

On Design and Analysis of Chemical Reaction Network Algorithms

Anne Condon

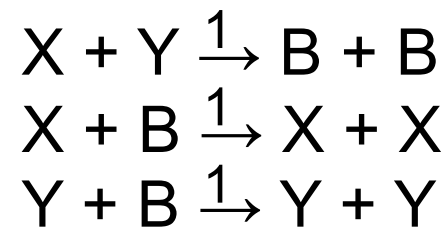
with Ben Chugg, Monir Hajiaghayi, David Kirkpatrick, Jan Manuch
The University of British Columbia

My Motivation: Computing in a Test Tube

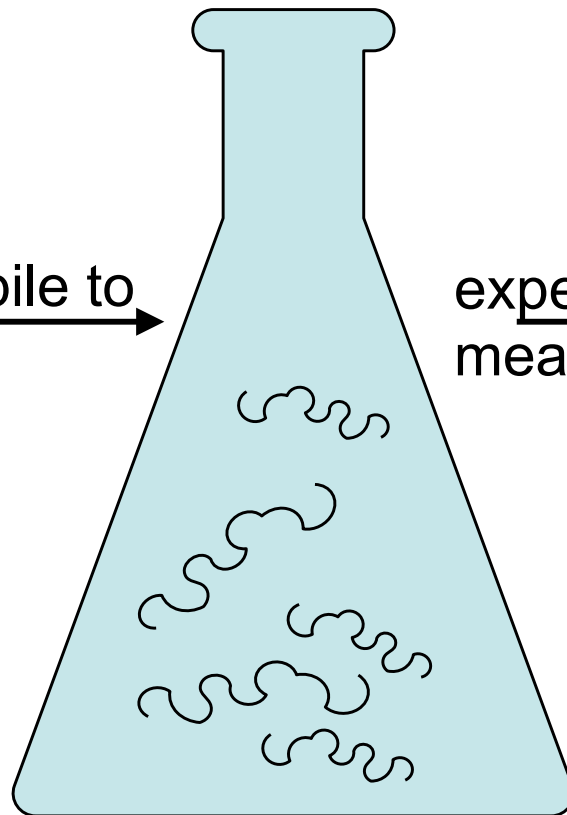
- Computing need not be limited to silicon!
- Computing with digital biomolecules such as DNA can facilitate sensing and mediation in wet environments, and can help us understand what goes on in such environments

My Motivation: Computing in a Test Tube

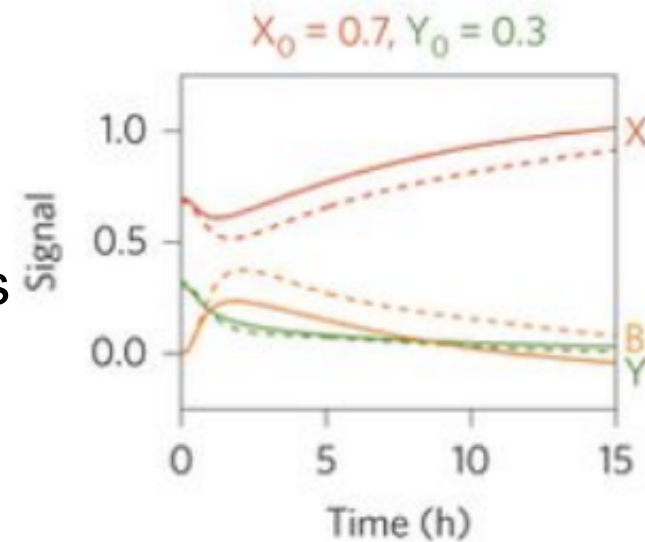
Chemical Reaction Network (CRN)



compile to
DNA



experimental
measurements



Soloveichik et al., 2013

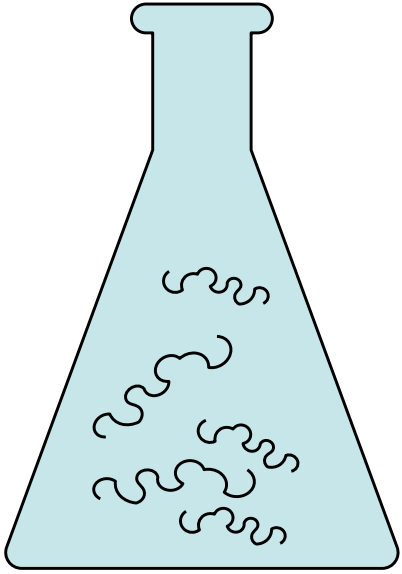
Chen et al., 2013

This Talk

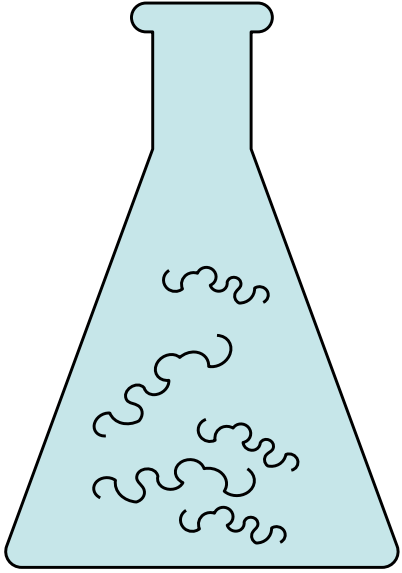
- What is the CRN computation model?
- Simple analysis of Approximate Majority CRNs
- On composing function-computing CRNs

Chemical Reaction Network (CRNs)

Chemical Reaction Network (CRNs)



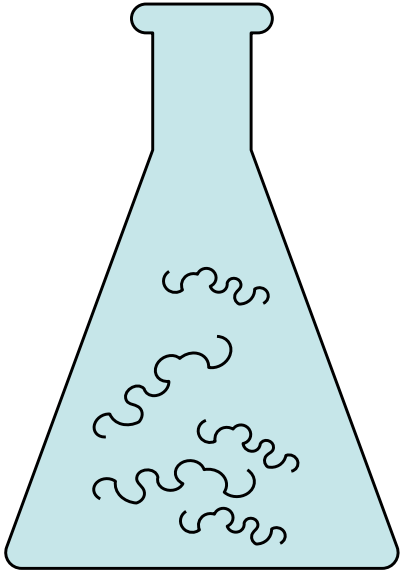
Chemical Reaction Network (CRNs)



Interactions

- Initially a well-mixed test tube contains n molecules, drawn from m species types
- Interactions of a fixed order o happen when o molecules collide
- An interaction is equally likely to involve any o of the constituent molecules
- This is a *stochastic* (as opposed to mass action), *asynchronous* model
- If the volume is proportional to n , the expected time for n interactions is $\Theta(1)$

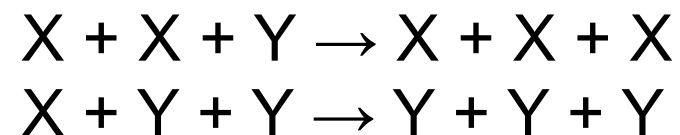
Chemical Reaction Network (CRNs)



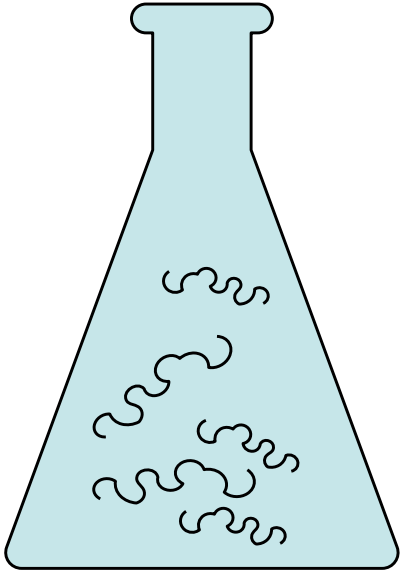
Reactions

- Some interactions may trigger *productive reactions* that change species counts, while preserving the total molecular count
- If species counts are $x_1 \dots x_m$ and reaction $r = (s_1 \dots s_m) \xrightarrow{k_r} (p_1 \dots p_m)$ is *applicable*, i.e., $s_i \leq x_i$, then the probability that an interaction results in reaction r is

$$k_r \left[\prod_{i=1}^m \binom{x_i}{s_i} \right] / \binom{n}{o}$$

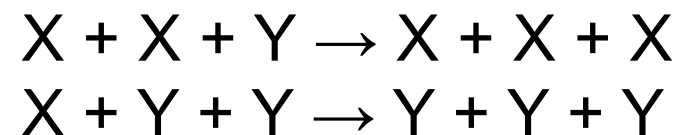


Chemical Reaction Network (CRNs)



