Chemistry *and Beyond*

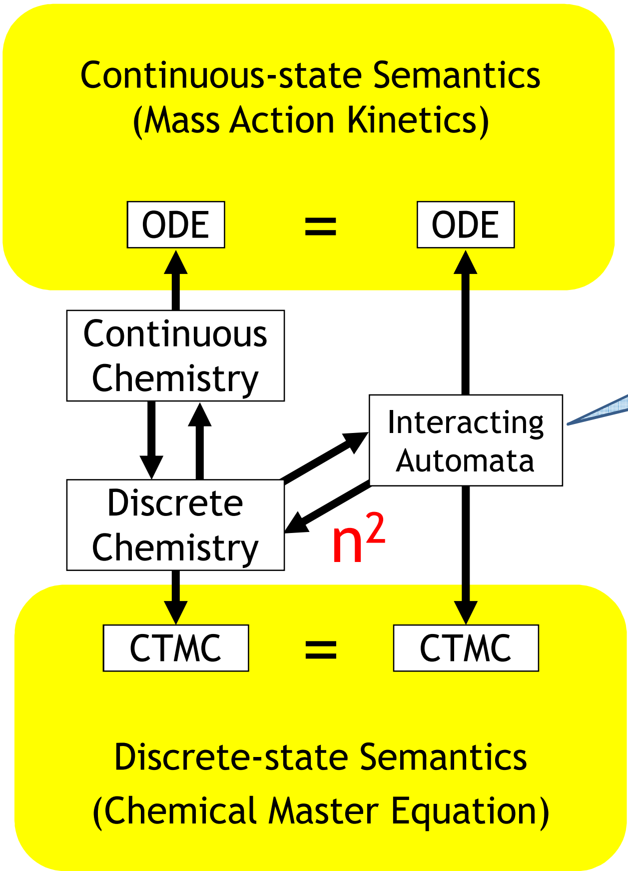
Luca Cardelli

Microsoft Research

Open Lectures for PhD Students in Computer Science
Warsaw 2009-05-08

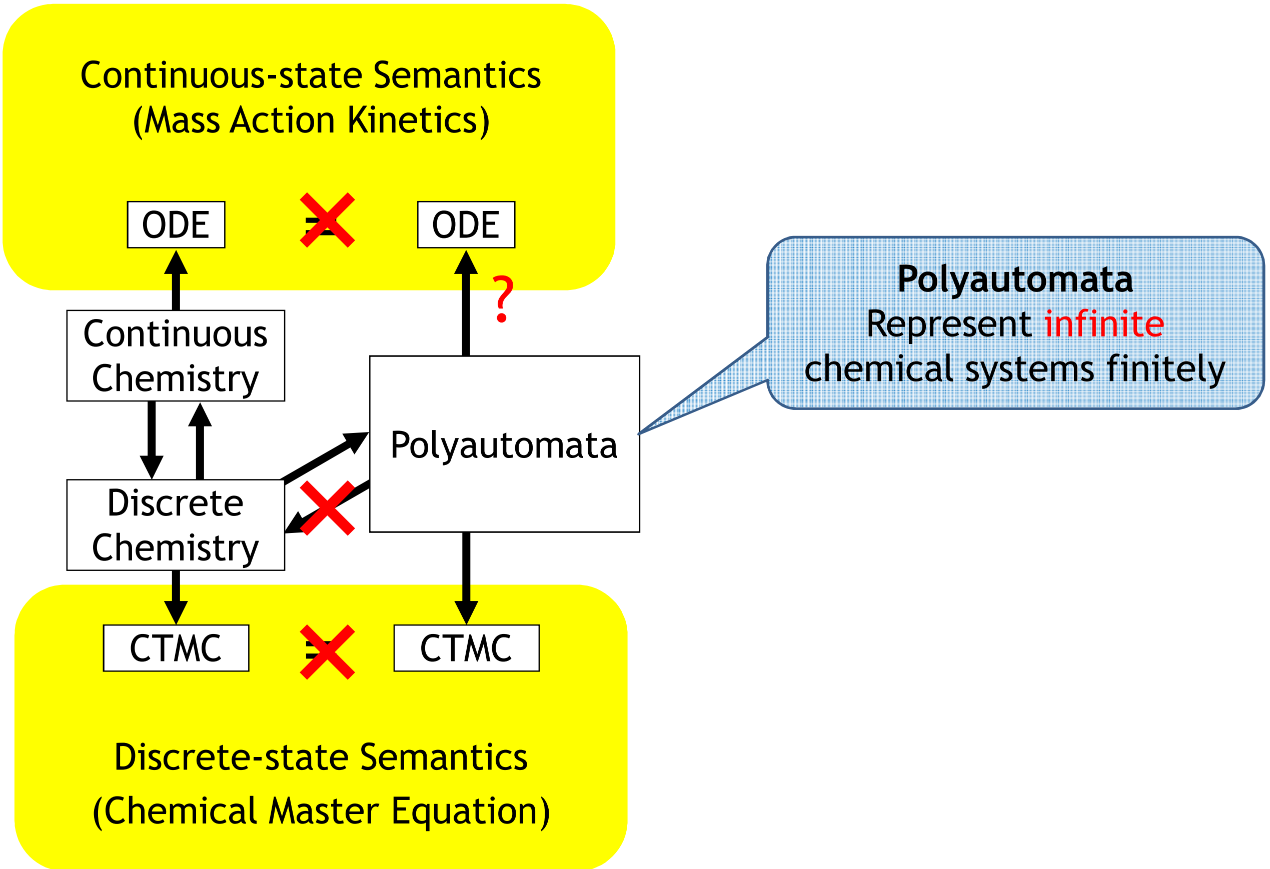
http://lucacardelli.name

Process Algebra is ‘Bigger’ than Chemistry

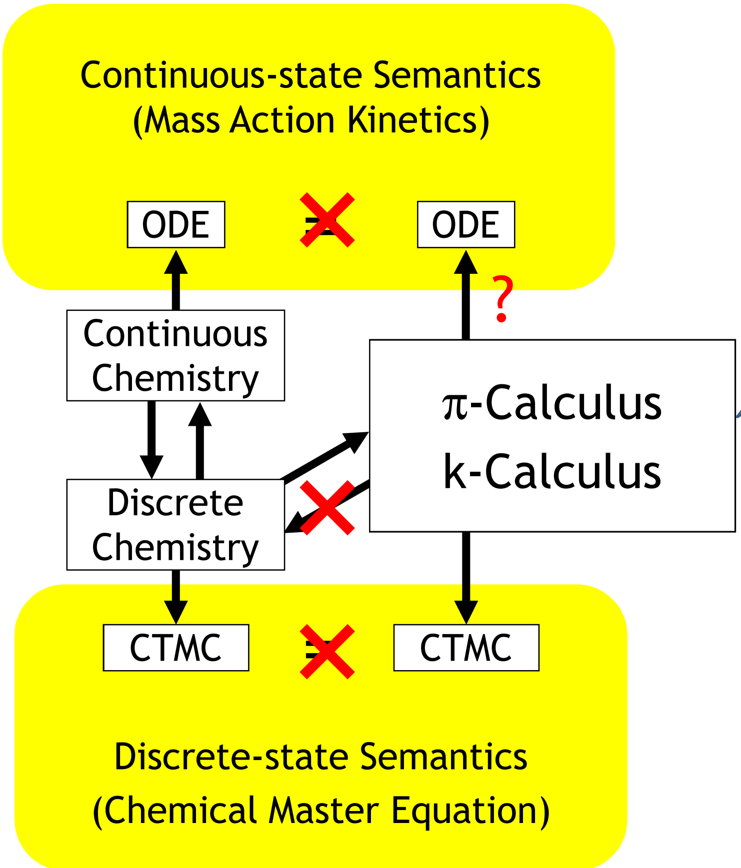


