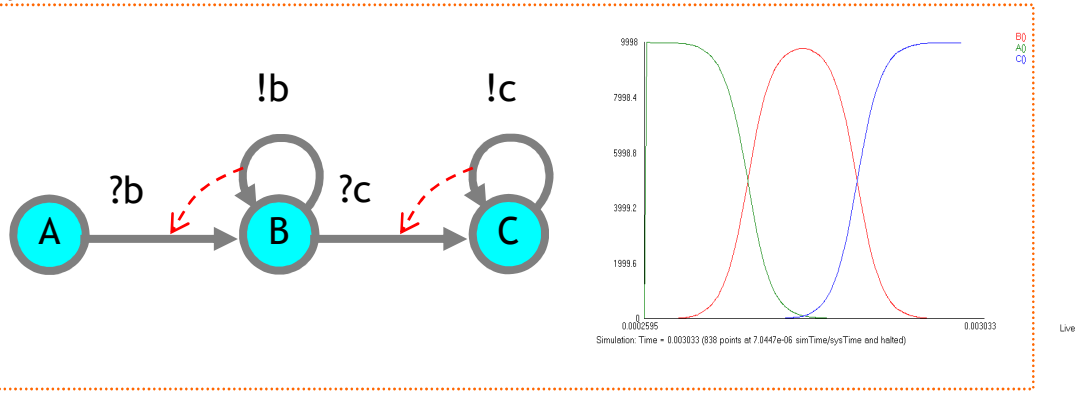
Cascade Amplifier



Ultrasensitive Switch

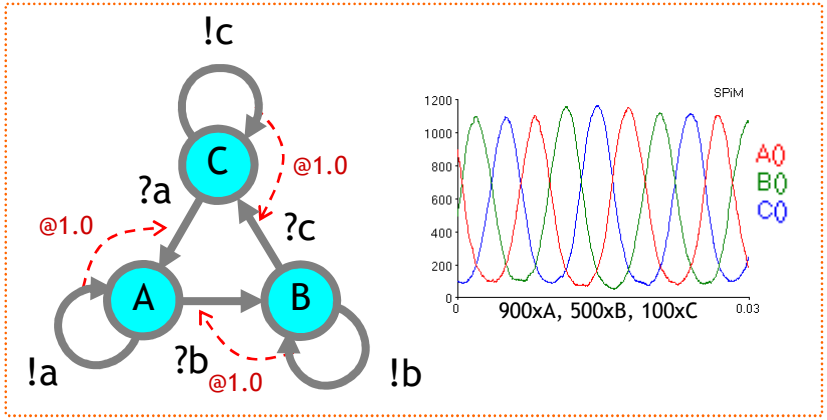


Symmetric Wave Generator

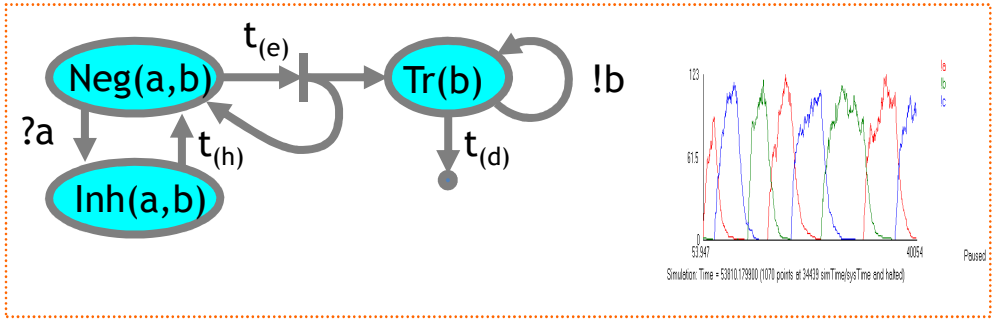


More Devices

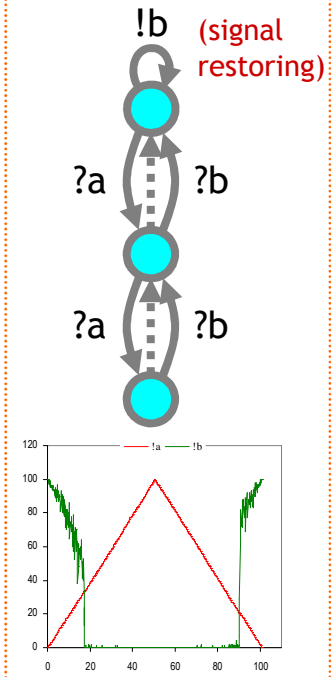
Oscillator



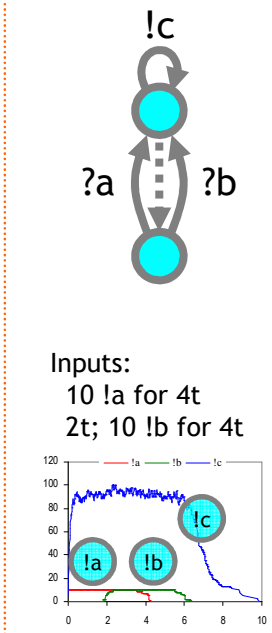
Repressilator (1 of 3 similar gates)



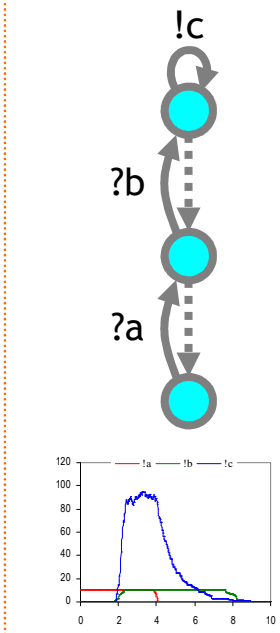
b = not a



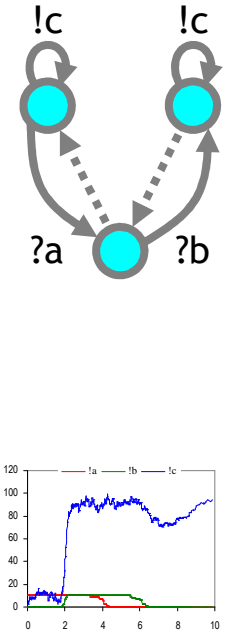
c = a or b



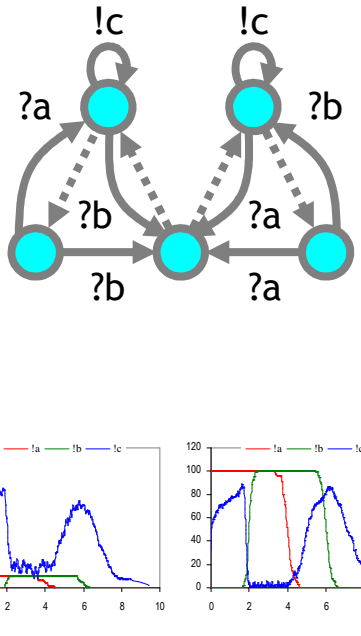
c = a and b



c = a imply b

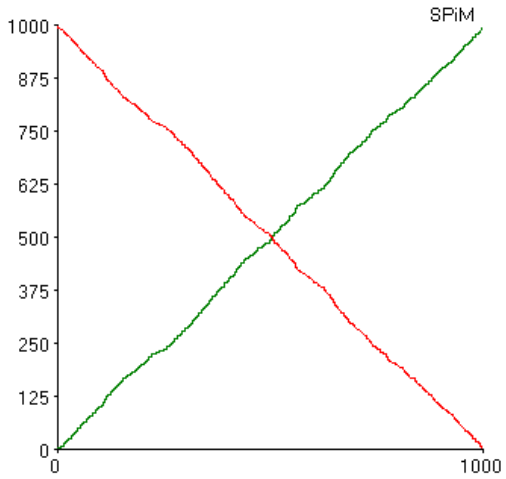


c = a xor b

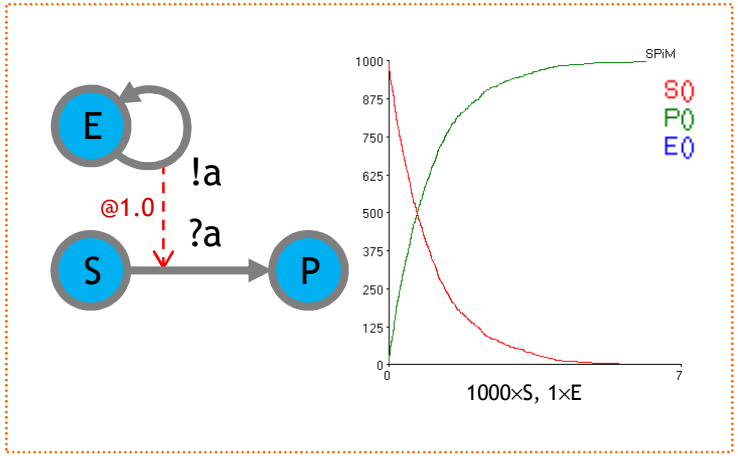


Design Exercise: Making Lines

Build me a population like this:



Second-order and Zero-order Regime

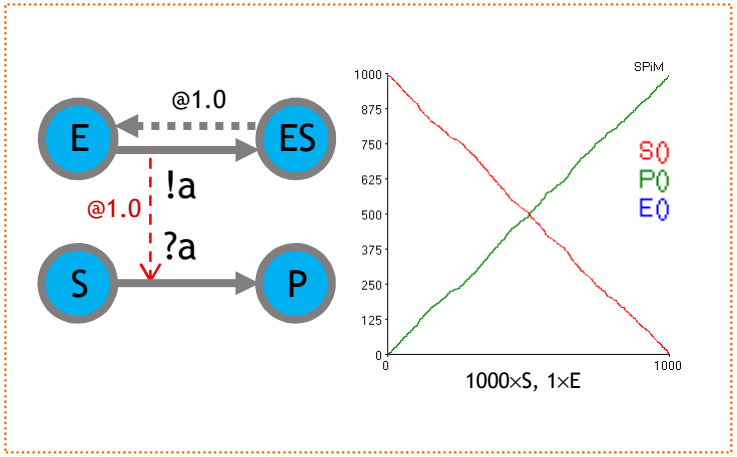


$E+S \rightarrow^{r} E+P$

```
directive sample 1000.0
directive plot S(); P(); E()

new a@1.0:chan()

let E() = !a; E()
and S() = ?a; P()
and P() = ()

run (1 of  E() | 1000 of S())
```

Second-Order Regime

$d[S]/dt = -r[E][S]$



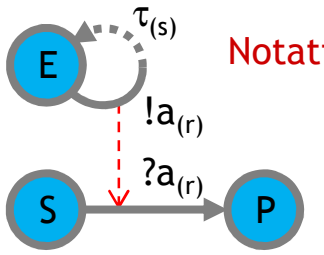
$E+S \rightarrow^{r} ES+P$

$ES \rightarrow^{s} E$

```
directive sample 1000.0
directive plot S(); P(); E()

new a@1.0:chan()

let E() = !a; delay@1.0; E()
and S() = ?a; P()
and P() = ()

run (1 of  E() | 1000 of S())
```

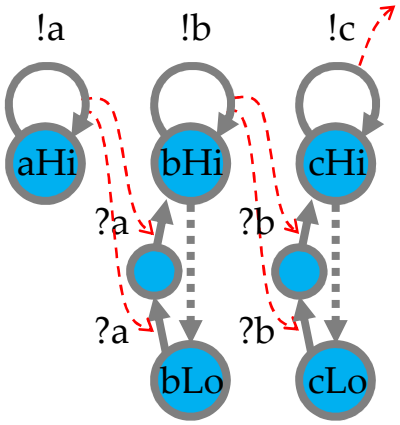
Zero-Order Regime

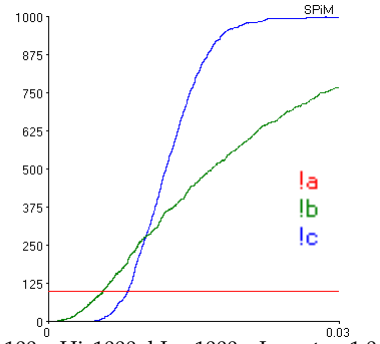
$d[S]/dt \cong -1$ (by assuming $d[ES]/dt = 0$)



Notation

Cascades



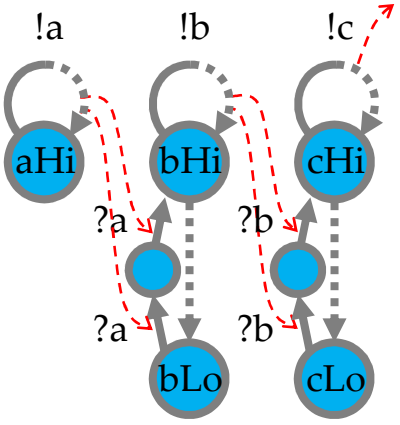


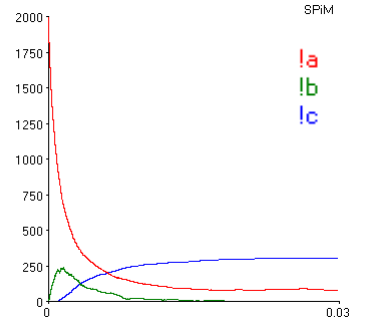
100×aHi, 1000×bLo, 1000×cLo, rates=1.0

Second-Oder Regime cascade:
a signal amplifier (MAPK)

aHi > 0 ⇒ cHi = max

```
directive sample 0.03
directive plot !a; !b; !c

new a@1.0:chan new b@1.0:chan new c@1.0:chan

let Amp_hi(a:chan, b:chan) =
  do !b; Amp_hi(a,b) or delay@1.0; Amp_lo(a,b)
and Amp_lo(a:chan, b:chan) =
  ?a; ?a; Amp_hi(a,b)

run 1000 of (Amp_lo(a,b) | Amp_lo(b,c))

let A() = !a; A()
run 100 of A()
```





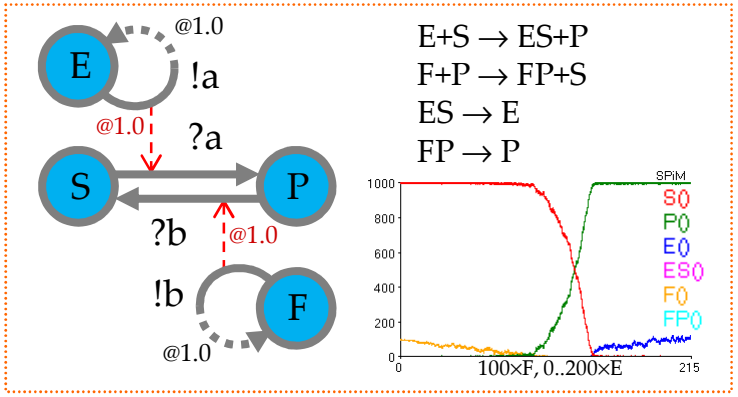
2000×aHi, 1000×bLo, 1000×cLo, rates=1.0

Zero-Oder Regime cascade:
a signal *divider!*

aHi = max ⇒ cHi = 1/3 max

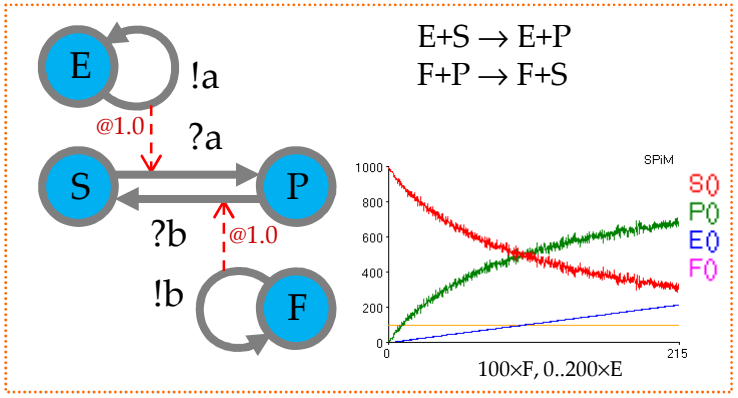
```
directive sample 0.03
directive plot !a; !b; !c

new a@1.0:chan new b@1.0:chan new c@1.0:chan

let Amp_hi(a:chan, b:chan) =
  do !b; delay@1.0; Amp_hi(a,b) or delay@1.0; Amp_lo(a,b)
and Amp_lo(a:chan, b:chan) =
  ?a; ?a; Amp_hi(a,b)

run 1000 of (Amp_lo(a,b) | Amp_lo(b,c))

let A() = !a; delay@1.0; A()
run 2000 of A()
```

Ultrasensitivity




```
directive sample 215.0
directive plot S(); P(); E(); ES(); F(); FP()

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

let S() = ?a; P()
and P() = ?b; S()

let E() = !a; delay@1.0; E()
and F() = !b; delay@1.0; F()

run 1000 of S()

let clock(t:float, tick:chan) =      (* sends a tick every t time *)
 (val ti = t/100.0 val d = 1.0/ti    (* by 100-step erlang timers *)
  let step(n:int) = if n<=0 then !tick; clock(t,tick) else delay@d; step(n-1)
  run step(100))
let Sig(p:proc(), tick:chan) = (p() | ?tick; Sig(p,tick))
let raising(p:proc(), t:float) =
 (new tick:chan run (clock(t,tick) | Sig(p,tick)))

run 100 of F()
run raising(E,1.0)
```

Zero-Order Regime
A small E-F inbalance causes a much larger S-P switch.

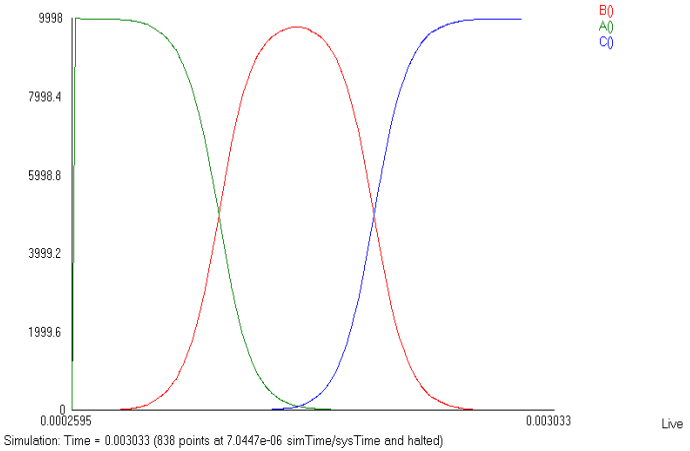



```
directive sample 215.0 1000
directive plot S(); P(); E(); F()

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

let S() = ?a; P()
and P() = ?b; S()

let E() = !a; E()
and F() = !b; F()

run 1000 of S()

let clock(t:float, tick:chan) =      (* sends a tick every t time *)
 (val ti = t/100.0 val d = 1.0/ti    (* by 100-step erlang timers *)
  let step(n:int) = if n<=0 then !tick; clock(t,tick) else delay@d; step(n-1)
  run step(100))
let Sig(p:proc(), tick:chan) = (p() | ?tick; Sig(p,tick))
let raising(p:proc(), t:float) =
 (new tick:chan run (clock(t,tick) | Sig(p,tick)))

run 100 of F()
run raising(E,1.0)
```

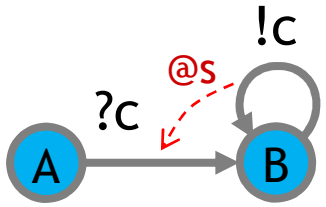
Second-Order Regime

Design Exercise: Making Waves

Build me a population like this:



Nonlinear Transition (NLT)



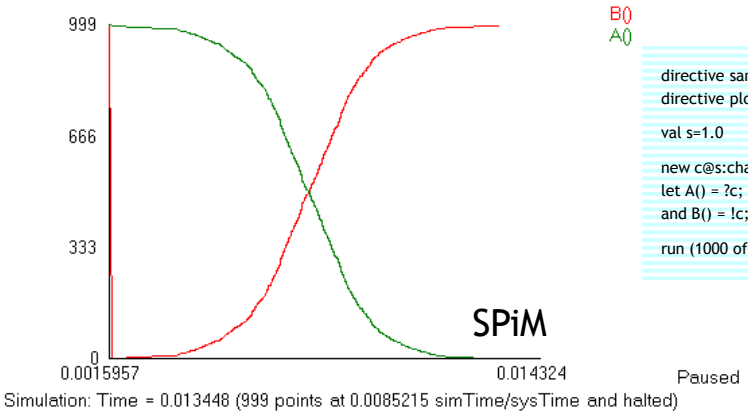
$A = ?c_{(s)};B$

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

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

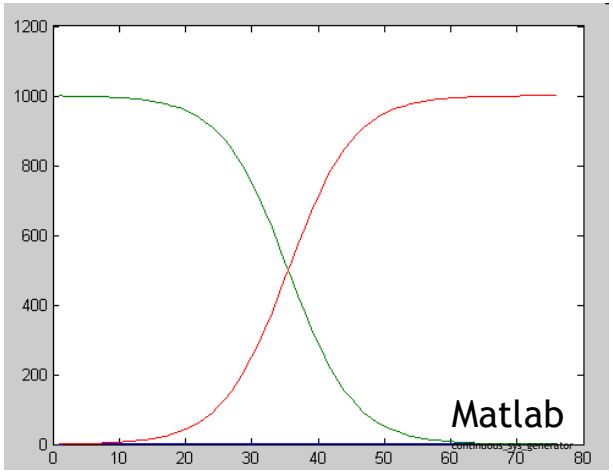
$d[A]/dt = -s[A][B]$

$d[B]/dt = s[A][B]$



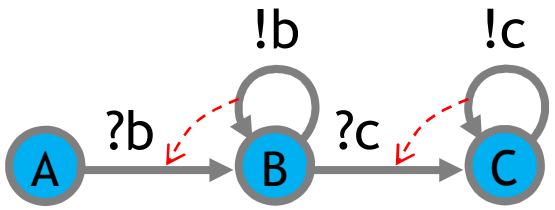

```
directive sample 0.02 1000
directive plot B(); A()

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



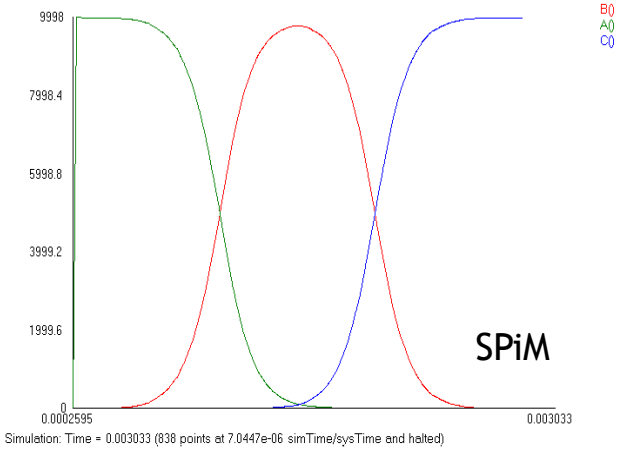
$d[B]/dt = [B]([A]-[C])$

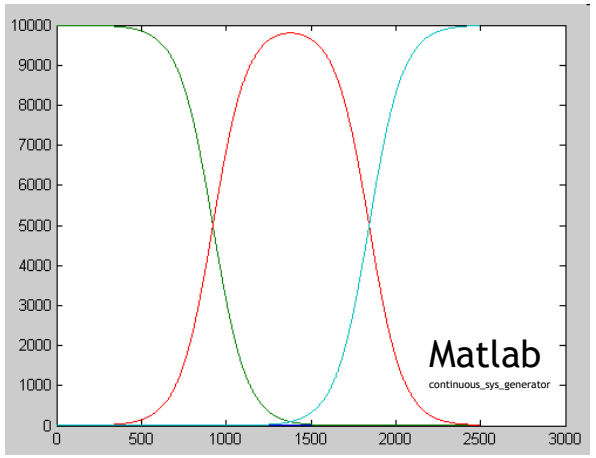
$A = ?b_{(1)};B$
$B = !b_{(1)};B \oplus ?c_{(1)};C$
$C = !c_{(1)};C$

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

$d[A]/dt = -[A][B]$
$d[B]/dt = [A][B]-[B][C]$
$d[C]/dt = [B][C]$

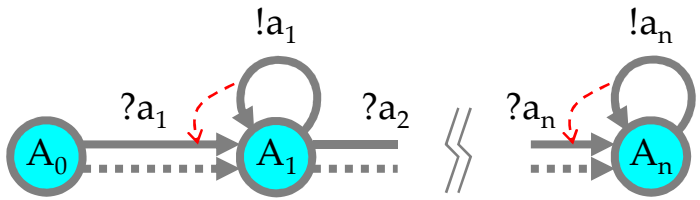
```
directive sample 0.0025 1000
directive plot B(); A(); C()