Computations

- Starting from given initial configuration (vector of species counts), a random sequence of interactions triggers a sequence of (not necessarily productive) reaction events
- The resulting random sequence of configurations is a *computation*
- The expected time for the computation is the number of interactions divided by n , the total molecule count



Approximate Majority

Approximate Majority

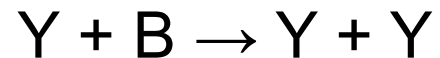
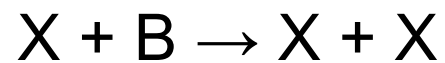
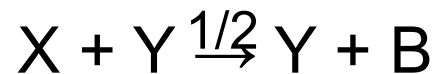
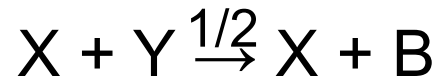
- Given an initial mixture with n molecules, some of species X and the rest of species Y , the goal is to reach consensus on the majority species, assuming the initial gap is large: $\Omega(\sqrt{n \lg n})$

Approximate Majority Background

- Widely studied in distributed systems, epidemiology, social networks, and voting theory [Becchetti et al. 2014, 2015; Cruise & Ganesh 2013; Doerr et al., 2011; Mossel et al., 2014; Perron et al., 2009; Mertzios et al., 2017; ...]
- Comparing counts is a basic building block in simulating counter machines by population protocols [Angluin et al., 2004]
- Chemical reaction networks that solve approximate majority can be found in the cell cycle switch in eukaryotes that induces mitosis [Cardelli & Csikász-Nagy 2012]

Approximate Majority Background

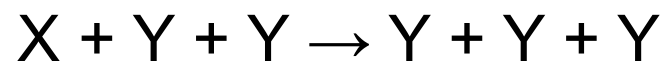
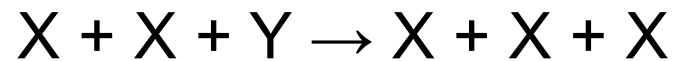
- Most closely related to our work is a population protocol of Angluin, Aspnes, and Eisenstat 2006, the Single-B CRN:



- “Unfortunately, while the protocol itself is simple, proving that it converges quickly appears to be very difficult” [Angluin et al.]

Approximate Majority: Our Work

- We provide a simple proof of correctness and efficiency of Single-B
- We first analyze a tri-molecular CRN, TRI:

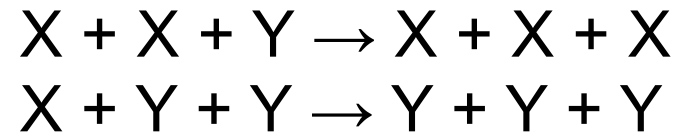


- We then show how Double-B and Single-B CRN emulate TRI
- We use the same general approach to analyze many variants: multi-valued consensus, consensus with uncertain reaction rates, Byzantine agents, ...

Approximate Majority: TRI Analysis

$$\begin{aligned} X + X + Y &\rightarrow X + X + X \\ X + Y + Y &\rightarrow Y + Y + Y \end{aligned}$$

Approximate Majority: TRI Analysis



Theorem: For any $\gamma \geq 1$, a computation of TRI reaches consensus on X , with probability $1 - \exp(-\Omega(\gamma \lg n))$, in $O(\gamma n \lg n)$ interaction events, provided initially the count of X exceeds that of Y by at least $\sqrt{\gamma n \lg n}$

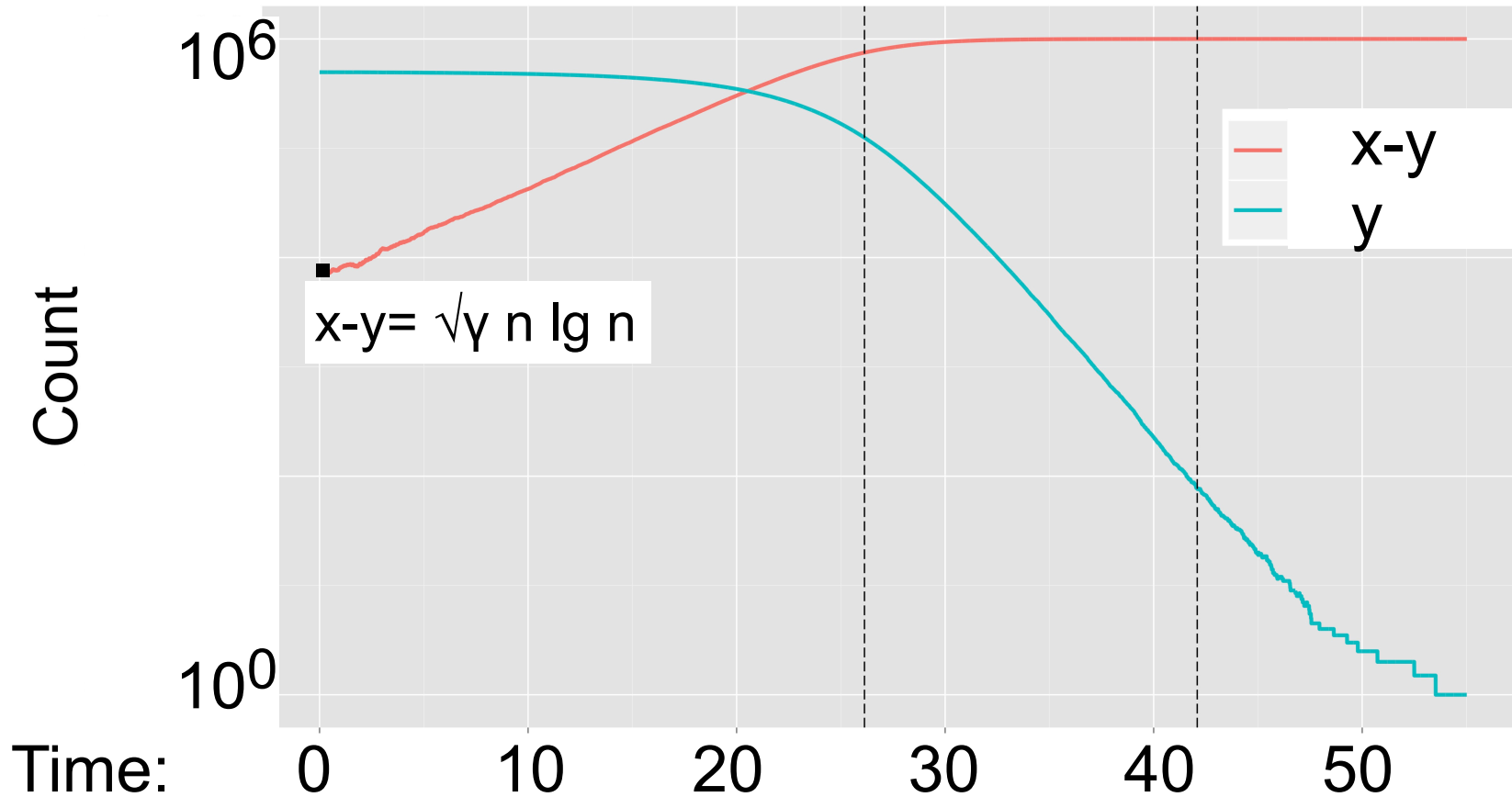
Approximate Majority: TRI Analysis

Analysis Tools:

Biased one-dimensional random walk: In a sequence of independent trials, each with success probability at least $p > 1/2$, the probability that the number of failures ever exceeds the number of successes by b is at most $((1-p)/p)^b$

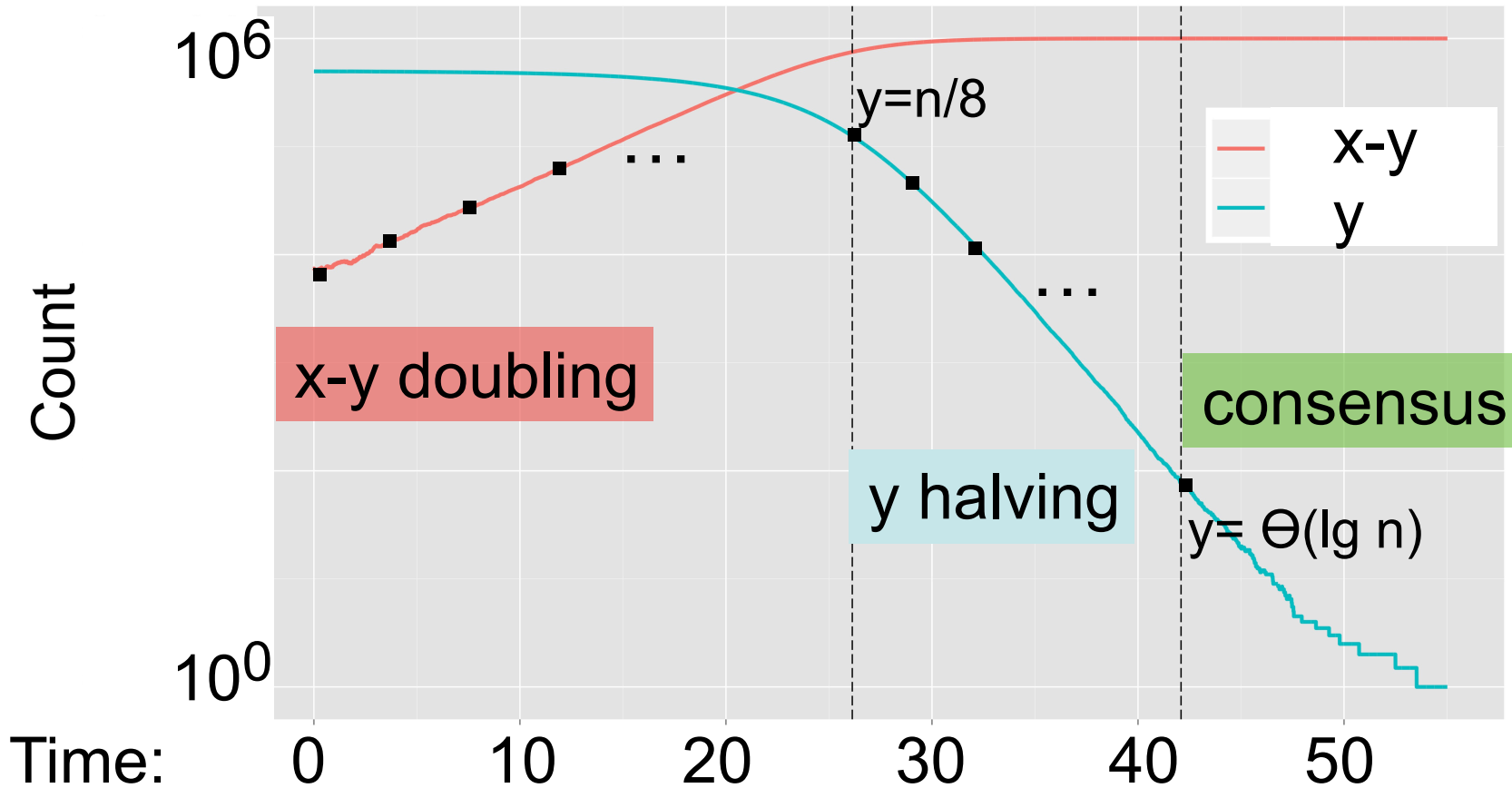
Chernoff bounds: In a sequence of independent trials, the probability that the number of successes differs from the expected value μ by more than $\delta\mu$ is at most $\exp(-\delta^2\mu/2)$

Approximate Majority: TRI Analysis



Random variables x and y denote the number of copies of X and Y during a computation of TRI

Approximate Majority: TRI Analysis

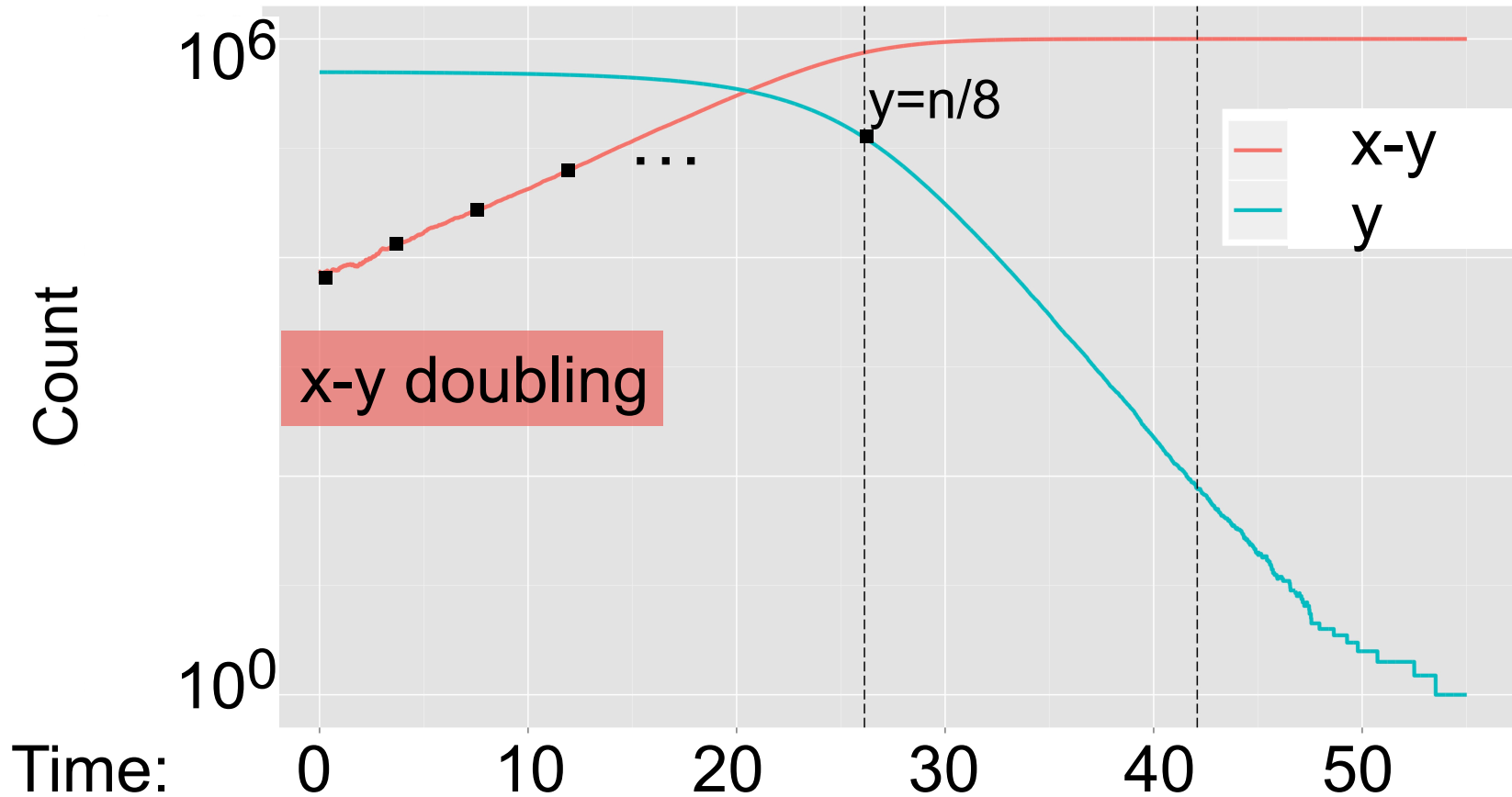


x-y doubling: $O(\lg n)$ stages with $O(n)$ interactions each

y halving: $O(\lg n)$ stages with $O(n)$ interactions each

consensus: One stage with $O(n \lg n)$ interactions

Approximate Majority: TRI Analysis



Analysis of a single stage of x-y doubling:

(a) low prob of significant backsliding

(b) assuming (a), high prob of doubling in $O(n)$ productive reactions

(c) high probability that (b) occurs in $O(n)$ interactions

Approximate Majority: Double-B Analysis

$$X + Y \rightarrow B + B \quad (0)$$

$$X + B \rightarrow X + X \quad (1)$$

$$Y + B \rightarrow Y + Y \quad (2)$$

Approximate Majority: Double-B Analysis



Correctness:

- Let b be the count of B's
- Let $\dot{x} = x + b/2$ and $\dot{y} = y + b/2$
- Reaction (0) leaves \dot{x} and \dot{y} unchanged,
- Reactions (1) and (2) change \dot{x} and \dot{y} by $1/2$ exactly as the two TRI reactions change x and y by 1

Approximate Majority: Other Analyses

Using same “emulation” approach, we can analyze several other CRNs:

- Multi-valued consensus
- Uncertain reaction rates
- Byzantine agents
- Initiation by infection

Approximate Majority: Summary

Simplicity achieved by

- Starting with the tri-molecular CRN
- Analyzing short stages where quantities don't change by more than a constant factor
- Separating analysis of productive reactions vs interactions

Approximate Majority: Open Problems

- Simple argument that consensus is reached quickly with a small gap (even if high error)?
- Algorithmic Chernoff bound?
- Analysis of the biological variants described by Cardelli et al?

This Talk

- What is the CRN computation model?
- Simple analysis of Approximate Majority CRNs
- On composing function-computing CRNs

What else can be computed by CRNs?

- Inputs n_1, \dots, n_k are represented by (unary) counts of species X_1, \dots, X_k
- Total count of input molecules is n

Prob[correct]<1	all computable functions	unbounded volume	Angluin et al., Cook et al.
Prob[correct]=1 (<i>Stable computation</i>)	semilinear functions	$\Theta(n)$ volume	Angluin et al., Doty et al.

What Are Semilinear Functions?

Semilinear functions $\mathbb{N}^k \rightarrow \mathbb{N}$ are expressible as a finite number of affine linear pieces over linear domains (whose union is \mathbb{N}^k)

Examples: mod, min, sum, difference, or compositions of these

Stable CRNs for Semilinear Functions

Example: mod

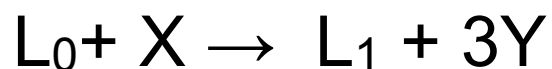
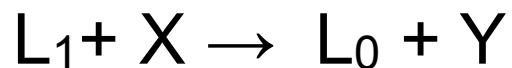
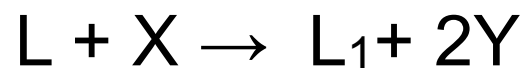
$$\begin{aligned} f(n) &= 2n-1, & n &= 0 \pmod{2} & // \text{ linear set } \{ 2i \mid i \in \mathbb{N} \} \\ &2n, & n &= 1 \pmod{2} \end{aligned}$$

Stable CRNs for Semilinear Functions

Example: mod

$$f(n) = \begin{cases} 2n-1, & n = 0 \pmod{2} \\ 2n, & n = 1 \pmod{2} \end{cases} \quad // \text{ linear set } \{ 2i \mid i \in \mathbb{N} \}$$

Stable CRN, with n copies of X as input, *plus a leader* L :

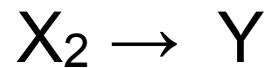
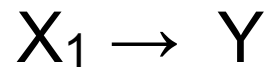


Stable CRNs for Semilinear Functions

Example: sum

$$f(n_1, n_2) = n_1 + n_2$$

Stable CRN, with n_1, n_2 copies of X_1, X_2 as input:



Stable CRNs for Semilinear Functions

Example: max

$$f(n_1, n_2) = \max(n_1, n_2) [= n_1 + n_2 - \min(n_1, n_2)]$$

$$f(n_1, n_2) = n_2, n_1 < n_2 \quad // \text{ linear set } \{ (0,1) + i_1(0,1) + i_2(1,1) \mid i \in \mathbb{N} \}$$
$$n_1, n_1 \geq n_2$$

Stable CRNs for Semilinear Functions

Example: max

$$f(n_1, n_2) = \max(n_1, n_2) [= n_1 + n_2 - \min(n_1, n_2)]$$

Stable CRNs for Semilinear Functions

Example: max

$$f(n_1, n_2) = \max(n_1, n_2) [= n_1 + n_2 - \min(n_1, n_2)]$$

Stable CRN, with n_1, n_2 copies of X_1, X_2 as input:

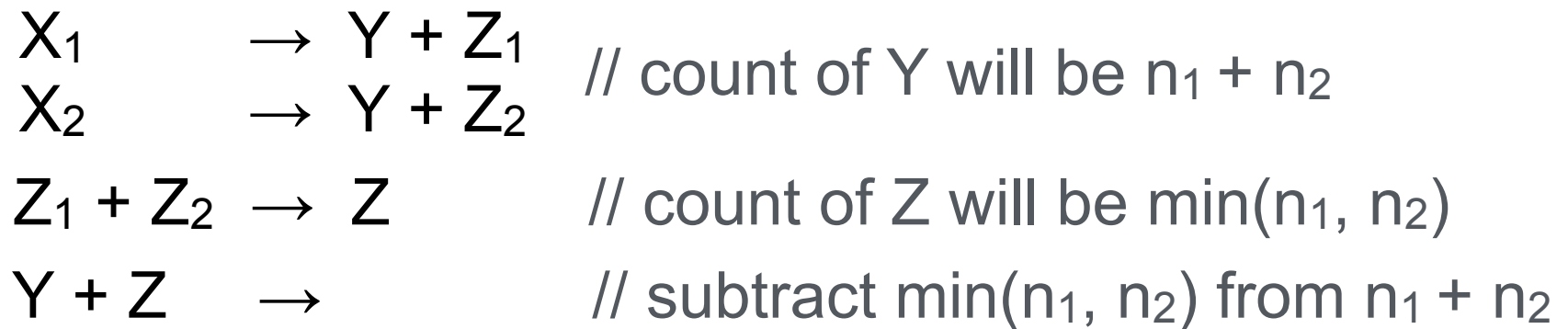


Stable CRNs for Semilinear Functions

Example: max

$$f(n_1, n_2) = \max(n_1, n_2) [= n_1 + n_2 - \min(n_1, n_2)]$$

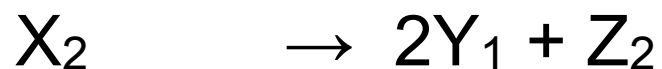
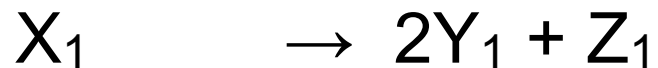
Stable CRN, with n_1, n_2 copies of X_1, X_2 as input:



Stable CRNs for Semilinear Functions

Example: What about $\min \{ 2\max(n_1, n_2), n_1 + 2n_2 \}$?