Interacting Automata
Represent finite chemical systems compactly

Process Algebra is 'Bigger' than Chemistry



Process Algebra is 'Bigger' than Chemistry



Full process algebra
Represent generative chemical systems finitely

On the Computational Power of Biochemistry

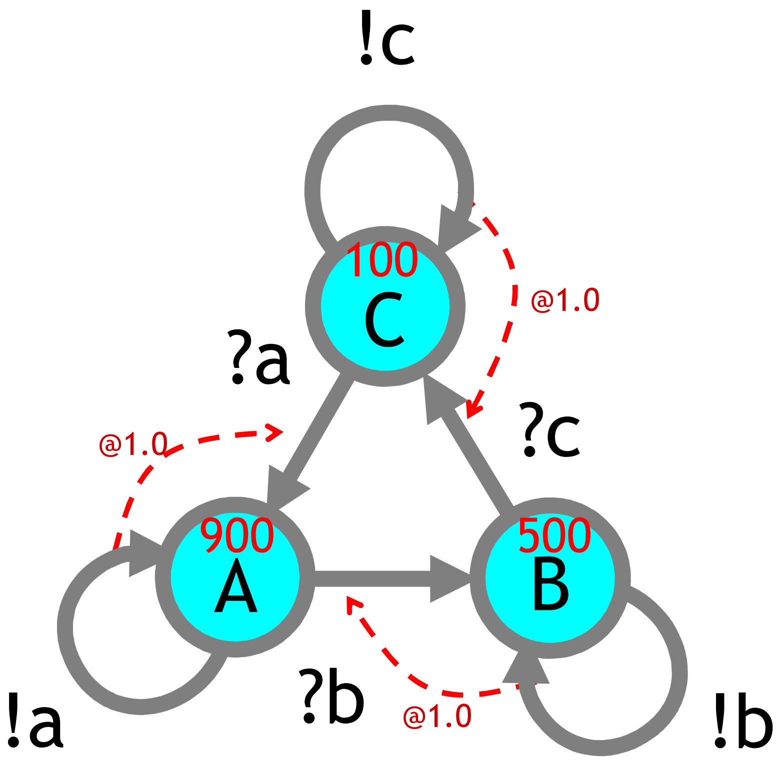
joint work with

