“No we can’t”

Impossibility of fast leader election in chemical reaction networks

David Doty

Workshop on Advances in Numerical and Analytic Approaches for the Study of Non-Spatial Stochastic Dynamical Systems in Molecular Biology

WoAiNaAAftSoNSSDSiMB 2016!

Isaac Newton Institute of Mathematical Sciences, University of Cambridge
April 5, 2016

Acknowledgments

co-author

David Soloveichik
University of Texas, Austin

Monir Hajiaghayi

Anne Condon

David Anderson

Banff International Research Station
for Mathematical Innovation and Discovery

Acknowledgments

co-author

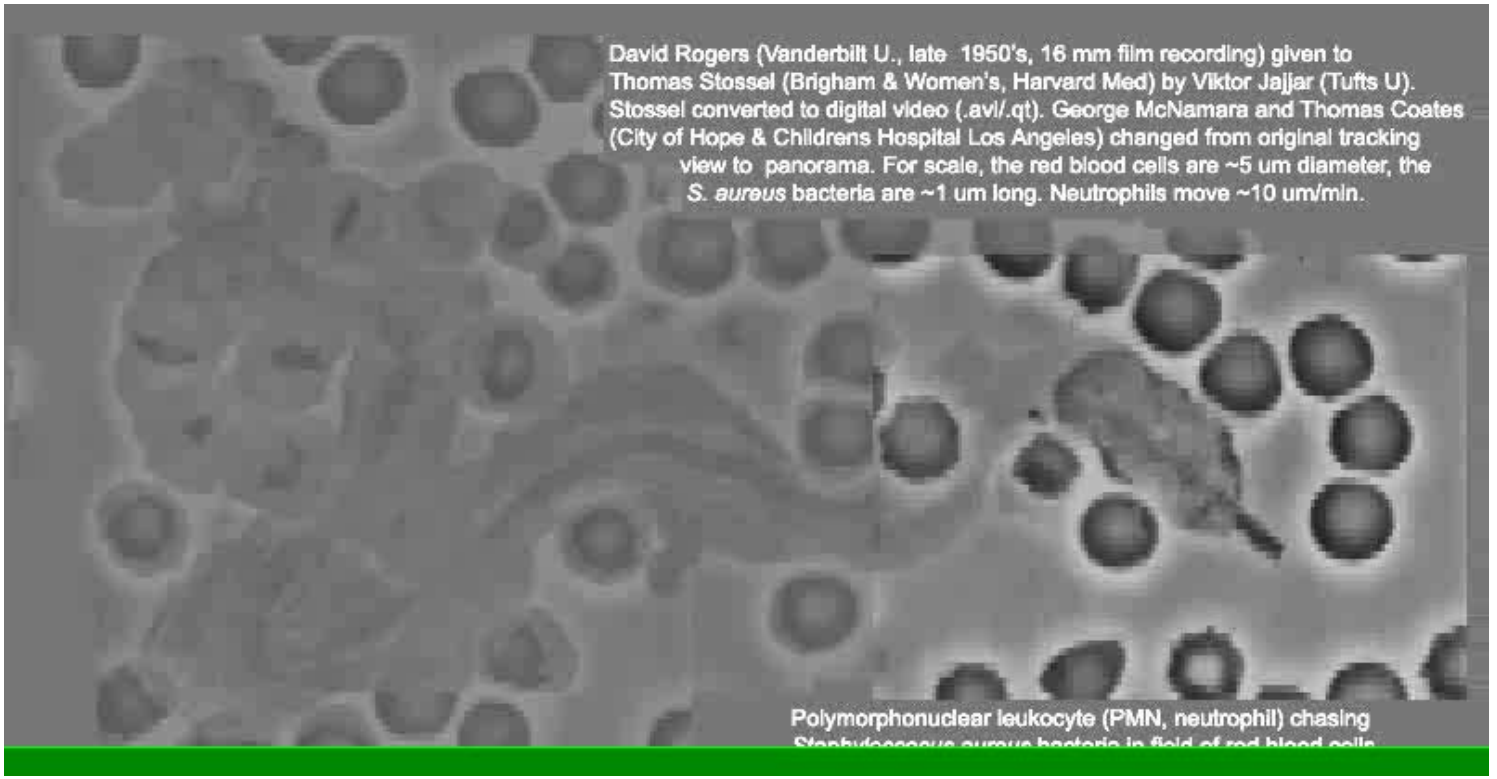
David Soloveichik
University of Texas, Austin

Monir Hajiaghayi

Anne Condon

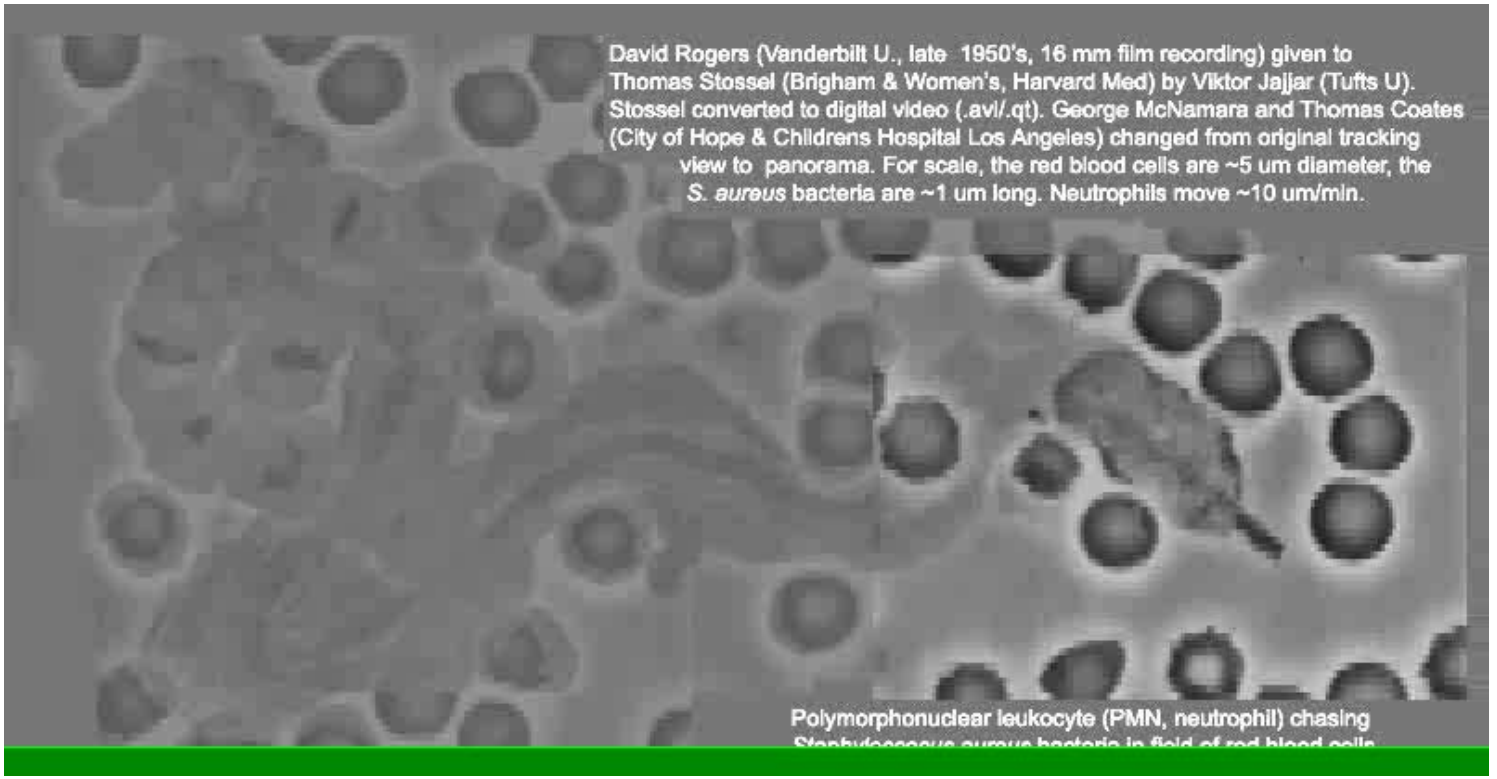
David Anderson

The Software of Life

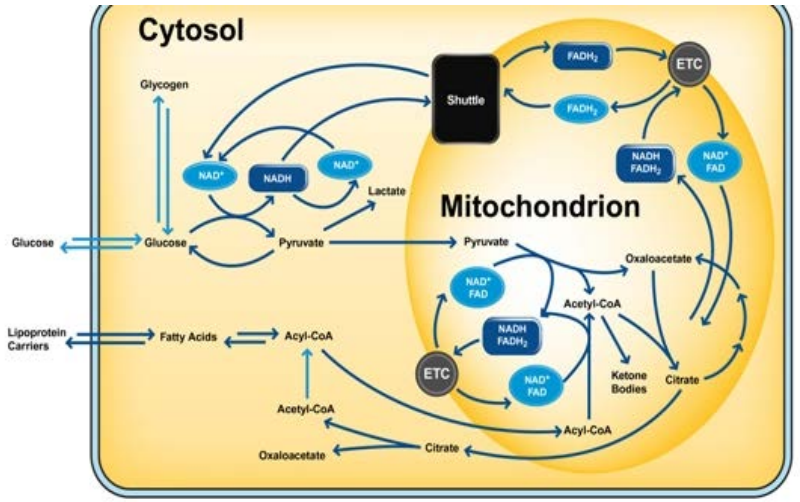


How does a cell compute?

The Software of Life

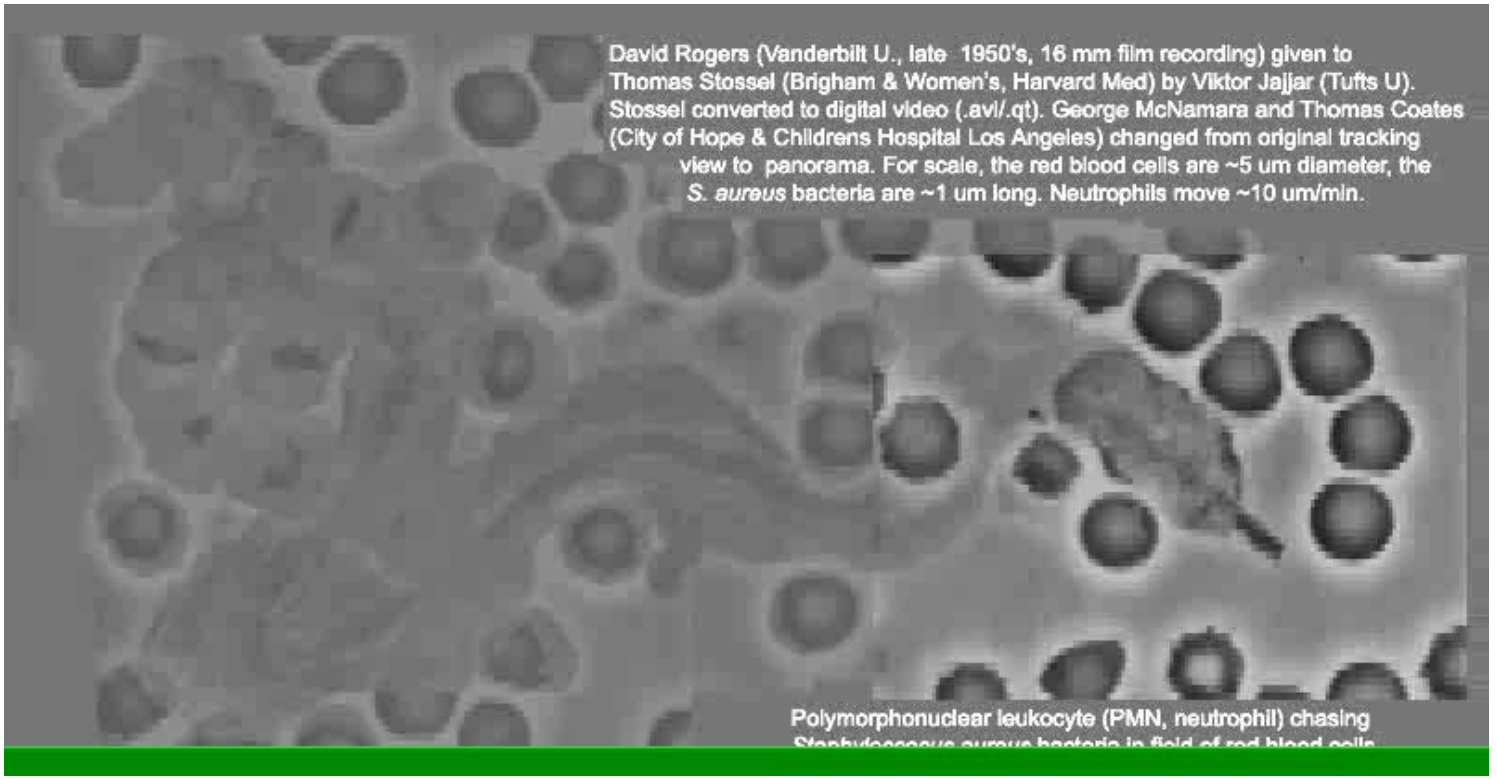


How does a cell compute?

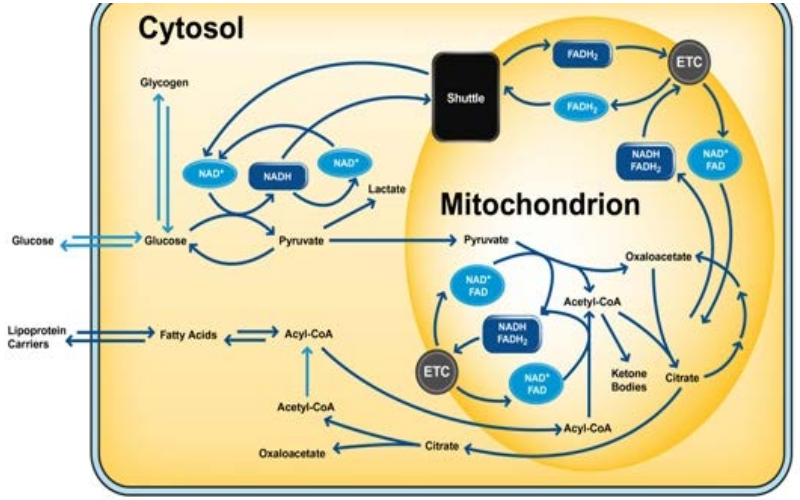


chemistry / geometry

The Software of Life



~~How does a cell compute?~~



What is possible to compute using chemistry ? ~~/ geometry~~

Chemical Reaction Network (CRN)

Chemical Reaction Network (CRN)

$$R \rightarrow P_1 + P_2$$

Chemical Reaction Network (CRN)

$$R \rightarrow P_1 + P_2$$

$$M_1 + M_2 \rightarrow D$$

Chemical Reaction Network (CRN)

$$R \rightarrow P_1 + P_2$$

$$M_1 + M_2 \rightarrow D$$

$$C + X \rightarrow C + Y$$

Chemical Reaction Network (CRN)

$$R \rightarrow P_1 + P_2$$

$$M_1 + M_2 \rightarrow D$$

$$C + X \rightarrow C + Y$$

This is traditionally a descriptive modeling language…
let's instead use it as a prescriptive programming language

Can we compute with chemistry?

"Not every CRN describes real chemicals!", i.e. "where's the *compiler*?"

Can we compute with chemistry?

"Not every CRN describes real chemicals!", i.e. "where's the *compiler*?"

Response: Soloveichik *et al.* [*PNAS* 2011] showed how to physically implement any CRN using *DNA strand displacement*

Can we compute with chemistry?

"Not every CRN describes real chemicals!", i.e. "where's the *compiler*?"

Response: Soloveichik *et al.* [*PNAS* 2011] showed how to physically implement any CRN using *DNA strand displacement*

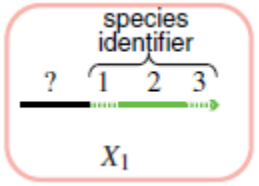
$$X_1 + X_2 \rightarrow X_3$$

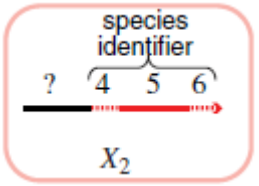
Can we compute with chemistry?

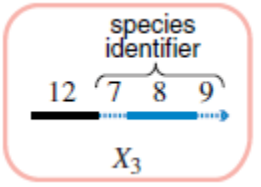
"Not every CRN describes real chemicals!", i.e. "where's the *compiler*?"

Response: Soloveichik *et al.* [*PNAS* 2011] showed how to physically implement any CRN using *DNA strand displacement*

$$X_1 + X_2 \rightarrow X_3$$





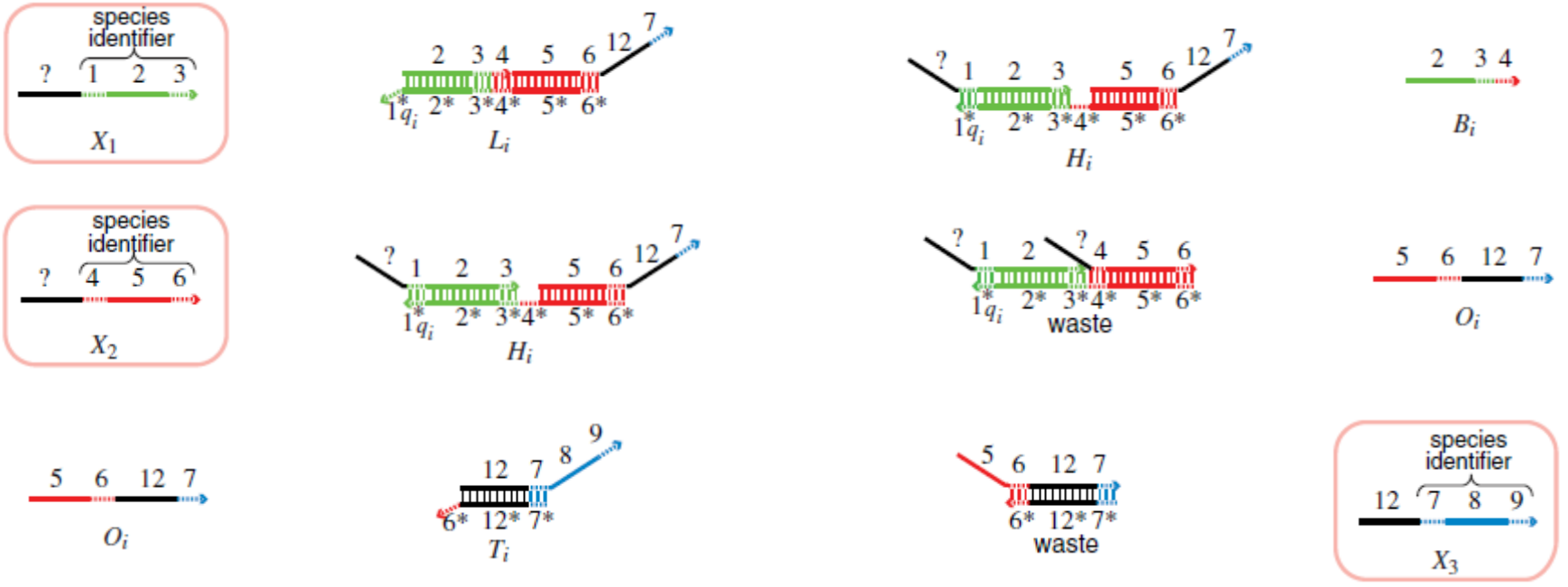


Can we compute with chemistry?

"Not every CRN describes real chemicals!", i.e. "where's the *compiler*?"

Response: Soloveichik *et al.* [*PNAS* 2011] showed how to physically implement <u>any</u> CRN using *DNA strand displacement*

$$X_1 + X_2 \rightarrow X_3$$

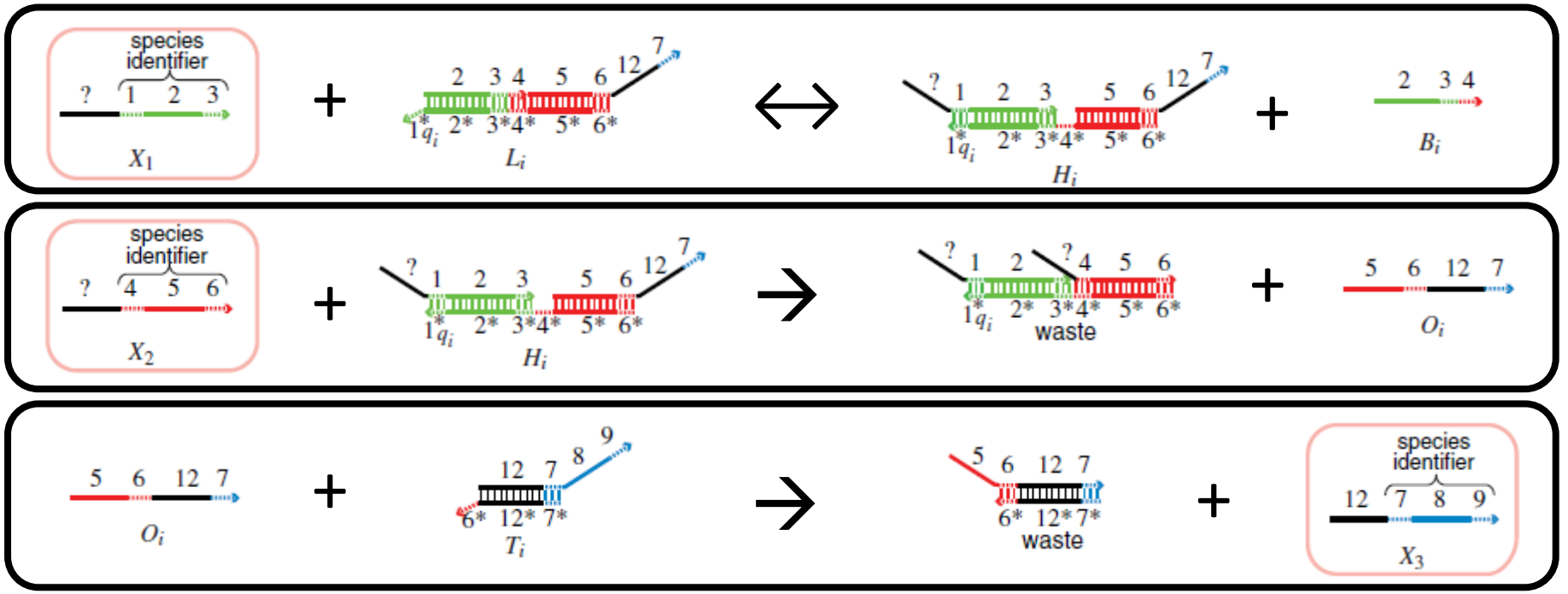


Can we compute with chemistry?