new b@1.0:chan new c@1.0:chan

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

run ((10000 of A()) | B() | C())
```

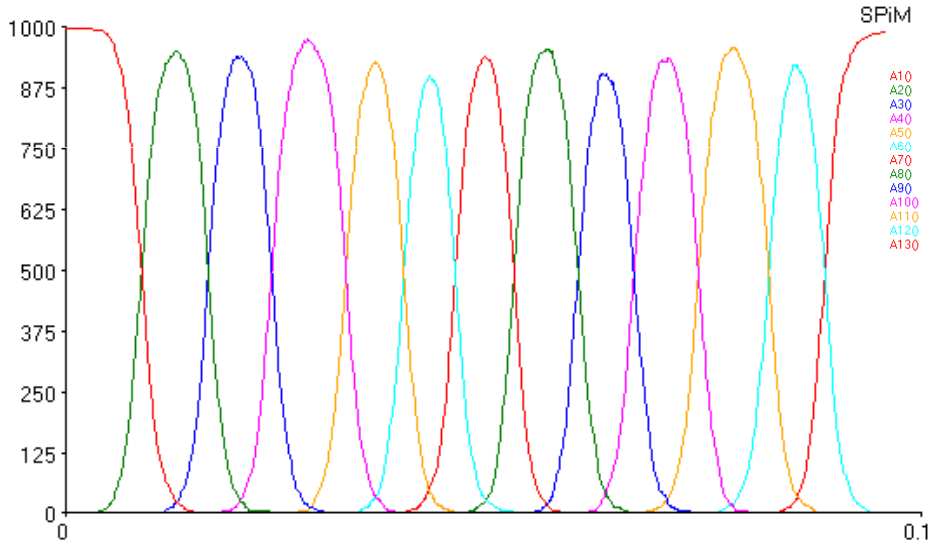





```
interval/step [0:0.000001:0.0025]
(A)   dx1/dt = -x1*x2          10000.0
(B)   dx2/dt = x1*x2 - x2*x3   1.0
(C)   dx3/dt = x2*x3           1.0
```

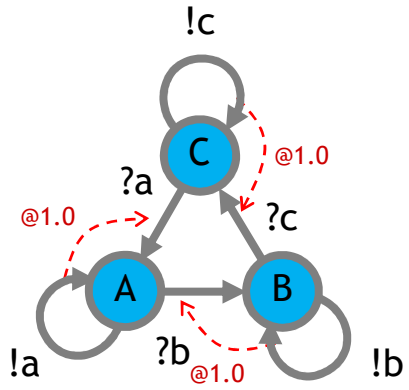
NLTs in Series: Soliton Propagation

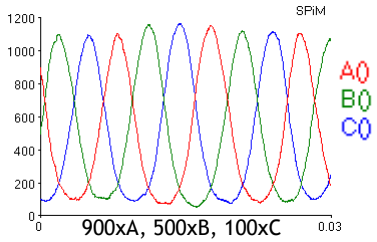





```
directive sample 0.1 1000
directive plot A1(); A2(); A3(); A4(); A5(); A6(); A7(); A8();
A9(); A10(); A11(); A12(); A13()

val r=1.0 val s=1.0

new a2@s:chan new a3@s:chan new a4@s:chan
new a5@s:chan new a6@s:chan new a7@s:chan
new a8@s:chan new a9@s:chan new a10@s:chan
new a11@s:chan new a12@s:chan new a13@s:chan
let A1() = do delay@r;A2() or ?a2; A2()
and A2() = do !a2;A2() or delay@r;A3() or ?a3; A3()
and A3() = do !a3;A3() or delay@r;A4() or ?a4; A4()
and A4() = do !a4;A4() or delay@r;A5() or ?a5; A5()
and A5() = do !a5;A5() or delay@r;A6() or ?a6; A6()
and A6() = do !a6;A6() or delay@r;A7() or ?a7; A7()
and A7() = do !a7;A7() or delay@r;A8() or ?a8; A8()
and A8() = do !a8;A8() or delay@r;A9() or ?a9; A9()
and A9() = do !a9;A9() or delay@r;A10() or ?a10; A10()
and A10() = do !a10;A10() or delay@r;A11() or ?a11; A11()
and A11() = do !a11;A11() or delay@r;A12() or ?a12; A12()
and A12() = do !a12;A12() or delay@r;A13() or ?a13; A13()
and A13() = !a13;A13()

run 1000 of A1()
```

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())
```

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

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

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

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

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

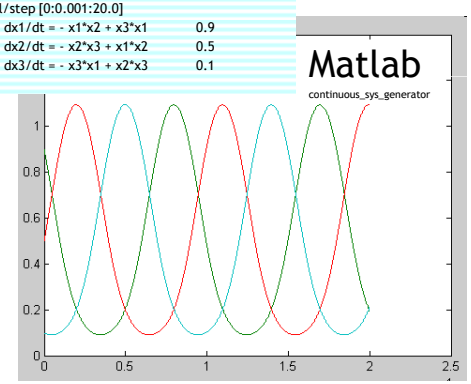
$C+A \rightarrow^{s} A+A$

$d[A]/dt = -s[A][B]+s[C][A]$

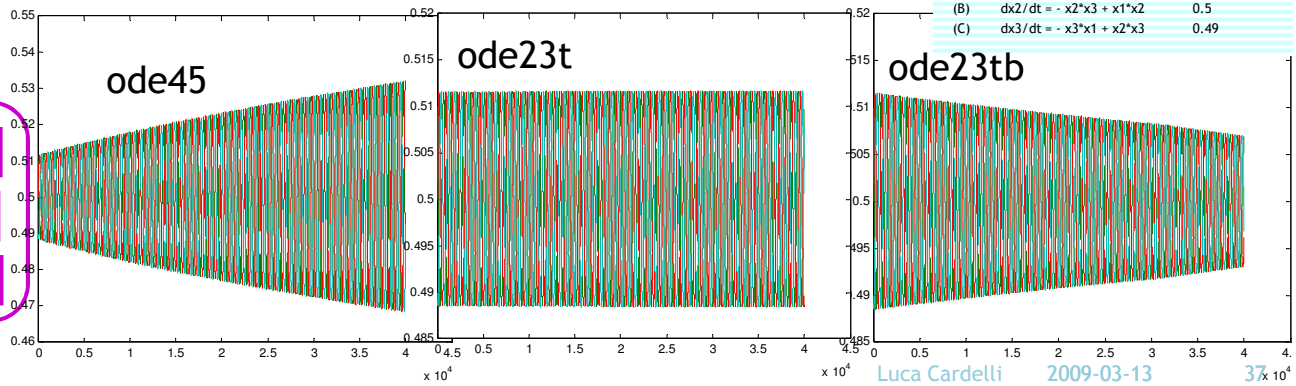
$d[B]/dt = -s[B][C]+s[A][B]$

$d[C]/dt = -s[C][A]+s[B][C]$

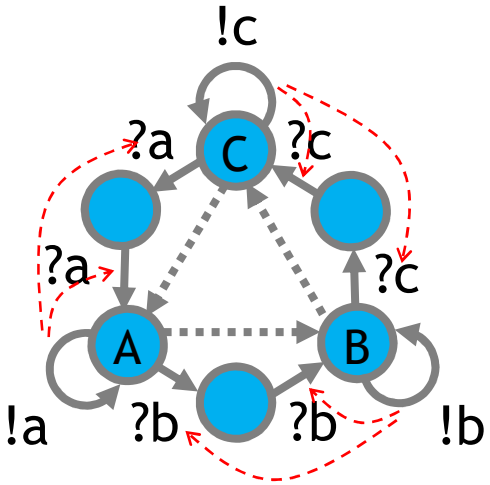
```
interval/step [0:0.001:20.0]
(A)   dx1/dt = - x1*x2 + x3*x1    0.9
(B)   dx2/dt = - x2*x3 + x1*x2    0.5
(C)   dx3/dt = - x3*x1 + x2*x3    0.1
```




```
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)



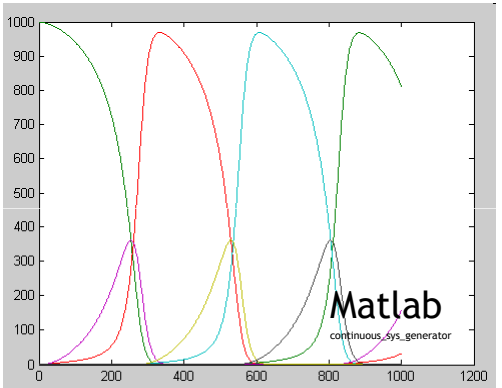

```
directive sample 0.1 1000
directive plot A1(); A2(); A3()