Gianluigi Zavattaro

University of Bologna

in: Algebraic Biology '08

Can this program terminate?



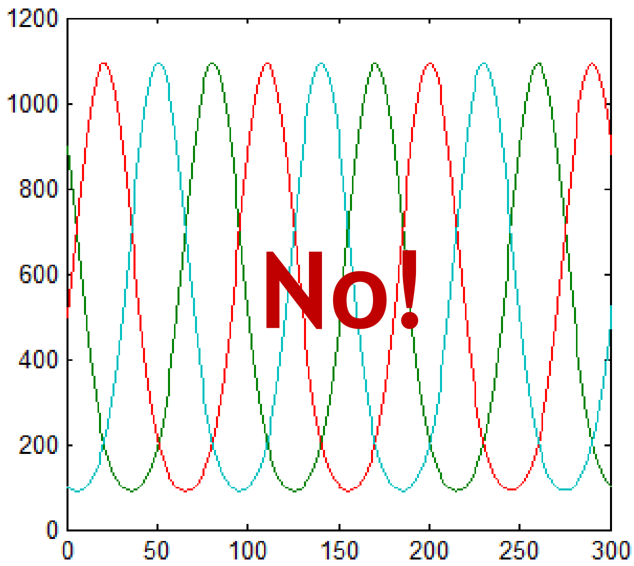
b: $A+B \rightarrow B+B$

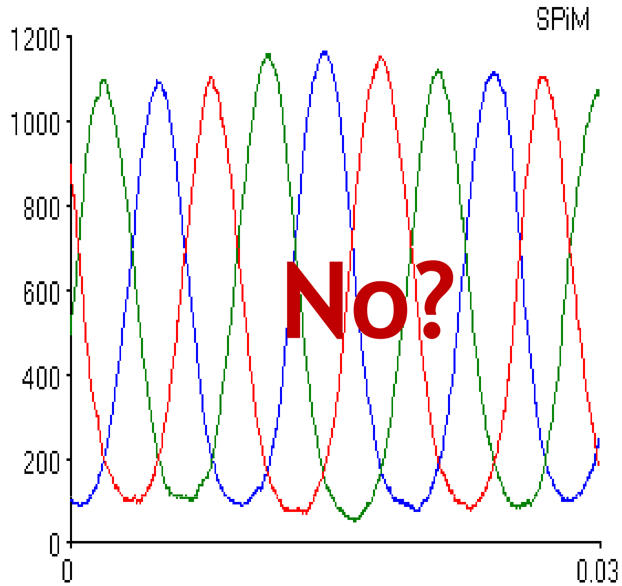
c: $B+C \rightarrow C+C$

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

$900A + 500B + 100C$

“Experimantal evidence”