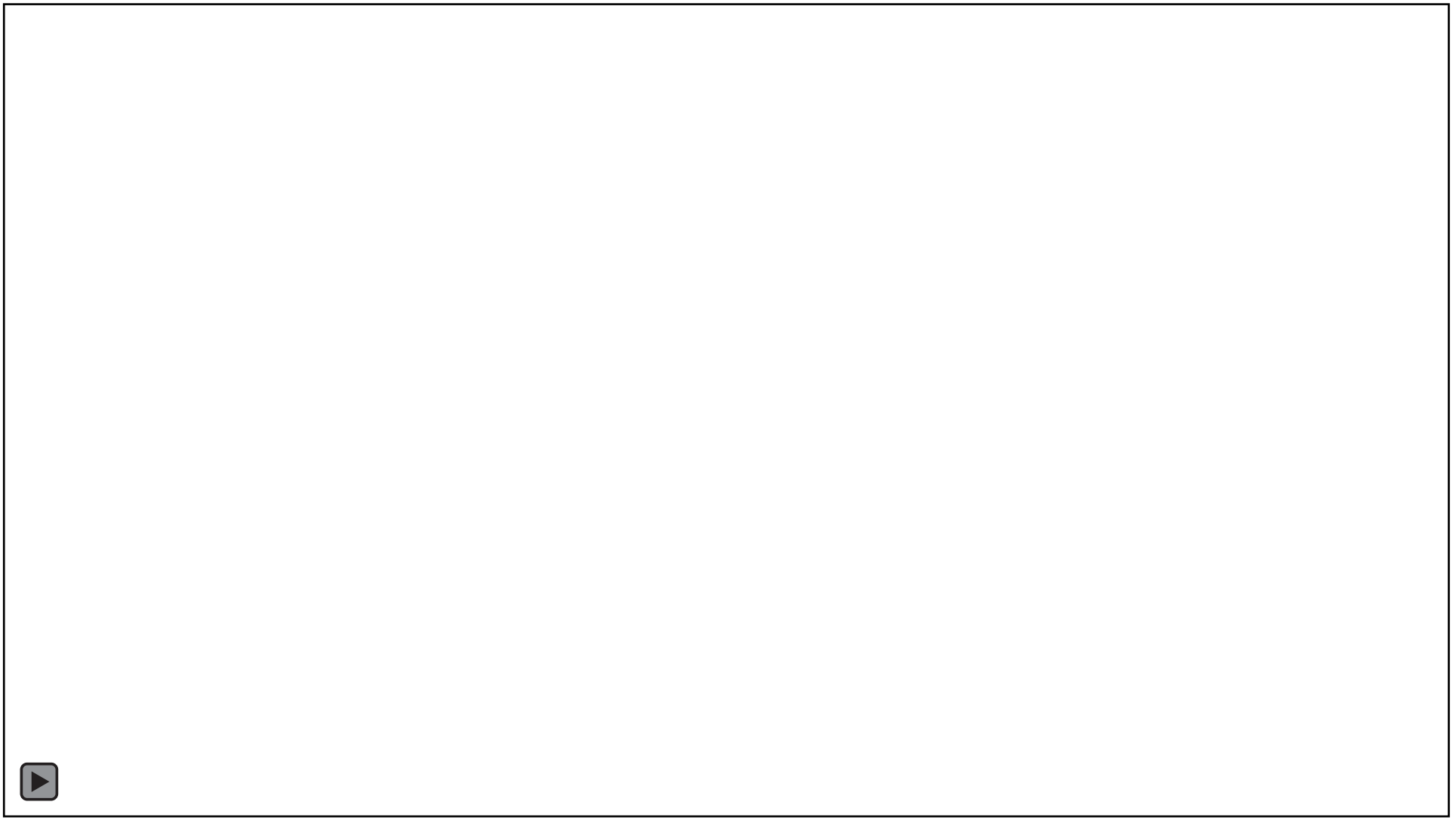
"Not every CRN describes real chemicals!", i.e. "where's the *compiler*?"

Response: Soloveichik *et al.* [*PNAS* 2011] showed how to physically implement <u>any</u> CRN using *DNA strand displacement*

$$X_1 + X_2 \rightarrow X_3$$

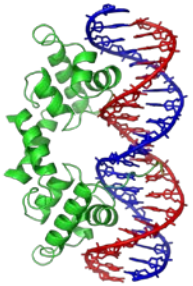


DNA strand displacement implementing A+B→C

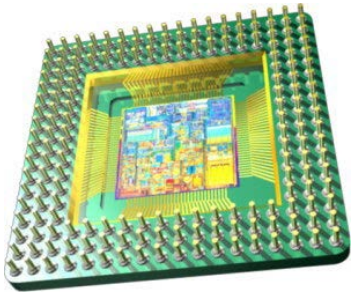


video courtesy Microsoft Research Cambridge

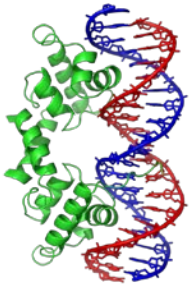
Why compute with chemistry?



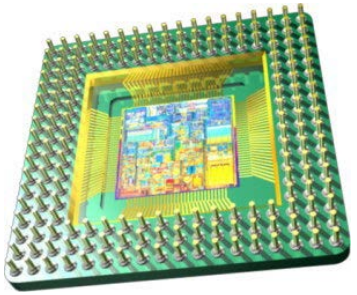
versus



Why compute with chemistry?

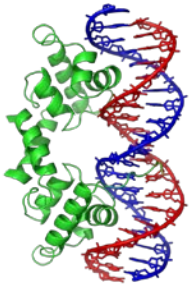


versus

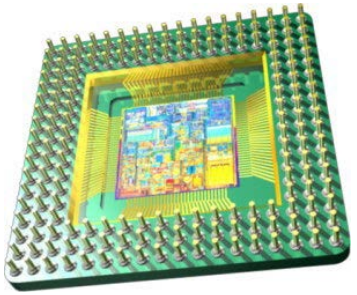


speed?

Why compute with chemistry?



versus

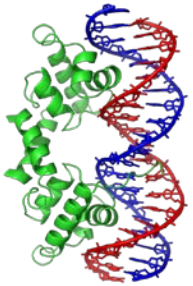


slow

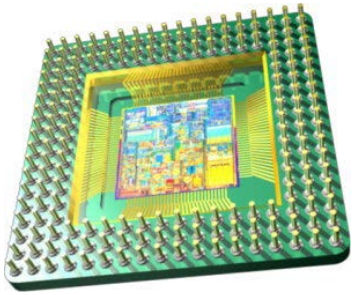
speed?

fast

Why compute with chemistry?



versus



slow

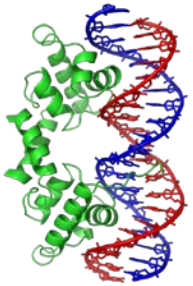
speed?

fast

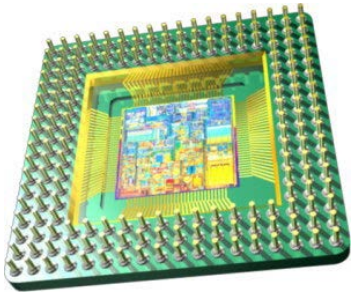
10-100nm

component size?

Why compute with chemistry?



versus



slow

10-100nm

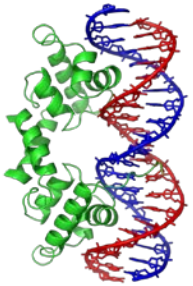
speed?

component size?

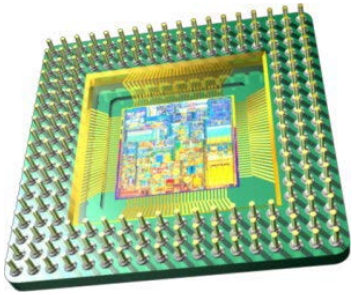
fast

10-100nm

Why compute with chemistry?

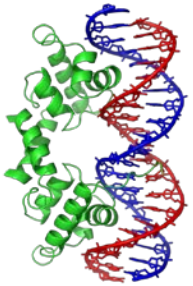


versus

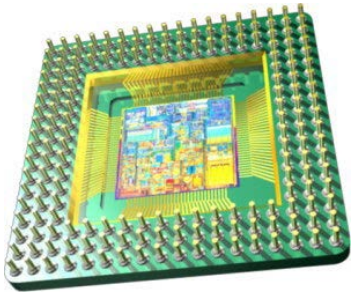


slow	speed?	fast
10-100nm	component size?	10-100nm
yes ✓	compatible with biological or other "wet" environments?	not easily

Why compute with chemistry?

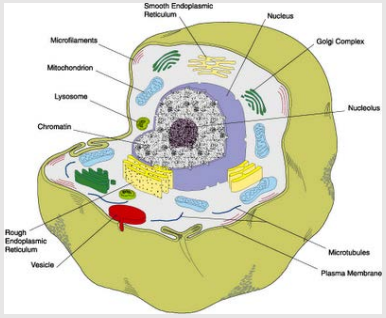


versus



slow	speed?	fast
10-100nm	component size?	10-100nm
yes ✓	compatible with biological or other "wet" environments?	not easily

cells



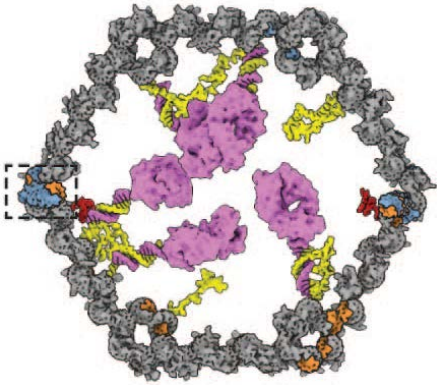
"smart" drug released only in certain cellular conditions

"chemical controller" to optimize yield of metabolically produced biofuels/drugs

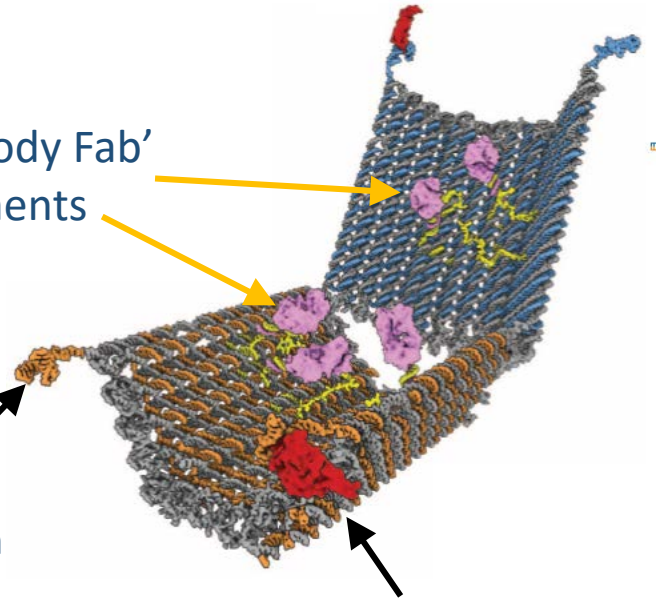
bioreactors

Why compute with chemistry?

opens to deliver only in presence of two protein antigens

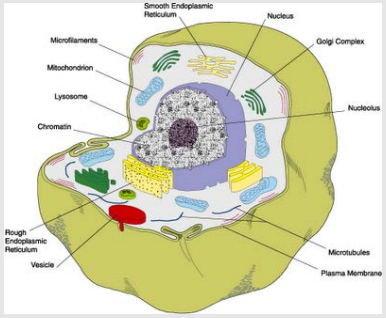


open



Douglas *et al*, *Science* 2012

cells



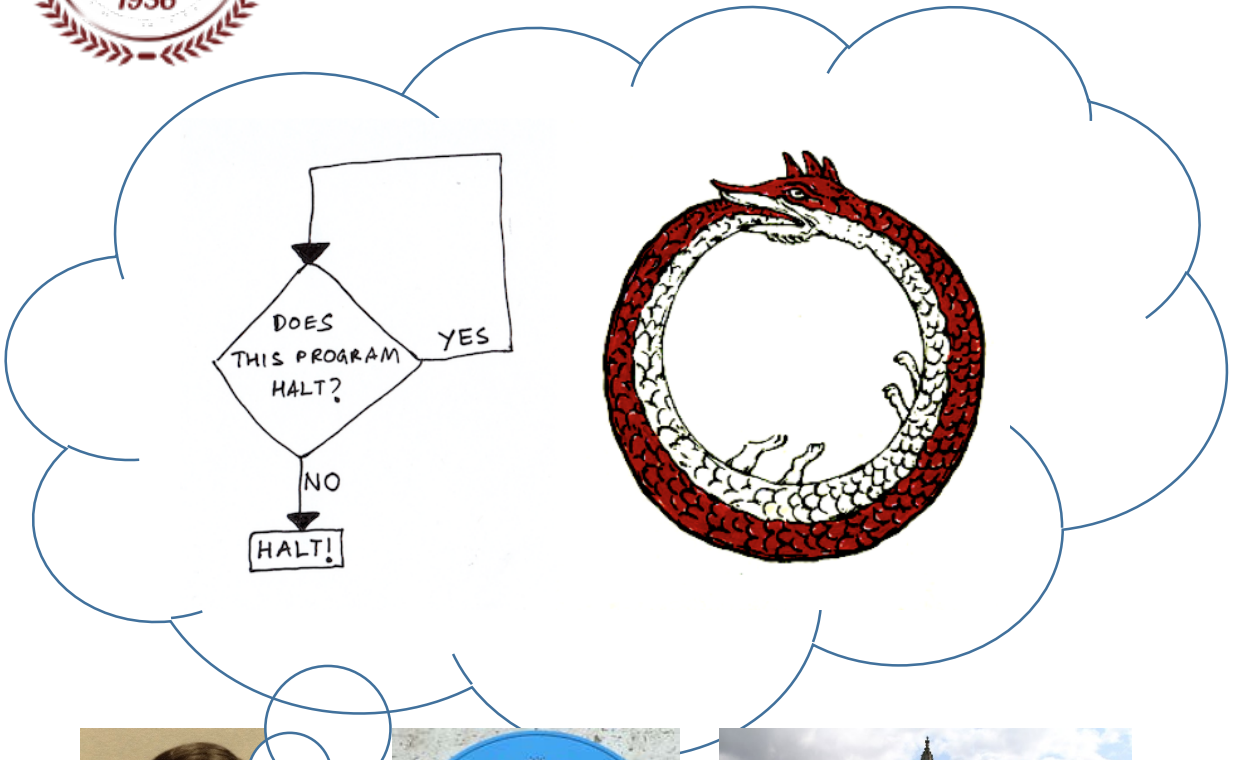
"smart" drug released only in certain cellular conditions

bioreactors

"chemical controller" to optimize yield of metabolically produced biofuels/drugs

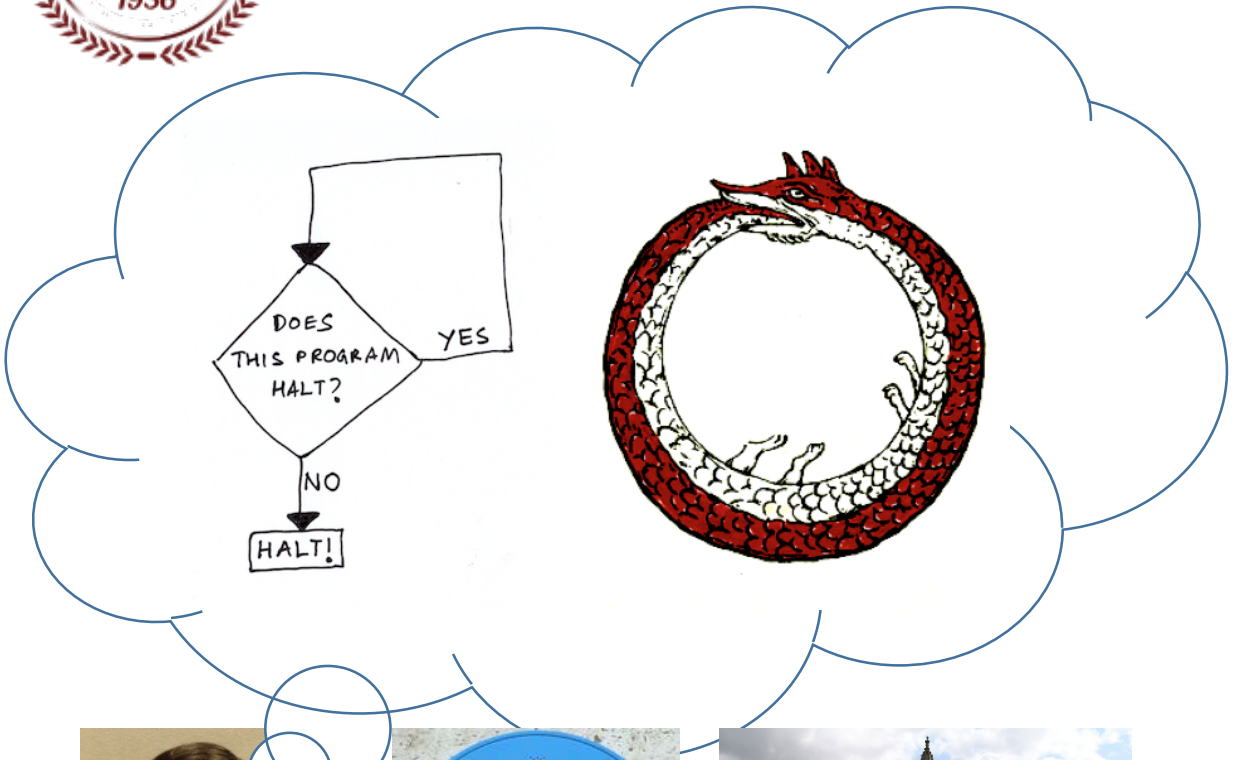
Theoretical Computer Science Approach

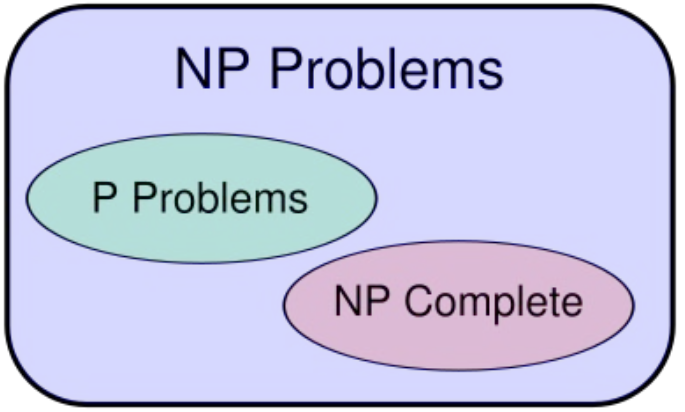
Theoretical Computer Science Approach



What computation is possible and what is not?

Theoretical Computer Science Approach





What computation is possible and what is not?

What computations necessarily take a long time and what can be done quickly? (Computational complexity)

Outline

- (Stochastic, discrete) chemical reaction networks
- Time lower bound on one computational task: *leader election*
- Open questions

Chemical Reaction Networks (CRNs)

- finite set of *d species* {*A*,*B*,*C*,*D*...}
- *configuration* $\boldsymbol{x} \in \mathbb{N}^d$: molecular counts of each species
- finite set of *reactions*: *e.g.*

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

$$C \xrightarrow{k_2} A + A$$

$$C + B \xrightarrow{k_3} C$$

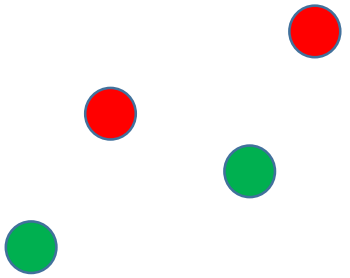
Example Execution

α: $A+B \rightarrow A+C$

β: $C \rightarrow A+A$

$A \quad B \quad C$

$\mathbf{x} = (2, 2, 0)$



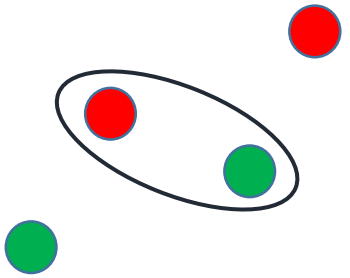
Example Execution

α: $A+B \rightarrow A+C$

β: $C \rightarrow A+A$

$A \quad B \quad C$

$\mathbf{x} = (2, 2, 0)$

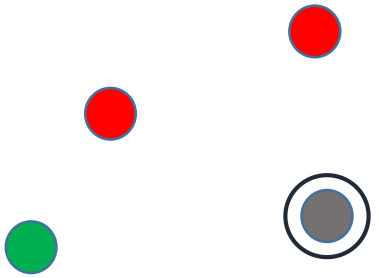


Example Execution

α: $A + B \rightarrow A + \boxed{C}$

β: $C \rightarrow A + A$

	A	*B*	*C*
x =	**2,**	**2,**	**0)**
	α ⇓		
	2,	**1,**	**1)**



Example Execution

α: $A+B \rightarrow A+C$

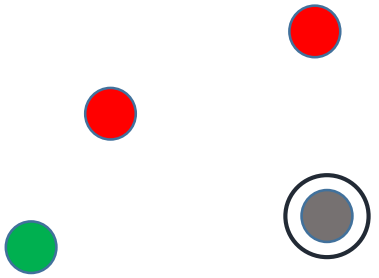
β: $\boxed{C} \rightarrow A+A$

$A \quad B \quad C$

$\mathbf{x = (2,\ 2,\ 0)}$

$\alpha \Downarrow$

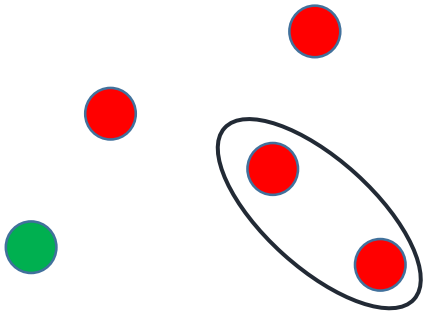
$\mathbf{(2,\ 1,\ 1)}$



Example Execution

α: $A+B \rightarrow A+C$

β: $C \rightarrow \boxed{A+A}$



$A \quad B \quad C$

$\mathbf{x} = (2, 2, 0)$

$\alpha \Downarrow$

$(2, 1, 1)$

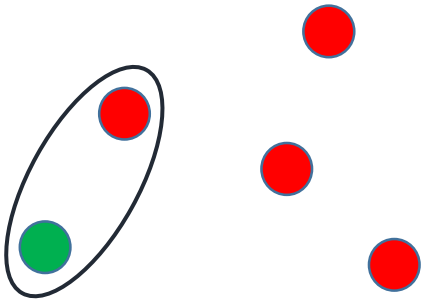
$\beta \Downarrow \qquad \Downarrow \alpha$ ...

