# On The Computational Power of Biochemistry


**Luca Cardelli**

Microsoft Research

with

**Gianluigi Zavattaro**

University of Bologna



International Summer School on Natural Computation, BNC'08

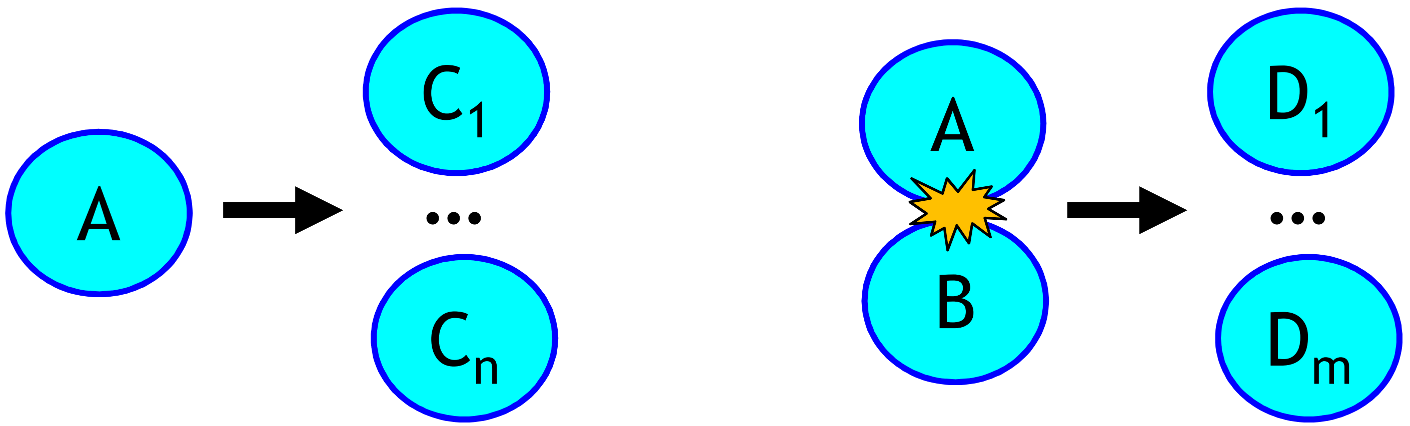
Bertinoro, 2008-09-21


http://LucaCardelli.name

# Biochemistry

# Basic Chemistry

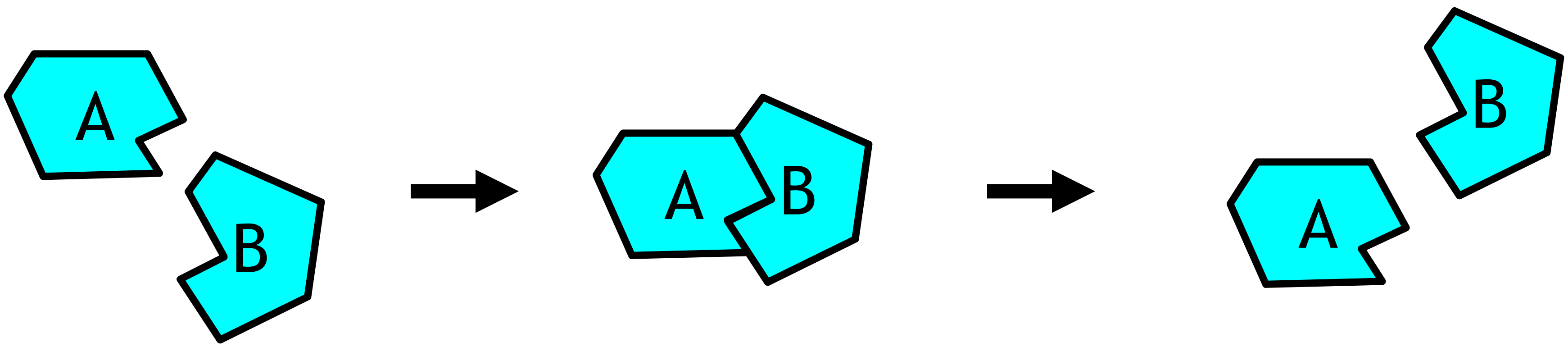
- Molecules belong to Species
- Behavior is described by reactions between species:
  - Monomolecular: $A \to C_1 + \ldots + C_n$
  - Bimolecular: $A + B \to D_1 + \ldots + D_m$




- A.k.a. FSRN (Finite Stochastic Reaction Networks [Sol'08])

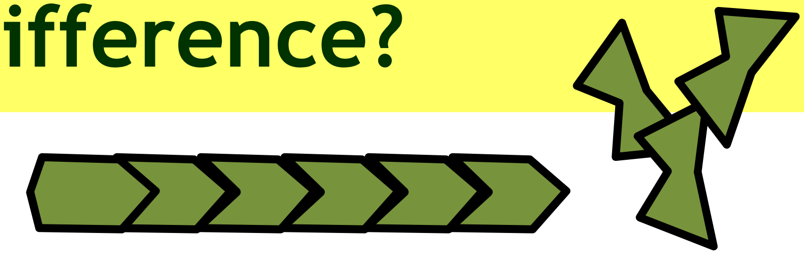
# Basic Biochemistry

- Molecules may also form reversible complexes
  - Association: **A + B → A:B**
  - Dissociation: **A:B → A + B**



# What's the Difference?

Consider linear polymerization:



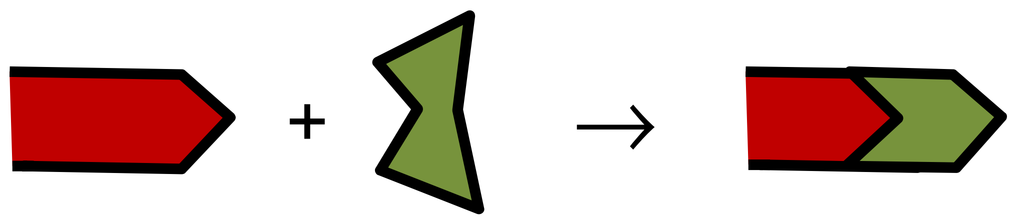
The "chemical program" for polymerization:

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

- 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 e.g. in the genome, so it cannot be infinite! Clearly, nature must be using a different "language" than basic chemistry:

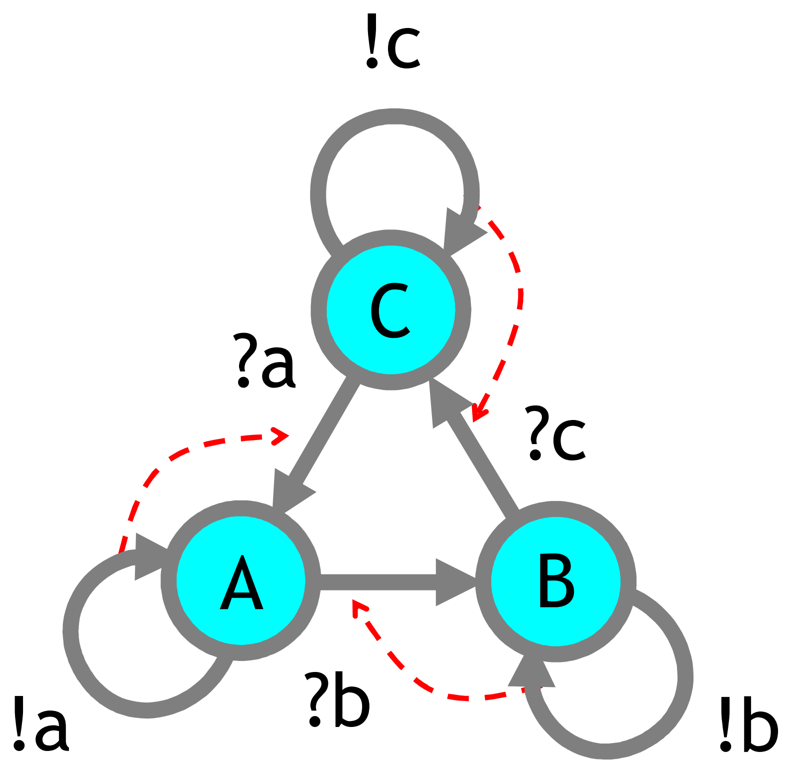



molecule with convex patch +
molecule with concave patch →
molecule with convex patch

- a finite program
- a local rule

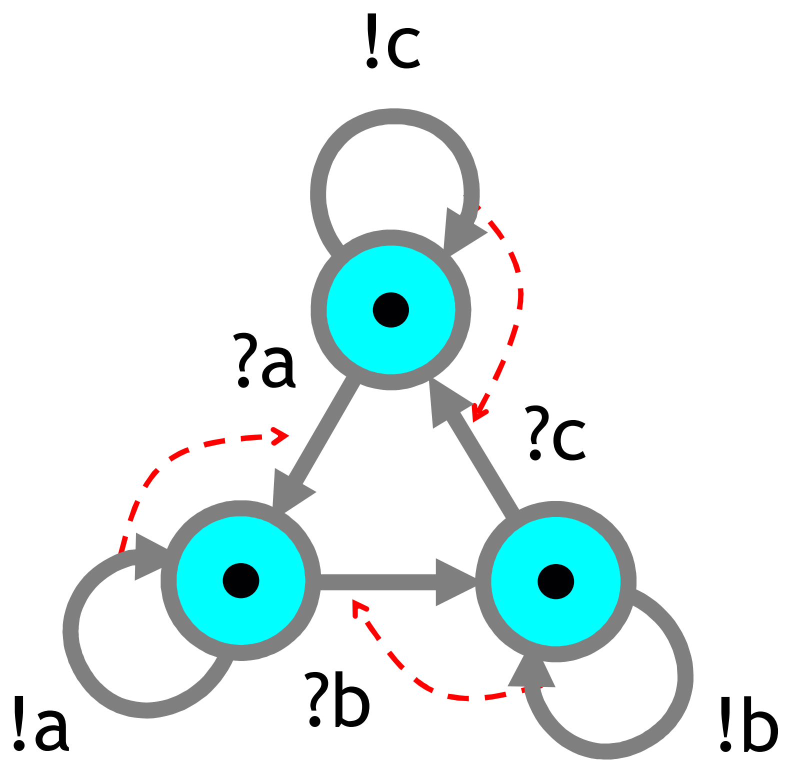
# Termination

# Example: Can it halt?



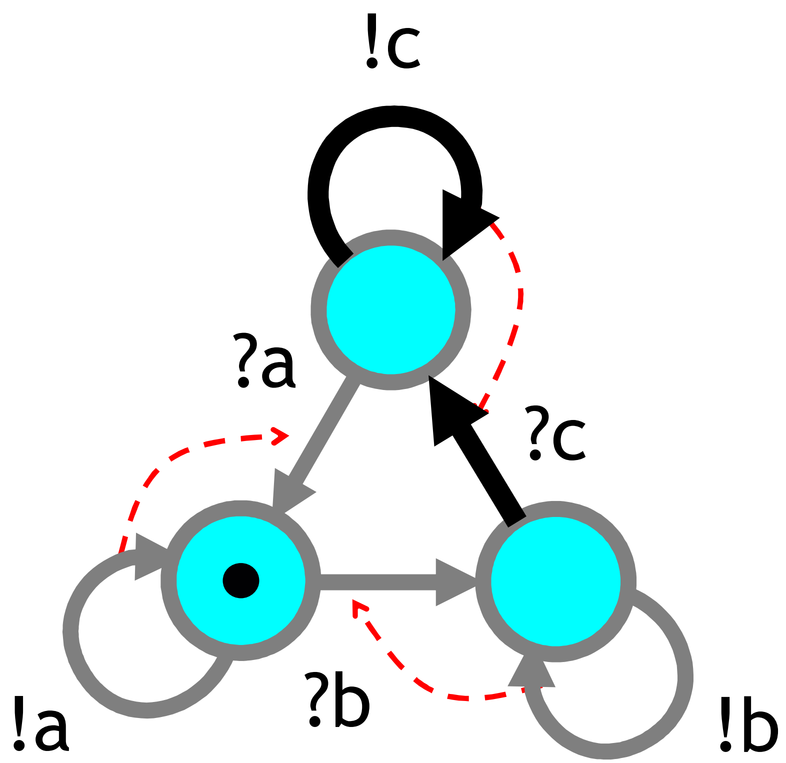
# Example: Can it halt?

3 Automata



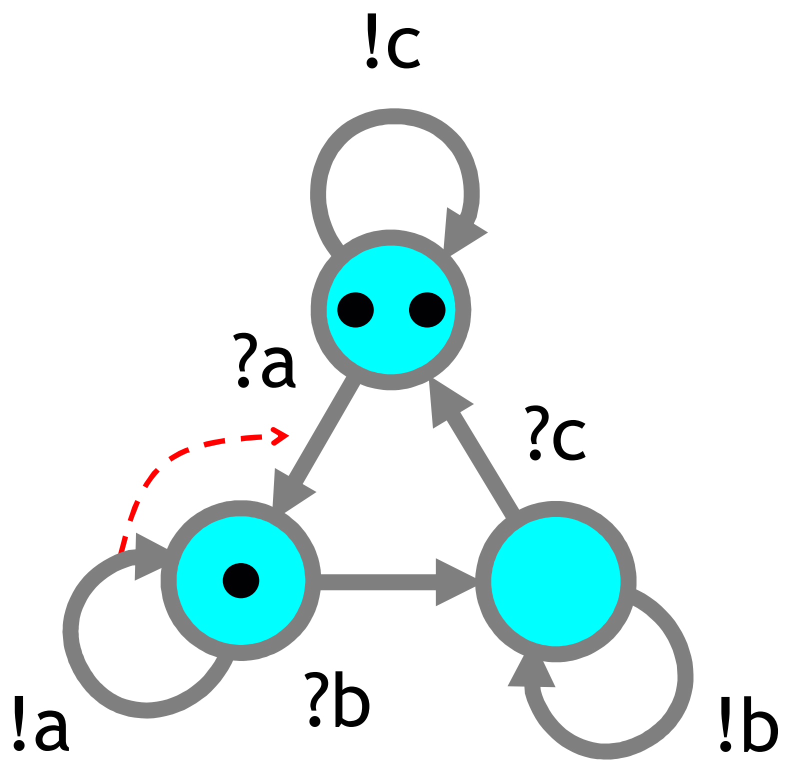
# Example: Can it halt?

3 Automata



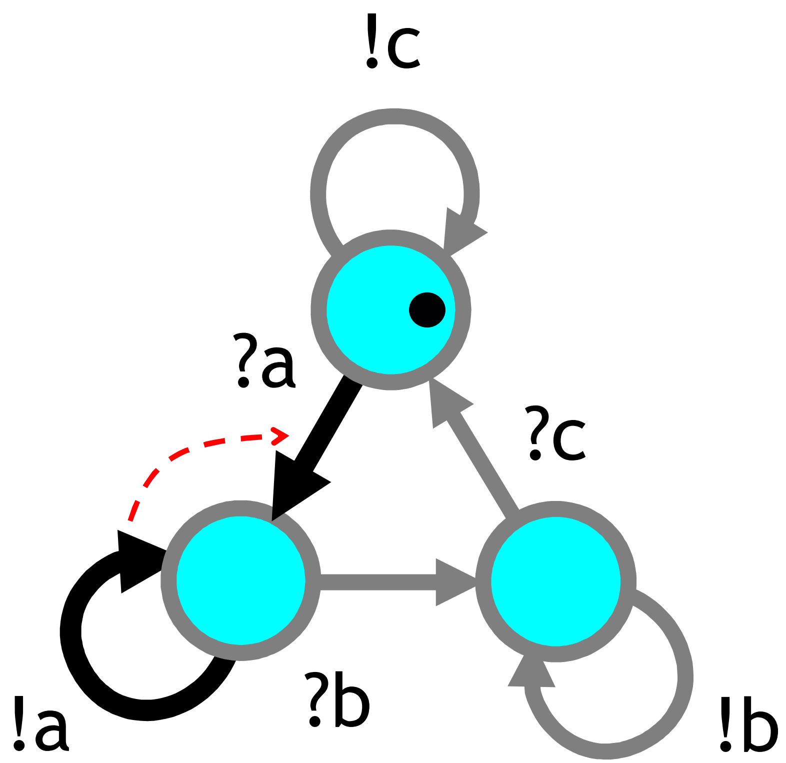
# Example: Can it halt?

3 Automata



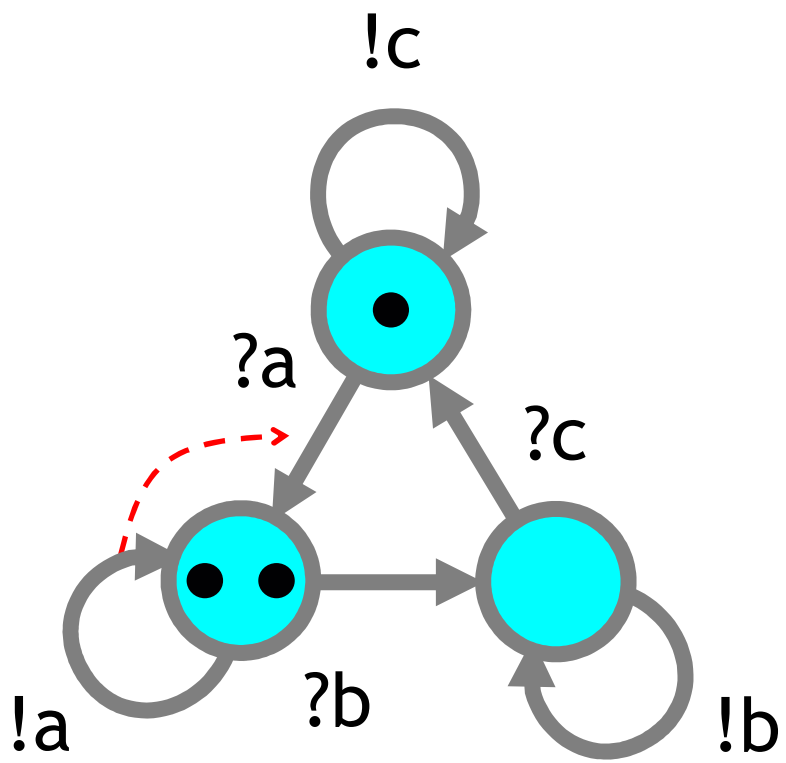
# Example: Can it halt?