Continuous-State Simulation

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

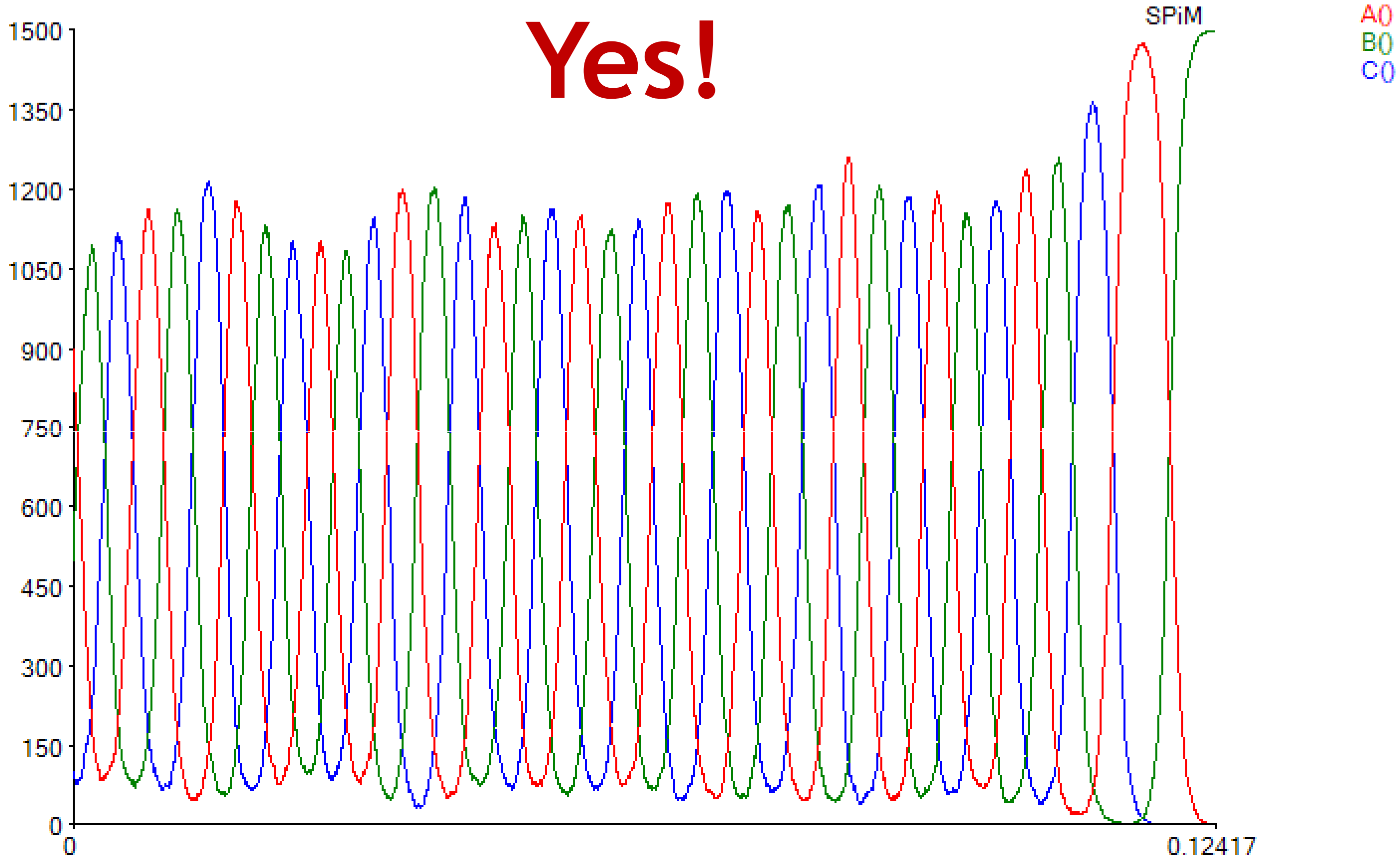
Discrete-State Simulation

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

new a@1.0:chan new b@1.0:chan new
c@1.0:chan
let A() = do !a;A() or ?b; B()
and B() = do !b;B() or ?c; C()
and C() = do !c;C() or ?a; A()

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

But in a longer simulation...



Discrete-State Simulation

0A + 1500B + 0C

Is termination (possible death) decidable in Chemistry?

- Termination in Chemistry is at least *subtle*. Is it decidable?

- Three equivalent definitions of "basic chemistry":
    - FSRN: Finite Stochastic Reaction Networks
      (finite systems of stochastic chemical reactions)
    - CGF (Interacting Automata): our process algebra.
    - Place-Transition (stochastic) Petri nets.

- Surprising answer: termination in basic chemistry is *decidable!*
    - (Soloveichik et al. *Computation with Finite Stochastic Chemical Reaction Networks*. In Nat. Computing. 2008) by reduction to a decidable problem in Petri Nets (reachability).