$(4, 1, 0)$

Example Execution

α: $A+B \rightarrow A+C$

β: $C \rightarrow A+A$



$A \quad B \quad C$

$x = (2, 2, 0)$

$\alpha \Downarrow$

$(2, 1, 1)$

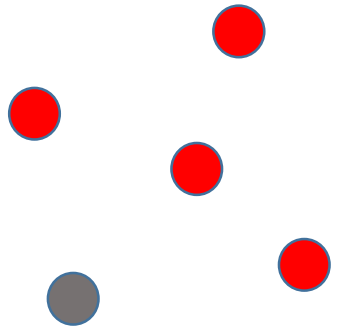
$\beta \Downarrow \qquad \Downarrow \alpha$ ...

$(4, 1, 0)$

Example Execution

α: $A+B \rightarrow A+C$

β: $C \rightarrow A+A$



	A	*B*	*C*
x =	**2,**	**2,**	**0)**
	α ⇓		
	2,	**1,**	**1)**
	β ⇓		⇘α
	4,	**1,**	**0)** ...
	α ⇓		
	4,	**0,**	**1)**
	...		

Stochastic kinetic model of CRNs

Solution volume v

System evolves via a continuous time Markov process:

- time until next reaction is exponentially distributed with rate $\sum prop_i$
- Pr[next reaction is rxn_j] = $prop_j / \sum prop_i$

reaction type	$prop_i$
$A \xrightarrow{k} \ldots$	$k \cdot \#A$
$A+B \xrightarrow{k} \ldots$	$k \cdot \#A \cdot \#B \,/\, v$
$A+A \xrightarrow{k} \ldots$	$k \cdot \#A \cdot (\#A-1) \,/\, v$

McQuarrie 1967, van Kampen, Gillespie 1977, etc

Main result (informally)

"Leader election" (getting to count 1 of a species) requires $\Omega(n)$ expected time

$$L+L \rightarrow L+F$$

Doty, Soloveichik, "Stable leader election in population protocols requires linear time"
DISC 2015: *International Symposium on Distributed Computing*

Leader election: what's the point?

Leader election: what's the point?

Getting "precise" quantities of species (*e.g.* exactly 1 *L*) from "uncontrolled" initial conditions (*e.g.* a lot of *A*).

- easy to add billions of molecules to test tube
- difficult to add 1

Leader election: what's the point?

Getting "precise" quantities of species (*e.g.* exactly 1 *L*) from "uncontrolled" initial conditions (*e.g.* a lot of *A*).

- easy to add billions of molecules to test tube
- difficult to add 1

Other kinds of computation *require* an initial leader.

Leader election: what's the point?

Getting "precise" quantities of species (*e.g.* exactly 1 *L*) from "uncontrolled" initial conditions (*e.g.* a lot of *A*).

- easy to add billions of molecules to test tube
- difficult to add 1

Other kinds of computation *require* an initial leader.

Versions of "leader election" in biology: centriole number, choice of olfactory receptor expression, ...

Leader election: what's the point?

First step towards establishing full theory of time complexity in CRN computation ("in-house" result)

Getting "precise" quantities of species (*e.g.* exactly 1 *L*) from "uncontrolled" initial conditions (*e.g.* a lot of *A*).

- easy to add billions of molecules to test tube
- difficult to add 1

Other kinds of computation *require* an initial leader.

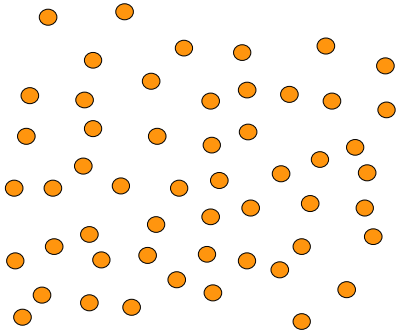
Versions of "leader election" in biology: centriole number, choice of olfactory receptor expression, ...

Defining **stable** leader election

Defining **stable** leader election

i

uniform initial configuration



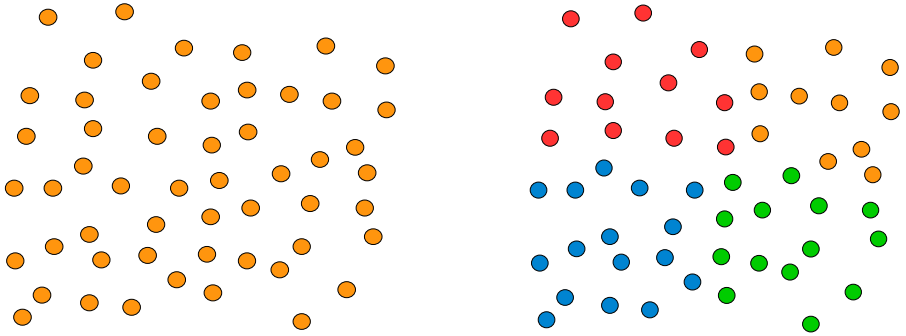
Defining **stable** leader election

$\forall$

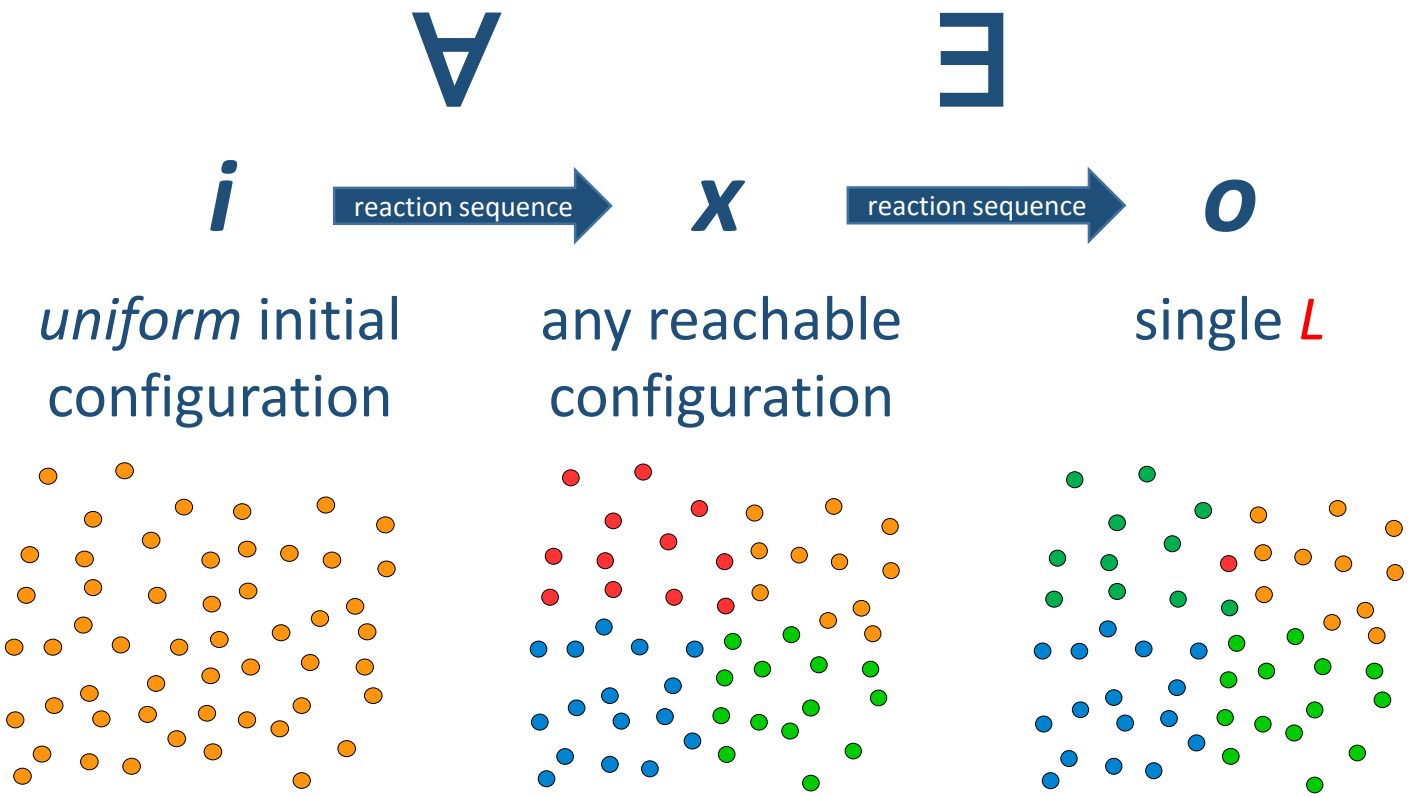
i → reaction sequence → x

uniform initial configuration

any reachable configuration



Defining **stable** leader election



Defining **stable** leader election

$\forall$ $\exists$ $\forall$

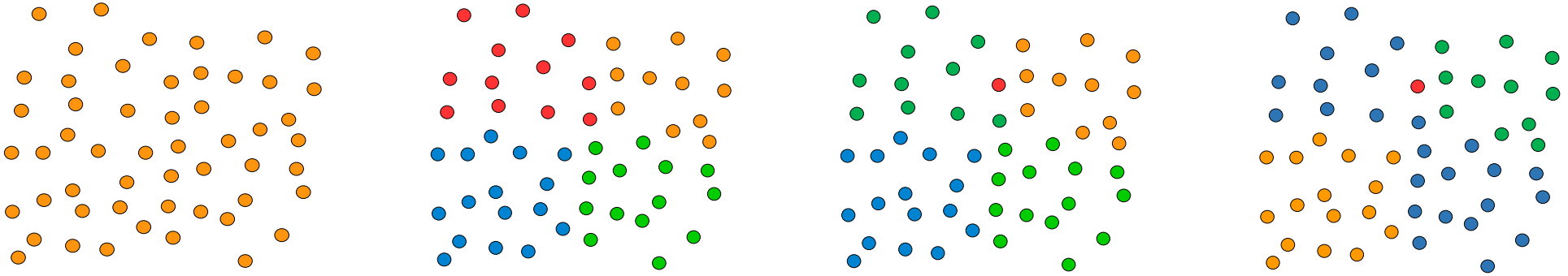
i → reaction sequence → ***x*** → reaction sequence → ***o*** → reaction sequence → ***o'***

uniform initial configuration

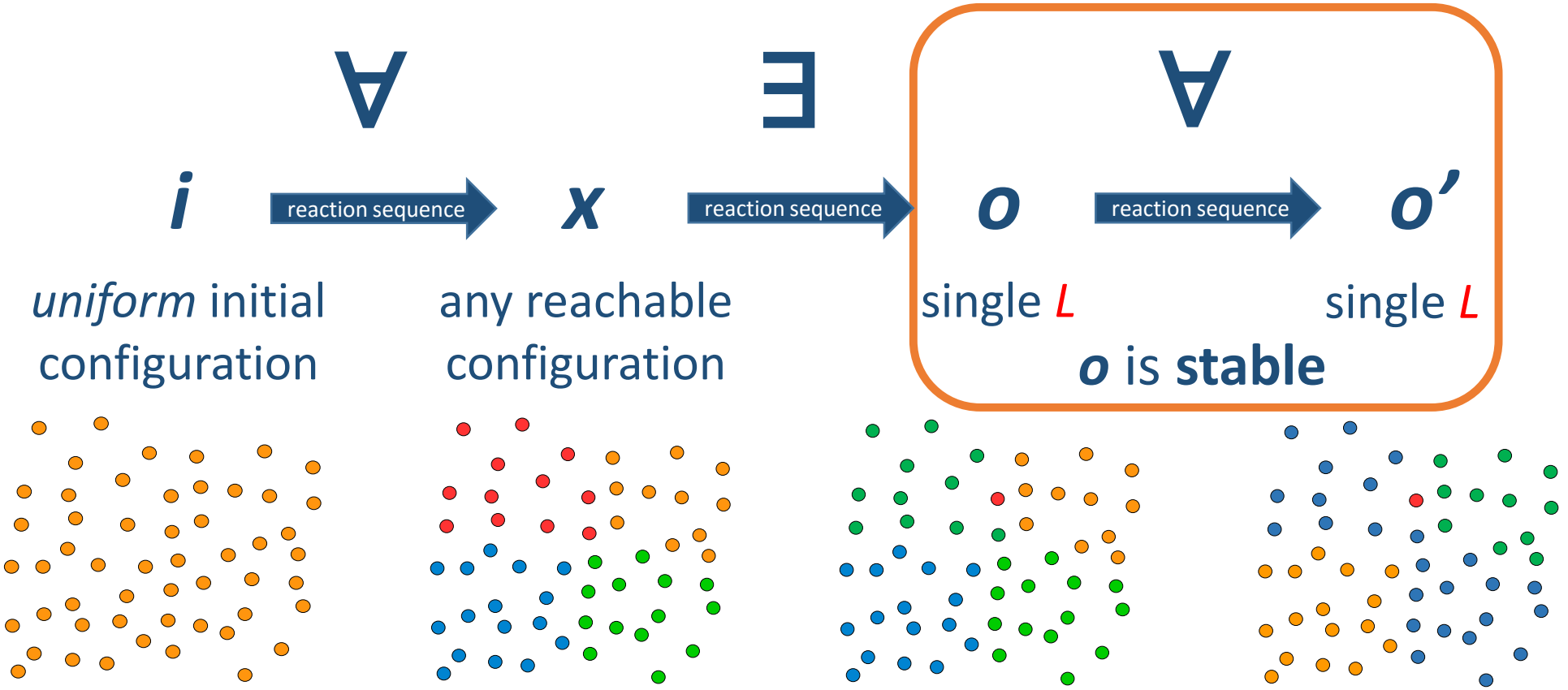
any reachable configuration

single *L*

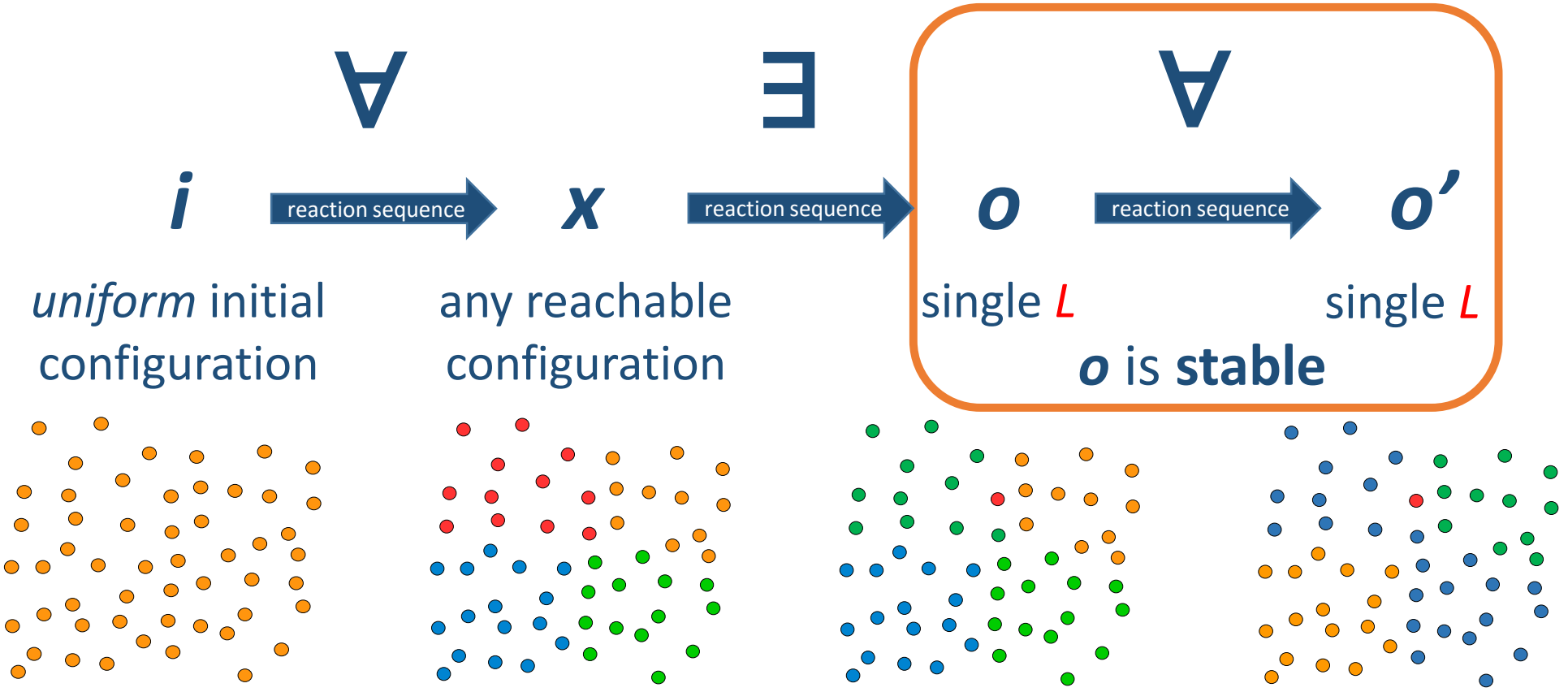
single *L*



Defining **stable** leader election



Defining **stable** leader election



If number of reachable configurations is finite, CRN **will** reach such a configuration with probability 1

Speed of Stable Computation

Speed of Stable Computation

How to fairly assess speed?

Like any respectable computer scientist…
by ignoring constant factors

Speed of Stable Computation

How to fairly assess speed?

Like any respectable computer scientist…
by ignoring constant factors

rate constants k_i bounded by 1

Speed of Stable Computation

How to fairly assess speed?

Like any respectable computer scientist…
by ignoring constant factors

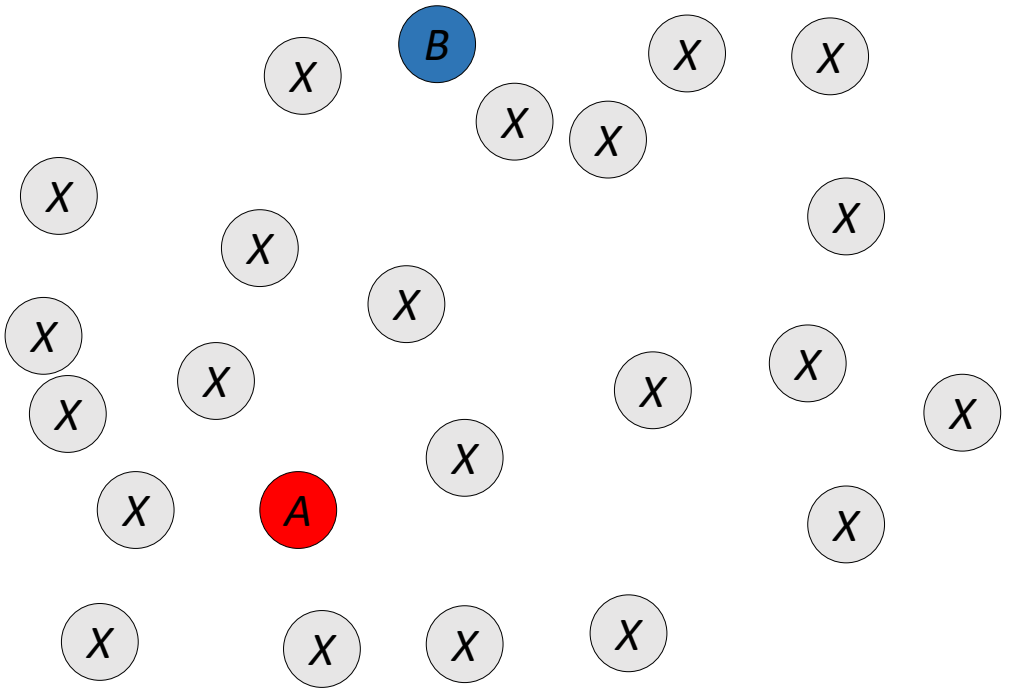
rate constants k_i bounded by 1

n = total molecular count
volume $v = \Theta(n)$
i.e., require bounded concentration
(finite density constraint)