3 Automata



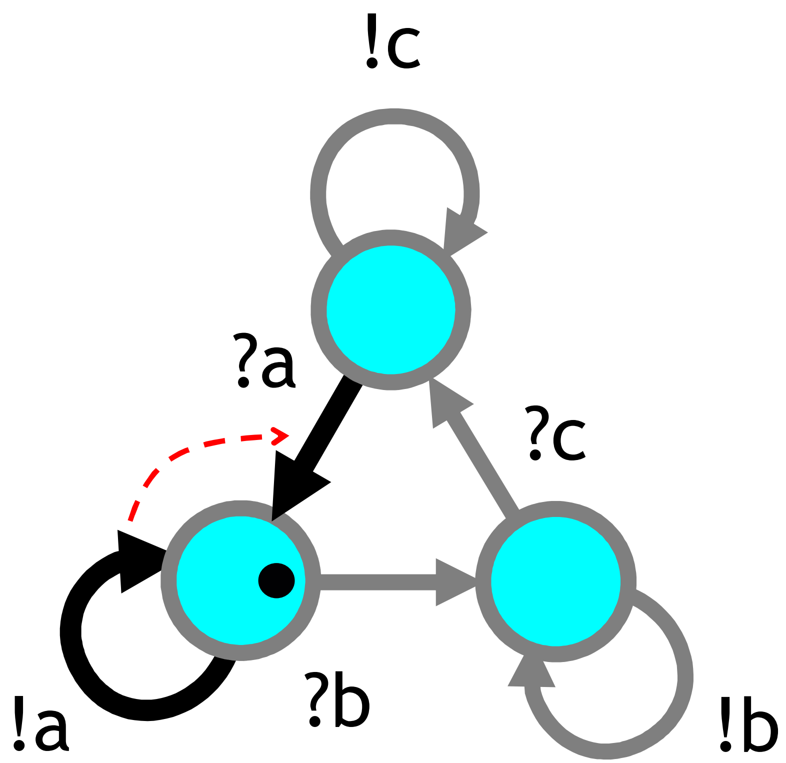
# Example: Can it halt?

3 Automata



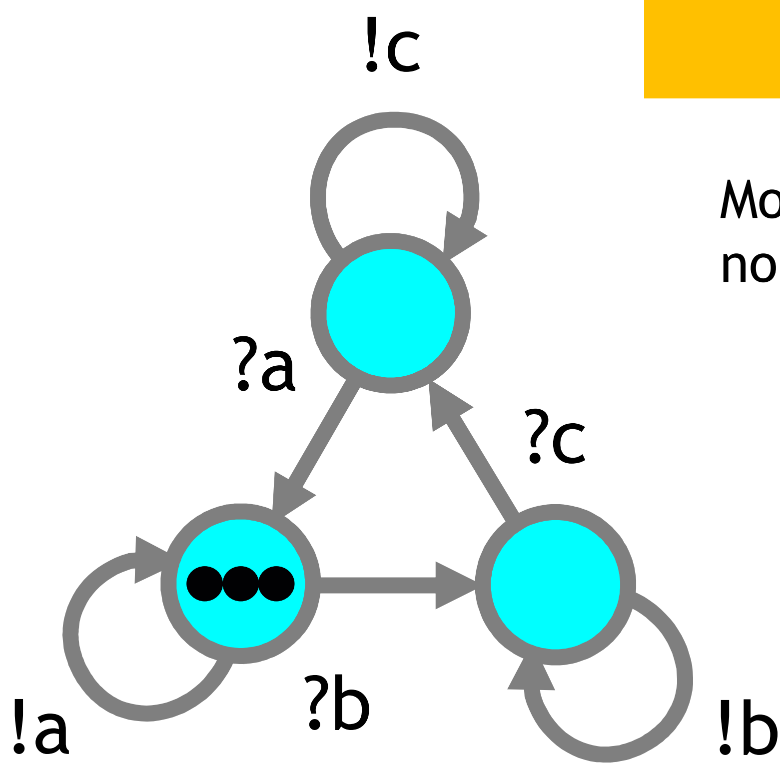
# Example: Can it halt?

3 Automata



# Example: Can it halt?

3 Automata

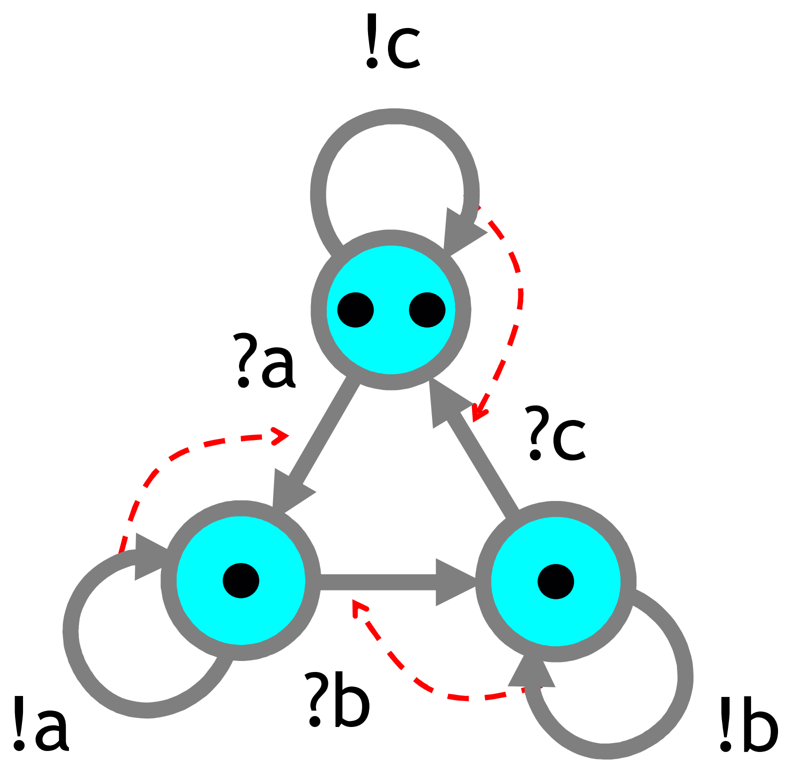



**YES**

Moreover, there is no infinite trace.

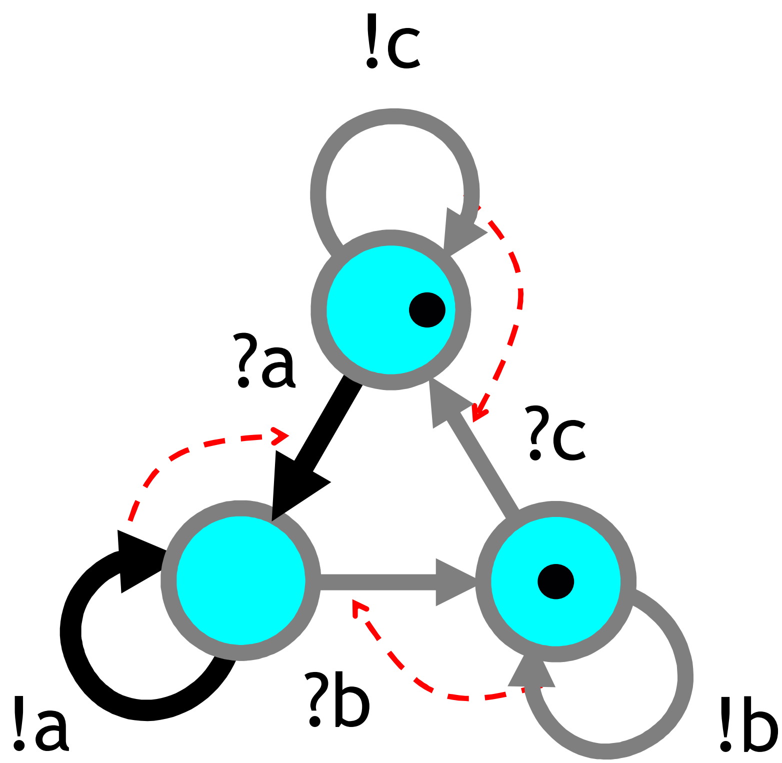
# Example: Can it halt?

4 Automata



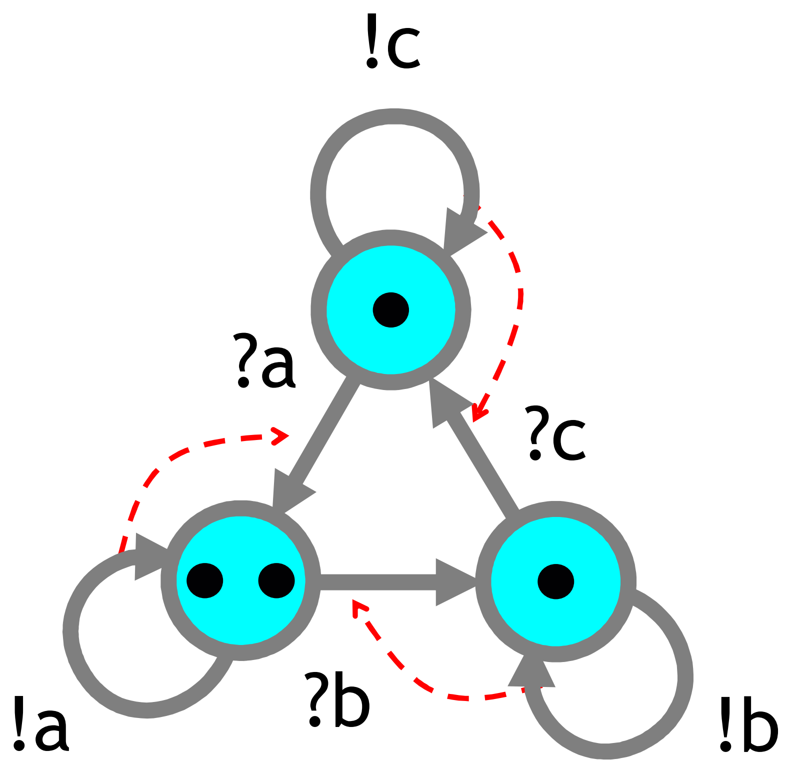
# Example: Can it halt?

4 Automata



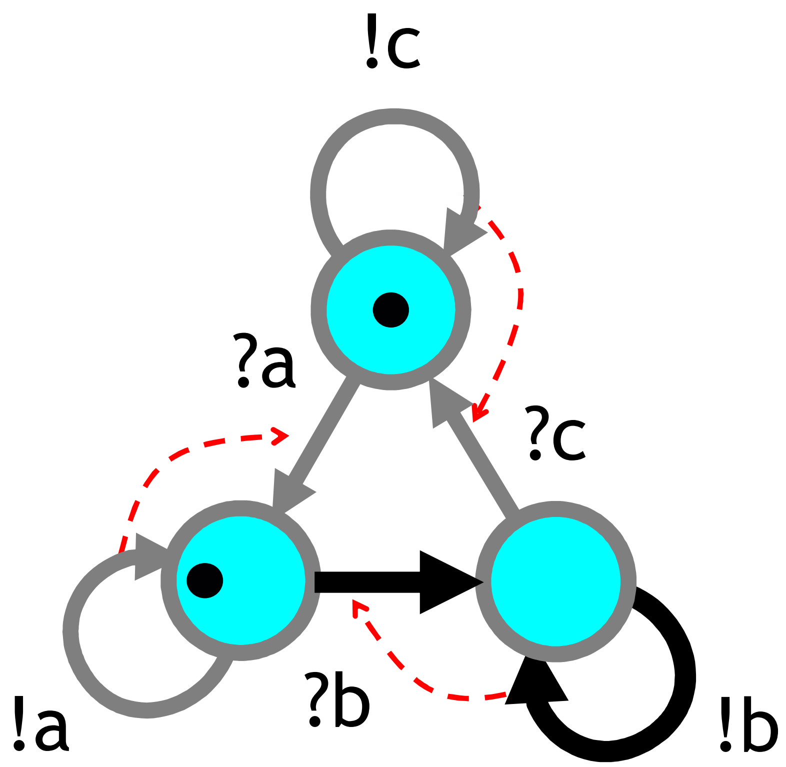
# Example: Can it halt?

4 Automata



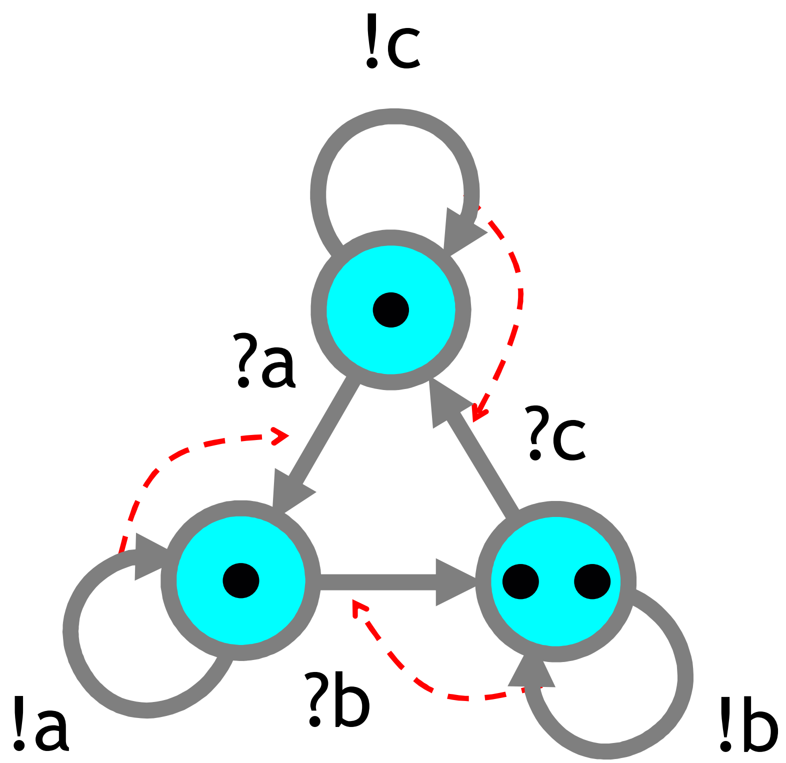
# Example: Can it halt?

4 Automata



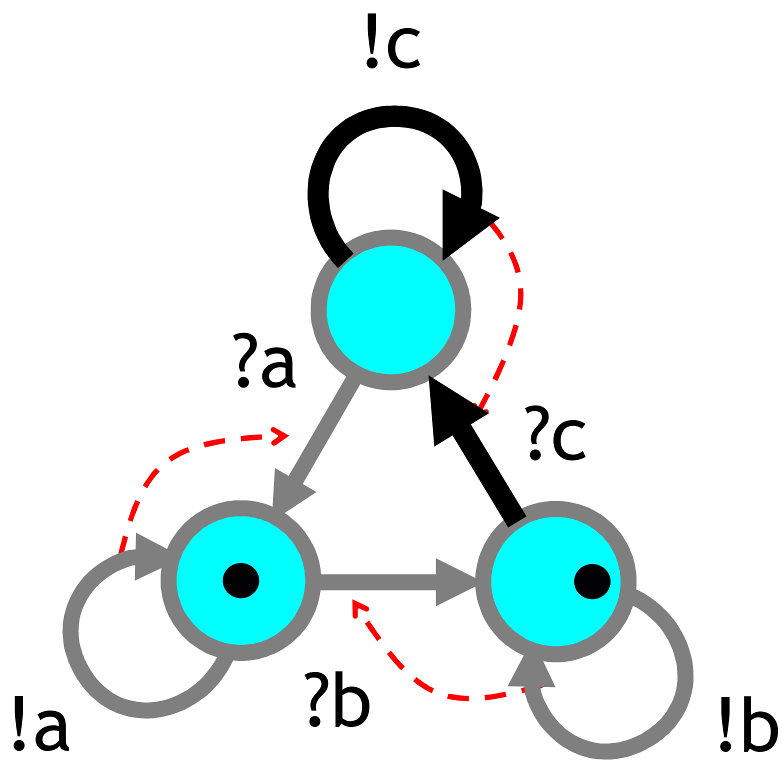
# Example: Can it halt?

4 Automata



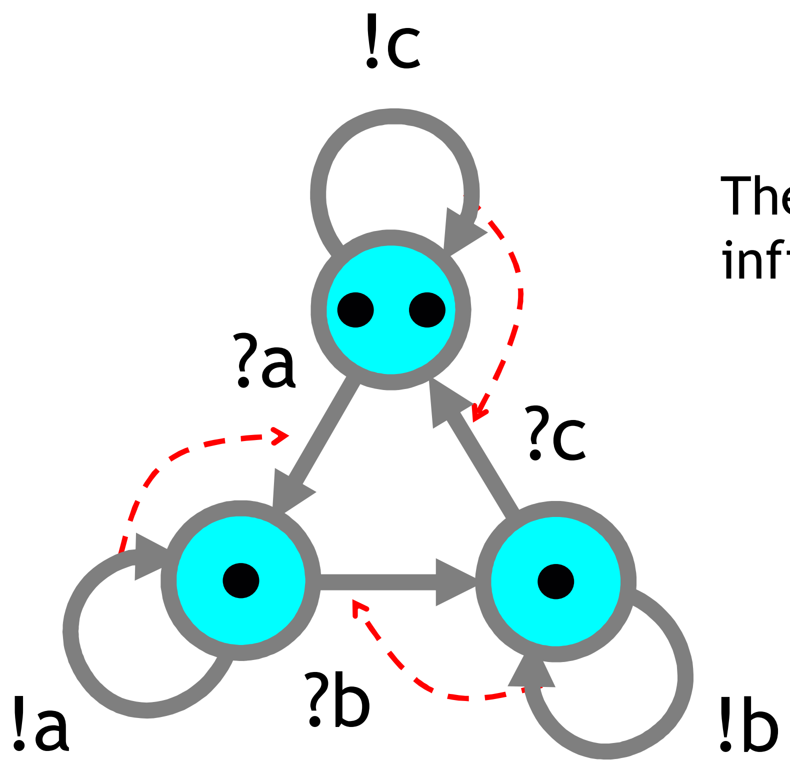
# Example: Can it halt?

4 Automata



# Example: Can it halt?

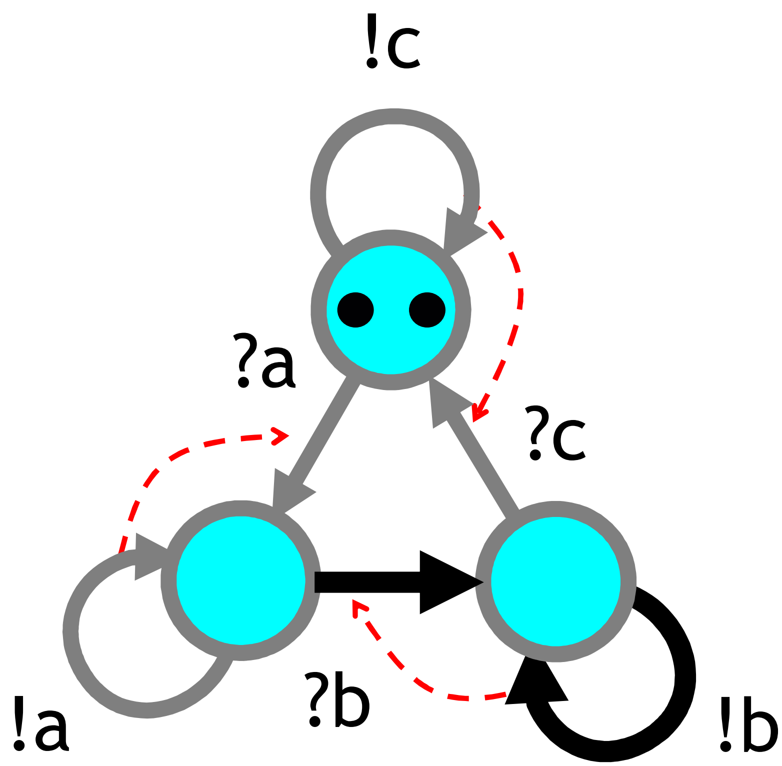
4 Automata




There is an infinite trace.