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()
```

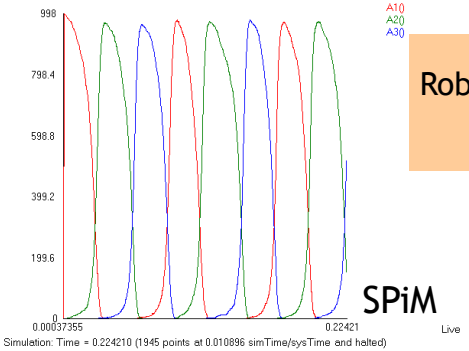
N.B. this does not deadlock!

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

Sustained Deterministic Oscillation

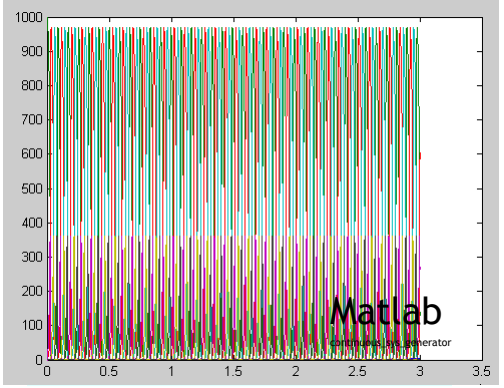
$A \rightarrow^r B$
$A+B \rightarrow^s A'+B$
$A'+B \rightarrow^s B+B$
$B \rightarrow^r C$
$B+C \rightarrow^s B'+C$
$B'+C \rightarrow^s C+C$
$C \rightarrow^r A$
$C+A \rightarrow^s C'+A$
$C'+A \rightarrow^s A+A$

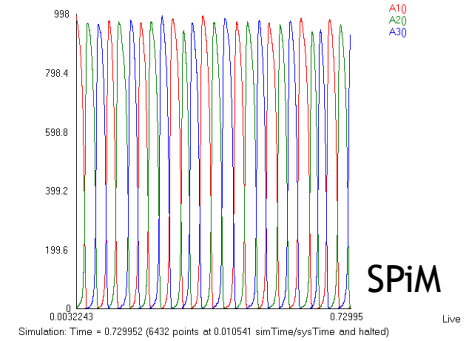




Robust Stochastic Oscillation

$d[A]/dt = -r[A]-s[A][B]+r[C]+s[C'][A]$
$d[B]/dt = -r[B]-s[B][C]+r[A]+s[A'][B]$
$d[C]/dt = -r[C]-s[C][A]+r[B]+s[B'][C]$
$d[A']/dt = -s[A'][B] + s[A][B]$
$d[B']/dt = -s[B'][C] + s[B][C]$
$d[C']/dt = -s[C'][A] + s[C][A]$





Semantics of Collective Behavior

“Micromodels”: Continuous Time Markov Chains

- The underlying semantics of stochastic π-calculus (and stochastic interacting automata). Well established in many ways.
  - Automata with rates on transitions.
- “The” correct semantics for chemistry, executable.
  - Gillespie stochastic simulation algorithm
- Lots of advantages
  - Compositional, compact, mechanistic, etc.
- But do not give a good sense of “collective” properties.
  - Yes one can do simulation.
  - Yes one can do program analysis.
  - Yes one can perhaps do modelchecking.
  - But somewhat lacking in “analytical properties” and “predictive power”.

"Macromodels": Ordinary Differential Equations

- The classical semantics of collective behavior.
  - E.g. kinetic theory of gasses.
  - They always ask: "How does you automata model relate to the 75 ODE models in the literature?"

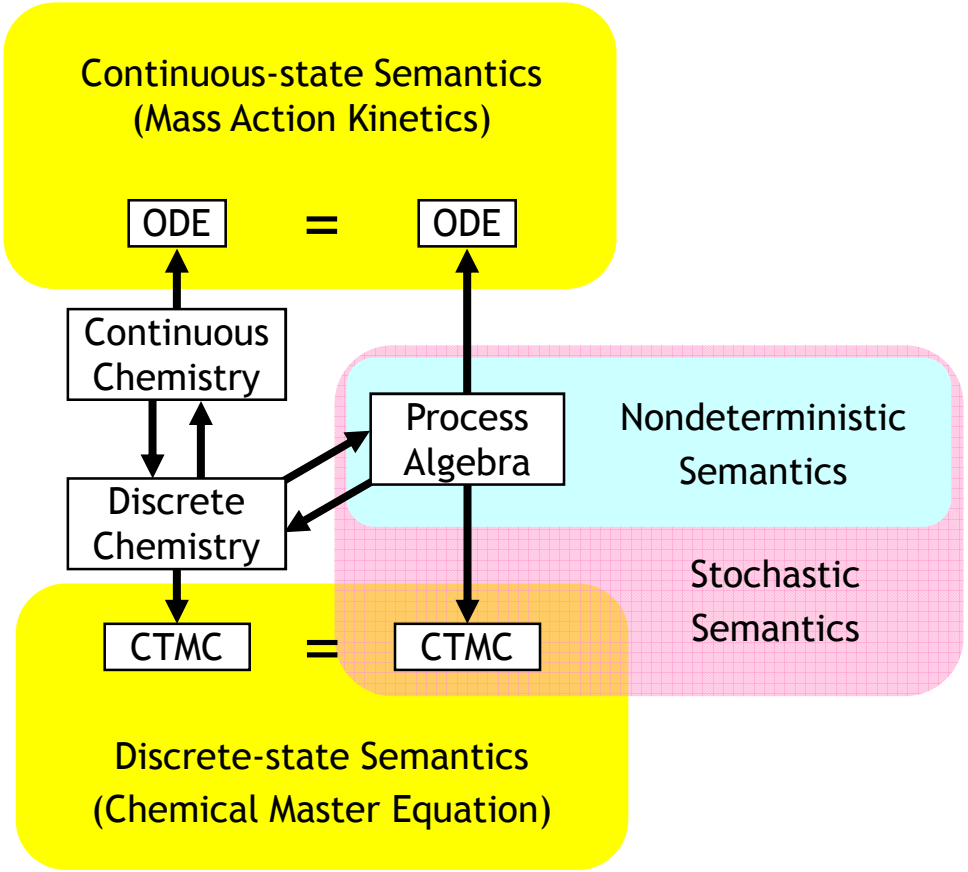
- Going from processes/automata to ODEs directly:
  - *In principle*: just write down the Rate Equation:
    - Let [S] be the "number of processes in state S" as a function of time.
    - Define for each state S:

      d[S]/dt = (rate of change of the number of processes in state S)

      Cumulative rate of transitions from any state S' to state S, times [S'],
      minus cumulative rate of transitions from S to any state S", times [S].
  - Fairly intuitive (rate = inflow minus outflow)

- Going to ODEs indirectly through chemistry
  - If we first convert processes to chemical reactions, then we can convert to ODEs by standard means!

The Two Semantic Sides of Chemistry

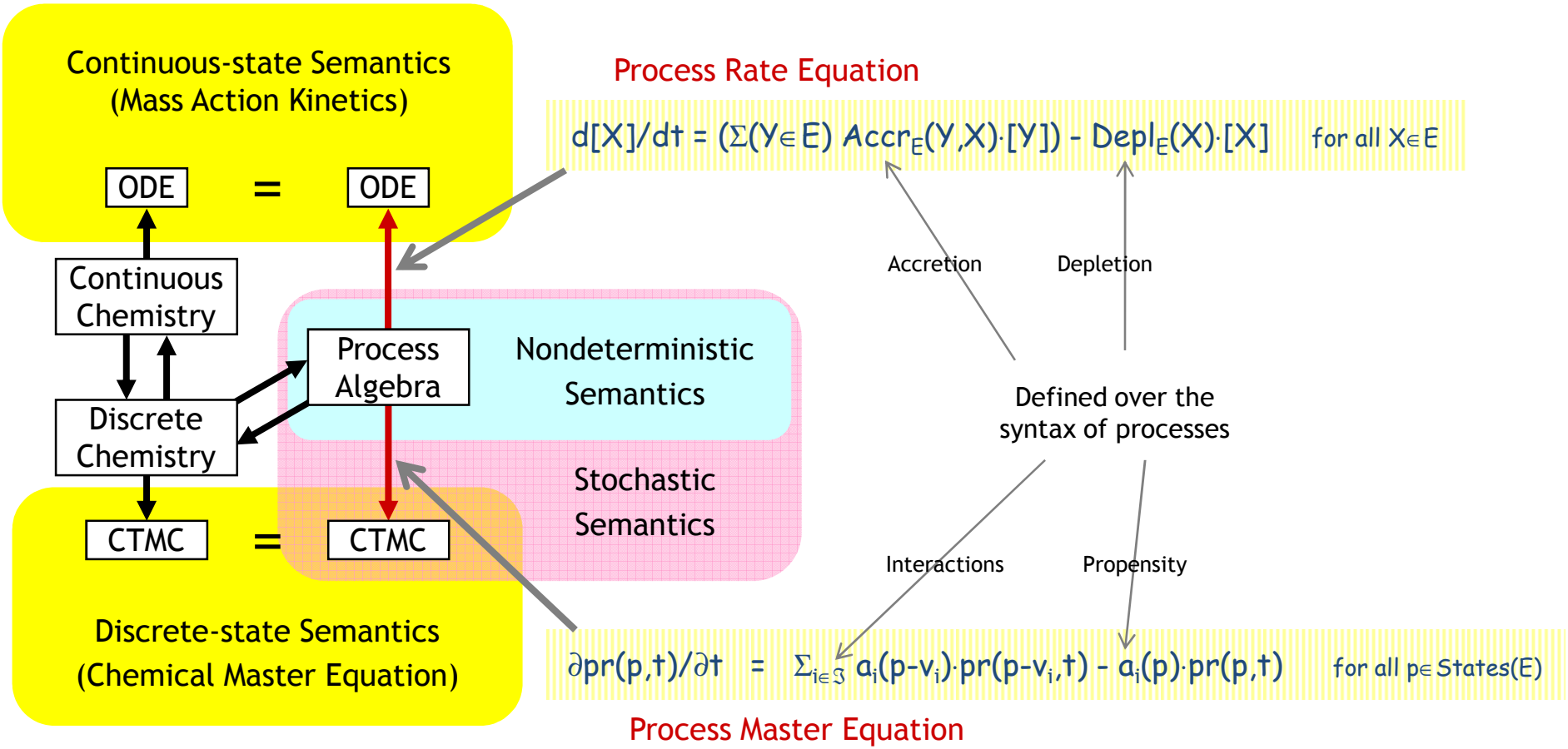


These diagrams commute via appropriate maps.

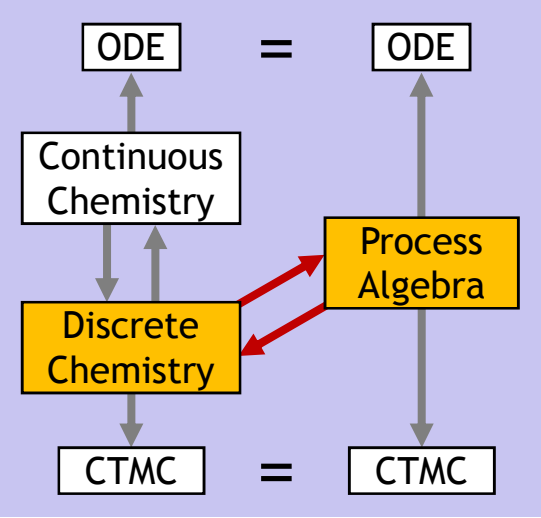
L. Cardelli: “On Process Rate Semantics” (TCS)

L. Cardelli: “A Process Algebra Master Equation” (QEST’07)

Quantitative Process Semantics



Stochastic Processes & Discrete Chemistry



Chemical Reactions (FSRN)

$A \rightarrow^{r} B_1 + \ldots + B_n$ $(n \geq 0)$	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)

$E ::= 0 \;:\; X=M,\, E$ — Reagents

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

$P ::= 0 \;:\; X \mid P$ — Solutions

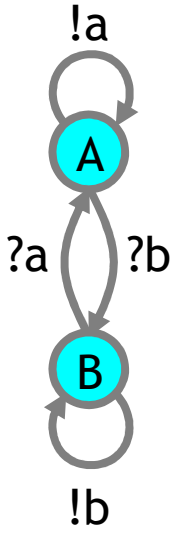
$\pi ::= \tau_{(r)} \;:\; ?a_{(r)} \;:\; !a_{(r)}$ — Actions (delay, input, output)

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

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

(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):

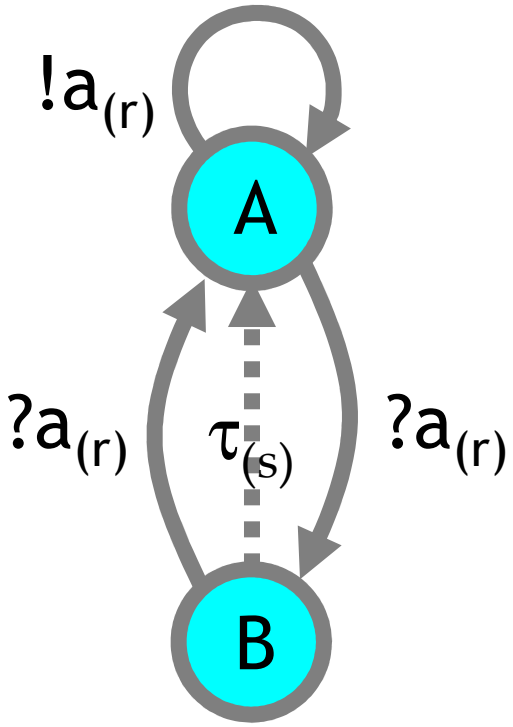
$A = !a;A \oplus ?b;B$ — Automaton in state A

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

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

From CGF to Chemistry

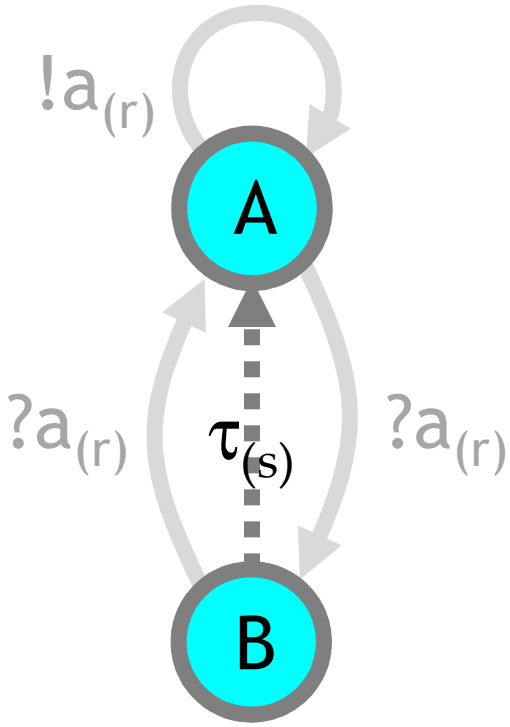
From CGF to Chemistry (by example)

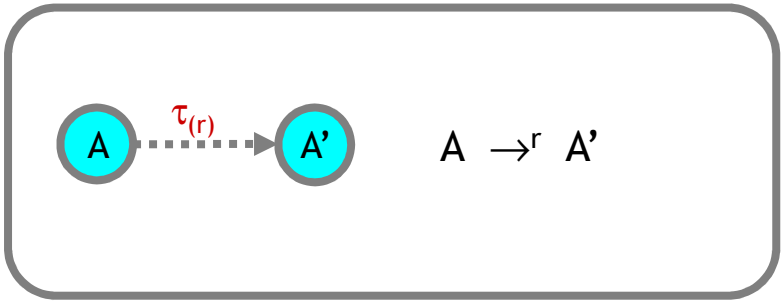


$A = !a_{(r)};A \oplus ?a_{(r)};B$

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

From CGF to Chemistry (by example)



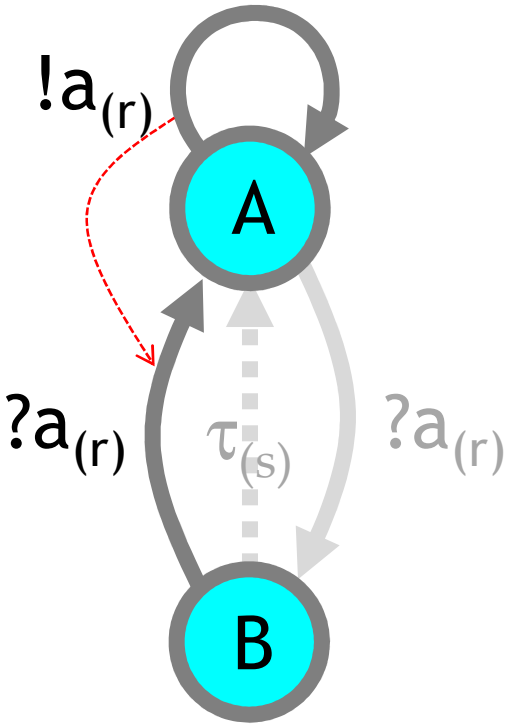


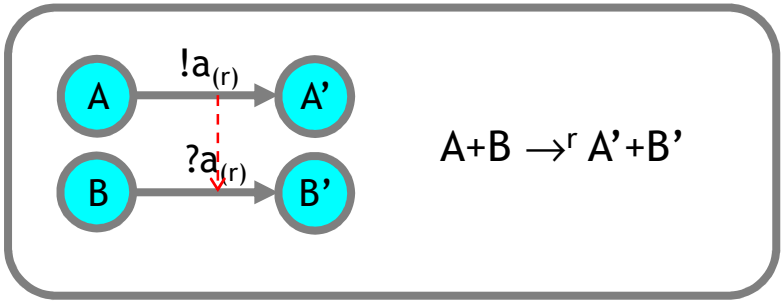
B →s A

A = !a;A ⊕ ?a;B

B = ?a;A ⊕ $\tau_{(s)}$;A

From CGF to Chemistry (by example)





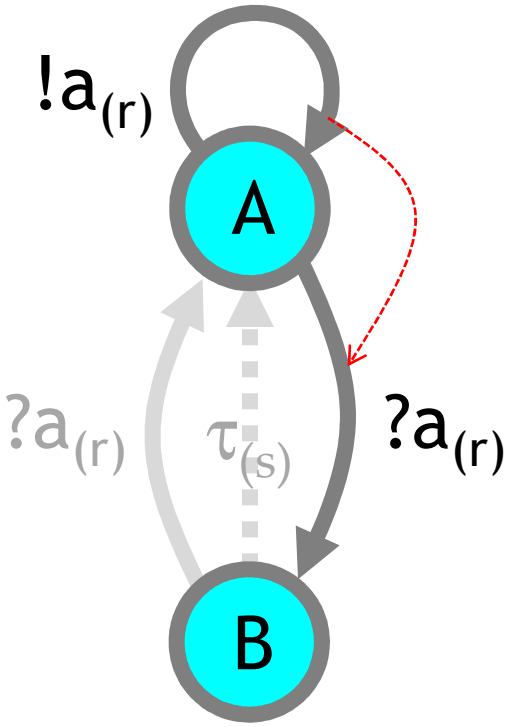
B →s A

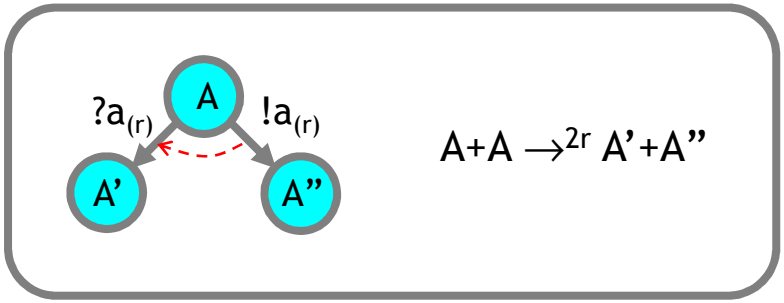
A+B →r A+A

A = !a;A ⊕ ?a;B

B = ?a;A ⊕ τ(s);A

From CGF to Chemistry (by example)





$B \rightarrow^{s} A$

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

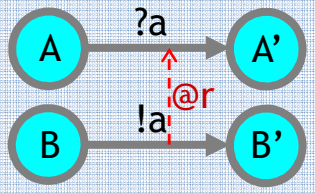
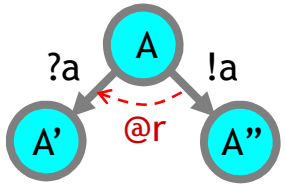
$A+A \rightarrow^{2r} A+B$

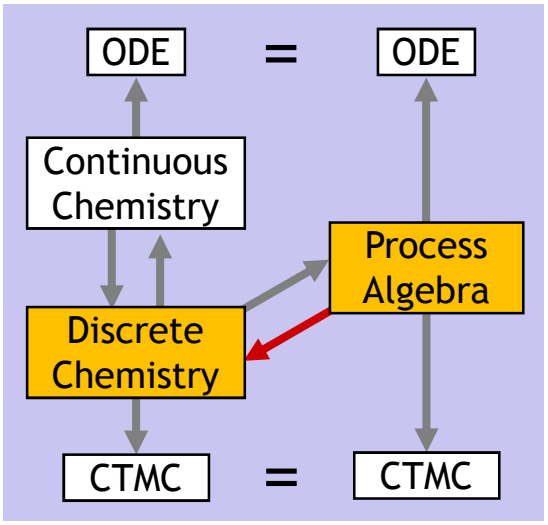
Double rate for homeo reactions

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

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

From CGF to Chemistry (by example)

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



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 Å-summand of the molecule M associated with the X reagent of E

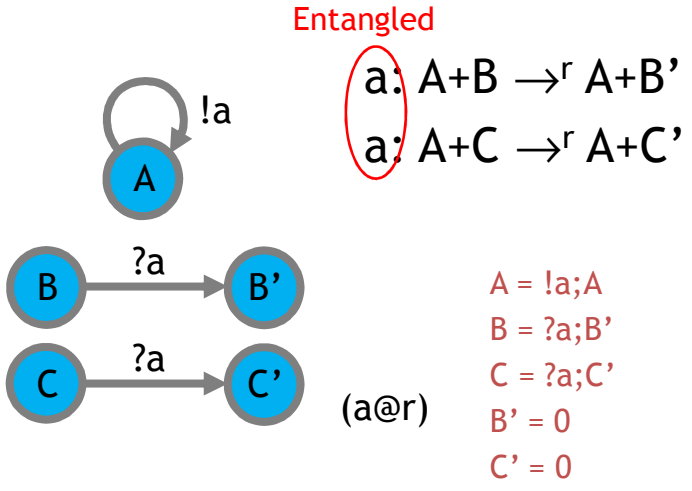
Chemical reactions for E,P: (N.B.: <...> are reaction tags to obtain multiplicity of reactions, and P is P with all the | changed to +)

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

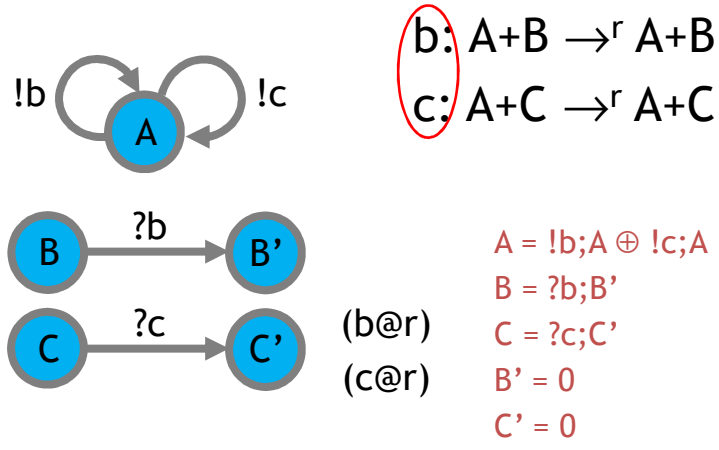
Initial conditions for P:

$$Ch(P) := P$$

Entangled vs Detangled



Entangled: Two reactions on one channel



Detangled: Two reactions on two separate channels

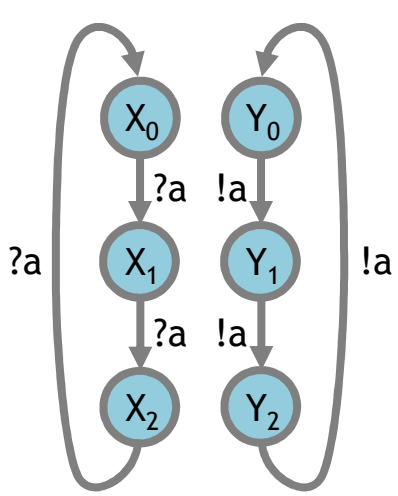
We need a semantics of automata that identifies automata that have the "same chemistry".

No traditional process algebra equivalence is like this!

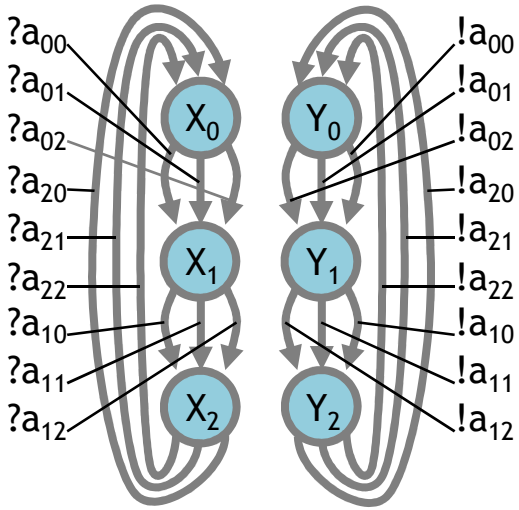
Entangled automata lead to more compact models than in chemistry.

Detangled automata are in simple correspondence with chemistry.

Entangled vs detangled



E_3



Detangle(E_3)

(closely related to
Pi(Ch(E_3)))

Chemical Parametric Form (CPF)

$E ::= 0 \;\vdots\; X(\mathbf{p})=M,\ E$ — Reagents

$M ::= 0 \;\vdots\; \pi;P \oplus M$ — Molecules

$P ::= 0 \;\vdots\; X(\mathbf{p}) \mid P$ — Solutions

$\pi ::= \tau_{(r)} \;\vdots\; ?a_{(r)}(\mathbf{p}) \;\vdots\; !a_{(r)}(\mathbf{p})$ — Actions

$CPF ::= E,P$ — with initial conditions

$\oplus$ 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*
p are vectors of names
p are vectors of distinct names when in binding position
Each free name a in E is assigned a fixed rate r: $a_{(r)}$

Not bounded-state systems.

Not finite-control systems.

But still finite-species systems.

A translation from CPF to CGF exists
(expanding all possible instantiation of parameters from the initial conditions)

An incremental translation algorithm exists
(expanding on demand from initial conditions)

Example:

$Neg(a,b) = ?a;\ Inh(a,b) \oplus \tau_e;\ (Tr(b) \mid Neg(a,b))$

$Inh(a,b) = \tau_h;\ Neg(a,b)$

$Tr(b) = !b;\ Tr(b) \oplus \tau_d;\ 0$

$Neg(x,x)$

CPF to CGF: Handling Parameters

Consider first the CPF subset with no communication (pure ?a, !a).

Grounding (replace parameters with constants)

where X/**p** is a name in bijection with <X,**p**>
(each X/**p** is seen as a separate *species*)

$$/(\pi_1;P_1 \oplus \ldots \oplus \pi_n;P_n) =_{def} \pi_1;/(P_1) \oplus \ldots \oplus \pi_n;/(P_n)$$
$$/(X_1(\mathbf{p}_1) \mid \ldots \mid X_n(\mathbf{p}_n)) =_{def} X_1/\mathbf{p}_1 \mid \ldots \mid X_n/\mathbf{p}_n$$

$$E ::= X_1(\mathbf{p}_1)=M_1, \ldots, X_n(\mathbf{p}_n)=M_n$$
$$M ::= \pi_1;P_1 \oplus \ldots \oplus \pi_n;P_n$$
$$P ::= X_1(\mathbf{p}_1) \mid \ldots \mid X_n(\mathbf{p}_n)$$
$$\pi ::= \tau_r \quad ?a \quad !a$$

Let N be the set of free names occurring in E.

E_G is the **Parametric Explosion** of E (still a finite species system)
computed by replacing parameters with all combinations of free names in E

$$E_G := \{(X/\mathbf{q} = /(M\{\mathbf{p}\leftarrow\mathbf{q}\})) \text{ s.t. } (X(\mathbf{p}) = M) \in E \text{ and } \mathbf{q} \in N^{\#\mathbf{p}}\}$$
$$P_G := /P \qquad \text{(simply ground the given initial conditions once)}$$

E_G is a CGF! To obtain the chemical reactions $Ch_P(E)$, just compute $Ch_G(E_G)$

$$Ch_P(E) = Ch_G(E_G)$$

CPF to CGF: Handling Communication

Grounding (replace parameters with constants)

just one main change: now also convert each input parameter into a ground choice of all possible inputs

N is the set of free names in E,P

#**p** is the length of **p**

n/**p** is a name in bijection with <n,**p**>

X/**p** is a name in bijection with <X,**p**>

(each X/**p** is seen as a separate *species*)

$$E ::= X_1(\mathbf{p}_1)=M_1, \ldots, X_n(\mathbf{p}_n)=M_n$$
$$M ::= \pi_1;P_1 \oplus \ldots \oplus \pi_n;P_n$$
$$P ::= X_1(\mathbf{p}_1) \mid \ldots \mid X_n(\mathbf{p}_n)$$
$$\pi ::= \tau_r \quad ?a(\mathbf{p}) \quad !a(\mathbf{p})$$

$$/_N(\tau_r;P) = \tau_r;\ /_N(P)$$
$$/_N(!a_{(r)}(\mathbf{p});P) = !a/\mathbf{p}_{(r)};\ /_N(P)$$
$$/_N(?a_{(r)}(\mathbf{p});P) = \oplus(\mathbf{q}\in N^{\#\mathbf{p}}) \text{ of } ?a/\mathbf{q}_{(r)};\ /_N(P\{\mathbf{p}\leftarrow\mathbf{q}\})$$
$$/_N(\pi_1;P_1 \oplus \ldots \oplus \pi_n;P_n) = /_N(\pi_1;P_1) \oplus \ldots \oplus /_N(\pi_n;P_n)$$
$$/_N(X_1(\mathbf{p}_1) \mid \ldots \mid X_n(\mathbf{p}_n)) = X_1/\mathbf{p}_1 \mid \ldots \mid X_n/\mathbf{p}_n$$

E_G is again the **Parametric Explosion** of E

$$E_G := \{(X/\mathbf{q} = /_N(M\{\mathbf{p}\leftarrow\mathbf{q}\})) \text{ s.t. } (X(\mathbf{p}) = M) \in E \text{ and } \mathbf{q} \in N^{\#\mathbf{p}}\}$$
$$P_G := /_N(P)$$ (simply ground the given initial conditions once)

$$Ch(E) = Ch_G(E_G)$$

E_G is a again a CGF!

CPF to CGF Translation. Ex: Neg(x,x)

E =

Neg(a,b) = ?a; Inh(a,b) ⊕ τ_e; (Tr(b) | Neg(a,b))
Inh(a,b) = τ_h; Neg(a,b)
Tr(b) = !b; Tr(b) ⊕ τ_d; 0
Neg(x,x)

----- initialization -----

E_c:= {Neg/x,x = ?x; Inh/x,x ⊕ τ_e; (Tr/x | Neg/x,x)}

----- iteration 1 -----

C := {Neg/x,x $\rightarrow^e$ Tr/x + Neg/x,x }

E_c:= {Neg/x,x = ?x; Inh/x,x ⊕ τ_e; (Tr/x | Neg/x,x)
Tr/x = !x; Tr/x ⊕ τ_d; 0}

----- iteration 2 -----

C := {Neg/x,x $\rightarrow^e$ Tr/x + Neg/x,x
Tr/x $\rightarrow^d$ 0
Tr/x + Neg/x,x $\rightarrow^{\rho(x)}$ Tr/x + Inh/x,x }

E_c:= {Neg/x,x = ?x; Inh/x,x ⊕ τ_e; (Tr/x | Neg/x,x)
Tr/x = !x; Tr/x ⊕ τ_d; 0
Inh/x,x = τ_h; Neg/x,x}

----- iteration 3 -----

C := {Neg/x,x $\rightarrow^e$ Tr/x + Neg/x,x
Tr/x $\rightarrow^d$ 0
Tr/x + Neg/x,x $\rightarrow^{\rho(x)}$ Tr/x + Inh/x,x
Inh/x,x $\rightarrow^h$ Neg/x,x}

E_c:= no change

----- termination -----

Neg/x,x $\rightarrow^e$ Tr/x + Neg/x,x
Tr/x $\rightarrow^d$ 0
Tr/x + Neg/x,x $\rightarrow^{\rho(x)}$ Tr/x + Inh/x,x
Inh/x,x $\rightarrow^h$ Neg/x,x
Neg/x,x

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 $\tau;P_i$ to $\langle X, v_{ij} \rangle$.

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 $\tau;P_i$ to $\langle X,v_{ij}\rangle$.

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 $\tau;P_i$ to $\langle X,v_{ij}\rangle$.

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: $B \rightarrow^{s} A$

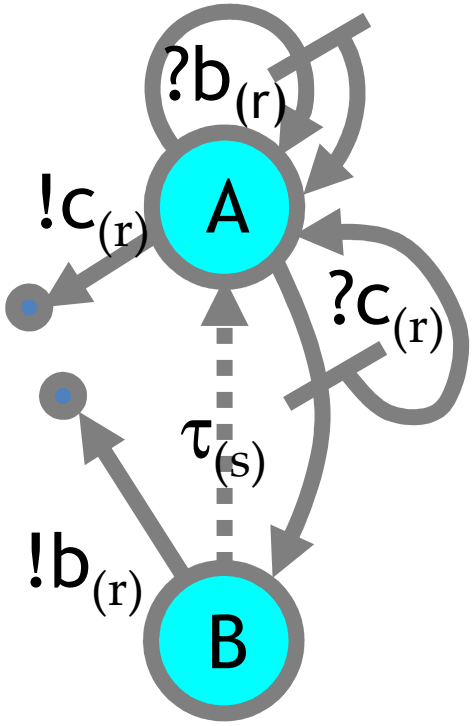
b: $A+B \rightarrow^{r} A+A$

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 $\tau;P_i$ to $\langle X,v_{ij}\rangle$.
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:

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

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

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	

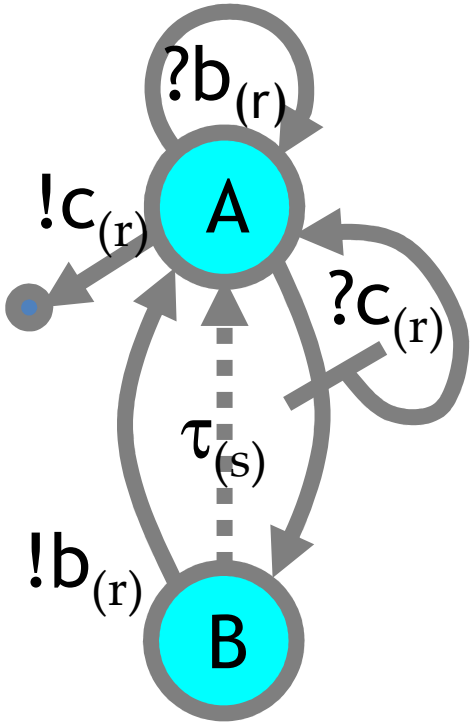
1: Fill the matrix by columns:

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

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

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



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	

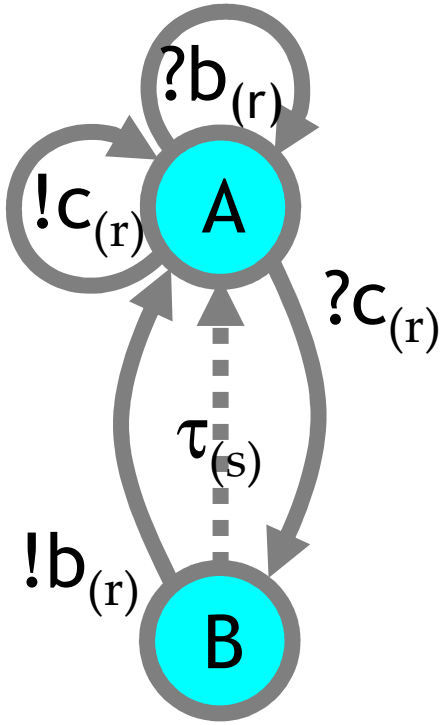
1: Fill the matrix by columns:

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

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

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



From Chemistry to Automata (by example)

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

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

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

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

Interaction Matrix

channels and rates
(1 per reaction)

Half-rate for homeo reactions

definitions
(1 per species)

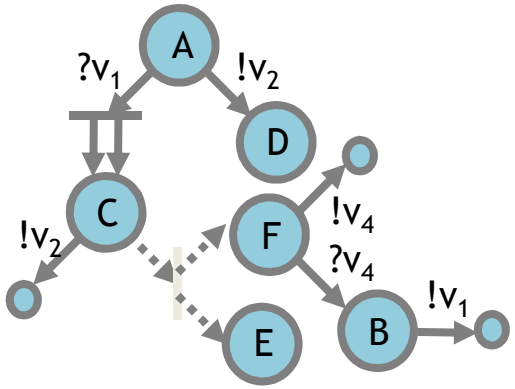
	$v_{1(k1)}$	$v_{2(k2)}$	$v_{3(k3)}$	$v_{4(k4/2)}$
A	?;(C|C)	?;D		
B	!;0			
C		!;0	τ;(E|F)	
D				
E				
F				?;B !;0

1: Fill the matrix by columns:

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

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

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



2: Read the result by rows:

A = $?v_{1(k1)};(C|C) \oplus ?v_{2(k2)};D$

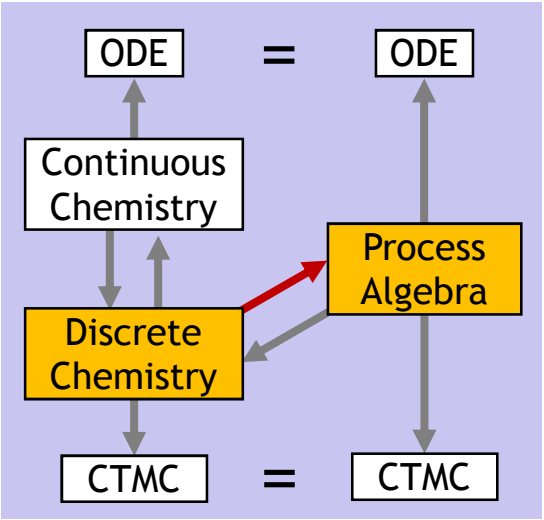
B = $!v_{1(k1)};0$

C = $!v_{2(k2)};0 \oplus \tau_{k3};(E|F)$

D = 0

E = 0

F = $?v_{4(k4/2)};B \oplus !v_{4(k4/2)};0$



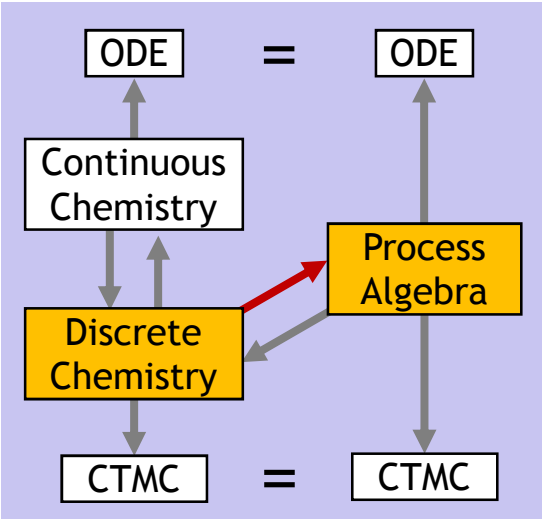
From Chemistry to CGF: Pi(C)

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):

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



Some Syntactic Properties

- C and Ch(Pi(C)) have the same reactions
  - (and their reaction labels are in bijection)

- Def: E is detangled if each channel appears once as ?a and once as !a.

- If C is a system of chemical reactions then Pi(C) is detangled.
  - (hence chemical reactions embed into a subclass of CGFs)

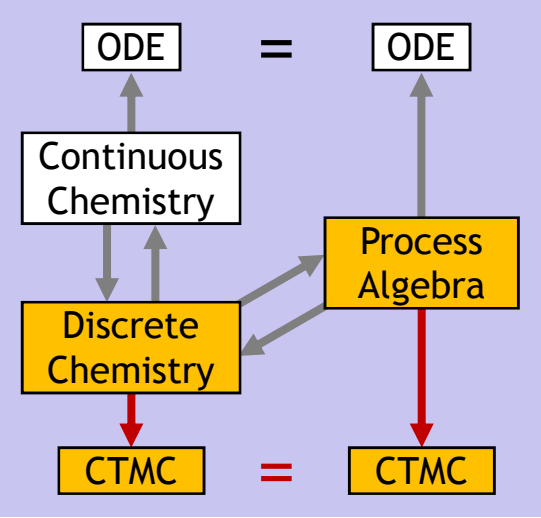
- Hence for any E, we have that Pi(Ch(E)) is detangled.
  - (E and Pi(Ch(E)) are "equivalent" CGFs, but that has to be shown later)

- Def: E,P is automata form if "|" occurs only (other than "|0") in P.

- Def: Detangle(E) is defined from Pi(Ch(E)) by replacing any occurrence pairs $?a_{(r)};(X|Y|0)$ and $!a_{(r)};0$ with $?a_{(r)};(X|0)$ and $!a_{(r)};(Y|0)$.

- If E is in automata form then Detangle(E) is (detangled and) in automata form
  - (but Pi(Ch(E)) may not be)

Discrete-State Semantics



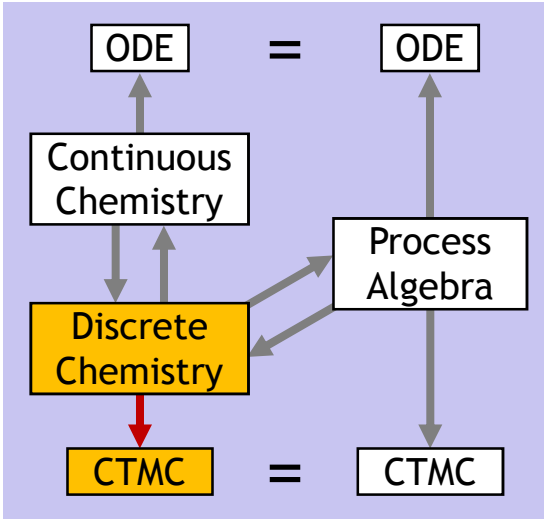
Discrete Semantics of Reactions

Syntax:

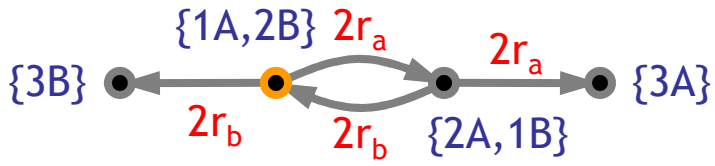
$A+B \rightarrow^{r} A+A$

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

$A+B+B$

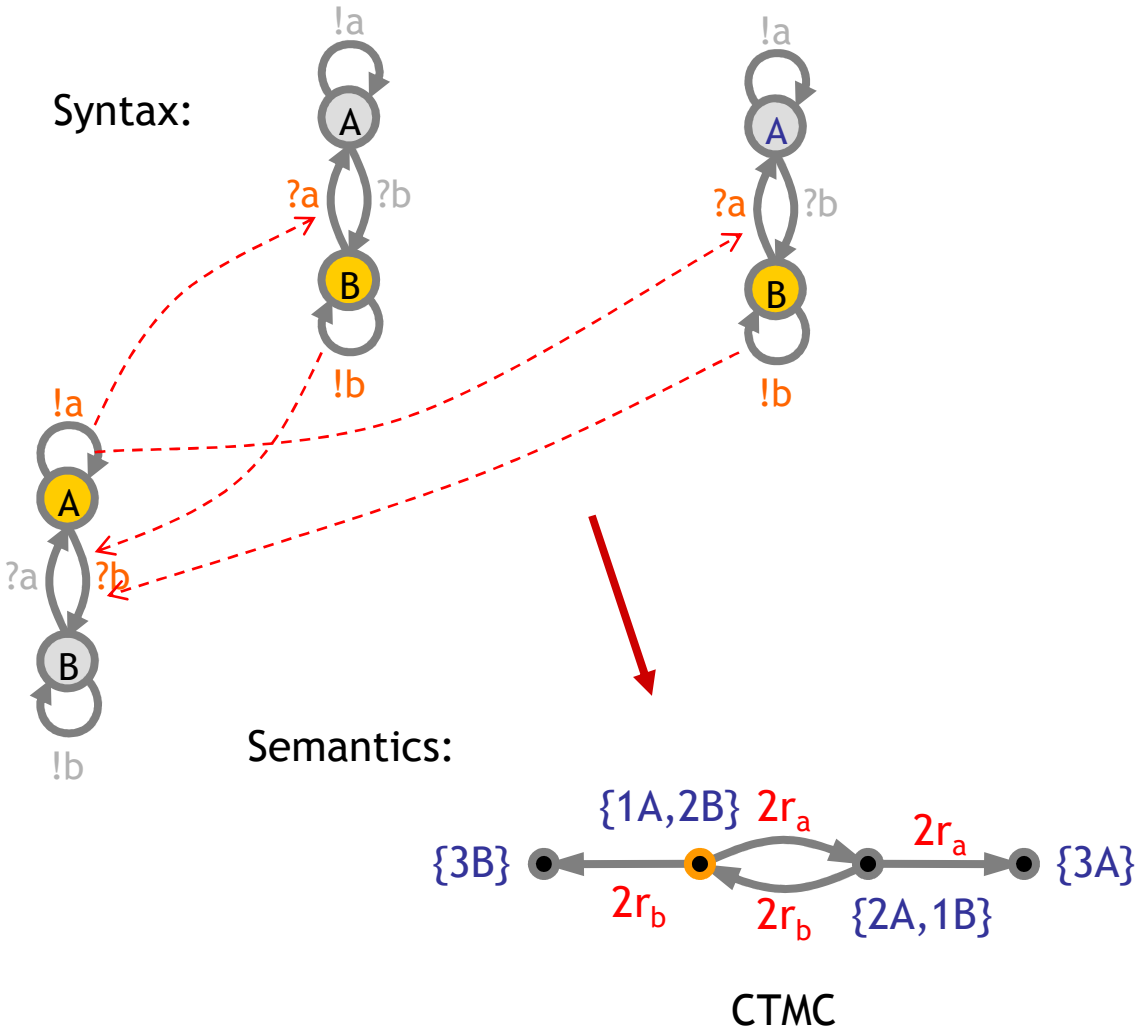


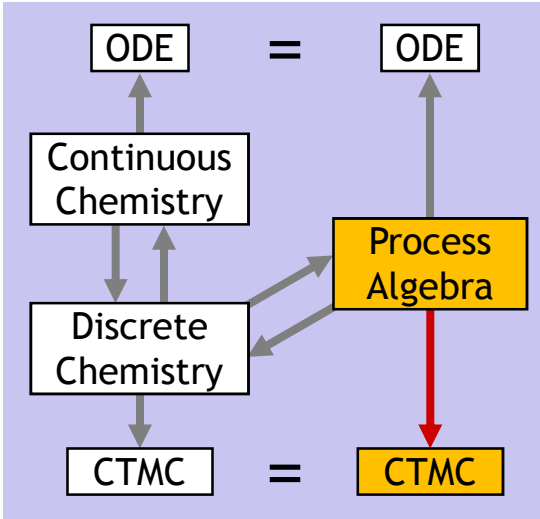
Semantics:



CTMC

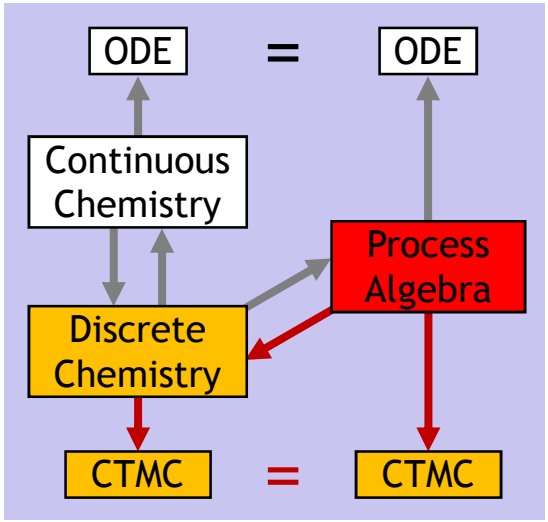
Discrete Semantics of Reagents

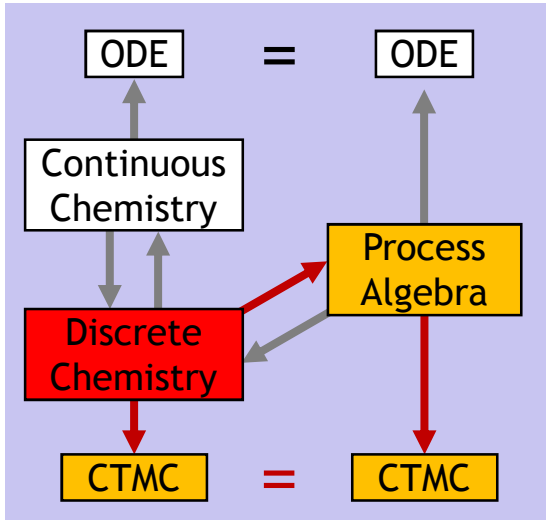




Discrete State Equivalence

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





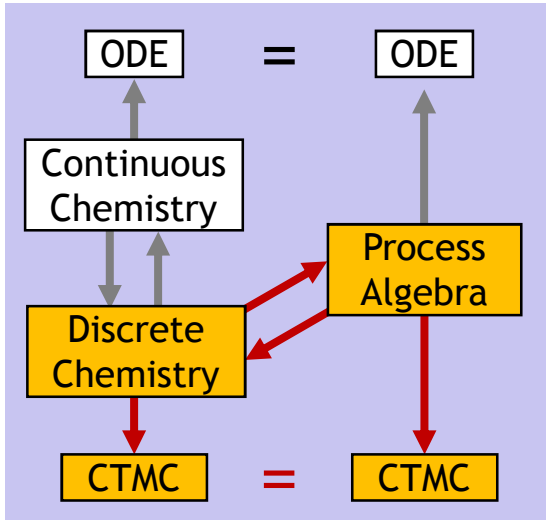
- For each E there is an E' ≋ E that is detangled (E' = Pi(Ch(E)))
- For each E in automata form there is an an E' ≋ E that is detangled and in automata form (E' = Detangle(E)).

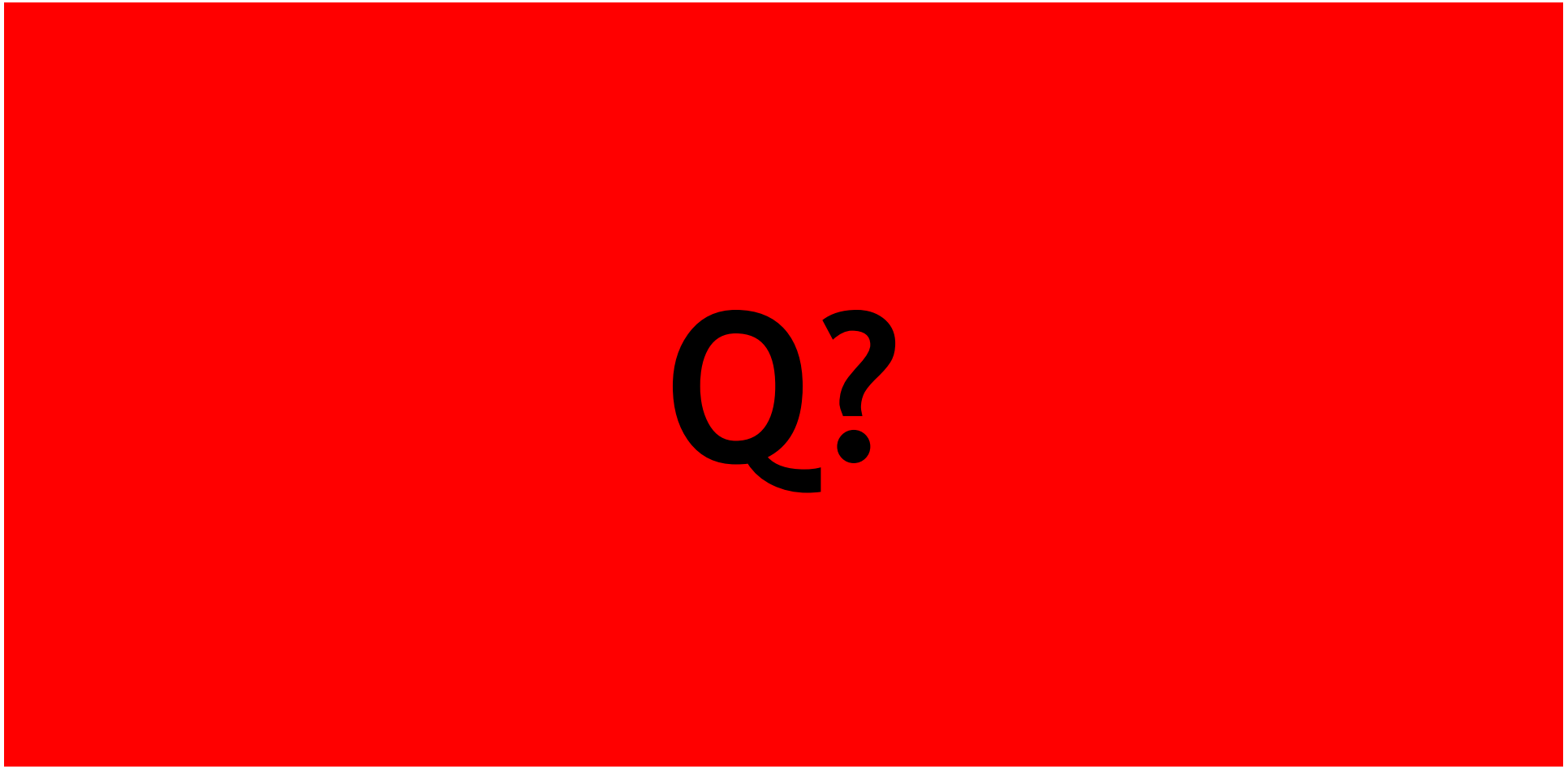
Interacting Automata = Discrete Chemistry

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

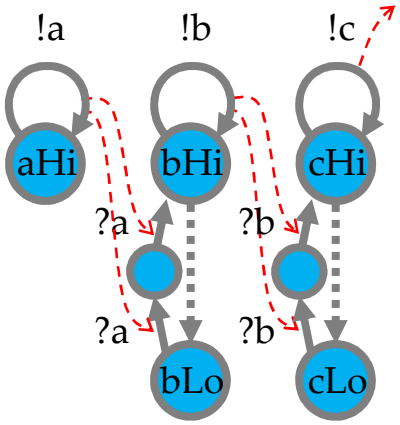
But CTMC are not the “ultimate semantics” because there are still questions of when two different CTMCs are actually equivalent (e.g. “lumping”).

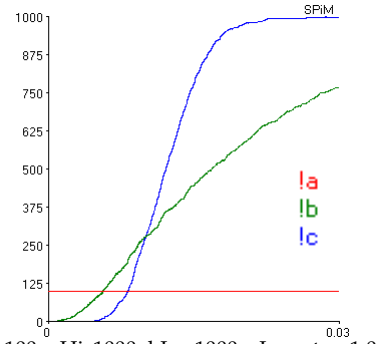
The “ultimate semantics” of chemistry is the *Chemical Master Equation* (derivable from the Chapman-Kolmogorov equation of the CTMC).




Q?

Exercise 1





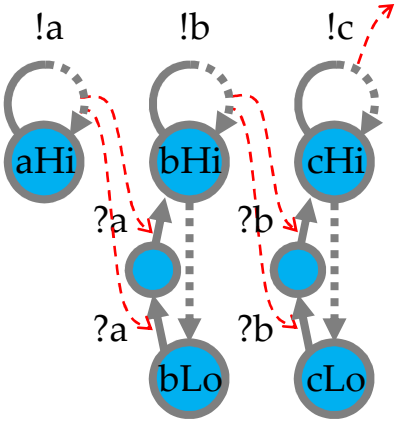
100×aHi, 1000×bLo, 1000×cLo, rates=1.0

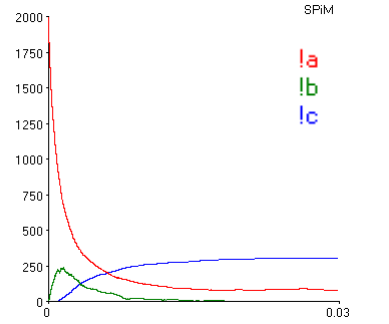
Second-Oder Regime cascade:
a signal amplifier (MAPK)

aHi > 0 ⇒ cHi = max