An Exponential Time Difference

n molecules

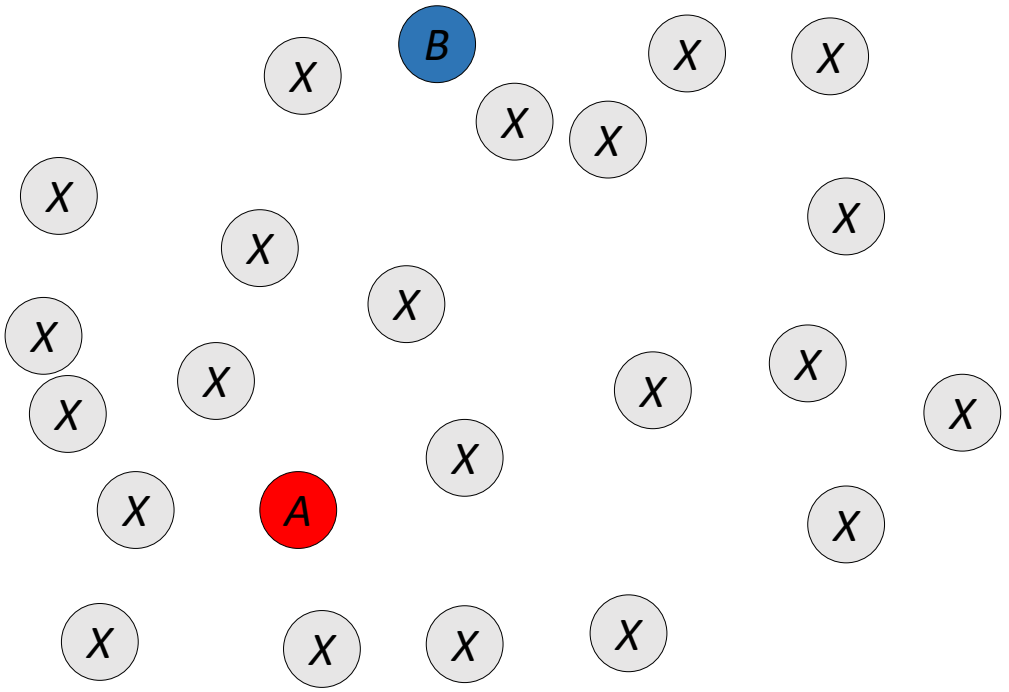
volume $v=\Theta(n)$



An Exponential Time Difference

n molecules

volume $v=\Theta(n)$



$$A+B \longrightarrow Y+B$$

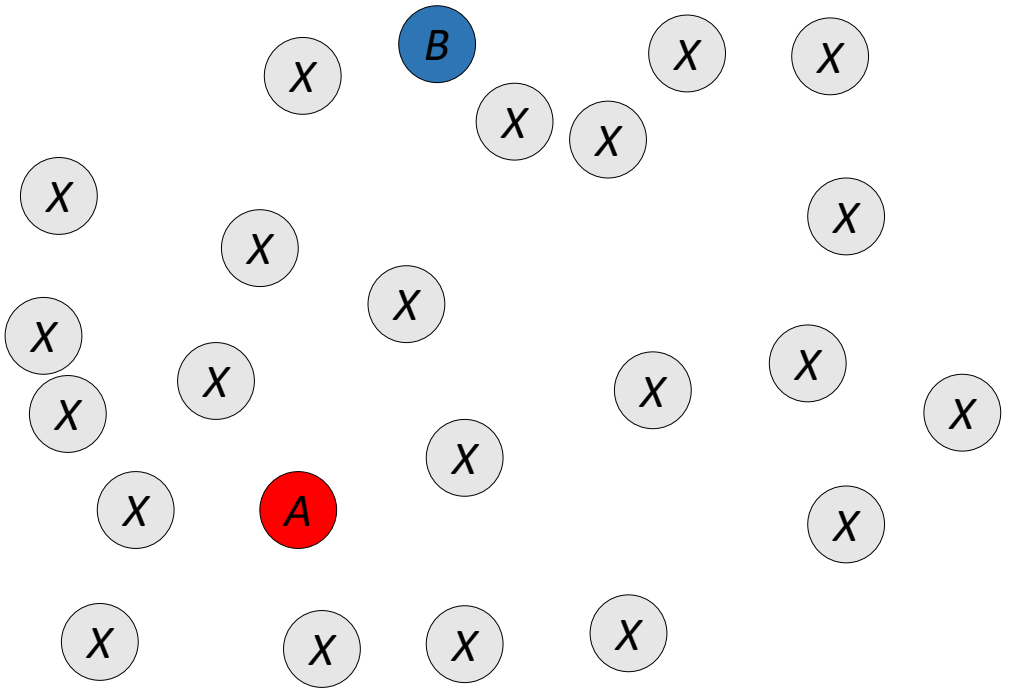
propensity: $\#A \cdot \#B \,/\, v = \Theta(1/n)$

expected time to produce *Y*: $\Theta(n)$

An Exponential Time Difference

n molecules

volume $v=\Theta(n)$



$$A+B \longrightarrow Y+B$$

propensity: $\#A \cdot \#B \,/\, v = \Theta(1/n)$

$$B+X \longrightarrow B+B$$
$$A+B \longrightarrow Y+B$$

expected time to produce *Y*:

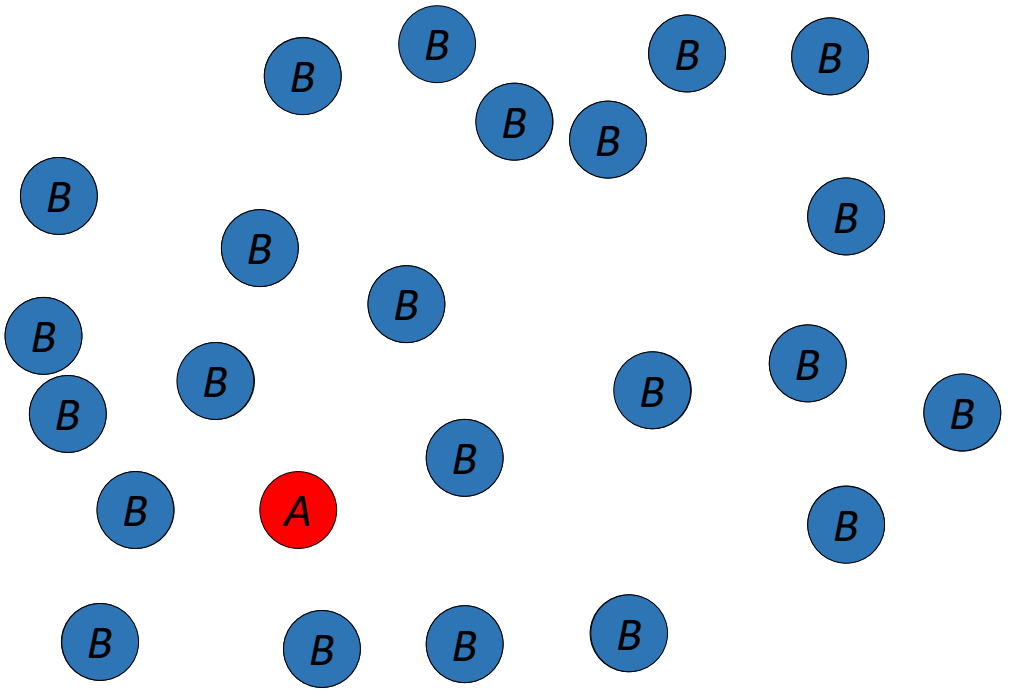
$\Theta(n)$

$\Theta(\log n)$

An Exponential Time Difference

n molecules

volume $v=\Theta(n)$



$$A+B\longrightarrow Y+B$$

propensity: $\#A\cdot\#B\ /\ v = \Theta(1/n)$

$$B+X\longrightarrow B+B$$
$$A+B\longrightarrow Y+B$$

expected time to produce *Y*:

$\Theta(n)$

$\Theta(\log n)$

Speed of naïve stable leader election

initial configuration: uniform (all n molecules of the same species)
output: stable configuration with $\#L = 1$

Speed of naïve stable leader election

initial configuration: uniform (all n molecules of the same species)
output: stable configuration with $\#L = 1$

Naïve algorithm:

initial config = $\{n\ L\}$

$L+L \rightarrow L+F$

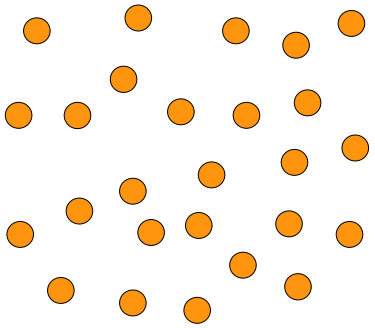
Speed of naïve stable leader election

initial configuration: uniform (all n molecules of the same species)
output: stable configuration with $\#L = 1$

Naïve algorithm:

initial config = $\{n\ L\}$

$L+L \rightarrow L+F$

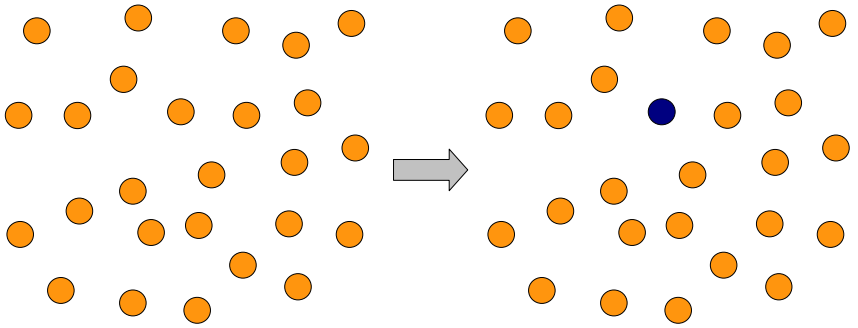


Speed of naïve stable leader election

initial configuration: uniform (all n molecules of the same species)
output: stable configuration with $\#L = 1$

Naïve algorithm:

initial config = $\{n\ L\}$ $\qquad L+L \rightarrow L+F$

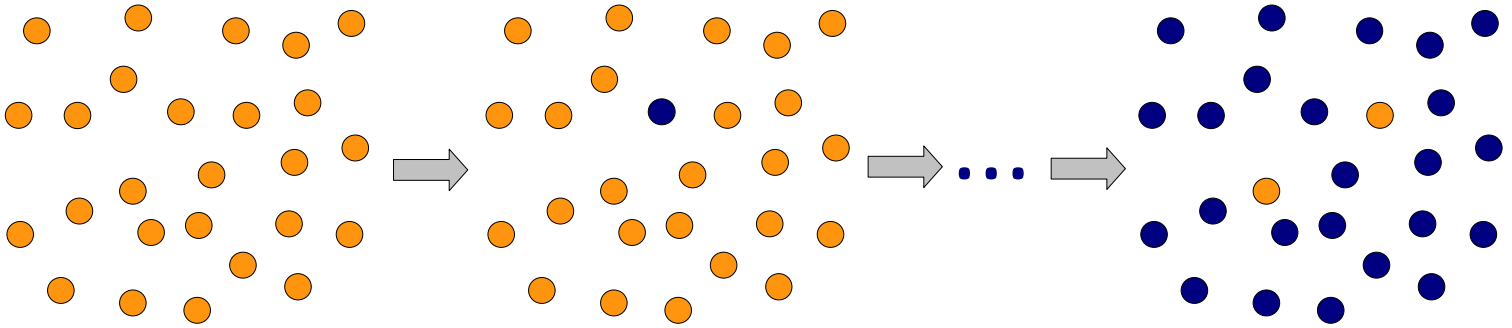


Speed of naïve stable leader election

initial configuration: uniform (all n molecules of the same species)
output: stable configuration with $\#L = 1$

Naïve algorithm:

initial config = $\{n\ L\}$ $\qquad L+L \rightarrow L+F$

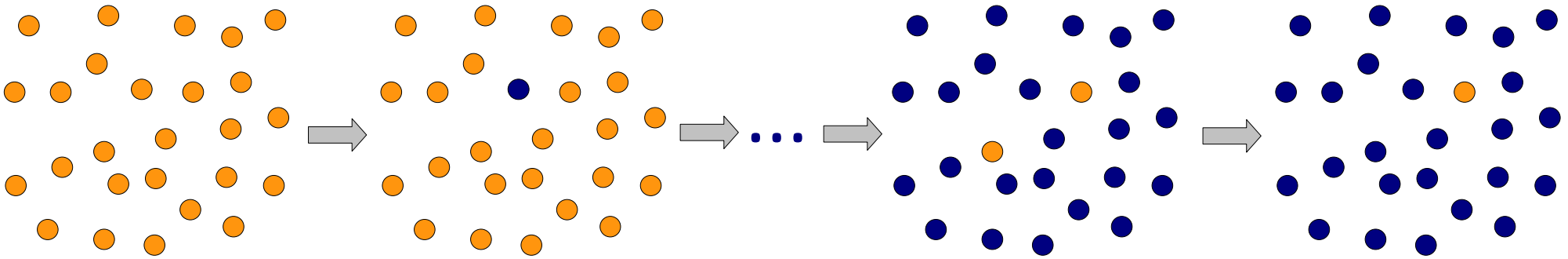


Speed of naïve stable leader election

initial configuration: uniform (all n molecules of the same species)
output: stable configuration with $\#L = 1$

Naïve algorithm:

initial config = $\{n\ L\}$ $\qquad L+L \rightarrow L+F$



Speed of naïve stable leader election

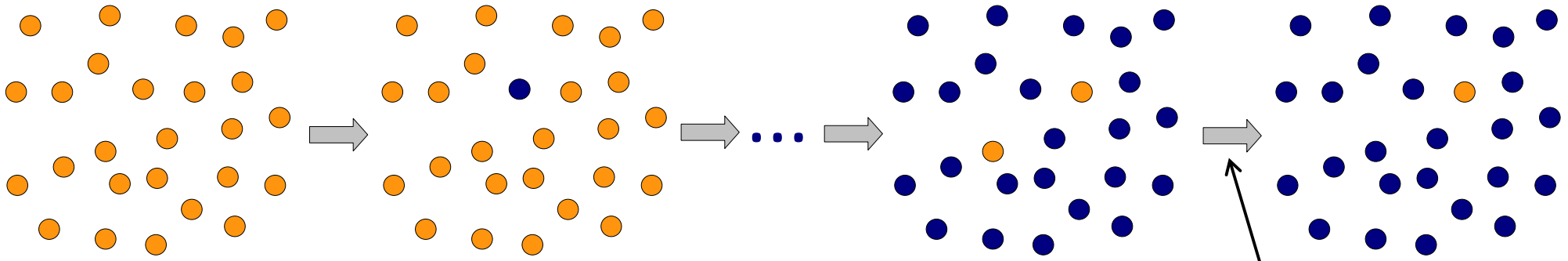
initial configuration: uniform (all n molecules of the same species)
output: stable configuration with $\#L = 1$

Naïve algorithm:

initial config = $\{n\ L\}$

$L+L \rightarrow L+F$

$\Theta(n)$ time



"2-bottleneck" reaction

Speed of naïve stable leader election

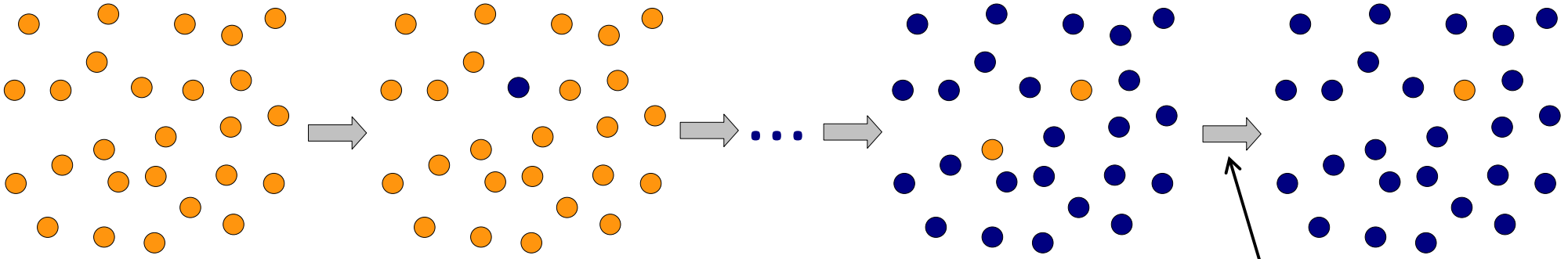
initial configuration: uniform (all n molecules of the same species)
output: stable configuration with $\#L = 1$

Naïve algorithm:

initial config = $\{n\ L\}$

$L+L \rightarrow L+F$

$\Theta(n)$ time



"2-bottleneck" reaction

Fast leader elimination

$L+F \rightarrow F+F$

Speed of naïve stable leader election

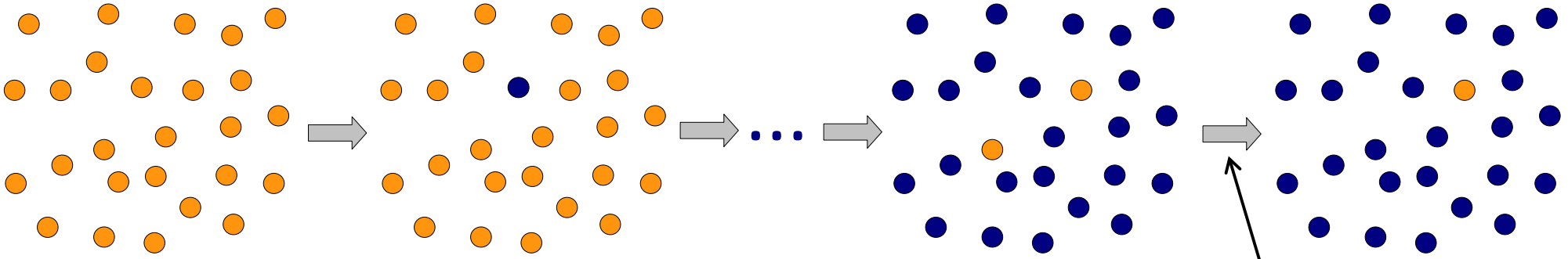
initial configuration: uniform (all n molecules of the same species)
output: stable configuration with $\#L = 1$

Naïve algorithm:

initial config = $\{n\ L\}$

$L+L \rightarrow L+F$

$\Theta(n)$ time



"2-bottleneck" reaction

Fast leader elimination

$L+F \rightarrow F+F$

$\Theta(\log n)$ time

one of these is always count $\geq n/2$

Main theorem

Any “reasonable” CRN stably electing a leader requires $\Omega(n)$ time to reach a stable configuration. ($L+L \rightarrow L+F$ is **optimal**)

Main theorem

Any "reasonable" CRN stably electing a leader requires $\Omega(n)$ time to reach a stable configuration. ($L+L \rightarrow L+F$ is **optimal**)

> "*reasonable*" ≈ from initial configuration with n molecules, at most $O(n)$ additional molecules produced in initial $O(1)$ time… implies probably all species eventually produced

Main theorem

Any "reasonable" CRN stably electing a leader requires $\Omega(n)$ time to reach a stable configuration. ($L+L \rightarrow L+F$ is **optimal**)

> "*reasonable*" ≈ from initial configuration with n molecules, at most $O(n)$ additional molecules produced in initial $O(1)$ time… implies probably all species eventually produced

Holds for *dense* initial configurations: all species present have count $\Omega(n)$

Main theorem

Any "reasonable" CRN stably electing a leader requires $\Omega(n)$ time to reach a stable configuration. ($L+L \rightarrow L+F$ is **optimal**)

"*reasonable*" ≈ from initial configuration with n molecules, at most $O(n)$ additional molecules produced in initial $O(1)$ time… implies probably all species eventually produced

Holds for *dense* initial configurations: all species present have count $\Omega(n)$

notion of denseness is compatible with *"mass action limit"*
a.k.a. "mean field approximation"

Entire proof?

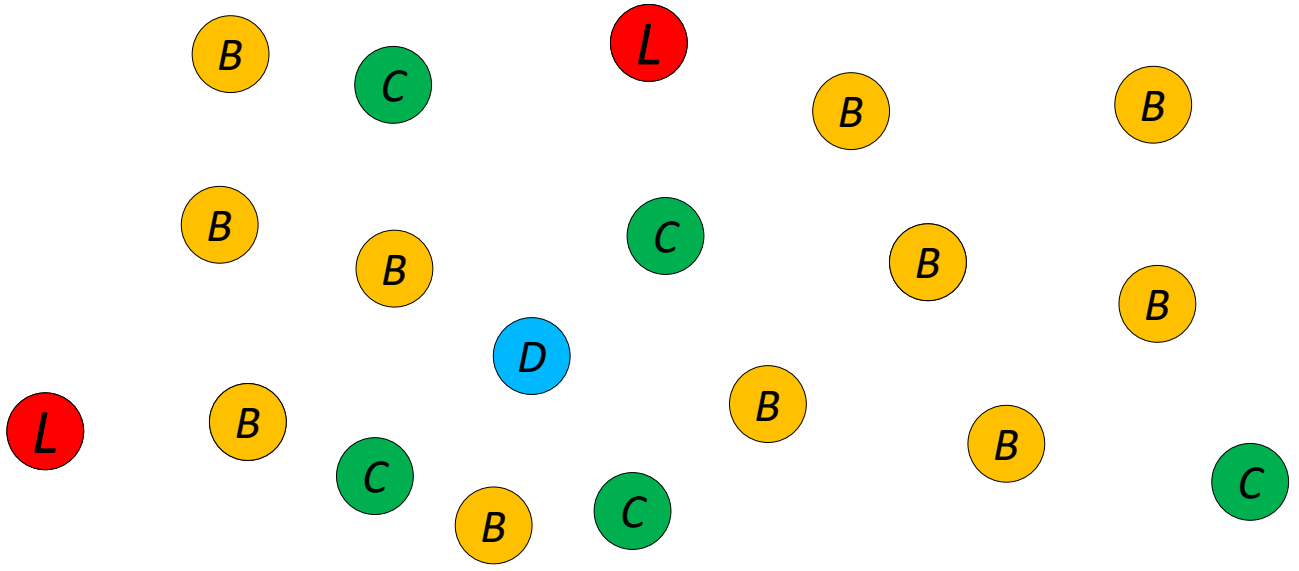
Entire proof?

Sublinear time implies ∃ reaction sequence resulting in 1 *L* with no *O*(1)-bottleneck reactions

Entire proof?

Sublinear time implies ∃ reaction sequence resulting in 1 *L* with no *O*(1)-bottleneck reactions

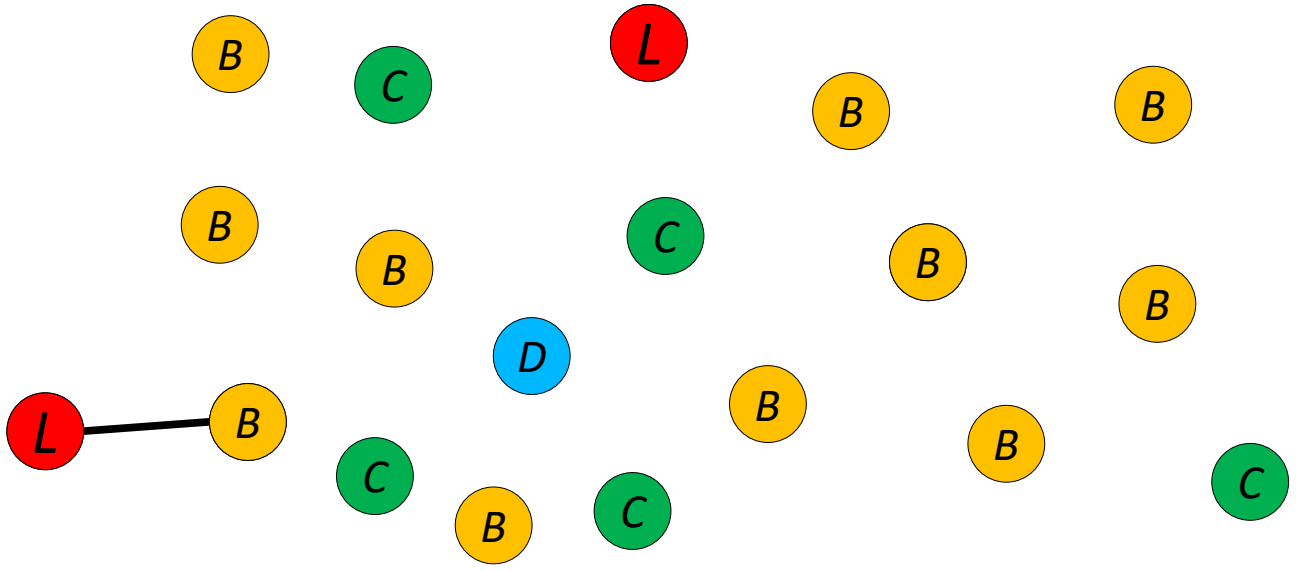
when *L*=2 and *L*+*B*→*C*+*D* occurs, *B*≫1 since it's not a bottleneck



Entire proof?