CRN for $2\max$ (with o/p Y_1)



CRN for $n_1 + 2n_2$ (with o/p Y_2)



Y_1

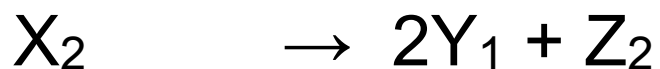
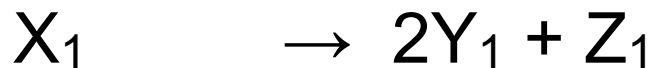
Y_2

CRN for \min
 $Y_1 + Y_2 \rightarrow Y$

Stable CRNs for Semilinear Functions

Example: What about $\min \{ 2\max(n_1, n_2), n_1 + 2n_2 \}$?

CRN for $2\max$ (with o/p Y_1)



CRN for $n_1 + 2n_2$ (with o/p Y_2)



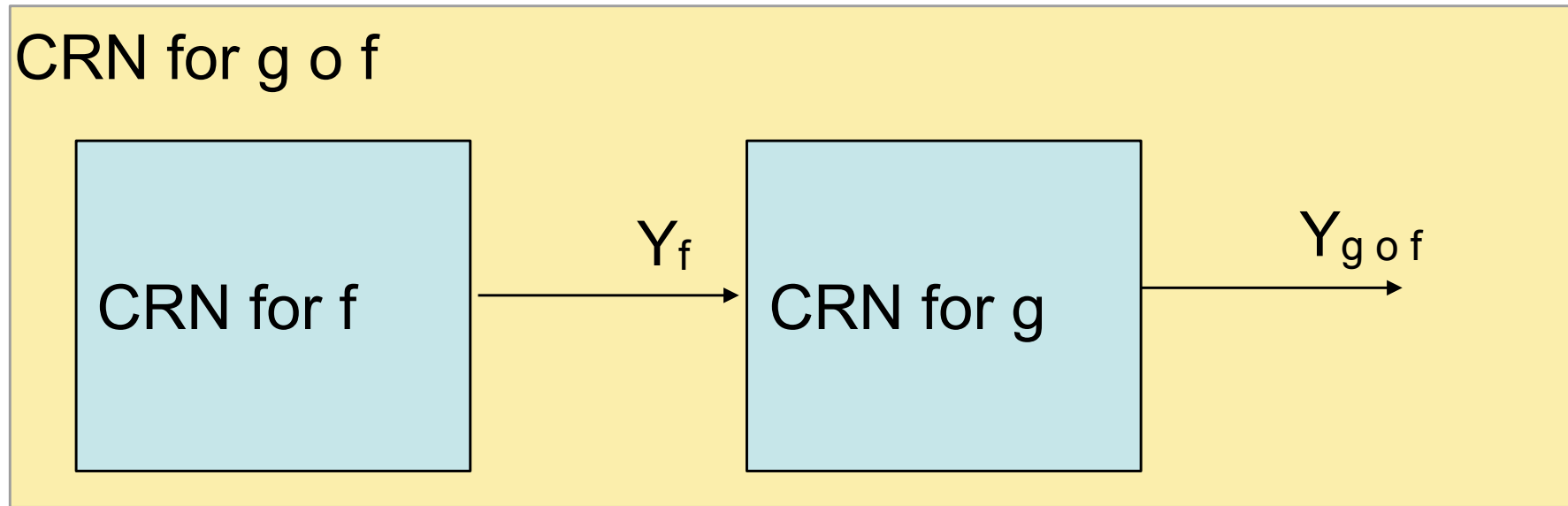
Y_1

Y_2

CRN for \min
 $Y_1 + Y_2 \rightarrow Y$

Problem: the min CRN may consume Y_1 's before the "2max" CRN has stabilized

Stable Function Composition



- Composition in this way is correct if the CRN for f is
- *non-decreasing*
 - *output oblivious*: its output is not a reactant of any of its reactions

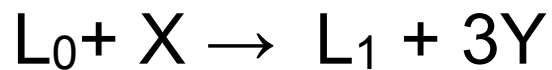
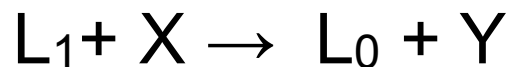
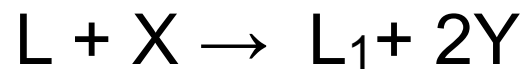
What Functions Are Output Oblivious?

Example: mod *if the linear coefficients match*

$$f(n) = 2n-1, \quad n = 0 \pmod{2}$$

$$2n, \quad n = 1 \pmod{2}$$

Stable CRN, with n copies of X as input, plus a leader L :



What Functions Are Output Oblivious?

Example: $\min(n_1, n_2)$

Stable CRN, with n_1 and n_2 copies of X_1 , and X_2 as input:



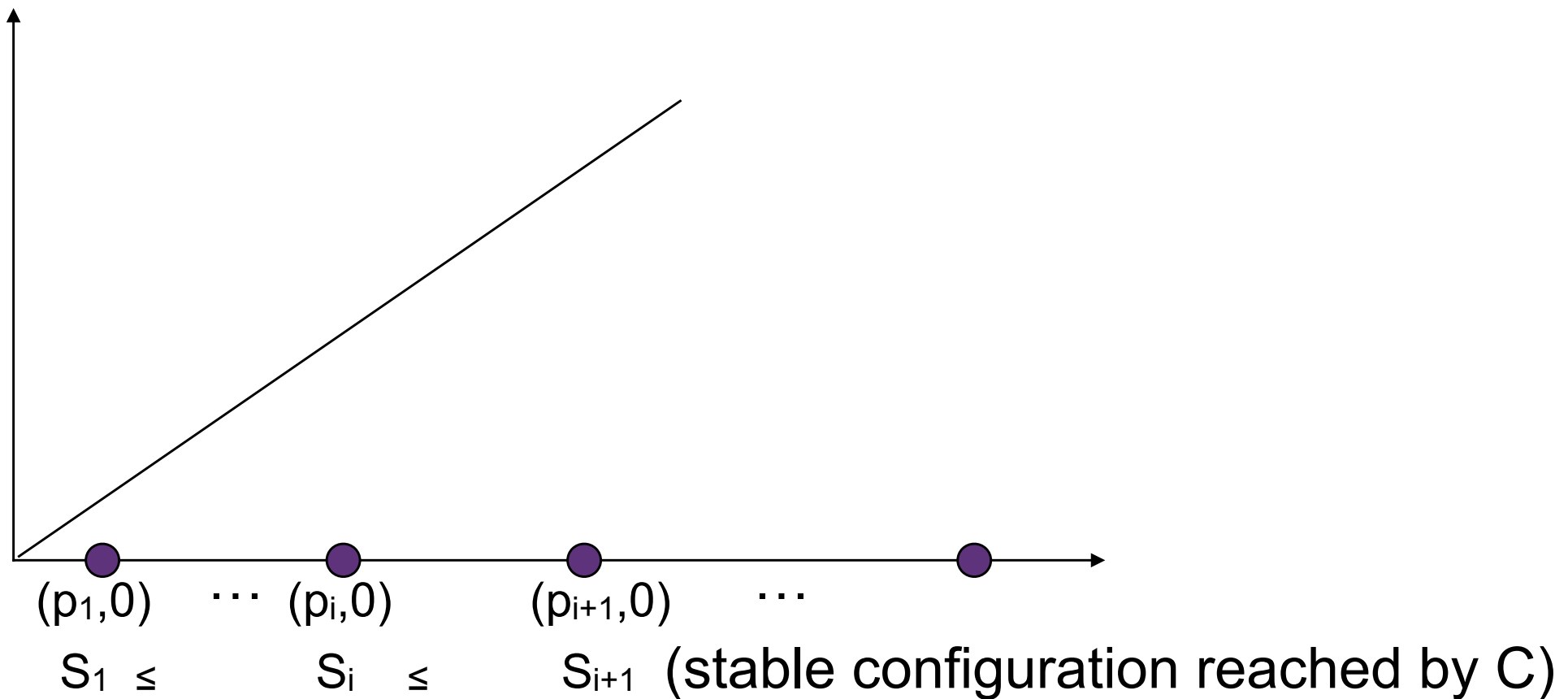
What Functions Are Output Oblivious?