```
directive sample 0.03
directive plot !a; !b; !c

new a@1.0:chan new b@1.0:chan new c@1.0:chan

let Amp_hi(a:chan, b:chan) =
  do !b; Amp_hi(a,b) or delay@1.0; Amp_lo(a,b)
and Amp_lo(a:chan, b:chan) =
  ?a; ?a; Amp_hi(a,b)

run 1000 of (Amp_lo(a,b) | Amp_lo(b,c))

let A() = !a; A()
run 100 of A()
```

Write these automata in CGF and translate them to chemical reactions.





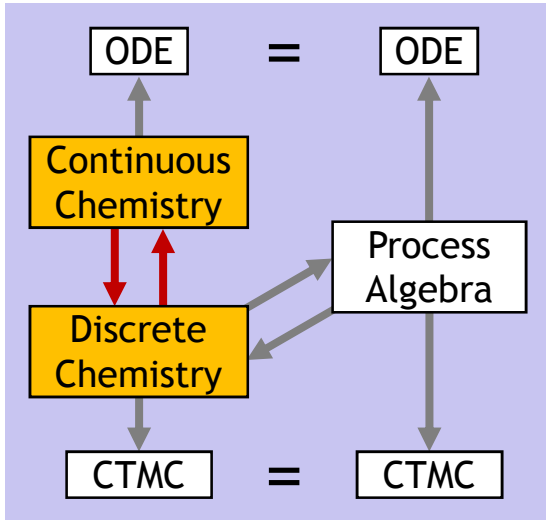
2000×aHi, 1000×bLo, 1000×cLo, rates=1.0

Zero-Oder Regime cascade:
a signal *divider!*

aHi = max ⇒ cHi = 1/3 max