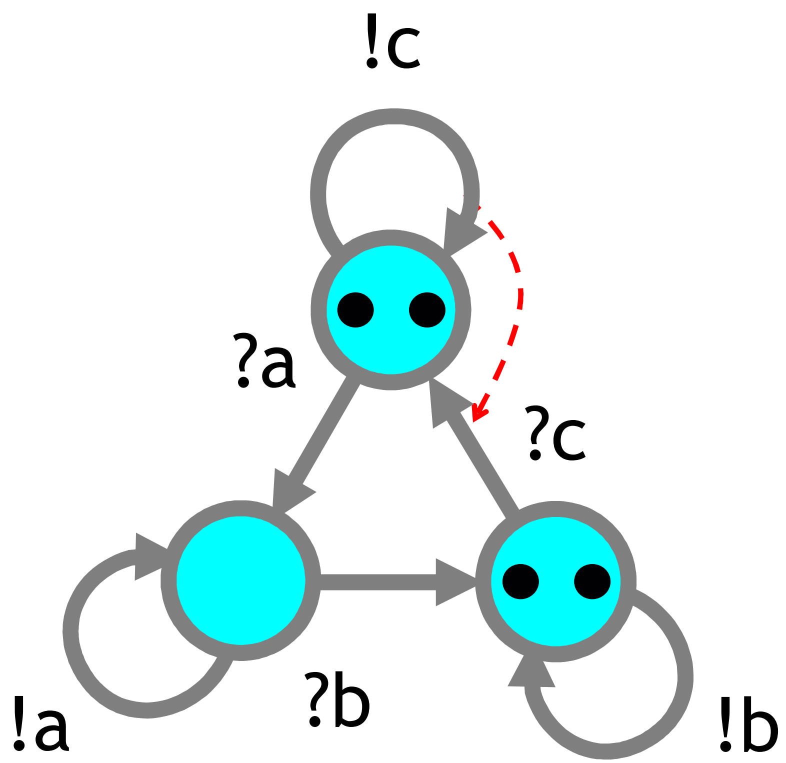
# Example: Can it halt?

4 Automata



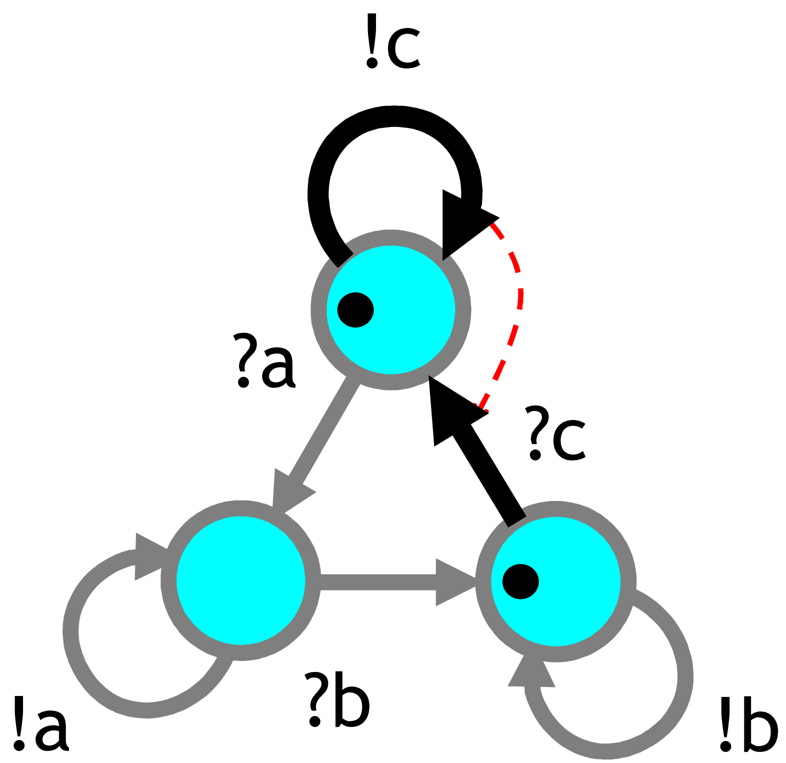
# Example: Can it halt?

4 Automata



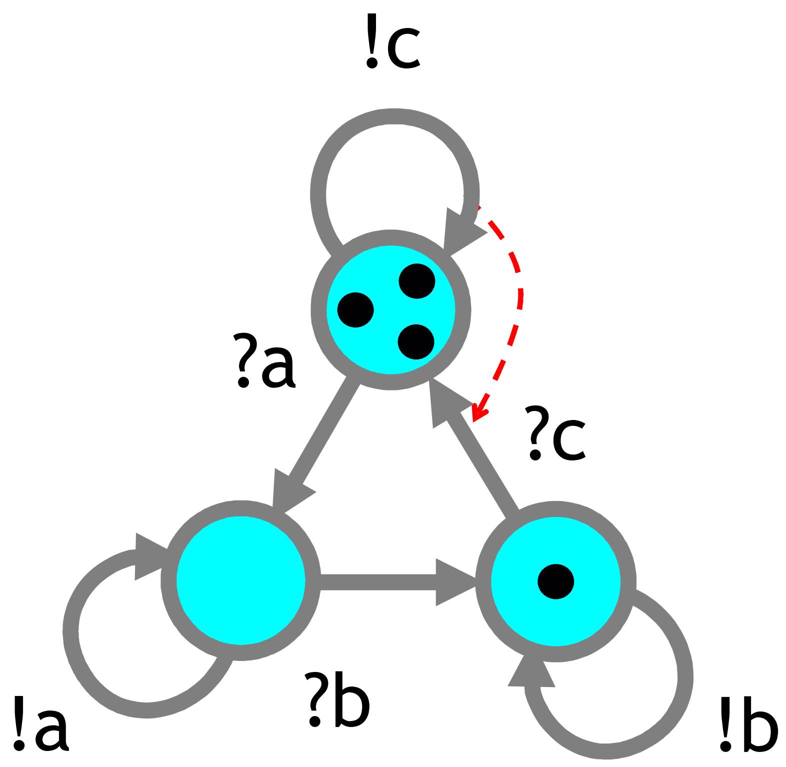
# Example: Can it halt?

4 Automata



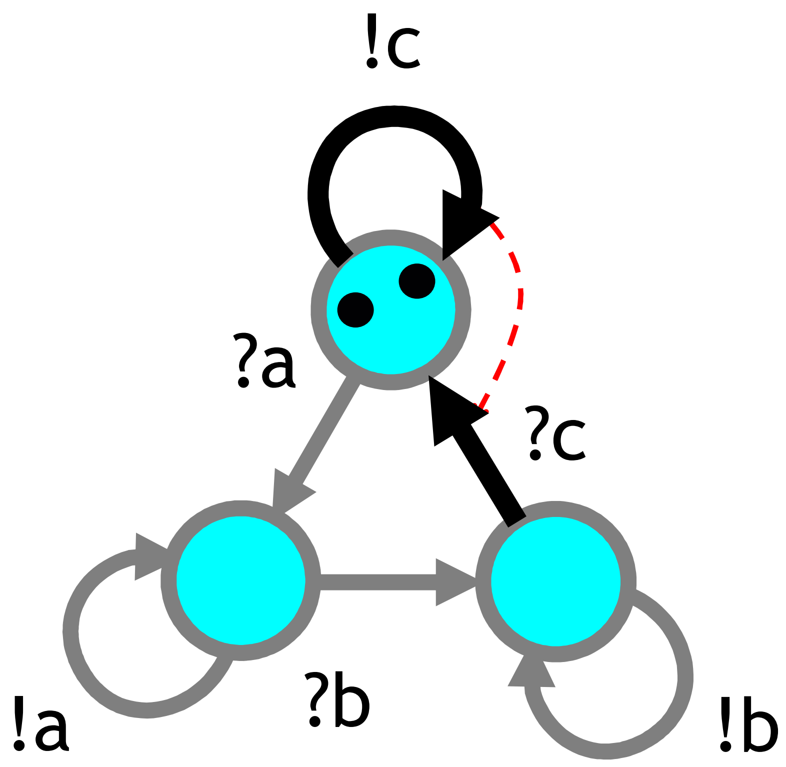
# Example: Can it halt?

4 Automata



# Example: Can it halt?

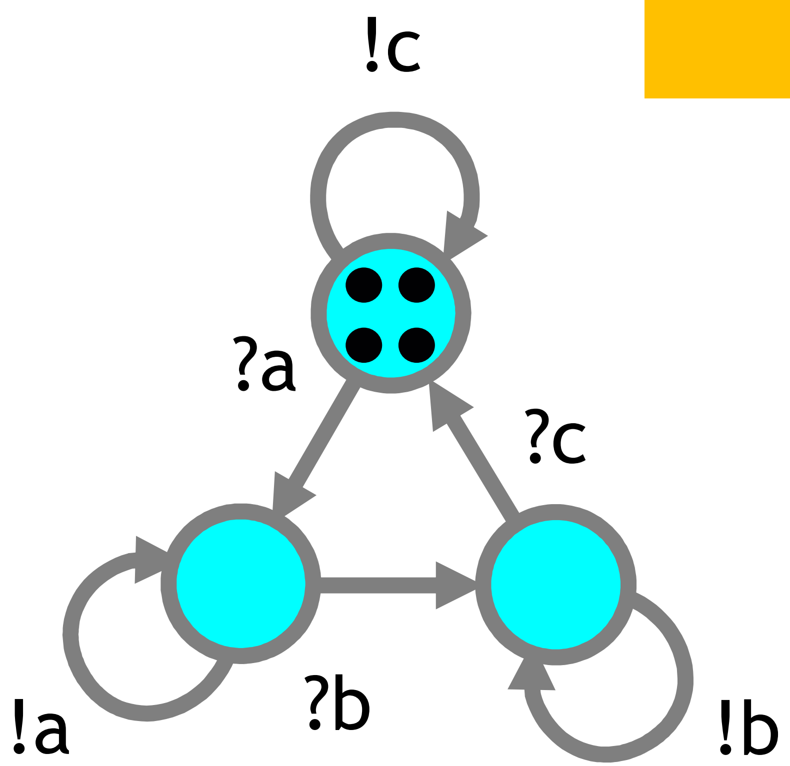
4 Automata



# Example: Can it halt?

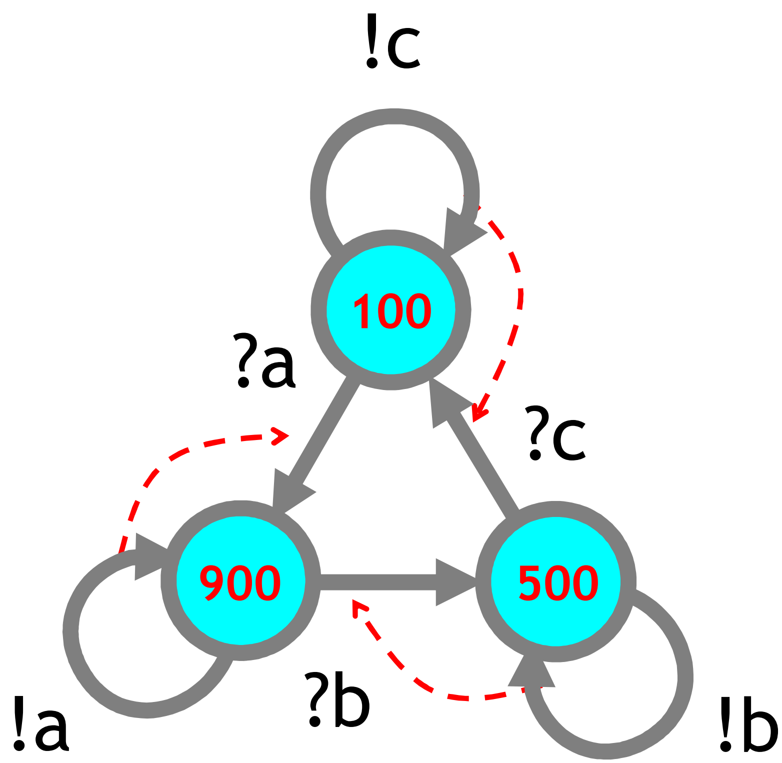
4 Automata

**YES**



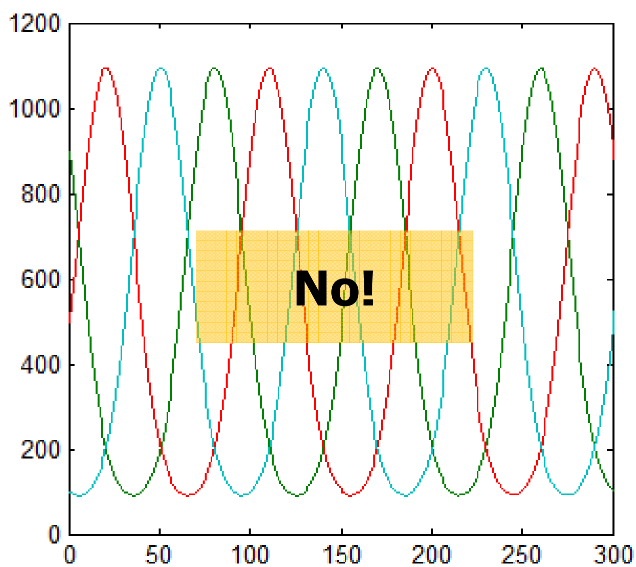
# Example: Can it halt?

1500 Automata



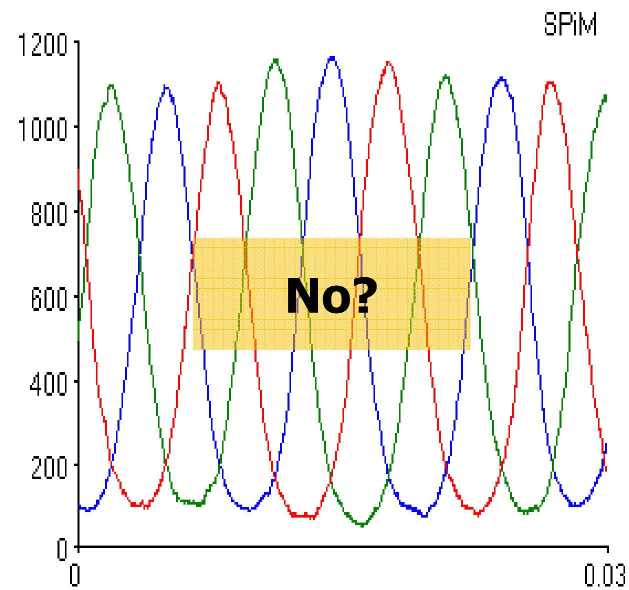
# Example: Can it halt?