Example: $\max(n_1, n_2)$ is *not* output oblivious

What Functions Are Output Oblivious?

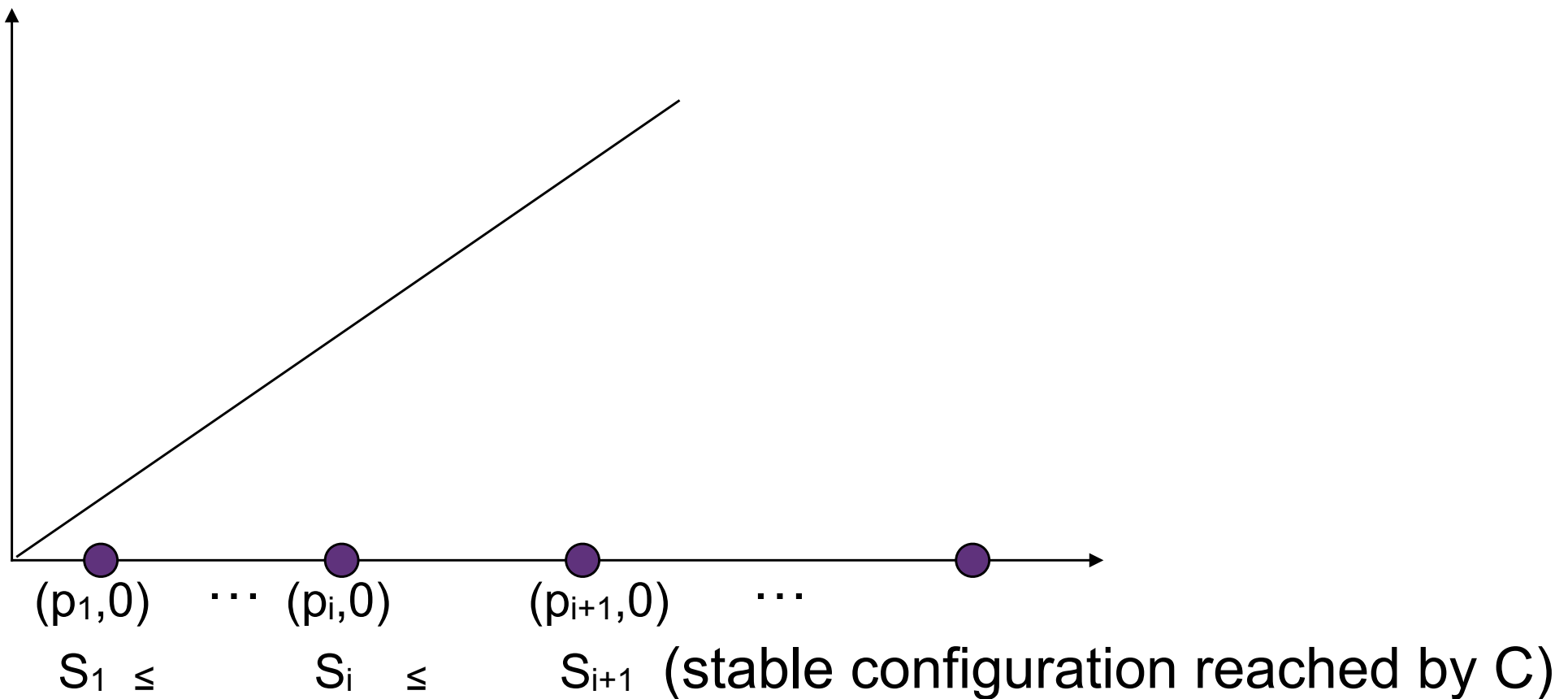
Example: $\max(n_1, n_2)$ is *not* output oblivious

Choose $p_1 < p_2 < \dots < p_i$ such that if S_i is a stable configuration reached by C on $(p_i, 0)$, then $S_i \leq S_{i+1}$



What Functions Are Output Oblivious?

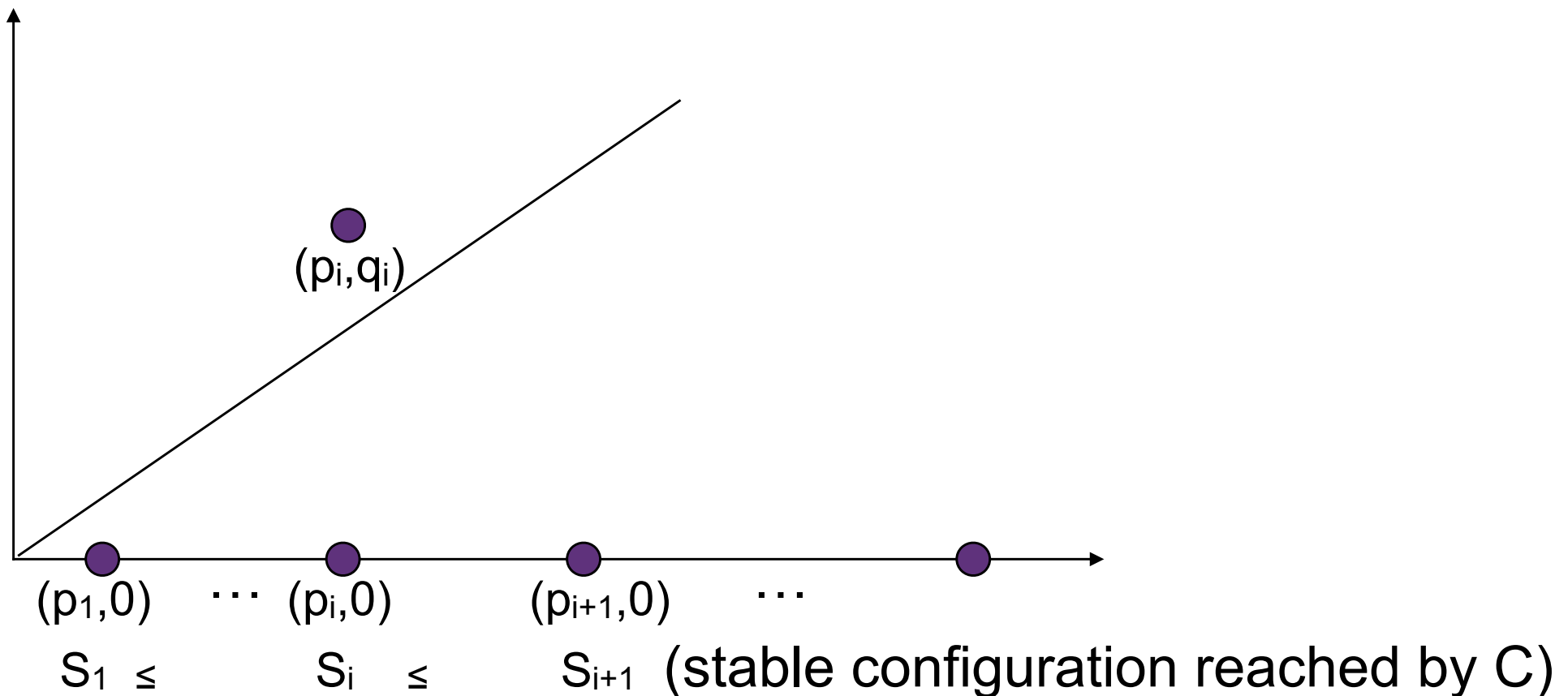
Example: $\max(n_1, n_2)$ is *not* output oblivious



What Functions Are Output Oblivious?