- Hence, basic chemistry cannot compute!
    - By Turing's theorem, termination for a universal computer is undecidable.
    - Hence basic chemistry is not Turing-complete.
    - (Although the full story is more subtle and interesting: stochastic chemistry can *approximate* Turing machines to arbitrary precision.)

Can Biochemistry Compute?

- Chemistry cannot compute; is that true of Biochemistry? Not necessarily.

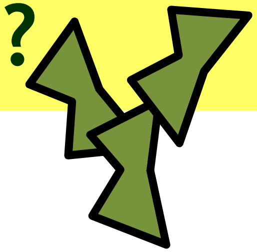
- Although Chemistry (FSRNs) encompasses huge complexity (e.g. chaotic systems), it is in fact unable to express (finitely) virtually *any* biological system of interest!! (and many non-biological ones)
  - So, how have people managed so far? By manipulating awkward infinite collections of chemical reactions or ODEs.

- The language of Biochemistry is intrinsically more powerful than the language of Chemistry: it can represent finitely systems that Chemistry can't. Since it is more powerful it can be Turing complete (and it is).

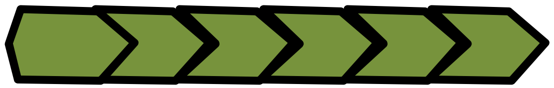
- What is the language of Biochemistry? Until recently, there wasn't one. Historically the first language used in that sense has been stochastic π-calculus, then (a bit more appropriately) k-calculus.

- The most elementary such language is "polyautomata".

C.vs. BioC. What's the Difference?



Consider linear polymerization:



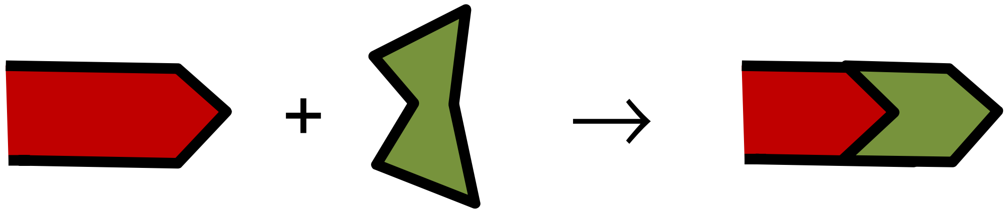
The "chemical program" for polymerization:

$P_0 + M \rightarrow P_1$
$P_1 + M \rightarrow P_2$
$P_2 + M \rightarrow P_3$
$P_3 + M \rightarrow P_4$
....

- an infinite (non-)program
- an infinite set of species
- an infinite set of ODEs

$P_{10757} + M \rightarrow P_{10758}$
Such specificity is unreal.

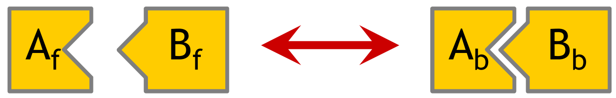
But "nature's program" for polymerization has to fit in the genome, so it cannot be infinite! Clearly, nature must be using a different "language" than basic chemistry:



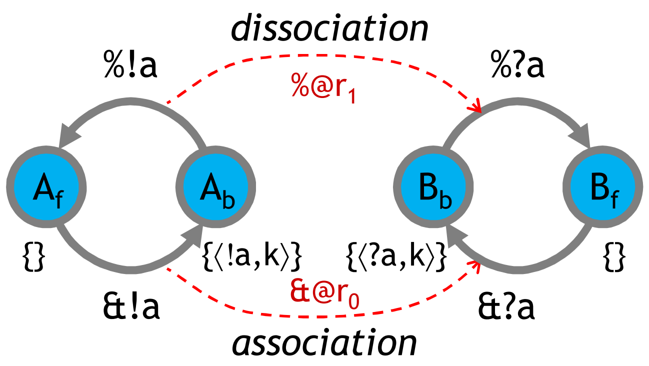
molecule with convex patch +
molecule with concave patch →
molecule with convex patch