## "Experimental Evidence"




Continuous-State Simulation

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




Discrete-State Simulation

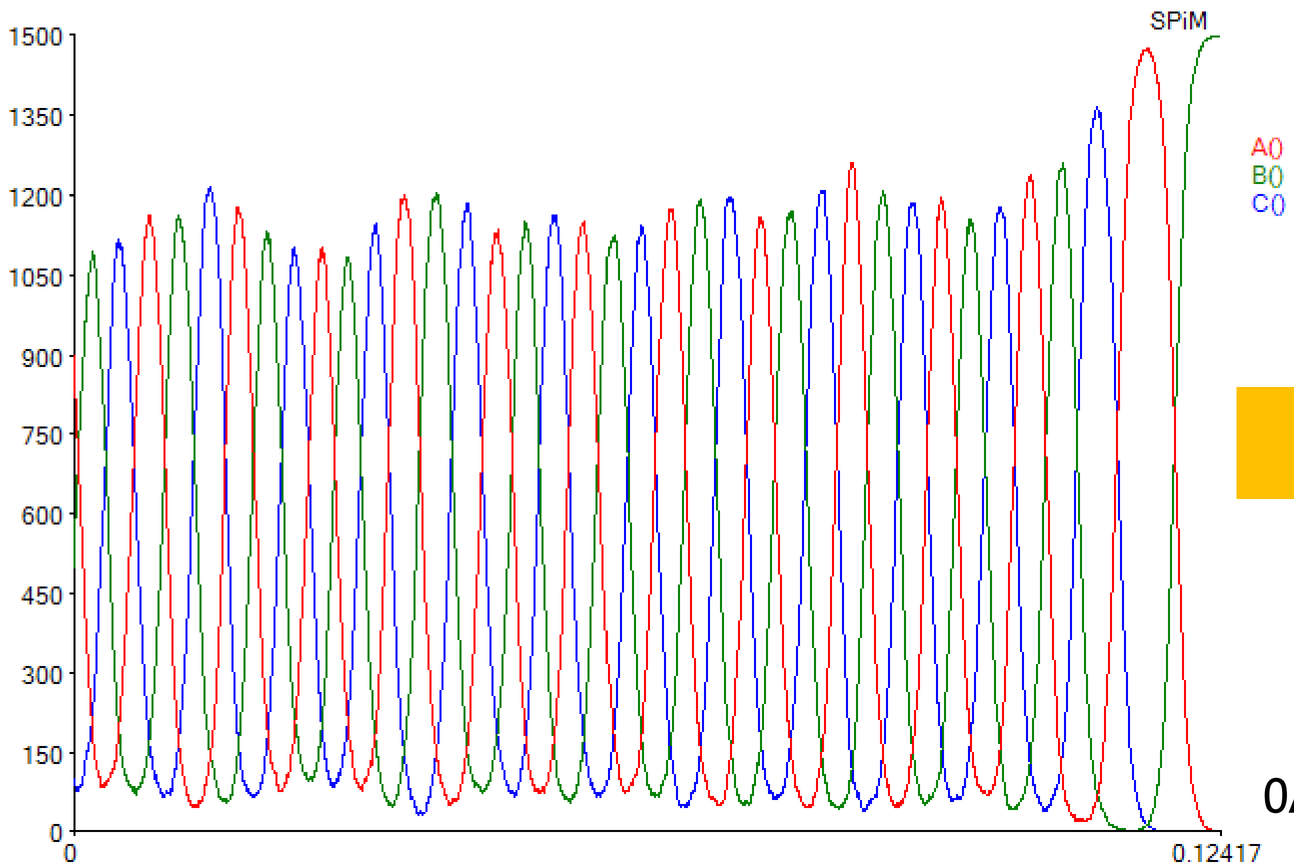
```
directive sample 0.03 1000
directive plot A(); B(); C()

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

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

# Example: Can it halt?

But in a longer experiment...

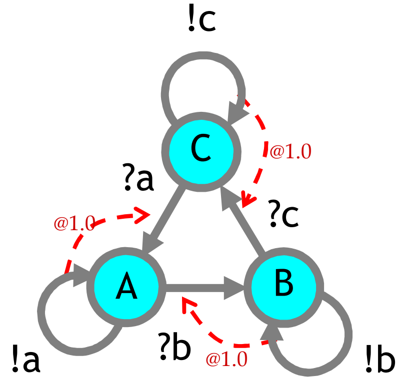



Discrete-State Simulation

**YES**

0A + 1500B + 0C

# Example: Can it halt?




Termination strategy

It *can* terminate. (Apply reaction b until no more A's, then apply reaction c until no more B's. Then all are C.)

Nondeterministic termination

It *may* diverge (with 4+ molecules).

Stochastic termination

The probability measure of the terminated states of the oscillator's CMTC is 1.

=> Stochastic fairness

It *cannot* diverge!

# Basic Chemistry Can't Compute!

# But it's all just Petri Nets!

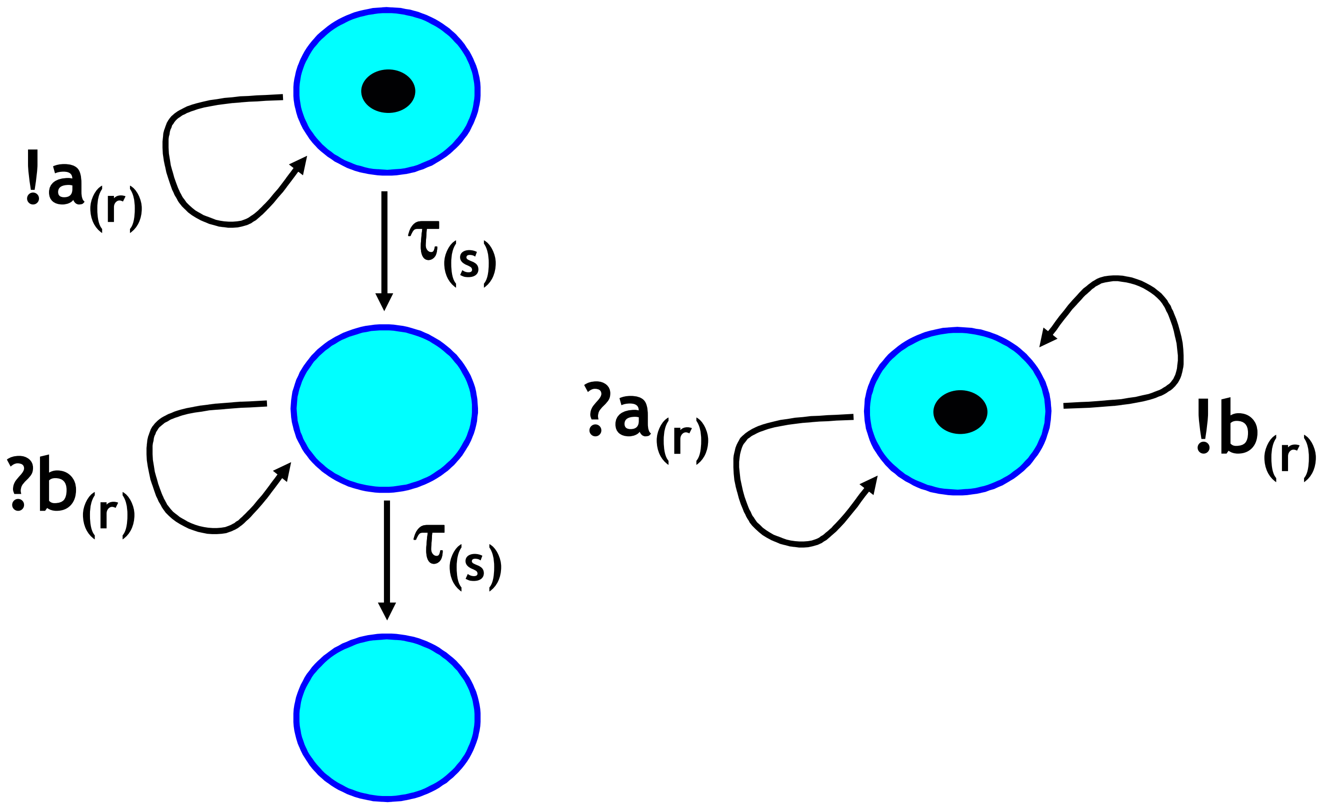
- It is possible to translate an arbitrary CGF (or FSRN) into a Place/Transition Petri Net.
  - Ignoring rates, and of course losing compositionality.

- Pretty much everything is decidable in P/T Nets.
  - In particular, reachability of a dead ("halting") state.

- Hence both CGF and FSRN are not Turing-complete!
  - Basic chemistry can't compute!
    (Soloveichik et. al., Natural Computing 2008)
  - Even though stochastic chemistry is extremely rich, e.g. it includes chaotic systems.

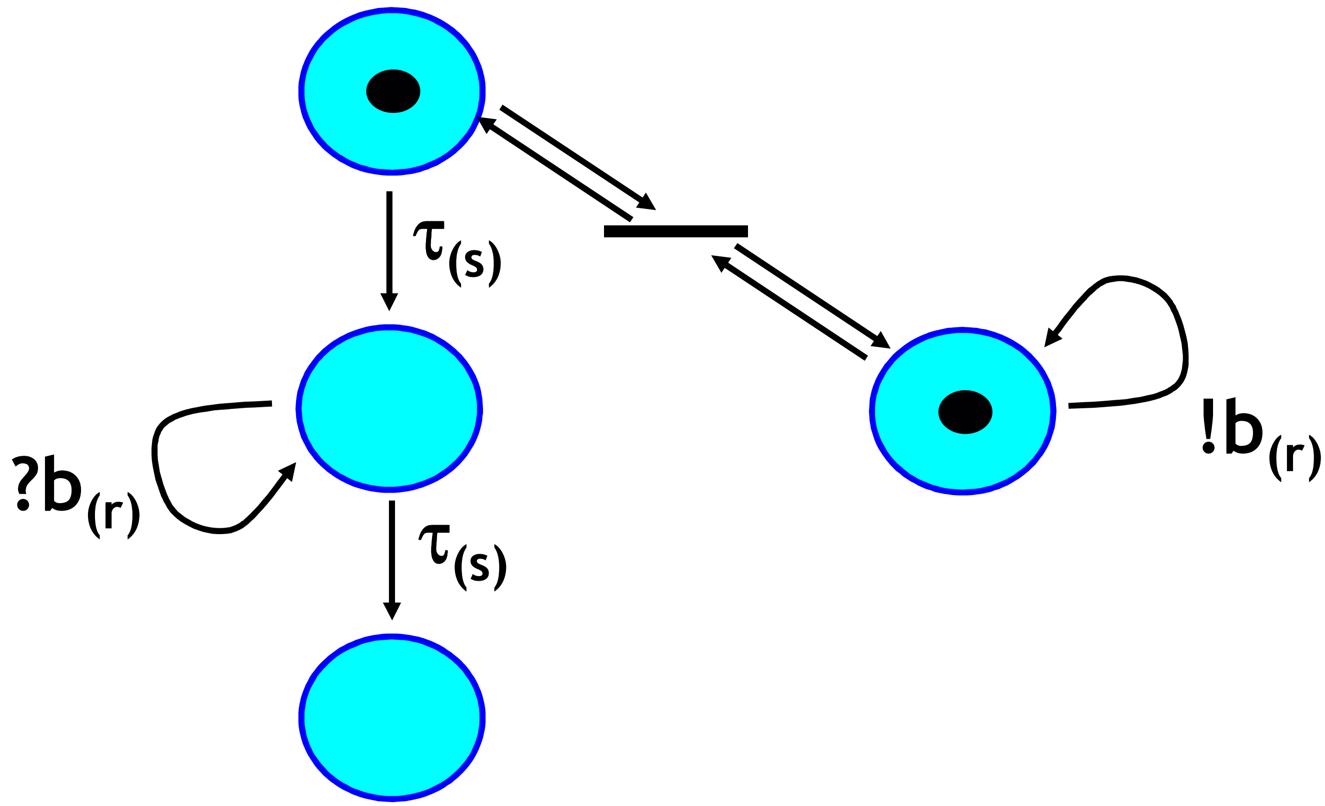
# A Petri net semantics for CGF

- One place for each Species
- One transition for each reaction



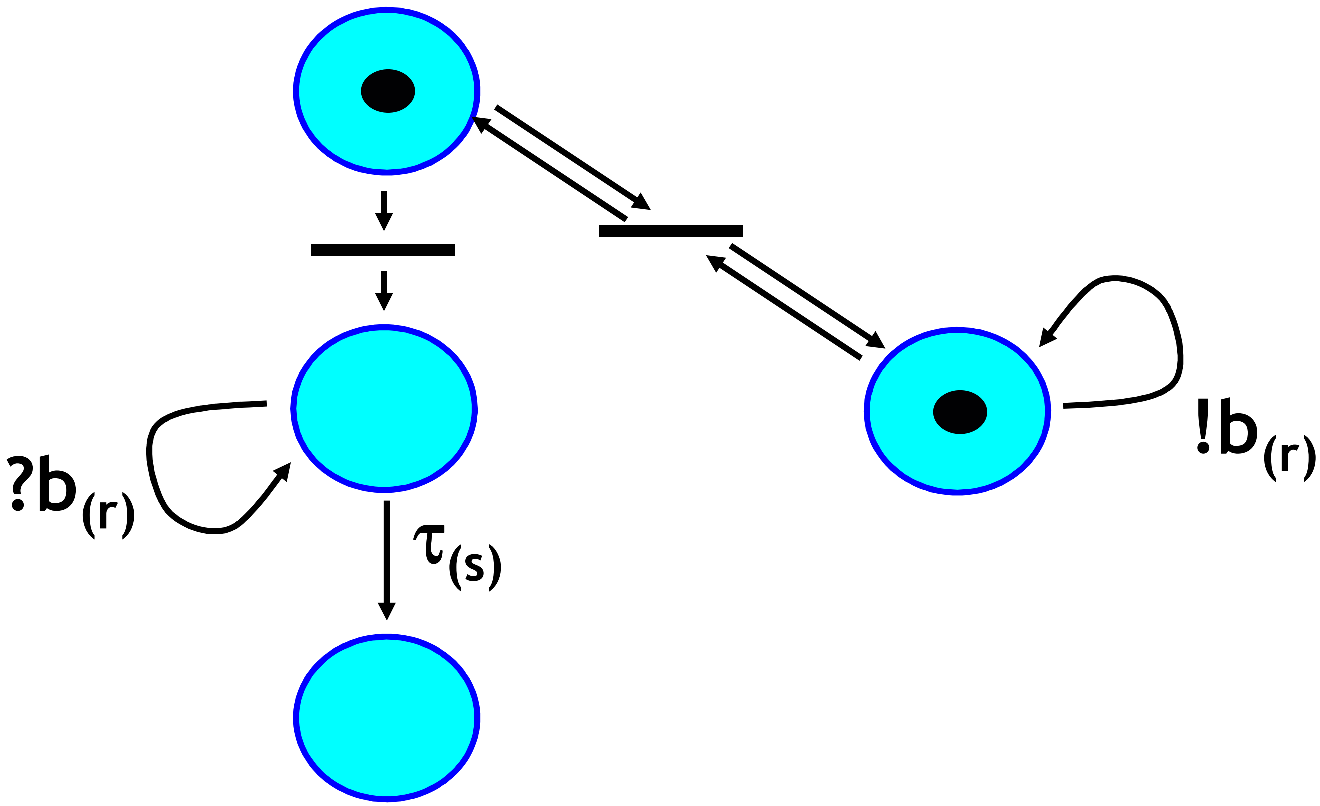
# A Petri net semantics for CGF

- One place for each Species
- One transition for each reaction



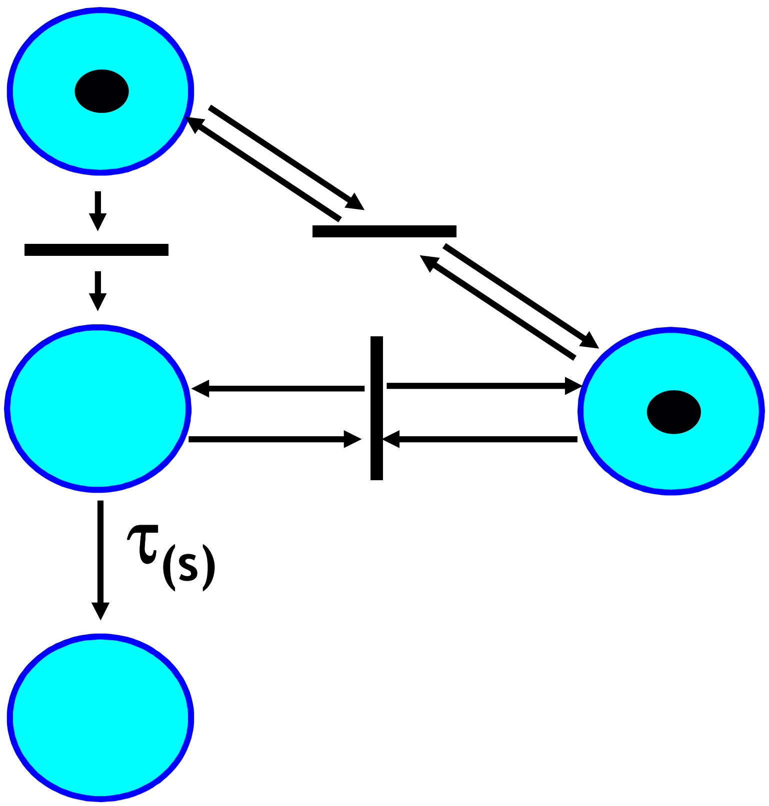
# A Petri net semantics for CGF

- One place for each Species
- One transition for each reaction



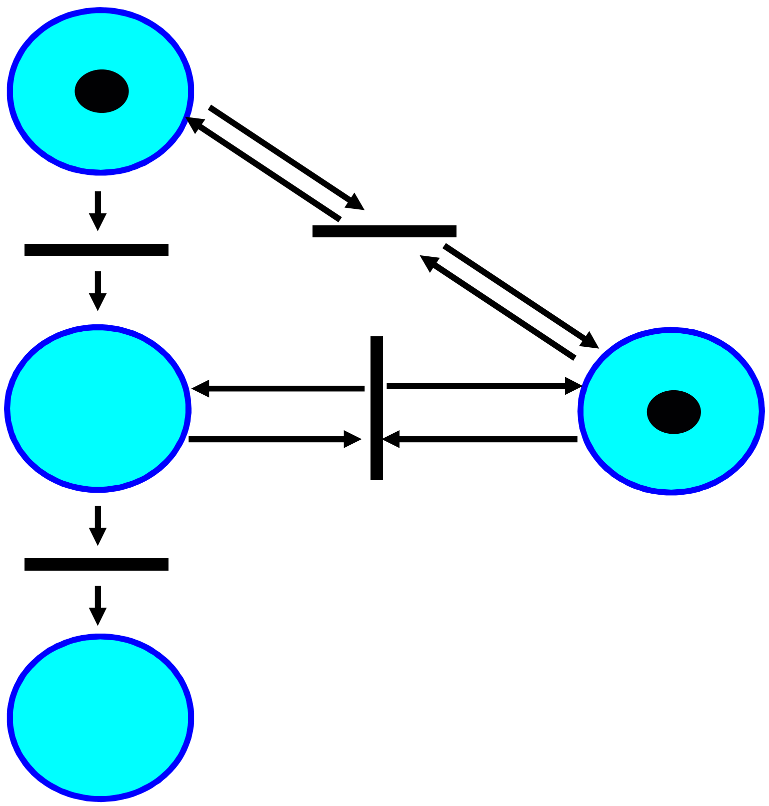
# A Petri net semantics for CGF

- One place for each Species
- One transition for each reaction



# A Petri net semantics for CGF

- One place for each Species
- One transition for each reaction