Sublinear time implies ∃ reaction sequence resulting in 1 *L* with no *O*(1)-bottleneck reactions

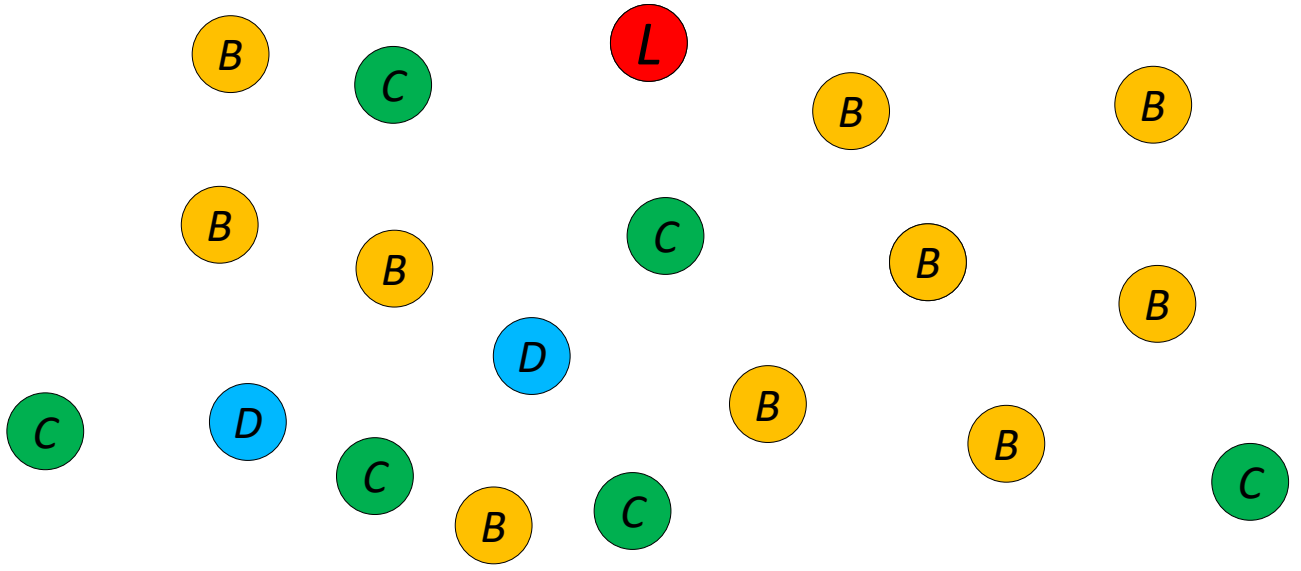
when *L*=2 and *L*+*B*→*C*+*D* occurs, *B*≫1 since it's not a bottleneck



Entire proof?

Sublinear time implies ∃ reaction sequence resulting in 1 *L* with no *O*(1)-bottleneck reactions

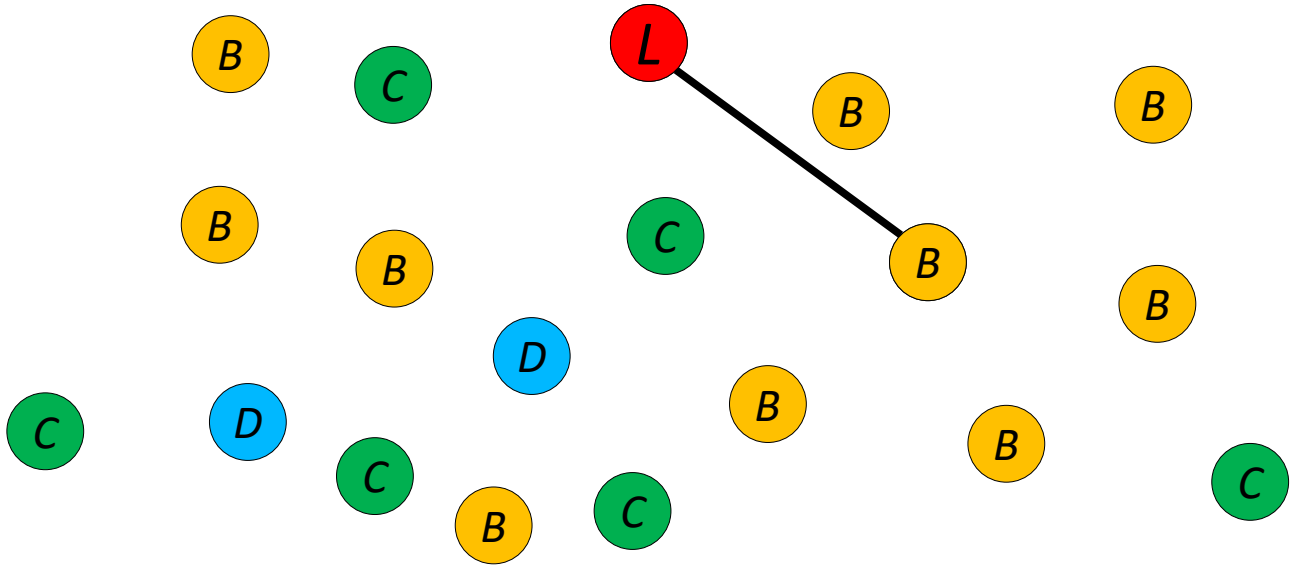
when *L*=2 and *L*+*B*→*C*+*D* occurs, *B*≫1 since it's not a bottleneck



Entire proof?

Sublinear time implies ∃ reaction sequence resulting in 1 *L* with no *O*(1)-bottleneck reactions

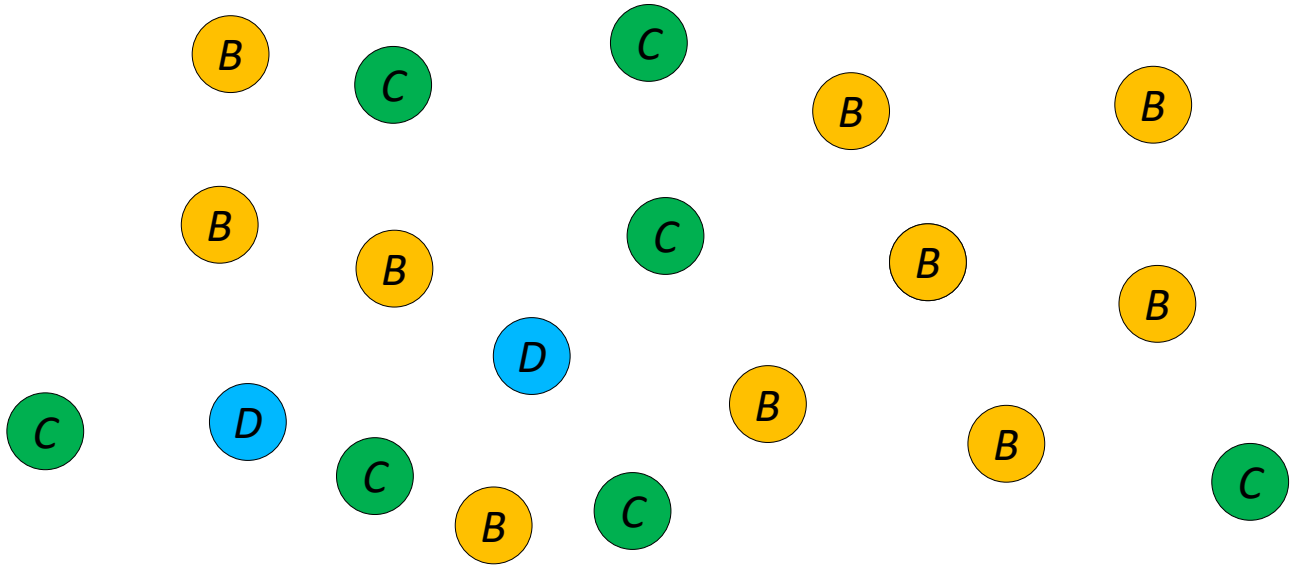
when *L*=2 and *L*+*B*→*C*+*D* occurs, *B*≫1 since it's not a bottleneck



Entire proof?

Sublinear time implies ∃ reaction sequence resulting in 1 *L* with no *O*(1)-bottleneck reactions

when *L*=2 and *L*+*B*→*C*+*D* occurs, *B*≫1 since it's not a bottleneck

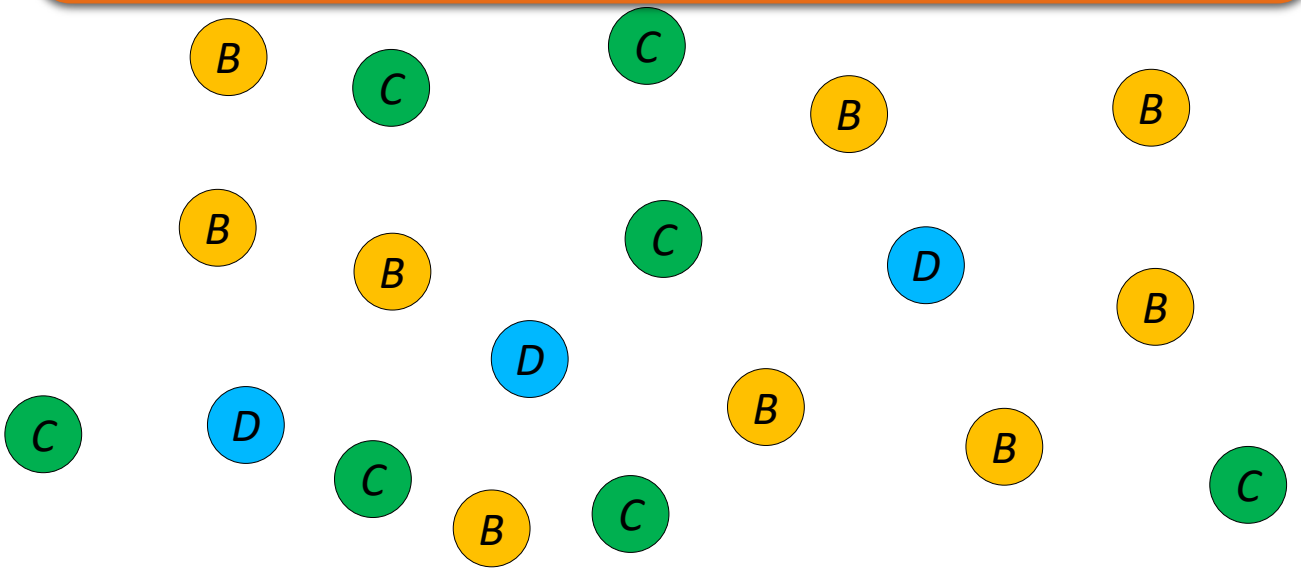


~~Entire proof?~~

Sublinear time implies ∃ reaction sequence resulting in 1 *L* with no *O*(1)-bottleneck reactions

when *L*=2 and *L*+*B*→*C*+*D* occurs, *B*≫1 since it's not a bottleneck

This configuration might not be stable!
(CRN could produce a new leader)



Proof must reason about initial configuration

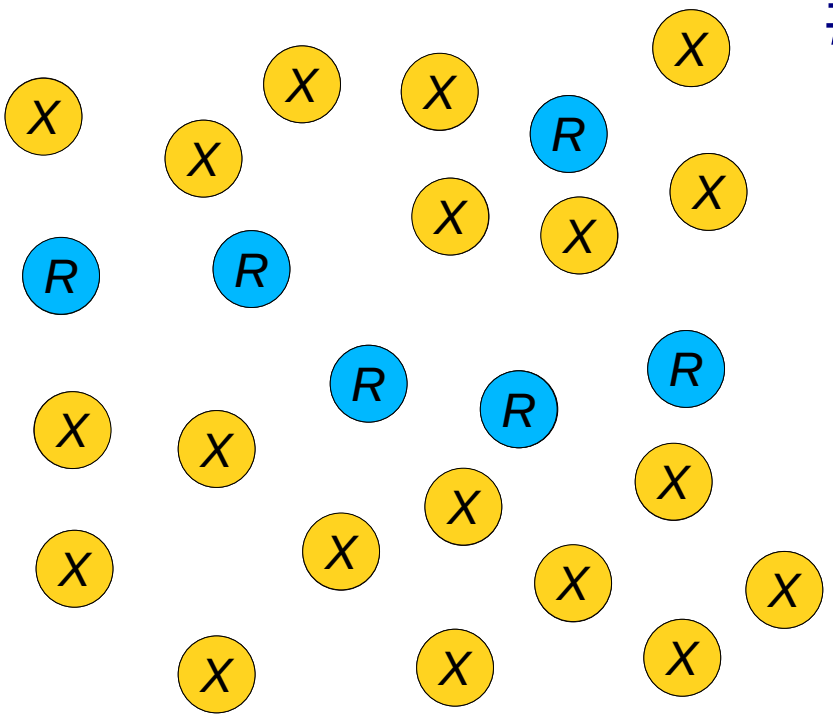
Sublinear time leader election from *non-uniform* initial configurations **is** possible!

$$\#R = n^{1/4}$$
$$\#X = n - n^{1/4}$$
$$\#L = 0$$

Proof must reason about initial configuration

Sublinear time leader election from *non-uniform* initial configurations **is** possible!

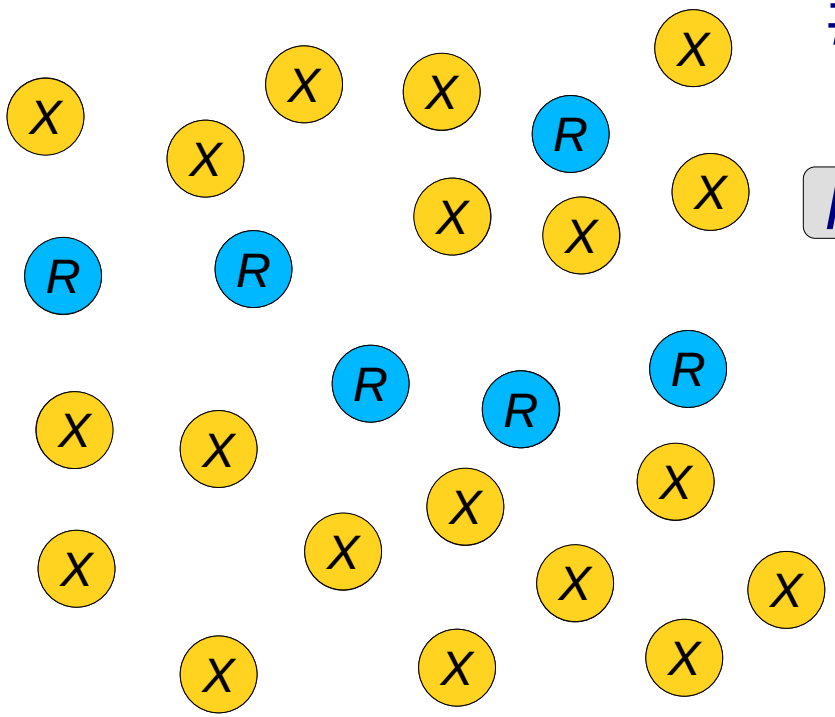
$$\#R = n^{1/4}$$
$$\#X = n - n^{1/4}$$
$$\#L = 0$$



Proof must reason about initial configuration

Sublinear time leader election from *non-uniform* initial configurations **is** possible!

$$\#R = n^{1/4}$$
$$\#X = n - n^{1/4}$$
$$\#L = 0$$



$R+R \rightarrow L+K$ // produce L $O(n^{1/2})$ time

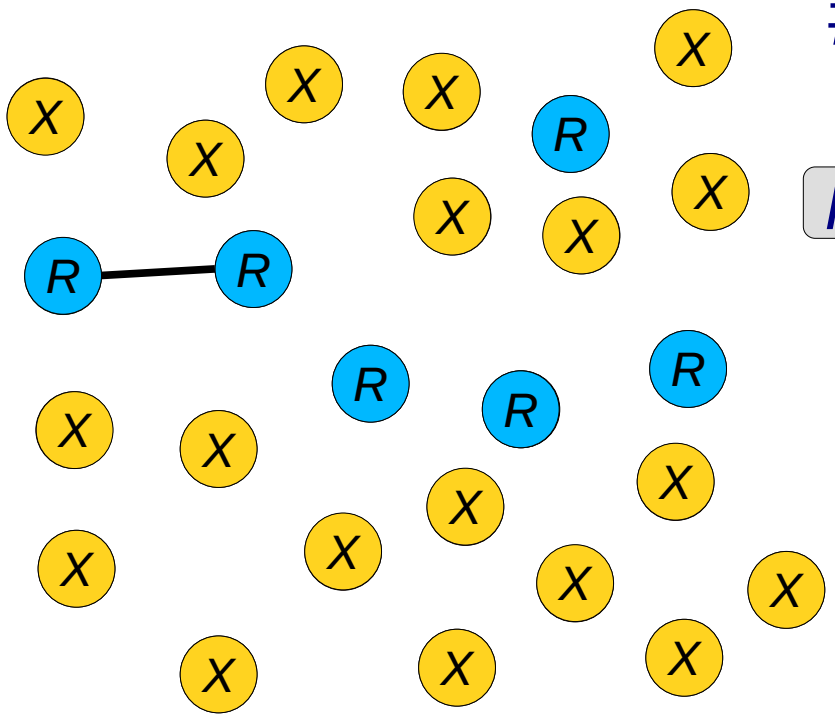
Proof must reason about initial configuration

Sublinear time leader election from *non-uniform* initial configurations **is** possible!

$\#R = n^{1/4}$

$\#X = n - n^{1/4}$

$\#L = 0$



$R+R \rightarrow L+K$ // produce L $O(n^{1/2})$ time

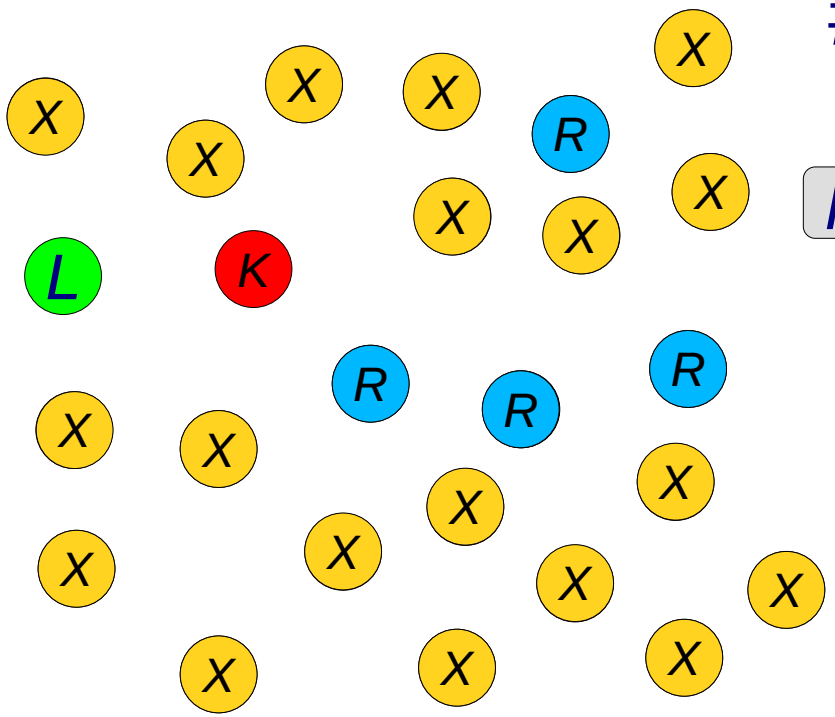
Proof must reason about initial configuration

Sublinear time leader election from *non-uniform* initial configurations **is** possible!

$\#R = n^{1/4}$

$\#X = n - n^{1/4}$

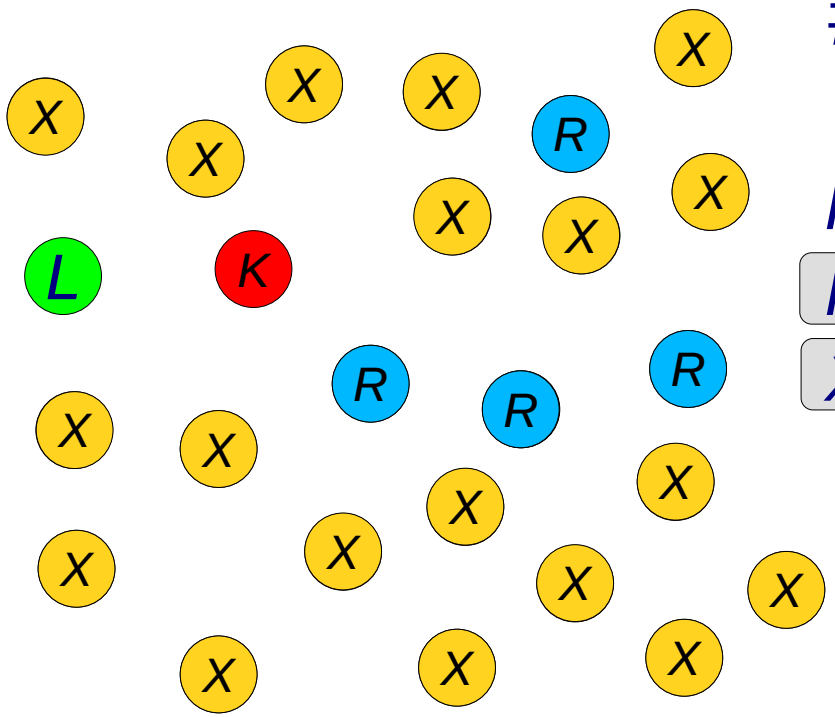
$\#L = 0$



$R+R \rightarrow L+K$ // produce L $O(n^{1/2})$ time

Proof must reason about initial configuration

Sublinear time leader election from *non-uniform* initial configurations **is** possible!