- a finite program
- a local rule

Biochemistry = Collision + Complexation



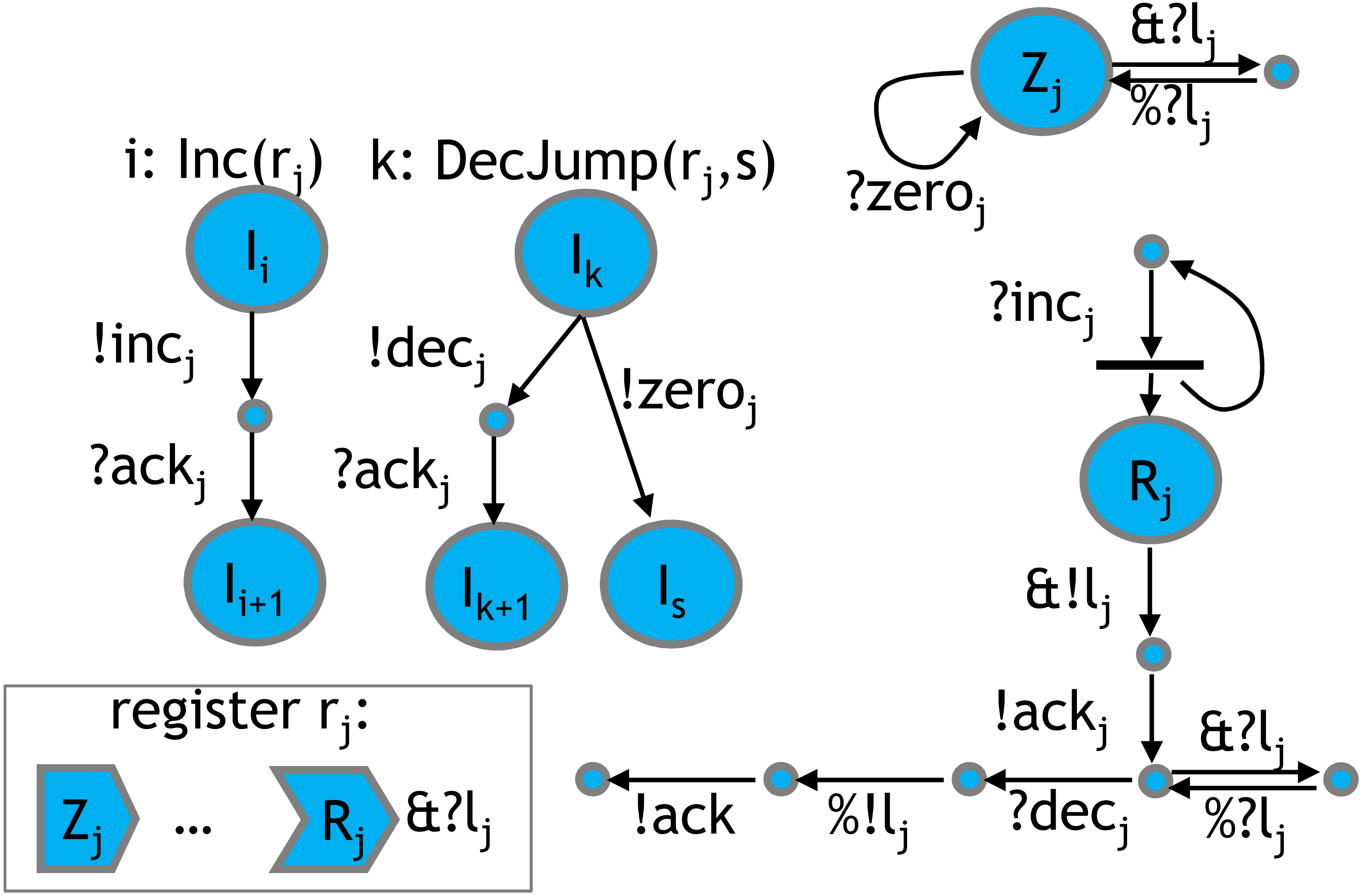
- Complexation is what proteins "do", in contrast to simpler chemicals.



Polyautomata
(polymerizing automata)

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

RAM encoding in BGF



Expressiveness of Biochemistry

- Basic chemistry (FSRN, or CGF) is not Turing-complete
  - By reduction to Petri Net reachability [Soleveichik&al.].
- Biochemistry (FSRN + complexation, or BGF) is Turing-complete.
  - By an encoding of Random Access Machines, using polymers for registers.
- A relatively simple extension of our CGF automata
  - But it is not as easy to find a corresponding extension of chemistry!
- More powerful process algebras of course *are* Turing complete
  - They (e.g. π-calculus) include BGF, but they also have mechanisms that are not directly biologically justifiable.
  - In BGF we have in a sense the minimal biologically-inspired extension of FSRN, and it is already Turing-complete.
- Intrinsic to biochemistry (but not to simple chemistry) is a Turing-complete mechanism.