# Termination Problems in Chemical Kinetics

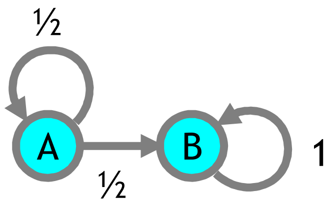
# Probability Measure for a Markov Chain

- 1-step probability
  - If a state A has n outgoing transitions to states $B_1, \ldots, B_n$, labeled with rates $r_1, \ldots, r_n$, the probability of going from A to $B_k$ in one step is:
  - $p^{(1)}(A,B_k) = r_k / \sum_i r_i$

- Many-step probability (Chapman-Kolmogorov equation)
  - The probability of going from A to B in n+m steps is the sum of all ways of going in n steps form A to any X and then in m steps from X to B.
  - $p^{(n+m)}(A,B) = \sum_X p^{(n)}(A,X)\, p^{(m)}(X,B)$

- Termination probability (reaching an absorbing state)
  - The probability of going from state A to an absorbing state B is the limit of going from A to B in n steps:
  - $p(A,B) = \lim_{n\to\infty} p^{(n)}(A,B)$

# Ex.:




$p^{(1)}(A,B) = 1/2$ $\qquad$ $p^{(1)}(A,A) = 1/2$ $\qquad$ $p^{(n)}(B,B) = 1$

$p^{(2)}(A,B) = p^{(1)}(A,A)\ p^{(1)}(A,B) + p^{(1)}(A,B)\ p^{(1)}(B,B) = 1/4 + 1/2 = 3/4$

$p^{(3)}(A,B) = p^{(1)}(A,A)\ p^{(2)}(A,B) + p^{(1)}(A,B)\ p^{(2)}(B,B) = 3/8 + 1/2 = 7/8$

$p^{(4)}(A,B) = p^{(1)}(A,A)\ p^{(3)}(A,B) + p^{(1)}(A,B)\ p^{(3)}(B,B) = 7/16 + 1/2 = 15/16$

…

$p(A,B) = \lim_{n\to\infty} p^{(n)}(A,B) = \lim_{n\to\infty} (n-1)/n = 1$

# Termination Problems

- Probability Measure
  - Let p be the probability measure associated to the computations in a CGF (E,P) that lead to a terminated solution.

- Existential Termination
  - (E,P) existentially terminates if $p > 0$.

- Universal Termination
  - (E,P) universally terminates if $p = 1$.

- Probabilistic Termination
  - (E,P) terminates with probability higher than $0 < \varepsilon < 1$, if $p > \varepsilon$.

# Termination Results

| | Stochastic | Nondeterministic |
|---|---|---|
| Existential Termination | Decidable[1] | Decidable[4] |
| Universal Termination | Undecidable[2] | Decidable[5] |
| Probabilistic Termination | Undecidable[3] | N.A. |

- Chemical kinetics is not Turing-complete[1]
- Chemical kinetics is Turing-complete up to an arbitrary error[3]
- Existential Termination is equally hard in stochastic and nondeterministic[1,4]
- Universal termination is harder in stochastic than in nondeterministicc[2,5]