$\#R = n^{1/4}$

$\#X = n - n^{1/4}$

$\#L = 0$

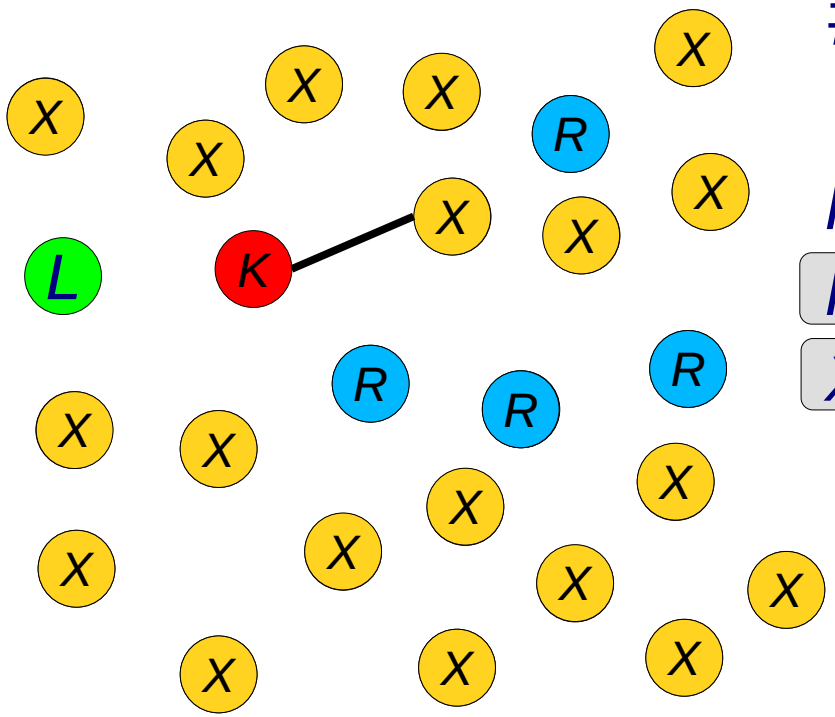
$R+R \rightarrow L+K$ // produce L $O(n^{1/2})$ time

$R+K \rightarrow K+K$ // stop 2nd L from being produced

$X+K \rightarrow K+K$ // $O(\log n)$ time to complete

Proof must reason about initial configuration

Sublinear time leader election from *non-uniform* initial configurations **is** possible!



$\#R = n^{1/4}$

$\#X = n - n^{1/4}$

$\#L = 0$

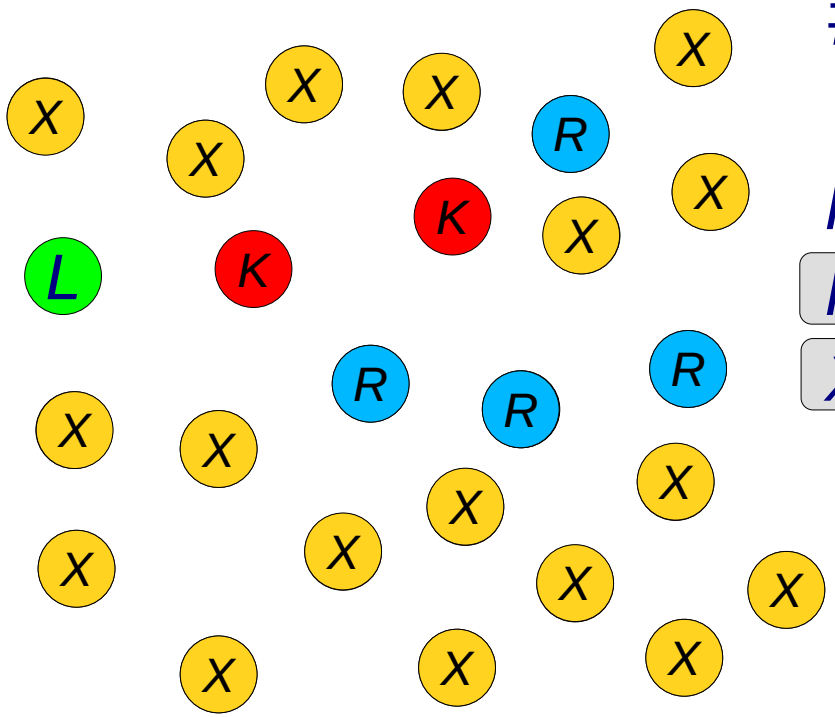
$R+R \rightarrow L+K$ // produce L $O(n^{1/2})$ time

$R+K \rightarrow K+K$ // stop 2nd L from being produced

$X+K \rightarrow K+K$ // $O(\log n)$ time to complete

Proof must reason about initial configuration

Sublinear time leader election from *non-uniform* initial configurations **is** possible!



$\#R = n^{1/4}$

$\#X = n - n^{1/4}$

$\#L = 0$

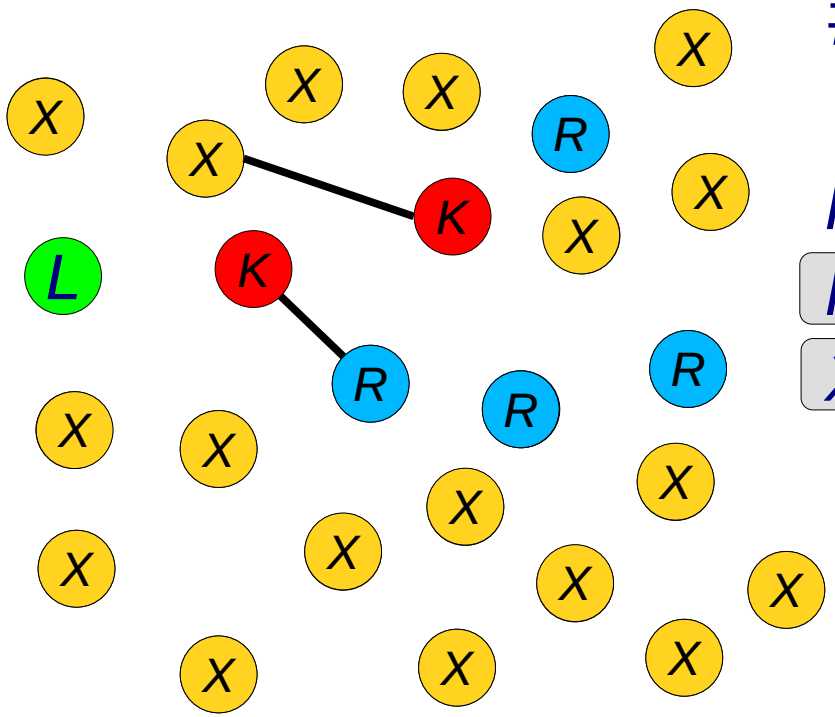
$R+R \rightarrow L+K$ // produce L $O(n^{1/2})$ time

$R+K \rightarrow K+K$ // stop 2nd L from being produced

$X+K \rightarrow K+K$ // $O(\log n)$ time to complete

Proof must reason about initial configuration

Sublinear time leader election from *non-uniform* initial configurations **is** possible!



$\#R = n^{1/4}$

$\#X = n - n^{1/4}$

$\#L = 0$

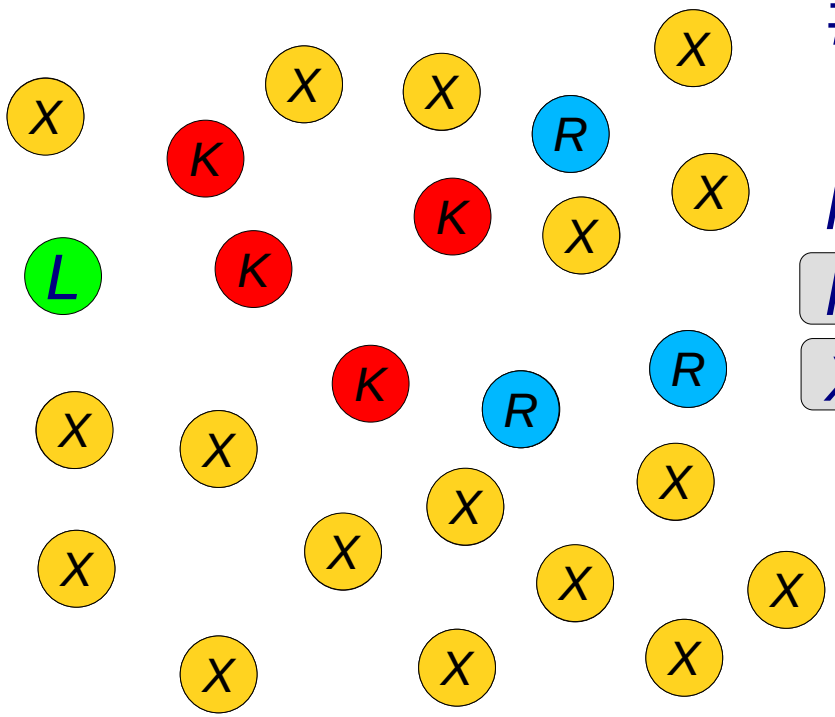
$R+R \rightarrow L+K$ // produce L $O(n^{1/2})$ time

$R+K \rightarrow K+K$ // stop 2nd L from being produced

$X+K \rightarrow K+K$ // $O(\log n)$ time to complete

Proof must reason about initial configuration

Sublinear time leader election from *non-uniform* initial configurations **is** possible!



$\#R = n^{1/4}$

$\#X = n - n^{1/4}$

$\#L = 0$

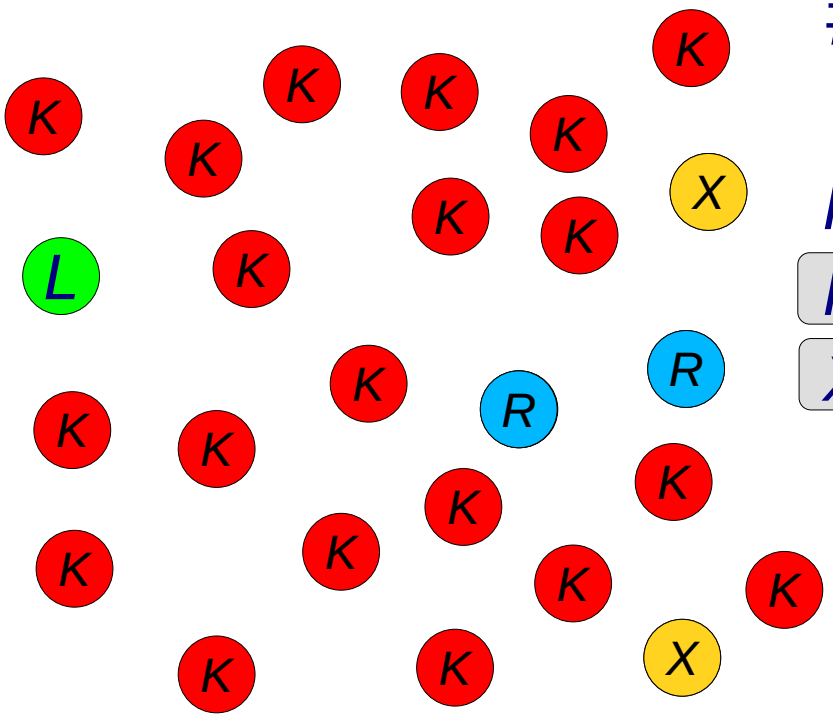
$R+R \rightarrow L+K$ // produce L $O(n^{1/2})$ time

$R+K \rightarrow K+K$ // stop 2nd L from being produced

$X+K \rightarrow K+K$ // $O(\log n)$ time to complete

Proof must reason about initial configuration

Sublinear time leader election from *non-uniform* initial configurations **is** possible!



$\#R = n^{1/4}$

$\#X = n - n^{1/4}$

$\#L = 0$

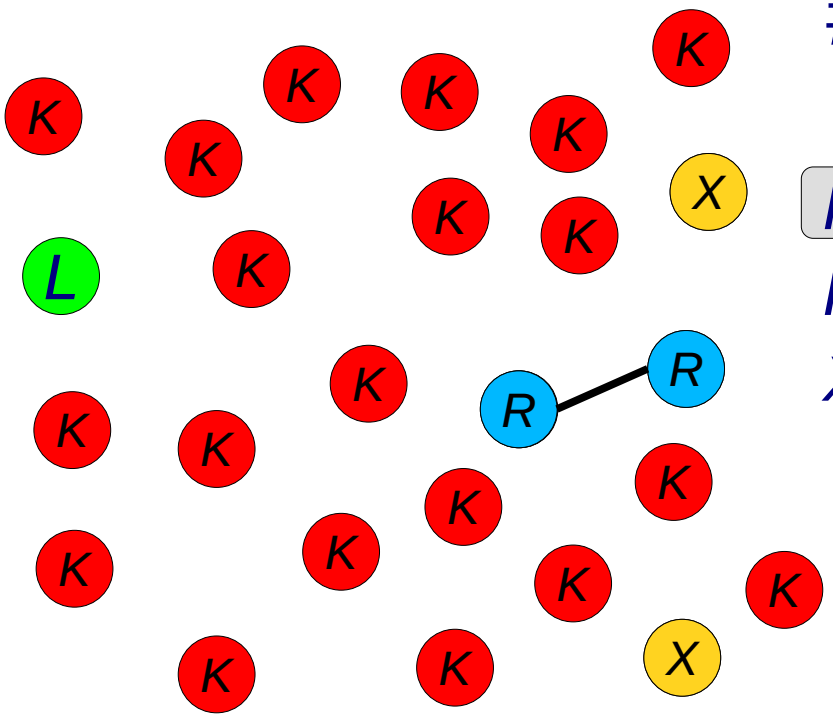
$R+R \rightarrow L+K$ // produce L $O(n^{1/2})$ time

$R+K \rightarrow K+K$ // stop 2nd L from being produced

$X+K \rightarrow K+K$ // $O(\log n)$ time to complete

Proof must reason about initial configuration

Sublinear time leader election from *non-uniform* initial configurations **is** possible!



$\#R = n^{1/4}$

$\#X = n - n^{1/4}$

$\#L = 0$

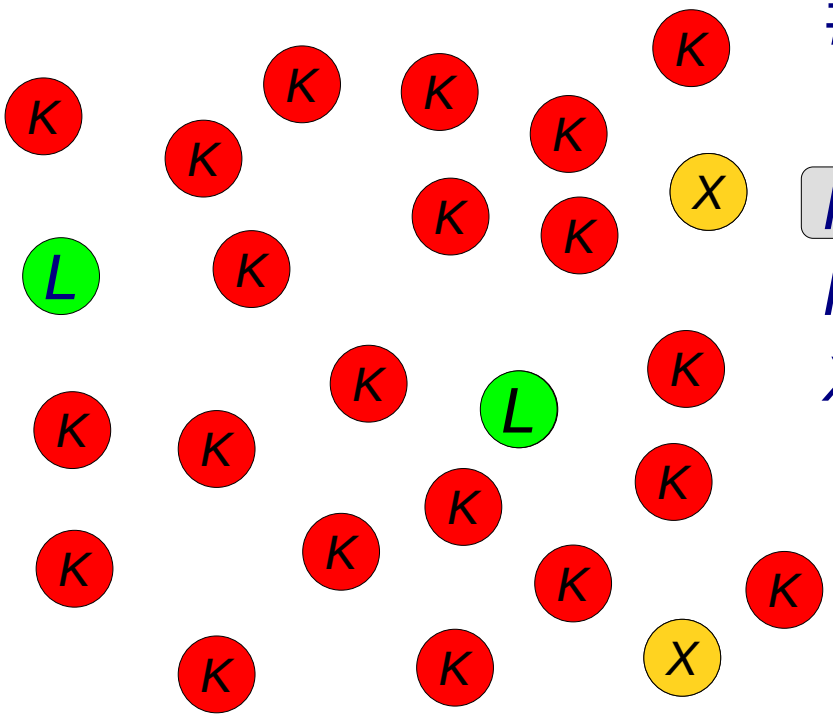
$R+R \rightarrow L+K$ // produce L $O(n^{1/2})$ time

$R+K \rightarrow K+K$ // stop 2nd L from being produced

$X+K \rightarrow K+K$ // $O(\log n)$ time to complete

Proof must reason about initial configuration

Sublinear time leader election from *non-uniform* initial configurations **is** possible!



$\#R = n^{1/4}$

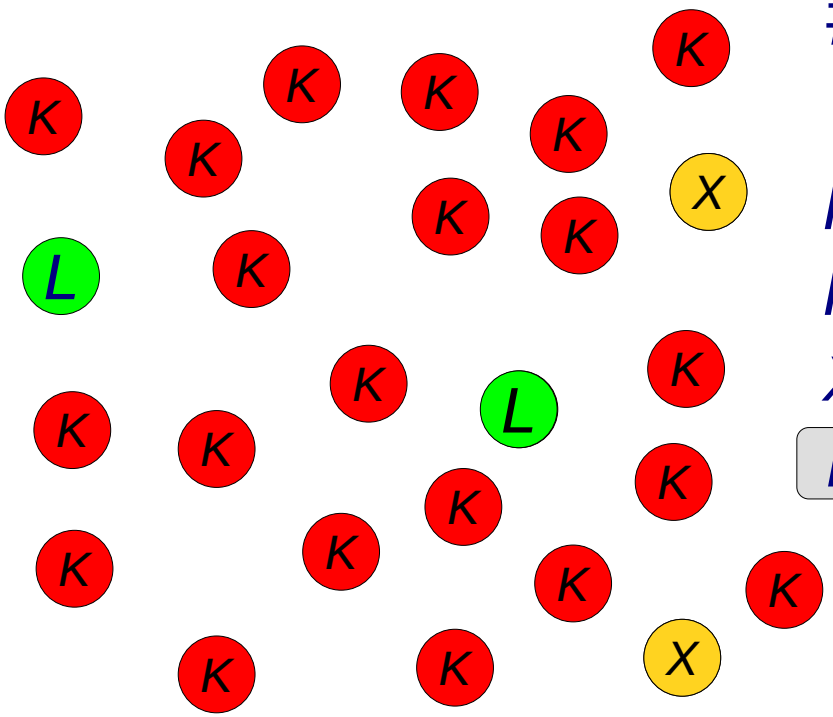
$\#X = n - n^{1/4}$

$\#L = 0$

$R+R \rightarrow L+K$	// produce L $O(n^{1/2})$ time
$R+K \rightarrow K+K$	// stop 2nd L from being produced
$X+K \rightarrow K+K$	// $O(\log n)$ time to complete

Proof must reason about initial configuration

Sublinear time leader election from *non-uniform* initial configurations **is** possible!



$\#R = n^{1/4}$

$\#X = n - n^{1/4}$

$\#L = 0$

$R+R \rightarrow L+K$ // produce L $O(n^{1/2})$ time

$R+K \rightarrow K+K$ // stop 2nd L from being produced

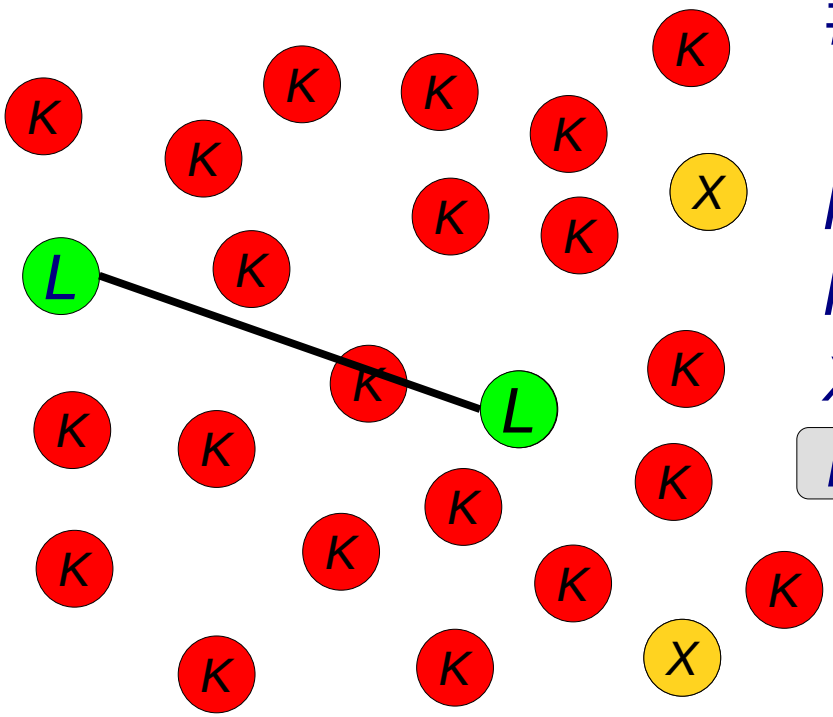
$X+K \rightarrow K+K$ // $O(\log n)$ time to complete

$L+L \rightarrow L+K$ // just in case 2nd L produced

// $O(n)$ time if needed (likely not)

Proof must reason about initial configuration

Sublinear time leader election from *non-uniform* initial configurations **is** possible!