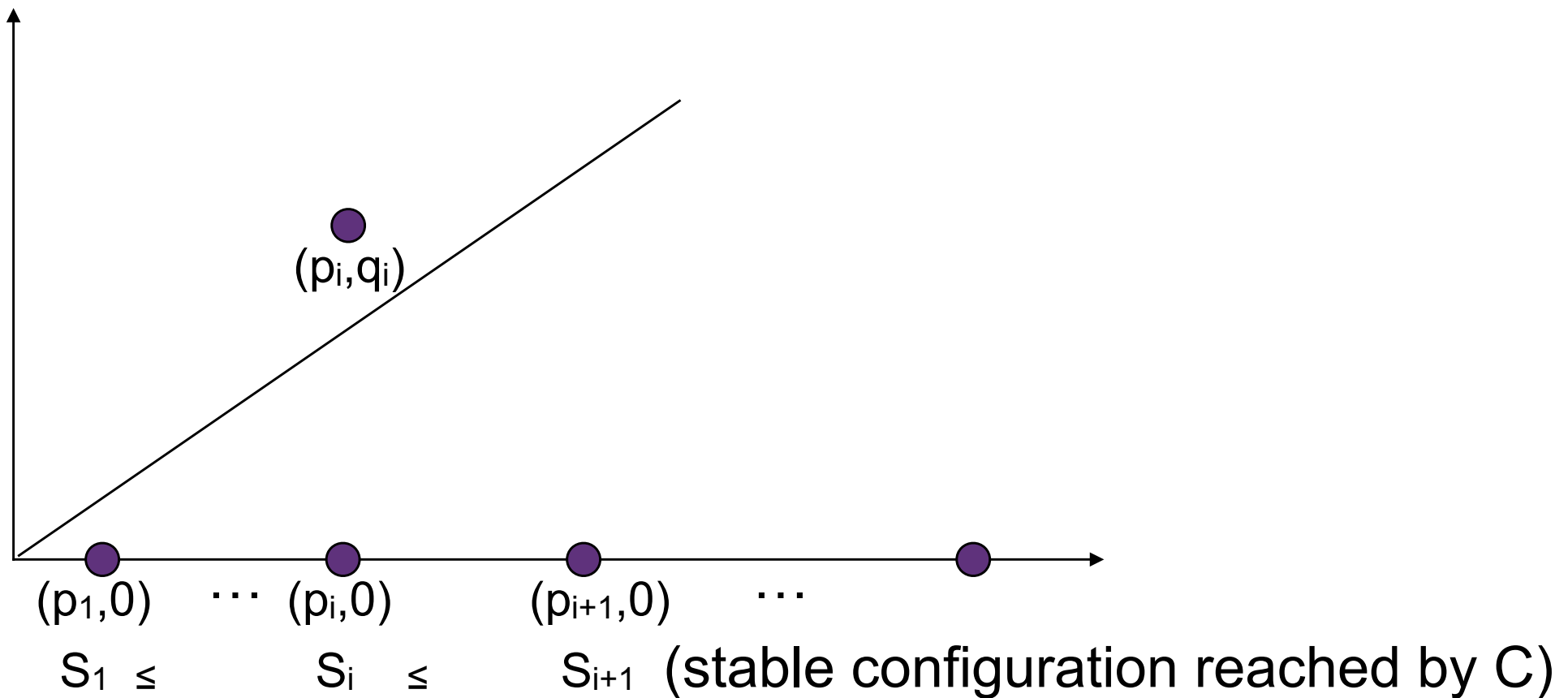
Example: $\max(n_1, n_2)$ is *not* output oblivious

Choose q_i such that $p_i < q_i < p_{i+1}$; then $\max(p_i, q_i) = q_i$



What Functions Are Output Oblivious?

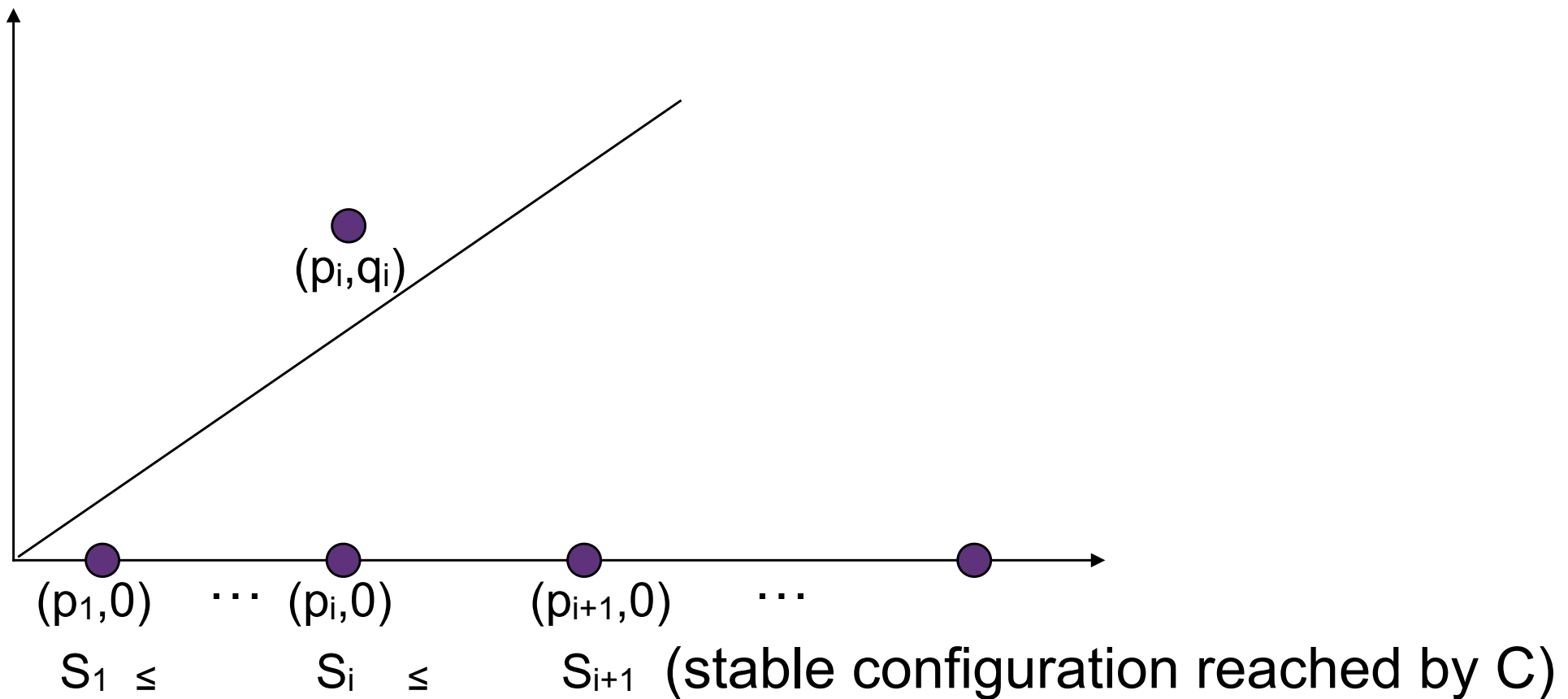
Example: $\max(n_1, n_2)$ is *not* output oblivious



What Functions Are Output Oblivious?