- The fairness implicit in stochastic computation makes checking universal termination undecidable[2]

([1,3] due to Soloveichik et. al., Natural Computing 2008)

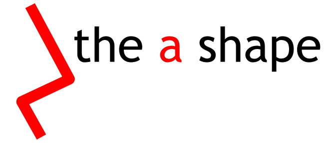
# Biochemical Ground Form

# “Turifying” Chemistry

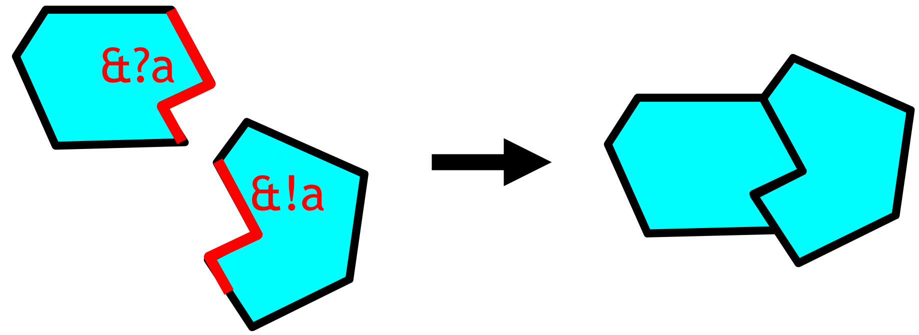
- What can we add to basic chemistry to make it Turing-complete?
- Lots of stuff
  - E.g. we can go from CGF to full $\pi$-calculus
- But is there…
  - A *basic* mechanism
  - which is also biologically *realistic*?

# Association and Dissociation in BGF

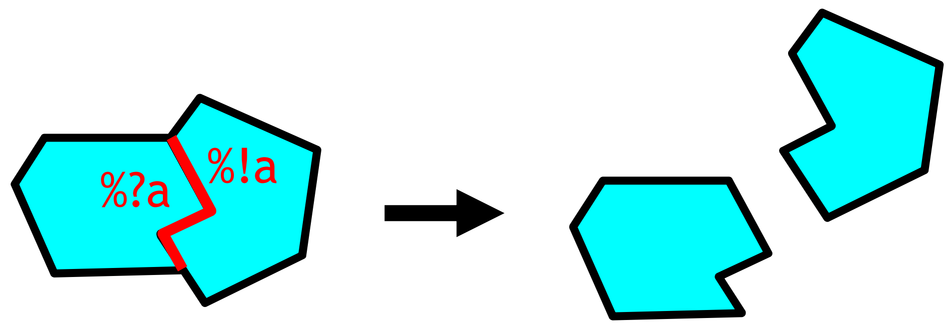
- Association patches are named




- & – association
  - &?a associate
  - &!a co-associate




- % – dissociation
  - %?a dissociate
  - %!a co-dissociate




- A given patch can *hold* only one association at a time
- Two molecules can dissociate only if *they* are associated

# Example: Linear Polymerization

$SF = \tau_{(rs)};\ S|SF$
$MF = \tau_{(rm)};\ M|MF$

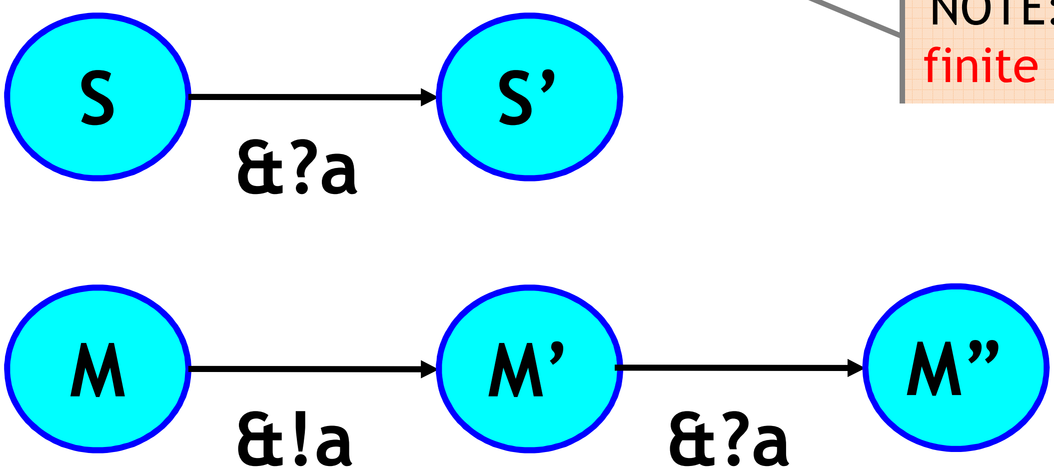
Seed factory and Monomer factory

S = &?a; S'
M = &!a; M'
M' = &?a; M"

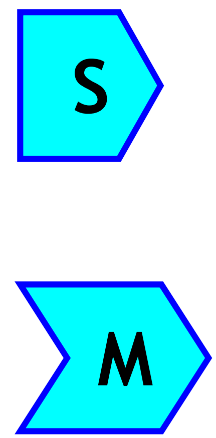
S' = ...
M" = ...

Any further behavior

NOTE: this is a finite program!

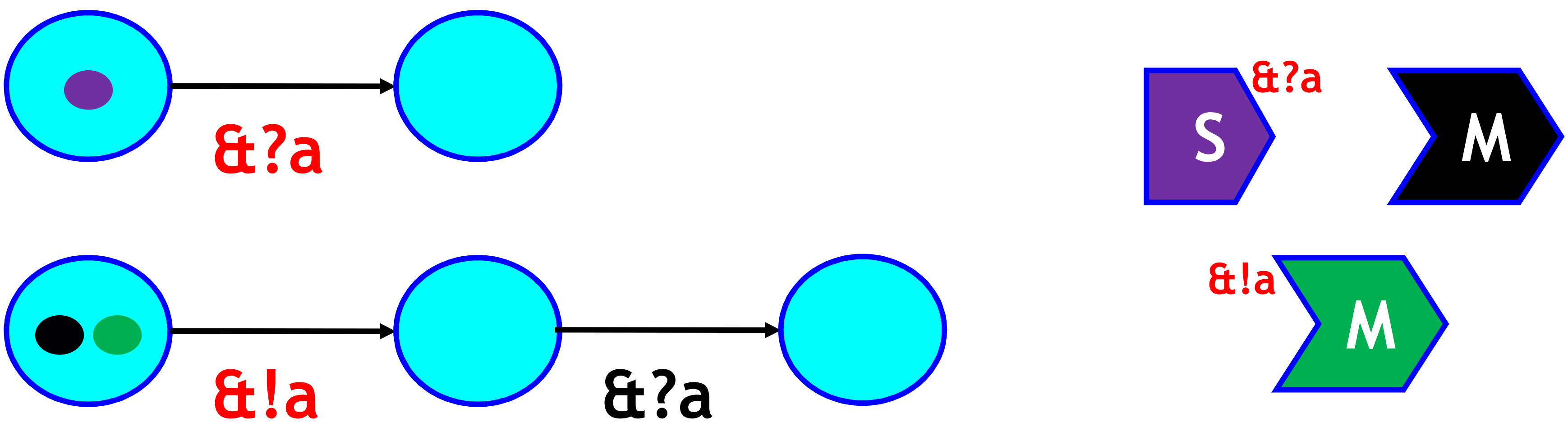



Seed

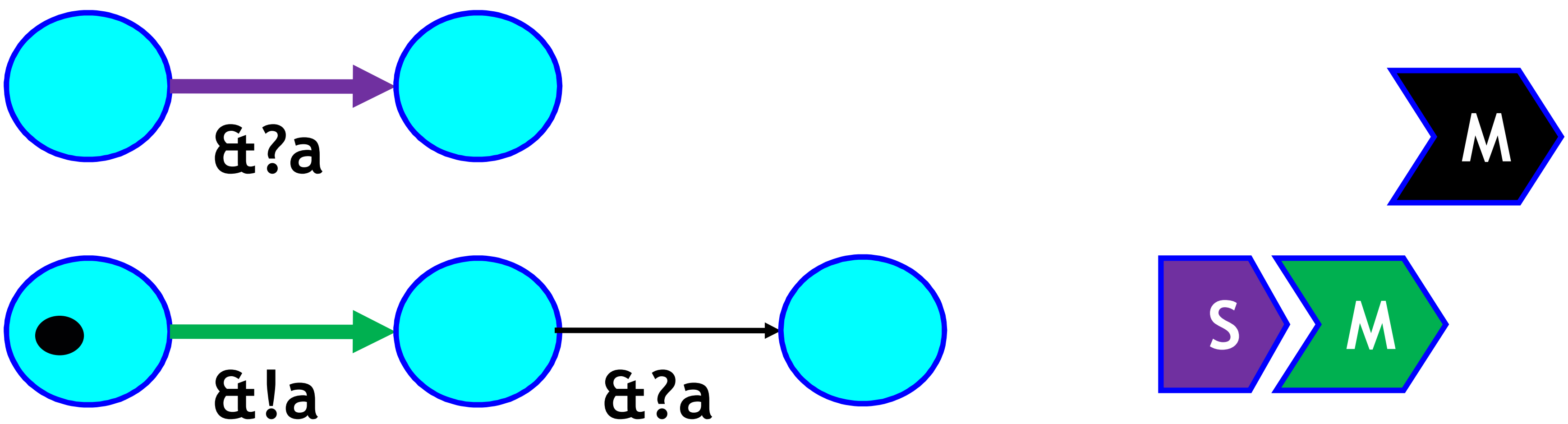



Monomer

# Example: Linear Polymerization

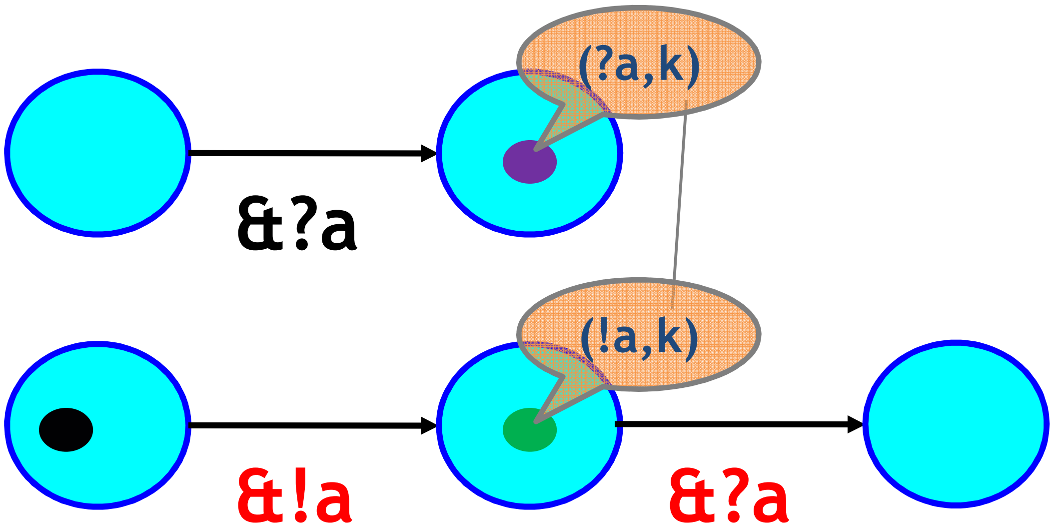


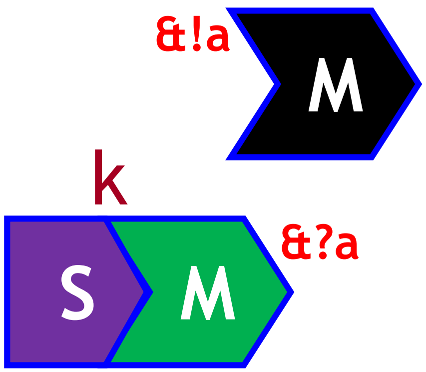
# Example: Linear Polymerization



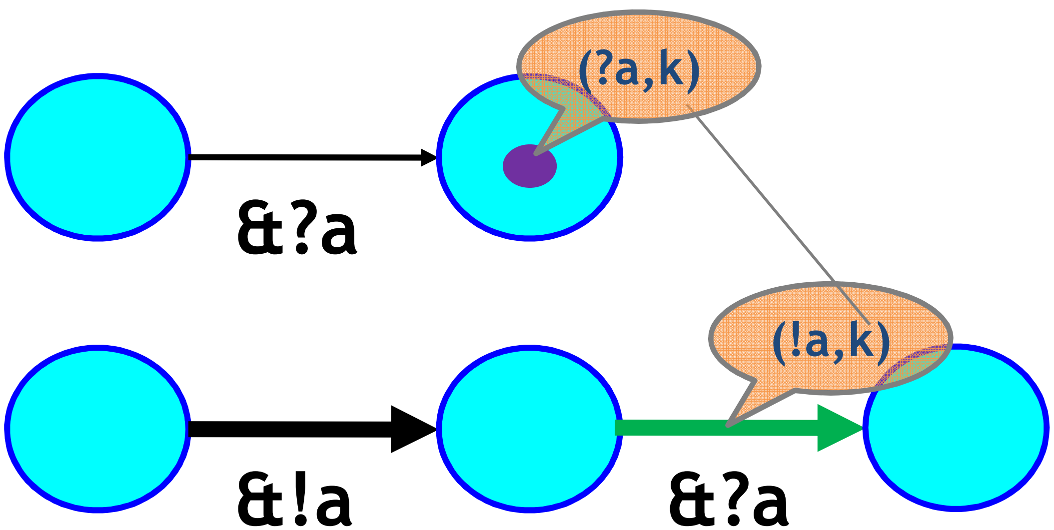
# Example: Linear Polymerization

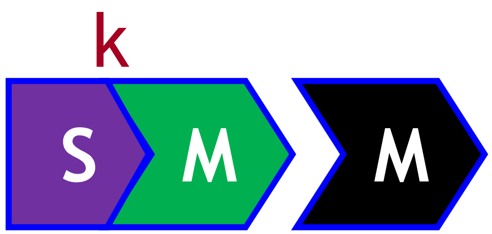
- Each association has a unique key
- Keys are stored in the molecule's association history



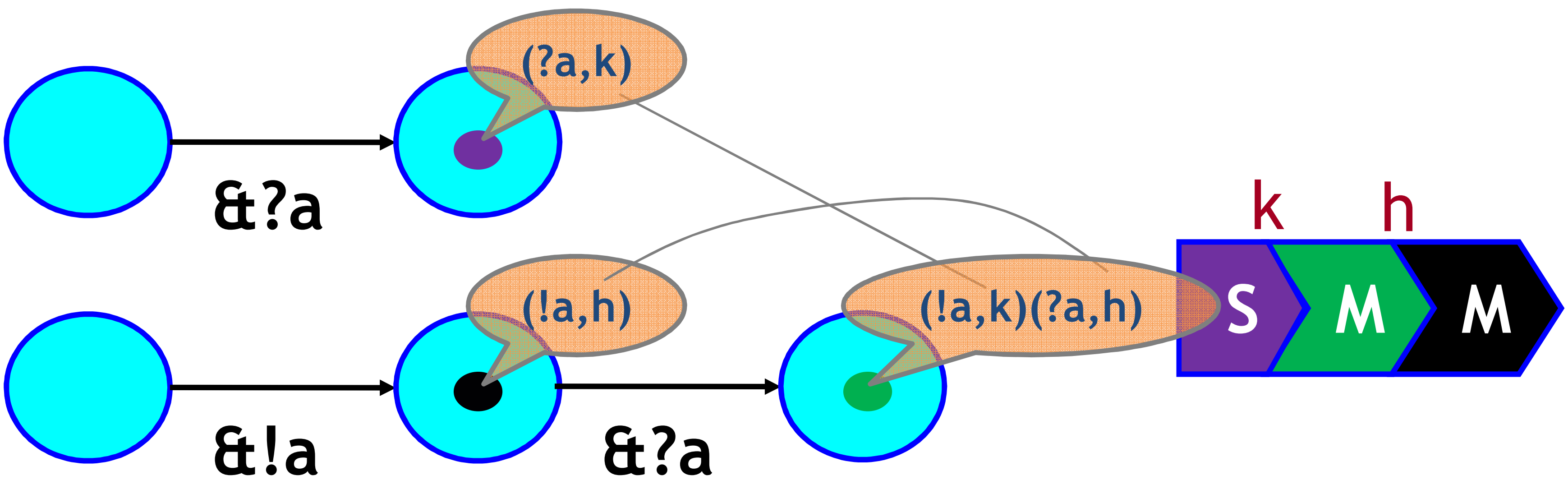



# Example: Linear Polymerization



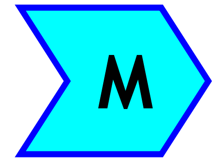



# Example: Linear Polymerization

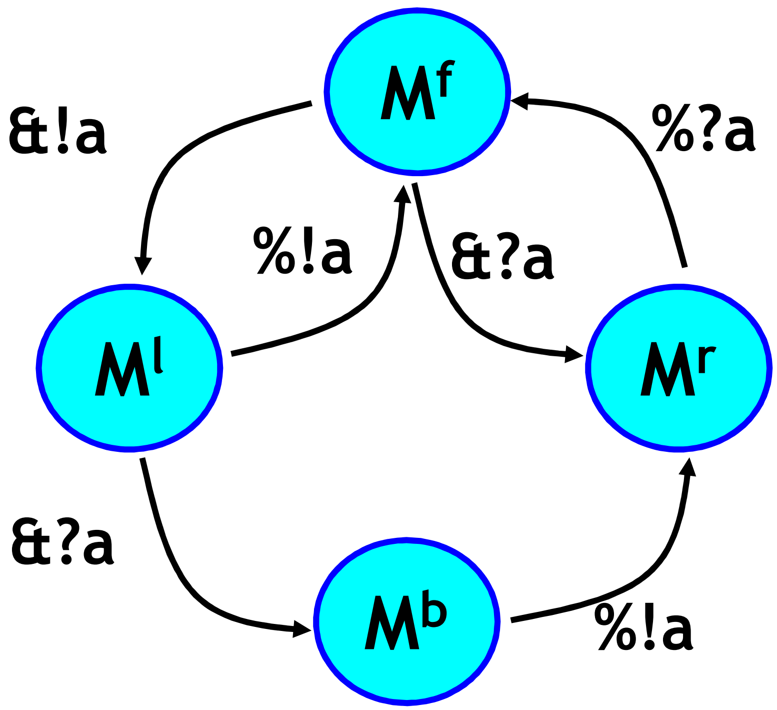


# Example: Actin Polymerization

Grows only to the right, shrinks only from the left



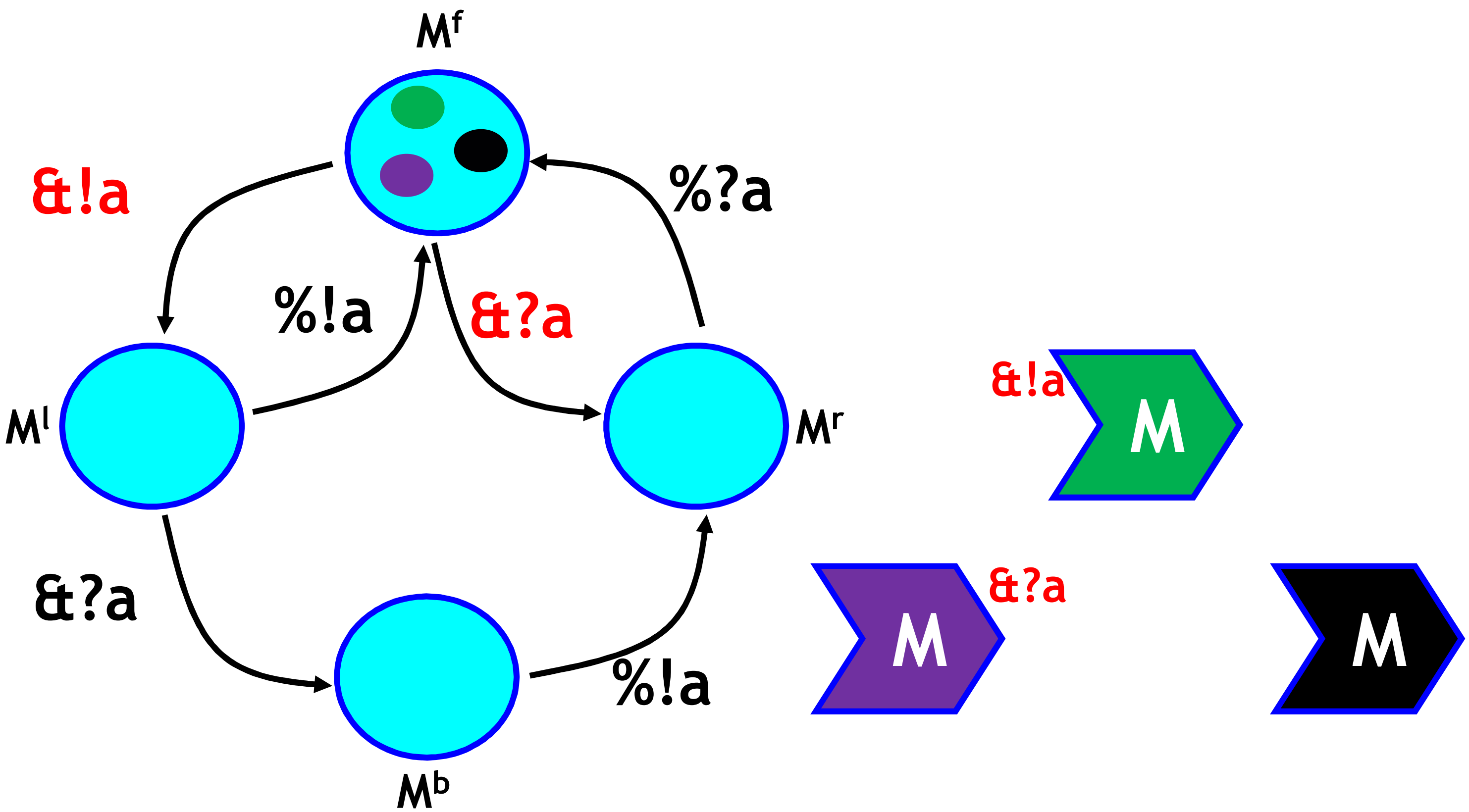

$M^f$ = free on both sides
$M^l$ = bound on the left
$M^r$ = bound on the right
$M^b$ = bound on both sides



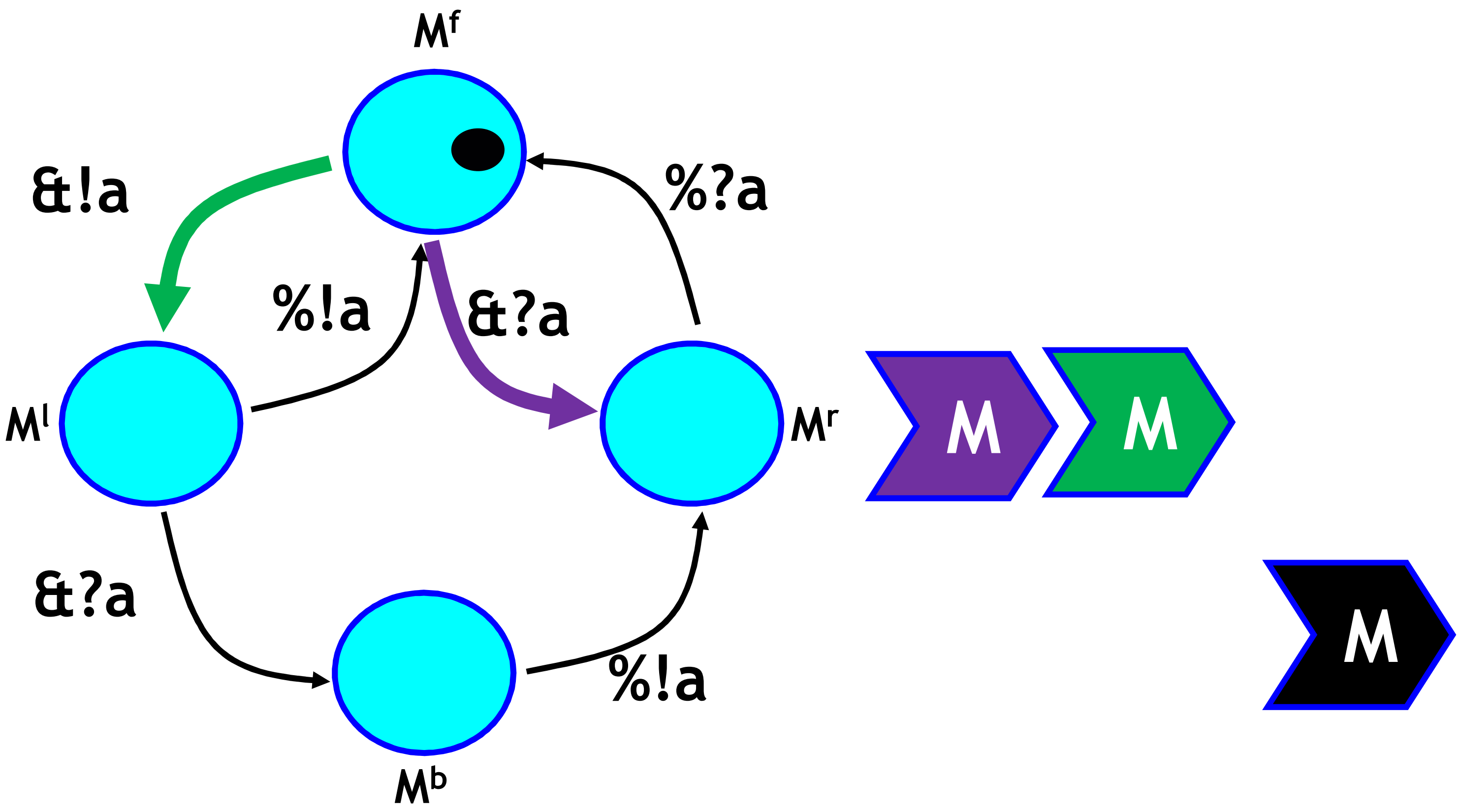

$$M^f = \&!a;\ M^l \oplus \&?a;\ M^r$$