```
directive sample 0.03
directive plot !a; !b; !c

new a@1.0:chan new b@1.0:chan new c@1.0:chan

let Amp_hi(a:chan, b:chan) =
  do !b; delay@1.0; Amp_hi(a,b) or delay@1.0; Amp_lo(a,b)
and Amp_lo(a:chan, b:chan) =
  ?a; ?a; Amp_hi(a,b)

run 1000 of (Amp_lo(a,b) | Amp_lo(b,c))

let A() = !a; delay@1.0; A()
run 2000 of A()
```

Discrete vs Continuous Chemistry



The "Type System" of Chemistry

The International System of Units (SI) defines the following physical units, with related derived units and constants; note that *amount of substance* is a base unit in SI, like length and time:

mol	(a base unit)	mole, unit of *amount of substance*
m	(a base unit)	meter, unit of *length*
s	(a base unit)	second, unit of *time*
$L = 0.001 \cdot m^3$		liter (volume)
$M = mol \cdot L^{-1}$		molarity (concentration of substance)
$N_A : mol^{-1} \cong 6.022 \times 10^{23}$		Avogadro's number (number of particles per amount of substance)

For a substance X:*mol*, we write [X]:*M* for the concentration of X, and $[X]^{\bullet} : M \cdot s^{-1}$ for the time derivative of the concentration.

A continuous chemical system (C,V) is a system of chemical reactions C plus a vector of initial concentrations V_X: M, one for each species X.

The rates of unary reactions have dimension s^{-1}.

The rates of binary reactions have dimension $M^{-1}s^{-1}$.

(because in both cases the rhs of an ODE should have dimension $M \cdot s^{-1}$).

Relating Concentration to Number of Molecules

For a given volume of solution V, the volumetric factor γ of dimension M^{-1} is:

$\gamma : M^{-1} = N_A V$ where N_A:mol^{-1} and V:L

#X / γ : M = concentration of X molecules

$\gamma \cdot$[X] : 1 = total number of X molecules (rounded to an integer).

The Gillespie Conversion

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

V = interaction volume

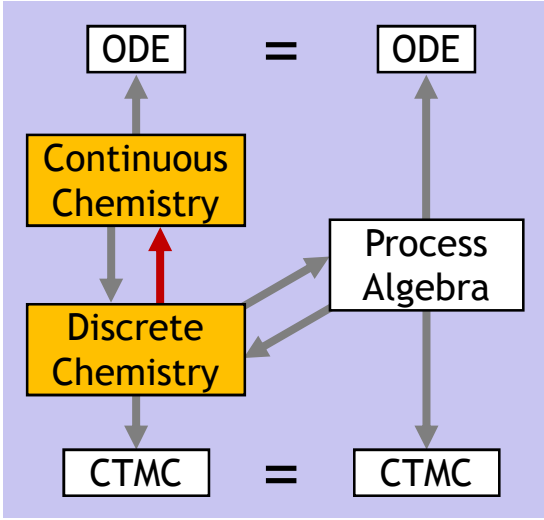
N_A = Avogadro's number

Think $\gamma = 1$

i.e. $V = 1/N_A$

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

molarity (concentration)



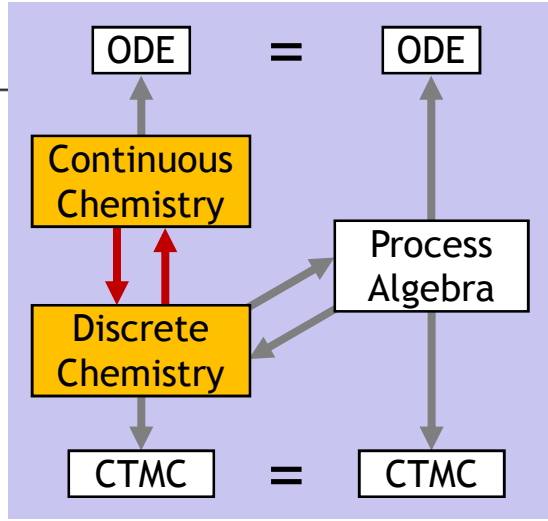
$Cont_\gamma$ and $Disc_\gamma$

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

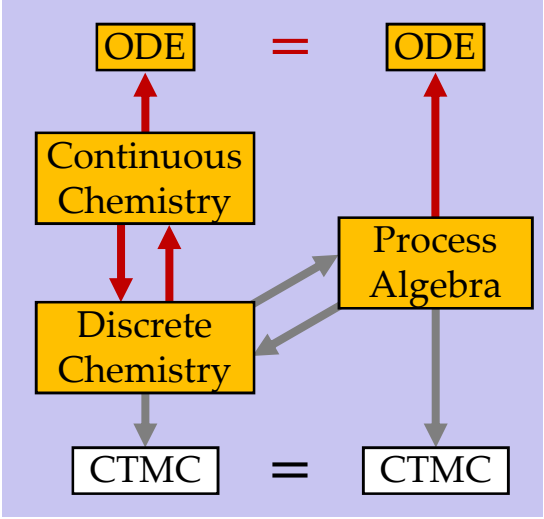
For a volumetric factor $\gamma:M^{-1}$, we define a translation $Cont_\gamma$ from a discrete chemical systems (C,P), with species X and initial molecule count $\#X_0 = \#X(P)$, to a continuous chemical systems (C,V) with initial concentration $[X]_0 = V_X$. The translation $Disc_\gamma$ is its inverse, 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 $\#X_0 = \lceil \gamma[X]_0 \rceil$	$[X]_0:M$	$X_0:mol$

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



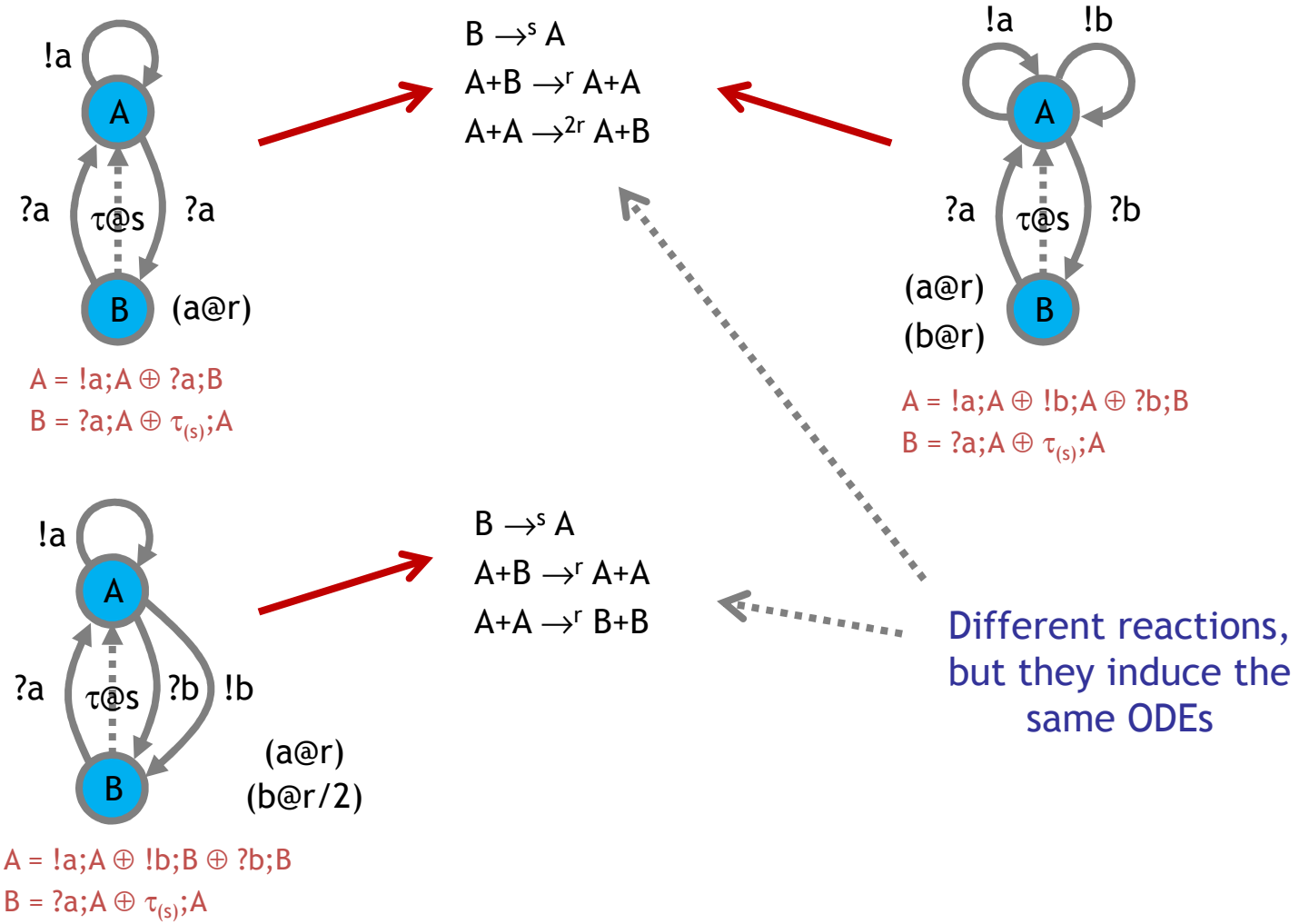
Continuous-State Semantics



Same Semantics

Could chemistry itself be that semantics?

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



From Reactions to ODEs (Law of Mass Action)

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

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

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

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

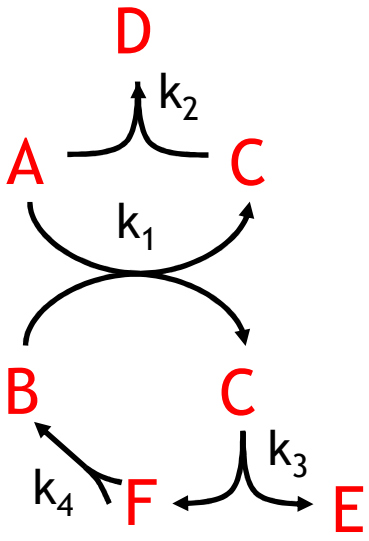
Write the coefficients by columns

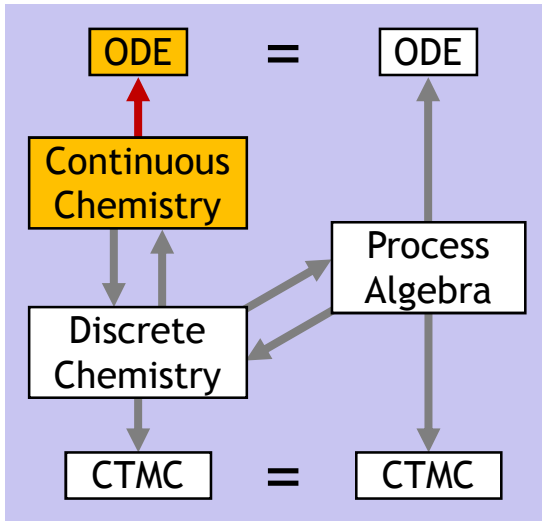
Stoichiometric Matrix

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

X





Quantity changes

Stoichiometric matrix

Rate laws

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

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

l	
l_1	$k_1[A][B]$
l_2	$k_2[A][C]$
l_3	$k_3[C]$
l_4	$k_4[F]^2$

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

Read the concentration changes from the rows

$d[A]/dt = -l_1 - l_2$

$d[B]/dt = -l_1 + l_4$

$d[C]/dt = 2l_1 - l_2 - l_3$

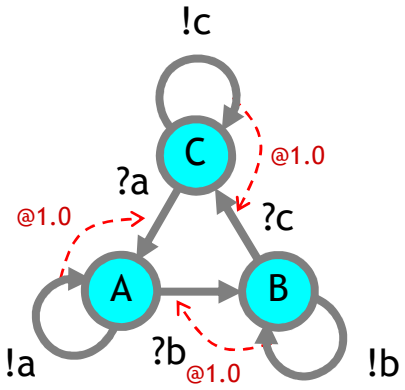
$d[D]/dt = l_2$

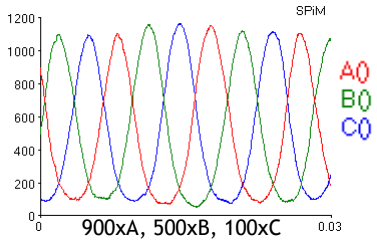
$d[E]/dt = l_3$

$d[F]/dt = l_3 - 2l_4$

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

From Processes to ODEs via Chemistry!





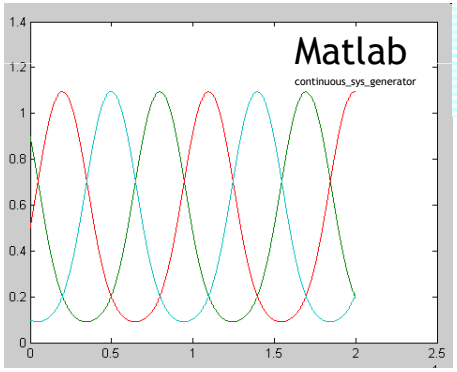

```
directive sample 0.03 1000
directive plot A(); B(); C()

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())
```

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

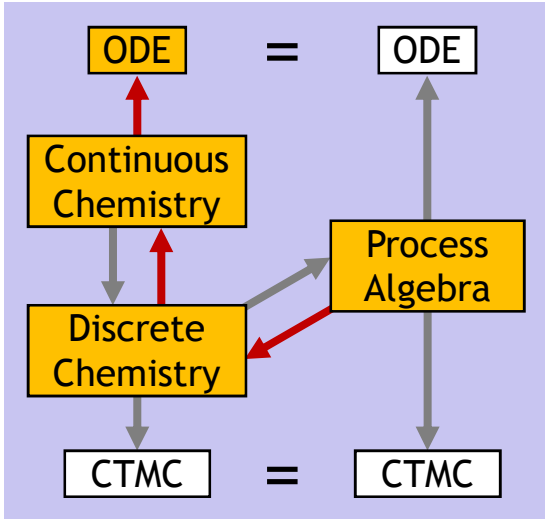
$$A+B \rightarrow^{s} B+B$$
$$B+C \rightarrow^{s} C+C$$
$$C+A \rightarrow^{s} A+A$$

$$d[A]/dt = -s[A][B]+s[C][A]$$
$$d[B]/dt = -s[B][C]+s[A][B]$$
$$d[C]/dt = -s[C][A]+s[B][C]$$

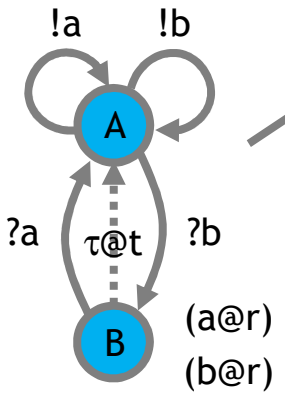
$(\gamma = 1)$




```
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
```



From Processes to ODEs via Chemistry!

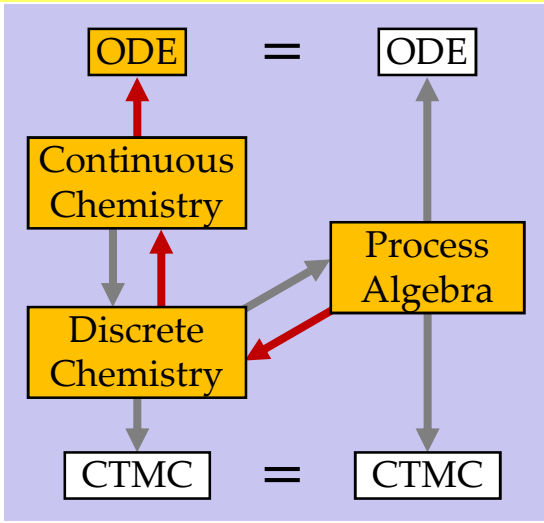


τ: $B \rightarrow^{t} A$
a: $A+B \rightarrow^{r} A+A$
b: $A+A \rightarrow^{2r} A+B$
(discrete reactions)

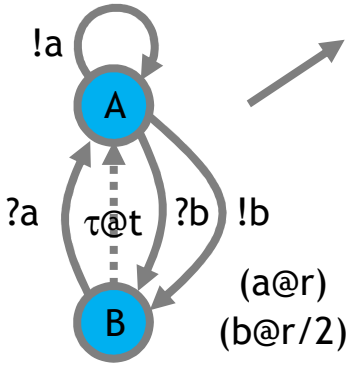
$B \rightarrow^{t} A$
$A+B \rightarrow^{r\gamma} A+A$
$A+A \rightarrow^{r\gamma} A+B$
(continuous reactions)

lose 1A at rate $r\gamma$

$d[A]/dt = t[B] + r\gamma[A][B] - r\gamma[A]^2$
$d[B]/dt = -t[B] - r\gamma[A][B] + r\gamma[A]^2$



Different chemistry but same ODEs, hence equivalent automata



τ: $B \rightarrow^{t} A$
a: $A+B \rightarrow^{r} A+A$
b: $A+A \rightarrow^{r} B+B$
(discrete reactions)

$B \rightarrow^{t} A$
$A+B \rightarrow^{r\gamma} A+A$
$A+A \rightarrow^{r\gamma/2} B+B$
(continuous reactions)

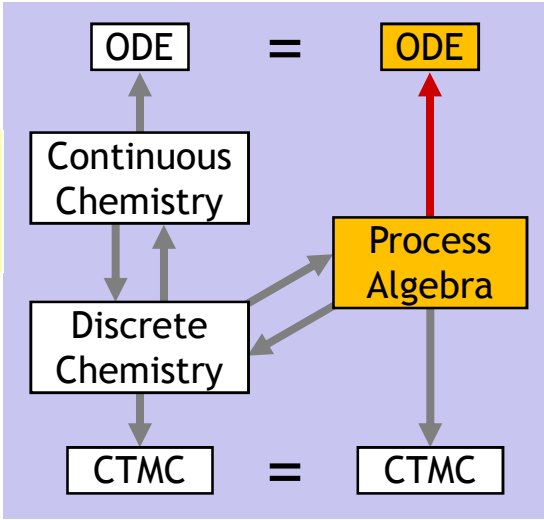
lose 2A at rate $r\gamma/2$

$d[A]/dt = t[B] + r\gamma[A][B] - r\gamma[A]^2$
$d[B]/dt = -t[B] - r\gamma[A][B] + r\gamma[A]^2$

Processes Rate Equation

Process Rate Equation for Reagents E in volume γ

$$d[X]/dt = (\Sigma(Y \in E)\ Accr_E(Y,X)\cdot[Y]) - Depl_E(X)\cdot[X]$$
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"

$Depl_E(X) =$
$\Sigma(i: E.X.i=\tau_{(r)};P)\ r +$
$\Sigma(i: E.X.i=?a_{(r)};P)\ r\gamma\cdot OutsOn_E(a) +$
$\Sigma(i: E.X.i=!a_{(r)};P)\ r\gamma\cdot InsOn_E(a)$

$Accr_E(Y, X) =$
$\Sigma(i: E.Y.i=t_{(r)};P)\ \#X(P)\cdot r +$
$\Sigma(i: E.Y.i=?a_{(r)};P)\ \#X(P)\cdot r\gamma\cdot OutsOn_E(a) +$
$\Sigma(i: E.Y.i=!a_{(r)};P)\ \#X(P)\cdot r\gamma\cdot InsOn_E(a)$

$InsOn_E(a) = \Sigma(Y \in E)\ \#\{Y.i \mid E.Y.i=?a_{(r)};P\}\cdot[Y]$
$OutsOn_E(a) = \Sigma(Y \in E)\ \#\{Y.i \mid E.Y.i=!a_{(r)};P\}\cdot[Y]$

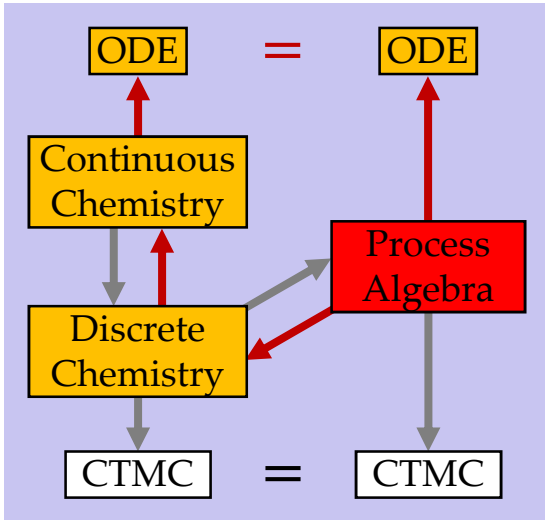
$X = \tau_{(r)};0 \rightarrow d[X]/dt = -r[X]$

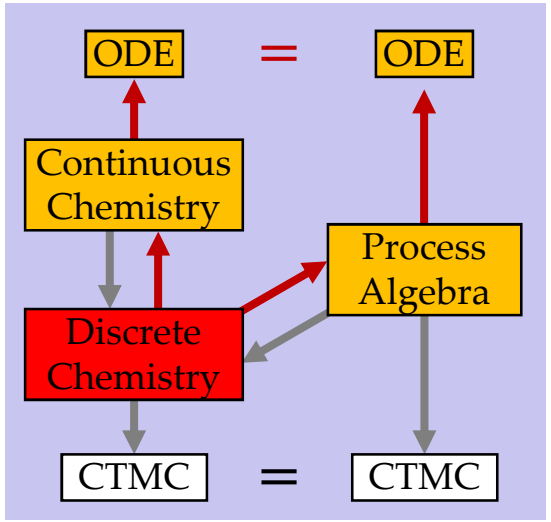
$X = ?a_{(r)};0 \rightarrow d[X]/dt = -r\gamma[X][Y]$
$Y = !a_{(r)};0 \rightarrow d[Y]/dt = -r\gamma[X][Y]$

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

Continuous State Equivalence

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

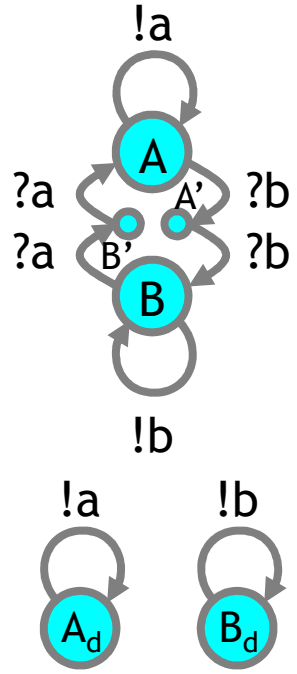


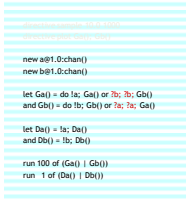


- For each E there is an E' ≈ E that is detangled (E' = Pi(Ch(E)))
- For each E in automata form there is an an E' ≈ E that is detangled and in automata form (E' = Detangle(E)).

Exercise 2

Q: What does this do?





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

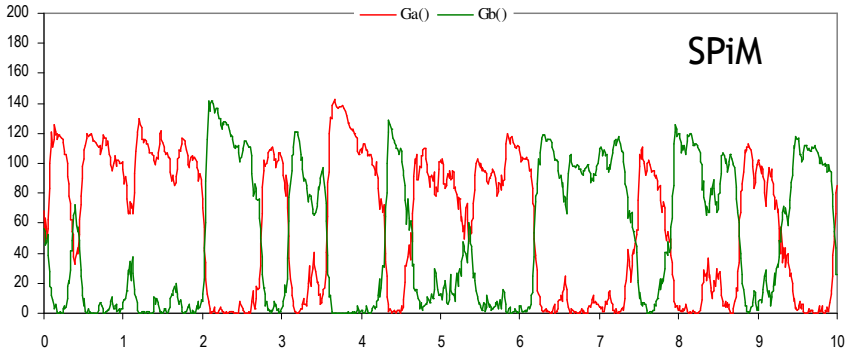
$B = !b_{(r)};B \oplus ?a;B'$ $B' = ?a;A$

$A_d = !a_{(r)};A_d$

$B_d = !b_{(r)};B_d$

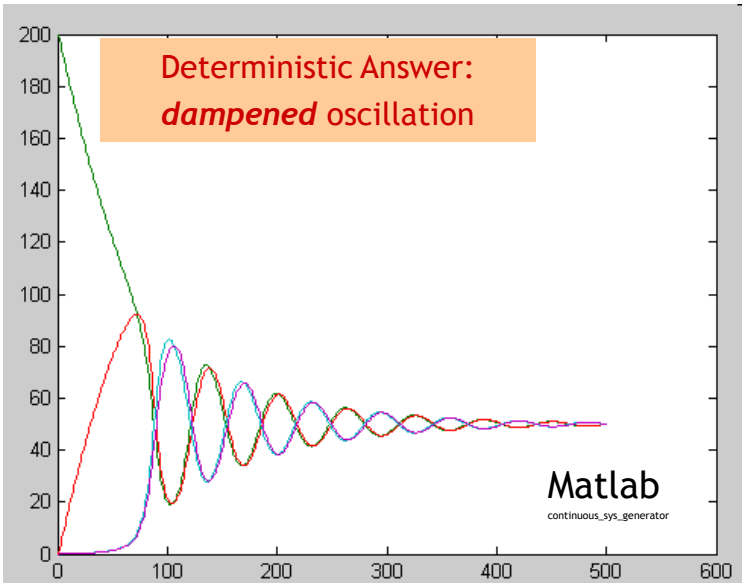
Derive the ODEs from these "Hysteric Groupies" automata. Either by going through the chemical reactions and the Law of Mass Action (easier), or directly from the Process Rate Equation.

Stochastic Answer:
robust quasi-oscillation



ODE predicts dampened oscillation, while the stochasic system keeps oscillating at max level.

Deterministic Answer:
dampened oscillation



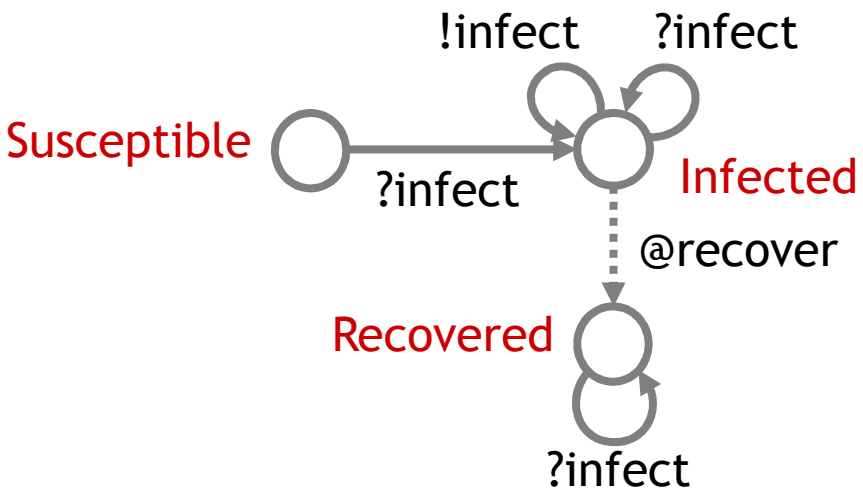
Epidemics

Non-Chemical Mass Action

Kermack, W. O. and McKendrick, A. G. "**A Contribution to the Mathematical Theory of Epidemics.**" *Proc. Roy. Soc. Lond. A* **115**, 700-721, 1927.

http://mathworld.wolfram.com/Kermack-McKendrickModel.html

Epidemics



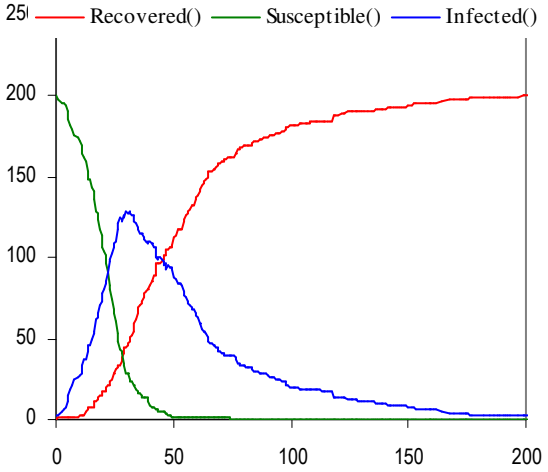