$\#R = n^{1/4}$

$\#X = n - n^{1/4}$

$\#L = 0$

$R+R \rightarrow L+K$ // produce L $O(n^{1/2})$ time

$R+K \rightarrow K+K$ // stop 2nd L from being produced

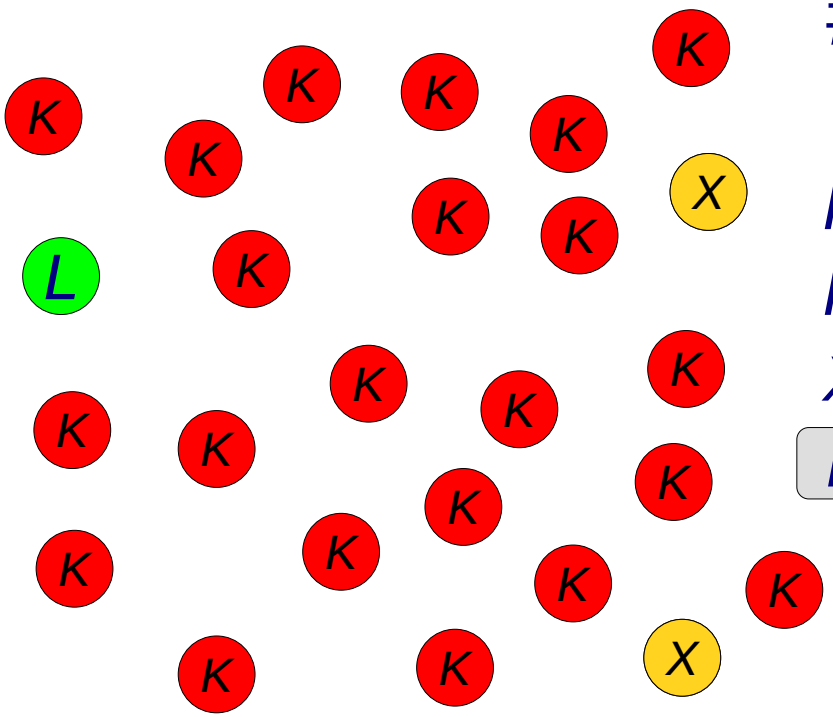
$X+K \rightarrow K+K$ // $O(\log n)$ time to complete

$L+L \rightarrow L+K$ // just in case 2nd L produced

// $O(n)$ time if needed (likely not)

Proof must reason about initial configuration

Sublinear time leader election from *non-uniform* initial configurations **is** possible!



$\#R = n^{1/4}$

$\#X = n - n^{1/4}$

$\#L = 0$

$R+R \rightarrow L+K$ // produce L $O(n^{1/2})$ time

$R+K \rightarrow K+K$ // stop 2nd L from being produced

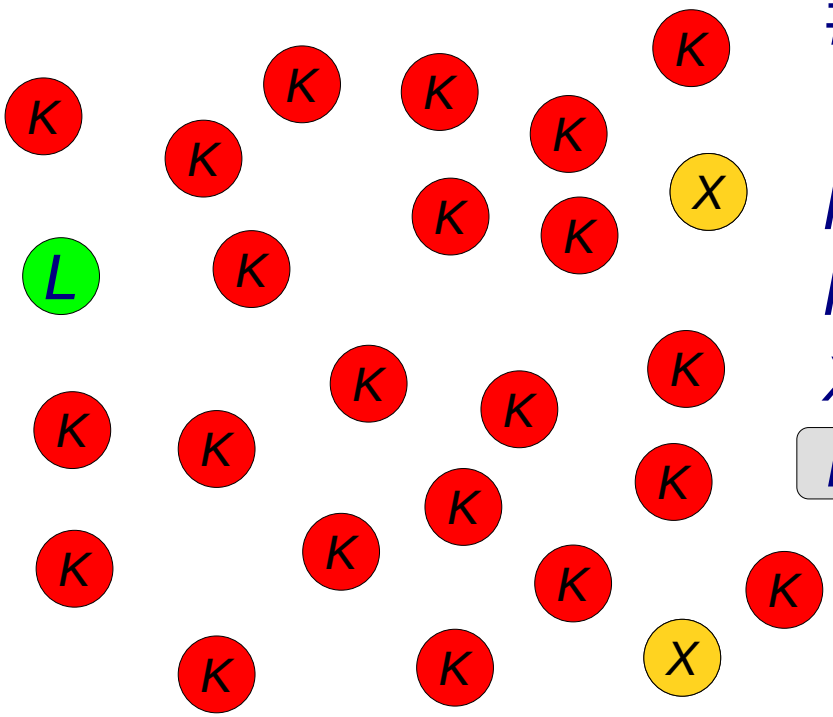
$X+K \rightarrow K+K$ // $O(\log n)$ time to complete

$L+L \rightarrow L+K$ // just in case 2nd L produced

// $O(n)$ time if needed (likely not)

Proof must reason about initial configuration

Sublinear time leader election from *non-uniform* initial configurations **is** possible!



$\#R = n^{1/4}$

$\#X = n - n^{1/4}$

$\#L = 0$

$R+R \rightarrow L+K$ // produce L $O(n^{1/2})$ time

$R+K \rightarrow K+K$ // stop 2nd L from being produced

$X+K \rightarrow K+K$ // $O(\log n)$ time to complete

$L+L \rightarrow L+K$ // just in case 2nd L produced

// $O(n)$ time if needed (likely not)

total expected time $O(n^{1/2} \log n)$
(proof not shown)

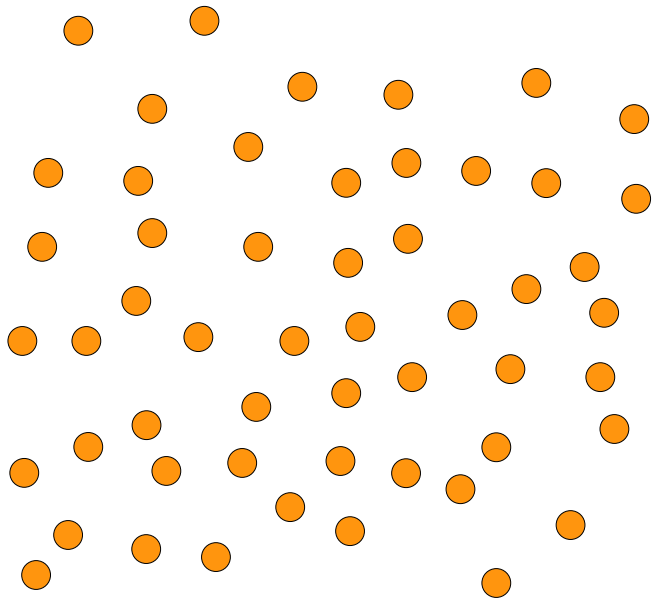
Large counts lemma

From uniform initial configuration **i** = {n A}, with probability > 99%, after $O(1)$ time, all species reach count $\Omega(n)$

[Doty, *SODA 2014: Symposium on Discrete Algorithms*]

Large counts lemma

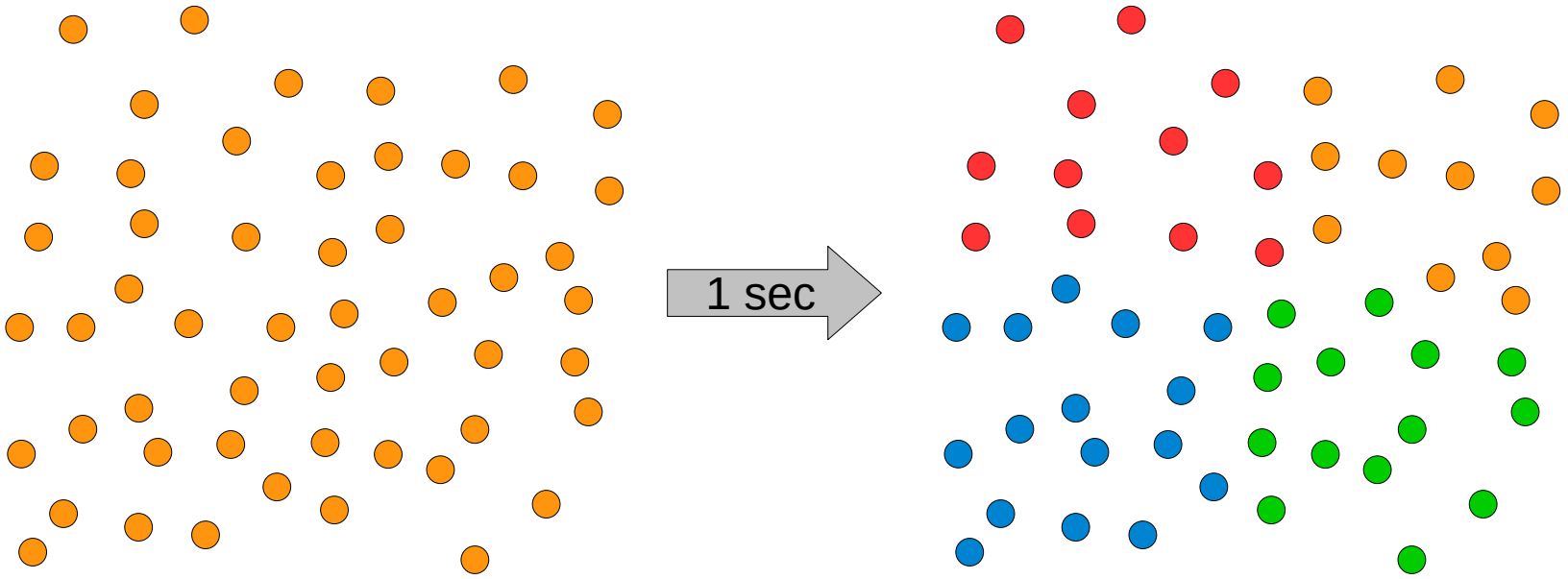
From uniform initial configuration **i** = {*n A*}, with probability > 99%, after *O*(1) time, all species reach count Ω(*n*)



[Doty, *SODA 2014: Symposium on Discrete Algorithms*]

Large counts lemma

From uniform initial configuration **i** = {*n A*}, with probability > 99%, after *O*(1) time, all species reach count Ω(*n*)



[Doty, *SODA 2014: Symposium on Discrete Algorithms*]

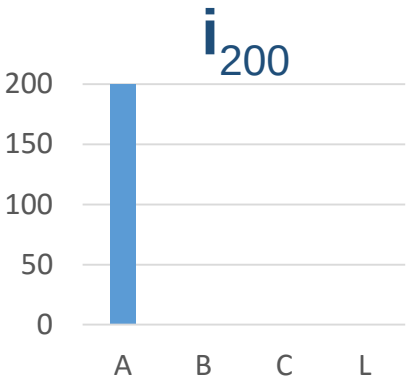
Main lemma

Main lemma

Main lemma

hypothesis of lemma

($\mathbf{i}_n$): uniform initial configurations (all n molecules are A)

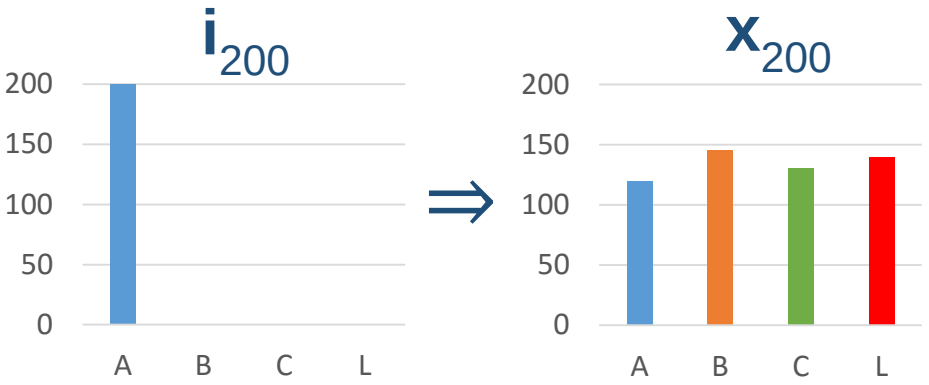


Main lemma

hypothesis of lemma

($\mathbf{i}_n$): uniform initial configurations (all n molecules are A)

($\mathbf{x}_n$): $\mathbf{i}_n \Rightarrow \mathbf{x}_n$ where $\mathbf{x}_n(S) \geq 0.01n$ for <u>all</u> species S (large counts lemma)



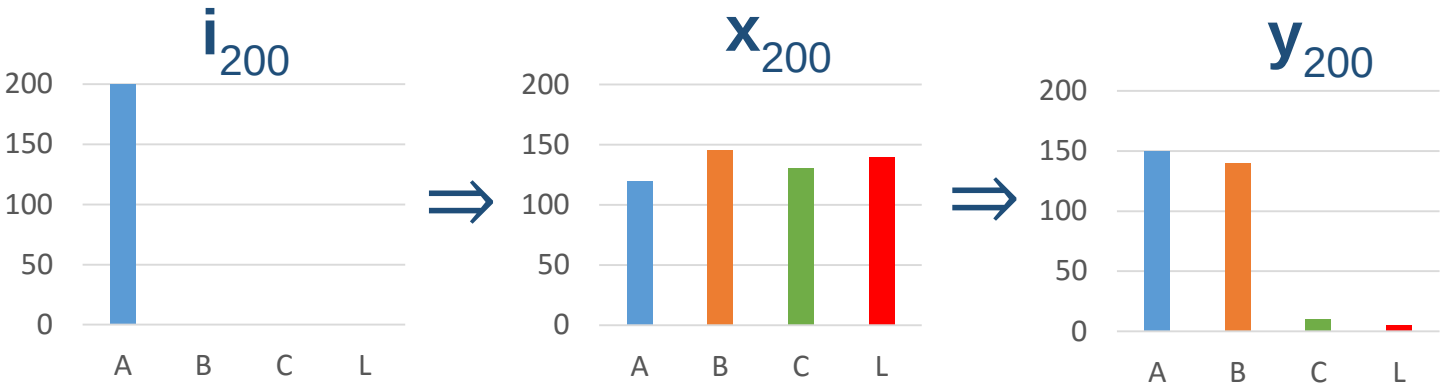
Main lemma

hypothesis of lemma

($\mathbf{i}_n$): uniform initial configurations (all n molecules are A)

($\mathbf{x}_n$): $\mathbf{i}_n \Rightarrow \mathbf{x}_n$ where $\mathbf{x}_n(S) \geq 0.01n$ for <u>all</u> species S (large counts lemma)

($\mathbf{y}_n$): $\mathbf{x}_n \Rightarrow \mathbf{y}_n$ no bottleneck (fast), $\mathbf{y}_n$ has stable leader (correct)



Main lemma

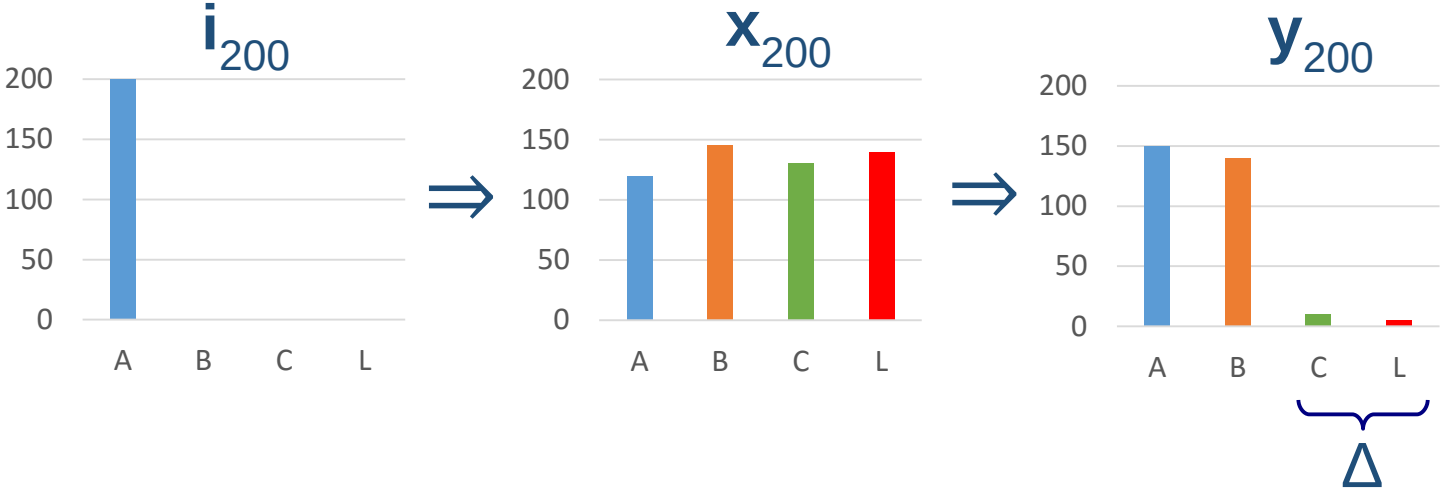
hypothesis of lemma

($\mathbf{i}_n$): uniform initial configurations (all n molecules are A)

($\mathbf{x}_n$): $\mathbf{i}_n \Rightarrow \mathbf{x}_n$ where $\mathbf{x}_n(S) \geq 0.01n$ for <u>all</u> species S (large counts lemma)

($\mathbf{y}_n$): $\mathbf{x}_n \Rightarrow \mathbf{y}_n$ no bottleneck (fast), $\mathbf{y}_n$ has stable leader (correct)

Definition. Δ = species with bounded counts in ($\mathbf{y}_n$) as $n \to \infty$



Main lemma

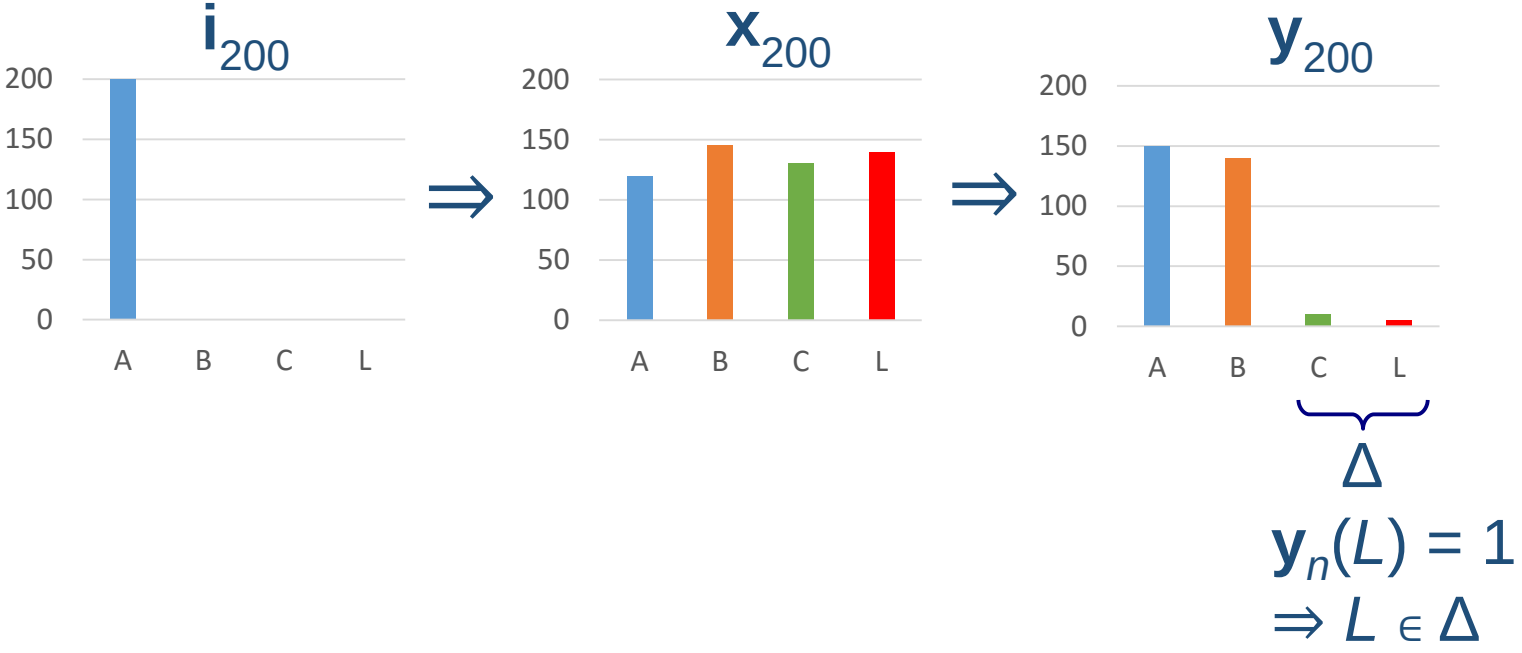
hypothesis of lemma

($\mathbf{i}_n$): uniform initial configurations (all n molecules are A)

($\mathbf{x}_n$): $\mathbf{i}_n \Rightarrow \mathbf{x}_n$ where $\mathbf{x}_n(S) \geq 0.01n$ for <u>all</u> species S (large counts lemma)

($\mathbf{y}_n$): $\mathbf{x}_n \Rightarrow \mathbf{y}_n$ no bottleneck (fast), $\mathbf{y}_n$ has stable leader (correct)

Definition. Δ = species with bounded counts in ($\mathbf{y}_n$) as $n \to \infty$



$\mathbf{y}_n(L) = 1$

$\Rightarrow L \in \Delta$

Main lemma

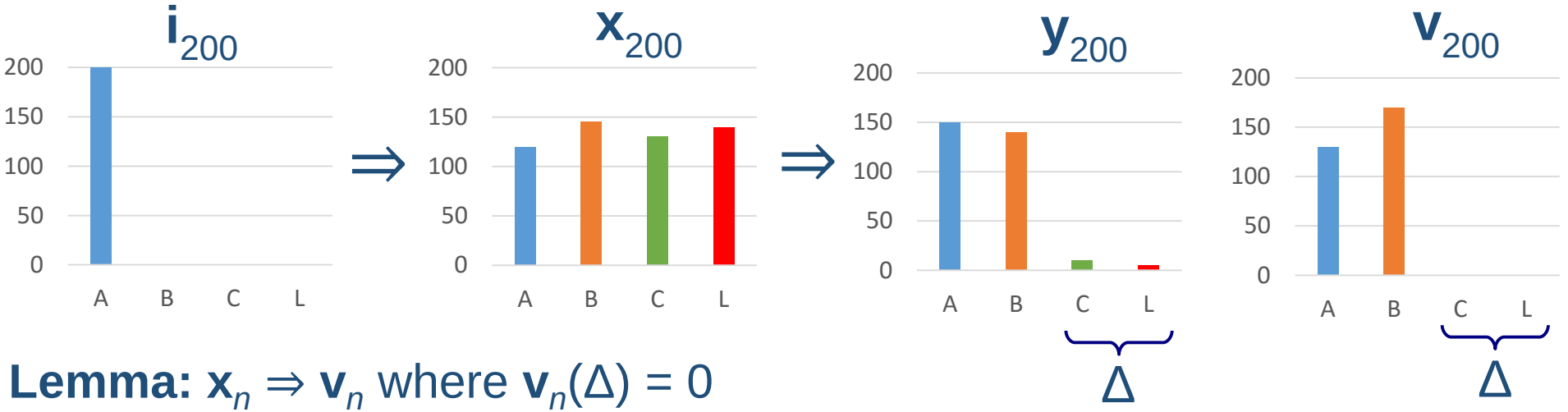
hypothesis of lemma

($\mathbf{i}_n$): uniform initial configurations (all n molecules are A)