$$M^l = \%!a;\ M^f \oplus \&?a;\ M^b$$
$$M^r = \%?a;\ M^f$$
$$M^b = \%!a;\ M^r$$

# Example: Actin Polymerization

- Purple associates with green

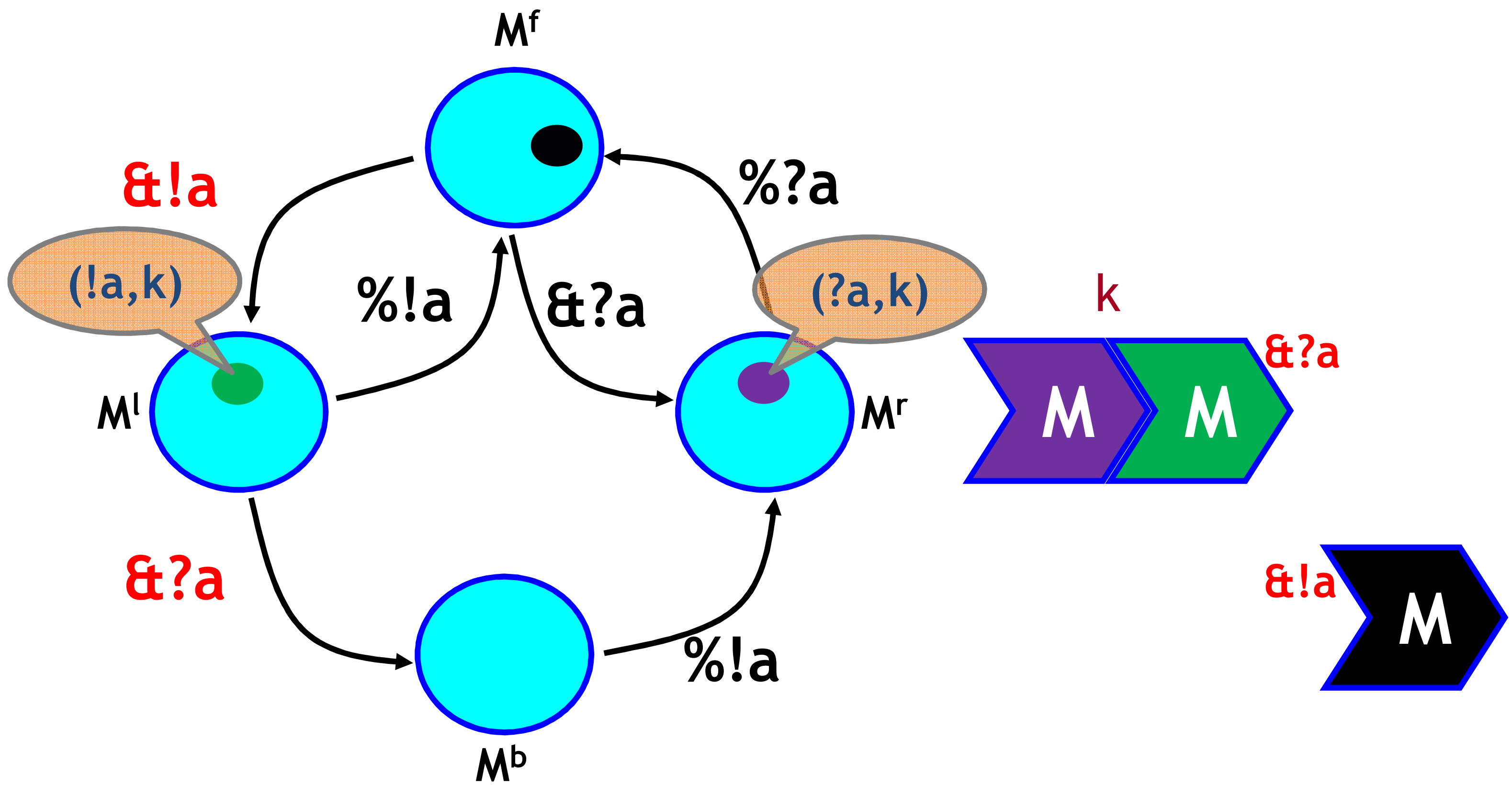


# Example: Actin Polymerization



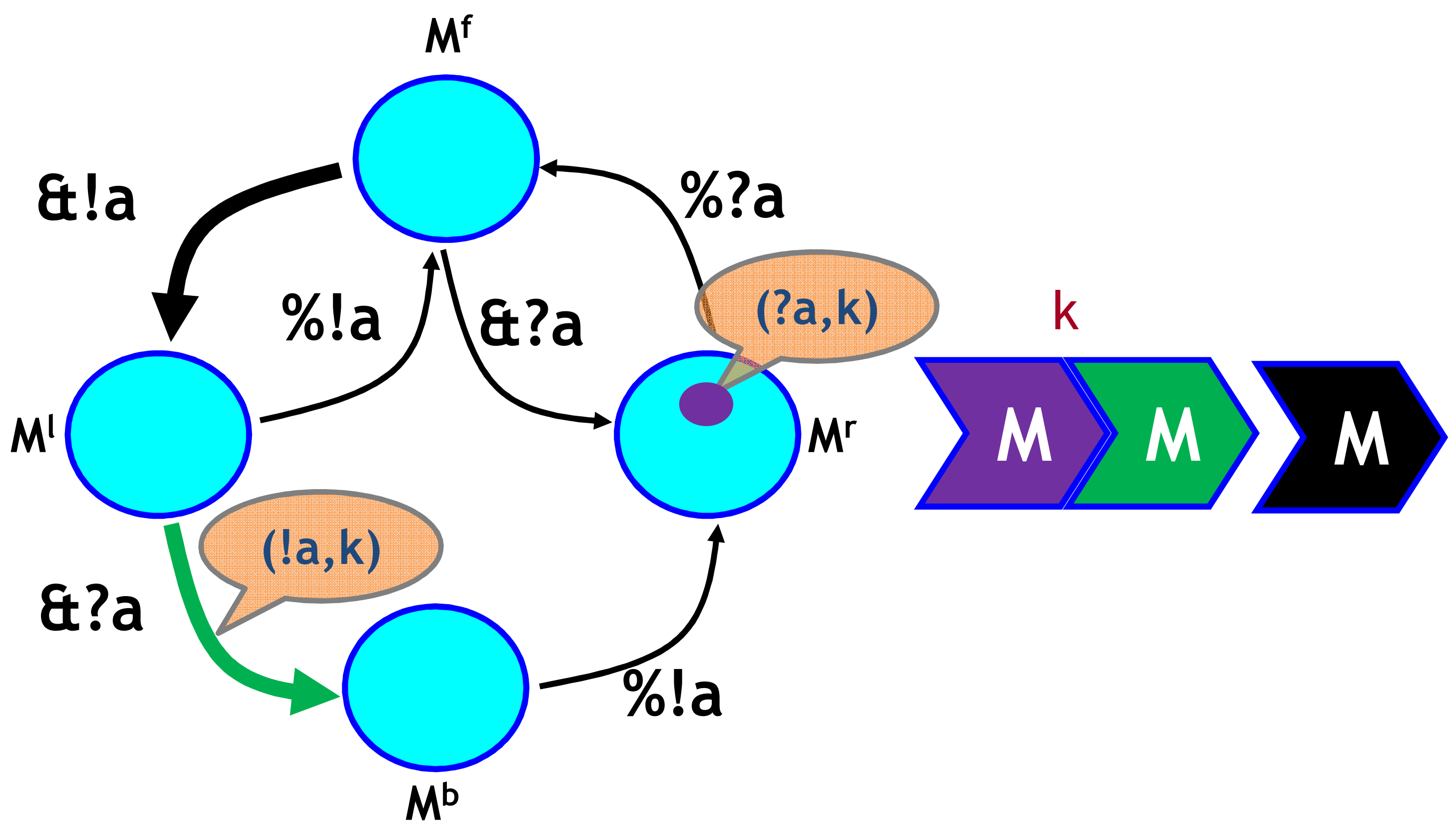
# Example: Actin Polymerization

- Each association has a unique key
  - Keys are stored in the molecule's history
- Black cannot associate with purple
  - No complementary actions available, enforcing the "grow only to the right" constraint



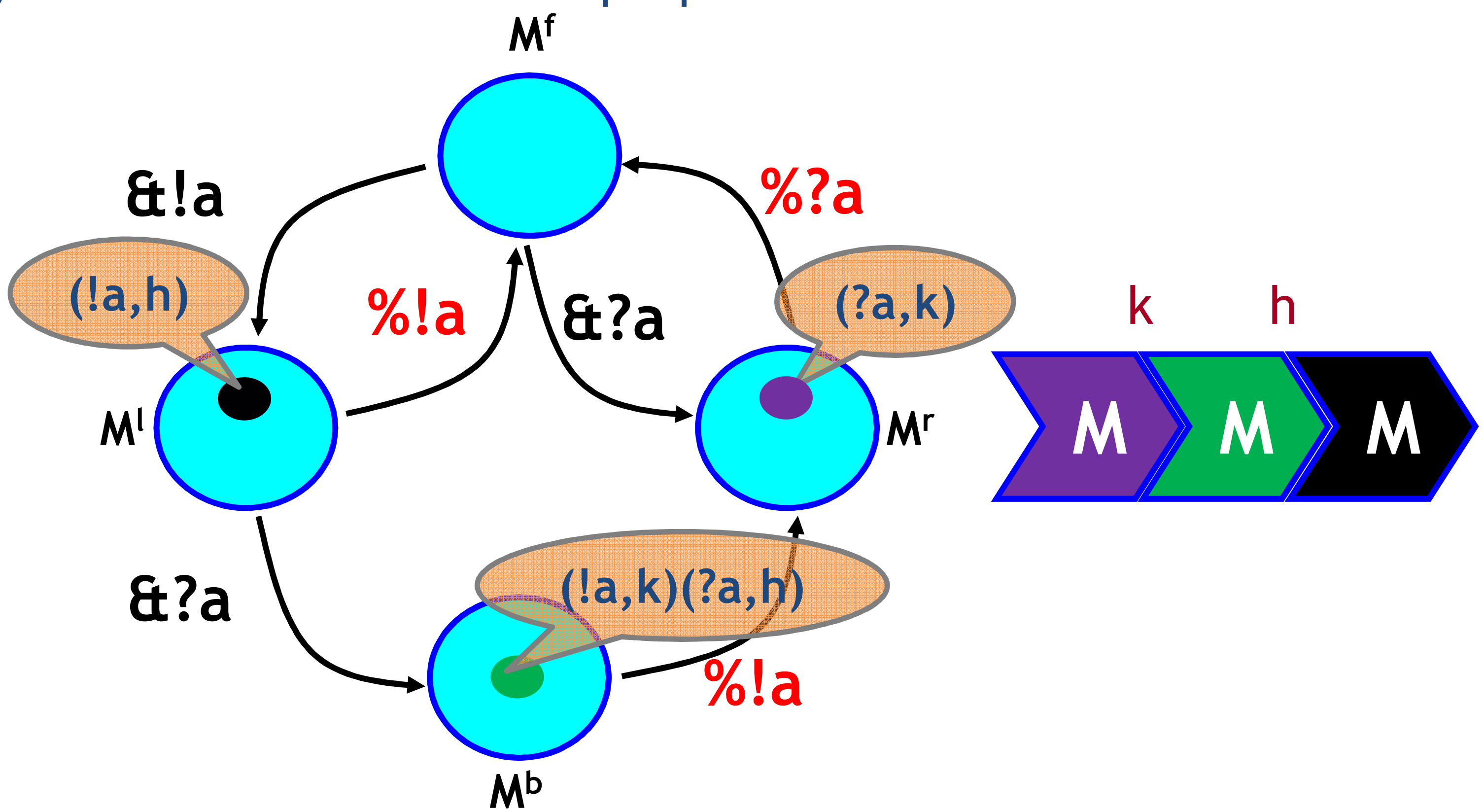
# Example: Actin Polymerization

- Green associates with black



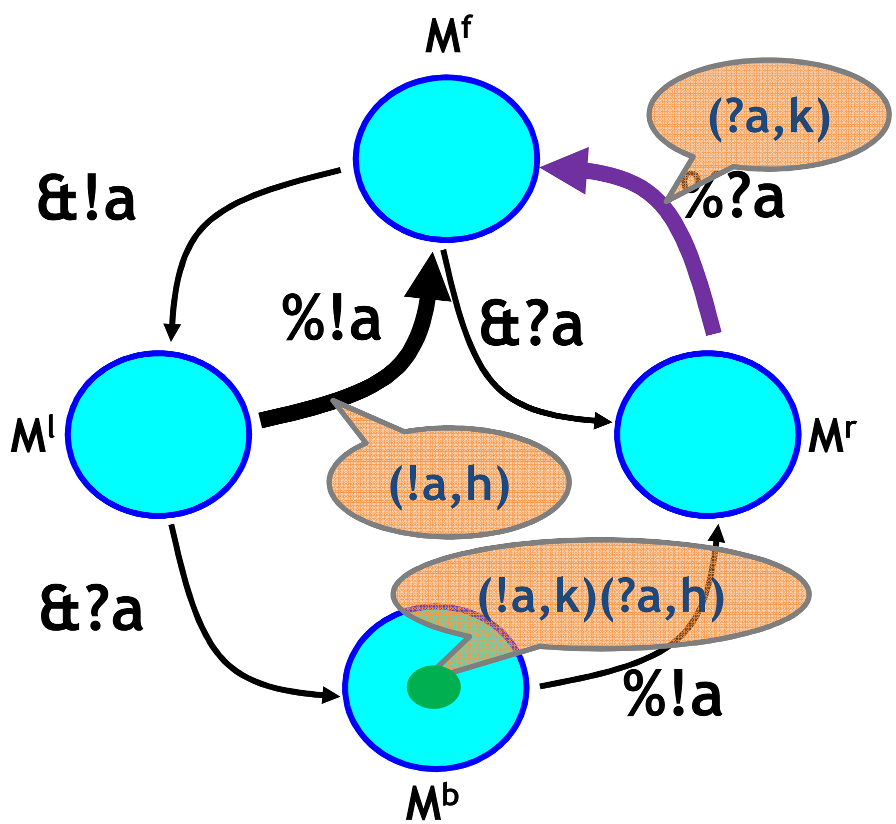
# Example: Actin Polymerization

- Black cannot dissociate from green
  - No complementary actions available, enforcing the “shrink only from left” constraint
- But black can dissociate from purple (really?)
- And green can dissociate from purple

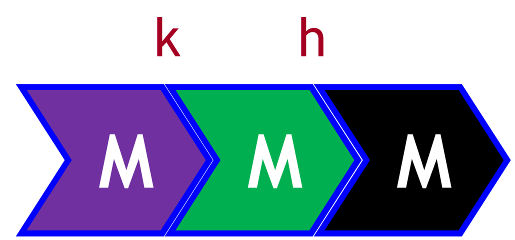


# Example: Actin Polymerization

- No, black cannot dissociate from purple
  - The association history prevents it

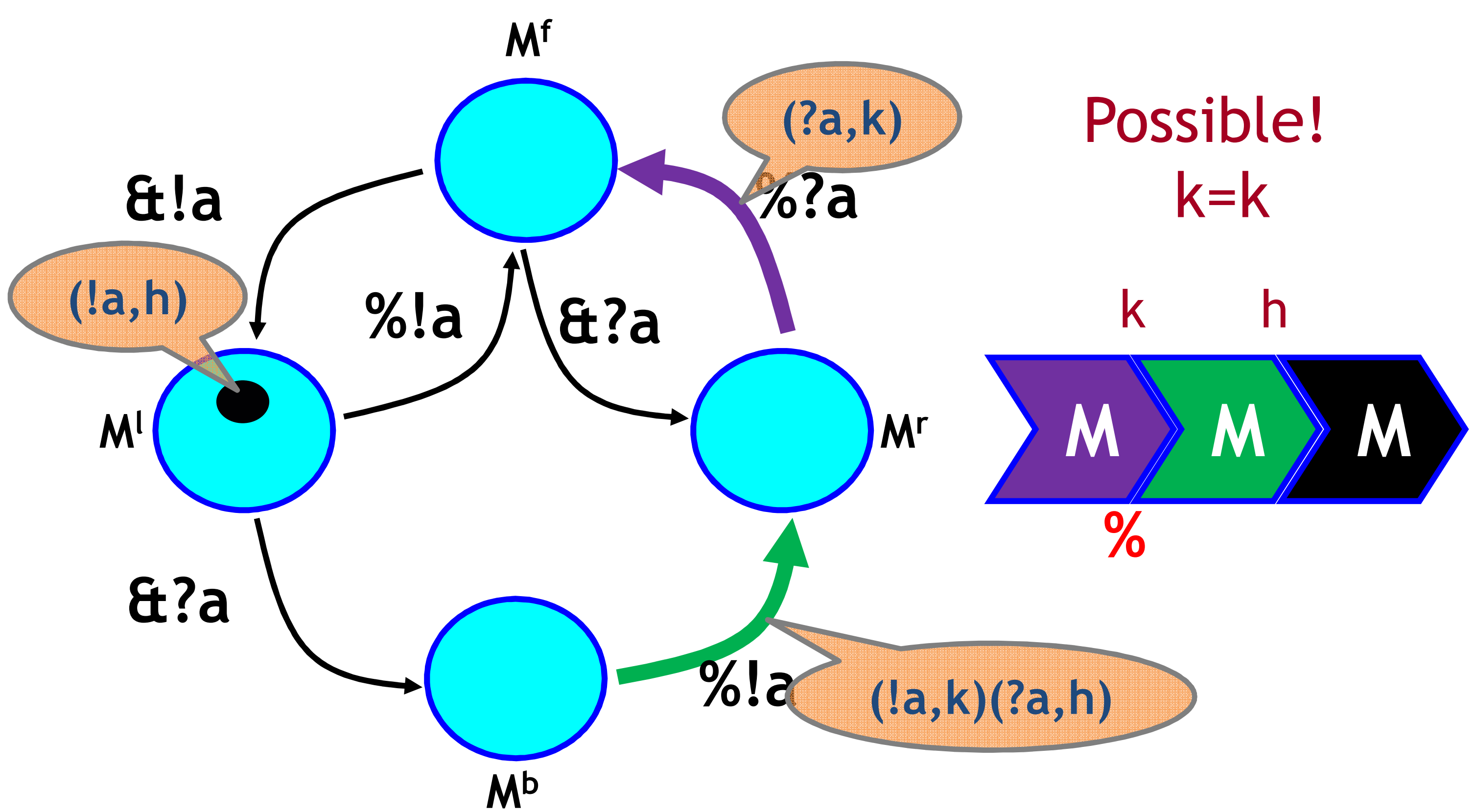



Not possible!
$h \neq k$



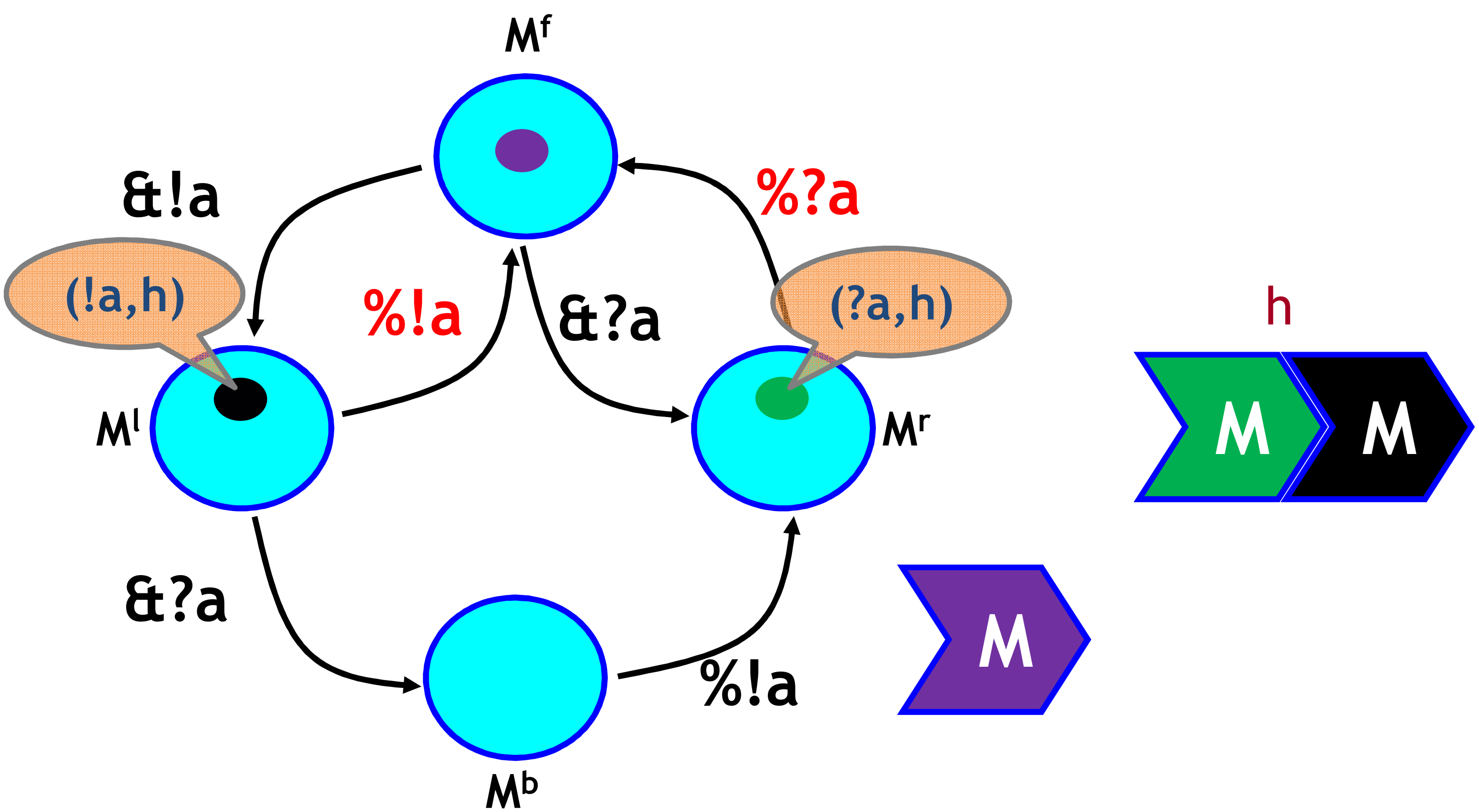
# Example: Actin Polymerization

- Purple dissociates from green



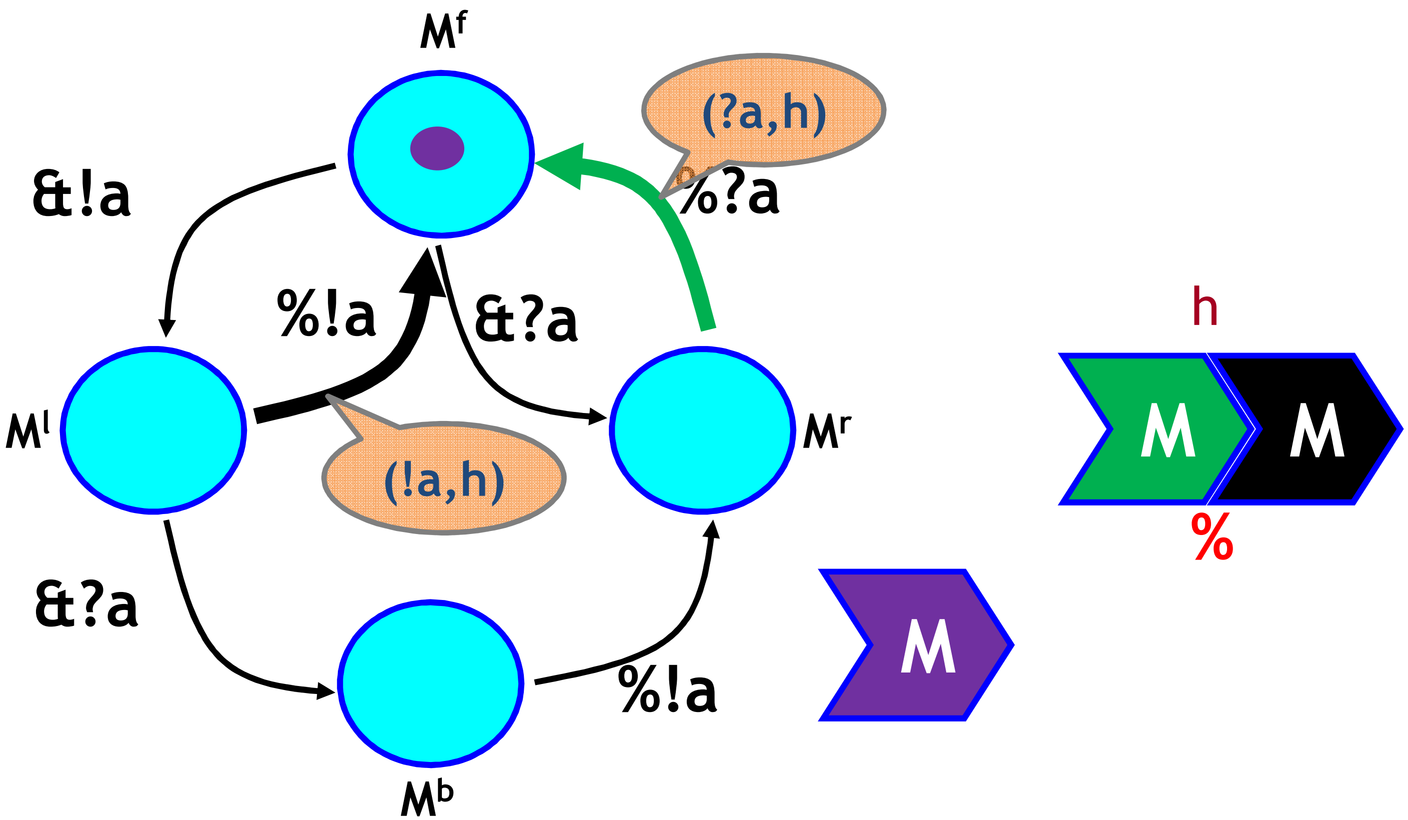
# Example: Actin Polymerization

- Now purple could reassociate to black
  on the other side, but we are not going to do that



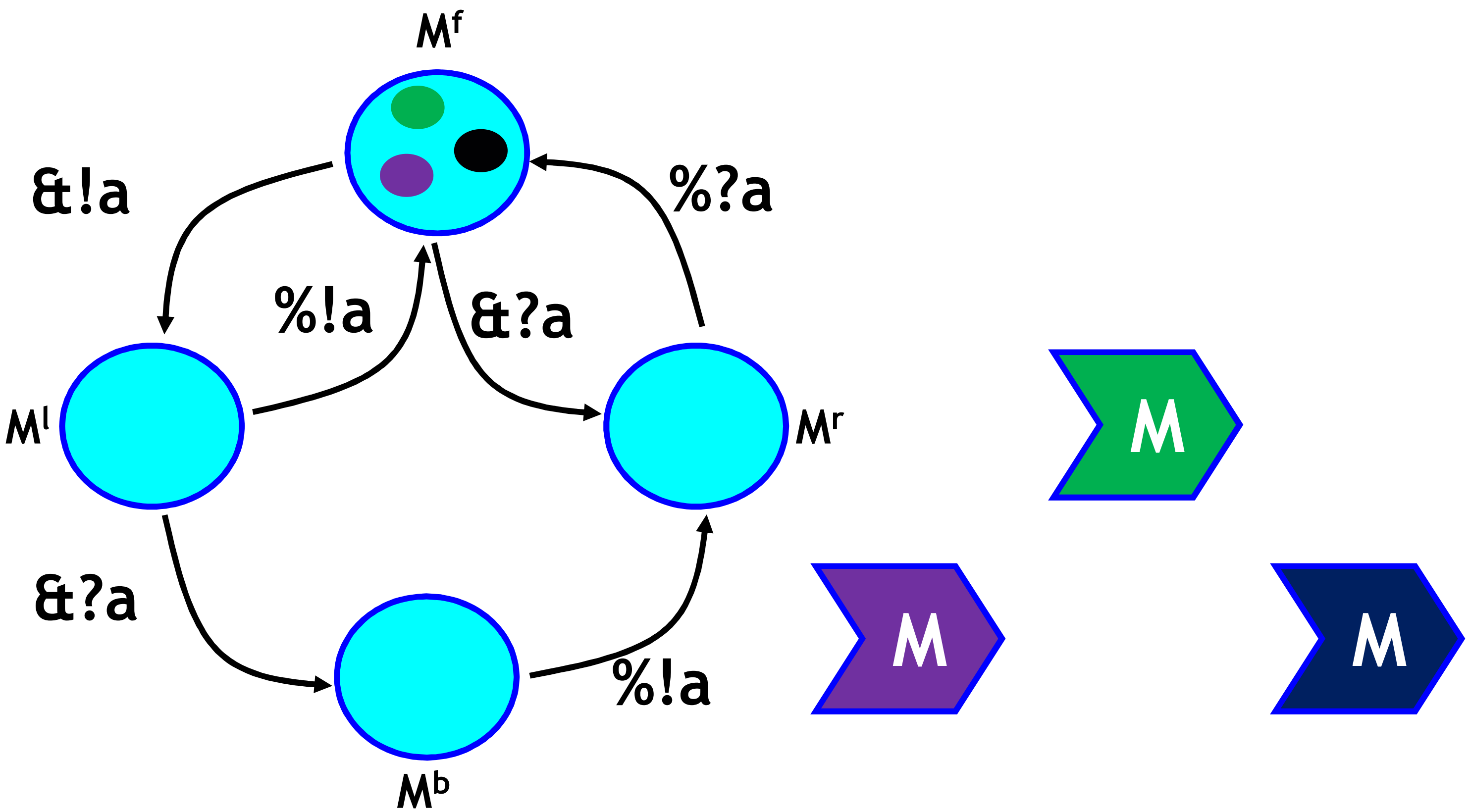
# Example: Actin Polymerization

- Green dissociates from black



# Example: Actin Polymerization

- Ready to start again



# Basic Biochemistry can Compute

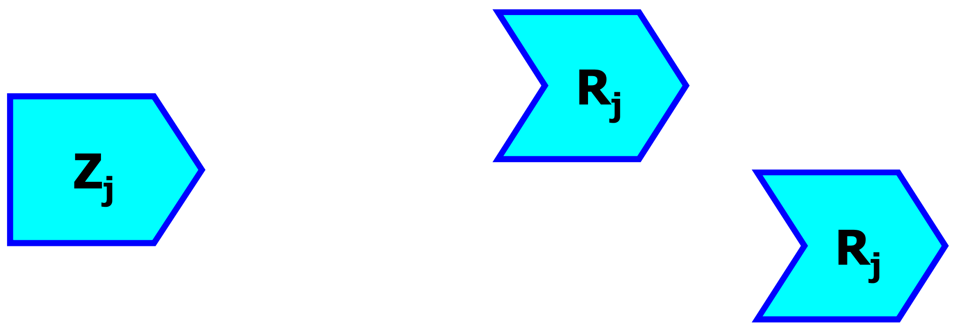
# Turing completeness of BGF

- Random Access Machines: [Min67]
  - **Registers**: $r_1 \ldots r_n$ hold natural numbers (unbounded)
  - **Program**: finite sequence of numbered instructions
    - **i: Inc($r_j$)**: add 1 to the content of $r_j$ and go to the next instruction
    - **i: DecJump($r_j$,s)**: if the content of $r_j$ is not 0 then decrease by 1 and go to the next instruction; otherwise jump to instruction s

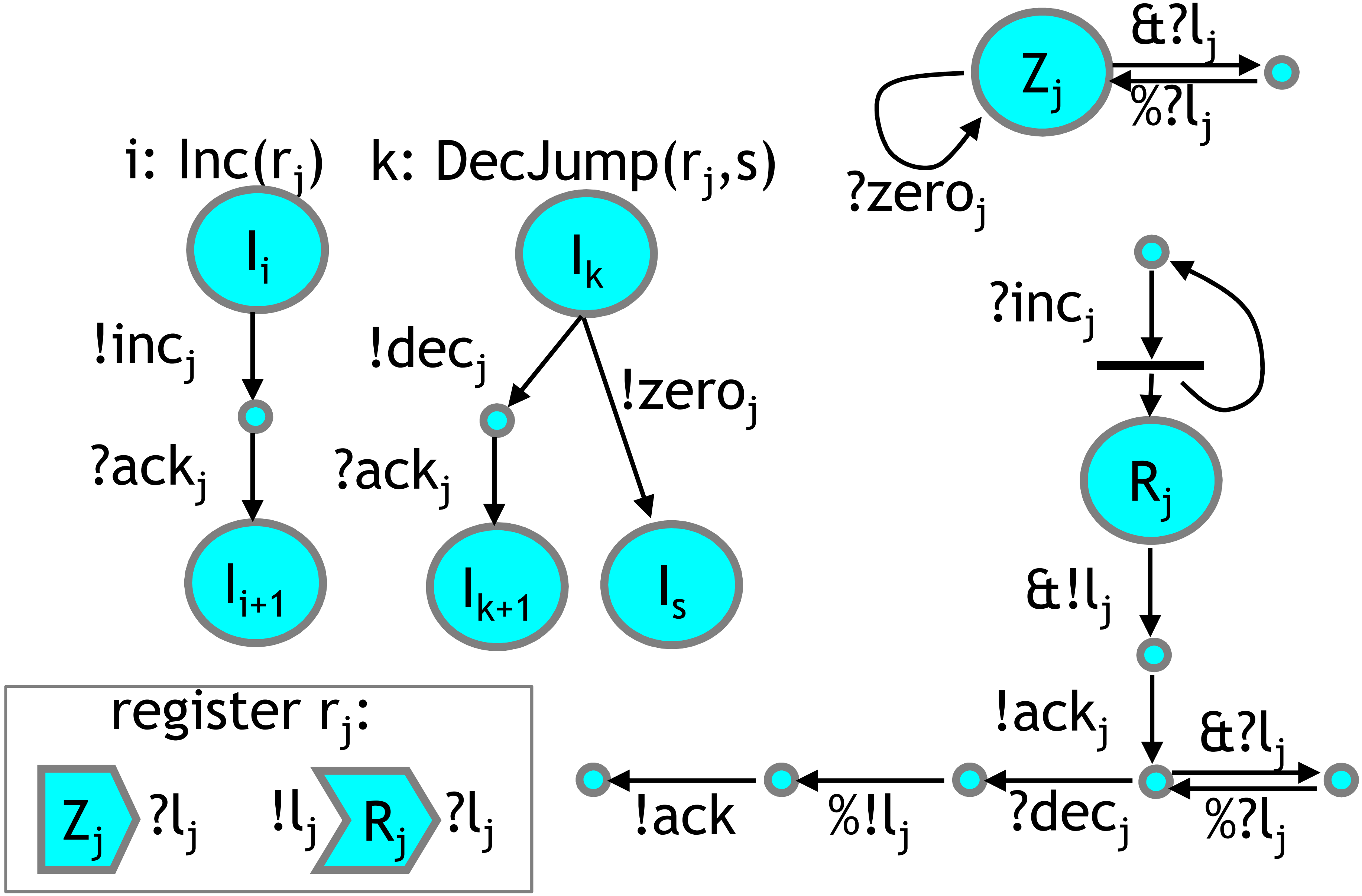
- There is a RAM encoding in BGF
  - But not, as we already showed, in CGF.