```
directive sample 500.0 1000
directive plot Recovered(); Susceptible(); Infected()

new infect @0.001:chan()
val recover = 0.03

let Recovered() =
  ?infect; Recovered()

and Susceptible() =
  ?infect; Infected()

and Infected() =
  do !infect; Infected()
  or ?infect; Infected()
  or delay@recover; Recovered()

run (200 of Susceptible() | 2 of Infected())
```

Developing the Use of Process Algebra in the Derivation and Analysis of Mathematical Models of Infectious Disease

R. Norman and C. Shankland

Department of Computing Science and Mathematics, University of Stirling, UK.
{ces,ran}@cs.stir.ac.uk

Abstract. We introduce a series of descriptions of disease spread using the process algebra WSCCS and compare the derived mean field equations with the traditional ordinary differential equation model. Even the preliminary work presented here brings to light interesting theoretical questions about the "best" way to defined the model.



ODEs

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

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

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

Differentiating Processes!

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

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

$I \rightarrow^{r} R$

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

"useless" reactions

$d[S]/dt = -t\gamma[S][I]$

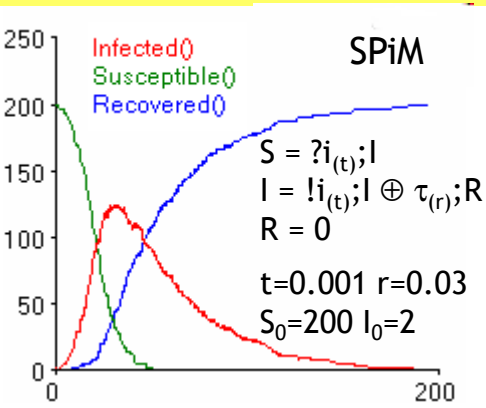
$d[I]/dt = t\gamma[S][I]-r[I]$

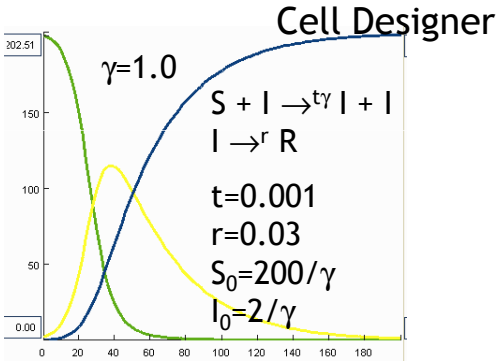
$d[R]/dt = r[I]$

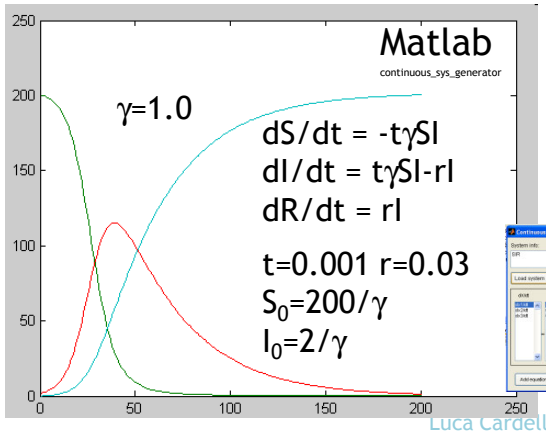
Automata produce the standard ODEs!

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

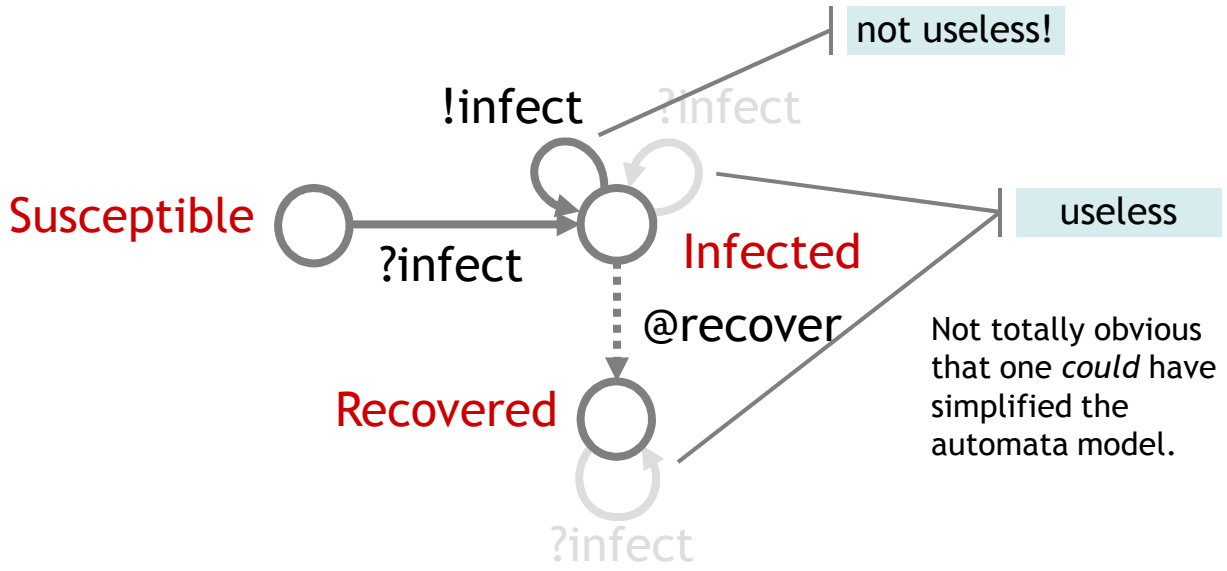
(the Kermack-McKendrick, or SIR model)







Simplified Model



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

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

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

$R = 0$

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

$I \rightarrow^{r} R$

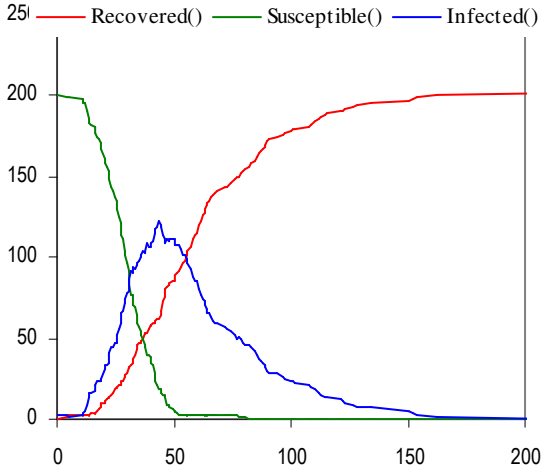
$d[S]/dt = -t\gamma[S][I]$

$d[I]/dt = t\gamma[S][I]-r[I]$

$d[R]/dt = r[I]$

Same ODE, hence equivalent automata models.

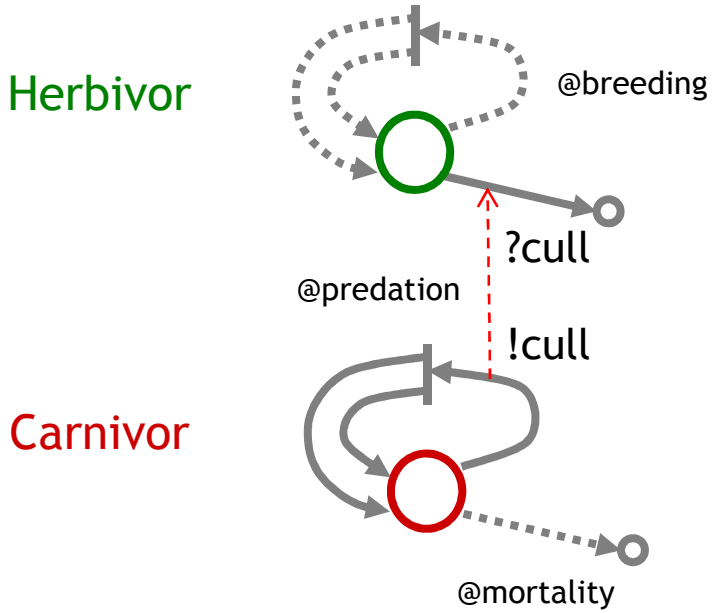
```
directive sample 500.0 1000
directive plot Recovered(); Susceptible(); Infected()

new infect @0.001:chan()
val recover = 0.03

let Recovered() =
  ()

and Susceptible() =
  ?infect; Infected()

and Infected() =
  do !infect; Infected()
  or delay@recover; Recovered()

run (200 of Susceptible() | 2 of Infected())
```



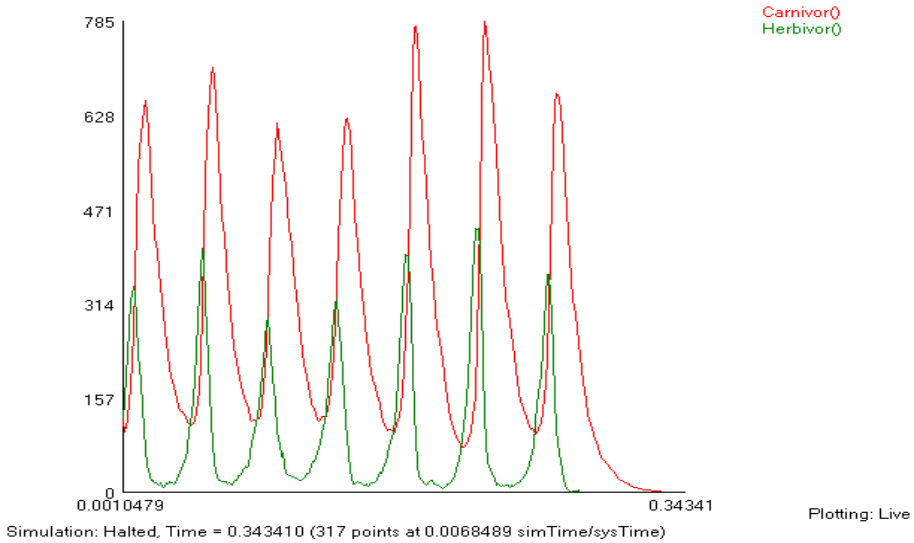
Lotka-Volterra

Unbounded Systems

Predator-Prey




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

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

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

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

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



An *unbounded state* system!

Lotka-Volterra in Matlab

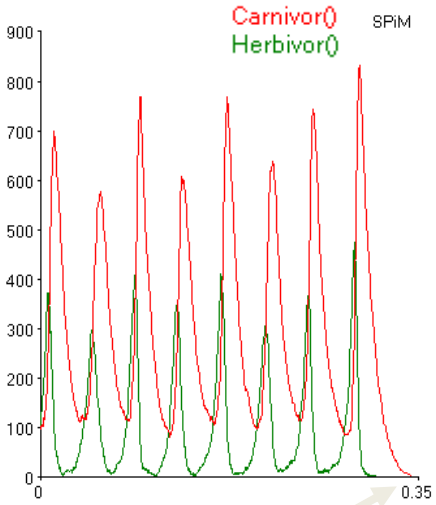
$H = \tau_b;(H|H) \oplus ?c_{(p)};0$
$C = \tau_m;0 \oplus !c_{(p)};(C|C)$
$\#H_0, \#C_0$

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

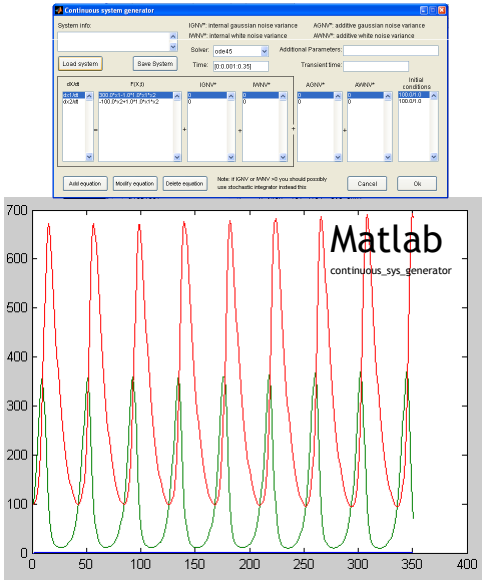
$d[H]/dt = b[H]-p\gamma[H][C]$
$d[C]/dt = -m[C]+p\gamma[H][C]$
$[H]_0 = \#H_0/\gamma$
$[C]_0 = \#C_0/\gamma$

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

```
directive sample 0.35 1000
directive plot Carnivor(); Herbivor()

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

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

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

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



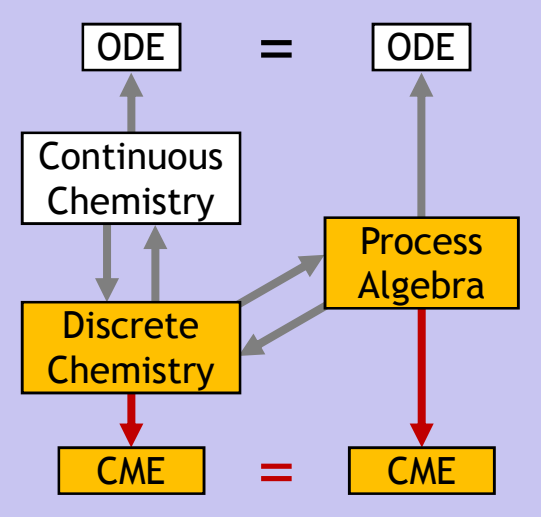
Extinction



No extinction

Which one is the “right prediction”?

Master Equation Semantics



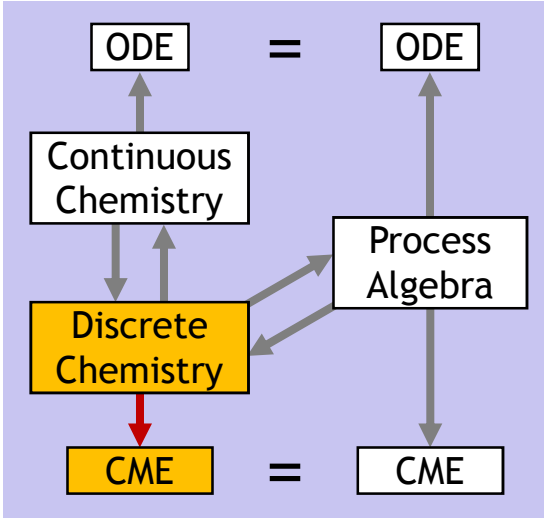
Chemical Master Equation

Chemical Master Equation for a chemical system C

$$\partial pr(s,t)/\partial t = \Sigma_{i \in 1..M}\, a_i(s-v_i)\cdot pr(s-v_i,t) - a_i(s)\cdot pr(s,t) \quad \text{for all } s \in States(C)$$

Reactions — Propensity

"The change of probability at time t of a state is:
the sum over all possible (kinds of) reactions of:
the probability at time t of each state leading to this one
times the propensity of that reaction in that state
minus the probability at time t of the current state
times the propensity of each reaction in the current state"



$s \in 1..N \to Nat$ is a *state* of the system with N chemical species

$pr(s,t) = Pr\{\chi(t)=s \mid \chi(0)=s_0\}$ is the conditional probability of the system χ being in state s at time t given that it was in state s_0 at time 0.

There are 1..M chemical reactions.

v_i is the state change caused by reaction i (as a difference)

$a_i(s) = c_i \cdot h_i(r)$ is the *propensity* of reaction i in state s, defined by a base reaction rate and a state-dependent count of the distinct combinations of reagents. (It depends on the kind of reactions.)

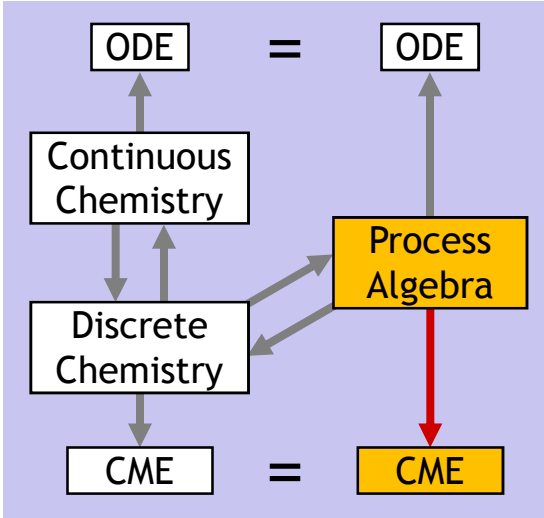
Process Algebra Master Equation

Process Master Equation for a system of reagents E

$$\partial pr(r,t)/\partial t = \Sigma_{i\in\Im} a_i(r\text{-}v_i)\cdot pr(r\text{-}v_i,t) - a_i(r)\cdot pr(r,t) \quad \text{for all } r\in States(E)$$

Interactions — Propensity

"The change of probability at time t of a state is:
the sum over all possible (kinds of) interactions of:
the probability at time t of each state leading to this one
times the propensity of that interaction in that state
minus the probability at time t of the current state
times the propensity of each interaction in the current state"



$r\in$ *species*$(E)\rightarrow$Nat is a *state* of the system

$pr(r,t) = Pr\{\chi(t)=r \mid \chi(0)=r_0\}$ is the conditional probability of the system χ being in state r at time t given that it was in state r_0 at time 0.

$\Im$ is the finite set of *possible interactions* arising from a set of reagents E. (All τ and all ?a/!a pairs in E)

v_i is the state change caused by interaction i (as a difference)

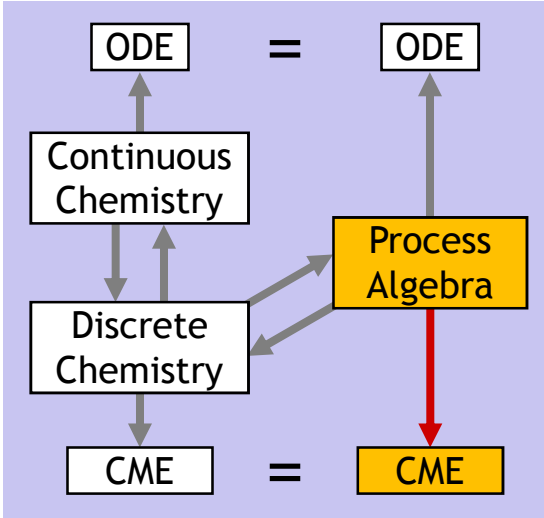
$a_i(r) = r_i\cdot h_i(r)$ is the *propensity* of interaction i in state r, defined by a base rate of interaction and a state-dependent count of the distinct combinations of reagents. (It depends on the kind of interaction.)

… details

Process Master Equation for Reagents E

$$\partial pr(r,t)/\partial t \;=\; \Sigma_{i\in\Im}\, a_i(r\text{-}v_i)\cdot pr(r\text{-}v_i,t) \;-\; a_i(r)\cdot pr(r,t) \qquad \text{for all } r\in States(E)$$

$pr(p,t) = Pr\{\mathbf{S}(t)=p \mid \mathbf{S}(0)=p_0\}$ is the conditional probability of the system being in state p (a multiset of molecules) at time t given that it was in state p_0 at time 0.



$\Im = \{\{X.i\}\ s.t.\ E.X.i = \tau_{(r)};Q\} \cup$
$\{\{X.i, Y.j\}\ s.t.\ E.X.i = ?n_{(r)};Q \text{ and } E.Y.j = !n_{(r)};R\}$
is the set of possible interactions in E

v_i is the *state change* caused by an interaction $i\in\Im$.