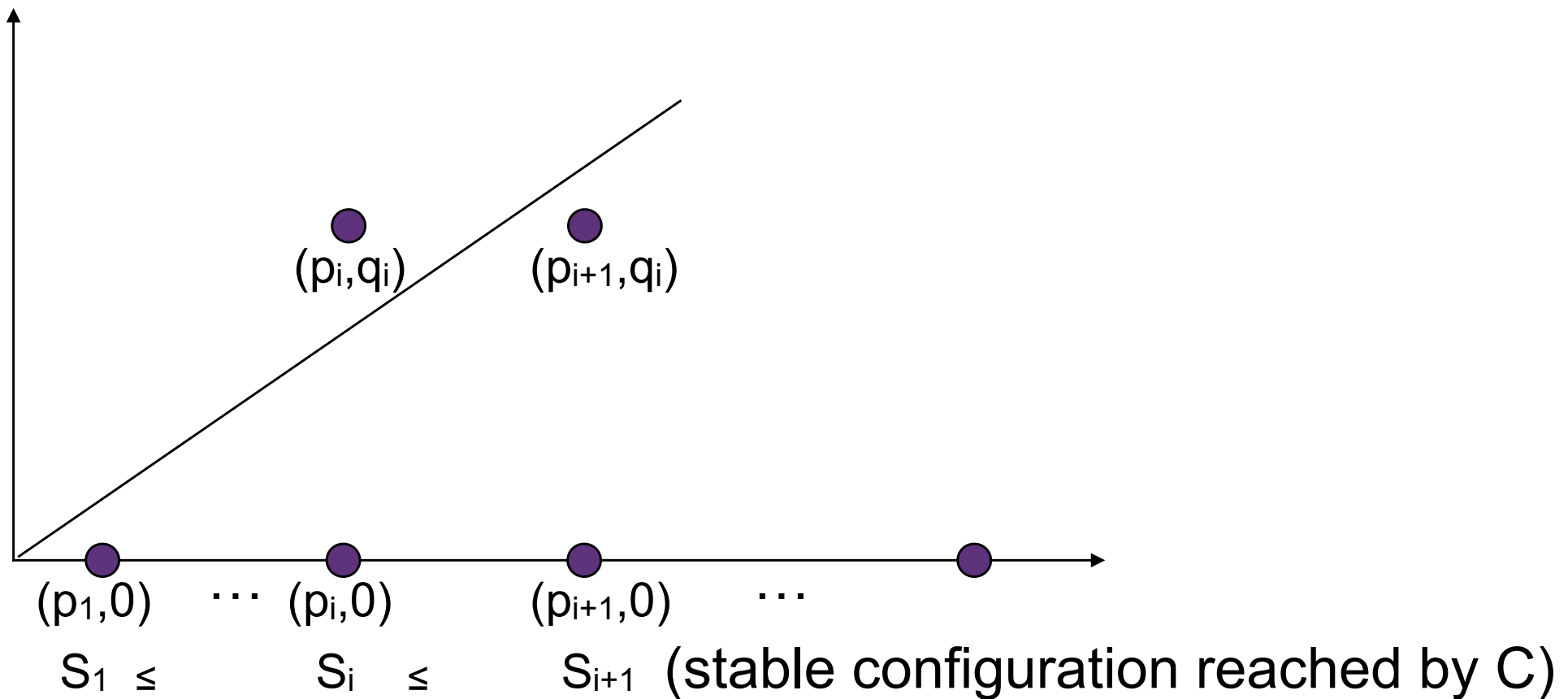
Example: $\max(n_1, n_2)$ is *not* output oblivious

There is computation of C on input (p_i, q_i) that first reaches S_i , producing p_i , and then goes on to produce $q_i - p_i$ additional outputs



What Functions Are Output Oblivious?

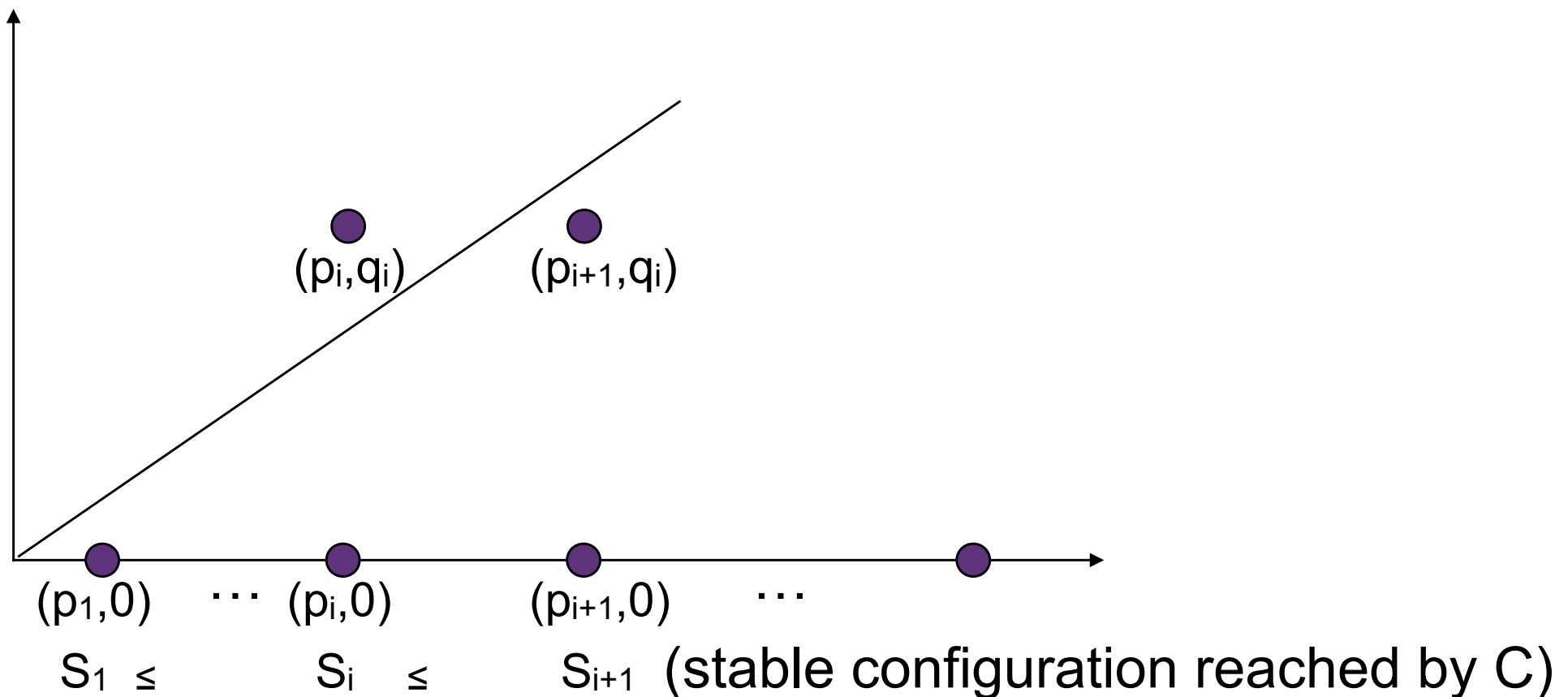
Example: $\max(n_1, n_2)$ is *not* output oblivious



What Functions Are Output Oblivious?

Example: $\max(n_1, n_2)$ is *not* output oblivious

But then on input (p_{i+1}, q_i) C can produce $p_{i+1} + (q_i - p_i)$ outputs, a contradiction



What Functions Are Output Oblivious?

One final output oblivious example...

A simple (non-decreasing) fissure function:

$$\begin{aligned} f(n_1, n_2) &= 3n_1 + 1, & n_1 < n_2 \\ &3n_1 (= 3n_2), & n_1 = n_2 \\ &3n_2 + 2, & n_1 \geq n_2 \end{aligned}$$

Stable Function Composition: Summary

A function can be stably computed by an output oblivious CRN with a leader if and only if it is the min of sums of non-decreasing mod and simple fissure functions

While semi linear functions can be stably computed by CRNs with or without a leader, the class of functions that can be stably computed by output oblivious CRNs without a leader is a subclass of those computable with a leader

Stable Function Composition: Open Problems

When is composition possible when functions are not output-oblivious (with the help of an intermediate CRN)?

In Closing

This talk focused on CRNs for approximate majority and output oblivious CRNs for stable function computation

There are many more fascinating problems on this topic!

... And the field of molecular programming encompasses many models other than CRNs