  - (Hence it is not possible to compile BGF to CGF.)

# Registers as Polymers

- Initially empty register $r_j$: a seed $\mathbf{Z_j}$
- Increment on $r_j$: produce a new monomer and associate it to the polymer
- Decrement on $r_j$: remove last monomer

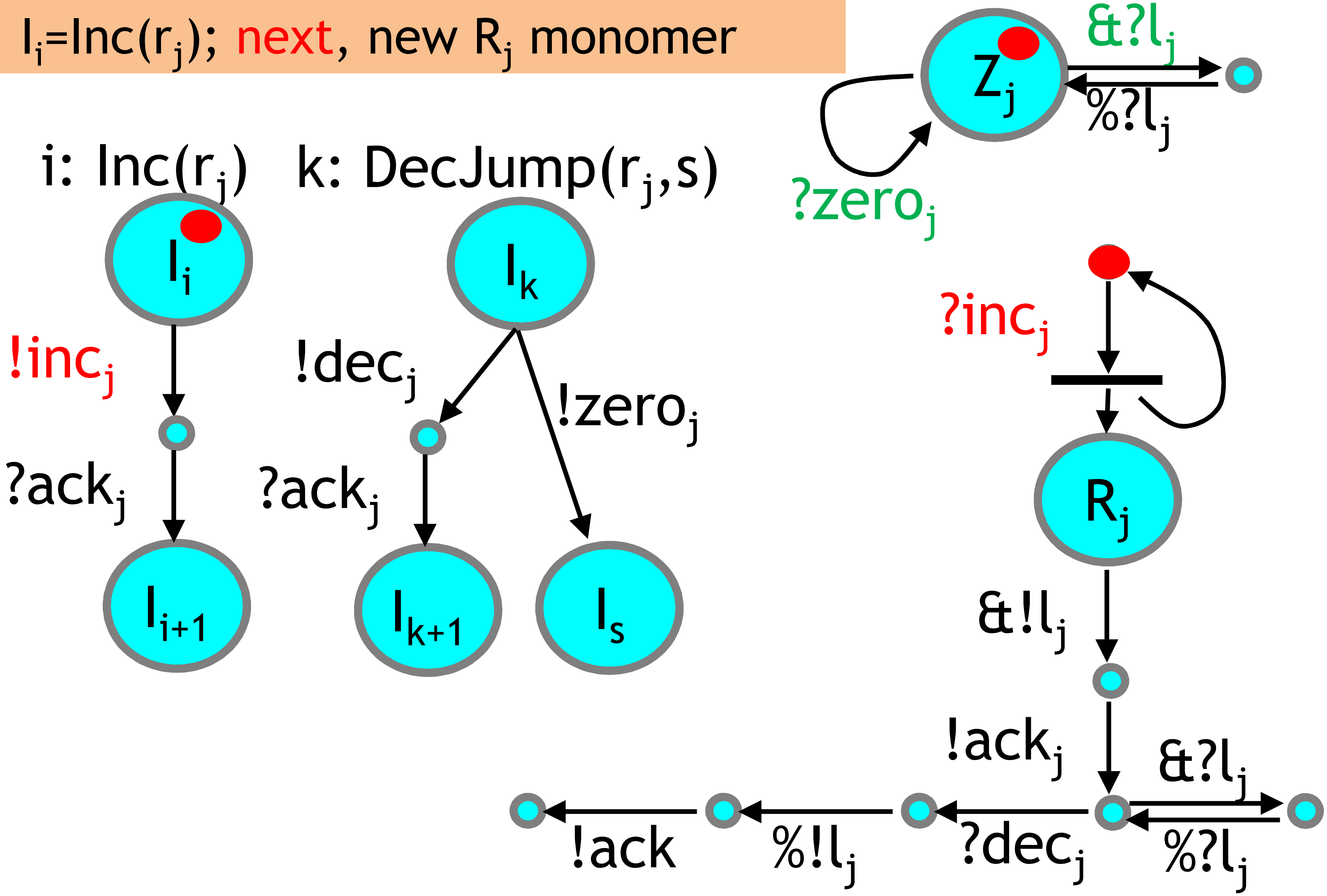


# RAM encoding in BGF



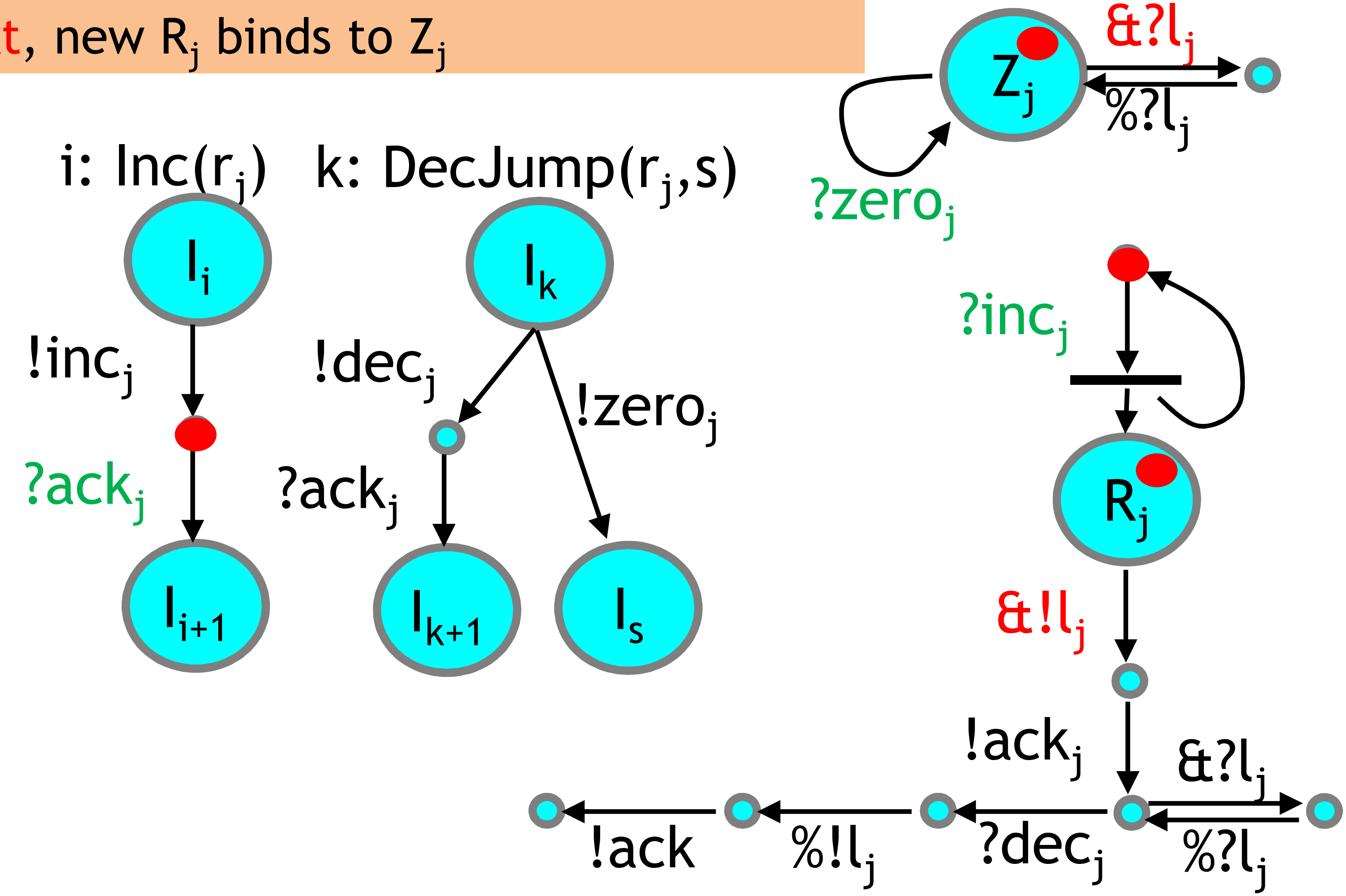
# RAM encoding in BGF

$r_j$=0; $I_i$=Inc($r_j$); next, new $R_j$ monomer



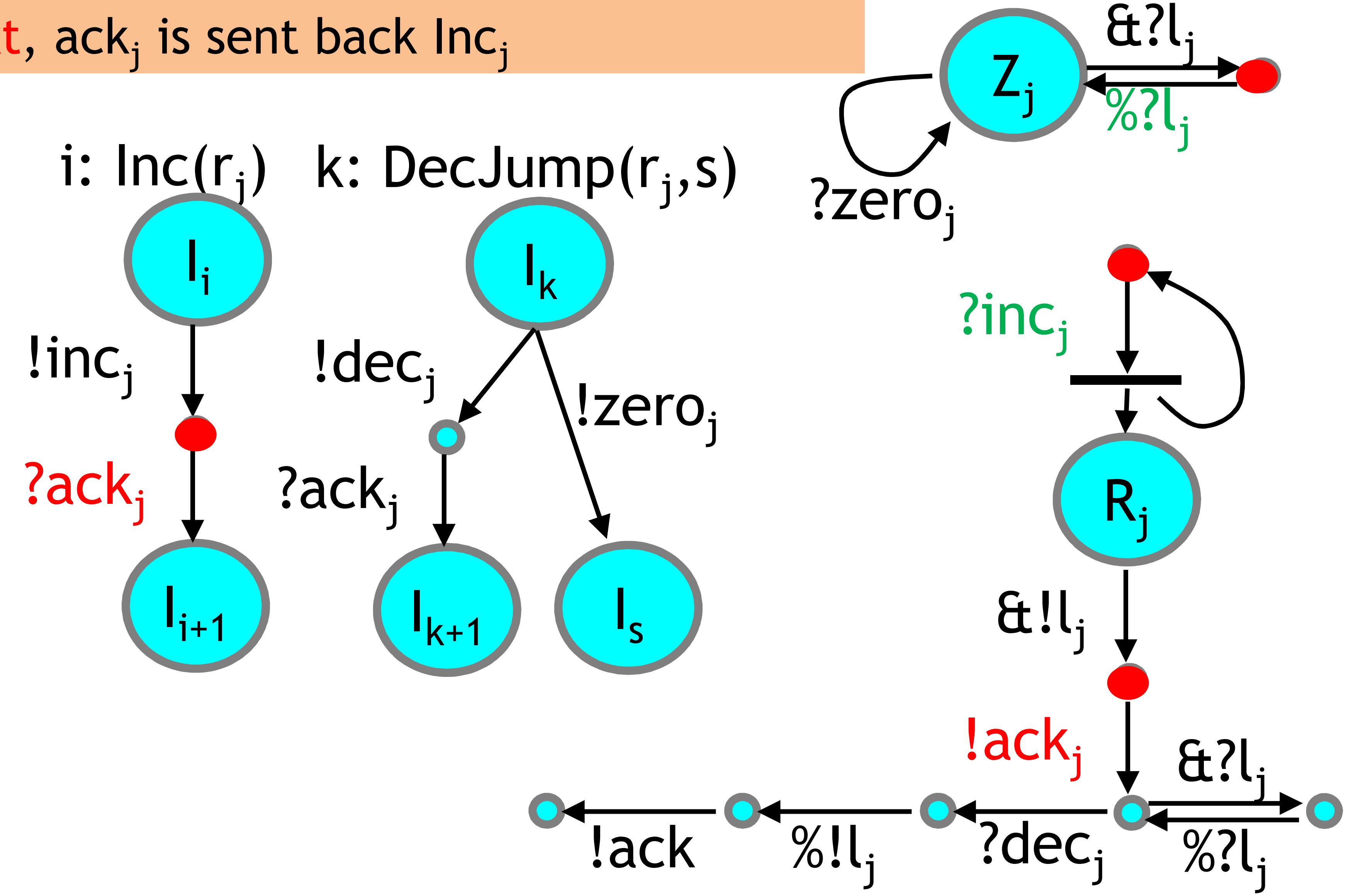
# RAM encoding in BGF

next, new $R_j$ binds to $Z_j$



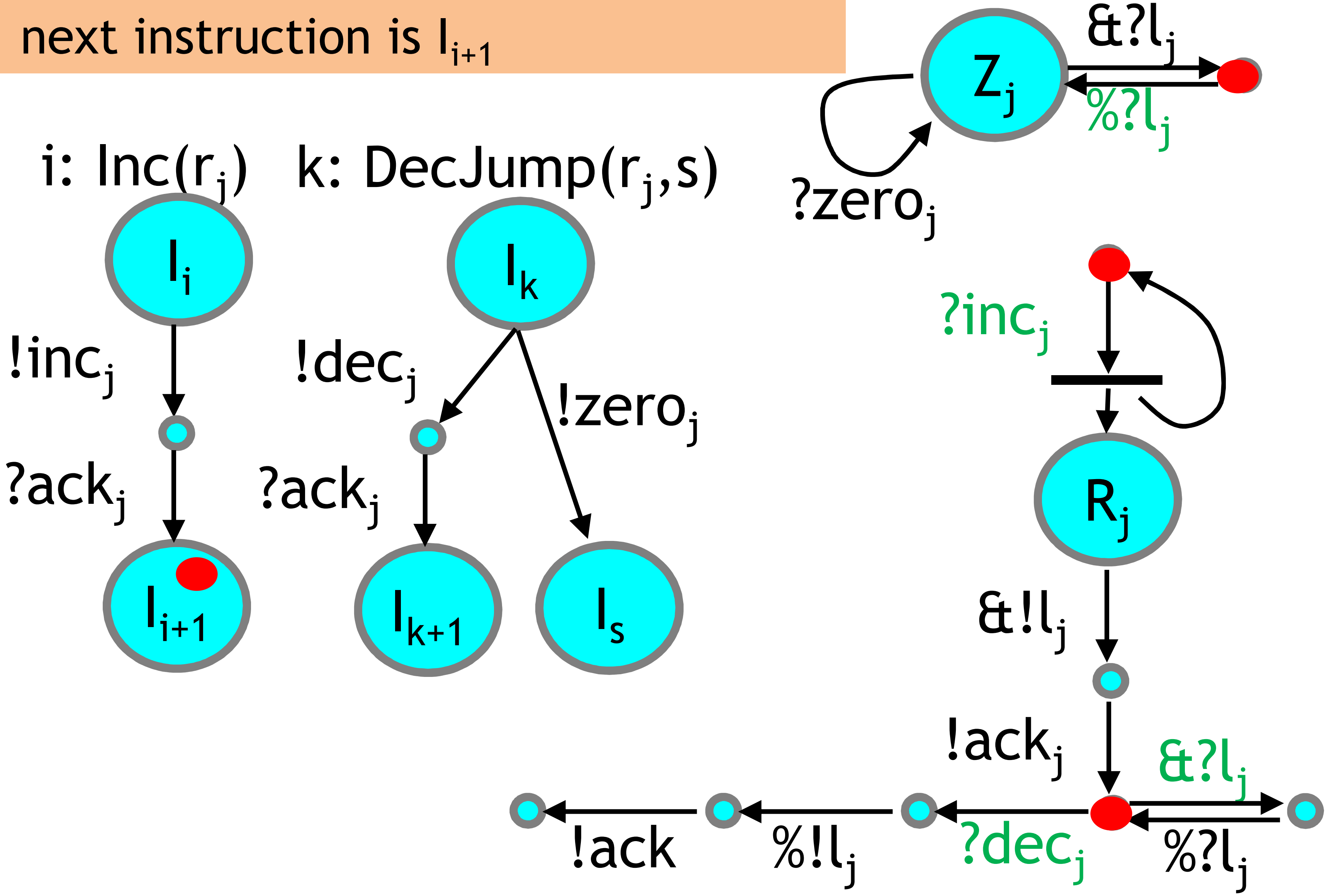
# RAM encoding in BGF

next, $ack_j$ is sent back $Inc_j$



# RAM encoding in BGF

$r_j$=1; next instruction is $I_{i+1}$



# Conclusions

# Conclusions

- Chemistry (CGF) is not Turing complete
  - It is decidable weather given a molecule will be produced.
  - Surprisingly (since this is decidable nondeterministically), it is undecidable whether a program will terminate with probability measure 1.
  - However, chemistry can (slowly) approximate a Turing machine to any degree of precision: it is undecidable whether a given molecule is *likely* to be produced.

- Biochemistry (BGF) is Turing complete.
  - Of course, $\pi$-calculus is Turing complete too, but it contains operators that do not have a direct biological interpretation.
  - The BGF a minimal extension of chemistry with biologically inspired operators (complexation/decomplexation) and is already Turing complete
  - Finite Turing-powerful programming constructs can be found in biochemistry but not in basic chemistry.

# Conclusions

- A theoretical result
  - Basic Biochemistry > Basic Chemistry
    (should please the biologists…)
- Some practical modeling implications:
  - A finite model in BGF (e.g. of polymerization)
    may correspond to an infinite model in FSRN
  - A model in BGF (e.g. of multiple protein phosphorylation states)
    may correspond to an $O(2^n)$ bigger model in FSRN
  - Even a model in CGF
    may correspond to an $O(n^2)$ bigger model in FSRN
- Process algebra modeling leads to:
  - Compact model presentation
  - Component-based modeling
  - Compositional (separate-subsystems) modeling