$v_i = -X+Q$ if $i = \{X.i\}\ s.t.\ E.X.i = \tau_{(r)};Q$

$v_i = -X-Y+Q+R$ if $i = \{X.i, Y.j\}\ s.t.\ E.X.i = ?n_{(r)};Q$ and $E.Y.j = !n_{(r)};R$

a_i is the *propensity* of interaction i in state p. Here $p^{\#X}$ is the number of X in p.

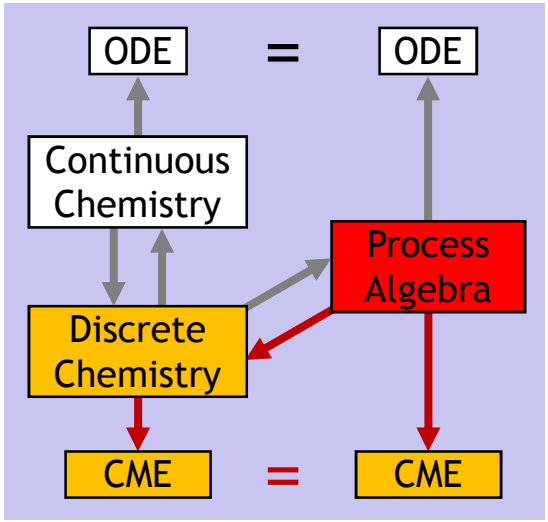
$a_i(p) = r\cdot p^{\#X}$ if $i = \{X.i\}\ s.t.\ E.X.i = \tau_{(r)};Q$

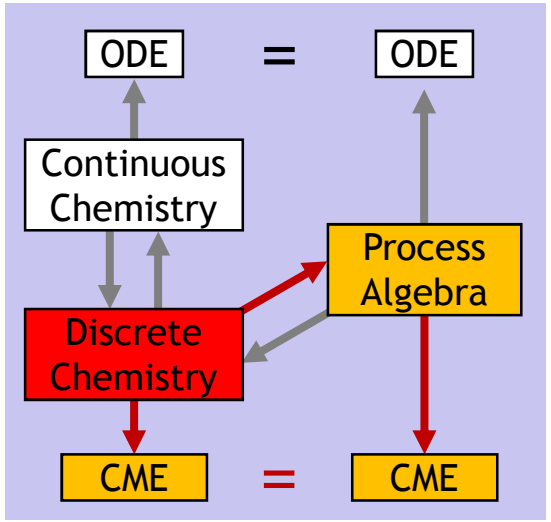
$a_i(p) = r\cdot p^{\#X}\cdot p^{\#Y}$ if $i = \{X.i, Y.j\}\ s.t.\ X\neq Y$ and $E.X.i = ?a_{(r)};Q$ and $E.Y.j = !a_{(r)};R$

$a_i(p) = r\cdot p^{\#X}\cdot(p^{\#X}-1)$ if $i = \{X.i, X.j\}\ s.t.\ E.X.i = ?a_{(r)};Q$ and $E.X.j = !a_{(r)};R$

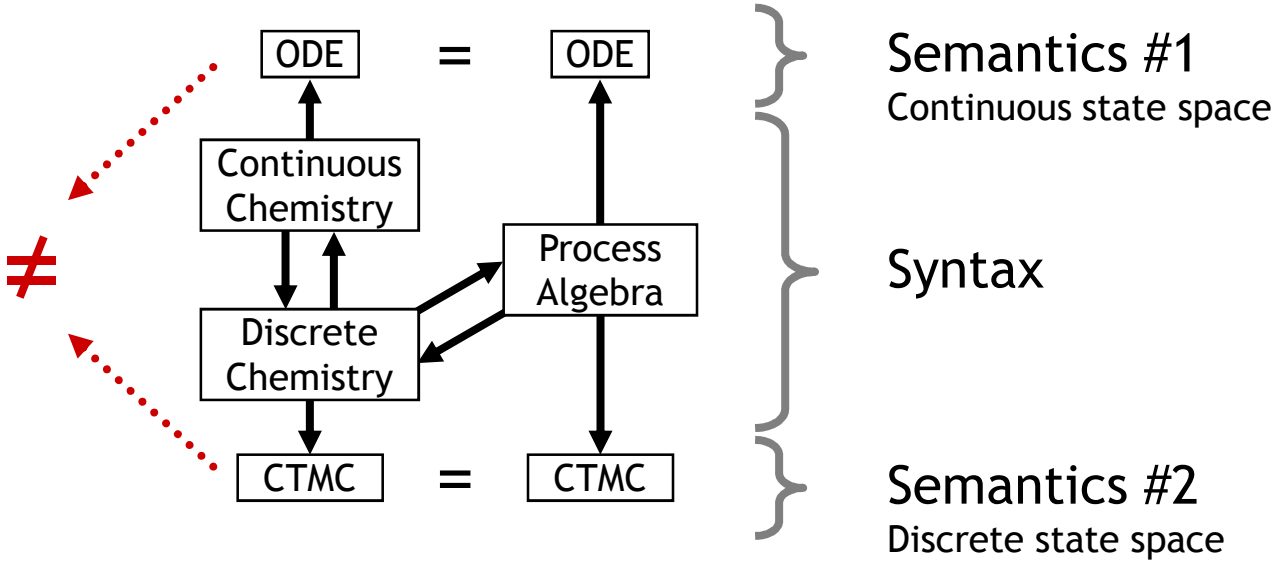
Equivalence of Master Equations

- Def: ≈ is equivalence of derived Master Equations (they are identical).
- Thm: E ≈ Ch(E)
- Thm: C ≈ Pi(C)





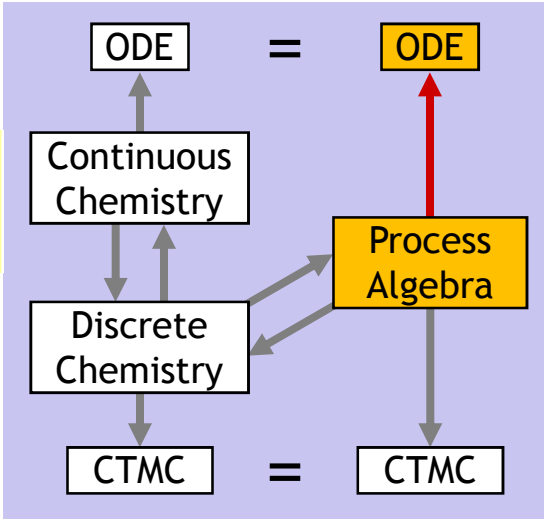
GMA ≠ CME


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

Processes to GMA Directly

Process Rate Equation for Reagents E in volume γ

$$d[X]/dt = (\Sigma(Y \in E)\ Accr_E(Y,X)\cdot[Y]) - Depl_E(X)\cdot[X]$$
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"

$Depl_E(X) =$
$\Sigma(i: E.X.i=\tau_{(r)};P)\ r +$
$\Sigma(i: E.X.i=?a_{(r)};P)\ r\gamma\cdot OutsOn_E(a) +$
$\Sigma(i: E.X.i=!a_{(r)};P)\ r\gamma\cdot InsOn_E(a)$

$Accr_E(Y, X) =$
$\Sigma(i: E.Y.i=t_{(r)};P)\ \#X(P)\cdot r +$
$\Sigma(i: E.Y.i=?a_{(r)};P)\ \#X(P)\cdot r\gamma\cdot OutsOn_E(a) +$
$\Sigma(i: E.Y.i=!a_{(r)};P)\ \#X(P)\cdot r\gamma\cdot InsOn_E(a)$

$InsOn_E(a) = \Sigma(Y \in E)\ \#\{Y.i \mid E.Y.i=?a_{(r)};P\}\cdot[Y]$
$OutsOn_E(a) = \Sigma(Y \in E)\ \#\{Y.i \mid E.Y.i=!a_{(r)};P\}\cdot[Y]$

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

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

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

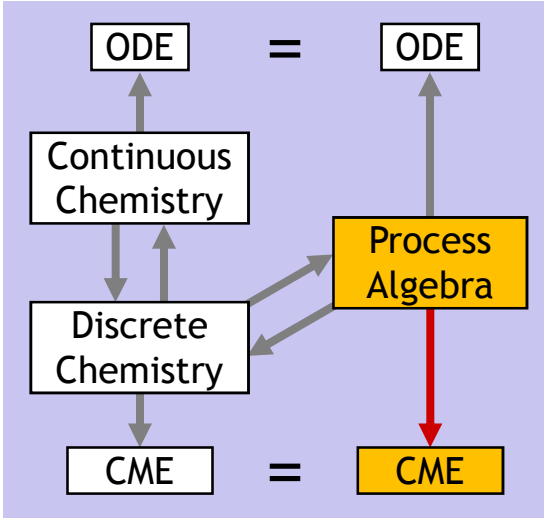
Process Algebra Master Equation

Process Master Equation for a system of reagents E

$$\partial pr(r,t)/\partial t = \Sigma_{i\in\Im}\, a_i(r\text{-}v_i)\cdot pr(r\text{-}v_i,t) - a_i(r)\cdot pr(r,t) \quad \text{for all } r\in States(E)$$

Interactions — Propensity

"The change of probability at time t of a state is:
the sum over all possible (kinds of) interactions of:
the probability at time t of each state leading to this one
times the propensity of that interaction in that state
minus the probability at time t of the current state
times the propensity of each interaction in the current state"



$r\in$ *species*$(E)\rightarrow$Nat is a *state* of the system

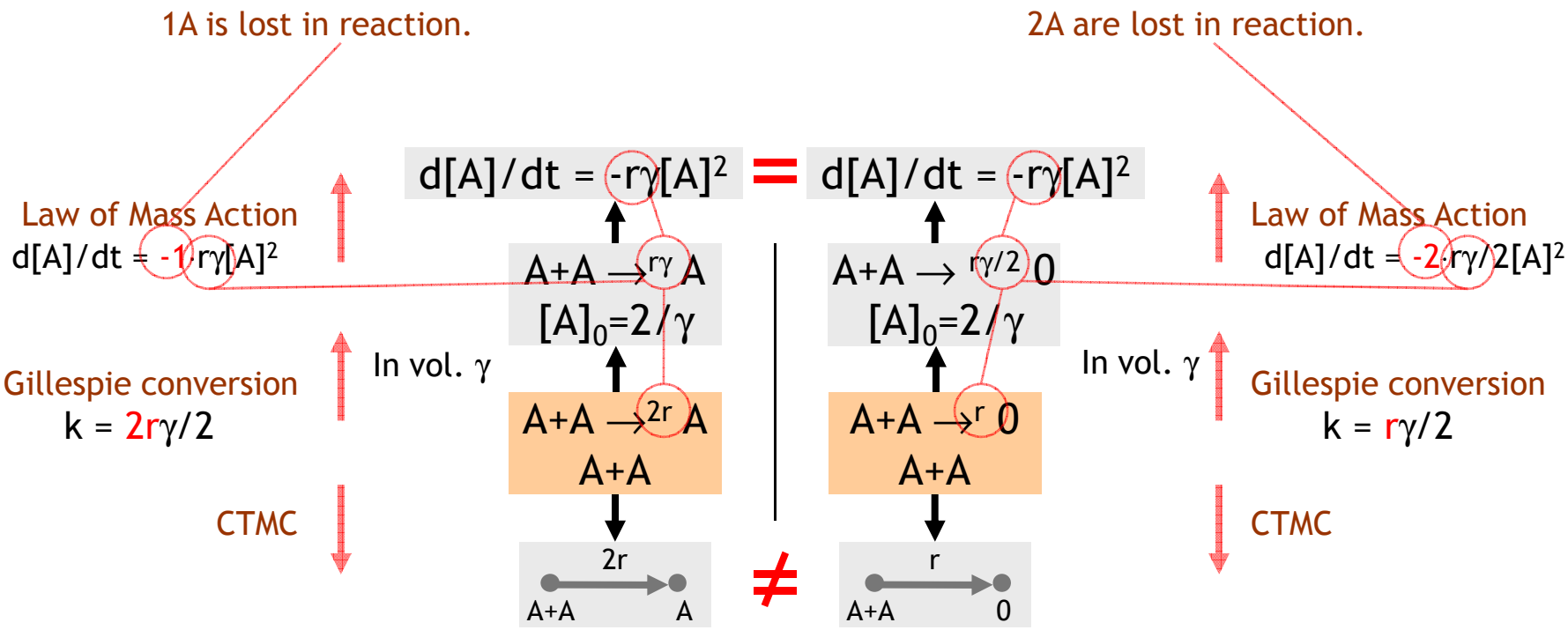
$pr(r,t) = Pr\{\chi(t)=r \mid \chi(0)=r_0\}$ is the conditional probability of the system χ being in state r at time t given that it was in state r_0 at time 0.

$\Im$ is the finite set of *possible interactions* arising from a set of reagents E. (All τ and all ?a/!a pairs in E)

v_i is the state change caused by interaction i (as a difference)

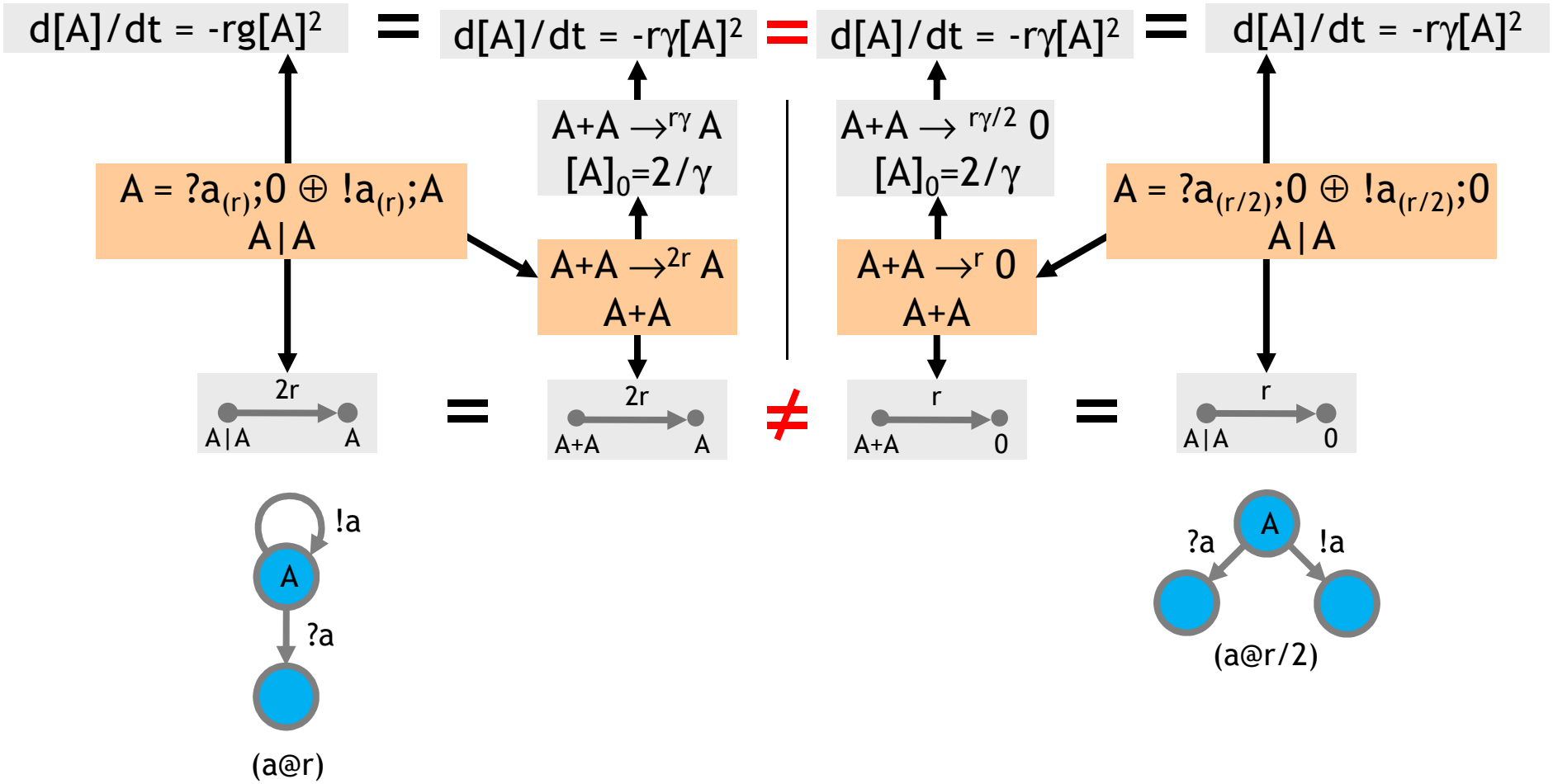
$a_i(r) = r_i\cdot h_i(r)$ is the *propensity* of interaction i in state r, defined by a base rate of interaction and a state-dependent count of the distinct combinations of reagents. (It depends on the kind of interaction.)

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

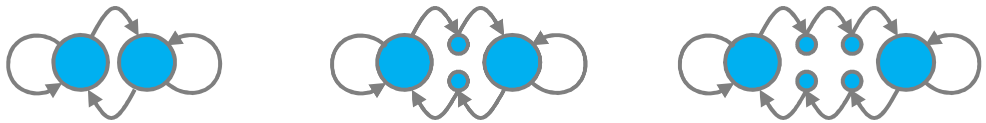


(For conservation of mass, consider instead $A+A \rightarrow^{2r} A+B$ vs. $A+A \rightarrow^{r} B+B$)

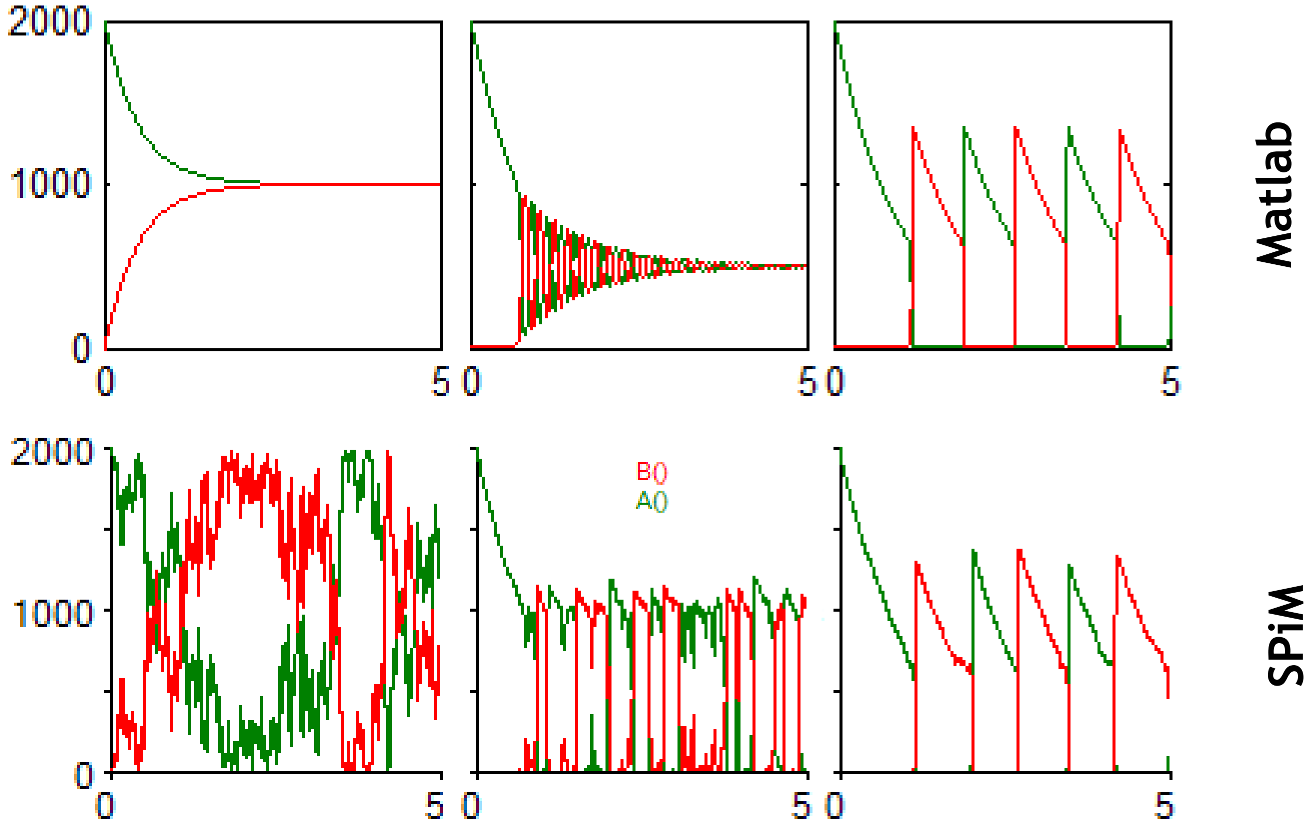
$A+A \rightarrow^{2r} A$ =? $A+A \rightarrow^{r} 0$



Continuous vs. Discrete Groupies



(all with doping)



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

```
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; 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())
```

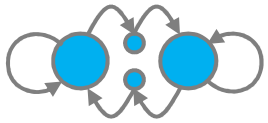
```
directive sample 5.0 1000
directive plot B(); A()

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

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

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

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

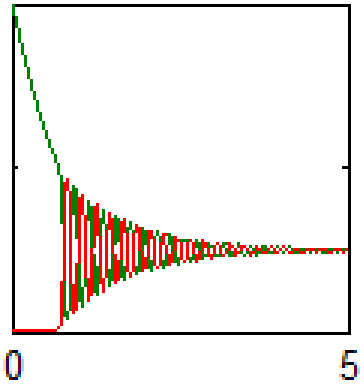
```
Groupe ODEs - Groupies.mat

[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
B dx3/dt=x3*x2-x1*x3-x3+x2, 0.0
B' dx4/dt=x1*x3-x1*x4+x3-x4, 0.0
```

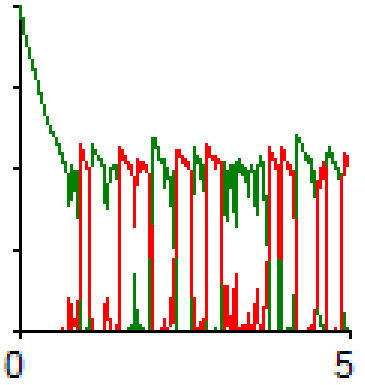
```
Groupe ODEs - Groupies Hysteric 2.mat

[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.

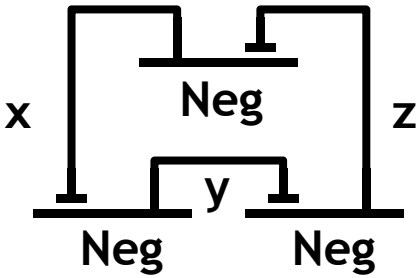


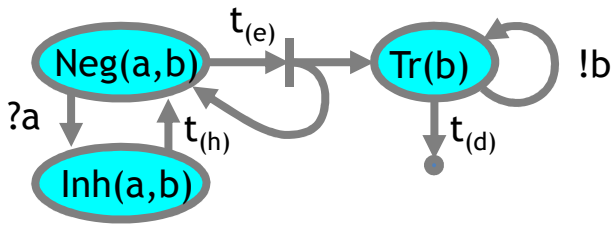
The 4 states are almost never equally occupied.