($\mathbf{x}_n$): $\mathbf{i}_n \Rightarrow \mathbf{x}_n$ where $\mathbf{x}_n(S) \geq 0.01n$ for <u>all</u> species S (large counts lemma)

($\mathbf{y}_n$): $\mathbf{x}_n \Rightarrow \mathbf{y}_n$ no bottleneck (fast), $\mathbf{y}_n$ has stable leader (correct)

Definition. Δ = species with bounded counts in ($\mathbf{y}_n$) as $n \to \infty$



Lemma: $\mathbf{x}_n \Rightarrow \mathbf{v}_n$ where $\mathbf{v}_n(\Delta) = 0$

$\mathbf{y}_n(L) = 1$
$\Rightarrow L \in \Delta$

Main lemma

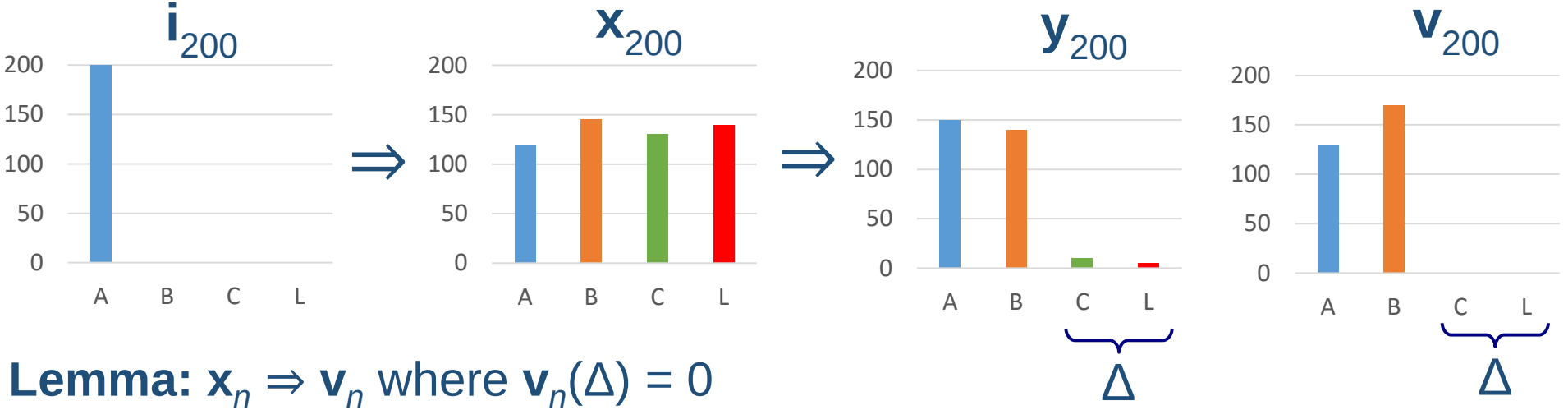
hypothesis of lemma

($\mathbf{i}_n$): uniform initial configurations (all n molecules are A)

($\mathbf{x}_n$): $\mathbf{i}_n \Rightarrow \mathbf{x}_n$ where $\mathbf{x}_n(S) \geq 0.01n$ for <u>all</u> species S (large counts lemma)

($\mathbf{y}_n$): $\mathbf{x}_n \Rightarrow \mathbf{y}_n$ no bottleneck (fast), $\mathbf{y}_n$ has stable leader (correct)

Definition. Δ = species with bounded counts in ($\mathbf{y}_n$) as $n \to \infty$



Lemma: $\mathbf{x}_n \Rightarrow \mathbf{v}_n$ where $\mathbf{v}_n(\Delta) = 0$

$\mathbf{y}_n(L) = 1$

$\Rightarrow L \in \Delta \quad \Rightarrow \mathbf{v}_n(L) = 0$

Main lemma

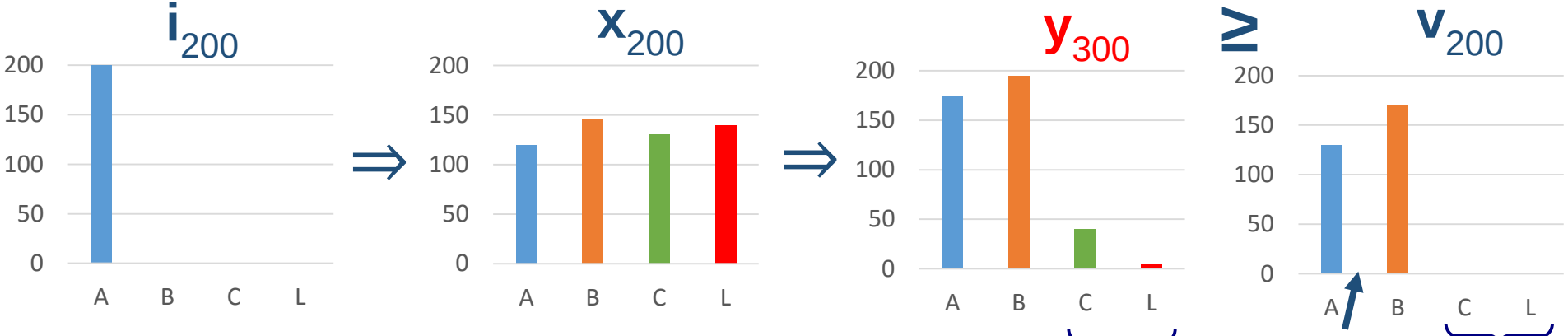
hypothesis of lemma

($\mathbf{i}_n$): uniform initial configurations (all n molecules are A)

($\mathbf{x}_n$): $\mathbf{i}_n \Rightarrow \mathbf{x}_n$ where $\mathbf{x}_n(S) \geq 0.01n$ for <u>all</u> species S (large counts lemma)

($\mathbf{y}_n$): $\mathbf{x}_n \Rightarrow \mathbf{y}_n$ no bottleneck (fast), $\mathbf{y}_n$ has stable leader (correct)

Definition. Δ = species with bounded counts in ($\mathbf{y}_n$) as $n \to \infty$



Lemma: $\mathbf{x}_n \Rightarrow \mathbf{v}_n$ where $\mathbf{v}_n(\Delta) = 0$

$\mathbf{y}_n(L) = 1$

$\Rightarrow L \in \Delta \quad \Rightarrow \mathbf{v}_n(L) = 0$

Main lemma

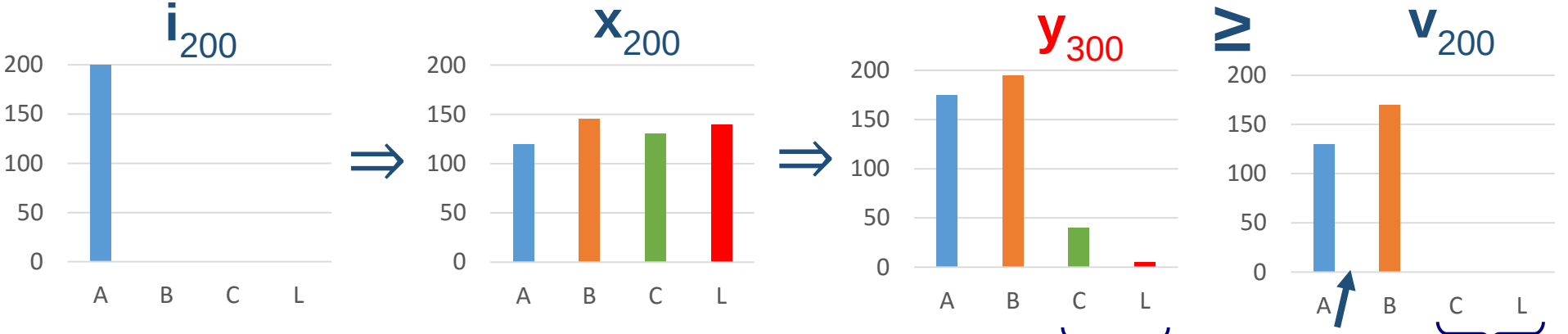
hypothesis of lemma

($\mathbf{i}_n$): uniform initial configurations (all n molecules are A)

($\mathbf{x}_n$): $\mathbf{i}_n \Rightarrow \mathbf{x}_n$ where $\mathbf{x}_n(S) \geq 0.01n$ for <u>all</u> species S (large counts lemma)

($\mathbf{y}_n$): $\mathbf{x}_n \Rightarrow \mathbf{y}_n$ no bottleneck (fast), $\mathbf{y}_n$ has stable leader (correct)

Definition. Δ = species with bounded counts in ($\mathbf{y}_n$) as $n \to \infty$



Lemma: $\mathbf{x}_n \Rightarrow \mathbf{v}_n$ where $\mathbf{v}_n(\Delta) = 0$

$\mathbf{y}_n(L) = 1$
$\Rightarrow L \in \Delta$ $\Rightarrow \mathbf{v}_n(L) = 0$

i.e., leader not elected with probability 1

Main lemma

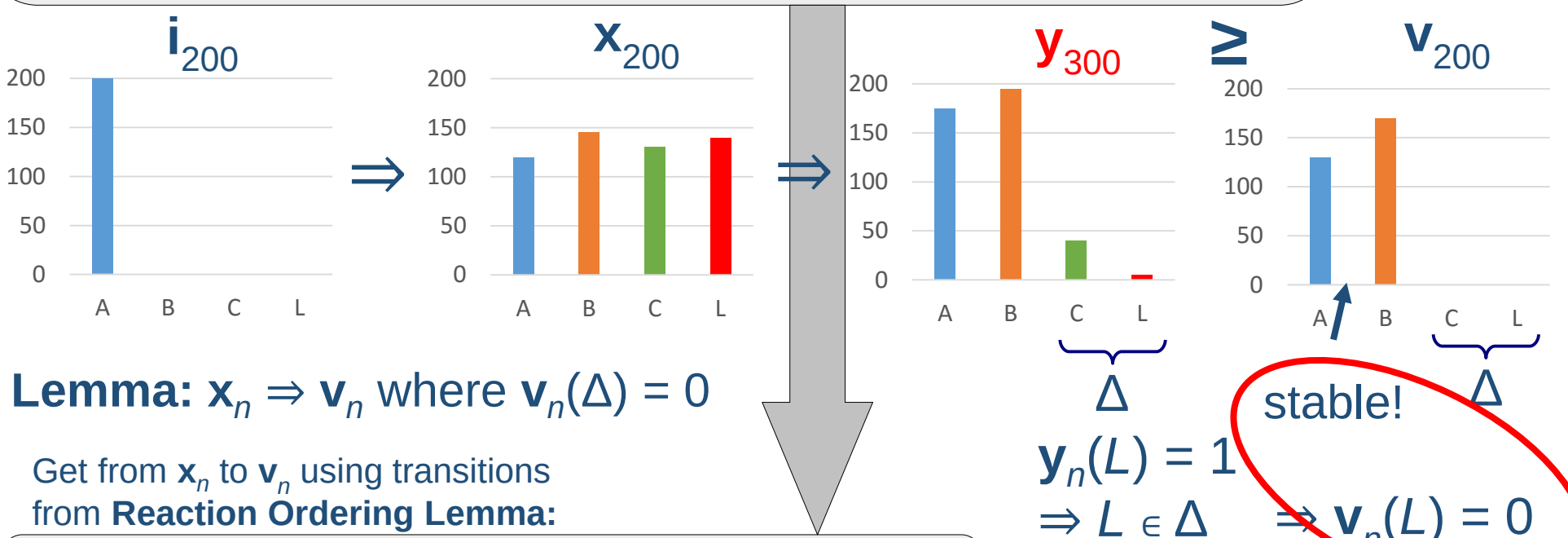
hypothesis of lemma

($\mathbf{i}_n$): uniform initial configurations (all n molecules are A)

($\mathbf{x}_n$): $\mathbf{i}_n \Rightarrow \mathbf{x}_n$ where $\mathbf{x}_n(S) \geq 0.01n$ for <u>all</u> species S (large counts lemma)

($\mathbf{y}_n$): $\mathbf{x}_n \Rightarrow \mathbf{y}_n$ no bottleneck (fast), $\mathbf{y}_n$ has stable leader (correct)

Definition. Δ = species with bounded counts in ($\mathbf{y}_n$) as $n \to \infty$



Lemma: $\mathbf{x}_n \Rightarrow \mathbf{v}_n$ where $\mathbf{v}_n(\Delta) = 0$

Get from $\mathbf{x}_n$ to $\mathbf{v}_n$ using transitions from **Reaction Ordering Lemma:**

There is an order on $\Delta = \{ D_1, D_2, \ldots, D_k \}$, and reactions $D_j + R \to P + P'$ where $R, P, P' \notin \{ D_1, D_2, \ldots, D_j \}$.

[Chen, Cummings, Doty, Soloveichik, *DISC 2014: Symposium on Distributed Computing*]

$\mathbf{y}_n(L) = 1$

$\Rightarrow L \in \Delta \quad \Rightarrow \mathbf{v}_n(L) = 0$

i.e., leader not elected with probability 1

Summary

Summary

Every (reasonable) CRN that stably elects a leader from a uniform initial configuration (all molecules are same species) requires time $\Omega(n)$.

Summary

Every (reasonable) CRN that stably elects a leader from a uniform initial configuration (all molecules are same species) requires time $\Omega(n)$.

More generally, negative result holds even if:

- initial configuration nonuniform but **dense** (each species present has count $\Omega(n)$)

Summary

Every (reasonable) CRN that stably elects a leader from a uniform initial configuration (all molecules are same species) requires time $\Omega(n)$.

More generally, negative result holds even if:

- initial configuration nonuniform but **dense** (each species present has count $\Omega(n)$)
- goal is **any** positive number of leaders under some constant

Summary

Every (reasonable) CRN that stably elects a leader from a uniform initial configuration (all molecules are same species) requires time $\Omega(n)$.

More generally, negative result holds even if:

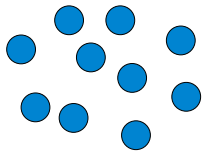
- initial configuration nonuniform but **dense** (each species present has count $\Omega(n)$)
- goal is **any** positive number of leaders under some constant
- there are **multiple** leader species $L_1, L_2, \ldots, L_k$ and we want to stabilize to $\sum \#L_i = 1$

Open questions

Convergence versus stabilization

Open questions

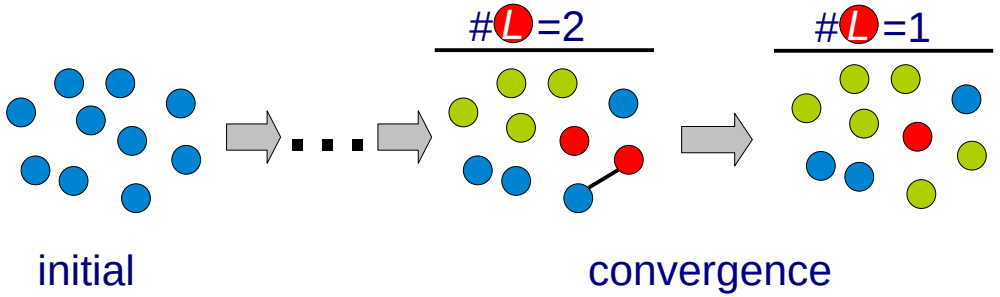
Convergence versus stabilization



initial

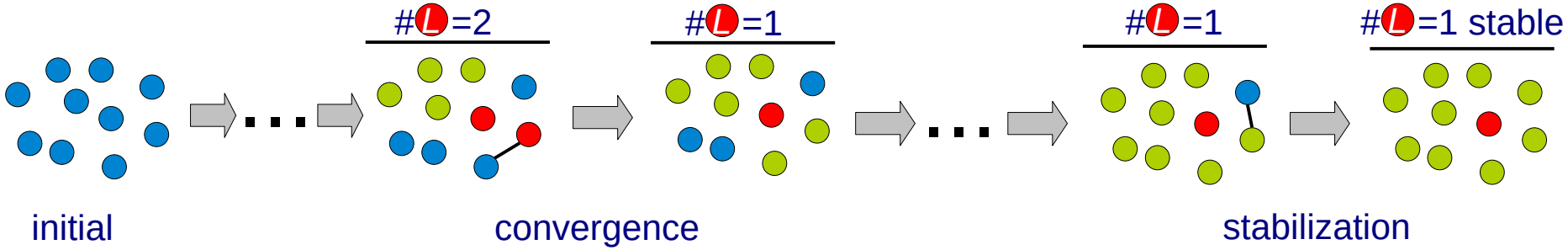
Open questions

Convergence versus stabilization



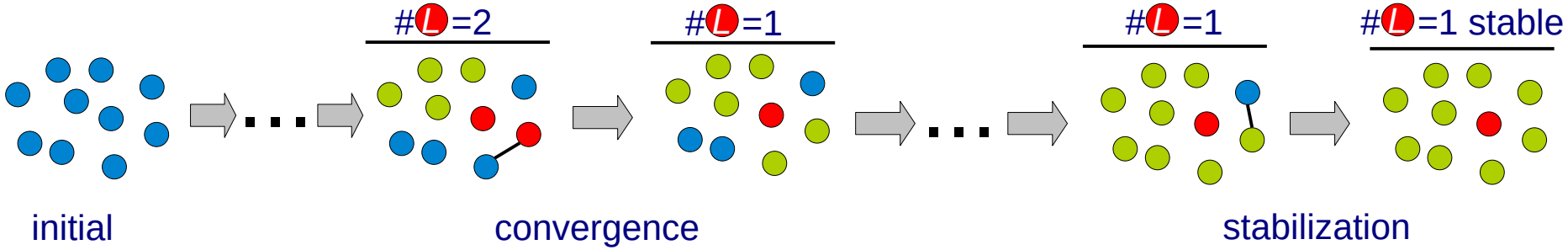
Open questions

Convergence versus stabilization



Open questions

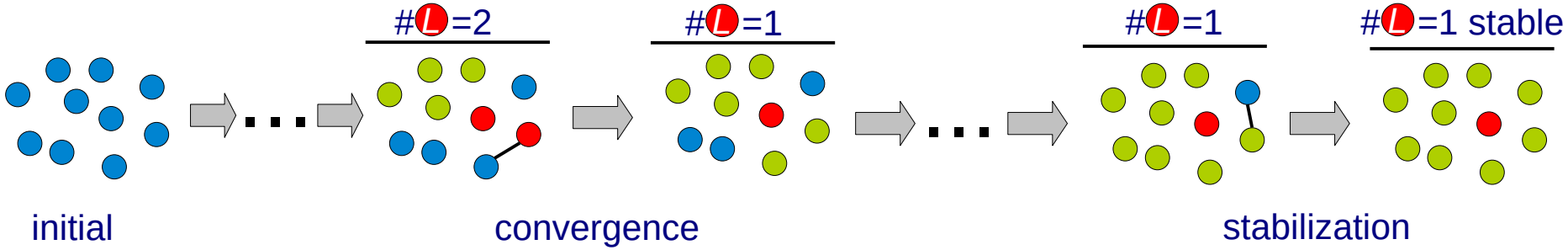
Convergence versus stabilization



Leader election takes Ω(*n*) time to stabilize; what about convergence?

Open questions

Convergence versus stabilization



Leader election takes $\Omega(n)$ time to stabilize; what about convergence?

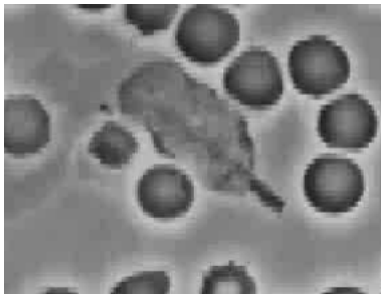
What initial configurations allow sublinear leader election?

dense configurations like { $n/2$ A, $n/4$ B, $n/4$ C }: **NO**

non-dense configurations like { n-$n^{1/4}$ X, $n^{1/4}$ R }: **YES**

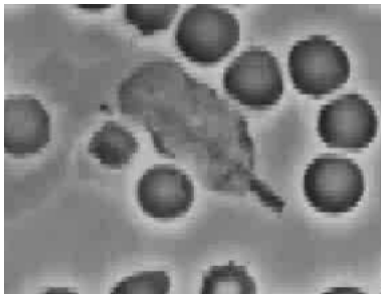
Take-home message (engineer’s perspective)

Take-home message (engineer's perspective)



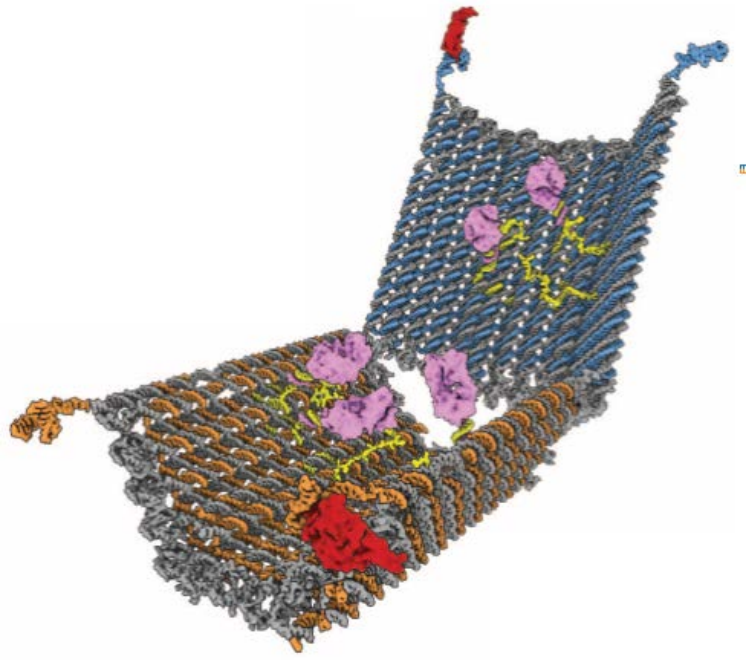
Ask not what nature *does*

Take-home message (engineer's perspective)



Ask not what nature *does*

Ask what nature *could* do…



Recruiting Ph.D. Students

David Soloveichik

David Doty

The University of Texas at Austin
Electrical and Computer Engineering

UCDAVIS
COMPUTER SCIENCE