And Yet It Moves

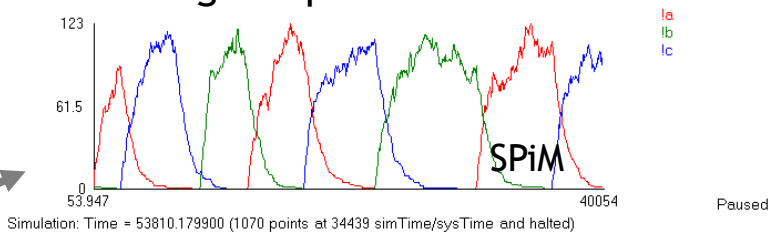
R.Blossey, L.Cardelli, A.Phillips: **Compositionality, Stochasticity and Cooperativity in Dynamic Models of Gene Regulation** (HFSP Journal)

The Repressilator





A fine stochastic oscillator over a wide range of parameters.



Parametric representation

$Neg(a,b) = ?a;\ Inh(a,b) \oplus \tau_e;\ (Tr(b) \mid Neg(a,b))$

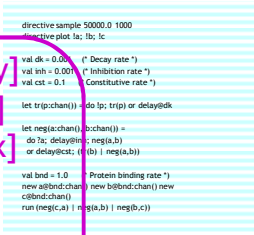
$Inh(a,b) = \tau_h;\ Neg(a,b)$

$Tr(b) = !b;\ Tr(b) \oplus \tau_g;\ 0$

$Neg(x_{(r)},y_{(r)}) \mid Neg(y_{(r)},z_{(r)}) \mid Neg(z_{(r)},x_{(r)})$

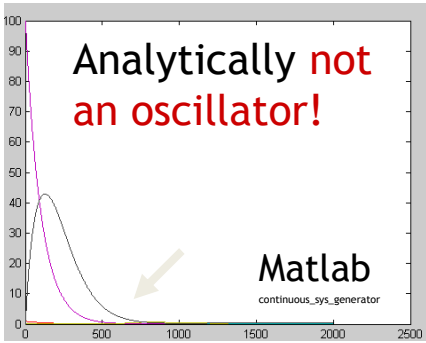
$Neg/x,y \to^{e} Tr/y + Neg/x,y$
$Neg/y,z \to^{e} Tr/z + Neg/y,z$
$Neg/z,x \to^{e} Tr/x + Neg/z,x$
$Tr/x + Neg/x,y \to^{r} Tr/x + Inh/x,y$
$Tr/y + Neg/y,z \to^{r} Tr/y + Inh/y,z$
$Tr/z + Neg/z,x \to^{r} Tr/z + Inh/z,x$
$Inh/x,y \to^{h} Neg/x,y$
$Inh/y,z \to^{h} Neg/y,z$
$Inh/z,x \to^{h} Neg/z,x$
$Tr/x \to^{g} 0$
$Tr/y \to^{g} 0$
$Tr/z \to^{g} 0$
$Neg/x,y + Neg/y,z + Neg/z,x$

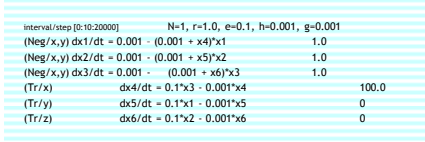
$d[Neg/x,y]/dt = -r[Tr/x][Neg/x,y] + h[Inh/x,y]$
$d[Neg/y,z]/dt = -r[Tr/y][Neg/y,z] + h[Inh/y,z]$
$d[Neg/z,x]/dt = -r[Tr/z][Neg/z,x] + h[Inh/z,x]$
$d[Inh/x,y]/dt = r[Tr/x][Neg/x,y] - h[Inh/x,y]$
$d[Inh/y,z]/dt = r[Tr/y][Neg/y,z] - h[Inh/y,z]$
$d[Inh/z,x]/dt = r[Tr/z][Neg/z,x] - h[Inh/z,x]$
$d[Tr/x]/dt = e[Neg/z,x] - g[Tr/x]$
$d[Tr/y]/dt = e[Neg/x,y] - g[Tr/y]$
$d[Tr/z]/dt = e[Neg/y,z] - g[Tr/z]$



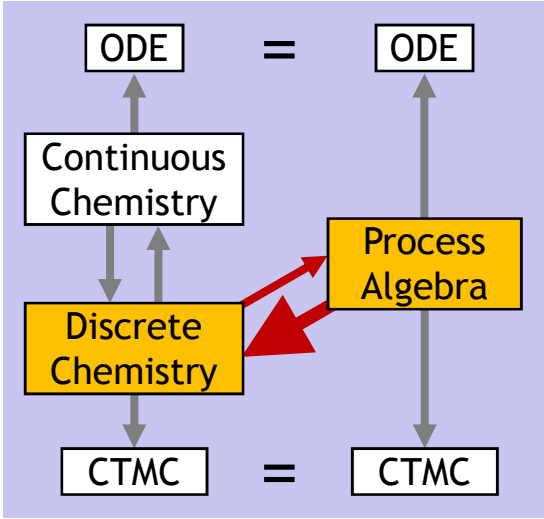
simplifying (N is the quantity of each of the 3 gates)

$d[Neg/x,y]/dt = hN - (h+r[Tr/x])[Neg/x,y]$
$d[Neg/y,z]/dt = hN - (h+r[Tr/y])[Neg/y,z]$
$d[Neg/z,x]/dt = hN - (h+r[Tr/z])[Neg/z,x]$
$d[Tr/x]/dt = e[Neg/z,x] - g[Tr/x]$
$d[Tr/y]/dt = e[Neg/x,y] - g[Tr/y]$
$d[Tr/z]/dt = e[Neg/y,z] - g[Tr/z]$





Model Compactness



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

$X_0 = ?a_{(r)};X_1$
$X_1 = ?a_{(r)};X_2$
$X_2 = ?a_{(r)};X_0$
$Y_0 = !a_{(r)};Y_1$
$Y_1 = !a_{(r)};Y_2$
$Y_2 = !a_{(r)};Y_0$

$Ch(E_3)$

$a_{00}: X_0+Y_0 \rightarrow^r X_1+Y_1$
$a_{01}: X_0+Y_1 \rightarrow^r X_1+Y_2$
$a_{02}: X_0+Y_2 \rightarrow^r X_1+Y_0$
$a_{10}: X_1+Y_0 \rightarrow^r X_2+Y_1$
$a_{11}: X_1+Y_1 \rightarrow^r X_2+Y_2$
$a_{12}: X_1+Y_2 \rightarrow^r X_2+Y_0$
$a_{20}: X_2+Y_0 \rightarrow^r X_0+Y_1$
$a_{21}: X_2+Y_1 \rightarrow^r X_0+Y_2$
$a_{22}: X_2+Y_2 \rightarrow^r X_0+Y_0$

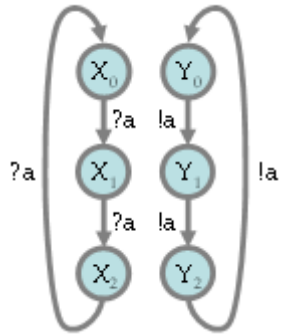
StoichiometricMatrix($Ch(E_3)$)

	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

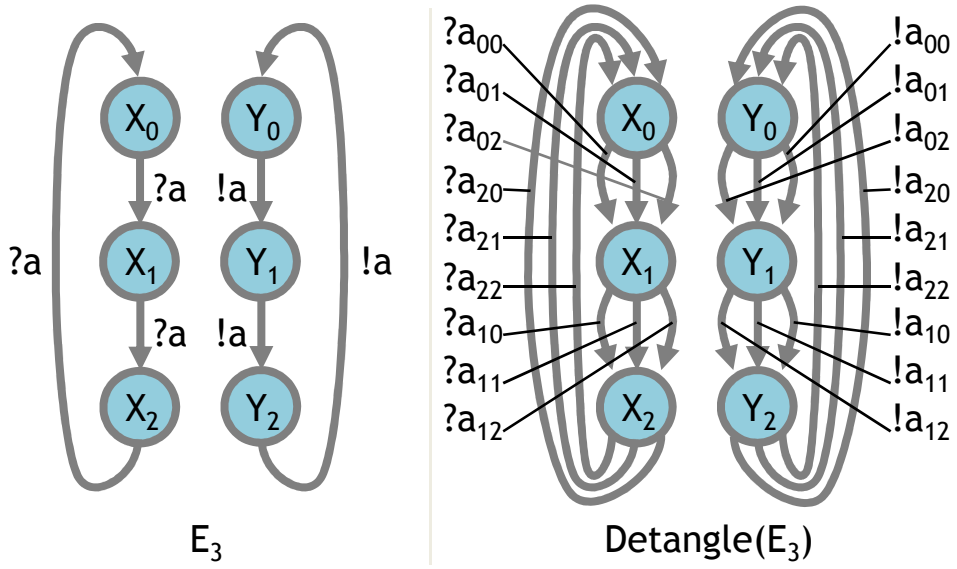
$ODE(E_3)$

$d[X_0]/dt = -r[X_0][Y_0] - r[X_0][Y_1] - r[X_0][Y_2] + r[X_2][Y_0] + r[X_2][Y_1] + r[X_2][Y_2]$
$d[X_1]/dt = -r[X_1][Y_0] - r[X_1][Y_1] - r[X_1][Y_2] + r[X_0][Y_0] + r[X_0][Y_1] + r[X_0][Y_2]$
$d[X_2]/dt = -r[X_2][Y_0] - r[X_2][Y_1] - r[X_2][Y_2] + r[X_1][Y_0] + r[X_1][Y_1] + r[X_1][Y_2]$
$d[Y_0]/dt = -r[X_0][Y_0] - r[X_1][Y_0] - r[X_2][Y_0] + r[X_0][Y_2] + r[X_1][Y_2] + r[X_2][Y_2]$
$d[Y_1]/dt = -r[X_0][Y_1] - r[X_1][Y_1] - r[X_2][Y_1] + r[X_0][Y_0] + r[X_1][Y_0] + r[X_2][Y_0]$
$d[Y_2]/dt = -r[X_0][Y_2] - r[X_1][Y_2] - r[X_2][Y_2] + r[X_0][Y_1] + r[X_1][Y_1] + r[X_2][Y_1]$

E_3



Entangled vs detangled

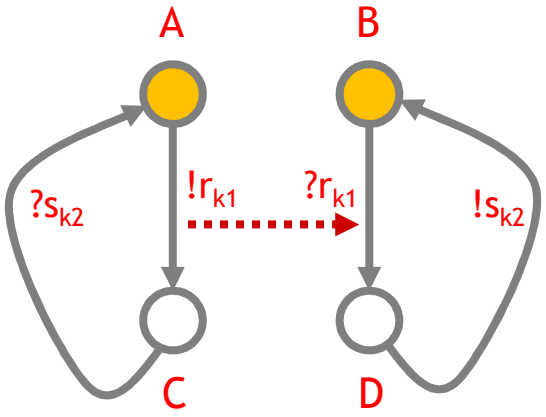


E_3 Detangle(E_3)

(closely related to
Pi(Ch(E_3)))

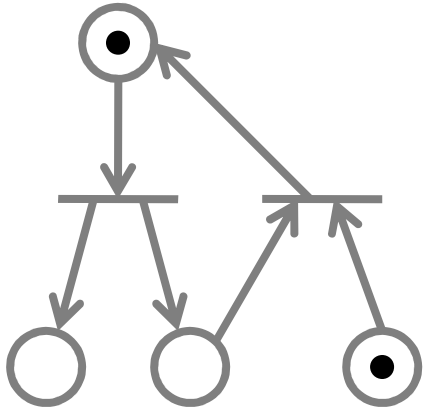
Model Maintenance

- Biology (unlike much of chemistry) is combinatorial
  - Biochemical systems have many regular repeated components
  - Components interact and combine in complex combinatorial ways
  - Components have local state
  - A biochemical system is vastly more compact that its potential state space



Or

- One may have to expand the state space during analysis, but must not do it during description

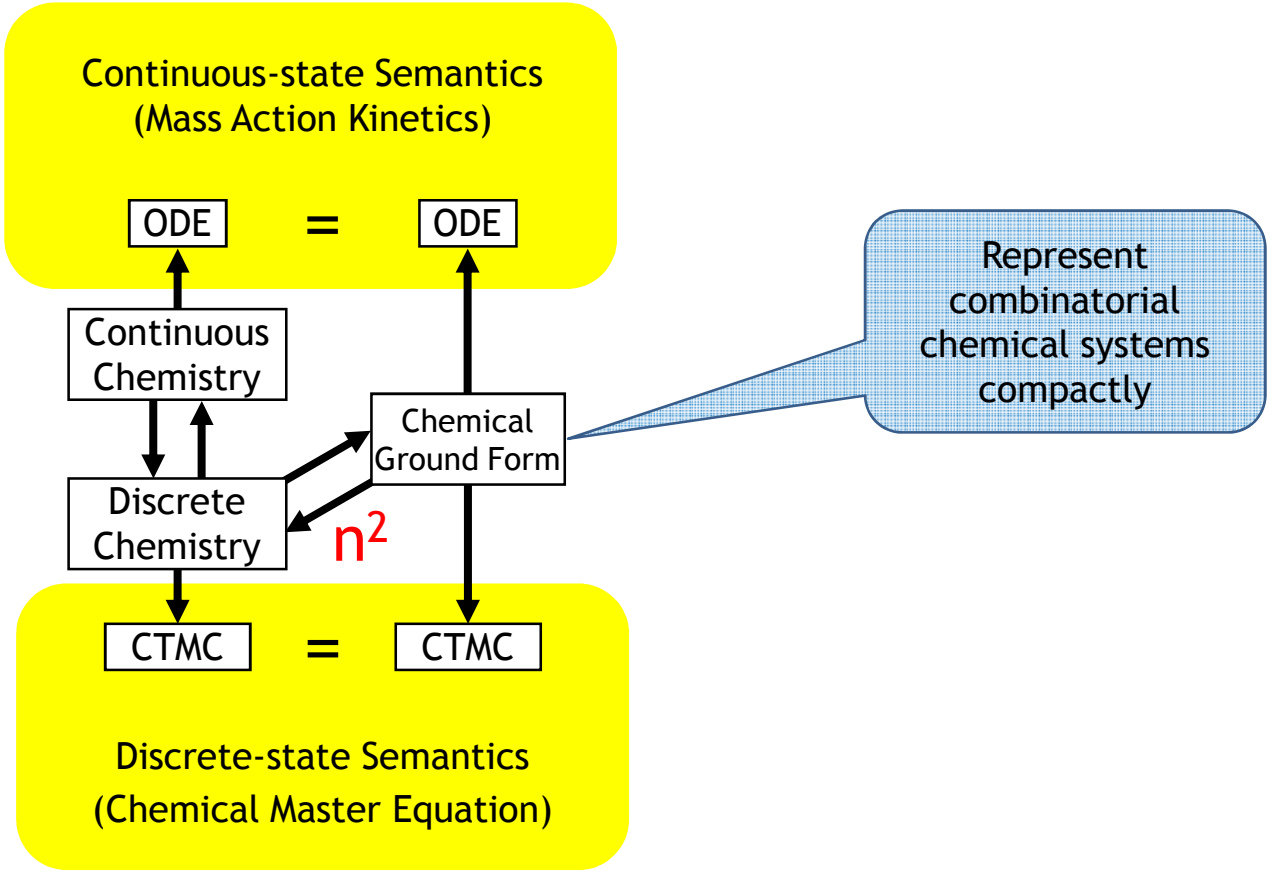


Or …

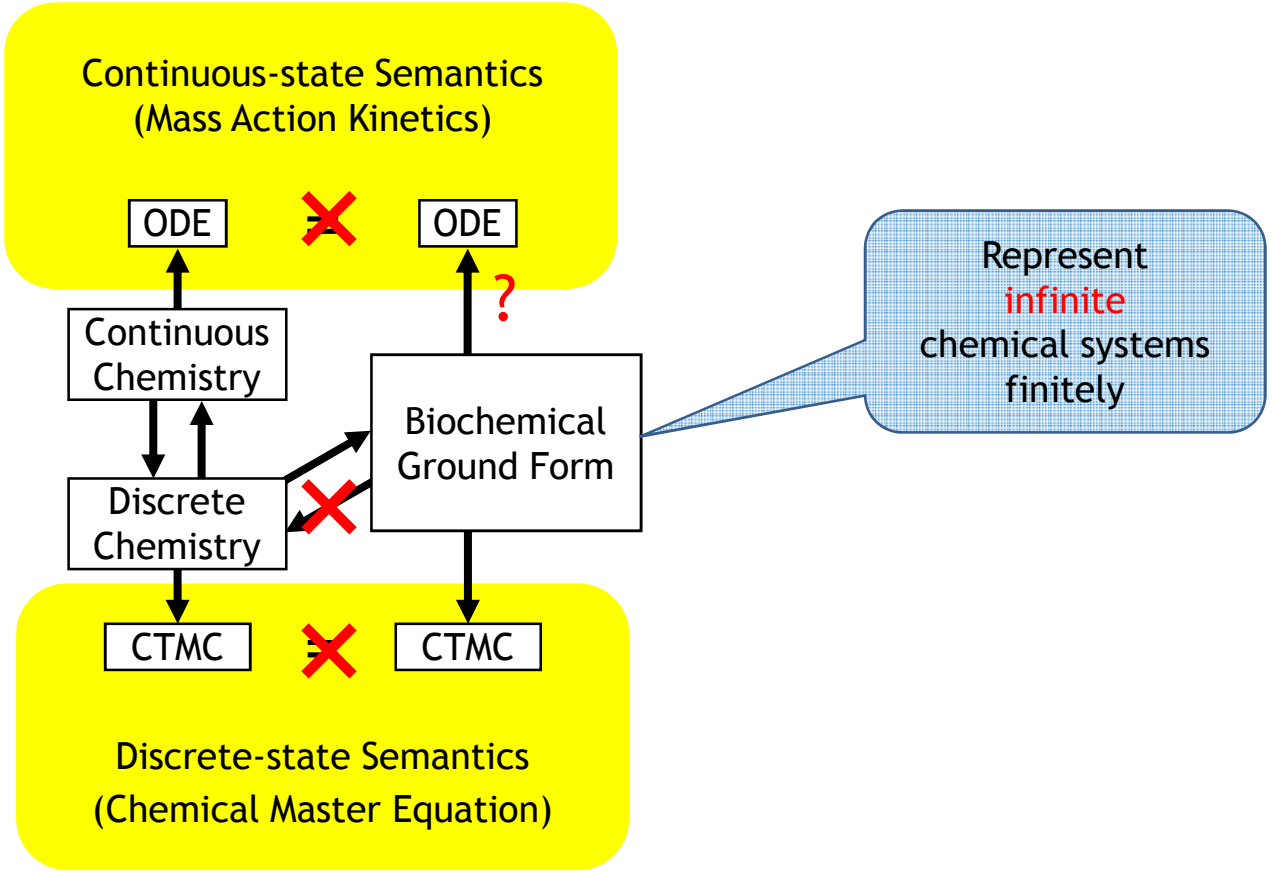
- There is a good way:
  - Describe biochemical systems compositionally
  - Each component with its own state and interactions
  - … as Nature intended…

Chemistry *and Beyond*

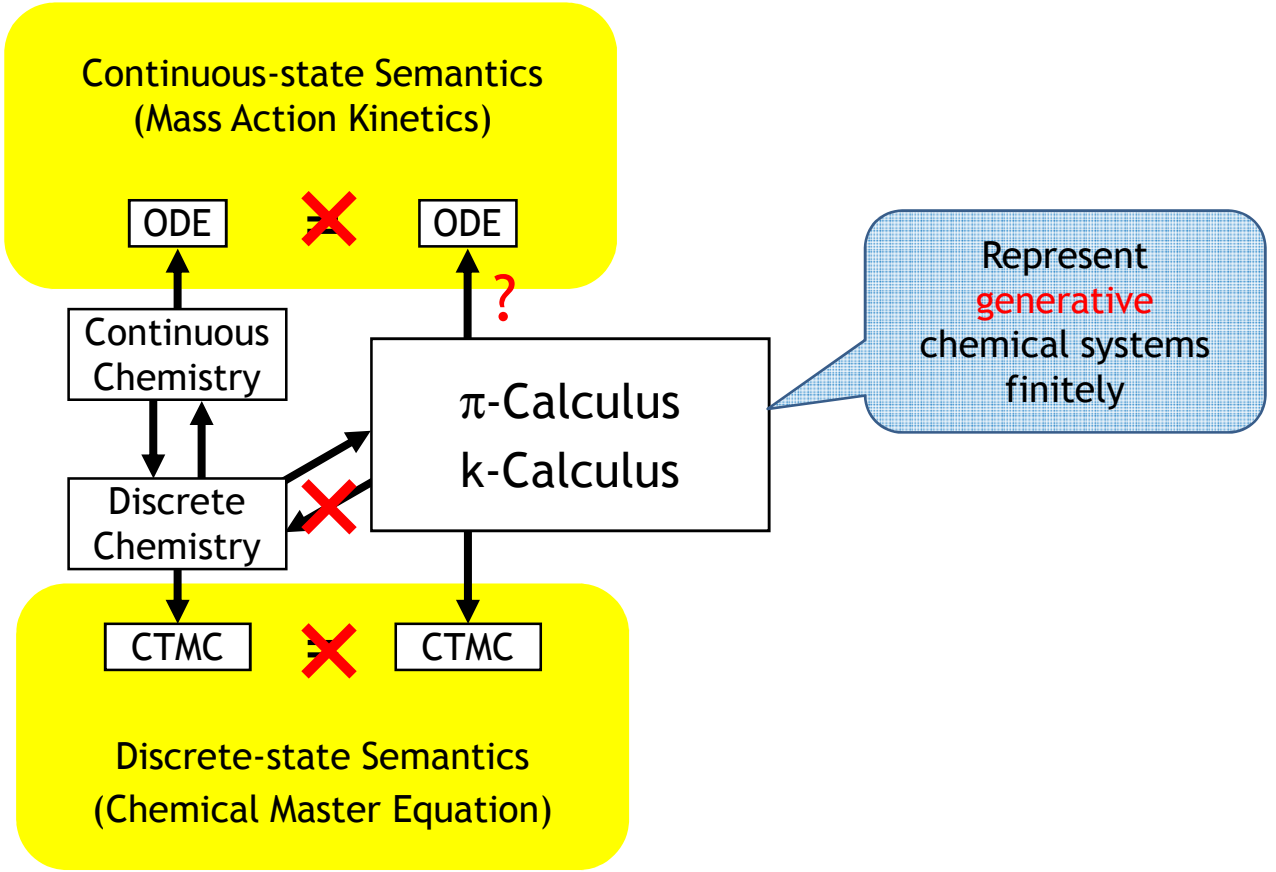
Process Algebra is 'Bigger' than Chemistry



Process Algebra is 'Bigger' than Chemistry

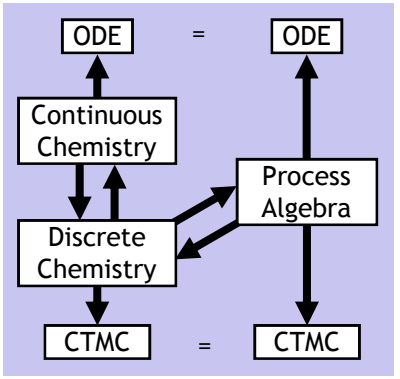


Process Algebra is 'Bigger' than Chemistry



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, …)